

Alfresco 2.1

Web Content Management

Product Evaluation Guide

Version 1.0



Copyright (c) 2007 by Alfresco and others.

Information in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Alfresco. The trademarks, service marks, logos or other intellectual property rights of Alfresco and others used in this documentation ("Trademarks") are the property of Alfresco and their respective owners. The furnishing of this document does not give you license to these patents, trademarks, copyrights or other intellectual property except as expressly provided in any written agreement from Alfresco.

The United States export control laws and regulations, including the Export Administration

Regulations of the U.S. Department of Commerce, and other applicable laws and regulations apply to this documentation which prohibit the export or re-export of content, products, services, and technology to certain countries and persons. You agree to comply with all export laws, regulations and restrictions of the United States and any foreign agency or authority and assume sole responsibility for any such unauthorized exportation.

If you need technical support for this product, contact Customer Support by email at support@alfresco.com. If you have comments or suggestions about this documentation, contact us at documentation@alfresco.com.

This edition applies to version 2.1 of the licensed program.

Contents

Introduction	1
Important Notes	1
Typographic Conventions	2
Web Content Management Product Overview	3
XML Content Authoring	3
Multi-channel Content Publishing	4
Sandboxed Content Development	4
Virtualization and In-context Preview	4
Content Staging, Promotion, Workflow, and Snapshots	5
Content Deployment	5
Link Validation	5
Content Launch and Expiration	5
Web Publishing Dashlets	6
Web Content Management Scenario Overview	7
Creating New Users	8
Creating New Web Forms	13
Creating New Web Projects	19
Importing the Sample Dynamic Website	28
Creating New Web Content: Company Profiles	33
Creating New Web Content: Press Releases	38
Content Launch and Expiration	42
Site Deployment	46
Snapshots and Rollback	48
Links Management	50
Links Management: User Sandboxes	52
Links Management: Workflow	55
Content Contribution: Alfresco Dashlets	59
Site Regeneration: Updating Templates	61
Virtual File System Interface	66
Summary	70

Introduction

The purpose of this Product Evaluation Guide is to:

- Introduce the basic concepts of Alfresco's Web Content Management solution
- Demonstrate how to set-up and configure Alfresco for managing a website
- Showcase a sample web publishing scenario
- Provide update on new Alfresco WCM 2.1 features and capabilities

Alfresco recommends end-users downloading the Alfresco 2.1 release walk through this guided tutorial and sample content to familiarize themselves with the Web Content Management extensions to Alfresco's 2.1 ECM platform offering.

Important Notes

Alfresco also recommends reviewing and keeping up-to-date on the latest around our WCM offering on our Developer Wiki. To bookmark this page for easy reference, please use the following link:

http://wiki.alfresco.com/wiki/New_Web_Content_Management_Plan

Typographic Conventions

The following typographic conventions are used in this Product Evaluation Guide:

Formatting Convention	Type of Information
Bold	Navigation, Menu, and Action links in the Alfresco web client
<i>Blue Italicized Bold</i>	Links in the sample website
<i>Italics</i>	Filenames and path references in the Alfresco sample website and source files. Also used for labels within Alfresco forms.
“Quoted Text”	Folder names within the Alfresco web client
Captioned Text	Important information to note

Web Content Management Product Overview

Alfresco Web Content Management (WCM) extends Alfresco 2.1's capabilities in the following areas:

- XML content authoring
- Multi-channel content publishing
- Sandboxed content development
- Virtualization and in-context preview of both static and dynamic web pages
- Content staging, promotion, workflow, and snapshots
- Links validation
- Content launch and expiration

XML Content Authoring

Alfresco WCM adds XForms support to enable Content Publishers to easily create reusable XML using a simple browser-based form. Alfresco enables Web Developers to upload and register XML Schemas (XSDs) within a central Form Library in the Alfresco Data Dictionary. Alfresco also enables Web Developers and Content Managers to subscribe individual Web Projects (websites or web applications) to use one or more forms from this library to create and edit content. Once configured for a given Web Project, Content Publishers can access a Web Form via the **Create New Content** Wizard and walk-through a step-by-step process for creating or editing XML assets.

Capabilities provided within a form include support for:

- WYSIWYG content authoring with customizable controls on a field-by-field basis
- Repeating elements, including single, compound, and nested elements, with defined minimum and maximum number of items
- Re-ordering and deletion of elements within a repeating list
- Enforced data types with pre-defined UI controls for data input
- Standard HTML data entry widgets, including checkboxes, drop-downs, radio-buttons, and multi-select lists
- File-browser control creating references other assets with support for constraint based on asset type (file or folder) and mimetype
- Mandatory / non-mandatory elements
- Read-only fields and default field values

- Client-side data validation
- Custom labels and alerts, including support for internationalized labels and alerts via resource bundles
- Multi-byte content entry
- Callouts for dynamic content selection options (used to generate conditional lists of values for users to select for radio buttons, single- or multi-select lists)
Form includes to reuse form definitions as components within other forms (for example, reused the form definition for an address across a multiple web forms)

Multi-channel Content Publishing

Alfresco WCM also adds support for enforcing rules to generate different output formats from created XML assets. Alfresco enables Web Developers to upload and associate one or multiple XSL, Freemarker, or XSLFO templates with an XSD and automatically enforce the application of one or more templates to generate content upon completion of the **Create New Content** wizard.

Sandboxed Content Development

To support creation and editing of web content, Alfresco provides support for a sandboxed development model. For any website or web application, Alfresco provides each user with their own special space – a sandbox – where they can make any number of changes to web content, including adding, editing, or deleting both files and folders. Alfresco tracks all content modifications made within a user's sandbox and maintains those changes in isolation from other user's working within their own sandbox environment. In this way, large teams of users can work independently on changes to the website without stepping over one another's work. Web Developers can modify source code, Web Designers can change site look-and-feel, and Content Publishers can create and modify content – all in parallel, in their own separate virtual staging environments, without risk of interfering with one another's changes.

Virtualization and In-context Preview

Core to sandboxes is the concept of virtualization and in-context preview. Virtualization means that each user in the context of their sandbox has a complete view of all current, approved, checked-in content along with those unique modifications made within the context of their sandbox. In-context preview means that when previewing any non-modified or modified asset in a sandbox, Alfresco provides a complete virtual view of the website as it would look if all changes in a sandbox were committed to the live site. In this manner, each user in the context of their sandbox can do rigorous and thorough QA for all changes they are posting to the website: Web Developers can ensure code changes to JSP, class files, or jars are working properly, Web Designers can test changes to CSS files, and Content Publishers can test that XML they are creating are properly generating HTML pages and being sourced in any number of dynamic web pages. Combined with sandboxed development, virtualization and in-context preview means that a large, diverse web team can easily collaborate on changes to the website with reduced risk and higher overall quality.

Content Staging, Promotion, Workflow, and Snapshots

When changes are QAed in the context of a sandbox, individual assets or entire change sets – a list of modified items in a sandbox – can be promoted from a sandbox to a Staging sandbox. Content is promoted from a user's sandbox to Staging by initiating the Submit Wizard. This wizard is initiated by clicking on either the **Submit** or **Submit All** actions in the Alfresco web client. Submitted changes are routed via workflow for one or multiple users to review and approve, either serially or in parallel using Alfresco's out-of-the-box sample editorial review workflow (or any custom-built WCM workflow). Once approved, a snapshot is automatically taken of Staging to provide an archive of the current version of the site. This snapshot is maintained over time to provide an audit trail and rollback point for previous versions of the site. Once a snapshot is taken, all committed changes are immediately reflected and available to each user in their own sandbox, enabling all users to consistently QA their potential changes against the latest and greatest version of the website.

Content Deployment

For known good snapshots of Staging, Alfresco 2.1 provides integrated deployment services to replicate incremental content changes to a remote file server, remote Alfresco server, or a combination. This integrated content deployment service also provides support for deploying a single web project to multiple remote servers for maximum run-time scalability. This service provides support for tracking which Staging snapshot is currently live, and provides audit trails for what content was recently replicated when a new site version is pushed.

Link Validation

When changes are committed to Staging, a master list of references for all assets is built, denoting references that are valid or broken. Within a sandbox, users can check links within a given website or web application, and use the Alfresco GUI to interactively correct and re-test links. Also, when committing new or modified assets to Staging (including both file and directory deletions), both end-users and reviewers in the content approval chain can view the incremental references that would be either corrected or broken were a change set to be approved and committed to the main site. This incremental links check enables users to more quickly and more effectively identify potential problems to ensure quality and site integrity on any site update.

Content Launch and Expiration

When promoting content to Staging, users can optionally set a specific launch and expiration date on the change set they are promoting. When a launch date is set, approved content changes are maintained in their own separate workflow sandbox until the specified time and date. Prior to that date, end-users can preview the future state of the website, immediately promote, or cancel the pending launch. As an example, members of a marketing team could use this facility to pre-generate a new version of the site to launch on Monday morning at 8am, getting all their changes finalized, approved, and staged prior to release on Friday the week before. In addition, for any changes promoted to the site, specific expiration dates can be set on a global or asset-by-asset basis. Upon expiration, end-users are automatically assigned the asset as a task so that they can make a determination whether the asset should be updated or removed from the site. This content expiration also works in conjunction with Alfresco's new Links Validation service, ensuring that the site's link integrity is incrementally tested against the new pending update. In this manner, content can not only be regularly refreshed through an automated, task-driven review process, but also be tested upon change to so the no errors on accidentally introduced in the form of broken links.

Web Publishing Dashlets

As an ECM platform, many knowledge workers using Alfresco do so primarily to create and collaborate on business documents. For these business users – Content Contributors – Alfresco 2.1 makes it easy to create and manage web content without having to expose the complexity of the web development environment for occasional contributors. In 2.1, Alfresco provides two new dashboard components, *My Web Forms* and *My Web Files* that put the needed set of publishing operations at the finger-tips of any knowledge worker, helping them to access and speed through the publishing process while masking the underlying process and structure of the Web Content Management system.

Web Content Management Scenario Overview

This Product Evaluation Guide will walk you through a basic scenario for importing and configuring a basic website in Alfresco and updating it a set of published changes. In this scenario, we will showcase how four different types of users will interact with the system:

Web Developers, responsible for:

- Creating new users
- Creating new Web Forms in the Alfresco Web Forms Library
- Creating and associate templates with Web Forms for multi-channel output
- Developing and maintaining application code using standard development tools against Alfresco's virtual file-system interface (CIFS)

Content Managers, responsible for:

- Creating new Web Projects, including:
 1. Setting up a basic Staging and Promotion model
 2. Configuring Web Forms and templates for web publishing
 3. Configuring workflow for editorial review of site changes
- Reviewing and approving content from the Alfresco Dashboard
- Rolling back published website changes by reverting or deploying older site snapshots

Content Publishers, responsible for:

- Creating and updating re-usable XML assets using browser-based forms
- Automatically generating web pages from multiple XML assets
- Promoting changes to the website and submitting for editorial review
- Staging and scheduling content releases for future publication
- Reviewing expired content for update or removal from the site
- Maintaining link integrity of a site for any promoted set of changes
-

Content Contributors, responsible for:

- Creating new documents and automatically promote PDF renditions to the website
- Creating and updating re-usable XML assets using browser-based forms

Content Reviewers, responsible for:

- Reviewing and approving content from the Alfresco dashboard
- Checking quality, correctness, and integrity of the site prior to content approval

Creating New Users

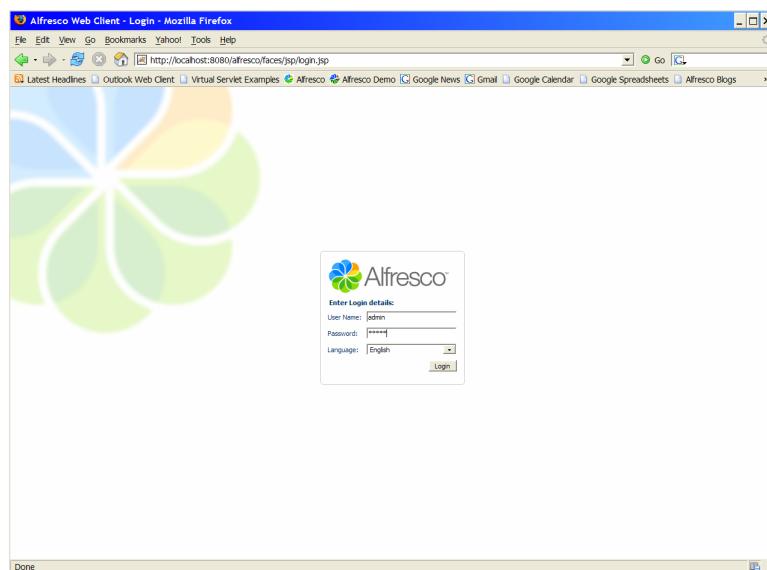
To get started, we will first need to log in to Alfresco and create the four users we will need to walk through this tutorial. We will use the following users for our scenario:

1. *Mark, Web Marketing Manager.* Mark will play the role of Content Manager. Mark will be responsible for reviewing and approving all changes to the website. Mark will also be responsible for managing the new Web Project we create, including adding and removing users and configuring Web Forms, templates, and workflow.
2. *Alice, Product Marketing Manager.* Alice will play the role of Content Publisher. Alice will be responsible for creating and editing new content for the website, including scheduling major new updates of the site for future site releases.
3. *Pat, Product Manager.* Pat will play the role of Content Contributor. Pat is primarily responsible for creating and editing business documents, typically as part of a collaboration project, that are automatically promoted upon completion to one or multiple websites. Pat is also an occasional contributor of web content directly to the site; unlike Alice our Product Marketing Manager, however, Pat needs simple entry points in web publishing that make the needs for specialized skills or training unnecessary.
4. *Lee, Legal Counsel.* Lee will play the role of Content Reviewer. Lee will be responsible for reviewing and approving select changes to the website only.

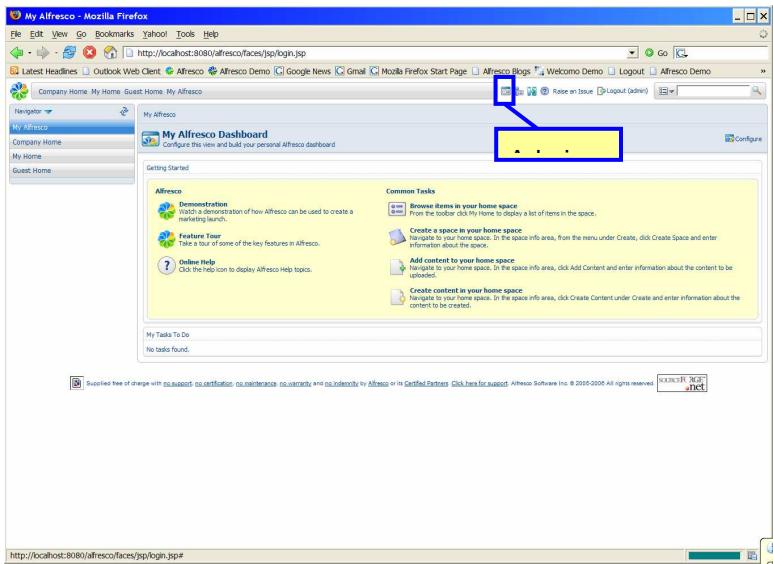
A fifth user, the Admin user, is the default system user. We will use the Admin user in the role of the Web Developer to create new user accounts, create our Web Forms and associated templates, and set-up our first Web Project on behalf of our Content Manager, Mark.

To begin in our capacity as a Web Developer, access Alfresco by going to the following URL and logging with the following information:

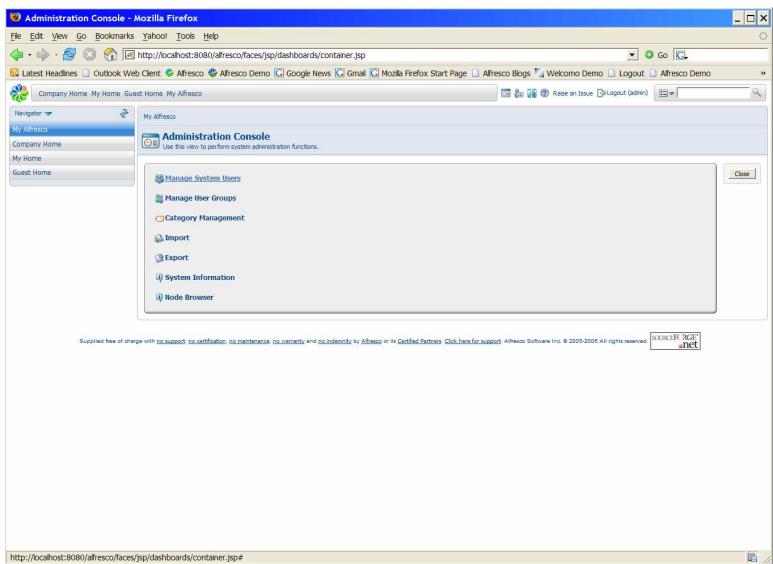
URL:	http://localhost:8080/alfresco
User Name:	admin
Password:	admin
Language:	English



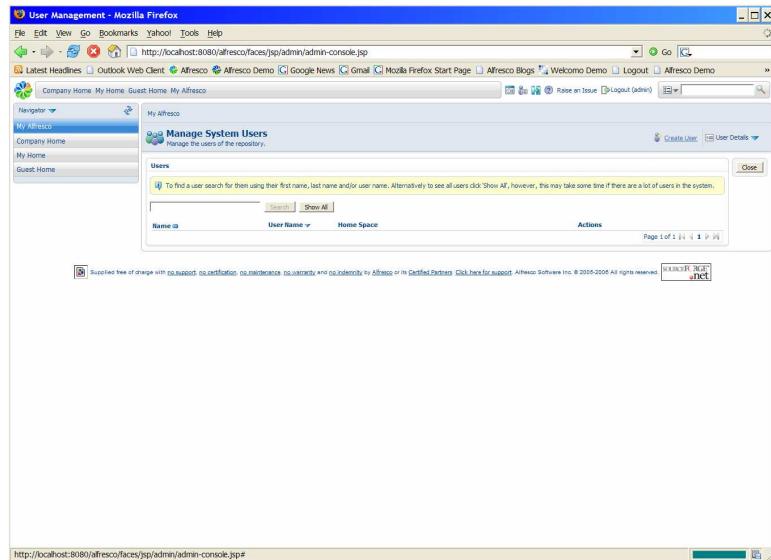
Click on the Administration Console icon.



Click on Manage System Users.



Click on Create User.



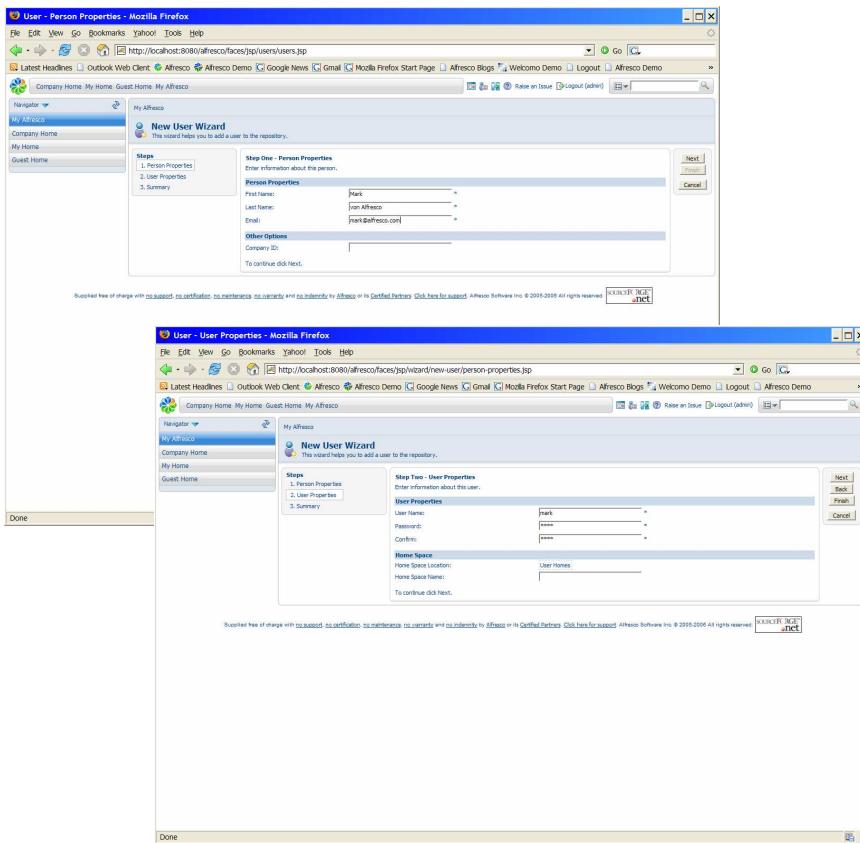
Type in details for our first user Mark, our Web Marketing Manager. In Step One of the Create User Wizard, enter the following:

User #1:

First Name:	Mark
Last Name:	von Alfresco
Email:	mark@alfrescodemo.org

Click **Next** when done. In Step Two of the **Create User Wizard**, enter the following additional data:

User Name:	mark
Password:	demo



Now that we have created Mark, our Web Marketing Manager, repeat this process three more times to create Alice, our Product Marketing Manager, Pat, our Product Manager, and Lee, our Legal Counsel. Use the following information:

User #2:

First Name:	Alice
Last Name:	von Alfresco
Email:	alice@alfrescodemo.org
User Name:	alice
Password:	demo

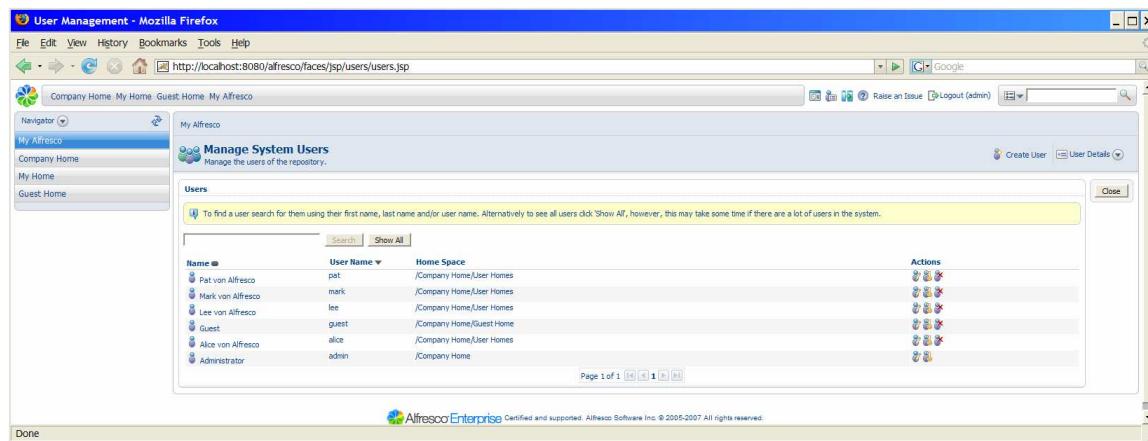
User #3:

First Name:	Pat
Last Name:	von Alfresco
Email:	pat@alfrescodemo.org
User Name:	pat
Password:	demo

User #4:

First Name: Lee
 Last Name: von Alfresco
 Email: lee@alfrescodemo.org
 User Name: lee
 Password: demo

Once our four users have been created, you can click on the **Show All** button in the **Manage System Users** console to confirm our newly created accounts. You should see the following when complete:



The screenshot shows a Mozilla Firefox browser window displaying the 'Manage System Users' page from the Alfresco application. The URL in the address bar is <http://localhost:8080/alfresco/faces/jsp/users/users.jsp>. The page title is 'User Management - Mozilla Firefox'. On the left, there's a sidebar with links like 'My Alfresco', 'Company Home', 'My Home', and 'Guest Home'. The main content area is titled 'Manage System Users' with the sub-instruction 'Manage the users of the repository.' Below this is a search bar with the placeholder 'To find a user search for them using their first name, last name and/or user name. Alternatively to see all users click "Show All", however, this may take some time if there are a lot of users in the system.' A 'Search' button is next to it, and a 'Show All' button is highlighted in yellow. The main table lists users with columns: 'Name', 'User Name', and 'Home Space'. The data is as follows:

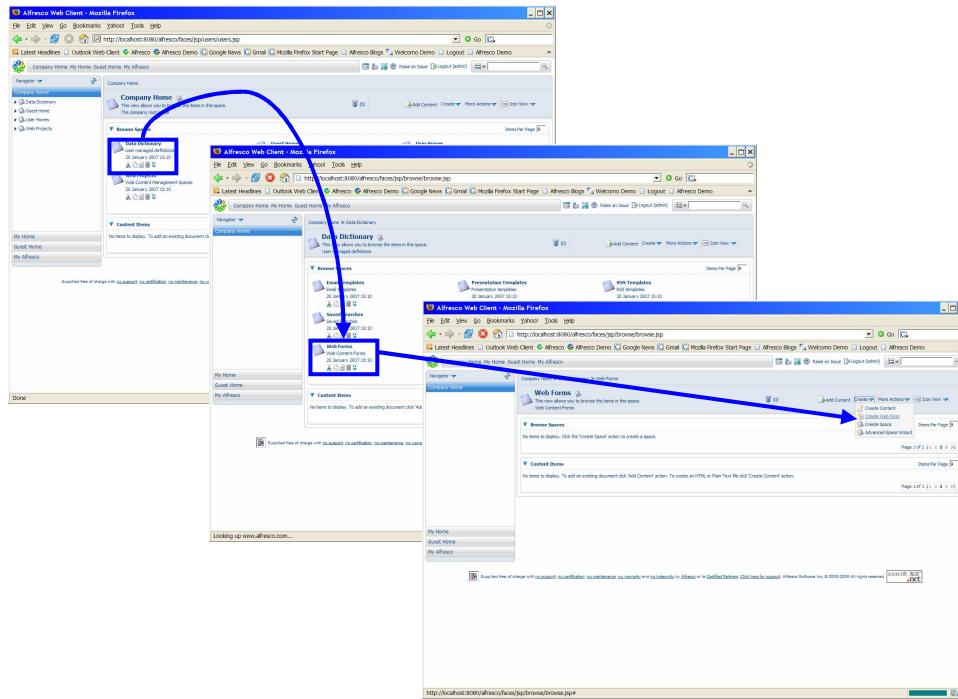
Name	User Name	Home Space	Actions
Pat von Alfresco	pat	/Company Home/User Homes	
Mark von Alfresco	mark	/Company Home/User Homes	
Lee von Alfresco	lee	/Company Home/User Homes	
Guest	guest	/Company Home/Guest Home	
Alice von Alfresco	alice	/Company Home/User Homes	
Administrator	admin	/Company Home	

At the bottom of the table, it says 'Page 1 of 1' with navigation icons. The footer of the page includes the Alfresco logo and the text 'Alfresco Enterprise Certified and supported. Alfresco Software Inc. © 2005-2007 All rights reserved.'

With our users now created, we will now move on to create the two Web Forms we will use for our first Web Project.

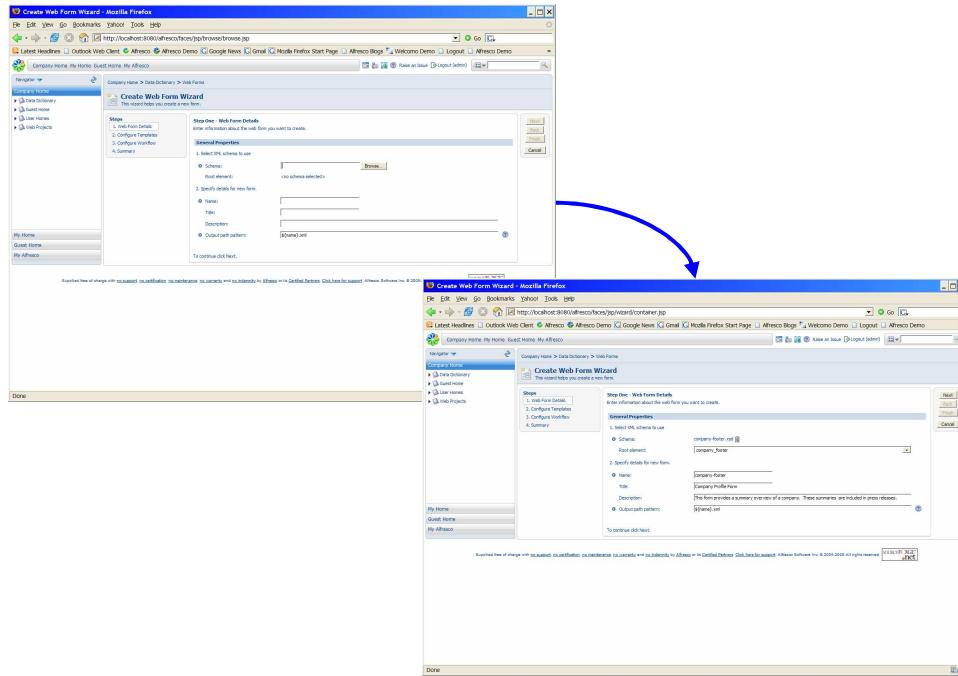
Creating New Web Forms

Click on the **Company Home** link in the upper-left hand corner of the Alfresco GUI. In the **Company Home** space, you should see four spaces: **Data Dictionary**, **Guest Home**, **User Homes**, and **Web Projects**. Click on **Data Dictionary**, and on the subsequent screen click on the **Web Forms** space. In the empty **Web Forms** space, select the **Create Web Form** action from the **Create** menu.



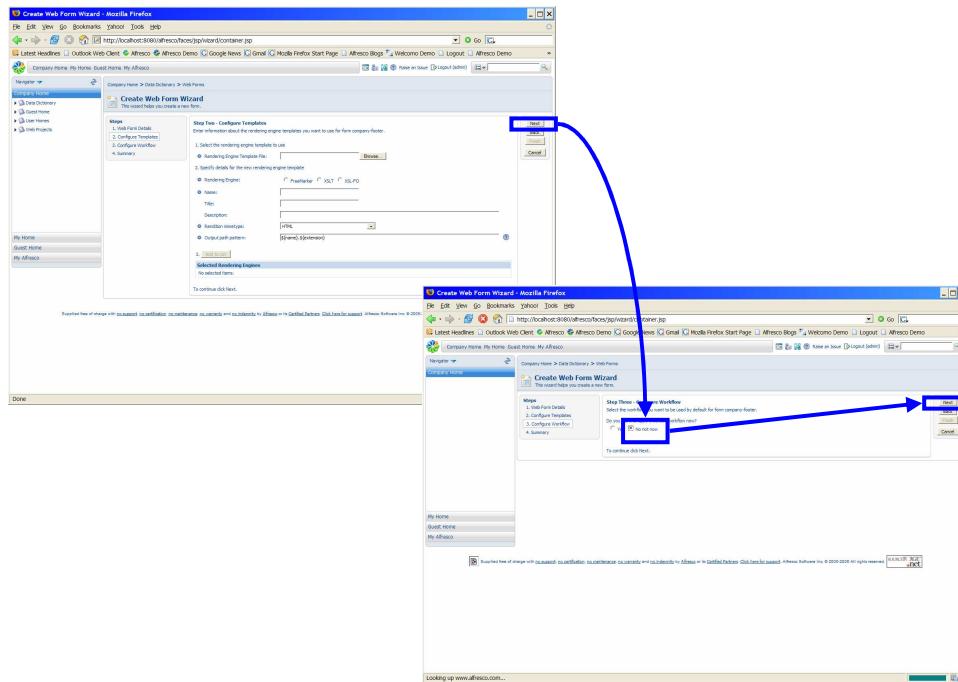
In the Step One of the **Create Web Form** Wizard, first click **Browse** to locate and upload an XSD from the Web Developer's local drive. Then, navigate to C:\Alfrescoextras\wcm\forms. Select the file *company-footer.xsd*. Upon returning to the wizard, note that the schema, *company-footer.xsd*, has been successfully uploaded. Now, fill in the following values for the rest of the form and then click **Next**.

Title: Company Profile Form
 Description: This form provides a summary overview of a company. These summaries are included in press releases.
 Output path: \${name}.xml (confirm default value)

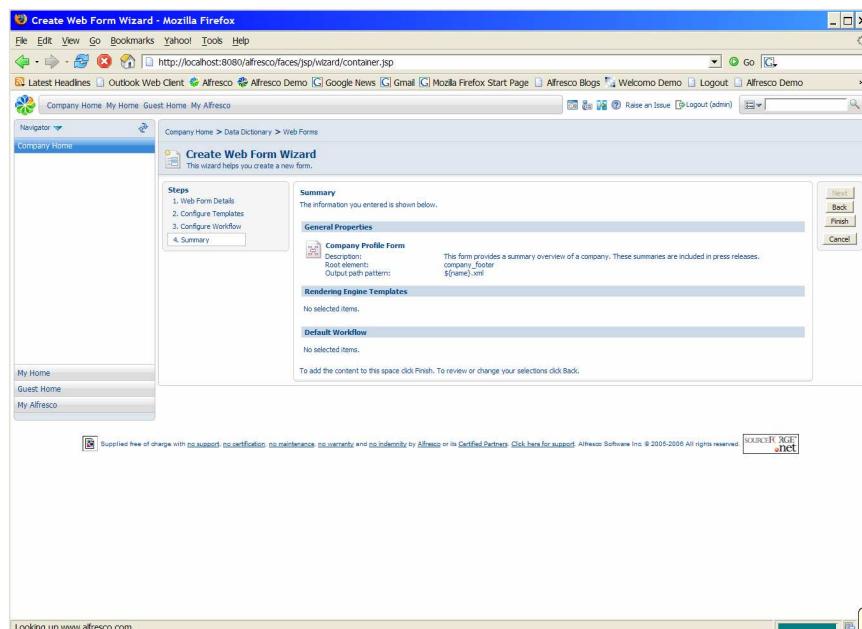


In Step Two of the **Create Web Form Wizard - Configure Templates** - leave all values blank and click **Next**. As Company Profiles are components sourced dynamically into individual pages, XML assets created by this form are not pre-generated.

In Step Three - **Configure Workflow** - select the “No not now” option and click **Next**. This step allows creation of default workflows for form data across all web projects. We will not set a default and instead configure workflow locally when creating our Web Project.



Preview the Summary screen. If everything is in order, click **Finish**. To correct anything, click **Back**.

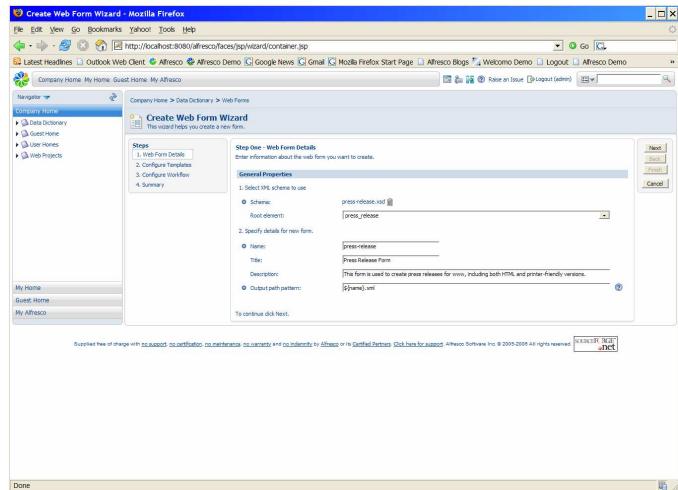


After finishing the **Create Web Form Wizard**, note the new space, **company-footer**. This is our newly-created form.

Now, re-run the **Create Web Form Wizard** to create the press release form by re-selecting **Create Web Form** from the **Create** menu.

In Step One of the Wizard, click **Browse** and select the file *press-release.xsd* from *C:\Alfresco\extras\wcm\forms*. After successfully uploading the schema *press-release.xsd*, enter the following values and then click **Next**.

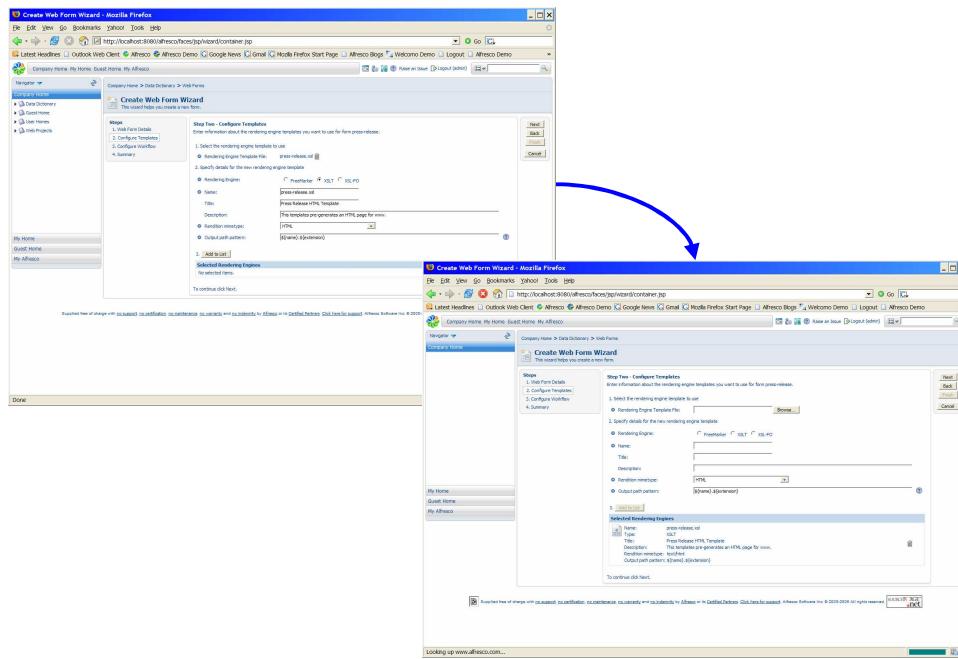
Title:	Press Release Form
Description:	This form is used to create press releases for www, including both HTML and printer-friendly versions.
Output path:	<code> \${name}.xml</code> (confirm default value)



Now, in Step Two of the **Create Form Wizard**, click the **Browse** button to associate a Rendering Engine Template file with the form. Select *press-release.xsl* from *C:\Alfresco\extras\wcm\forms* and, after successfully uploading the file, confirm that the appropriate rendering engine, XSLT, has been chosen by default. Then, enter the following values for the prompted fields:

Title:	Press Release HTML Template
Description:	This template pre-generates an HTML page for www.
Rendition mimetype:	HTML (confirm default selection)
Output path:	<code> \${name}.\${extension}</code> (confirm default value)

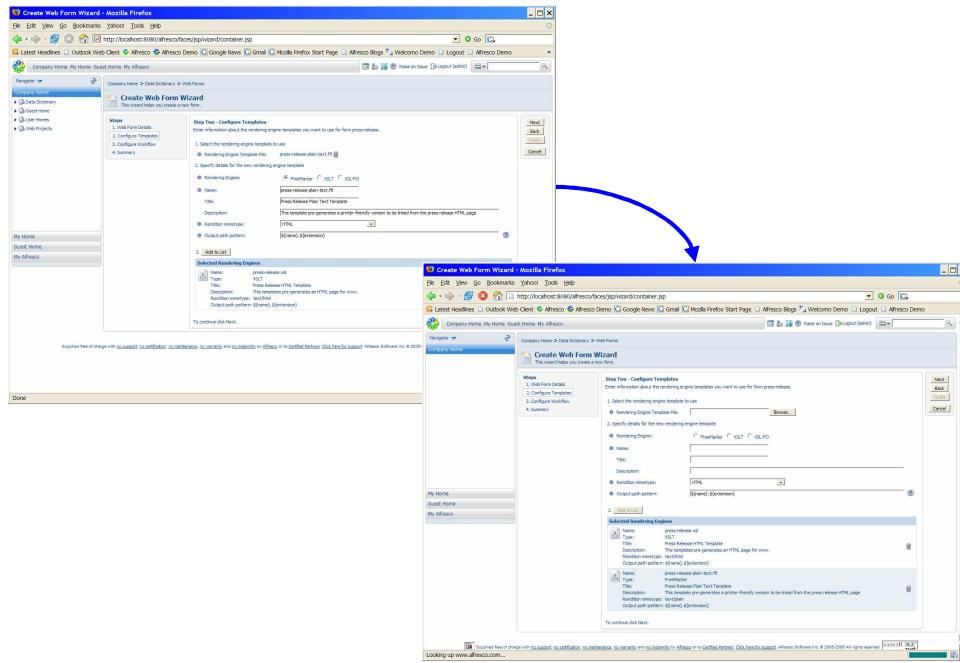
When finished, click on the **Add to List** button. See the new template association in the summary screen.



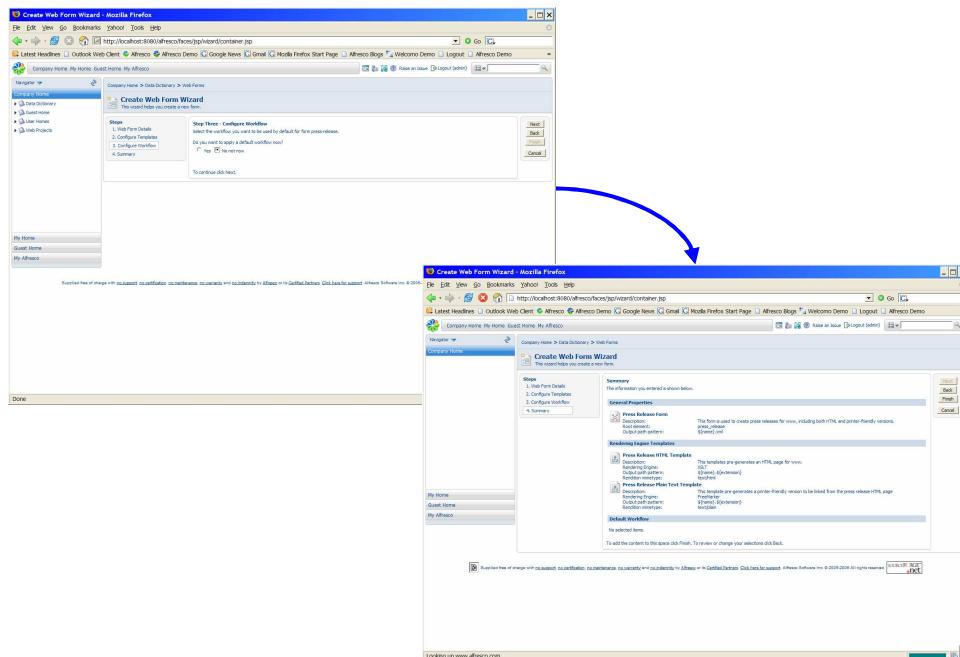
Now, add a second template association. Click the **Browse** button again, and upload the file **press-release-text.ftl** in **C:\alfresco\extras\wcm**. After successfully uploading the file, confirm that the appropriate rendering engine, Freemaker, has been chosen by default, and then use the following data for the remainder of the form:

Title:	Press Release Plain Text Template
Description:	This template pre-generates a printer-friendly version to be linked from the press release HTML page.
Rendition mimetype:	Plain Text (change default selection)
Output path:	<code> \${name}.\${extension}</code> (confirm default value)

As before, when complete click the **Add to List** button and confirm the new template association in the summary screen.



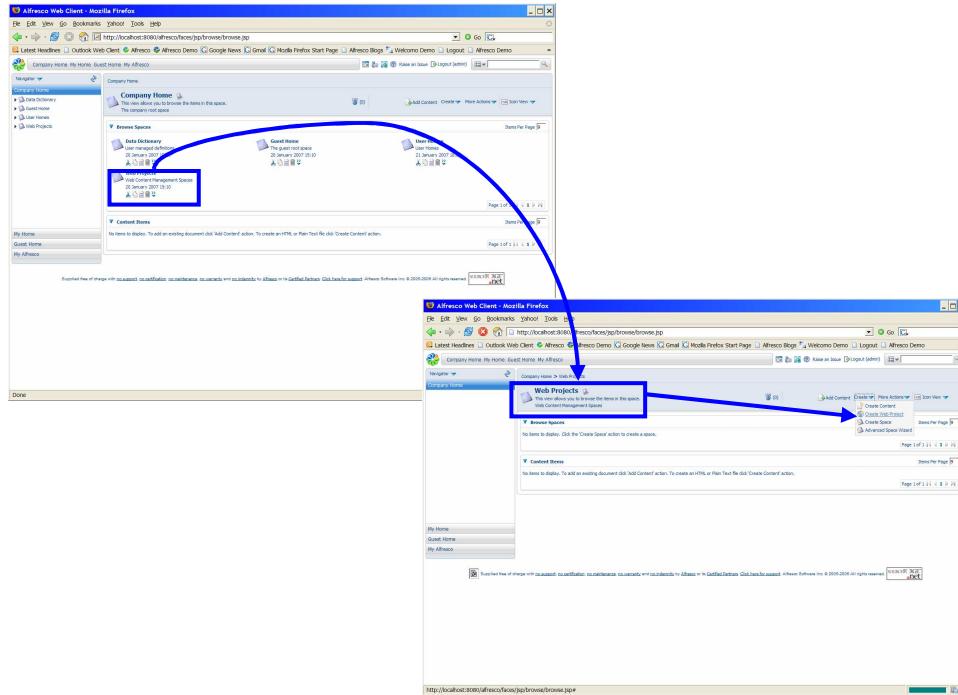
After clicking **Next**, in Step Three - Configure Workflow - select the “No not now” option. Click **Next** and in the Summary screen, confirm your configuration options for the press release form. If everything is in order, click the **Finish** button. If any changes need to be made, click the **Back** button.



Now, we have set-up and configured the default settings for our Web Forms. We are ready to create our first Web Project and use these forms to create and publish content.

Creating New Web Projects

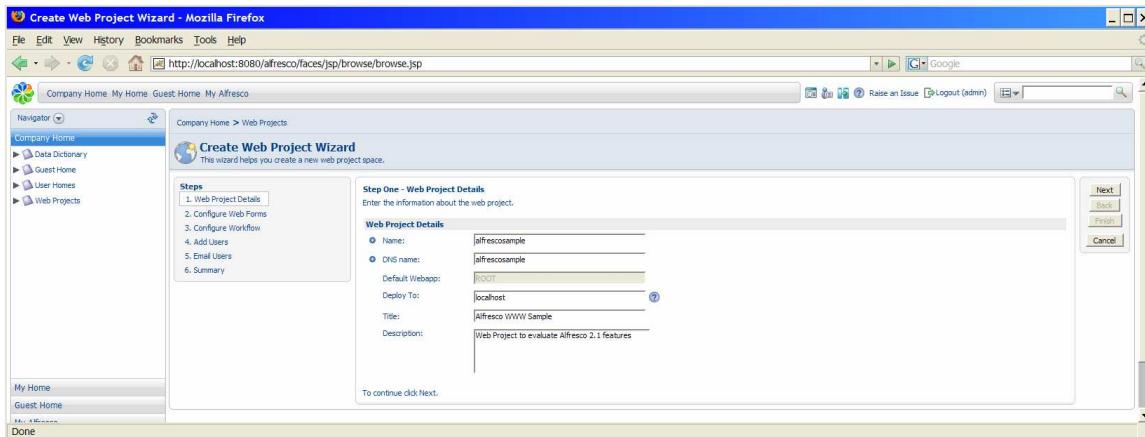
To create a new Web Project, navigate back to **Company Home**. Click on **Web Projects** and in the empty Web Projects space select **Create Web Project** from the **Create** menu.



In Step One of the **Create Web Project** Wizard, enter the name of your Web Project: "alfrescosample". When done, enter the DNS name, "alfrescosample". This DNS name will be used by our **Virtualization Server** to create a preview context on-the-fly for review and test of our website changes. Next, in the **Deploy To** field, enter "localhost". This specifies the name of the server to which we will deploy finalized, approved snapshots of our Web Project for public consumption. For our evaluation purposes, we will simply deploy new site version to a separate Alfresco repository on our same local server machine. This will avoid us having to set-up and configure a second box.

Lastly, fill in the *Title* and *Description* for our Web Project. Use the following values:

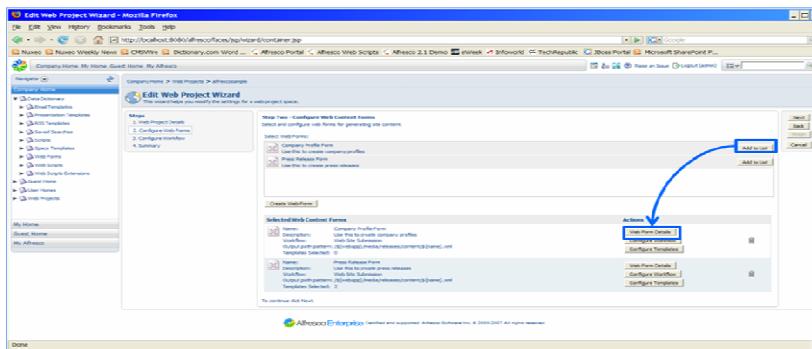
Title: Alfresco WWW
Description: Web Project to evaluate Alfresco 2.1 features.



When complete, click **Next**.

In Step Two - *Configure Web Forms* - see that this Web Projects has two Web Forms it can subscribe to. We will subscribe to both Web Forms in this Web Project and configure each for our Web Project's website's unique content generation and workflow requirements.

Next to both Company Profile Form and Press Release Form, click the **Add to List** button. In the **Selected Web Content Forms** section, click on the **Web Form Details** button next to the Company Profile form.



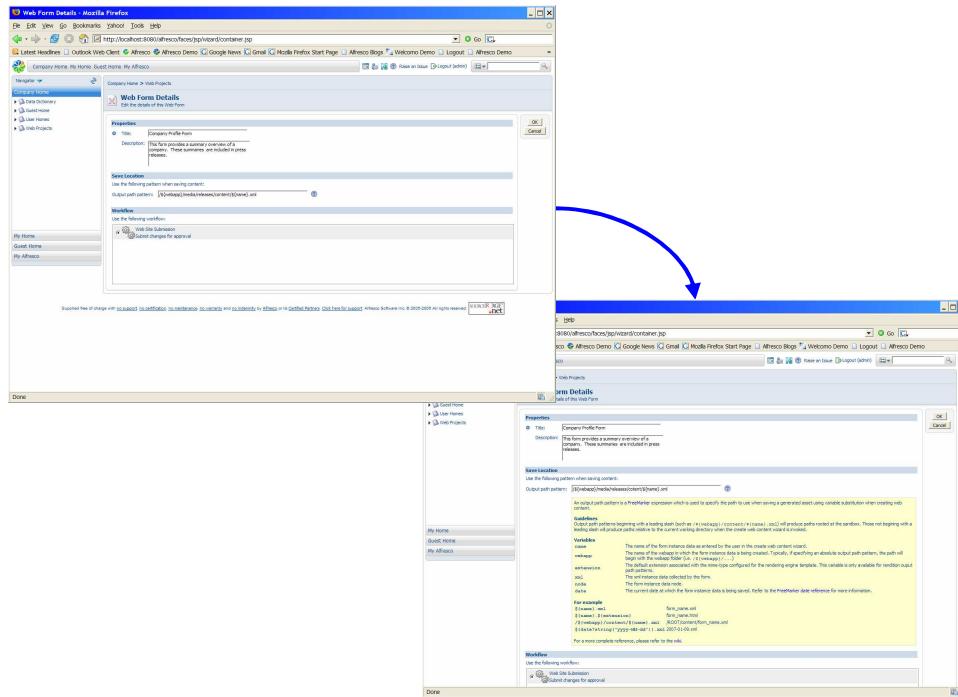
In Company Footer **Web Form Details**, accept the default name and description for the form for use in your Web Project. For output path pattern, type the following value:

Output path: `/${webapp}/media/releases/content/${name}.xml`

This expression specifies both the name and folder location of XML assets generated when using this form. Please note for this evaluation to include the leading "/" in the output path, which roots the generated path at the root of our web app. Without the leading slash, all generated paths will be relative to current folder location where the **Create Web Content** Wizard is initiated. For

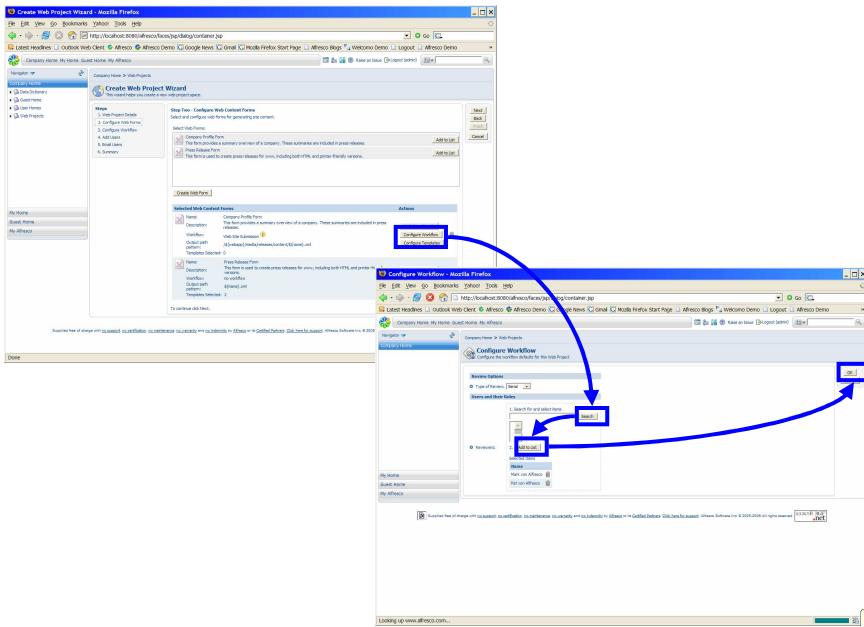
purposes of this demo, we want all paths fixed from the root of our website. Click on the “?” icon next the output path field to learn more details on available options for this expression.

Next, in the **Workflow** section, override the default no workflow option (seen by no workflows being currently selected) and select the single workflow option, *Web Site Submission*. When complete, click **OK**.



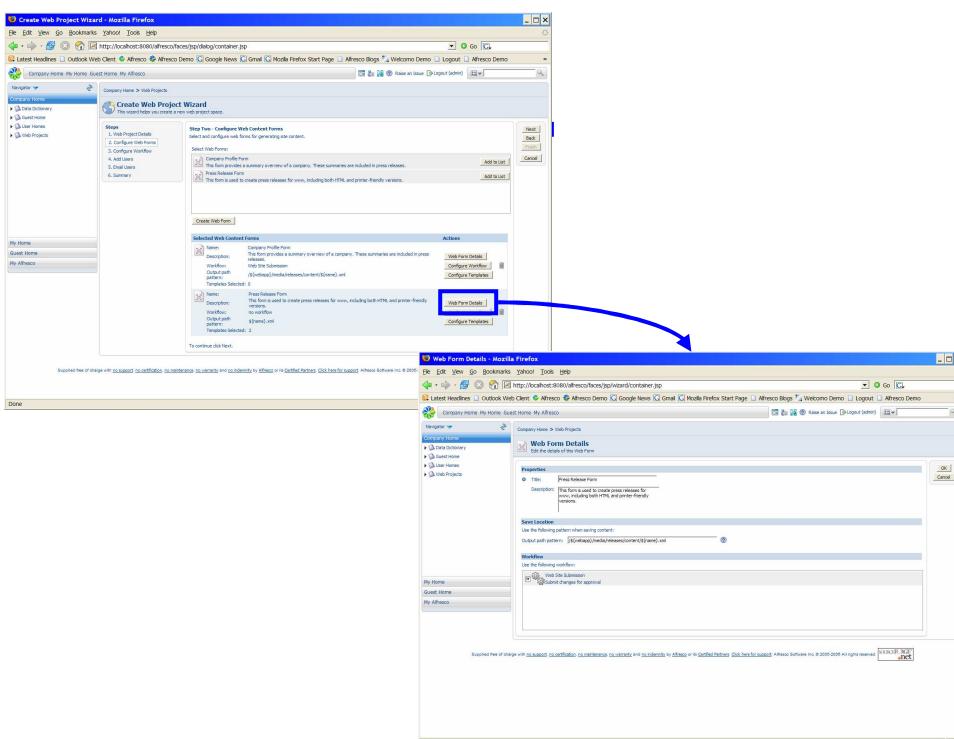
After completing **Web Form Details** for Company Footer, note the attention (“!”) icon next to workflow. Since we have selected the *Web Site Submission* workflow for this form, we must now configure that workflow. Click on the **Configure Workflow** button next to Company Footer. In the **Configure Workflow** screen, accept the default type of review, “Serial”.

For company footers, our Web Marketing Manager, Mark, must review and approve all profiles. To add Mark to the review list, type Mark’s name in the Search box, click the **Search** button, and in the Search Results window select “Mark von Alfresco”. Click the **Add to List** button, then click **OK**.

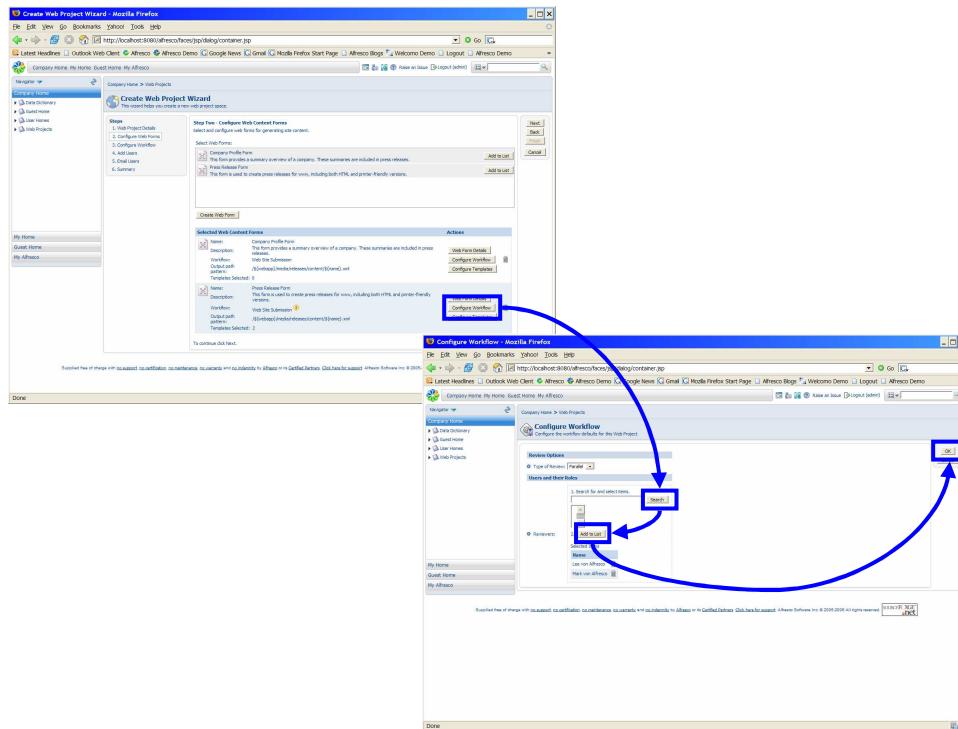


Next, configure the Press Release Form for use in our Web Project. Click the **Web Form Details** button for the Press Release Form and use the following values and then click **OK** when done.

- | | |
|--------------|---|
| Title: | Accept default value |
| Description: | Accept default value |
| Output path: | /\${webapp}/media/releases/content/\${name}.xml (same as Company Profile) |
| Workflow: | Web Site Submission |



Next, click the **Configure Workflow** button next to the Press Release Form. In the **Configure Workflow** screen, change the default type of workflow from “Serial” to “Parallel”. Then, add both our Web Marketing Manager, Mark, and our Legal Counsel, Lee, to the list of reviewers. When done, click the **OK** button.

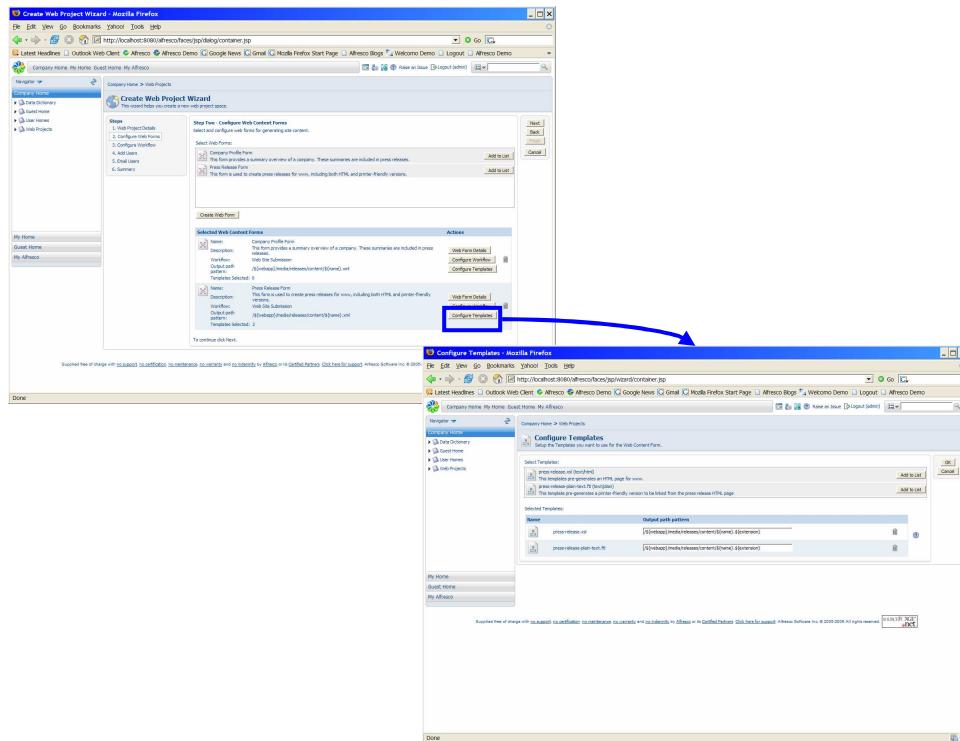


Now that workflow is configured for our Press Release Form, click the **Configure Templates** button. Note that we did not need to Configure Templates for Company Footer as that Web Form had no templates associated with it.

In the **Configure Templates** screen, accept the default option of using both available templates associated with this Web Form for our Web Project. In the future, other Web Projects may choose to use different templates or no templates at all (meaning no content is statically pre-generated). For output path pattern, enter the following for both templates:

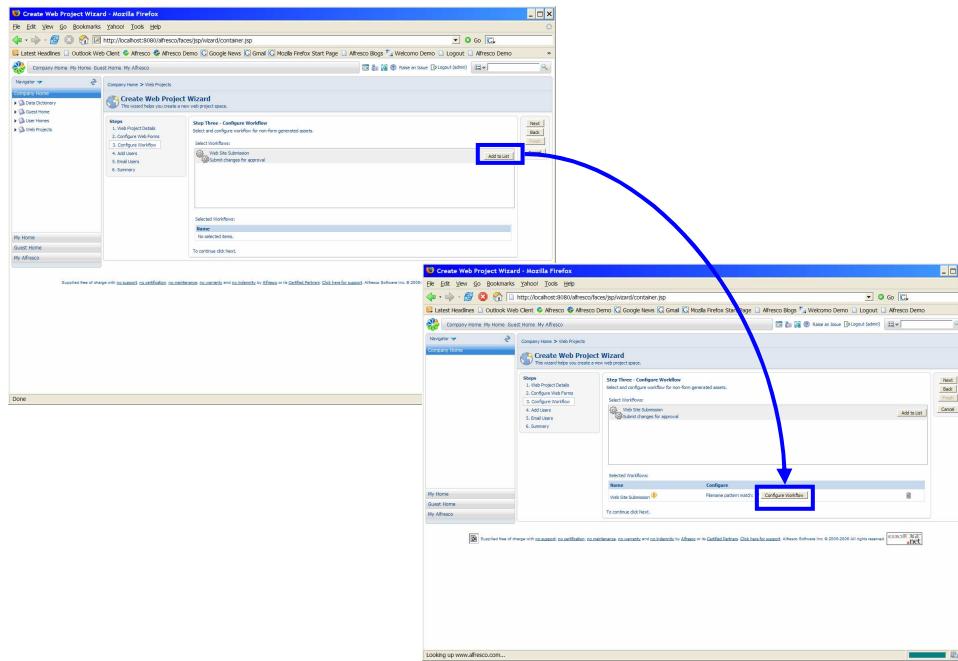
Output path: `/${webapp}/media/releases/content/${name}.${extension}`

This value governs how each template generates web assets, including both filename and output directory location. When done, click the **OK** button. After reviewing the configuration summaries for both Web Forms, click **Next**.



After completing Step Two, in Step Three now configure workflow for all changes to web assets that are not generated via Web Forms.

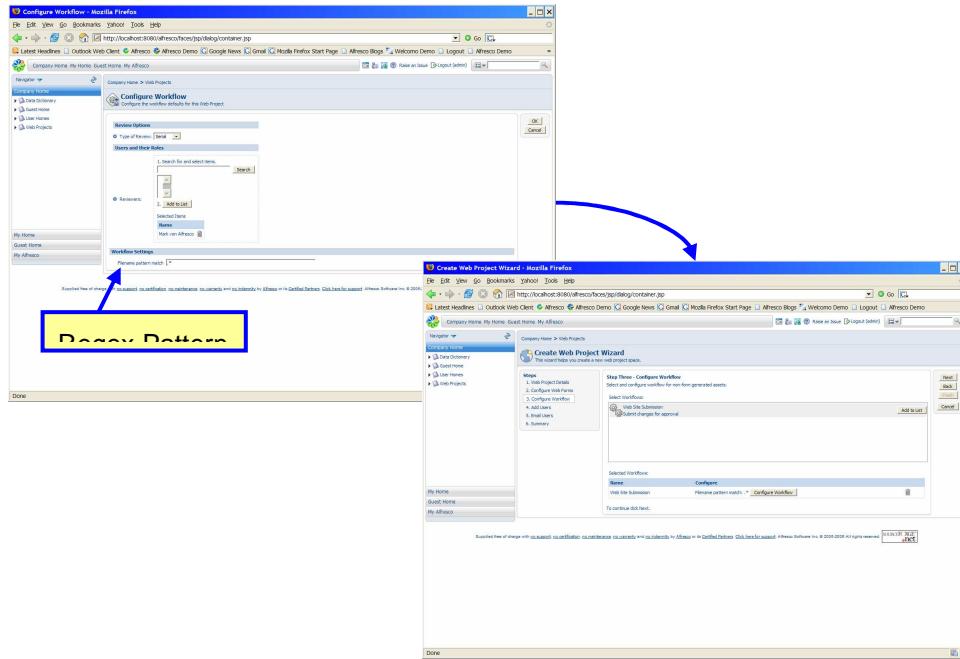
In the **Select Workflow** window, click the **Add to List** button next to the **Web Site Submission** workflow. In the **Selected Workflow** window, note the attention ("!") icon next our workflow. Click the **Configure Workflow** button next to our selected workflow.



In the **Configure Workflow** dialog, accept the default “Serial” review type. Add our Web Marketing Manager, Mark, as the sole reviewer for non-form generated web assets.

In **Workflow Settings**, note the default regex pattern match, `.*`. This default means than any asset other than those generated via a Web Form will go through this review process. Note that you can add the Web Site Submission workflow multiple times in Step Three of this wizard, and for each instance configure a different chain of reviewers for different sections of the websites or types of assets by modifying the regex pattern match in **Workflow Settings**. Examples includes having an additional reviewer – the Creative Director – for any assets under `/assets` or a Product Manager for anything under `/products`.

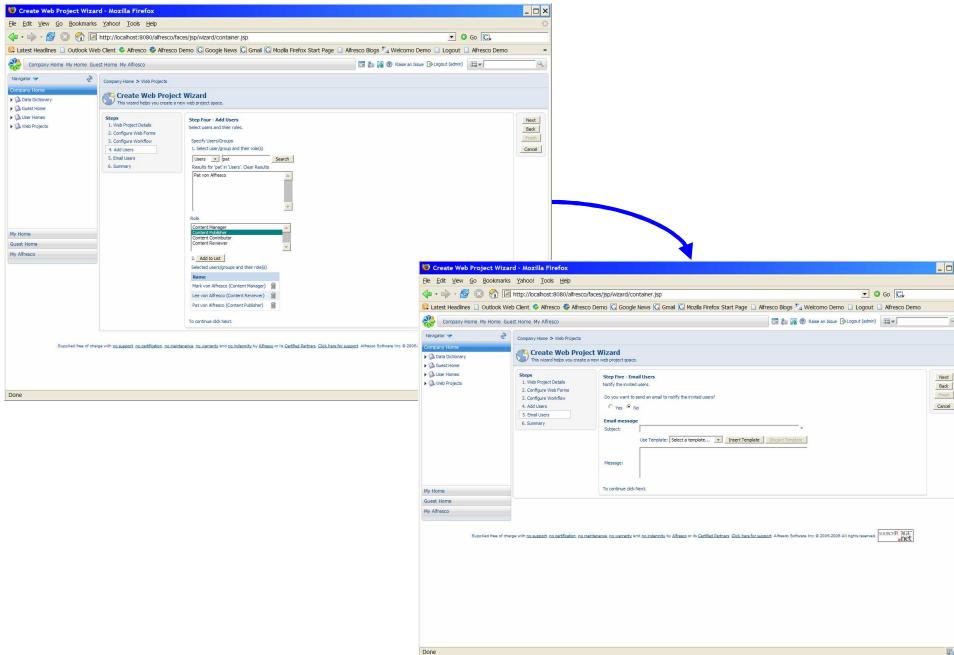
When complete, click **OK**. Returning to Step Three of the **Create Web Project** Wizard, examine the summary and click **Next**.



In Step Four, let's add our users. First, type "Mark" in the Search box and click **Search**. Select Mark from the Search Results window and then select Mark's role, "Content Manager". Then, Click **Add to List**. Confirm that Mark has been added as a Content Manager in the summary at the bottom of the window.

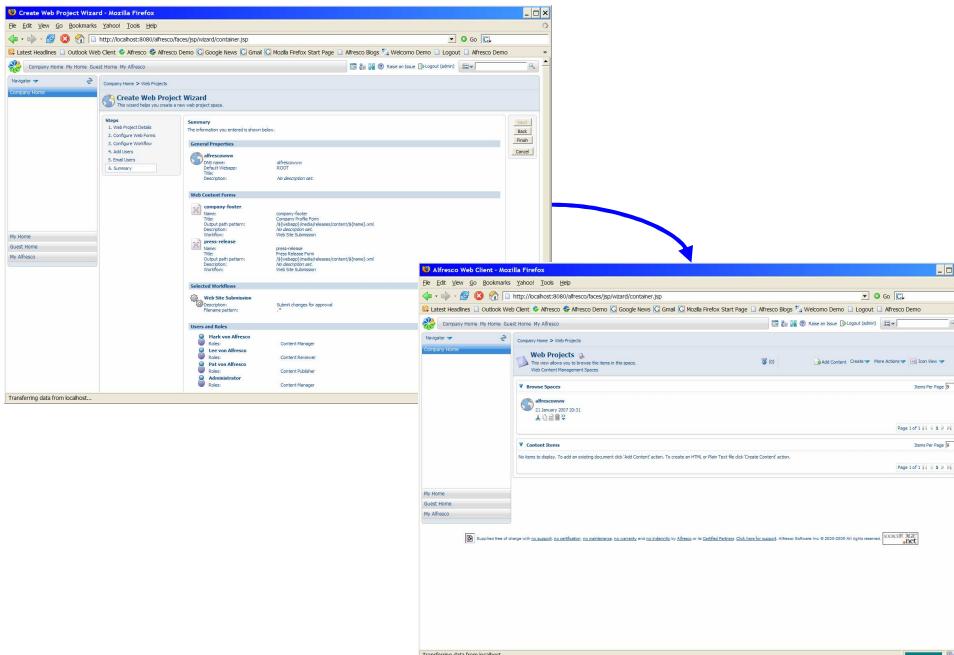
Repeat this process for both Alice, Pat, and Lee. Add Alice as a Content Publisher (Alice, as Product Marketing Manager, does contribute content to the website), add Pat as a Content Contributor (Pat, as a Product Manager, typically does not work within a Web Project and simply publishes documents from a collaboration space) and Lee as a Content Reviewer (as Legal Counsel, Lee does not contribute content to the www site). When done, click **Next**.

In Step Five, Email Users, accept the default option "No" for notifying users they have been added to the new Web Project. Click **Next**.



Finally, in Step Six, review a summary of our newly created Web Project. If everything is in order, click the **Finish** button. If any changes need to be made, click the **Back** button.

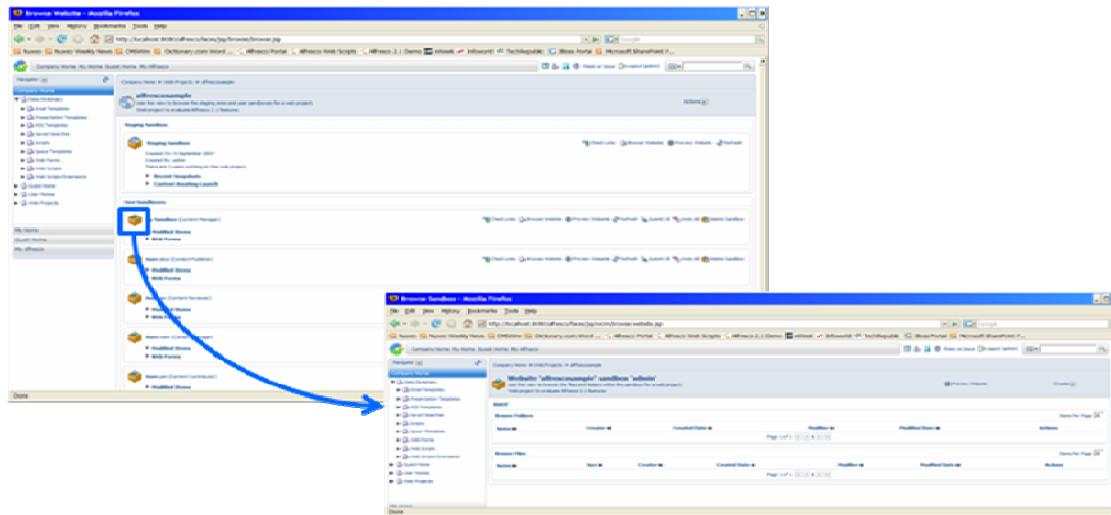
To see our newly created Web Project, *alfrescowww*, navigate to */CompanyHome/Web Projects*.



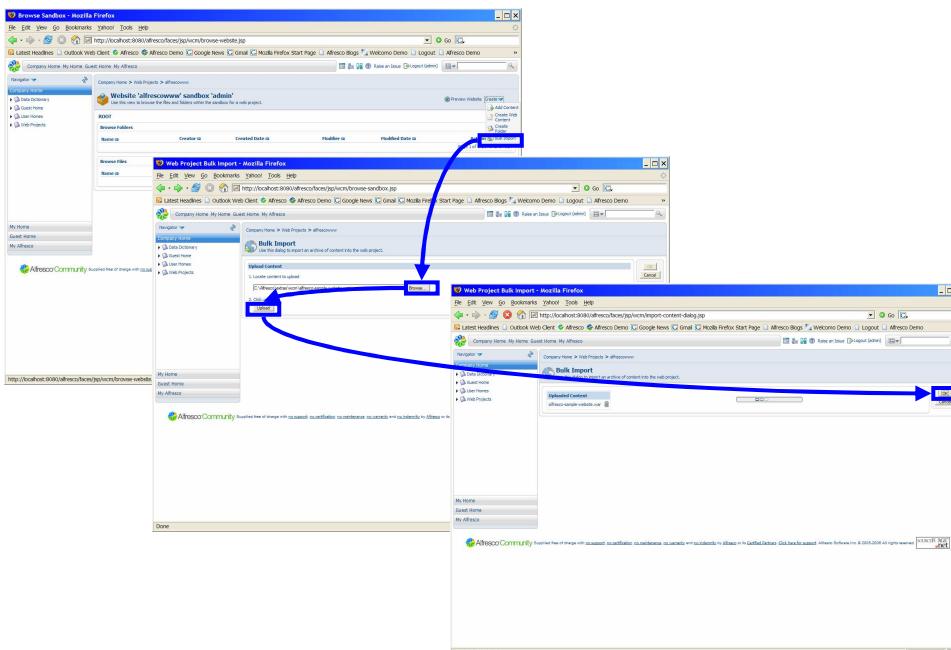
Importing the Sample Dynamic Website

With our Web Forms and Web Project created, let's import our default website content and submit to Staging to enable users to create and manage new content updates.

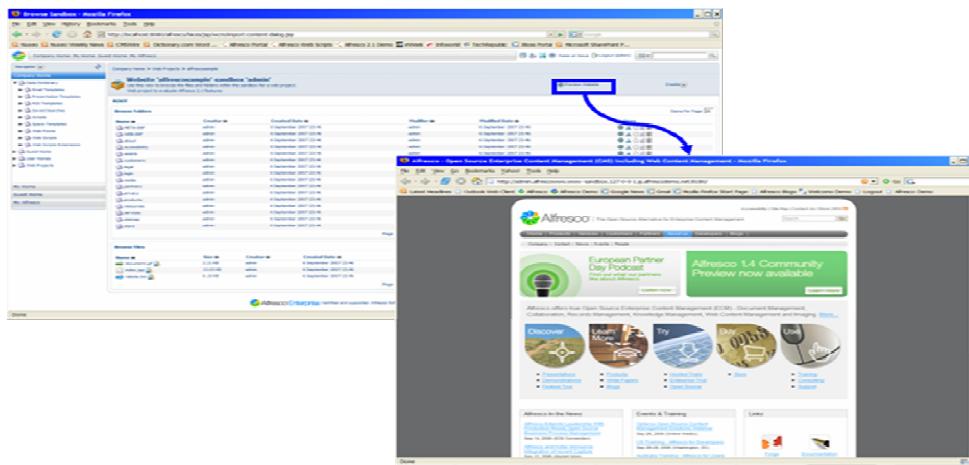
Click the *alfrescosample* Web Project in */CompanyHome/Web Projects*. In the Sandbox view of *alfrescosample*, click either the **Browse Website** action or the Sandbox icon for the Admin sandbox. See the current contents of the Admin sandbox, which is empty.



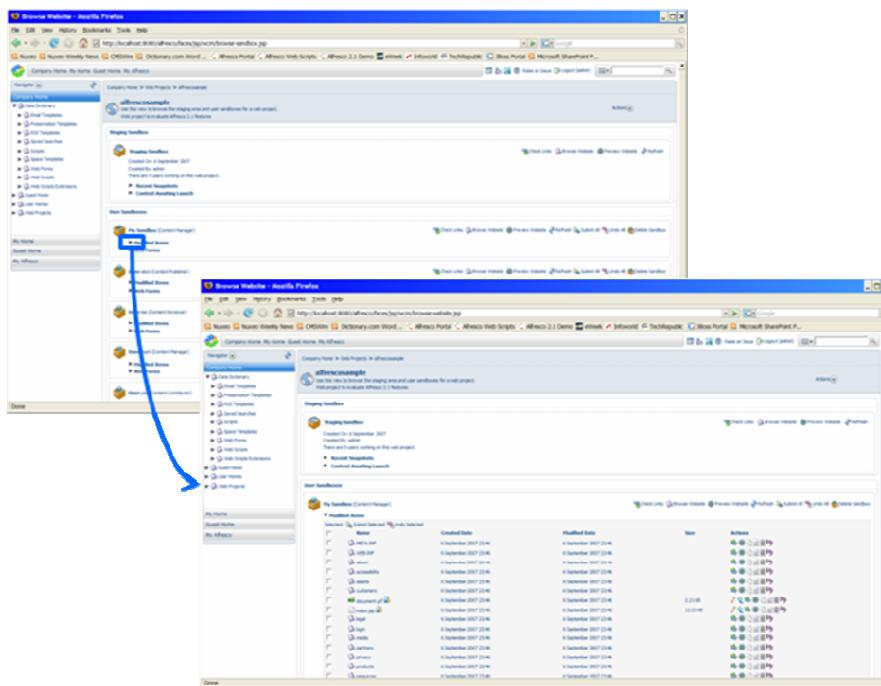
In the empty Admin sandbox, select the **Bulk Import** action from the **Create** menu. In the **Bulk Import** dialog, click the **Browse** button. Select the file, *alfresco-sample-website.war* from *C:\Alfrescoextras\wcm*. Once selected, click the **Upload** button. Once uploaded, click **OK**. The WAR file will be imported into the current directory, which by default is the ROOT web app.



Once the WAR file is expanded, preview the set of expanded assets in the Admin sandbox. Click the **Preview Website** action in the upper-right-hand corner to preview the site.



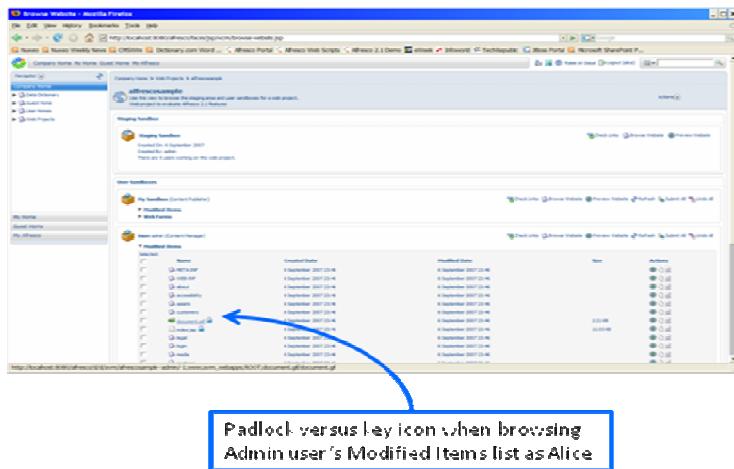
Now, click the *alfrescosample* Web Project in breadcrumb and return to the Web Project View. Click on **Modified Items** link in the Admin sandbox. See the list of all the recently imported assets from the sample war file.



Notice on your list of **Modified Items** the icon next to each file asset. This is a lock icon. This icon means that the current user, our Web Developer Admin, is editing this asset (in this case, by creating it as a new item during the **Bulk Import** process). Whenever an asset is created, deleted, or modified, Alfresco will first attempt to automatically grab a lock on that asset. If Alfresco can successfully take the lock, the user will be able to create or modify the asset. If Alfresco cannot take the lock, the user will be unable to complete their action. In Alfresco WCM then, end-users do not need to explicitly check-out an asset in order to work with it; instead, locks are automatic and implicit in every modification the user makes to the website.

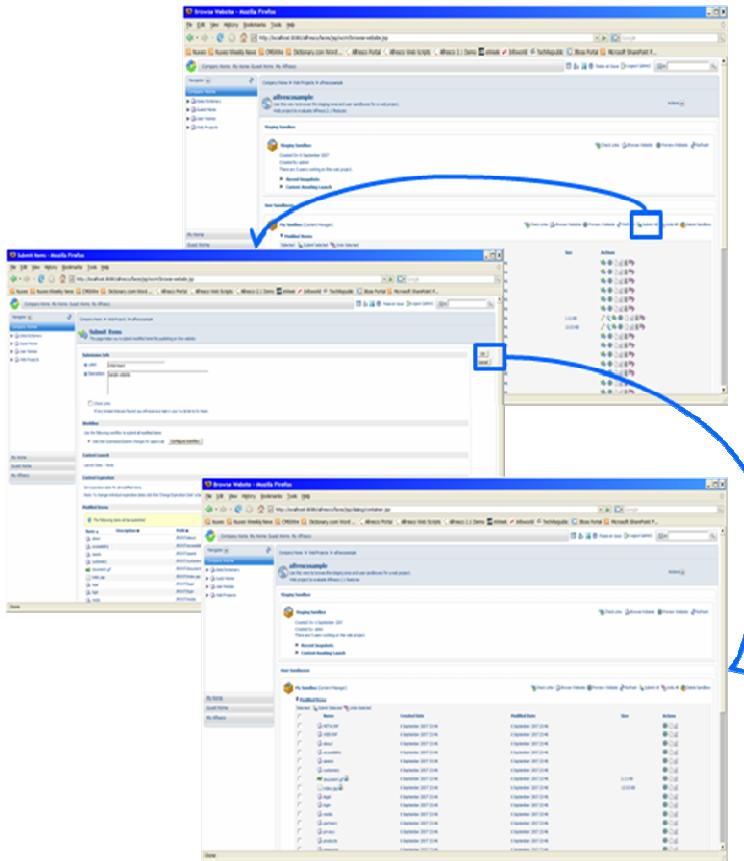
Looking at the lock icon, note that it is represented as a padlock with a key. The key means that you, the current logged in user, own that lock and are capable of editing, submitting, or undoing (reverting) changes to that asset. If you were to log in as a separate user, say our Content Publisher Alice, you'll see a different icon – a plain padlock icon. The plain padlock with a key means that the resource is currently non-editable. A convenient tooltip will also tell you who currently owns the lock. Because someone else has that resource locked, you will also see that your Actions list no longer has icons for **Edit** or **Delete**. Because your attempt to secure a lock as Alice would fail in any event, the Alfresco GUI conveniently strips away both **Edit** and **Delete** so that you don't accidentally try to do someone with that resource only to be thwarted with a friendly message telling you that someone has the asset locked. Simple, but a friendly way to save time and frustration for your end-users.

To see this in action, click **Logout** in the upper-right corner of your screen and then log in as Alice. After navigating to the *alfrescosample* Web Project, expand the Admin user's **Modified Items** list. Note the padlock icon (meaning the asset is locked but not by you). Now, navigate into the Admin user's sandbox by clicking on the sandbox icon. In the directory listing, note both the different icon and the lack of available **Edit** and **Delete** actions. These actions are first and foremost disabled because Alice does not have access rights to the Admin's sandbox for modifying content. Even if Alice had rights to this sandbox, however, we would see this same limited set of actions due to the lock held by the Admin user. Later in the tutorial as we modify existing content items in one sandbox, log out and log back in as a different user to see in that second user's sandbox how locked assets are not editable. Don't stray too far from the path of the tutorial, however. Make sure you can get back to where you were originally!



Now, let's log out as Alice and log in once again as our Admin user, navigating back to our Web Project and returning to our expanded list of **Modified Items**. From the list of **Modified Items**, click the **Submit All** action. In the **Submit Dialog**, provide a *Label* and *Description* for the submission (which will become the name and description of the corresponding Snapshot in Staging). Enter "Initial Import" for *Label* and "Sample website" for *Description*. Also in the **Submit Dialog**, note the selected checkbox, **Check Links**. Having this checkbox select will automatically initiate a workflow-driven links validation check. Deselect this option for now, as we will examine links management in greater depth later in the tutorial. For now, after providing both the label and description, confirm the list of assets you are promoting to Staging and click **OK**.

When complete, notice that the submitted items in the **Modified Items** list are no longer editable, even though the Admin user is the lock owner. This is because the assets have now been promoted for editorial review. While assets are under review, those assets are now only modifiable in the context of the actual review or edit task itself. This prevents the Admin user from making modifications to the asset underneath the reviewer and having potentially the wrong set of changes approved and promoted to **Staging**.

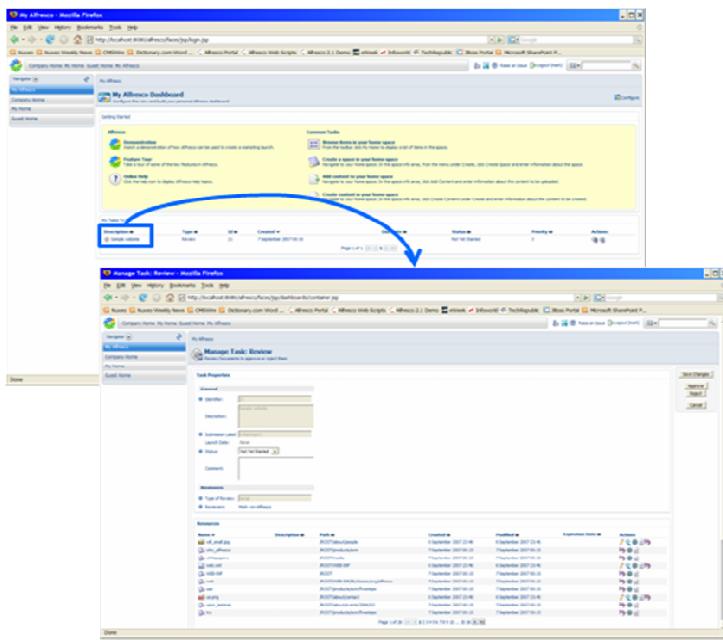


Now, log out as Admin and log in as our Content Manager, Mark.

On Mark's dashboard, note that Mark has a new review task with description "Sample website".

Click on the task, and see the lists of imported assets awaiting Mark's approval. Click **Approve**.

Note the task on the Admin's Task List. Open the task and click the Task Done button. The default WCM workflow includes a notification to the submitter of the successful approval. This notification is done via a task that the submitter must acknowledge to clear. For advanced users, the standard WCM workflow can be modified to replace this task with an email notification task, which would send the approval via an email rather than an item on the submitter's Task List.



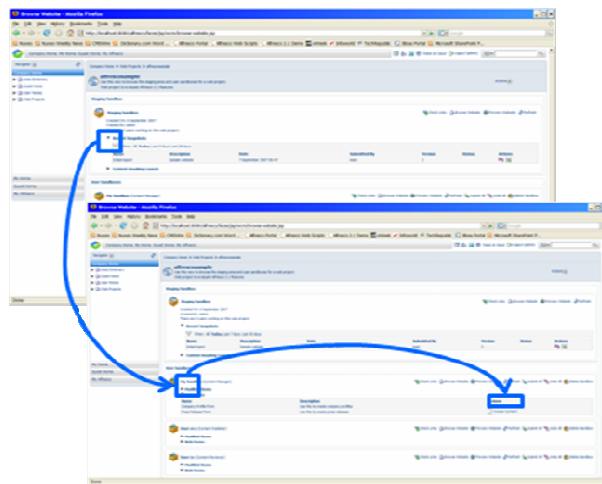
Now, we are complete with setting up our first Web Project. Logout as user Mark and log in again as Admin. On the Admin user's dashboard, note the notification task indicating that the submission was approved. Clear this task (see above Note). We are now ready to start creating and editing content on our site.

Creating New Web Content: Company Profiles

Return to the *alfrescosample* Web Project View. Click on the **Recent Snapshots** link in Staging. See that we have a new version of our website – Version 3 – which is the Initial Import (Versions 1 and 2 of the website are the original, empty versions of Staging).

Collapse the list of Snapshots. Expand the list of Web Forms under the Admin sandbox. See our two Web Forms, Company Profile and Press Release.

Now, we are going to start creating some new web pages. To start, click on the **Create Content** link next to our Company Profile form.



In the **Create Content** Wizard, enter “Alfresco” for the name of the new asset. Click **Next**.

In the Company Profile form, fill in the following values:

Profile #1:

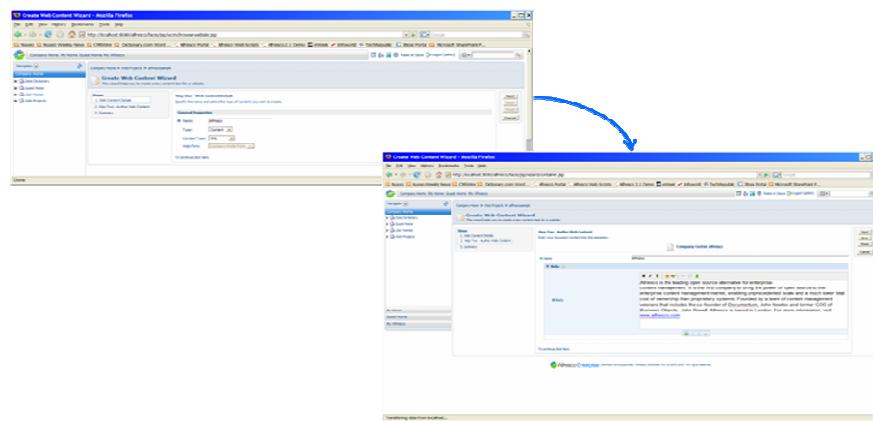
Name:

Alfresco

Body:

Alfresco is the leading open source alternative for enterprise content management. It is the first company to bring the power of open source to the enterprise content management market, enabling unprecedented scale and a much lower total cost of ownership than proprietary systems. Founded by a team of content management veterans that includes the co-founder of Documentum, John Newton and former COO of Business Objects, John Powell. Alfresco is based in London. For more information, visit www.alfresco.com.

When complete, click **Next**.



If the Press Release Form does not come up and instead generates an error message, make certain you have first started your Virtualization Server. This Web Form uses an in-line callout to dynamically generate a list of selectable Company Profiles. Generating this Web Form is dependent upon the Virtualization Server running. If you encounter this issue and your Virtualization Server is running, click the **Refresh** action on your sandbox in the Web Project view. This will force the Virtualization Server to reload your virtualized web app context for your sandbox, enabling proper preview and Web Form generation.

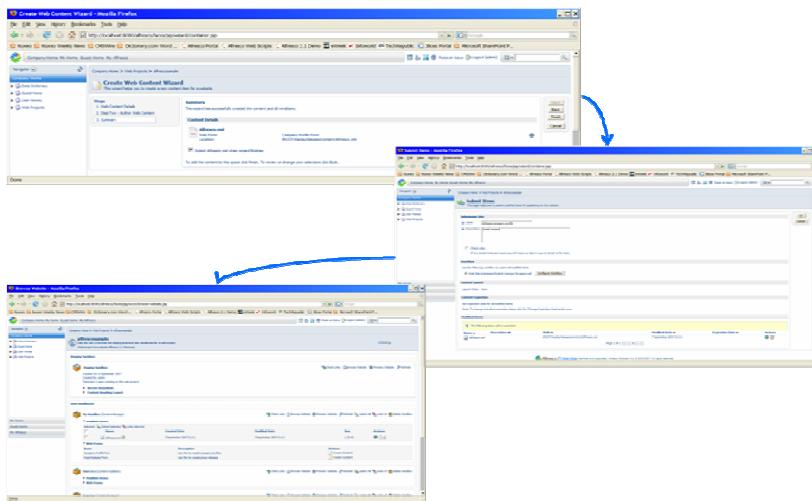
When completed with the form, note that the asset *Alfresco.xml* was generated. Note the filepath: */ROOT/media/releases/content/Alfresco.xml* generated from the output path expression we specified when creating our Web Project.

Also note the **Submit** checkbox underneath the generated filename. To speed publication, and because we no longer intend to work on this asset, select this checkbox and click **Finish**.

Notice now you are again returned to the **Submit Dialog**. Instead of having to initiate a separate submit process after completing the wizard, the convenient **Submit** checkbox can automatically launch our submission process for us. In the **Submit Dialog**, type “Alfresco Profile” for *Label* and “Added Alfresco profile” for *Description*. Also, make certain to deselect the **Check Links** checkbox. Click **OK**.

When returned to your Web Project view, expand your list of **Modified Items**. Notice you have the newly created asset, *Alfresco.xml*, which is currently locked by the Admin user but not editable. This asset is now under editorial review and until approved or rejected, is available only for preview.

We'll go through the review process later once we've created more Company Profiles.



Now, run through the **Create Content** Wizard using the Company Profile form three more times. Use the following data below. For each form, select the auto-submit option at the conclusion of the wizard.

Profile #2:

Name: Interdoc
Name: Interdoc
Body: InterDoc is a leader in the Enterprise Content Management field that provides business solutions to enable operational efficiency of critical business processes. The company has decades of experience in improving operational efficiency for commercial enterprises and government organizations and has used this experience to create a new generation of software solutions based on a core of open source software. For more information, visit www.interdoc.com.

Profile #3:

Name: Easypress
Name: Easypress
Body: Easypress Technologies (www.easypress.com) provides software for converting content between XML and QuarkXPress. Its flagship products include Atomik Roundtrip™, which provides full bidirectional XML support for QuarkXPress; Atomik Xport™, which provides automated QuarkXPress-to-XML conversion; and Atomik Import™, which provides flexible import of XML content into QuarkXPress documents. Easypress Technologies is an operating division of EvolvedGroup Limited and is headquartered in Guildford, UK.

Profile #4:

Name World Economic Forum
Name: World Economic Forum
Body: The World Economic Forum (<http://www.weforum.org>), based in Geneva, Switzerland, is an independent organization committed to improving the state of the world. Funded by the contributions of 1,000 of the world's

foremost corporations, the Forum acts in the spirit of entrepreneurship in the global public interest to further economic growth and social progress.

When completed, you should see the following four modified items in your sandbox:

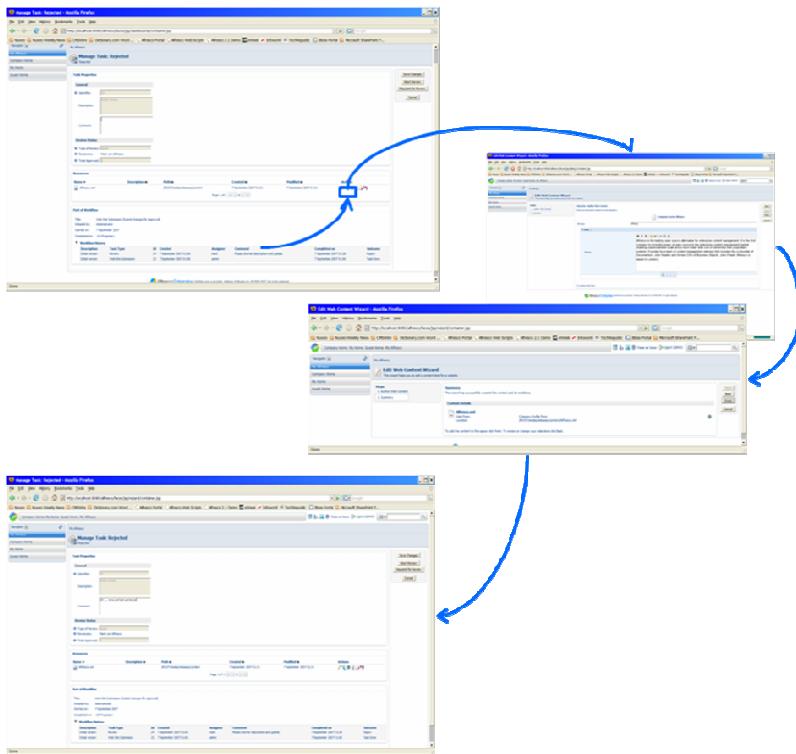
Name	Description	Created Date	Modified Date	Size	Actions
Alfresco.xml	Use this to create company profiles	7 September 2007 03:05	7 September 2007 03:05	1.05 KB	<input type="button" value="Check Links"/> <input type="button" value="Browse Website"/> <input type="button" value="Preview Website"/> <input type="button" value="Submit All"/> <input type="button" value="Delete Sandbox"/>
Interdoc.xml	Use this to create press releases	7 September 2007 03:14	7 September 2007 03:14	0.04 KB	<input type="button" value="Check Links"/> <input type="button" value="Browse Website"/> <input type="button" value="Preview Website"/> <input type="button" value="Submit All"/> <input type="button" value="Delete Sandbox"/>
Easypress.xml		7 September 2007 03:12	7 September 2007 03:12	0.12 KB	<input type="button" value="Check Links"/> <input type="button" value="Browse Website"/> <input type="button" value="Preview Website"/> <input type="button" value="Submit All"/> <input type="button" value="Delete Sandbox"/>
World Economic Forum.xml		7 September 2007 03:15	7 September 2007 03:15	1.05 KB	<input type="button" value="Check Links"/> <input type="button" value="Browse Website"/> <input type="button" value="Preview Website"/> <input type="button" value="Submit All"/> <input type="button" value="Delete Sandbox"/>

When completed with all four forms, log out as Admin and log in again as our Web Content Manager, Mark. As we did before, from Mark's dashboard, see the four outstanding tasks for each of the four new Company Profiles we created and automatically submitted after completing the wizard.

For the Interdoc, Easypress, and World Economic Forum profiles, open the **Task Dialog** and click **Approve**. For the Alfresco profile, let's review the process behind rejecting and re-submitting an asset for approval. In the **Task Dialog** for the Alfresco profile, in the Comment box write "Please shorten description and update". Once done, click **Reject**.

Once rejected, our serial review workflow will re-assign the asset as a task to the original submitter for modification. Logout as Mark and log in again in Admin. In the Admin user's dashboard, see the new task and open the **Task Dialog**. In the **Workflow History** section, note that the comment in the second task. This is the task assigned to Mark whose outcome is noted as "Rejected". After reviewing what changes Mark needs in order to approve the asset, click the **Edit** action on *Alfresco.xml* to launch the **Edit Content Wizard**.

In the wizard, remove the last sentence in the profile description field. Then, after clicking **Next**, notice in the **Summary** screen that you no longer have the checkbox option to auto-submit upon completion. This is because the asset has already been submitted and is currently under editorial review. Click **Finish**, and note that you are returned to the **Task Dialog** associated with the asset you just modified. In that **Task Dialog**, type the Comment “Removed last sentence and updated” and then click **Resubmit for Review**.



Now, log out as Admin and log back once again as Mark. In Mark's **Task List**, notice the “(2)” next to the task description. This number represents the current review cycle we are in for this content submission. The initial submission and review was review cycle 1. Since this asset was rejected, re-edited, and re-submitted to Mark, we are now review cycle 2. This is a convenient way for Mark to keep close track on those items he needs to be more careful reviewing; items going through a number of review cycles should have their **Workflow History** comments evaluated carefully to ensure all the proper modifications and corrections are being made.

After opening the **Task Dialog**, review the Admin user's last submit comment. Then, click the **Approve** action to promote this asset to Staging. This final promotion to Staging will once again send a notification task to the Admin user, and it will also release the lock held by the Admin user on the asset *Alfresco.xml*. This means that now any other user will be able to take a lock and edit *Alfresco.xml* in their own sandbox. Lastly, this final promotion to Staging also creates a new Staging snapshot. We'll look at the list of snapshots in Staging later, once we've finished creating all our of our web content.

Our four Company Profiles are now completed and submitted to Staging. We can now take the next step and use these profiles in our Press Release Form to create a number of new web pages for our website to highlight exciting news from Alfresco around our latest product launch.

Creating New Web Content: Press Releases

To create our press releases, return to the Web Project View and click the **Create Content** link next to the Press Release Form.

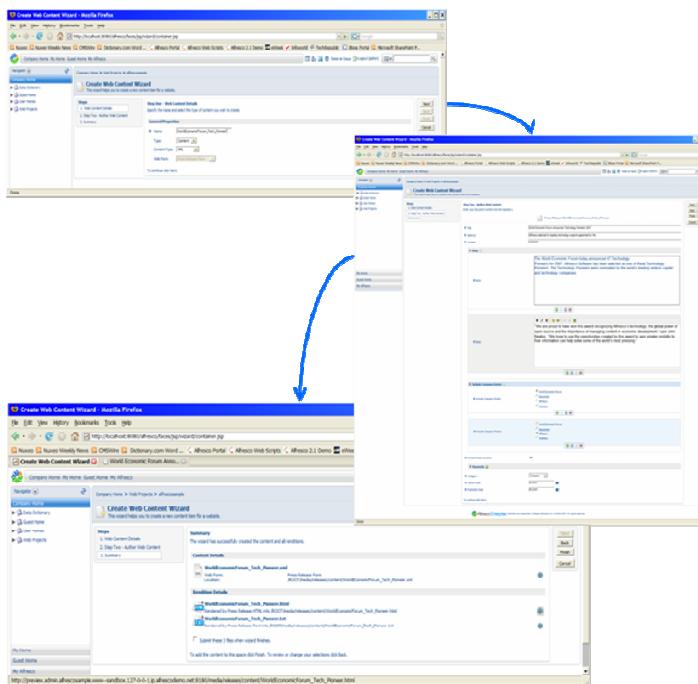
For our first press release, use the following data:

Press Release #1:

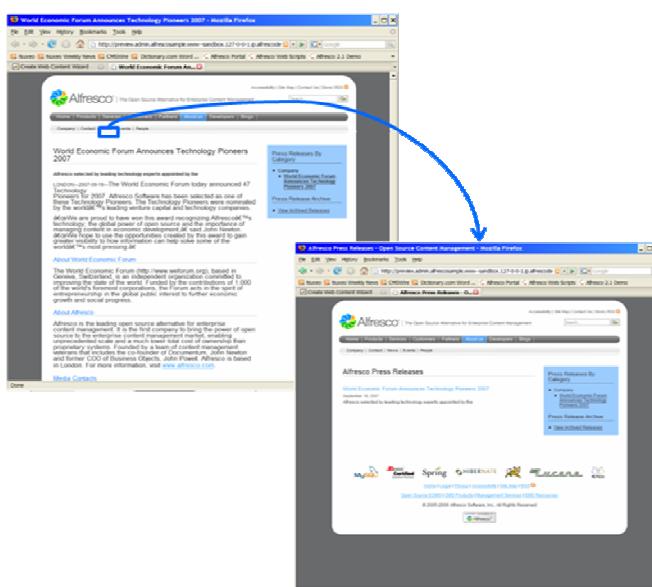
<i>Name:</i>	WorldEconomicForum_Tech_Pioneer
<i>Title:</i>	World Economic Forum Announces Technology Pioneers 2007
<i>Abstract:</i>	Alfresco selected by leading technology experts appointed by the World Economic Forum
<i>Location:</i>	LONDON
<i>Body #1:</i>	The World Economic Forum today announced 47 Technology Pioneers for 2007. Alfresco Software has been selected as one of these Technology Pioneers. The Technology Pioneers were nominated by the world's leading venture capital and technology companies.
<i>Body #2:</i>	"We are proud to have won this award recognizing Alfresco's technology, the global power of open source and the importance of managing content in economic development," said John Newton. "We hope to use the opportunities created by this award to gain greater visibility to how information can help solve some of the world's most pressing."
<i>Include Footer #1:</i>	World Economic Forum
<i>Include Footer #2:</i>	Alfresco
<i>Include Media:</i>	Yes
<i>Keywords:</i>	None (not used in demo)
<i>Category:</i>	Company
<i>Launch Date:</i>	Anything (not used in demo)
<i>Expiration Date:</i>	Anything (not used in demo)

After filling out each form, click **Next**.

In the **Summary** screen, note that both templates configured for the Press Release form have been applied and generated two output files, an HTML and Plain Text page. Note the filepath for the generated content, which corresponds to the rule we configured (via the output path pattern) in the **Create Web Project Wizard**.



Click the **Preview** icon next to the generated HTML page. See the virtualized view of the website with the pending changes. Preview the generated HTML page, check the link to the generated text file at the bottom of the page, and click on the **News** link in the submenu to see that the JSP index page now has a link to the to-be-published content. If any changes need to be made, click the **Back** button to modify and re-generate. When satisfied with the completed web page, click the **Finish** button. Make certain to leave the **Submit** checkbox UNSELECTED. We will be creating all three press releases first prior to submitting for review to update the site all at once with a series of new announcements.



With our first press release completed, now repeat the process two more times using the following data:

Press Release #2:

<i>Name:</i>	Easypress_OEM
<i>Title:</i>	Alfresco and Easypress Technologies announce OEM Agreement for Atomik Dynamic Publisher, powered QuarkXPress Server 7
<i>Abstract:</i>	Alfresco and Easypress Technologies Announce ground-breaking Collaborative Publishing
<i>Location:</i>	LONDON
<i>Body #1:</i>	Alfresco, the first and leading provider of an open source enterprise content management solution, today announced an OEM agreement to support the launch of Easypress Technologies' new Atomik® Dynamic Publisher product, which works in conjunction with the new QuarkXPress® Server 7 from Quark.
<i>Include Footer #1:</i>	Easypress
<i>Include Footer #2:</i>	Alfresco
<i>Include Media:</i>	Yes
<i>Keywords:</i>	None (not used in demo)
<i>Category:</i>	Company
<i>Launch Date:</i>	Anything (not used in demo)
<i>Expiration Date:</i>	Anything (not used in demo)

Press Release #3:

<i>Name:</i>	InterDoc_Alfresco_Solution
<i>Title:</i>	InterDoc Chooses Alfresco to Deliver Solution for the Government of Quebec
<i>Abstract:</i>	Alfresco continues Government Momentum in Canada
<i>Location:</i>	LONDON
<i>Body #1:</i>	Alfresco Software Inc., the first provider of an open source enterprise content management solution, today announced that it has been chosen by InterDoc to create a new regulatory government solution for customers including the government of Quebec.
<i>Include Footer #1:</i>	InterDoc
<i>Include Footer #2:</i>	Alfresco
<i>Include Media:</i>	Yes
<i>Keywords:</i>	None (not used in demo)
<i>Category:</i>	Company
<i>Launch Date:</i>	Anything (not used in demo)
<i>Expiration Date:</i>	Anything (not used in demo)

When completed with all three forms, return to the Web Project view and expand the list of **Modified Items** in the Admin sandbox by clicking the **Modified Items** link above the list of available Web Forms.

Note the complete list of all nine (9) newly created assets staged within the Admin sandbox.

Click the **Preview Website** link, and in the preview window, browse to [About Us → News](#). See the JSP page virtualize a unique dynamic view of the website as staged in the Admin sandbox. Test all three links to the newly created web pages.

Next, confirm that the newly created web pages are only available in the Admin sandbox. Click on the **Preview Website** link in Staging. Navigate to the [About Us → News](#) page. See that no press releases are currently available.

If you do not see the list of all published press releases in your sandbox preview, clear your cache and re-preview the page.

Content Launch and Expiration

Prior to completing this section, you will need to update the default schedule settings for content expiration. The default schedule for assigning expired items is once per day at 3:30am. To explore content expiration, we will need to change this default to every 5 minutes. Please see Appendix A on how to change this default setting, or skip past the last part of this section on Content Expiration.

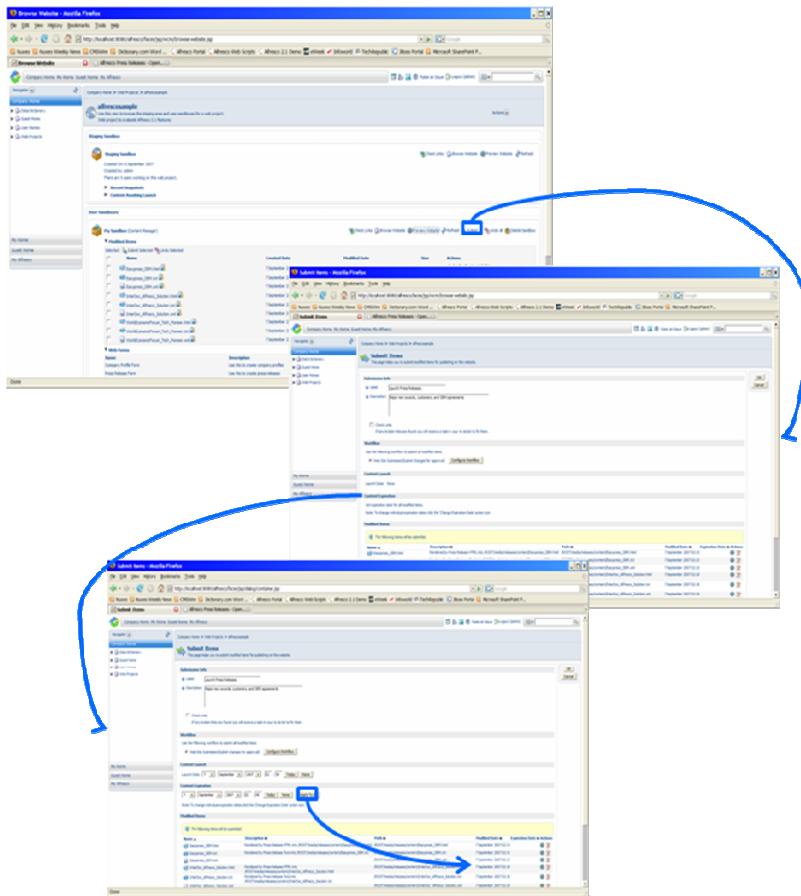
Now that we have finished all the updates we need to make to the site and have confirmed how those changes will look on www, let's publish these changes and submit for review by clicking the **Submit All** link in the upper-right of the **Modified Items** list (Note: We can promote just a single item or select list of items as an alternative to submitting all three press releases at once).

In the **Submit Dialog**, confirm the list of assets to be promoted for review and submission to Staging. For *Label*, enter "Launch Press Releases" and for *Description*, enter "Major new awards, customers, and OEM, and agreements". After this, deselect the **Check Links** checkbox; once again, we'll be examining this later in the tutorial.

In the **Content Launch** section, click on the **NONE** link next to the *Launch Date* label. Select a launch time that is between 10 and 15 minutes ahead of the current time (this is approximately how long we'll need to get through this next section). This launch date will be used to separately stage these assets once approved in their own mini-Staging sandbox. At our specified launch time, these assets will be automatically promoted from this mini-Staging sandbox into Staging proper. While in their own separate staging environment, these pending changes will be fully previewable against the evolving state of the current website as seen in Staging. This is useful for things like finalizing a set of site updates for an early Monday morning launch. Changes can be completed and approved the Wednesday before, and on both Thursday and Friday anyone can look forward and preview what the site will look like on Monday – inclusive of all incremental changes that have been made since the change were originally approved and Staged. In case anyone makes a change on Thursday or Friday that potentially impacts the Monday release, being able to continually preview against the evolving state of the website is one more way to QA your site to ensure its quality and accuracy.

Finally, set a *Content Expiration* date for your content. Select a global value that is between 5 and 10 minutes post our launch time. Hit **Apply To All** and note that in your list of submitted assets, the *Expiration* field has been filled in. Also note in the Actions list next to each item the **Expiration** icon. Clicking this icon will allow you to set specific expiration dates on an asset-by-asset basis to override your globally set default. This content expiration value will be used as part of a scheduled workflow process to automatically assign a task to the content author to review the asset for potential update or removal from the site. By ensuring Content Expiration dates are set on all your assets at time of submission, you'll be able to have an effective, auditable policy of regular site updates to keep your content fresh and relevant.

When complete, click **OK**.



Now, let's review and approve these assets as our two reviewers, Mark and Lee. Remember, for press releases we have configured a parallel review process. To see this, log in first as Mark, and then as Lee. Note that both users have at the same time the same exact task. When logged in as either user, open the **Task Dialog** and note that within this review task, while you can **Preview** the generated web pages (and correspondingly the entire dynamic site), you cannot edit. Because this is a parallel review task, and both reviewers at the same time may be looking at the asset, by default each reviewer has a READ-ONLY view. If your reviewers potentially need to make edits to the content, you can enable this by using a Serial review process. Because each reviewer in a Serial review process has exclusive access to the asset, modifications can be safely made without having any fear of potential conflicts or overrides (although Alfresco support branching and merging within the underlying repository, notions of conflicts edits and merges are a potential source of confusion for average business users. For this reason, any possibility of concurrent edits is prevented). To change the type of workflow from Parallel to Serial, you can access the **Edit Web Project Settings Wizard** under the **Actions** menu in your Web Project view. You can explore changing workflows after we complete this tutorial. For now, let's approve this content and see how content launch works in practice.

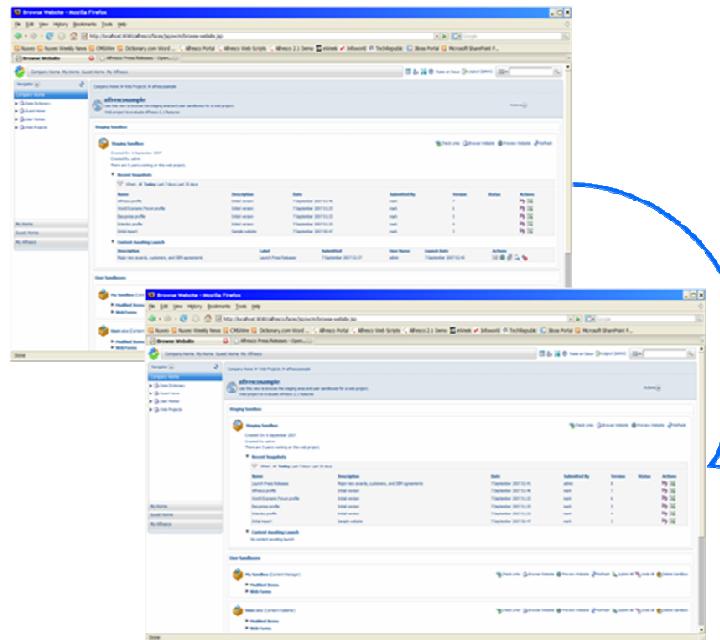
Logged in as either Mark or Lee, click **Approve** in the **Task Dialog**. Then, after one reviewer has approved the asset, log in as the Admin user to inspect her task list. Note that unlike in the serial review case, the Admin user has not been assigned a notification task. This is because this is a parallel review process, and although one of your reviewers has approved the content, the second reviewer has not yet approved the submission.

Let's do that now. Log in as your second user, and click **Approve**. When done, once again log in as Admin and note the task assigned to you in your task list with the status *Submission Pending*. Your press releases have now been approved and are currently being maintained in their separate staging environment until our set launch date. Open the **Task Dialog** and confirm the assets that are to be submitted to Staging. Also note your option to either **Abort Submit** or **Submit Now**. These are simple ways for the Admin user to override the set content launch date. Let's ignore these options for now.

At a later point once completed with this tutorial, you can experiment with having one or both the reviewer reject with comments and run through multiple review cycles. In the interest of time, we'll leave this for you to explore on your own at a later point.

Navigate back into the *alfrescosample* Web Project and expand the **Recent Snapshots** section and the *Content Awaiting Launch* section. Note that we have 5 current Staging snapshots corresponding to our initial import and each of the company profiles we submitted. Also note that we do not yet have a snapshot for the press releases we just approved. See these press releases in the *Content Awaiting Launch* section. Note in this section our options to either **Submit Now** or **Abort Submit**. These are the same options we just noted in the Admin user's **Task Dialog**. Also note that we can inspect the assets that make up this pending submission by clicking **View Details** and opening up the **Task Dialog** screen. Similarly, notice that we can preview this future state of the website, and, using the **Visual Comparison** action, bring up side-by-side preview windows for both the current Staging area and this future version of Staging. Click this now, and in each window, navigate to the **News** directory. Notice the multiple updates to the site that will be committed in just a few short moments. Now, close both preview windows. It's time to wait out our launch countdown to see the automatic promotion to Staging.

Once time is up, refresh your view of the *alfrescosample* Web Project. Note the newly created snapshot, version 8:



Now that our press releases have been committed, we'll want to ensure our audiences can see them by deploying this new snapshot to our production server environment outside our firewall.

Before we do that though, let's first finish looking at content lifecycle management, and first turn our attention to our about-to-expire content.

After another 5 minutes (or whatever time interval you specified between launch and expiration), return to the Admin user's dashboard, and note the new task on the task list, "Expired content in alfrescosample". Open this task and see all of the press releases we just successfully published. Click **Edit** on any one of the press releases in the **Task Dialog**. Note that you can click **Edit** on either the source XML asset or any of the generated assets. If you click **Edit** on a generated asset, the corresponding XML form will automatically be opened for you.

Once completed with your updates, click **Done** and **Re-submit All**. This will bring you back to the **Submit Dialog**, where you can provide a new *Label* and *Description* for your submit. You can also specify a new launch and expiration date for these items. And, as normal, once the **Submit Dialog** is closed, the updated press releases will go through their normal review and submit process. We'll assume you are already comfortable with this for now. Instead, we'll now take a look at last leg of the publishing process: deployment.

The screenshot shows the 'Submit Items' dialog in Mozilla Firefox. The 'Submission Info' section has 'Label' set to 'Updated press releases' and 'Description' set to 'updated press releases'. The 'Workflow' section shows a workflow step 'Web Site Submitter/Submit changes for approval' with a 'Configure Workflow' link. The 'Content Launch' section has 'Launch Date' set to 'None'. The 'Modified Items' table lists several items under 'The following items will be submitted':

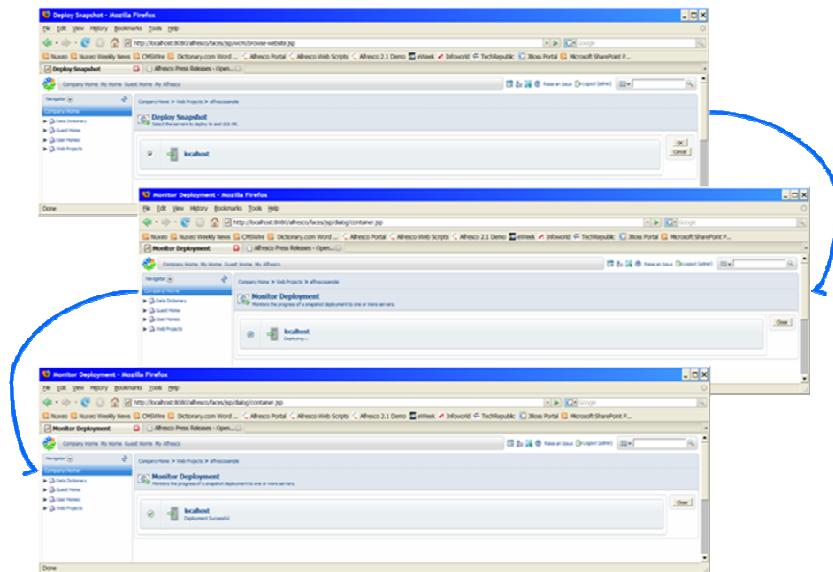
Name	Description	Path	Modified Date	Expiration Date	Actions
Alfresco_OEM.html	Rendered by Press Release HTML into /ROOT/media/releases/content/Alfresco_OEM.html	/ROOT/media/releases/content/Alfresco_OEM.html	7 September 2007 03:30	7 September 2007 03:30	
Alfresco_OEM.txt	Rendered by Press Release Text into /ROOT/media/releases/content/Alfresco_OEM.txt	/ROOT/media/releases/content/Alfresco_OEM.txt	7 September 2007 03:30	7 September 2007 03:30	
Alfresco_OEM.xls	Rendered by Press Release XLS into /ROOT/media/releases/content/Alfresco_OEM.xls	/ROOT/media/releases/content/Alfresco_OEM.xls	7 September 2007 03:30	7 September 2007 03:30	
InterDoc_Alfresco_Solution.html	Mapped to /Press Release HTML into /Root/mediareleases/content/InterDoc_Alfresco_Solution.html	/Root/mediareleases/content/InterDoc_Alfresco_Solution.html	7 September 2007 03:30	7 September 2007 03:30	
InterDoc_Alfresco_Solution.txt	Mapped to /Press Release Text into /Root/mediareleases/content/InterDoc_Alfresco_Solution.txt	/Root/mediareleases/content/InterDoc_Alfresco_Solution.txt	7 September 2007 03:30	7 September 2007 03:30	
InterDoc_Alfresco_Solution.xls	Mapped to /Press Release XLS into /Root/mediareleases/content/InterDoc_Alfresco_Solution.xls	/Root/mediareleases/content/InterDoc_Alfresco_Solution.xls	7 September 2007 03:30	7 September 2007 03:30	
WorldOfConcerns_Tech_Pioneer.html	Rendered by Press Release HTML into /Root/mediareleases/content/WorldOfConcerns_Tech_Pioneer.html	/Root/mediareleases/content/WorldOfConcerns_Tech_Pioneer.html	Downloaded by Press Release HTML into /Root/mediareleases/content/WorldOfConcerns_Tech_Pioneer.html	7 September 2007 03:30	

Site Deployment

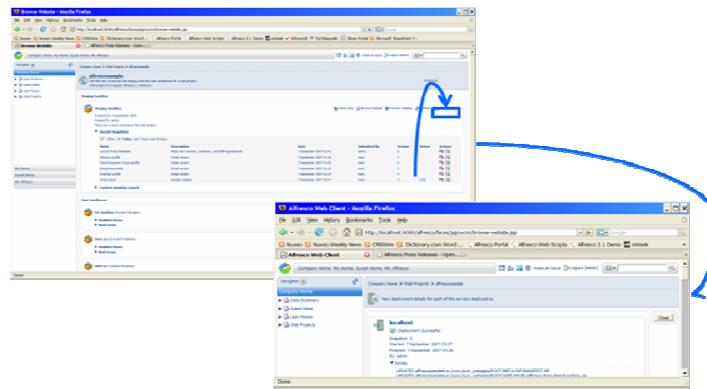
Now that we've published a new version of the site, let's deploy it and then look at how we can use deployment to rollback (and in the next section, examine the concept of rollback in even greater detail).

Remember that we configured our Web Project to use *localhost* as a target server. This means that we will replicate the incremental changes that comprise a selected snapshot to this named server (as a single guaranteed transaction). For evaluation purposes, we will deploy to a single server machine – the machine we are currently using for this evaluation – and replicate content between one Alfresco web content repository (our Staging sandbox) and to a second, run-time store (one not visible in our web client). This second store is what we will configure our production web server to use for hosting our publicly-accessible site. This second store is one that is not exposed via the Alfresco web client and is maintained entirely by our deployment process. As it is a full Alfresco web content repository, this store maintains its own version history, with each successfully completed deployment responsible for creating a new snapshot of the remote store. When deploying content, we use the known snapshot in the remote server for fast comparison to build a list of asset updates needed to synchronize the deployed site. But this is too much detail for this tutorial's purpose on deployment; if interested, you can learn more about deployment now on our wiki here: <http://wiki.alfresco.com/Deployment>.

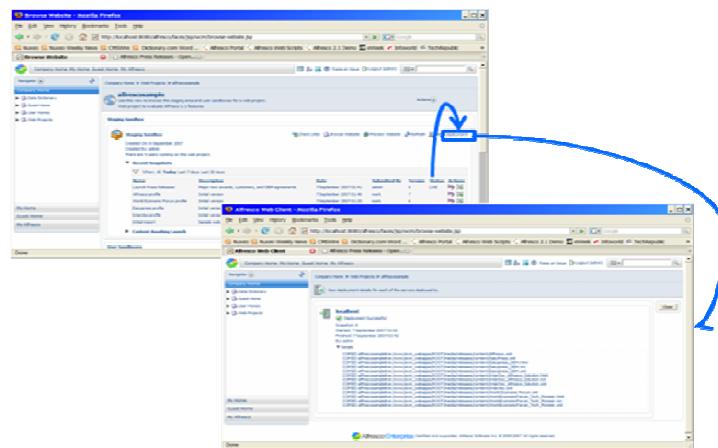
Let's first deploy the initial version of our site after our **Bulk Import**, version 3. In the Actions list next to version 3, click the **Deploy** icon. In the next window, confirm the selection of our remote target server, *localhost*. Click **OK** and watch the status indicator to note when the deployment is completed. In this initial deployment of our sample website, the deployment server is first identifying that no remote repository exists for this web project and then creating one on-the-fly. It then recognizes that this new repository is empty and proceeds with doing a full synchronization – transferring all assets into the new store and creating its first snapshot. The auto-generated name of the “hidden” repository on our current Alfresco server is the current Web Project’s name appended with the word “live”. In this case, the full name of the target run-time repository then is *alfrescosamplelive*.



Click **Close** on the **Monitor Deployment Status Dialog** upon successful deployment and, when returned to the **alfrescosample** Web Project view, expand once again the **Recent Snapshots** section under Staging. Note that version 3 is currently listed as being LIVE. To confirm what we deployed to make version 3 live, click on **View Deployment**.



Now, click **Close** and let's try deploying another version of our site, version 8. Open your **Recent Snapshots** list again and click the **Deploy** icon a second time, this time for our last published snapshot. Confirm the target server. When the deployment is complete, confirm in the Web Project view that version 8 is currently the live version of the site. Re-open the deployment report by clicking on **View Deployment**. Note the incremental changes that we successfully published to the site:



Snapshots and Rollback

There are three ways we can use snapshots to rollback our site:

- Switch run-time environment to serve a previous remote repository snapshot
- Re-deploy a previous Staging snapshot to the live remote repository
- Revert to prior Staging snapshot and re-deploy

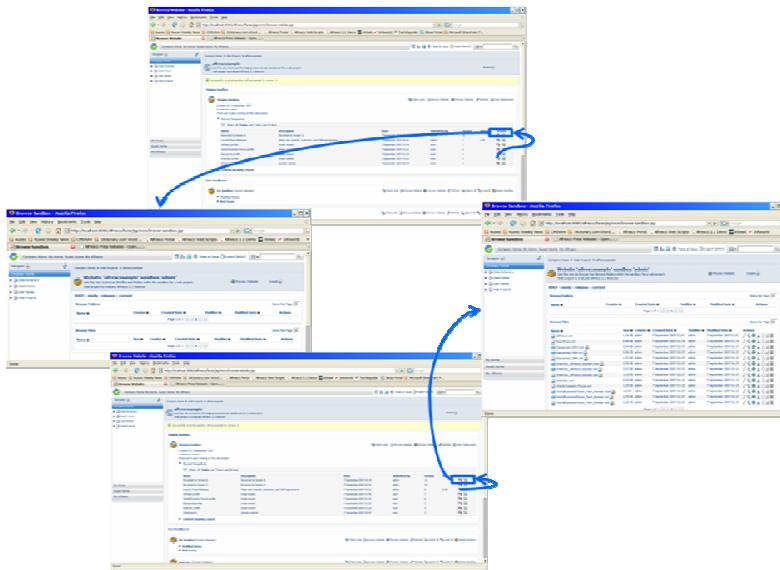
In the first case, remember that the target of our deployment is another Alfresco repository. This repository maintains a history of snapshots of all site versions it has received from our Staging sandbox. In our case, since we deployed both versions 3 and 8, the remote Alfresco repository will have two snapshots corresponding to each one of these versions of Staging. In the live environment then, even without having to formally ‘rollback’ or redeploy, our live web app can simply be retargeted to serve an older site version. This is useful, for example, in the event of a catastrophic failure. If the Alfresco server instance used for authoring were to be down, and your remote site somehow hacked or corrupted, you can still access and immediately reserve a known good state of your website by simply re-mapping your webserver or app server to a different context path - the path to the last known good site version of the site. Access to older snapshots on the live site is also useful for certain types of sites – for example a news site – where you may want to let users browse older, archived versions of the site instead of just archived versions of individual pages. In your generated web pages, you can easily include links to a fully functioning state of the same website, albeit an older version. This is an advanced topic, however, that while important to note is not necessary to review in any greater depth in this tutorial. For more information here, please review our wiki at <http://wiki.alfresco.com>.

In the second case, just as we rolled the site forward from version 3 to version 8 using the **Deploy** action, so too could we rollback by redeploying any previous version of the site – version 7, 6, 5, 4, or even version 3 once again. When redeploying an older version, all changes in the remote server will be undone – newly added files and directories deleted, any modified files reverted, etc. Since we’ve already examined deployment previously, we won’t review that in any more detail here. However, you can follow the example we walk through earlier when we reviewed deployment to test this out in various on your own.

In the third case, we can also rollback the site not only for purposes of ensuring an older version is currently being delivered in the live environment, but also for purposes of re-edits prior to deployment. An example of where this is useful is the rollout of some new section of the site or some new application functionality. In this case, simply redeploying an older site version is not enough – you also need to rollback the current working copy of the site for your Contributors, Publishers, Developers, and Designers to correct any errors, re-stage, and then re-deploy.

Rolling back your authoring and development environment to a last known good state for modification prior to redeployment is straightforward. In the Actions list next to the snapshot you want to rollback to, click on the **Revert** icon. Do this now for version 3 of the website. Note now that we have created a new version of Staging, version 10, that is revert back the state of the site as it existed in version 3. Please note that although a new version of the site, no underlying content has been duplicated in the repository; the new version is simply a new entry into the history chain of Staging noting that an older version has been pulled forward.

Once this is done, remember that all sandboxes in the Web Project are baselined off the current view of Staging. This means that each user instantly has an updated context for purposes of preview and edit. In this case, that means that each of the profiles and press releases we created no longer exist in each users sandbox – everyone has been wiped clean. You can see this by browsing into the `/media/releases/content` directory in any sandbox, and see that the contents of this directory are now empty. Similarly, you can click **Preview** on any sandbox and look at the **News** index page and see that it is back to its original blank state (if you continue to see a list of press release, refresh your cache – this is your browser playing tricks on you).

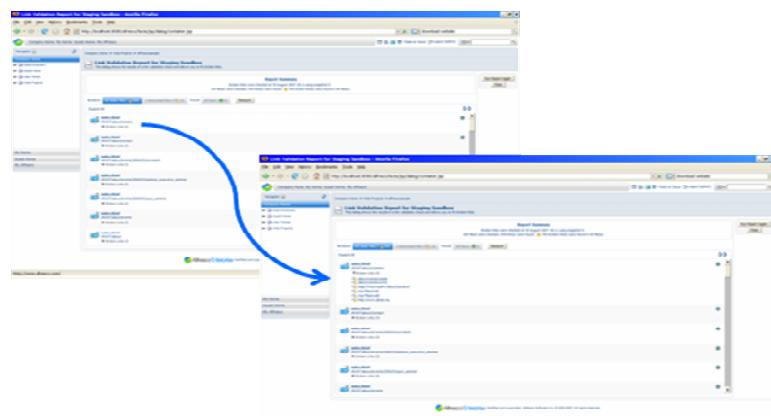


Links Management

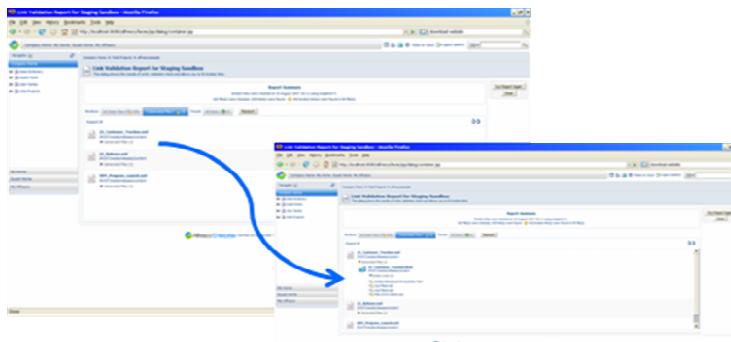
When a new snapshot is committed to Staging, Alfresco automatically scans each asset – both the data within the source file itself, and the “view” of that asset as served via an http request (to simulate the real page view an end-user would get). Alfresco uses this to build a master table of all references contained within the website, and from this allows end users to generate a report on the current integrity of references within your Web Project.

In the Web Project view, click the **Check Links** action next to the Staging sandbox. You will see the links report generated for the current version of our sample site. In this sample report, you’ll note that the Links Validation service examined 160 files in our sample site, in which there were a total of 659 links, 442 of which were broken. Since we purposely did not include all assets in the Alfresco website in our sample, this list of broken references is to be expected. We’ll see how we can use this links report to clean-up our sample website later. First, let’s simply look at the structure of the overall report.

You will notice that the links report is broken down into tabbed sections. The first tabbed section displays a list of all web assets not generated by Alfresco using a Web Form. Mostly typically, this list of assets may include hand-crafted HTML pages created in an authoring tool like DreamWeaver that you imported into Alfresco in a drag-and-drop manner via our CIFS interface. In this view, you can see each of the assets that have a broken reference. You can expand any one these assets to see what the actual broken references are:



The second tabbed view is a list of XML assets generated by Web Forms. If the Links Validation service determines that a web page has a broken reference, and that web page was in turn a page generated from an XML via a template, Alfresco will list the source XML as the relevant item for the end user to focus on correcting. The other pages when regenerated will be automatically fixed once the XML is edited. This is likewise useful as multiple pages may be generated from a single XML. In this view, you can see a list of XMLs that need correcting, the web pages they generated that contain broken references, and what those specific broken references are:

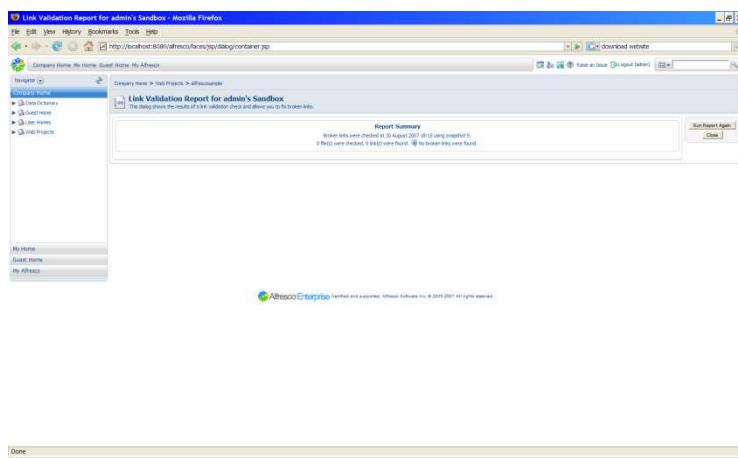


Lastly, you will see a tabbed section for Fixed Items. This report can be used as a tool for helping users within the context of their sandbox or a task to interactively edit, fix, and re-test content prior to submitting. In these contexts, each item in both the *Static Files* and *Generated Files* list provide an **EDIT** action to allow the user to fix the asset and recount the number of broken links. Those assets fully corrected by the end-user will show in this *Fixed Items* tab. Since we are currently viewing the links report of Staging, and since Staging is a read-only store (all modifications to Staging must first be made in a user sandbox, tested, and potentially workflowed for review), we do not see the **EDIT** action in our current view. Let's make some modifications in a user sandbox now to see exactly how this process works.

Links Management: User Sandboxes

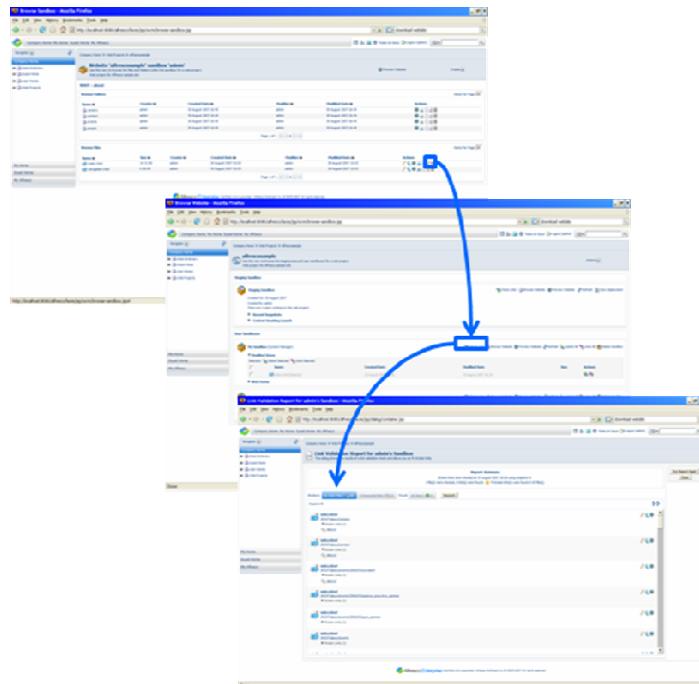
Within a user's sandbox, the notion of the overall state of the integrity of the state is not necessarily the most relevant information. For this reason, when reviewing a links report in a user sandbox, the report generated is an *incremental* check against those items that are modified within that sandbox. In this way, any end user can quickly and easily check whether or not their unique changes affect the site in a positive or negative way (breaking or fixing links). This lets end user focus on what most relevant and important to them: am I introducing a change that may impact site quality?

In the Web Project view, click the **Check Links** action against the Admin sandbox. Remember from our links report run against Staging that our site already has some serious referential integrity issues (as do many existing unmanaged static and dynamic sites when newly imported into a CMS). When run against our Admin user's sandbox, however, since the Admin user has no modified items, the links report runs a clean bill of health – there is nothing that the Admin user is doing that is potentially going to impact the overall correctness of site links.



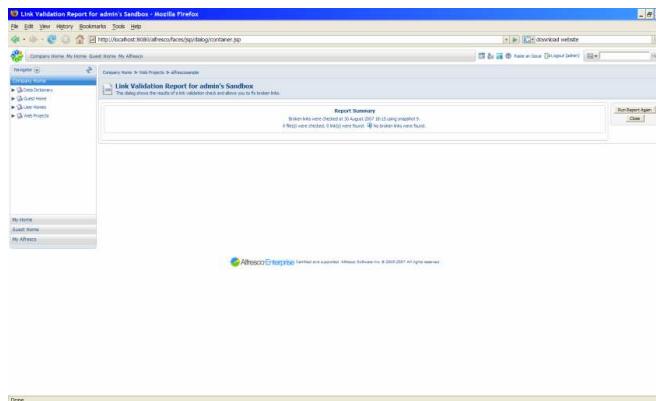
Now, let's see what happens when we do make a destabilizing changes. For this, navigate into the Admin user's sandbox into the */about* directory. In that directory list, in the Actions list, click the **Delete** action next to the static file (non-Web Form generated) *index.html*. In the **Deletion Confirmation Dialog**, confirm your deletion.

In your Web Project view, note that the Admin user's **Modified Items** list is now showing our recent deletion. Click on the **Check Links** action once again. Look at the **Report Summary** at the top of the report, and note that one file (our deletion) was checked and it had 0 links (the file doesn't exist anymore, and so it does indeed have no data and therefore no links). Also note that outside our single modified asset, there are 69 other assets that now have a total of 74 broken links. Looking at the report breakdown, we can see that 66 of those are static HTML assets created using a standard HTML authoring tool, and 3 are those assets we generated via Web Forms:



Now that we've seen the potentially disastrous effect deletions this asset would have on the site, in your Web Project view click **Undo All** on the Admin user's sandbox. Note that the **Modified Items** list is no longer showing any modified items, as we have now reverted the entire Admin user's sandbox back to its original, unmodified state (the current state of Staging, which contains the undeleted asset */about/index.html*).

Re-run the links report for the Admin sandbox. See that no incremental broken links are being reported and that we are indeed back to a whole and good state:



But what would have happened had the Admin user not checked the links report prior to committing her change, or worse, continued to promote that change regardless of its effect on the site?

For that, we turn now to workflow and how links validation is enforced as part of your business process.

Links Management: Workflow

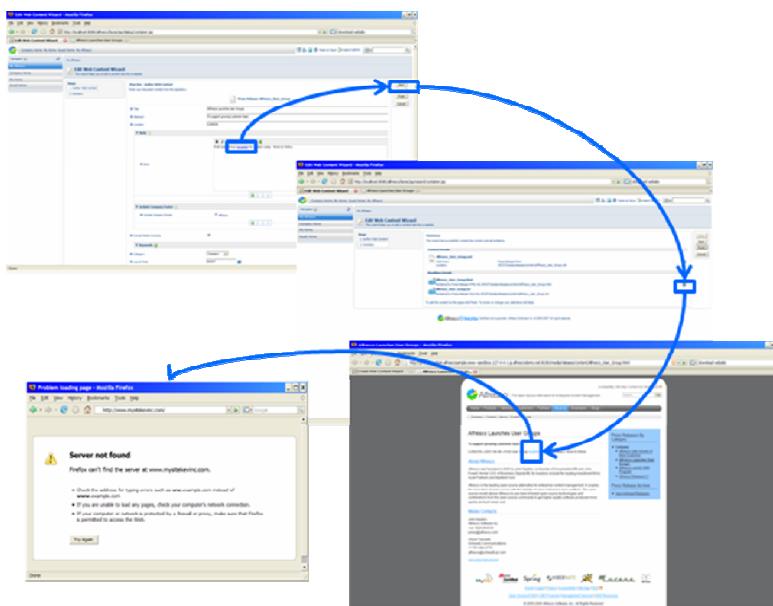
To see how we can enforce links validation as part of our business process, let's first create a new press release, this time with a hyperlink reference to a non-existent external resource embedded within our main body content using our TinyMCE in-line hyperlink control.

In the Admin user's sandbox, click the **Create Content** link next to our Press Release Web Form. Use the following as input data:

Press Release #5:

<i>Name:</i>	User_Group_Launch
<i>Title:</i>	Alfresco Launches Worldwide User Groups
<i>Abstract:</i>	Strong demand from customers and partners
<i>Location:</i>	LONDON
<i>Body #1:</i>	User groups launched in Boston and Los Angeles with more to follow.
<i>Include Footer #1:</i>	Alfresco
<i>Include Media:</i>	Yes
<i>Keywords:</i>	None (not used in the demo)
<i>Category:</i>	Company
<i>Launch Date:</i>	Anything (not used in demo)
<i>Expiration Date:</i>	Anything (not used in demo)

In the main body field in the Web Form, select the word "Boston" and click on the **Hyperlink** icon on the TinyMCE toolbar. Where prompting for an URL, type an arbitrary fictitious value to some nonexistent external resource, for example, <http://sample.alfresco.com>. When done, click **Next** and in the **Summary** screen click on the **Preview** icon for the generated HTML page. See the hyperlink you just created in the generated web page, click in the link, and confirm that indeed you get a 404 error.

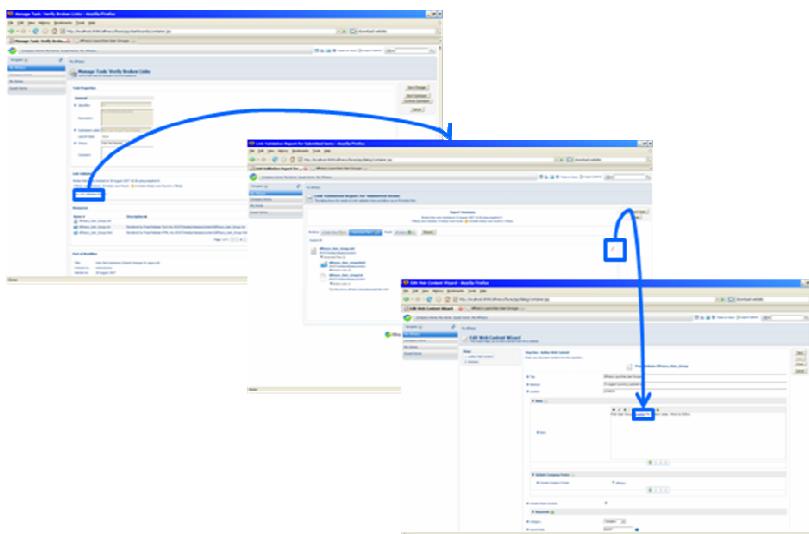


After confirming your dead hyperlink, instead of clicking **Back** in the **Summary** screen, instead select the **Submit** checkbox and click **Finish**. In the **Submit Dialog**, confirm the assets you are promoting to Staging, and fill in an appropriate *Label* and *Description* for your submission. Click **OK**.

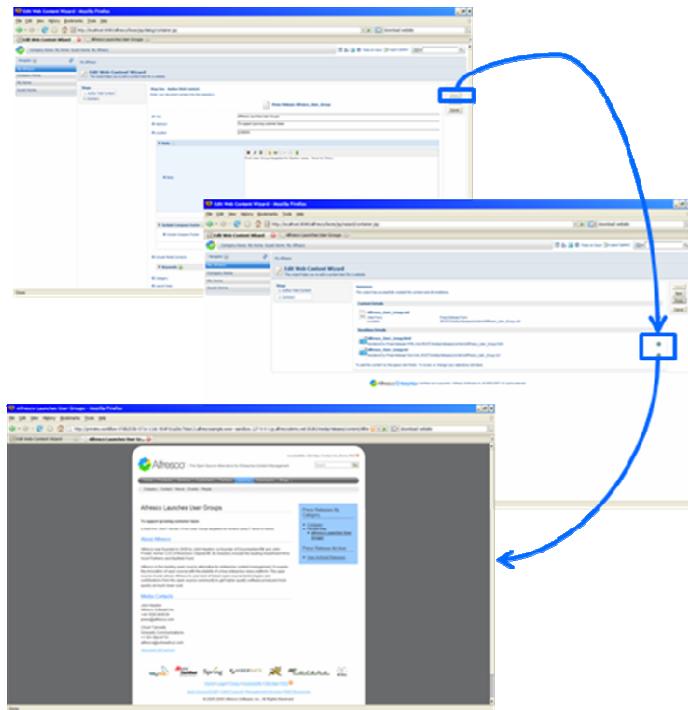
Now, click on **My Alfresco** to return to the Admin user's dashboard. Note that instead of having a clear task list in your dashboard that instead there is a new assigned task, *Verify Broken Links*.

Open the **Task Dialog** for this task. In the **Task Dialog**, note the additional section, **Links Validation**, and note the summary statistics: of 3 files submitted, there are 4 broken links in 2 of those files. Those broken links are in the assets generated from our source XML, the HTML and plain text renditions. Click on **View Links Validation Report** to inspect further the source of the problem.

In the **Links Validation Report**, see that there is 1 generated file with two renditions, each of which has broken links. Now that we've confirmed the source of the error, we can immediately fix it by clicking on the **EDIT** icon to the right-hand side of our XML asset. After launching directly from the **Link Validation Report** into the Web Form for modifying the XML, select the hyperlinked text "Boston". We're about to fix this link.

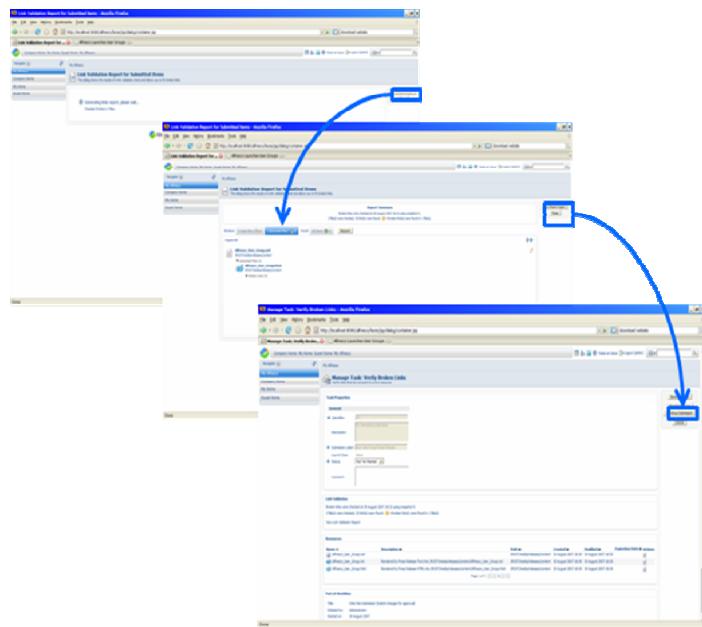


Once the link is selected, click on the **Unlink** icon in the TinyMCE rich-text editor. Confirm that the word is no longer linked by noting that it is no longer blue and underlined. Click **Next** and in the **Summary** screen, double-check your work by clicking **Preview** and noting in the generated page that “Boston” is no longer hyperlinked to our fictitious, non-existent external reference.



Once confirmed, close the **Preview** window and click **Finish** in the **Summary** screen of the **Edit Content Wizard**. Notice that you are returned back to the **Link Validation Report**. Here, click on the **Recount** action to re-check your submitted items for broken reference. In the resulting report, notice that as a result of your edit, there are now only 4 broken links, as 2 have just now been corrected. Progress!

Now that we've fixed that error, let's temporarily bypass the remaining 4 broken links. These links are automatically generated via our templates, and these links we'll correct later by updating our templates and using our **Regenerate** action. This action will regenerate all web pages based on those templates so that we can easily apply a global fix for all broken references. For our purposes now, we've done all we can do to ensure the quality of our content. We'll leave the site regeneration to the Web Developer responsible for the templates. Click **Close** in the **Link Validation Report** and when returned to the **Task Dialog** for the **Verify Broken Links** task, click **Continue Submission**.



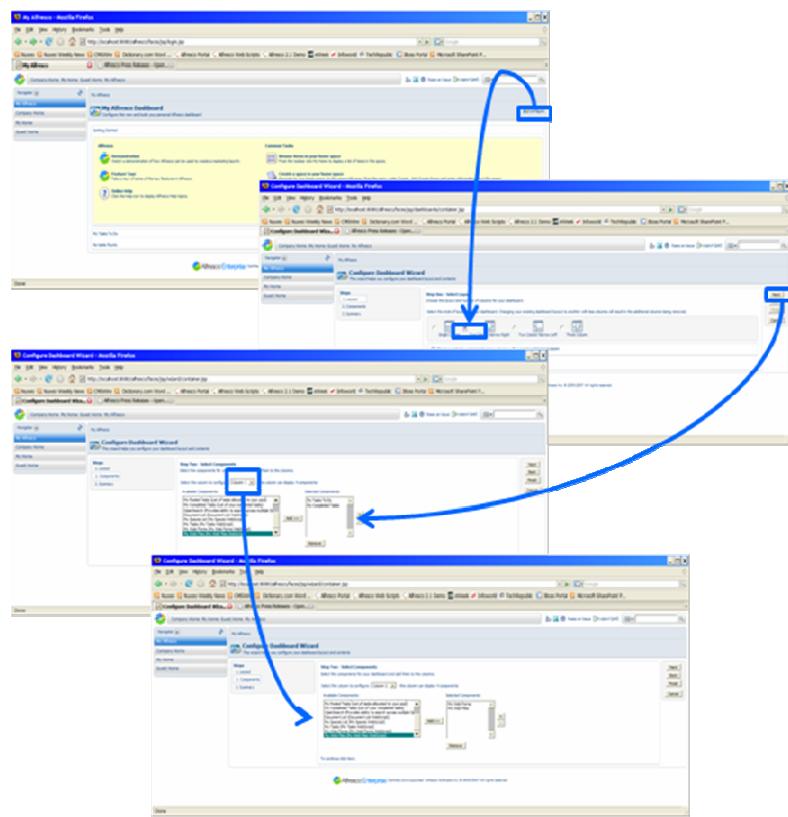
We've now continued with our normal workflow approval process configured via the **Create Web Project Wizard** for press releases. Please do note, however, that when reviewing this press release from their Task List, our press release reviewers Mark and Lee will each have access to the **Link Validation** report. Just in case we weren't honest, both Mark and Lee can see in the **Workflow History** in the **Task Dialog** that broken links were identified, a task assigned for the Admin user to verify and fix those links, and the content re-submitted for approval. Knowing this, both Mark and Lee can pay extra attention when previewing the generated web pages and look directly at the **Links Validation** report to confirm all necessary fixes. And if not – either one of them can reject. As we saw earlier when rejecting submitted content, this will mean a second edit and review cycle (and potentially third or fourth) for the Admin user, Mark, and Lee.

Content Contribution: Alfresco Dashlets

Now that we've gone through some scenarios showcasing publishing, links checking, launch, expiration, deployment, and rollback, let's look at that same publishing process again – but this time as our Content Contributor Pat, who is primarily someone focus on creating and collaborating on business documents and prefers to have little knowledge about web publishing and web projects.

Logout as the Admin user and log in as Pat. In Pat's Alfresco Dashboard, click on the **Configure** action in the upper-right-hand side of your screen. We are going to add to Pat's dashboard two lightweight web publishing dashlets. With these dashlets, Pat will have ready access to all services he needs to publish content to a website – all without having to navigate into a Web Project or needing to learn all the details about how content for the Web is staged and managed.

In the **Configure Dashboard Wizard**, select the second option, *Two Column Narrow Right*, as Pat's layout preference. Then, click **Next**. In the **Select Components** screen, select *Getting Started* in the right-hand scrolling list and click on the **Remove** button. Then, from the left-hand scrolling list, select **Completed Tasks**. Click **Add**. When done, select *Column Two* from the drop-down list above both scrolling lists. From the left-hand side list, select the *My Web Forms* component and click **Add**. Then, select the *My Web Files* component and click **Add** a second time. Confirm that both components are listed in the right-hand side scrolling list.

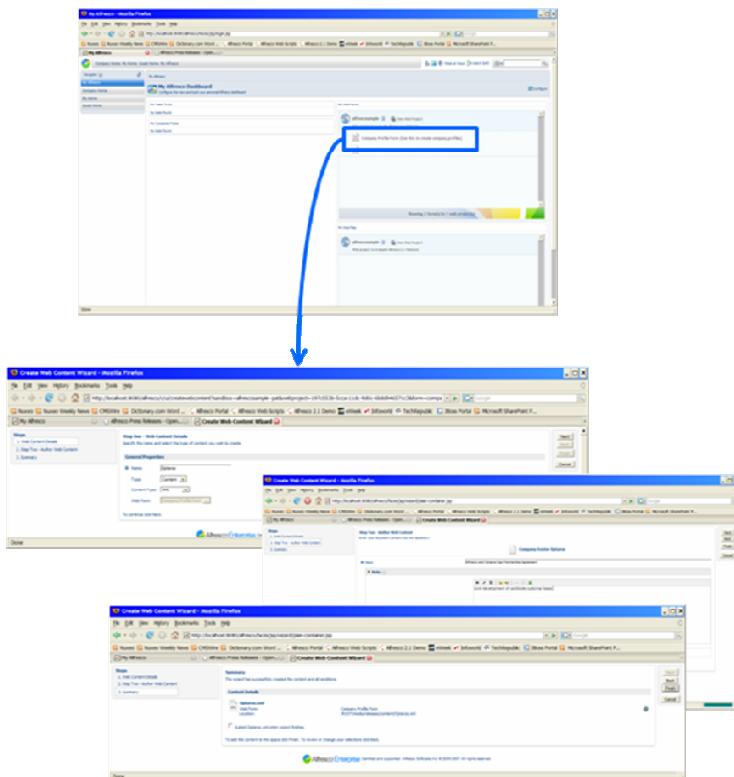


Click **Finish** and return to Pat's dashboard, noting the new layout and two new web publishing dashlets. With these dashlets, Pat will be able to create and manage his web content directly from his Alfresco dashboard. As an occasional web contributor, letting Pat work directly from his

dashboard along with his task list cuts down the number of things Pat needs to learn – making his life easier and making rollout a cinch.

Now, let's explore publishing using these dashlets. In the first dashlet, *My Web Forms*, hover over the website, *alfrescosample*. This is the only website Pat has authorization to publish to, and hence is the only website Pat sees in his dashlets. Although there may be other websites managed within Alfresco, because Pat is not a contributor to those, they are filtered from his view via this dashlet. This is not only a convenience feature for Pat, but also a security feature for IT.

Click on the name of the web form *Company Press Release*. Note that the **Create Content Wizard** comes up, allowing Pat to enter a fill and fill in details for this content item. Fill in any sample data you wish (as you are getting a handle on Alfresco web publishing). In the **Summary** screen, ensure that the **Submit** option is not selected and click **Finish**.



Now back at Pat's dashboard, let's look at the *My Web Files* dashlet. Since Pat completed the previous Web Form to save work-in-progress as he moved on to another, non-web publishing task, Pat will want to return back to that content item, finalize edits, and publish to the site.

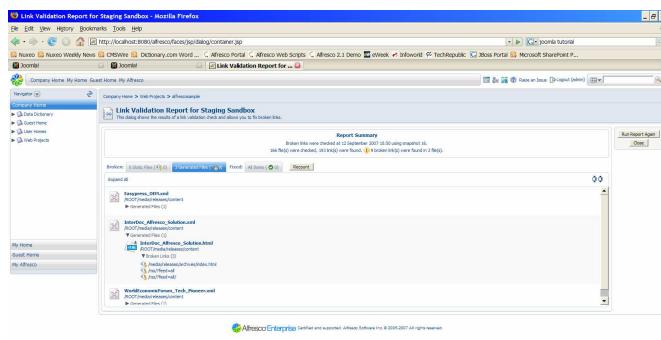
Click on the *alfrescosample* header in the *My Web Files* dashlet. See in the expand resource list that Pat has one modified, locked asset. Also notice the **Edit** action available to Pat next to the resource. To finish Pat's work, click **Edit** and walk through the **Edit Content Wizard** to finish the authoring process. When complete, check the **Submit** option at the **Summary** screen and then **Finish**. Complete the items in the **Submit Dialog**, potentially setting a launch date for Pat's changes, and then click **OK**. Pat's changes are now in queue!

Site Regeneration: Updating Templates

Now that we've gone through various publishing scenarios, let's turn to two advanced user topics: working with the virtual file-system interface (CIFS) and with Web Forms in the Data Dictionary.

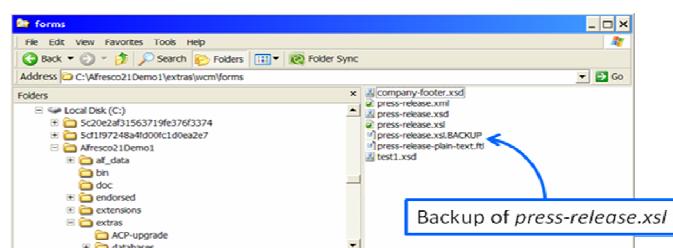
We'll first look at working with Web Forms. Recalling from our earlier look at the **Links Validation Report**, remember that our page template was generating our HTML page with broken links. We are going to correct that template now as a developer, and then update all those web pages based on this template.

First, let's re-run our report by clicking on **Check Links** next to the Staging sandbox in our sample web project. Click on the **Generated Files** link, and expand anyone of the listed XML's to see the broken links in our generated HTML pages:

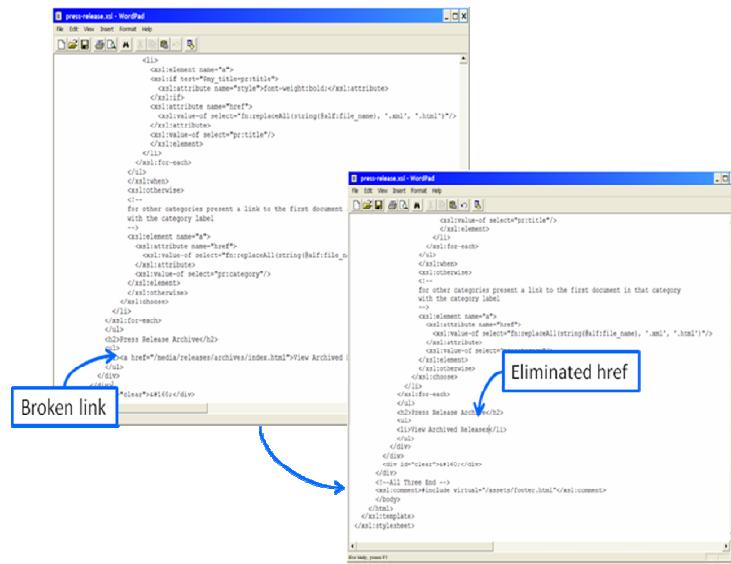


Now, let's go an update our XSL that we use to generate this HTML. In that XSL, we are going to clean-up that code which is generating these links.

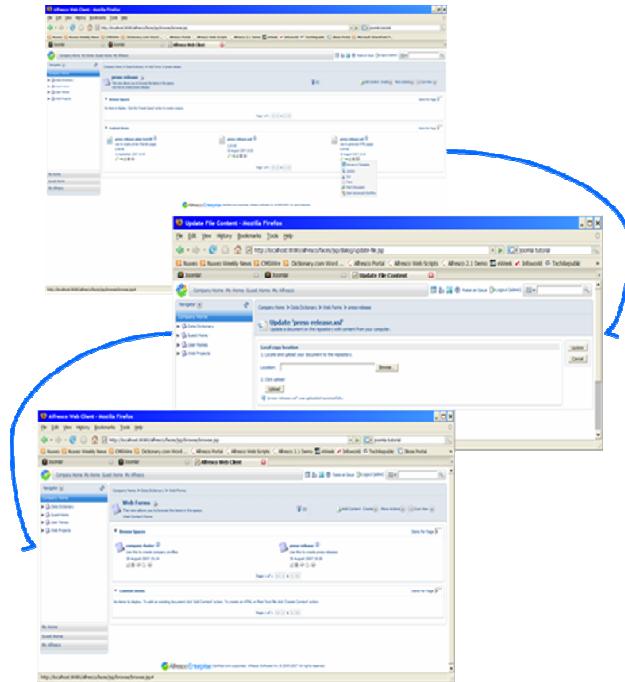
Navigate into your `/extras/wcm/forms` directory and first make a backup copy of the file `press-release.xsl`:



Once you've made a copy of the existing file, let's open up the original. Do a search on the text string `/archive/index.html`. Note the `<a href>` to our missing file. Now, delete the `<a href>` and `` tags as in the following example:



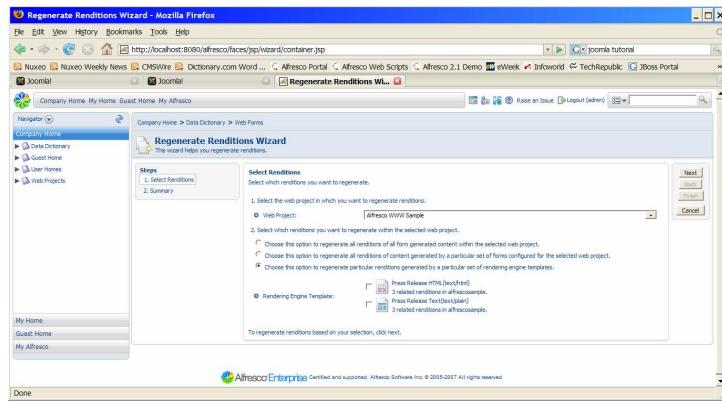
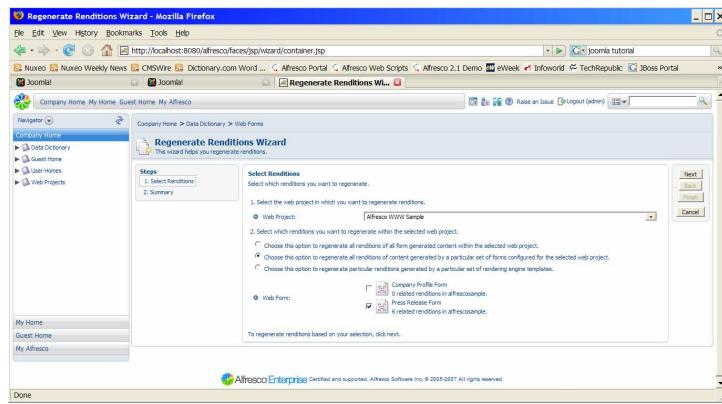
Once saved, turn back to the Alfresco Web Client. Make certain you are logged in as Admin (as our Web Developer). Navigate into */Data Dictionary/Web Forms/press-release*. Under the *More Actions* icon, select the **Update** action. In the **Update Dialog**, click **Browse** and select our updated file *press-release.xls* in the navigation window. Once selected, click **Upload** and then click **Update**. Now that we've got our new template in the **Data Dictionary**, click on the **Web Forms** link in the navigation bar and return to the space containing the entire list of Web Forms in our system. Next to the *press-release* Web Form space.



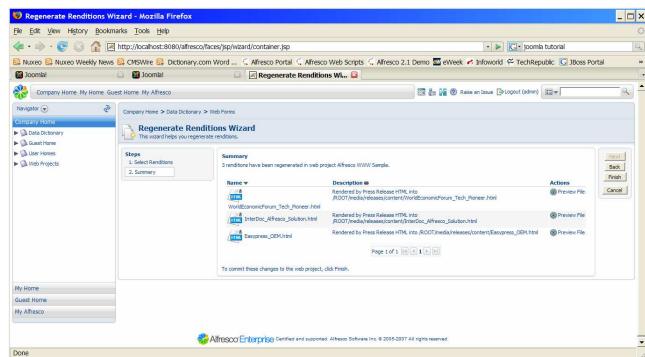
Now, click on the **Regenerate** icon next to *press-release*. Notice that we have three options:

- For a selected Web Project, regenerate all renditions associated with all Web Form
- For a selected Web Project, regenerate all renditions associated with a specific Web Form
- For a selected Web Project, regenerate a specific rendition associated with a specific Web Form

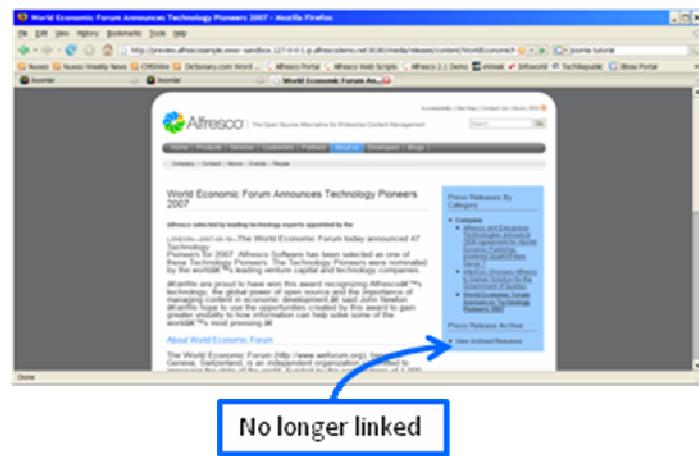
Click on each of these last two options. See in the first case that you can choose to regenerate all renditions generated from either the Company Profile or Press Release Web Forms. Notice that since the Company Profile Web Form has no templates associated with it, it is not selected. In the second case see that for the Press Release Web Form, you can choose whether to regenerate either the plain text or HTML renditions of your press releases or both.



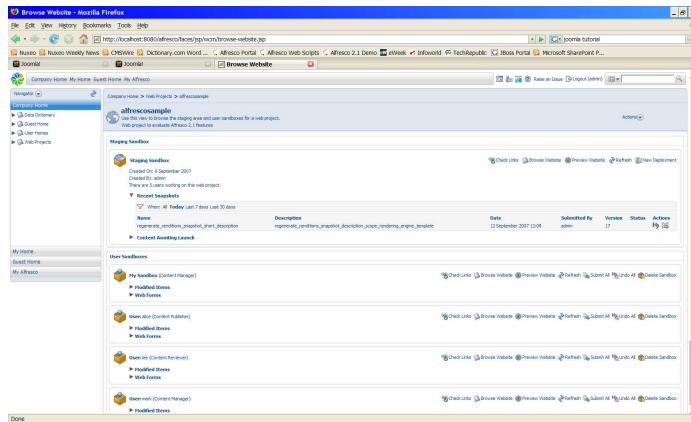
Since we just have one updated template, let's choose this third option and select the Press Release HTML template. Click **Next**.



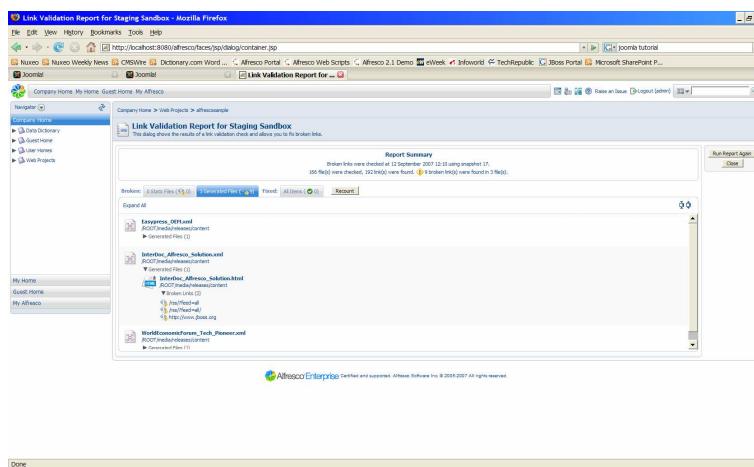
See that our template has been re-applied to our three generated HTML renditions. Click on the **Preview File** action and see in the re-generated page that **View Archived Press Releases** is no longer hyperlinked. After previewing, click **Finish**.



Returning to your Web Project, now note that you have an auto-generated submission to and snapshot of Staging with your updated web pages:



Now, click on **Check Links** again and select the **Generated Files** tab. Look once again at the expanded list of broken links in your generated HTML page. Notice that we no longer find the broken link to `/media/archives/index.html`.

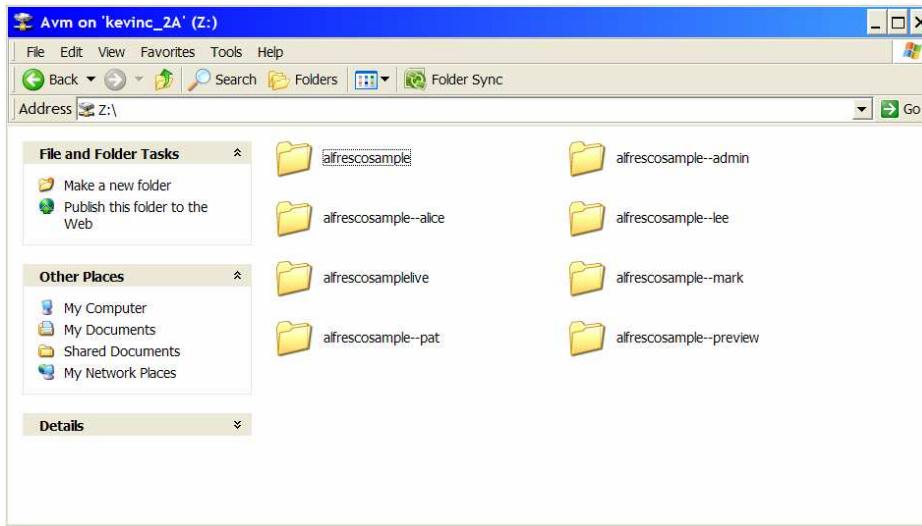


Now, to fully correct our both our hand-crafted HTML pages and our Web Form-generated web pages, let's turn to our virtual file-system interface and correct the headers and footer HTML fragments we are including via our XSL to fix all the remaining broken links on our site.

Virtual File System Interface

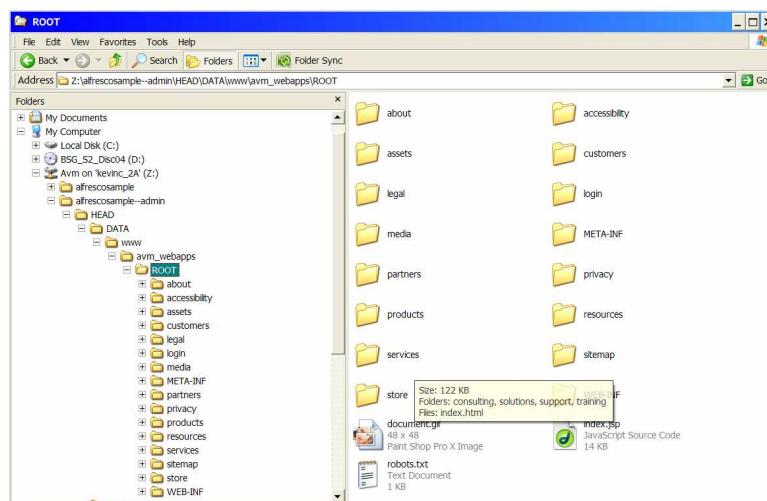
In Windows Explorer, let's map a network drive to our Alfresco virtual file-system interface. Under the **Tools** menu, select **Map Network Drive** and in the dialog select an open drive letter (for example, the Z: drive). Click on the *different user name* link and enter the value "admin" for both user name and password. In the *Folder* field, enter the value `\your_machinename\AVM` (please note that capital letter 'A' appended to the name of your machine name).

You'll see in your Explorer view all of our available sandboxes:

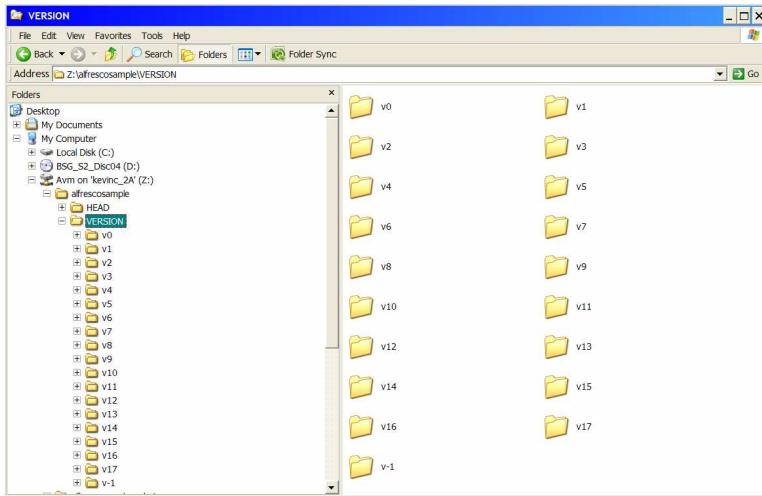


In this view, notice that we have one sandbox, *alfrescosample*. This corresponds to Staging. Notice that we also have sandboxes for each of our users, for example Alice's sandbox, *alfrescosample--alice*. Notice too that we have an additional sandbox, *alfrescosamplelive*. This is the sandbox associated with the "live" website that we are serving to our external audiences. This is the content repository we automatically created and keep updated using our deployment service.

Now, let's navigate into the Admin sandbox:

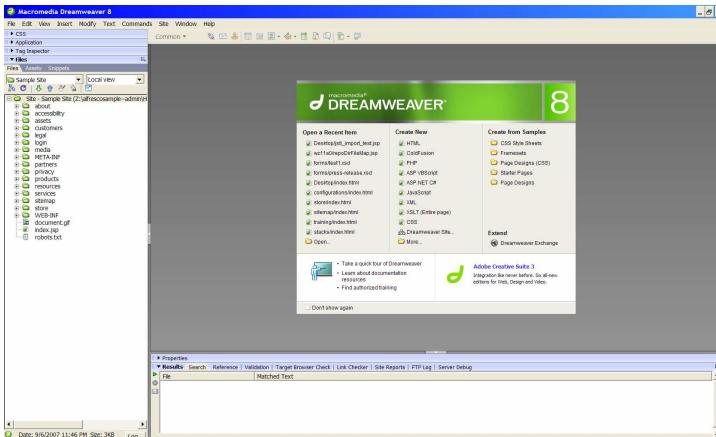


When navigating into the Admin sandbox, notice you will first open the *HEAD* folder (this is the current working version of the contents of your sandbox. As each sandbox is it's own repository, each sandbox could have it's own set of historical snapshots. In our current web client, however, we do not allow users to snapshot their own personal sandbox. This is done for Staging only. If you want to see historical snapshots, however, you can navigate into Staging and see the entire list of all snapshots we've taken by going into the *VERSIONS* folder:



Now, let's return back to the ROOT web app of the Admin sandbox (*/HEAD/DATA/www/avm_webapps/ROOT*). We are going to launch our editing tool of choice, in this case DreamWeaver, and map this location as our website root in order to making changes to our code and hand-crafted HTML pages.

If you have Macromedia DreamWeaver installed, launch this application and create a new site, *Sample Site*, using the **New Site** command under the **Site** menu. After setting your doc root to Z:\alfrescosample—admin\HEAD\DATA\www\avm_webapps\ROOT and choosing the option to set your links relative to a document, note how DreamWeaver automatically reads each file in the repository via CIFS to build an index of content and return a directory navigator for the contents of your Web Project:

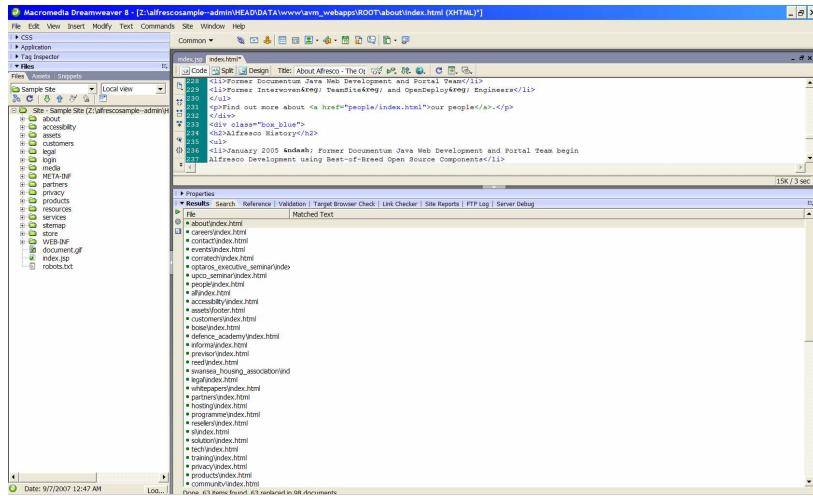


First, we are going to remove all broken links to www.jboss.org. To do this, use DreamWeaver's Find and Replace command. In the **Find** field, type the following:

```
<a href="http://www.jboss.org"></a>
```

In the **Replace** field, leave the value blank. Then click **Replace All**.

Notice how DreamWeaver can seamless access each file via CIFS and remove this link. See the large number of files DreamWeaver updates:



Going back to our Web Client, navigate into your Web Project and expand the **Modified Items** list for the Admin user. See this same list of files showing up as modified, previewable, and submittable in the GUI:

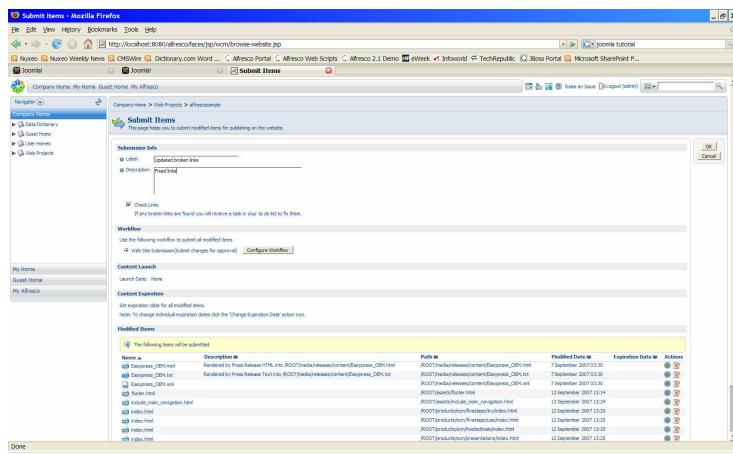
The screenshot shows the Alfresco Web Client interface. On the left, there is a navigation sidebar with links like 'Company Home', 'My Home', 'Guest Home', and 'My Alfresco'. The main content area shows a 'Modified Items' list for a project named 'alfrescoexample'. The list displays a table with columns for 'Created Date', 'Modified Date', 'Size', and 'Actions'. Each row represents a modified file, such as 'index.html', with its specific details like date, size, and file path.

You will notice the items modified via CIFS do not have a lock automatically taken on them. It is presumed that users working via CIFS are power users (Developers and Designers) and therefore are allowed to make concurrent modifications. This does leave open the possibility of conflicting edits. In the future, the GUI will support automatically conflicts detection and asset merging at time of submit. For now, any submit of an unlocked item modified via CIFS will simply override any conflicting change when promoted. Because all checked in content is immediately available to all sandboxes, the time window for a potential conflict and override is small. Please do use with care, however, and remember that you can revert a file or rollback an entire submission if needed.

Now, let's do the same thing again for each of the following six text strings:

1. /rss.php?feed=all"
 2. /rss/?feed=coverage"
 3. /rss/?feed=all"
 4. /about/careers/jobs
 5. /about/careers/why
 6. /login/?returnpath=/about/careers/

When complete, return back to your modified items list and preview the Admin user's view of the website in her sandbox. When done, click **Submit All** and promote your changes to Staging. When promoting, keep the **Check Links** option selected in the **Submit Dialog**.



Viola! Our website is now up-to-date and complete. New company profiles, new press releases, and free of any and all broken reference. Job well done!

Summary

That concludes our walk-through of the Web Content Management capabilities of Alfresco 2.1. In this walk-through, we saw the following:

- XML content authoring
- Multi-channel content publishing
- Sandboxed content development
- Content staging, promotion, workflow, and snapshots
- Virtualization and in-context preview for both static and dynamic web pages
- Content deployment
- Content launch and expiration
- Links management
- Simple web publishing through dashlets

Thanks for evaluating Alfresco 2.1 Web Content Management. To learn more about the capabilities provided in this release, including further details on our roadmap, please visit our wiki at the following location:

http://wiki.alfresco.com/wiki/New_Web_Content_Management_Plan

Once again, thanks again for evaluating Alfresco 2.1 Web Content Management.