

EXO PLATFORM

VERSION 2.0

A complete guide to
Enterprise Content Management technology

Table of Contents

What is ECM?	3
I) Concept	4
a) JCR-JSR 170	4
b) WebDav	4
c) Draft, Production, and Backup	5
d) Versioning	6
e) Locking	6
f) Workspace	7
g) Node Lifecycle	7
II) Features	8
a) Capturing	8
b) Storing	8
c) Managing	8
d) Publishing	8
e) Backing up	8
ECM Overview	9
A) ECM File Explorer	9
Managing node	9
a) Create a folder	9
b) Add a document	11
c) Upload a document	11
d) Create a ticketing ID	12
View Node	12
a) View References	12
b) View Node Type	13
c) View Permission	14
d) View Properties	14
Manage Node Actions	14
a) Manage Versions	14
b) Manage Categories	16
c) Manage Actions	17
I) Send Mail Action	17
II) Transform Binary to Text Action	18
III) Workflow Action	20
d) Export Node	20
e) Import Node	20
B) ECM Admin	21
Manage Node type	21
Manage Templates	22
Manage Scripts	25
Manage Rules	26
Create Action Type	26
Manage View	29
Content Validation Workflow	30
Browse Category	35

What is ECM?

Whether your company is small or large your company will probably generate large volumes of information everyday to collaborate and communicate with your customers, partners and employees. These information are critical to your company success because it enriches your company productivity, relationship, and efficiency. However, contents are commonly spread across multiple location and does not fall into a structured formats of automated file systems or database. Therefore, it is very time consuming and difficult for you to find relevant content which can result in loss of efficiency and productivity. Enterprise Content Management is a solution that brings new value to improve your operational productivity and efficiency. It allows you to transform unstructured content to structured content through the process of capturing, storing, managing, publishing, and backing up while securely distributing it. eXo platform ECM portlet gives you a portal solution that can help you achieve these processes and it is designed so that you can leverage your business content across all formats for competitive gain. It also provides an environment for employees to share and collaborate digital content as well as delivering a comprehensive unified solution with rich functionalities.

eXo platform ECM 1.0 Concepts and Features

1) Concept

We will give you a brief concept behind ECM portlet technology such as what is JCR-JSR 170, WebDav, draft-production-backup, versioning, locking, workspaces, and node lifecycle. These concepts will help you better understand ECM portlet in order for you to apply it easily to your work.

a) JCR-JSR 170

JCR stands for Java Content Repository which is a specification that normalizes the storing of data in an abstract way. These repositories are composed of workspaces that can store content in a hierarchical structure. JCR specification has services that enable read and write level as well as versioning and locking. eXo platform has successfully implemented JCR that can optimize connection to all databases such as Oracle, DB2, MySQL as shown in the figure 1 below.

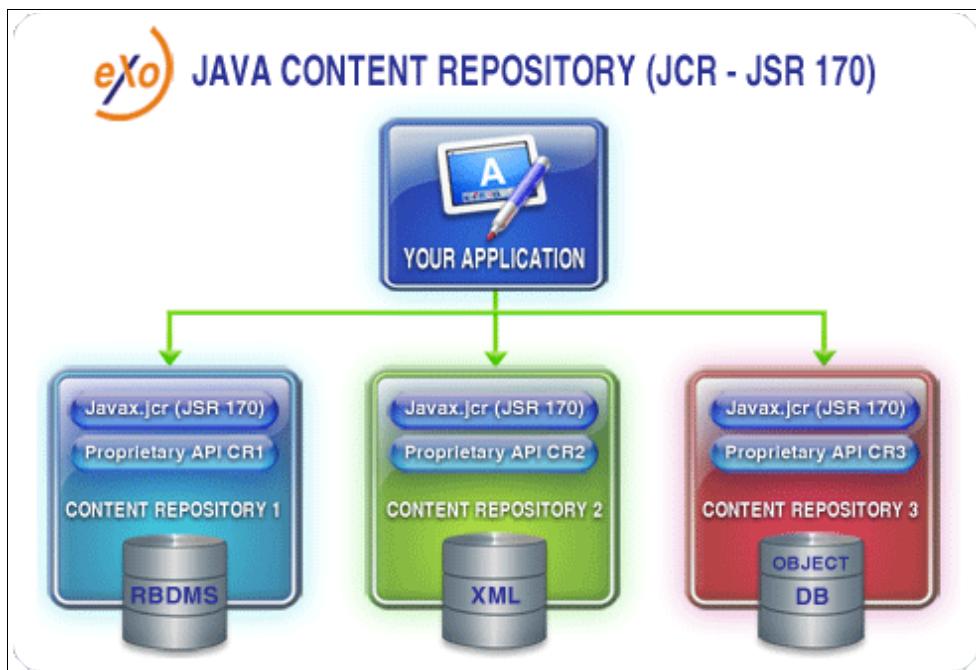


Figure 1 Java Content Repository

b) WebDav

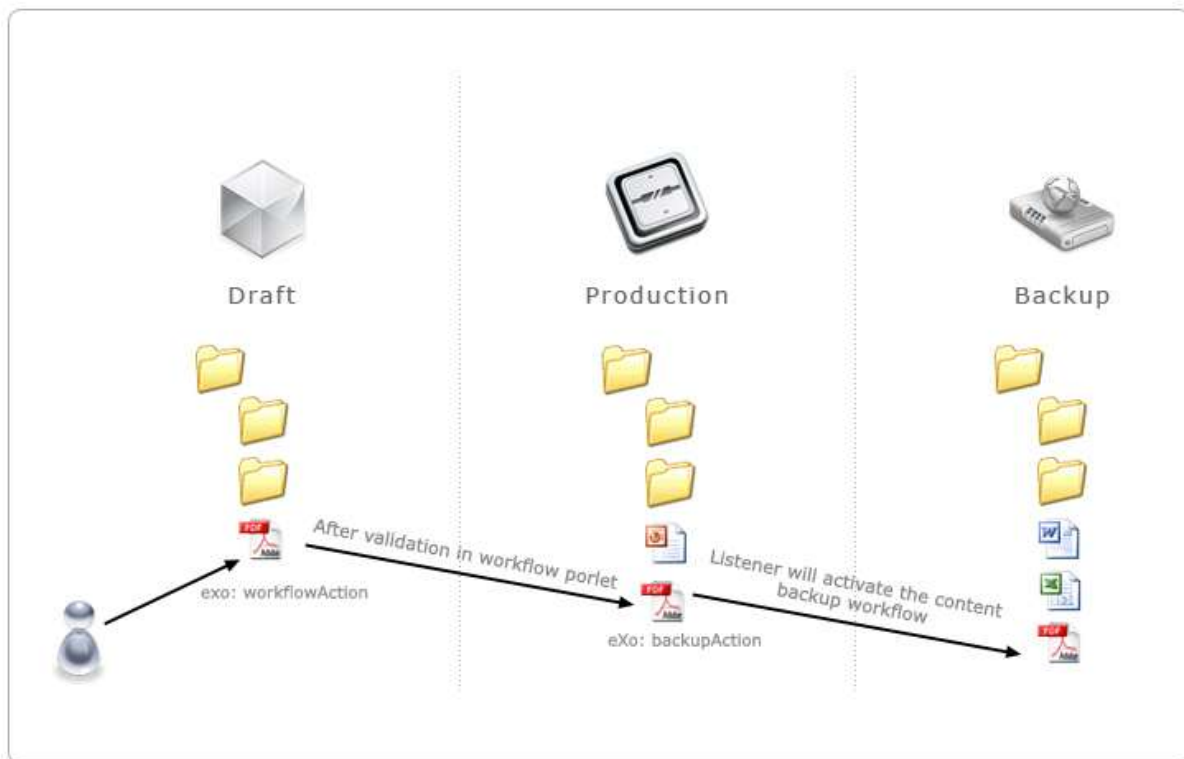
WebDAV stands for "Web-based Distributed Authoring and Versioning". It is a set of extensions to the HTTP protocol which allows users to collaboratively edit and manage files on remote web servers. A classic example of WebDav is when a writer has document and upload that document to the JCR workspace.

c) Draft, Production, and Backup

Here is a simple example of how online text publication works. Within the draft workspace, an author has several ways to create a document. He can create a draft document offline and then uploads it to the server through WebDav or he may use a document template in ECM to create his draft document. The author can then manage the content versioning before sending it to the production workspace using the workflowAction script.

In between the draft and the production process is the evaluation workflow where the authorized validator make the selection to refuse, delegate, approve, or disapprove a document. If a document is approved it will be moved to the production workspace to be published.

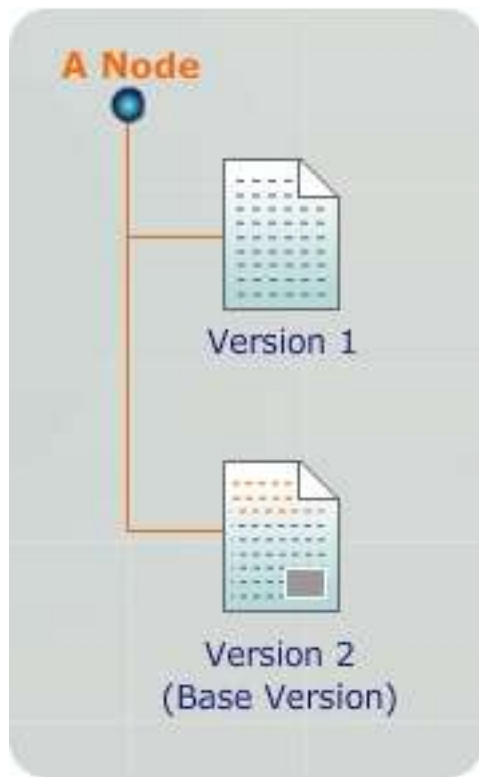
Once the published date expired a script is generated to move this document from the production to the backup workspace. The image below shows the different stages of a document from a draft to production then to the backup workspace.



d) Versioning

Versioning means that at any given time the node's state can be saved for possible future recovery and the action of saving a new version is called check in. A workspace may

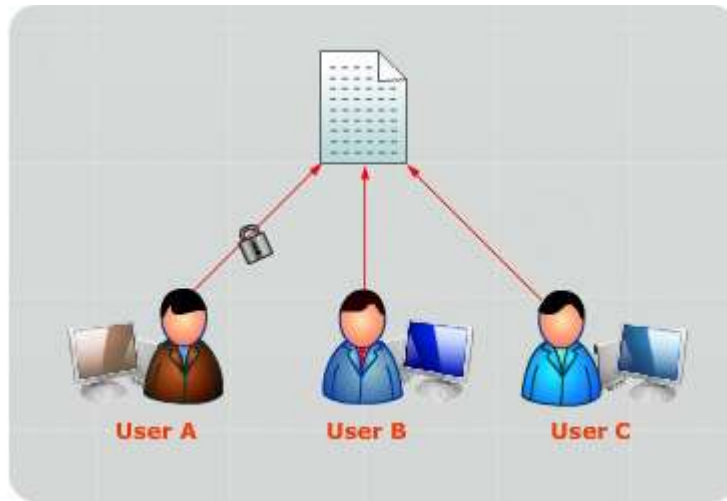
contain both versionable and nonversionable nodes. A version exist as part of a version history graph that describes the predecessor/successor relations among version of a particular versionable node as shown in the figure below.



Base version refers to the actual version of your document and every time you check in the base version increase.

e) Locking

Locking allows a user to temporarily lock nodes in order to prevent other users from changing them. If a lock applies to a particular node that node cannot be changed by anyone except the user who is the token holder for that lock. For example, there are three users viewing the same document from different locations and user A decide to lock the document for editing purposes. User B and user C can still view the document but cannot edit it until user A unlock it. The picture below depicts this example.



f) Workspace

A content repository is composed of a number of workspaces. Each workspace contains a single rooted tree item. In the case of eXo platform, the ECM file systems workspace contains draft, production, backup, and digital-assets.



Figure 1 ECM workspaces

g) Node Lifecycle

Node lifecycle refers to read, add, and remove of a node. It has to do with the state of the node in relation with an action, for example, if the action is sent mail and the lifecycle is “add” then every time a node is added an email is sent.

II) Features

Introducing the next-generation platform for Enterprise Content Management that provides an Open Source alternatives relying on open standard to proprietary solution. eXo ECM 1.0 focuses on content capturing, storing, managing, publishing, and backing up.

a) Capturing

eXo ECM provides several ways to submit structured and unstructured content into a repository such as importing xml format, submit via web forms, and upload content using WebDav.

b) Storing

ECM portlet relies on JCR implementation to store content to a database and provide several other back end implementation that can act as connectors to other ECM or data warehouses. Those contents will only be accessible through the normalized JCR API. Searching, locking, versioning as well as security functionalities are part of the store ECM layer.

c) Managing

eXo has a set of powerful portlets to view, create, move and monitor structured and unstructured document. It also introduces the concept of nodes lifecycle to manage advance document validation operations as well as ordering document publication and backup. With the concept and implementation of portal community config, it is easy for ECM portlet to utilize the power of collaboration and document sharing among the same groups. The lifecycle comprises of three items: read, add, and remove.

d) Publishing

eXo ECM provides a portlet that references to the JCR directories and display content in the publish directory. The portlet will extract the children and render them using available templates. Other ways to publish content is through email action concept and it is possible to create a XML file out of a document tree and send it.

e) Backing up

Critical documents and information must be preserved in a safe place for later use, eXo ECM provides a backup workspace that builds on top of a different data source and by populating it using the JCR action concept to insure data preservation.

ECM Overview

eXo platform ECM environment has two main portlets: ECM file explorer and ECM admin portlet. ECM file explorer lets you create, upload, and manage your content draft, production, and backup whereas ECM admin lets you create templates, scripts, rules, action type, and add/edit node type.

We will show you the ECM file explorer functionalities first and later we show you how you can use ECM admin to manage your code.

A) ECM File Explorer

The ECM file explorer allows you to manage node, view node details, and manage node actions. Managing node consist of allowing you to create a node, edit that node, and send your node via email. View node details lets you view your node references, type, permission, and properties. Manage node actions let you manage your node versioning, categories, actions, and import/export of your node.

Managing node

a) Create a folder

The ECM file explorer is like any other known file explorer, it allows you to create folders and lets you organize folders into meaningful repositories. To create a folder click on the add folder icon and enter the folder name. There are two types of node type you must choose from: nt:unstructured and nt:folder. Choosing nt:unstructured allows you to add structured and unstructured document whereas choosing nt:folder doesn't allow you to add unstructured document. Another difference is that you can add properties to the unstructured folder. Subfolder in nt:unstructured folder can be of nt:folder or nt:unstructured but subfolder in nt:folder can't accept a nt:unstructured. The figure 1 below shows two folders: an unstructured and structured folder.

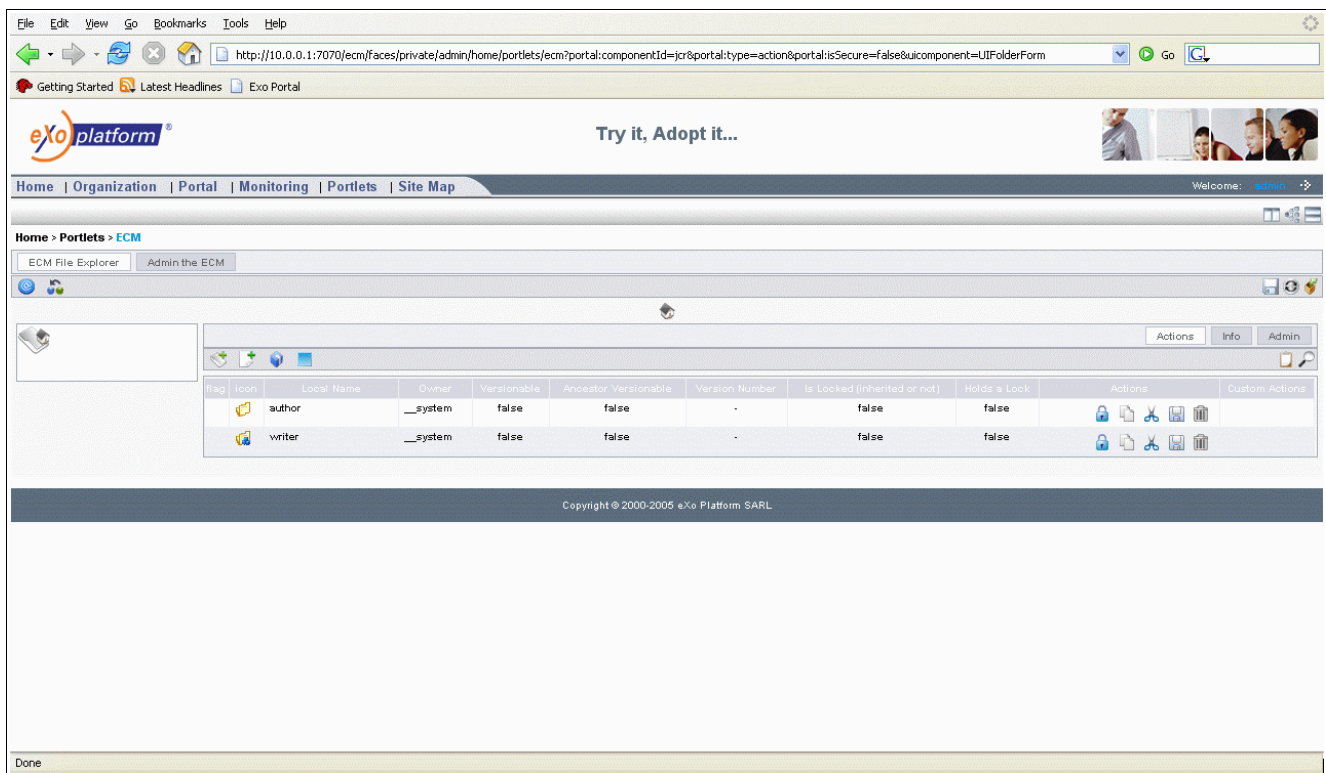


Figure 1 nt:structured is the folder writer and nt:folder is the folder author

Same-Name Siblings in unstructured folder allows you to create folder that can share the same name and these folders are grouped together by embedding an array-like notation within the path. This practice stems from the need to allow Xpath-based queries on the repository. However, as opposed to the semantics of Xpath, a name in a content repository path that does not explicitly specify an index implies an index of 1. For example, `/a/b/c` is equivalent to `/a[1]/b[1]/c[1]`. The figure 2 below shows same-name siblings where we have created two folder with the name author.

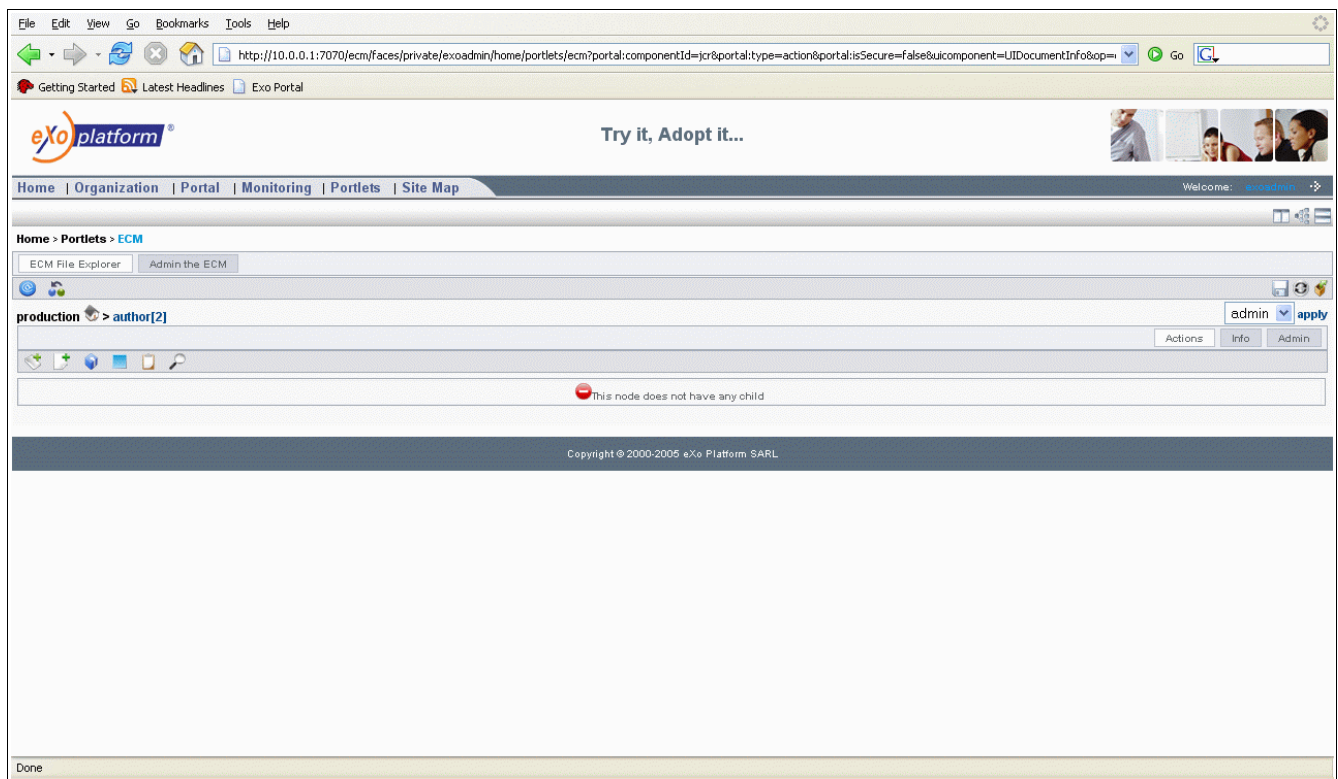


Figure 2 author[2] is a folder with the same-name sibling of author

b) Add a document

Using the add document icon will allow you to add two type of document templates: structured and unstructured. The nt:file is a structured document template that can be created in a structured and unstructured folder whereas the exo:article is a structured document template that can only be created in an unstructured folder. Unstructured document lets you add properties to your document which you couldn't otherwise in a structured folder.

Handling same-name sibling when adding a document for the following case 1) adding a document in a structured folder. ECM portlet will prompt a message notifying name duplication when a folder name already exist. 2) add a document in an unstructured folder. If the a document exist then the new document will override the existing document.

c) Upload a document

There are times when you work offline or someone has sent you a document and you want to bring this document online, you can do this by simply uploading your file. To upload your file -click on the upload icon and locate your file then click save before you exit.

d) Create a ticketing ID

Ticketing ID lets you attach an URL links in your email for your recipient to access

your document via that link. This allows you to share your document without having your recipient access your workspace. To create a ticketing ID, fill out the ticketing ID form and click on the save button to generate an URL link. Once the URL link is generated the email is sent.

Field	Description
Users	Receiver of the document
Permission	Setting group and membership type
Creator	Owner of the document
Access Time	Set the date when the document can be accessed
Access Limit	Limit the number of allowable access.
Description	Text description
Send to mail	Email address of the receiver
Ticket ID	Clicking on the save button after filling out the form will generate a URL link to the document.

Table 1 Ticketing ID form fields

View Node

a) View References

View references let you view the reference of a node in relation with a category. Clicking on the view references icon will display all nodes that links to the categories folder. For example, you created a folder author and use manage categories icon to add this folder to the categories news folder `cms/categories/cms/news`, the author folder is now referencing the folder news. Go to `cms/categories/cms/news` and click on view references icon and you will see `/author/exo:category` as shown in figure 1 below.

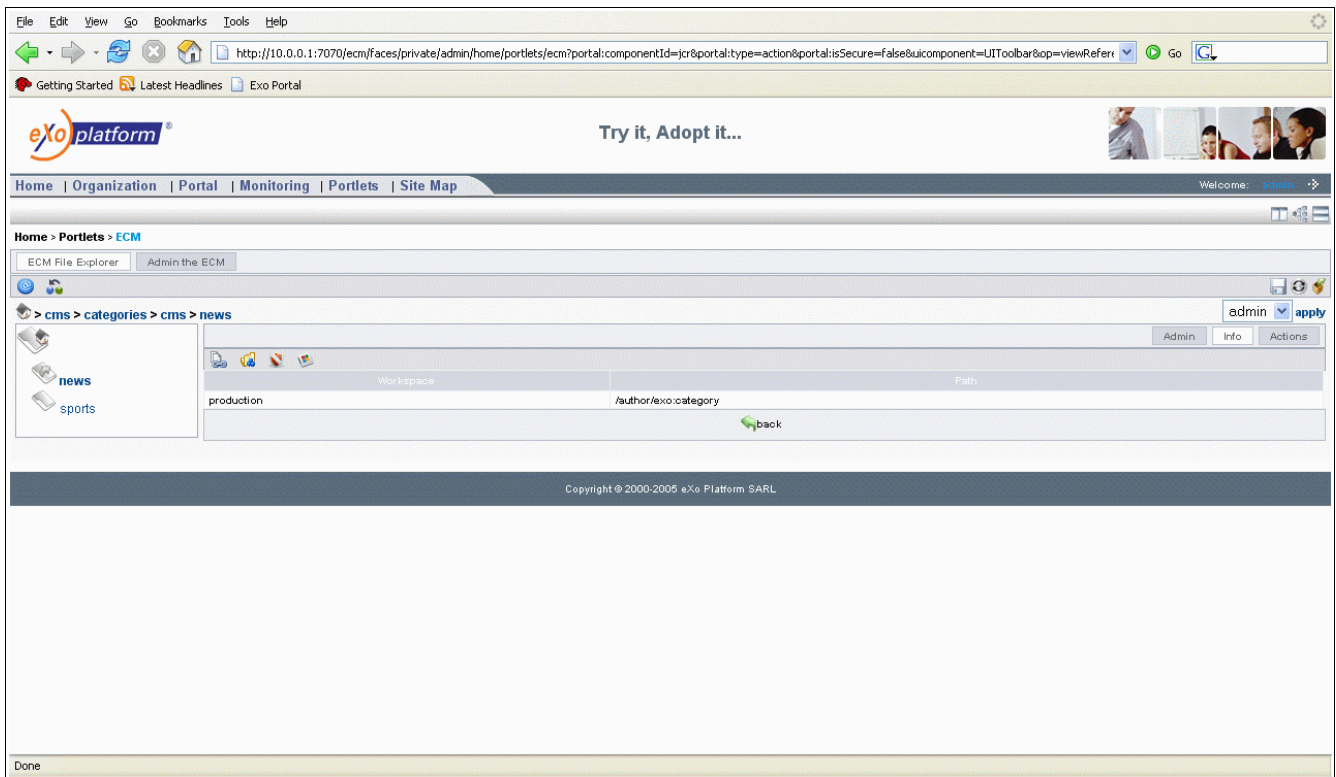


Figure 1 View references

b) View Node Type

View node type gives you detailed description of your node. It will display information for all your nodetype, the node property definition, and the child node definition that belong to your node as shown in figure 1 below.

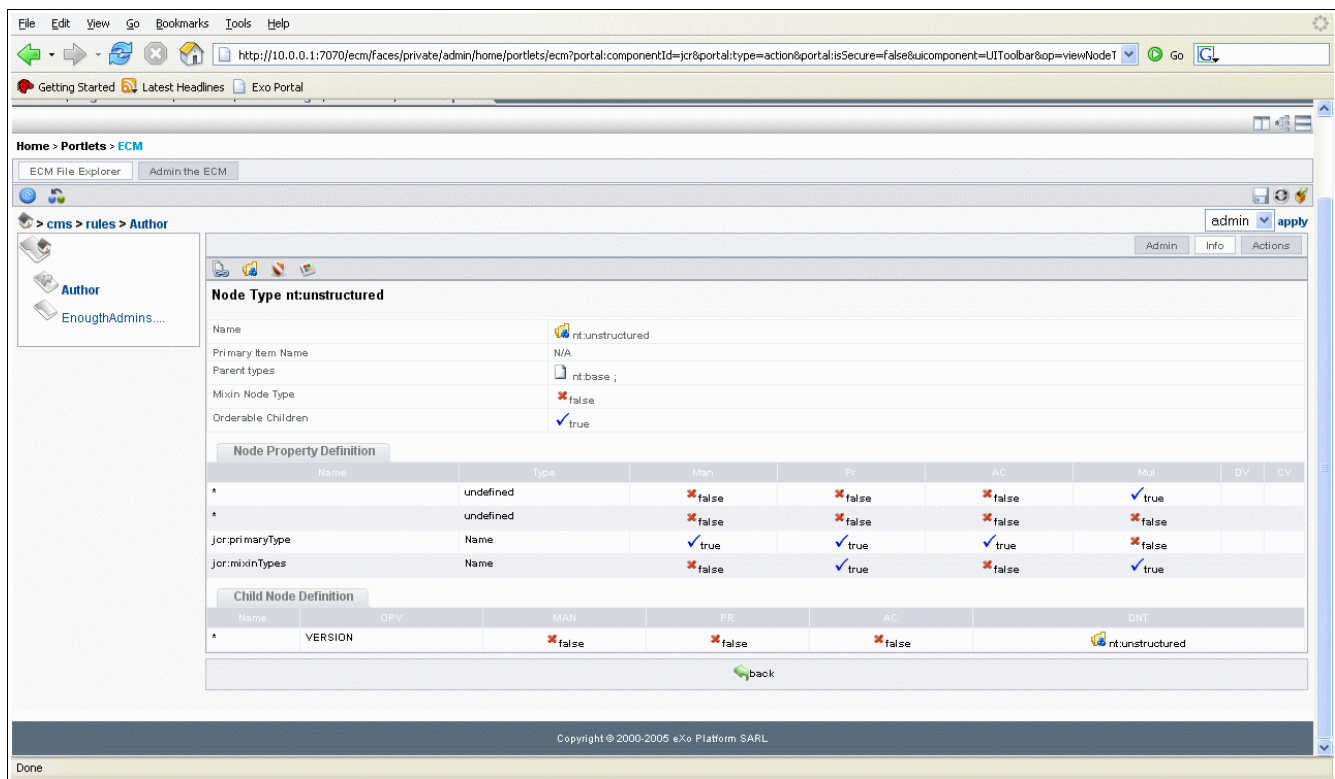


Figure 1 View Node Type

c) View Permission

View permission lets you view all the permission setting as well as letting you add the permission for your node. Permission setting allow and at the same time restrict users accessibility to a node based on group and membership type. Only users that matches the permission setting for a node are allowed to manage that node.

d) View Properties

View properties lets you view and add different properties to your node. Adding properties to your node only works with unstructured and not with structured node. Adding properties to unstructured node will define the node attribute for easier grouping.

Manage Node Actions

a) Manage Versions

There are times when you have different copies of the same document, you need a mechanism to manage the version of each copies. Manage version in eXo ECM lets you manage your document versioning. There are two ways where you can activate the versioning tree: uploading a file and using the manage version icon. Here is an example of how you can initiate your node versioning.

Upload a file

- i) There are times when you work offline and upload your document online, then you have made some changes and upload it online again. Everytime you upload an existing document, the versioning process is activated. You will see that the file versioning number is 1 and a check out icon appeared as shown in figure 1 below. If you click on the check out icon it will become a check in icon. Check in means save as another version therefore clicking on it will increase your versioning number.

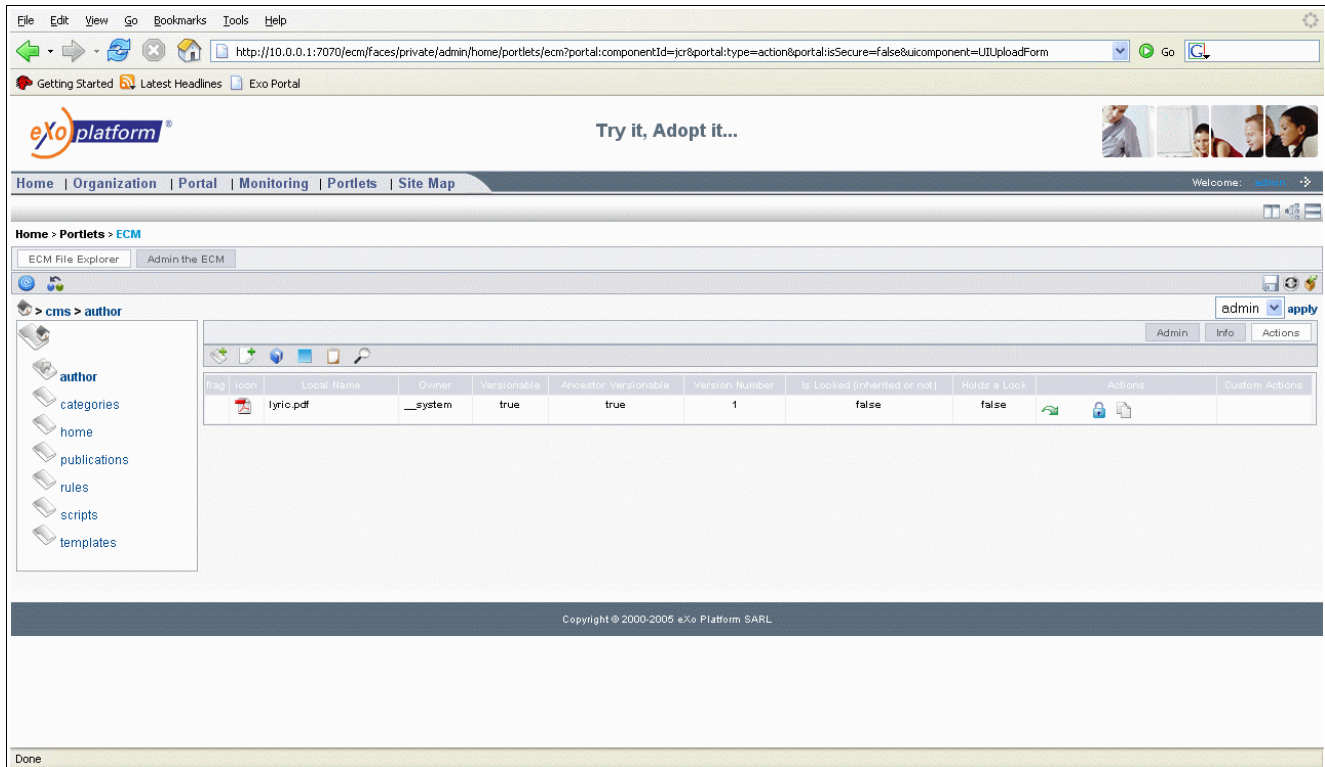


Figure 1 check in version

- i) To view the versioning tree, click on your node file and then click on the manage version icon. You will see the versioning tree where base version is the latest version as shown in figure 2 below.

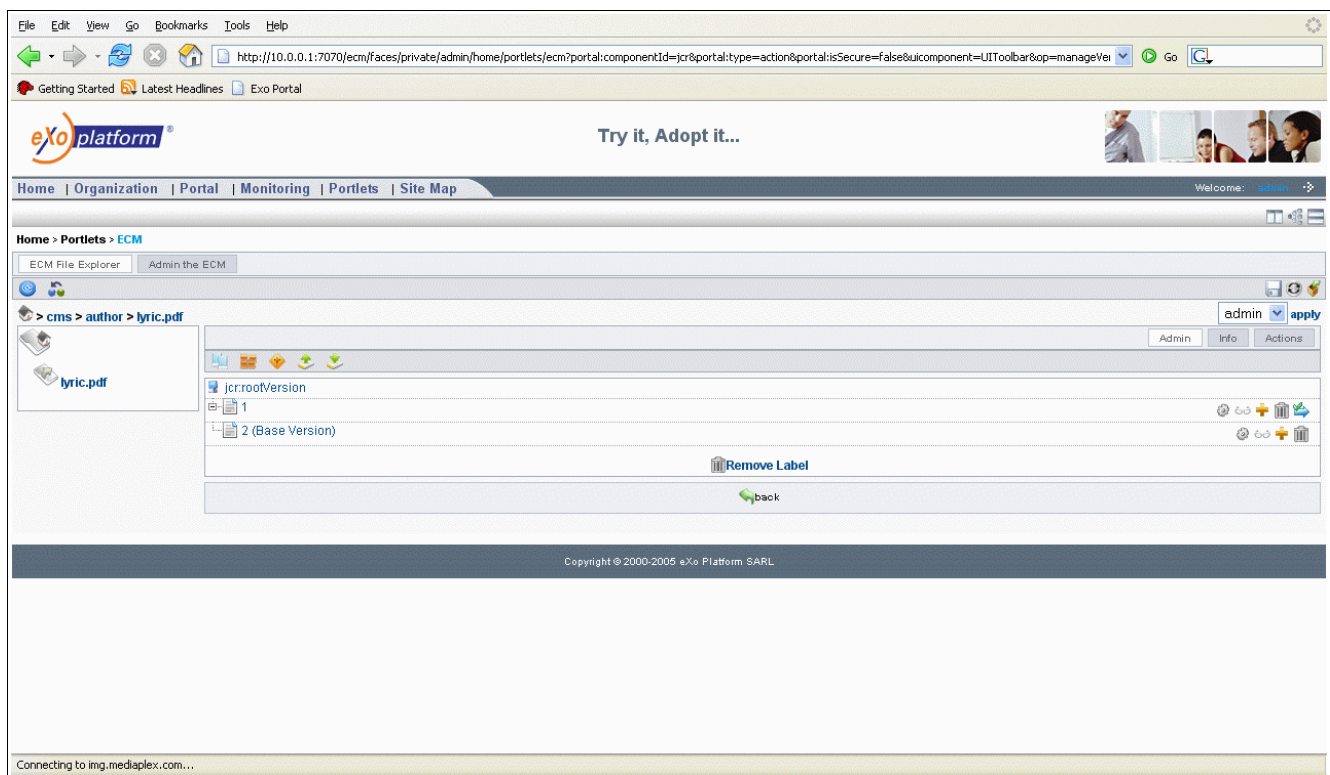


Figure 2 Versioning tree

- ii) To restore base version back to node 1 press on the restore icon next to it.

Versioning Icon

You have a document and you want to improve this document but you want to keep the original version. Activating the versioning tree allows you to work with a base version while maintain several versions of this document for later restoration.

- i) Activating the versioning using the manage version icon as follow. Click on the document you want to start versioning. Then click on manage version icon.
- ii) Click on enable versioning icon. You will be asked to click save to enable versioning. Click save. You should see “jcr:rootVersion” .
- iii) Go back to your document and you will see the check in icon. Click on the edit icon to edit your document then click on check in to save it to the versioning tree.
- iv) Once the check in icon is click the versioning number increase and the check out icon appeared.
- v) Click on the manage version icon and you will see your document tree where base version is your document current version.

b) Manage Categories

Manage categories icon lets you add your node to reference cms categories so that you can view your node in the categories browser. Let's do an example.

- i) Create a folder. In our example, we created a folder author.
- ii) Click on manage categories icon and click on add categories. Select a folder, in our example, we selected the news folder. Once you have selected the folder, your node folder author is now referencing the news folder.
- iii) Navigate to the news folder in your categories browser and you will see your folder displayed as shown in figure 1 below.

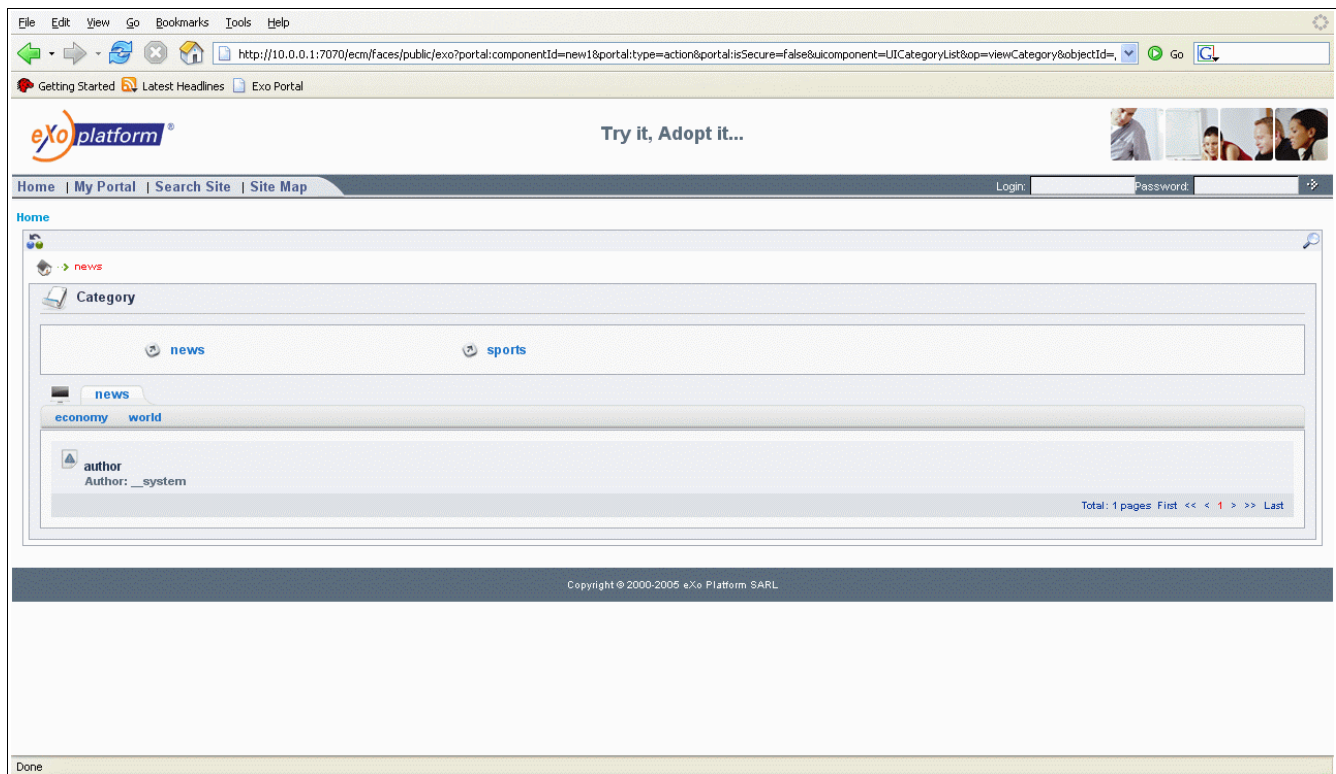


Figure 1 Categories browser

c) Manage Actions

Manage actions let you create action that will either prompt a send mail, activate the workflow portlet content evaluation process, or transform a binary node to text.

1) Send Mail Action

Send mail action executes depending on the lifecycle of your document. For example, if you select the remove lifecycle then every time you remove a document from your folder the send mail action is executed to send an email to your recipient. Let's see this in more details.

- i) Create a folder and click on manage actions icon.

- ii) Select sendMailAction from the drop down list box. Enter the name id, choose the lifecycle type from the drop down list box, enter optional description, and email address of your recipient.
- iii) Go back to your folder and you will see a script action in the custom action column as shown in figure 1 below.

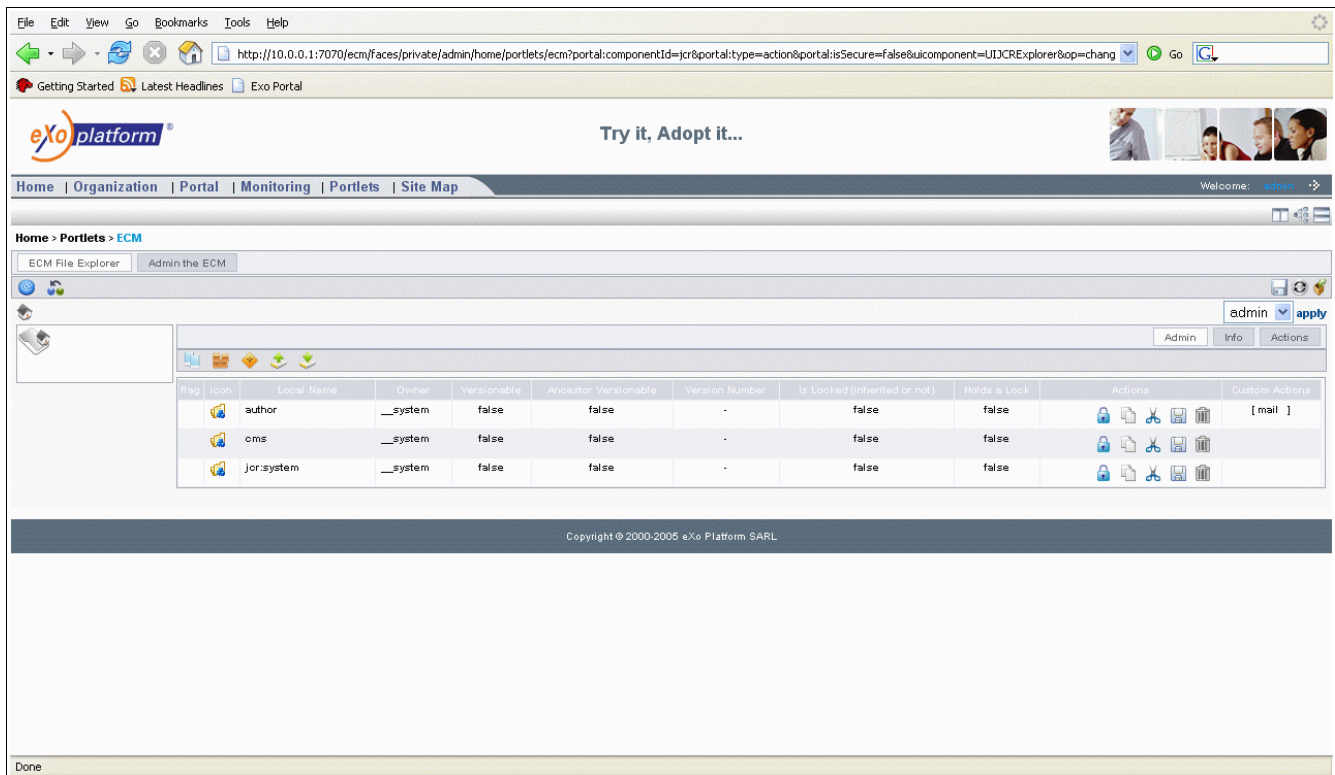


Figure 1 sendMailAction script action

- iv) Click on the script action to activate the script. Go inside the folder. If you set your document life cycle to be read or remove or add then when you read or remove or add your document it will activate the email.

II) Transform Binary to Text Action

Transform binary to text action lets you transform your “.pdf” or “.doc” files into “.text”. Here is an example.

- i) Uploaded a pdf file and click on the manage actions icon.
- ii) Select in the listbox the transformBinaryToTextAction.
- iii) Enter the name id, the lifecycle for your document, and optional description. Click on the save button before exiting.
- iv) You should see [exo:transformBinaryToTextAction] in the instance of action type column. Click on this instance to activate the script.
- v) Return to the folder that contain your pdf document and you will see the transformbinarytotextaction name id as shown in figure 1 below.

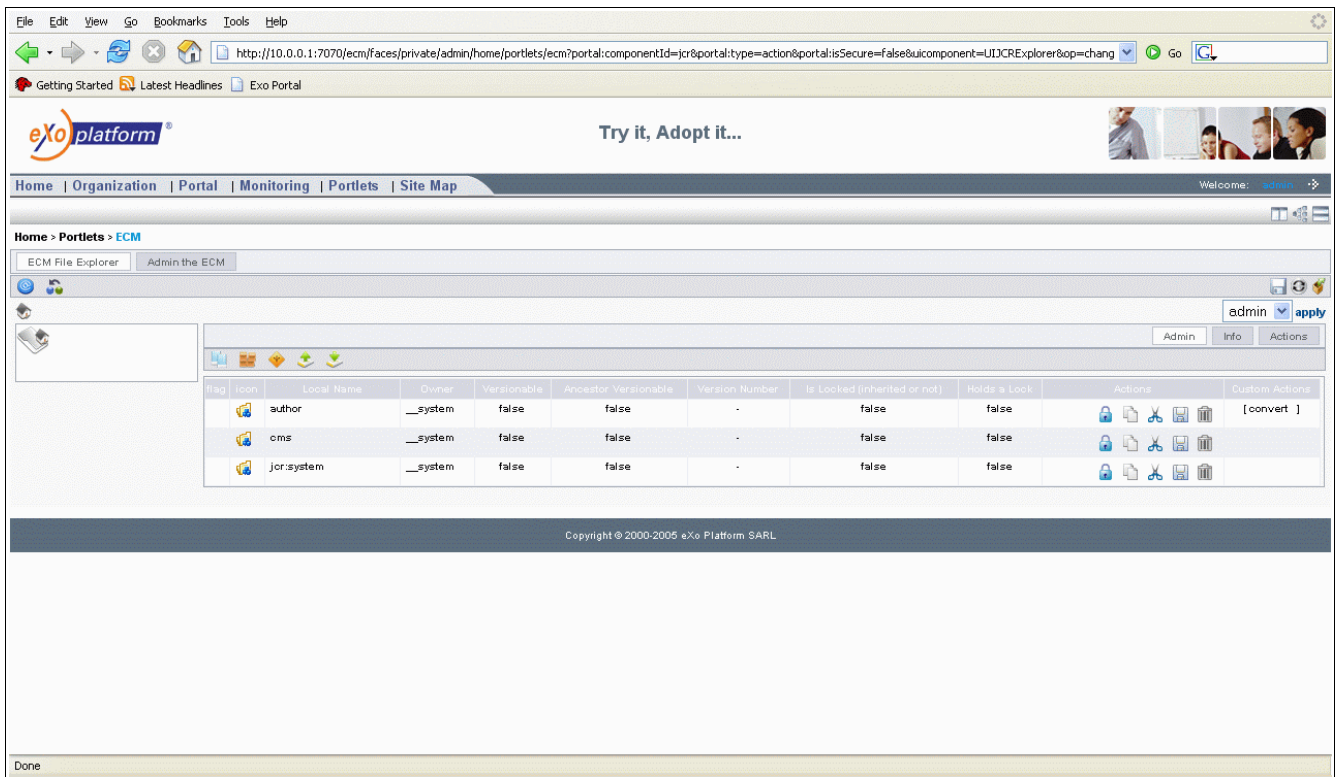


Figure 1 Transformbinarytotextaction

- vi) Click on the action to execute the script.
- vii) Go inside the folder and you will see the script name, the pdf file, and the text file as shown in figure 2 below.

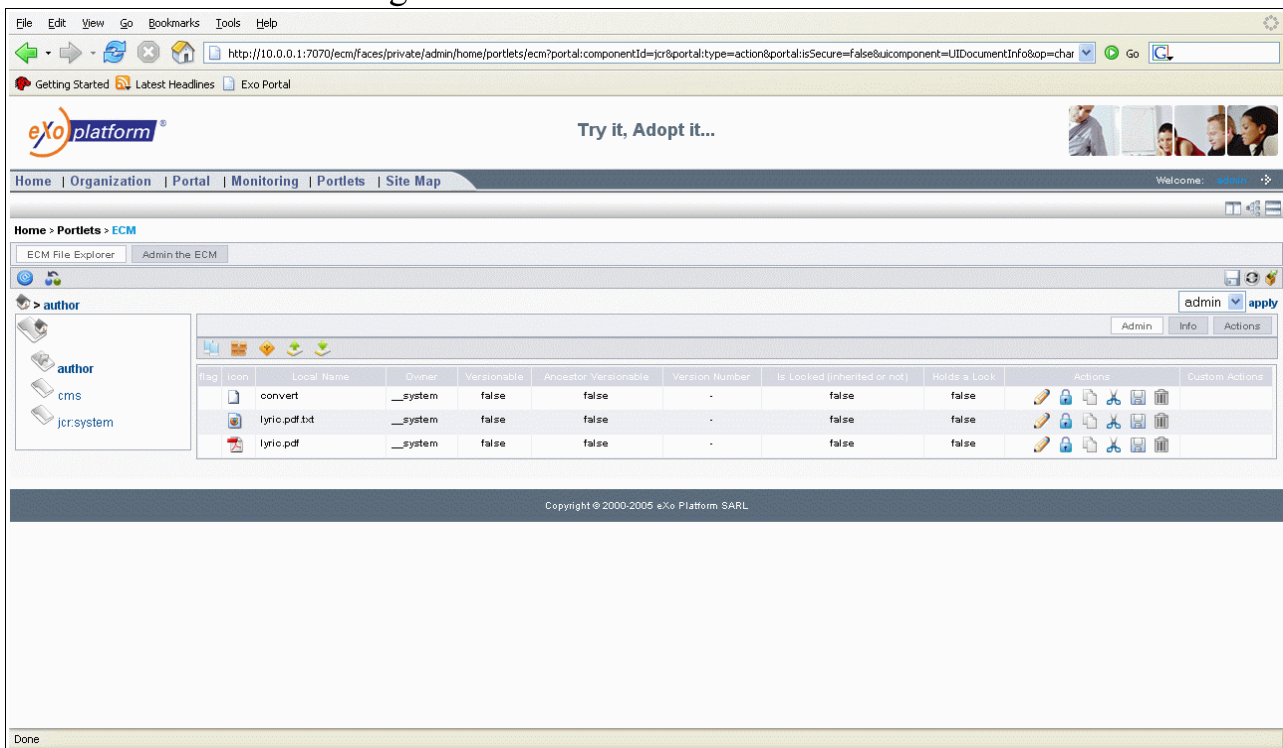


Figure 2 pdf file has successfully converted to a text file.

III) Workflow Action

Creating a workflow action for your document or folder will set it for content validation workflow. For example, a writer completed an article and the article is ready for publishing but the content of the article needs to be validated before it goes online. The first step is to have the document validate by a validator. Once the document has been validated and approved it will be moved to a specific location. Here is how you can set the workflow action with the manage action icon.

- i) Click on Manage Action icon and click on add.
- ii) Select from the drop down list box the workflowAction.
- iii) Enter value in workflowAction form.

<i>Field</i>	<i>Form</i>
Name ID	Name of your workflowAction
Lifecycle	The life cycle of the document
Description	Your comment
Validator	The person to validate your content
Workspace	Workspace where the content will be moved
With path	The path folder for which the content will be moved to

- iv) Click save after you have completed entering the form fields. You will then see the the workflow action script in the instance of action type column.

d) Export Node

Export node lets you package all your workspace data to your directory for portability and backup purposes. The content hierarchy will be exported as an xml document that can be imported in any JCR compliant implementation. There are two formats which you can select: xml document and xml system view. You only check the xml system view when you want to export the jcr:system folder.

e) Import Node

Import node lets you import all your folders data and information back to your workspaces. The following code shows how simple it is to import some content we have uploaded.

```
byte[] content = uiupload.getContent() ;
try {
    session.importXML(node.getPath(), new ByteArrayInputStream(content),
        ImportUUIDBehavior.IMPORT_UUID_CREATE_NEW) ;
} catch (Exception e) {
```

```
JCRExceptionManager.process(iprovider, e);
}
```

The UIUpload component is a JSF one whose goal is to render an upload HTML field inside a multipart form and to extract and wrap the uploaded result so that we can simply get it with the method `uiupload.getContent()`.

The JCR API provides the interface `ImportUUIDBehavior` that defines how the implementation should behave when the XML tree contains nodes with UUIDs that conflict with the UUIDs of nodes that are already in the repository. The four properties are talkative enough and we will not describe them more.

```
public interface ImportUUIDBehavior {
    public static final int IMPORT_UUID_CREATE_NEW = 0;
    public static final int IMPORT_UUID_COLLISION_REMOVE_EXISTING = 1;
    public static final int IMPORT_UUID_COLLISION_REPLACE_EXISTING = 2;
    public static final int IMPORT_UUID_COLLISION_THROW = 3;
}
```

B) ECM Admin

Manage Node type

Manage node type lets you create your own node type, export and import existing node type using xml format. With eXo platform ECM portlet, you can register your node type during runtime and with a user friendly interface you don't have to worry about the xml structure behind it. Let's see how you can create a node type in the following example.

- i) Click on manage node type icon. You will see a list of existing node type. To create your own node type click on the add icon.
- ii) Below describes each fields you must enter in order to create your node type. Add property and add child are optional.

<i>Add node type</i>	<i>Description</i>
Node type name	Name of the node
Is Mixin Type	True or false for versioning
Orderable child nodes	True or false for child nodes order
Primary item name	
Super types	

<i>Add property</i>	<i>Description</i>
Definition name	Name of the property
Required type	
Is auto created	True or false for auto created
Is mandatory	True or false for mandatory
On parent version	
Is protected	
Is multiple	
Value constraints	

<i>Add child</i>	<i>Description</i>
Definition name	
Default primary type	
Is auto created	
Is mandatory	
On parent version	
Is protected	
Is same name sibling	
Required primary types	

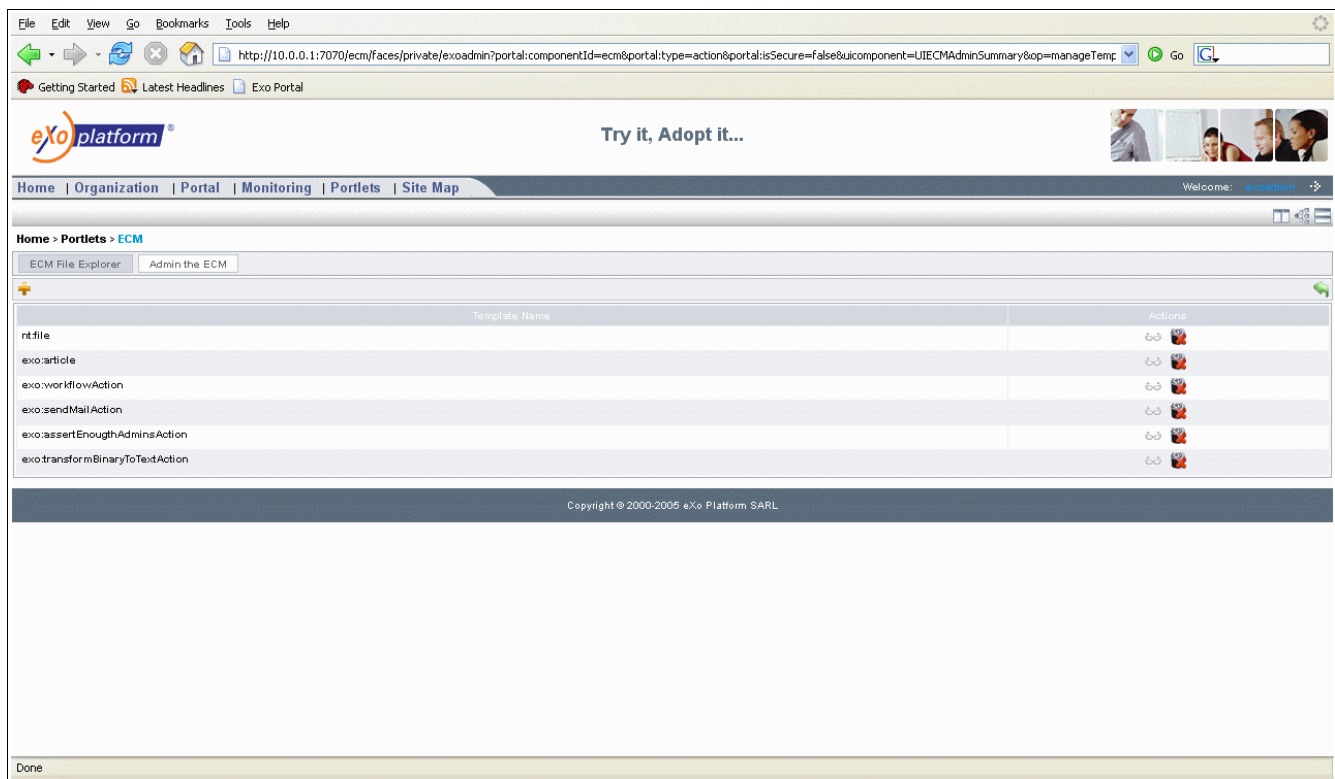
iii) Click save when you are done.

Manage Templates

Once you have created a node type you can bind this node type to a template. A template is a velocity script that produces HTML code to build the template interface for your node. Each nodes type must have two templates: dialog and view template. eXo platform ECM portlet comes with several existing node templates, but we will show you how you can create your own node templates in the following example:

- i) Click on the manage templates icon. You can see that there are several predefined template as shown in the figure 1 below.

Figure 1 Manage template environment



- ii) Click on add node type icon.
- iii) In the add a new type environment choose from the drop down list “nt:versionLabel”.
- iv) Insert this code in the dialog section

```
#jsfDialogFormField("inputName=hiddenInput1" "jcrPath=/node/jcr:content" "nodetype=nt:resource" "visible=false")
#jsfDialogFormField("inputName=hiddenInput2" "jcrPath=/node/jcr:content/jcr:encoding" "visible=false" "UTF-8")
#jsfDialogFormField("inputName=hiddenInput3" "jcrPath=/node/jcr:content/jcr:mimeType" "visible=false" "text/html")
#jsfDialogFormField("inputName=hiddenInput4" "jcrPath=/node/jcr:content/jcr:lastModified" "widget=date"
"visible=false")
```

```
<div class="title">Version Label</div>
<div class="detail">
<div class="row">
<label>Version Name</label>
#jsfDialogFormField("inputName=name" "jcrPath=/node" "editable=false" "visible=if-not-null")
</div>
<div class="row">
<label>Comment</label>
#jsfDialogFormField("inputName=summary" "jcrPath=/node/jcr:content/jcr:data" "widget=textarea")
</div>
</div>
```

The first four lines allows us to create hidden components with pre configured values such as in that case the file encoding (UTF-8) or the mime type (text/html).

Then it comes to the HTML generated part. The widget value tells what type of JSF UIComponent should be used and hence what will be the generated HTML. The “jcrPath” directive attribute tells to which location the field value should be stored. That way it is possible to create a complex dynamic form that will end into the creation of a

JCR node.

v) Insert this code in the view section.

```
#set($node = $uicomponent.getNode())
#set($wikiService = $uicomponent.getWikiService())
#set($wikiContext = $uicomponent.getWikiContext())
#set($node = $node.getNode("jcr:content"))
#set($nodetype = $node.getPrimaryNodeType().getName())

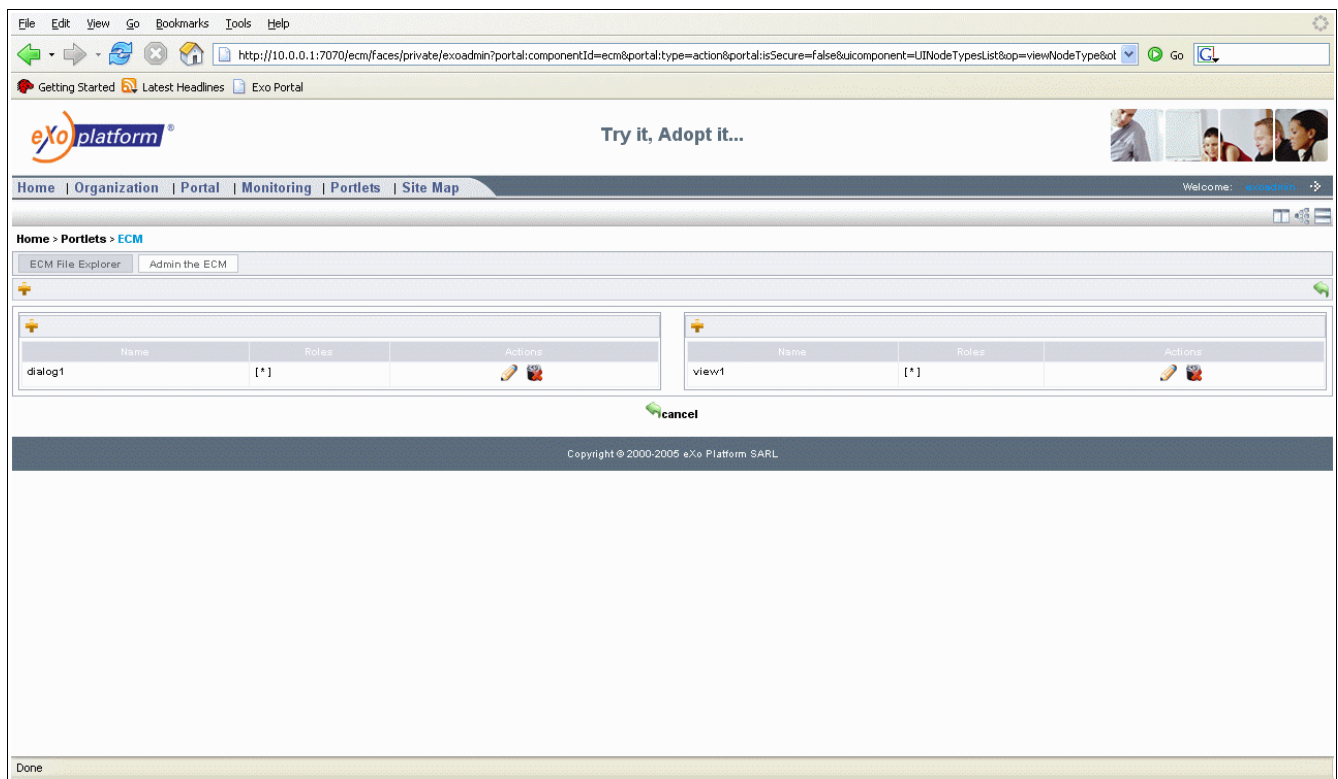
#if($uicomponent.isNodeTypeSupported(${nodetype}))
  #set($oldNode = $node)
  $uicomponent.setNode($node)
  #parse("jcr:/cms/templates/${nodetype}/views/view1")
  $uicomponent.setNode($oldNode)
#else
  #if($node.hasProperty("jcr:data"))
    #if($node.hasProperty("jcr:mimeType"))
      #set($mime = $node.getProperty("jcr:mimeType").getString())
      #if($mime.startsWith("text"))
        #set($text = ${node.getProperty("jcr:data").getString()})
        $wikiService.toXHTML($text, $wikiContext)
      #else
        <a href="/${uicomponent.getPortalName()}/jcr/${uicomponent.getWorkspaceName()}${node.getPath()}">Download This File</a>
      #end
    #else
      <a href="/${uicomponent.getPortalName()}/jcr/${uicomponent.getWorkspaceName()}${node.getPath()}">Download This File</a>
    #end
  #else
    <a href="/${uicomponent.getPortalName()}/jcr/${uicomponent.getWorkspaceName()}${node.getPath()}">Download This File</a>
  #end
#end
```

The idea for the view template code is simply to render some HTML with information stored on a JCR Node object.

vi) Once you are done you will see that the nt:versionLabel appeared in the list of template. Should you wish to edit click on the view icon next to it as shown in figure 2 below.

Figure 2 Manage dialog and view template

vii) To verify what you have done, click on the add document and in the drop down list you will see nt:versionLabel.



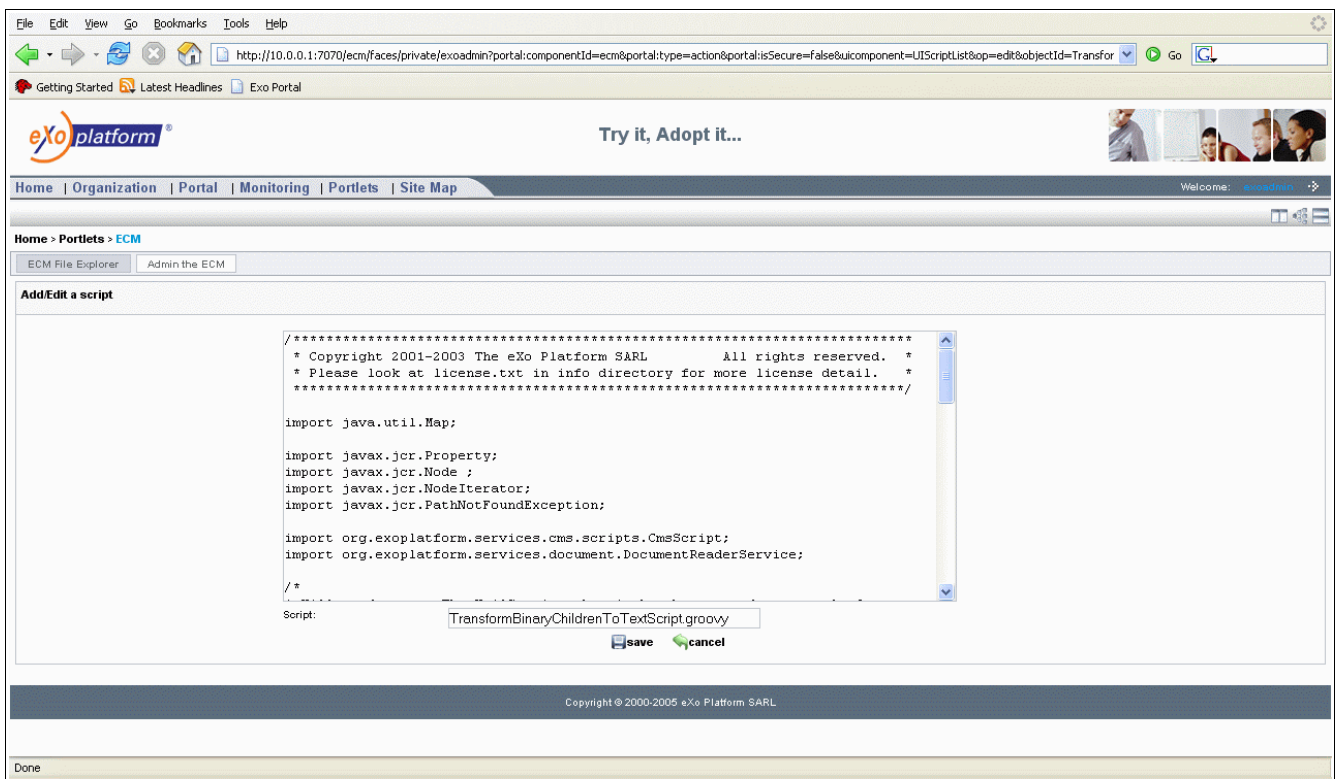
Manage Scripts

eXo platform ECM provides an interface for you to add and edit script at runtime and to make this possible eXo platform had to modify the class loading mechanism in Groovy in order to invalidate the loaded cache object of the class loader when a script is modified online as well as loading the imported object from JCR.

We have seen in the manage action sections script such as *[sendmail]* or *[transformbinarytotext]*, these are groovy script that process the business logic when an action is activated from the ECM system.

You can add a new script or edit an existing one. Below in figure 1 is a edit version of the *[transformbinarytotext]* script.

Figure 1 TransformBinarytoText edit mode



Manage Rules

eXo platform ECM manage rules environment lets you plug rules with ECM Actions through native Drools support and it allows you to add or edit your business rules when a document is added, removed, or read.

Create Action Type

Create an action type lets you view and add ActionType online. It is important to understand that an ActionType is a simple custom NodeType that extends the “exo:action” nodetype as shown in figure 1 below.

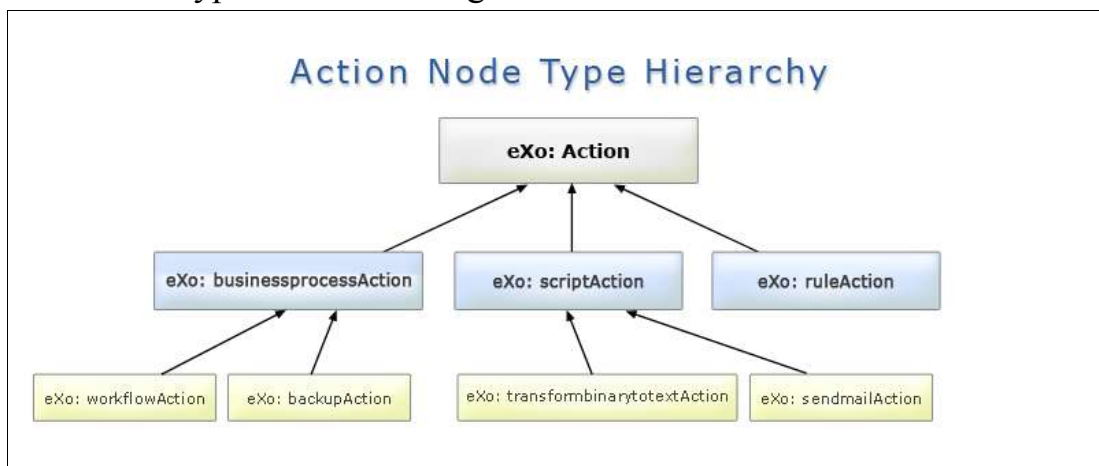


Figure 1 Action Node Type Hierarchy

eXo platform ECM has created several action types already and some of it will activate groovy script and others will activate Business Processes or Business Rules. The figure

2 below shows all the predefined action types.

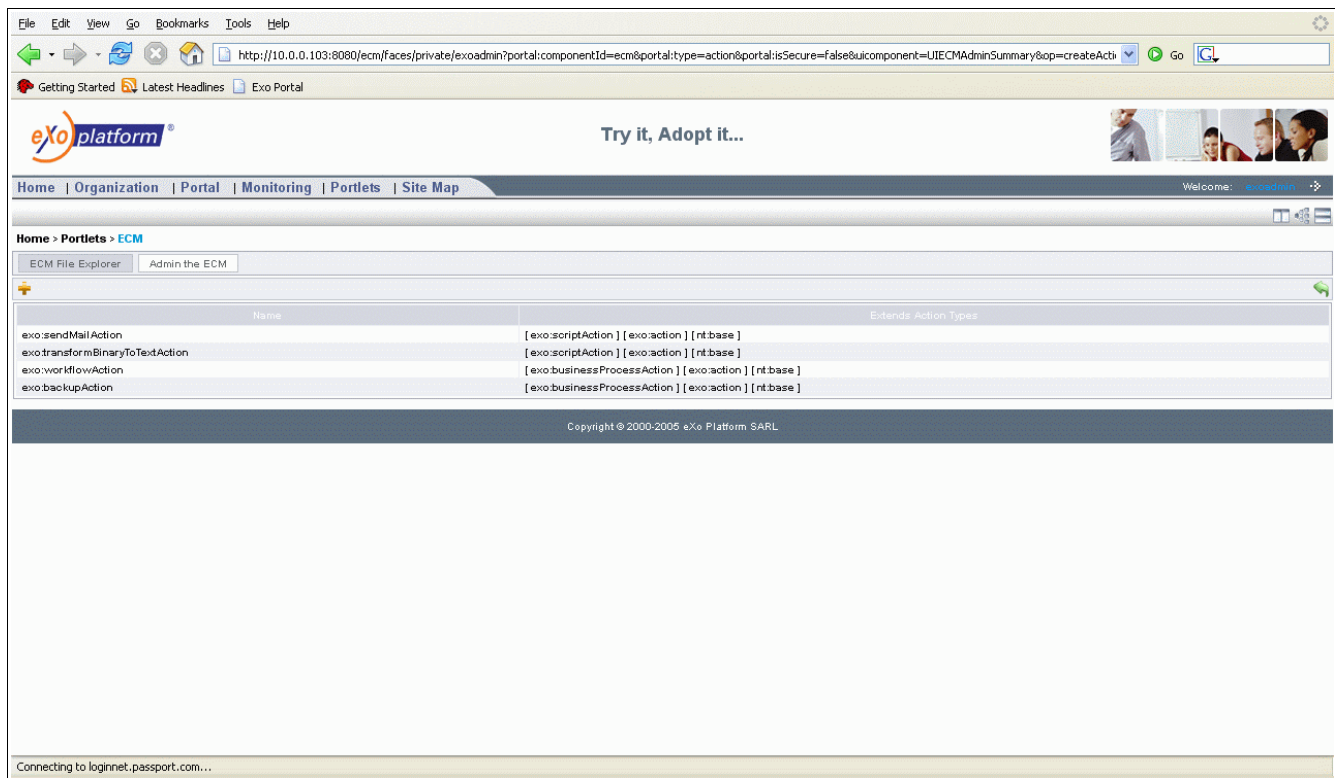


Figure 2 Action types

Depending on the ActionType you want to create you will have to extend one of the default action type such as scriptAction, ruleAction or businessprocessAction. Let's see how action type is created.

- i) Click on add action type icon.
- ii) Select from the drop down list box what action you want to extend. Note that each time you select an extended action type the form below will dynamically change according to the extended action. For example in the case of a `exo:businessProcessAction` parent type, the second form will allow you to select among all the deployed business processes as shown in figure 3.

Figure 3 Add action type

The screenshot shows the eXo Platform web interface. At the top, there's a navigation bar with links: Home, Organization, Portal, Monitoring, Portlets, Site Map. Below this, a breadcrumb trail shows 'Home > Portlets > ECM'. The main content area is titled 'Add/Edit a Action Type'. It contains a form with the following fields:

- Extends Action Type:** A dropdown menu showing 'exo.businessProcessAction'.
- Name:** A text input field with a hint '(starts with "exo.")'.
- Is the Action a Move:** A checkbox.
- Business Processes:** A dropdown menu showing 'content-backup'.
- Variables:** A text input field with a plus icon to add more.

At the bottom of the form are 'save' and 'cancel' buttons. The footer of the page reads 'Copyright © 2000-2005 eXo Platform SARL'.

iii) The variable fields allow you to provide more information to the script, rule or business process. In the case of the publication workflow, the “validator” key is a variable used by the business process to tell the identity of the next actor in the process.

Note that you only tell about the variable key and not the value of it. Indeed the value will only be entered when an action instance of that new type will be created thanks to the associated dialog template.

Note that once an ActionType is defined, you will have to create a DocumentType out of it, aka create the dialog and view templates. The dialog template will need to include the variable as fields.

The ActionType services works with plugins that can be easily added. Each plugin is responsible of an action type; hence there are 3 by default. Each plugin configuration XML file can also be loaded as predefined action instances such as the publication or the backup workflows. This XML will register a BPActionPlugin into the ActionServiceContainer. The BPActionPlugin holds an instance of type `exo:businessProcessAction` named “publication” with all the associated properties.

Manage View

Manage view lets you customize as well as restrict certain functionalities based on users and group. It lets you manage how workspace is viewed based on the permission

setting. For example, you can create a view that allows all users in group admin to view: add document, upload, and view references icon. Follow the instruction below to create your own view.

i) Click on Manage View icon. As you can see from the table in figure 1 below, it shows the view name, the permission, and the view tab. Click on add view icon to create another view.

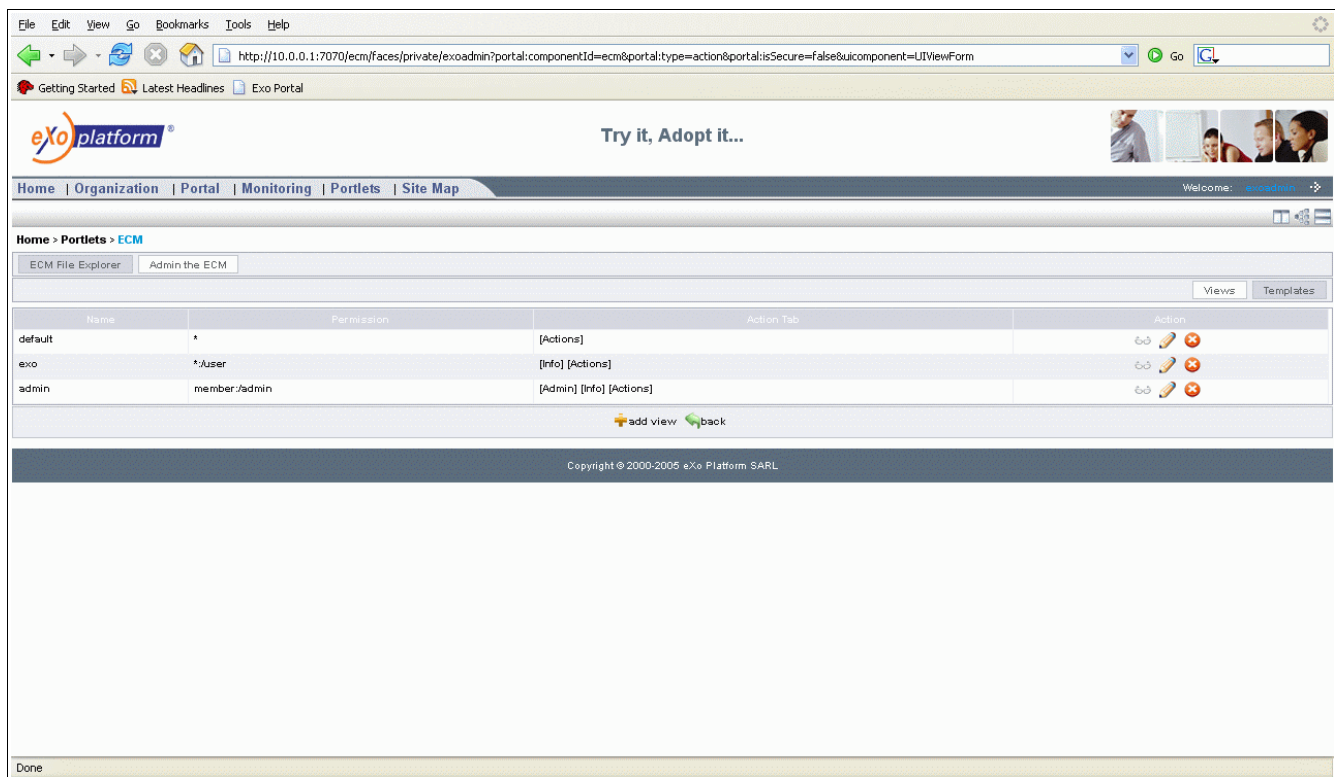


Figure 1 Manage View

ii) The table 1 below describes the add view form. Fill the form and click save.

<i>Field</i>	<i>Description</i>
View Name	The name of your view
Permission	Select group and membership type permission
Tabs	Click on the add tabs icon and enter the tabs name. Choose from the list all the available actions.
Template	Template consist of Thumbnailsview, MacView, and ListView.

Content Validation Workflow

Let's take a global view of how online text publication works:

An author creates a draft document offline and then uploads it to the server through WebDAV protocol. Once uploaded and stored, an action is launched to transform the document into a text format for content validation by an authorized editor. As soon as the validation process ends the document is published. The published document will be moved to the archived location in the backup workspace when its publication date expires. The figure 1 below is the content validation workflow diagram that shows the content evaluation such as disapprove, delegate, approve, and refuse.

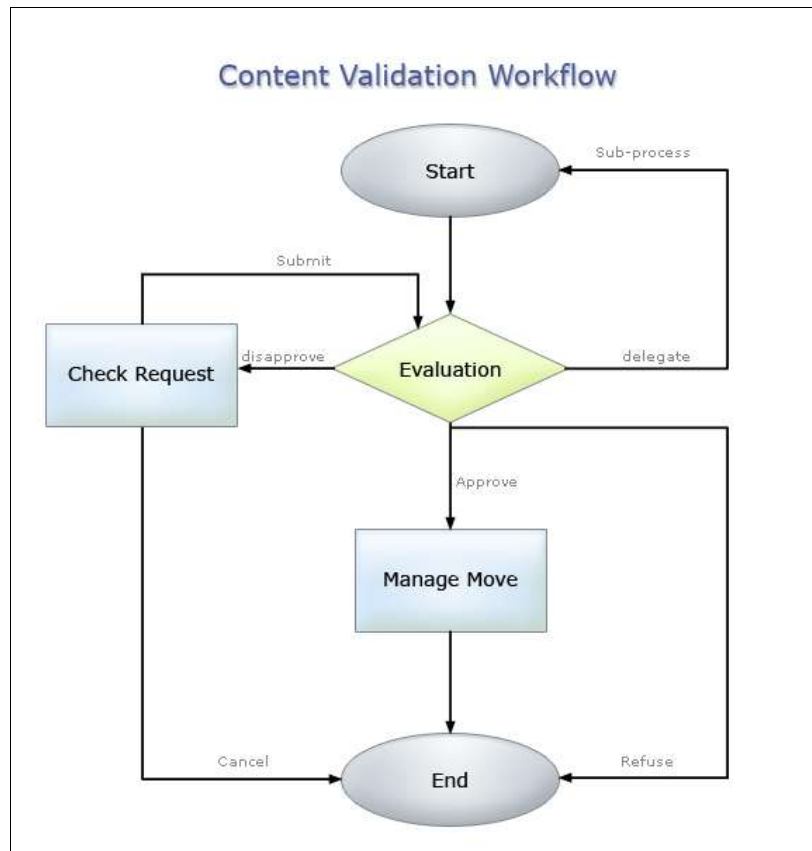


Figure 1 Online text publication

In the manage action section we have discussed the workflowAction script and we have showed you how you can set a folder to become the content validation activation folder. eXo ECM portlet has preconfigured *cms/categories* news folder to be the content validation activation folder and all users in /company/directors group are the validator. The figure 2 below shows the preconfigured workflow script in the draft workspace publications folder.

The screenshot displays the eXo Platform web interface. At the top, there is a navigation bar with links like 'Getting Started', 'Latest Headlines', and 'Exo Portal'. Below this, a breadcrumb trail shows 'home > Portlets > ECM'. The main content area is titled 'cms > publications' and contains a form for editing a document. The form has several fields: 'Name' (value: publication), 'Id' (value: publication), 'Name' (value: publication), 'Lifecycle' (value: add), 'Description' (value: publication workflow), 'Validator' (value: member/company/direction), 'To workspace' (value: production), and 'with path' (value: /cms/publications). There are 'save' and 'cancel' buttons at the bottom of the form. The footer of the page indicates 'Copyright © 2000-2005 eXo Platform SARL'.

File Edit View Go Bookmarks Tools Help

http://10.0.0.1:7070/ecm/faces/private/exoadmin/home/portlets/ecm?portal:componentId=jcr&portal:type=action&portal:isSecure=false&uiComponent=UIDocumentInfo&op=

Getting Started Latest Headlines Exo Portal

home > Portlets > ECM

ECM File Explorer Admin the ECM

cms > publications

admin apply

Actions Info Admin

Edit the document

Name: publication

ContentValidation

Id: publication

Name: publication

Lifecycle: add

Description: publication workflow

Validator: member/company/direction

To workspace: production

with path: /cms/publications

save cancel

Copyright © 2000-2005 eXo Platform SARL

Done

Figure 2 eXo preconfigured workflow publication script

By copying a document in the draft workspace news folder, the validation workflow process is activated and a task is now waiting for a member of the group / company/direction to response. In our example the admin user is a member of that group as shown in figure 3 below.

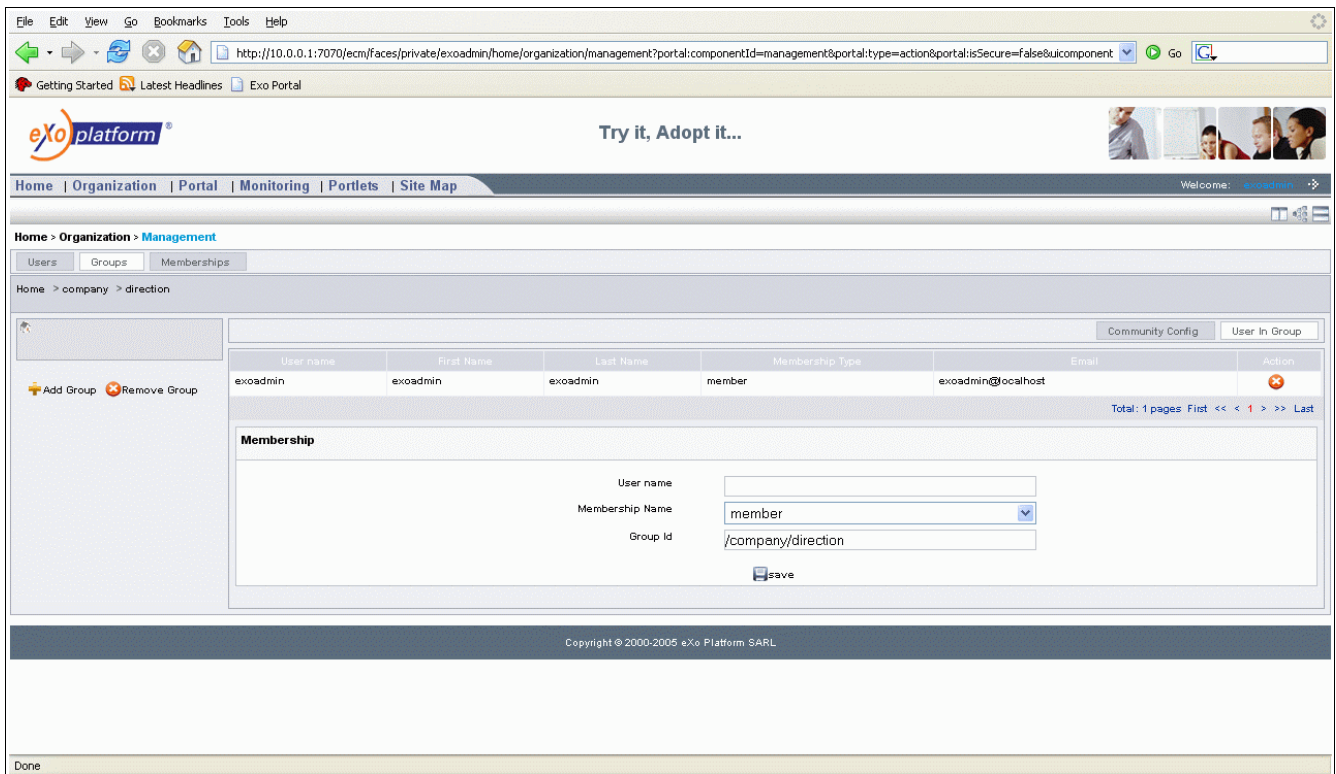


Figure 3 Display users in the /company/direction group

To check that the process has started we can go to the workflow portlet. It contains two tabs: one shows the status of all the current versions of the deployed processes and the other one shows the timers waiting for a task to complete or to be activated as shown in figure 4 below.

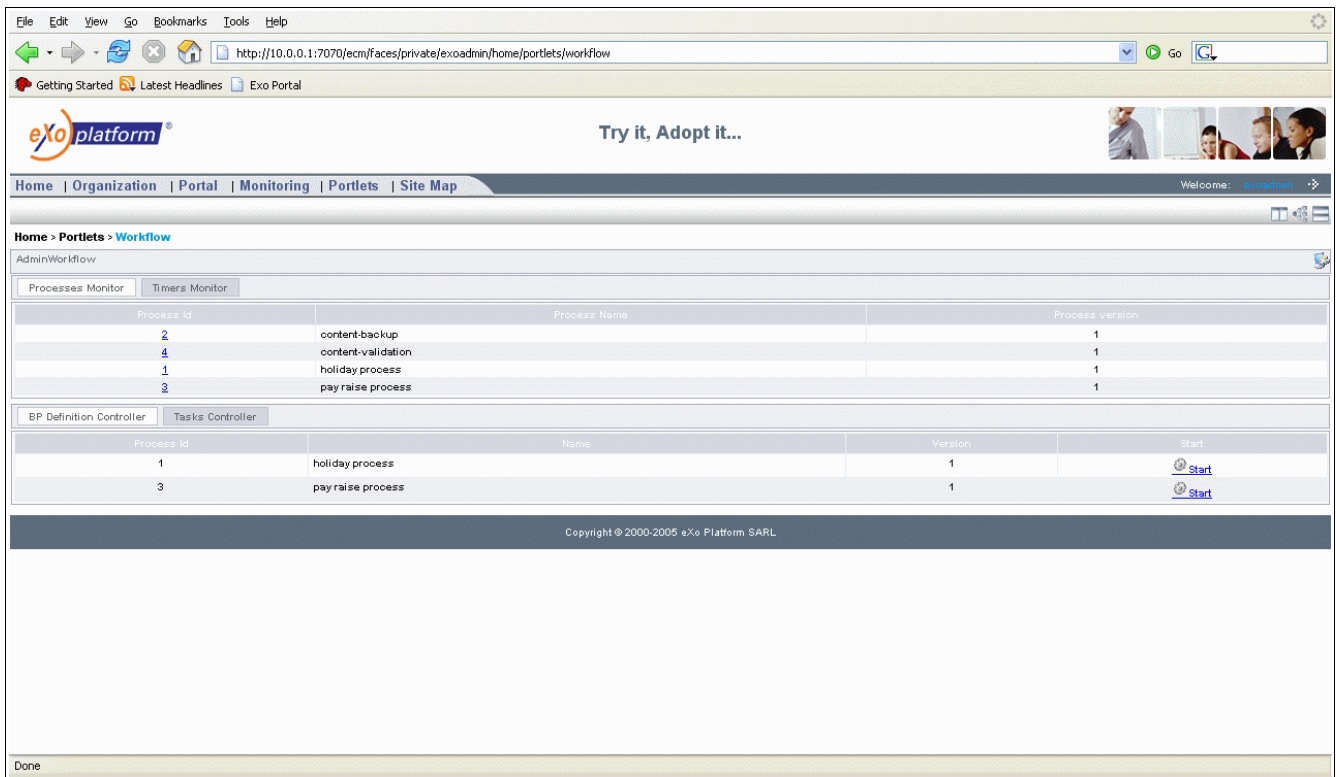


Figure 4 Workflow portlet

The edit mode icon located on the right hand side of the workflow portlet contains a form that allows uploading a new business process archive or a new version of an already deployed one. If you click on the content validation process tab you will see that a process instance is started but not yet finished. The processes tab also displays information on the instance history such as the activity that was completed, the end date and who completed it.

If the workflowAction folder is activated then your workflow portlet evaluation activity will have a waiting process that needs to be validated by members of the validator group. In our case admin user is a member of the /company/direction group so when he visualizes his normal activity Task List he will see tasks waiting for him.

Here is how you can view and manage the task evaluation:

- i) To view the evaluation task list click on the instance id next to the content validation in the processes monitor tab.
- ii) Click on the tasks controller tab. You will see that there is a evaluation process waiting as shown in figure 5 below.

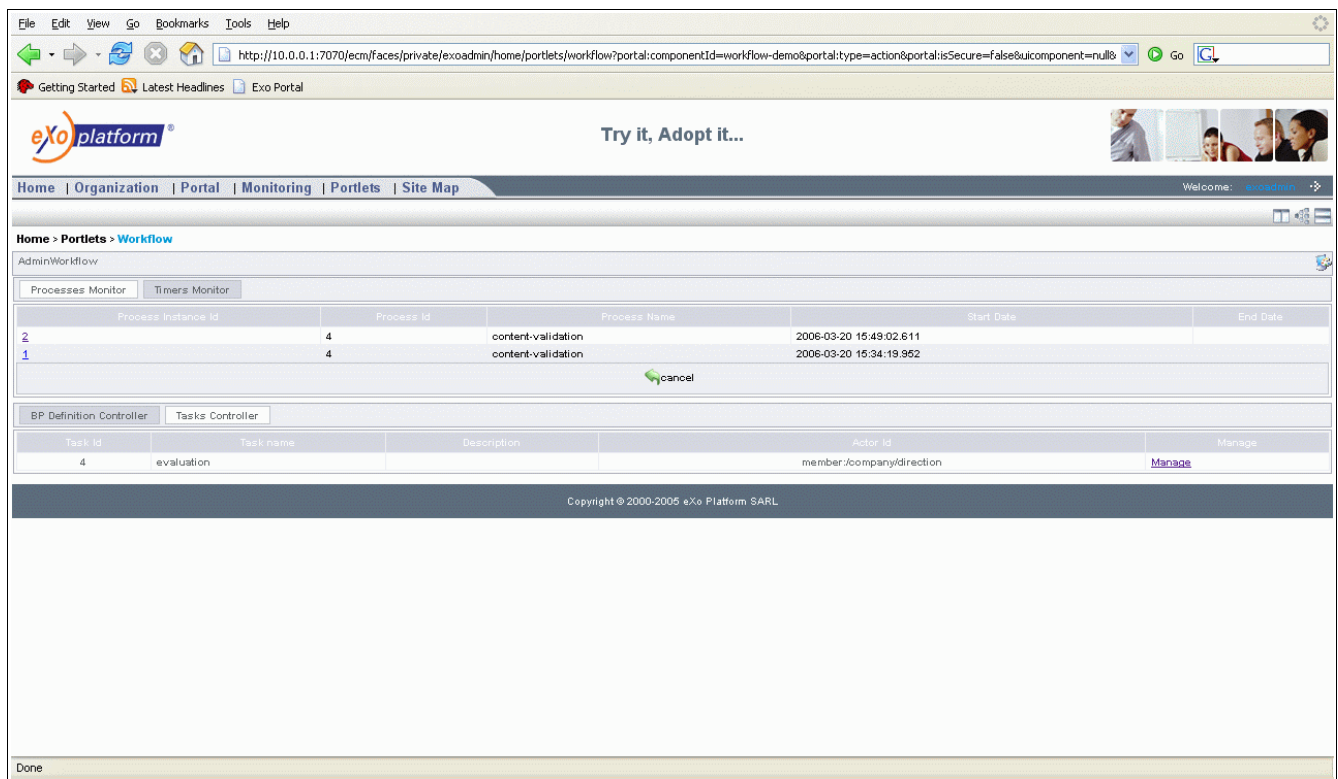


Figure 5 Evaluation process

iii) Click on the manage and you will be taken to the document evaluation portlet as shown in figure 6 below.

ADMIN VIEW

Document: MY DOC VERSION 1

Creator: admin

Publication starts on: 11 Dec 2005 18:00

Publication ends on: 11 Dec 2005 18:00

Delegate the validation:

☐ Approve ☐ Disapprove ☐ Delegate ☐ Refuse ☐ Cancel

Figure 6 Evaluation document portlet

iv) Refer back to the figure 1 above, it shows the document evaluation process for which a validator must take. The document can be disapproved (and the document creator will then see a task asking for changes in his personal task list), refused (the process ends) or delegated to an individual or a group of people (technically this will launch a recursive sub business process, as many times as we have delegations).

v) Here we choose the approve options for our document and choose the publication start date and the publication end date from the activity form. In other words, once the document is validated it will only be copied from the draft to the production workspace when the publication start date begins. The document will be moved to the location where the browse category reference so that it can be displayed. When the publication end date is reached the

document will then be moved to the backup workspace and will therefore not be web published anymore.

- vi) Check if the content has been backup by accessing the backup workspaces. Note that in eXo JCR, each workspace can have its own data source therefore for physical security reasons the backup service should be located in another DB or file system as the draft and production workspaces.

Browse Category

Browse category is a portlet that displays JCR data, it acts as a view to ECM repositories. The figure 1 below shows ECM news and sports folders and its children.

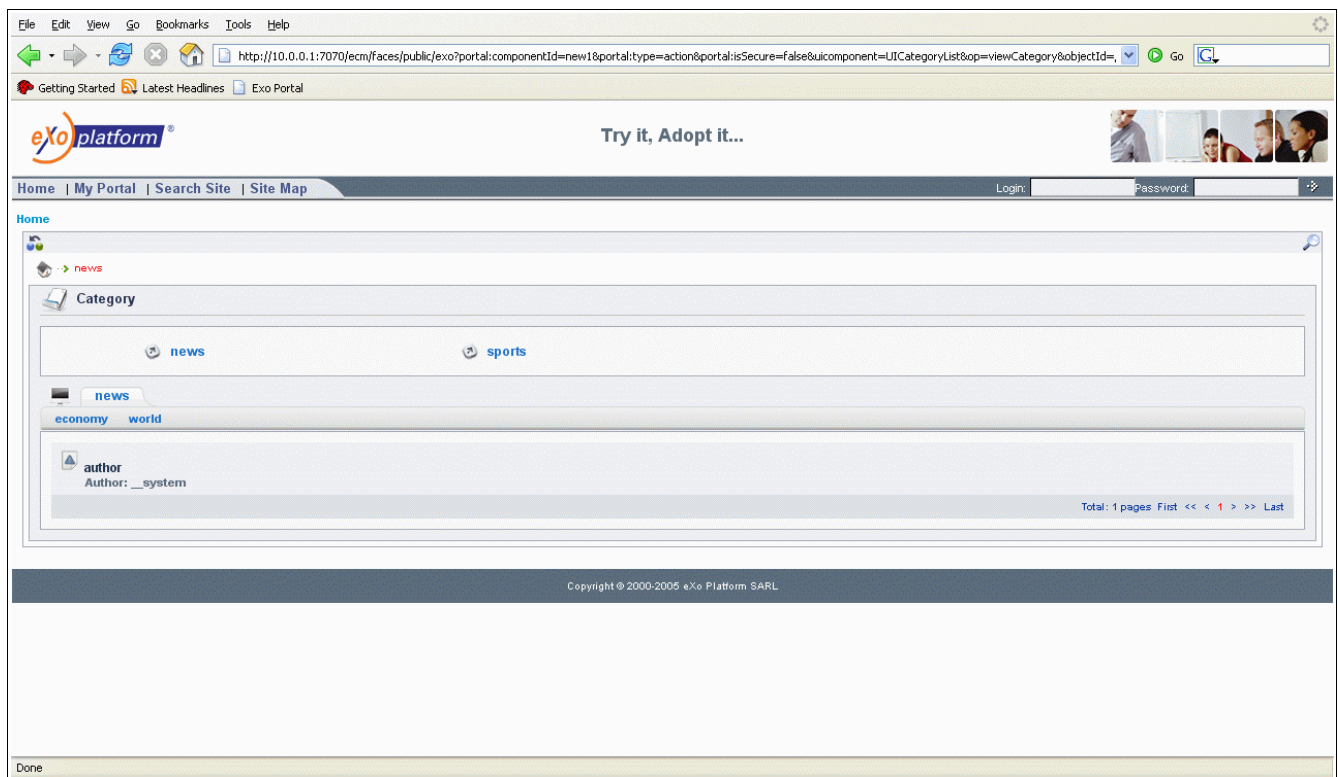


Figure 1 Browse category view to ECM folders

You can configure the browse category portlet to reference any ECM workspaces and folders. Here's how;

- i) From the browse category portlet, click on the information icon on the upper right hand side. It will display the workspace configuration form as shown in figure 2 below.

File Edit View Go Bookmarks Tools Help

http://10.0.0.1:7070/ecm/faces/private/exoadmin/home/portlets/categorybrowser?portal:componentId=categorybrowser&portal:portletMode=edit

Getting Started Latest Headlines Exo Portal

eXo platform Try it, Adopt it...

Home | Organization | Portal | Monitoring | Portlets | Site Map

Welcome: **exoadmin**

Home > Portlets > Browse Category

Category Browser

Workspace Config

Workspace: production

Categories Path: /cms/categories/cms

Template: org/exoplatform/portlets/content/categorybrowser/i

Detail Box Template: org/exoplatform/portlets/content/categorybrowser/i

Reference Document: ☒

Children Document: ☒

View Toolbar: ☒

Detail Maximized: ☐

Target Page:

Target Portlet (ComponentId):

save

Copyright © 2000-2005 eXo Platform SARL

Done

Figure 2 Workspace Config

ii) The table 1 below describes the workspace config

Field	Description
Workspace	The workspace for which your browse category reference to.
Categories path	The folder path for which you want browse category to display.
Template	Default template rendering
Detail box template	Default box template rendering
Reference Document	Display reference documents
Children Document	Display children from document root
View Toolbar	Show category toolbar
Detail maximized	
Target Page	
Target Portlet (Component id)	

Table 1 Workspace Config