

# CADI user manual

(version beta)

GICI group

Department of Information and Communications Engineering

Universitat Autònoma Barcelona

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

September 2007

## 1 Overview

CADI is a novel implementation of the JPEG2000 Interactive Protocol (Part 9) standard. Although there are other applications that incorporate the JPEG2000 standard, CADI is designed and programmed with the aim to provide a flexible framework to test and develop new ideas inside the JPIP protocol in a easy way. The application provides different functionalities that can be controlled via its parameters. These parameters can be passed as program arguments, selecting some of the algorithms that the application incorporates. Each one of these parameters and its valid arguments are explained in the application help and in this manual.

The main goal in this development is to provide a framework to test new procedures to serve and receive images. The design allows an easy incorporation of new algorithms and support for other compressed images. The GICI group has already incorporated new algorithms based on its research and encourages other researchers to use CADI to test their own ideas.

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@abra.uab.es](mailto:gici-dev@abra.uab.es)).

We hope you enjoy it,

GICI group

## 2 Usage

CADI software is divided in 2 applications: the server and the client. Both programs are encapsulated in jar files in the *dist/* directory and can be executed separately using the JVM. Each application accepts its own parameters which can be passed as simple command arguments, i.e. *java -jar dist/CADIServer.jar -p 80*. The distribution incorporates a shell script both for the server and the client called *CADIServer* and *CADIClient* to facilitate executions in a GNU/Linux environment. CADI may use a great amount of memory. Hence, it is recommended to set the maximum amount of memory that the application can allocate (usually the same as your computer RAM) via the *-Xmx* parameter of the JVM, i.e. *java -Xmx512m -jar dist/CADIServer.jar*.

The usage of CADI is very simple. To serve an image you can use the server with some specific parameters. All the functionalities of CADI are explained with the application parameters, so it is recommended to read them to know what you can do. Do not worry if you select incompatible parameters or functionalities; the application detects these problems and displays warning messages to the user.

When you have a compressed codestream hosted in the server, you can use the client to retrieve and display it. The retrieved image can be saved with other common formats.

CADIServer can manage codestreams compliant with JPEG2000 Standard generated by other applications. However not all the options defined in the Standard are implemented.

A typical use of the application is as follows:

```
# CADIServer
```

```
# CADIClient
```

or if you do not use the shell scripts:

```
# java -Xmx512m -jar dist/CADIServer
```

```
# java -Xmx512m -jar dist/CADIClient
```

### 3 Parameters

In this section all supported parameters for the server are explained. The CADIVIEWER has a graphical interface and all parameters are configured through menus. This documentation is extracted from the application help and it explains all the algorithms incorporated to CADI. By reading them you will know all the functionalities of the application. 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:

<b>--longParameter</b>	{parameter arguments}
<b>-shortParameter</b>	<i>Mandatory:</i> Yes/No
<i>Explanation:</i>	Parameter explanation
<i>Default:</i>	Parameter default values.

#### 3.1 CADIServer parameters

CADI Server beta 0.3

<b>--ports</b>	{int[int [int [...]]]}
<b>-p</b>	<i>Mandatory:</i> No   <i>Max reps:</i> 1
<i>Explanation:</i>	Ports where server will listen to the client request.
<i>Default:</i>	80

<b>--numThreads</b>	{int}
<b>-nt</b>	<i>Mandatory:</i> No   <i>Max reps:</i> 1
<i>Explanation:</i>	Number of threads that will be launched to process the request.
<i>Default:</i>	1

<b>--targetsPath</b>	{String}
<b>-tp</b>	<i>Mandatory:</i> No   <i>Max reps:</i> 1
<i>Explanation:</i>	Root directory where the targets (image files) are stored. Default value is the directory where the server is launched.
<i>Default:</i>	

<b>--logFile</b>	{string}
<b>-lf</b>	<i>Mandatory:</i> No   <i>Max reps:</i> 1
<i>Explanation:</i>	File where logs are saved.
<i>Default:</i>	

<b>--logXML</b>	{boolean}
<b>-lx</b>	<i>Mandatory:</i> No   <i>Max reps:</i> 1
<i>Explanation:</i>	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.
<i>Default:</i>	0

<b>--logEnabled</b>	
<b>-le</b>	<i>Mandatory:</i> No   <i>Max reps:</i> 1
<i>Explanation:</i>	Enables or disables the log.
<i>Default:</i>	

<b>--cacheDirectory</b>	{string}
<b>-cd</b>	<i>Mandatory:</i> No   <i>Max reps:</i> 1
<i>Explanation:</i>	Directory used as a temporal directory to save the cache data (not implemented yet).
<i>Default:</i>	

<b>--rateDistortion</b>	{int}
<b>-rd</b>	<i>Mandatory:</i> No   <i>Max reps:</i> 1
<i>Explanation:</i>	Indicates the rate distortion method which is used in a JPEG2000 image to calculate de Window Of Interest. Available values are: 1- data bins are sendes following the file layer ordering 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 (not implemented yet).
<i>Default:</i>	1

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

<b>--help</b>	
<b>-h</b>	<i>Mandatory:</i> No   <i>Max reps:</i> 1
<i>Explanation:</i>	Displays this help and exits program.
<i>Default:</i>	

## 4 Examples

### SERVER WITH THE DEFAULT PARAMETERS

*#!/CADIServer*

This execution will launch the CADIServer with the default options. The default options launch the server listening to in the 80 port, using four threads, and using the layer order as the rate distortion method.

### VIEWER

*#!/CADIViewer*

This execution will launch the CADIViewer. CADIViewer is a user-friendly interface and all options are set through menus.

### SERVER USING ANOTHER PORT

*#!/CADIServer -p 2080*

This execution will launch the server listening to the client request in the port 2080.

### CHANGE THE IMAGES PATH

*#!/CADIServer -tp workDir*

This execution will launch the CADIServer using the *workDir* as the root path for the images.

### CHANGE THE RATE DISTORTION METHOD

*#!/CADIServer -rd 2*

This execution will launch the server using the CPI method to send the response.