



The Complete Single Source Imaging Solution

FSI Server Celum Bridge

CELUM ALWAYS AHEAD.

FSI Server Celum Bridge

Developed by:

NeptuneLabs GmbH
Lagesche Str. 32
D-32657 Lemgo
Germany

© 2010-2011 NeptuneLabs. All rights reserved.

Last updated: February 2011

FSI Server Version	2
Manual Revision	004

All brands and product names are trademarks or registered trademarks of the respective producers.

FSI Viewer is a registered trademark of NeptuneLabs GmbH, Germany.

NeptuneLabs FSI Server celum IMAGINE Integration

FSI Server Celum Bridge

1. Preface

FSI Server Celum Bridge integrates the single source imaging server FSI Server into the celum IMAGINE DAM system and is added as a plug-in to the celum IMAGINE installation.

Folders, categories and keywords in celum IMAGINE can be configured for synchronisation with the FSI Server (see chapter 4.3). All assigned images will be exported to the FSI Server and synchronized either periodically by a background task or on any user interaction like delete, rename etc.

The FSI Server Celum Bridge supports two different export modes:

- 1) FILE export
- 2) HTTP upload



Shows the web interface of celum IMAGINE with integrated FSI Server Celum Bridge

Exported images (2) can be viewed by all FSI Viewer components (FSI Viewer, FSI Pages, FSI Viewer 3D and FSI Showcase). The FSI Server Celum Bridge therefore integrates a preview component into the sidebar (3) of the celum IMAGINE web interface and also provides several popup windows (4) for more detailed views or as collections of images, e.g. in FSI Pages (5).



This document only refers to celum IMAGINE. However the FS1 Server Celum Bridge will also function for the other Celum products, e.g. celum SWIFT.

The FS1 Server Celum Bridge supports:

celum IMAGINE 4.7.0 and higher

Table of Contents

1.	Preface	4
2.	Installation.....	6
3.	Configuration	7
3.1	FSI Server Celum Bridge parameters	8
3.2	Libraries and classes	12
3.3	Logging	13
3.4	Menus and windows	14
3.5	Controller and JSP	15
3.6	JavaScript Popup Windows.....	16
3.7	Synchronisation task.....	18
3.8	Sidebar Panel in celum IMAGINE	20
4.	Best Practices.....	21
4.1	Active Menu	21
4.2	Code-Snippet	21
4.3	Synchronizing.....	22
4.4	FSI Viewer Presets	25
5.	Troubleshooting and known problems.....	26
5.1	Export paths and permissions.....	26
5.2	Network.....	26
5.3	Licences	27
5.4	Frequently overwriting of images.....	27
5.5	Synchronisation period too short	27
5.6	"trashbin not found" in Celum 4.7.x.....	28
5.7	Memory Usage	28
5.8	No FSI Viewer visible	29
5.9	FSI Viewer is shown without image.....	29
5.10	Updates and Backup.....	30
5.11	Blacklist - Handling failed exports	30
5.12	Changing the root path of celum IMAGINE	32
6.	How it works all together.....	33

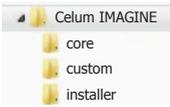
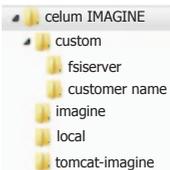
2. Installation

The FSI Server Celum Bridge is fully integrated into the celum IMAGINE installation and consists of JAVA class files, JSP (Java Server Pages), HTML etc and some configuration files. All files are placed in the plug-in folder 'fsiserver' which simply needs to be copied into the 'custom' folder of the celum IMAGINE installation.

- Copy the FSI Server Celum Bridge plug-in folder into the current celum IMAGINE installation (B) or the celum IMAGINE install folder (A).
- FSI Server Celum Bridge license: insert the FSI Server Celum Bridge license file 'licence.xml' into '*%Celum IMAGINE%/custom/fsiserver*'.
- Sidebar Panel: copy the file 'WEB-INF-plugins-imagine.ui-js-Imagine-gui-previewlegacy-PreviewPanel-xxxx.js' into the following folder and rename it to 'PreviewPanel.js':

```
%Celum IMAGINE%/imagine/WEB-INF/plugins/imagine.ui/js/Imagine/gui/previewlegacy/PreviewPanel.js
```

Plug-In: FSI Server Celum Bridge is distributed as a zip-archive including the folder 'fsiserver' which contains the plug-in with its configuration files.

<p>(A) Integrate FSI Server Celum Bridge into installation or update procedure of celum IMAGINE. The zipped package of celum IMAGINE contains only two directories: core and installer. If a plug-in like the FSI Server Celum Bridge should be integrated into the installation procedure, then add the folder 'custom' and copy the folder 'fsiserver' including the plug-in into it (see below). Please refer to the celum IMAGINE installation guide for further information (licence files, sql driver, Ghostscript and OpenOffice).</p>	
<p>(B) Add FSI Server Celum Bridge to an existing celum IMAGINE installation: the celum IMAGINE plug-in for the FSI Server integration should be placed in: <i>%Celum IMAGINE%/custom/</i> Please note: refer to 'trashbin not found' in 'Troubleshooting and known problems' (see 5.8).</p>	

3. Configuration

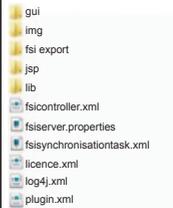
In general please take care of trailing spaces or other invisible characters at the end of your configuration lines. Some problems might occur with boolean variables in the case if there values aren't properly defined ('true' or 'false').

The file `%Celum IMAGINE%/custom/config.properties` is autogenerated during the installation of celum IMAGINE and contains a single line guiding to the customer configuration.

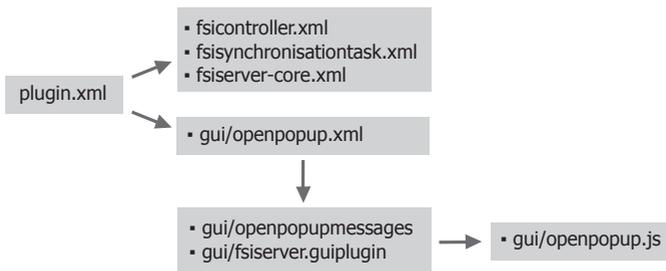
In the case that more than one plug-in is used, each plug-in has to be enabled by a statement like: `enabledCustomerPlugins=custom.%pluginname%`

```
customer = customer name
enabledCustomerPlugins=custom.fsiserver
```

All global configuration parameters and the celum licence file will be placed in the folder named as 'anycustomername' and the FSI Server Celum Bridge is placed in `%CelumIMAGINE%/custom/fsiserver` where its configuration and the licence file 'licence.xml' can be found.

<p>The configuration file 'config.properties' directs celum IMAGINE to look for a plug-in in the folder:</p>	
<p><code>%Celum IMAGINE%/custom/fsiserver</code></p>	

The configuration files are interdependent:



See the following configuration files for installation dependant parameters:

3.1 <i>.../custom/fsiserver/fsiserver.properties</i>	General parameters for the FSI Server Celum Bridge (export directory, FSI Server URL, ...). The filename is specified in fsiserver-core.xml
3.2 <i>.../custom/fsiserver/plugin.xml</i>	Integration specific settings like path to JAVA classes and paths to several configuration files for the FSI Server Celum Bridge components
3.3 <i>.../custom/fsiserver/log4j.xml</i>	Logging parameters for logging modes, logging files and directories

3.1 FSI Server Celum Bridge parameters fsiserver.properties

The file 'custom/fsiserver/fsiserver.properties' contains some general configuration parameters for the FSI Server Celum Bridge. You may change the value of all parameters except 'fsi.FSIViewerEnabled' while the FSI Server Celum Bridge is running.

```
fsi.server.url = http://host:port/fsi
```

This URL refers to the FSI Server and is used by all FSI Viewer components in celum IMAGINE

e.g.: <http://host:80/fsi/viewer/fsi.swf?ImagePath=fsibrige/inbox/image.jpg>

If you are using HTTPS then make sure that your server provides a valid SSL certificate. Browsers might not accept content from security enabled hosts with non valid certificates (see troubleshooting 5.10).

```
fsi.export.keepnewestfile = true
```

If this flag is set to true (default = true) only the newest file will be exported to the FSI export folder. No other image with an older modification date and with identical names can overwrite the newest images.

```
fsi.export.directory = /fsiserver/import
```

This property specifies the export directory, to where all images will be downloaded

by the FSI Server Celum Bridge, if the HTTP mode is not enabled (see below)

This directory is the same as the importer of the FSI Server uses for scanning and importing to the FSI Server.

```
fsi.export.http = true  
  
fsi.export.http.login = admin  
fsi.export.http.password = admin
```

The property 'fsi.export.http' switches between the two export modes (file system, HTTP upload). If this flag is set to 'true' all images will be transferred directly to the FSI Server without being copied to the local filesystem (parameter 'fsi.export.directory' will be ignored). This mode is disabled by default.

The FSI Server additionally needs a login which is specified by the parameter values of 'fsi.export.http.login' and 'fsi.export.http.password'.

The FSI Server address for the HTTP mode is the same as specified by 'fsi.server.url'

If HTTP mode is selected the FSI Server Celum Bridge optionally can use a proxy server. By default the proxy server port is 8080 and no login or password are used.

```
fsi.export.proxy.host      = proxy.server.host  
fsi.export.proxy.port     = 8080  
fsi.export.proxy.login    = proxy  
fsi.export.proxy.password = proxy
```

If HTTP mode is selected the FSI Server Celum Bridge is able to reduce the current used bandwidth for data transfer. By default there is no limitation configured. To modify the bandwidth limit you'll have to set the bandwidth parameter as float value. The FSI Server Celum Bridge will then be throttled by this value in Mbit/second.

```
fsi.openapi.bandwidth.mbps = 100
```

Example:

```
fsi.openapi.bandwidth.mbps = 0.1 → 100.000 bits/second → 12.5 kilobytes/second  
fsi.openapi.bandwidth.mbps = 1   → 1.000.000 bits/second → 125 kilobytes/second  
fsi.openapi.bandwidth.mbps = 10  → 10.000.000 bits/second → 1.25 megabytes/second
```

```
fsi.server.profile = fsibridge
```

This name is also used as profile name for the corresponding FSI Server importer profile. The value of this property 'fsi.server.profile' will be used as root directory for any FSI Server image URL like:

```
http://host:port/fsi/server?type=image&source=fsibridge/inbox/image.jpg
```

or

```
http://host:port/fsi/viewer/fsi.swf?ImagePath=fsibridge/inbox/image.jpg
```

See chapter 6. 'How it works' for more information

```
fsi.export.naming.assetid = false
```

If this value is set to true, each exported asset will be named by its celum IMAGINE id and not by its asset name.

The path will be the same in any mode. (Possible values are: false or true; default value is: false). The file suffix describing the asset file type will be appended to the current file name but depends on the following configuration parameter: 'fsi.export.naming.withoutfilesuffix'.

```
fsi.export.naming.withoutfilesuffix = false
```

If this value is set to true, each exported asset name will not be suffixed by its type. The path will be the same in any mode. (Possible values are: false or true; default value is: false).

Example for 'FSI-EXPORT/inbox/image.jpg':

<pre>fsi.export.naming.withoutfilesuffix = false fsi.export.naming.assetid = false</pre>	→	fsibridge/inbox/image.jpg
<pre>fsi.export.naming.withoutfilesuffix = true</pre>	→	fsibridge/inbox/image
<pre>fsi.export.naming.assetid = true</pre>	→	fsibridge/inbox/101.jpg

Parameters for the FSI Viewer components can be combined to presets. Presets which should be available for the FSI Server Celum Bridge need to be specified by their name in a comma separated list. The configuration is split into the folder

name where the presets are stored (inside the FSI Server installation folder) and the preset names itself. The FSI Server Celum Bridge supports presets for the FSI Viewer, FSI Viewer 360, FSI Pages and FSI Showcase.

The first entry in any list will be used as default preset (see also chapter 4.4)

```
fsi.viewer.preset.viewer.path = viewer_presets
fsi.viewer.preset.viewer.presets = no_nav,no_user_interface,minimal
fsi.viewer.preset.3d.path = viewer3d_presets
fsi.viewer.preset.3d.presets = 3d_spin_and_blur,3dthumbnail
fsi.viewer.preset.pages.path = pages_presets
fsi.viewer.preset.pages.presets = backgroundimage, catalog, catalog_links,
catalog_links_chapters, fancy_photo_album_with_text, fancythumbnail,
hard_cover_book,lightweight,photo,photo_text,thumbnail
fsi.viewer.preset.showcase.path = showcase_presets
fsi.viewer.preset.showcase.presets = default_floating_bottom, default_floating_top,
default_inside_bl, default_inside_br, default_inside_tl, default_inside_tr,
default_outside_bottom, default_outside_left, default_outside_right,default_outside_top,
showcase_3dfolders, showcase_info, showcase_text
```

```
fsi.fsiViewerEnabled = true
```

This flag enables the FSI Server Celum Bridge and the FSI Viewer for the sidebar panel in the celum IMAGINE web interface. If this flag is set to false the FSI Viewer components will only be shown by accessing the corresponding menu items.

3.2 Libraries and classes plugin.xml

The configuration file 'plugin.xml' references configuration files, JAVA classes and libraries:

```
<runtime>
  <library id="libs"
    path="relative:../custom/fsiserver/
    lib/*.jar" type="code" />
  <library id="classes" path="relative:../custom/fsiserver/
    classes/" type="code" />
</runtime>
```

The extensions of the FSI Server Celum Bridge are defined in the files 'gui/openpopup.xml', 'fsisynchronisationtask.xml' and 'controller.xml'.

```
<extension plugin-id="imagine.core" point-id="context"
  id="rootContextExtension">
  <parameter id="contextPath">
    <value>
      relative:../custom/fsiserver/gui/openpopup.xml
      relative:../custom/fsiserver/fsisynchronisationtask.xml
      relative:../custom/fsiserver/fsiserver-core.xml
    </value>
  </parameter>
</extension>

<extension plugin-id="imagine.ui"
  point-id="dispatcher.ims"
  id="defaultDispatcherExtension">
  <parameter id="contextPath">
    <value>
      relative:../custom/fsiserver/fsicontroller.xml
    </value>
  </parameter>
</extension>
```

"fsiserver-core.xml" defines the filename for the FSI Server Celum Bridge configuration file.

```
<bean id="fsiserverProperties" class="java.lang.String">
    <constructor-arg value="relative:../custom/fsiserver/
    fsiserver.properties" />
</bean>
```

3.3 Logging log4j.xml

The file 'log4j.xml' defines the mode in which the logger is active (TRACE, DEBUG, INFO, WARN, ERROR or FATAL), to where the log files are written, which date formats have to be used and also if log files should be compressed and archived on a daily basis.

```
<logger name="FSIServerPlugin">
    <level value="INFO" />
    <appender-ref ref="logfile" />
</logger>
```

The directory in which the file 'log4j.xml' can be found is specified in the main configuration file for the FSI Server Celum Bridge ('custom/fsiserver/fsiserver.properties') as property: 'fsi.logging.config.directory'.

default: custom/fsiserver/log4j.xml

The default output directory for the logger is defined by the '\${ims.root}' property which corresponds to the installation directory of the FSI Server Celum Bridge. The default path for all log files is defined as:

```
<param name="File" value="${ims.root}/../local/log
    FSIServerCelumBridge.log"/>
```

Please note that the format specifications for the log codes are placed in the following resource file.

custom/fsiserver/classes/com/neptunelabs/fsiserver/celum/resources/log4j_codes.properties

(For further information please see: <http://logging.apache.org/log4j>).

3.4 Menus and windows gui/openpopup.xml

The file 'custom/fsiserver/gui/openpopup.xml' contains some essential configuration parameters needed for the extension of the celum IMAGINE web interface with menu items and separate windows:

1. localized strings
2. configuration files for gui extensions
3. menu items

1. All used strings for the web interface are localized and defined in separate files for which the file prefix is defined as follows:

```
<bean id="fsiserverOpenpopupMessageSourceOverride"
      class="java.lang.String">
  <constructor-arg>
    <value>relative:../custom/fsiserver/gui
    openpopupmessages</value>
  </constructor-arg>
</bean>
```

The global path will be extended by a localized file suffix. Currently there are only files in german and english available:

'openpopupMessages_de.properties' and 'openpopupMessages_en.properties'

2. Any extension to the web interface needs further configuration and this is done in files which are specified in this section:

```
<bean id="fsiserverGuiPluginLocation" class="java.lang.
      String">
  <constructor-arg value="relative:../custom/gui/fsiserver/
  fsiserver.guiplugin"/>
</bean>
```

For a precise example please see the section below:
'popup windows - fsiserver.guiplugin and openpopup.js' (see 3.6).

3. Each menu item which is added to the context menu of any selected asset is specified in 'gui/openpopup.xml'.

```

<property name="fsiViewerMenuItem">
  <bean class="ims.web2.menu.MenuItem" autowire="no">
    <property name="id" value="fsiViewerMenuItem"/>
    <property name="name"
      value="openpopup.fsiViewerMenuItem"/>
    <property name="call"
      value="openpopup.openFsiViewer"/>
    <property name="callParams" value="{width:710,
      height:460}"/>
    <property name="icon" value="../resources/fsiserver/
      img/viewer_icon.png"/>
  </bean>
</property>

```

The menu item in this example allows the user to show the FSI Viewer in a separate window. The name of this menu item is 'fsiViewerMenuItem' and its string which is shown in the web interface is specified as 'openpopup.fsiViewerMenuItem' in the (localized) dictionary files like e.g. 'openpopupMessages_en.properties':

```
openpopup.fsiViewerMenuItem=Open FSI Viewer
```

By the property 'call' and its value 'WindowOpener.openFsiViewer' the selection of this menu item and the following action is configured by opening a window and sending a HTTP request to celum IMAGINE (configured in 'custom/fsiserver/js/WindowOpener.js' - see below).

3.5 Controller and JSP fsicontroller.xml

The file 'controller.xml' is related to the main plug-in 'imagine.ui' which controls the events coming from the celum IMAGINE web interface. The configured name for this controller bean ('fsiviewer.do') is added to the default path '/ims' and the current server address.

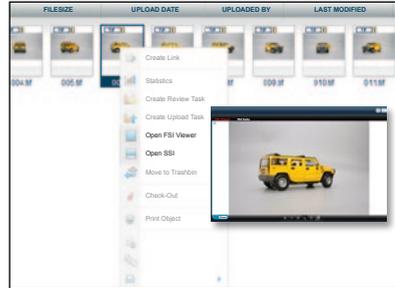
Example:

```

<bean id="fsiViewerController"
  name="/fsiviewer.do"
  class="com.neptunelabs.fsiserver.celum.controller.
  FSiViewerController"/>

```

In this case the bean 'fsiViewerController' is bound to a context menu action of an image. After having selected an image a context menu is available including the function 'Open with FSI Viewer'. After choosing this action the following server request is sent to celum IMAGINE (example):



<http://host:port/ims/fsiviewer.do?assetId=14878>

This request calls the request handler implemented in:

`com.neptunelabs.fsiserver.celum.controller.FSIViewerController`

The response of the request handler is redirected to a JSP file (JAVA Server Page) where the result is rendered into HTML and (in this case) a pop window shows the FSI Viewer with the image related to the asset id 14878.

All JSP files are placed in: `%Celum IMAGINE%/custom/fsiserver/jsp/`.

3.6 JavaScript Popup Windows fsiserver.guipugin and openpopup.js

The file 'fsiserver.guipugin' is used for adding additional resources to the Javascript parts of the celum IMAGINE web interface like e.g. CSS files. 'fsiserver.guipugin' references the file 'openpopup.js'.

The more important part for the FSI Server bridge is the definition of the file:

`%Celum IMAGINE%/custom/fsiserver/gui/openpopup.js`.

```
<bean id="fsiserver-internalMainViewIncludes"
      class="ims.web2.guipugin.DefaultGuiPlugin">
  <property name="javascriptFilesAsCommaSeparatedString">
    <value>
      relative:../custom/fsiserver/gui/openpopup.js
    </value>
  </property>
  ...
</bean>
```

Each of these calls opens a separate window in the web interface and displays the result of the controller beans HTTP request (defined in 'custom/fsiserver/fsicontroller.xml').

Example: When the user chooses a single asset by using the context menu in the web interface a call to a related 'openpopup' JavaScript object will be done. The 'openpopup' object makes a HTTP request to celum IMAGINE which calls the specified controller object from where the result is embedded in a JSP (JAVA Server Page) to show the corresponding FSI Viewer component.

In the example for controller.xml (see above) the related menu item sends a call with the name 'openFsiViewer'. The call function is defined in the file '/custom/fsiserver/gui/openpopup.js' as:

```
openpopup.openFsiViewer = function(config, assetDisplayItems)
{
    var url = "fsiviewer.do?" + getItemParameter(assetDisplay
        Items[0]);
    //Imagine.sdk.openWindow('fsiViewerWindow',
    {window:{height:config.height, width:config.width}, url:url});
    Imagine.sdk.openPopupWindow(url,
    {window:{height:config.height, width:config.width}});
}
```

After having opened the window, a HTTP Request is sent to celum IMAGINE and handled by the corresponding controller class. For example this request calls the request handler and passes an asset id as parameter:

<http://server:port/ims/fsiviewer.do?assetId=14878>

The call is handled by the corresponding controller (specified in fsicontroller.xml)

com.neptunelabs.fsiserver.celum.controller.FSiViewerController

The result of the controller object is redirected to the related JSP (JAVA Server Page) in e.g. *%Celum IMAGINE%/custom/fsiserver/jsp/fsiviewer.jsp*.

Please note: The file 'openpopup.js' provides two different calls where either popup windows or internal frames will be opened in celum IMAGINE. In each case only one of these entries is activated. By changing the comment tags '//' you can decide whether you prefer the external or internal popup windows.

Popup windows in general may be blocked by your web browser applications for security reasons. If you've configured popup windows for the FSI Server Celum Bridge the make sure to allow these for your browsers.

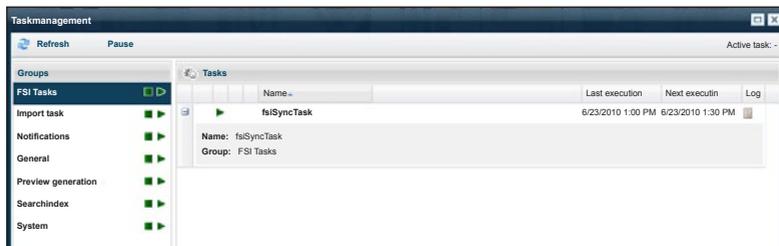
3.7 Synchronisation task fsisynchronisationtask.xml

The FSI Server Celum Bridge includes a synchronisation task which compares the current state of the celum IMAGINE system with the FSI Server.

All supported assets assigned to the FSI Server export folder or any of its sub-folders in celum IMAGINE will be exported to the FSI Server.

a) Missing assets will be downloaded from celum IMAGINE and copied to the FSI Server import directory.

b) Non-existing files in celum IMAGINE will be removed from the FSI Server import directory.



Each task in celum IMAGINE needs a specification with group, name and time schedule. According to this configuration the synchronisation task will be executed periodically.

%Celum IMAGINE%/custom/fsiserver/fsisynchronisationtask.xml

The file 'fsisynchronisationtask.xml' is related to the main plug-in 'imagine.core' and configures the synchronisation task execution interval by a cron-job syntax:

```
<bean id="fsiSyncTask"
      class="com.neptunelabs.fsiserver.celum.task.
        FSISyncTask"
      parent="abstractSdkTask"
      lazy-init="default"
      autowire="default"
      dependency-check="default">
  ....
  <property name="cronExpression"
    value="0 */30 * * * ?" />
  <property name="name"
    value="FSI Synchronisation Task" />
  <property name="group" value="FSI Tasks" />
</bean>
```

The bean 'fsiSyncTask' is named 'FSI Synchronisation Task' and added to the task group 'FSI Tasks' where the current state (running or sleeping) can be monitored in the administration console of the celum IMAGINE web interface (Administration System Tasks → FSI Tasks).

The synchronisation task compares the current state of the FSI Server with the assets stored by celum IMAGINE and copies or removes files on the FSI Server by need.

In this example the synchronisation task for all assets managed by celum IMAGINE with the same assets exported to the FSI Server is implemented in 'com.neptunelabs.fsiserver.celum.task.FSISyncTask' and identified by the id 'fsiSyncTask'. The task is executed every 30 minutes (regardless of how long it took to finish the previous synchronisation) beginning from the next full time period.

e.g. start of celum IMAGINE server at 12:23 → next synchronisation at 12:30

The next synchronisation will start 30 minutes later, after the previous task was started (at 13:00 o'clock in upper example). If the previous task is not finished then the current task will be skipped. In these cases the synchronisation period should be raised (e.g. use 1 minute per 1000 assets).

3.8 Sidebar Panel in celum IMAGINE PreviewPanel.js

The file PreviewPanel.js comes as:

WEB-INF/plugins-imagine.ui-js-Imagine-gui-previewlegacy-PreviewPanel.js

and needs to be copied into the following folder
(overwrite existing PreviewPanel.js):

*%Celum IMAGINE%/imagine/WEB-INF/plugins/imagine.ui/js/Imagine/gui/
previewlegacy/PreviewPanel.js*

The JavaScript file 'PreviewPanel.js' has a very special meaning to the FSI Server Celum Bridge and is essential for a full integration of the FSI Server and FSI Viewer capabilities into the celum IMAGINE system.

The file 'PreviewPanel.js' is used for the sidebar component of the celum IMAGINE web interface and manages the display of images with either the FSI Viewer flash components or as plain HTML image tags.

In 'PreviewPanel.js' an AJAX Request is executed each time a single image is selected in the celum IMAGINE web interface. As a result of the AJAX request the FSI Viewer may be shown in the sidebar if images are already exported and available on the FSI Server.



Please note: The import of images always starts after the scanner of the FSI Server has found new images in the export folder of the FSI Server Celum Bridge. The import can take some time, depending on how many images need to be imported. This is the reason why images might not be available for the FSI Viewer components immediately.

4.3 Synchronizing

It is possible to synchronize images assigned to folders, collections or keywords in celum IMAGINE export folder with the FSI Server export directory by using the 'Synchronize FSI Folder' function in the folders context menu on the left side or alternatively in the active menu on the top. The synchronisation is only available for separate folder structures and will be executed recursively with all of its subfolders. Only the FSI Sync Task will synchronize the whole FSI Export folder structure.

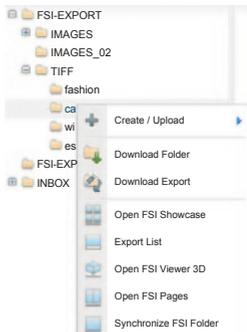
This function might be very time and resource consuming depending on the amount of subfolders and images and shouldn't be used too often.

The export and the synchronization of assets between celum IMAGINE and the FSI Server can be managed by using the context menu of any collection or folder and choosing the "Synchronize FSI Folder" or "Synchronize FSI Collection" option.

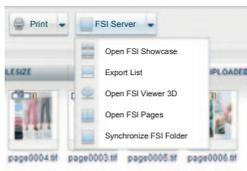
Adding, renaming and removing of export folders is only possible for administrators and privileged users.

The table gives an overview about the current state of each folder/collection with the following columns:

1. A symbol is shown for any synchronizable folder. By clicking on this symbol a synchronization of the assigned export folder will start immediately (except a full synchronisation is currently active)
2. The time the last synchronisation finished or was started by a user manually.
3. A remove button for any synchronized folder/collection. Using this button removes the current folder/collection from the export list. No synchronisation will be done any longer for this folder/collection.
4. An add button for any folder/collection which is currently not synchronized. Pressing this button adds that folder/collection to the export list and enables it for the synchronisation. By pressing the synchronisation symbol in column 1 a synchronisation of the current added folder will start immediately.
5. The ID of the folder/collection
6. The path of the folder/collection. By clicking this path a text input field will be displayed in the next column where the current path can be assigned to a different export path (this does not work for non synchronized folders/collections)



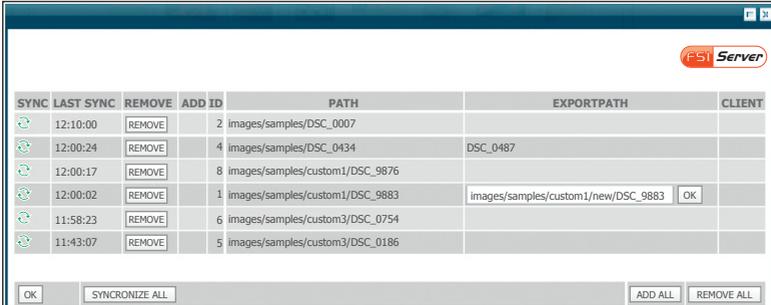
Context menu



Active menu

7. If the export path differs from the path in celum IMAGINE this column displays that path and the text input field for modifications (see above). By clicking this path a text input field is displayed in this column to enter a new path or delete the modified path.

8. The client name



"OK" Button

closes the window and refreshes the celum IMAGINE interface.

"SYNCHRONIZE ALL"

pressing this button synchronizes all folders/collections listed in this window which are enabled for synchronisation

"ADD ALL"

adds all folders/collections listed in this window to the synchronisation list.

"REMOVE ALL"

removes all folders/collections listed in this window to the synchronisation list.

The FSI Server Celum Bridge stores the synchronization configuration in the following files:

```
%CelumIMAGINE%/custom/fsiserver/fsiexport/fsiexport-folder-list.xml
%CelumIMAGINE%/custom/fsiserver/fsiexport/fsiexport-collection-list.xml
```

Any synchronized folder or collection is listed in one of these files and embedded in a <folder> tag:

```
<fsiexport>
  <folders>
    ...
    <folder>
      <id>35</id>
      <path>/celum/path/</path>
      <exportpath>/celum/export/path/</exportpath>
      <modified>14:50:11 15.02.2011</modified>
      <synchronized>17:21:00 15.02.2011</synchronized>
    </folder>
    ...
  </folders>
</fsiexport>
```

Although these files are automatically generated by the FSI Server Celum Bridge it is possible to modify or remove some of these tags by hand which will inevitably affect the current state of the FSI Server on the next synchronisation.

- Removing a <folder> tag will remove the corresponding folder from the FSI Server
- Adding or modifying a <exportpath> tag will remove the previous used folder on the FSI Server and cause a new export to the new folder on the FSI Server
- Removing a <exportpath> tag will remove the previously used export folder on the FSI Server and cause a export of all assigned documents to the folder named by <path>
- Modifying the <id> tag will exchange the current <folder> tag with a <folder> tag assigned to the new folder (if any exists) but will keep the already defined <exportpath> tag.

Modifying or removing the fsiexport list files might cause a loss of data !

If you need to use a backup of the fsiexport list files then copy them to the 'custom/fsiserver/fsiexport/' folder and wait until the FSI Server Celum Bridge detects and automatically reloads them. To make sure no to loose any data you also may stop Celum IMAGINE and restore the fsiexport list files after that.

Please also see Chapter 5.10 'Backup and Updates' and Chapter 5.11 'Black list – Handling failed exports' for further information.

4.4 FSI Viewer Presets

The FSI Viewer supports different parameters which can be combined in presets. The presets are configured in 'custom/fsiserver/fsiserver.properties' and will be available for the user in the select box in the upper right corner of each FSI Viewer panel.



5. Troubleshooting and known problems

5.1 Export paths and permissions

The FSI Server Celum Bridge needs read/write permissions to the export folder in the file system, which is specified by the parameter 'fsi.export.directory' in '*%Celum IMAGINE/custom/fsiserver/fsiserver.properties*'.

For windows:

```
fsi.export.directory = c:\\fsi\\export\\path
```

For Linux systems:

```
fsi.export.directory = /fsi/export/path
```

Alternatively you may specify the path as network path in UNC notation:

```
fsi.export.directory = \\server\fsi\export\path
```

Linux systems:

```
fsi.export.directory = //server/fsi/export/path
```

5.2 Network

The FSI Server Celum Bridge synchronizes the FSI Server with the celum IMAGINE DAM system. It realises this either by exporting directly to the file system or by transferring the images via HTTP protocol.

Network access is needed if the HTTP transfer mode is enabled or for the web interface components to find out whether an image is already imported by the FSI Server and viewable by any of the FSI Viewer components in celum IMAGINE.

To support these features the FSI Server Celum Bridge shouldn't be blocked by any firewall. The address of the FSI Server is defined in *%CelumIMAGINE/custom/fsiserver/fsiserver.properties*.

```
fsi.server.url = http://host:port/fsi/
```

5.3 Licences

The FSI Server Celum Bridge needs a licence file called "licence.xml" copied in to the plug-in folder `%CelumIMAGINE/custom/fsiserver/`.

5.4 Frequently overwriting of images

celum IMAGINE internally addresses all assets and folders by unique numeric ids. Asset or folder names must not be unique in celum IMAGINE. In contrast the FSI Server needs unique paths in combination with image names or unique image names in each folder.

This may lead to some problems if different images in celum IMAGINE have the same path/name or have the same name in the same folder and should be exported to the FSI Server. There is no rule of how to solve this problem and so all existing images with identical names in the same path will be overwritten. In combination with the synchronisation task this may happen again and again because the synchronisation will overwrite images if creation date or file size is different to the already stored image. The only way avoiding this issue is to keep all image paths unique or to avoid different images with same names in one single folder.

A situation like this may occur by using the Celum smart[upload application and adding already existing images to celum IMAGINE again. In this case celum IMAGINE will contain images with identical names but different modification/creation dates and/or file sizes.

To handle situations like this the configuration parameter:

```
fsi.export.keepnewestfile = true
```

was introduced (see 3.1). If this flag is set to true (default = true) only the newest file will be exported to the FSI export folder. No other image with an older modification date and with identical names can overwrite the newest images.

5.5 Synchronisation period too short

The synchronisation task is configured like a cron job and will be executed periodically. To avoid too heavy server loads or to avoid the problem that a synchronisation needs to be executed although another synchronisation task is still running the interval between each synchronisation task execution must be long enough. For the beginning you can estimate that time span as follows:

rule of thumb: 1 minute per 1000 assets

Any user can start a smaller synchronisation job for separate folders independent of the synchronisation task.

5.6 "trashbin not found" in Celum 4.7.x

celum IMAGINE refuses to start after having updated celum IMAGINE or having installed another FSI Server Celum Bridge as new version or backup.

Solution: celum IMAGINE v 4.7.x adds some entries to the configuration file during the startup phase in:

```
%Celum IMAGINE/custom/fsi-server/fsi-server.properties
```

Especially the entries related to the trashbin folder might cause problems if you copy a backup of the current FSI Server Celum Bridge plug-in into a new installation of celum IMAGINE where the trashbin folder either is not known or is known but has a different id. In these cases celum IMAGINE will refuse to start. To avoid this problem please remove the following two lines from 'fsi-server.properties'.

```
trashbin.folderId=1
gui.search.defaultCriterion={showFolders:[false],showLinks:
[false],showTextObjects[true],showContainerObjects:[true],
showOtherObjects:[true],excludedFolderIds:[1]}
```

Please note: In the case that you are updating the FSI Server Celum Bridge you may append these configuration parameters to the new "fsi-server.properties" file.

5.7 Memory Usage (celum IMAGINE inclusive Tomcat Server)

There might be the need for raising the memory used by the tomcat server running the celum IMAGINE system.

If you need to do this, then you should add a similar entry to your catalina.sh (or catalina.bat for windows):

```
JAVA_OPTS="-Xms1024m -Xmx2048m"
```

This example start parameter sets the initial memory to 1024MB and allows the Tomcat to use up to 2048MB for the celum IMAGINE servlet.

Place at the top (but not in the first line) of your "catalina.sh" or "catalina.bat" in:

```
%CelumIMAGINE%/tomcat_imagine%/bin/catalina.sh
```

5.8 No FSI Viewer visible

Some reasons why the FSI Viewer may not be visible in celum IMAGINE:

- The FSI Server needs to import and prepare all images. Depending on the amount and individual size of images this may take some time till a certain image is available for the FSI Viewer.
- If your FSI Server is configured for HTTPS like e.g `https://www.domain.com/fsi/` and its SSL certificate is not valid (probably expired or for non registered test systems) then your browser might deny content from the FSI Server with the consequence that no FSI Viewer will be shown while the user will not get any feedback (error dialog). The FSI Server Bridge itself is able to handle this situation and for all users this problem can be easily solved by manually calling the FSI Server URL e.g `https://www.domain.com/fsi/` and then accept the SSL certificate by hand (error dialog).
- Make sure that the parameter 'fsi.fsiViewerEnabled' is set to 'true'. Otherwise the sidebar panel of celum IMAGINE will ignore the FSI Viewer. (See 3.1 'FSI Server Celum Bridge parameters')
- To embed the FSI Viewer in the sidebar panel of celum IMAGINE the file `PreviewPanel.js` needs to be exchanged by an extended version which comes with this plug-in. (See 3.8 'Sidebar panel in celum IMAGINE')
- The extended sidebar panel for the FSI Server Celum Bridge makes use of an AJAX request to the FSI Server and FSI Server Celum Bridge and may get a timeout if the network connection is too slow. Additionally make sure that all HTTP requests to either celumIMAGINE with the FSI Server Celum Bridge and the FSI Server aren't blocked by any firewall.

5.9 FSI Viewer is shown without image

This may happen occasionally if your network connection is too slow and the background AJAX request gets an timeout error.

The FSI Server marks a image as known even if it is currently scanned and imported but not yet finished. In this case the FSI Viewer shows a default image or might be empty.

In other cases this may happen occasionally if your network connection is too slow and the background AJAX request gets an timeout error.

5.10 Updates and Backup

By default any install archive of the FSI Server Celum Bridge will not contain any licence or export list configuration file and will only overwrite the file "fsiserver.properties".

If you don't want to back up the complete folder of the FSI Server Celum Bridge or if you're updating the bridge and need to keep the current configurations then make copies of the following files

```
%CelumIMAGINE%/custom/fsiserver/licence.xml
%CelumIMAGINE%/custom/fsiserver/fsiserver.properties
```

and

```
%CelumIMAGINE%/custom/fsiserver/fsiexport/fsiexport-folder-list.xml
%CelumIMAGINE%/custom/fsiserver/fsiexport/fsiexport-collection-list.xml
```

After having updated the bridge you can easily restore the last known state by replacing the default files with your copies.

To avoid loss of data on the FSI Server it is strongly recommended to check the existence of all files listed above before you restart Celum IMAGINE. Missing entries in fsiexport-folder-list.xml or fsiexport-collection-list.xml may cause a loss of data on the FSI Server when the FSI Server Celum Bridge executes the next synchronisation. Folders which are not contained in these export list files will be removed from the FSI Server. Keeping, modifying or removing the file 'black-list.xml' is not critical (see 5.11).

5.11 Blacklist - Handling failed exports

The file black-list.xml contains information about all failed export transactions between Celum IMAGINE and the FSI Server. For example a typical error occurs when a document type is not known to the FSI Server. In this case the FSI Server will reject the upload with an „Unknown Media Type“ error. To avoid multiple uploads of the same already rejected document any failed transaction will be added to the black-list in:

```
%CelumIMAGINE%/custom/fsiserver/fsiexport/black-list.xml
```

You may remove the black list completely or edit it by hand if you want to enable a next try. If you want to remove single entries then you should delete these elements from the list which are enclosed by <document> tags like:

```

<fsiexport>
  <blacklist>
    ....
    <document>
      <id>100</id>
      <clientid>2</clientid>
      <path>/client2/path/to/image.jpg</path>
      <info>[HTTP] [415] Unsupported media type</info>
      <modified>14:51:36 15.02.2011</modified>
    </document>
    ....
  </blacklist>
</fsiexport>

```

In some cases problems might occur during a file transfer (e.g. lost server connection). The black list also contains document elements for these failed transactions including an additional expiry timestamp. After 30 min (minimum by default) the document will be removed from the black list and can be transferred again (depending on the next synchronization schedule).

In contrast to the fsiexport list files in the same folder you may remove the blacklist.xml without causing any loss of data on the FSI Server.

5.12 Changing the root path of celum IMAGINE

By default Celum IMAGINE is installed and accessible as:

http://server/ims/login.do

If you want to change the installation path you'll have to apply the new path to 3 occurrences in 2 files of the FSI Server Celum Bridge too:

e.g. change the root path from „ims“ to „newrootpath/ims“ → http://server/newrootpath/ims/login.do

1) line ~63 in %celumIMAGINE%/custom/fsiserver/jsp/fsisync.jsp:

```
uri = '/ims/fsisync.do?<%=folderCollectionSwitchParameter%>=' +  
folderId + '&syncAction=' + syncAction;
```

change to:

```
uri = '/newrootpath/ims/fsisync.do?<%=folderCollectionSwitchParamet  
er%>=' + folderId + '&syncAction=' + syncAction;
```

2) line ~272 in %celumIMAGINE%/custom/fsiserver/jsp/fsisync.jsp:

```
out.println("<input class=\"action\" type=button value=\"OK\"  
onClick=\"top.location.href='/ims/main.do';\">");
```

change to:

```
out.println("<input class=\"action\" type=button value=\"OK\"  
onClick=\"top.location.href='/newrootpath/ims/main.do';\">");
```

3) line ~635 in %celumIMAGINE%\image\WEB-INF\plugins\image.ui\js\Imagine\gui\previewlegacy\PreviewPanel.js

The AJAX request using this url determines if either a simple image or the FSI Viewer has to be shown in the preview panel.

```
url : '/ims/fsipreview.do',
```

change to:

```
url : '/newrootpath/ims/fsipreview.do',
```

6. How it works all together

celum IMAGINE / FSI Server Celum Bridge / FSI Server

This chapter shows the interdependence of several parameters for the FSI Server and the FSI Server Celum Bridge if you are using the file export mode. For the HTTP mode the FSI Server Celum Bridge only needs the FSI Server URL, profile name and login data as configuration while the FSI Server itself will be installed remote on a different host and its configuration will be completely independent.

The FSI Server Celum Bridge exports and synchronizes images from celum IMAGINE to the import folder of the FSI Server. This means that some configuration parameters of the FSI Server Celum Bridge must meet the configuration of the FSI Server importer unit which is defined in a different place.

FSI Server Celum Bridge	FSI Server
The parameters for the FSI Server Celum Bridge can all be found in ' <i>%Celum IMAGINE%/custom/fsiserver/fsiserver.properties</i> '. (see 3.1)	The parameters for the FSI Server can be found in <i>%FSIServer%/configuration/fsiserver/importer_profiles/</i> <i>(%TOMCATINSTALLDIR%/webapps/fsi/WEB-INF/importer_profiles/</i> if installed as servlet/war-archive)
fsi.server.profile = fsibrige	The filename 'fsibrige.xml' itself names the profile as 'fsibrige' (without file suffix) <i>...../importer_profiles/fsibrige.xml</i>
fsi.export.directory = /fsiserver/import	<Location>/fsiserver/import</Location>

FSI Server Celum Bridge	FSI Server
	<pre> <?xml version="1.0" encoding="UTF-8"?> <ImportProfile> <Enabled>true</Enabled> <Origin> <Accessor>filesystem </Accessor> <Location>/fsiserver/import </Location> </Origin> <Conversion> <Format>JPEG </Format> <Quality>98</Quality> </Conversion> </ImportProfile> </pre>

The table shows which parameters should meet and where to find them.

Example: the FSI Server Celum Bridge synchronizes the celum IMAGINE folder 'FSI-EXPORT' with the export folder in the filesystem: '/fsiserver/import'.

```

fsi.server.url = http://host:80/fsi
fsi.server.profile = fsibrige
fsi.export.directory = /fsiserver/import

```

System	Path
celum IMAGINE	/inbox/image.jpg
FSI Server Celum Bridge FSI Server importer	/fsiserver/import/inbox/image.jpg
FSI Server	fsibrige/inbox/image.jpg

The example file image.jpg will be accessible with the FSI Viewer by following URL:

<http://host:80/fsi/viewer/fsi.swf?ImagePath=fsibrige/inbox/image.jpg>.

Index

Active Menu.....	20
Code-Snippet	20
Configuration	6
JavaScript Popup Windows	15
Controller and JSP.....	14
Export paths.....	25
FSI Server Celum Bridge parameters	7
Installation	5
Libraries and classes	11
Licences	26
Logging.....	12
Memory Usage.....	27
Menus and windows.....	13
Network	25
No FSI Viewer for HTTPS connections	28, 29
Permissions	25
Sidebar panel	19
Synchronisation	26
Synchronisation period	26
Synchronisation task	17
Synchronising	21, 24
Trashbin	27
Troubleshooting	25