Redbooks
ibm.com/redbooks

Building Integrated Websites with IBM Digital Experience

Bernd Beilke

Darren Cacy

Dmitri Nevedrov

Dineshbabu Ramakrishnan

JayaPrasad Thambuganipalle

Basem Zaben

WebSphere

Redbooks

International Technical Support Organization

Building Integrated Websites with IBM Digital Experience

January 2016

SG24-8313-00

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

First Edition (January 2016)

This edition applies to IBM WebSphere Portal and Web Content Manager Version 8.5.

Contents

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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 grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

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 jurisdictions 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

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

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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 actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate 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. The sample programs are provided “AS IS”, without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

DataPower®
DB2®
developerWorks®
Enterprise Storage Server®
IBM®
IBM Connections™
IBM MobileFirst™
Passport Advantage®
PureApplication®
PureSystems®
Rational®
Redbooks®
Redbooks (logo) ®
Sametime®
WebSphere®

The following terms are trademarks of other companies:

SoftLayer, and SoftLayer device are trademarks or registered trademarks of SoftLayer, Inc., an IBM Company.

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

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

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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

Find and read thousands of IBM Redbooks publications

- ► Search, bookmark, save and organize favorites
- ► Get personalized notifications of new content
- ► Link to the latest Redbooks blogs and videos

Get the latest version of the Redbooks Mobile App

iOS **Download Now** Android

Promote your business in an IBM Redbooks publication

Place a Sponsorship Promotion in an IBM® Redbooks® publication, featuring your business or solution with a link to your web site.

Qualified IBM Business Partners may place a full page promotion in the most popular Redbooks publications. Imagine the power of being seen by users who download millions of Redbooks publications each year!

ibm.com/Redbooks
About Redbooks → Business Partner Programs

THIS PAGE INTENTIONALLY LEFT BLANK

Preface

A digital experience is a personalized experience that provides employees, customers, business partners, and citizens with a single point of interaction with people, content, and applications anywhere, anytime, and from any device.

The *IBM® Digital Experience* is a platform that is used to build powerful contextual websites. The strengths of the platform include the ability to mix applications and web content into a coherent user experience. Developers can build upon a prescriptive standard to build reusable building bricks, which can be used by line-of-business (LOB) users in a flexible way. LOB users can assemble pages from these building bricks and from rich web content. The page creation is performed inline by easy drag-and-drop operations without requiring sophisticated IT skills.

This IBM Redbooks® publication describes how a team can build a website starting from a new installation of Digital Experience. The book provides examples of the basic tasks that are needed to get started with building a proof-of-concept (PoC) website example. The resulting example website illustrates the value and key capabilities of the Digital Experience suite, featuring IBM WebSphere® Portal and IBM Web Content Management.

The target audiences for this book include the following groups:

- Decision makers and solution architects considering Digital Experience as a platform for their internal or external facing website.
- Developers who are tasked to implement a PoC and must be enabled to start quickly and efficiently, which includes the integration of existing back-end systems.
- A wide range of IBM services and sales professionals who are involved in selling IBM software and designing client solutions that include Digital Experience.

Note: The examples that are shown in this book are based on IBM WebSphere Portal and Web Content Manager Version 8.5 CF 08.

Authors

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

Bernd Beilke is a certified IT Specialist in the IBM Collaboration Solutions Brand based in Berlin, Germany. He joined IBM in 1997 as a Services Consultant and is working as a Client Technical Professional for Digital Experience Solutions in Germany, Europe, and the Middle East supporting customer engagements and doing field enablement and training. His areas of expertise include WebSphere Portal, eForms, and Social Software Solutions.

Darren Cacy is a technical sales specialist for IBM in the United States. He covers all of North America, providing deep technical skills in WebSphere Portal, Web Content Manager, and IBM Connections™. He has extensive experience in application development and design. Darren joined IBM in 1996 as an IT Architect and spent six years creating Internet and intranet websites for IBM customers. He moved to technical sales in 2002, and now spends his time showing customers how to get the most out of the IBM collaboration portfolio.

Dmitri Nevedrov has been with IBM as an IT Architect since 2007 and has spent time on application server, web services, middleware, and portal and content management technologies. During this time, he has worked on solution design, PoCs, and development. Before joining IBM, he worked at BEA Systems, contributing to development of WebLogic Portal and Workshop products, and at Quark, where he worked on publishing and image-processing tools. Dmitri wrote a number of tutorials and IBM developerWorks® papers on performance and integration methods. He has PhD and masters degrees in physics and electrical engineering.

Dineshbabu Ramakrishnan is an Advisory Software Engineer working for IBM in the United States. He is the team lead for the WebSphere Portal and Web Content Manager Level 2 support team providing deep technical skills and support on the Digital Experience product portfolio to customers worldwide. He has over 15 years of experience in both development and support of IBM WebSphere Application Server, WebSphere Portal Server, Web Content Manager, and has been working on WebSphere products since 2001. He is also an IBM Certified Solution Developer and IBM Certified System Administrator for WebSphere Portal and Web Content Manager.

JayaPrasad Thambuganipalle is a Senior Accelerated Value Leader (AVL) in US. Before becoming an AVL, JP was a lead architect in IBM Enterprise Storage Server® lab services. Before joining IBM, JP worked with top five global investment and retails banks as an infrastructure specialist and architect. He has presented extensively at various IBM conferences, and authored several IBM Redbooks publications and white papers.

Basem Zaben is a Chief Architect and Development Manager at Elm. Elm is a Saudi Arabia company that has been successfully providing services, products and solutions over the past decade to both public and private sectors in the local Saudi Arabia market. Basem’s areas of expertise includes Java Platform, Enterprise Edition, WebSphere Portal and Web Content Manager, IBM ECM, IBM MobileFirst™, IBM BPM, IBM DataPower®, IBM Connections, IBM CLM, IBM Web Experience Factory, Applications and Infrastructures Security. With 15 years of experience in the IT industry, Basem has a deep level of knowledge about how to deliver and build secure and highly performing solutions and enterprise architectures in industries such as banking, telecom, government, education, oil, and gas.

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

Thomas Stober
STSM, Lead Architect, WebSphere Portal Foundation, IBM Deutschland

Stefan Hepper
Digital Experience Development- Rich Media Edition Architect, Master Inventor, IBM US

Jonathan Booth
Senior Architect, STSM, Digital Experience Tooling, IBM US

Jonathan J. Lidaka
Front-end Engineer, Digital Experience and Mobile Development, IBM US

Navneet Goyal
Senior Developer, Digital Experience, IBM US

Faisal Taimoor Khan
Technical Integration Manager at Elm, Saudi Arabia

Howard Krovetz
WebSphere Portal Test, IBM US

Marcela Adan
Deb Landon
Ann Lund
International Technical Support Organization

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

IBM Digital Experience overview

This chapter introduces the concept of *IBM Digital Experience* and the business drivers for organizations to build meaningful and effective digital experiences for their customers and employees.

This chapter describes the components of IBM Digital Experience and how they can be combined to build an exceptional digital experience. It also introduces the Digital Experience offerings that are available to create and deliver digital experience capabilities to your customers and employees.

The remaining chapters of this book describe how to get started by implementing some of the core components of a digital experience to create an integrated website by using tools and artifacts that are provided as part of the Digital Experience offerings.

This chapter covers the following topics:

- 1.1, “What is a digital experience” on page 2
- 1.2, “IBM Digital Experience capabilities” on page 3
- 1.3, “IBM Digital Experience solution offerings and products” on page 15

1.1 What is a digital experience

A digital experience is a *personalized experience* that provides employees, customers, business partners, and citizens with a single point of *interaction with people, content, and applications* anywhere, anytime, and from any device. A digital experience platform aggregates content from various sources into a unified user experience allowing people to make better informed decisions.

1.1.1 Business drivers and value

When organizations use all the capabilities that a true digital experience can provide, they can realize real business benefits. They can see benefits and find value in various business processes. Providing outstanding digital experiences to customers and business partners and better engaging employees generates more successful business outcomes.

When customers are more engaged, they are more likely to recommend the products and services of a company and become loyal and trusted customers. When you make it easier for your business partners to conduct business with you, stronger partnerships are formed.

The digital experience yields greater overall organizational efficiencies, including targeted marketing, anticipating customer issues, breaking down communication barriers, and more.

Exceptional online experiences drive better business results for several reasons:

- Improved customer service: Companies can achieve a reduction in customer defection rate and as a result increase profits.
- Increased sales: Companies can increase sales revenue and improve efficiencies.
- Maximize employee productivity: Organizations can retain the knowledge of staff within the company and increase customer satisfaction.
- Increased visibility: Companies can increase website traffic.
- Faster product development: Companies can develop and bring new products to market faster.
- Reduce employee turnover: Organizations can reduce turnover as highly engaged employees are less likely to leave their organizations than highly disengaged employees.

1.1.2 Business results

There are several examples of organizations achieving greater business results by implementing solutions that are based on the IBM Digital Experience platform. Here are some of those examples:

- Faster product development: An organization with a product development team scattered around the world brought new products to market more quickly by using a solution based on IBM WebSphere Portal and collaboration technologies.
- Extend and improve the quality of service: A healthcare organization delivered new digital services and enhanced care coordination by deploying collaborative intranet and extranet capabilities. The solution (interactive patient portal and mobile applications that are built on Digital Experience software) allows them to extend their physical healing environment into a virtual environment, providing capabilities that go beyond simple access to personal health records.

- Increased market exposure: An organization increased their product brand recognition by developing mini-sites that are targeted at specific audiences. They also shared the mini-sites on social media.
- Improved IT staff's productivity: A private college with stakeholders that include students, staff, alumni, and prospects has limited resources. Their IT staff was challenged to provide online information and applications for the different audiences and security requirements. They developed three portals by using Digital Experience software, including IBM WebSphere Portal integrated with IBM software for communications, collaboration, content management, and social business. The ease of application development improved the IT staff productivity, enabling them to deploy applications that delivered better service to students, alumni, and the public.

1.2 IBM Digital Experience capabilities

Figure 1-1 shows an overview of the capabilities in IBM Digital Experience. Each component that is shown in Figure 1-1 represents a powerful and comprehensive part of Digital Experience.

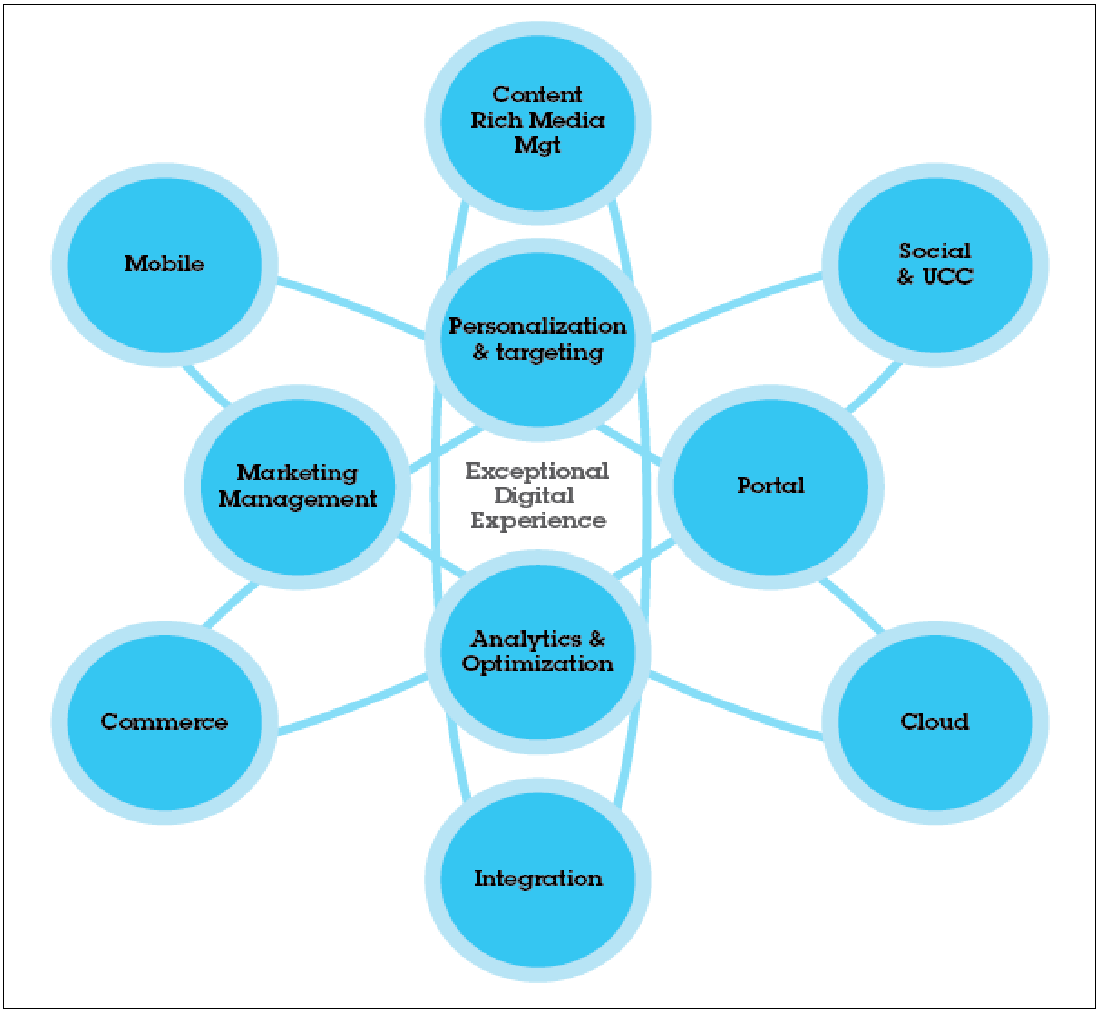


Figure 1-1 Digital Experience capabilities

Digital Experience is a robust, integrated platform that is built for flexibility with the following attributes:

- Integrated: Speed time to market with a core platform.
- Open: Use and extend existing data sources through robust integration capabilities.
- Instrumented: Support evidence-based decision making through analytics.
- Contextual: Increase customer satisfaction with digital experiences that are tailored to user location, attributes, preferences, device, and behaviors.
- Flexible: Adapt to handle the latest advancements and technology changes.
- Line-of-business (LOB)-friendly: Enable LOB to manage the digital experience lifecycle.

The following sections describe in more detail some of the digital experience capabilities that are shown in Figure 1-1 on page 3:

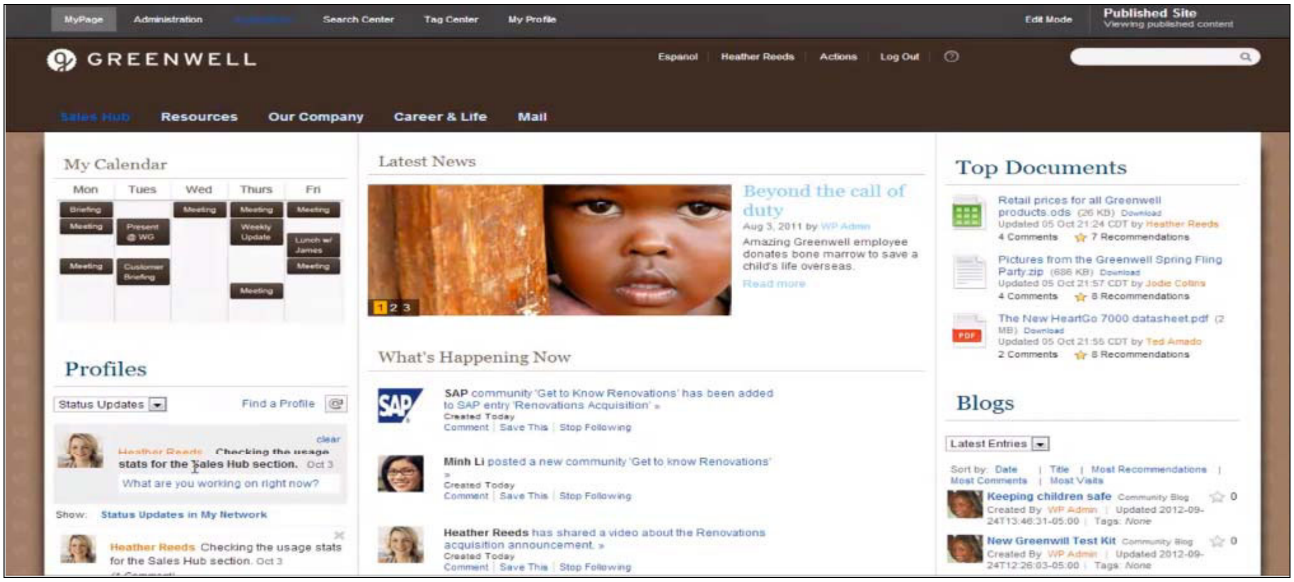
- 1.2.1, “Portal” on page 4
- 1.2.2, “Content management” on page 5
- 1.2.3, “Target content” on page 7
- 1.2.4, “Social” on page 7
- 1.2.5, “Integration” on page 8
- 1.2.6, “Analytics” on page 11
- 1.2.7, “Cloud” on page 12
- 1.2.8, “E-commerce” on page 13

1.2.1 Portal

IBM WebSphere Portal is the foundation of the digital experience capabilities. It is the infrastructure that holds everything together. Customers can use WebSphere Portal to manage and create their digital properties in a secure way. It includes the following properties:

- A framework for creating digital web experiences so you can integrate and aggregate multiple back-end systems together into a single point of entry for your customers and employees.
- A secured platform based on IBM WebSphere Application Server and all the IBM security features that come with this proven technology.
- Powerful globalization capabilities so you can engage your users at a global level.
- Provides capabilities to control navigation and presentation throughout the entire digital experience. WebSphere Portal makes it simple to brand, create, manage, secure, and deploy digital properties for customers and employees.

Figure 1-2 shows an example of an engaging digital experience based on WebSphere Portal bringing together various applications and capabilities.



Beyond the call of duty
Aug 3, 2011 by WP Admin
Amazing Greenwell employee donates bone marrow to save a child's life overseas.
Read more

SAP community 'Get to Know Renovations' has been added to SAP entry 'Renovations Acquisition' »
Created Today
Comment | Save This | Stop Following

Minh Li posted a new community 'Get to know Renovations' »
Created Today
Comment | Save This | Stop Following

Heather Reeds has shared a video about the Renovations acquisition announcement. »
Created Today
Comment | Save This | Stop Following

Retail prices for all Greenwell products.ods (26 KB) Download
Updated 05 Oct 21:24 CDT by Heather Reeds
4 Comments ☆ 7 Recommendations

Pictures from the Greenwell Spring Fling Party.zip (686 KB) Download
Updated 05 Oct 21:57 CDT by Jodie Collins
4 Comments ☆ 8 Recommendations

The New HeartGo 7000 datasheet.pdf (2 MB) Download
Updated 05 Oct 21:55 CDT by Ted Amado
2 Comments ☆ 8 Recommendations

Heather Reeds Checking the usage stats for the Sales Hub section. Oct 3
What are you working on right now?

Heather Reeds Checking the usage stats for the Sales Hub section. Oct 3

Keeping children safe Community Blog ☆ 0
Created By WP Admin | Updated 2012-09-24T13:46:31-05:00 | Tags: None

New Greenwill Test Kit Community Blog ☆ 0
Created By WP Admin | Updated 2012-09-24T12:26:03-05:00 | Tags: None

Figure 1-2 WebSphere Portal, the foundation of the digital experience

To summarize, WebSphere Portal provides the following capabilities:

- Provides the foundation framework for the digital experience.
- Manages the digital brand.
- Creates and manages pages, the site map, and navigation.
- Integrates and aggregates applications and content.
- Populates content and applications by using the drag-and-drop method.
- Engages global audiences.
- Secures access to resources.

The following sections describe some of the components that you can add to the foundation that is provided by WebSphere Portal to create an exceptional digital experience.

1.2.2 Content management

To start building the structure over the foundation of WebSphere Portal, you must create and assemble content as part of your digital experience. IBM Web Content Manager facilitates intuitive content creation for crafting messages that resonate with the target audience.

Web Content Manager provides capabilities for LOB professionals to create easily content through drag-and-drop capabilities, inline editing, and rich media browsing. A LOB person can use it to assemble and create a digital experience. The ease of use that is provided by Web Content Manager is important because LOB users without in-depth IT skills own the content and are responsible for creating and maintaining it.

Web Content Manager provides powerful content creation and collaborative management tools that LOB professionals and content owners can use to create quickly and manage rich digital content without IT involvement. LOB professionals must be able to control the content. Web Content Manager provides the tools that they need to be productive through the content management lifecycle.

Web Content Manager capabilities allow LOB users to create, assemble, and rearrange on-screen content and add rich media. Users can attach their content to a workflow and submit it to an approval process before publishing the content.

Figure 1-3 shows an example of powerful content creation by using Web Content Manager with inline editing and digital asset management, which is part of IBM Web Content Manager Rich Media Edition.

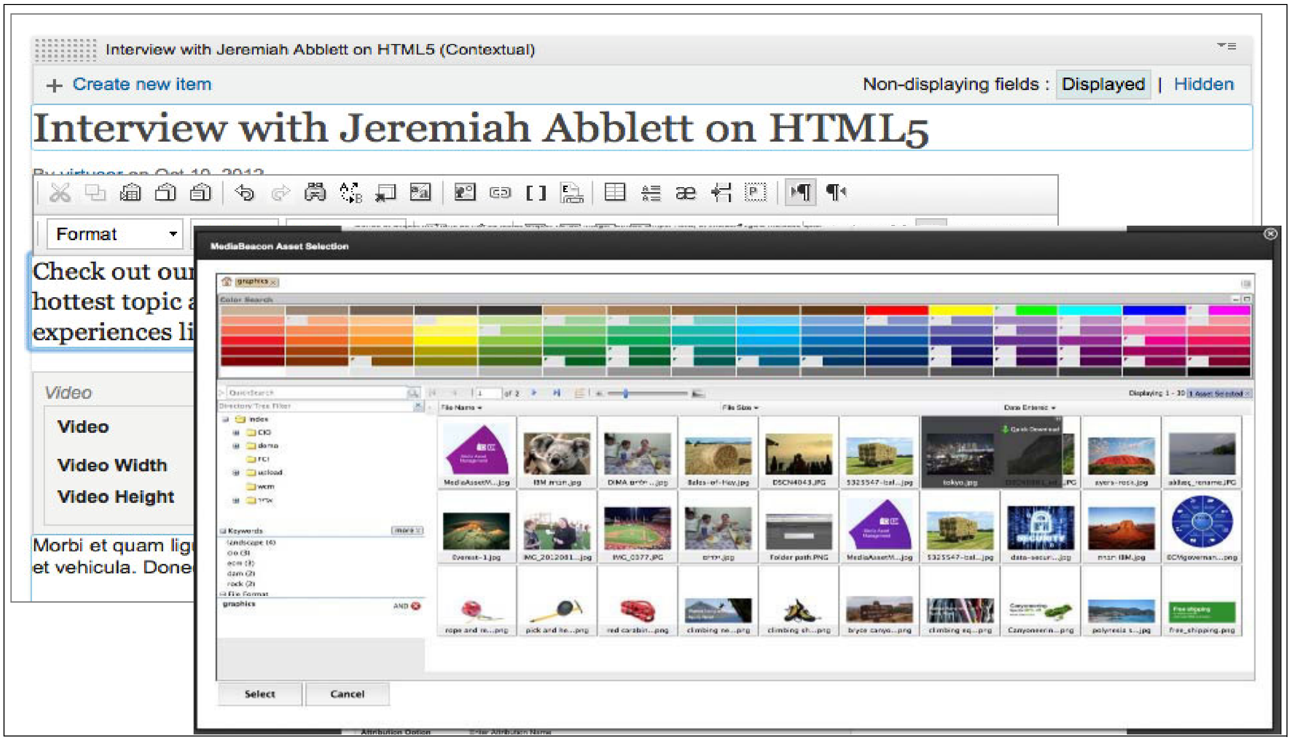


Figure 1-3 Web Content Manager

To summarize, Web Content Manager provides the following capabilities:

- Enables LOB professionals to create, approve, make versions of, publish, and syndicate web content through an intuitive, dynamic, and contextually aware toolbar and built-in project management features.
- Includes an extensive catalog of prebuilt, digital content templates and lightweight themes to support responsive design for microsite and web page creation.
- Provides flexible workflows to help ensure that the content lifecycle follows business processes and approvals before going live.
- Provides advanced rich text editor capabilities, including social media integration.
- Supports rich media and HD video integration.
- Enables content governance with web content analytics.
- Helps you keep relevant content in front of users and measure and share content popularity through content rating and tagging.
- Manages creation of multilingual content.

1.2.3 Target content

The structure includes the portal foundation and the tools to create and manage compelling content. The next step is to target and personalize the content to provide targeted experiences and reach customers, regardless of how they want to be reached, including mobile device delivery. Targeting content results in more productive employees or better and deeper relationships with customers and business partners.

Personalization allows a portal or website to choose which content appears for particular website visitors. Personalization capabilities enable LOB professionals to individualize messages to speak directly to individual audience needs.

For example, a site that uses personalization can show different news articles to managers than to regular employees, give users access to applications based on their role, or target different information to valued customers. You can also personalize the experience based on the device the user is accessing the experience from, for example, a tablet, smartphone, or notebook.

To summarize, by targeting the digital experience, you can accomplish the following tasks:

- Deliver an optimized and personalized digital experience through relevant and targeted, dynamic content delivery that can be based on role, preferences, and real-time analytic features, tracking things such as clicks, browser, device, identity, location, and time of day.
- Increase employee productivity by making it easier and faster to find information and access personalized content and applications.

1.2.4 Social

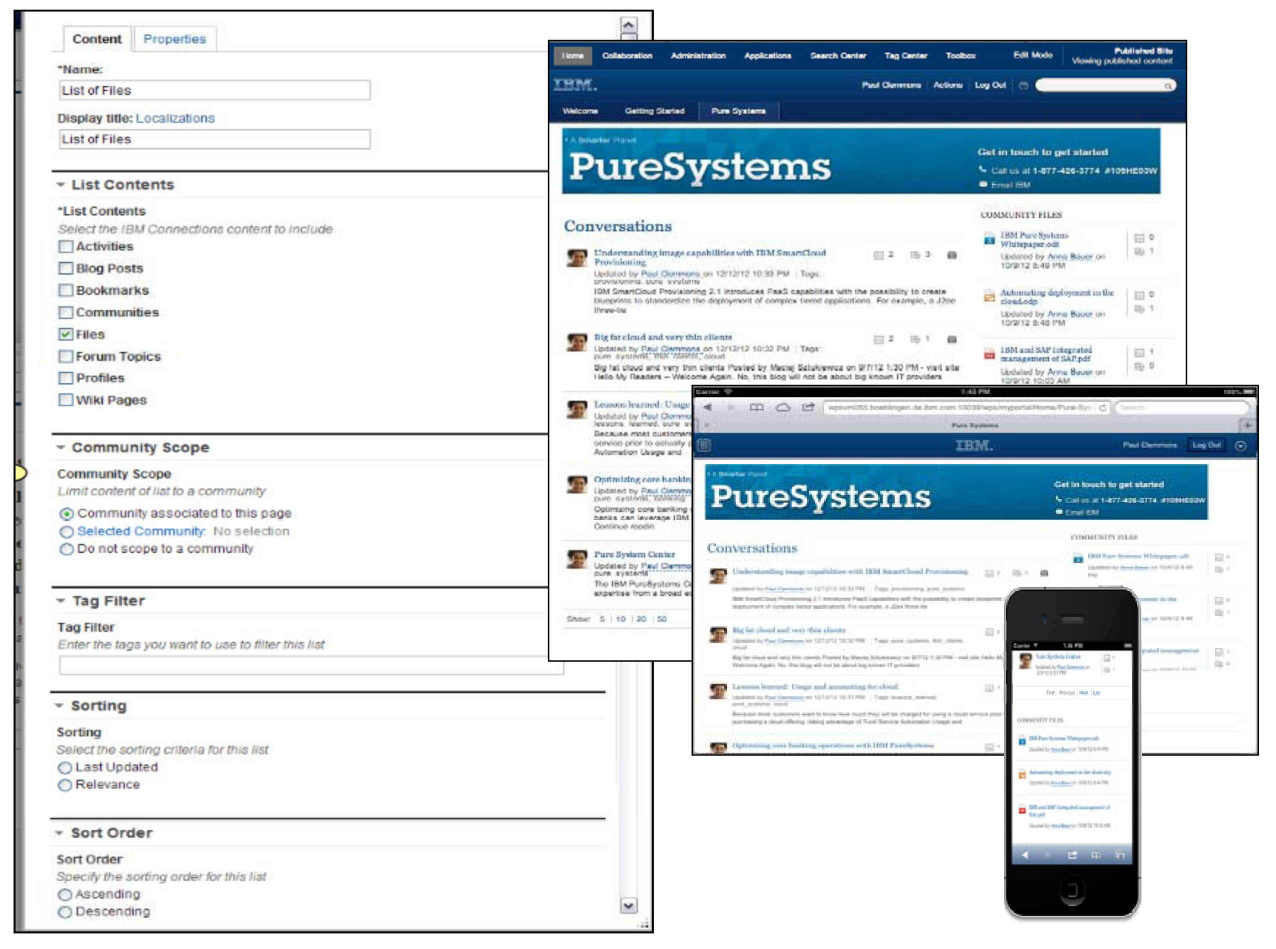
Businesses rely on social networks and online communities to collaborate with peers, business partners, and extended teams. They build and reinforce these relationships through the ability to communicate and share information and experiences without delay, wherever the users go and through the devices of their choice.

The emergence of social business as a major component of today’s business strategies reflects the many ways that technology is enhancing, and often driving, interpersonal connections between customers, colleagues, suppliers, and business partners.

Customers increasingly rely on these relationships, and recommendations from their social networks, to guide purchasing decisions. Employees use social tools to foster collaboration, make better business decisions, connect to subject matter experts (SMEs), and to work more efficiently. Supply chain business partners use these same social tools to predict demand, optimize supply chain mechanics, eliminate process bottlenecks, and shorten time to value.

More organizations are embedding social capabilities into the digital experience they create for their customers and employees. WebSphere Portal page editors can use social rendering to feature social data that is hosted on a remote IBM Connections server in the context of portal pages. They can create web content items that present, for example, a list of specific blog posts, files, or discussion topics. As a result, page editors can focus on assembling meaningful portal pages that are enriched by social data.

Figure 1-4 shows an example of rendering social data that is hosted on a remote IBM Connections server in the context of portal pages and on multiple devices.



Content | Properties

*Name:
List of Files

Display title: Localizations
List of Files

List Contents

*List Contents
Select the IBM Connections content to include
□ Activities
□ Blog Posts
□ Bookmarks
□ Communities
☑ Files
□ Forum Topics
□ Profiles
□ Wiki Pages

Community Scope

Community Scope
Limit content of list to a community
● Community associated to this page
○ Selected Community: No selection
○ Do not scope to a community

Tag Filter

Tag Filter
Enter the tags you want to use to filter this list

Sorting

Sorting
Select the sorting criteria for this list
○ Last Updated
○ Relevance

Sort Order

Sort Order
Specify the sorting order for this list
○ Ascending
○ Descending

Home
Collaboration
Administration
Applications
Search Center
Tag Center
Toolbox
Edit Mode
Published Site
Viewing published content
IBM
Paul Clemmons
Actions
Log Out
Welcome
Getting Started
Pure Systems
PureSystems
Get in touch to get started
Call us at 1-877-426-3774 #109HE03W
Email IBM
Conversations
COMMUNITY FILES
Understanding image capabilities with IBM SmartCloud Provisioning
Big fat cloud and very thin clients
Lessons learned: Usage
Optimizing core banking
Pure System Center
IBM PureSystems Whitepaper.odt
Automating deployment in the cloud.odp
IBM and SAP integrated management of SAP.pdf
Updated by Anna Bauer on 10/9/12 8:48 PM
Show: 5 | 10 | 20 | 50

Figure 1-4 Embedding social data in the digital experience

To summarize, by embedding social data into the digital experience, you can accomplish the following tasks:

- Engage customers to deepen relationships, capture insights, and unleash innovation and productivity.
- Provide a consistent and single source for integrated business applications, content, and social capabilities.
- Use social rendering capabilities to easily infuse IBM Connections social services, for example, blogs, wikis, and files into business applications.
- Easily customize how social content is displayed within the digital experience.
- Provide a multi-channel delivery with anytime or anywhere rendering.

1.2.5 Integration

Up to this point, the digital experience structure that was built on the portal foundation was centered on content. Content is important, but what really differentiates the digital experience and makes it even more powerful is to surround the content with transactional integration capabilities.

Integration capabilities can use your existing IT investments and web assets, further extending their value.

Integration is a key capability of the IBM WebSphere Portal platform that enables organizations to integrate and use the existing back-end infrastructure in their digital experience. It features ready-to-use connectors and RESTful services that can facilitate seamless integration with your enterprise and desktop applications, commerce solutions, web and cloud-based services, widgets and portlets, data analytics solutions, and more.

As shown in Figure 1-5, WebSphere Portal provides a wide range of integration capabilities, from simple options that allow an agile design and integration approach and do not require advanced programming skills, to more complex options that provide more advanced adaptability and personalization capabilities.

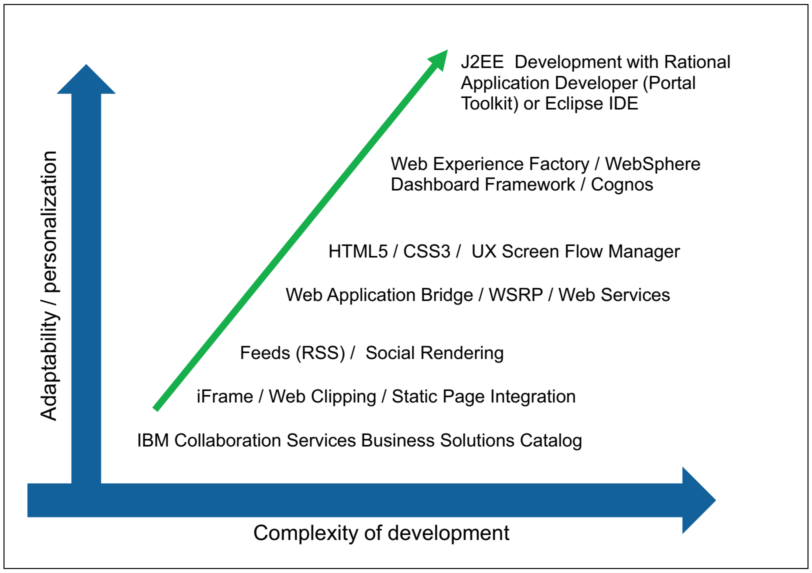

J2EE Development with Rational Application Developer (Portal Toolkit) or Eclipse IDE
Web Experience Factory / WebSphere Dashboard Framework / Cognos
HTML5 / CSS3 / UX Screen Flow Manager
Web Application Bridge / WSRP / Web Services
Feeds (RSS) / Social Rendering
iFrame / Web Clipping / Static Page Integration
IBM Collaboration Services Business Solutions Catalog
Adaptability / personalization
Complexity of development

Figure 1-5 WebSphere Portal range of integration capabilities

To summarize, here are the main characteristics of WebSphere Portal integration capabilities:

- Deliver new return on investment (ROI) from existing technology investments.
- Provide a wide range of fast and cost-effective UI integration options.
- Provide a comprehensive framework for standards-based custom application development using Eclipse foundation.
- Support script-based development.

Mobile

To be successful in the marketplace, organizations must make the digital experience available anywhere, anytime, and from any device. To accomplish this goal, the mobile experience must render successfully and consistently on mobile devices.

Creating an experience for multiple devices is essential, but targeting the correct set of devices can be challenging. Responsive web design (RWD) is a popular approach for creating a single website that optimizes content and layout automatically based on screen size, device, and orientation, eliminating the need to design for a specific user device preference.

RWD provides content parity between mobile devices and desktop channels, which enhances user experience and brand consistency. Seamless changes in screen size, from small to large, are now possible while the order of the content is maintained. Content maintenance is simplified by having one site that is represented by one set of assets.

WebSphere Portal provides a responsive website. It ships with themes and templates that are based on RWD technology that dynamically formats the UI and adjusts it to the appropriate screen size. If the user accesses the experience from a smartphone, the experience automatically adjusts itself to the screen size. If the user accesses the experience from a tablet, the experience dynamically reformats and updates itself to take advantage of the additional screen space that is available.

WebSphere Portal themes are also based on adaptive design. This technology queries the device to push the appropriate CSS stylesheet and formatting.

WebSphere Portal ready-to-use themes provide two new responsive page navigation designs for mobile devices. One is aimed at smartphones, and the other is designed for tablets. The user agent for a device is parsed to determine which navigation to render on the portal page.

Besides the native responsive and adaptive design in the WebSphere Portal themes, the following resources and products are available:

- WebSphere Portal Mobile Experience

  The WebSphere Portal Mobile Experience demonstrates how to use a lightweight architecture that provides exceptional performance on mobile devices. You can start with the included sample code and then customize it to meet your specific business needs.

  The Mobile Experience includes the following features:

  - Sample mobile navigation widgets that follow common mobile application navigation patterns.
  - Sample mobile page layouts that demonstrate common content interaction patterns.

  The Mobile Experience is available on the IBM Collaboration Solutions Catalog, which is available at the following website:

  `https://greenhouse.lotus.com/catalog/home_full.xsp?fProduct=WebSphere%20Portal&fSearchValue=mobile`

- IBM Mobile Portal Accelerator

  If your site must support a broad range of mobile devices, Mobile Portal Accelerator provides a mobile multi-channel server solution. Mobile Portal Accelerator offers a multi-channel server and mobile device repository that can automate web page presentation to over 8,400 mobile devices. Each of those devices can be optimized to the display and services capabilities that include smartphones, tablets, handheld phones, and kiosk devices. Site developers can write the content. Then, the multi-channel server solution automates the presentation for optimized display on thousands of mobile devices.

  For more information, see the IBM Mobile Portal Accelerator website:

  `http://www.ibm.com/software/products/en/ibmmobiportacce`

- Development tools

  Both IBM Web Experience Factory and IBM Rational Application Developer include functions to help you develop exceptional web experiences for your mobile site visitors:

  - Web Experience Factory includes new smartphone builders and samples to help developers generate mobile portlets and pages faster.
  - IBM Rational® Application Developer includes tools that are designed to help you develop portlet applications for WebSphere Portal.

- IBM MobileFirst

  You can integrate WebSphere Portal with IBM MobileFirst to extend applications and content to a wide range of mobile devices so you can create a rich cross-platform hybrid mobile experience and provide multi-channel support to your web communities. You can use MobileFirst to create hybrid applications that add native device functions to your portal.

To summarize, here are some of the mobile capabilities within IBM Digital Experience:

- RWD web experience themes that dynamically match the mobile device form factor and support customer interaction and navigation patterns.
- Adaptive design by using device classification capabilities.
- Built on open web technologies (HTML5, CSS3, and JavaScript).
- Use of on device features (for example, camera, secure auxillary storage, accelerometer, and maps) provided by IBM MobileFirst integration.
- Reduce bandwidth requirements and improve responsiveness for mobile applications.

1.2.6 Analytics

Eventually, you want to start tracking how customers and employees are using the digital experience. You can use real-time analytics to understand how the digital experience is affecting your business and to optimize the digital experience.

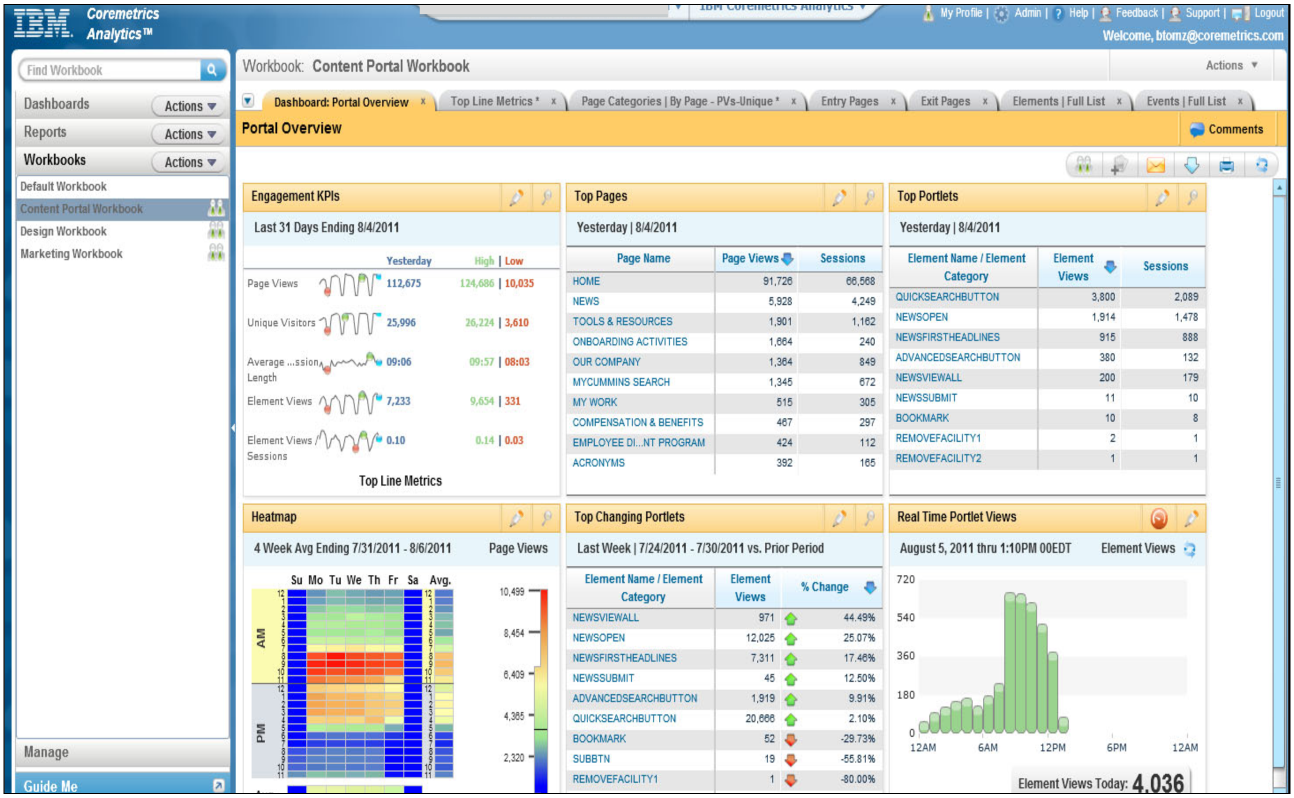
Web analytics page overlays can deliver in-place, real-time web page and channel delivery analysis. You can tag pages, portlets, or web content, and measure results through campaign and custom tags for greater insight into web content value and user and channel interactions.

You can use WebSphere Portal Active Site Analytics to show graphical statistics reports about individual portal resources, such as pages or portlets. These reports are called *Active Site Analytics overlay reports*.

With WebSphere Portal Integration for Enterprise Marketing Management, you can use IBM Marketing Center or IBM Interact to deliver targeted offers to a spot on the page.

You can also integrate IBM Digital Analytics (previously Coremetrics) with WebSphere Portal. The real-time information that is captured by WebSphere Portal Active Site Analytics can be sent to IBM Digital Analytics (or a third-party analytics provider) for processing.

Figure 1-6 shows a report analyzing the information that is collected by a WebSphere Portal website.



Engagement KPIs

Last 31 Days Ending 8/4/2011

	Yesterday	High | Low
Page Views	112,675	124,686 | 10,035
Unique Visitors	25,996	26,224 | 3,610
Average ...ssion Length	09:06	09:57 | 08:03
Element Views	7,233	9,654 | 331
Element Views / Sessions	0.10	0.14 | 0.03

Top Line Metrics

Top Pages

Yesterday | 8/4/2011

Page Name	Page Views	Sessions
HOME	91,726	66,568
NEWS	5,928	4,249
TOOLS & RESOURCES	1,901	1,162
ONBOARDING ACTIVITIES	1,664	240
OUR COMPANY	1,364	849
MYCUMMINS SEARCH	1,345	672
MY WORK	515	305
COMPENSATION & BENEFITS	467	297
EMPLOYEE DI...NT PROGRAM	424	112
ACRONYMS	392	165

Top Portlets

Yesterday | 8/4/2011

Element Name / Element Category	Element Views	Sessions
QUICKSEARCHBUTTON	3,800	2,089
NEWSOPEN	1,914	1,478
NEWSFIRSTHEADLINES	915	888
ADVANCEDSEARCHBUTTON	380	132
NEWSVIEWALL	200	179
NEWSSUBMIT	11	10
BOOKMARK	10	8
REMOVEFACILITY1	2	1
REMOVEFACILITY2	1	1

Top Changing Portlets

Last Week | 7/24/2011 - 7/30/2011 vs. Prior Period

Element Name / Element Category	Element Views		% Change
NEWSVIEWALL	971	▲	44.49%
NEWSOPEN	12,025	▲	25.07%
NEWSFIRSTHEADLINES	7,311	▲	17.46%
NEWSSUBMIT	45	▲	12.50%
ADVANCEDSEARCHBUTTON	1,919	▲	9.91%
QUICKSEARCHBUTTON	20,666	▲	2.10%
BOOKMARK	52	▼	-29.73%
SUBBTN	19	▼	-55.81%
REMOVEFACILITY1	1	▼	-80.00%

Figure 1-6 Analyze content and site usage

To summarize, by adding analytics to your digital experience, you can accomplish the following goals:

- Deliver real-time insight at a glance.
- Generate reports, dashboards, and workbooks to help you analyze content, marketing, and site design effectiveness.
- Segment your users to understand their tendencies and interests.
- Collect granular data to understand specific portlet performance and user navigation.
- Recognize trends and predict behaviors from directly within the context of the digital experience.

1.2.7 Cloud

Digital Experience on Cloud solutions provide a hosted, fully managed digital experience platform for delivering accelerated time to market. Companies can build and deliver compelling, socially aware, omnichannel digital experiences for their customers, business partners, and employees on the devices of their choice, more quickly and with lower costs.

IBM Digital Experience on Cloud solutions are provisioned, deployed, customized, monitored, and maintained by IBM experts to ensure security compliance, continuity of operations, and back-end integrations. To ensure high availability, daily backups, and disaster recovery of data centers, IBM PureApplication® patterns for Digital Experience are deployed to the IBM SoftLayer® cloud infrastructure. Additionally, an IBM Client Success Manager ensures smooth onboarding, adoption, and continued usage by acting as your IBM liaison.

1.2.8 E-commerce

IBM WebSphere Commerce is a software platform framework for e-commerce. It includes marketing, sales, customer, and order processing functions in a tailorable and integrated package.

With WebSphere Commerce, the entire end-to-end shopping experience is supported by a wide variety of marketing and merchandising capabilities. IBM WebSphere Commerce Enterprise is an omnichannel e-commerce platform that enables business-to-consumer and business-to-business sales to customers across multiple channels: web, mobile, social, store, and phone. It supports better marketing, selling, and fulfillment with precision marketing, merchandising tools, site search, customer experience management, catalog and content management, social commerce, and advanced starter stores.

E-commerce is no longer simply about presenting and selling products. It is about delivering a smarter shopping experience that is seamless and integrated across all customer touch points. Simply delivering a successful transaction is not enough. Your organization must build long-lasting online relationships with customers.

Successful web experience strategies address the intricate inter-play between online marketing, sales, and customer service throughout the customer interaction lifecycle. From a process standpoint, to enable successful online customer relationships often means that you must carefully weave together functional business applications, information, business processes, and other organizational resources that go beyond the core online commerce domain.

WebSphere Portal and WebSphere Commerce work together to support better customer web experiences in several different use cases. For example, a customer self-service website scenario illustrates how WebSphere Portal and WebSphere Commerce work together to bring to a single point all of the required applications, data, content, and tasks that are required to support a specific customer need. In this scenario, a customer performs a number of self-service activities:

- Running online processes and services
- Researching and resolving problems and questions
- Joining communities of interest
- Managing bills and loyalty programs
- Researching products, policies, and services

To ensure a positive web experience, it is necessary to integrate many enterprise processes, applications, and data sources together into a compelling web interface with personalization for the correct content and channel. WebSphere Portal is the primary presentation layer with WebSphere Commerce delivering targeted order management services as part of an integrated web user experience. The experience relies on other enterprise applications, cloud services, social capabilities from inside and outside the organization, and a robust content management solution that simplifies management, editing, and integration of content from many enterprise sources. The integration capabilities of IBM Digital Experience that are described in 1.2.5, “Integration” on page 8 are key to implementing this scenario.

In other examples, such as commerce sites, organizations can choose to use WebSphere Commerce as the user interface and use services and content from WebSphere Portal selectively. Web Content Manager provides dynamic content to support precision marketing and catalog content, and IBM Connections provides dedicated social applications.

IBM WebSphere Commerce, with the Web Content Management capabilities of Digital Experience, provides a powerful solution.

Here are some benefits of using WebSphere Commerce and Web Content Management together:

- Decrease IT costs for site development and updates by putting business users in control of promotional site content.
- Increase customer conversion rates through personalized offers and dynamic content.
- Increase site *stickiness* with rich media and compelling content.
- Increase visibility of particular catalog categories or items.
- Increase order size by using e-spots for cross-sell and up-sell opportunities.

Figure 1-7 on page 15 shows typical WebSphere Commerce and IBM Digital Experience integration scenarios:

- Portal as the front end

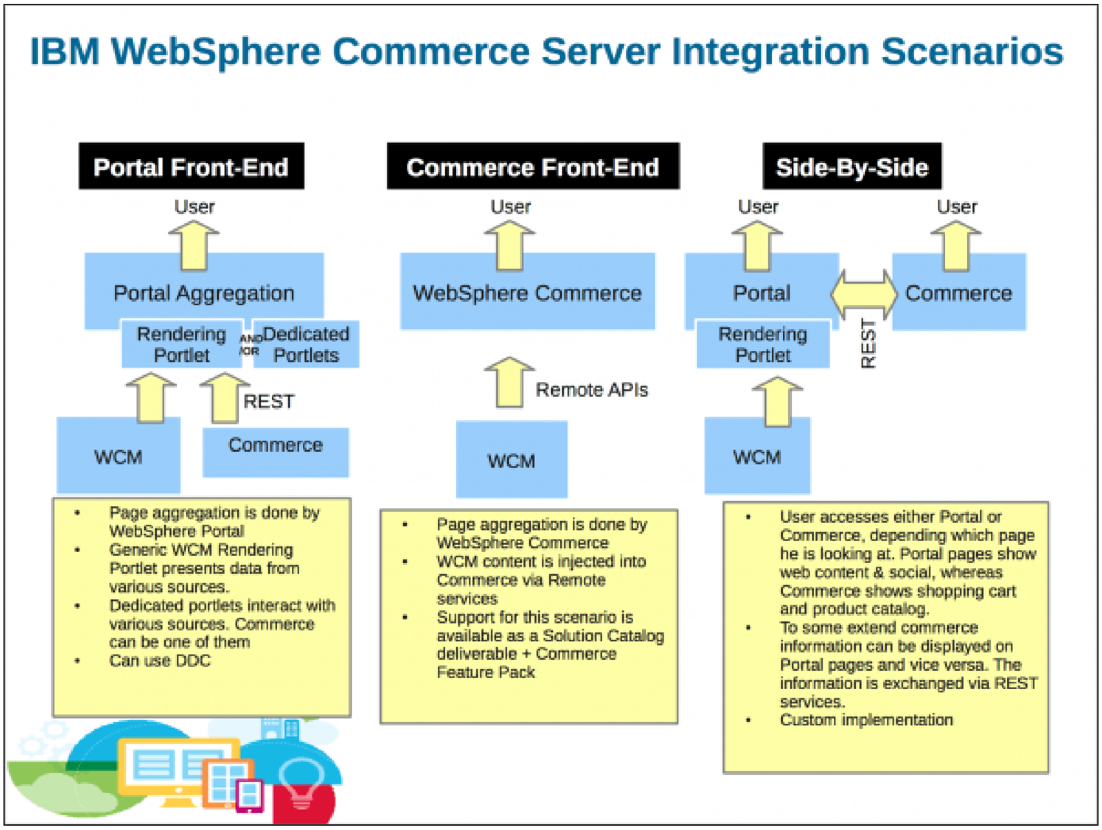
  Use WebSphere Portal to aggregate all your marketing and commerce content in the same environment. WebSphere Commerce provides example portlets that can deploy to your portal environment.

- Commerce as the front end

  Use the Web Content Manager API and REST services to serve content to the WebSphere Commerce environment. The WebSphere Commerce environment controls the full user experience.

- Side-by-side

  Users are directed to the appropriate environment depending on the requested page. Shared resources are requested between the environments through REST services.



IBM WebSphere Commerce Server Integration Scenarios

- Page aggregation is done by WebSphere Portal
- Generic WCM Rendering Portlet presents data from various sources.
- Dedicated portlets interact with various sources. Commerce can be one of them
- Can use DDC

- Page aggregation is done by WebSphere Commerce
- WCM content is injected into Commerce via Remote services
- Support for this scenario is available as a Solution Catalog deliverable + Commerce Feature Pack

- User accesses either Portal or Commerce, depending which page he is looking at. Portal pages show web content & social, whereas Commerce shows shopping cart and product catalog.
- To some extend commerce information can be displayed on Portal pages and vice versa. The information is exchanged via REST services.
- Custom implementation

Figure 1-7 WebSphere Commerce and IBM Digital Experience integration scenarios

1.3 IBM Digital Experience solution offerings and products

This section provides an overview of the solution offerings and products that are available from IBM that you can use to create and deliver digital experience capabilities to your customers and employees.

1.3.1 Solution offerings

The core solution offerings include the following ones:

- IBM Digital Experience Manager

  Digital Experience Manager helps organizations to deliver consistent, personalized, engaging, and interactive experiences across the range of channels and devices. Organizations can take advantage of flexible, responsive design capabilities that help ensure a consistent experience across web, mobile web, and hybrid mobile and web applications. It enables business users to create innovative digital experiences and reduce IT costs and reliance.

  For more information, see the Digital Experience Manager website:

  http://www.ibm.com/software/products/en/ibm-digital-experience-manager

- IBM Customer Experience Suite

  BM Customer Experience Suite helps organizations deliver to their customers a consistently engaging, personalized, and relevant digital experience across multiple channels and touch points. It provides rich, integrated capabilities for managing web content, real-time social communications, robust customer and commerce self-service capabilities, business analytics, and mobile device delivery.

  For more information, see the IBM Customer Experience Suite website:

  `http://www.ibm.com/software/products/en/ibmcustxpersuit`

- IBM Employee Experience Suite

  IBM Employee Experience Suite provides a broad and deep toolset for enabling socially infused web experiences for employees of medium and large enterprises. These solutions help drive innovation and collaboration by empowering employees with the ability to easily find and share relevant information across multiple platforms and diverse geographies and within the context of integrated business applications.

  For more information, see the IBM Employee Experience Suite website:

  `http://www.ibm.com/software/products/en/employee-experience-suite`

1.3.2 Products

Here are the key enabling products to help organizations deliver exceptional digital experiences:

- IBM WebSphere Portal

  IBM WebSphere Portal delivers digital content and applications with core portal services that aggregate applications and content as role-based applications for rich and personalized multichannel web experiences.

  WebSphere Portal supports workflows, content management, social services, and mobile web delivery. It is based on open industry standards and offers simplified usability and administration and cost-effective scalability. These capabilities provide the tools that are required to provide an exceptional digital experience to your audiences.

- IBM Web Content Manager

  IBM Web Content Manager delivers personalized web experiences to your customers based on their behaviors, roles, language, devices, location, and preferences. You can use IBM Web Content Manager to keep content fresh and relevant, and extend your reach through simple syndication across multiple channels and social media, with dynamic content, rating, tagging, commenting, and other interactive features. IBM Web Content Manager can help drive higher conversion rates, improve customer loyalty, and increase user engagement.

- IBM Forms

  IBM Forms is a unified, integrated package that enables LOB and IT users to collect data and automate processes through agile web applications and through classic document-based forms applications. The IBM Forms Experience Builder capability enables nontechnical users to create and distribute customized surveys, feedback forms, order forms, and more, without IT involvement.

  With IBM Forms, you can automate forms-based business processes for improved efficiency, more responsive customer service, and shortened time to value.

- IBM Web Experience Factory

  Web Experience Factory is a model-based development tool that creates applications that are rich, interactive digital experiences for delivery on desktop browsers, smartphones, tablets, and IBM MobileFirst. Web Experience Factory makes it easier to create, customize, deploy, and maintain multichannel portlets, widgets, and digital applications for desktop browsers, tablets, and smartphones. It is designed to enable developers to use easily their company core assets and automatically assemble them into custom, high-value portlets.

- IBM Connections

  IBM Connections is a social software platform that helps people engage with networks of experts in the context of critical business processes. IBM Connections facilitates simple, seamless access to your professional network, including colleagues, customers, and business partners. New capabilities include better ways to connect to additional networks of engaged, active participants, and to work more agilely and transparently to drive business value.

- IBM Sametime®

  IBM Sametime unified communications and collaboration software provides presence awareness, instant messaging, and online meetings capabilities for real-time collaboration within the Digital Experience. IBM Sametime makes it easier to locate expertise, chat, share files, and collaborate with colleagues and business partners through enterprise instant messaging.

- IBM MobileFirst Platform Foundation

  IBM MobileFirst is an open, comprehensive, and advanced mobile application platform to build, run, and manage mobile applications. It provides an open and comprehensive platform to not build, test, run, and manage native, hybrid, and mobile web applications. Available as an on-premises or private cloud solution, IBM MobileFirst Platform Foundation can help reduce both application development and maintenance costs, improve time-to-market, and enhance mobile application governance and security.

2

Architecture of an IBM Digital Experience environment

This chapter provides an overview of the Digital Experience architecture and infrastructure environment.

For more information about the Digital Experience product, see the Digital Experience website:

`http://www.ibm.com/software/products/en/category/digital-experience`

This chapter covers the following topics:

- 2.1, “Architecture of a Digital Experience environment” on page 20
- 2.2, “Deployment options” on page 23
- 2.3, “Integrating your IT” on page 29

2.1 Architecture of a Digital Experience environment

This section introduces the typical architecture of a Digital Experience environment. There are various factors that influence the deployment architecture of such an environment. Here are some of these factors and requirements:

- Everything is virtualized.
- Social portals with blogs, wikis, tags, ratings, and so on, are part of a website.
- Mobile access is expected.
- User interfaces are implemented by using a responsive design.
- Self-service features are offered through the social portal.
- The internet is the primary method of communication.
- The emphasis is on cross-team collaboration.
- Content and services are personalized.
- Users want to access what they want, when they need it.
- Portals and web content must be multilingual.
- An implementation of active site analytics is part of every website.
- The search function must work for the content that is delivered through the portal, and you must be able to find what you want with the fewest key words.
- Multi-tenancy portals serve different audiences on a share infrastructure.

To meet these requirements, the Digital Experience platform must be placed in the context of the supporting components that deliver the required engagement experience.

Figure 2-1 shows how the Digital Experience platform is positioned to access and integrate components that both attract users and service data from core business systems.

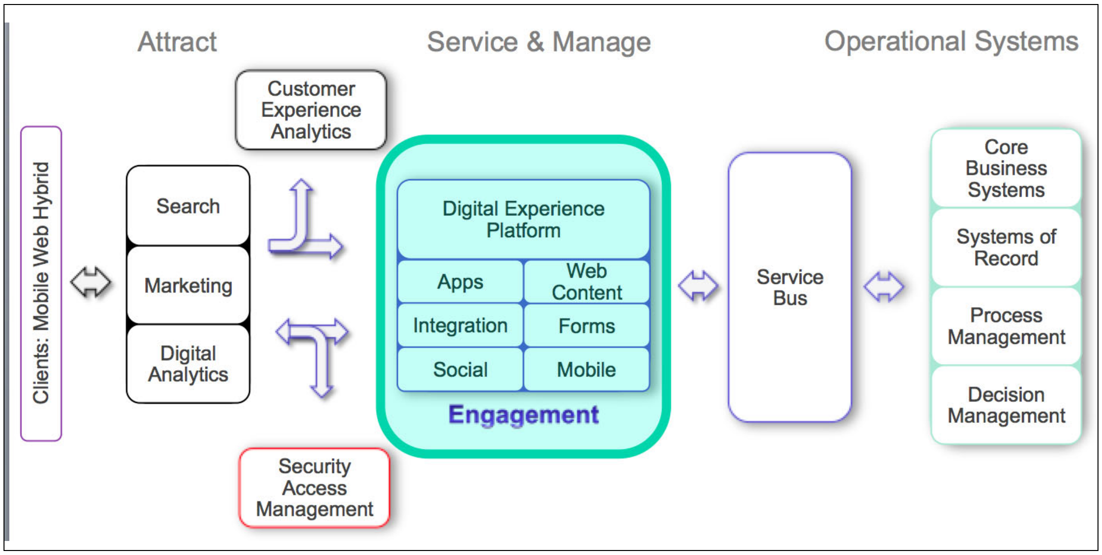


Figure 2-1 Digital Experience engagement architecture

The Digital Experience platform itself is often split into an authoring environment and a delivery environment, which means that content authors can work independently from the production platform and have full control over the content and information that is then published to the production environment.

Figure 2-2 shows how the components of the Digital Experience platform can be deployed and connected in these environments. You can run both environments in a cluster and share common components, such as an LDAP directory or the database server.

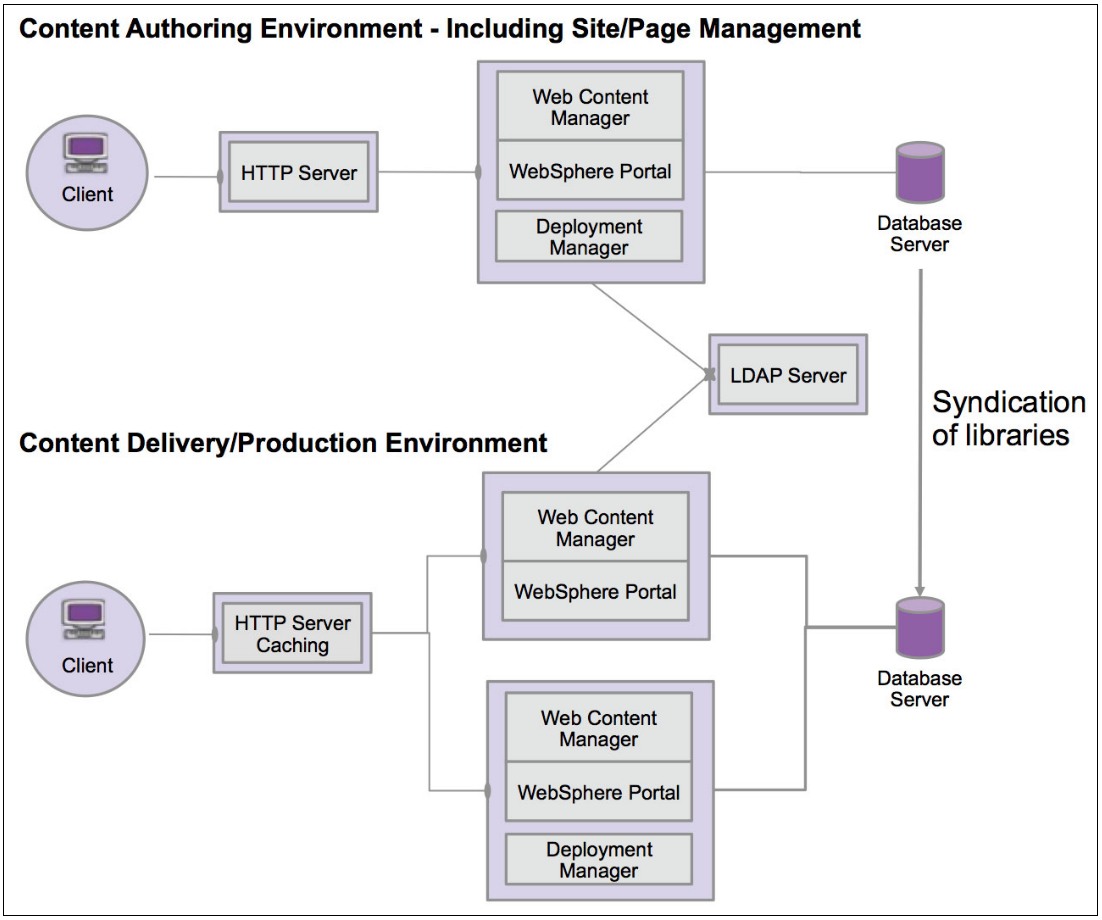


Figure 2-2 Digital Experience authoring and delivery environments

Typically each WebSphere Portal instance includes Web Content Manager, which is a core component of the Digital Experience platform. There are several reasons for the WebSphere Portal and Web Content Manager convergence:

- The Digital Experience platform requires features normally associated with traditional portals and with traditional Web Content Manager systems.
- IBM is working to integrate the technologies and processes around WebSphere Portal and Web Content Manager for a number of releases. Although they can still be licensed and used independently, they are tightly integrated technically. This integration is key to enabling the business to drive and manage the web experience lifecycle.
- IBM plans to continue to place more power into the hands of the website owners so they can do more independently as needed.

- The IT team can use templates, and the corresponding automation framework, to build the key parts to enable the business to quickly act and react.
- Projects and approvals provide appropriate levels of checks and balances.

From a functional perspective, the WebSphere Portal and Web Content Manager components are placed in a multitier environment between the DMZ and the intranet, which hosts the core business systems.

Figure 2-3 shows a typical functional placement of the Digital Experience components in a four-tier architecture in the context of the supporting business systems to be integrated by a website.

However, with the newer and more flexible cloud deployment options, the placement of the Digital Experience components can shift, allowing for hybrid deployments as well. These options are introduced in 2.2, “Deployment options” on page 23.

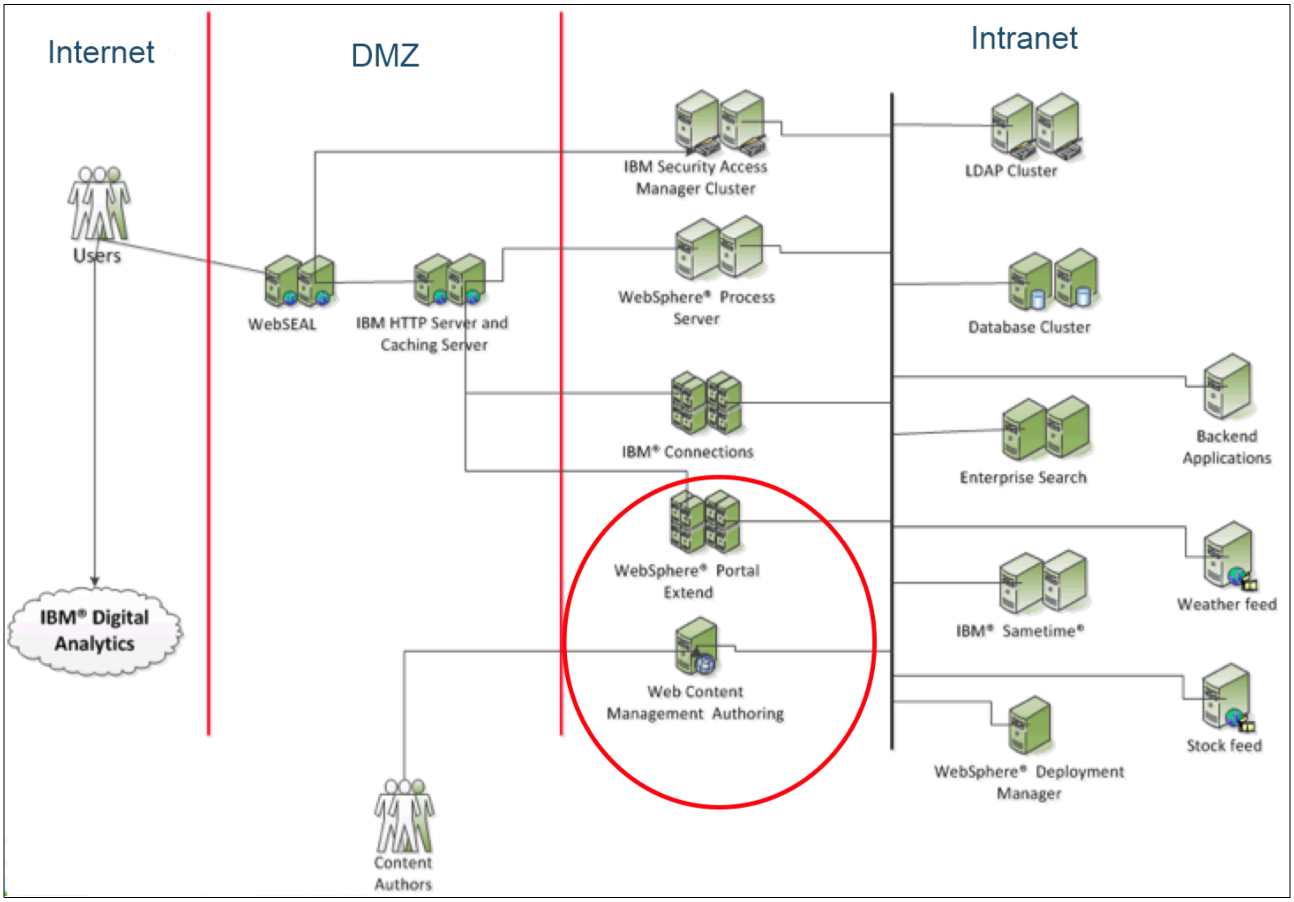


Figure 2-3 Typical functional placement of Digital Experience components

2.2 Deployment options

IBM WebSphere Portal provides flexible deployment options that range from proof-of-concept (PoC), where you can examine and test functions, to a highly available and scalable production environment.

For more information about the products that are part of the Digital Experience portfolio, see 1.3, “IBM Digital Experience solution offerings and products” on page 15. You can also see to the Digital Experience website:

http://www.ibm.com/software/products/en/category/digital-experience

In general, you can choose between the following three different deployment options that are available for the Digital Experience product:

- 2.2.1, “IBM Digital Experience on Cloud” on page 23
- 2.2.2, “Digital Experience Pattern on IBM PureApplication System” on page 26
- 2.2.3, “Digital Experience on-premises” on page 28

Table 2-1 provides a comparison of these deployment options.

Table 2-1 Comparison of Digital Experience deployment options

Digital Experience on Cloud	On-Premise Hosting	Infrastructure as a Service
► Check that the Digital Experience Cloud product option fits your sizing requirements. ► Good if you want to get started quickly, without infrastructure delays. ► Back-ends are either already on cloud or staying on premises.	► You require components that are not available inside the product. ► Custom applications or databases. ► Products that are not part of the Digital Experience on Cloud option. ► Hosted in a location that the Digital Experience on Cloud option does not cover. ► You still want a managed service.	► You want to manage the application yourself. ► You require components that are not available inside the product. ► Require complete control over the cloud infrastructure. ► Your focus is development or test environments.
Consider Digital Experience on Cloud. For details, see 2.2.1, “IBM Digital Experience on Cloud” on page 23.	Consider engaging Lab Services from IBM or an IBM Business Partner. For details, see 2.2.3, “Digital Experience on-premises” on page 28.	Consider PureApp Service on SoftLayer. For details, see 2.2.2, “Digital Experience Pattern on IBM PureApplication System” on page 26.

2.2.1 IBM Digital Experience on Cloud

Digital Experience on Cloud is a fully managed, off-premise digital experience platform. You can use it to build compelling, socially aware, omnichannel digital experiences for your customers, business partners, and employees.

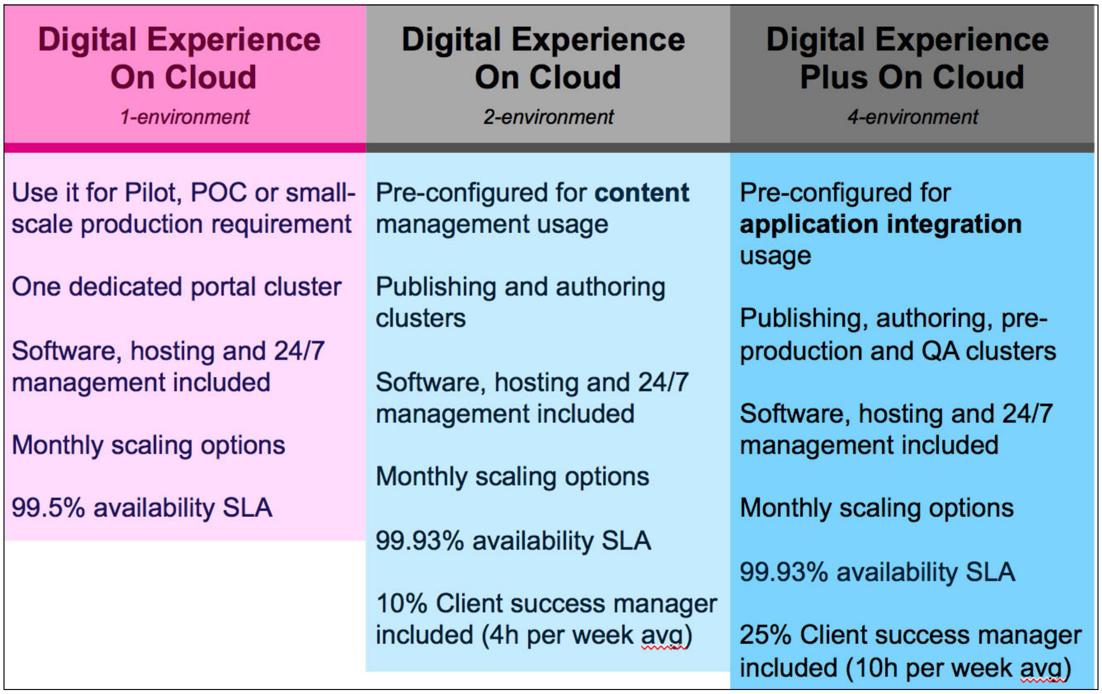
For more information, see the following Digital Experience on Cloud websites:

- http://www.ibm.com/software/collaboration/digitalexperience/cloud.html
- http://www.ibm.com/software/products/en/digital-experience-cloud

There are two offerings that are available for the Digital Experience for Cloud product:

- The *Digital Experience on Cloud* offering enables rapid setup and onboarding that is targeted at customers that are focused on site and content management with standard integration needs.
- The *Digital Experience Plus on Cloud* offering offers a robust solution to enterprise customers looking to offer a secure, advanced experience to their users through integration with Java applications, mobile apps, digital analytics, and so on.

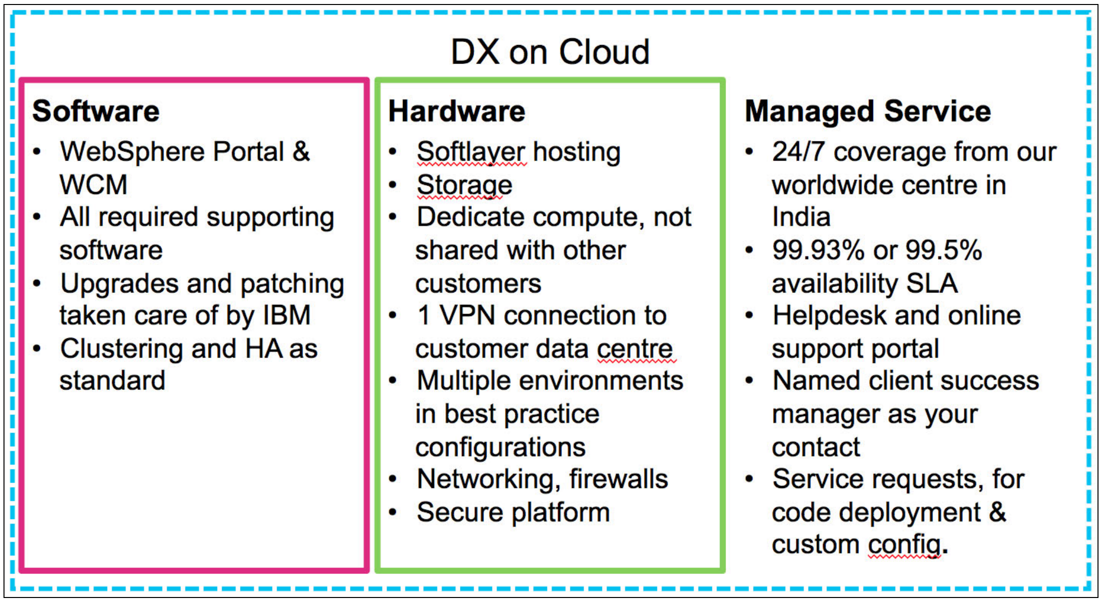
Figure 2-4 provides details about the Digital Experience for Cloud offerings based on different environments.



Digital Experience On Cloud *1-environment*	**Digital Experience On Cloud** *2-environment*	**Digital Experience Plus On Cloud** *4-environment*
Use it for Pilot, POC or small-scale production requirement	Pre-configured for **content** management usage	Pre-configured for **application integration** usage
One dedicated portal cluster	Publishing and authoring clusters	Publishing, authoring, pre-production and QA clusters
Software, hosting and 24/7 management included	Software, hosting and 24/7 management included	Software, hosting and 24/7 management included
Monthly scaling options	Monthly scaling options	Monthly scaling options
99.5% availability SLA	99.93% availability SLA	99.93% availability SLA
	10% Client success manager included (4h per week avg)	25% Client success manager included (10h per week avg)

Figure 2-4 Digital Experience on Cloud product offerings

The Digital Experience on Cloud product offerings are composed of the necessary software and hardware components and the managed service to operate a full Digital Experience environment. Figure 2-5 provides a list of the included components and services.



DX on Cloud

Software

- WebSphere Portal & WCM
- All required supporting software
- Upgrades and patching taken care of by IBM
- Clustering and HA as standard

Hardware

- Softlayer hosting
- Storage
- Dedicate compute, not shared with other customers
- 1 VPN connection to customer data centre
- Multiple environments in best practice configurations
- Networking, firewalls
- Secure platform

Managed Service

- 24/7 coverage from our worldwide centre in India
- 99.93% or 99.5% availability SLA
- Helpdesk and online support portal
- Named client success manager as your contact
- Service requests, for code deployment & custom config.

Figure 2-5 Digital Experience on Cloud components

The Digital Experience on Cloud offerings include dedicated topologies that host up to three environments for production, pre-production, and quality assurance, as shown in Figure 2-6.

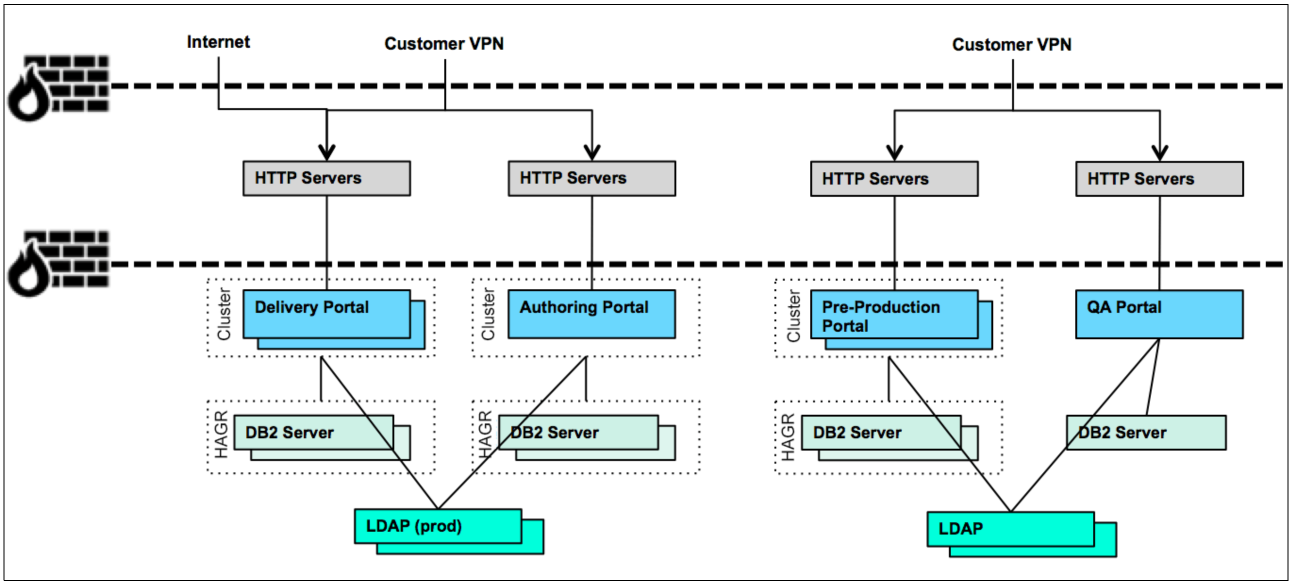


Figure 2-6 Digital Experience on Cloud topology

2.2.2 Digital Experience Pattern on IBM PureApplication System

IBM PureSystems® is a product family of expertly integrated systems from IBM that combine software and hardware to give you a pre-configured, efficient virtualization platform. The offering from PureSystems that works with the WebSphere Portal Hypervisor Edition is the IBM PureApplication System (PureApp), and includes IBM middleware images and patterns to customize and deploy, relying on the pattern engine to handle image and instance management.

You can use the Digital Experience Pattern to experience the fully functional WebSphere Portal Server and Web Content Manager instance that is deployed through PureApp. The contained image also includes definitions and configurations for multiple parts of a typical WebSphere Portal and Web Content Manager deployment, so you can build simple to more complex topologies quickly and easily.

Digital Experience Patterns V2.0 for WebSphere Portal Server V8.5 and Digital Experience Patterns V2.0 for Web Content Manager V8.5 include IBM WebSphere Portal Server and IBM Web Content Manager preconfigured patterns and script packages based on expert practices, including single-server and multiple-server clustered deployments, database configuration, directory federation, and more, which can automate the setup of simple to complex Digital Experience deployments on IBM PureApplication System V2.0.

PureApp Patterns automate the following capabilities in a deployment:

- Application topology
- Pre-integrated across components
- Pre-configured and tuned
- Pre-configured monitoring and security
- Preinstalled on an operating system
- Lifecycle management

The PureApp Pattern installs WebSphere Portal and Web Content Manager, accepts many distinct inputs directly, and then scripts the entire deployment of the full product package across several nodes, as shown in Figure 2-7.

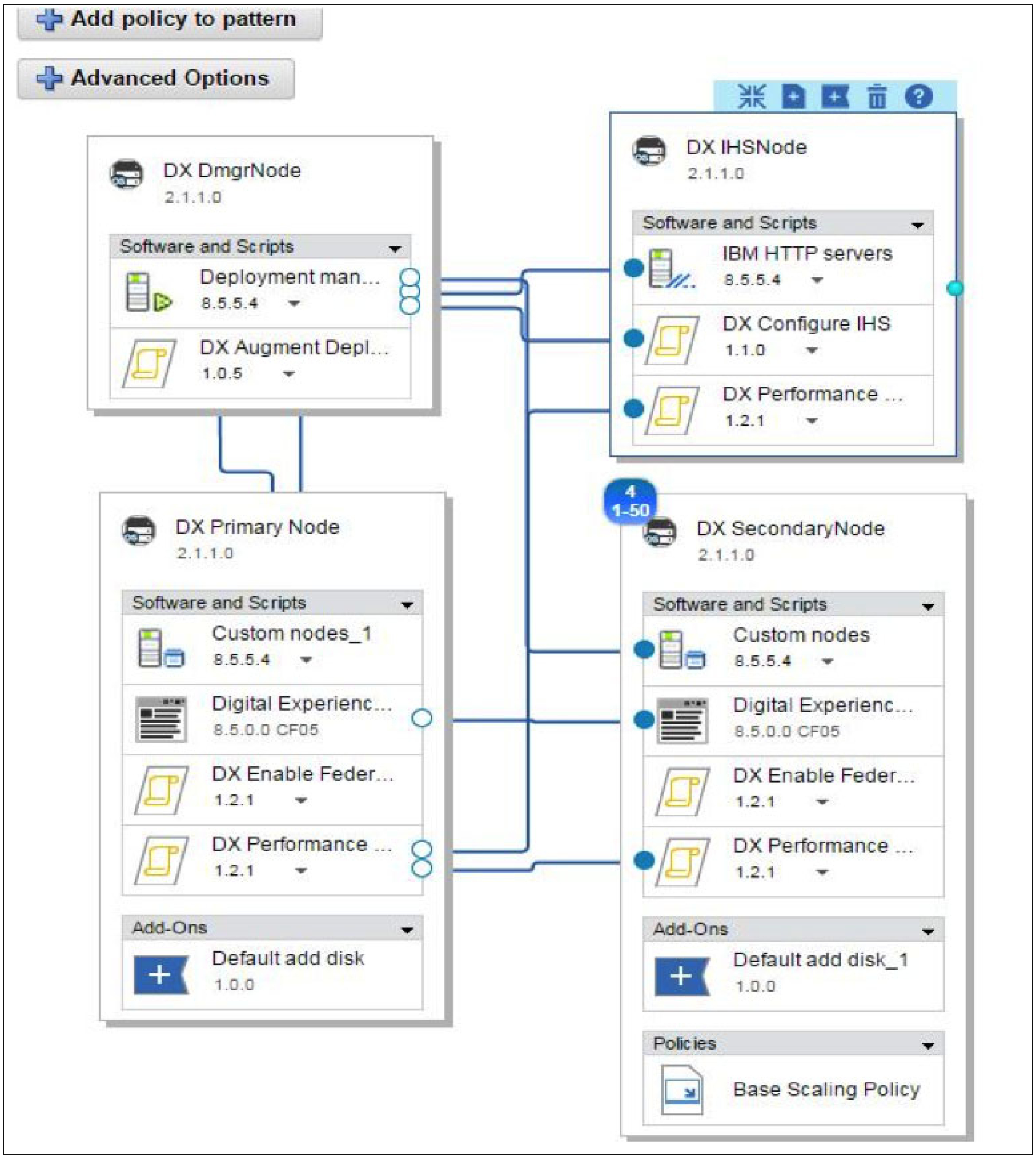


Figure 2-7 PureApp Pattern for Deployment on several nodes

The current pattern also does further configurations:

- Cleans up the virtual machine (VM) after deployment.
- Updates the Portal databases.
- Adds vertical cluster members.
- Adds and configured IBM HTTP Server instances.
- Enabled federated security.
- Performance tuning.

PureApp V2.0 separates the operating system from the middleware. You can use it to combine several software components in a single VM and handles the dependencies between the components.

2.2.3 Digital Experience on-premises

A sample site can also be built on Digital Experience platform on-premises. However, unlike the Digital Experience on Cloud offering, customers must plan to install, configure, and maintain the environment.

The production environment represents the servers that incoming web traffic accesses to experience your digital experience. Before you install IBM WebSphere Portal in a production environment, you must assess your hardware and software needs, possible database configurations, security options, and LDAP server options. This environment can be composed of multiple clusters or a stand-alone server, depending on your business needs.

The general recommendation is to host on a cluster server to ensure failover, balance, and distribute the traffic. In a cluster, the portals share a common configuration and the load is distributed evenly across all cluster instances. Figure 2-8 shows a sample on-premises topology.

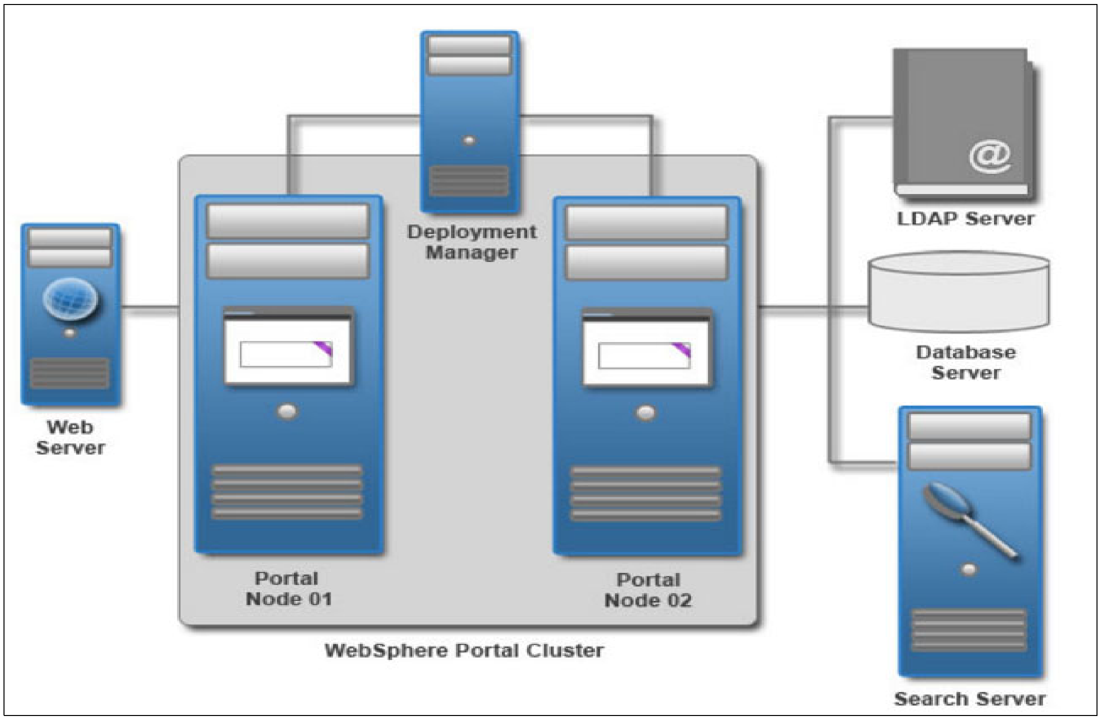


Figure 2-8 WebSphere Portal on-premises cluster topology

For more information about system requirements and installation instructions for WebSphere Portal, see the *Planning to install WebSphere Portal* topic in the IBM WebSphere Portal 8.5.0 IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/plan/plan_installation.dita

You can get the software as a physical media on a DVD. Installing from physical media is practical when installing on a limited number of servers. Another option is to get the electronic images. Customers can use their IBM Passport Advantage® account for downloading the electronic images of WebSphere Portal.

The installation process is simple. You can use IBM Installation Manager or get the software from a live repository. To download and install WebSphere Portal directly from a live repository, you need IBM Installation Manager, and you select the portal repository from the list.

For more information about how to set up a WebSphere Portal cluster environment, see the *Step-by-Step Cluster Guide for IBM WebSphere Portal V8.5* guide, found at:

http://www.lotus.com/ldd/portalwiki.nsf/dx/Step-by-Step_Cluster_Guide_for_IBM_WebSphere_Portal_v8.5

2.3 Integrating your IT

To integrate your existing IT systems, The Digital Experience product offers various client- and server-side technologies to match your technical, skill, and business requirements for your use case.

Figure 2-9 shows a general four-tier delivery architecture and a general way that the client- and server-side integration technologies are used.

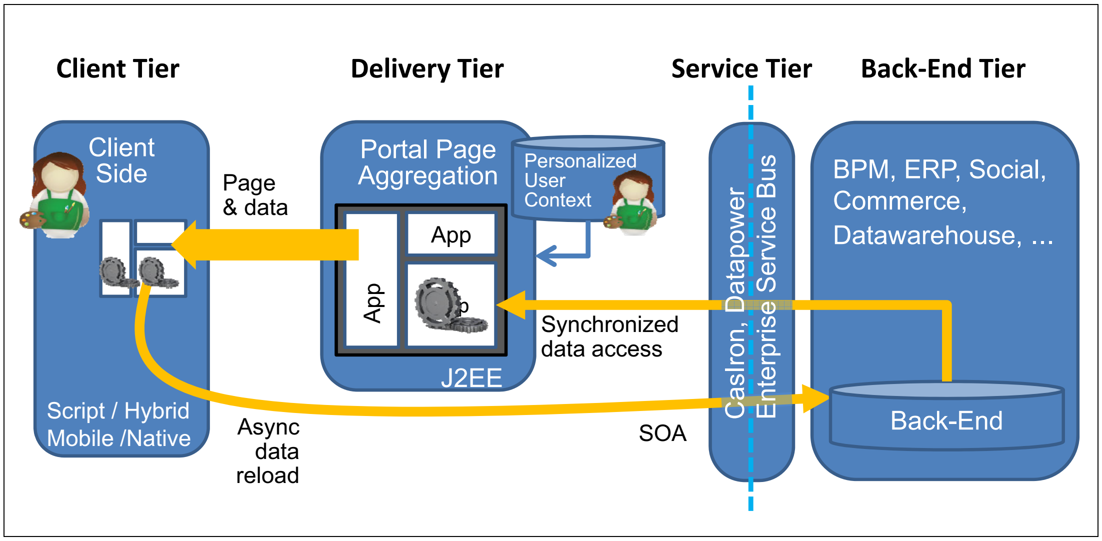


Figure 2-9 Four-tier delivery architecture

The following tiers are shown in Figure 2-9:

- Client tier

  The client tier can render scripted or mobile websites in a browser, hybrid, or native applications.

- Delivery tier

  WebSphere Portal aggregates and personalizes the page content from various sources and delivers it to the client tier.

- Service tier

  An optional service tier can be implemented to interface the back-end tier as managed services to the delivery tier. There are several products available from IBM to deliver this capability, such as Castiron, DataPower, or IBM ESB.

- Back-End tier

  The back-end tier hosts the business applications from various vendors and data to be placed in the Digital Experience product.

Table 2-2 provides a comparison of the client- and server-side integration technologies.

Table 2-2 Comparison of client- and server-side integration technologies

Client-side integration technologies	**Server-side integration technologies**
Allows you to run client-side libraries (AngularJS, Dojo, Jquery, and so on) or implement as a native app (iOS/Android).	Allows you to run server-side code (Java, JSP, and Java Platform, Enterprise Edition).
Ajax calls or updates can be achieved without page refresh.	Aggregate a personalized page for the current user context and role.
Back-end data or a SOA layer can be directly accessed by using REST calls.	Back-end data or a SOA layer can be accessed from server-side code.

For developing custom applications and views that run on WebSphere Portal, there are a number of tools that are available. With the recent additions of IBM Script Portlet and IBM Digital Data Connector (DDC) to the tools portfolio, you now have the following options for tools:

- IBM Web Application Bridge: Integrates external (non-portal) web applications into the Digital Experience product, configures single sign-on to those remote applications, and changes the markup that they return.

  For more information about Web Application Bridge, see 5.3, “Integrating existing web applications with IBM Web Application Bridge” on page 177.

- IBM Script Portlet: Builds JavaScript-based portlets by using industry-standard skills and widely available libraries, in the cloud or on-premises with no code deployment.

  For more information about IBM Script Portlet, see 5.4, “Using the IBM Script portlet” on page 192.

- Digital Data Connector (DDC): Uses IBM Web Content Manager components to render any external data, with little to no Java coding or deployment.

  For more information about DDC, see 5.2, “Digital Data Connector” on page 134.

- IBM Forms Experience Builder: Builds complete data entry form solutions from the web browser without any coding.

  For more information about Forms Experience Builder, see 5.5, “Integrating Forms Experience Builder” on page 199.

- IBM Web Experience Factory: Uses model-based code generation tools to build portlets and generate multiple variations automatically.

- Rational Application Developer: Develop Java based portlets in an Eclipse-based integrated developer environment (IDE) with a comprehensive tool set featuring wizards and drag.

Figure 2-10 shows an overview of the more business-driven integration options, except for Web Experience Factory and Rational Application Developer. These tools provide a more sophisticated, business-use-managed means to integrate either web, or service-enabled back-end tier applications.

> **Note:** In Figure 2-10 and Figure 2-11 on page 32, the following acronyms are used because of space limitations. These acronyms are *not* approved by IBM:
>
> - Web Application Bridge (WAB)
> - Web Services for Remote Portlets (WSRP)
> - Forms Experience Builder (FEB)
> - Unified Task List (UTL)

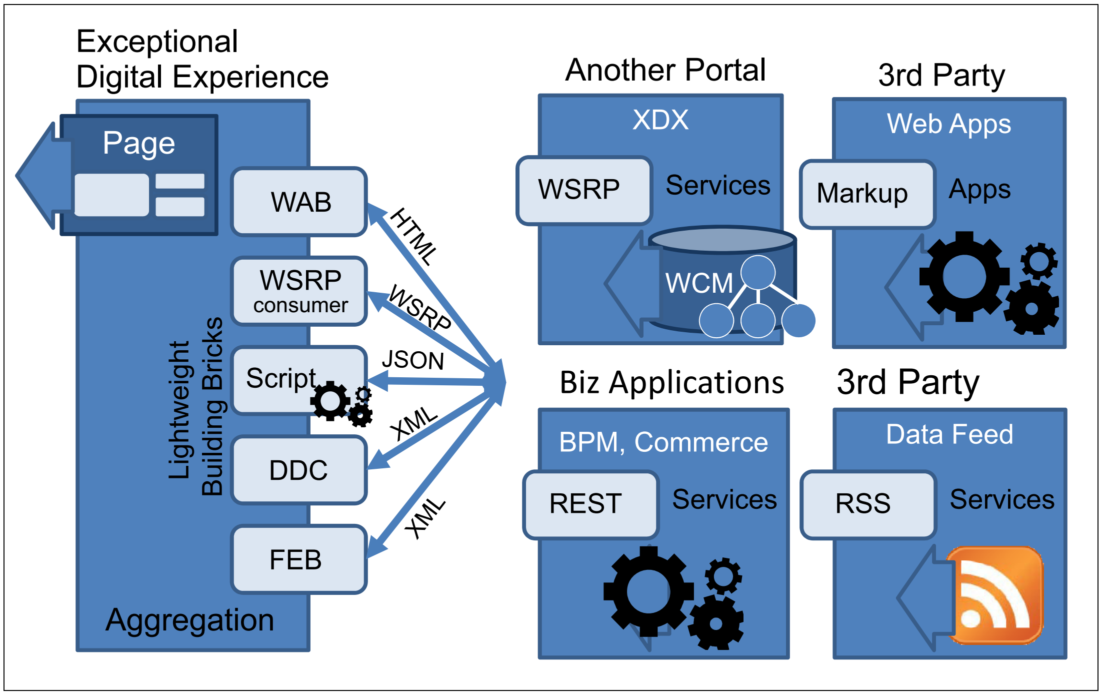


Figure 2-10 Business driven lightweight integration options

This book covers several integration options to distinct products and back-end systems and focuses on the more business-driven lightweight integration options that WebSphere Portal offers. These options offer the benefit of not having to code an integration by using Java Platform, Enterprise Edition skills, but rather use a configurative approach consuming services and markup from the systems and rendering them into a portal.

The customization of these integration options is done with standard web design technologies such as HTML, JavaScript, and CSS instead of building Java Platform, Enterprise Edition components with a local IDE. This approach opens new opportunities for business users to configure and manage these integrations and for employees and business partners with available standard web design skill to extend and customize them.

Figure 2-11 shows how these products are integrated and what technology component in WebSphere Portal is used for this integration.

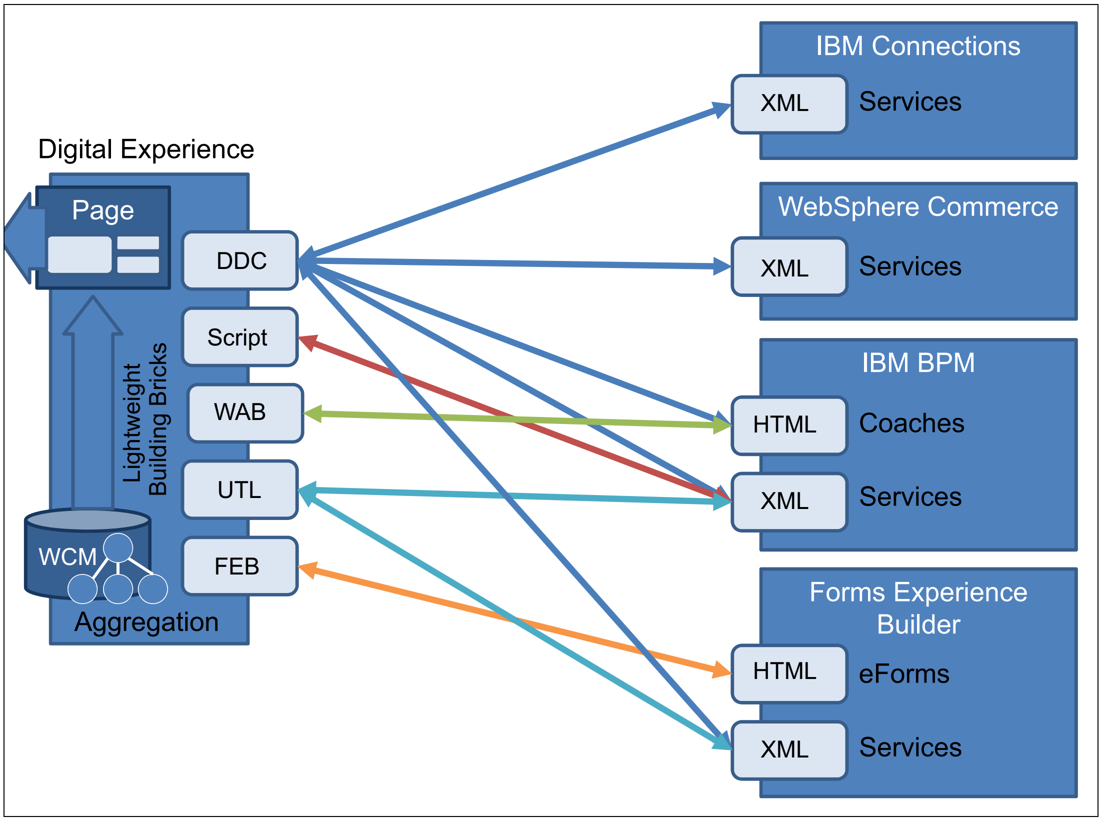


Figure 2-11 Integration options and technologies used

Chapter 5, "Lightweight development and integration options" on page 131 introduces each of the following tools and provides guidance about selecting the most appropriate tool for your requirements and organization. It begins with a guide to choosing the correct tool, and then provides an overview of each of the different tools, highlighting the features, benefits, and typical use cases of each tool.

- IBM Connections is a premiere Social Business platform that can be effectively integrated by using the DDC. The solution that is used is called *social rendering*. For more information, see 5.2, "Digital Data Connector" on page 134.
- WebSphere Commerce is a platform for building sophisticated web shopping solutions and can also be integrated by using DDC. IBM provides ready-designed components or templates.
- IBM Business Process Manager (IBM BPM) is an enterprise process management platform that you can use to design and run flexibly many business processes. Accessing the REST API and the coach user interfaces, the Digital Experience product offers options such as DDC, Unified Task List, and Web Application Bridge to integrate IBM BPM in different ways.
- IBM Forms Experience Builder is a solution that non-technical users can use to build sophisticated web applications for form-based business processes. Forms Experience Builder can be integrated by using the Forms Experience Builder portlet to visualize the e-forms and by using the DDC to access the submitted data for rendering a tasklist for the business process. For more information, see 5.5, "Integrating Forms Experience Builder" on page 199 and 5.2.6, "Use case 3: Integrating with IBM Forms Experience Builder" on page 163.

3

Getting started with IBM WebSphere Portal themes

WebSphere Portal provides a framework that you can use to render an entire website. WebSphere Portal *themes* control the presentation layer of your website. The WebSphere Portal theme framework provides a foundation for the website user interface. It includes components to enable website navigation, rendering on different devices, search, and so on, and it makes user interaction easier. The theme also enables visual consistency and controls page layout and portlet-rendered markup.

This chapter introduces WebSphere Portal themes, the theme elements, and the basic theme concepts that you need as a prerequisite to building a website with WebSphere Portal.

This chapter covers the following topics:

- 3.1, “Introduction to themes, skins, and layouts” on page 34
- 3.2, “Creating a theme with the Theme Manager” on page 40
- 3.3, “Theme optimization” on page 46
- 3.4, “Managing themes by using IBM Digital Experience File Sync” on page 48
- 3.5, “Customizing the theme” on page 48
- 3.6, “Support for mobile devices” on page 54
- 3.7, “Web Developer Toolkit for IBM Digital Experience” on page 56

3.1 Introduction to themes, skins, and layouts

To some extent, the theme governs aspects of the entire website. It renders several elements of the page, such as the header and banner, navigation, search bar, page layout, and footer, as shown in Figure 3-1.

Themes define a consistent navigation interaction model by specifying the navigation hierarchy and structure. They also define a top-level stylesheet to specify colors and fonts.

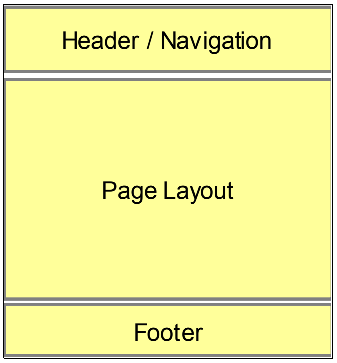


Figure 3-1 Basic page elements that are rendered by a theme

A theme is assigned to a portal page or page group. You can define multiple, similar themes so that different places in your portal can have some differences.

Skins are part of the theme framework. A skin defines the appearance of the area surrounding a portlet. It includes a header area, which contains the portlet menu and possibly the content menu, and a body area, which contains the body of the portlet.

Each portlet on a page can be assigned a skin. The skin defines the colors and frame around the portlet, and the presentation of common portlet actions, such as maximize, minimize, edit, configure, and help.

Skins affect portlets or a piece of content on a page. In particular, a skin defines the portlet title bar and border. Each portlet on a page can have a different skin that is associated with it. Skins can also be used to affect advanced navigation, such as a tree.

Skins contain images that create the visual effects. These images are the main component of a skin. These effects can be composed of lines, shadows, and backgrounds. The icon images, such as help, edit, minimize, and maximize, appear in the portlet title bar.

The *layout* of the page renders the columns on the portal page.

Figure 3-2 shows the main artifacts that are described in this section.

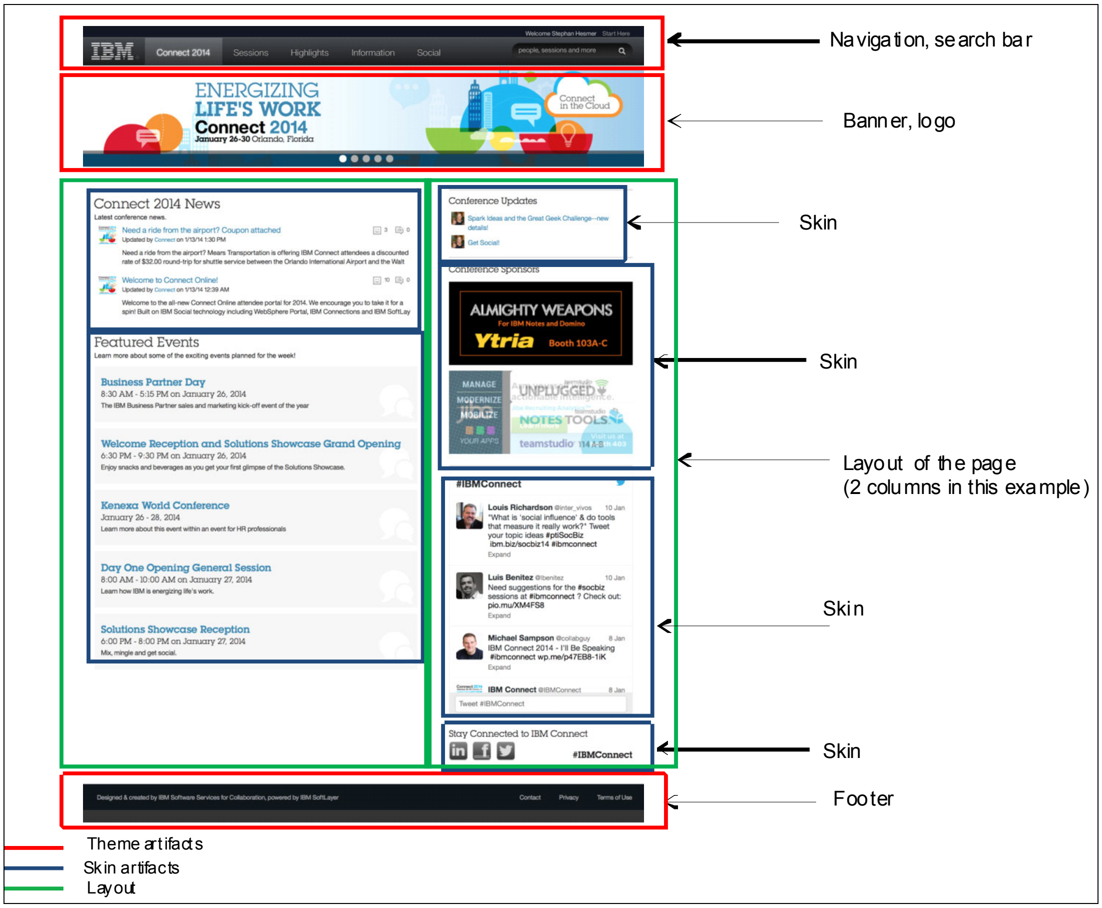


Figure 3-2 Themes and skins artifacts and page layout

The following topics are covered in this section:

- 3.1.1, "WebSphere Portal V8.5 themes" on page 35
- 3.1.2, "Theme resources" on page 36
- 3.1.3, "Understanding the portal page rendering flow" on page 39

3.1.1 WebSphere Portal V8.5 themes

WebSphere Portal V8.5 was initially shipped with a single default theme called *Portal8.5*. In addition, WebSphere Portal V8.5 CF08 now includes a theme template called *Simple*. However, the Simple theme is only a template, which means it cannot be applied to a page or customized like the default theme; rather, you always must copy it in the Theme manager before you do anything. For more information about the Theme manager, see 3.2, "Creating a theme with the Theme Manager" on page 40.

Portal8.5 theme

The Portal8.5 theme is the default modularized and optimized theme template that is used by WebSphere Portal. It incorporates both adaptive and responsive design elements. You can copy this theme to create a custom theme.

The WebSphere Portal V8.5 modularized theme provides a flexible framework that provides the following capabilities:

- Minimizes the download size by giving you the control to specify just the capabilities that are needed for a certain scenario as a theme default or on a per page basis.
- Minimizes the number of requests by combining necessary resources.

The Portal8.5 theme is adaptive to the rendering device and optimizes content to ensure an enhanced user experience across mobile channels.

The Portal8.5 theme contains two types of files:

- Static resources (for example, JavaScript)
- Dynamic content (for example, JSP files)

Simple theme

You can use the Simple theme template to assemble quickly a theme with just a few operations. The template contains fewer files than the Portal 8.5 theme. To create a custom theme from a Simple theme template, modify the HTML, JavaScript, and JSON resources. These changes are done by using WebDAV.

Important: The Simple theme is only a template, which means it cannot be applied to a page or customized like the default theme; you always must copy it into the Theme manager before you do anything. For more information about the Theme manager, see 3.2, “Creating a theme with the Theme Manager” on page 40.

Here are some of the features of the Simple theme template:

- The Simple theme is responsive and adapts to the container window.
- The Simple theme includes four options for navigation. Each option includes a dynamic content spot. You can use the navigation options that are provided by changing the dynamic content spot. For more information about dynamic content spots, see 3.5.5, “Dynamic content spots” on page 51.
- By default, the files in the Simple theme template are English only. However, you can add other locales to the custom version of the theme.
- Images use Scalable Vector Graphics (SVG), which are high definition and scale better on mobile devices.
- The Simple theme has fewer cascading stylesheet (CSS) files than the Portal8.5 theme.

3.1.2 Theme resources

The theme framework includes static and dynamic resources.

Static resources

Static resources are a collection of templates for page rendering, HTML, and translated HTML files, images, and stylesheets.

Static resources include the following items:

- JavaScript, CSS, and markup
- Templates for page rendering

  For example, the theme template file, `theme.html`, contains the main HTML markup that the portal server serves up for your theme's pages. It renders and gives the position of all the elements on the outermost level.

  There are a few levels inside, one being the page layout, which is determined by the layout template, and the innermost border is the skin. The markup for the skin can be found in the `skin.html` file.

Static resources are stored in the WebDAV file store on the WebSphere Portal Server. Anything that is stored on WebDAV is static content, which does not need to be compiled and redeployed onto a server. Many issues within the theme can be easily resolved by looking at the Theme Analyzer for guidance.

Static files can be accessed with a WebDAV client or by using the IBM Digital Experience File Sync (DXSync) (for more information, see 3.4, “Managing themes by using IBM Digital Experience File Sync” on page 48).

Location of static resources

Here is the main WebDAV entry point:

```
http://<host>:<port>/wps/mycontenthandler/dav/fs-type1/
```

The following list describes the location of various static resources for the Portal8.5 and Simple themes. Resources for themes that are created by copying the Portal8.5 or Simple themes are in the corresponding path under the appropriate theme name.

- Theme directory:

  `fs-type1:\themes\<theme_name>\`
- Theme templates, and the location of the `theme.html` file:
  - Portal8.5 theme:

    `fs-type1:\themes\Portal8.5\nls\`
  - Simple theme:

    `fs-type1:\themes\Simple`
- Skin templates:
  - Portal8.5 theme:
    - Location of the hidden skin:

      `fs-type1:\themes\Poratl8.5\skins\Hidden\nls\`
    - Location of the standard skin:

      `fs-type1:\themes\Portal8.5\skins\Standard\nls\`
    - Location of the NoSkin skin:

      `fs-type1:\themes\Portal8.5\skins\NoSkin\`
  - Simple theme - location of the standard skin:

    `fs-type1:\themes\Simple\skins\Standard\`

- Layout templates:
  - Portal8.5 theme:

    `fs-type1:\themes\Portal8.5\layout-templates\`
  - Simple theme:

    `fs-type1:\themes\Simple\layout-templates\`
- Menu definitions:
  - Portal8.5 theme:

    `fs-type1:\themes\Portal8.5\menuDefinitions\`
  - Simple theme:

    `fs-type1:\themes\Simple\menuDefinitions\`

Location of script resources

The script resources are in the following directories. The JavaScript resources are available as part of Portal V8.5, and can be used in a theme. However, they cannot be modified by the users.

- Dojo module:

  `portalserver_root\theme\wp.theme.dojo\installedApps\dojo.ear\dojo.war\`
- JQuery module:

  ```
  portalserver_root\theme\wp.theme.jquery\installedApps\wp.theme.jquery.ear\wp.th
  eme.jquery.war
  ```

Dynamic resources

Dynamic resources in the portal page are rendered through *dynamic content* s*pots*. Dynamic content spots reference JSP files or other dynamic resources. Within the theme template, there are various dynamic content spots that the WebSphere Portal server processes, typically through JSPs, and replaces with dynamically calculated static markup for the rendered page.

Location of dynamic resources

Dynamic theme resources are stored in a Web Application Archive (WAR) file and are in the following directory:

```
portalserver_root\theme\wp.theme.themes\default85\installedApps\DefaultTheme85.ear
\DefaultTheme85.war\themes\html\dynamicSpots
```

The entry point for a theme is either `Default.jsp` or `Plain.jsp`. Both files are in the following directory:

```
portalserver_root\theme\wp.theme.themes\default85\installedApps\DefaultTheme85.ear
\DefaultTheme85.war\themes\html
```

The dynamic content spots that are referenced by the theme template are in the following directory:

```
portalserver_root\theme\wp.theme.modules\webapp\installedApps\ThemeModules.ear\The
meModules.war\themes\html\dynamicSpots
```

3.1.3 Understanding the portal page rendering flow

This section describes the flow that renders a portal page. Figure 3-3 shows the sequential flow of a user request through the WebSphere Portal Server and the components that are involved to compose the resulting page that is sent back to the user.

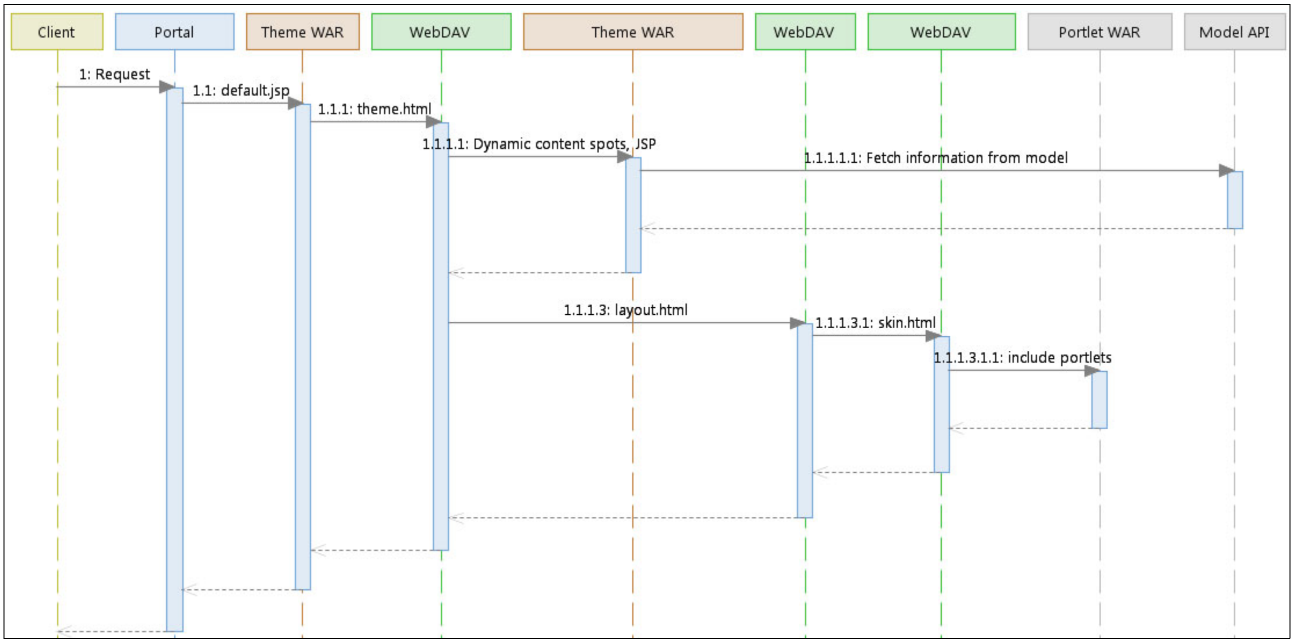


Figure 3-3 Sequential flow of a user request through the WebSphere Portal Server

A user initiates a request from a browser. The request arrives at the WebSphere Portal Server, where the aggregation starts with the `default.jsp`, which includes other dynamic JSPs that are used to display page elements.

The subsequent flow consists of two logical parts that start static and dynamic theme resources. The static resources include CSS stylesheets and JavaScript files. The JSP files are the dynamic resources. The static resources are stored in WebDAV and the dynamic resources are deployed and stored in the theme WAR file. The static skin template can be used to render the portlet content on the page. The dynamic resources can call back-end systems, such as a database, web services, or rules engines, which is shown in Figure 3-3 as Model API.

Here is the sequence that is shown in Figure 3-3:

- 1: The user initiates a request from a browser. The request arrives at the WebSphere Portal Server. The WebSphere Portal Server sends the request to the theme application (Theme WAR), where the aggregation of the theme elements starts to build the theme.
- 1.1: The starting point is the `default.jsp` file. This file delegates aggregation to the theme template.

- 1.1.1: The request goes to the WebDAV repository, which contains static components of a theme, such as HTML files, CSS stylesheets, and so on. The `theme.html` file is aggregated. Two subflows are then generated:
  - 1.1.1.1: `Theme.html` includes various dynamic content spots references. This flow retrieves the dynamic content, for example, navigation, an address, phone number, and any data that is likely to change.
  - 1.1.1.1.1: This flow retrieves data from a back-end system programmatically from a JSP.
  - 1.1.1.3: This flow aggregates and renders the static components, such as the theme layout (`layout.html`) and skin (`skin.html`, 1.1.1.3.1).
  - 1.1.1.3.1.1: This flow retrieves the portlets that are configured on the page.

3.2 Creating a theme with the Theme Manager

This section describes the use of the Theme Manager portlet to create themes from templates and edit existing themes. The Theme Manager portlet can be used to copy, edit, export, delete, and explore the themes on your portal.

Note: The Theme Manager was introduced in WebSphere Portal V8.5 CF08.

To access the Theme Manager portlet, complete the following steps:

1. From the WebSphere Portal administration console, click **Applications menu** → **Theme Development**, as shown in Figure 3-4.

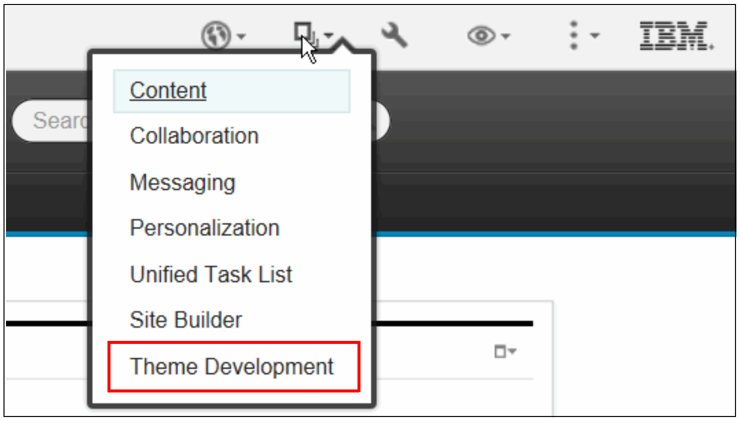
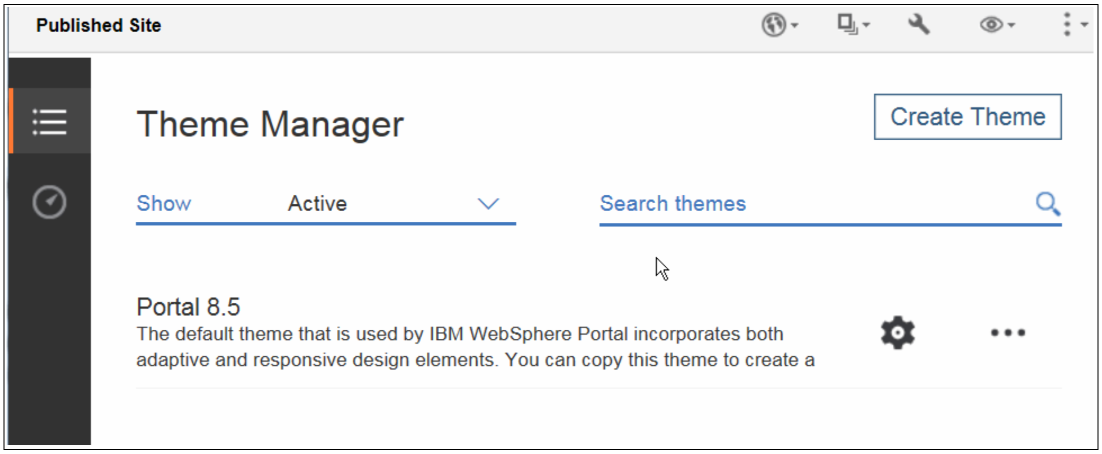


Figure 3-4 Access the Theme Manager portlet

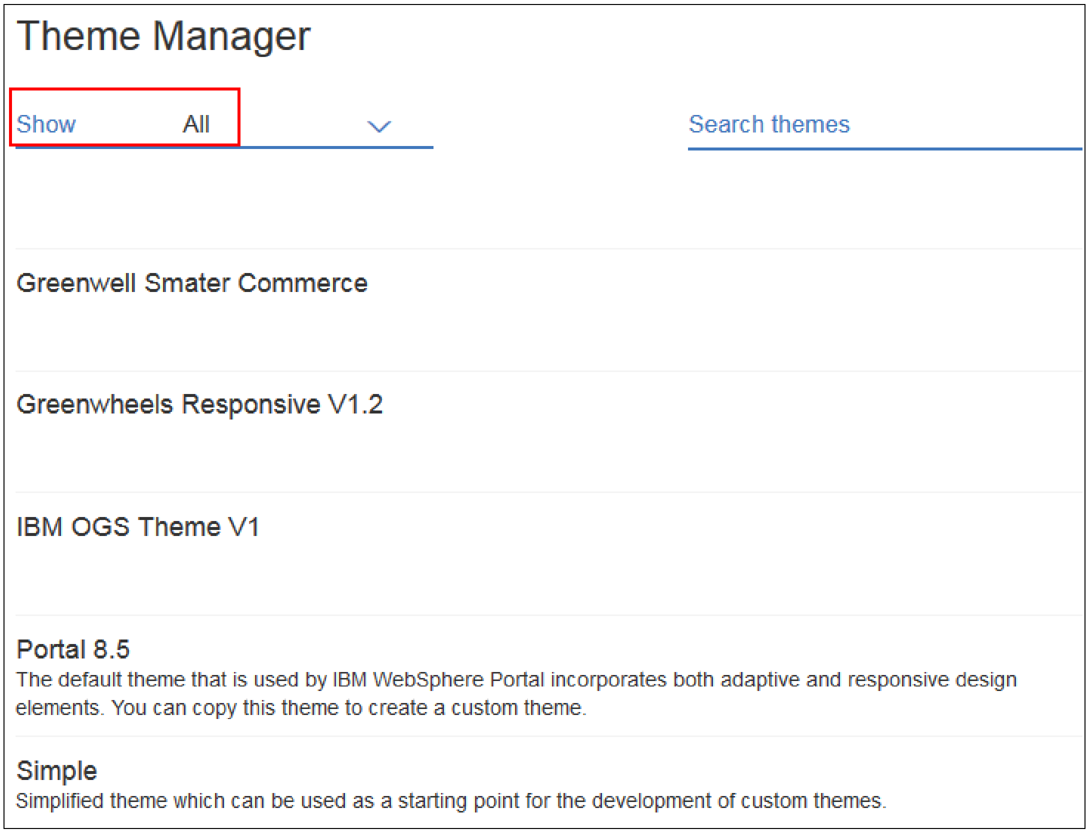
2. The Theme Manager tool is displayed, as shown in Figure 3-5.



Portal 8.5
The default theme that is used by IBM WebSphere Portal incorporates both adaptive and responsive design elements. You can copy this theme to create a

Figure 3-5 Theme Manager portlet

3. By default, the Theme Manager shows only the active themes. To list all the available themes, select **All** in the Show drop-down menu, as shown in Figure 3-6.



Greenwell Smater Commerce

Greenwheels Responsive V1.2

IBM OGS Theme V1

Portal 8.5
The default theme that is used by IBM WebSphere Portal incorporates both adaptive and responsive design elements. You can copy this theme to create a custom theme.

Simple
Simplified theme which can be used as a starting point for the development of custom themes.

Figure 3-6 Display all the available themes

The following Theme Manager topics are covered in this section:

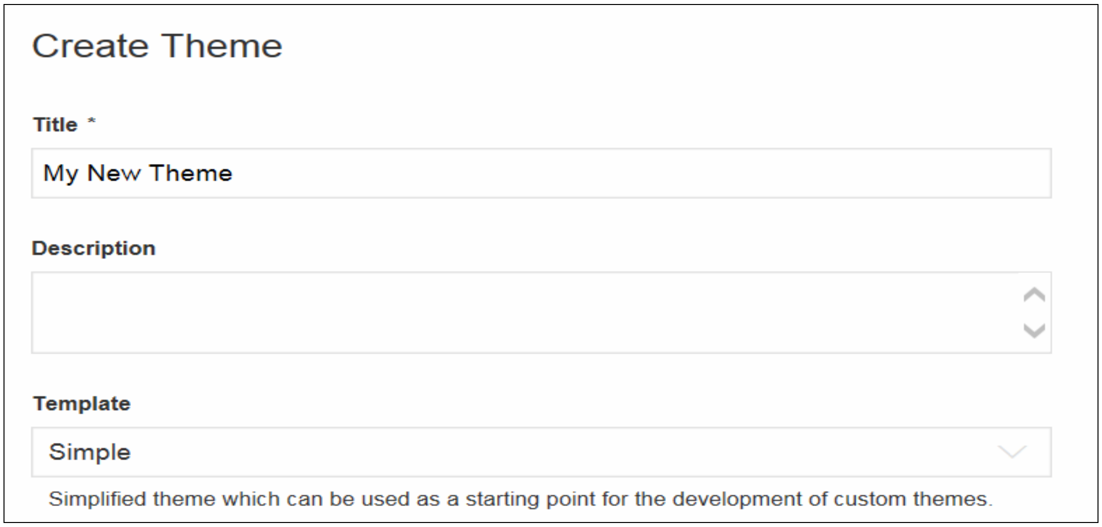
- 3.2.1, “Creating a theme” on page 42
- 3.2.2, “Copying a theme” on page 43
- 3.2.3, “Managing theme properties” on page 45

For more information about how to create manually a custom theme, see Appendix B, “Creating a custom theme” on page 225.

3.2.1 Creating a theme

To create a theme, complete the following steps:

1. In Theme Manager, click **Create Theme** (see Figure 3-5 on page 41).
2. On the Create Theme window, enter the following information:
   – A theme title and an optional theme description, as shown in Figure 3-7.



Create Theme

Title *

My New Theme

Description

Template

Simple

Simplified theme which can be used as a starting point for the development of custom themes.

Figure 3-7 Create a theme

> **Note:** By default, the theme title that is used is the system name. The system name is used for the WebDAV path, the theme unique name, and the skin unique name.

   – Select a theme template. The Portal8.5 theme and Simple theme templates are shipped with WebSphere Portal V8.5 and CF 08 respectively, and are available for immediate use. You can also create a custom theme template.

   – Enter a theme title that is unique and includes only valid characters.

3. Click **Create**.

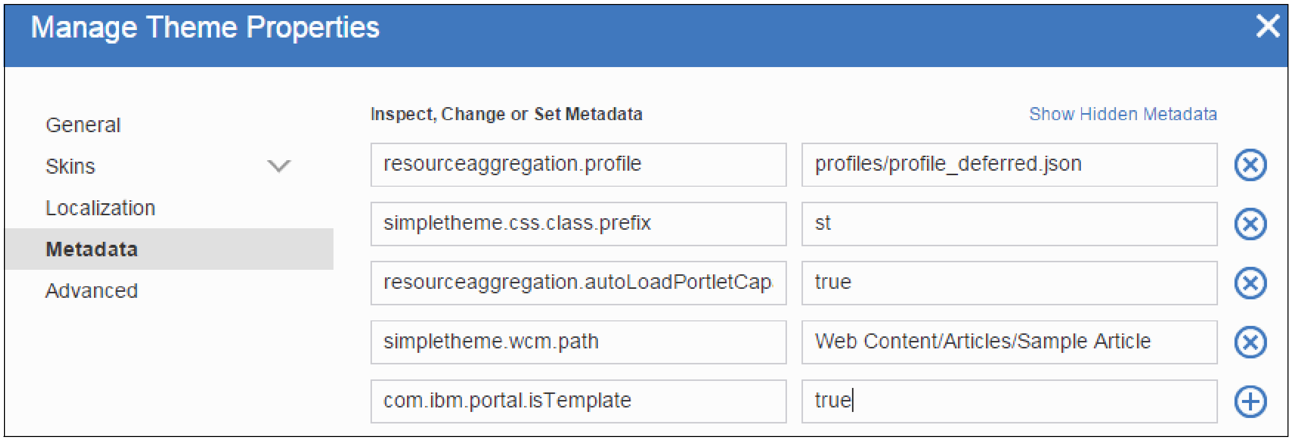
When the theme is created successfully, a success message is displayed; otherwise, an error message is displayed.

You can designate a theme that you created as a template. Only WebDAV-based themes are supported as templates. To designate your own theme as a template, add the following theme metadata to your theme:

```
com.ibm.portal.isTemplate = true
```

To add the metadata to your theme, complete the following steps:

1. In the Theme Manager (see Figure 3-5 on page 41), click the gear icon (Manage properties) to open the Manage Theme Properties window,
2. On the Manage Theme Properties window, click **Metadata**, as shown in Figure 3-8.



resourceaggregation.profile	profiles/profile_deferred.json
simpletheme.css.class.prefix	st
resourceaggregation.autoLoadPortletCap	true
simpletheme.wcm.path	Web Content/Articles/Sample Article
com.ibm.portal.isTemplate	true

Figure 3-8 Add theme metadata

3. Add the following theme metadata to your theme and click the plus icon:

   ```
   com.ibm.portal.isTemplate = true
   ```

4. Click **Done**. Now, your theme can be used as a theme template.

3.2.2 Copying a theme

A simple way to start creating your own custom themes is by copying the WebSphere Portal default theme. This approach ensures that your theme has all the required elements for the theme to function.

> **Note:** Do not modify the standard WebSphere Portal theme directly because future fix packs and service changes can overwrite your changes.

For more information about customizing with themes, see the following sources:

- *Developing Themes for WebSphere Portal 8.5* article:

  `http://www.lotus.com/ldd/portalwiki.nsf/xpViewCategories.xsp?lookupName=Developing%20Themes%20for%20WebSphere%20Portal%208.5`

- The *Customizing the theme* topic in the IBM WebSphere Portal V8.5.0 IBM Knowledge Center:

  `http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/dev-theme/themeopt_cust.dita`

- Appendix B, “Creating a custom theme” on page 225

You can quickly copy an existing theme by using the Theme Manager by completing the following steps:

1. In the Theme Manager, click the **More actions** icon and click **Copy** (Figure 3-9).

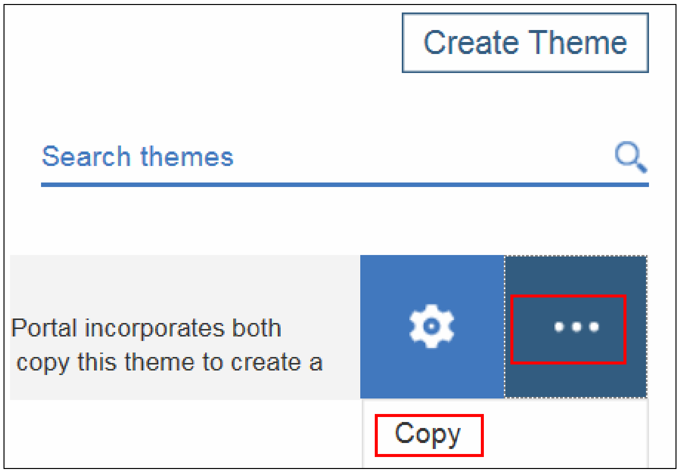


Figure 3-9 Copy a theme

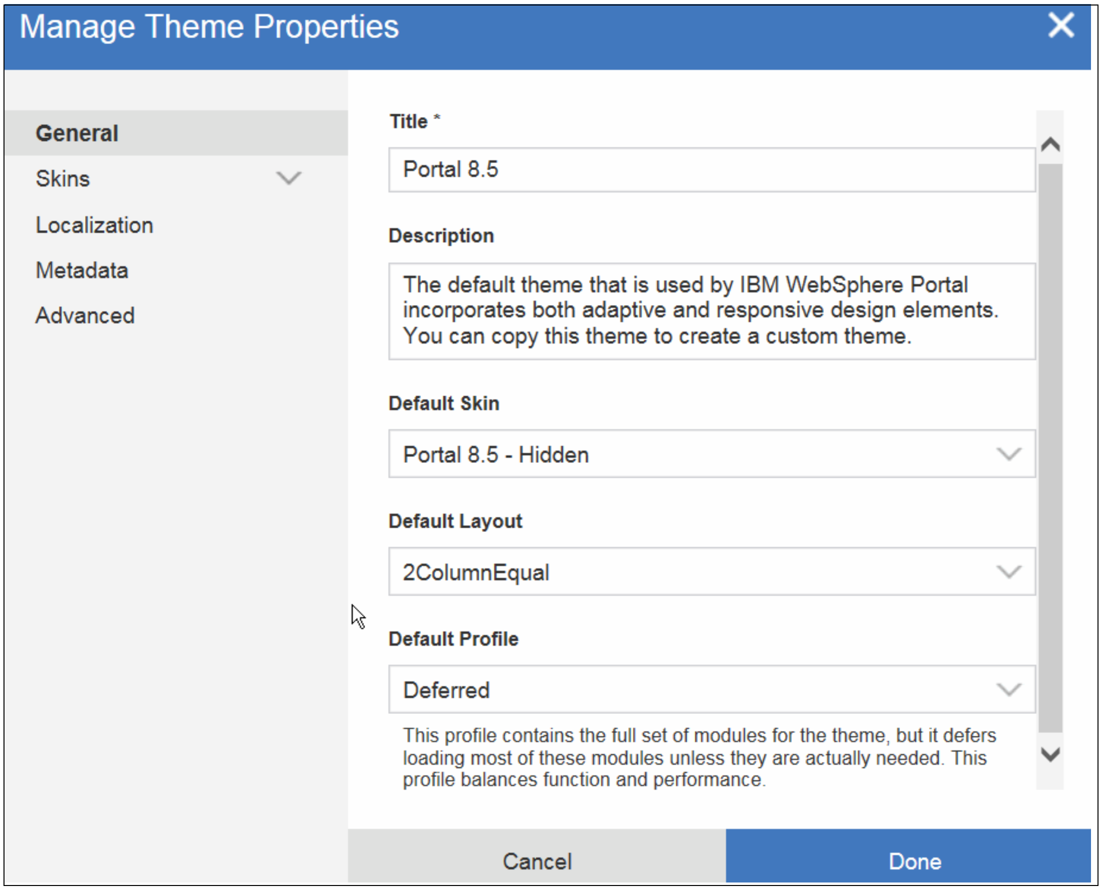
2. Enter a theme title and an optional theme description, and click **Done**.

   When you copy an existing theme, the system automatically titles the copied theme with the name of the existing theme and the word *copy*. For example, the copy of the existing theme `My Theme` would be titled `My Theme copy`.

   By default, the theme title is used as the system name. The system name is used for the WebDAV path, the theme unique name, and the skin unique name.

3.2.3 Managing theme properties

You can modify the properties of a theme and its skins by clicking the gear icon (Manage properties) in the Theme Manager portlet to open the Manage Theme Properties window, which is shown in Figure 3-10. In this window, you can modify the properties of both WAR-based and WebDAV-based themes. However, not all modification options are available for WAR-based themes.



Title *

Portal 8.5

Description

The default theme that is used by IBM WebSphere Portal incorporates both adaptive and responsive design elements. You can copy this theme to create a custom theme.

Default Skin

Portal 8.5 - Hidden

Default Layout

2ColumnEqual

Default Profile

Deferred

This profile contains the full set of modules for the theme, but it defers loading most of these modules unless they are actually needed. This profile balances function and performance.

Figure 3-10 Edit theme properties

When you select a theme in the Theme Manager and open the Manage Theme Properties window, all of the properties for the theme and its skins are displayed. The window includes five panes: General, Skins, Localization, Metadata, and Advanced.

After modifying the properties of a theme, click **Done** to save your changes and return to the Theme Manager. If your changes cannot be validated, an error message is displayed and the window remains open. To return to the Theme Manager without saving your changes, click **Cancel**.

3.3 Theme optimization

Theme optimization provides a framework that minimizes the number of requests by aggregating resources, minimizes download size by controlling the capabilities that are needed for a certain scenario, and improves theme performance.

Theme optimization uses modules and profiles, and by applying these concepts, it is possible to turn on and off an arbitrary number of features for certain pages. You can pull in theme resources only when needed, and develop modules independent of each other.

The module framework allows extensions to contribute to different areas of the page to provide flexibility, enhance the user experience, and maximize performance.

The intention of the theme modularization framework is to decouple feature enablement from the theme code itself. You can use this technique to develop themes more easily from an outside-in approach with little knowledge about the details about how the underlying code for a particular feature in WebSphere Portal works.

The framework provides logical points where modules can contribute data into a theme at run time and to optimize those contributions by combining them where possible. Multiple disparate remote sources can be combined into one request for greater performance.

You also can use theme optimization to enable and disable features of a theme by using a simple module inclusion profile. Theme developers can spend their time focusing on the interface design of the theme itself without worrying about the details of how to get portal features to work correctly within their theme. It also provides them an easy way of turning off features that they do not need in one environment that they might use in another environment.

3.3.1 Theme Optimization Analyzer

Use the Theme Optimization Analyzer to view, but not edit, all parts of the theme optimization framework inside of WebSphere Portal, including pages, profiles, modules, and so on. You can use it for easier development and troubleshooting. The validation report analyzes your theme and theme components for known issues and reports the number of errors, warnings, and informational messages. It also includes a detailed explanation about how to fix the errors that occur.

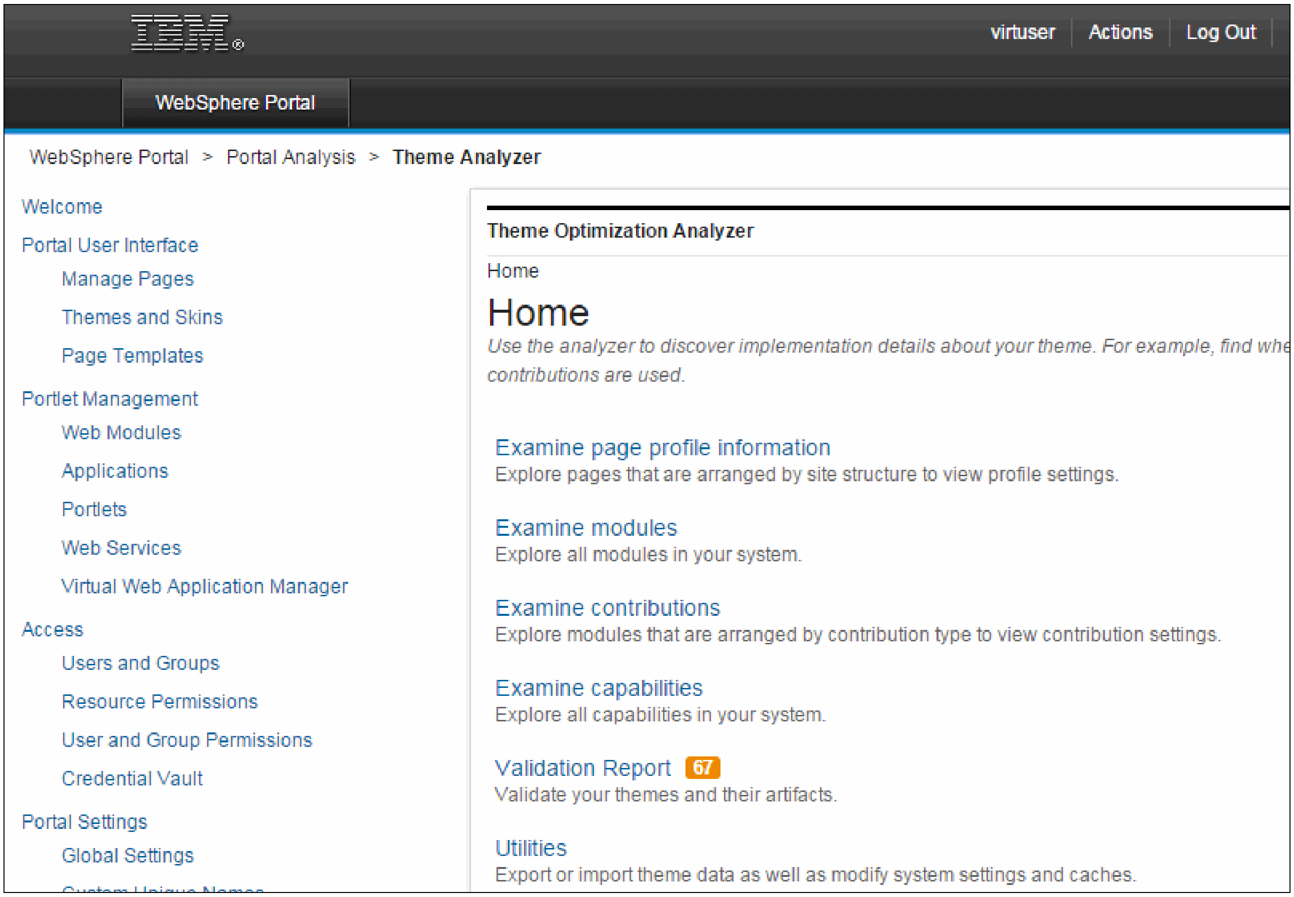
You can use the Theme Optimization Analyzer portlet to see which pages have specific profiles that are set or inherited. You can also see which profiles are available and belong to which theme.

Additionally, you can browse and explore all aspects of the available modules. You can see what modules are loaded for a specific profile or all modules of the entire system. You can drill down into the dependency hierarchy to understand interdependencies and get different views on it, such as a parent view. The module explorer also features a rich search set so that you can easily find modules that contribute certain resources or capabilities, or browse all exposed data.

To access the Theme Optimization Analyzer, complete the following steps:

1. Open the WebSphere Portal administration console.
2. Click **Portal Analysis** → **Theme Analyzer**.

3. The Theme Optimization Analyzer portlet is displayed, as shown in Figure 3-11.



virtuser | Actions | Log Out

WebSphere Portal

WebSphere Portal > Portal Analysis > Theme Analyzer

Welcome
Portal User Interface
Manage Pages
Themes and Skins
Page Templates
Portlet Management
Web Modules
Applications
Portlets
Web Services
Virtual Web Application Manager
Access
Users and Groups
Resource Permissions
User and Group Permissions
Credential Vault
Portal Settings
Global Settings

Theme Optimization Analyzer

Home

Home

Use the analyzer to discover implementation details about your theme. For example, find whe contributions are used.

Examine page profile information
Explore pages that are arranged by site structure to view profile settings.

Examine modules
Explore all modules in your system.

Examine contributions
Explore modules that are arranged by contribution type to view contribution settings.

Examine capabilities
Explore all capabilities in your system.

Validation Report 67
Validate your themes and their artifacts.

Utilities
Export or import theme data as well as modify system settings and caches.

Figure 3-11 Theme Optimization Analyzer portlet

You can also access the Theme Optimization Analyzer by clicking **Open applications menu** → **Theme Development** and then clicking the Theme Analyzer icon (Figure 3-12).

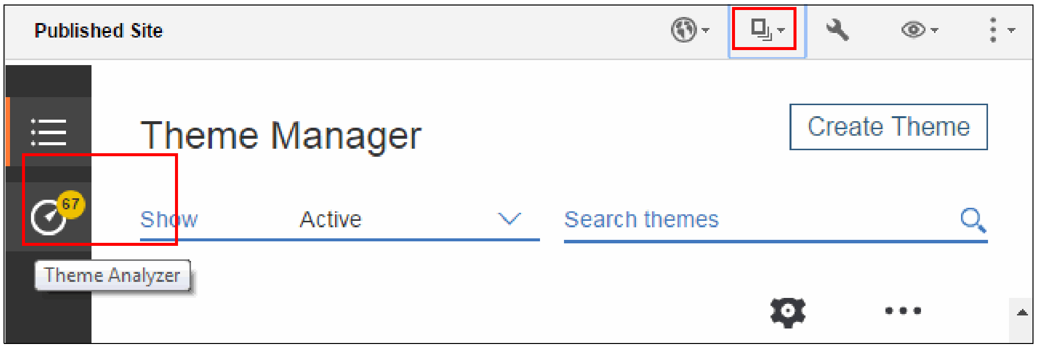


Figure 3-12 Access the Theme Optimization Analyzer portlet

For more information, see the *Theme Optimization Analyzer* topic in IBM WebSphere Portal V8.5 IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/dev-theme/themeopt_an_analyzer.dita

3.4 Managing themes by using IBM Digital Experience File Sync

DXSync synchronizes all of your static theme resources with your local development environment so that you can modify them with your favorite editor. DXSync synchronizes any WebDAV-based WebSphere Portal themes with your local workstation. It can be used instead of a WebDAV client.

DXSync watches file system changes in the background. It pulls the theme files to your workstation and uploads any changes you make. DXSync runs on Windows, OSX, and Linux environments. It supports full two-way synchronization and recognizes merge conflicts.

Note: DXSync is unsupported, but can be used on WebSphere Portal Version 8.5.

For information about how to install and connect to DXSync, see Appendix A, “Installing and connecting with IBM Digital Experience File Sync” on page 219.

For more information about how to install, initialize, troubleshoot, and use DXSync, see the following website:

`https://github.com/digexp/dxsync/wiki`

3.5 Customizing the theme

Themes can be customized to provide flexibility, enhance the user experience, and maximize performance. For a detailed description of various theme customization options, see the *Customizing the theme* topic in the WebSphere Portal V8.5.0 IBM Knowledge Center:

`http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/dev-theme/themeopt_cust.dita?lang=en`

The following sections provide information about customizing some of the theme resources:

- 3.5.1, “Changing the theme logo” on page 48
- 3.5.2, “Customizing the header and navigation” on page 49
- 3.5.3, “Customizing the footer” on page 50
- 3.5.4, “Working with layout templates” on page 50
- 3.5.5, “Dynamic content spots” on page 51
- 3.5.6, “Working with skins” on page 51

3.5.1 Changing the theme logo

You can change the theme logo to customize your portal site and rebrand it to reflect your business. The default logo that appears in the banner is a blank placeholder image.

Example 3-1 shows the default code that displays the blank placeholder image.

Example 3-1 Code that displays the default logo

```
<span class="wpthemeBranding">
     <img alt="IBM Logo"
src="/wps/themeModules/themes/html/dynamicSpots/icons/blank.gif">
     <span class="wpthemeAltText">IBM Logo</span>
</span>
```

This logo is provided to the theme with a CSS style class that can be modified or overridden by packaged styles on the customization shelf. It also updates as new styles are selected. You do not have to modify the theme template.

To change the theme logo in `.wpthemeLogo` in the following files, complete the following steps:

- `fs-type1:themes\Your_custom_theme\css\default\default_view.css`
- `fs-type1:themes\Your_custom_theme\css\default\default_view.css.uncompressed.css`

1. Remove the display attribute from the style class.
2. Define a path to your image for the background-image attribute.
3. Modify the height and width.

Note: The Simple theme uses SVG images for the logo, so these steps do not apply.

To change your header text to display your logo next to the navigation, complete the following steps:

1. Apply the white style on the customization shelf.
2. In the `fs-type1:themes\Your_custom_theme\nls\theme_locale.html` file, enter the following block of code, where IBM WebSphere Portal is the name of your portal site:

   ```
   <div class="wpthemeLogo wpthemeLeft">
        <span class="wpthemeAltText">IBM WebSphere Portal</span>
   </div>
   ```

Note: There is no HTML image element to display the logo. It is provided to the theme through a CSS style class. You do not have to modify the theme template. Also, for a Simple theme, these steps do not apply, as this is the default.

You can also make the changes that are described in this section by using DXSync, as described in 3.4, “Managing themes by using IBM Digital Experience File Sync” on page 48.

3.5.2 Customizing the header and navigation

The `head.jsp` file provides the header information for the page, which is the dynamic content for the HTML head element. The `navigation.jsp` file provides the navigation controls for the pages.

These JSP files are in the following directory:

```
<theme_name>/installedApps/DefaultTheme85.ear/DefaultTheme85.war/themes/html/dynam
icSpots
```

You can use dynamic content spots to determine what is displayed by top, primary, and secondary navigation. Use the navigation.jsp file to map properties to the dynamic content spot IDs in the theme.html file. Rendering of the navigation is done with a single JSP file with <ul> and <li> tags. For more information about dynamic content spots, see 3.5.5, "Dynamic content spots" on page 51.

When you customized your existing theme, it is possible that you modified or replaced the existing navigation. To preserve those changes in your new modularized theme, consider the following items:

- The dynamic spot mechanism in the theme.html files still applies to modularized themes, so your navigation.jsp still works automatically through the dynamic spot in your theme.html files.
- If your navigation is complex and uses its own resources such as .js or .css files, then consider making it a new module. The resources can then be logically grouped in one place and optimally combined when loaded.

For more information about customizing the navigation, see the *Customizing navigation* topic in the IBM WebSphere Portal V8.5 IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/dev-theme/themeopt_cust_nav.dita?lang=en

3.5.3 Customizing the footer

The footer.jsp file, found in the <ThemeName>/themes/html/dynamicSpots directory, provides the footer information for the page.

When customizing an existing modularized theme, the dynamic spot mechanism in the theme.html files still applies, so the footer.jsp file still works automatically through the dynamic spot in your theme.html files.

If your footer is complex and uses its own resources such as .js or .css files, then consider making it a new module. The resources can then be logically grouped in one place and optimally combined when loaded.

3.5.4 Working with layout templates

You can add layout templates so that they can be assigned to portal pages or update the ready-use layouts to modify the existing skins or change how the pages display.

Layout templates control the layout and positioning of the content on a page. The static layout template is called layout.html. The author of the layout template defines the HTML fragment markup and CSS for the layout of the page. The HTML fragment uses a microformat to specify containers and components, such as portlets and iWidgets to include in the page.

The layout templates are stored in the following WebDAV file store because the layout templates are scoped per theme:

```
/fs-type1/themes/myCustomTheme/layout-templates
```

To add a layout template so that it can be assigned to portal pages, create a folder or copy and rename an existing layout template folder inside the root folder. Include all CSS files in the theme that are used by the new layout template.

For more information about working with layout templates, see the *Layouts* topic in the IBM WebSphere Portal V8.5 IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/dev-theme/themeopt_cust_layout.dita?lang=en

3.5.5 Dynamic content spots

After you create a theme, you can add dynamic content to the static content of your custom theme by using dynamic content spots. Dynamic content spots are hook points in your static markup, such as the `theme.html` file that delegates somewhere else to inject more markup at that spot of the page. It can be delegated to any URI that is known to the system. For example, it can point to a server-side component, such as a JSP or servlet.

The static template files use dynamic content spots to reference JSP files or other dynamic resources. The dynamic resources are stored in a WAR file.

You can add dynamic content to your custom theme by using either client-side or server-side logic.

For more information about working with dynamic content spots, see the *Dynamic content spots* topic in the IBM WebSphere Portal V8.5 IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/dev-theme/themeopt_cust_jsp.dita?lang=en

3.5.6 Working with skins

You can apply ready-use skins to your portal pages, modify the existing skins, or add your own custom skin to change how your pages display.

Skins work with themes to define the appearance of portlets. For example, the stylesheets in the theme specify the color and font of the portlet title bar. The skin also defines the shape of the title bar. To create a complete visual package, design the theme and skin together and package as one. You can tightly control the visual elements of the portal by packaging one theme with one skin.

For more information about working with skins, see the *Skins* topic in the IBM WebSphere Portal V8.5 IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/dev-theme/themeopt_cust_scopeskin.dita?lang=en

Skin templates

The static skin template `skin.html` is in the root directory of the skin folder of the WebDAV file store. The `skin.html` file provides the full markup for decoration around a portlet. As with theme templates, you can use dynamic content spots to add dynamic elements to the skin template at run time.

WebSphere Portal provides a base skin template and localized skin templates.

The static skin template `skin.html` file is in the root directory of the skin in the following WebDAV file store because skins are scoped per theme:

```
\fs-type1:\themes\custom_theme\skins\<skin-name>\
```

There are localized `skin.html` files within the following directory to support national languages:

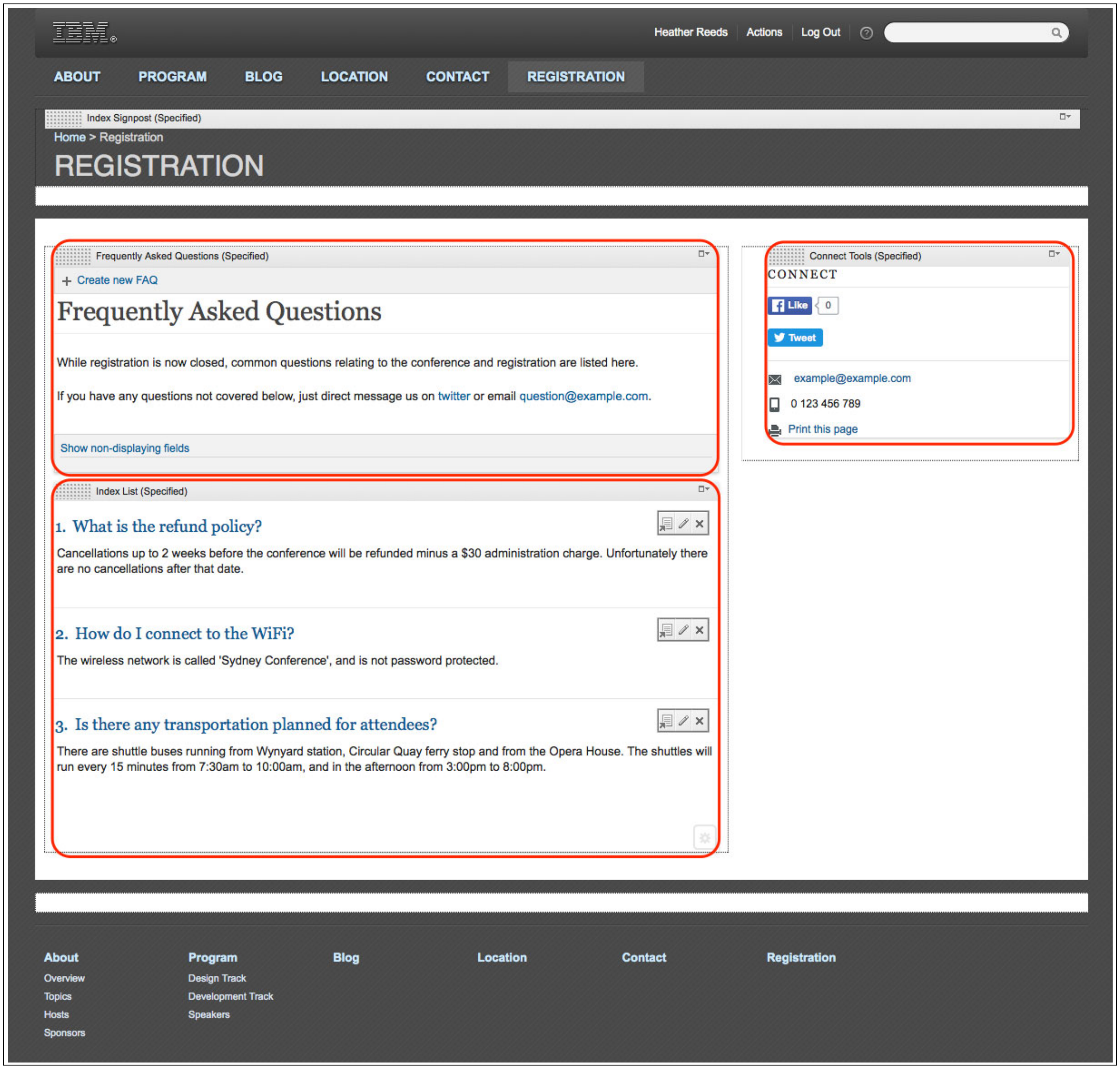
```
\fs-type1:\themes\custom_theme\skins\<skin-name>\nls
```

Default skins

The default theme includes the following skins that can be used as a basis for creating custom skins:

- Hidden skin: The Hidden skin is the default skin. In view mode, this skin does not display the title bar or any decorations. In edit mode, the skin displays the title bar and decorations around the portlet. This skin was created for Web Content Manager content that is exposed on external facing websites, where edit mode is not likely to be shown to users so the title bar is hidden. However, administrators or users that have authority to edit the Web Content Manager portlet can put a page into edit mode and configure the portlet.
- HiddenPlus skin: The HiddenPlus skin displays only the portlet title bar when you hover your cursor over the portlet in edit mode. In view mode, this skin does not display the title bar or any decorations.
- Standard skin: The Standard skin displays the portlet title bar in both edit and view mode. The title bar shows the title of the portlet and the skin action menu. This skin imitates the default skin from past releases.
- NoSkin skin: The NoSkin skin does not show the title bar or any decorations around the portlet. This skin is used if you do not need to show the skin actions.

Figure 3-13 shows the skins around the portlets on a sample web page.



Frequently Asked Questions

While registration is now closed, common questions relating to the conference and registration are listed here.

If you have any questions not covered below, just direct message us on twitter or email question@example.com.

1. What is the refund policy?

Cancellations up to 2 weeks before the conference will be refunded minus a $30 administration charge. Unfortunately there are no cancellations after that date.

2. How do I connect to the WiFi?

The wireless network is called 'Sydney Conference', and is not password protected.

3. Is there any transportation planned for attendees?

There are shuttle buses running from Wynyard station, Circular Quay ferry stop and from the Opera House. The shuttles will run every 15 minutes from 7:30am to 10:00am, and in the afternoon from 3:00pm to 8:00pm.

example@example.com

0 123 456 789

Figure 3-13 Skins around the portlets on a web page

Creating skins

A preferred approach to creating custom skins is to copy one of the default skins files, then add images, JavaScript files, and other custom resources.

> **Tip:** Do not modify the default skins directly because this skin can be updated by service fix packs and override your changes.

You can create skins in the following scopes:

- Theme-based: You can scope a skin to one particular theme. Use this scope if the skin relies specifically on code within the theme or has a specific function that is useful only in that particular theme. These skins are under the root folder of the theme, and create a specific link between the skin and theme. This is the preferred approach.
- Global: The global scope where the skin is not specific for any theme. Global skins are in the skins folder under the WebDAV root.

3.6 Support for mobile devices

This section describes how to incorporate responsive design features to render content on mobile devices.

During the requirements gathering phase and when making macro design decisions, you should evaluate your target audience, their size, and profile. For example, consider factors such as number of users that use your site, their browsers, versions, and devices, and whether they are using desktops, tablets, or smartphones to browse the site.

If the website you build is an internal portal and most of the clients are tablets, then optimize the navigation patterns for tablets. Also, consider usability and accessibility factors, and network speed, for example, avoid rendering heavy content that takes long to load on mobile devices.

It is a preferred approach to tailor the theme to render differently for specific device types to make the navigation of the site easier for the users.

3.6.1 Mobile navigation markup

The mobile navigation markup for both smartphones and tablets is created by the `mobileNavigation.jsp` file of the theme that is found in the following directory:

```
PortalServer_root/theme/wp.theme.themes/default85/installedApps/DefaultTheme85.ear
/DefaultTheme85.war/themes/html/dynamicSpots
```

The `mobileNavigation.jsp` file is controlled by the mobile navigation dynamic spot in `theme.html`:

```
dyn-cs:id:85theme_mobileNavigation
```

The reason this `dyn-cs` loads is because of device class support. For more information, see 3.6.2, “Device class support” on page 55.

The navigation on mobile devices is rendered by the `mobileNavigation.jsp` file. Therefore, the primary, secondary, breadcrumb, and side navigation that is used on the desktop do not produce any output for a mobile device. The navigation is rendered for mobile, but is hidden when the page loads. On a tablet, the arrow button in the banner can be tapped to reveal the navigation. On a smartphone, the user can scroll up to see the first-level navigation pages.

Because smartphones have little space, site designers might want to hide certain first-level pages, such as Administration, for these devices. Add the `com.ibm.portal.mobile.Hidden` metadata to the page to hide certain first-level pages. By default, the Administration, Application, and Tag Center pages are hidden.

It might happen that navigation menu does not fit into the maximum height set by a style definition. If so, the height can be adjusted by editing the `mobilenav.css` file of the custom theme in the following WebDAV location:

`\fs-type1/themes/myCustomTheme/css/`

Look for the style declaration that is shown in Example 3-2, change the height, and save the file.

Example 3-2 Style declaration to change the navigation menu height

```
.wpthemeMobileNav ul.wpthemeExpandNav {
   /* navigation lists */
   max-height: 100em;
}
```

The WebSphere Portal V8.5 theme provides new responsive page navigation designs for mobile devices. One is aimed at smartphones, and the other is designed for tablets. The user agent for a device is parsed to determine which navigation to render on the portal page.

When you create portlets and the page elements within them, you can now define your own style overrides as needed, which also includes using media queries and device classes to make the web page responsive and adaptive to various resolutions and device types.

For more information about mobile navigation, see the *Mobile navigation* topic in the IBM WebSphere Portal V8.5 IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/rwd/rwd_add_navphone.dita?lang=en

3.6.2 Device class support

Device classes are used in WebSphere Portal as an abstraction for common properties for the device of a client. For example, tablet computers can be grouped into device class tablets because they share a form factor and possibly other traits, such as a touch interface, or additional hardware sensors.

Device classes can be implemented to organize clients into groups. After you define a device class, you can then assign it to clients. You can assign multiple device classes to a client to help scope your environment.

The following device classes are defined:

- Smartphone
- Tablet
- Android
- BlackBerry
- iemobile
- ios
- IBM MobileFirst

Note: You can create additional device classes, or you can remove any of these device classes.

You can use device classes in the following ways:

- Clients: A client can be assigned a device class by assigning a capability. The name of the capability must start with `com.ibm.portal.devicesupport.deviceclass=` and continue with the administrative name of a device class, as provided with their definition.
- Pages: Device classes can be used to filter pages from the content model.
- Layout templates: A layout template for a static portal page can specify variants for device classes that use a naming convention. The layout template that matches the client's device class can then be used to render the page.
- Theme modules: A contribution in a theme module can define subcontributions for a specific device class.

For more information about device classes, see the *Device class* topic in the IBM WebSphere Portal V8.5 IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/dev-theme/themeopt_devclass.dita?lang=en

3.7 Web Developer Toolkit for IBM Digital Experience

The new Web Developer Toolkit for IBM Digital Experience product is part of the continuing IBM initiative to provide greater support for modern web development and design tools and technology on the Digital Experience platform.

IBM is now offering an open source `Node.js`-based toolkit for Digital Experience developers and designers starting with WebSphere Portal Version 8.5 CF 08. This `Node.js`-based toolkit runs on a developer’s workstation. It can be used to work with the following artifacts:

- Applications: Script portlets
- Site design: Portal themes
- Content design: Web Content Manager presentation templates and components

The toolkit runs on Windows, Mac, and Linux environments by using `Node.js`.

The toolkit provides the following functions:

- Provides a dashboard UI for working with applications and designs:
  - Manages push, pull, and watch for synchronizing local files with the server
  - Includes other useful tools, such as a script application test server and tools for working with theme modules and profiles
- Enables designers to work with Web Content Manager design artifacts locally as plain HTML and image files:
  - Supports presentation templates and components (HTML, image, stylesheet, and text)
  - Includes pull, push, and watch functions for moving files to and from server
  - Available from the dashboard UI or by using a command-line interface (CLI)

Figure 3-14 shows the toolkit architecture.

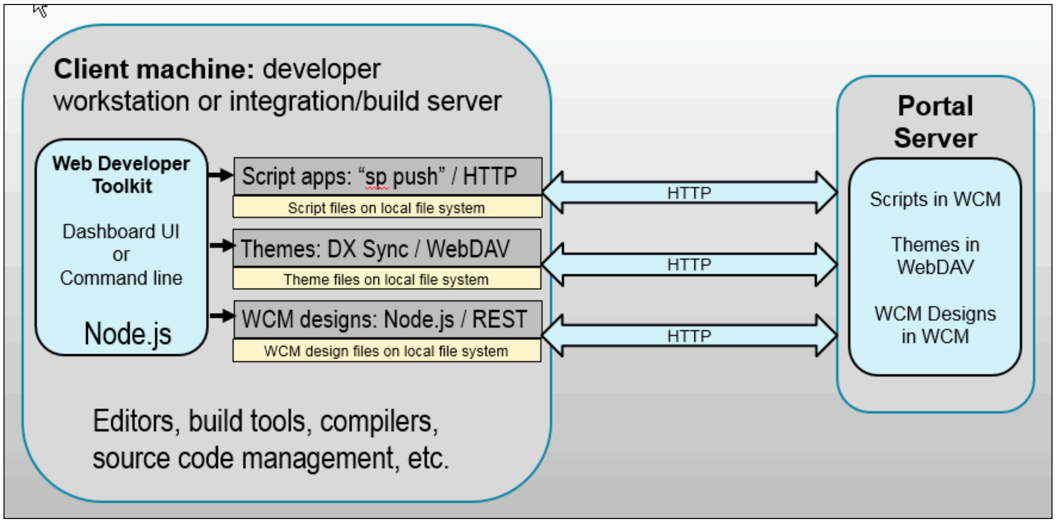

Client machine: developer workstation or integration/build server
Web Developer Toolkit
Dashboard UI or Command line
Node.js
Script apps: "sp push" / HTTP
Script files on local file system
Themes: DX Sync / WebDAV
Theme files on local file system
WCM designs: Node.js / REST
WCM design files on local file system
HTTP
HTTP
HTTP
Portal Server
Scripts in WCM
Themes in WebDAV
WCM Designs in WCM
Editors, build tools, compilers, source code management, etc.

Figure 3-14 Web Developer Toolkit for Digital Experience architecture

For more information about the Web Developer Toolkit for Digital Experience product, see the IBM Digital Experience Developer Center:

https://developer.ibm.com/digexp/

For installation instructions, see the GitHub repository:

https://github.com/OpenNTF/WebDevToolkitForDx

4

Building the website

Setting up a website includes creating pages, adding navigation, setting up search, and adding content to the site. As described in Chapter 3, “Getting started with IBM WebSphere Portal themes” on page 33, themes are used to customize the portal look-and-feel. This chapter describes how to build a simple website with IBM WebSphere Portal. It describes how to create content and other components with IBM Web Content Manager and how to render the content and components to build a fully functioning site.

This chapter covers the following topics:

For more information about building a website, see the *Setting up a website* topic in the IBM WebSphere Portal V8.5.0 IBM Knowledge Center:

https://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/site/site_setup.dita

4.1 Website building blocks

IBM WebSphere Portal and Web Content Manager are the foundation building blocks that you can use to create the components of a website framework, such as pages, navigation, themes, content, content libraries, and more.

The following components are some of the basic building blocks of building a website:

- Themes, profiles, and skins

  The theme controls the presentation layer of your website. The theme profile contains modules, and controls which modules to load per page. A skin surrounds the portlet on a page. For more information about theme and skins, see Chapter 3, “Getting started with IBM WebSphere Portal themes” on page 33.

- Pages

  A page is an organization element that contains portlets. Starting with WebSphere Portal Version 8.0, pages are stored in the portal database and can be exported by using XMLAccess, but now have additional data that is stored in Web Content Manager and acts as Web Content Manager site areas and can be syndicated. By managing portal pages from within Web Content Manager, you can apply features such as workflow, version control, and syndication to portal pages. For more information about pages, see 4.13, “Presentation templates” on page 115.

- Portlets

  Portlets are small, web-based applications that are independently developed, deployed, managed, and displayed. After the portlet is deployed, you can use it multiple times on different pages. WebSphere Portal also provides a rendering portlet to render content from the Web Content Manager system.

- Content

  Rich content and elegant user experiences drive the success of websites. Each digital experience consists of many different types of content from many various sources.

- Components

  You use components to store elements that are used in more than one area of your website. For example, a company logo or a copyright notice is stored in Web Content Manager Image or File components.

4.2 Planning a website

Before you build a website, it is considered a preferred practice to take the time to analyze, plan, and design the entire system that supports the website. You must plan the website and the web content system that you use to reuse artifacts that make up the site and manage the lifecycle of your contents. This type of planning is done by an information architect.

> **Important:** It is important to have an understanding of the objectives, deliverables, and scope of a web content system. Your website definition outlines the *what*, *why*, and *who* of the project and can be used throughout the life of your website.

You should also consider the roles and users that are involved in building a website and installing the infrastructure. Project managers, business analysts, architecture and design team, deployment team, and development team are examples of the roles and of the type of work that is performed by the people who create and manage a website.

A single person can be responsible for more than one of the roles. The roles that you implement in your organization to support your website are determined by the size and complexity of the system being deployed. Not all the roles are required for every website, but all aspects of these roles must be considered during any system deployment.

This section describes the following topics when planning for a website:

- 4.2.1, "Information architecture" on page 61
- 4.2.2, "Planning your website by using an analysis document" on page 62
- 4.2.3, "Designing a prototype website by using HTML" on page 64
- 4.2.4, "The design document" on page 64
- 4.2.5, "Roadmap to building a web content system" on page 66
- 4.2.6, "Roles and authorization" on page 67
- 4.2.7, "Creating reusable assets" on page 69
- 4.2.8, "Authoring tools" on page 69

4.2.1 Information architecture

The information architecture describes the information structure of the website and how users navigate through the site.

Site map

The website structure that the information architect designs determines what pages and website areas must be created to give your site a hierarchical structure. The *site map* describes the structure of the site and determines what pages and site areas are required for your site. For example:

- Home page
  - News page
  - Products page
    - Product site area 1
    - Product site area 2
    - Product site area 3
  - Downloads page
  - Support page

Content types

The content types that are identified by the information architect determine what authoring templates are required for your authoring system. For example, your website might require the following content types:

- Section home pages
- News articles
- Employee profiles
- Product information
- Photo galleries
- Legal disclaimers

Content profiling and taxonomies

The information architect is responsible for determining the information taxonomies (organization of information) that are required by users to profile content. This information determines the content that is displayed within menu components.

Here is an example of a taxonomy for a financial services company:

- Financial
  - Banking Solutions
  - Interest Rates
    - Personal
    - Business
    - Corporate
- News

4.2.2 Planning your website by using an analysis document

Use an analysis document to record the information that is gathered from stakeholders. It describes the design of the website, its content, and its features.

The following sections include some examples of the analysis that you can undertake when you design a web content system.

User analysis

To design a website that supports the needs of the company and the users, you must know your audience. It is important to determine your users at this early stage of the project. Here are some of the questions that you should answer during user analysis:

- Who are the users?
- Who are the most important groups?
- What do they want to do on the site?
- What makes them return to the site?
- What is their level of experience with the web?

To help you understand your main user groups even further, you can develop personas and scenarios:

- Personas

  A persona is a fictional person who represents a major user group for your site. By using the information that is gathered about your users, create a person who represents each main user group and give them the following information:

  - A name and picture
  - Demographics, such as age, education, and family status
  - Job role and responsibilities
  - Their goals and tasks in relation to the site
  - A background on their computer and web usage

- Scenarios

  A scenario is a story of how users might experience the site. Scenarios help you visualize the site and its users. They can help you view the navigation process as a whole. Scenarios are also useful in validating the website design after it is finished and can be used in usability testing. Use your persona, and give it a task to accomplish on the site. Write a story about how the character uses the site to finish the task.

Competitive analysis

If you are building a public website, it is useful to look at what the competition is doing. Generate a list of competitors and document things you like and dislike about their sites.

Website requirements

Website requirements describe the features and functions of a website. They document what the site must have and also what users can do. The requirements are not used to describe how to build the website, which is detailed in the design document (see 4.2.4, "The design document" on page 64).

For example, website requirements can include:

- General features:
  - Search
  - Contact details
- Allow users to accomplish the following tasks:
  - Purchase a product
  - Sign up for a newsletter
  - Complete a timesheet online
- Include the following content and site areas:
  - Press releases
  - Policies and guides
  - Links to related articles

Content inventory

It is useful to identify the types of content that make up the site. As your new website might be a redesign of an existing site, identify what content exists and what new content must be written. Create a content inventory and add any existing web pages and potential types of content of which you can think.

Types of content include the following items:

- Static content, such as copyright notices and privacy statements
- Dynamic content, such as latest news and product campaigns
- Transactional content, such as logon pages and registration pages for email newsletters

When you create a content inventory, you can collect the following information:

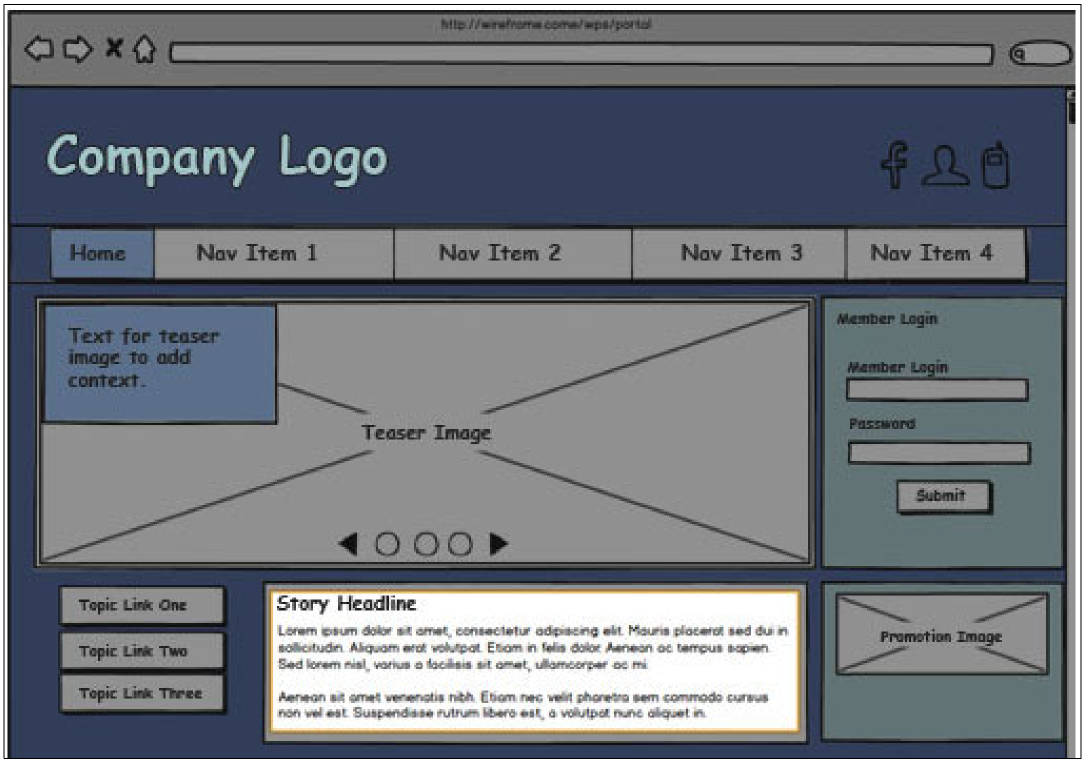
- A brief description
- Topic area or category
- Priority
- Format, such as a web page, a file, or on paper
- Intended audience
- Related content
- Created date
- Last modified date
- Owner
- Author
- Expiration date

4.2.3 Designing a prototype website by using HTML

Before you create a design document for your web content system, it can be useful to create a prototype of your site by using HTML. This prototype is based on the outline that is defined in your project plan and the data that is gathered in your analysis document.

The site structure, design, and HTML code that you develop for your prototype can be used as the basis for many of the items that are defined in your design document.

Figure 4-1 show a sample HTML design prototype.



Text for teaser image to add context.

Story Headline

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris placerat sed dui in sollicitudin. Aliquam erat volutpat. Etiam in felis dolor. Aenean ac tempus sapien. Sed lorem nisl, varius a facilisis sit amet, ullamcorper ac mi.

Aenean sit amet venenatis nibh. Etiam nec velit pharetra sem commodo cursus non vel est. Suspendisse rutrum libero est, a volutpat nunc aliquet in.

Figure 4-1 Sample HTML design prototype

4.2.4 The design document

When you have defined the project and created an analysis document, you then define the requirements of your web content system in a design document. The design document outlines what types of content is needed for your site, how it is structured, how content is authored, and what the final website looks like. The design document includes all or some of the following items:

- Server architecture

  Technical architects define what web content environments are required for your system and the servers that are required for each environment. This information ensures that you have sufficient hardware to support your web content system.

- Security architecture

  The security architecture describes what groups are required for your site and what access is required for different groups to the authoring portlet and rendered website.

- Information architecture

  The information architecture describes the information structure of the site and how users browse through the site.

- Design architecture

  The design architecture describes what your website looks like, and what components are needed to build your site. You must define presentation templates, components, and themes.

- Authoring architecture

  The authoring architecture describes what types of content are required for the site and what change management strategies are applied when you update content and design. You must define what authoring templates are required for your system, what workflows are used to manage changes, and what folders are used to group items in the authoring portlet.

- Content acquisition architecture

  The content acquisition architecture is used to define what existing content is imported into your web content system, how it is imported, and what content must be created.

- Delivery architecture

  Your technical architect and information architect must define what delivery methods are most appropriate for the website that you are delivering.

- Maintenance architecture

  You must plan for the tasks that maintain the health and integrity of your web content system. Your technical architect and database architect must define what maintenance procedures are required for your system, and when and how often they must be run.

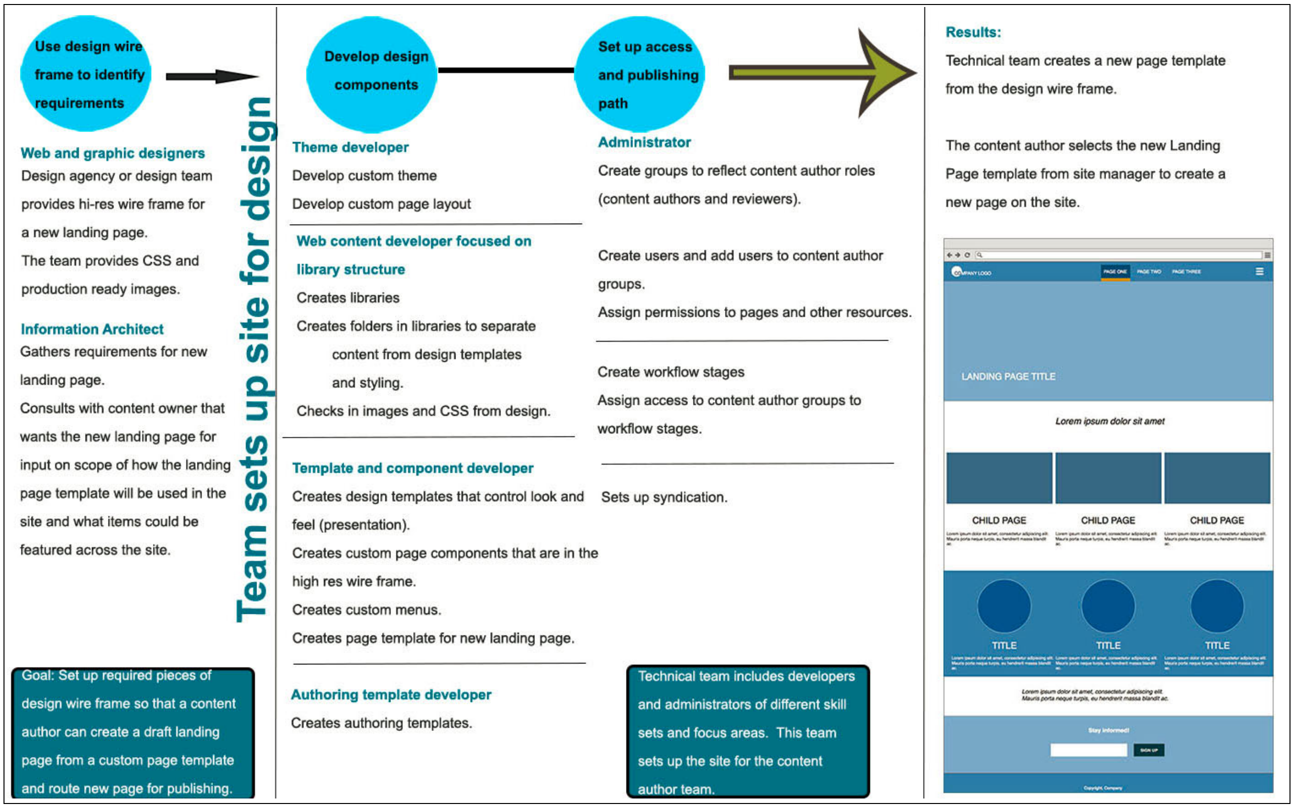
4.2.5 Roadmap to building a web content system

To build a web content system, you must design an authoring system, configure a delivery environment, and enable syndication.

Figure 4-2 shows the high-level steps to create a roadmap to building a web content system and provides an example of a typical flow for planning, designing, and building a website. It also includes the roles that are required to implement the roadmap and the tasks that are assigned to those roles.

The WebSphere Portal technical design and implementation team includes a range of skill levels and focus areas. To select the roadmaps that best fit the focus areas of your technical teams, see the *Roadmaps* topic in the IBM WebSphere Portal V8.5.0 IBM Knowledge Center:

https://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/install/deployment_patterns.dita



Use design wire frame to identify requirements

Web and graphic designers
Design agency or design team provides hi-res wire frame for a new landing page.
The team provides CSS and production ready images.

Information Architect
Gathers requirements for new landing page.
Consults with content owner that wants the new landing page for input on scope of how the landing page template will be used in the site and what items could be featured across the site.

Goal: Set up required pieces of design wire frame so that a content author can create a draft landing page from a custom page template and route new page for publishing.

Team sets up site for design

Develop design components

Theme developer
Develop custom theme
Develop custom page layout

Web content developer focused on library structure
Creates libraries
Creates folders in libraries to separate content from design templates and styling.
Checks in images and CSS from design.

Template and component developer
Creates design templates that control look and feel (presentation).
Creates custom page components that are in the high res wire frame.
Creates custom menus.
Creates page template for new landing page.

Authoring template developer
Creates authoring templates.

Set up access and publishing path

Administrator
Create groups to reflect content author roles (content authors and reviewers).

Create users and add users to content author groups.
Assign permissions to pages and other resources.

Create workflow stages
Assign access to content author groups to workflow stages.

Sets up syndication.

Technical team includes developers and administrators of different skill sets and focus areas. This team sets up the site for the content author team.

Results:
Technical team creates a new page template from the design wire frame.

The content author selects the new Landing Page template from site manager to create a new page on the site.

COMPANY LOGO
PAGE ONE
PAGE TWO
PAGE THREE
LANDING PAGE TITLE
Lorem ipsum dolor sit amet
CHILD PAGE
CHILD PAGE
CHILD PAGE
TITLE
TITLE
TITLE
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris porta neque turpis, eu hendrerit massa blandit ac.
Stay informed!
SIGN UP
Copyright, Company

Figure 4-2 Plan a website - design flow example

Typically, you separate roles between a content author and user interface editor or developer.

The content author starts the publishing process of new or existing content that goes through a workflow, for example, the content starts in a draft stage. The content author edits or enters data (for example, text, images, references, or links) in the content fields that are defined by authoring template on which the content is based. The content author then submits content for the next workflow stage (workflows can be simple, or express workflows, or complex, consisting of several stages, such as approval or expiration, and include custom actions). Now, the content is delivered and published on the website.

The user interface developer focuses on how content is presented on the page. In particular, the presentation template contains HTML code or references to other components that define this situation. How the content is displayed can be based on the web user profile, security data, geographical location, or role in the organization that is defined by the security model, taxonomy, or site structure.

You can also use various aggregation components, such as navigator or a menu that aggregate multiple content items based on certain criteria, for example content belonging to the same category is placed in the same site area. These design elements contribute to the information architecture, which should be established and documented before starting to develop and populate the Web Content Management system. Business requirements that include non-functional considerations should also be a driving factor during this planning and designing process.

4.2.6 Roles and authorization

You can restrict access to selected users and groups to the views within the Web Content Manager authoring portlet, the items that are managed by the authoring portlet, and to elements and pages that are displayed within a website.

There are three levels of access controls for web content:

- Web Content library

  This library contains all the Web Content Manager content and components. For more information about the Web Content library, see 4.6, “Content library” on page 82.

  Library-level access controls access to the library as a whole. If granted access, it provides an entry point to the library. A user needs at least contributor access to a library to have access to it on the authoring portlet.

- Item type per library

  Item-type access controls define the item type views and tasks that a user can access within the authoring portlet for a particular library. The permissions that are set for item types in a library do not automatically give you access to individual items. They give you access only to specific tasks and views within the authoring portlet.

- Item level

  Item-level access controls the actions that a user can run on an individual item. If the items are workflowed, the access control is set by using the workflow so that users have different roles based on the workflow stage.

  For example, users that have the manager role on a component type have access to the Purge and Unlock actions for that component. However, these actions are not available unless the user also has the manager role on the individual component and on the component type. The manager role must be granted to both the *type* and the *item* for the user to perform these actions.

For more information about roles and authorization, see the *Roadmap: Creating content authors, groups, and setting permissions* topic in the IBM WebSphere Portal V8.5.0 IBM Knowledge Center:

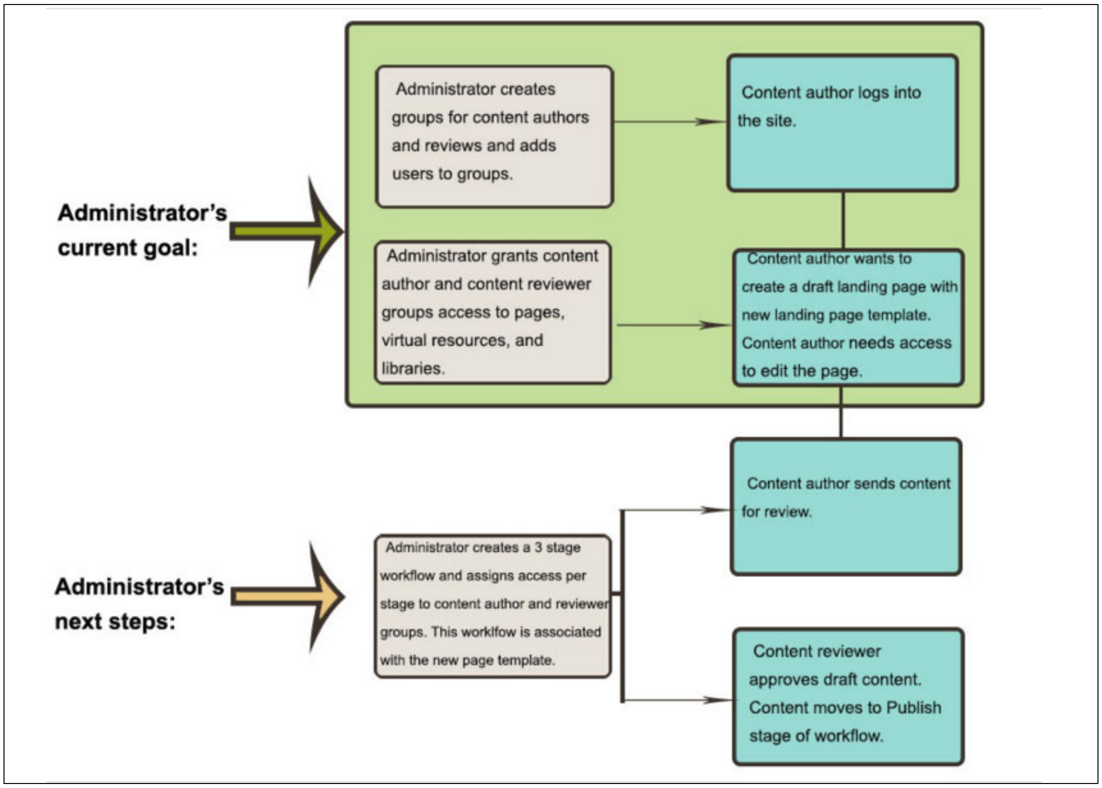
http://www.ibm.com/support/knowledgecenter/SSYJ99_8.5.0/install/rm_create_site_admin_access.dita

An administrator must understand the roles and goals of content authors, reviewers, and approvers to grant the appropriate permissions. For example, a content author might want to create a landing page from a page template. For this type of site update, the content author creates a project and then creates a page from a page template in site manager. To accomplish these goals, the content author must exist in the system and belong to a content author group. The content author needs access to projects to create or enter a project, and access to other resources types, such as pages, to create a draft page.

Next, a content reviewer role is introduced to the content authoring story. You must add the content reviewer to the system and add the content reviewer to a content reviewer group. By adding users with similar roles to the same group, you can quickly assign access to a group rather than to individual users.

Finally, you must add publishers as users to the system and create a publishing group to contain them. You want publishers to have the same authoring capabilities as content authors and reviewers, but with additional authority to publish site updates.

Figure 4-3 shows a flow chart that highlights the content authors and reviewers goals and the tasks that administrators must complete for these roles to accomplish their goals.



Administrator's current goal:
Administrator creates groups for content authors and reviews and adds users to groups.
Content author logs into the site.
Administrator grants content author and content reviewer groups access to pages, virtual resources, and libraries.
Content author wants to create a draft landing page with new landing page template. Content author needs access to edit the page.
Content author sends content for review.
Administrator's next steps:
Administrator creates a 3 stage workflow and assigns access per stage to content author and reviewer groups. This worklfow is associated with the new page template.
Content reviewer approves draft content. Content moves to Publish stage of workflow.

Figure 4-3 Content authors and reviewers goals and the administrator' task to enable those goals

4.2.7 Creating reusable assets

You use reusable assets to store or generate content that is used in more than one place on your website. Reusable assets include the following items:

- Page templates

  Content authors use the page templates to create quickly pages that are consistent with your site design. They do not have to waste time to configure settings that are probably consistent across your site, such as theme selection and page layout. The page templates are also tied to theme and layout.

  For more information about page templates, see 4.12, “Pages and page templates” on page 114.

- Authoring templates

  Authoring templates are like forms that content authors can use to create content. The default settings are defined for the items that are created by using the authoring template. There are two types of authoring templates

  - Site area
  - Content

  For more information about authoring templates, see 4.7, “Authoring templates” on page 85.

- Presentation templates

  You use a presentation template to define the layout of your web content. You use tags to determine which properties, elements, or components are displayed.

  For more information about presentation templates, see 4.13, “Presentation templates” on page 115.

- Template mappings

  Template mappings are used to determine which presentation templates are used to display each site area or content item.

- Content items

  Content items are created from authoring templates. A single content item can be used one time on the website or it can be reused in different areas of the website.

  For more information about content items, see 4.11, “Web Content Manager” on page 113.

- Components

  You use components to store elements that are used in more than one area of your website, for example, a company logo or a copyright notice.

  For more information about components, see 4.9, “Components” on page 107.

4.2.8 Authoring tools

There are multiple ways to create and manage content, such as by using the site toolbar, inline editing, and the authoring portlet.

Site toolbar

The site toolbar is useful for managing and editing content regularly. It provides convenient access to many capabilities directly from the website. Content authors can create pages, add content and application to pages, and manage page parameters.

Figure 4-4 shows an overview of the site toolbar.

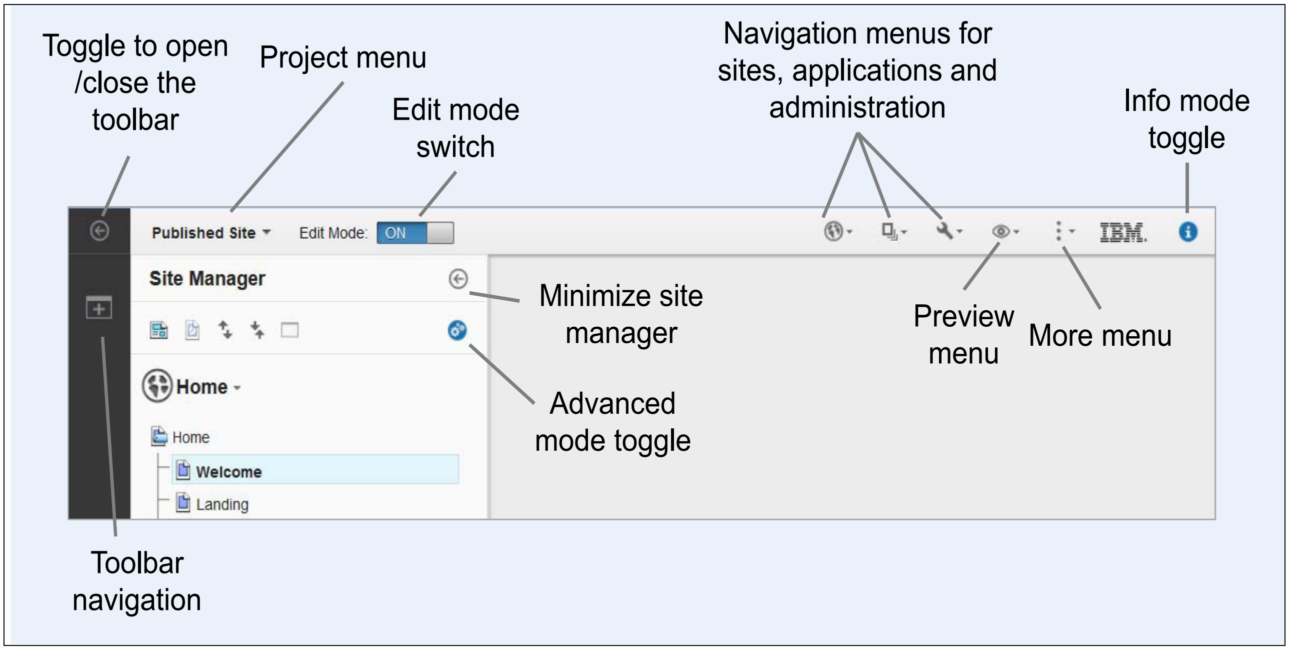


Figure 4-4 WebSphere Portal site toolbar

The site toolbar also provides the application menu (Figure 4-5) for easy and quick access to other sections of the portal site to perform administration and authoring tasks.

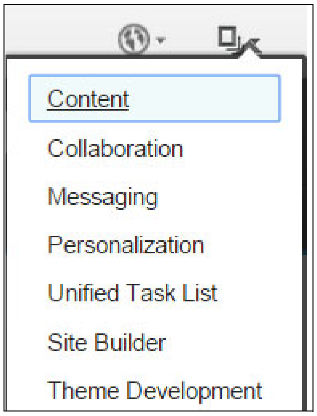


Figure 4-5 Application menu

Web Content inline editing strategies

You can use an inline editing system to deliver editable websites, such as an intranet or a wiki. It combines the features of both an authoring system and a delivery system.

Inline editing

Inline editing provides users with edit access to a content item to edit that item from within the web page itself instead of using the authoring portlet. This feature is available when you display content with a web content viewer portlet.

Creating an authoring tools element

The authoring tool element is used to add authoring portlet functions to web pages. When you create an authoring tool element, you must define the layout of the authoring tool and any required actions, and select parameters for each action layout as required.

Authoring portlet

The authoring portlet is another interface that can be used by content authors, site administrators, and site developers. From the authoring portlet, content authors use forms, called *authoring templates*, to add content to the website. You can use the authoring portlet to customize the Web Content Manager user interface to simplify the content authoring process for your content creators.

Custom portal pages for authoring

You do not have to use the default Web Content Manager page to create content. You can create a portal page to act as the home page of your authoring system.

Authoring system access strategies

The roles that you assign each library on your authoring system determines what views and features in an authoring portlet are accessible to your users.

Grant each user or group access to roles and item types to match the kind of work they perform. For example:

- Assign website designers editor access to authoring templates and presentation templates because they are required to create authoring templates.
- Assign website designers and web content authors editor access to components if they are both required to create components.
- Content approvers are assigned only contributor access to content because they are not required to create content items, but need approver access to content items during a workflow.

Custom authoring interfaces

You can use the Web Content Manager API and remote action parameters to create customized authoring interfaces specifically for your content creators.

You might not want to use an authoring portlet as the user interface for all your users. In some cases, it might be better to create a custom authoring interface by using the Web Content Manager API and remote action parameters. For example, you can create a simple content authoring interface for a specific content authoring team.

Here are some other techniques that you can use to customize your authoring interfaces:

- Custom launch pages

  You can configure an authoring portlet to use a launch page of your own design instead of the default user interface. A custom launch page can either be a JSP or HTML file. You use remote actions to call different views and functions from the authoring portlet user interface.

  You also can use the web content API to add other functions to your launch page. After you create a custom launch page, you then configure your authoring portlet to use the custom launch page instead of the default authoring portlet user interface.

- Remote actions

  Remote actions are used in the query string of a URL to trigger actions from the Web Content Manager application. You can use remote actions to add standard Web Content Manager functions to a custom user interface.

Accessing the authoring portlet

To access the authoring portlet, log in to WebSphere Portal, click the Applications menu icon in the toolbar, and select **Content**, as shown in Figure 4-5 on page 70. The Web Content Authoring portlet is displayed, as shown in Figure 4-6.

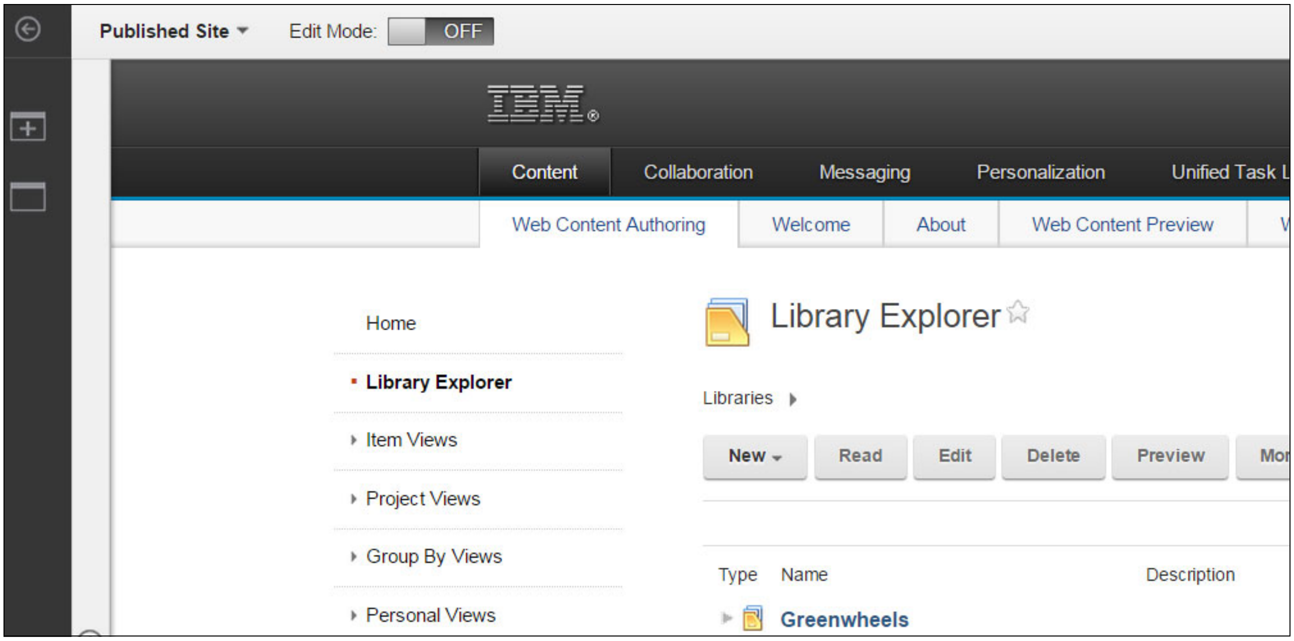


Figure 4-6 Web Content Authoring portlet

4.3 Site toolbar and site manager

The site toolbar and site manager provide quick access to common content authoring tools and a complete view of the navigation of your site. Without leaving the context of your website, content authors can create and modify pages and content items, create vanity URLs for promotions, and much more.

Figure 4-7 shows the site toolbar in a portal page and how to access the site manager.

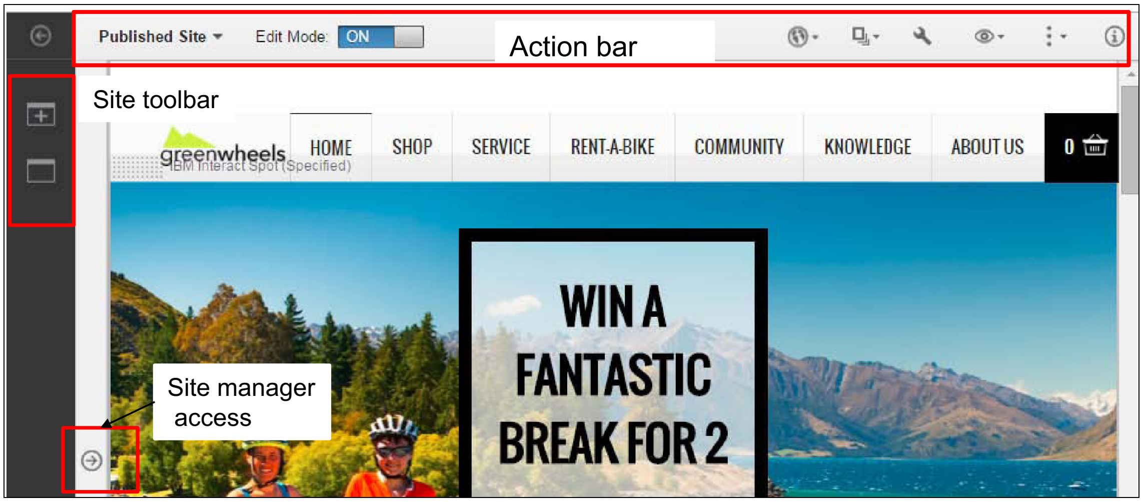


Figure 4-7 Site toolbar and site manager

Action bar

Figure 4-8 shows the action bar, which functions as the entry point into the site toolbar and site manager. The action bar is also where content authors can enable editing and information modes. Additionally, you can preview your site, access projects, and more.

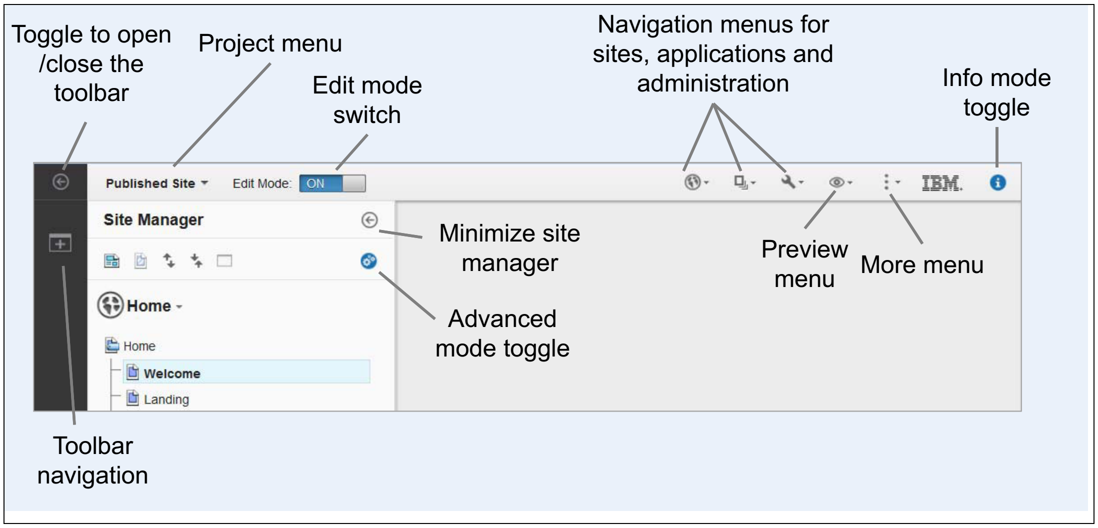


Figure 4-8 Action bar

Edit mode

Edit mode must be on to edit pages and content items in the website. When you turn on edit mode, the site toolbar and site manager open automatically.

Information mode

If you are unfamiliar with the site toolbar, turn on information mode. Information mode controls the user assistance in the user interface. When information mode is on, inline user assistance and hover help is available for many controls. After you become familiar with the site toolbar, you can turn off information mode.

Projects

The Projects menu group content and page edits across your website to make managing and publishing related changes easier. From the Projects menu, you can create a project or manage existing projects.

> **Note:** If you are not working in a project, then the project indicator reads *Published Site*, which means that you are editing the live site. If you are working in a project, then the project indicator displays the project name.

Preview menu

Use the Preview menu to make sure that your changes display as intended. You can preview your site as an authenticated user or as an unauthenticated site visitor.

Site toolbar and site manager access

Use this toggle icon ⬅ to open and close the site toolbar. By default, the toolbar provides access to page components and applications to add to your page, and a main point of entry into the online help. Your administrator can configure the toolbar with additional tabs as needed.

By default, opening the site toolbar also opens site manager, where you can view the hierarchy of all of the pages in your site and create and edit pages and content items. You can also delete, copy and paste, move, and rename pages and content items from within site manager.

Site manager

The site manager has optimized features for content authors. If you are a content author who is responsible for creating and editing content items and pages, expect to spend most of your time working with the site manager.

The site manager provides content authors and administrators with a comprehensive view of a website hierarchy and navigation. In the site manager tree view, you can view the pages and content items that make up the website. You can also view hidden items and associated site areas in the site manager tree view. To view a page in your website, click the name of that page in site manager. The page loads in the page view frame.

The site manager tree view provides more than a central user interface for managing pages and content items. You can also complete common actions, such as creating, editing, and deleting pages and content items. You can also move, copy, and paste pages and content items from the site manager tree view.

> **Note:** The menu displays different options depending on whether you are viewing the menu for a page or a content item. When you create a content item from the menu of a page, that content item is added to the site area that is associated with the parent page. This site area can either be the portal page site area within the portal site library or a different site area that is mapped to the page.

Figure 4-9 shows an example view of a website in the site manager.

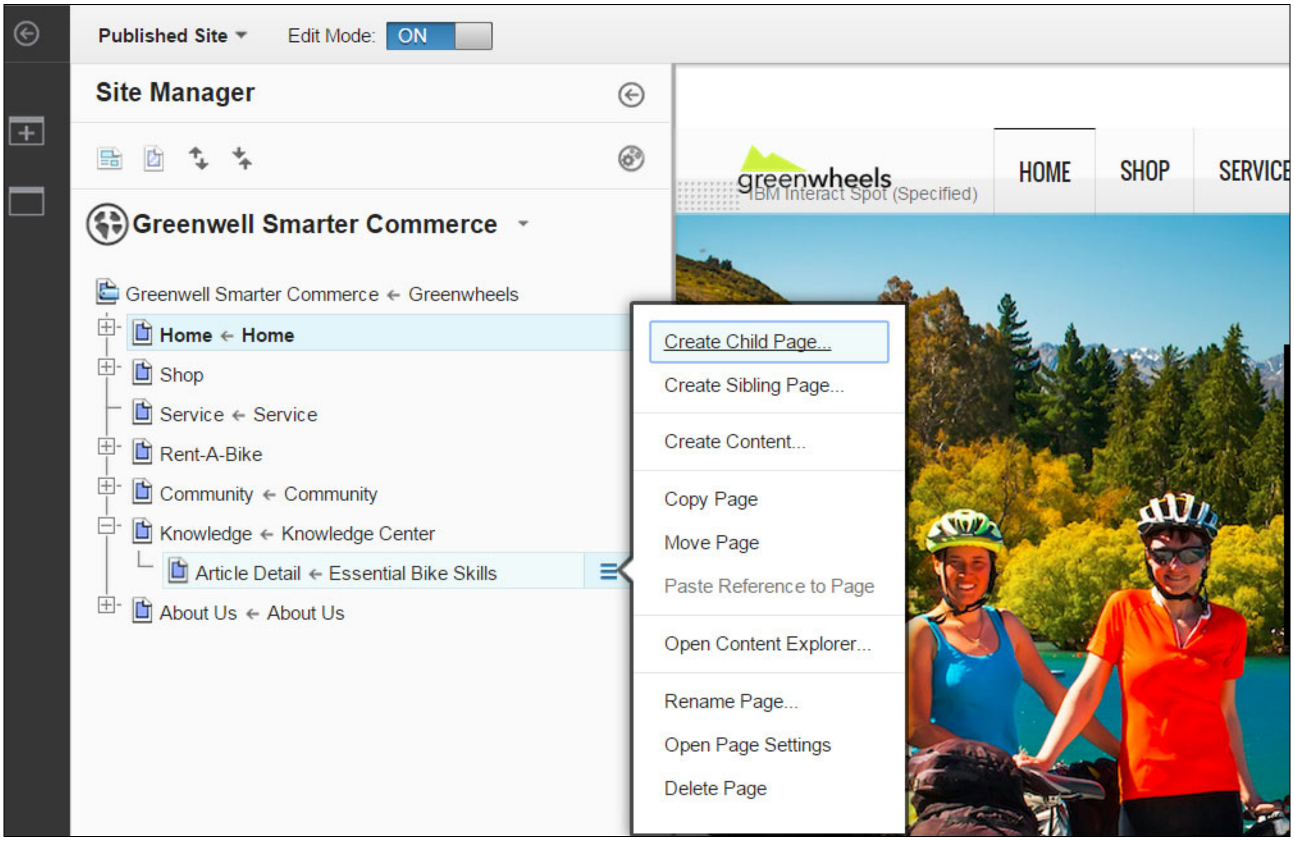


Figure 4-9 Example of a site manager view

4.4 Workflow management

You can manage changes to web content items and managed pages either by creating drafts, by using workflows, or by adding items to projects.

There are three major status levels that an item can be in at any one time: Draft, Published, or Expired. The current state of an item indicates where the item exists within a change management process, and where that item can be viewed and accessed.

You use workflows to control the access to, verification, and eventual approval of items. Only if an item is approved at all stages up to a published stage can it be viewed on your website.

Adding a workflow to managed pages: The default workflow that is used for published pages has only a draft and approved state. However, if the default workflow is not sufficient, you can define custom workflows and use them with published pages.

A workflow must have at least one stage, but typically has more, and it always flows in a linear pattern. The following examples are ways that you can use a workflow:

- Review the accuracy of content.
- Review content for legal implications.
- Review content to ensure that it meets accessibility guidelines.
- Ensure that no malicious code such as cross-scripting attacks are added to content.

A reject stage can be specified, which is a stage that is run when a document is declined, before moving it to the first stage of the workflow. If the item is rejected at any stage, someone with editor access must correct or amend the item and resubmit it into the selected workflow for approval. All items that are rejected (regardless of the stage they are at in the approval process) are sent back to the first (creation) stage of the workflow.

You can also specify that a comment must be entered on every move a document makes in the workflow or only on specific stages. This comment is added to the document's history section.

Another important feature is that workflows can ensure that you create drafts of items and do not directly edit the published version. Typically in the development process, you normally do not want content authors to edit the published version.

4.4.1 Workflow stages

Workflow stages are the building blocks of a workflow. You must create at least one stage before you can create a workflow.

Workflow stages determine the following things:

- What actions to run when an item enters or exits a workflow stage
- The access levels of users or groups within that stage

In most cases, actions are run when an item enters a stage. For example, you add a scheduled move action to run when entering a stage so that it is enabled when an item enters that stage.

However, if you set a scheduled move action to run when leaving a stage, it never runs. The most common type of actions to run when leaving a stage are email actions, when you want to notify users that an item exited a workflow stage.

4.4.2 Workflow actions

Each workflow stage contains a set of actions that include actions that are run when an item enters the stage and actions that are run when an item exits the stage.

Note: Exit actions are restricted to non-scheduled actions because they must be run immediately.

Table 4-1 describes the actions that you can use for workflow stages.

Table 4-1 Workflow stage actions

Action	Details
Publish	Changes an item status from Draft to Published. The item is available on the rendered site. An item is published when it enters a workflow stage that contains a publish action and the selected published date and time are reached.
Expire	Changes an item status from Published to Expired. The document is no longer available on the site. An item is expired when it enters a workflow stage that contains an expire action and the selected expire date and time are reached.

Action	Details
Email	This action sends emails when it runs. You can create email actions and specify who the recipients are. You can select email approvers, authors, and owners. You also can create a list of other users or groups to email. A link to the item to be reviewed is included in the email.
Schedule Move	This action performs a scheduled move to the next stage on a specified date. A list-box allows users to select one of four date types that are entered on each individual document, or you can specify a static date.
Version	This action creates a version of an item when it runs.
Custom	You also can create custom workflow actions by creating a custom workflow plug-in. These actions can be used and scheduled within a workflow like other workflow actions.

For a sample workflow, see the *Workflow example* topic in the IBM WebSphere Portal V8.5.0 IBM Knowledge Center:

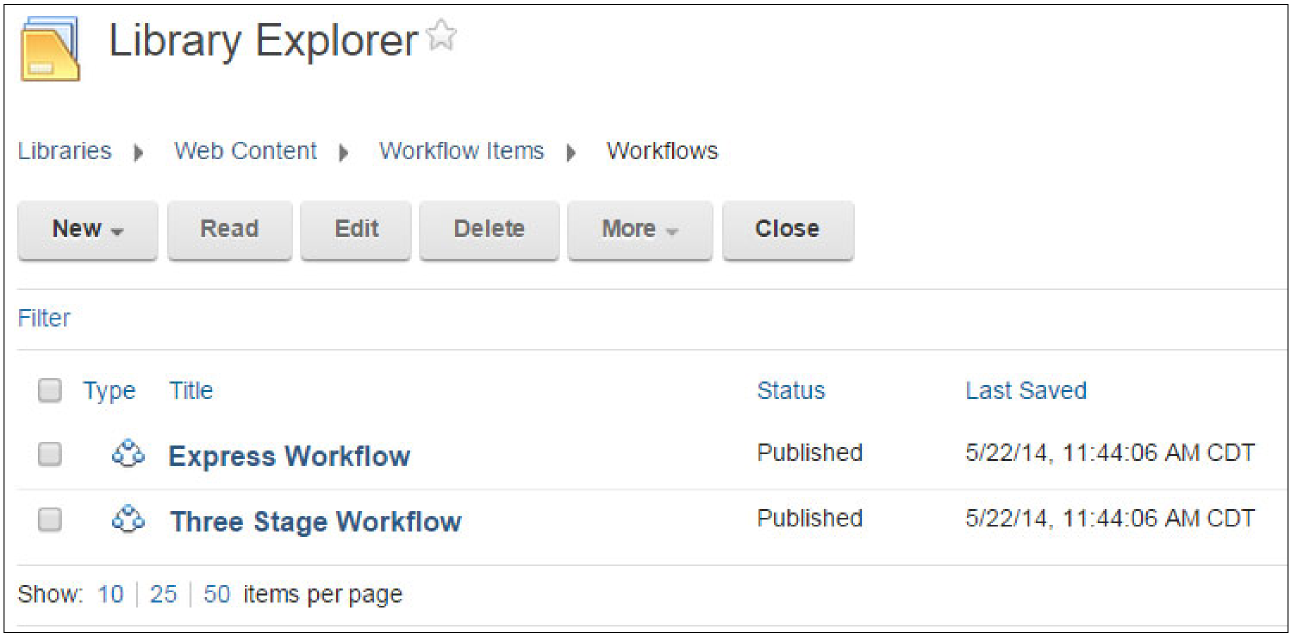
http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/wcm/wcm_dev_workflows_example.dita

Joint approval

Joint approval is used in cases where approval from multiple users is required before an item is moved to the next stage. You specify which stages you want to be jointly approved. If joint approval is active, then all the approvers that are specified for this stage must approve the document. The exception is an administrator, who can force a document to the next stage.

Default workflow and workflow actions

When you create a Web Content Manager library with default items in the new library, the workflows Express Workflow and Three Stage Workflow are included to help you get started faster, as shown in Figure 4-10.



□	Type	Title	Status	Last Saved
□		Express Workflow	Published	5/22/14, 11:44:06 AM CDT
□		Three Stage Workflow	Published	5/22/14, 11:44:06 AM CDT

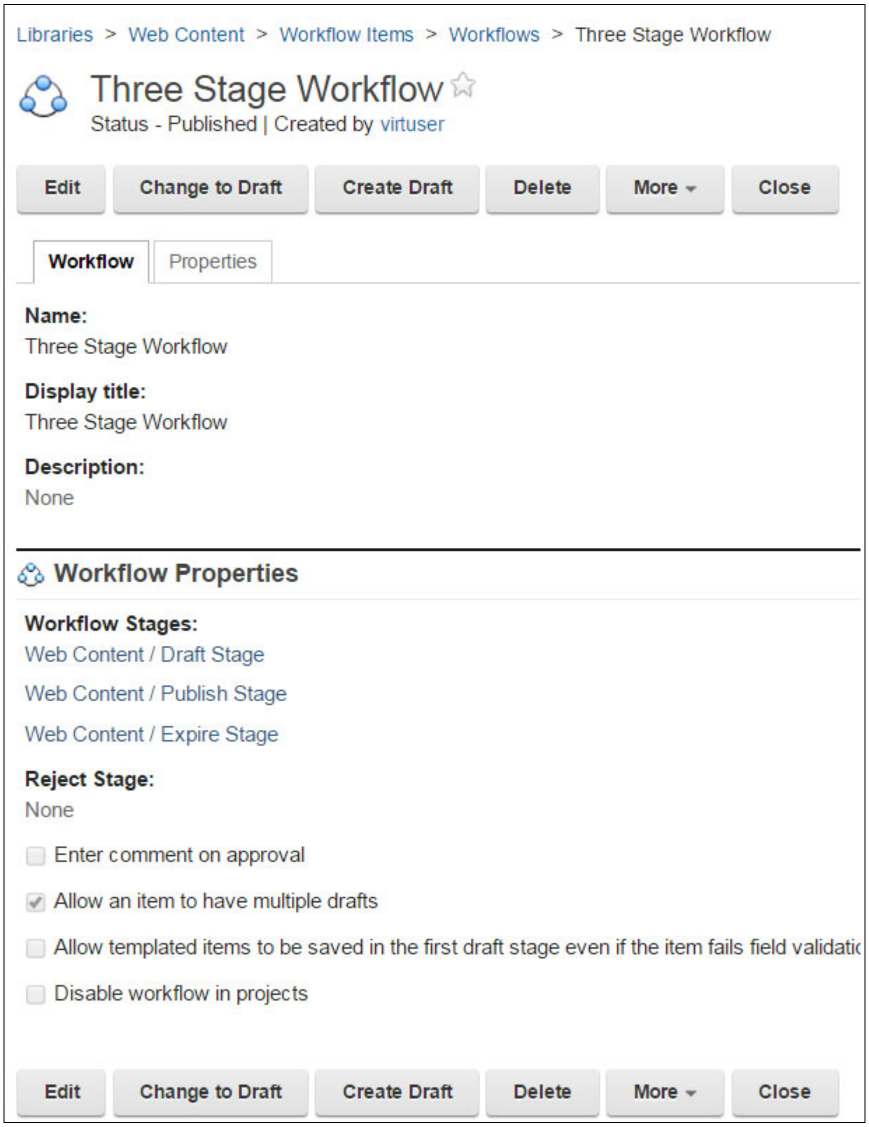
Figure 4-10 Default ready-to-use workflows

Express workflow

The Express workflow is a simple workflow with just one Publish stage. This workflow has the publish workflow action triggered when entering this stage.

Three Stage workflow

The Three Stage workflow contains Draft, Publish, and Expire stages. The Web Content / Publish action is run when entering the Publish stage. The Web Content / Expire action is run when entering the Expire stage (Figure 4-11).



Libraries > Web Content > Workflow Items > Workflows > Three Stage Workflow

Three Stage Workflow

Status - Published | Created by virtuser

Name:
Three Stage Workflow

Display title:
Three Stage Workflow

Description:
None

Workflow Properties

Workflow Stages:
Web Content / Draft Stage
Web Content / Publish Stage
Web Content / Expire Stage

Reject Stage:
None

☐ Enter comment on approval

☑ Allow an item to have multiple drafts

☐ Allow templated items to be saved in the first draft stage even if the item fails field validatic

☐ Disable workflow in projects

Figure 4-11 Three Stage default workflow

For step-by-step instructions about adding workflows to managed pages, see the *Adding workflow to managed pages* topic in the IBM WebSphere Portal V.5.0 IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/wcm/wcm_mngpages_addworkflow.dita

4.5 Projects and project templates

You can use Web Content Manager projects to create the site and to coordinate multiple updates to the site without impacting the published site.

Tip: Although you can update the site without creating a project, it is a preferred practice to use projects to simplify the publishing process.

4.5.1 Projects

Projects help you organize and coordinate changes to your site. You can explore improvements and receive feedback and input from colleagues during the review process and publish the changes to your website at the same time to simplify the publication process. The changes or editions to the website can be grouped and published with a single operation, rather than publishing each website item one by one.

Projects provide the content owner and the content author the flexibility that is needed to streamline the creation, review, and publication process.

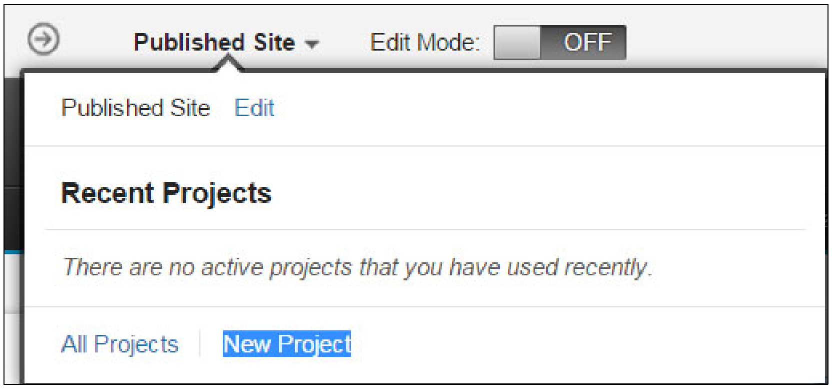
Changes to published sites involve more than a single update and might span multiple site areas. With projects, content authors can focus on what they do best, creating or updating content based on input from others. Content owners can focus on managing the flow of the content in projects with project templates. A project template defines the publish method, workflow, and approval options.

Creating a project

This section describes how to create and use projects for site updates.

To create a project, complete the following steps:

1. Click **Published Site** → **New Project**, as shown in Figure 4-12.



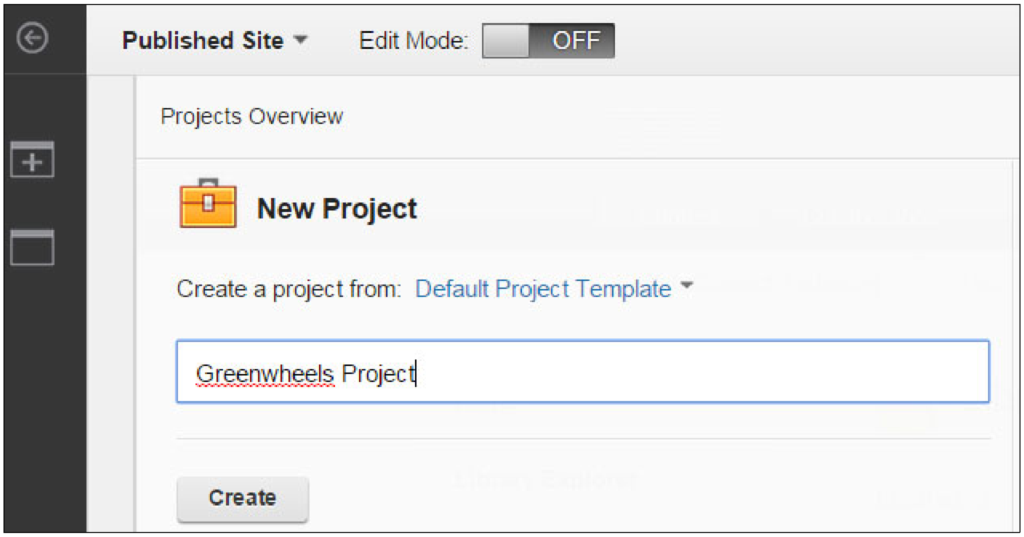
There are no active projects that you have used recently.

Figure 4-12 Create a project

2. In the New Project section, enter the following information:

   – Select a template from the **Create a project from** menu to create a project. By default, the Default Project Template is selected.

   – Enter a name for the project in the field. By default, you see your user ID and the date as a sample project name. In the example that is shown in Figure 4-13, the project name is Greenwheels Project.



Create a project from: Default Project Template

Greenwheels Project

Figure 4-13 Enter a project name

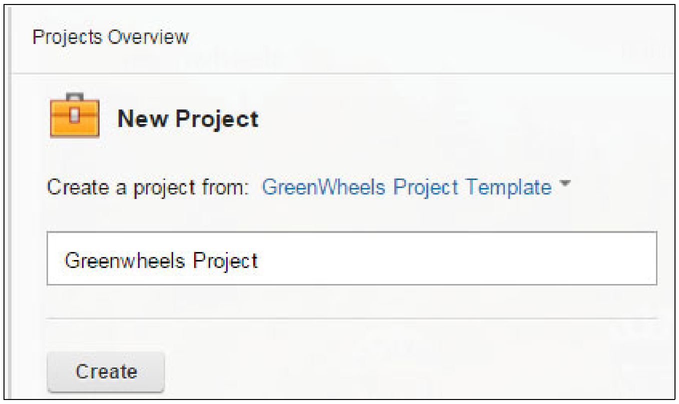
3. Click **Create**.

4.5.2 Project template

By default, a project is created based on the Default Project Template. You can select a specific template to create a project. If you do not see the project template that you need, create a project template to add to the “Create a project from” menu.

Note: You can also change the default project template per virtual portal since WebSphere Portal V8.5 Fix Pack 07 and later in the Web Content Manager Authoring user interface.

Figure 4-14 shows an example of a project that is created based on a GreenWheels Project Template.



Projects Overview

New Project

Create a project from: GreenWheels Project Template

Greenwheels Project

Figure 4-14 Create a project by using a custom project template

Project templates are the foundation of your projects. As the content owner, you use the project template to set up an initial flow and structure to your project. To create a project template, you must first create a project. After you create a project and define project logistics, you can save the project as a project template for others to use.

Project templates allow the content owner to prevent the content creator from modifying the resulting project.

The three main sections of a project template (Publish Option, Approval, and Custom Action) include the following options:

- Hide section check box
- Editors: Add Editors selector to add editors
- Viewers: Add Viewers selector to add viewers

After a content owner defines the Publish Option, Approval, and Custom Action sections of a project template, they can control the content authors that can view and edit the section. If the Hide section check box is selected, only the users who are listed as editors or viewers have access to the section. If no users are listed, then only users with the Manager role and above have access to this section.

If the Hide section check box is not selected, but there are users who are selected in either or both the editor and viewer selections, only the users that are selected for the roles have access. Users with manager and above access levels always have selection access.

Projects that are created from a project template do not show the controls to hide the section, but they do inherit the controls from the project template. Users with the Manager role and above can see the controls for a project that is created from a project template, but they must click **Show Hidden Fields** to do so.

For authoring templates, you can update the template and reapply it to existing content. However, you cannot update a project template and reapply it to an existing project. This function is not supported because of the relatively short lifecycle of the project.

The controls on the project template are shown to users with manager access and above by default, but require the use of the Show Hidden Fields button for control of the project.

For more information about project templates, see the *Creating a project template* topic in the IBM WebSphere Portal V8.5.0 IBM Knowledge Center:

https://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/help/panel_help/toolbar_project_create_template.dita

4.6 Content library

Content libraries store the assets for your website, such as pages, content, images, authoring and presentation templates, and workflows.

All pages are stored in a Portal Site library. The Portal Site library is immediately available and includes the welcome page, getting started page, and more. You can create more libraries to store your website content and assets. You can also create multiple libraries to organize your content or to reflect your company's structure. In most cases, you must create a minimum of two libraries:

- A design library where you store all the items that are required for the web content system itself.
- A content library that is used to store the content that is developed by your content creators.

To create a library, complete the following steps. This library contains all the content, components, authoring templates, presentation templates, and so on that are used in the website.

1. Log in to WebSphere Portal and click the **Open portal administration** icon, as shown in Figure 4-15.

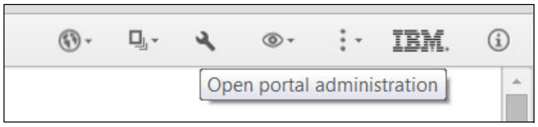
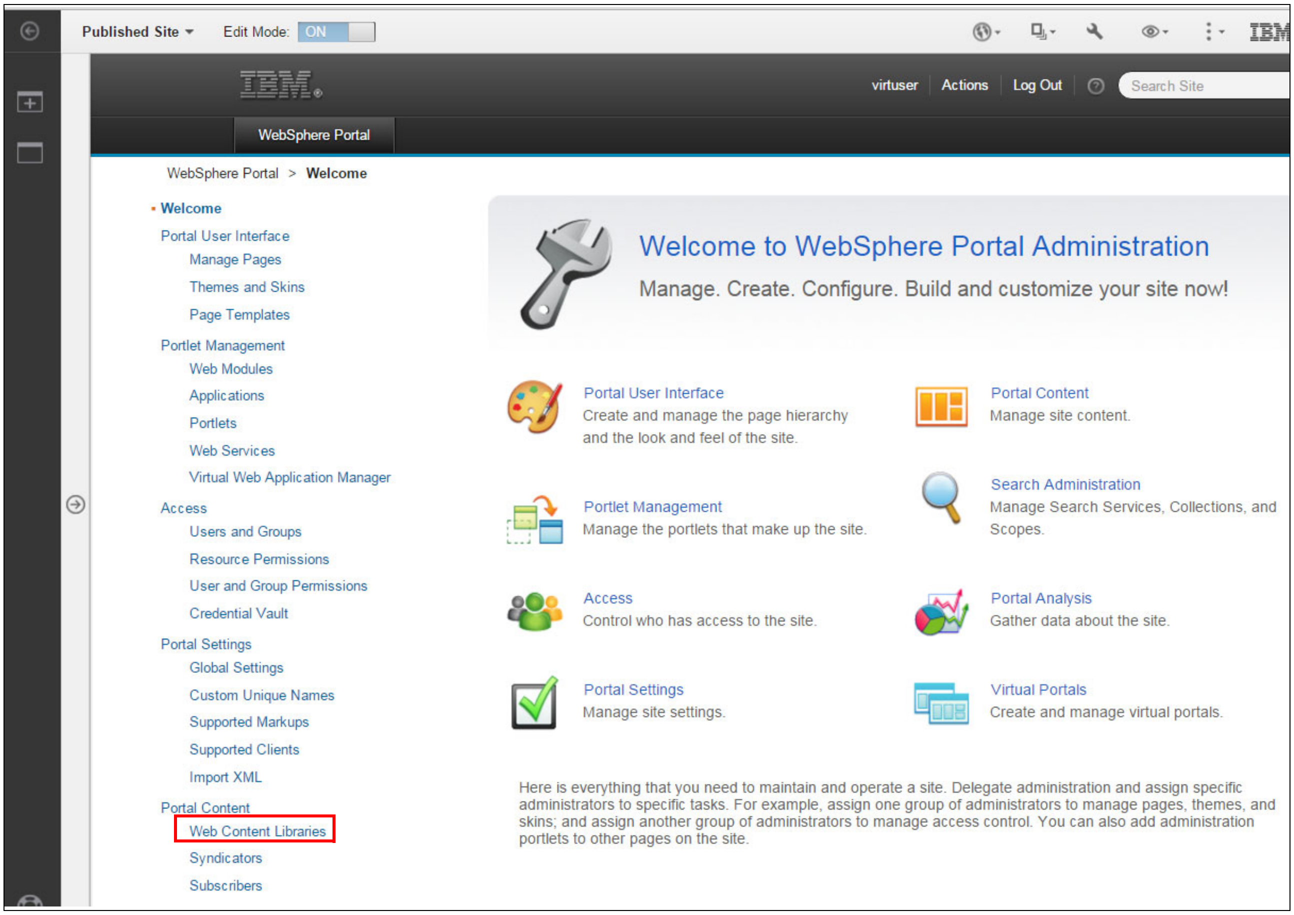


Figure 4-15 Open the WebSphere Portal administration

2. In the WebSphere Portal Administration window, click **Web Content Libraries**, as shown in Figure 4-16.



Welcome to WebSphere Portal Administration

Manage. Create. Configure. Build and customize your site now!

Portal User Interface
Create and manage the page hierarchy and the look and feel of the site.

Portal Content
Manage site content.

Portlet Management
Manage the portlets that make up the site.

Search Administration
Manage Search Services, Collections, and Scopes.

Access
Control who has access to the site.

Portal Analysis
Gather data about the site.

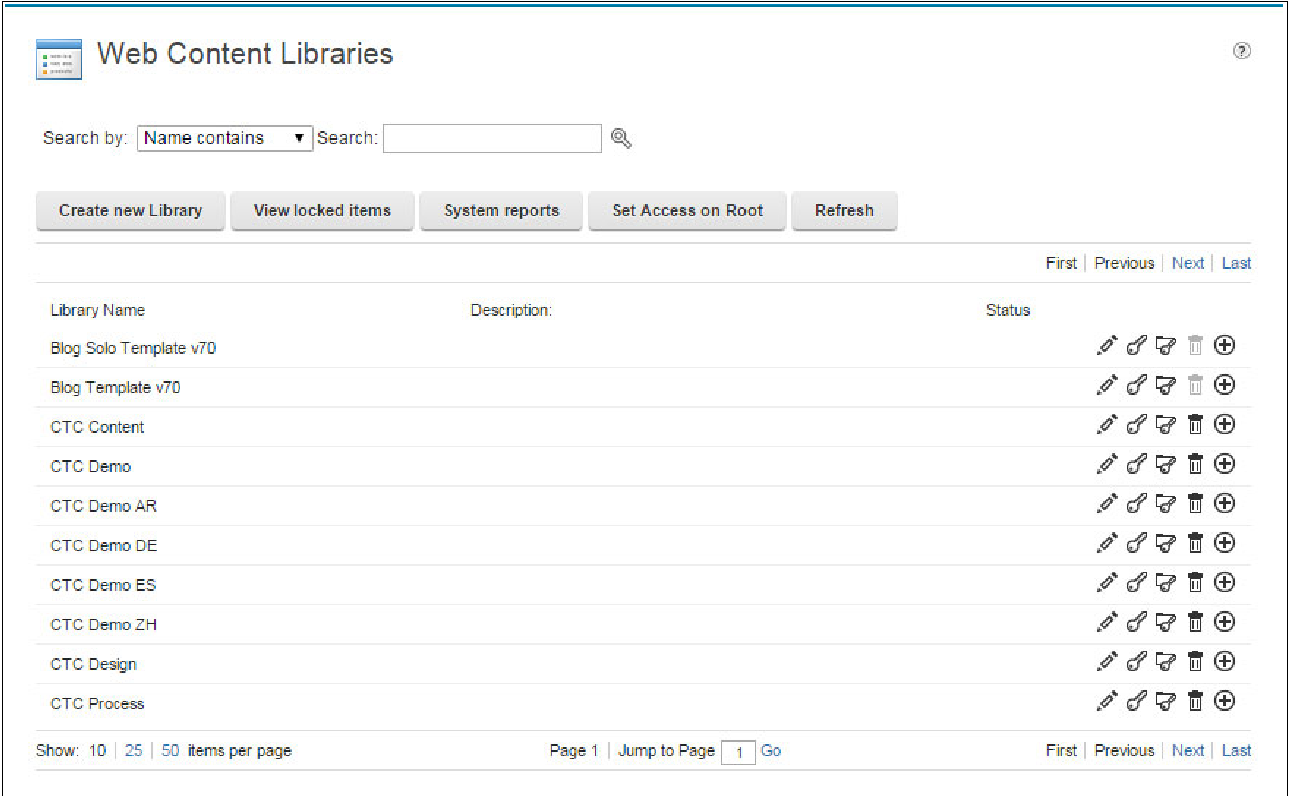
Portal Settings
Manage site settings.

Virtual Portals
Create and manage virtual portals.

Here is everything that you need to maintain and operate a site. Delegate administration and assign specific administrators to specific tasks. For example, assign one group of administrators to manage pages, themes, and skins; and assign another group of administrators to manage access control. You can also add administration portlets to other pages on the site.

Figure 4-16 Access the Web Content library

3. In the Web Content Libraries portlet, click **Create New Library**, as shown in Figure 4-17.



Web Content Libraries

Search by: Name contains ▾ Search:

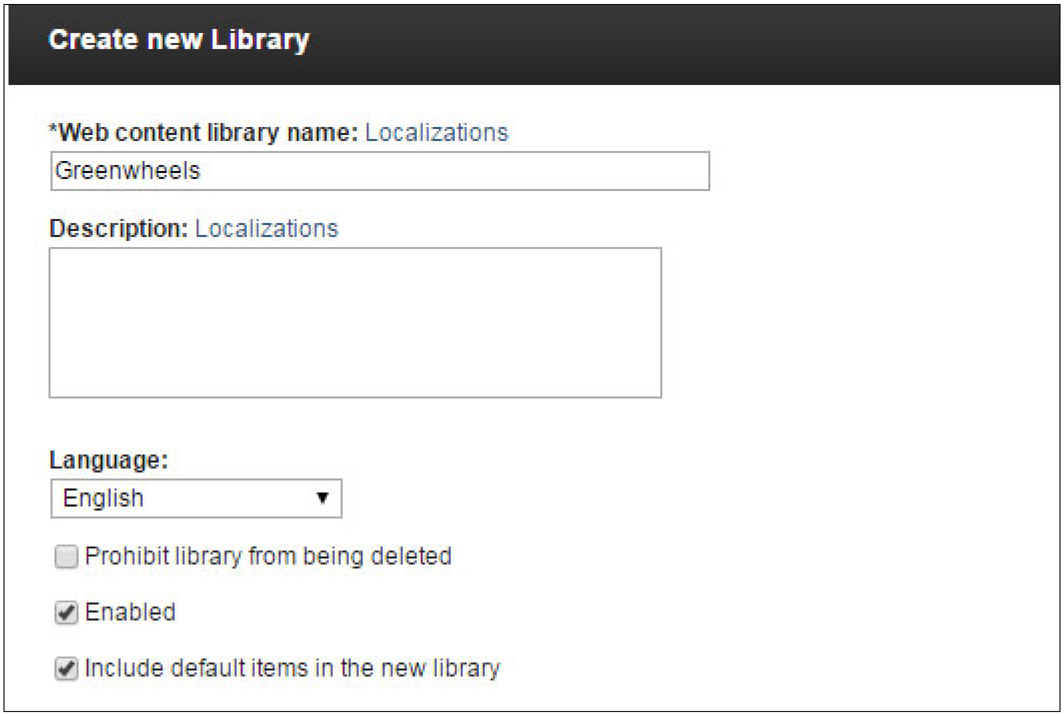
Create new Library
View locked items
System reports
Set Access on Root
Refresh
First | Previous | Next | Last

Library Name	Description:	Status
Blog Solo Template v70		
Blog Template v70		
CTC Content		
CTC Demo		
CTC Demo AR		
CTC Demo DE		
CTC Demo ES		
CTC Demo ZH		
CTC Design		
CTC Process		

Show: 10 | 25 | 50 items per page
Page 1 | Jump to Page 1 Go
First | Previous | Next | Last

Figure 4-17 Web Content Libraries window

4. In the Create new library window, enter a Web Content library name and select the **Include default items in the new library** check box, as shown in Figure 4-18. Click **OK**.



Create new Library

*Web content library name: Localizations

Greenwheels

Description: Localizations

Language:

English ▾

□ Prohibit library from being deleted

☑ Enabled

☑ Include default items in the new library

Figure 4-18 Create a web content library

4.7 Authoring templates

Authoring templates are like forms that content authors can use to create content. There are two types in authoring template:

- Site area
- Content

You can create authoring templates for your content authors to use as they develop content for the website. The authoring template is mapped to a presentation template. You can use one presentation template for multiple authoring templates. This approach removes presentation considerations and treatment from the content. The website design is updated by modifying the presentation template instead of modifying multiple pieces of content.

For more information about authoring templates, see the *An overview of authoring templates* topic in the IBM WebSphere Portal V8.5.0 IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/wcm/wcm_dev_auth-temp_overview.dita

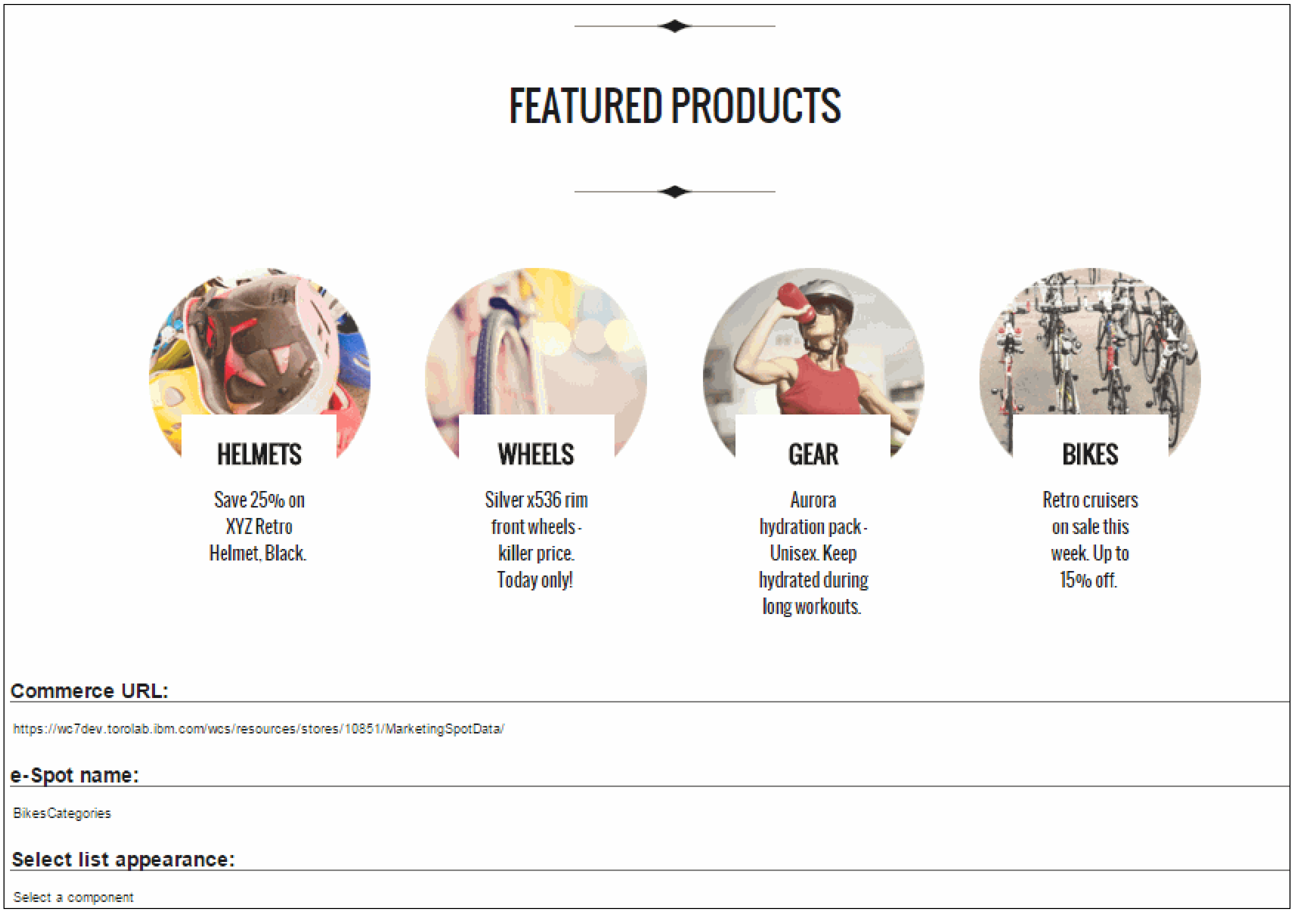
The following topics are covered in this section:

- 4.7.1, “Example: Creating Web Content Manager content” on page 86
- 4.7.2, “Creating simple and complex authoring templates” on page 97

4.7.1 Example: Creating Web Content Manager content

This example in this section shows how to create an authoring template that content creators use to create Web Content Manager content to be rendered on a WebSphere Portal page by a Web Content Viewer portlet. The Web Content Manager retrieves the content elements, such as product ID and image from a WebSphere Commerce store, by using an IBM Digital Data (DDC) plug-in.

Figure 4-19 shows a sample portal home page with the Web Content Viewer portlet rendering the Feature products content that is created by using the authoring template greenwheels.shop.advance, which is created in this example.



FEATURED PRODUCTS

HELMETS

Save 25% on XYZ Retro Helmet, Black.

WHEELS

Silver x536 rim front wheels - killer price. Today only!

GEAR

Aurora hydration pack - Unisex. Keep hydrated during long workouts.

BIKES

Retro cruisers on sale this week. Up to 15% off.

Commerce URL:

https://wc7dev.torolab.ibm.com/wcs/resources/stores/10851/MarketingSpotData/

e-Spot name:

BikesCategories

Select list appearance:

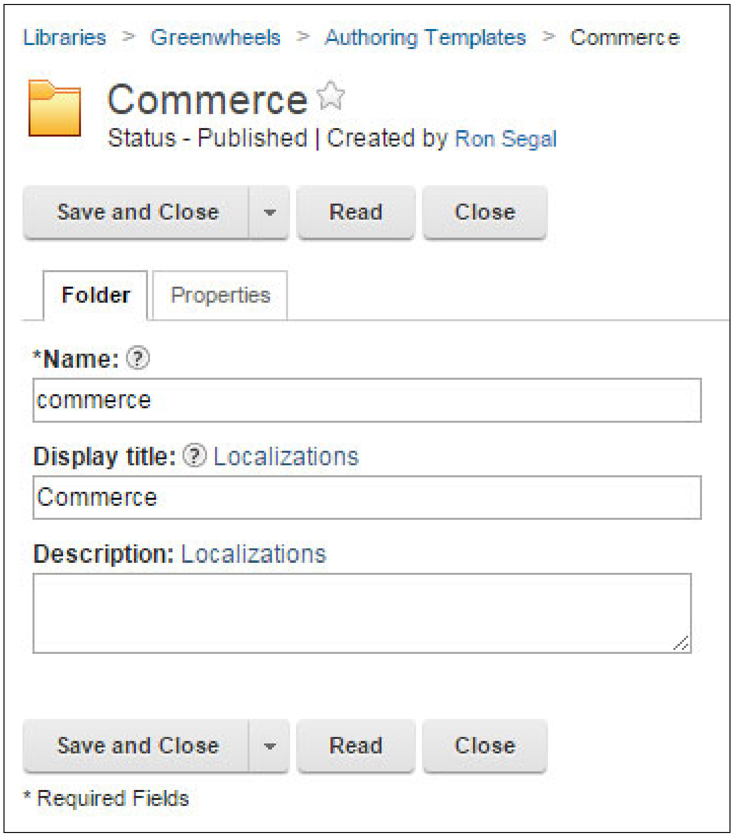
Select a component

Figure 4-19 Sample web page with Web Content Viewer portlet rendering content from WebSphere Commerce

To create the greenwheels.shop.advance authoring template for this example, complete the following steps:

1. Click the **Open applications menu** icon in the toolbar and select **Content**.
2. Select the library where you want to store the authoring template. Greenwheels is used in this example.
3. Click **Authoring Templates**.
4. Click **New** → **Folder**. In this example, the greenwheels.shop.advance authoring template is stored in the Commerce folder in the Greenwheels library.

5. Enter **Commerce** in the Name field, as shown in Figure 4-20. Click **Save and Close**. The Commerce folder is created in the Greenwheels library.



Status - Published | Created by Ron Segal

*Name: commerce

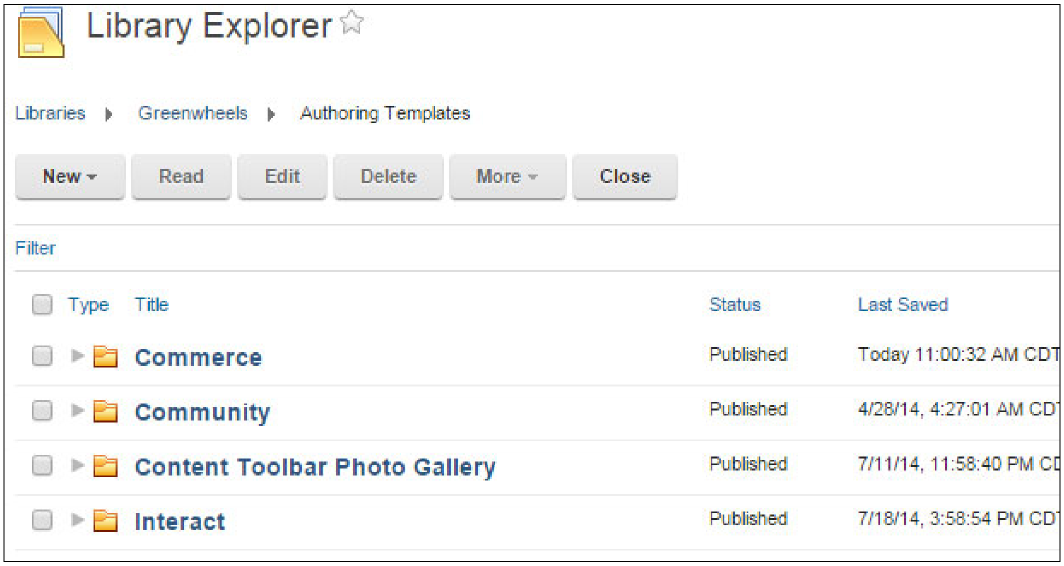
Display title: Commerce

Description:

* Required Fields

Figure 4-20 Create the Commerce folder

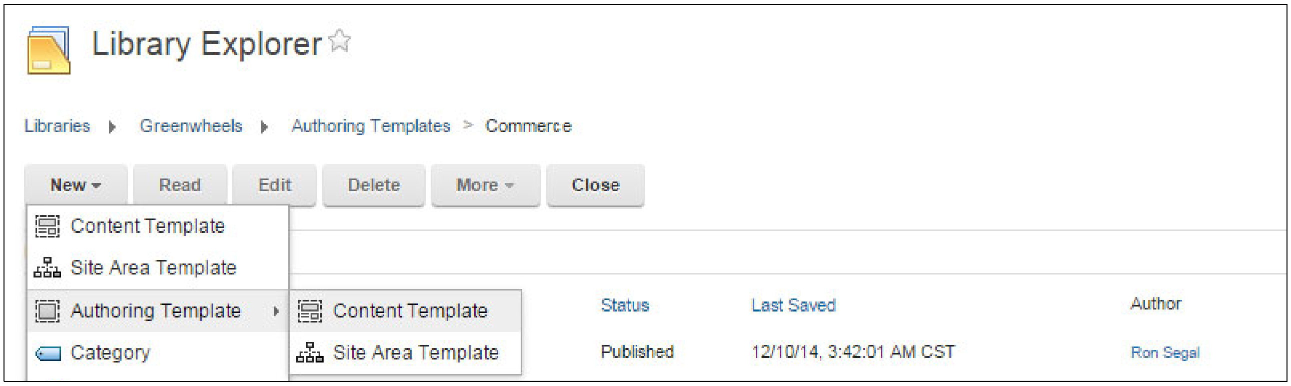
6. In the Library Explorer window, click **Commerce** to open the Commerce folder (Figure 4-21).



Type	Title	Status	Last Saved
▶	Commerce	Published	Today 11:00:32 AM CDT
▶	Community	Published	4/28/14, 4:27:01 AM CD
▶	Content Toolbar Photo Gallery	Published	7/11/14, 11:58:40 PM CD
▶	Interact	Published	7/18/14, 3:58:54 PM CD

Figure 4-21 Open the Commerce folder

7. Click **New** → **Authoring Template** → **Content Template**, as shown in Figure 4-22.



Status	Last Saved	Author
Published	12/10/14, 3:42:01 AM CST	Ron Segal

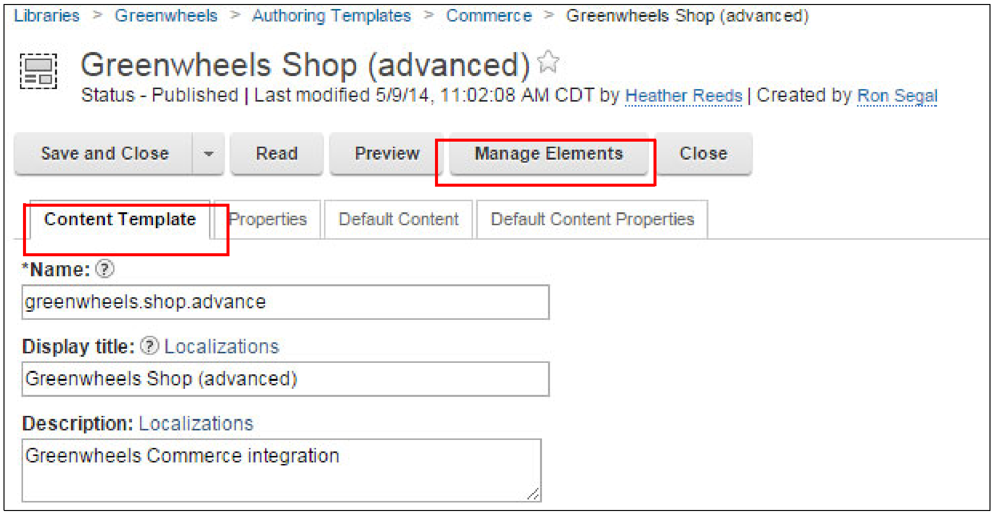
Figure 4-22 Selecting to create an authoring template

8. In the Content Template tab (Figure 4-23), enter the following information:

   – Name: greenwheels.shop.advance

   – Display title: Greenwheels Shop (advanced)

   Click **Manage Elements**.


Libraries > Greenwheels > Authoring Templates > Commerce > Greenwheels Shop (advanced)
Greenwheels Shop (advanced)
Save and Close
Read
Preview
Manage Elements
Close
Content Template
Properties
Default Content
Default Content Properties
Localizations
Localizations

Status - Published | Last modified 5/9/14, 11:02:08 AM CDT by Heather Reeds | Created by Ron Segal

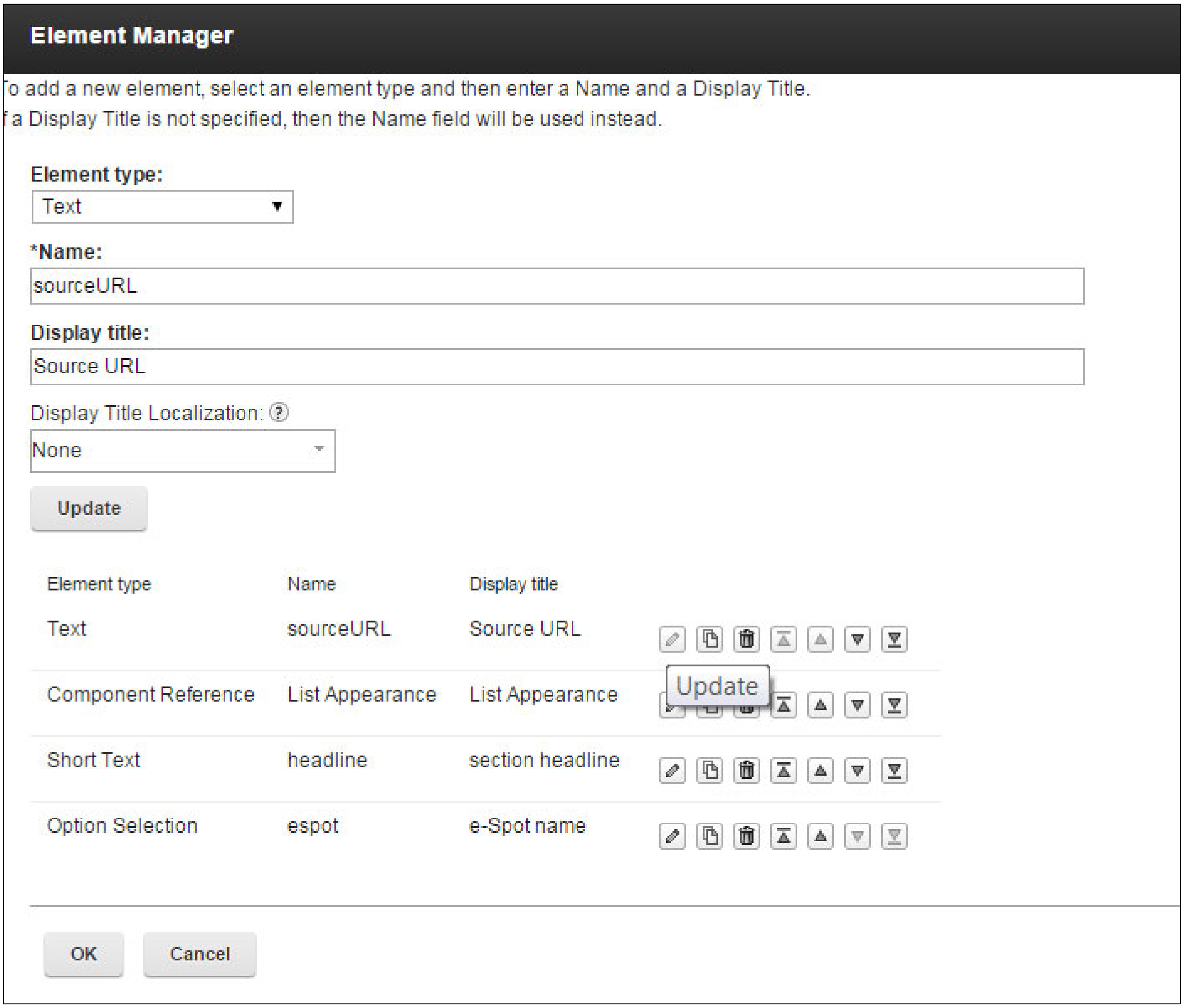
*Name: greenwheels.shop.advance

Display title: Greenwheels Shop (advanced)

Description: Greenwheels Commerce integration

Figure 4-23 Create an authoring template

9. In the Element Manager window, add an element type Text, as shown in Figure 4-24:
   a. Select **Text** from the Element type menu.
   b. Enter a Name (sourceURL in this example).
   c. Enter a Display Title (Source URL in this example).
   d. Click **OK**.



Element Manager

o add a new element, select an element type and then enter a Name and a Display Title.
f a Display Title is not specified, then the Name field will be used instead.

Element type: Text

*Name: sourceURL

Display title: Source URL

Display Title Localization: None

Element type	Name	Display title
Text	sourceURL	Source URL
Component Reference	List Appearance	List Appearance
Short Text	headline	section headline
Option Selection	espot	e-Spot name

Figure 4-24 Add elements to the authoring template

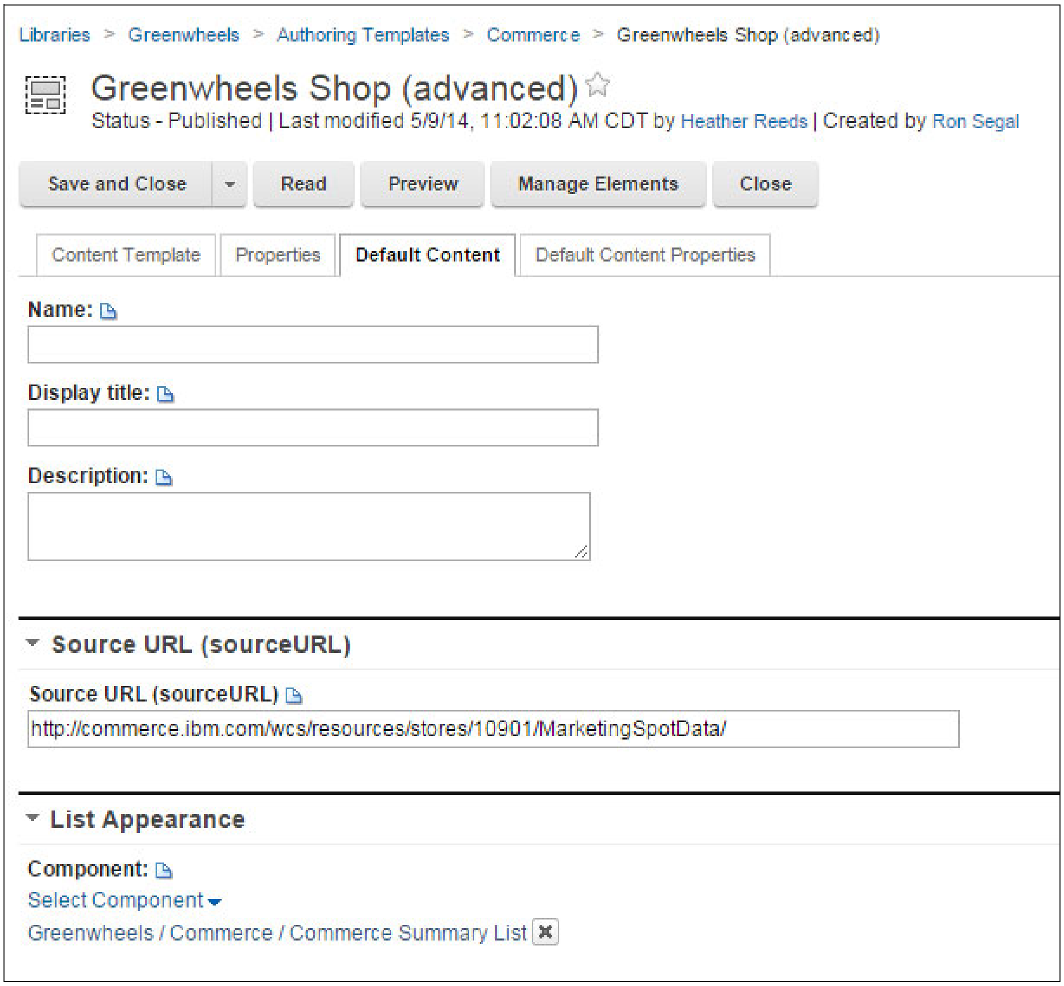
10. Repeat step 9 to add elements of type Component Reference, Short Text, and Option Selection.

11. You can specify default values for each field and element in the authoring template to make it easier and more efficient for an author to create items and streamline the item creation process.

> **Note:** As described in the introduction to this example, this example is a special case where the Web Content Manager content retrieves the content to be rendered by the Web Content Viewer portlet from WebSphere Commerce.

Click the **Default Content** tab (see Figure 4-25). The sourceURL element contains the URL to the WebSphere Commerce server that contains the Greenwheels Store data to be rendered in the portal.

In the List Appearance section, click **Select Component** and then select the appropriate Summary List component. In this example, the **Commerce Summary List** was selected.


Libraries > Greenwheels > Authoring Templates > Commerce > Greenwheels Shop (advanced)

Greenwheels Shop (advanced)

Status - Published | Last modified 5/9/14, 11:02:08 AM CDT by Heather Reeds | Created by Ron Segal

Name:

Display title:

Description:

Source URL (sourceURL)

Source URL (sourceURL)

http://commerce.ibm.com/wcs/resources/stores/10901/MarketingSpotData/

List Appearance

Component:

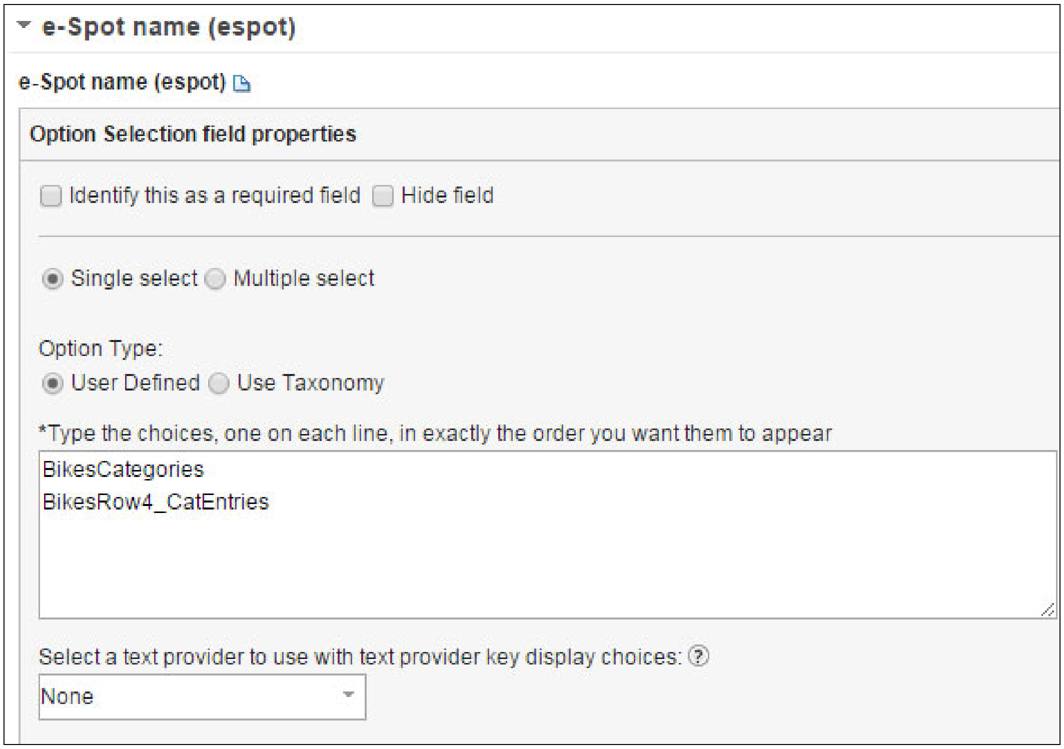
Select Component

Greenwheels / Commerce / Commerce Summary List

Figure 4-25 Specify the source URL and list appearance for WebSphere Commerce data

12. You can use an Option Select element type to provide options to the content author. These options appear in a list. For this Commerce example, the values that are selected here filter the results that are retrieved from the WebSphere Commerce Server. Enter the default values for Option Selection element eSpots, as shown in Figure 4-26. Click **Save and Close**.

    Now, content authors can use this authoring template to create the content.



▾ e-Spot name (espot)

e-Spot name (espot)

Option Selection field properties

□ Identify this as a required field □ Hide field

● Single select ○ Multiple select

Option Type:

● User Defined ○ Use Taxonomy

*Type the choices, one on each line, in exactly the order you want them to appear

```
BikesCategories
BikesRow4_CatEntries
```

Select a text provider to use with text provider key display choices: ?

None

Figure 4-26 Select the Option Selection element properties

Now that the authoring template is created, you can use that template to create content items. To create the Web Content Manager content, Featured Content, complete the following steps:

1. Click the **Open applications menu** icon in the toolbar and select **Content**.
2. Select the library where you want to store the content, which is Greenwheels in this example.

3. Click **New** → **Content** → **Greenwheels Shop (advanced)**, as shown in Figure 4-27.

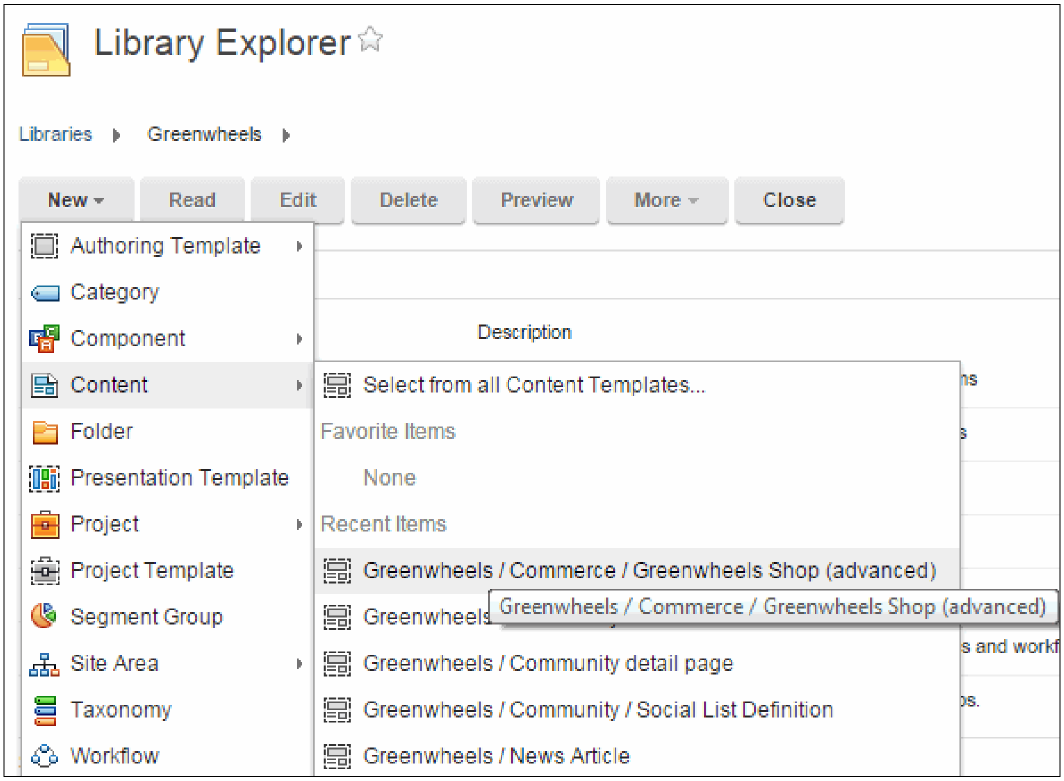
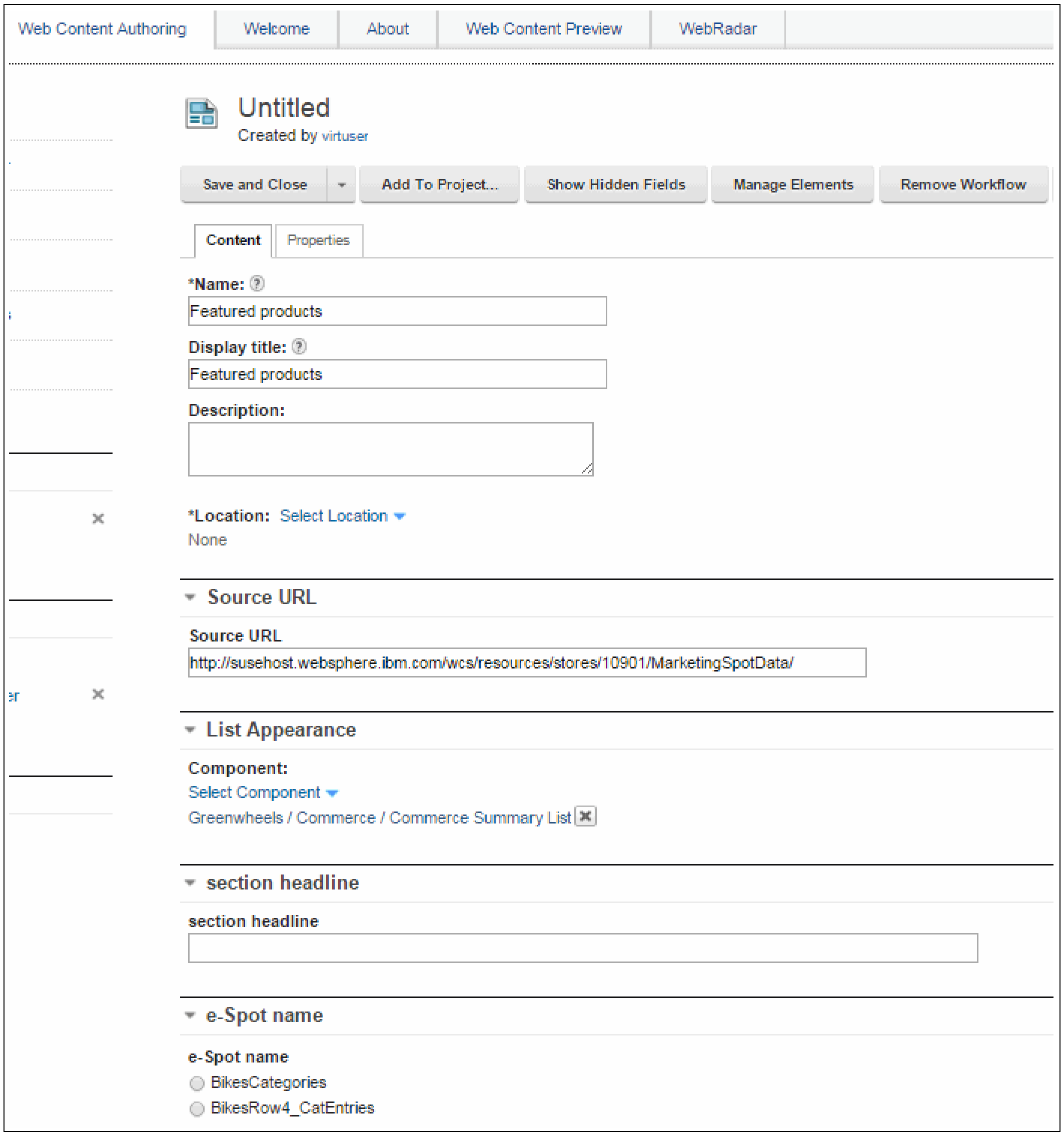


Figure 4-27 Create Featured Products content based on Greenwheels Shop (advanced) (1 of 2)

4. Complete the form, as shown in Figure 4-28. For more information about the components, see 4.9, “Components” on page 107. Click **Save and Close**.



*Name: Featured products

Display title: Featured products

Description:

*Location: Select Location
None

Source URL

Source URL: http://susehost.websphere.ibm.com/wcs/resources/stores/10901/MarketingSpotData/

List Appearance

Component: Select Component
Greenwheels / Commerce / Commerce Summary List

section headline

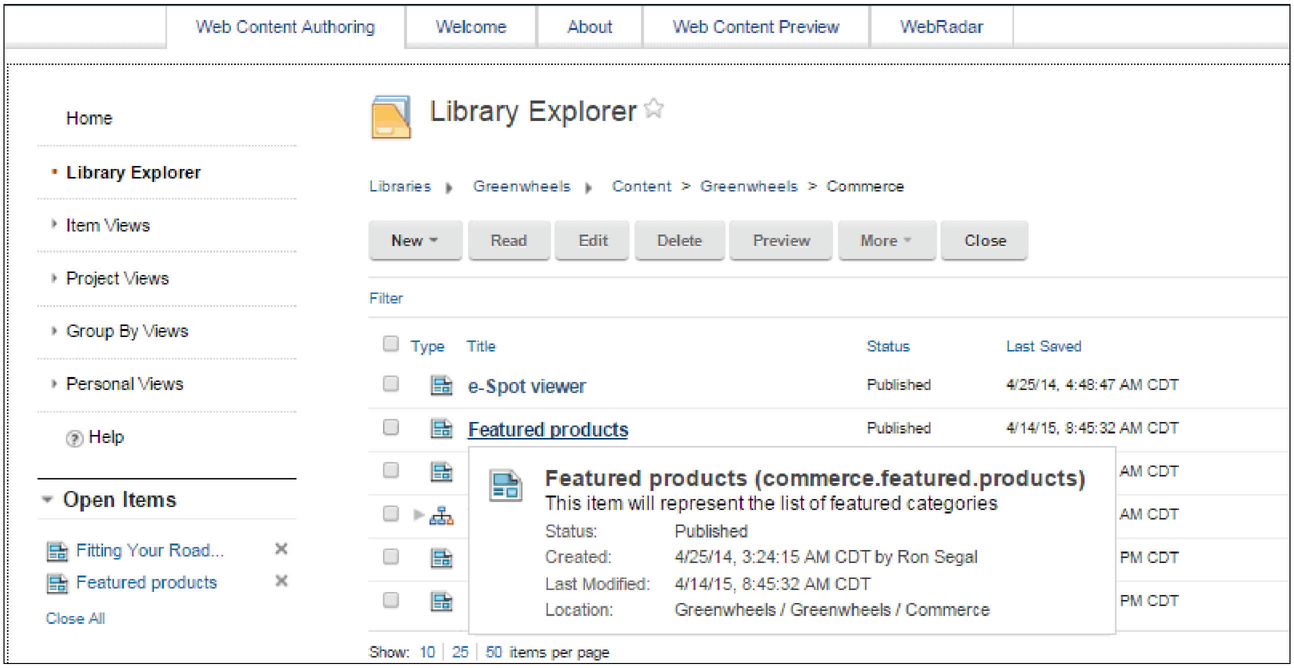
section headline:

e-Spot name

e-Spot name
○ BikesCategories
○ BikesRow4_CatEntries

Figure 4-28 Create Featured products content based on Greenwheels Shop (advanced) (2 of 2)

Figure 4-29, Figure 4-30 on page 95, and Figure 4-31 on page 96 show the Featured products content that is created in content library Greenwheels by using the Greenwheels Shop (advanced) authoring template.

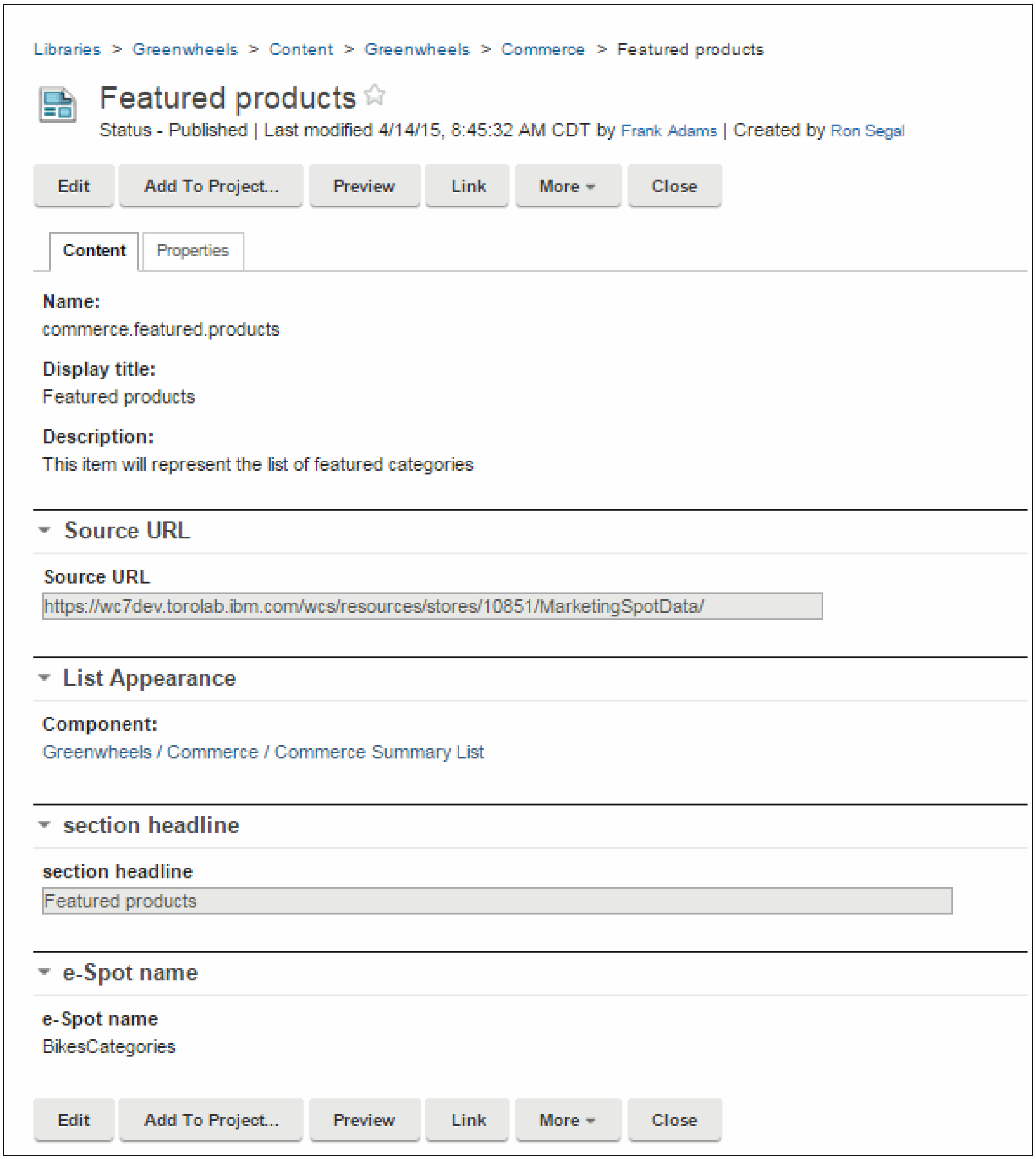


Type	Title	Status	Last Saved
	e-Spot viewer	Published	4/25/14, 4:48:47 AM CDT
	Featured products	Published	4/14/15, 8:45:32 AM CDT
			AM CDT
			AM CDT
			PM CDT
			PM CDT

Featured products (commerce.featured.products)
This item will represent the list of featured categories

Status: Published
Created: 4/25/14, 3:24:15 AM CDT by Ron Segal
Last Modified: 4/14/15, 8:45:32 AM CDT
Location: Greenwheels / Greenwheels / Commerce

Figure 4-29 Feature products - Web Content Manager content



Libraries > Greenwheels > Content > Greenwheels > Commerce > Featured products

Featured products

Status - Published | Last modified 4/14/15, 8:45:32 AM CDT by Frank Adams | Created by Ron Segal

Name:
commerce.featured.products

Display title:
Featured products

Description:
This item will represent the list of featured categories

Source URL

Source URL
https://wc7dev.torolab.ibm.com/wcs/resources/stores/10851/MarketingSpotData/

List Appearance

Component:
Greenwheels / Commerce / Commerce Summary List

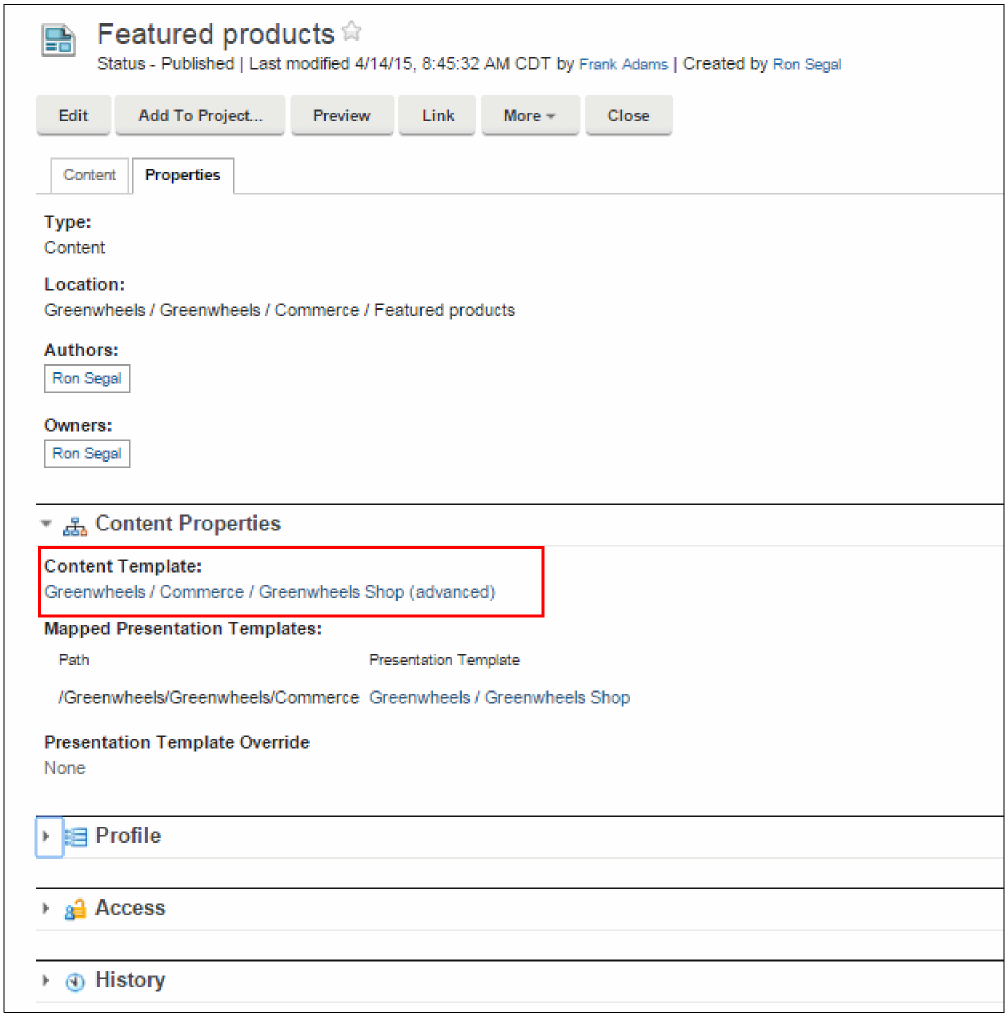
section headline

section headline
Featured products

e-Spot name

e-Spot name
BikesCategories

Figure 4-30 Featured products - Content tab



Status - Published | Last modified 4/14/15, 8:45:32 AM CDT by Frank Adams | Created by Ron Segal

Type:
Content

Location:
Greenwheels / Greenwheels / Commerce / Featured products

Authors:
Ron Segal

Owners:
Ron Segal

Content Template:
Greenwheels / Commerce / Greenwheels Shop (advanced)

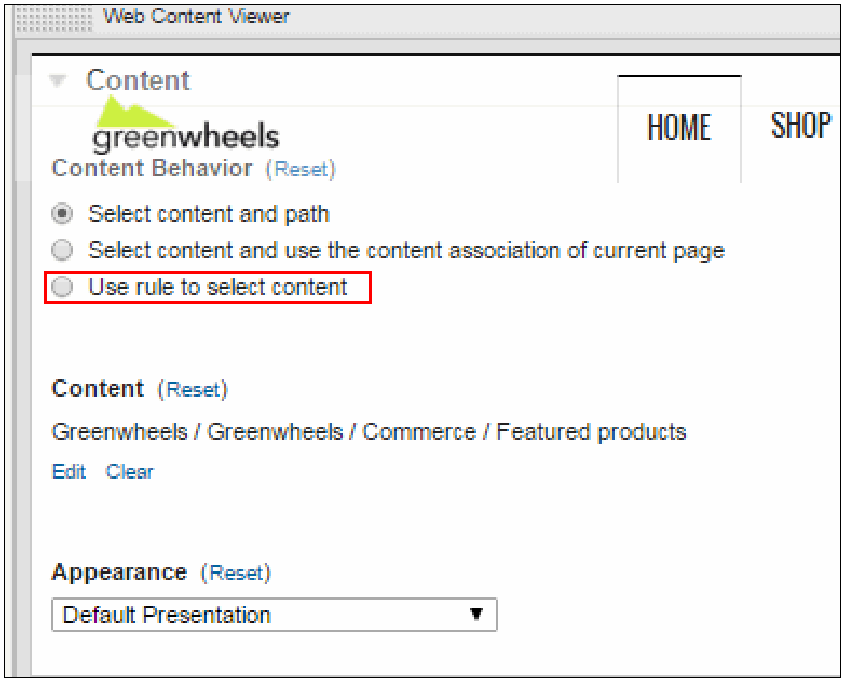
Mapped Presentation Templates:

Path	Presentation Template
/Greenwheels/Greenwheels/Commerce	Greenwheels / Greenwheels Shop

Presentation Template Override
None

Figure 4-31 Featured products - Properties tab

Figure 4-32 shows the Web Content Viewer portlet configuration to render the content.


Web Content Viewer

Content

Content Behavior (Reset)

● Select content and path
○ Select content and use the content association of current page
○ Use rule to select content

Content (Reset)

Greenwheels / Greenwheels / Commerce / Featured products

Edit Clear

Appearance (Reset)

Default Presentation ▼

Figure 4-32 Web Content Viewer configuration

4.7.2 Creating simple and complex authoring templates

The authoring template defines elements that are used in a web content. It provides a separation of authoring and presentation components, where the business user or author enters the content data and the web developer writes code to display this data on the page. The template can be simple and include only one element, such as text, HTML fragment, or image. It also can be also complex and include multiple elements.

This section describes examples of how to create authoring templates for Knowledge Center content and News Article content, which are part of a fictitious digital experience that is provided by the Greenwheels company.

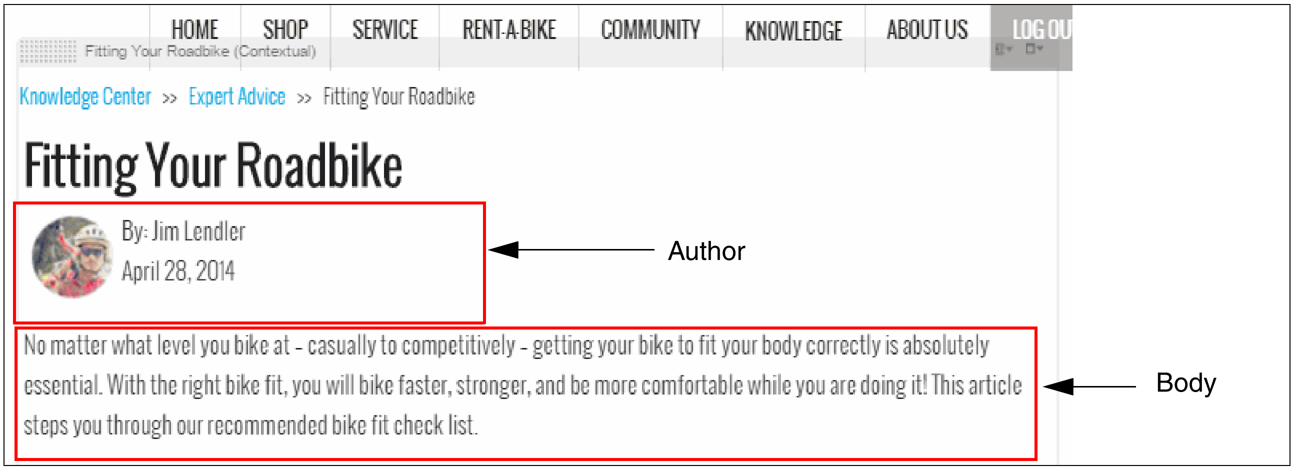
The following examples are presented in this section:

- “Example: Creating simple content for Knowledge Center information” on page 97
- “Creating complex content for Knowledge Center information” on page 101

Example: Creating simple content for Knowledge Center information

This example shows the characteristics of a simple authoring template that can be used by content creators to create content for the Knowledge Center that is part of the Greenwheels digital experience.

Figure 4-33 shows simple Knowledge Center content as seen by the users of the Greenwheels demonstration digital experience. This content was created with the simple authoring template that is described in this example.



Knowledge Center >> Expert Advice >> Fitting Your Roadbike

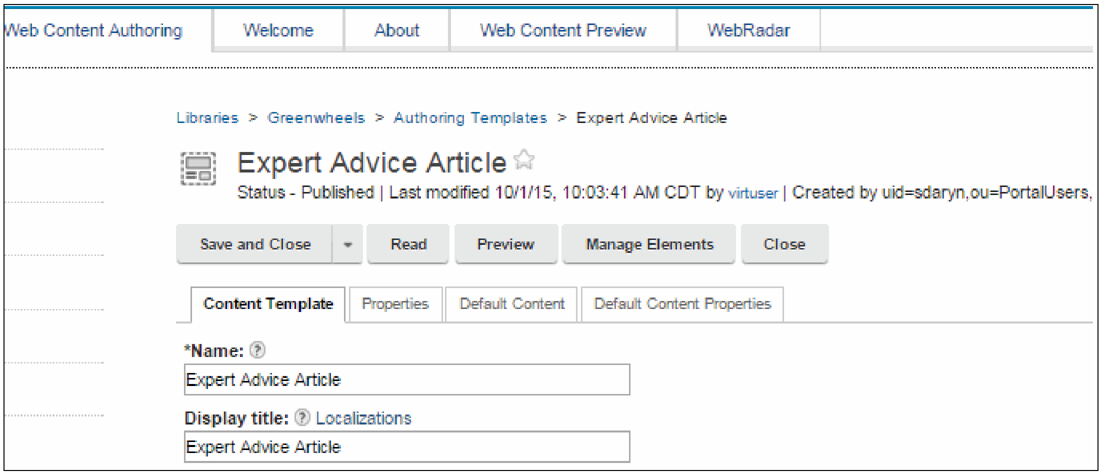
Fitting Your Roadbike

By: Jim Lendler
April 28, 2014

No matter what level you bike at – casually to competitively – getting your bike to fit your body correctly is absolutely essential. With the right bike fit, you will bike faster, stronger, and be more comfortable while you are doing it! This article steps you through our recommended bike fit check list.

Figure 4-33 Simple Knowledge Center content in the Greenwheels demonstration digital experience

Figure 4-34 shows the main characteristics of the Expert Advice Article authoring template that is used to create the simple content that is featured in Figure 4-33.



Libraries > Greenwheels > Authoring Templates > Expert Advice Article

Expert Advice Article

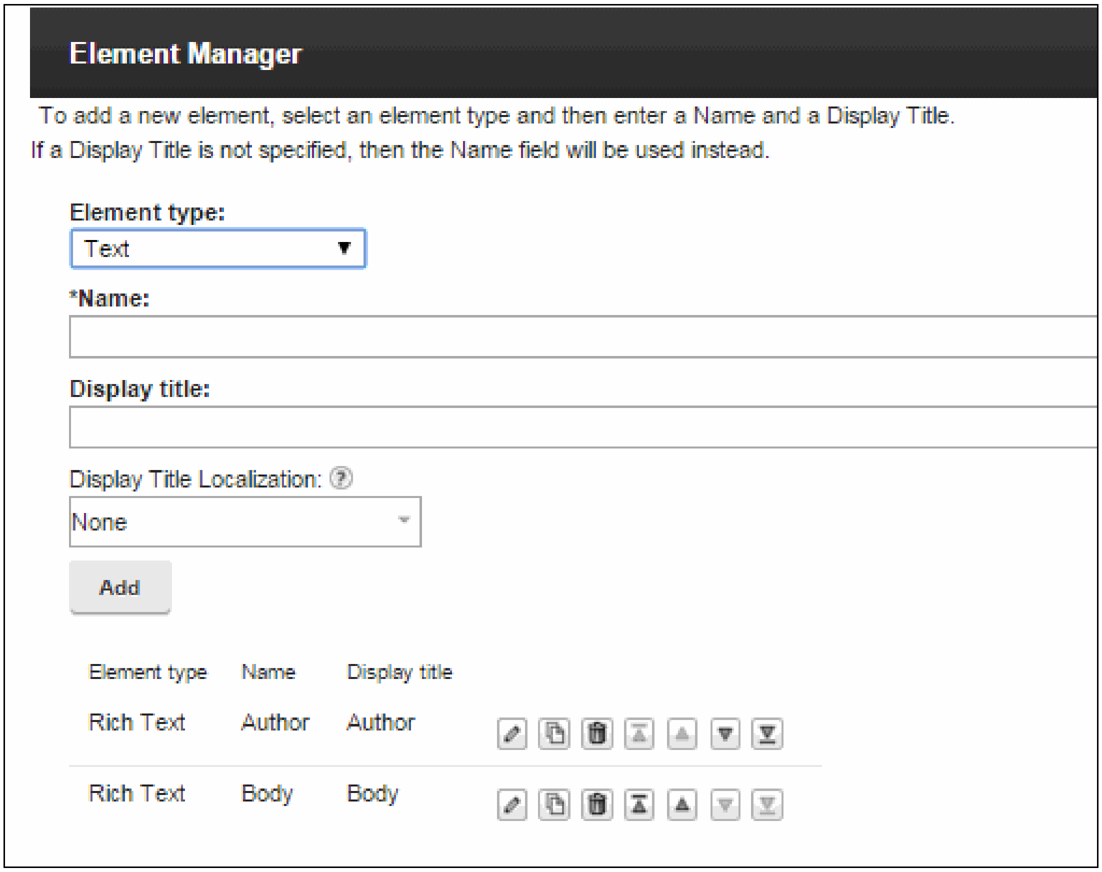
Status - Published | Last modified 10/1/15, 10:03:41 AM CDT by virtuser | Created by uid=sdaryn,ou=PortalUsers,

*Name: Expert Advice Article

Display title: Localizations Expert Advice Article

Figure 4-34 Sample authoring template to create simple content

As shown in Figure 4-35, the authoring template in this example includes only two rich text elements. More elements of various types can be added, as described steps 9 on page 89 and 10 on page 89.



Element Manager

To add a new element, select an element type and then enter a Name and a Display Title.
If a Display Title is not specified, then the Name field will be used instead.

Element type: Text

*Name:

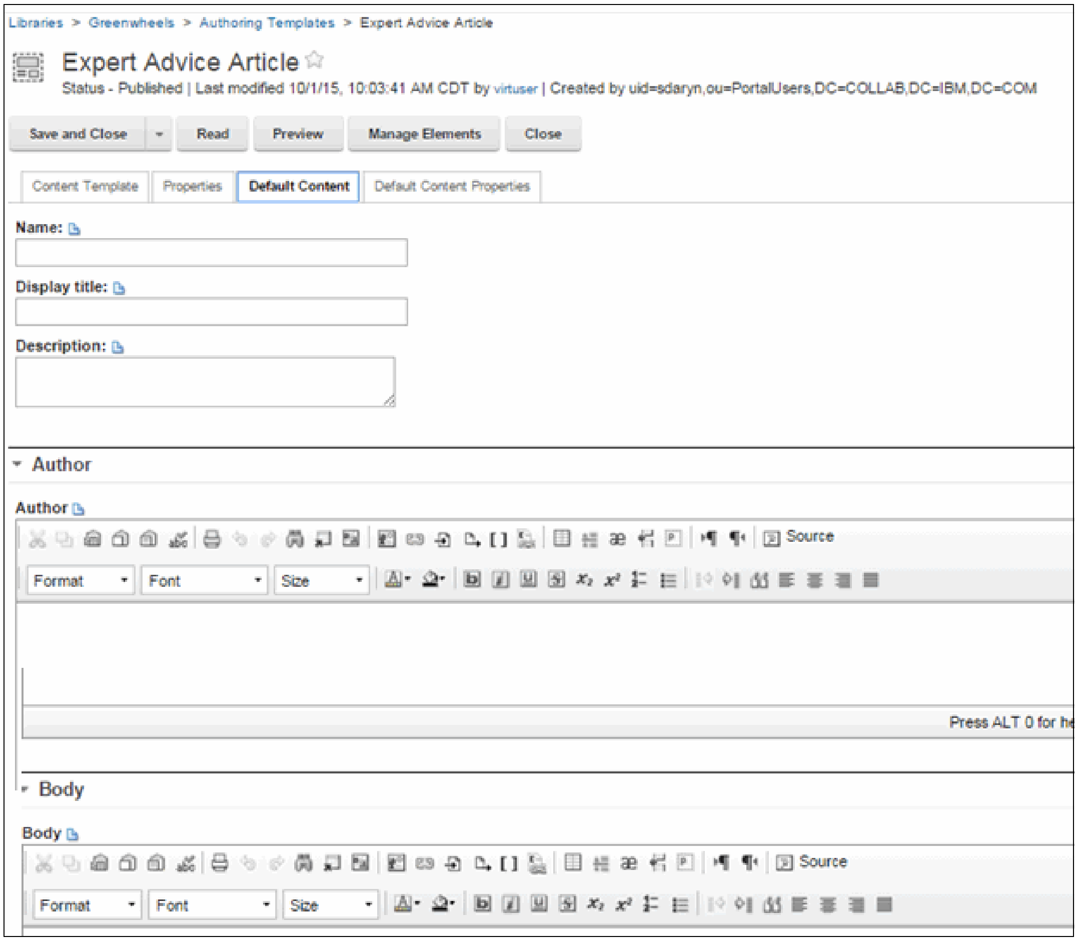
Display title:

Display Title Localization: None

Element type	Name	Display title
Rich Text	Author	Author
Rich Text	Body	Body

Figure 4-35 Sample authoring template containing only two rich text elements

Figure 4-36 shows the elements in the simple Expert Advice Article authoring template that are displayed by editing the template and clicking **Manage Elements**.



Libraries > Greenwheels > Authoring Templates > Expert Advice Article

Expert Advice Article

Status - Published | Last modified 10/1/15, 10:03:41 AM CDT by virtuser | Created by uid=sdaryn,ou=PortalUsers,DC=COLLAB,DC=IBM,DC=COM

Name:

Display title:

Description:

Author

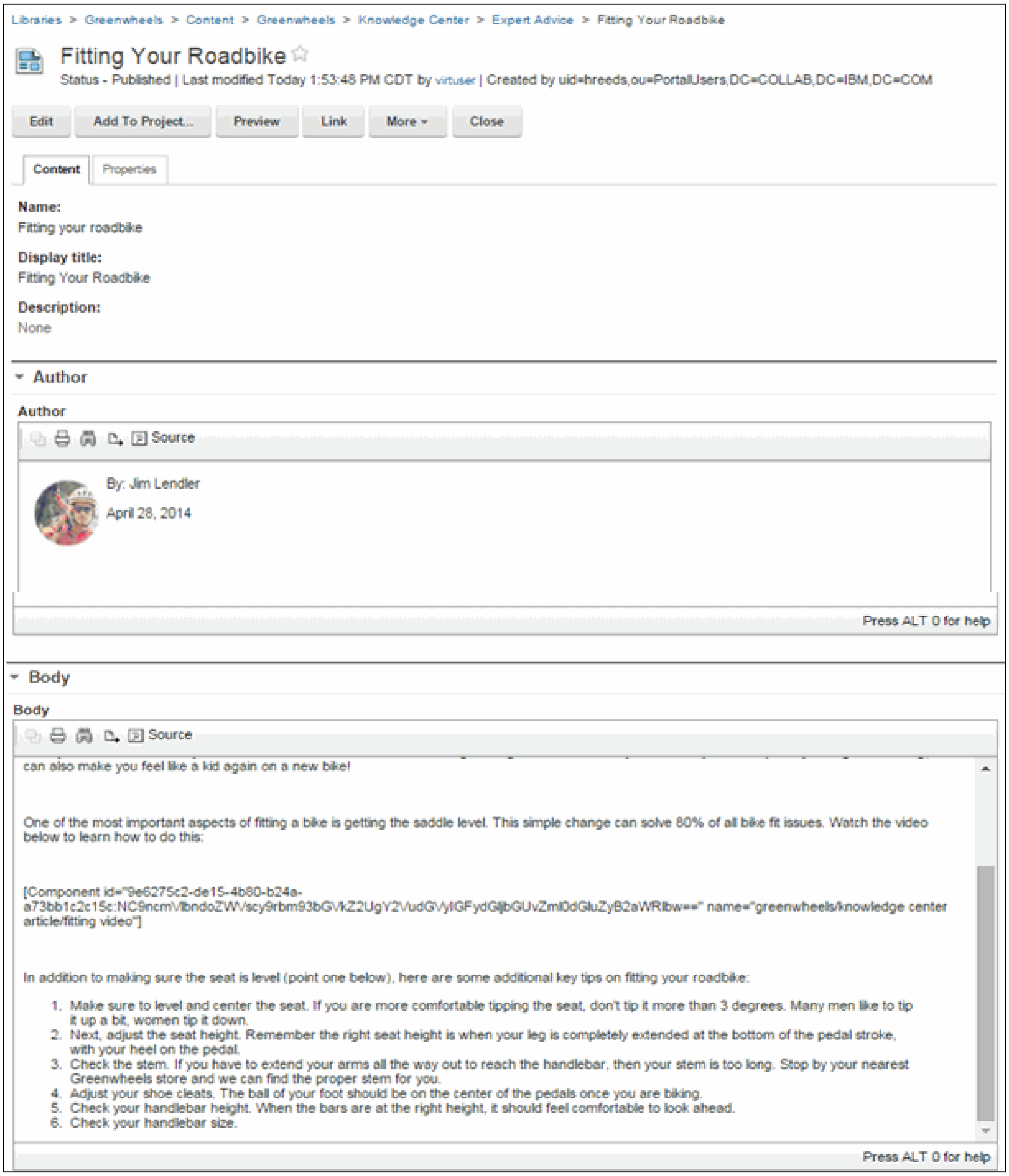
Author

Body

Body

Figure 4-36 Sample authoring template - Default Content tab

Figure 4-37 shows the Web Content Manager content *Fitting Your Roadbike*, which is based on the simple *Expert Advice Article* authoring template.


Libraries > Greenwheels > Content > Greenwheels > Knowledge Center > Expert Advice > Fitting Your Roadbike
Edit
Add To Project...
Preview
Link
More ▾
Close
Content
Properties
Source
Press ALT 0 for help

Fitting Your Roadbike

Status - Published | Last modified Today 1:53:48 PM CDT by virtuser | Created by uid=hreeds,ou=PortalUsers,DC=COLLAB,DC=IBM,DC=COM

Name:
Fitting your roadbike

Display title:
Fitting Your Roadbike

Description:
None

▾ Author

Author

By: Jim Lendler

April 28, 2014

▾ Body

Body

can also make you feel like a kid again on a new bike!

One of the most important aspects of fitting a bike is getting the saddle level. This simple change can solve 80% of all bike fit issues. Watch the video below to learn how to do this:

[Component id="9e6275c2-de15-4b80-b24a-a73bb1c2c15c:NC9ncmVlbndoZWVscy9rbm93bGVkZ2UgY2VudGVyIGFydGljbGUvZml0dGluZyB2aWRlbw==" name="greenwheels/knowledge center article/fitting video"]

In addition to making sure the seat is level (point one below), here are some additional key tips on fitting your roadbike:

1. Make sure to level and center the seat. If you are more comfortable tipping the seat, don't tip it more than 3 degrees. Many men like to tip it up a bit, women tip it down.
2. Next, adjust the seat height. Remember the right seat height is when your leg is completely extended at the bottom of the pedal stroke, with your heel on the pedal.
3. Check the stem. If you have to extend your arms all the way out to reach the handlebar, then your stem is too long. Stop by your nearest Greenwheels store and we can find the proper stem for you.
4. Adjust your shoe cleats. The ball of your foot should be on the center of the pedals once you are biking.
5. Check your handlebar height. When the bars are at the right height, it should feel comfortable to look ahead.
6. Check your handlebar size.

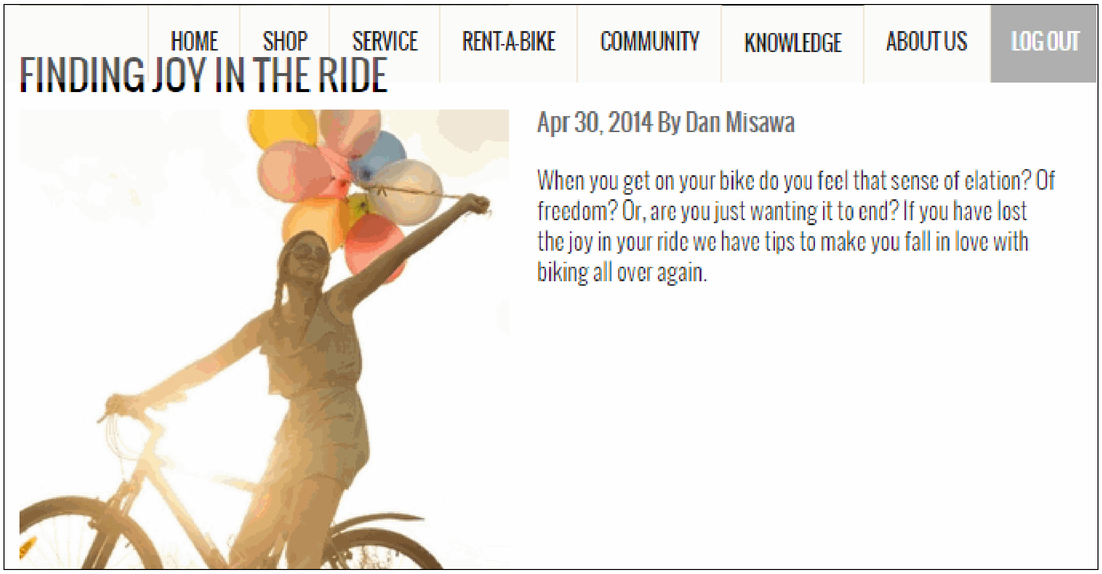
Figure 4-37 Sample Web Content Manager content based on a simple authoring template

Creating complex content for Knowledge Center information

This example shows that the characteristics of a complex authoring template that is called *Knowledge Center Article*, which can be used by content creators to create more complex content for the Knowledge Center that is part of the Greenwheels demonstration digital experience.

The authoring template in this example includes several elements of various types.

Figure 4-38 shows Knowledge Center content as seen by the users of the Greenwheels digital experience. This content was created based on the authoring template that is described in this example.



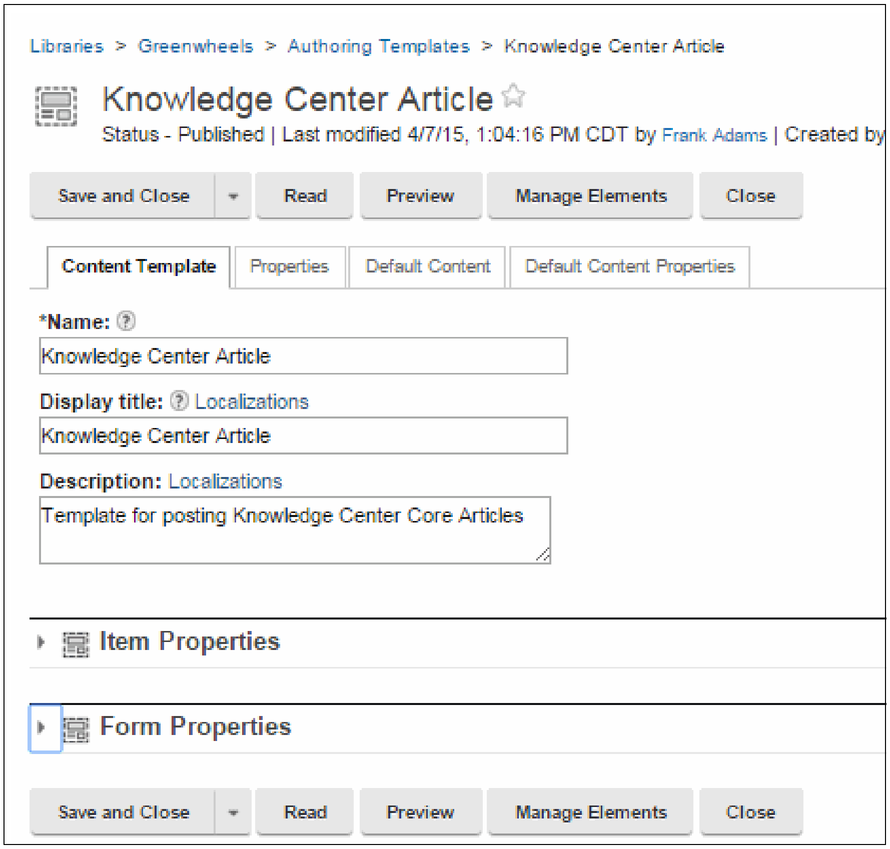
FINDING JOY IN THE RIDE

Apr 30, 2014 By Dan Misawa

When you get on your bike do you feel that sense of elation? Of freedom? Or, are you just wanting it to end? If you have lost the joy in your ride we have tips to make you fall in love with biking all over again.

Figure 4-38 Complex Knowledge Center content in the Greenwheels demonstration digital experience

Figure 4-39 shows the main characteristics of the Knowledge Center Article authoring template that is used to create the content that is featured in Figure 4-38.



Knowledge Center Article ☆

Status - Published | Last modified 4/7/15, 1:04:16 PM CDT by Frank Adams | Created by

*Name: Knowledge Center Article

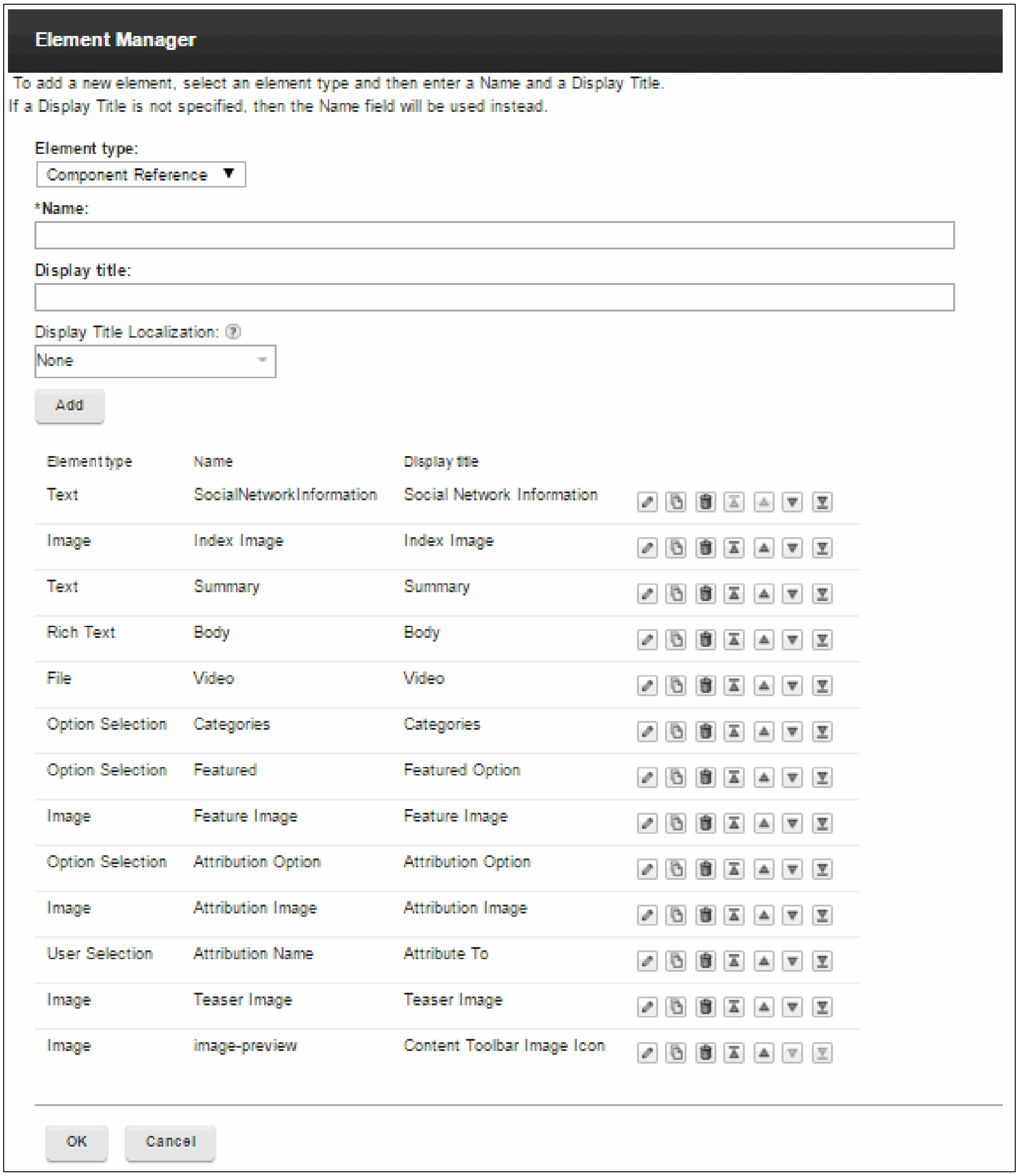
Display title: Localizations
Knowledge Center Article

Description: Localizations
Template for posting Knowledge Center Core Articles

Figure 4-39 Sample authoring template to create complex content

Figure 4-40 shows the elements in the Knowledge Center Article authoring template that are displayed by editing the authoring template and clicking **Manage Elements**.

This sample authoring template contain several elements of various element types, such as Text, Rich Text, File, Image, Option Selection, and User Selection. The elements are used to include video, images, data about community integration, and more.



Element Manager

To add a new element, select an element type and then enter a Name and a Display Title.
If a Display Title is not specified, then the Name field will be used instead.

Element type: Component Reference

*Name:

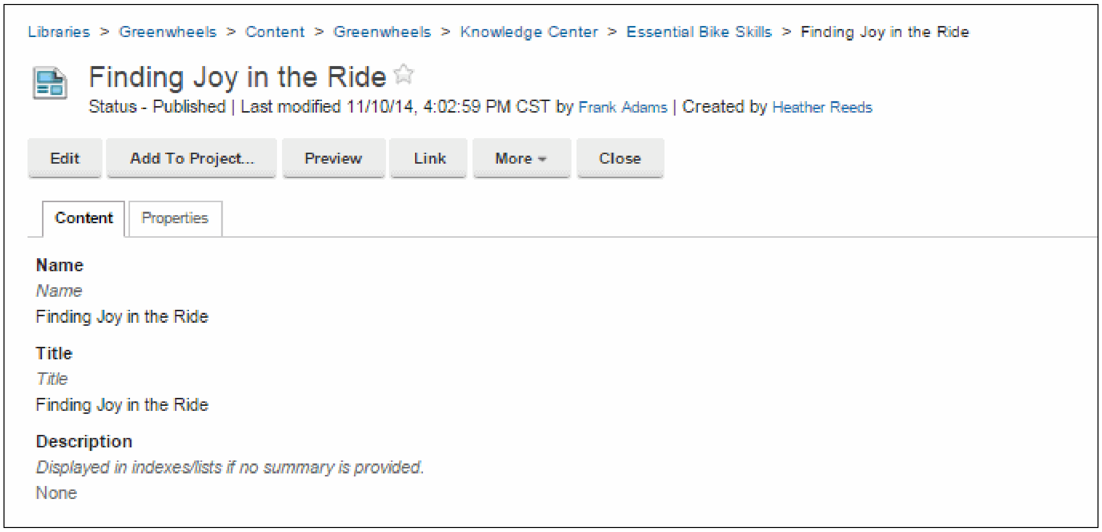
Display title:

Display Title Localization: None

Element type	Name	Display title
Text	SocialNetworkInformation	Social Network Information
Image	Index Image	Index Image
Text	Summary	Summary
Rich Text	Body	Body
File	Video	Video
Option Selection	Categories	Categories
Option Selection	Featured	Featured Option
Image	Feature Image	Feature Image
Option Selection	Attribution Option	Attribution Option
Image	Attribution Image	Attribution Image
User Selection	Attribution Name	Attribute To
Image	Teaser Image	Teaser Image
Image	image-preview	Content Toolbar Image Icon

Figure 4-40 Sample authoring template containing several element types

Figure 4-41, Figure 4-42, Figure 4-43 on page 105, and Figure 4-44 on page 106 show the Web Content Manager content Finding Joy in the Ride based on the Knowledge Center Article authoring template.



Libraries > Greenwheels > Content > Greenwheels > Knowledge Center > Essential Bike Skills > Finding Joy in the Ride

Finding Joy in the Ride

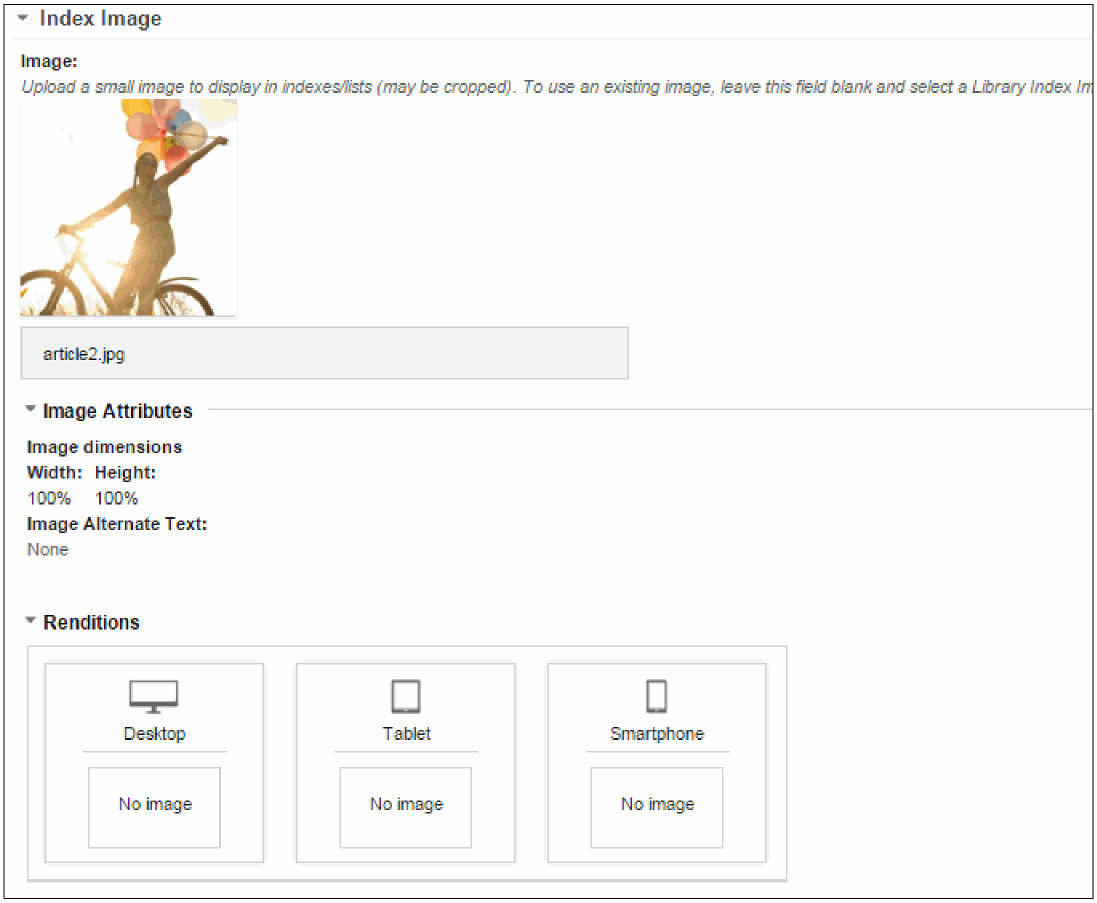
Status - Published | Last modified 11/10/14, 4:02:59 PM CST by Frank Adams | Created by Heather Reeds

Name
Name
Finding Joy in the Ride

Title
Title
Finding Joy in the Ride

Description
Displayed in indexes/lists if no summary is provided.
None

Figure 4-41 Web Content Manager content based on complex authoring template (1 of 4)



Index Image

Image:
Upload a small image to display in indexes/lists (may be cropped). To use an existing image, leave this field blank and select a Library Index Im

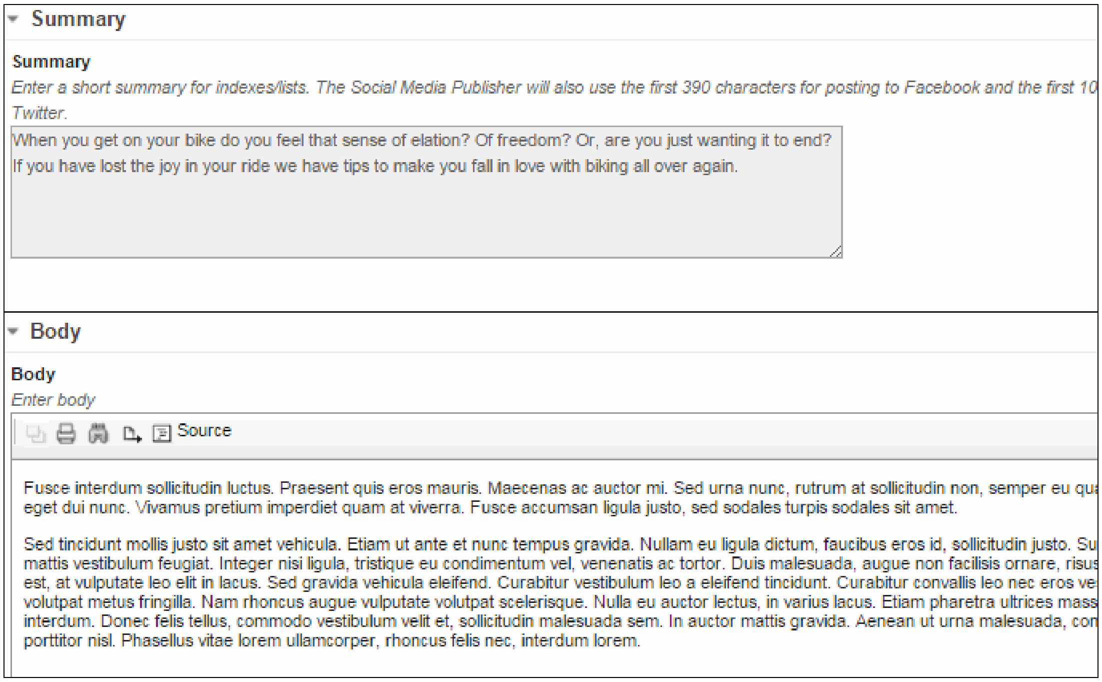
article2.jpg

Image Attributes

Image dimensions
Width: **Height:**
100% 100%
Image Alternate Text:
None

Renditions

Figure 4-42 Web Content Manager content based on complex authoring template (2 of 4)



Summary

Enter a short summary for indexes/lists. The Social Media Publisher will also use the first 390 characters for posting to Facebook and the first 10
Twitter.

When you get on your bike do you feel that sense of elation? Of freedom? Or, are you just wanting it to end?
If you have lost the joy in your ride we have tips to make you fall in love with biking all over again.

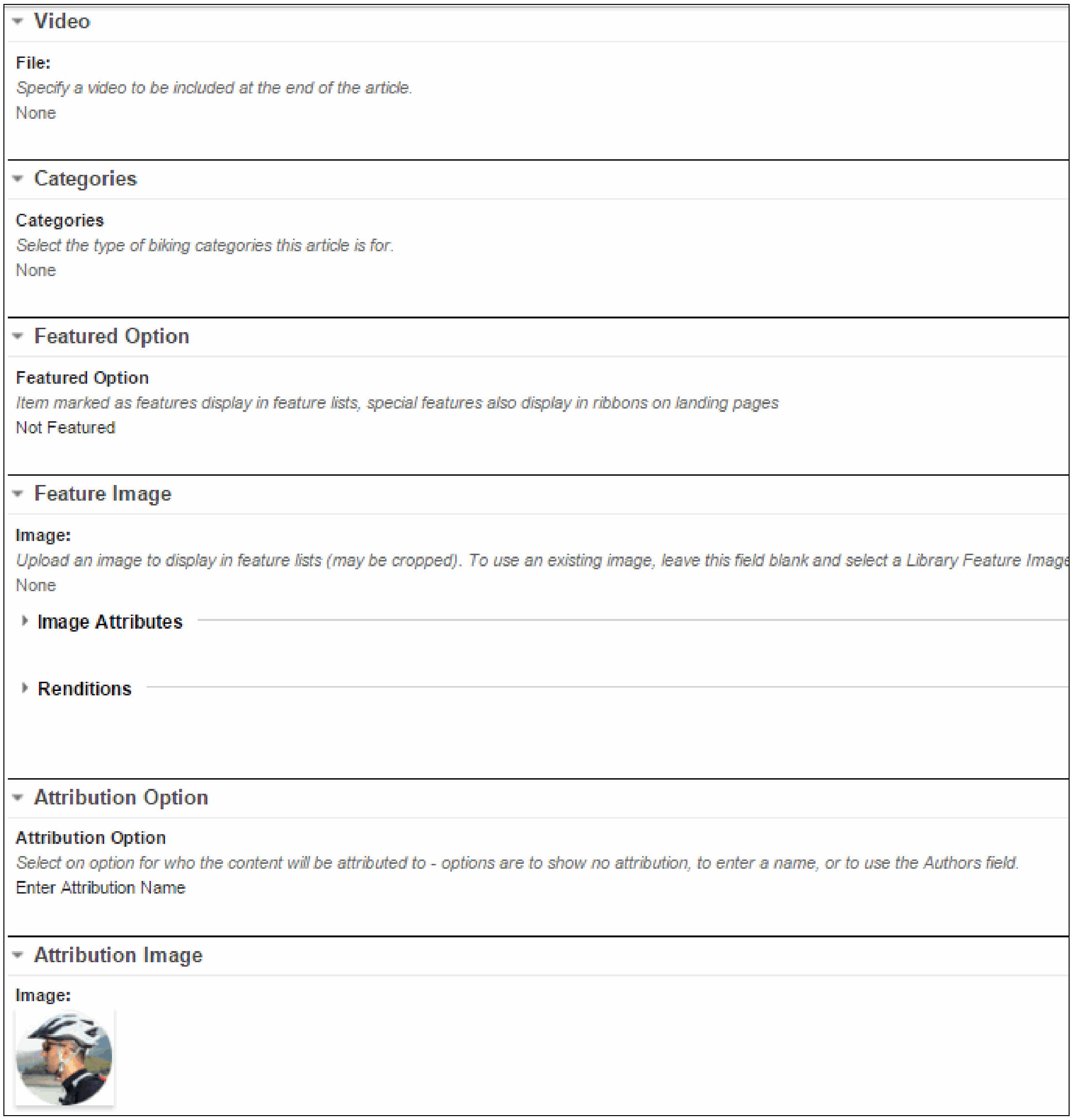
Body

Enter body

Fusce interdum sollicitudin luctus. Praesent quis eros mauris. Maecenas ac auctor mi. Sed urna nunc, rutrum at sollicitudin non, semper eu qu
eget dui nunc. Vivamus pretium imperdiet quam at viverra. Fusce accumsan ligula justo, sed sodales turpis sodales sit amet.

Sed tincidunt mollis justo sit amet vehicula. Etiam ut ante et nunc tempus gravida. Nullam eu ligula dictum, faucibus eros id, sollicitudin justo. Su
mattis vestibulum feugiat. Integer nisi ligula, tristique eu condimentum vel, venenatis ac tortor. Duis malesuada, augue non facilisis ornare, risus
est, at vulputate leo elit in lacus. Sed gravida vehicula eleifend. Curabitur vestibulum leo a eleifend tincidunt. Curabitur convallis leo nec eros ve
volutpat metus fringilla. Nam rhoncus augue vulputate volutpat scelerisque. Nulla eu auctor lectus, in varius lacus. Etiam pharetra ultrices mass
interdum. Donec felis tellus, commodo vestibulum velit et, sollicitudin malesuada sem. In auctor mattis gravida. Aenean ut urna malesuada, cor
porttitor nisl. Phasellus vitae lorem ullamcorper, rhoncus felis nec, interdum lorem.

Figure 4-43   Web Content Manager content based on complex authoring template (3 of 4)



Video

File:

Specify a video to be included at the end of the article.

None

Categories

Categories

Select the type of biking categories this article is for.

None

Featured Option

Featured Option

Item marked as features display in feature lists, special features also display in ribbons on landing pages

Not Featured

Feature Image

Image:

Upload an image to display in feature lists (may be cropped). To use an existing image, leave this field blank and select a Library Feature Image

None

Attribution Option

Attribution Option

Select on option for who the content will be attributed to - options are to show no attribution, to enter a name, or to use the Authors field.

Enter Attribution Name

Attribution Image

Image:

Figure 4-44 Web Content Manager content based on complex authoring template (4 of 4)

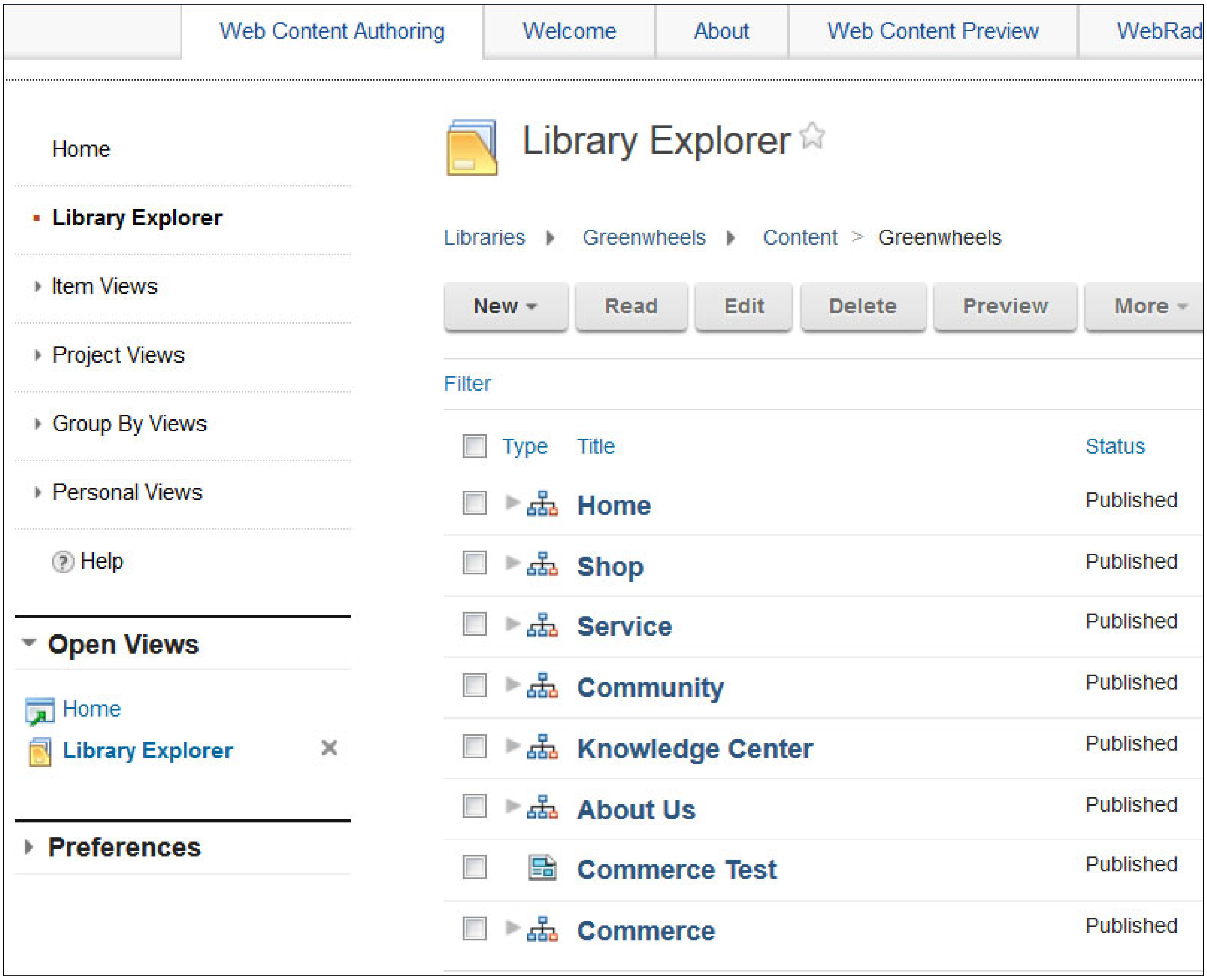
4.8 Site area

Site areas are used to group specific categories of the content. In some cases, site areas are similar to the website navigation. Any site has navigation for the home page (which is also the landing page), About Us, and so on.

For the Greenwheels demonstration digital experience, the site areas are created in Web Content Manager as follows:

- Home
- Shop
- Service
- Community
- Knowledge
- About Us

Each of these site areas contains the content that is relevant to the specific area. Figure 4-45 shows the top-level site areas in the Greenwheels demonstration.



□	Type	Title	Status
□		Home	Published
□		Shop	Published
□		Service	Published
□		Community	Published
□		Knowledge Center	Published
□		About Us	Published
□		Commerce Test	Published
□		Commerce	Published

Figure 4-45 Site area for the Greenwheels demonstration sample site

4.9 Components

You create components to store a single element that is reused in multiple locations in your website, for example, a company logo or a copyright notice. You can also use components to encapsulate markup so that this markup can be reused by different Web Content Manager artifacts or create components that contain scripting logic, such as JSP code.

There are three types of components that you can use to build digital experiences:

- Static components are used to store static content, such as text, files, images, or link components.
- Dynamic components are used to generate dynamically content that is based on the parameters set in the component properties, such as menu, navigator, or personalization components.
- Tool components are used to create tools that can be added to web pages for users to do tasks, such as search, inline editing, and paging through pages of links, such as authoring tools component, and page navigation component.

This section demonstrates the usage of reusable components in the Greenwheels example and looks at the home page that renders the following components:

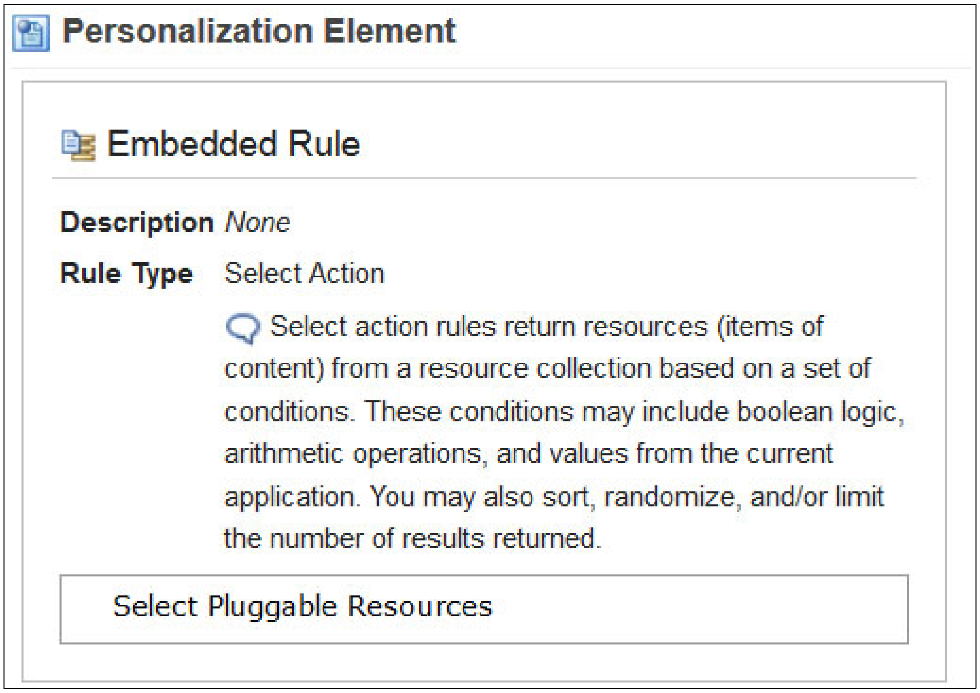
- Featured products: An HTML component.
- Latest news: An HTML component.
- Products: A combination of HTML and personalization components that embed WebSphere Commerce integration to retrieved products data and display them as a list or table.

The HTML component in the web content management system is a reusable artifact that can contain HTML code that is shared across the system or it can be used as a wrapper that embeds HTML code referencing other web content management artifacts.

The Featured product in the example is an HTML component that contains a link to an image in the web content management system. Similarly, the Featured news and Events components render images and text artifacts in the web content management system.

The Products component is a more complicated item that combines data from a remote commerce server. As shown in Figure 4-46, a personalization element is used to display images on a remote WebSphere Commerce server.

For more information about personalization, see 4.14, “Personalization” on page 119.



Personalization Element

Embedded Rule

Description *None*

Rule Type Select Action

Select action rules return resources (items of content) from a resource collection based on a set of conditions. These conditions may include boolean logic, arithmetic operations, and values from the current application. You may also sort, randomize, and/or limit the number of results returned.

Figure 4-46 Personalization element

Example 4-1 shows Compare products content that retrieves an array of product information from WebSphere Commerce and displays the information on a page that is rendered by WebSphere Portal.

Example 4-1 Code to start a DDC plug-in

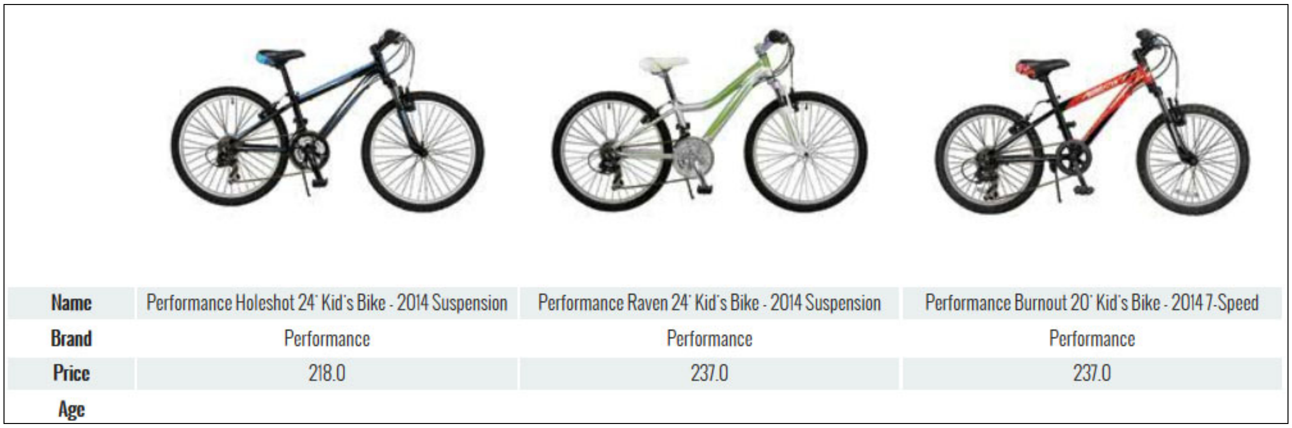
```
[Plugin:ListRenderingContext extension-id="ibm.portal.ddc.xml" action="set"
profile="ibm.portal.wc_catalog" compute="once"
attribute="source=https://wc7dev.torolab.ibm.com/wcs/resources/store/10851/product
view/byIds?[Plugin:RenderParam type='private'
defaultValue='id=16573&id=16577&id=16581'
key='selectedProducts']&responseFormat=xml"]
```

As shown in Example 4-2, you can also build a list of products belonging to certain group, such as “bicycles”. The plug-in uses product IDs as a parameter.

Example 4-2 Code to render retrieved products attributes

```
{
 "image":"[Plugin:ResourceURL proxy="true" compute="once" url="[Element
context='current' type='content' key='sourceURL'][AttributeResource
attributeName='thumbnail' separator=',']"]",
 "name":"[AttributeResource attributeName="name" separator=","]",
 "manufacturer":"[AttributeResource attributeName="manufacturer" separator=","]",
 "kidsage":"[AttributeResource attributeName="kidsage" separator=","]",
 "isnew":"[AttributeResource attributeName="isnew" separator=","]",
 "price":"[AttributeResource attributeName="priceValue" separator=","]"
}
```

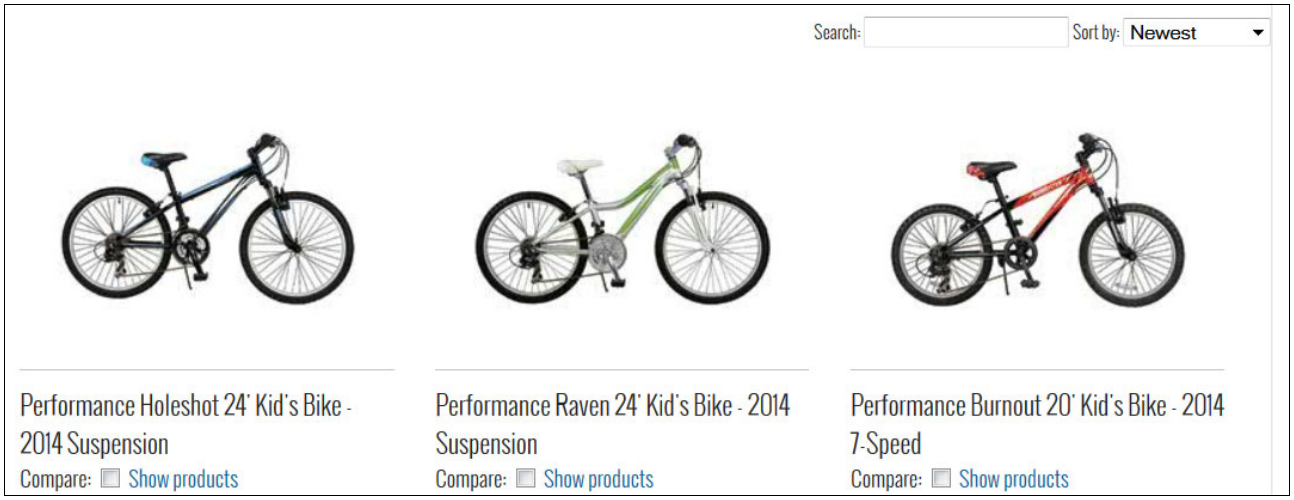
Figure 4-47 illustrates the scenario where a list of all products belonging to a certain group is displayed.



Name	Performance Holeshot 24' Kid's Bike - 2014 Suspension	Performance Raven 24' Kid's Bike - 2014 Suspension	Performance Burnout 20' Kid's Bike - 2014 7-Speed
Brand	Performance	Performance	Performance
Price	218.0	237.0	237.0
Age			

Figure 4-47 List of products belonging to a specific group - results returned by the DDC plug-in

Figure 4-48 illustrates the scenario where a list of products to compare is displayed.



Search: Sort by: Newest

Performance Holeshot 24' Kid's Bike - 2014 Suspension
Compare: □ Show products

Performance Raven 24' Kid's Bike - 2014 Suspension
Compare: □ Show products

Performance Burnout 20' Kid's Bike - 2014 7-Speed
Compare: □ Show products

Figure 4-48 List of comparable products returned by the DDC plug-in

Tip: These examples use the URL of the WebSphere Commerce server. It is considered a preferred practice to create a component in Web Content Manager that embeds the host name so that you can reference this component across multiple presentation elements and easily update the data in one place.

4.10 Web Content Viewer portlet

The type of delivery method that you use to deliver web content to your viewers depends on the type of content that is being delivered and the type of viewers for which the website is intended. Using tools such as Web Content Viewer portlets, content associations, and web content page templates, you can build portal pages and display web content. You also can combine web content with other portlet-based content. Content associations tie viewers and portal pages to the site structure of your web content libraries.

When you add web content portlets to your site, you must edit the settings of these portlets. To configure a web content portlet, select one of the following options from the menu in the portlet title bar:

- Configure

  The Configure option specifies settings for all users of all instances of a web content portlet, regardless of the page on which the portlet instance appears. Configure is also system-wide, including virtual portals.

- Edit Shared Settings

  The Edit Shared Settings option specifies settings for all users of this instance of a web content portlet. Changes that you make in the Edit Shared Settings mode are not reflected in other instances of a web content portlet.

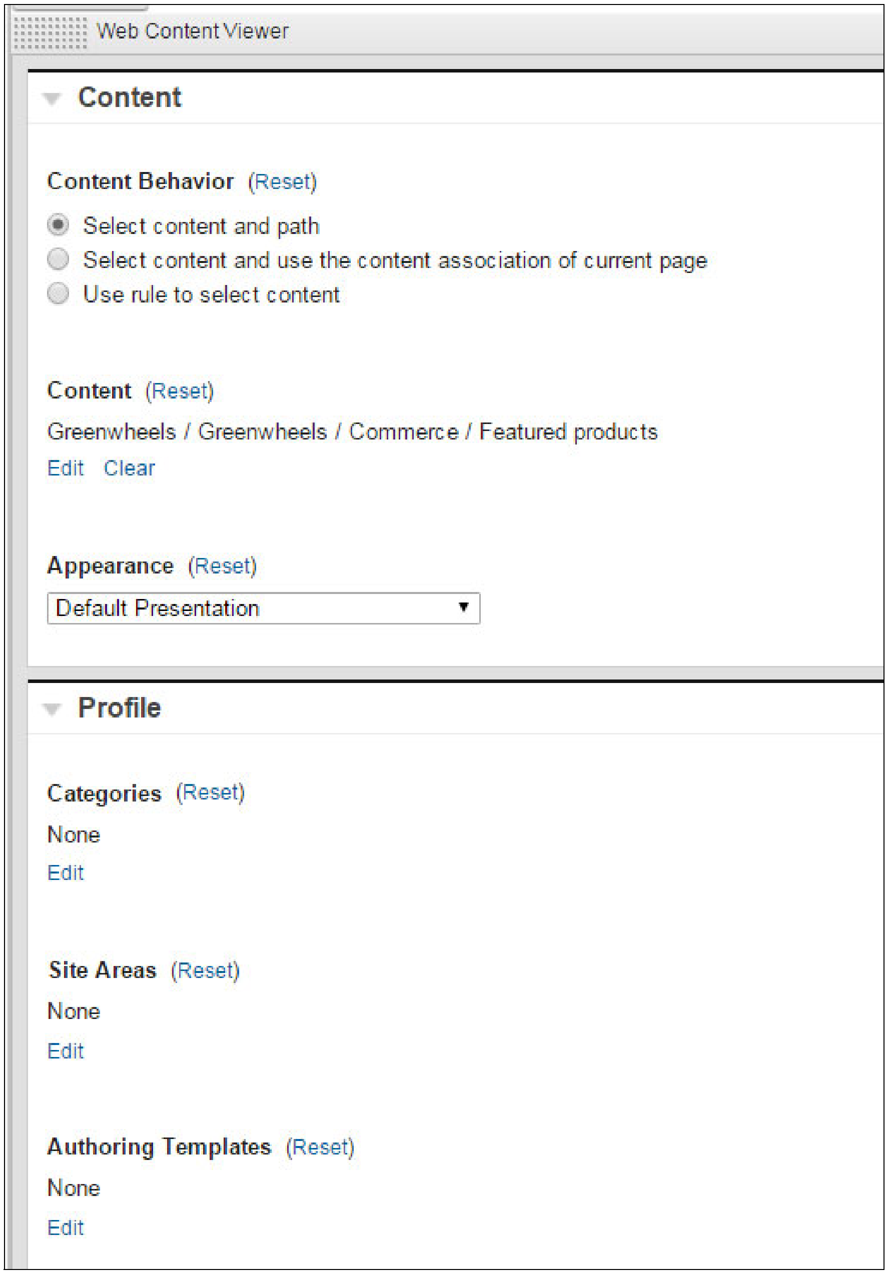
  Note: When the shared settings of a portlet are edited, the default Configure mode settings are not displayed on a page, even if you edit the Configure mode settings. To restore a portlet to the Configure mode, you must delete the portlet from the page and add it back.

  For information about the settings of the Web Content Viewer portlet, see the *Editing the settings of a Web Content Viewer* in topic in the IBM WebSphere Portal V8.5.0 IBM Knowledge Center:

  http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/help/panel_help/wcm_deploy_delivery_local-rendering-portlet.dita

Figure 4-49 shows the Edit Shared Setting for the Web Content Viewer portlet to display Featured Products in the home page, as shown in Figure 4-19 on page 86.

The Web Content Viewer portlet in this example is configured to render the Featured products content (shown in Figure 4-29 on page 94, Figure 4-30 on page 95, and Figure 4-31 on page 96) and created by using the greenwheels.shop.advance authoring template, as described in 4.7.1, “Example: Creating Web Content Manager content” on page 86.



Web Content Viewer

Content

Content Behavior (Reset)

● Select content and path
○ Select content and use the content association of current page
○ Use rule to select content

Content (Reset)

Greenwheels / Greenwheels / Commerce / Featured products

Edit Clear

Appearance (Reset)

Default Presentation ▼

Profile

Categories (Reset)

None

Edit

Site Areas (Reset)

None

Edit

Authoring Templates (Reset)

None

Edit

Figure 4-49 Edit Shared Settings view for Web Content Viewer

A portal page can have one or multiple instances of the Web Content Viewer portlets and other portlets that make up the complete page and that render multiple pieces of content all in a single page.

4.11 Web Content Manager

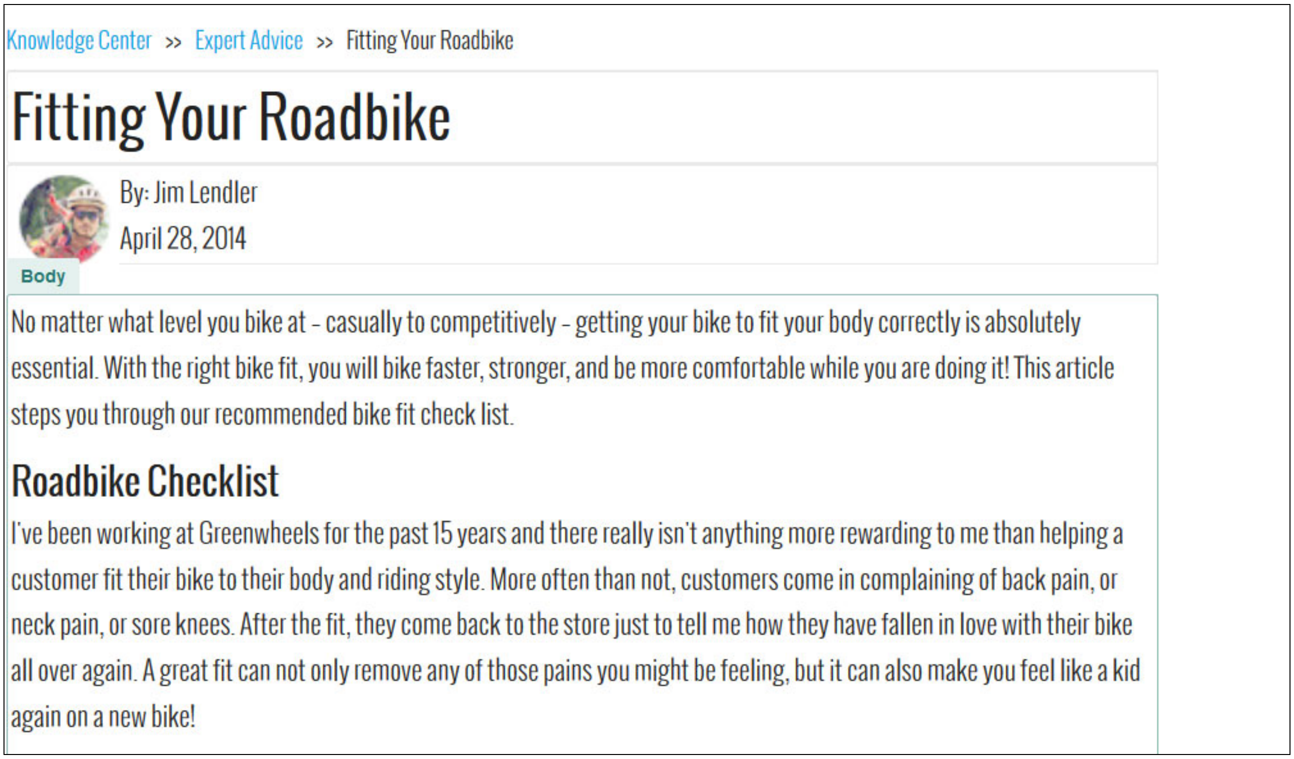
Rich content and elegant user experiences drive the success of websites. In IBM Digital Experience, Web Content Manager is used to store and manage content. Web Content Manager separates the content from the presentation layer. Content authors enter information in authoring templates. The authoring template is then associated with a presentation template.

The presentation template defines how the content is rendered and presented to the visitors of the site. For more information about the presentation template, see 4.13, “Presentation templates” on page 115.

As a result, you can rapidly change the appearence of your content without editing the same elements repeatedly. You can use the Web Content Viewer Portlet to render content from Web Content Manager.

Many capabilities are included to retrieve content from external sources. You can use Web Content Integrator, REST services for IBM Web Content Manager, IBM Digital Data Connector (DDC) for WebSphere Portal, specialized portlets, IBM Web Services for Remote Portlets, and other mechanisms to retrieve and render content from sources outside your digital experience.

Figure 4-50 shows an example of content in the Knowledge Center in the Greenwheels digital experience demonstration that is rendered by the Web Content Viewer portlet.



Knowledge Center >> Expert Advice >> Fitting Your Roadbike

Fitting Your Roadbike

By: Jim Lendler

April 28, 2014

Body

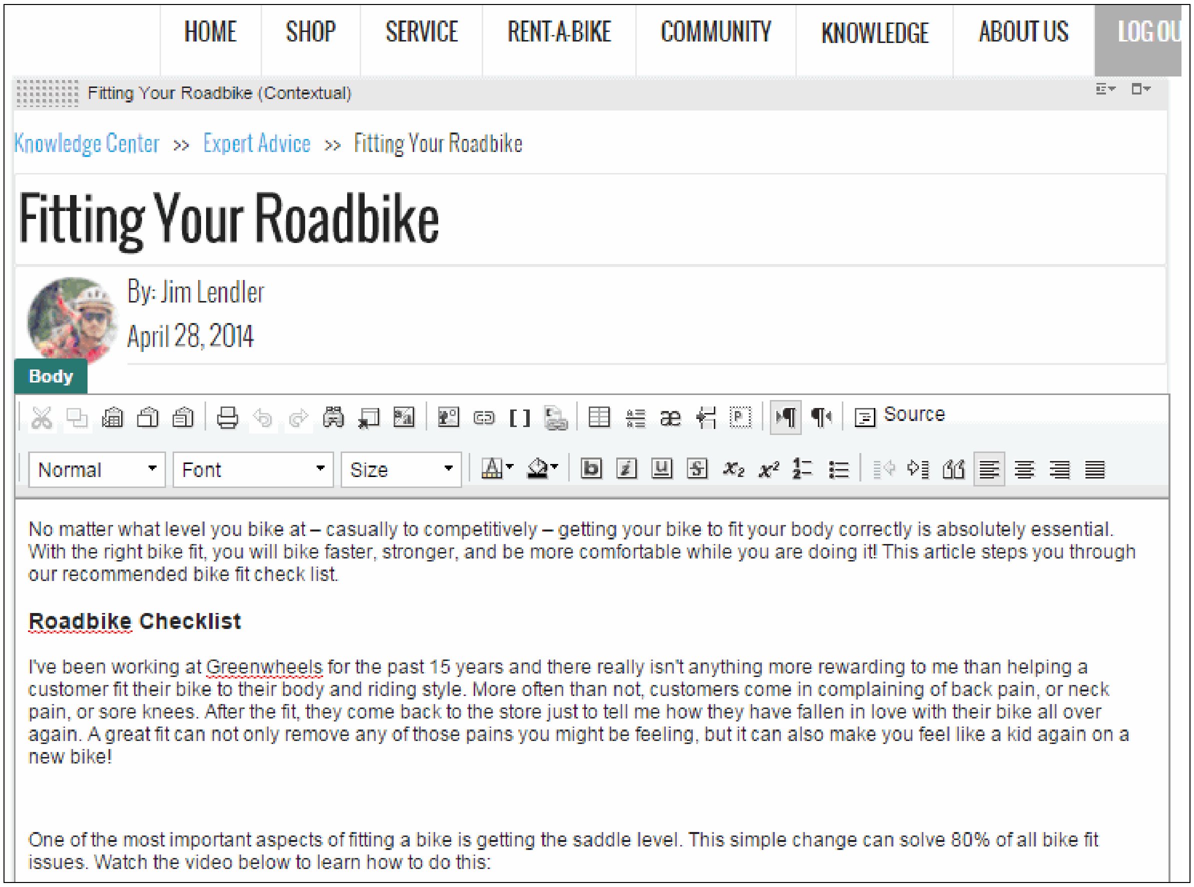
No matter what level you bike at - casually to competitively - getting your bike to fit your body correctly is absolutely essential. With the right bike fit, you will bike faster, stronger, and be more comfortable while you are doing it! This article steps you through our recommended bike fit check list.

Roadbike Checklist

I've been working at Greenwheels for the past 15 years and there really isn't anything more rewarding to me than helping a customer fit their bike to their body and riding style. More often than not, customers come in complaining of back pain, or neck pain, or sore knees. After the fit, they come back to the store just to tell me how they have fallen in love with their bike all over again. A great fit can not only remove any of those pains you might be feeling, but it can also make you feel like a kid again on a new bike!

Figure 4-50 Knowledge Center content

Inline editing provides users with edit access to a content item to edit that item from within the web page itself instead of using the authoring portlet. This feature is available when you display content with a Web Content Viewer portlet, as shown in Figure 4-51.


HOME
SHOP
SERVICE
RENT-A-BIKE
COMMUNITY
KNOWLEDGE
ABOUT US
LOG OU
Fitting Your Roadbike (Contextual)
Knowledge Center >> Expert Advice >> Fitting Your Roadbike
Body
Source
Normal
Font
Size

Fitting Your Roadbike

By: Jim Lendler

April 28, 2014

No matter what level you bike at – casually to competitively – getting your bike to fit your body correctly is absolutely essential. With the right bike fit, you will bike faster, stronger, and be more comfortable while you are doing it! This article steps you through our recommended bike fit check list.

Roadbike Checklist

I've been working at Greenwheels for the past 15 years and there really isn't anything more rewarding to me than helping a customer fit their bike to their body and riding style. More often than not, customers come in complaining of back pain, or neck pain, or sore knees. After the fit, they come back to the store just to tell me how they have fallen in love with their bike all over again. A great fit can not only remove any of those pains you might be feeling, but it can also make you feel like a kid again on a new bike!

One of the most important aspects of fitting a bike is getting the saddle level. This simple change can solve 80% of all bike fit issues. Watch the video below to learn how to do this:

Figure 4-51 Inline editing example

4.12 Pages and page templates

Content authors use page templates to create quickly pages that are consistent with your site design. They do not have to waste time configuring settings that are probably consistent across your site, such as theme selection, page layout, and more.

Page templates include common page elements and make page creation faster. Page templates depend on themes and layout templates and can have content underneath them.

A page template can include the following elements:

- Portlet entities, including portlet preferences
- Page layout and style
- Theme and skin settings
- Portlet wires for communication with other portlets
- Page parameters
- Page description (all languages)

4.13 Presentation templates

A presentation template is used to define the layout of your web content. Tags are used to determine which properties, elements, or components are displayed. Every content item must be associated with at least one presentation template. The authoring template that is used to create a content item has a default presentation template defined. This default can be overridden by a content author when creating content. A different presentation template can also be used at run time (for example, to render the content differently on mobile devices).

For more information about the different rendering modes that are supported, see the *Rendering modes for web content* topic in the IBM WebSphere Portal V8.5.0 IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/wcm/wcm_rendering_modes.dita

> **Defining presentation for Web Content Viewer portlets:** When you display content with WebSphere Portal, the presentation template defines only the layout of content that is displayed in a Web Content Viewer portlet. The overall page design is determined by which page layout is selected, which portlets are added to the page, and the selected theme.

In the Greenwheels example that is used in this chapter, the presentation template Greenwheels Shop defines the presentation for the Featured Product content that was created by using the greenwheels.shop.advance authoring template (see 4.7.1, “Example: Creating Web Content Manager content” on page 86).

To create the presentation template, complete the following steps:

1. Log in to WebSphere Portal, click the Applications menu icon in the toolbar, and then select **Content**, as shown in Figure 4-5 on page 70.
2. Click **Library Explorer** and select the library where you want to store the presentation template, which is **Greenwheels** in this example.
3. Click **New** and select **Presentation Template**, as shown in Figure 4-52.

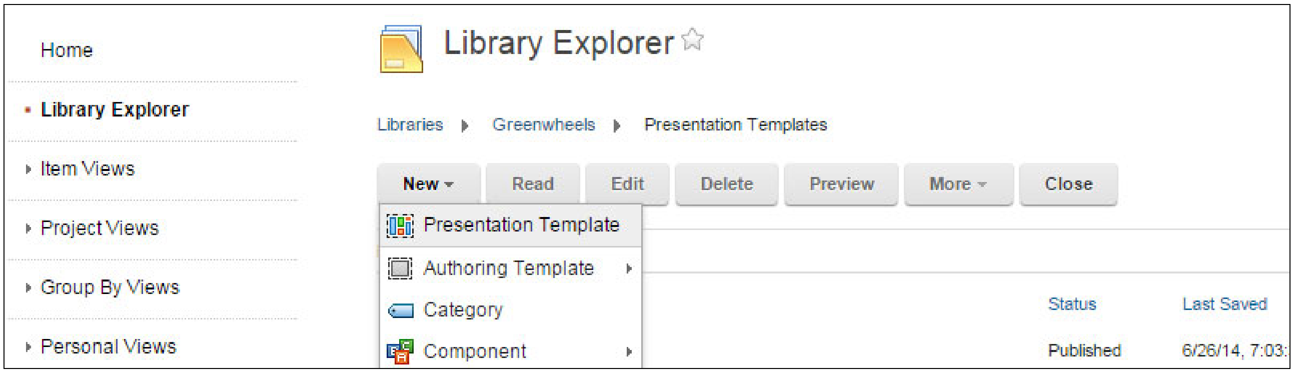
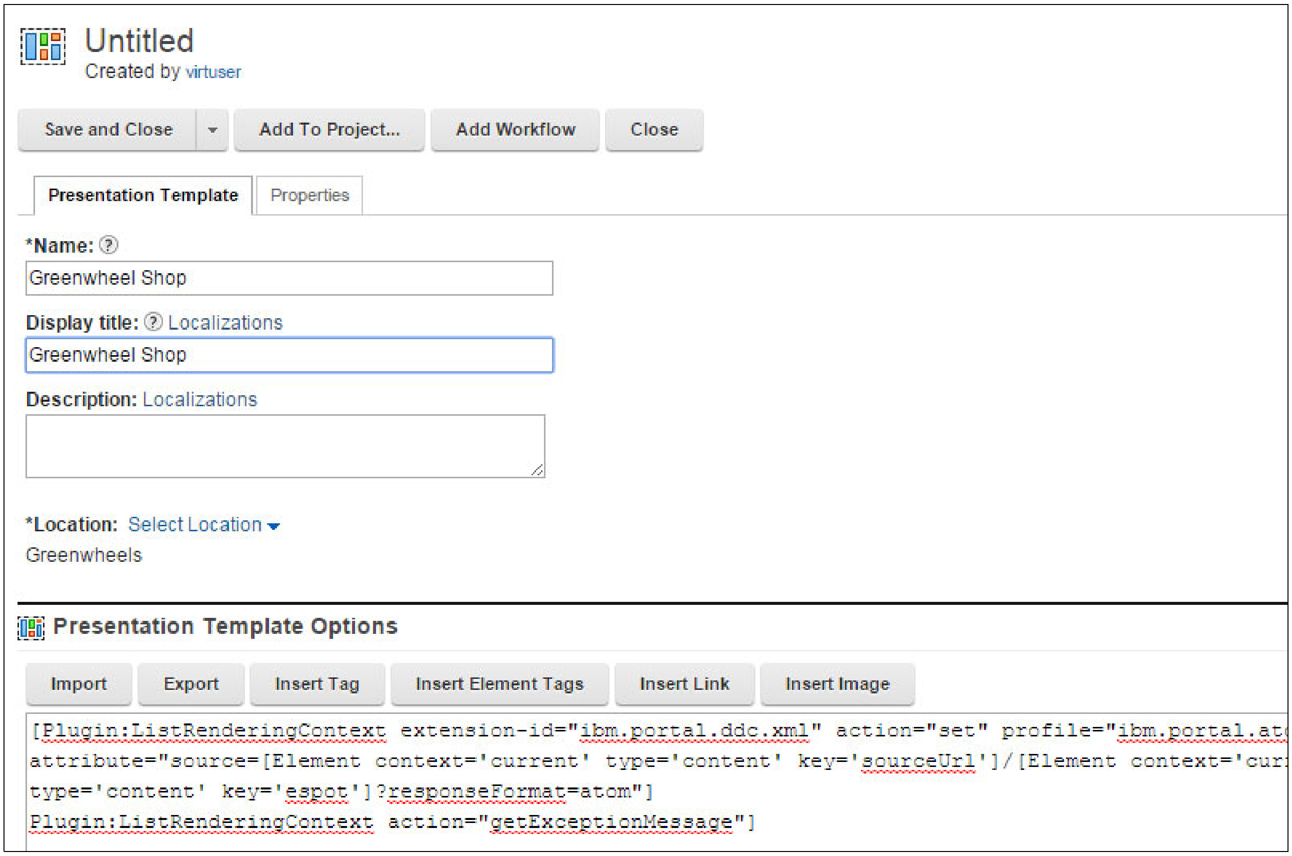


Figure 4-52 Select to create a presentation template

4. In the Presentation Template tab, enter the following information, as shown in Figure 4-53:
   - Presentation template name: Greenwheels Shop
   - Display title: Greenwheels Shop



*Name: Greenwheel Shop

Display title: Localizations Greenwheel Shop

Description: Localizations

*Location: Select Location Greenwheels

```
[Plugin:ListRenderingContext extension-id="ibm.portal.ddc.xml" action="set" profile="ibm.portal.at
attribute="source=[Element context='current' type='content' key='sourceUrl']/[Element context='cur
type='content' key='espot']?responseFormat=atom"]
Plugin:ListRenderingContext action="getExceptionMessage"]
```

Figure 4-53 Create a presentation template

5. In the Presentation Template Options section, enter the sample code that is shown in Example 4-3. This is the section where you add the markup that determines how the content is rendered in the Portal page.

Example 4-3 Presentation template options

```
[Plugin:ListRenderingContext extension-id="ibm.portal.ddc.xml" action="set"
profile="ibm.portal.atom" attribute="source=[Element context='current' type='content'
key='sourceUrl']/[Element context='current' type='content' key='espot']?responseFormat=atom"]
[Plugin:ListRenderingContext action="getExceptionMessage"]
[Component name="greenwheels/gw.section.title"]
[Element context="current" type="content" key="List Appearance" compute="always"]
[Plugin:PageMode pageMode="EDIT" compute="once"]
<h1>Commerce URL:</h1>
[EditableElement context="current" type="content" key="sourceUrl" mode="inplace"]
  [Element context="current" type="content" key="sourceUrl" ifempty="Enter feed URL here..."]
[/EditableElement]
<h1>e-Spot name:</h1>
[EditableElement context="current" type="content" key="espot" mode="inplace"]
   [Element context="current" type="content" key="espot"]
[/EditableElement]
<h1>Select list appearance:</h1>
[EditableElement context="current" type="content" key="List Appearance" mode="inplace"]
   Select a component
[/EditableElement]
[/Plugin:PageMode]
```

When the page is rendered, the element tags enter the values for `sourceURL` and other elements when the Feature Product content was created.

The Plugin:ListRenderingContext plug-in uses the value to fetch the content from WebSphere Commerce and renders it in the Web Content Viewer portlet.

Use Web Content Manager tags to reference elements within presentation templates and element designs. For more information about the tags that are used in Example 4-3, see the *Creating web content tags* topic in the IBM WebSphere Portal V8.5.0 IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/help/panel_help/wcm_dev_referencing_tags.dita

Rendering plug-ins are referenced by using the plug-in tag. The plug-in tags that are used in Example 4-3 are preinstalled rendering plug-ins. You can create your own custom plug-ins. For more information, see the *Creating a plug-in tag* topic in the IBM WebSphere Portal V8.5.0 IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/help/panel_help/wcm_dev_referencing_plugin.dita

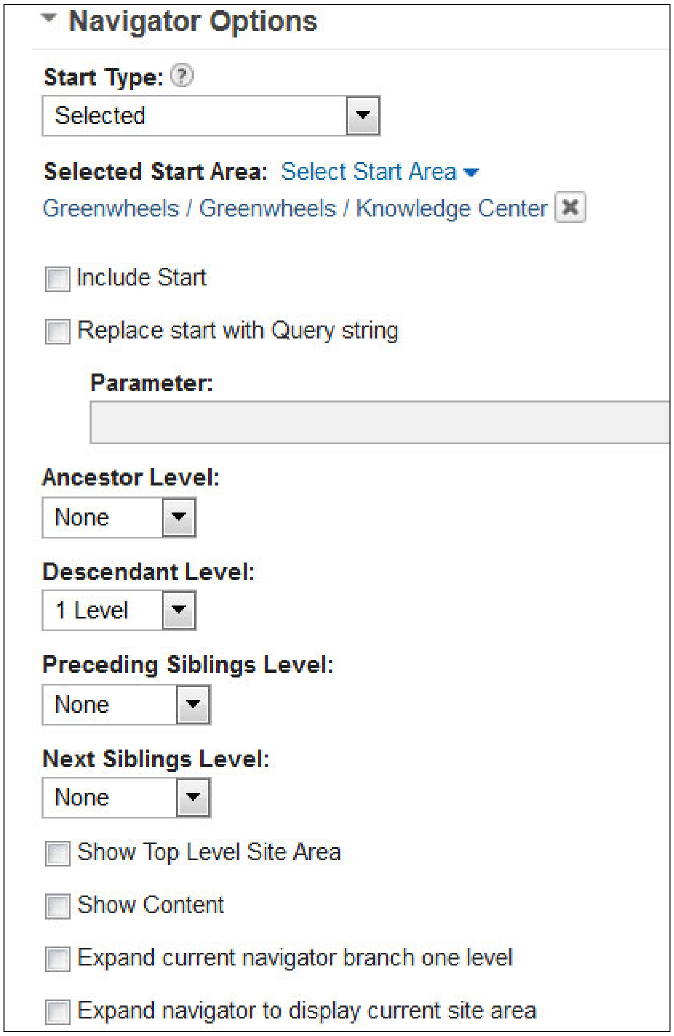
6. Click **Save and Close**. The presentation template is created.

4.13.1 Navigator components

Web Content Manager provides a navigator component that displays a list of site areas in the form of web links. You can use the navigator component to define how the site area parameters query content. The navigator component is identified by a name, title, and description, much like other Web Content Management artifacts.

Figure 4-54 shows the Navigator Options for a sample navigator component. Notice the difference between menu and navigator elements:

- Menu components aggregate content artifacts based on certain predefined criteria, such as common category.
- Navigator components organize content hyperlinks based on the logical site structure.



Navigator Options

Start Type: Selected

Selected Start Area: Select Start Area

Greenwheels / Greenwheels / Knowledge Center

□ Include Start

□ Replace start with Query string

Parameter:

Ancestor Level: None

Descendant Level: 1 Level

Preceding Siblings Level: None

Next Siblings Level: None

□ Show Top Level Site Area

□ Show Content

□ Expand current navigator branch one level

□ Expand navigator to display current site area

Figure 4-54 Navigator component

When you create a navigator component, you can indicate a site area that is the top level in the navigation. You can also specify how many levels down the navigator component renders. Combining these two attributes of a navigator gives you the ability to create a complete site map of your entire site, or a brief navigation of microsite links within the larger site.

> **Note:** Any site area that is rendered in a navigator should have a default content item that is defined so that some content is displayed when the link is clicked.

In the example that is shown in Figure 4-54, the **Selected** element was used for Start Type to specify the particular site area from which to start the navigator. You can also dynamically set a site based on a Query string element. If you aggregate site area links into the navigator component, it is important that a site area has a default content item so that when a user clicks a link, valid content is presented on the web page.

You can use the navigator component to set the number of ancestors, descendants, and siblings that are displayed.

After you define the logical structure of the navigator component, you can add the design options that are used for displaying the results of navigation. For that purpose, you can add HTML code to the Navigator result design field. You can also use the Separator and Header, and Footer fields for adding design that is presented to separate items in a list that are generated by the navigator component and to have elements before and after the generated list. The No result design field of the navigator component is rendered when the component returns an empty list.

Finally, it is important to set access rights for the navigator component. The access rights can be either inherited or set to a specific group, such as User, Contributor, Editor, or Manager.

For more information about the navigator component, see the *Creating a navigator component* topic in the IBM WebSphere Portal V8.5.0 IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/help/panel_help/wcm_dev_elements_navigator_creating.dita?lang=en

4.14 Personalization

Personalization can recognize a specific site visitor based on a profile. It can also determine the characteristics of a user based on previous purchases, products or pages viewed, or other attributes. If a visitor belongs to a particular geographic region, content that is specific to that region can be targeted at the visitor. The page is assembled with the personalized information, and the visitor sees a personalized page.

You can define content through a number of applications, including Web Content Manager. Personalization automatically detects the content definition from these applications.

After you define the content type, attributes of the content are shown to the rule author. The rule author can use these attributes to make conditions. These conditions can define if and when certain content is displayed, or even if certain actions such as database updates and triggered emails might occur.

Here are a few examples of using personalization:

- An intranet portal that displays a page and content based on users roles within an organization and their authorization profile.
- A news portal that renders weather and news information based on a user’s geographical location.
- A shopping website that renders pages according to the browser and device that is used by web user.

For more information about personalization, see the *Personalization* topic in the IBM WebSphere Portal V8.5.0 IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/help/panel_help/oob_pzn.dita?lang=en

4.14.1 Example: Content device personalization

You can use WebSphere Portal V8.5 to target content at users based on personalization rules that can be created by content authors without the need for IT involvement.

Content authors must have the proper access permissions to view or create rules. For more information about the permissions that you must set, see the *Access permissions* topic in the IBM WebSphere Portal V8.5.0 IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/admin-system/sec_acc_rights.dita

Here are the different types of rules that you can create with personalization:

- Select content
- Update
- Profiler (also called segment) binding
- Email
- Recommended content
- Visibility

For more information about these personalization rules, see the *Types of rules* topic in the IBM WebSphere Portal V8.5.0 IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/help/panel_help/pznRulesOverviewHelp.dita?lang=en

This section provides an example about how to use personalization by using the Profiler rule to target content. The following example shows you how to create a rule that determines what device the site visitor is using. That rule is then used to choose specific content that is displayed only on that device.

To implement this example, complete the following steps:

1. Go to the Personalization pages by selecting **Personalization** from the Applications menu in the portal toolbar, as shown in Figure 4-55.

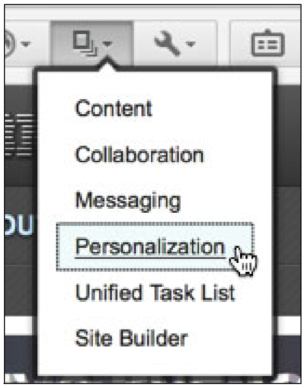


Figure 4-55 Selecting Personalization from the toolbar

2. On the Personalization page, click the **Business Rules** tab, as shown in Figure 4-56, to access the Personalization Navigator.

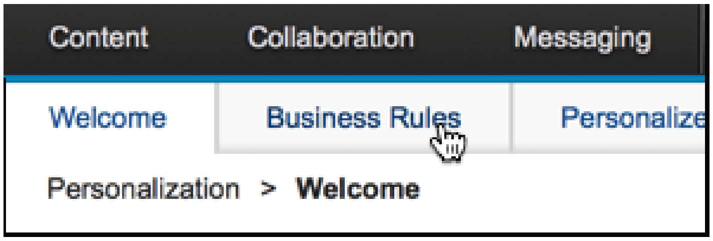


Figure 4-56 Business Rules tab

The Personalization Navigator shows all the personalization rules that you can access. You can create, modify, delete, export, and import rules by using this portlet. From the Personalization Navigator menu, click **New** → **Rule**, as shown in Figure 4-57.

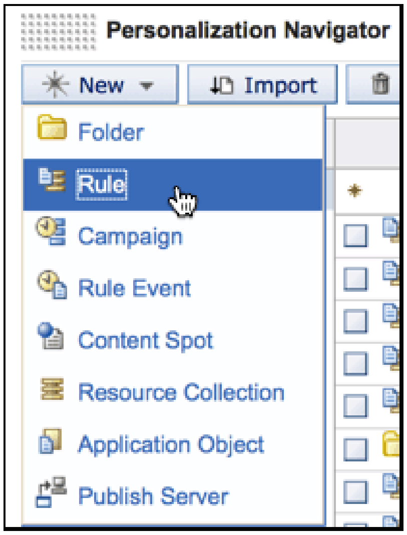
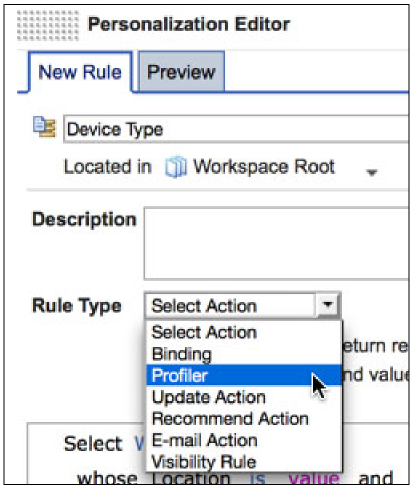


Figure 4-57 Create a business rule

3. On the New Rule window, give the rule a name, such as Device Type. For the Rule Type, select **Profiler**, as shown in Figure 4-58.



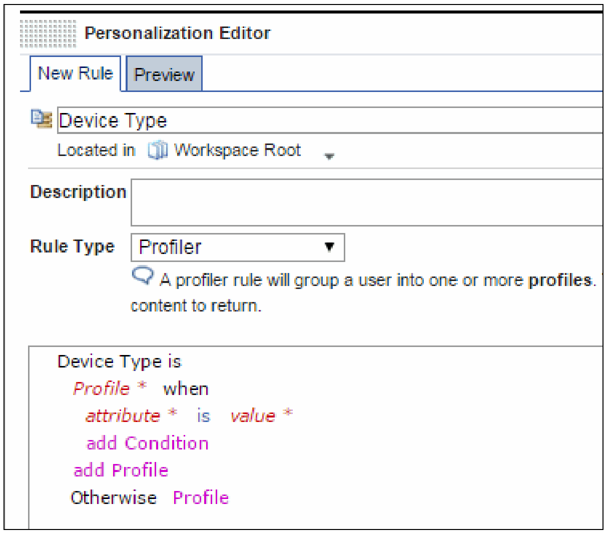
Device Type

Located in Workspace Root

Rule Type: Select Action

Figure 4-58 Select a rule type of Profiler

You now see an interface like the one that is shown in Figure 4-59, which site owners can use to create rules by using an English-like sentence structure. This interface makes it easy to create powerful rules that can be reused throughout the site without involving IT. The business rule is ready for you to edit; it contains links that are placeholders for the values that you provide. You enter some of the values into fields, and others you select from a list in the following steps.



Device Type

Located in Workspace Root

Rule Type: Profiler

A profiler rule will group a user into one or more **profiles**. content to return.

```
Device Type is
  Profile *   when
    attribute *   is   value *
    add Condition
  add Profile
  Otherwise  Profile
```

Figure 4-59 Personalization editor - Profiler rule

4. On line 2, click ***Profile ****. A field appears. Enter the value Mobile Device and click **Submit**.

5. On line 3, click ***attribute ****. In the menu that appears, select **Device** → **Device Class**, as shown in Figure 4-60.

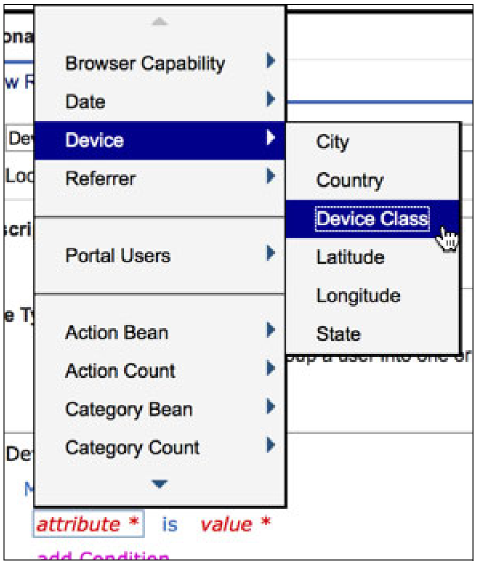


Figure 4-60 Select an attribute of Device Class

6. Still on line 3, click ***value ****. In the menu that appears, select **smartphone**, as shown in Figure 4-61.

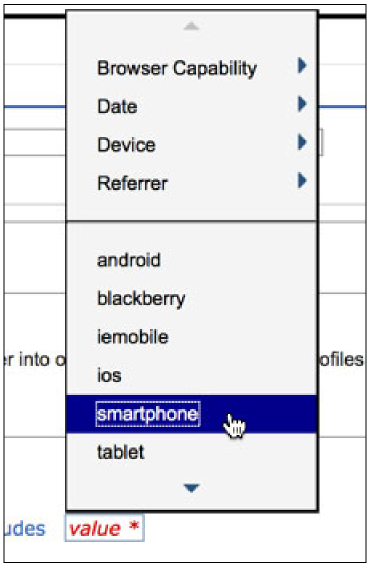


Figure 4-61 Select the smartphone device type

You created a rule that correctly identifies a smartphone, as determined by a set of profiles that are included in the portal. IBM updates these profiles with each release of the product. You can also add to or modify these profiles. The following steps enhance this rule to add tablets to the smartphone that was added.

7. On line 4, click **add Condition**. From the menu that appears, select **Device** → **Device Class**, as you did in step 5.
8. Still on line 4, click ***value **** and select **tablet**, which is similar to step 6.

9. You must make one change before saving the rule. The default behavior when you add multiple conditions is to *and* them together. In this example, the wanted behavior is to *or* them. On line 3, click **and** and change it to **or**, as shown in Figure 4-62.

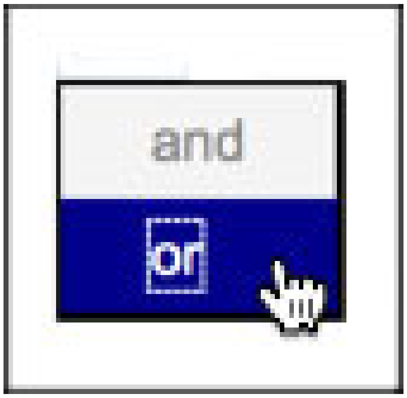
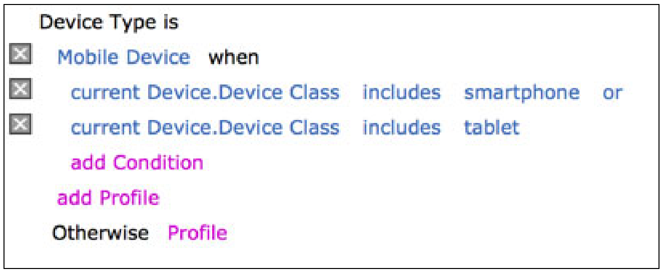


Figure 4-62 Change the condition to "or"

10. Your rule should look like the one that is shown in Figure 4-63. Click **Save**.



Device Type is
☒ Mobile Device when
☒ current Device.Device Class includes smartphone or
☒ current Device.Device Class includes tablet
add Condition
add Profile
Otherwise Profile

Figure 4-63 The completed rule

This rule can now be applied to pages, portlets, or content within portlets. You use this rule to change the content that a portlet displays by using a technique called *content targeting*.

In this example, the site has an image map that the administrator wants to replace with a promotion of their mobile app. Desktop browsers should still see the image map, and mobile browsers should see the mobile app promotion.

To target the correct content to the relevant devices, complete the following steps:

1. Go to the page that has the content you want to target, and put the page into edit mode.

   The portlet that contains the content has a content menu, which is left of the two menu icons at the upper right of the portlet. Click this menu and select **Configure Spot**, as shown in Figure 4-64.

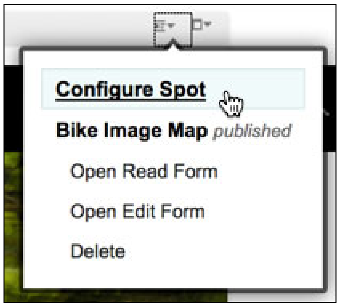
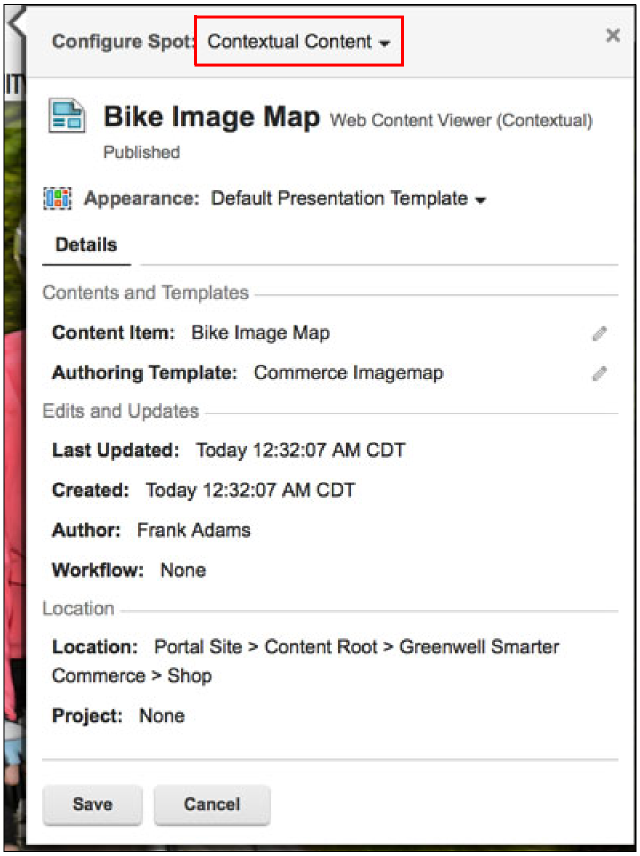


Figure 4-64 Content menu options

2. A dialog box opens. It shows that the spot is configured for Contextual Content, as shown in Figure 4-65.



Configure Spot: Contextual Content

Bike Image Map Web Content Viewer (Contextual)

Published

Appearance: Default Presentation Template

Details

Contents and Templates

Content Item: Bike Image Map

Authoring Template: Commerce Imagemap

Edits and Updates

Last Updated: Today 12:32:07 AM CDT

Created: Today 12:32:07 AM CDT

Author: Frank Adams

Workflow: None

Location

Location: Portal Site > Content Root > Greenwell Smarter Commerce > Shop

Project: None

Figure 4-65 Configure Spot - Contextual Content

3. Click the **Contextual Content** menu and select **Targeted Content**, as shown in Figure 4-66.

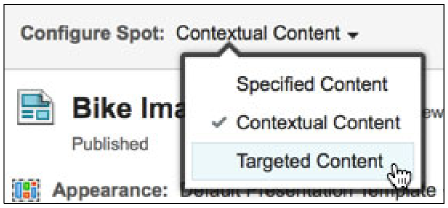

Configure Spot: Contextual Content
Bike Ima
Published
Specified Content
✓ Contextual Content
Targeted Content
Appearance: Default Presentation Template

Figure 4-66 Select Targeted Content

4. The Targeted Content Summary window shows the state of this content spot. Currently, it has only one item defined. Click **Add Content**, as shown in Figure 4-67.

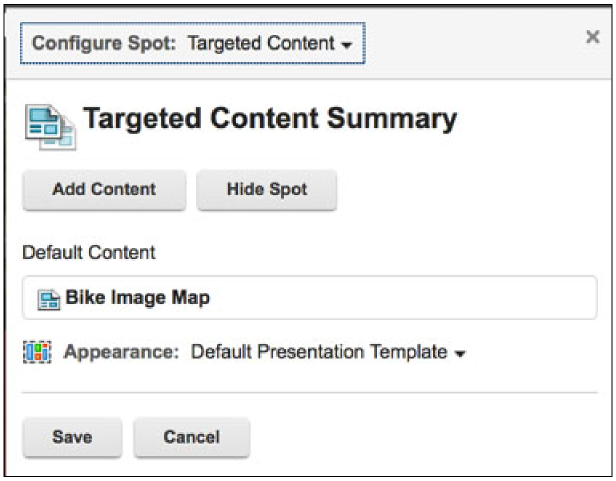
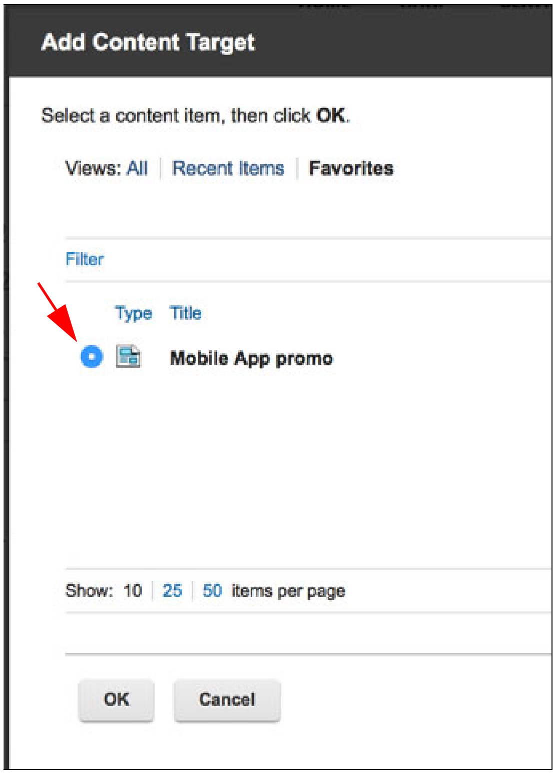


Figure 4-67 Targeted Content Summary

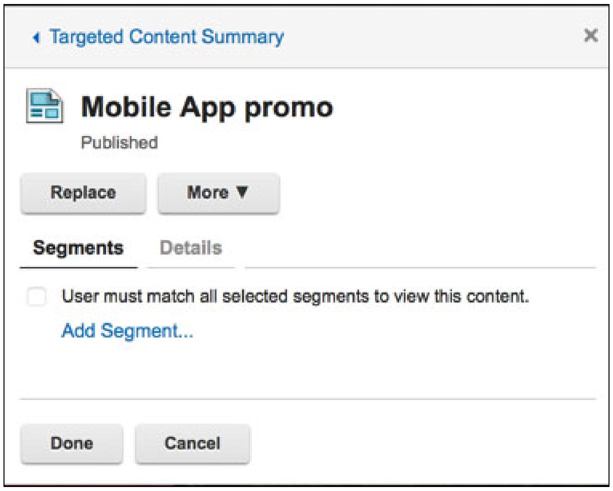
5. A window opens to show all the Web Content Manager libraries that are available to the user who is logged in. Go to an existing content item, select it, and click **OK**. In this example, the Mobile App promotion is selected, as shown in Figure 4-68.


Add Content Target
Views: All | Recent Items | Favorites
Filter
Type Title
Mobile App promo
Show: 10 | 25 | 50 items per page
OK
Cancel

Select a content item, then click **OK**.

Figure 4-68 Select the wanted content item

The Mobile App content should now be selected, as shown in Figure 4-69.



□ User must match all selected segments to view this content.

Figure 4-69 Select the Mobile App promo

6. Click **Add Segment**. A segment is a subset of the visitors to your site, which is filtered by the rule that you select in the following step.

7. You are presented with a list of the Profiler rules on the server. You see only the rules that the logged-in user can view, and the Device Type rule should be among them.

   Click the twistie icon to the left of Device Type to expand it, and click **Add** next to Mobile Device, as shown in Figure 4-70.

   You can select multiple segments, but in this case this segment is the only one needed. Click **Done**.

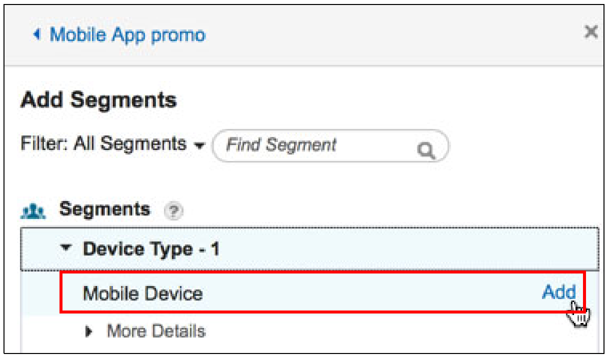


Figure 4-70 Add a segment

8. The content targeting configuration now shows that the Mobile Device segment is selected, as shown in Figure 4-71. Click **Done**.

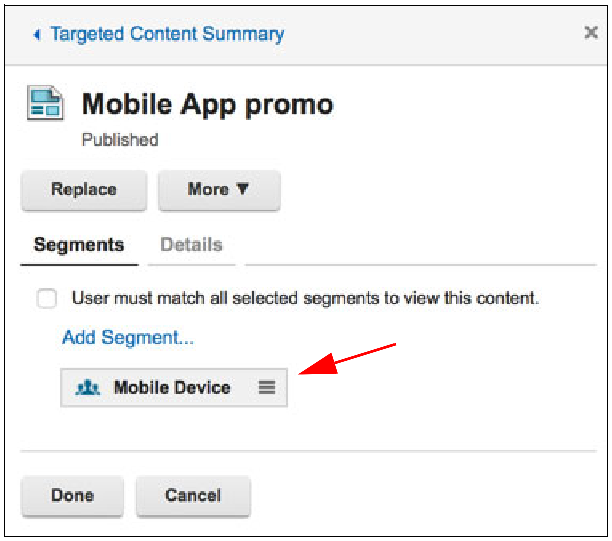


Figure 4-71 Mobile Device segment selected

9. You have now configured this content spot to show the Mobile app promotion to mobile devices and the default content (the image map) to everyone else. Click **Save**.

View the page from desktop and mobile browsers, and observe that the content is displayed appropriately. Figure 4-72 shows the content that is rendered by a desktop browser.

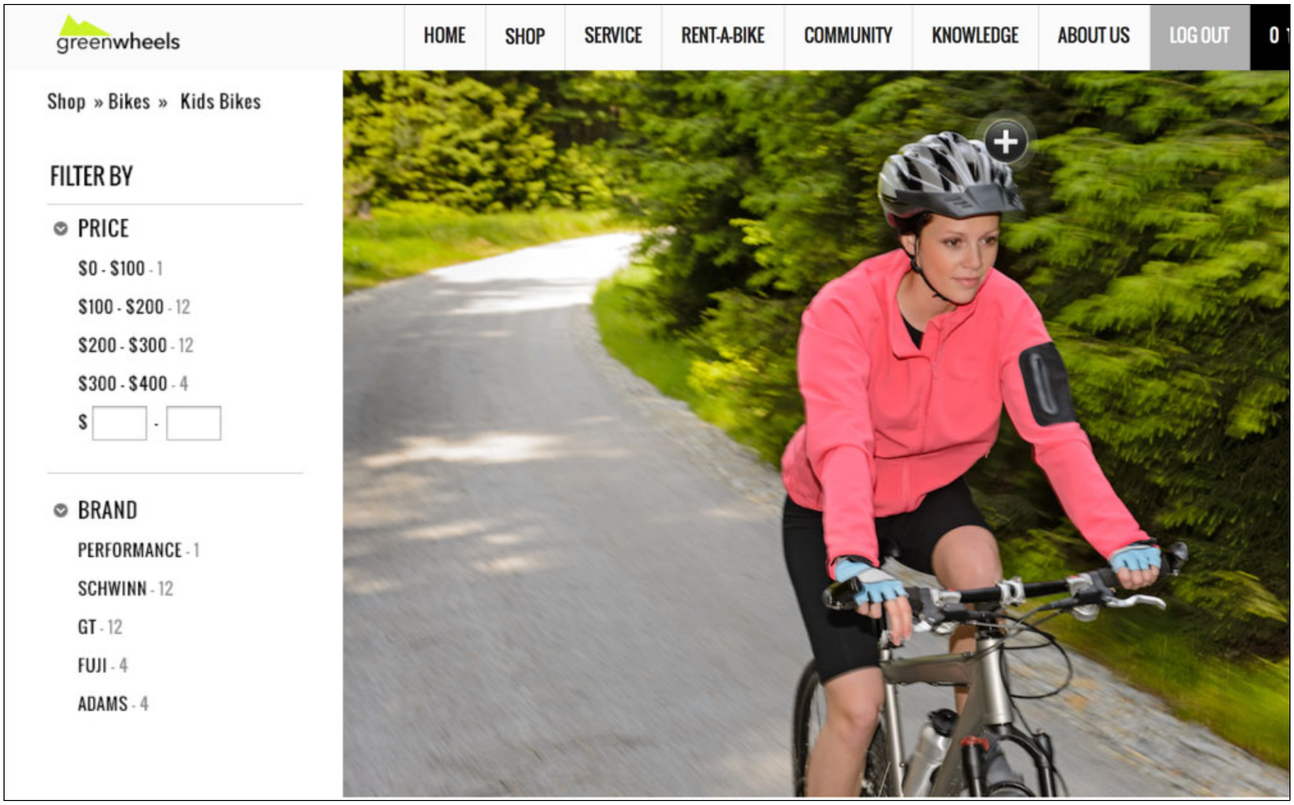
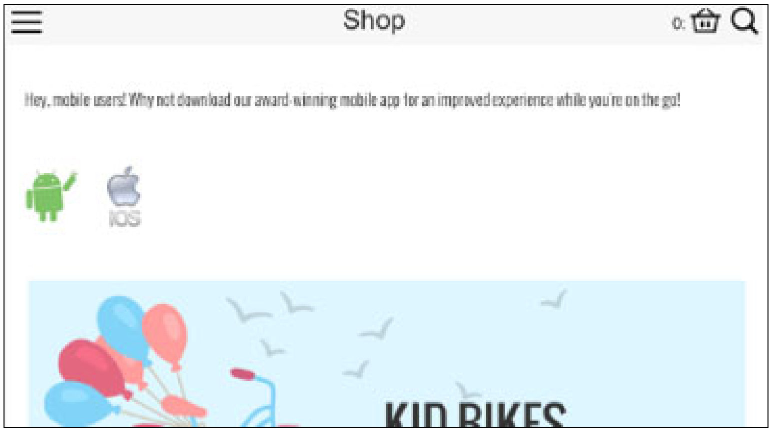


Figure 4-72 The page as rendered by a desktop browser

Figure 4-73 shows the view from a mobile browser.



Hey, mobile users! Why not download our award-winning mobile app for an improved experience while you're on the go!

Figure 4-73 The page as rendered by a mobile browser

5

Lightweight development and integration options

This chapter describes the development and integration options in IBM WebSphere Portal with a focus on the tools that do not require programming in Java Platform, Enterprise Edition but use common web technologies such as HTML, JavaScript, cascading stylesheets (CSS), and simplified design user interfaces. These tools and artifacts are used to develop stand-alone web applications and components for your digital experience.

The options that are described in this chapter allow for a more agile design and integration approach and use a more common and less specialized skill set that is available in IT departments and by line-of-business (LOB) professionals and business partners in your organization. These options facilitate a self-service approach that is driven directly through the user interface of IBM WebSphere Portal and not through comprehensive external integrated development environments (IDEs), such as Eclipse or IBM Rational Application Developer.

This chapter introduces the technologies that are used and documents technical use cases that are made available for reuse and repurposing in other environments.

This chapter covers the following topics:

Note: The website that is used in the examples in this chapter is a fictional website and is not accessible.

5.1 Some useful criteria for comparing tools

Integration technologies offer new ways to build and integrate your digital experience website. However, there are some distinct differences and some overlaps in the capabilities and use cases they support.

This chapter starts with a comparison of the tools so that you can select the tools that fit your use case. This section includes tools that are not covered in detail in subsequent sections of this chapter, such as IBM Rational Application Developer and IBM Web Experience Factory.

Here are some useful criteria for evaluating and comparing the tools:

- Skills: What are the skills you have or want to grow in your team, and how do they match the primary skills that are used with the tool?
- Typical use cases: What are your typical use case requirements, and how do they match the optimum use cases for the tool?
- Deployment: Do you prefer to deploy applications by putting Java Platform, Enterprise Edition code on the server, or by updating a content repository or database?

5.1.1 Guidelines for selecting a tool

Figure 5-1 provides general guidance for selecting the appropriate tool based on the distinct capabilities that differentiate the tools.

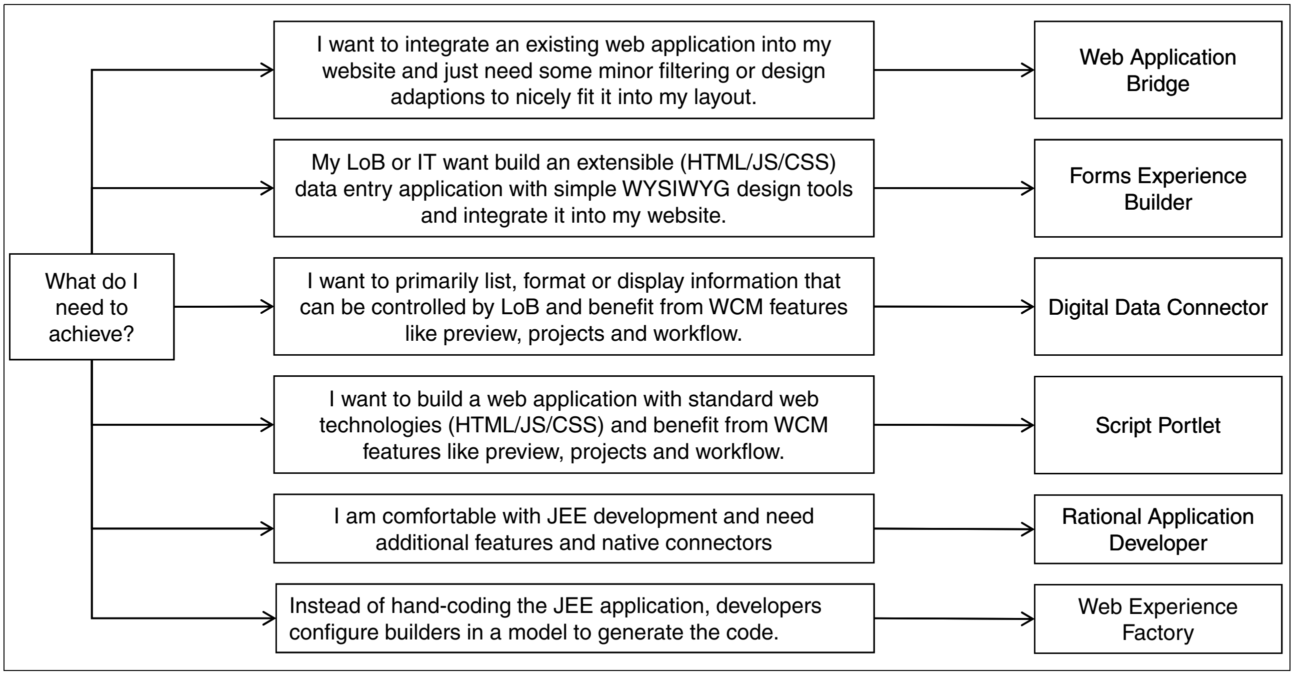

What do I need to achieve?
I want to integrate an existing web application into my website and just need some minor filtering or design adaptions to nicely fit it into my layout.
Web Application Bridge
My LoB or IT want build an extensible (HTML/JS/CSS) data entry application with simple WYSIWYG design tools and integrate it into my website.
Forms Experience Builder
I want to primarily list, format or display information that can be controlled by LoB and benefit from WCM features like preview, projects and workflow.
Digital Data Connector
I want to build a web application with standard web technologies (HTML/JS/CSS) and benefit from WCM features like preview, projects and workflow.
Script Portlet
I am comfortable with JEE development and need additional features and native connectors
Rational Application Developer
Instead of hand-coding the JEE application, developers configure builders in a model to generate the code.
Web Experience Factory

Figure 5-1 Comparison of the different tools that are available

Notice the following points in Figure 5-1 on page 132:

- If you want to integrate an existing web application into your website and need a tool that can apply some content filtering to the markup and provide secure access to your back-end application through a proxy, then the IBM Web Application Bridge is the tool of choice.
- If you need a tool that enables non-technical users to build sophisticated data entry form applications that can easily be extended with readily available web development skills such as HTML, JavaScript, or CSS, then IBM Forms Experience Builder is a good fit.
- If you are primarily listing, formatting, or displaying information, consider the IBM Digital Data Connector (DDC). By using this tool, LOB professionals can configure and manage the integration and take advantage of all the Web Content Manager benefits, such as inline preview, projects, and approval workflows.
- If you are building applications that are more general than those listed previously, consider using the IBM Script portlet because it uses widely available skills (HTML, JavaScript, and CSS) and it has all the benefits of Web Content Manager for management and deployment (in context preview, projects, approval workflow, and so on.)
- If you are most comfortable with Java Platform, Enterprise Edition deployment or if you want to use additional features such as data access connectors, IDEs from IBM such as Web Experience Factory and IBM Rational Application Developer might be the choice for your needs.

5.1.2 Tools comparison

Table 5-1 compares the tools that are described in this chapter.

Table 5-1 Tools comparison

Tool	Primary skills involved	Typical use cases	Primary application code deployment
IBM Web Application Bridge. See 5.3, “Integrating existing web applications with IBM Web Application Bridge” on page 177.	WebSphere Portal administration. Optionally, Java Platform, Enterprise Edition to filter the markup from the external website.	Reusing markup that is generated by an existing web application.	None. Optionally, a JAR filter extension.
IBM Forms Experience Builder. See 5.5, “Integrating Forms Experience Builder” on page 199.	No technical skill required. Optionally, HTML, JavaScript, and CSS skills to extend the application.	Any kind of simple or sophisticated data entry forms with access to a web service infrastructure for back-end data integration.	IBM DB2®.
IBM Digital Data Connector (DDC) See 5.2, “Digital Data Connector” on page 134.	Web Content Manager. Additionally, HTML, JavaScript, and CSS skills to customize the list rendering template.	Lists and details views of external data (social, commerce, search, other data sources) with some update functions. Server-side rendering.	Web Content Manager.

Tool	Primary skills involved	Typical use cases	Primary application code deployment
IBM Script portlet. See 5.4, "Using the IBM Script portlet" on page 192.	HTML, JavaScript, and CSS JavaScript libraries and frameworks. Optionally, some Web Content Manager for application management.	Any type of application user interface that uses data from REST and JSON services. Client-side applications.	Web Content Manager.
IBM Rational Application Developer	Java and Java Platform, Enterprise Edition, HTML, CSS, and JavaScript. Optionally, user interface frameworks such as JSF and Spring.	Any type of application user interface, with server-side data access support provided by Web Experience Factory builders. Client-side or server-side applications or a combination of both with full flexibility.	Server WAR.
IBM Web Experience Factory	Web Experience Factory (builder configuration) and Java, HTML, CSS, and JavaScript for additional customization.	Any type of application user interface, with server-side data access support provided by Web Experience Factory builders. Client-side or server-side applications or a combination of both.	Server WAR.

5.2 Digital Data Connector

The DDC component in WebSphere Portal provides a simplified approach to integrating and interacting with content external to WebSphere Portal. It is already extensively used in rendering social-based content from IBM Connections in what is called *social rendering*. Furthermore, it can be used to integrate many other content sources, such as Forms Experience Builder, IBM Business Process Manager, and IBM WebSphere Commerce.

The following DDC topics are covered in this section:

- 5.2.1, "Overview of the Digital Data Connector" on page 134
- 5.2.2, "High-level architecture the Digital Data Connector" on page 135
- 5.2.3, "Use cases for the Digital Data Connector" on page 138
- 5.2.4, "Use case 1: Social rendering to integrate with IBM Connections" on page 138
- 5.2.5, "Use case 2: Integrating with IBM Business Process Manager" on page 151
- 5.2.6, "Use case 3: Integrating with IBM Forms Experience Builder" on page 163
- 5.2.7, "Use Case 4: Integrating with IBM WebSphere Commerce" on page 175

5.2.1 Overview of the Digital Data Connector

DDC was first introduced as a new component in WebSphere Portal V8.0.0.1 CF11. DDC provides a powerful framework to integrate data from external data sources on your portal pages by using IBM Web Content Manager presentation components. External data is not stored directly in Web Content Manager, but rendered virtually so updates can be immediately seen.

DDC is an extensible framework that provides an open approach to rendering many different types of data sources, including news feeds, task lists, contact information, product catalog information, and many others. By default, XML sources (for example, Atom and RSS) can be integrated with no extra coding. DDC also supports bidirectional operations, meaning that information can be submitted back into the external system, for example, when posting a comment to a blog entry from IBM Connections rendered with DDC.

With DDC, your website designers can use Web Content Manager presentation components to generate the web page markup for your external data. They can use all the Web Content Manager data management facilities for managing the visualization of your external data. These facilities include content syndication, version handling, workflow, and targeting. They can manage the design components in the same way as the other Web Content Manager content and design components.

Here are some of the major benefits of this approach:

- Your Web Content Manager designers can fully control the visual appearance of the integrated data.
- Coding skills are not required for consuming and rendering external data.
- Designers can visualize the external data in the same way in which they visualize data that is stored in Web Content Manager.
- As a result, designers can visualize the external data in a way that is consistent with the corporate design of your overall website by reusing existing Web Content Manager components.
- To adjust quickly existing visualizations of the data or create visualizations for new kinds of external data, you no longer need the help of software developers or the IT department. Your website designers can start working on the presentation templates directly from your portal pages that show the data. Designers use the inline editing capabilities of Web Content Manager.
- Your website designers make updates to the Web Content Manager design components in the project scope. This way, they can keep updates in draft stage until all updates to the project are completed, approved, and published.

You can use DDC in the following ways:

- You can code a Java plug-in, commonly called a *DDC plug-in*, that hooks into the DDC. The plug-in loads the external data and transforms it into a generic DDC data structure called *bean lists*. You can then have the bean lists rendered on your portal pages by using standard Web Content Manager rendering methods.
- You can use the generic XML DDC plug-in that is built into DDC. You can use this plug-in to integrate remote XML data without writing or deploying extra Java code.

  This plug-in supports various parameters that you can use to specify from where the plug-in obtains the XML data and how it transforms the data. The transformation turns a specific XML document format into the generic DDC bean list data structure. With DDC, you can define these transformations in a declarative way so that you can use arbitrary XML formats without having to write transformation code.

5.2.2 High-level architecture the Digital Data Connector

DDC is based on plug-ins. There are two plug-ins, Custom Java and Generic XML. The plug-in integrates with the external data source to load data into a generic DDC data structure called a beans list. This beans list is then consumed by a Web Content Manager presentation template that is then rendered by the Web Content Viewer portlet.

Figure 5-2 illustrates the high-level architecture of the rendering flow within DDC.

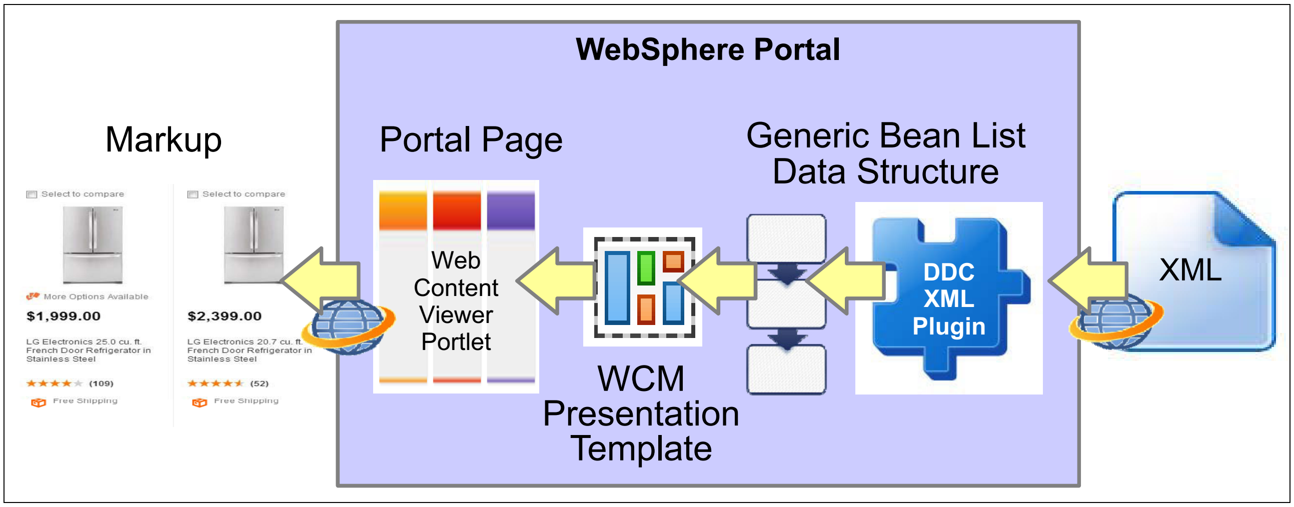

WebSphere Portal
Markup
Select to compare
More Options Available
$1,999.00
LG Electronics 25.0 cu. ft. French Door Refrigerator in Stainless Steel
(109)
Free Shipping
Select to compare
$2,399.00
LG Electronics 20.7 cu. ft. French Door Refrigerator in Stainless Steel
(52)
Free Shipping
Portal Page
Web Content Viewer Portlet
WCM Presentation Template
Generic Bean List Data Structure
DDC XML Plugin
XML

Figure 5-2 Digital Data Connector rendering flow

To integrate external data into WebSphere Portal, you develop plug-ins that hook into the DDC framework. To do so, you have the following alternatives:

- If your external data is available in XML format, you can use the generic XML DDC plug-in to integrate your data. In this case, you do not have to write any code. Instead, you integrate data that comes from a specific XML data source by deploying a list-rendering profile.
- You can also write a DDC plug-in for the WebSphere Application Server extension registry. This approach provides full flexibility but requires creating and deploying Java code.
- You can also use a combination of both approaches. In such a setup, you create a custom DDC plug-in. This custom plug-in delegates the loading and transformation of XML data to the generic XML DDC plug-in.

  You can use the custom plug-in to determine the correct source URL based on the current rendering context. For example, the plug-in can generate a specific product query URL to an e-commerce server. It bases the query on user preferences, request attributes, user agent, and other context information. Furthermore, the custom plug-in can modify the bean list objects that the generic XML DDC plug-in delegate returns. For example, the plug-in can add extra attributes that are computed from the data that is contained in the original bean list object.

The following sections describe the main building blocks of the DDC framework.

Digital Data Connector plug-in

This term refers to Java plug-ins that hook into the DDC framework. DDC plug-ins implement the `com.ibm.portal.wcm.plr.BeanListProvider` Java interface as defined by the public DDC Java API. DDC plug-ins load external data and transform this data into a generic data structure that can be rendered by using list design components.

WebSphere Portal V8.5 comes with two preinstalled DDC plug-ins:

- A DDC plug-in for integrating social data that is made available by remote IBM Connections servers. This DDC plug-in is used by the social lists feature.
- A generic XML DDC plug-in. You can use this provider directly for integrating arbitrary remote XML data. You do not have to write or deploy more Java code.

Bean list

This term refers to the abstract data structure that a DDC plug-in generates based on external data. Individual bean list objects can be transformed into markup by using list appearance components (see "List appearance components" on page 137).

List-rendering profile

A list-rendering profile defines the set of attributes that is available in the beans that are contained in bean lists that a DDC plug-in generates. Furthermore, the generic XML DDC plug-in supports list-rendering profiles to define the mapping between the XML data structure and the actual attribute values. You define this mapping by associating the attribute definitions in the profile with individual XPath expressions.

Digital Data Connector selection rule

The DDC selection rule refers to specific rules that you can create in the WebSphere Portal personalization component or in Web Content Manager personalization components. A DDC selection rule is a personalization rule of type Select Action defined to work on the Pluggable Resources resource collection. You do not have to add any additional properties or selection criteria to this rule. This rule is used to trigger the DDC framework when you render the selection rule in a personalization component. For more information about personalization, see 4.14, "Personalization" on page 119.

List appearance components

List appearance components refer to a Web Content Manager personalization component that contains or references a DDC selection rule. The list appearance component defines the visual design of your list by defining the markup fragments that are generated during rendering of the lists. The individual data fragments that the DDC plug-in loads can be referenced by the list design by using the `[AttributeResource]` tag.

> **Note:** This reference mechanism works the same way as when you render personalization components that select data from the Web Content or the Web Components resource collections. The set of supported `attributeName` parameter values for the `[AttributeResource]` tag depends on the specific DDC plug-in that you use.

List-rendering context

List-rendering context refers to the context that the `[Plugin:ListRenderingContext]` rendering plug-in generates to control the contents of your social lists. This context includes all of the following information:

- The DDC plug-in that you want to be called to build the bean list object from the actual external data.
- The selected list-rendering profile.
- A list of custom attributes.
- Access to the current portlet request and response objects.
- A reference to the currently rendered list definition content item.

The addressed DDC plug-in then evaluates the list-rendering context so that it can query the appropriate set of remote information. You can establish the list-rendering context by adding a `[Plugin:ListRenderingContext]` Web Content Manager tag to your presentation templates. You add this tag before you include the DDC design component.

List definition

The list definition is a Web Content Manager content item. It holds the following items:

- Information that is needed to establish the required list-rendering context.
- A reference to a list appearance component that is responsible for generating a specific visual design for presenting the data that is contained in the resulting bean list object.

5.2.3 Use cases for the Digital Data Connector

As there are many applications and systems that provide a sophisticated web and REST Service API, many use cases are available to use the DDC to integrate with your IT infrastructure. You can then benefit from authoring and update operations through Web Content Manager-generated user interfaces.

Here are some typical DDC use cases:

- Lists and details the display of external data by using a Web Content Manager-driven appearance.
- Social, commerce, search, Content Management Interoperability Services (CMIS) documents, feeds, or any other data sources.
- Server-side rendering.
- REST-based create, read, update, and delete functions.

The following sections describe use cases for sample DDC components that are provided by IBM:

- 5.2.4, “Use case 1: Social rendering to integrate with IBM Connections” on page 138
- 5.2.5, “Use case 2: Integrating with IBM Business Process Manager” on page 151
- 5.2.6, “Use case 3: Integrating with IBM Forms Experience Builder” on page 163
- 5.2.7, “Use Case 4: Integrating with IBM WebSphere Commerce” on page 175

The process of building a custom DDC plug-in is not described in this book. You can further explore this option by accessing the following resources:

- *“Local Files” – Custom DDC Plugin Example available now!* on IBM developerWorks:

  https://developer.ibm.com/digexp/blog/2014/08/04/local-files-custom-ddc-plugin-example-available-now/

- The *Creating and deploying custom Digital Data Connector plug-ins* topic in the IBM WebSphere Portal V8.5.0 IBM Knowledge Center:

  http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/social/plrf_crt_dply_cust_beanlst_prvdr.dita?lang=en

5.2.4 Use case 1: Social rendering to integrate with IBM Connections

Social rendering enables page editors to feature social data that is hosted on a remote IBM Connections server in the context of portal pages.

IBM Connections is a foundation for social business that is about connecting people, information, and processes to provide the foundation for a social business, as shown in Figure 5-3.

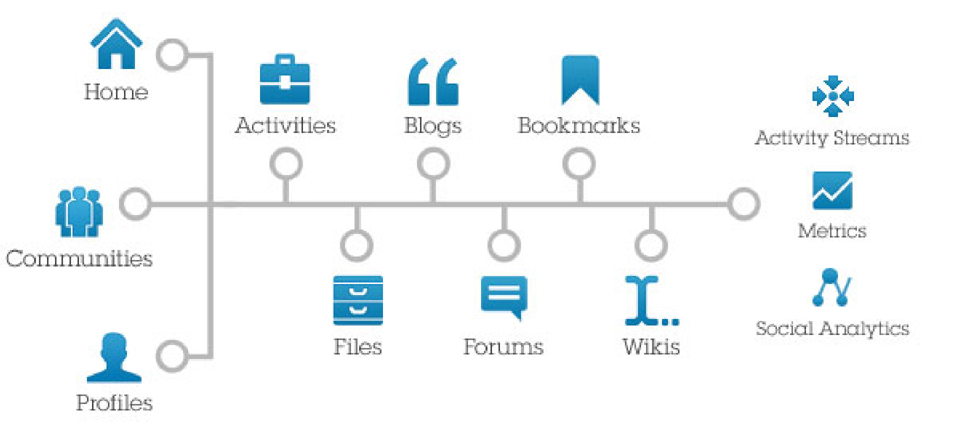


Figure 5-3 Foundation for a social business

IBM Connections provides the following core capabilities. The use of these features is analyzed and aggregated by the included Social Analytics and Metrics functions.

- Profiles: Connect with people that you must collaborate with and add them to your network so that you can build a set of useful contacts. Increase your visibility by updating your profile with information to help people learn about you and your skills.
- Communities: A community provides the means for users to stay in touch, share information, and exchange ideas. Communities provide an excellent way to connect members of a project team, organize a task force researching an emerging technology, or bring together a group of people who share any interest.
- Activities: Activities help your team interact in an online location where ideas about a project and project resources are created, collected, and shared. Add entries and to-do lists to the activity to capture required tasks and store associated files.

  Here are examples of projects that you might track with an activity:

  - Driving a sales process to a close
  - Preparing for an important meeting
  - Writing a report for a client
  - Hiring an employee
- Files: Files enables you and others to upload, organize, share, and collaborate on files easily. It provides an easy way to share files and ideas with people and communities, and a main storage place for your work. Files can be shared, tagged, organized in folders, and accessed from other applications, such as Activities.
- Blogs: Blogs are online journals that you can use to share information within an organization in an efficient, dynamic style. In a business setting, blogs are a great tool for delivering timely information with a personal touch.

  Blogs are a flexible medium for communicating within an organization. Use a blog to inform employees about a new product, or use a blog to inform colleagues about topics that range from experiences at a trade show to tips about using new technology. Blogs are a fast, effective way to share your news and views.

- Forums: A forum is an online discussion board where people can ask questions, share their experiences, and discuss topics of mutual interest. They are an excellent way to create social connections and a sense of community. They can also help you to cultivate an interest group about a particular subject.

  Use the Forums application to start discussions about a specific topic or to debate solutions to shared problems. By participating in a forum, you can exchange ideas, ask questions, and use the expertise of people in your organization.
- Wikis: A wiki is a collection of editable pages about a specific subject and is used by a team to collaborate on content about the subject.
- Bookmarks: Bookmarks are a set of social bookmarking tools that you can use to save, organize, and share internet and intranet bookmarks.

Many organizations have used IBM Connections solutions to reinvent and improve critical business processes by embedding social solutions into their digital experience, as shown in Figure 5-4.

Figure 5-4 Social business processes

The following example use cases or repeatable patterns show how you can improve business processes and provide a measurable return on investment (ROI) when using IBM Connections solutions:

- Customer engagement: The primary business processes for this pattern are sales, customer service, marketing, and ongoing customer relationship management. Organizations wanting to improve customer engagement can use IBM Connections to accomplish the following tasks:
  - Provide a common customer experience to sales, service, and marketing processes and channels (online, contact center, in-person, mobile, and others).
  - Implement collaboration tools for conversing with individual customers and key influencers from customer acquisition and onboarding to aftermarket services.
  - Deploy social analytics and monitoring tools to *listen* to and understand customer sentiment and gain insight into existing and future customer needs.
  - Discuss insights about customers in internal collaborative social systems, and identify, prioritize, and develop new product and service capabilities.

  - Enable rich media experiences that engage customers and aid service and call center preferred practices.
  - Foster the development of customer communities to engage clients and provide additional levels of support.
- Innovation: The primary business processes for this pattern include research and development, product and service management, business strategy, and operations transformation. Organizations wanting to accelerate innovation can use IBM Connections to accomplish the following tasks:
  - Deploy collaboration tools for more open communication and to guide innovation toward delivering the type of value wanted, such as product versus business model.
  - Engage the crowd (internal and external) to vet new ideas.
  - Deploy a portal that combines content, social, and advanced mobile features to provide an exceptional digital experience for customers, business partners, and employees while managing access by role.
- Recruiting and onboarding: The primary business processes for this pattern include hiring, training, and human capital management. Implement collaboration tools for conversing with individual customers and key influencers from customer acquisition and onboarding to aftermarket services.
- Mergers and acquisitions: The primary business processes for this social business pattern are marketing, customer service, sales, onboarding, corporate communications, and talent management. Organizations wanting to improve and streamline their mergers and acquisitions processes can use IBM Connections to accomplish the following tasks:
  - Create and use a social network that consists of senior leadership, HR, and corporate communications to engage in a conversation on the vision and value of the new organization.
  - Establish collaborative spaces for post-merger integration teams to work effectively.
  - For the key areas of expertise and value, use social network analysis to identify the organizational dynamics and engage the *tippers* to create and run plans to integrate and grow the networks.
  - Build focused communities in sales and marketing to maintain customer focus and present a single message.
- Workplace and safety: The primary business processes for this pattern are manufacturing, maintenance, safety, and organizational or citizen communications. Organizations wanting to improve their workplace and safety performance can use IBM Connections to accomplish the following tasks:
  - Speed communication of new or changed safety regulations, policies, and procedures.
  - Minimize or eliminate project execution delays that arise from actual or potential safety issues.
  - Improve innovation in safety procedures by increasing dialog between safety experts and workers.
  - Use existing applications and devices to provide line workers with the latest, relevant safety expertise automatically and contextually.
  - Create communities that connect safety experts with workers so they can share new or revised safety practices.
  - Speed prevention and emergency response with social communications tools that link employees with relevant expertise for on-the-spot decision making.
  - Use social analytics to predict the likelihood of incidents.

  - Use culturally and personality-aligned onboarding and recruitment capabilities to recruit more risk-averse individuals for high accident-propensity roles.
  - Deploy social knowledge management and training capabilities to enable adoption of safer practices and situational awareness of high accident situations.
- Performance acceleration: The primary business processes for this pattern are strategic management, mergers and acquisitions, corporate communications, change management, and all processes that are most impacted by the business priorities of an organization. Organizations wanting to improve their performance compared to their strategic priorities can use IBM Connections to accomplish the following tasks:
  - Build broad social networks that are supported by role-based digital experiences that are aligned to the most critical organizational priorities to optimize performance to the goals.
  - Establish bidirectional, multi-channel communications focused on performance against key priorities, personalized to role and audience (including business partners, supply chains, and customers).
  - Embed the identification of resources with relevant expertise and other related communities, documents, and more into core applications and processes.
  - Create personalized performance scorecards that integrate key data and actions to achieve the strategic priorities.
- Supply chain: The primary business processes for this pattern are manufacturing, risk management, supplier management, logistics, sales, and customer service. Organizations looking to improve supply chain efficiencies can use IBM Connections to accomplish the following tasks:
  - Engage customer communities and use broad social analytics to better forecast demand.
  - Establish broad supply chain communities (in and outside the company) to benchmark and share preferred practices.
  - Improve transparency and risk management with operational communities across tier 1 and associated tier 2 suppliers.
  - Augment logistics with real-time communications to match carrier availability with freight.
  - Branch into social design, including communities that let engineers communicate through the manufacturing cycle with users.
  - Effectively turn engineering inside-out to showcase contributor’s expertise across departments and teams.

Social rendering

Social rendering provides the following components:

- A set of social list definitions.
- Two formatting components for visualizing the social lists in two different ways: a condensed simple list design and a comprehensive list design.

In general, social lists combine a specific query for social data from IBM Connections with a formatting component that determines the look of the query results. The formatting components are Web Content Manager personalization components. The social list definitions are stored as Web Content Manager content items of the type Social Objects Query. This way social list definitions function the same ways as other Web Content Manager content with regards to versions, workflow, and syndication.

Page editors can define social lists. A social list shows the results of a specific query for social data from IBM Connections. For example, the social objects can be lists of blog posts, files, or discussion topics. Page editors can control the visual appearance of the social objects in the query result list. They do so by selecting the formatting component of choice from a predefined, yet extensible set of formatting components.

Site designers provide the formatting components and define them in a style that is consistent with your portal pages. They can also customize and extend this set of formatting components. This way, your page editors do not need to know the markup generation and CSS styling details of the formatting components. Even without that knowledge, they can still choose from a rich but consistent set of visual designs for your lists.

As a result, the page editors can focus on assembling meaningful portal pages that are enriched by social data. They select the appropriate content for the individual portal pages and the social lists. They can use inline editing capabilities to select the visual appearance of the social list. They can also adjust the selection logic of the social list. For example, they can show the most recent blog entries that are created in the IBM Connections community to which the current page is associated. Page editors can select individual social lists from a set of predefined social list definitions and drop them onto portal pages. When a page editor drops a social list onto a portal page, the corresponding content item that represents the social list definition is copied to the page. The page editor can then modify it independently of other social lists on the same page or other pages.

For more information about social rendering, see the *Social rendering* topic in the IBM WebSphere Portal V8.5.0 IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/social/soc_rendr_ovu.dita?lang=en

Adding IBM Connections forum topics to your website

Before you can use social rendering in your IBM WebSphere Portal, you must set up IBM Connections integration. Afterward, your users can start working with the default view definitions that social rendering provides. For more information, see the *Roadmap: How to work with social rendering* topic in the IBM WebSphere Portal V8.5.0 IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/social/soc_rendr_roadmap.dita?lang=en

This section describes how to add forum topic entries from IBM Connections to your website by using the provided Social Rendering components. This section also shows how you can adapt the simple and comprehensive presentation templates to your needs.

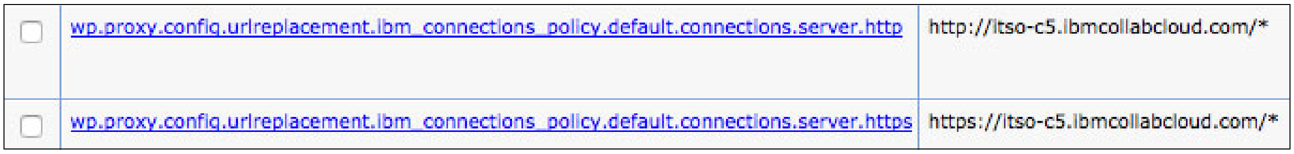
Installing the Social Rendering components consists of downloading the installation package from the IBM Collaboration Solutions Catalog and following the instructions found in its `readme.txt` file. The installation package can be found at the following location:

https://greenhouse.lotus.com/plugins/plugincatalog.nsf/assetDetails.xsp?action=editDocument&documentId=666D72234CA1D57685257CEE0050009A

In WebSphere Portal V8.5, the social rendering and DDC features are enabled by default. You must download and install the IBM Connections portlets, which are a different package than the Social Rendering components. Even if you do not plan to use these portlets, the Social Rendering components depend on the configuration steps that are performed when you install them, so be sure to complete this step.

When your WebSphere Portal Server is configured correctly, you see two entries defined in the WP ConfigService resource provider. To confirm, complete the following steps:

1. Log in to the WebSphere Application Server Integrated Solutions Console.
2. Click **Resources** → **Resource Environment** → **Resource Environment Providers**.
3. From the list, click **WP ConfigService**.
4. Click **Custom Properties**.
5. Confirm that the Connections entries for your server are listed in these properties, as shown in Figure 5-5.



☐	wp.proxy.config.urlreplacement.ibm_connections_policy.default.connections.server.http	http://itso-c5.ibmcollabcloud.com/*
☐	wp.proxy.config.urlreplacement.ibm_connections_policy.default.connections.server.https	https://itso-c5.ibmcollabcloud.com/*

Figure 5-5 Sample Connections configuration entries in WebSphere Portal

The following steps can be performed by a WebSphere Portal administrator or a site designer who has the authority to edit a page:

1. Go to the page where you want to add the IBM Connection forum entries and switch the page to Edit Mode.
2. (Optional) You might want to associate this page with a specific IBM Connections community. With this association in place, the Social Rendering portlets you add to the page can automatically render wikis, blogs, and other content that is associated with that community. To configure this association, run the following steps:
   a. Click the link to open the menu for the page, and select **Open Page Settings**, as shown in Figure 5-6.

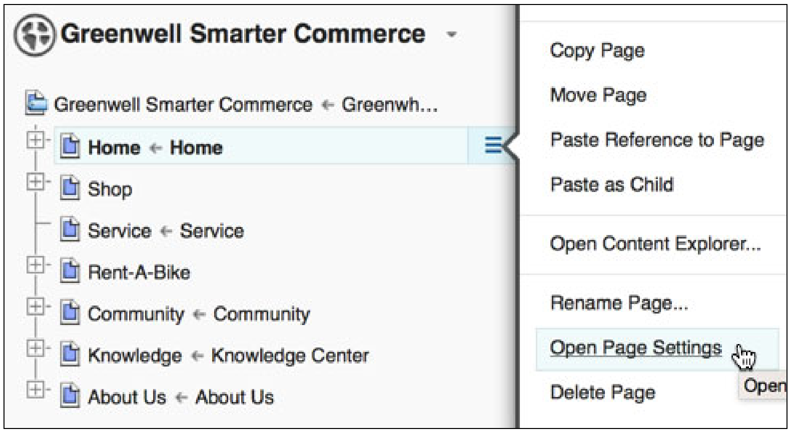
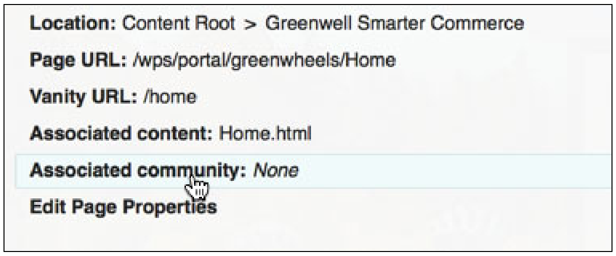


Figure 5-6 Select the Open Page Settings option

b. Click **Associated Community**, as shown in Figure 5-7.



Location: Content Root > Greenwell Smarter Commerce

Page URL: /wps/portal/greenwheels/Home

Vanity URL: /home

Associated content: Home.html

Associated community: *None*

Edit Page Properties

Figure 5-7 Select the Associated community option

c. The window that opens shows "No community association". Click **Select a community**.

d. Select from the list of Communities available to you by selecting the appropriate radio button and then clicking **Select**.

e. Click **OK** to close the window. The Community is now associated with the page.

3. Add the List of Forum Topics portlet to your page:

a. Click the '+' icon to **Add page components and applications** (see Figure 5-8).

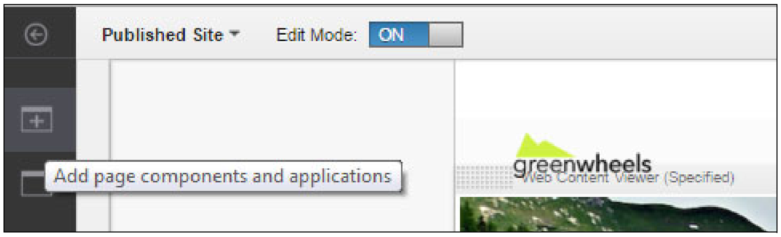

Published Site
Edit Mode:
ON
Add page components and applications
greenwheels
Web Content Viewer (Specified)

Figure 5-8 Add page components

b. Click **Social Content**.

c. Add the **List of Forum Topics** portlet to your page by dragging it or by hovering your cursor over the icon and clicking the '+' sign (see Figure 5-9).

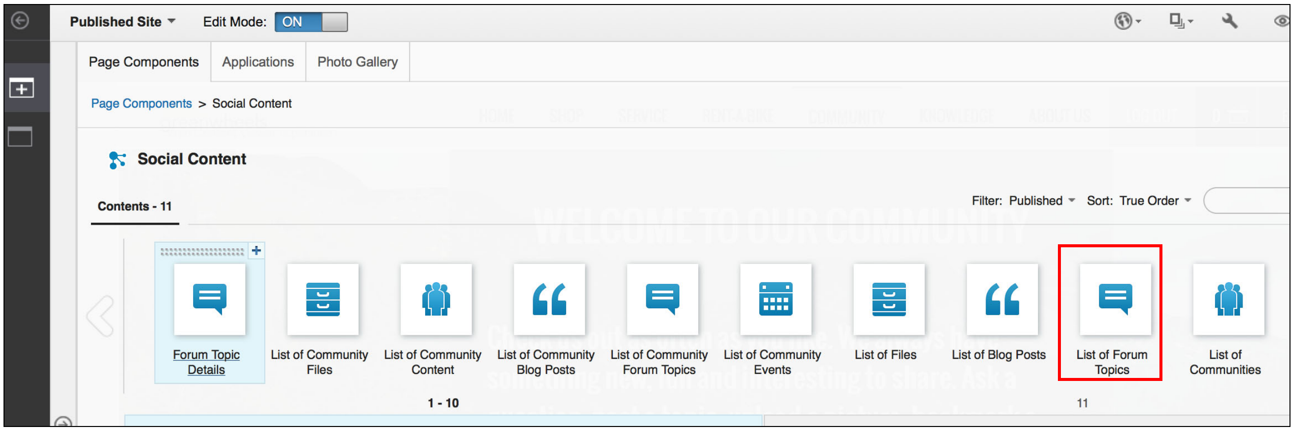


Figure 5-9 Add social content to a page

For more information about adding content to a page, see the following topics in the IBM WebSphere Portal V8.5.0 IBM Knowledge Center:

- *Adding reusable page components to your page*:

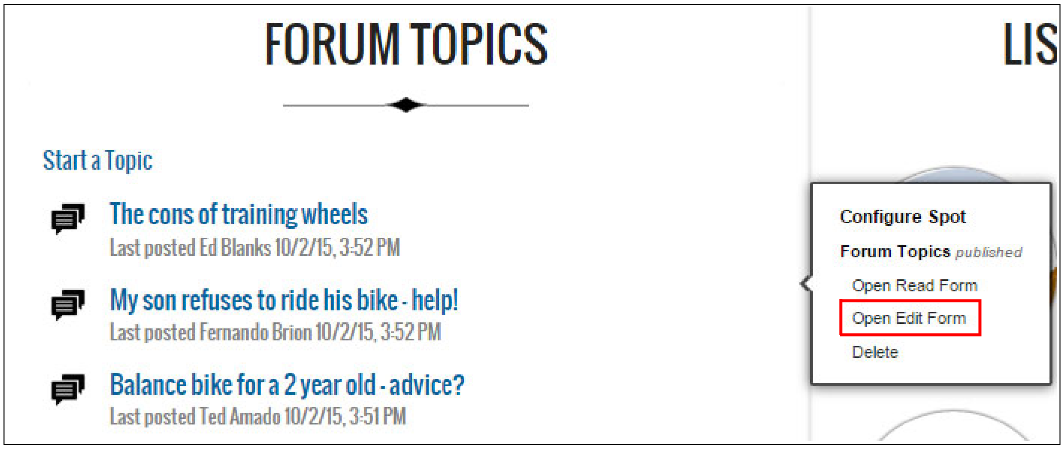
  `http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/help/panel_help/toolbar_add_comp.dita?lang=en`

- *Adding applications to your page*:

  `http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/help/panel_help/toolbar_add_app.dita?lang=en`

4. Edit the properties of the portlet by hovering your cursor over the portlet and selecting **Open edit form** from the menu, as shown in Figure 5-10.

> **Note:** This menu opens when you hover your cursor over that portlet or as a second menu in the portlet, depending on the skin profile you configured for the page.



FORUM TOPICS

Start a Topic

The cons of training wheels
Last posted Ed Blanks 10/2/15, 3:52 PM

My son refuses to ride his bike - help!
Last posted Fernando Brion 10/2/15, 3:52 PM

Balance bike for a 2 year old - advice?
Last posted Ted Amado 10/2/15, 3:51 PM

Figure 5-10 Open the edit form for a portlet

5. Select the content source and filter down to the appropriate Community.

   You can now change the content source to different IBM Connections components. For this use case, Forum Topics is sufficient.

   Limit the contents of the social list to the wanted community that holds the forum entries you want to display on the page, as shown in Figure 5-11 on page 147.

   The following filter options are available under the Filter by Community setting:

   - Limit to the community that is associated with this page.

     If you select this option, the list contains only social objects from the community that is associated with the current page.

   - Limit to selected community.

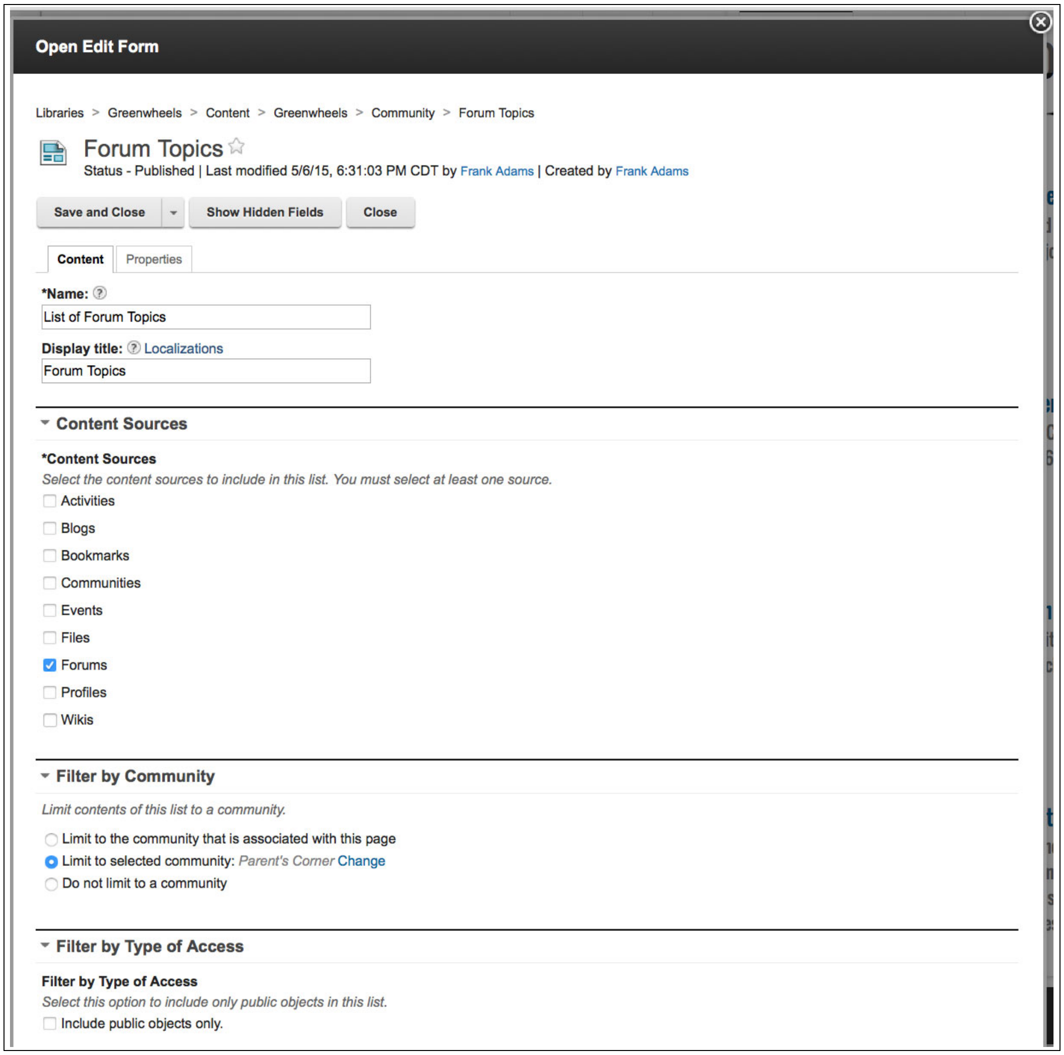
     You can use this option to select a community. If you select a community, the list contains only social objects from the selected community.

   - Do not limit to a community.

     If you select this option, the list is not limited to social objects that belong to communities. Social objects are listed, independent of whether they belong to a community or not.

Note: There are additional filter and order options available that let you optimize the contents to your use case. For more information, see the *Customizing social list definitions by using inline editing* topic in the IBM WebSphere Portal V8.5.0 IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSYJ99_8.5.0/social/soc_rendr_cust_socl_list.html



Open Edit Form

Libraries > Greenwheels > Content > Greenwheels > Community > Forum Topics

Forum Topics

Status - Published | Last modified 5/6/15, 6:31:03 PM CDT by Frank Adams | Created by Frank Adams

*Name: List of Forum Topics

Display title: Localizations Forum Topics

Content Sources

*Content Sources

Select the content sources to include in this list. You must select at least one source.

- □ Activities
- □ Blogs
- □ Bookmarks
- □ Communities
- □ Events
- □ Files
- ☑ Forums
- □ Profiles
- □ Wikis

Filter by Community

Limit contents of this list to a community.

- ○ Limit to the community that is associated with this page
- ● Limit to selected community: *Parent's Corner* Change
- ○ Do not limit to a community

Filter by Type of Access

Filter by Type of Access

Select this option to include only public objects in this list.

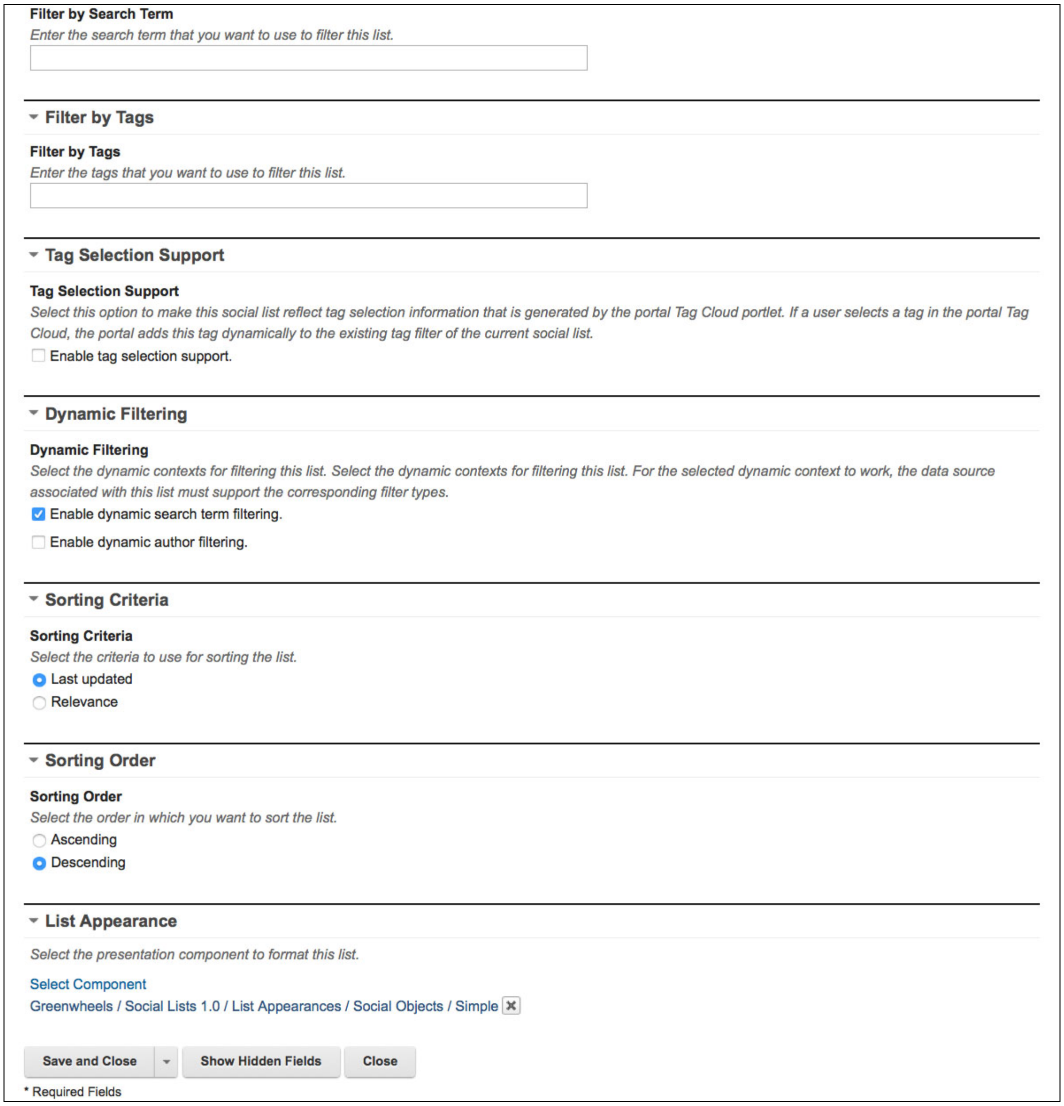
- □ Include public objects only.

Figure 5-11 Configure the social list properties

6. Select the list appearance to render the correct information.

   In the List Appearance settings at the bottom of the Edit Form, you can select the wanted list appearance by selecting the presentation component, as shown in Figure 5-12.

   Social Rendering comes with two generic presentation templates for a simple or comprehensive rendering of social list information. Step 7 shows you how to modify the presentation components.



Filter by Search Term

Enter the search term that you want to use to filter this list.

▾ **Filter by Tags**

Filter by Tags

Enter the tags that you want to use to filter this list.

▾ **Tag Selection Support**

Tag Selection Support

Select this option to make this social list reflect tag selection information that is generated by the portal Tag Cloud portlet. If a user selects a tag in the portal Tag Cloud, the portal adds this tag dynamically to the existing tag filter of the current social list.

☐ Enable tag selection support.

▾ **Dynamic Filtering**

Dynamic Filtering

Select the dynamic contexts for filtering this list. Select the dynamic contexts for filtering this list. For the selected dynamic context to work, the data source associated with this list must support the corresponding filter types.

☑ Enable dynamic search term filtering.

☐ Enable dynamic author filtering.

▾ **Sorting Criteria**

Sorting Criteria

Select the criteria to use for sorting the list.

● Last updated

○ Relevance

▾ **Sorting Order**

Sorting Order

Select the order in which you want to sort the list.

○ Ascending

● Descending

▾ **List Appearance**

Select the presentation component to format this list.

Select Component

Greenwheels / Social Lists 1.0 / List Appearances / Social Objects / Simple ☒

* Required Fields

Figure 5-12 Configure the Social List properties

7. Modify the List Appearance by completing the following steps (see Figure 5-13 on page 149):

   a. Click the List Appearance component. In this example, click **Greenwheels** → **Social Lists 1.0** → **List Appearances** → **Social Objects** → **Simple** to open the presentation component in the Edit Form.

   b. From here, click **[Component name="greenwheels/social lists 1.0** → **component designs** → **simple/type dispatcher entry"]** in the Result design section.

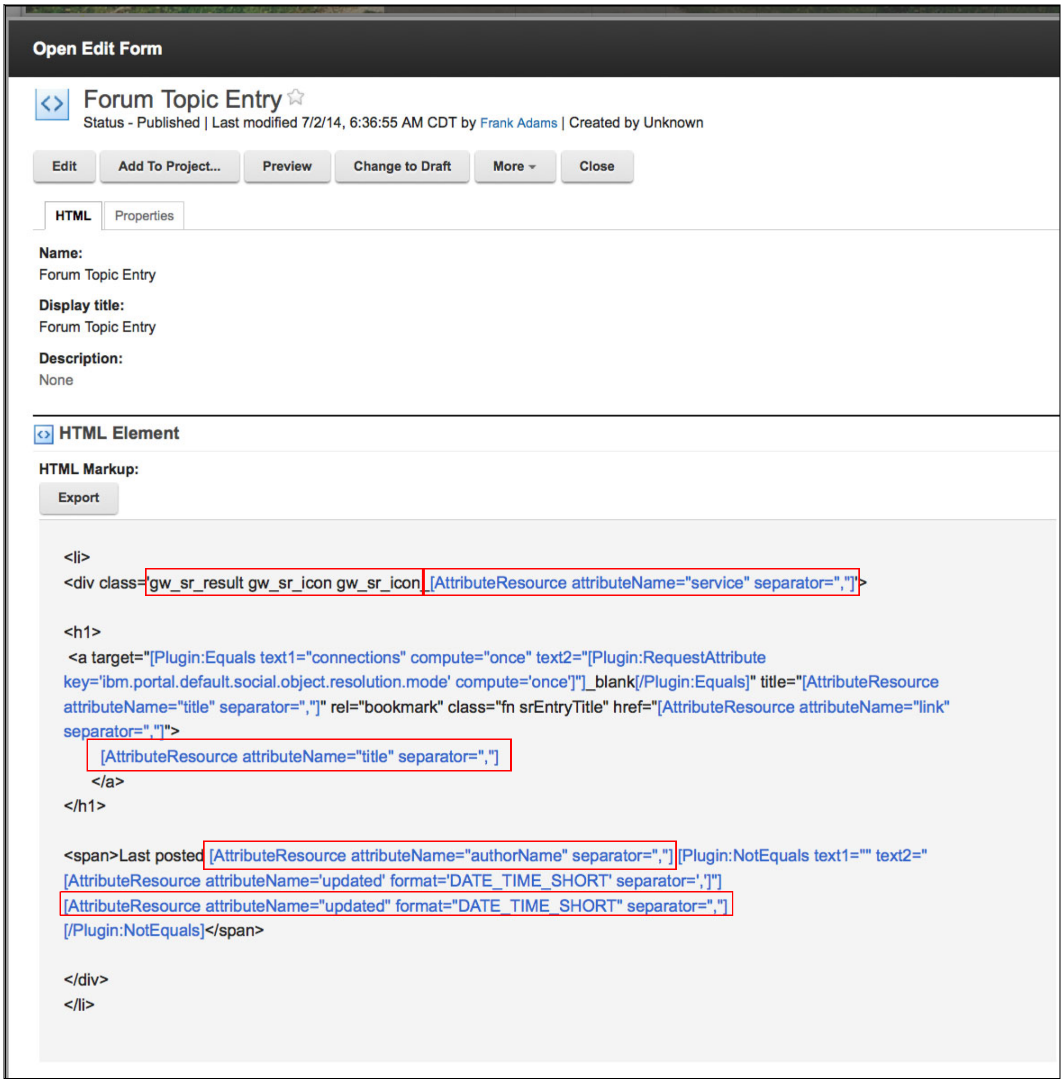
The type dispatcher entry is a Web Content Manager component that links to the correct rendering component, depending on the type of content coming from IBM Connections.

c. Scroll down to the HTML Element section and click **[Component name="greenwheels → social lists 1.0/component designs → simple → forum topic entry"]** to open the Forum Topic Entry component

d. In the Form Topic Entry page, you can find the rendering markup for the forum entries.

e. Find the main elements of the Connections Forum entry in the presentation markup.

The `AttributeResource` tag renders specific attributes from the forum topic entry in IBM Connections into the presentation markup. This way, you can easily modify or enhance the markup to suit your needs. In this example, a `div` tag is added with class references to the presentation HTML component, which then gives you full control over the styling of that list of blog entries. The referenced `div` classes must be added to the overall CSS stylesheet or to, for example, the header section of the list presentation component.



Name:
Forum Topic Entry

Display title:
Forum Topic Entry

Description:
None

HTML Markup:

```
<li>
<div class='gw_sr_result gw_sr_icon gw_sr_icon_[AttributeResource attributeName="service" separator=","]'>

<h1>
 <a target="[Plugin:Equals text1="connections" compute="once" text2="[Plugin:RequestAttribute
key='ibm.portal.default.social.object.resolution.mode' compute='once']"]_blank[/Plugin:Equals]" title="[AttributeResource
attributeName="title" separator=","]" rel="bookmark" class="fn srEntryTitle" href="[AttributeResource attributeName="link"
separator=","]">
     [AttributeResource attributeName="title" separator=","]
    </a>
</h1>

<span>Last posted [AttributeResource attributeName="authorName" separator=","] [Plugin:NotEquals text1="" text2="
[AttributeResource attributeName='updated' format='DATE_TIME_SHORT' separator=',']"]
[AttributeResource attributeName="updated" format="DATE_TIME_SHORT" separator=","]
[/Plugin:NotEquals]</span>

</div>
</li>
```

Figure 5-13   Forum topic entry rendering component

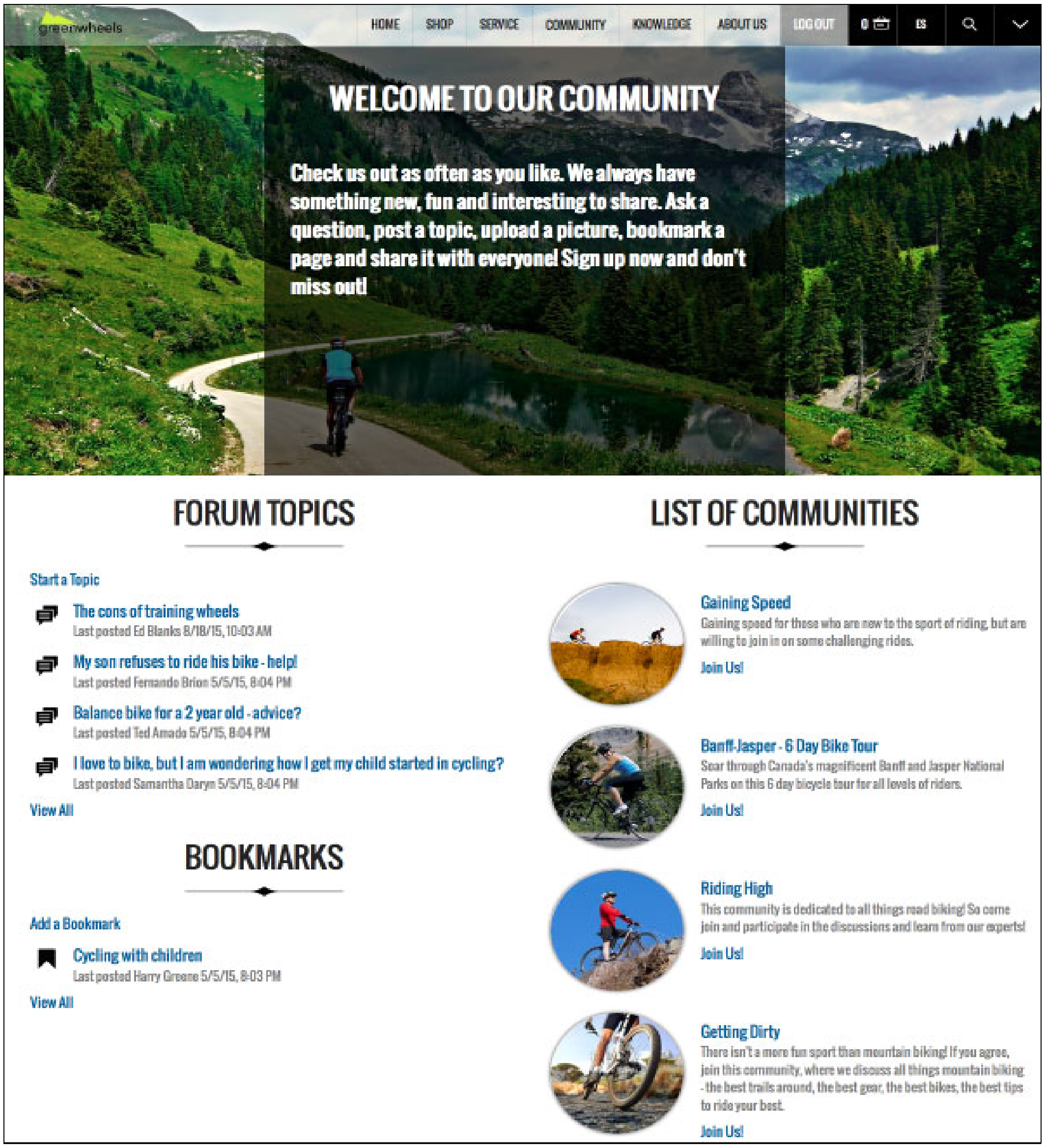
Table 5-2 provides a description of the key elements rendered into this presentation template.

Table 5-2 HTML markup of the list presentation template for blog entries

HTML / IBM Web Content Manager tag	Referenced elements
`<div class='gw_sr_result gw_sr_icon></div>`	`Div` classes in CSS to customize the appearance of the forum topic list to the overall website design.
`[AttributeResource attributeName="service" separator=","]`	The service type of IBM Connections that is returned to display an appropriate icon.
`[AttributeResource attributeName="title" separator=","]`	The title of the forum entry title in IBM Connections.
`[AttributeResource attributeName="authorName" separator=","]`	The name of the author who created this forum entry.
`[AttributeResource attributeName="updated" format=”DATE_TIME_SHORT” separator=","]`	The updated time stamp of the forum entry in the wanted format.

8. Click **Edit** to open the form in edit mode and modify it by using the Insert Tag dialog. Use the Insert Tag dialog to add metadata from the IBM Connections Forum to the presentation component to be shown in your website.

9. Switch off Edit Mode on the page to find the forum topic entry presented on the current page, as shown in Figure 5-14.



WELCOME TO OUR COMMUNITY

Check us out as often as you like. We always have something new, fun and interesting to share. Ask a question, post a topic, upload a picture, bookmark a page and share it with everyone! Sign up now and don't miss out!

FORUM TOPICS

Start a Topic

The cons of training wheels
Last posted Ed Blanks 8/18/15, 10:03 AM

My son refuses to ride his bike - help!
Last posted Fernando Brion 5/5/15, 8:04 PM

Balance bike for a 2 year old - advice?
Last posted Ted Amado 5/5/15, 8:04 PM

I love to bike, but I am wondering how I get my child started in cycling?
Last posted Samantha Daryn 5/5/15, 8:04 PM

View All

BOOKMARKS

Add a Bookmark

Cycling with children
Last posted Harry Greene 5/5/15, 8:03 PM

View All

LIST OF COMMUNITIES

Gaining Speed
Gaining speed for those who are new to the sport of riding, but are willing to join in on some challenging rides.
Join Us!

Banff-Jasper - 6 Day Bike Tour
Soar through Canada's magnificent Banff and Jasper National Parks on this 6 day bicycle tour for all levels of riders.
Join Us!

Riding High
This community is dedicated to all things road biking! So come join and participate in the discussions and learn from our experts!
Join Us!

Getting Dirty
There isn't a more fun sport than mountain biking! If you agree, join this community, where we discuss all things mountain biking - the best trails around, the best gear, the best bikes, the best tips to ride your best.
Join Us!

Figure 5-14 Website community page with forum topics added

5.2.5 Use case 2: Integrating with IBM Business Process Manager

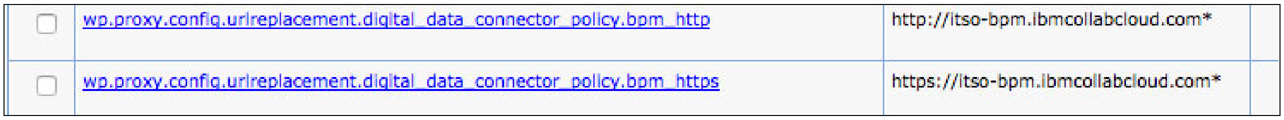
DDC also allows WebSphere Portal to become a front end for IBM Business Process Manager (IBM BPM) pulling data into WebSphere Portal by using the IBM BPM REST APIs. IBM BPM can then behave in a *headless* manner, acting as a data source for WebSphere Portal.

The OpenNTF website provides a sample that contains four content items that you can drag onto your WebSphere Portal pages. The content items allow a user to work with exposed processes, tasks, task details, and coaches stored in the IBM BPM server. These content items are provided as examples of how you can easily integrate WebSphere Portal and IBM BPM by using DDC. For more information, see the *Business Process Manager Sample for IBM Digital Data Connector in WebSphere Portal* topic at the following website:

http://openntf.org/main.nsf/project.xsp?r=project/Business%20Process%20Manager%20Sample%20for%20IBM%20Digital%20Data%20Connector%20in%20WebSphere%20Portal

The instructions that come with the sample document the installation and configuration. Be sure to follow the instructions to configure correctly the Ajax proxy. The components rely on the proxy to function correctly. When your WebSphere Portal Server is configured correctly, you see two entries defined in the WP ConfigService resource provider. To confirm, complete the following steps:

1. Log in to the WebSphere Application Server Integrated Solutions Console.
2. Click **Resources** → **Resource Environment** → **Resource Environment Providers**.
3. From the list, click **WP ConfigService**.
4. Click **Custom Properties**.
5. Confirm that the IBM BPM entries are listed in these properties, as shown in Figure 5-15.



□	wp.proxy.config.urlreplacement.digital_data_connector_policy.bpm_http	http://itso-bpm.ibmcollabcloud.com*	
□	wp.proxy.config.urlreplacement.digital_data_connector_policy.bpm_https	https://itso-bpm.ibmcollabcloud.com*	

Figure 5-15 Proxy entries defined by the installation of the DDC components for IBM BPM

Business process sample

The business process that is used in this example is Vendor Onboarding. A vendor who is already registered in the portal submits a request to sell a new product. After it is submitted, the request goes through an approval cycle by management and executives.

There are three groups of users in this example:

- Greenwheels Vendors
- Greenwheels Management
- Greenwheels Executives

These groups are created in LDAP, so you can assign them to each lane in the process.

The process consists of four lanes:

- Vendors
- Management
- Executives
- System

Figure 5-16 shows the process that is used in this example. This example includes the following steps:

- “Step 1: Submit the selling product” on page 153
- “Step 2: Management review” on page 156
- “Step 3: Executive approvals” on page 158

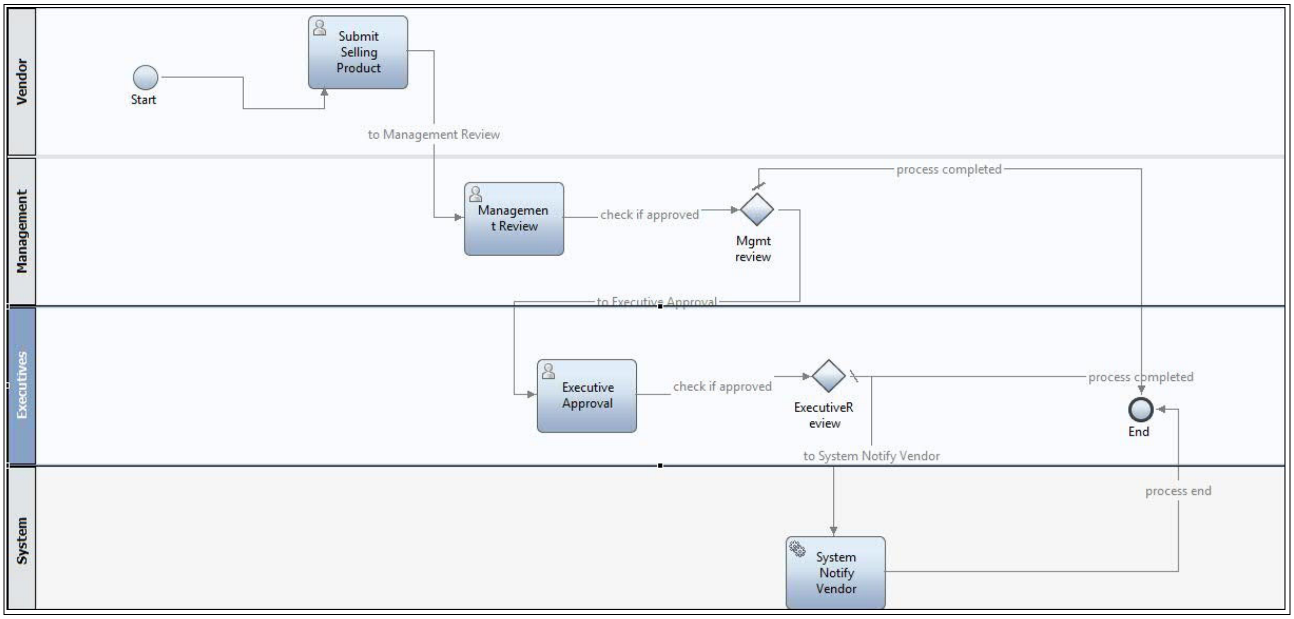


Figure 5-16 The sample IBM BPM process

Table 5-3 shows the LDAP group that is assigned to each lane.

Table 5-3 LDAP group that is assigned to each lane

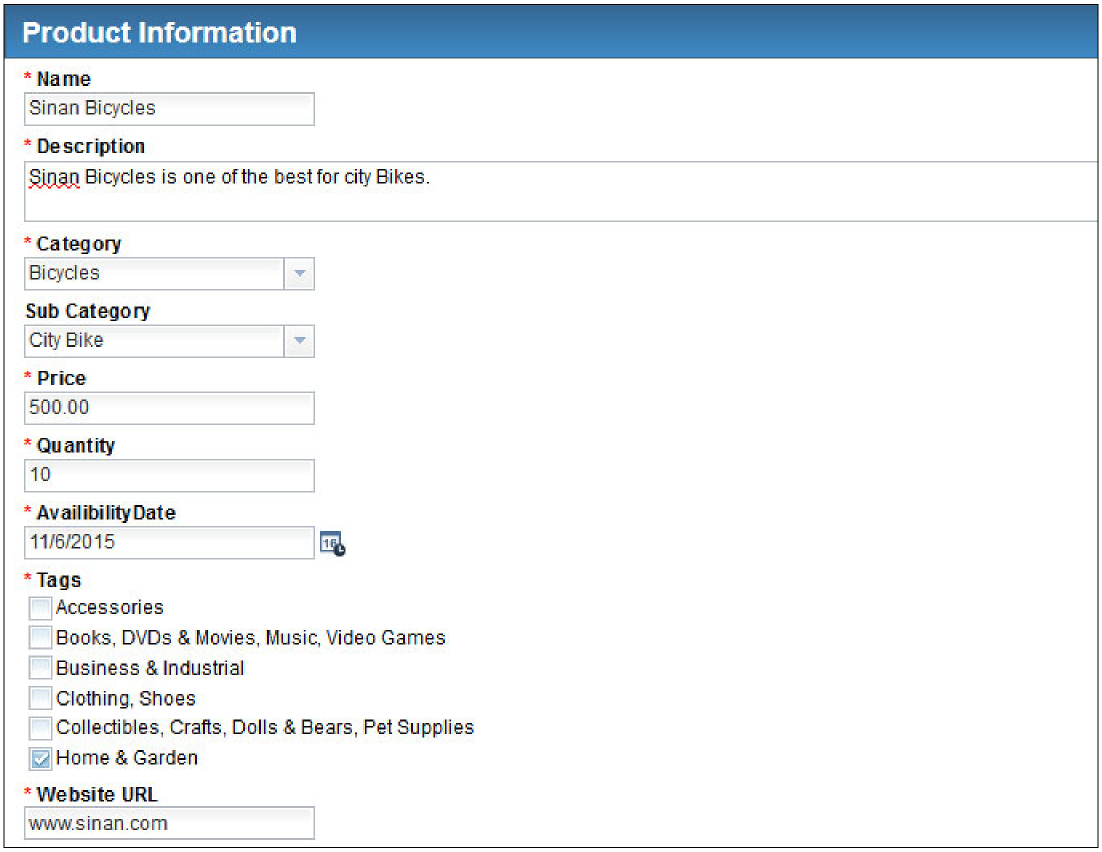
Lane	LDAP group
Vendor	Greenwheels Vendor
Management	Greenwheels Management
Executives	Greenwheels Executives

Step 1: Submit the selling product

In the first step in the process, the vendors submit the product that they are requesting to sell. The business object consists of the following items:

- Product Name
- Product Description
- Category
- Sub Category
- Price
- Quantity
- Available Date
- Product Images

Figure 5-17 represents the first step setup page that is built by using IBM BPM Coach.



Product Information

* Name

Sinan Bicycles

* Description

Sinan Bicycles is one of the best for city Bikes.

* Category

Bicycles

Sub Category

City Bike

* Price

500.00

* Quantity

10

* AvailibilityDate

11/6/2015

* Tags

□ Accessories

□ Books, DVDs & Movies, Music, Video Games

□ Business & Industrial

□ Clothing, Shoes

□ Collectibles, Crafts, Dolls & Bears, Pet Supplies

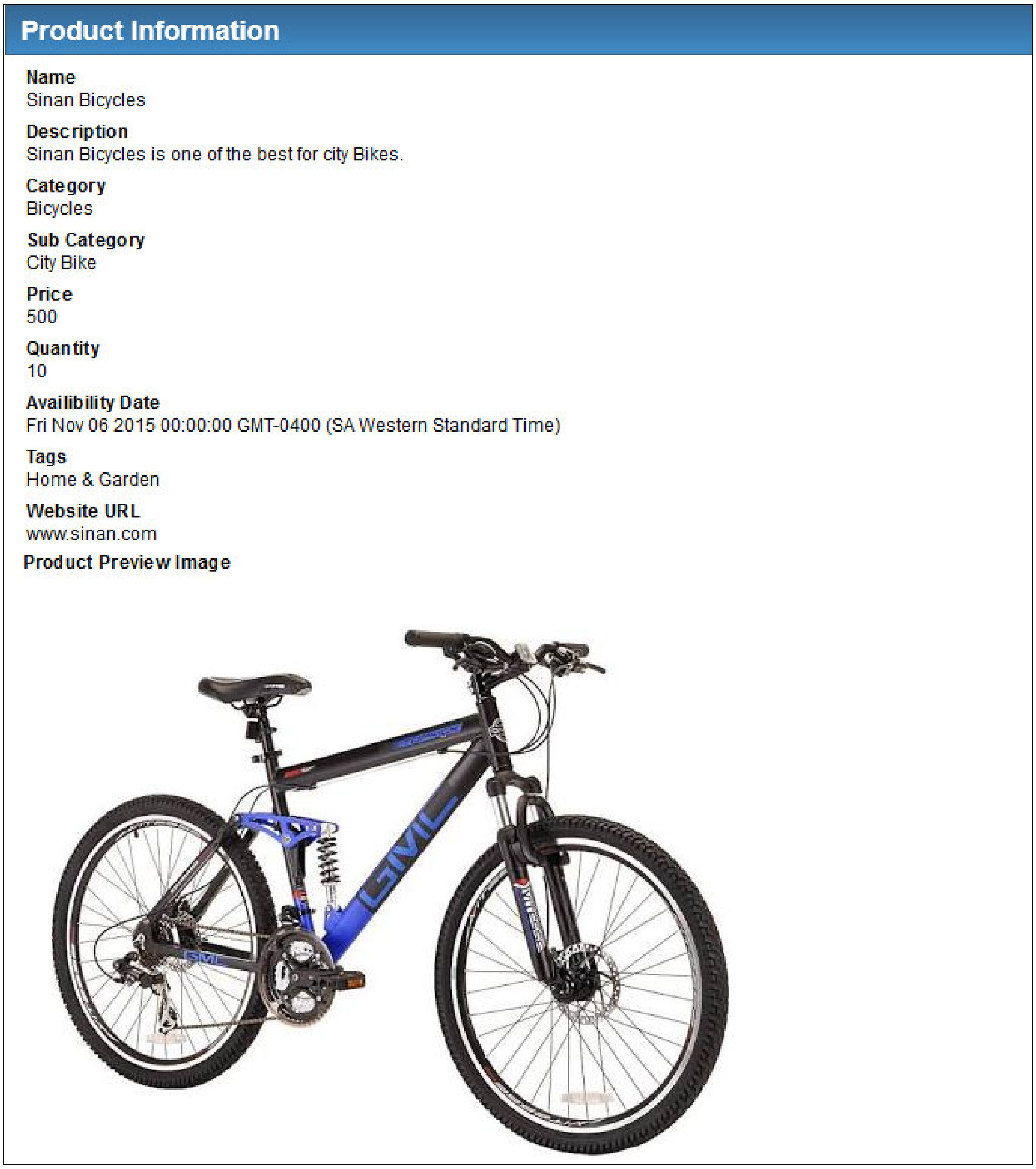
☑ Home & Garden

* Website URL

www.sinan.com

Figure 5-17 The IBM BPM Coach showing the first step in the process

Figure 5-18 shows the IBM BPM Coach confirmation page for step 1 of the business process in this example.



Product Information

Name
Sinan Bicycles

Description
Sinan Bicycles is one of the best for city Bikes.

Category
Bicycles

Sub Category
City Bike

Price
500

Quantity
10

Availibility Date
Fri Nov 06 2015 00:00:00 GMT-0400 (SA Western Standard Time)

Tags
Home & Garden

Website URL
www.sinan.com

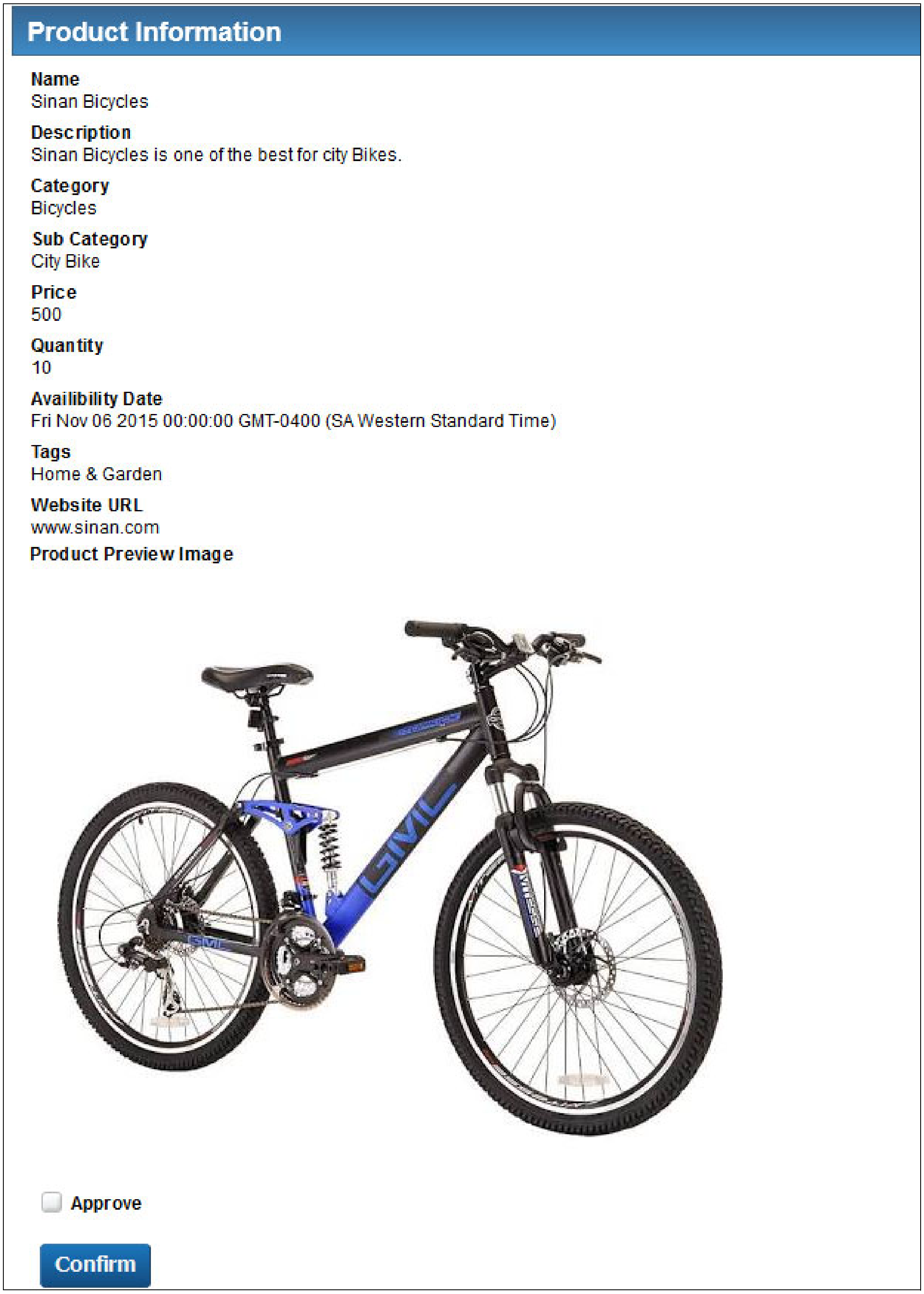
Product Preview Image

Figure 5-18 IBM BPM Coach confirmation page for step 1

Step 2: Management review

After the vendors submit the request to sell a new product, it moves to management review.

Any user within the Greenwheels Management LDAP group can find the request in the tasks list. Managers can see the details that were entered in “Step 1: Submit the selling product” on page 153. To approve the request, managers select the **Approve** check box and click **Confirm**, as shown in Figure 5-19 on page 157. Only one approval is required.



Product Information

Name
Sinan Bicycles

Description
Sinan Bicycles is one of the best for city Bikes.

Category
Bicycles

Sub Category
City Bike

Price
500

Quantity
10

Availibility Date
Fri Nov 06 2015 00:00:00 GMT-0400 (SA Western Standard Time)

Tags
Home & Garden

Website URL
www.sinan.com

Product Preview Image

□ **Approve**

Figure 5-19 Managers approve the request

Step 3: Executive approvals

Step 3 starts after a manager approves the vendor's request to sell the product. If a manager did not approve it, the process ends and the request does not go to executives.

Next, an executive reviews the request and either approves it or rejects it.

If the request is approved, it goes to the final step, which is System. In this example, the System has no tasks to complete and the process ends.

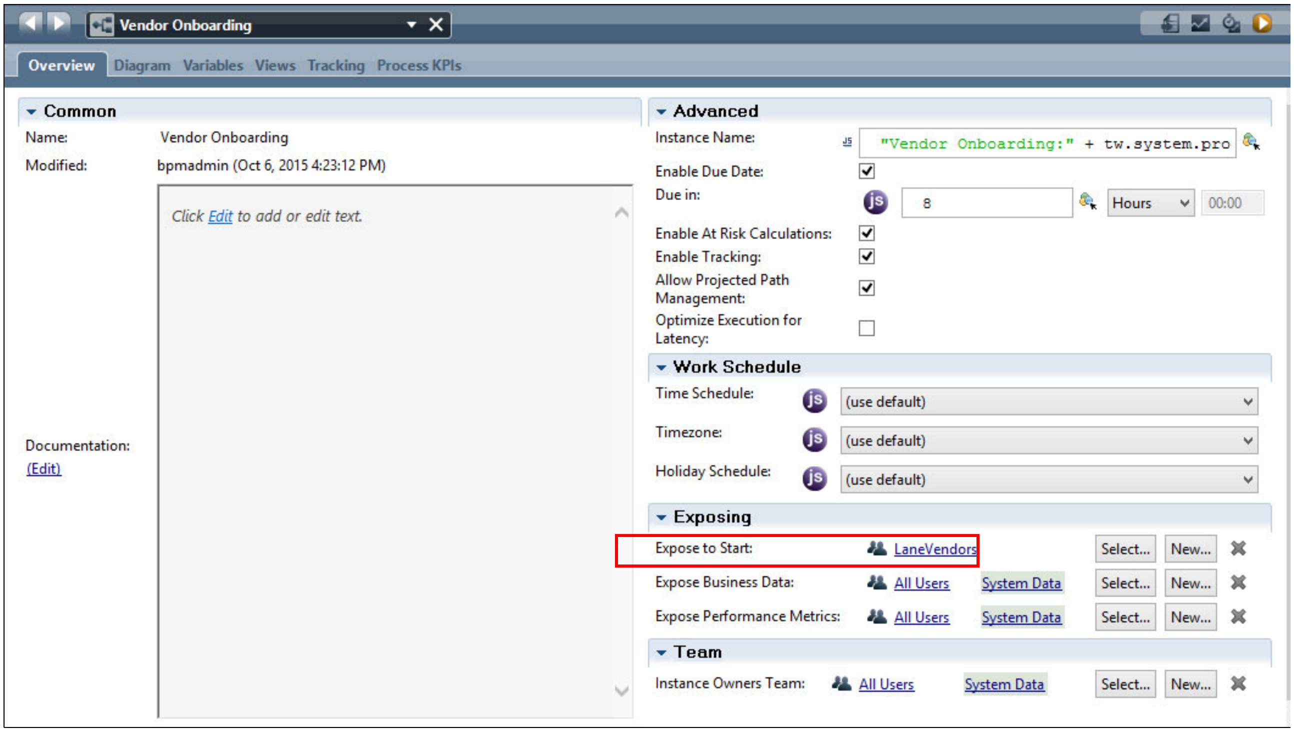
Digital Data Connector and IBM BPM configuration

This section highlights the configuration that is required on IBM BPM to integrate it with DDC. The DDC configuration is described in "Installing and configuring the Digital Data Connector components for IBM BPM" on page 158.

DDC provides a portlet that you can use to start processes. Each process can be started by a specific group or groups that are defined in LDAP and defined in the Expose to start setting of this process in IBM BPM.

To give the LaneVendor group the ability to start this process, complete the following steps:

1. Click the **Overview** tab in the Vendor Onboarding process configuration.
2. In the Exposing settings, select **LaneVendors** for Expose to start, as shown in Figure 5-20.



Name: Vendor Onboarding
Modified: bpmadmin (Oct 6, 2015 4:23:12 PM)
Documentation: (Edit)
Click Edit to add or edit text.

Instance Name: "Vendor Onboarding:" + tw.system.pro
Enable Due Date: ☑
Due in: 8 Hours 00:00
Enable At Risk Calculations: ☑
Enable Tracking: ☑
Allow Projected Path Management: ☑
Optimize Execution for Latency: □

Time Schedule: (use default)
Timezone: (use default)
Holiday Schedule: (use default)

Expose to Start: LaneVendors
Expose Business Data: All Users System Data
Expose Performance Metrics: All Users System Data

Instance Owners Team: All Users System Data

Figure 5-20 Configure the group that can start the IBM BPM process

Installing and configuring the Digital Data Connector components for IBM BPM

Download and install the Business Process Manager Sample for DDC in WebSphere Portal from the OpenNTF website:

http://openntf.org/main.nsf/project.xsp?r=project/Business%20Process%20Manager%20Sample%20for%20IBM%20Digital%20Data%20Connector%20in%20WebSphere%20Portal&cm_mc_uid=89300298032614374211666&cm_mc_sid_50200000=1444317803

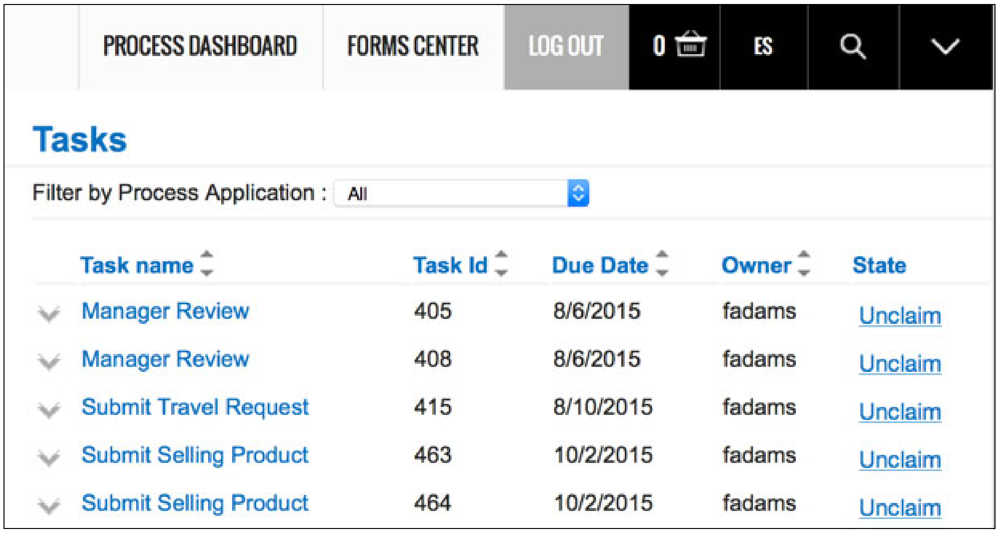
Read the installation instructions that are provided with the sample. Extract the file onto the portal server and run the two ConfigEngine tasks that are described in the document.

After restarting the server, modify the two Web Content Manager components that are identified in the document to point to your IBM BPM server.

Note: The DDC Coach content item can use either a proxy or a direct connection. In our testing, the Proxy setting did not work well with all types of coaches. If your coach does not render as you expect, consider switching to a direct connection.

The presentation components that are provided with the IBM BPM DDC package in the sample *Business Process Manager Sample for IBM Digital Data Connector in WebSphere Portal* from the OpenNTF website work well with the default Portal8.5 theme, but it is easy to make changes to customize the appearance to match any portal site.

The goal of this example is to change the header section of the Tasks portlet to look like other portlets. When the Tasks portlet is added to a page by default, it looks like the image that is shown in Figure 5-21.



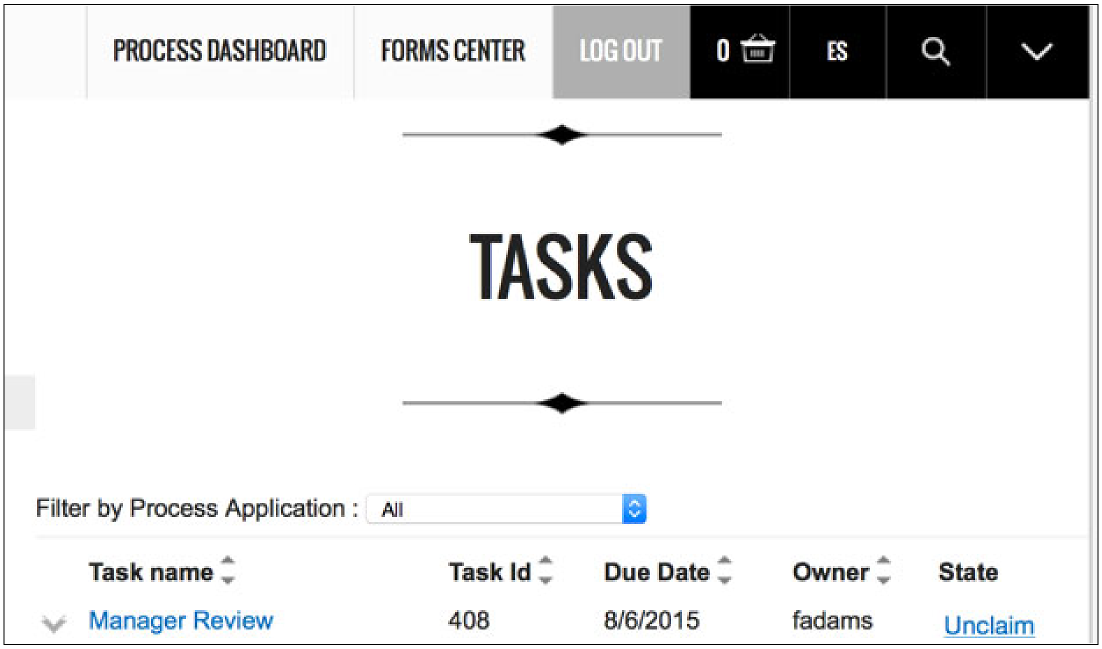
Tasks

Filter by Process Application : All

Task name	Task Id	Due Date	Owner	State
Manager Review	405	8/6/2015	fadams	Unclaim
Manager Review	408	8/6/2015	fadams	Unclaim
Submit Travel Request	415	8/10/2015	fadams	Unclaim
Submit Selling Product	463	10/2/2015	fadams	Unclaim
Submit Selling Product	464	10/2/2015	fadams	Unclaim

Figure 5-21 DDC task list default rendering

The objective of this example is to change a few elements to make the style of the title and table headers more consistent with the Greenwheels theme that is used in this example, as shown in Figure 5-22.



Filter by Process Application : All

Task name	Task Id	Due Date	Owner	State
Manager Review	408	8/6/2015	fadams	Unclaim

Figure 5-22 DDC Task list customized

To complete this change, you must change the Web Content Manager presentation components that are used in DDC. Use the Web Content Manager Authoring portlet to locate and edit those components, as described in the following steps:

1. Access the Web Content Manager Authoring Portlet by clicking the applications menu icon and then selecting **Content**, as shown in Figure 5-23.

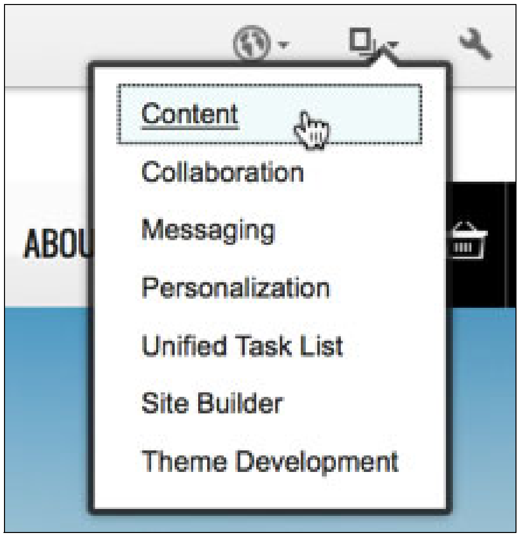
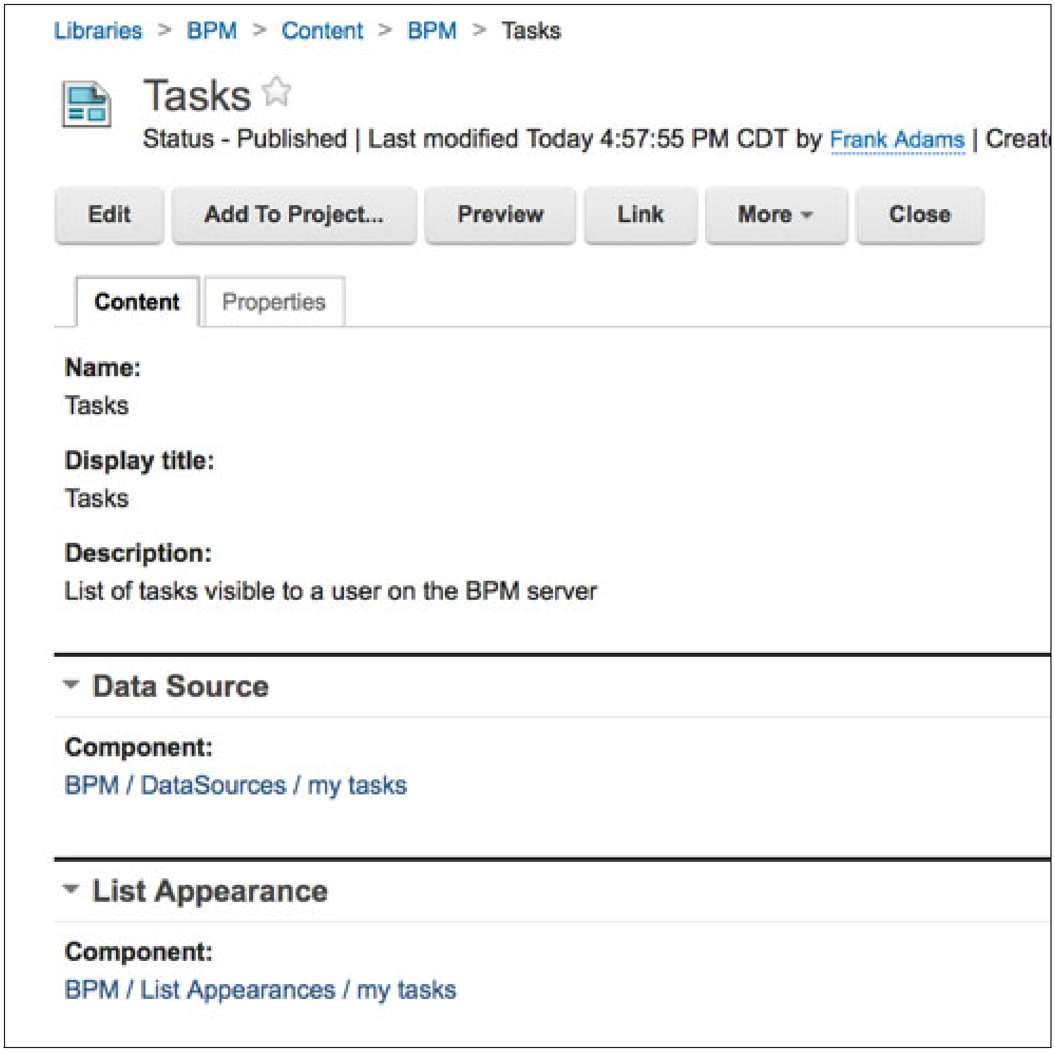


Figure 5-23 Go to the Web Content Manager Authoring Portlet

2. Open the Tasks content item in the IBM BPM library in the IBM BPM Site Area, as shown in Figure 5-24.



Libraries > BPM > Content > BPM > Tasks

Tasks

Status - Published | Last modified Today 4:57:55 PM CDT by Frank Adams | Creat

Name:
Tasks

Display title:
Tasks

Description:
List of tasks visible to a user on the BPM server

Data Source

Component:
BPM / DataSources / my tasks

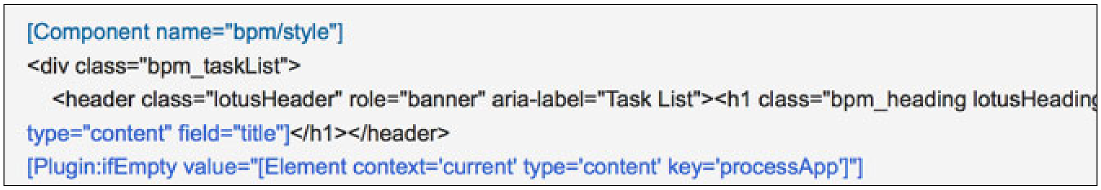
List Appearance

Component:
BPM / List Appearances / my tasks

Figure 5-24 Tasks item in IBM BPM library

3. The Tasks content item has a List Appearance component. Many of the functions of DDC for IBM BPM are found in Web Content Manager components. A component is a discrete unit within Web Content Manager that can contain server-side (code, variables, or data) or client-side (HTML, JavaScript, or CSS) features. Click **BPM** → **List Appearances** → **my tasks** to open it.

4. Scroll down to the List Presentation Markup element and locate the Header section. You can see that it contains a combination of HTML and Web Content Manager tags, as shown in Figure 5-25.



```
[Component name="bpm/style"]
<div class="bpm_taskList">
  <header class="lotusHeader" role="banner" aria-label="Task List"><h1 class="bpm_heading lotusHeading
type="content" field="title"]</h1></header>
[Plugin:ifEmpty value="[Element context='current' type='content' key='processApp']"]
```

Figure 5-25 List presentation markup

5. This is the first component to edit. To do so, scroll to the top of the form and click **Edit** to put the component into Edit mode.
6. Scroll down to the List Presentation Markup section. For the site in this example, insert the following HTML code into the text area both before and after the header tag, as shown in Example 5-1. This HTML code creates the images that appear above and below the title.

Example 5-1 HTML to add to the List Presentation Markup section

```
<div class="gw_line">
        <div class="gw_1pxline"><img width="100%" border="" height="2" alt=""
src="/wps/wcm/myconnect/69dc29ab-be1a-44eb-817c-9af439e17c89/diamond1px.png?MOD
=AJPERES&amp;CACHEID=69dc29ab-be1a-44eb-817c-9af439e17c89"></div>
        <div class="gw_diamond"><img width="32" border="" height="12" alt=""
src="/wps/wcm/myconnect/59821c6f-d13b-41f2-86e0-89bf94381cc2/diamond.png?MOD=AJ
PERES&amp;CACHEID=59821c6f-d13b-41f2-86e0-89bf94381cc2"></div>
        <div class="gw_1pxline"><img width="100%" border="" height="2" alt=""
src="/wps/wcm/myconnect/69dc29ab-be1a-44eb-817c-9af439e17c89/diamond1px.png?MOD
=AJPERES&amp;CACHEID=69dc29ab-be1a-44eb-817c-9af439e17c89"></div>
    </div>
```

7. For this Greenwheels example, change the code that creates the title by replacing the line, as shown in Example 5-2

Example 5-2 HTML to change the style of the title

```
replace this line:

<header class="lotusHeader" role="banner" aria-label="Task List"><h1
class="bpm_heading lotusHeading">[Property context="current" type="content"
field="title"]</h1></header>

with this line:

<header class="lotusHeader" role="banner" aria-label="Task List"><div
class="gw_sectionTitle">[Property context="current" type="content"
field="title"]</div></header>
```

8. To view your changes, scroll to the top of the page and click **Save**, as shown in Figure 5-26.

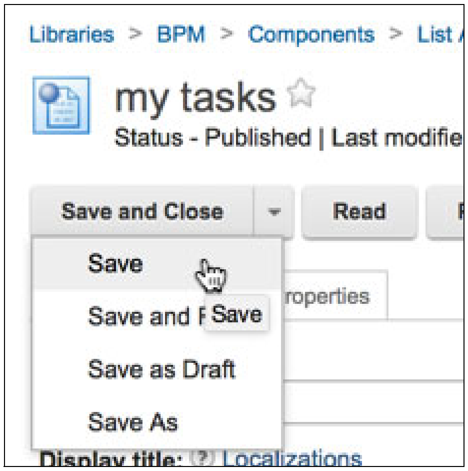
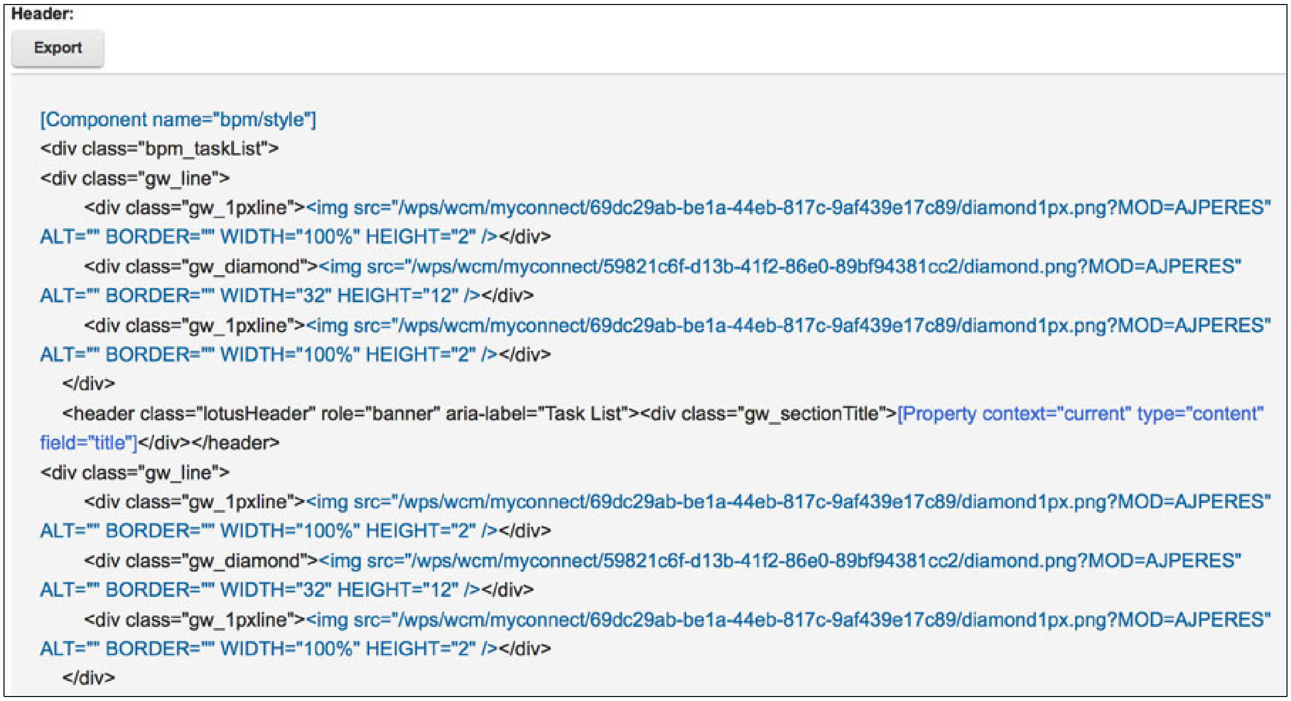


Figure 5-26 Save the my tasks component

9. You can now open another browser session and load the page that has this Tasks portlet to see how it looks. The list presentation markup header section now looks what is shown in Figure 5-27.



```
[Component name="bpm/style"]
<div class="bpm_taskList">
<div class="gw_line">
      <div class="gw_1pxline"><img src="/wps/wcm/myconnect/69dc29ab-be1a-44eb-817c-9af439e17c89/diamond1px.png?MOD=AJPERES"
ALT="" BORDER="" WIDTH="100%" HEIGHT="2" /></div>
      <div class="gw_diamond"><img src="/wps/wcm/myconnect/59821c6f-d13b-41f2-86e0-89bf94381cc2/diamond.png?MOD=AJPERES"
ALT="" BORDER="" WIDTH="32" HEIGHT="12" /></div>
      <div class="gw_1pxline"><img src="/wps/wcm/myconnect/69dc29ab-be1a-44eb-817c-9af439e17c89/diamond1px.png?MOD=AJPERES"
ALT="" BORDER="" WIDTH="100%" HEIGHT="2" /></div>
   </div>
   <header class="lotusHeader" role="banner" aria-label="Task List"><div class="gw_sectionTitle">[Property context="current" type="content"
field="title"]</div></header>
<div class="gw_line">
      <div class="gw_1pxline"><img src="/wps/wcm/myconnect/69dc29ab-be1a-44eb-817c-9af439e17c89/diamond1px.png?MOD=AJPERES"
ALT="" BORDER="" WIDTH="100%" HEIGHT="2" /></div>
      <div class="gw_diamond"><img src="/wps/wcm/myconnect/59821c6f-d13b-41f2-86e0-89bf94381cc2/diamond.png?MOD=AJPERES"
ALT="" BORDER="" WIDTH="32" HEIGHT="12" /></div>
      <div class="gw_1pxline"><img src="/wps/wcm/myconnect/69dc29ab-be1a-44eb-817c-9af439e17c89/diamond1px.png?MOD=AJPERES"
ALT="" BORDER="" WIDTH="100%" HEIGHT="2" /></div>
   </div>
```

Figure 5-27 List presentation header

10. Change the stylesheet that is used by this portlet to conform to the demonstration site styles.

    The stylesheet is in a component that is called style in the same Web Content Manager library and is referenced at the top of the header section with the tag `[Component name=”bpm/style”]`. Click this link to open the stylesheet component.

11. Click **Edit** in the style component to put it into Edit mode and make any changes as needed. Save your changes and refresh the page that has the Tasks portlet to view your changes.

An advantage of using Web Content Manager to render the DDC portlets is that all these changes can be managed in a Web Content Manager project. For information about using Web Content Manager projects, see 4.5, “Projects and project templates” on page 79.

5.2.6 Use case 3: Integrating with IBM Forms Experience Builder

Forms Experience Builder provides a REST interface called the Data Access REST API. This interface allows external systems to retrieve the data that is captured by Forms Experience Builder applications. This data can be easily consumed by using the DDC and displayed by using existing WebSphere Content Manager presentation techniques.

Furthermore, the DDC portlet presenting the list data through Web Content Manager can communicate with a Forms Experience Builder portlet on the sample page through public render parameters to send data to the form in the other portlet. For example, there can be a link to a specific form to be rendered in the Forms Experience Builder, as shown in Figure 5-36 on page 175.The communication flow is shown in Figure 5-28.

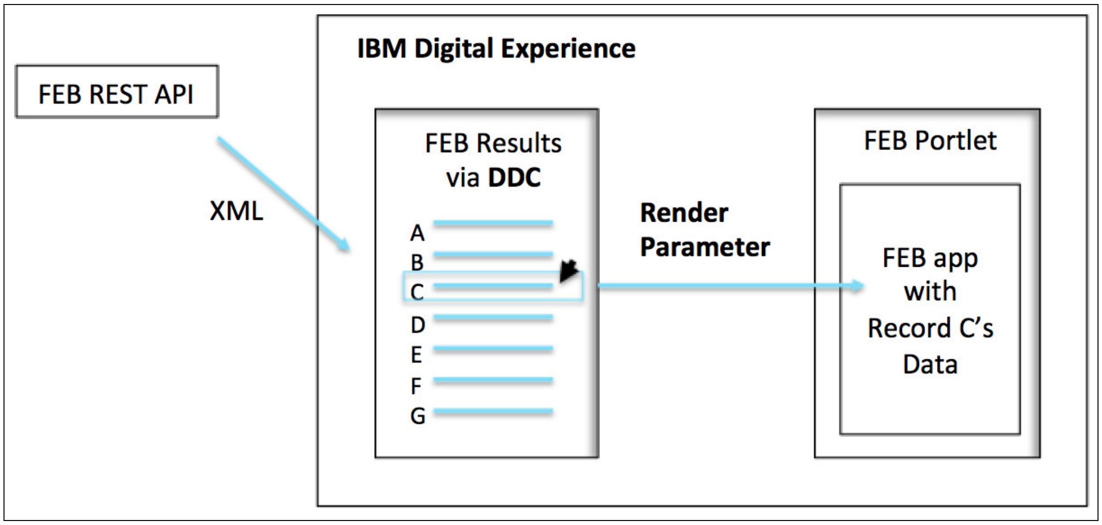


Figure 5-28 Overview of using DDC with Forms Experience Builder

This example describes how to extend the Greenwheels demonstration website with a bike rental function based on a Forms Experience Builder form. DDC can then be used to implement a task list that presents the requests that were submitted based on their approval stage.

The presentation template can be customized and designed to suit the needs of the use case and the corporate design of the Greenwheels demonstration website in this example. Using page parameters, the selected form entry in the list can then be presented by using the Forms Experience Builder portlet with a simple configuration.

The following sections provide the steps to set up and customize this scenario. There are two main configuration steps that are required to implement this example:

1. Create a list rendering profile so that the DDC can understand the structure of the data that is returned by the Data Access REST API, as described in “Creating a list rendering profile” on page 164.
2. Create a list appearance componentthat can display the data in an appealing way. For example, showing images such as the picture of a bicycle and a link to start the associated form. See “Creating a list appearance component” on page 167.

Creating a list rendering profile

To create a list rendering profile, complete the following steps:

1. In a text editor, open the `ListRenderingProfileService.properties` file, which is in the following directory:

   ```
   wp_profile\PortalServer\config\ListRenderingProfileService.properties
   ```

   Make the wanted field values from the form available as DDC attributes in Web Content Manager by adding the respective path references to the attribute in the XML response from the Forms Experience Builder server, as shown in Example 5-3 on page 167.

Table 5-4 lists the DDC properties that are used in this example to configure the list rendering profile and to reference the field values in Forms Experience Builder.

Table 5-4 List rendering profile attributes used in this example

DDC profile attribute	Description
`rentals.Name=ibm.febBikeRental.atom`	`rentals.Name` specifies the name of the new list rendering profile in this example.
`rentals.BeanListProviderID=ibm.portal.ddc.xml`	The available XML bean list provider for DDC is used in this example.
`rentals.Extends=ibm.portal.atom`	Extend the existing `ibm.portal.atom` list rendering profile.

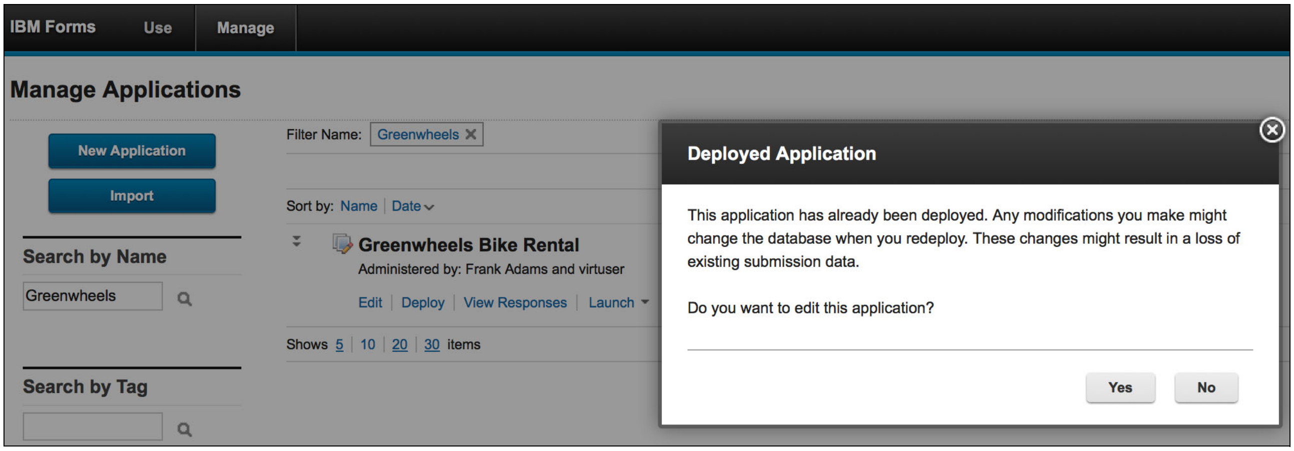
`rentals.ItemAttribute.F_BicycleID=`
`./atom:content/F_Form1/F_BicycleID`
	`rentals.ItemAttribute.F_BicycleID` tells DDC to create an attribute called `F_BycicleID`
	`./atom:content/F_Form1/F_BicycleID` is the location of the data in the XML response.

2. Look up the element IDs in the Forms Experience Builder form to reference the wanted fields in the XML response.

   To find the referenced element IDs, such as `F_BicycleID`, you can look them up in the Forms Experience Builder design experience. For this example, here are the steps:

   a. Open the Greenwheels Bike Rental application in the Manage tab in IBM Forms by clicking the **Edit** link.

   b. In the Deployed Application window, click **Yes** to confirm that you want to edit the deployed application, as shown in Figure 5-29.

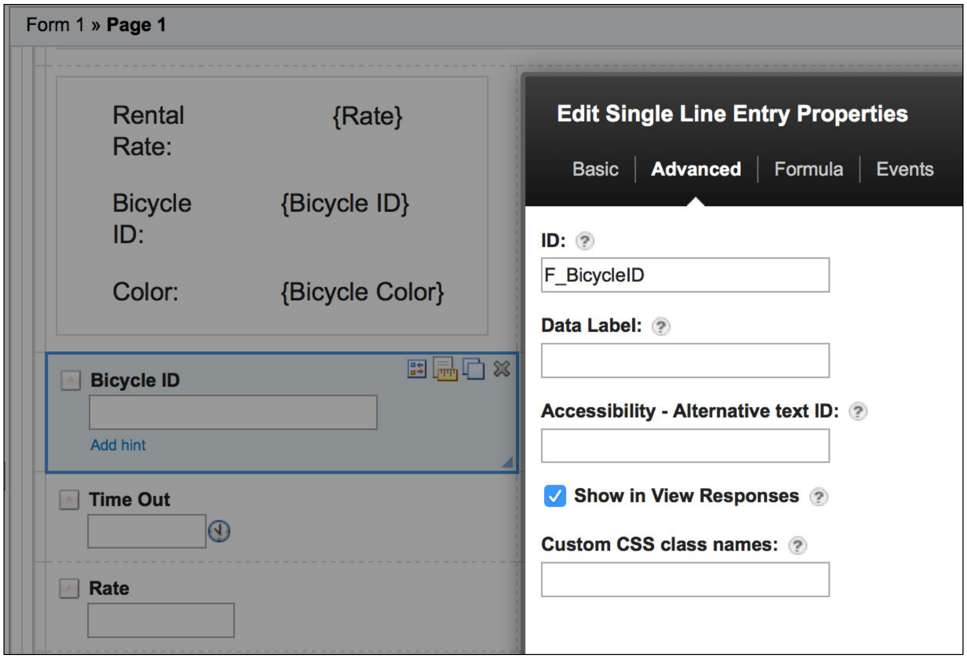


This application has already been deployed. Any modifications you make might change the database when you redeploy. These changes might result in a loss of existing submission data.

Do you want to edit this application?

Figure 5-29 Edit the Greenwheels bike rental application in Forms Experience Builder

c. In the Forms Experience Builder design experience, open the properties dialog box of an element and inspect the element in the Advanced tab, as shown in Figure 5-30. The value in the ID field is what you must reference in the path expression for the item attribute in the list rendering profile.



Rental Rate: {Rate}

Bicycle ID: {Bicycle ID}

Color: {Bicycle Color}

Bicycle ID

Time Out

Rate

Edit Single Line Entry Properties

ID: F_BicycleID

Data Label:

Accessibility - Alternative text ID:

☑ Show in View Responses

Custom CSS class names:

Figure 5-30 Look up the element ID in the Forms Experience Builder properties dialog box

You can easily reference additional item attributes from the Forms Experience Builder form to make them available in Web Content Manager if you want to show different or additional information in your task list in WebSphere Portal.

Example 5-3 shows the fully configured list rendering profile for this scenario.

Example 5-3 Properties of the list rendering profile service

```
------------------------------------------------ #
# Properties of the List Rendering Profile Service #
# ------------------------------------------------ #
################
# ATOM profile #
################
rentals.Name=ibm.febBikeRental.atom
rentals.BeanListProviderID=ibm.portal.ddc.xml
rentals.ItemAttribute.id=./atom:id
rentals.NamespaceMapping.atom=http://www.w3.org/2005/Atom
rentals.ResourceBundleBaseName=com.ibm.workplace.wcm.pzn.xml.AtomBeanListProvider
rentals.Extends=ibm.portal.atom
rentals.ItemAttribute.F_BicycleID=./atom:content/F_Form1/F_BicycleID
rentals.ItemAttribute.F_BicycleColor=./atom:content/F_Form1/F_BicycleColor
rentals.ListItemSelection=//atom:entry
rentals.ItemAttribute.F_Rate=./atom:content/F_Form1/F_Rate
rentals.ItemAttribute.F_Name=./atom:content/F_Form1/F_Name
rentals.ItemAttribute.F_EmailAddress=./atom:content/F_Form1/F_EmailAddress
rentals.ItemAttribute.F_Date=./atom:content/F_Form1/F_Date
rentals.ItemAttribute.F_TimePickup=./atom:content/F_Form1/F_TimePickup
rentals.ItemAttribute.F_TimeReturn=./atom:content/F_Form1/F_TimeReturn
rentals.ItemAttribute.F_Accessories=./atom:content/F_Form1/F_Accessories
rentals.ItemAttribute.F_RentalAmount=./atom:content/F_Form1/F_RentalAmount
rentals.ItemAttribute.F_ImageURL=./atom:content/F_Form1/F_ImageURL
rentals.ItemAttribute.F_ProfileImage_id=./atom:content/F_Form1/F_ProfileImage/id
```

3. Run following `ConfigEngine` task to update DDC with the new rendering profile information:

   ```
   \wp_profile\ConfigEngine\ConfigEngine.sh update-properties
   ```

   If the `ConfigEngine` task ends with the message `Build Successful`, the new rendering profile is available in Web Content Manager to create a list appearance.

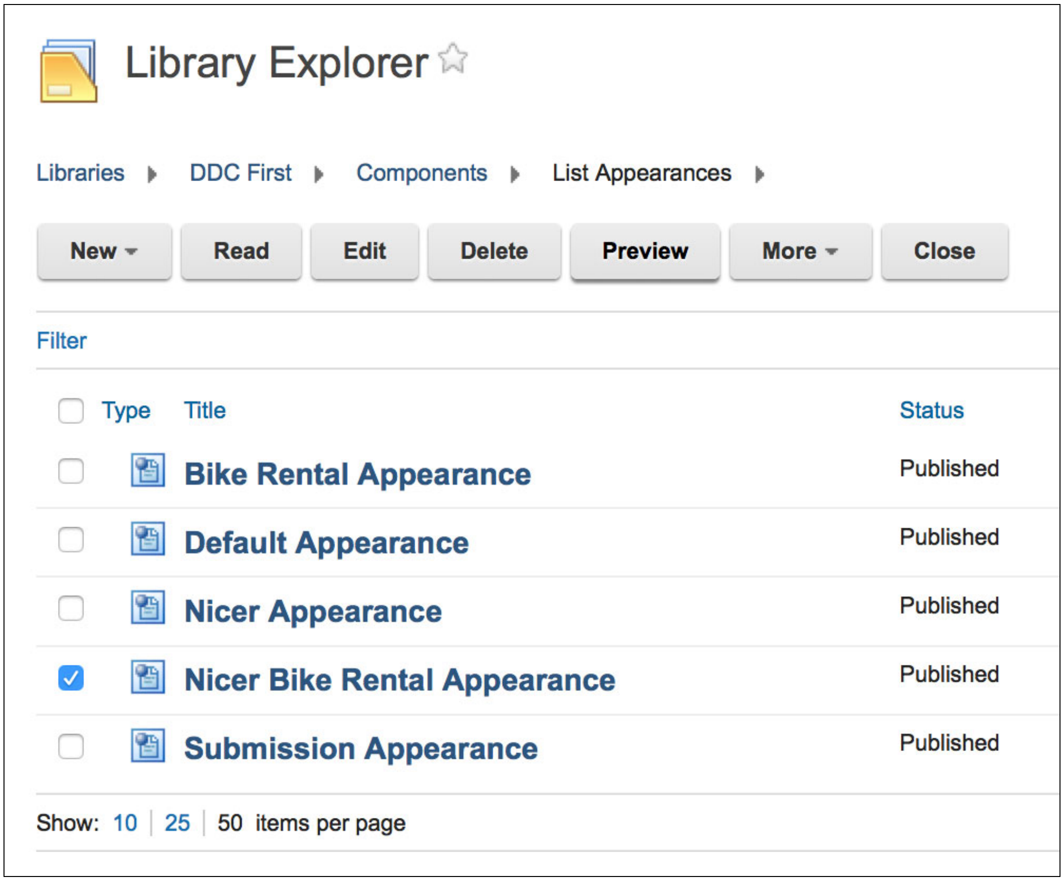
Creating a list appearance component

After the data structure of the Data Access REST API call to DDC is described, a new list appearance can be built by using the attributes in the list rendering profile to display the Forms Experience Builder data in the DDC portlet.

List appearances contain three main sections: header, result design, and footer. In a list appearance, you specify the HTML and CSS of each of these sections. At run time, the DDC portlet displays the header. For every row of data that is returned in the REST call, DDC includes an instance of the result design. The footer is inserted at the end. The result design section is where you reference the attributes that are specified in the list rendering profile, that is, where you display the data for a form.

In a list appearance, you can specify the building blocks to create an HTML list, an HTML table, or a set of DIVs. Complete the following steps:

1. Find and edit the Default Appearance by clicking **Libraries** → **DDC First** → **Components** → **List Appearances**, as shown in Figure 5-31.

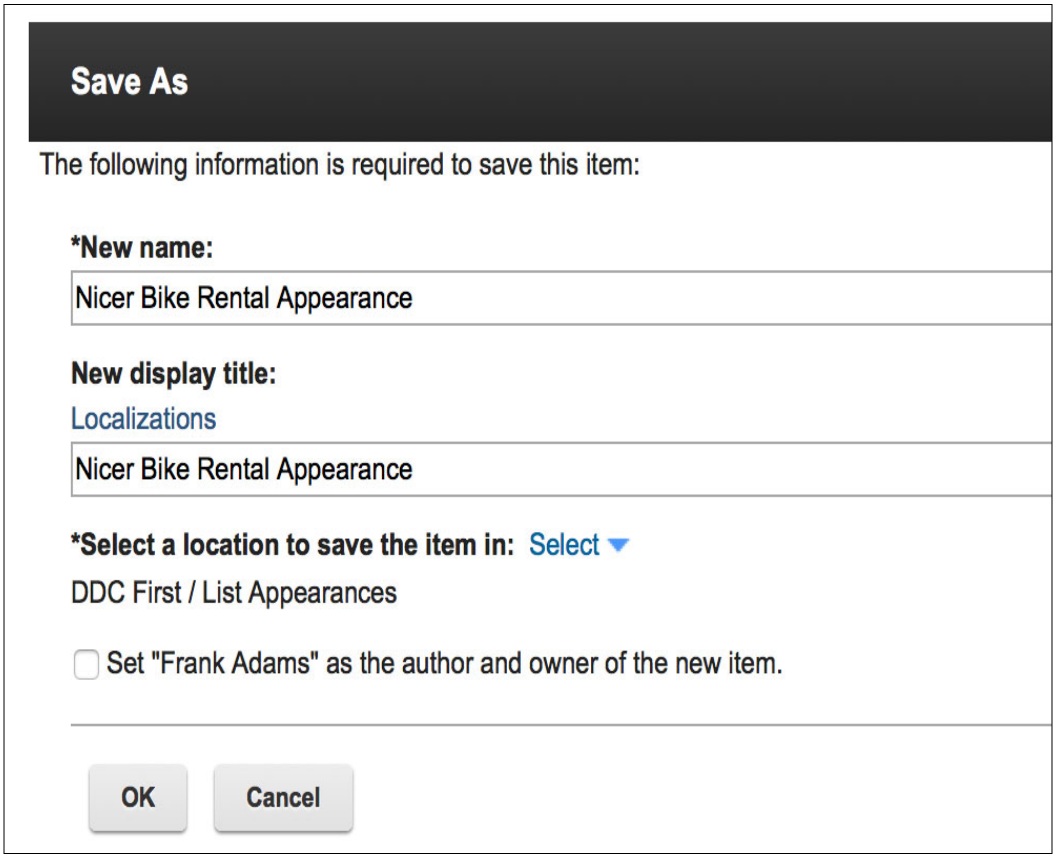


□	Type	Title	Status
□		Bike Rental Appearance	Published
□		Default Appearance	Published
□		Nicer Appearance	Published
☑		Nicer Bike Rental Appearance	Published
□		Submission Appearance	Published

Show: 10 | 25 | 50 items per page

Figure 5-31 Access List Appearances through IBM Web Content Manager authoring

2. Click **Save As** and enter a name for the list appearance, for example, Bike Rental Appearance, as shown in Figure 5-32. You can access the list appearances through the library explorer in the Web Content Manager authoring portlet by clicking **Content** → **Web Content Authoring** in the Applications menu on the right side of the WebSphere Portal toolbar.



Save As

The following information is required to save this item:

*New name: Nicer Bike Rental Appearance

New display title: Localizations

Nicer Bike Rental Appearance

*Select a location to save the item in: Select ▾

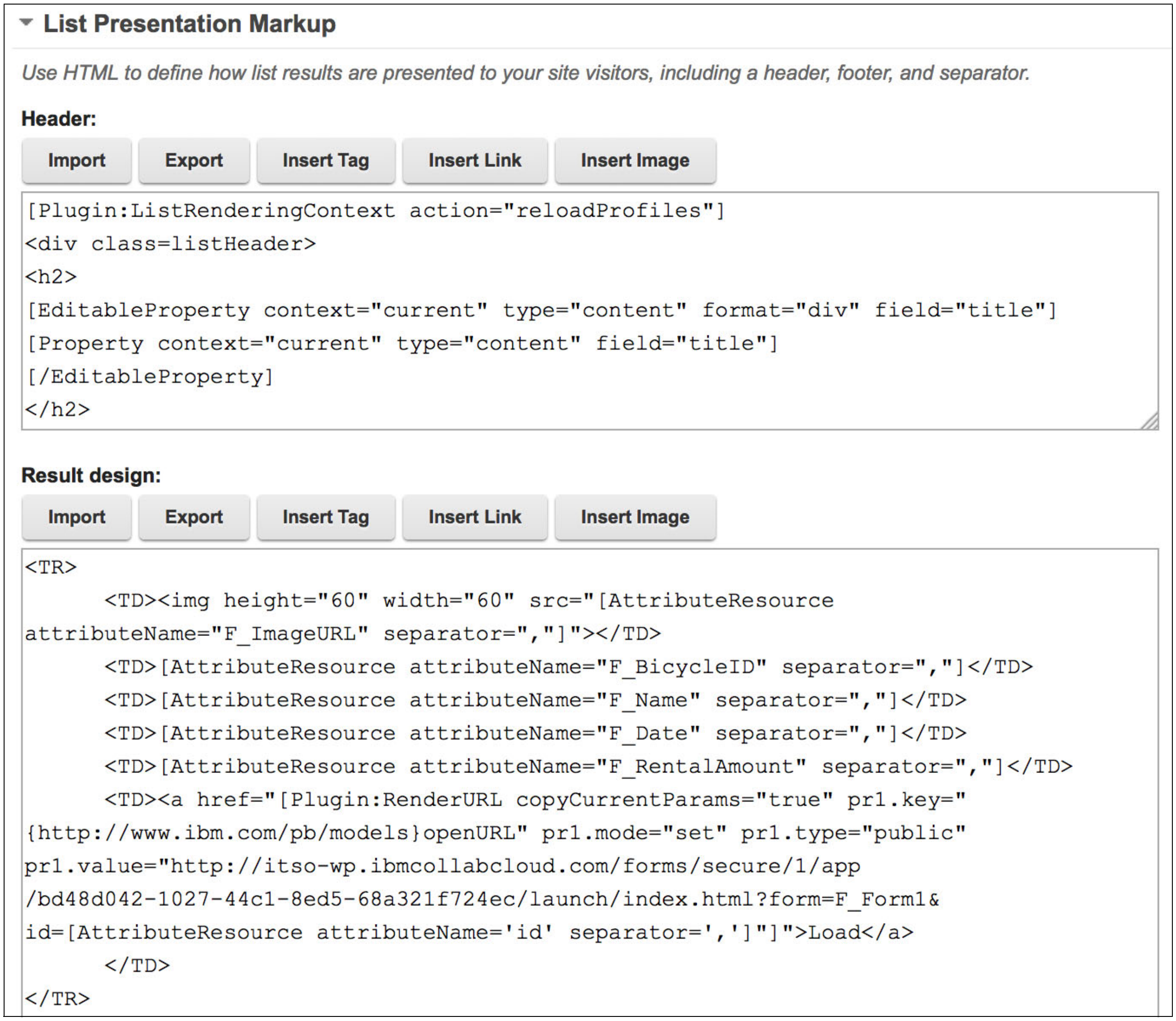
DDC First / List Appearances

□ Set "Frank Adams" as the author and owner of the new item.

Figure 5-32 Copy the Default Appearance by saving it as a new name

3. Open the new Bike Rentals List Appearance and change the contents to present the data and layout for this scenario.

   In the List Presentation Markup section, use the Header, Result Design, and Footer to design your List Appearance, as shown in Figure 5-33.



Use HTML to define how list results are presented to your site visitors, including a header, footer, and separator.

Header:

```
[Plugin:ListRenderingContext action="reloadProfiles"]
<div class=listHeader>
<h2>
[EditableProperty context="current" type="content" format="div" field="title"]
[Property context="current" type="content" field="title"]
[/EditableProperty]
</h2>
```

Result design:

```
<TR>
      <TD><img height="60" width="60" src="[AttributeResource
attributeName="F_ImageURL" separator=","]"></TD>
      <TD>[AttributeResource attributeName="F_BicycleID" separator=","]</TD>
      <TD>[AttributeResource attributeName="F_Name" separator=","]</TD>
      <TD>[AttributeResource attributeName="F_Date" separator=","]</TD>
      <TD>[AttributeResource attributeName="F_RentalAmount" separator=","]</TD>
      <TD><a href="[Plugin:RenderURL copyCurrentParams="true" pr1.key="
{http://www.ibm.com/pb/models}openURL" pr1.mode="set" pr1.type="public"
pr1.value="http://itso-wp.ibmcollabcloud.com/forms/secure/1/app
/bd48d042-1027-44c1-8ed5-68a321f724ec/launch/index.html?form=F_Form1&
id=[AttributeResource attributeName='id' separator=',']"]">Load</a>
      </TD>
</TR>
```

Figure 5-33 List Presentation Markup section

In the List Presentation Header, you specify the table header markup and additional style information to generate a nicer looking table and to style the markup to your corporate website design. You can also add the styling information to your overall website CSS stylesheet instead of keeping it directly in the rendering component.

Example 5-4 shows the sample HTML markup for the List Presentation Header. The first tag for the ListRenderingContext plug-in tells DDC to reload all list rendering profiles every time the DDC portlet is rendered. List rendering profiles are cached by default, so this tag is useful if you are changing your list rendering profile and need to test them.

In the Header, you can place style information to build an appealing or streamlined presentation of your list data. This example uses a CSS table generator to render a nicer table presentation and specific fonts and colors to match the overall website design. Again, you can place this style information in your website CSS stylesheet for centralized maintenance.

Finally, you can find the column headers for the Forms Experience Builder data that is presented in the list in the table tags at the end of the header section.

Example 5-4 List Presentation Header

```
[Plugin:ListRenderingContext action="reloadProfiles"]
<div class=listHeader>
<h2>
[EditableProperty context="current" type="content" format="div" field="title"]
[Property context="current" type="content" field="title"]
[/EditableProperty]
</h2>
</div>
<style>
.listHeader {
   padding:7px;
   font-size:14px;
   font-family:oswaldbook,sans-serif;
   font-weight:bold;
   color:#000000;
}
.CSSTableGenerator {
   margin:0px;padding:0px;
   width:99%;
   border:1px solid rgba(0, 0, 0, 0.1);
...
</style>
<div class="CSSTableGenerator">
<TABLE>
   <TR>
      <TD>Picture</TD>
      <TD>Bicycle ID</TD>
      <TD>Name</TD>
      <TD>Date</TD>
      <TD>$$$</TD>
      <TD></TD>
   </TR>
```

4. In the result design of the list presentation, you reference the wanted attributes within your table row data by using the Insert a Tag dialog box in Web Content Manager, as shown in Figure 5-34. You use this dialog box to add easily reference Web Content Manager tags instead of entering the data manually. Complete the following steps:

   a. Place the cursor in your HTML result design in the location where you want to insert the data element from Forms Experience Builder and click **Insert Tag** to open the corresponding dialog box.

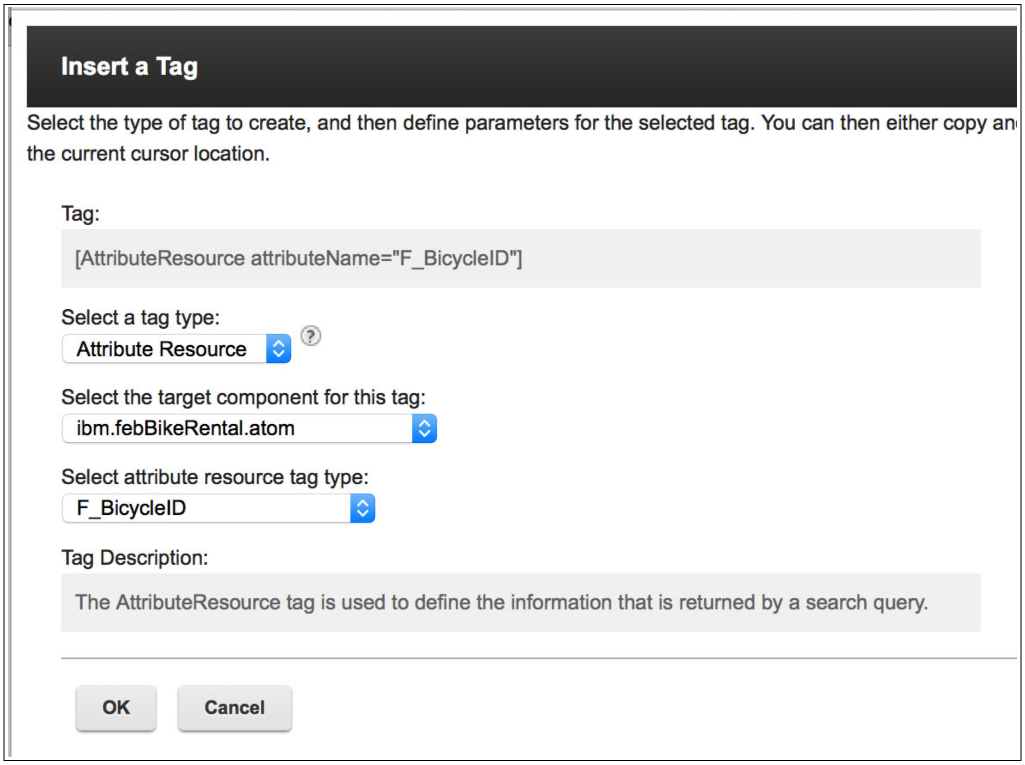
   b. In the Insert a Tag dialog box, enter the following information:

      i. Select **Attribute Resource** as the tag type.

      ii. Select **ibm.feb.BikeRental.atom** as the target component for this tag.

      iii. Select **F_BicycleID** or any other Forms Experience Builder element that you want to reference as an attribute resource tag type.

   Click **OK** to close the dialog box.



Insert a Tag

Select the type of tag to create, and then define parameters for the selected tag. You can then either copy an
the current cursor location.

Tag:

[AttributeResource attributeName="F_BicycleID"]

Select a tag type:

Attribute Resource

Select the target component for this tag:

ibm.febBikeRental.atom

Select attribute resource tag type:

F_BicycleID

Tag Description:

The AttributeResource tag is used to define the information that is returned by a search query.

Figure 5-34 Inserting an attribute tag into the result design

The following Web Content Manager tag is inserted into the HTML result design:

```
[AttributeResource attributeName="F_BicycleID" separator=","]
```

Example 5-5 shows the result design for this scenario with the following three distinct data element types referenced from the Forms Experience Builder form application:

- Attribute `F_ImageURL` references the URL of the bicycle picture and is presented within an `img` tag.
- Attributes `F_BicycleID`, `F_Name`, `F_Date`, and `F_RentalAmount` are normal fields from the Forms Experience Builder application to present meaningful metadata to the user of the task list.
- Attribute `id` in the last table column is returning the process ID of the list Forms Experience Builder data record and is used to construct a URL parameter that points to the corresponding Forms Experience Builder form that is set as a page parameter on the portal page. The Forms Experience Builder portlet on the same page then receives that page parameter, retrieves the Forms Experience Builder form with the data completed, and renders it, as shown in Figure 5-36 on page 175.

Example 5-5 List presentation result design

```
<TR>
      <TD><img height="60" width="60" src="[AttributeResource attributeName="F_ImageURL"
separator=","]"></TD>
      <TD>[AttributeResource attributeName="F_BicycleID" separator=","]</TD>
      <TD>[AttributeResource attributeName="F_Name" separator=","]</TD>
      <TD>[AttributeResource attributeName="F_Date" separator=","]</TD>
      <TD>[AttributeResource attributeName="F_RentalAmount" separator=","]</TD>
     <TD><a href="[Plugin:RenderURL copyCurrentParams="true" pr1.key="{http://www.ibm.com/pb/models}openURL"
pr1.mode="set" pr1.type="public"
pr1.value="http://itso-wp.ibmcollabcloud.com/forms/secure/1/app/bd48d042-1027-44c1-8ed5-68a321f724ec/launch/i
ndex.html?form=F_Form1&id=[AttributeResource attributeName='id' separator=',']"]">Load</a>
      </TD>
</TR>
```

5. Add a DDC portlet from the Page Components tab and a Forms Experience Builder Portlet from the Applications tab of the toolbar to your page by completing the same steps that are described in step 3 on page 145.

6. Configure the DDC portlet to render the wanted information.

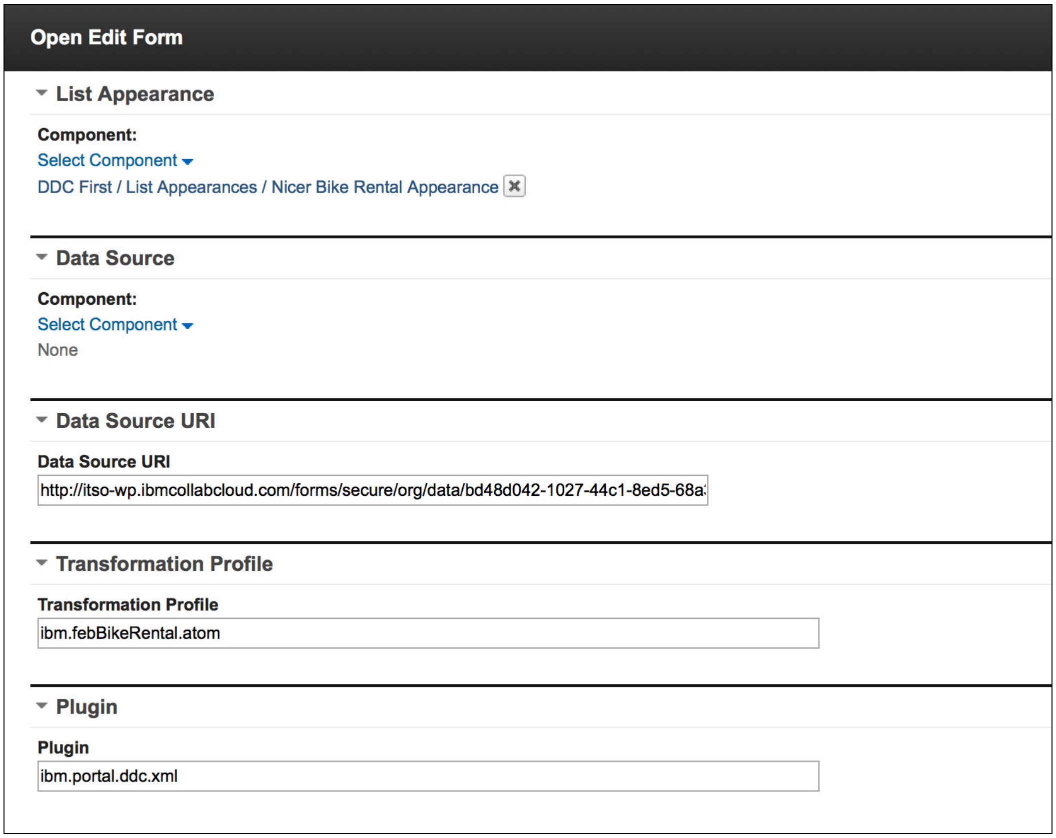
   Hover your cursor over the new DDC portlet and select **Open Edit Form**. The configuration dialog box opens. Enter the following information, as shown in Figure 5-35.

   – Select Component: **List Appearance** → **Component** → **Nicer Bike Rental Appearance**

   – Data Source URI of the Forms Experience Builder service call.

   – Transformation Profile **ibm.febBikeRental.atom**, as specified in the list rendering profile.

   Click **OK** to close the dialog box.



Open Edit Form

List Appearance

Component:
Select Component
DDC First / List Appearances / Nicer Bike Rental Appearance ☒

Data Source

Component:
Select Component
None

Data Source URI

Data Source URI
http://itso-wp.ibmcollabcloud.com/forms/secure/org/data/bd48d042-1027-44c1-8ed5-68a

Transformation Profile

Transformation Profile
ibm.febBikeRental.atom

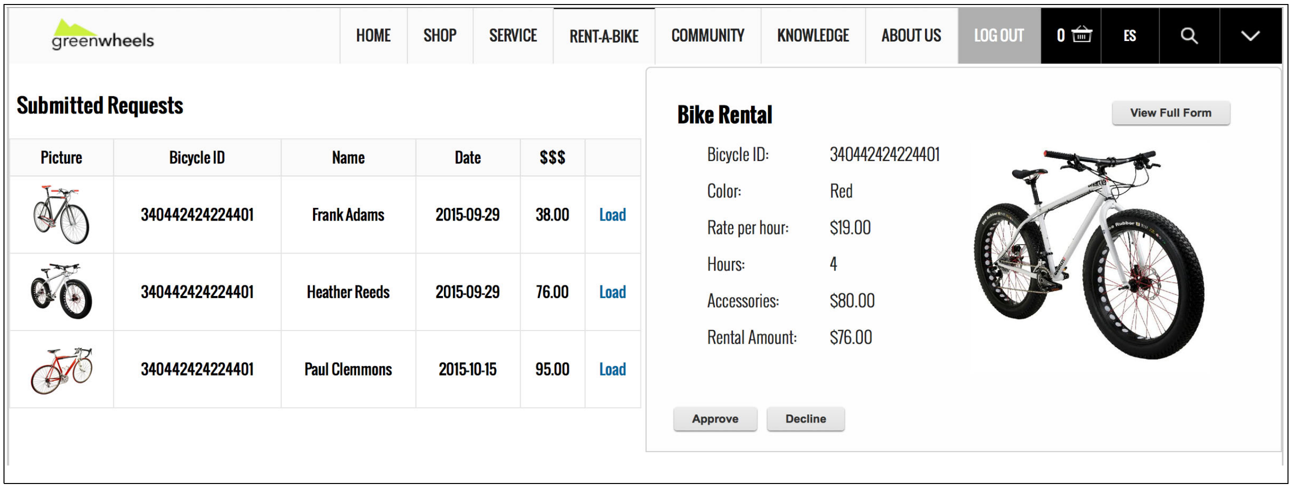
Plugin

Plugin
ibm.portal.ddc.xml

Figure 5-35 Open Edit Form

7. Turn off **Edit Mode**.

The page now renders the list of submitted Bike Rental requests on the left, as shown in Figure 5-36. When you click the **Load** link, the Forms Experience Builder portlet on the right renders the Bike Rental form with the corresponding data set prepopulated.



Submitted Requests

Picture	Bicycle ID	Name	Date	$$$	
	340442424224401	Frank Adams	2015-09-29	38.00	Load
	340442424224401	Heather Reeds	2015-09-29	76.00	Load
	340442424224401	Paul Clemmons	2015-10-15	95.00	Load

Bike Rental

Bicycle ID:	340442424224401
Color:	Red
Rate per hour:	$19.00
Hours:	4
Accessories:	$80.00
Rental Amount:	$76.00

Figure 5-36 DDC rendering submitted requests from Forms Experience Builder

5.2.7 Use Case 4: Integrating with IBM WebSphere Commerce

DDC allows WebSphere Portal to become a front end for WebSphere Commerce, pulling data into the portal by using the WebSphere Commerce REST APIs. WebSphere Commerce can then behave in a *headless* manner, acting as a data source for WebSphere Portal.

The sample WebSphere Commerce Sample for DDC in WebSphere Portal includes 16 content items that you can drag on your portal pages, as shown in Figure 5-37. You can download the sample and installation manual from the OpenNTF website:

http://www.openntf.org/main.nsf/project.xsp?r=project/WebSphere%20Commerce%20Sample%20for%20IBM%20Digital%20Data%20Connector%20in%20WebSphere%20Portal

Each content item corresponds to a function that is available in a typical store front, for example, a shopping cart or wish list. These content items are provided as examples of how you can easily integrate WebSphere Portal and WebSphere Commerce by using DDC.

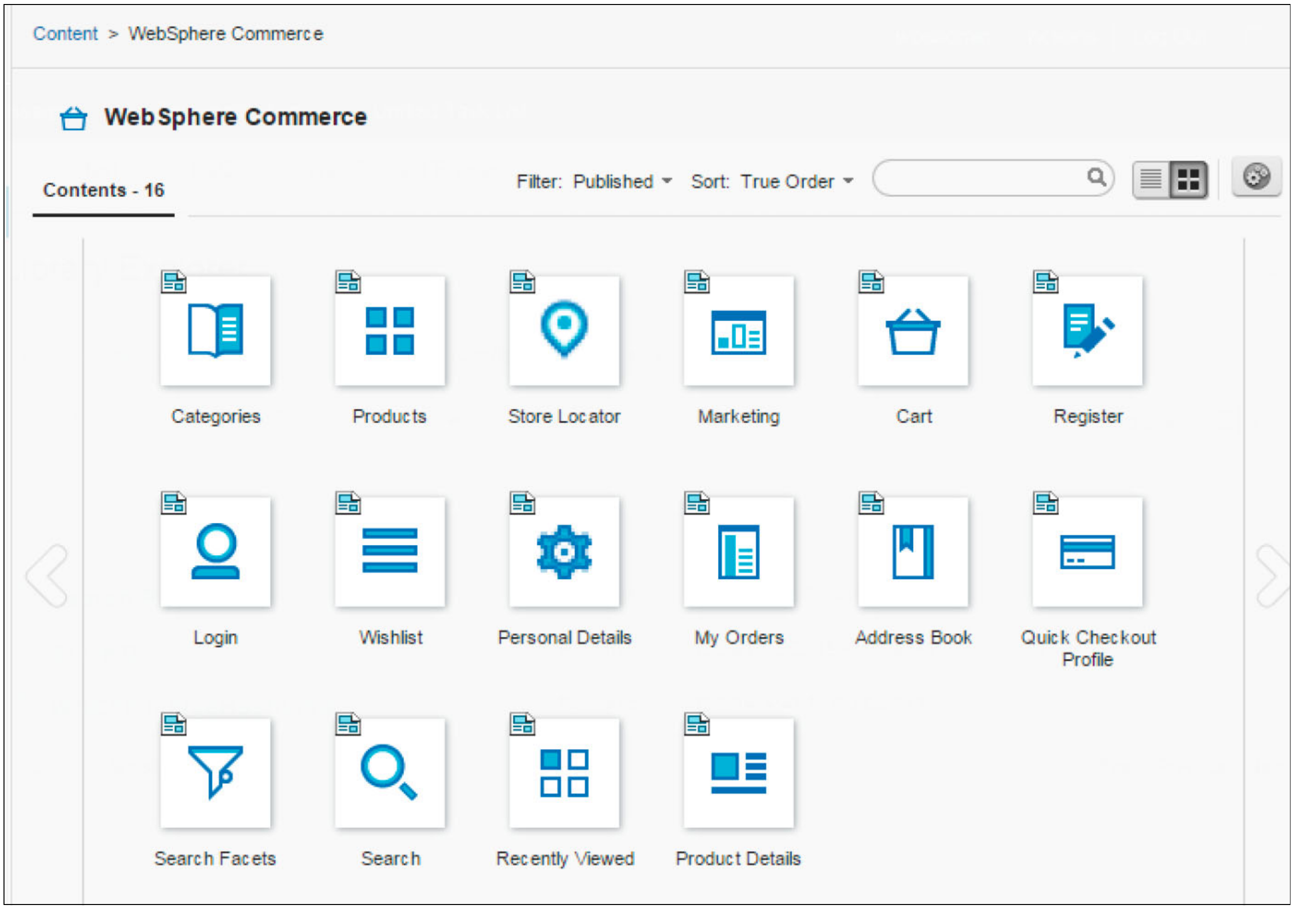
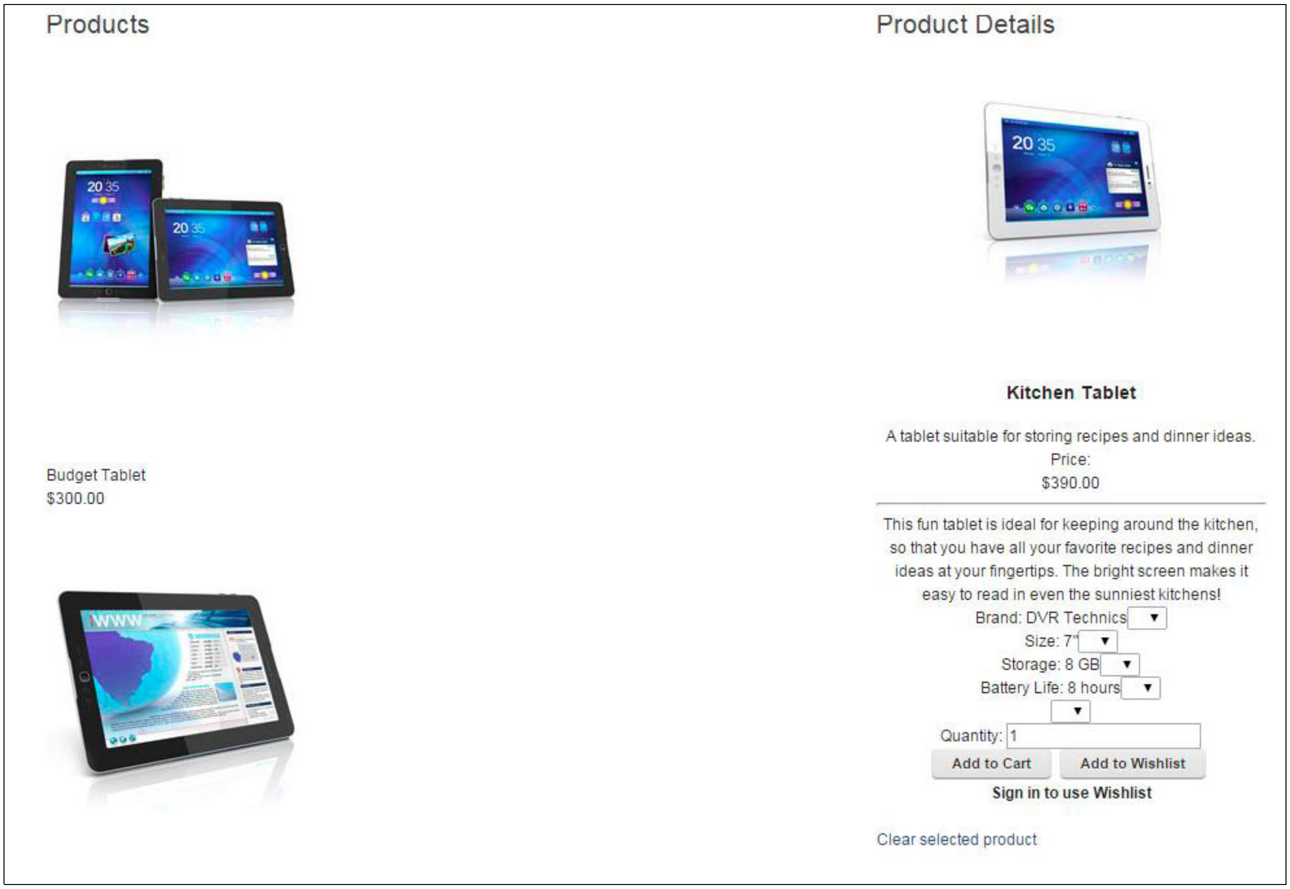


Figure 5-37 DDC WebSphere Commerce components in the WebSphere Portal toolbar

The sample components provide commonly used functions in a WebSphere Commerce server, such as product listings, search, and shopping cart.

To use these components, drop them on to your page. For example, Figure 5-38 shows how the Products and Product Details components are added to a page.

These components can be further customized to meet you website design requirements by customizing the HTML markup in the list presentation template, as shown in the previous examples of this section.



Products

Budget Tablet
$300.00

Product Details

Kitchen Tablet

A tablet suitable for storing recipes and dinner ideas.
Price:
$390.00

This fun tablet is ideal for keeping around the kitchen, so that you have all your favorite recipes and dinner ideas at your fingertips. The bright screen makes it easy to read in even the sunniest kitchens!

Brand: DVR Technics ▼
Size: 7" ▼
Storage: 8 GB ▼
Battery Life: 8 hours ▼
▼
Quantity: 1

Figure 5-38 Product and Product Details components added to a page

5.3 Integrating existing web applications with IBM Web Application Bridge

The Web Application Bridge is a feature of WebSphere Portal that integrates external (non-portal) web applications into the digital experience. With Web Application Bridge, you can turn almost any web application into a portlet. You can configure single sign-on to those remote applications, and change the markups that they return.

One advantage of using Web Application Bridge is that it requires no changes to the external application. All the business logic, navigation, routing, processing, and rendering remains unchanged, preserving the investment that you made in that application. Web Application Bridge repurposes that application to run as part of the overall digital experience.

Web Application Bridge uses a reverse proxy and iFrame technology. These components are part of the Web Application Bridge framework and are configured by using an administrative portlet provided with the feature. With this portlet, you can create and modify the portlets that render the remote application.

Web Application Bridge is installed along with WebSphere Portal V8.5. Updates can be found on the IBM Collaboration Solutions Catalog:

https://greenhouse.lotus.com/catalog/

Be sure to check this website for the latest version, as new features are added on a regular basis.

The following Web Application Bridge topics are covered in this section:

- 5.3.1, “Web Application Bridge components” on page 178
- 5.3.2, “Web Application Bridge architecture” on page 179
- 5.3.3, “Web Application Bridge use cases” on page 179
- 5.3.4, “Filtering” on page 180
- 5.3.5, “Web Application Bridge use case example” on page 182

5.3.1 Web Application Bridge components

This section provides an overview of the Web Application Bridge components.

Virtual Web Application Manager

The Virtual Web Application Manager component is a portlet that is installed in the Administration pages of WebSphere Portal. Administrators use this portlet to add applications, configure single sign-on, and perform other management tasks.

Web Dock portlet

Web Dock portlets are created by using the Virtual Web Application Manager. Each Web Dock portlet reflects an external web application that can then be deployed in a portal page. A portal site can have any number of Web Dock portlets. Each Web Dock portlet has a number of settings that can be configured to customize its display.

Reverse proxy

Web Application Bridge uses its own reverse proxy engine to access remote web applications. No additional portal configuration is required. Web Application Bridge uses the outbound HTTP connection service, which was introduced in IBM WebSphere Portal V8.5 to replace the Ajax Proxy, and creates the necessary HTTP connection configurations based on the settings that you provide in the Virtual Web Application Manager.

Web Application Bridge engine

The Web Application Bridge engine component provides the logic and resources to manage the other components of the feature.

Content provider

Content provider describes the external website that contains the application that you want to render as a portlet in WebSphere Portal. If you want to show an IBM BPM Coach in Web Application Bridge, for example, then IBM BPM is the content provider. A content provider can contain more than one application. Each application is often referred to as a *virtual web application*.

Content provider profile

A content provider profile contains the basic characteristics of a content provider. It has the URL of the content provider and a name that you assign.

Content provider profile policy

A content provider profile must have at least one profile policy defined to function. When you create a profile, a default policy is created for you, which is sufficient for basic functionality. However, a policy can also define allowable HTTP methods and MIME types, and other connection attributes. In addition, the policy can determine which URL patterns are allowed, and indicate whether filters should be applied to incoming or outgoing data. The policyis also where single sign-on is configured. Administrators do most of their work in a policy, so they should have a good understanding of what it can do for you.

5.3.2 Web Application Bridge architecture

Figure 5-39 shows the conceptual flow when a Web Application Bridge application is started.

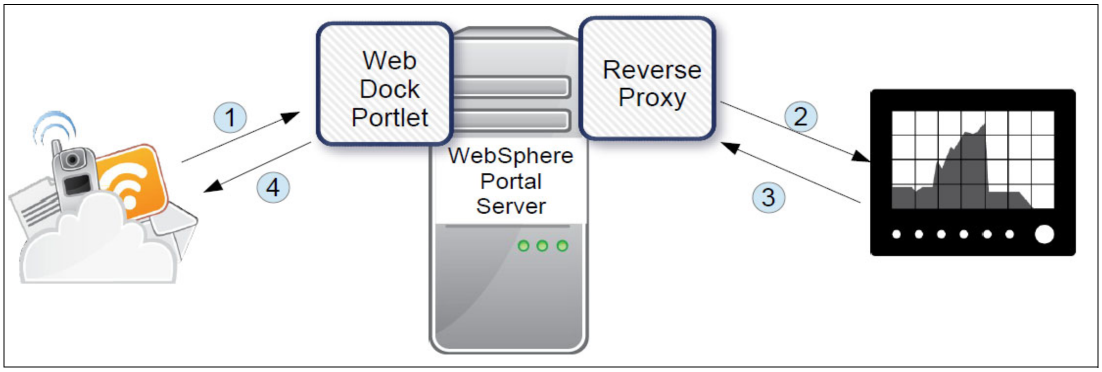


Figure 5-39 Web Application Bridge basic flow

Here is the flow that is depicted in Figure 5-39:

1. The client, for example, a browser or mobile device, starts a portal page that has a Web Dock portlet.
2. Web Application Bridge forwards the request to the content provider after applying any settings from the content provider profile, such as filters or HTTP headers.
3. The content provider returns a response back to the Web Application Bridge engine.
4. Web Application Bridge applies any settings from the content provider profile and forwards the response back to the client.

5.3.3 Web Application Bridge use cases

You should consider using Web Application Bridge when you are looking for a quick and simple way to integrate external web applications into the digital experience. An ideal application for this use case has the following characteristics:

- The application uses standard authentication technologies or does not require authentication at all. Web Application Bridge supports the following authentication approaches:
  - HTTP Basic
  - HTTP Digest

  - Form-based
  - SPNEGO
  - SAML
- The application can run inside an iFrame. Some applications try to detect whether they are in the top-most frame, and reload themselves into that top frame if they are not already there. Because Web Application Bridge renders the application in an iFrame, an ideal application does not have this limitation. However, there are ways to work around this problem.
- The application does not open other browser windows.

In the example scenario (Greenwheels demonstration website) that is used in this section (see 5.3.5, "Web Application Bridge use case example" on page 182), the objective is to create a portlet that modifies the Coach markup so it fits better with the appearance of the site and dynamically loads an IBM BPM Coach without a full-page refresh. Web Application Bridge is an excellent tool for this use case.

5.3.4 Filtering

Web Application Bridge can filter the HTTP requests and responses between the client and the external application. You can use this filtering to modify the information flowing to and from the remote application. This feature is valuable if the external application has certain characteristics that makes it difficult for that application to behave as a portlet. Consider an application that includes JavaScript code that prevents the application from being rendered in an iFrame. You can write a filter to remove or disable the JavaScript code.

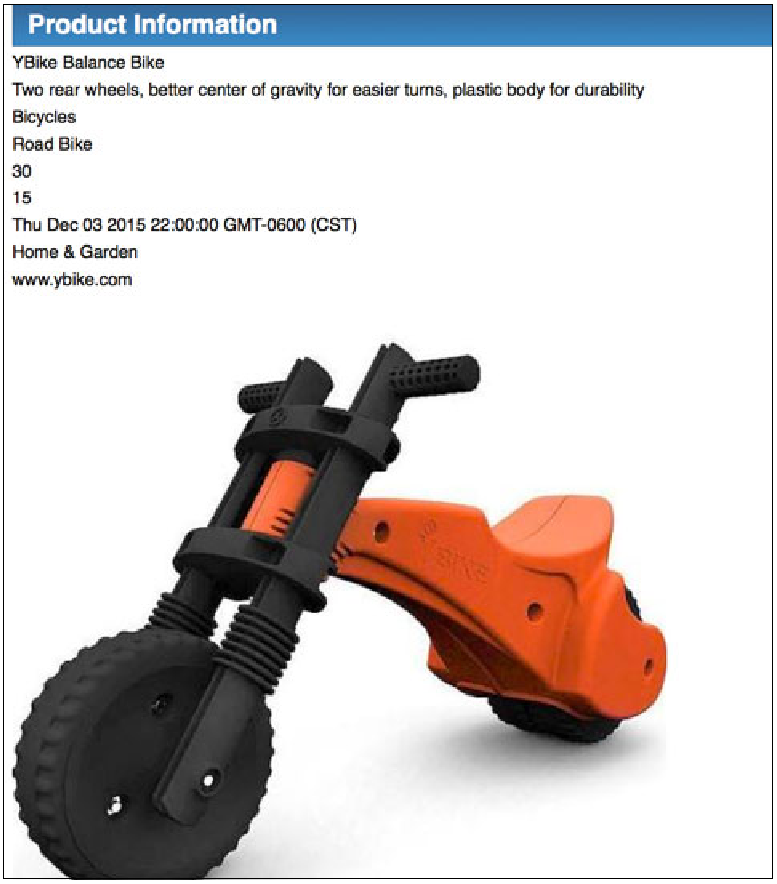
In addition, the appearance of the application can be modified before it is rendered in the browser, integrating it into the overall digital experience. You can remove HTML elements, such as logout buttons, which are not applicable while the application is running as a portlet. You can modify the styles to make the application look like it belongs in the portal.

All these modifications are being made to the markup, which is returned by the remote application, which means that the remote application itself is unchanged and can still serve users who access it from outside of the portal.

To implement Web Application Bridge filtering, you must complete the following steps:

1. Write Java code to perform the filtering.
2. Copy the code to the server.
3. Modify a Content Provider policy to use that filter.

In this example, a response filter is created to modify the markup of the Coach returned by the IBM BPM server. Figure 5-40 shows the original Coach.



Product Information

YBike Balance Bike

Two rear wheels, better center of gravity for easier turns, plastic body for durability

Bicycles

Road Bike

30

15

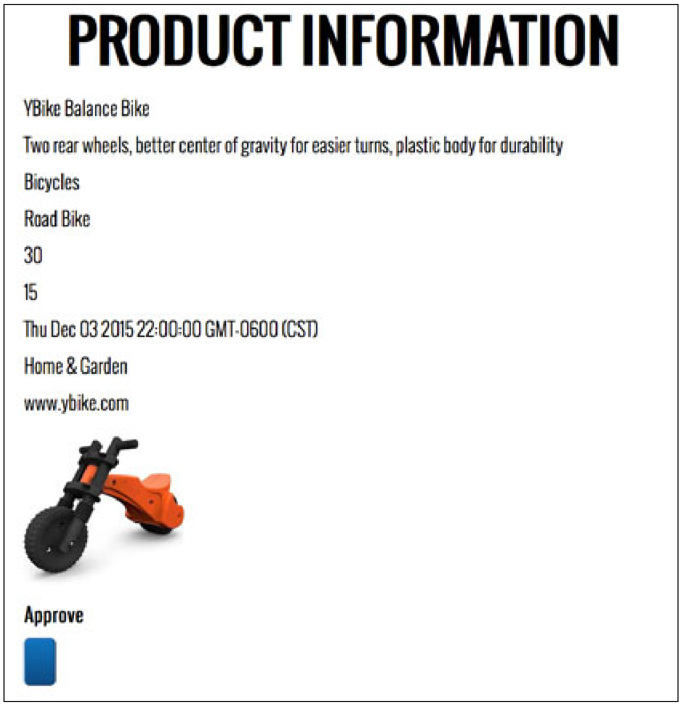
Thu Dec 03 2015 22:00:00 GMT-0600 (CST)

Home & Garden

www.ybike.com

Figure 5-40 Web Application Bridge Coach portlet before applying filtering

Figure 5-41 shows the coach after the changes.



PRODUCT INFORMATION

YBike Balance Bike

Two rear wheels, better center of gravity for easier turns, plastic body for durability

Bicycles

Road Bike

30

15

Thu Dec 03 2015 22:00:00 GMT-0600 (CST)

Home & Garden

www.ybike.com

Approve

Figure 5-41 Web Application Bridge Coach portlet after applying filtering

5.3.5 Web Application Bridge use case example

To implement the Web Application Bridge example that is described in this section, complete the following steps:

1. Create a Java class that is called `ITSO_CoachWABFilter`, which implements `javax.servlet.Filter`.

> **Note:** The complete example that is shown in this use case can be downloaded from the IBM Redbooks web server. The example is contained in the `ITSO_WAB_Filter.zip` file. For more information how to download this file, see Appendix C, “Additional material” on page 245.

2. Create a `doFilter` method as required by `javax.servlet.Filter`. This method is started when the filter is called.
3. In the `doFilter` method:
   a. Retrieve the output stream (the HTML markup) from the remote application.
   b. Convert the stream to a string.
   c. Modify the string.
   d. Return the modified string to be rendered by the portal.

Example 5-6 uses JSoup, an open source library for modifying HTML. You can use any library of your choice or modify the string directly. For more information about JSoup, see the JSoup Library website:

http://jsoup.org/

Example 5-6 Use JSoup to modify the markup that is returned by the remote application

```
      // parse HTML
      Document doc = Jsoup.parse(originalContent);

      // add a class to the body tag
      Element body = doc.getElementsByTag("body").get(0);
      body.addClass("centered");
      // add a stylesheet from our theme to the head section
      Element headTag = doc.getElementsByTag("head").get(0);
      Element linkEl = headTag.appendElement("link");
      linkEl.attr("type","text/css");
      linkEl.attr("rel","stylesheet");

linkEl.attr("href","http://itso-wp.ibmcollabcloud.com/wps/contenthandler/dav/fs-type1/themes/gre
enwell-responsive-V1_2/gw_module/css/doc.css");
      // add inline styles to the head section
      Element styleEl = headTag.appendElement("style");
      styleEl.text(".text img {height:100px;width:100px;}.centered {width:400px;margin-left:
auto;margin-right: auto;}");
      // replace text (in this case a set of styles) directly in the markup
      String modifiedContent = doc.toString().replaceFirst("BPMSectionHeader box blue",
"gw_sectionTitle");
      return modifiedContent;
```

4. After the filter code is complete, create a JAR file with all the code components and copy it to the portal server in the following location:

   ```
   <profile_dir>/installedApps/<cell_name>/wp.vwat.servlet.ear.ear/wp.vwat.servlet
   .war/WEB-INF/lib
   ```

   If your code requires other JAR files that are not on the portal class path, you can copy them into this directory as well.

   Note: For the Web Application Bridge engine to load the classes in this JAR file, you might need to stop and start the `wp.vwat.servlet.ear` application by using the WebSphere Integrated Solutions Console. After the first time, the application detects if you replaced this JAR file and automatically reloads it.

5. Log on to WebSphere Portal as an administrator. Go to the Virtual Web Application Manager, as shown in Figure 5-42.

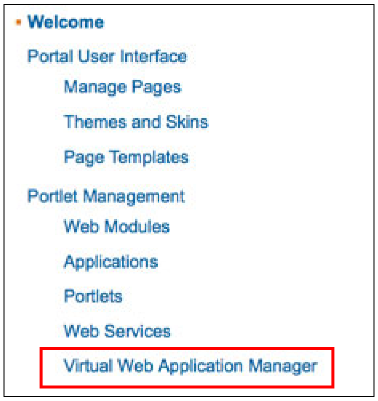
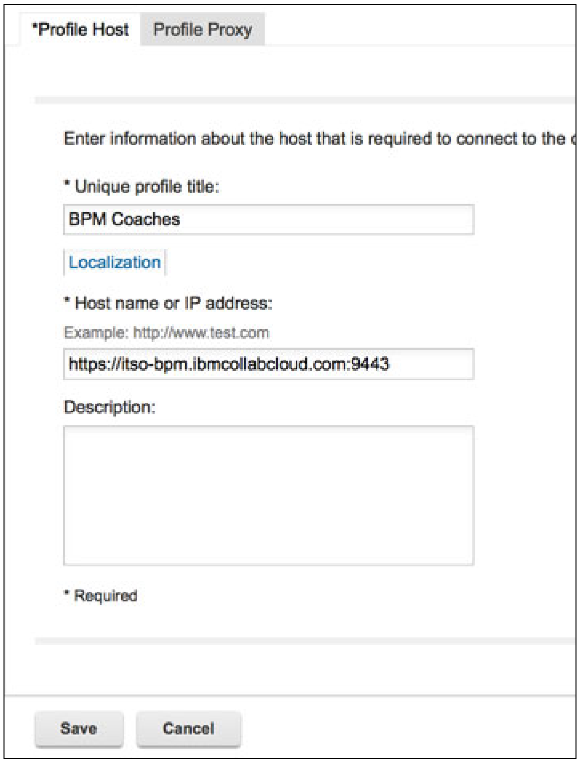


Figure 5-42 Access the Virtual Web Application Manager

6. Create a Content Provider profile, as shown in Figure 5-43, and click **Save**.



Enter information about the host that is required to connect to the c

* Unique profile title:
BPM Coaches

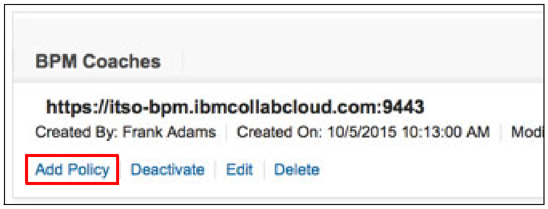
* Host name or IP address:
Example: http://www.test.com
https://itso-bpm.ibmcollabcloud.com:9443

Description:

* Required

Figure 5-43 Create a Content Provider profile

7. Click **Add Policy**, as shown in Figure 5-44.



BPM Coaches

https://itso-bpm.ibmcollabcloud.com:9443

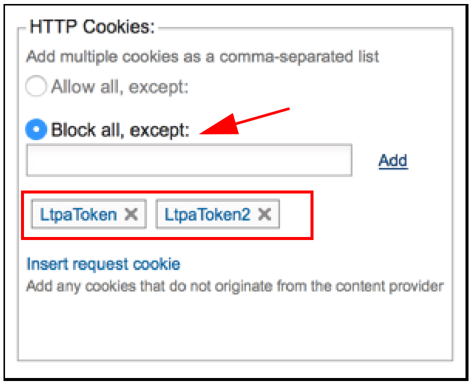
Created By: Frank Adams | Created On: 10/5/2015 10:13:00 AM | Modi

Figure 5-44 Add a policy

8. On Add Policy dialog box's **General** tab, give the policy a name.
9. Click the **Request** tab.
   a. In the HTTP Cookies section, click **Block All, except**, and enter LtpaToken. Click **Add**.
   b. Enter **LtpaToken2** and click **Add** again.

   Your window should look like the one that is shown in Figure 5-45.

> **Note:** By default, Web Application Bridge does not pass any cookies from the browser to the remote application. Adding these two cookies tells Web Application Bridge to pass them on. In this example, single sign-on is configured between WebSphere Portal and IBM BPM, which enables the IBM BPM server to identify the user.



HTTP Cookies:

Add multiple cookies as a comma-separated list

○ Allow all, except:

● Block all, except:

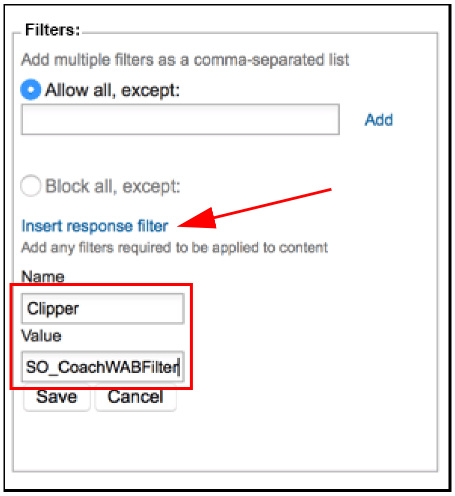
LtpaToken ✕ LtpaToken2 ✕

Insert request cookie

Add any cookies that do not originate from the content provider

Figure 5-45 Add the LtpaToken and LtpaToken2 cookies to the request

10. Click the **Response** tab. At the bottom of the form, click **Insert Response Filter** and provide a name and a value. In this example, the Name is the word Clipper and the Value is the full class name of the class you created previously. In this example, the class name is com.ibm.itso.wab.ITSO_CoachWABFilter. The form should look similar to Figure 5-46. Click **Save**.



Filters:

Add multiple filters as a comma-separated list

● Allow all, except:

○ Block all, except:

Insert response filter

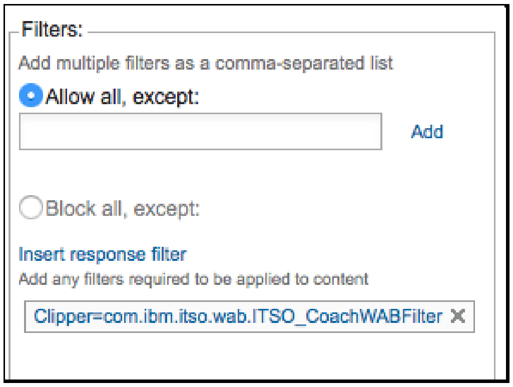
Add any filters required to be applied to content

Name: Clipper

Value: SO_CoachWABFilter

Figure 5-46 Add a filter to the policy

11. The filter should now look similar to the one that is shown in Figure 5-47.



Filters:

Add multiple filters as a comma-separated list

● Allow all, except:

○ Block all, except:

Insert response filter

Add any filters required to be applied to content

Clipper=com.ibm.itso.wab.ITSO_CoachWABFilter ✕

Figure 5-47 Web Dock filter configuration

12. Click **Save** again to save this policy.

You created a Content Provider profile. You also defined a policy to provide single sign-on and to filter the content that is returned by the remote applications started by this profile. The first part of this use case is completed, which is to modify the appearance of the coach portlet.

To start a coach without a full page refresh, you must use the interportlet communication feature of Web Application Bridge.

Interportlet communication

Web Application Bridge provides the ability for remote applications to communicate with portlets running on the portal server. A remote application can both receive messages from portlets and send messages to portlets. Web Application Bridge handles the interaction between the portlets, so no changes are required to the remote applications. For more information about interportlet communication, see the *Web Application Bridge inter-portlet communication* topic in the IBM WebSphere Portal V8.5.0 IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/help/panel_help/h_wab_ipc.dita?lang=en

For the Greenwheels demonstration website that is used as an example in this book, the objective is to create a Coach portlet that uses client-side interportlet communication to load dynamically a Coach without requiring a full-page refresh. This approach provides a better user experience than the full-page refresh, which is the default behavior of the Coach portlet that comes with the DDC components for IBM BPM.

This example shows how to create a Web Dock portlet and configure it to receive an OpenAjax message. It then shows how to modify the DDC Task portlet to post an OpenAjax message to the Web Dock portlet.

Complete the following steps:

1. Log on to WebSphere Portal as administrator. Click **Virtual Web Application Manager**, as shown in Figure 5-48.

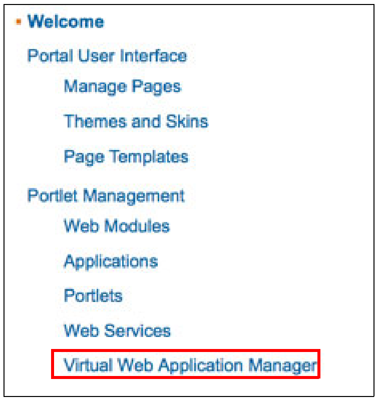


Figure 5-48 Select the Virtual Web Application Manager

2. Click **Web Dock Applications** → **Create Web Dock Application**.
3. On the Web Dock Application dialog box, enter the following information:
   - Enter a title for the application.
   - Select the Content Provider profile that you created in step 6 on page 184.
   - In the Resource path field, enter the location of a placeholder HTML file that is displayed when the portlet is first rendered. You must create this file and place it on the IBM BPM server in the following location:

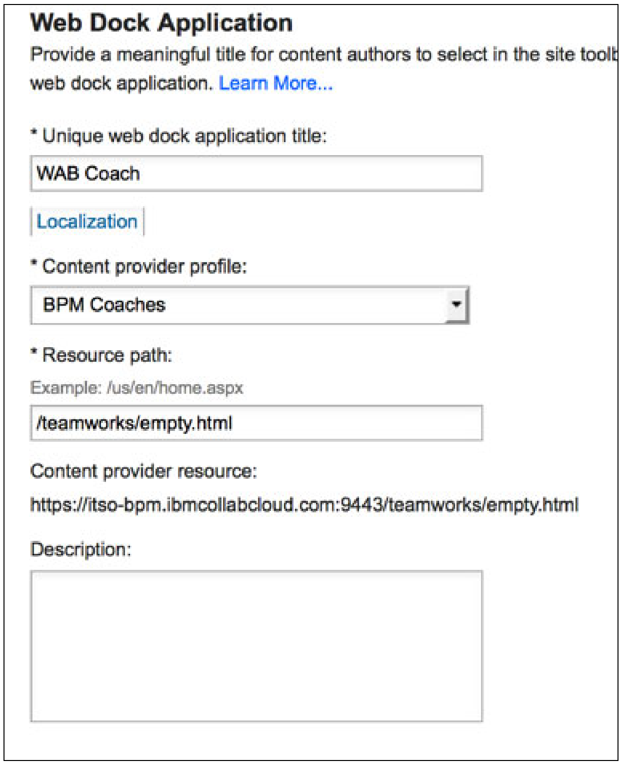
     `/opt/IBM/BPM/v8.5/profiles/<profile_name>/installedApps/<cell_name>/IBM_BPM_Teamworks_SingleCluster.ear/teamworks.war`

Example 5-7 shows the placeholder HTML file `empty.html` that is used in this example:

Example 5-7 Placeholder HTML file empty.html

```
<html>
<head></head>
<body></body>
</html>
```

The form should look like the one that is shown in Figure 5-49. Click **Save**.



Web Dock Application

Provide a meaningful title for content authors to select in the site toolt
web dock application. Learn More...

* Unique web dock application title:

WAB Coach

Localization

* Content provider profile:

BPM Coaches

* Resource path:

Example: /us/en/home.aspx

/teamworks/empty.html

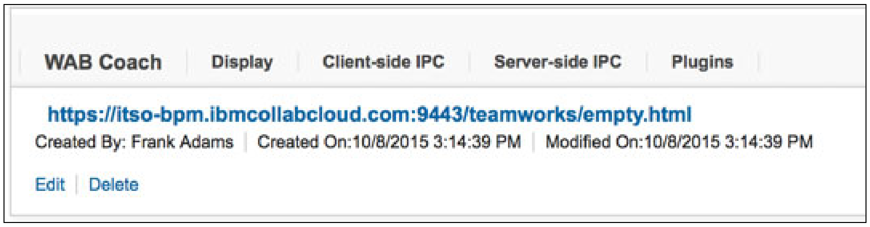
Content provider resource:

https://itso-bpm.ibmcollabcloud.com:9443/teamworks/empty.html

Description:

Figure 5-49 Create a Web Dock application

4. Your Web Dock application should look like Figure 5-50.



https://itso-bpm.ibmcollabcloud.com:9443/teamworks/empty.html

Created By: Frank Adams | Created On:10/8/2015 3:14:39 PM | Modified On:10/8/2015 3:14:39 PM

Figure 5-50 Completed Web Dock configuration

5. The Web Dock portlet in this example adjusts its size dynamically to accommodate the HTML markup from the Coach.

   You must add a parameter to indicate that the portlet should adjust the size. On the WebSphere Portal administration interface, expand **Portlet Management** and click **Portlets**, as shown in Figure 5-51.

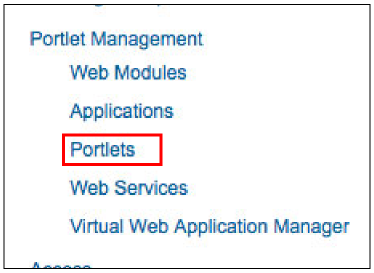
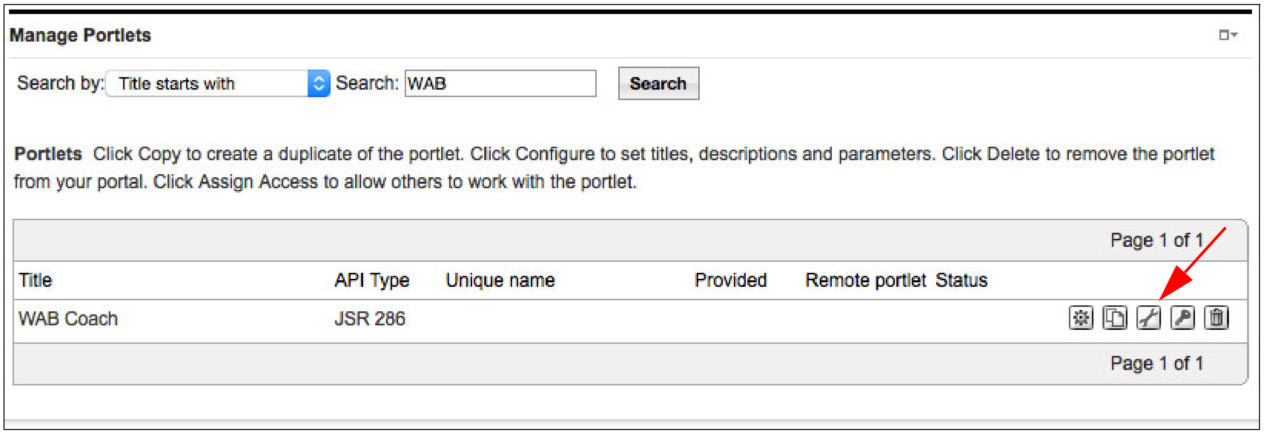


Figure 5-51 Select portlets administration

6. Search for your portlet in the list. The portlet name is the title you that specified in step 3 on page 187, which is Web Application Bridge Coach in this example. Click the wrench icon to configure the portlet, as shown in Figure 5-52.



Manage Portlets

Search by: Title starts with Search: WAB

Portlets Click Copy to create a duplicate of the portlet. Click Configure to set titles, descriptions and parameters. Click Delete to remove the portlet from your portal. Click Assign Access to allow others to work with the portlet.

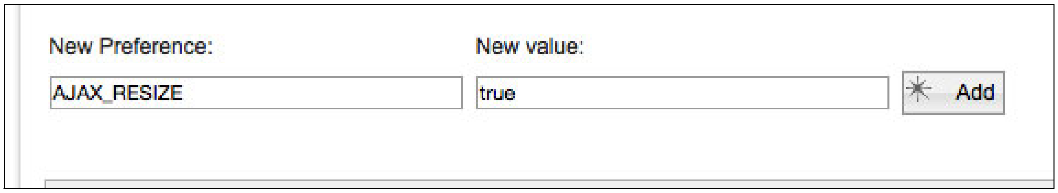
Page 1 of 1

Title	API Type	Unique name	Provided	Remote portlet	Status
WAB Coach	JSR 286				

Page 1 of 1

Figure 5-52 Clicking the Configure portlet icon

7. In the New Preference field, enter AJAX_RESIZE. In the Value field, enter true. Your values should look like the image that is shown in Figure 5-53. Click **Add**.

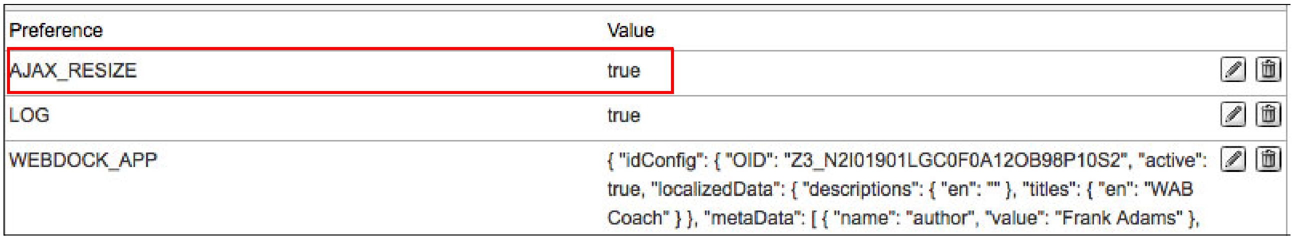


New Preference: AJAX_RESIZE

New value: true

Figure 5-53 Create a preference for this portlet

Figure 5-54 shows the AJAX_RESIZE preference in the list. Click **OK**. The portlet is now configured.



Preference	Value
AJAX_RESIZE	true
LOG	true
WEBDOCK_APP	{ "idConfig": { "OID": "Z3_N2I01901LGC0F0A12OB98P10S2", "active": true, "localizedData": { "descriptions": { "en": "" }, "titles": { "en": "WAB Coach" } }, "metaData": [{ "name": "author", "value": "Frank Adams" },

Figure 5-54 Portlet that is configured for dynamic resizing

8. Add the portlet to the page that has the Tasks portlet (for more information about adding a portlet to a page, see step 3 on page 145) and then configure it on that page.
9. Edit the Web Dock portlet settings. Click the drop-down menu in the upper right and select **Edit Shared Settings**, as shown in Figure 5-55.

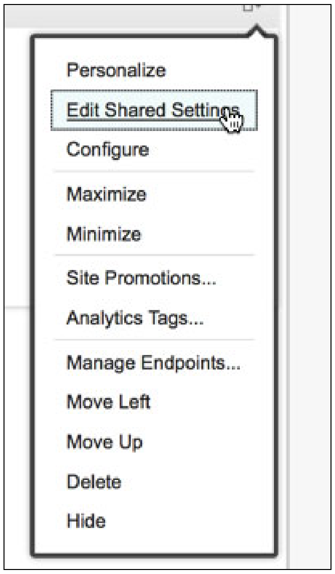
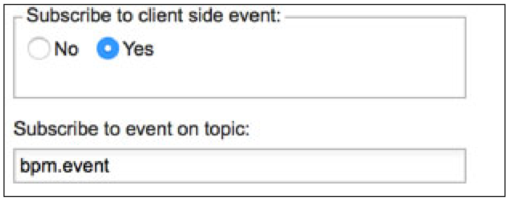


Figure 5-55 Edit the settings of the Web Dock portlet

10. On the Display tab, make sure that Dynamic Size is set to **Always**.
11. Click the **Client Side IPC** tab. There are many settings in this tab. The settings that are relevant to this example are shown in Figure 5-56.
    - Select **Yes** for Subscribe to client-side event.
    - In the Subscribe to event on topic field, enter bpm.event.

    Click **Save**. The Web Dock portlet is configured.



Subscribe to client side event:

○ No ● Yes

Subscribe to event on topic:

bpm.event

Figure 5-56 Subscribe to the client-side event

12. Change the DDC Task portlet to send an OpenAjax message. Because all DDC components are in Web Content Manager, use the Web Content Manager Authoring Porltet to find and edit the appropriate components.

    An examination of the Tasks content item shows that it uses a component that is called *my tasks*, which can be found in the IBM BPM library by clicking **BPM** → **Components** → **Scripts** → **my tasks**. Use the Authoring Portlet to go to this component and open it in Edit mode.

13. Insert the code that is shown in Example 5-8 just before the closing `</script>` tag.

> **Note:** When the Web Dock portlet was configured, `bpm.event` was specified as the event name. The portlet adds as a prefix the value `com.ibm.vwat.event` to the event name. Therefore, it listens for the topic `com.ibm.vwat.event.bpm.event`.

Save your changes. This JavaScript function is now available in the portlet.

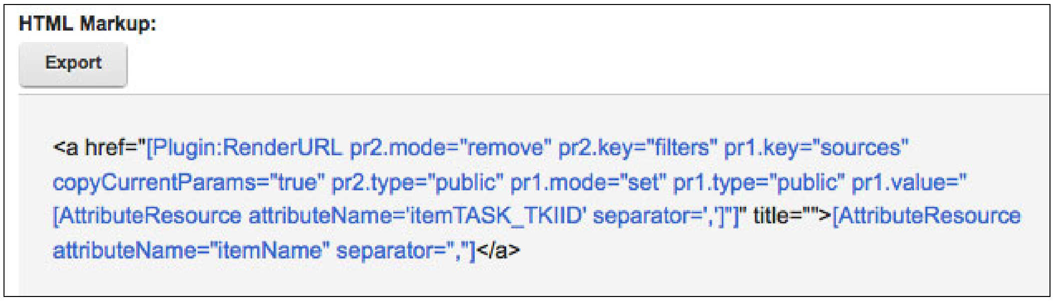
Example 5-8 JavaScript code to send an OpenAjax message

```
function passToWABPortlet(taskID)
{
   // we could get these two values from WCM
   var bpm_host = "itso-bpm.ibmcollabcloud.com";
   var bpm_port = "9443";
   // construct Coach URL
   var coachURL = "https://" + bpm_host + ":" + bpm_port +
"/teamworks/process.lsw?zWorkflowState=1&zResetContext=true&zTaskId=" + taskID;
   OpenAjax.hub.publish("com.ibm.vwat.event.bpm.event", coachURL);
}
```

14. Modify the link that starts the Coach portlet.

    Click **BPM** → **Components** → **List Appearances** → **Item Name**. You can see the markup for the name link, as shown in Figure 5-57.

> **Note:** This markup creates a link back to the Portal server in to set a public render parameter, and that is what requires a full page refresh. Public render parameters are part of the URL, so the URL must change every time a different task name is clicked.



```
<a href="[Plugin:RenderURL pr2.mode="remove" pr2.key="filters" pr1.key="sources"
copyCurrentParams="true" pr2.type="public" pr1.mode="set" pr1.type="public" pr1.value="
[AttributeResource attributeName='itemTASK_TKIID' separator=',']"]" title="">[AttributeResource
attributeName="itemName" separator=","]</a>
```

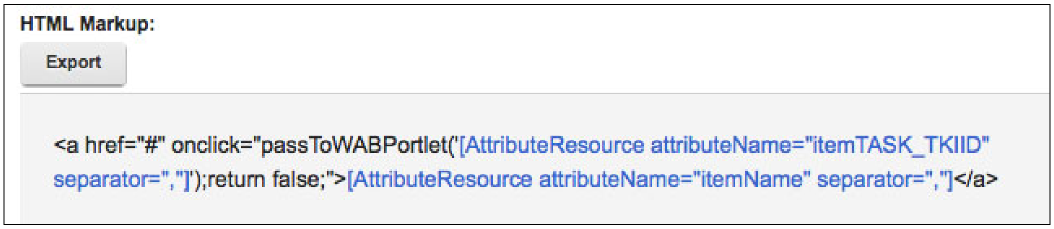
Figure 5-57 The original item name link

15. Click **Edit** to put the component into edit mode. Replace the entire content of the HTML element with the code that is shown in Example 5-9.

Example 5-9 HTML for the Item Name component

```
<a href="#" onclick="passToWABPortlet('[AttributeResource
attributeName="itemTASK_TKIID" separator=","]');return
false;">[AttributeResource attributeName="itemName" separator=","]</a>
```

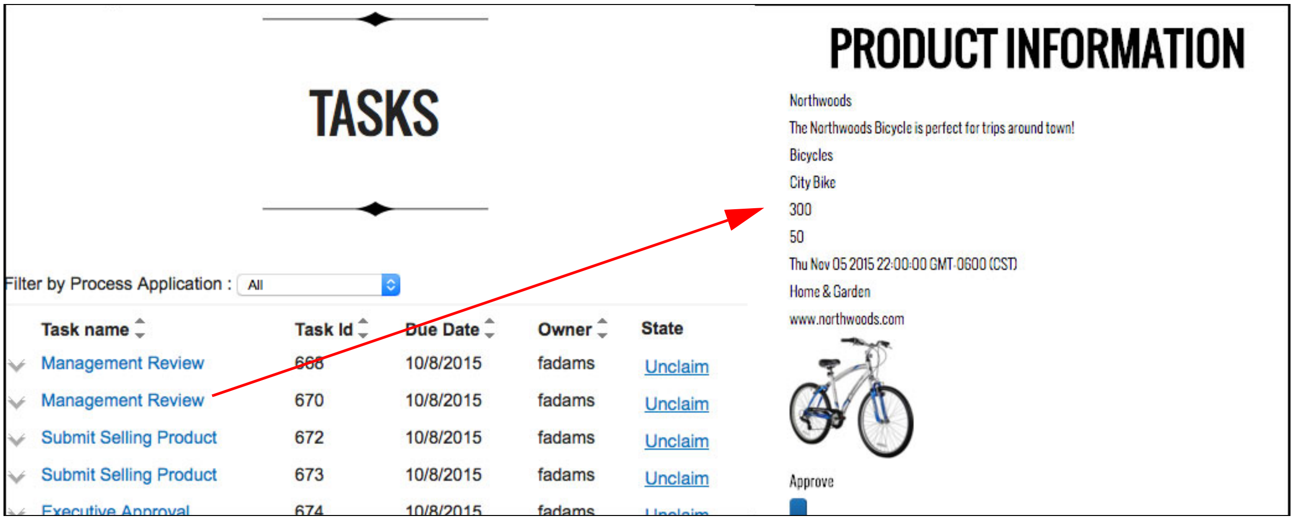
Save your changes. Figure 5-58 shows the resulting HTML element.



```
<a href="#" onclick="passToWABPortlet('[AttributeResource attributeName="itemTASK_TKIID"
separator=","]');return false;">[AttributeResource attributeName="itemName" separator=","]</a>
```

Figure 5-58 Item Name component after changes

16. You complete the necessary changes to the DDC components and the configured Web Dock portlet on the same page as the Task portlet. Test your changes. Figure 5-59 shows the results. When you click a task name link, the appropriate coach appears, filtered to fit into the appearance of the website without a full-page refresh.



Task name	Task Id	Due Date	Owner	State
Management Review	668	10/8/2015	fadams	Unclaim
Management Review	670	10/8/2015	fadams	Unclaim
Submit Selling Product	672	10/8/2015	fadams	Unclaim
Submit Selling Product	673	10/8/2015	fadams	Unclaim
Executive Approval	674	10/8/2015	fadams	Unclaim

Northwoods
The Northwoods Bicycle is perfect for trips around town!
Bicycles
City Bike
300
50
Thu Nov 05 2015 22:00:00 GMT-0600 (CST)
Home & Garden
www.northwoods.com

Figure 5-59 Click the task name in the Tasks portlet to open the appropriate coach portlet

5.4 Using the IBM Script portlet

The IBM Script portlet enables quick and easy portlet development for WebSphere Portal by using web development skills such as HTML, CSS, and JavaScript.

The following IBM Script portlet topics are covered in this section:

- 5.4.1, “Overview of the IBM Script portlet” on page 193
- 5.4.2, “High-level architecture of the IBM Script portlet” on page 193
- 5.4.3, “Use cases for the IBM Script portlet” on page 194
- 5.4.4, “Example: Integrating IBM BPM data” on page 194

5.4.1 Overview of the IBM Script portlet

The following items summarize the IBM Script portlet approach:

- All the application code (HTML, JavaScript, and CSS) is stored and managed in Web Content Manager.
- From a browser, you can work in the IBM Script portlet editor, by using syntax highlighting and auto-indent.
- Applications can be initially developed by using your tool of choice, then imported into IBM Script portlet.
- Data access is done with Ajax and REST services by using JSON data. These services can come from any provider, whether from an external one that uses a portal Ajax proxy or from Web Experience Factory or other provider.
- The key added value features of WebSphere Portal are available. Features such as inter-portlet communications, portlet preferences, responsive and adaptive design, and public render parameters are available.

Here are some of the key benefits of the IBM Script portlet approach:

- You can build portlets by using skills that you can easily find in the marketplace, for example, JavaScript, HTML, CSS, and jQuery.
- Line-of-business (LOB) professionals are less dependent on central IT. A small team of script programmers can make new portlets and customizations quickly and easily.
- You can build portlets without deploying any new code on the server.
- You can create a portlet in minutes simply by accessing a cloud server from a browser without installing any tools, for example, by copying and pasting JavaScript and HTML snippets.
- All the features of Web Content Manager can be used for managing applications:
  - Projects and workflow can be used for approval by business or IT before applications go live.
  - All coding changes can be scoped to the same project as other changes to web content or portal-managed content.
  - You can create libraries of JavaScript, CSS, data, or markup, and link to them from within the WebSphere Content Manager user interface.

5.4.2 High-level architecture of the IBM Script portlet

Using the IBM Script portlet, web developers can build standard web applications that run along other content and applications as part of the complete digital experience website. There is no Java development or server code deployment that is needed and you can use your favorite libraries and frameworks, such as Angular, Bootstrap, Handlebars, Backbone, or jQuery.

Also, you can use your favorite modern web development tools such text editors, `Node.js` tools (Bower, grunt/gulp, Yeoman, and so on), and any other tools for working with HTML, JavaScript, or CSS.

Applications can be instantly updated on a local or remote server over HTTP and tested in a stand-alone fashion.

Figure 5-60 shows an overview of the capabilities of the IBM Script portlet.

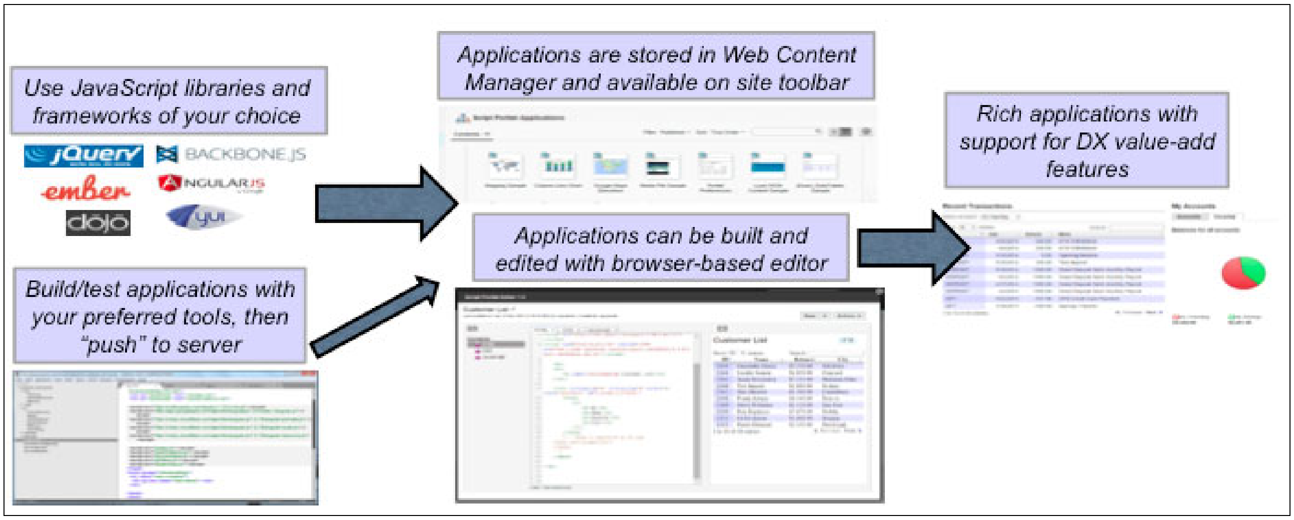

Use JavaScript libraries and frameworks of your choice
jQuery
BACKBONE.JS
ember
ANGULARJS
dojo
YUI
Build/test applications with your preferred tools, then "push" to server
Applications are stored in Web Content Manager and available on site toolbar
Applications can be built and edited with browser-based editor
Rich applications with support for DX value-add features

Figure 5-60 IBM Script portlet overview

5.4.3 Use cases for the IBM Script portlet

Typical use cases for the IBM Script portlet include the following types of applications:

- Any type of application user interface
- Forms, views, charts, and so on
- Single-page applications that use any JavaScript libraries and frameworks accessing data with REST or JSON services
- Client-side applications

Section 5.4.4, "Example: Integrating IBM BPM data" on page 194 provides an example of using the IBM Script portlet to integrate IBM BPM data.

5.4.4 Example: Integrating IBM BPM data

Accessing IBM BPM data from an IBM Script portlet is straightforward. IBM BPM V8.5.x provides a comprehensive REST interface that developers can use to start easily IBM BPM tasks and receive the results in either XML or JSON.

For more information about the REST interface, see the *IBM Business Process Manager REST Interface* topic in the IBM WebSphere Portal V8.5.0 IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.6/com.ibm.wbpm.ref.doc/rest/bpmrest/index.htm

IBM BPM includes a test tool, the REST API Tester, which is a web application that developers cause to construct and test the REST calls. Developers can use this tool to create a URL that they can then copy into their applications, knowing that the URL is correctly constructed, and that it returns the expected results. The URL can also be run directly from the tool so that developers can examine the results of the call before adding it to their applications.

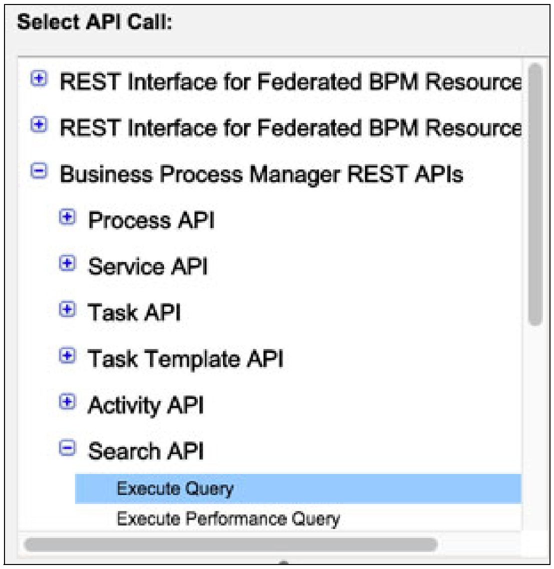
The REST API Tester can be accessed from the following URL:

```
https://<BPM host name>:9443/bpmrest-ui
```

In this example, managers want to see a report of the various active tasks for the process called *Vendor OnBoarding*. They want to see how many active tasks there are in total, and how many tasks are in each step. This query can be created by using the REST API Tester.

Complete the following steps:

1. Log on to the REST API Tester.
2. From the Select API Call window, expand **Business Process Manager REST APIs** → **Search API** and click **Execute Query**, as shown in Figure 5-61.



Select API Call:
REST Interface for Federated BPM Resource
REST Interface for Federated BPM Resource
Business Process Manager REST APIs
Process API
Service API
Task API
Task Template API
Activity API
Search API
Execute Query
Execute Performance Query

Figure 5-61 REST API Tester

3. In the first Condition field in the lower left pane of the browser, enter the following string to specify that the query should return only tasks from your process. In this example, the process is called Vendor OnBoarding.

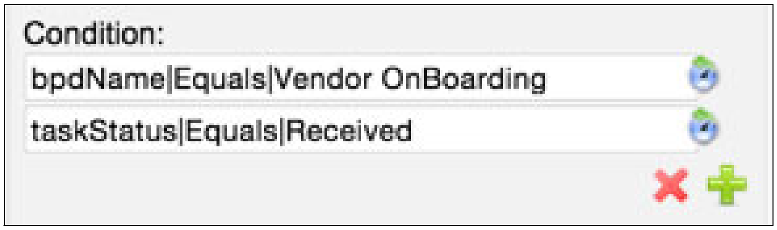
   ```
   bpdName|Equals|Vendor OnBoarding
   ```

4. Click the green plus icon below the Condition field to add a second field. In the second field, enter the following string to specify that the query should return only tasks with a status of Received:

   ```
   taskStatus|Equals|Received
   ```

   Note: In IBM BPM, a task can have a status of Completed or Received. Received means that the task is assigned, but no one has taken action on it yet.

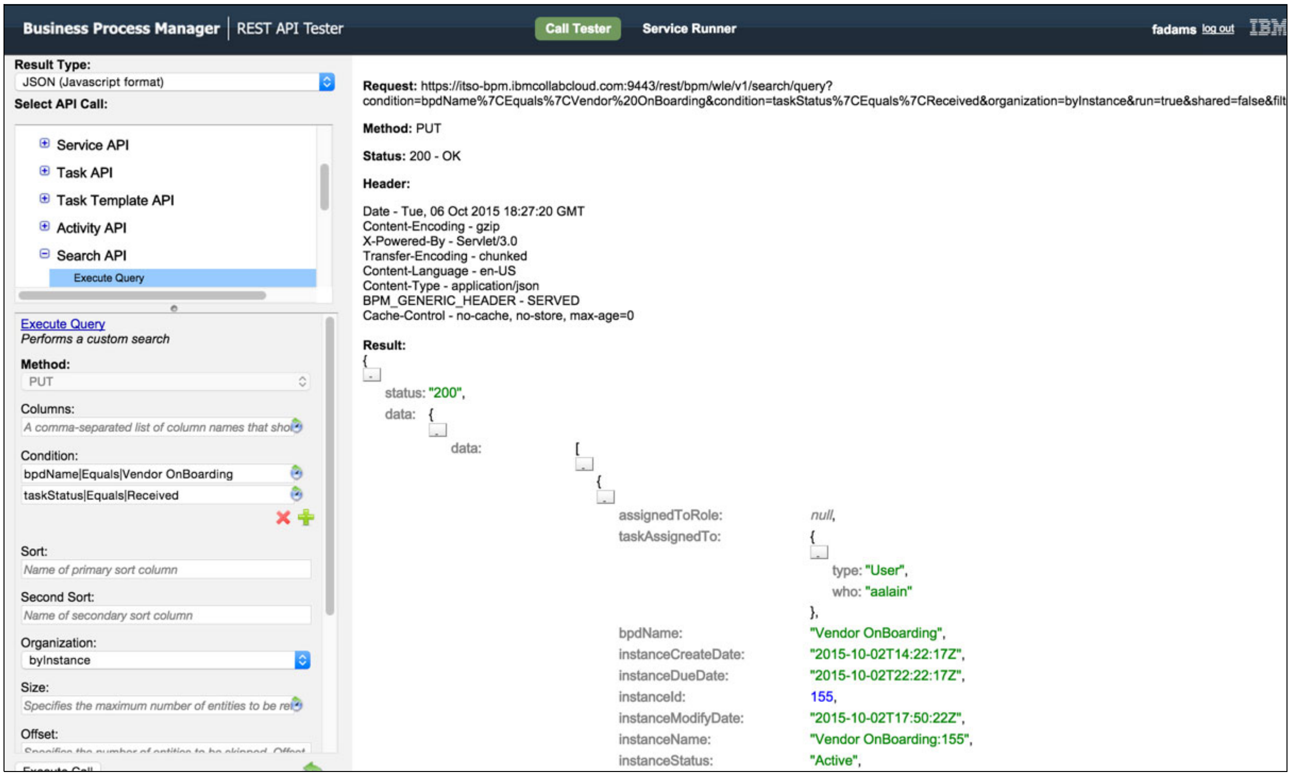
   Figure 5-62 shows the condition fields for this example.



Condition:
bpdName|Equals|Vendor OnBoarding
taskStatus|Equals|Received

Figure 5-62 Conditions

5. Clear the **filterByCurrentUser** check box. You want to see all tasks, not just those tasks that are associated with the logged-in user.
6. Click **Execute Call**. Figure 5-63 shows the results.



Result Type: JSON (Javascript format)

Select API Call:

Execute Query
Performs a custom search

Method: PUT

Columns: *A comma-separated list of column names that sho*

Condition: bpdName|Equals|Vendor OnBoarding
taskStatus|Equals|Received

Sort: *Name of primary sort column*

Second Sort: *Name of secondary sort column*

Organization: byInstance

Size: *Specifies the maximum number of entities to be re*

Offset: *Specifies the number of entities to be skipped. Offset*

Execute Call

Request: https://itso-bpm.ibmcollabcloud.com:9443/rest/bpm/wle/v1/search/query?condition=bpdName%7CEquals%7CVendor%20OnBoarding&condition=taskStatus%7CEquals%7CReceived&organization=byInstance&run=true&shared=false&filt

Method: PUT

Status: 200 - OK

Header:

```
Date - Tue, 06 Oct 2015 18:27:20 GMT
Content-Encoding - gzip
X-Powered-By - Servlet/3.0
Transfer-Encoding - chunked
Content-Language - en-US
Content-Type - application/json
BPM_GENERIC_HEADER - SERVED
Cache-Control - no-cache, no-store, max-age=0
```

Result:

```
{
  status: "200",
  data:  {
         data:      [
                      {
                        assignedToRole:       null,
                        taskAssignedTo:       {
                                                 type: "User",
                                                 who: "aalain"
                                              },
                        bpdName:              "Vendor OnBoarding",
                        instanceCreateDate:   "2015-10-02T14:22:17Z",
                        instanceDueDate:      "2015-10-02T22:22:17Z",
                        instanceId:           155,
                        instanceModifyDate:   "2015-10-02T17:50:22Z",
                        instanceName:         "Vendor OnBoarding:155",
                        instanceStatus:       "Active",
```

Figure 5-63 IBM BPM REST API Tester results

7. The URL that was constructed by the IBM BPM REST API Tester is at the top of the window on the right (it is an HTTP PUT). Portlet developers can use this URL in their server-side code with no changes.

   However, to use it in an IBM Script portlet, you must go through the Ajax proxy. The proxy already is configured, as described in 5.2.5, “Use case 2: Integrating with IBM Business Process Manager” on page 151.

   Change the URL to use the proxy as follows:

   – Before:

   `https://itso-bpm.ibmcollabcloud.com:9443/rest/bpm/wle/v1/search/query?....`

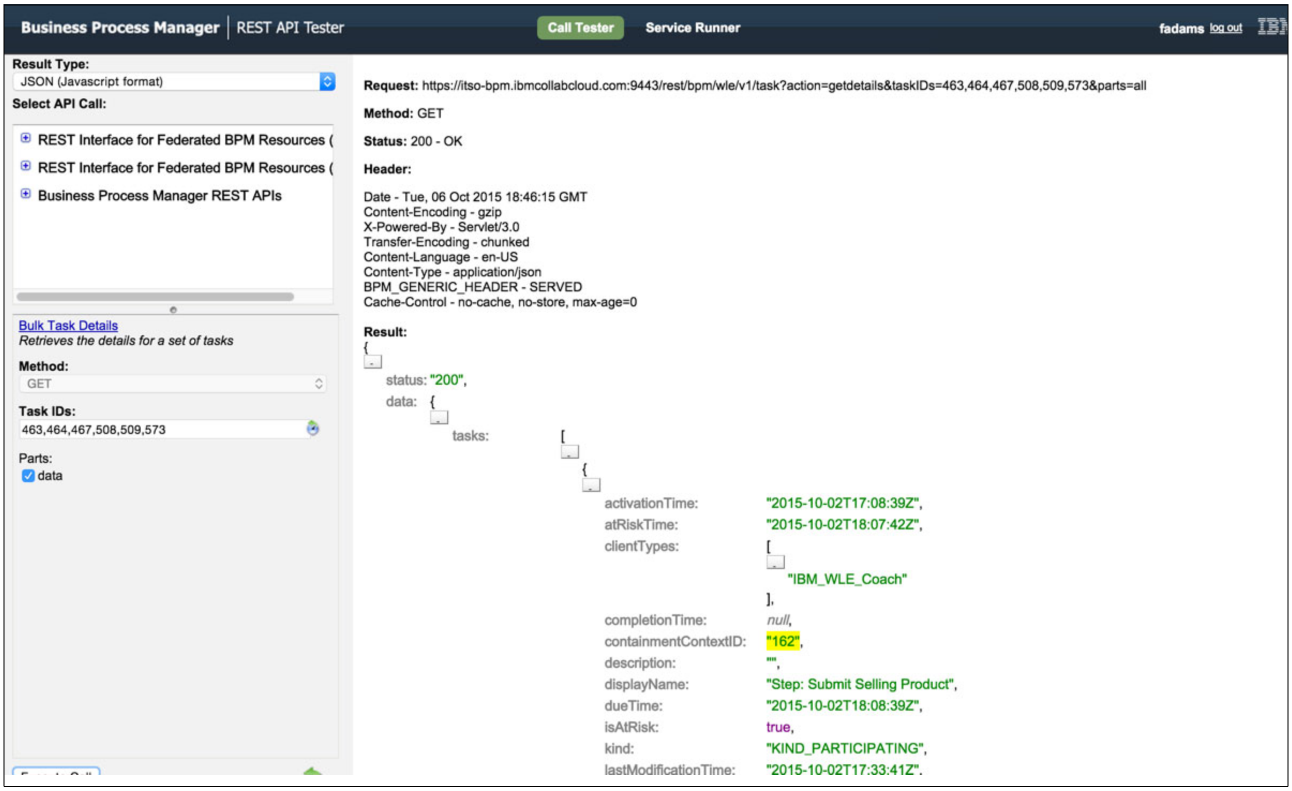
   – After:

   `http://itso-wp.ibmcollabcloud.com/wps/proxy/https``/itso-bpm.ibmcollabcloud.com:9443/rest/bpm/wle/v1/search/query?...`

   Now, the URL can be used in JavaScript running in a portlet.

8. One more REST API is needed to return task details for all tasks in a chosen step. IBM BPM has an API for this request called *Bulk Task Details*. You can find it under the Task API node in the Select API Call window at the upper left of the browser. Test it out by entering a few valid Task IDs (you can find them in the JSON results of the previous call you created).

   Figure 5-64 shows the results that you should see when you run this call.


Business Process Manager
REST API Tester
Call Tester
Service Runner
fadams
log out
REST Interface for Federated BPM Resources (
REST Interface for Federated BPM Resources (
Business Process Manager REST APIs

Result Type: JSON (Javascript format)

Select API Call:

Bulk Task Details
Retrieves the details for a set of tasks

Method: GET

Task IDs: 463,464,467,508,509,573

Parts: ☑ data

Request: https://itso-bpm.ibmcollabcloud.com:9443/rest/bpm/wle/v1/task?action=getdetails&taskIDs=463,464,467,508,509,573&parts=all

Method: GET

Status: 200 - OK

Header:

```
Date - Tue, 06 Oct 2015 18:46:15 GMT
Content-Encoding - gzip
X-Powered-By - Servlet/3.0
Transfer-Encoding - chunked
Content-Language - en-US
Content-Type - application/json
BPM_GENERIC_HEADER - SERVED
Cache-Control - no-cache, no-store, max-age=0
```

Result:

```
{
   status: "200",
   data:  {
          tasks:        [
                           {
                             activationTime:         "2015-10-02T17:08:39Z",
                             atRiskTime:             "2015-10-02T18:07:42Z",
                             clientTypes:            [
                                                        "IBM_WLE_Coach"
                                                     ],
                             completionTime:         null,
                             containmentContextID:   "162",
                             description:            "",
                             displayName:            "Step: Submit Selling Product",
                             dueTime:                "2015-10-02T18:08:39Z",
                             isAtRisk:               true,
                             kind:                   "KIND_PARTICIPATING",
                             lastModificationTime:   "2015-10-02T17:33:41Z",
```

Figure 5-64 IBM BPM REST API Tester results

9. If you look through the JSON that is returned, you can see that the call returns the data that is entered by users of this process. This is the expected result, so these two queries are enough to complete the use case. In order for a user to see the details of all tasks, that user must be associated with those task steps in IBM BPM.

> **Note:** The sample IBM Script portlet that is developed for this use case uses these two queries to create a dashboard showing all active tasks for the chosen process, with a drill-down into the details of all tasks in a selected step. This sample code can be downloaded from the IBM Redbooks web server.
>
> The example is contained in the `SP_BPM_Analysis.zip` file. For more information how to download this file, see Appendix C, "Additional material" on page 245.
>
> The sample contains a JavaScript file that is called `script.js`, which does most of the work. This script uses JQuery, so you can use any JavaScript framework.

When the page loads, the code to start the REST call is performed, as shown in Example 5-10.

Example 5-10 JavaScript code to start the IBM BPM REST call

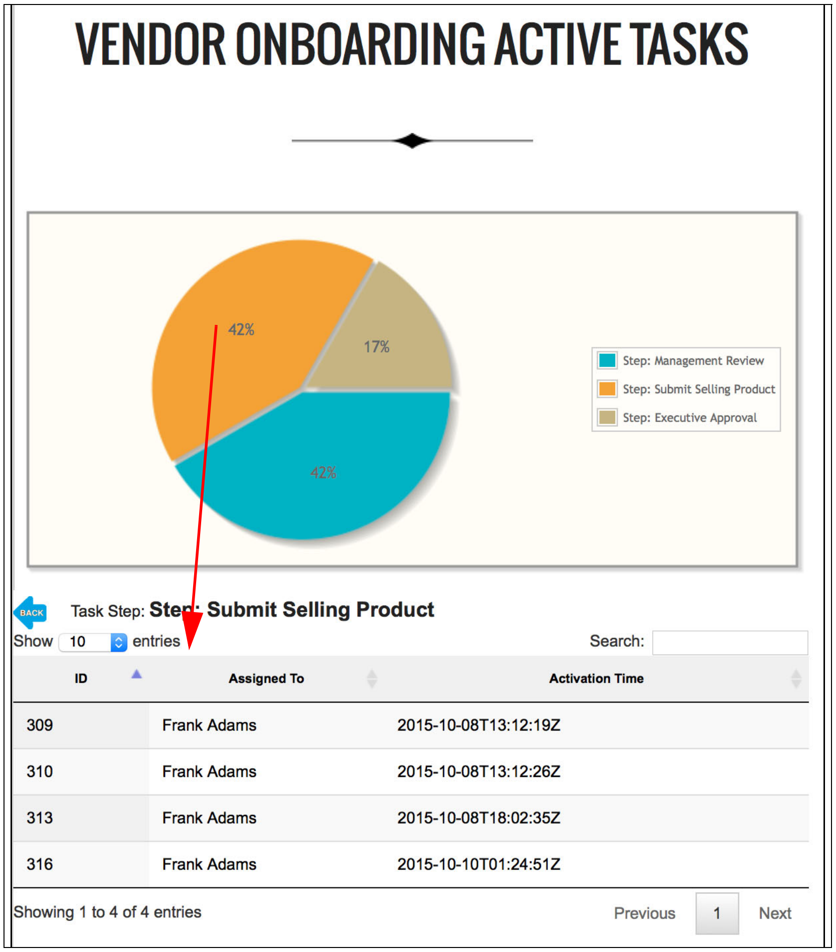
```
$( document ).ready(function() {

   // call the BPM query to get active tasks for this process
    var taskTable;
    var theURL =
"/wps/proxy/https/itso-bpm.ibmcollabcloud.com:9443/rest/bpm/wle/v1/search/query?co
ndition=bpdName%7CEquals%7CVendor%20Onboarding&condition=taskStatus%7CEquals%7CRec
eived&organization=byInstance&run=true&shared=false&filterByCurrentUser=false";
   $.ajax({
     method: "PUT",
     url: theURL,
     contentType: "application/json"
   })
   .error(function(results){
      alert('an error has occurred');
   })
     .done(function( results ) {
        showChart(results.data.data);
   });

});
```

Figure 5-65 shows the final result of the IBM Script portlet scenario fetching data from IBM BPM. You can see the pie chart rendering requests in different stages of the business process. When you click a stage section, you can drill down to the list of tasks in that stage.

As an additional function, you can then enable the task list to send the URL to the respective coach rendering portlet to allow directly the user to work with and complete the human tasks.


VENDOR ONBOARDING ACTIVE TASKS
42%
17%
42%
🟦 Step: Management Review
🟧 Step: Submit Selling Product
🟫 Step: Executive Approval
BACK
Show 10 entries
Search:
Previous
1
Next

Task Step: **Step: Submit Selling Product**

ID	Assigned To	Activation Time
309	Frank Adams	2015-10-08T13:12:19Z
310	Frank Adams	2015-10-08T13:12:26Z
313	Frank Adams	2015-10-08T18:02:35Z
316	Frank Adams	2015-10-10T01:24:51Z

Showing 1 to 4 of 4 entries

Figure 5-65 Vendor Onboarding active tasks

5.5 Integrating Forms Experience Builder

Business users can use Forms Experience Builder to build and deploy rich self-service web form applications in a point-and-click fashion with nothing more than a web browser. Business users who are comfortable creating spreadsheets can quickly learn and use this tool.

The following Forms Experience Builder topics are covered in this section:

- 5.5.1, “Overview of Forms Experience Builder” on page 200
- 5.5.2, “High-level architecture of Forms Experience Builder” on page 200
- 5.5.3, “Use cases for Forms Experience Builder” on page 201
- 5.5.4, “Example: Creating a registration form for your website” on page 202

5.5.1 Overview of Forms Experience Builder

Applications are generated based on a modern web infrastructure of HTML, JavaScript, and CSS. Each form automatically generates its own secure database, eliminating the need for special skills or IT staff involvement. Business users can easily specify workflow logic and access control for automating processes. You can deploy forms as stand-alone web applications, through a portlet by using WebSphere Portal software, to mobile devices, or even as part of a community survey in IBM Connections.

You can deploy Forms Experience Builder as a central self-service environment so that all of your business users can create and deploy their own applications. Users design, deploy, and access their application with nothing more than a browser. Deployed in this fashion, Forms Experience Builder can support a vast number of business users. Business users can decide who has access to the applications they design and deploy and even the time frame in which they want to make the applications available.

You can extend the Forms Experience Builder environment by exposing other systems and data as services for business users to map into the applications they build. For example, IT can expose a service that lets users find out which products a customer purchased. With this service, business users can design a form that lets them select a customer from a drop-down menu and see a list of products that the customer purchased, or perhaps the business users build a requisition form that lets an employee look up a manager and email address for routing a form for approval. AFter IT exposes services to the Forms Experience Builder catalog, business users can easily integrate them into compelling web forms over and over again, without the aid of IT.

5.5.2 High-level architecture of Forms Experience Builder

Figure 5-66 shows how Forms Experience Builder creates a complete web application in addition to the form user interface that includes access control, workflow, reporting, and database capabilities in a single deployment.

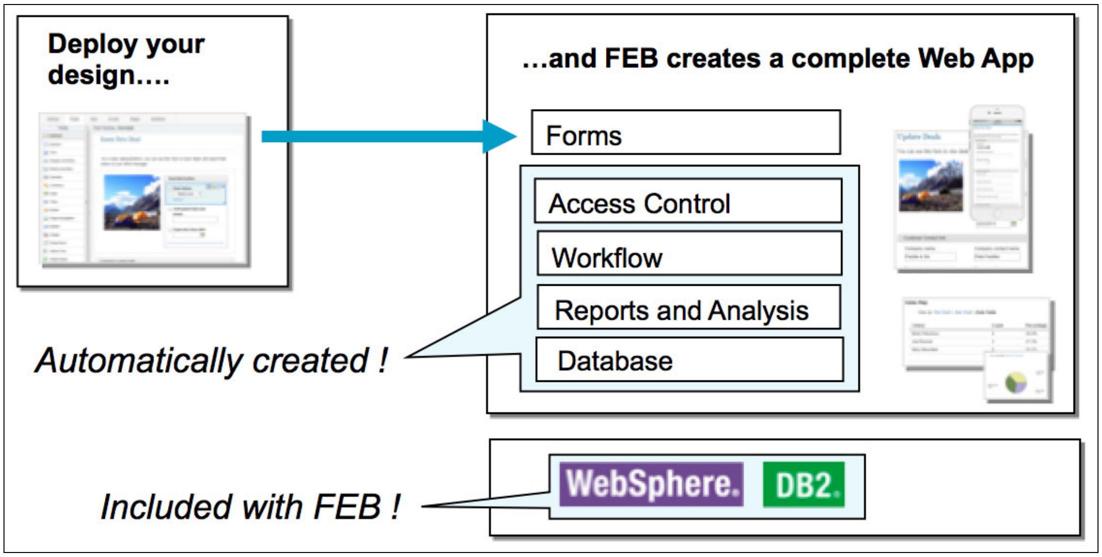


Figure 5-66 Forms Experience Builder overview

Forms Experience Builder can be deployed as a stand-alone WebSphere application, but can also run on the same instance of WebSphere Application Server as the portal. Forms Experience Builder needs a DB2 or Oracle database to store its applications and data and can use any LDAP directory and mail provider that is defined to WebSphere Application Server.

Figure 5-67 shows how Forms Experience Builder can be deployed as a stand-alone solution or as part of a WebSphere Portal deployment.

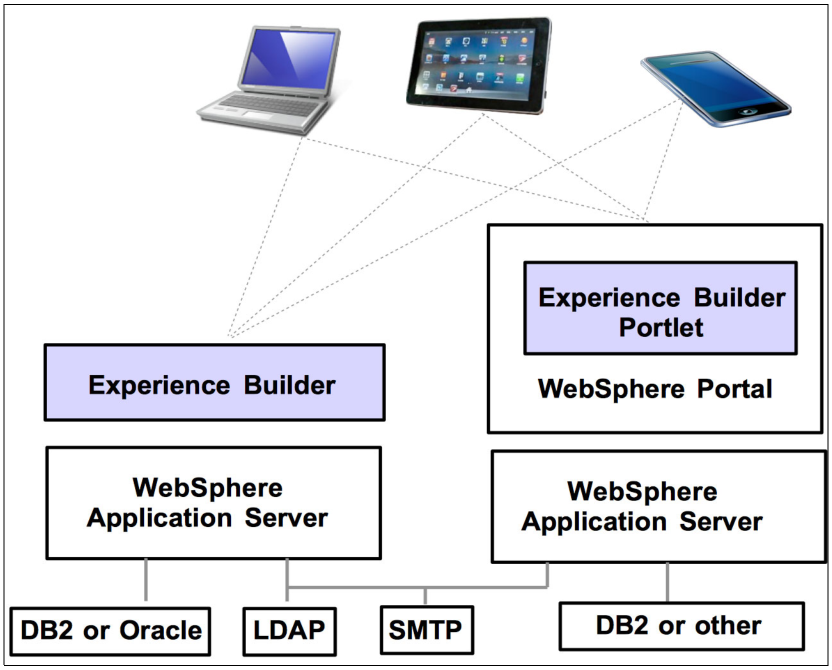

Experience Builder Portlet
WebSphere Portal
Experience Builder
WebSphere Application Server
WebSphere Application Server
DB2 or Oracle
LDAP
SMTP
DB2 or other

Figure 5-67 Forms Experience Builder deployment architecture

5.5.3 Use cases for Forms Experience Builder

An exceptional digital experience is all about anticipating and responding to customer needs, and web forms are an important contributor to a successful digital experience.

Web forms play an important role in the delivery of services and how to engage customers and employees. Web forms can be used for customers to enroll in programs, access services, and even provide feedback about products and services. An exceptional forms experience greatly improves the likelihood that customers engage in the programs that you offer. A bad or mediocre experience likely will be abandoned.

Well-designed web forms can help you get to know your customers, manage your business, improve operational efficiencies, and more. They enable end-to-end automation and help you deliver self-services to customers and employees. Forms can increase the value of your digital experience and offer new ways to reach out to your customers and employees.

The forms experience must be dynamic and engaging. It should anticipate and provide content that assists a user in achieving a goal. The experience should unfold dynamically as the user answers questions and makes choices. Ask questions only when they are relevant. Use profile information to personalize the experience. Prepopulate fields with known information to assist the user. The form itself should share a common look and context with the overall web experience you create.

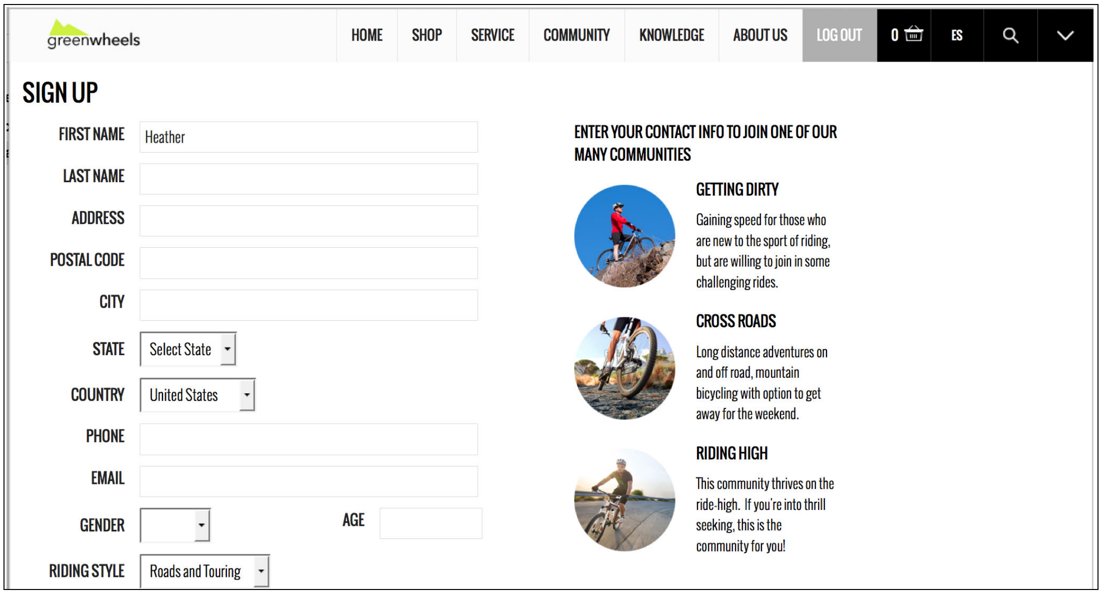
The example in 5.5.4, “Example: Creating a registration form for your website” on page 202 shows you how to build and integrate a dynamic community registration form into a website. The web form can be designed and integrated into portal by a non-technical user by using the prebuild integration between Forms Experience Builder and WebSphere Portal.

5.5.4 Example: Creating a registration form for your website

This section describes how to create a registration form and add it to an existing or a new page on a website.

The form itself is designed so that it fits nicely into the styling of the website, as shown in Figure 5-68.

To accomplish this goal, you can either build a combined stylesheet for your website and Forms Experience Builder, or keep a separate CSS in the Forms Experience Builder form just for commands and classes affecting your form design.



SIGN UP

FIRST NAME Heather

LAST NAME

ADDRESS

POSTAL CODE

CITY

STATE Select State

COUNTRY United States

PHONE

EMAIL

GENDER

AGE

RIDING STYLE Roads and Touring

ENTER YOUR CONTACT INFO TO JOIN ONE OF OUR MANY COMMUNITIES

GETTING DIRTY

Gaining speed for those who are new to the sport of riding, but are willing to join in some challenging rides.

CROSS ROADS

Long distance adventures on and off road, mountain bicycling with option to get away for the weekend.

RIDING HIGH

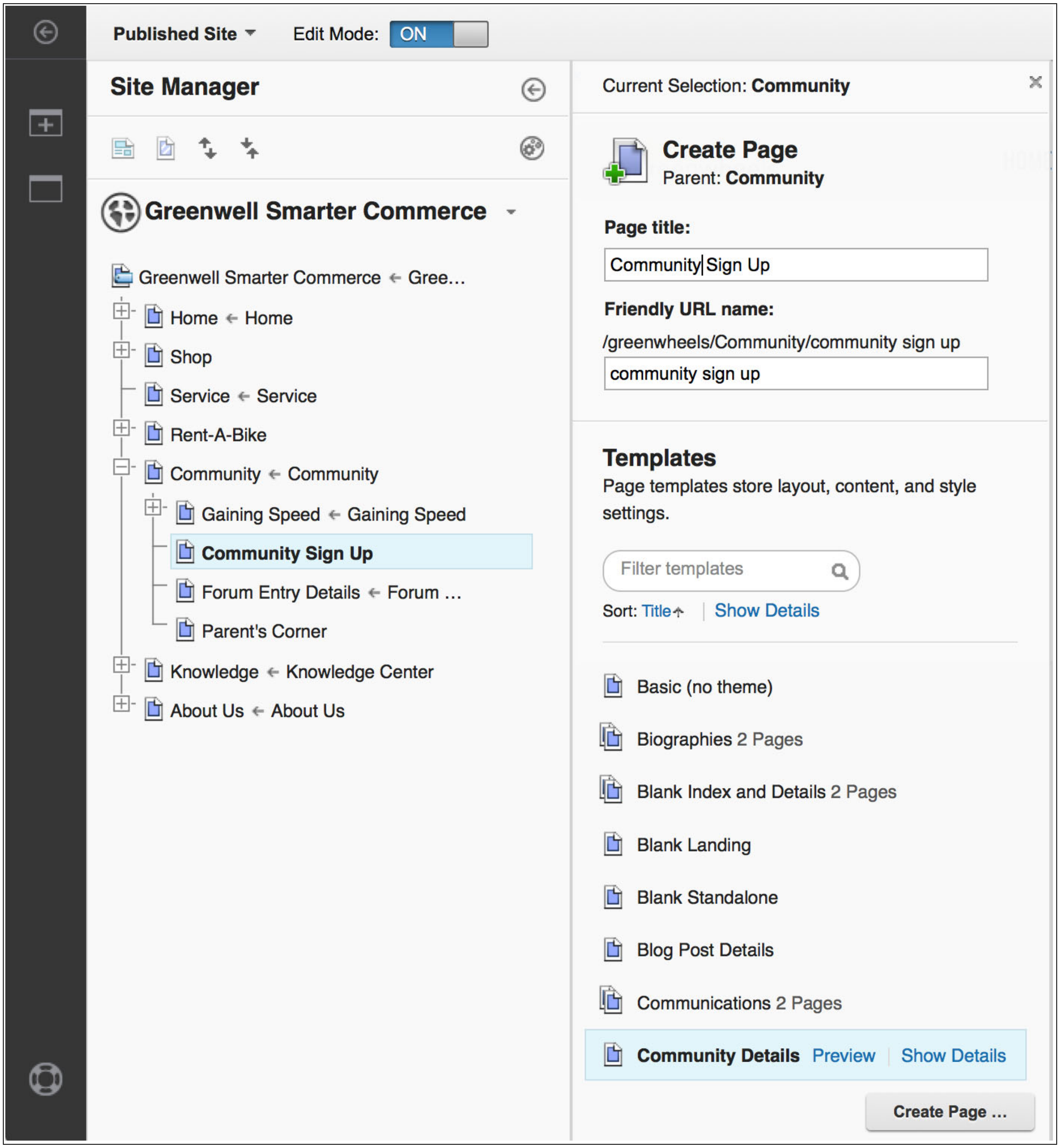
This community thrives on the ride-high. If you're into thrill seeking, this is the community for you!

Figure 5-68 Website sign-up form

To create the registration form, complete the following steps:

1. Create a portal page.

   You create a page in WebSphere Portal by expanding the Site Manager on the left and opening the menu while hovering your cursor over the current page name. Click **Create Page** from the menu and select an appropriate page template. Because the form can consume a considerable amount of space, select the **Basic** template and a single or two-column layout for the page, as shown in Figure 5-69.



Published Site

Edit Mode: ON

Current Selection: **Community**

Create Page

Parent: **Community**

Page title:

Community Sign Up

Friendly URL name:

/greenwheels/Community/community sign up

community sign up

Templates

Page templates store layout, content, and style settings.

Figure 5-69 Create a portal page for the Forms Experience Builder portlet

2. Add the Forms Experience Builder portlet to the page.

   Open the Page Components menu at the upper left of the toolbar, click the **Applications** tab, and find Forms Experience Portlet in the list of available applications, as shown in Figure 5-70.

   Then, either drag the portlet on to the page or simply hover the cursor over the portlet and click the **+** sign to add it to the page.

Tip: You can use the full text search to find quickly the portlet.

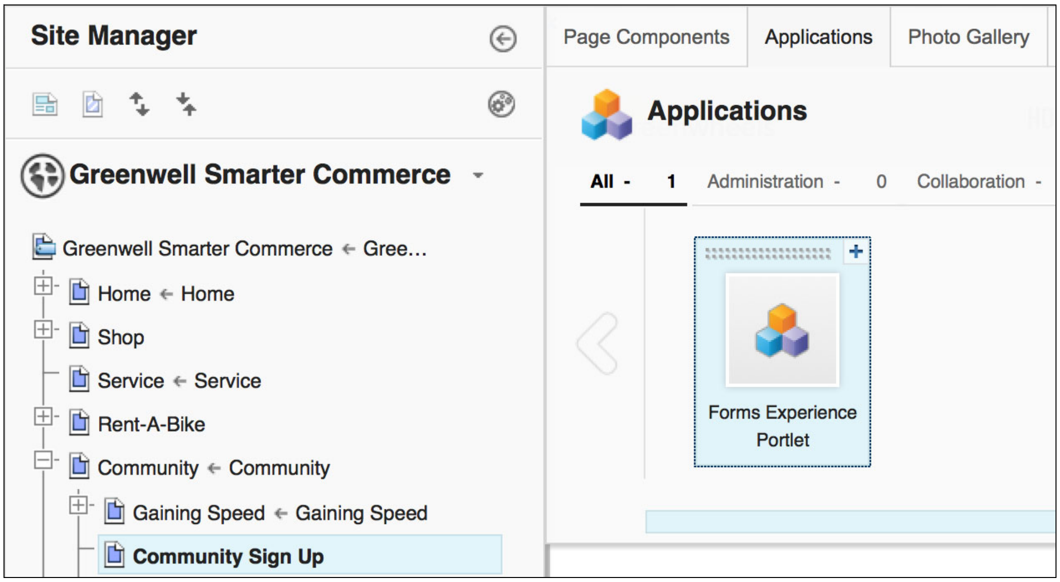


Figure 5-70 Add the Forms Experience portlet to a page

3. Configure the portlet to show the wanted form:

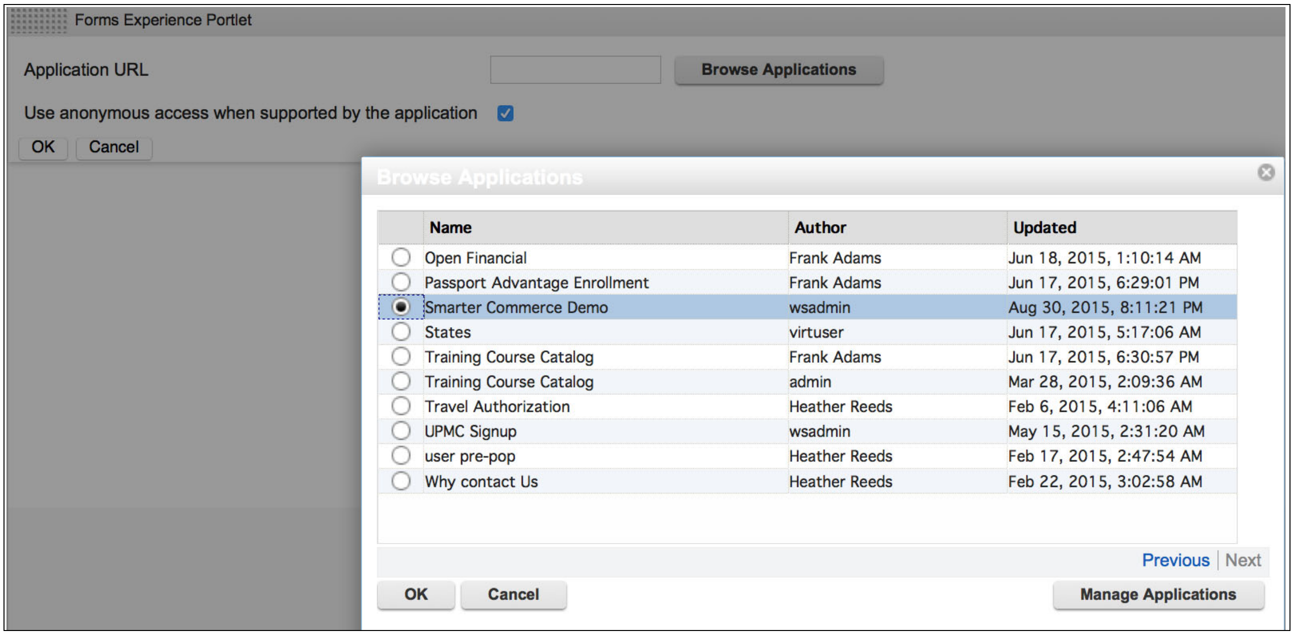
   a. Select **Edit Shared Setting** from the portlet menu.

   b. Click **Browse Applications** to select an existing form application from the list of available form applications on the Forms Experience Builder server, as shown in Figure 5-71.

   > **Note:** Alternatively, open Forms Designer and create an application by clicking **Manage Applications** (see Figure 5-71 on page 205).

   c. Click **OK** to close the portlet configuration

   d. Switch off the edit mode for the page to see the form rendered on the website, as shown in Figure 5-68 on page 202.



Application URL

Use anonymous access when supported by the application ☑

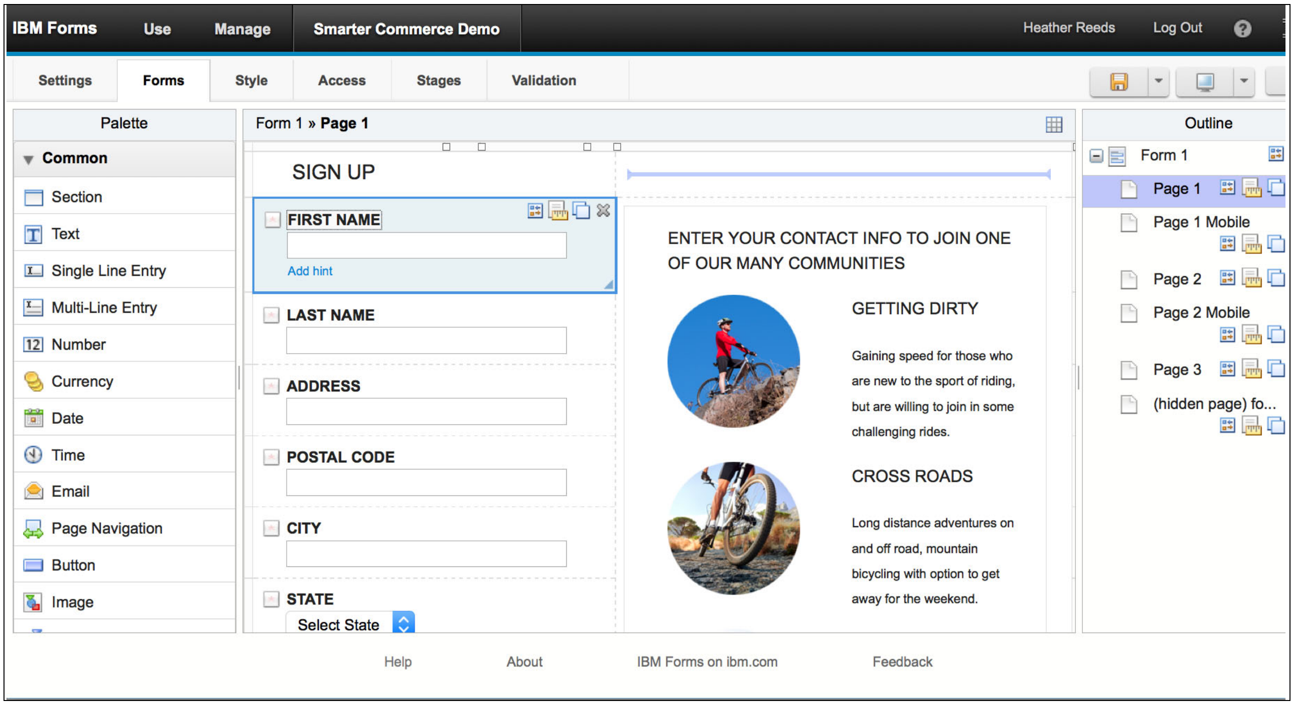
	Name	Author	Updated
○	Open Financial	Frank Adams	Jun 18, 2015, 1:10:14 AM
○	Passport Advantage Enrollment	Frank Adams	Jun 17, 2015, 6:29:01 PM
●	Smarter Commerce Demo	wsadmin	Aug 30, 2015, 8:11:21 PM
○	States	virtuser	Jun 17, 2015, 5:17:06 AM
○	Training Course Catalog	Frank Adams	Jun 17, 2015, 6:30:57 PM
○	Training Course Catalog	admin	Mar 28, 2015, 2:09:36 AM
○	Travel Authorization	Heather Reeds	Feb 6, 2015, 4:11:06 AM
○	UPMC Signup	wsadmin	May 15, 2015, 2:31:20 AM
○	user pre-pop	Heather Reeds	Feb 17, 2015, 2:47:54 AM
○	Why contact Us	Heather Reeds	Feb 22, 2015, 3:02:58 AM

Figure 5-71 Browse the existing Forms Experience Builder applications

4. Design a form application in the Forms Experience Builder design experience.

   When you click **Manage Applications**, as shown in Figure 5-71, the Forms Designer opens in a separate dialog box. Here, you can easily design a form application by using common design patterns, such as drag, property dialog boxes, and rule wizards.

You can start by dragging widgets from the palette on the left onto the design canvas and configuring their properties through inline editing or the property box that opens by clicking the context icons of a widget, as shown in Figure 5-72.



ENTER YOUR CONTACT INFO TO JOIN ONE OF OUR MANY COMMUNITIES

GETTING DIRTY

Gaining speed for those who are new to the sport of riding, but are willing to join in some challenging rides.

CROSS ROADS

Long distance adventures on and off road, mountain bicycling with option to get away for the weekend.

Figure 5-72 Forms Experience Builder design experience

For more information about the Forms Experience Builder and tutorials, see the following sources:

- IBM Forms Experience Builder IBM Knowledge Center:

  http://www.ibm.com/support/knowledgecenter/SS6KJL_8.6.1/Welcome/FEB_Welcome.html

- IBM Forms wiki:

  http://www.lotus.com/ldd/lfwiki.nsf

5.6 Unified Task List portlet

The Unified Task List portlet provides a single point of integration across multiple systems and displays tasks that WebSphere Portal users must complete to advance workflows. The Unified Task List portlet features a service provider layer that accesses, retrieves, and formats workflow events from a back-end system. The service provider layer is coupled with a presentation layer that renders the data in a visually appealing user interface. The separation of the service layer and the presentation layer lets you develop, test, and deploy your own customized Unified Task List portlet in a way that conforms with service-oriented architecture (SOA).

Here are the main components of the Unified Task List portlet:

- Task providers

  Task providers are services that access back-end systems to retrieve tasks. You can create multiple task providers to aggregate tasks from several back-end systems.

  For example, you can create a task provider to access, retrieve, and format tasks from a particular back-end system. You can then create another task provider to access, retrieve, and format tasks from a different back-end system. The result of these two task providers is that a single set of tasks is displayed in the user interface, but originate from two different back-end systems.

- Task provider instance

  Task provider instances are services that access back-end systems to retrieve tasks. Task provider instances are in the Task Provider Instance Registry (TPIR) and contain the parameters that you specify in task providers.

- Task Provider Instance Registry (TPIR)

  The TPIR contains task provider instance configurations. A task provider instance configuration contains a set of parameters that are required to connect to a back-end system and a unique ID to map the parameters to the appropriate task provider. The TPIR service is in WebSphere Application Server and stores the task provider configurations in an XML variable. The TPIR service also provides a service to get and modify task provider instances.

- Task dispatcher

  The task dispatcher acts as a link between the Unified Task List portlet and the task providers. When an action occurs in the portlet, the task dispatcher retrieves task provider instance configurations from the TPIR service and calls a **`getTaskList`** service operation on each task provider instance configuration.

  You can also configure the Unified Task List portlet to use a cached task dispatcher if you do not want to access the back-end systems each time an action occurs in the user interface. The cached task dispatcher uses dynamic caching in WebSphere Application Server to store and retrieve tasks.

- Task handler

  Task handlers define the behavior of the Unified Task List portlet when users select a task to advance a workflow. The task handlers determine how the Unified Task List portlet connects to the tasks that the users must complete.

  You can configure task handlers to connect the Unified Task List portlet to tasks in the following ways:

  - Use the WebSphere Portal property broker to open a portlet where users can complete the task. The portlet that opens can be on the same portal page as the Unified Task List portlet or a different portal page.
  - Use dynamic portal pages, or task pages, to open task processing and task supporting portlets.
  - Provide an external URL that starts a web application.

  You can also customize the Unified Task List portlet in Web Experience Factory if you want to provide functions for completing tasks. For example, if there is a task to approve or deny a request in a human resources workflow, you can use WIBM Web Experience Factory builders to provide a simple menu with approve and deny actions.

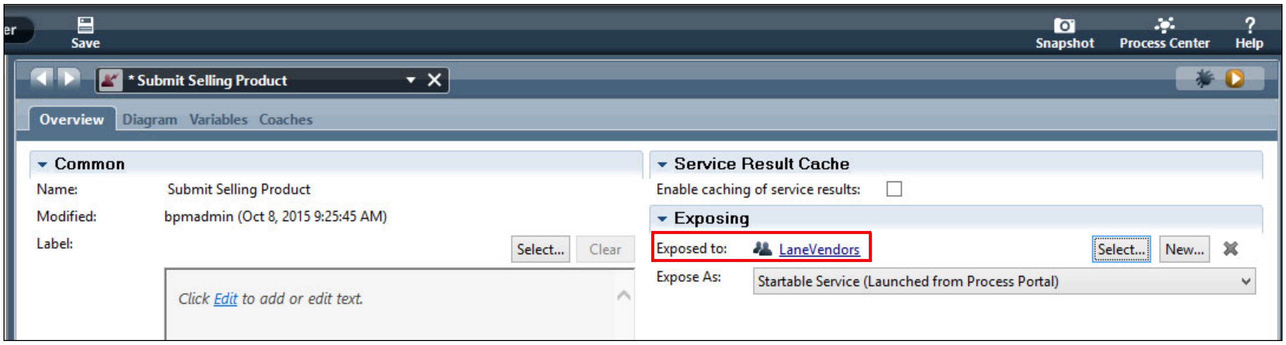
Customizing the Unified Task List portlet

You can customize the Unified Task List portlet at run time as a WebSphere Portal administrator. Customizing at run time involves accessing the deployed Unified Task List portlet to configure the portlet settings.

5.6.1 Integrating with IBM BPM

This section highlights the IBM BPM configuration that is required to integrate with the Unified Task List portlet. To make the three process steps in the process example accessible from the Unified Task List portlet task handling, the coach of each step must be exposed. To do so, access the Overview tab of the coach of the step that you want to expose. In this example, the coach for each of the three steps is exposed.

Figure 5-73 shows how the Submit Selling Product coach is exposed to Lane Vendors.



Name: Submit Selling Product

Modified: bpmadmin (Oct 8, 2015 9:25:45 AM)

Label:

Enable caching of service results: □

Exposed to: LaneVendors

Expose As: Startable Service (Launched from Process Portal)

Figure 5-73 Expose the Submit Selling Product coach

To configure the Unified Task List portlet and an IBM BPM Coach portlet to surface both the task information and the coach from IBM BPM, complete the following steps:

1. Create a WebSphere Portal page with a simple one- or two-column layout.
2. Add both the Unified Task List portlet and the Coach portlet to the page.
3. Select **Configure** from the portlet menu to configure the Unified Task List portlet.

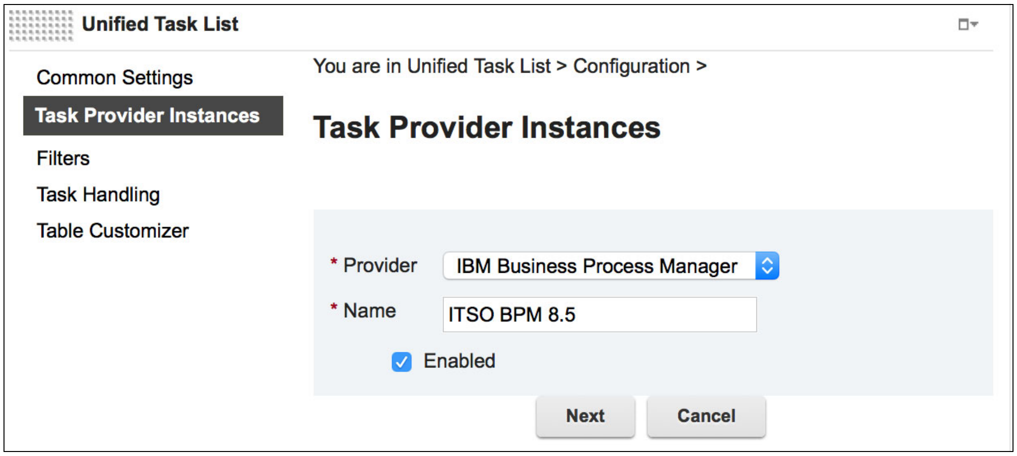
4. Create a Task Provider (see Figure 5-74):

   a. Click **Add** in the Task Provider Instances section.

   b. Select **IBM Business Process Manager** as the Provider.

   c. Enter a name for the Task Provider.

   d. Click **Next**.



* Provider: IBM Business Process Manager
* Name: ITSO BPM 8.5
☑ Enabled

Figure 5-74 Configure the IBM BPM task provider

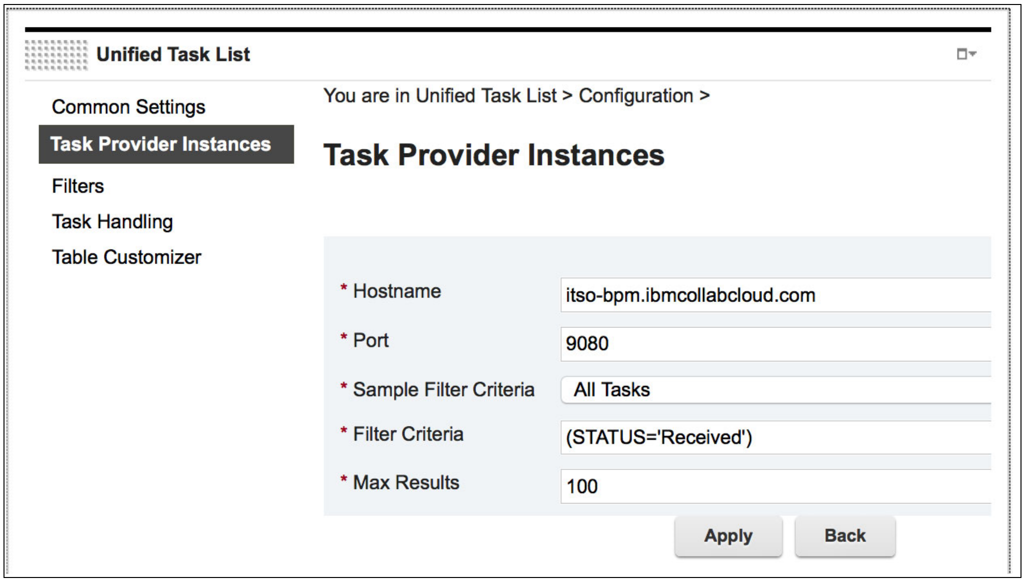
   e. Enter the host name and port for the IBM BPM Server.

   f. Select the Sample Filter Criteria of **All Tasks**. You might also display claimed or ready tasks depending on your use case.

   g. Leave the Filter Criteria *as is* or enter one of the following values:

      - (STATE=8): Tasks with a claimed state.
      - (STATE=2): Tasks in a release state.
      - (STATUS='Received'): Tasks that are available to the current user.

h. Update the value in the Max Results field if you want to change the number of tasks that can be returned from IBM BPM, as shown in Figure 5-75.



Task Provider Instances

* Hostname	itso-bpm.ibmcollabcloud.com
* Port	9080
* Sample Filter Criteria	All Tasks
* Filter Criteria	(STATUS='Received')
* Max Results	100

Figure 5-75 Configure the IBM BPM task provider (continued)

i. Click **Apply**.

j. Click **Apply** again in the list of Task Provider Instances.

5. Configure a Task Handler for each task type in your business process:

   a. Click **Add** in the Task Handling section.

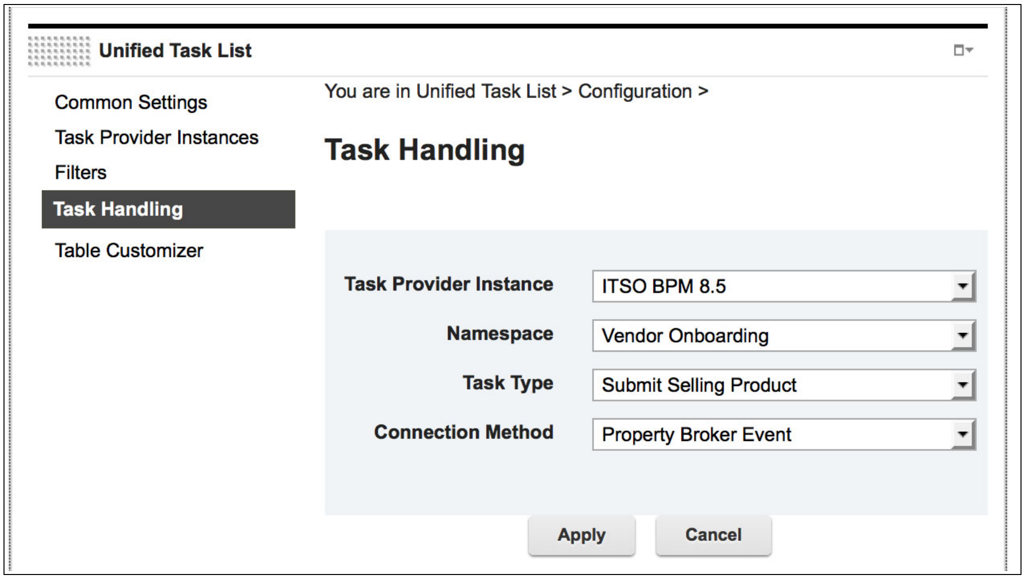
   b. Select the Task Provider instance that you created in step 4 on page 214.

   c. Select your business process in the Namespace drop-down list.

   d. Select the task of your business process in the Task Type drop-down list.

e. Select the proper Connection Method for the task handler. For this example, select **Property Broker Event** to have the Coach portlet on the same page, as shown in Figure 5-76. Here are the Connection Method options:

   - Dynamic Portal Page: If you select **Dynamic Portal Page**, you must specify the unique ID of the portal page in the Unique ID of Portal Page field.
   - External URL: If you select **External URL**, you must specify the URL in the External URL Pattern field.
   - Property Broker Event: No other parameters are required if **Property Broker Event** is selected.



Task Handling

Task Provider Instance: ITSO BPM 8.5
Namespace: Vendor Onboarding
Task Type: Submit Selling Product
Connection Method: Property Broker Event

Figure 5-76 Configure the IBM BPM Task Handler

f. Click **Apply**.

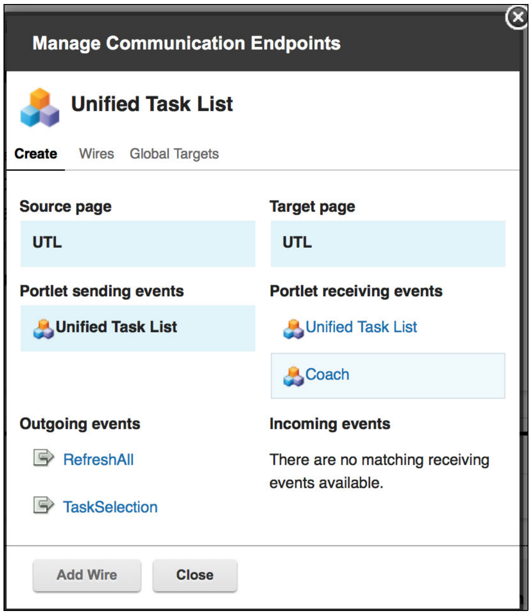
g. Click **Apply** again in the list of Task Provider Instances.

6. Optionally, adjust the appearance of the Unified Task List portlet in the Common Settings and the contents of the task list in the Table Customizer section.
7. Click **OK** to return to the render mode of the Unified Task List portlet.

8. Configure **Property Broker Events** to wire the two portlets on the page together:

   a. Select **Manage Endpoints** from the Unified Task List portlet menu.

   b. Click **Coach** in the Portlet receiving events section, as shown in Figure 5-77.



Manage Communication Endpoints

Unified Task List

Source page: UTL

Target page: UTL

Portlet sending events: Unified Task List

Portlet receiving events: Unified Task List, Coach

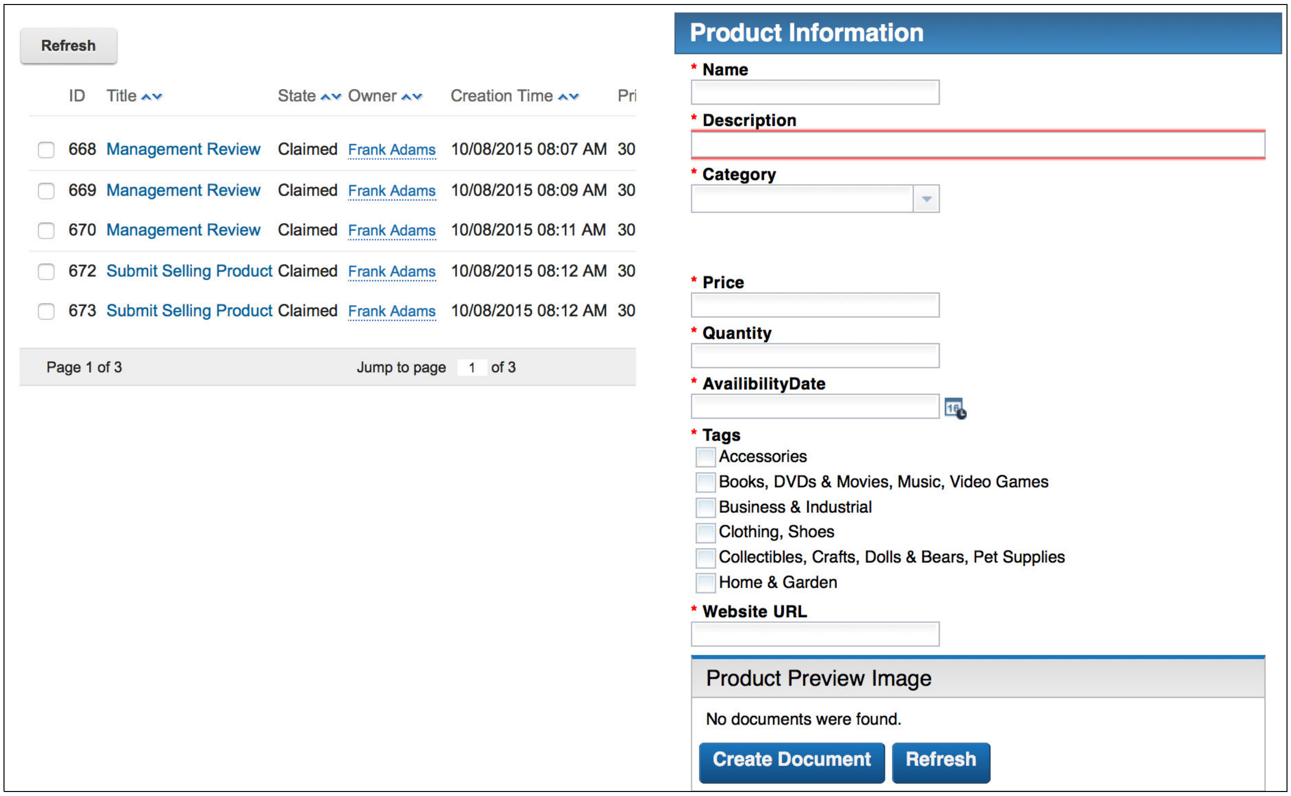
Outgoing events: RefreshAll, TaskSelection

Incoming events: There are no matching receiving events available.

Figure 5-77 Manage Communication Endpoints

   c. Click **Add Wire** and then click **Close** to return to the WebSphere Portal page.

You should now see the list of tasks from the IBM BPM process rendered in the Unified Task List portlet. You can claim received tasks, and when you click the link text on a claimed task, the corresponding coach is loaded in the coach portlet, as shown in Figure 5-78.



Refresh

	ID	Title	State	Owner	Creation Time	Pri
□	668	Management Review	Claimed	Frank Adams	10/08/2015 08:07 AM	30
□	669	Management Review	Claimed	Frank Adams	10/08/2015 08:09 AM	30
□	670	Management Review	Claimed	Frank Adams	10/08/2015 08:11 AM	30
□	672	Submit Selling Product	Claimed	Frank Adams	10/08/2015 08:12 AM	30
□	673	Submit Selling Product	Claimed	Frank Adams	10/08/2015 08:12 AM	30

Page 1 of 3 Jump to page 1 of 3

Product Information

* Name

* Description

* Category

* Price

* Quantity

* AvailibilityDate

* Tags

□ Accessories
□ Books, DVDs & Movies, Music, Video Games
□ Business & Industrial
□ Clothing, Shoes
□ Collectibles, Crafts, Dolls & Bears, Pet Supplies
□ Home & Garden

* Website URL

Product Preview Image

No documents were found.

Create Document Refresh

Figure 5-78 WebSphere Portal page showing a Unified Task List portlet in combination with an IBM BPM Coach

5.6.2 Integrating with Forms Experience Builder

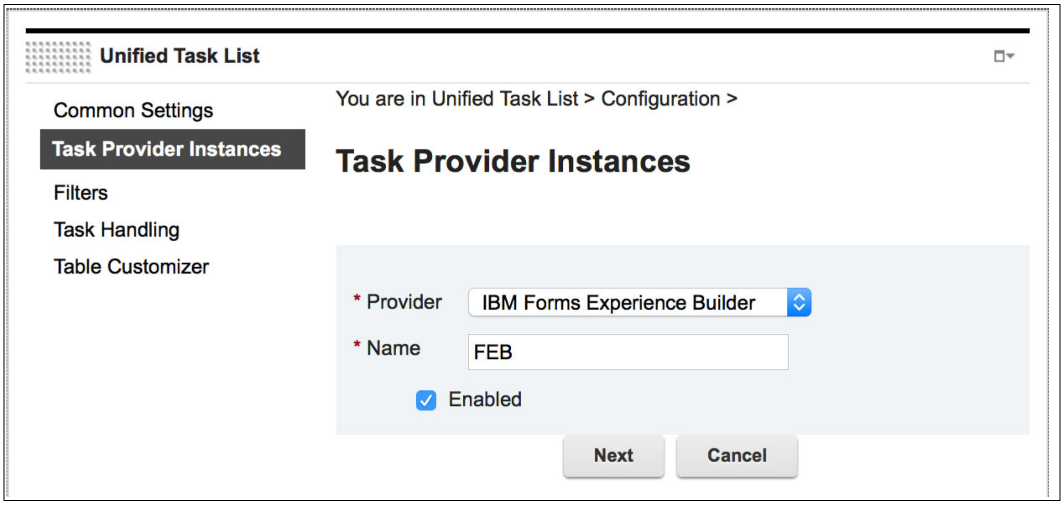
Another use case for the Unified Task List portlet is to display a list of available tasks from Forms Experience Builder for the logged in user. This option is similar to using DDC, as described in 5.2.6, “Use case 3: Integrating with IBM Forms Experience Builder” on page 163, where you have a lot less control over the list rendering than using DDC, but have more standard features, such as task claiming and list presentation filters that are available. Also, there are no technical steps that are involved, such as HTML markup editing, but instead there is a simple series of configuration steps.

This section describes the steps that are required to use the Unified Task List portlet to render the Forms Experience Builder task data in the bike rentals demonstration scenario that is used for the examples in this chapter. The bike rental form introduced in 5.2.6, “Use case 3: Integrating with IBM Forms Experience Builder” on page 163 is used in this example.

The following steps describe how to configure the Unified Task List portlet and the Forms Experience Builder portlet to surface both the task information and the form from Forms Experience Builder:

1. Create a WebSphere Portal page with a simple one- or two-column layout.
2. Add both the Unified Task List portlet and the Forms Experience Builder portlet to the page.
3. Configure the Unified Task List portlet. Select **Configure** from the portlet menu.

4. Create a task provider instance (see Figure 5-79):
   a. Click **Add** in the Task Provider Instances section.
   b. Select **IBM Forms Experience Builder** as the Provider.
   c. Enter a name for the Task Provider.
   d. Click **Next**.



Unified Task List

You are in Unified Task List > Configuration >

Task Provider Instances

* Provider: IBM Forms Experience Builder

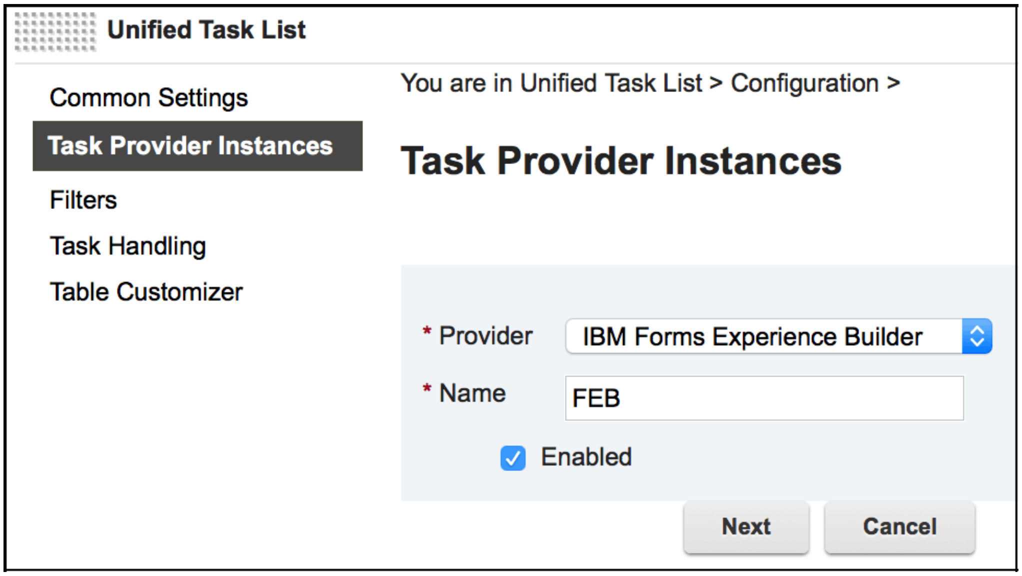
* Name: FEB

☑ Enabled

Figure 5-79 Configure a task provider

e. Complete the Task Provider Instances window, as shown in Figure 5-80:

- Enter the URL path to the Forms Experience Builder server in the Server Endpoint field.
- Select the application from the Forms Experience Builder server from the Application drop-down list.
- Select the form within the Forms Experience Builder application from the Form drop-down list.
- Select the workflow stage of the form submission to display from the Stage drop-down list.



* Provider: IBM Forms Experience Builder

* Name: FEB

☑ Enabled

Figure 5-80 Configure a task provider (continued)

f. Click **Apply** and then click **Apply** again in the list of Task Provider Instances.

5. Configure a Task Handler for each task type in your business process:

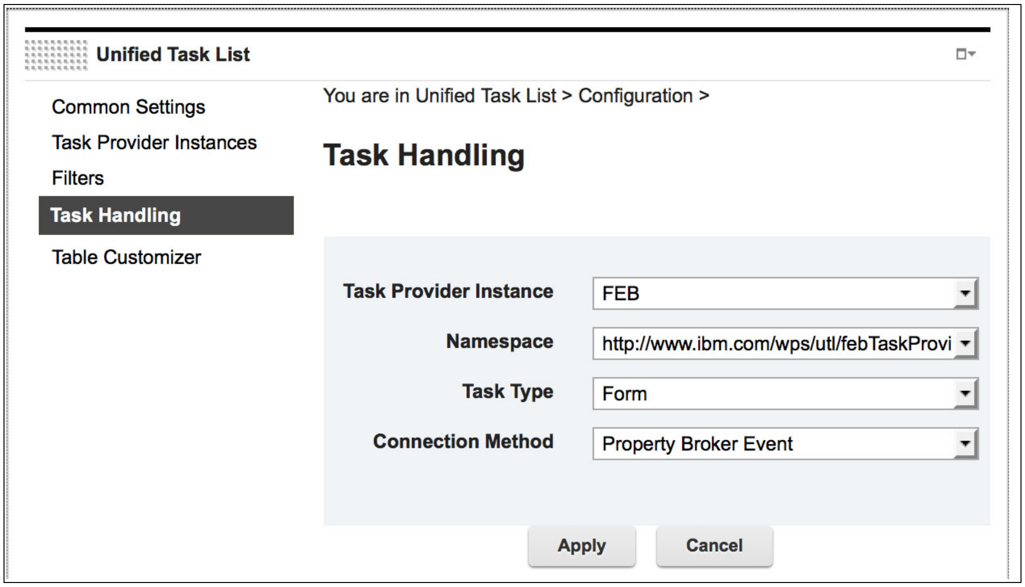
a. Click **Add** in the Task Handling section.

a. Select the task provider instance that you created in step 4 on page 214.

b. Select your business process in the Namespace field and the default selection for the Task Type.

c. Select the correct connection method for the task handler. In this example, select **Property Broker Event** to have the Coach portlet on the same page, as shown in Figure 5-81. Here are the Connection Method options:

- Dynamic Portal Page: If you select **Dynamic Portal Page**, you must specify the unique ID of the portal page in the Unique ID of Portal Page field.
- External URL: If you select **External URL**, you must specify the URL in the External URL Pattern field.
- Property Broker Event: No other parameters are required if **Property Broker Event** is selected.



Task Provider Instance: FEB
Namespace: http://www.ibm.com/wps/utl/febTaskProvi
Task Type: Form
Connection Method: Property Broker Event

Figure 5-81 Configuring the Forms Experience Builder task handler

d. Click **Apply** and then click **Apply** again in the list of Task Provider Instances.

6. Optionally, adjust the appearance of the Unified Task List portlet in the Common Settings and the contents of the task list in the Table Customizer section.
7. Click **OK** to return to the render mode of the Unified Task List portlet.
8. Configure Property Broker Events to wire the two portlets on the page together:

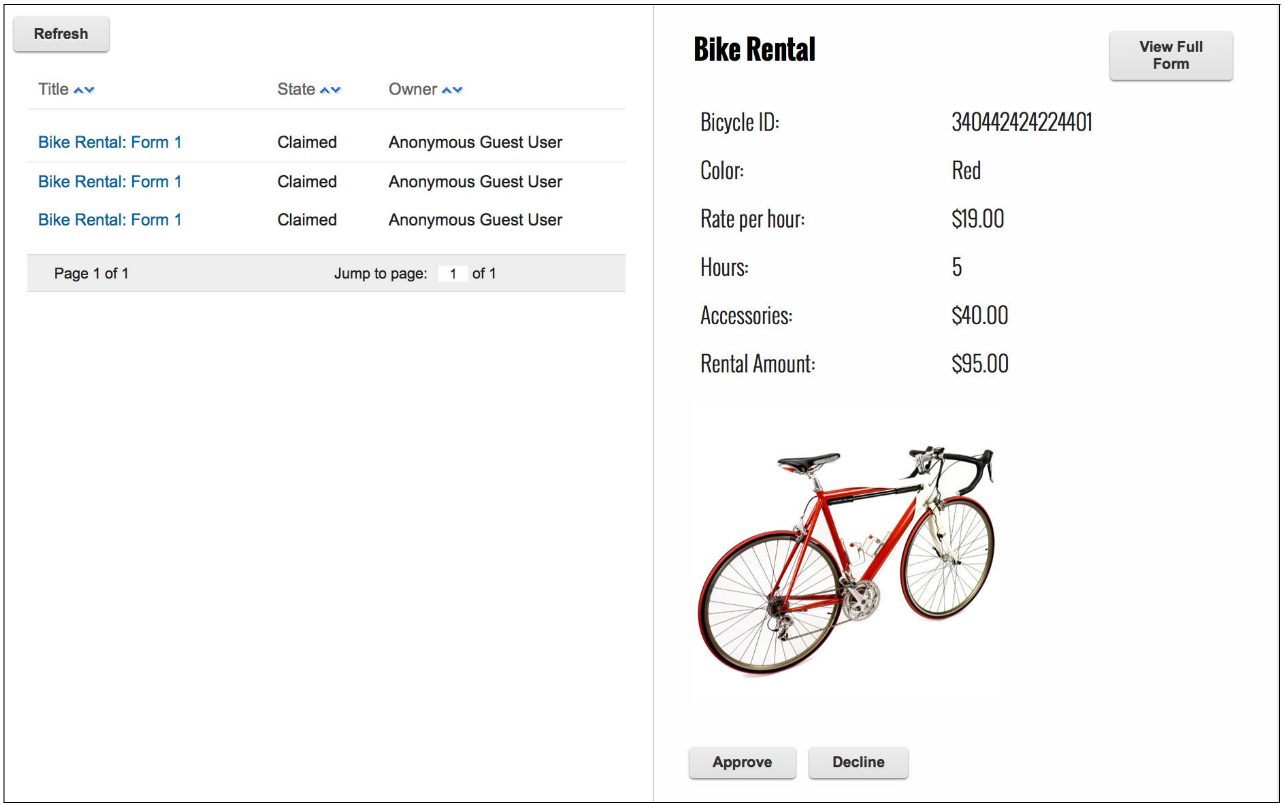
   a. Select **Manage Endpoints** from the Unified Task List portlet menu.

   b. Click **Forms Experience Builder** in the Portlet receiving events section, as shown in Figure 5-77 on page 212.

   c. Click **Add Wire**.

   d. Click **Close** to return to your WebSphere Portal page.

You should now see the list of tasks from the Forms Experience Builder application rendered in the Unified Task List portlet. You can claim received tasks, and when you click the link text on a claimed task, the respective form is loaded in the Forms Experience Builder portlet, as shown in Figure 5-82.



Title	State	Owner
Bike Rental: Form 1	Claimed	Anonymous Guest User
Bike Rental: Form 1	Claimed	Anonymous Guest User
Bike Rental: Form 1	Claimed	Anonymous Guest User

Page 1 of 1 Jump to page: 1 of 1

Bike Rental

Bicycle ID:	340442424224401
Color:	Red
Rate per hour:	$19.00
Hours:	5
Accessories:	$40.00
Rental Amount:	$95.00

Figure 5-82 WebSphere Portal page showing Unified Task List portlet in combination with a Forms Experience Builder form

A

Installing and connecting with IBM Digital Experience File Sync

Digital Experience File Sync (DXSync) synchronizes any WebDAV-based WebSphere Portal theme with your local workstation. It replaces your existing WebDAV client and watches file system changes in the background. It pulls the theme files to your workstation and uploads any changes you make.

This appendix provides basic information about how to install DXSync on your local workstation and to connect to your WebSphere Portal server to synchronize the themes directory.

To learn more about DXSync, refer to the DXSync wiki:

`https://github.com/digexp/dxsync/wiki`

> **Note:** IBM Digital Experience File Sync runs on Windows, OSX, and Linux, and can be used with WebSphere Portal V8.5, but it is not formally supported currently.

This appendix covers the following topics:

- “Installing DXSync” on page 220
- “Initializing DXSync” on page 220

Installing DXSync

To install DXSync on your workstation, complete the following steps:

1. Install the `node.js` source code for your operating system from the `nodejs.org` website:
   - For Windows or Macintosh, go to the following website:

     https://nodejs.org/download/docs/v0.12.7/download/
   - For Linux, find the appropriate installation script on the GitHub website:

     https://github.com/creationix/nvm
2. Download IBM Digital Experience File Sync from the following website:

   https://github.com/digexp/dxsync/tree/master/release
3. Extract the compressed file.
4. Change to the extracted directory and run **`install.cmd`** if you are using Windows or **`install.sh`** for Linux or OSX.

Initializing DXSync

You must initialize IBM Digital Experience File Sync to synchronize with a directory by completing the following steps:

1. Create a local directory, for example, `/ITSO simple theme`.
2. In the command line, change to your new directory, for example:

   ```
   C:\>cd itso simple theme
   ```
3. Run the following command to initialize a theme with your new directory:

   ```
   C:\ITSOPortal 8.5theme>dxsync init
   ```
4. Respond to the following prompts to connect to your server, as shown in Example A-1. Each prompt automatically fills with a default value. You can press the Enter key to use the default values.

Example A-1 Responses to DXSync initialization prompts

```
Hostname:  itso-wp2.ibmcollabcloud.com
Username:  (wpsadmin) virtuser
Password:
Path to the content handler servlet:  (/wps/mycontenthandler)
Secure Connection (https)?:  (true) false
Port:  (10039)
```

5. DXSync connects to the WebSphere Portal Server and retrieves the themes found, as shown in Example A-2. Enter the number that corresponds to the theme that you want to synchronize.

Example A-2 DXSync - connecting to the WebSphere Portal Server and retrieving themes

```
Connecting to the remote server...
Fetching themes from
http://itso-wp2.ibmcollabcloud.com:10039/wps/mycontenthandler/dav/fs-type1/them
es/
The following themes were found. Please select which one you would like to
connect to:
1. ITSO Portal 8.5 theme
2. ITSO simple theme
3. Portal8.5
4. Simple
5. ThemeDevSite
6. Toolbar8.5
7. angularBootstrap
8. greenwell-bank
9. greenwell-responsive-V1_2
10. greenwellsmartercommerce
11. ibmOGSThemeV1
12. stratusprompt: Enter a number between 1 and 11:  2
```

6. DXSync returns the following messages:

```
Successfully setup theme synchronization.
```

```
You are ready to execute the synchronization with the 'run' command.
```

7. Run the following command:

```
C:\ITSO simple theme>dxsync run
```

> **Note:** The `dxsync run` command initiates a two-way synchronization between your workstation and your server. IBM Digital Experience File Sync watches your files in the background and restarts a full synchronization at intervals of time that you configure.

8. File synchronization takes place, as shown in Example A-3.

Example A-3 DXSync theme synchronization

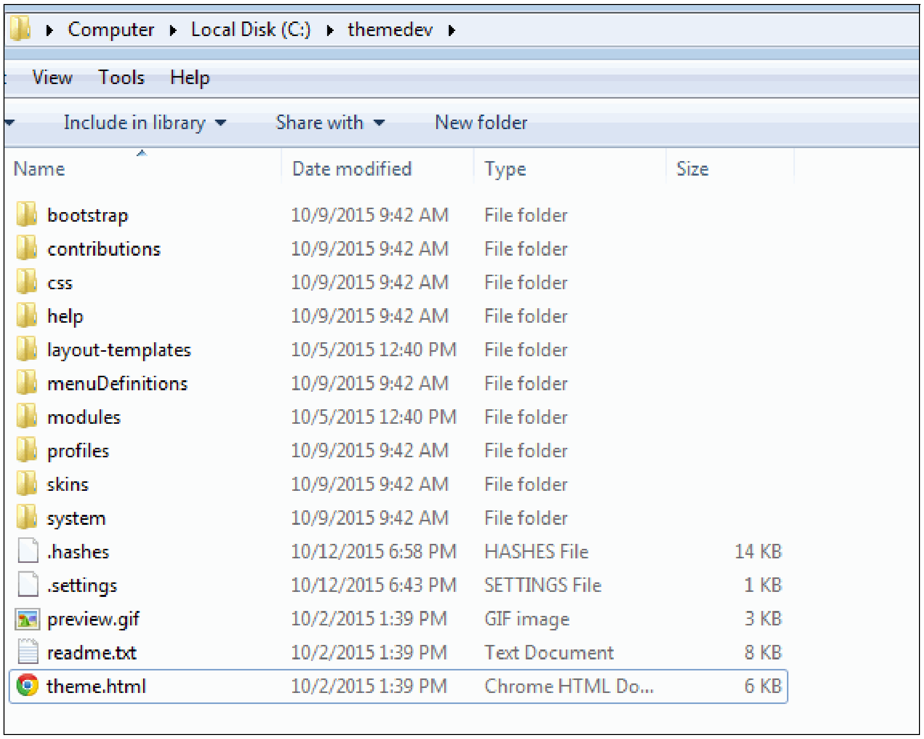
```
IBM Digital Experience File Sync v1.0.1
***************************************
Configuration:
  URL: http://itso-wp2.ibmcollabcloud.com:10039/wps/mycontenthandler/dav/fs-type1/themes/ITSO
simple theme
  username: virtuser
  password: ********
  secure: false
  syncIntervalMin: 5
  threads: 10
  filterRegex: ^\.|~$|.bak$|.resolve$|.delete$|.conflict$

Started 2-way synchronization... Phase 1...
File download complete: readme.txt
File download complete: preview.gif
File download complete: theme.html
File download complete: bootstrap/readme.txt
.... truncated .....
File download complete: modules/st_svg/config/retrieveSprite.js.uncompressed.js.
preCBT.js
Phase 2...
Synchronization complete.

Synchronization Report:
Total      : 164
Uploaded   : 0
Downloaded : 164
Deleted    : 0 / 0 (Local/Remote)
Conflicts  : 0 / 0 (Resolved)
No Action  : 0
Errors     : 0
Due to platform limitations on Windows (Open file handle handling) deleting of directories is
not supported while watching files. In order to delete a folder while watching use another
operating system such as OSX or Linux. On Windows, please stop watching, delete the folder and
then restart the process.

Watching files in 'C:/ITSO simple theme' for changes...
Idle...
Starting next synchronization in 5 minutes...
```

9. The theme on your workstation is now connected to your WebSphere Portal Server and you are ready to synchronize your theme files. Figure A-1 shows the Simple theme files on the workstation. You can edit your files locally and they are synchronized periodically with the WebSphere Portal Server.



Name	Date modified	Type	Size
bootstrap	10/9/2015 9:42 AM	File folder	
contributions	10/9/2015 9:42 AM	File folder	
css	10/9/2015 9:42 AM	File folder	
help	10/9/2015 9:42 AM	File folder	
layout-templates	10/5/2015 12:40 PM	File folder	
menuDefinitions	10/9/2015 9:42 AM	File folder	
modules	10/5/2015 12:40 PM	File folder	
profiles	10/9/2015 9:42 AM	File folder	
skins	10/9/2015 9:42 AM	File folder	
system	10/9/2015 9:42 AM	File folder	
.hashes	10/12/2015 6:58 PM	HASHES File	14 KB
.settings	10/12/2015 6:43 PM	SETTINGS File	1 KB
preview.gif	10/2/2015 1:39 PM	GIF image	3 KB
readme.txt	10/2/2015 1:39 PM	Text Document	8 KB
theme.html	10/2/2015 1:39 PM	Chrome HTML Do...	6 KB

Figure A-1 Simple theme files on the workstation

B

Creating a custom theme

This appendix explains how you can build a custom theme that is based on the Simple theme. As described in 3.1.2, “Theme resources” on page 36, an IBM WebSphere Portal theme is a collection of static and dynamic resources, such as JSPs, HTML, JavaScript, and stylesheet files, which are used together to provide the appearance for the website. This appearance includes a color scheme, element positioning on a page, and menu structure.

> **Tip:** It is considered a preferred practice to build a theme based on the default theme template provided with WebSphere Portal. The themes that are provided with WebSphere Portal are well-tuned and can potentially reduce high maintenance cost. For more information about the default them templates, see 3.1.1, “WebSphere Portal V8.5 themes” on page 35.

This appendix covers the following topics:

- “Copying a theme to customize” on page 226
- “Styles” on page 233
- “Navigation” on page 235
- “XMLAccess” on page 241

For more information about working with themes, see the following websites:

- *Developing Themes for WebSphere Portal 8.5* article:

  `http://www.lotus.com/ldd/portalwiki.nsf/xpViewCategories.xsp?lookupName=Developing%20Themes%20for%20WebSphere%20Portal%208.5`

- The *Manually create a copy of your theme* topic in the IBM WebSphere Portal V8.5.0 IBM Knowledge Center:

  `http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/dev-theme/themeopt_cust_copytheme.dita`

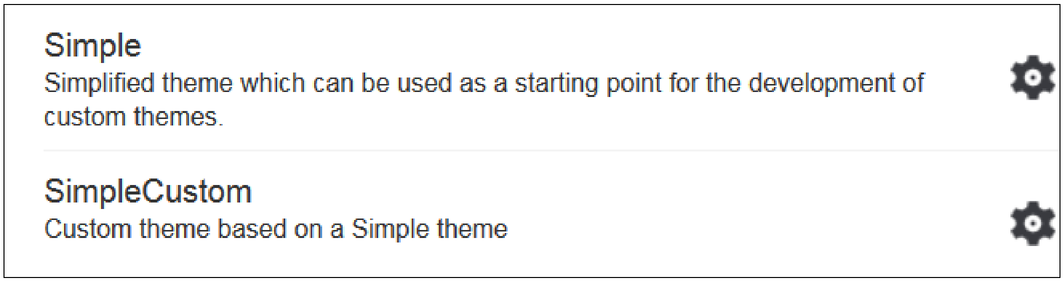
Copying a theme to customize

To create a custom theme based on the Simple theme, complete the following steps.

> **Note:** As a prerequisite to running the steps in this example, you need IBM Digital Experience File Sync (DXSync) and `node.js`. Download and install `node.js` for the operating system you are using. For more information about using DXSync, see 3.4, “Managing themes by using IBM Digital Experience File Sync” on page 48.

1. From the WebSphere Portal administration console, click **Applications menu** → **Theme Development** to start the Theme Manager. Copy the Simple theme to a new theme called SimpleCustom. See Figure B-1.

   For more information about using the Theme Manager, see 3.2, “Creating a theme with the Theme Manager” on page 40.

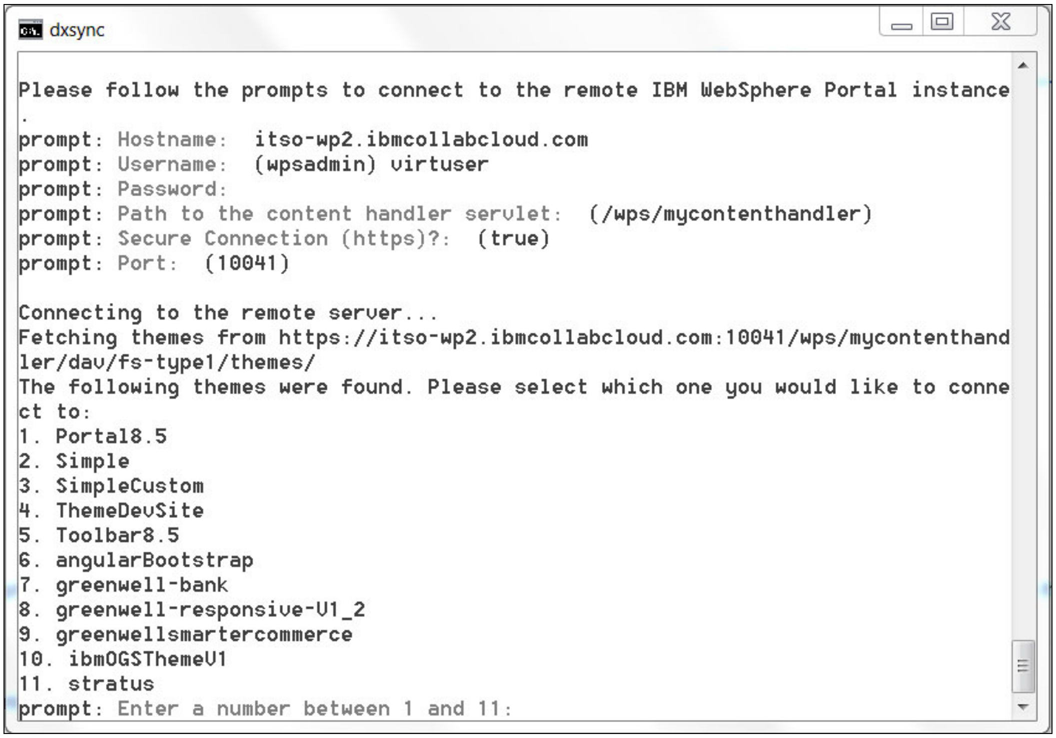


Simple
Simplified theme which can be used as a starting point for the development of custom themes.

SimpleCustom
Custom theme based on a Simple theme

Figure B-1 Copied theme called SimpleCustom in Theme Manager

2. Run the DXSync file sync program to copy the SimpleCustom theme files to your local folder (Figure B-2).

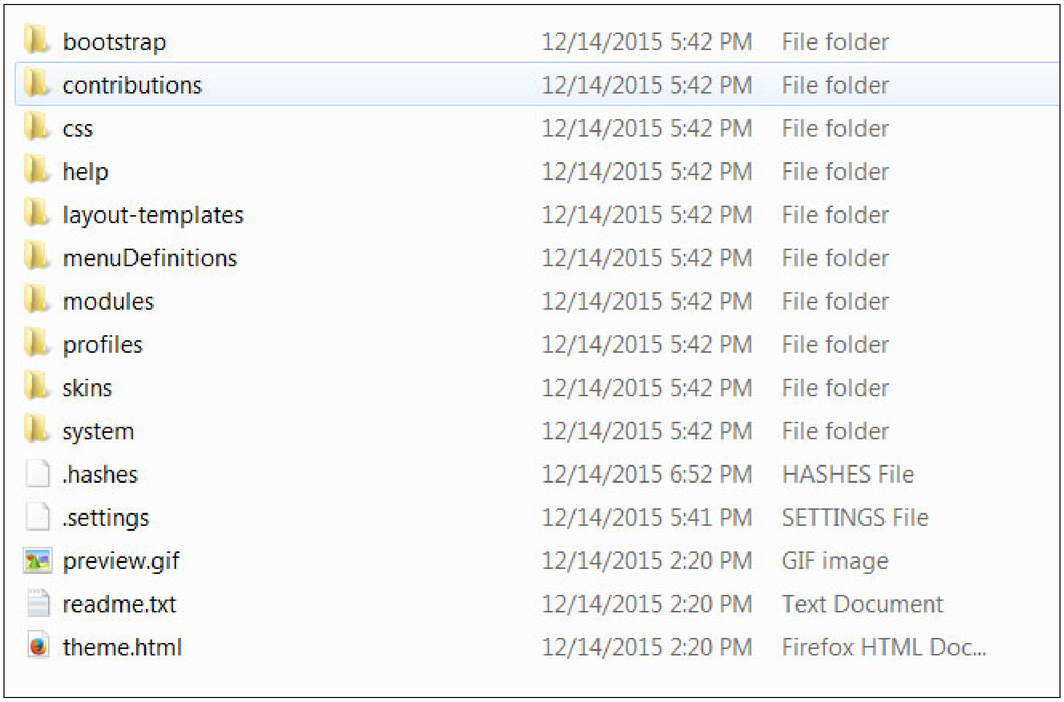


```
Please follow the prompts to connect to the remote IBM WebSphere Portal instance
.
prompt: Hostname:  itso-wp2.ibmcollabcloud.com
prompt: Username:  (wpsadmin) virtuser
prompt: Password:
prompt: Path to the content handler servlet:  (/wps/mycontenthandler)
prompt: Secure Connection (https)?:  (true)
prompt: Port:  (10041)

Connecting to the remote server...
Fetching themes from https://itso-wp2.ibmcollabcloud.com:10041/wps/mycontenthand
ler/dav/fs-type1/themes/
The following themes were found. Please select which one you would like to conne
ct to:
1. Portal8.5
2. Simple
3. SimpleCustom
4. ThemeDevSite
5. Toolbar8.5
6. angularBootstrap
7. greenwell-bank
8. greenwell-responsive-V1_2
9. greenwellsmartercommerce
10. ibmOGSThemeV1
11. stratus
prompt: Enter a number between 1 and 11:
```

Figure B-2 DXSync file synchronization utility

For example, if you create a folder that is called `themedev` and then run DXSync from that folder, you see the theme files copied, as shown in Figure B-3.

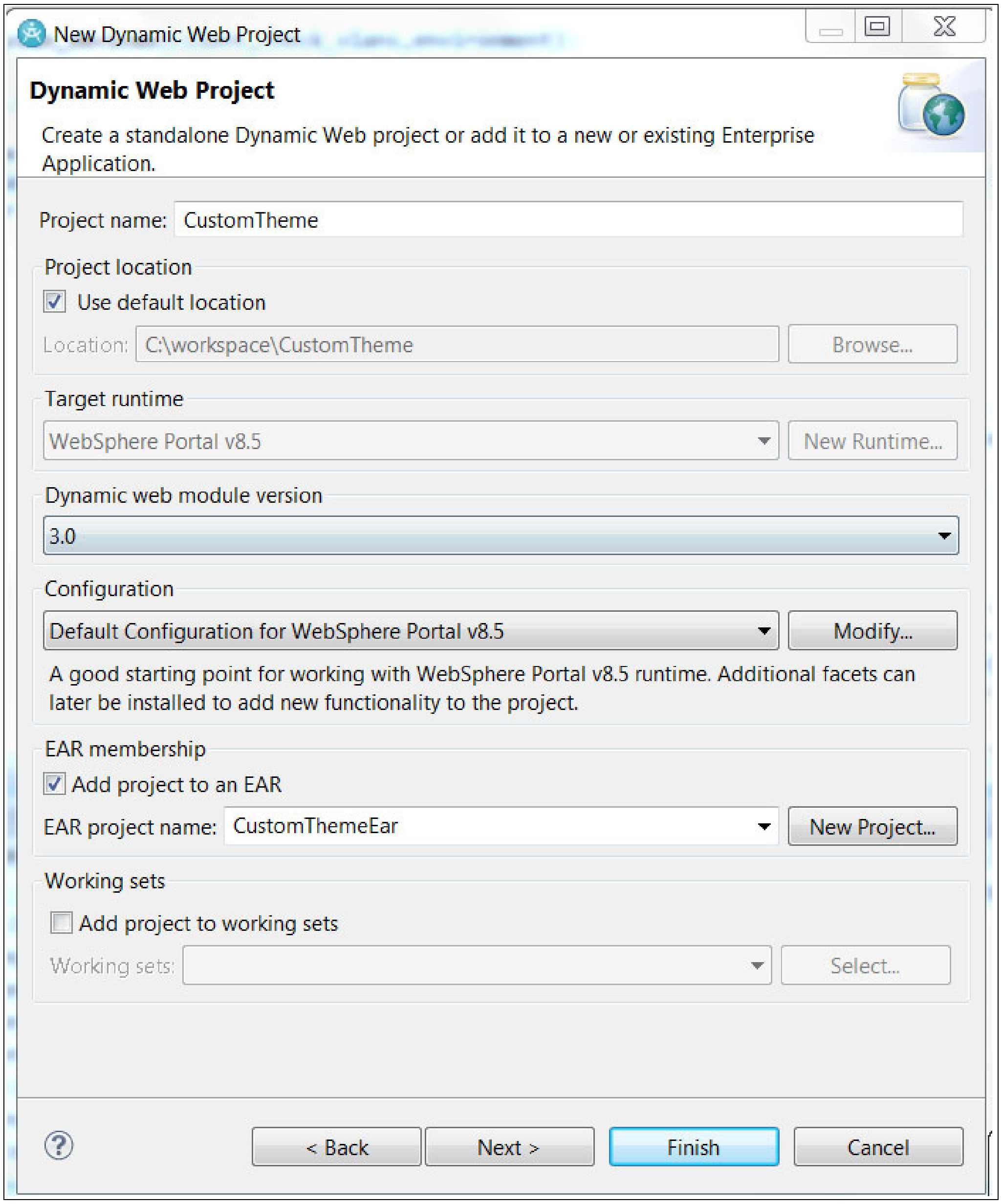


bootstrap	12/14/2015 5:42 PM	File folder
contributions	12/14/2015 5:42 PM	File folder
css	12/14/2015 5:42 PM	File folder
help	12/14/2015 5:42 PM	File folder
layout-templates	12/14/2015 5:42 PM	File folder
menuDefinitions	12/14/2015 5:42 PM	File folder
modules	12/14/2015 5:42 PM	File folder
profiles	12/14/2015 5:42 PM	File folder
skins	12/14/2015 5:42 PM	File folder
system	12/14/2015 5:42 PM	File folder
.hashes	12/14/2015 6:52 PM	HASHES File
.settings	12/14/2015 5:41 PM	SETTINGS File
preview.gif	12/14/2015 2:20 PM	GIF image
readme.txt	12/14/2015 2:20 PM	Text Document
theme.html	12/14/2015 2:20 PM	Firefox HTML Doc...

Figure B-3 File structure in the themedev folder after running the DXSync utility

3. To customize a theme by adding the dynamic content, you must deploy the theme as a web application archive (WAR) file:

   a. Open the IBM Rational Application Developer client and create a Dynamic Web Project. Enter the information that is shown in Figure B-4. In this example, the Project Name is CustomTheme. Click **Finish**.



New Dynamic Web Project

Dynamic Web Project

Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name: CustomTheme

Project location

☑ Use default location

Location: C:\workspace\CustomTheme

Target runtime: WebSphere Portal v8.5

Dynamic web module version: 3.0

Configuration: Default Configuration for WebSphere Portal v8.5

A good starting point for working with WebSphere Portal v8.5 runtime. Additional facets can later be installed to add new functionality to the project.

EAR membership

☑ Add project to an EAR

EAR project name: CustomThemeEar

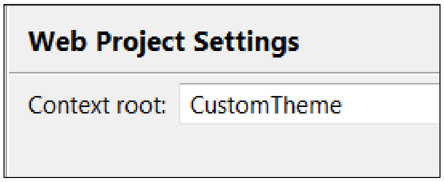
Working sets

□ Add project to working sets

Working sets:

Figure B-4 Create a dynamic web project in Rational Application Developer

b. Change the name of the context root in the Web Project Settings to the project name, which is CustomTheme in this example (Figure B-5).



Web Project Settings

Context root: CustomTheme

Figure B-5 Adjust the context root name in the Web Project Settings

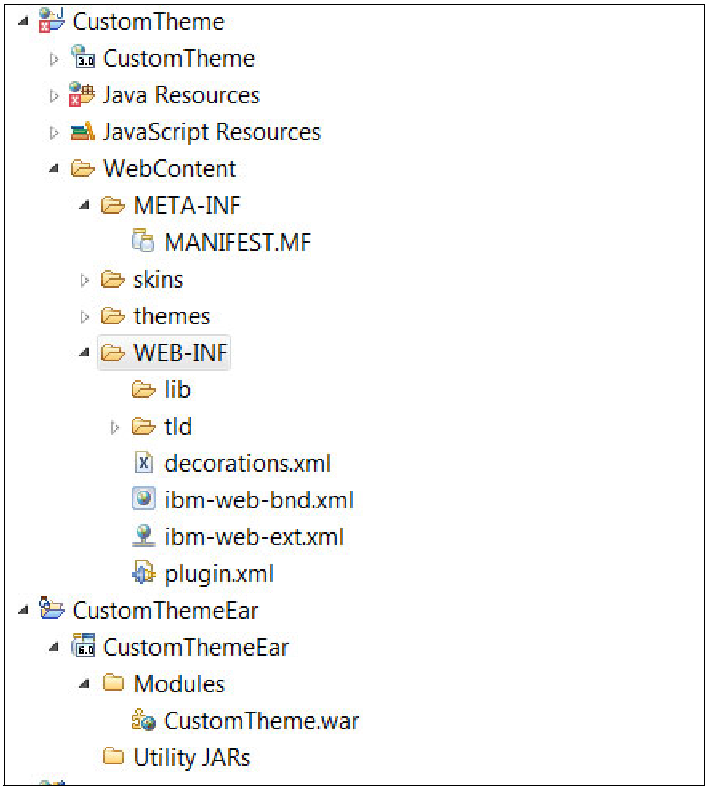
4. Find the WebContent folder in your Rational Application Developer project and copy the following items from your local file system into the WebContent folder:

   – themes folder
   – skins folder
   – tld folder into the WEB-INF folder
   – `decorations.xml` into the WEB-INF folder
   – `plugin.xml` into the WEB-INF folder

   The structure should look similar to the one that is shown in Figure B-6.

   For more information, see the *Copying the dynamic resources for your theme* topic in the IBM WebSphere Portal V8.5 IBM Knowledge Center:

   http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/dev-theme/themeopt_cust_copy_dyntheme.dita



```
CustomTheme
    CustomTheme
    Java Resources
    JavaScript Resources
    WebContent
        META-INF
            MANIFEST.MF
        skins
        themes
        WEB-INF
            lib
            tld
            decorations.xml
            ibm-web-bnd.xml
            ibm-web-ext.xml
            plugin.xml
CustomThemeEar
    CustomThemeEar
        Modules
            CustomTheme.war
        Utility JARs
```

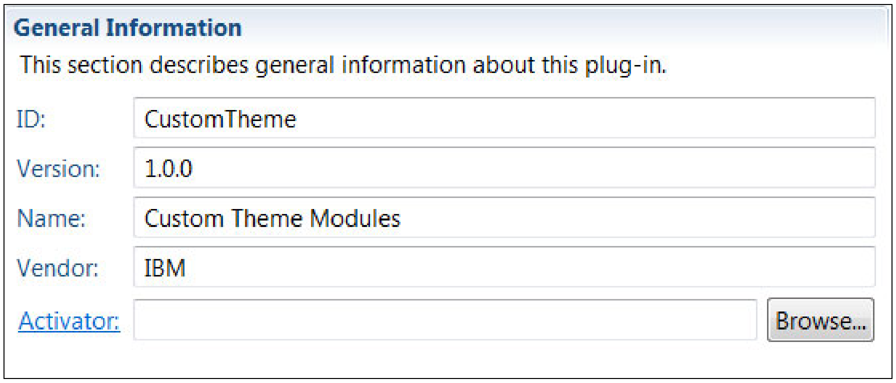
Figure B-6 Final custom theme project structure

5. Modify the `plugin.xml` file. By default, the `plugin.xml` file has an assigned IBM name, ID, and description, but you can change these items to your own unique name and ID. This action helps with validation when the theme is deployed so that the WebSphere Portal Server does not confuse the new custom theme with the default one that exists.

   You can edit the `plugin.xml` file directly in the XML source code or from the XML editor, or by using the IBM Rational Application Developer client. Figure B-7 shows editing the `plugin.xml` file by using the IBM Rational Application Developer client. In this example, the ID is changed to CustomTheme and the Name is changed to Custom Theme Modules.

   For more information about performing this step, see the *Modify the dynamic resource references for your theme* topic in the IBM WebSphere Portal V8.5 IBM Knowledge Center:

   http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/mp/dev-theme/themeopt_cust_copy_modifystatres.dita



General Information

This section describes general information about this plug-in.

ID: CustomTheme

Version: 1.0.0

Name: Custom Theme Modules

Vendor: IBM

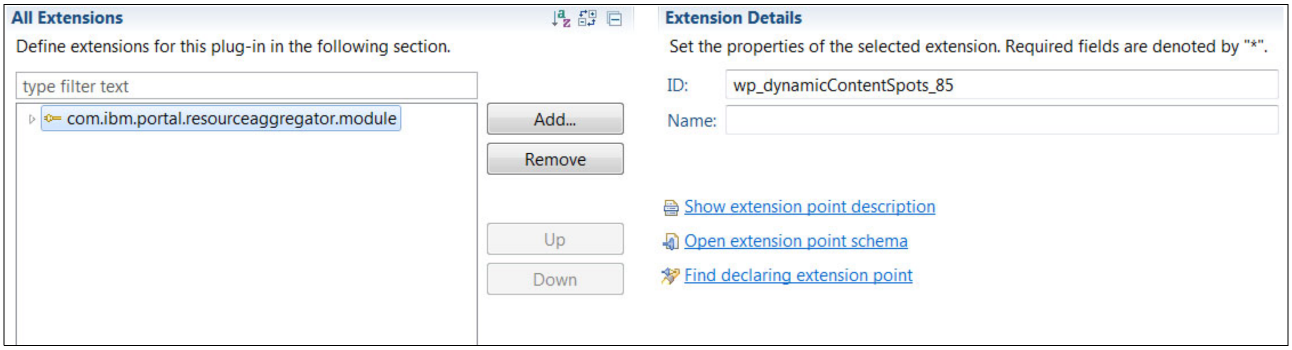
Activator:

Figure B-7 Update plugin.xml

6. Similar to modules, the `plugin.xml` file has list of modules. If you create your own dynamic content spot, you can rename the name of the module in the list to distinguish it from the default module. For example, if you create your own custom footer for the theme, you can reference it by adjusting the following reference in the `plugin.xml` file:

   ```
   <sub-contribution type="markup" ref-id="customtheme_footer">
   ```

   You must modify the dynamic resource references to link to the static resources for your theme. Dynamic content spots are defined through a module called `wp_dynamicContentSpots_85`. This module is defined in the `plugin.xml file`, which was copied when you copied your theme. See Figure B-8.



All Extensions

Define extensions for this plug-in in the following section.

type filter text

com.ibm.portal.resourceaggregator.module

Extension Details

Set the properties of the selected extension. Required fields are denoted by "*".

ID: wp_dynamicContentSpots_85

Name:

Add...
Remove
Up
Down
Show extension point description
Open extension point schema
Find declaring extension point

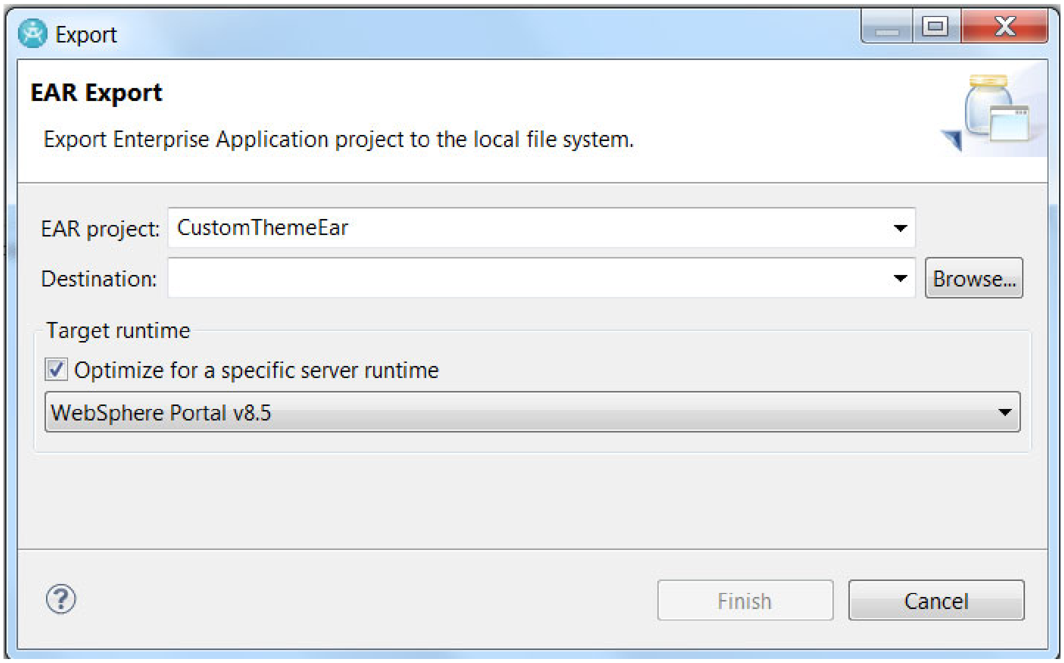
Figure B-8 Updating plugin.xml modules

For more information about dynamic content spots, see 3.5.5, “Dynamic content spots” on page 51.

7. Copy the static resources created in the themelist folder by using the WebDAV client. For example, the following resources are copied:

```
http://host:port/wps/mycontenthandler/dav/themelist/
http://host:port/wps/mycontenthandler/dav/skinlist/
```

8. Export the theme as a deployable archive. Right-click the EAR project and export it as a `*.ear` archive onto the file system, as shown in Figure B-9.



EAR Export

Export Enterprise Application project to the local file system.

EAR project: CustomThemeEar

Destination:

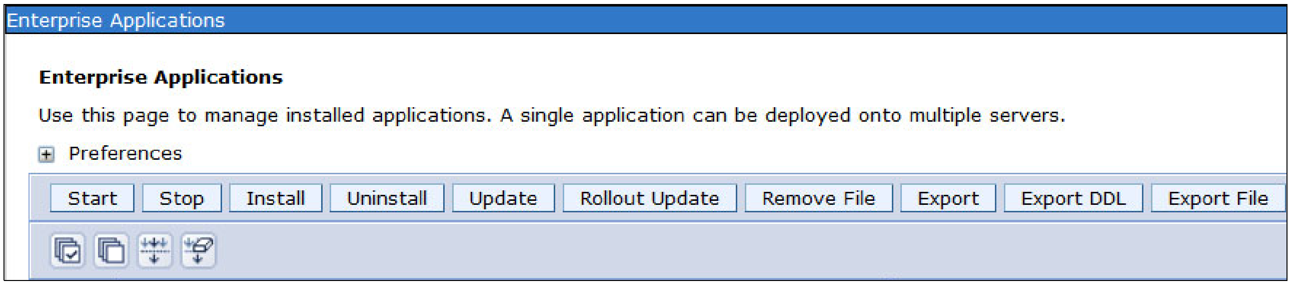
Target runtime

☑ Optimize for a specific server runtime

WebSphere Portal v8.5

Figure B-9 Export a theme as an EAR file

9. Deploy the theme EAR file by completing the following steps:
   a. Open the WebSphere Application Server Administrative Console.
   b. Click **WebSphere enterprise application** to open the Enterprise Applications window.
   c. On the Enterprise Applications window, click **Install** (Figure B-10).



Enterprise Applications

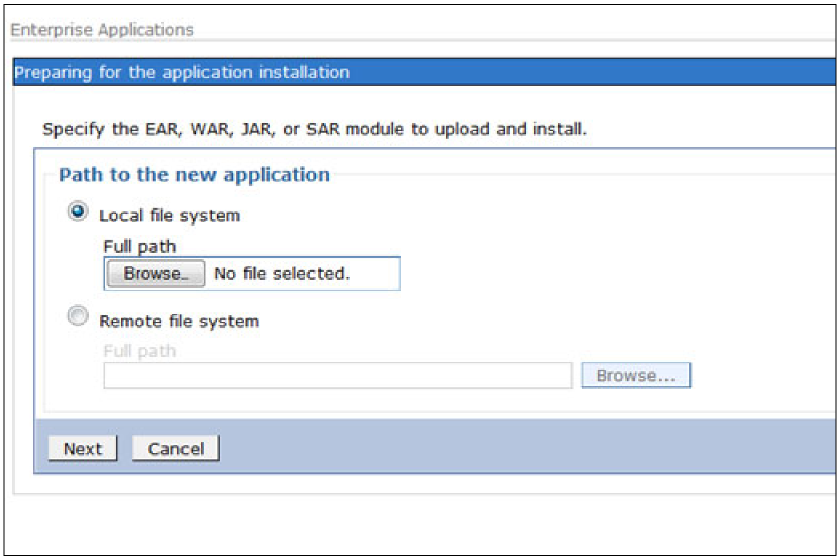
Enterprise Applications

Use this page to manage installed applications. A single application can be deployed onto multiple servers.

⊞ Preferences

Figure B-10 Enterprise Applications window

d. Specify the path to the `*.ear` file created earlier, as shown in Figure B-11. Click **Next**.



Enterprise Applications

Preparing for the application installation

Specify the EAR, WAR, JAR, or SAR module to upload and install.

Path to the new application

● Local file system

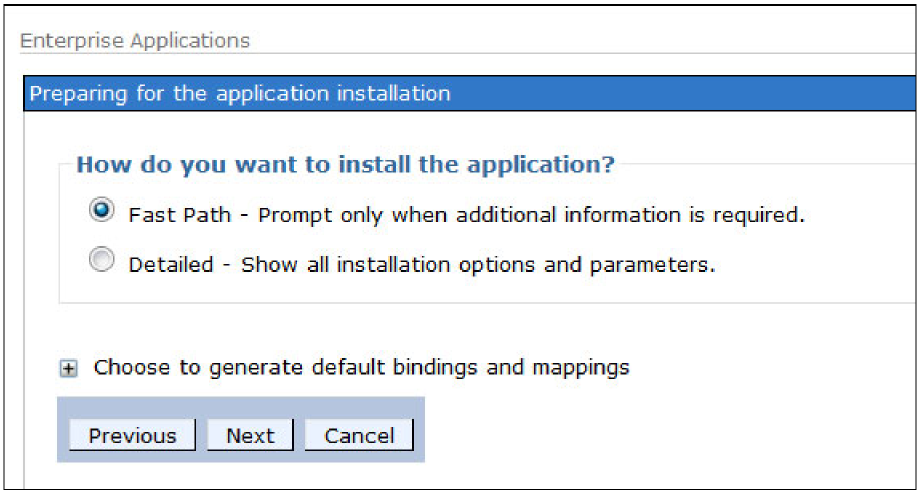
Full path: No file selected.

○ Remote file system

Full path

Figure B-11 Deploy the theme EAR file

e. When prompted for how you want to install the application, select **Fast Path**, as shown in Figure B-12. Then, expand **Choose to generate default bindings and mappings** and select **Generate Default Bindings**. Click **Next**.



Enterprise Applications

Preparing for the application installation

How do you want to install the application?

● Fast Path - Prompt only when additional information is required.

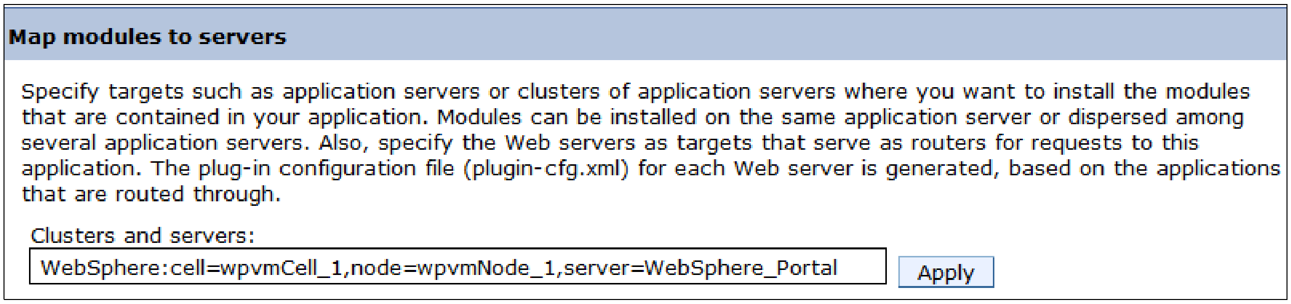
○ Detailed - Show all installation options and parameters.

⊞ Choose to generate default bindings and mappings

Figure B-12 Fast Path selection

f. Select your installation option values and click **Next**.

g. On the Map Modules to Servers window (Figure B-13), select the custom theme module and then select **server=WebSphere_Portal**. Click **Apply** and then click **Next**.



Map modules to servers

Specify targets such as application servers or clusters of application servers where you want to install the modules that are contained in your application. Modules can be installed on the same application server or dispersed among several application servers. Also, specify the Web servers as targets that serve as routers for requests to this application. The plug-in configuration file (plugin-cfg.xml) for each Web server is generated, based on the applications that are routed through.

Clusters and servers:

WebSphere:cell=wpvmCell_1,node=wpvmNode_1,server=WebSphere_Portal

Figure B-13 Mapping module to server

h. Click **Finish**.

i. When the EAR file is installed, click **Save directly to the master configuration**.

j. Select the check box next to your custom theme EAR file (GWTheme in this example) in the table of enterprise applications and click **Start**.

To make the theme available in WebSphere Portal, complete the following steps:

1. Open the WebSphere Portal administration console.
2. Click **Portal User Interface** → **Themes and Skins**.
3. Select your custom theme (SimpleCustom in this example) in the themes list and click **Edit theme**.
4. Select the skin that you want to set as default in Skins for this theme list and click **Set as default portal skin**.
5. Click **OK**.

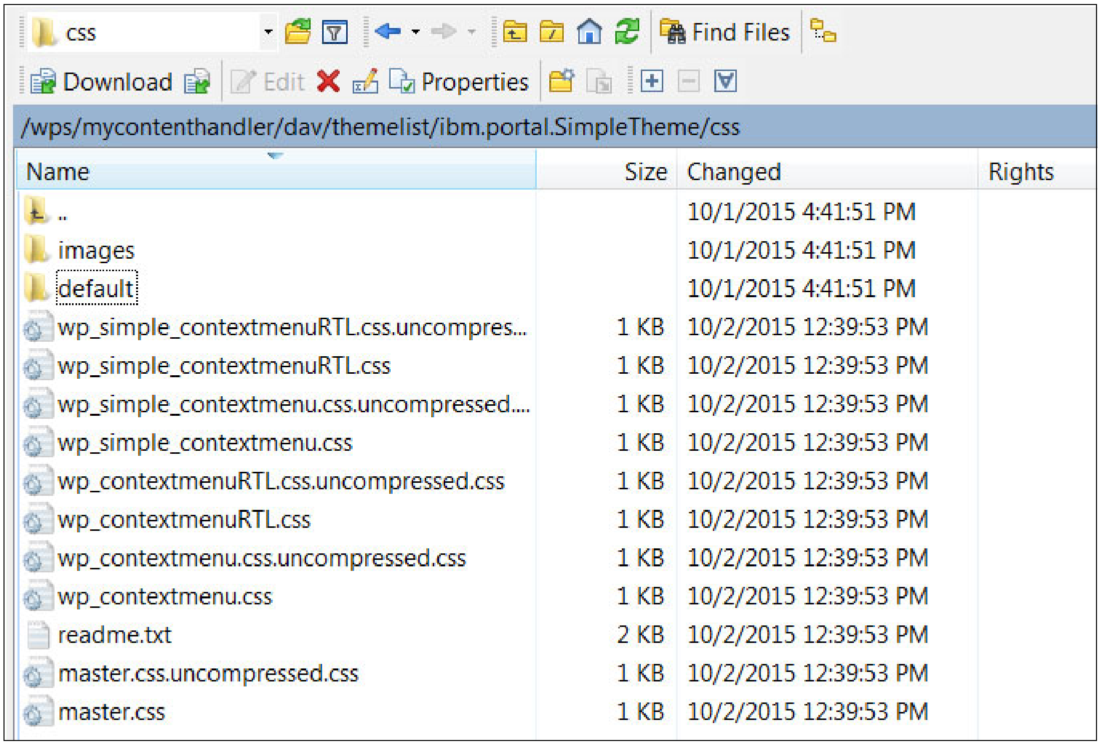
The custom theme is now available for use in the portal. To verify, create a portal page, edit the page properties, and assign your theme to the page.

Styles

As described in 3.1.2, “Theme resources” on page 36, the theme includes dynamic and static resources. You can customize the static resources, which include CSS stylesheets and JavaScript files.

The CSS style files can be edited and created in Rational Application Developer, and include style class definitions, either new or overriding the existing class definitions.

For example, if you create a `custom.css` stylesheet file by using the WebDAV client, you can copy this file into the css folder of your customized theme. You can then open the `styles.json` file for editing. This file is in the system folder of your custom theme. Here, you register the styles that are created. See Figure B-14.



/wps/mycontenthandler/dav/themelist/ibm.portal.SimpleTheme/css

Name	Size	Changed	Rights
..		10/1/2015 4:41:51 PM	
images		10/1/2015 4:41:51 PM	
default		10/1/2015 4:41:51 PM	
wp_simple_contextmenuRTL.css.uncompres...	1 KB	10/2/2015 12:39:53 PM	
wp_simple_contextmenuRTL.css	1 KB	10/2/2015 12:39:53 PM	
wp_simple_contextmenu.css.uncompressed....	1 KB	10/2/2015 12:39:53 PM	
wp_simple_contextmenu.css	1 KB	10/2/2015 12:39:53 PM	
wp_contextmenuRTL.css.uncompressed.css	1 KB	10/2/2015 12:39:53 PM	
wp_contextmenuRTL.css	1 KB	10/2/2015 12:39:53 PM	
wp_contextmenu.css.uncompressed.css	1 KB	10/2/2015 12:39:53 PM	
wp_contextmenu.css	1 KB	10/2/2015 12:39:53 PM	
readme.txt	2 KB	10/2/2015 12:39:53 PM	
master.css.uncompressed.css	1 KB	10/2/2015 12:39:53 PM	
master.css	1 KB	10/2/2015 12:39:53 PM	

Figure B-14 Style definitions

Table B-1 provides a list of CSS style files and their descriptions for the Simple theme.

Table B-1 CSS style files for the Simple theme

CSS style files	Description
`/css/master.css`	All individual CSS files that belong to the theme.
`/css/default/banner.css`	Classes that support the banner that appears at the beginning of the page.
`/css/default/content.css`	Classes that support the main content portion of the page.
`/css/default/contextmenu*.css`	Classes that support the simple menu framework.
`/css/default/footer.css`	Classes that support the footer that appears at the end of the page.
`/css/default/general.css`	Classes that are applied to HTML tags.
`/css/default/images.css`	Classes that are applied to SVG images, such as size and fill color.
`/css/default/navigation.css`	Classes that are applied to one_level, two_level, and mega_menu navigation.
`/css/default/skinLayout.css`	Classes that support skins and what is visible in the content area in Edit mode.
`/css/default/utility.css`	Generalized classes that are used throughout the theme.
`/css/default/wp_contextmenu*.css`	Classes that support the menu framework.

Navigation

You can create a custom navigation toolbar based on the theme artifacts that are provided by WebSphere Portal. The starting point for customization is the `theme.html` file. The `theme.html` file links to the dynamic content spots so you can change or replace these dynamic artifacts with your own implementation.

Such an implementation involves writing JSP fragment code. However, in case of the Simple theme, the idea is to use generic markup that most people do not have to change, but create styles for, so there is no requirement to change the JSPs.

There are a wide range of options for building navigation. For example, you can code a dynamic spot to include links to Web Content Management artifacts, content, or component items. This approach is useful when you want to create a navigation based on the logical structure of site areas in Web Content Manager, or if you design a common footer (such as `About us` or `Address`) fragment where the links point to the data that is stored in Web Content Manager.

Dynamic artifacts are defined through a module that is referenced in the `plugin.xml` file. The `plugin.xml` file is in the WEB-INF folder of the theme that is shipped with WebSphere Portal. You copied this folder in step 4 on page 229. The sample `plugin.xml` file that is shown in Example B-1 references a dynamic spot.

The Simple theme provides four options for navigation: st_one_level_nav, st_two_level_nav, st_mega_menu_nav, and st_bootstrap_nav. Each of these options includes a dynamic content spot.

For more information about dynamic spots, see 3.5.5, “Dynamic content spots” on page 51.

Example B-1 Reference a dynamic spot

```
<a rel="dynamic-content"
   href="res:/CustomThemeContext/themes/html/MyTheme/dynamicContent.jsp"></a>
```

The `theme.html` file customization is shown in Example B-2.

Example B-2 Customized theme.html file

```
<title>IBM Digital Experience</title>
                    <use xlink:href="#stBee"></use>
                 </svg>
              </a>
           </div><!-- end logo -->

           <div id="stNavigation" class="stBanner stNavigation">
              <!-- Option 1: one level navigation
              <a rel="dynamic-content" href="dyn-cs:id:st_one_level_nav"></a>
              -->
              <!-- Option 2: two level navigation
              <a rel="dynamic-content" href="dyn-cs:id:st_two_level_nav"></a>
              -->
              <!-- Option 3: three level mega menu navigation -->
              <a rel="dynamic-content" href="dyn-cs:id:st_mega_menu_nav"></a>

              <!-- Option 4: Boostrap navigation
              <a rel="dynamic-content" href="dyn-cs:id:st_bootstrap_nav"></a>
```

```
               -->
            </div><!--end primary nav banner-->
         </div><!--end main banner-->
      </header>

      <!-- these wptheme* classes are required for the hidden widget container -->
      <div class="stMainContent wpthemeMainContent" role="main" aria-label="IBM
Digital Experience Main">
         <!-- required - do not remove -->
         <a rel="dynamic-content"
href="state.portlet:portlet?include=windowState&amp;include=portletMode&amp;mime-t
ype=text/html"></a>

         <div id="layoutContainers" class="stLayoutContainers
stLayoutContainersHidden stGroup wpthemeLayoutContainersHidden">
            <a rel="dynamic-content" href="lm:template"></a>
         </div>
      </div>
```

Example B-3 demonstrates the customization of styles, defining the colors and widths of drop-down menus in the navigation toolbar.

Example B-3 Style definitions

```
/*** Page navigation (as opposed to footer) ***/
.stPageNavigation div {
   float: left;
}
.stPageNavigation nav:after {
   content: "";
   display: table;
   clear: both;
}

/* top level navigation */
.stPageNavigation nav ul {
   padding: 0;
   margin: 0;
   list-style: none;
   overflow: visible
}
/* top level list item */
.stPageNavigation nav ul li {
   margin: 0;
   display: inline;
   float: none;
}
/* secondary level list item */
.stPageNavigation nav ul li li{
   display: inline-block;
   float: left;
}
/* at 768px display of navigation becomes vertical */
@media ( max-width : 768px) {
   .stPageNavigation nav ul li {
      border-top: 1px solid #006d5d;
      margin: 0px;
      display: block;
      float: left;
      width: 100%;
   }
   /* no border in children */
   .stPageNavigation nav ul li div li {
      border-top: none;
   }
}
/* unselected top level links */
.stPageNavigation nav > ul > li > div > a {
   color: #fff;
   display: block;
   font-weight: normal;
   padding: 16px 20px;
   text-decoration: none;
   font-size: 16px;
   text-transform: uppercase;
   max-width: 200px;
   overflow: hidden;
   text-overflow: ellipsis;
   white-space: nowrap;
}
.stNoTouch .stPageNavigation nav > ul > li > div > a:hover, .stNoTouch
.stPageNavigation nav > ul > li > div > a:focus {
   color: #fff;
   text-decoration: none;
}
/* top level link child indicator */
.stPageNavigation .stNavHasChild {
   padding: 0 0 0 6px;
}
@media ( max-width : 768px) {
   .stPageNavigation nav > ul > li > div > a {
      padding: 0.875em 1.25em;
   }
   /* area around the link */
   .stNavTapArea {
      width: 200px;
   }
   /* top level link child indicator not visible */
   .stPageNavigation .stNavHasChild {
      display: none;
   }
}
/* selected top level link */
.stPageNavigation .stNavSelected > div {
   background-color: #008571;
}
/* second level in menu is vertical */
.stPageNavigation nav ul ul ul li {
   display: block;
   float: none;
   font-weight: normal;
   text-transform: none;
}
/* second level in menu */
.stPageNavigation nav ul div ul ul {
   float: none;
   display: list-item;
   position: relative;
}
/* column padding */
.stPageNavigation div > ul > li {
    padding-bottom: 1em;
}
/* top level (horizontal) padding */
.stPageNavigation li > div > ul > li  > a{
    padding: 1em 4.167em .5em 0;
}
/* lower level (vertical) padding */
.stPageNavigation div > ul > li > ul > li {
    padding: .5em 4.167em .5em 0;
}
@media ( max-width : 768px) {
   /* column padding */
   .stPageNavigation div > ul > li {
      padding-bottom: 0;
   }
   /* top level (horizontal) padding */
   .stPageNavigation li > div > ul > li  > a{
      padding: 1.071em 1.429em;
   }
   /* lower level */
   .stPageNavigation div > ul > li > ul > li {
      padding: 1.071em 1.429em;
   }
}
/* second level is also the top level of the mega-menu (horizontal) */
.stPageNavigation div > ul > li {
   font-size: 14px;
   font-weight: bold;
   color: #121212;
   text-transform: uppercase;
}
/* second level padding */
.stPageNavigation div > ul > li > a{
    padding: 1em 2.5em 1em 0;
}
/* parent page link within the sub menu */
.stPageNavigation .stNavParent {
   font-weight: normal;
   text-transform: uppercase;
}
/* second level links */
.stPageNavigation .stNavChild li a:link,.stPageNavigation .stNavChild li a:visited
{
   background-color: #ffffff;
   color: #000000;
   display: block;
   text-decoration: none;
   width: 100%;
   overflow: hidden;
   text-overflow: ellipsis;
   max-width: 200px;
}
/* second level node selected, second level node hover */
.stPageNavigation .stNavChild .stNavChildSelected a span, .stPageNavigation
.stNavChild a:hover span {
   color: #006d5d;
}
/* mobile navigation second level links */
@media ( max-width : 768px) {
   /* parent page link within the sub menu */
   .stPageNavigation .stNavParent {
      display: none;
   }
   .stPageNavigation .stNavChild li a:link,.stPageNavigation .stNavChild li
a:visited {
      background: none;
      color: #ddd;
      max-width: 100%
   }
   .stPageNavigation .stNavChild .stNavChildSelected a span {
      color: #6eedd8;
   }
}
/* second level node selected but third is not*/
.stPageNavigation .stNavChild .stNavChildSelected ul li a span {
   color: #000000;
}
@media ( max-width : 768px) {
   .stPageNavigation .stNavChild .stNavChildSelected ul li a span {
      color: #ddd;
   }
}
/* second level navigation for wide screen */
@media ( min-width : 768px) {
   /* the div that contains the second level */
   .stPageNavigation .stOpenNavLevel.stSecondLevelNav {
      position: absolute;
      float: left;
      background: none repeat scroll 0 0 #FFFFFF;
      border-bottom: 1px solid #006d5d;
      left: 0;
      right: 0;
      padding-left: 25px;
      z-index: 99;
   }
}
/* the mobile navigation second level */
@media ( max-width : 768px) {
   .stPageNavigation .stOpenNavLevel.stSecondLevelNav {
      background: none repeat scroll 0 0 #003c32;
      position: inherit;
   }
   .stPageNavigation nav ul div {
      position: inherit;
      background: none;
      border-bottom: none;
      padding-left: 0;
      width: 100%;
   }
   .stPageNavigation nav ul .stSecondLevelNavOpen {
      display: block;
   }
}
/* keep secondary nav div open when secondary page is selected */
.stPageNavigation .stOpenNavLevel {
   display: block;
}
```

Example B-4 shows a code snippet for a dynamic spot example for customizing a logo.

Example B-4 Example of dynamic spot for customizing a logo

```
/*** BANNER ***/
/**
   This contains the styles that are used to display the banner at the top of the 
page. The banner contains the
   logo, navigation, actions menu, search, and profile menu. This is a good place 
to begin customizing the
   look of your theme.
**/
.stBanner {
   background-color: #006d5d;
   padding: 0;
   position: static;
   box-sizing: border-box;
   font-size: 75%;
}
.stBanner.stGroup {
   /*height: 55px; (this makes menu disappear if there are too many pages) */
   overflow: hidden;
}
.stBanner ul {
   padding: 0;
   overflow: visible;
   list-style-type: none;
}
/* Banner menus */
ul.stBannerMenus>li {
   background-color: #006d5d;
   display: block;
   float: left;
   margin: 0;
   color: #fff;
   padding: 0;
   height: 54px;
}
```

```
.stNoTouch ul.stBannerMenus > li > a:hover, .stNoTouch ul.stBannerMenus > li >
a.selected {
   background-color: #008571;
}
.stBanner ul.stBannerMenus>li>span,.stBanner ul.stBannerMenus li a:link,.stBanner
ul.stBannerMenus li a:visited,.stBanner ul.stBannerMenus li a:active {
   color: #fff;
   font-weight: normal;
   display: -webkit-inline-flex;
   display: -ms-inline-flexbox;
   display: inline-flex;
   margin: 0;
   padding: 15px;
}
.stBanner .stBranding {
   display: inline-block;
   border: 0;
   padding: 4px 25px 0;
   height: 48px;
}
@media ( max-width : 768px) {
   .stBanner .stBranding {
      padding: 2px 15px 0;
   }
}
.stBannerPhoto {
   height: 24px;
   width:24px;
   border-radius: 50%;
}
```

XMLAccess

XMLAccess (or portal configuration interface) is a command-line batch-processing utility for exporting and importing various portal configuration. You can use this utility to perform portal configuration updates. You can use XMLAccess to export an entire portal configuration or parts of a configuration, for example, specific pages, to an XML file. You can then re-create the exported configuration from such a file on another portal.

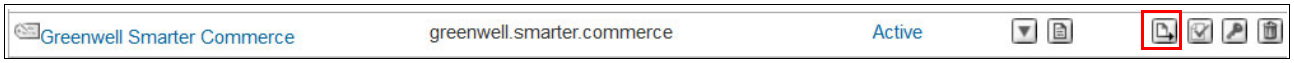
XMLAccess makes it easier for the duplication of a portal environment, for example, from a development server to a production server.

There are many XMLAccess examples that are shipped with WebSphere Portal. You can find these examples in the following directory:

```
portalserver/doc/xml-samples
```

To export a page hierarchy, complete the following steps:

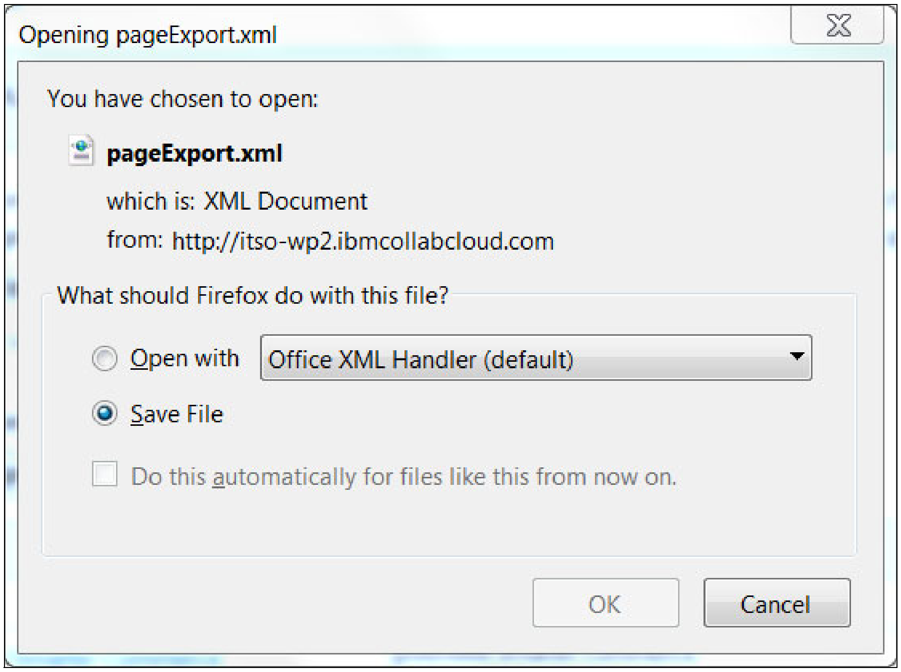
1. Log in to the WebSphere Portal administration console.
2. Go to **Manage Pages** and select the page structure that you want to export. Click the **Export** icon, as shown in Figure B-15.



Greenwell Smarter Commerce greenwell.smarter.commerce Active

Figure B-15 Export a page hierarchy

3. When prompted, click **Yes** to export the page hierarchy.
4. Save the exported XML file to the local file system, as shown in Figure B-16.



Opening pageExport.xml

You have chosen to open:

pageExport.xml

which is: XML Document

from: http://itso-wp2.ibmcollabcloud.com

What should Firefox do with this file?

○ Open with Office XML Handler (default)

● Save File

□ Do this automatically for files like this from now on.

Figure B-16 Save the exported XML file

If you must re-create the page structure, you can import the XML file by completing the following steps:

1. From the WebSphere Portal administration console, click **Portal Settings** → **Import XML** (Figure B-17).

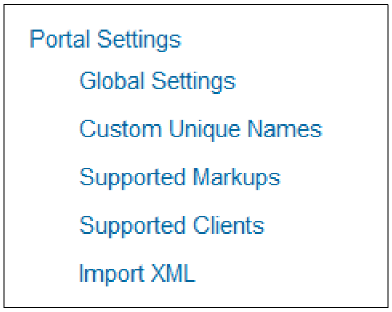
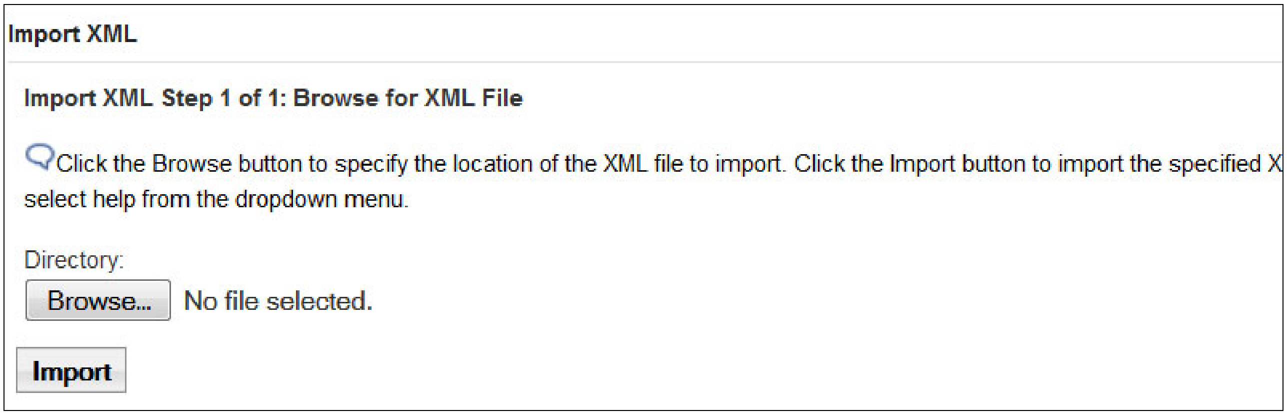


Figure B-17 Import an XML file

2. On the Import XML window, click **Browse** to select the XML file to import from the local file system and then click **Import** (Figure B-18). The page structure is restored.



Import XML

Import XML Step 1 of 1: Browse for XML File

Click the Browse button to specify the location of the XML file to import. Click the Import button to import the specified X select help from the dropdown menu.

Directory:

Figure B-18 Select the XML file to import

C

Additional material

This appendix refers to additional material that can be downloaded from the internet, as described in the following sections.

Locating the web material

The web material that is associated with this book is available in softcopy on the Internet from the IBM Redbooks web server. Point your web browser at the following website:

ftp://www.redbooks.ibm.com/redbooks/SG248313/

Alternatively, you can go to the IBM Redbooks website:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the IBM Redbooks form number, SG248313.

Using the web material

The additional web material that accompanies this book includes the following files:

File name	*Description*
ITSO_WAB_Filter.zip	Compressed code samples
SP_BPM_Analysis.zip	Compressed HTML documents

Downloading and extracting the web material

Create a subdirectory (folder) on your workstation, and extract the contents of the compressed files into this folder.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

Other publications

These publications are also relevant as further information sources:

- *Developing Themes for WebSphere Portal 8.5* article:

  http://www.lotus.com/ldd/portalwiki.nsf/xpViewCategories.xsp?lookupName=Developing%20Themes%20for%20WebSphere%20Portal%208.5
- IBM Digital Experience File Sync documentation:

  https://github.com/digexp/dxsync/wiki
- IBM Digital Experience: IBM WebSphere Portal and IBM Web Content Manager V8.5 documentation:

  http://www.ibm.com/support/knowledgecenter/SSHRKX_8.5.0/welcome/wp_welcome.html
- “Local Files” – Custom DDC Plug-in Example available now!:

  https://developer.ibm.com/digexp/blog/2014/08/04/local-files-custom-ddc-plugin-example-available-now/
- Step-by-Step Cluster Guide for IBM WebSphere Portal V8.5:

  http://www.lotus.com/ldd/portalwiki.nsf/dx/Step-by-Step_Cluster_Guide_for_IBM_WebSphere_Portal_v8.5

Online resources

These websites are also relevant as further information sources:

- IBM Business Process Manager Sample for IBM Digital Data Connector in WebSphere Portal:

  http://openntf.org/main.nsf/project.xsp?r=project/Business%20Process%20Manager%20Sample%20for%20IBM%20Digital%20Data%20Connector%20in%20WebSphere%20Portal
- IBM Collaboration Solutions Catalog:

  https://greenhouse.lotus.com/catalog/
- IBM Customer Experience Suite:

  http://www.ibm.com/software/products/en/ibmcustxpersuit
- IBM Digital Experience:

  http://www.ibm.com/software/products/en/category/digital-experience
- IBM Digital Experience Cloud websites:
  - http://www.ibm.com/software/collaboration/digitalexperience/cloud.html
  - http://www.ibm.com/software/products/en/digital-experience-cloud

- IBM Digital Experience Manager:

  http://www.ibm.com/software/products/en/ibm-digital-experience-manager
- IBM Employee Experience Suite:

  http://www.ibm.com/software/products/en/employee-experience-suite
- IBM Mobile Portal Accelerator:

  http://www.ibm.com/software/products/en/ibmmobiportacce
- IBM WebSphere Commerce Sample for IBM Digital Data Connector in WebSphere Portal:

  http://www.openntf.org/main.nsf/project.xsp?r=project/WebSphere%20Commerce%20Sample%20for%20IBM%20Digital%20Data%20Connector%20in%20WebSphere%20Portal
- JSoup Library:

  http://jsoup.org/

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Redbooks

Building Integrated Websites with IBM Digital Experience

(0.2”spine)
0.17”<->0.473”
90<->249 pages

SG24-8313-00

ISBN 0738441341

Printed in U.S.A.