

Product Sheet

# MediaStore Enterprise

## MAM Client Application Module

**SI MEDIA s.r.l.**

**HeadQuarters:** Via Vostanza, 5 - 31039 Riese Pio X (TV) - Italy

**T** +39 0423 750075 **F** +39.0423 750150 **E** info@si-media.tv

**www.si-media.tv**

 @SIMedia1978

 SI Media

**APAC Branch Office:** 21 Serangoon North Ave 5, #06-04

Ban Teck Han Building, 554864 Singapore

**T** +65 8432 5394

## MediaStore Enterprise

**MediaStore** is the MAM (*Media and Digital Asset Management system*) proposed by SI Media. **MediaStore** is an application suite including server and client modules and is available in three different editions:

-**MediaStore Archive**

-**MediaStore Broadcast**

-**MediaStore Enterprise**

**MediaStore** is essentially a reliable database specifically addressed to the management of companies' digital assets: video, audio, images and documents.

With **MediaStore** it is possible to ingest with the desired format, catalogue the contents, archive and deliver them when needed.

**MediaStore** supports many languages, including Chinese, Russian and Arabic.

**MediaStore** is based on Microsoft SQL Server as it is written in C #, .NET and XML/UNICODE.

Server modules in **MediaStore** consist in a Windows® background service responsible for the execution of the tasks and an application for configuration, administration, monitoring and maintenance of the system.

Each change to the system's configuration is real-time stored into the database and made available to the on-air system.

The *Workflow* manager in **MediaStore** can automate import/export operations; it can send and receive *assets* of heterogeneous formats from multiple sources (**MediaRec**, *watchfolders*, third party *ingestions*, etc.) towards different destinations (*web*, *mobile*, *playout*).

Client module in **MediaStore** consists in the management of the broadcaster contents archive. It allows to search through the archive (Simple, Full Text and Boolean search engine), to edit and add metadata, to index and categorize contents, to edit and cut them.

## MAM MediaStore Enterprise

**MediaStore** is also composed by several other features which can be integrated to the **MediaStore Broadcast** configuration to fit the best solution based on customer's requirements, with a list of other elements that can be added and integrated to complete the system.

The additional options that the customer will be available choosing our **MediaStore Enterprise** solution are:

### **MAM MediaStore Enterprise Key-Features:**

- **Import from P2 and XD-Cam**
- **Export to FTP servers, local or remote folders, youtube**
- **Base-band ingest (VTR, satellites, live feeds..)**
- **Basket creation and EDL export to editing stations (NLE)**
- **Integrated proxy-editing and final rendering**
- **Edit-while-ingest and play-while-ingest**

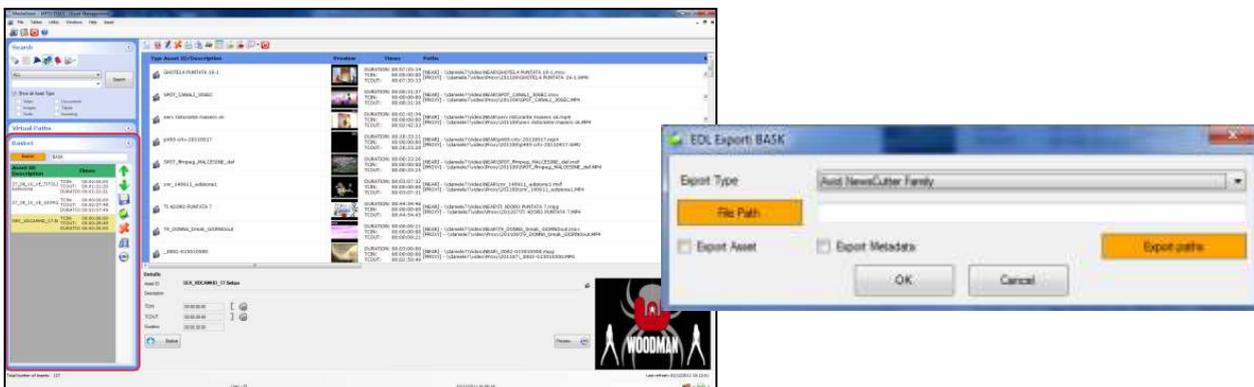
All the other features available for **MediaStore Archive** and **MediaStore Broadcast** are included in **MediaStore Enterprise**, both server and client application modules.

- Import from P2 or XDCam as an import option, with a direct link from the client interface



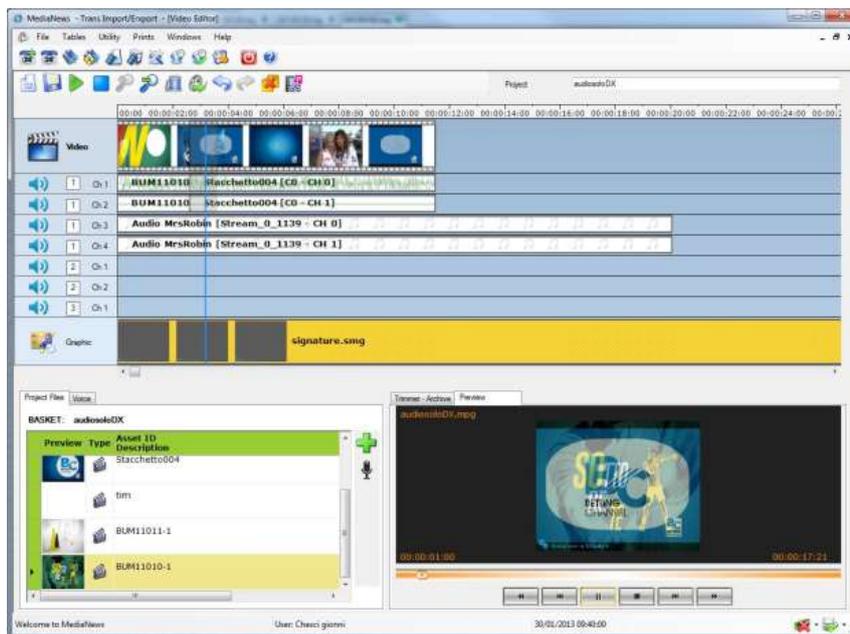
(P2 import clips list)

- Export to FTP servers, local or remote folders
- Implementation of the Base-Band ingest from VTR, satellites, live feed... through **MediaRec** (our software to manage the ingestion of new contents)
- Basket creation and EDL export to editing stations (NLE). Browsing contents through the MAM archive often leads to the creation of a new *asset*. This new contents(baskets) can be realized by assembling and elaborating existent assets or clips of them (archive material) along to newly ingested material. The basket can be exported in a format called *EDL: Editing Decision List*, but the client module supports many editing applications like Avid Media Composer, Apple Final Cut Pro and Express, Adobe Premiere CS4 e 5, Sony Vegas Pro and Grass Valley Edius. When exporting EDL file, it's possible to export also the contents directly to the editing station, along with an XML metadata file. All these operations are done automatically by **MediaStore**.



(Basket creation and export)

- Integrated proxy-editing to create and edit baskets and transcode them into new assets. It allows to work with video and audio tracks, add effects and transitions like fade and crossfade or insert graphics leveraging **MediaCG** engine and to execute the final operation of rendering, execute by **MediaStore** that will generate a new video file to be used.



(Video editing module)

- Edit-while-ingest, that has to be supported by the NLE system in use by the customer, that enables to mark and in general edit an asset while the ingestion session is still open, and play-while-ingest, that has to be supported by the Play Out system in use and enables to put an asset on air while the ingestion session is not closed yet.

## **Server application module:**

**MediaStore Archive** is an application suite for the management of the media *assets* life cycle. Assets types supported are: video files, audio files, images, analogue or digital tapes and documents.

**MediaStore** manages *assets* creation (*ingest or content creation*), transfers across different levels of storage, deletion and delivery to final destinations (*web, mobile, playout*) or archiving.

At the time of creation, the new *asset* is identified and analysed; the first set of metadata is automatically inserted as soon as the content enters the system.

According to the user *workflow* the *asset* can be archived or moved to other storage devices, or it can be processed (categorization and cataloguing, low-resolution (i.e. proxy) creation and/or management, speech-to-text detection, etc.), according to the customer configuration.

At every moment, following the rules defined into the integrated *workflow* manager, **MediaStore** can move the *assets* to *OnLine Archives* or *OffLine Archives*, as needed, automatically.

This rule-based automation frees operators from any manual activity.

The *Workflow* manager can automate import from FTP sources and watch-folders.

The configuration is dynamic and can be adapted at any time to the evolving situation and needs. **MediaStore Archive** is designed to be *cooperative* and *multi-threading*; new servers can be added at any time to scale up the system elaboration power in order to fairly distribute the workload between them.

**MediaStore Archive** consists of two modules: **MediaStore** and **MediaStore Manager**. The first one is a Windows® background service responsible for the execution of the tasks. The second one is a client application for configuration, administration, monitoring and maintenance of the system.

Each change to the system's configuration is real-time stored into the database and can be made available immediately to the on-air system.



## Retrieving of main metadata of incoming assets

With **MediaStore Archive** it is possible to retrieve the main metadata of each asset that enters the system both coming from FTP or from watch-folders.

It means that this information will remain available in the system, visible and editable.

- **Wrapper** = MXF\_OP1a
- **Content** = Mpeg2
- **Video Size** = 720x576
- **Field Order** = InterlacedTopBottom
- **Video Frame Rate** = 25
- **Video format** = PAL
- **Aspect-Ratio** = Ratio 4\_3
- **Stream Count** = 2
- **External Essences** = False
- **Dropped Frame** = False
- **StreamId** = 05010100
- **Stream Type** = Video
- **Data-Rate** = 25
- **Time Scale** = 1
- **Frame Rate** = 25
- **Bit per Frame** = 0
- **Frame start** = 0 (00:00:00:00)
- **Frame count** = 185 (00:00:07:10)
- **Properties** =
  - **HOffset** -> 0
  - **HLen** -> 12288
  - **FOffset** -> 39415808
  - **FLen** -> 124
  - **IdxEntries** ->
  - **IdxUnitSize** -> 212992
- **StreamId** = 06011000
- **Stream Type** = Audio
- **Audio Format** = Unknown
- **Data Rate** = 48000
- **Time Scale** = 1
- **Sample Rate** = 48000
- **Bps** = 0 (0 Kbps)
- **Bit depth** = 0
- **Channel count** = 4

Example of retrieving of the metadata of an MXF file

## Automated management for Hashing, Metadata and *Filling*

Each new *asset* that has to be added to the archive is elaborated by **MediaStore**.

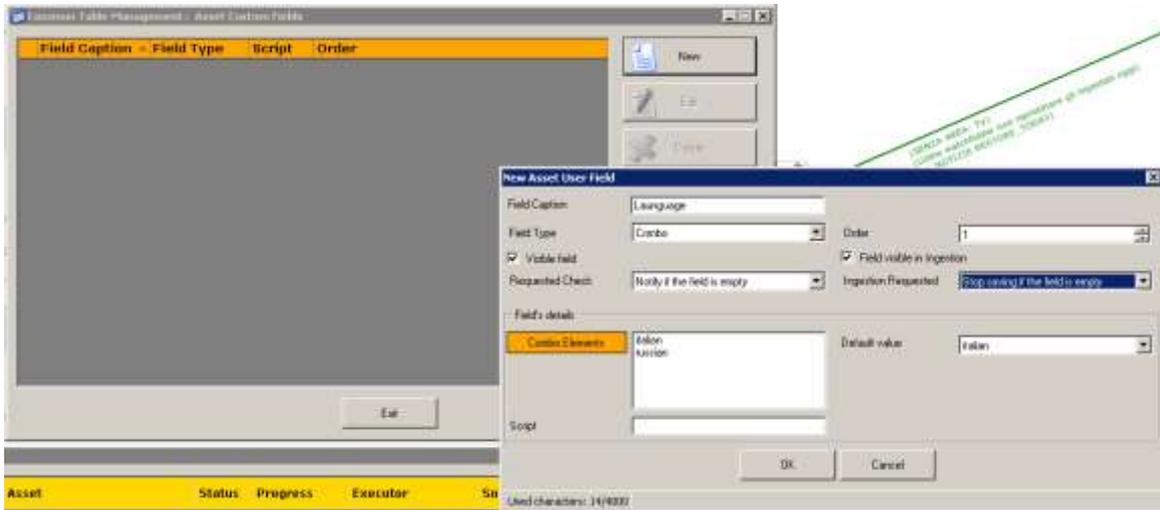
First step is to retrieve available metadata (descriptive information) like name, duration, *aspect ratio*, file size, etc. If the file contains an essence like MXF, GXF, MOV, IMX, etc., the internal available metadata are retrieved too (*filling*). This step is very quick and includes also the so called *hashing* (or *digital signature*) of the *asset*; the asset is so linked to a univocal code which allows comparing it to the archive contents in order to identify and avoid duplicates.

This feature allows an effective usage of the storage space.

Once checked the *hash*, the *asset* is saved into the database and ready to be used and transferred to any storage device of the system.

All these elaborations run in a separate thread, without interfering with the other running tasks of **MediaStore**.

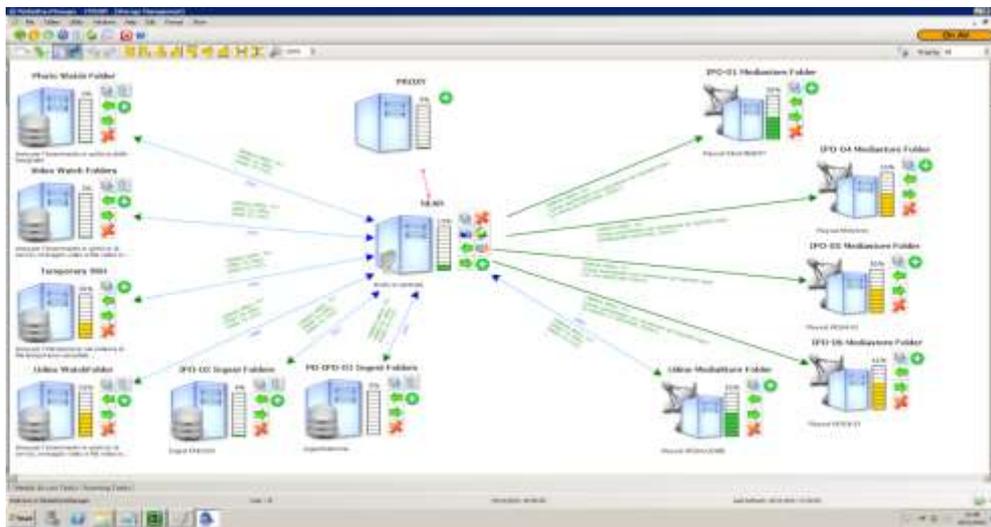
**MediaStore** provides the operator with an initial set of fields for categorization, but it's possible to add new custom fields. These custom fields can be mandatory, predefined with default values or calculated by scripts. They can be filled during the ingestion phase.



(Customization of asset Metadata fields)

## Workflow management and integrated monitoring

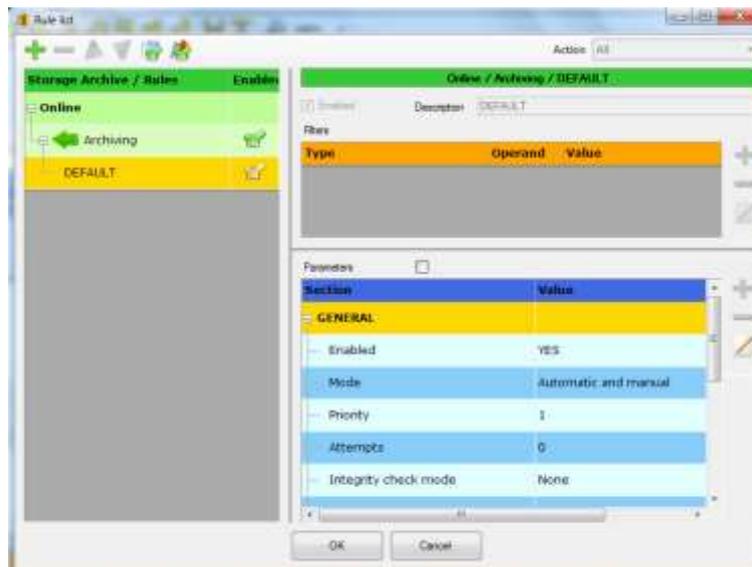
**MediaStore Archive** integrates a complete *workflow* manager. It is a design tool to set up the global architecture of the system and its behaviour. It also integrates a set of monitoring tools to survey the system status and warn administrators about conditions that require their attention. On the same window are located panels to display the running and waiting tasks executed by the installed instances of **MediaStore**. **MediaStore Manager** main console is intended to be the place where system administrators set up the system first and then maintain it.



(Workflow example of a non-trivial system)

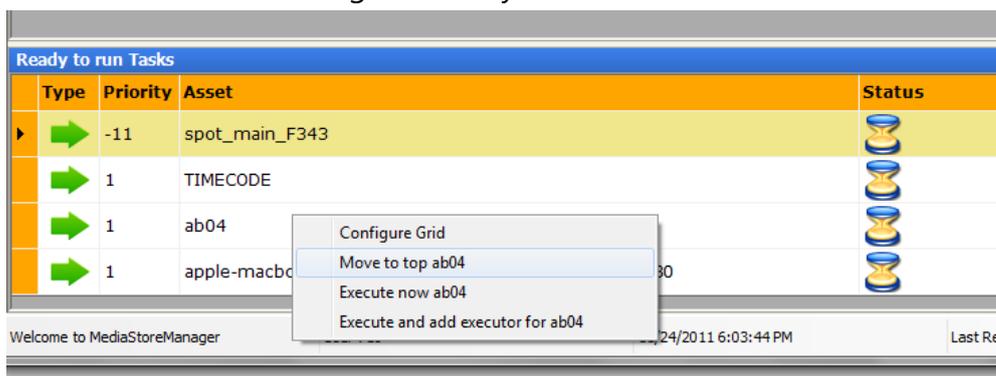
The first step is the storage device representation, in order to get a visual representation as in the above picture. Next step is to assign each one its behaviour, tuning the rules that will define the workflow.

Every rule can be customized: deletion, archiving, restoring, transcoding, etc. and to every action can be assigned a priority.



(Example of the definition of a rule)

These rules will lead to different behaviours which can be tested directly inside the workflow editor, without affecting the live system.



## Hierarchical storage management

**MediaStore Archive** is a full featured *HSM* (Hierarchical Storage Management). Each storage device that is controlled by **MediaStore** can be categorized as *OnLine*, *NearLine* or *OffLine* archive.

*OnLine* storages are typically playout machines, video servers or ingest workstations.

They usually feature fast and efficient disks, but with high per-byte cost. In order to provide more storage space is common to have a central archive, called *NearLine*, with higher capacity but lower performances and costs. *NearLine* usually acts as a cache memory for the *OnLine* storages.

The *OnLine*'s contents that are not needed anymore for playout are *archived* into the *NearLine*, and restored when needed. To this scenario is also possible to add the lowest level of the hierarchical storage, the *OffLine*. This storage has lower performances than

other levels, but much more capacity with minimum per-byte cost. *OffLine* are usually LTO tape-based archives.

Those three hierarchical levels are the standard configuration in a broadcast environment.

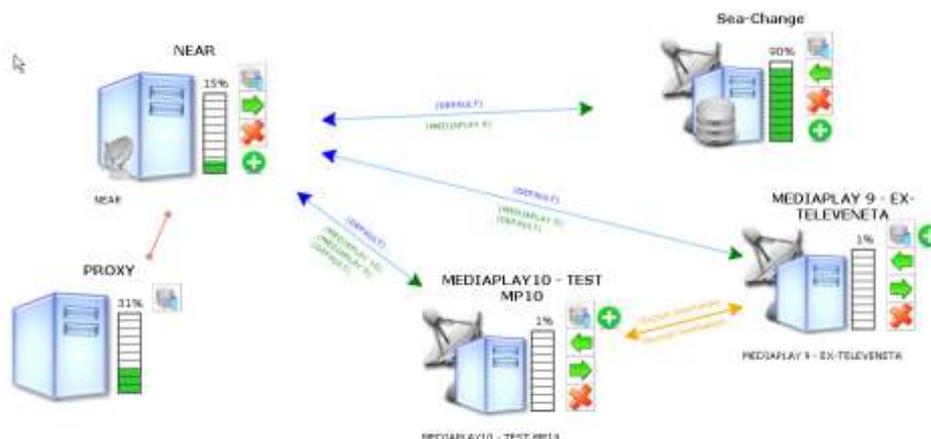


(Example of an HSM set up)

With **MediaStore** the HSM can be defined in a flexible and dynamic way, and its management is fully automated.

**MediaStore** will take care to move the contents according to the workflow needs. This configuration allows to store a very large number of files, without the need to delete anything, saving future re-digitalization of old deleted material.

**MediaStore** fully implements the HSM paradigm and extends it, allowing the broadcaster to customize every single step thanks to the *workflow* manager.



(Customized HSM set up, with heterogeneous storages)

## Import

*Import* operations aims to add new contents into the system. These assets can be acquired along with their metadata, and **MediaStore** will take care of storing them and associate them to the content (*filling*).

Once imported, this new contents can be transcoded (*normalized*) to conform to the other MAM contents.

With **MediaStore Archive** it is possible to import contents from FTP servers and watchfolders.

Automated *import* operations (the ones defined in the *workflow* set up) are directly managed by **MediaStore** and require no operator actions.

## Integration with all SI Media applications and 3<sup>rd</sup> party applications

**MediaStore** is a complete standalone platform, but expresses its full potential when joined with the other SI Media modules, like the *Newsroom* (MediaNews and MediaNewsPlay) and the *Automation* (MediaList and MediaPlay). The level of integration between these three modules is as strong as if they were one.

Both for *traffic* and *newsroom* clients. The communication across different applications is delegated to the centralized database.

MediaStore can be integrated in a unique flow with 3<sup>rd</sup> party automations, traffic systems or NRCS.

## Integration with all major IT vendors and Broadcast brands

**MediaStore** not only runs on IT platforms like IBM, HP, Dell and Intel,(with AJA, Black Magic...), but also with all major broadcast brands like Omneon (Harmonic) , Xor-Media, GrassValley, Harris and EVS, Ross Video, Quantel.



(Some of the worldwide adopted brands supported by MediaStore)

## **Client application module:**

**MediaStore Client** is the application module for the management of the broadcaster contents archive. User can acquire new *assets*, search through the archive, index, tag and (if required by the customer) export to editing stations (NLE or *Non Linear Editing System*) Transcoders or other devices.

**MediaStore Client** features a set of tools for advanced search and preview of contents archive, also using low-res copies of the contents, managing Proxy-based media asset browsing and editing.

It allows a detailed indexing and categorization of contents, both using the capabilities of **MediaStore Client**'s search engine (Simple – Like Style, Full-Text - Google Style, Advanced – Boolean Style) and virtual path management in order to customize the organization of the contents available in the archive.

The metadata schema is flexible and fully customizable: the customer can define the fields with the metadata he's interested in and set his own configuration.

**MediaStore Client** can be integrated with **MediaNews** (NRCS for *newsroom*) and **MediaList** (*traffic* for automation), for centralizing the operator's job in a single machine.

For example a journalist will be able to cut and edit clips and add its own voiceover audio, then create a clip ready for broadcast, with all the functions integrated in the the same machine.

### Key Features:

- Detailed indexing and categorization of contents
- Full Text contents search
- Customizable metadata scheme
- Virtual and Multi-Virtual Path management
- Proxy-based media asset browsing and editing
- Other special functions
- Integration with MediaNews and MediaList



### Detailed indexing and categorization of contents

Content indexing is one of the main goals of **MediaStore Client**. Every detail associated with the content can be stored and successively retrieved from the archive.

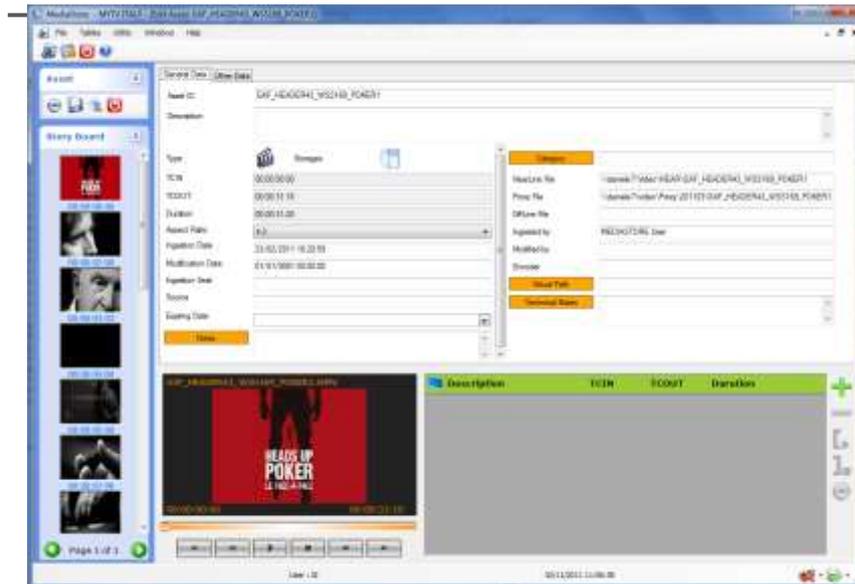
There are two kinds of indexing: automatic or manual.

Automatic indexing is done by **MediaStore Server** during the so called *filling* phase that aims to gather as much as possible *asset* related data.

Usually retrieved metadata are: file name, time length, file size, video source, encoder, operator name, date and time. Other type-specific information can be retrieved, like aspect ratio, video format, bitrate, etc...

This set of data can satisfy the needs for small archives, but when the size grows it's required a deeper level of categorization, that's where the operator manually customizes metadata. Thanks to the **MediaStore Client** graphical interface the operator can insert every kind of metadata, without restriction on number or dimension. The operator can also add, at any time, new categorization fields for better definition of the *asset* to be ingested.

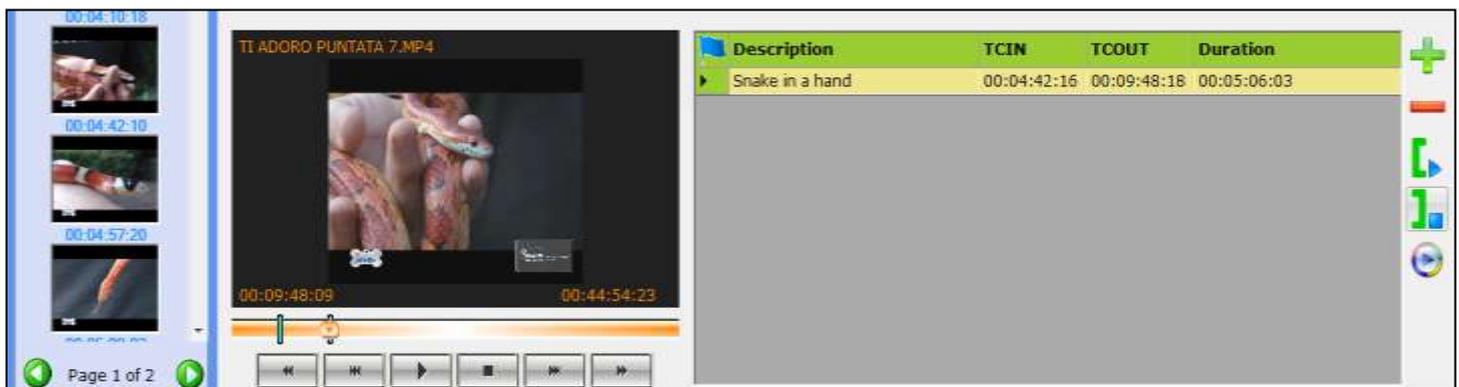
Every custom field can have the mandatory attribute, that prevent the *asset* to be saved if the fields it's leaved empty.



(Asset metadata editing form)

When video file content is heterogeneous, like a single file that contains different topics, is possible to mark a subclip, and associate its own metadata.

These sub clips are called *bookmarks*. These bookmarks are just a logical mark of the original video, but they can be cut and become a new asset, this through the feature called *Consolidation*. Thanks to the *Consolidation*, the original physical asset will be split in as many new physical assets as the bookmarks are.



(Bookmark definition form)

## Full Text contents search

One of the key feature of **MediaStore Client** is to provide several search types, each suitable for the different categorization of the contents archive. The goal is to find quickly the desired content. There are three types of search:

Simple search (*Like style*), *Full-Text* search (*Google style*) and *Advanced Full-Text (Boolean Style)*. Each one of this search style allows to select a subset of fields where the search will be applied.

- **Simple search (Like Style)**

This search style looks for a specific string in all asset text fields. It allows to use the wildcard character '%'. Here are some examples:

- Search term: "**The**" or "**The%**" looks for all *assets* whose metadata contain a word starting with "**The**".
- Search term: "**%The**" or "**%The%**" looks for all *assets* whose metadata contain a word with the substring "**The**".
- Search term: "**%Ab%Cd**" looks for all *assets* whose metadata contains the two sub string "**Ab**" and "**Cd**" in that order.

This search style is particularly useful when applied to a single metadata field.

- **FullText search (Google Style)**

This kind of search focuses on the whole word instead of substring of it. Like for Google search engine, the operator will look for the words he wants to find, and more words will be looking for, the better the results. *Full-Text* search can also take into account synonyms and antonyms.

This is the most used search style, and the one that provide the most accurate results accuracy.

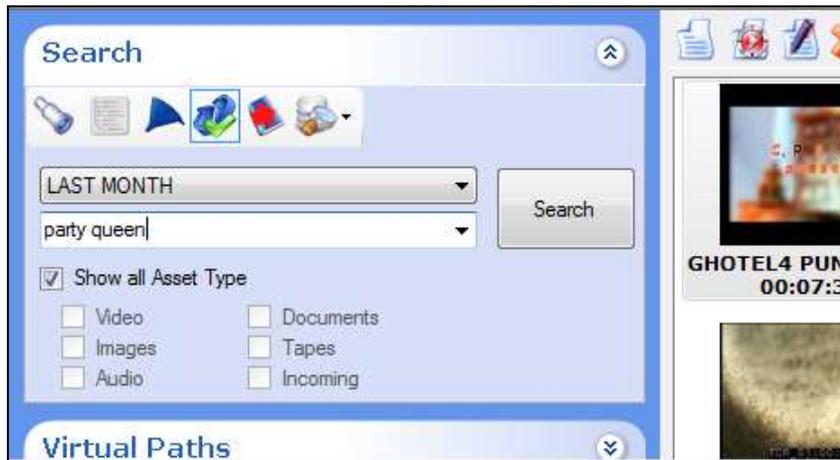
- **Advanced Full-Text search (Boolean Style)**

This search style is much comparable to the previous *Full-Text* search, but provide a way to combine searched terms with the boolean operators (AND, OR, NOT), expressing more complex search rules. For example:

- Search term: "**fruit NOT apple**" will be looking for all *assets* containing the word "**fruit**", except the ones containing also the word "**apple**".

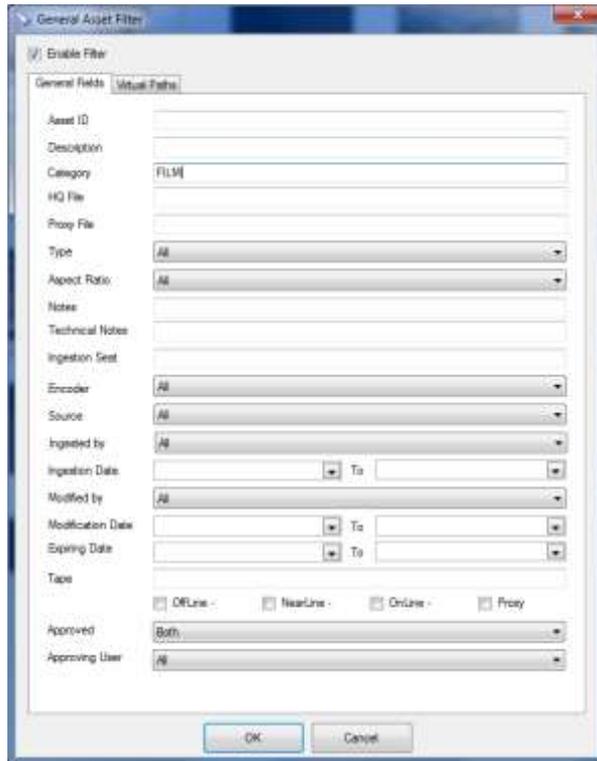
- Search term: “**apple OR pear**” will be looking for all *assets* with the word “**apple**” or the word “**pear**” or both of them.
- Search term: “**apple AND pear**” will be looking for all *assets* containing both the word “**apple**” and the word “**pear**”.

This is the most expressive search style, but requires more skill by the operator in the creation of the search term.



(Detail of a search form of MediaStore)

Besides these three search types, **MediaStore Client** also provides the ability to search over specific fields, thanks to the “General Asset Filter” form. This form can also be used to achieve a pre-filter, on the result of which make a second search. For example it’s possible to select the “FILM” category as a pre-filter and to search on this subset of results with the previously mentioned search styles.



(General Asset Filter form of MediaStore)

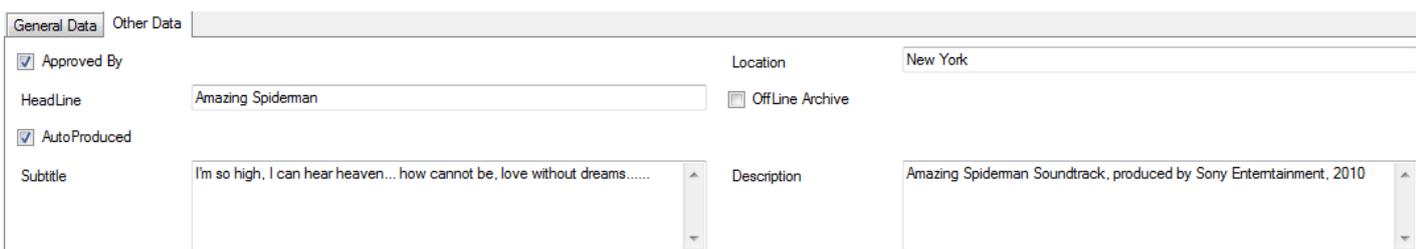
It is also possible to save the created filters and recall them later in order to reduce the time of the search.

### Customizable metadata scheme

As previously introduced, it is possible to add any additional asset field, in order to allow the operator to manage the metadata scheme in a fully customized way.

This custom data can be used also to search, filtering the contents with these fields.

Every custom field can have the mandatory attribute, that prevents the *asset* to be saved if the fields it's leaved empty.



(Example of custom fields in MediaStore)

## *Virtual and Multi-Virtual Path management*

**MediaStore Client** provides the management and the search of the *Virtual Paths* and *Multi-Virtual Paths*.

The *Virtual Path* is a tree structure (created and managed by the broadcaster to suit its archive strategy) which aims to support indexing and categorization of the contents archive.

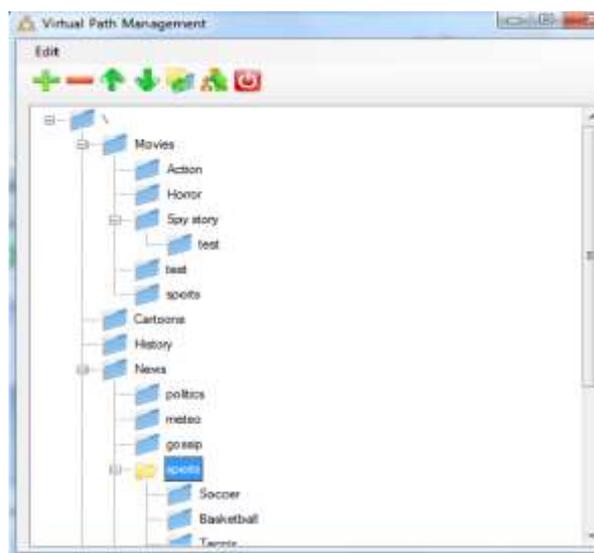
It allows to manage the assets categorizing them in a tree structure, to narrow down searches at certain branches of the logical tree.

The customer can at any moment increase the structure with new branches at the same level or creating lower levels.

With multi-virtual path management it is also possible to assign more paths to the same asset (for example the same football clip can be indexed as sports and news at the same time).

This tool can be used to empower the search engine (searching directly through virtual folders) of **MediaStore Client** and to keep organized the asset archive.

**MediaStore Client** allows to create a virtual path starting from the real paths where the contents are situated in the system.



(Example of virtual path management in MediaStore)

### *Proxy-based media asset browsing and editing*

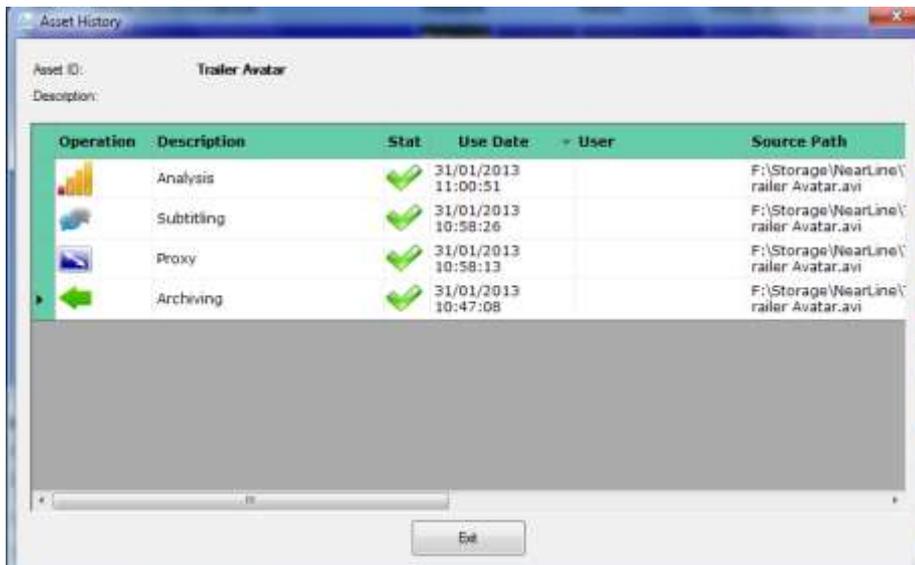
**MediaStore** can manage 3<sup>rd</sup> party proxy-based media asset for browsing and editing. It means that the operator can see the preview of each content available in the system in low-res quality, and use it to edit it, to create subclips and bookmarks from each client workstation that has access to the path where the proxy copies are situated in the system.

### **Other special functions**

**MediaStore Client** also provides possibility to set a manual import from a local physical path. It allows for example to a journalist to directly import contents from his laptop to the central archive in an easier way.

Besides for each content of the asset archive MediaStore Client keeps all the information regarding the history of each asset.

It means that the operator can check the status of each task, the date and the user that executed any of these tasks.



Operation	Description	Stat	Use Date	User	Source Path
	Analysis		31/01/2013 11:00:51		F:\Storage\NearLine\railer Avatar.avi
	Subtitling		31/01/2013 10:58:26		F:\Storage\NearLine\railer Avatar.avi
	Proxy		31/01/2013 10:58:13		F:\Storage\NearLine\railer Avatar.avi
	Archiving		31/01/2013 10:47:08		F:\Storage\NearLine\railer Avatar.avi

(Example of History Asset in MediaStore)

## Integration with MediaNews and MediaList

There is a deep integration between **MediaStore Client** and **MediaNews** (our NRCS software) and **MediaList** (our traffic software): from each of these two modules the operator can access the features of **MediaStore Client** thanks to the embedded GUIs.

### Other features:

**MediaStore Broadcast** provides also other features:

- cooperation and multi-threading
- file signature technology
- fault tolerance with main/backup solution

## Cooperation and Multi-threading

**MediaStore Broadcast** is a *Multi-threading* and *Cooperative* platform.

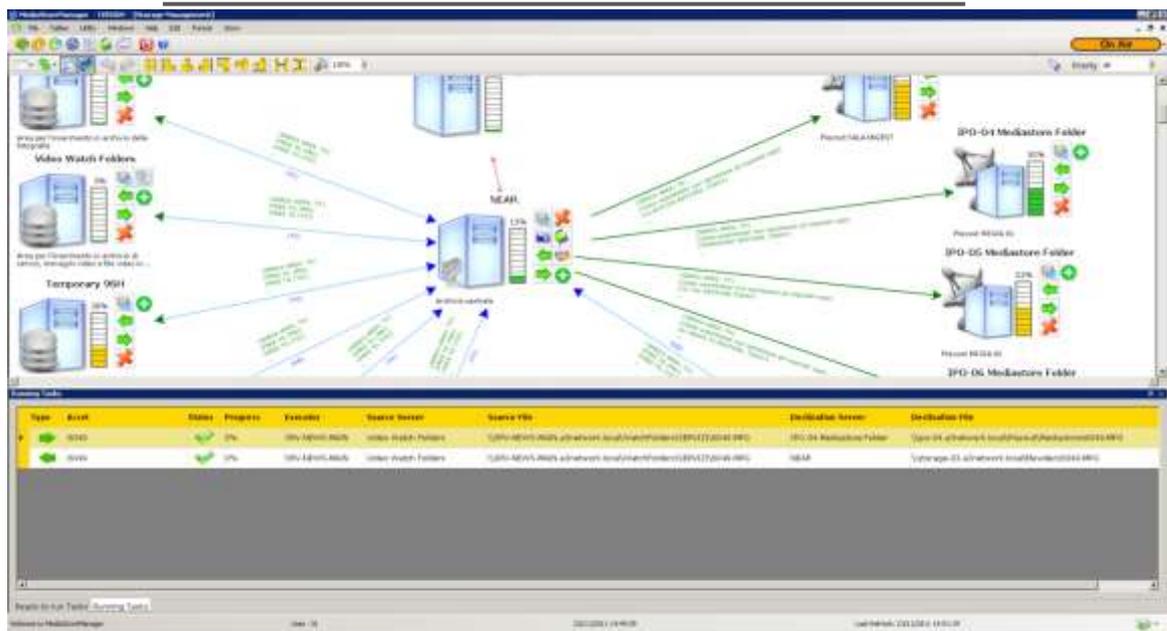
*Multi-threading* stands for multiple tasks executed simultaneously. For example the system can copy several assets while deleting some others. This feature avoid performance bottleneck when multiple requests come to a server. The concurrent operations number can be tuned to the hardware performance or to other constraints.

*Cooperative* means that more than one server (*instance*) can run **MediaStore**.

Multiple servers can be configured to execute the same kind of task (copies, proxy generation, transcoding, etc...). Usually assets transfers and import/export are handled by more than one server.

These **MediaStore** instances will share the incoming requests in order to cut down execution time. This kind of *load balancing* increases reliability, providing a sort of *active/active backup*.

If one of the servers goes down, its tasks will be taken on charge by other running servers.



(MediaStore: main monitoring window with running tasks)

Type	Asset	Status	Priority	Progress	Executor	Source Server	Destination Server
	BULSATCOM MALAYSIA 16x9nev	▶	2	64%	STEFANOV		
	progam_italy_2011-04-01_epic	▶	20	50%	STEFANOV	Near	TV backup
	Copy (103) of Copy of Copy of f	▶	1	0%	STEFANOV	TV backup	
	MultiAudio_V0	✓	3		STEFANOV	Proxy	

(MediaStore: running processes)

## File signature technology

**MediaStore Broadcast** provides also a content management that consists in a file signature technology in order to prevent repeated archiving of the same contents and not have a waste of space in the system.

## Fault tolerance with main/backup solution

The system can be configured to be fully fault tolerance, with main / backup solution to ensure the continuous functioning of the system.

## MAM MediaStore Enterprise Key-Features:

- **Import from P2 and XD-Cam**
- **Export to FTP servers, local or remote folders, youtube**
- **Base-band ingest (VTR, satellites, live feeds..)**
- **Edit-while-ingest and play-while-ingest**
- **Basket creation and EDL export to editing stations (NLE)**
- **Integrated proxy-editing and final rendering**

### Import from P2 and XD-Cam

With **MediaStore Enterprise** it is possible to configure a direct import from P2 and XD-CAM.

When contents are imported from P2 or XDCAM material, they're already marked, so there is no need to *consolidate* them, since it is possible to select the clips before importing into archive.

**MediaStore Client** fully supports P2 and XDCAM material. There is a dedicated import procedure that allows operator to preview clips and grab metadata.

Sel.	Name	Importing Details	Preview	Times	File
<input checked="" type="checkbox"/>	0001RY	P2 Import 2 Duration :00:00:15:14 Aspect Ratio :16:9		Start: 16/12/2010 13:16:01 End: 16/12/2010 13:16:01	Video: \\stefanov\E\zz-VideoRe Audio: \\stefanov\E\zz-VideoRe Audio: \\stefanov\E\zz-VideoRe Audio: \\stefanov\E\zz-VideoRe Audio: \\stefanov\E\zz-VideoRe
<input type="checkbox"/>	0001RZ	P2 Import 2 Duration :00:00:15:14 Aspect Ratio :4:3		Start: 16/12/2010 13:06:31 End: 16/12/2010 13:06:31	Video: \\stefanov\E\zz-VideoRe Audio: \\stefanov\E\zz-VideoRe Audio: \\stefanov\E\zz-VideoRe Audio: \\stefanov\E\zz-VideoRe Audio: \\stefanov\E\zz-VideoRe

(P2 import clips list)

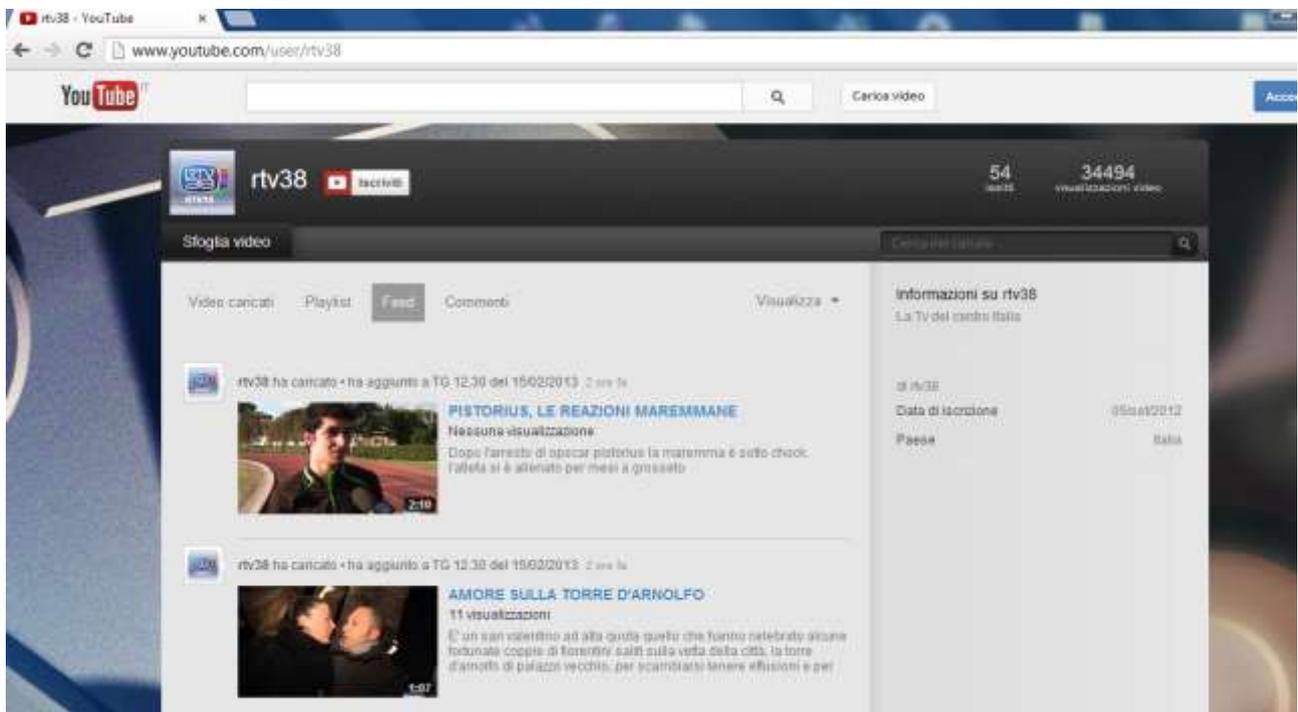
## Export to FTP servers, local or remote folders, youtube

**MediaStore Enterprise** provides also the possibility to set and configure an automatic export of contents or baskets.

Once the operator has set an export destination path for these contents, (that can be an ftp server, a local or remote folder or external providers such as youtube), defining the rule in **MediaStore Manager**, all these tasks will be automatically executed without any additional job for the operator.

Defining the rule, the operator not only can choose the destination of each export, but also the desired format (that can be different from destination to destination) in order to set an automatic transcoding of the content he wants to export.

Regarding NRCS, **MediaStore Enterprise** allows to export not only the clips, but also all the metadata regarding a story (title, details, summary..) or a rundown, through an xml data exchange.



(Example of a youtube channel, Rtv38-SIMedia's customer)

## Base-band ingest (VTR, satellites, live feeds..)

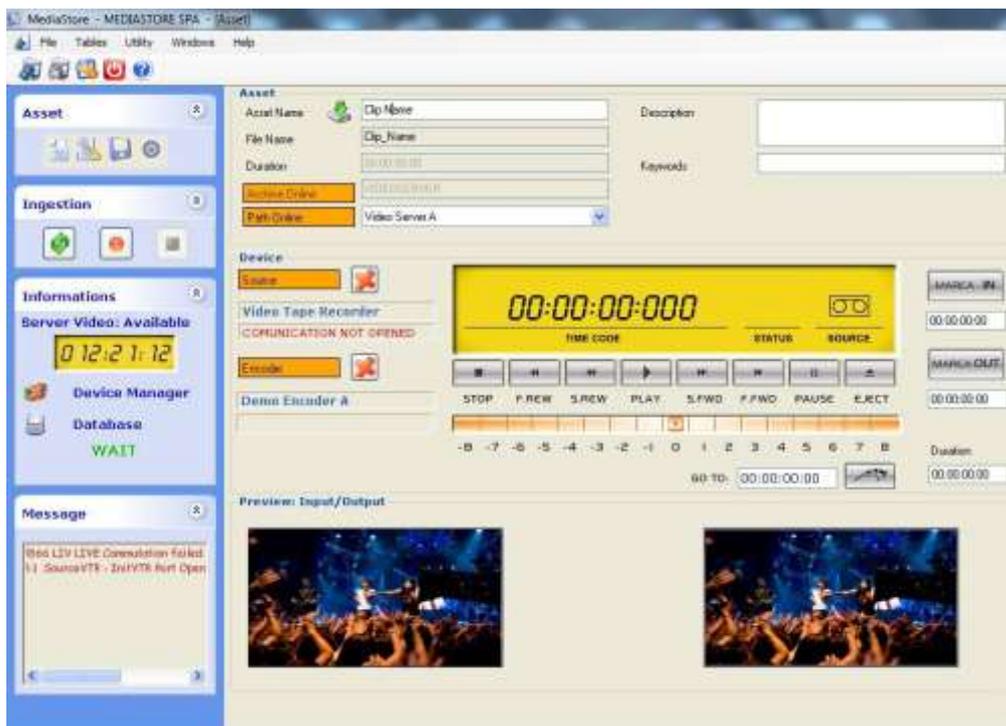
**MediaStore Enterprise** integrates all the ingest features of the **MediaRec** module.

**MediaStore Enterprise** provides both manual and automatic acquisition sessions, from every video source, like *live feed*, VTR, satellite, studio, etc.

With **MediaStore Enterprise** the operator can continuously check available space for new acquisitions and create automatic recording session which will start at planned time.

Using the same ingest window it's possible to edit the new *asset* metadata and preview video signal and audio levels.

**MediaStore Enterprise** can dynamically control the available *encoders* (local or remote) and acquire simultaneously the video and the low resolution (*proxy*) signal. It can also control remotely external devices like servers, VTR or routers for the automatic assignment of the correct video source related to the selected *encoder*.



(Ingestion window)

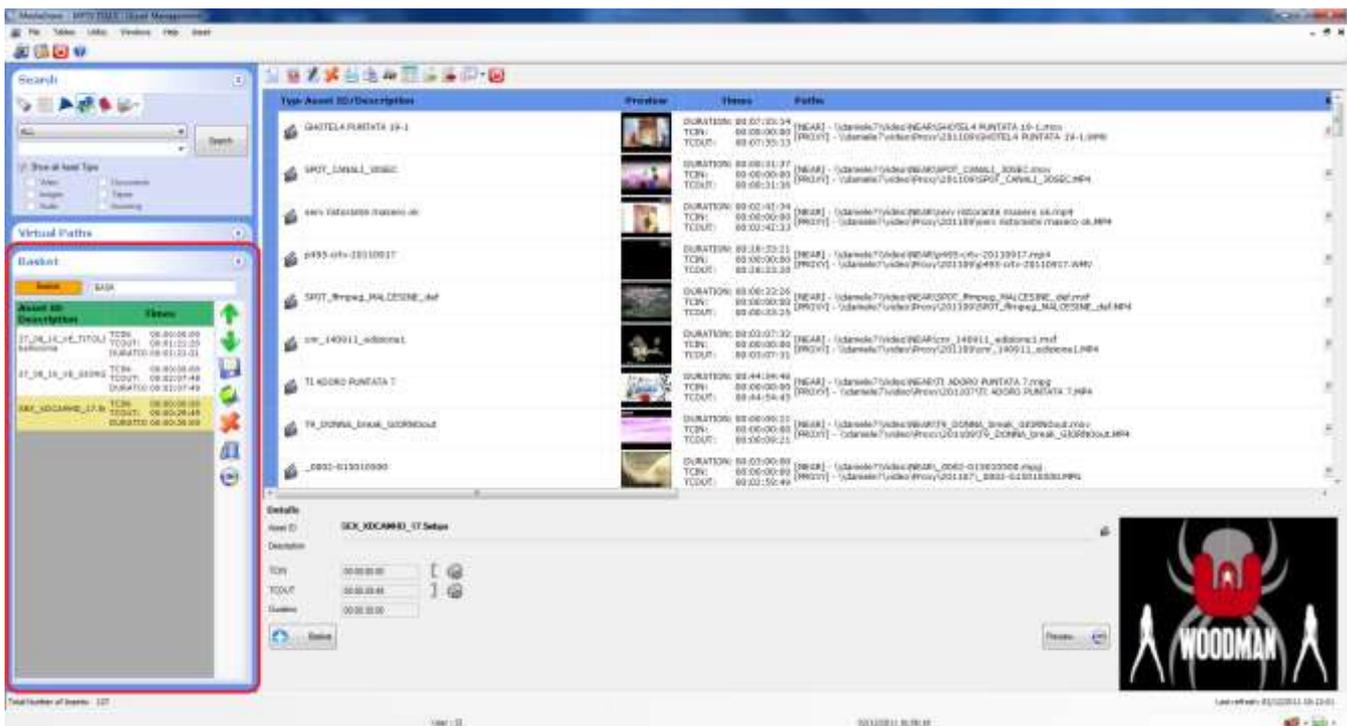
With **MediaRec** it is possible to edit the clip during the ingestion session and also to put the content on air while the ingestion session is not closed yet (Edit-while-ingest, Play-while-ingest). This solution is always available using AJA boards and/or video servers that support this function.

## Basket creation and EDL export to editing stations (NLE)

Browsing contents through the MAM archive often leads to the creation of a new *asset*. This new content can be realized by assembling and elaborating existent assets (archive material) along to newly ingested material.

For that purpose the journalist who wants to create a news report using **MediaStore Enterprise** will be browsing archive *assets*, look for the ones desired thanks to the search features provided by the application; once he found what's looking for, he can put everything in a *basket*.

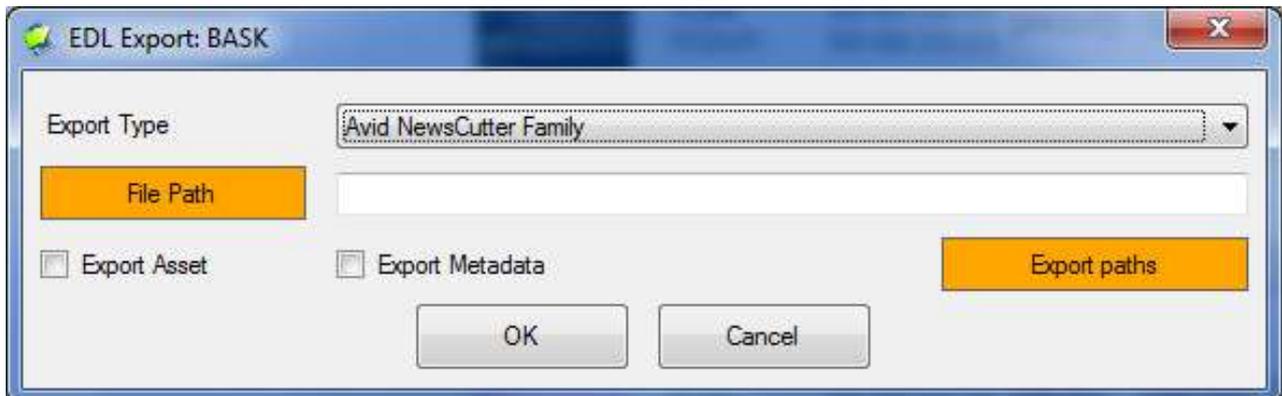
This *basket* can contain assets or just clips of them. The following picture shows in red the *basket* area. Adding an *asset* to the basket is as simple as drag and dropping the asset onto the *basket* area.



(The basket area framed in red)

Once defined a *basket* the journalist can proceed to the editing. There are two way at this point: use the integrated proxy-editing module of **MediaStore Enterprise**, or export the *basket* content to the editing station.

The basket will be exported in a format called *EDL: Editing Decision List*. **MediaStore Enterprise** supports many editing applications like Avid Media Composer, Apple Final Cut Pro and Express, Adobe Premiere CS4 e 5, Sony Vegas Pro and Grass Valley Edius.



(EDL export form)

When exporting EDL file, it's possible to export also the contents directly to the editing station, along with an XML metadata file. All this operations are done automatically by **MediaStore**. At this point the operator can import the EDL file in his editing application and will see the timeline exactly as defined in the *basket*.

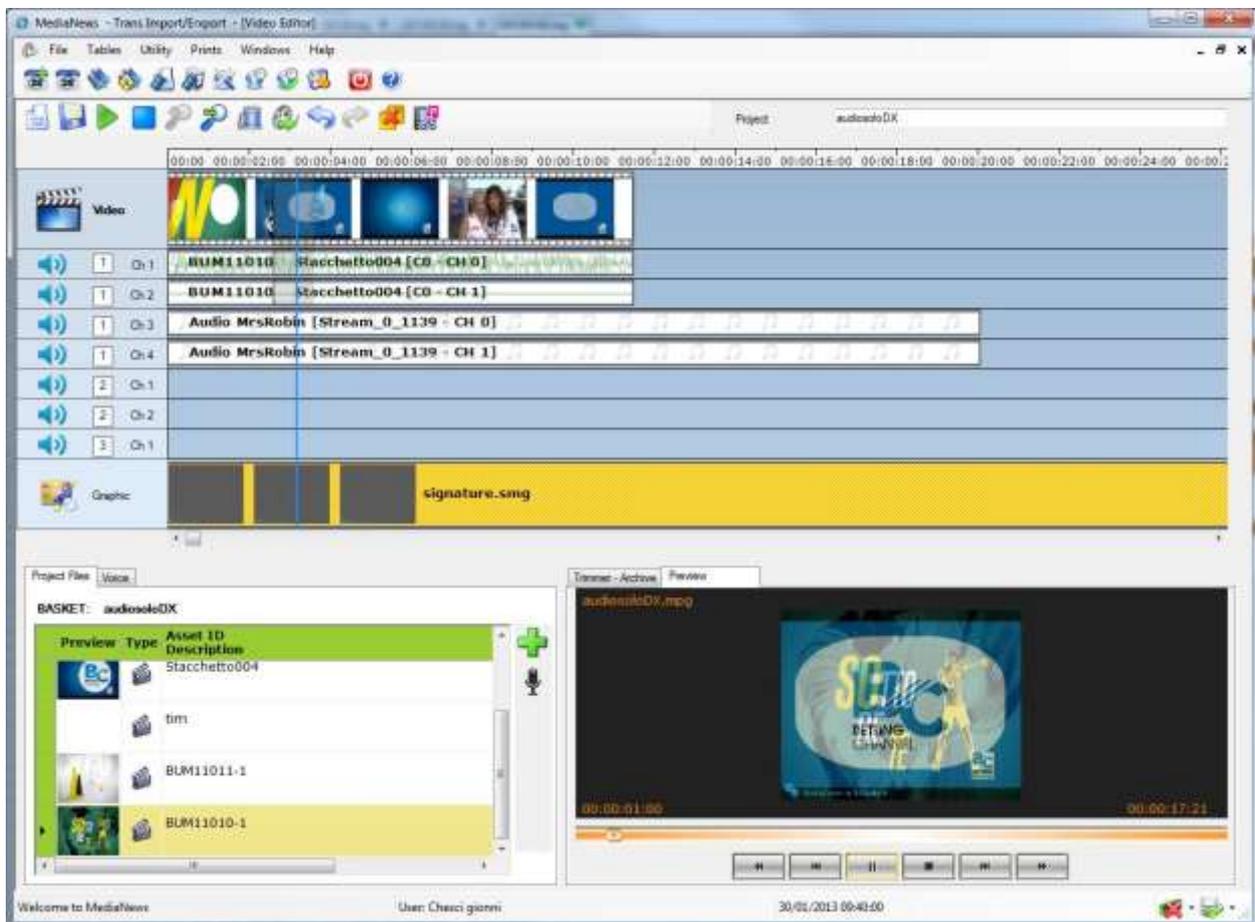
## Integrated proxy-editing and final rendering

The alternative to an external video editor is to use the integrated proxy-editing and *transcoding* engine. **MediaStore Enterprise** editing module display a *timeline* with the *assets* contained in a *basket* and provide an effective set of tools to edit the clips.

This tools allows the journalist to concatenate clips, add a voiceover audio track, untie video from audio, add effects and transitions like fade and crossfade or insert a graphic leveraging **MediaCG** engine.

It is also possible to have a preview of the project together with the CG.

The final operation is the rendering, executed by **MediaStore** that will generate a new video file ready to be used.



(Proxy-editing module)

SI Media editing module was designed in a way, to provide a clear and intuitive interface, with all the features needed by a newsroom. The live preview provides the operator a way to see at every moment its work in progress.

Output video format actually supported are: MPEG1, MPEG2, MPEG4, H-264/AVC, AVI-DV, Adobe Flash.