SHIWA Repository Admin & Developer Manual

14th March 2014

This manual documents the SHIWA application repository. Sections 1-2 describe the key entities, actors and use cases, section 3-13 describe how the provided features can be accessed via the GUI.

TABLE OF CONTENTS

1. INTRODUCTION

1.1. BACKGROUND

Researchers of all disciplines, from Life Sciences to Computational Chemistry, create and use ever-increasing amounts of complex data, and rely more and more on compute-intensive modelling, simulation and analysis.

Scientific workflows have become a key paradigm for managing complex tasks and have emerged as a unifying mechanism for handling scientific data. Workflows capture the essence of the scientific process, providing a means to describe it via logical data- and/or workflows. Workflows are mapped onto concrete Distributed Computing Infrastructures (DCIs) to perform large-scale experiments.

The learning curve to use workflows, however, is demanding because workflows typically have their own user interfaces/APIs, description languages, provenance strategies, and enactment engines, which are not standard and are not interoperable. Therefore it is difficult to reuse and share workflows, this inhibits the growth in uptake and proliferation of workflows in scientific practice.

1.2. SHIWA SIMULATION PLATFORM

User communities from all around Europe use many kinds of different workflow languages. They develop workflows using one of the workflow engines. Workflow development, testing and publication are time consuming processes and require specific expertise. These limit the number of available workflows, so it is important to share them. Workflows developed for one workflow system is normally not compatible with workflows of other workflow systems. In the past if two user communities using different workflow systems wanted to collaborate, they had to re-implement the workflows in their own workflow system. This situation can be resolved by emerging new workflow interoperability technologies provided by the SHIWA Simulation Platform.

According to the new SHIWA technologies publicly available workflows can be used by different research communities working on different workflow systems and are enabled to run on multiple distributed computing infrastructures. As a result workflow communities are not locked anymore in to their own workflow system and are able to execute workflows on several distributed computing infrastructure.

1.3. KEY COMPONENTS

The SHIWA Simulation Platform offers users production-level services supporting workflow interoperability. As part of the SHIWA Simulation Platform the SHIWA Repository facilitates publishing and sharing workflows, and the SHIWA Portal enables their actual enactment and execution in different DCIs. The simulation platform supports use cases targeting various scientific domains or subdomains will serve to drive and evaluate this platform from a user's perspective.

1.4. LINKS

The SHIWA homepage is http://www.shiwa-workflow.eu
The SHIWA Simulation platform can be found at https://ssp-test.cpc.wmin.ac.uk/liferay-portal-6.1.0/en
The SHIWA Repository can be found at

http://repo-test.cpc.wmin.ac.uk/shiwa-repo/

1.5. REFRENCING WORKFLOWS AND IMPLEMENTATIONS

To reference a workflow or implementation with a URL (**U**niform **R**esource **L**ocator), copy the URL in the address bar while viewing the required workflow or implementation.
The link will be of the format:
http://shiwa-repo.cpc.wmin.ac.uk/shiwa-repo/public/edit-application.xhtml?appid=4752
or
http://shiwa-repo.cpc.wmin.ac.uk/shiwa-repo/public/edit-implementation.xhtml?impid=3208

1.6. HOW TO REGISTER AS A USER

Should you want to register, to enable workflow development, please send an email to shiwa-repo-admin@cpc.wmin.ac.uk.

2. ENTITY, ACTOR & USE-CASE SPECIFICATION

2.1. ENTITY DEFINITIONS

Workflow. This entity represents an abstract workflow. It describes the inputs and outputs and explains what the workflow does. It also specifies sample inputs and outputs (configurations), and some further information.

Implementation. This entity represents an implementation (or concrete workflow) of a workflow. It strictly follows the input and output definitions of the abstract workflow and implements the functionality given in the workflow description. It contains or references (via e.g. URLs) the workflow definition file, the implementation description files, dependencies to run the workflow on its associated workflow engine and the workflow's graph.

Engine. This entity represents a workflow engine that is able to interpret and execute a given implementation.

User. This entity represents a repository user associated with a specific role (repo admin, workflow developer, e-scientist).

Group. This entity grants read/write/download rights to a particular workflow for a set of users (the members of the group).

Platform. This entity describes in which desktop and/or service Grid environment the implementation can be executed.

Files. This entity contains the files related to workflows and their implementations.

2.2. ACTOR DEFINITIONS

E-scientist. This actor is the consumer of the contents of the repository, i.e. workflow engines and workflows to run experiments. This actor should not register with the repository to browse and search the repository.

Workflow Developer. This actor is the creator and maintainer of the contents of the repository, i.e. workflows and their implementations and configurations. The actor should register with the repository to be able to upload, modify and delete workflows.

Repository Administrator. This actor is a system administrator has the highest role among the actors. His task is to maintain the SHIWA repository and to support all other users.

2.3. REPOSITORY MODEL

Users represent all actors (e-scientists, workflow developers and repository administrators). We distinguish among actors based on their roles and the corresponding access rights. Workflow developers may own workflows and their implementations. They can upload, modify and delete workflows. E-scientists can browse validated workflows and their validated implementations, download and run them on the SHIWA Simulation Platform – pending access permissions being granted

The repository also enables workflow developers and repository administrators to create and manage groups. Groups support controlled access to workflows and their implementations.

Workflows have implementations and attributes. Implementations have attributes and files. They are also associated with workflow engines.

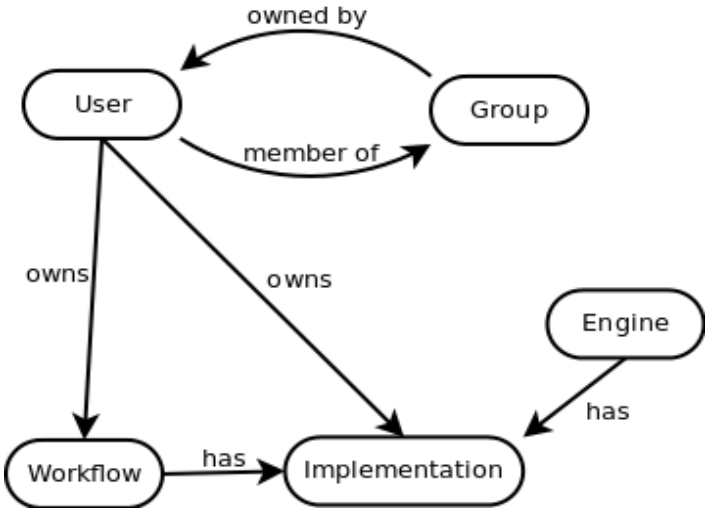


Figure 1: Repository model

2.4. USE CASES

This manual covers the use cases of all users.

2.4.1. E-SCIENTIST

They are the consumers of the contents of the repository and can access the following functionality (see Figure 2: E-Scientist use case).

Browse workflows (see section 6.1)
Browsing includes searching and listing public workflows based on their metadata.

Browse implementations (see section 7.1)
E-scientists can browse public implementations of the workflows selected by the "Browse workflows" operation.

List files (see section 7.1.5)

E-scientists can list files belonging to workflows or implementations selected by above operations.

Download (see sections 6 and 7)
Users can download workflows and their related entities (implementations, configurations and files).

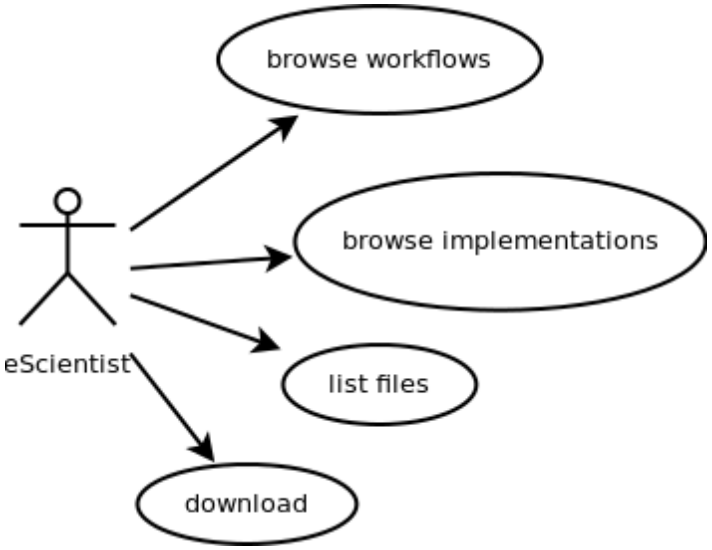


Figure 2: E-Scientist use case

2.4.2. Workflow developer

They are the creators and maintainers of the contents of the repository, i.e. workflows and their implementations and configurations.

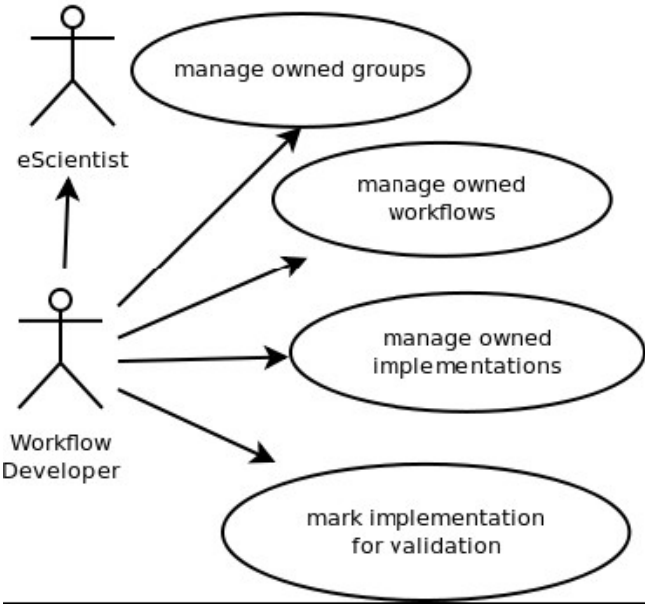


Figure 3: Workflow Developer use case

Browse workflows and implementations
In addition to being allowed to browse public workflows and implementations, as per the e-scientist, the workflow developer can also browse private workflows which are marked as being readable by others. The workflow developer will only be allowed to

download the files relating to such workflows, in the event that the private workflow is also downloadable by others.

Manage owned groups

Workflow developers can create user groups. They will own these groups, i.e. they will be the group leaders. They as group leaders can display, modify and delete these groups.

Manage owned workflows

Workflow developers are allowed to upload, modify, delete and download workflows the repository. Workflow developers can also manage workflows belonging to groups of which they are a member if group modification access being granted by the owner – this enables collaborative work on workflows.

Developers are allowed to upload, modify, delete and download files inputs associated to workflows.

Manage owned implementations

Workflow developers are allowed to upload, modify, delete and download workflow implementations. Workflow developers can also manage implementations of workflows belonging to groups of which they are a member if group modification access being granted by the owner – this enables collaborative work on implementations.

Developers are allowed to upload, modify, delete and download files inputs associated to implementations

Mark workflow or implementation for publication

Workflow owners are allowed to mark their private workflows as being public, thus making said workflow visible to e-scientists. Workflows may be published independently of their implementations.

Make implementation submittable

Workflow owners are allowed to configure a Submittable Execution Node (SEN) for public implementations they own. Once correctly configured, the implementation can be toggled to being submittable. From this point forward, the implementation will be submittable using the SHIWA Simulation Platform.

Review implementations

Workflow developers are allowed to review any readable implementations, by allocating the workflow upto 10 stars. This realises a community validation process.

2.4.3. Repository administrator

This actor is a system administrator has the highest role among the actors. His task is to maintain the SHIWA repository and to support all other users.

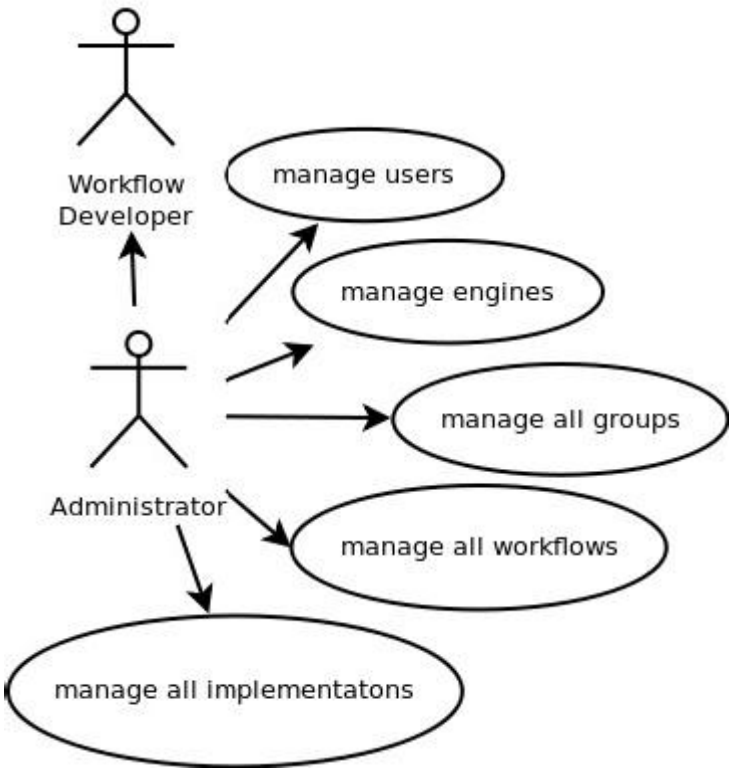


Figure 4: Repository administrator use case

Manage users
Administrators can create, delete or modify users.

Manage engines
Administrators can create, delete or modify workflow engines.

Manage domains

Administrators can add or remove domains or subdomains.

Manage all groups
Similarly to workflow developers administrators can create user groups.
They can display, modify and delete all groups, irrespective of the owner.

Manage workflows
Administrators are allowed to upload workflows into the repository and modify, delete and download all workflow in the repository.

Manage implementations
Administrators are allowed to upload implementations into the repository, modify, delete and download all implementations in the repository.

3. GUI STRUCTURE

Repository features can be accessed using the main menu on the top. The following chapters go through the Workflows and Implementations tabs and describe the provided functionality. Information related to the selected tab is displayed in a table. Rows of a table can be filtered by entering text into the text field below any column

title. Actions can be initiated using an “Actions” tab on the right. See illustration in Figure 5.

This is the GUI structure of the public view, which can be viewed without any login credentials.

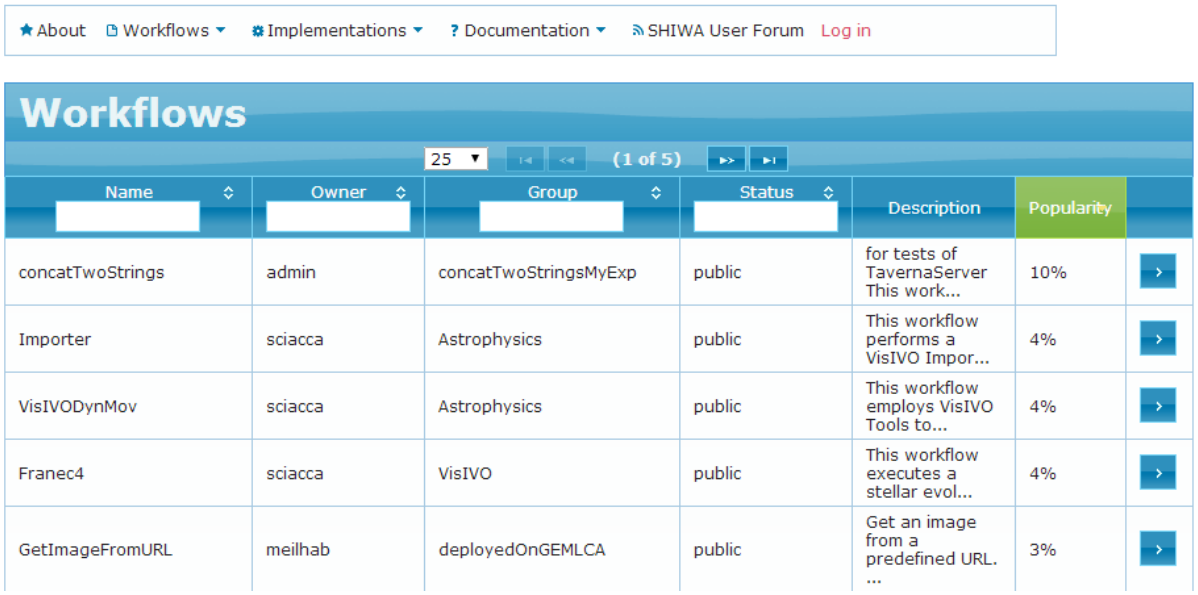


Figure 5: GUI Structure

The Home view interface can be used to list the user’s owned workflows, groups of which the member is a member of and groups the user owns. See section 6 for workflow management, and section 12 for Group management.

5. MANAGE USER PROFILE

The *My details* and *My password* tabs on Figure 6 can be used to manage the users details and to change the users password (see Figure 8 and Figure 9)

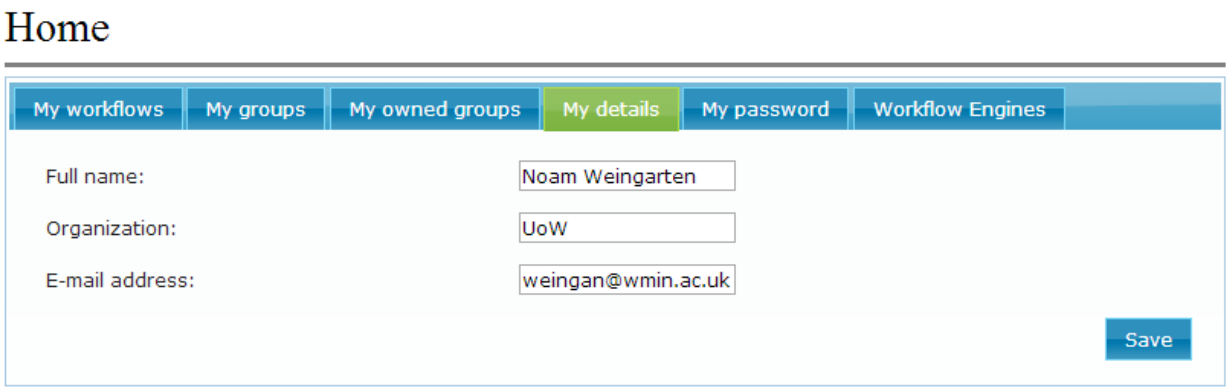


Figure 6: My Details tab

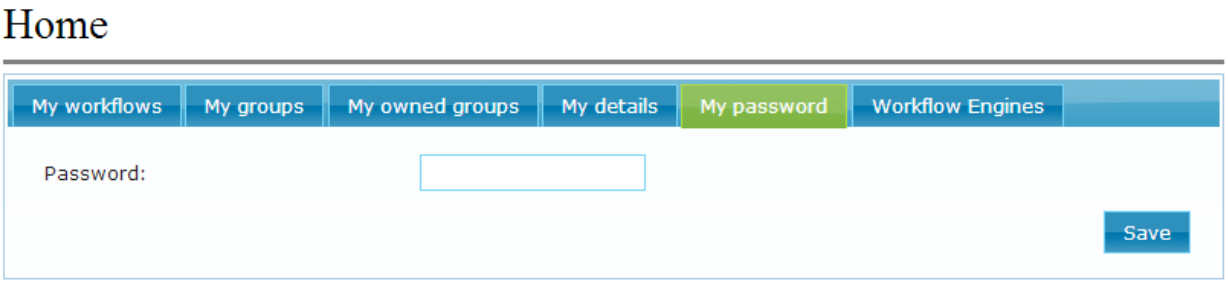


Figure 8: My Password tab

6. WORKFLOW MANAGEMENT

6.1. LIST AND SEARCH WORKFLOWS

6.1.1. WORKFLOW BROWSE VIEW

Figure 9 below, displays the Workflow Browse view. This view displays basic information about workflows and implementations in a convenient form. The readable workflows available will be; owned by the user logged in, readable by a group of which the user is a member, readable by others and public workflows. By clicking the Details button more information is displayed about the Workflow and all its implementations.

Inputs, Outputs and Datasets tabs can be further expanded, to display more information.

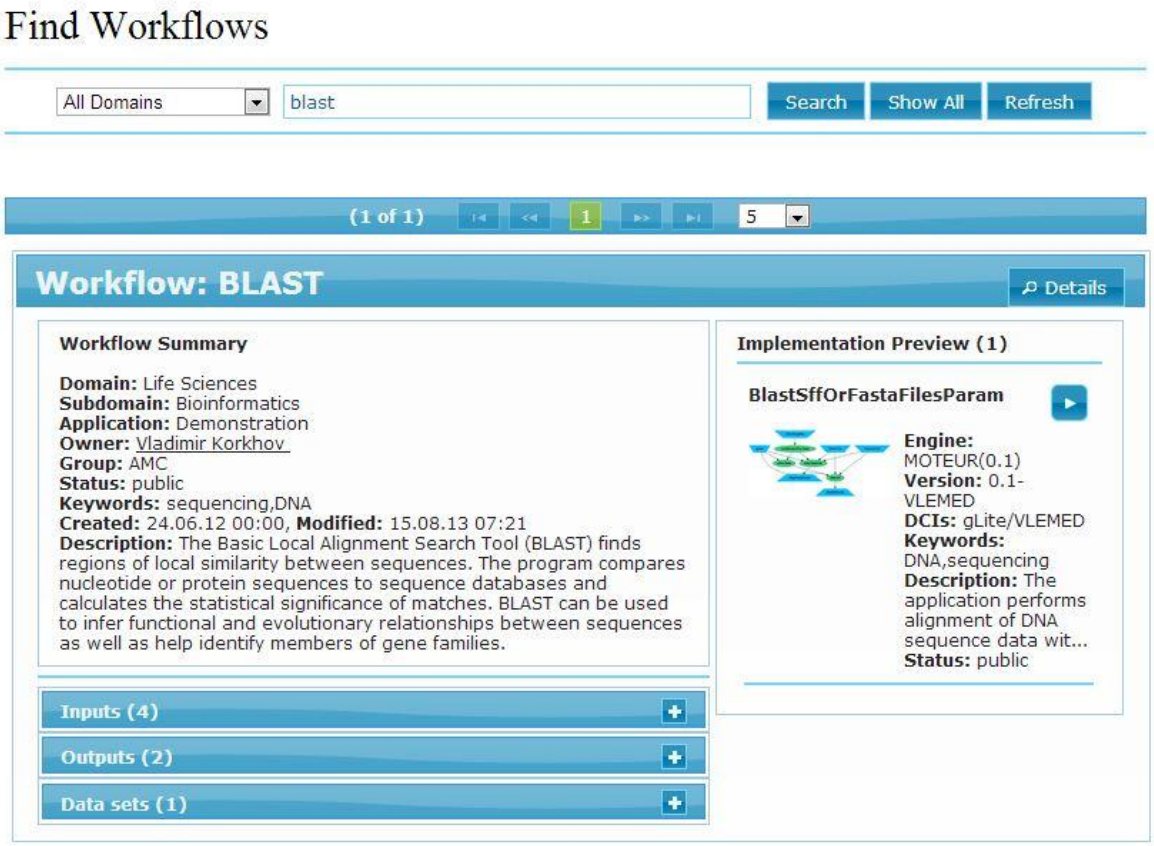


Figure 9: Workflows Browse view

Textual search in workflow or implementation records can be performed using the search box. The search operation can be restricted to application domains or sub-domains.

6.1.2. WORKFLOW TABLE VIEW (FIGURE 10)

The Workflows Table view can be used to list and filter readable workflows by Name, Owner or Group.

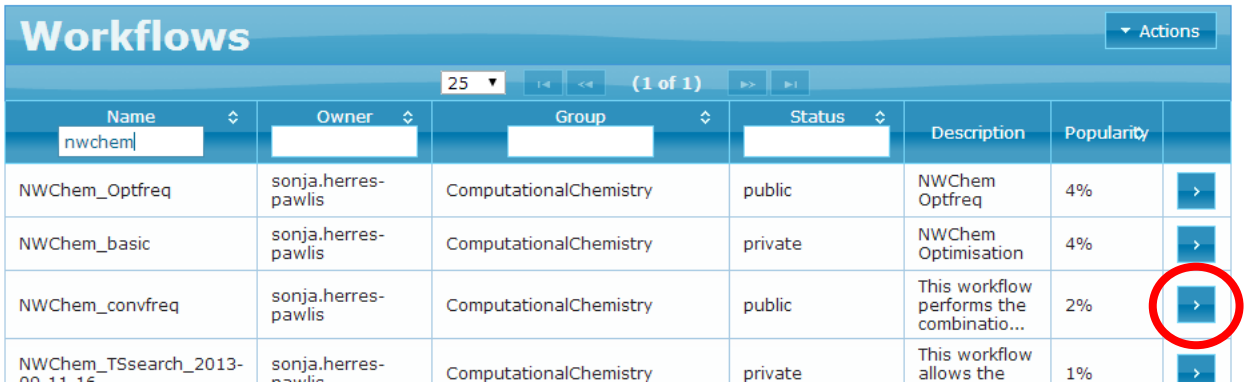


Figure 10: Workflows table view

All available information about workflows and their implementations can be viewed by selecting the workflow in this view (see Workflow details (Figure 11),

Workflow Attributes (Figure 12), Workflow files Download (Figure 13) and Workflow Implementations (Figure 14))

To select a workflow click the Select Icon  of the workflow of interest. The icon has been circled in red in Figure 10.

The Action->New can be used to create new workflows (see 6.2 Create Workflows)

6.1.3. WORKFLOW DETAILS (FIGURE 11)

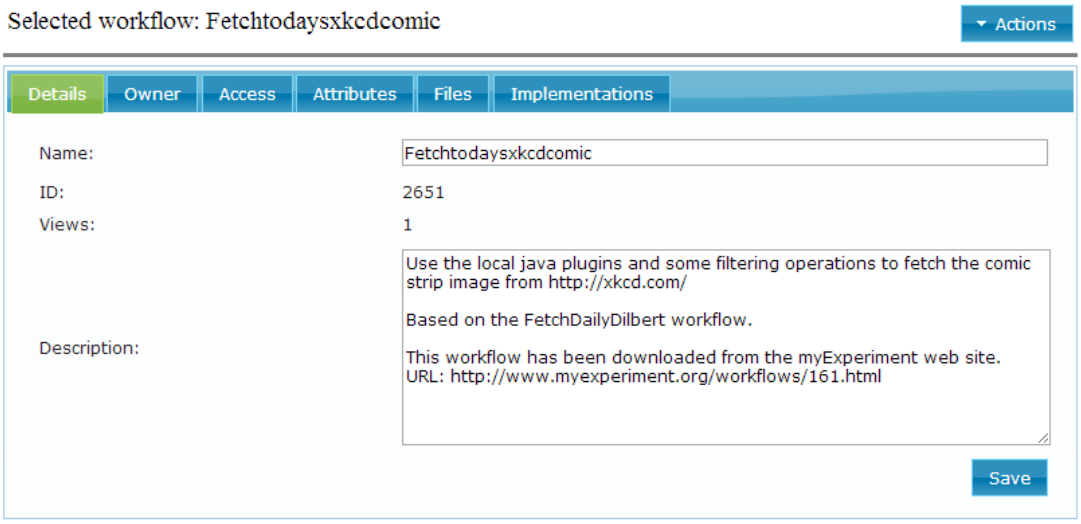
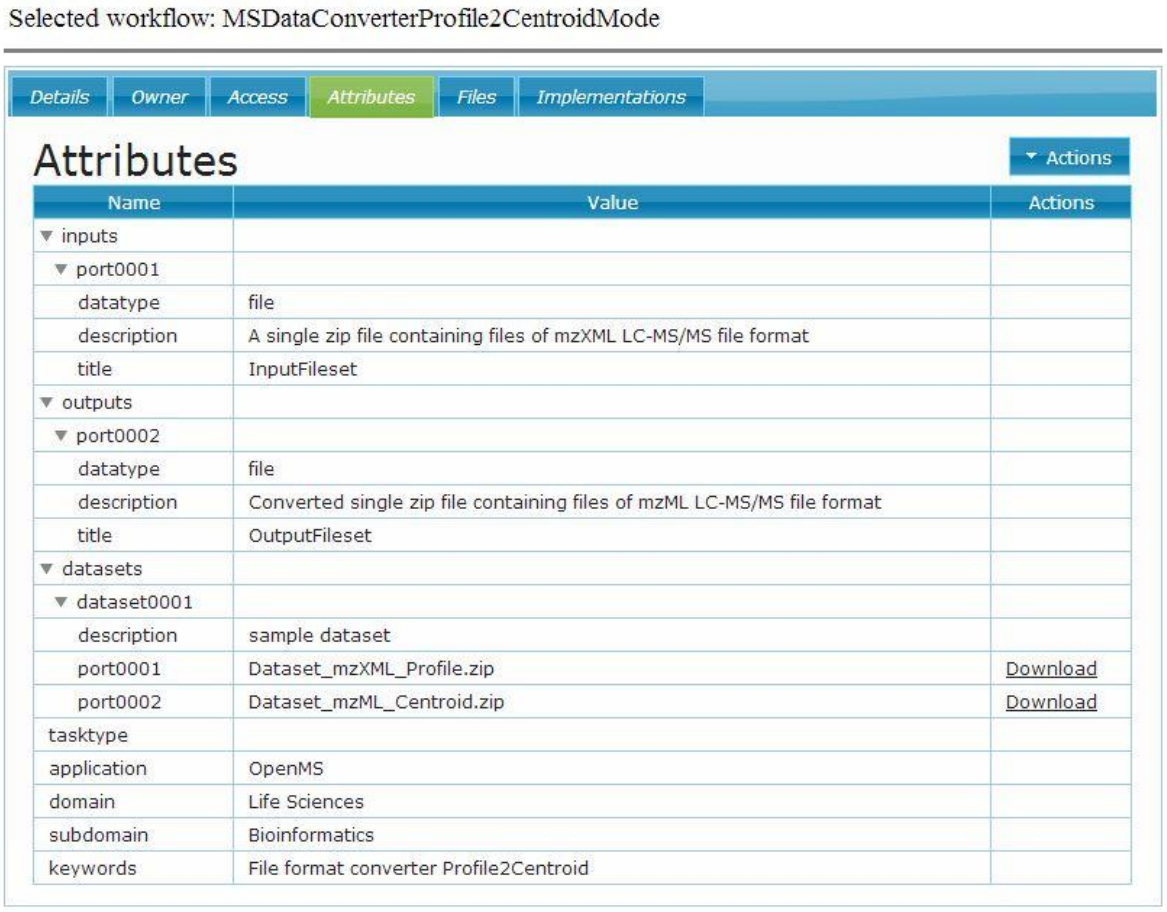


Figure 11: Workflow details page

6.1.4. WORKFLOW ATTRIBUTES (FIGURE 12)

Workflow attributes can be listed by clicking on the Attributes tab of a particular Workflow.



Name	Value	Actions
▼ inputs		
▼ port0001		
datatype	file	
description	A single zip file containing files of mzXML LC-MS/MS file format	
title	InputFileset	
▼ outputs		
▼ port0002		
datatype	file	
description	Converted single zip file containing files of mzML LC-MS/MS file format	
title	OutputFileset	
▼ datasets		
▼ dataset0001		
description	sample dataset	
port0001	Dataset_mzXML_Profile.zip	Download
port0002	Dataset_mzML_Centroid.zip	Download
tasktype		
application	OpenMS	
domain	Life Sciences	
subdomain	Bioinformatics	
keywords	File format converter Profile2Centroid	

Figure 12: Workflow attributes page

The Expand/Collapse button in the Action control can be used to display all the attributes of the workflow. Annex 1 describes the metadata structure of the attributes, and provides example values Annex 2 presents the Workflow metadata structure. These attributes allow straightforward categorisation of workflows and improve the browsing and search operations significantly. The input and output attributes with their sub-attributes define inputs and outputs of the Workflow. The dataset attribute specifies values of input parameters passed to workflow inputs, and they can also specify example outputs.

WORKFLOW FILES DOWNLOAD (FIGURE 13)

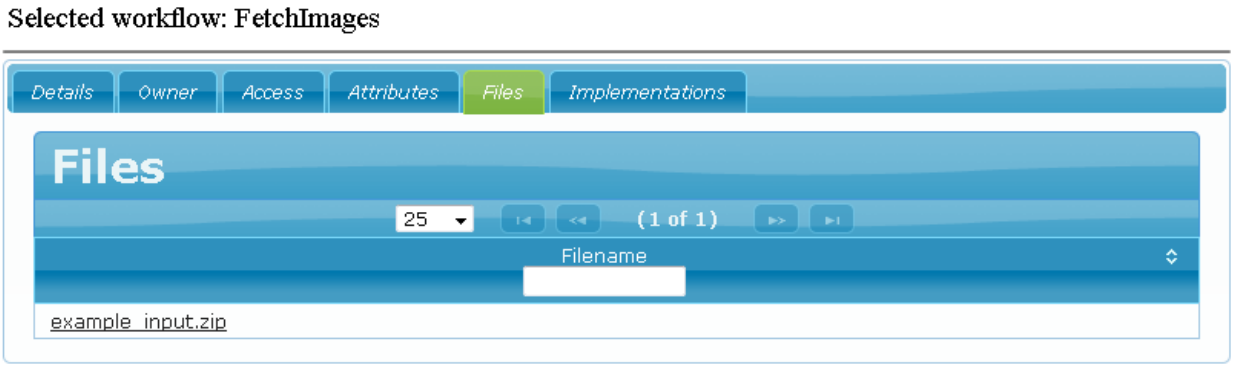


Figure 13: Workflows files page

Files associated with the abstract workflow definition can be downloaded from this page by right-clicking the files and selecting “Save as ….” or appropriate, as per your browser.

If files are not permitted to be downloaded, only their names will be displayed, but no live-links for download will be generated.

6.1.5. WORKFLOW IMPLEMENTATIONS (FIGURE 14)

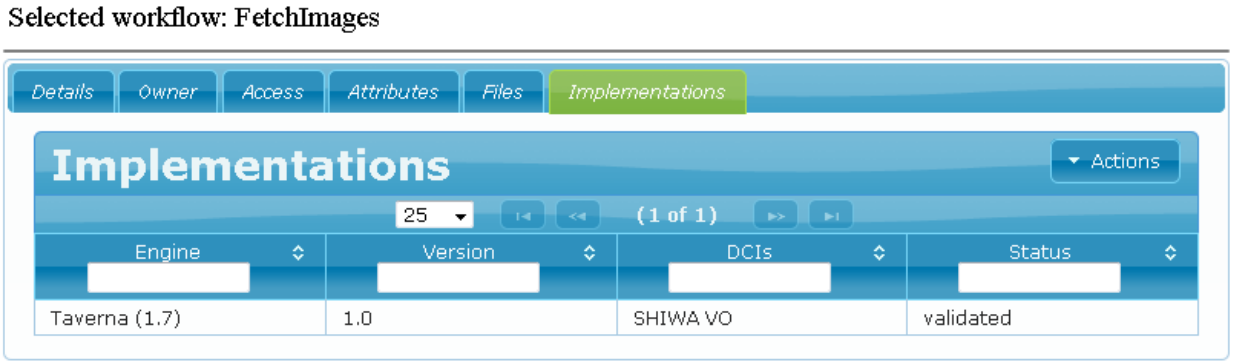


Figure 14: Workflows Implementations page

This page lists the implementations of the selected workflow.

Selecting any of the implementations directs users to the implementation details of that Implementation (see Implementations details).

6.2. CREATE WORKFLOWS

Workflows can be created by selecting the *Create workflow* option from the toolbar (see Figure 15) and the New Workflow interface should be used to create the workflow (see Figure 16).

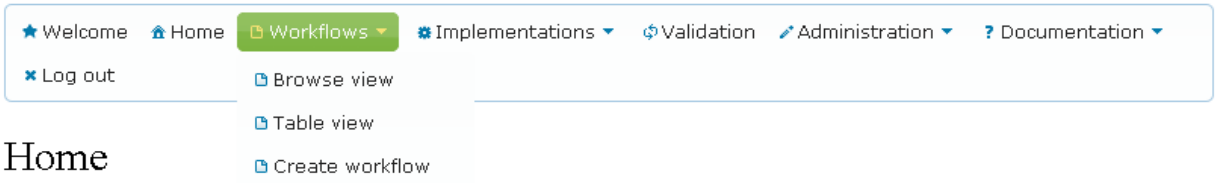


Figure 15: Create workflow

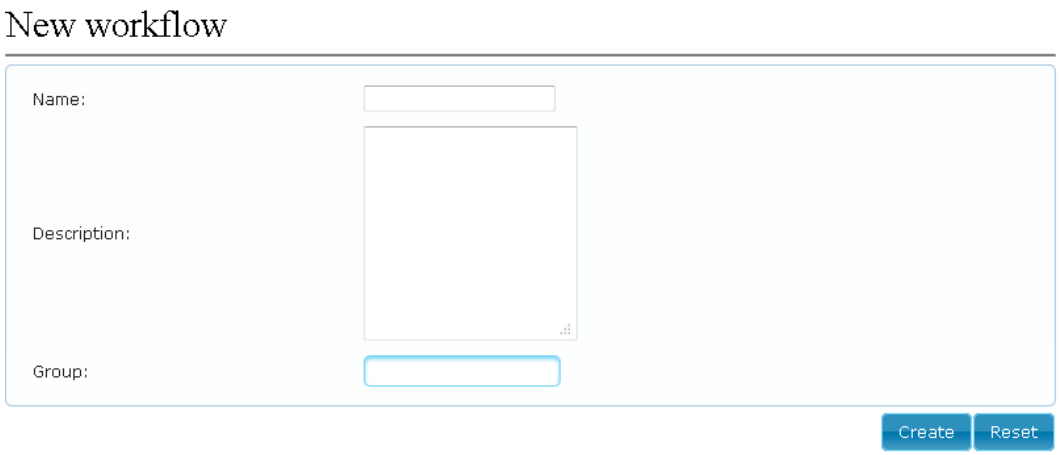


Figure 16: New Workflow interface

Notes:

- Workflow names can only contain alphanumeric, "-" and "_" characters and must be between 3 and 250 characters long.
- Workflows must be associated with a group. A workflow may only be associated with a group of which the owner is a member.

If you haven't created a group yet, you can do so from the Home > My owned groups page, using the Actions>New button. Alternatively contact an existing group's owner to add you as a member.

Note: Creating a group, and thereby becoming the owner of said group, will not automatically give you membership of the group. You will have to add yourself to said group.

The Workflow is now created, and can be modified (see below).

6.3. MODIFY/DELETE WORKFLOW

Selecting an owned or modifiable workflow from the Table view of workflows (see Figure 10) or from the user's own My Workflows view.

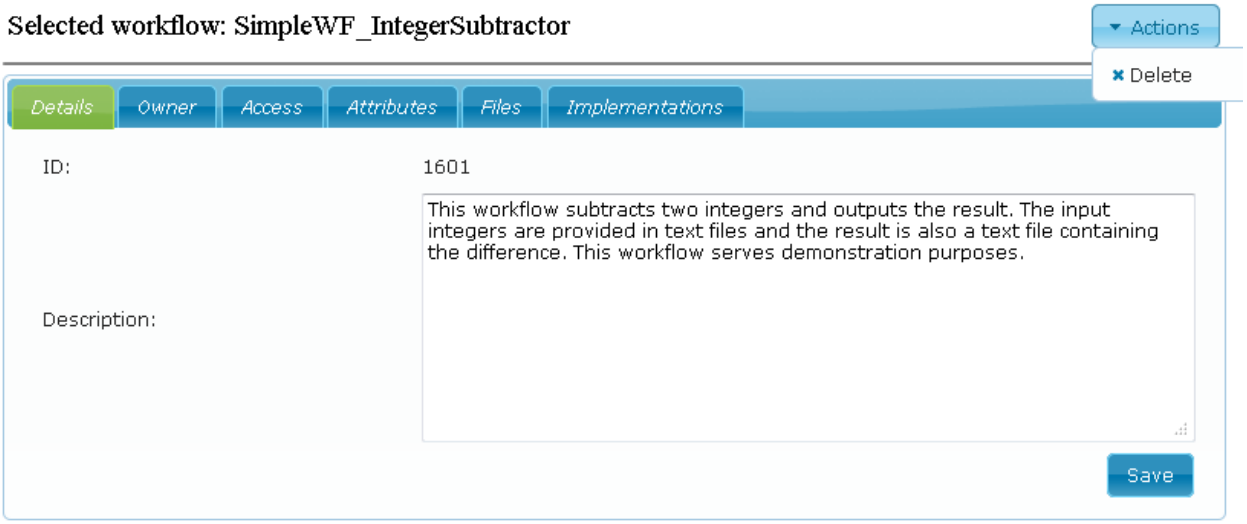


Figure 17: Workflow details – Details tab

The Workflow Details tab can be used to modify the workflow description.

Note: the Action button can be used to delete the workflow.

OWNERSHIP

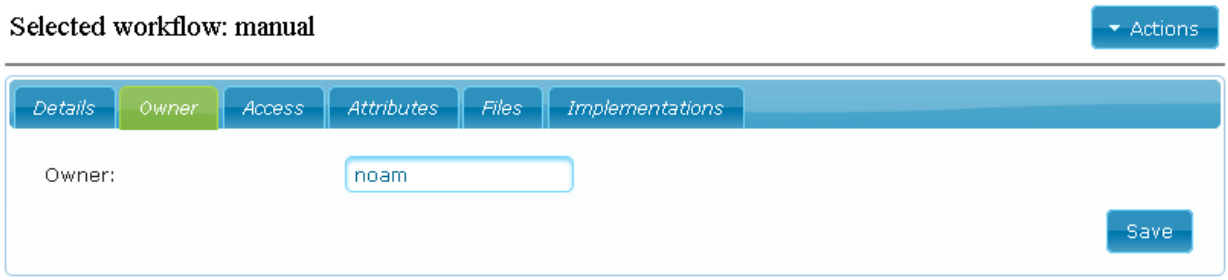


Figure 18: Workflow Owner – Owner tab

The Workflow Owner tab can be used to change ownership of the workflow

6.3.1. ACCESS CONTROL

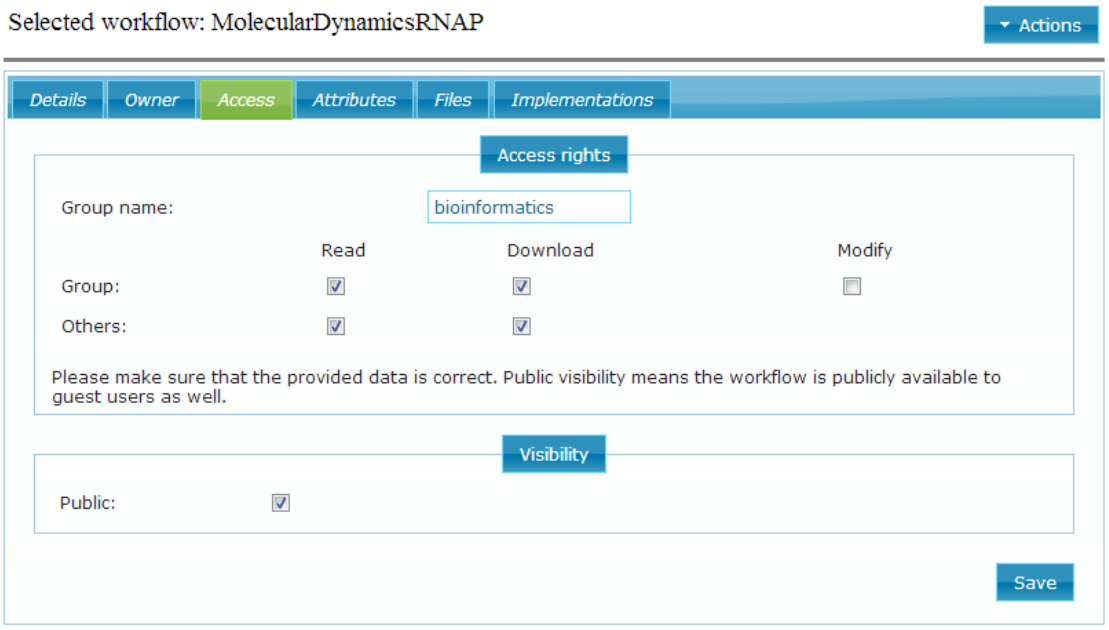


Figure 19: Workflow Access Control – Access tab

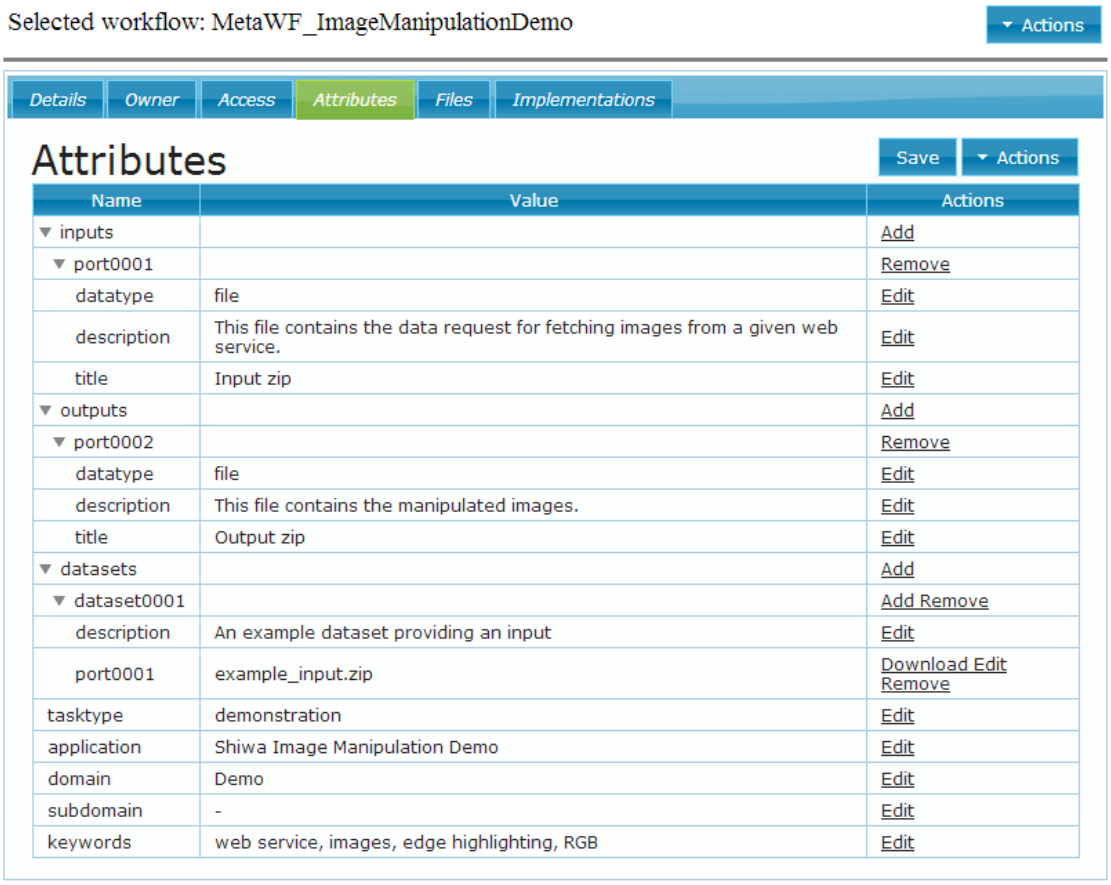
The Workflow Owner or an Administrator may modify the access control as required.

Workflows marked as being readable, downloadable and/or modifiable by the group, will afford respective rights members of said group.

Workflows marked as being readable and/or downloadable by others, will afford respective rights to any workflow developer who has logged in to the repository.

Workflows marked as being public will be readable and downloadable by any e-scientist, whether they have signed into the repository or are using the repository in public mode.

6.3.2. ATTRIBUTES



Name	Value	Actions
inputs		Add
port0001		Remove
datatype	file	Edit
description	This file contains the data request for fetching images from a given web service.	Edit
title	Input zip	Edit
outputs		Add
port0002		Remove
datatype	file	Edit
description	This file contains the manipulated images.	Edit
title	Output zip	Edit
datasets		Add
dataset0001		Add Remove
description	An example dataset providing an input	Edit
port0001	example_input.zip	Download Edit Remove
tasktype	demonstration	Edit
application	Shiwa Image Manipulation Demo	Edit
domain	Demo	Edit
subdomain	-	Edit
keywords	web service, images, edge highlighting, RGB	Edit

Figure 20: Workflow Attributes – Attributes tab

Workflow attributes can be specified using this interface.

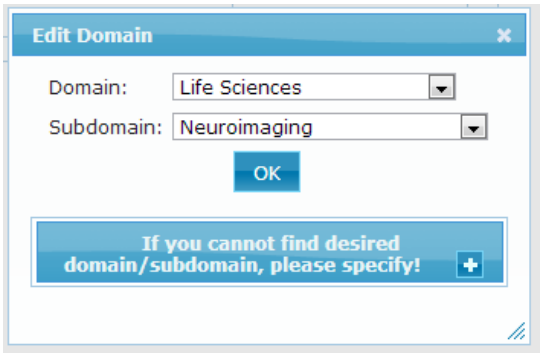


Figure 20A: Workflow Attributes – Domain selector

The workflow domain and subdomain should be selected using the above interface. This interface is accessible by editing either the domain or subdomain. Should the desired domain or subdomain not already be listed, please use the extra interface to request inclusion of the desired domain or subdomain.

6.3.3. FILES

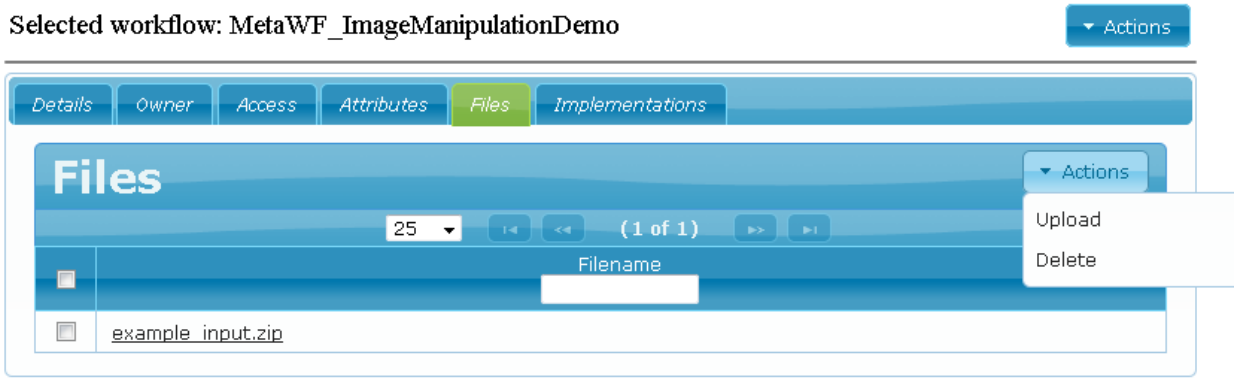


Figure 21: Workflow Files – Files tab

Files can be uploaded to or deleted from the Workflow using the above interface.

6.3.4 IMPLEMENTATIONS

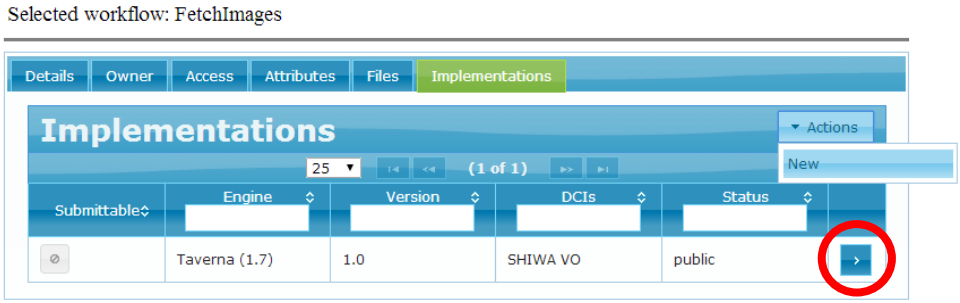


Figure 22: Workflows Implementations - Implementation tab

Implementations of workflows can be created using this tab, as will be explained below.

To select an Implementation click the Select Icon of the workflow of interest. The icon has been circled in red in Figure 22.

7. IMPLEMENTATION MANAGEMENT

7.1. LIST AND SEARCH IMPLEMENTATIONS

7.1.1. IMPLEMENTATION BROWSE VIEW

Figure 23 below, displays the Implementation Browse view. This view displays most of the information about available implementations in a convenient form. Dependencies and Configurations can be expanded to display more information.

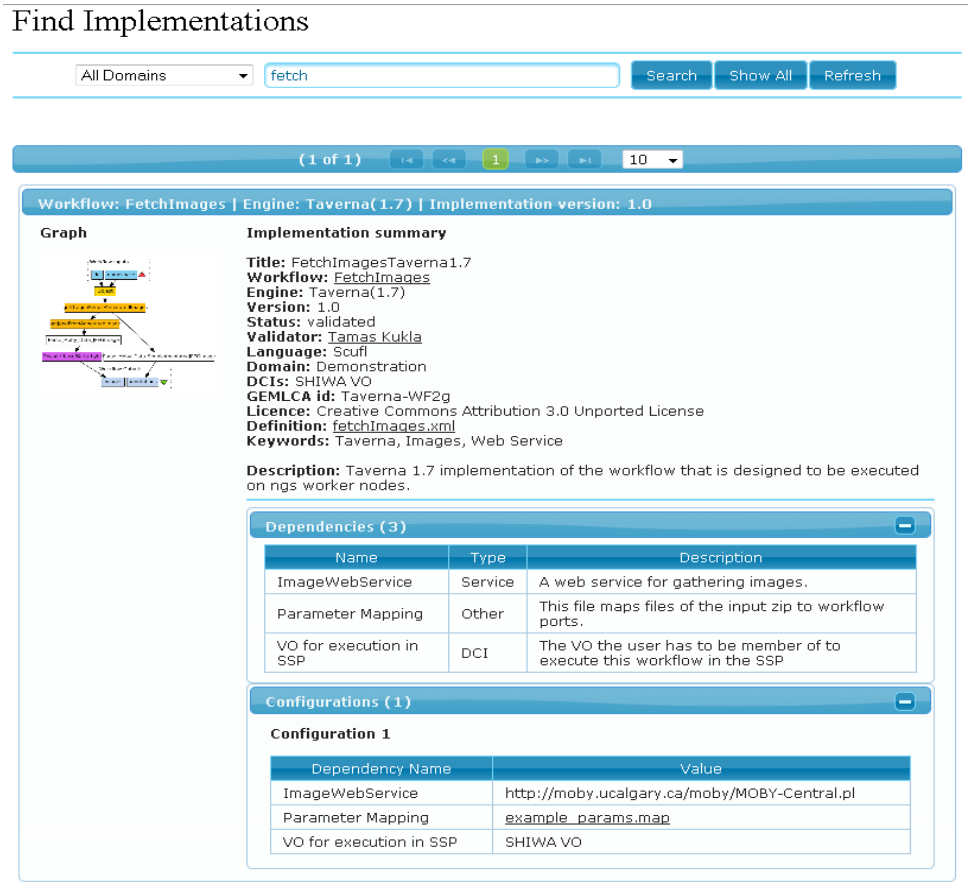


Figure 23: Browse Implementation view

Textual search in workflow implementation records can be performed using the search box. This search can be restricted to specific application domains or subdomains.

Workflow developers are able to submit reviews of public implementations owned by other developers, this helps to maintain a level of objectivity of the community validation. The displayed rating of an implementation is the average of the most recent rating given by any developer who has reviewed this implementation. If a developer gives a second review of an implementation, the previous review will be discarded.

7.1.2. IMPLEMENTATION TABLE VIEW

The Implementations Table View can be used to list and filter available Implementations by Workflow, Engine, Version or DCI.

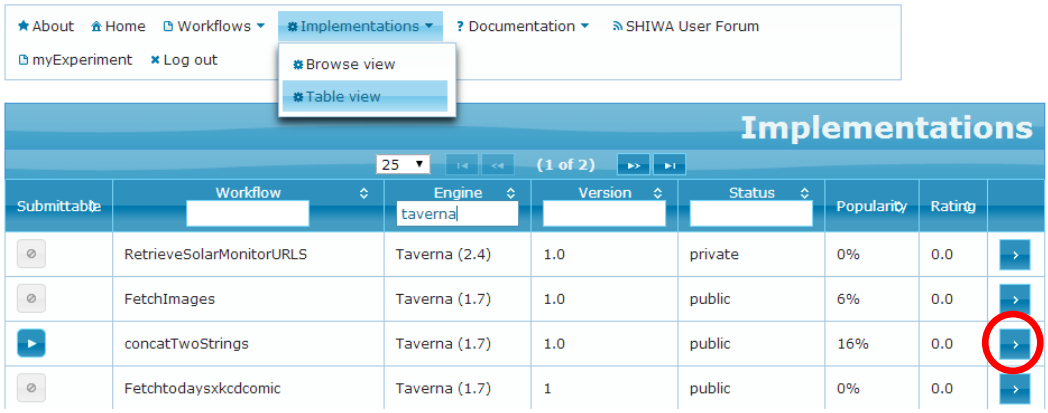


Figure 24: Implementations Table view

All available information about Implementations can be viewed by selecting the Implementation from this view (see sections 7.1.3, 7.1.4, 7.1.5)

To select an Implementation click the Select Icon ▸ of the workflow of interest. The icon has been circled in red in Figure 24.

7.1.3. IMPLEMENTATIONS DETAILS

The implementation's details page is illustrated in Figure 25.

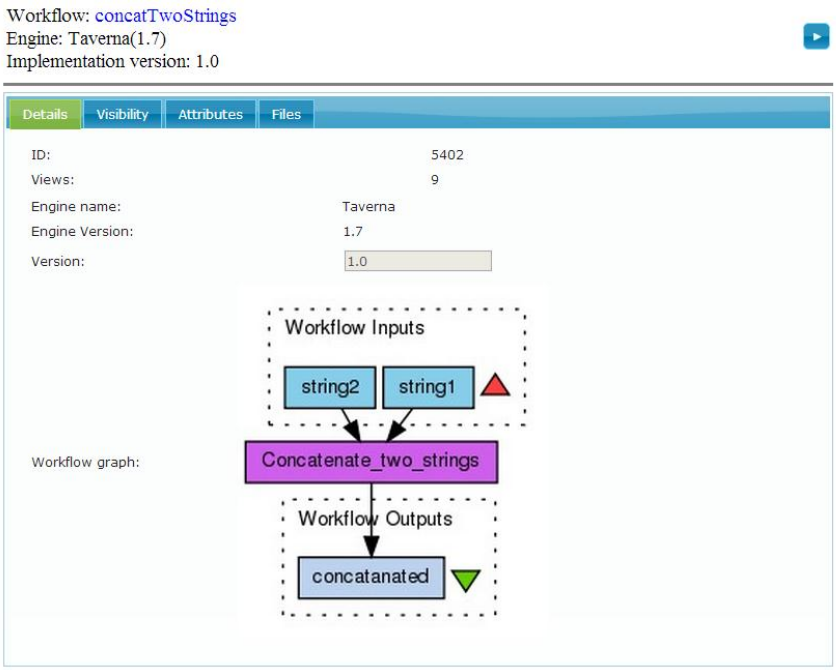


Figure 25: Implementation Details page

The *Attributes* tab can be used to open the attributes of the selected Implementation.

The *Files* tab can be used to list and download files held in the repository for the selected implementation.

7.1.4. IMPLEMENTATION ATTRIBUTES

Implementation attributes can be opened by clicking on the attributes tab of a given implementation as illustrated in Figure 26.

The left column of the attributes table contains attribute names, while the right column contains attribute values.

Similarly to workflows, the metadata template is used to kelp the definition of most common attributes. The three key attributes are: definition, dependencies and configurations. The definition attribute, is the workflow definition file i.e. the executable to be interpreted by the workflow engine. The dependency attribute can be any requirement of the particular implementation. These can include for instance files, executables, libraries or VO memberships required for execution. Configuration attributes resolve these dependencies.

Name	Value	
dependencies		
dep0001		
type	Service	
description	A web service for gathering images.	
title	ImageWebService	
dep0002		
type	Other	
description	This file maps files of the input zip to workflow ports.	
title	Parameter Mapping	
dep0003		
type	DCI	
description	The VO the user has to be member of to execute this workflow in the SSP	
title	VO for execution in SSP	
configurations		
conf0001		
dep0001	http://moby.ucalgary.ca/moby/MOBY-Central.pl	
dep0002	example_params.map	Download
dep0003	SHIWA VO	
title	FetchImagesTaverna1.7	
description	Taverna 1.7 implementation of the workflow that is designed to be executed on ngs worker nodes.	
definition	fetchImages.xml	Download
graph	Taverna_wf2.png	Download
language	Scufl	
rights		
licence	Creative Commons Attribution 3.0 Unported License	
keywords	Taverna, Images, Web Service	
uuid		

Figure 26: Implementation attribute table

The definition file can be downloaded from this page by clicking on the *download* live-link.

7.1.5. IMPLEMENTATION FILES DOWNLOAD

As with Workflow files, these can be downloaded from the Implementation Files view, by clicking on the respective live-link (see Figure 27).

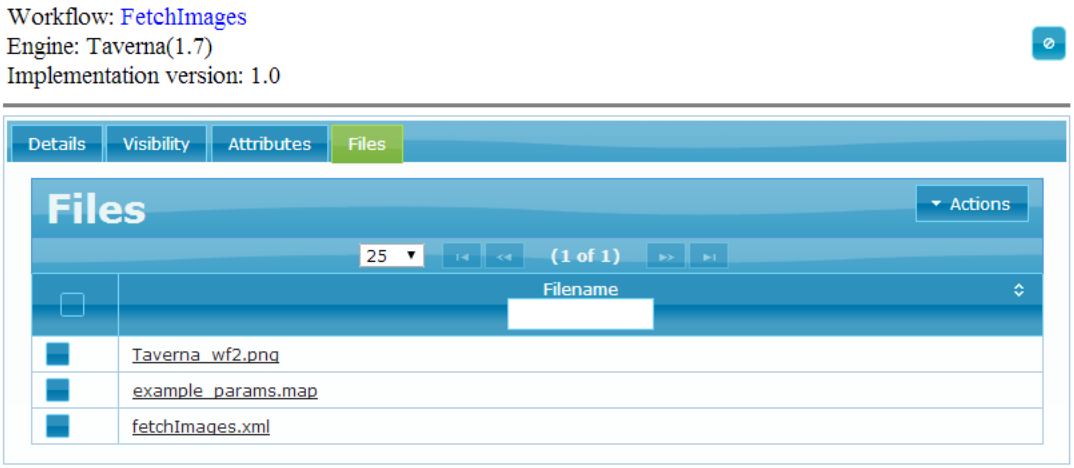


Figure 27: Implementation Files – Files tab

7.1.6. CREATE IMPLEMENTATION

(See Section 6.3.4

6.3.4 Implementations)

7.2. MODIFY IMPLEMENTATION

7.2.1. DETAILS

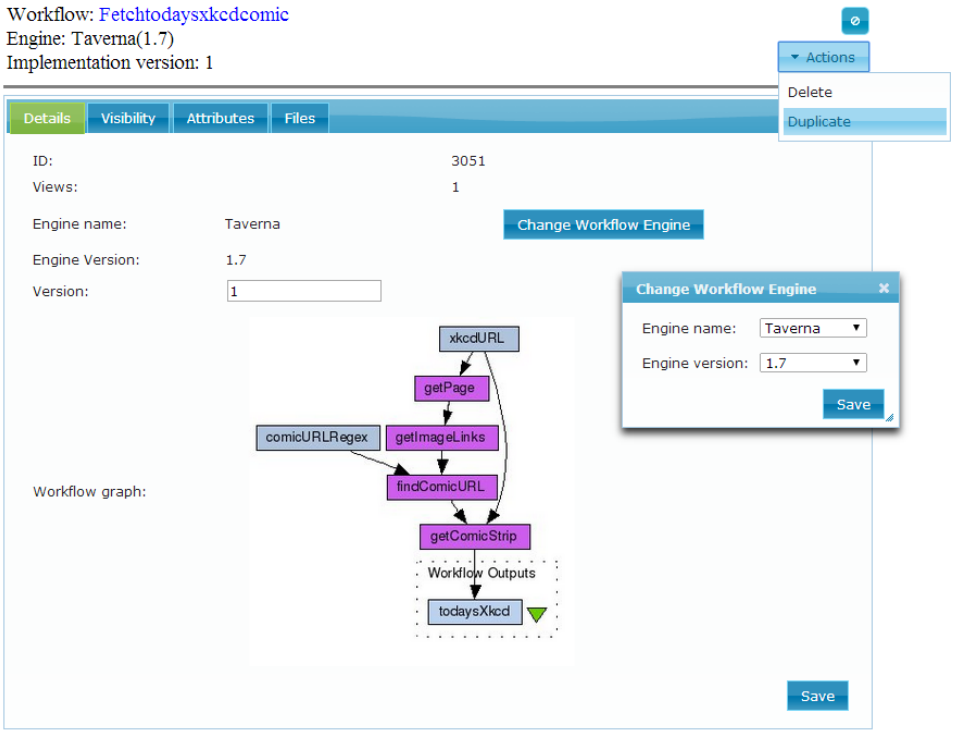


Figure 28: Modify Implementation details - Details tab

The above interface can be used to modify the version of implementations the user has permissions to modify.

The Actions button can be used to delete or duplicate the implementation.

The Change Workflow Engine button can be used to modify the Engine and Engine version. However it is recommended that should you wish to modify Implementations, the Implementation should be duplicated, and the duplicate modified.

7.2.2. ATTRIBUTES

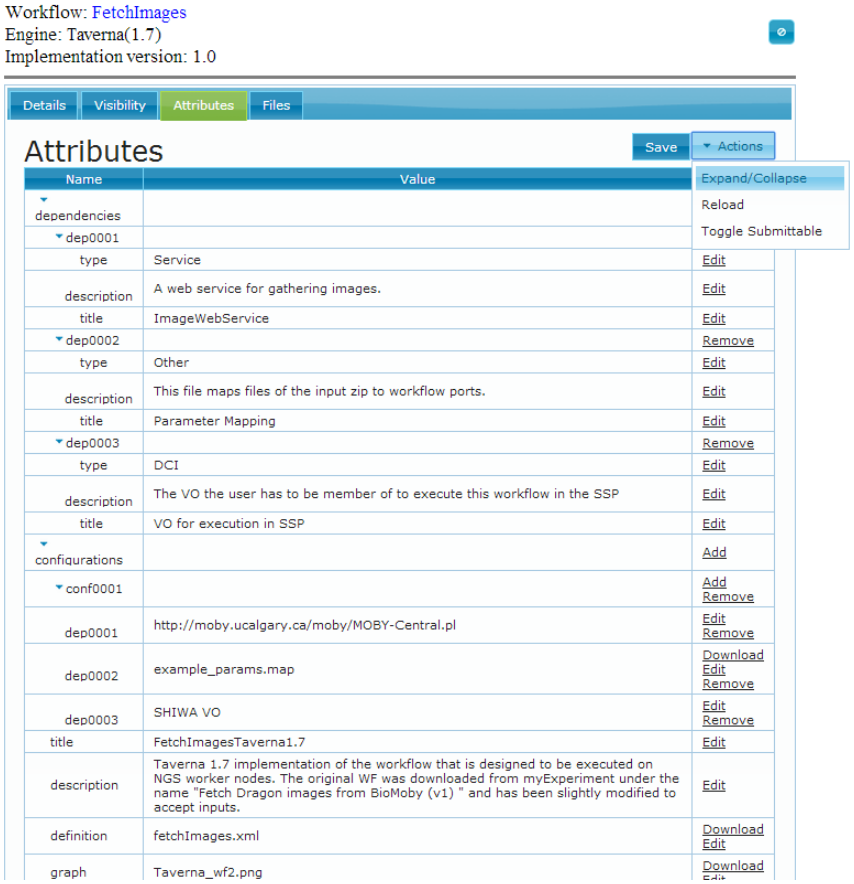


Figure 29: Modifying implementation attributes – Attributes tab

The Expand/Collapse action can be used to expand or collapse the listing.
The Add, Edit or Remove links can be used to add, edit or remove attributes respectively.
The Reload action can be used to reload the attributes.
The Save button should be used if modifying or adding any attributes.
The Toggle Submittable action is discussed further in Section 9.1.

7.2.3. FILES

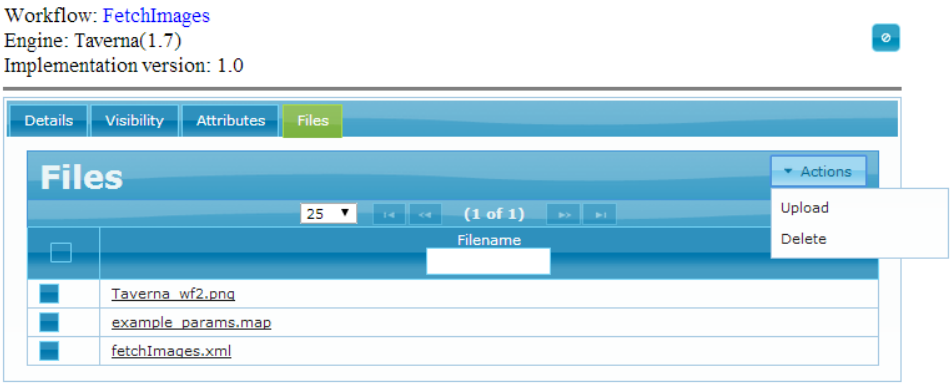


Figure 30: Modifying implementations - Files tab

7.2.4. GRAPH IMAGES OF IMPLEMENTATIONS

A screenshot of the implementation can be made using the native workflow graphical editor. The screenshot can be uploaded in the repository and aligned with the implementation (see Figure 28). The image should be either a jpg, png or gif.

The required resolution can depend on the workflow complexity - note that the thumbnail images will have a width of 150 pixels, and so the resolution should be sufficient to provide a recognizable thumbnail at 150 pixels across.

The Image file should be uploaded to the implementation, as a file (see Figure 30)

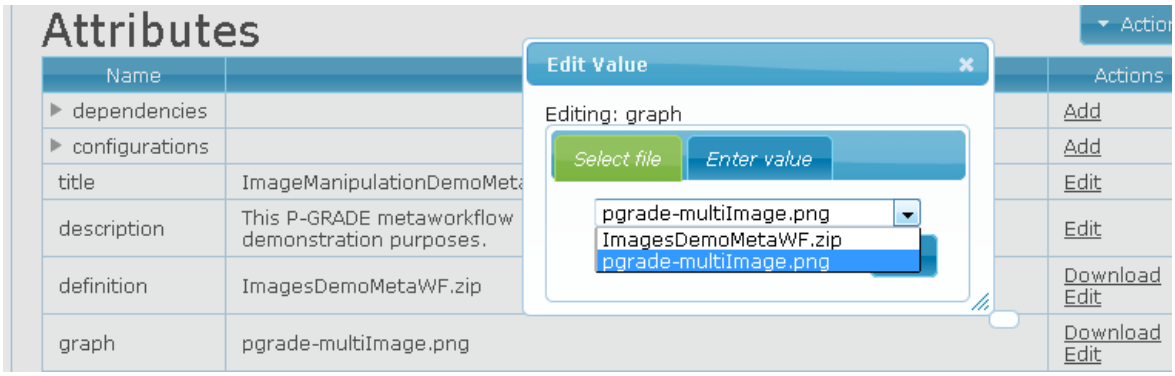


Figure 31: Selecting graph image file

Once uploaded, the file can be selected from the uploaded implementation files, on editing the graph attribute of the said implementation (see Figure 31).

7.2.5. VISBILITY

The Visibility tab can be used to view and modify the publication status of the implementation. Unless an implementation is marked as public, only the owner and users allowed to view or modify its parent abstract workflow will be able to view or modify it. An implementation marked as public, will be visible to all e-scientists, whether logged in or not.

Only implementations marked as submittable will be embeddable in the SSP from the SHIWA repository for execution. Only implementation marked as public will be able to be toggled to submittable – other requirements apply (see Section 9.1)

7.2.6. WORKFLOW EXECUTION

A workflow's implementation can be “platform enabled” by specifying appropriate execution details in the Submission Execution Node. Once this is completed, the implementation can be toggled to Submittable the workflow can be executed from within the SHIWA Simulation Platform.

For more information and prerequisites, see section 9 .

8. IMPORT WORFLOWS FROM MYEXPERIMENT

8.1. PREFACE

The SHIWA Repository is connected to myExperiment which is a social networking site and Virtual Research Environment (VRE) designed for people to share, discover and reuse workflows.

The SHIWA Repository GUI is extended to facilitate browsing and importing publicly shared Taverna1 and Taverna2 workflows from the myExperiment site. This feature of the SHIWA Repository can be accessed using the main menu bar on the top. This feature of the GUI can only be viewed after logging in (see Figure 32Figure 37).

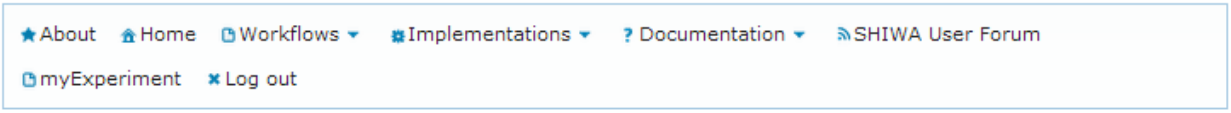


Figure 32: Accessing myExperiment integration feature

The following chapters explain how to import workflows from myExperiment to the SHIWA Repository and describe the provided functionality.

8.2. IMPORTING WORKFLOWS

All available information about finding and importing workflows from myExperiment to the SHIWA Repository can be viewed by selecting “myExperiment” from the menu bar on the top. The provided link (http://www.myExperiment.org) on the page can be used to browse publicly shared workflows on the myExperiment site.

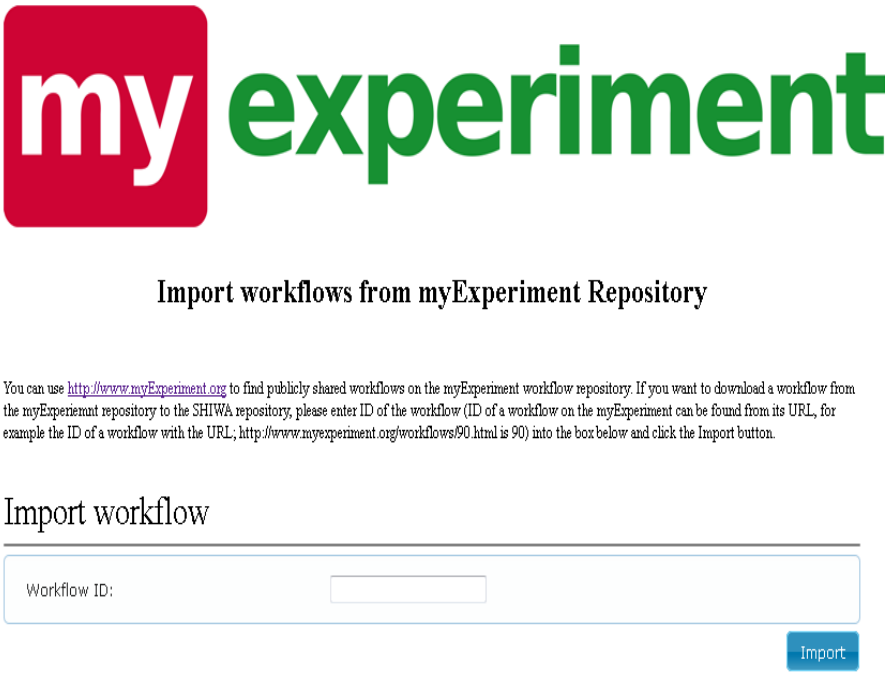


Figure 33: my Experiment workflow import page

A workflow can be downloaded from the myExperiment site to the SHIWA Repository by entering the workflow ID (ID of a workflow on the myExperiment site can be found in its URL, for example the ID of a workflow with the URL; http://www.myExperiment.org/workflows/90.html is 90) into the workflow ID box and clicking the “Import” button (see Figure 33).

By entering the ID a workflow into the provided box and clicking the “Import” action button, the workflow can be imported from myExperiment site and a new workflow and implementation can be created automatically in the SHIWA Repository. and the required workflow files are uploaded. Messages are displayed on the top of the page about the success or failure of the operation (see Figure 39).

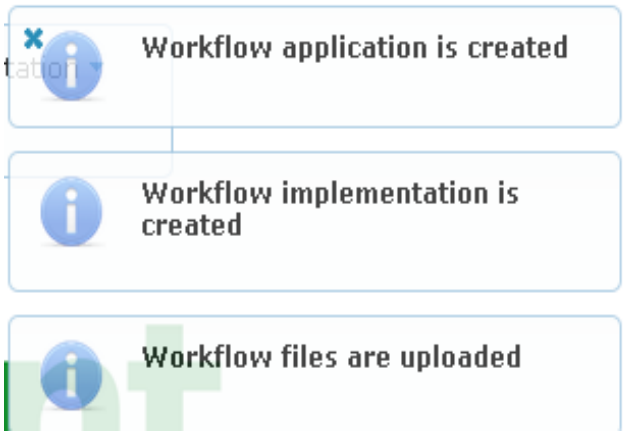


Figure 34: Importing a workflow from myExperiment

The newly created workflows can be found using Workflows Table or Workflows Browse view. The workflow names are changed slightly by removing empty spaces (see 35):

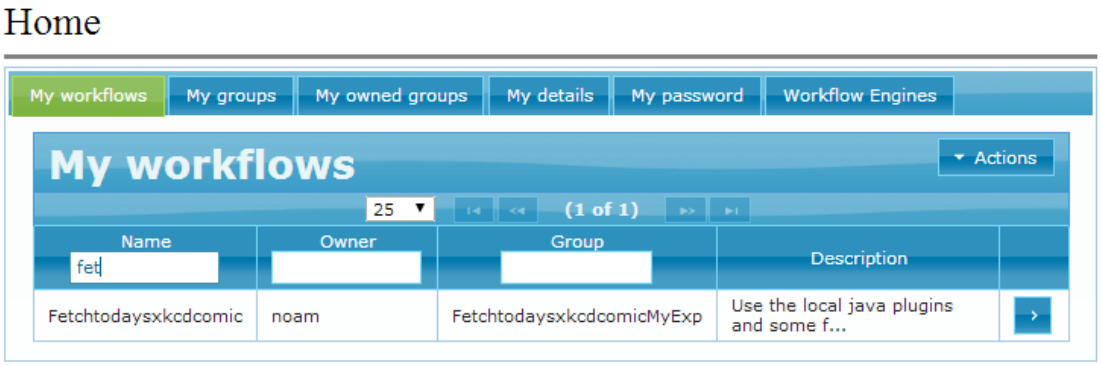


Figure 35: Newly created workflow

8.3. INPUT PORTS CONFIGURATION

If a workflow has input ports, the input data should be put in files and uploaded to the workflow application. The correspondent input ports are configured using the uploaded files (see Figure 36).

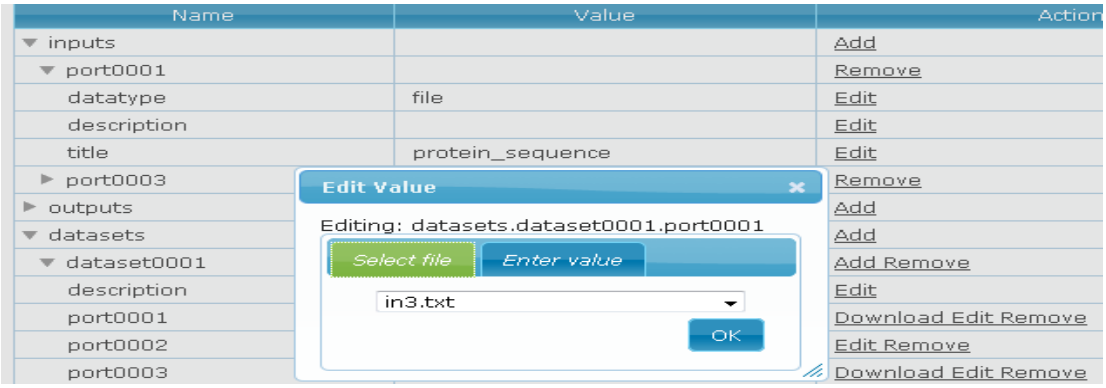


Figure 36: Configuring input ports

9. SUBMISSION SERVICE EXECUTION

The following chapter outlines the creation of Submission Execution Nodes (SEN), and in turn enabling the submission of implementation to the Submission Service in order to run a workflow implementation in the SSP.

9.1. PREFACE

In order to run a workflow implementation in the SSP, once a workflow and implementation have been correctly configured and SEN created, the implementation must be toggled to submittable.

This process will enable the execution of the workflow implementation using the SHIWA Portal.

The List of Workflow Implementations (see Figure 37) displays the implementation's execution status. A workflow is list submittable if the workflow implementation has been already toggled to submittable. The submittable status indicates that a workflow implementation can be executed on the SSP.
Submittable implementations are indicated by the blue icon in the left column.

Figure 37: Implementations list

In order for implementations to be deployed as submittable, several pre-requisites must be fulfilled:

- Implementation must be set for a submittable workflow engine.
  - The Submitability of a workflow engine is indicated in the Implementation's page by the icon, as in Figure 38.
  - This icon will be grey if the workflow engine is not marked as submitable through the SSP.
- The Implementation of the Workflow and the Workflow itsself must be public (see 7.2.5).
- The Implementation must have a configured Submission Execution Node associated with it. (see 9.2).
- The Implementation must have a Workflow Definition file associated with it (see 7.2.2).

9.2. CREATE SUBMISSION EXECUTION NODE (SEN)

Once the SEN of a workflow implementation has been correctly and completely described in the repository, it is ready to be toggled to submittable and made available for execution through the SHIWA Simulation Portal. The workflow implementation's SEN can be created and configured from the Implementation's Attributes page (see Figure 38).

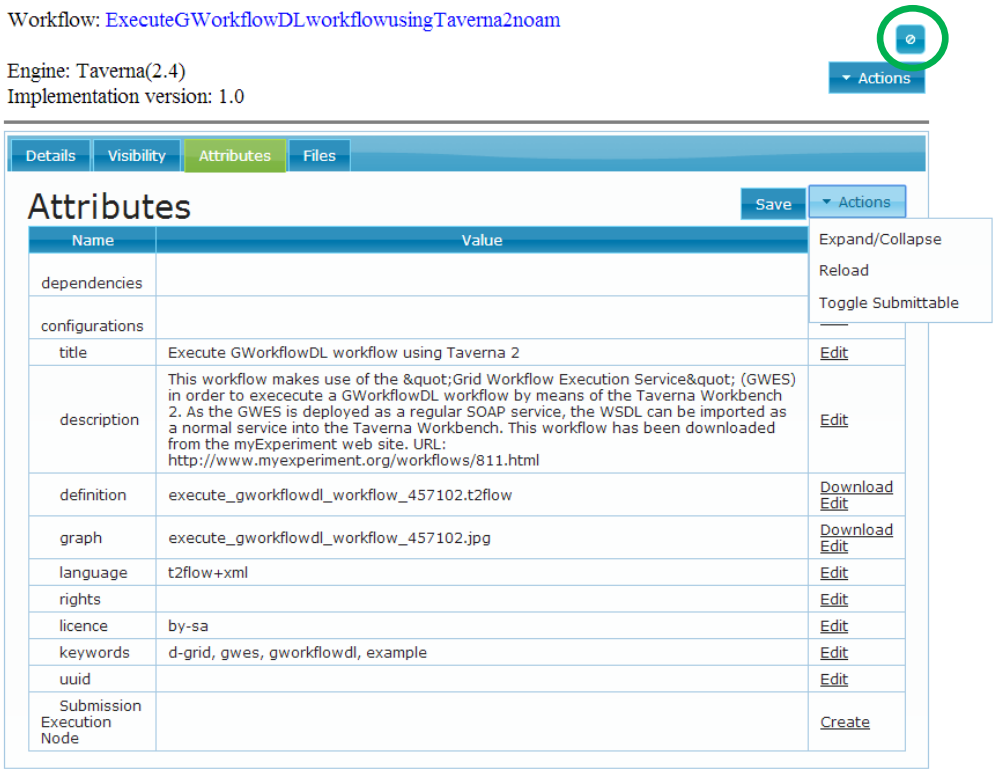


Figure 38: Implementation Attributes

On clicking the "Create" action next to the attribute "execution", a dialog box is displayed.

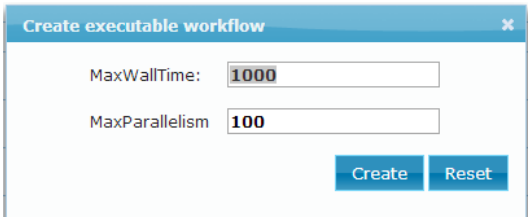


Figure 39: SEN creation dialog

Advanced users can configure details of the execution back-end at this stage (see Figure 39):

- **Maximum Walltime**: This is the maximum execution time in minutes – after which the execution will be suspended even if it is still running.
- **Maximum Parallelism**: This is the maximum number of parallel jobs of a process

The "Create" button will create the workflow implementation's execution.

9.3. CONFIGURE PARAMETERS

▾ Submission Execution Node		Remove
maxWallTime	1000	Edit
maxParallelism	100	Edit
parameters		Add

Figure 40: Configured Execution Back-end

The execution section of the Implementation's Attributes can now be expanded to add and define parameters by selecting the "Add" tab (see Figure 40).

First, the user should drag down the ParameterID dialog, to select a unique parameter number. Next, the user should select the type of parameter they are describing. Execution parameters can be categorized into 4 types (see Figure 41):

- **INPUT_PORT** and **OUTPUT_PORT** types of parameters should be used to configure inputs or outputs which have been described in the Workflow's Attributes (see Figure 12).
- **DEPENDENCY** type parameters can be used to configure dependencies which have been described in the Implementation's Attributes (see Figure 38).
- **CUSTOM** type parameters can be used to configure parameters which are not described elsewhere in the repository, but are required for execution. These should only be used by advanced users.

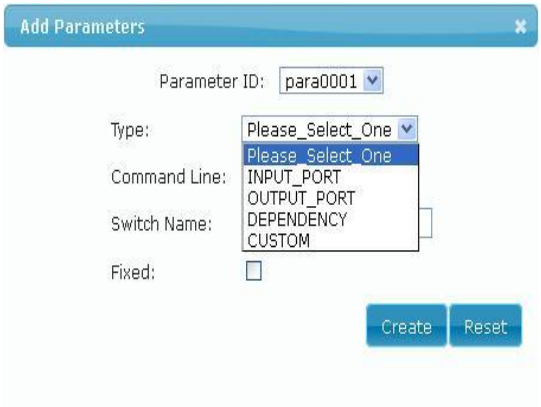


Figure 41: Select Parameter Type

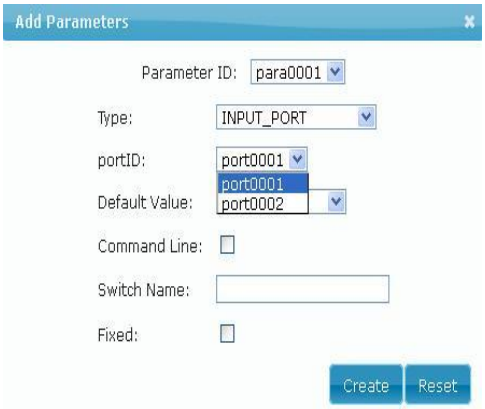


Figure 42: Select port

If the parameter is a file it should be associated with a port. First, the user should select the port type and define its ID (see Figure 42) - as it was configured in the Workflow's Attributes.

Next, the user should select which file to associate with the port as the default input file or output file (see Figure 43) in the "Default Value" area.

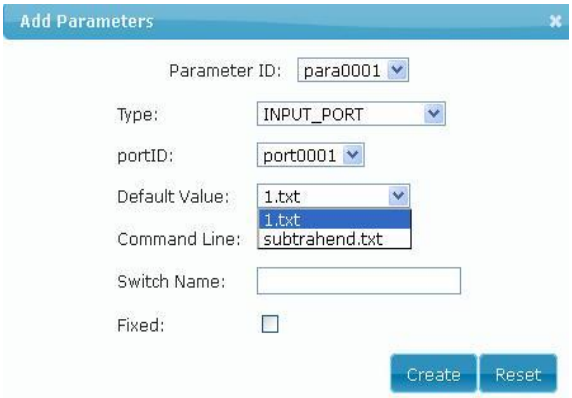


Figure 43 : Select file

Non-file type parameters, will display to the user in the dialog the ability to enter in the default value.

The user can also configure other details of this parameter:

- **Command Line**: This specifies whether this parameter is a command line parameter.
- **Switch Name**: If the parameter is a command line parameter the switch name can be specified here.
- **Fixed**: This defines whether the user can modify this parameter from the SSP.

The user interfaces for DEPENDENCY and CUSTOM typed parameters are displayed in Figure 44 and Figure 45 respectively.

Figure 44: Dependency type parameter

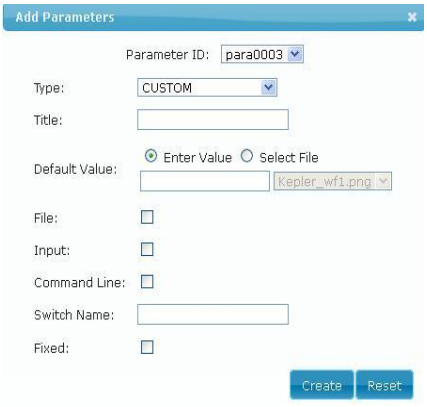


Figure 45: Custom type parameter

Once the parameter is configured, the user should click "Create" button.

The user should repeat this process to describe all parameters.

9.4. TOGGLING IMPLEMENTATION TO SUBMITTABLE

Before proceeding it is important at this stage to understand the submittable states and notations.

The icon in an Implementation's page, indicates that this implementtaion is configured for a non-submittable workflow engine.

The icon in an Implementation's page (see Figure 38), indicates that this implementtaion is configured for a submittable workflow engine – but at current is not submittable.

The icon in an Implementation's page, indicates that this implementation has been succesfully toggled to submittable

Once the SEN has been fully configured, where the workflow engine is submittable, the action tab displays the available actions (see Figure 46):

- **Expand/Collapse**: can be used to either expand or collapse nested attributes.
- **Reload**: reloads the last saved table of Implementation Attributes for this Implementation.
- **Save**: saves the Implementation Attributes.
- **Toggle Submittable**: this control can either toggle a non-submittable implementation to being submittable, or a submittable implementation to being non-submittable.

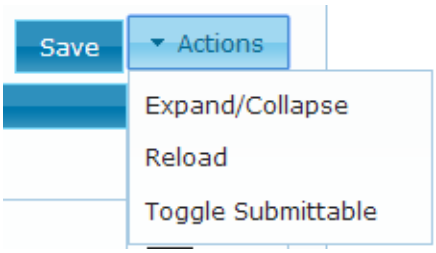


Figure 46: Actions for toggling submitability of execution

Not all of the actions in Figure 46 will be available, only the ones which are appropriate, given the submitability of the workflow engine.

Once toggled to submittable, the icon wil be replaced by .

At this stage, the concrete workflow can be selected and executed from the SSP.

If the toggling fails, an error message will appear in the browse, advising of other conditions that have not been met. Please follow the instructions provided, and reattempt the toggling to submittable.

9.5. MODIFY SUBMITABLE EXECUTION NODE

An implementation's SEN configuration (see a sample execution illustrated in Figure 47) can be modified using the following operations:

- To modify the value an attribute of an: click **Edit** next to the attribute.
- To add new parameters to the execution: click **Add** next to the attribute **parameters**.

After you performed your changes click **Save** next to **Actions** at the top of the page. The revised SEN will take effect immediately.

It is advised however, to duplicate the implementation, and apply the modifications to the duplicate implementation.

▾Submission Execution Node		Remove
maxParallelism	100	Edit
maxWallTime	1000	Edit
▾parameters		Add
▾para0001		Edit Remove
cmdLine	true	
defaultValue	string1	
file	true	
fixed	false	
input	true	
portId	port0001	
switchName	-inputfile string1	
title	string1	
type	INPUT_PORT	
▾para0002		Edit Remove
cmdLine	true	
defaultValue	string2	
file	true	
fixed	false	
input	true	
portId	port0002	
switchName	-inputfile string2	
title	string2	
type	INPUT_PORT	
▾para0003		Edit Remove
cmdLine	false	
defaultValue	concatanated	
file	true	
fixed	false	
input	false	
portId	port0003	
switchName		
title	concatanated	
type	OUTPUT_PORT	

Figure 47: Sample Implementation Execution

10. WORKFLOW AND IMPLEMENTATION ACCESS POLICY AND PUBLICATION

10.1 ACCESS RIGHTS OF WORKFLOWS - GROUP

Members of a group can be granted view, download and/or modify rights to a specific workflows which is associated with said group.

10.2 ACCESS RIGHT OF WORKFLOW – OTHERS

All other e-scientists who have logged into the repository can be granted view or download rights to a workflow.

10.3 PUBLIC WORKFLOWS

Once a workflow is marked as being public, this workflow will be viewable and downloadable by any e-scientist irrespective of whether they have logged in or not. Workflows should only be marked as public once they are in a fitting state for publication.

10.4 PUBLIC IMPLEMENTATIONS OF PUBLIC WORKFLOWS

Once an implementation is marked as being public, this implementation will be viewable and downloadable by any e-scientist irrespective of whether they have logged in or not.
Implementations should only be marked as public once they are in a fitting state for publication.
Implementations can only be marked public once the parent workflow has been marked public.
Only public Implementations may be toggled to submittable (see Section 9)

10.5 PRIVATE IMPLEMENTATIONS OF PUBLIC WORKFLOWS

These would typically be implementations of a workflow which are currently in development, or which the workflow owner wishes to remain private.

11. USER MANAGEMENT

Only administrators may manage users.

11.1 BROWSE USERS

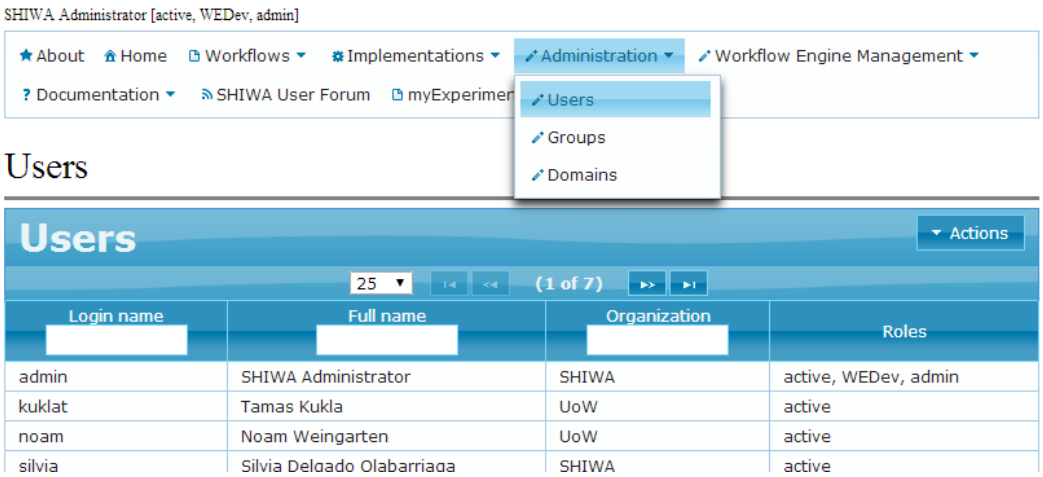


Figure 51: Browse users

Users can be listed, by selecting *Users* from the *Administration* tool.

11.2 CREATE USERS

The *Action* button from Figure 51 can be used to create new users.

11.3 MODIFY USERS

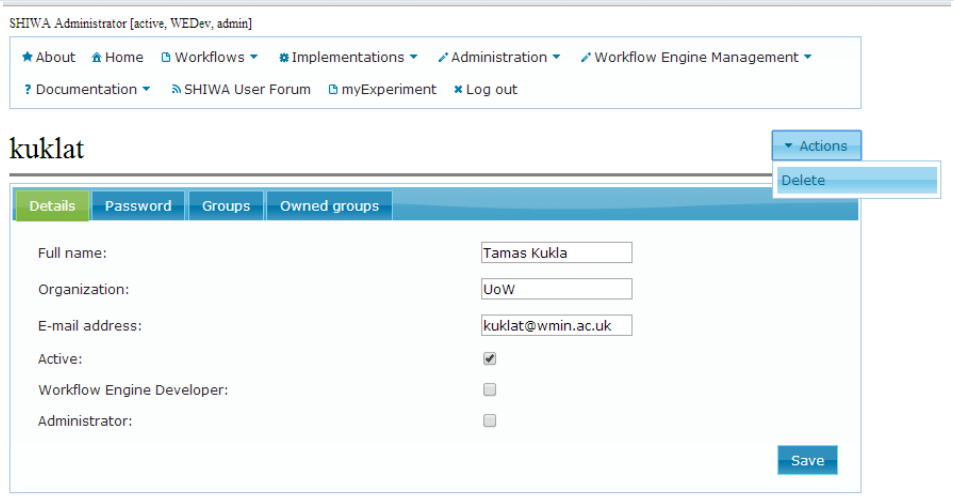


Figure 52: Modify user – details

This interface can be used to modify the users details, and the roles of the user;
Active is the default role, which denotes a workflow developer.
Administrator is the highest role.
The *Password* tab can be used to change the user's password.
The *Groups* tab can be used to list groups of which the selected user is a member.
The *Owned* Groups tab can be used to list the groups the user owns.

11.4 DELETE USERS

The *Actions* button from Figure 52 can be used to delete the user.

12 GROUP MANAGEMENT

Only administrators can list all groups, and perform modifications on such.

12.1 BROWSE GROUPS

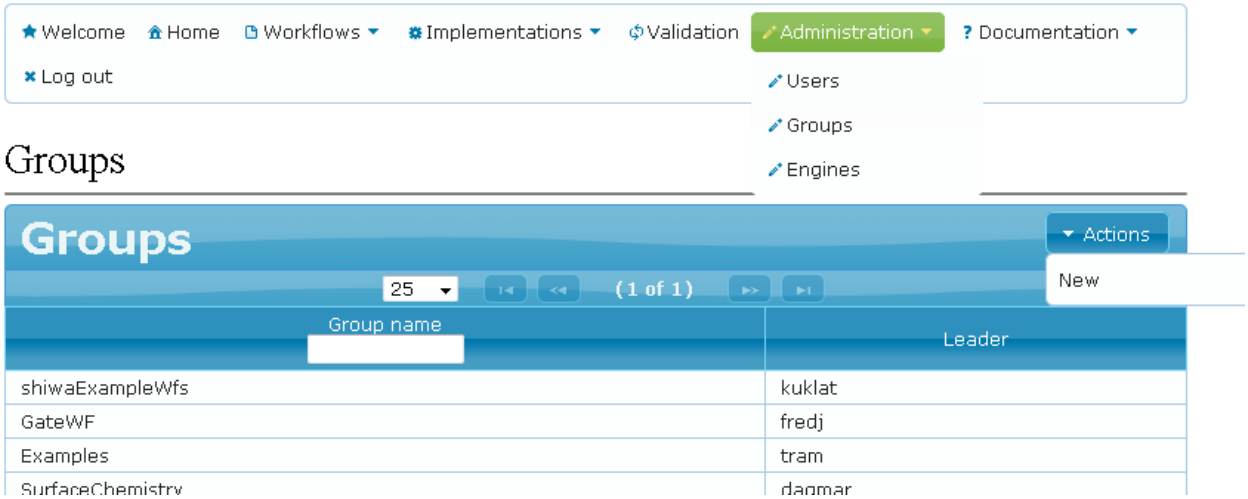


Figure 53: Browse groups

This interface can be used to browse the groups.

12.2 CREATE GROUPS

The *Actions* button from Figure 53 can be used to create new groups.

12.3 MODIFY GROUPS

Groups selected from Figure 53 can be modified.

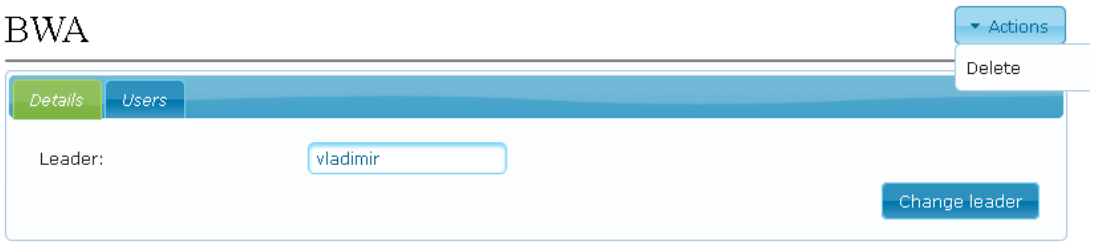


Figure 54: Modify groups

The group leader can be modified.

The *Users* tab can be used to list and modify the group members.

12.4 DELETE GROUPS

The Actions button in Figure 54 can be used to delete groups.

13 WORKFLOW ENGINE MANAGEMENT

13.1 BROWSE ENGINES

All users who have logged in are able to browse the Workflow Engines (see Figure 55) and examine their details.

Figure 55: Browse engines

Only administrators have permissions to create and manage workflow engines.

Other users will not have access to any controls which either create or modify workflow engines.

Should workflow developers require new workflow engines, they should contact administrators.

13.2 CREATE ENGINES

Engines may be created, using the Actions button in Figure 55.

The interface in Figure 55a should now be completed.

Note, you will not be able to modify the information in Figure 55a in the future.

Figure 55a: Create new workflow engine

14 DOMAIN MANAGEMENT

Only administrators may create and manage workflow engines.

Should workflow developers require new workflow engines, they should contact administrators.

Figure 57: Doman management interface

14.1 ADD DOMAIN

To add a new domain, the name of the new domain should be entered in the *Domain* box, and the respective Add button pressed.

14.2 ADD SUBDOMAIN

To add a subdomain to a domain, the name of the new or existing domain should be entered in the *Domain* box and the name of the new subdomain should be entered in the *Subdomain(s)* box. Several subdomains may be added simultaneously, by separating them with semi-colons.

14.3 DELETE DOMAIN OR SUBDOMAIN

The Delete buttons in Figure 57 can be used to delete domains or subdomains. The database will not allow used domains or subdomains to be removed.

15 SERVLET INTERFACE

The SHIWA Repository Servlet Frontend allows upload and download of entities such as workflows, implementations, and data sets in a zip based bundle file format, called SHIWA Bundle, which is a mechanism for physically aggregating resources in a zip file for publishing and archiving. The SHIWA Repository Servlet Frontend was mainly designed to enable communication and workflow exchange between the SHIWA Repository and the SHIWA Desktop. Further information on the SHIWA Bundle and the SHIWA Desktop can be found on the following URL:

https://www.shiwa-workflow.eu/wiki/-/wiki/Main/SHIWA+Desktop.

The frontend consists of four servlets: ListContents, ValidateSignature, DownloadBundle, and UploadBundle.

ListContents allows to gather basic information about the Workflows, their Implementations and Configurations.

DownloadBundle allows to download a Workflow, an Implementation or a Configuration either separately or in a single bundle file.

UploadBundle allows to upload a Workflow, an Implementation or a Configuration either separately or in a single bundle file.

SignatureValidation allows to check if the signature of an Implementation meets the signature of the Workflow which it will be added to. This is to be checked before an Implementation Bundle is uploaded to the Repository.

The followings describe in detail the functionality of these servlets and provide usage examples.

LISTING KEY INFORMATION ABOUT REPOSITORY ITEMS

function:
list summaries of workflows user can read/download
servlet:
org.shiwa.repository.toolkit.servlet.ListContents
usage example:
http://uname:passwd@hostname/shiwa-repo/workflows
returns:
with a list of Workflow Summary Objects containing workflow_id, workflow_name, workflow_description, keywords if successful, error message otherwise

function:
list summaries of workflows user can modify
servlet:
org.shiwa.repository.toolkit.servlet.ListContents
usage example:
http://uname:passwd@hostname/shiwa-repo/workflows/modify
returns:
with a list of Workflow Summary Objects containing worfklow_id, workflow_name, workflow_description, keywords if successful, error message otherwise

function:
list summaries of implementations of a particular workflow
servlet:
org.shiwa.repository.toolkit.servlet.ListContents
usage example:
http://uname:passwd@hostname/shiwa-repo/workflows/1002/imps
returns:

with a list of Implementation Summary Objects containing implementation_id, workflow_id, implementation_version, engine_name, engine_version, title, description, keywords, \verbDCIs+ if successful, error message otherwise

function:
list summaries of configurations of a particular workflow
servlet:
org.shiwa.repository.toolkit.servlet.ListContents
usage example:
http://uname:passwd@hostname/shiwa-repo/workflows/1002/confs
returns:
with a list of Configuration Summary Objects containing configuration_id, description if successful, error message otherwise

SIGNATURE VALIDATION

function:
check whether the given signature is valid
servlet:
org.shiwa.repository.toolkit.servlet.ValidateSignature, input file: signature file
usage example:
http://uname:passwd@hostname/shiwa-repo/validatesignature
returns:
with an accepted message if signature is valid, invalidation details otherwise Upload bundle

function:
upload a workflow/implementation/configuration - depending on the contents of the bundle file (with validation check)
servlet:
org.shiwa.repository.toolkit.servlet.UploadBundle
input file:
bundle file
usage example:
http://uname:passwd@hostname/shiwa-repo/uploadbundle
returns:
with an accepted message if signature is valid and upload is successful, error message otherwise

function:
upload a workflow/implementation/configuration - depending on the contents of the bundle file (ignore validation check)
servlet:
org.shiwa.repository.toolkit.servlet.UploadBundle
input file:
bundle file
usage example:
http://uname:passwd@hostname/shiwa-repo/uploadbundle/force
returns:
with an accepted message if upload is successful, error message otherwise

DOWNLOAD BUNDLE

function:
download a workflow
servlet:
org.shiwa.repository.toolkit.servlet.DownloadBundle
usage example:
http://uname:passwd@hostname/shiwa-repo/downloadbundle/1002
returns:
with a bundle file containing the workflow and related metadata if successful, error message otherwise

function:
download an implementation
servlet:
org.shiwa.repository.toolkit.servlet.DownloadBundle
usage example:
http://uname:passwd@hostname/shiwa-repo/downloadbundle/1002/2051
returns:

with a bundle file containing an implementation and related metadata if successful, error message otherwise

function:
download an implementation with parent workflow
servlet:
org.shiwa.repository.toolkit.servlet.DownloadBundle
usage example:
http://uname:passwd@hostname/shiwa-repo/downloadbundle/1002/2051?wf=true
returns:
with a bundle file containing an implementation, plus its parent workflow and related metadata if successful, error message otherwise

function:
download a workflow with implementations
servlet:
org.shiwa.repository.toolkit.servlet.DownloadBundle
usage example:
http://uname:passwd@hostname/shiwa-repo/downloadbundle/1002?imps=2051,2052,2053
returns:
with a bundle file containing the workflow, the requested set of implementations and related metadata if successful, error message otherwise

function:
download a workflow with configurations
servlet:
org.shiwa.repository.toolkit.servlet.DownloadBundle
usage example:
http://uname:passwd@hostname/shiwa-repo/downloadbundle/1002?confs=1,2,3
returns:
with a bundle file containing the workflow, the requested set of configurations and related metadata if successful, error message otherwise

function:
download a workflow with implementations and configurations
servlet:
org.shiwa.repository.toolkit.servlet.DownloadBundle
usage example:
http://uname:passwd@hostname/shiwa-repo/downloadbundle/1002?imps=2051,2052,2053&confs=1,2,3
returns:
with a bundle file containing the workflow, the requested set of implementations, configurations and related metadata if successful, error message otherwise

16 LIMITATIONS

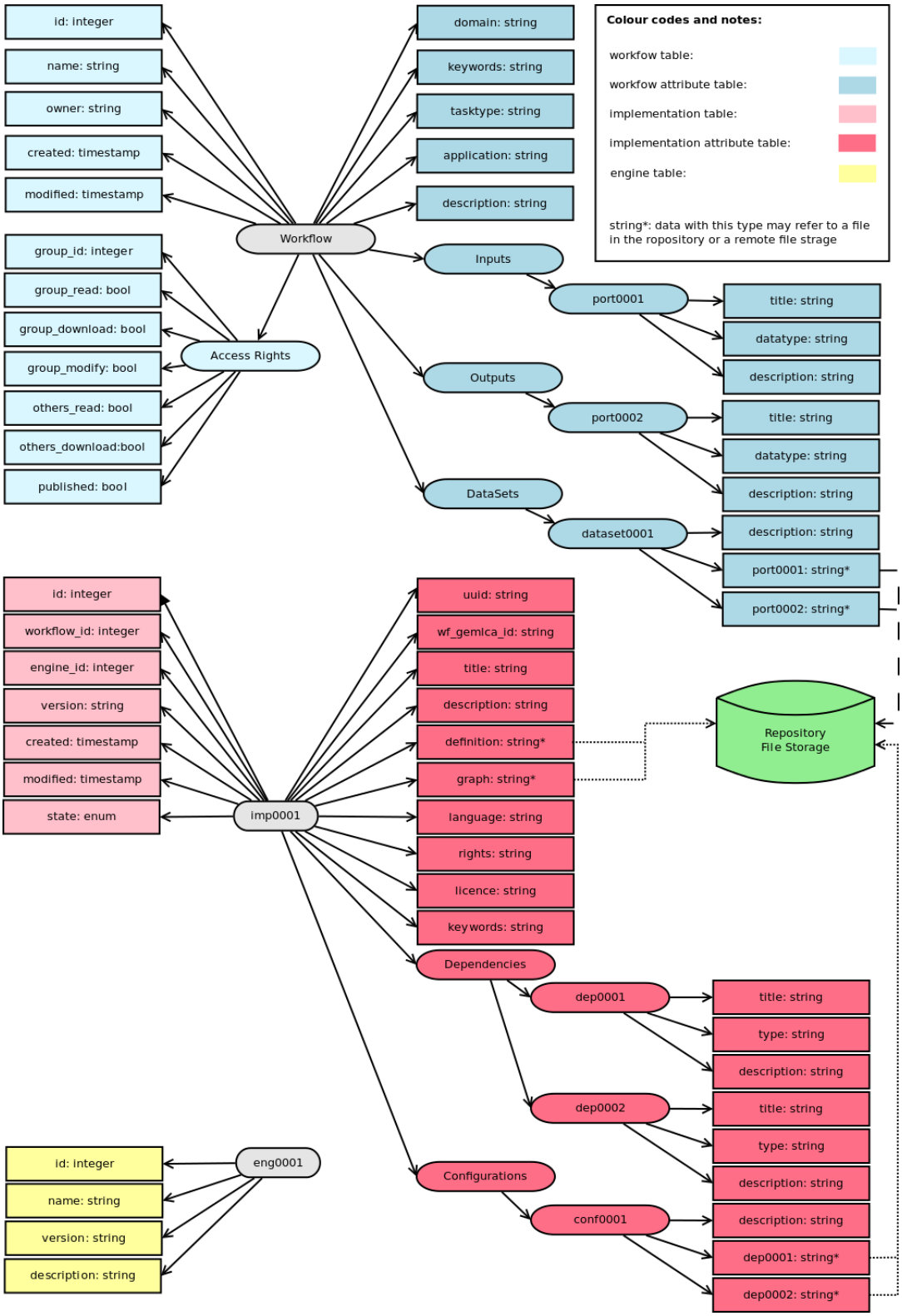
- It is not recommended to open the repository in multiple browser tabs.
- Concurrent editing is currently not supported

Workflow metadata			Example value	Description	Table	Type	Mapping to SHIWA Desktop
id			1001	workflow identifier	workflow	int	
name			Factorial	workflow name	workflow	string	
owner_id			1008	workflow owner id	workflow	int	
group_id			exampleGroup	user group for defining access rights	workflow	int	
group_read			TRUE	whether group members can see wf/impl. data	workflow	bool	
group_download			TRUE	whether group members can download wf/impl. files	workflow	bool	
group_modify			TRUE	whether group members can modify wf/impl. data and upload files	workflow	bool	
others_read			TRUE	whether registered users can see wf/impl. data	workflow	bool	
others_download			TRUE	whether registered users can download wf/impl. files	workflow	bool	
published			TRUE	whether unregistered users can see wf/impl. data and download files	workflow	bool	
created			6/5/2011 13:12	workflow creation time	workflow	timestamp	
modified			6/7/2011 16:59	time of last modification	workflow	timestamp	
application			GATE	name of the application which the wf is part of	workflow_attr.	string	workflow->shiwa:application
description			This workflow …	workflow description	workflow_attr.	string	
domain			Mathematics	scientific domain	workflow_attr.	string	workflow->shiwa:domain
keywords			factorial, integer	workflow keywords	workflow_attr.	string	
tasktype			demo	type of task the workflow represents	workflow_attr.	string	workflow->shiwa:tasktype
inputs	\			list of workflow inputs			
	port0001	\		first input port			shiwa:inport
		title	PositiveInteger	name of the port	workflow_attr.	string	shiwa:inport->dc:title
		datatype	file	data type of port	workflow_attr.	string	shiwa:inport->rdf:datatype
		description	this file contains an integer	port description	workflow_attr.	string	shiwa:inport->dc:description
outputs	\			list of output ports			
	file0002	\		first output port			shiwa-outport
		title	Factorial	name of the port	workflow_attr.	string	shiwa:outport->dc:title
		datatype	file	data type of port	workflow_attr.	string	shiwa:outport->rdf:datatype
		description	this file contains the factorial of the input integer	port description	workflow_attr.	string	shiwa:outport->dc:description
datasets	\			List of input/output configurations			
	dataset0001	\		First configuration			
		description	This dataset...	An examle dataset	workflow_attr.	string	shiwa:dataset->de:description
		port0001	input.dat	example value for port0001	workflow_attr.	string	shiwa:portref->rdf:value
		port0002	output.dat	example value for port0002	workflow_attr.	string	shiwa:portref->rdf:value

Annex Table 1: Workflow metadata attributes

Implementation metadata			Example value	Description	Table	Type	Mapping to SHIWA Desktop
id			1002	implementation identifier	implementation	int	
workflow_id			1001	identifier of the abstract workflow that the impl. implements	implementation	int	
engine_id			1005	workflow engine identifier	implementation	int	
version			1.01	implementation version	implementation	string	
created			6/15/2011 4:12	workflow creation time	implementation	timestamp	workflow->dcterms:created
modified			6/21/2011 11:24	time of last modification	implementation	timestamp	workflow->dcterms:modified
state			VALIDATED	implementation status	implementation	enum	
uuid			1234-1234-1234	uuid of implementation	imp_attr.	string	workflow->dc:identification
title			FetchImagesTaverna	title of the implementation	imp_attr.	string	workflow->dc:title
description			this implementation is ….	implementation description	imp_attr.	string	workflow->dc:description
definition			workflow.xml	workflow descriptor file	imp_attr.	string	workflow->shiwa:definition
graph			workflow.png	workflow graph screenshot	imp_attr.	string	
language			SCUFL	language of the workflow descriptor	imp_attr.	string	workflow->shiwa:language
rights			© SHIWA	copyright information	imp_attr.	string	workflow->dc:rights
licence			Demo licence	licence information	imp_attr.	string	workflow->dcterms:licence
keywords			Taverna, Images, Web Service	keywords used for searching	imp_attr.	string	
dependencies	\			List of dependencies: files needed for executing factorial.sh. It can be empty in the case of DGs.	imp_attr.		
	dep0001	\		first dependency	imp_attr.		shiwa:dependency
		title	Image Service	Title of the first dependency	imp_attr.	string	shiwa:dependency->dc:title
		description	A web service for gathering images.	Description of first dependency	imp_attr.	string	shiwa:dependency->dc:description
	dep0002	\			imp_attr.		shiwa:dependency
		title	Parameter Mapping	Title of the second dependency	imp_attr.	string	shiwa:dependency->dc:title
		description	This file maps files of the input zip to workflow ports.	Description of second dependency	imp_attr.	string	shiwa:dependency->dc:description
configurations	\			List of dependency configurations. A configuration resolves all dependencies of the executable. It can be empty if no dependencies.	imp_attr.		
	conf0001	\		first configuration	imp_attr.		
		description	This configuration...	A description of the configuration	imp_attr.	string	shiwa:configuration->de:description
		dep0001	http://moby.ucalgary.ca/...	to resolve dep0001	imp_attr.	string	shiwa:dependencyref->rdf:value
		dep0002	example_params.map	to resolve dep0002	imp_attr.	string	shiwa:dependencyref->rdf:value

Annex Table 2 Implementation metadata attributes



Annex Figure 2: SHIWA Repository metadata graph