“VCASMO Presentation Engine API” Specification

by VCASMO Chief Developer, Luar Yen

draft version, 19th December, 2007

1st revision, 26th October, 2011

2nd revision, 22nd November, 2012

3rd revision, 26th November, 2014

Overview

VCASMO Presentation Engine API (aka VMAPI) is designed for business client (aka Client) to leverage *VCASMO simple and easy to use workflow* on create presentation, and integrate to the existing system with little manpower's (aka developer) programming effort needed.

Workflow on Create Presentation

Core workflow

Upload asset → Choose asset and arrange slide sequence to video → Watch the presentation.

Related workflow

Edit slide sequence, insert new slide/replace video, add/edit subtitle.

Server Requirement

Access VMAPI

Windows/Linux, capable to run a server side language (ASP/ASP.Net/PHP/ColdFusion/Ruby on Rail), which can make REST call. Optional: Database*

** Existing website may need database, but VMAPI part can function without touching any database.*

API Authentication

To use VMAPI, developer must be member of VCASMO.com. An unique **developer key** will be issued based on developer loginID in VCASMO. It is used for server-side communication with VMAPI, so this key is important and must be keep in secret.

Developer has to generate **domain key**. Each developer can generate several domain keys for the test server and production server to use. It is used in browser/client level (possible to view by public). VMAPI is based on the domain key to match the return domain in VMAPI system, in order to return the result or files. Therefore, this authentication and call back mechanism, VMAPI is secured to use without worry of unauthorized party is using the stolen domain key.

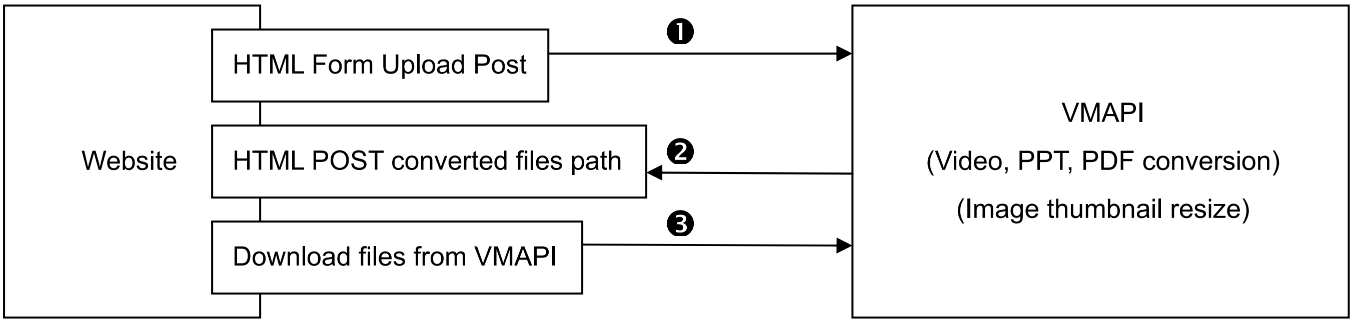
Website Architecture Work with VMAPI

VMAPI need to send back files to your server in one occasions: sequence data file of presentation. Developer have to prepare for VMAPI send back files through HTML POST.

For HTML POST, developer can check IP address of VMAPI to prevent invalid submission. Developer can get an approved IP addresses list of VMAPI through *vcasmo.auth.serverip*.

Database-less workflow

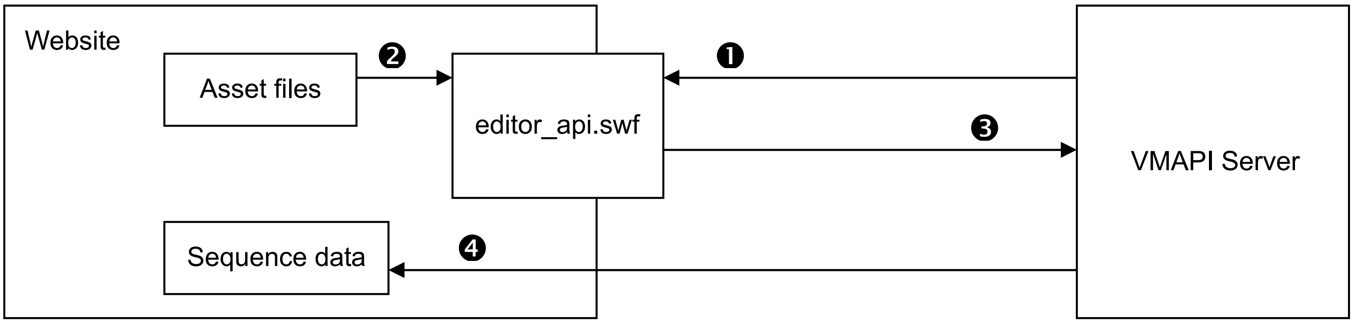
Upload asset



Website
HTML Form Upload Post
HTML POST converted files path
Download files from VMAPI
❶
❷
❸
VMAPI
(Video, PPT, PDF conversion)
(Image thumbnail resize)

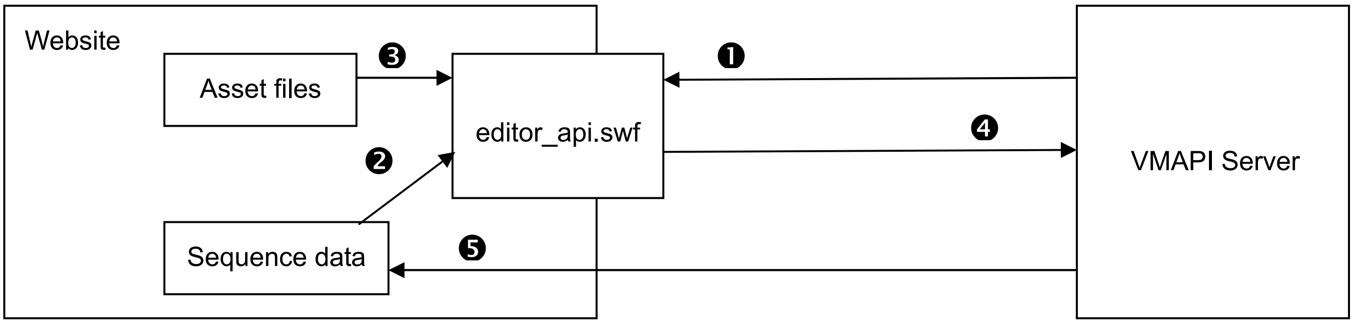
1. User can upload the file directly to VMAPI server in HTML POST. VMAPI will redirect the uploaded result page to your server, therefore, the whole upload process is seamless integrated with client existing website. You can also let user upload the file to your server first (if you want to keep the raw file copy), then send the file to our server by HTML POST.
2. When conversion is finished, VMAPI will inform you and you can download the files to your server. Converted files will be kept in VMAPI for 3 days only, after that will be deleted permanently.

Choose asset, create new presentation



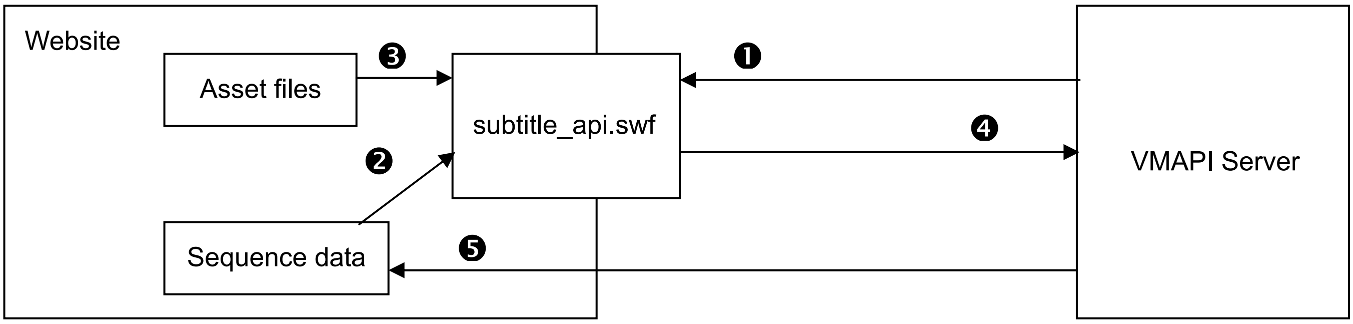
1. editor_api.swf is hosted in VMAPI server, for make sure latest version and authorize proper usage based on domain key.
2. editor_api.swf load in the asset file. It is a XML format response which store the asset full path and additional user identity information.
3. When slide sequence is done, editor_api.swf will ask VMAPI server to generate 2 sequence data files, send back to your server with additional user identity information.
4. Sequence data files store the slide cue point time with the video, it will be used for presentation playback.

Edit presentation / Insert extra asset to presentation



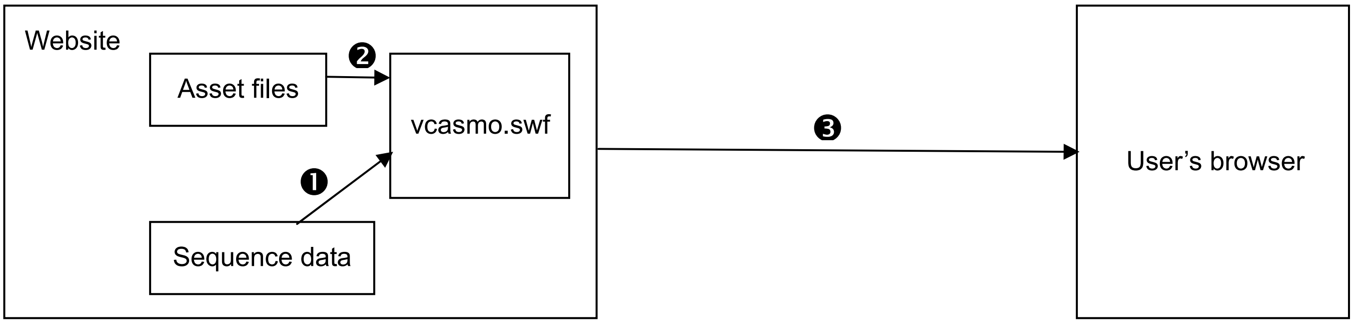
1. editor_api.swf is hosted in VMAPI server, for make sure latest version and authorize proper usage based on domain key.
2. editor_api.swf load sequence data file for display the existing slide arrangement.
3. editor_api.swf load in the asset file.
4. When slide sequence is done, editor_api.swf will ask VMAPI server to generate 2 new sequence data files, send back to your server and overwrite old one.
5. Sequence data file store the slide cue point time with the video, it will be used for presentation playback.

Create / Edit subtitle for presentation



1. subtitle_api.swf is hosted in VMAPI server, for make sure latest version and authorize proper usage based on domain key.
2. subtitle_api.swf load sequence data file for display the existing slide arrangement or existing subtitle data if have.
3. subtitle_api.swf load in the asset file.
4. When subtitle is done, subtitle_api.swf will ask VMAPI server to generate 2 new sequence data files, send back to your server.
5. Sequence data file store the slide cue point time with the video, it will be used for presentation playback.

Presentation playback (non-server deployment option)



Presentation playback is just like hosting offline version of VCASMO presentation in your server (Optional: to enabled video seeking, client have to install MP4 pseudo streaming* in the web server). It is no longer nesscery to use any service from VMAPI. Therefore, if the client stop subscrible VMAPI service, all the existing presentations are still working. On the other hand, client cannot create new presentation or edit existing presentation anymore.

* VMAPI recommend H264 Streaming Module http://h264.code-shop.com/trac

API Documentation – Getting Start

Register domain in VMAPI Admin Panel

Developer has to register the following information in VMAPI Admin Panel (https://api.vcasmo.com/my_login.php) for file send back. For example:

For HTML POST send back

Base domain path: http://wwww.domain.com/vcasmo/

Each time, developer specified call *sendback path*, he has to state the file name only, then VMAPI will combine it with base domain to issue the call.

Example 1: callback: *uploaddone.php*. VMAPI will call:
http://wwww.domain.com/vcasmo/uploaddone.php

Example 2: callback: *file/save.asp*. VMAPI will call:
http://wwww.domain.com/vcasmo/file/save.asp

All VMAPI callback uses POST method.

Prepare crossdomain.xml for Flash

The sequence editor and subtitle editor are developed in Flash and hosted in VMAPI server. In order to let the Flash can receive/send data to your server-side program, you have to add crossdomain.xml in your root of web server (http://www.domain.com/crossdomain.xml) with the following content inside the XML file:

```
<?xml version="1.0" encoding="UTF-8"?>
<cross-domain-policy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.adobe.com/xml/schemas/PolicyFile.xsd">
    <allow-access-from domain="*.vcasmo.com" secure="false" />
    <allow-access-from domain="d1ap128wkmh29n.cloudfront.net" secure="false" />
    <allow-access-from domain="static1-luarsproduction.netdna-ssl.com" secure="false" />
    <site-control permitted-cross-domain-policies="master-only"/>
</cross-domain-policy>
```

Prepare CORS for HTML5

Presentation player HTML5 version will load sequence data XML file from your server, since browser has cross domain protection against one domain load data from another domain, you have to configure web server response CORS header. For detail, please check here: http://enable-cors.org/

```
Access-Control-Allow-Origin: http://static.vcasmo.com https://d1ap128wkmh29n.cloudfront.net https://
static1-luarsproduction.netdna-ssl.com
```

REST Response Format

REST is the simplest response format to use - it's a simple XML block.

A method call returns this:

```
<?xml version="1.0" encoding="utf-8" ?>
<rsp stat="ok">
    [xml-payload-here]
</rsp>
```

If an error occurs, the following is returned:

```
<?xml version="1.0" encoding="utf-8" ?>
<rsp stat="fail">
    <err method=”[method-name]” code="[error-code]" msg="[error-message]" />
</rsp>
```

If the API provide non-XML format response, when error occurs, it will return as the following:

HTML / Flash

```
stat=fail&method=[method-name]&code=[error-code]&msg=[error-message]
```

Comma string

No return

JSON

```
{
  "methodName": "[method-name]",
  "errorCode": [error-code],
  "errorMessage": "[error-message]"
}
```

API Namespace to GET Path

In this document, all API are described as namespace format, actually query path is GET method as following:

vcasmo.xxx.yyy = http://api.vcasmo.com/xxx/yyy or https://api.vcasmo.com/xxx/yyy

Example

checking the conversion process of file, *vcasmo.asset.status*:

http://api.vcasmo.com/asset/status/authkey=a1B2c3D4eF&queueid=12345

API for Asset Upload and Conversion

vcasmo.auth.serverip

Get the VMAPI server IP address for valid file send back is from a valid VMAPI server.

Arguments

format (optional)

1=XML, 2/3=Comma string, 4=JSON, default: 1.

Example

http://api.vcasmo.com/auth/serverip?format=1

Response

XML

```
<?xml version="1.0" encoding="utf-8" ?>
<rsp stat="ok">
    <iplist total="3">
        <ip value="123.456.7.7" />
        <ip value="123.456.7.8" />
        <ip value="123.456.7.9" />
    </iplist>
</rsp>
```

Comma string (no space)

```
123.456.7.7,123.456.7.8,123.456.7.9
```

JSON

```
{
  "iplist": {
    "total": 3,
    "ip": [
      "123.456.7.7",
      "123.456.7.8",
      "123.456.7.9"
    ]
  }
}
```

Error codes

105: Service currently unavailable.

vcasmo.asset.upload

Upload asset to VMAPI server.

Step 1

When you create a HTML form for end user to upload file, please include this JavaScript file in between <head></head>

```
<script type="text/javascript"
src="https://api.vcasmo.com/js_upload.php?[arguments]"></script>
```

Arguments

domainkey (required)

Your domain key.

filedata (required)

File upload to VMAPI

sendback_path (required, maxlength:200)

When file is converted, VMAPI will call this sendback_path and let you download the converted file.

return_method (optional)

When a file is uploaded, return result through what method. 1=XML, 2=HTML, 3=Flash, default: 2.

return_success_path (required when *return_method=2*)

When a file is uploaded successful, return to a page in your server with response argument in URL query string.

return_fail_path (required when *return_method=2*)

When a file upload is failed, return to a page in your server with response argument in URL query string.

userid (optional, maxlength:100)

Your own membership system unique user ID, it helps you to distinguish send back file belong to which user.

custom1 (optional, maxlength:100)

Further custom parameter developer may needed when VMAPI send back file.

custom2 (optional, maxlength:100)

Further custom parameter developer may needed when VMAPI send back file.

Example

```
<script type="text/javascript" src="https://api.vcasmo.com/js_upload.php?
domainkey=12345&sendback_path=uploaddone.php&return_success_path=uploaddone.php&userid=lu
ar&custom1=123&custom2=abc"></script>
```

Step 2

Paste the following code for HTML form, do not include other form value / hidden value

```
<form method="POST" action="" enctype="multipart/form-data" onsubmit="return
vcasmoUpload(this)">
  <input type="hidden" name="AWSAccessKeyId" value="" />
  <input type="hidden" name="key" value="" />
  <input type="hidden" name="policy" value="" />
  <input type="hidden" name="signature" value="" />
  <input type="hidden" name="success_action_redirect" value="" />
  <input type="hidden" name="x-amz-storage-class" value="" />
  <input type="hidden" name="acl" value="" />
  <input name="file" type="file">
  <input type="submit" value="Send">
</form>
```

Response

XML

```
<?xml version="1.0" encoding="utf-8" ?>
<rsp stat="ok">
    <upload oldfilename="mypresentation.ppt" filename="abcDEF123" folder="200712201530"
        fileext="ppt" rawfilesize="1048576" filetype="ppt" queueid="123"
        userid="luar" custom1="123" custom2="pro" />
</rsp>
```

Flash

```
oldfilename=mypresentation.ppt&filename=abcDEF123&folder=200712201530&fileext=ppt&rawfile
size=1048576&filetype=ppt&queueid=123&userid=luar&custom1=123&custom2=pro
```

oldfilename

Original uploaded filename.

filename

New filename return by VMAPI.

folder

Folder under userid folder in VMAPI (internal use)

fileext

Original uploaded file extension.

rawfilesize

orginal uploaded file size, unit: byte.

filetype

File type symbol used in VMAPI, mp4=video, jpg=image (included gif, png, jpg, jpeg), mp3=MP3, ppt=PowerPoint, pdf=PDF, swf=Flash.

queueid

Queue list ID in VMAPI conversion process.

userid, custom1, custom2

Just return what VMAPI received back to your server.

Error codes

100: Invalid developer key.

101: Invalid domain Key.

105: Service currently unavailable.

200: File size is too large.

201: File size is zero byte.

202: Invalid file size format.

vcasmo.asset.sendback (Not API)

When file is converted, VMAPI will send back the file through HTML POST. VMAPI will POST to *sendback_path* with the following parameters:

filelist (array of string)

Converted files download full path (VMAPI send back as filelist[], most of server-side processing language will receive it as filelist and convert it into an array)

rawfilepath (string)

Original file uploaded by user

isready (integer)

2=converted, 3=convert failed (If conversion process is failed, filelist[] is empty).

totalslide (integer, optional)

Total number of slides in a PowerPoint or PDF.

duration (integer, optional)

Duration of video / mp3, unit: second.

videoseqno (integer, optional)

Total number of video file pieces for HTTP Live Streaming(HLS)

filename (8 characters string)

Uploaded file name.

filetype (3 characters string)

File type symbol used in VMAPI, mp4=video, jpg=image, mp3=MP3, ppt=PowerPoint, pdf=PDF, swf=Flash

queueid (integer, required)

Queue list ID in VMAPI conversion process.

userid, custom1, custom2 (optional)

Just return what VMAPI received back to your server.

vcasmo.asset.status

Checking the conversion process of file.

Arguments

authkey (required)

Your developer key.

queueid (required)

Queue list ID assigned by VMAPI received from *vcasmo.asset.upload*.

Example

http://api.vcasmo.com/asset/status/authkey=a1B2c3D4eF&queueid=12345

Response

XML

```
<?xml version="1.0" encoding="utf-8" ?>
<rsp stat="ok">
    <status isready="2" />
</rsp>
```

isready

0=not convert, 1=converting, 2=no such record / conversion is finished (When file is converted successfully or failed, VMAPI will clear the record after vcasmo.asset.sendback)

Error codes

100: Invalid developer key.

105: Service currently unavailable.

vcasmo.asset.cancelconvert

Conversion queue of a file can be cancelled only when the process is not started. That is *vcasmo.asset.status* return *isready=0*.

Arguments

authkey (required)

Your developer key.

queueid (required)

Queue list ID assigned by VMAPI received from *vcasmo.asset.upload*.

Example

http://api.vcasmo.com/asset/cancelconvert/authkey=a1B2c3D4eF&queueid=12345

Response

XML

```
<?xml version="1.0" encoding="utf-8" ?>
<rsp stat="ok">
    <status cancel="0" />
</rsp>
```

cancel

0=Conversion queue of a file is cancelled, 1=file is converting, cannot cancel, 2=no such record / conversion is finished (When file is converted successfully or failed, VMAPI will clear the record after vcasmo.asset.sendback).

Error codes

100: Invalid developer key.

105: Service currently unavailable.

File Naming Rule and Thumbnail Used in VMAPI

All uploaded file, VMAPI will rename with 8 random characters.

Video

Converted file: filename.mp4

Thumbnail: filename_l.jpg (640x480), filename_m.jpg (411x308), filename_k.jpg (240x180), filename_s.jpg (80x60)

MP3

No need convert: filename.mp3

Flash

No need convert: filename.swf

Image

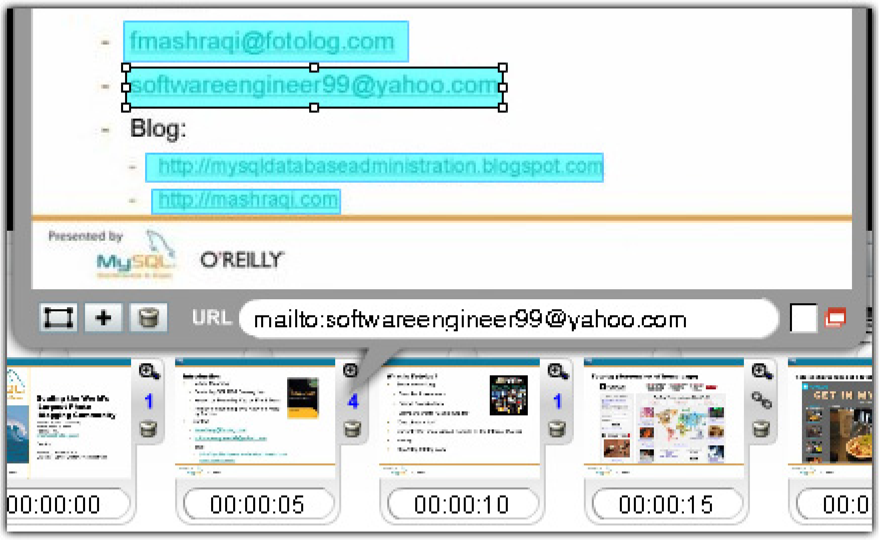
Resized file: filename_l.ext (640x480), filename_m.ext (411x308), filename_k.ext (240x180), filename_s.ext (80x60) (ext=jpg / jpeg / gif / png, portrait image will resize to keep within the height limit)

PowerPoint / PDF

Converted file: filename_x_l.jpg (640x480), filename_x_m.jpg (411x308), filename_x_k.jpg (240x180), filename_x_s.jpg (80x60) (x=slideno, start from 1)

API for Pass Chosen Asset to Editor

Sequence editor (editor_api.swf, aka editor) is an easy to use editor for arrange slide sequence with the video to create presentation in VCAMSO. Apart from arrange slide sequence, it can also allow user to add clickable link on the slide. So we encourage developers to include the editor directly in your own application without re-invent the wheel.



URL mailto:softwareengineer99@yahoo.com

To include the editor directly to your application, using the following HTML code:

```
<div class="vm_embed_editor_div" style="width:100%;height:100%;overflow:hidden;"><iframe
class="vm_embed_editor_iframe" src="//api.vcasmo.com/embed_api_editor.htm?
domainkey={arguments}&assetfile_path={arguments}&
sendback_path={arguments}&return_success_path={arguments}" allowtransparency="true"
width="100%" height="100%" seamless="seamless" scrolling="no" frameborder="0"
marginwidth="0" marginheight="0" allowfullscreen mozallowfullscreen webkitallowfullscreen
oallowfullscreen msallowfullscreen></iframe></div>
```

You should prepare the own user interface for users to choose asset. Each presentation only allow one video or mp3 file with unlimited image / PowerPoint / PDF files. You should check the chosen asset meet this requirement before passing to the editor. So, the most challenge part is how to pass the chosen asset path into the editor.

Pass through XML file (require server-side program)

This is the same workflow currently used in VCASMO.com. Suppose asset data is stored in database, you can pass the chosen assetid as comma string and the query file path to editor.

Arguments to Editor in HTML

domainkey (required)

Your domain key.

assetfile_path (required)

Server-side program for editor to query the asset file path to the program (please provides in full path).

sendback_path (required)

When user click save, the editor send the data to VMAPI. If the presentation is successfully created, editor will send to this path through HTML POST includes: ts, userid, custom1, custom2, too. VMAPI uses domainkey to combine with sendback_path as full path in VMAPI server, so please provides partial path here.

return_success_path (required)

When user click save, the editor send the data to VMAPI. If the presentation is successfully created, editor will redirect the current page to a success page (please provides in full path) with response argument in URL query string includes: ts, userid, custom1, custom2.

return_error_path (optional)

If any error happen on saving, it will redirect to an error page (please provides in full path) with response argument in URL query string includes: userid, custom1, custom2, *msg*.

msg: extra error information from editor

msg=saveFail	VMAPI license expired, cannot save.
msg=assetfile_path	Cannot load asset file.

img_size_suffix (optional)

If your image file naming is not follow VMAPI rule, you can set *img_size_suffix="."*, editor will load the image as *filename+img_size_suffix+fileext*. For filetype=ppt/pdf, even you use own naming rule, editor still load each slide based on: *filename+_x+img_size_suffix+fileext*, (x=slideno, start from 1). So if your slide sequence naming rule is different, you can pass as separate images (filetype=jpg).

userid, custom1, custom2 (optional)

Just return what VMAPI received back to your server.

Example

These arguments have to put it as URL-encoded variables in HTML code passing into editor.

```
<div class="vm_embed_editor_div" style="width:100%;height:100%;overflow:hidden;"><iframe
class="vm_embed_editor_iframe" src="//api.vcasmo.com/embed_api_editor.htm?
domainkey=g5H6i7J8k&assetfile_path=http://www.yourdomain.com/getAsset.php?sessid=123&
sendback_path=saveSeqData.php&return_success_path=step2.php" allowtransparency="true"
width="100%" height="100%" seamless="seamless" scrolling="no" frameborder="0"
marginwidth="0" marginheight="0" allowfullscreen mozallowfullscreen webkitallowfullscreen
oallowfullscreen msallowfullscreen></iframe></div>
```

Query Asset File Path Program Response

You should prepare the server-side program accept the arguments send form editor and return the result in XML as the following format:

XML

```
<?xml version="1.0" encoding="utf-8" ?>
<assets>
    <asset filename="P0nDKB79" filetype="jpg"
    folder="http://demo.kcly.com/vcasmo_offline/2" fileext="jpg" />
    <asset filename="Ut5x80W9" totalslide="8" filetype="ppt"
    folder="http://demo.kcly.com/vcasmo_offline/2" fileext="ppt" />
    <asset filename="O6wgInHO" duration="2009" filetype="mp4"
    folder="http://awsvideo.vcasmo.com/92/20120323091811" fileext="mp4" />
</assets>
```

filename (required)

First 8 characters, based on naming rules to load thumbnail.

folder (required)

Full path of this asset file located, start with http / https (no need last forward slash).

fileext (required)

File extension, without "."

filetype (required)

File type symbol used in VMAPI, mp4=video, jpg=image, mp3=MP3, ppt=PowerPoint, pdf=PDF, swf=Flash.

totalslide (required for PowerPoint / PDF)

Total number of slides in a PowerPoint or PDF.

duration (required for video / MP3)

Duration of video / mp3, unit: integer second. (if you cannot get duration of mp3, you can pass *duration="0"*)

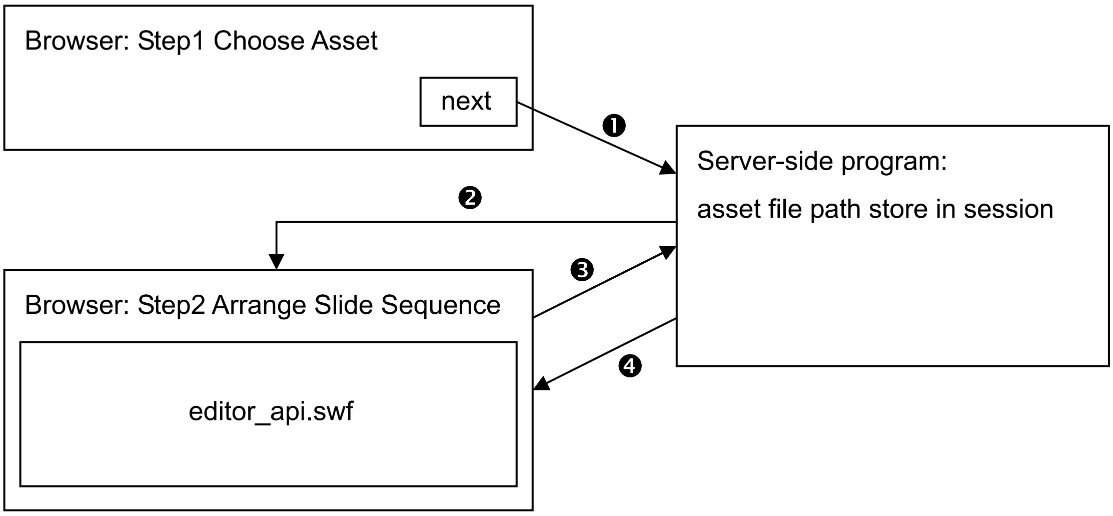
streamserver, streamfolder (optional for video / MP3)

For editor and presentation (HTML5 version), it will play video / MP3 through progressive download. For presentation (Flash version), it support Flash Media Server streaming, so you can provide these two extra arguments if you want to use streaming. eg: Amazon CloudFront: *streamserver="rtmp://abc123.cloudfront.net/cfx/st"*, *streamfolder="subfolder1/subfolder2"*. Note: progressive download and streaming, the filename must be same.

Editor will based on result to load asset file: ***folder + filename + "." + fileext***

Developer's Tip

You can create an interface for user to choose the asset, the asset list is directly list out what files there under the user asset folder in the server. When user chose asset and go to next step, ❶ you can pass all the chosen file path to server and store in session and ❷ redirect to next page, the editor ❸ query your server-side program defined in *assetfile_path*, this program will ❹ return the data in session as XML.


Browser: Step1 Choose Asset
next
❶
Server-side program:
asset file path store in session
❷
❸
Browser: Step2 Arrange Slide Sequence
editor_api.swf
❹

We recommend provide image / video size and follow the naming rule stated in [File Naming Rule and Thumbnail Used in VMAPI](). However, you can also provide your own image / video size and naming rule. For maximum image size supported by VMAPI, it is 1400 pixel (width) x 1050 pixel(height). No limit on video size, but you have to consider the payback capability in different mobile devices and user’s bandwidth condition, so we recommend 640 x 480 or 720p video.

API for Create New Presentation

When user finish the creation in editor and click save. Editor will send the slide sequence to VMAPI server. VMAPI send back the sequence data file to your server through HTML POST.

vcasmo.presentation.sendback (Not API)

VMAPI will POST to *sendback_path* with the following parameters:

swfdata (required)

Sequence data SWF file, for Flash version presentation playback.

xmldata (required)

Sequence data XML file, for HTML5 version presentation playback.

ts (required)

timestamp integer.

subtitle_langid, subtitle_langvalue (optional)

If presentation created with subtitle, these parameters will be included, e.g. if subtitle is Japanese, *subtitle_langid=2*, *subtitle_langvalue=日本語*. If subtitle language is not in the list of subtitle editor, user can input a custom language value, so *subtitle_langid*=, subtitle_langvalue=custom+lang:+Klingon (VMAPI supported languages can get from vcasmo.presentation.language)

userid, presentationid, subtitleid, custom1, custom2 (optional)

Just return what VMAPI received back to your server.

API for Edit Presentation / Insert Extra Asset to Presentation

The workflow is similar to create presentation. If user want to insert extra asset to presentation, you have to use the solution stated in API for Pass Chosen Asset to Editor to tell the editor where is extra asset file path, with the existing sequence data file location. If user want to edit presentation only, just tell the editor where the existing sequence data file located, asset related part can be omitted.

Insert Extra Asset to Presentation Using Sequence Editor

Arguments to Editor in HTML

domainkey (required)

sendback_path (required)

return_success_path (required)

userid, presentationid, custom1, custom2 (optional)

assetfile_path (optional)

If no assetfile_path provide, it means edit presentation, otherwise, means insert extra asset to presentation.

img_size_suffix (optional)

If your image file naming is not follow VMAPI rule, when you want to insert extra asset, extra asset file naming is not follow VMAPI rule, too. Editor cannot support mixed naming rule(one use VMAPI rule, one use own rule), both must be the same rule.

return_error_path (optional)

If any error happen on saving, it will redirect to an error page (please provides in full path) with response argument in URL query string includes: userid, presentationid, custom1, custom2, *msg*.

msg: extra error information from editor

msg=saveFail	VMAPI license expired, cannot save.
msg=assetfile_path	Cannot load asset file.
msg=seqdata_path	Cannot load sequence data SWF file.

return_success_path (optional)

When user click “See It Now”, the editor send the data to VMAPI. If the presentation is successfully created, editor will redirect the current page to a success page (please provides in full path) with response argument in URL query string includes: ts, userid, presentationid, custom1, custom2. (If this parameter is omitted, editor will keep on the current screen after user click “Save”, and “See It Now” button will be hidden)

seqdata_path (required)

The full path of sequence data SWF file for the presentation.

Example

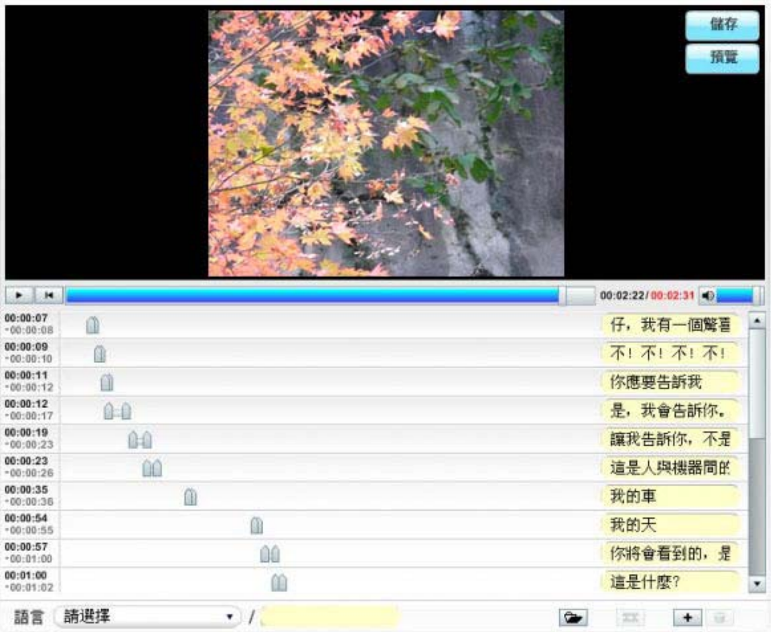
These arguments have to put it as URL-encoded variables in HTML code passing into editor.

```
<div class="vm_embed_editor_div" style="width:100%;height:100%;overflow:hidden;"><iframe
class="vm_embed_editor_iframe" src="//api.vcasmo.com/embed_api_editor.htm?
domainkey=g5H6i7J8k&assetfile_path=http://www.yourdomain.com/getAsset.php?sessid=123&seqd
ata_path=http://www.yourdomain.com/user/asset/123.swf&
sendback_path=saveSeqData.php&return_success_path=step2.php" allowtransparency="true"
width="100%" height="100%" seamless="seamless" scrolling="no" frameborder="0"
marginwidth="0" marginheight="0" allowfullscreen mozallowfullscreen webkitallowfullscreen
oallowfullscreen msallowfullscreen></iframe></div>
```

API for Subtitle

When a subtitle is added to a presentation, the subtitle data is stored into sequence data file, too. Therefore, suppose a presentation with three translation versions. It has 8 sequence data files (normal no subtitle version and three translated versions). Developers have to design a good architecture to organize these different files, in order to avoid overwrite each other. Subtitle editor (subtitle_api.swf) is an easy to use editor for adding subtitle to the presentation in VCAMSO. So we encourage developers to include the subtitle editor directly in your own application without re-invent the wheel.

To include the editor directly to your application, using the following HTML code:



00:00:07 -00:00:08	仔，我有一個驚喜
00:00:09 -00:00:10	不！不！不！不！
00:00:11 -00:00:12	你應要告訴我
00:00:12 -00:00:17	是，我會告訴你。
00:00:19 -00:00:23	讓我告訴你，不是
00:00:23 -00:00:26	這是人與機器間的
00:00:35 -00:00:36	我的車
00:00:54 -00:00:55	我的天
00:00:57 -00:01:00	你將會看到的，是
00:01:00 -00:01:02	這是什麼?

```
<div class="vm_embed_editor_div" style="width:100%;height:100%;overflow:hidden;"><iframe
class="vm_embed_editor_iframe" src="//api.vcasmo.com/embed_api_subtile.htm?
domainkey={arguments}&assetfile_path={arguments}&
sendback_path={arguments}&return_success_path={arguments}" allowtransparency="true"
width="100%" height="100%" seamless="seamless" scrolling="no" frameborder="0"
marginwidth="0" marginheight="0" allowfullscreen mozallowfullscreen webkitallowfullscreen
oallowfullscreen msallowfullscreen></iframe></div>
```

Arguments to Subtitle Editor in HTML

domainkey (required)

Your domain key.

seqdata_path (required)

The full path of sequence data file for the presentation. For editing subtitle, use the sequence data file with subtitle inside. For add subtitle, use the original sequence data file.

import_path (optional)

It is the path to another sequence data file with the subtitle applied on the same presentation. Import other's created subtitle timeline cue point. Then the user do not have to re-do again, he can just do the translation.

srt_path (optional)

*.SRT / SSA file. Directly import the text and timeline cue point into the subtitle editor. For non-English subtitle file,

please use UTF-8 encoding. If both *import_path* and *srt_path* are specified, *import_path* will be ignored.

sendback_path (required)

Sequence data SWF file will send back together with parameters.

return_success_path (optional)

When user click save, the subtitle editor send the data to VMAPI. If the subtitle is successfully created, subtitle editor will redirect the current page to a success page with response argument in URL query string includes: ts, userid, presentationid, subtitleid, custom1, custom2. (If this parameter is omitted, editor will keep on the current screen after user click “Save”)

return_error_path (optional)

If any error happen on saving, it will redirect to an error page with response argument in URL query string includes: userid, presentationid, subtitleid, custom1, custom2.

auto_save (optional)

Set *auto_save=1*, subtitle editor will call VMAPI immediately when loaded, it is useful when you want to do off screen (hidden iframe) auto subtitled presentation creation with *.SRT / SSA file, or update all the subtitled presentations after basic presentation was updated. Auto save works only when *sendback_path* is provided, and there are subtitles existed in the presentation.

userid, presentationid, subtitleid, custom1, custom2 (optional)

Just return what VMAPI received back to your server.

Example

These arguments have to put it as URL-encoded variables in HTML code passing into editor.

```
<div class="vm_embed_ subtitle _div" style="width:100%;height:100%;overflow:hidden;"><iframe
class="vm_embed_subtitle_iframe" src="//api.vcasmo.com/embed_api_subtitle.htm?
domainkey=g5H6i7J8k&seqdata_path=http://www.yourdomain.com/user/asset/123.swf&
sendback_path=saveSeqData.php&return_success_path=step2.php" allowtransparency="true"
width="100%" height="100%" seamless="seamless" scrolling="no" frameborder="0"
marginwidth="0" marginheight="0" allowfullscreen mozallowfullscreen webkitallowfullscreen
oallowfullscreen msallowfullscreen></iframe></div>
```

Developer’s Tip

When original presentation has any changes (e.g. slide sequence timing), those subtitled presentations are remain unchanged. You can set *seqdata_path=new original presentation SWF file* and *import_path=subtitled presentation SWF file* for user to update the changes on subtitled presentation.

vcasmo.presentation.language

Get the VMAPI supported subtitle language list.

Arguments

format (optional)

1=XML, 2=Comma string, default: 1.

Example

http://api.vcasmo.com/presentation/language?format=1

Response

XML

```
<?xml version="1.0" encoding="utf-8" ?>
<rsp stat="ok">
    <subtitles total="3">
        <subtitle id="1">
                <lang><![CDATA[English]]></lang>
                <lang_en><![CDATA[English]]></lang_en>
        </subtitle>
        <subtitle id="2">
                <lang><![CDATA[日本語]]></lang>
                <lang_en><![CDATA[Japanese]]></lang_en>
        </subtitle>
        <subtitle id="3">
                <lang><![CDATA[國語/中文(正體)]]></lang>
                <lang_en><![CDATA[Mandarin (Traditional Chinese)]]></lang_en>
        </subtitle>
    </subtitles>
</rsp>
```

Comma string (no space)

```
1,English,English,2,日本語,Japanese,3,國語/中文(正體),Mandarin (Traditional Chinese)
```

Presentation Playback

VCASMO presentation player (vcasmo.swf) is the core file for playback the presentation. You can host this file in your server. Therefore, if you stop use VMAPI, you can still keep all your existed presentation functioning, just cannot edit or create new one.

Arguments to Presentation Player through HTML

seqdata_path (required)

Full path to sequence data SWF file. If user is watched from iOS device, it will auto change to sequence data XML file.

use_xml_data (optional)

If you want Flash version and HTML5 version share the same XML (i.e. dump sequence data SWF file, your server does not need to store that file if you do not need to edit presentation again, add/edit subtitle), set *use_xml_data=1* and seqdata_path=full path of sequence data XML file. (Note: reason to use sequence data SWF file for Flash version, because SWF version file size is smaller and lower overhead when loading and parsing in Flash version)

For extra argument related to customize the player look and feel or skinning, please refer to “VCASMO Presentation Player Skinning Guide”.

Example

These arguments have to put it as URL-encoded variables in HTML code passing into editor.

```
<div class="vm_embed_div" style="width:100%;height:100%;overflow:hidden;"><iframe
class="vm_embed_iframe" src="//api.vcasmo.com/embed_api_player.htm?seqdata_path=
http://www.yourdomain.com/user/asset/123.swf" allowtransparency="true" width="100%"
height="100%" seamless="seamless" scrolling="no" frameborder="0" marginwidth="0"
marginheight="0" allowfullscreen mozallowfullscreen webkitallowfullscreen oallowfullscreen
msallowfullscreen></iframe></div>
```

Extra Arguments support in Presentation Player

canprint (optional)

default: slides can be printed, set *canprint=0* to disable it.

traneffect (optional)

default: there is fade in/out effect when slide changes from one to another, set *traneffect=none* to disable it.

videoonright (optional)

default: video position on left and slide position on right, set *videoonright=1* to swap video and slide position.

For other arguments support in presentation player, please refer to Customization Guide in VCASMO Labs website

Developer’s Tip

We recommend host VCASMO presentation player in your server. You can download necessary files from VMAPI Admin Panel (https://api.vcasmo.com/my_login.php).

Suppose VCASMO presentation player main HTML file (*embed_api_player.htm*) located in your server path: http://www.yourdomain.com/vcasmo/web_api/embed_api_player.htm, change the above embed code **bold red color part** to *www.yourdomain.com/vcasmo/web_api*