



Automating Production of Cross Media Content for Multi-channel Distribution

www.AXMEDIS.org

DE9.2.2

Mock up of automating content production and formatting into CMSs of integrators, the prototype

Version: 1.0

Date: 26/04/2006 16:38:00

Responsible: DSI (revised and approved by the coordinator)

Project Number: IST-2-511299

Project Title: AXMEDIS

Deliverable Type: Public

Visible to User Groups: Yes (only the document and the demonstrators)

Visible to Affiliated: Yes (only the document and the demonstrators)

Visible to Public: Yes (only the document and the demonstrators)

Deliverable Number: DE9.1.2

Contractual Date of Delivery: see annex I

Work-Package contributing to the Deliverable: WP 9.2

Nature of the Deliverable: Prototype

Author(s): DSI

Abstract:

This deliverable is related to the mock up of WP9.2 that has been improved in this period by producing a first real prototype.

Keyword List: AXMEDIS content processing, integration with legacy CMS.

Table of Contents

1	EXECUTIVE SUMMARY AND REPORT SCOPE	3
2	INTRODUCTION	3
3	PROTOTYPE DESCRIPTION.....	4
4	BIBLIOGRAPHY	10

1 Executive Summary and Report Scope

This deliverable is the accompanying document for the Prototype having the name common to the deliverable.

2 Introduction

WP9.2 has been improved in this period by producing a first real prototype integrated with AXMEDIS Content Processing Tools.

Content providers, content integrators, aggregator and distributors are using specific technology to compose and format content. These technologies, often coming from one big software solution, could be good in doing some task but poor or missing in other tasks.

For this reason the composition and formatting algorithms and tools conceived in the AXMEDIS framework could be reused inside the proprietary CMS solution to enhance the content workflow. For example a content integrator (content ingestion and composite objects production) can compose its own objects stored in the content databases, using algorithms of AXMEDIS or a content distributor can integrate inside its formatting process such algorithms.

Integrating the AXMEDIS Content Processing Area (see the Figure below) inside the proprietary CMS means having a set of digital content processing tools to:

- efficiently collect the components needed for producing composed objects, using a direct access and query to the database;
- find/produce alternatives/adaptations for the components that present potential distribution problems (too big files);
- structure the components, highlighting the relations among them;
- bind the structure of content and content collection to some presentation and formatting styles;
- format/adapt content for a variety of channels,
- protect digital content according to MPEG-21 models and related AXMEDIS tools, including license production;
- process and produce metadata and managing the estimation of fingerprint and descriptors;
- import from other sources (e.g., existing CMSs) content and metadata to be used for the production process.

The tools provided are also capable to automate the ingestion of content and the related metadata (images, video, audio, documents, etc.), already available in the Content Management Systems (CMS) of the factory.

Moreover the prototype presents the following additional functionalities with respect to the AXMEDIS Content Processing Tool:

- Connection to access in reading information (resources and metadata) from several different databases and protocols by means of Focuseek crawler, see DE3.1.2.2.8;
- Connection to access in reading information from Lobster database and eXact database of ILABS with an extension of Focuseek crawler;
- Connection to access in reading information from XAURA database in MySQL of TISCALI with Focuseek crawler;
- Connection to access in reading and writing information with HP database with XML interface developed for the AXCP in the prototype. HP DMP is well suited for video media, especially in a IPTV or WebTV environment, where the distribution is performed through IP Networks.
- Connection to access in reading and writing information XML database of ANSC with WSDL interface developed for Focuseek, and also with direct connection with WSDL from AXCP script;
- Connection to access in reading and writing information MySQL database of AFI with Focuseek, and also with direct connection with WSDL/FTP from AXCP script;

The following table summarises the technologies used in the CMS of the partners for content query, content access and content upload:

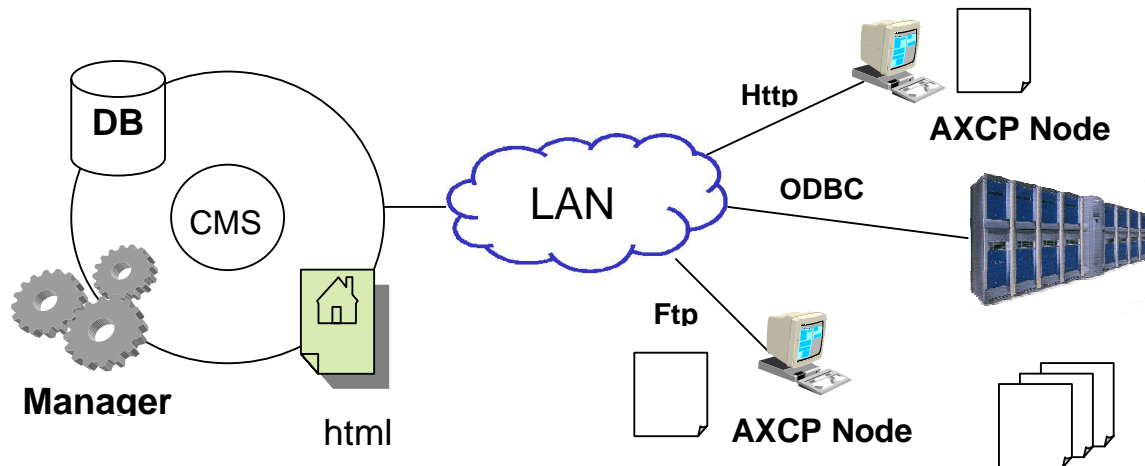
CMS	Content Query		Content Access			Content Upload				
	WS	DB	http	ftp	net fs	upload control		file transfer		
ILABS	X		X			X		X		
TISCALI	(X)	X	X		X	X				X
HP		X				X	X	X	X	
ANSC	X		X			X		?	?	
AFI		X	X	X			X	X		

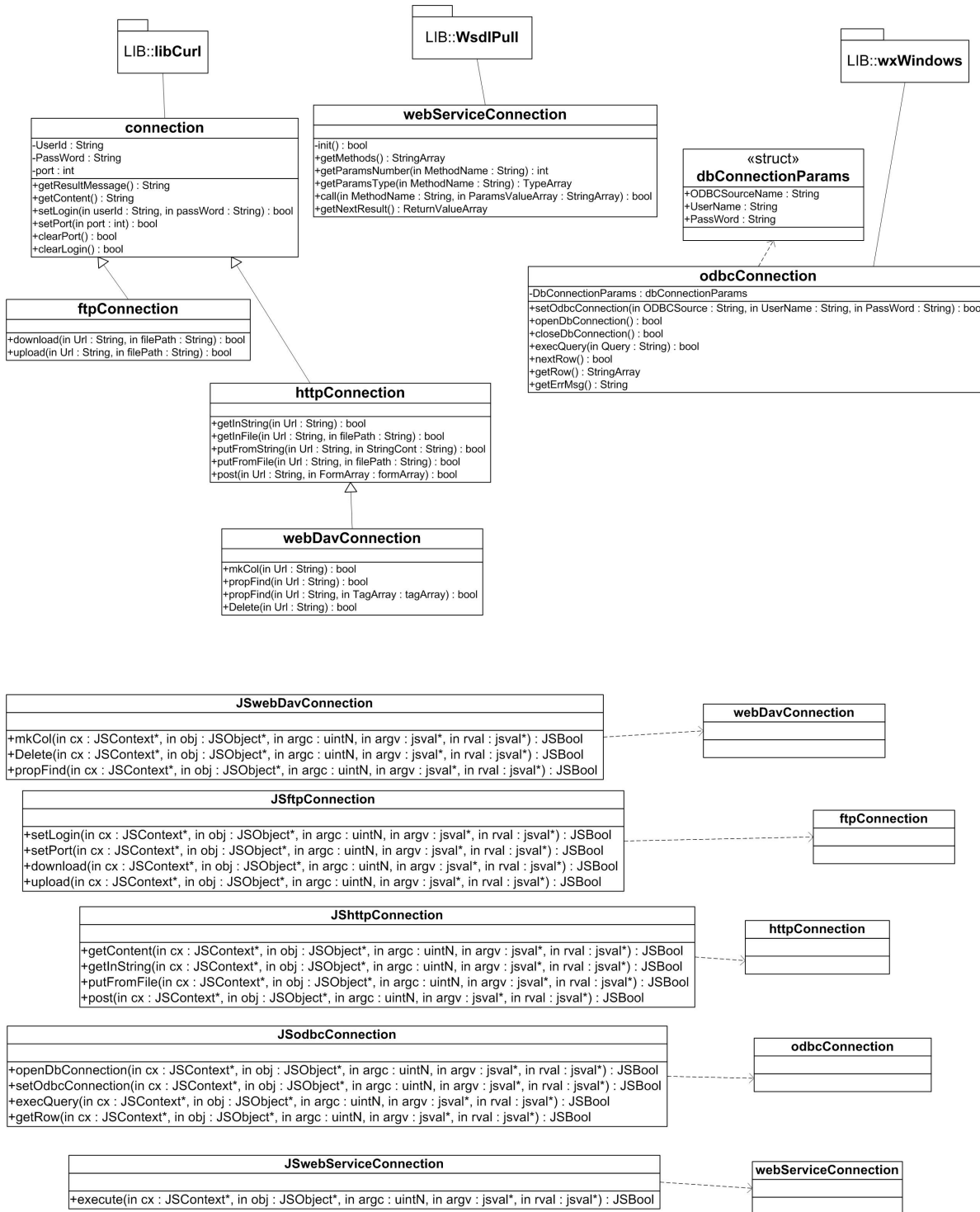
WS= Web Service, DB=Database, net fs=network file system

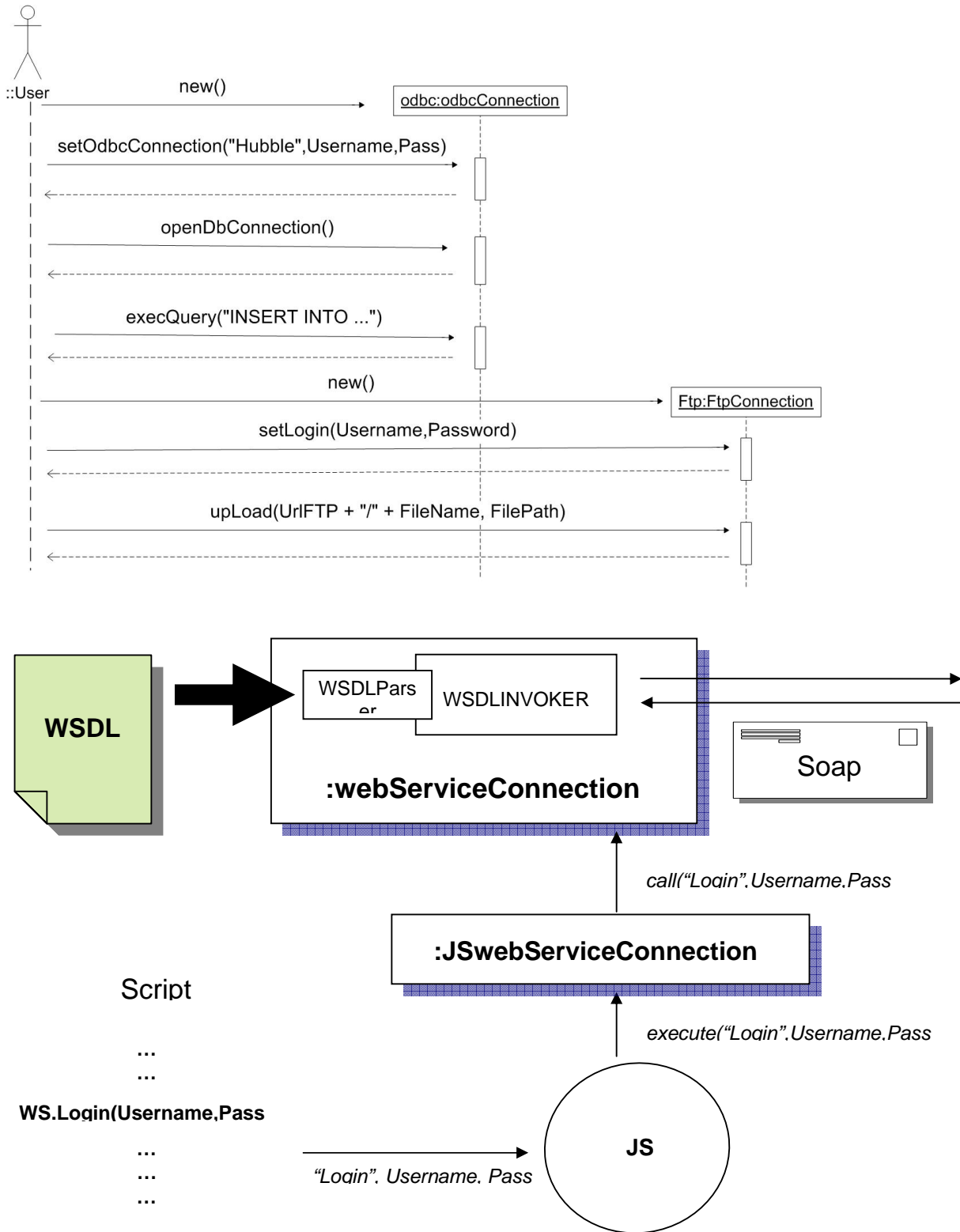
3 Prototype Description

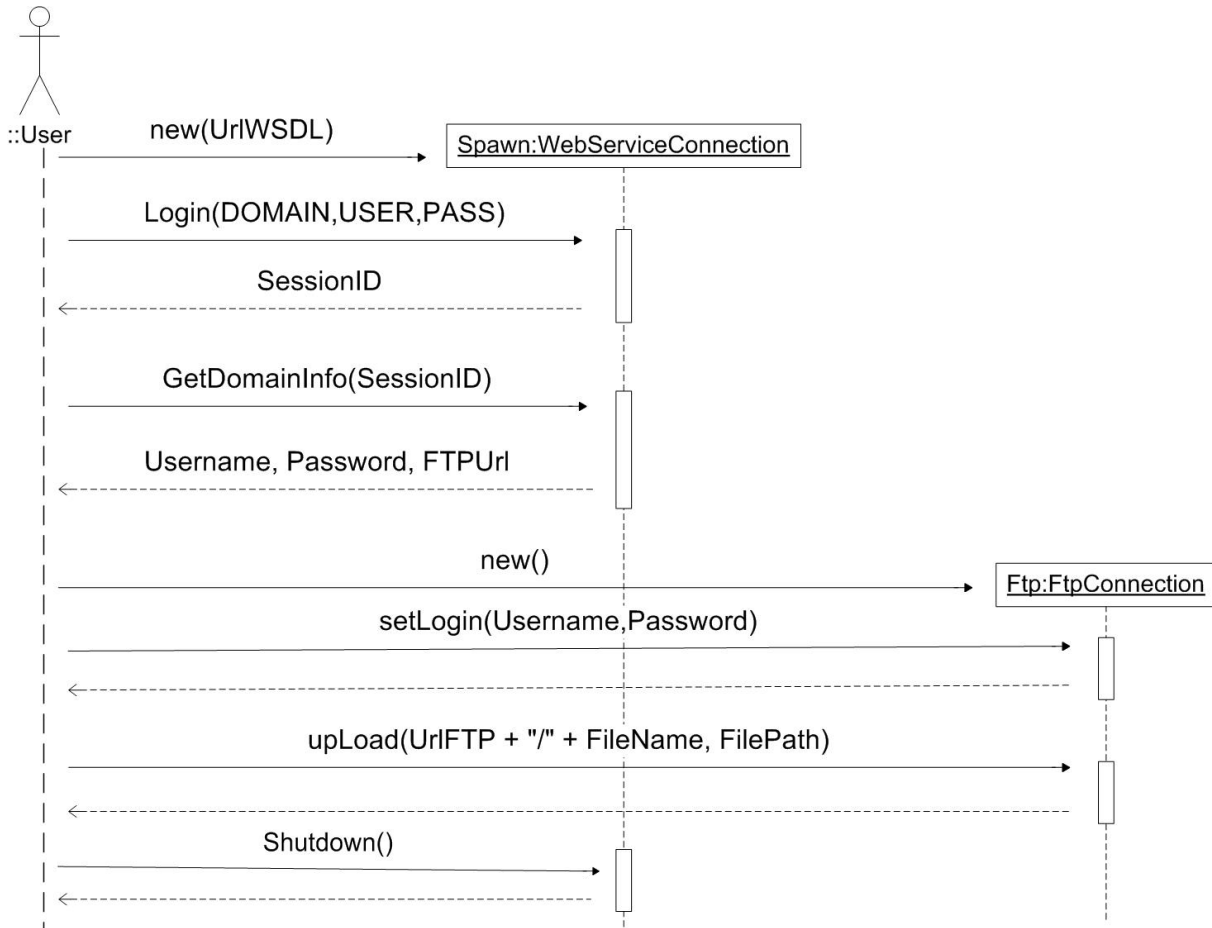
The added technologies in the prototype are:

- Possibility of using into the AXMEDIS script language FTP commands to read and write files;
- Possibility of using into the AXMEDIS script language WSDL actions to access to Web Services in general. It can be used for accessing to database or for other means;
- Possibility of using into the AXMEDIS script language HTTP commands to read and write information according to that protocol;
- Possibility of using into the AXMEDIS script language access to the file system resources, copy, move, delete, etc. of files;
- Possibility of using into the AXMEDIS script language access to the operating system API, such as those related to put in execution processes;
- Possibility of using into the AXMEDIS script language access to XML files for reading and modifying them. Loading/saving XML files in simples and automatic manner.
- Possibility of using into the AXMEDIS script language commands related to compress/ZIP uncompress/unzip files.
- Possibility of using into the AXMEDIS script language SQL commands for ODBC databases. This allows to load and save content in that manner.
- Possibility of using into the AXMEDIS script language to access in reading and writing information of databases via WEBDAV protocol.









In the following, an example of AXMEDIS script for composition and adaptation of AXMEDIS objects is reported. The script shows the use of some classes mentioned above and produces different AXMEDIS objects constituted of a video and at most `maxDoc` documents. The list of videos and documents are provided by two `AxSelection` objects: `videoSel` and `docSel`. They are used to retrieve the array of AXOIDs from the database creating instances of `AxmedisObject`. Each instance is able to retrieve the corresponding video or document. The `maxDoc` parameter is a global variable and it is used to store and fix the maximum number of documents to retrieve from the documents selection and associate with a video resource. The Dublin Core metadata are generated by the `createDublinCoreTitle()` function for each new composite AXMEDIS object associated with the `axObj` instance of `AxmedisObject` by adding an `AxDublinCore` object with the title. The video resource adaptation is performed by the `VideoProcessing` plugin object by means of the `Converting` method: each video is converted in the format specified by the `outFormat` rule parameter. The `VideoProcessing` plugin is a dependency for the script and has to be included in the XML description in order to instantiate the `AxCPPPlugin` object in charge to load the plugin and provide its functions. All new AXMEDIS objects are successively formatted using the `SMILTemplate` and `SMILStyle` objects to find the best template and style to be used as input to the `SMILFormat` object. Finally, the SMIL description is added as resource to the AXMEDIS objects and then they are stored in the AXMEDIS Database by the `uploadToDB()` method of `AxmedisObject` class.

```

function getTitle(axObj) {
    var dc = axObj.getDC();
    return dc.getElement("dc.title");
}

```

```

function createDublinCoreTitle(axObj, title) {
    var dc = new AxDublinCore();
    dc.addElement("dc:title", title);
    axObj.addMetadata(dc);
}
/* Start of script: videos and docs arrays collect results of two selections objects. They provide AXMEDIS Objects to use in the
composition process */

var video_axoids = videoSel.resolve();
var doc_axoids = docSel.resolve();
var template = new SMILTemplate();
var style = new SMILStyle();
var formatter = new SMILFormat();
for(v in video_axoids) {
    var axObj = new AxmedisObject();
    var obj = new AxmedisObject (video_axoids [v]);
    var nDoc = 0;
    var title= GetTitle(obj);
    // extraction of the video resource reference
    var vRes = obj.getContent();
    // adaptation of the resource
    VideoProcessing.Converting(vRes,outFormat, vRes);
    // embed the adapted object
    axObj.addContent(obj);
    for (d in doc_axoids) { // adding maxDoc documents
        axObj.addContent(doc_axoids [d]);
        nDoc++;
        if(nDoc>maxDoc)
            break;
    }
    createDublinCoreTitle(axObj, "Collection of "+title);
    var template = new SMILTemplate(axObj);
    var style = new SMILStyle(axObj);
    // the formatter return an AxResource with the SMIL
    smilRes = formatter.createSMIL(template,style);
    axObj.addContent(smilRes);
    // the Axmedis Object is stored into the DB
    axObj.uploadToDB();
}

```

Example of script to implement a publication

```

var Spawn = new WebServiceConnection (SpawnWsdlUrl) ;

var SessionID = Spawn.Login (DOMAIN,USER,PASS) ;

var GetDomainInfoResponse = Spawn.GetDomainInfo (SessionID) ;

var Parser = new XMLparser() ;
Parser.Set (GetDomainInfoResponse) ;
var UserName = Parser.GetValue ("UserName") ;
var PassWord = Parser.GetValue ("PassWord") ;
var FTPDestinationUrl = Parser.GetValue ("Url") ;

var FileName = Spawn.InsertPackageRequest (SessionID,PackID,ObjType) ;

var Zipper = new ZipArchiver() ;
var PIFPath = "C:/H725PIF.zip";
Zipper.createNewZipArchive (PIFPath) ;
Zipper.addFileToZip ("C:/manifest.xml") ;
Zipper.addFileToZip ("C:/H725.bin") ;

var Ftp = new FtpConnection() ;
Ftp.setLogin (UserName, PassWord) ;
Ftp.upload (FTPDestinationUrl+" / "+FileName, PIFPath) ;

```



```
Spawn.NotifyUploadDone();  
Spawn.Shutdown();
```

Script to get a document:

```
var Spawn = new WebServiceConnection(SeahorseWsdlUrl);  
  
var SessionID = Spawn.Login(DOMAIN,USER,PASS);  
  
var GetDomainInfoResponse = Spawn.GetDomainInfo(SessionID);  
  
var Parser = new XMLparser();  
Parser.Set(GetDomainInfoResponse);  
var DomainUrl = Parser.GetValue("domainurl");  
var CourseBasePath = Parser.GetValue("coursebasepath");  
var FolderName = Spawn.FindPackage(SessionID,Query);  
  
var Url = DomainUrl + "/" + CourseBasePath + "/" + FolderName;  
  
var Http = new HttpConnection();  
Http.getToString(Url+"/manifest.xml");  
  
var Content = Http.getContent();  
  
Spawn.Shutdown();
```

Script to make a publication:

```
var odbc = new OdbcConnection();  
odbc.setOdbcConnection("Hubble",UserName,PassWord);  
odbc.openDbConnection();  
  
odbc.execQuery("INSERT INTO DocumentsResources VALUES  
( 'unibo199Bltr', 'Xeron199B', 'Tiziano  
Casella', 'http://hubble.astrolabs.it/xeron199B.pdf' )");  
  
odbc.closeDbConnection();  
  
var ftp = new FtpConnection();  
ftp.setLogin(UserName,PassWord);  
ftp.upload\("ftp://hubble.astrolabs.it",DestinationPath\);
```

Script to get a document:

```
var odbc = new OdbcConnection();  
odbc.setOdbcConnection("Hubble",UserName,PassWord);  
odbc.openDbConnection();  
  
odbc.execQuery("SELECT Url FROM DocumentsResources WHERE  
DocID='unipi9D12Cdar'");  
  
var Url = odbc.getRow();  
  
odbc.closeDbConnection();  
  
var http = new HttpConnection();
```

```
http.getInString(URL);  
http.getInFile(Uri,FilePath);
```

4 Bibliography

- DE 9.2.1 specification of automating content production and formatting into CMSs of integrators
- DE3-1-2-2-6 Specification of AXMEDIS Content Processing
- <http://curl.haxx.se/libcurl/>
- <http://wsdlpull.sourceforge.net/>
- <http://www.wxwidgets.org/>