



Automating Production of Cross Media Content for Multi-channel Distribution

www.AXMEDIS.org

DE5.0.1.1 AXMEDIS Major Tools User Manuals

Version: 1.2

Date: 12/09/2006

Responsible: FUPF (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: DE5.0.1.1

Contractual Date of Delivery: M22

Actual Date of Delivery: 12/09/2006

Title of Deliverable: AXMEDIS Major Tools User Manuals

Work-Package contributing to the Deliverable: WP5

Task contributing to the Deliverable: WP5

Nature of the Deliverable: report

Author(s): FUPF, and almost all partners

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

Keyword List: Authoring, content editing, content processing, players, GRID, content production, fingerprint estimation, content descriptors, metadata editor, metadata mapper, programme and publication, scheduler, workflow, DRM editor, license editor, plug ins, adaptation, etc.

Table of Content

1	INTRODUCTION	8
2	AXMEDIS EDITOR (DSI).....	9
2.1	MAIN FUNCTIONALITIES	9
2.2	RELATIONSHIP WITH OTHER TOOLS.....	9
2.3	DETAILED DESCRIPTION OF THE FUNCTIONALITIES AND SCREENSHOTS	9
2.3.1	Create a new AXMEDIS Object	9
2.3.2	The Activities Window.....	11
2.3.3	Adding object reference	13
2.3.4	Adding embedded object	15
2.3.5	Adding Dublin Core metadata	17
2.3.6	Open an existing object.....	19
2.3.7	Extract an embedded resource	20
2.3.8	Modifying an AXMEDIS Object stored on Database.....	20
2.3.9	The Query dialog	21
2.3.10	The resource property dialog.....	23
2.3.11	The Content processing plugins	24
2.4	THE AXMEDIS EDITOR MENUS	26
2.4.1	The File menu	26
2.4.2	Edit menu	26
2.4.3	The Resource View menu.....	27
2.4.4	The Editors/Viewers menu	27
2.5	AXMEDIS RESOURCE VIEWER (DSI).....	29
2.5.1	Main functionalities	29
2.5.2	Relationship with other tools	29
2.5.3	Image Viewer - Detailed description of the functionalities and Screenshots.....	29
2.5.4	Audio Player - Detailed description of the functionalities and Screenshots	30
2.5.5	Video Player - Detailed description of the functionalities and Screenshots.....	31
2.5.6	Document Viewer - Detailed description of the functionalities and Screenshots	31
2.5.7	SMIL Player - Detailed description of the functionalities and Screenshots	31
2.5.8	MPEG4 Player - Detailed description of the functionalities and Screenshots	32
2.6	AXMEDIS METADATA EDITOR (UNIVLEEDS).....	33
2.6.1	Main functionalities	33
2.6.2	Relationship with other tools	33
2.6.3	Detailed description of the functionalities and Screenshots	33
2.6.3.1	Adding Metadata Elements	33
2.6.3.2	Inserting Metadata Elements.....	34
2.6.3.3	Deleting Metadata Elements	35
2.6.3.4	Editing Metadata Elements	35
2.7	AXMEDIS METADATA MAPPER EDITOR (UNIVLEEDS)	37
2.7.1	Main functionalities	37
2.7.2	Relationship with other tools	37
2.7.3	Detailed description of the functionalities and Screenshots	37
2.7.3.1	Creating a Transformation XSLT to map Metadata	37
2.7.3.2	Creating a Transformation XSLT to map Metadata	37
2.7.3.3	Transforming Metadata using the XSLT	39
2.8	AXMEDIS VISUAL AND BEHAVIOUR EDITOR (EPFL)	41
2.8.1	Main functionalities	41
2.8.2	Relationship with other tools	41
2.8.3	Detailed description of the functionalities and Screenshots	41
2.8.3.1	Editing the SMIL components with Hierarchy Editor and Viewer.....	41
2.8.3.2	Editing the visual scenes with Visual Editor and Viewer	41

2.8.3.3	Editing temporal behaviour of media resources with Behaviour Editor and Viewer	42
2.8.3.4	Previewing the result with SMIL player	43
2.9	AXMEDIS DRM EDITOR (FUPF)	44
2.9.1	Main functionalities	44
2.9.2	Relationship with other tools	44
2.9.3	Detailed description of the functionalities and Screenshots	44
2.10	AXMEDIS WORKFLOW EDITOR (IRC)	48
2.10.1	Main functionalities	48
2.10.2	Relationship with other tools	48
2.10.3	Detailed description of the functionalities and Screenshots	48
2.11	AXMEDIS PROTECTION INFORMATION EDITOR (FHGIGD)	49
2.11.1	Main functionalities	49
2.11.2	Relationship with other tools	49
2.11.3	Detailed description of the functionalities and Screenshots	49
2.11.3.1	Viewing of Protection Information	50
2.11.3.2	Editing of Protection Information	51
3	AXMEDIS PC PLAYER (DSI)	52
3.1	MAIN FUNCTIONALITIES	52
3.2	RELATIONSHIP WITH OTHER TOOLS	52
3.3	DETAILED DESCRIPTION OF THE FUNCTIONALITIES AND SCREENSHOTS	52
4	AXMEDIS PDA PLAYER (TISCALI, DSI)	55
4.1	MAIN FUNCTIONALITIES	55
4.2	RELATIONSHIP WITH OTHER TOOLS	55
4.3	DETAILED DESCRIPTION OF THE FUNCTIONALITIES AND SCREENSHOTS	55
5	AXMEDIS ACTIVE X CONTROL (DSI)	56
5.1	MAIN FUNCTIONALITIES	56
5.2	RELATIONSHIP WITH OTHER TOOLS	56
5.3	DETAILED DESCRIPTION OF THE FUNCTIONALITIES AND SCREENSHOTS	56
5.3.1	AXMEDISViewerCtrl	58
5.3.1.1	Basic Interface	58
5.3.1.2	Visual Control Interface	60
5.3.1.3	Time Control Interface	61
5.3.1.4	Generic Command Interface	62
5.3.1.5	Examples of usage	63
6	AXMEDIS DRM EDITOR (FUPF)	65
6.1	MAIN FUNCTIONALITIES	65
6.2	RELATIONSHIP WITH OTHER TOOLS	65
6.3	DETAILED DESCRIPTION OF THE FUNCTIONALITIES AND SCREENSHOTS	65
7	AXMEDIS DRM VIEWER (FUPF)	69
7.1	MAIN FUNCTIONALITIES	69
7.2	RELATIONSHIP WITH OTHER TOOLS	69
7.3	DETAILED DESCRIPTION OF THE FUNCTIONALITIES AND SCREENSHOTS	69
8	AXMEDIS RULE EDITOR (DSI)	70
8.1	MAIN FUNCTIONALITIES	70
8.1.1	The Menu Bar	71
8.1.2	ToolBar Area	73
8.1.3	Workspace Area	73
8.1.4	Output area	75
8.1.5	MS Windows Firewall Alert at startup	75
8.1.6	Configuration	76
8.1.7	Tools, Viewers and Editors	77
8.1.8	Selection Editor ToolBar	78
8.2	RELATIONSHIP WITH OTHER TOOLS	80
8.3	DETAILED DESCRIPTION OF THE FUNCTIONALITIES AND SCREENSHOTS	80
8.3.1	Editing on AXCP Rule	80

8.3.2	Creating a new rule	80
8.3.3	Loading an existing rule	81
8.3.4	Editing a rule	82
8.3.5	Debugging Rule functionalities	83
8.3.6	Activating a rule	85
8.3.7	Creating a selection Parameter	85
8.3.8	Editing a selection Parameter	88
8.3.9	Testing a query	88
8.3.10	Load and Save a selection from/to the Selection Archive	88
8.3.11	Load and Save a selection from/to the File System	89
8.3.12	Conclusion	89
8.4	HOW TO CREATE A RULE, A SMALL TUTORIAL	89
8.4.1	EXAMPLE 1 – Rule without parameters (Simple rule)	89
8.4.2	EXAMPLE 2 – Rule with Parameters	93
8.4.3	EXAMPLE 3 – Rule for creating AXMEDIS Object	94
9	AXMEDIS RULE ENGINE - RULE SCHEDULER (DSI)	96
9.1	MAIN FUNCTIONALITIES	96
9.2	RELATIONSHIP WITH OTHER TOOLS	96
9.3	DETAILED DESCRIPTION OF THE FUNCTIONALITIES AND SCREENSHOTS	96
9.3.1	Menu bar	96
9.3.2	Rules/Jobs Table	98
9.3.3	Remote Executors Table	99
9.3.4	Auxiliary dialogs	99
9.3.5	Rule Properties Dialog	99
9.3.6	Executor Profile Dialog	99
9.3.7	Logs Dialog	99
9.3.8	Suspend Rule Dialog	100
9.3.9	Preferences Dialog	100
9.3.10	Configuration Parameters	101
9.3.11	Activating and stopping a rule	102
9.3.12	Conclusion	102
10	AXMEDIS RULE ENGINE - RULE EXECUTOR (DSI)	103
10.1	MAIN FUNCTIONALITIES	103
10.2	RELATIONSHIP WITH OTHER TOOLS	103
10.3	DETAILED DESCRIPTION OF THE FUNCTIONALITIES AND SCREENSHOTS	103
10.4	HOW TO USE THE GRID AND RULE EXECUTOR STAND ALONE, A SMALL TUTORIAL	104
10.4.1	Rule execution in the AXCP GRID environment	104
10.4.2	Rule execution by means the stand alone version of the Rule Executor	107
11	AXMEDIS PROGRAMME & PUBLICATION EDITOR (UNIVLEEDS)	108
11.1	MAIN FUNCTIONALITIES	108
11.2	RELATIONSHIP WITH OTHER TOOLS	108
11.3	DETAILED DESCRIPTION OF THE FUNCTIONALITIES AND SCREENSHOTS	110
11.3.1	Creating a P&P Programme	111
11.3.2	Loading an Existing P&P Programme	111
11.3.3	Editing a P&P Programme	112
11.3.4	Querying for AXMEDIS Objects in the P&P Editor	113
11.3.5	Testing a complete P&P Programme	114
11.3.6	Activating and Stopping a P&P Programme	115
11.3.7	Configuring the P&P Editor	115
12	AXMEDIS PROGRAMME & PUBLICATION ENGINE (UNIVLEEDS)	117
12.1	MAIN FUNCTIONALITIES	117
12.2	RELATIONSHIP WITH OTHER TOOLS	117
12.3	DETAILED DESCRIPTION OF THE FUNCTIONALITIES AND SCREENSHOTS	118
13	AXMEDIS PROGRAMME & PUBLICATION ENGINE MONITOR (UNIVLEEDS)	119
13.1	MAIN FUNCTIONALITIES	119

13.2	RELATIONSHIP WITH OTHER TOOLS.....	119
13.3	DETAILED DESCRIPTION OF THE FUNCTIONALITIES AND SCREENSHOTS	119
13.3.1	Running the Monitor	119
13.3.2	Status Information	119
13.3.3	Programme Managements.....	119
14	AXMEDIS WORKFLOW AND WORKFLOW MANAGER - OPENFLOW (IRC).....	120
14.1	MAIN FUNCTIONALITIES.....	120
14.2	RELATIONSHIP WITH OTHER TOOLS.....	120
14.3	DETAILED DESCRIPTION OF THE FUNCTIONALITIES AND SCREENSHOTS	120
15	AXMEDIS WORKFLOW AND WORKFLOW MANAGER – MICROSOFT BIZTALK (IRC)	129
15.1	MAIN FUNCTIONALITIES.....	129
15.2	RELATIONSHIP WITH OTHER TOOLS.....	129
15.3	DETAILED DESCRIPTION OF THE FUNCTIONALITIES AND SCREENSHOTS	129
16	AXMEDIS PLUGINS.....	130
16.1	AUDIO ADAPTATION PLUGIN (EPFL)	130
16.1.1	Main functionalities.....	130
16.1.2	Relationship with other tools.....	130
16.1.3	Detailed description of the functionalities and Screenshots.....	130
16.1.3.1	FFMPEG.....	130
16.1.3.2	libsndfile	134
16.1.3.3	Tricks/Errors.....	139
16.2	AUDIO DESCRIPTOR PLUGIN (EPFL)	140
16.2.1	Main functionalities.....	140
16.2.2	Relationship with other tools.....	140
16.2.3	Detailed description of the functionalities and Screenshots.....	140
16.2.3.1	Low Level Descriptors Extraction.....	141
16.2.3.2	Segmentation into Silence / Speech / Noise / Music	144
16.2.3.3	Music Genre recognition.....	147
16.2.3.4	Tempo detection	151
16.3	MULTIMEDIA ADAPTATION PLUGIN (EPFL)	154
16.3.1	Main functionalities.....	154
16.3.2	Relationship with other tools.....	154
16.3.3	Detailed description of the functionalities and Screenshots.....	154
16.3.3.1	EXTRACT MEDIA TRACK.....	156
16.3.3.2	MP4 TO 3GP.....	157
16.3.3.3	CAT MULTIMEDIA FILES	158
16.3.3.4	MP4 TO ISMA.....	159
16.3.3.5	ADD MULTIMEDIA FILES.....	160
16.3.3.6	TO MP4.....	161
16.3.3.7	DELAY TRACK.....	161
16.3.3.8	REMOVE TRACK	162
16.3.3.9	EXTRACT FROM START TO END	163
16.3.3.10	MP4 TO AVI.....	164
16.4	DOCUMENT CRYPTLIB PLUGIN (EPFL).....	165
16.4.1	Main functionalities.....	165
16.4.2	Relationship with other tools.....	165
16.4.3	Detailed description of the functionalities and Screenshots.....	165
16.5	PLAGIARISM DETECTION PLUGIN (DIPITA).....	170
16.5.1	Main functionalities.....	170
16.5.2	Relationship with other tools.....	170
16.5.3	Detailed description of the functionalities and Screenshots.....	170
16.6	DOCUMENT DESCRIPTOR EXTRACTOR PLUGIN (DIPITA)	173
16.6.1	Main functionalities.....	173
16.6.2	Relationship with other tools.....	173
16.6.3	Detailed description of the functionalities and Screenshots.....	173
16.7	DOCUMENT ADAPTATION PLUGIN (DIPITA)	176
16.7.1	Main functionalities.....	176
16.7.2	Relationship with other tools.....	176
16.7.3	Detailed description of the functionalities and Screenshots.....	176

16.8	AUDIO FP PLUGIN (FHGIGD)	179
16.8.1	Main functionalities	179
16.8.2	Relationship with other tools	179
16.8.3	Detailed description of the functionalities and Screenshots	179
16.9	M2ANY - AUDIO FP PLUGIN (FHGIGD)	185
16.9.1	Main functionalities	185
16.9.2	Relationship with other tools	185
16.9.3	Detailed description of the functionalities and Screenshots	185
16.10	VIDEO FP PLUGIN (FHGIGD)	190
16.10.1	Main functionalities	190
16.10.2	Relationship with other tools	190
16.10.3	Detailed description of the functionalities and Screenshots	190
16.11	GENERIC RESOURCE FILES FP PLUGIN (FHGIGD)	193
16.11.1	Main functionalities	193
16.11.2	Relationship with other tools	193
16.11.3	Detailed description of the functionalities and Screenshots	193
16.12	VIDEO DESCRIPTOR PLUGIN (FHGIGD)	199
16.12.1	Main functionalities	199
16.12.2	Relationship with other tools	199
16.12.3	Detailed description of the functionalities and Screenshots	199
16.13	VIDEO ADAPTATION PLUGIN (FHGIGD)	202
16.13.1	Main functionalities	202
16.13.2	Relationship with other tools	202
16.13.3	Detailed description of the functionalities and Screenshots	202
16.14	WORKFLOW EDITOR PLUGIN (IRC)	203
16.14.1	Main functionalities	203
16.14.2	Relationship with other tools	203
16.14.3	Detailed description of the functionalities and Screenshots	203
16.15	WORKFLOW RULE EDITOR PLUGIN (IRC)	206
16.15.1	Main functionalities	206
16.15.2	Relationship with other tools	206
16.15.3	Detailed description of the functionalities and Screenshots	206
16.16	WORKFLOW ENGINE PLUGIN (IRC)	210
16.16.1	Main functionalities	210
16.16.2	Relationship with other tools	211
16.16.3	Detailed description of the functionalities and Screenshots	211
16.17	RINGTONE ADAPTATION PLUGIN (IRC)	212
16.17.1	Main functionalities	212
16.17.2	Relationship with other tools	212
16.17.3	Detailed description of the functionalities and Screenshots	212
16.17.3.1	Convert Function	212
16.17.3.2	Resample Function	215
16.17.3.3	getInfo Function	217
16.17.3.4	Clip Function	218
16.18	IMAGE PROCESSING PLUGIN (DSI)	220
16.18.1	Main functionalities	220
16.18.2	Relationship with other tools	221
16.18.3	Detailed description of the functionalities and Screenshots	221
16.18.3.1	Conversion	222
16.18.3.2	Import	222
16.18.3.3	Resize	223
16.18.3.4	Contrast	224
16.18.3.5	Edge	224
16.18.3.6	Emboss	225
16.18.3.7	Blur	226
16.18.3.8	GaussianBlur	226
16.18.3.9	Median	227
16.18.3.10	Mirror	228
16.18.3.11	Noise	228
16.18.3.12	Despeckle	229
16.18.3.13	Equalize	229
16.18.3.14	Enhance	231

16.18.3.15	ExtractChannel	231
16.18.3.16	Grayscale	232
16.18.3.17	Magnify	232
16.18.3.18	Minify	233
16.18.3.19	Modulate	233
16.18.3.20	Monochrome	234
16.18.3.21	Negate	234
16.18.3.22	Normalize	235
16.18.3.23	OilPaint	235
16.18.3.24	Quality	236
16.18.3.25	Quantize	237
16.18.3.26	Raise	237
16.18.3.27	ReduceNoise	238
16.18.3.28	Replace	239
16.18.3.29	FloodFill	240
16.18.3.30	Roll	241
16.18.3.31	Rotate	241
16.18.3.32	Scale	242
16.18.3.33	Shear	243
16.18.3.34	Shade	243
16.18.3.35	Spread	244
16.18.3.36	SetOpacity	245
16.18.3.37	SubImage	245
16.18.3.38	GetInfo	246
16.18.3.39	SetMaskColour	247
16.18.3.40	Paste	248
16.18.3.41	Test	248

1 Introduction

This deliverable aim is to describe the User Manuals of AXMEDIS tools. These tools are the following:

- AXMEDIS Editor
- AXMEDIS Player
- AXMEDIS DRM Editor
- AXMEDIS DRM Viewer
- AXMEDIS ContProc Rule Editor
- AXMEDIS Rule Scheduler
- AXMEDIS Prog&Pub Editor
- AXMEDIS Prog&Pub Engine
- AXMEDIS Prog&Pub Monitor
- AXMEDIS Workflow and Workflow Manager
- AXMEDIS Plugins

In the rest of this deliverable, the User Manual of these tools are described.

2 AXMEDIS Editor (DSI)

2.1 Main functionalities

The AXMEDIS Editor is used to create AXMEDIS Objects embedding "raw" digital resources or other AXMEDIS Objects.

The AXMEDIS Editor allows to:

- create a new AXMEDIS Object
- add resources (images, videos, documents, etc.)
- add AXMEDIS Objects from file or from database
- manipulate the AXMEDIS Object structure (remove any element, move the elements)
- save to file
- upload to database
- load an AXMEDIS Object from file
- load an AXMEDIS Object from database
- view the resources with the internal viewers
- manipulate the resources using the Content Processing Plugins
- associate metadata to the AXMEDIS Object (using the Metadata Editor),
- define the Potential Available Rights for the Object (using the internal DRM Editor)
- create a SMIL presentation for the resources inside the Object (using the Visual Editor and Behaviour Editor)
- define the protection information for the object (using the Protection Editor)
- see the status of the object in the Workflow process (using the Workflow Viewer)
- be launched from the workflow to do a specific job and when the user finishes it can notify the activity completion and thus to proceed to the next step in the production process.

2.2 Relationship with other tools

The AXMEDIS Editor:

- uses the AXMEDIS Database to search/retrieve for content and to store content
- can be launched from the Workflow engine and interact with it
- uses the content processing plugins to manipulate the digital resource


2.3 Detailed description of the functionalities and Screenshots

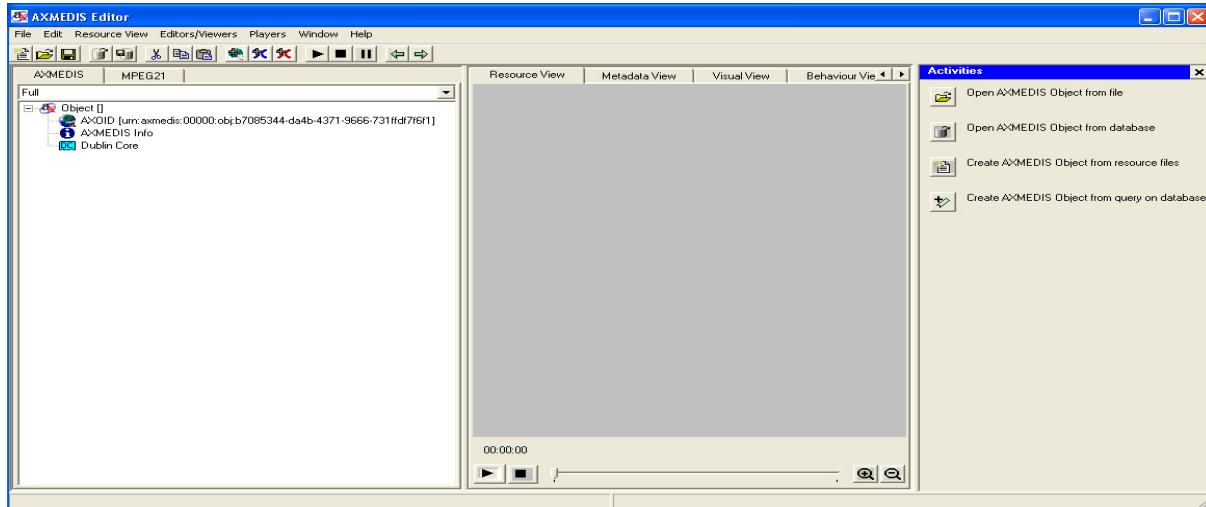
This section aims to describe the use of the AXMEDIS Editor to create AXMEDIS objects embedding "raw" digital resource or other AXMEDIS objects.

Initially section 2.3.1 shows a short tutorial to create a new AXMEDIS object.


Subsequently the next sections will show all the features available in the Editor more in detail. The menus and the integration of the AXMEDIS Editor with a database will also be explained.

2.3.1 Create a new AXMEDIS Object

To create a new AXMEDIS object select **File/New** from the Menu or use the  button on the toolbar




The AXMEDIS Editor will appear divided in three main parts:

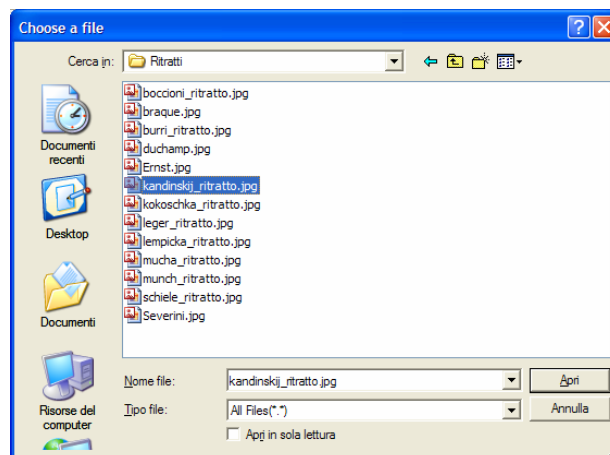
- the Tree view window on the left, that provides a view of the AXMEDIS object structure; selecting the MPEG-21 panel it is possible to see the hierarchical view of the MPEG-21 structure of the object;
- the Panels View window in the middle, which shows the selected panel;
- the Activities windows on the right, that helps to find easily the main features of the Editor. If necessary this windows can be closed by clicking on the  button in the high-right corner to enlarge the Resource View window

In the Tree window initially are present only three items:

- the **AXOID**, that is the unique AXMEDIS Object Identification number
- the **AXMEDIS Info** item, double clicking on it will show information about the object creator, the creation date, etc.
- the **Dublin Core** item, containing the metadata information in Dublin Core standard format

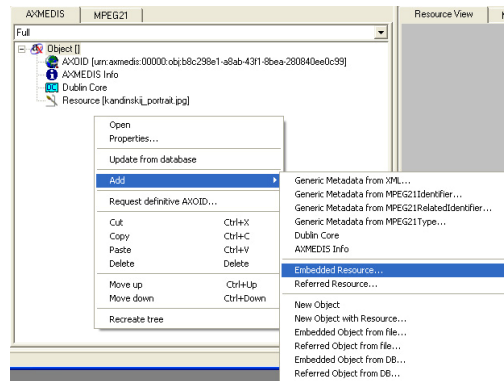
The first thing to do to create a new object is to add a number of digital resources initially stored in the hard disk as single files.

To add a resource from a file on the local hard disk select the  button from the toolbar and select the file to be added:



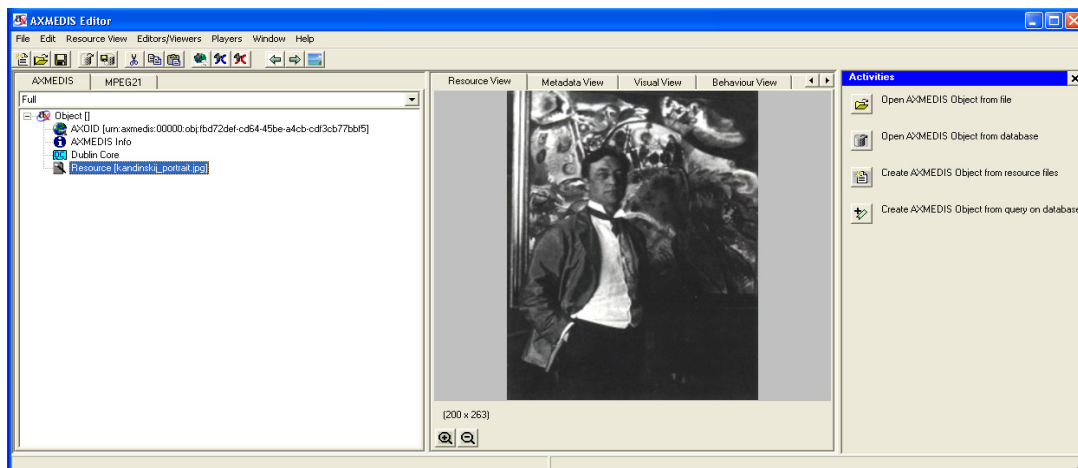
It is possible to add a resource also using the drag-and-drop feature, dragging the resource in the tree window. The AXMEDIS Editor is capable to automatically recognize all the most important file formats for documents (txt, doc, pdf, rtf, etc.), images (gif, tiff, png, jpg, etc.), audio (wma, mp3, wav, etc.), video (mpeg, mpg, etc.) and to play them using the internal player.

Finally it is possible to add a resource using the contextual menu with a right click on the tree window and selecting **Add/Embedded Resource...**



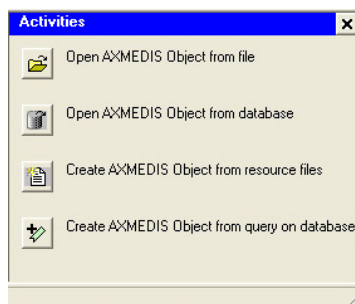
When a resource is correctly embedded, it appears in the tree window with an icon identifying the type of the resource followed by the prefix *Resource* and the resource name closed by square brackets.

Double clicking on the resource opens the ImageViewer:

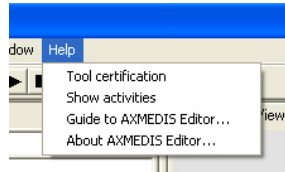


2.3.2 The Activities Window

The Activities window permits a simply access to basic features using intuitive shortcuts.



If not visible, it is possible to open the Activities window by clicking in the **Help/Show activities** menu.




The available features in the Activities window are the following:

- **Open AXMEDIS Object from file**
- **Open AXMEDIS Object from database**
- **Create AXMEDIS Object from resource files**
- **Create AXMEDIS Object from query on database**

Clicking on  **Open AXMEDIS Object from file** simply opens a window to select and load an AXMEDIS object from the disk.


Clicking on  **Open AXMEDIS Object from database** the Query dialog is open. See section 2.3.9 for more details about the Query dialog.

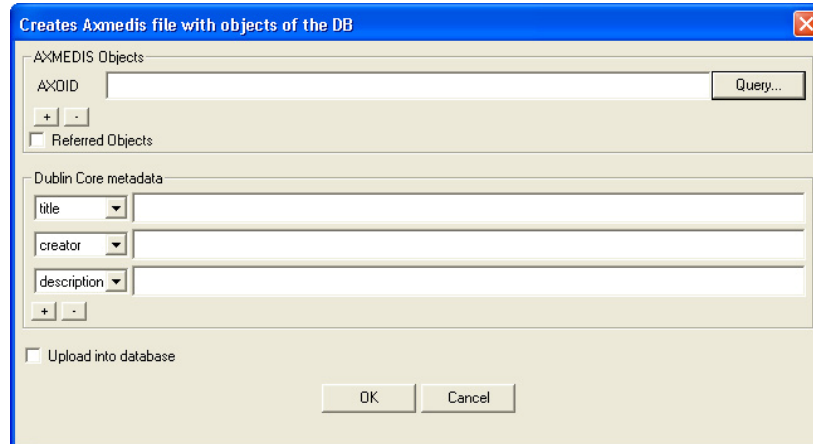
Clicking on  **Create Object from resource files** opens a new dialog to help a user to create a new AXMEDIS object from scratch.

The **Resources** box permits to add external resource selecting them from the disk. Pressing the **+** button a new resource is added, otherwise pressing the **–** button the resource is deleted.

The **Dublin Core metadata** box permits to add metadata of the object before his creation as AXMEDIS object.

Ticking the **Upload into database** option the object is automatically uploaded into the database. Pressing OK the new object is created and, eventually, uploaded into the database.

Clicking  **Create AXMEDIS Object from query on database** opens a new dialog to help a user to create a new AXMEDIS object from scratch using existing objects stored in the database.




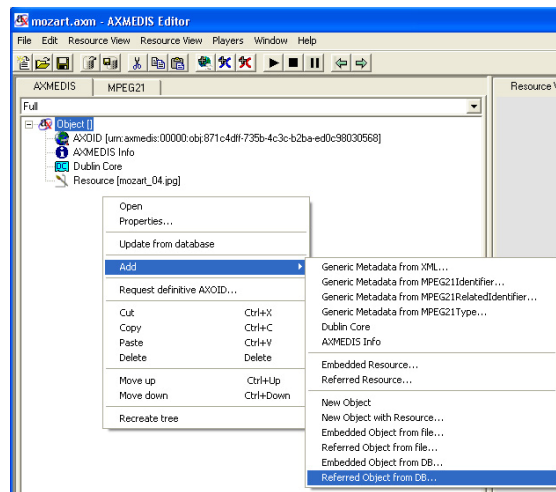
The functioning is very similar to the previous.

The only difference is in the first box now named **AXMEDIS Objects** that permits to add an object querying it from the database. Pressing the **Query** button the Query dialog is open. See section 2.3.9 for more details about the Query dialog.

2.3.3 Adding object reference

Using this feature it is possible to add into an AXMEDIS object a reference to another object stored into the database.

This feature is accessible in the toolbar pressing the  button or **Edit/Add/Referred Object from DB** in the menu selecting.



Selecting this option, a query dialog is open. The dialog permits to execute a search into the database using a number of pre-defined fields. If you need more information about the Query dialog see section 2.3.9

It is possible to search an object using the following parameters:

- **Object creator**
- **Title**
- **Coverage**
- **Format**
- **Type**
- **Subject**
- **Description**
- **Creation Date**
- **AXInfo information** (such as Status, Distributor, Owner and Access Mode)

For example, to search into the database an AXMEDIS object containing a comedy film, it is necessary to write this parameter into the Type field, select into the Info Results box the information that will appear in the results (for example the creator and the title), and press the Submit button.

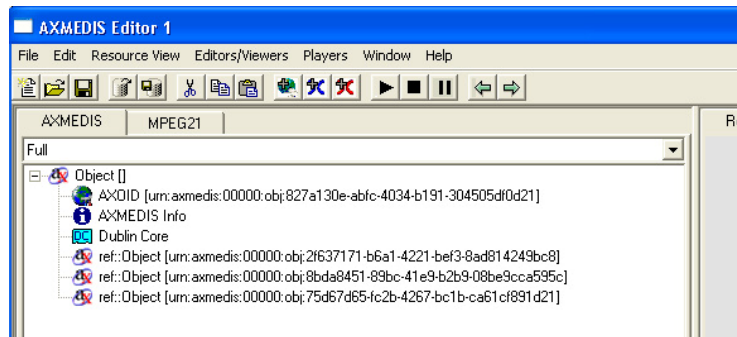
A new dialog will appear where it is necessary to insert the User Name and the password to access the database.

Pressing OK the query is sent to the database and the results are shown in a new window

Creator	Title	Object Id
Nigel Cole	L'erba di Grace	urn:axmedis:00000:obj:005978ae-4dee-4afc-ad34-2595d118a3c6
Buster Keaton	La palla numero 13	urn:axmedis:00000:obj:12a9b8b8-1d31-4ce7-bc27-4b3f7de6f3bf
William Dieterle	Il tesoro del santo	urn:axmedis:00000:obj:2f637171-b6a1-4221-bef3-8ad814249bc8
Nigel Cole	L'erba di Grace	urn:axmedis:00000:obj:429cf2c-e563-40e8-adc1-c07c0ba5c2ba
H.C. Potter	Follie di Jazz (Second Chorus)	urn:axmedis:00000:obj:4e5a3346-ed4b-4706-b000-49a991aa829d
Willis Goldbeck	Love Laughs at Andy Hardy	urn:axmedis:00000:obj:6d891556-b678-42c6-8c3f-7eabe1b2814a
Buster Keaton	Il Navigatore	urn:axmedis:00000:obj:75d67d65-fc2b-4267-bc1b-ca61df891d21
NULL	The perils of Pauline	urn:axmedis:00000:obj:7efbdc5-f28b-4239-9391-9bdf27e45490
Preston Sturges	The sin of Harold Diddlebock	urn:axmedis:00000:obj:85fbc66f-d380-4646-b4be-4afc1309e8
NULL	La mia brUNETTA preferita	urn:axmedis:00000:obj:8bda8451-89bc-41e9-b2b9-08be9cca595c
Jean Becker	I ragazzi del Marais	urn:axmedis:00000:obj:8e44625d-c88c-4f95-baa5-070467b8c26f
NULL	L'uomo in più	urn:axmedis:00000:obj:97f93cb0-d584-4b17-b8ed-9329b47b678a
Andrew V. McLaglen	McLintock!	urn:axmedis:00000:obj:9afb3beb-0f1e-447c-bbf4-3109883a3a35
Edward Sedgwick	Il cameraman	urn:axmedis:00000:obj:a08e51cc-2ad0-42f0-bf74-55d9a157c594
Henry Koster	The Inspector General	urn:axmedis:00000:obj:d23f2ea8-1643-4711-88d6-a54a69a360ce


At this point it is possible to select the object in the list and press the OK button to continue. It is also possible to select more than only one object pressing the <SHIFT> or the <CTRL> keys and selecting all the needed objects.

After pressing OK, the selected objects are added in the Editor as reference objects (identified by the **ref** prefix).



2.3.4 Adding embedded object

Using this feature it is possible to add into an AXMEDIS object another object stored into the database.

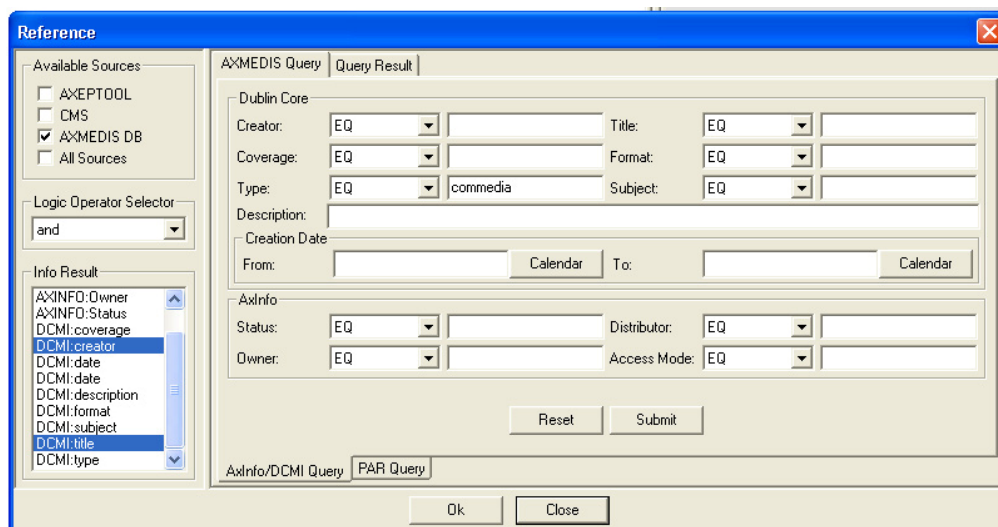
This feature is accessible in the toolbar pressing the  button or in the menu selecting **Edit/Add/Embedded Object from DB**. The procedure is very similar to what explained in the previous section.

Selecting this option, a query dialog is open. The dialog permits to execute a search into the database using a number of pre-defined fields. If you need more information about the Query dialog see section 2.3.9

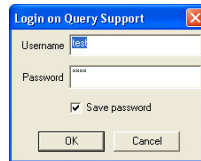
It is possible to search an object using the following parameters:

- **Object creator**
- **Title**
- **Coverage**
- **Format**
- **Type**
- **Subject**
- **Description**
- **Creation Date**
- **AXInfo information** (such as Status, Distributor, Owner and Access Mode)

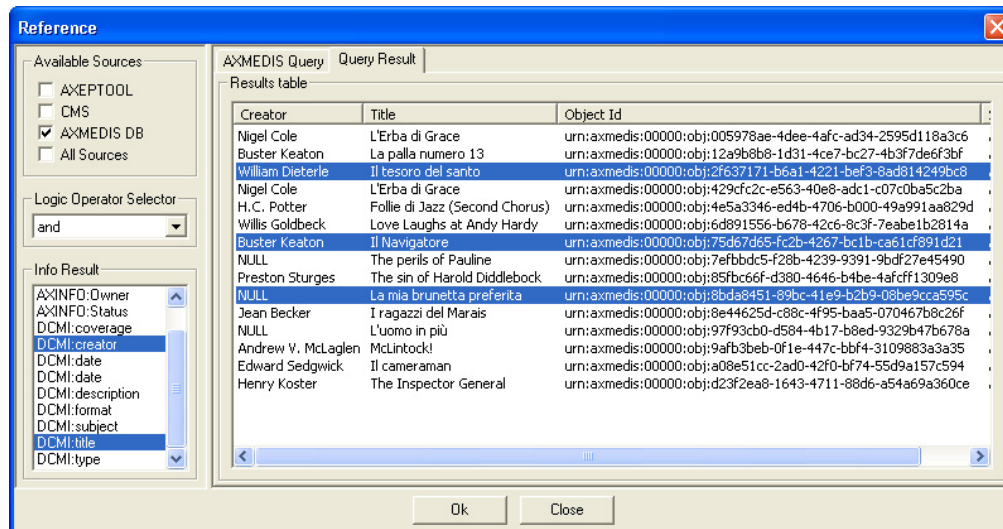
For example, to search into the database a comedy film, it is necessary to write this parameter into the **Type** field, select into the **Info Results** box the information that will appear in the results (for example the creator and the title), and press the Submit button.



A new dialog will appear where it is necessary to insert the User Name and the password to access into the database.

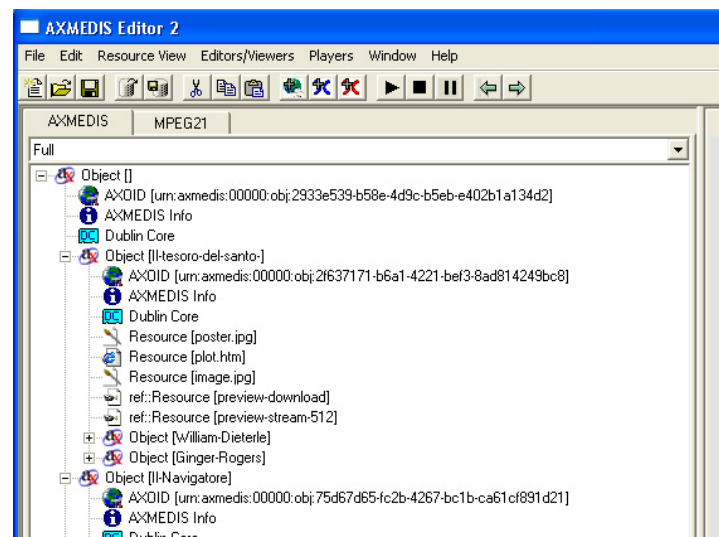


Pressing OK the query is sent to the database and the results are shown in a new window



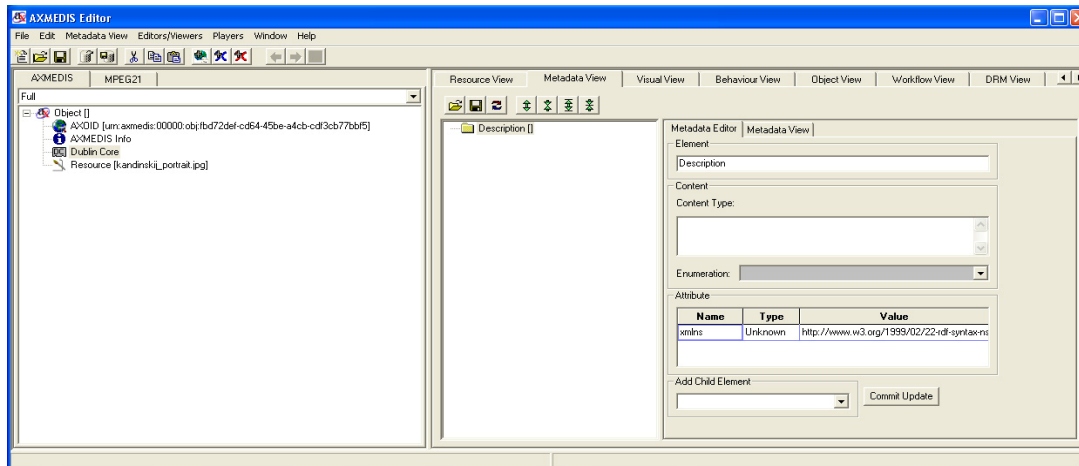
At this point it is possible to select the object in the list and press the OK button to continue. It is also possible to select more than only one object pressing the <SHIFT> or the <CTRL> keys and selecting all the needed objects.

After the OK pressing, The selected objects are added in the Editor as embedded objects (identified by the **Object** prefix).

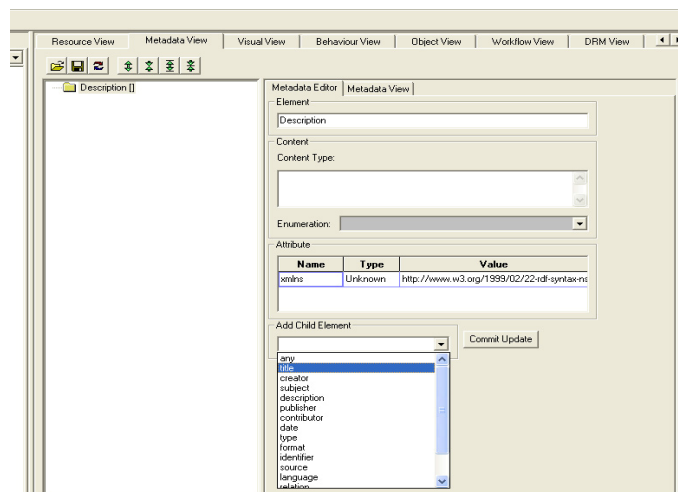


2.3.5 Adding Dublin Core metadata

Double clicking on the **Dublin Core** item the metadata editor is opened:

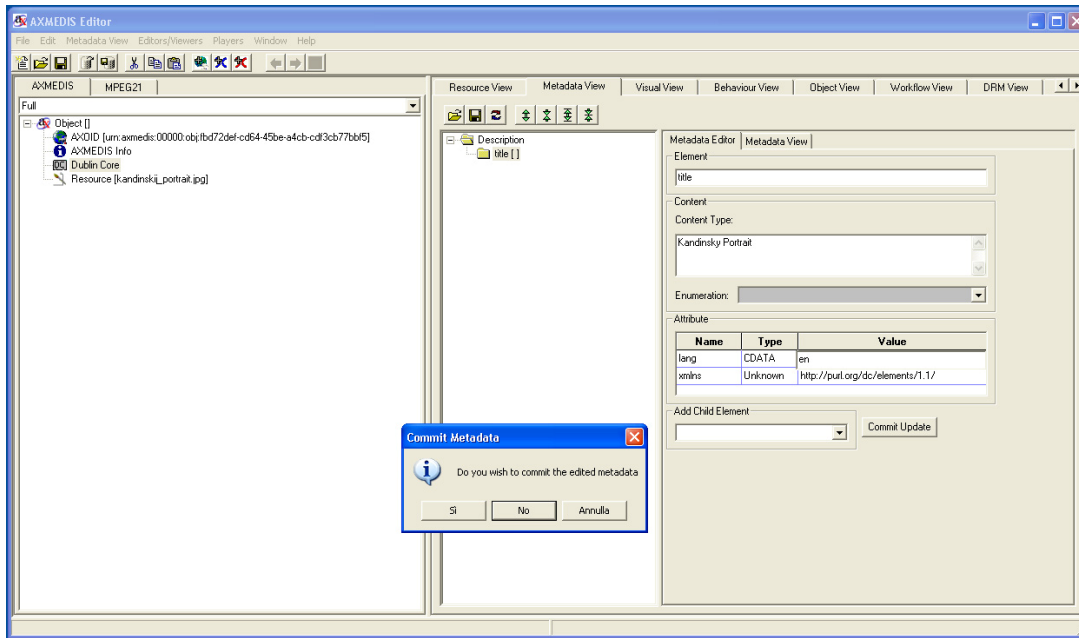


It is possible to add a new metadata element choosing it from the **Add Child Element** cascading menu list

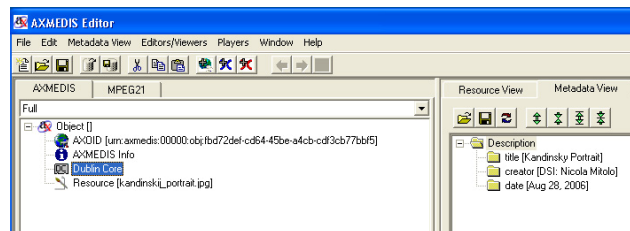


Selecting the **Title** element from the list it is possible to enter the value in the **Content Type** box and the language in the **Attribute** box. To confirm definitely the changes it is necessary to press the **Commit Update** button and confirm the commitment in the next dialog box by pressing the “YES” button.

Please remember that to insert the language element in the **Attribute** box it is necessary to press the *Enter* key to confirm the insertion.

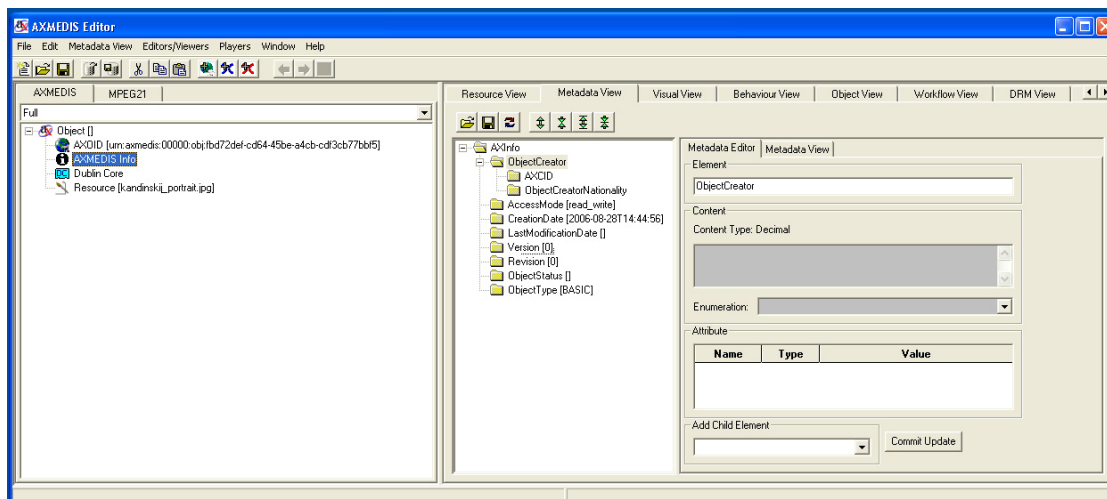


Naturally, using the **Add Child Element** cascading menu it is possible to add more elements in the **Description** tree. Only after the insertion of all the needed metadata elements it is necessary to confirm the changes pressing the **Commit Update** button, which update the Description elements accordingly:




Please note that if you select another resource in the tree view all metadata information will be lost if you have not pressed the **Commit Update** button before.

Double clicking on the **AXMEDIS Info** the **Metadata Viewer** shows the AXMEDIS specific information associated with the object:

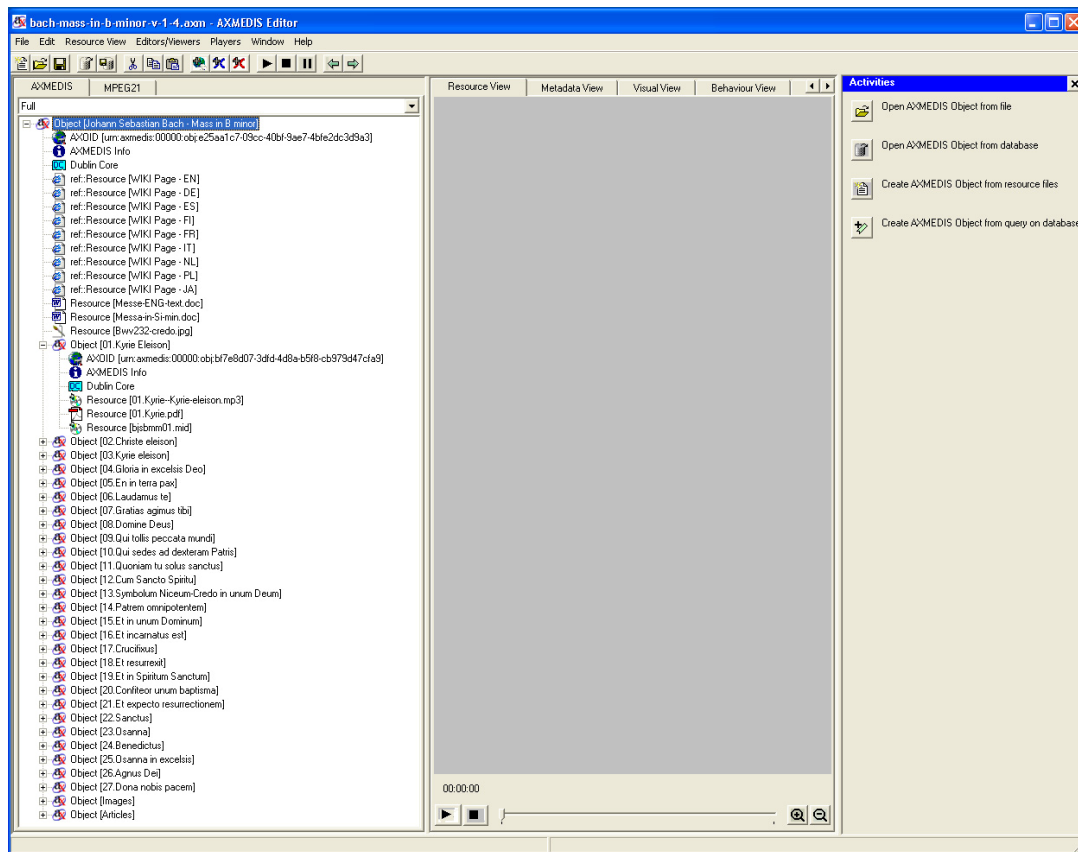


The Object Creator information is automatically added getting information from the configuration (**File/Configuration...** from the menu)


The object can be overwritten on the local hard disk using **File/Save** (or using the  button on the toolbar) or saved as new file using **File/Save As...** from the menu.

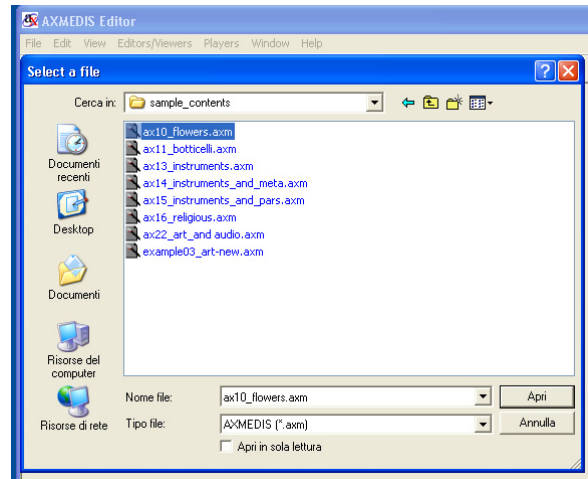
Also the object can be uploaded on the AXMEDIS Database using the  button on the toolbar or **File/Upload into Database...** from the menu.

The AXMEDIS Editor permits to create very easily complex objects with both simple and nested resources. The following figure shows an example of complex object with a big number of digital resources.



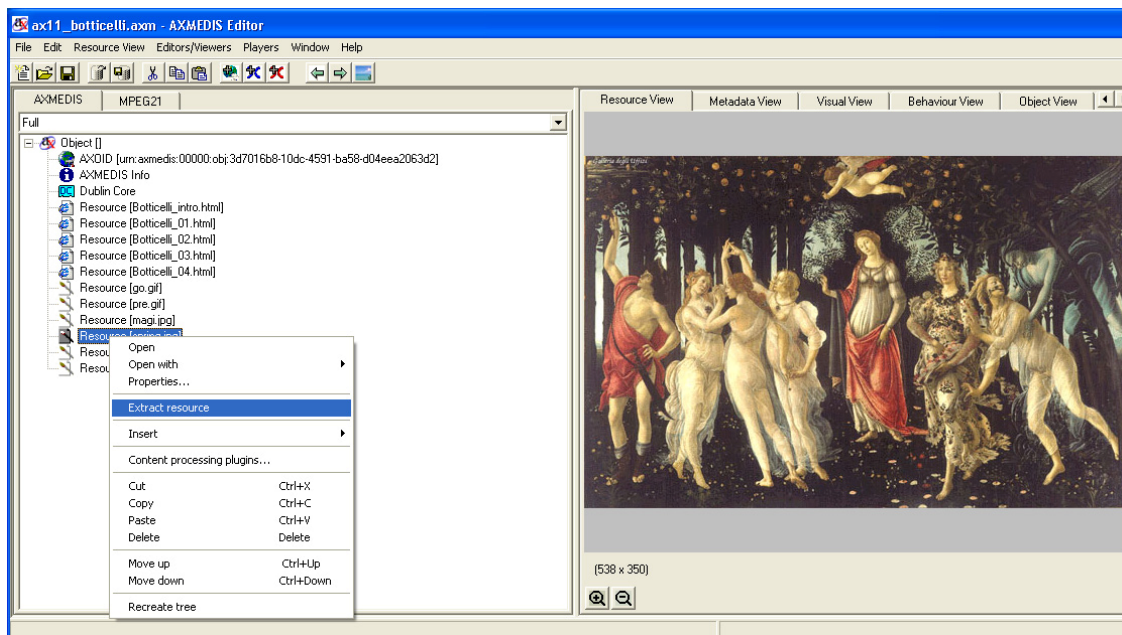
2.3.6 Open an existing object

To open an existing AXMEDIS object select **File/Open...** from the Menu or use the  button on the toolbar. The "Select a file" window appears where it is possible to select an AXMEDIS object to be opened.




2.3.7 Extract an embedded resource

The AXMEDIS Editor permits to extract a resource embedded in an object to be saved as external file. Select the resource in the tree window and right click on it to open the contextual menu. Select **Extract resource**, to save the file on the disk.

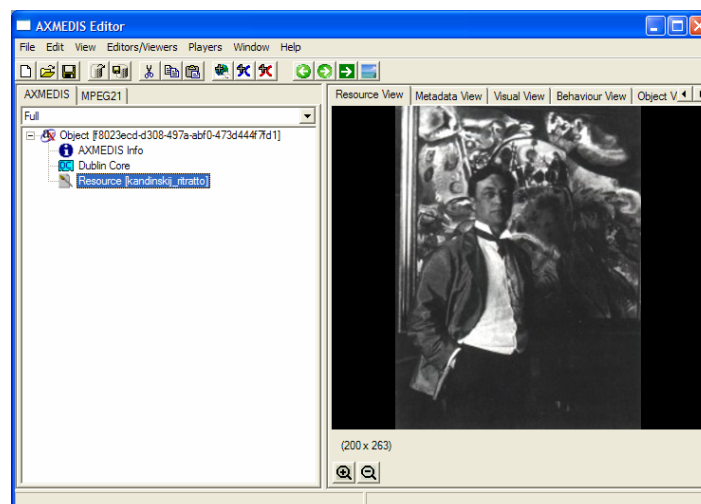


2.3.8 Modifying an AXMEDIS Object stored on Database

To search for the AXMEDIS object on the Database select the  button on the toolbar or using **File/Open from Database...** in the menu, the query dialog is opened and it is searched for an object with the title containing "portrait":

Pressing the Submit button, if the authentication procedure is successfully passed, the query is sent to the database and the results are shown:

Pressing OK the object is recovered from the database and opened:



2.3.9 The Query dialog

The Query dialog is the window that permits to start a query into the database to search a specific AXMEDIS object using parameters defined by the user.

When the Query dialog is open, the user have to choose the sources where apply the query in the **Available Sources** box (1). The predefined available sources are **AXEPTOOL**, **CMS**, **AXMEDIS DB** and **All** of these.

The search criteria supported by the Query dialog are organized into two main groups, (2) search by **Dublin Core** metadata information (with a specific sub-box for the search by creation date) and search by **AxInfo** information (3). Each group, along with its associated set of search criteria, is exposed on the dialog interface. Using this interface, a user enters a search string in the preferred field and chooses the correct criteria using comparison operators associated with that string (4).

The available criteria are:

GT, (>) the searched value is greater than the specified value

LT, (<) the searched value is less than the specified value

EQ, (=) the searched value is equal to the string specified by the user

GE, (>=) the searched value is greater or equal to the searched value

LE, (<=) the searched value is less than or equal to the specified value

NE, (!=) the searched value is not equal to the specified value

STARTWITH, the searched word start with the string specified by the user

ENDWITH, the searched word end with the string specified by the user

CONTAINS, the searched word contains the string specified by the user

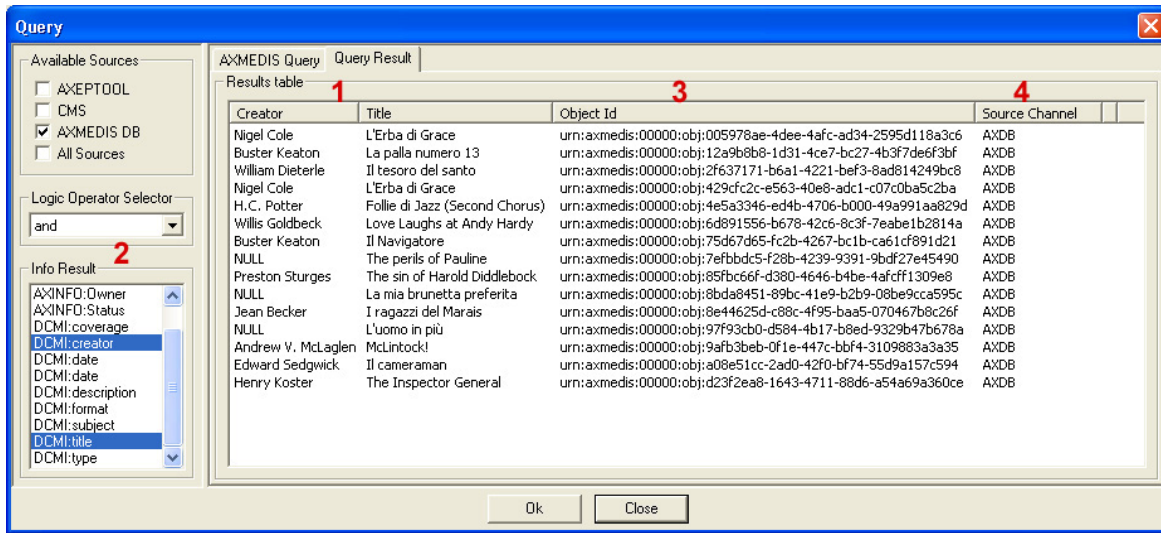
The **Logic Operator Selector** box (5) specifies the and/or option to be applied for all the fields specified by the user.

Once all the search criteria have been specified, a user chooses the **Info Results** (6) to be showed in the **Query Result**. It is possible to select more that one Info selecting them pressing the <SHIFT> or the <CTRL> keys.

The information available for the results are divided in two groups, the **AXINFO** and the Dublic Core Metadata Information (prefix **DCMI**).

The Reset button delete all the inserted information. The Submit button start the query. After the pressing of the Submit button, it is necesasry to insert the Username and the Password in the **Login on Query Support** dialog to authorize the query.

The results of the query are showed in the Query Result tab.

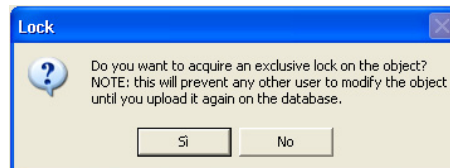


The information showed in the Query Result tab (1) are those selected in the Info Result box (2) followed by the UNR with the object ID (3) and the Source Channel (4).

In the previous figure the results table shows the AXMEDIS objects with Creator and Title fields as selected in the Info Results tab. The string "NULL" indicates that the info is not present in the object.

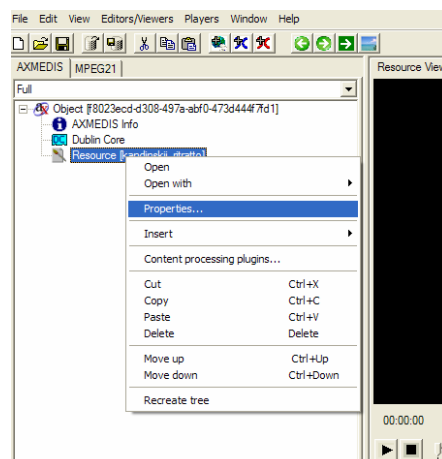
To load an object into the AXMEDIS Editor it is necessary to select it in the list and to press the OK button. It is also possible to select more than one object pressing the <SHIFT> or the <CTRL> keys and selecting all the needed objects.

After the OK pressing again it is necessary to insert the Username and the Password to Login. For some activities it is possible that a **Lock** dialog will appear asking if the user want to acquire the exclusive lock of the object to prevent any other user to modify the same object until the next upload on the database.

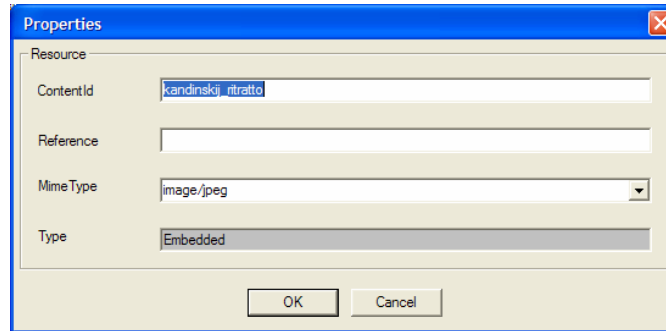


2.3.10 The resource property dialog

The Properties of a resource (and for any other element) can be edited right clicking on the element and selecting **Properties...** from the contextual menu:



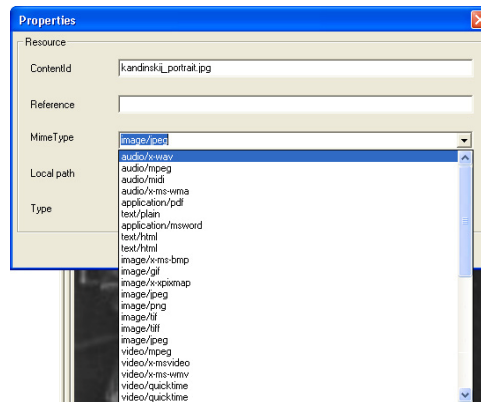
and the following dialog is opened and the properties can be modified:



The **ContentId** is the name identifying the resource that appears in the Tree window closed into square brackets. As default the **ContentId** value is the same as the filename.

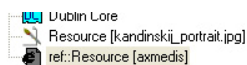
The **Reference** field contains (if present) the URL address of an external resource

The **MimeType** field identifies the file type and the format



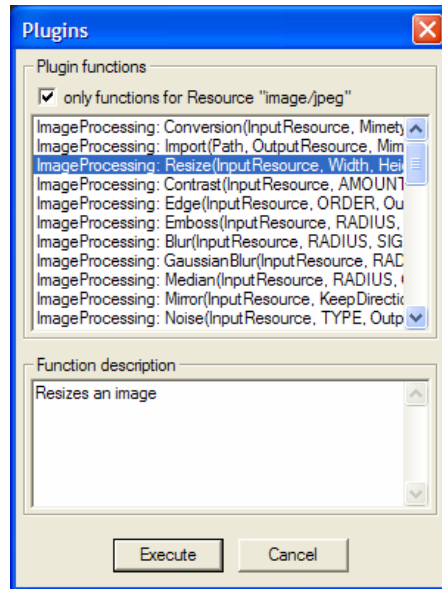
Finally the **Type** field specifies if a resource is embedded in the AXMEDIS object or if is an external resource not physically present into the object.

If a resource is not embedded in the AXMEDIS object, it is represented in the Tree windows with the prefix "ref::". In the following figure, the [AXMEDIS] external resource is a link to an external web site.

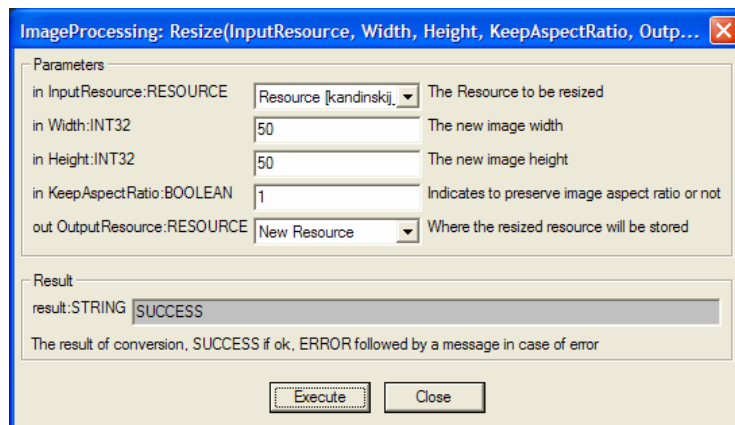


2.3.11 The Content processing plugins

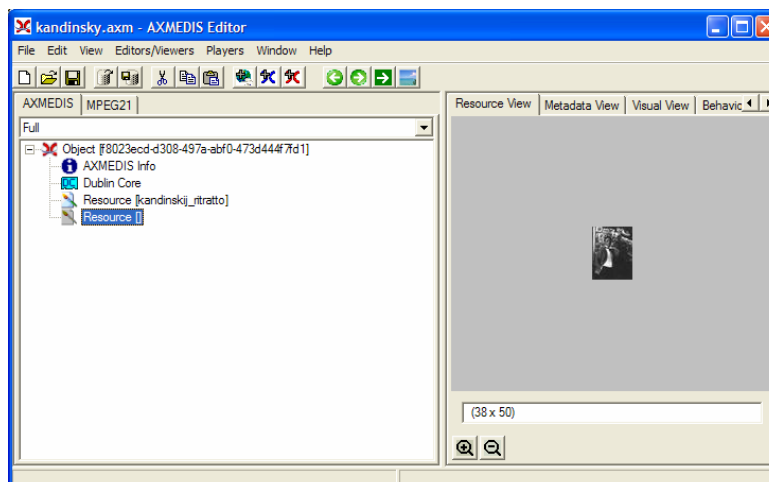
On a Resource the contextual menu enables to use content processing plugins. Selecting the **Content processing plugins...** menu option the plugin function list is provided with the list of applicable plugins functions (based on resource mimetype):



Selecting a plugin function and pressing the **Execute** button a dialog is presented allowing to provide the arguments for the function, clicking on the **Execute** button the plugin function is executed:



In this example we will resize an image using the Resize function in the ImageProcessing plugin. The result is the following:



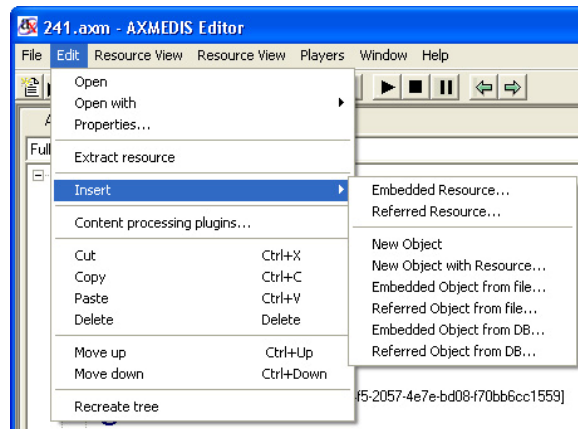
2.4 The AXMEDIS Editor menus

2.4.1 The File menu

The File menu contains a set of functions related to the edited (or to be loaded) object.

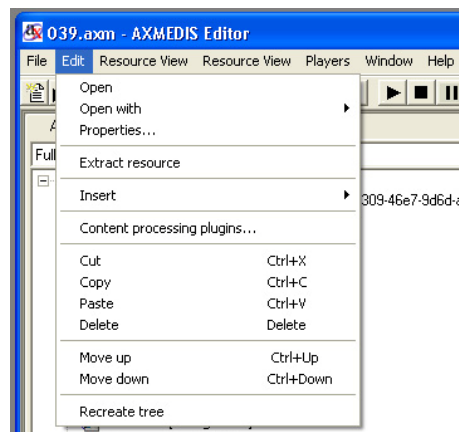
They are:

- **New**, to create a new AXMEDIS object
- **Open**, to open an AXMEDIS object loading it from the disk
- **Open from database**, to open an object searching it from the database
- **Save**, to overwrite the object
- **Save as...**, to save an object specifying the filename
- **Upload into Database**, to upload an object into the database
- **Close**, to close the current opened object
- **Notify Workflow activity completion...**, to be used when the AXMEDIS Editor has been launched from the Workflow Server to notify that the activity to be done has been completed;
- **Configuration**, shows a number of dialogs to set the correct configuration of the Editor
- **Plugins...**, shows a box with the list of available plugins
- **Recent files**, shows a list of the recent opened objects
- **Exit**, to close the AXMEDEIS Editor



2.4.2 Edit menu

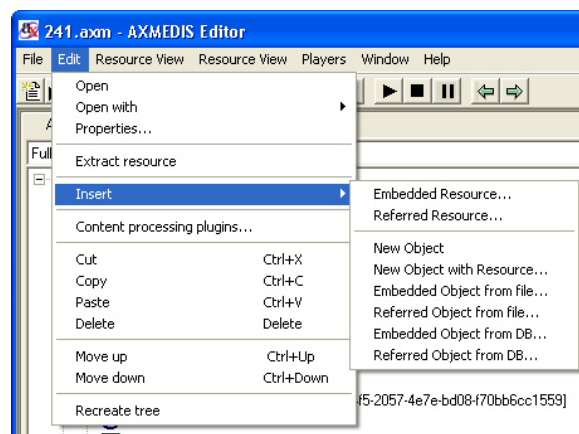
The **Edit** menu contains a set of functionalities available for the resources in the object tree.



When a resource is selected in the tree, using the **Edit** menu is possible to select one of the following features:

- **Open**, open or play the resource in the Resource View panel
- **Open with**, play or view the selected resource selecting one of the viewers listed in the menu

- **Properties**, view and modify the properties of the resource (see section 2.3.10 for a detailed explanation)
- **Extract resource**, to save an embedded resource in the disk as file
- **Insert**, opens a menu with:
 - o **Embedded Resource...**, add an embedded resource in the object (see section 2.3.6 for a detailed explanation)
 - o **Referred Resource...**, add an external resource
 - o **New Object**, insert a new empty object in the tree
 - o **New Object with Resource...**, insert an object in the tree selecting it from the disk
 - o **Embedded Object from file...**, insert and embedded object from file
 - o **Referred Object from file...**, insert a reference to an object choosing it from as file in the disk
 - o **Embedded Object from DB...**, insert an embedded object from the database
 - o **Referred Object from DB...**, insert a reference to an object choosing it from the database



- **Content processing plugins...**, opens a new window with the list of available plugins for the selected resource. Please see section 2.3.11 for more details.
- **Cut, Copy, Paste** and **Delete** options
- **Move up, Move Down** to change the resource position in the tree
- **Recreate tree**, to force the tree update

2.4.3 The Resource View menu

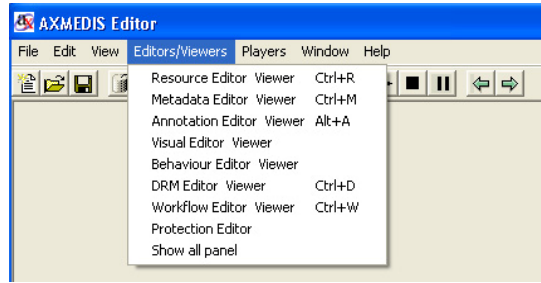
The **Resource View** menu shows a set of functionalities available according with the resource showed in the **Resource View** window. The list of functionalities is the same showed with a right click in the **Resource View** window

2.4.4 The Editors/Viewers menu

The **Editors/Viewers** menu contains the list of available editors and panel that are:

- the **Resource Editor and Viewer**
- the **Metadata Editor and Viewer**
- the **Annotation Editor and Viewer**
- the **Visual Editor and Viewer**
- the **Behaviour Editor and Viewer**
- the **DRM Editor and Viewer**
- the **Protection Editor**

The **Show all panel** function display all the available panels in the Panel View window



Please refer to the detailed section available for each panel for a detailed explanation of the functionalities available.

2.5 AXMEDIS Resource Viewer (DSI)

2.5.1 Main functionalities

The Resource Viewer allows to view and in some cases also edit the resources in the AXMEDIS Object, it is composed of:

- Image Viewer
- Audio Player
- Video Player
- Document Viewer
- SMIL Player
- MPEG4 Player

The functionalities provided are explained in the following sections.

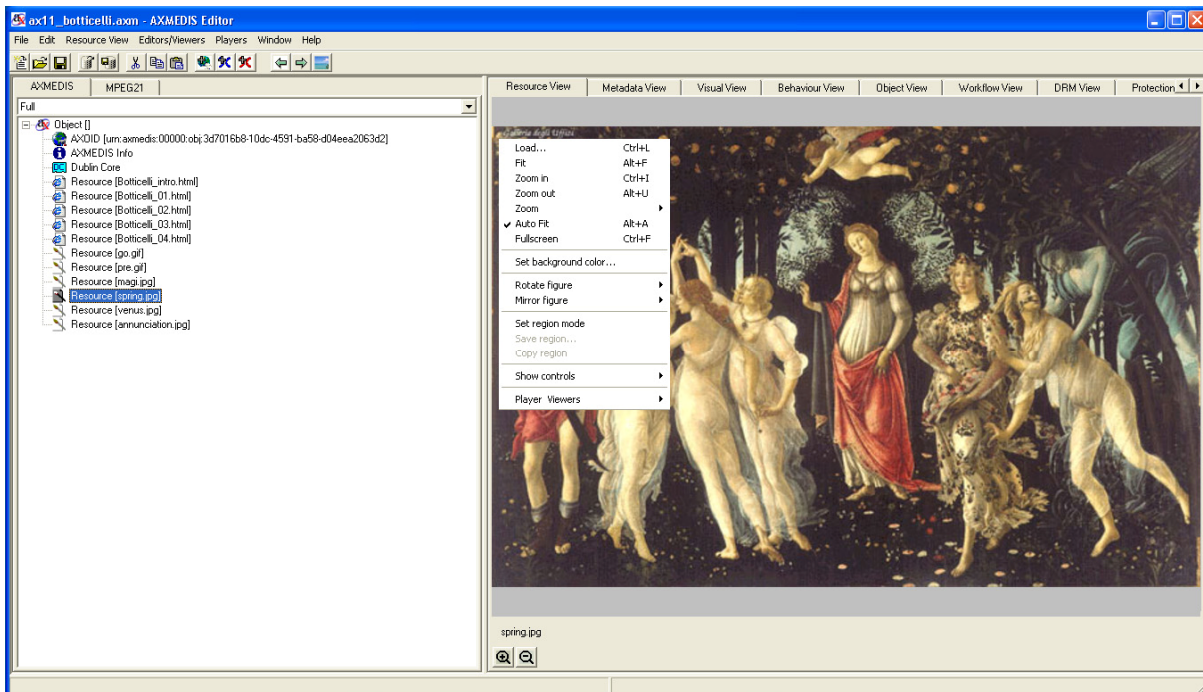
2.5.2 Relationship with other tools

The Resource viewer is integrated as a part of the AXMEDIS Editor.

2.5.3 Image Viewer - Detailed description of the functionalities and Screenshots

The Image Viewer is the part of the AXMEDIS Editor visualizing the images embedded as resources in an AXMEDIS object.

Double clicking on an image resource opens the Image Viewer. Right clicking on the image to open the contextual menu.



Features in the contextual menu of the image player are:

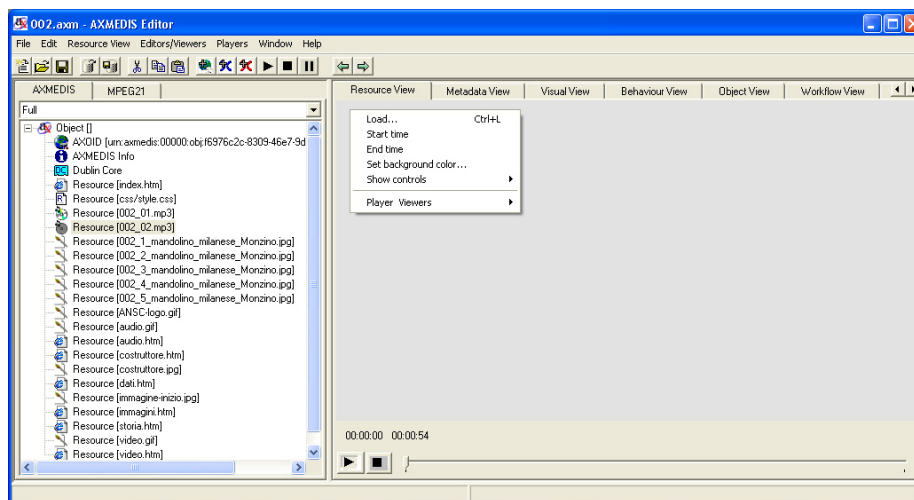
- **Load**, opens an external resource;
- **Fit**, fits the rendered image to the windows size;
- **Zoom In**, enlarges the image size view;
- **Zoom out**, reduces the image size view;
- **Zoom**, from 1% to 3000% to reduce or enlarge the size of the image by the given percent;

- **Fullscreen**, to zoom the image to fit the size of the screen. To return to the normal view size, right click on the image and deselect the “Fullscreen” option in the contextual menu;
- **Set background colour**, changes the colour of the background choosing it using a colour palette
- **Rotate figure**, rotates the image by 90, 180 and 270 degrees
- **Mirror Figure**, creates a mirror image reflecting in the horizontal or in the vertical direction
- **Set region mode**, permits to select a portion of the image and to copy it as new resource in the object or as new file in the disk.
- **Show controls**, hides or shows additional information and functions below the image, such as
 - o the “Status” with the name and the size of the image;
 - o the “Zoom buttons”;
- **Players viewers**, opens a different player.

2.5.4 Audio Player - Detailed description of the functionalities and Screenshots

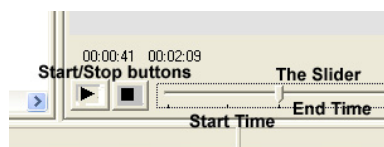
The Audio Player is the part of the AXMEDIS Editor playing the audio resources embedded in an AXMEDIS object

Double clicking on an audio resource opens the Image Viewer. Right clicking on the audio player to open the contextual menu.



Features in the contextual menu of the audio player are:

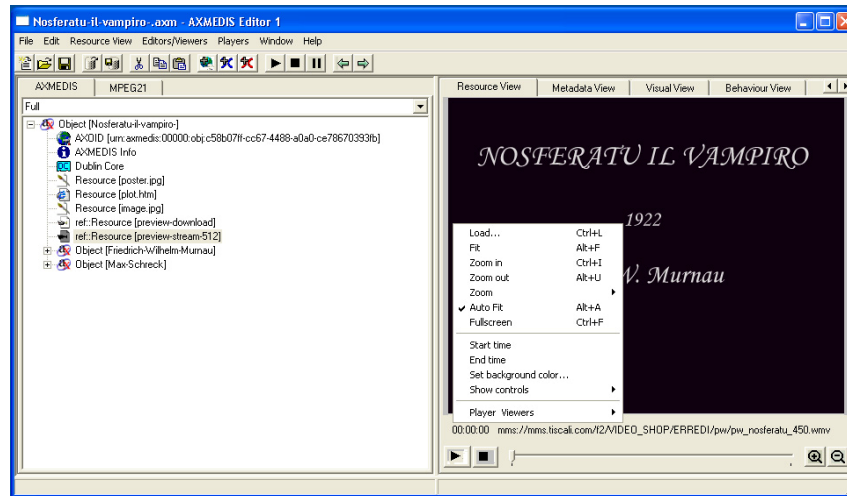
- **Load**, opens an external resource;
- **Start Time**, sets the point to start the audio playing
- **End Time**, sets the point to stop the audio playing
- **Set background colour**, changes the colour of the background choosing it using a colour palette
- **Show controls**, hides or shows additional information and functions, such as
 - o the “Status” with the name of the file and the total duration,
 - o the Current Time,
 - o the Play/Stop buttons
 - o the Slider;
- **Players viewers**, opens a different player.



2.5.5 Video Player - Detailed description of the functionalities and Screenshots

The Video Player is the part of the AXMEDIS Editor playing the videos embedded as resources in an AXMEDIS object

Double clicking on a video resource opens the Video Player. Right clicking in the video player opens the contextual menu.



Features in the contextual menu of the Video Player are:

- **Load**, opens an external resource;
- **Fit**, fits the rendered video to the windows size;
- **Zoom In**, enlarges the video size view;
- **Zoom out**, reduces the video size view;
- **Zoom**, from 1% to 3000% to reduce or enlarge the size of the video by the given percent;
- **Fullscreen**, to zoom the video to fit the size of the screen. To return to the normal view size, right click on the image and deselect the “Fullscreen” option in the contextual menu;
- **Start Time**, sets the point to start the video playing
- **End Time**, sets the point to stop the video playing
- **Set background colour**, changes the colour of the background choosing it using a colour palette
- **Show controls**, hides or shows additional information and functions, such as
 - o the “Status” with the name of the file,
 - o the Current Time,
 - o the Play/Stop buttons
 - o the Slider;
 - o the Zoom buttons
- **Players viewers**, opens a different player.

2.5.6 Document Viewer - Detailed description of the functionalities and Screenshots

The Document Viewer is the part of the AXMEDIS Editor visualizing the documents embedded as resources in an AXMEDIS object

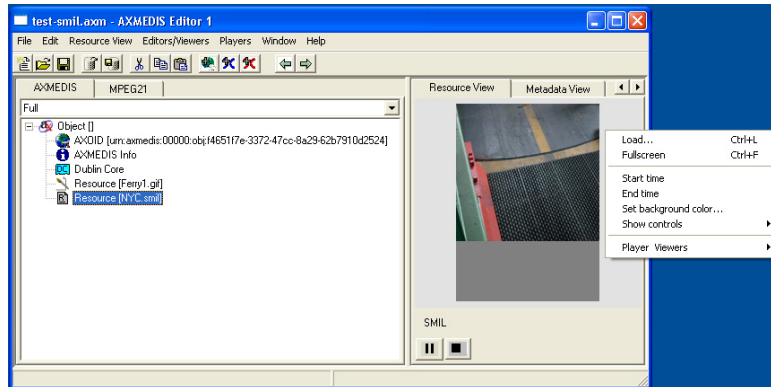
Double clicking on a document resource (typically an html, pdf, .doc, .rft, or .txt resource) opens the corresponding Document Player according to the document type.

Particularly the html viewer permits to view images or play resources linked into the html code and stored as single resources into the same AXMEDIS object.

2.5.7 SMIL Player - Detailed description of the functionalities and Screenshots

The SMIL Player is the part of the AXMEDIS Editor visualizing the SMIL resources embedded in an AXMEDIS object

Double clicking on a SMIL resource in the tree opens the SMIL Player. Right clicking in the SMIL Player opens the contextual menu.



Features in the contextual menu of the SMIL Player are:

- **Load**, opens an external resource;
- **Fullscreen**, to view the SMIL resource in fullscreen. To return to the normal view, right click on the image and deselect the “Fullscreen” option in the contextual menu;
- **Start Time**, sets the point to start the SMIL resource playing
- **End Time**, sets the point to stop the SMIL resource playing
- **Set background colour**, changes the colour of the background choosing it using a colour palette
- **Show controls**, hides or shows additional information and functions, such as
 - o the “Status” with the name of the file,
 - o the Play/Stop buttons
- **Players viewers**, opens a different player.

2.5.8 MPEG4 Player - Detailed description of the functionalities and Screenshots

The MPEG-4 player is a part of the AXMEDIS Editor playing the MPEG-4 resources embedded in an AXMEDIS object.

Functionalities are the same of the Video player. Please, see section 2.5.5 for more details.

2.6 AXMEDIS Metadata Editor (UNIVLEEDS)

2.6.1 Main functionalities

The Metadata Editor allows the user to add, edit, delete and view metadata elements including Dublin Core and AXMEDIS Info (AxInfo) using a simple HCI interface with pop up menus and editing boxes.

2.6.2 Relationship with other tools

The Metadata Editor is integrated as a part of the AXMEDIS Editor.

2.6.3 Detailed description of the functionalities and Screenshots

The Metadata Editor view can be opened by selecting “Metadata View” in the tab or by double clicking in the Hierarchical View on the left side of the main AXMEDIS Editor. We see in Figure 1 the AXMEDIS Editor with the Metadata Editor comprising of a Tree View displaying the metadata elements and editing windows to edit the element content type.

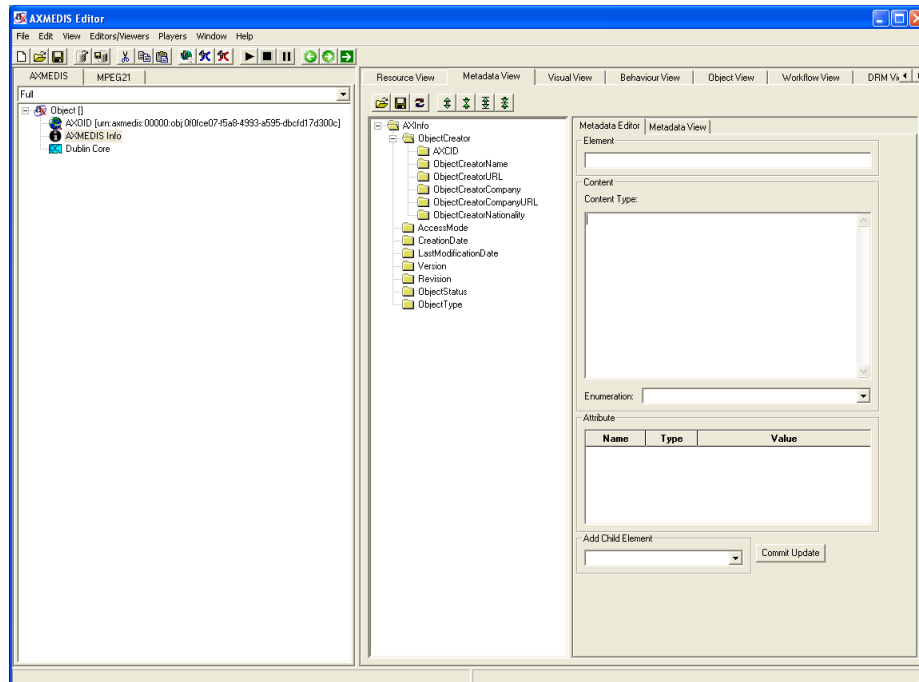


Figure 1

The Editor provides the following functionalities:

- Adding child metadata elements to a Metadata element
- Inserting metadata elements
- Deleting Metadata elements
- Editing Metadata elements

2.6.3.1 Adding Metadata Elements

Adding metadata elements to the metadata is achieved by right clicking on the element the user wishes to add a child to. A pop menu appears and navigating to the “Add New Child Element ...” a list of valid child elements is presented. These elements are derived from the metadata Grammar inserted when loading metadata with an associated schema.

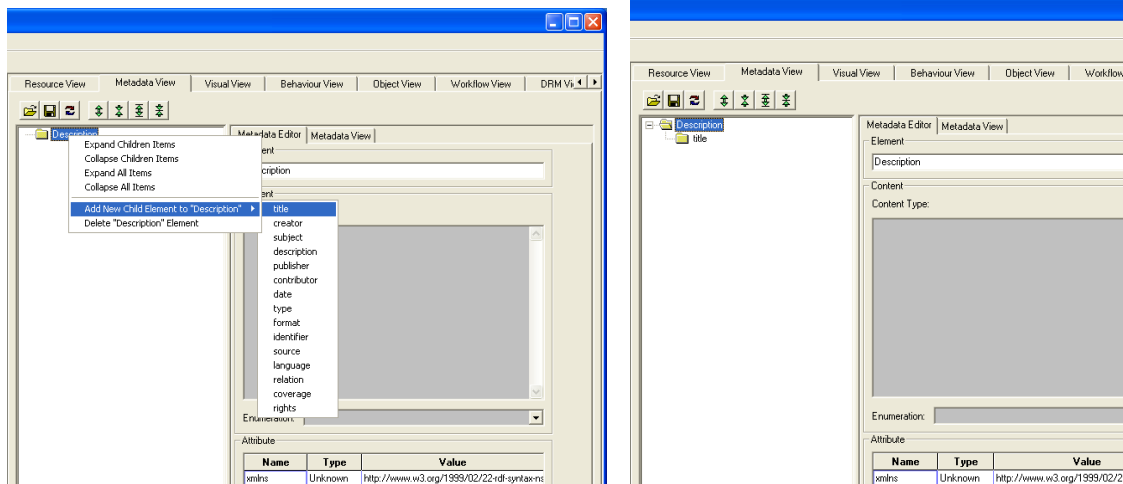


Figure 2

Another method for adding a child element is to use the Add Child drop down box in the main panel highlighted in Figure 3.

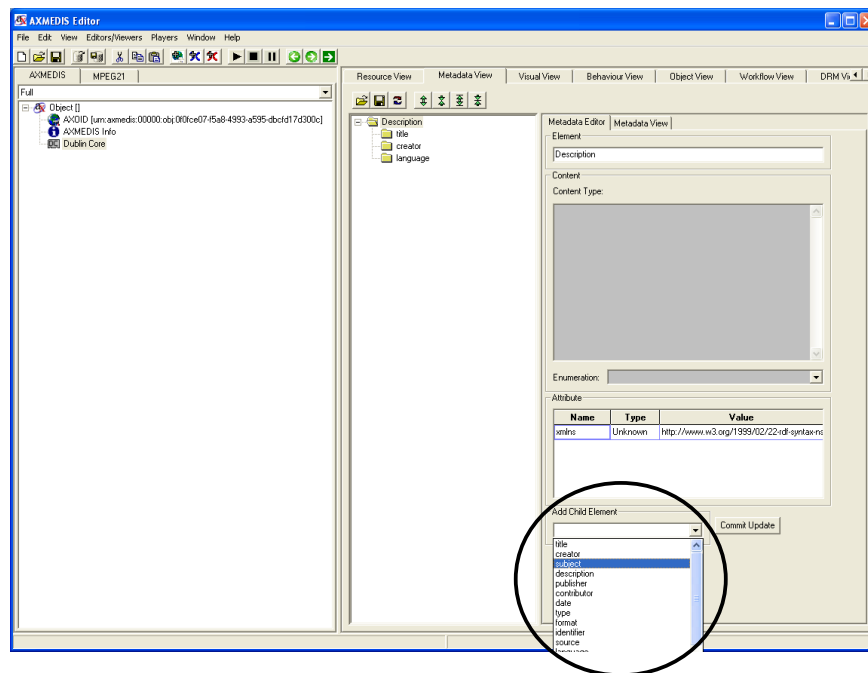


Figure 3

2.6.3.2 Inserting Metadata Elements

To insert elements into the metadata is similar to the Adding functionality. To insert an element, the user right clicks on the element. The new element will be inserted above the selected element as shown in Figure 4. The left screenshots demonstrate the user right clicking on the Dublin Core Element “language” and inserting the Dublin Core element “creator”. In the right screenshot, the tree view shows the “creator” inserted above “language” and below “title”.

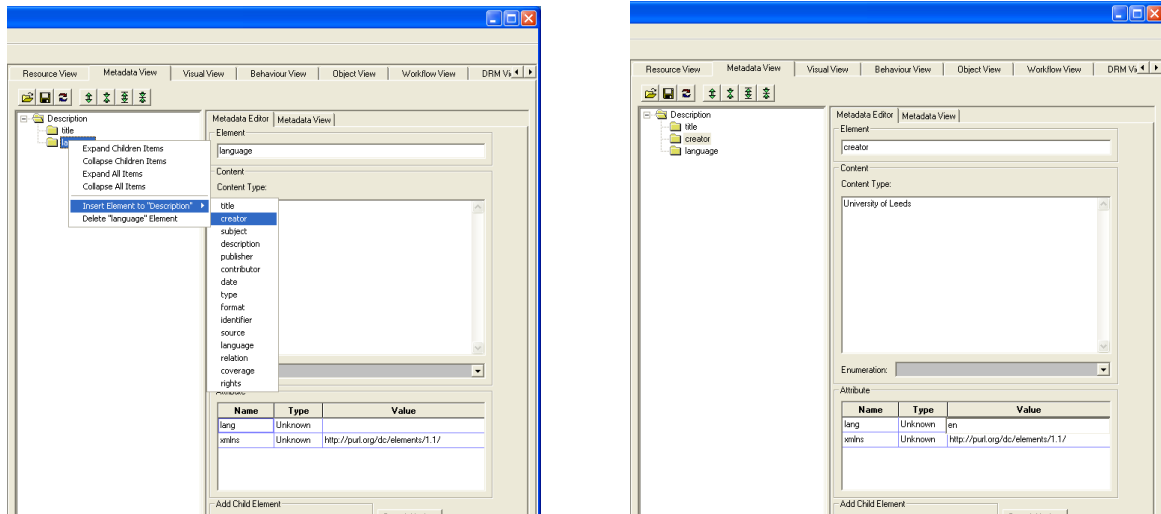


Figure 4

2.6.3.3 Deleting Metadata Elements

The user right clicks on the element they wish to delete and navigates the pop up menu to remove the metadata element (see Figure 5). In this example the Dublin Core element “Creator” is removed.

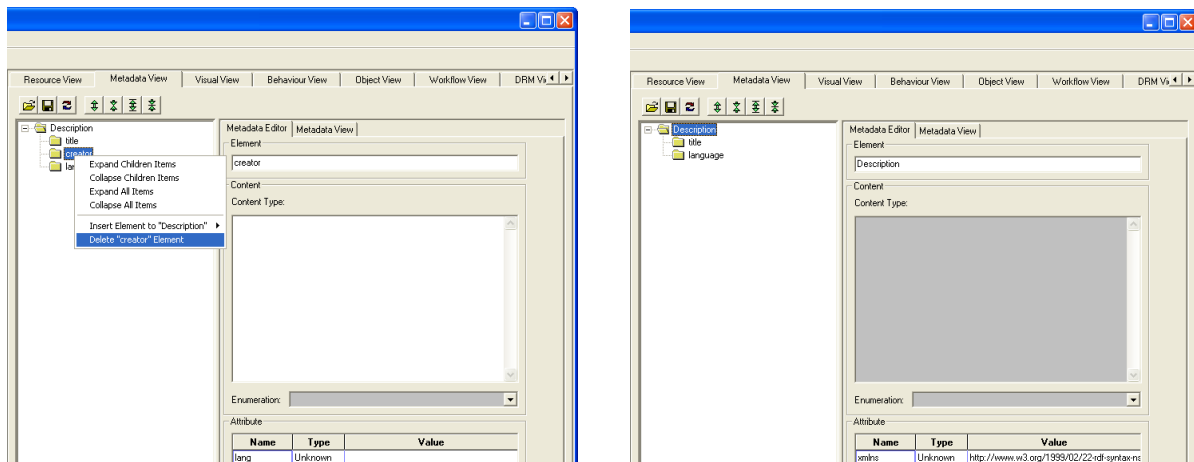


Figure 5

2.6.3.4 Editing Metadata Elements

Editing the metadata element is achieved by using the text box in the Content frame, the “Enumeration” drop down box and the Attribute grid as shown in Figure 6. Fields are edited by selecting the view and adding the values you wish to add the metadata. If the schema does not allow the fields to be edited, for example there are no enumerations in the Figure 6. In Figure 5, the element currently selected does not have a text node and therefore both the content and enumeration area are grayed out.

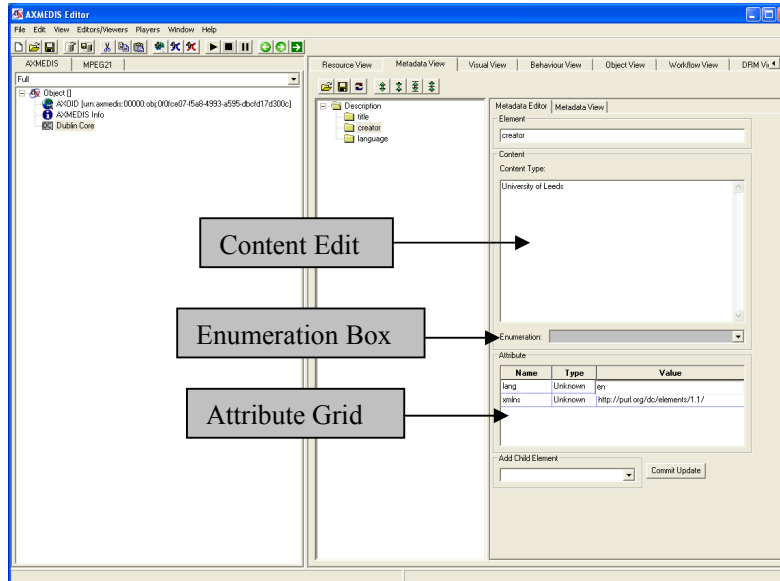


Figure 6

To edit elements with enumeration choices, the user selected the enumeration value they require from the drop down box as shown in Figure 7.

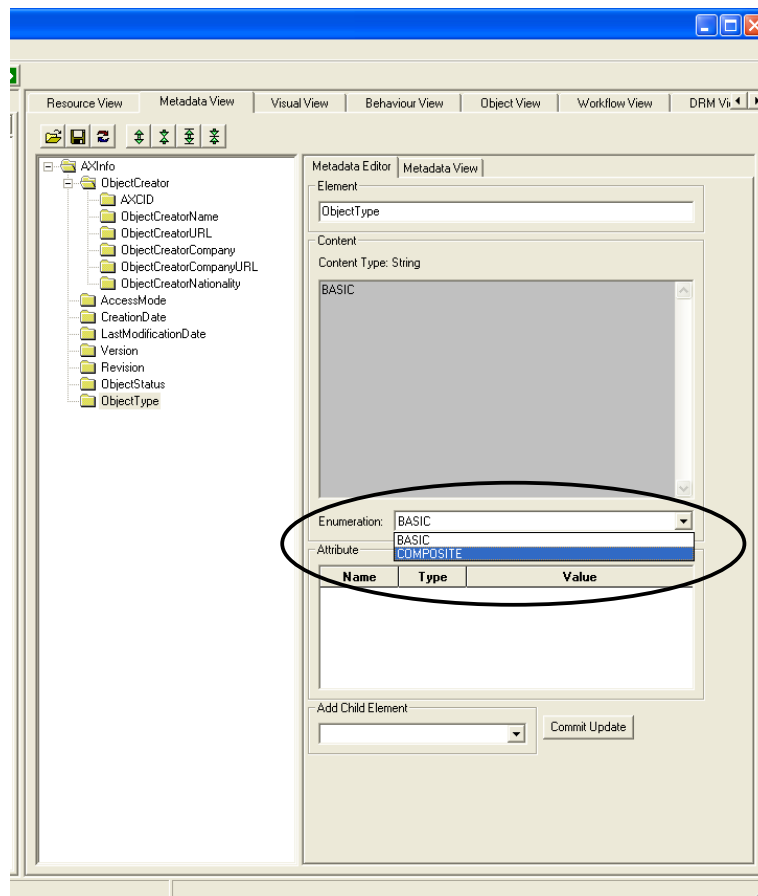


Figure 7

2.7 AXMEDIS Metadata Mapper Editor (UNIVLEEDS)

2.7.1 Main functionalities

This is a GUI interface where a user can define mapping information to enable transformation between metadata languages. This mapping can be used to generate a stylesheet which can then be used to transform metadata information.

2.7.2 Relationship with other tools

The resulting stylesheet can be used to transform metadata documents using the content processing tools.

2.7.3 Detailed description of the functionalities and Screenshots

2.7.3.1 Creating a Transformation XSLT to map Metadata

The main window of the metadata mapper consists of three components, the left hand component displays a tree view of the source metadata once it has been loaded, the right hand component displays a tree view of the destination metadata and the middle component is used to show the relationship between elements on either side.

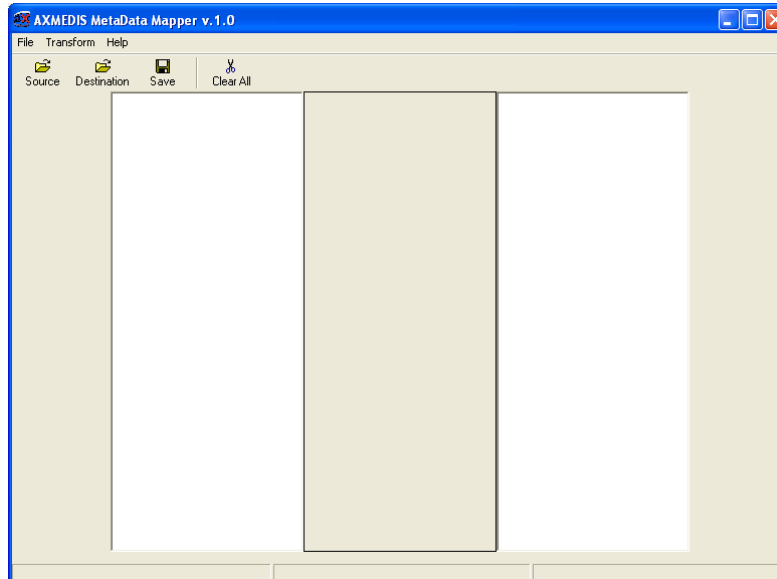


Figure: Metadata Mapper

2.7.3.2 Creating a Transformation XSLT to map Metadata

To create an XSLT stylesheet, we first need to load two metadata documents. The first should contain an instance of the metadata you want to transform from (source metadata language), the second should contain an instance of the metadata you want to transform to (destination metadata language). This is achieved by selecting the source and destination toolbar buttons in the metadata mapper tool and navigating the file open dialogue to the documents you want to open.

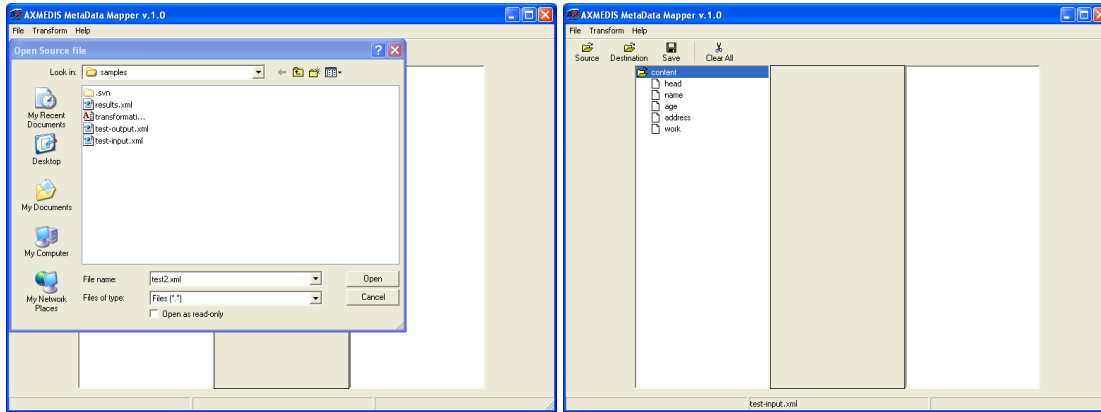


Figure: Opening the Source File

Once the two metadata documents have been successfully loaded, they will be displayed in the tree components as shown in the screenshot below.

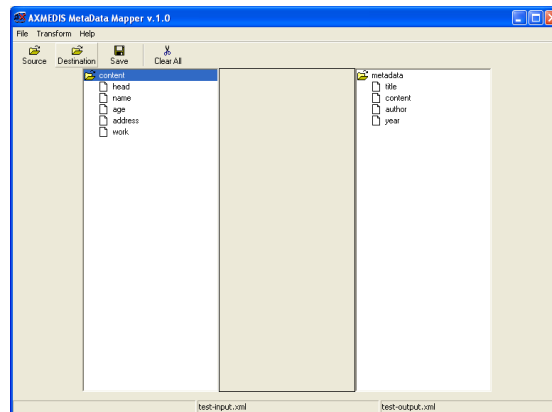


Figure: After opening source and destination Files

Relationships can now be made by selecting an element from the left hand side and then selecting the related element from the right hand side. These connections can be updated by selecting a new element on the right hand side while the left hand side of a connected node is selected. A connection can be deleted by right clicking the connected element.

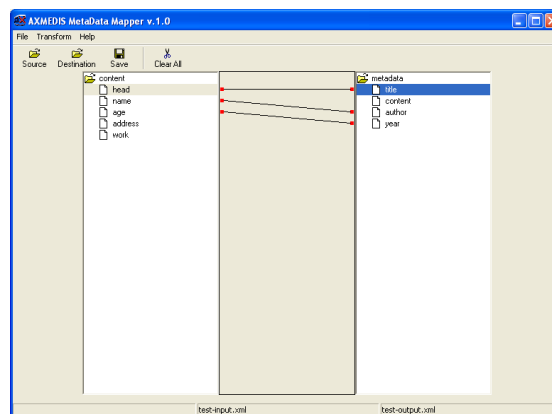


Figure: Creating mapping between source and destination

Once the user has finished mapping metadata from source to destination language a XSLT stylesheet can be saved which contains all the connection information. This is achieved by selecting the save toolbar button and saving a file using the ‘file save’ dialogue.

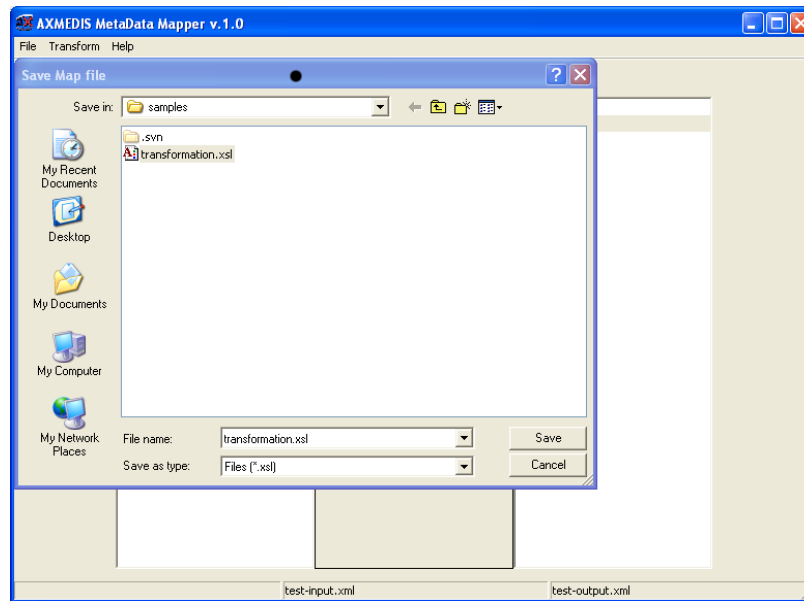


Figure: Saving Mapping as an XSLT file (.xsl)

2.7.3.3 Transforming Metadata using the XSLT

Now that we have a stylesheet describing how to map metadata from the source language to the target language, we can transform a document. This is achieved using the ‘Transform’ menu. First an input file is selected to be transformed; then an output filename is chosen for the transformed metadata; finally the transform function is called and the metadata is mapped to the destination language according to the stylesheet rules and saved to file.

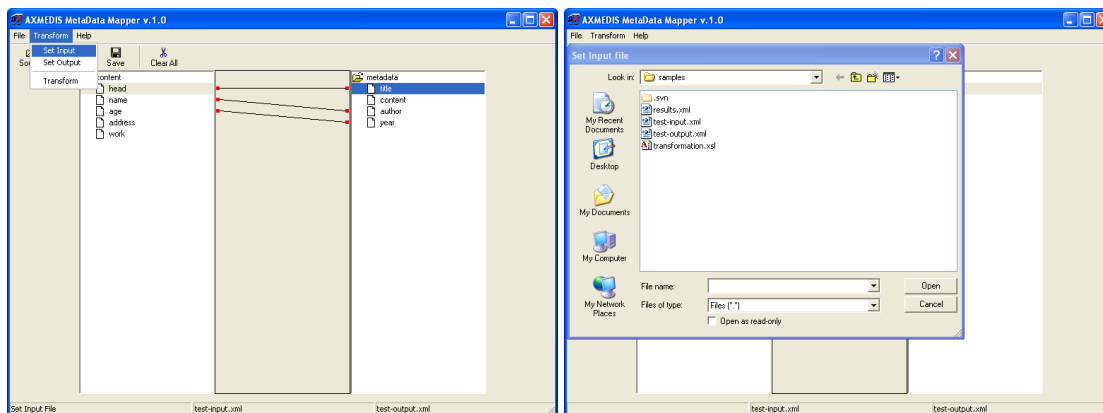


Figure: Setting Metafile to transform after creating anXSLT file (.xsl)

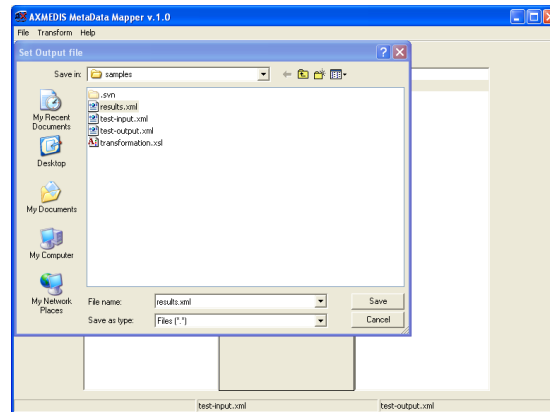


Figure: Setting an output file for the new Metadata after transformation

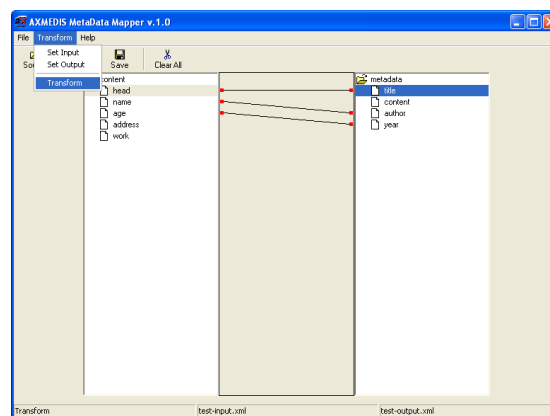


Figure: Transforming the Metadata

2.8 AXMEDIS Visual and Behaviour Editor (EPFL)

2.8.1 Main functionalities

The SMIL editor and player allows user to create, edit and display SMIL content.

2.8.2 Relationship with other tools

The Metadata Editor is integrated as a part of the AXMEDIS Editor.

2.8.3 Detailed description of the functionalities and Screenshots

2.8.3.1 *Editing the SMIL components with Hierarchy Editor and Viewer*

After viewing the media resource inside the AXMEDIS object, the user can use the SMIL editor to create SMIL content. The SMIL editor is composed of three authoring modular: Hierarchy Editor, Visual Editor and Behavior Editor.

The Hierarchy Editor allows the user to construct and edit SMIL hierarchy components. The hierarchy editing of SMIL components is the first step before the spatial visual editing and temporal synchronization. To create an element inside SMIL object, we have to determine what kind of structure of these elements. To be more specific, we have to define it is container (par/seq) or object. The Figure 1 is a snapshot of the GUI of the hierarchy structure. With the right click the element or container, there will pop up a menu list of “add container par” “add container seq” “add element”, “delete the children”, etc which helps the user to modify the structure of the SMIL hierarchy structure.

2.8.3.2 *Editing the visual scenes with Visual Editor and Viewer*

The user can use the visual editor to create visual scenes by toggling the square on the left and left click the canvas. The user can choose the positions of these visual scenes and the size of them. Also the user can entitle each visual scene with some names or some comments. After creating the visual scenes for holding the media resources, the user can choose the visual scenes for the media resources. On the Hierarchy Editor, right click the element, there will pop up a combo box to select the relevant regions which are shown on the visual editor.

Figure 1 is a snapshot of showing visual editor and viewer to demonstrate the visual scenes

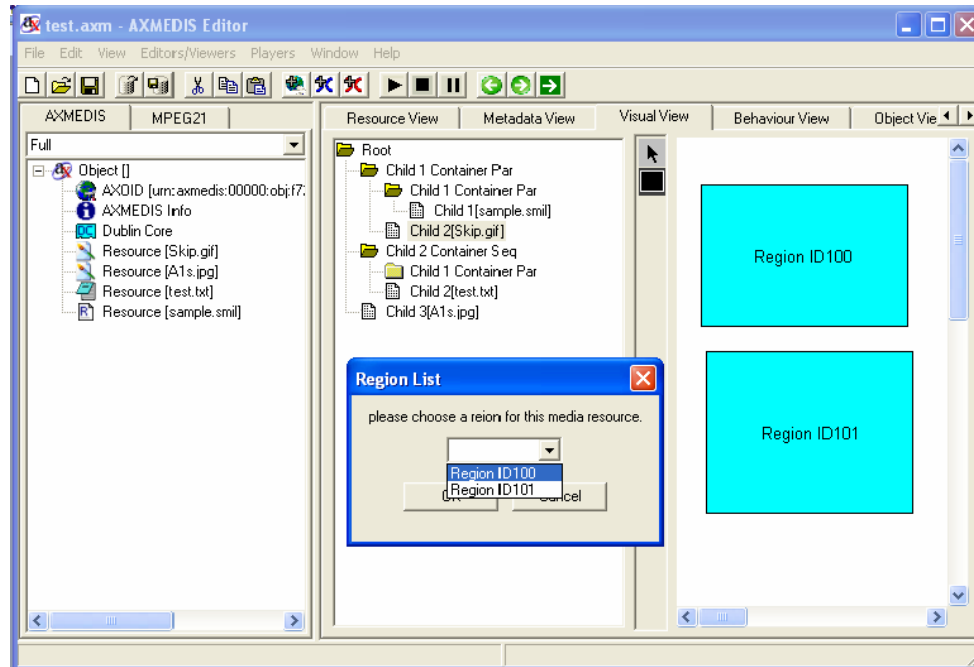


Figure 1 – Visual editor and viewer

2.8.3.3 Editing temporal behaviour of media resources with Behaviour Editor and Viewer

When the media resources are included into the visual scenes by visual editor and viewer, the user can use behaviour editor and viewer to edit the temporal behaviour of the media resources. The media resources included inside the visual scenes will automatically be represented on the canvas of the behaviour editor. And when the visual scenes are removed by the visual editor and viewer, the behaviour editor will remove the resources inside this visual scene correspondingly. There is a timeline on the top of the canvas with the appearance like a ruler with unit and integral value. This timeline is to help the user to edit the displaying temporal information of the media resources. For each media resources, there is a bar to represent the start and end of the displaying time for the media resources. The user can modify the position and the length of the bar to modify the temporal behaviour of the media resources.

Figure 2 is a snapshot showing behaviour editor and viewer to demonstrate the timeline and the bar.

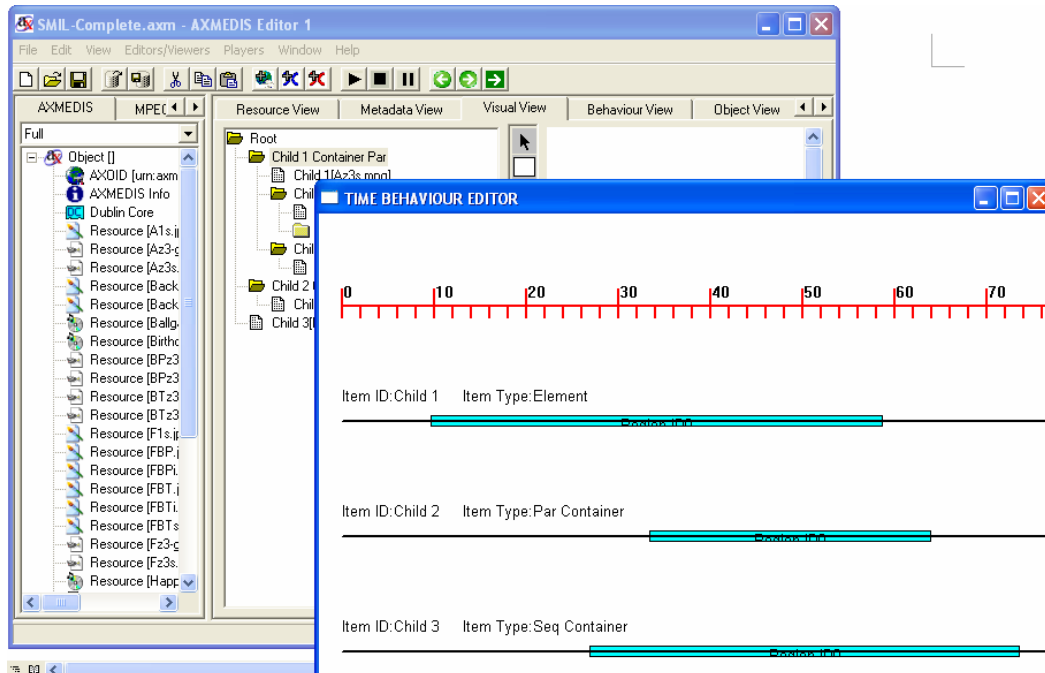
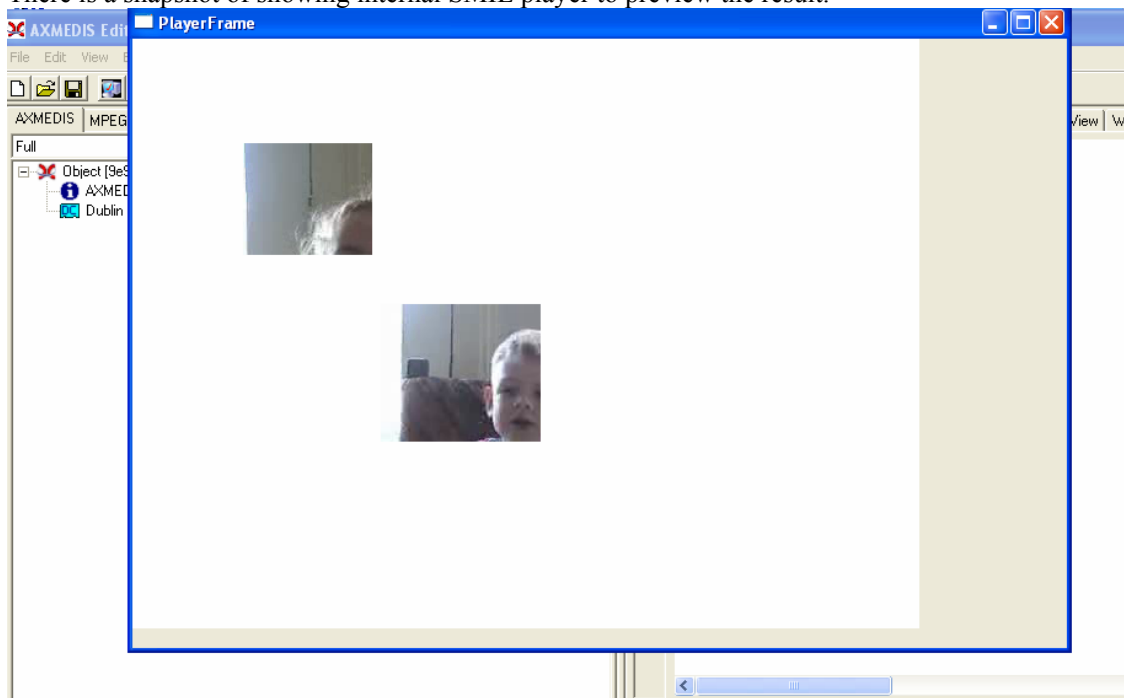


Figure 2 – Behaviour editor and viewer

2.8.3.4 Previewing the result with SMIL player

After editing SMIL, the user can save it and display it with the internal SMIL player. There is a menu on the frame of the axeditor with a list of options “SMIL Player: Load”, “SMIL Player: Play” “SMIL Player: Stop” “SMIL Player: Pause”.

There is a snapshot of showing internal SMIL player to preview the result.



2.9 AXMEDIS DRM Editor (FUPF)

2.9.1 Main functionalities

“*DRM Editor and Viewer*” is a software application to view and eventually edit MPEG-21 REL Licenses. This user manual refers to its version 2.1.9. “*DRM Editor and Viewer*”. It has editing functionalities, being capable of integrating as well in a future axviewer with only viewing features.

Its main features are:

Feature
Load a license from a XML file
Load license from a remote host (PMS Server)
Create a new license form the scratch
Display graphically the license
Store a license in a XML file
Store a license in a remote host (PMS Server)
Edit (modify) a License
Attach a license to an AXMEDIS Object
Create PARs, and internal PARs.

2.9.2 Relationship with other tools

The licenses created with the AXMEDIS DRM Editor can be stored locally or in a remote server (AXMEDIS PMS Server). In order to be able to store licenses in the remote server, a connection to it must be available, as explained in next chapter.

2.9.3 Detailed description of the functionalities and Screenshots

The main window shows two areas: in the left a hierarchically structure is displayed (also called tree), which is used to navigate through the license elements; in the right side there is a panel showing information related to the element marked in the tree. Functionalities can be accessed through the menu options, the buttons in the button bar, or the buttons that appear in the panel. Next we summarise the DRM Editor and Viewer main functionalities:

- DRM Editor and Viewer. Open a license from a file (see Figure 1). The panel displays the file path or an alternative description.

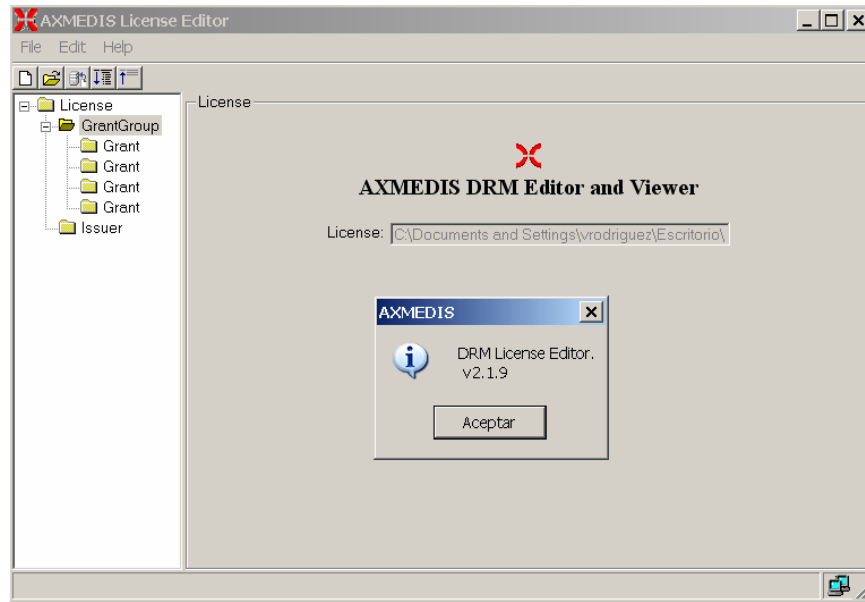


Figure 1: DRM Editor view

- DRM Editor. Create a new license . When a new license is created, it is empty. The tree appears with an “issuer” element and with a “Grant Group” element, but they are initially empty.
- DRM Editor. Modify a license.

When the tree is clicked on its “issuer element”, the Editor shows the issuer panel, where the issuer and the date of issuance can be set (see Figure 2).

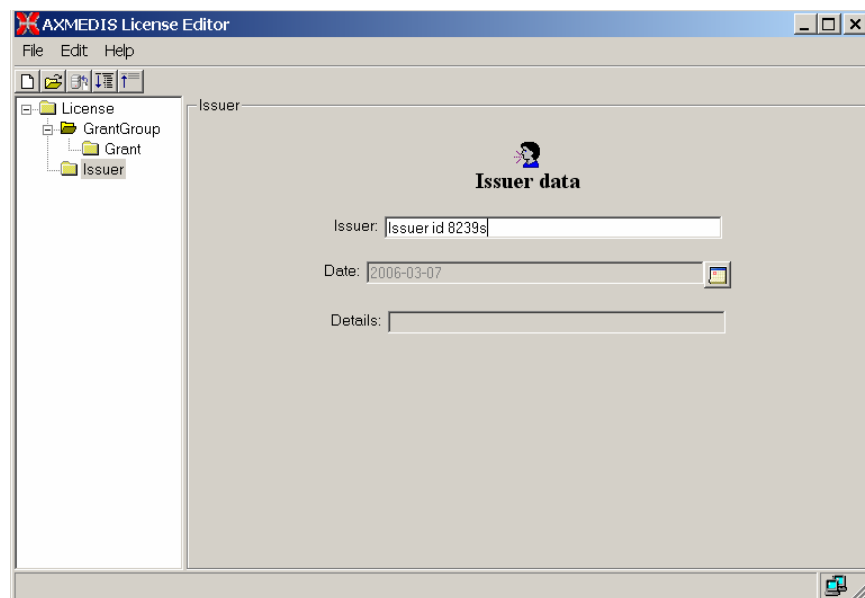


Figure 2: DRM Editor License Issuer view

When the tree is clicked on its “GrantGroup element”, a Grant Group panel is displayed. (see Figure 3). From this panel it is possible to add or delete grants. The first icon creates an empty grant, the second deletes it. Once a grant has been created, it will be displayed in the tree.

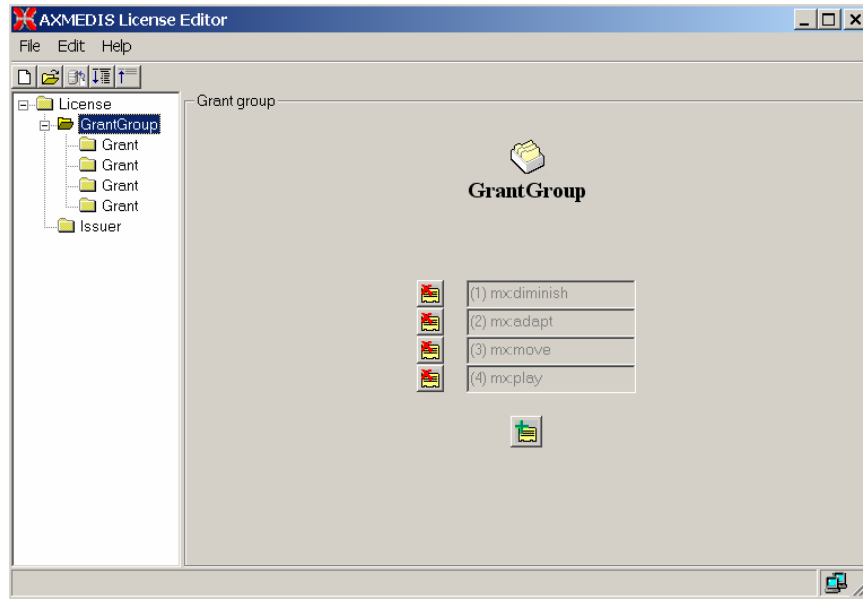


Figure 3: DRM Editor and Viewer License Grant Group view

- When the tree is clicked on its “Grant” element (see Figure 4), the Grant panel is shown. From this panel, the following actions can be performed:
 - To change the following fields: principal, right, resource.
 - To add and remove conditions. The conditions that can be set are: number, interval, territory and/or fee. Each of them can be modified.

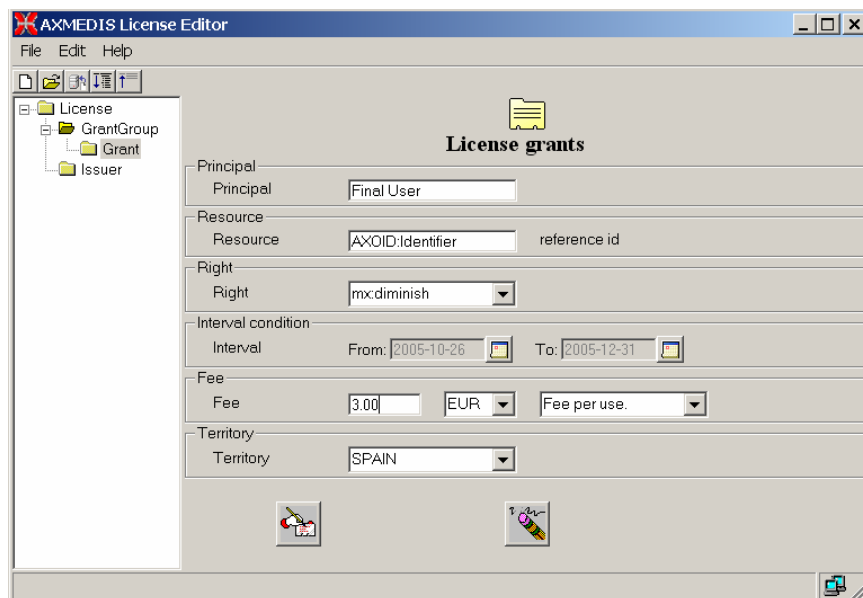


Figure 4: DRM Editor License Grant view

- When the *issue* right is chosen, the license becomes a “distributor license”, and the Grants underneath are Grants to be given by the distributor. This is reflected in the tree, where a new sublevel is displayed containing additional grants.

- DRM Editor. Store a license to a file. This option is performed when selection the “File -> Save to File” option of the Editor.
- DRM Editor. Store a license into a remote server .

Limitations to be fixed in subsequent versions:

- In the current version only up to 8 grants can be added to the same grant group.
- Only a condition of each type can be added to a grant.

If the DRM Editor and Viewer are used in its classic version (the one which displays a simple tree) these limitations are overcome.

2.10 AXMEDIS Workflow Editor (IRC)

2.10.1 Main functionalities

The AXMEDIS workflow Editor/Viewer provides user with the workflow information for the object being edited/viewed. It also shows the Workflow Engine's Interface for the user to see his worklist.

The Workflow information that is displayed are as follows:

Parameter	Information
Title	The
Process	The workflow process that is being executed for the selected AXOID
Activity	The Activity that is currently being executed for the selected AXOID
Priority	The priority of the Workflow Process
Status	The status of the Workflow Process
Actor	The Actor responsible for the current Activity
AXRQID	The Workflow Request ID issued by the workflow engine for current request.

2.10.2 Relationship with other tools

This tool is embedded as a part of Axeditor. It uses workflow editorPlugin to communicate with the Workflow Engine through Workflow Gateways.

2.10.3 Detailed description of the functionalities and Screenshots

- In order to view the workflow information, the user should select the “Workflow View” tab in the Axeditor. On activation the editor will show empty fields for the parameters.
- Meanwhile it will retrieve the “Work List” for the logged user from the workflow engine.
- When the user clicks on the “Request Workflow Information” button, the editor retrieves the workflow parameters from the Workflow Engine and displays in the upper half of the viewer.
- If the information cannot be retrieve an error message is displayed.

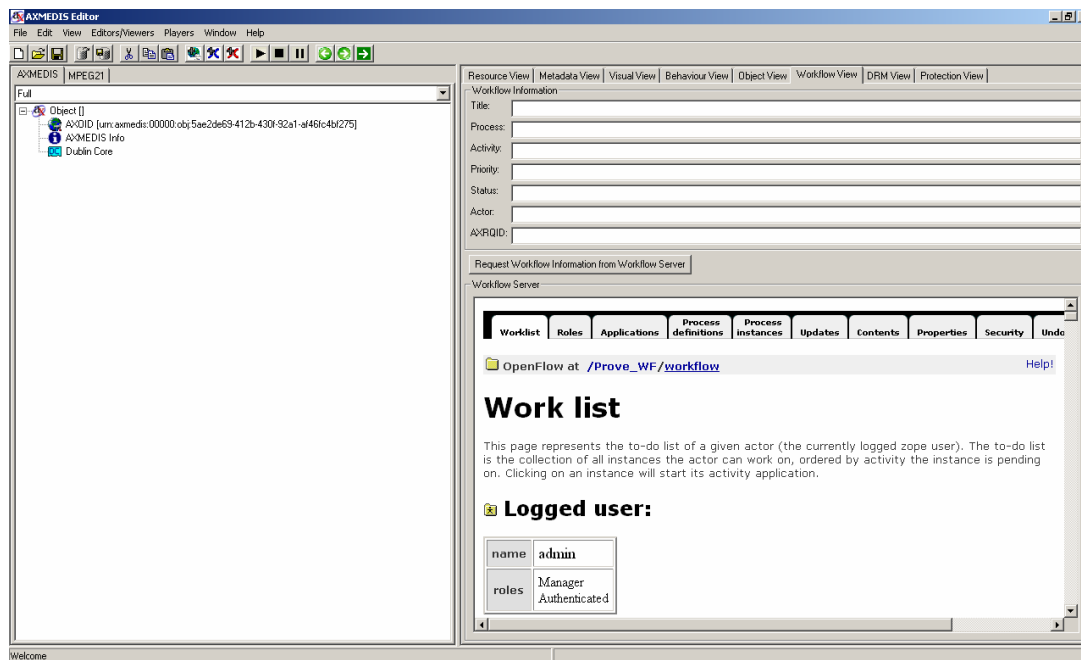


Figure 1: Workflow viewer

2.11 AXMEDIS Protection Information Editor (FHGIGD)

2.11.1 Main functionalities

The Protection Information Editor and Viewer provides the functionalities to view and edit protection information.

The main features are:

- The user can browse the protection information, the list of protection operations that were applied to the selected part of an AXMEDIS object.
- The user can view detailed information about a specific protection operation including all parameters and the protection target.
- The user can alter the order of different protection operations.
- The user can delete one of the protection operations from the list of protection steps.
- The user can select one of the available tools for protection, e.g. encryption, scrambling or compression, and add an additional protection operation to a specific part of an AXMEDIS object.

2.11.2 Relationship with other tools

The Protection Information Editor and Viewer uses the Protection Processor to access protection information, to apply protection operations to a specific part of an AXMEDIS object.

2.11.3 Detailed description of the functionalities and Screenshots

Here's an example on how to use the AXMEDIS Protection Information Editor and Viewer. A part of an object is selected in the tree view of the object shown on the left side of the editor. For editing and viewing of protection information the tab "Protection Information" must be selected on the right side of the editor window.

The current user interface for the protection of resources and the viewing of protection information is shown in the following picture.

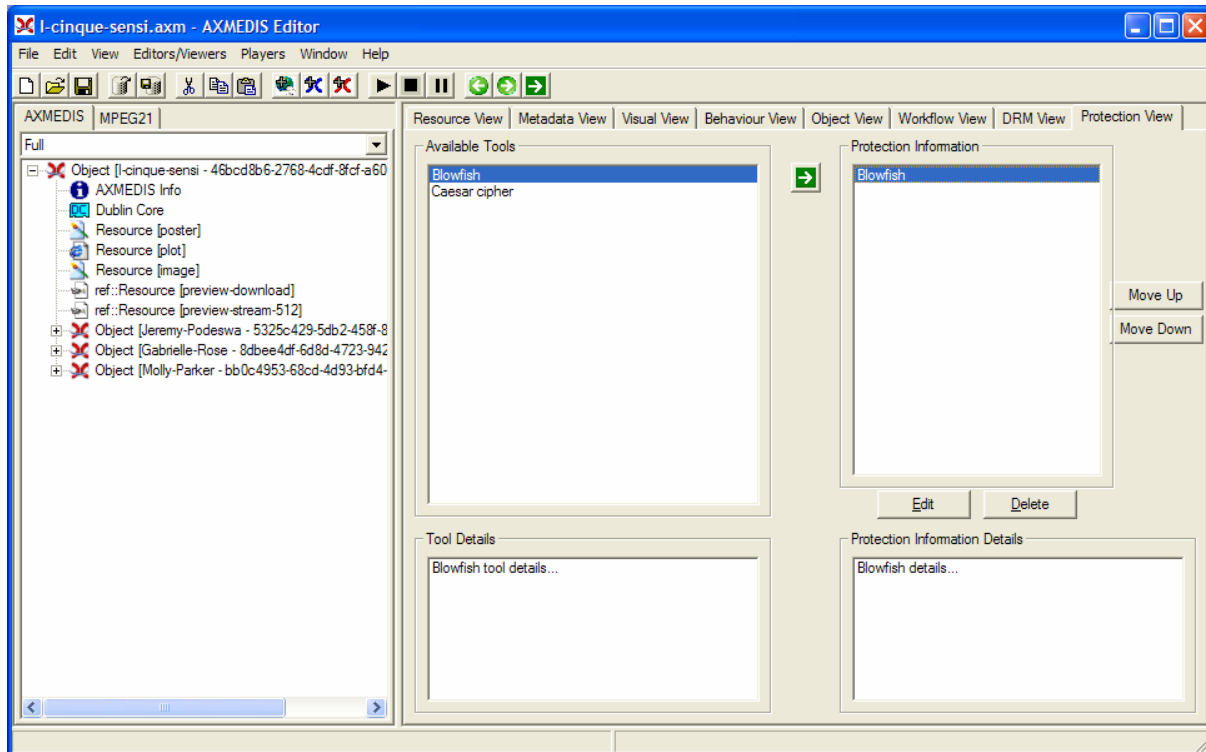


Figure. Protection Information Editor and Viewer user interface

2.11.3.1 Viewing of Protection Information

After the user selected a specific part of an object and switched to the protection view (as described above) a window appears which shows on the right side a list of the protections steps that were applied to this part of the object.

When the user selects one of these operations more detailed information is shown in the lower right part of the window ("Protection Information Details"). The user can open an additional window by double clicking one of the protection operations. This window displays the name of the tool that was applied, a list of the different parameters and the target to which this protection operation was applied to (see figure below).

Protection Operation

Tool Name
[Tool Description]

Parameters

[Parameter 1]	
[Parameter 2]	
[Parameter 3]	
[Parameter 4]	

Target
[Description of Resource]

OK Cancel

Figure: Parameter setting for a specific Protection Operation

2.11.3.2 Editing of Protection Information

When a user wants to change the protection information he has to select a part of an AXMEDIS and view the protection information as described above. The order of the different protection operations can be changed by selecting a specific operation and clicking on the up or down button on the right side.

If a user want so add an additional protection step he can select a protection tool from the list on the left side of the protection window and click on the green arrow in the middle to add it to the list of protection steps. A new window appears in which the specific parameters for this new protection operation, e.g. key length.

A user can also edit existing protection operations by double clicking on an operation in the protection view window. A window appears that displays the different parameters for this protection operation which can be edited and saved by the user.

3 AXMEDIS PC Player (DSI)

3.1 Main functionalities

The AXMEDIS Player allows to:

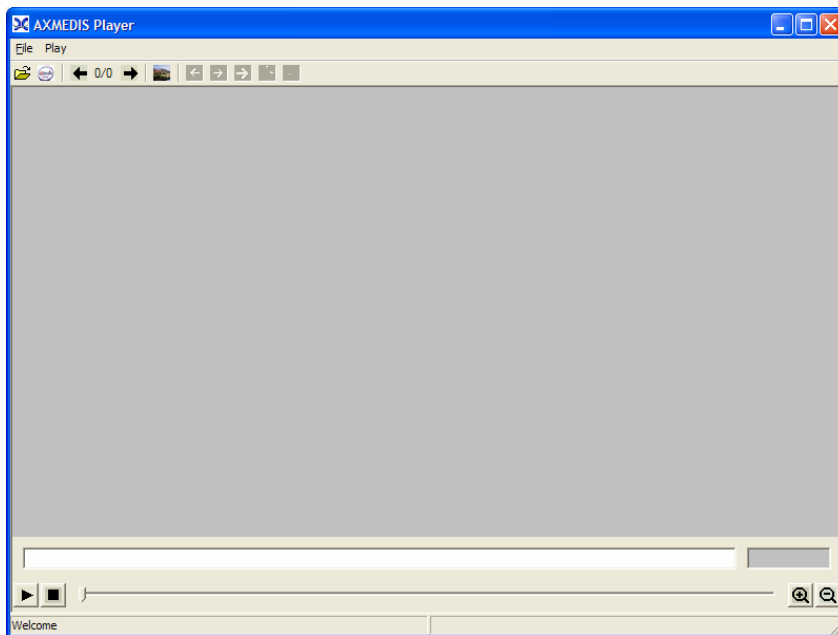
- view the resources present inside the object
- view the metadata of the object
- view the SMIL presentation built
- view the AXMEDIS structure

3.2 Relationship with other tools

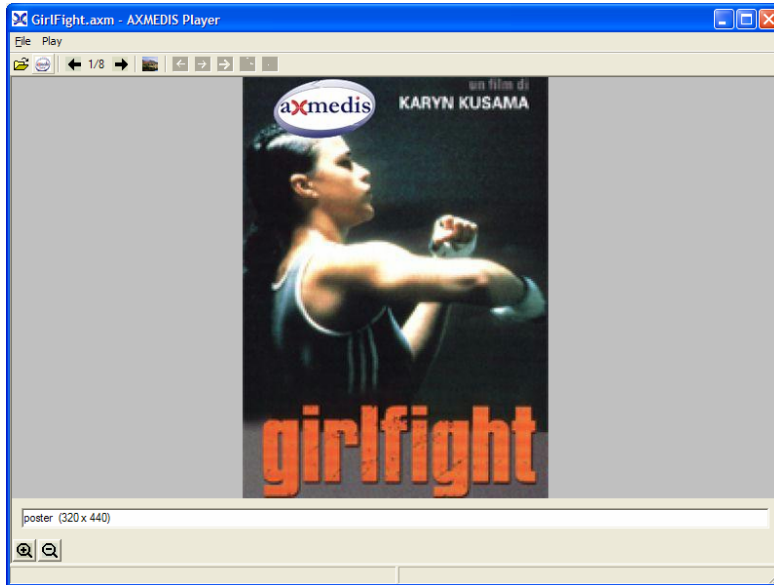
None

3.3 Detailed description of the functionalities and Screenshots

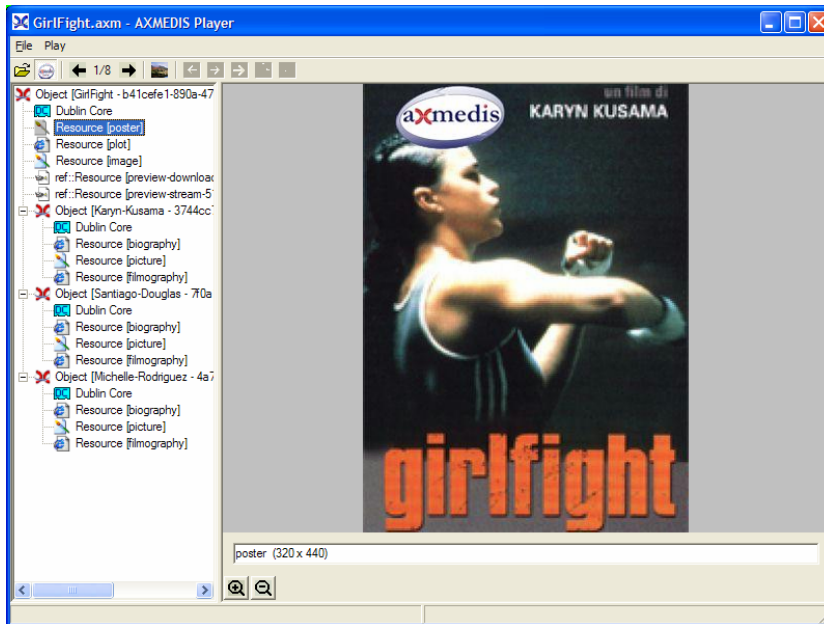
When the player is started an AXMEDIS object may be loaded using the **File/Open...** menu or using the button on the toolbar.

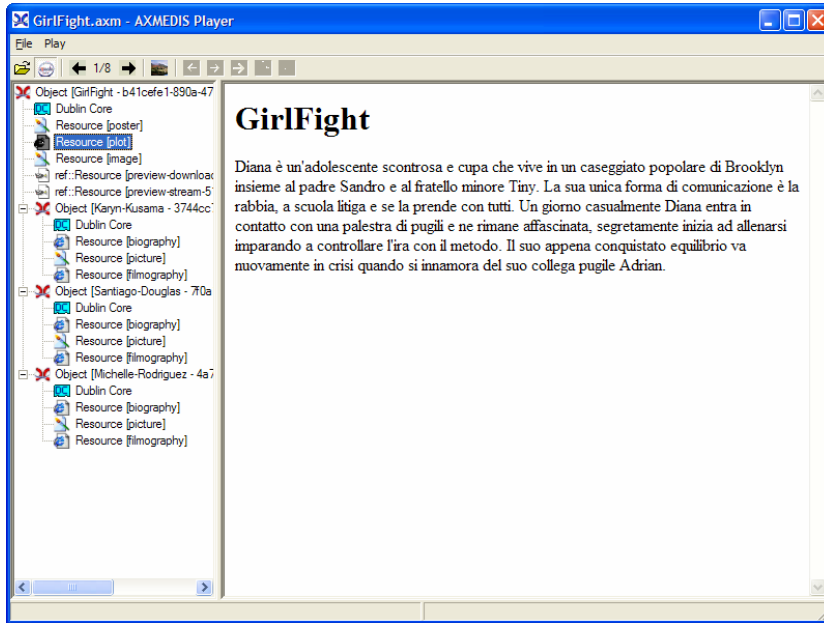


When the file is opened the first resource inside is opened:

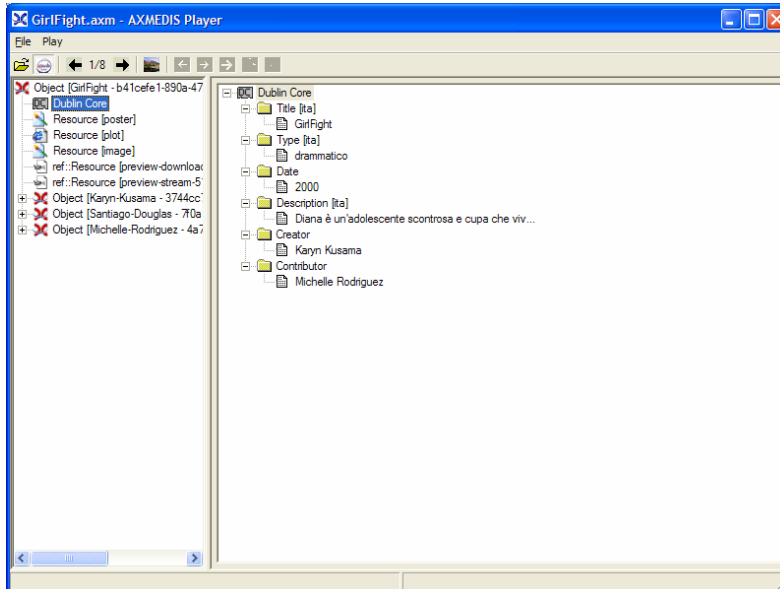


The content of the AXMEDIS object may be browsed using the arrows in the toolbar or using the hierarchy view (opened using the button in toolbar)





The metadata of the object may be view double clicking on the Dublin Core element in the hierarchy view:



4 AXMEDIS PDA Player (TISCALI, DSI)

NOT INCLUDED IN THIS VERSION

4.1 Main functionalities

NOT INCLUDED IN THIS VERSION

4.2 Relationship with other tools

NOT INCLUDED IN THIS VERSION

4.3 Detailed description of the functionalities and Screenshots

NOT INCLUDED IN THIS VERSION

5 AXMEDIS ActiveX Control (DSI)

5.1 Main functionalities

The AXMEDIS ActiveX can be used to visualize and use AXMEDIS Objects inside MS Windows applications or inside web browsers.

The ActiveX exposes a set of functionality allowing to:

- open an AXMEDIS object from file system or from an URL (downloading)
- access to some information on the AXMEDIS Object, (e.g. content count, content mimetype)
- visualize the content that is present inside the AXMEDIS Object.
- control the visualization of the content (play, pause, stop, etc.)
- hide/show the hierarchy view of the object
- visualize the Dublin Core Metadata of the whole object
- show the licence available for the user on the object (not yet available)

5.2 Relationship with other tools

None

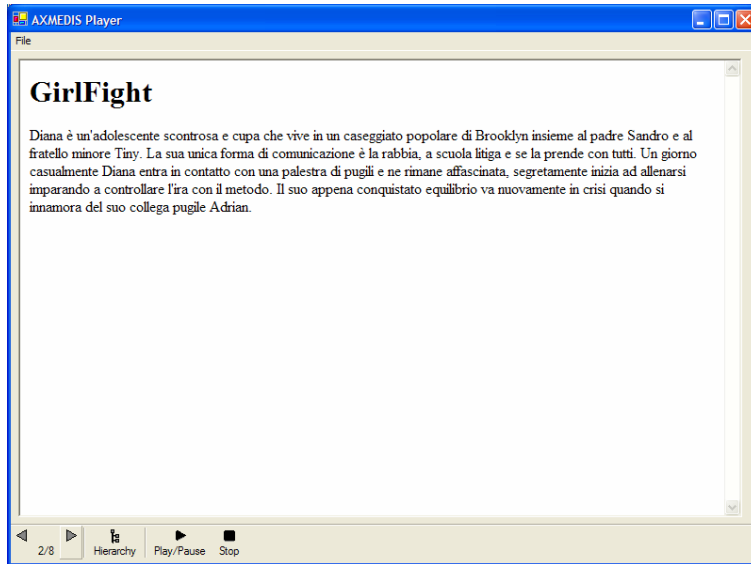
5.3 Detailed description of the functionalities and Screenshots

The following pictures show the use of the AXMEDIS ActiveX inside a Visual Basic Application implementing a basic player. The ActiveX user interface does not present any button controls to allow the customization of the user interface, the visual area is used to display the resources.

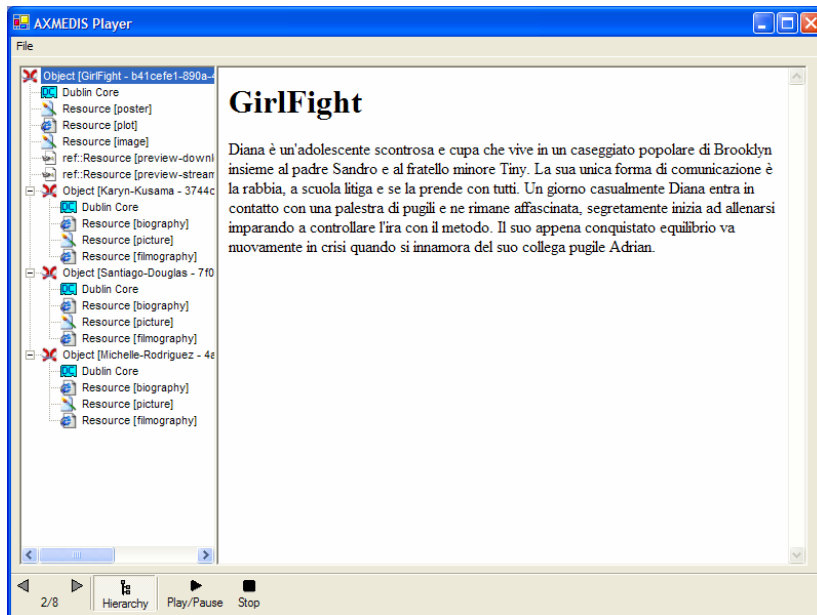


The Left and Right arrow buttons (defined in the VB application) use the *OpenContent(n)* method of the ActiveX to open a specific content stored in the AXMEDIS object, the Hierarchy button use the *ViewHierarchy* property to view/hide the AXMEDIS Hierarchy, the Play/Stop button use the *Play/Stop* methods to control the execution of Audio/Video resources.

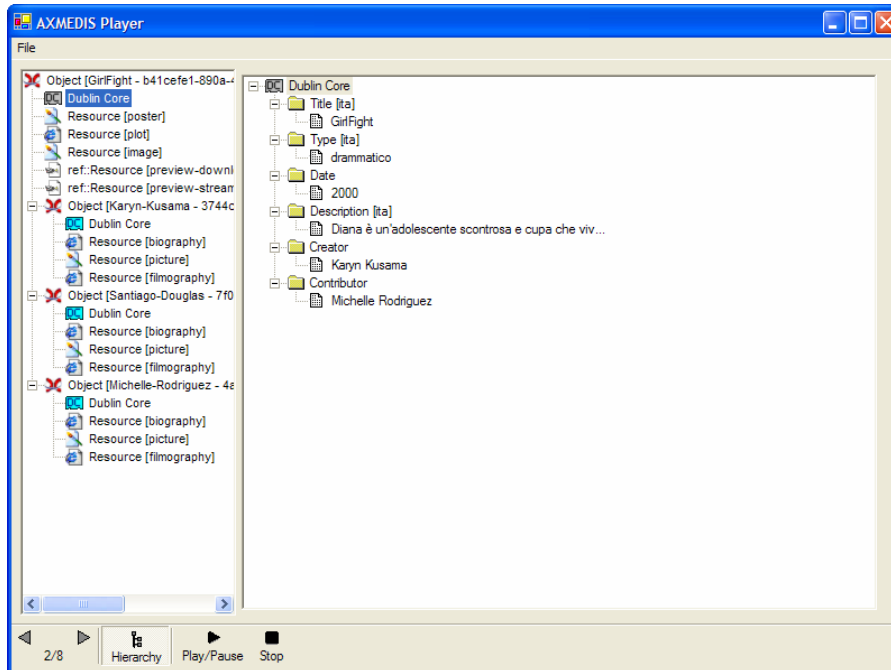
For example clicking on the Right arrow the second resource is shown (the movie plot):



and clicking on the Hierarchy Button the hierarchy is shown:



Double clicking on the Dublin Core element the Metadata is shown:



5.3.1 AXMEDISViewerCtrl

5.3.1.1 Basic Interface

Method Load	
<i>short Load(string url)</i>	
Description	Loads and AXMEDIS object from an URL or from file system, it tries to open the first SMIL resource available or otherwise the first Resource/AXObject available
Input parameters	<ul style="list-style-type: none"> <i>string url</i> – the url or file to be opened (e.g. "http://AXMEDIS.org/demo/demo1.axm", "c:\AXMEDIS-objects\test.axm"), protocols supported will be http and ftp,
Return value	returns 0 if the object can be successfully loaded, 1 if the object cannot be loaded, 2 if the content cannot be opened (missing license)
Remarks	In case of load failure an empty object is present
Property Src (read write)	
<i>string Src</i>	
Description	the URL of the object loaded, when set the object is loaded (not opening content if the object is protected)
DefaultValue	""
Property ContentCount (read only)	
<i>short ContentCount</i>	
Description	number of content elements (resources/axobjects) at the first level
DefaultValue	0
Property BackgroundColor (read write)	
<i>string BackgroundColor</i>	
Description	the color to be used as background (e.g. "ffffff" for white)
DefaultValue	"ffffff"
Property ViewHierarchy (read write)	
<i>bool ViewHierarchy</i>	

Description	indicates if the AXMEDIS Hierarchy view has to be shown or not.
DefaultValue	false
Method GetContentType	
<i>string GetContentType(short contentPos)</i>	
Description	allows to get which kind of content is present at a specific position (1..ContentCount)
Input parameters	<ul style="list-style-type: none"> <i>short contentPos</i> – position of the content to be analyzed
Return value	returns "RESOURCE" or "AXOBJECT"
Remarks	In case of invalid position returns ""
Method GetContentMimeType	
<i>string GetContentMimeType(short contentPos)</i>	
Description	allows to get the mimetype of content at a specific position (1..ContentCount)
Input parameters	<ul style="list-style-type: none"> <i>short contentPos</i> – position of the content to be analyzed (1..ContentCount)
Return value	returns a mimetype for a resource and the mimetype of the resource really played in case of an embedded AXMEDIS object (e.g. it can return "application/smil" if the object contains a SMIL resource)
Remarks	In case of invalid position returns ""
Method OpenContent	
<i>void OpenContent(short contentPos)</i>	
Description	allows to open a content at a specific position (0..ContentCount), position 0 means the object itself. If the content to be opened is an AXMEDIS object it opens the first SMIL resource present otherwise it opens the first resource present
Input parameters	<ul style="list-style-type: none"> <i>short contentPos</i> – position of the content to be opened (0..ContentCount)
Return value	none
Remarks	none
Method ShowMetadata	
<i>void ShowMetadata()</i>	
Description	allows to show the metadata of the whole object (using Dublin Core information)
Input parameters	none
Return value	none
Remarks	none
Method ShowLicense	
<i>void ShowLicense()</i>	
Description	allows to show the license available on the object
Input parameters	none
Return value	none
Remarks	none
Property NeedLicense (read only)	
<i>bool NeedLicense</i>	
Description	Indicates if a license is needed to view the object
DefaultValue	false
Property HaveLicense (read only)	
<i>bool HaveLicense</i>	
Description	Indicates if is available a license to view the object
DefaultValue	false
Method AcquireLicense	
<i>void AcquireLicense(string licenseID)</i>	
Description	allows to acquire locally a license given the license ID (if it is allowed)

Input parameters	<ul style="list-style-type: none"> <i>string licenseID</i> – contains the license identification code
Return value	none
Remarks	The method is used to preload in the local cache the license allowing to use it offline even using the AXMEDIS Player. The license could be even for another AXMEDIS Object not for the one shown in the ActiveX (which can be a preview version)

5.3.1.2 Visual Control Interface

Property SupportVisualControl (read only)	
<i>bool SupportVisualControl</i>	
Description	indicates if for the currently opened content the visual control methods can be used
DefaultValue	false
Property Zoom (read write)	
<i>double Zoom</i>	
Description	is the zoom factor to be applied (1 = 100%)
DefaultValue	false
Remark	if the zoom factor is changed the AutoFit property is set to false
Property AutoFit (read write)	
<i>bool AutoFit</i>	
Description	when true indicates to resize the visual content to be fitted inside the window
DefaultValue	true
Remark	none
Method Fit	
<i>void Fit(double width, double height)</i>	
Description	fits the zoom factor to display the image at the desired size (keeping aspect ratio)
Input parameters	<ul style="list-style-type: none"> <i>double width, height</i> - the image size in pixels
Return value	none
Remarks	the AutoFit property is set to false
Method FitToWindow	
<i>void FitToWindow()</i>	
Description	fits the visual content to the size of the image
Input parameters	none
Return value	none
Remarks	the AutoFit property is maintained
Method ZoomIn	
<i>void ZoomIn(double perc)</i>	
Description	Increments the zoom factor of the given percentage
Input parameters	<ul style="list-style-type: none"> <i>double perc</i> – the percentage used to increment the zoom factor (e.g. 10 for 10%)
Return value	none
Remarks	the AutoFit property is set to false
Method ZoomOut	
<i>void ZoomOut(double perc)</i>	
Description	Decrements the zoom factor of the given percentage
Input parameters	<ul style="list-style-type: none"> <i>double perc</i> – the percentage used to decrement the zoom factor (e.g. 10 for 10%)
Return value	none
Remarks	the AutoFit property is set to false
Property FullScreen (read write)	

<i>bool FullScreen</i>	
Description	when true switch to full screen mode
DefaultValue	false
Remark	none
Method Print	
<i>void Print()</i>	
Description	Prints the visual content
Input parameters	none
Return value	none
Remarks	none

5.3.1.3 Time Control Interface

Property SupportTimeControl (read only)	
<i>bool SupportTimeControl</i>	
Description	indicates if for the currently opened content the time control methods can be used
DefaultValue	false
Method Play	
<i>void Play()</i>	
Description	Starts playing the content or continue in case of pause
Input parameters	none
Return value	none
Remarks	has no effect if it is already playing
Method Pause	
<i>void Pause()</i>	
Description	Pause the execution or continue execution
Input parameters	none
Return value	none
Remarks	none
Method Stop	
<i>void Stop()</i>	
Description	Stops execution
Input parameters	none
Return value	none
Remarks	none
Method JumpToTime	
<i>void JumpToTime(ulong time)</i>	
Description	Set the execution time in milliseconds
Input parameters	<ul style="list-style-type: none"> <i>ulong time</i> – the time at which jump the execution
Return value	none
Remarks	none
Property IsPlaying (read only)	
<i>bool IsPlaying</i>	
Description	indicates if the content is being played
DefaultValue	false
Property Duration (read only)	
<i>ulong Duration</i>	

Description	indicates duration in milliseconds
DefaultValue	0
Property CurrentTime (read only)	
<i>ulong CurrentTime</i>	
Description	indicates current execution time in milliseconds
DefaultValue	0
Property StartTime (read write)	
<i>ulong StartTime</i>	
Description	indicates the time in milliseconds where to start
DefaultValue	0
Property EndTime (read write)	
<i>ulong EndTime</i>	
Description	indicates the time in milliseconds where to end execution
DefaultValue	Duration

5.3.1.4 Generic Command Interface

This interface allows to execute commands giving a command identifier. The command identifier are:

Command ID	Value
"Load"	string with the URL
"Src"	string
"OpenContent"	content position
"ContentCount"	
"BackgroundColor"	string
"ShowMetadata"	none
"ShowLicense"	none
"NeedLicense"	bool
"HaveLicense"	bool
"Zoom"	double
"ZoomIn"	double
"ZoomOut"	double
"AutoFit"	
"Fit"	string "w,h" (e.g. "100, 400")
"FullScreen"	bool
"Print"	none
"Play"	none
"Stop"	none
"Pause"	none
"IsPlaying"	bool
"CurrentTime"	ulong (read only)
"Duration"	ulong (read only)
"StartTime"	ulong
"EndTime"	ulong

Method execCommand	
<i>HRESULT execCommand(string cmdID, bool showUI, variant value, bool* ret)</i>	
Description	executes the command described in cmdID, with an optional argument contained in value
Input parameters	<ul style="list-style-type: none"> • <i>string cmdID</i> – command to be done • <i>bool showUI</i> – indicates if UI update is needed • <i>variant value</i> – the argument for the command

	<ul style="list-style-type: none"> <i>bool* ret</i> – contains if the command has been executed or not
Return value	S_OK if ok
Remarks	none
Method queryCommandEnabled	
<i>HRESULT queryCommandEnabled(string cmdID, bool* ret)</i>	
Description	looks if the specified command can be successfully executed in the current context
Input parameters	<ul style="list-style-type: none"> <i>string cmdID</i> – command to be checked <i>bool* ret</i> – contains the result (true if the command is enabled)
Return value	S_OK if ok
Remarks	none
Method queryCommandIndeterm	
<i>HRESULT queryCommandIndeterm(string cmdID, bool* ret)</i>	
Description	looks if the specified command is in the indeterminate state
Input parameters	<ul style="list-style-type: none"> <i>string cmdID</i> – command to be checked <i>bool* ret</i> – contains the result (true if the command is in the indeterminate state)
Return value	S_OK if ok
Remarks	none
Method queryCommandStatus	
<i>HRESULT queryCommandStatus(string cmdID, bool* ret)</i>	
Description	indicates if the specified command has been executed on the object or not
Input parameters	<ul style="list-style-type: none"> <i>string cmdID</i> – command to be checked <i>bool* ret</i> – contains the result
Return value	S_OK if ok
Remarks	none
Method queryCommandSupported	
<i>HRESULT queryCommandSupported(string cmdID, bool* ret)</i>	
Description	looks if the specified command is supported
Input parameters	<ul style="list-style-type: none"> <i>string cmdID</i> – command to be checked <i>bool* ret</i> – contains the result (true if the command is supported)
Return value	S_OK if ok
Remarks	none
Method queryCommandValue	
<i>HRESULT queryCommandValue(string cmdID, variant* ret)</i>	
Description	returns the current value for a command
Input parameters	<ul style="list-style-type: none"> <i>string cmdID</i> – command <i>variant* value</i> – contains value
Return value	S_OK if ok
Remarks	none

5.3.1.5 Examples of usage

The following is an example of use of the activex to make a preview of an AXMEDIS object:

```

activex=new AXMEDISViewerCtrl();
activex.BackColor="#ffffff";

//open the object downloading it from an URL
activex.Load(http://AXMEDIS.org/demo/metropolis.axm);

//show the metadata of the whole object
activex.ShowMetadata();

//look for a license and if not present acquire it
if(activex.NeedsLicense && !activex.HaveLicense)
    activex.AcquireLicense(licenseID);
else
    activex.ShowLicense();
AXMEDIS

```

```
// open all content elements inside the object and make a preview for 10s each
for(int i=1; i<=activex.ContentCount; i++)
{
    activex.OpenContent(i);

    //if currently opened content is image/video/document fit it to 100x100 preview
    if(activex.SupportVisualControl)
        activex.fit(100,100);

    //if currently opened content is audio/video preview it for 10s other wise wait for 10s
    if(activex.SupportTimeControl)
        activex.EndTime=10000; //10s preview
    else
        wait(10000); //wait 10s
}
```

6 AXMEDIS DRM Editor (FUPF)

6.1 Main functionalities

“*DRM Editor and Viewer*” is a software application to view and eventually edit MPEG-21 REL Licenses. This user manual refers to its version 2.1.9. “*DRM Editor and Viewer*” and comes in the form of two different applications. They are two different executable files, although they come with the same `setup.exe` installation file, which in turn will ask whether one or both programs are to be installed.

Its main features are:

Feature	DRM editor	DRM viewer
Load a license from a XML file	Yes	Yes
Load license from a remote host (PMS Server)	Yes	Yes
Create a new license form the scratch	Yes	No
Display graphically the license	Yes	Yes
Store a license in a XML file	Yes	No
Store a license in a remote host (PMS Server)	Yes	No
Edit (modify) a License	Yes	No

6.2 Relationship with other tools

This tool can be embedded as a component in other tools (i.e. `axeditor`).


The licenses created with the AXMEDIS DRM Editor can be stored locally or in a remote server (AXMEDIS PMS Server). In order to be able to store licenses in the remote server, a connection to it must be available, as explained in next chapter.

6.3 Detailed description of the functionalities and Screenshots

The application is a window application. (see Figure 1). As in a common window application, the following elements appear: menu bar, button bar, status bar. The status bar shows an icon informing whether connection to the server is available or not. If it does not exist, only file operations are possible



The main window shows two areas: in the left a hierarchically structure is displayed (also called tree), which is used to navigate through the license elements; in the right side there is a panel showing information related to the element marked in the tree. Functionalities can be accessed through the menu options, the buttons in the button bar, or the buttons that appear in the panel. Next we summarise the DRM Editor and Viewer main functionalities:

- DRM Editor and Viewer. Open a license from a file  (see Figure 1). The panel displays the file path or an alternative description.

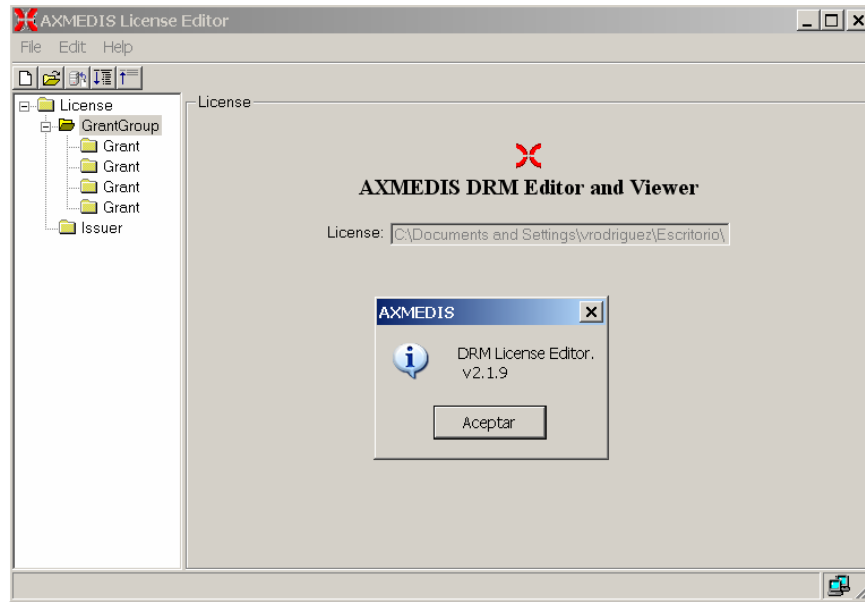



Figure 1: DRM Editor view

- DRM Editor. Create a new license . When a new license is created, it is empty. The tree appears with an “issuer” element and with a “Grant Group” element, but they are initially empty.
- DRM Editor. Modify a license.
 - When the tree is clicked on its “issuer element”, the Editor shows the issuer panel, where the issuer and the date of issuance can be set (see Figure 2).

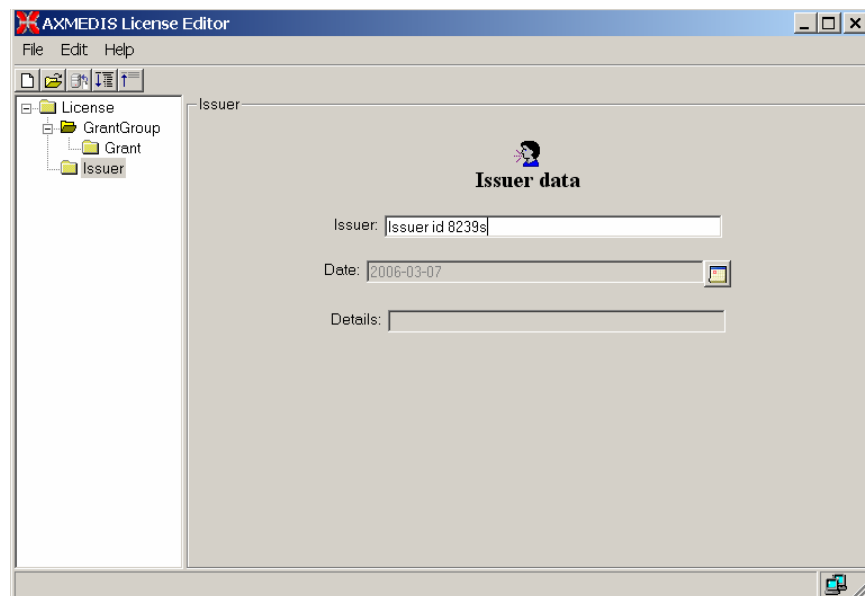




Figure 2: DRM Editor License Issuer view

- When the tree is clicked on its “GrantGroup element”, a Grant Group panel is displayed. (see Figure 3). From this panel it is possible to add  or delete  grants. The first icon creates an empty grant, the second deletes it. Once a grant has been created, it will be displayed in the tree.

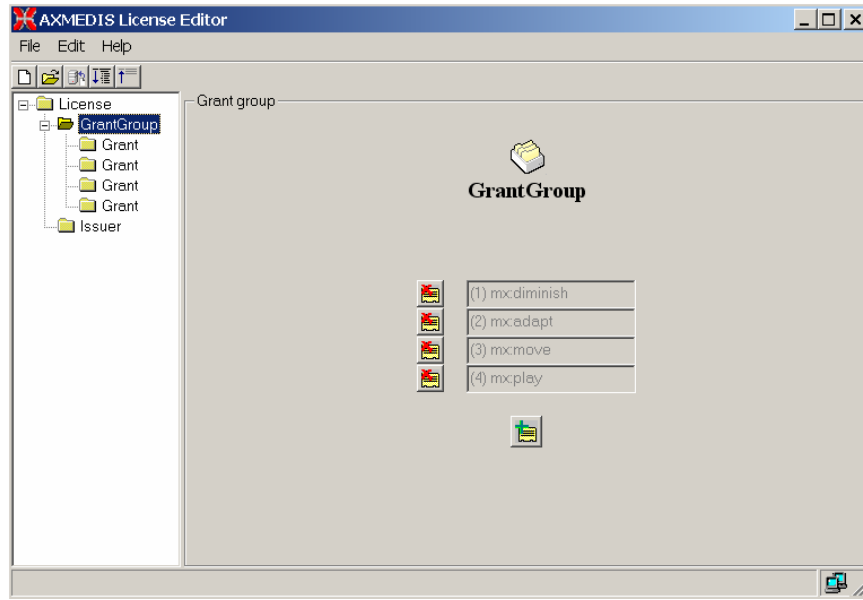




Figure 3: DRM Editor and Viewer License Grant Group view

- When the tree is clicked on its “Grant” element (see Figure 4), the Grant panel is shown. From this panel, the following actions can be performed:
 - To change the following fields: principal, right, resource.
 - To add  and remove  conditions. The conditions that can be set are: *number*, *interval*, *territory* and/or *fee*. Each of them can be modified.

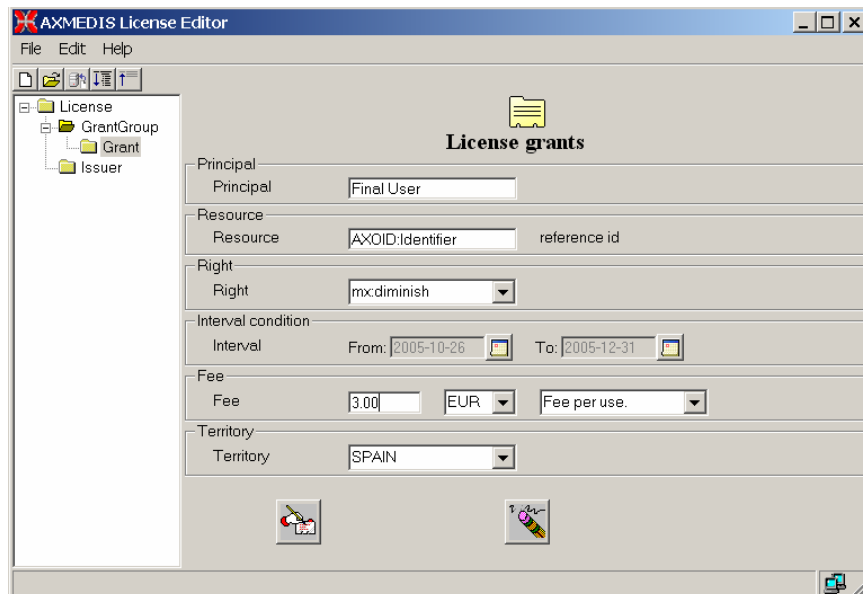



Figure 4: DRM Editor License Grant view

- When the *issue* right is chosen, the license becomes a “distributor license”, and the Grants underneath are Grants to be given by the distributor. This is reflected in the tree, where a new sublevel is displayed containing additional grants.

- DRM Editor. Store a license to a file. This option is performed when selection the “File -> Save to File” option of the Editor.
- DRM Editor. Store a license into a remote server .

Limitations to be fixed in subsequent versions:

- In the current version only up to 8 grants can be added to the same grant group.
- Only a condition of each type can be added to a grant.

If the DRM Editor and Viewer are used in its classic version (the one which displays a simple tree) these limitations are overcome.

7 AXMEDIS DRM Viewer (FUPF)

7.1 Main functionalities

Refer to section “4 DRM Editor” for the viewing functionalities of “DRM Editor and Viewer”.

7.2 Relationship with other tools

Refer to section “4 DRM Editor” for the viewing functionalities of “DRM Editor and Viewer”.

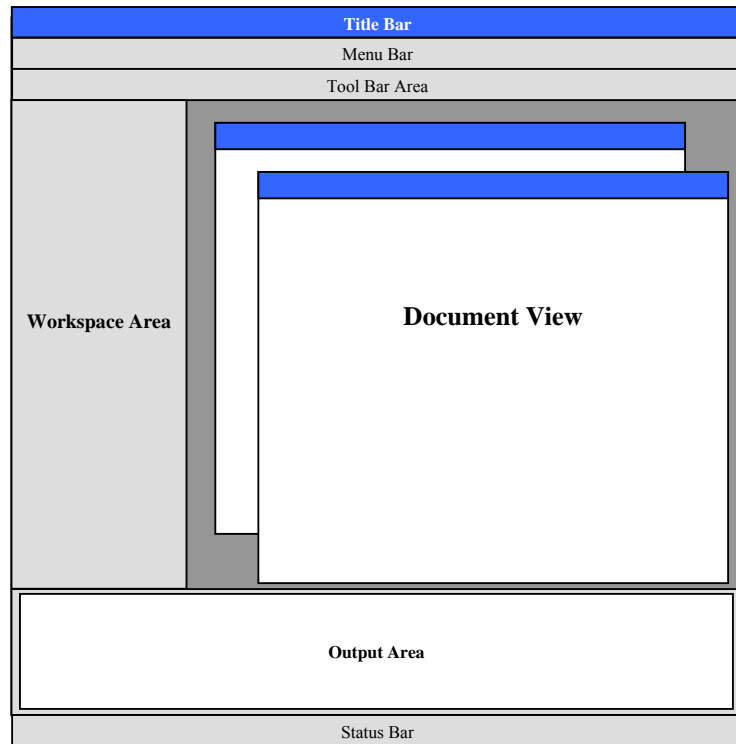
7.3 Detailed description of the functionalities and Screenshots

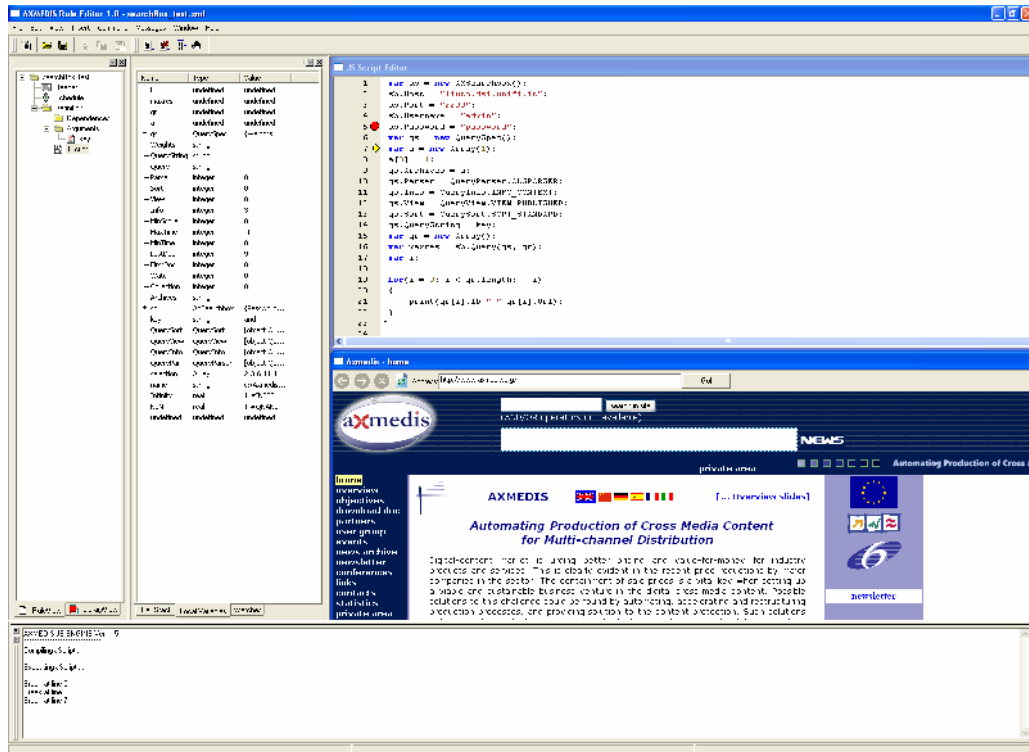
Refer to section “4 DRM Editor” for the viewing functionalities of “DRM Editor and Viewer”.

8 AXMEDIS Rule Editor (DSI)

8.1 Main functionalities

The AXCP Rule Editor GUI is a MDI window that manages a AXCP rule document. It provides a set of tools and views to help the user during the editing and building of rule. It hosts an instance of the AXCP rule executor in order to provide functionalities for debugging, testing and validating the script code associated with a rule. To help the user in writing rule, the editor is equipped with an Help on line and area where the user can access to a library of script functions. The GUI is structured as:





Main view of the AXMEDIS Rule Editor GUI

8.1.1 The Menu Bar

The menu bar is constituted of the following entries:

File

- **New** – create a new rule document
- **Open** – Open a AXCP rule in the Rule Editor
- **Close** – Close the current rule document
- **Save** – save the current rule using the current file name
- **Save as** – save the current rule by name
- **Import JScript** – import a script in the rule
- **Export JScript** – export the script on file system
- **Properties** – it shows a report on the
- **Page setup** – allow preparing the page for printing document
- **Print preview** – open the print preview dialog
- **Print** – send the document to the printer
- **Recent Files** – History of files
- **Exit** – Quit the editor

Edit

- **Copy** – copy a selection in the clipboard
- **Paste** – paste a selection available in the clipboard
- **Cut** – delete and copy a selection in the clipboard
- **Delete** – delete a selection
- **Find...** – Search a word in the text
- **Find next** – search again for a new location of the current text
- **Replace...** – replace a word with another
- **Replace again** – replace again the word with a new entry
- **Match brace** – match the brace

- **Go to...** – go to a specific line text
- **Advanced menu:**
 - **Indent increase**
 - **Indent reduce**
 - **Overwrite mode**
 - **Wrap mode**
 - **Show line endings**
 - **Show indent guides**
 - **Show line numbers**
 - **Show long line markers**
 - **Show whitespace**
- **Select All** – select all content
- **Select line** – select the line where cursor is blinking

View

- **Workspace** – It opens the Workspace area
- **Output** – It opens the Output area

Insert

- **Selection** – Adds a selection item in the rule
- **Parameter** – Adds a parameter item in the rule
- **Script** - Adds a script item in the rule
- **Dependency** - Adds a dependency item in the rule

Commands

- **Activate Rule** – It is the activate rule command and will allow sending the current rule to the scheduler and the notification to the AXMEDIS Workflow Manager. A connection with the Rule Engine Scheduler will be open in order to perform the installation of rule in the Scheduler.
- **Check rule** – Tests the feasibility of the rule (like a compiler plus some tests on AXMEDIS objects and estimation of some parameters such as the files complexity and required workload)
- **Rules List...** – Shows the list of rules inside the repository of the Rule Editor
- **Find Rule...** – Allows making queries to the rules repository of the Rule Editor
- **Debug**
 - **Go** – Enter in the debug mode or if the script is stopped, continue execution until the script is finished, or a breakpoint is reached.
 - **Stop** – Stop the script execution and close the debug mode
 - **Step Over** - Executes the current line of the script, then pauses. This differs from the "Trace" command in that it will not step into functions and scripts that are called by the current line.
 - **Trace Into** – Executes the current line of the script, then pauses. This differs from the "Step" command in that if the current line calls a function, or another script, the debugger will trace into the called function or script.
- **Breakpoint:**
 - **Insert/Remove** – Set a breakpoint on the currently selected line of the script code. Every time the selected line is reached, the debugger will stop. Clear a breakpoint from the currently selected line of the script code.
 - **View list** – Open the debug window showing all breakpoints in the script code.

Messages

- **Last message** – Displays the last message sent by the AXMEDIS Workflow Manager
- **Messages List** - Displays the list of messages sent by the AXMEDIS Workflow Manager
- **Notify activity completion** – it open the dialog for notifying the completion of the activity to the AXMEDIS Workflow Manager

Window (provided automatically by the MDI GUI)

- **Cascade**
- **Tile Horizontal**
- **Tile Vertical**
- **Next** – Activate the next document view
- **Previous** - Activate the previous document view
- **Arrange Icons** – Arrange the all minimised document views
- **Close All** – Close all document views
- **Windows list**

Help

- **About** – Information about the authors, version, etc

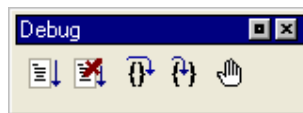
8.1.2 ToolBar Area

The toolbar area will host a set of icon buttons that will allow calling functions without accessing to the menu bar. The toolbar area will be based on dockable toolbars and will allow the dynamic customisation by adding or removing sub-toolbars. For this end the editor will provide sub-toolbars for:

- *Standard* – it will provide main functionalities for managing rule files and editing (new, open, save, etc...) for:
 - *New rule document*
 - *Open from disk*
 - *Save*
 - *Cut*
 - *Copy*
 - *Paste*



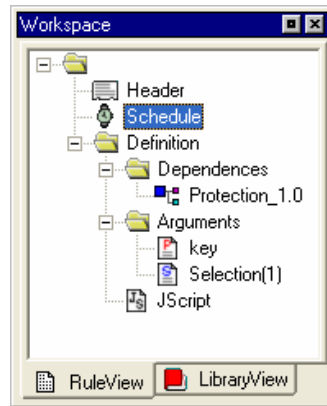
- *Debug* – it will provide main controls for:
 - *Start Debug*
 - *Stop Debug*
 - *Step Over*
 - *Trace Into*
 - *Add/Remove Breakpoint*



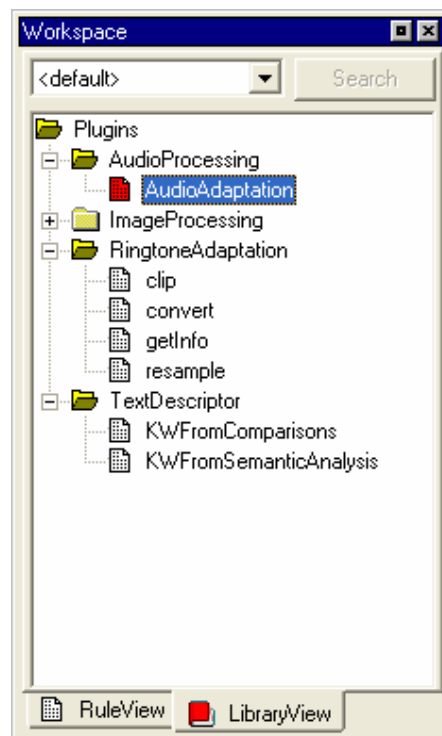
8.1.3 Workspace Area

It is a resizable docking panel and includes a notebook control where the rule view and the library view are shown.

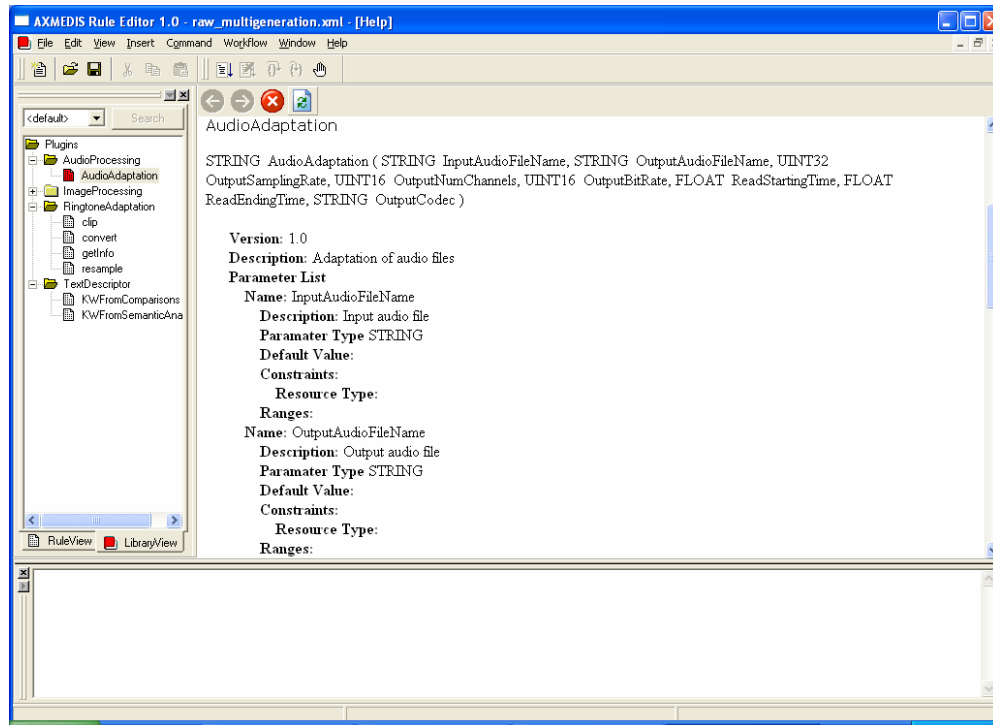
Rule Tree View - Such view is a tree view that display the structure of the rule. A dynamic popup menu is available for a quick access to functions that allows the quick management of items (edit and view metadata, delete, Insert, Cancel, Move up/Down, Rename, Open/Edit, ...). Appropriate icons allow identifying intuitively components of rule in view. In the following picture the actual structure of the Rule View area is depicted:



Library view – It is an on-line book that could be used as help by the user. It displays the set of functionalities provided by the Plugins installed and automatically detected by the editor. It is a tree control that permits to show and browse plugins module and the functionalities that they provide according to their profile.

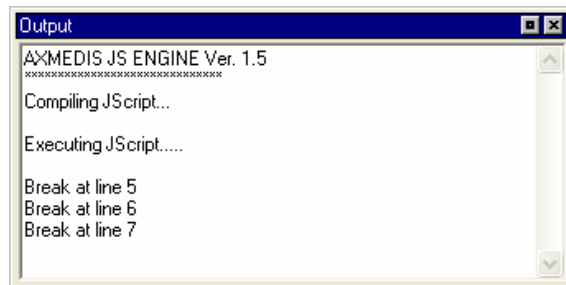


The user can see the documentation associated with each selected function by interacting with each item of the tree by double clicking or accessing to a contextual popup menu. The selected documentation is displayed in the **Text/Html document view**.



8.1.4 Output area

This is a text control where messages, textual description, errors, debugging info, alert, etc... will be displayed.



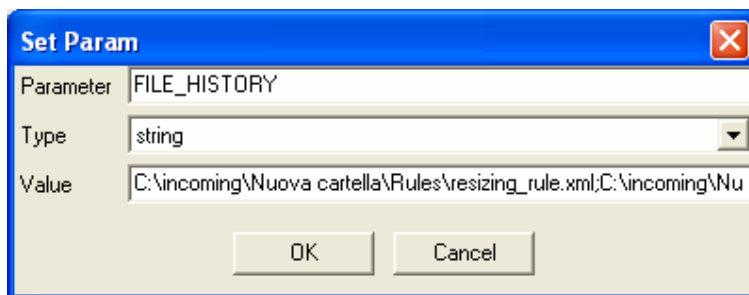
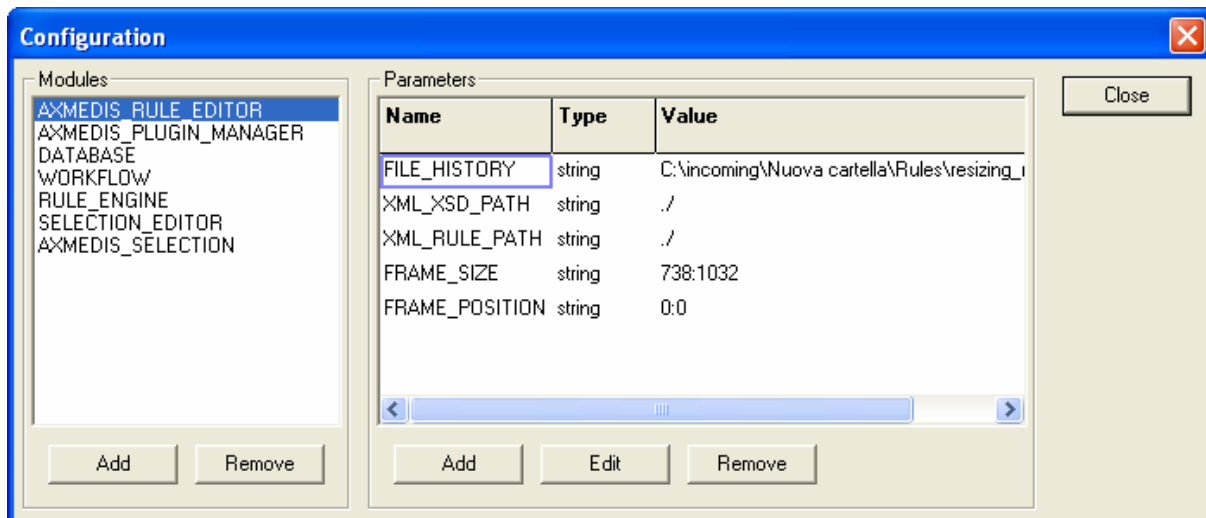
8.1.5 MS Windows Firewall Alert at startup

The picture bellow shows the Windows Security Alert Dialog. To run AXCP Tools, please unlock the application clicking on the "Unlock" button. This operation allows AXCP tools to use network services and run properly.



8.1.6 Configuration

The AXCP Rule Editor allows accessing to the configuration dialog when it is necessary to modify the configuration file.



In the following tables the set of parameters regarding the configuration of the editor are listed. Such parameters are grouped into modules as reported below:

AXMEDIS Rule Editor

Config parameter	Possible values
XML_RULE_PATH	it is the directory where the rule will be saved
XML_XSD_PATH	It is the directory where xml schema (XSD files) are stored
FRAME_SIZE	it is the information about the last width and height of the main frame
FRAME_POSITION	It is the information about the last position (x,y) of the main frame
FILE_HISTORY	It is the list of recent rule documents

Workflow Manager

Config parameter	Possible values
workflowUrl	It is the URL for the workflow plugin
gatewayUrl	It is the gateway URL for the workflow

AXMEDIS Plugin Manager

Config parameter	Possible values
PLUGINS_PATH	It is the directory where the DLL of plug-ins with their profiles (workflow, adaptation, descriptor and fingerprint estimators) are stored.

AXMEDIS Database

Config parameter	Possible values
user	The user name for logging into Database
passwd	The password for logging into Database
LoaderWSEndPoint	It is the URL for the loader
HTTPPath	It the HTTP path
UploadPath	It is the Upload path
SaverWSEndPoint	It is the URL for the saver

AXMEDIS Rule Engine

Config parameter	Possible values
gatewayUrl	it is the URL of the AXCP Rule Scheduler/GRID

AXMEDIS Selection

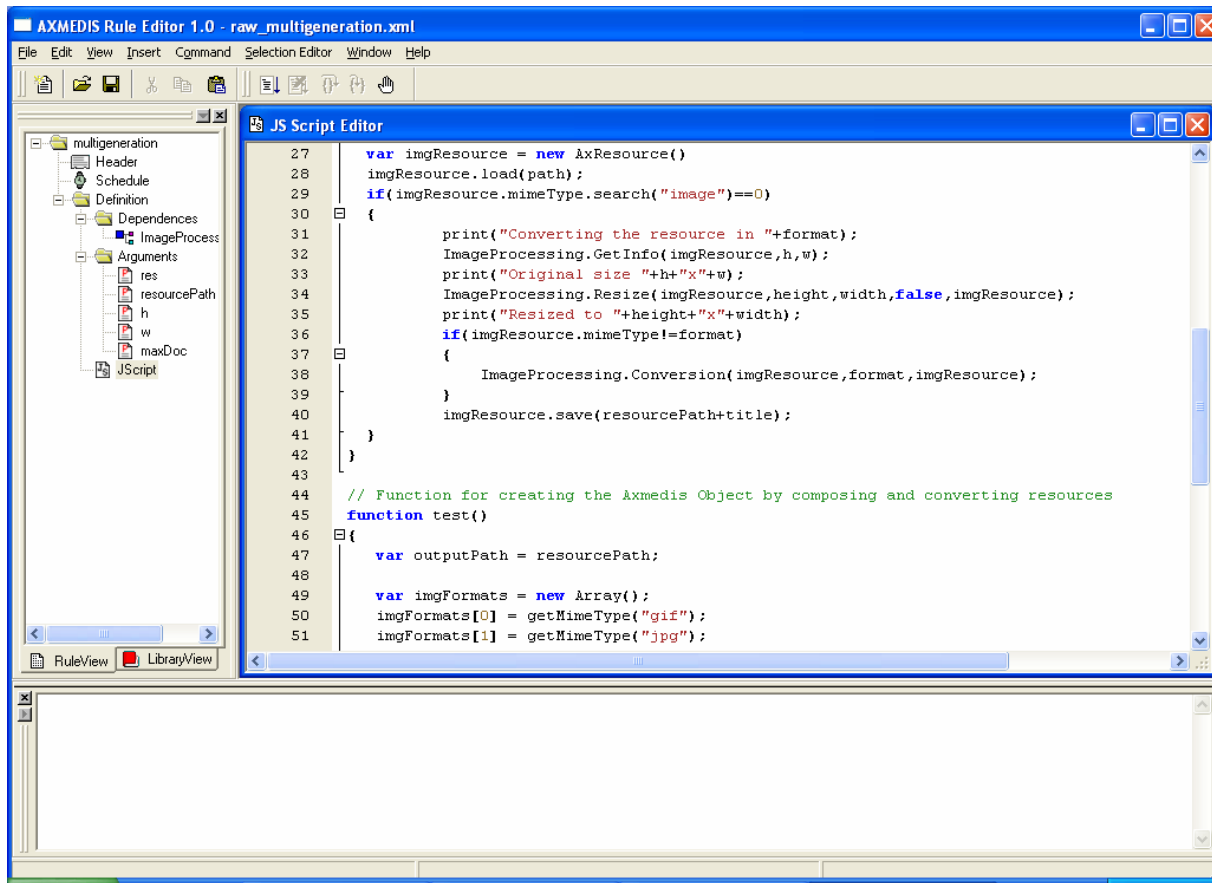
Config parameter	Possible values
MAIN_QUERY_SUPPORT_WSDL	It is the URL of the WSDL for using the Main Query Support
SELECTION_ARCHIVE_WSDL	It is the URL of the WSDL for using the Selection Archive

8.1.7 Tools, Viewers and Editors

Some different types of tools and editor were designed for visualizing and/or editing different type of documents.

1. **Javascript editing window** – This is the window client where the user can use to write the script code. It is a multiline text control where it is possible to edit the script. The textual editor supports some facilities such as:
 - Auto completion of words - a window listing possible completions for strings the user has typed
 - Syntax highlighting – keywords will be coloured
 - Brace highlighting
 - Folding/Hiding - making lines invisible or visible. It shows or hides a range of lines.
 - Multiple views - to have multiple views of the same Document. (Split view)
 - Breakpoint insertion/removal – to control the code in the debugging session

- Visualisation of line numbers



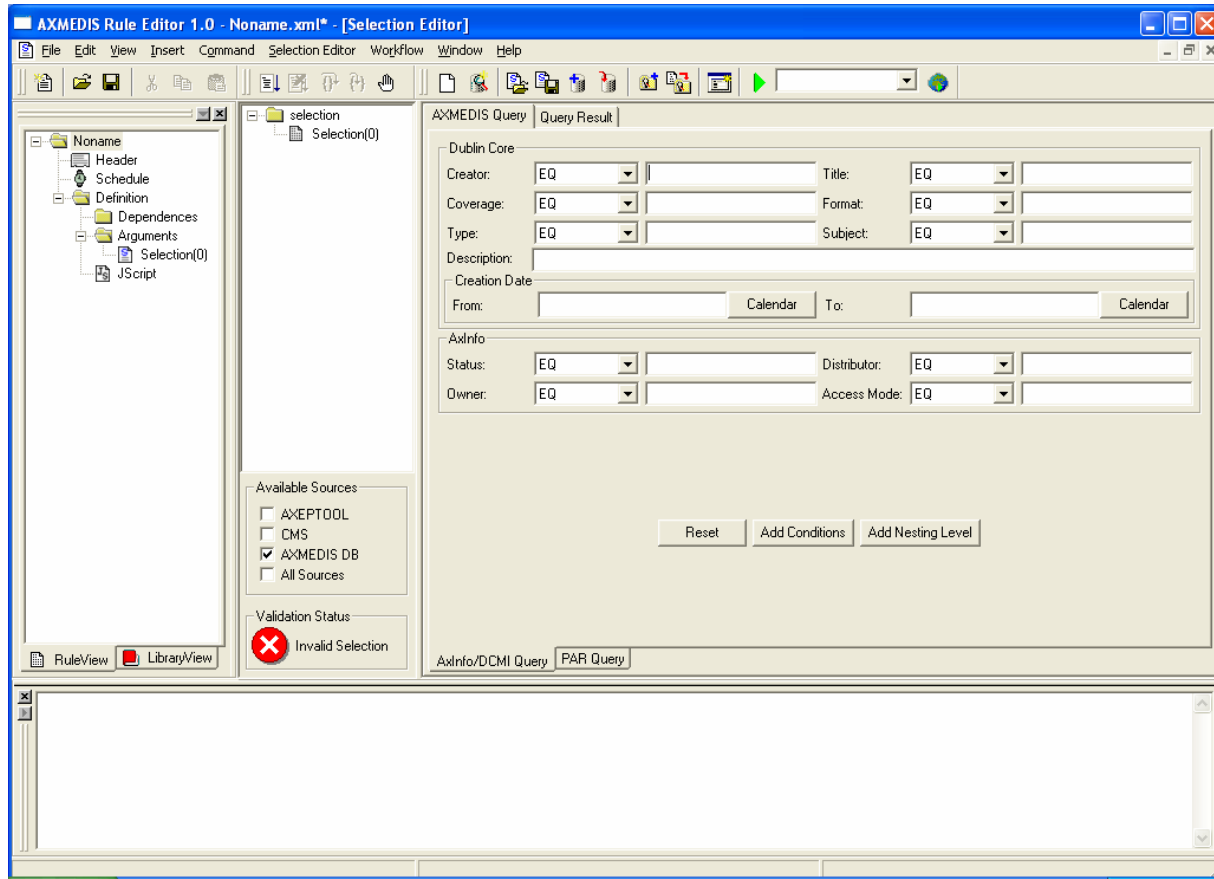
2. **Text/Html document view** – This is the window for the visualisation of the documentation provided by the help on line. It is opened when the user double clicks on a voice of the index in the *Library* view or when the internal help is called. It provides functionalities for browsing TXT or HTML pages. For example, all the information related to the description of a function selected from the *Library* view are shown in such window.
3. **Selection Editor** - It is will be an interactive html page that will be displayed by means the HTML document viewer. It will provide functionality for:
 - a. Edit a selection
 - b. Save/Load a selection
 - c. Actualise the selection

8.1.8 Selection Editor ToolBar

The selection editor provides a quick access to functions by means the following toolbar for:

1. Clear the selection
2. Insert a New Query
3. Open a Selection from disk
4. Save a Selection on disk
5. Import Selection from DB
6. Export Selection into DB
7. Import Query from Disk
8. Export Query to disk
9. Customize Query Panels

10. Run query
11. Selected query
12. Run selection



AXMEDIS Query

Query Result

Results table

Title	Description	Version	Object Id	Source Cha...	St
Il tesoro del santo	In un bordello italiano una ...	1	08180e7b-...	AXDB	
Follie di Jazz (Second Ch...	Mentre combattono strenu...	1	0944c363-...	AXDB	
McIntock!	Classica Western/Comedy ...	1	18054fe6-2...	AXDB	
La mia brunetta preferita	Nel braccio della morte dell...	1	2397bc2d-...	AXDB	
L'Erba di Grace	Il marito di Grace si è suicid...	1	3dc0e5e3-...	AXDB	
Love Laughs at Andy Ha...	Tornato dalla seconda Guer...	1	4142e15e-...	AXDB	
The Inspector General	Un venditore ambulante di ...	1	45810070-...	AXDB	
Il cameraman	Essendosi innamorato della ...	1	5f10d2d0-1...	AXDB	
La palla numero 13	Sherlock Jr e Ward tentano...	1	76e10529-...	AXDB	
I ragazzi del Marais	Regione del Marais, lungo l...	1	91e9165c-...	AXDB	
Il Navigatore	Rollo decide di sposare la s...	1	950ef7ed-2...	AXDB	
The perils of Pauline	Siamo agli inizi del secolo, q...	1	abdd95fe-d...	AXDB	
The sin of Harold Diddle...	Vent'anni dopo il suo trionf...	1	bcafec3f-7...	AXDB	
			did5	AXEPTOOL	
			did9	AXEPTOOL	
			did1	AXEPTOOL	

Add AX:OID

Query Result Panel

8.2 Relationship with other tools

The AXCP Rule Editor is related to Workflow tools and AXCP Rule Scheduler. The former for monitoring the rule creation and editing activity, the latter for posting and installing rules into the AXCP Grid Environment.

Finally, the current version of the Rule Editor embeds the Selection Editor for creating, testing and simulating complex queries onto the AXMEDIS Query Support.

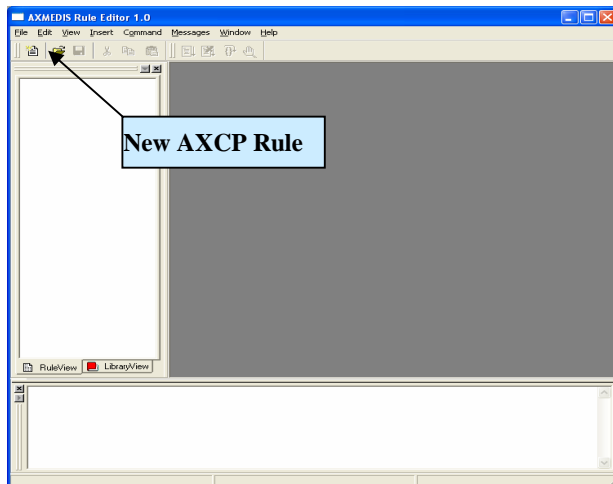
8.3 Detailed description of the functionalities and Screenshots

8.3.1 Editing on AXCP Rule

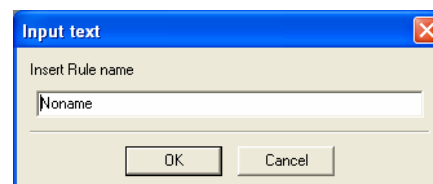
A content producer or manager wishes to create a content processing rule called AXCP Rule for manipulating/creating AXMEDIS multimedia objects. The AXCP Rule Editor is the editor for writing AXCP Rule. The AXCP Rule Editor GUI is a MDI window that manages a rule document at a time. The editor provides a set of tools and views to help the user during the editing and building of rule, writing JavaScript code. By opening the AXCP Rule Editor the user can create a rule to specify the time of firing. Once a rule has been created, the user can save it for re-editing at a future time or test the rule. Testing the rule enables the rule to be checked and debugged for any arising problems when executing the Javascript code and allow the user within the editor to make changes to successfully activating the rule.

8.3.2 Creating a new rule

To create a rule, the user starts the AXCP Rule Editor and from the opening screen creates a new Rule by selecting “New” from the tool bar or using the File menu (File → New) or using the keyboard shortcut “Ctrl+N” (see Screenshot 1). On requesting a new rule, the dialog box pops up and the user enters the name of the draft AXCP Rule and selects “OK” (see Screenshot 2).

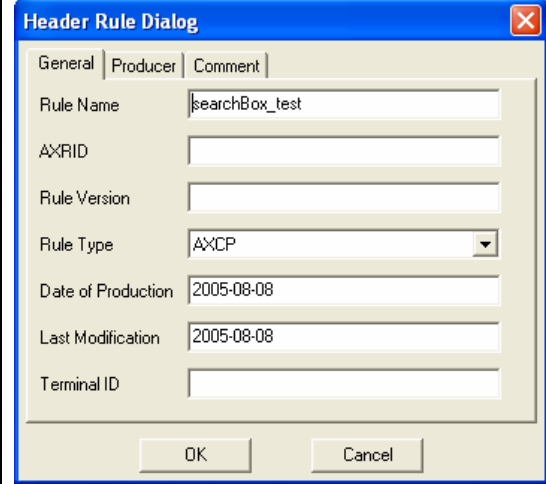
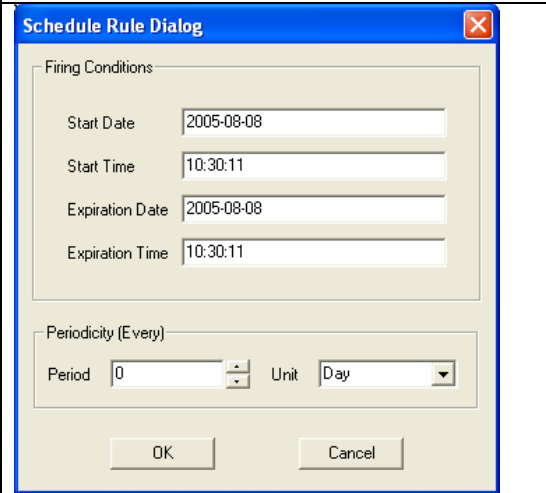
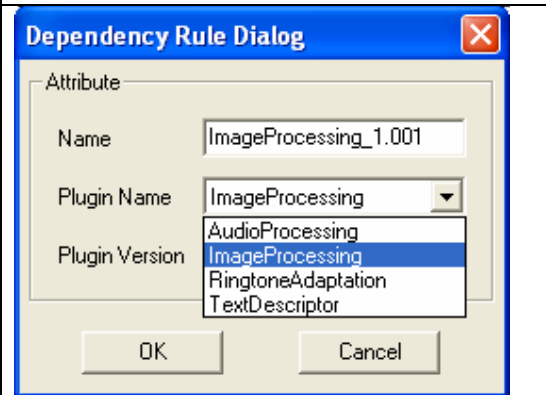


Open a new rule



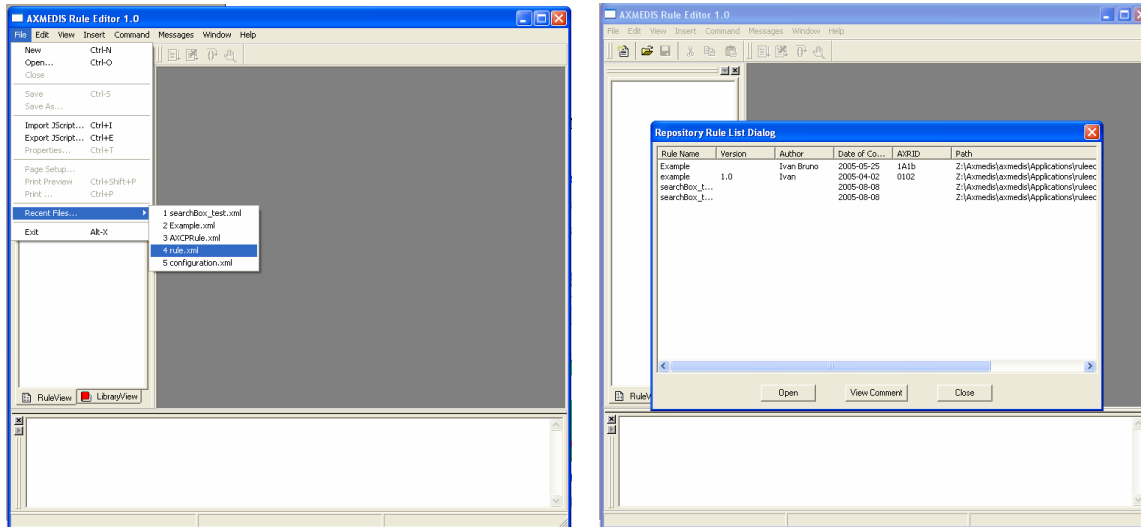
Input rule name dialog

After the rule name has been entered, the new rule is ready for editing with the ‘tree view’ used as a workspace and the editing dialogs for editing the rule details and making a rule schedule.

 <p>The screenshot shows the 'Header Rule Dialog' with tabs for 'General', 'Producer', and 'Comment'. The 'General' tab is active, showing fields for Rule Name (searchBox_test), AXRID, Rule Version, Rule Type (AXCP), Date of Production (2005-08-08), Last Modification (2005-08-08), and Terminal ID. There are OK and Cancel buttons at the bottom.</p>	<p>Header dialog</p> <p>This is the dialog that allows filling fields of the header section. The dialog is an OK/Cancel modal dialog in a notebook style with <i>General</i>, <i>Producer</i> and <i>Comment</i> tab where the list of items to edit is displayed.</p> <p>Note: The AXRID field is read only and it is pre-filled with the identification code assigned by the rule editor</p>
 <p>The screenshot shows the 'Schedule Rule Dialog' with a 'Firing Conditions' section containing Start Date (2005-08-08), Start Time (10:30:11), Expiration Date (2005-08-08), and Expiration Time (10:30:11). Below is a 'Periodicity (Every)' section with a Period field set to 0 and a Unit dropdown set to Day. There are OK and Cancel buttons at the bottom.</p>	<p>Schedule dialog</p> <p>This dialog allows filling/editing fields for a schedule item. The dialog is as an OK/Cancel modal dialog and displays the list of items to edit. The dialog allows setting the start date and time, the expiration date and time, and the conditions of periodicity (how many times in term of unit such as days, week, month and so on)</p>
 <p>The screenshot shows the 'Dependency Rule Dialog' with an 'Attribute' section. It has fields for Name (ImageProcessing_1.001), Plugin Name (ImageProcessing), and Plugin Version. A dropdown menu is open for Plugin Name, showing options: AudioProcessing, ImageProcessing (highlighted), RingtoneAdaptation, and TextDescriptor. There are OK and Cancel buttons at the bottom.</p>	<p>Dependency dialog</p> <p>This dialog allows filling fields for a dependency item. The dialog is an OK/Cancel modal dialog and displays the list of available plugins in order to facilitate the choice.</p>

8.3.3 Loading an existing rule

Instead of creating a new rule, the user may wish to edit an existing draft rule. The user may have saved it as a file somewhere or saved it in the AXCP Rule Repository. By selecting 'open' or 'Rule List' or selecting a file in the history list ('Recent Files...'), as seen in screenshot 4 and 5, the rule editor can load an existing draft rule from the Repository or elsewhere on the system.



Open commands shows the command Open dialog

The *Rule List* command opens a rules list modal dialog displaying all rules stored in the repository of the AXCP Rule editor. In this window, the list of rules will be organised in a table built on the following subset of metadata:

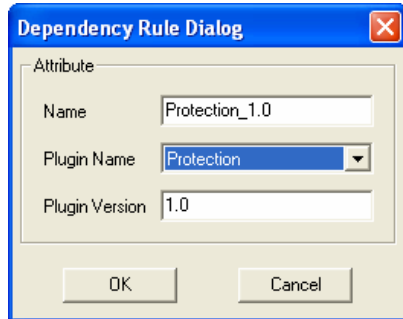
- *Rule Name*
- *Rule Version*
- *Author*
- *Date of composition*
- *Rule ID (AXRID)*

The user can select a specific rule in order to open it in the rule editor. Such operation is possible by pushing the *Open* button or double clicking on the line of the chosen rule. The user can visualize the comment associated with rule by pushing the *View Comment* button, the comment is displayed the *Output Area*. Otherwise the user can cancel the operation by closing the dialog or pushing the *Close* button.

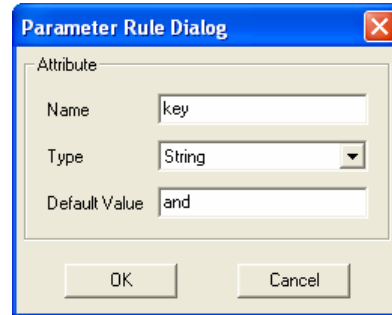
8.3.4 Editing a rule

The user can use the set of dialog and text editor to edit the AXCP rule (as shown in screenshots), and edit rule data such as:

- Parameter dialog for editing the attributes of a rule parameter
- Dependency dialog for editing the attributes of a AXMEDIS PlugIn
- XML Selection Editor (XML viewer/editor for the XML representation of selections) and JavaScript Editor based on Scintilla Editor for text/javascript code editing. It provides full editing capabilities (copy, cut, paste, redo, undo, syntax highlighting, etc...), print preview, page setup and print functionalities, syntax highlighting, brace highlighting, folding/hiding of lines, breakpoint insertion/removal, visualisation of line numbers



Dependency dialog

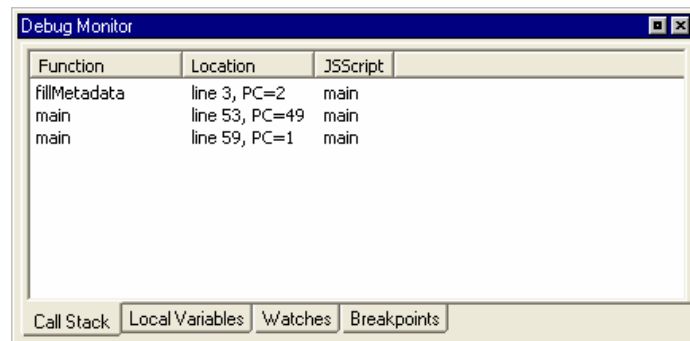


Parameter dialog

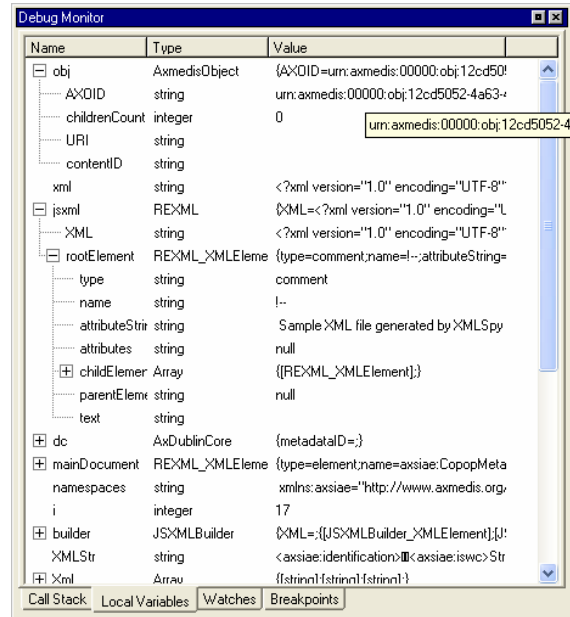
8.3.5 Debugging Rule functionalities

The editor provides functions to add/remove breakpoints (F9), start debug (F5), next breakpoint (F5), step over (F10) and step into (F11), stack calls monitoring, local variables visualization. In the output window a textual output provides internal errors or communication when script runs and during the debug session.

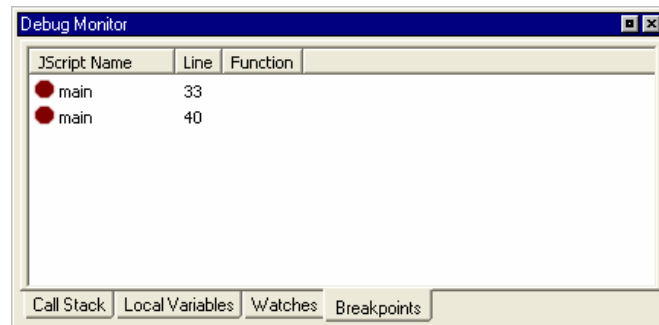
1. **Call Stack** – This is a page of the Debug dockable window that displays stack of functions calls. It displays the script name and the line of the call. Double clicking on an entry level of the stack allows focusing the view of the script where the corresponding call is located.



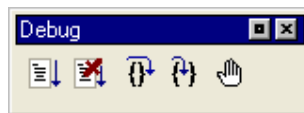
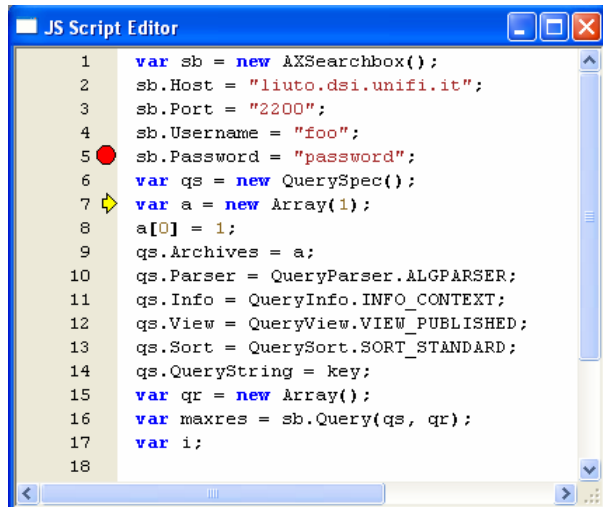
2. **Local Variables** – This is a page of the Debug dockable window that displays variables and instances of objects allocated by the script. They are displayed as tree list control with folding-unfolding capability for displaying the list of attributes of the object instance.



3. **Watches** – Not available in the current version.
4. **Breakpoints** - This a page of the Debug dockable window that displays the list of breakpoints inserted in the scripts: each breakpoint is associated with the name of the script and relative line number and status (Enabled or Disabled). Double clicking on a breakpoint allows focusing the view of the script where the breakpoint is placed.



5. **Debug markers:**
 - A red filled circle indicates an Enabled breakpoint
 - A red empty circle indicates a Disabled breakpoint
 - A yellow arrow indicates the line that will be executed.



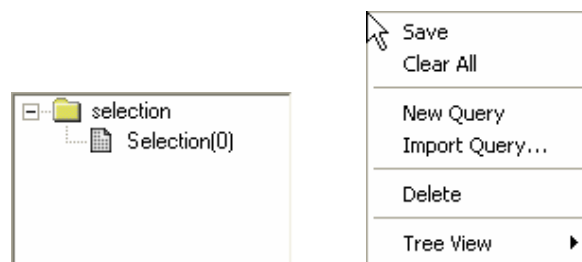
Main navigation interface of a Debug session

8.3.6 Activating a rule


The current version of the prototype allows installing a rule in the rule engine and notifying the completeness of the rule to the workflow manager. The activation can be done by using the *Activate* function in the Command menu or manually as described in the scenario 2.

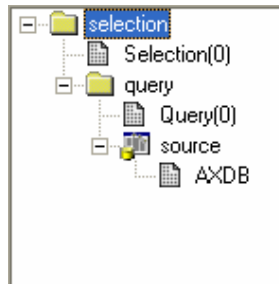
8.3.7 Creating a selection Parameter

This section describes how to create a selection document as parameter for an AXCP Rule. The user has to Insert a selection parameter in the Arguments section (Insert→ Selection), the Selection Editor opens an empty Selection document.



To add a new query the user can:

- Access to the popup menu on the tree view by right clicking on the “selection” item and then selecting the “New Query” function
- Use the AddQuery  button of the toolbar



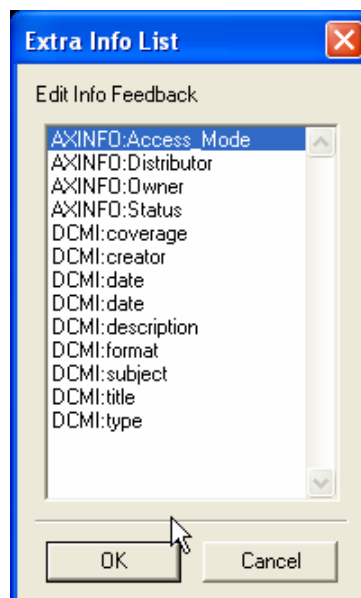
The new query becomes the current query and all operations affect it. The current query is displayed in the toolbar:



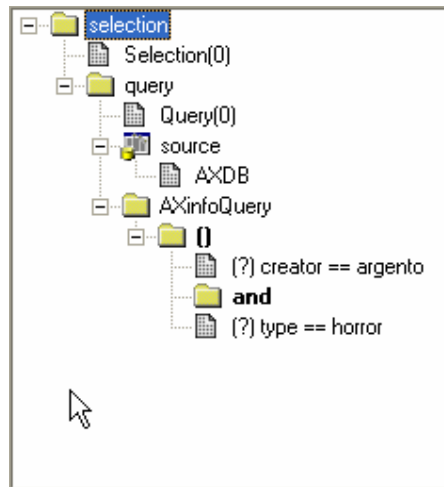
The new query is inserted and is filled with the source data choices currently set on the Available Sources. The source can be modify before adding or after the query. In both case the user has to selects sources in the corresponding panel. After the query insertion by right clicking on the new “query” item the popup menu on the tree view the user can select the “Set Query Source” function to apply the new sources.



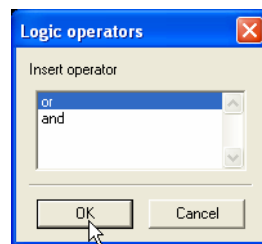
By means the Set Extra Info function on the same popup menu, the user can fill the query with the list of information to retrieve when the query is submitted to the Main Query Support. The function opens a multiple choice dialog as following:



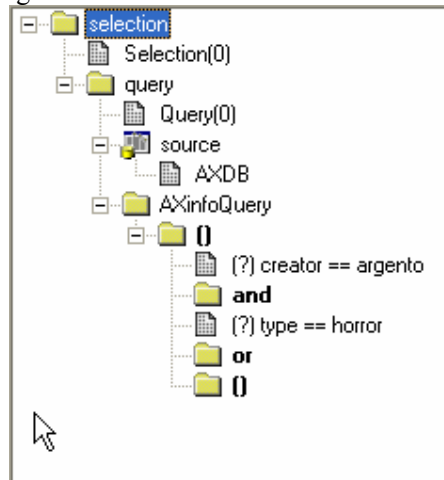
The user can start to insert conditions in the query by filling the fields in the Query Composer in a single or both panel. After filling, the user has to press the “AddCondition(s)” button to add them in the query.



To add a nesting level, the user has to press the “()” button in the Query Composer. A dialog will appear asking for the logic connector to use



The “()” will be inserted in the query and it will be the current level that can be edit. To change level the user has to select an existing nesting level.



During the editing the document is validated at runtime and the Validation Status panel provides the current status of the document:




To remove an item the user can select the item on the tree and by accessing to the popup menu calls the “delete” function.

8.3.8 Editing a selection Parameter

We assume to have an existing selection parameter to edit. The user has to open the popup menu by right clicking on the selection item in the tree view.

8.3.9 Testing a query

The user selects the query or by clicking on the query item of the tree or by choosing it on the toolbar in the choice box. Then, the user clicks on the  button on the toolbar to send the submission to the Query Support. The result is displayed in the Query Result panel.

AXMEDIS Query

Query Result


Results table


Title	Description	Version	Object Id	Source Cha...	St
Il tesoro del santo	In un bordello italiano una ...	1	08180e7b-...	AXDB	
Follie di Jazz (Second Ch...	Mentre combattono strenu...	1	0944c363-...	AXDB	
McLintock!	Classica Western/Comedy ...	1	18054fe6-2...	AXDB	
La mia brunetta preferita	Nel braccio della morte dell...	1	2397bc2d-...	AXDB	
L'Erba di Grace	Il marito di Grace si è suicid...	1	3dc0e5e3-...	AXDB	
Love Laughs at Andy Ha...	Tornato dalla seconda Guer...	1	4142e15e-...	AXDB	
The Inspector General	Un venditore ambulante di ...	1	45810070-...	AXDB	
Il cameraman	Essendosi innamorato della ...	1	5f10d2d0-1...	AXDB	
La palla numero 13	Sherlock Jr e Ward tentano...	1	76e10529-...	AXDB	
I ragazzi del Marais	Regione del Marais, lungo l...	1	91e9165c-...	AXDB	
Il Navigatore	Rollo decide di sposare la s...	1	950ef7ed-2...	AXDB	
The perils of Pauline	Siamo agli inizi del secolo, q...	1	abd995fe-d...	AXDB	
The sin of Harold Diddle...	Vent'anni dopo il suo trionf...	1	bc4fec3f-7...	AXDB	
			did5	AXEPTOOL	
			did9	AXEPTOOL	
			did1	AXEPTOOL	

Add AXDID

8.3.10 Load and Save a selection from/to the Selection Archive

A selection document can be loaded/saved from/to the Selection Archive.


To save a selection into the Archive, the user can clicks on the  icon.


To open an existing selection by loading from the Archive the user can click on the  icon and the list of available selection document is displayed in the Selection Archive panel. To ask for loading a document, the user selects the selection and by pressing the Open button, it will be loaded and the displayed in the editor.



8.3.11 Load and Save a selection from/to the File System

A selection document can be loaded/saved from/to the File System.

To save a selection into the File System , the user can clicks on the  icon.

To open an existing selection by loading from the File System the user can click on the  icon

8.3.12 Conclusion


Using the AXCP Rule Editor, the user can write an AXCP Rule and edit it to set when and how the digital contents have to be processed. Once checked, the activated rule is sent to the AXCP Rule Engine which uses AXMEDIS tools to run the javascript that describes the content processing procedure.

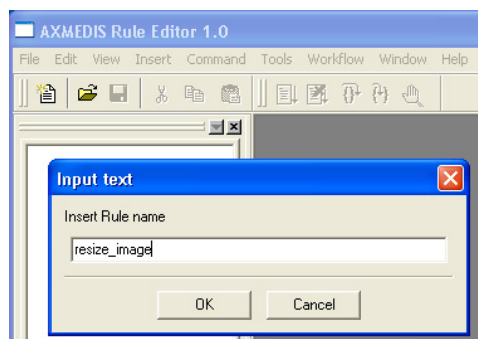
8.4 How to Create a rule, a small tutorial

In this tutorial we will show how to create a number of simple rules from scratch.

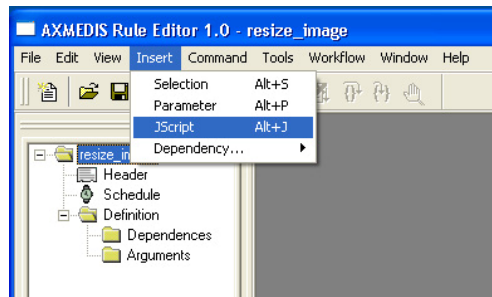
8.4.1 EXAMPLE 1 – Rule without parameters (Simple rule)

The first example will show how to create a simple script for resizing an image resource named *AXMEDIS_logo.png* stored in the C:\ path and saving the new resized image on the disk.

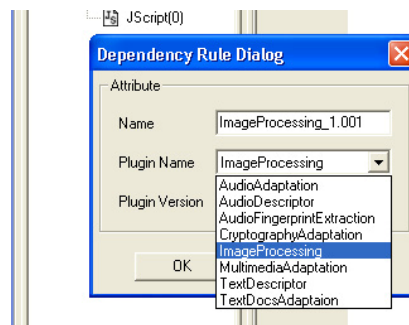
- Choose menu **File/New** or click on the  icon;
- Insert the name of the rule , for example *resize_image*;



- Choose the menu **Insert/JScript**, a new window will be opened where to write the JavaScript for the rule;



- In the tree view (Rule view in the workspace panel) select “Dependences” folder and right click on it. Choose **Insert Dependencies/Tool** in the menu to use functionalities provided by the AXMEDIS plugins.
- In the Dependency Rule Dialog select **ImageProcessing** in the **Plugin Name** choice box



- Selecting the **JScript** element on the tree view, the JS Editor, previously opened, will be activated. Then, write the following javascript piece of code in it

```

1 // 1) create an empty resource
2 var image = new AxResource();
3
4 // 2) load the image file by the selected path
5 image.load("C:\\AXMEDIS_logo.png");
6
7 // 3) Use the Image Processing plugin for scaling the image
8 ImageProcessing.Resize(image,320,200,true,image);
9 // the Resize function of the Image Processing plugin scales the image at 320*200
10 // size maintaining the Aspect ratio and overwriting the new image
11
12 // 4) the scaled image is saved as "img_scaled.png"
13 image.save("C:\\img_scaled");

```

Note: The line with “//” are comments and describe the meaning of javascript instruction.

In this example, the instruction to resize the image is the function in line 8 `ImageProcessing.Resize` followed by a number of parameters into parenthesis. Selecting the **LibraryView** tab it is possible to see the complete list of available processing functions. By double clicking on the **Resize** function in the **ImageProcessing** folder will appear the help window showing the needed parameters.


In this case the **Resize** function needs the following parameters:

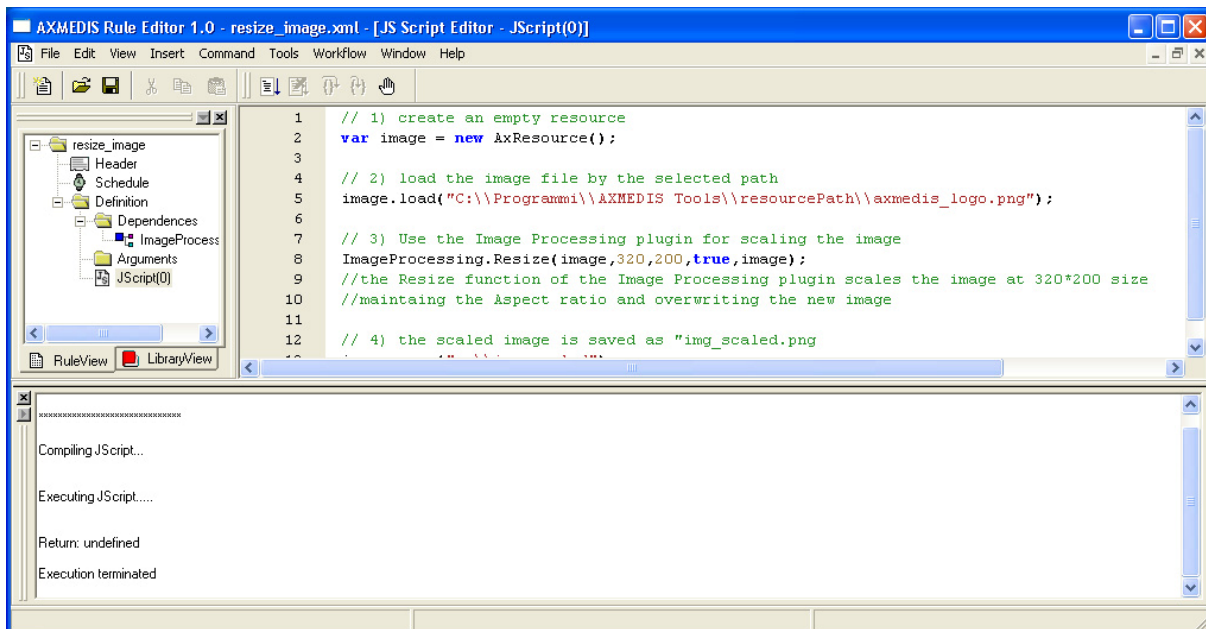
- the resource to be resized (i.e. in our script is `image`)
- the new image width (i.e. 320 in our script)
- the new image height (i.e. 200 in our script)

- an indication for preserving the image aspect ratio or not (i.e. `true` in our case will preserve the aspect ratio)
- the output resource, (i.e. `image` in our case: this will replace the original image with the new resized one)

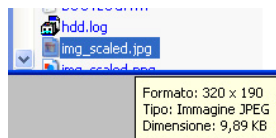
The order of parameters is important to use correctly the function and avoid possible execution errors.

Now, we can continue to produce our first rule by:

- Saving the rule in the menu **File/Save**
- Running the rule pressing the **Start Debug** button 
- The output windows, on bottom of the editor, will show if errors are present in the script, the “Execution terminated” message advises the end of run.



- After the rule execution , it is possible to open the new resource created on the disk in the C:\\ path and named *img_scaled.png*.



When we save a rule initially edited as JavaScript code the Rule Editor saves it in an xml file. In the previous example, the *image_resize.xml* file is the following:

```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <Rule xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="Rule_Axmedis.xsd">
- <Header>
  <Rule_Name>resize_image</Rule_Name>
  <AXRID>axcprule:95409e53-f678-46ba-b046-841de1b727ce</AXRID>
  <Rule_Version />
  <Rule_Type>AXCP</Rule_Type>
  <Software_Name />
  <Version_of_software />
  <Date_of_production>2006-08-31</Date_of_production>
  <Author />
  <Affiliation />
  <URL />
  <Comment />
  <Last_Modifications>2006-08-31</Last_Modifications>
  <Terminal_ID />
  <Cost />
  <Work_Item_ID />
</Header>
- <Schedule>
- <Run>
  <Date>2006-08-31</Date>
  <Time>17:15:21</Time>
  <Periodicity Unit="Day">0</Periodicity>
  <Expiration_Date>2006-08-31</Expiration_Date>
  <Expiration_Time>17:15:21</Expiration_Time>
</Run>
  <Status>Inactive</Status>
</Schedule>
- <Definition>
- <AXCP_Rule>
  <Arguments />
  <Rule_Body>
    - <JS_Script name="JScript(0)">
      - <![CDATA[
        // 1) create an empty resource
        var image = new AxResource();

        // 2) load the image file by the selected path
        image.load("C:\\Programmi\\AXMEDIS Tools\\resourcePath\\axmedis_logo.png");

        // 3) Use the Image Processing plugin for scaling the image
        ImageProcessing.Resize(image,320,200,true,image);
        //the Resize function of the Image Processing plugin scales the image at 320*200 size
        //maintaining the Aspect ratio and overwriting the new image

        // 4) the scaled image is saved as "img_scaled.png"
        image.save("c:\\img_scaled");
      ]]>
    </JS_Script>
  </Rule_Body>
- <Dependencies>
  - <Dependency>
    <Plug_In_name>ImageProcessing</Plug_In_name>
    <Version>1.001</Version>
  </Dependency>
</Dependencies>
</AXCP_Rule>
</Definition>
</Rule>

```

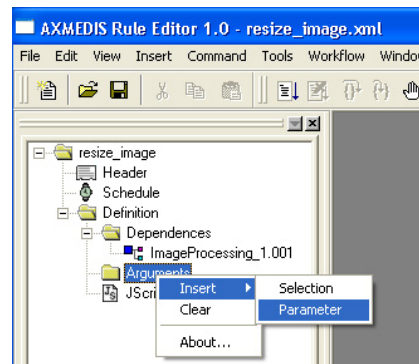

8.4.2 EXAMPLE 2 – Rule with Parameters

In the following example we will generalize the previous script inserting some generic arguments in the script.

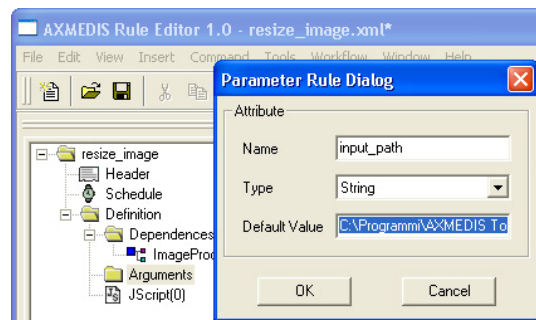
Instead to specify directly in the rule where the resource has to be loaded and saved (as specified in line 5 and 13 of the previous script), we can specify a generic argument in the script. In this manner the rule will remain valid and will be not modified in the future.

To do this we have to add a number of parameters in our rule.

- right click on the **Arguments** element in the tree view and select **Insert/Parameter** in the contextual menu;



- Add a first argument (the resource to be resized, in this case is the AXMEDIS logo in png format stored in the C:\Programmi\AXMEDISTools\resourcePath path)
 - **Name** input_path
 - **Type** String
 - **Default Value** C:\Programmi\AXMEDISTools\resourcePath\AXMEDIS_logo.png



- Add a second argument (the resized output resource will be saved in C:)
 - **Name** output_path
 - **Type** String
 - **Default Value** c:\img_scaled
- Add a second argument (the new image width is 320 pixels)
 - **Name** width
 - **Type** Integer
 - **Default Value** 320
- Add a third argument (the new image height is 200 pixel)
 - **Name** height
 - **Type** Integer

- **Default Value** 200
- Add a forth argument for a conversion of the image in jpeg format
 - **Name** out_mime_type
 - **Type** String
 - **Default Value** image/jpeg

Now we have to change the script deleting the specific path overwriting then with the new generic arguments and adding a new line for the format conversion. The script is the following:

```

1  // 1) create an empty resource
2  var image = new AxResource();
3
4  // 2) load the image file by the "input_path" argument
5  image.load(input_path);
6
7  // 3) Use the Image Processing plugin for scaling the image
8  ImageProcessing.Resize(image,width,height,true,image);
9  // the "Resize" function of the Image Processing plugin scales the image at 320*200
10 // size maintaining the Aspect ratio and overwriting the new image
11
12 // 4) Use the Image Processing plugin for "Conversion" the image
13 ImageProcessing.Conversion(image,out_mime_type,image);
14 // the "Conversion" function of the Image Processing plugin converts the image
15 // and overwriting the new image
16
17 // 5) the scaled image is saved in the location specified by the "output_path" argument
18 image.save(output_path);

```

When the rule is executed a new image named `img_scaled.jpg` will be saved in C:

8.4.3 EXAMPLE 3 – Rule for creating AXMEDIS Object

As third example we will see how apply the rule to an AXMEDIS object.

More in deep we will create a new AXMEDIS object with the new converted image as embedded resource.

To do this we have firstly to delete the output image path in line 18 adding only three instructions:

```

// Create an empty AXMEDIS object
var axObj = new AXMEDISObject();

// Add the image resource as a new content
axObj.addContent(image);

// Save the AXMEDIS object
axObj.save("c:\\newAXMEDISObject.axm");

```

The final script is the following:

```

1  // 1) create an empty resource
2  var image = new AxResource();
3
4  // 2) load the image file by the "input_path" argument
5  image.load(input_path);
6
7  // 3) Use the Image Processing plugin for scaling the image
8  ImageProcessing.Resize(image,width,height,true,image);
9  // the "Resize" function of the Image Processing plugin scales the image at 320*200
10 // size maintaining the Aspect ratio and overwriting the new image
11
12 // 4) Use the Image Processing plugin for "Conversion" the image
13 ImageProcessing.Conversion(image,out_mime_type,image);
14 // the "Conversion" function of the Image Processing plugin converts the image
15 // and overwriting the new image

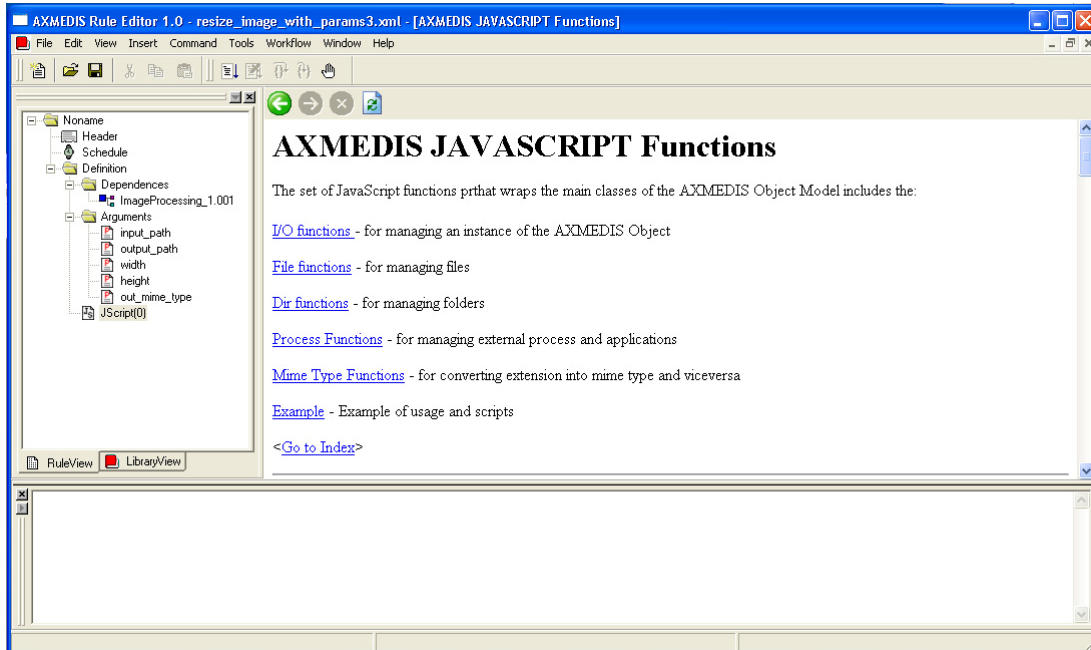
```

```

16
17 // Create an empty AXMEDIS object
18 var axObj = new AXMEDISObject();
19
20 // Add the image resource as a new content
21 axObj.addContent(image);
22
23 // Save the AXMEDIS object
24 axObj.save("c:\\newAXMEDISObject.axm");

```

All the available functions interacting with the AXMEDIS Editor are listed and explained in the **AXMEDIS Javascript Reference Manual** available in the **Help/About** menu.



9 AXMEDIS Rule Engine - Rule Scheduler (DSI)

9.1 Main functionalities

The AXCP Rule engine is divided in two main components:

- **Rule Scheduler (Server Side)** – It consists of the an internal Scheduler and Dispatcher. It performs the operations of rule installation, rule firing, rule executor discovering and management, rules dispatching, communication with the AXMEDIS environment, etc....
- **Rule Remote Executor (Client Side)** – It is the executor of rules and consists of a script engine based on JavaScript (JS) SpiderMonkey released by Mozilla. It runs the JavaScript code associated with rule.

9.2 Relationship with other tools

The AXCP Rule Scheduler is related to Workflow tools for monitoring the GRID activity and rules.

9.3 Detailed description of the functionalities and Screenshots

The Rule Scheduler GUI is the main window that allows the interaction with the Scheduler. It is constituted of:

1. A menu bar
2. Two main areas where the list of rules and the list of remote executors are displayed.
3. A status bar where the current clock and the current date are displayed.

9.3.1 Menu bar

It provides the access to the following set of functions:

1. Program

- a. **Add Rule** – It allows to load rules and install into the scheduler
- b. *Launch scheduler* - Start the scheduler activity.
- c. *Stop scheduler* - Stop the scheduler activity.
- d. *Backup* - Backup Copy of the last jobs list.
- e. *Restore* – Restore a backup copy
- f. *Minimize* - It reduces at icon on the taskbar.
- g. *Exit* - Close the application.
- h. *Start Grid Peer Functions* – It starts the components for P2P network access

AddRule	Ctrl-A
Launch scheduler	Ctrl-L
Stop Scheduler	Ctrl-S
Backup	Alt-B
Restore	Alt-R
Minimize	Alt-M
Exit	Alt-X
Start Grid Peer functions	Ctrl-F
test replace parameters	
test replace schedule	
test install whole rule as string	
test replace parameters and schedule	

2. Settings

- a. *Preferences* - Open an editable dialog with the set of configuration parameters.

3. View

- a. *Refresh* – Update the list of jobs and list of remote executors.
- b. *Arrange* – Repainting modes of tables in the main frame
 - i. *Top* – It shows only the top table (Table of rules)
 - ii. *Bottom* – It shows only the bottom table (Table of executors)
 - iii. *Vertical* – It shows tables vertically
 - iv. *Horizontal* – It shows tables horizontally

- c. *Rule Properties...* - Open a Rule Properties dialog.
- d. *Executor Profile...* - Open an Executor Profile dialog.
- e. *Logs...* - Open a dialog to show the list of log messages
- f. *Debug Monitor...* - A dialog for debug purpose

Refresh	Alt-E
Arrange	▶
Rule Properties...	Ctrl-R
Executor Profile...	Ctrl-E
Logs...	Alt-L
Debug Monitor...	Alt-O

4. Commands

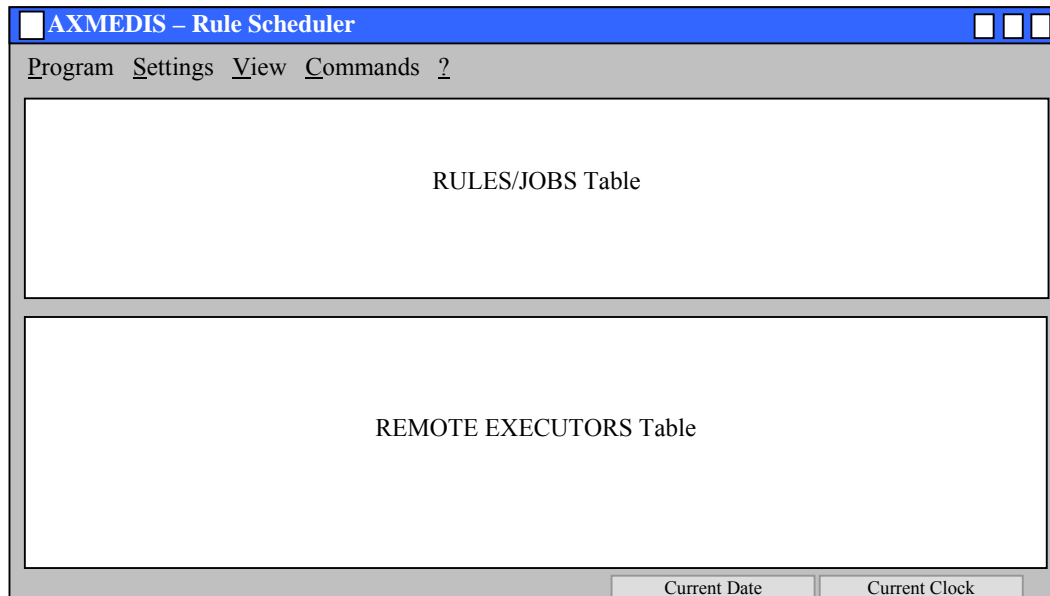
- a. *Enable Rule* - Put in the “ACTIVE” status the current selected inactive rule.
- b. *Disable Rule* - Put in the “INACTIVE” status the current selected active rule.
- c. *Kill Rule* - Kill the current execution of the current selected rule.
- d. *Pause Rule* - Put in pause the execution the current selected rule.
- e. *Resume Rule* - Resume the execution of the current selected rule.
- f. *Suspend Rule...* - Open a dialog to edit the temporal interval for rule resuming and then suspend the current selected rule.
- g. *Remove Rule* – Remove the rule from the list of rules

Enable Rule	Alt-N
Disable Rule	Alt-D
Kill Rule	Alt-K
Pause Rule	Alt-P
Resume Rule	Alt-S
Suspend Rule...	Alt-U
Remove Rule...	Ctrl-M

5. Other

- a. *Help* - Open the On Line help.
- b. *About* - Open a dialog with credits.

All this functionalities are also accessible by means shortcuts.



The screenshot shows the 'AXMEDIS - Rule Scheduler' window. It has a menu bar with 'Program', 'Settings', 'View', 'Commands', and '?'. Below the menu bar is a table with the following columns: Rule Name, AXRID, Rule Version, Rule Status, Job ID, Executor ID, Start Time, Start Date, Periodicity, and Number of Runs. The table lists 40 rules, mostly named 'searchBox_...', with various statuses like 'completed', 'running', 'delayed', and 'N° Runs' ranging from 0 to 1. At the bottom of the window, there is a status bar showing the date '23/09/2005' and time '16:35:19'.

Rule Name	AXRID	Rule Version	Rule Status	Job ID	Executor ID	Start Time	Start Date	Periodicity	Number of Runs
searchBox_...	9	-1	completed	9	-1	16:05:11	09/23/05	0	1
searchBox_...	10	-1	completed	10	-1	16:05:11	09/23/05	0	1
searchBox_...	11	-1	completed	11	-1	16:05:11	09/23/05	0	1
searchBox_...	12	-1	completed	12	-1	16:05:11	09/23/05	0	1
searchBox_...	13	2	running	13	2	16:05:11	09/23/05	0	0
searchBox_...	14	-1	completed	14	-1	16:05:11	09/23/05	0	1
searchBox_...	15	-1	completed	15	-1	16:05:11	09/23/05	0	1
searchBox_...	16	3	running	16	3	16:05:11	09/23/05	0	0
searchBox_...	17	-1	completed	17	-1	16:05:11	09/23/05	0	1
searchBox_...	18	-1	completed	18	-1	16:05:11	09/23/05	0	1
searchBox_...	19	-1	completed	19	-1	16:05:11	09/23/05	0	1
searchBox_...	20	-1	completed	20	-1	16:05:11	09/23/05	0	1
searchBox_...	21	-1	completed	21	-1	16:05:11	09/23/05	0	1
searchBox_...	22	-1	completed	22	-1	16:05:11	09/23/05	0	1
searchBox_...	23	-1	completed	23	-1	16:05:11	09/23/05	0	1
searchBox_...	24	8	running	24	8	16:05:11	09/23/05	0	0
searchBox_...	25	-1	completed	25	-1	16:05:11	09/23/05	0	1
searchBox_...	26	-1	completed	26	-1	16:05:11	09/23/05	0	1
searchBox_...	27	-1	completed	27	-1	16:05:11	09/23/05	0	1
searchBox_...	28	-1	completed	28	-1	16:05:11	09/23/05	0	1
searchBox_...	29	-1	completed	29	-1	16:05:11	09/23/05	0	1
searchBox_...	30	-1	completed	30	-1	16:05:11	09/23/05	0	1
searchBox_...	31	-1	completed	31	-1	16:05:11	09/23/05	0	1
searchBox_...	32	-1	completed	32	-1	16:05:11	09/23/05	0	1
searchBox_...	33	7	running	33	7	16:05:11	09/23/05	0	0
searchBox_...	34	-1	completed	34	-1	16:05:11	09/23/05	0	1
searchBox_...	35	9	running	35	9	16:05:11	09/23/05	0	0
searchBox_...	36	6	running	36	6	16:05:11	09/23/05	0	0
searchBox_...	37	-1	delayed	37	-1	16:05:11	09/23/05	0	0
searchBox_...	38	-1	delayed	38	-1	16:05:11	09/23/05	0	0
searchBox_...	39	-1	delayed	39	-1	16:05:11	09/23/05	0	0
searchBox_...	40	-1	delayed	40	-1	16:05:11	09/23/05	0	0

Executor Name	IP	CPU	Clock	OS	Transfer Rate	HD Space	Status	Rule ID	Executor ID	Workload p...	Start Time	End Ti
DISIT-01	192.168.0.197	intel	1800	Windows N...	-1	1073741824	busy	2	1	0.000000	15:04:38	15:04
DISIT-04	192.168.0.105	intel	1800	Windows N...	-1	452984832	busy	13	2	0.000000	15:04:05	15:04
DISIT-03	192.168.0.52	intel	1800	Windows N...	-1	89128960	busy	16	3	0.000000	15:27:33	15:27
DISIT-02	192.168.0.43	intel	1800	Windows N...	-1	84934656	busy	5	4	0.000000	15:45:09	15:45
MIRCOFANT	192.168.0.64	intel	1800	Windows N...	-1	1073741824	busy	6	5	0.000000	15:53:06	15:53
VENOM-WORK	192.168.0.103	intel	1800	Windows N...	-1	2147483648	busy	36	6	0.000000	16:01:28	16:01
M386	192.168.0.49	intel	1800	Windows N...	-1	2147483648	busy	33	7	0.000000	16:05:11	16:05
DISIT-05	192.168.0.102	intel	1800	Windows N...	-1	0	busy	24	8	0.000000	16:25:49	16:25
HOMER	192.168.0.101	intel	1800	Windows N...	-1	2147483648	busy	35	9	0.000000	16:35:48	16:35

9.3.2 Rules/Jobs Table

It is the area where scheduled rules are displayed. It is a list control constituted of a set of columns where the following list of metadata are displayed:

- *Rule name* –the name of the rule
- *Rule version* –the version of rule
- *Rule status* –the current status of rule
- *Rule ID* –the identifier of rule
- *Executor ID* –the identifier of the executor associated with rule
- *Start Time* –the time to fire the rule
- *Start Date* –the date to fire the rule
- *Periodicity* –the periodic attribute
- *N° Runs* –the number of time the rule was fired.

Name	Version	Status	ID	Executor ID	Start Date	Start Time	Periodicity	N° Runs

The following functionalities are provided by means a contextual popup menu:

- Ordering rules alphabetically by name
- Ordering rules by start running time
- Ordering rules by ID

9.3.3 Remote Executors Table

It is the area where remote executors are displayed. It is a list control constituted of a set of columns where the following list of metadata is displayed:

- *Name* - Computer Name
- *IP* - IP address
- *CPU* - CPU & Clock
- *OS* - OS & Version
- *Ping* – The network capabilities in term of transmission time.
- *HD Space* – The space available on the disk of the executor
- *Status* – The status of the executor
- *Rule ID* – The ID of the running rule
- *Executor ID* – The Id of the executor assigned by the scheduler
- *Start Time* – At what time the run is started.

Name	IP	CPU	OS	Ping	HD Space	Status	ID	RuleID	Start Time

The following functionalities are provided by means a contextual popup menu:

- Ordering executors alphabetically by computer name
- Ordering executors by ID

9.3.4 Auxiliary dialogs

The Scheduler GUI is supported by the following set of dialogs:

9.3.5 Rule Properties Dialog

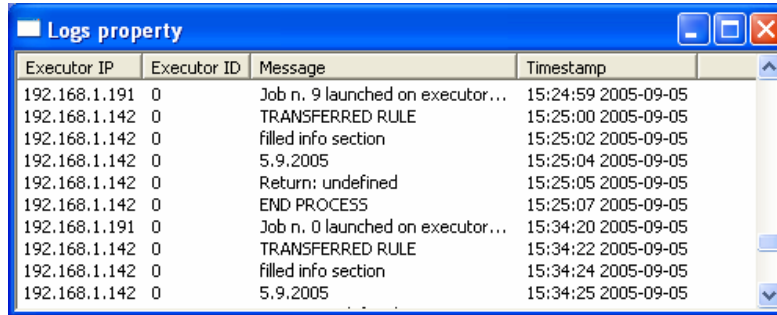
It is an editable no modal dialog where the properties of the selected rule are displayed. Some of these properties are extracted from the XML file associated with rule.

9.3.6 Executor Profile Dialog

It is a not editable no modal dialog where the properties of the selected executor are displayed. Some of these properties are extracted from the executor profile.

9.3.7 Logs Dialog

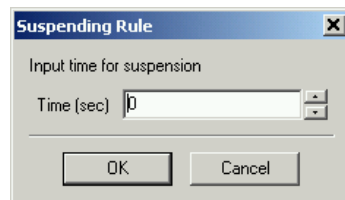
This dialog allows viewing the logs of scheduler activity.



Executor IP	Executor ID	Message	Timestamp
192.168.1.191	0	Job n. 9 launched on executor...	15:24:59 2005-09-05
192.168.1.142	0	TRANSFERRED RULE	15:25:00 2005-09-05
192.168.1.142	0	filled info section	15:25:02 2005-09-05
192.168.1.142	0	5.9.2005	15:25:04 2005-09-05
192.168.1.142	0	Return: undefined	15:25:05 2005-09-05
192.168.1.142	0	END PROCESS	15:25:07 2005-09-05
192.168.1.191	0	Job n. 0 launched on executor...	15:34:20 2005-09-05
192.168.1.142	0	TRANSFERRED RULE	15:34:22 2005-09-05
192.168.1.142	0	filled info section	15:34:24 2005-09-05
192.168.1.142	0	5.9.2005	15:34:25 2005-09-05

9.3.8 Suspend Rule Dialog

It is an editable no modal dialog where the user puts the time for the suspension.

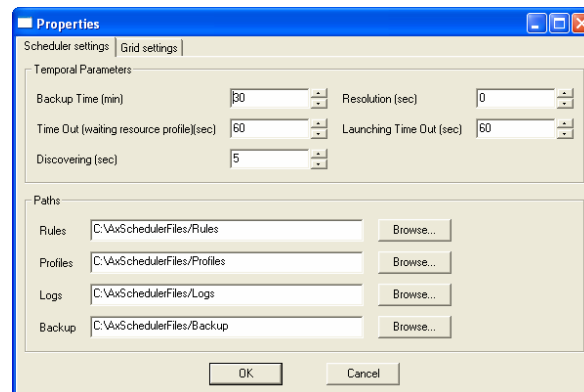


9.3.9 Preferences Dialog

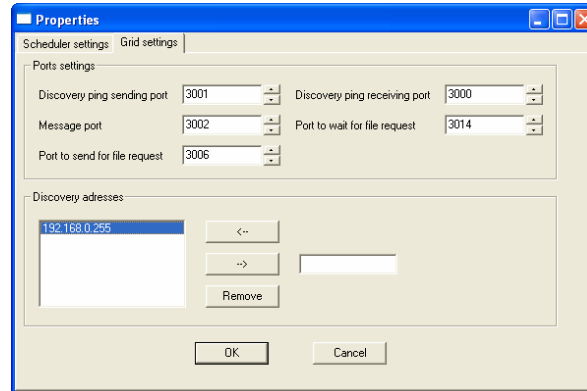
It is a tabbed dialog that allows editing settings parameters regarding the scheduler activity (*Scheduler settings*) and the GRID support (*GRID settings*).

Scheduler settings – It consists of a set of configuration parameters contains settings about:

- **Backup Time** - Backup interval for logging the set of submitted rule and tracing operations. It is expressed in minutes.
- **Time Out** - Time out on client activity. It is expressed in seconds.
- **Time Resolution** - Time Resolution of the scheduler. It is expressed in seconds.
- **Refresh Time** - Time Resolution for discovering new rule executors
- **Rules Path** - Rule Repository Path
- **Log Path** - Log Repository Path
- **Profile Path** - Executor Profile Repository Path
- **Backup Path** – The path where the scheduler periodically saves the current rules list.



Grid settings – It provides a set of settings to setup the communication support. It allows to define the number of ports to use when receiving file, messages, sending files, responding to the discovering request. It allows also to define IPs of LANs to use when the scheduler performs the discovering of peers.



9.3.10 Configuration Parameters

In this section the set of parameters regarding the configuration of the rule scheduler are listed. Such parameters are grouped into modules as reported below:

AXMEDIS Rule Scheduler Frame

Config parameter	Possible values
FRAME_SIZE	it is the frame size information
FRAME_POSITION	it is the frame position information
XML_XSD_PATH	It is the directory where xml schema (XSD files) are stored

AXMEDIS Rule Scheduler Settings

Config parameter	Possible values
BACKUP_TIME	It is the interval of Backup
RESOLUTION	It is the resolution in terms of time
TIME_OUT	It is the value of time out for waiting in the communication
DISCOVERING	It the discovering time
LAUNCHING_TIME_OUT	It is the maximum interval time for launching a rule
RULES_PATH	it is the directory where the rule will be saved
PROFILES_PATH	It is the directory where profiles of executors are stored
BACKUP_PATH	It is the directory where the backup file is saved
LOGS_PATH	It is the directory where logs are stored

AXMEDIS_GRID_SUPPORT_SETTINGS

Config parameter	Possible values
SERV_PORT	It is the value of the port used by the Grid Interface
SERV_PORT2	It is the value of the port used by the Grid Interface
SERV_PORT3	It is the value of the port used by the Grid Interface
SERV_PORT4	It is the value of the port used by the Grid Interface
SERV_PORT5	It is the value of the port used by the Grid Interface
DEFAULT_NET	It is the list of IP address

AXMEDIS Plugin Manager

Config parameter	Possible values
PLUGINS_PATH	it is the frame size information

WORKFLOW

Config parameter	Possible values
workflowUrl	it is the URL for workflow plugin

gatewayUrl	it is the gatewayURL for workflow plugin

9.3.11 Activating and stopping a rule

When the user is happy with his AXCP Rule and validated it with a quick and/or full trial, the final option is to activate the AXCP Rule in the AXCP Rule Engine. In the current version of the prototype the user has to install manually the AXCP in the Scheduler by selecting “*Add rule*” in Program menu. The scheduler will process the rule information and at the specified times will distribute the rule to a rule executor. During the running of the rule, the user can also stop it from the Scheduler by selecting “*Kill Rule*” in the Command menu.

9.3.12 Conclusion

Using the AXCP Rule Scheduler, the user can manually setup the AXCP Rule Engine and install an AXCP Rule in the AXCP Rule Engine. The user can also monitor the whole activity of the engine by means of logs. Using the AXCP GRID Node Executor, the user add a new Node Executor to the GRID Environment. Using the AXCP Stand Alone Executor, the user can run AXCP Rule independently from the GRID.

10 AXMEDIS Rule Engine - Rule Executor (DSI)

10.1 Main functionalities

The AXCP Rule engine is divided in two main components:

- **Rule Scheduler (Server Side)** – It consists of the an internal Scheduler and Dispatcher. It performs the operations of rule installation, rule firing, rule executor discovering and management, rules dispatching, communication with the AXMEDIS environment, etc....
- **Rule Remote Executor (Client Side)** – It is the executor of rules and consists of a script engine based on JavaScript (JS) SpiderMonkey released by Mozilla. It runs the JavaScript code associated with rule.

10.2 Relationship with other tools

The AXCP Rule Scheduler is related to Workflow tools for monitoring the GRID activity and rules.

10.3 Detailed description of the functionalities and Screenshots

The AXCP Rule Executor is provided as two executable files:

- **Stand alone AXCP Executor** - To run the Executor with an AXCP Rule, the user has to open the CMD dialog of Windows and type the line command using the option 0 as following:
axruleexecutor.exe <rule path> -0
- **AXCP Grid Node Executor** - To launch the AXCP Grid Node, the user has to double click on axcpgridnone.exe executable file, the application is ready to be discovered by the AXCP Rule Scheduler and it is in waiting state. The User Interface of the rule executor as GRID Node is shown in the following picture. It is a console application and the output of the engine for direct messages.

```

C:\Programmi\AXMEDIS Tools\AXCP-tools\ruleexecutor\axcpgridnode.exe
*****
* Welcome on AXMEDIS AXCP GRID NETWORK *
*****
Initializing AXCP GRID NODE
-> Starting GRID peer functions...
-> GRID peer functions started
-> GRID Cpu Monitor started
AXCP GRID NODE Ready
Profile request
Profile sent

```

Before the first launch of both applications, the user should setup the configuration file in order to provide the right value to the constants and parameters used by the application. See the following tables for the Configuration Parameters

AXMEDIS Rule Executor

Config parameter	Possible values
XML_RULE_PATH	it is the directory where the rule will be saved
XML_XSD_PATH	It is the directory where xml schema (XSD files) are stored
LOGS_PATH	It is the directory where logs are stored

AXMEDIS_GRID_SUPPORT_SETTINGS (only for GRID Node)

Config parameter	Possible values
SERV_PORT	It is the value of the port used by the Grid Interface

SERV_PORT2	It is the value of the port used by the Grid Interface
SERV_PORT3	It is the value of the port used by the Grid Interface
SERV_PORT4	It is the value of the port used by the Grid Interface
SERV_PORT5	It is the value of the port used by the Grid Interface
DEFAULT_NET	It is the list of IP address

AXMEDIS Plugin Manager

Config parameter	Possible values
PLUGINS_PATH	It is the directory where the DLL of plug-ins with their profiles (workflow, adaptation, descriptor and fingerprint estimators) are stored.

AXMEDIS Database

Config parameter	Possible values
user	The user name for logging into Database
passwd	The password for logging into Database
LoaderWSEndPoint	It is the URL for accessing to the load web service
HTTPath	
UploadPath	
SaverWSEndPoint	It is the URL for accessing to the save web service

AXMEDIS Selection

Config parameter	Possible values
MAIN_QUERY_SUPPORT_WSDL	It is the URL of the WSDL for using the Main Query Support
SELECTION_ARCHIVE_WSDL	It is the URL of the WSDL for using the Selection Archive

Conclusion

Using the AXCP Rule Scheduler, the user can manually setup the AXCP Rule Engine and install an AXCP Rule in the AXCP Rule Engine. The user can also monitor the whole activity of the engine by means of logs. Using the AXCP GRID Node Executor, the user add a new Node Executor to the GRID Environment. Using the AXCP Stand Alone Executor, the user can run AXCP Rule independently from the GRID.

10.4 How to use the GRID and Rule Executor stand alone, a small tutorial

In this tutorial we will see how to use the GRID (Rule Scheduler and Rule Executors) and the stand alone version of the Rule Executor by using the script of the rule created in section **Errore. L'origine riferimento non è stata trovata.** and reported below:

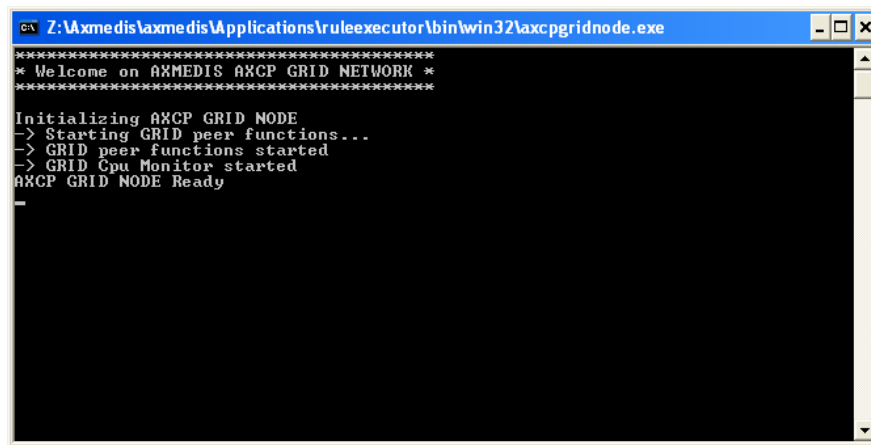
```

1  // 1) create an empty resource
2  var image = new AxResource();
3
4  // 2) load the image file by the selected path
5  image.load("C:\\AXMEDIS_logo.png");
6
7  // 3) Use the Image Processing plugin for scaling the image
8  ImageProcessing.Resize(image,320,200,true,image);
9  // the Resize function of the Image Processing plugin scales the image at 320*200
10 // size maintaining the Aspect ratio and overwriting the new image
11
12 // 4) the scaled image is saved as "img_scaled.png"
13 image.save("c:\\img_scaled");

```

10.4.1 Rule execution in the AXCP GRID environment

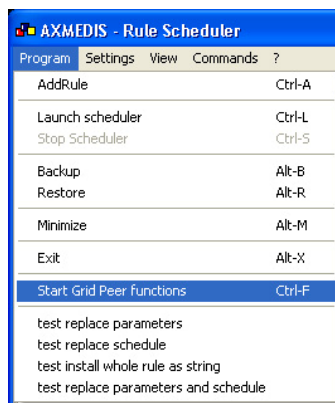
First of all it is necessary to install in one or more PCs the **Rule Executor**, then start them by double clicking on **axcpgridnode.exe** executable file. It is also possible to start it in the local machine. If the **Rule Executor** started correctly you will see the following messages.



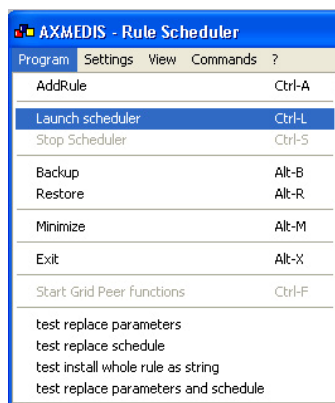
```

Z:\Axmedis\axmedis\Applications\ruleexecutor\bin\win32\axcpgridnode.exe
*****
* Welcome on AXMEDIS AXCP GRID NETWORK *
*****
Initializing AXCP GRID NODE
-> Starting GRID peer functions...
-> GRID peer functions started
-> GRID Cpu Monitor started
AXCP GRID NODE Ready
  
```

Then start the **Rule Scheduler** executable file. In the menu **Program** select **Start Grid Peer Functions**;



In the menu **Program** select **Launch Scheduler**



After a few seconds the list of computer connected in the P2P network appears with a number of additional information

Executor N...	IP Address	CPU Type	Clock	OS	Transfer Rate	HD Space	Status	Job ID	Executor ID	Cpu Usage	From	To
MUSICA-ATT...	192.168.0.86	x86 Family ...	2.19 (GHz)	Windows N...	1 (MB/s)	3.49 (GB)	ready	-1	1	30.00%	15:00:00	15:59:00
IVAN-PORTA...	192.168.0.85	x86 Family ...	1.60 (GHz)	Windows N...	14 (KB/s)	23.45 (GB)	ready	-1	6	30.00%	15:00:00	15:59:00

The AXCPGrid node will show message about the profile request as depicted in the following picture:

```

C:\Programmi\AXMEDIS Tools\AXCP-tools\ruleexecutor\axcpgridnode.exe
*****
* Welcome on AXMEDIS AXCP GRID NETWORK *
*****
Initializing AXCP GRID NODE
-> Starting GRID peer functions...
-> GRID peer functions started
-> GRID Cpu Monitor started
AXCP GRID NODE Ready
Profile request
Profile sent

```

Now it is possible to add a rule to be processed dragging it into the Scheduler or using the **Add Rule** function in the **Program** menu. The rule appears in the bottom area.

If the **Rule Status** is **Inactive**, the rule has to be activated by right-clicking on it and selecting the **Enable Rule** option in the contextual menu:

AXMEDIS - Rule Scheduler											
Program Settings View Commands ?											
Rule Name	AXRID	Rule Version	Rule Status	Job ID	Executor ID	Start Time	Start Date	Periodicity	Number of ...	Installation ...	Expiration ...
resize_image	axcp rule:9...		Inactive	0		21	07/09/2006	0 Day(s)	0	2006-09-08...	2007-08-31...

If the temporal conditions are not valid the status becomes failure. This means that you have to modify by using the rule editor the Schedule item by inserting the right value for start time, start date, expiration time and date. Then, installing again the rule in the scheduler.

During the rule execution the showed status is **Running** and the Rule Scheduler shows also the computer executing the rule in the **Executor ID** column.

When the rule has been executed its status changes in **Completed**.

AXMEDIS - Rule Scheduler

Rule Name	AXRID	Rule Version	Rule Status	Job ID	Executor ID	Start Time	Start Date	Periodicity	Number of ...	Installation ...	Expiration ...
resize_image	axcprule:9...		Completed	0	-1	15:36:21	07/09/2006	0 Day(s)	1	2006-09-07...	2007-08-31...

Executor N...	IP Address	CPU Type	Clock	OS	Transfer Rate	HD Space	Status	Job ID	Executor ID	Cpu Usage	From	To
MUSICA-ATT...	192.168.0.86	x86 Family ...	2.19 (GHz)	Windows N...	1 (MB/s)	3.49 (GB)	ready	-1	1	30.00%	15:00:00	15:59:00
IVAN-PORTA...	192.168.0.85	x86 Family ...	1.60 (GHz)	Windows N...	14 (kB/s)	23.45 (GB)	ready	-1	6	30.00%	15:00:00	15:59:00

07/09/2006 15:57:06

During the rule execution, the Rule Executor window could show output messages if coming from the script or internal ones.

```

C:\>C:\axcpgridnode\axruleexecutor c:\resize_image.xml
*****
* Welcome on AXMEDIS AXCP Rule Executor *
*****

Initializing AXCP Executor...
Running AXCP Rule...
Return: undefined

C:\>

```

10.4.2 Rule execution by means the stand alone version of the Rule Executor

In this section we will see how to execute a rule using the Command Prompt. We suppose that the rule to be executed is stored in C:\ as *resize_image.xml* and the *axcpgridnode.exe* executable running.

To open the Command Prompt click **Run** in the **Start** menu, type **cmd** and then click **OK**

To successfully launch the Rule Executor it is necessary also to specify the rule to be applied. For example, if the Rule Executor executable (*axruleexecutor.exe*) is in the path C:\AXCPGridNode\ and the rule is in c:\resize_image.xml the command syntax is the following:

```
C:\>C:\AXCPGridNode\axruleexecutor c:\resize_image.xml
```

During the rule execution also the Rule Executor window shows output messages.

```

C:\>C:\axcpgridnode\axruleexecutor c:\resize_image.xml
*****
* Welcome on AXMEDIS AXCP Rule Executor *
*****

Initializing AXCP Executor...
Running AXCP Rule...
Return: undefined

C:\>

```

11 AXMEDIS Programme & Publication Editor (UNIVLEEDS)

The P&P Editor is a GUI, which interacts with AXMEDIS Query Support to make selections from the Query results in order to schedule some programmes (e.g. day, week, month, and year) with the following programme specifications (rules):

- WHAT: the AXMEDIS object of interest
- WHERE: destination channel, where to publish e.g. iTV or kiosk or other, and “where” profile
- WHEN: date, time, slot, duration
- HOW: direct transfer, reference or require formatting engine

The representation of the above programmes rules is represented using XML.

11.1 Main functionalities

The Programme and Publication Programme Editor provides the following functionality:

- Create: a Programme Manager uses a GUI to create some P&P Programmes using the Query Support User Interface to browse the AXMEDIS database, to select, to schedule, and to return with a list of relevant objects
- Edit: a Programme Manager make changes to the programme rules selected from a list in the GUI read from the P&P Programme Repository. Changes can also include using Query Support to browse the AXMEDIS database to select and to return with a list of relevant objects
- Save: send the P&P Programme to a P&P Repository for archiving (configured local file)
- Test: Test the P&P Programme through the P&P Engine
- Activate: send the P&P Programme to the P&P Engine

Functionalities presented in the P&P Editor User Manual are sectioned with the following sections:

- Creating a P&P Programme
- Loading an Existing P&P Programme
- Editing a P&P Programme
- Querying for AXMEDIS Objects in the P&P Editor
- Testing a complete P&P Programme
- Activating and Stopping a P&P Programme
- Configuring the P&P Editor

11.2 Relationship with other tools

The P&P Editor communicates with the P&P Engine either directly or using AXMEDIS Workflow and with the AXMEDIS Database (AXDB) using Query Support as shown in the following Figure.

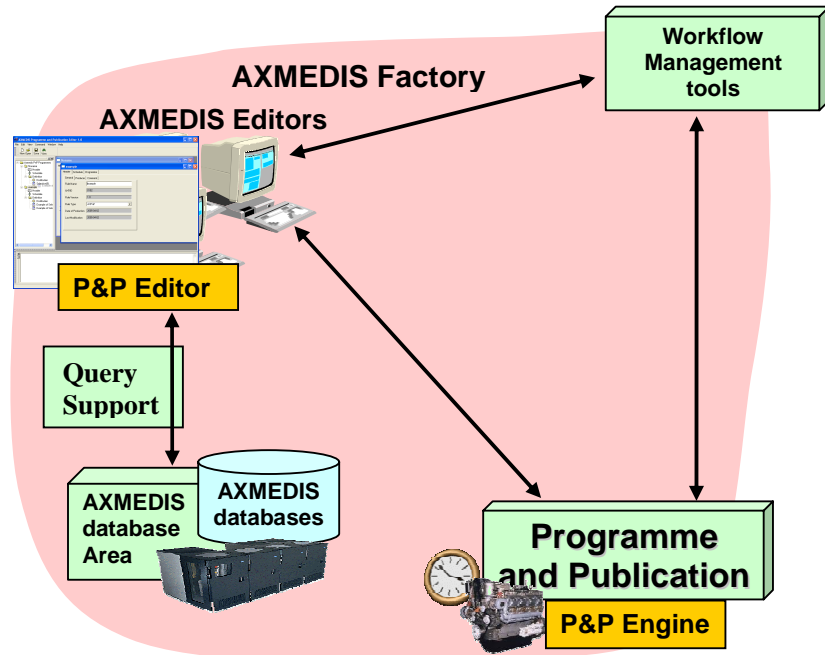


Figure: Relation between the P&P Editor and other AXMEDIS tools

When the actor is required to add a new object to the P&P Programme for scheduling, the actor opens the Query Support Dialog window in the P&P Editor and queries for an object by adding details to the query text fields and drop down selections. The results are returned to the Dialog and objects selected are added to the P&P Programme.

On activation or test, the P&P Editor connects to the P&P Engine (through AXMEDIS Workflow if it has been activated) and sends a completed P&P Programme for processing by the P&P Engine.

11.3 Detailed description of the functionalities and Screenshots

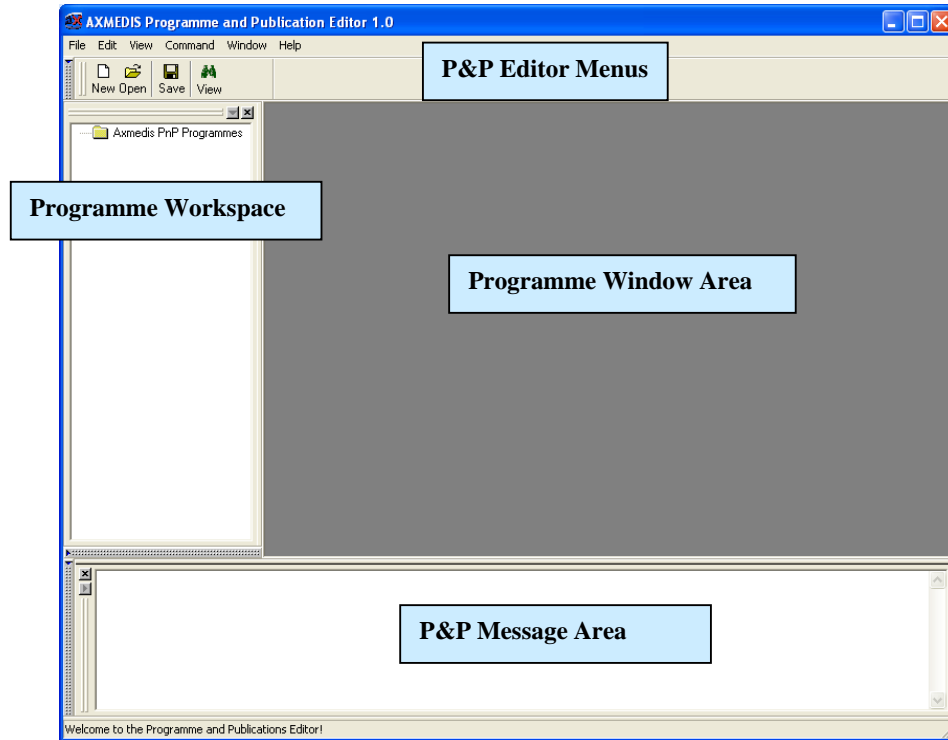


Figure: Initial P&P Editor View

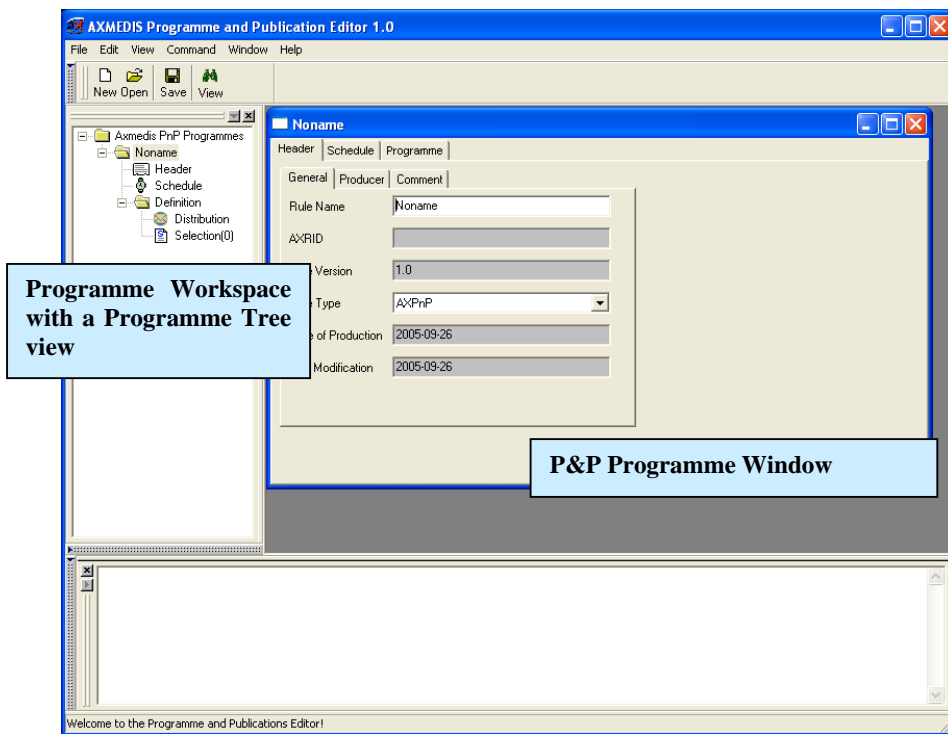


Figure: P&P Editor Displaying Programme Window and Workspace with Tree View if the AXML Programme

11.3.1 Creating a P&P Programme

To create a P&P Programme, the programme manager starts the P&P Editor and from the opening screen creates a new P&P Programme by selecting “New” from the tool bar or using the File menu (File → New) or using the keyboard short cut “Ctrl-N” (see Figure 1). On requesting a new programme, the dialog box pops up and the programme managers enters the name of the draft P&P Programme and selects “OK” (see Figure 2).

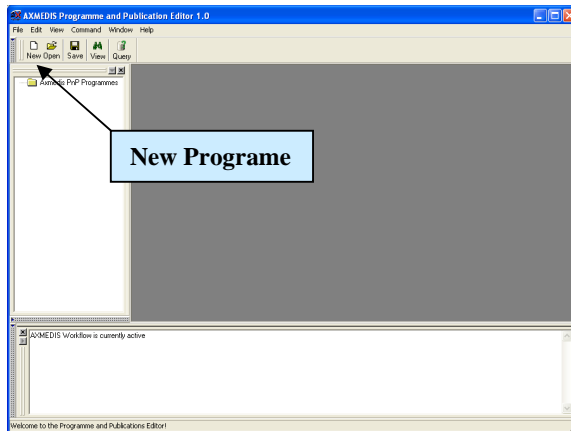


Figure 1

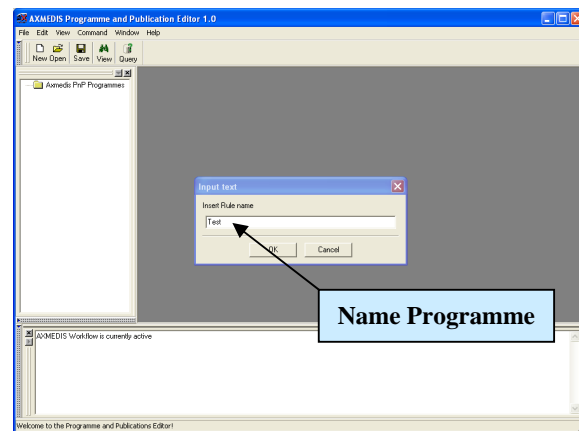


Figure 2

After the P&P Programme name has been OK'd, the new P&P Programme is ready for editing with the 'tree view' used as a workspace and the P&P Programme window for editing the programme details and making a programme schedule (see Figure 3).

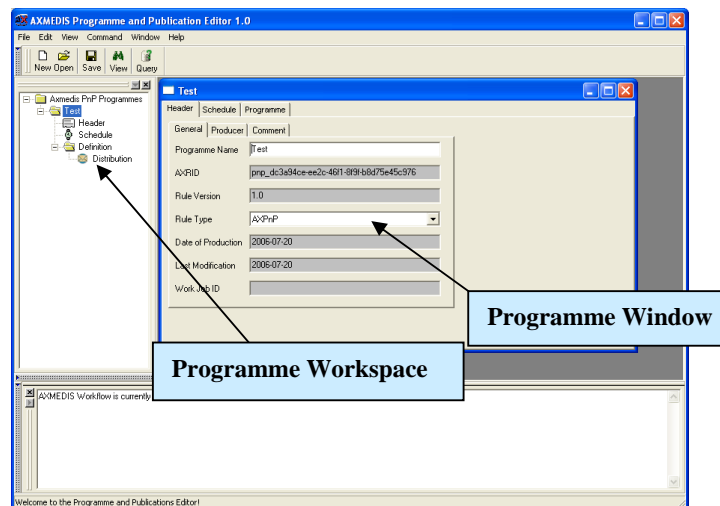


Figure 3

11.3.2 Loading an Existing P&P Programme

The programme manager may wish to edit an existing draft programme that has already been saved. The programme manager may have saved the P&P Programme as a file on the local system or in the P&P Programme Repository. By selecting 'Open', (see Figure 4 and 5) or 'P&P Programme Repository', (see Figure 5 and 6), the programme manager can load an existing draft programme from either the P&P Programme Repository or elsewhere on the local system.

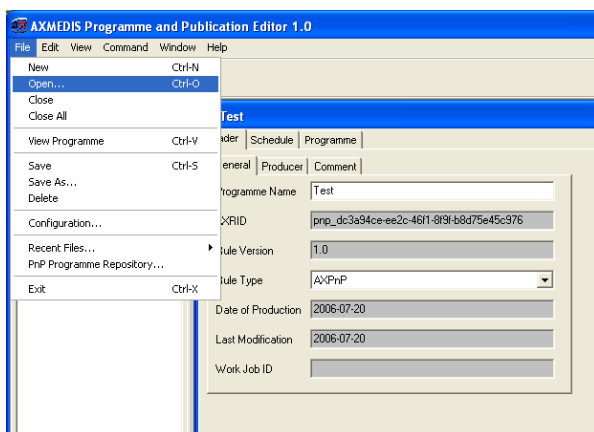


Figure 4

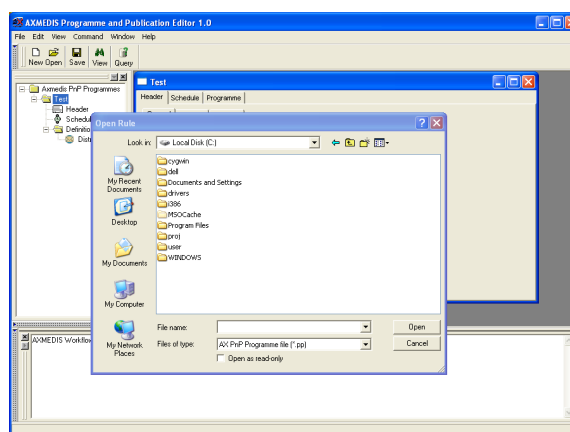


Figure 5

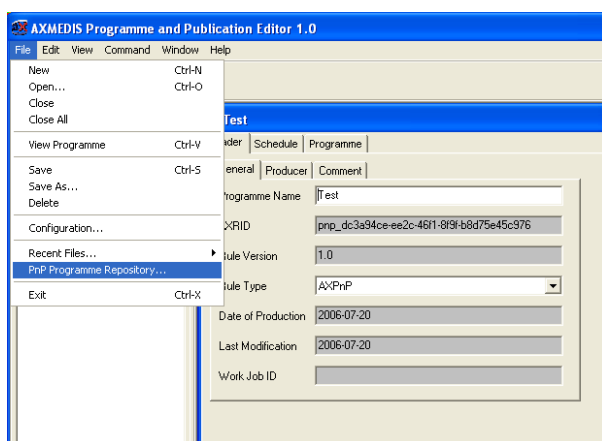


Figure 6

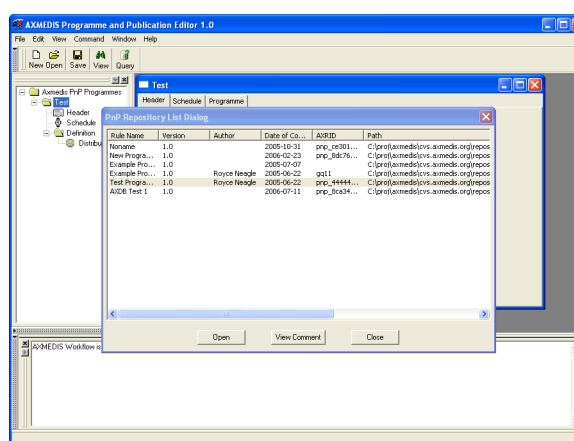


Figure 7

11.3.3 Editing a P&P Programme

The programme manager can use the editor to edit multiple P&P Programmes (as shown in Figure 8), and edit P&P Programme data such as general, producer and content information; and set the specifications for delivering a multimedia object. This includes setting the channel and terminal as well as distribution date and time.

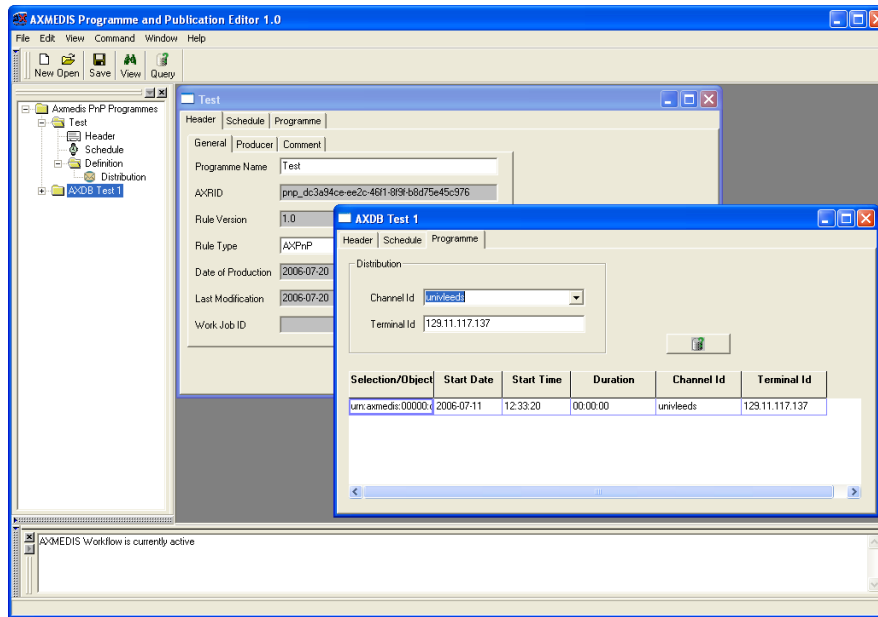


Figure 8

Editing functionalities include being able to drag and drop information from one P&P Programme to another using the workspace tree area as shown in the tree view in Figure 9 and the popup box in Figure 10.

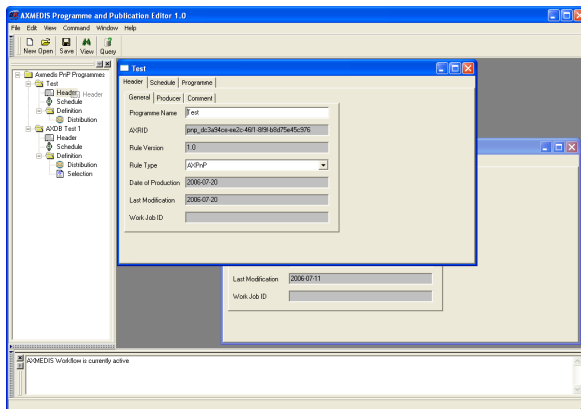


Figure 9

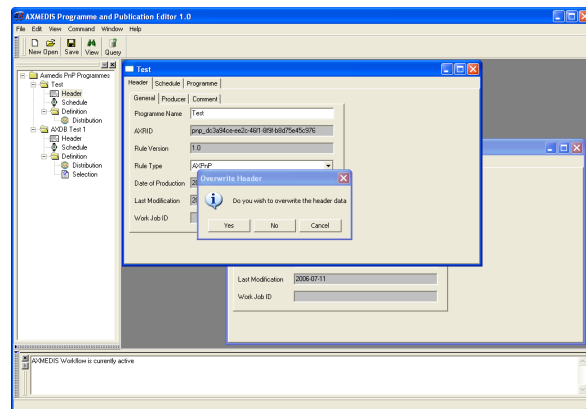


Figure 10

11.3.4 Querying for AXMEDIS Objects in the P&P Editor

The P&P Editor has integrated the Query Selection Dialog to select AXMEDIS Objects to load into the P&P Programme. This is accessed by clicking on the Query button in the “Programme View” in the Programme Window or in the Tool Bar, as highlighted in Figure 11

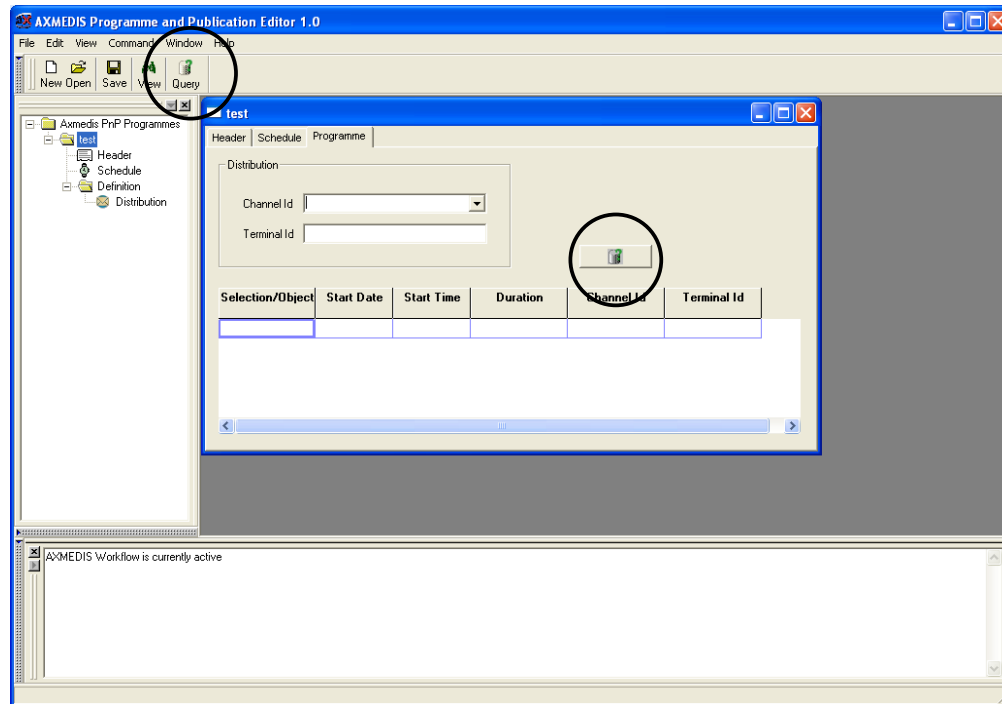


Figure 11

In Figure 12 we see the Dialog Query view to create a search query to find AXMEDIS objects and in Figure 13 the result of a query. One or more objects can be selected in the results view and by clicking OK adds the objects to the P&P Programme

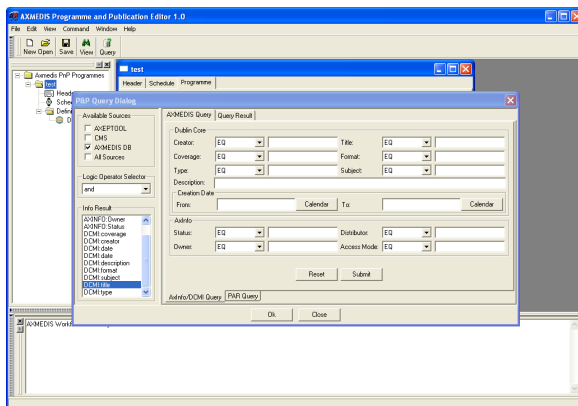


Figure 12

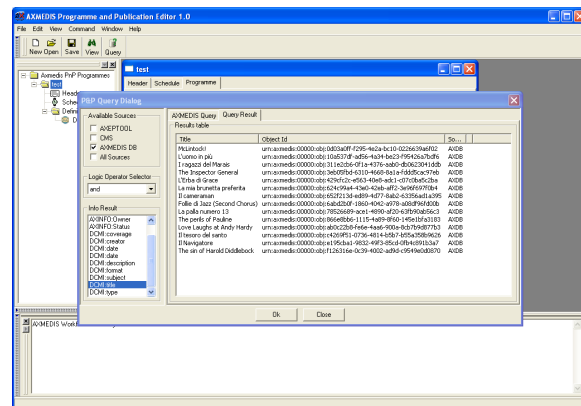


Figure 13

11.3.5 Testing a complete P&P Programme

After the creation and editing of a complete P&P Programme ready for distribution, the programme manager has two options: the first is to run a quick trial; this ensures the programme devised is valid without any of the processing taking place in the P&P Engine (see Figure 14). This means request are made to AXMEDIS tools to make sure tasks can be accomplished for processing the P&P Programme. A message is return to the P&P Editor informing the user if the trial was successful or not. The second option is to run a full trial which enforces the P&P engine to request all the processing to be run if required with the exception of the final distribution to the distribution area. The processing jobs are requested and executed and on completion the results are returned. The final action of distribution is the only processing not requested in the full trial.

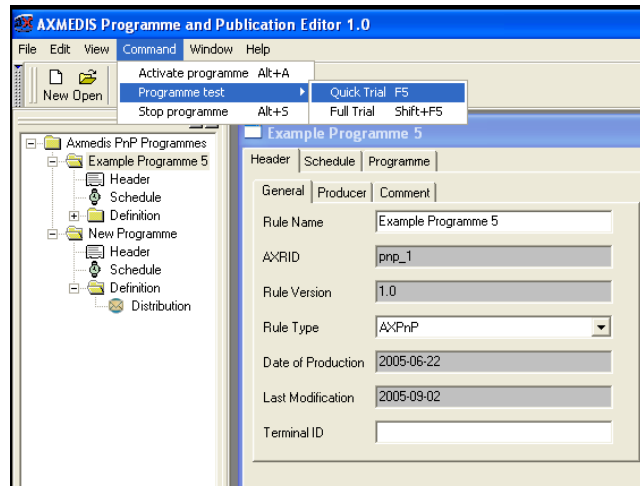


Figure 14

11.3.6 Activating and Stopping a P&P Programme

A complete P&P Programme to distribute AXMEDIS objects is activated in the Command menu (see Figure 14). This sends the P&P Programme to the P&P Engine to process the programme information, request formatting if required for the specified distribution channel and at the specified times distributes the correctly adapted multimedia object to the distribution channel. During the lifespan of the active programme, the programme manager can also stop a P&P Programme from the P&P Editor by selecting “*Stop Programme*” in the Command menu as shown in Figure 14.

11.3.7 Configuring the P&P Editor

The P&P Editor has relationships with other modules within the AXMEDIS project and therefore the P&P Editor is required to be configured. This is achieved by opening the Configuration dialog using the File menu as shown in Figure 15 and Figure 16.

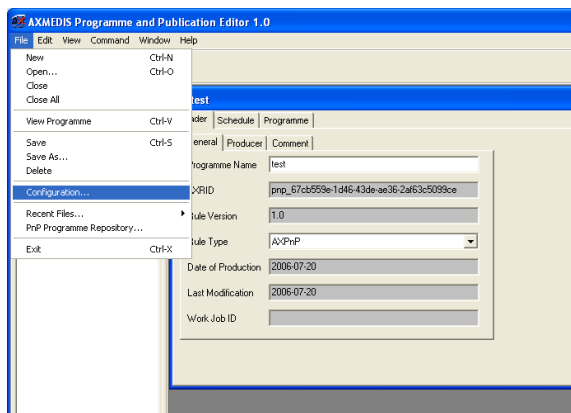


Figure 15

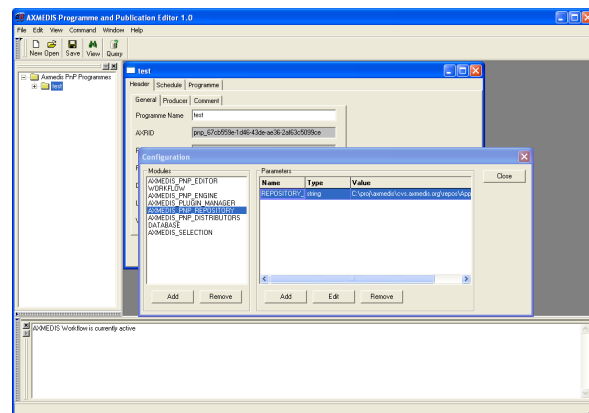


Figure 16

The main configuration parameters required are the settings to communicate with the other AXMEDIS modules and tools and to define the Distribution Channels which are sent to the P&P Engine. The following are the parameters that are required to be set:

- Path to the schemas used for parsing the P&P Programmes (AXMEDIS_PNP_EDITOR → XML_XSD_PATH): This is currently in the same directory as the executable of the P&P Editor.
- Address to the P&P Engine (AXMEDIS_PNP_ENGINE → SERVER_ADDRESS): The URI needs to be set with the port number 3000 (e.g. localhost:3000)

- Location of the P&P Programme Repository (AXMEDIS_PNP_REPOSITORY → REPOSITORY_PATH): This can be defined anywhere on the local system.
- Address to the AXMEDIS Query Support (AXMEDIS_SELECTION → MAIN_QUERY_SUPPORT_WSDL)
- AXMEDIS Workflow: To configure this parameter, please see the manual related to Workflow (See section 14 and 15). If this is not set, then all communications go directly to the modules and not through Workflow.
 - The Plug-in path for AXMEDIS Plug-ins (AXMEDIS_PLUGIN_MANAGER → PLUGINS_PATH)
 - The Workflow Gateway URL if the editor is to be connected to the AXMEDIS Workflow (WORKFLOW → gatewayUrl)
- And add the distribution channels to the P&P Editor (AXMEDIS_PNP_DISTRIBUTOR → <YOUR_CHANNEL>)
 - Name of Channel (e.g. EUTELSAT) as the parameter
 - with values <Name to display>|<URI of the distribution channel> (e.g. test|localhost)

12 AXMEDIS Programme & Publication Engine (UNIVLEEDS)

The Programme and Publication Engine will be developed exploiting the work performed for the Publication tool in WP4.4. This will allow the reception of specific commands (requests) for creating content produced by exploiting the capabilities of the AXMEDIS formatting engine. In addition, the Programme and Publication Engine will also have the capabilities for the publication the Programme based on the specific programmes.

The active engine is continuous running software accessing the system clock to process a list of programmes, which consists of “rules” to make available AXMEDIS objects to the specified destination channels at the correct time, taking into account the transfer and/or formatting (if required) time. This is achieved by the input of *activated P&P Programme* for scheduled distribution.

12.1 Main functionalities

The active Programme and Publications Engine’s main function is to continually run looking for active publication rules and make the objects in the rules available for distribution. The main points to consider:

- Access to correct system clock
- Keep a track of newly activated P&P Programmes to add to the delivery system
- The API to the AXMEDIS formatting engine (AXCP) to request an appropriate format for distribution and retrieving the correct object from the AXDB
- Providing the AXMEDIS objects to a AXMEDIS distribution server allowing for delivery time

12.2 Relationship with other tools

The P&P Engine is integrated with AXMEDIS Database (AXDB) and P&P Editor and the AXMEDIS Content Processing Engine (AXCP). Communication to the AXMEDIS tools can be performed with or without AXMEDIS Workflow. The relationship between the AXMEDIS tools and the P&P Engine can be seen in the Figure below.

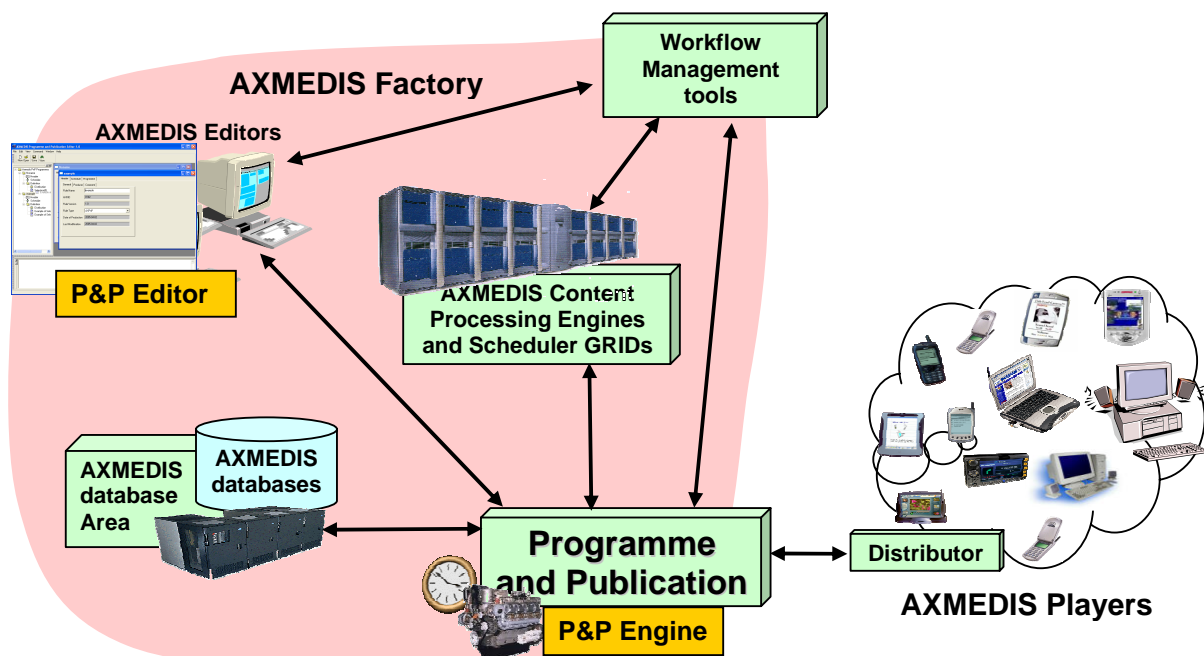
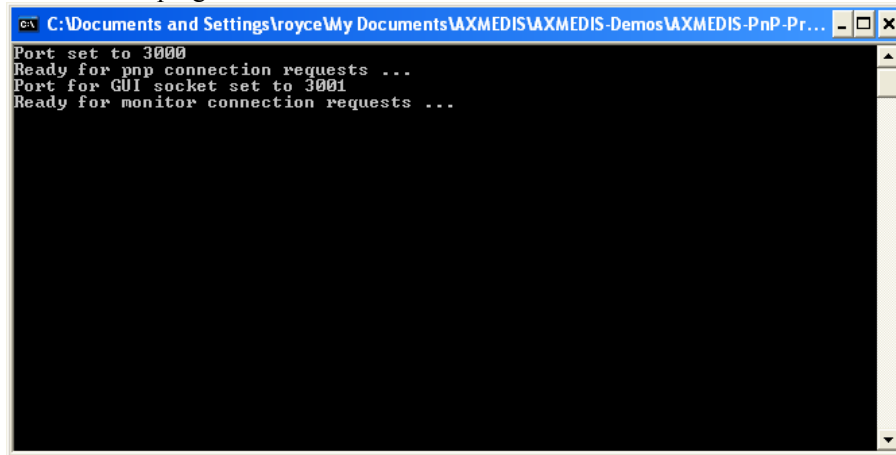


Figure: Relation between the P&P Engine and other AXMEDIS tools

12.3 Detailed description of the functionalities and Screenshots

The P&P Engine is a continuous running engine processing P&P Programmes from the P&P Editor or On-Demand either directly or through WF. When set running, the user is presented with a DOS window as shown in the Figure below including the port that the P&P Engine is using to listen to the P&P Editor (in this example 3000) and the port to listen to the P&P Engine Monitor (3001). Other communications are handled using the Workflow SOAP plugin.



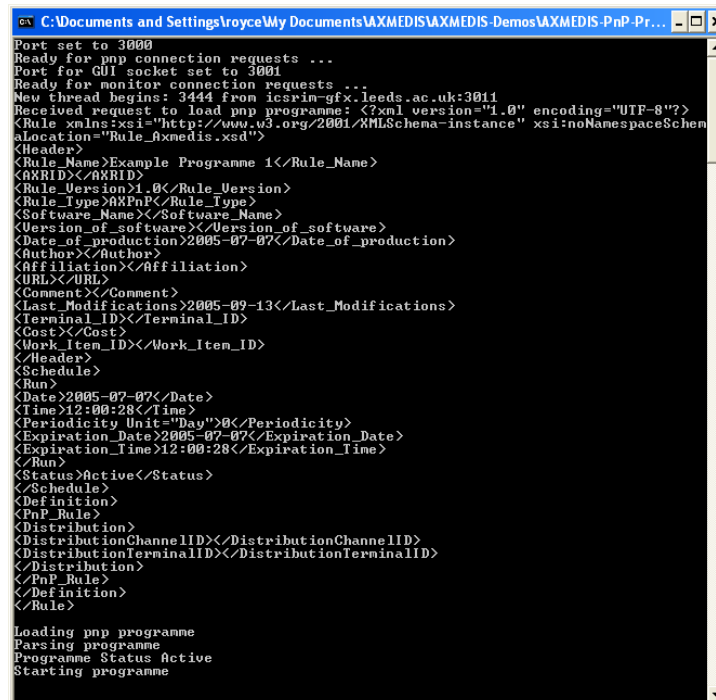
```

C:\Documents and Settings\royce\My Documents\AXMEDIS\AXMEDIS-Demos\AXMEDIS-PnP-Pr...
Port set to 3000
Ready for pnp connection requests ...
Port for GUI socket set to 3001
Ready for monitor connection requests ...

```

Figure: DOS Prompt with the initiated P&P Engine

The DOS prompt informs the user when P&P Programmes have been activated by displaying the parsed programme and starting the distribution of a multimedia object defined in the programme at the specified time. This can be seen in the figure below.



```

C:\Documents and Settings\royce\My Documents\AXMEDIS\AXMEDIS-Demos\AXMEDIS-PnP-Pr...
Port set to 3000
Ready for pnp connection requests ...
Port for GUI socket set to 3001
Ready for monitor connection requests ...
New thread begins: 3444 from icsr-in-gfx.leeds.ac.uk:3011
Received request to load pnp programme: <?xml version="1.0" encoding="UTF-8"?>
<Rule xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="Rule_Axmedis.xsd">
  <Header>
    <Rule_Name>Example Programme 1</Rule_Name>
    <AXRID></AXRID>
    <Rule_Version>1.0</Rule_Version>
    <Rule_Type>AXPnP</Rule_Type>
    <Software_Name></Software_Name>
    <Version_of_software></Version_of_software>
    <Date_of_production>2005-07-07</Date_of_production>
    <Author></Author>
    <Affiliation></Affiliation>
    <URL></URL>
    <Comment></Comment>
    <Last_Modifications>2005-09-13</Last_Modifications>
    <Terminal_ID></Terminal_ID>
    <Cost></Cost>
    <Work_Item_ID></Work_Item_ID>
  </Header>
  <Schedule>
    <Run>
      <Date>2005-07-07</Date>
      <Time>12:00:28</Time>
      <Periodicity_Unit>"Day">0</Periodicity>
      <Expiration_Date>2005-07-07</Expiration_Date>
      <Expiration_Time>12:00:28</Expiration_Time>
    </Run>
    <Status>Active</Status>
  </Schedule>
  <Definition>
    <Distribution>
      <DistributionChannelID></DistributionChannelID>
      <DistributionTerminalID></DistributionTerminalID>
    </Distribution>
    <PnP_Rule>
      </Definition>
    </Rule>
  </Rule>
</Rule>
Loading pnp programme
Parsing programme
Programme Status Active
Starting programme

```

Figure: P&P Engine parsing and starting a programme to distribute and AXMEDIS object

13 AXMEDIS Programme & Publication Engine Monitor (UNIVLEEDS)

13.1 Main functionalities

The main functionality of the P&P Engine Monitor is to provide the user with information concerning the status of the P&P Engine and the status of the activated P&P Programmes being processed by the P&P Engine. It can also be used to send an abort request directly to the P&P Engine to remove a particular programme from the engine.

13.2 Relationship with other tools

The P&P Engine Monitor is a viewing management tool for the P&P Engine. It communicates to the P&P Engine requesting status information concerning the engine and active P&P Programmes currently being processed by the P&P Engine.

13.3 Detailed description of the functionalities and Screenshots

13.3.1 Running the Monitor

The first step when running the P&P Engine Monitor is to connect to the Engine. Select connect from the action menu and then enter the address and port number. Once a connection has been established the status tab will show that a connection has been made and a list of all active programmes will be displayed in the programme management tab.

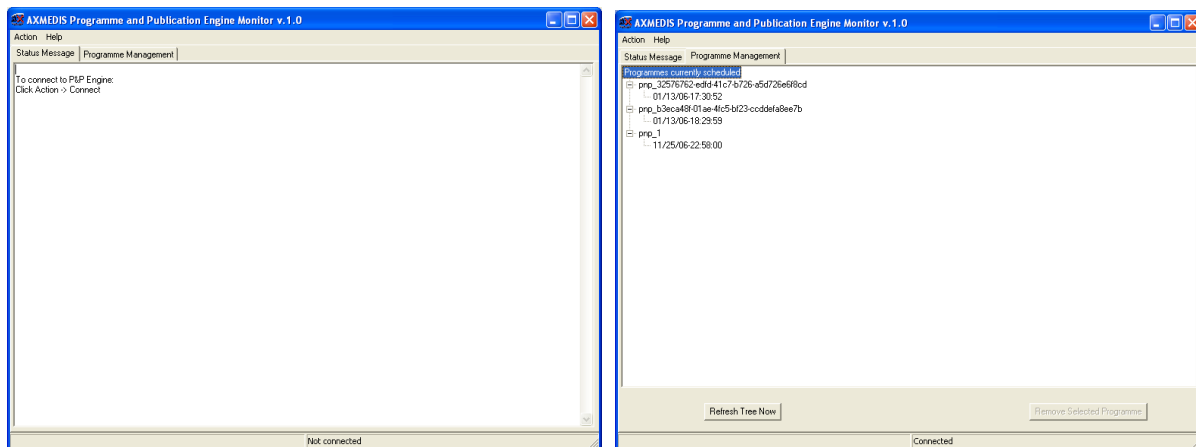


Figure: P&P Engine Monitor with Status messages view (left screen shot) and P&P Programme Management View (right screenshot)

To delete an active programme from the engine the user must first select a programme. Then click on the 'remove selected programme' button. There is a small delay while the engine wakes up the programme and destroys it before the window is refreshed and the programme is removed.

13.3.2 Status Information

The status information tab is used to output all logging information (connection/disconnection/updates to programme list) and errors (connection errors etc.) to the user.

13.3.3 Programme Managements

The programme management tab displays information about programmes active on the P&P Engine. This information consists of programme IDs and the scheduled distribution time for the next object within the programme to be distributed. It also contains the 'remove selected programme' button which allows a user to delete an active programme from the engine.

14 AXMEDIS Workflow and Workflow manager - Openflow (IRC)

14.1 Main functionalities

- Openflow is a workflow engine developed and released as free software under a GNU GPL licence.
- It is based on an object-oriented structure and has a powerful exception handling system along with dynamic redesign support.
- These features make OpenFlow much more flexible than any other existing workflow engines.
- OpenFlow supports most of the open standards (XML/XML-RPC) including also the web standards.
- It has got a simple access to most of the relational DBMSs and thus it facilitates the integration of heterogeneous system.
- OpenFlow is activity-based, web-based, WFMC inspired, built and integrated with the application server Zope.
- OpenFlow is capable of running on most operating systems including Linux, Windows 9x, NT/2000, XP, MacOS.
- Through an integrated role assignment system, OpenFlow can assign tasks and activities to single users or workgroups and also to automatic applications.
- At every moment OpenFlow can trace the complete history of a certain situation e.g. participants involved, activities and actions executed and invoked.
- It is possible to carry out performance and efficiency analysis and verify the correct implementation of the adopted model.
- OpenFlow is strongly web-oriented and offers complete support for developing and executing workflows via a browser.
- The interaction with OpenFlow uses simple HTTP requests as in, for example, process modelling, assignment of users to activities, definition of the interaction with the applications.
- Every user receives his task which interacts with appropriate applications through the network

14.2 Relationship with other tools

The Openflow Engine should be deployed on Zope Server. Additionally the user should import the Workflow Adaptors (Prove_WF.zexp and extensions) in the Zope Server. The openflow Engine communicates to AXMEDIS tools through Workflow Gateways (Request & Response) which should be deployed on IIS server with .NET V2 or later.

14.3 Detailed description of the functionalities and Screenshots

Openflow runs on the Zope platform which is managed through the “Zope Management Interface” using industry standard browsers, typically by logging on as the administrator (admin) at URL <http://localhost:8080/manage>. The screen shot below shows an example of this management interface.

Creating a new process in openflow is a multi-step process which begins with adding an OpenFlow container using the Zope management interface as shown below (delineated by a an ellipse in red).

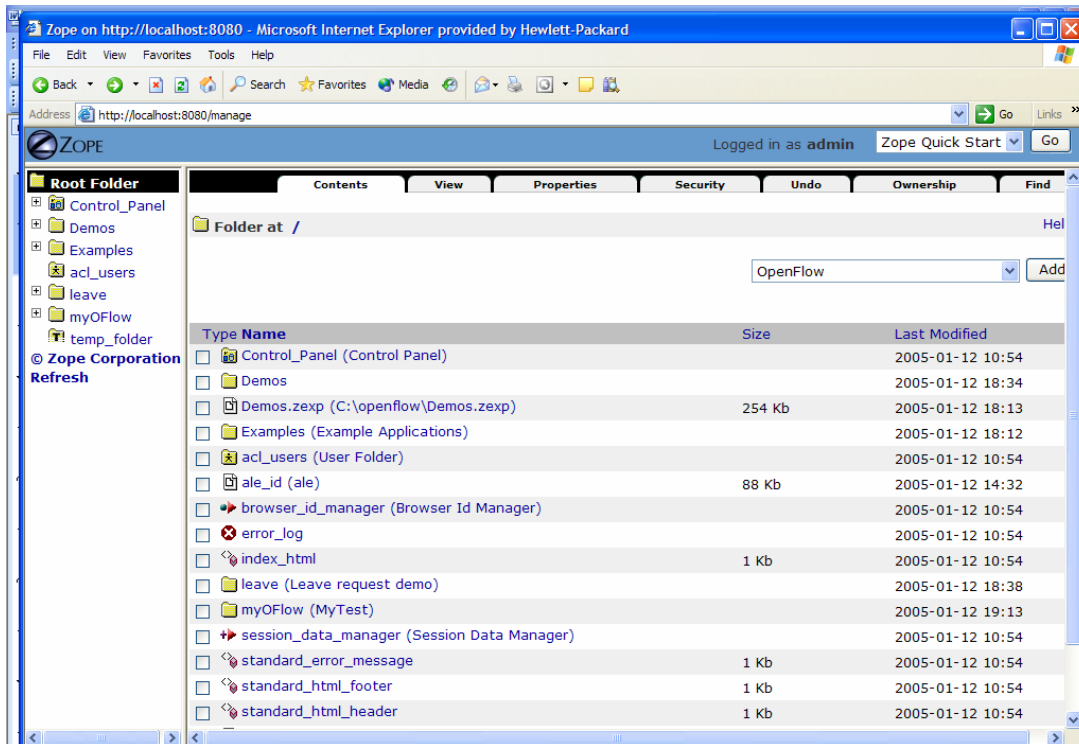


Figure 1: Adding an OpenFlow container through the Zope Management Interface

During the creation of the OpenFlow container, the name of the container must be specified as shown in the next screen-shot.

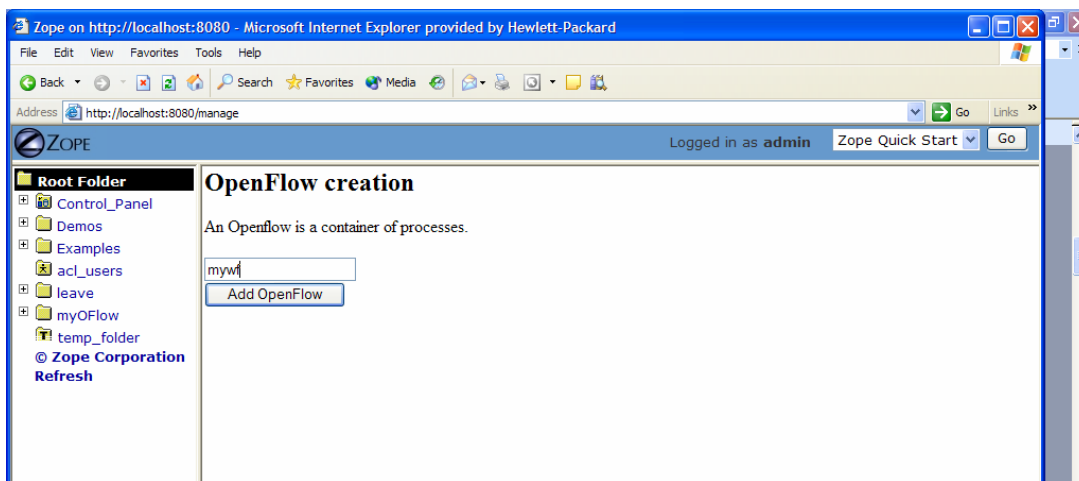


Figure 2: Creating the OpenFlow container

Next it is necessary to define the process and the activities pertaining to the process, together with their transitions (From Activity and To Activity). These operations are performed by accessing the tabs in the Openflow container as shown in the following screen-shots:

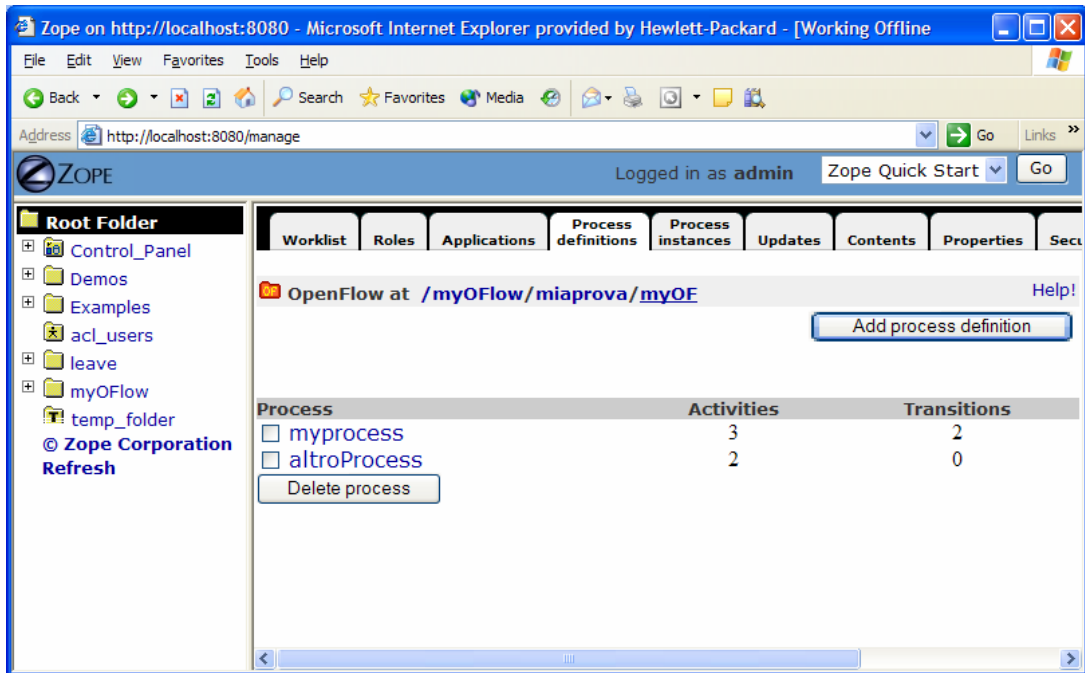


Figure 3: The process definition tab

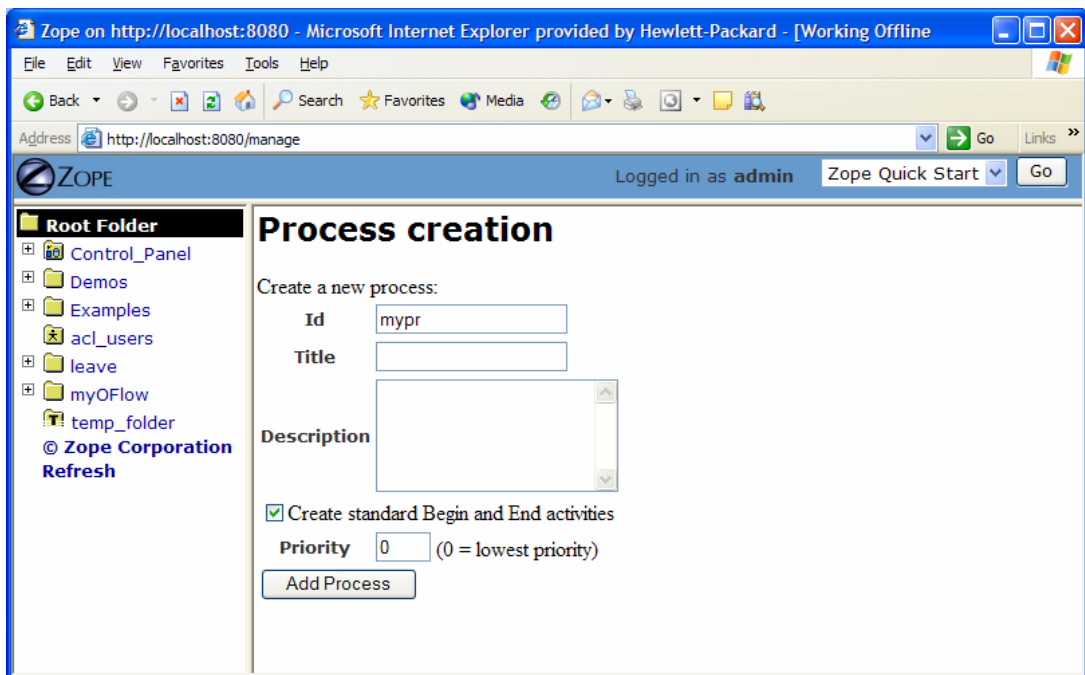


Figure 4: Creating a new Process definition

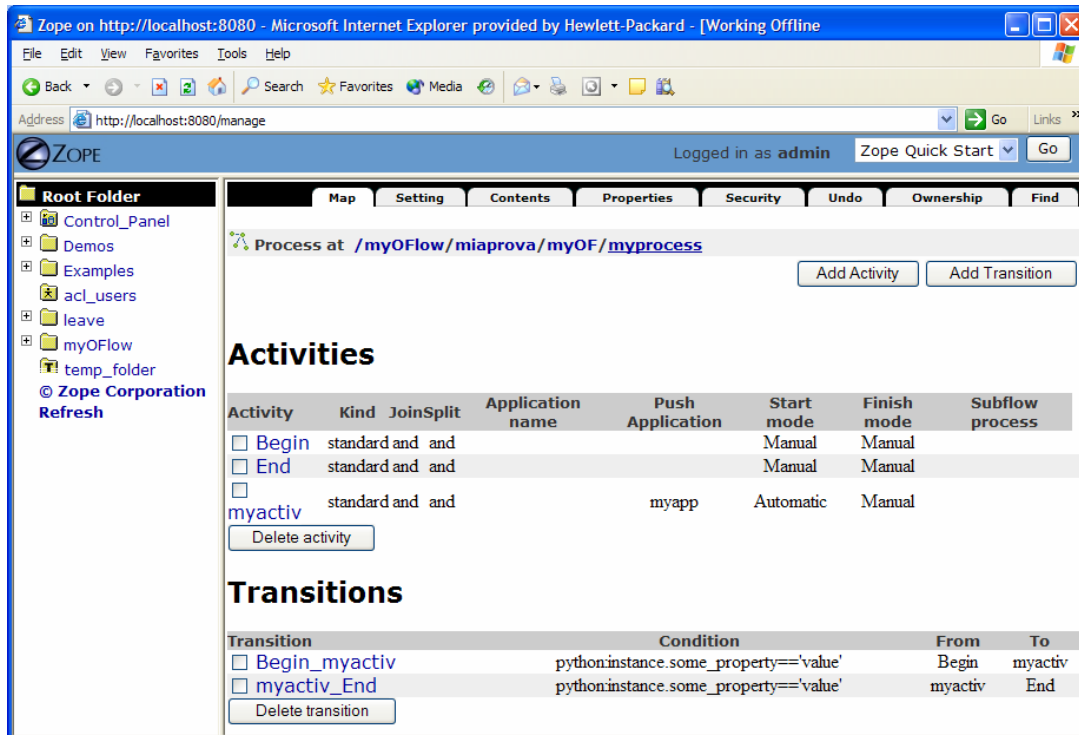


Figure 5: Management of activity and transitions of a process

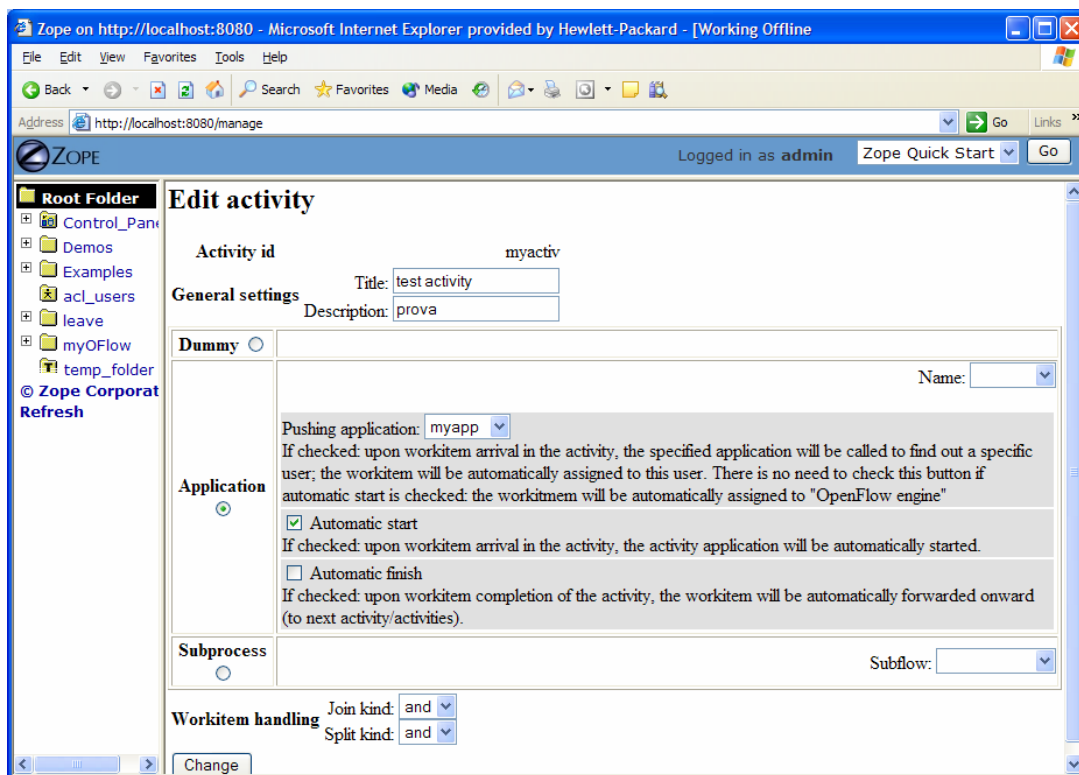


Figure 6: Editing a process activity

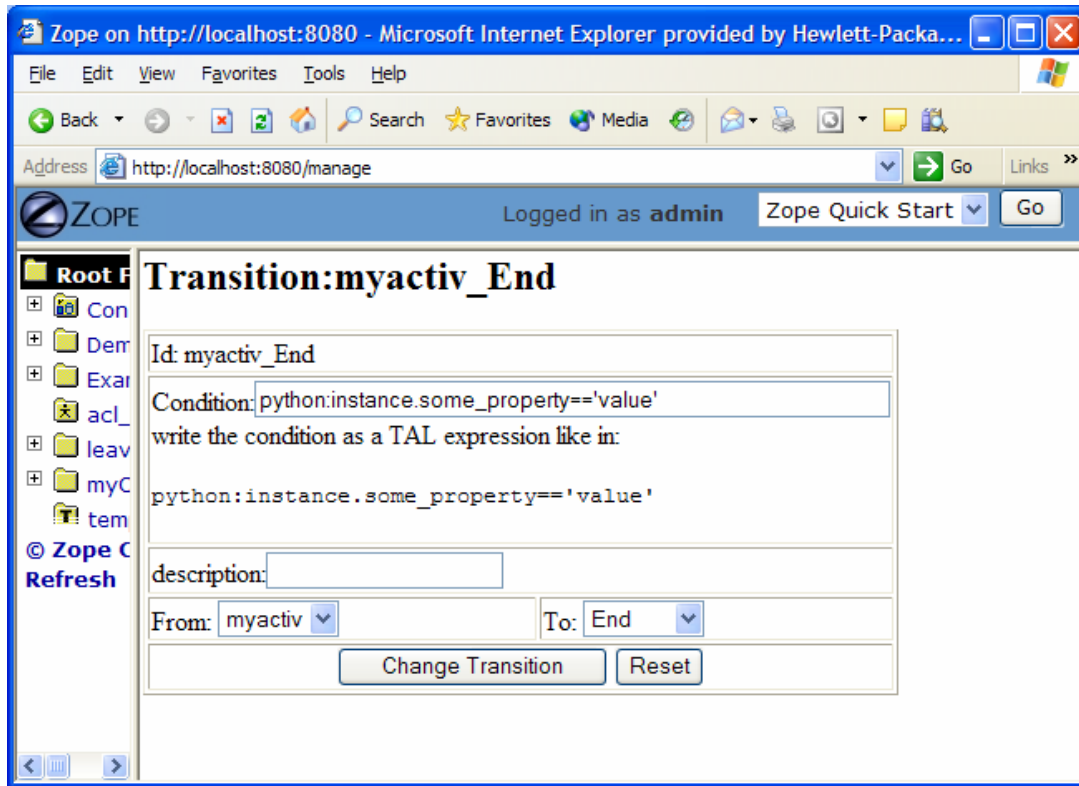


Figure 7: Defining process transition and related conditions

Applications associated to the activities are then specified selecting the Applications Tab.

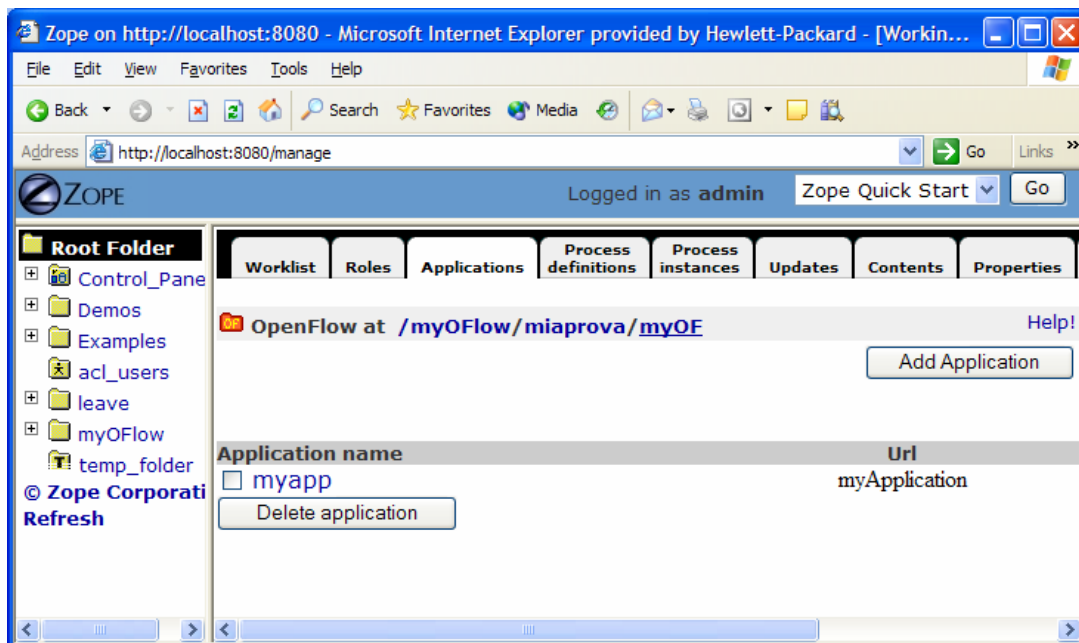


Figure 8: Defining process applications

The users and roles are configured as Zope users and roles as access control list (acl_users).

Once a process has been defined it can be tested. An instance of the process can be created and executed directly in the processflow-instance management tab shown below.

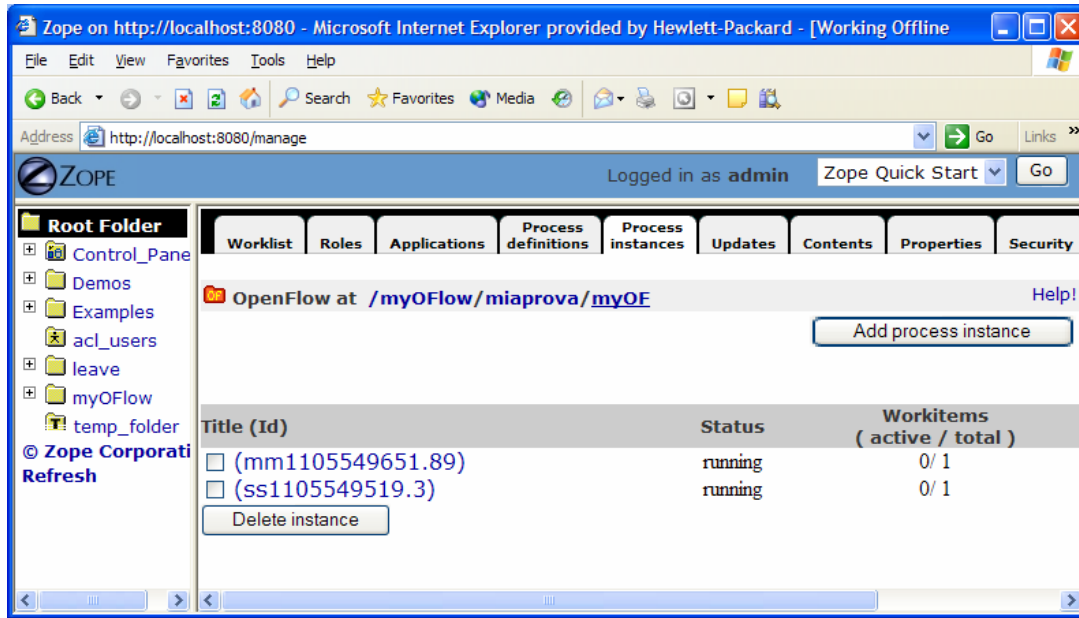


Figure 9: Process instance management tab

The following Figure shows the of the workitems involved in the process instance that has been created.

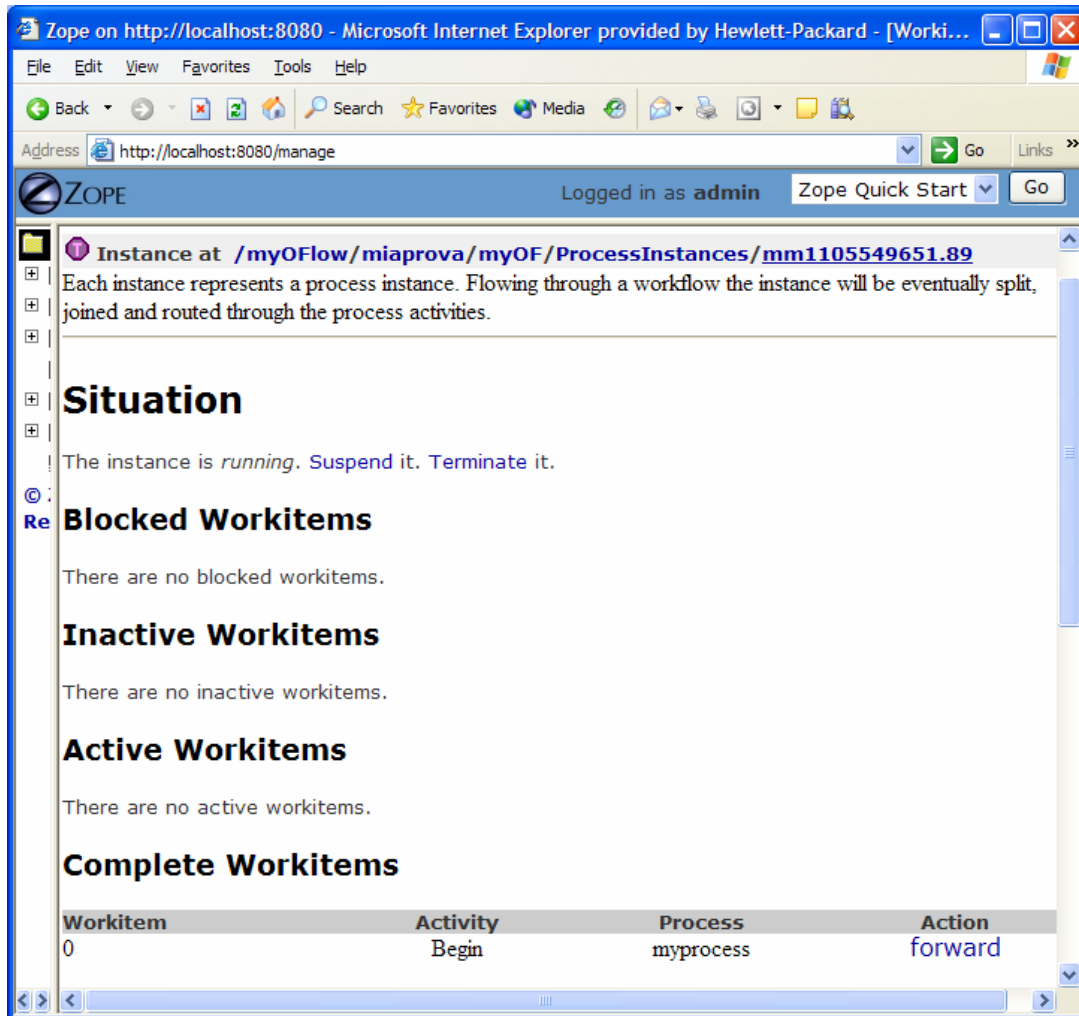


Figure 10: Monitoring and management of a specific process instance

Process Example:

The following simple example illustrates a process to request a AXMEDIS object manipulation (*a mock-up process*). This is an example of explicit forwarding to different actors having different roles. The first actor requests the creation of a new AXMEDIS object by filling out a form. The request goes to the second actor (called Socius) who checks that the request is acceptable. The request is then forwarded to the third actor (called Prefectus) for approval.

The following steps are necessary for the above example process to be enacted:

The first actor (called Tertius) enters an AXMEDIS object manipulation request by filling out the following form as shown in the screen-shot below:

Leave request demo - Microsoft Internet Explorer provided by Hewlett-Packard

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media

Address http://localhost:8080/leave/leave_startform Go Links

Request for AXMEDIS object manipulation

Start date End date

Type

Reason

Figure 11: Tertius' AXMEDIS object manipulation form

According to the processflow, the request goes to the next actor (called Socius). When Socius logs in, his work list shows that there is a workitem in his worklist as shown in the screen-shot below:

Leave request demo - Microsoft Internet Explorer provided by Hewlett-Packard

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print Mail News RSS Feeds

Address http://localhost:8080/leave/mywork Go Links

Worklist for socius

Activity	Instance	Status	Actor	Action
Begin	Request of AXMEDIS object manipulation by tertius	inactive	socius	Activate

[To frontpage](#)

Figure 12: Socius' worklist and workitem activation

To execute the workitem, the actor (Socius) has to activate the workitem (Begin) and perform the related activities. Next this actor either forwards the workitem to the next actor, which in this case is the supervisor (called Prefectus), or rejects the request; as illustrated by the screen-shot below:

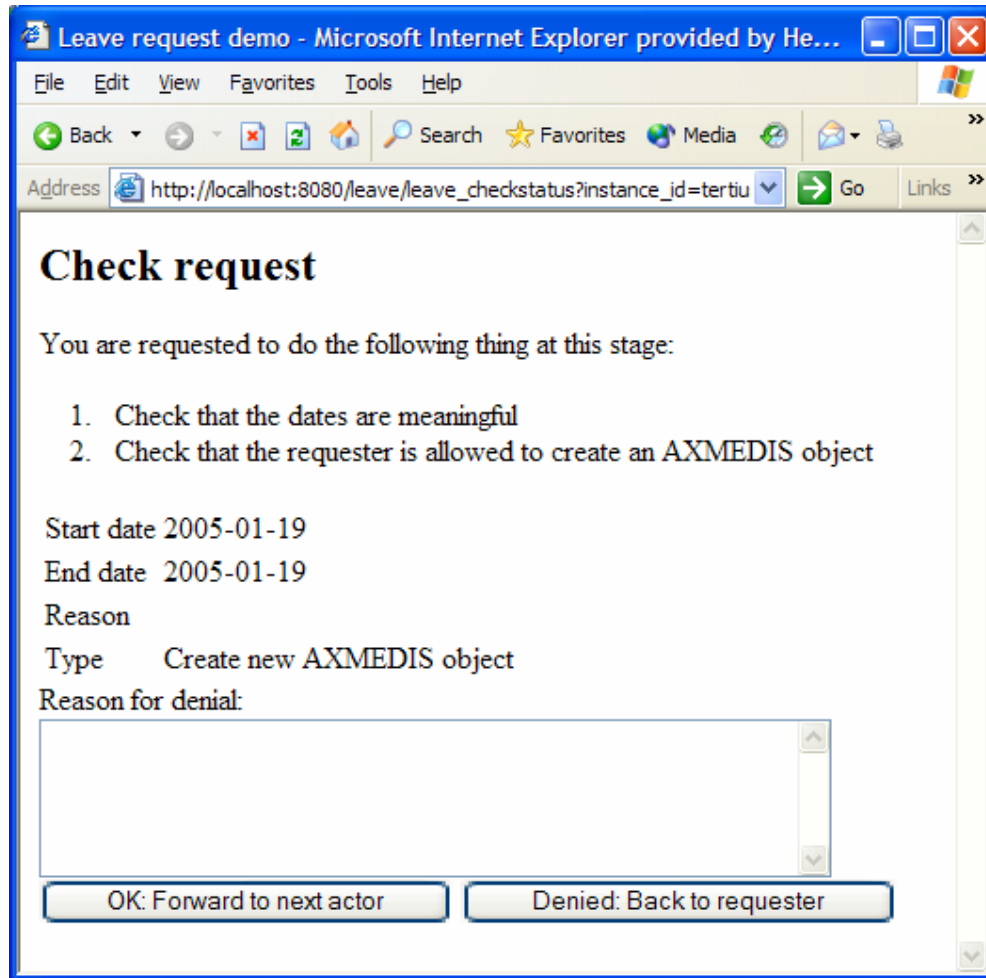


Figure 13: Socius' workitem execution and forwarding

Then the activity is forwarded to the last actor and the process ends.

15 AXMEDIS Workflow and Workflow manager – Microsoft Biztalk (IRC)

15.1 Main functionalities

- BizTalk Server 2004, an integration server, lets you to develop, deploy, and manage integrated business processes and XML-based Web services
- Traditionally, BizTalk Server has been used for application integration, where the following two scenarios are most important:
 - Connecting applications within a single organization, commonly referred to as enterprise application integration (EAI)
 - Connecting applications in different organizations, often called business-to-business (B2B) integration
- BizTalk Server 2004 also adds technology for creating Human Workflow Services (HWS), making possible business processes that people can interact with from Microsoft Outlook and other familiar clients
- The HWS infrastructure is accessed through Web services, and so it can be used by any client application

15.2 Relationship with other tools

The Microsoft Biztalk Server Communicates to AXMEDIS Tools using Workflow Plug-ins directly without the need of Workflow Gateways.

15.3 Detailed description of the functionalities and Screenshots

Please refer to <http://www.microsoft.com/biztalk/default.mspx> for detailed description of functionality and usage for Microsoft Biztalk Server.

16 AXMEDIS PLUGINS

16.1 Audio Adaptation Plugin (EPFL)

16.1.1 Main functionalities

The `audio_adaptation_plug-in` allows adapting audio content to various use cases. For example, it can be used for transcoding applications to transform a high-quality audio file into a low bit rate audio file well suited for distribution on a network with reduced bandwidth. This document shows the main functionalities provided by a first prototype of the tool forming a minimal user guide.

16.1.2 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.

16.1.3 Detailed description of the functionalities and Screenshots

16.1.3.1 FFMPEG

Here's an example on how to use the FFmpeg audio adaptation transcoding function as a plug-in with for the AXMEDIS editor.

The plug-in must be applied on an audio resource of an AXMEDIS object. The adaptation plug-in is called by right-clicking on the interesting resource and selecting the 'Plugin...' command:

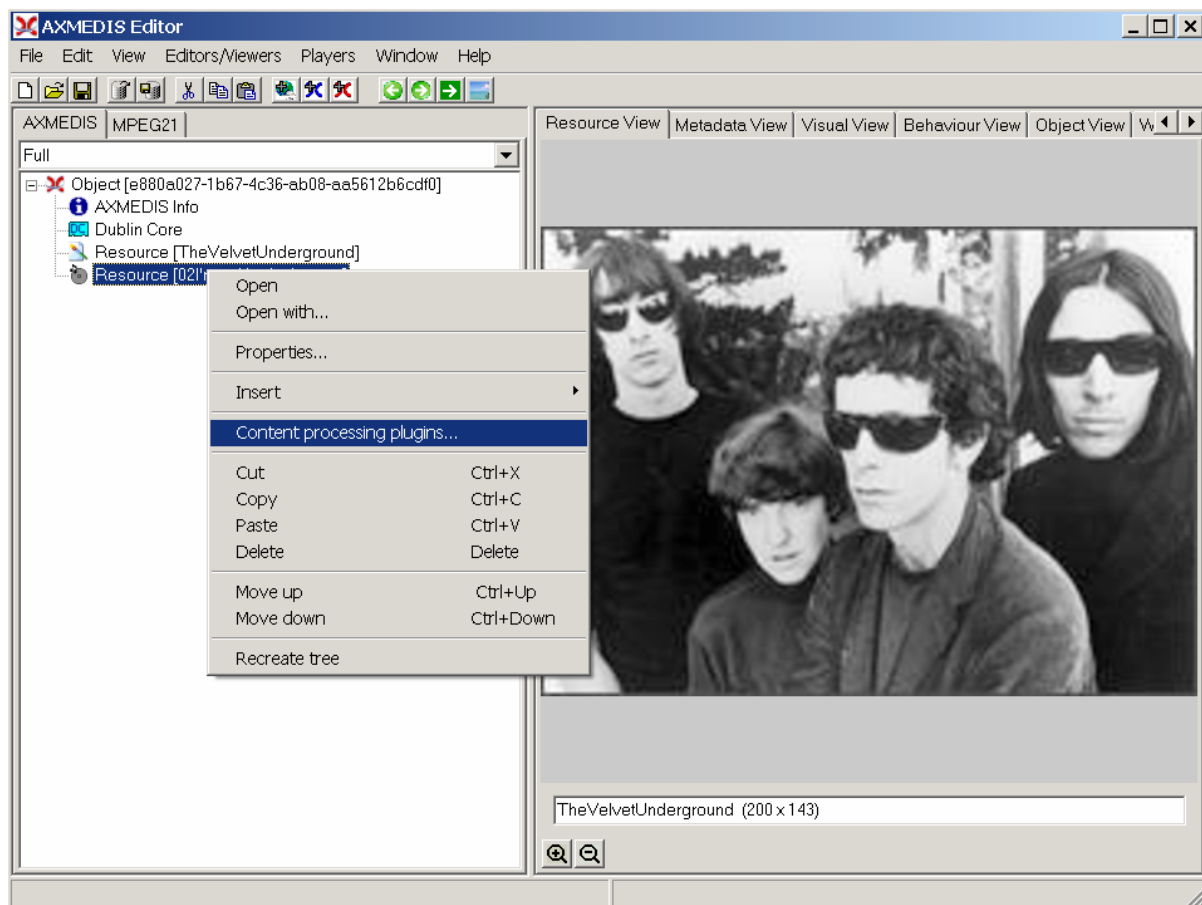


Figure 1 – Calling the content processing plug-ins

A window showing the functionalities available for the kind of resource selected appears:

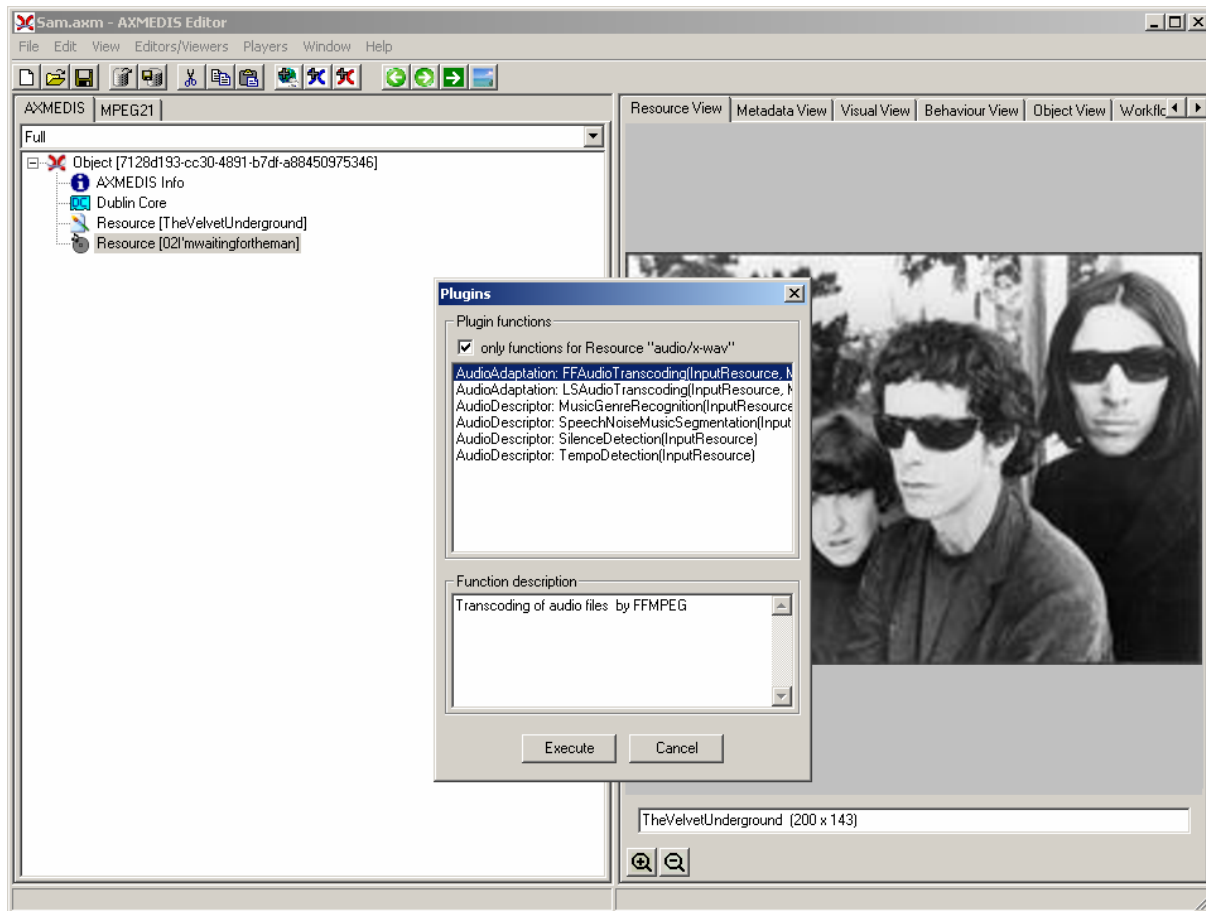


Figure 2 – Selecting the FFMPEG audio transcoding function

The first audio adaptation function available is the FFmpeg transcoding function which is selected by clicking on FFAudioAdaptation: FFAudioTranscoding. A new window appears showing the interface to the audio transcoding function. In the example of the following figure, the transcoding function is used to create a 10 second snapshot with reduced bit rate of the input audio file:

Mp3 compression is selected with a bit rate of 64 kB (which corresponds to a low quality)

Further bit rate reduction is achieved by using a lower sampling rate for the output (22050 Hz) and mixing audio channels into a single mono channel

Only a portion of 10 seconds of the input resource is selected (starting at time 10 seconds and ending at time 20 seconds)

A snapshot with reduced bit rate is particularly useful to allow a customer to pre-view an item before purchasing the corresponding high quality object.

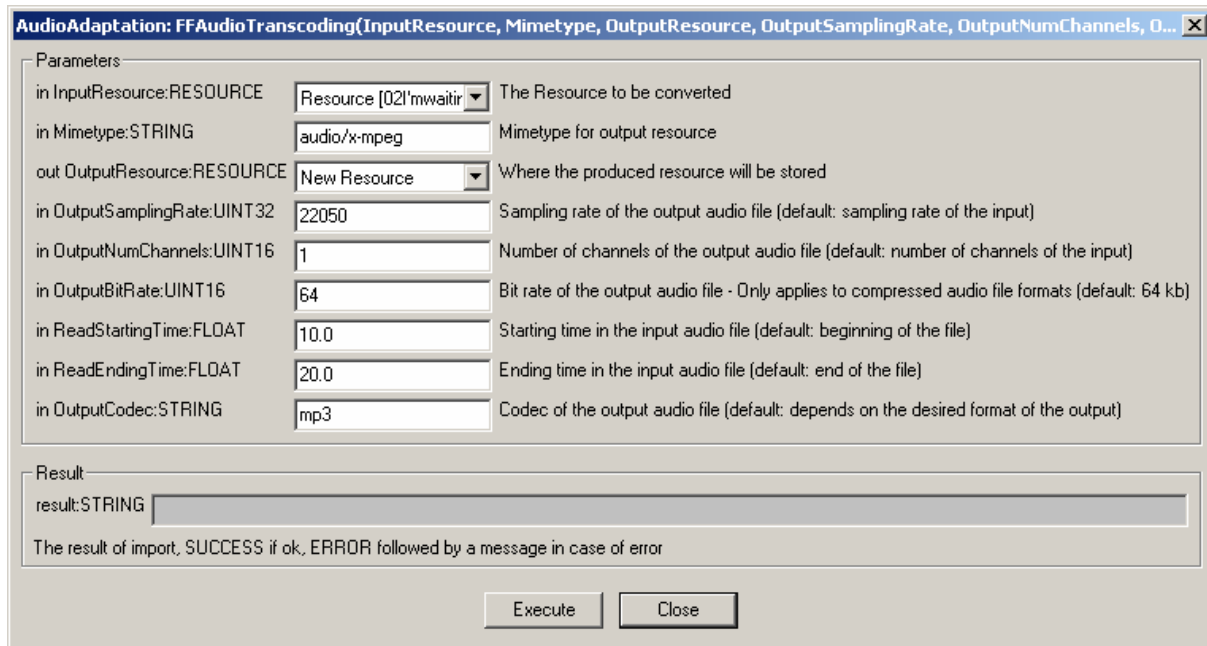


Figure 3 – The FFMPEG audio transcoding function

Here follows a more complete description of the parameters of the FFMPEG audio transcoding function:

InputResource

Description: the resource to be converted

Parameter Type: AxResource

Default Value:

Constraints:

Resource Type: audio

Resource Format: x-mpeg (.mp3), x-aiff (.aif, .aiff), x-wav (.wav), basic (.au, .snd), x-ms-wma (.wma), x-vorbis (.ogg), x-pn-realaudio (.ra, .ram)

Ranges:

MimeType

Description: MimeType for the output resource

Parameter Type: string

Default Value:

Constraints:

Resource Type: audio

Resource Format: x-mpeg, x-aiff, x-wav, basic, x-vorbis, x-ac3

Ranges:

OutputResource

Description: Where the output resource will be stored

Parameter Type: AxResource

Default Value:

Constraints:

Range:

OutputSamplingRate

Description: The sampling rate of the output resource in Hertz

Parameter Type: uint32

Default Value: by default, the sampling rate of the input resource is used

Constraints:

Range:

OutputNumChannels

Description: The number of channels of the audio resource after transcoding

Parameter Type: uint16

Default Value: by default, the number of channels of the input resource is used

Constraints:

Range:

OutputBitRate

Description: The bit rate of the audio resource after transcoding in kilo-Bytes (this parameter is used when transcoding towards a compressed audio format such as MP3)

Parameter Type: uint16

Default Value: by default, the bit rate is set to 64 kB

Constraints:

Range:

ReadStartingTime

Description: set the beginning of the output resource to ReadStartingTime seconds from the beginning of the input resource

Parameter Type: float

Default Value: by default, the read starting time is set to 0 seconds which means that the input resource is considered from the beginning

Constraints:

Range:

ReadEndingTime

Description: set the end of the output resource at ReadEndingTime seconds from the beginning of the input resource

Parameter Type: float

Default Value: by default, the read ending time is set to the end of the input resource

Constraints:

Range:

OutputCodec

Description: set the codec of the output resource; depending on the mime type selected for the output resource, only a certain subset of codec will be supported (the following table shows the possible codecs according to the possible mime types)

Parameter Type: string

Default Value: the default codec depend on the mime type selected for the output resource (the following table shows the default codec according to the possible mime types)

Constraints:

Range:

Result

Result Type: string

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

File Formats

For a list of codecs and formats supported by FFMPEG, please refer to section 34.1.

AXMEDIS

Mime type accepted

audio/x-wav
audio/x-ms-wma
audio/basic
audio/x-mpeg
audio/x-vorbis
audio/x-pn-realaudio
audio/x-ac3
audio/x-dv
audio/x-mace
audio/x-adpcm
audio/x-aac
audio/32KADPCM
audio/amr
video/x-mpeg
video/x-mpeg2
video/mp4
video/x-raw
video/x-h263
video/x-mjpeg
video/x-ms-wmv
video/x-ms-asf
video/x-flv
video/x-svq
video/x-dv
video/x-h264
video/x-indeo
video/x-vp3
video/x-ffv
video/x-vcr
video/x-msvideo
video/x-nut
application/x-pcm
application/vnd.rn-realmedia

16.1.3.2 *libsndfile*

Here's an example on how to use the *libsndfile* audio adaptation transcoding function as a plug-in with for the AXMEDIS editor.

The plug-in must be applied on an audio resource of an AXMEDIS object. The adaptation plug-in is called by right-clicking on the interesting resource and selecting the 'Plugin...' command:

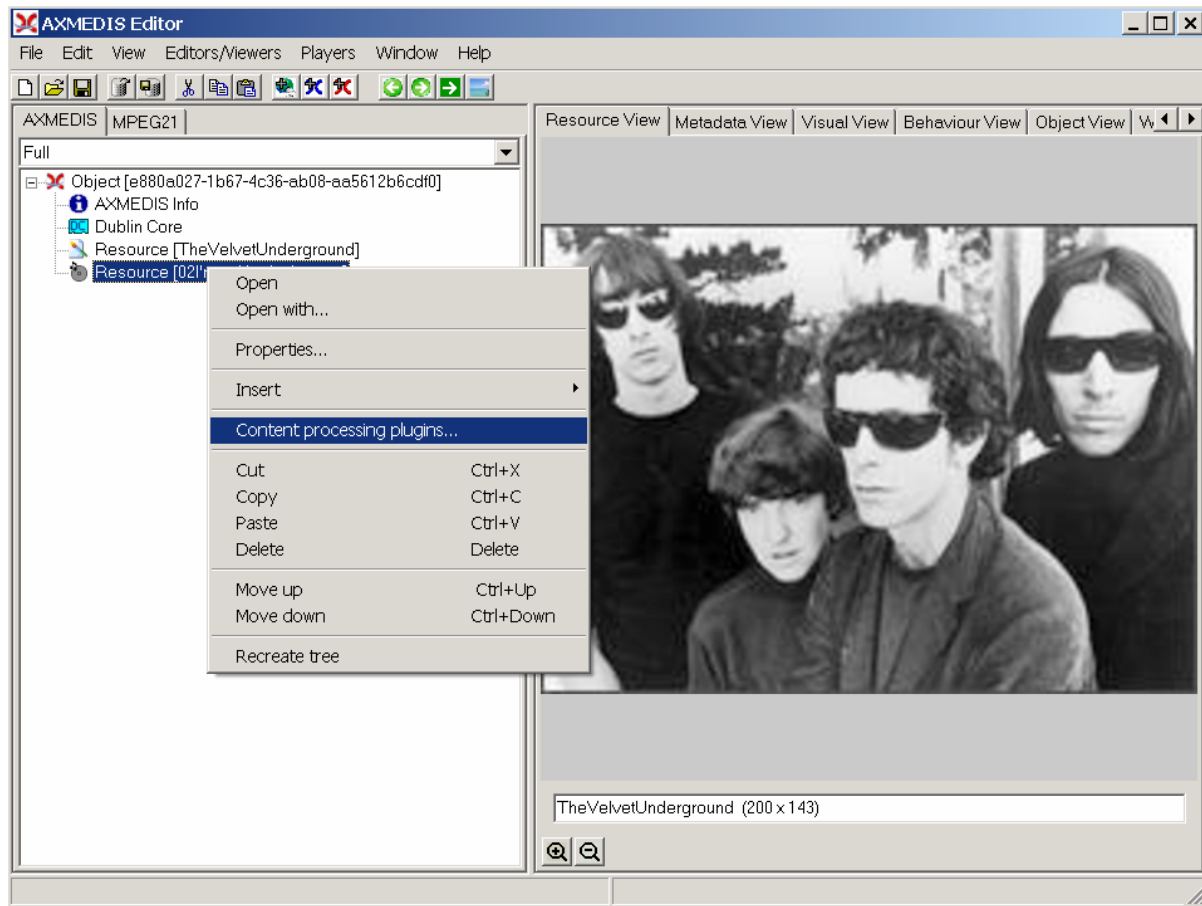


Figure 4 – Calling the content processing plug-ins

A window showing the functionalities available for the kind of resource selected appears:

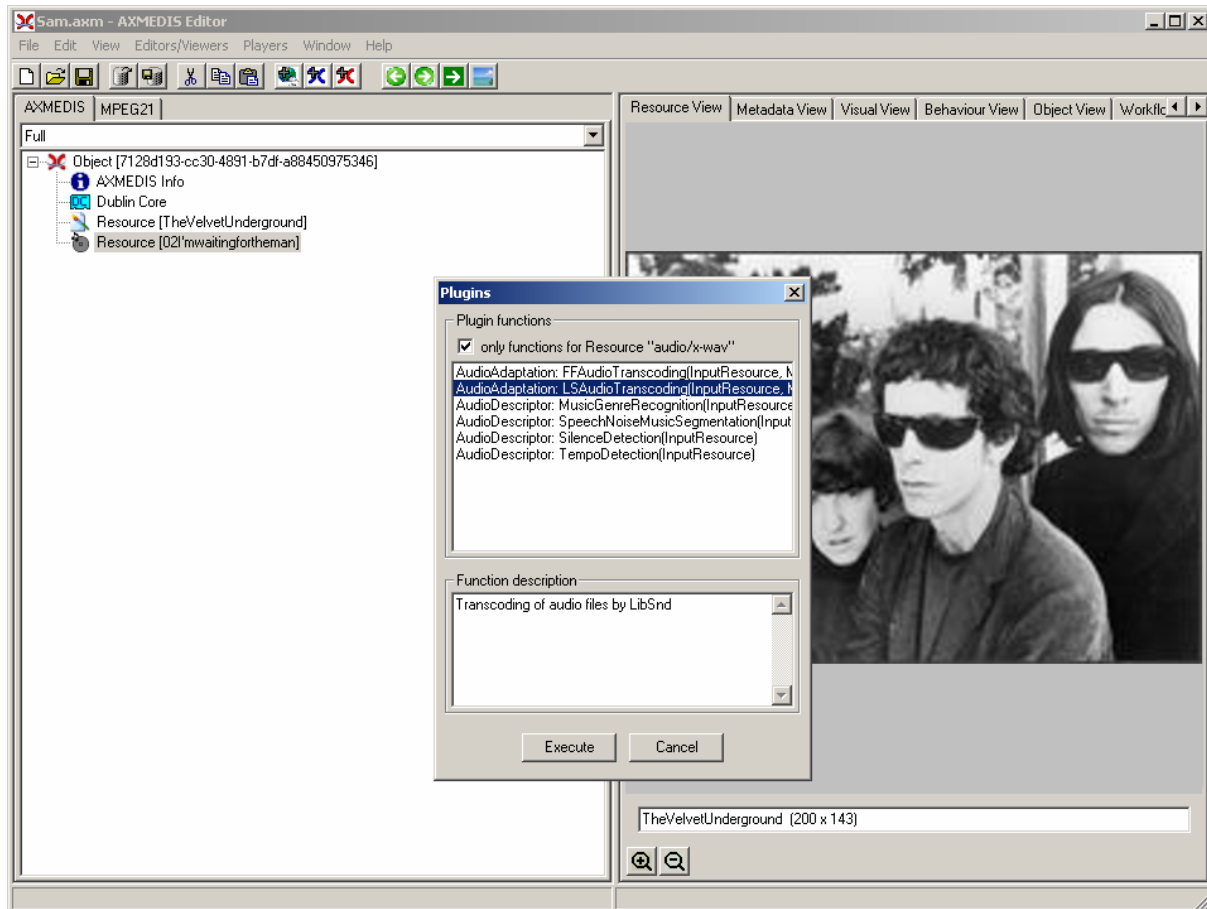


Figure 5 – Selecting the libsndfile audio transcoding function

The first audio adaptation function available is the libsndfile transcoding function which is selected by clicking on LSAudioAdaptation: LSAudioTranscoding. A new window appears showing the interface to the audio transcoding function. In the example of the following figure, the transcoding function is used to create a 10 second snapshot with reduced bit rate of the input audio file:

AIFF format

Only a portion of 8 seconds of the input resource is selected (just the beginning of the sound track)

Such a snapshot could be useful for small audio sampling.

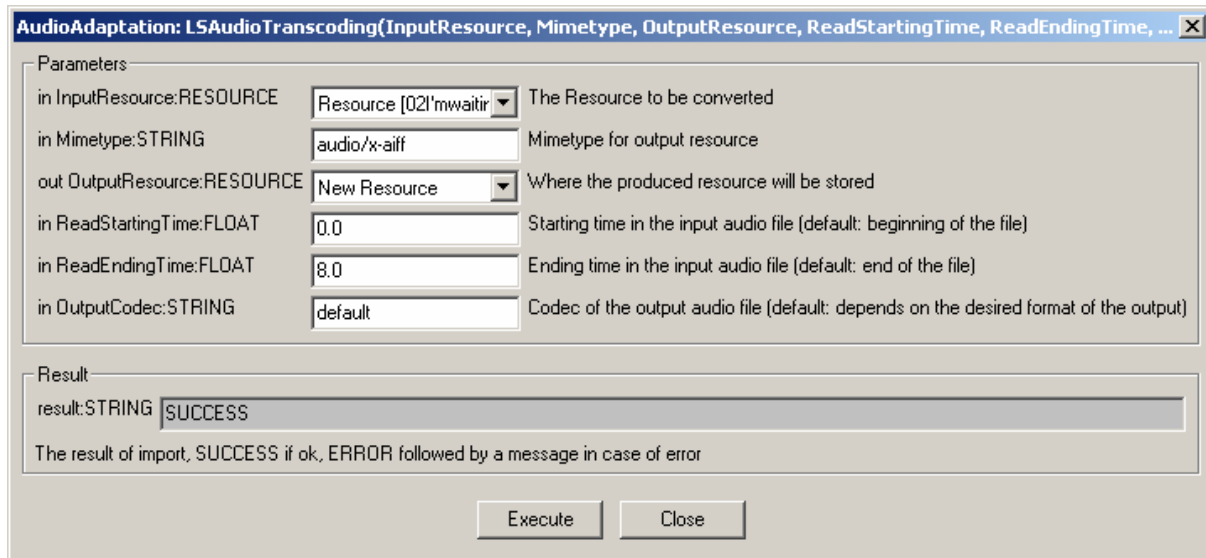


Figure 6 – The libsndfile audio transcoding function

Description: encode an audio file in another format or another codec and change its sample rate and number of audio channels if needed.

Signature:

```
string Transcoding(AxResource InputResource, string MimeType, AxResource OutputResource, float
ReadStartingTime, float ReadEndingTime, string OutputCodec)
```

Parameter List:

InputResource

Description: the resource to be converted

Parameter Type: AxResource

Default Value:

Constraints:

Resource Type: audio

Resource Format: x-mpeg (.mp3), x-aiff (.aif, .aiff), x-wav (.wav), basic (.au, .snd),

x-ms-wma (.wma), x-vorbis (.ogg), x-pn-realaudio (.ra, .ram)

Ranges:

MimeType

Description: MimeType for the output resource

Parameter Type: string

Default Value:

Constraints:

Resource Type: audio

Resource Format: x-mpeg, x-aiff, x-wav, basic, x-vorbis, x-ac3

Ranges:

OutputResource

Description: Where the output resource will be stored

Parameter Type: AxResource

Default Value:

Constraints:

Range:

ReadStartingTime

Description: set the beginning of the output resource to ReadStartingTime seconds from the beginning of the input resource

Parameter Type: float

Default Value: by default, the read starting time is set to 0 seconds which means that the input resource is considered from the beginning

Constraints:

Range:

ReadEndingTime

Description: set the end of the output resource at ReadEndingTime seconds from the beginning of the input resource

Parameter Type: float

Default Value: by default, the read ending time is set to the end of the input resource

Constraints:

Range:

OutputCodec

Description: set the codec of the output resource; depending on the mime type selected for the output resource, only a certain subset of codec will be supported (the following table shows the possible codecs according to the possible mime types)

Parameter Type: string

Default Value: the default codec depend on the mime type selected for the output resource (the following table shows the default codec according to the possible mime types)

Constraints:

Range:

Result

Result Type: string

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

Libsndfile supported types and codecs

For a list of codecs and formats supported by the Libsndfile library, please refer to section 34.5.

Mime Type accepted :

audio/x-wav
audio/x-basic
audio/x-paris
audio/x-svx
audio/x-nist
audio/x-voc
audio/x-ircam
audio/x-w64
audio/x-sd2
audio/x-flac
application/x-pcm
application/x-pagerecall

16.1.3.3 Tricks/Errors

Some tricks for plugin testing :

Don't use the same file for input and output (generates an unknown error, due to the editor ?)

Don't try to use vorbis related files with these actual FMPEG dlls (.ogg)

We endured problems with AMR and it could still fails.

If the reply is that this is Unknown output mime type, check the corresponding mime-type table.

If you don't have to trunk the audio files don't change ReadStartingTime and ReadEndingTime options. (the same caution may be used for other FFMPEG advanced options)

16.2 Audio Descriptor Plugin (EPFL)

16.2.1 Main functionalities

The audiodescriptorplugin aims at extracting automatically metadata from audio signals by audio signal analysis. The functionalities implemented include a segmentation algorithm, a music genre recognizer, a tempo detection algorithm plus a set of low level descriptors extraction algorithms. This document describes these functionalities forming a minimal user guide.

16.2.2 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.

16.2.3 Detailed description of the functionalities and Screenshots

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

The plug-in must be applied on an audio resource of an AXMEDIS object. The descriptor extraction plug-in is called by right-clicking on the interesting resource and selecting the 'Plugin...' command (see figure 1).

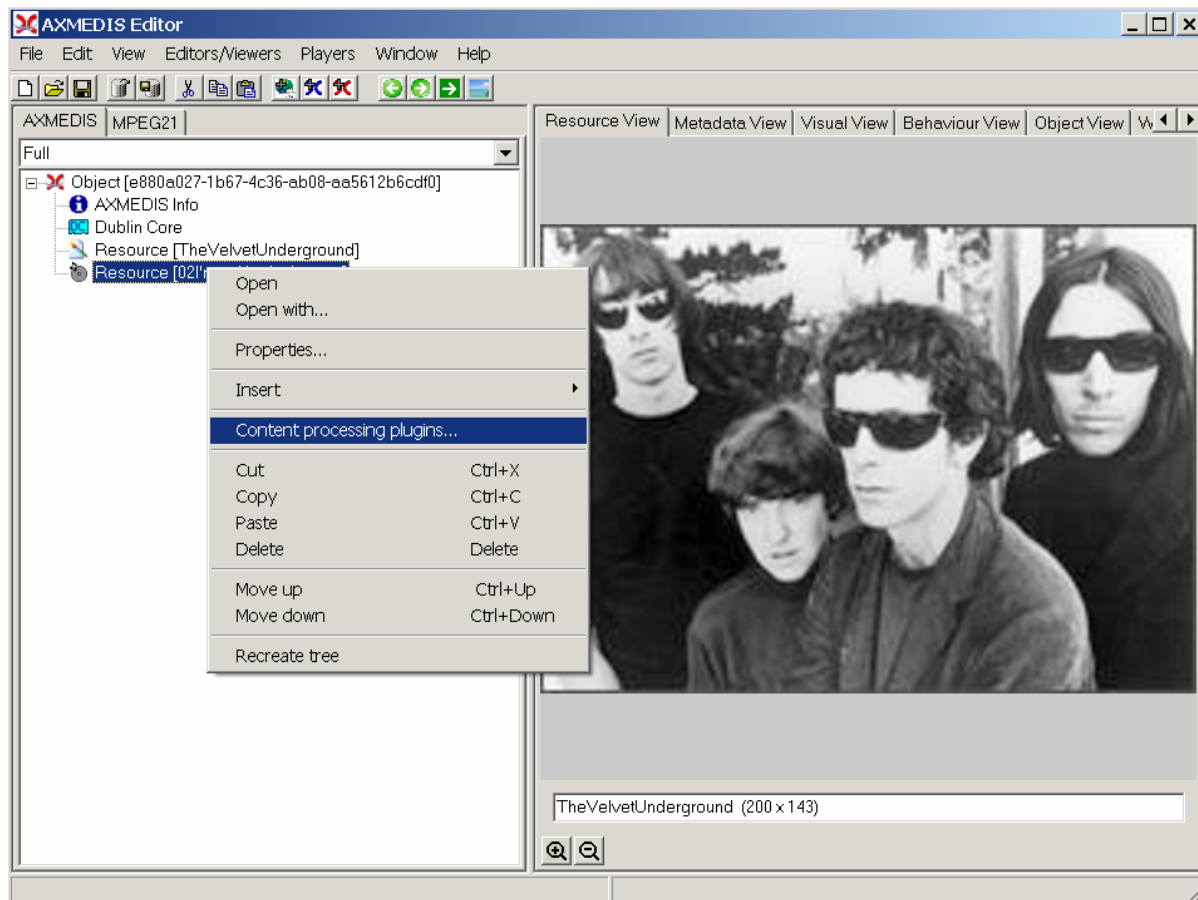


Figure 1 – Calling the content processing plug-ins

A window showing the functionalities available for the kind of resource selected appears (see figure 2).

In the following parts, we discuss independently each available functionality.

16.2.3.1 Low Level Descriptors Extraction

The Low Level Descriptors extraction algorithm allows extracting MPEG-7 compliant descriptors of the audio signal. Such descriptors are said to be “low-level” since they describe the shape and the properties of the audio signal but can not be directly interpreted by humans (as opposed to “high-level” descriptors such as music genre or tempo). The Low-Level Descriptors extraction algorithm is launched by selecting the **AudioDescriptor: LowLevelDescriptors** function:

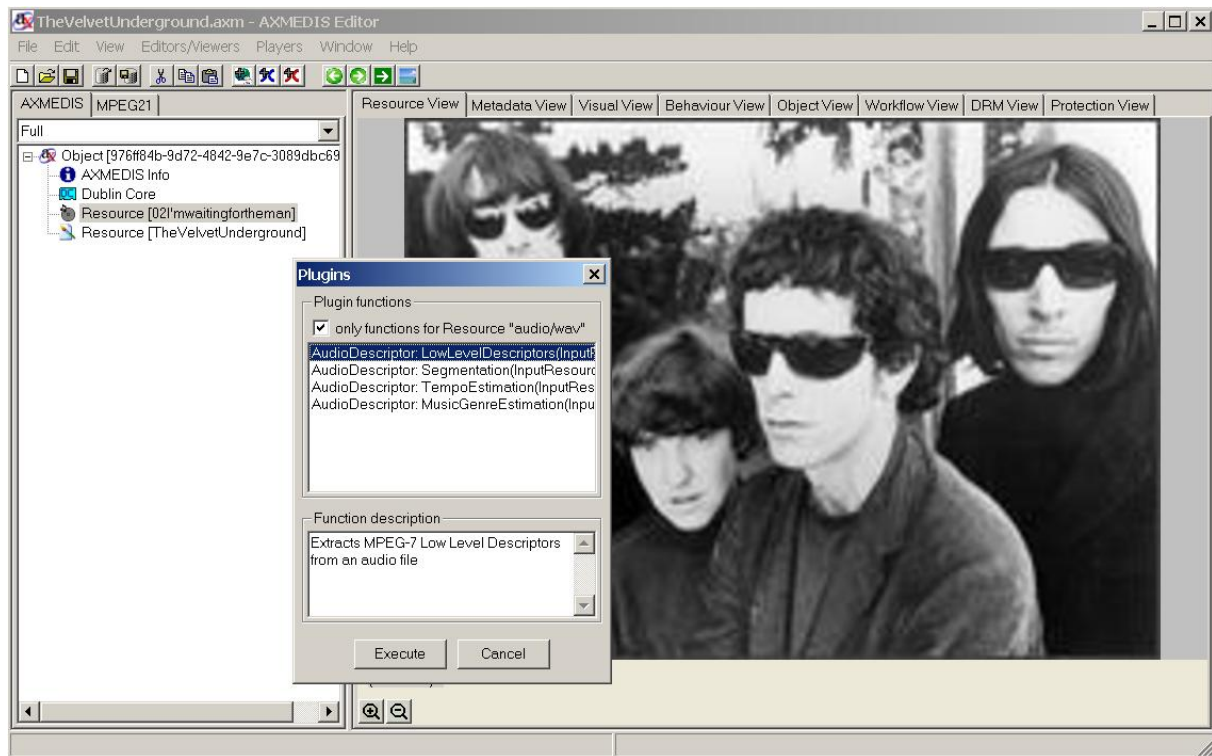


Figure 2 – Selecting the low level descriptors extraction function

A new window appears showing the interface to the low level descriptors extraction algorithm (see figure 3).

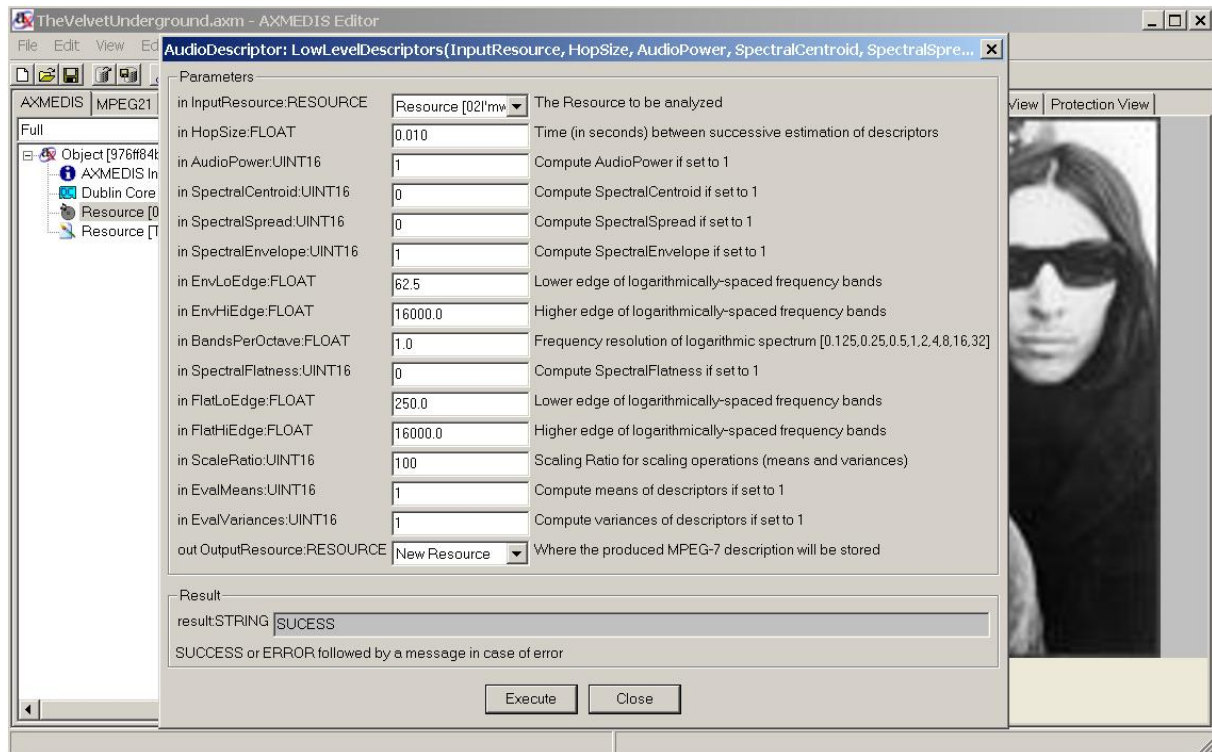


Figure 3 – The low level descriptors extraction function

The input audio file to be analysed is selected with first parameter. The last parameter allows specifying where the resulting MPEG-7 compliant description will be saved. The other parameters allow selecting which low level descriptors should be extracted. The analysis is launched by clicking on the **Execute** button. Once the analysis completed, one can display the resulting MPEG-7 description by double-clicking on the resource in which the description was saved (figure 4).

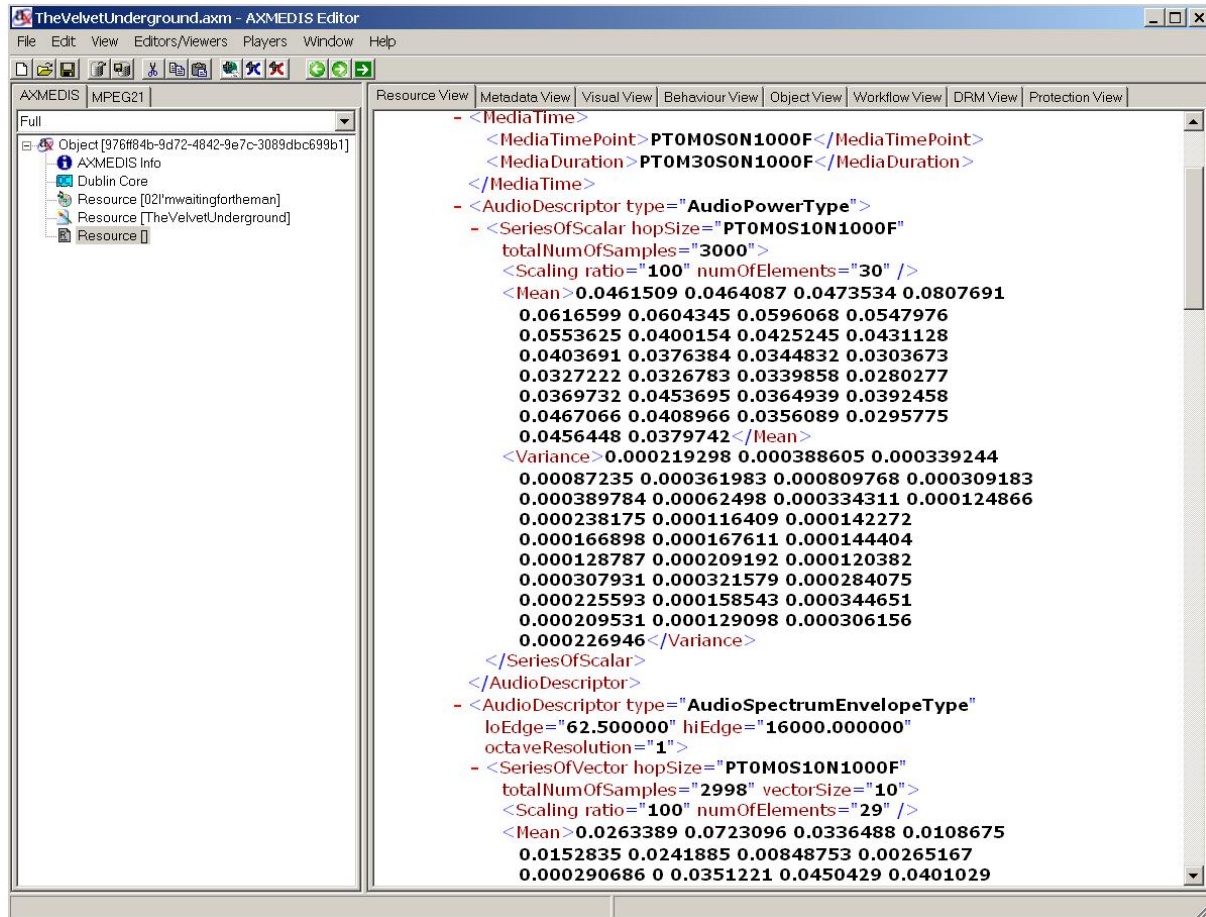


Figure 4 – The resulting MPEG-7 description

Here follows a more complete description of the parameters of the tempo estimation function:

- **InputResource**: the audio resource to be analysed; reading of audio resources is supported for the following mime types (corresponding to uncompressed audio formats):
 1. audio/x-aiff (.aif, .aiff)
 2. audio/x-wav (.wav)
 3. audio/x-basic (.au, .snd)
- **HopSize**: the HopSize defines the temporal distance (in seconds) between two successive analyses (set to 10 ms by default).
- **AudioPower**: computes the AudioPower descriptor if set to 1. The AudioPower descriptor describes the temporally-smoothed instantaneous power of the audio signal. Instantaneous power is a useful measure of the amplitude of a signal as a function of time.
- **SpectralCentroid**: computes the SpectralCentroid descriptor if set to 1. The SpectralCentroid descriptor describes the center of gravity of the log-frequency power spectrum. It is an economical description of the shape of the power spectrum. It indicates whether the power spectrum is dominated by low or high frequencies and, additionally, it is correlated with a major perceptual dimension of timbre, i.e. sharpness.
- **SpectralSpread**: computes the SpectralSpread descriptor if set to 1. The SpectralSpread descriptor describes the second moment of the log-frequency power spectrum. SpectralSpread is an economical descriptor of the shape of the power spectrum that indicates whether it is concentrated in the vicinity

of its centroid, or else spread out over the spectrum. It allows differentiating between tone-like and noise-like sounds.

- **SpectralEnvelope**: computes the SpectralEnvelope descriptor if set to 1. The SpectralEnvelope descriptor describes the spectrum of the audio according to a logarithmic frequency scale. A logarithmic frequency axis is used to conciliate requirements of concision and descriptive power. Peripheral frequency analysis in the ear roughly follows a frequency axis.
- **EnvLoEdge**: set lower edge of logarithmically-spaced frequency bands for the extraction of the SpectralEnvelope descriptor (62.5 Hz by default).
- **EnvHiEdge**: set higher edge of logarithmically-spaced frequency bands for the extraction of the SpectralEnvelope descriptor (16000.0 Hz by default).
- **BandsPerOctave**: frequency resolution of logarithmic spectrum for the extraction of the SpectralEnvelope descriptor (width of each spectrum band between EndLoEdge and EnvHiEdge).
- **SpectralFlatness**: computes the SpectralFlatness descriptor if set to 1. The SpectralFlatness descriptor properties of the spectrum of an audio signal within a given number of frequency bands. This descriptor expresses the deviation of the signal's power spectrum over frequency from a flat shape (corresponding to a noise-like or impulse-like signal). A high deviation from a flat shape may indicate the presence of tonal components.
- **FlatLoEdge**: set lower edge of logarithmically-spaced frequency bands for the extraction of the SpectralFlatness descriptor (250.0 Hz by default).
- **FlatHiEdge**: set higher edge of logarithmically-spaced frequency bands for the extraction of the SpectralFlatness descriptor (16000.0 Hz by default).
- **ScaleRatio**: the ScaleRatio is the number of original samples represented by each scaled sample when using a scaling operation (such as mean or variance of the descriptors)
- **EvalMeans**: computes the mean of the descriptors if set to 1.
- **EvalVariances**: computes the variance of the descriptors if set to 1.
- **OutputResource**: the MPEG-7 description of the audio resource.

16.2.3.2 Segmentation into Silence / Speech / Noise / Music

The Silence / Speech / Noise / Music segmentation algorithm allows segmenting the audio stream into 4 kind of semantically coherent segments:

- **Silence segment**: silence segments are defined as regions of the audio file in which no significant sound is heard.
- **Speech segment**: speech segments are defined as regions of the audio file in which spoken content is dominant.
- **Music segment**: music segments are defined as regions of the audio file in which music content is dominant.
- **Noise segment**: noise segments are defined as regions of the audio file in which noise is dominant; noise is loosely defined as audio content which is not speech, music nor silence.

The segmentation algorithm is called by selecting the **AudioDescriptor: Segmentation** function:

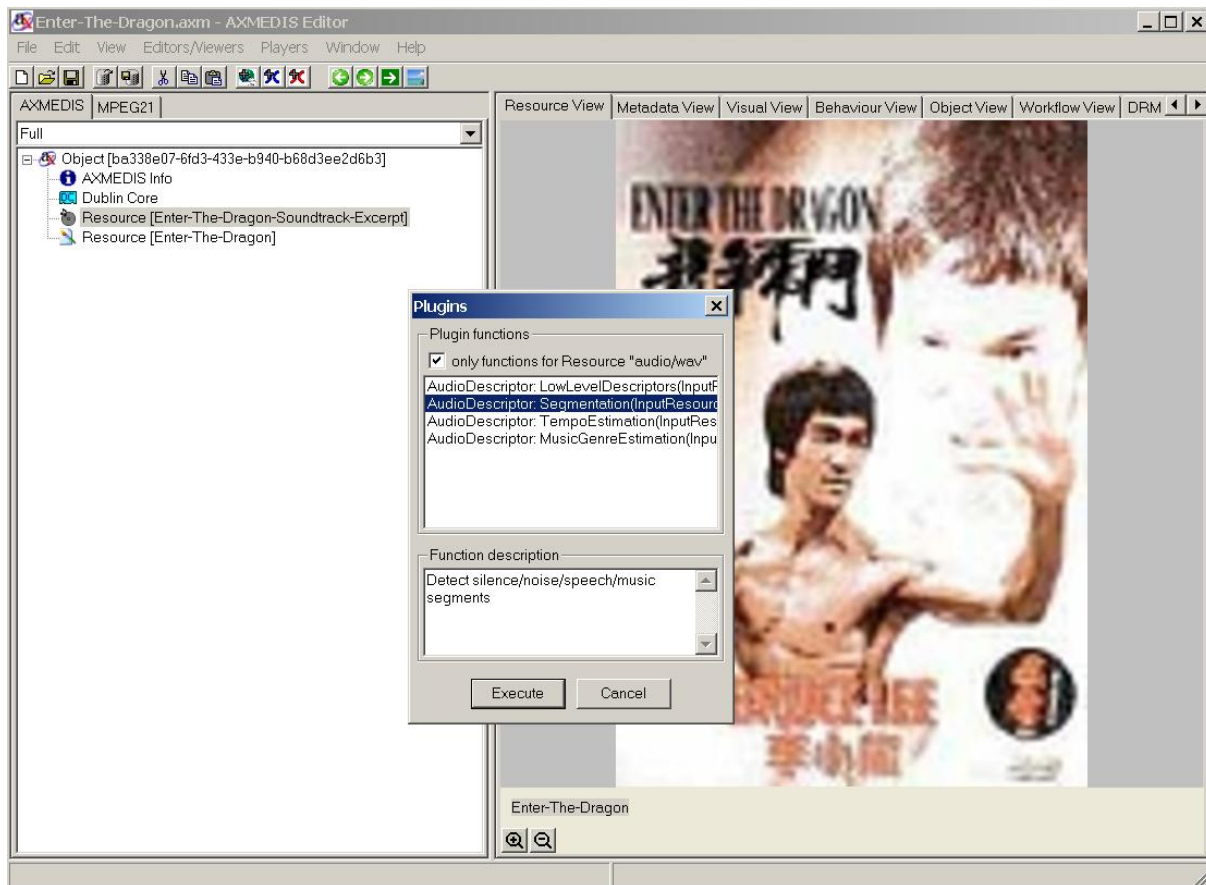


Figure 5 – Selecting the segmentation function

A new window appears showing the interface to the segmentation function (see figure 6).

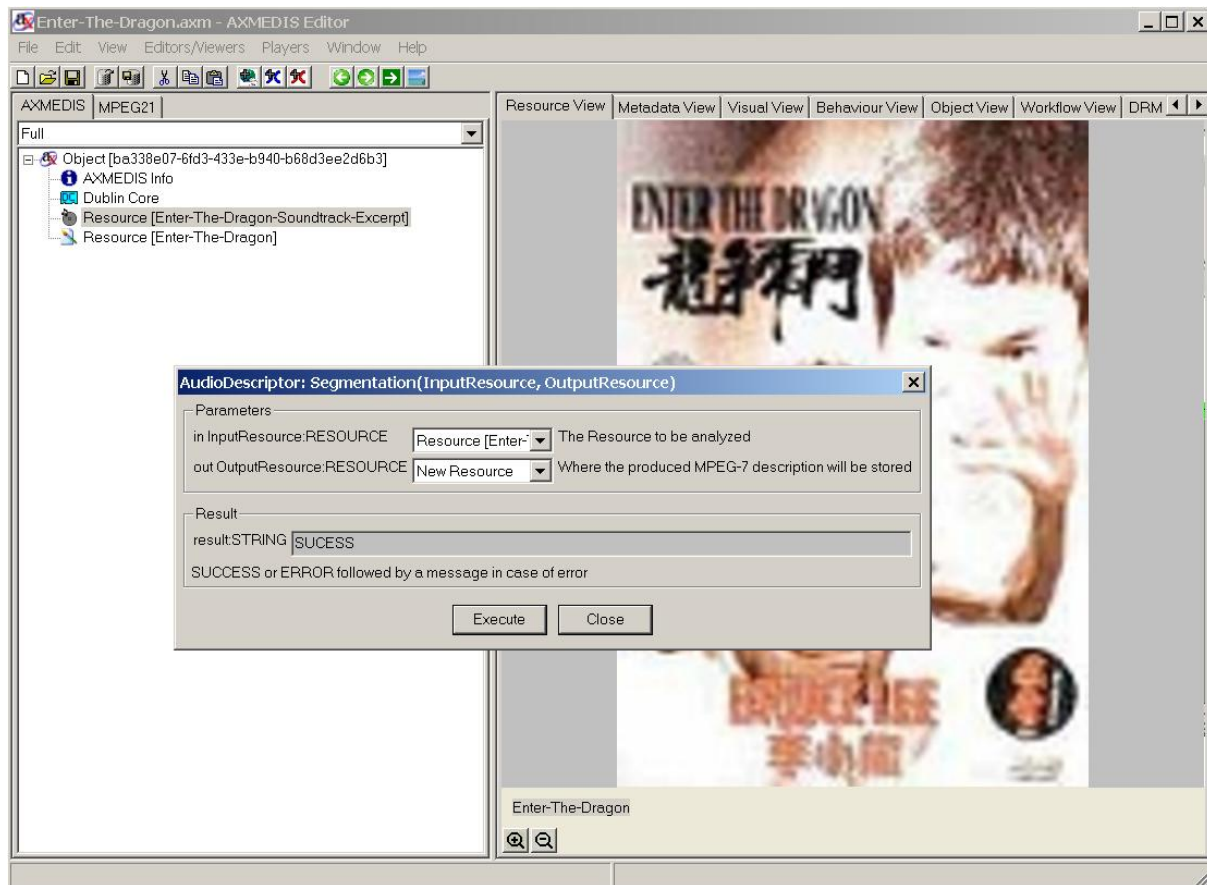


Figure 6 – The speech/noise/music segmentation function

The input audio file to be analysed is selected with first parameter while the second parameter allows specifying where the resulting MPEG-7 compliant description will be saved. The analysis is launched by clicking on the **Execute** button. Once the analysis completed, one can display the resulting MPEG-7 description by double-clicking on the resource in which the description was saved (figure 7).

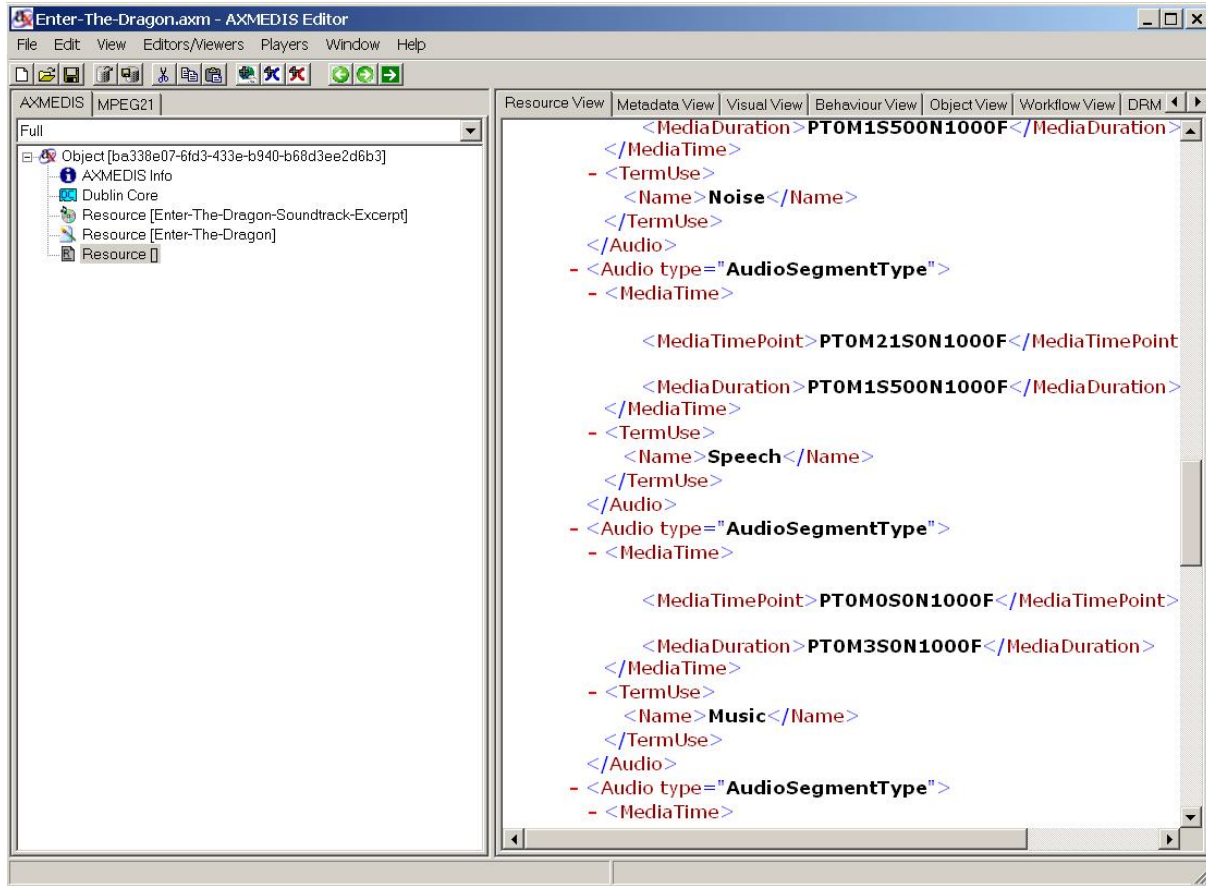


Figure 7 – The resulting MPEG-7 description

Here follows a more complete description of the parameters of the segmentation function:

- **InputResource**: the audio resource to be analysed; reading of audio resources is supported for the following mime types (corresponding to uncompressed audio formats):
 1. audio/x-aiff (.aif, .aiff)
 2. audio/x-wav (.wav)
 3. audio/x-basic (.au, .snd)
- **OutputResource**: the MPEG-7 description of the audio resource.

16.2.3.3 Music Genre recognition

The Music Genre recognizer allows characterizing music segments in terms of music genres. The provided model classifies music into one of the following categories:

- Classical
- Jazz

- Rap
- Rock

The Music Genre recognizer is called by selecting the **AudioDescriptor: MusicGenreEstimation** function:

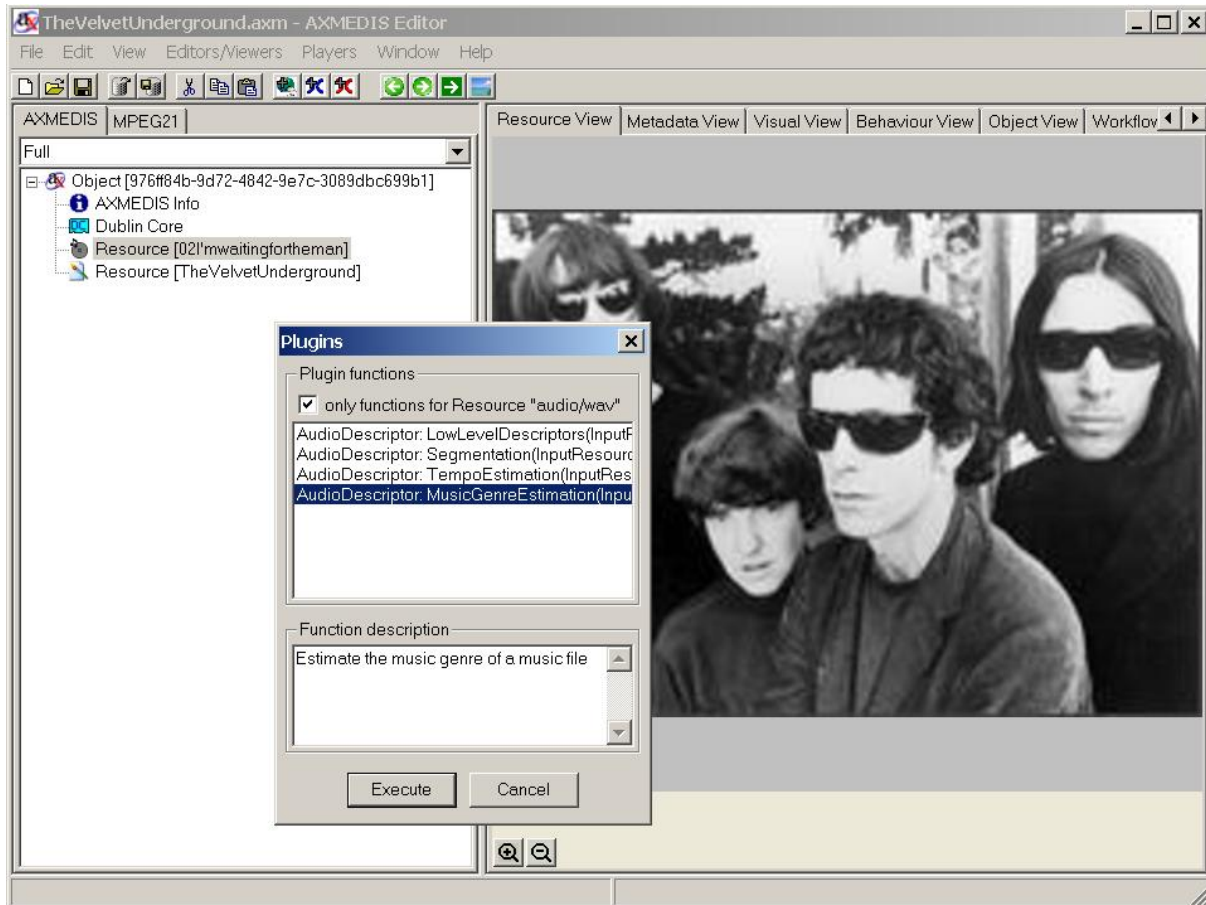


Figure 8 – Selecting the music genre recognition function

A new window appears showing the interface to the music genre recognizer (see figure 9).

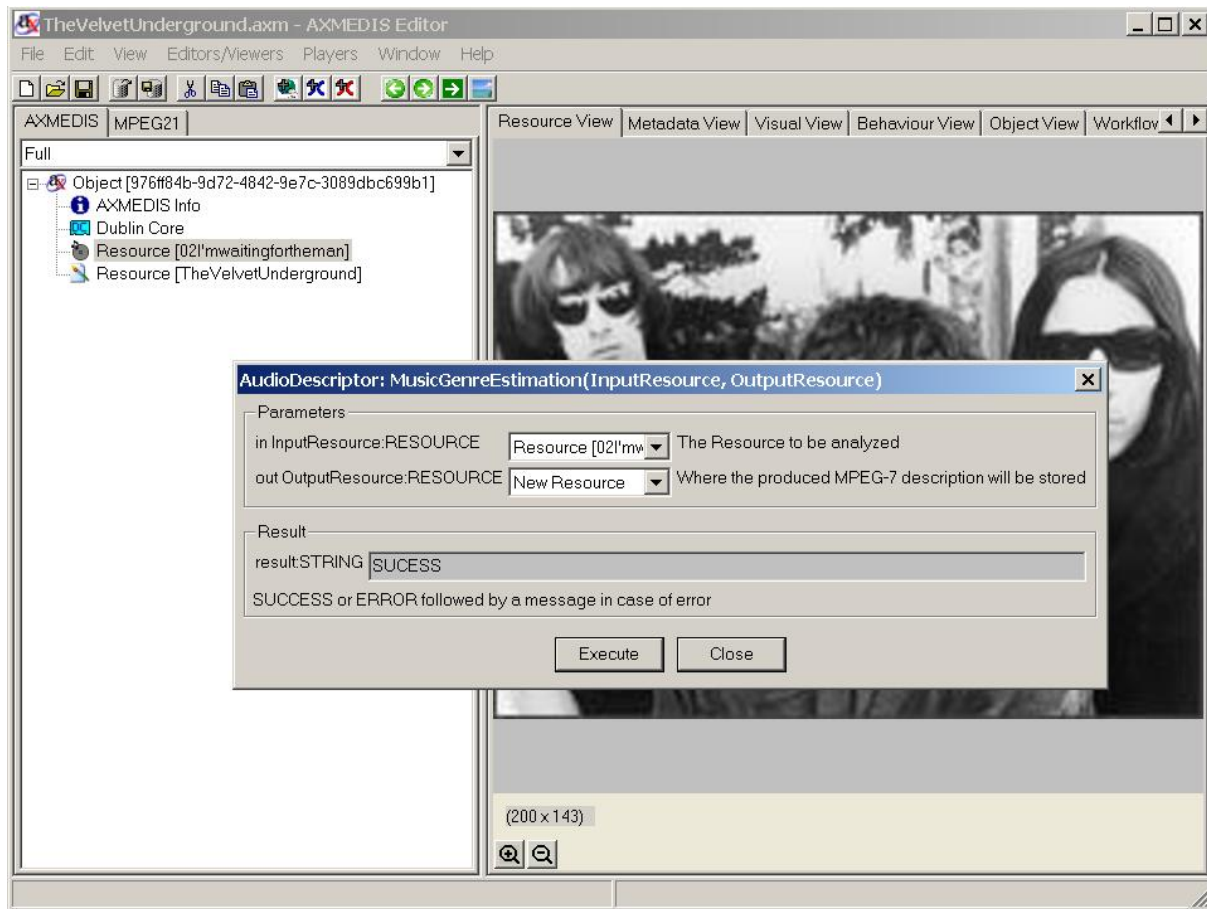


Figure 9 – The music genre recognition function

The input audio file to be analysed is selected with first parameter while the second parameter allows specifying where the resulting MPEG-7 compliant description will be saved. The analysis is launched by clicking on the **Execute** button. Once the analysis completed, one can display the resulting MPEG-7 description by double-clicking on the resource in which the description was saved (figure 10).

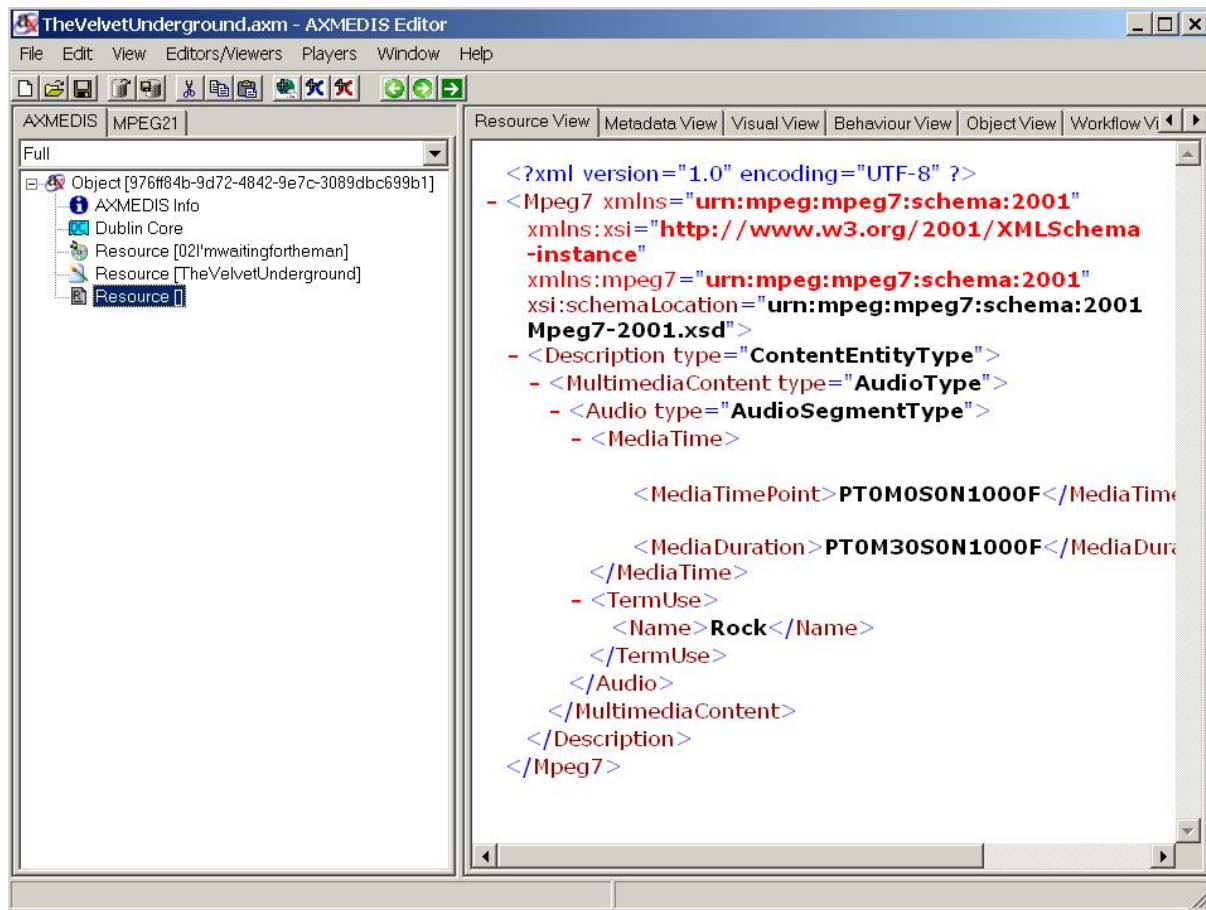


Figure 10 – The resulting MPEG-7 description

Here follows a more complete description of the parameters of the music genre recognition function:

- **InputResource**: the audio resource to be analysed; reading of audio resources is supported for the following mime types (corresponding to uncompressed audio formats):
 4. audio/x-aiff (.aif, .aiff)
 5. audio/x-wav (.wav)
 6. audio/x-basic (.au, .snd)
- **OutputResource**: the MPEG-7 description of the audio resource.

16.2.3.4 Tempo detection

The tempo detection algorithm allows detecting the tempo in beats per minute of a music segment. It is launched by selecting the **AudioDescriptor: TempoEstimation** function:

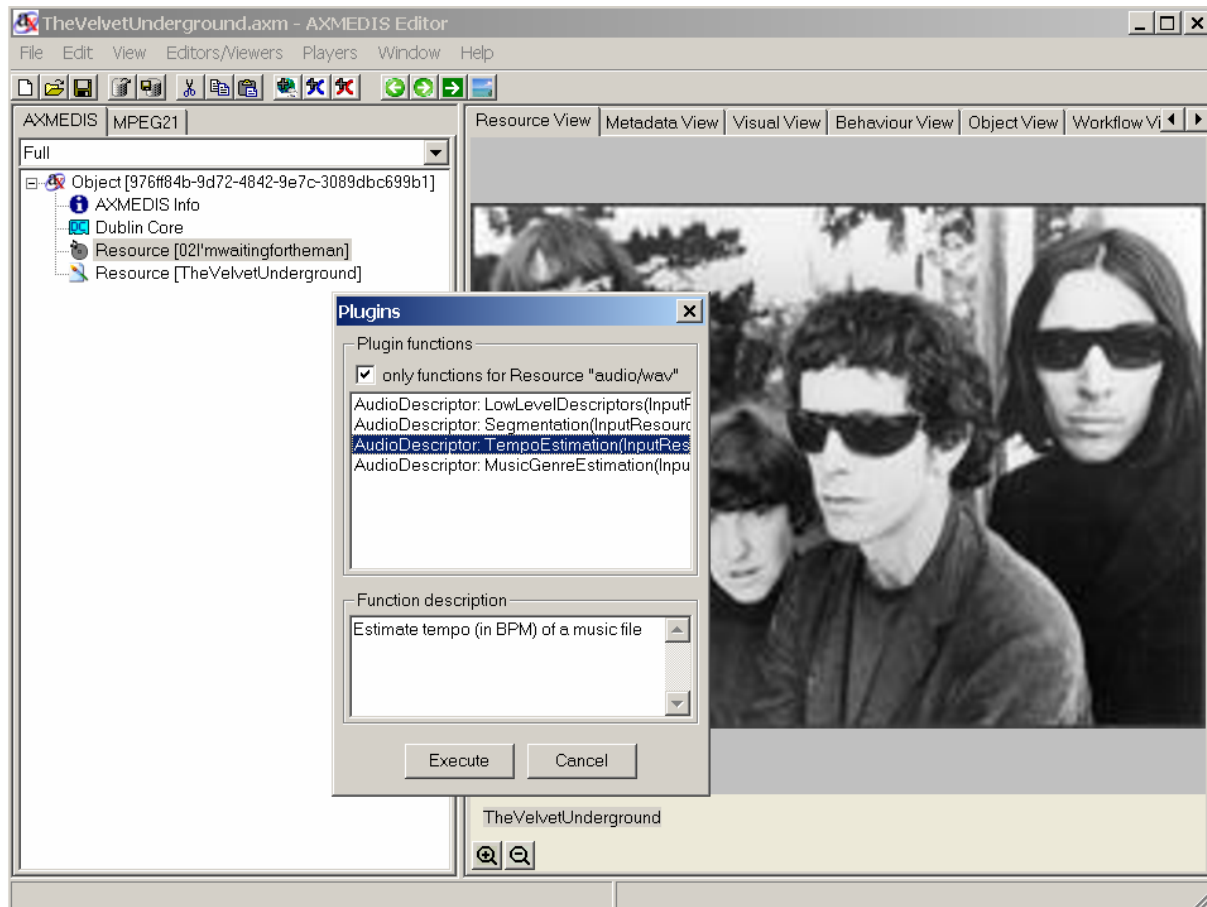


Figure 11 – Selecting the tempo estimation function

A new window appears showing the interface to the music genre recognizer (see figure 12).

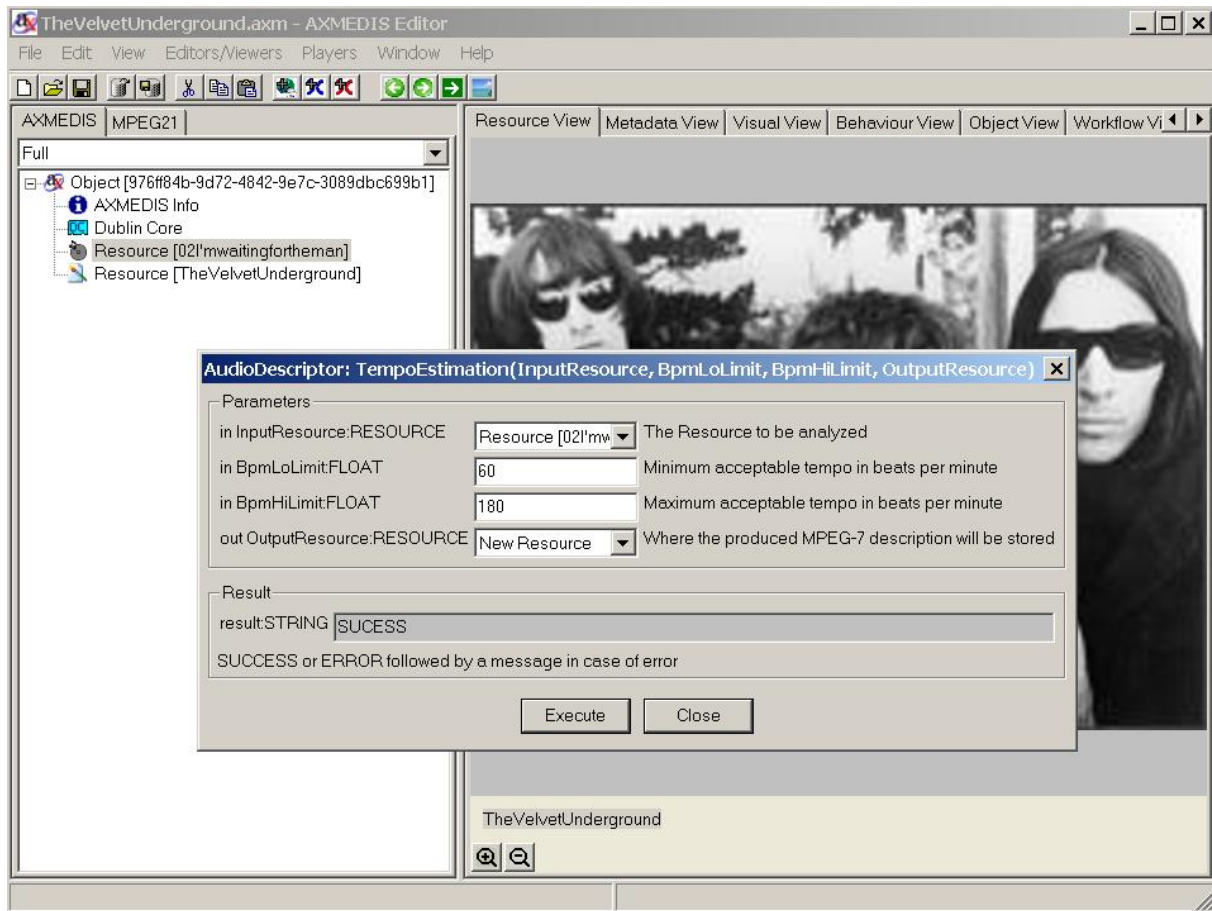


Figure 12 – The tempo estimation function

The input audio file to be analysed is selected with first parameter. The last parameter allows specifying where the resulting MPEG-7 compliant description will be saved. The parameters **BpmLoLimit** and **BpmHiLimit** allow to set boundaries to the estimation of tempo, i.e. the estimated tempo will fit in between these limits. The analysis is launched by clicking on the **Execute** button. Once the analysis completed, one can display the resulting MPEG-7 description by double-clicking on the resource in which the description was saved (figure 13).

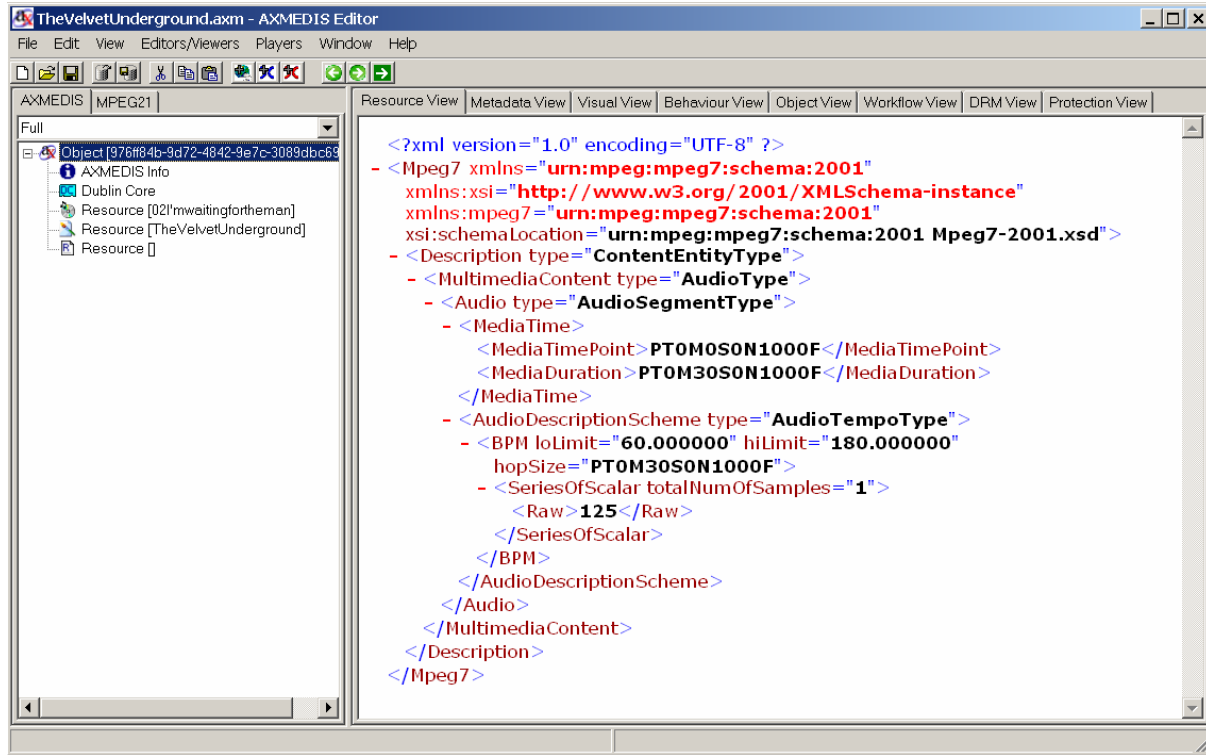


Figure 13 – The resulting MPEG-7 description

Here follows a more complete description of the parameters of the tempo estimation function:

- **InputResource**: the audio resource to be analysed; reading of audio resources is supported for the following mime types (corresponding to uncompressed audio formats):
 1. audio/x-aiff (.aif, .aiff)
 2. audio/x-wav (.wav)
 3. audio/x-basic (.au, .snd)
- **BpmLoLimit**: the minimum acceptable tempo in beats per minute (BPM).
- **BpmHiLimit**: the maximum acceptable tempo in beats per minute (BPM).
- **OutputResource**: the MPEG-7 description of the audio resource.

16.3 Multimedia Adaptation Plugin (EPFL)

16.3.1 Main functionalities

The multimedia_adaptation_plugin allows adapting multimedia content to various use cases. For example, it can be used to transcode an MP4 file into a 3GP file or to extract the media resources embedded into a complex multimedia file. The plug-in is composed by five functions that are: Extract Media Track, Mp4 To 3GP, Mp4 to Isma, Add Media files and Cat Media Files. This document shows the main functionalities provided by a first prototype of the tool forming a minimal user guide.

16.3.2 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.

16.3.3 Detailed description of the functionalities and Screenshots

Here is an example on how to use the plug-in with the AXMEDIS editor.

The plug-in must be applied on an mp4 resource of an AXMEDIS object. The adaptation plug-in is called by right-clicking on the interesting resource and selecting the ‘Content processing plugins...’ command (see figure 1).

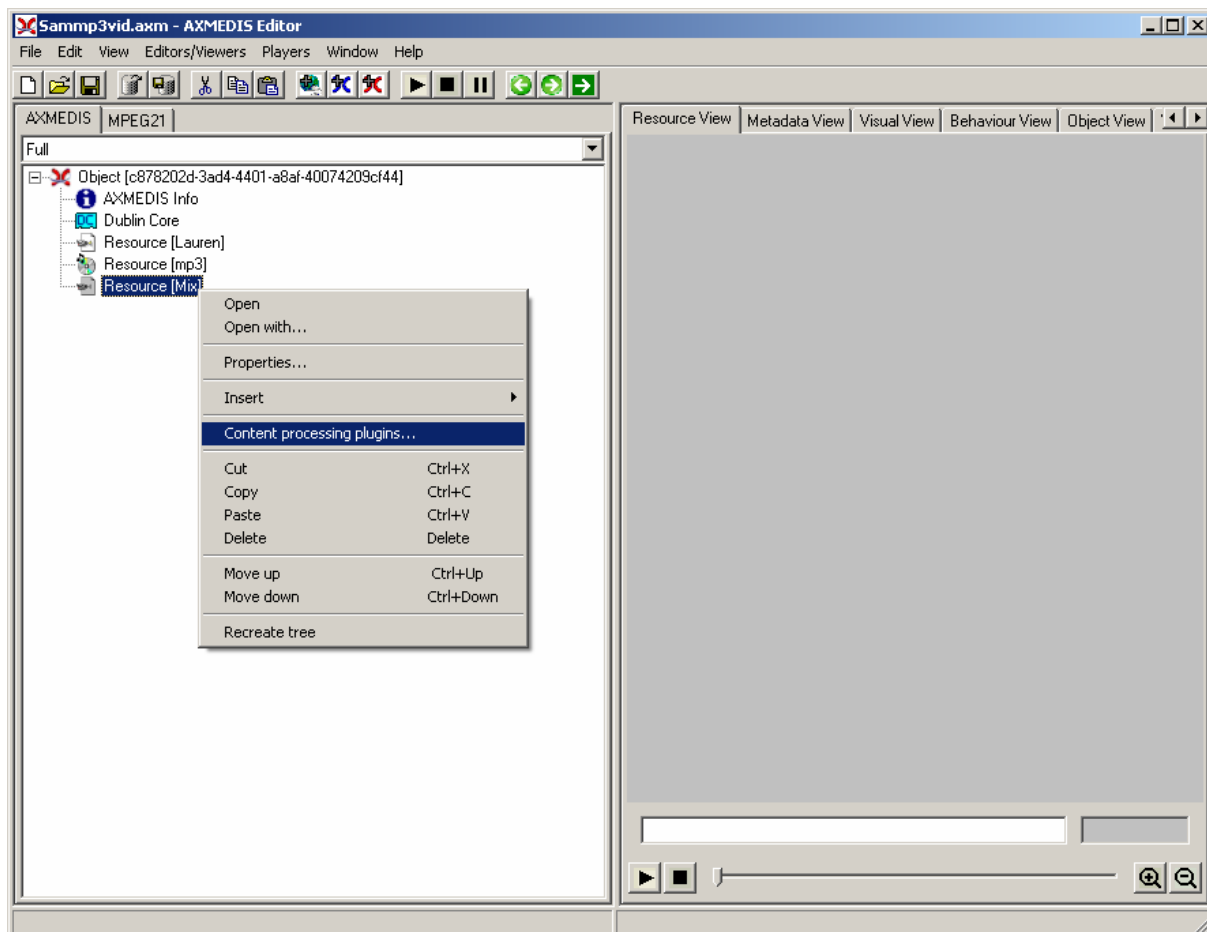


Figure 1 – Calling the multimedia adaptation plug-in

A window showing the functionalities available for the kind of resource selected appears (see figure 2).

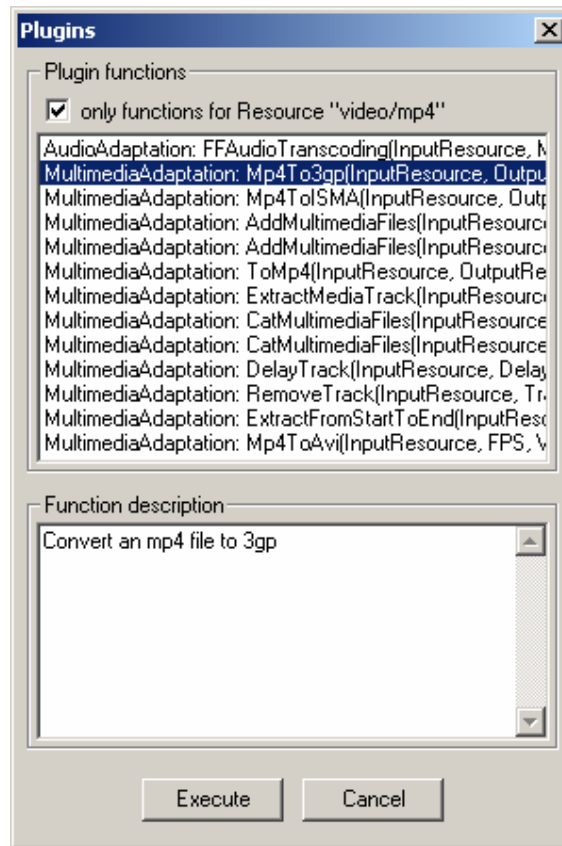


Figure 2 – Selecting one of the multimedia adaptation functions

The following section summarizes the multimedia adaptation functions.

16.3.3.1 EXTRACT MEDIA TRACK

This function extracts one track from the original source (without deleting this track) and leaves it into a separated new file. The supported output mime types are: video/x-cmp, video/x-msvideo, video/mp4 and audio/x-gsm.

This function is selected by clicking on MultimediaAdaptation: ExtractMediaTrack.. After clicking “execute”, a new window appears showing the interface to the extraction function (see figure 3).

Figure 3 – The Extract Media Track function

Here follows a brief description of the parameters of the Extract Media Track function:

- **InputResource**: the multimedia resource where the track is going to be extracted (not deleted). Its use is only allowed for the video/mp4 (.mp4) files.
- **OutputResource**: the multimedia resource after the extraction, it is to say, the track already extracted into a new file. It can be part of all the allowed mpeg-4 compliant formats.
- **TrackID**: track to be extracted from the InputResource. If the track does not exist, the result will show “ERROR: Bad parameter”.
- **Mimetype**: mime type of the OutputResource. Supports video/x-cmp, video/x-msvideo, video/mp4 and audio/x-gsm.

16.3.3.2 MP4 TO 3GP

This function will translate the input resource that is supposed to be .mp4 into a new file with the 3gp format. This function is selected by clicking on MultimediaAdaptation: Mp4To3GP. After clicking “execute”, a new window appears showing the interface to the Mp4 to 3GP function (see figure 4).

MultimediaAdaptation: Mp4To3gp(InputResource, OutputResource, KeepSys)

Parameters

in InputResource:RESOURCE	Resource [Mix]	Input File to be converted
out OutputResource:RESOURCE	New Resource	Where the produced resource will be stored
in KeepSys:BOOLEAN	true	Keep systems tracks

Result

result:STRING

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

Execute Close

Figure 4 – The Mp4 to 3GP function

Here follows a brief description of the parameters of the Mp4 to 3GP function:

- **InputResource**: The multimedia resource to be translated into 3gp. At first, it is only allowed to use an mp4 resource.
- **OutputResource**: The multimedia output that is obtained after the transformation performed by the function. The obtained file is an .3gp file.
- **KeepSys**: If it should keep system tracks within the translation.

16.3.3.3 CAT MULTIMEDIA FILES

This function concatenates two whole multimedia resources and gives a new file containing the result of the concatenation.

This function is selected by clicking on MultimediaAdaptation: CatMultimediaFiles. After clicking “execute”, a new window appears showing the interface to the concatenation function (see figure 5).

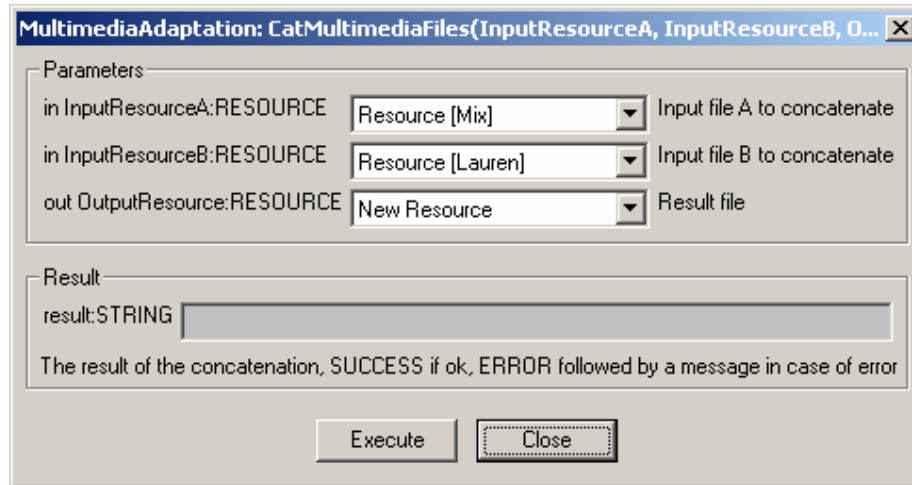


Figure 5 – The Cat Media Files function

Here follows a brief description of the parameters of the Cat Media Files function:

- **InputResourceA**: It is one of the multimedia sources to concatenate. It will be the first in the timeline of the output file. By the moment, it is only allowed to introduce .mp4 files.
- **InputResourceB**: It is one of the multimedia sources to concatenate. It will be included after the InputResourceA into the new output resource. By the moment, it is only allowed to introduce .mp4 files.
- **OutputResource**: Is the result of the concatenation of InputResourceA and InputResourceB. The format of this file is .mp4.

16.3.3.4 MP4 TO ISMA

This function converts the input resource to the ISMA specification.

The function is selected by clicking on MultimediaAdaptation: Mp4ToISMA. After clicking “execute”, a new window appears showing the interface to the conversion function (see figure 6).

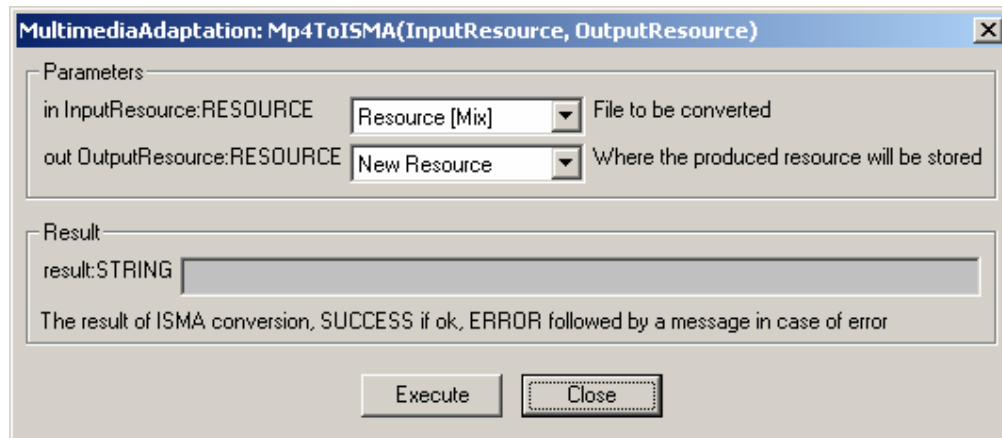


Figure 6 – The Mp4 to ISMA function

Here follows a brief description of the parameters of the Mp4 to ISMA function:

- **InputResource**: Mp4 file to be converted into the ISMA specification.
- **OutputResource**: Output file of the conversion to ISMA of the InputResource.

16.3.3.5 ADD MULTIMEDIA FILES

This function takes multimedia resources and adds them as new tracks into new or already existing mp4 file. It must specify the size (amount of seconds) of the multimedia resource that is imported and when should it begin inside the destination file, it is to say, the delay of the new track.

The function is selected by clicking on MultimediaAdaptation: AddMultimediaFiles. After clicking “execute”, a new window appears showing the interface to the Add Multimedia Files function (see figure 7).

Figure 7 – The Add Media Files function

Here follows a brief description of the parameters of the Add Media Files function:

- **InputResource**: File to be included into a new MP4 file.
- **BaseResource**: Base MP4 file where to add the InputResource
- **Delay**: Delay in milliseconds to be applied at the track to be included into the MP4 output file.
- **ImportLength**: Number of seconds to import from the input file starting from the beginning.
- **TrackID**: Track to extract in the file. If empty take the whole file.
- **FPS**: Frames per sample of the new track. 0 means source file FPS.
- **Lang**: Optional: Language code of the new track
- **OutputResource**: Output file where the track is included

16.3.3.6 TO MP4

This function converts the input resource to Mp4.

The function is selected by clicking on MultimediaAdaptation: ToMp4. After clicking “execute”, a new window appears showing the interface to the conversion function (see figure 8).

Figure 8 – The To Mp4 function

Here follows a brief description of the parameters of the To Mp4 function:

- **InputResource**: File to be converted to Mp4.
- **OutputResource**: Output file of the conversion to Mp4 of the InputResource.

16.3.3.7 DELAY TRACK

This function set the delay to a track from a mp4 file.

The function is selected by clicking on MultimediaAdaptation: DelayTrack. After clicking “execute”, a new window appears showing the interface to the Delay Track function (see figure 9).

Figure 9 – The Delay Track function

Here follows a brief description of the parameters of the Delay Track function:

- **InputResource**: Mp4 file where to delay a track.
- **Delay**: New delay in milliseconds applied at the track of the MP4 output file.
- **TrackID**: Track to be delayed in the file.
- **OutputResource**: Output Mp4 file where the track is included delayed.

16.3.3.8 REMOVE TRACK

This function removes a track from a mp4 file.

The function is selected by clicking on MultimediaAdaptation: RemoveTrack. After clicking “execute”, a new window appears showing the interface to the Remove Track function (see figure 10).

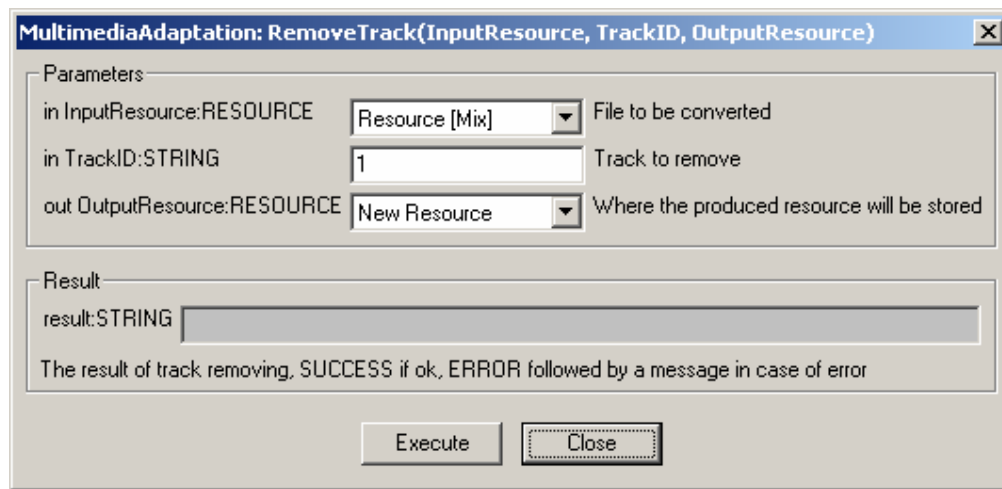


Figure 10 – The Remove Track function

Here follows a brief description of the parameters of the Remove Track function:

- **InputResource**: Mp4 file where to remove a track.
- **TrackID**: Track to be removed from the file.
- **OutputResource**: Output mp4 file where the track is removed.

16.3.3.9 EXTRACT FROM START TO END

This function extracts a new mp4 file from a mp4 file by time limitation.

The function is selected by clicking on MultimediaAdaptation: ExtractFromStartToEnd. After clicking “execute”, a new window appears showing the interface to the Extract from Start to End function (see figure 11).

Figure 11 – The Extract from Start to End function

Here follows a brief description of the parameters of the Extract from Start to End function:

- **InputResource**: File where to extract the new mp4 file.
- **Start**: Start of extraction in seconds
- **End**: End of extraction in seconds
- **OutputResource**: Output mp4 file limited by time

16.3.3.10 MP4 TO AVI

This function will translate the input resource that is supposed to be .mp4 pure BIFS file into a new file with the avi format.

This function is selected by clicking on MultimediaAdaptation: Mp4ToAvi. After clicking “execute”, a new window appears showing the interface to the Mp4 to Avi function (see figure 12).

Figure 12 – The Mp4 to Avi function

Here follows a brief description of the parameters of the Mp4 to Avi function:

- **InputResource**: The Mp4 pure BIFS file to be translated to Avi format.
- **FPS**: Extraction frame rate (0 compute from the BIFS track duration)
- **Width**: Width of the bifs scene (0 takes original size)
- **Height**: Height of the bifs scene (0 takes original size)
- **OutputResource**: The multimedia output that is obtained after the transformation performed by the function. The obtained file is an .avi file.

16.4 Document CryptLib Plugin (EPFL)

16.4.1 Main functionalities

Cryptlibplugin is a tool that can encrypt/decrypt AXMEDIS object. This document shows the main functionalities provided by a first prototype of the tool forming a minimal user guide.

16.4.2 Relationship with other tools

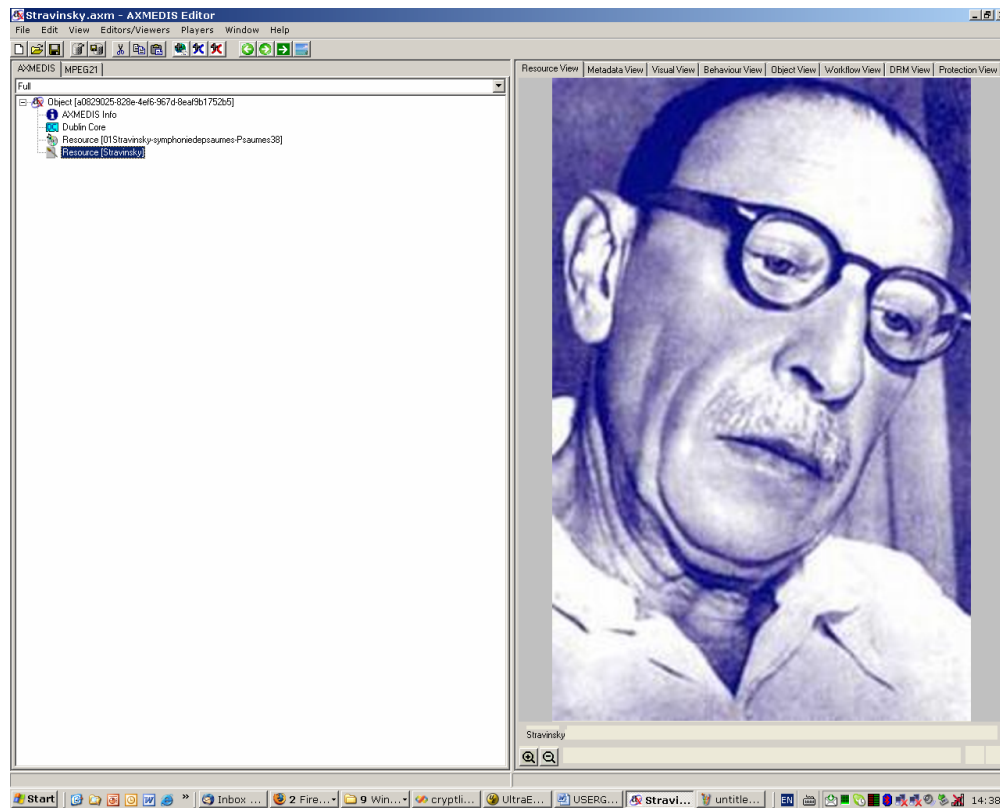
This tool is implemented as a plug-in. Like other plug-ins, its functionality is available via the AXMEDIS Editor and the AXMEDIS Processing Engine.

16.4.3 Detailed description of the functionalities and Screenshots

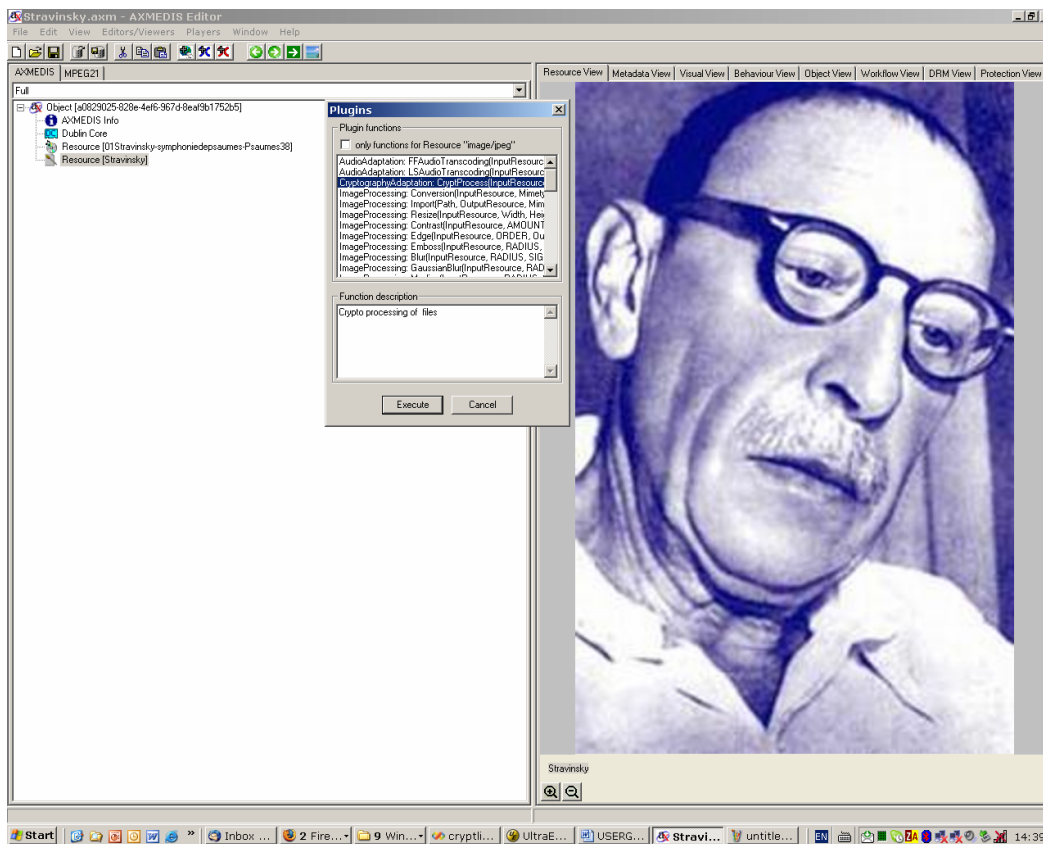
Here's an example on how to use the plug-in with Axeditor.

The plug-in can be applied to all AXMEDIS objects. In the resourcePath directory, there is a sample AXMEDIS file.

Select one of the AXMEDIS object.

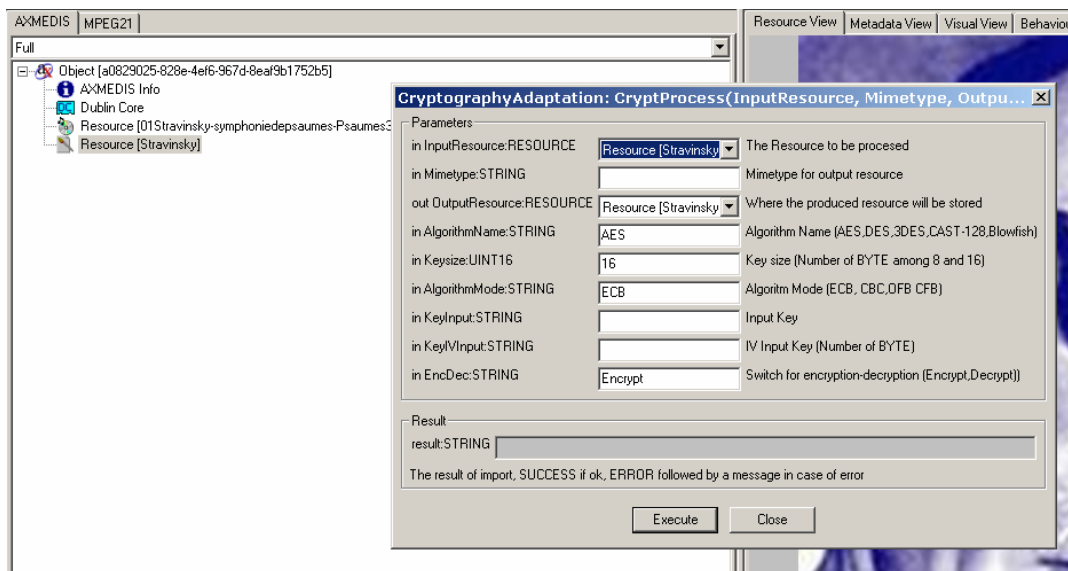


Then select ‘Content Processing Plug-in...’ command; the following window should appear:



There is one function available: Select Cryptography Adaptation.

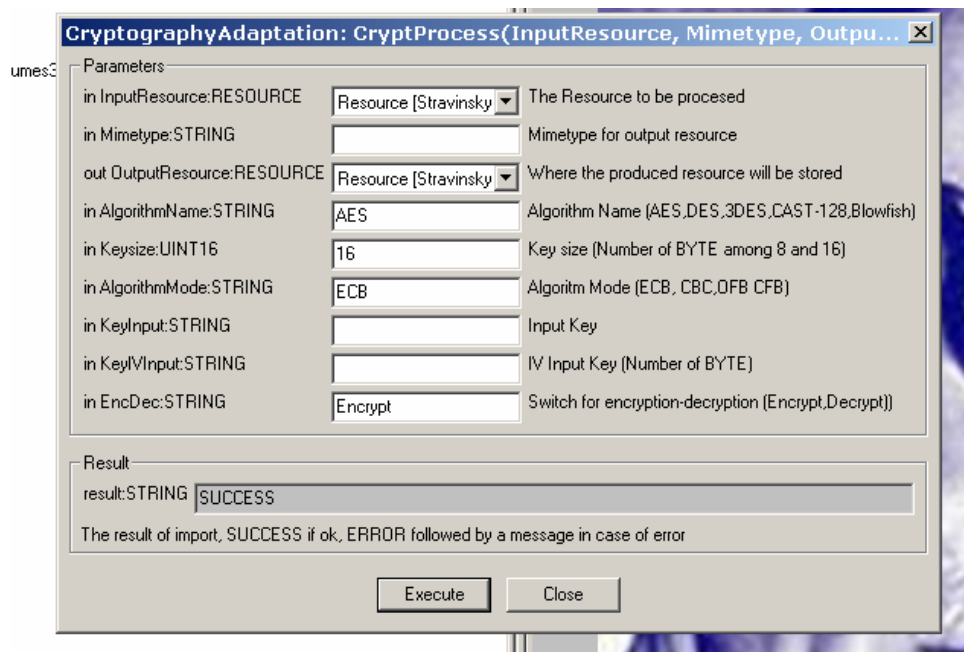
The function accepts the parameters showed in the next figure



In detail:

- **InputResource:** it describes the resource input
- **Mimetype:** it describes the object type
- **OutputResource:** it is the destination object
- **AlgorithmName:** it is the encryption/decryption algorithm that the user wants to use. The algorithms supported are AES, DES, 3DES, CAST-128 and Blowfish. The key size is related to the algorithm selected.
- **Keysize:** it is the number of byte of the input key. The DES algorithm wants, for instance, 8 bytes (64 bit key). The AES works with 128 bits.
- **KeyInput:** it is the input key
- **AlgorithmMode:** it is the encryption/decryption mode. If the user wants to use a mode different from ECB, he has to provide the KeyIVInput parameter.
- **KeyIVInput:** it is the key needed for the mode different from ECB
- **EncDec:** it is the flag to select the encryption or decryption operation-

Clicking on execute makes the plug-in run. Output is given in the 'out' field:

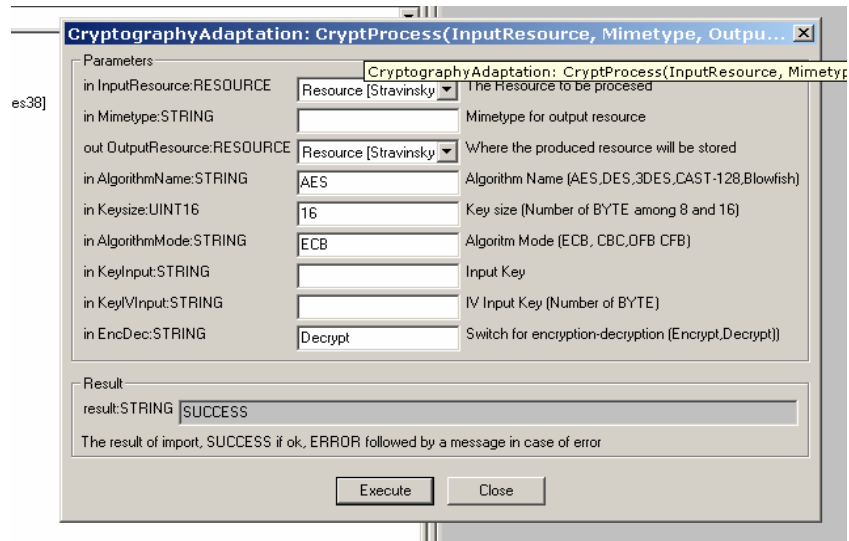


If you try to open the "new" AXMEDIS resource an error appears.

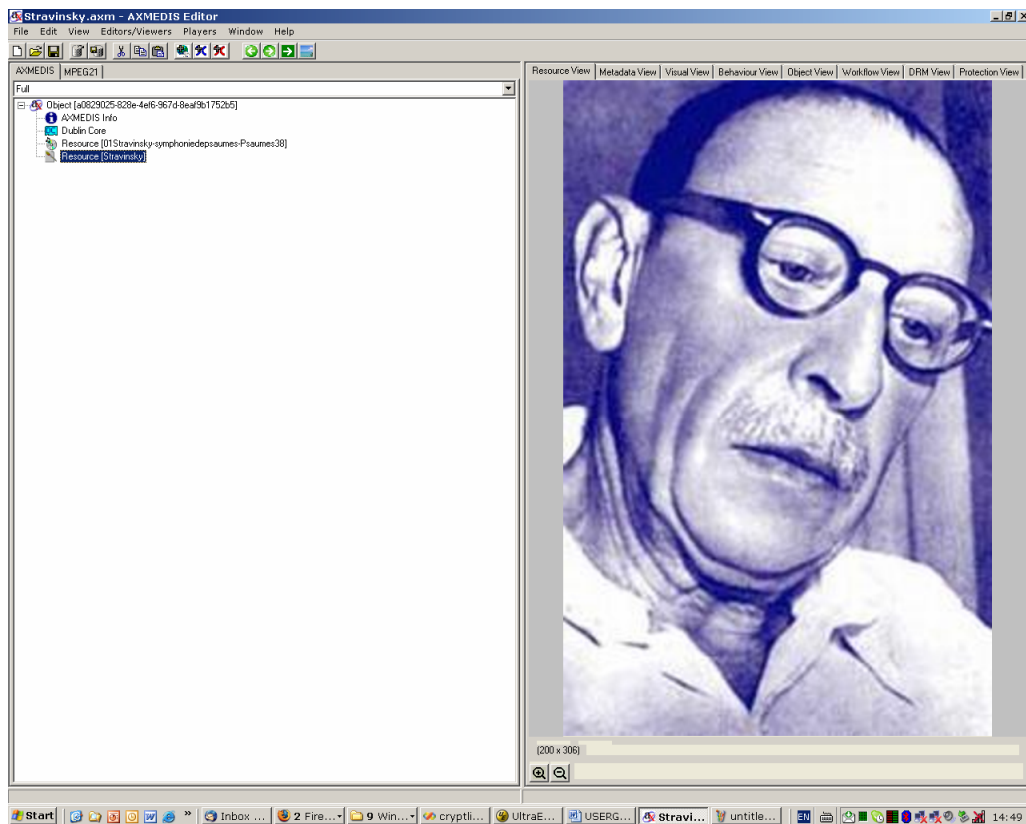


Due to the encryption of the object the image cannot be read.

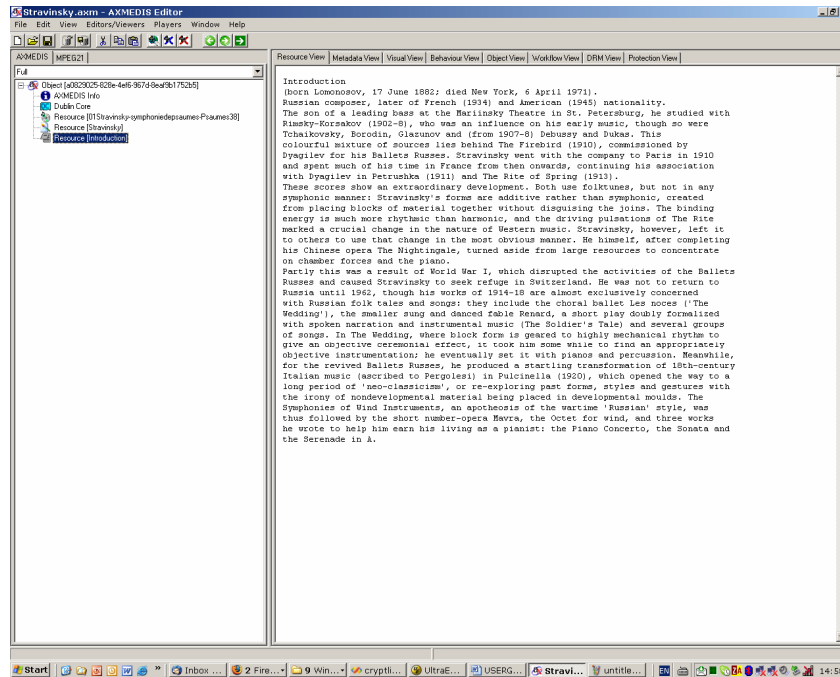
Doing the "Decrypt" operation the resource can be open again.



The result:

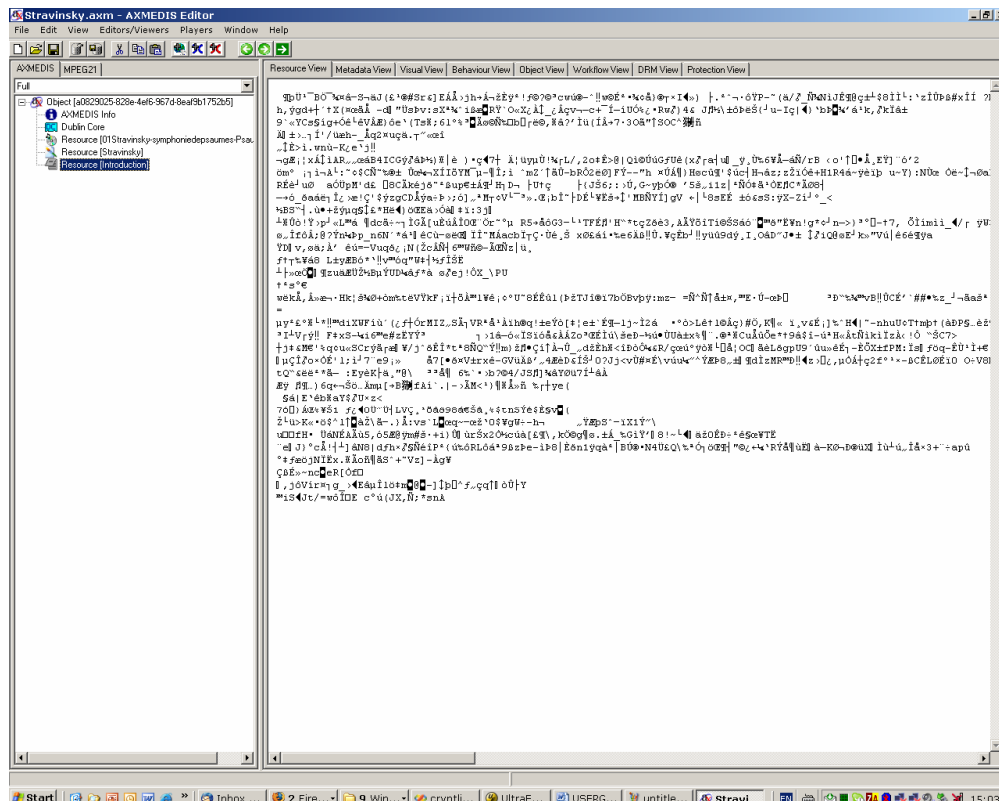


The encryption can be seen better using a text resource. Add the Introduction.txt file that you can find in the resourcePath folder.



Follow the steps previously described for the plug in.

As the user can see the text file now is a "random" sequence of chars.



16.5 Plagiarism Detection Plugin (DIPITA)

16.5.1 Main functionalities

The plagiarism plugin is meant to detect possible plagiarism of textual documents.

It is based on an algorithm that takes into account the plagiarist behaviour.

This behaviour is modeled as a set of actions like insertion, deletion, substitution or transposition and gives as a result a similarity value which is normalized between 0 and 1.

Next we will show the main functionalities provided by a first prototype of the tool.

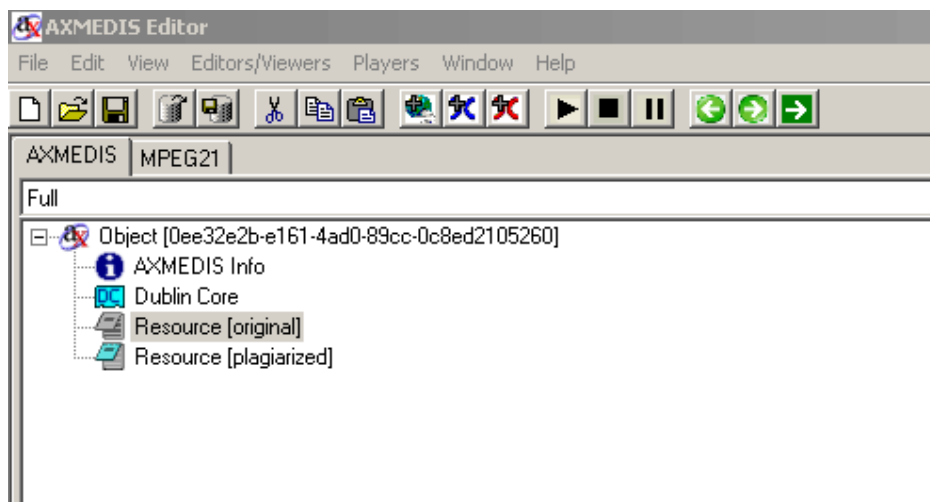
16.5.2 Relationship with other tools

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.

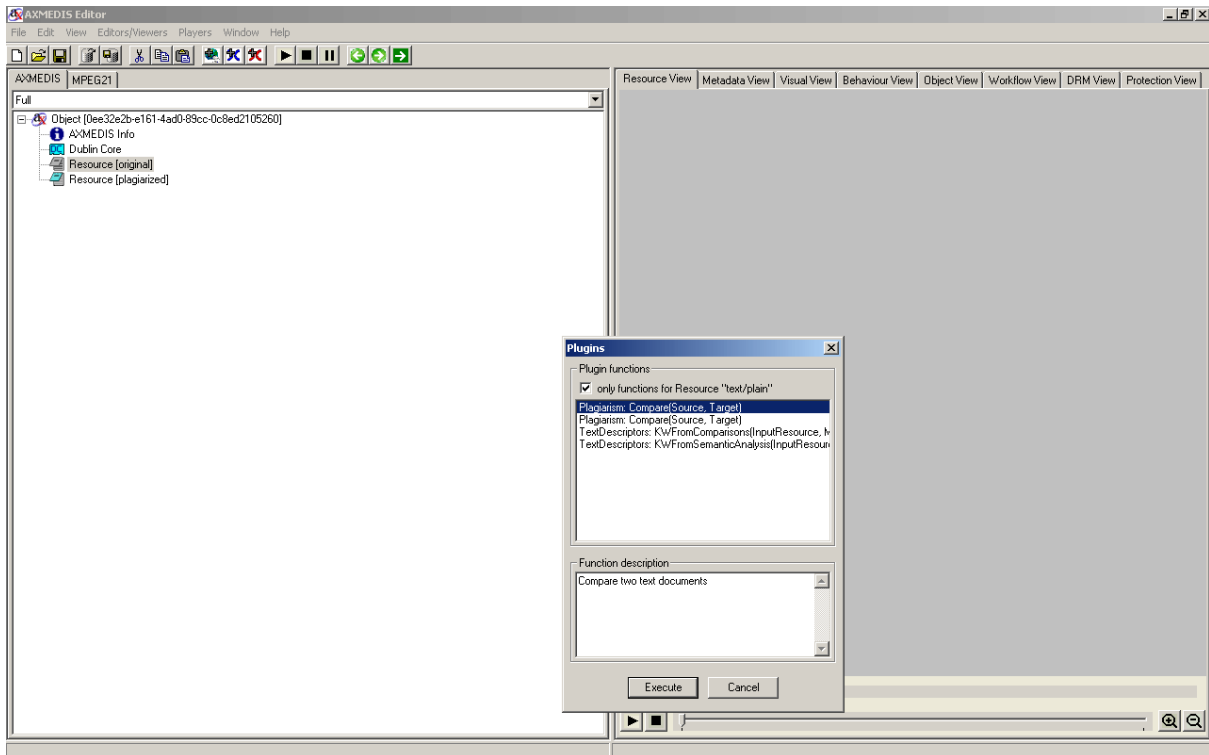
16.5.3 Detailed description of the functionalities and Screenshots

Here's an example on how to use the plug-in within the Axeditor. The plug-in can be applied only to plain text resource.

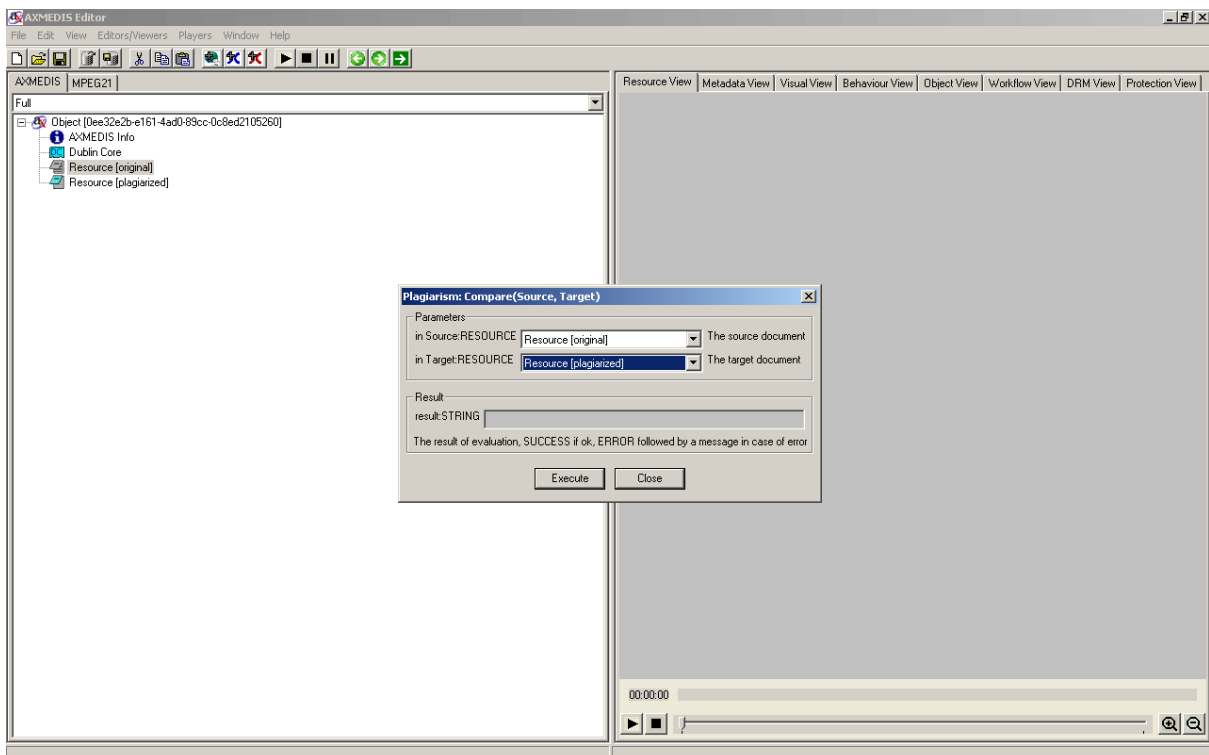
Create a new AXMEDIS object and add the text files to be checked (original and suspect plagiarized) as embedded resources.



Then selecting the ‘Content Processing Plug-in...’ command; the following window should appear:

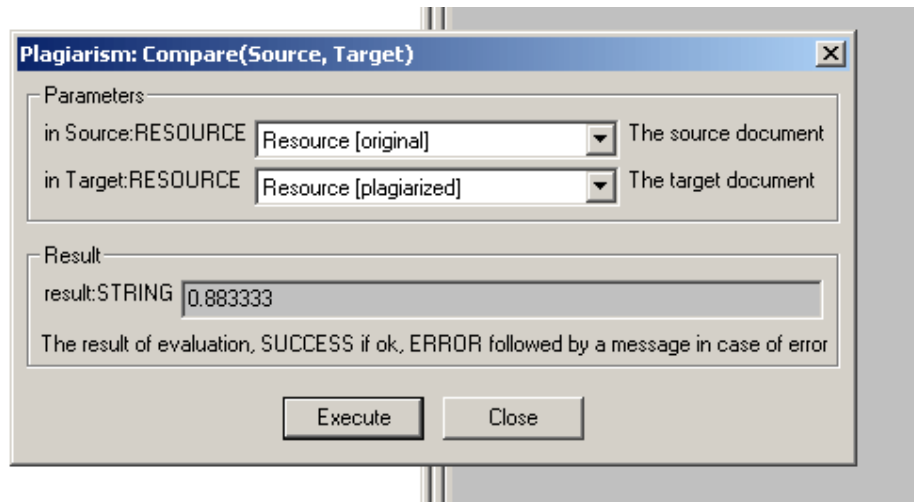


Select the plagiarism plugin the following window should appear:



Select the the source and target file to be compared and then click on execute.

Clicking on execute makes the plug-in run. The algorithm compares two plain text documents and gives as a result a similarity value which is normalized between 0 and 1. Output is given in the 'out' field.



16.6 Document Descriptor Extractor Plugin (DIPITA)

16.6.1 Main functionalities

Descriptor_extractor_plug-in is a tool that can extract high-level metadata from text documents. So far, metadata include single and multi-word keywords. This document shows the main functionalities provided by a first prototype of the tool forming a minimal user guide.

16.6.2 Relationship with other tools

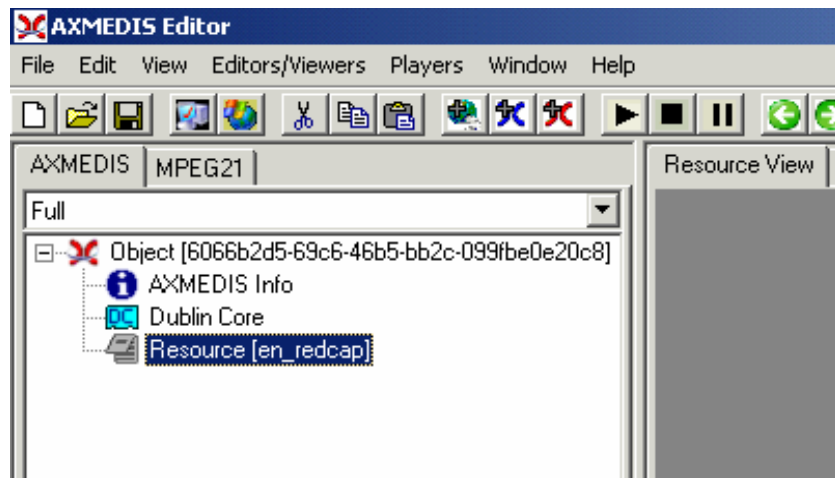
This tool is implemented as a plug-in. Like other plug-ins, its functionality is available via the AXMEDIS Editor and the AXMEDIS Processing Engine.

16.6.3 Detailed description of the functionalities and Screenshots

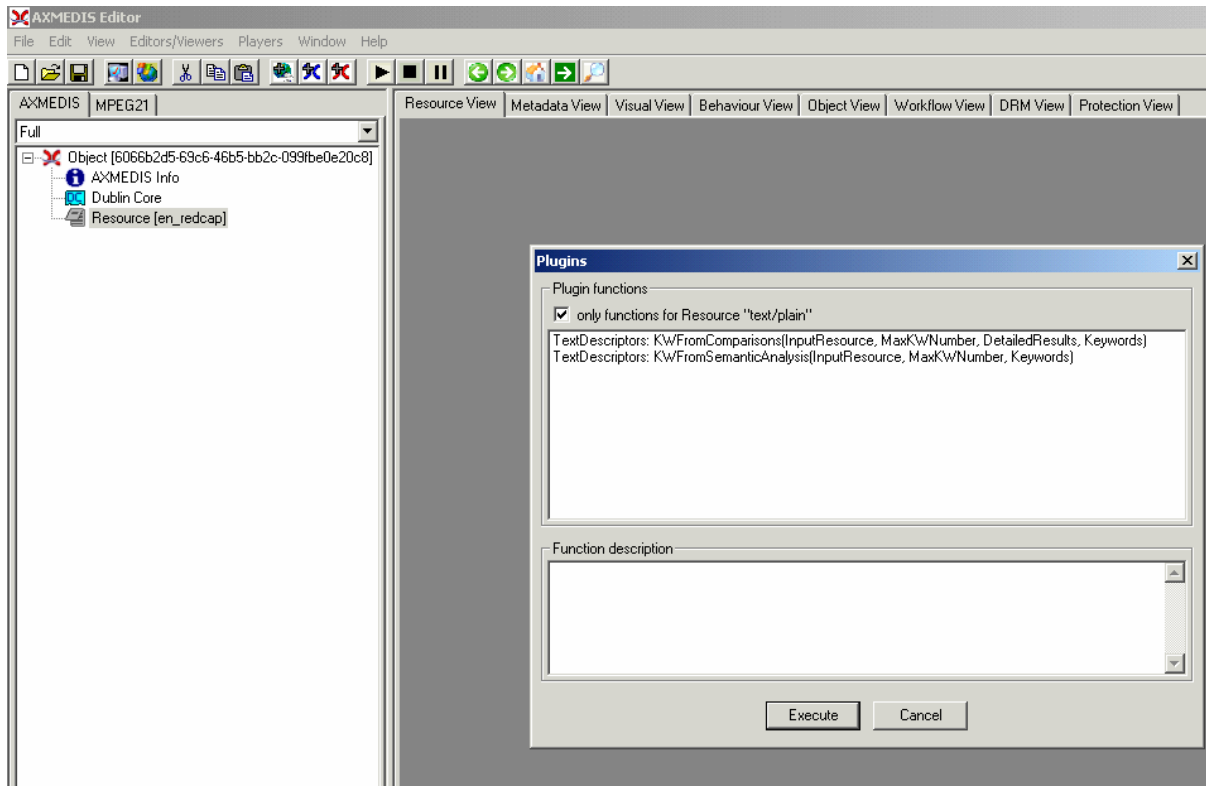
Here's an example on how to use the plug-in with Axeditor.

The plug-in can be applied only to plain text resources and will give meaningful results only to English texts. In the resourcePath directory, there is a sample file to test: en_redcap.txt. It's the well known Red Cap tale.

Create a new AXMEDIS object and add the txt file as an embedded resource.



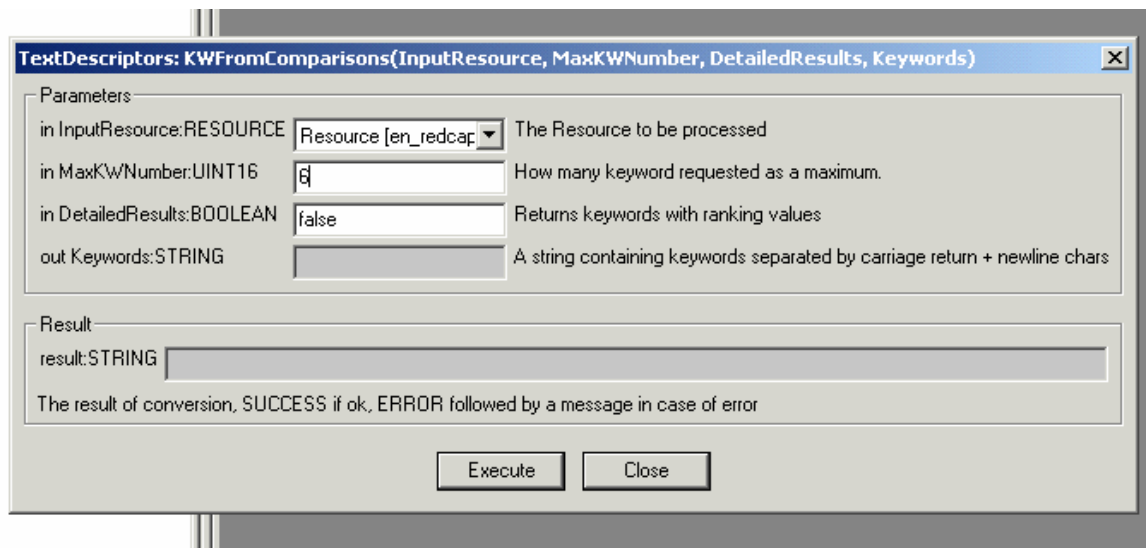
Then select 'Content Processing Plug-in...' command; the following window should appear:



There are 2 functions available:

- KWFromComparisons: extracts single and multi-words making a statistical comparison against a reference corpus (British National Corpus);
- KWFromSemanticAnalysis: extracts single and multi-words making a further analysis with the help of a semantic resource (WordNet).

Both functions accept a parameter, the number of keywords requested:



Clicking on execute makes the plug-in run. Output is given in the 'out' field as a carriage-return separated list of words/multi-words:

The screenshot shows a Windows-style dialog box titled "TextDescriptors: KWFromComparisons(InputResource, MaxKWNumber, DetailedResults, Keywords)". The dialog is divided into two main sections: "Parameters" and "Result".

Parameters section:

- in InputResource:RESOURCE:** A dropdown menu showing "Resource [en_redcap...]" with a description: "The Resource to be processed".
- in MaxKWNumber:UINT16:** A text input field containing the number "6" with a description: "How many keyword requested as a maximum."
- in DetailedResults:BOOLEAN:** A text input field containing the word "false" with a description: "Returns keywords with ranking values".
- out Keywords:STRING:** A text input field containing the string "grandmother||wolf||great_" with a description: "A string containing keywords separated by carriage return + newline chars".

Result section:

- result:STRING:** A text input field containing the word "SUCCESS".
- Below the field is a description: "The result of conversion, SUCCESS if ok, ERROR followed by a message in case of error".

At the bottom of the dialog, there are two buttons: "Execute" and "Close".

16.7 Document Adaptation Plugin (DIPITA)

16.7.1 Main functionalities

Document_adaptation _plug-in is a tool that can transcode text documents between various formats. So far, only PDF to plain text and html to plain text are supported. This document shows the main functionalities provided by a first prototype of the tool forming a minimal user guide.

16.7.2 Relationship with other tools

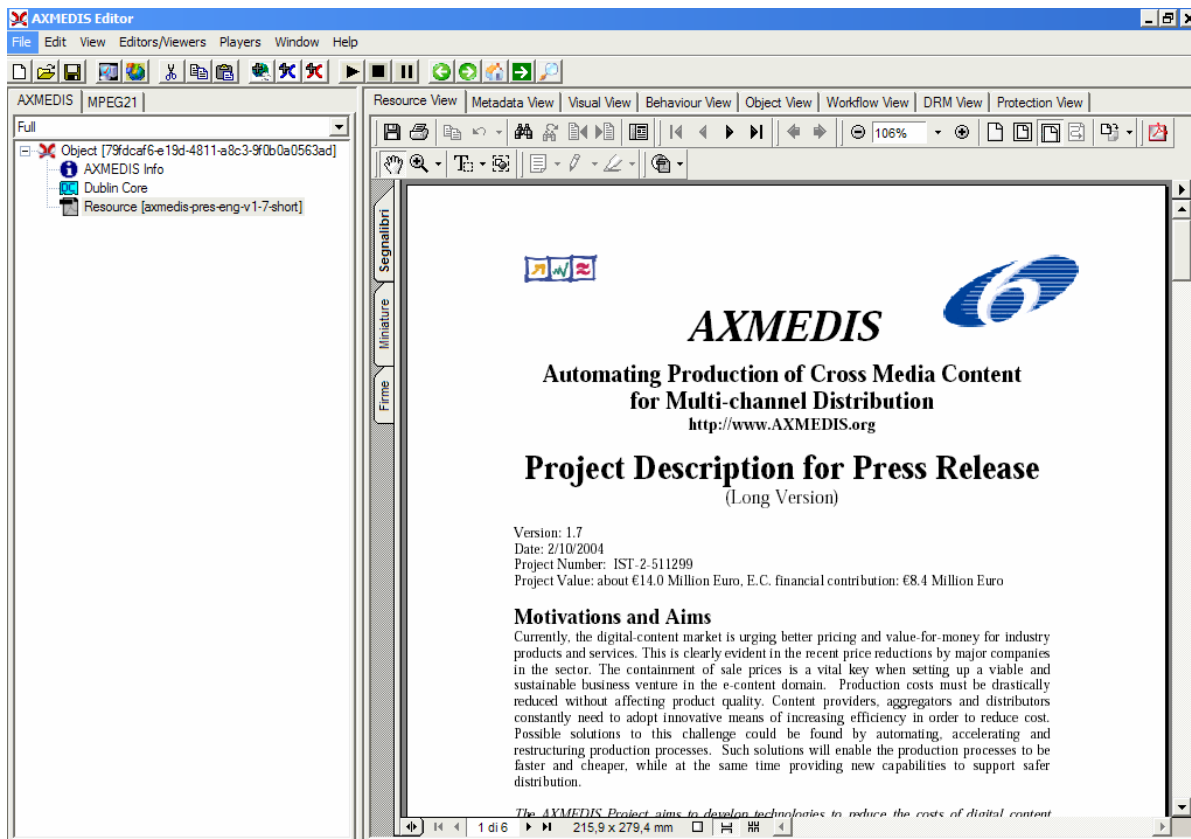
This tool is implemented as a plug-in. Like other plug-ins, its functionality is available via the AXMEDIS Editor and the AXMEDIS Processing Engine.

16.7.3 Detailed description of the functionalities and Screenshots

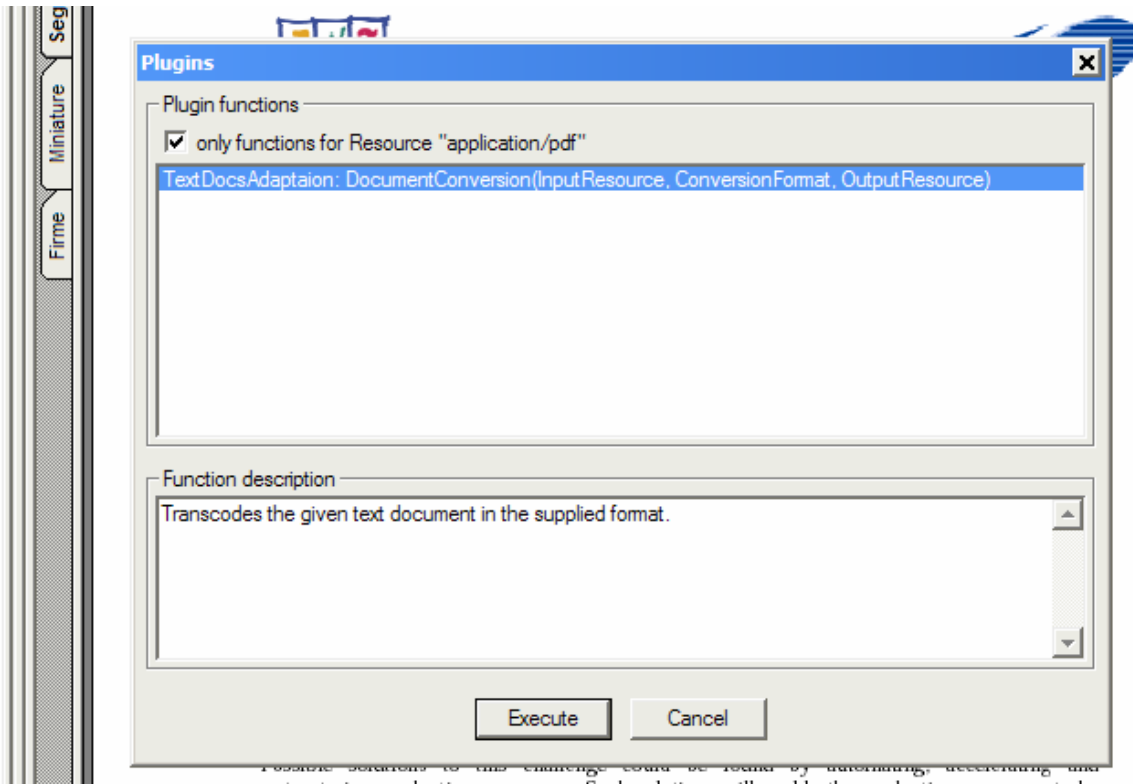
Here's an example on how to use the plug-in with Axeditor.

By now, the plug-in can be applied only to PDF and HTML resources and the output will be text/plain only. In the package there is a sample PDF file to test: AXMEDIS-pres-eng-v1-7-short.pdf.

Create a new AXMEDIS object and add the PDF file as an embedded resource.



Then select 'Content Processing Plug-in...' command; the following window should appear:

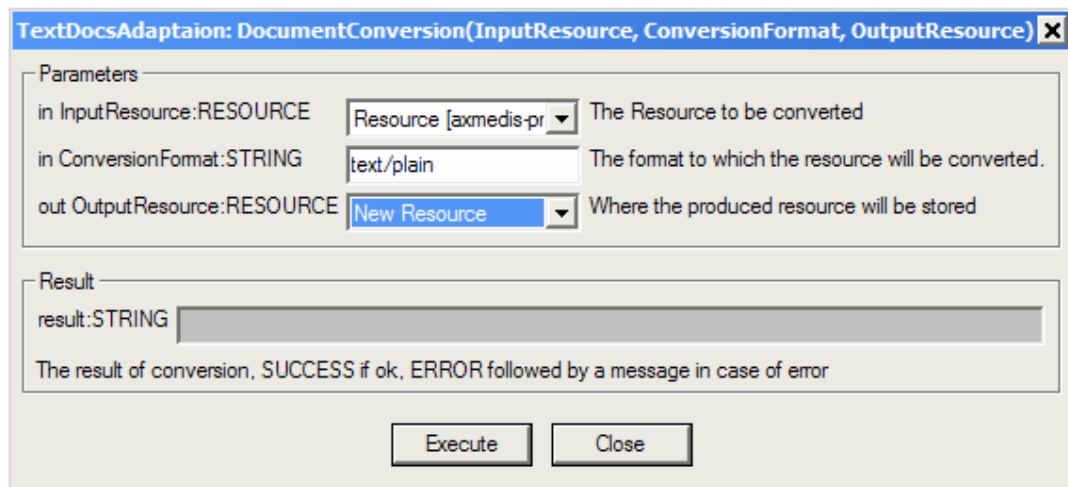


There is only one function available:

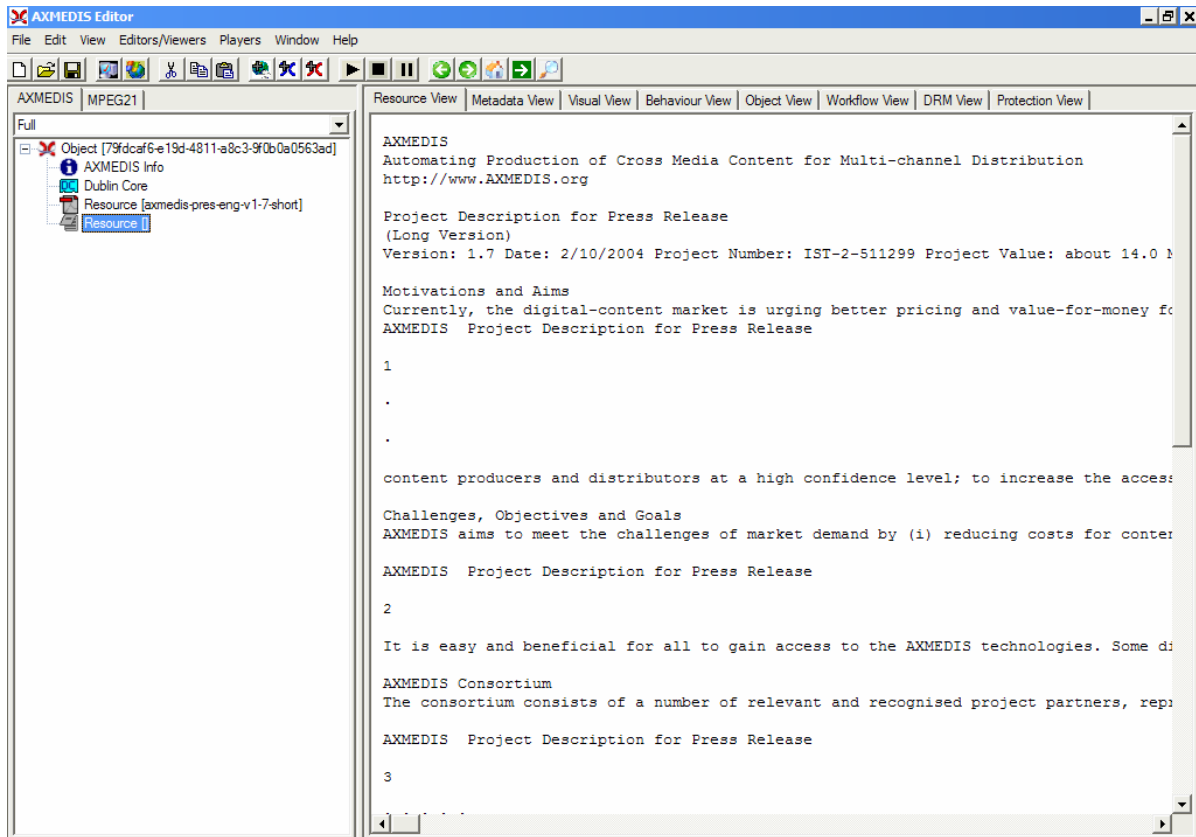
- DocumentConversion: it will make the transcoding to the format specified as the requested parameter.

The only accepted value for the parameter is: text/plain

Make output a new resource, and click execute



Here's the plain text version of the file:



16.8 Audio FP Plugin (FHGIGD)

16.8.1 Main functionalities

Audio_fingerprint __plug-in is a tool that extracts an audio fingerprint of a given audio stream within a multimedia file. The audio stream can be embedded either in a normal audio file (mpg, wav, wma, etc...) or within a video file (mpeg, wmv, avi, etc...).

16.8.2 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.

16.8.3 Detailed description of the functionalities and Screenshots

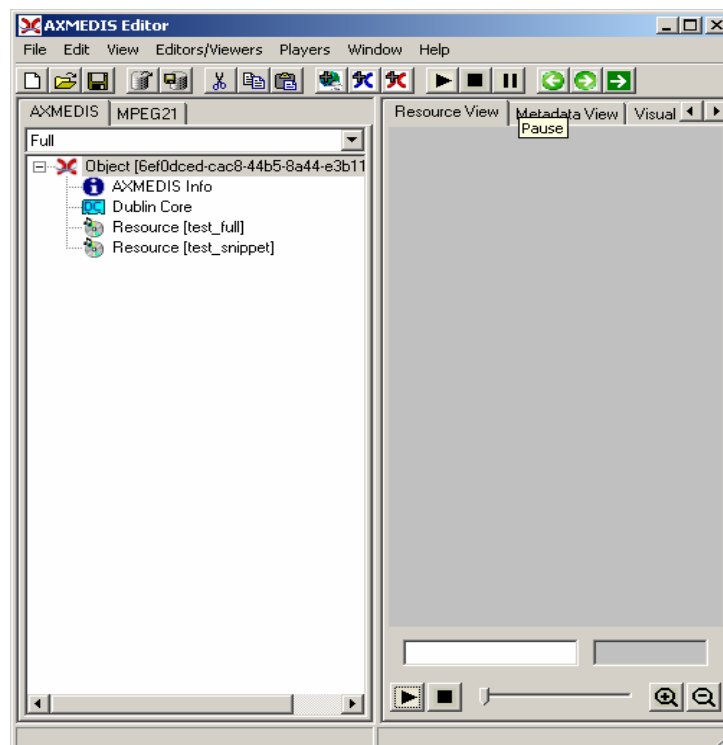
Information related to the installation can be found in file README.txt.

Audio_fingerprint __plug-in is a tool that extracts an audio fingerprint of a given audio stream within a multimedia file. The audio stream can be embedded either in a typical audio file (mpg, wav, wma, etc...) or within a video file (mpeg, wmv, avi, etc...). This document shows the main functionalities provided by a first prototype of the tool forming a minimal user guide. For testing purposes the prototype also includes a basic matching function.

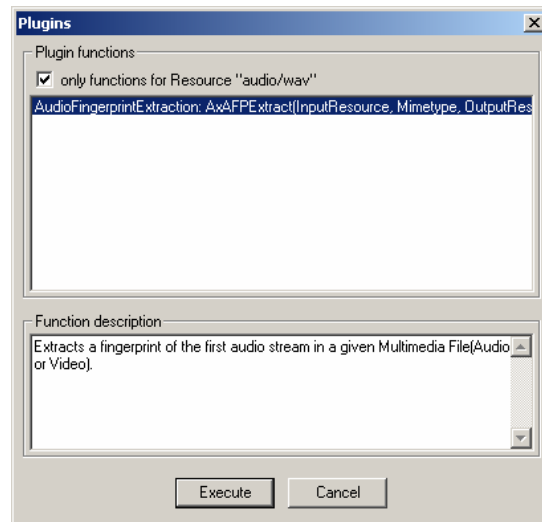
The plug-in can be applied to any audio or video resources, provided this was declared in the mime type attribute of the resource. The output is a binary file containing the fingerprint itself and an image file of any supported format (more than 90 MIME-type image formats). The PNG format is recommended.

For demonstration purposes the package contains a 10 second sample wave file: "test_full.wav" and a 3 second short extract of this file: "test_snippet.wav".

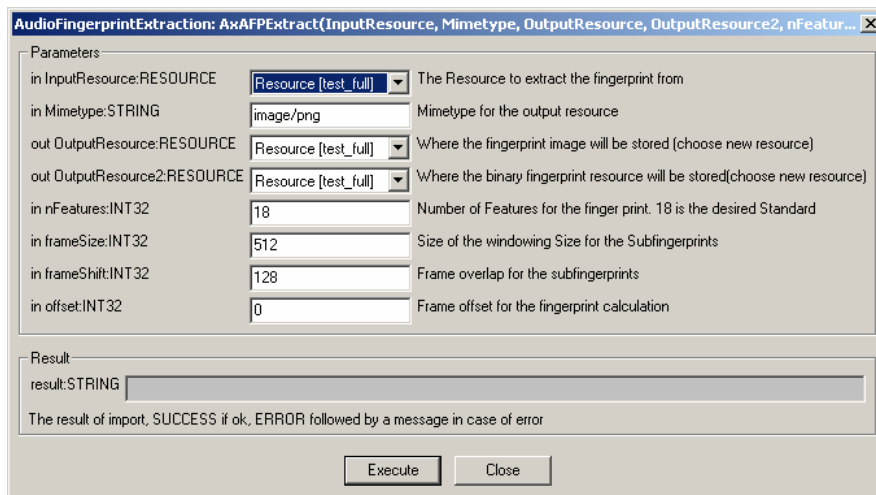
Create a new AXMEDIS object and add the wave files as embedded resources by right clicking on the object.



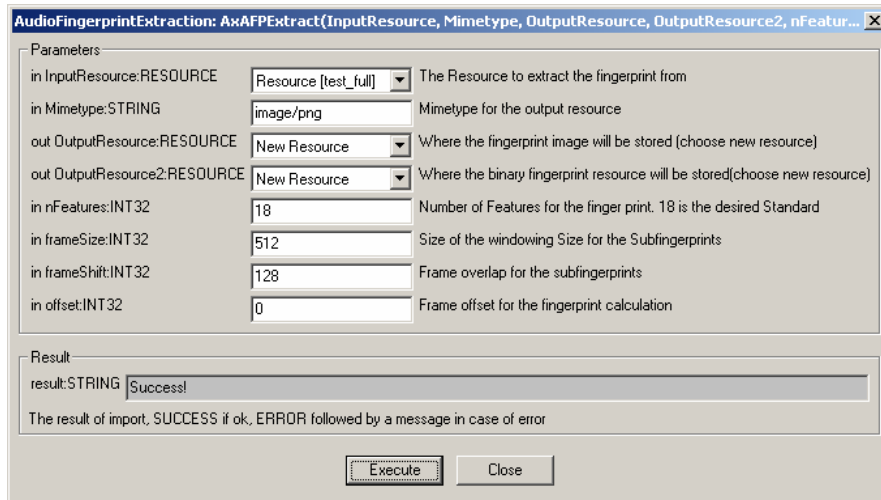
Right click on the “test_full.wav” resource and select ‘Content Processing Plug-ins’



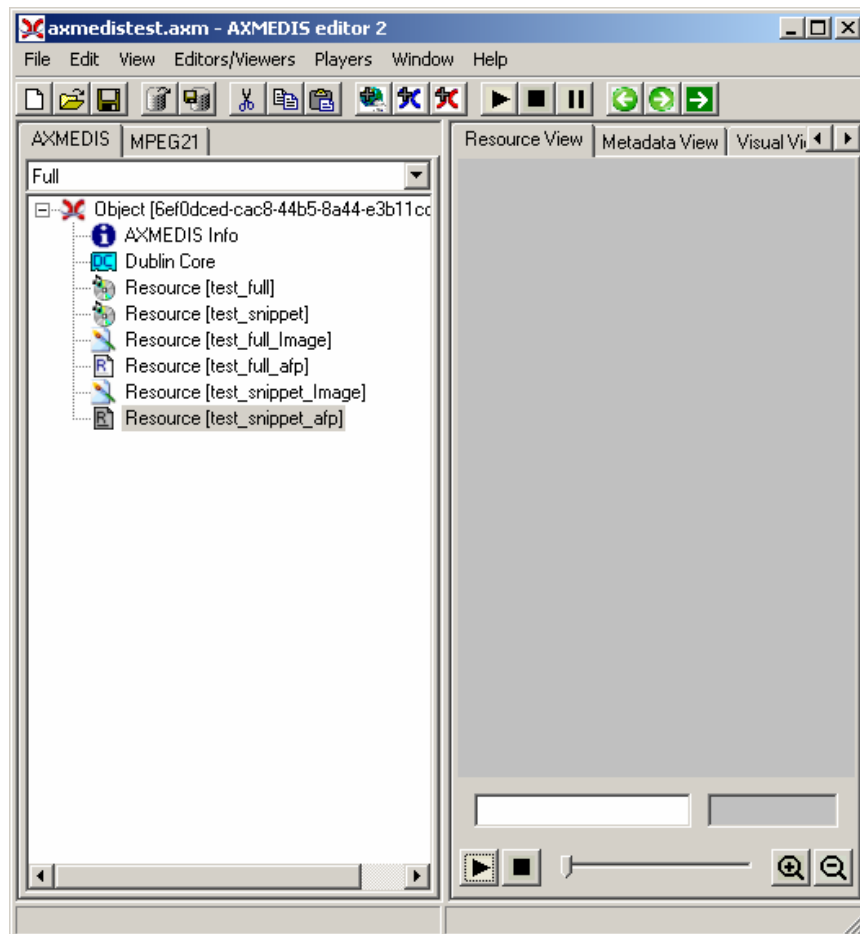
Select ‘AudioFingerprintExtraction’ and the following window should appear:



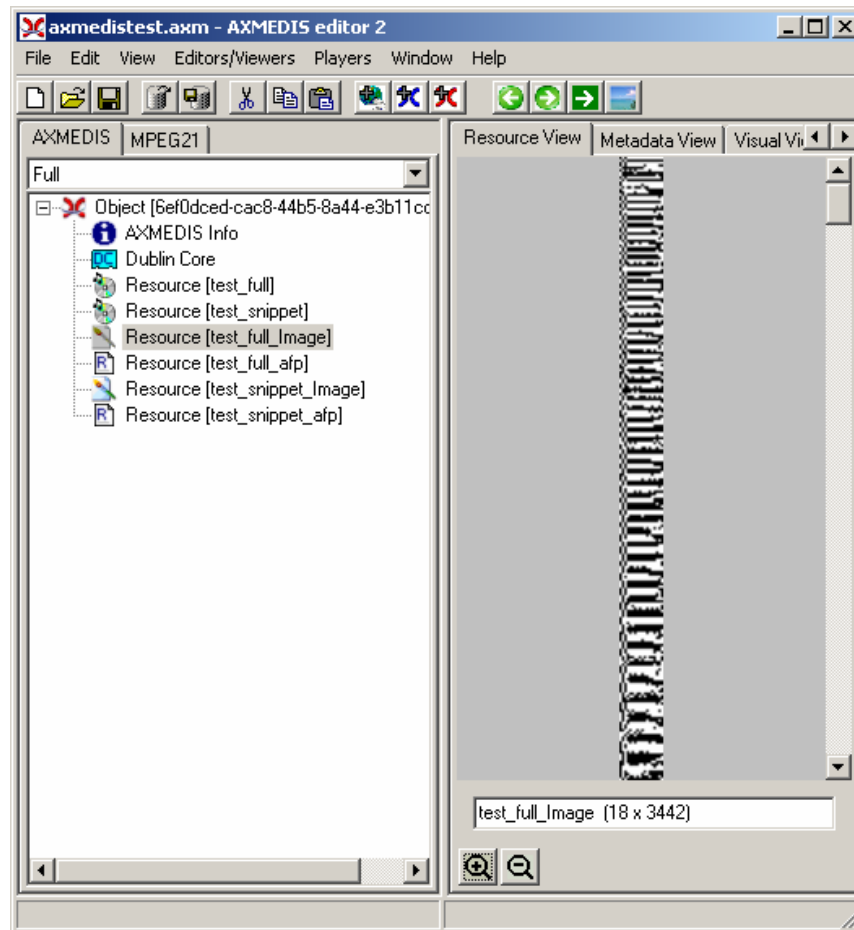
The recommended value for the MIME-type parameter is: image/png. Choose for both outputs a new resource and click execute.



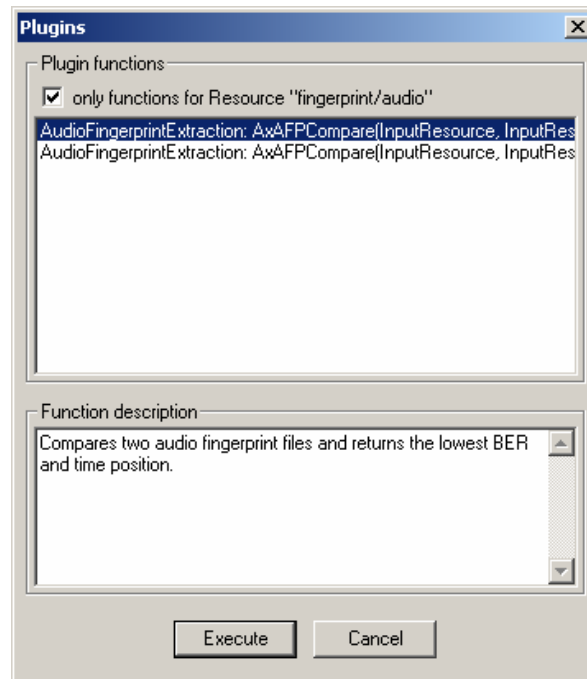
After receiving the “success”-message, close the window and you should have a new image resource and a new resource of the mimetype “fingerprint/audio” in the AXMEDIS editor. It is advised to edit the objects properties with the original resource filename



To view it, just double-click on the image resource. Here's the graphical display of the fingerprint:

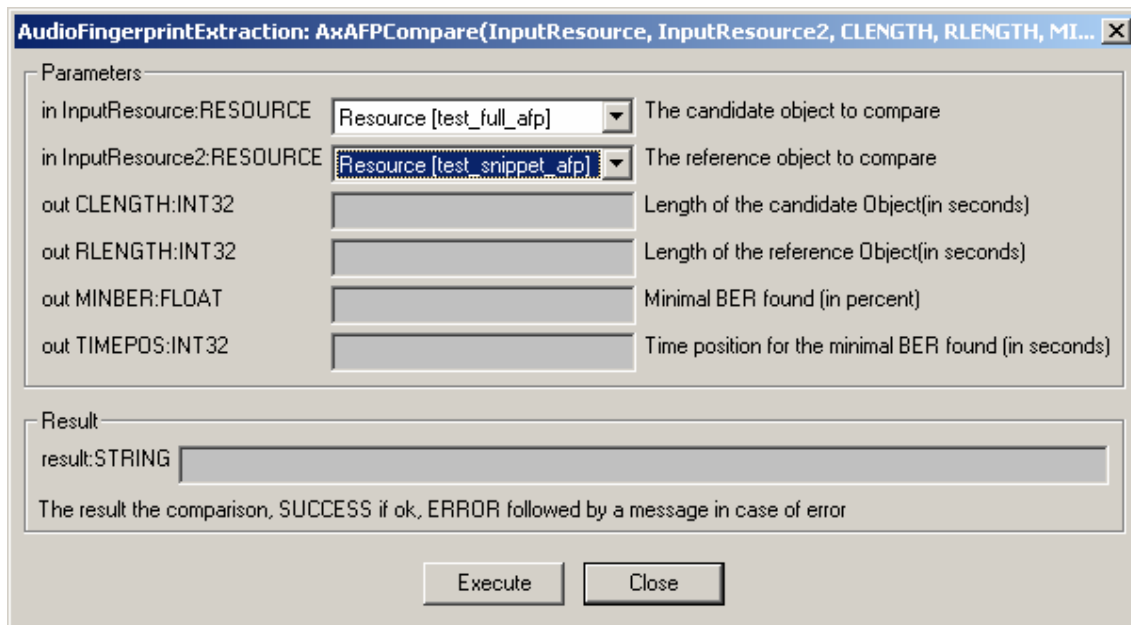


For demonstration purposes a basic matching function was implemented. To apply it the, extraction steps for the “test_snippet.wav” file have to be repeated. After the extraction process has finished right click on any “fingerprint/audio” object, select the “AXAFPCcompare” function and execute it:



On the following window just select the two fingerprint files to be compared and click on execute.

The order is not important; the algorithm automatically decides the candidate and reference object order:



The result shows the length in seconds of the original candidate and reference objects (the audio files the fingerprint was extracted from).

AudioFingerprintExtraction: AxAFPCompare(InputResource, InputResource2, CLENGTH, RLENGTH, MINBER, TIMEPOS)

Parameters

in InputResource:RESOURCE	Resource [test_full_afp]	The candidate object to compare
in InputResource2:RESOURCE	Resource [test_snippet_afp]	The reference object to compare
out CLENGTH:INT32	2	Length of the candidate Object(in seconds)
out RLENGTH:INT32	9	Length of the reference Object(in seconds)
out MINBER:FLOAT	9.175	Minimal BER found (in percent)
out TIMEPOS:INT32	2	Time position for the minimal BER found (in seconds)

Result

result:STRING Success!

The result the comparison, SUCCESS if ok, ERROR followed by a message in case of error

Execute Close

The MINBER value is the minimal BER found during the comparison. The TIMEPOS is the time position for the minimal BER.

The result value will show “Success!” if the comparison was performed without errors. Else an error message will display. Please note, that the “Success!” message is a display of the technical success of the comparison process and not the matching probability.

16.9 M2ANY - Audio FP Plugin (FHGIGD)

16.9.1 Main functionalities

M2ANYAudio_fingerprint_plugin is a tool that extracts an audio fingerprint of a given audio stream within a WAV file. This document shows the main functionalities provided by a first prototype of the tool forming a minimal user guide. A matching function is still in development.

16.9.2 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.

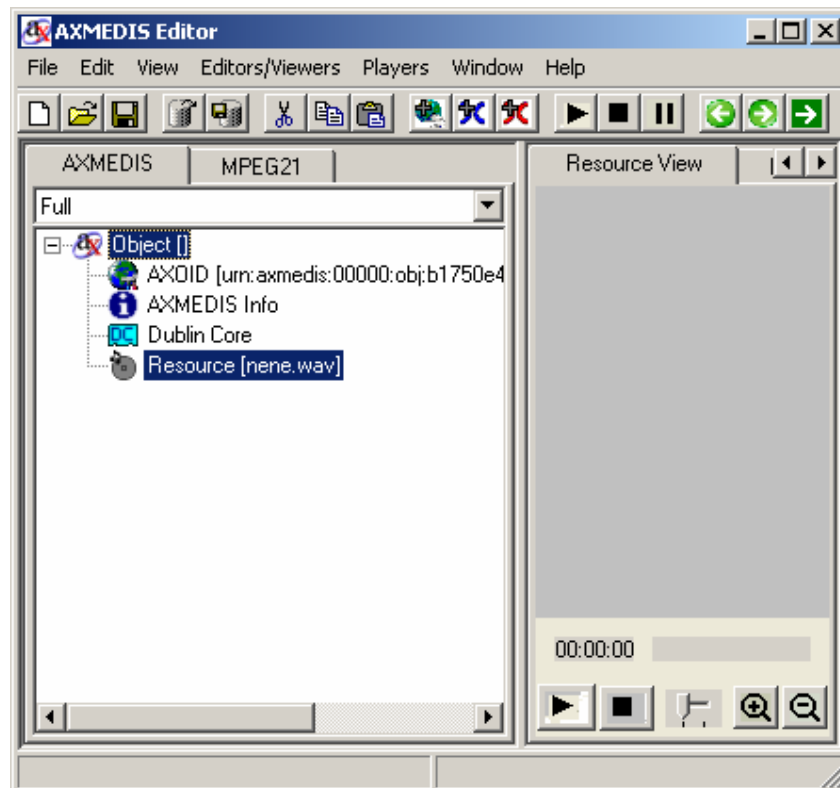
16.9.3 Detailed description of the functionalities and Screenshots

CAVEAT: This guide assumes you licensed the corresponding technology from M2ANY! The licensed executables (cfymain.exe, xtrmain.exe, cfy.dll, xtr.dll and asign.dll) have to be installed in the plug-in directory! You also need to indicate the path to an existing fingerprint database which is also provided by M2ANY.

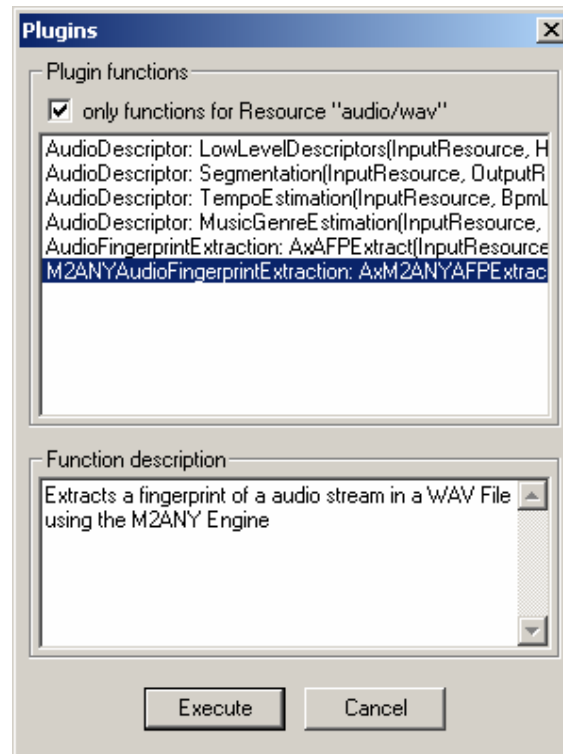
Information related to the installation can be found in file README.txt.

The plug-in can be applied to WAV files, provided this was declared in the MIME type attribute of the resource. The output is a binary file containing the fingerprint in binary format.

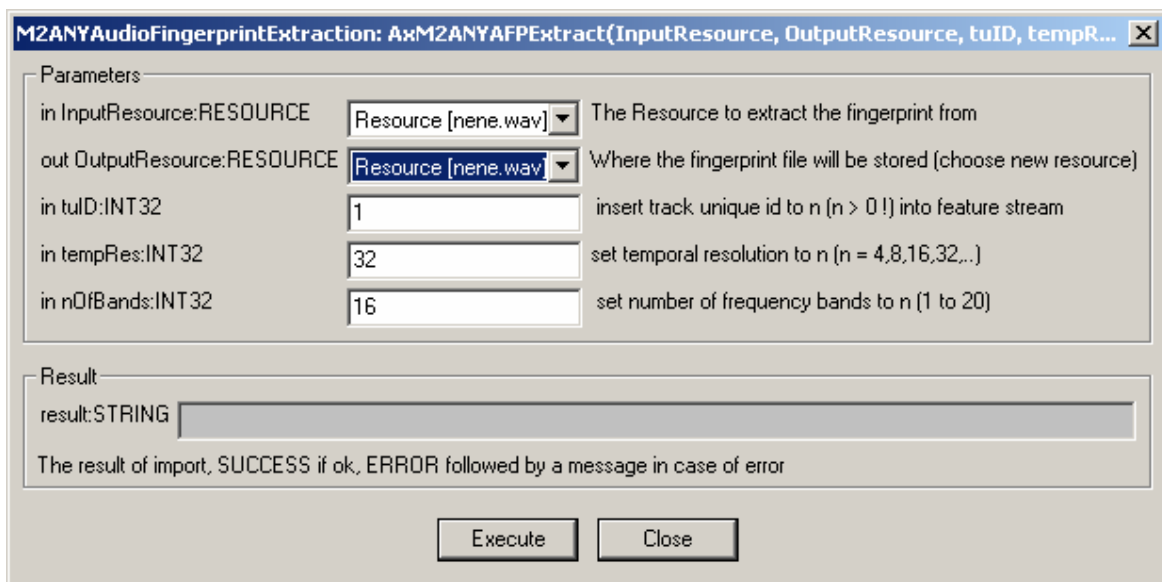
Create a new AXMEDIS object and add a wave file as an embedded resource by right clicking on the object. The provided audio file excerpt (“nene.wav”) is used here as a complete fingerprint of the complete song is contained within the M2ANY Database.



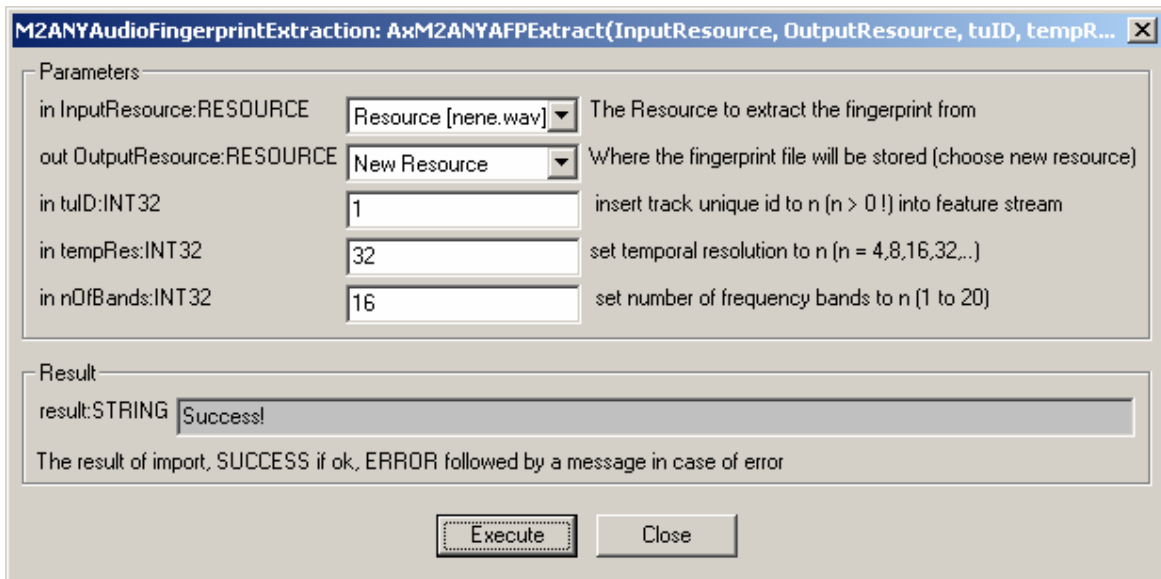
Right click on the WAV resource and select 'Content Processing Plug-ins'



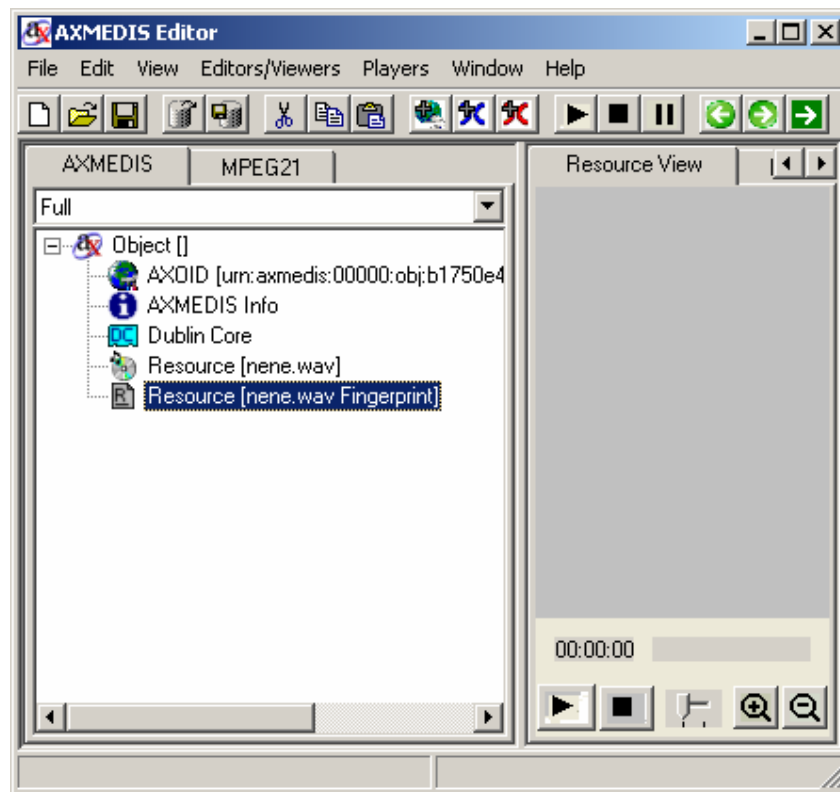
Select 'M2ANYAudioFingerprintExtraction' and click "Execute" the following window should appear:



Choose for the output a new resource and click execute.

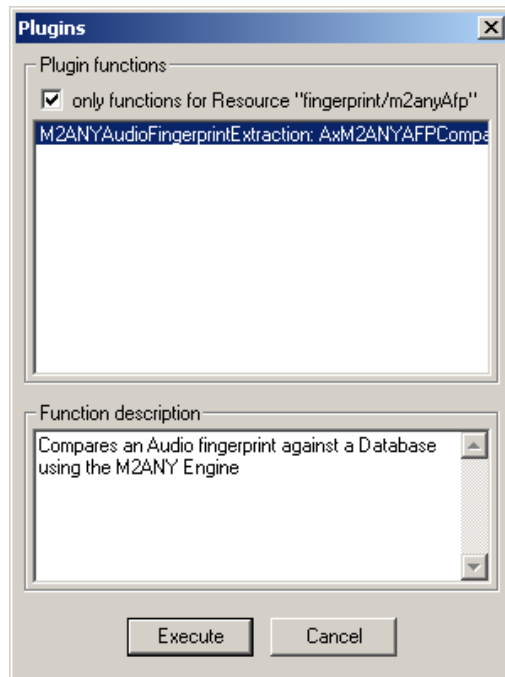


A console window will shortly appear indicating the extraction process. After receiving the “success”-message, close the window and you should have a new resource of the MIME type “fingerprint/m2anyAfp” in the AXMEDIS editor. It is advised to edit the objects properties with the original resource filename

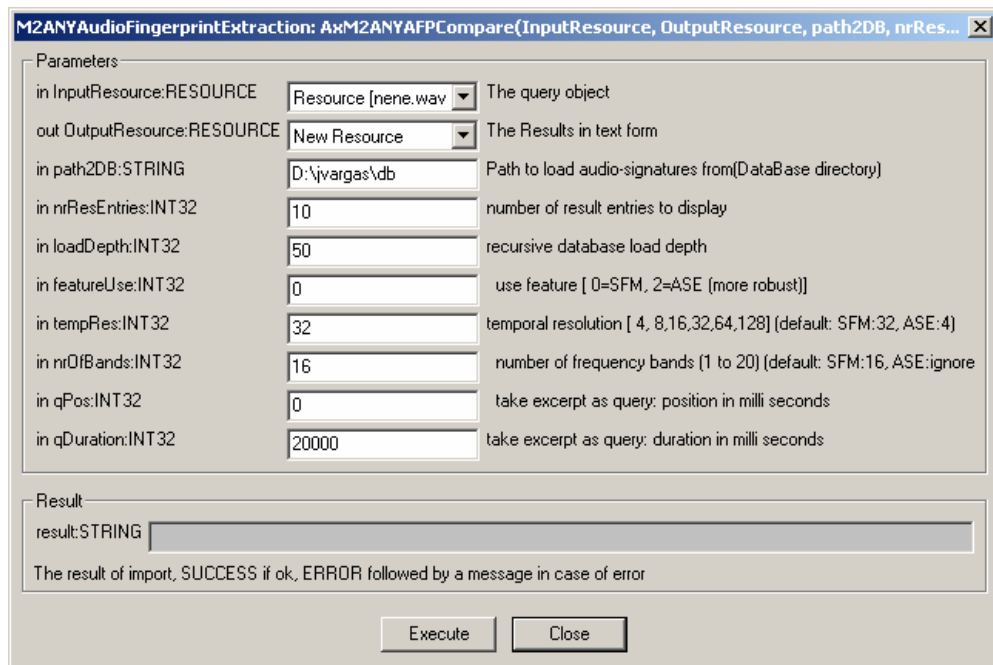


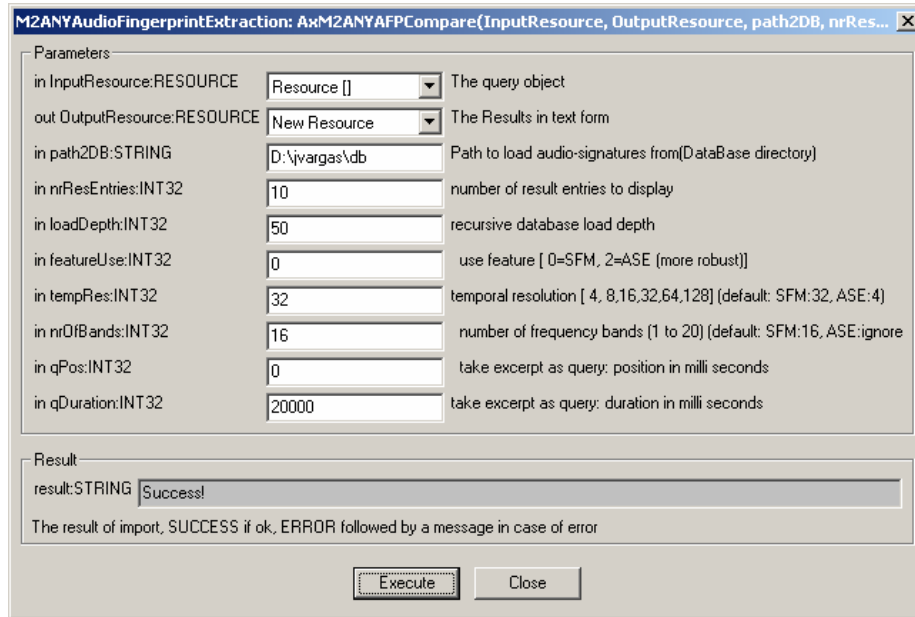
For demonstration purposes a basic matching function was included. For this section the fingerprint previously generated will be used.

After the extraction process has finished right click on any “fingerprint/ m2anyAfp” object, select the “M2ANYAXAFPCmpare” function and execute it:



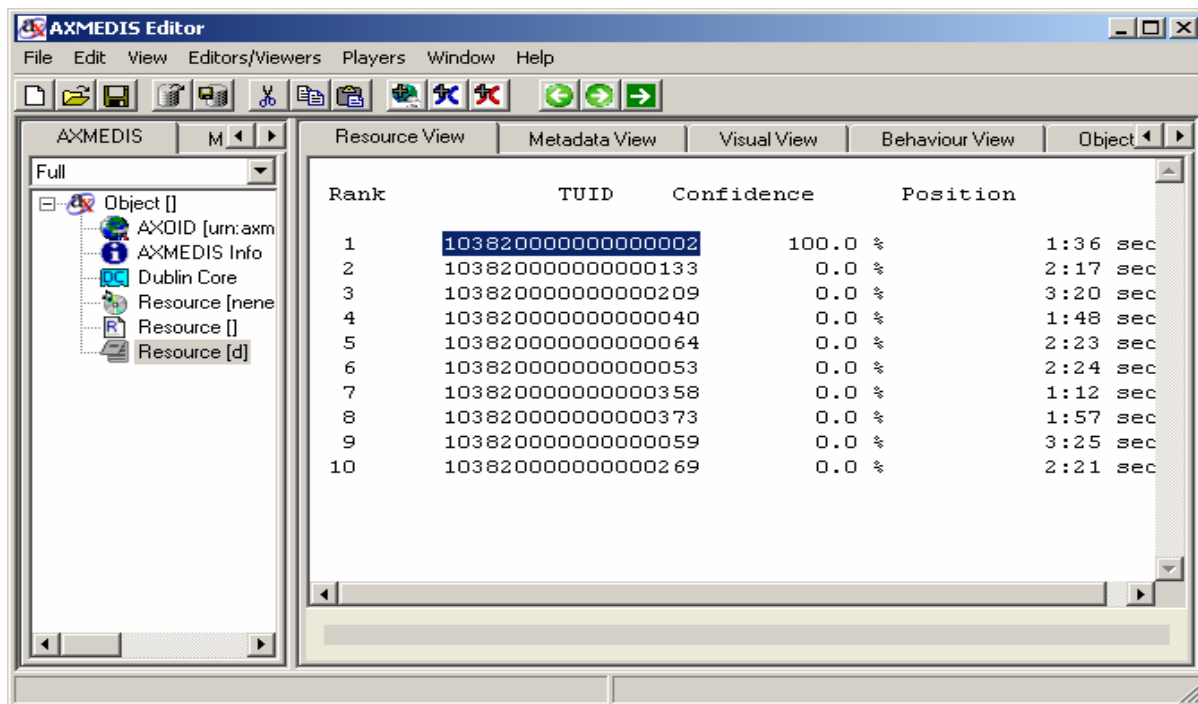
On the following window select a new resource as the output resource and provide the path to an existing database directory (containing at least 10 fingerprint files in it) and click on execute:





The result value will show “Success!” if the comparison was performed without errors. Else an error message will display. The resulting resource will be a text file. Please edit its properties and add an arbitrary content ID. This prevents the AXEditor from crashing. This issue is soon to be fixed.

The matching TUID (Track unique ID) for the excerpt should be the ID: 103820000000000002 which belongs to the song “7 Seconds” by Neneh Cherry. (Unfortunately the provided binary demonstration package does not allow resolving the TUIDs to their source files.)



16.10 Video FP Plugin (FHGIGD)

16.10.1 Main functionalities

Video_fingerprint_plugin is a tool that extracts a fingerprint of a given video stream. The video stream can be embedded in a video file (mpeg, wmv, avi, etc...).

This document shows the main functionalities provided by a first prototype of the tool forming a minimal user guide.

16.10.2 Relationship with other tools

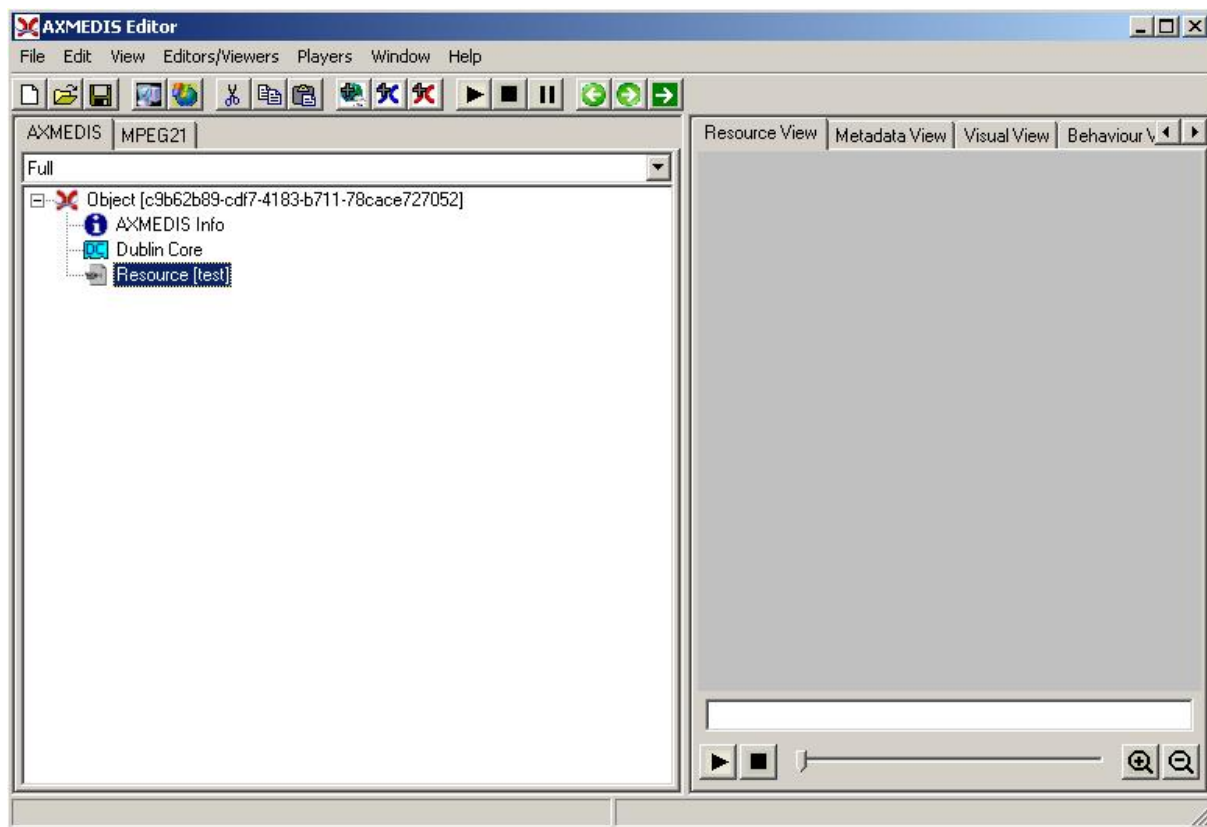
This tool is implemented as a plug-in. Like other plug-ins, its functionality is available via the AXMEDIS Editor and the AXMEDIS Processing Engine.

16.10.3 Detailed description of the functionalities and Screenshots

Information related to the installation can be found in file README.txt.

The plug-in can be applied to any video resources, provided this was declared in the mime type attribute of the resource. The output can be any mimetype within the image/* format. The **PNG** format is recommended. In the package there is a sample mpg file to do a test: <test.mpg>.

Create a new AXMEDIS object and with a right click, add the mpg file as an embedded resource.



With a right Click on the resource, select 'Content Processing Plug-ins'. Then you should search for the option 'VideoFingerprintExtraction' and click execute.

After the selection, the following window should appear:

The screenshot shows a dialog box titled "videoFingerprintExtraction: AxVFPEXtract(InputResource, Mimetype, OutputResource, f...)". It contains two main sections: "Parameters" and "Result".

Parameters:

- in InputResource:RESOURCE:** A dropdown menu with "Resource [test]" selected. Description: "The Resource to extract the fingerprint from".
- in Mimetype:STRING:** An empty text field. Description: "Mimetype for the output resource".
- out OutputResource:RESOURCE:** A dropdown menu with "Resource [test]" selected. Description: "Where the produced resource will be stored".
- in frames:INT32:** A text field with "50" entered. Description: "Number of frames to be processed".

Result:

result:STRING [Empty text field]

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

Buttons: "Execute" and "Close".

The recommended value for the Mimetype parameter is: image/png.
Make output a new resource, select the desired number of frames to be processed and click execute.

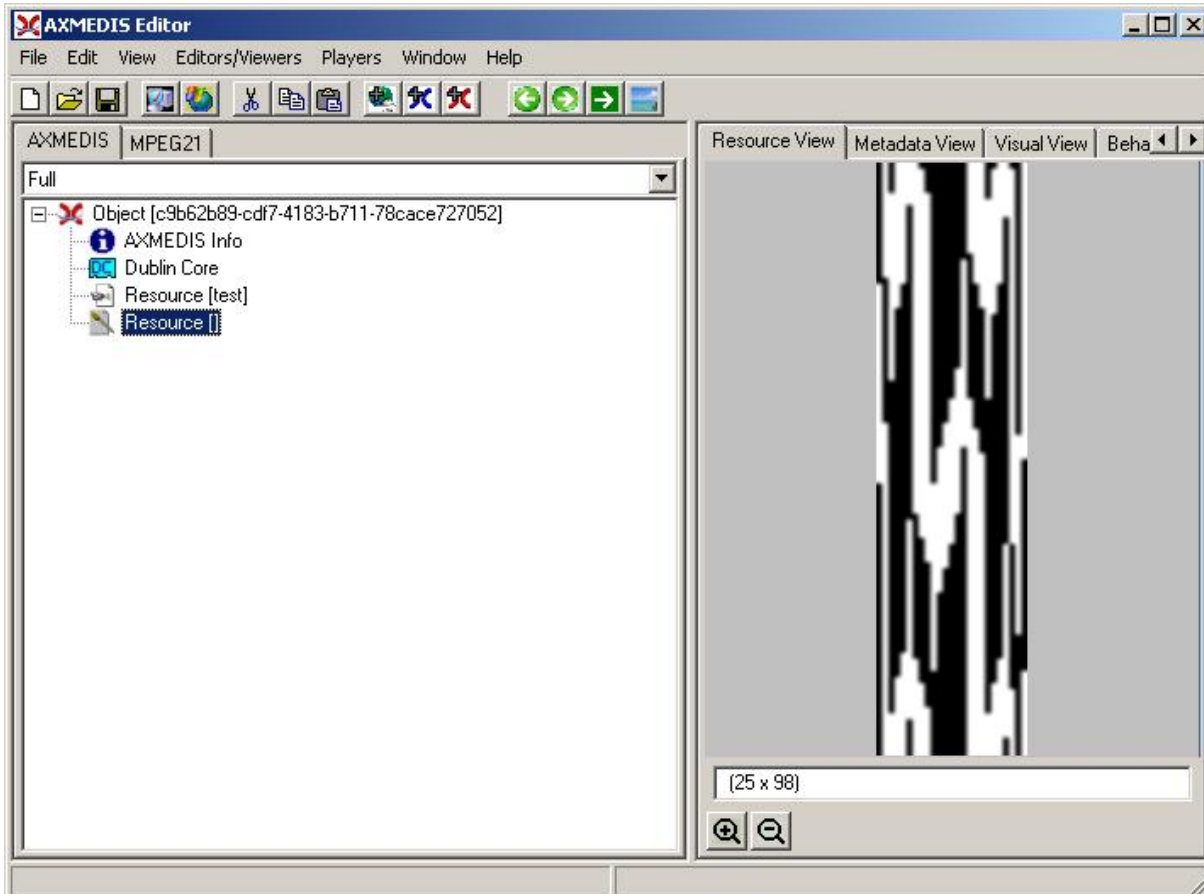
The screenshot shows the same dialog box as above, but with the following changes:

- in Mimetype:STRING:** The text field now contains "image/png".
- out OutputResource:RESOURCE:** The dropdown menu now has "New Resource" selected.
- in frames:INT32:** The text field now contains "100".

The "Result" section and buttons remain the same as in the previous screenshot.

After receiving the 'success' message, close this window. You should have a new resource in the AXMEDIS editor. With a right click in these resource, select 'Properties' and declare its mimetype as the same you declared on the execution window.

With a double click in the resource, the following graphical of the fingerprint will be showed:



16.11 Generic Resource Files FP Plugin (FHGIGD)

16.11.1 Main functionalities

GenericFiles_fingerprint _plug-in is a tool that calculates a fingerprint (a cryptographic hash) for a given arbitrary file. This document shows the main functionalities provided by a first prototype of the tool forming a minimal user guide.

16.11.2 Relationship with other tools

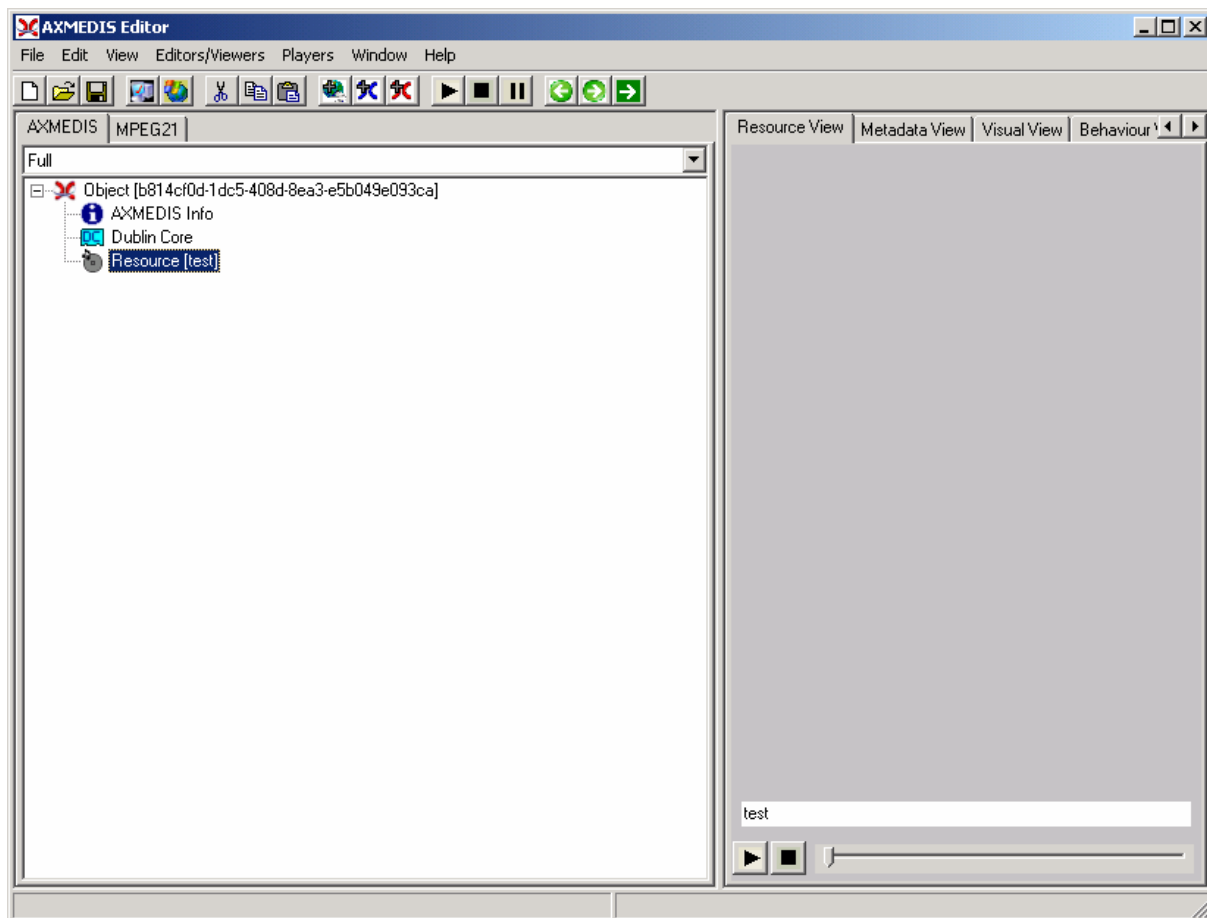
This tool is implemented as a plug-in. Like other plug-ins, its functionality is available via the AXMEDIS Editor and the AXMEDIS Processing Engine.

16.11.3 Detailed description of the functionalities and Screenshots

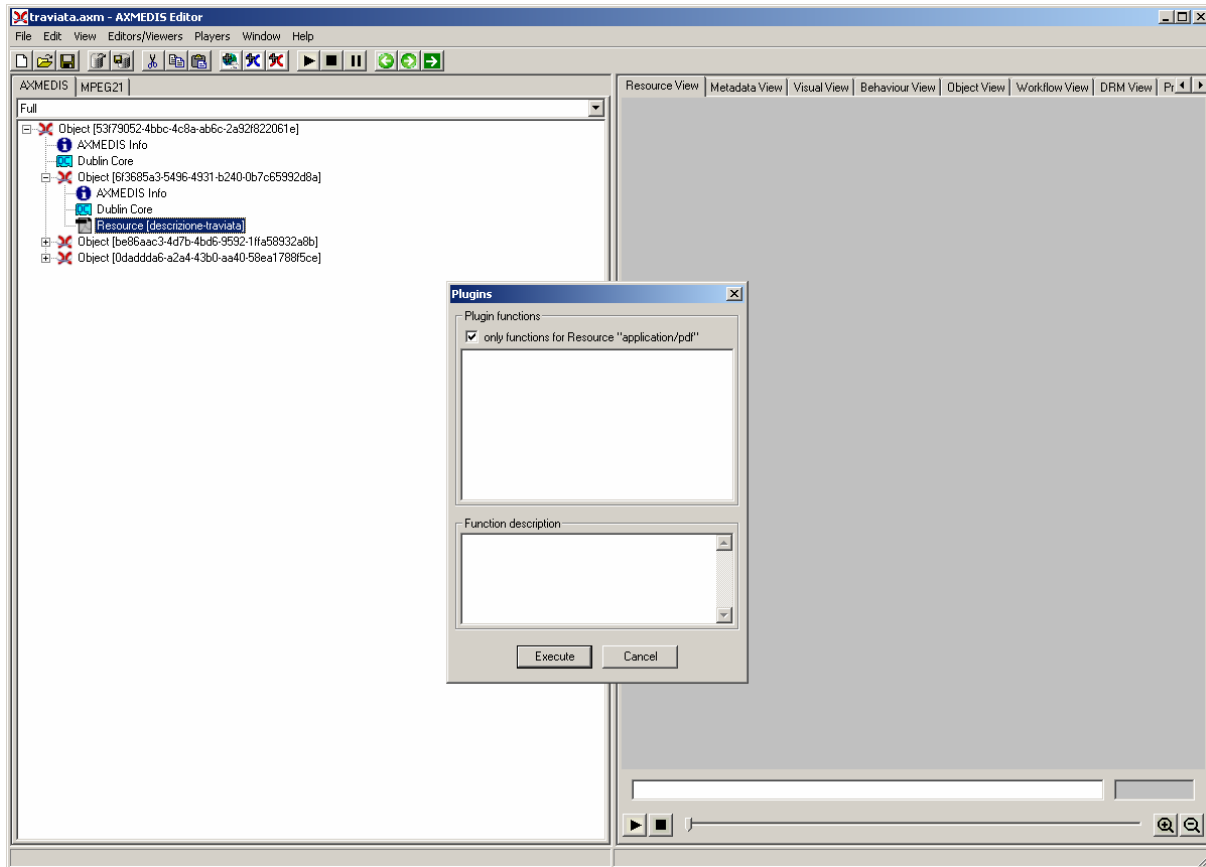
Information related to the installation can be found in file README.txt.

The plug-in can be applied to any content type. The output is a string. In the package there is a test file: test.pdf.

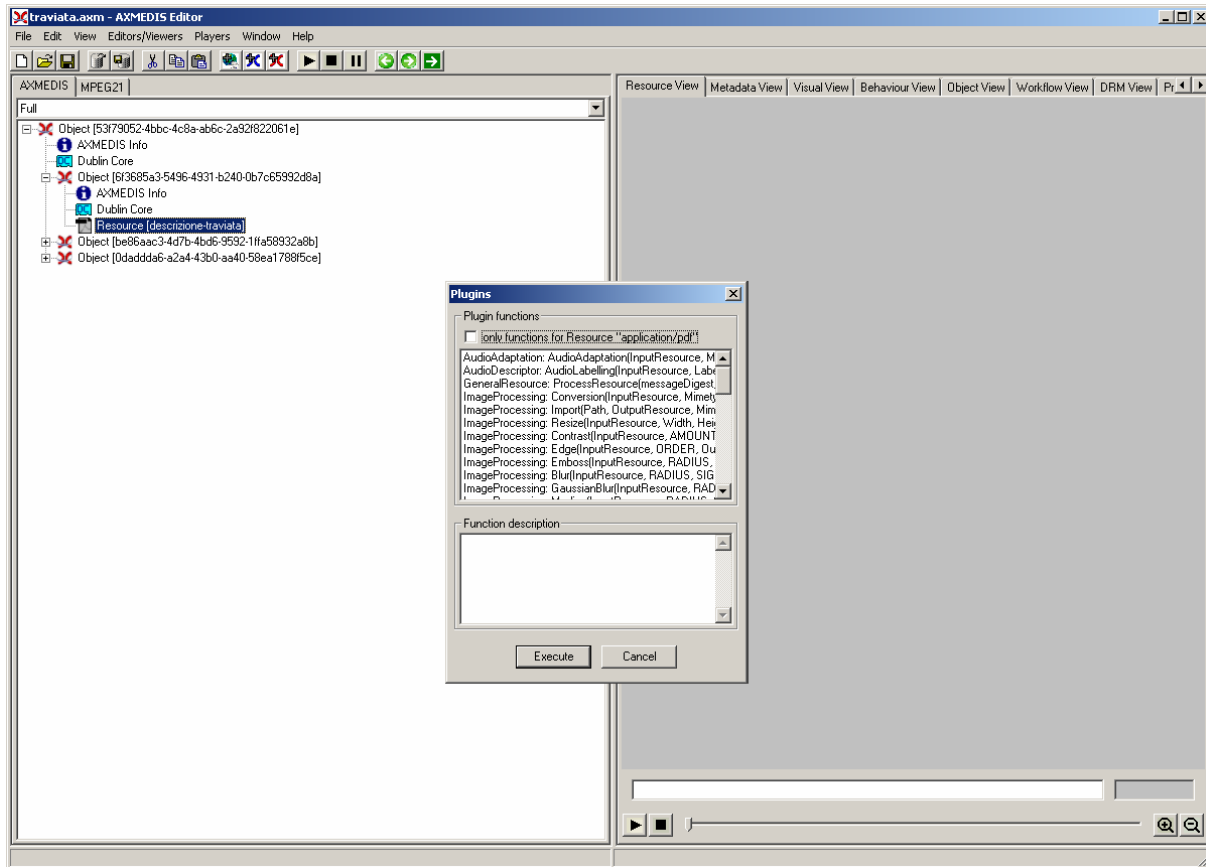
Create a new AXMEDIS object and add the wav file as an embedded resource.



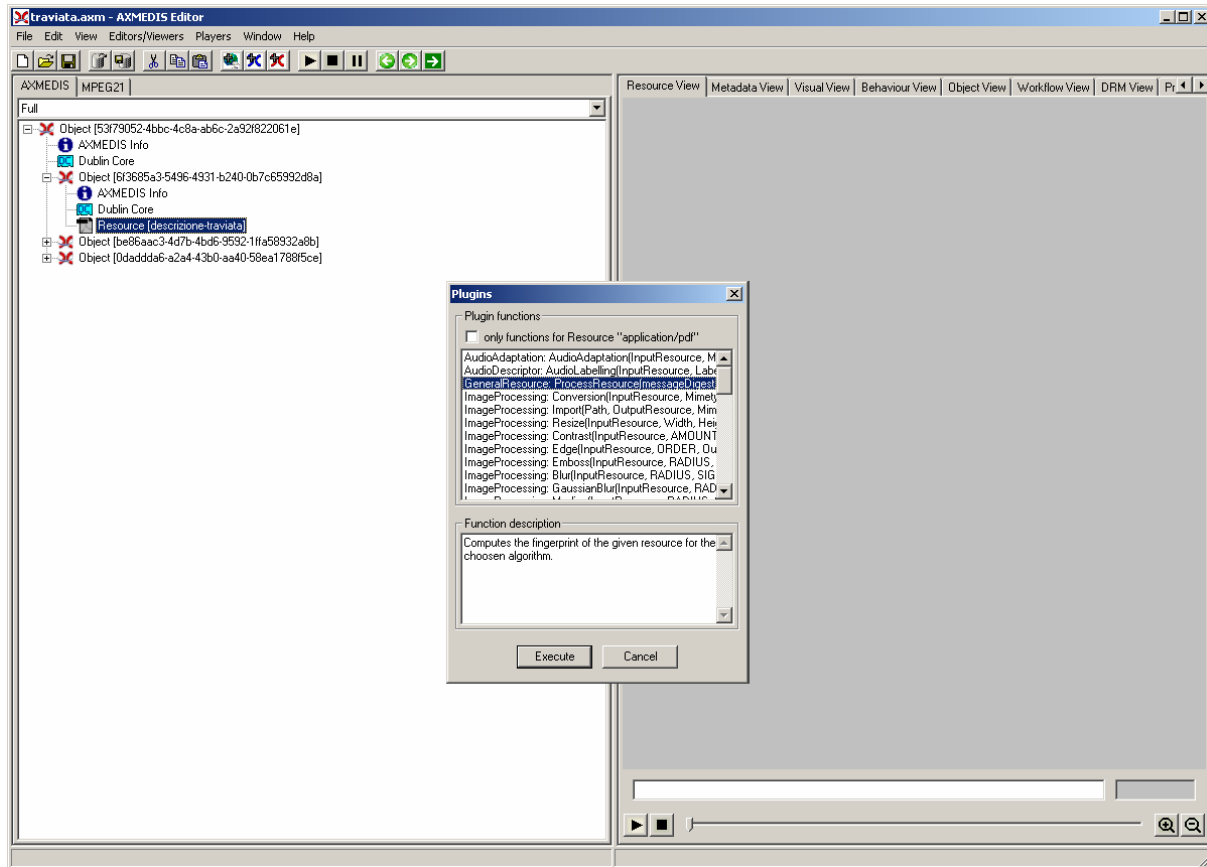
Right Click on the resource and select 'Content Processing Plug-ins'. The following window should appear:



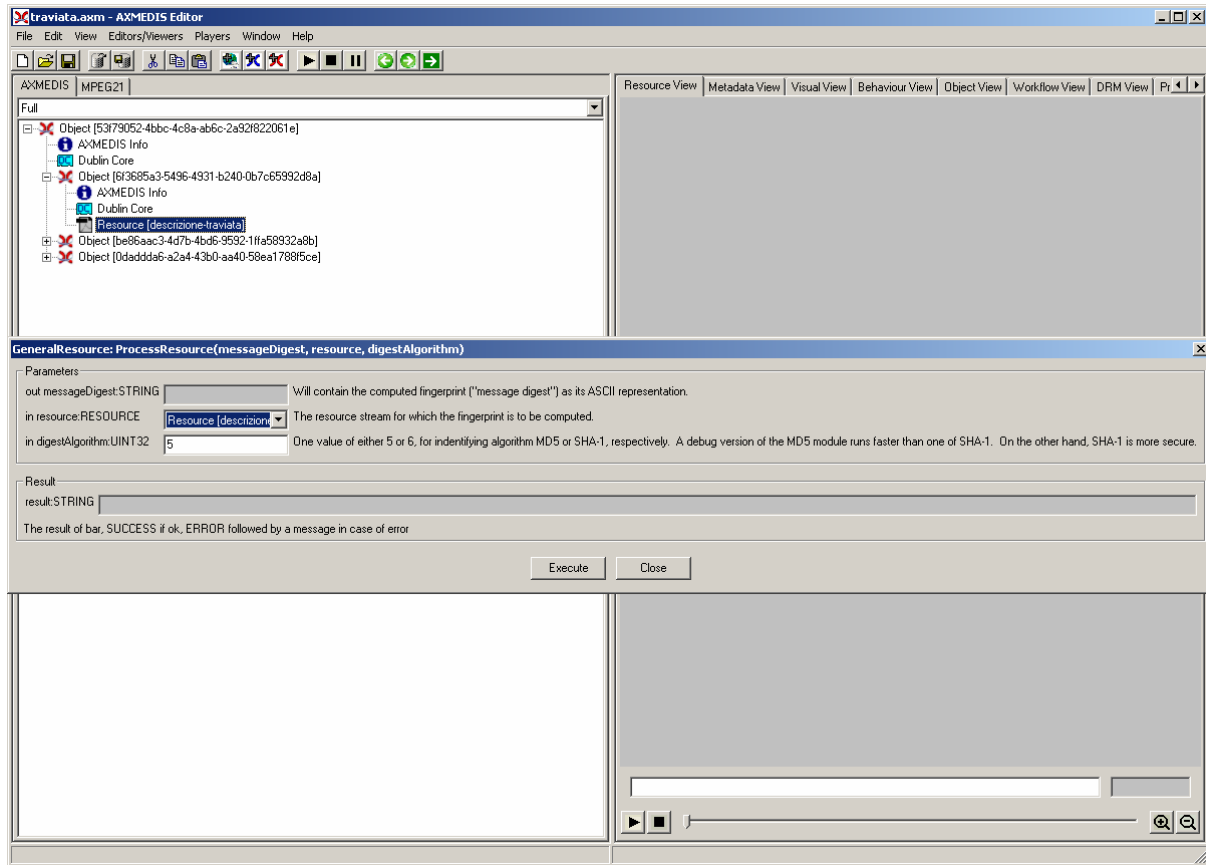
Unselect the check-box (“only functions for Resource XXX”):



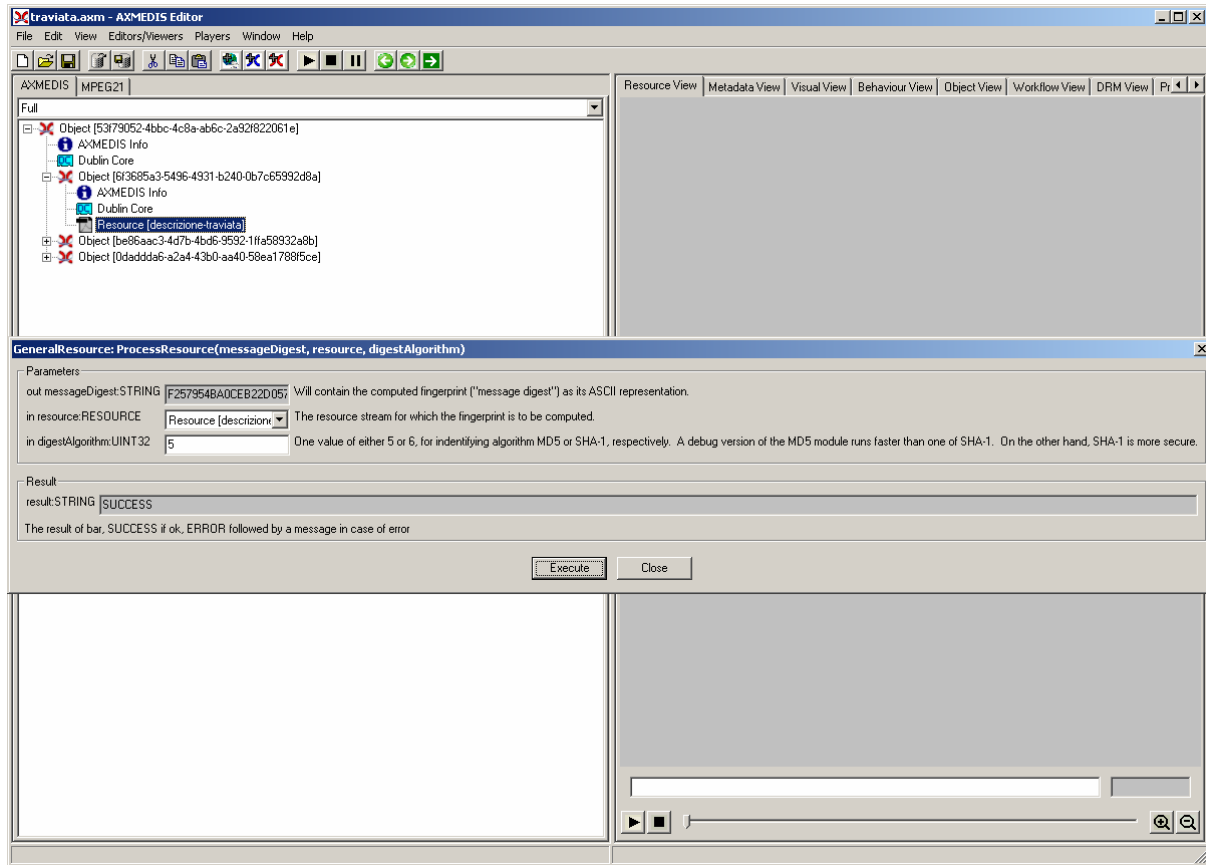
Select 'GenericResource::ProcessResourceMessageDigest'



Select the available algorithm (5: MD5, 6: SHA-1):



So far, the result is calculated and shown in a dialog:



16.12 Video Descriptor Plugin (FHGIGD)

16.12.1 Main functionalities

MP7 Videodescriptor Extraction plug-in is a tool that extracts a MPEG-7 XML descriptor from a given MPEG video file. This document shows the main functionalities provided by a first prototype of the tool forming a minimal user guide.

16.12.2 Relationship with other tools

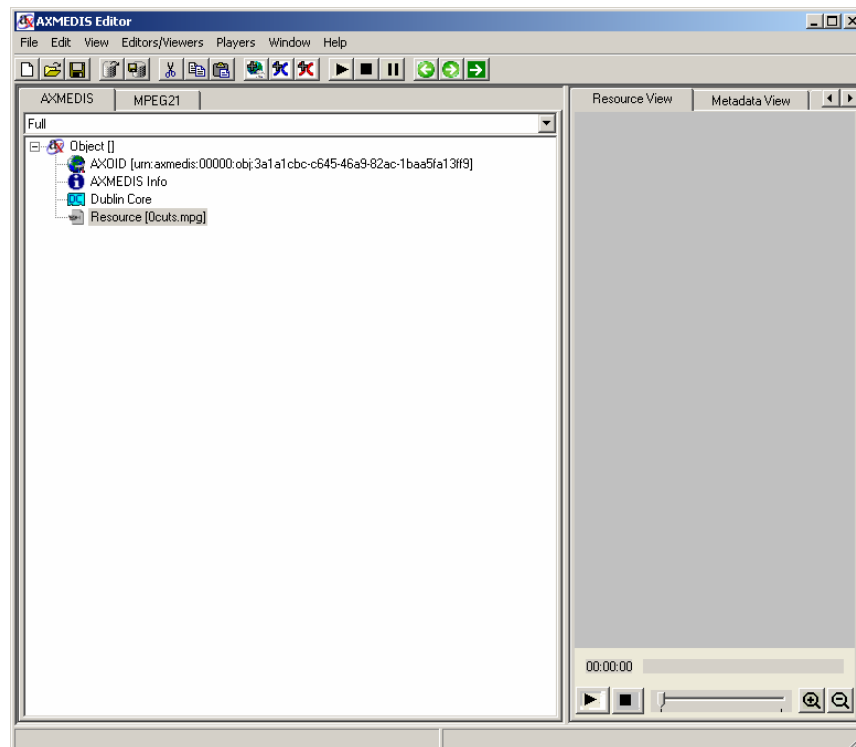
This tool is implemented as a plug-in. Like other plug-ins, its functionality is available via the AXMEDIS Editor and the AXMEDIS Processing Engine.

16.12.3 Detailed description of the functionalities and Screenshots

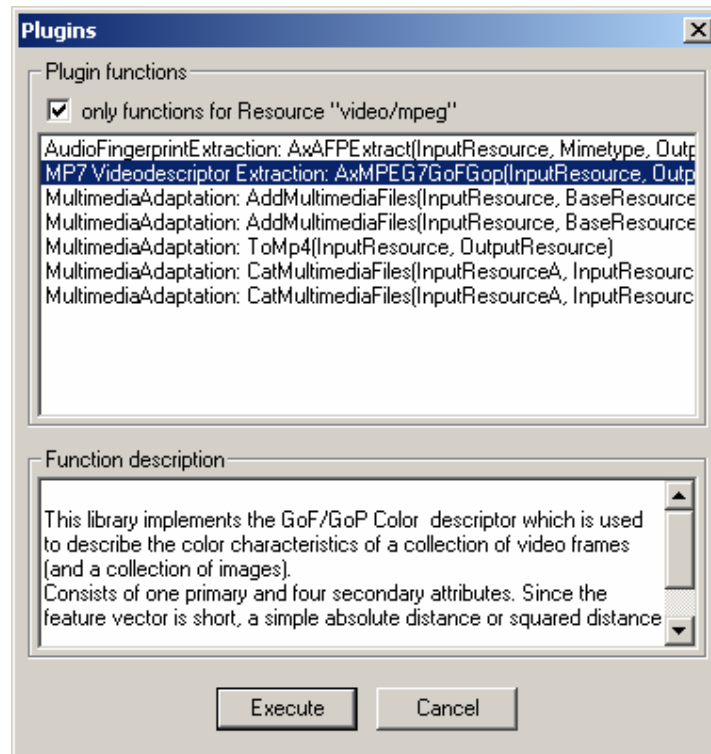
Information related to the installation can be found in file README.txt.

The plug-in can be applied to video MPEG resources, provided this was declared in the mime type attribute of the resource. The output is a XML file containing the descriptor data.

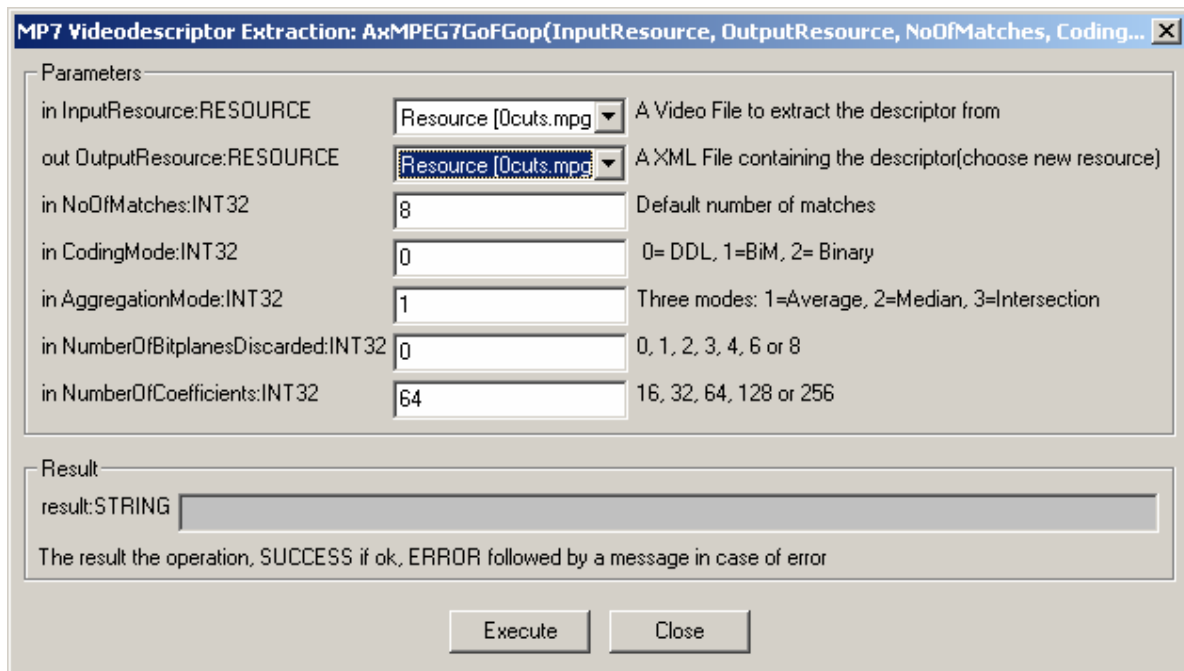
Create a new AXMEDIS object and add a MPEG file as a embedded resource by right clicking on the new created object.



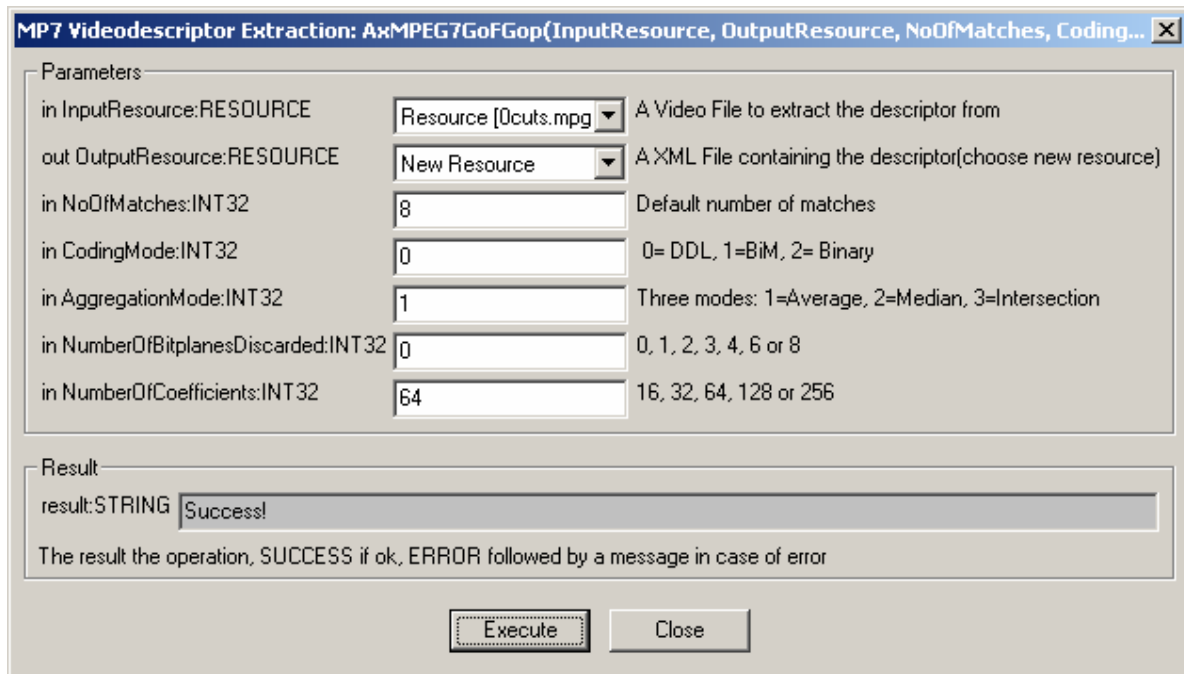
Right click on the MPEG resource and select 'Content Processing Plug-ins'



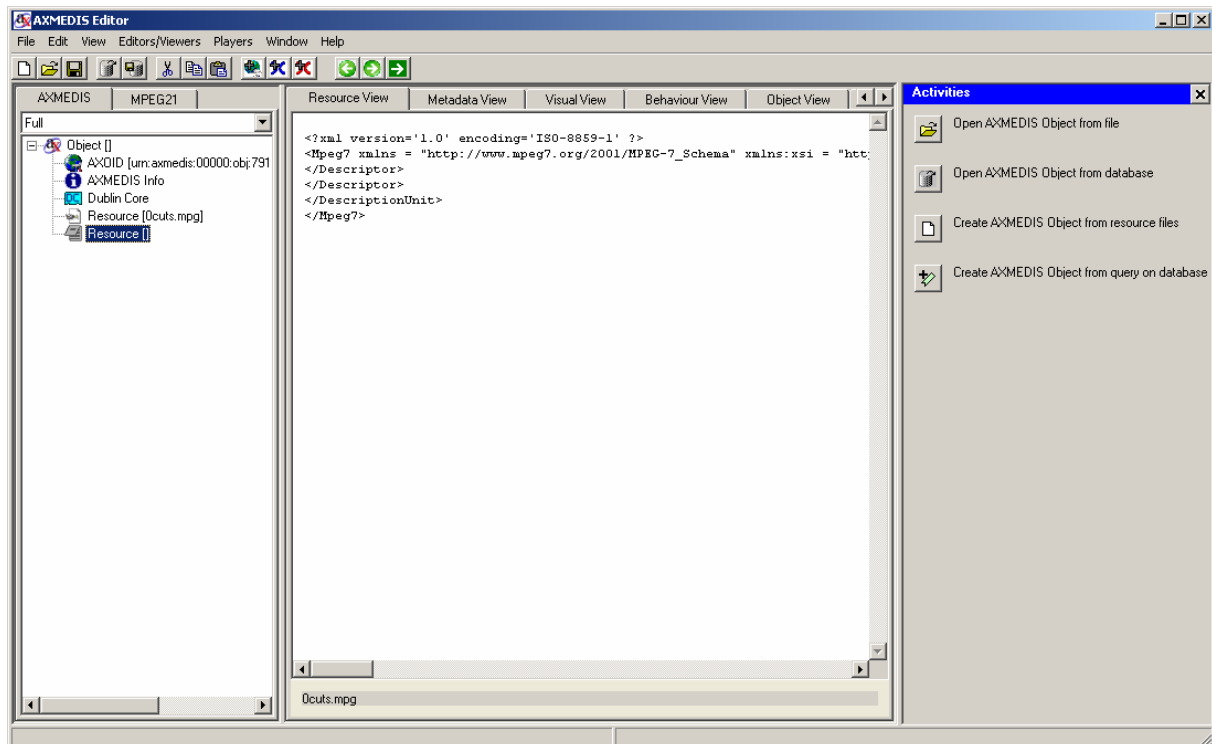
In the upcoming menu select 'MP7 Videodescriptor Extraction: AxMPEG7GoFGop'. Press "Execute" and the following window should appear:



The present configuration should be optimal for most use cases. Choose as the output a new resource and click “Execute”. A console window will appear and stay for a few seconds. This is normal due to the nature of the used original Extraction module.



After receiving the “success”-message, close the window and you should have a text resource in the AXMEDIS editor containing the XML descriptor data. It is advised to edit the objects properties and add the original resource filename



16.13 Video Adaptation Plugin (FHGIGD)

NOT INCLUDED IN THIS VERSION

16.13.1 Main functionalities

NOT INCLUDED IN THIS VERSION

16.13.2 Relationship with other tools

NOT INCLUDED IN THIS VERSION

16.13.3 Detailed description of the functionalities and Screenshots

NOT INCLUDED IN THIS VERSION

16.14 Workflow Editor Plugin (IRC)

16.14.1 Main functionalities

The Workflow Editor Plug-in is a library used by AXMEDIS Editor to enable communication with AXMEDIS workflow engine. It exposes following functions:

Functionality	Details
Notification of Completion	This method is invoked by the editor to send back the notification towards workflow engine for the completion of previously issues asynchronous request.
Get Workflow Information	This method is invoked by the application to request workflow information from workflow engine.
Editing of Object	This method is invoked through a webservice call coming from Workflow Request Gateway. This method will load the specified object (AXOID) and will allow the user to edit the object in the AXMEDIS Editor.
View Object Attributes	This method is invoked through a webservice call coming from Workflow Request Gateway. This method will allow the workflow engine to retrieve the object attributes for the object specified by the AXOID.
Add History Information	This method is invoked through a webservice call coming from Workflow Request Gateway. This method will allow the workflow engine to add the object history for the object specified by the AXOID.

16.14.2 Relationship with other tools

The Workflow Editor Plug-in is loaded by the AXMEDIS Editor. This plug-in communicates directly with AXMEDIS Workflow Request Gateway and AXMEDIS Workflow Response Gateway. For Microsoft Biztalk server, this plug-in communicates directly with the workflow engine.

16.14.3 Detailed description of the functionalities and Screenshots

The functions of this plug-in are invoked by AXMEDIS Editor. Also the Functions can be invoked by the workflow engine automatically. The following Screenshots shows the empty AXMEDIS Editor.

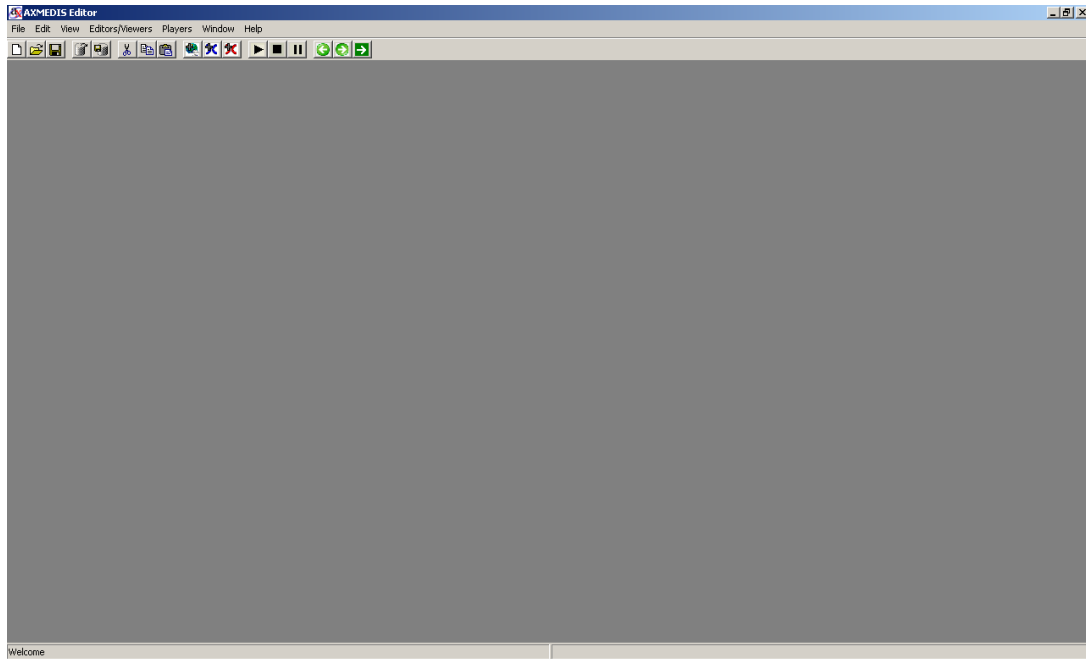


Figure 1: Empty AXMEDIS Editor

When the Workflow Editor Plug-in receives request from Workflow Engine for the editing of Object, it will load the specified object from the AXMEDIS Database or if the AXOID is not specified then it will create a new object as shown in the following figure:

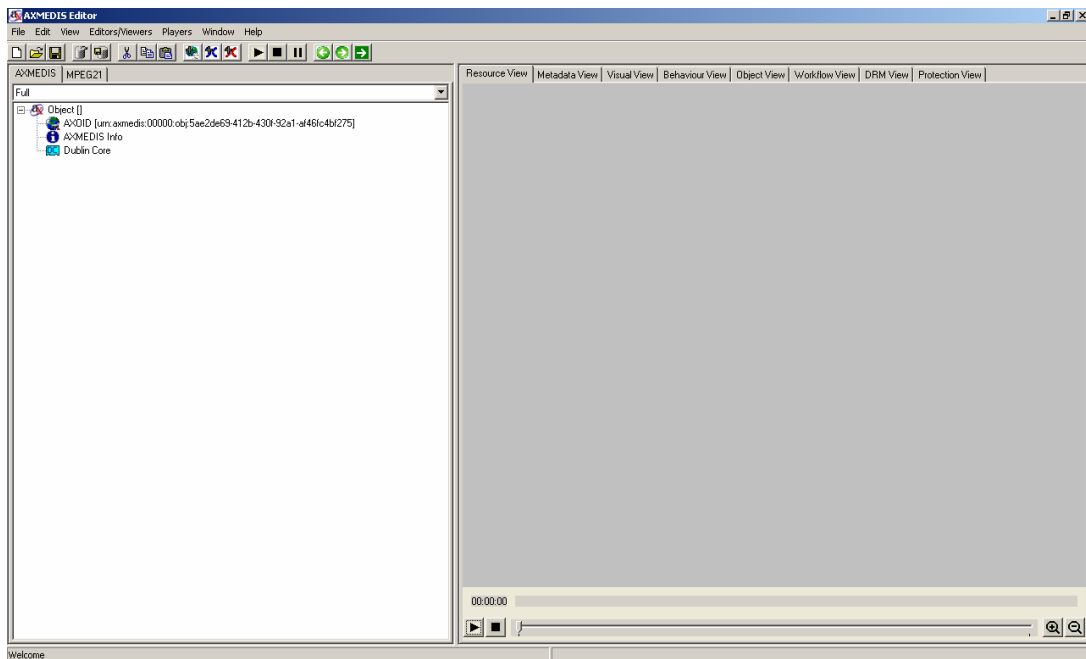


Figure 2: Creation of new Object following the Workflow Request

The user can also select the “Workflow View” tab from the Editor. This will display the workflow related information for the object currently being edited/viewed as shown in following figure.

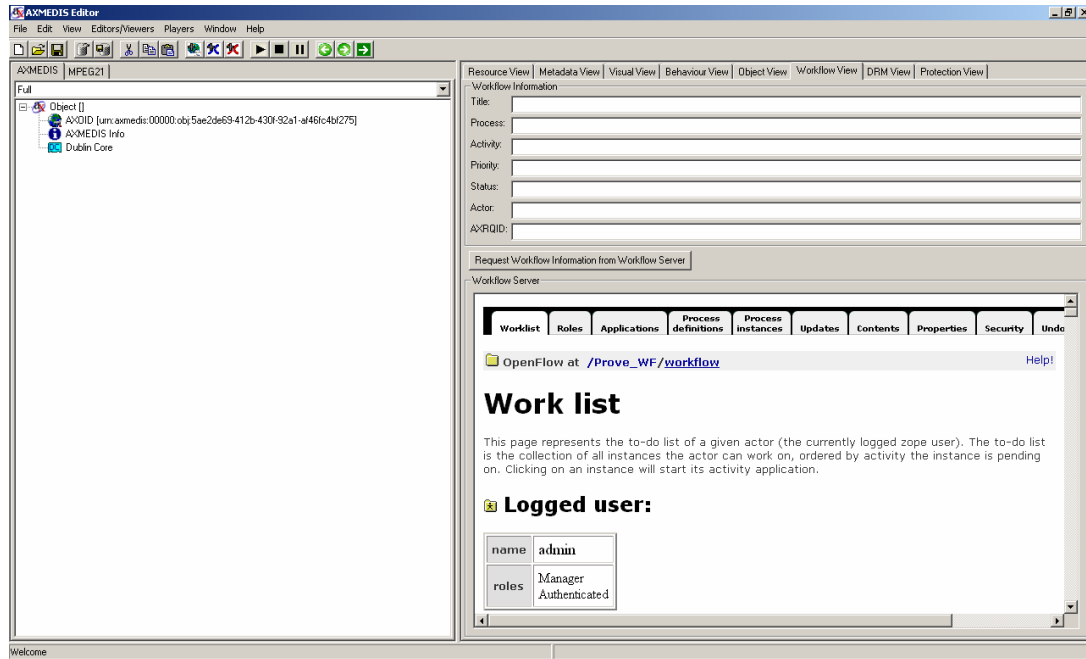


Figure 3: The Workflow View for the Object

When the editing of Object is completed. The user can notify workflow by selecting “Notify Workflow Activity Completion” command from the File Menu as shown in the following figure. This will send a notification signal to workflow engine.

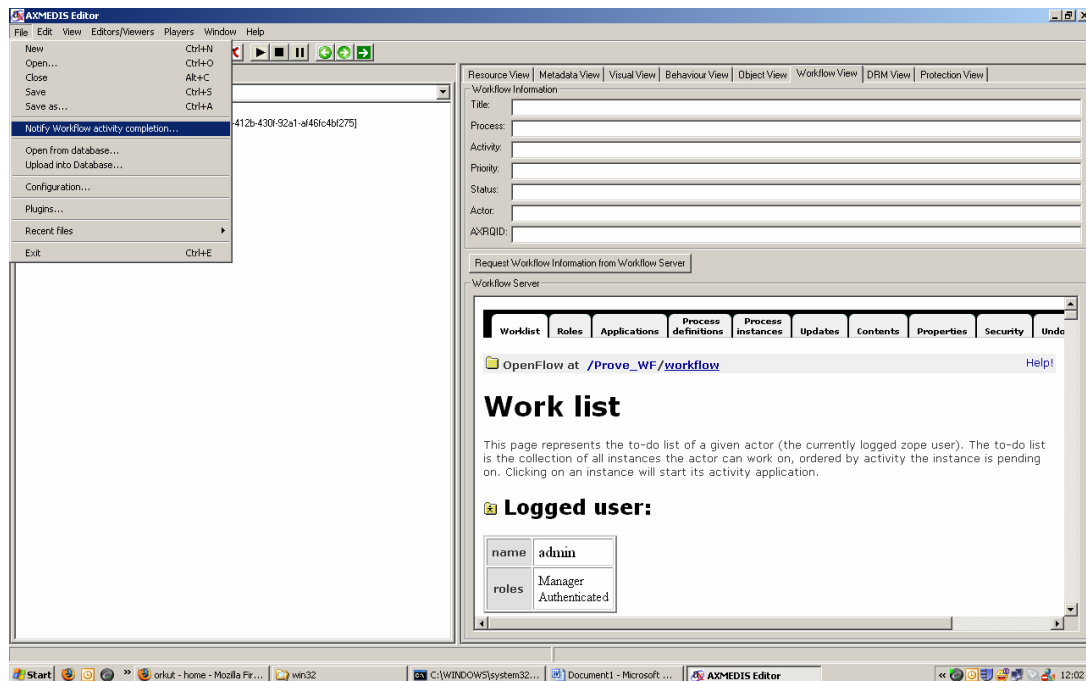


Figure 4: Sending Notification of Completion to Workflow Engine

16.15 Workflow Rule Editor Plugin (IRC)

16.15.1 Main functionalities

The Workflow Rule Editor Plug-in is a library used by AXCP Rule Editor and AXMEDIS PnP Editor to enable communication with AXMEDIS workflow engine. It exposes following functions:

Functionality	Details
Notification of Completion	This method is invoked by the editor to send back the notification towards workflow engine for the completion of previously issues asynchronous request.
Editing of AXCP Rule	This method is invoked through a webservice call coming from Workflow Request Gateway. This method will load the AXCP Rule editor with the specified Rule Header and will allow the user to edit the AXCP Rule in the AXCP Rule Editor.
Edit PnP Programme	This method is invoked through a webservice call coming from Workflow Request Gateway. This method will load the AXMEDIS PnP editor with the specified Programme Header and will allow the user to edit the AXMEDIS PnP Programme in the PnP Editor.

16.15.2 Relationship with other tools

The Workflow Rule Editor Plug-in is loaded by the AXCP Rule Editor and PnP Editor. This plug-in communicates directly with AXMEDIS Workflow Request Gateway and AXMEDIS Workflow Response Gateway. For Microsoft Biztalk server, this plug-in communicates directly with the workflow engine.

16.15.3 Detailed description of the functionalities and Screenshots

The functions of this plug-in are invoked by AXCP Rule Editor. Also the Functions can be invoked by the workflow engine automatically. The following Screenshots shows the empty AXCP Rule Editor.

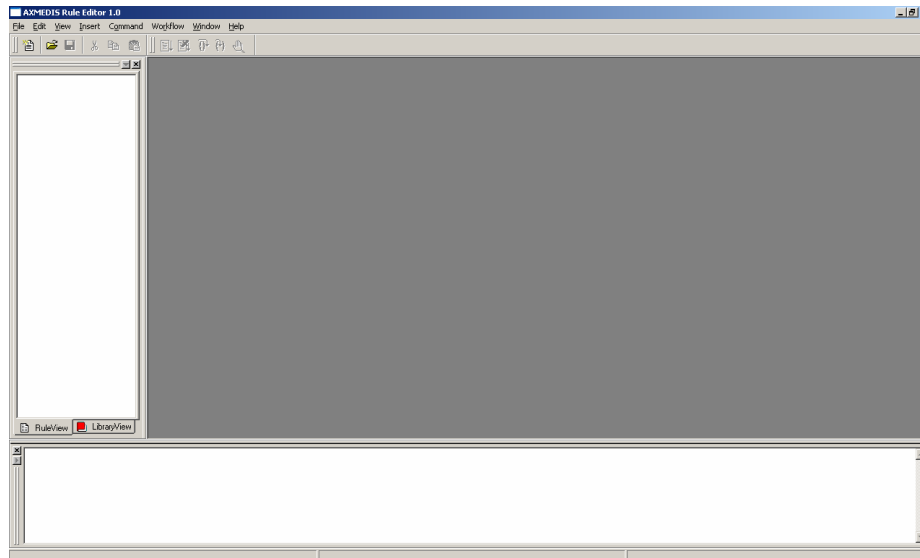


Figure 1: Empty AXCP Rule Editor

When the Workflow Rule Editor Plug-in receives request from Workflow Engine for the editing of a Rule, it will load the specified XML Rule from the Workflow Request as shown in the following figure:

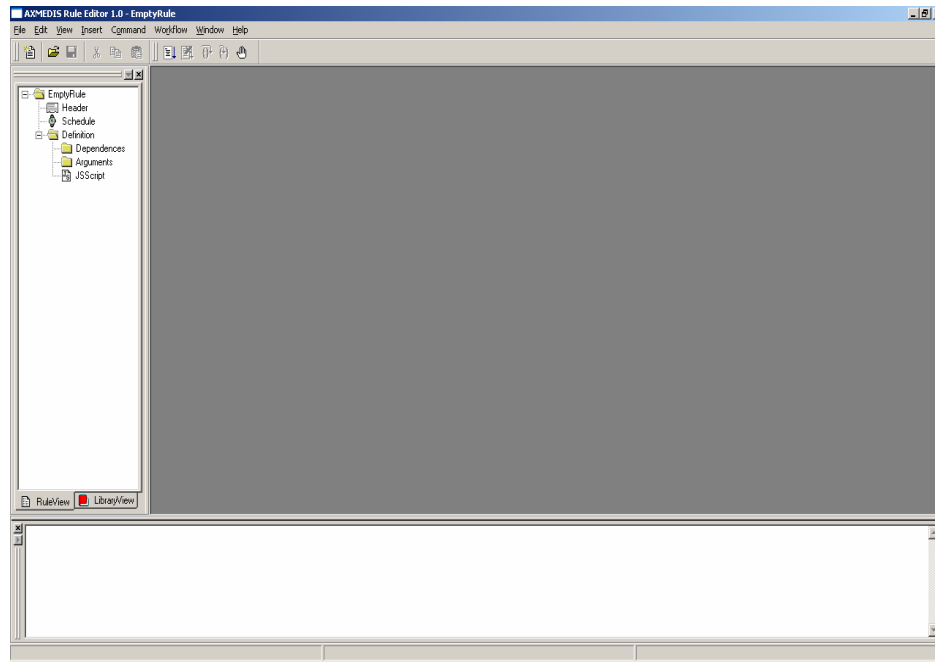


Figure 2: Rule Loaded from Workflow Engine

When the editing of the rule is completed. The user can notify workflow by selecting “Notify Workflow Activity Completion” command from the Workflow Menu as shown in the following figure. This will send a notification signal to workflow engine.

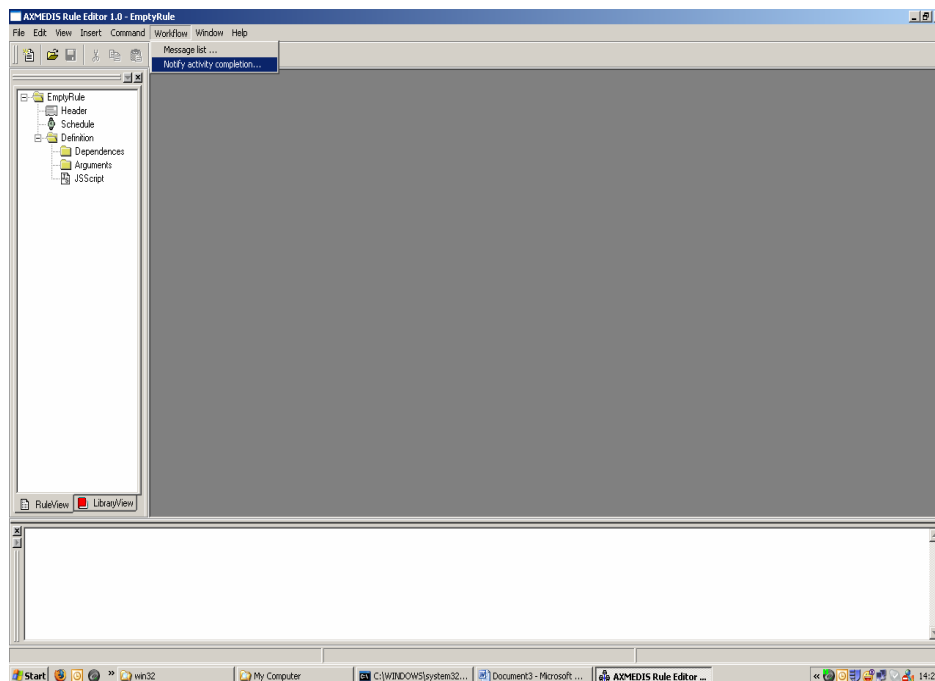


Figure 3: Sending of Notification of Completion

The User can confirm the notification command by selecting 'OK' from the message box.

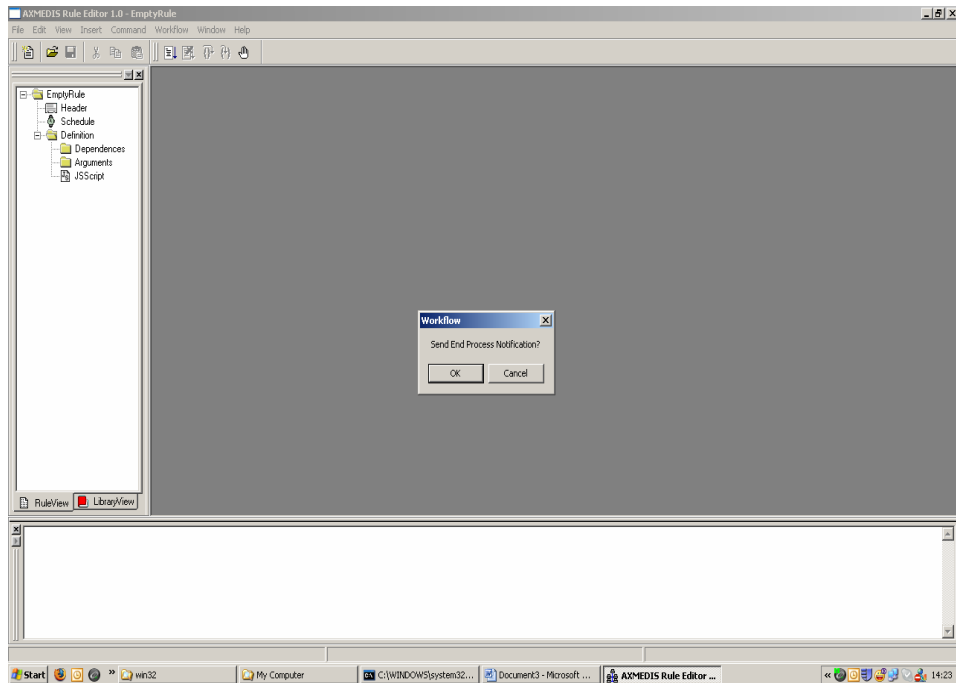


Figure 4: Confirmation for the Notification

The functions of this plug-in can also be invoked by PnP Editor. Also the Functions can be invoked by the workflow engine automatically. The following Screenshots shows the empty PnP Editor.

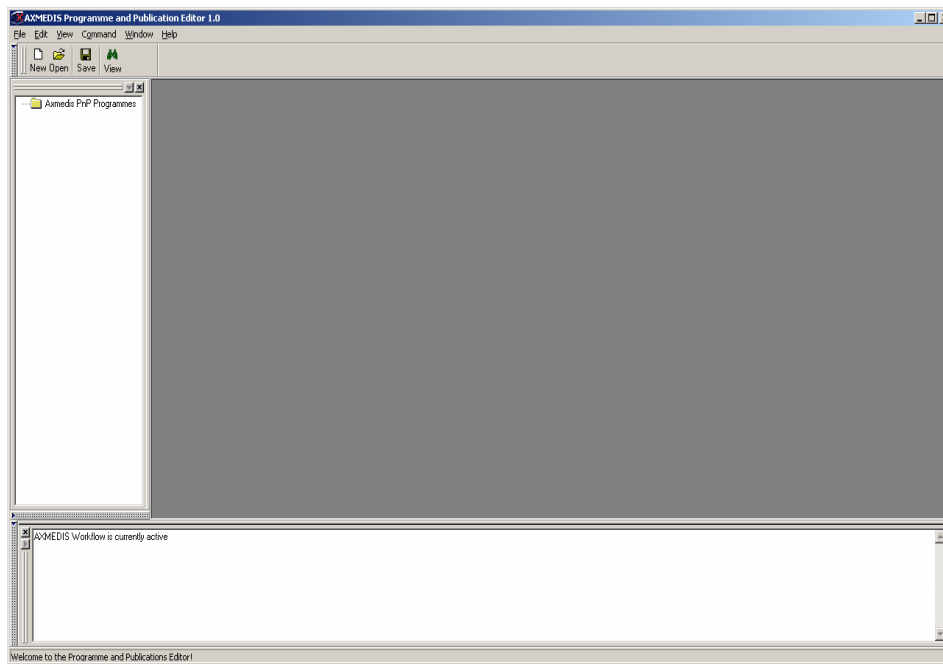


Figure 5: Empty PnP Editor

When the Workflow Rule Editor Plug-in receives request from Workflow Engine for the editing of a PnP Programme, it will load the specified XML from the Workflow Request as shown in the following figure:

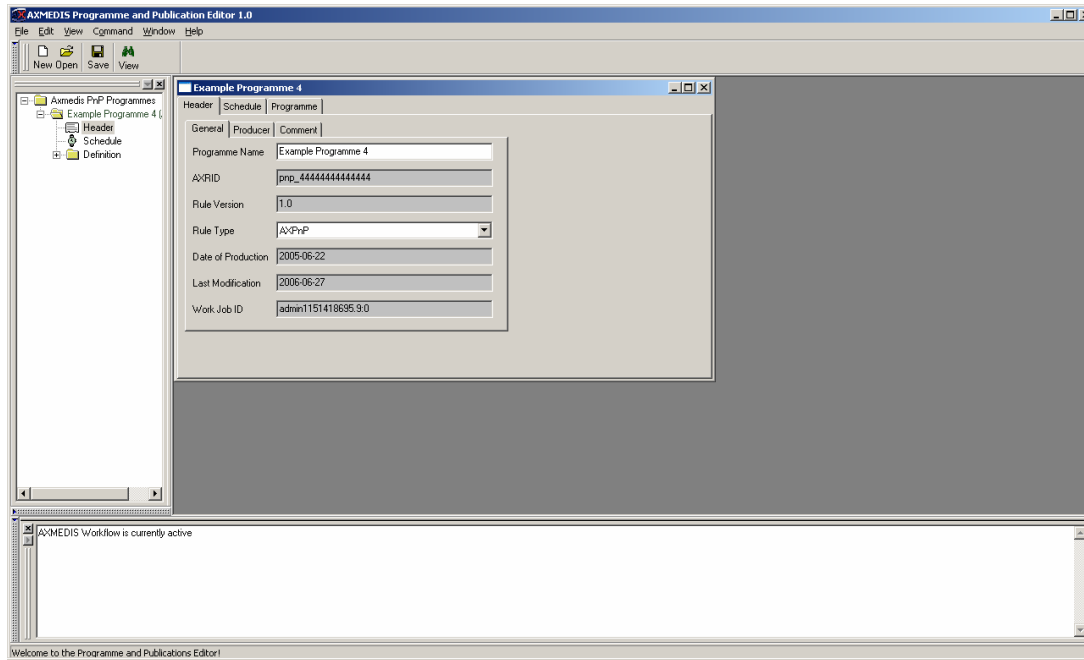


Figure 6: Programme Loaded from Workflow Engine

When the editing of the Programme is completed. The user can notify workflow by Activating the Programme in the PnP Engine as shown in the following figure. This will send a notification signal to workflow engine.

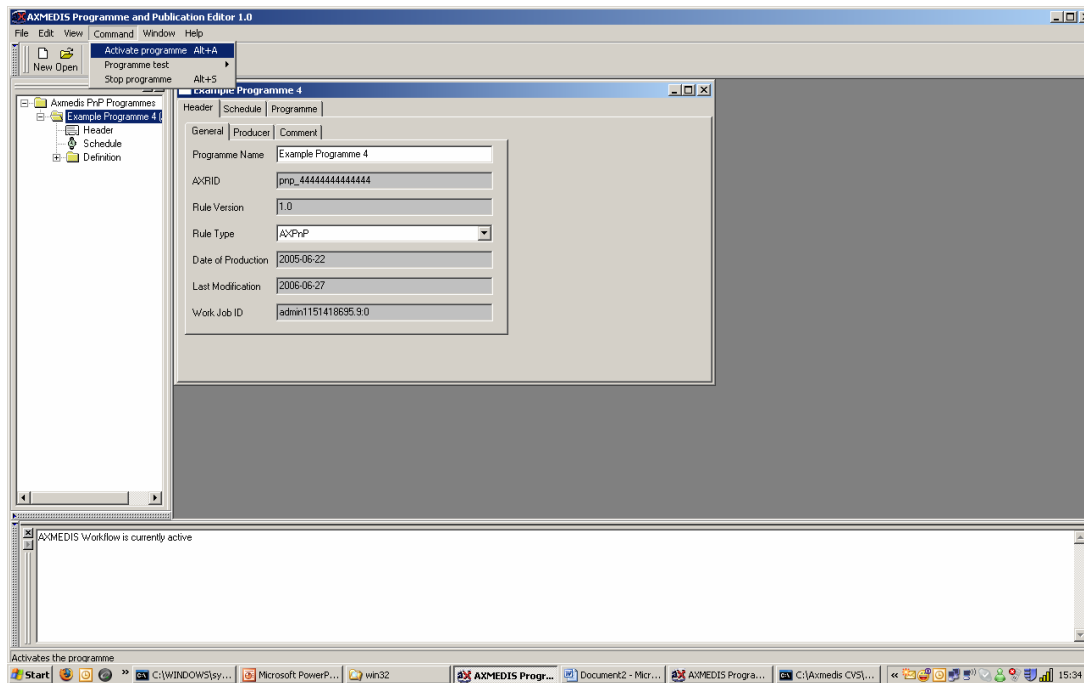


Figure 7: Sending of Notification of Completion

16.16 Workflow Engine Plugin (IRC)

The Workflow Engine Plug-in is loaded by the AXCP Scheduler and by PnP Engine. Generally the functions offered by this plug-in are invoked automatically by these tools without the intervention of the User.

16.16.1 Main functionalities

Functionality	Details
Notification of Completion	This method is invoked by the editor to send back the notification towards workflow engine for the completion of previously issues asynchronous request.
Workflow Process Request	This method is invoked by the PnP engine to send request for activation of a workflow process identified the supplied processID.
Install and Activate	This method allows the workflow engine to install a new rule in the AXCP engine. The ruleID of the newly installed rule will be returned as ruleID.
Run Rule	This method allows the workflow engine to run a rule in the AXCP engine as per the supplied parameters.
Deactivate Rule	This method allows the workflow engine to deactivate a rule in the AXCP engine as per the supplied parameters.
Suspend Rule	This method allows the workflow engine to suspend a rule in the AXCP engine as per the supplied parameters.
Pause Rule	This method allows the workflow engine to pause a rule in the AXCP engine as per the supplied parameters.
Kill Rule	This method allows the workflow engine to kill a rule in the AXCP engine as per the supplied parameters.
Remove Rule	This method allows the workflow engine to remove a rule from the AXCP engine as per the supplied parameters.
Resume Rule	This method allows the workflow engine to resume a rule in the AXCP engine as per the supplied parameters.
Get Rule Status	This method allows the workflow engine to know the status of the a rule in the AXCP engine as per the supplied parameters.
Get Rule Log	This method allows the workflow engine to know the run log for a rule in the AXCP engine as per the supplied parameters.
Get List of Rules	This method allows the workflow engine to retrieve the list of currently installed rules in the AXCP engine as per the supplied parameters.
Get Rule	This method allows the workflow engine to retrieve the rule schema from the AXCP engine as per the supplied parameters.
Status Request to PnP Engine	This method allows the workflow engine to retrieve the status of the PnP engine.
Suspend PnP Programme	This method allows the workflow engine to suspend a program in the PnP engine.
Abort PnP Programme	This method allows the workflow engine to abort a program in the PnP engine.
Resume PnP Programme	This method allows the workflow engine to resume a program in the PnP engine.
Activate PnP Programme	This method allows the workflow engine to activate a program in the PnP engine.
Workflow Notification	This method allows the workflow engine to send the notification to the PnP engine for the previously issues request to activate a process.

16.16.2 Relationship with other tools

The Workflow Engine Plug-in is loaded by the AXCP Scheduler and PnP Engine. This plug-in communicates directly with AXMEDIS Workflow Request Gateway and AXMEDIS Workflow Response Gateway. For Microsoft Biztalk server, this plug-in communicates directly with the workflow engine.

16.16.3 Detailed description of the functionalities and Screenshots

The functionality offered by this plug-in is hidden from the user. Hence there are no screenshots for it.

16.17 Ringtone Adaptation Plugin (IRC)

16.17.1 Main functionalities

Ringtone Adaptation refers to the adaptation of ringtones of popular formats to enhance usability and manage the variable delivery to cater for heterogeneous client devices and user requirements on-demand. It can be used to transcode the ringtones depending on the client devices. For eg some mobile phones may support only low bit rate ringtones while others will be having restrictions on the size of the ringtone.

16.17.2 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.

16.17.3 Detailed description of the functionalities and Screenshots

A step by step Example showing how to use the functions is given below.

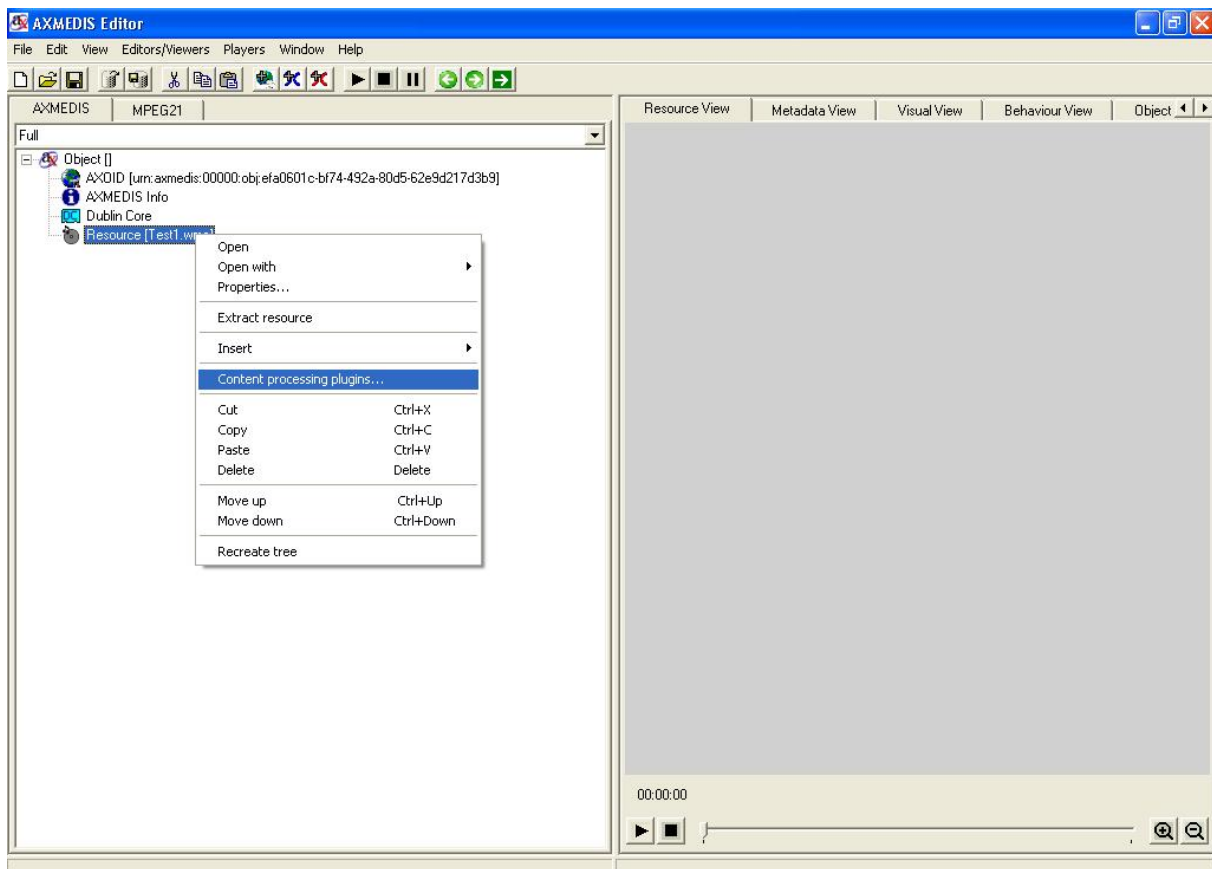
16.17.3.1 Convert Function

Description: Used to Convert a ringtone to different formats. The formats supported currently are x-mpeg (.mp3), x-aiff (.aif, .aiff), x-wav (.wav), basic (.au, .snd), x-ms-wma (.wma), x-vorbis (.ogg), x-pn-realaudio (.ra, .ram)

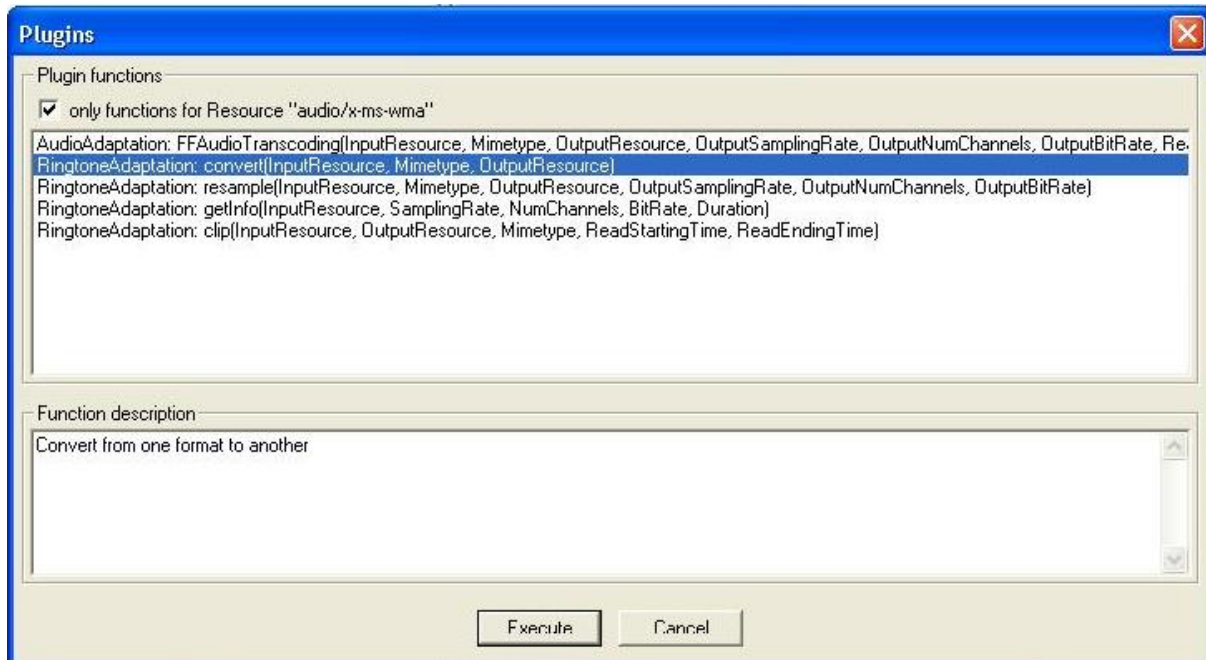
How to use:

Load an embedded resource (audio/ringtone file) into the AXMEDIS Editor

Right click on the resource and select Content processing plugins

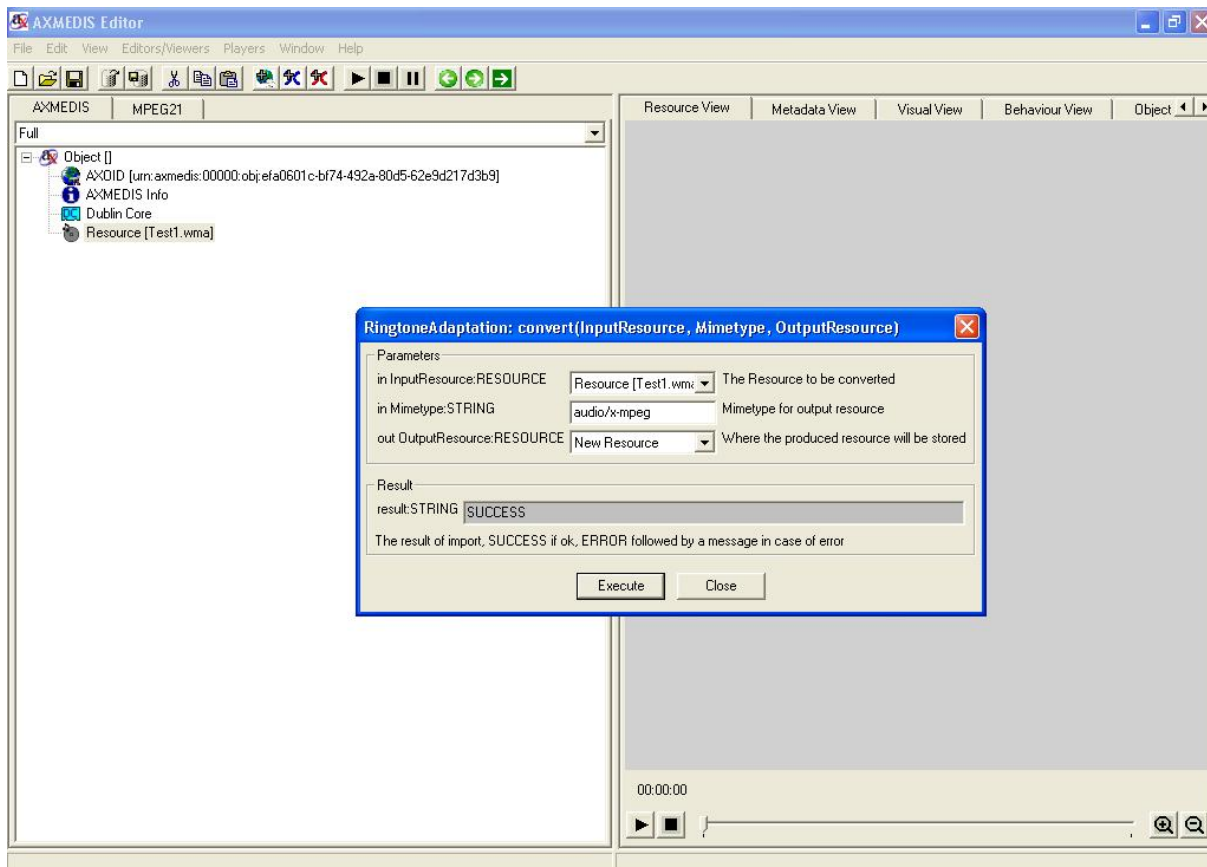


It will pop up a new window showing the different content process plugins available for the particular resource, in our case it is ringtone.



Select the Convert function to convert the ringtone to any popular format. The formats supported are x-mpeg (.mp3), x-aiff (.aif, .aiff), x-wav (.wav), basic (.au, .snd), x-ms-wma (.wma), x-vorbis (.ogg), x-pn-realaudio (.ra, .ram)

It will take you to the next screen where you can specify the various parameters for converting the ringtone. Once you enter the parameters and click execute, it will convert the ringtone to the appropriate format. If the ringtone conversion is successful then in result's space you can see SUCCESS or else it will return Error along with an error message.



The description of each parameter is given below.

InputResource

Description: The Resource to be converted

Parameter Type *AxResource*

Default Value:

Constraints:

Resource Type: audio

Resource Format: x-mpeg (.mp3), x-aiff (.aif, .aiff), x-wav (.wav), basic (.au, .snd),
x- ms-wma (.wma), x-vorbis (.ogg), x-pn-realaudio (.ra, .ram)

Ranges:

Mimetype

Description: Mimetype for output resource

Parameter Type *string*

Default Value:

Constraints:

Resource Type: audio

Resource Format: x-mpeg, x-aiff, x-wav, basic, x-vorbis, x-ac3

OutputResource

Description: Where the produced resource will be stored

Parameter Type *AxResource*

Default Value:

Constraints:

Result

Result type: *string*

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

16.17.3.2 Resample Function

Description: Resamples the input file (i.e. changing frequency, bitrate, sampling rate etc)

RingtoneAdaptation: resample(InputResource, Mimetype, OutputResource, OutputSamplingRate, OutputNumChann... ✕

Parameters

in InputResource:RESOURCE	<input type="text" value="Resource [Test1.wma]"/>	The Resource to be converted
in Mimetype:STRING	<input type="text" value="audio/x-mpeg"/>	Mimetype for output resource
out OutputResource:RESOURCE	<input type="text" value="New Resource"/>	Where the produced resource will be stored
in OutputSamplingRate:UINT32	<input type="text" value="44100"/>	Sampling rate of the output audio file (default: sampling rate of the input)
in OutputNumChannels:UINT16	<input type="text" value="2"/>	Number of channels of the output audio file (default: number of channels of the input)
in OutputBitRate:UINT16	<input type="text" value="192"/>	Bit rate of the output audio file - Only applies to compressed audio file formats (default: 64 kb)

Result

result:STRING

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

InputResource

Description: The Resource to be converted

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: audio

Resource Format: x-mpeg (.mp3), x-aiff (.aif, .aiff), x-wav (.wav), basic (.au, .snd),
x-ms-wma (.wma), x-vorbis (.ogg), x-pn-realaudio (.ra, .ram)

Ranges:

Mimetype

Description: Mimetype for output resource

Parameter Type STRING

Default Value:

Constraints:

Resource Type: audio

Resource Format: x-mpeg, x-aiff, x-wav, basic, x-vorbis, x-ac3

OutputResource

Description: Where the produced resource will be stored

Parameter Type RESOURCE

Default Value:

Constraints:

OutputSamplingRate

Description: Sampling rate of the output audio file (default: sampling rate of the input)

Parameter Type UINT32

Default Value:

Constraints:

Resource Type:

Ranges:

OutputNumChannels

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

Parameter Type UINT16

Default Value:

Constraints:

Resource Type:

Ranges:

OutputBitRate

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

Parameter Type UINT16

Default Value:

Constraints:

Resource Type:

Ranges:

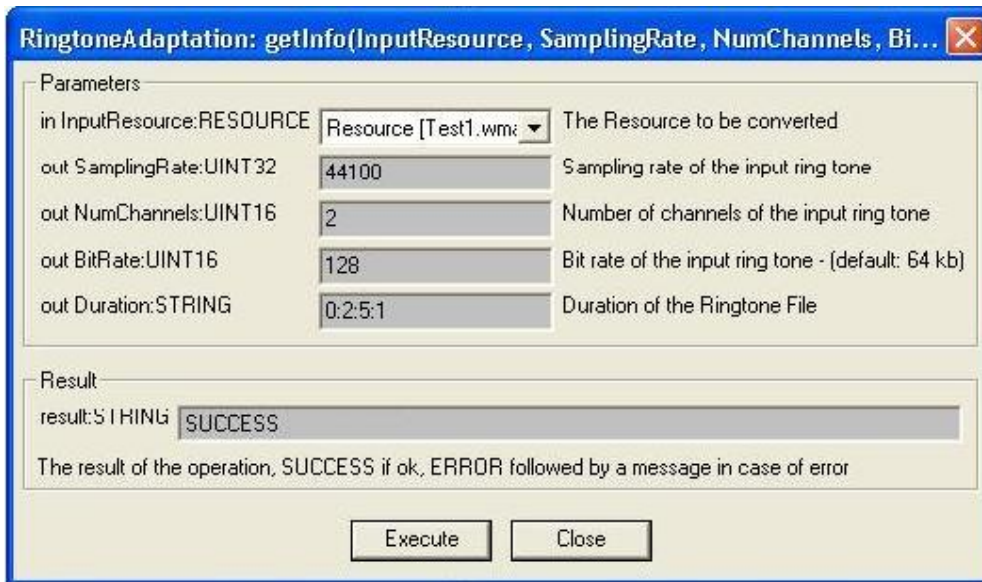
Result

Result type: STRING

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

16.17.3.3 **getInfo Function**

Description: Get all the information about the input Ring Tone



InputResource

Description: The Resource to be converted

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: audio

Resource Format: x-mpeg (.mp3), x.aiff (.aif, .aiff), x-wav (.wav), basic (.au, .snd),
x-ms-wma (.wma), x-vorbis (.ogg), x-pn-realaudio (.ra, .ram)

Ranges:

SamplingRate

Description: Sampling rate of the input ring tone

Parameter Type UINT32

Default Value:

Constraints:

Resource Type:

Ranges:

NumChannels

Description: Number of channels of the input ring tone

Parameter Type UINT16

Default Value:

Constraints:

Resource Type:

Ranges:

BitRate

Description: Bit rate of the input ring tone - (default: 64 kb)

Parameter Type UINT16

Default Value:

Constraints:

Resource Type:

Ranges:

Duration

Description: Duration of the Ringtone File (In the format Hours: Mins: Secs: milliseconds)

Parameter Type STRING

Default Value:

Constraints:

Resource Type:

Ranges:

Result

Result type: STRING

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

16.17.3.4 Clip Function

Description: Clips the file for the specified time (for e.g. reducing it to a 30 sec clip)

RingtoneAdaptation: clip(InputResource, OutputResource, Mimetype, ReadStartingT...

Parameters

in InputResource:RESOURCE	Resource [Test1.wink]	The Resource to be converted
out OutputResource:RESOURCE	New Resource	Where the produced resource will be stored
in Mimetype:STRING	audio/x-mpeg	Mimetype for output resource
in ReadStartingTime:FLOAT	0.0	Starting time for the clip(default: beginning of the file)
in ReadEndingTime:FLOAT	30.0	Ending time for the clip (default: end of the file)

Result

result:STRING **SUCCESS**

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

Execute Close

InputResource

Description: The Resource to be converted

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: audio

Resource Format: x-mpeg (.mp3), x.aiff (.aif, .aiff), x-wav (.wav), basic (.au, .snd),
x-ms-wma (.wma), x-vorbis (.ogg), x-pn-realaudio (.ra, .ram)

Ranges:

OutputResource

Description: Where the produced resource will be stored

Parameter Type RESOURCE

Default Value:

Constraints:

Mimetype

Description: Mimetype for output resource

Parameter Type STRING

Default Value:

Constraints:

Resource Type: audio

Resource Format: x-mpeg, x-aiff, x-wav, basic, x-vorbis, x-ac3

ReadStartingTime

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

Parameter Type FLOAT

Default Value:

Constraints:

Resource Type:

Ranges:

ReadEndingTime

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

Parameter Type FLOAT

Default Value:

Constraints:

Resource Type:

Ranges:

Result

Result type: STRING

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

16.18 Image Processing Plugin (DSI)

16.18.1 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
- **Emboss**, 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
- **Roll**, 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.**

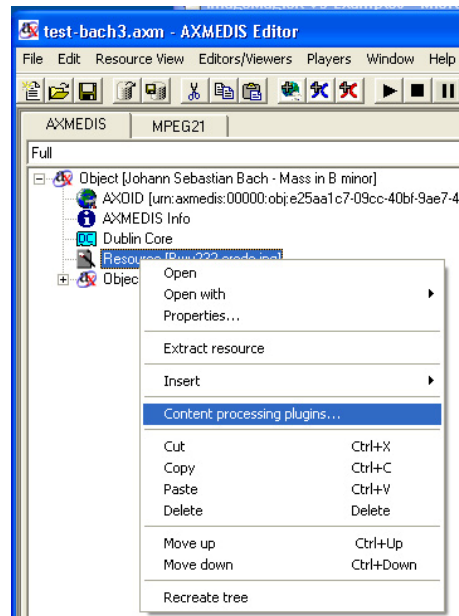
16.18.2 Relationship with other tools

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.

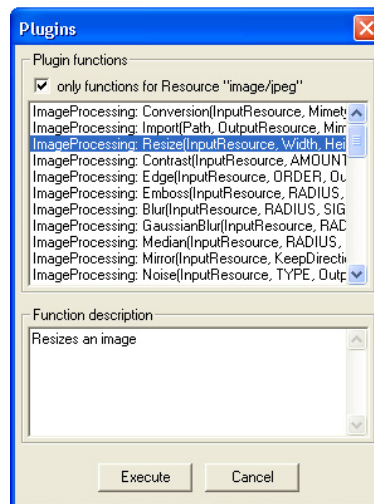
16.18.3 Detailed description of the functionalities and Screenshots

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



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 next sections 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>

16.18.3.1 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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: Mimetype

Description: Mimetype for output resource

Parameter Type STRING

Default Value:

Constraints:

Name: OutputResource

Description: Where the produced resource will be stored

Parameter 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 [C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml - go_to_top#go_to_top](#)

16.18.3.2 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

Parameter Type STRING

Default Value:

Constraints:

Name: OutputResource

Description: Where the imported resource will be stored

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: MimeType

Description: Test

Parameter 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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go to top](#)#[go to top](#)

16.18.3.3 *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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: Width

Description: The new image width

Parameter Type INT32

Default Value:

Constraints:

Name: Height

Description: The new image height

Parameter Type INT32

Default Value:

Constraints:

Name: KeepAspectRatio

Description: Indicates to preserve image aspect ratio or not

Parameter Type BOOLEAN

Default Value:

Constraints:

Name: OutputResource

Description: Where the resized resource will be stored

Parameter 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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go to top](#)#[go to top](#)

16.18.3.4 **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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: AMOUNT

Description: The contrast amount

Parameter Type INT32

Default Value:

Constraints:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go to top](#)#[go to top](#)

16.18.3.5 **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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png
Ranges:
Name: ORDER
Description: The Order Edge
Parameter Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Parameter 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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go to top#go to top](#)

16.18.3.6 *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
Parameter Type RESOURCE
Default Value:
Constraints:
Resource Type: image
Resource Format: jpeg gif png
Ranges:
Name: RADIUS
Description: The Radius Emboss
Parameter Type INT32
Default Value:
Constraints:
Name: SIGMA
Description: The sigma Emboss
Parameter Type INT32
Default Value:
Constraints:
Name: OutputResource
Description: Where the manipulated resource will be stored
Parameter 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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go to top#go to top](#)

16.18.3.7 *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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go to top#go to top](#)

16.18.3.8 *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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - go to top#go to top

16.18.3.9 *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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - go to top#go to top

16.18.3.10 *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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: KeepDirection

Description: The KeepDirection Mirror

Parameter Type BOOLEAN

Default Value:

Constraints:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 [C:\Programmi\AXMEDIS Tools\AXMEDIS-](C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml)

<plugins\imageprocessingplugin.xml> - [go to top](#)#[go to top](#)

16.18.3.11 *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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: TYPE

Description: The Type Noise

Parameter Type INT32

Default Value:

Constraints:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go_to_top#go_to_top](#)

16.18.3.12 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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go_to_top#go_to_top](#)

16.18.3.13 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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go_to_top#go_to_top](#)

16.18.3.14 *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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 [C:\Programmi\AXMEDIS Tools\AXMEDIS-](#)

[plugins\imageprocessingplugin.xml - go to top#go to top](#)

16.18.3.15 *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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: CHANNEL

Description: The Channel ExtractChannel

Parameter Type INT32

Default Value:

Constraints:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go to top#go to top](#)

16.18.3.16 *Grayscale*

STRING Grayscale (RESOURCE InputResource, RESOURCE OutputResource)

Version: 1.0

Description: Grayscale the image

Parameter List

Name: InputResource

Description: The Resource to be manipulated

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go to top#go to top](#)

16.18.3.17 *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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go to top#go to top](#)

16.18.3.18 *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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go to top](#)#[go to top](#)

16.18.3.19 *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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: BRIGHTNESS

Description: Brightness modulate

Parameter Type INT32

Default Value:

Constraints:

Name: SATURATION

Description: Saturation modulate

Parameter Type INT32

Default Value:

Constraints:

Name: HUE

Description: Hue modulate

Parameter Type INT32

Default Value:

Constraints:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go to top#go to top](#)

16.18.3.20 *Monochrome*

STRING Monochrome (RESOURCE InputResource, RESOURCE OutputResource)

Version: 1.0

Description: Monochrome the image

Parameter List

Name: InputResource

Description: The Resource to be manipulated

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go to top#go to top](#)

16.18.3.21 *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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: GRAYSCALE

Description: Where the manipulated resource will be stored

Parameter Type: BOOLEAN

Default Value:

Constraints:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go to top](#)#[go to top](#)

16.18.3.22 *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

Parameter Type: RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go to top](#)#[go to top](#)

16.18.3.23 *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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: RADIUS

Description: the radius OilPaint

Parameter Type INT32

Default Value:

Constraints:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go to top#go to top](#)

16.18.3.24 *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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: LEVEL

Description: the quality of the compress level

Parameter Type INT32

Default Value:

Constraints:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go to top#go to top](#)

16.18.3.25 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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: NCOLORS

Description: the number of color

Parameter Type INT32

Default Value:

Constraints:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml - go_to_top#go_to_top

16.18.3.26 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

Parameter 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

Parameter 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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go to top#go to top](#)

16.18.3.27 *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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go_to_top#go_to_top](#)

16.18.3.28 *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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: R1

Description: r1

Parameter Type INT32

Default Value:

Constraints:

Name: G1

Description: g1

Parameter Type INT32

Default Value:

Constraints:

Name: B1

Description: b1

Parameter Type INT32

Default Value:

Constraints:

Name: R2

Description: r2

Parameter Type INT32

Default Value:

Constraints:

Name: G2

Description: g2

Parameter Type INT32

Default Value:

Constraints:

Name: B2

Description: b2

Parameter Type INT32

Default Value:

Constraints:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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

[C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml - go to top#go to top](#)

16.18.3.29 *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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: X

Description: x

Parameter Type INT32

Default Value:

Constraints:

Name: Y

Description: y

Parameter Type INT32

Default Value:

Constraints:

Name: B

Description: b

Parameter Type INT32

Default Value:

Constraints:

Name: R

Description: r

Parameter Type INT32

Default Value:

Constraints:

Name: G

Description: g

Parameter Type INT32

Default Value:

Constraints:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml - go_to_top#go_to_top

16.18.3.30 *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 columns and rows)

Parameter List

Name: InputResource

Description: The Resource to be manipulated

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: X

Description: x

Parameter Type INT32

Default Value:

Constraints:

Name: Y

Description: y

Parameter Type INT32

Default Value:

Constraints:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml - go_to_top#go_to_top

16.18.3.31 *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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: ANGLE

Description: Number of the degrees

Parameter Type INT32

Default Value:

Constraints:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go to top](#)#[go to top](#)

16.18.3.32 *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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: WIDTH

Description: Width

Parameter Type INT32

Default Value:

Constraints:

Name: HEIGHT

Description: Height

Parameter Type INT32

Default Value:

Constraints:

Name: MODE

Description: Mode

Parameter Type INT32

Default Value:

Constraints:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go to top#go to top](#)

16.18.3.33 *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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: XSHEAR

Description: XSHEAR

Parameter Type INT32

Default Value:

Constraints:

Name: Yshear

Description: Yshear

Parameter Type INT32

Default Value:

Constraints:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go to top#go to top](#)

16.18.3.34 *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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: AZIMUTH

Description: AZIMUTH

Parameter Type INT32

Default Value:

Constraints:

Name: ELEVATION

Description: ELEVATION

Parameter Type INT32

Default Value:

Constraints:

Name: COLOR

Description: COLOR

Parameter Type BOOLEAN

Default Value:

Constraints:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml - go_to_top#go_to_top

16.18.3.35 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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: AMOUNT

Description: AMOUNT

Parameter Type INT32

Default Value:

Constraints:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go to top](#)#[go to top](#)

16.18.3.36 *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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: LEVEL

Description: LEVEL

Parameter Type INT32

Default Value:

Constraints:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 <C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml> - [go to top](#)#[go to top](#)

16.18.3.37 *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

Parameter 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

Parameter Type INT32

Default Value:

Constraints:

Name: Y

Description: y coordinate of the top-level corner of the rectangle

Parameter Type INT32

Default Value:

Constraints:

Name: WIDTH

Description: Width member

Parameter Type INT32

Default Value:

Constraints:

Name: HEIGHT

Description: Height member

Parameter Type INT32

Default Value:

Constraints:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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 C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml - go_to_top#go_to_top

16.18.3.38 *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 analysis

Parameter 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 [C:\Programmi\AXMEDIS Tools\AXMEDIS-](C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml)

[plugins\imageprocessingplugin.xml](C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml) - go to top#go to top

16.18.3.39 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 [C:\Programmi\AXMEDIS Tools\AXMEDIS-](C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml)

[plugins\imageprocessingplugin.xml](C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml) - go to top#go to top

16.18.3.40 *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

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: InputResource2

Description: The Resource paste

Parameter Type RESOURCE

Default Value:

Constraints:

Resource Type: image

Resource Format: jpeg gif png

Ranges:

Name: X

Description: X

Parameter Type INT32

Default Value:

Constraints:

Name: Y

Description: Y

Parameter Type INT32

Default Value:

Constraints:

Name: COMPOSE

Description: Compose

Parameter Type INT32

Default Value:

Constraints:

Name: OutputResource

Description: Where the manipulated resource will be stored

Parameter 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

[C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml - go to top#go to top](#)

16.18.3.41 *Test*

RESOURCE Test (RESOURCE InputResource, AXOM Axom)

Version: 1.0

Description: Test an image

Parameter List

Name: InputResource

Description: The Resource to be tested

Parameter Type RESOURCE

Default Value:

Constraints:

Name: Axom

Description: The object

Parameter 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

[C:\Programmi\AXMEDIS Tools\AXMEDIS-plugins\imageprocessingplugin.xml - go to top#go to top](#)