Automating Production of Cross Media Content for Multi-channel Distribution

www.AXMEDIS.org

DE12.2.4.2
AXMEDIS-ELTEO Content Modeling and Production

Version: 1.0
Date: 08-02-2008
Responsible: VRS

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: DE12.2.4.2 Contractual Date of Delivery: M42 Actual Date of Delivery: 08/02/2008 Title of Deliverable: Document Work-Package contributing to the Deliverable: WP12.2. Task contributing to the Deliverable: WP12.2.4 Nature of the Deliverable: report Author(s): VRS,TEO

Abstract:
This report is a deliverable of workpackage WP 12.2.4.2 AXMEDIS-ELTEO Content Modelling and Production. The document describes the general architecture of VRS AXMEDIS content processing tools, their integration with real-life third-party video processing tools, and use cases and scenarios of a content producer producing video content for IPTV streaming and downloading on PC. It also specifies the VRS content processed through AXMEDIS system.
Keyword List:
Content processing, content adaptation, AXCP, AxObject creation, streaming, video, MPEG-4, MPEG2-TS, P2P

AXMEDIS Copyright Notice

The following terms (including future possible amendments) set out the rights and obligations licensee will be requested to accept on entering into possession of any official AXMEDIS document either by downloading it from the web site or by any other means.

Any relevant AXMEDIS document includes this license. PLEASE READ THE FOLLOWING TERMS CAREFULLY AS THEY HAVE TO BE ACCEPTED PRIOR TO READING/USE OF THE DOCUMENT.

1. **DEFINITIONS**
   i. "**Acceptance Date**" is the date on which these terms and conditions for entering into possession of the document have been accepted.
   ii. "**Copyright**" stands for any content, document or portion of it that is covered by the copyright disclaimer in a Document.
   iii. "**Licensor**" is AXMEDIS Consortium as a de-facto consortium of the EC project and any of its derivations in terms of companies and/or associations, see www.axmedis.org
   iv. "**Document**" means the information contained in any electronic file, which has been published by the Licensor's as AXMEDIS official document and listed in the web site mentioned above or available by any other means.
   v. "**Works**" means any works created by the licensee, which reproduce a Document or any of its part.
2. **LICENCE**
   1. The Licensor grants a non-exclusive royalty free licence to reproduce and use the Documents subject to present terms and conditions (the **Licence**) for the parts that are own and proprietary property the of AXMEDIS consortium or its members.
   2. In consideration of the Licensor granting the Licence, licensee agrees to adhere to the following terms and conditions.
3. **TERM AND TERMINATION**
   1. Granted Licence shall commence on Acceptance Date.
   2. Granted Licence will terminate automatically if licensee fails to comply with any of the terms and conditions of this Licence.
   3. Termination of this Licence does not affect either party's accrued rights and obligations as at the date of termination.
   4. Upon termination of this Licence for whatever reason, licensee shall cease to make any use of the accessed Copyright.
   5. All provisions of this Licence, which are necessary for the interpretation or enforcement of a party's rights or obligations, shall survive termination of this Licence and shall continue in full force and effect.
   6. Notwithstanding License termination, confidentiality clauses related to any content, document or part of it as stated in the document itself will remain in force for a period of 5 years after license issue date or the period stated in the document whichever is the longer.
4. **USE**
   1. Licensee shall not breach or denigrate the integrity of the Copyright Notice and in particular shall not:
      i. remove this Copyright Notice on a Document or any of its reproduction in any form in which those may be achieved;
      ii. change or remove the title of a Document;
      iii. use all or any part of a Document as part of a specification or standard not emanating from the Licensor without the prior written consent of the Licensor; or
      iv. do or permit others to do any act or omission in relation to a Document which is contrary to the rights and obligations as stated in the present license and agreed with the Licensor
5. **COPYRIGHT NOTICES**
   1. All Works shall bear a clear notice asserting the Licensor's Copyright. The notice shall use the wording employed by the Licensor in its own copyright notice unless the Licensor otherwise instructs licensees.
6. **WARRANTY**
   1. The Licensor warrants the licensee that the present licence is issued on the basis of full Copyright ownership or re-licensing agreements granting the Licensor full licensing and enforcement power.
   2. For the avoidance of doubt the licensee should be aware that although the Copyright in the documents is given under warranty this warranty does not extend to the content of any document which may contain references or specifications or technologies that are covered by patents (also of third parties) or that refer to other standards. AXMEDIS is not responsible and does not guarantee

that the information contained in the document is fully proprietary of AXMEDIS consortium and/or partners.

3. Licensee hereby undertakes to the Licensor that he will, without prejudice to any other right of action which the Licensor may have, at all times keep the Licensor fully and effectively indemnified against all and any liability (which liability shall include, without limitation, all losses, costs, claims, expenses, demands, actions, damages, legal and other professional fees and expenses on a full indemnity basis) which the Licensor may suffer or incur as a result of, or by reason of, any breach or non-fulfillment of any of his obligations in respect of this License.

7. **INFRINGEMENT**

1. Licensee undertakes to notify promptly the Licensor of any threatened or actual infringement of the Copyright which comes to licensee notice and shall, at the Licensor's request and expense, do all such things as are reasonably necessary to defend and enforce the Licensor's rights in the Copyright.

8. **GOVERNING LAW AND JURISDICTION**

1. This Licence shall be subject to, and construed and interpreted in accordance with Italian law.
2. The parties irrevocably submit to the exclusive jurisdiction of the Italian Courts.

Please note that:

- You can become affiliated with AXMEDIS. This will give you the access to a huge amount of knowledge, information and source code related to the AXMEDIS Framework. If you are interested please contact P. Nesi at nesi@dsi.unifi.it. Once affiliated with AXMEDIS you will have the possibility of using the AXMEDIS specification and technology for your business.
- You can contribute to the improvement of AXMEDIS documents and specification by sending the contribution to P. Nesi at nesi@dsi.unifi.it
- You can attend AXMEDIS meetings that are open to public, for additional information see WWW.axmedis.org or contact P. Nesi at nesi@dsi.unifi.it

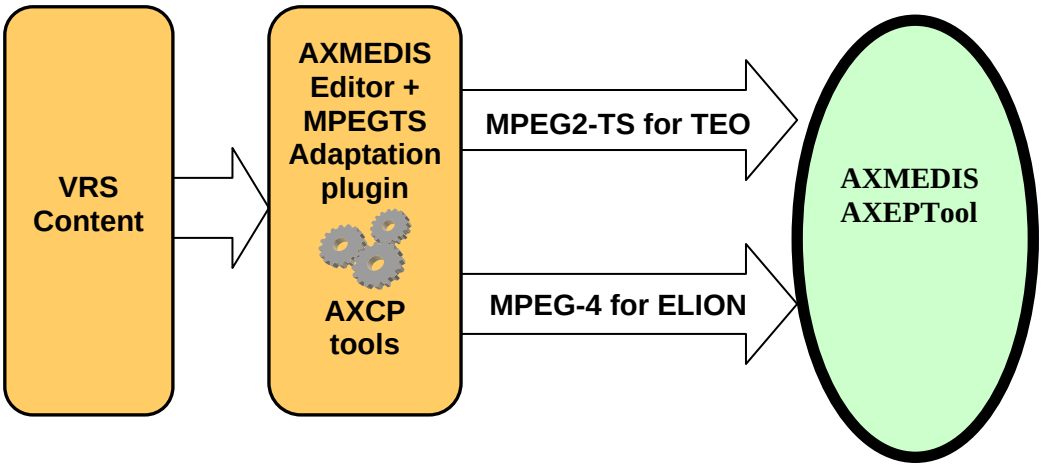
Table of Contents

1 Executive Summary and Report Scope

This report documents the work undertaken by AXMEDIS ELTEO take-up partner VRS. VRS is a content partner charged with validation of AXMEDIS content modeling tools and mechanisms by formatting its proprietary content, namely video clips from the funny hidden camera series VRS Kamera. This report describes the general architecture of VRS AXMEDIS content processing tools, their integration with real-life third-party video processing tools, and use cases and scenarios of a content producer producing video content for IPTV streaming and downloading on PC. It also specifies the VRS content processed through AXMEDIS system.

2 Introduction

The key role of VRS in AXMEDIS demonstrator was to provide video content for ELTEO partners: IPTV STB (TEO) and PC (Elion). Schematic description of VRS factory integration into ELTEO demonstrator is depicted below.

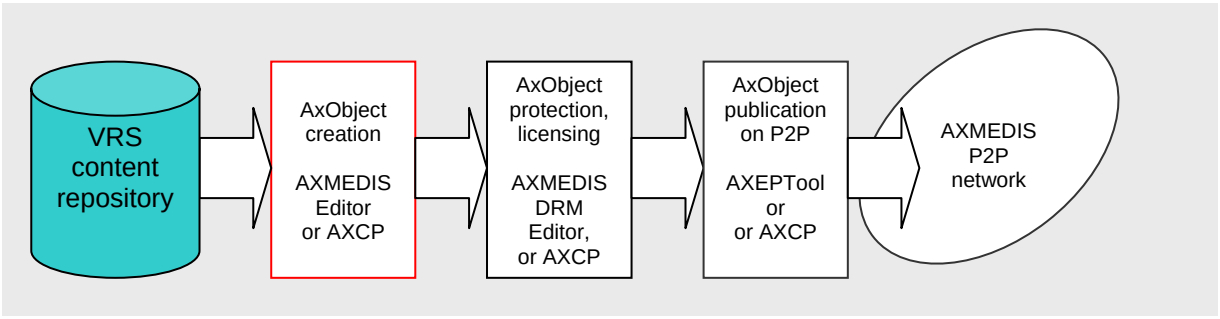


VRS objective was to test AXMEDIS Editor for manual content production and AXCP tools for automated content processing, specifically:

- to create protected AXMEDIS object;
- to publish it to P2P B2B network;
- to issue the mother licenses for TEO VoD users and Elion PC users.

3 General architecture of content production and processing demonstrator (VRS)

The figure below illustrates the basic VRS factory model.



The content processing workflow consists of the following steps and tools used:

VRS content processing steps	External tools used	AXMEDIS tools used
1. Extraction of content from VRS content repository	Sony Vegas Pro 8, Elecard XMuxer	AXCP Rule Editor
2. AxObject creation	-	AXMEDIS Editor, AXCP RuleEditor
3. AxObject publication on P2P	-	AXEPTool, AXCP Rule Editor

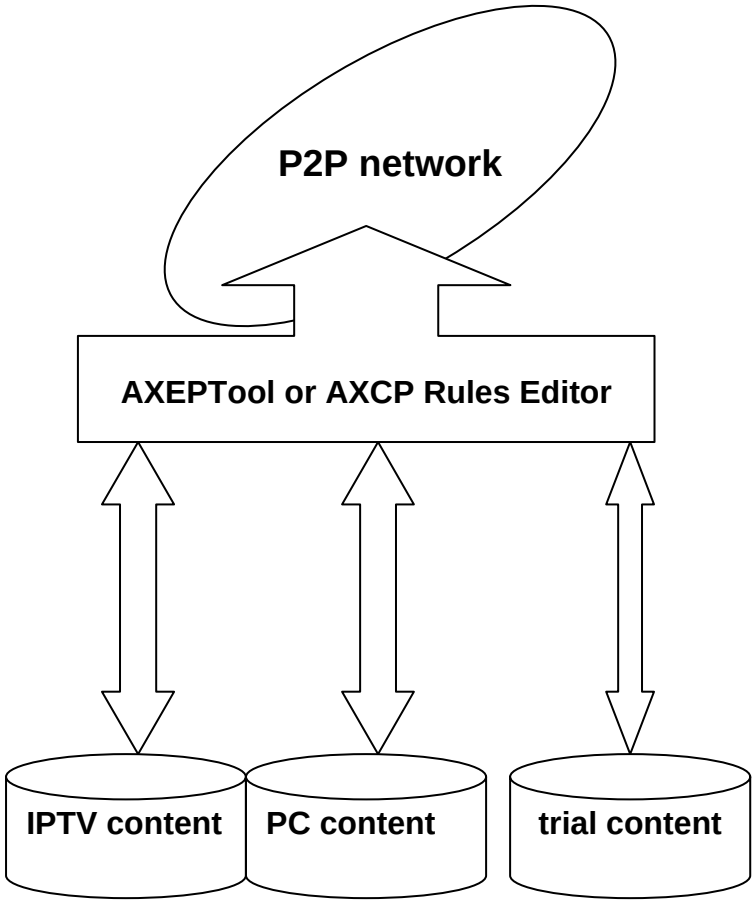
This basic workflow can be done manually using AXMEDIS Editor, AXMEDIS DRM Editor and AXEPTool, or using the number of AXMEDIS rules (Create AxObject, etc…), processed with the AXCP Rule Editor.

3.1 Structure of AXMEDIS content repository

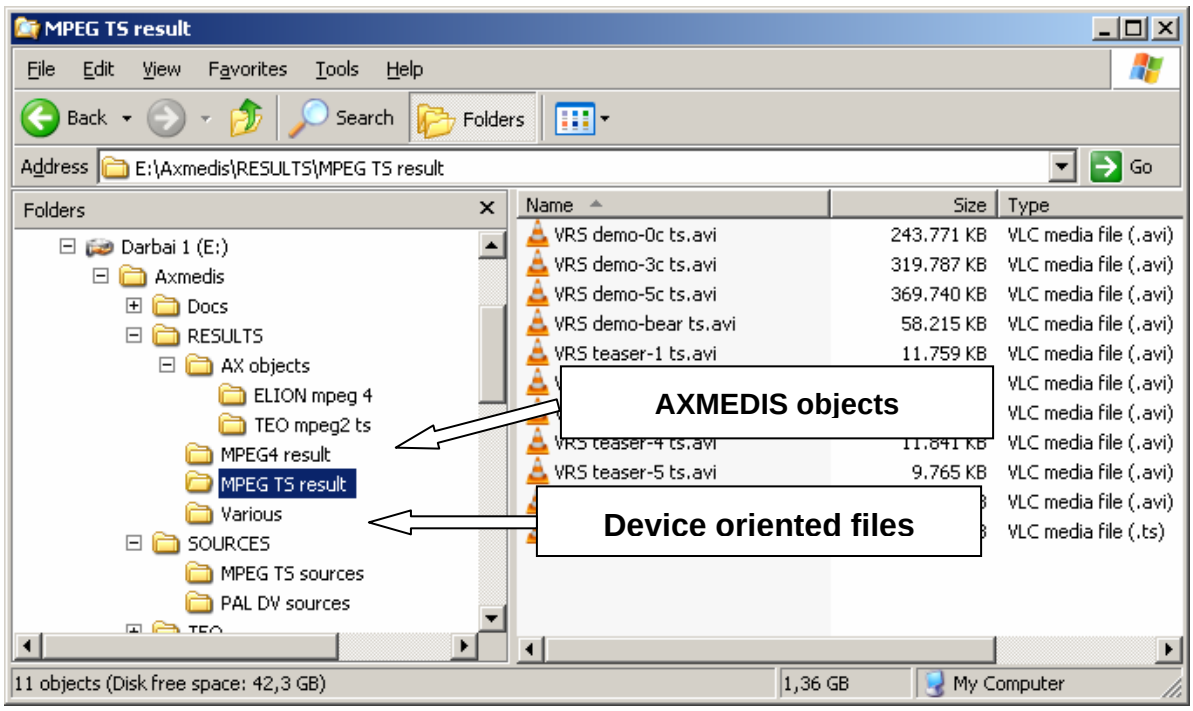
3.1.1 General description

VRS factory content is placed in local filesystem and is based on device-oriented formats:

- AxObjects for TEO IPTV STB,
- AxObjects for ELION PC
- Trial AxObjects for testing



VRS factory basic file system structure



3.1.2 Minimal set of data needed

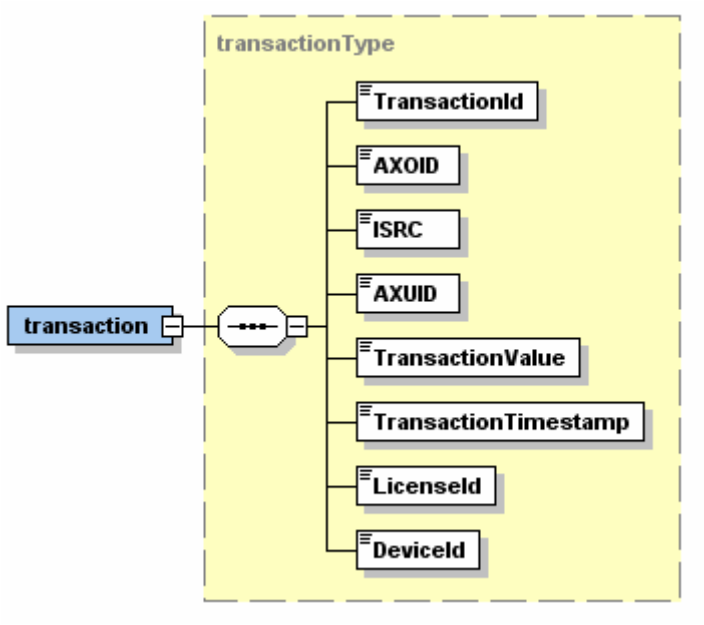
The minimal set of data necessary for administrative purposes:

Data Name	Data Type	Data Description
TransactionId	String	Unique transaction ID
AXOID	String	Unique descriptor of the object inside the AXMEDIS system
ISRC	String	Unique standard ID for identifying the piece
AXUID	String	User that has performed the transaction
TransactionValue	String	The overall value of the transaction
TransactionTimestamp	DateTime	Timestamp of transaction occurrence
LicenseId	String	The identifier of the license used to grant access to the object
DeviceId	String	STB which consumed content fingerprint

3.1.3 File format for data input

The file format can be one of the following:

- XML file Y (if yes provide an XML schema)



VRS Exchange file Data Format XSD Schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
	<xs:complexType name="transactionType">
		<xs:sequence>
			<xs:element name="TransactionId" type="xs:string"/>
			<xs:element name="AXOID" type="xs:string"/>
			<xs:element name="ISRC" type="xs:string"/>
			<xs:element name="AXUID" type="xs:string"/>
			<xs:element name="TransactionValue" type="xs:string"/>
			<xs:element name="TransactionTimestamp"
type="xs:dateTime"/>
			<xs:element name="LicenseId" type="xs:string"/>
			<xs:element name="DeviceId" type="xs:string"/>
		</xs:sequence>
	</xs:complexType>
	<xs:element name="transaction" type="transactionType"/>
</xs:schema>
```

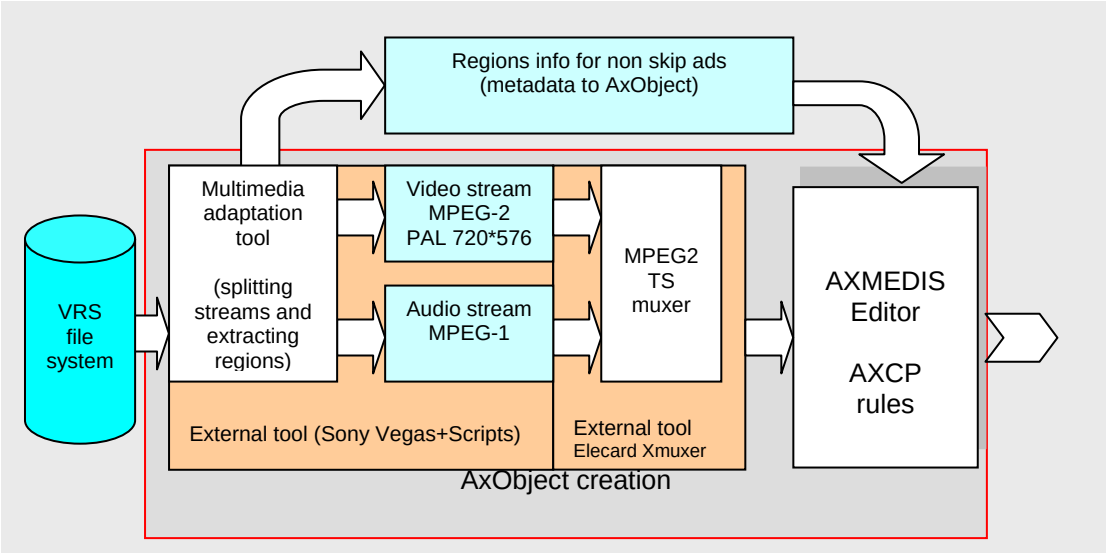
- CSV (comma separated values) N (if yes specify field sequence)
- Fixed length format N (if yes specify format)
- Other plain text format N (if yes specify format)

3.2 AxObject creation

The explication below shows the extension of AXMEDIS Object creation section to produce protected AxObject for ELTEO demonstrator.

- The regions for non-skippable ads must be defined
- Due to the IPTV STB hardware requirements (RAW UDP streaming) - MPEG2 TS resource must be embedded into AxObject;
- This MPEG21 Digital Item scrambled with AXCP MPEGTSAdaptationPlugin using DVB-CSA algorithm;

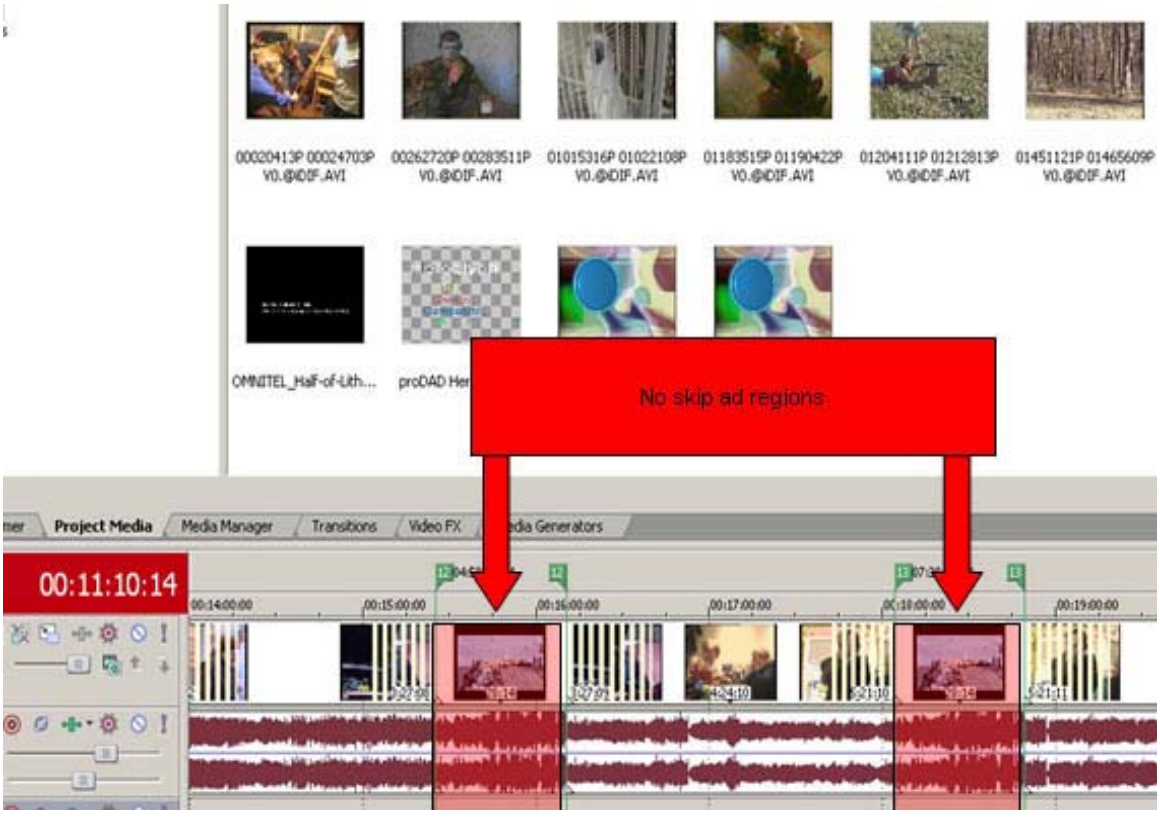
- This transformed AXMEDIS Object protected using AXOM Protection Processor and put to AXDB;



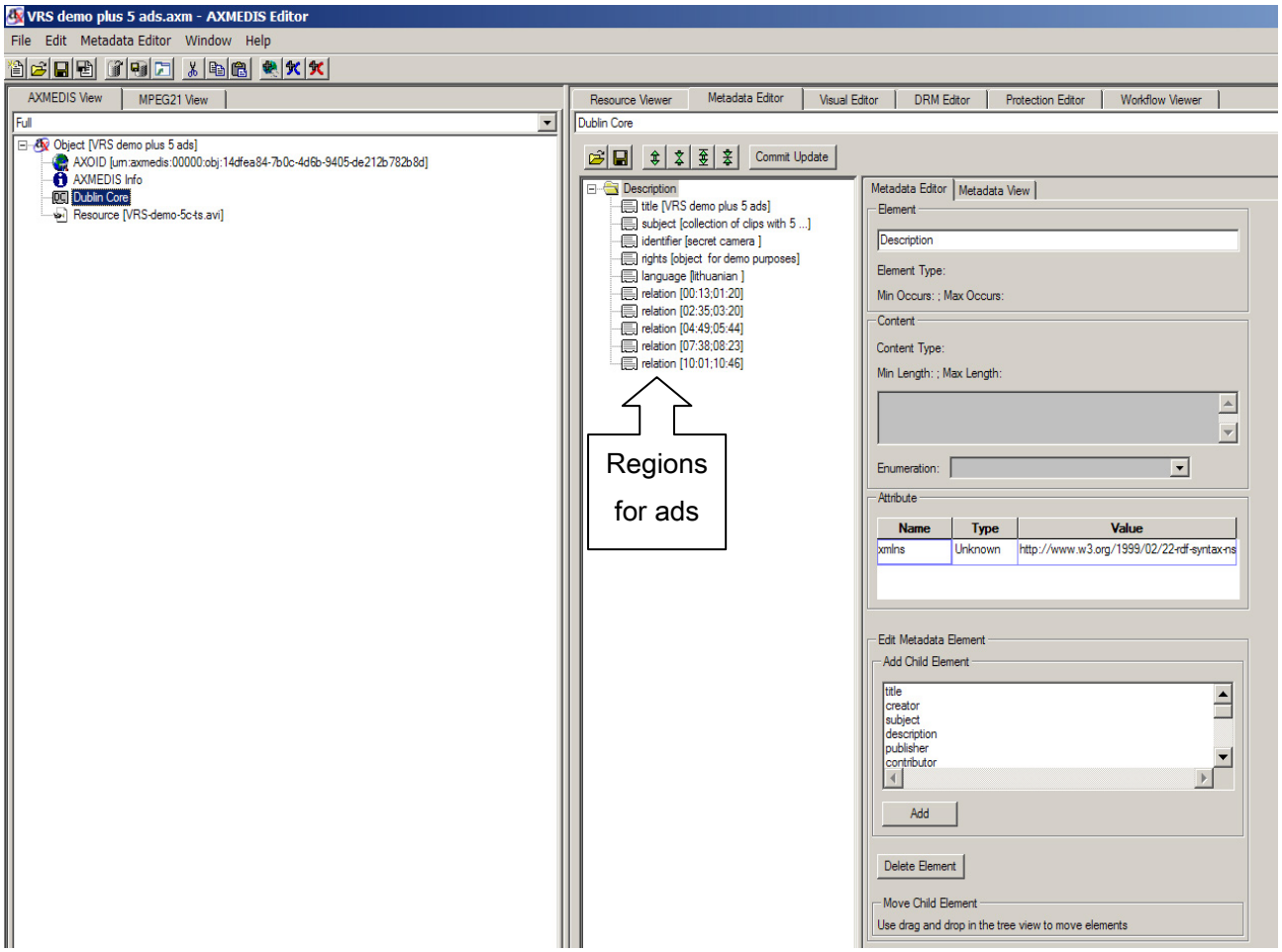
- Metadata in Axmedis objects describes general ranking of VRS media files – themes of video content etc. Also metadata contains the regions for embedded non-skippable ads, with the beginning and end time based on programme clock reference transported within MPEG-TS.

Firstly in the process it is necessary to define inserted ad regions, exporting the timecodes to XML file. This later will be used as metadata in AxObject to define non skippable regions location for PCR of the STB. Also the same external tool - Sony Vegas is splitting interleaved broadcast ready media file to video and audio streams according to desired template of STB – video stream MPEG-2 CBR 4Mbps, 720x576, 25 fps and audio stream MPEG-1 224 Kbps.

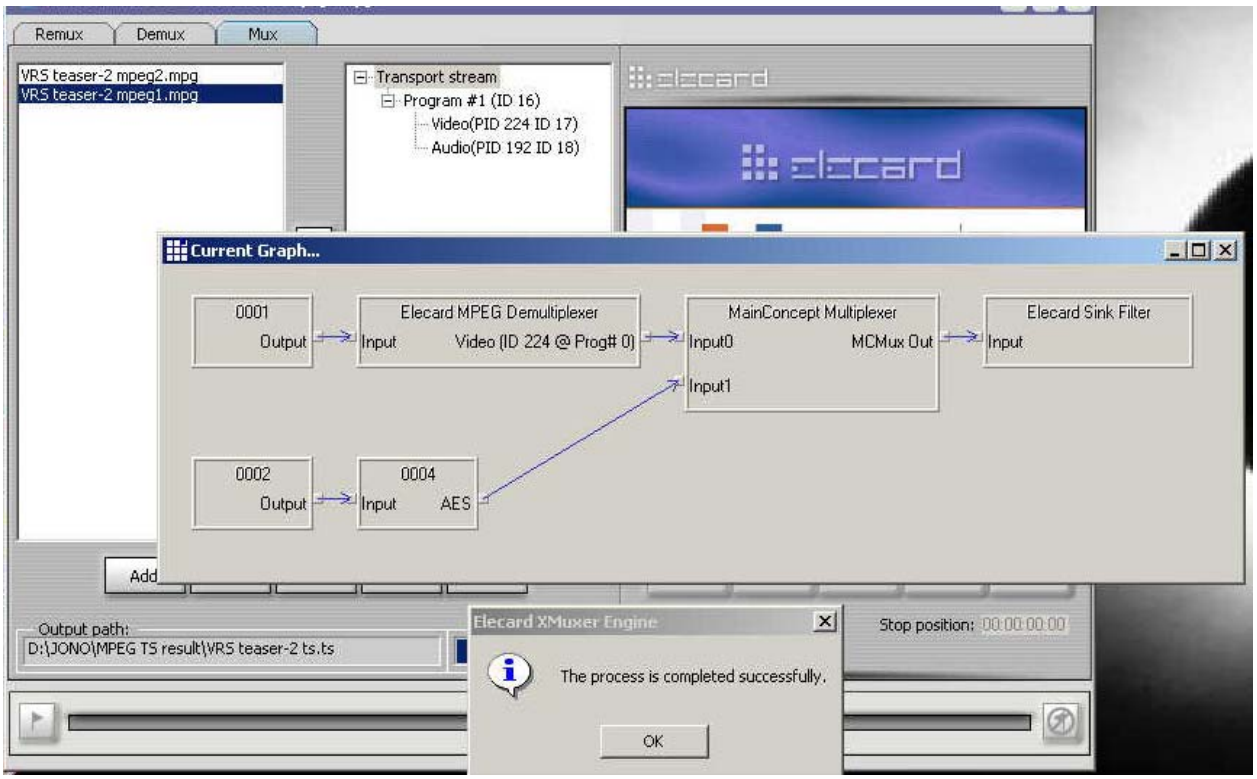
The picture below illustrates creation of regions for non skippable ads with external tool.

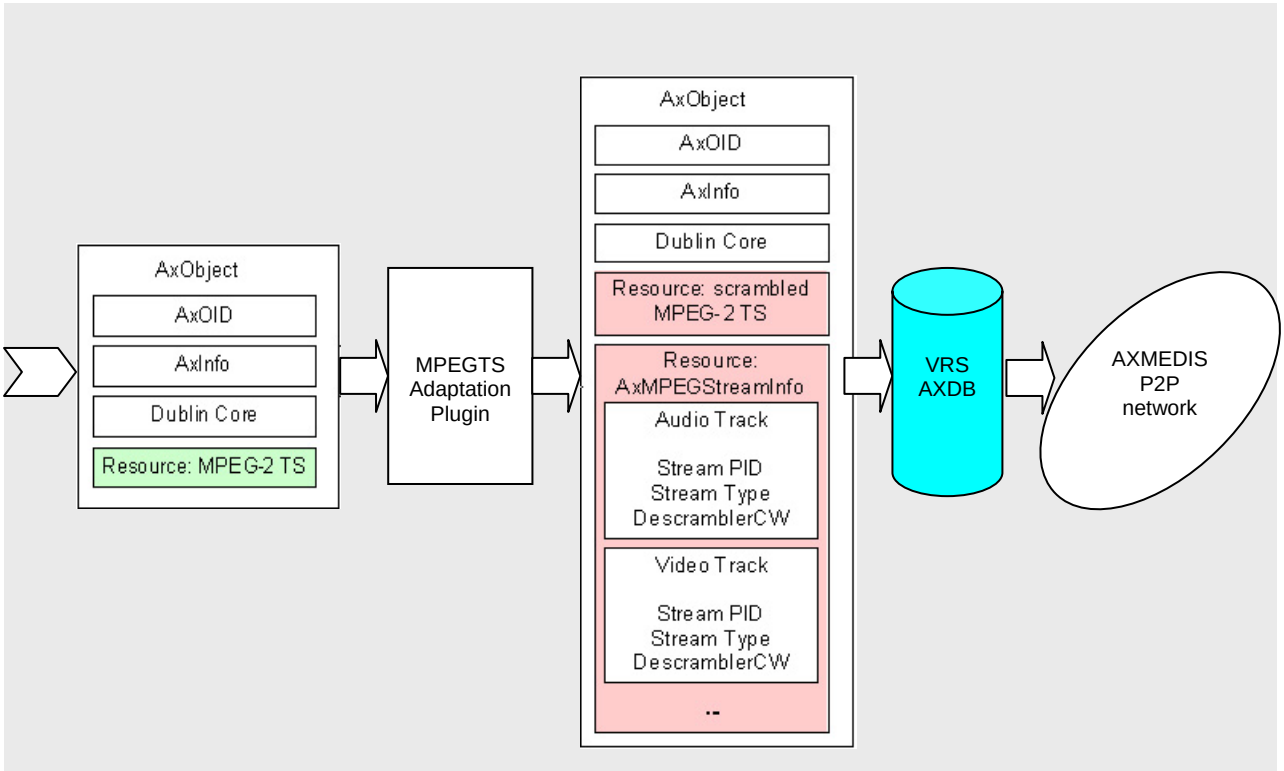


The regions TC exported to XML file, later used in MPEG-21 metadata DC



Another external tool Elecard XMuxer is called to format MPEG-2 TS file from the streams:





The produced content has to be adapted for streaming requirements at VRS Factory using AXMEDIS Editor or AXCP. For this purpose the AXCP MPEGTSAdaptationPlugin has been developed by ELTEO partners TEO and KTU. This plug-in performs two tasks:

- Scrambles MPEG-2 Transport Stream (TS) resource using DVB-CSA algorithm. The MPEGTSAdaptationPlugin must encrypt all video and audio channels of stream and leave unencrypted frames containing Program Clock Reference (PCR) which is critical for streaming control. Each scrambled frame of transport stream has to be marked as scrambled according to ISO13818-1 standard.

- The second task of plug-in is to collect information about all scrambled elementary MPEG-2 TS streams (ES). This task associates each ES with DVB-CSA Control Word (CW) which was used for scrambling the ES. This information is returned by the plug-in as a new RESOURCE which is appended to the transformed AXMEDIS Object.

After adoption the transformed AXMEDIS Object must consist of:

1. Scrambled MPEG-2 Transport Stream;
2. Information about scrambled MPEG-2 TS elementary streams and DVB-CSA CWs in resource AxMPEGStreamInfo.

This transformed AXMEDIS Object is protected using AXOM and has to be put to VRS AXDB.

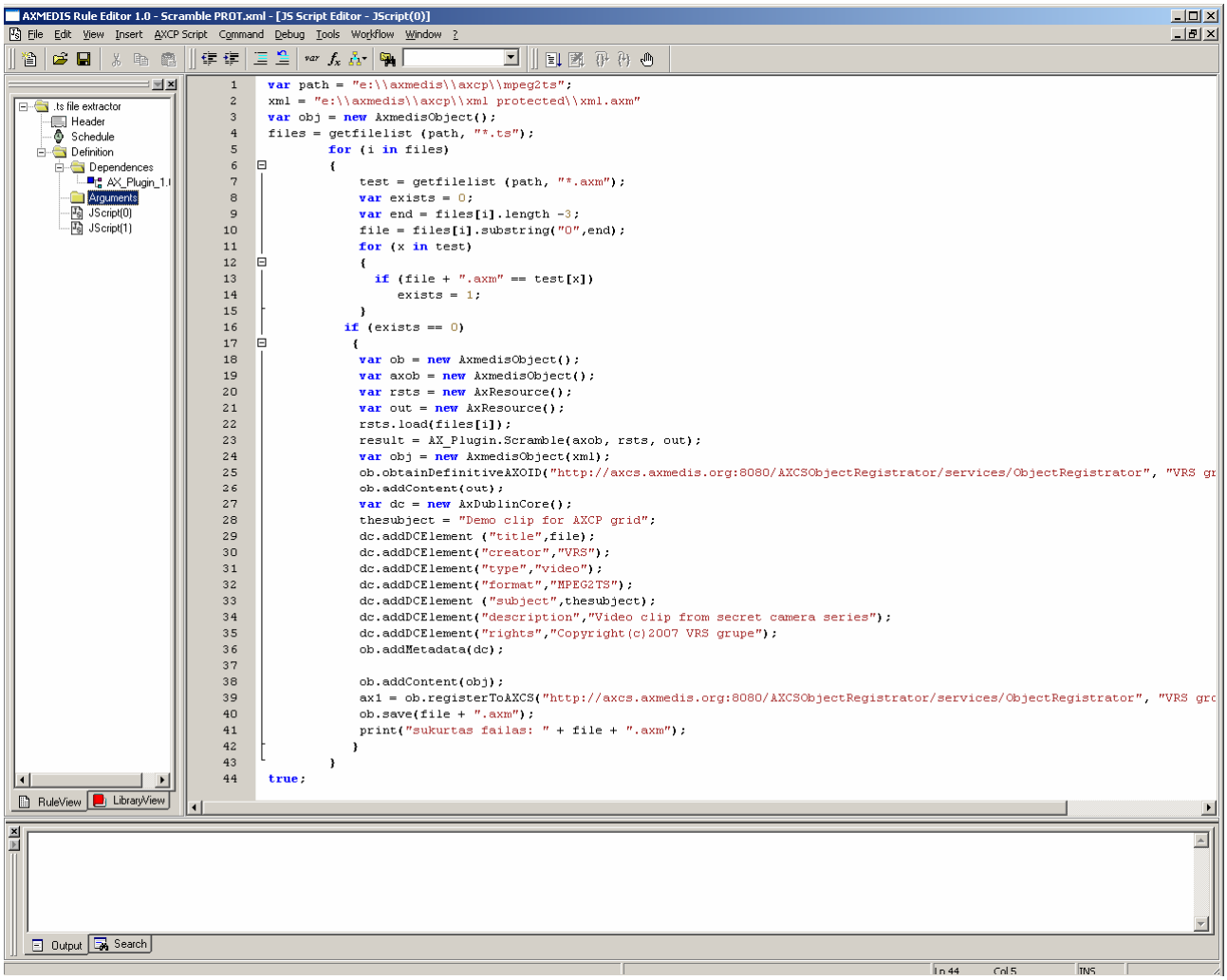
All above steps described is proceeded with the single AXCP rule - **Scramble.axr**

```
<?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>.ts file extractor</Rule_Name>
    <AXRID>axcprule:e3cb5098-08b0-4905-b017-165f8d289f6c</AXRID>
    <Rule_Version></Rule_Version>
    <Rule_Type>AXCP</Rule_Type>
    <Software_Name></Software_Name>
    <Version_of_software></Version_of_software>
    <Date_of_production>2006-11-29</Date_of_production>
    <Author></Author>
    <Affiliation>KTU</Affiliation>
    <URL></URL>
    <Comment></Comment>
    <Last_Modifications>2008-01-17</Last_Modifications>
    <Terminal_ID></Terminal_ID>
    <Cost></Cost>
    <Work_Item_ID></Work_Item_ID>
  </Header>
  <Schedule>
    <Run>
      <Date>2007-07-09</Date>
      <Time>09:21:10</Time>
      <Periodicity Unit="Minute">1</Periodicity>
      <Expiration_Date>2007-11-29</Expiration_Date>
      <Expiration_Time>09:21:10</Expiration_Time>
    </Run>
    <Status>Inactive</Status>
  </Schedule>
  <Definition>
    <AXCP_Rule>
      <Arguments/>
      <Rule_Body>
        <JS_Script name="JScript(0)"><![CDATA[var path =
"e:\\axmedis\\axcp\\mpeg2ts";
xml = "e:\\axmedis\\axcp\\xml protected\\xml.axm"
var obj = new AxmedisObject();
files = getfilelist (path, "*.ts");
        for (i in files)
        {
            test = getfilelist (path, "*.axm");
```

```
            var exists = 0;
            var end = files[i].length -3;
            file = files[i].substring("0",end);
            for (x in test)
            {
              if (file + ".axm" == test[x])
                 exists = 1;
            }
          if (exists == 0)
           {
            var ob = new AxmedisObject();
            var axob = new AxmedisObject();
            var rsts = new AxResource();
            var out = new AxResource();
            rsts.load(files[i]);
            result = AX_Plugin.Scramble(axob, rsts, out);
            var obj = new AxmedisObject(xml);

ob.obtainDefinitiveAXOID("http://axcs.axmedis.org:8080/AXCSObjectRegi
strator/services/ObjectRegistrator", "VRS group", "824ac61d");
            ob.addContent(out);
            var dc = new AxDublinCore();
            thesubject = "Demo clip for AXCP grid";
            dc.addDCElement ("title",file);
            dc.addDCElement("creator","VRS");
            dc.addDCElement("type","video");
            dc.addDCElement("format","MPEG2TS");
            dc.addDCElement ("subject",thesubject);
            dc.addDCElement("description","Video clip from secret
camera series");
            dc.addDCElement("rights","Copyright(c)2007 VRS grupe");
            ob.addMetadata(dc);

            ob.addContent(obj);
            ax1 =
ob.registerToAXCS("http://axcs.axmedis.org:8080/AXCSObjectRegistrator
/services/ObjectRegistrator", "VRS group", "824ac61d");
            ob.save(file + ".axm");
            print("sukurtas failas: " + file + ".axm");
           }
        }
true;]]>
        </JS_Script>
        <JS_Script name="JScript(1)">

          <![CDATA[function getfilelist (
getflielist_dir,        //directory
getflielist_mask        // mask e.g. "*.*"
)
{
  var list = new Array();
  if (!existsDir(getflielist_dir))  // check that the directory
exists
  {
         return (list)
  }
  var file = getFirstFile(getflielist_dir, getflielist_mask);
  if (file != null)
```

```
  {
    list.push (file);
  }
  while (file != null)
  {
     var nextfile = getNextFile();
     if (nextfile == null)
     {
       break;
     }
     list.push (nextfile);
     nextfile = undefined;
    }
  return (list);
}]]>
        </JS_Script>
      </Rule_Body>
      <Dependencies>
        <Dependency>
          <Plug_In_name>AX_Plugin</Plug_In_name>
          <Version>1.001</Version>
        </Dependency>
      </Dependencies>
    </AXCP_Rule>
  </Definition>
</Rule>
```

AXCP Rule Editor is used to process the content for TEO IPTV STB automatically:



Then the license generation rule is proceeded to generate licenses to AXIODs:

```
<?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>ScriptLicensingOnDemand</Rule_Name>
    <AXRID>axcprule:72f9a7a8-5ef0-49c1-9667-9dbbd36745e2</AXRID>
    <Rule_Version></Rule_Version>
    <Rule_Type>AXCP</Rule_Type>
    <Software_Name></Software_Name>
    <Version_of_software></Version_of_software>
    <Date_of_production>2006-10-23</Date_of_production>
    <Author></Author>
    <Affiliation></Affiliation>
    <URL></URL>
    <Comment></Comment>
    <Last_Modifications>2008-01-17</Last_Modifications>
    <Terminal_ID></Terminal_ID>
    <Cost></Cost>
    <Work_Item_ID></Work_Item_ID>
  </Header>
  <Schedule>
    <Run>
```

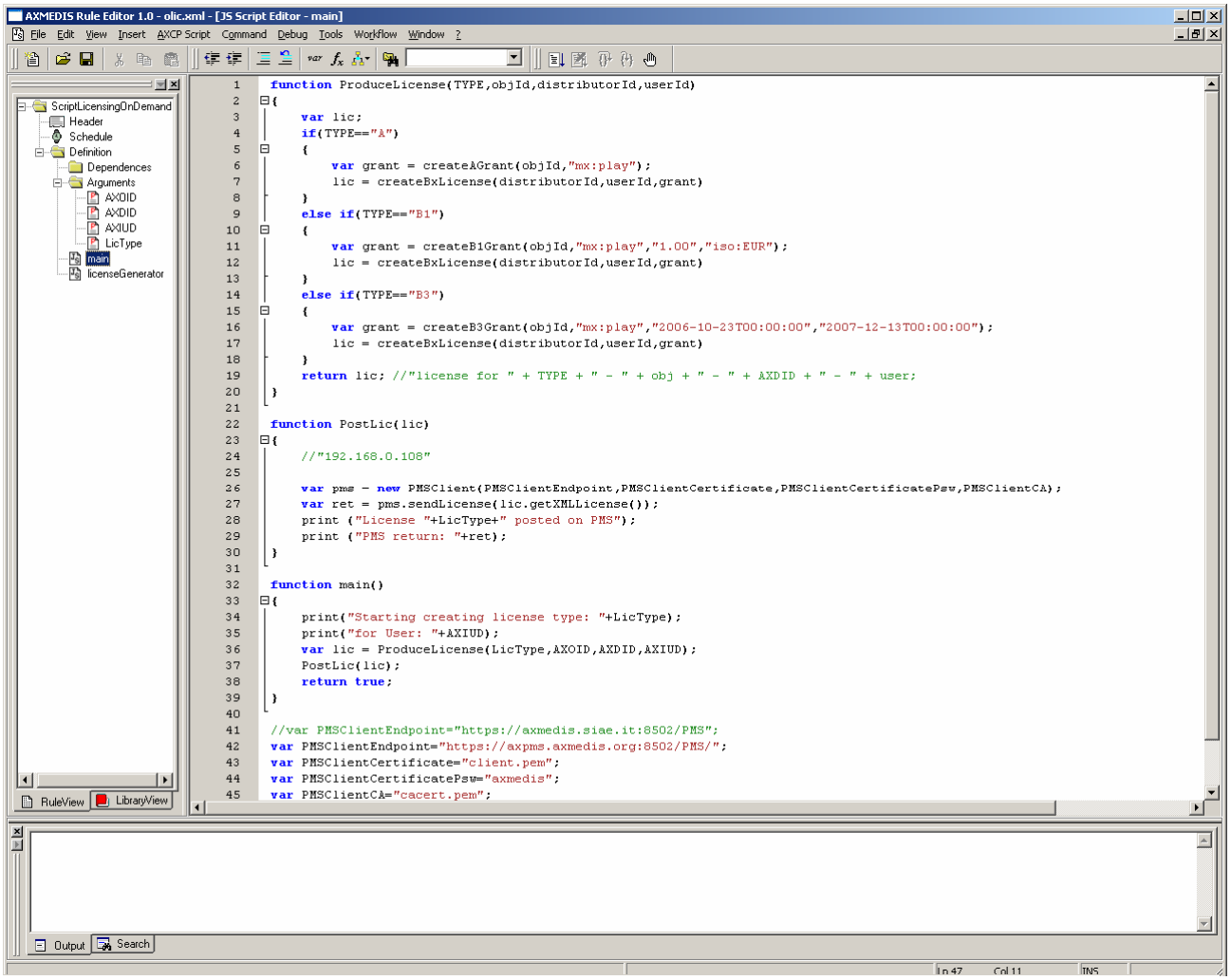
```
      <Date>2006-10-23</Date>
      <Time>23:59:49</Time>
      <Periodicity Unit="Day">0</Periodicity>
      <Expiration_Date>2006-10-23</Expiration_Date>
      <Expiration_Time>23:59:49</Expiration_Time>
    </Run>
    <Status>Inactive</Status>
  </Schedule>
  <Definition>
    <AXCP_Rule>
      <Arguments>
        <Parameter Name="AXOID" Type="String">@AXOID@</Parameter>
        <Parameter Name="AXDID" Type="String">@AXDID@</Parameter>
        <Parameter Name="AXIUD" Type="String">@AXUID@</Parameter>
        <Parameter Name="LicType" Type="String">A</Parameter>

      </Arguments>
      <Rule_Body>
        <JS_Script name="main"><![CDATA[function
ProduceLicense(TYPE,objId,distributorId,userId)
{
    var lic;
    if(TYPE=="A")
    {
        var grant = createAGrant(objId,"mx:play");
        lic = createBxLicense(distributorId,userId,grant)
    }
    else if(TYPE=="B1")
    {
        var grant = createB1Grant(objId,"mx:play","1.00","iso:EUR");
        lic = createBxLicense(distributorId,userId,grant)
    }
    else if(TYPE=="B3")
    {
        var grant = createB3Grant(objId,"mx:play","2006-10-
23T00:00:00","2007-12-13T00:00:00");
        lic = createBxLicense(distributorId,userId,grant)
    }
    return lic; //"license for " + TYPE + " - " + obj + " - " + AXDID
+ " - " + user;
}

function PostLic(lic)
{
    //"192.168.0.108"

    var pms = new
PMSClient(PMSClientEndpoint,PMSClientCertificate,PMSClientCertificate
Psw,PMSClientCA);
    var ret = pms.sendLicense(lic.getXMLLicense());
    print ("License "+LicType+" posted on PMS");
    print ("PMS return: "+ret);
}

function main()
{
    print("Starting creating license type: "+LicType);
    print("for User: "+AXIUD);
```

```
    var lic = ProduceLicense(LicType,AXOID,AXDID,AXIUD);
    PostLic(lic);
    return true;
}

//var PMSClientEndpoint="https://axmedis.siae.it:8502/PMS";
var PMSClientEndpoint="https://axpms.axmedis.org:8502/PMS/";
var PMSClientCertificate="client.pem";
var PMSClientCertificatePsw="axmedis";
var PMSClientCA="cacert.pem";

a = main();]]>
        </JS_Script>
        <JS_Script name="licenseGenerator">


<![CDATA[/*********************************************************
* A License Generation functions from MB License *
*********************************************************/
// Completed - it works
function createALicense(creatorId,resId,userId)
{
    return
createMotherLicense(creatorId,resId,userId,"mx:play",null);
}
/*********************************************************
*   Grant: function for creating grant for A license    *
*********************************************************/
function createAGrant(resId, right)
{
     var grant = new Grant();
     var rights = new Right();
     rights.setRight(right);
     var res = new Resource();
     res.resourceId = resId;
     res.diType = 1;
     grant.setResource(res);
     grant.setRight(rights);
     return grant;
}
/*********************************************************
* B License Generation functions from MB License *
*********************************************************/
//it works but conditions check is missing
function createB1FromMB1License(lic,userId)
{
   var b1Lic = new License();
   var grant = lic.grantgroup.grants[0];
   var principal = grant.getPrincipal();
   var subgrant = grant.resource.ggResource.grants[0];
   var rIssuer = new Issuer();
   rIssuer.setIssuer(principal.name);
   var rGrantGroup=new GrantGroup();
   var rGrant = new Grant()
   rGrant.setPrincipal(grant.principal);
   rGrant.setResource(subgrant.resource);
   rGrant.setRight(subgrant.right);
   rGrantGroup.addGrant(rGrant);
```

```
   b1Lic.setGrantGroup(rGrantGroup);
   b1Lic.setIssuer(rIssuer);
   return b1Lic;
}
// Missing the Interval or the way to set the
"validityIntervalFloating"
function createB2FromMB2License(lic,userId)
{
   var b2Lic = new License();
   var grant = lic.grantgroup.grants[0];
   var principal = grant.getPrincipal();
   var subgrant = grant.resource.ggResource.grants[0];

   var rIssuer = new Issuer();
   rIssuer.setIssuer(principal.name);
   var rGrantGroup=new GrantGroup();
   var rGrant = new Grant()
   rGrant.setPrincipal(grant.principal);
   rGrant.setResource(subgrant.resource);
   rGrant.setRight(subgrant.right);
   rGrantGroup.addGrant(rGrant);
   b2Lic.setGrantGroup(rGrantGroup);
   b2Lic.setIssuer(rIssuer);
   return b2Lic;
}
// Completed - it works
function createB3FromMB3License(lic,userId)
{
   var b3Lic = new License();
   var grant = lic.grantgroup.grants[0];
   var principal = grant.getPrincipal();
   var subgrant = grant.resource.ggResource.grants[0];

   var rIssuer = new Issuer();
   rIssuer.setIssuer(principal.name);
   var rGrantGroup=new GrantGroup();
   var rGrant = new Grant()
   rGrant.setPrincipal(grant.principal);
   rGrant.setResource(subgrant.resource);
   rGrant.setRight(subgrant.right);
   rGrantGroup.addGrant(rGrant);
   b3Lic.setGrantGroup(rGrantGroup);
   b3Lic.setIssuer(rIssuer);
   return b3Lic;
}
/*************************************************
*           Bx License Generation functions         *
*************************************************/
function createB3Grant(resId, right, notBefore, notAfter)
{
        var grant = new Grant();
        var rights = new Right();
        var timecondition = new Interval();
        var limit = new LNumber();
        var res = new Resource();
        res.resourceId = resId;
        res.diType = 1;
        limit.count = 10;
```

```
        rights.setRight(right)
        timecondition.notBefore = notBefore;
        timecondition.notAfter = notAfter;
        grant.addCondition(timecondition);
        grant.addCondition(limit);
        grant.setRight(rights);
        grant.setResource(res);
        return grant;
}
function createB2Grant(resId, right, time)
{
        var grant = new Grant();
        var rights = new Right();
        var res = new Resource();
        res.resourceId = resId;
        res.diType = 1;
        rights.setRight(right);
        grant.setResource(res);
        grant.addCondition(timecondition);
        grant.addCondition(limit);
        grant.setRight(right);
        return grant;
}
function createB1Grant(resId, right, amount, currency)
{
     var grant = new Grant();
     var fee = new Fee();
     var rights = new Right();
     rights.setRight(right);
     fee.amount = parseFloat(amount);
     fee.currency = "iso:EUR";
     fee.tipoFee = 1;
     var res = new Resource();
     res.resourceId = resId;
     res.diType = 1;
     grant.setResource(res);
     grant.addCondition(fee);
     grant.setRight(rights);
     return grant;
}
/*************************************************
*     Bx License Generation function from grant     *
*************************************************/
function createBxLicense(distributorId,userId,grant)
{
    var license = new License();
    var grantGroup = new GrantGroup();
    var issuer = new Issuer();
    var principal = new Principal();

    principal.setName(userId);
    grant.setPrincipal(principal);
    grantGroup.addGrant(grant);
    issuer.setIssuer(distributorId);
    license.setGrantGroup(grantGroup);
    license.setIssuer(issuer);
    return license;
}
```

```
/*****************************************************
*           Mother License Generation function      *
*****************************************************/
function createMotherLicense(creatorId,userId,grant)
{
    var license = new License();
    var grantGroup = new Array(new GrantGroup(),new GrantGroup());
    var grant0 = new Grant();
    var issuer = new Issuer();
    var rights = new Right();
    var principal = new Principal();
    var res = new Array (new Resource(),new Resource());

    //level 1
    grantGroup[1].addGrant(grant);

    //level 0 - grant Group
    res[0].setGrantGroup(grantGroup[1]);
    rights.setRight("r:issue");
    principal.setName(userId);
    grant0.setPrincipal(principal);
    grant0.setResource(res[0]);
    grant0.setRight(rights);
    grantGroup[0].addGrant(grant0);
    //level 0 - issuer
    issuer.setIssuer(creatorId);
    license.setGrantGroup(grantGroup[0]);
    license.setIssuer(issuer);
    return license;
}]]>
        </JS_Script>
      </Rule_Body>
      <Dependencies/>
    </AXCP_Rule>
  </Definition>
</Rule>
```



To create AxObjects for ELION PC demonstrator the AXCP rule **VRSobjectsforELION.axr** is used. It contains compilation of scripts – AxObject creation, scrambling, generating licenses, posting licenses on PMS, uploading AxObject to AXDB and posting to P2P network:

```
<?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>VRSFull</Rule_Name>
    <AXRID>axcprule:504175f0-2886-49e6-b952-6d6d0ed98af5</AXRID>
    <Rule_Version></Rule_Version>
    <Rule_Type>AXCP</Rule_Type>
    <Software_Name></Software_Name>
    <Version_of_software></Version_of_software>
    <Date_of_production>2007-12-14</Date_of_production>
    <Author></Author>
    <Affiliation></Affiliation>
    <URL></URL>
    <Comment></Comment>
    <Last_Modifications>2007-12-14</Last_Modifications>
    <Terminal_ID></Terminal_ID>
    <Cost></Cost>
    <Work_Item_ID></Work_Item_ID>
  </Header>
```

```
  <Schedule>
    <Run>
      <Date>2007-12-14</Date>
      <Time>08:30:30</Time>
      <Periodicity Unit="Day">0</Periodicity>
      <Expiration_Date>2007-12-14</Expiration_Date>
      <Expiration_Time>08:30:30</Expiration_Time>
    </Run>
    <Status>Inactive</Status>
  </Schedule>
  <Definition>
    <AXCP_Rule>
      <Arguments>
        <Parameter Name="mainMask" Type="String">*.ts</Parameter>
        <Parameter Name="inputDirectory"
Type="String">c:\AX\temp</Parameter>
        <Parameter Name="outputDirectory"
Type="String">c:\ax\temp</Parameter>
        <Parameter Name="AXDID"
Type="String">URN:AXMEDIS:00002:BUS:5A4C5289-F317-3EFA-A0D1-
77CC5DFBE7BE</Parameter>
        <Parameter Name="AXIUD"
Type="String">URN:AXMEDIS:00002:BUS:3A364EAD-48B5-3A38-80B0-
9B7E47759FD8</Parameter>

      </Arguments>
      <Rule_Body>
        <JS_Script name="_main"><![CDATA[    var
PMSClientEndpoint="https://axpms.axmedis.org:8502/PMS";
    var PMSClientCertificate="client.pem";
    var PMSClientCertificatePsw="axmedis";
    var PMSClientCA="cacert.pem";

dirToSingleObjects (inputDirectory, outputDirectory, mainMask);]]>
        </JS_Script>
        <JS_Script name="dirToSingleObjects">

          <![CDATA[// puts one file in one object and saves it in
outputDir (full path)
function strip_filename (URL)
{
    var file = URL;
      for (i=URL.length-1; i>= 0; i--) {
         if (URL[i] == "/" || URL[i] == "\\" )
         {
             file = file.slice(i+1);
             break;
         }
      }
    return (file);
}
function strip_extension (extesnsion_file)
{
    var no_extension_file = extesnsion_file;
      for (i=extesnsion_file.length-1; i>=0; i--) {
         if (extesnsion_file[i] == ".")
         {
             no_extension_file  = no_extension_file .slice(0,i);
```

```
            break;
          }
       }
    return (no_extension_file);
}
function getFilelist (
getflielist_dir,        //directory
getflielist_mask        // mask e.g. "*.*"
)
{
  if (getflielist_mask == undefined) getflielist_mask = "*.*";
  var list = new Array();
  if (!existsDir(getflielist_dir))
  // check that the directory exists
  {
         return (list)
  }
  var file = getFirstFile(getflielist_dir, getflielist_mask);
  if (file != null)
  {
    list.push (file);
  }
  while (file != null)
  {
     var nextfile = getNextFile();
     if (nextfile == null)
     {
       break;
     }
     list.push (nextfile);
     nextfile = undefined;
    }
  return (list);
}

function PostLic(lic)
{
    //"192.168.0.108"

    var pms = new
PMSClient(PMSClientEndpoint,PMSClientCertificate,PMSClientCertificate
Psw,PMSClientCA);
    var ret = pms.sendLicense(lic.getXMLLicense());
    print ("License posted on PMS");
    print ("PMS return: "+ret);
}
function ProduceLicense(TYPE,objId,distributorId,userId)
{
    var lic;
    if(TYPE=="A")
    {
        var grant = createAGrant(objId,"mx:play");
        lic = createMotherLicense(distributorId,userId,grant)
    }
    else if(TYPE=="A1")
    {
        var grant = createAGrant(objId,"mx:play");
        lic = createBxLicense(distributorId,userId,grant)
```

```
    }
    else if(TYPE=="B1")
    {
        var grant = createB1Grant(objId,"mx:play","1.00","iso:EUR");
        lic = createBxLicense(distributorId,userId,grant)
    }
    else if(TYPE=="B3")
    {
        var grant = createB3Grant(objId,"mx:play","2006-10-
23T00:00:00","2007-08-13T00:00:00");
        lic = createBxLicense(distributorId,userId,grant)
    }
    return lic; //"license for " + TYPE + " - " + obj + " - " + AXDID
+ " - " + user;
}
function fileToObject (filePath, outputDir)
{
    xml = "c:\\temp\\xml.axm";
    obj_final = new AxmedisObject ();

obj_final.obtainDefinitiveAXOID("http://axcs.axmedis.org:8080/AXCSObj
ectRegistrator/services/ObjectRegistrator", "VRS group", "824ac61d");
    obj_temp = new AxmedisObject ();

    res_input_ts = new AxResource ();
    res_input_ts.load (filePath);
    res_output_ts = new AxResource ();

    result = AX_Plugin.Scramble(obj_temp, res_input_ts,
res_output_ts); //gaunam scramblinta ts'a

    obj_final.addContent (res_output_ts);   //pridetas ts'as

    dc = obj_final.getDublinCore ();
    dc.addDCElement ("title",strip_extension(strip_filename
(filePath)));
    dc.addDCElement("creator","VRS");
    dc.addDCElement("type","video");
    dc.addDCElement("format","MPEG-4");
    dc.addDCElement ("subject","Demo clip for ELION");
    dc.addDCElement("description","Video clip from secret camera
series");
    dc.addDCElement("rights","Copyright(c)2007 VRS grupe");
//pridetas dc

    //makeXML();

    obj_xml = new AxmedisObject (xml);
    obj_final.addContent(obj_xml);     // pridetas uzprotectintas
xml'as


obj_final.registerToAXCS("http://axcs.axmedis.org:8080/AXCSObjectRegi
strator/services/ObjectRegistrator", "VRS group", "824ac61d");

    obj_final.save (outputDir+"\\"+strip_extension(strip_filename
(filePath))+".axm");
```

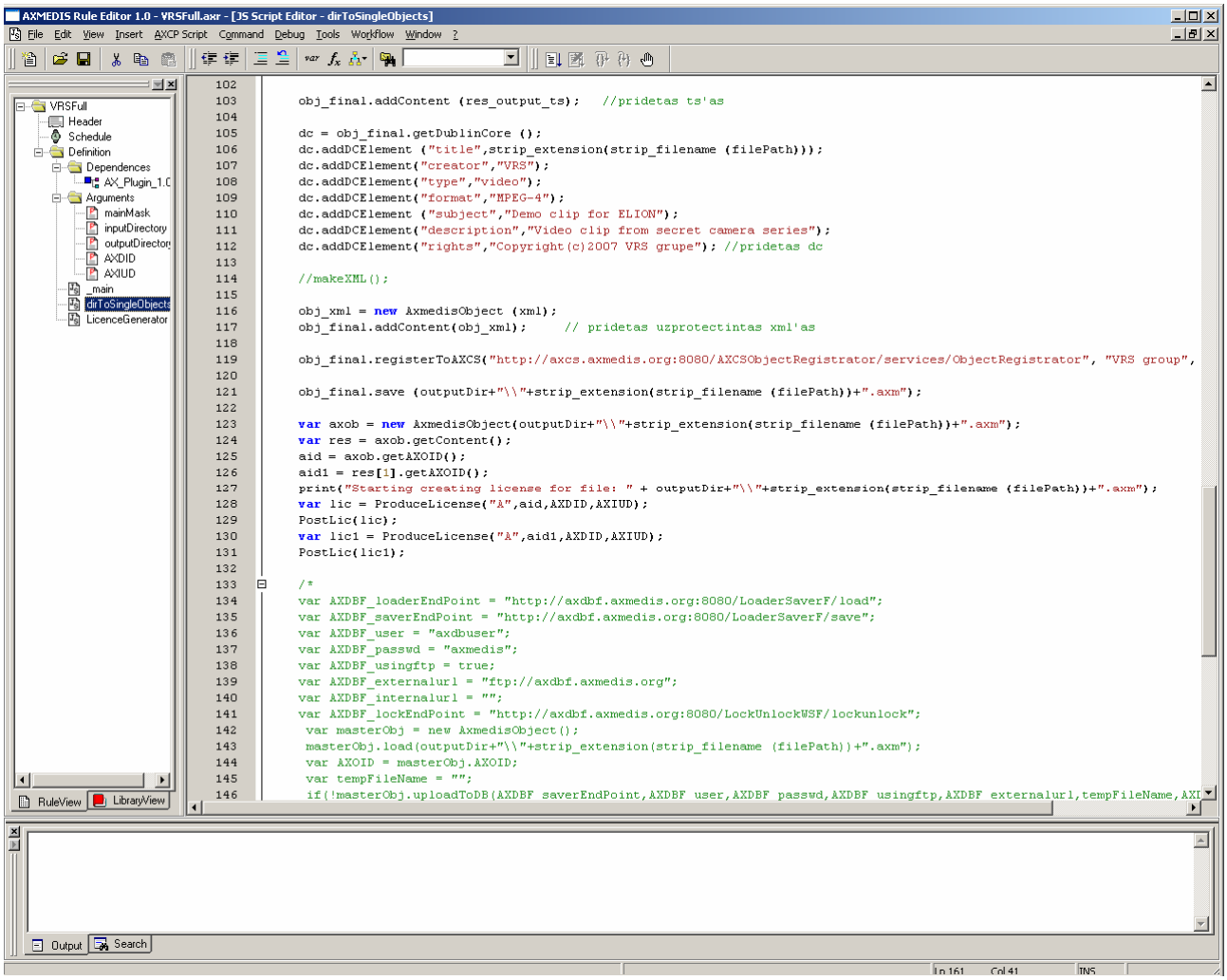
```
    var axob = new
AxmedisObject(outputDir+"\\"+strip_extension(strip_filename
(filePath))+".axm");
    var res = axob.getContent();
    aid = axob.getAXOID();
    aid1 = res[1].getAXOID();
    print("Starting creating license for file: " +
outputDir+"\\"+strip_extension(strip_filename (filePath))+".axm");
    var lic = ProduceLicense("A",aid,AXDID,AXIUD);
    PostLic(lic);
    var lic1 = ProduceLicense("A",aid1,AXDID,AXIUD);
    PostLic(lic1);

    /*
    var AXDBF_loaderEndPoint =
"http://axdbf.axmedis.org:8080/LoaderSaverF/load";
    var AXDBF_saverEndPoint =
"http://axdbf.axmedis.org:8080/LoaderSaverF/save";
    var AXDBF_user = "axdbuser";
    var AXDBF_passwd = "axmedis";
    var AXDBF_usingftp = true;
    var AXDBF_externalurl = "ftp://axdbf.axmedis.org";
    var AXDBF_internalurl = "";
    var AXDBF_lockEndPoint =
"http://axdbf.axmedis.org:8080/LockUnlockWSF/lockunlock";
     var masterObj = new AxmedisObject();
     masterObj.load(outputDir+"\\"+strip_extension(strip_filename
(filePath))+".axm");
     var AXOID = masterObj.AXOID;
     var tempFileName = "";

if(!masterObj.uploadToDB(AXDBF_saverEndPoint,AXDBF_user,AXDBF_passwd,
AXDBF_usingftp,AXDBF_externalurl,tempFileName,AXDBF_internalurl,AXDBF
_lockEndPoint))
    {
        var error = "--Upload request failure: "+ AXOID;
        print(error);
        return false;
    }
    masterObj.dispose ();
    masterObj = null;
    */

    var start = true;
    var axeptoolUri =
"http://localhost:7780/WebServices/P2PMonitoring";
    var trackerUri = "http://axtrk.axmedis.org:8080/AXTrackv2/";
    var manager = new AXP2PManager(axeptoolUri, trackerUri);
    var Uri = outputDir+"\\"+strip_extension(strip_filename
(filePath))+".axm";
    var pubstatus = manager.publishObject(Uri);
    print ("Publishing to P2P = " + pubstatus);

}
function dirToSingleObjects (dirPath, outputPath, mask)
{
    if (mask == undefined) mask = "*.*";
    allFiles = getFilelist (dirPath, mask)
```

```
    counter = 0;

    for each (singleFile in allFiles)
    {
        counter++;
        print ("Creating a single object for
"+strip_filename(singleFile)+" - "+counter+" / "+allFiles.length);
        print ("-- Output is:
"+outputPath+"\\"+strip_extension(strip_filename(singleFile))+".axm")
;
        fileToObject (singleFile, outputPath);
    }
}]]>
        </JS_Script>
        <JS_Script name="LicenceGenerator">


<![CDATA[/*********************************************************
* A License Generation functions from MB License *
*********************************************************/
// Completed - it works
function createALicense(creatorId,resId,userId)
{
    return
createMotherLicense(creatorId,resId,userId,"mx:play",null);
}
/*************************************************************
*   Grant: function for creating grant for A license    *
*************************************************************/
function createAGrant(resId, right)
{
     var grant = new Grant();
     var rights = new Right();
     rights.setRight(right);
     var res = new Resource();
     res.resourceId = resId;
     res.diType = 1;
     grant.setResource(res);
     grant.setRight(rights);
     return grant;
}
/*********************************************************
* B License Generation functions from MB License *
*********************************************************/
//it works but conditions check is missing
function createB1FromMB1License(lic,userId)
{
   var b1Lic = new License();
   var grant = lic.grantgroup.grants[0];
   var principal = grant.getPrincipal();
   var subgrant = grant.resource.ggResource.grants[0];
   var rIssuer = new Issuer();
   rIssuer.setIssuer(principal.name);
   var rGrantGroup=new GrantGroup();
   var rGrant = new Grant()
   rGrant.setPrincipal(grant.principal);
   rGrant.setResource(subgrant.resource);
   rGrant.setRight(subgrant.right);
```

```
   rGrantGroup.addGrant(rGrant);
   b1Lic.setGrantGroup(rGrantGroup);
   b1Lic.setIssuer(rIssuer);
   return b1Lic;
}
// Missing the Interval or the way to set the
"validityIntervalFloating"
function createB2FromMB2License(lic,userId)
{
   var b2Lic = new License();
   var grant = lic.grantgroup.grants[0];
   var principal = grant.getPrincipal();
   var subgrant = grant.resource.ggResource.grants[0];

   var rIssuer = new Issuer();
   rIssuer.setIssuer(principal.name);
   var rGrantGroup=new GrantGroup();
   var rGrant = new Grant()
   rGrant.setPrincipal(grant.principal);
   rGrant.setResource(subgrant.resource);
   rGrant.setRight(subgrant.right);
   rGrantGroup.addGrant(rGrant);
   b2Lic.setGrantGroup(rGrantGroup);
   b2Lic.setIssuer(rIssuer);
   return b2Lic;
}
// Completed - it works
function createB3FromMB3License(lic,userId)
{
   var b3Lic = new License();
   var grant = lic.grantgroup.grants[0];
   var principal = grant.getPrincipal();
   var subgrant = grant.resource.ggResource.grants[0];

   var rIssuer = new Issuer();
   rIssuer.setIssuer(principal.name);
   var rGrantGroup=new GrantGroup();
   var rGrant = new Grant()
   rGrant.setPrincipal(grant.principal);
   rGrant.setResource(subgrant.resource);
   rGrant.setRight(subgrant.right);
   rGrantGroup.addGrant(rGrant);
   b3Lic.setGrantGroup(rGrantGroup);
   b3Lic.setIssuer(rIssuer);
   return b3Lic;
}
/*************************************************
*           Bx License Generation functions         *
*************************************************/
function createB3Grant(resId, right, notBefore, notAfter)
{
        var grant = new Grant();
        var rights = new Right();
        var timecondition = new Interval();
        var limit = new LNumber();
        var res = new Resource();
        res.resourceId = resId;
        res.diType = 1;
```

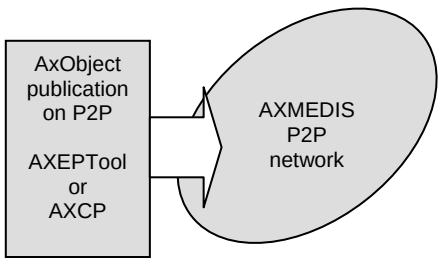
```
        limit.count = 10;
        rights.setRight(right)
        timecondition.notBefore = notBefore;
        timecondition.notAfter = notAfter;
        grant.addCondition(timecondition);
        grant.addCondition(limit);
        grant.setRight(rights);
        grant.setResource(res);
        return grant;
}
function createB2Grant(resId, right, time)
{
        var grant = new Grant();
        var rights = new Right();
        var res = new Resource();
        res.resourceId = resId;
        res.diType = 1;
        rights.setRight(right);
        grant.setResource(res);
        grant.addCondition(timecondition);
        grant.addCondition(limit);
        grant.setRight(right);
        return grant;
}
function createB1Grant(resId, right, amount, currency)
{
     var grant = new Grant();
     var fee = new Fee();
     var rights = new Right();
     rights.setRight(right);
     fee.amount = parseFloat(amount);
     fee.currency = "iso:EUR";
     fee.tipoFee = 1;
     var res = new Resource();
     res.resourceId = resId;
     res.diType = 1;
     grant.setResource(res);
     grant.addCondition(fee);
     grant.setRight(rights);
     return grant;
}
/*************************************************
*     Bx License Generation function from grant     *
*************************************************/
function createBxLicense(distributorId,userId,grant)
{
    var license = new License();
    var grantGroup = new GrantGroup();
    var grantGroup1 = new GrantGroup();
    var issuer = new Issuer();
    var principal = new Principal();

    principal.setName(userId);
    grant.setPrincipal(principal);
    grantGroup.addGrant(grant);
    issuer.setIssuer(distributorId);
    license.setGrantGroup(grantGroup);
    license.setIssuer(issuer);
```

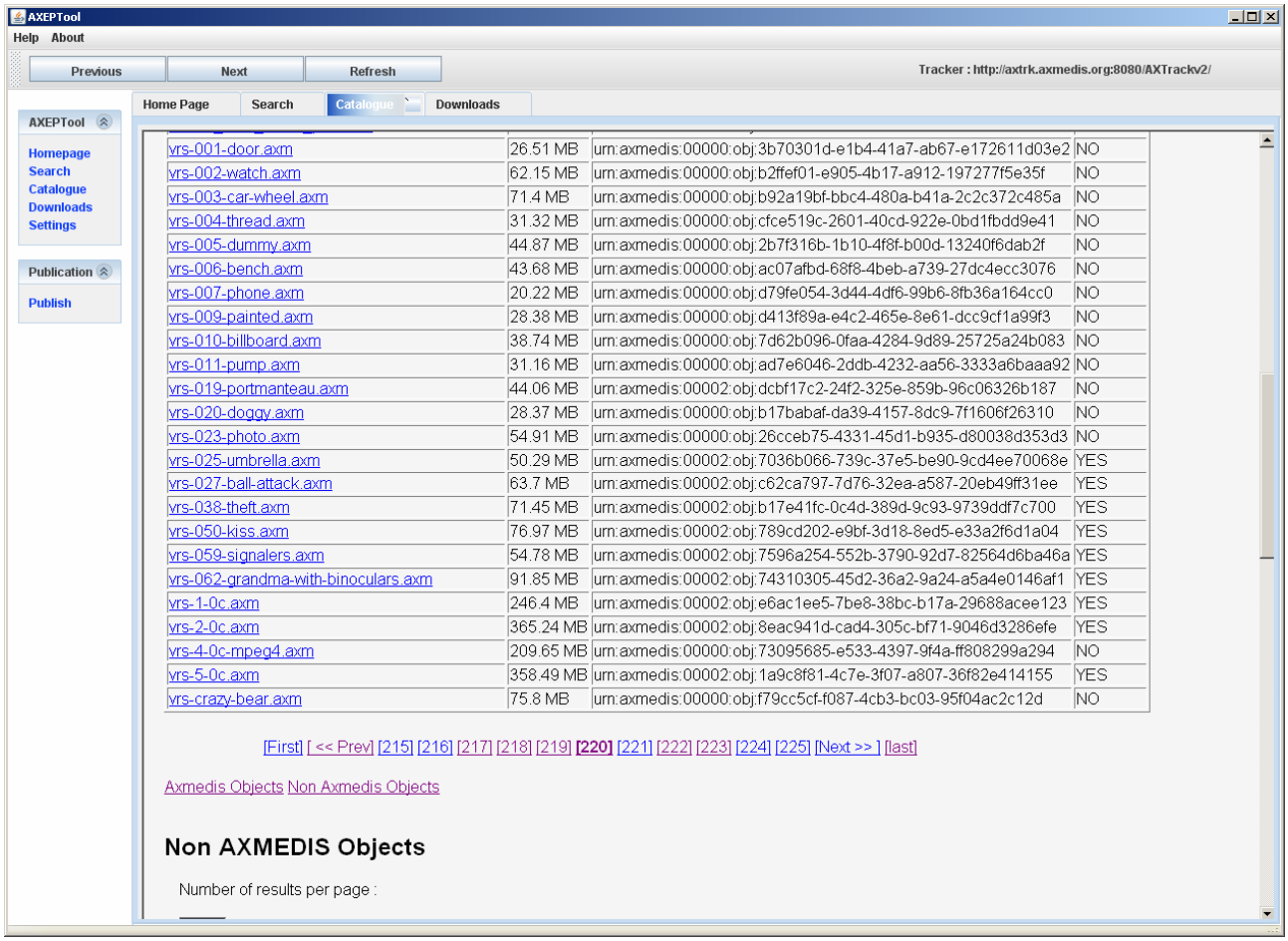
```
    return license;
}
/*****************************************************
*           Mother License Generation function      *
*****************************************************/
function createMotherLicense(creatorId,userId,grant)
{
    var license = new License();
    var grantGroup = new Array(new GrantGroup(),new GrantGroup());
    var grant0 = new Grant();
    var issuer = new Issuer();
    var rights = new Right();
    var principal = new Principal();
    var res = new Array (new Resource(),new Resource());

    //level 1
    grantGroup[1].addGrant(grant);

    //level 0 - grant Group
    res[0].setGrantGroup(grantGroup[1]);
    rights.setRight("r:issue");
    principal.setName(userId);
    grant0.setPrincipal(principal);
    grant0.setResource(res[0]);
    grant0.setRight(rights);
    grantGroup[0].addGrant(grant0);
    //level 0 - issuer
    issuer.setIssuer(creatorId);
    license.setGrantGroup(grantGroup[0]);
    license.setIssuer(issuer);
    return license;
}]]>
        </JS_Script>
      </Rule_Body>
      <Dependencies>
        <Dependency>
          <Plug_In_name>AX_Plugin</Plug_In_name>
          <Version>1.001</Version>
        </Dependency>
      </Dependencies>
    </AXCP_Rule>
  </Definition>
</Rule>
```



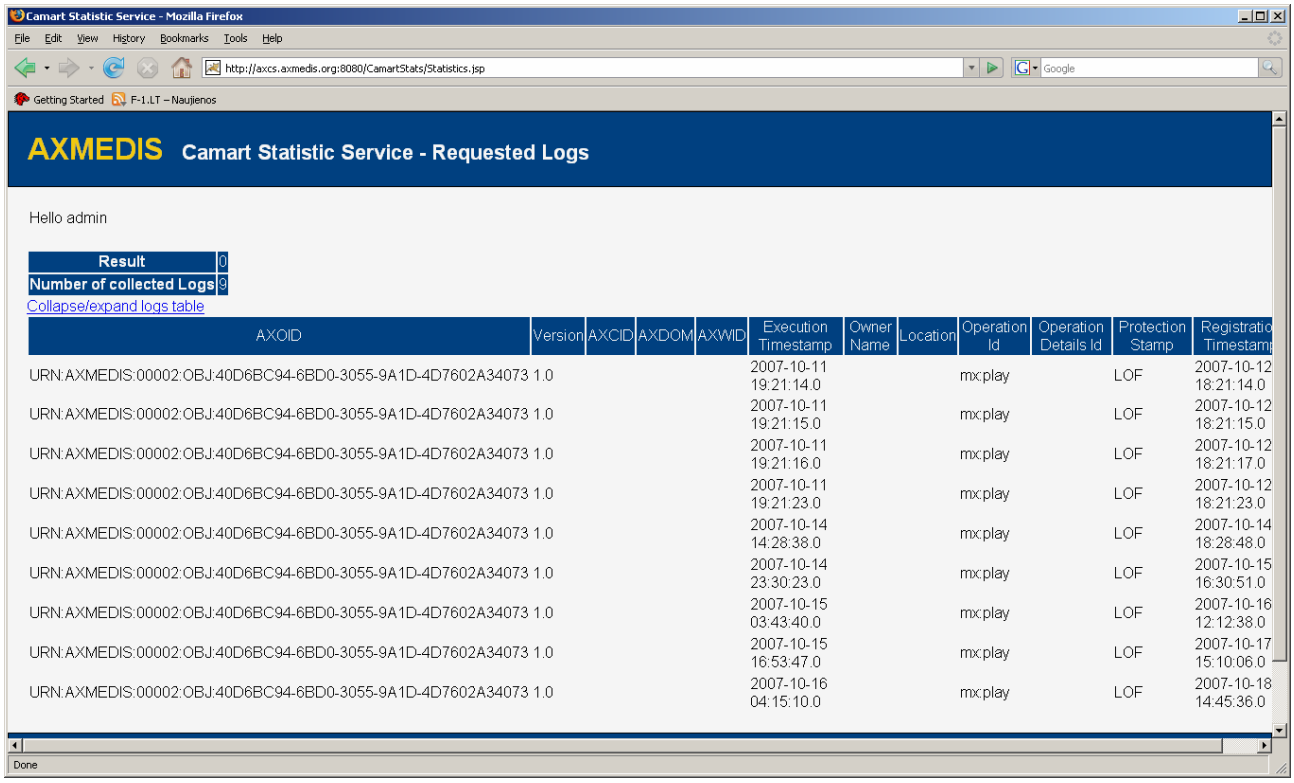
3.3 AxObject publication on P2P

Created AxObjects are published on P2P network by publishObject AXCP rule or manually with AXEPTool





VRS factory produced AxObjects are shared to P2P network with AXEPTool

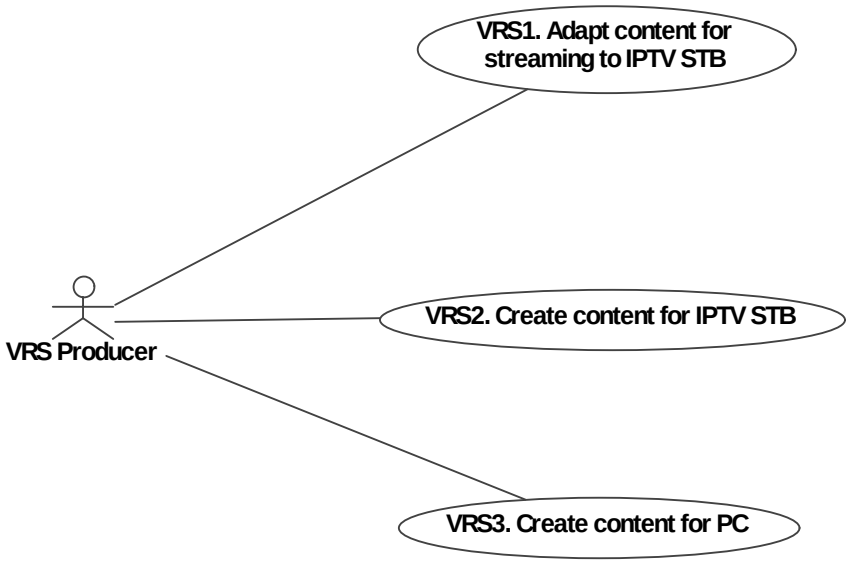


All action logs with AXOIDs are monitored by CAMART tool

4 VRS factory use cases

In this Use Case model VRS Content Producer actions are described. VRS Content Producer can:

- Adapt video content for streaming to IPTV STB;
- Create content for IPTV STB for streaming (TEO demonstrator);
- Create content for PC (Elion demonstrator).



Content creation for both platforms consists of several steps. These steps can be performed automatically using AXCP scripts or manually using AXMEDIS Editor, DRM Editor, AXEPTool, etc. Content adaptation for streaming to IPTV STB must be done manually regarding necessity use of external tools.

4.1 Content adaptation for streaming to IPTV STB

UCId	VRS1.
Use case	Adapt video content for streaming to IPTV STB
Description	VRS Content Producer adapts video content for IPTV STB (TEO demonstrator) manually using external tools (Sony Vegas, Elecard XMuxer)
Actors	VRS Content Producer
Assumptions	VRS Content Producer has video material, which can be used for creating new AXMEDIS object and has tools for MPEG2-TS file formation
Steps	1. Firstly in the process it is necessary to define inserted ad regions, exporting the timecodes to XML file. For this job Sony Vegas external tool is used. Timecode XML file later will be used as metadata in AxObject to define non skippable regions location for PCR of the STB. Also the same external tool - Sony Vegas is used for splitting interleaved broadcast ready media file to video and audio streams according to desired template of STB – video stream MPEG-2 CBR 4Mbps, 720x576, 25 fps and audio stream MPEG-1 224 Kbps. 2. External tool Elecard XMuxer is called to format MPEG-2 TS file from the audio and video streams.
Post-conditions	Video content is prepared in required format for streaming.
Variations	

Asynchronous actions	

4.2 Producer creates AXMEDIS object for streaming to IPTV STB

UCId	VRS2.
Use case	Create content for IPTV STB
Description	VRS Content Producer creates streaming video AXMEDIS object for IPTV STB (TEO demonstrator)
Actors	VRS Content Producer
Assumptions	VRS Content Producer has prepared streaming video content, which can be used for creating new AXMEDIS object and has installed AXMEDIS tools
Steps	1. New AXMEDIS object is created and MPEG-2 TS video resource and metadata are added. (This Use Case is described in "DE2.1.1.2.2 Use Cases and Scenarios, First Update of DE2.1.1b") 2. Executing MPEGTSAdaptation plugin for scrambling MPEG-2 Transport Stream (TS) resource using DVB-CSA algorithm. The MPEGTSAdaptationPlugin encrypt all video and audio channels of stream and leave unencrypted frames containing Program Clock Reference (PCR) which is critical for streaming control. Each scrambled frame of transport stream has to be marked as scrambled according to ISO13818-1 standard. The second task of plug-in is to collect information about all scrambled elementary MPEG-2 TS streams (ES). This task associates each ES with DVB-CSA Control Word (CW) which was used for scrambling the ES. 3. New AXMEDIS object is protected and distribution license is generated (This Use Case is described in "DE2.1.1.2.2 Use Cases and Scenarios, First Update of DE2.1.1b") 4. New AXMEDIS object is published in B2B P2P Axmedis network using AXEPTool (This Use Case is described in "DE2.1.1.2.2 Use Cases and Scenarios, First Update of DE2.1.1b").
Post-conditions	New streaming video AXMEDIS object is created and published in B2B P2P Axmedis network
Variations	
Asynchronous actions	

4.3 VRS Producer creates AXMEDIS object content for PC

UCId	VRS3.
Use case	Create content for PC
Description	VRS Producer creates Axmedis objects for PC platform (Elion demonstrator)
Actors	VRS Producer
Assumptions	VRS Producer has video material, which can be used for creating new AXMEDIS object and has installed AXMEDIS tools
Steps	1. MultimediaAdaptation plugin is executed for resource processing (mpeg-4 for Elion PC). 2. VideoAdaptation plugin is executed for video resizing (640x480, 15fps for Elion PC). 3. New AXMEDIS object is created and video resource and metadata are added. (This Use Case is described in "DE2.1.1.2.2 Use Cases and Scenarios, First Update of DE2.1.1b") 4. New AXMEDIS object is protected and distribution license is generated (This Use Case is described in "DE2.1.1.2.2 Use Cases and Scenarios,

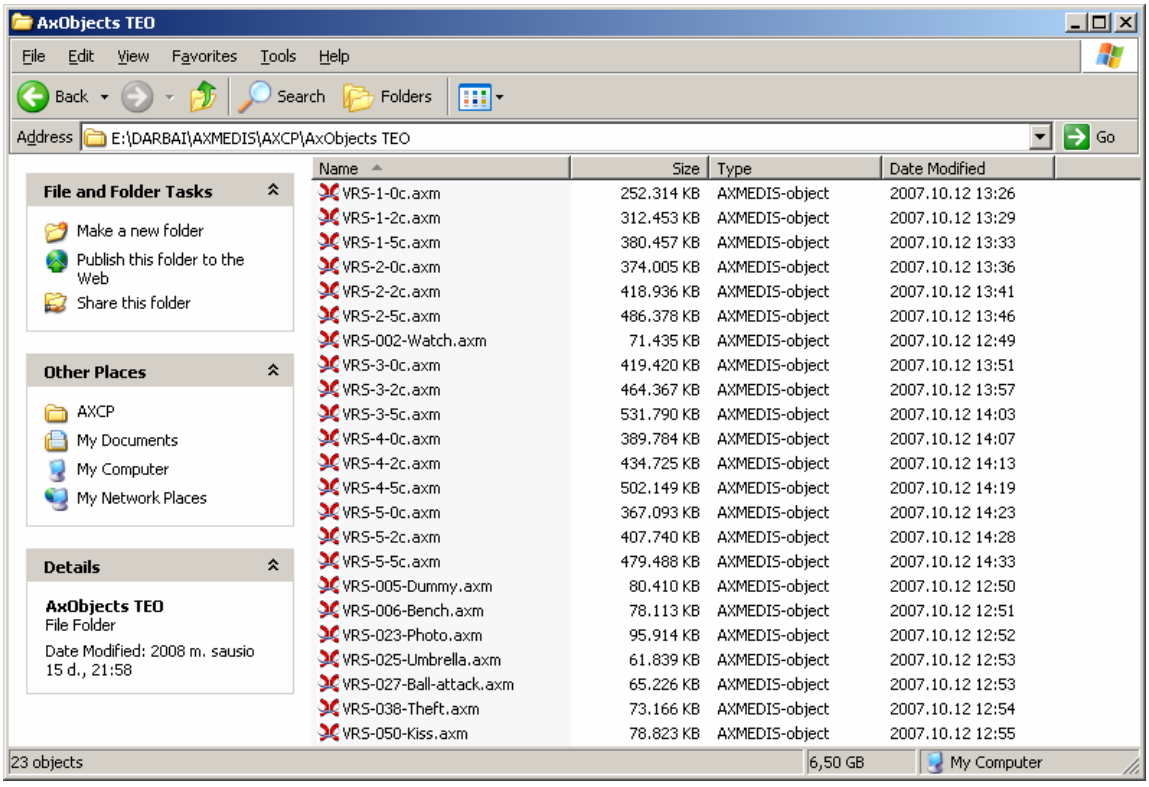
	First Update of DE2.1.1b") 5. New AXMEDIS object is published in B2B P2P Axmedis network using AXEPTool (This Use Case is described in "DE2.1.1.2.2 Use Cases and Scenarios, First Update of DE2.1.1b").
Post-conditions	New AXMEDIS object for PC platform is created and published in B2B P2P Axmedis network
Variations	
Asynchronous actions	

4.4 VRS content processed

All VRS media content resources employed in AXMEDIS are based on VRS own authored TV series "Funny hidden camera":

- collection of video clips from "VRS Kamera" TV program series;
- a number of "thematic" clips - home video, local celebrities, wedding, etc…

According to project objectives, the special compilation of video clips with or without ads was to be produced to test some commercial model for VoD clients of ELTEO demonstrator:

- 5 unique "Home video" blocks with 5 ad inserts;
- 5 unique "Home video" blocks with 2 ad inserts;
- 5 unique "Home video" blocks with no ads;
- 5 unique "Home video" blocks arranged in different manner with 5 ad inserts;
- 

5 VRS demonstrator metrics and specification of content processed

As part of the content processing exercise, the following metrics were measured and produced in order to provide feedback on the effectiveness of AXMEDIS content processing tools:

- Formats produced:
  - MPEG-2 TS – PAL resolution for streaming to STB
  - MPEG-4 – PC resolution (Elion demonstrator)
- Number of content items produced per day - 20 video clips
- Number of content items produced at the same time – 20 video clips
- Average processing lead time from ingestion from source to uploading on P2P of a 600 MB clip for streaming – 10 min
- Average processing lead time from ingestion from source to uploading on P2P of a 10 MB clip for downloading onto PC – 3 min
- time to process one 600 Mb clip:
  - Content defining, timeline editing, ads inserting, elementary streams export -3 min.
  - Transport stream muxing -1 min.
  - AXCP rule AxObject creation and scrambling -3 min
  - AXCP rule license generation and posting to PMS -1 min
  - AXCP rule publishing on P2P network- 2 min.

The automatic content processing with AXCP grid tools enables editor to create content for certain media channel minimum 2-3 times faster than proceeding the same procedures manually. The crossmedia content preparation platform on Axmedis Framework is marking the new basics in future content editing toolls.

6 Issues identified

During the process of using AXMEDIS content processing tools, the following issues have been identified:

Issue	Suggested solution
Long processing times for larger files	To divide content to smaller AxObjects
Offten issue about slow responses from PMS server	Resolve the PMS server availibility 24\7
AXEPTool status is often undefined- status of all the processes „freezing“	More informative GUI
The basic AXCP rules must be standardised for the wider usage in AXCP grid, AXCP Rule Editor has no basic help files for users	Filling „Comment“ can be useful for editors

7 List of objects available for demonstration

The table below represents the list of objects available for demonstration.

No	Filename	URI	AXOID	Title	Subject	Creator	Version	Description	Language	Total no. of resources	Format(s) included	Resolution	Total file size Kb	P2P via AXCEPTool?	DVD for conferences etc.?	via TISCALI PC & P2P?	via ILABS ANSC PDA?	via ILABS mobile?	via ELION STB?	via TEO STB?	via BBC STB?	via EUTELSAT/UNIVLEEDS/MBI?	via TI Mobile?	Must-be protected using AX/OMA	Duration Unlimited/Until 2010?	Can be adapted?	Can be transcoded?	play?	printed?
1	VRS teaser 1.axm	VRS teaser 1.axm		VRS teaser 1	VRS kamera teaser	VRS	1	Short clip from "themes" series No 1	LTU	1	MPEG-2 TS	720x576	13563	Y	N	N	N	N	Y	Y	N	N	N	N	L	Y	Y	Y	N
2	VRS teaser 2.axm	VRS teaser 2.axm		VRS teaser 2	VRS kamera teaser	VRS	1	Short clip from "themes" series No 2	LTU	1	MPEG-2 TS	720x576	13637	Y	N	N	N	N	Y	Y	N	N	N	N	L	Y	Y	Y	N
3	VRS teaser 3.axm	VRS teaser 3.axm		VRS teaser 3	VRS kamera teaser	VRS	1	Short clip from "themes" series No 3	LTU	1	MPEG-2 TS	720x576	17190	Y	N	N	N	N	Y	Y	N	N	N	N	L	Y	Y	Y	N
4	VRS teaser 4.axm	VRS teaser 4.axm		VRS teaser 4	VRS kamera teaser	VRS	1	Short clip from "themes" series No 3	LTU	1	MPEG-2 TS	720x576	15790	Y	N	N	N	N	Y	Y	N	N	N	N	L	Y	Y	Y	N
5	VRS teaser 5.axm	VRS teaser 5.axm		VRS teaser 5	VRS kamera teaser	VRS	1	Short clip from "themes" series No 4	LTU	1	MPEG-2 TS	720x576	12797	Y	N	N	N	N	Y	Y	N	N	N	N	L	Y	Y	Y	N
6	VRS teaser 6.axm	VRS teaser 6.axm		VRS teaser 6	VRS kamera teaser	VRS	1	Short clip from "themes" series No 5	LTU	1	MPEG-2 TS	720x576	17063	Y	N	N	N	N	Y	Y	N	N	N	N	L	Y	Y	Y	N
7	VRS demo no ads.axm	VRS demo no ads.axm		VRS demo no ads	VRS kamera	VRS	1	Collection of clips with no ads	LTU	1	MPEG-2 TS	720x576	325.000	Y	N	N	N	N	Y	Y	N	N	N	N	L	Y	Y	Y	N
8	VRS demo plus 2 ads.axm	VRS demo plus 2 ads.axm		VRS demo plus 2 ads	VRS kamera	VRS	1	Collection of clips with 2 embedded ads	LTU	1	MPEG-2 TS	720x576	393.780	Y	N	N	N	N	Y	Y	N	N	N	N	L	Y	Y	Y	N
9	VRS demo plus 5 ads.axm	VRS demo plus 5 ads.axm		VRS demo plus 5 ads	VRS kamera	VRS	1	Collection of clips with 5 embedded ads	LTU	1	MPEG-2 TS	720x576	492.988	Y	N	N	N	N	Y	Y	N	N	N	N	L	Y	Y	Y	N
10	VRS teaser 1-mpeg4.axm	VRS teaser 1.axm		VRS teaser 1	VRS kamera teaser	VRS	1	Short clip from "themes" series No 1	LTU	1	ffdshow mpeg 4	720x576	2760	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	L	Y	Y	Y	N
11	VRS teaser 2-mpeg4.axm	VRS teaser 2.axm		VRS teaser 2	VRS kamera teaser	VRS	1	Short clip from "themes" series No 2	LTU	1	ffdshow mpeg 4	720x576	2300	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	L	Y	Y	Y	N
12	VRS teaser 3-mpeg4.axm	VRS teaser 3.axm		VRS teaser 3	VRS kamera teaser	VRS	1	Short clip from "themes" series No 3	LTU	1	ffdshow mpeg 4	720x576	2666	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	L	Y	Y	Y	N
13	VRS teaser 4-mpeg4.axm	VRS teaser 4.axm		VRS teaser 4	VRS kamera teaser	VRS	1	Short clip from "themes" series No 4	LTU	1	ffdshow mpeg 4	720x576	3515	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	L	Y	Y	Y	N
14	VRS teaser 5-mpeg4.axm	VRS teaser 5.axm		VRS teaser 5	VRS kamera teaser	VRS	1	Short clip from "themes" series No 5	LTU	1	ffdshow mpeg 4	720x576	2559	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	L	Y	Y	Y	N

No	Filename	URI	AXOID	Title	Subject	Creator	Version	Description	Language	Total no. of resources	Format(s) included	Resolution	Total file size Kb	P2P via AXCEPTool?	DVD for conferences etc.?	via TISCALI PC & P2P?	via ILABS ANSC PDA?	via ILABS mobile?	via ELION STB?	via TEO STB?	via BBC STB?	via EUTELSAT/UNIVLEEDS/MBI	via TI Mobile?	Must-be protected using AX/OMA	Duration Unlimited/Until 2010?	Can be adapted?	Can be transcoded?	play?	printed?
15	VRS teaser 6-mpeg4.axm	VRS teaser 6.axm		VRS teaser 6	VRS kamera teaser	VRS	1	Short clip from "themes" series No 6	LTU	1	ffdshow mpeg 4	720x576	2559	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	L	Y	Y	Y	N
16	VRS-demo-0c-mpeg4.axm	VRS-demo-0c-mpeg4.axm		VRS-demo-0c-mpeg4	VRS kamera	VRS	1	Collection of clips with no ads	LTU	1	ffdshow mpeg 4	720x576	62744	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	L	Y	Y	Y	N
17	VRS-demo-2c-mpeg4.axm	VRS-demo-2c-mpeg4.axm		VRS-demo-2c-mpeg4	VRS kamera	VRS	1	Collection of clips with 2 embedded ads	LTU	1	ffdshow mpeg 4	720x576	78554	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	L	Y	Y	Y	N
18	VRS-demo-5c-mpeg4.axm	VRS-demo-5c-mpeg4.axm		VRS-demo-5c-mpeg4	VRS kamera	VRS	1	Collection of clips with 5 embedded ads	LTU	1	ffdshow mpeg 4	720x576	101789	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	L	Y	Y	Y	N
19	VRS-crazy bear-mpeg4.axm	VRS-crazy bear-mpeg4.axm		VRS-crazy bear-mpeg4	VRS kamera clip	VRS	1	Candid camera clip	LTU	1	ffdshow mpeg 4	720x576	19915	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	L	Y	Y	Y	N
20	VRS-flat tyre-mpeg4.axm	VRS-flat tyre-mpeg4.axm		VRS-flat tyre-mpeg4	VRS kamera clip	VRS	1	Candid camera clip	LTU	1	ffdshow mpeg 4	720x576	31905	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	L	Y	Y	Y	N
21	VRS-heavy suitcase-mpeg4.axm	VRS-heavy suitcase-mpeg4.axm		VRS-heavy suitcase-mpeg4	VRS kamera clip	VRS	1	Candid camera clip	LTU	1	ffdshow mpeg 4	720x576	45114	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	L	Y	Y	Y	N
22	VRS-painted!-mpeg4.axm	VRS-painted!-mpeg4.axm		VRS-painted!-mpeg4	VRS kamera clip	VRS	1	Candid camera clip	LTU	1	ffdshow mpeg 4	720x576	29061	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	L	Y	Y	Y	N
23	VRS-the bench-mpeg4.axm	VRS-the bench-mpeg4.axm		VRS-the bench-mpeg4	VRS kamera clip	VRS	1	Candid camera clip	LTU	1	ffdshow mpeg 4	720x576	44734	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	L	Y	Y	Y	N
24	VRS-the colonel-mpeg4.axm	VRS-the colonel-mpeg4.axm		VRS-the colonel-mpeg4	VRS kamera clip	VRS	1	Candid camera clip	LTU	1	ffdshow mpeg 4	720x576	81628	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	L	Y	Y	Y	N
25	VRS-the doors-mpeg4.axm	VRS-the doors-mpeg4.axm		VRS-the doors-mpeg4	VRS kamera clip	VRS	1	Candid camera clip	LTU	1	ffdshow mpeg 4	720x576	27148	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	L	Y	Y	Y	N
26	VRS-the moving painting-mpeg4.axm	VRS-the moving painting-mpeg4.axm		VRS-the moving painting-mpeg4	VRS kamera clip	VRS	1	Candid camera clip	LTU	1	ffdshow mpeg 4	720x576	39054	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	L	Y	Y	Y	N
27	VRS-the watch-mpeg4.axm	VRS-the watch-mpeg4.axm		VRS-the watch-mpeg4	VRS kamera clip	VRS	1	Candid camera clip	LTU	1	ffdshow mpeg 4	720x576	63640	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	L	Y	Y	Y	N

8 Conclusion

VRS, as a content owner and producer, has completed its workpackage related to content modeling for AXMEDIS and integration of it into ELTEO demonstrator for two operators: TEO and Elion. By setting up AXMEDIS factory with content processing tools at its premises, VRS has fulfilled the task of gaining expertise in content transformation process using AXMEDIS tools integrated with other professional video editing tools, and successfully tested AXMEDIS framework for its multi-channel, rights managed capability.

It must be recognized that the trial content, although of premium quality, original and copyright protected, was not intended and is not to be used for commercial purposes by any third parties other than AXMEDIS partners for demonstration of AXMEDIS.

Although the activity of VRS within AXMEDIS has been limited to technical aspects while working with its own authored content, a number of issues and challenges have been identified for the AXMEDIS Framework or similar structures that relate to other, non technical aspects.

These include:

1) In order for AXMEDIS to become an widely adopted system of digital content distribution, aside from solving the technical issues, a critical mass of system participants – content providers – should be ensured, with compelling, well organized, consumable content. Our experience indicates that technology alone, however functional, will remain technology if content is not there.
2) A transparent system of automatic distribution rights and rules should be developed and embedded into AXMEDIS tools in order to provide ready templates for paperless contract generation. It is our understanding, however, that currently there is a lack of harmonization of digital contract regulations or rules in Europe and further research and understanding of this is needed. In the meantime, we see this as a major impediment for widespread adoption of a system like AXMEDIS.

9 Bibliography

[1] DE12.2.2.1 AXMEDIS-ELTEO requirements

[2] DE12.2.3.1 AXMEDIS-ELTEO specification

[3] DE12.2.4.1 AXMEDIS-ELTEO Component Integration Prototypes and Documentation

[4] DE12.2.5.1 AXMEDIS-ELTEO Demonstrator Set Up

[5 Technical Note No. 6001 Video on Demand Distribution for IPTV Digital Set-Top-Box

[6] Technical Note No. 6501 Elion Content on Demand Distribution for PC

[7] AXMEDIS project website – www.axmedis.org

[8] Demonstrator web pages: general – www.axmedis.org/com

[9] Demonstrator web pages. TEO demonstrator:

http://www.axmedis.org/com/index.php?option=com_content&task=view&id=72&Itemid=51

[10] Demonstrator web pages. Elion demonstrator:

http://www.axmedis.org/com/index.php?option=com_content&task=view&id=75&Itemid=46

10 Glossary

A

AXCP, **AX**medis **C**ontent **P**rocessing

Tools set for automatic content production, adaptation and protection of AXMEDIS object and their publication on a P2P environment.

AXDB, **AX**medis **D**ata**B**ase
Database for storing Axmedis objects

B
B2B, **B**usiness to **B**usiness
Refers to one business communicating with or selling to another.

D
DRM, **D**igital **R**ights **M**anagement
A system for protecting the copyrights of digital content.

I
IPTV, **I**nternet **P**rotocol **T**elevision
A system where a digital television service is delivered by using Internet Protocol over a network infrastructure

M
MPEG TS, **M**oving **P**icture **E**xperts **G**roup **T**ransport **S**tream
Communications protocol for audio, video, and data which is specified in MPEG-2 Part 1, Systems (ISO/IEC standard 13818-1).

P
P2P, Peer-to-**P**eer
Computer network which uses diverse connectivity between participants in a network and the cumulative bandwidth of network participants rather than conventional centralized resources where a relatively low number of servers provide the core value to a service or application

S
STB, **S**et **T**op **B**ox
A device that connects to a television and an external source of signal, turning the signal into content which is then displayed on the television screen

X
XML, e**X**tensible **M**arkup **L**anguage
A metalanguage used to design markups languages.