Automating Production of Cross Media Content for Multi-channel Distribution

www.AXMEDIS.org

AXMEDIS Content Processing GRID: Script Language User Manuals

(addition to DE5-0-1-1 AXFW User Manual)

Version: 1.1
Date: 23/09/2006
Responsible: DSI: I. Bruno, N. Mitolo (revised and approved by coordinator)

Project Number: IST-2-511299
Project Title: AXMEDIS
Deliverable Type: report
Visible to User Groups: yes
Visible to Affiliated: yes
Visible to the Public: yes

Deliverable Number: additional annex to DE5.0.1.1
Contractual Date of Delivery: M24
Actual Date of Delivery: 22/09/2006
Title of Deliverable: AXMEDIS content processing GRID, Script Language User Manual
Work-Package contributing to the Deliverable: WP5
Task contributing to the Deliverable: WP5
Nature of the Deliverable: report
Author(s): DSI and all partners involved in the AXCP content processing

Abstract: This document contains the user manuals of the major tools provided by AXMEDIS

Keyword List: MPEG-21, AXOM, transcoding, adaptation, video, image, document, ringtones, profiling, formatting, styling, SMIL, XSLT, metadata, mapping, GRID control.

Table of Content

1 Introduction

This deliverable aim is to describe the AXMEDIS Script included in the AXMEDIS tools.

This document should be consulted together with:

- AXMEDIS Framework for ALL: axmedis-de5-1-2-1-axmedis-for-all-v2-7.pdf , http://www.axmedis.org/documenti/view_documenti.php?doc_id=1728
- AXMEDIS Major Tools User Manual: axmedis-de5-0-1-1-axmedis-user-manuals-v-1-2.pdf, http://www.axmedis.org/documenti/view_documenti.php?doc_id=2347
- AXMEDIS Framework Guidelines updated version with AXMEDIS Plug in SDK axmedis-de5-1-3-1-framework-guidelines-v2-2.pdf, http://www.axmedis.org/documenti/view_documenti.php?doc_id=2197

2 AXMEDIS overview of content processing capabilities

A content factory can be built on the basis of AXMEDIS tools in a scalable and flexible manner. Also tuning for example, GRID size, database size and type, number of authoring tools, number and types of tools/algorithms and libraries for processing content, licenses, integration support based on Workflow or not, etc. This allows setting up a large range of configurations to satisfy the needs of small and large content producers, integrators, and distributors.

The **AXMEDIS Database Area** includes the AXMEDIS/MPEG-21 database model, supporting the storage and access to AXMEDIS content via a large set of metadata for each object grouped in what it is called AXInfo, and that can be customized with your needs. The database also includes produced licenses for the objects, history of performed actions on content, potentially available rights for each digital resource, models of contracts, etc. The AXInfo includes Dublin core plus descriptors and many other metadata for managing protection, lifecycle, etc. Any descriptors and metadata can be added in a flexible manner. Thus, different AXMEDIS factories may be based on different AXInfo and metadata, while automatic adapters can be defined and activated. The database area is based on a scalable database, a powerful AXMEDIS Database manager, and an effective **AXMEDIS Query Support** endowed of an easy to use user interface. The User may perform queries to search for objects and content located in the CMSs, in the local AXMEDIS database and in the virtual database comprised of the AXMEDIS content accessible/published via the P2P network of AXEPTools in the AXMEDIS B2B Network.

The **AXMEDIS Content Processing Area** (AXCP Area) is based on a GRID solution for automating all the activities to be performed for the production, and processing of content. The major tools are the **AXCP GRID Node (Engine) and AXCP Scheduler,** which are respectively the single node (computer) of the GRID and the organizer of processes on the GRID Nodes. They implement a scalable solution to process from smaller collections to huge amount of content per day, per minute. The processing algorithms can be specified in terms of script code (in Spider Monkey) allowing the manipulation of complex AXMEDIS data types and simple digital resources and content in general, and for the direct access to the AXMEDIS database and processing queries with the help of the AXMEDIS Query Support. The solution allows the writing of any kind of content processing algorithms, to activate them automatically on some query result, and these can be put in execution as independent processes on a scalable GRID for massive production and processing of digital resources in respect of the DRM.

The available data types, operators and accessible algorithms allow manipulation of any digital resources in a large number of formats. Algorithms can be defined for massive content composition (packaging, combination, etc.) and content layout formatting (synchronization, image and screen layout, from image sequence to video, etc.), content adaptation (change in resolution, subsampling, change in format, etc.), transcoding, coding, decoding, fingerprint extraction, estimation of descriptors, license adaptation and transcoding, license production and verification, etc.;

The users of the **AXMEDIS Content Processing Area** can code in terms of Java Script rules any kind of processing procedures and algorithms to manipulate/produce:

- Any digital resource:
    - such as images (more than 150 different formats), audio (more than 50 formats), video (more than 50 formats), documents (TXT, PS, HTML, PDF, RTF, DOC, etc.), multimedia (more than 20 formats including MPEG-4, HTML, LOM, etc.);
    - for their transcoding, adaptation, feature and descriptor extraction, recognition, certification, etc.;
    - with functionalities of many well know and powerful processing libraries such as: FFMPEG lib, LibSNDFile, TreTagger, DocFrac, GhostScript, XPDF, HTMLDOC, ImageMagik, MP4Box, Xerces, XALAN, CCPP, etc. (if you are interested in adding more libraries please contact AXMEDIS people);
- Packages and their composition and formatting
    - AXMEDIS objects with AXInfo Metadata and indexing,
    - MPEG-21 Objects,
    - including digital resources, metadata (e.g., Dublin Core, etc.),
    - protection information, etc.
- Protected objects and resources, managing protection information:
    - by using MPEG-21 IPMP model, and format
    - using algorithms such as encryption/decryption, scramble, compression, key production, different sizes for keys, etc.
- Licenses on the basis of the business models chosen:
    - stating grants, conditions, etc.,
    - verifying license consistency with respect to the potentially available rights, with the license in production, etc.,
    - by using formalism of MPEG-21 REL, and with OMA ODRL – MPEG-21 REL transcoding
- Automatic content and information access
    - database accesses (ODBC, etc.) with direct facilities;
    - database access by means of crawling facilities to access to a larger set of possible database models. They may contain digital content, resources, files, metadata, administrative and licensing information, etc., and can be physically located in several different computer systems and based on several different database models: ODBC, MySQL, ORACLE, MS-SQL, Tamino Lobster XML, etc., or files systems. The access to this information is performed by means of Focuseek Crawler;
    - file system and operating system access;
    - http and ftp accesses;
    - AXMEDIS database access with query support, actualization of selections, active queries, etc.
- Device capabilities format and processing facilities, to take into account for adaptation and/or processing;
- User Profile format and processing facilities, to take into account for adaptation and/or processing;
- WSDL facilities for the activation of WEB services dynamically on the basis of their definition;
- XML facilities for the application of styles and general processing;
- SMIL facilities for the application of templates and styles and processing;
- etc.

The algorithms and procedures used in the AXCP Area can be expanded by using the AXMEDIS Plug In technology that allows customizing and easily expanding the processing capabilities of the AXMEDIS GRID. Algorithms for the extraction of fingerprint, descriptors, adaptation, content processing, DRM adaptation, metadata adaptation, are built as pluggable algorithms. ***Any other library, model and format and related algorithms for their manipulation can be plugged in the AXCP in a very easy manner.***

The AXMEDIS **Workflow Management tools** include a set of micro tools and interfaces which are pervasively connected to all the AXMEDIS tools and plug-ins to allow interfacing the whole content factory to Workflow tools such as Open Flow and BizTalk. The control is performed to define AXMEDIS factory workflow policies and to manage inter-factory workflows policies.

The **AXMEDIS Editor** is the authoring tool for manually producing AXMEDIS objects when needed and for supporting the designer to create the scripts for the AXCP that could be considered macros of the AXMEDIS Editor. It is based on the AXMEDIS Object Model, called AXOM and based on MPEG-21, and all the modules and tools to manipulate and create AXMEDIS objects and related information and digital resources such as:

- resource hierarchy viewer and editor;
- visual and behavioral viewer and editor to show/manipulate visual and temporal aspects of related digital resources according to SMIL;
- metadata editor and viewer, to manipulate and view general XML metadata and specific AXInfo metadata;
- DRM viewer and editor to create and verify the licenses;
- Protection Information viewer and editor to specify, apply and browse protection aspects on the basis of the MPEG-21 IPMP format with extension of AXMEDIS;
- set of plug-ins to use algorithms for content processing as those mentioned and used in the AXCP Area mentioned above;
- set of plug-ins to allow the integration of AXMEDIS Editor within other editing and viewing applications such as: Video Editors, Image Editors, etc.;
- an interface to connect the AXMEDIS Editor with other external powerful editor tools;
- an interface with workflow (OpenFlow and BizTalk);
- set of internal viewers and players for digital resources such as document, images, video, MPEG-4, and audio resources, etc., for more than 250 different file formats.

3 AXMEDIS Framework

The AXMEDIS Framework is the set of information and tools that is at the basis of the above mentioned applications and solutions. In the next Figure, the simplified version of the AXMEDIS Framework structure is reported. It contains all the necessary tools to set up a large set of services and solutions in the area of content production, protection and distribution. The AXMEDIS Framework is an infrastructure on which several other models for content modeling, protection, production, DRM and distribution can be built in a very simple manner reusing the components and functionalities provided.

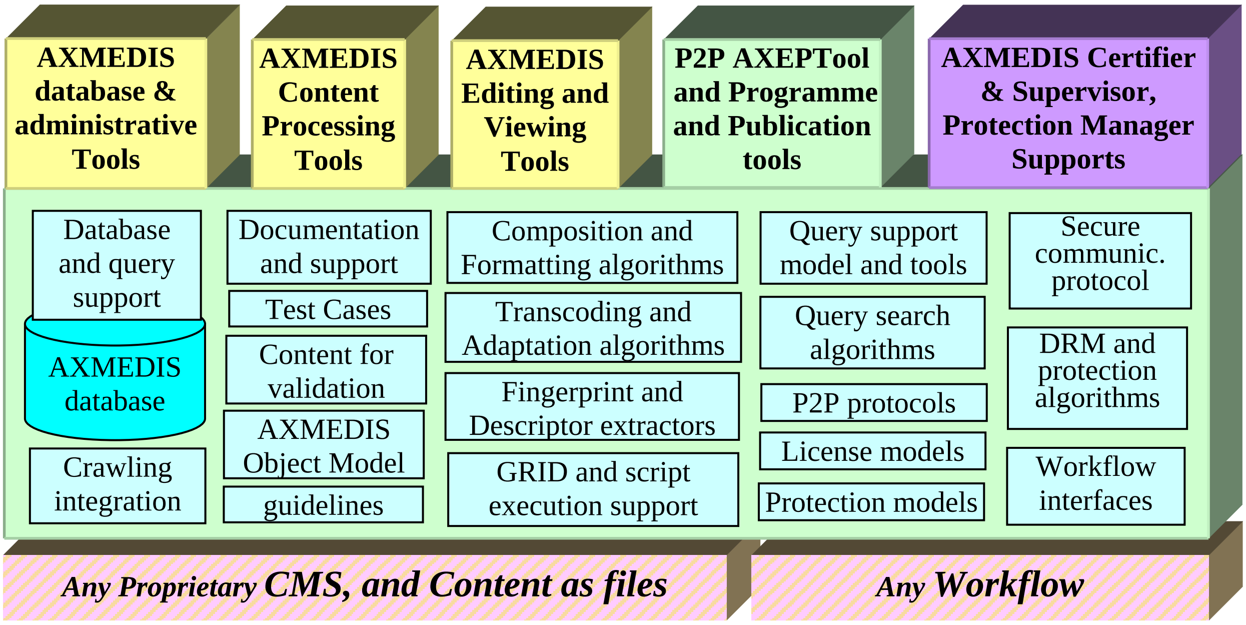

AXMEDIS database & administrative Tools
AXMEDIS Content Processing Tools
AXMEDIS Editing and Viewing Tools
P2P AXEPTool and Programme and Publication tools
AXMEDIS Certifier & Supervisor, Protection Manager Supports
Database and query support
AXMEDIS database
Crawling integration
Documentation and support
Test Cases
Content for validation
AXMEDIS Object Model
guidelines
Composition and Formatting algorithms
Transcoding and Adaptation algorithms
Fingerprint and Descriptor extractors
GRID and script execution support
Query support model and tools
Query search algorithms
P2P protocols
License models
Protection models
Secure communic. protocol
DRM and protection algorithms
Workflow interfaces
Any Proprietary CMS, and Content as files
Any Workflow

Fig.4 -- AXMEDIS Framework structure, a simplified view

The general infrastructure gives a common ground on the base of which other content based applications and tools can be built. In addition, to the modules and tools described before, the most relevant parts of the **AXMEDIS Framework** are:

- Requirements and their revisions,
- test cases and uses cases and their revisions,
- content for validations, both single resources and metadata and demonstrative AXMEDIS objects,
- general documentation of AXMEDIS tools and supports, including the:
    - whole specification of the AXMEDIS framework and the
    - detailed technical documentation of the source code,
- CVS tree with source codes of the several modules of the AXMEDIS framework,
- examples of AXCP scripts modeling algorithms for content compositions and formatting, for transcoding and adaptation, for extraction of fingerprint and descriptors, content processing, license manipulation and verification, license adaptation, etc., for many different formats of digital resources and for any categories of them: audio, video, document, multimedia, images, animations, text, metadata, etc.,
- examples and models of licenses,
- example and models for protection information,
- examples of workflow usage and programming for controlling AXMEDIS Factories,
- examples of queries and selections for accessing to the database,
- tutorials on content:
    - general aspects and state of the art,
    - content production,
    - content protection,
    - on AXMEDIS tools,

    - on distribution tools,
    - on general AXMEDIS aspects, etc.,
- guidelines for source code production for contributing to the AXMEDIS framework,
- guidelines on content production and distribution,
- guidelines for the production of AXMEDIS Plug-ins for AXCP and AXMEDIS Editors,
- guidelines for the production of licenses on the basis of contracts,
- ready to use/install AXMEDIS tools such as: AXMEDIS Players, AXEPTool, AXMEDIA tool, AXMEDIS Editors, AXMEDIS Programme and Publication tools, AXMEDIS Content Processing Tools, AXCS, AXMEDIS PMS.
-

3.1 Accessing to the AXMEDIS Framework

The present status of the AXMEDIS Framework can be obtained from its coordinator or partners. Demonstrations of the AXMEDIS tools and of the whole AXMEDIS Framework are provided at AXMEDIS conferences and in other occasions listed on the AXMEDIS Portal. The AXMEDIS Framework can be accessed by all affiliated partners. The Affiliation to AXMEDIS is performed by subscribing an Affiliation Agreement with an AXMEDIS Contractor. The Affiliation Agreement and the list of Contractors are accessible on the AXMEDIS portal.

There are many **reasons to get affiliated to AXMEDIS,** which can be summarized as follow:

- Obtaining access to an *open platform* that can be customized for your production., protection and distribution needs;
- *Reduction of costs* for content gathering, processing, production, protection and distribution;
- Adopting a standard model (MPEG-21) for content and licenses modeling and thus for inserting DRM in your business;
- Establishing contacts with other business partners interested in exploiting similar technology;
- Acquiring a greater control about content usage;
- Creating customized players;
- Exploiting and trial of new business models;
- Exploiting capabilities of secure legal P2P distribution;
- Setting up and create a customized distribution channel interoperable with others;
- Setting up some new service (empowering your present solution) on the basis of AXMEDIS technology;
- Setting up of one-stop service for content protection and DRM set up;
- Allowing reporting to your business customers which rights are exploited on their content;
- Allowing the management of rights reporting for multimedia products;
- Allowing using a solution that can be safer and more flexible with respect to state of the art;
- Saving money in accessing at innovative technologies for content production and distribution, integrated environment;
- Accessing to strongly innovative technology to trial it;
- Contributing to the AXMEDIS Framework is allowing you to continuing accessing to the framework reducing the costs for its accessibility.

Research institutions and technology providers are interested in getting affiliated with AXMEDIS to:

- make visible, promote, produced algorithms and tools that can be used for content processing and modeling and that can be in some how integrated into the AXFW. These tools may be provided as demonstrators with limited capabilities;
- exploit the AXMEDIS Framework to make business with it for the reasons reported in the above list;
- add new content models and new DRM models and make them interoperable with MPEG-21 and others already in place on AXMEDIS;
- test new algorithms and tools with respect to the state of the art solutions, in a very easy and cheep manner;
- access at low cost a framework by means of which several different configurations and solutions may be built to cover the needs of the value chain actors and tested with low effort;

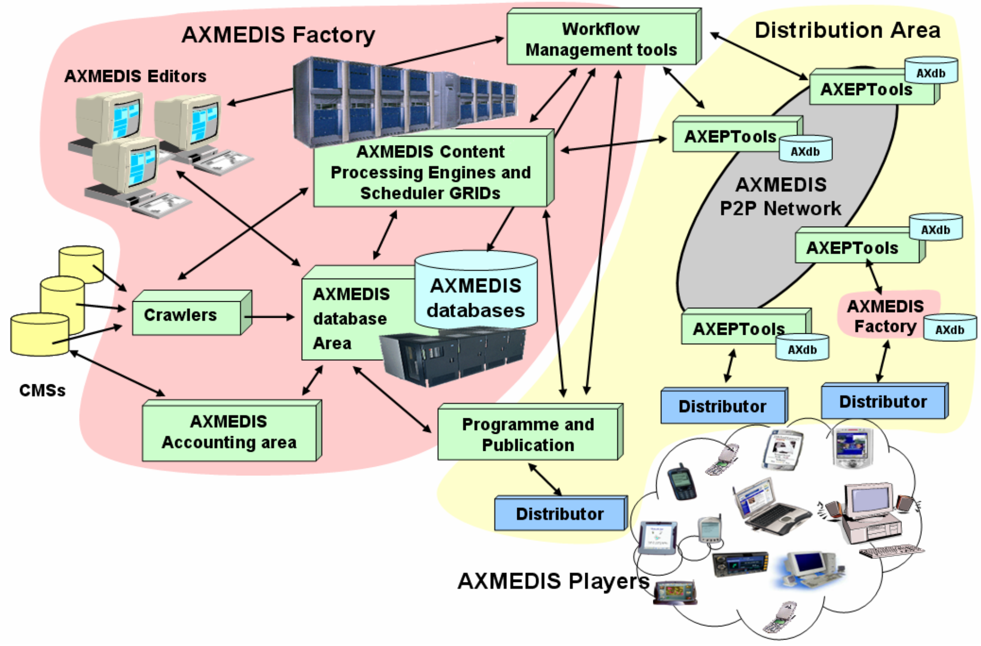
- access at tools based on MPEG-21 standard;
- collaborate with very relevant and well known research institution and companies of the areas;
- etc.

The present **status of the AXMEDIS Framework** can be obtained from its coordinator. Demonstrations of the AXMEDIS tools and of the whole AXMEDIS Framework are provided at AXMEDIS conferences and in other occasions listed on the AXMEDIS Portal. The AXMEDIS Framework can be accessed by Affiliated Partners. The Affiliation to AXMEDIS may be performed by subscribing an Affiliation Agreement with an AXMEDIS Contractor.

The AXMEDIS 2006 conference will be held in Leeds in December 2006. The Call For Papers is open until April 2006. http://www.axmedis.org/axmedis2006/

3.2 AXMEDIS Content Production

The Content Production area of AXMEDIS is mainly focussed on what it is called AXMEDIS Factory automating: (i) the packaging containing the digital resources (the real content), (ii) adapting and transcoding, (iii) protecting content and producing corresponding Prot-Info, (iv) publishing and distributing the produced package, (v) producing licenses for the users, etc.



Content Object Production

- Existing contents (e.g., resources and metadata located in databases, CMSs, file systems) are crawled and collected using automated processes and rules (e.g., Crawler, rule editing);
- Crawled contents or new contents can be inserted into the AXMEDIS Database;
- Content can be automatically packaged in AXMEDIS model (which is MPEG-21 compliant);
- Content production and elements (metadata, resources, protection information and licenses) can be processed manually with authoring tools and editor or automatically with AXMEDIS Content Processing tools based on executable rules that can be hosted on single or massive GRID of computers;
- Authoring tools can be used to insert/revise metadata, define protection information, define DRM licenses models, to modify content or simply to view the objects;
- Various processing are offered (adaptation, transcoding, protection, etc.), either automatically or manually (using GUI editor);
- The various components and digital resources can be glued together by means of SMIL based templates and style that may be used to define the usage interface (format) of the whole object: karaoke, collection, menus, sliding presentation, buttons, live, animations, etc.;

- Results can be sorted in a database or on file systems or published in towards distributors or on the B2B P2P AXMEDIS network;
- They are now available on the AXMEDIS network, for further aggregation, distribution, etc., to be searched, modified, or shared;
- Queries and P2P allow retrieving content located in all the connected AXMEDIS Factories;
- The queries can be activated to automatically react at eventual changes in the sources, and thus to perform an automatic production/update;
- Queries can be performed on the basis of classification and identification metadata, but also on technical features, descriptors, licensing information (PAR), etc.;
- Once identified the objects or queries, they can be used as input parameter of processing rules for the GRID AXMEDIS Content Processing;
- All these activities can be governed by Workflow Management Tools for defining process production flow and information of the content factory and among different factories.

3.3 AXMEDIS Content Processing Capabilities

The AXMEDIS Content Processing Area has been designed to provide a set of digital content processing tools to aid content designers to create rules/script for **automating** content production and processing and in more details for:

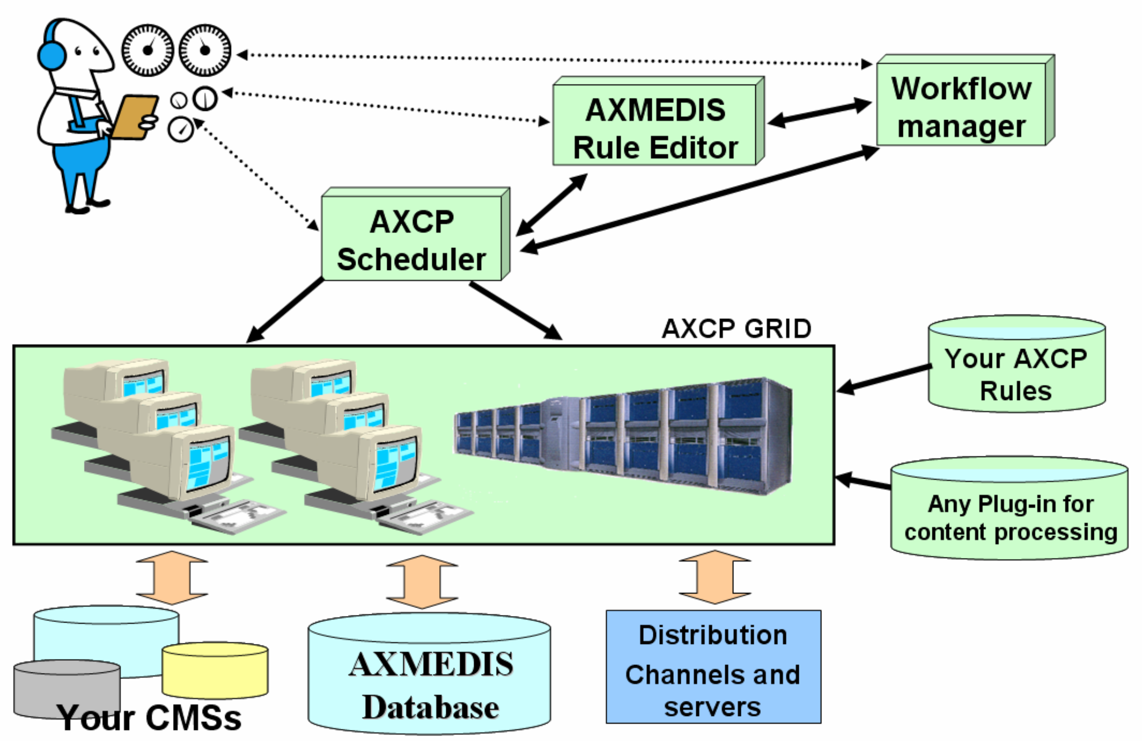
- **Content Ingestion and Gathering** from
  - Content Management Systems, from file system, or protocols;
  - processing resources and coupling them with metadata;
- **Content Retrieval** from
  - AXMEDIS database;
  - other AXMEDIS content Factories by means of the P2P tools, namely AXEPTool;
  - Content Management Systems, CSMs;
- **Content Storage** into
  - an AXMEDIS database;
  - ODBC based databases, other databases via Web Services, and other models will added later;
- **Content Processing** such as
  - digital resources adaptation, extraction of descriptors, transcoding, synchronisation, metadata processing, estimation of fingerprint, watermarking, indexing, content summarization, etc.;
  - metadata manipulation, mapping and adaptation;
- **Content Composition** for
  - creation of content components or objects as a combinations of raw assets such as Text, Images, Audio, Video (actual shot), Animation (synthetic), metadata, descriptors, licenses, multimedia objects such as MPEG-4, HTML, SCORM, macromedia tool file, animations, games, etc.;
  - creation of content as linear or hierarchical combination of content components;
- **Content Formatting**
  - gluing content elements together by means of SMIL based templates and applying style sheets to define the usage interface (format, layout) of the whole collection of content elements and the interested content usage paradigms (leaving open some parameters). For example, karaoke, collection browsing, selection menus, sliding presentation, stable background with a window with live video, animated text moving on an image, running text, etc.;
  - optimization of styling parameters left open or defining them manually to arrange for example: best fitting of images in the screen, optimizing the amount of text in the page, best time fitting, etc.;
- **Content Protection** such as:
  - protection of digital resources and full objects with their complex structure;
  - creation of Protection Information parameters, such as keys, or other features;

  - applying Protection Information model to AXMEDIS object, segmenting digital resources, slicing objects, applying encryption, scrambling, compression, and many other algorithms;
  - posting specific protection information for each given object to the AXMEDIS Certifier and Supervisor server;
- **Content Licensing** for
  - generating license from license model and additional information, storing licenses, posting them on license server automatically;
  - transcoding/translating licenses;
  - invocation of some verification algorithms about licenses and available rights to simulate the usage from the user site;
- **Content Publication and Distribution** towards
  - any distribution channel, producing programme and its schedule;
  - P2P network of other AXMEDIS Factories of content integrators, producers, and distributors.

3.4 AXMEDIS Content Processing Tools

In order to exploit the above list of features to process/manipulate content, resources, licenses, XML, SMIL, databases, protection information, etc., the AXMEDIS Content Processing Area is governed by a set of tools:

- **AXMEDIS Rule Editor**: to produce, debug, test and validate the executable AXCP Rules that can be:
  - written with a simple **AXCP language for content production** which is an extension of Java script;
  - created as macros from AXMEDIS Editor and authoring tools;
  - tested, debugged and validated on the AXMEDIS Rule Editor;
  - activated for content processing on any AXCP GRID Node or on a single computer;
  - used for B2B or B2C purpose;
  - used/parameterized for producing content on demand or to be integrated in your content factory;
  - activated from your Workflow Manager engine via web service;
  - activated by changes in remote objects and queries in the local database and on the P2P network.
- **AXCP GRID**: a set of general purpose or specialized computers to execute AXCP Rules governed by the AXCP Rule Scheduler;
- **AXCP GRID Node**: a single general purpose or specialized computer of the AXCP GRID;
- **AXCP Scheduler**: to schedule the AXCP rule of GRID nodes according to the content production and processing needs in terms of time and resources.



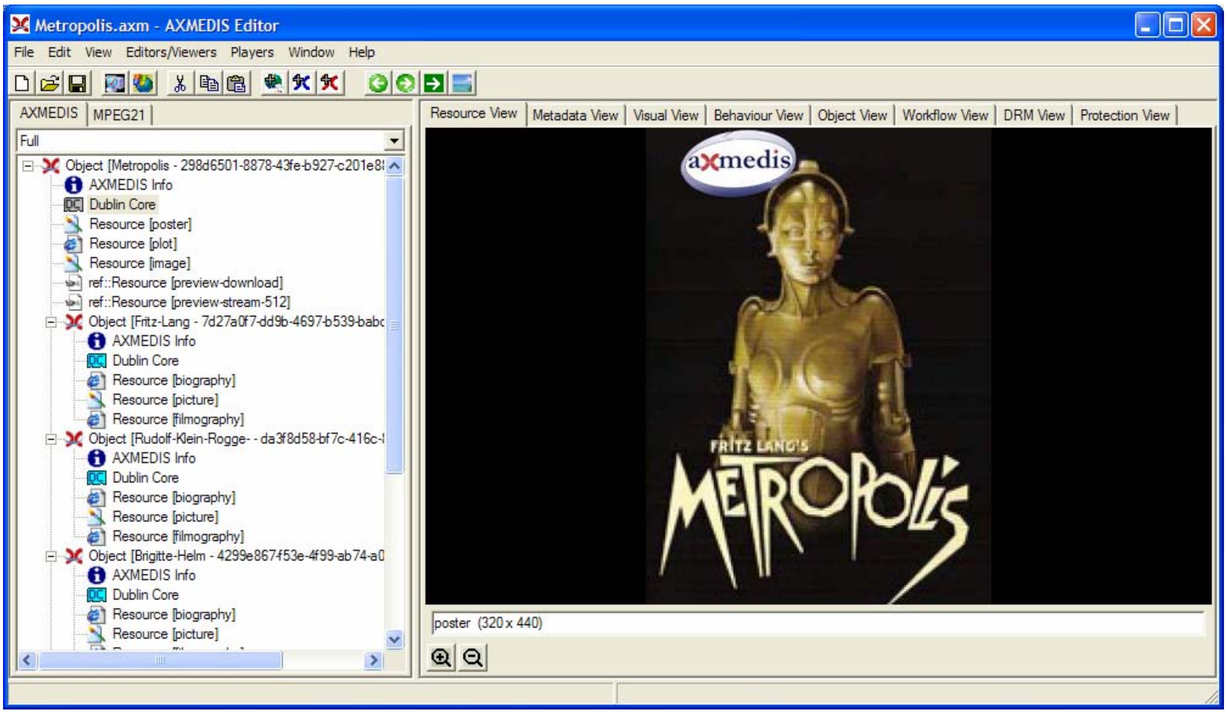
The processing capabilities and functionalities reported in the previous section and that can be exploited from AXCP Tools and Rules can be simply expanded by means of realizing or installing a set of additional plug-ins. The AXMEDIS Plug-in technology is open since:

- the specification of plug-in format is public;
- a plug-in tool kit (including examples and source code for creating those plug in) are public and accessible to all;
- a large part of the above mentioned features are provided by means of AXMEDIS Plug-ins. This demonstrate the solution flexibility;
- any user or third party company can create its own plug-in or include in a plug-in any open third party library including those open source.

3.5 AXMEDIS Editors, the authoring tools

AXMEDIS Editor is used for the manual production of AXMEDIS Objects, and allows creating and manipulating object features and their different aspects:

- ***Structure***, to add, remove, move digital resources and metadata inside the AXMEDIS Object (hierarchy editor and viewer of the object);
- ***Resource manipulation***, to use content processing plug-ins for generic and customizable resource manipulation (the same algorithms and tools used in the AXMEDIS Content Processing);
- ***Metadata editing***, allowing to edit/mapping the metadata related to digital resources and objects (metadata editor and viewer);
- ***Visual and behavioral editing,*** allowing to define content usage paradigm with SMIL organization of resources to present/layouting the digital content contained inside the AXMEDIS Object (visual and behavioral editor and viewer);
- ***Protection editing,*** allowing to specify the protection algorithms to be used for the AXMEDIS Object protection, and thus to define the Protection Information (Protection Editor and Viewer);
- ***DRM editing***, allowing to produce and verify licenses for end users and/or distributors of the AXMEDIS Object as well as the Potentially Available Rights, PAR, that could be acquired on objects shared in the P2P Network (DRM editor and viewer);
- ***Workflow***, allowing editing and viewing the status and the work to be done on the AXMEDIS Objects involved in the workflow process (workflow editor and viewer).


Metropolis.axm - AXMEDIS Editor
File Edit View Editors/Viewers Players Window Help
AXMEDIS MPEG21
Full
Object [Metropolis - 298d6501-8878-43fe-b927-c201e8
AXMEDIS Info
Dublin Core
Resource [poster]
Resource [plot]
Resource [image]
ref::Resource [preview-download]
ref::Resource [preview-stream-512]
Object [Fritz-Lang - 7d27a0f7-dd9b-4697-b539-babc
AXMEDIS Info
Dublin Core
Resource [biography]
Resource [picture]
Resource [filmography]
Object [Rudolf-Klein-Rogge- - da3f8d58-bf7c-416c-
AXMEDIS Info
Dublin Core
Resource [biography]
Resource [picture]
Resource [filmography]
Object [Brigitte-Helm - 4299e867-f53e-4f99-ab74-a0
AXMEDIS Info
Dublin Core
Resource [biography]
Resource [picture]
Resource View Metadata View Visual View Behaviour View Object View Workflow View DRM View Protection View
axmedis
FRITZ LANG'S
METROPOLIS
poster (320 x 440)

Moreover, **AXMEDIS Editor:**

- allows to perform queries to look for content to produce AXMEDIS Objects integrating other AXMEDIS objects coming from the internal AXMEDIS Database, the AXMEDIS P2P network or possibly from the factory CMS;
- allows download/upload AXMEDIS Objects from/to the AXMEDIS Database and file system;
- can be controlled by the AXMEDIS workflow system to integrate manual operations on AXMEDIS objects inside the production process formalized with the workflow system;
- can be used to inspect automatically generated objects for validation;
- can finalize the production of automatically produced objects;
- can produce rules/script to be used as templates for the AXMEDIS Content Processing environment.

3.6 Content Adaptation facilities

AXMEDIS objects can be created to be distributed over heterogeneous networks and towards different kind of terminals, client tools/devices. Moreover, the people who will ultimately consume and interact with the content may have different behavior and preferences, and the best formats to provide them the best experience on their terminal could be different. Consequently, digital items may need to be adapted to fit any particular usage environment. This is the goal of AXMEDIS content adaptation tools which aim at achieving interoperable transparent access to (distributed) advanced multimedia content by shielding Users from network and terminal installation, management and implementation issues.

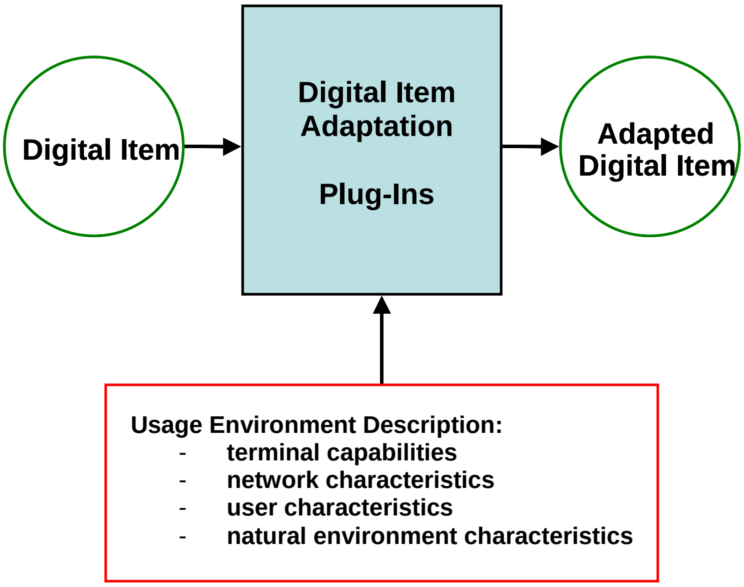
Adaptation may involve:

- transcoding of digital resources that means to change the format (for example, from TIF to GIF, from MPEG-4 to video, from a video to MPEG-4, from MPEG-4 video to a MPEG-2, from audio to symbolic music, from audio to MIDI, from audio in PCM to a ringtone format for mobiles, etc.), re-sampling, shrinking, stretching; In some cases, some features related to resolution, interactivity, are lost in the process in favor of having the content usable in another format. Typically the content is produced in a format that is transcoded scaling down its features and not the vice versa;
- manipulation of licenses, reduction of license scope, reduction of time or territorial information, transformation of format, translation of license (such as passing from ODRL to MPEG-21 REL), etc.;
- manipulation of metadata, metadata mapping, metadata reduction, translation of metadata, etc.

The **adaptation process** can be performed:

- during the content production (for digital resources, licenses and metadata) by exploiting functionalities accessible from the AXMEDIS Editors and/or for the AXMEDIS Content Processing tools
- directly on the player terminal/device (mainly for digital resources and metadata). In this case, ISO/IEC 21000 (MPEG-21) specified a set of normalized tools for the adaptation of digital content describing the usage environment of a digital item to command adaptation tools. Within the AXMEDIS players, MPEG-21 usage environment descriptions are used to drive the adaptation tools considering:
  - Terminal capabilities (codec, formats, input-output, etc., supported by the terminal),
  - Network characteristics (for example, the minimum guaranteed bandwidth of a network),
  - User characteristics (presentation preferences, auditory or visual impairment etc.),
  - Natural environment characteristics (for example, the illumination characteristics that may affect the perceived display of visual information).

The conceptual architecture of the adaptation engine is shown below: a digital item is subject to adaptation thanks to dedicated plug-ins to produce the adapted digital item; the adaptation performed by the plug-ins is parameterized according to MPEG-21 usage environment descriptions.

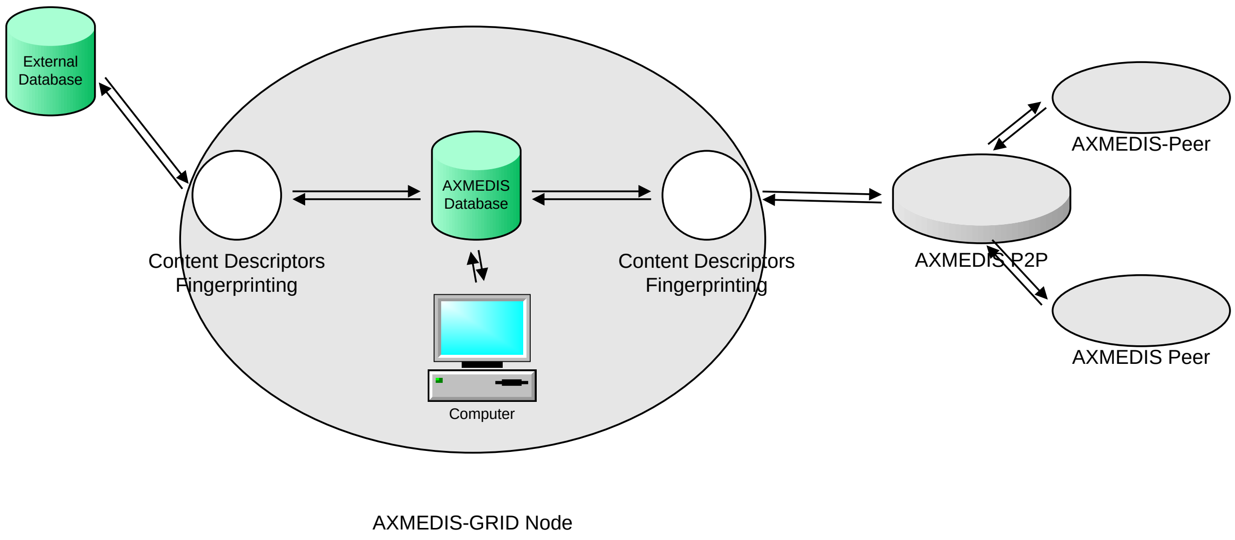


3.7 Content Fingerprint and Descriptors extraction

Among the several content processing algorithms that can be applied in AXMEDIS, a large set can be classified as extractors or estimators of Fingerprint and/or Descriptors of content.

Content fingerprints and descriptors can be used in the AXMEDIS framework for different purposes:

- Classification and recognition of content and/or digital resources;
- Identification of a single piece of content/resource and thus for content certification;
- Authentication/verification of content integrity.



They can be of classified as:

- **High-level descriptors** to describe content with a set of high-level features independent on the format and content resolution. They are typically high level features immediately related with concepts understandable by humans, such as: rhythm and tonality for music, subject for text, etc. They can be very easily used as Descriptors by humans to make queries in the data bases, and thus to retrieve similar content;
- **Digital fingerprints** (or **perceptual hash values**). They can be compared with human fingerprints and may be used to identify a specific content and are robust against data transformation;
- **Low-level descriptors** to describe digital information at lower level and sometime these descriptors are not independent on the format and resolution, such as: energy for music, spectra for images, dynamic, duration, etc. Similar content may share similar low-level descriptors. In some cases, they are used at the basis to estimate high-level descriptors. They can be used for recognition of content as well;
- **Low-level fingerprints** to estimate a value from a specific digital resource, in many cases estimated with algorithms that do not take into account the content type, such as: cryptographic hash value that can be estimated for any digital file. They can be used for content and digital resource verification of consistency and authentication.

In AXMEDIS, Content Fingerprints and Descriptors can be:

- manually selected and estimated via the AXMEDIS Editors;
- automatically estimated and stored in the object metadata or other places by means of an AXMEDIS Content Processing Rule script. For example: during the acquisition/crawling of content from CMS, during content composition and/or formatting, during any content processing also included in the production on demand;
- estimated to verify integrity of content when the content is opened by an AXMEDIS Editor, or by an AXMEDIS Player, or processed by a AXMEDIS P2P tool such as AXEPTool or AXMEDIA, etc., that is any time that an AXMEDIS object is loaded into the AXOM core component of AXMEDIS Framework.

4 AXMEDIS Content Processing Script Language User Manual

4.1 AXOM, AXMEDIS Model and Javascript classess

The set of JavaScript Classes that wraps the main classes of the AXMEDIS Object Model includes the:

AxmedisObject - for managing an instance of the AXMEDIS Object
AxInfo - for managing the metadata of the AXInfo section.
AxDublinCore - for managing an instance of Dublin Core metadata section
AxResource - for managing digital resources embedded or to be embedded in the AXMEDIS bject.
AxMetadata - for managing an instance of an AxMetadata object to cope with generic metadata
Example - Example of usage and scripts

4.1.1 AxmedisObject

It is the mapping of an AXMEDIS Object in JavaScript. According to the specification of the AXMEDIS OBJECT MANAGER and Axmedis Data Model, it provides and wraps methods to:
• Create an empty AXMEDIS object with own AXOM by instantiating a new Axmedis Object.
• Create and fill an AXMEDIS object with own AXOM by loading content from an URL.
• Add/Remove an Element/Object to the AXMEDIS object. The addition of an element returns the new object reference inside the Axmedis Object.
• Get all Elements/Objects. It returns a Javascript array of Element/Objects.
• Add Resource, it adds a digital resource (audio, video, text, etc…) to a specific Element/Object. It returns the new resource object reference inside the Axmedis Object.
• Remove a Resource (audio, video, text, etc…) by using the object Resource reference.
• Get Resources. It returns a Javascript array of Resource objects.
• Add an AXInfo, Dublin Core or generic metadata object. It returns the new metadata object reference inside the Axmedis Object.
• Get the AXInfo, the Dublin Core metadata
• Remove any metadata object.

Exposed properties:

string AXOID
It provides the current axoid

number childrenCount
It represemts the number of Children items

string URI
Where the object is located

string contentID
The content identifier

Exposed methods:

Exposed methods:

AxmedisObject()
Empty constructor for an empty AXMEDIS Object.

AxmedisObject (string URI)
Constructor with an URI parameter. The Axmedis Object is loaded by using the URI. It can specifies or a file system or a database location.

addMetadata(AxInfo obj)
It adds an AxInfo metadata objects AxInfo.

addMetadata(AxDublinCore obj)
It adds an AxDublinCore metadata.

addMetadata(AxMetadata obj)
It adds an AxMetadata object.

array getGenericMetadata()
It returns an array of generic metadata objects or null.

AxDublinCore getDublinCore()
It returns an AxDublinCore metadata object

AxInfo getAxInfo()
It returns an AxInfo metadata object

removeMetadata (AxInfo obj)
It removes an AxInfo metadata objects AxInfo.

removeMetadata (AxDublinCore obj)
It removes an AxDublinCore metadata.

removeMetadata (AxMetadata obj)
It removes a generic metadata object

addContent(AxResource res)
It adds an AxResource content instance

addContent(AxmedisObject res)
It adds an AxmedisObject content instance

addContent(Document res)
It adds a Document coming from Searchbox tool transforming it in AxResource

array getContent()
It returns an array of AxResource and AxmedisObject content or null

removeContent (AxResource res)
It deletes an existing AxResource content instance

removeContent (AxmedisObject res)
It deletes an existing AxmedisObject content instance

save(string path)
It saves the axmedis object onto filesystem

uploadToDB(string URI)
It uploads the axmedis object into database at the specified URI

obtainDefinitiveAXOID()
The AXMDEDIS Object is univoclly defined by asking for a Definitive AXOID

registerToAXCS()
The AXMDEDIS Object is registerd to AXCS.

4.1.2 AxInfo

It maps and allows managing the metadata of the AXINFO in JavaScript. This class manages the access to individual elements and fields in AXINFO metadata, this class maps all the functionalities provided by AxInfo class exposing setter and getter methods for accessing to data. It allows to manage:

- ObjectCreator information
- Owner information
- Distributor information
- Object Status information
- PromoOf information
- Workflow information
- Fingerprints information
- PAR information

Exposed properties

//Contibutor
number objectContributorCount
the number of ObjectContributor present

//Owner section
string ownerID
the code identifying the owner

string ownerIDCoding
coding used to identify the owner

string ownerName
the name of the owner

stringownerURL
URL of the owner

string ownerCompany
company of the owner

string ownerCompanyUrl
company URL of the owner

string ownerNationality
nationality of the owner

// Distributor section
number distributorCount
number of Distributors present

// Object Status section

string objectAccessMode
the access mode: “READ_ONLY” or “READ_WRITE” string.

Date objectCreationDate
the local date and time of object creation

Date objectLastModificationDate
the local date and time of object modification

number objectVersion
version of the object

number objectRevision
revision of the object

string objectStatus
current status of the object

string objectType
object type (“BASIC” or “COMPOSITE”)

string objectProtectionStamp
the protection stamp

// PromoOf section
number promoOfAXOIDCount
count of AXOID in the PromoOf section

// Workflow section
string workflowItemID
WorkflowWorkItemID

string workflowWorkspaceInstanceID
WorkflowWorkspaceInstanceID

// Internal PAR
boolean hasInternalPAR
how many Internal PAR sections are present (0 or 1)

string internalPARStatus
the internal PAR status

// PAR section
boolean hasPAR
how many PAR sections are present (0 or 1)

string PARStatus
the PAR status

string PARLicensingURL
the licensing URL

// Status section
boolean isProtected
if the object is protected or not. True means protected.

boolean isGoverned
if the object contains a licence or not.

Exposed methods

AxInfo()
Empty contructor

AxInfo(AxInfo info)
Copy constructor. It create a copy of info object

ObjectContributor Management
number addObjectContributor(number pos)
Adds a new ObjectContributor in the position given (starting from 0), position -1 means to add at the end. The return value indicates the position in which it is added.

removeObjectContributor (number pos)
Removes an ObjectContributorfrom the position specified

number getObjectContributorCount()
Returns the number of ObjectContributor present

number findObjectContributorByAXCID(string axcid)
Returns the position of an ObjectContributor with a specific AXCID. It returns -1 if not found.

number findObjectContributorByName(string nam)
Returns the position of an ObjectContributor with a specific Name. It returns -1 if not found.

string getObjectContributorAXCID(number refNum)
allow getting the AXCID value for an ObjectContributor identified by position refNum

setObjectContributorAXCID(number refNum, string axcid)
allow setting the the AXCID value for an ObjectContributor identified by position refNum

string getObjectContributorName(number refNum)
allow getting the Name value for an ObjectContributor identified by position refNum

setObjectContributorName(number refNum, string name)
allow setting the Name value for an ObjectContributor identified by position refNum

string getObjectContributorURL(number refNum)
allow getting the URL value for an ObjectContributor identified by position refNum

setObjectContributorURL(number refNum, string URL)
allow setting the URL value for an ObjectContributor identified by position refNum

string getObjectContributorCompany(number refNum)
allow getting the Company value for an ObjectContributor identified by position refNum

setObjectContributorCompany(number refNum, string company)
allow setting the Company value for an ObjectContributor identified by position refNum

string getObjectContributorCompanyURL (number refNum)
allow getting the CompanyURL value for an ObjectContributor identified by position refNum

setObjectContributorCompanyURL(number refNum, string URL)
allow setting the CompanyURL value for an ObjectContributor identified by position refNum

string getObjectContributorNationality(number refNum)
allow to get and set the Nationality value for an ObjectContributor identified by position refNum

setObjectContributorNationality(number refNum, string nationality)
allow setting the Nationality value for an ObjectContributor identified by position refNum

Owner Management
string getOwnerID()
allow getting the code identifying the owner

setOwnerID(string value)
allow setting the code identifying the owner

string getOwnerIDCoding()
allow getting the coding used to identify the owner

setOwnerIDCoding(string value)
allow setting the coding used to identify the owner

string getOwnerName()
allow getting the name of the owner

setOwnerName(string value)
allow setting the name of the owner

string getOwnerURL()
allow getting the URL of the owner

setOwnerURL(string value)
allow setting the URL of the owner

string getOwnerCompany()
allow getting the company of the owner

setOwnerCompany(string value)
allow setting the company of the owner

string getOwnerCompanyURL()
allow getting the company URL of the owner

setOwnerCompanyURL(string value)
allow setting the company URL of the owner

string getOwnerNationality()
allow getting the nationality of the owner

setOwnerNationality(string value)
allow setting the nationality of the owner

number addOwnerDescription(number pos)
adds a new description of the owner at the position specified or at the end if position is -1. The return value indicates the position where it is added.

removeOwnerDescription(number pos)
removes the description specified

string getOwnerDescription(number pos)
allow getting the value of the description

setOwnerDescription(number pos, string description)
allow setting the value of the description

string getOwnerDescriptionLanguage(number pos)
allow getting the value of the description language

setOwnerDescriptionLanguage(number pos, string description)
allow seeting the value of the description language

Distributor Management
addDistributor()
adds a Distributor if not present.

removeDistributor()
removes the Distributor

number getDistributorCount()
returns the number of Distributors present

string getDistributorAXDID()
allow getting the AXDID value for the Distributor

setDistributorAXDID(string value)
allow setting the AXDID value for the Distributor

string getDistributorName()
allow setting the Name value for the Distributor

setDistributorName(string value)
allow getting the Name value for the Distributor

string getDistributorURL()
allow getting the URL value for the Distributor

setDistributorURL(string value)
allow setting the URL value for the Distributor

string getDistributorNationality()
allow getting the Nationality value for the Distributor

setDistributorNationality(string value)
allow setting the Nationality value for the Distributor

Object Status
string getAccessMode()
it returns the access mode: “READ_ONLY” or “READ_WRITE” string.

setAccessMode(string value)
allow setting the Access the object specifying strings “READ_ONLY” or “READ_WRITE”

Date getCreationDate()
get the local date and time of object creation

Date getLastModificationDate()
get the local date and time of object modification

number getVersion()
get the version of the object

number getRevision()
get the revision of the object

string getObjectStatus()
allow getting current status of the object, the status values are factory dependent and set by the workflow therefore cannot be defined a priori.

setObjectStatus(string value)
allow setting the current status of the object, the status values are factory dependent and set by the workflow therefore cannot be defined a priori.

string getObjectType()
allow to get object type (“BASIC” or “COMPOSITE”)

boolean getObjectIsProtected()
allows getting if the object is protected or not. True means protected.

setObjectIsProtected(Boolean value)
allows to get and set if the object is protected or not

string getProtectionStamp()
allows getting the protection stamp

setProtectionStamp(string value)
allows setting the protection stamp

boolean getObjectIsGoverned()
allow getting if the object contains a licence or not. The license is not stored in the axinfo, the setter should be used to update the axinfo when the licence is added/removed from the object

setObjectIsGoverned(in value:bool)
allow setting if the object contains a licence or not. The license is not stored in the axinfo, the setter should be used to update the axinfo when the licence is added/removed from the object

Object Creator Management
string getObjectCreatorAXCID()
allow getting the AXCID value for the ObjectCreator

setObjectCreatorAXCID(string val)
allow setting the AXCID value for the ObjectCreator

string getObjectCreatorName()
allow getting the Name value for the ObjectCreator

setObjectCreatorName(string name)

allow setting the Name value for the ObjectCreator

string getObjectCreatorURL()
allow getting the URL value for the ObjectCreator

setObjectCreatorURL(string url)
allow setting the URL value for the ObjectCreator

string getObjectCreatorCompany()
allow getting the Company value for the ObjectCreator

setObjectCreatorCompany(string company)
allow setting the Company value for the ObjectCreator

string getObjectCreatorCompanyURL()
allow getting the CompanyURL value for the ObjectCreator

setObjectCreatorCompanyURL(string URL)
allow setting the CompanyURL value for the ObjectCreator

string getObjectCreatorNationality()
allow getting the the Nationality value for the ObjectCreator

setObjectCreatorNationality(string nationality)
allow setting the Nationality value for the ObjectCreator

PromoOf Management
addPromoOfAXOID(string axoid, number position)
adds a new AXOID in the PromoOf section, the position indicates where to put the AXOID, -1 means at the end

removePromoOfAXOID(number position)
removes the AXOID in the position specified

number getPromoOfAXOIDCount()
get the count of AXOID in the PromoOf section

string getPromoOfAXOID(number position)
allow getting the AXOID in a specified position

setPromoOfAXOID(number position, string value)
allow setting the AXOID in a specified position

Workflow Status
string getWorkflowWorkItemID()
allow getting the WorkflowWorkItemID

setWorkflowWorkItemID (string value)
allow setting the WorkflowWorkItemID

string getWorkflowWorkspaceInstanceID()
allow getting the WorkflowWorkspaceInstanceID

setWorkflowWorkspaceInstanceID (string value)
allow setting the WorkflowWorkspaceInstanceID

Internal Potential Available Rights Management
addInternalPotentialAvailableRights()
adds a new Internal PAR section

removeInternalPotentialAvailableRights()
removes the Internal PAR section

getInternalPotentialAvailableRightsCount()
gets how many Internal PAR sections are present (0 or 1)

string getInternalPotentialAvailableRightsStatus()
allow getting the internal PAR status

setInternalPotentialAvailableRightsStatus(string value)
allow setting the internal PAR status

License getInternalPotentialAvailableRightsLicense()
gets the license

Potential Available Rights Management
addPotentialAvailableRights()
adds a new PAR section if not present

removePotentialAvailableRights()
removes the PAR section

getPotentialAvailableRightsCount()
gets how many PAR sections are present (0 or 1)

string getPotentialAvailableRightsLicensingURL()
allow getting the licensing URL

setPotentialAvailableRightsLicensingURL (string value)
allow setting the licensing URL

string getPotentialAvailableRightsStatus()
allow getting the PAR status

setPotentialAvailableRightsStatus(string value)
allow setting the PAR status

License getPotentialAvailableRightsLicense()
gets the license

Object History Management
string getHistoryOfVersion(number version)
gets the history of a version as an XML string

4.1.3 AxResource

The class models the AxResource type of the Axmedis Framework, it is used to store digital resource (audio, video, image, text, etc…) both raw data and Axmedis digital component. This class wraps the AxResource class and provides functionalities exposing setter and getter methods to:

- access to the mime type
- access to the byte stream of the resource

- create a new resource and to embed a file or a reference inside a resource object

Exposed attributes

string mimeType
The MimeType associated with the extension of digital resource

string ref
reference

string contentID
content identifier

string encoding
encoding type

Exposed methods

load (string path)
It loads a Raw Resource from filesystem

save (string path)
It saves a Raw Resource onto filesystem

4.1.4 AxDublinCore

The class models the AxDublinCore type of the Axmedis Framework, it is used to store metadata in Dublin Core format.
This class d provides functionalities exposing setter and getter methods to:

- retrieve the description of metadata
- to store the metatada
- to get the number of elements
- to remove an element

Exposed properties

string metadataID

Exposed methods

addDCElement(string type, string value, string language)
Add a new element specifying type, value and language

addDCElement(string type, string value)
Add a new element specifying type, value. The language has an empty value

removeDCElement(string type, number refNum)
Delete an element specifying type and reference number.

string getDCElementValue(string type, number refNum)
Return an element specifying type and reference number.

string getDCElementValue(string type)
Return an element specifying type. The reference number is 0

setDCElementValue(string type, number refNum, string value)
Modify the element specifying type, reference number and value

string getDCElementLanguage(string type, number refNum)
Return the language of the element specifying type and reference number.

string getDCElementLanguage(string type)
Return the language of the element specifying type. The reference number is 0

setDCElementLanguage(string type, number refNum, string language)
Modify the language of element specifying type, reference number and language

number getDCElementCount(string type)
Return the number of elements specifying type.

AxMetadata

The class models the AxMetadata type of the Axmedis Framework, it is used to store metadata in XML format.
This class d provides functionalities exposing setter and getter methods to:

- retrieve the XML description of metadata
- store the XML of metatada

Exposed properties

string namespace
The used namespace in the XML description. It is a readonly attribute.

Exposed methods

AxMetadata ()
Empty constructor

AxMetadata (string xml)
Constructor with parameter. It builds an object storing the XML passed as a string

addXML (string xml)
It stores an XML description of metadata passed as a string

string getXML ()
It returns the string with XML of the metadata description

4.1.5 Examples of usage

```
// Function for creating the Dublin Core information
function createDC(obj,title)
{

    dc = obj.getDublinCore();
    dc.addDCElement("creator","AXCP Rule Editor");
    dc.addDCElement("title",title);
    dc.addDCElement("type","conversion to "+format);
    dc.addDCElement("description","Testing JSScript with resized and converted
images");
}
function adaptAxImages(axobj,height,width,mimeType)
{
    print("Creating adapted image");
    var h1 = new Array(1);
    var w1 = new Array(1);
    h1[0]=0;
    w1[0]=0;
    var resource = axobj.getContent();
    var i = 0;
    for (i in resource)
    {
        var res = resource[i];
        val = resource[i].mimeType.search("image");
        if(val==0)
        {
            ImageProcessing.GetInfo(resource[i],h1,w1);
            print("Original resource size is "+h1[0]+"x"+w1[0]);
            print("Resizing the resource at "+height+"x"+width);
            ImageProcessing.Resize(resource[i],height,width,false,resource[i]);
            ImageProcessing.GetInfo(resource[i],h1,w1);
            print("New resource size is "+h1[0]+"x"+w1[0]);
            if(resource[i].mimeType != format)
            {
                print("Converting the resource in "+format);
                ImageProcessing.Conversion(resource[i],mimeType,resource[i]);
            }
        }
    }
}
function cloneAxmedisObject(sourceObj,targetObj)
{
    var metadata = sourceObj.getGenericMetadata();
    if(metadata!=null)
    {
        for(j in metadata)
        targetObj.addMetadata(metadata[j]);
    }
    targetObj.addMetadata(sourceObj.getAxInfo());
    var resources = sourceObj.getContent();
    if(resources!=null)
    {
        for(k in resources)
        targetObj.addContent(resources[k]);
    }
}

// Function for creating the Axmedis Object by composing and converting
resources
function test()
{
    var imgFormat = getMimeType(format);
    selection.resolveQuery("test","test",0);
    var documentList = selection.getAXDBResult();
    for (i in documentList)
    {
        uri ="axdb://"+documentList[i];
        var axmedisObject = new AxmedisObject(uri);
        var dublinCore = axmedisObject.getDublinCore();
        var title = dublinCore.getDCElementValue("title")+"_Resized";
        var newObj = new AxmedisObject();
        cloneAxmedisObject(axmedisObject, newObj);
        createDC(newObj,title);
        adaptAxImages(newObj,h,w,imgFormat);
        print("Storing Object on disk");
        newObj.save(resourcePath+title+".axm");
    }
    return true;
}

//Entry point of the script
test();
```

4.2 Network

Javascript connection classes are a set of classes that provide the communiction support to the AXCP Engine by modeling in terms of primitive functionalities the network and database access. They consist in:

Form - The class models a simple form of a http web page
HttpConnection - The class models the http connection by providing primitive methods for accessing and retrieving information by means the http protocol
FtpConnection - The class models the http connection by providing primitive methods for accessing and retrieving information by means the http protocol
OdbcConnection - The class models the http connection by providing primitive methods for accessing and retrieving information by means the http protocol
WebServiceConnection - The class models a meta class for managing web services by loading WSDL description and dynamically creating services (methods).
Example - Example of usage and scripts

4.2.1 Form

This class models a form.

Exposed Properties

string formName
It is the name of the form

string formType
It is the type of form

string formValue
It is the value of form to post

Exposed methods
No methods

4.2.2 HttpConnection

The class provides the support to the http connection.

Exposed methods

HttpConnection()
Create a new empty HttpConnection object

string getResultMsg()
Return the last Result Message: an Exit Succes is returned otherwise the message string provides the type of fault.

string getContent()
Return the html page as string

boolean setLogin(string username, string password)
Set the account parameter for login

boolean getLogin()
Get the status of Login parameters, it returns true if they are set, false otherwise.

boolean clearLogin()
Reset the account parameters for login

boolean setPort(string port)
set a Port at the port number value

boolean clearPort()
Clear the port number resuming the default port.

boolean getToString(string url)
Get function, copy content at the specified url to a String. It is used in conjuction with getContent().

boolean getToFile(string url, string filePath)
Get function, copy content at the specified url to a File at filePath

boolean putFromFile(string url, string filePath)
Put function, put a content From a File at filePath to a specified url

boolean putFromString (string url, string buffer)
Put function, put content From a String buffer to a specified url

boolean post(string url, array jsform)
Post at the specified url the array of JSForm data It allows performing the POST http command to the specified Url. We suppose to use this command to send to the server contents of some form-fields inside a html page. Before using this command it is necessary to create an array of JSForm object to use with the POST command.

4.2.3 FtpConnection

The class provides the support to the Ftp connection.

Exposed methods

FtpConnection ()
Create a new empty FtpConnection object

string getResultMsg()
Return the last Result Message: an Exit Succes is returned otherwise the message string provides the type of fault.

string getContent()
Return the content as string

setLogin(string username, string password)
Set the account parameter for login

boolean getLogin()
Get the status of Login parameters, it returns true if they are set, false otherwise.

clearLogin()
Reset the account parameters for login

setPort(string port)
set a Port at the port number value

clearPort()
Clear the port number resuming the default port.

download(string Url,string FilePath)
It performs the download of a file. The location of file is provided by the Url parameter, the file is stored on file system at the FilePath location.

upload(string Url,string FilePath)
It allows uploading a file from the FilePath locationon own file system into a Url of the ftp server.

4.2.4 OdbcConnection

The class provides the support to an Odbc based connection

Exposed methods

OdbcConnection()
Create a new empty OdbcConnection object

setDbConnection(string OdbcSource, string UserName, string PassWord)
It allows setting the connection parameters. It has to be called before opening a Database connection.

openDbConnection()
It opens a Database connection after the call of setDbConnection.

execQuery(string query)
It allows making a SQL query. Results are provided in a table and can be retrieved by calling the access methods: nextRow and getRow

nextRow()
It allows browsing the table of results row by row. It set the internal pointer to the current row.

Array getRow()
It allows retrieving content from the last pointed row. Values are returned in a array of string. The lenght of the array depends on the number of column of the table.

string getErrorMessage()
It returns the last error message.

closeDbConnection()
It closes the Database connection.

4.2.5 WebServiceConnection

Class name in Javascript is WebServiceConnection

Exposed Propertie

wsdlURI
It is the URI used to load the WSDL

Exposed methods

WebServiceConnection(string URI)
Costructor method loads the WSDL specified by the URI string input parameter.

All other methods are defined dynamically according to the loaded WSDL,
All input parameters are string.

4.2.6 Examples of usage

```
var ftp = new FtpConnection();
ftp.setLogin (UserName, PassWord);
ftp.download (URL, DestinationPath);
print(ftp.getResultMsg());

var http = new HttpConnection();
var cont ent = new string();
http.setPort(80);
http.getToString(URL);
content = http.getContent();
print(content);

var http = new HttpConnection();
var formArray = new Array(2);
var content = new String();
formArray[0] = new Form(”FormName”, ”FormType”, ”FormValue”);
formArray[1] = new Form(”FormName”,”FormType” ,”FormValue”);
http.post(URL, formArray);
content = http.getContent();
print(http.getContent());

var webDav = new WebDavConnection();
webDav.mkCol(URL);
content = webDav.getContent();
print(content);

var odbc = new OdbcConnection();
var RowArray = new Array(3) ;
odbc.setOdbcConnection(DbName, UserName, PassWord);
odbc.openDbConnection();
odbc.execQuery(QUERY) ;
RowArray[0] = odbc.getRow();
odbc.nextRow();
RowArray[1] = odbc.getRow();
print(odbc.getErrorMsg());
odbc.closeDbConnection();

var Array = new Array();
var WebService = new WebServiceConnection(URL);
Array = WebService.generateAxoid(”mario ”, ”xxxxx”, ”Pluto”);
```

4.3 SearchBox Javascript Classes

The searchbox bridge provides the following set of JavaScript classes:
AXSearchbox - The main object for accessing searchbox.
Document - Used to handle documents as opaque objects.
MetadataValue - Used to hold metadata.
QueryParser - Used to specify the query string parser to use.
QueryInfo - Used to specify the information returned as result of a query.

QueryView - Used to restrict the set of documents returned as result of a query.
QueryAtomType - Used to specify the type of a QueryAtom.
QueryAtom - Used to build a query in RPN notation.
QuerySliceWeight - Used to specify slice weights.
QuerySpec - Used to submit a query.
QueryResult - Used to return information on a query result
Example - Example of usage and scripts

4.3.1 AXSearchbox

The main object for accessing searchbox.

Exposed Properties:
host [string]
Host where searchbox server is running.

port [string]
Port where searchbox server is listening.

username [string]
Username for authentication.

password [string]
Password for authentication.

Exposed Methods:
integer query(in QuerySpec query, out QueryResults[] results)
Performs the query specified by query and returns matching documents into results. The maximum number of results is returned.

string addDocument(in integer arcid, in string url, in Document doc)
Adds a document to the specified arcid. The URL of the document is specified in url and the document in doc. The ID of the document is returned. The caller must have write access rights for the archive where the document is stored.

void removeDocument(in string docid)
Removes a document from the archive it belongs. The ID of the document is specified in docid. The caller must have write access rights for the archive where the document is stored.

Document getDocument(in string docid)
Returns the cached copy of the page with specified docid. The document is returned. The caller must have read access rights for the archive where the document is stored.

string getDocumentFFF(in string docid)
Returns the FFF representation of the page with specified docid. The FFF XML is returned. The caller must have read access rights for the archive where the document is stored.

DocumentMetadata getDocumentMetadata(in string docid)
Returns an object that describes the metadata associated to the document (either applied via MetadataTemplates, or stored in the FFF). The caller must have read access rights for the archive where the document is stored.

string getDocumentURL(in string docid)
Returns the URL of the document docid. The caller must have read access rights for the archive where the document is stored.

string normalizeURL(in string url)
Returns the normalized URL.

void applyMetadataTemplate(in string docid, in integer templateid, in MetadataValue[] metadata)
Applies a MetadataTemplate to a document, with the specified variable metadata. The caller must have write access rights for the archive where the document is stored.

void deapplyMetadataTemplate(in string docid, in integer templateid) - Deapplies a metadata template from a document. The caller must have write access rights for the archive where the document is stored.

integer[] enumAppliedMetadataTemplates(in string docid)
Returns the IDs of the applied MetadataTemplates for the specified document. The caller must have read access rights for the archive where the document is stored.

4.3.2 Document

Used to handle documents as opaque objects.

Exposed Properties:
mimeType [string]
MIME type of the document.

size [integer]
size (in bytes) of the document.

Exposed Methods:
void read(in string filename)
Reads document from file.

void write(in string filename)
Writes document to file.

void readFromBuffer(in integer address, in integer size)
Reads a document from a memory buffer at the specified address with the specified size.

void writeToBuffer(in integer address, in integer size)
Writes a document in a memory buffer at the specified address with the specified maximum size.

4.3.3 MetadataValue

Used to hold metadata.

Exposed Properties:
key [string]
Metadata key.

slice [integer]
Slice where metadata is stored.

value [string]
Metadata value.

4.3.4 DocumentMetadata

Used to describe the metadata associated with a document.

Exposed Methods:
string getValue(in string key)
Gets the first value of the metadata with the specified key. Returns and empty string if the metadata was not found.

string[] getValues(in string key)
Gets all the values of the metadata with the specified key.

MetadataValue getMetadataValue(in string key)
Gets the first MetadataValue object for the metadata with the specified key. Returns null if the metadata was not found.

MetadataValue[] getMetadataValues(in string key)
Gets all the MetadataValue objects for the metadata with the specified key.

MetadataValue[] getAllMetadataValues(in string key)
Gets all the MetadataValue objects for every metadata of the document.

4.3.5 QueryParser

Used to specify the query string parser to use.

Static Exposed Properties:
NOPARSER - Don't parse, the query is submitted using QueryAtoms.
RPNPARSER - Use the RPN parser.
ALGPARSER - Use the ALG parser.
NETPARSER - Use the NET parser.

4.3.6 QueryInfo

Used to specify the information returned as result of a query.

Static Exposed Properties:
INFO_NONE - For each result no additional info is returned.
INFO_URL - For each result the URL is returned.
INFO_TITLE - For each result the URL and the title is returned.
INFO_CONTEXT - For each result the URL, the title, the mime type and the contexts where the keywords specified into the query have been found are returned.
INFO_TEMPLATE_METADATA - For each result the URL, the title, the mime type, the contexts where the keywords specified into the query have been found and the metadata added by templates to the document are returned.
INFO_ALL_METADATA - For each result the URL, the title, the mime type, the contexts where the keywords specified into the query have been found and all the metadata of the document are returned.

4.3.7 QueryView

Used to restrict the set of documents returned as result of a query.

Static Exposed Properties:
VIEW_PUBLISHED - The query is applied to all the documents currently in the archive.
VIEW_CORECHANGED - The query is applied only to the documents currently in the archive that have changed in the core of the text. Only applicable to an historicizing archive.

4.3.8 QuerySort

Used to specify the sorting of documents returned as result of a query.

Static Exposed Properties:
SORT_STANDARD - The standard sorting is used.
SORT_RELEVANCE - The documents are ordered by relevance score.
SORT_SCORE - The documents are ordered by their intrinsic score.
SORT_TIME_NEWER - The documents are ordered by change timestamp, more recently changed documents first.
SORT_TIME_OLDER - The documents are reverse-ordered by change timestamp, least recently changed documents first.

4.3.9 QueryAtomType

Used to specify the type of a QueryAtom.

Static Exposed Properties:
ATOM_WORD - QueryAtom is a keyword to find.
ATOM_WILDCARD_WORD - QueryAtom is a keyword with wildcards to find.
ATOM_NOT - QueryAtom is a logic NOT.
ATOM_AND - QueryAtom is a logic AND between other QueryAtoms.
ATOM_OR - QueryAtom is a logic OR between other QueryAtoms.
ATOM_NEAR - QueryAtom is logic NEAR between words.
ATOM_META - QueryAtom is a meta-keyword to find.
ATOM_META_RANGE - QueryAtom is a meta-keyword range to find.
ATOM_WILDCARD_META - QueryAtom is a meta-keyword with wildcards to find.

4.3.10 QueryAtom

Used to build a query in RPN notation.

Exposed Properties:
type [integer]
Current QueryAtom type (from QueryAtomType).

meta [string]
Contains the meta-keyword type. Only for META QueryAtoms.

param [string]
If the current type is WORD or META it contains the keyword (or meta-keyword) to find. Otherwise, if the current type is AND, OR or NEAR it contains the decimal representation of the number of QueryAtom involved in the expression, and if the current type is META_RANGE it

contains the lower boundary of the range. For the NOT type, only the value "1" is allowed in this field.

param1 [string]
If the current type is META_RANGE it contains the upper boundary of the range. Otherwise, if the current type is NEAR it contains the decimal representation of the sloppyness allowed. An empty or 0 sloppyness makes the NEAR match an exact phrase, a sloppyness greater than 0 sets the maximum number of keyword swaps that can be made to match the query.

4.3.11 QuerySliceWeight

Used to specify slice weights.

Exposed Properties:
slice [integer]
Dict ID. The following dict IDs can be used:
1. Author
2. Keyword
3. Abstract
4. Invisible
5. Marginal normal text
6. Marginal emphasized text
7. Marginal link text
8. Marginal remote link text
8. Marginal header text
10. Central normal text
11. Central emphasized text
12. Central link text
13. Central remote link text
14. Central header text
15. Title

weight [integer]
Slice weight. Must be greater or equal to 0.

4.3.12 QuerySpec

Used to submit a query.

Exposed Properties:
archives [array of integers]
IDs of the archives you want to query. Leave empty if you want to query a collection or a watch.

collection [integer]
ID of the collection you want to query. If you want to query archives or a watch, use 0.

watch [integer]
ID of the watch you want to query. If you want to query archives or a collection, use 0.

firstDoc [integer]
Index of the first document (starting from 0) returned. It must be less than lastDoc.

lastDoc [integer]
Index of the last document (starting from 0). It must be greater than firstDoc.

minTime [integer]
Oldest Timestamp (expressed in number of seconds since January 1st 1970 GMT) of query results. All older documents will be rejected.

maxTime [integer]
Newest Timestamp (expressed in number of seconds since January 1st 1970 GMT) of query results. All newer documents will be rejected.

minScore [integer]
Minimum score of query results. All documents with lower score will be rejected.

info [integer from QueryInfo]
Detail level of query results.

view [integer from QueryView]
Document set restrictions of query.

sort [integer from QuerySort]
Result document set sorting type.

parser [integer from QueryParser]
Parser to use to parse the query string.

query [array of QueryAtoms]
Query in RPN notation all list of QueryAtoms. The QueryAtom sequence must produce a stack with only one element. Only used if NOPARSER is specified as QueryParser.

queryString [string]
Query string to be parsed. Only used if RPNPARSER, ALGPARSER or NETPARSER is specified as QueryParser.

weights [array of QuerySliceWeights]
Slice weights. You can pass an empty vector to use the default slice weights. To disable a slice in the current query, you must pass an entry for the slice with a weight of 0. If you don't pass an entry for a certain slice, that slice will have its default weight.

4.3.13 QueryResult

Used to return information on a query result.

Exposed Properties:
id [string]
ID of the document, guaranteed to be unique across archives.

url [string]
URL of the document.

title [string]

Title of the document.

mimeType [string]
Mime type of the document.

contexts [array of strings]
Contexts of the document where the keywords have been found.

timestamp [integer]
Document timestamp (expressed in number of seconds since January 1st 1970 GMT).

serverTime [integer]
Document timestamp (expressed in number of seconds since January 1st 1970 GMT), as reported by the server.

score [integer]
Document score (expressed as percentage * 10000).

archives [array of integers]
The archives the document belongs to.

categories [string]
Private use.

metadata [array of MetadataValues]
Metadata associated with the document.

templates [array of integers]
IDs of the MetadataTemplates applied to the document.

4.3.14 Examples of usage

```
var sb = new AXSearchbox();
sb.host = "localhost";
sb.port = "2200";
sb.username = "admin";
sb.password = "password";
var qs = new QuerySpec();
var a = new Array(1);
a[0] = 1;
qs.archives = a;
qs.parser = QueryParser.ALGPARSER;
qs.info = QueryInfo.INFO_CONTEXT;
qs.view = QueryView.VIEW_PUBLISHED;
qs.sort = QuerySort.SORT_STANDARD;
qs.queryString = "string";
qs.firstDoc = 0;
qs.lastDoc = 1;
var qr = new Array();
var maxres = sb.query(qs, qr);
var i, j; for(i = 0; i < qr.length; ++i)
{

    print(qr[i].id+" "+qr[i].url);
    var doc = sb.getDocument(qr[i].id);
    var links = sb.getDocumentOutlinks(qr[i].id);
    var meta = sb.getDocumentMetadata(qr[i].id);
    var metavalues = meta.getAllValues();
    print("-> "+doc.mimeType+" ["+doc.size+"]");
    for(j = 0; j < links.length; ++j)
    {

        print("O-> "+links[j]);

    }
    if(metavalues != null)
    {

        for(j = 0; j < metavalues.length; ++j)
        {

            print("M-
>"+metavalues[j].key+"["+metavalues[j].slice+"]="+metavalues[j].value);

        }

    }

}
```

4.4 AXMEDIS JAVASCRIPT Functions

The set of JavaScript functions prthat wraps the main classes of the AXMEDIS Object Model includes the:

I/O functions - for managing an instance of the AXMEDIS Object
File functions - for managing files
Dir functions - for managing folders
Process Functions - for managing external process and applications
Mime Type Functions - for converting extension into mime type and viceversa
Example - Example of usage and scripts

4.4.1 I/O functions

print(string msg)
Print a string on the current I/O device. It could be the GUI, the console or other.

4.4.2 File functions

boolean writeToFile(string filePath, string buffer)
Write a string buffer to a file at filePath

boolean appendToFile(string filePath, string buffer)
Append a string buffer to a file at filePath

boolean removeFile(string filePath)
Remove a file

boolean existsFile(string filePath)
Test if a file at filePath exists

string readFromFile(string filePath)
Return a string buffer from a file at filePath

boolean copyFile(string filePath,string targetPath)
Copy the file at filePath to the targetPath returning TRUE in the success case.

boolean renameFile(string filePath,string extension)
Command to change the extension of a file at filePath returning TRUE in the success case.

string getFirstFile(string path,string wildcard)
Provide the first file in the path folder. The wildcard parameter is a filter on files extension ("*.*", "*.txt", etc.). The default value for wildcard parameter is "*.*"If the directory has not files null is returned

string getNextFile(string path,string wildcard)
Provide the first file in the path folder. The wildcard parameter is a filter on files extension ("*.*", "*.txt", etc.). The default value for wildcard parameter is "*.*". If the directory has not files null is returned. This function has to be used in conjuction with the getFirstFile.

array getAllFiles(string path, string wildcard)

Search all files matching the wildcard in the path dir and in all subdir recursively.

4.4.3 Dir functions

boolean createDir(string dirPath)
Create a directory at dirPath returning TRUE in the success case.

boolean removeDir(string dirPath)
Remove the directory at dirPath returning TRUE in the success case.

boolean existsDir(string dirPath)
Test if the directory at dirPath exists

boolean changeDir(string dirPath)
Command to change the folder path into the filesystem returning TRUE in the success case.

array listDir(string path,string wildcard)
Provide the list of directories in the path folder. The wildcard parameter is a filter on the extension of directories ("*.*", "*.txt", etc.). The default value for wildcard parameter is "*.*"

4.4.4 Process Functions

boolean execute(string commandLine)
Invoke the execution of an external process/application by means the command line and parameters. The call is synchronous and returns TRUE in the success case.

4.4.5 Mime Type Functions

string getMimeType(string ext)
Get the mime type from the file extension

string mimeTypeToExt(string mimetype)
Get the extension from the mime type

4.4.6 Examples of usage

```
The script parse an xml string loaded from filesystem the xml is loaded by means
the 'readFromFile' function that returns a string with the xml. The ShowXML
calls the function in the showXML script. It uses the XML parser provided by the
"rexml" script. Such script define an XML object in JS and gives some tools to
browse and retrieve information from the xml elements

var str = readFromFile(xml);
var obj = new AxMetadata();
obj.addXML(str);
var xml = obj.getXML();
ShowXML(xml);
function ShowXML(strXML)
{
    var xmlDoc = new REXML(strXML);
    print("The root element " + xmlDoc.rootElement.name + " has " +
xmlDoc.rootElement.childElements.length + " child elements.");
    for (var i=0; i<xmlDoc.rootElement.childElements.length; i++)
    {
        var item = xmlDoc.rootElement.childElements[i];
        print("Child element of type " + item.type + " "+item.name );
    }
}
The following script shows how to browse directories in the unit C:
var l = listDir("c:\\"); //It provides the list of folder in c:
for (i in l)
{
    print(l[i]);
    var f = getFirstFile(l[i]);
    while(f!=null){
        print(f); // here, operations to do with the file name
        f = getNextFile();
    }
}
```

4.5 Formatting Javascript Classes

To be completed

4.6 License Javascript Classes

To be completed

4.7 Profiling Javascript Classes

To be completed

4.8 Selection Javascript Classes

To be completed

5 AXMEDIS Content Processing Functionalities in Plug-ins

5.1 Audio Adaptation plug-in

Category: ContentProcessing
Identifier: AudioAdaptation
Library: AudioAdaptation Version: 1.001
Vendor: Axmedis
Main Library: audioadaptationplugin.dll

Description: Plugin for audio processing

FunctionList:
 * FFAudioTranscoding
 * LSAudioTranscoding

5.1.1 FFAudioTranscoding

STRING FFAudioTranscoding (RESOURCE InputResource, STRING Mimetype, RESOURCE OutputResource, UINT32 OutputSamplingRate, UINT16 OutputNumChannels, UINT16 OutputBitRate, FLOAT ReadStartingTime, FLOAT ReadEndingTime, STRING OutputCodec)

Version: 1.0
Description: Transcoding of audio files by FFMPEG
Parameter List
Name: InputResource
Description: The Resource to be converted
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: audio
Resource Format: x-wav x-ms-wma basic x-mpeg x-vorbis x-pn-realaudio x-ac3 x-dv x-mace x-adpcm x-aac 32KADPCM amr x-pn-realaudio video
Resource Format: x-mpeg x-mpeg2 mp4 x-raw x-h263 x-mjpeg x-ms-wmv x-ms-asf x-flv x-svq x-dv x-h264 x-indeo x-vp3 x-ffv x-vcr x-msvideo x-nut application
Resource Format: x-pcm vnd.rn-realmedia
Ranges:
Name: Mimetype
Description: Mimetype for output resource
Paramater Type STRING
Default Value:
Constraints:
Name: OutputResource
Description: Where the produced resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Name: OutputSamplingRate
Description: Sampling rate of the output audio file (default: sampling rate of the input)
Paramater Type UINT32
Default Value:

Constraints:

Resource Type:

Ranges:

Name: OutputNumChannels

Description: Number of channels of the output audio file (default: number of channels of the input)

Paramater Type UINT16

Default Value:

Constraints:

Resource Type:

Ranges:

Name: OutputBitRate

Description: Bit rate of the output audio file - Only applies to compressed audio file formats (default: 64 kb)

Paramater Type UINT16

Default Value:

Constraints:

Resource Type:

Ranges:

Name: ReadStartingTime

Description: Starting time in the input audio file (default: beginning of the file)

Paramater Type FLOAT

Default Value:

Constraints:

Resource Type:

Ranges:

Name: ReadEndingTime

Description: Ending time in the input audio file (default: end of the file)

Paramater Type FLOAT

Default Value:

Constraints:

Resource Type:

Ranges:

Name: OutputCodec

Description: Codec of the output audio file (default: depends on the desired format of the output)

Paramater Type STRING

Default Value:

Constraints:

Resource Type:

Ranges:

Result: Result

Result type: STRING

Result Description: The result of import, SUCCESS if ok, ERROR followed by a message in case of error

5.1.2 LSAudioTranscoding

STRING LSAudioTranscoding (RESOURCE InputResource, STRING Mimetype, RESOURCE OutputResource, FLOAT ReadStartingTime, FLOAT ReadEndingTime, STRING OutputCodec)

Version: 1.0
Description: Transcoding of audio files by LibSnd
Parameter List
Name: InputResource
Description: The Resource to be converted
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: audio
Resource Format: x-wav x-aiff basic x-paris x-svx x-nist x-voc x-ircam x-w64 x-sd2 x-flac application
Resource Format: x-pcm x-pagerecall
Ranges:
Name: Mimetype
Description: Mimetype for output resource
Paramater Type STRING
Default Value:
Constraints:
Name: OutputResource
Description: Where the produced resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Name: ReadStartingTime
Description: Starting time in the input audio file (default: beginning of the file)
Paramater Type FLOAT
Default Value:
Constraints:
Resource Type:
Ranges:
Name: ReadEndingTime
Description: Ending time in the input audio file (default: end of the file)
Paramater Type FLOAT
Default Value:
Constraints:
Resource Type:
Ranges:
Name: OutputCodec
Description: Codec of the output audio file (default: depends on the desired format of the output)
Paramater Type STRING
Default Value:
Constraints:
Resource Type:
Ranges:
Result: Result
Result type: STRING
Result Description: The result of import, SUCCESS if ok, ERROR followed by a message in case of error

5.2 Audio Descriptor plug-in

Category: ContentProcessing
Identifier: AudioDescriptor
Library: AudioDescriptor **Version:** 1.001
Vendor: Axmedis
Main Library: audiodescriptorplugin.dll

Description: Plug-in for audio descriptors extraction

FunctionList:
* LowLevelDescriptors
* Segmentation
* TempoEstimation
* MusicGenreEstimation

5.2.1 LowLevelDescriptors

STRING LowLevelDescriptors (RESOURCE InputResource, FLOAT HopSize, UINT16 AudioPower, UINT16 SpectralCentroid, UINT16 SpectralSpread, UINT16 SpectralEnvelope, FLOAT EnvLoEdge, FLOAT EnvHiEdge, FLOAT BandsPerOctave, UINT16 SpectralFlatness, FLOAT FlatLoEdge, FLOAT FlatHiEdge, UINT16 ScaleRatio, UINT16 EvalMeans, UINT16 EvalVariances, RESOURCE OutputResource)

Version: 1.0
Description: Extracts MPEG-7 Low Level Descriptors from an audio file
Parameter List
Name: InputResource
Description: The Resource to be analyzed
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: audio
Resource Format: x-wav x-aiff wav aiff basic
Ranges:
Name: HopSize
Description: Time (in seconds) between successive estimation of descriptors
Paramater Type FLOAT
Default Value:
Constraints:
Resource Type:
Ranges:
Name: AudioPower
Description: Compute AudioPower if set to 1
Paramater Type UINT16
Default Value:
Constraints:
Resource Type:
Ranges:
Name: SpectralCentroid
Description: Compute SpectralCentroid if set to 1
Paramater Type UINT16
Default Value:

Constraints:
Resource Type:
Ranges:
Name: SpectralSpread
Description: Compute SpectralSpread if set to 1
Paramater Type UINT16
Default Value:
Constraints:
Resource Type:
Ranges:
Name: SpectralEnvelope
Description: Compute SpectralEnvelope if set to 1
Paramater Type UINT16
Default Value:
Constraints:
Resource Type:
Ranges:
Name: EnvLoEdge
Description: Lower edge of logarithmically-spaced frequency bands
Paramater Type FLOAT
Default Value:
Constraints:
Resource Type:
Ranges:
Name: EnvHiEdge
Description: Higher edge of logarithmically-spaced frequency bands
Paramater Type FLOAT
Default Value:
Constraints:
Resource Type:
Ranges:
Name: BandsPerOctave
Description: Frequency resolution of logarithmic spectrum [0.125,0.25,0.5,1,2,4,8,16,32]
Paramater Type FLOAT
Default Value:
Constraints:
Resource Type:
Ranges:
Name: SpectralFlatness
Description: Compute SpectralFlatness if set to 1
Paramater Type UINT16
Default Value:
Constraints:
Resource Type:
Ranges:
Name: FlatLoEdge
Description: Lower edge of logarithmically-spaced frequency bands
Paramater Type FLOAT
Default Value:
Constraints:
Resource Type:

Ranges:
Name: FlatHiEdge
Description: Higher edge of logarithmically-spaced frequency bands
Paramater Type FLOAT
Default Value:
Constraints:
Resource Type:
Ranges:
Name: ScaleRatio
Description: Scaling Ratio for scaling operations (means and variances)
Paramater Type UINT16
Default Value:
Constraints:
Resource Type:
Ranges:
Name: EvalMeans
Description: Compute means of descriptors if set to 1
Paramater Type UINT16
Default Value:
Constraints:
Resource Type:
Ranges:
Name: EvalVariances
Description: Compute variances of descriptors if set to 1
Paramater Type UINT16
Default Value:
Constraints:
Resource Type:
Ranges:
Name: OutputResource
Description: Where the produced MPEG-7 description will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: SUCCESS or ERROR followed by a message in case of error

5.2.2 Segmentation

STRING Segmentation (RESOURCE InputResource, RESOURCE OutputResource)

Version: 1.0
Description: Detect silence/noise/speech/music segments
Parameter List
Name: InputResource
Description: The Resource to be analyzed
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: audio

Resource Format: x-wav x-aiff wav aiff basic
Ranges:
Name: OutputResource
Description: Where the produced MPEG-7 description will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: SUCCESS or ERROR followed by a message in case of error

5.2.3 TempoEstimation

STRING TempoEstimation (RESOURCE InputResource, FLOAT BpmLoLimit, FLOAT BpmHiLimit, RESOURCE OutputResource)

Version: 1.0
Description: Estimate tempo (in BPM) of a music file
Parameter List
Name: InputResource
Description: The Resource to be analyzed
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: audio
Resource Format: x-wav x-aiff wav aiff basic
Ranges:
Name: BpmLoLimit
Description: Minimum acceptable tempo in beats per minute
Paramater Type FLOAT
Default Value:
Constraints:
Resource Type:
Ranges:
Name: BpmHiLimit
Description: Maximum acceptable tempo in beats per minute
Paramater Type FLOAT
Default Value:
Constraints:
Resource Type:
Ranges:
Name: OutputResource
Description: Where the produced MPEG-7 description will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: SUCCESS or ERROR followed by a message in case of error

5.2.4 MusicGenreEstimation

STRING MusicGenreEstimation (RESOURCE InputResource, RESOURCE OutputResource)

Version: 1.0
Description: Estimate the music genre of a music file
Parameter List
Name: InputResource
Description: The Resource to be analyzed
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: audio
Resource Format: x-wav x-aiff wav aiff basic
Ranges:
Name: OutputResource
Description: Where the produced MPEG-7 description will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: SUCCESS or ERROR followed by a message in case of error

5.3 Audio Fingerprint Extraction Plug-in

Category: ContentProcessing
Identifier: AudioFingerprintExtraction
Library: AudioFingerprintExtraction **Version:** 1.001
Vendor: Axmedis
Main Library: audioFPplugin.dll

Description: Audio Fingerprint Extraction

FunctionList:

- AxAFPExtract

5.3.1 AxAFPExtract

STRING AxAFPExtract (RESOURCE InputResource, STRING Mimetype, RESOURCE OutputResource, INT32 nFeatures, INT32 frameSize, INT32 frameShift, INT32 offset)

Version: 1.0
Description: Extracts a fingerprint of the first audio stream in a given Multimedia File(Audio or Video).
Parameter List
Name: InputResource
Description: The Resource to extract the fingerprint from
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: audio
Resource Format: x-wav wav x-aiff x-ms-wma mpeg video
Resource Format: x-msvideo mpeg x-ms-wmv quicktime avi
Ranges:
Name: Mimetype
Description: Mimetype for the output resource
Paramater Type STRING
Default Value:
Constraints:
Name: OutputResource
Description: Where the produced resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Name: nFeatures
Description: Number of Features for the finger print. 18 is the desired Standard
Paramater Type INT32
Default Value:
Constraints:
Resource Type:
Ranges:
Name: frameSize
Description: Size of the windowing Size for the Subfingerprints
Paramater Type INT32
Default Value:
Constraints:
Resource Type:
Ranges:

Name: frameShift
Description: Frame overlap for the subfingerprints
Paramater Type INT32
Default Value:
Constraints:
Resource Type:
Ranges:
Name: offset
Description: Frame offset for the fingerprint calculation
Paramater Type INT32
Default Value:
Constraints:
Resource Type:
Ranges:
Result: Result
Result type: STRING
Result Description: The result of import, SUCCESS if ok, ERROR followed by a message in case of error

5.4 Cryptography Adaptation Plug-in

Category: ContentProcessing
Identifier: CryptographyAdaptation
Library: CryptographyAdaptation **Version:** 1.001
Vendor: Axmedis
Main Library: cryptlibplugin.dll

Description: Plugin for security processing

FunctionList:

- CryptProcess

5.4.1 CryptProcess

STRING CryptProcess (RESOURCE InputResource, STRING Mimetype, RESOURCE OutputResource, STRING AlgorithmName, UINT16 Keysize, STRING AlgorithmMode, STRING KeyInput, STRING KeyIVInput, STRING EncDec)

Version: 1.0
Description: Crypto processing of files
Parameter List
Name: InputResource
Description: The Resource to be procesed
Paramater Type RESOURCE
Default Value:
Constraints:
Name: Mimetype
Description: Mimetype for output resource
Paramater Type STRING
Default Value:
Constraints:
Name: OutputResource
Description: Where the produced resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Name: AlgorithmName
Description: Algorithm Name (AES,DES,3DES,CAST-128,Blowfish)
Paramater Type STRING
Default Value:
Constraints:
Resource Type:
Ranges:
Name: Keysize
Description: Key size (Number of BYTE among 8 and 16)
Paramater Type UINT16
Default Value:
Constraints:
Resource Type:
Ranges:
Name: AlgorithmMode
Description: Algoritm Mode (ECB, CBC,OFB CFB)
Paramater Type STRING
Default Value:

Constraints:
Resource Type:
Ranges:
Name: KeyInput
Description: Input Key
Paramater Type STRING
Default Value:
Constraints:
Resource Type:
Ranges:
Name: KeyIVInput
Description: IV Input Key (Number of BYTE)
Paramater Type STRING
Default Value:
Constraints:
Name: EncDec
Description: Switch for encryption-decryption (Encrypt,Decrypt))
Paramater Type STRING
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of import, SUCCESS if ok, ERROR followed by a message in case of error

5.5 Text Descriptor Plug-in

Category: ContentProcessing
Identifier: TextDescriptor
Library: Descriptor **Version:** 1.001
Vendor: Axmedis
Main Library: descriptorextractorplugin.dll

Description: Plugin for descriptors extraction from text documents

FunctionList:

- KWFromComparisons
- KWFromSemanticAnalysis

5.5.1 KWFromComparisons

STRING KWFromComparisons (RESOURCE InputResource, UINT16 MaxKWNumber, STRING Keywords)

Version: 1.0
Description: Retrieves the keywords exploiting frequency lists corpus only.
Parameter List
Name: InputResource
Description: The Resource to be converted
Paramater Type RESOURCE
Default Value:
Constraints:
Name: MaxKWNumber
Description: How many keyword requested as a maximum.
Paramater Type UINT16
Default Value:
Constraints:
Name: Keywords
Description: A string containing keywords separated by '@@@'
Paramater Type STRING
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.5.2 KWFromSemanticAnalysis

STRING KWFromSemanticAnalysis (RESOURCE InputResource, UINT16 MaxKWNumber, STRING Keywords)

Version: 1.0
Description: Retrieves the keywords exploiting frequency lists corpus and WordNet synsets relations.
Parameter List
Name: InputResource
Description: The Resource to be converted
Paramater Type RESOURCE
Default Value:
Constraints:
Name: MaxKWNumber

Description: How many keyword requested as a maximum.
Paramater Type UINT16
Default Value:
Constraints:

Name: Keywords
Description: A string containing keywords separated by '@@@'
Paramater Type STRING
Default Value:
Constraints:

Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.6 Text to Docs Adaptation Plug-in

Category: ContentProcessing
Identifier: TextDocsAdaptaion
Library: Adaptation **Version:** 1.001
Vendor: Axmedis
Main Library: documentadaptation.dll

Description: Plugin for transcoding text documents

FunctionList:

- DocumentConversion

5.6.1 DocumentConversion

STRING DocumentConversion (RESOURCE InputResource, STRING ConversionFormat, RESOURCE OutputResource)

Version: 1.0
Description: Transcodes the given text document in the supplied format.
Parameter List
Name: InputResource
Description: The Resource to be converted
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: application
Resource Format: pdf text
Resource Format: html
Ranges:
Name: ConversionFormat
Description: The format to which the resource will be converted.
Paramater Type STRING
Default Value:
Constraints:
Name: OutputResource
Description: Where the produced resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.7 Multimedia Adaptation plug-in

Category: ContentProcessing
Identifier: MultimediaAdaptation
Library: MultimediaAdaptation Version: 1.001
Vendor: Axmedis
Main Library: multimediaadaptationplugin.dll

Description: Plugin for multimedia processing

FunctionList:

- Mp4To3gp
- Mp4ToISMA
- AddMultimediaFiles
- ToMp4
- ExtractMediaTrack
- CatMultimediaFiles
- DelayTrack
- RemoveTrack
- ExtractFromStartToEnd
- Mp4ToAvi

5.7.1 Mp4To3gp

STRING Mp4To3gp (RESOURCE InputResource, RESOURCE OutputResource, BOOLEAN KeepSys)

Version: 1.0
Description: Convert an mp4 file to 3gp
Parameter List
Name: InputResource
Description: Input File to be converted
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: audio
Resource Format: mp4 video
Resource Format: mp4 mp4v-es
Ranges:
Name: OutputResource
Description: Where the produced resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type:
Ranges:
Name: KeepSys
Description: Keep systems tracks
Paramater Type BOOLEAN
Default Value:
Constraints:
Resource Type:
Ranges:
Result: Result
Result type: STRING

Result Description: The result of 3gp conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.7.2 Mp4ToISMA

STRING Mp4ToISMA (RESOURCE InputResource, RESOURCE OutputResource)

Version: 1.0
Description: Convert an mp4 file to ISMA specification
Parameter List
Name: InputResource
Description: File to be converted
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: audio
Resource Format: mp4 video
Resource Format: mp4 mp4v-es
Ranges:
Name: OutputResource
Description: Where the produced resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type:
Ranges:
Result: Result
Result type: STRING
Result Description: The result of ISMA conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.7.3 AddMultimediaFiles

STRING AddMultimediaFiles (RESOURCE InputResource, RESOURCE BaseResource, UINT32 Delay, DOUBLE ImportLength, STRING TrackID, UINT32 FPS, STRING Lang, RESOURCE OutputResource)

Version: 1.0
Description: Import a media file into a multimedia file (samples are added to new tracks)
Parameter List
Name: InputResource
Description: The media to add
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: audio
Resource Format: x-mpeg x-aac amr evrc smv x-vorbis x-gsm video
Resource Format: x-media x-nhnt x-info x-cmp x-m4v x-h263 x-h264 x-msvideo mpeg dvd mp4 mp4v-es 3gpp xmt xmta mp4-bt vrml image
Resource Format: jpeg png application
Resource Format: x-cdlink text
Resource Format: srt sub ttxt texml model
Resource Format: x3d+xml x3d+vrml
Ranges:
Name: BaseResource

- **Description:** Base resource where to add the new media
- **Paramater Type** RESOURCE
- **Default Value:**
- **Constraints:**
  - **Resource Type:** audio
  - **Resource Format:** x-mpeg x-aac amr evrc smv x-vorbis x-gsm video
  - **Resource Format:** x-media x-nhnt x-info x-cmp x-m4v x-h263 x-h264 x-msvideo mpeg dvd mp4 mp4v-es 3gpp xmt xmta mp4-bt vrml image
  - **Resource Format:** jpeg png application
  - **Resource Format:** x-cdlink text
  - **Resource Format:** srt sub ttxt texml model
  - **Resource Format:** x3d+xml x3d+vrml
- **Ranges:**

Name: Delay

- **Description:** Delay in milliseconds of the new track
- **Paramater Type** UINT32
- **Default Value:**
- **Constraints:**
  - **Resource Type:**
- **Ranges:**

Name: ImportLength

- **Description:** Number of seconds to import from input file (starting from the beginning)
- **Paramater Type** DOUBLE
- **Default Value:**
- **Constraints:**
  - **Resource Type:**
- **Ranges:**

Name: TrackID

- **Description:** Track to extract
- **Paramater Type** STRING
- **Default Value:**
- **Constraints:**
  - **Resource Type:**
- **Ranges:**

Name: FPS

- **Description:** Frames per sample
- **Paramater Type** UINT32
- **Default Value:**
- **Constraints:**
  - **Resource Type:**
- **Ranges:**

Name: Lang

- **Description:** Language code
- **Paramater Type** STRING
- **Default Value:**
- **Constraints:**
  - **Resource Type:**
- **Ranges:**

Name: OutputResource

- **Description:** Where the produced resource will be stored
- **Paramater Type** RESOURCE
- **Default Value:**
- **Constraints:**
  - **Resource Type:**
- **Ranges:**

Result: Result

Result type: STRING

Result Description: The result of adding files, SUCCESS if ok, ERROR followed by a message in case of error

5.7.4 ToMp4

STRING ToMp4 (RESOURCE InputResource, RESOURCE OutputResource)

Version: 1.0
Description: Import a media file into a multimedia file (samples are added to new tracks)
Parameter List
Name: InputResource
Description: The Resource to be converted
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: audio
Resource Format: x-mpeg x-aac amr evrc smv x-vorbis x-gsm video
Resource Format: x-media x-nhnt x-info x-cmp x-m4v x-h263 x-h264 x-msvideo mpeg dvd mp4 mp4v-es 3gpp xmt xmta mp4-bt vrml image
Resource Format: jpeg png application
Resource Format: x-cdlink text
Resource Format: srt sub ttxt texml model
Resource Format: x3d+xml x3d+vrml
Ranges:
Name: OutputResource
Description: Where the produced resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type:
Ranges:
Result: Result
Result type: STRING
Result Description: The result of mp4 conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.7.5 ExtractMediaTrack

STRING ExtractMediaTrack (RESOURCE InputResource, RESOURCE OutputResource, UINT32 TrackID, STRING Mimetype)

Version: 1.0
Description: Extracts a track into a new file
Parameter List
Name: InputResource
Description: Input file to extract track
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: audio
Resource Format: mp4 video
Resource Format: mp4 mp4v-es 3gpp
Ranges:
Name: OutputResource

Description: Where the produced resource (track) will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type:
Ranges:
Name: TrackID
Description: Track to extract
Paramater Type UINT32
Default Value:
Constraints:
Resource Type:
Ranges:
Name: Mimetype
Description: Mimetype for output resource
Paramater Type STRING
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of the extraction, SUCCESS if ok, ERROR followed by a message in case of error

5.7.6 CatMultimediaFiles

STRING CatMultimediaFiles (RESOURCE InputResourceA, RESOURCE InputResourceB, RESOURCE OutputResource)

Version: 1.0
Description: Concatenates two multimedia files
Parameter List
Name: InputResourceA
Description: Input file A to concatenate
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: audio
Resource Format: x-mpeg x-aac amr evrc smv x-vorbis x-gsm video
Resource Format: x-media x-nhnt x-info x-cmp x-m4v x-h263 x-h264 x-msvideo mpeg dvd mp4 mp4v-es 3gpp xmt xmta mp4-bt vrml image
Resource Format: jpeg png application
Resource Format: x-cdlink text
Resource Format: srt sub ttxt texml model
Resource Format: x3d+xml x3d+vrml
Ranges:
Name: InputResourceB
Description: Input file B to concatenate
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: audio
Resource Format: x-mpeg x-aac amr evrc smv x-vorbis x-gsm video
Resource Format: x-media x-nhnt x-info x-cmp x-m4v x-h263 x-h264 x-msvideo mpeg dvd mp4 mp4v-es 3gpp xmt xmta mp4-bt vrml image
Resource Format: jpeg png application

Resource Format: x-cdlink text
Resource Format: srt sub ttxt texml model
Resource Format: x3d+xml x3d+vrml
Ranges:
Name: OutputResource
Description: Result file
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type:
Ranges:
Result: Result
Result type: STRING
Result Description: The result of the concatenation, SUCCESS if ok, ERROR followed by a message in case of error

5.7.7 DelayTrack

STRING DelayTrack (RESOURCE InputResource, UINT32 Delay, STRING TrackID, RESOURCE OutputResource)

Version: 1.0
Description: Delay a track
Parameter List
Name: InputResource
Description: File to be converted
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: audio
Resource Format: mp4 video
Resource Format: mp4 mp4v-es 3gpp
Ranges:
Name: Delay
Description: Delay in milliseconds of the track
Paramater Type UINT32
Default Value:
Constraints:
Resource Type:
Ranges:
Name: TrackID
Description: Track to delay
Paramater Type STRING
Default Value:
Constraints:
Resource Type:
Ranges:
Name: OutputResource
Description: Where the produced resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type:
Ranges:
Result: Result

Result type: STRING
Result Description: The result of track delay, SUCCESS if ok, ERROR followed by a message in case of error

5.7.8 RemoveTrack

STRING RemoveTrack (RESOURCE InputResource, STRING TrackID, RESOURCE OutputResource)

Version: 1.0
Description: Remove a track
Parameter List
Name: InputResource
Description: File to be converted
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: audio
Resource Format: mp4 video
Resource Format: mp4 mp4v-es 3gpp
Ranges:
Name: TrackID
Description: Track to remove
Paramater Type STRING
Default Value:
Constraints:
Resource Type:
Ranges:
Name: OutputResource
Description: Where the produced resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type:
Ranges:
Result: Result
Result type: STRING
Result Description: The result of track removing, SUCCESS if ok, ERROR followed by a message in case of error

5.7.9 ExtractFromStartToEnd

STRING ExtractFromStartToEnd (RESOURCE InputResource, DOUBLE Start, DOUBLE End, RESOURCE OutputResource)

Version: 1.0
Description: Extracts a file from start to end
Parameter List
Name: InputResource
Description: File to be converted
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: audio
Resource Format: mp4 video
Resource Format: mp4 mp4v-es 3gpp

Ranges:
Name: Start
Description: Start of extraction in seconds
Paramater Type DOUBLE
Default Value:
Constraints:
Resource Type:
Ranges:
Name: End
Description: End of extraction in seconds
Paramater Type DOUBLE
Default Value:
Constraints:
Resource Type:
Ranges:
Name: OutputResource
Description: Where the produced resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type:
Ranges:
Result: Result
Result type: STRING
Result Description: The result of the extraction by time, SUCCESS if ok, ERROR followed by a message in case of error

5.7.10 Mp4ToAvi

STRING Mp4ToAvi (RESOURCE InputResource, FLOAT FPS, UINT32 Width, UINT32 Height, RESOURCE OutputResource)

Version: 1.0
Description: Convert a MP4 BIFS pure file to AVI
Parameter List
Name: InputResource
Description: Input MP4 BIFS pure file(no audio, no image, no video) to convert to AVI
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: video
Resource Format: mp4 mp4v-es
Ranges:
Name: FPS
Description: Extraction framerate (default:0 computed from the BIFS track duration)
Paramater Type FLOAT
Default Value:
Constraints:
Resource Type:
Ranges:
Name: Width
Description: Width of the bifs scene (default:0 original size)
Paramater Type UINT32
Default Value:
Constraints:

Resource Type:
Ranges:
Name: Height
Description: Height of the bifs scene (default:0 original size)
Paramater Type UINT32
Default Value:
Constraints:
Resource Type:
Ranges:
Name: OutputResource
Description: Result file
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type:
Ranges:
Result: Result
Result type: STRING
Result Description: The result of the conversion to avi, SUCCESS if ok, ERROR followed by a message in case of error

5.8 Ringtone Adaptation plug-in

Category: ContentProcessing
Identifier: RingtoneAdaptation
Library: RingtoneAdaptation **Version:** 1.001
Vendor: Axmedis
Main Library: ringtoneadaptationplugin.dll

Description: Plugin for Ring Tone processing

FunctionList:
- convert
- resample
- getInfo
- clip

5.8.1 convert

STRING convert (RESOURCE InputResource, STRING Mimetype, RESOURCE OutputResource)

Version: 1.0
Description: Convert from one format to another
Parameter List
Name: InputResource
Description: The Resource to be converted
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: audio
Resource Format: x-wav x-aiff x-ms-wma basic mpeg mid
Ranges:
Name: Mimetype
Description: Mimetype for output resource
Paramater Type STRING
Default Value:
Constraints:
Name: OutputResource
Description: Where the produced resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of import, SUCCESS if ok, ERROR followed by a message in case of error

5.8.2 resample

STRING resample (RESOURCE InputResource, STRING Mimetype, RESOURCE OutputResource, UINT32 OutputSamplingRate, UINT16 OutputNumChannels, UINT16 OutputBitRate)

Version: 1.0
Description: Resample the input file (ie changing frequecy, bitrate etc)
Parameter List
Name: InputResource

Description: The Resource to be converted
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: audio
Resource Format: x-wav x-aiff x-ms-wma basic mpeg mid
Ranges:
Name: Mimetype
Description: Mimetype for output resource
Paramater Type STRING
Default Value:
Constraints:
Name: OutputResource
Description: Where the produced resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Name: OutputSamplingRate
Description: Sampling rate of the output audio file (default: sampling rate of the input)
Paramater Type UINT32
Default Value:
Constraints:
Resource Type:
Ranges:
Name: OutputNumChannels
Description: Number of channels of the output audio file (default: number of channels of the input)
Paramater Type UINT16
Default Value:
Constraints:
Resource Type:
Ranges:
Name: OutputBitRate
Description: Bit rate of the output audio file - Only applies to compressed audio file formats (default: 64 kb)
Paramater Type UINT16
Default Value:
Constraints:
Resource Type:
Ranges:
Result: Result
Result type: STRING
Result Description: The result of import, SUCCESS if ok, ERROR followed by a message in case of error

5.8.3 getInfo

STRING getInfo (RESOURCE InputResource, UINT32 SamplingRate, UINT16 NumChannels, UINT16 BitRate, STRING Duration)

Version: 1.0
Description: Get all the information about the input Ring Tone
Parameter List
Name: InputResource
Description: The Resource to be converted
Paramater Type RESOURCE

Default Value:
Constraints:
Resource Type: audio
Resource Format: x-wav x-aiff x-ms-wma basic mpeg mid
Ranges:
Name: SamplingRate
Description: Sampling rate of the input ring tone
Paramater Type UINT32
Default Value:
Constraints:
Resource Type:
Ranges:
Name: NumChannels
Description: Number of channels of the input ring tone
Paramater Type UINT16
Default Value:
Constraints:
Resource Type:
Ranges:
Name: BitRate
Description: Bit rate of the input ring tone - (default: 64 kb)
Paramater Type UINT16
Default Value:
Constraints:
Resource Type:
Ranges:
Name: Duration
Description: Duration of the Ringtone File
Paramater Type STRING
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of the operation, SUCCESS if ok, ERROR followed by a message in case of error

5.8.4 clip

STRING clip (RESOURCE InputResource, STRING Mimetype, FLOAT ReadStartingTime, FLOAT ReadEndingTime)

Version: 1.0
Description: Clip the file for the specified time (for e.g. reducing it to a 30 sec clip)
Parameter List
Name: InputResource
Description: The Resource to be converted
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: audio
Resource Format: x-wav x-aiff x-ms-wma basic mpeg mid
Ranges:
Name: Mimetype
Description: Mimetype for output resource
Paramater Type STRING

Default Value:

Constraints:

Name: ReadStartingTime

Description: Starting time for the clip(default: beginning of the file)

Paramater Type FLOAT

Default Value:

Constraints:

Resource Type:

Ranges:

Name: ReadEndingTime

Description: Ending time for the clip (default: end of the file)

Paramater Type FLOAT

Default Value:

Constraints:

Resource Type:

Ranges:

Result: Result

Result type: STRING

Result Description: The result of import, SUCCESS if ok, ERROR followed by a message in case of error

5.9 Metadata Mapping Plugin

To be added in the next version

5.10 Audio recognition and monitoring Plugin

To be added in the next version

5.11 Image Processing Plugin

This plug in uses the ImageMagick Library, please for additional information refer to user manual of that library and tools.

5.11.1 Main functionalities

The image processing plug-in allows adapting image resources to various use case. For example it can be used to convert different image formats, to apply various effects, to resize, to mirror, etc. In total the plug-in is composed of forty-one functions that are:

- **Conversion**, to convert the image
- **Import**, to import an image
- **Resize**, to resize the image
- **Contrast**, to change the image contrast
- **Edge**, to highlight edges of the image
- **Embross**, to highlight edges with 3D effect
- **Blur**, to blur the image
- **GaussianBlur**, to apply a Gaussian Blur to the image
- **Median**, to apply a median filter to the image
- **Mirror**, to mirror the image
- **Noise**, to add noise in the image
- **Despeckle**, to reduce the noise from the image using the despeckle filter
- **Equalize**, to apply an histogram equalization to the image
- **Enhance**, to minimize the noise of the image
- **ExtractChannel**,
- **GrayScale**, to convert a coloured image to grayscale
- **Magnify**, to scale up the image
- **Minimize**, to scale down the image
- **Modulate**, to modulate hue, saturation, and brightness of the image
- **Monochrome**, to create a monochrome image
- **Negate**, to negate colours in the image
- **Normalize**, to increase contrast by normalizing the pixel values
- **OilPaint**, to create a image looks like oil painting
- **Quality**, to change the JPEG/MIFF/PNG compression
- **Quantize**, to set the preferred number of colours in the image
- **Raise**, to highlight or dark the edges of an image to give a 3D raised or lowered effect
- **ReduceNoise**, to reduce the noise of the image
- **Replace**, to replace the image
- **FloodFill**, to apply a flood-fill texture
- **Rool**, to roll the image by a specified number of columns and rows
- **Rotate**, to rotate the image specifying a number of degrees

- **Scale**, to scale the image by using a specified ratio
- **Shear**, to create a parallelogram by sliding the image by X of Y axis
- **Shade**, to shade the image using distant light source
- **Spread**, to spread pixels randomly
- **SetOpacity to set the opacity of the image**
- **SubImage**,
- **GetInfo**, to see information about the image
- **SetMaskColour,**
- **Paste,** to paste the image
- **Test,** to test the image

A More detailed description of these functionalities is available in section **Errore. L'origine riferimento non è stata trovata.**

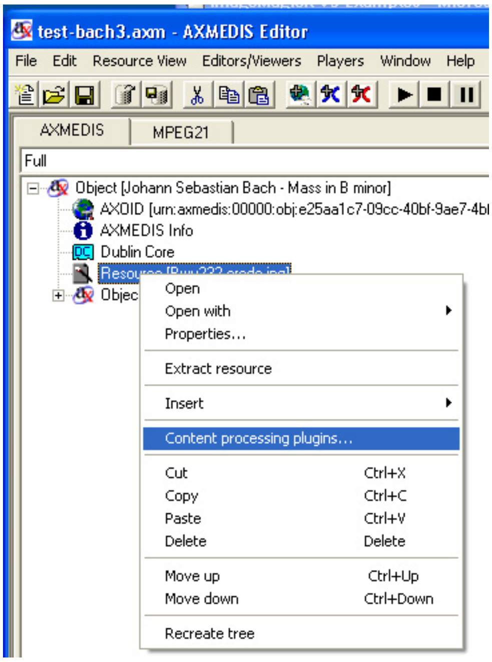
5.11.2 Relationship with other tools

This tool is implemented as a plug-in. Like other plug-ins, its functionality is available via the AXMEDIS Editor and the AXMEDIS Processing Engine.

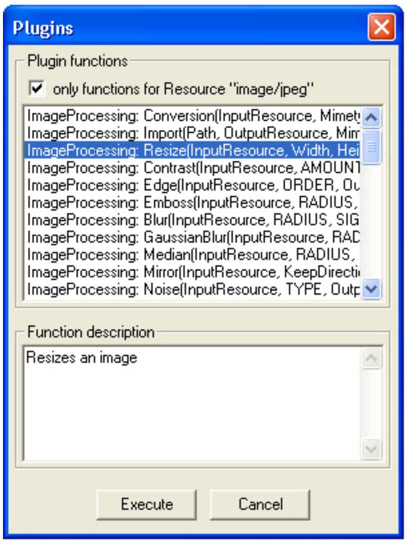
5.11.3 Detailed description of the functionalities and Screenshots

Here's an example on how to use the plug-in with AXMEDIS Editor.

The plug-in can be applied to all images resources in all formats embedded into an AXMEDIS object. Selecting one resource in the tree and right clicking, select **Content Processing plugins…**



A new dialog will appear will the list of available functionalities. Selecting a functionality will a appear a brief description in the **Function description** box.



Plugin functions

☑ only functions for Resource "image/jpeg"

ImageProcessing: Conversion(InputResource, Mimet
ImageProcessing: Import(Path, OutputResource, Mim
ImageProcessing: Resize(InputResource, Width, Hei
ImageProcessing: Contrast(InputResource, AMOUNT
ImageProcessing: Edge(InputResource, ORDER, Ou
ImageProcessing: Emboss(InputResource, RADIUS,
ImageProcessing: Blur(InputResource, RADIUS, SIG
ImageProcessing: GaussianBlur(InputResource, RAD
ImageProcessing: Median(InputResource, RADIUS,
ImageProcessing: Mirror(InputResource, KeepDirecti
ImageProcessing: Noise(InputResource, TYPE, Outp

Function description

Resizes an image

Selecting the appropriate function and pressing the **Execute** button a new dialog appears with a number of fields to be filled-in. the aspect of the dialog and the number of fields is different for each function.

Please, refer to section ?? for a detailed description of the values needed for each functionalities.

In the **Output Resource** cascading menu is possible to decide if the function will produce a new resource or will overwrite the old one.

Here's a brief analysis of image processing functionalities.

Since the image processing plug-in is based on the GPL source code of ImageMagick, for a more detailed description of these functionalities, please refer to the following links:

- ImageMagick website: http://www.imagemagick.org/script/index.php
- **The Definitive Guide to ImageMagick** by Michael Still, available on: http://www.amazon.com/Definitive-Guide-ImageMagick/dp/1590595904/sr=8-1/qid=1157030444/ref=pd_bbs_1/104-0533291-5821542?ie=UTF8
- Examples of ImageMagick usage are available here: http://www.cit.gu.edu.au/~anthony/graphics/imagick6

5.11.3.1 Conversion

STRING Conversion (RESOURCE InputResource, STRING Mimetype, RESOURCE OutputResource)

Version: 1.0

Description: Convert an image

Parameter List

Name: InputResource

Description: The Resource to be converted

Paramater Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: Mimetype

Description: Mimetype for output resource

Paramater Type STRING

Default Value:

Constraints:
Name: OutputResource
Description: Where the produced resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.2 Import

STRING Import (STRING Path, RESOURCE OutputResource, STRING MimeType)

Version: 1.0
Description: Import an image
Parameter List
Name: Path
Description: Path to the image
Paramater Type STRING
Default Value:
Constraints:
Name: OutputResource
Description: Where the imported resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: MimeType
Description: Test
Paramater Type STRING
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.3 Resize

STRING Resize (RESOURCE InputResource, INT32 Width, INT32 Height, BOOLEAN KeepAspectRatio, RESOURCE OutputResource)

Version: 1.0
Description: Resizes an image
Parameter List
Name: InputResource
Description: The Resource to be resized
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image

Resource Format: jpeg gif png
Ranges:
Name: Width
Description: The new image width
Paramater Type INT32
Default Value:
Constraints:
Name: Height
Description: The new image height
Paramater Type INT32
Default Value:
Constraints:
Name: KeepAspectRatio
Description: Indicates to preserve image aspect ratio or not
Paramater Type BOOLEAN
Default Value:
Constraints:
Name: OutputResource
Description: Where the resized resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.4 Contrast

STRING Contrast (RESOURCE InputResource, INT32 AMOUNT, RESOURCE OutputResource)

Version: 1.0
Description: Change image contrast
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: AMOUNT
Description: The contrast amount
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in

case of error

5.11.3.5 Edge

STRING Edge (RESOURCE InputResource, INT32 ORDER, RESOURCE OutputResource)

Version: 1.0
Description: Edge image (highlight edges in image). The radius is the radius of the pixel neighbourhood.. Specify a radius of zero for automatic radius selection.
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: ORDER
Description: The Order Edge
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.6 Emboss

STRING Emboss (RESOURCE InputResource, INT32 RADIUS, INT32 SIGMA, RESOURCE OutputResource)

Version: 1.0
Description: Emboss image (highlight edges with 3D effect). The radius_ parameter specifies the radius of the Gaussian, in pixels, not counting the center pixel. The sigma_ parameter specifies the standard deviation of the Laplacian, in pixels.
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: RADIUS
Description: The Radius Emboss
Paramater Type INT32
Default Value:

Constraints:
Name: SIGMA
Description: The sigma Emboss
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.7 Blur

STRING Blur (RESOURCE InputResource, INT32 RADIUS, INT32 SIGMA, RESOURCE OutputResource)

Version: 1.0
Description: Blur image. The radius_ parameter specifies the radius of the Gaussian, in pixels, not counting the center pixel. The sigma_ parameter specifies the standard deviation of the Laplacian, in pixels.
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: RADIUS
Description: The Radius Blur
Paramater Type INT32
Default Value:
Constraints:
Name: SIGMA
Description: The sigma Blur
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.8 GaussianBlur

STRING GaussianBlur (RESOURCE InputResource, INT32 RADIUS, INT32 SIGMA, RESOURCE OutputResource)

Version: 1.0
Description: GaussianBlur the image
Parameter List
Name: InputResource
Description: Gaussian blur image. The number of neighbor pixels to be included in the convolution mask is specified by 'width_'. For example, a width of one gives a (standard) 3x3 convolution mask. The standard deviation of the gaussian bell curve is specified by 'sigma'.
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: RADIUS
Description: The Radius GaussianBlur
Paramater Type INT32
Default Value:
Constraints:
Name: SIGMA
Description: The sigma GaussianBlur
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.9 Median

STRING Median (RESOURCE InputResource, INT32 RADIUS, RESOURCE OutputResource)

Version: 1.0
Description: Median the image
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: RADIUS
Description: The Radius Median
Paramater Type INT32
Default Value:

Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.10 Mirror

STRING Mirror (RESOURCE InputResource, BOOLEAN KeepDirection, RESOURCE OutputResource)

Version: 1.0
Description: Mirror the image
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: KeepDirection
Description: The KeepDirection Mirror
Paramater Type BOOLEAN
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.11 Noise

STRING Noise (RESOURCE InputResource, INT32 TYPE, RESOURCE OutputResource)

Version: 1.0
Description: Noise the image
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image

Resource Format: jpeg gif png
Ranges:
Name: TYPE
Description: The Type Noise
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.12 Despeckle

STRING Despeckle (RESOURCE InputResource, RESOURCE OutputResource)

Version: 1.0
Description: Despeckle image (reduce speckle noise)
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.13 Equalize

STRING Equalize (RESOURCE InputResource, RESOURCE OutputResource)

Version: 1.0
Description: Equalize image (histogram equalization)
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image

Resource Format: jpeg gif png
Ranges:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.14 Enhance

STRING Enhance (RESOURCE InputResource, RESOURCE OutputResource)

Version: 1.0
Description: Enhance image (minimize noise)
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.15 ExtractChannel

STRING ExtractChannel (RESOURCE InputResource, INT32 CHANNEL, RESOURCE OutputResource)

Version: 1.0
Description: ExtractChannel the image
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: CHANNEL
Description: The Channel ExtractChannel
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in

case of error

5.11.3.16 Grayscale

STRING Grayscale (RESOURCE InputResource, RESOURCE OutputResource)

Version: 1.0
Description: Grayscale the image
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.17 Magnify

STRING Magnify (RESOURCE InputResource, RESOURCE OutputResource)

Version: 1.0
Description: Magnify image by integral size
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.18 Minify

STRING Minify (RESOURCE InputResource, RESOURCE OutputResource)

Version: 1.0
Description: Reduce image by integral size
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.19 Modulate

STRING Modulate (RESOURCE InputResource, INT32 BRIGHTNESS, INT32 SATURATION, INT32 HUE, RESOURCE OutputResource)

Version: 1.0
Description: Modulate percent hue, saturation, and brightness of an image.Modulation of saturation and brightness is as a ratio of the current value(1.0 for no change). Modulation of hue is an absolute rotation of -180 degrees to +180 degrees from the current position corresponding to an argument range of 0 to 2.0 (1.0 for no change).
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: BRIGHTNESS
Description: Brightness modulate
Paramater Type INT32
Default Value:
Constraints:
Name: SATURATION
Description: Saturation modulate
Paramater Type INT32
Default Value:
Constraints:
Name: HUE

Description: Hue modulate
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.20 Monochrome

STRING Monochrome (RESOURCE InputResource, RESOURCE OutputResource)

Version: 1.0
Description: Monochrome the image
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.21 Negate

STRING Negate (RESOURCE InputResource, BOOLEAN GRAYSCALE, RESOURCE OutputResource)

Version: 1.0
Description: Negate colors in image. Replace every pixel with its complementary color (white becomes black, yellow becomes blue, etc.). Set grayscale to only negate grayscale values in image.
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png

Ranges:
Name: GRAYSCALE
Description: Where the manipulated resource will be stored
Paramater Type BOOLEAN
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.22 Normalize

STRING Normalize (RESOURCE InputResource, RESOURCE OutputResource)

Version: 1.0
Description: Normalize image (increase contrast by normalizing the pixel values to span the full range of color values)
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.23 OilPaint

STRING OilPaint (RESOURCE InputResource, INT32 RADIUS, RESOURCE OutputResource)

Version: 1.0
Description: Oilpaint image (image looks like oil painting)
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image

Resource Format: jpeg gif png
Ranges:
Name: RADIUS
Description: the radius OilPaint
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.24 Quality

STRING Quality (RESOURCE InputResource, INT32 LEVEL, RESOURCE OutputResource)

Version: 1.0
Description: JPEG/MIFF/PNG compression level (default 75).
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: LEVEL
Description: the quality of the compress level
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.25 Quantize

STRING Quantize (RESOURCE InputResource, INT32 NCOLORS, RESOURCE OutputResource)

Version: 1.0
Description: Preferred number of colors in the image. The actual number of colors in the image may be less than your request, but never more. Images with less unique colors than specified with this option will have any duplicate or unused colors removed.

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: NCOLORS
Description: the number of color
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.26 Raise

STRING Raise (RESOURCE InputResource, INT32 WIDTH, INT32 HEIGHT, INT32 XOFFSET, INT32 YOFFSET, BOOLEAN RISED, RESOURCE OutputResource)

Version: 1.0
Description: Raise image (lighten or darken the edges of an image to give a 3-D raised or lowered effect)
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: WIDTH
Description: The width is parts of the geometry specification are measured in pixels
Paramater Type INT32
Default Value:
Constraints:
Name: HEIGHT
Description: The height is parts of the geometry specification are measured in pixels
Paramater Type INT32
Default Value:
Constraints:
Name: XOFFSET
Description: The left edge of the object is to be placed xoffset pixels in from the left edge of the image.
Paramater Type INT32

Default Value:
Constraints:
Name: YOFFSET
Description: The top edge of the object is to be yoffset pixels below the top edge of the image.
Paramater Type INT32
Default Value:
Constraints:
Name: RISED
Description: raisedFlag
Paramater Type BOOLEAN
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.27 ReduceNoise

STRING ReduceNoise (RESOURCE InputResource, INT32 ORDER, RESOURCE OutputResource)

Version: 1.0
Description: Reduce noise in image using a noise peak elimination filter.
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: ORDER
Description: order
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.28 Replace

STRING Replace (RESOURCE InputResource, INT32 R1, INT32 G1, INT32 B1, INT32 R2, INT32

G2, INT32 B2, RESOURCE OutputResource)

Version: 1.0
Description: Replace the image
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: R1
Description: r1
Paramater Type INT32
Default Value:
Constraints:
Name: G1
Description: g1
Paramater Type INT32
Default Value:
Constraints:
Name: B1
Description: b1
Paramater Type INT32
Default Value:
Constraints:
Name: R2
Description: r2
Paramater Type INT32
Default Value:
Constraints:
Name: G2
Description: g2
Paramater Type INT32
Default Value:
Constraints:
Name: B2
Description: b2
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.29 FloodFill

STRING FloodFill (RESOURCE InputResource, INT32 X, INT32 Y, INT32 B, INT32 R, INT32 G, RESOURCE OutputResource)

Version: 1.0

Description: Flood-fill texture across pixels that match the color of the target pixel and are neighbors of the target pixel. Uses current fuzz setting when determining color match.

Parameter List

Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:

Name: X
Description: x
Paramater Type INT32
Default Value:
Constraints:

Name: Y
Description: y
Paramater Type INT32
Default Value:
Constraints:

Name: B
Description: b
Paramater Type INT32
Default Value:
Constraints:

Name: R
Description: r
Paramater Type INT32
Default Value:
Constraints:

Name: G
Description: g
Paramater Type INT32
Default Value:
Constraints:

Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:

Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.30 Roll

STRING Roll (RESOURCE InputResource, INT32 X, INT32 Y, RESOURCE OutputResource)

Version: 1.0

Description: Roll image (rolls image vertically and horizontally) by specified number of columnms and rows)

Parameter List

Name: InputResource

Description: The Resource to be manipulated

Paramater Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: X

Description: x

Paramater Type INT32

Default Value:

Constraints:

Name: Y

Description: y

Paramater Type INT32

Default Value:

Constraints:

Name: OutputResource

Description: Where the manipulated resource will be stored

Paramater Type RESOURCE

Default Value:

Constraints:

Result: Result

Result type: STRING

Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.31 Rotate

STRING Rotate (RESOURCE InputResource, INT32 ANGLE, RESOURCE OutputResource)

Version: 1.0

Description: Rotate image counter-clockwise by specified number of degrees.

Parameter List

Name: InputResource

Description: The Resource to be manipulated

Paramater Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: ANGLE

Description: Number of the degrees

Paramater Type INT32

Default Value:

Constraints:

Name: OutputResource

Description: Where the manipulated resource will be stored

Paramater Type RESOURCE

Default Value:

Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.32 Scale

STRING Scale (RESOURCE InputResource, INT32 WIDTH, INT32 HEIGHT, INT32 MODE, RESOURCE OutputResource)

Version: 1.0
Description: Resize image by using simple ratio algorithm
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: WIDTH
Description: Width
Paramater Type INT32
Default Value:
Constraints:
Name: HEIGHT
Description: Height
Paramater Type INT32
Default Value:
Constraints:
Name: MODE
Description: Mode
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.33 Shear

STRING Shear (RESOURCE InputResource, INT32 XSHEAR, INT32 Yshear, RESOURCE OutputResource)

Version: 1.0
Description: Shear image (create parallelogram by sliding image by X or Y axis). Shearing slides one edge of an image along the X or Y axis, creating a parallelogram. An X direction shear slides an edge along

the X axis, while a Y direction shear slides an edge along the Y axis. The amount of the shear is controlled by a shear angle. For X direction shears, x degrees is measured relative to the Y axis, and similarly, for Y direction shears y degrees is measured relative to the X axis. Empty triangles left over from shearing the image are filled with the color defined as borderColor.

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: XSHEAR
Description: XSHEAR
Paramater Type INT32
Default Value:
Constraints:
Name: Yshear
Description: Yshear
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.34 Shade

STRING Shade (RESOURCE InputResource, INT32 AZIMUTH, INT32 ELEVATION, BOOLEAN COLOR, RESOURCE OutputResource)

Version: 1.0
Description: Shade image using distant light source. Specify azimuth_ and elevation_ as the position of the light source. By default, the shading results as a grayscale image.. Set colorShading_ to true to shade the red, green, and blue components of the image.

Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: AZIMUTH
Description: AZIMUTH
Paramater Type INT32
Default Value:

Constraints:
Name: ELEVATION
Description: ELEVATION
Paramater Type INT32
Default Value:
Constraints:
Name: COLOR
Description: COLOR
Paramater Type BOOLEAN
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.35 Spread

STRING Spread (RESOURCE InputResource, INT32 AMOUNT, RESOURCE OutputResource)

Version: 1.0
Description: Spread pixels randomly within image by specified amount.
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: AMOUNT
Description: AMOUNT
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.36 SetOpacity

STRING SetOpacity (RESOURCE InputResource, INT32 LEVEL, RESOURCE OutputResource)

Version: 1.0
Description: Set the opacity of the image.
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: LEVEL
Description: LEVEL
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.37 SubImage

STRING SubImage (RESOURCE InputResource, INT32 X, INT32 Y, INT32 WIDTH, INT32 HEIGHT, RESOURCE OutputResource)

Version: 1.0
Description: SubImage image.
Parameter List
Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: X
Description: x coordinate of the top-level corner of the rectangle
Paramater Type INT32
Default Value:
Constraints:
Name: Y
Description: y coordinate of the top-level corner of the rectangle
Paramater Type INT32
Default Value:
Constraints:
Name: WIDTH
Description: Width member
Paramater Type INT32

Default Value:
Constraints:
Name: HEIGHT
Description: Height member
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.38 GetInfo

STRING GetInfo (RESOURCE InputResource, INT32 WIDTH, INT32 HEIGHT)

Version: 1.0
Description: Return the size of the image.
Parameter List
Name: InputResource
Description: The Resource under analisys
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: WIDTH
Description: The width of the Image
Paramater Type INT32
Default Value:
Constraints:
Name: HEIGHT
Description: The height of the Image
Paramater Type INT32
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.39 SetMaskColour

STRING SetMaskColour (RESOURCE InputResource, INT32 R, INT32 G, INT32 B, RESOURCE OutputResource)

Version: 1.0
Description: Set the color
Parameter List

Name: InputResource
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: R
Description: Red
Paramater Type INT32
Default Value:
Constraints:
Name: G
Description: Green
Paramater Type INT32
Default Value:
Constraints:
Name: B
Description: Blue
Paramater Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Paramater Type RESOURCE
Default Value:
Constraints:
Result: Result
Result type: STRING
Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.40 Paste

STRING Paste (RESOURCE InputResource1, RESOURCE InputResource2, INT32 X, INT32 Y, INT32 COMPOSE, RESOURCE OutputResource)

Version: 1.0
Description: Paste image
Parameter List
Name: InputResource1
Description: The Resource to be manipulated
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: InputResource2
Description: The Resource paste
Paramater Type RESOURCE
Default Value:
Constraints:
Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: X

Description: X

Paramater Type INT32

Default Value:

Constraints:

Name: Y

Description: Y

Paramater Type INT32

Default Value:

Constraints:

Name: COMPOSE

Description: Compose

Paramater Type INT32

Default Value:

Constraints:

Name: OutputResource

Description: Where the manipulated resource will be stored

Paramater Type RESOURCE

Default Value:

Constraints:

Result: Result

Result type: STRING

Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error

5.11.3.41 Test

RESOURCE Test (RESOURCE InputResource, AXOM Axom)

Version: 1.0

Description: Test an image

Parameter List

Name: InputResource

Description: The Resource to be tested

Paramater Type RESOURCE

Default Value:

Constraints:

Name: Axom

Description: The object

Paramater Type AXOM

Default Value:

Constraints:

Result: Result

Result type: RESOURCE

Result Description: The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error