CLM Distribution module

TECHNICAL GUIDELINES v.2.1

(iPad version)

CONTENTS

Overview

The **Closed Loop Marketing (CLM)** software allows to:

- Deliver multimedia content, documents and digital material to Field Forces;
- Provide basic information on content usage by Field Forces (e.g. first and last content access, number of accesses for each content);
- Define a content validity range, to be sure that Field Forces are always working with the correct and always updated digital support.

This document contains technical guidelines that help e-detailing makers to produce contents compliant with the CLM platform.
The document includes information regarding the development of contents for iPad devices.

In the paragraphs below it will be described relevant aspects such as:

- Platform requirements
- Best practices for content creation
- Menarini devices specifications
- Masterbook capabilities usage
- Tracking capabilities usage

1. General Requirements

In order to start to create CLM contents, there are some general requirements to be considered:

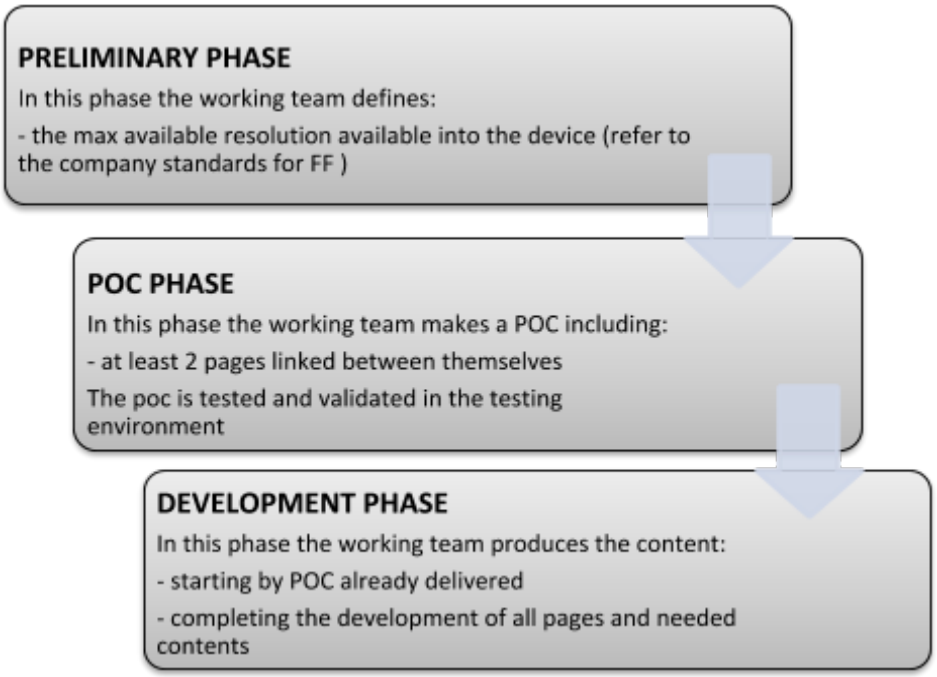
- The content created should pay attention to the battery usage on field forces devices and to consider the number of visualization with a charged battery.
- The content should be optimized in term of sizing in order to be easily downloaded via 3G connection. See Video Integration chapter.

1. Visual implementation

The realization of HTML5 contents should respect indications included in this section.

1.1. VisualAid process making lifecycle

In order to leverage the visualAid making process, it's convenient to follow the approach explained into the following workflow:



This workflow assures that developed visualAid are well made and ready for the target devices, so that during the "Development" phase the developing team has already worked over following aspects:

- Dimensioning of content over target device

1.2. Paths and Change location management

There are some indications to be followed regarding the used paths and change location.
The developer needs to follow the following rules:

- Change of locations must be implemented only by using Javascript code, not by HTML statements
- In JavaScript, the change of location must be implemented by using the function ***goToLink(path)*** available within the initial setup script (see section 3.1)
- the change location *paths* must be identified in absolute URI. You could use a syntax like the following one:
  path = location.href.substring(0,location.href.lastIndexOf('/'))+'/'+$(this).data('target');

1.3. JQuery Mobile Integration

In case you are using the Javascript library **jquery mobile v.1.4.5,** we recommend adding this script before the import statement of the library in order to avoid the error "*Failed to execute 'replaceState' on 'History' <local_URL>".*

```
<script>
   $(document).bind('mobileinit',function(){
      $.mobile.pushStateEnabled = false;
   });
</script>
<script type="text/javascript" src="js/jquery.mobile-1.4.5.min.js"></script>
```

In any case we suggest to be always "state of the art" from this point of view, and so to use the most recent and stable library version available.

1.4. Video integration

The following table contains the native supported video formats for iPad (on standard iPad model).

iPad
iPad device supports natively following list of video formats: • H.264 video up to 4K, 30 frames per second, High Profile level 4.2 with AAC-LC audio up to 160 Kbps, 48kHz, stereo audio or Dolby Audio up to 1008 Kbps, 48kHz, stereo or multichannel audio, in .m4v, .mp4, and .mov file formats; • MPEG-4 video up to 2.5 Mbps, 640 by 480 pixels, 30 frames per second, Simple Profile with AAC-LC audio up to 160 Kbps per channel, 48kHz, stereo audio or Dolby Audio up to 1008 Kbps, 48kHz, stereo or multichannel audio, in .m4v, .mp4, and .mov file formats; • Motion JPEG (M-JPEG) up to 35 Mbps, 1280 by 720 pixels, 30 frames per second, audio in ulaw, PCM stereo audio in .avi file format (for additional details visit: https://sites.google.com/menarini.com/ff-helpdesk where is referenced the official page of Apple for the standard device)

In addition, to make quicker the upload/download process of a visual, it is recommended to reduce the video size as possible.

1.4.1. Example of video compression

To convert video to the suggested video formats and also to reduce their size, you can use for example the following video converters: Freemake Video Converter, Free MP4 Video Converter (http://www.dvdvideosoft.com/products/dvd/Free-MP4-Video-Converter.htm).

Below a simple tutorial that shows how to compress a 55 MB MP4 video through the tool Free MP4 Video Converter. You can download this tool here: http://www.dvdvideosoft.com/products/dvd/Free-MP4-Video-Converter.htm.
(1) open the tool, drag and drop the original video
(2) set the field Preset to WebM.
(3) then click on Convert.

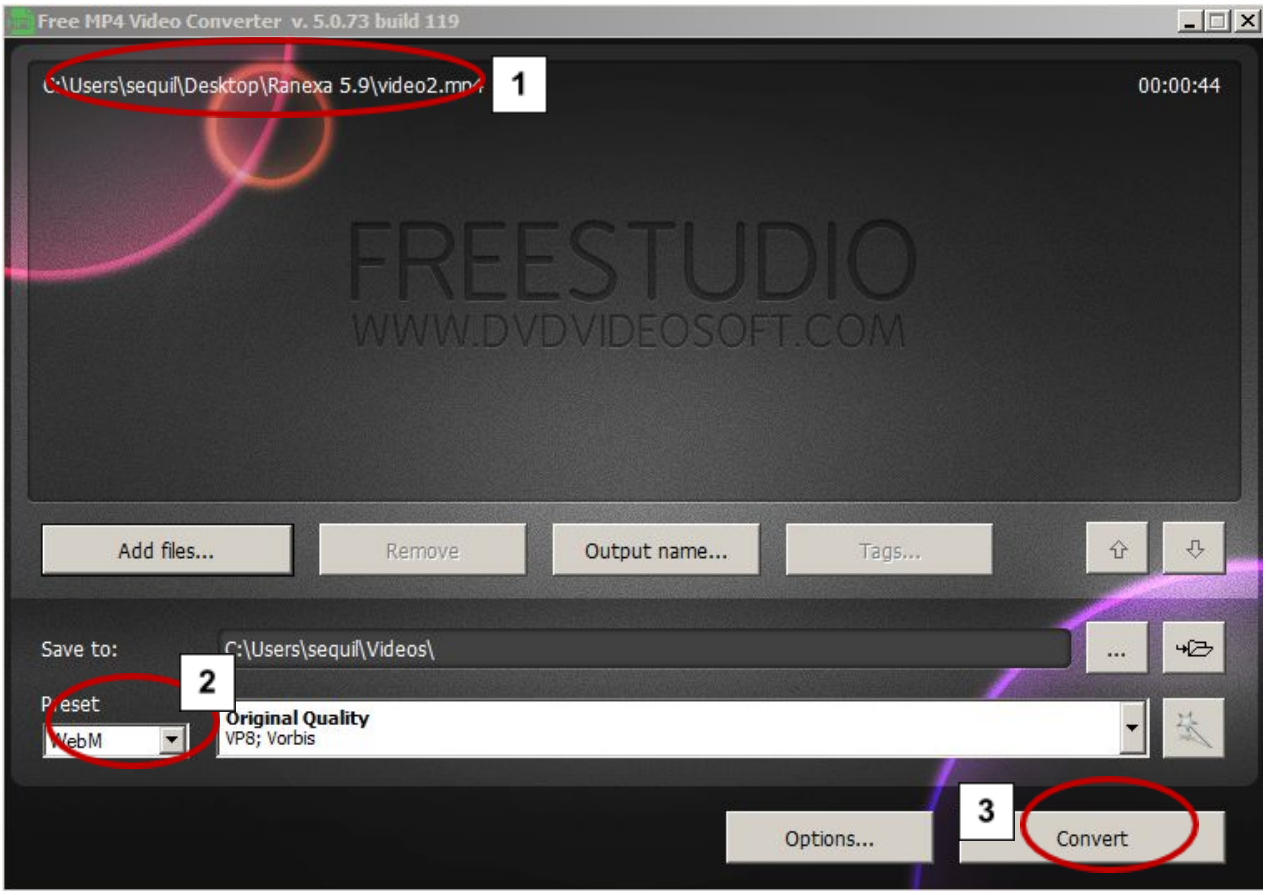


Figure 1 – Free MP4 Video Converter

The **WebM** video obtained now has a lower size: **7.7 MB**. We have **reduced the video size of 86%.**

1.5. Indications for Tracking tags insertion

In case that your content have to include the Tracking feature (see also chapter 4.2 for technical implementation), then the insertion of tracking tags have to respect some recommendations in order to better browse them into the analytics (i.e. business intelligence reports, analytics,..) and in order to obtain an homogenous outcome.
Let start from the "*CLM – Tracking Content Specification form*" (in Excel format) that has been compiled by the content owner (i.e.: PM).
This document includes important information that will be tracked and available into tracked data for future important analysis:

- The page identifier and its description
- The tracked item (identifier, type, event to track)

Generally, in accord to the level of analysis requested, over a page are possible 2 levels of tracked events:

- the page events
- item events inside the page

The function *Tracking.startEvent()* takes in input the following event fields to track events:

- eventId
- eventDescription

that will be used in the previous two cases as explained into the following table:

Tracked item	Given input arguments
Page item	**eventId** = *composed by the following values (separated by a pipeline):* **<u>for Corporate Product Content</u>** - *"CORP" costant keyword* - *ISO country code* - *Department* - *Content identifier* - *Page name* - *Page name* - *"Page" constant* - *Event type* - *Semantic type* **<u>for Affiliate Product Content</u>** - *"***" as separator* - *"AFF" constant keyword* - *ISO code of country* - *Department* - *Content identifier* - *Affiliate page name* - *Affiliate page name* - *"Page" constant keyword* - *Event type* - *Semantic type* **eventDescription** = *description of the event.*
Widget item (inside the page)	**eventId** = *composed by the following values (separated by a pipeline):* **<u>for Corporate Product Content</u>** - *"CORP" constant keyword* - *ISO country code* - *Department* - *Content identifier* - *Page name* - *Item name* - *Item type* - *Event type* - *Semantic type*

	for Affiliate Product Content - *"***" as separator* - *"AFF" constant keyword* - *ISO code of country* - *Department* - *Content identifier* - *Affiliate Page name* - *Affiliate Item name* - *Item type* - *Event type* - *Semantic type* **eventDescription** = the *event description*

Tracked events on the page, have to be identified into the tracked tags within specific identifiers. The following table shows the map between tracked events and their identifiers:

2. Advanced capabilities

CLM platform includes some advanced functionalities (MasterBook and Tracking).
The platform exposes some API providing to the content makers needed functionalities to implement them.

Development phase

Masterbook

The Masterbook can't be tested outside the CLM app, so the developer needs to test it directly on the iPad Test device.
iPad native app already manages the Bookmarks navigation applying an additional wrapping front-end layer compliant with these methods and they are not needed.

Tracking

During the initial development phase the developer can test the visual and simulate the Tracking functionalities by importing manually the following library: Tracking-lib.js. (Preliminary Test Phase)

```
<script src="Tracking-lib.js">
```

The development library *tracking-lib.js* is available together with other shared material.
This library simulates the Tracking functionalities by logging in the Chrome console the tracked information.

Once the preliminary test phase is completed, the visual-aid can be uploaded inside the *M-Field-Agency* app.

In order to upload the visual the developer needs to:

- Remove the import statements from the visual html file
  `<script src="Tracking-lib.js">`
- Upload the visual NOT including the Tracking libraries because they are already present inside the app.

Just to summarize this process:

- The agencies develop the content using the "development libraries" (having the same name and method signature of the actual one but don't need CLM and track on the browser console)
- Then they remove the "development libraries" from the folder and upload the content in Web-Agency app where the actual libraries are used.

2.1. Tracking

The Tracking library implements the following functions:

- Tracking.startEvent(eventId, eventDescription, attribute, [cron, testEvent])
- Tracking.endEvent(startEventReturnId)
- [Tracking.closeSession(sessionId)]

Tracking.startEvent(Event_ID, event_description, attribute, cron, [is_test_event])

This function is used to start Tracking an event. The function has 5 parameters:

- **Event_ID:** unique event identifier. It is a string value. (required)
- **Event_description:** event description. It is a string value. (required)
- **Attribute:** this field can be used to store extra information, for example user input (textbox, checkbox selected…). It is a string value.
- **Cron**: it is a boolean value. It should be set to true if you need to track the event duration.
- **Is_test_event**: if true, it means it is a test event. When an event is flagged as 'test', automatically the session is flagged as test as well. This argument is optional for iPad because directly driven by device; in case is inserted, the iPad value will override it.

The method returns an auto-generated unique id for the new event. This id will be used eventually in order to call the method endEvent.

```
Tracking.startEvent('eventId','eventDes',null,true,false).then(
        function(idEvent){

        ,function(err){
                console.error(err);
        }
);
```

Tracking.endEvent(Event_ID)

This function is used to close a Tracking event by passing the event identifier. The id is the one retrieved by startEvent method call.

Note

The eVisual-Aid should be inserted with tags, at all appropriate occurrence of events that can be technically captured in the eVisual-Aid such as opening & closing the visual, navigations, clicks, opening & closing embedded contents for tracking the usage.

Example: how to track a Duration event

"Let suppose that the content has to be created with the tracking feature over a video content. The users want to track the duration of the video browsing event"

In this case the developer has to proceed as follows:

- Identify which are the ***eventId*** and ***eventDescription*** attributes (see the "*Tracking Content form"* file)
- Over the video opening event insert the function call *Tracking.startEvent(**eventId**, **eventDescription**, null, true, false)*
- *Over the video closing event* insert the function call *Tracking.endEvent(**eventId**)*

Example: how to track a Time spent on the page event

"Let suppose that the content has to be created with the tracking feature over a page. The users want to track the time spent on the page under tracking event"

In this case the developer has to proceed as follows:

- Identify which are the ***eventId*** and ***eventDescription*** attributes to be used for the Page item in accord to the "*Tracking Content form"* file
- Over the page loading event insert the function call

*Tracking.startEvent(**eventId**, **eventDescription**, null, true, false)*

- *Over the page exit event* insert the function call *Tracking.endEvent(**eventId**)*

3. Other technical indications

This section includes other technical indications that could be useful.

3.1. Responsive design

In cases explained into the functional guide lines, it could be decided to use the responsive design. The responsive design lets visual contents fit multiple screen resolutions, and in this case it is necessary follow the Responsive Design rules.
In order to make the visual screen adaptive, the DOM elements dimensions shouldn't have fixed values (for example *width:1024px* or *height:768px* indicates fixed dimensions), but percentage values.
Below a sample code of percentage usage:

```
<html>
  <head></head>
<body>
  <div style="width:100%; height:10%; background-color:yellow;" > Menu toolbar </div>
  <div style="width:100%; height:90%; background-color:orange;"> Content section </div>
</body>
</html>
```

4. Device Specifications

The content will be used by field force using:

- iPad models used by The Menarini Group
-

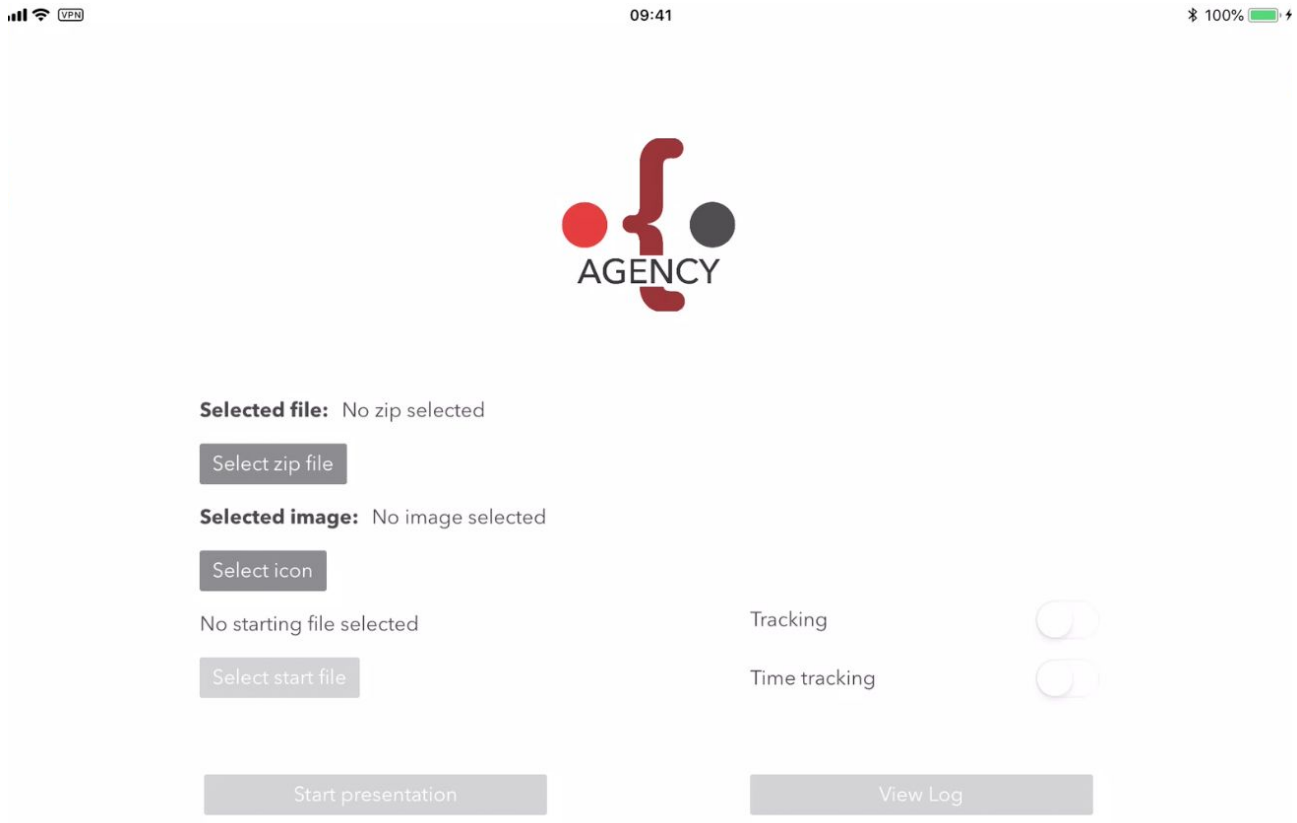
For iPad models, the resolution is the same for each device; please refers to the iPad specs (https://sites.google.com/menarini.com/ff-helpdesk).

5. Development and testing process

The development and testing process foresees that the M-Field-Agency app can be used in order to check inserted tracking tags and verify the visual correctness over a real iPad.

The steps in order to work with the CLM Web-Agency app are:

- download the app “M-Field-Agency” provided to start the work
- Install the app over your iPad (of same model used by field forces) using the proper tools (i.e. Apple Configurator)
- on iPad trust the certificate of app from configuration functions "Device Management"
- start the app
- load the e-detailing content into the iPad memory (i.e. using AirPlay)
- upload the content inside the “M-Field-Agency app” using “select zip file” and “select icon” functions



- define the start file of content (i.e. index.html file) and decide if enable the tracking capabilities
- start the presentation: you can display the content and than verify the tracking logs accessing to the “View Log” section

CHANGE VERSION HISTORY

Version	Date	Change Description
1.2	15/09/2016	Added CVM Hardware Minimal Requirements Removed section Content Test Toolkit. Added sections: Create a shared folder inside the CVM, CLM User Manual
1.3	15/9/2016	Revision of tracking patterns in order to gather useful data
1.6	18/10/2016	Improved CLM User Manual section
	05/05/2017	Document improvements: - Additional indications regarding technical tracking capabilities - Revision of Content Inspector inclusion
1.8	21/06/2017	Addition of "3.1. VisualAid process making lifecycle" section
1.9	23/11/2017	Document improvements: - Inclusion of iPad device
2.1	19/7/2018	Revision of web-agency test process and tools adoption