Catra Streaming Platform Manual

Author
Reviewer

Revision Draft
Last update March 14, 2006

1.Version Control

Date	Version	Comments	Author
21/11/2004	Draft	First revision issued	…

2.Summary

3.Glossary 

Catra Streaming Platform

The Catra Streaming Platform refers to all modules that made up the streaming platform and includes the following software components:

- Catra Streaming Server
- Catra Streaming Platform Integration

Catra Streaming Server

The Catra Streaming Server is the actual streaming server of the Catra Streaming Platform.

Catra Streaming Platform Integration

It is the way to interact with the Streaming platform and to integrate it with an external component.

4.Architecture

The Catra Streaming Platform is a streaming platform implementing the 3GPP streaming standard (www.3gpp.org).

This platform is made of different separate modules:

Catra Streaming Server that represent the actual streaming server within the platform;

Catra Streaming HTTP Interface, providing an interface that can be called through HTTP GET to interact with the Streaming Server;

Catra Streaming Integration represents the way an external component can interact with the Streaming Server for functionalities like authentication, authorization, real time billing and so on;

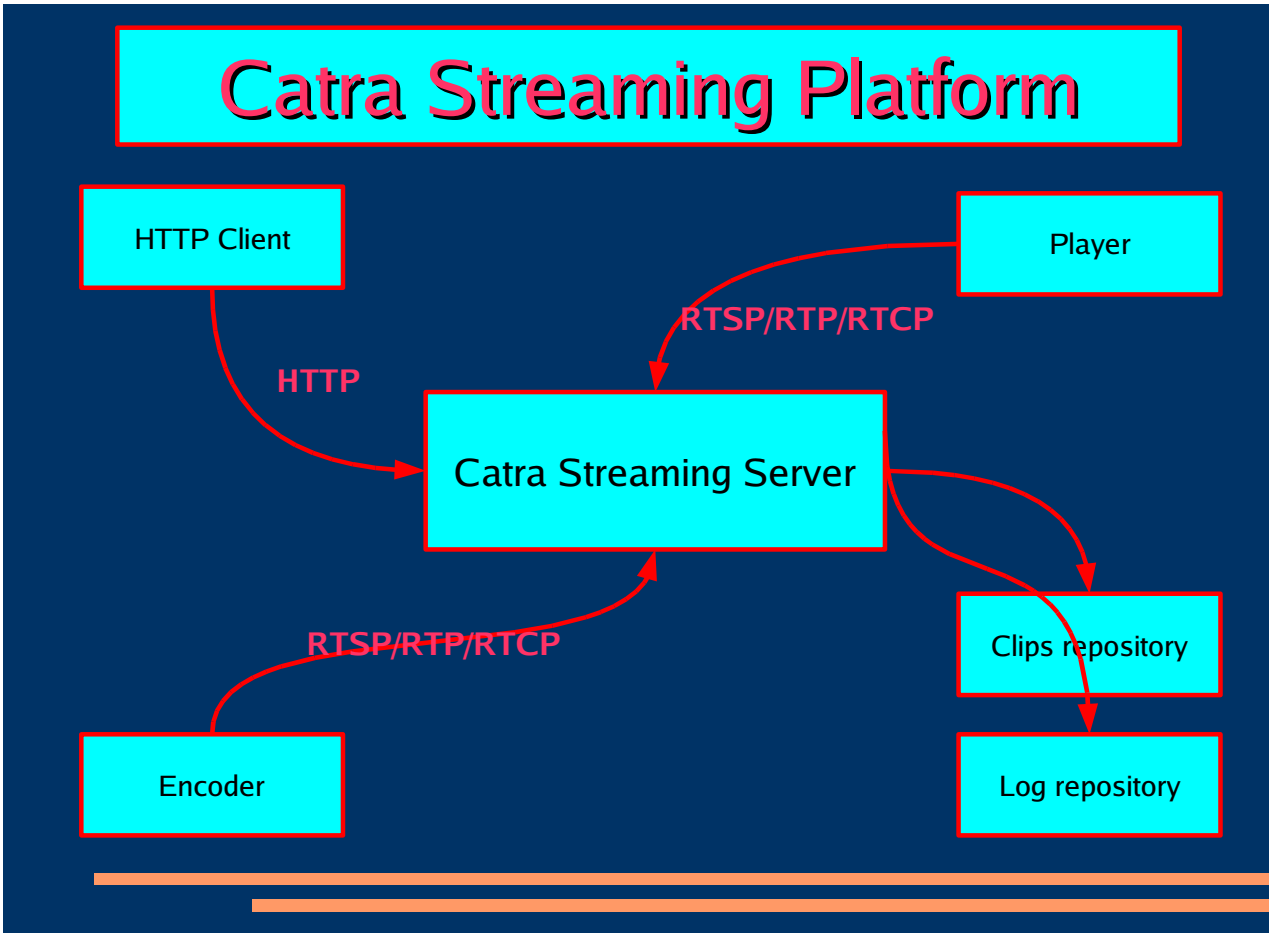
Beside these software components, the Catra Streaming Platform is comprised also of the following:

Clips Repository, a physical repository for the media storage;

Log Repository, a physical repository for the storage of the software modules activity logs and for streaming requests subscriber logs (CDR);

Application Server (optional), the Java Servlets (java.sun.com/products/servlet/) container accomplishing the integration side with other platforms.

Here’s the architectural diagram of the Catra Streaming platform:*Catra Streaming Server* internal arhitecture



4.1 *Catra Streaming Server* internal arhitecture

The Catra Streaming Server is completely based on open standard and architectures. It adheres to a huge number of internet and 3GPP standard protocols such as RTSP, RTP, RTCP and SDP.
Upon Catra Streaming Server execution, it starts with reading the configuration file in order to obtain its initialization parameters, such as the clips repository path, log files repository path and others.
It then starts listening for streaming requests on the configured network interface bound to a properly set IP address and port number (554 as default).
Whenever a streaming media player, implemented on a pc or on telephone terminal, performs a streaming requests to the Catra Streaming Server supplying within the RTSP URL the requested content, it opens different communication channels with the streaming server:

RTSP on TCP transport protocol creating the communication channel for the streaming session control through RTSP command options (such as DESCRIBE, SETUP, PLAY, PAUSE)

RTCP on UDP transport protocol creating the communication channel for the streaming session statistic purpose on a periodic basis;

RTP on UDP transport protocol creating the communication channel for streaming session payload delivery;

The Catra Streaming Server during while executing and serving streaming requests collects all necessary data regarding its activity and log in a persistent manner on compressed circular log files configurable on a per dimension and aging basis.

For performance reasons, allocated network resources are freed within a given time should they become stale or a timeout has expired.

5.Catra Streaming Platform Integration Guide

There are two way to interact with the Catra Streaming Platform through its HTTP interface:

1. using the Catra Streaming Platform as ‘Server’ calling the HTTP methods
2. using the Catra Streaming Platform as ‘Client’ using the HTTP ‘hooks’ provided by the platform

---

5.1*Catra Streaming Platform HTTP* Server Interface 

The Catra Streaming HTTP Interface represents the way to interact with the streaming server to perform different activities.

The HTTP interface is composed by a list of URLs that can be called by any HTTP client (including a HTTP browser) returning an HTTP answer encapsulating an XML.

The HTTP Interface can be grouped in different logical categories:

- administratives interface
- monitoring interface

The following table describes the administrative HTTP interface of the streaming platform:

Description	**URL**	**URL parameters**
KillStreamingSession: kill the specified streaming session NOT IMPLEMENTED YET	http://<HTTP IP>:<HTTP Port>/catraStreamingServer?command=killStreamingSession&RTSPSessionIdentifier=<RTSPSessionIdentifier>	• <HTTP IP>: IP address of the streaming server listening the HTTP interface • <HTTP Port>: Port of the streaming server listening the HTTP interface • <RTSPSessionIdentifier>: Identifier of the RTSP session
Shutdown: ask the streaming server to go down	http://<HTTP IP>:<HTTP Port>/catraStreamingServer?command=shutdown	• <HTTP IP>: IP address of the streaming server listening the HTTP interface • <HTTP Port>: Port of the streaming server listening the HTTP interface

Description	URL	URL parameters
GetContent: this method returns the list of the contents specified by <RelativeDirectory> directory inside the 'clips repository'	http://<HTTP IP>:<HTTP Port>/catraStreamingServer?command=getContents&Directory=<RelativeDirectory>	• <HTTP IP>: IP address of the streaming server listening the HTTP interface • <HTTP Port>: Port of the streaming server listening the HTTP interface • <RelativeDirectory>: the directory relative to the 'clips repository' (i.e.: must be "/" to have the contents of the clips repository root)
AddDirectory: add a directory to the clips repository	http://<HTTP IP>:<HTTP Port>/catraStreamingServer?command=addDirectory&NewDirectory=<NewDirectoryRelativePath>	• <HTTP IP>: IP address of the streaming server listening the HTTP interface • <HTTP Port>: Port of the streaming server listening the HTTP interface • <NewDirectoryRelativePath>: relative path to the clips repository of the new directory
DeleteDirectory: delete a directory from the clips repository	http://<HTTP IP>:<HTTP Port>/catraStreamingServer?command=deleteDirectory&DirectoryToBeDeleted=<DirectoryRelativePathToBeDeleted>	• <HTTP IP>: IP address of the streaming server listening the HTTP interface • <HTTP Port>: Port of the streaming server listening the HTTP interface • <DirectoryRelativePathToBeDeleted>: relative path to the clips repository of the directory to be deleted
DumpContent: read the file representing the content and dump the metadata of the content	http://<HTTP IP>:<HTTP Port>/catraStreamingServer?command=dumpContent&ContentRelativePath=<ContentRelativePath>	• <HTTP IP>: IP address of the streaming server listening the HTTP interface • <HTTP Port>: Port of the streaming server listening the HTTP interface • <ContentRelativePath>: Relative path of the content to be dumped
GetTracksInfo: read the file representing the content and return information about the tracks included in the file	http://<HTTP IP>:<HTTP Port>/catraStreamingServer?command=getTracksInfo&ContentRelativePath=<ContentRelativePath>	• <HTTP IP>: IP address of the streaming server listening the HTTP interface • <HTTP Port>: Port of the streaming server listening the HTTP interface • <ContentRelativePath>: Relative path of the content to be analized
GetConfiguration: return the configuration of the streaming server	http://<HTTP IP>:<HTTP Port>/catraStreamingServer?command=getConfiguration	• <HTTP IP>: IP address of the streaming server listening the HTTP interface • <HTTP Port>: Port of the streaming server listening the HTTP interface
SetConfiguration: modify a specific value of an item of the configuration	http://<HTTP IP>:<HTTP Port>/catraStreamingServer?command=setConfigurationValue&SectionName=<SectionName>&	• <HTTP IP>: IP address of the streaming server listening the HTTP interface • <HTTP Port>: Port of the streaming

Description	URL	URL parameters
	ItemName=<ItemName>&NewItemValue=<NewItemValue>	server listening the HTTP interface • <SectionName>: Name of the section of the configuration file • <ItemName>: Name of the item, inside the specified section, of the configuration file • <NewItemValue>: new value for the specified item

The following table describes the monitoring HTTP interface of the streaming platform:

Description	URL	URL parameters
GetConnectedUsers: return all the streaming session running on the streaming server with the relative details	http://<HTTP IP>:<HTTP Port>/catraStreamingServer?command=getConnectedUsers	• <HTTP IP>: IP address of the streaming server listening the HTTP interface • <HTTP Port>: Port of the streaming server listening the HTTP interface
GetServerInfo: return information about the server	http://<HTTP IP>:<HTTP Port>/catraStreamingServer?command=getServerInfo	• <HTTP IP>: IP address of the streaming server listening the HTTP interface • <HTTP Port>: Port of the streaming server listening the HTTP interface

---

5.2*Catra Streaming Platform HTTP* Client Interface

The Catra Streaming Platform could be integrated with external modules.

In fact it provides the following hooks:

1. at the beggining of the streaming session
2. at the end of the streaming session

The first integration could be used to perform any activities that must be done before the starting of the streaming session. Examples of activities could be:

- authentication of the customer requesting the streaming
- authorization the customer to stream
- communication the starting of the streaming to the billing platform
- communication with other platforms

The second integration could be used to perform any activities that must be done when the streaming session has finished. Some examples could be:

- save the point where the streaming is finished
- communication the ending of the streaming to the billing platform
- communication with other platforms

The behaviour of the Catra Streaming Platform is just the calling of an HTTP GET method at the beggining of a streaming session and at the end.

These HTTP GET methods are completely configurable through the configuration file.

5.2.1Hook at the beginning of the streaming

The hook at the beggining of the streaming causes the calling of a HTTP GET method by the Catra Streaming Server before the streaming is started.

To configure an hook at the beginning of each streaming session it is necessary to initialize correctly the configuration items of the 'BeginningHook' section inside the configuration file (see #10.Catra Streaming Server configuration file|outline).

The HTTP GET could refer to a simple text file or a complicated servlet or any other target, what it is important is the format of the HTTP body returning from the call.
In particular, the format of the body must be compliance to the following format:

```
<[AUTHORIZED | NOT_AUTHORIZED | FAILURE]\n
<message>>\n
```

An example could be:

```
<AUTHORIZED
this is any kind of message>
```

The HTTP GET method called at the beggining of the streaming affects the behaviour of the streaming session. In fact only if the return of the body is 'AUTHORIZED' the streaming will start, in all the other cases the streaming is rejected.

5.2.2Hook at the ending of the streaming

The hook at the ending of the streaming causes the calling of a HTTP GET method by the Catra Streaming Server at the ending of the streaming session.

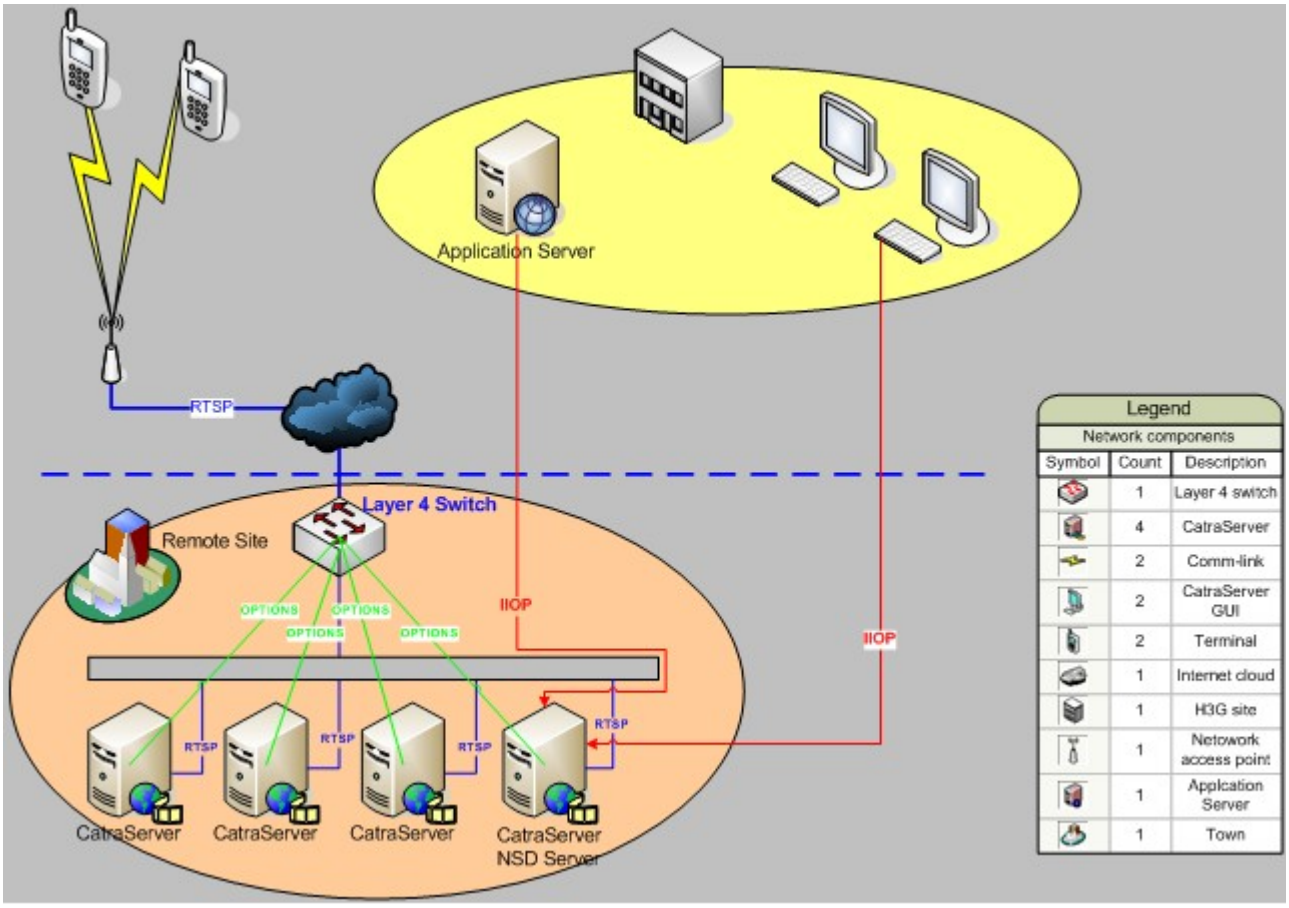
To configure an hook at the beginning of each streaming session it is necessary to initialize correctly the configuration items of the ‘EndingHook’ section inside the configuration file (see #10.Catra Streaming Server configuration file|outline).

The HTTP GET could refer to a simple text file or a complicated servlet or any other target.

The HTTP GET method called at the ending of the streaming does not affect the behaviour of the streaming session that it is already finished.

6.Physical Architecture

This paragraph describes an example of how the Catra Streaming Platform could be implemented.
For the sake of simplicity and in order to maintain high robustness requirements, the platform could be merge different software components within the same machine. For example each Catra Streaming Server can host its log repository and the Clips repository.
RTSP streaming requests will be load balanced to the Catra Streaming Severs pool through the use of a Layer 4 load balancer switch. The switch is in addition responsible for Catra Streaming Servers pool member fail detection. The failure detection will be performed sending RTSP-OPTIONS command on each Catra Streaming Server pool member and waiting for their response.



7.Interoperability

The Catra Streaming Server is compliant with the following **standards**:

- 3GPP TS 26.234 – PSS Protocols and codecs (Release 5)
- rfc2326: Real-Time Streaming Protocol (RTSP)
- rfc2327: Session Description Protocol (SDP)
- rfc1889: A Transport Protocol for Real-Time Applications (RTP)
- rfc2429: RTP Payload Format for the 1998 Version of ITU-T Rec. H.263 Video (H.263+)
- rfc3016: RTP Payload Format for MPEG-4 Audio/Visual Streams
- rfc3267: Real-Time Transport Protocol (RTP) Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs
- rfcisma

The Catra Streaming Server accepts the following **audio codecs**:

- AAC, GSMAMR, AMR-WB.

The **video** codecs accepted are:

- MPEG-4 and H263.

7.1Players complaint

The Catra Streaming Server is a streaming server compliant with 3GPP and ISMA standards and is able to interoperate with any standards-based media players.
The Catra Streaming Server, actually, has been tested with the following players:

- Quicktime ISMA player
- Philips ISMA and 3GPP player
- MPEG4IP ISMA player
- 3GPP players installed in UMTS phones actually in commerce

7.2Live encoders complaint

The Catra Streaming Server is compliant with any live encoder able to push RTP packets for audio and video. In this case the Catra Streaming Server will receive an RTSP request referring a SDP file including all the information about the live session.
Generally the SDP file is generated by the live encoder itself but can be also written manually.
Actually, the Catra Streaming Server has been tested with the following live encoder:

- mp4live encoder from the MPEG4IP open project
- Mpegable Broadcaster encoder
- Helix Mobile Producer
- Any encoder based on the xvid encoder
- Popwire encoder

8.Installation & Configuration

8.1Operating system requirements 

The operating systems required for the Catra Streaming Server are:

- any Linux distribution with Kernel v2.4 or later
- any HPUX 11 or later
- any Sun Solaris
- any Windows

Here are the TCP/IP optimization to be performed on Linux/unix environment.

Inside the /etc/sysctl.conf configuration file:

- `fs.file-max = 65536`
- `net.ipv4.tcp_max_syn_backlog = 16384`
- `net.ipv4.tcp_fin_timeout = 15`
- `net.ipv4.tcp_keepalive_time = 1800`
- `net.ipv4.tcp_window_scaling = 0`
- `net.ipv4.tcp_sack = 0`
- `net.ipv4.tcp_timestamps = 0`
- `net.ipv4.tcp_tw_reuse = 1`
- `net.ipv4.tcp_tw_recycle = 1`
- `net.ipv4.ip_local_port_range = 1024    65535`

Inside the /etc/security/limits.conf configuration file:

- `* soft nofile 65535`
- `* hard nofile 65535`

The Catra Streaming Server can be installed also on any MS Windows Platform but, for a production environment, we recommend to install it on a Linux/unix machine.

8.2Software requirements 

The streaming platform is built on top of the catralibraries (http://sourceforge.net/projects/catralibraries). Since these libraries are provided already in the Catra Streaming Platform package, you must not to download them.

Therefore no specific software requirements are needed for the server machines. Just follow the installation instructions, the installation package include all the required software.

8.3Installation activities 

Here are the instruction to be performed to install the Catra Streaming Server:

1. In order to perform the installation of the Catra Streaming Server, logon to the target machine e copy the Catra Streaming Server package to the target directory.

2. 

This step assume that the shell type of the root account is a bash shell.
If the root account is using another shell type you have to change
the way to initialize the environment variables.

```
tar xvf CatraStreamingServer.tar
su -        (login as root)
cp etc/catraStreamingServer.sh /
```

Open the "/catraStreamingServer.sh" file with your preferred text file editor (such as **vi**)
and initialize correctly the following environment variable:

- CATRASTREAMINGCONFIGURATIONPATHNAME: to be initialized with the absolute path name of the CatraStreamingServer.cfg file included in the Catra Streaming Platform package (i.e.: /home/giuliano/catrastreaming-1.7-pc-i386-linux-2.6/cfg/CatraStreamingServer.cfg)
- PATH: add the path where it is included the Catra Streaming Server executable (i.e.: /home/giuliano/catrastreaming-1.7-pc-i386-linux-2.6/bin)
- LD_LIBRARY_PATH: add the path where it is included the Catra Streaming Server libraries (i.e.: /home/giuliano/catrastreaming-1.7-pc-i386-linux-2.6/lib)

Open the ~/.bashrc file and add the following line:

```
. /catraStreamingServer.sh
```

2. 

Explode the Catra Streaming Server package.

Create the System environment variable called CATRASTREAMINGCONFIGURATIONPATHNAME and initialize it with the absolute path name of the CatraStreamingServer.cfg file included in
the Catra Streaming Platform package
(i.e.: d:\catrastreaming-1.7-pc-i386-Windows/cfg/CatraStreamingServer.cfg).
To create the System variable:

open the properties window of the 'Computer Resources' icon
select the 'Advanced' tab
click on the 'Environment Variables' button
add the CATRASTREAMINGCONFIGURATIONPATHNAME System variable

Remember that after you added the CATRASTREAMINGCONFIGURATIONPATHNAME System environment variable, you have to restart the machine in order to make visible the variable to any

application.

3. 

   Follow the #8.5.Catra Streaming Server configuration file|outline chapter to configure the Catra Streaming Platform (cfg/CatraStreamingServer.cfg file).

4. 

   To register the Catra Streaming Server it is necessary to run the following command as **root** (or Administrator for Windows platform):

```
su -        (this command only for linux)
catraStreamingServer -i
```

   Verify that the operation has had success looking the /tmp/ServiceDebug.log file for linux/unix platform and C:/ ServiceDebug.log for Windows Application.

   See #9.1.Start of the Catra Streaming Server |outline to start the streaming server.

8.4Post-installation check list 

For the simplicity of the installation steps, there are no specific actions in order to verify that the installation is correctly performed.

8.5Catra Streaming Server configuration file

The Catra Streaming Server configuration is easily performed through the modification of the configuration file.
Though this file includes several configuration parameters, for the sake of use a minimal set of them is reported in order to start the server and make it works.

 **For a detailed explanation of configuration paramaters refer to CatraStreamingServer.cfg configuration file.**

8.5.1Minimal configuration of the Catra Streaming Server

In order to define a minimal configuration of the Catra Streaming Server you must alter the CatraStreamingServer.cfg file provided by with the installation package.
Therefore open the *CatraStreamingServer.cfg* file with your preferred text file editor (such as **vi** for linux, **WordPad** for Windows) and enter the suitable parameter in accordance with your target environment.

```
[StreamingServer]
      ContentRootPath = <absolute path of the ClipDir>
     Standard = <The standard to be used. Possible values are: '3GPP' (mobile) or 'ISMA'
(internet)>
[IPConfiguration]
      LogicalIPAddressForRTSP = <RTSP listening address>
      LogicalIPAddressForRTP = <IP address for RTP traffic>
     LogicalIPAddressForRTCP = <IP address for RTCP traffic>
[SystemLogs]
      BaseTraceFileName = <absolue path name for system trace files>
[SubscriberLogs]
      BaseTraceFileName = <absolue path name for subscriber trace files>
```

These changes are sufficient to allow the Catra Streaming Server to run.

8.5.2CatraStreamingServer.cfg configuration file

The CatraStreamingServer.cfg is the only configuration file of the Catra Streaming Server.
Here is a detailed description of each parameter included inside this configuration file:

Section	Item	Value Type	Default	Description
StreamingServer	Name	String	No default	Unique logical name for the Streaming server.
	RTSPRequestsPort	Positive integer	554	RTSP listener port. Since the 554 default port needs that the Streaming Server must to be executed as root (or administrator in Windows system) right, it is better to choice another port major than 1024 (for ex. 7777) and perform a NAT rule into the Layer 4 load balancer.
	ContentRootPath	String	No default	Contents root path. On linux/unix environment remember to use '/' as directory separator and '\' in Windows systems.
	MaxRTSPSessions	Positive integer	500	Max RTSP sessions managed by every Catra Streaming Server
	StandardVerify that the operation has had success looking How to read the System trace of a Catra Streaming Server. In case the System trace does not contain any information, check on /tmp/ServiceDebug.log for linux/unix platform and C:/ServiceDebug.log for Windows Application.	String	3GPP	The standard to be used. Possible values are: '3GPP' (mobile) or 'ISMA' (internet).
IPConfiguration	LocalIPAddressForRTSP	IP address	No default	Local IP address to be used for RTSP traffic
	LocalIPAddressForRTP	IP address	No default	Local IP address to be used for RTP traffic
	LocalIPAddressForRTCP	IP address	No default	Local IP address to be used for RTCP traffic
	RTP_RTCPStartingPort	Positive integer	20000	Starting reserved port for RTP/RTCP communications
	RTP_RTCPReservedPorts	Positive integer	25000	Number of reserved ports for RTP/RTCP communications
Scheduler	SchedulerSleepTimeInMilliSecs	Positive integer	500	The scheduler inside the streaming server checks periodically the expiration of his tasks and the period is specified in milliseconds by this parameter
	CheckServerSocketPeriodInMilliSecs	Positive integer	600	The streaming server checks any new RTSP connection periodically and the period is specified in milliseconds by this parameter
	CheckSocketsPoolPe	Positive	600	Once the RTSP connection arrives, the

	riodInMilliSecs	integer		streaming server checks any new received RTSP commands periodically and the period is specified in milliseconds by this parameter
	SendRTCPPacketsPeriodInMilliSecs	Positive integer	5000	For each RTSP session, the streaming server sends statistical RTCP packets to the player periodically and the period is specified in milliseconds by this parameter.
	ReceiveRTCPPacketsPeriodInMilliSecs	Positive integer	5000	Once the RTSP connection arrives, the streaming server checks any new received statistical RTCP packets periodically and the period is specified in milliseconds by this parameter
	RTSP_RTCPTimeout	Positive integer	60	The RTSP connection will be interrupted if the session does not receive any RTSP or RTCP packets within this timeout (measured in seconds). Generally not receiving any RTSP command and RTCP packets by the Streaming Server, it means the network connection between server and player is down. Default according to rfc2326 is 60.
	PauseTimeout	Positive integer	120	The RTSP connection in PAUSE state will be interrupted if the session does not change his state by a new RTSP command within this timeout (measured in seconds). PauseTimeout must be major than RTSP_RTCPTimeout.
Cache	FileCacheSizeInBytes	Positive integer	1024000	The size of the cache used to read the clip file is specified by this parameter. That cache represents the lower level of cache.
	MaxMp4FilesNumberInMemory	Positive integer	40	At the end of an on-demand RTSP connection, the streaming server maintains a lot of information inherent the relative clip file just streamed. This parameter indicates the number of MP4 files which the relative information have to be maintained in cache. That cache represents the higher level of cache.
	Mp4FilesNumberToDeleteOnOverflow	Positive integer	10	When another MP4 file must be inserted in the cache and the cache is already full, a number of oldest (less recent requested) MP4 file information will be release from the cache to create space for the new requests. This parameter represents the number of MP4 files information to be released from the cache and it must be less than *MaxMp4FilesNumberInMemory*.
	BucketsNumberForMP4FilesCache	Positive integer	50	The MP4 files information is inserted into a hash map. This parameter represents the buckets number used by the hash map.
	UseMP4ConsistencyCheck	Boolean (true or false)	false	This parameter indicates if the streaming server must interrupt the on-demand request referring a MP4 file not well done according the MP4 standard.

Authorization	Activated	Boolean (true or false)	false	The streaming server could ask to an external component the authorization to stream a request. This parameter indicates if this functionality is enabled.
	Servlet	String	“/authorization/servlet/AuthorizationProxy?op%61getAuthorization&"	To have the authorization to stream, the streaming server activates a servlet through an HTTP GET. This parameter indicates the servlet pathname to be called by the streaming server. The streaming server adds also some parameters that could be used by the servlet authorizing the streaming request. The GET HTTP request will be: <servlet pathname> + "PlayerURL=" + <PlayerURL> + "&PlayerIP=" + <PlayerIP> The <PlayerURL> parameter value is encoded according to the simple substitutions: • '?' with '@' • '=' with <diesis character> • '&' with '~' The servlet must decode the <PlayerURL> to obtain the original <PlayerURL> parameter value. Since the configuration item cannot contain the '=' character, if it is necessary to use it to initialize this parameter, use '%61' instead of '='.
	LocalIPAddressForHTTP	IP address	No default	Local IP address to be used for HTTP traffic
	WebServerIPAddress	IP address	No default	IP address of the WEB Server to be used
	WebServerPort	Positive integer	8080	WEB server listener port
	HTTPRequestTimeoutInSecs	Positive integer	10	The streaming server waits the HTTP servlet answer for a specified period in seconds
Commit	Activated	Boolean (true or false)	false	At the end of a streaming session, the streaming server could inform an external component that the streaming session is finished. This parameter indicates if this functionality is enabled.
	Servlet	String	“/authorization/servlet/AuthorizationProxy?op%61commit&"	To inform an external component that the streaming request is finished, the streaming server activates a servlet through an HTTP GET. This parameter indicates the servlet pathname to be called by the streaming server. The streaming server adds also some parameters that could be used by the servlet. The GET HTTP request will be: <servlet pathname> + "PlayerURL=" + <PlayerURL> + "&PlayerIP=" + <PlayerIP> + "&AuthorizationMessage=" + <AuthorizationMessage> + "&LastRelativeTransmissionTimeInSecs=" + <LastRelativeTransmissionTimeInSecs>

				The <PlayerURL> parameter value is encoded according to the simple substitutions: • '?' with '@' • '=' with <diesis character> • '&' with '~' The servlet must decode the <PlayerURL> to obtain the original <PlayerURL> parameter value. Since the configuration item cannot contain the '=' character, if it is necessary to use it to initialize this parameter, use '%61' instead of '='.
	LocalIPAddressForHTTP	IP address	No default	Local IP address to be used for HTTP traffic
	WebServerIPAddress	IP address	No default	IP address of the WEB Server to be used
	WebServerPort	Positive integer	8080	WEB server listener port
	HTTPRequestTimeoutInSecs	Positive integer	10	The streaming server waits the HTTP servlet answer for a specified period in seconds
SystemLogs	BaseTraceFileName	String	No default	The system log files are generated by the Tracer to include information about all the activities made by the streaming server. It generates one new log files when the current log file reaches a specific size or after a specified period. Since we have many files generated by the streaming server, this parameter indicates the base trace file name (an absolute path name); the specific trace file name is obtained concatenating to this parameter a sequential number. On linux/unix environment remember to use '/' as directory separator instead of '\' that is used for Windows platforms.
	MaxTraceFileSize	Positive integer	10000	This parameter is the max size in K-bytes that a trace file can reach. Once the trace file reaches that specific size, the Tracer will generate a new trace file.
	TraceFilePeriodInSecs	Positive integer	36000	This parameter is the period in seconds after that the Tracer will generate a new trace file.
	CompressedTraceFile	Boolean (true or false)	true	The parameter indicates if the Tracer must compress the trace file once it is completed.
	TraceFilesNumberToMaintain	Positive integer	900	The Tracer will generate a new trace file each time a specific trace size or period is reached. This parameter indicates the max number of trace files the Logs Repository must have. One this number is reached, the Tracer deletes the oldest trace file.
	TraceOnTTY	Boolean (true or	false	The Tracer writes always every trace on the trace file. If this parameter is initialized to true,

		false)		every trace is written also to the standard output.
	CacheSizeOfTraceFile	Positive integer	1000	The Tracer does not write the traces directly into the trace file, but it uses a cache for performance reason. This parameter represents the size in K-byte of this cache. If CacheSizeOfTraceFile is initialized to -1, the cache will not be used and the traces will be flushed as they are added to the Tracer
	TraceLevel	Enumerative	LDBG6	The quantity of information you have inside the trace files is determined from the trace level indicated by this parameter. The trace level can be initialized to one of the following values: • LDBG1 • LDBG2 • LDBG3 • LDBG4 • LDBG5 • LDBG6 • LINFO • LMESG • LWRNG • LERRR • LFTAL
	ListenTracePort	Positive integer	6010	The Tracer can change dynamically his configuration (any parameter specified inside this section) without stopping the process but just sending a specified command through a socket. This parameter specifies the socket port where the System Tracer will listen.
SubscriberLogs	BaseTraceFileName	String (Absolute path name)	No default	The subscriber log files are generated by the Tracer at the end of each streaming session to generate a CDR including all the information regarding the session. It generates one new log files when the current log file reaches a specific size or after a specified period. Since we have many files generated by the streaming server, this parameter indicates the base trace file name (an absolute path name), the specific trace file name is obtained concatenating to this parameter a sequential number. On linux/unix environment remember to use '/' as directory separator instead of '\' that is used for Windows platforms.
	MaxTraceFileSize	Positive integer	1000	This parameter is the max size in K-bytes that a trace file can reach. Once the trace file reaches that specific size, the

				Tracer will generate a new trace file.
	TraceFilePeriodInSecs	Positive integer	36000	This parameter is the period in seconds after that the Tracer will generate a new trace file.
	CompressedTraceFile	Boolean (true or false)	True	The parameter indicates if the Tracer must compress the trace file once it is completed.
	TraceFilesNumberToMaintain	Positive integer	900	The Tracer will generate a new trace file each time a specific trace size or period is reached. This parameter indicates the max number of trace files the Logs Repository must have. One this number is reached, the Tracer deletes the oldest trace file.
	TraceOnTTY	Boolean (true or false)	False	The Tracer writes always every trace on the trace file. If this parameter is initialized to true, every trace is written also to the standard output.
	CacheSizeOfTraceFile	Positive integer	100	The Tracer does not write the traces directly into the trace file, but it uses a cache for performance reason. This parameter represents the size in K-byte of this cache. If CacheSizeOfTraceFile is initialized to -1, the cache will not be used and the traces will be flushed as they are added to the Tracer.
	TraceLevel	Enumerative	LINFO	The quantity of information you have inside the trace files is determined from the trace level indicated by this parameter. In case of subscriber log, this parameter must be initialized to LINFO.
	ListenTracePort	Positive integer	6011	The Tracer can change dynamically his configuration (any parameter specified inside this section) without stopping the process but just sending a specified command through a socket. This parameter specifies the socket port where the System Tracer will listen.
	RequestLogFormat	String	"Request summary. Start streaming session: %SYYYY%-%SMM%-%SDD% %SHH%:%SMI%:%SSS%:%SMILL% | Client IP: %CIP% | URL: %URLWP% | Par.: %URLP% | User agent: %USERAGENT% | Status: %STATUS%	At the end of each streaming session, the subscriber Tracer logs a summary of the streaming request with the format described by this parameter. This parameter is just a string where it is possible to put some keywords that the streaming server interpreters and substitute with the appropriate value. Here are the list of the keywords you can use: • %SYYYY%: streaming session starting year 4 digits (i.e.: 2004) • %SMM%: streaming session starting month 2 digits (i.e.: 01 for January) • %SDD%: streaming session starting day 2 digits (i.e.: 17) • %SHH%: streaming session starting

			Standard used: %STD% | Is live: %CL% | Duration: %DUR% | Video codec: %VCOD% | Video average bit rate: %VBR% | Video packets sent: %VPS% | Video bytes sent: %VBS% | Video RTCP Packets received: %VRTCPP% | Video RTCP bytes received: %VRTCPB% | Video jitter: %VJIT% | Audio codec: %ACOD% | Audio average bit rate: %ABR% | Audio packets sent: %APS% | Audio bytes sent: %ABS% | Audio Packets lost: %APL% | Audio RTCP Packets received: %ARTCPP% | Audio RTCP bytes received: %ARTCPB% | Audio jitter: %AJIT% | Connection time: %CT% | Last relative time: %LRT% | Streaming time: %STREAMTIME% | Authorization message: %AUTHORIZATIONMESSAGE%"	hour 2 digits (i.e.: 18) • %SMI%: streaming session starting minutes 2 digits (i.e.: 35) • %SSS%: streaming session starting seconds 2 digits (i.e.: 15) • %SMILL%: streaming session starting millseconds 4 digits (i.e.: 0876) • %CIP%: Client IP address • %URLWP%: URL requested without parameters • %URLP%: parameters included into the URL • %USERAGENT%: User Agent (player identifier) • %STATUS%: Status of the session (Error or Success) • %STD%: standard used (ISMA or 3GPP). NA if the connection is live. • %CL%: true if the connection is live, false if it is off-line • %DUR%: duration of the movie (max 3 decimal digits), NA if the connection is live • %VCOD%: video codec (NA if there is not audio track) • %VPS%: video packets sent (NA if there is not video track) • %VBS%: video bytes sent (NA if there is not video track) • %VPL%: video packets lost (NA if there is not video track) • %VBR%: video average bit rate (NA if there is not video track) • %VRTCPP%: video RTCP packets received (NA if there is not video track) • %VRTCPB%: video RTCP bytes received (NA if there is not video track) • %VJIT%: video jitter (NA if there is not video track) • %ACOD%: audio codec (NA if there is not audio track) • %APS%: audio packets sent (NA if there is not audio track) • %ABS%: audio bytes sent (NA if there is not audio track) • %APL%: audio packets lost (NA if there is not audio track) • %ABR%: audio average bit rate (NA if there is not audio track) • %ARTCPP%: video RTCP packets received (NA if there is not video track)

				• %ARTCPB%: video RTCP bytes received (NA if there is not video track) • %AJIT%: audio jitter (NA if there is not video track) • %CT%: connection time (in seconds) starting with the acception of the client connection and finishing when the connection goes down • %LRT%: last relative time of the clip. NA if the connection is live • %STREAMTIME%: seconds of streaming (playing state) without PAUSE periods • %AUTHORIZATIONMESSAGE%: message or error returned by the authorization servlet
	FlushTraceAfterEachRequest	Boolean (true or false)	false	The Tracer flushes the traces when the cache size is reached (CacheSizeOfTraceFile parameter). If this parameter is initialized to true, the flush of the trace is made at the end of each streaming session
System	StreamingServerProcessorsNumber	Positive integer	2	This parameter indicates the number of the threads started by the Streaming Server. Normally it is initialized according the number of the processors running on the current machine.
	MaxServerBandWidthInKbps	Positive integer	32000	The Streaming Server will not accept any new streaming session in case it is having a throughput major than the specified parameter. This parameter indicates the max bandwidth supplied by the Streaming Server (Kbps).
	MaxPayloadSizeInBytes	Positive integer	1470	This parameter indicates the max payload size of each RTP packet. If the player requires (RTSP header ‘blocksize’) a max packet size less than the configured here, the player value will be used. If the player requires a max packet size greater than the configured here, the configured value will be used.
	RTPPacketsNumberToPrefetch	Positive integer	3	It is the packets pre-built from the Streaming Server during each streaming session for each track. This parameter is used for rfcisma (AAC), rfc3267 (AMR).
	SamplesNumberToPrefetch	Positive integer	3	It is the packets pre-built (in term of media samples) from the Streaming Server during each streaming session for each track. This parameter is used for rfc2429 (H.263), rfc3016 (AAC, MPEG4)
	MaxLiveSourcesNu	Positive	10	A live feed is a RTP flow generally coming from

	mber	integer		an encoder. This parameter indicates the max number of live feeds the server can handle.
	PlayDelayTimeInMilliSeconds	Positive integer	100	After the RTSP handshake between player and server, the Streaming Server starts to send RTP packets to the player. Since the RTP data is going over UDP, to make sure that the RTSP PLAY answer arrives to the player before the first RTP packet, the Streaming Server can introduce an artificial delay in his schedule. This parameter indicates the artificial delay between the PLAY answer and the first RTP packet.
	SendRTPMaxSleepTimeInMilliSecs	Positive integer	400	This parameter indicates the max delay time in case the RTP packet is built too early
	IsOverBufferEnabled	Boolean (true or false)	false	Set if the overbuffer is enabled
	SendingInterval	Positive integer	200	This parameter, expressed in milliseconds, is a way to control the algorithm establishing if a RTP packet should be sent or not. Basically, all the RTP packets with relative transmission time inside the sending interval (llCurrentTime + lSendInterval) will be transmitted
	InitialWindowSizeInBytes	Positive integer	No default	Max window size in bytes of the RTP over buffer
	MaxSendAheadTimeInSec	Positive integer	25	In case the client supports over buffering, this is the farthest in advance the server will send a packet to a client
	OverbufferRate	Double	2.0	
	MaxSpeedAllowed	Double	4.0	We can send the RTP packets faster or slower then the encoded bit rate according the RTCP reports. This value indicated that we cannot send the RTP packets faster then 4 times the encoded bit rate Remark: Also the player, according the RTSP protocol, can suggest a Speed field in the PLAY command.
	UseOfHintingTrackIfExist	Boolean (true or false)	false	This parameter indicates the usage or not of the hinting track in case it exists
	BuildOfHintingTrackIfNotExist	Boolean (true or false)	false	This parameter indicates if it is necessary to build the hint track in case it does not exist. This parameter is considered only if the UseOfHintingTrackIfExist parameter is true
	License	String	No default	Crypted license of the Streaming Server

8.6Un-installation 

Here are the instruction to be performed to un-install the Catra Streaming Server:

1. 

   To un-register the Catra Streaming Server it is necessary to run the following command as root (or Administrator for Windows platform):

   ```
   catraStreamingServer -u
   ```

   Verify that the operation has had success looking the /tmp/ServiceDebug.log file for linux/unix platform and C:/ ServiceDebug.log for Windows Application.

2. 

   Remove the directory including the Catra Streaming Platform.

3. 

   Remove the “/catraStreamingServer.sh” file and the line

   ```
   . /catraStreamingServer.sh
   ```

   from the ~/.bashrc file.

4. 

   Remove the CATRASTREAMINGCONFIGURATIONPATHNAME system environment variable

9.Operations

Once the Catra Streaming Platform is installed and configured, we are ready to run it and start numerous activities. This chapter describes all the activities you can do with this platform as for example how to run a Catra Streaming Server, how to check the status of the platform or of a specific request, and so on….
Have a nice work with the Catra Streaming Platform.

9.1Start of the Catra Streaming Server

There are many way to stop the Catra Streaming Platform:

1. 
   logon to the Catra Streaming Server machine and if you use the graphical environment (i.e.: GNOME)start the server through the 'Services' program reachable from the menù following: 'Desktop' -> 'System Settings' -> 'Server Settings'
1. 
   logon to the Catra Streaming Server machine and through a shell type

   ```
   /sbin/service catraStreaming start
   ```

   or

   ```
   catraStreamingServer -start
   ```

1. 
   Open the 'Services' program and start the 'Catra Streaming Platform' service.

Verify that the operation has had success looking How to read the System trace of a CatraStreaming Server.
In case the System trace does not contain any information, it means that we had some problems to start the service, therefore, check the /tmp/ServiceDebug.log file for linux/unix platform and C:/ ServiceDebug.log for Windows Application.

9.2Stop of the Catra Streaming Server

There are many way to stop the Catra Streaming Platform:

1. 


Using the HTTP ‘shutdown’ command, see #5.1.Catra Streaming Platform HTTP Server Interface| outline

2. 


logon to the Catra Streaming Server machine and if you use the graphical environment (i.e.: GNOME)stop the platform through the ‘Services’ program reachable from the menù following: ‘Desktop’ -> ‘System Settings’ -> ‘Server Settings’

3. 


logon to the Catra Streaming Server machine and through a shell type

```
/sbin/service catraStreaming stop
```

or

```
catraStreamingServer -stop
```

1. 

Open the ‘Services’ program and stop the ‘Catra Streaming Platform’ service.

Verify that the operation has had success looking How to read the System trace of a CatraStreaming Server.
In case the System trace does not contain any information, it means that we had some problems to stop the service, therefore, check the /tmp/ServiceDebug.log file for linux/unix platform and C:/ ServiceDebug.log for Windows Application.

9.3How to know the status of a Catra Streaming Server 

There are many way to know the status of the Catra Streaming Platform:

1. 
2. 

Using the HTTP ‘getServerInfo’ command, see #5.1.Catra Streaming Platform HTTP Server Interface|outline

3. 
logon to the Catra Streaming Server machine and if you use the graphical environment (i.e.: GNOME) through the ‘Services’ program reachable from the menù following: ‘Desktop’ -> ‘System Settings’ -> ‘Server Settings’

4. 
logon to the Catra Streaming Server machine and through a shell type

```
/sbin/service catraStreaming status
```

or

```
catraStreamingServer -status
```

1. 
Open the ‘Services’ program and see the status of the ‘Catra Streaming Platform’ service.

9.4How to change a configuration parameter of a Catra Streaming Server

Use the ‘SetConfiguration’ HTTP interface or

Using the HTTP Interface:

1. Open a HTTP browser
2. Use the SetConfiguration HTTP interface to change the configuration item

Without using the HTTP Interface:

1. logon to the Catra Streaming server machine
2. Edit the CatraStreamingServer.cfg configuration file and change the configuration parameter

The Catra Streaming Server will not be affected by this change until it is not restarted.

9.5How to set the clip repository to a Catra Streaming Server

Follow How to change a configuration parameter of a CatraStreaming Server to change the following parameter: ‘StreamingServer’ section, ‘ContentRootPath’ item.

9.6How to start an off-line streaming session using the ISMA (Internet) standard

Fill the content directory (*StreamingServer* Section and *ContentRootPath* item of the CatraStreamingServer.cfg configuration file) with the clip you want to stream. Be sure that the Standard item into the configuration file is initialized to ‘3GPP’. Run an ISMA player (for example the MPEG4IP player or the QuickTime player)
Open the URL:
rtsp://<CatraStreamingPlatformIP>:<CatraStreamingPlatformPort>/<ClipRelativePathName>
where
<CatraStreamingPlatformIP> is the IP of the Load Balancer in front of the Catra Streaming Servers or the IP of a specific Catra Streaming Server
<CatraStreamingPlatformPort> is the socket port configured to listen RTSP commands as specified inside the configuration file (*StreamingServer* Section and *RTSPRequestsPort* item)
<ClipRelativePathName> is the clip relative path name respect to the content root path.

9.7How to start an off-line streaming session using the 3GPP (mobile phone) standard

Fill the content directory (*StreamingServer* Section and *ContentRootPath* item of the CatraStreamingServer.cfg configuration file) with the clip you want to stream. Be sure that the Standard item into the configuration file is initialized to '3GPP'.
Run a 3GPP player (for example the Philips player)
Open the URL:
rtsp://<CatraStreamingPlatformIP>:<CatraStreamingPlatformPort>/<ClipRelativePathName>
where
<CatraStreamingPlatformIP> is the IP of the Load Balancer in front of the Catra Streaming Servers or the IP of a specific Catra Streaming Server
<CatraStreamingPlatformPort> is the socket port configured to listen RTSP commands as specified inside the configuration file (*StreamingServer* Section and *RTSPRequestsPort* item)
<ClipRelativePathName> is the clip relative path name respect to the content root path.

9.8How to start a live streaming session (ISMA or 3GPP)

Fill the content directory (*StreamingServer* Section and *ContentRootPath* item of the CatraStreamingServer.cfg configuration file) with the SDP file referring the live session. Generally the SDP file is generated by the live encoder but it can be also written manually. Since the Catra Streaming Server recognize the SDP file though the extension of the file, it is important that the extension is '.sdp'.
Run a 3GPP or ISMA player according if the live encoder is respectively 3GPP or ISMA
Open the URL:
rtsp://<CatraStreamingPlatformIP>:<CatraStreamingPlatformPort>/<SDPRelativePathName>
where
<CatraStreamingPlatformIP> is the IP of the Load Balancer in front of the Catra Streaming Servers

or the IP of a specific Catra Streaming Server
<CatraStreamingPlatformPort> is the socket port configured to listen RTSP commands as specified inside the configuration file (*StreamingServer* Section and *RTSPRequestsPort* item)
<SDPRelativePathName> is the SDP relative path name respect to the content root path.

The SDP file must be something like:

```
v=0
o=- 1115315157130610 1115315157130612 IN IP4 10.214.138.5
s=capture.sdp
c=IN IP4 10.214.138.5
t=0 0
b=AS:68
a=control:*
m=video 20000 RTP/AVP 96
b=AS:52
a=rtpmap:96 MP4V-ES/90000
a=control:trackID/1
a=fmtp:96 profile-level-id=8; config=000001b008000001b509000001000000012000086c4002b182c2090a31f;
m=audio 20002 RTP/AVP 97
b=AS:16
a=rtpmap:97 MP4A-LATM/8000
a=fmtp:97 cpresent=0;config=40002B200000
a=control:trackID/2
```

9.9How to start an off-line streaming session from a specific start point 

To start an off-line streaming session from a specific start point it is necessary to follow the instructions of How to start an off-line streaming session using the ISMA standard or How to start an off-line streaming session using the 3GPP standard according if you want to stream using the ISMA or 3GPP standard with the exception to add the 'StartTimeInSecs' parameter to the URL to specify the start point.
Therefore the URL will be:
rtsp://<CatraStreamingPlatformIP>:<CatraStreamingPlatformPort>/<ClipRelativePathName>?StartTimeInSecs=<StartTime>

where

<StartTime> is the start point in seconds you want to start the streaming

9.10How to start an off-line or live streaming session specifying a session timeout 

To start an off-line or live streaming session from a specific start point it is necessary to follow the instructions of How to start an off-line streaming session using the ISMA standard or How to start an off-line streaming session using the 3GPP standard or How to start a live streaming session (ISMA or 3GPP) according if you want to stream a clip or a live session with the exception to add the 'StreamingSessionTimeoutInSecs' parameter to the URL to specify the timeout. Therefore the URL will be:
rtsp://<CatraStreamingPlatformIP>:<CatraStreamingPlatformPort>/<ClipRelativePathName>?StreamingSessionTimeoutInSecs=<Timeout>

where

<Timeout> is the timeout in seconds of the streaming session

9.11How to set the system trace level of a Catra Streaming Server 

1. Follow How to change a configuration parameter of a CatraStreaming Server to change the foillowing parameter: 'SystemLogs' section, 'TraceLevel' item.

Refer to Appendix C – Trace levels and related information of the CatraStreaming Server to know details about the trace levels and information associated to each trace level.

9.12How to flush the cache of the System and Subscriber Trace of a Catra Streaming Server

If the CacheSizeOfTraceFile configuration item is set to '-1', the traces are always flushed. If it is not set to '-1':

1. logon to the Catra Streaming server machine
2. 

```
tracerClient <CatraStreamingServerIPAddress> <SystemLogTracePort> flushTraceFileCache
tracerClient <CatraStreamingServerIPAddress> <SubscriberLogTracePort> flushTraceFileCache
```

9.13How to read the System or Subscriber Trace of a Catra Streaming Server

1. logon to the Catra Streaming Server machine
2. Verify that the System Trace level is initialized according the information you want to know.

   Refer to Appendix C – Trace levels and related information of the CatraStreaming Server to know details about the trace levels and information associated to each trace level.

   Refer to How to set the system trace level of a CatraStreaming Server to modify the trace level of a Catra Streaming Server.

3. If the Tracer is configured to use the Cache (CacheSizeOfTraceFile configuration item initialized to a value different of '-1'), flush the trace following the instructions for How to flush the cache of the system and subscriber trace of a CatraStreaming Server.

4. Read the System or Subscriber Trace file (the path is specified in ‘BaseTraceFileName’ item of the CatraStreamingServer.cfg) with your preferred text file editor (such as vi).

9.14How to configure the format of the Subscriber logs (CDRs) of a Catra Streaming Server 

1. Follow How to change a configuration parameter of a CatraStreaming Server to change the following parameter: ‘SubscriberLogs’ section, ‘RequestLogFormat’ item.

Refer to Appendix D – CDRs format to know details about the information that a CDR can include and his format inside the subscriber log.

9.15How to change the cache parameters of a Catra Streaming Server 

1. Follow How to change a configuration parameter of a CatraStreaming Server to change the following parameters:
‘Cache’ section, ‘FileCacheSizeInBytes’ item
‘Cache’ section, ‘MaxMp4FilesNumberInMemory’ item
‘Cache’ section, ‘Mp4FilesNumberToDeleteOnOverflow’ item
‘Cache’ section, ‘BucketsNumberForMP4FilesCache’ item
‘Cache’ section, ‘RequestLogFormat’ item

Refer to Appendix E – Cache sizing parameters to know details about the meaning of the cache sizing parameters.

10.Contacts 

For any kind of information please do not hesitate to communicate through the catrastreaming-support@catrasoftware.it e-mail address.

The mailing list is: catrastreaming-list@catrasoftware.it.
To subscribe to the mailing list send an e-mail to catrastreaming-list-subscribe@catrasoftware.it with an empty subject and body. You will receive a message, unfortunately in Italian, and you should just reply to confirm the subscription.
To unsubscribe to the mailing list send an e-mail to catrastreaming-list-unsubscribe@catrasoftware.it with an empty subject and body. You will receive a message, unfortunately in Italian, and you should just reply to confirm the un-subscription.

SDP file example

```
v=0
o=- 1115315157130610 1115315157130612 IN IP4 10.214.138.5
s=capture.sdp
c=IN IP4 10.214.138.5
t=0 0
b=AS:68
a=control:*
m=video 20000 RTP/AVP 96
b=AS:52
a=rtpmap:96 MP4V-ES/90000
a=control:trackID/1
a=fmtp:96                                                                    profile-level-id=8;
config=000001b008000001b509000001000000001200086c4002b182c2090a31f;
m=audio 20002 RTP/AVP 97
b=AS:16
a=rtpmap:97 MP4A-LATM/8000
a=fmtp:97 cpresent=0;config=40002B200000
a=control:trackID/2
```

CatraStreamingServer.cfg configuration file example

Catra Streaming Server Errors

Trace levels and related information of the Catra Streaming Server

CDRs information and their format inside the Subscriber log

Cache sizing parameters