



IBM Lotus Web Content Management software and enterprise content management integration patterns.

Dr. Dieter Buehler, architect, WebSphere Portal development team, IBM Software Group

Krishnan Hariharan, offering manager, Lotus Web Content Management software, IBM Software Group

Stefan Hepper, WebSphere Portal and Lotus Web Content Management System team, IBM Software Group

Steve Pritko, architect, Lotus Web Content Management and WebSphere Portal software, IBM Software Group

| Contents | |
|----------|--|
| 2 | Introduction |
| 4 | Scenarios for integration with ECM: what's new in Lotus Web Content Management 6.1.5 software |
| 15 | Writing a custom resource and resource collection |
| 17 | Integration using JSP components |
| 23 | Integration using the Lotus Web Content Management API |
| 24 | Which scenario should you choose? |
| 25 | Why IBM? |
| 26 | Accelerate the time to value of your software investment |

Introduction

Large organizations are rapidly reaching the conclusion that unstructured content needs to be managed in the same manner that critical corporate data and information located in databases has been managed for decades. This becomes even more important as corporations are now adopting ways to leverage the Internet to conduct their business, thus making the reuse of content more crucial. Hence, reduction of disparate content repositories and/or federation of critical corporate content existing in those repositories are critical to attaining more efficient management of resources and concomitantly unleashing the value of the previously isolated content in order to achieve better utilization and business value from that data. Furthermore, adopting an enterprise strategy for content management enables the organization to more effectively apply corporate standards and manage resources in comparison to the uncoordinated and costly management of a dozen separate document systems and their integration with an online presence.

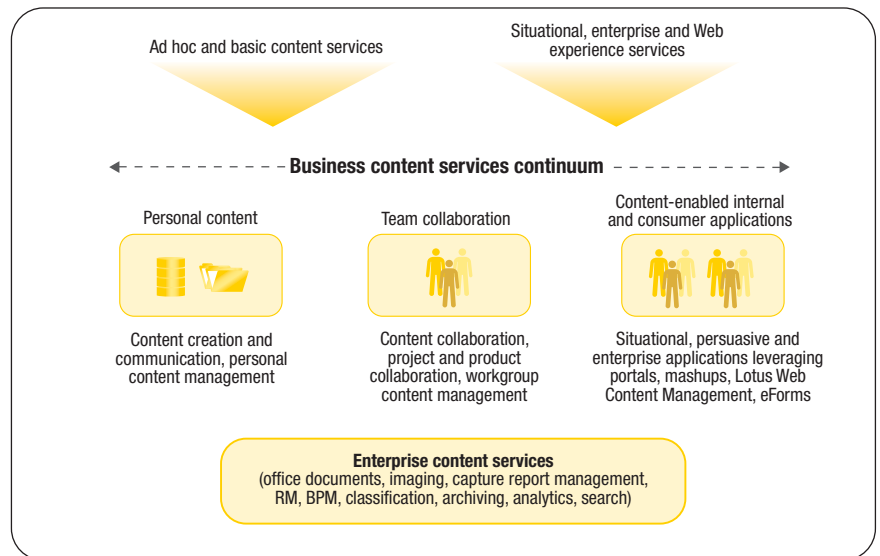


Figure 1. The IBM market view for content

Ad hoc and basic content services provide direct access to enterprise content management (ECM) repositories through a desktop and collaborative interface, enabling workers to leverage tools they are familiar with as an alternative choice to ECM repositories from IBM, where their enterprise content is securely stored. With increased adoption of Internet-based solutions for public-facing, internal-facing or partner Web sites, situational, enterprise and Web experience services provide a single interface for IBM clients to leverage existing investments in team collaboration tools (such as IBM Lotus® Quickr™ software), ECM repositories (IBM FileNet® and IBM Content Manager software) and publish their documents to the Internet via IBM WebSphere® Portal, IBM Lotus Mashups and IBM Lotus Web Content Management software. A single Web content interface, via a Lotus Web Content Management solution, can help provide mechanisms to align accurate, up-to-date information with the people who affect your business.

A powerful and sophisticated Lotus Web Content Management solution can:

- *Accelerate the development and delivery of critical business information across the enterprise.*
- *Enable organizations to create, maintain and publish Web content to Internet, intranet, extranet and portal assets with comprehensive capabilities.*
- *Deliver content to specific audiences via a high level of personalization and through a variety of channels.*

This paper explores patterns for integrating Lotus Web Content Management, FileNet, Content Manager and Lotus Quickr software. Many of the approaches presented in this paper apply to other content management systems as well.

Scenarios for integration with ECM: what's new in Lotus Web Content Management 6.1.5 software

The IBM WebSphere Portal 6.1.5 feature pack provides a new set of integration capabilities with ECM and collaborative document repositories such as IBM FileNet 4.0.1, IBM Content Manager 8.4 and IBM Lotus Quickr 8.1.1 software. All of the previously mentioned products support a common REST protocol to access folders and documents. WebSphere Portal 6.1.5 software leverages this protocol to allow you to:

- *Select a document and insert that document as a link via the rich text editor into a Lotus Web Content Management content item.*
- *Display a live feed from the ECM system via a Lotus Web Content Management personalization component. WebSphere Portal 6.1.5, which Lotus Web Content Management takes advantage of, provides a special personalization component that understands the Representational State Transfer (REST) feed and gives you access to all the metadata in that feed. This allows you to create a customized view of that feed to the ECM system using Lotus Web Content Management software. More details on this can be found in the Federated Documents resource collection section.*

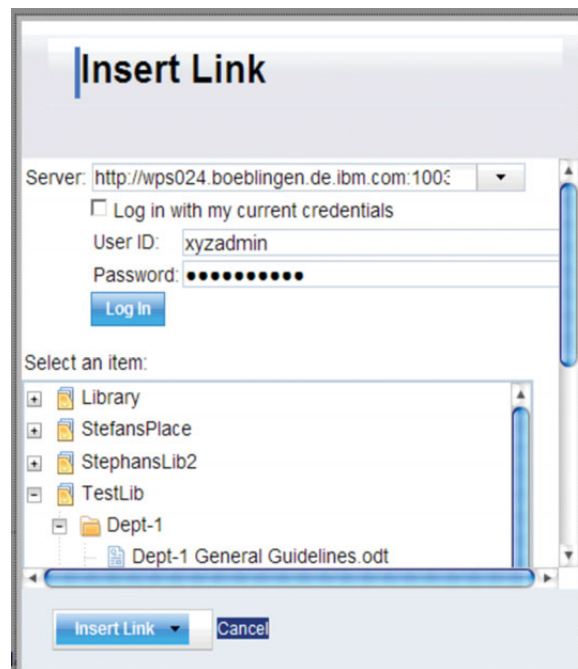


Figure 2. Document picker for Lotus Web Content Management software.

In addition to accessing the ECM system via the above-mentioned REST service, Lotus Web Content Management 6.1.5 software includes the Web Content Integrator (WCI) component in order to consume documents with their metadata from the ECM system into Lotus Web Content Management software. The WCI component allows consuming RSS feeds and translating them into Lotus Web Content Management content items. WebSphere Portal software provides converters for IBM FileNet, IBM Content Manager and Lotus Quickr software. This solution helps store and manage documents in compliance with the auditing requirements in the ECM system and publishes the latest version of the document to Lotus Web Content Management software in order to render the document with the best performance out of Lotus Web Content Management software. More details on the WCI can be found in the Web Content Integrator section.

Another integration capability that Lotus Web Content Management 6.1.5 software provides is access via the Web Distributed Authoring and Versioning (WebDAV) protocol. WebDAV is a standard (<http://tools.ietf.org/html/rfc4918>) that defines an extension on top of the HTTP protocol for WebDAV. The implementation of Lotus Web Content Management 6.1.5 software allows content owners to read, update, create and delete file resources, images, presentation templates and categories. This protocol can be used to push files or images from the ECM system to the Lotus Web Content Management system via HTTP.

These newly added capabilities make it very easy to reference documents in ECM systems, display complete folder views of the ECM system in WebSphere Portal software, or even copy a complete set of documents into Lotus Web Content Management software and render them out of the Lotus Web Content Management environment.

WebSphere Portal Personalization capabilities

WebSphere Portal software includes a robust personalization engine that can be used to integrate with external content sources. Once these integrations are configured, the data sets are referred to as resource collections. Different selection criteria can be used to query these resource collections.

This integration approach returns a list of content objects—for example, documents, images and database records. Lotus Web Content Management software manages the presentation of content that is returned by a query of the resource collection.

Lotus Web Content Management software includes built-in resource collections for Web content, federated documents (cf. the Federated Documents resource collection) and portal user information. To generate code for resource collections for Lightweight Directory Access Protocol (LDAP) directories and structured query language (SQL) databases, IBM Rational® Application Developer software provides easy-to-use wizards. The application programming interfaces (APIs) and other interfaces required for defining a resource collection to the personalization engine are all public. The APIs are documented in the personalization Javadoc™, which is available with the product install under <wps>/doc/Javadoc/Personalization.

The Federated Documents resource collection

The Federated Documents resource collection extends the WebSphere Portal Personalization component with a new resource collection, which allows retrieving document metadata for documents stored on remote IBM Lotus Quickr, IBM Content Manager or IBM FileNet servers. The retrieved metadata can be rendered (i.e., transferred into any kind of HTML markup) for end users using Lotus Web Content Management personalization components. In addition to rendering the metadata of those documents, it is also possible to render links for downloading and for opening those documents in the context of the hosting content management system.

From a network communication perspective, the document metadata is acquired from the remote systems using the Document Services Atom feed as supported by IBM Lotus Quickr, IBM Content Manager and IBM FileNet servers. Since the Document Services feed is based on the Atom format (<http://tools.ietf.org/html/rfc4287>), the Federated Documents resource collection is not limited to consume feeds exposed by the said content management systems but can consume all kinds of data provided by any other Atom feed.

Federated Documents selection rules

A selection rule for the Federated Documents resource collection can be defined using the Personalization Editor portlet. The single selection criteria supported for a selection rule operating on the federated documents resource collection in WebSphere Portal 6.1.5 software is the specific document folder on the remote system. A corresponding selection result consists of all document metadata of documents contained in that folder. The folder itself is identified by its corresponding document services feed URL pointing to the Atom feed of document metadata for those documents. It is not required, though still possible, to manually enter the Atom feed URL for the folder of choice, but the document picker widget can be used to select the folder in a file-manager style graphical user interface (GUI). In addition to the folder selection URL, the selection rule can be parameterized with the maximum number of document metadata entries to retrieve from the remote system.

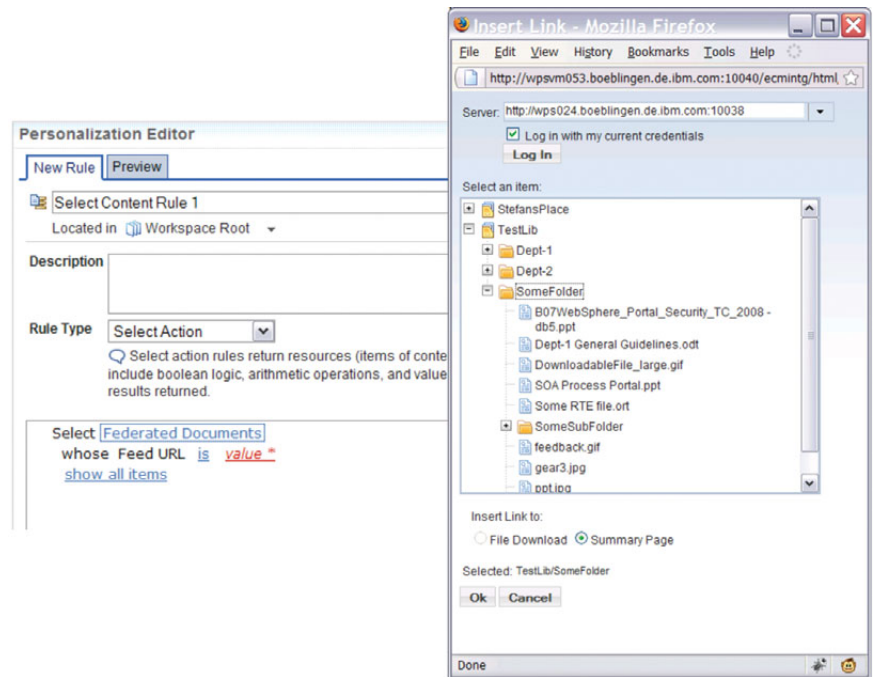


Figure 3. Personalization Editor and picker.

To select entries of some other Atom feed available on the network (e.g., http://www.flickr.com/services/feeds/photos_public.gne?tags=IBM&format=atom pointing to the feed of current Flickr photos tagged with “IBM”), the Atom feed URL can be directly pasted into the “Feed URL” text field of the Personalization Editor portlet.

Like every other selection rule, the selection rules operating on the federated documents resource collection can be combined with personalization profiler and binding rules to select different folders based on the request context (e.g., based on user attributes or session context). This allows targeting specific sets of documents to individual groups of users based on their business needs or personal preference.

Rendering federated documents data

The individual metadata exposed by the resources contained in a federated documents resource collection can be rendered using the Lotus Web Content Management personalization component. The Lotus Web Content Management personalization component allows specifying the HTML fragments that can be generated for the individual resources contained in the selection rule selection result.

A Lotus Web Content Management personalization component is primarily defined by a specific personalization selection rule and a design section defining the specific HTML fragments that can be generated for individual resources selected by the given selection rule. The personalization rule can be either embedded into the personalization component or consist of a reference to a corresponding selection rule in the personalization repository. The individual resource metadata values exposed by a given resource instance at hand can be added to the generated HTML markup by added corresponding `<AttributeResource>` tags. The “attributeName” attribute of this tag can be used to identify a specific metadata value of a given resource.

Table 1. The following “attributeName” values are supported for the federated documents resource collection.

| attributeName | Atom feed mapping | Federated document mapping |
|---------------|---|--|
| id | //atom:entry/atom:id | Prints out the unique ID of the document. |
| title | //atom:entry/atom:title | Prints out the title of the document. The actual result depends on the corresponding attribute mapping that needs to exist at the remote content management system. If no such mapping exists, the file name will be printed. |
| authors | //atom:entry/atom:author/ atom:name | Prints out the names of the authors of the given document. The actual result depends on the corresponding attribute mapping that needs to exist at the remote content management system. If no such mapping exists, an empty string will be printed. |
| published | //atom:entry/atom:published | Indicates the point in time of the first availability of the given document. The actual result depends on the corresponding attribute mapping that exists at the remote content management system. If no such mapping exists, an empty string will be printed. |
| updated | //atom:entry/atom:updated | Indicates the point in time of the last update to the given document. The actual result depends on the corresponding attribute mapping that exists at the remote content management system. If no such mapping exists, an empty string will be printed. |
| contentType | //atom:entry/atom:link [@rel="enclosure"]/@type | Prints out the Multipurpose Internet Mail Extensions (MIME) type of the given document. If this information is not served by the remote content management system, an empty string will be printed. |
| contentLink | //entry/link[@rel="enclosure"]/@href (turned into an absolute URL) | Prints out the absolute URL that can be used to download the given document. |
| viewLink | //atom:entry/atom:link [@rel="alternate"]/@href (turned into an absolute URL) | Prints out the absolute URL that can be used to open the given document in context of the remote content management user interface. If no such URL is returned by the remote content management system, an empty string will be printed. |
| size | //atom:entry/atom:link [@rel="enclosure"]/@length | Prints out the size of the document in bytes. If this information is not served by the remote content management system, an empty string will be printed. |

Example: The following text shows a very simple federated documents design using HTML table row elements to render the document metadata:

Header:

```
<table border=1>
<th>Title</th>
<th>Author</th>
<th>Download</th>
<th>Edit</th>
<th>Type</th>
<th>Size</th>
<th>Updated</th>
<th>Published</th>
```

Design for each menu search result:

```
<tr>
<td><b><AttributeResource attributeName="title"/></b></td>
<td><AttributeResource attributeName="authors" separator=","/></td>
<td><a target="_blank" href="<AttributeResource
attributeName="contentLink"/>">download</a></td>
<td><a target="_blank" href="<AttributeResource
attributeName="alternateLink"/>">open</a></td>
<td><AttributeResource attributeName="contentType"/></td>
<td><AttributeResource attributeName="size"/></td>
<td><AttributeResource attributeName="updated"/></td>
<td><AttributeResource attributeName="published"/></td>
</tr>
```

Footer:

```
</table>
```

More sophisticated designs can be used to generate a more appealing user experience. The design itself is often independent of the selection rule and may be reused across personalization components by defining a corresponding HTML component and referencing it from multiple locations. The following figure 4 shows the rendering result of the simple design given above, a somewhat nicer alternative design for the same data and a further design for another generic Atom feed (using the Flickr feed URL mentioned earlier).

| Title | Author | Download | Edit | Type | Size | Updated | Published |
|--|----------|--------------------------|----------------------|------------|---------|-----------------|------------------|
| DownloadableFile_large.gif | xyzadmin | download | open | image/gif | 549 | 5/6/09 9:25 PM | 5/6/09 9:25 PM |
| SOA Process Portal | | | | powerpoint | 2746368 | 4/20/09 1:13 PM | 2/26/09 2:48 PM |
| Dept-1 General Guidelines.odt | | | | | | | 1/20/09 1:05 PM |
| ppt.jpg | | | | | | | 1/9/09 10:21 AM |
| processpic.jpg | | | | | | | 1/11/09 11:23 AM |
| feedback.gif | | | | | | | 1/3/09 4:27 PM |
| B07WebSphere_Portal_Security_TC_2008 - dt5.ppt | | | | | | | 1/26/09 2:44 PM |
| Some RTE file.ort | | | | | | | 1/3/09 3:57 PM |
| gear3.jpg | | | | | | | 1/26/09 2:49 PM |

Shared Docs

Shared Documents

- [DownloadableFile_large.of](#)
Updated: 5/6/09 9:25 PM. Authors: xyzadn
- [SOA Process Portal](#)
Updated: 4/20/09 1:13 PM. Authors: xyzad
- [Dept-1 General Guidelines.odf](#)
Updated: 4/20/09 1:05 PM. Authors: xyzad
- [ppt.jpg](#)
Updated: 4/9/09 10:21 AM. Authors: xyzad
- [processpic.jpg](#)
Updated: 3/11/09 11:23 AM. Authors: xyzad
- [feedback.of](#)
Updated: 3/3/09 4:27 PM. Authors: Dieter
- [B07WebSphere_Portal_Security_TC_20](#)
Updated: 3/2/09 4:24 PM. Authors: xyzadn
- [Some RTE file.ort](#)
Updated: 3/3/09 3:59 PM. Authors: xyzadn
- [gear3.jpg](#)
Updated: 2/26/09 2:49 PM. Authors: xyzad

Current photos tagged with IBM

Current IBM Photos at flickr

- [IBM7337](#)
Uploaded By: Michael A. Lowry Tags: switzerland,ibm,zurich,hobbyclub,zf
- [IBM7793](#)
Uploaded By: Michael A. Lowry Tags: switzerland,ibm,zurich,hobbyclub,zf
- [IBM7659](#)
Uploaded By: Michael A. Lowry Tags: switzerland,ibm,zurich,hobbyclub,zf
- [IBM7692](#)
Uploaded By: Michael A. Lowry Tags: switzerland,ibm,zurich,hobbyclub,zf
- [IBM7794](#)
Uploaded By: Michael A. Lowry Tags: switzerland,ibm,zurich,hobbyclub,zf
- [IBM7772](#)
Uploaded By: Michael A. Lowry Tags: switzerland,ibm,zurich,hobbyclub,zf
- [IBM7573](#)
Uploaded By: Michael A. Lowry Tags: switzerland,ibm,zurich,hobbyclub,zf
- [IBM7768](#)
Uploaded By: Michael A. Lowry Tags: switzerland,ibm,zurich,hobbyclub,zf
- [IBM7669](#)
Uploaded By: Michael A. Lowry Tags: switzerland,ibm,zurich,hobbyclub,zf
- [IBM7777](#)
Uploaded By: Michael A. Lowry Tags: switzerland,ibm,zurich,hobbyclub,zf

Figure 4. Rendered content.

Security and configuration aspects

The actual remote access to the Atom feed data is done using the WebSphere Portal Asynchronous JavaScript™ and XML (AJAX) proxy component, which allows fine-grained control over the URLs used in selection rules and the HTTP cookies/headers forwarded with such requests. This allows propagating single sign-on credentials (like the WebSphere Lightweight Third Party Authentication [LTPA] token) with requests to document management servers located in the single sign-on domain of the portal while at the same time preventing those sensitive tokens from being sent to non-trusted external sites.

In addition to security configuration, it is furthermore possible to configure specific cache lifetime and cache size values for the federated documents caches.

Web Content Integrator

Prior to Lotus Web Content Management 6.1.5 software, the WCI was available as a module for migrating content from external sources into Lotus Web Content Management software. IBM has invested significant resources to enable this as an out-of-the-box capability inside Lotus Web Content Management software. With WCI, any third-party content repository capable of creating feeds can publish content into Lotus Web Content Management software. A plug-in monitors those feeds and consumes the data as a series of Lotus Web Content Management content items, metadata, site areas and categories. When ingesting content from an ECM repository, the metadata is stored and mapped to the corresponding Lotus Web Content Management component, thus making it easy for a customer to leverage out-of-the-box capabilities to manage content or documents. Since WCI is native to Lotus Web Content Management software, customers have the flexibility to monitor content/documents from within Lotus Web Content Management software every time the ECM system actively triggers events to publish new feeds to Lotus Web Content Management software.

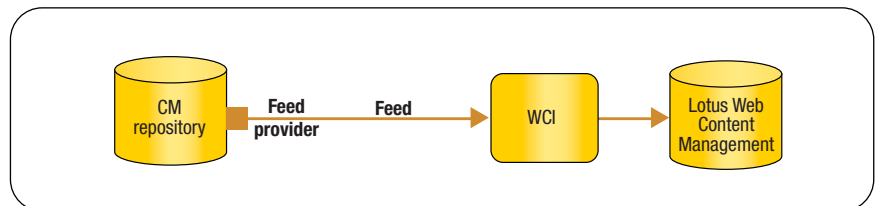


Figure 5. Using WCI to ingest content from ECM repositories.

As shown in figure 5, using WCI, a content feed for ECM systems (such as FileNet and Content Manager software) makes content items become native Web content objects within Lotus Web Content Management software. Workflow, access control, search and personalization can be applied as if the content were entered directly through Lotus Web Content Management software. The content can be linked just like any other piece of Web content and can be configured to display the content via Lotus Web Content Management software. Users already familiar with Lotus Web Content Management software will not need to learn new components, tools or integration patterns. Integration is handled seamlessly on the back end with no disruption to ECM content authors or Lotus Web Content Management site designers. The structure and placement of content within Lotus Web Content Management software are handled as a function of where and how the feeds are generated, minimizing direct user involvement.

New users to Lotus Web Content Management software will find ECM content as native Web content. For an ECM user, content is automatically updated into the Lotus Web Content Management system through feeds when contributed, modified, moved, recategorized or deleted in an ECM system. Clients should consider integrating their ECM systems with Lotus Web Content Management software via WCI in situations when the content or documents have already gone through significant review and policy management in the respective collaborative or ECM repository and are now fully functional and available to be published on the Web using Lotus Web Content Management or WebSphere Portal software.

In-line content editing in Lotus Web Content Management software and robust URL addressability in FileNet and other ECM software makes it easy for content owners to link directly to the source of the original content in the FileNet application. For other ECM systems, JavaServer™ Pages (JSP™) components with embedded API calls can be used to address in-line content editing.

Writing a custom resource and resource collection

WebSphere Portal 6.1.5 software includes a new Federated Document resource collection that can satisfy most of your needs. If not, or if you need to integrate with another content system that does not provide a REST API, you can write your own personalization resource collection as follows.

The resource implementation is the class that represents a row, object or item in the data store. In the case of the JavaBeans™ API for Content Manager software, the resource implementation may simply wrap a `com.ibm.mm.beans.CMBItem`. A `CMBItem` in turn represents a document, folder or row in Content Manager software.

A read method, or *getter*, can be exposed on the resource implementation for each property in the schema of the `CMBItem` if the schema is known when creating the resource collection. A generic Content Manager or FileNet resource collection, which is capable of dynamically handling practically any type of resource collection, is also possible. However, its implementation is significantly more complex.

The ResourceDomain3 interface is used to implement a personalization resource collection to connect to Content Manager software. The findResourcesByQuery method is the most important method of the ResourceDomain3 implementation. This method takes the query object from the rules engine and transforms it into a query string, which the data store can understand, such as SQL, LDAP filter and XPath. The findResourcesByQuery method then must run the query. The result for Content Manager software will be an iterator of objects that contains CMBItems. The CMBItems must be extracted from the result set and wrapped in a resource; then the list of resources must be returned by the findResourcesByQuery method. In short, the ResourceDomain3 is responsible for taking the query object and returning a list of resources that match that query.

The ResourceManager3 implementation is used for writing to or updating data accessed by a resource collection. Since reusing data already in FileNet or Content Manager software is a goal, the implementations of these methods should throw exceptions indicating that write operations are not supported.

A FileNet or Content Manager resource collection offers some of the most flexible integration options. Once the resource collection is built, business users and content authors can write rules to decide which FileNet or Content Manager items appear in certain locations on the page. The rules can respond to the page, user, user actions, time, date or any number of criteria. In this way, a single rule can control content on many pages.

A personalization resource collection can be created once and reused in many places through the use of rules. As personalization improves, end-user satisfaction with the Web site also improves, with little or no extra cost.

This integration approach requires writing code. The complexity of this code depends on the complexity of the data model and the complexity of the queries that need to be supported. For example, a `findResourcesByQuery` implementation handles all queries; however, if you know that you will only write and implement certain rules, it is much easier to detect and handle these queries individually.

Integration using JSP components

The Lotus Web Content Management JSP component can be used to add custom code to a presentation template. The component allows JSP technology to be embedded in a Lotus Web Content Management page that can call out to any Java™ API. Calls can be made to WebSphere Information Integrator Content Edition software to access Content Manager items or FileNet objects. The JSP can also directly call FileNet or Content Manager APIs.

Because this is an API-based approach, it's a very flexible method of integration. In addition, JSP components may be written to respond to the context or page automatically. This form of integration requires FileNet or Content Manager skills, with minimal Lotus Web Content Management or portal skills necessary.

Section 7.5.4 of the IBM Redbooks® publication, IBM Workplace Web Content Management Java Edition V2 and IBM DB2® Content Manager V8 Working Together (<http://www.redbooks.ibm.com/abstracts/sg246368.html>), includes sample JSP code for this pattern of integration.

The JSP can call the Lotus Web Content Management API to access information from the current piece of Web content. The retrieved FileNet or Content Manager items can vary depending on the values that were set when the content was created.

For example, an authoring template can include a field for metadata such as region. The JSP can get the value of the region for the current piece of content and query FileNet or Content Manager software for items with an attribute matching that region code. The JSP then displays all the items with that region code.

In this scenario, a region code would be entered when contributing a piece of content, and then the presentation template would automatically use that region code to display appropriate documents from the ECM system along with the content.

The creator of the authoring template can improve the usability for the content author by using custom JSP code for the content path field or by using an optional selection field to constrain the content creator's choices. A common application for this is a drop-down list of preset values.

JSP components can also be built to handle content changes that can subsequently be submitted back to FileNet or Content Manager software. You can precede a managed Web front end with business processes driven from FileNet or Content Manager software.

Using field-level JSP components for content authoring

Lotus Web Content Management software can also use custom JSP components to override individual fields on an authoring template. These field-level JSP components control the rendering of a field when a piece of content is being contributed. When a content contributor adds new content, this can be in the form of new information, or information selected from a back-end content management system. This all happens within the Lotus Web Content Management user interface (UI). The JSP code that controls this data entry can also be written to allow content to be written back to the ECM system.

For example, when a content author uses a field in the Lotus Web Content Management UI to select a piece of content, this content can be stored either in the Lotus Web Content Management application or in an external content management system source. The template designer can design the JSP component to a variety of external items, such as a selection of folders, individual items or queries. When the content is rendered on a Web page, the JSP component gets the value the author stored in the text element by means of your custom JSP component. The JSP component then uses this value to determine which elements to display or use from the other content management system.

Example JSP for content authoring

When designing a Lotus Web Content Management site, the authoring template or JSP designer decides which documents should be available for selection and how they should be presented to the content contributors.

For example, figure 6 shows an example JSP component that references an existing FileNet document. There is also a button to create a new FileNet document. In this example, the JSP component is set to select documents that match a particular query and present them in a drop-down selection field.

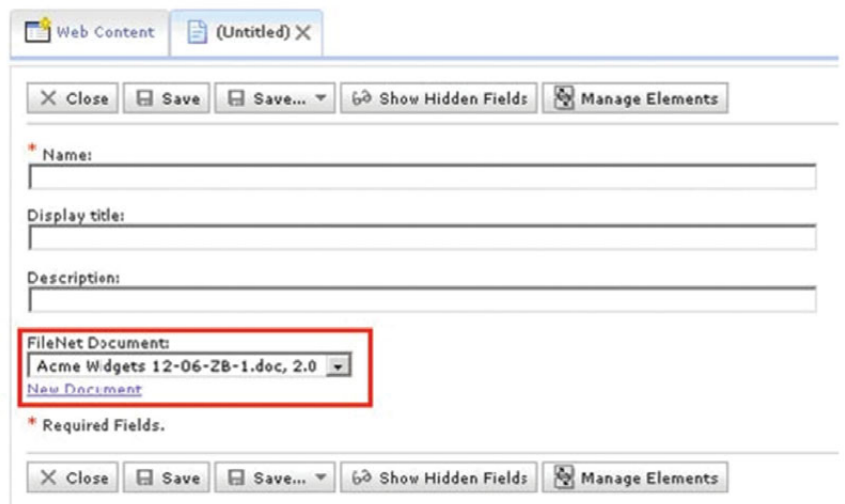


Figure 6. A JSP component references an existing FileNet document. The area enclosed in the red box is rendered by custom JSP code.

The content contributor can select an existing document or use the new button to upload a document to the FileNet software. The new button opens a page in the FileNet workplace to upload a document. The FileNet workplace UI can be customized with a template and custom styles. An example page is depicted in figure 7.

Figure 7. Contributing a document to the FileNet workplace.

When the user is finished, the document is uploaded to the FileNet software, and the previous Lotus Web Content Management page is reopened and refreshed with the new document's name appearing in the selection field, as seen in figure 8.



Figure 8. A new document is uploaded to the FileNet software and automatically selected in Lotus Web Content Management software.

This approach is used to build Web pages like those depicted in figure 8. The federated content component builds in a similar level of integration for Content Manager software. Federated content components only work with Content Manager content; custom JSP components can apply to any content management system with a Java API. Custom JSP components also allow more flexibility in how the content selection process is presented to the content contributor.

This approach can be used to contribute content directly to another content management system when a URL-addressable Web client is available for the content management system. An example of this is the new document button shown earlier in figure 6.

Integrated ECM and WCM

Integrating your ECM and WCM systems allows you to quickly leverage enterprise assets in building your websites. Content can be managed and workflowed from an ECM repository like FileNet P8 or Content Manager. Powerful capabilities are built into WCM to allow you to do this out of the box.

| Player | Bandwidth |
|----------------------|-----------|
| Windows Media Player | Low High |

Figure 9. A document and text from the FileNet application are displayed in a Lotus Web Content Management content item.

Integration using the Lotus Web Content Management API

Lotus Web Content Management software also provides an API that includes read and write functions for content and read methods for many other elements in the software. The Lotus Web Content Management Javadoc is provided with the install at <http://<host>:<port>/wps/wcm/webinterface/api-java doc/index.html>. Calls to the Lotus Web Content Management API may be triggered by events in FileNet or Content Manager software, including FileNet and Content Manager workflow actions. Using this API, users can automatically create content in Lotus Web Content Management software when a FileNet or Content Manager item reaches a certain workflow stage.

Using the Lotus Web Content Management API from within a FileNet or Content Manager workflow also allows entire Lotus Web Content Management sites and site areas to be created in response to new FileNet or Content Manager items. For instance, if a folder is added to FileNet or Content Manager software, a new site area is created in Lotus Web Content Management software.

However, the Lotus Web Content Management API requires more extensive custom code than other integration approaches.

Which scenario should you choose?

Which scenario is best for each situation? It all depends on the system and what needs to be accomplished. Table 2 compares some of the more common factors that need to be considered when integrating Lotus Web Content Management software with an existing ECM system such as, but not limited to, FileNet and Content Manager software. Depending on your detailed requirements, the time and effort required to implement each scenario may differ. Hence, please take that into consideration before implementing one option over the other.

Table 2. Applicable scenarios for integrating IBM Lotus Web Content Management software with an ECM system

| | Personalization resource collection | Feed consumption from ECM repositories | Web Content Integrator (WCI) | Integration using JSP components | Lotus Web Content Management API |
|--|-------------------------------------|--|------------------------------|----------------------------------|----------------------------------|
| Reference content item or lists of items | ■ | ■ | ■ | ■ | ■ |
| Support personalization and selection queries | ■ | ■ | ■ | ■ | ■ |
| Use Lotus Web Content Management workflow and security | | | ■ | ■ | ■ |
| Use Lotus Web Content Management presentation tags on ECM content | ■ | ■ | ■ | | |
| Allow ECM items to be edited or contributed from WebSphere Portal software | | | ■ | | |

Choosing the right approach—or combination of approaches—depends upon the requirements of each project. For instance, it is possible to use the feed resource collection to select items from FileNet or Content Manager software and use the JSP component to display the item once it's selected, allowing content contribution within the portal and complex selection. A JSP component combined with the WCI allows in-line editing and seamless publication of ECM content.

Why IBM?

IBM Lotus Web Content Management software provides options for integrating with existing enterprise content management and collaborative document management solutions in order to help optimize your software investment, reduce the need to change your back-end content storage methods and avoid forcing your employees to learn a new application. This flexibility—combined with other features of Lotus Web Content Management software, including personalized dynamic delivery of content, social software capabilities such as blogs and wikis, Web Content Integrator for aggregating content, the separation of content and presentation, and easy-to-use rich text editing capabilities that shift the focus of authoring in the hands of business users—distinguishes IBM Lotus Web Content Management software in the Web content management marketplace.

Accelerate the time to value of your software investment

IBM Software Services for Lotus and select IBM Business Partners can help you better understand your technology options and how to leverage Lotus collaboration solutions to help you lower your IT total cost of ownership and increase your organization's productivity. Find out more about the technical consulting, training and Software Accelerated Value Program services available to help you accelerate your success with IBM technology. For more information, go to ibm.com/software/lotus/services.

For more information

To learn more about the IBM Lotus Web Content Management solution and how it can integrate with your existing systems, please contact your local IBM representative or IBM Business Partner, or visit:

ibm.com/software/lotus/products/webcontentmanagement

About the authors

Dr. Dieter Buehler works as an architect within the IBM WebSphere Portal development team in the IBM Laboratory in Boeblingen, Germany. He drove the Portal Security and Business Process Integration architecture until 2008, when Dr. Buehler joined the Lotus Web Content Management team.

Krishnan Hariharan is the offering manager for IBM Lotus Web Content Management software and leads the strategic direction, positioning and go-to-market strategies for the product. As a member of the product management team, Krishnan also helps drive the product road map and integrations with other products in the IBM portfolio.

Stefan Hepper is part of the IBM WebSphere Portal and Lotus Web Content Management System team where he is the responsible architect for the Lotus Web Content Management and WebSphere Portal convergence and Lotus Web Content Management content delivery. Previously he led the Java Portlet Specifications JSR 168 and JSR 286 and has worked in the portal area since 2001. Stefan is based in the IBM Silicon Valley Lab in California.

Steve Pritko is currently an architect of IBM Lotus Web Content Management and IBM WebSphere Portal software, focusing on integration and deployment of Web content solutions. He has more than five years of experience working on Web content management technologies.



© Copyright IBM Corporation 2009

Lotus Software
IBM Software Group
One Rogers Street
Cambridge, MA 02142
U.S.A.

Produced in the United States of America
November 2009
All Rights Reserved

IBM, the IBM logo, ibm.com, Lotus, and WebSphere are registered trademarks of International Business Machines Corporation in the United States, other countries or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at ibm.com/legal/copytrade.shtml

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates.

The information contained in this documentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, it is provided "as is" without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software. IBM customers are responsible for ensuring their own compliance with legal requirements. It is the customer's sole responsibility to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws.



Recyclable, please recycle

LOW14013-USEN-00