CADI user manual

(version 3.1)

GICI group

Department of Information and Communications Engineering

Universitat Autònoma Barcelona

http://www.gici.uab.es - http://www.gici.uab.es/CADI

June 5, 2012

Contents

1 Overview

CADI is an implementation of the JPEG2000 Interactive Protocol (Part 9) standard. Since the JPIP protocol is a client-server protocol, CADI is composed of four applications: CADIServer (that implements a JPIP server), CADIViewer (implmenting a Graphical User Interface for the JPIP client side), CADIClient (implementing a JPIP client), and CADIProxy (that implments a JPIP proxy). All applications are encapsulated in jar files in the *dist/* directory and can be executed separately using the JVM.

The implementation provides the basic functionalities of the JPIP protocol, Basic Communications Profile, and several capabilities of the Enhanded Communication Profile.

The source code of CADI has been designed and programmed with the aim to provide a flexible framework, allowing users to easily integrate their modules to test and develop new ideas inside the JPIP protocol in a easy way.

All applications, except for CADIViewer, provide different functionalities that can be controlled via their command line parameters, they are passed as program arguments. Each one of these parameters and their valid arguments will be detailed in the section devoted to each of the four applications.

A development manual is also available and the API is well documented to facilitate an easy understanding, extension and modification of the application. All the design and implementation details are widely explained in these manuals and the source code is commented. All these information is publicly available in the web page http://www.gici.uab.es/CADI, where you can download the sources and the manuals. To guarantee a free distribution, CADI has the General Public License (GNU GPL) of the Free Software Foundation (http://www.fsf.org/licensing/licenses/gpl.html).

We have tried to make a good design and efforts have been made to develop and implement the application as useful as desirable. However, we could not foresee all the needs of CADI users, so we will thank you for all the suggestions and comments that you can report to us (gici-dev@deic.uab.es).

We hope you enjoy it,

GICI group

2 CADIServer

The CADIServer application implements the server side of the JPIP protocol, allowing JPIP clients to download JPEG2000 images.
CADIServer does not have a graphical interface and it must be launched from the command line. Since the *CADIServer.jar* is in the *dist/* directory, the following command `java -jar dist/CADIServer.jar` will launch the server using the default options (see section 2.1 for detailed options).

2.1 CADIServer parameters

In this section all supported parameters for the server command line are explained. For a detailed description of each element in the table displayed for each parameter, please see 5.2.

−−**ports**	{int[int [int [...]]]}		
−**p**	*Mandatory:* No	*Max reps:* 1	
Explanation:	Ports where server will listen to the client request.		
Default:	80		

−−**numThreads**	{int}	
−**nt**	*Mandatory:* No	*Max reps:* 1
Explanation:	Number of threads that will be launched to process the request.	
Default:	1	

−−**targetsPath**	{string}	
−**tp**	*Mandatory:* No	*Max reps:* 1
Explanation:	Root directory where the targets (image files) are stored. Default value is the directory where the server is launched.	
Default:		

−−**logFile**	{string}	
−**lf**	*Mandatory:* No	*Max reps:* 1
Explanation:	File where logs are saved.	
Default:		

−−**logXML**	{boolean}	
−**lx**	*Mandatory:* No	*Max reps:* 1
Explanation:	XML format is used in the log file. Value is a boolean: 0 indicates simple file format is used and 1 indicates XML format is used.	
Default:	0	

−−**logEnabled**	{boolean}	
−**le**	*Mandatory:* No	*Max reps:* 1
Explanation:	Enables or disables the log. See the ”-ll” parameter for more information about the detail level of logs.	
Default:		

−−**logLevel**	{int}	
−**ll**	*Mandatory:* No	*Max reps:* 1
Explanation:	Is the severity of the messages which will be logged. The ”-le” parameter is set automatically. Available values are: 2- logs informative messages 3- logs warning messages 4- logs error messages when a log level is set, all upper levels are automatically set but lower severity messages are filtered.	
Default:	2	

−−**cacheDirectory**	{string}	
−**cd**	*Mandatory:* No	*Max reps:* 1
Explanation:	Directory used as a temporal directory to save the cache data (not implemented yet).	
Default:		

−−**deliveringMode**	{int}		
−**dm**	*Mandatory:* No	*Max reps:* 1	
Explanation:	Indicates the rate distortion method which is used in a JPEG2000 image to calculate de Window Of Interest. Available values are: 1- precinct data are delivered layer by layer just as they appear in the codestream 2- the CPI method is uses to calculate the WOI. It only can be used with request that belongs to a session. 3- the CoRD method is uses to calculate the WOI. 4- applies a window scaling factor to each precinct to be delivered consisting on the overlap factor of each precinct with the window of interest requested. OBS: Option 3 has been temporary disabled.		
Default:	4		

−−**cpiType**	{int}	
−**cpit**	*Mandatory:* No	*Max reps:* 1
Explanation:	Indicates, within the CPI rate-distortion method, the subtype which will be used to delivery the requested WOI. This parameter must be only available when the rate-distortion (-rd) parameter is 2. Otherwise, it won't be taken into account. Available values are: 1- Only one packet for each precinct is generated. Therefore, delivered image has only one quality layer. 2- Creates one packet per each bit plane. Therefore, delivered image has as quality layers as number of bit planes. 3- Creates one packet per each coding pass. Therefore, delivered image has as quality layers as number of coding passes. 4- Images are delivered following the SCALE method. This parameter is only allowd if the "-dm" parameter is 2.	
Default:	0	

−−**cordType**	{int}	
−**cordt**	*Mandatory:* No	*Max reps:* 1
Explanation:	Indicates, within the CPI rate-distortion method, the subtype which will be used to delivery the requested WOI. This parameter must be only available when the rate-distortion (-rd) parameter is 2. Otherwise, it won't be taken into account. Available values are: 1- Is the classic CoRD algorithm. 2- Is a modification of the CoRD algorithm in order to reduce the overhead of packet and JPIP message headers. This parameter is only allowd if the "-dm" parameter is 3.	
Default:	0	

−−**deliveryProgressionOrder**	{int}	
−**dpo**	*Mandatory:* No	*Max reps:* 1
Explanation:	Indicates the progression order which will be used to delivery data. If this parameter is not set, the progression order of the codestream will be used. Otherwise, the progression order of the codestream will not be taken into account. Available values are: 0- Layer-Resolution-Component-Position (LRCP). 1- Resolution-Layer-Component-Position (RLCP). 2- Resolution-Position-Component-Layer (RPCL). 3- Position-Component-Resolution-Layer (PCRL). 4- Component-Position-Resolution-Layer (CPRL). This parameter is only allowed if the "-dm" parameter is 1.	
Default:		

−−**keepAliveTiemout**	{int}	
−**kt**	*Mandatory:* No	*Max reps:* 1
Explanation:	Specifies the timeout (in milliseconds) that the socket is kept opened and waiting for new client requests.	
Default:	1	

−−**maxRate**	{int}		
−**mr**	*Mandatory:* No	*Max reps:* 1	
Explanation:	Specifies the maximum rate (bytes per second) which will be used to delivery data (0 means unlimited).		
Default:	0		

−−**trafficShaping**	{int}		
−**ts**	*Mandatory:* No	*Max reps:* 1	
Explanation:	Allows to choose a trafic shaping algorithm. Allowed values are: 0- None algoritm is applied. 1- The token-bucket algoritm is applied. Data are transmitted at the constant rate fixed by the "-mr" parameter rate, but it also allows data busts. 2- The leaky-bucket algoritm is applied. Data are delivered at the constant rate defined in the "-mr" option.OBS: This parameter requires the "-mr" parameter.		
Default:	0		

−−**independentMessageHeaders**	{booelan}		
−**imh**	*Mandatory:* No	*Max reps:* 1	
Explanation:	Indicates which form of JPIP messsage headers will be used (independent or dependent). The independent form is a long form where the message headers are completely self-describing. Meanwhile, the depenent form message headers make use of information in the headers of previous messages. The dependent form makes a shorter message header than the independent form, but sequence ordering of received message must be taken into account.		
Default:	1		

−−**predictiveModel**	{boolean}		
−**pm**	*Mandatory:* No	*Max reps:* 1	
Explanation:	This parameter indicates that a predictive model can be applied in the image delivering. The predictive model to be applied is read from a text file whose name is the same as the compressed image but with the extension "pm". The file must have a line for each spatial region (precinct) with the following format "precinct_id value", where the precinct_id is the unique precinct identifier defined in the JPIP protocol and the value is a real number in the range [0, 1] with the relevance of the relevance of the precinct. The remainder precincts not included in the file will be considered with relevance 0. And if the beginning-of-line character is an #, it is considered a comment and ignored.This parameter is only allowed if the "-dm" parameter is 4.		
Default:	0		

−−**help**			
−**h**	*Mandatory:* No	*Max reps:* 1	
Explanation:	Displays this help and exits program.		
Default:			

−−**warranty**			
−**w**	*Mandatory:* No	*Max reps:* 1	
Explanation:			
Default:			

−−**liability**			
−**l**	*Mandatory:* No	*Max reps:* 1	
Explanation:			
Default:			

−−**copyright**			
−**c**	*Mandatory:* No	*Max reps:* 1	
Explanation:			
Default:			

2.2 Examples

Some basic examples of CADIServer usage and combination of the different options of the command line parameters are shown in this section.

- Launch the server with the default options. Server is listening to in the port 80, the current directory is the path where images must be placed, it launches as threads as number of processors, and it uses the Window Scaling Factor as strategy for delivering images.

```
$java -jar dist/CADIServer.jar
```

- Display the help information.

```
$java -jar dist/CADIServer.jar -h
```

- Launch the server listening to on por 2080.

```
$java -jar dist/CADIServer.jar -p 2080
```

- Change the root path from where images (targets) are red to the *workDir* directory.

```
$java -jar dist/CADIServer.jar -tp workDir
```

- Change the rate-distortion algorithm used to deliver images.

```
$java -jar dist/CADIServer.jar -rd 2
```

- Adjust the maximum rate images are delivering (10KB) and set the the token-backut as traffic shaping algorithm.

```
$java -jar dist/CADIServer.jar -ts 1 -mr 10000
```

3 CADIClient

The CADI client application implements the client side of the JPIP protocol. It is a command line application to get images from a JPIP server and save them in a file.
Session cannot be used!.

3.1 CADIClient parameters

In this section all supported parameters for the server command line are explained. For a detailed description of each element in the table displayed for each parameter, please see 5.2.
CADIClient version 3.1

−−**server**	{string}		
−**s**	*Mandatory:* No	*Max reps:* 1	
Explanation:	Server name where the image to be retrieved is.		
Default:	localhost		

−−**port**	{int}		
−**p**	*Mandatory:* No	*Max reps:* 1	
Explanation:	Port number where the server is listening to client requests.		
Default:	80		

−−**proxyServer**	{string}		
−**ps**	*Mandatory:* No	*Max reps:* 1	
Explanation:	Server name where the image to be retrieved is.		
Default:			

−−**proxyPort**	{int}		
−**pp**	*Mandatory:* No	*Max reps:* 1	
Explanation:	Port number where the server is listening to client requests.		
Default:	8080		

−−**target**	{string}		
−**t**	*Mandatory:* No	*Max reps:* 1	
Explanation:	Targets to be retrieved from the server. If this parameter is not set, the uri (”-u”) parameter must be appear.		
Default:			

−−**uri**	{string}		
−**u**	*Mandatory:* No	*Max reps:* 1	
Explanation:	Is an Uniform Resource Identifier of the target to be retrieved. If this parameter is not set, the target (”-t”) parameter must be appear.		
Default:			

−−**outputImage**	{string}		
−**o**	*Mandatory:* Yes	*Max reps:* ∞	
Explanation:	Output image file name where recovered image samples will be stored. Image file type will be decided depending on the extension. Valid file types are: RAW/IMG raw data (if outputImage is raw data, ”-og” argument is mandatory) PNM/PGM/PPM PGM is used when image have 1 component, PPM when there are 3 components and PNM for both TIFF JPEG (not recommended because it degenerates recovered image) BMP (not recommended) ATTENTION: if recovered image samples are signed only RAW/IMG data with a signed data type will preserve image samples consistently. Using other file format will cause image data damages.		
Default:			

−−**ouputImageGeometry**	{int int}		
−**og**	*Mandatory:* No	*Max reps:* 1	
Explanation:	Image raw data type. Parameters are: 1- data type. Possible values are: 0- boolean (1 byte) 1- unsigned int (1 byte) 2- unsigned int (2 bytes) 3- signed int (2 bytes) 4- signed int (4 bytes) 5- signed int (8 bytes) 6- float (4 bytes) 7- double (8 bytes) 2- Byte order (0 if BIG ENDIAN, 1 if LITTLE ENDIAN)		
Default:			

−−**compnents**	{int[int [int [...]]]}	
−**cs**	*Mandatory:* No	*Max reps:* 1
Explanation:	Components of the image to be retrieved. This parameters is not compatible with the components ranges (”-cr”) parameter	
Default:		

−−**componentRanges**	{int-int [int-int [int-int [...]]]}	
−**cr**	*Mandatory:* No	*Max reps:* 1
Explanation:	Ranges of components. This parameters is not compatible with the components (”-cr”) parameter. ATTENTION: This option is not implemented yet!	
Default:		

−−**resolutionLevel**	{int}	
−**rl**	*Mandatory:* No	*Max reps:* 1
Explanation:	Is the resolution level at which the image is retrieved. This parameter is not compatible with frame size (”-fs”) parameter.	
Default:		

−−**frameSize**	{int int}	
−**fs**	*Mandatory:* No	*Max reps:* 1
Explanation:	Is the resolution level associate with the Window of Interest. The first value is the width, and the second one is the height. This parameter is not compatible with resolution level (”-rl”) parameter.	
Default:		

−−**regionOffset**	{int int}	
−**ro**	*Mandatory:* No	*Max reps:* 1
Explanation:	Is the top-left coordinates of the Window of Interest. The first value is the left coordinate, and the second on is the top coordinate.	
Default:		

−−**regionSize**	{int int}	
−**rs**	*Mandatory:* No	*Max reps:* 1
Explanation:	Is the size of the Window of Interest. The first value is the width, and the second one is the height.	
Default:		

−−**layers**	{int}	
−**ly**	*Mandatory:* No	*Max reps:* 1
Explanation:	Specifies the maximum number of layers which will retrieved for the target.	
Default:	0	

−−**quality**	{int}	
−**q**	*Mandatory:* No	*Max reps:* 1
Explanation:	Specifies the maximum quality which will retrieved for the target. ATTENTION: This option is not implemented yet!	
Default:	100	

−−**roundDirection**	{int}		
−**rd**	*Mandatory:* No	*Max reps:* 1	
Explanation:	Specifies how the image resolution shall be selected. Allowed values: 1- round-down 2- round-up 3- closest		
Default:	1		

−−**maxTargetLength**	{int}	
−**ml**	*Mandatory:* No	*Max reps:* 1
Explanation:	Specifies the maximum amount of bytes the client wants the server to send in response to the client request.	
Default:	0	

−−**help**		
−**h**	*Mandatory:* No	*Max reps:* 1
Explanation:	Displays this help and exits program.	
Default:		

−−**warranty**		
−**w**	*Mandatory:* No	*Max reps:* 1
Explanation:		
Default:		

−−**liability**		
−**l**	*Mandatory:* No	*Max reps:* 1
Explanation:		
Default:		

−−**copyright**		
−**c**	*Mandatory:* No	*Max reps:* 1
Explanation:		
Default:		

3.2 Examples

- Display the help information.

```
$java -jar dist/CADIClient.jar -h
```

- Request the image "image.jpc" from a JPIP server running on the local machine on port 80. The downloaded image is saved in the file "output_image.ppm".

```
$java -jar dist/CADIClient.jar -t image.jpc -o output_image.ppm
```

- Request the image "image.jpc" from a JPIP server running on the machine "jpipserver.domain" on port 9000. The downloaded image is saved in the file "output_image.ppm".

```
$java -jar dist/CADIClient.jar -t image.jpc -s jpipserver.domain -p 9000
-o output_image.ppm
```

- Request the image "image.jpc" from a JPIP server running on the machine "jpipserver.domain" on port 9000 through a JPIP proxy that is running on the machine "jpipproxy.domain" on port 8080.

```
$java -jar dist/CADIClient.jar -t image.jpc -s jpipserver.domain -p 9000
-ps jpipproxy.domain -pp 8080 -o output_image.ppm
```

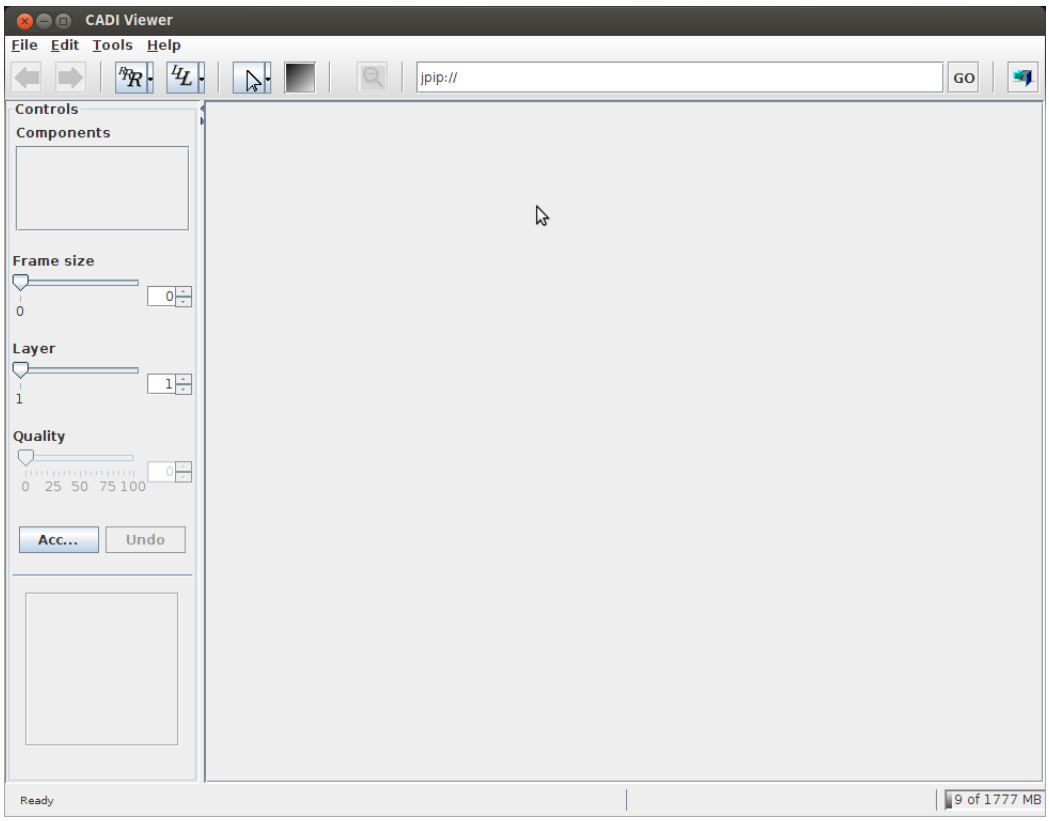
- Request a window/region of the image. The client requests for 10 layers of a region of the image ”image.jpc” defined by the frame size of 5000×4000 pixels located at position 1000×1000 and size 1024×768.

```
$java -jar dist/CADIClient.jar -t image.jpc -fs 5000 4000 -ro 1000 1000
-rs 1024 768 -ly 10 -o output_image.ppm
```

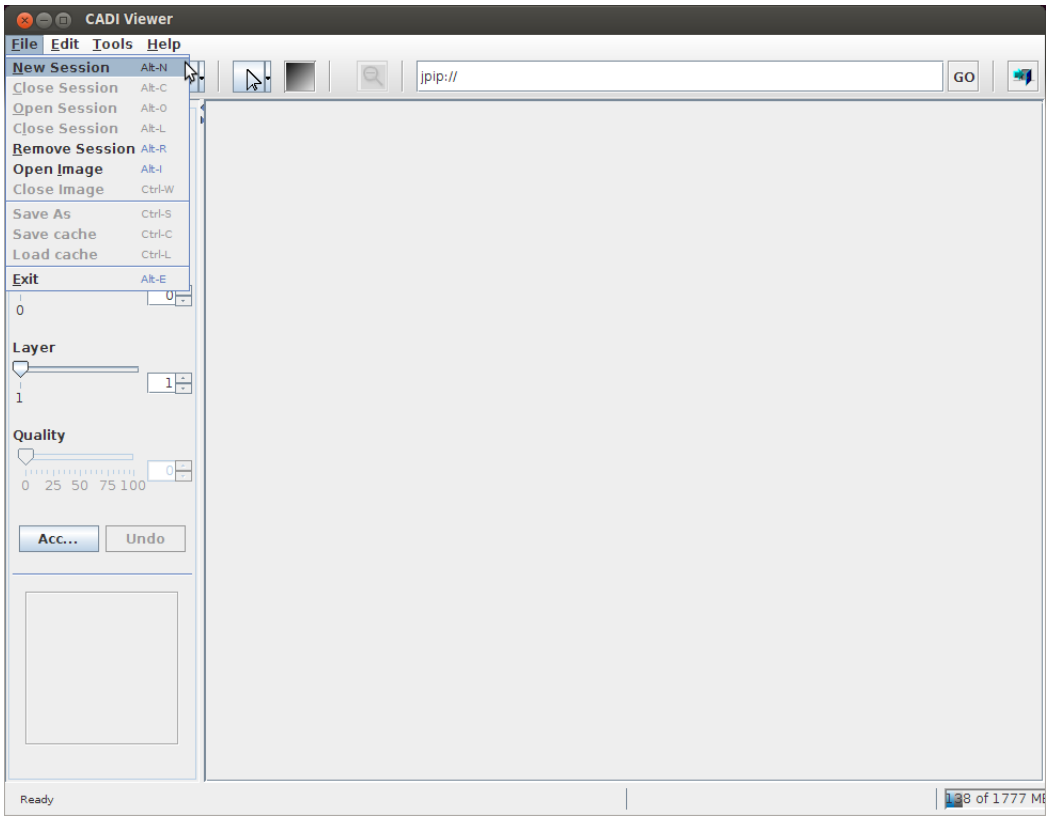
4 CADIViewer

The CADIViewer application is a Graphical User Interface of the CADI client.

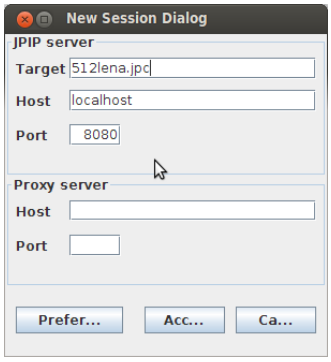
Once the CADIViewer has been launched (`java -jar dist/CADIViewer.jar`) next window will be displayed.



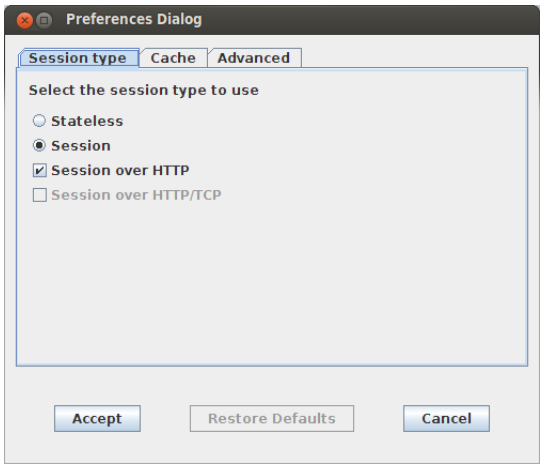
Next step is to open a new session through the menu *File* > *New Session* or by means of the shortcut *Alt+N*. It will open a New Session Dialog window.



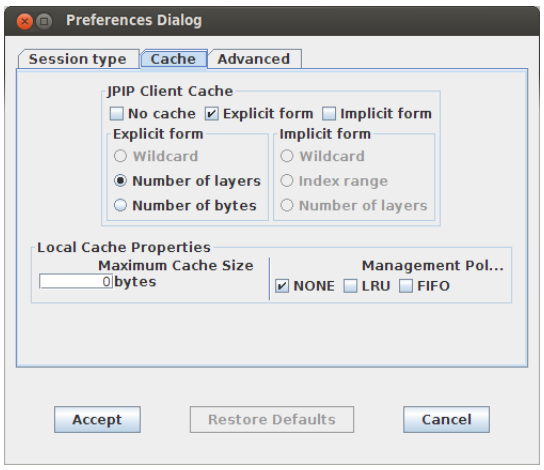
In the *New Session Dialog* window there are two sections: JPIP server and Proxy server. Only parameters in the JPIP server are mandatory. Then, the *Target* is the JPEG2000 image to be requested, the *Host* is the server where the JPIP server is running on, and the *Port* is the port on where the JPIP server is listening to. Regarding the Proxy server section, the *Host* is the server name where the JPIP proxy is running on, and the *Port* is the port of the JPIP proxy. Moreover, advanced preferences can be configured if click *Preferences* button. button



The *Preferences dialog window* is opened if the *Preference* boton has been clicked. In this dialog window prefences are grouped in three tabs. *Session type* allows to configure the connection to the JPIP server using Statless or Stateful sessions (only Sessions over HTTP are available). Stateful connections are recommended because they reduce the amount of data to be sent to the server.

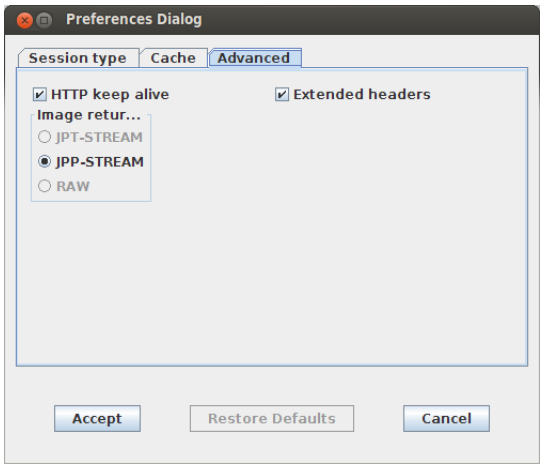


The tab *Cache* in the *Preferences dialog window* allows to configure the type of cache to be used in the client or not to use cache. Each type of cache can be qualified to be more precise in the type to use.

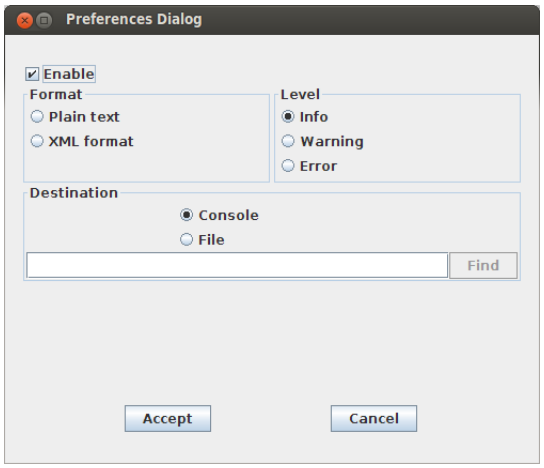


Regarding the tab *Advanced* in the *Preferences dialog window* allows to set detailed options of the JPIP protocol. It is

not recommened to modify this options unless you are sure about their meaning.



The menu *Edit* > *Preferences* opens the *Preferences* dialog window to configure the logs of the application. Logs can be enabled or disables in the *Enable* check. If enabled, it is mandatory to choose the format, the level of detail, and the destination (console or file).



5 CADIProxy

The CADIProxy applications implements a JPIP proxy. It is a command line application that not only caches all images server through the proxy but it can do prefetching.

5.1 CADIProxy parameters

−−**ports**	{int[int [int [...]]]}	
−**p**	*Mandatory:* No	*Max reps:* 1
Explanation:	Ports where server will be listening to the client requests.	
Default:	8080	

−−**numThreads**	{int}	
−**nt**	*Mandatory:* No	*Max reps:* 1
Explanation:	Number of threads that will be launched to process client requests.	
Default:	1	

−−**type**	{int}	
−**t**	*Mandatory:* No	*Max reps:* 1
Explanation:	Indicates the type of proxy that will be used. Allowed values are: 1- transparent proxy 2- cached proxy. All data transmitted from servers to clients are cached. 3- cache proxy with prefetching. It is an extension of the ”cached proxy” adding the capability of prefetching. OBS: Option 3 can be qualified by ”-pwt” or ”-pm” parameters. If none of them is set, deafult options to be considered are: ”-pdh=3”, ”-pwt=1”, and ”-mp=0.1” , respectively	
Default:	3	

−−**prefetchingDataHistory**	{int}	
−**pdh**	*Mandatory:* No	*Max reps:* 1
Explanation:	Is the data, previous windows of interest requested, used to build the actual WOI to be prefetched. Allowed values: 1- uses the history of the windows of interest requested by all clients over an image. View windows are sorted following a FIFO strategy for all clients. 2- prediction of windows of interest to be downloaded are done for each client taken into account only the historic wois for that client. 3- computes the prefetching woi using the lastest window of interest requested by all clients over each image. OBS: This option can only be set when the –type parameter is 3 (cached proxy with prefetching).	
Default:	3	

−−**prefetchingWOIType**	{int}	
−**pwt**	*Mandatory:* No	*Max reps:* 1
Explanation:	Allows to choose how the Windows of Interest to be prefetching is built from the historic of data. Allowed values: 1- prediction of the window of interest to be prefetched is based on weighted wois of the historic of windows of interest. Probability of movements can be modified by means of the ”-mp” parameter. 2- the window of interest to be prefected is the bounding box of the historic window of interest. OBS: This parameter is only allowed if the ”–type” parameter is 3 (cached proxy with prefetching). OBS: This parameter is not compatible with the ”-pm” parameter.	
Default:	1	

−−**movementProbabilities**	{float float float float float float float float float float}
−**mp**	*Mandatory:* No *Max reps:* 1
Explanation:	Probabilities of the movements to be used by the prefetching. Values must be sorted according with the following criterion: right, upright, up, upleft, left, downleft, down, downright, zoom in, zoom out. OBS: The sum of all values must be less or equal than 1. OBS: This option can only be set when the –type parameter is 3 (cached proxy with prefetching) and ”-pwt” is 1.
Default:	0.1

−−**predictiveModel**	{string}
−**pm**	*Mandatory:* No *Max reps:* 1
Explanation:	This parameter indicates that a predictive model, semantic information, is applied in the image delivering and prefetching. Predictive model to be applied is read from a text file located at path given by this parameter and whose name is the same as the compressed image but with the extension ”pm”. The file must have a line for each spatial region (precinct) with the following format ”precinct_id value”, where the precinct_id is the unique precinct identifier defined in the JPIP protocol and the value is a real number in the range [0, 1] with the relevance of the precinct. The remainder precincts not included in the file will be considered with relevance 0. And if the beginning-of-line character is an #, it is considered a comment and ignored. OBS: This parameter is only allowed if the ”–type” parameter is 3 (cached proxy with prefetching). OBS: This parameter is not compatible with the ”-pwt” parameter. OBS: If there is not a semantic file associated with an image, prefetching is done considering the option 1 of the -pwt parameter.
Default:	

−−**logFile**	{string}
−**lf**	*Mandatory:* No *Max reps:* 1
Explanation:	File where logs are saved.
Default:	

−−**logXML**	{boolean}
−**lx**	*Mandatory:* No *Max reps:* 1
Explanation:	XML format is used in the log file. Value is a boolean: 0 indicates simple file format is used and 1 indicates XML format is used.
Default:	0

−−**logEnabled**	{boolean}
−**le**	*Mandatory:* No *Max reps:* 1
Explanation:	Enables or disables the log. See the ”-ll” parameter for more information about the detail level of logs.
Default:	

−−**logLevel**	{int}
−**ll**	*Mandatory:* No *Max reps:* 1
Explanation:	Is the severity of the messages which will be logged. The ”-le” parameter is set automatically. Available values are: 2- logs informative messages 3- logs warning messages 4- logs error messages when a log level is set, all upper levels are automatically set but lower severity messages are filtered.
Default:	2

−−**cacheDirectory**	{string}
−**cd**	*Mandatory:* No *Max reps:* 1
Explanation:	Directory used as a temporal directory to save the cache data (not implemented yet).
Default:	

−−**maxRate**	{int}	
−**mr**	*Mandatory:* No	*Max reps:* 1
Explanation:	Specifies the maximum rate (bytes per second) which will be used to delivery data (0 means unlimited).	
Default:	0	

−−**trafficShaping**	{int}	
−**ts**	*Mandatory:* No	*Max reps:* 1
Explanation:	Allows to choose a trafic shaping algorithm. Allowed values are: 0- None algoritm is applied. 1- The token-bucket algoritm is applied. Data are transmitted at the constant rate fixed by the "-mr" parameter rate, but it also allows data busts. 2- The leaky-bucket algoritm is applied. Data are delivered at the constant rate defined in the "-mr" option.OBS: This parameter requires the "-mr" parameter.	
Default:	0	

−−**help**		
−**h**	*Mandatory:* No	*Max reps:* 1
Explanation:	Displays this help and exits program.	
Default:		

−−**warranty**		
−**w**	*Mandatory:* No	*Max reps:* 1
Explanation:		
Default:		

−−**liability**		
−**l**	*Mandatory:* No	*Max reps:* 1
Explanation:		
Default:		

−−**copyright**		
−**c**	*Mandatory:* No	*Max reps:* 1
Explanation:		
Default:		

5.2 Examples

In this section all supported parameters for the server command line are explained. For a detailed description of each element in the table displayed for each parameter, please see 5.2.

- Launch the JPIP proxy with the defualt options. Thus, the proxy is running on port 8080, it uses as number of threads as number of processors, and it is configure to work on prefetching mode with all the movements equally likely.

```
$java -jar dist/CADIProxy.jar -h
```

- Display the help information.

```
$java -jar dist/CADIProxy.jar -h
```

- Proxy working as a transparent proxy.

```
$java -jar dist/CADIProxy.jar -t 1
```

- Proxy working in *cache mode*.

```
$java -jar dist/CADIProxy.jar -t 2
```

- Proxy in prefethcing mode. Proxy is configured to do prefetching using as historic the lastest Window of Interest requested by all clients over each image and predicting the region to the image to be requested as a weigthing of the historic. Probabilities used to compute the next potential region are the same for all the movements (0.1).

```
$java -jar dist/CADIProxy.jar -t 3 -pdh 3 -pwt 1
```

- This example is the same as the previous one, but in this case the probabilities used to compute the next potential region of the image to be requested have been set to 0.075 for the panning movements and 0.2 for the zooms.

```
$java -jar dist/CADIProxy.jar -t 3 -pdh 3 -pwt 1 -mp 0.075 0.075 0.075
0.075.075 0.075 0.075 0.075 0.2 0.2
```

- Proxy is configure to work as a prefetching proxy using the predictive model qualifier to compute the next potential region to be requested. Files containing the semantic information about images are in the *pred_models* directory.

```
$java -jar dist/CADIProxy.jar -t 3 -pdh 3 -pm pred_models/
```

Annex: Parameters description

Parameters have two formats: the long and the short specification. Long specification has −− at the beginning while short specification has − (it does not matter which one you choose). Each parameter has its own arguments, which usually are integers, floats, booleans (0 to indicate false and 1 to indicate true) or strings. If the user specifies some invalid arguments, the application will display warning messages. Most of these parameters are not mandatory. When they are not specified default values are used. The following table shows how each parameter will be displayed in this manual:

−−**longParameter**	{parameter arguments}
−**shortParameter**	*Mandatory:* Yes/No
Explanation:	Parameter explanation
Default:	Parameter default values.

Annex: Requirements

Libraries

CADISoftware only needs the JRE version 1.6 or higher to run.

Compilation

Compile the CADISoftware is easy because there is a *build.xml* file ready to compile the project. The binaries of the complied project will be in the *dist/* directory. It must be noted that the Java Advanced Imaging (JAI) is necessary to compile the source code.

JVM parameters

The amount of memory depends on the CADI Software application. Thus, the memory requirements of CADIServer are low and they will depend on the number of images being served because it only works with a indexed image. However, in the CADIClient or CADIViewer the amount of memory depends on the dimensions of the image region to be decoded, the larger is the region the more memory is needed. And regarding the CADIProxy, it gathers both requirements of CADIServer and CADIClient.
In order to avoid a Java Heap Memory error when CADISoftware is run, it is recommended to set the maximum amount of memory that the application can allocate via the *-Xmx* parameter of the JVM, i.e. *java -Xmx1024m -jar dist/CADIServer.jar*.