



FOCALPOINT SERVER
COLLABORATION DEFINED

THE WORKFLOW COLLABORATION TOOL

**AN ENVIRONMENT THAT TAKES AWAY
WORKFLOW COMPLEXITY AND EMPOWERS:**

- **USERS TO CORRECTLY NAME,
SAVE & FIND FILES**
- **MANAGERS TO MANAGE**
- **COMPANIES TO CONTROL
RISK AND REPUTATION**

Version 4.2

<http://www.focalpointserver.com>

© Copyright 2011-2013 ConSol* Consulting & Solutions Software GmbH and iMedia Creative Technologies Ltd. All rights reserved worldwide.

ConSol* Consulting & Solutions Software GmbH and iMedia Creative Technologies Ltd have made every effort to make sure the information contained in this manual is accurate and reliable. However, neither ConSol* Consulting & Solutions Software GmbH nor and iMedia Creative Technologies Ltd assumes any liability for any errors that may appear in this manual or any damages any entity may claim out of the use of this manual, and this manual is provided on as "as is" basis. The ownership of any third party names mentioned in this manual, their products, and associated patents belong to them.

TABLE OF CONTENTS

1. Introduction	5
1.1 How does FocalPoint Server work?	5
1.2 Technical Requirements	6
1.3 Where to begin?	6
1.4 What is a tag?	8
1.5 What is a workspace?	9
1.6 What is a Portfolio?	10
1.7 How to add a Portfolio	11
1.9 Deployment Scenarios	12
1.9.1 Tasks of the administrator	12
1.8 What is a schema?	12
2. Administration Interface	13
2.1 Overview of the Main Configuration tabs	13
2.1.1 The users Tab	13
2.1.2 The roles Tab	14
2.1.3 The roots Tab	16
2.1.4 The templates Tab	16
2.1.5 The tags Tab	17
2.1.5.1 Hierarchies;	18
2.1.6 The workspaces Tab	20
2.1.6.1 Workspaces/common panel:	22
2.1.6.2 Workspaces/application panel:	23
2.1.6.3 Workspaces/tags panel:	24
2.1.6.4 Workspaces/naming panel:	27
2.1.6.5 Workspaces/display panel:	27
2.1.6.6 Workspaces/notification panel:	28
2.1.7 The search Tab	29
2.1.8 The server scripts Tab	31
2.1.9 The licence Tab	32
2.1.10 The import/export buttons	33
2.1.11 Other basic buttons	33
3. Disaster Recovery	34
3.1 Backing up data using mongodump tool	34
3.2 Setting up the backup server	34

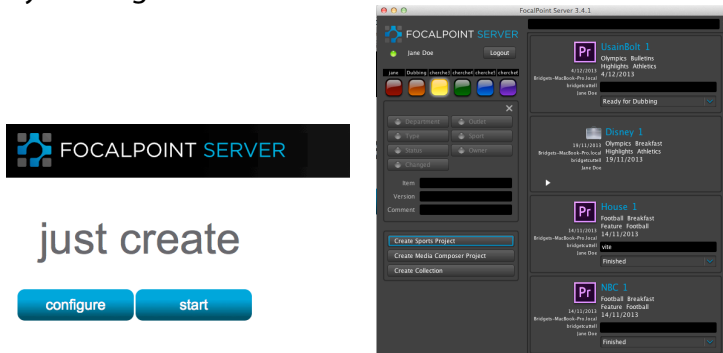
3.3 Restoring backed up data using mongorestore tool	35
<i>A-Appendix</i>	<i>36</i>

1. Introduction

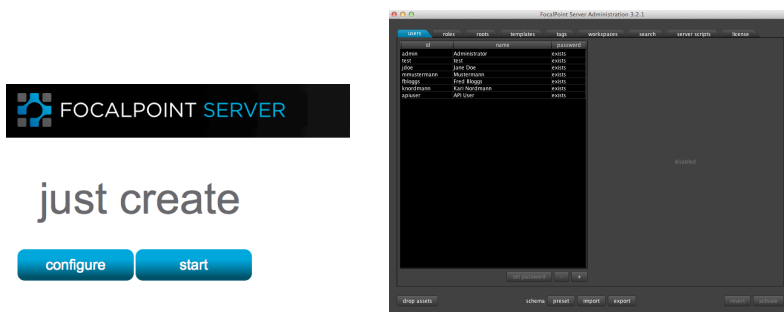
FocalPoint Server has an intuitive Client Interface designed to make an editor's life easier when it comes to creating, naming, using, finding, sharing and versioning Final Cut Pro, Avid Media Composer, Premiere and other types of graphics and VFX projects. It eases workflow problems and collaboration issues.

It consists of three elements:

- The Client Interface is the front end that runs on each client workstation. This is reached by clicking **Start**.



- The Configuration Interface which allows the administrator to customize each FocalPoint Server client installation to the workflow requirements of the site.



- The API Interface
(For third party integration e.g. in order to function with a storage vendor etc)

This guide introduces you to the Configuration Interface.

Additionally, training movies explaining the configuration of the FocalPoint Server can be found on the FocalPoint Server website (<http://www.focalpointserver.com>) in the Demo Section.

1.1 How does FocalPoint Server work?

FocalPoint Server provides the end user, typically editors and producers, with an easy to use interface for running their editing, graphics or VFX applications. It manages the many assets, projects or documents they create. The system uses **tags** set up in a **schema** at configuration time, to provide one click access to all of the projects involved in a production. On creating a project, the user only fills in very few **tags**. The **Search Lights** display creates a clear and graphically intuitive interface to then search for these **tags**.

FocalPoint Server can scale up exponentially and will grow as big as any asset collection, thanks to its mongo database.

It is a new way of thinking about *projects*, as opposed to *media*, mimicking the real life thought processes of experienced editors. FocalPoint Server enforces intelligent project naming and enables sharing projects and re-versioning projects.

FocalPoint Server Administrator interface hides the complexity from the end user and deals with it in a centralized and coherent manner. In order to successfully configure FocalPoint Server, the administrator should understand **schemas, workspaces, and tags**.

1.2 Technical Requirements

Technical requirements for FocalPoint Server Workgroup Version (Max 15 Users)

Minimum Requirements (One Server):

- x86-64, 3Ghz processor and above, quad core
- 8GB memory
- Ubuntu LTS 12.04 64Bit/Redhat EL 6.2 64Bit/CentOS 6.2 64Bit
- OSX 10.7 and above (Lion)
- Windows 2008 R2
- RAID 1+0 drive configuration for FocalPoint Server data (independent of OS drive(s))
- Java 7
- For Linux VM installations of FocalPoint Server only: Linux package haveged

Technical requirements for FocalPoint Server Enterprise Version

Minimum Requirements:

- One server (for High Availability with or without hardware load balancers, and for robust database redundancy, two servers are mandatory):
- x86-64, 3Ghz processor and above, quad core
- 16GB memory
- Ubuntu LTS 12.04 64Bit/Redhat EL 6.2 64Bit/CentOS 6.2 64Bit
- RAID 1+0 drive configuration for FocalPoint Server data (independent of OS drive(s))
- Horizontal Scaling achieved by adding pairs of servers.
- Java 7

For Linux VM installations of FocalPoint Server only: Linux package haveged

1.3 Where to begin?

Follow the instructions that come with the installation package to install the software. A discussion with the users of the system is necessary to decide how their projects should be named. Once decided, it is relatively simple to mirror the workflow in filling out the Configuration Interface. Once installed, a web browser is used to access the FocalPoint Server Configuration Interface:

Eg. <http://192.168.1.1:8080/focalpoint-server>

A startup screen will appear in the browser with two buttons: “configure” and “start.”



just create



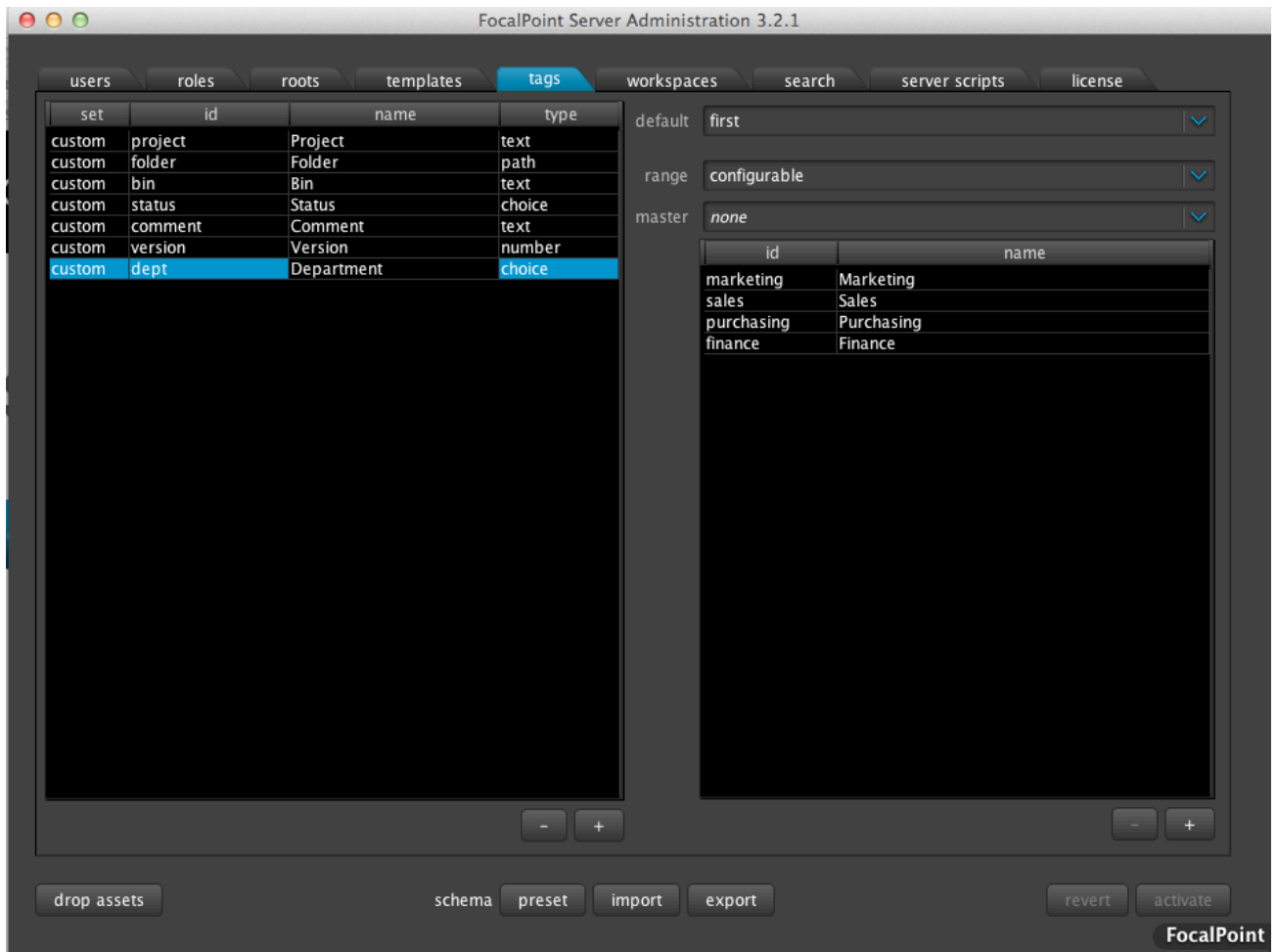
Click the **configuration** button and the FocalPoint Server Configuration application will launch. The default login and password is:

admin
admin

This password should be changed as soon as possible from a security point of view. In order to use FocalPoint Server you will need a trial or permanent licence. See [2.1.9 the licence tab](#).

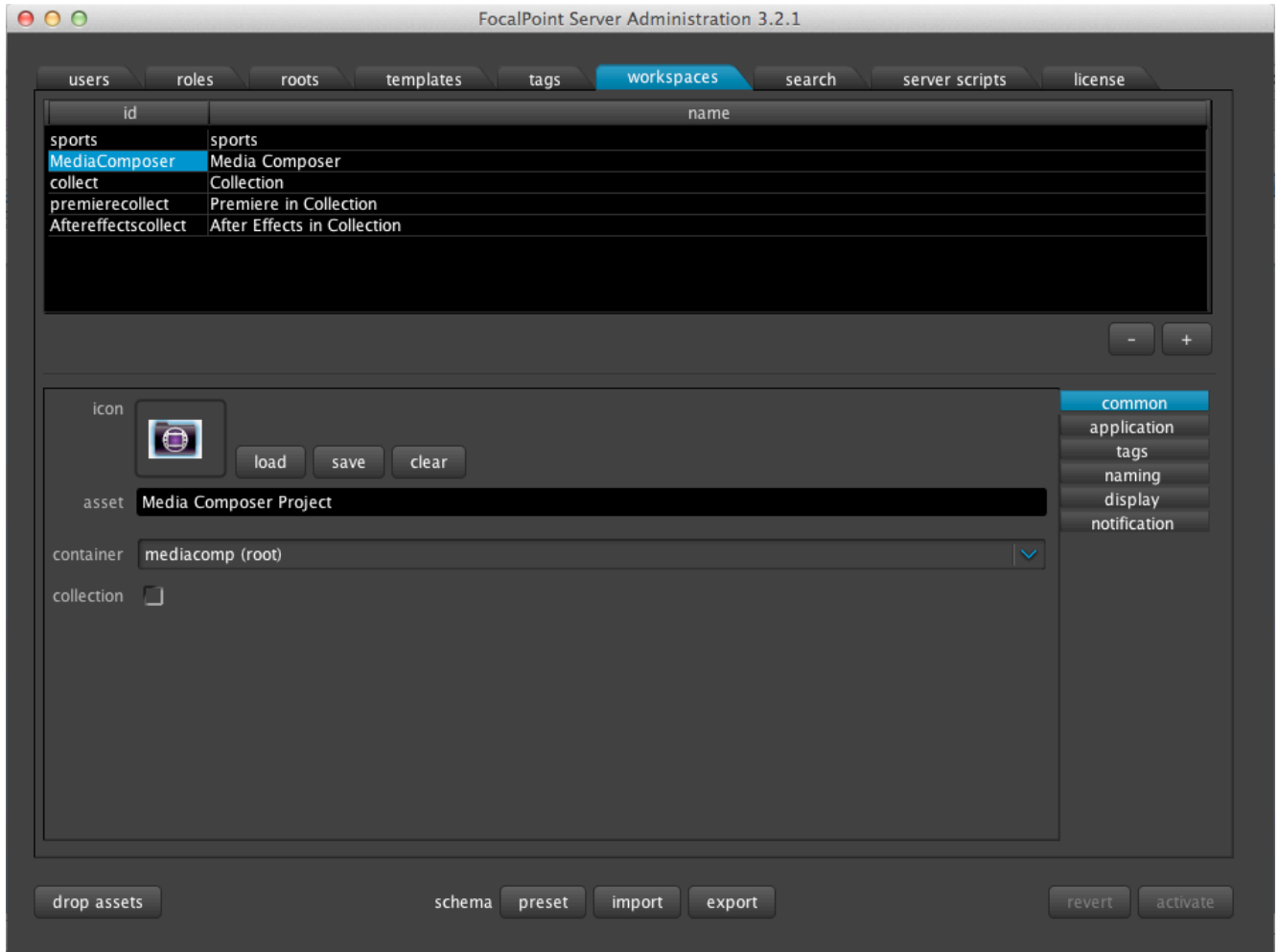
1.4 What is a tag?

Tags are the basic pieces of information about an asset. Tags are used by FocalPoint Server to describe the various attributes of an asset such as name, ownership, status etc. Tags also keep track of the assets through the production workflow process. Tags are set up by the administrator to mirror the organizational requirements at the site. Most tags are visible to the user, however there are also some internal tags that are used by the administrator only, e.g Timestamp. See the picture below for an Office suite administrative set of tags created in the Configuration Interface.



1.5 What is a workspace?

Generally, a workspace is a personalized application. That is, there is a workspace for Final Cut Pro, a workspace for Avid Media Composer etc. This workspace contains a template with preset settings, so that a customized template opens within each application. For example a Final Cut Workspace might open a Final Cut Project with the resolution, colour saturation etc already defined. A user may have multiple workspaces if they use multiple applications. Each workspace will have its own **Create** button in the User Interface, and its own set of tags. (See below for the Media Composer Workspace in the Admin Interface and then the corresponding **Create Media Composer** button in the User Interface)

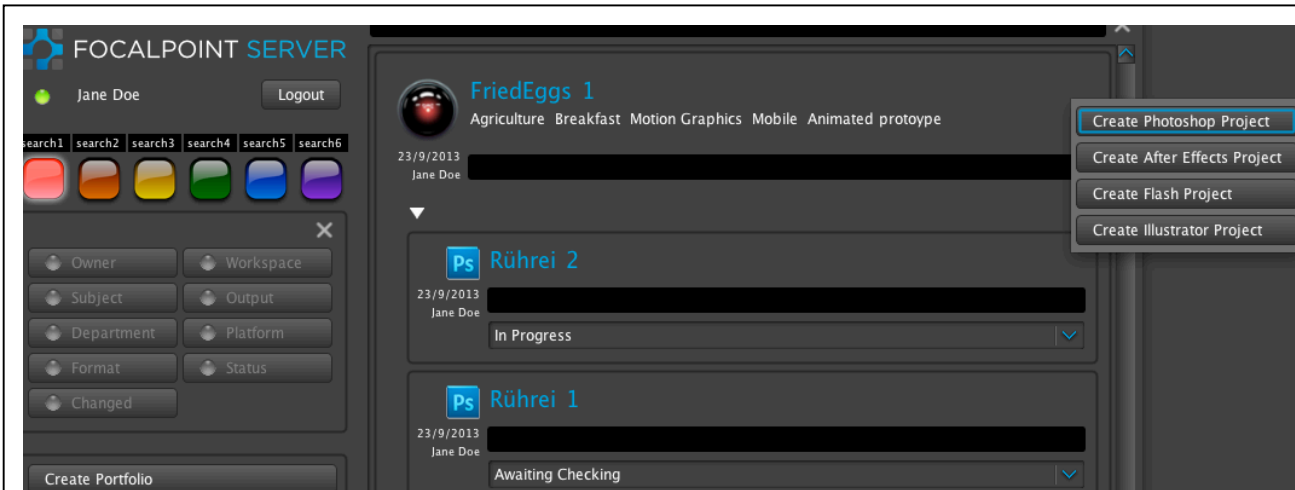




1.6 What is a Portfolio?

Additionally, there is a special workspace that is not associated with any one particular application, but contains a group of applications. Such a workspace, together with its associated applications is called a **Portfolio**. (This is typically used in Model B deployment, though can also be used in Model A in certain specific scenarios. See *1.8 Deployment Scenarios*.)

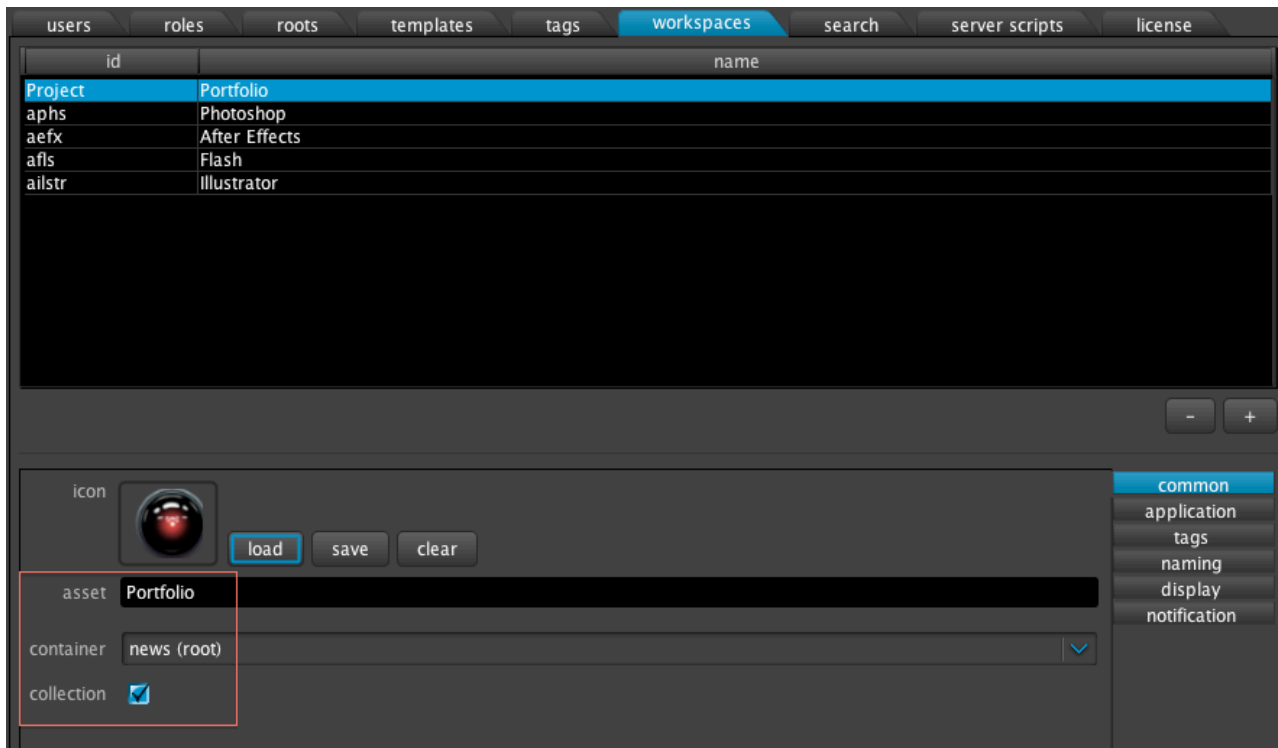
This Portfolio will have a button of its own to **Create Portfolio**, which will create the Portfolio folder. Then in order to open an application, you must choose one *within* the **Portfolio**. The image below shows an example of the Client Interface with a **Portfolio** workspace with Photoshop, After Effects, Flash, and Illustrator.



1.7 How to add a Portfolio

In the file system a Portfolio corresponds to a folder and assets within the Portfolio are various individual child project files within that folder. These applications (Photoshop, After Effects, Flash, Illustrator) will have their own naming and display mechanisms in their own workspaces in the admin interface. The Portfolio will have one root, and every workspace will have the same root, so that the entire project is saved in one place.

Within the **Workspaces** tab, in the **common** panel, the Portfolio is given an **asset** Project/Portfolio. The **container** is the root defined in the roots tab and the checkbox next to **collection** is checked.



1.8 What is a schema?

A schema consists of user roles, tags, workspaces and search configuration. It is your customization of FocalPoint Server. An organization typically has one schema that may include several workspaces and Portfolios.

1.9 Deployment Scenarios

While FocalPoint Servers affords a lot of flexibility, one of the key considerations is whether to use a model involving centralized shared storage or not.

Model A (local storage):

If the project assets are very small as in FCP7, MC or Premiere, it is recommended not to use centralized shared storage. In this case a copy of the project assets, along with all the metadata associated with the projects is always kept in the FocalPoint Server database. It is strongly recommended to use a High Availability configuration with at least two servers. RAID 10 configuration over an even number of disks is strongly recommended where the FocalPoint Server database will be written. The number of disks required for the RAID 10 configuration depends on the number of project assets.

Model B (shared central storage) :

If the project assets are large with embedded images such as Adobe Photoshop projects, it is recommended to use centralized shared storage where the projects are kept in the centralized shared server, but FocalPoint Server database keeps track of the metadata associated with the projects. Again it is strongly recommended to use a High Availability configuration with two servers and RAID 10 configuration as outlined in the preceding paragraph.

Combination of Model A and Model B:

It is also possible to design a schema with a combination of workflows: no centralized storage for project assets that are very small and centralized storage for project assets that are large.

1.9.1 Tasks of the administrator

Typical tasks of the administrator are:

- establishing **naming** conventions for new projects
- defining **workspaces** with **templates** for the applications
- defining **custom user interfaces** using **tags**
- defining the Search Lights **buttons**
- setting up Growl **notifications** of project status updates
- managing user and password **accounts** and **permissions**
- managing licence codes

It often easiest to start with the users tab and work along each tab, filling in each tab in order. A description of each tab follows.

2. Administration Interface

2.1 Overview of the Main Configuration tabs

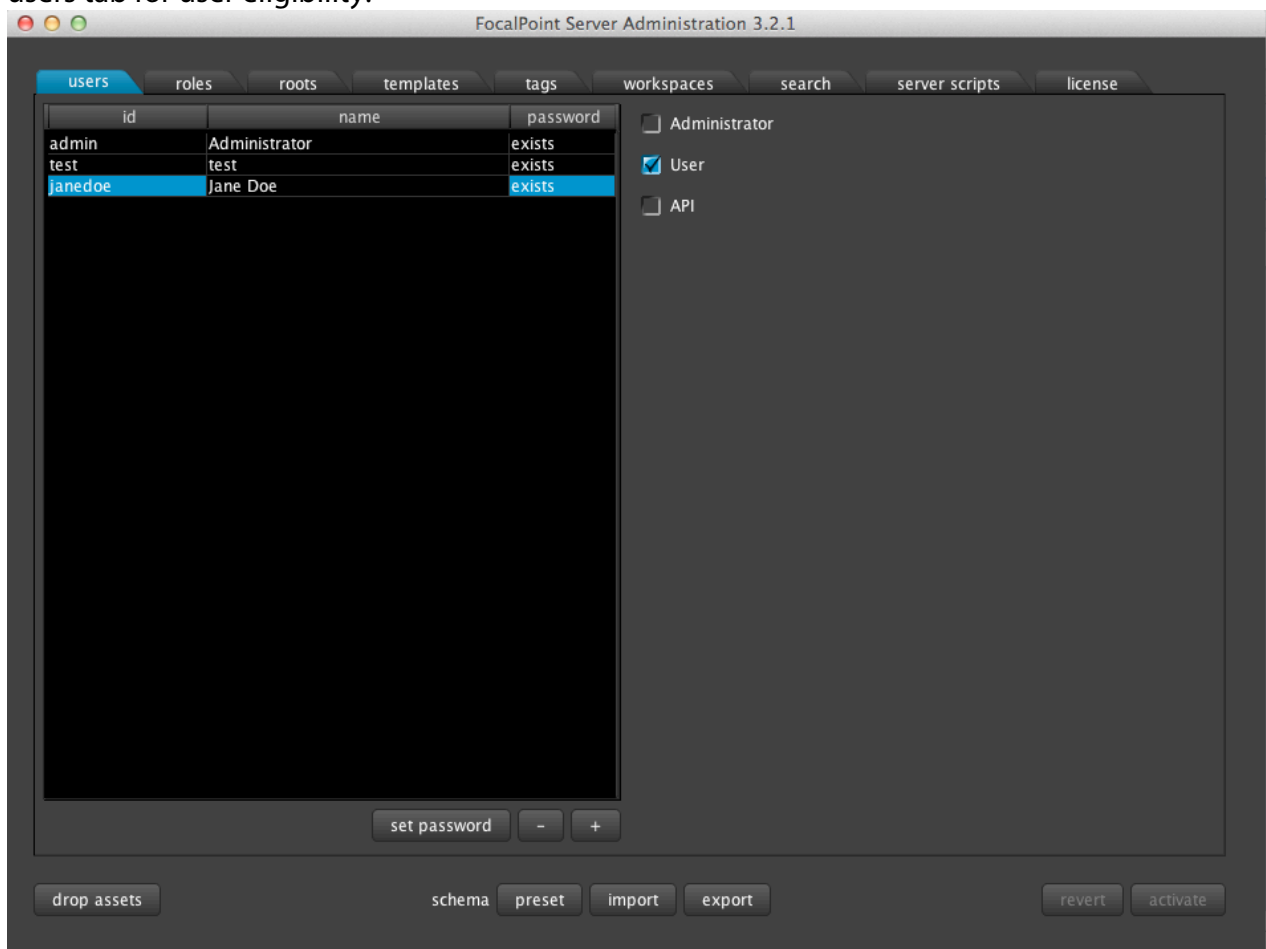
In every configuration tab Standard plus (+) and minus (-) buttons can be used to add or delete tags contained within the interface. Other standard conventions across the interface are the ability to click in a field to edit it, and to tab between fields. Drag and drop functionality is also provided within a panel and from one panel to another.

In this chapter, the functionality is described, and in Chapter 3 examples will be shown.

2.1.1 The users Tab

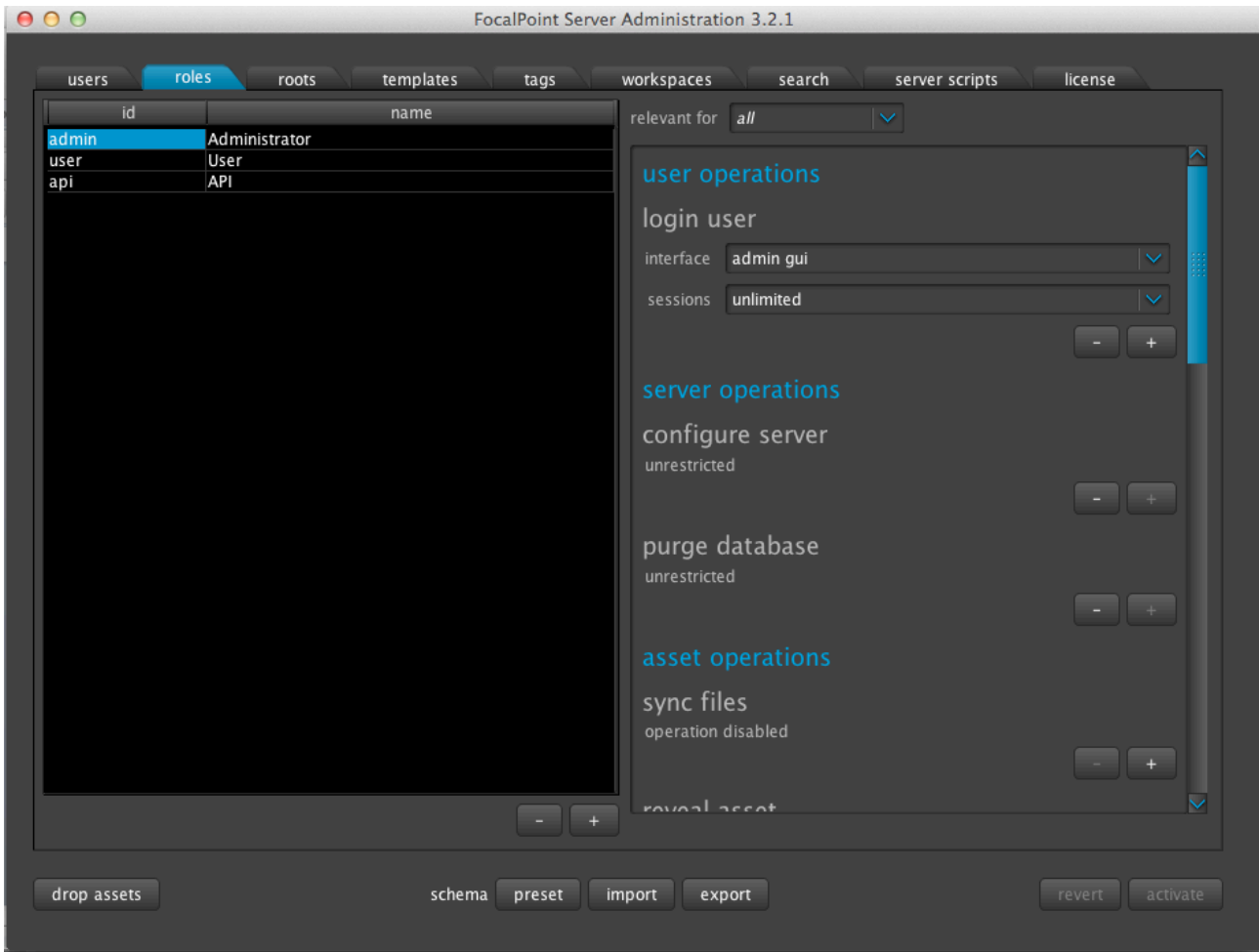
The users tab provides a mechanism to add users and set passwords. The **set password** button is necessary after creating a user. It uses the standard plus (+) and minus (-) buttons to add or delete users.

In the right-hand side of the list of users, there are three default role types with check boxes, namely **Administrator**, **User** and **API**. A user can be enrolled in one or more of these roles by clicking on the check box next to the role. The permissions for these roles are defined in the roles tab (2.1.2) where additional custom roles can be defined. These will then appear under the users tab for user eligibility.



2.1.2 The roles Tab

The roles tab assigns permissions to each role, as well as adding and removing roles. These roles are for the User Interface, but also the Administration Interface and the API Interface. The API Interface is for external applications to communicate with FocalPoint Server. Additional role types can be created under the roles tab and groups of users can be assigned to these role types and permissions. A user may have multiple roles.



By default, three role types are provided and the following permissions are set:

Admin role, relevant for admin interface:

Administrators who have been assigned this role can (enabled by default)

- (i) login to the admin GUI interface
- (ii) configure FocalPoint Server
- (iii) purge the FocalPoint Server database of all entries

Users role, relevant for user interface:

Users who have been assigned this role can (enabled by default);

User operations:

-login to the Client Interface.

Asset operations (by default applies to all workspaces and assets owned by both the user and others):

- (i) copy (sync) projects and associated metadata from their client machines to the FocalPoint Server database in the case of Model A (see 1.1.4), and copy metadata associated with projects in the case of Model B (see 1.1.4)
- (ii) reveal asset, i.e., show path leading to where asset is kept on storage
- (iii) view an asset, i.e., display an asset for viewing within an editing application and this is write protected
- (iv) edit asset, i.e, open the asset for editing
- (v) create a new asset
 - (i) Portfolio
 - (vi) procreate asset, i.e, in the case a workspace is a Portfolio (1.1.2), create a child asset to the
 - (vii) reversion asset, i.e, create a copy of an existing asset with a new version number
- (viii) delete asset, i.e, delete an asset from the client, data still lives in the FocalPoint Server database (project and metadata in Model A (1.1.4), and metadata in Model B (1.1.4))



These are the permissions buttons for these capabilities on each project in the Client Interface.
Tag operations (by default applies to all workspaces, assets owned by both the user and others, and all tags):

- (i) see tag, ie, show the value of a tag
- (ii) edit tag, i.e, allow to edit the value of a tag

API role, relevant for API Interface:

Users who have been assigned this role, through external applications can (enabled by default);

User operations:

- login to the FocalPoint Server API interface.

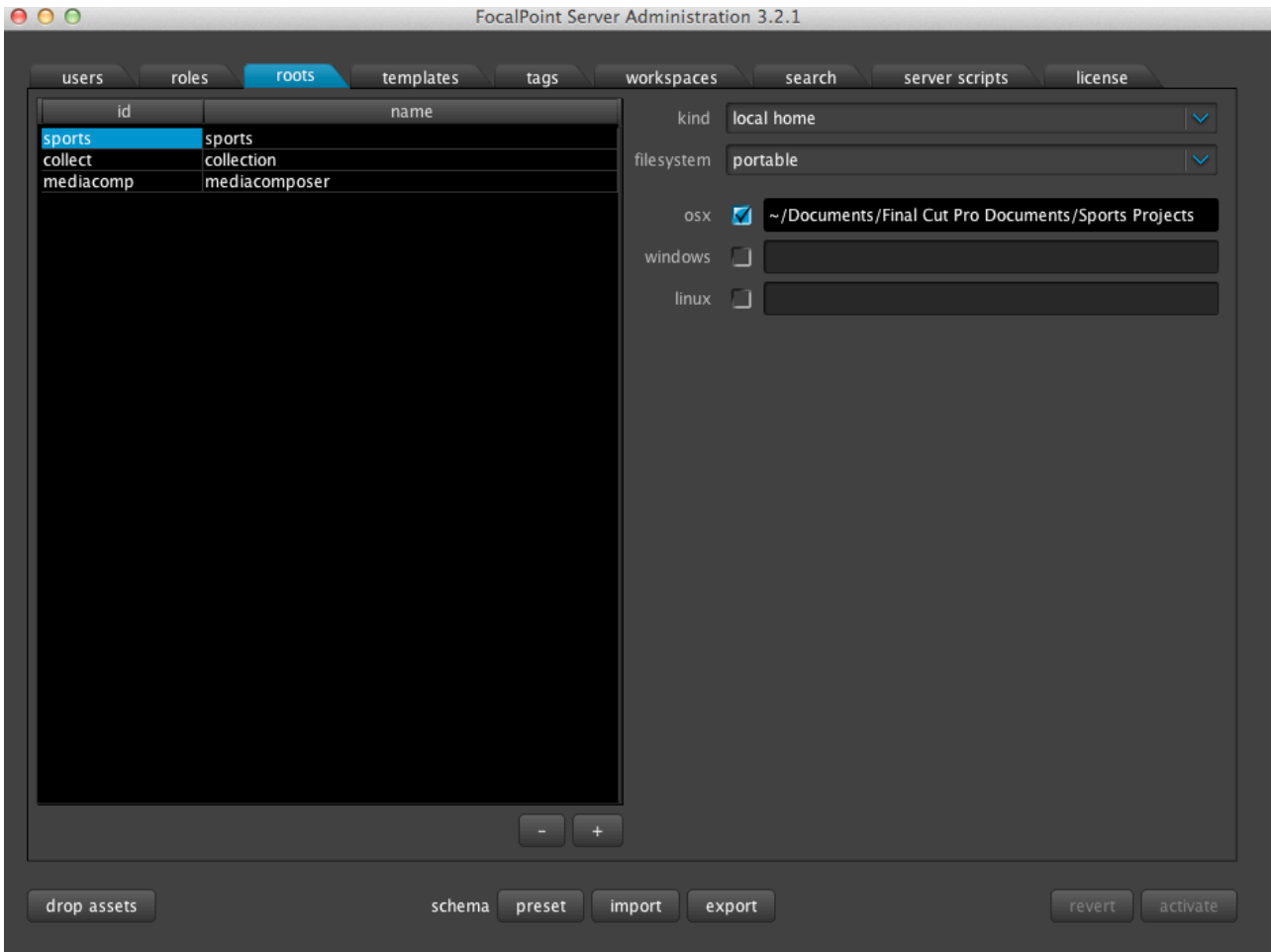
Asset operations (by default applies to all workspaces and when applicable assets owned by both the users and others):

- (i) create a new asset
- (ii) delete asset, i.e, delete an asset from the client, data still lives in the FocalPoint Server database (project and metadata in Model A (1.1.4), and metadata in Model B (1.1.4))
- (iii) purge asset, i.e, delete an asset completely from the database
- (iv) restore asset, i.e, restores a deleted asset

Tag operations (by default applies to all workspaces, assets owned by both the user and others, and all tags): edit tag, i.e, allow to edit the value of a tag

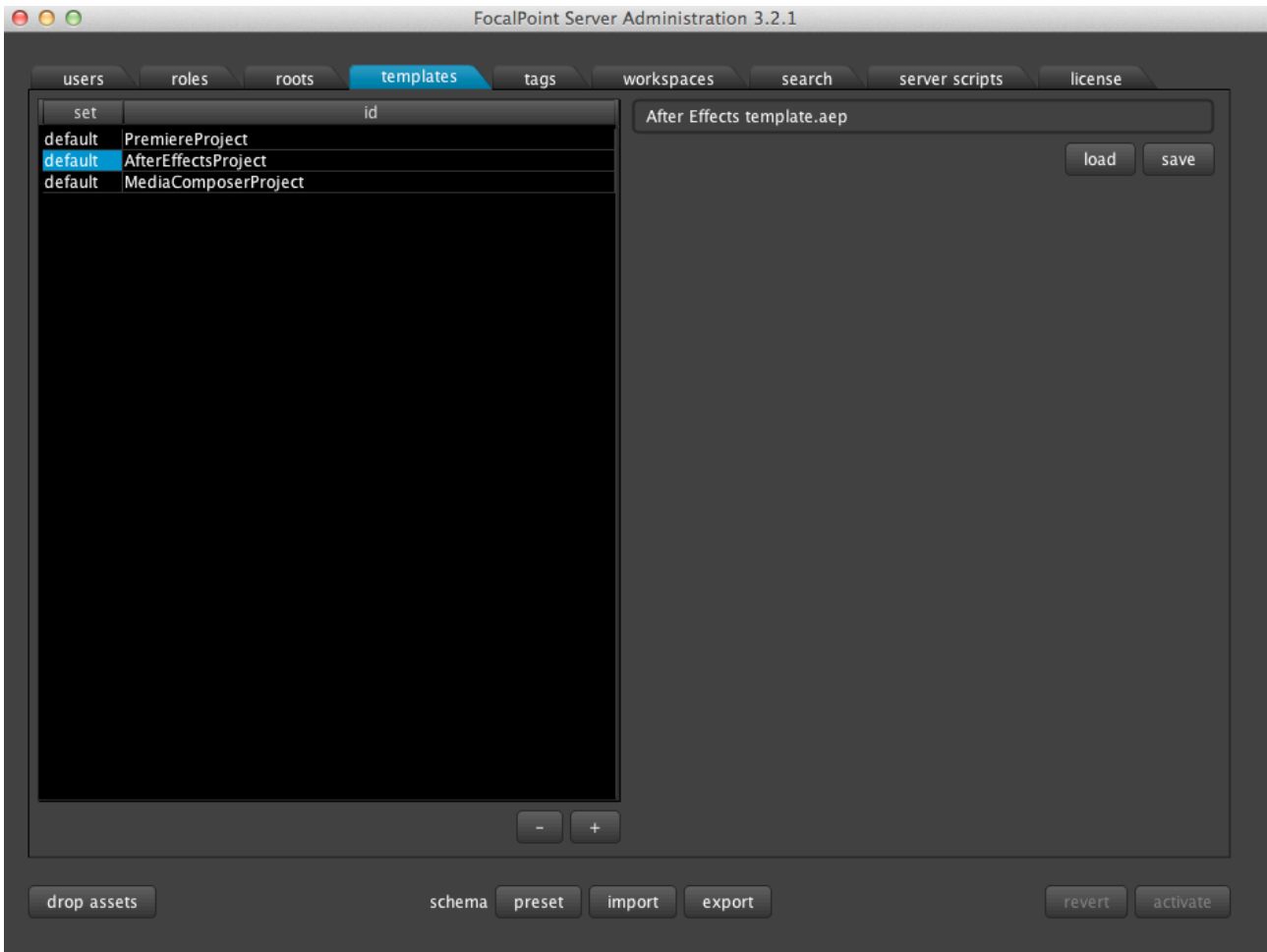
2.1.3 The roots Tab

Each Workspace needs a root. A root is defined by the root id and name, and assigns the project paths for the contents of each project or Portfolio. This can be on OSX, windows or linux, on shared or local storage – simply in My Documents. The type of file system is taken into account, where restrictions on the number of characters and the types of characters may apply.

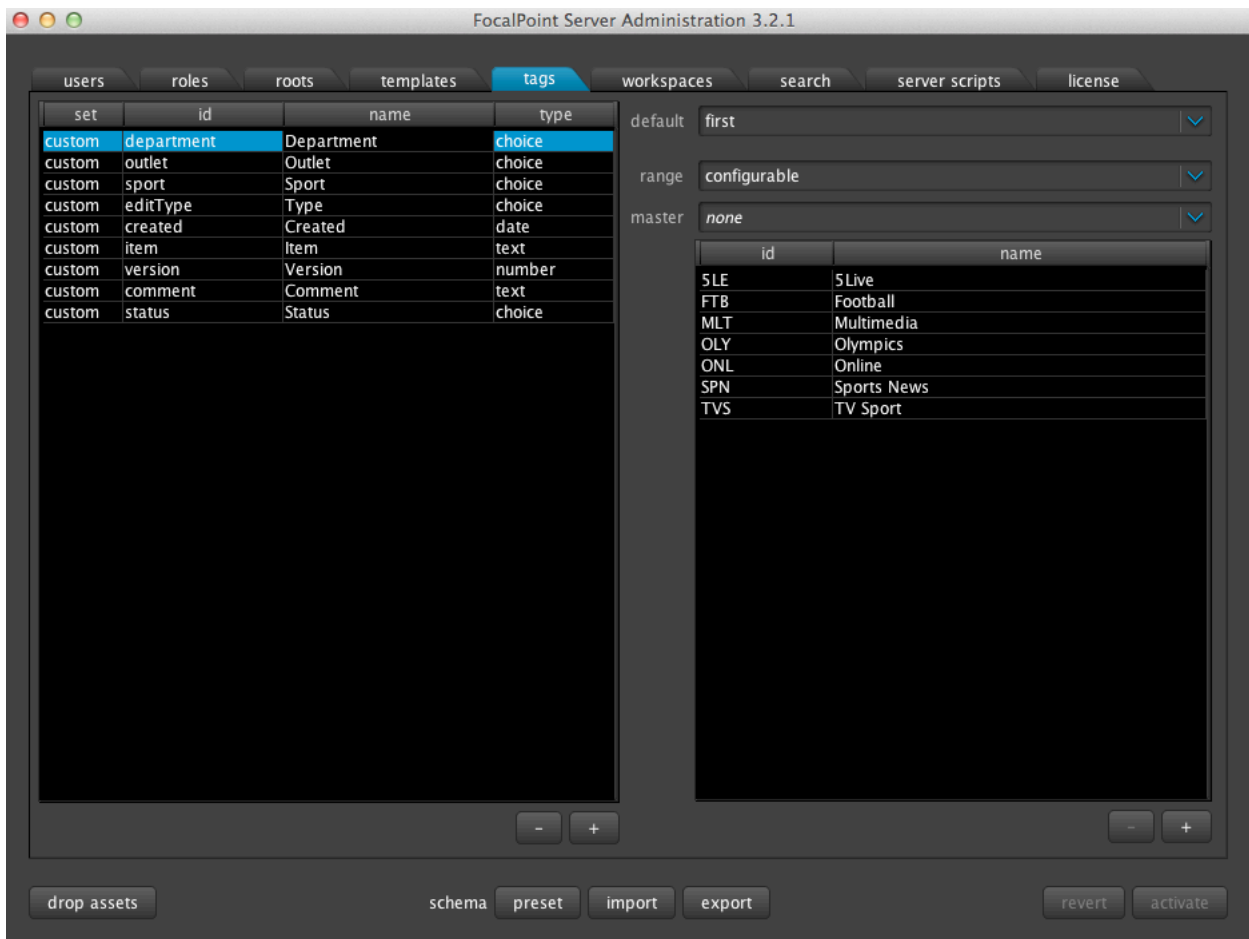


2.1.4 The templates Tab

Here one can load a predefined template file, for each workspace. So when an application is opened, a customized template within that application automatically opens. For example, Final Cut project files with specific settings, timelines, and sequences. It is possible to have multiple templates per application, and define an id and name for each of them. (For example, this could open a different template for each Project, despite using the same application)



2.1.5 The tags Tab



The tags tab in the Configuration GUI allows the administrator to define sets of tags. The tags on the left are a pool of tags that you will use throughout the Interface. The values for these tags are then created on the right. Tags can have different types the attribute they are describing. They can be of type;

text (textual value)

number (numeric value)

date (calendar date)

instant (timestamp with millisecond precision)

path (subsection of a file path)

multi (allows the user to enter multiple entries in the Client Interface)

choice (indicating one out of a fixed set of possible choices)

Once this type is chosen, it can be further defined with default variants in the **range** field. For example, for the type **choice** the **range** can be;

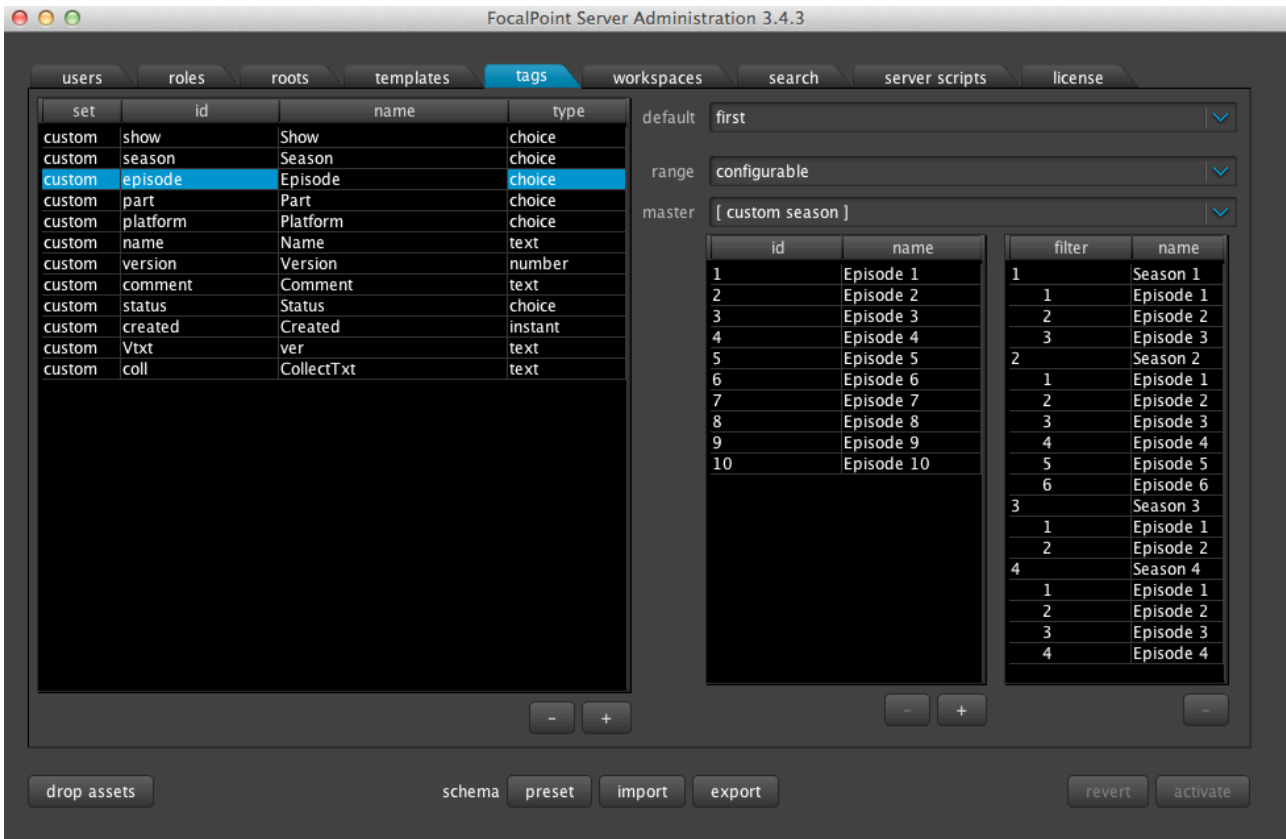
configurable (an administrator-configured list of possible tags)

roots (list of all roots)

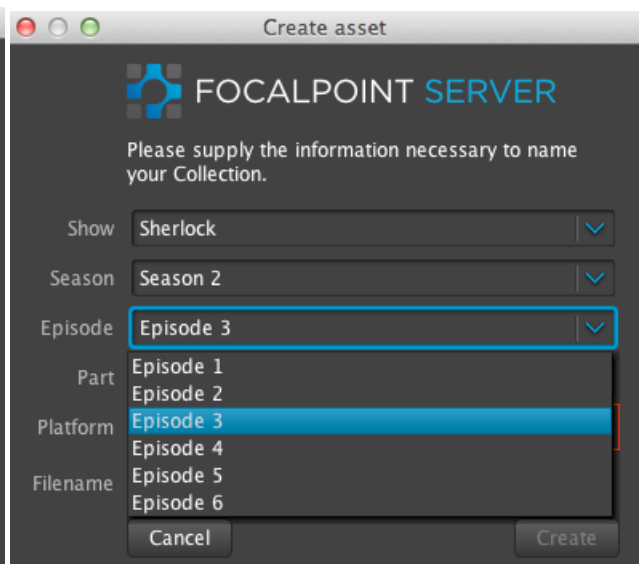
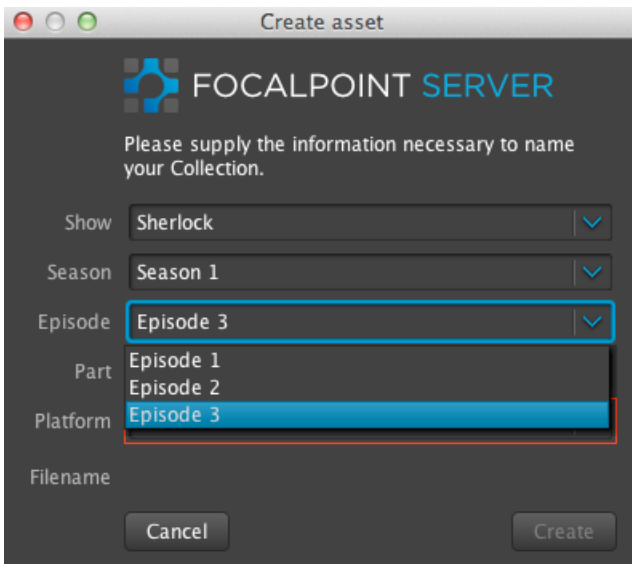
workspaces (list of all workspaces)

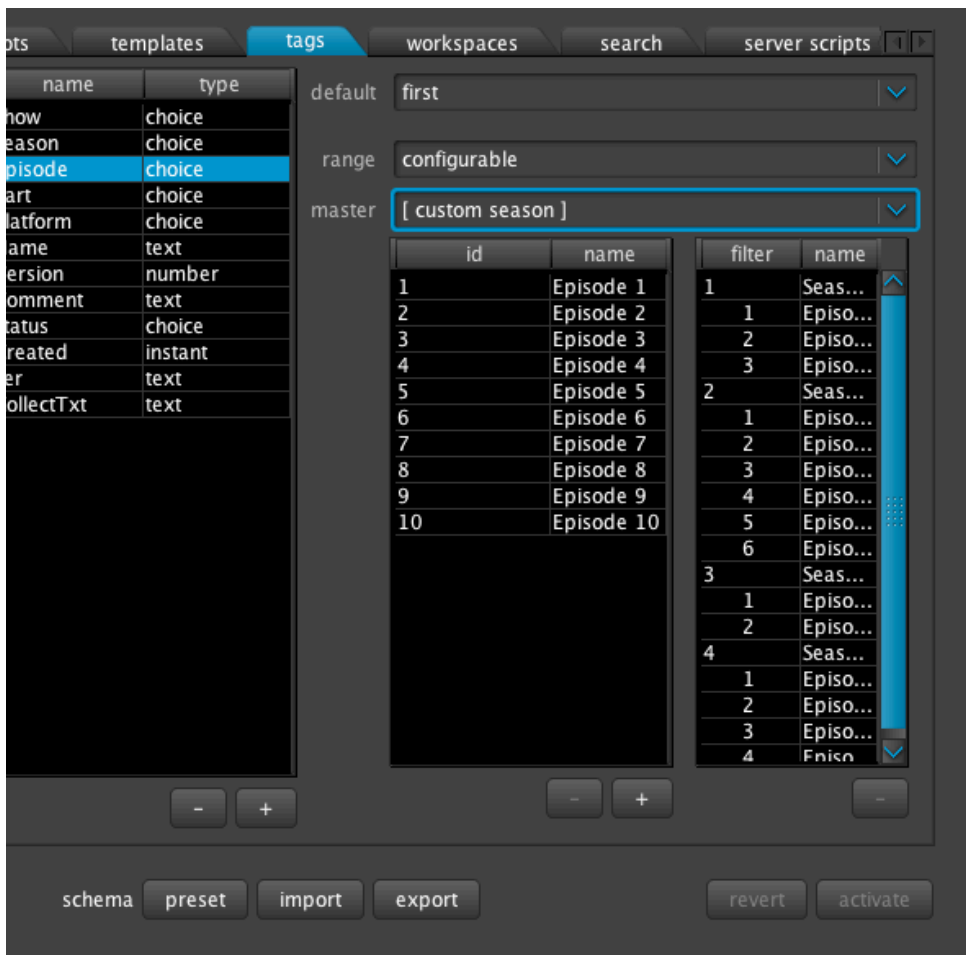
users (list of all users)

2.1.5.1 Hierarchies;



In addition, as you can see above, the administrator can create a *dependent hierarchies of tags*. So that once a **master** tag is chosen, only a selection of dependant tags can appear. For example, once the Season is selected as **Season 1**, then only the relevant **Episodes** will be selectable.





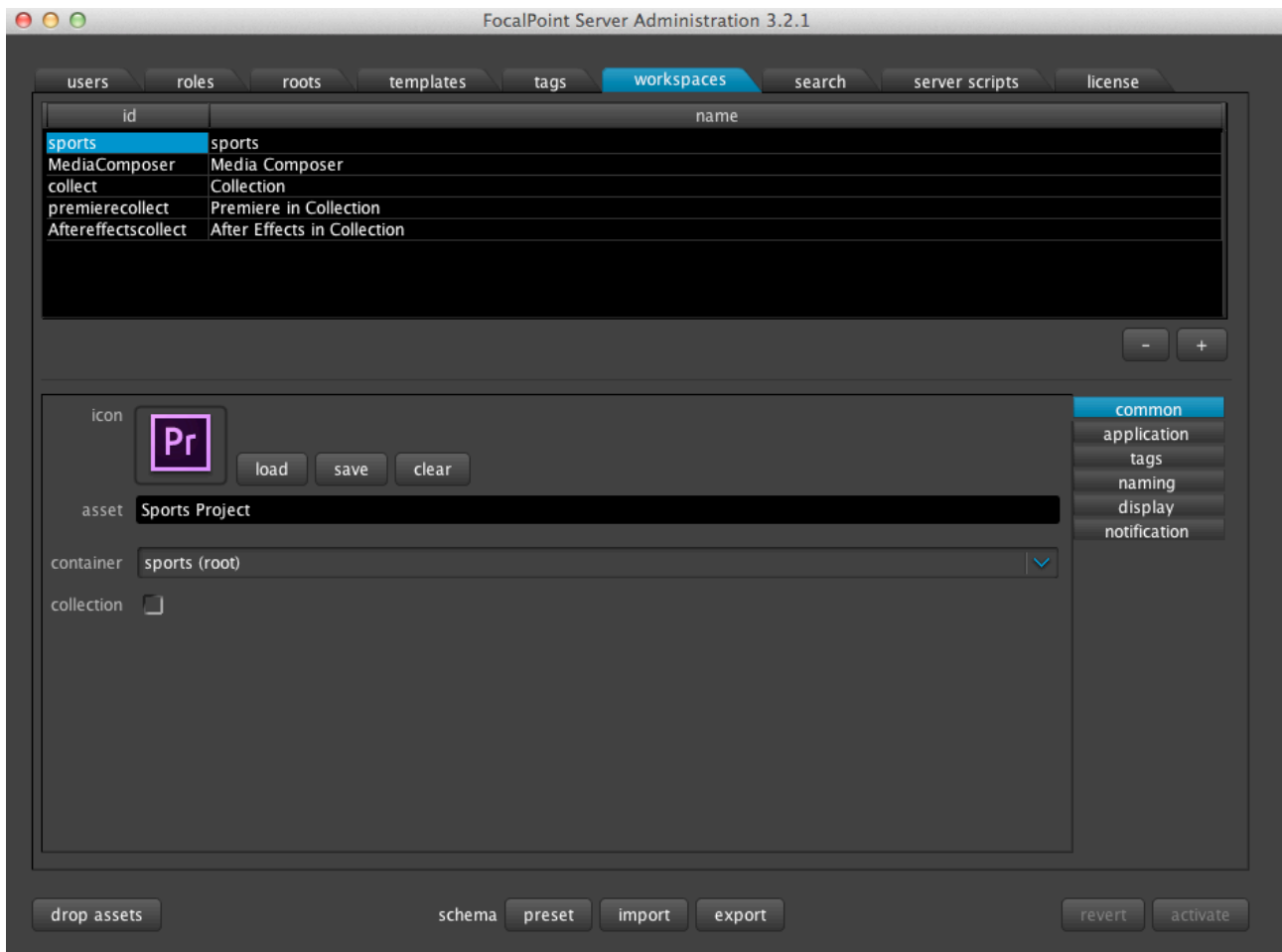
Default : The default tag to appear.

Range : In order to create dependencies, the range must be configurable

Master : Here you define which tag you are dependent on. We have selected 'Episode' in the left hand panel. The Episode selection depends on the Season, so the 'master' tag is Season.

You can achieve this by dragging and dropping the relevant number of **Episodes** under each **Season** in the panels on the right hand side.

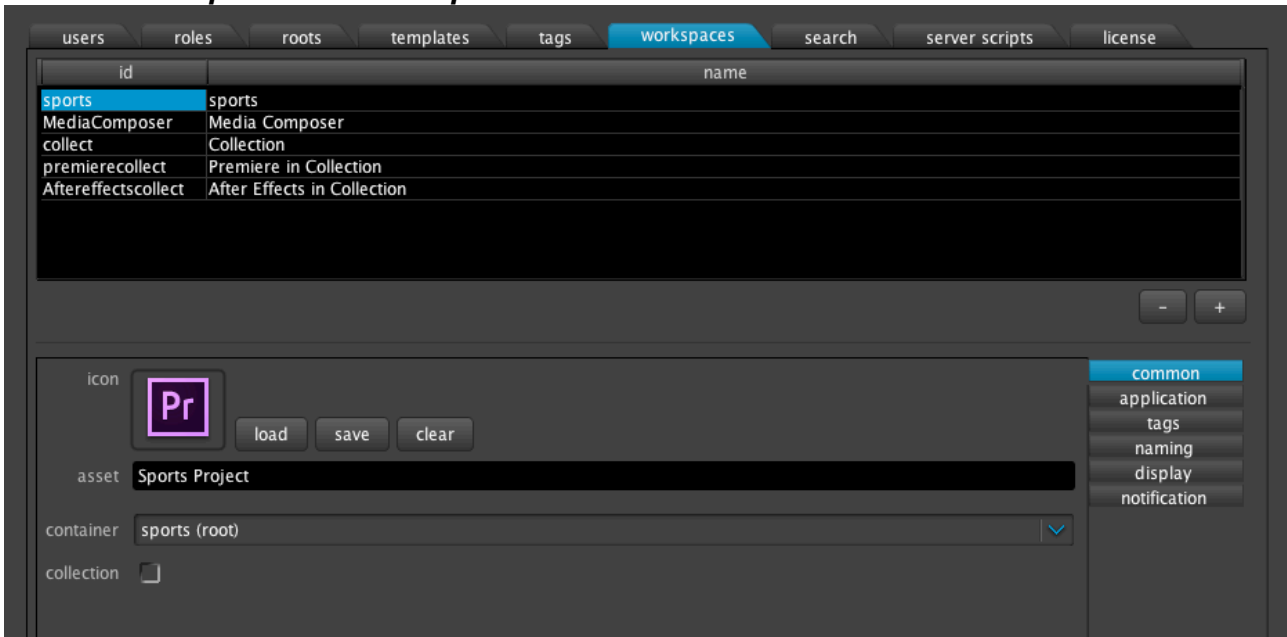
2.1.6 The workspaces Tab



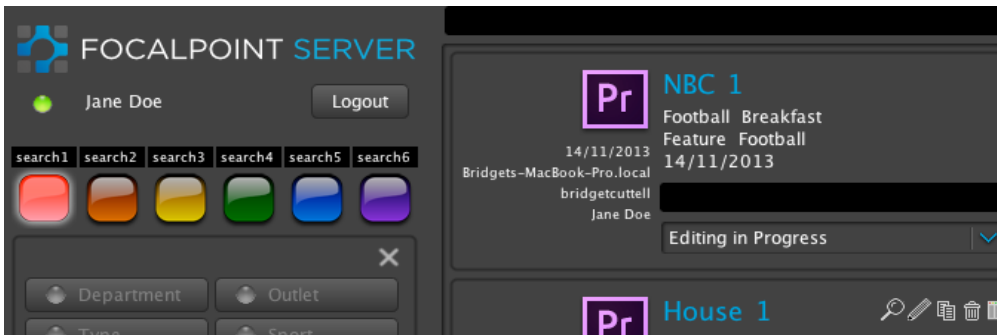
The Workspaces tab allows the administrator create, remove and edit Workspaces. (Definition of Workspace 1.1.2) Here the name, the icon, the application, the manner the project is displayed etc can all be changed.

Within the workspaces tab, there are 6 additional panels for each workspace: common, application, tags, naming, display, and notification.

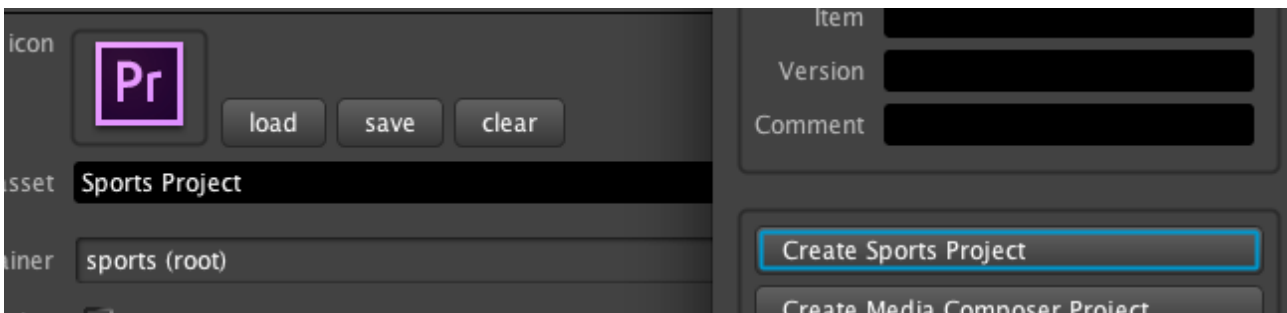
2.1.6.1 Workspaces/common panel:



Using the common tab, one can define the id and name associated with workspaces and Portfolio workspaces. An icon can be chosen for each application by using the file requestor to navigate to a suitable png or tiff file. This icon will then be used to represent the application in the Client Interface, as seen below.



Furthermore, one can define the custom name on the **Create** button in the Client Interface with the **asset** field.



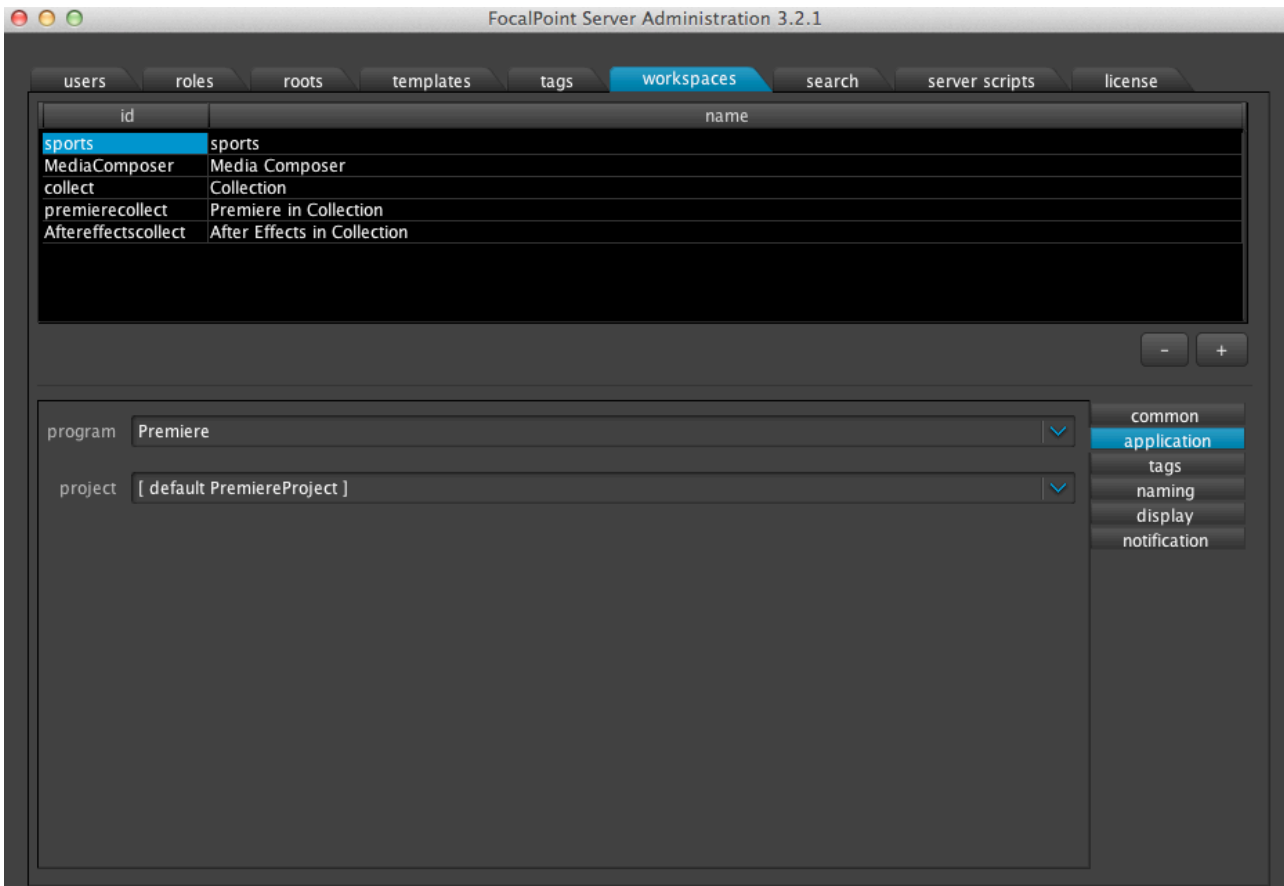
In the **container** field the appropriate id is chosen from the roots tab.

A Portfolio will have a **container: Root** as well as the Collection box ticked. The id will be a root of the same name created in the roots tab. From then on every application that you want to appear in that Portfolio must also have the same root. (In the case above, adding a Final Cut

possibility to the Sports Project button, you must have a new workspace for Final Cut within the Portfolio, and this workspace will have the **container: Sports (root)**.

In order to add a separate single application Workspace – i.e a button **saying Create Premiere Project** – you will need the **container** to be a **root** of its own. The **root** you have already created in the **roots** tab as Premiereroot, for example.

2.1.6.2 Workspaces/application panel:



The application for the Workspace is chosen here, under **program**. Under **default**, the template from the templates tab is selected. (Note, there can be multiple templates for one application if you wish to start multiple different projects from one application).

One can also choose the **program** to be **Generic** (eg., for simple applications like text editors that do not require special treatments of files and GUIs) and specify regular expression patterns for paths (the default value is `.*`).

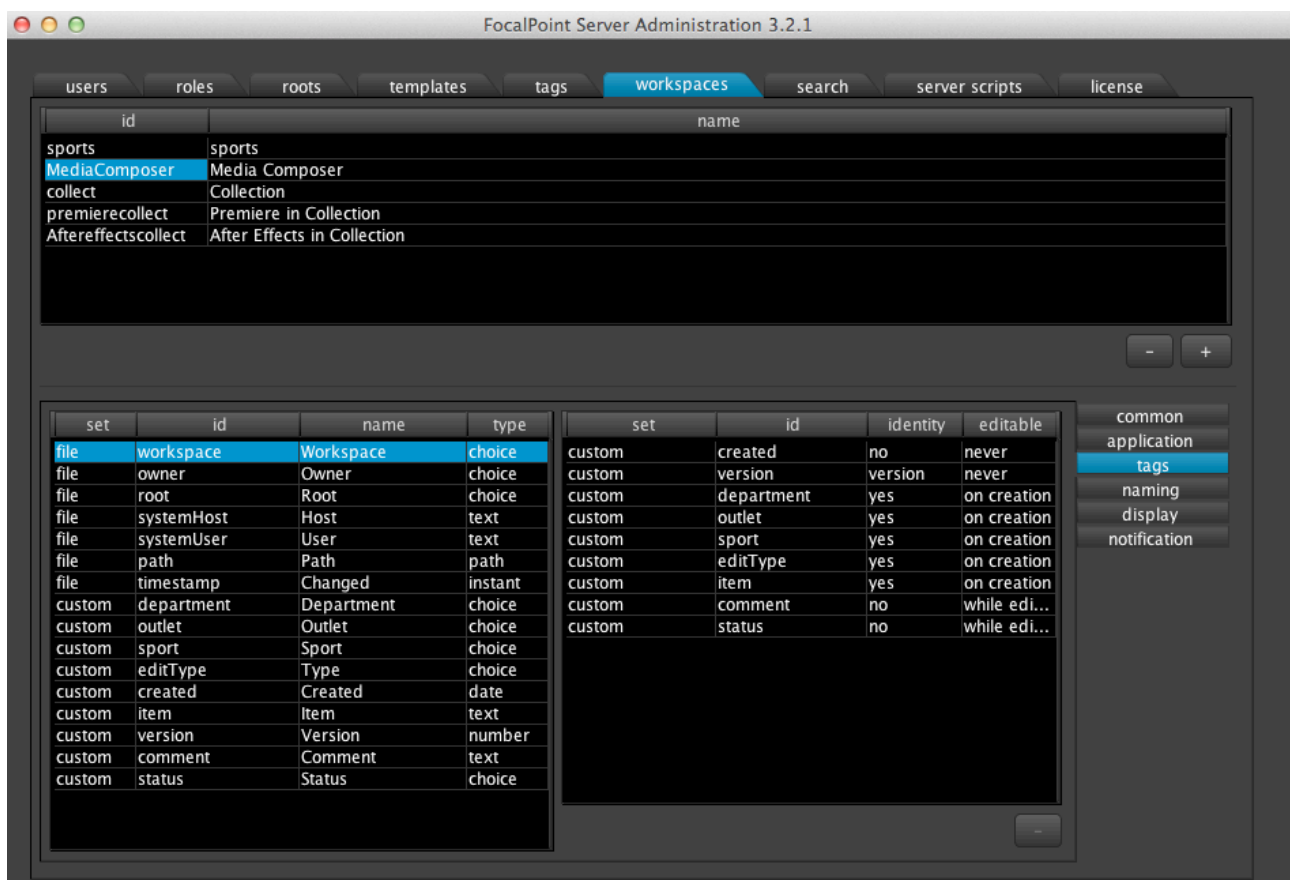
However, the most advanced feature here is to choose the **program** to be **Scripted**. Upon choosing this, one can write and invoke clients scripts, actioned by the Client Interface. Any custom actions can be scripted here, such as media being dragged and dropped (in case of Model B and Portfolios).

Supported for Mac OS X clients: applescript, bash, perl, and javascript
Supported for Windows clients: vbscript and javascript
Supported for Linux clients: bash, perl and javascript

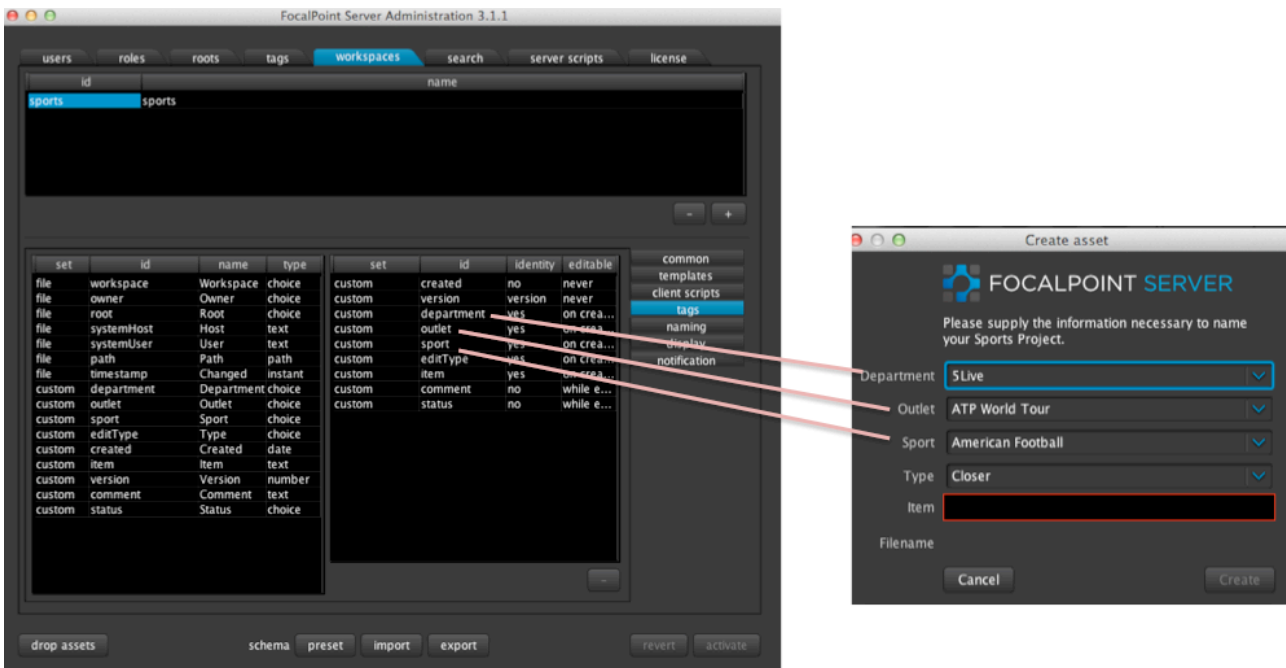
For example, new projects can be created based on the template files, loaded in the templates panel and called via scripts. Two example scripts are given in Appendix A: (i) an applescript that performs custom actions for a FCP7 workspace in Model A (ii) a perl script that performs custom actions for a Portfolio workspace in Model B, including communicating with an external program (Root6's Content Agent) via its SOAP interface for specific actions.

The variables in such scripts that can be passed to a script from FocalPoint Server are based on the tags in the schema and are prefixed with fp_(see examples at the beginning of these scripts referred to above).

2.1.6.3 Workspaces/tags panel:



The pool of tags defined in the main **tags** tab, are shown here on the left along with certain system tags. The administrator can drag and drop tags from that panel to the panel on the right. These selected tags on the right are what will occur in the Client Interface upon the creation of a project, as seen below.



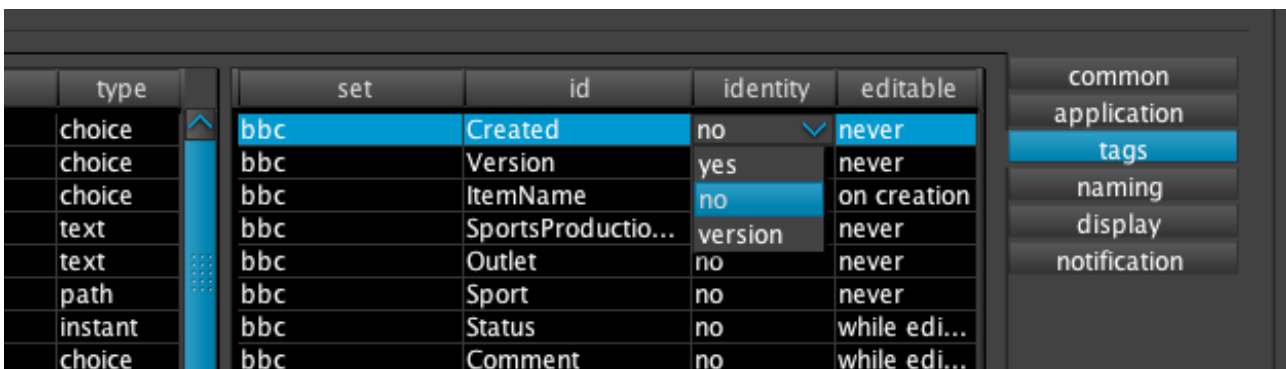
The order in which these tags are placed defines their position within the menu. This menu can vary for each Workspace.

Identity column: every asset must have a unique identity. No two tags can have the same name. An asset is unique if at least one of the tags in its identity is different from all other assets.

Yes: this tag is considered when checking if an asset has a unique id (e.g item name)

No: this tag is not considered when checking if an asset is unique (e.g. 'Created'- timestamp or 'Sport' – there can be many projects about the same sport.)

Version: an incremental increase every reversion. E.g. 1 to 2



editable: when or if the tag is editable for the user.

Never: The user can never edit this, e.g. a timestamp

On Creation: Upon the creation of a project (**Create Project** menu, e.g. Department, Episode etc)

While Editing: Anytime while the Project appears in the Project list on the right of the Client Interface, e.g. a status or comment field.

Note that here, Sport is labeled as **editable – never** because in the panel on the right

set	id	identity	editable
bbc	Created	no	never
bbc	Version	version	never
bbc	ItemName	yes	on creation
bbc	SportsProductio...	no	while editi
bbc	Outlet	no	never
bbc	Sport	no	never
bbc	Status	no	while edi...
bbc	Comment	no	while edi...

common

application

tags

naming

display

notification

Note:

- If a tag can be changed "while editing", that is after the asset is already created, then you can never set it to "yes" within **identity**. This is because that would mean the identity/name of an asset would change over time, and then not be searchable by the original tags.
- When re-versioning, everything must be the same, but you want an automatically incremented version number, so it must be a number tag in the **tags** tab, and never editable in **Workspace/tags**.

FocalPoint Server Administration 3.2.1

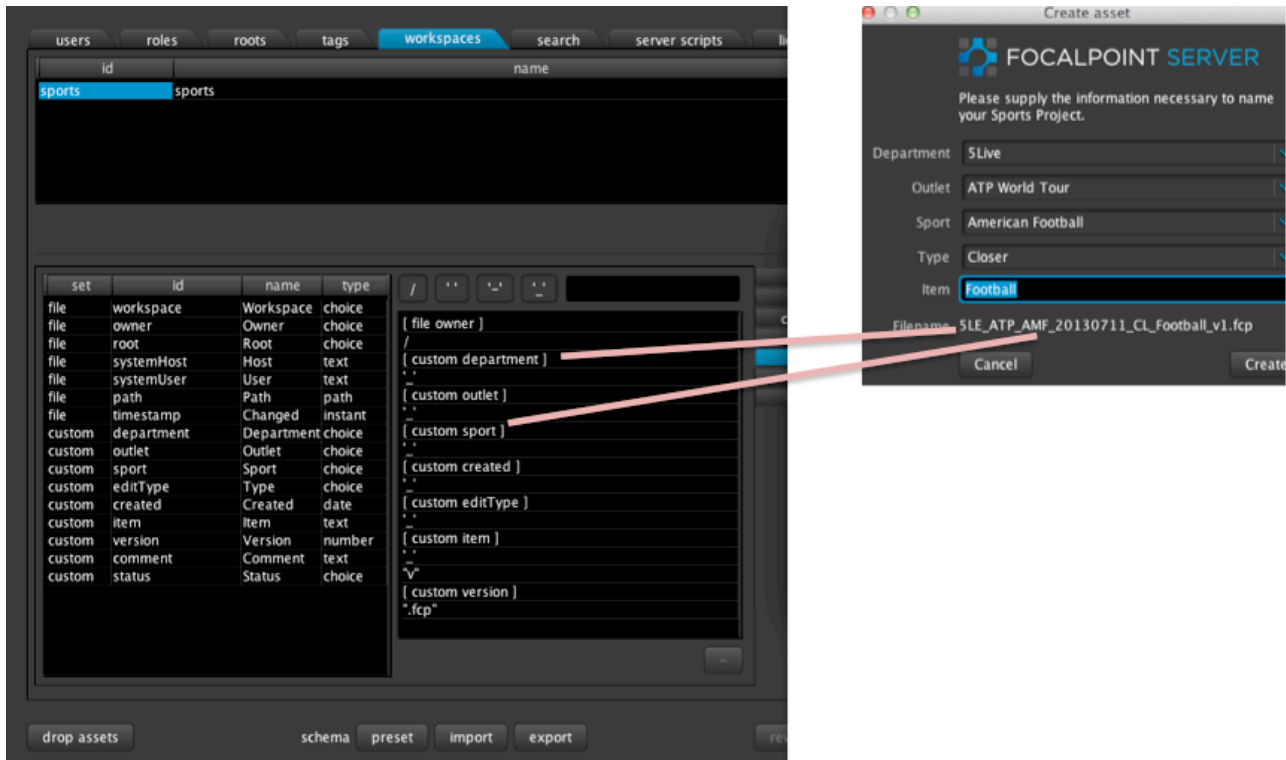
users
roles
roots
templates
tags
workspaces
search
server scripts

set	id	name	type
bbc	Outlet	Outlet	choice
bbc	Sport	Sport	choice
bbc	Created	Created	date
bbc	ContentProducer	Content Producer	text
bbc	ProgrammNumber	Programm Number	text
bbc	ContentLink	Content Link	choice
bbc	ProjectName	Container Name	text
bbc	ItemName	Project Name	text
bbc	ClipDescription	Clip Description	multi
bbc	MediaSource	Media Source	choice
bbc	OriginalRecording...	Original Recording Date	date
bbc	RetentionNotes	Retention Notes	text
bbc	RetentionRecomm...	Retention Recommen...	choice
bbc	TalentUid	Talent UID	text
bbc	RightsUid	Rights UID	text
bbc	KillDate	Kill Date	date
bbc	RightsTrafficLight	Rights Traffic Light	choice
bbc	JupiterInfo	Jupiter Info	text
bbc	DestinationTv	Destination TV	choice
bbc	RestinationRadio	Destination Radio	choice
bbc	RequestedBy	Requested By	text
bbc	SourceId	Source ID	text
bbc	Version	Version	number
bbc	Status	Status	choice
bbc	Comment	Comment	text
ca	Host	CA Hostname	text
ca	TemplateFolder	CA Template Folder	text
ca	TemplateName	CA Template Name	text
ca	MediaGridFolder	Media Grid Folder	text
ca	WinMediaGridFolder	Windows Media Grid F...	text
bbc	Type	Type	choice

default 1

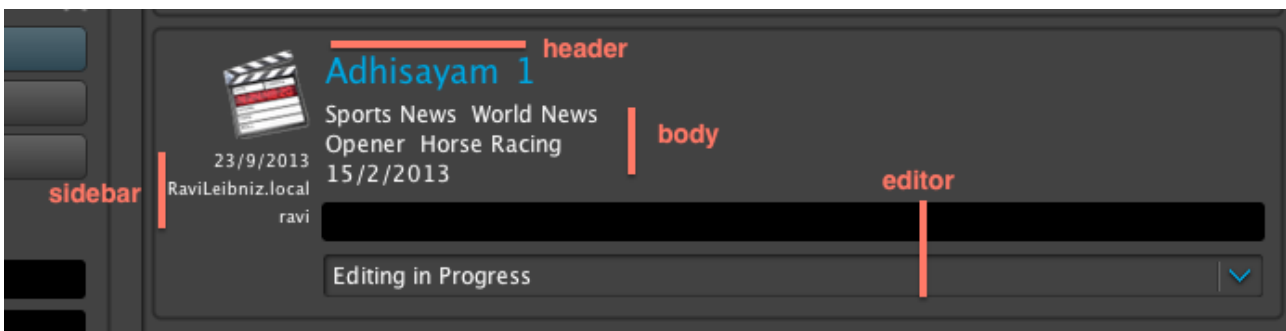
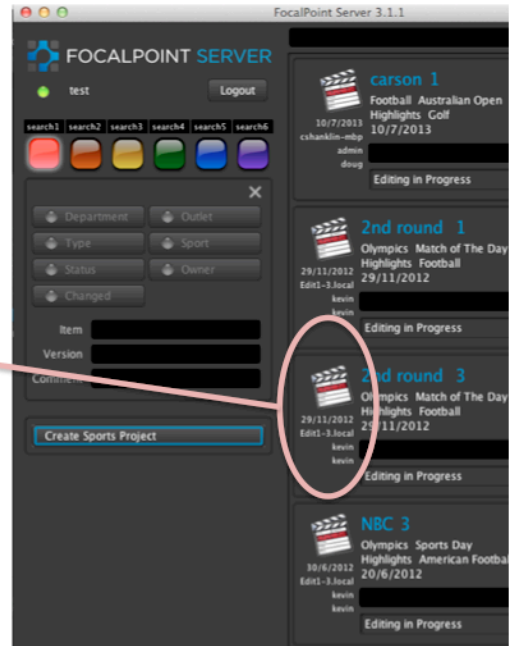
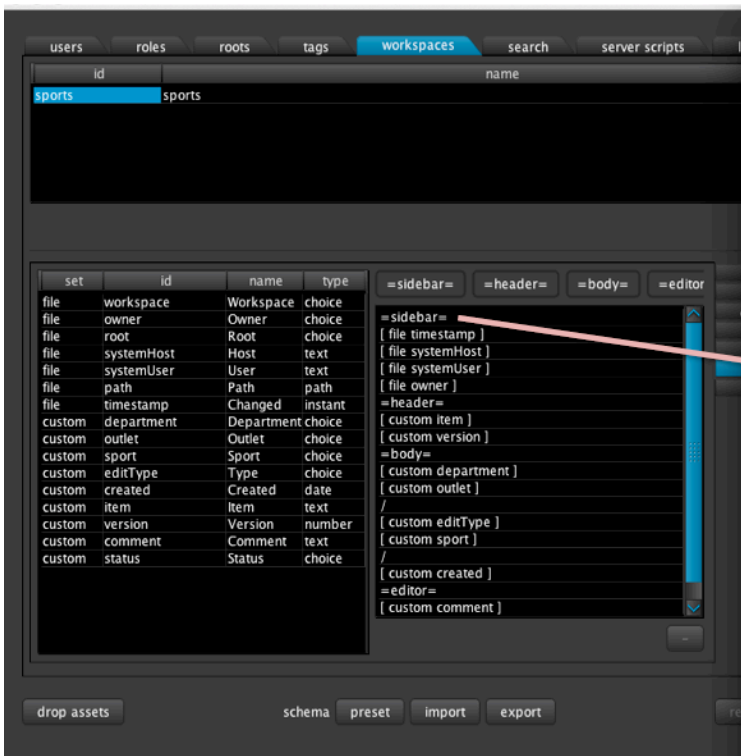
2.1.6.4 Workspaces/naming panel:

The naming panel allows the administrator to define how to name a project; in what combination acronyms of the tags appear to name the project file.



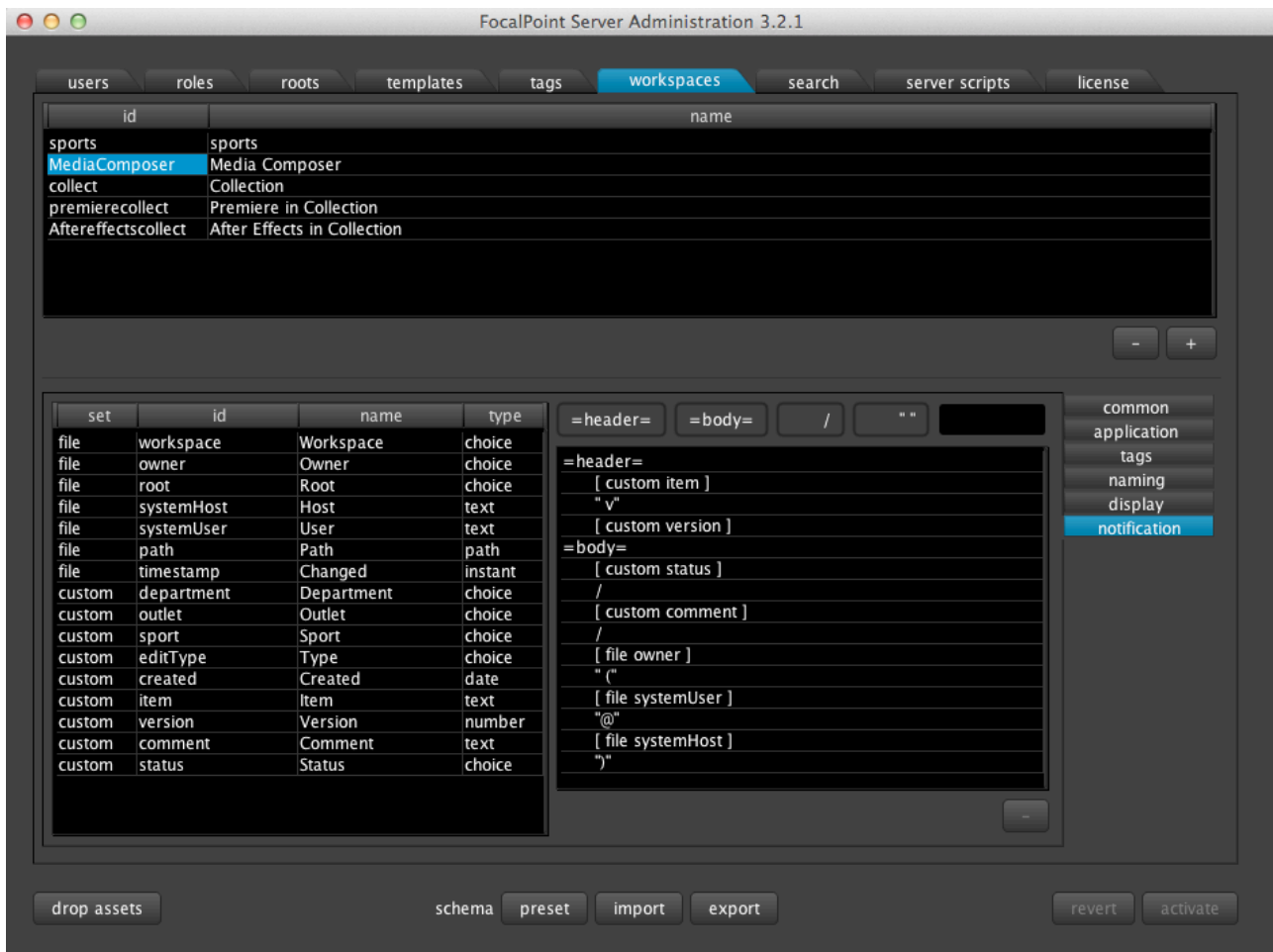
These tags be dragged and dropped from the tags in the left-hand side. Separators in the naming are dragged from the panel above the list on the right-hand side. Extra formatting (e.g. "v") and file extension (e.g. .fcp or .aep) options are provided here by typing them into the text field above the right panel, and then dragging and dropping them into place.

2.1.6.5 Workspaces/display panel:



The display panel controls how the project information is displayed on the Project List of the Client Interface. This again consists of tags that are dragged and dropped from the tags in the left-hand side. The Toolbar elements sidebar, header, body and editor relate to portions of the FocalPoint Server client application display.

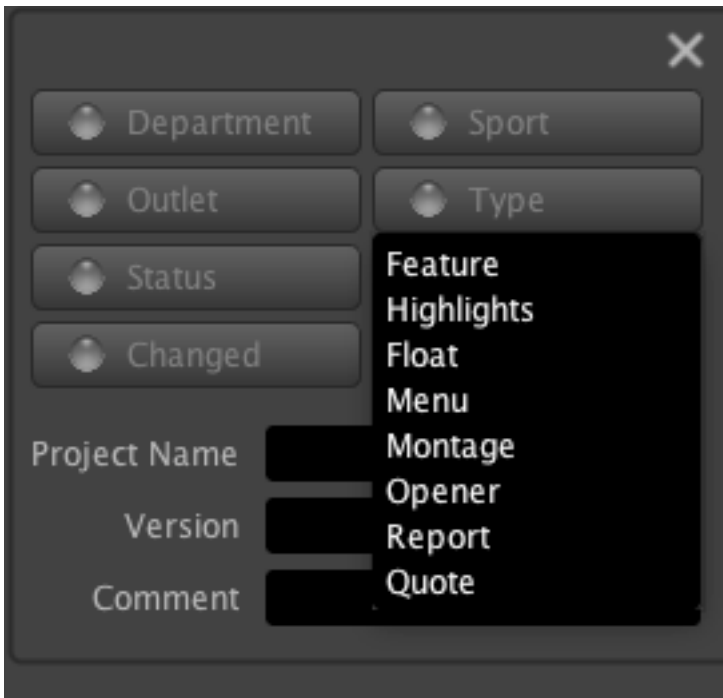
2.1.6.6 Workspaces/notification panel:



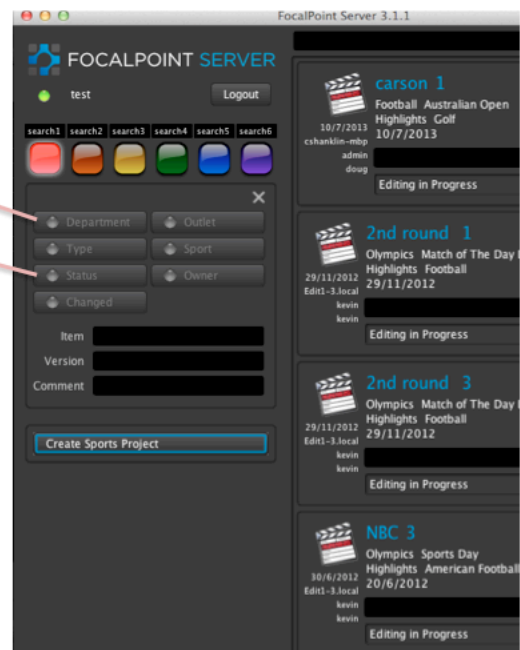
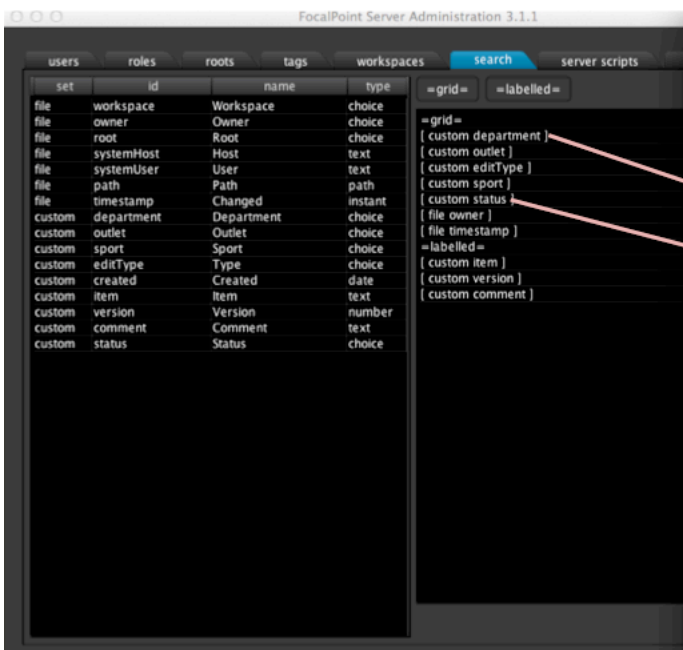
The notification panel controls what information the Growl notification server displays to the users (assuming Growl is installed). Having Growl notifications means information is displayed to all users who are logged into the FocalPoint Server network upon the change of Status of projects. Even if the user has “buried” the FocalPoint Client Server window under other applications, information will still be displayed via Growl. Growl itself must be installed upon each users machine for this mechanism to function. The latest versions of Growl also provide “Roll-Up’s” that will inform the user of any changes to projects from users logged into FocalPoint Server even when they have not been in attendance, so that no alerts are missed. As with the display panel, tags can be dragged from the left panel to the right to make up what appears in the notification, as well as where it is displayed.

2.1.7 The search Tab

The search tab allows the FocalPoint Server administrator to define the search button matrix on the Client Interface. It is also possible to add freeform text searches underneath if required.



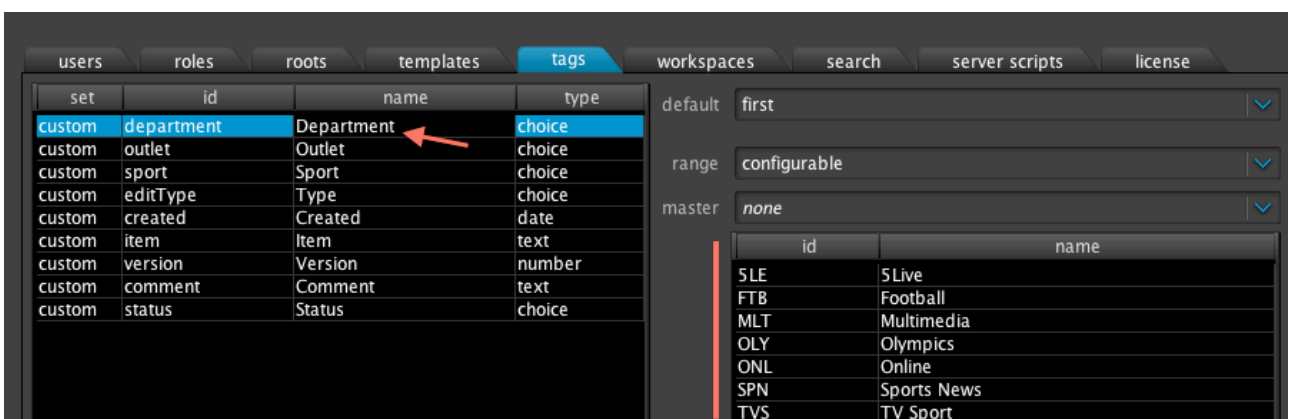
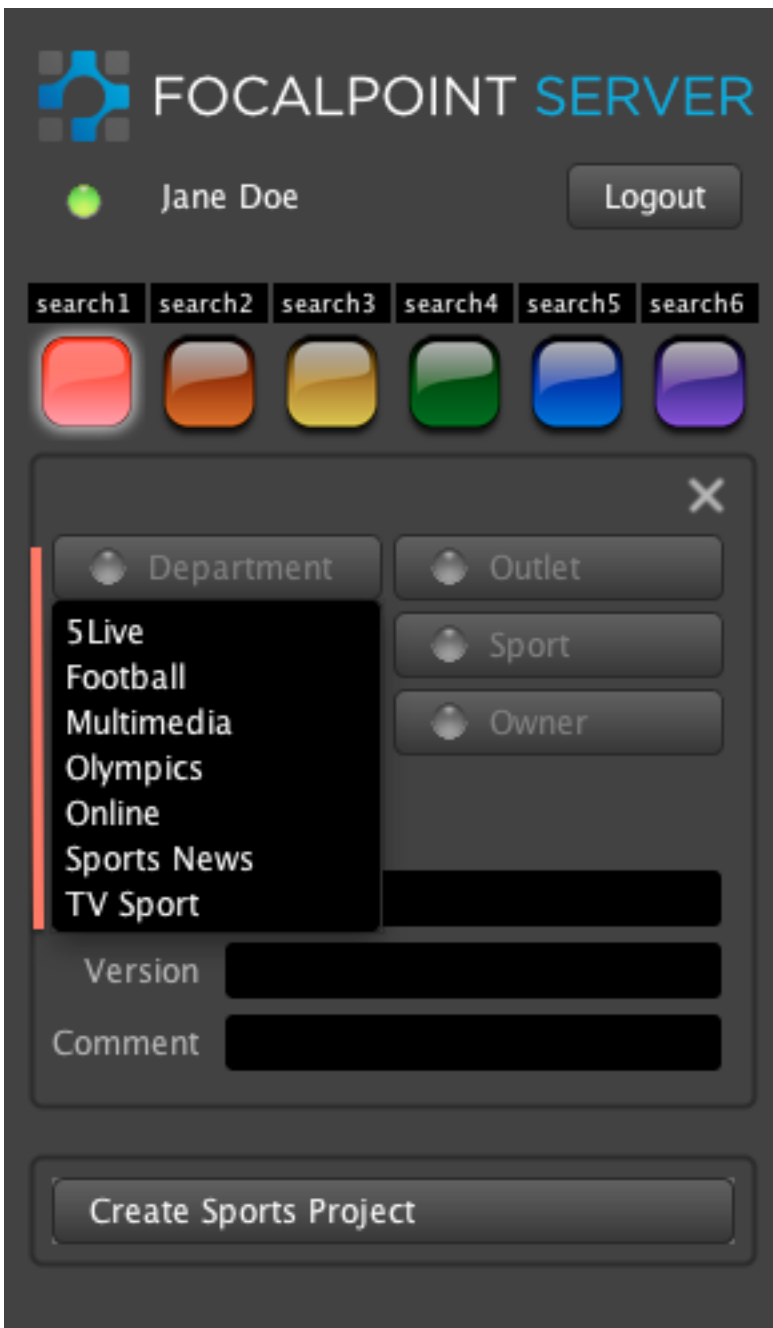
These searches filter all assets in your Project List on the right. Multiple search tags can be selected to narrow down a search.



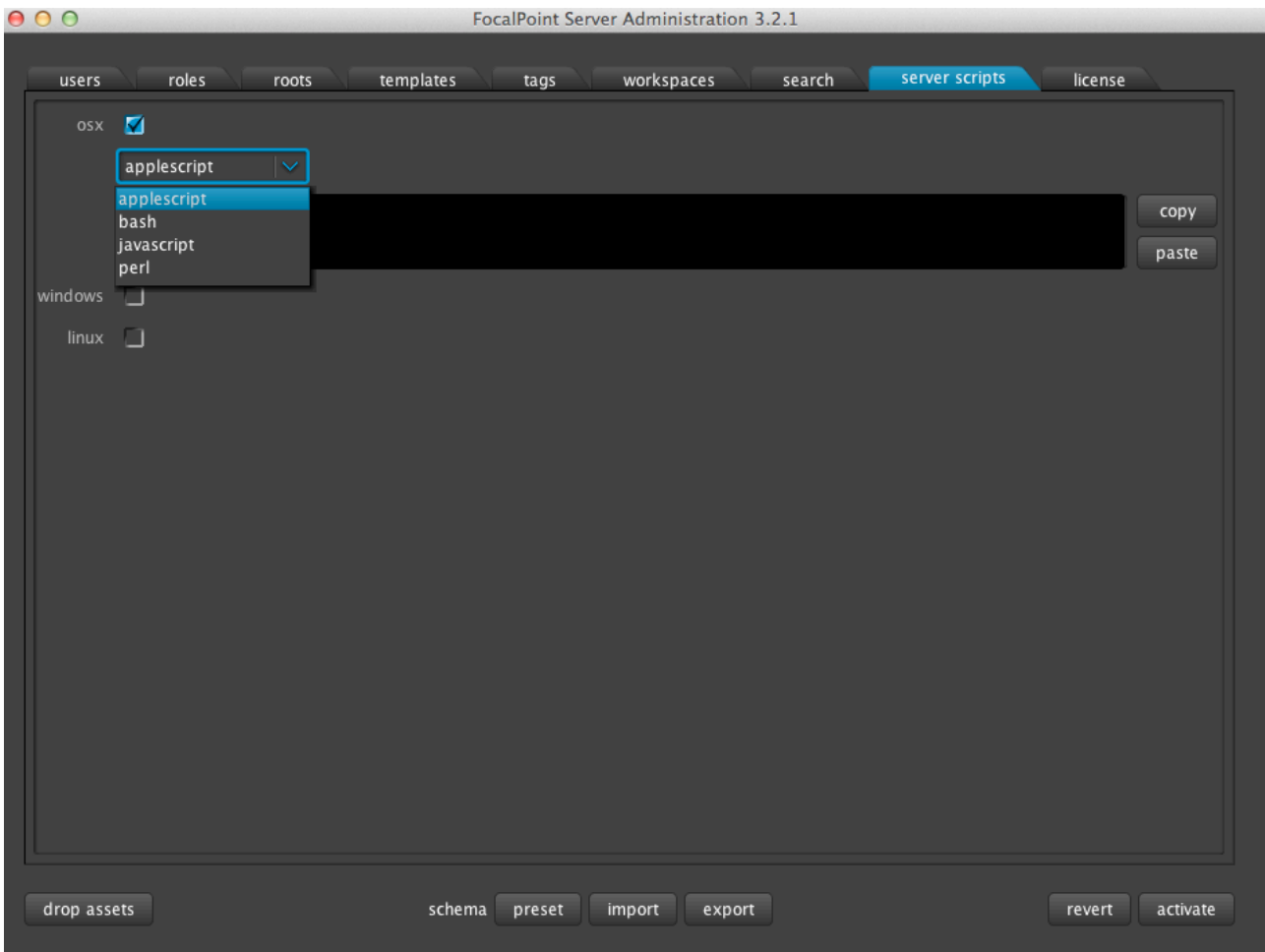
Again similar to the workspaces tab, the left-hand panel contains all the tags defined under the tags tab, along with certain system tags. These tags can be dragged and dropped to the right-hand panel to define the search matrix.

- =**grid**= search buttons in the grid
- =**labelled**= optional freeform text fields underneath

Note; The order that the possible tags appear under the search button is dictated by the order they are entered previously in the tags tab.

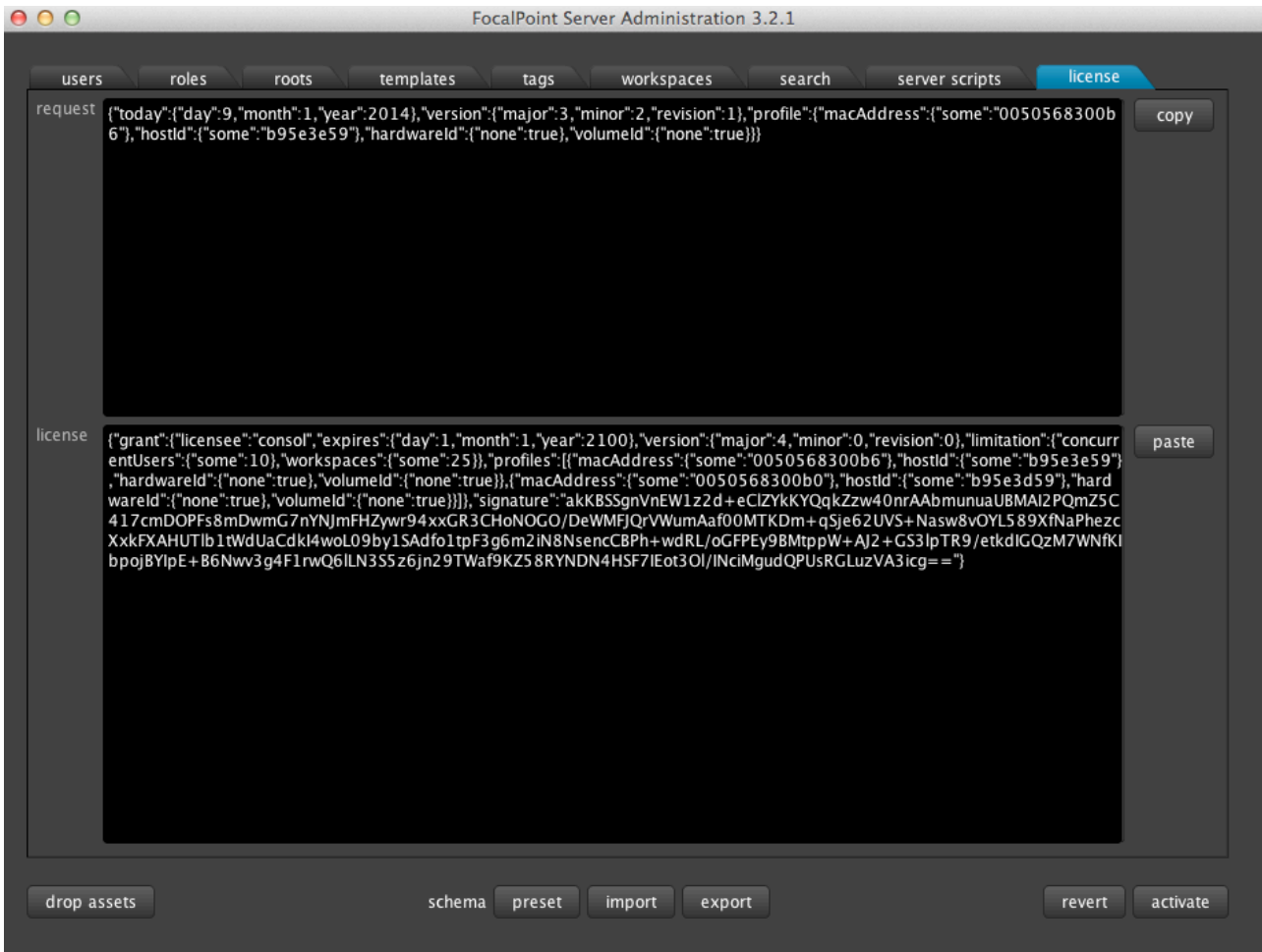


2.1.8 The server scripts Tab



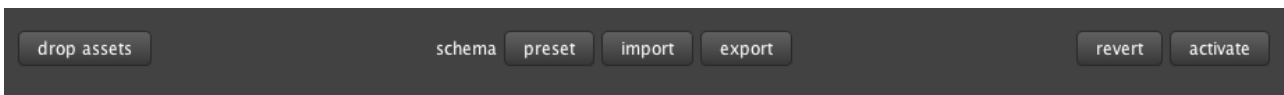
This is a scripting API on the server side and can be highly customized for individual workflows (eg., e-mail notifications, status reports in conjunction with third party applications and more). Scripts here can be arranged through your FocalPoint Server contact or (with permission) with a coder within the client company.

2.1.9 The licence Tab



The license tab has two windows: request and license. The string in the request window should be provided to the vendor, who will provide the appropriate license string. This should be pasted in the license window.

2.1.10 The import/export buttons



On clicking the preset button, the administrator will be presented with a set of pre-defined schemas that can be loaded and modified according to the workflow needs of a specific facility.

The **export** button can be used to save an existing schema (with a .fps extension) to anywhere on your machine.

The **import** button can be used to load a previously exported schema form anywhere on your machine.

2.1.11 Other basic buttons

Revert button – restores to last saved state (the last time the **activate** button was pressed)

Activate button – updates all changes made on Client Interface (it also saves the changes).

Drop Assets button - While many changes to the schema do not require dropping assets, some do. For example, adding a new department or new user do not require dropping assets. Other changes require the database to refresh itself. For example, changing tags within the naming convention will require current assets to change their names. In such cases, all the assets should be dropped from the database, and automatically re-entered by clicking the **drop assets** button.

3. Disaster Recovery

FocalPoint Server uses **mongoDB** as its backend database. It is important to maintain a backup strategy to recover from catastrophic scenarios, even in the case of High Availability installations.

Any effective mongoddb backup strategy will be effective for FocalPoint Server.

An example backup/restore scenario is provided below using **mongoDB** tools.

3.1 Backing up data using mongodump tool

Always use the FocalPoint Server shipped versions of `mongodump` and `mongorestore` tool, this will ensure restore operations work.

The database backup should always be planned and performed during a period of low to zero activity.

If a HA FocalPoint Server is being backed up, please make sure the backups are performed from one of the secondary member(s). This allows the master (production) mongoddb to serve data. Hence there is no service outage during backup.

We recommend a slave/separate node for backing up the database: a dedicated server which is hidden and with no voting rights, and exclusively used for performing backups. The slave node need not match the primary and/or secondary servers in CPU and memory requirements, but should have adequate disk space. In addition, the following considerations apply.

- `mongodump` can not perform full dump and compress data on the fly.
- `mongorestore` can not perform restore on compressed data.

3.2 Setting up the backup server

Install the `mongoddb.deb` from the FocalPoint Server installation package.

In a normal HA setup first server is always the master and second server is always acting as slave. The backups are stored under `/opt/fpoint/backup/dmp`.

The following actions are performed as the `root` user in a terminal window.

(i) Create the backup directory:

```
mkdir -p /opt/fpoint/backup/dmp
```

(ii) Save the following script under `/usr/local/bin` as `fpointdmp.sh` after replacing the value of the `DBHOST` below with that of the backup server.

```
#!/bin/bash

PATH=/usr/local/bin:/usr/bin:/bin
MONGODMP=/opt/fpoint/mongo/mongodb/bin/mongodump
DBHOST=fpserver-02
DBPORT=27017
TAR=/bin/tar
BACKUPDIR=/opt/fpoint/backup/dmp
DATE=`date +%Y_%m_%d_%H_%M_%S`
FILENAME=focalpoint_dmp_${DATE}.tar.bz2

dumping_fpoint(){
    cd ${BACKUPDIR} && $MONGODMP --host=$DBHOST:$DBPORT --oplog \
-o DUMP_${DATE} && $TAR -cjf $FILENAME DUMP_${DATE}
    [ $? -eq 0 ] && return 0
    echo "ERROR: mongodump failed to create dumpfile: $1" >&2
    return 1
}

cleanup (){
    echo 'cleaning up....'
    cd $BACKUPDIR && rm -rf DUMP_${DATE}
    [ $? -eq 0 ] && return 0
    echo "ERROR: cleanup failed" >&2
    return 1
}

echo "Backup of Database Server $HOST"

dumping_fpoint && cleanup
echo "Dump of focalpoint db is performed - $FILENAME"
```

(iii) To perform backups just invoke `fpointdmp.sh` from the terminal command line. It can also be invoked via cron for regular backups.

3.3 Restoring backed up data using mongorestore tool

First copy the backed up data to a location with enough disk space. In the following scenario the directory `/opt/fpoint/restore` is used. All actions are performed as the root user in a

terminal window.

(i) Create the restore directory

```
mkdir -p /opt/fpoint/restore
```

(ii) Copy the dump file to the restore directory. In the step below, the example restore file is focalpoint_dmp_2013_09_23_15_05_38.tar.bz2.

```
cp focalpoint_dmp_2013_09_23_15_05_38.tar.bz2 /opt/fpoint/restore
```

(iii) Uncompress the contents of the restore file.

```
tar jxvf focalpoint_dmp_2013_09_23_15_05_38.tar.bz2
```

(iv) perform restore action (all in one line). In the example below, substitute the name of fpserver-01 with that of the master server.

```
/opt/fpoint/mongo/mongodb/bin/mongorestore -h fpserver-01  
-d focalpoint -drop  
/opt/fpoint/backup/mongodb/DUMP_2013_03_22_11_57_16/focalpoint
```

A-Appendix

(i) An example of an applescript to create Final Cut Pro 7 projects with specific templates for a scripted workflow in Model A.

```
-----  
-----  
---- parameters  
-----  
-----  
  
-- fp_action          String    view, edit, create, reversion or  
delete  
  
-- fp_created         File new file to exist after this script  
-- fp_original        File original data in a read-only temporary  
file  
-- fp_deleted         File project file to be gone after this script  
-- fp_edited          File existing project file to be opened for  
editing  
-- fp_viewed          File existing project file to be viewed  
-- fp_revealed        File existing bin file to be revealed in  
finder  
-- fp_updated         File existing project file that has been  
changed
```

```

-- fp_template_0    File readonly empty template project file
-- fp_template_1    File readonly timeline XML file
-- fp_template_2    File readonly default project's bin XML file
-- fp_template_3    File readonly warning timeline for view action

-- fp_xyz_outlet    String      xyz outlet
-- fp_xyz_item      String      xyz item
-- fp_xyz_version    Number      xyz version
-- fp_xyz_created    Date xyz creation date (DD/MM/YYYY)

```

```

-----
-----
---- final cut utilities
-----
-----

```

```

property kFCPSaveAndCloseProject : 0
property kFCPDiscardAndCloseProject : 1

```

```

--start FCP
on FCPActivate()
    tell application "Final Cut Pro" to activate
end FCPActivate

```

```

--get FCP to open a file
on FCPOpen(aFile)
    tell application "Final Cut Pro"
        «event KeyGofcP» given «class fcpP»:aFile
    end tell
end FCPOpen

```

```

--get FCP to insert XML into a project (you refer to the project
as a file)
on FCPInsertXML(aFile, someXML)
    tell application "Final Cut Pro"
        «event KeyGiXML» given «class fcpP»:aFile, «class
xmlID»:someXML
    end tell
end FCPInsertXML

```

```

--get FCP to Close ProjectFile (and optionally save as per the
flags)
on FCPSaveAndClose(aFile, saveFlags)
    tell application "Final Cut Pro"
        «event KeyGcfcP» given «class fcpP»:aFile, «class
fcCF»:saveFlags
    end tell
end FCPSaveAndClose

```

```

--get FCP to save aFile as newFile
on FCPSaveAs(aFile, newFile)
    tell application "Final Cut Pro"

```

```

        «event KeyGsfcP» given «class fcpP»:aFile, «class
fcpU»:newFile
        end tell
end FCPSaveAs

```

```

-----
-----
---- string utilities
-----
-----

```

```

on replace(source, search, replacement)
    set oldDelim to AppleScript's text item delimiters
    set AppleScript's text item delimiters to the search
    set the item_list to every text item of source
    set AppleScript's text item delimiters to the replacement
    set output to the item_list as string
    set AppleScript's text item delimiters to oldDelim
    return output
end replace

```

```

on shellEscape(aFile)
    set aFilePath to (POSIX path of aFile)
    "" & replace(aFilePath, "'", "'\''\'") & ""
end shellEscape

```

```

-----
-----
---- file utilities
-----
-----

```

```

on mkParentDirs(aFile)
    set aShellPath to shellEscape(aFile)
    do shell script "mkdir -p \"$(dirname " & aShellPath & ")\\""
end mkParentDirs

```

```

on copyFile(aSource, aTarget, anAllowOverwrite)
    -- tell application "Finder" to duplicate aSource to aTarget
    set aSourcePath to shellEscape(aSource)
    set aTargetPath to shellEscape(aTarget)
    set shellScript to "cp " & aSourcePath & " " & aTargetPath
    if not anAllowOverwrite then
        set shellScript to "[ -f " & aTargetPath & " ] || " &
shellScript
    end if
    do shell script shellScript
end copyFile

```

```

on writableFile(aTarget)
    set aTargetPath to shellEscape(aTarget)
    set shellScript to "chmod u+w " & aTargetPath
    do shell script shellScript
end writableFile

```

```

on deleteFile(aFile)
    tell application "Finder" to delete aFile
end deleteFile

-----
-----
---- actual work is done here
-----
-----

on formatDate(input)
    set oldDelims to AppleScript's text item delimiters
    try
        set AppleScript's text item delimiters to "/"
        set wantedItems to text items 1 thru 2 of input
        set AppleScript's text item delimiters to "-"
        set output to wantedItems as string
        set AppleScript's text item delimiters to oldDelims
        return output
    on error
        set AppleScript's text item delimiters to oldDelims
    end try
end formatDate

on ViewFileName(myFile)
    set oldDelimiters to AppleScript's text item delimiters
    set AppleScript's text item delimiters to {" ":""}
    set pathItems to text items of (myFile as text)
    set fileName to "VIEWING_" & last item of pathItems
    set last item of pathItems to fileName
    set first item of pathItems to ""
    set AppleScript's text item delimiters to {"/"}
    set resultFile to pathItems as string
    set resultPath to POSIX file resultFile
    set AppleScript's text item delimiters to oldDelimiters
    set Sourcefile to POSIX path of myFile
    set Destination to POSIX path of resultPath
    do shell script "mv " & quoted form of Sourcefile & " " &
quoted form of Destination
    return resultPath
end ViewFileName

on fpop_create()
    -- "%s_%s_v%s_%02d-%d"
    set timelineName to fp_xyz_Outlet & "_" & fp_xyz_ItemName &
"_v" & fp_xyz_Version & "_" & formatDate(fp_xyz_Created)

    -- must contain **REPLACE TIMELINENAME**
    set timeLineTemplate_xml to (read fp_template_1 as «class
utf8»)

    set defaultBins to (read fp_template_2 as «class utf8»)

```

```

-- mkParentDirs(fp_created)
  FCPActivate()
  FCPOpen(fp_template_0)
  set namedTimeline to replace(timelineTemplate_xml, "**REPLACE
TIMELINENAME**", timelineName)
  FCPInsertXML(fp_template_0, namedTimeline)
  FCPInsertXML(fp_template_0, defaultBins)
  FCPSaveAs(fp_template_0, fp_created)
end fpop_create

on fpop_delete()
  deleteFile(fp_deleted)
end fpop_delete

on fpop_edit()
  FCPActivate()
  FCPOpen(fp_edited)
end fpop_edit

on fpop_reversion()
  -- "%s_%s_v%s_%02d-%d"
  set timelineName to fp_xyz_Outlet & "_" & fp_xyz_ItemName &
  "_v" & fp_xyz_Version & "_" & formatDate(fp_xyz_Created)

  -- must contain **REPLACE TIMELINENAME**
  set timeLineTemplate_xml to (read fp_template_1 as «class
utf8»)

-- mkParentDirs(fp_created)
  FCPActivate()
  FCPOpen(fp_original)
  set namedTimeline to replace(timelineTemplate_xml, "**REPLACE
TIMELINENAME**", timelineName)
  FCPInsertXML(fp_original, namedTimeline)
  FCPSaveAs(fp_original, fp_created)
end fpop_reversion

on fpop_view()
  set viewTemplate_xml to (read fp_template_3 as «class utf8»)
  FCPActivate()
  set viewFile to ViewFileName(fp_viewed)
  FCPOpen(viewFile)
  FCPInsertXML(viewFile, viewTemplate_xml)
end fpop_view

on fpop_reveal()
  tell application "Finder"
    reveal fp_revealed
    activate
  end tell
end fpop_reveal

on fpop_drop()
end fpop_drop

```



```

on fpop_updated()
end fpop_updated

-- dispatch function
if fp_action = "create" then
    fpop_create()
else if fp_action = "reversion" then
    fpop_reversion()
else if fp_action = "delete" then
    fpop_delete()
else if fp_action = "view" then
    fpop_view()
else if fp_action = "edit" then
    fpop_edit()
else if fp_action = "reveal" then
    fpop_reveal()
else if fp_action = "drop" then
    fpop_drop()
else if fp_action = "updated" then
    fpop_updated()
else
    display dialog ("unexpected fp_action " & fp_action)
end if

```

(ii) An example of a perl script to create a Portfolio in shared storage storage and perform appropriate pre-defined actions in Model B:

```

#-----
# parameters
#-----

# fp_action      String  view, edit, create, reversion or delete

# fp_created     File     new file to exist after this script
# fp_original    File     original data in a read-only temporary
file
# fp_deleted     File     project file to be gone after this script
# fp_edited      File     existing project file to be opened for
editing
# fp_viewed      File     existing project file to be viewed
# fp_updated     File     existing project file that has been
changed
# fp_dropped     File     dropped file location

```

```
# fp_template_project_xyzSportCMP File readonly "XYZ Sport
CMP.xml"
```

```
#-----
-----
```

```
# prepare
```

```
#-----
-----
```

```
use File::Copy qw/ copy cp /;
use File::Path qw(make_path remove_tree);
use LWP::UserAgent;
use Data::Dumper;
```

```
sub get_templates;
sub get_template_by_name;
sub submit_encoding_job;
sub get_wolkflowstatus;
sub do_drop;
sub get_clip_info;
sub get_sourcefilemetadata;
```

```
my $dispatch = {
    create      => sub { make_path($fp_created);
                        system "cp",
"$fp_template_project_xyzSportCMP", "$fp_created"; },
    delete     => sub { remove_tree($fp_deleted); },
    edit       => sub { print "opening $fp_edited"; },
    reversion  => sub { system "cp", "-R", "$fp_original",
"$fp_created"; },
    view       => sub { print "viewing $fp_viewed"; },
    reveal     => sub { system "open", "$fp_revealed"; },
    update     => sub { print "### updated $fp_updated"; },
    drop      => sub { my $clip_desc = get_clip_info;
                        do_drop($clip_desc);
                    },
};
```

```
if ( exists $dispatch->{$fp_action} ) {
    $dispatch->{$fp_action}();
} else {
    print "### unknown action: $fp_action";
}
```

```
##### CA Stuff
```

```
sub do_drop {
    my $clip_desc = shift;

    my $project = (split "/", $fp_original)[-1];
    my $dropdir = "$fp_tag_ca_MediaGridFolder/fp_" . $project .
'_ ' . $clip_desc . '_v1';

    while (-e $dropdir) {
        $dropdir =~ s/^(.+)(\d+)$/{my $r=$2;$r++;"$1$r"}/e;
```

```

    }
    make_path($dropdir);
    make_path("$dropdir/transcoded");

    my $windropdir = $dropdir;
    $windropdir =~
s/^\$fp_tag_ca_MediaGridFolder(.+)/\$fp_tag_ca_WinMediaGridFolder$1/
;

    open(my $log, ">", "$dropdir/transcoding.log")
        or die "cannot open > transcoding.log: $!";

    system "cp", "$fp_dropped", "$dropdir";
    system "cp", "$fp_template_0", "$dropdir";
    system "cp", "$fp_template_0", "$dropdir/transcoded";
    my $dropfile = (split "/", $fp_dropped)[-1];

    my $ua = LWP::UserAgent->new( agent => 'Apache-
HttpClient/4.1.1 (java 1.5)' );
    my $hostname = $fp_tag_ca_Host;

    print $log "----- get_templates\n\n";
    my $folder_name = $fp_tag_ca_TemplateFolder;
    my $templates = get_templates($log, $ua, $hostname,
$folder_name);

    my $id = get_template_by_name($templates,
$fp_tag_ca_TemplateName);
    my $outputfileextension = $templates-
>{$id}{OutputFileExtensions}[0];
    my $outputfilename = $dropfile;
    $outputfilename =~ s/(.+)\.[^.]+?$/\$1$outputfileextension/;

    print $log "----- submit_encoding_job\n\n";
    print $log "----- file: $dropdir/$dropfile\n\n";

#    my $job_id = submit_encoding_job($log, $ua, $hostname, $id,
"$dropdir/$dropfile", "$dropdir/transcoded/$dropfile");
    my $job_id = submit_encoding_job($log, $ua, $hostname, $id,
"$windropdir/$dropfile",
"$windropdir/transcoded/$outputfilename");

    my $out = '';
    my $retval = {};

    do {
    #    print "polling\n";
        sleep 1;
        print $log "----- get_wolkflowstatus\n\n";
        $retval = get_wolkflowstatus($log, $ua, $hostname,
$job_id);
        $out = $retval->{State};
    } while ( $out ne 'Completed' );

```

```

    print $log Dumper($retval);
    print $log "-----\n\n";
    close $log;
}

sub get_wolkflowstatus {
    my $log = shift;
    my $ua = shift;
    my $hostname = shift;
    my $jobid = shift;

    my $completionstate = '';
    my $state = '';
    my $errormessage = '';

    my $overallstate = {};

    my $xml = qq#
    <soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/">
        <soapenv:Header/>
        <soapenv:Body>
            <tem:JobAPI_GetWorkflowStatus>
                <tem:workflowjobID>$jobid</tem:workflowjobID>
            </tem:JobAPI_GetWorkflowStatus>
        </soapenv:Body>
    </soapenv:Envelope>
    #;

    my $req = HTTP::Request->new( 'POST' =>
"http://$hostname:8731/ContentCentral/ContentCentral_PublicJobAPIS
ervice/" );
    $req->header( 'SOAPAction'          =>
"http://tempuri.org/ContentCentral_PublicJobAPIService/JobAPI_GetW
orkflowStatus" );
    $req->header( 'Host'                 => "$hostname:8731" );
    $req->header( 'Proxy-Connection'    => 'Keep-Alive' );
    $req->header( 'Accept-Encoding'     => 'gzip,deflate' );
    $req->header( 'Content-Length'      => length $xml );
    $req->content_type( 'text/xml;charset=UTF-8' );
    $req->content($xml);

    my $res = $ua->request($req);
    my $wolkflowstatus_response = $res->as_string;

    print $log "-----\nget_wolkflowstatus\n";
#    print "$wolkflowstatus_response\n\n";

```

```

        if ($wolkflowstatus_response =~
m#<a:CompletionState>(.*?)</a:CompletionState>.*?<a:WorkflowNodeSt
ates>#s) {
            $completionstate = $1;
        } elsif ($wolkflowstatus_response =~
m#</a:WorkflowNodeStates>.*?<a:CompletionState>(.*?)</a:Completion
State>#s){
            $completionstate = $1;
        } else {
            print $log "no CompletionState found!\n";
            exit;
        }

        if ($wolkflowstatus_response =~
m#<a:State>(.*?)</a:State>.*?<a:WorkflowNodeStates>#s) {
            $state = $1;
        } elsif ($wolkflowstatus_response =~
m#</a:WorkflowNodeStates>.*?<a:State>(.*?)</a:State>#s){
            $state = $1;
        } else {
            print $log "no CompletionState found!\n";
            exit;
        }

        if ($wolkflowstatus_response =~
m#<a:ErrorMessage>(.*?)</a:ErrorMessage>.*?<a:WorkflowNodeStates>#
s) {
            $errormessage = $1;
        } elsif ($wolkflowstatus_response =~
m#</a:WorkflowNodeStates>.*?<a:ErrorMessage>(.*?)</a:ErrorMessage>
#s){
            $errormessage = $1;
        } else {
            print $log "no ErrorMessage found!\n";
        }

        $overallstate = {    CompletionState => $completionstate,
                            State           => $state,
                            ErrorMessage    => $errormessage,
                            };

        print $log Dumper($overallstate);

        return $overallstate;
    }

sub submit_encoding_job {
    my $log = shift;
    my $ua = shift;
    my $hostname = shift;
    my $template_id = shift;
    my $source = shift;
    my $destination = shift;

```

```

my $priority = 3;

my $job_id;
my $sourceFileMetadata = get_sourcefilemetadata;
my $xml = qq#
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/"
xmlns:root="http://schemas.datacontract.org/2004/07/Root6.Stingray
.ContentCentral">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:JobAPI_SubmitEncodingJob>
      <tem:sourceFilePath>$source</tem:sourceFilePath>

<tem:destinationFilePath>$destination</tem:destinationFilePath>
      <tem:templateID>$template_id</tem:templateID>
      $sourceFileMetadata
      <tem:priority>$priority</tem:priority>
    </tem:JobAPI_SubmitEncodingJob>
  </soapenv:Body>
</soapenv:Envelope>
#;

my $req = HTTP::Request->new( 'POST' =>
"http://$hostname:8731/ContentCentral/ContentCentral_PublicJobAPIS
ervice/" );
$req->header( 'SOAPAction'          =>
"http://tempuri.org/ContentCentral_PublicJobAPIService/JobAPI_Subm
itEncodingJob" );
$req->header( 'Host'                => "$hostname:8731" );
$req->header( 'Proxy-Connection'    => 'Keep-Alive' );
$req->header( 'Accept-Encoding'     => 'gzip,deflate' );
$req->header( 'Content-Length'      => length $xml );
$req->content_type( 'text/xml;charset=UTF-8' );
$req->content($xml);

my $res = $ua->request($req);

my $encoding_job_response = $res->as_string;

print $log "submitted xml:\n$xml\n\n";
print $log "$encoding_job_response\n";

if ($encoding_job_response =~
m!<JobAPI_SubmitEncodingJobResult>(.*?)</JobAPI_SubmitEncodingJobR
esult>!) {
  $job_id = $1;
} else {
  print $log "no JobID found!\n";
  exit;
}
print $log "$job_id\n";

```

```

    return $job_id;
}

sub get_templates {
    my $log = shift;
    my $ua = shift;
    my $hostname = shift;
    my $folder_name = shift;
    my $templates = {};

    my $xml = qq#
    <soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/">
    <soapenv:Header/>
    <soapenv:Body>
        <tem:Database_GetPlatinumTemplates>
            <tem:folderName>$folder_name</tem:folderName>
        </tem:Database_GetPlatinumTemplates>
    </soapenv:Body>
</soapenv:Envelope>
#;

    my $req = HTTP::Request->new( 'POST' =>
"http://$hostname:8731/ContentCentral/ContentCentral_PublicJobAPIS
ervice/" );
    $req->header( 'SOAPAction'          =>
"http://tempuri.org/ContentCentral_PublicJobAPIService/Database_Ge
tPlatinumTemplates" );
    $req->header( 'Host'                 => "$hostname:8731" );
    $req->header( 'Proxy-Connection'    => 'Keep-Alive' );
    $req->header( 'Accept-Encoding'     => 'gzip,deflate' );
    $req->header( 'Content-Length'      => length $xml );
    $req->content_type( 'text/xml;charset=UTF-8' );
    $req->content($xml);

    my $res = $ua->request($req);
    my $templates_response = $res->as_string;

    while ( $templates_response =~
m!<a:CA_StoreTemplate>(.*?)</a:CA_StoreTemplate>!sg ) {
        my $template_xml = $1;
        my $template_id;
        my $name;
        my $details = '';
        my $ofd = '';
        my $ofes = [];

        if ($template_xml =~
m!<a:TemplateID>(.*?)</a:TemplateID>! ) {
            $template_id = $1;
        } else {
            print $log "no TemplateID found!\n";
            next;

```

```

    }

    if ($template_xml =~ m!<a:Name>(.*?)</a:Name>! ) {
        $name = $1;
    } else {
        print $log "no Template Name found!\n";
        next;
    }

    $details = $1
        if $template_xml =~ m!<a:Details>(.*?)</a:Details>!;

    $ofd = $1
        if $template_xml =~
m!<a:OutputFileDescription>(.*?)</a:OutputFileDescription>!;

        if ($template_xml =~
m!<a:OutputFileExtensions[^>]+?>(.*?)</a:OutputFileExtensions>!s)
        {
            my $ofe = $1;
            while ( $ofe =~ m!<b:string>(.*?)</b:string>!sg ) {
                push @$ofes, $1;
            }
        }

        $templates->{$template_id} = {      Name => $name,
                                           Details =>
$details,

        OutputFileDescription => $ofd,

        OutputFileExtensions => $ofes,

                                           };
    }

    print $log Dumper($templates);

    return $templates;
}

sub get_template_by_name {
    my $templates = shift;
    my $name = shift;

    return (grep { $templates->{$_}{Name} eq $name } keys
%$templates)[0];
}

sub get_clip_info {
#   my $name_script = <<END;
#   tell application "AppleScript Runner"
#       try
#           display dialog "Clip Name:" default answer ""
buttons {"Ok"} default button 1

```



```

#             return (text returned of result)
#         end try
#     end tell
#END
#

my $desc_script = <<END;
tell application "AppleScript Runner"
    try
        display dialog "Clip description:" default answer
"" buttons {"Ok"} default button 1
        return (text returned of result)
    end try
end tell
END

#my $one_script = <<END;
#tell application "AppleScript Runner"
#    try
#        display dialog "Act on Files" buttons {"Ingest as
clips with original TC", "Create one file"}
#    end try
#end tell
#END
#

# my $clip_name;
# do {
#     $clip_name = `osascript -e '$name_script'`;
#     $clip_name =~ s/^\s+|\s+$//g;
#     if ($clip_name) {
#         print "Clip Name:\n$clip_name\n\n";
#     }
# } while (not $clip_name);
#

my $clip_desc;
do {
    $clip_desc = `osascript -e '$desc_script'`;
    $clip_desc =~ s/^\s+|\s+$//g;
    if ($clip_desc) {
        print "Clip description:\n$clip_desc\n\n";
    }
} while (not $clip_desc);

# return { clip_name => $clip_name, clip_desc => $clip_desc};
return $clip_desc;
}

sub get_sourcefilemetadata {
    my $groupname = 'XYZ Sport CMP';

    my $cmp_convert = {
        'Sports Production Unit' => {
            '5live'           => '5Live',

```

```

'Football'    => 'Football',
'Multimedia' => 'Multimedia',
'Olympics'    => 'Olympics',
'Online'      => 'Online',
'SportNews'   => 'Sport News',
'TVSPORT'     => 'TV Sport',
},
'Outlet' => {
'ATV'    => 'Athletics TV - ATV',
'ATP'    => 'ATP World Tour - ATP',
'AUO'    => 'Australian Open - AUO',
'BRK'    => 'Breakfast - BRK',
'BRM'    => 'British Masters - BRM',
'BOD'    => 'British Olympic Dreams - BOD',
'BUL'    => 'Bulletins - BUL',
'CHC'    => 'Challenge Cup - CHC',
'CHA'    => 'Championship - CHA',
'DAC'    => 'Davis Cup - DAC',
'EUR'    => 'Euro Championships - EUR',
'WC'     => 'FIFA World Cup - WC',
'FSC'    => 'Final Score - FSC',
'FF'     => 'Football Focus - FF',
'F1'     => 'Formula One - F1',
'FRO'    => 'French Open - FRO',
'GRN'    => 'Grand National - GRN',
'HAT'    => 'Hardtalk - HAT',
'HDL'    => 'Headlines - HDL',
'IF1'    => 'Inside F1 - IF1',
'MAL'    => 'Magners League - MAL',
'MD1'    => 'Match of the Day - MD1',
'MD2'    => 'Match of the Day 2 - MD2',
'MDL'    => 'Match of the Day Live - MDL',
'MGP'    => 'Moto GP - MGP',
'OYS'    => 'Olympic Sports - OYS',
'ONL'    => 'Online - ONL',
'PGA'    => 'PGA Championships - PGA',
'RB'     => 'Red Button - RB',
'ROA'    => 'Royal Ascot - ROA',
'RYC'    => 'Ryder Cup - RYC',
'SCO'    => 'Scottish Open - SCO',
'6NA'    => 'Six Nations - 6NA',
'SS'     => 'Ski Sunday - SS',
'STD'    => 'Sport Today - STD',
'SPOTY'  => 'Sports Personality - SPOTY',
'SPD'    => 'Sportsday - SPD',
'SPG'    => 'Sportsgathering - SPG',
'SOY'    => 'Summer Olympics - SOY',
'TMS'    => 'Test Match Special - TMS',
'TBR'    => 'The Boat Race - TBR',
'DER'    => 'The Derby - DER',
'FLS'    => 'The Football League Show - FLS',
'LCS'    => 'The League Cup Show - LCS',
'MAS'    => 'The Masters - MAS',
'OPE'    => 'The Open - OPE',

```

```

'UDF' => 'Undefined -UDF',
'WIM' => 'Wimbledon - WIM',
'WOY' => 'Winter Olympics - WOY',
'WBO' => "Women's British Open - WBO",
'WFF' => 'World Football Focus - WFF',
'SAY' => 'World Football Have Your Say - SAY',
'WMP' => 'World Match Play - WMP',
'WLD' => 'World News - WLD',
'WSN' => 'World Sports News - WSN',
'FO' => 'French Open - FO',
},
'Sport' => {
'AMF' => 'American Football - AMF',
'ANG' => 'Angling - ANG',
'ARC' => 'Archery - ARC',
'ATH' => 'Athletics - ATH',
'AUR' => 'Aussie Rules - AUR',
'BAD' => 'Badminton - BAD',
'BAS' => 'Baseball - BAS',
'BKB' => 'Basketball - BKB',
'BVB' => 'Beach Volleyball - BVB',
'BIL' => 'Billiards - BIL',
'BOW' => 'Bowls - BOW',
'BOX' => 'Boxing - BOX',
'CAN' => 'Canoeing - CAN',
'CWG' => 'Commonwealth Games - CWG',
'CRI' => 'Cricket - CRI',
'CUR' => 'Curling - CUR',
'CYC' => 'Cycling - CYC',
'DAR' => 'Darts - DAR',
'DIS' => 'Disabled Sports - DIS',
'DIV' => 'Diving - DIV',
'EQU' => 'Equestrian - EQU',
'FEN' => 'Fencing - FEN',
'FTB' => 'Football - FTB',
'GSP' => 'Gaelic Sports - GSP',
'GEN' => 'Generic Sports - GEN',
'GBL' => 'Goalball - GBL',
'GLF' => 'Golf - GLF',
'GRH' => 'Greyhound Racing - GRH',
'GYM' => 'Gymnastics - GYM',
'HBL' => 'Handball - HBL',
'HAG' => 'Hang Gliding - HAG',
'HOC' => 'Hockey - HOC',
'HOR' => 'Horse Racing - HOR',
'IHO' => 'Ice Hockey - IHO',
'ISK' => 'Ice Skating - ISK',
'JUD' => 'Judo - JUD',
'KAR' => 'Karate - KAR',
'LAC' => 'Lacrosse - LAC',
'MAA' => 'Martial Arts - MAA',
'MOP' => 'Modern Pentathlon - MOP',
'MOT' => 'Motor Racing - MOT',
'NET' => 'Netball - NET',

```

```

'NSP' => 'Non Sport - NSP',
'OLY' => 'Olympic Games - OLY',
'POL' => 'Polo - POL',
'PBR' => 'Power Boat Racing - PBR',
'RAF' => 'Rafting - RAF',
'ROW' => 'Rowing - ROW',
'RGL' => 'Rugby League - RGL',
'RGU' => 'Rugby Union - RGU',
'SAI' => 'Sailing - SAI',
'SHO' => 'Shooting - SHO',
'SKI' => 'Skiing - SKI',
'SLI' => 'Sliding - SLI',
'SNO' => 'Snooker - SNO',
'SQU' => 'Squash - SQU',
'SWI' => 'Swimming - SWI',
'TBT' => 'Table Tennis - TBT',
'TKW' => 'Taekwondo - TKW',
'TEN' => 'Tennis - TEN',
'TRM' => 'Trampolining - TRM',
'TRI' => 'Triathlon - TRI',
'UDF' => 'Undefined - UDF',
'VOL' => 'Volleyball - VOL',
'WAP' => 'Water Polo - WAP',
'WAS' => 'Water Skiing - WAS',
'WEI' => 'Weightlifting - WEI',
'WIS' => 'Wind Surfing - WIS',
'WIN' => 'Winter Sports - WIN',
'WRE' => 'Wrestling - WRE',
},

'Content Link' => {
  'PROGRAMME' => 'Programme PasB',
  'RUSHES'    => 'Rushes PasB',
},

'Media Source' => {
  '1'  => 'Baseband - 2nd Pres Area Studio',
  '2'  => 'Baseband - Line Feed HD',
  '3'  => 'Baseband - Line Feed SD',
  '4'  => 'Baseband - Live Feed HD',
  '5'  => 'Baseband - Live Feed SD',
  '6'  => 'Baseband - MCSL Studio',
  '7'  => 'Baseband- Sport News Studio',
  '8'  => 'Card - EX',
  '9'  => 'Card - P2',
  '10' => 'Card - SxS',
  '11' => 'Card - XF',
  '12' => 'Davina',
  '13' => 'Disk - Optical',
  '14' => 'Disk - XDCam',
  '15' => 'Edit Suite',
  '16' => 'Email',
  '17' => 'F1XFly',
  '18' => 'F1XSAN',
}

```

```

'19' => 'Fabric',
'20' => 'File - FTP',
'21' => 'File - IP Feed',
'22' => 'File - Web Stream',
'23' => 'Jupiter',
'24' => 'SPLib',
'25' => 'SPLib Comp',
'26' => 'Tape - BetaSP',
'27' => 'Tape - Digibeta',
'28' => 'Tape - DVCam',
'29' => 'Tape - DVCPro',
'30' => 'Tape - DVCProHD',
'31' => 'Tape - HDCam',
'32' => 'Tape - HDCamSR',
'33' => 'Tape - HVD',
'34' => 'Tape - IMX',
'35' => 'Tape - MiniDV',
'36' => 'USB - Stick or Drive',
'37' => 'XFile',
},

'Retention Recommendation' => {
  'archive' => 'Archive',
  'delete'  => 'Delete',
  'review'  => 'Review',
},

'Rights Traffic Light' => {
  'red'     => 'Red',
  'amber'   => 'Amber',
  'green'   => 'Green',
},

'Destination TV' => {
  '1'  => 'Salford Jupiter',
  '2'  => 'TVC Jupiter',
  '3'  => 'Millbank Jupiter',
  '4'  => 'Birmingham Quantel',
  '5'  => 'Newcastle Quantel',
  '6'  => 'Plymouth Quantel',
  '7'  => 'Leeds Quantel',
  '8'  => 'Nottingham Quantel',
  '9'  => 'Hull Quantel',
  '10' => 'Cambridge Quantel',
  '11' => 'Norwich Quantel',
  '12' => 'Oxford Quantel',
  '13' => 'Southampton Quantel',
  '14' => 'Tunbridge Quantel',
  '15' => 'West1 Quantel',
},

'Destination Radio' => {
  '1'  => 'XYZ Local Berkshire',
  '2'  => 'XYZ Local Bradford & West Yorkshire',

```

'3' => 'XYZ Local Bristol',
 '4' => 'XYZ Local Cambridgeshire',
 '5' => 'XYZ Local Cornwall',
 '6' => 'XYZ Local Cumbria',
 '7' => 'XYZ Local Derby',
 '8' => 'XYZ Local Devon',
 '9' => 'XYZ Local Dorset',
 '10' => 'XYZ Local Essex',
 '11' => 'XYZ Local Gloucestershire',
 '12' => 'XYZ Local Guernsey',
 '13' => 'XYZ Local Hampshire',
 '14' => 'XYZ Local Humber',
 '15' => 'XYZ Local Isle Of Man',
 '16' => 'XYZ Local Jersey',
 '17' => 'XYZ Local Kent',
 '18' => 'XYZ Local Lancashire',
 '19' => 'XYZ Local Leeds',
 '20' => 'XYZ Local Leicester',
 '21' => 'XYZ Local Lincoln',
 '22' => 'XYZ Local Lincolnshire',
 '23' => 'XYZ Local Liverpool',
 '24' => 'XYZ Local Manchester',
 '25' => 'XYZ Local Norfolk',
 '26' => 'XYZ Local Northamptonshire',
 '27' => 'XYZ Local North Yorkshire',
 '28' => 'XYZ Local Nottingham',
 '29' => 'XYZ Local Oxford',
 '30' => 'XYZ Local Somerset',
 '31' => 'XYZ Local Southern Counties',
 '32' => 'XYZ Local Suffolk',
 '33' => 'XYZ Local Tees',
 '34' => 'XYZ Local Three Counties',
 '35' => 'XYZ Local Tyne',
 '36' => 'XYZ Local Wear',
 '37' => 'XYZ Local Wiltshire',
 '38' => 'XYZ Oxford Radio Website',
 '39' => 'Bush Hse - WS Radio - Left +Right',
 '40' => 'Bush Hse - WS Radio - Left',
 '41' => 'Bush Hse - WS Radio - Right',
 '42' => 'Bush Hse - WS Radio - Split Track',
 '43' => 'JEX File Transmitter - UK audio-only',
 '44' => 'Radio Berkshire Website',
 '45' => 'Radio Bristol Website',
 '46' => 'Radio Cambridgeshire Website',
 '47' => 'Radio Cambridge Website',
 '48' => 'Radio Cornwall Website',
 '49' => 'Radio Coventry and Warwickshire Website',
 '50' => 'Radio Coventry & Warwickshire Website',
 '51' => 'Radio Cumbria Website',
 '52' => 'Radio Derby Website',
 '53' => 'Radio Devon Website',
 '54' => 'Radio Essex Website',
 '55' => 'Radio Gloucestershire Website',
 '56' => 'Radio Guernsey Website',

'57' => 'Radio Hereford and Worcester Website',
 '58' => 'Radio Hereford & Worcester Website',
 '59' => 'Radio Humberside Website',
 '60' => 'Radio Jersey Website',
 '61' => 'Radio Kent Website',
 '62' => 'Radio Lancashire Website',
 '63' => 'Radio Leeds Website',
 '64' => 'Radio Leicester Website',
 '65' => 'Radio Lincolnshire Website',
 '66' => 'Radioman Berkshire',
 '67' => 'Radioman Bristol',
 '68' => 'Radioman Cambridge',
 '69' => 'Radio Manchester Website',
 '70' => 'Radioman Cornwall',
 '71' => 'Radioman Coventry and Warwickshire',
 '72' => 'Radioman Coventry & Warwickshire',
 '73' => 'Radioman Cumbria',
 '74' => 'Radioman Derby',
 '75' => 'Radioman Essex',
 '76' => 'Radioman Exeter',
 '77' => 'Radioman Gloucester',
 '78' => 'Radioman Guernsey',
 '79' => 'Radioman Hereford and Worcester',
 '80' => 'Radioman Hereford & Worcester',
 '81' => 'Radioman Humberside',
 '82' => 'Radioman Jersey',
 '83' => 'Radioman Kent',
 '84' => 'Radioman Lancashire',
 '85' => 'Radioman Leeds',
 '86' => 'Radioman Leicester',
 '87' => 'Radioman Lincolnshire',
 '88' => 'Radioman London MHS',
 '89' => 'Radioman London - MHS',
 '90' => 'Radioman Manchester',
 '91' => 'Radioman Merseyside',
 '92' => 'Radioman Newcastle',
 '93' => 'Radioman Norfolk',
 '94' => 'Radioman Northampton',
 '95' => 'Radioman Norwich',
 '96' => 'Radioman Nottingham',
 '97' => 'Radioman Oxford',
 '98' => 'Radioman Plymouth',
 '99' => 'Radioman SCR Brighton',
 '100' => 'Radioman SCR Guildford',
 '101' => 'Radioman Sheffield',
 '102' => 'Radioman Shropshire',
 '103' => 'Radioman Solent',
 '104' => 'Radioman Somerset',
 '105' => 'Radioman Stoke',
 '106' => 'Radioman Suffolk',
 '107' => 'Radioman Surrey - Guildford',
 '108' => 'Radioman Sussex - Brighton',
 '109' => 'Radioman Tees',
 '110' => 'Radioman Test TVC',

```

'111' => 'Radioman Three Counties',
'112' => 'Radioman Wiltshire',
'113' => 'Radioman WM',
'114' => 'Radioman York',
'115' => 'Radio Merseyside Website',
'116' => 'Radio Newcastle Website',
'117' => 'Radio Norfolk Website',
'118' => 'Radio Northamptonshire Website',
'119' => 'Radio Nottingham Website',
'120' => 'Radio Sheffield Website',
'121' => 'Radio Shropshire Website',
'122' => 'Radio Solent Website',
'123' => 'Radio Somerset Sound Website',
'124' => 'Radio Southern Counties Website',
'125' => 'Radio Stoke Website',
'126' => 'Radio Suffolk Website',
'127' => 'Radio Tees Website',
'128' => 'Radio Three Counties Website',
'129' => 'Radio Wiltshire Website',
'130' => 'Radio WM Website',
'131' => 'Radio York Website',
'132' => 'TVC VCS - Left+Right .wav',
'133' => 'TVC VCS - Left .wav',
'134' => 'TVC VCS - Right .wav',
'135' => 'TVC VCS - Split Track .wav',
'136' => 'VCS - TV Centre, London into *News
Traffic',
      'none' => 'None',
    },
  },
};

my $sourceFileMetadata = qq|
<tem:sourceFileMetadata>
  <root:CA_MetadataEntry>
    <root:DisplayName>Sports Production Unit</root:DisplayName>
    <root:GroupName>$groupname</root:GroupName>
    <root:Value>$cmp_convert->{'Sports Production
Unit'}{$fp_tag_xyz_SportsProductionUnit}</root:Value>
  </root:CA_MetadataEntry>
  <root:CA_MetadataEntry>
    <root:DisplayName>Outlet</root:DisplayName>
    <root:GroupName>$groupname</root:GroupName>
    <root:Value>$cmp_convert-
>{'Outlet'}{$fp_tag_xyz_Outlet}</root:Value>
  </root:CA_MetadataEntry>
  <root:CA_MetadataEntry>
    <root:DisplayName>Content Producer</root:DisplayName>
    <root:GroupName>$groupname</root:GroupName>
    <root:Value>$fp_tag_xyz_ContentProducer</root:Value>
  </root:CA_MetadataEntry>
  <root:CA_MetadataEntry>
    <root:DisplayName>Sport</root:DisplayName>
    <root:GroupName>$groupname</root:GroupName>

```



```

    <root:Value>$cmp_convert-
>{'Sport'}{$fp_tag_xyz_Sport}</root:Value>
</root:CA_MetadataEntry>
<root:CA_MetadataEntry>
    <root:DisplayName>Programme Number</root:DisplayName>
    <root:GroupName>$groupname</root:GroupName>
    <root:Value>$fp_tag_xyz_ProgrammNumber</root:Value>
</root:CA_MetadataEntry>
<root:CA_MetadataEntry>
    <root:DisplayName>Content Link</root:DisplayName>
    <root:GroupName>$groupname</root:GroupName>
    <root:Value>$cmp_convert->{'Content
Link'}{$fp_tag_xyz_ContentLink}</root:Value>
</root:CA_MetadataEntry>
<root:CA_MetadataEntry>
    <root:DisplayName>Project/Item Name</root:DisplayName>
    <root:GroupName>$groupname</root:GroupName>
    <root:Value>$fp_tag_xyz_ProjectName</root:Value>
</root:CA_MetadataEntry>
<root:CA_MetadataEntry>
    <root:DisplayName>Clip Description</root:DisplayName>
    <root:GroupName>$groupname</root:GroupName>
    <root:Value>$fp_tag_xyz_ClipDescription</root:Value>
</root:CA_MetadataEntry>
<root:CA_MetadataEntry>
    <root:DisplayName>Media Source</root:DisplayName>
    <root:GroupName>$groupname</root:GroupName>
    <root:Value>$cmp_convert->{'Media
Source'}{$fp_tag_xyz_MediaSource}</root:Value>
</root:CA_MetadataEntry>
<root:CA_MetadataEntry>
    <root:DisplayName>Original Recording Date</root:DisplayName>
    <root:GroupName>$groupname</root:GroupName>
    <root:Value>$fp_tag_xyz_OriginalRecordingDate</root:Value>
</root:CA_MetadataEntry>
<root:CA_MetadataEntry>
    <root:DisplayName>Retention Notes</root:DisplayName>
    <root:GroupName>$groupname</root:GroupName>
    <root:Value>$fp_tag_xyz_RetentionNotes</root:Value>
</root:CA_MetadataEntry>
<root:CA_MetadataEntry>
    <root:DisplayName>Retention
Recommendation</root:DisplayName>
    <root:GroupName>$groupname</root:GroupName>
    <root:Value>$cmp_convert->{'Retention
Recommendation'}{$fp_tag_xyz_RetentionRecommendation}</root:Value>
</root:CA_MetadataEntry>
<root:CA_MetadataEntry>
    <root:DisplayName>Talent UID</root:DisplayName>
    <root:GroupName>$groupname</root:GroupName>
    <root:Value>$fp_tag_xyz_TalentUid</root:Value>
</root:CA_MetadataEntry>
<root:CA_MetadataEntry>
    <root:DisplayName>Rights UID</root:DisplayName>

```

```

        <root:GroupName>$groupname</root:GroupName>
        <root:Value>$fp_tag_xyz_RightsUid</root:Value>
</root:CA_MetadataEntry>
<root:CA_MetadataEntry>
    <root:DisplayName>Kill Date</root:DisplayName>
    <root:GroupName>$groupname</root:GroupName>
    <root:Value>$fp_tag_xyz_KillDate</root:Value>
</root:CA_MetadataEntry>
<root:CA_MetadataEntry>
    <root:DisplayName>Rights Traffic Light</root:DisplayName>
    <root:GroupName>$groupname</root:GroupName>
    <root:Value>$cmp_convert->{'Rights Traffic
Light'}{$fp_tag_xyz_RightsTrafficLight}</root:Value>
</root:CA_MetadataEntry>
<root:CA_MetadataEntry>
    <root:DisplayName>Jupiter Info</root:DisplayName>
    <root:GroupName>$groupname</root:GroupName>
    <root:Value>$fp_tag_xyz_JupiterInfo</root:Value>
</root:CA_MetadataEntry>
<root:CA_MetadataEntry>
    <root:DisplayName>Destination TV</root:DisplayName>
    <root:GroupName>$groupname</root:GroupName>
    <root:Value>$cmp_convert->{'Destination
TV'}{$fp_tag_xyz_DestinationTv}</root:Value>
</root:CA_MetadataEntry>
<root:CA_MetadataEntry>
    <root:DisplayName>Destination Radio</root:DisplayName>
    <root:GroupName>$groupname</root:GroupName>
    <root:Value>$cmp_convert->{'Destination
Radio'}{$fp_tag_xyz_RestinationRadio}</root:Value>
</root:CA_MetadataEntry>
<root:CA_MetadataEntry>
    <root:DisplayName>Requested By</root:DisplayName>
    <root:GroupName>$groupname</root:GroupName>
    <root:Value>$fp_tag_xyz_RequestedBy</root:Value>
</root:CA_MetadataEntry>
<root:CA_MetadataEntry>
    <root:DisplayName>Source ID</root:DisplayName>
    <root:GroupName>$groupname</root:GroupName>
    <root:Value>$fp_tag_xyz_SourceId</root:Value>
</root:CA_MetadataEntry>
</tem:sourceFileMetadata>
|;
    return $sourceFileMetadata;
}

```