flashmaps

AreaSelector Guide

Version 2.6.2

flashmaps

pushing the **limits** of **mapping**

Index

1. Introduction

1.1. Overview

This guide provides complete information about Flashmaps AreaSelector, and the best practices to integrate it into your website.

1.2. Architecture

Think about AreaSelector as a "map machine" available to provide mapping capabilities to your databases.

Basically, AreaSelector is a Macromedia Flash object which offers an API consisting of a series of JavaScript/Flash functions. These will allow you to send commands to the map, like loading a theme, focusing on a specific area, etc.

You can configure AreaSelector by editing XML files containing the area and POI (Points of Interest) information which define your map properties (e.g. modifying colors, icons, events, and assigning actions to events).

In a nutshell, this API (Application Programming Interface) and the XML files permit a complete integration of your AreaSelector in a flexible, dynamic and easy way.

A few of these functionalities take advantage of HTML-to-Flash and Flash-to-HTML communication. Windows browsers like Internet Explorer, Firefox, Netscape or Opera do fully support it, but some Mac browsers do not. This is due to some limitations with Macromedia Flash Player 7.

1.3. Builder Care

All AreaSelector technologies are developed at Flashmaps from scratch. This enables us to implement virtually everything you may need to fit your requirements.

Email us at support@flashmaps.com or call us toll free at **1-866-392-0071** to share your most ambitious mapping wishes with us, or let us know if we can assist you in anyway.

2. AreaSelector

AreaSelector allows you to integrate in your website a fully dynamic map containing a series of features which make it easy to interact with your website.

AreaSelector is a Macromedia Flash file which permits easy integration in any kind of website, as well as combining it with any flash application.

2.1. AreaSelector maps

AreaSelector comes with four main map types; namely:

2.1.1. AreaSelector USA States

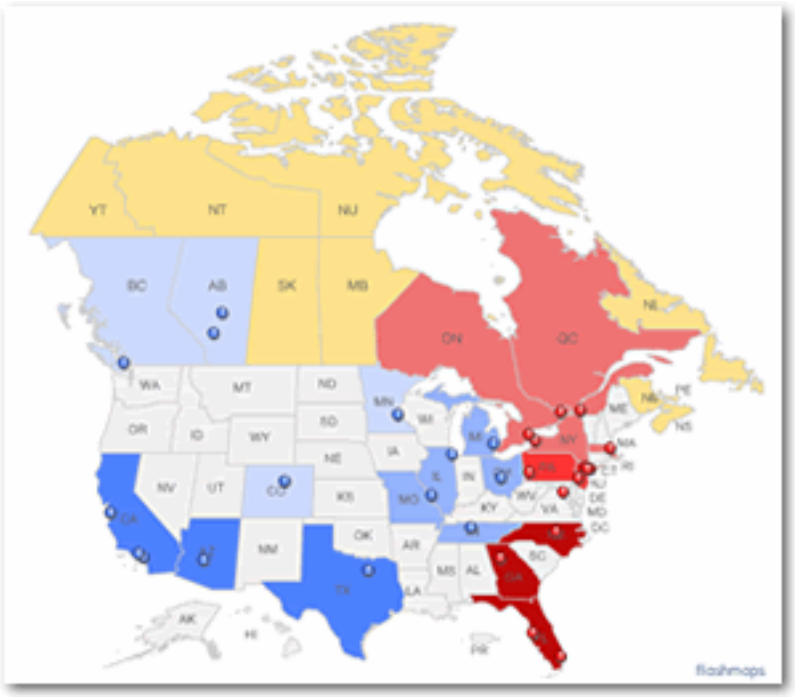
AreaSelector USA States is made up of two levels:

- Level 1: USA country.
- Level 2: USA state.

2.1.2. AreaSelector USA States and Canadian Provinces

AreaSelector USA + Canada contains the USA with its states and Canada with its provinces.

- Level 1: USA and Canada.
- Level 2: USA state or Canada province.



2.1.3. AreaSelector USA States and Counties

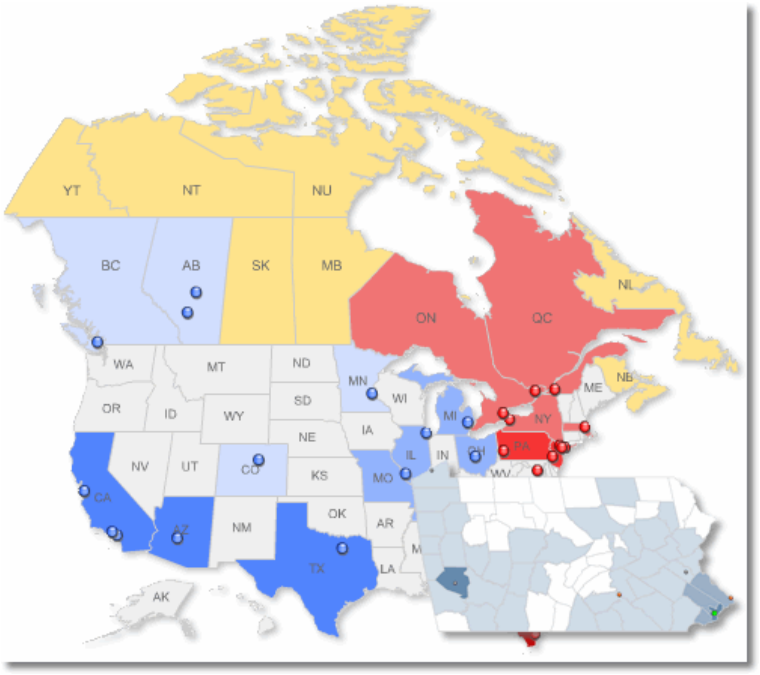
AreaSelector USA + Counties consists of three levels:

- Level 1: USA country.
- Level 2: USA state.
- Level 3: USA County.

2.1.4. AreaSelector USA States, Counties and Canadian Provinces

AreaSelector USA states and counties, and Canada provinces contains the USA with its states and counties, and Canada with its provinces.

- Level 1: USA and Canada.
- Level 2: USA state or Canada provinces.
- Level 3: USA County.



2.1.5. AreaSelector World

AreaSelector World is made up of three levels:

- Level 1: World.
- Level 2: Continent.
- Level 3: Country.

2.1.6. AreaSelector World and USA States

AreaSelector World and USA States is made up of four levels:

- Level 1: World.
- Level 2: Continent.
- Level 3: Country.
- Level 4: USA state.

2.1.7. AreaSelector World, USA States and Canada provinces

AreaSelector World, USA States and Canada provinces are made up of four levels:

- Level 1: World.
- Level 2: Continent.
- Level 3: Country.
- Level 4: USA state or Canada provinces.

2.1.8. AreaSelector World, USA States and Counties

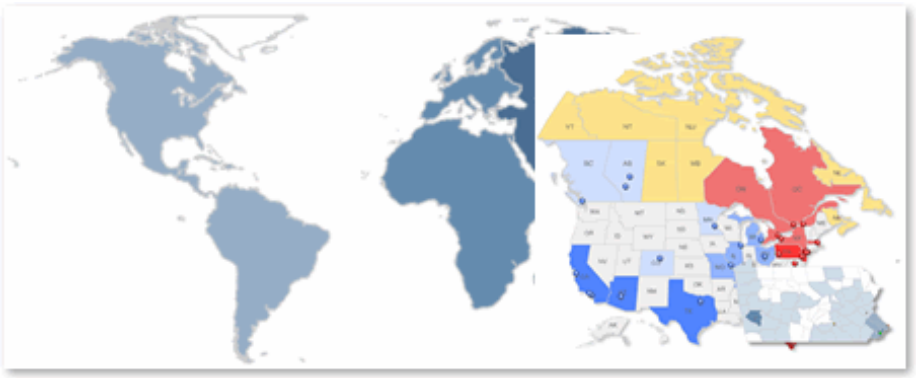
AreaSelector World, USA States and counties is made up of five levels:

- Level 1: World.
- Level 2: Continent.
- Level 3: Country.
- Level 4: USA state.
- Level 5: USA County.

2.1.9. AreaSelector World, USA States, USA Counties and Canada provinces

AreaSelector World, USA States and counties, and Canada provinces are made up of five levels:

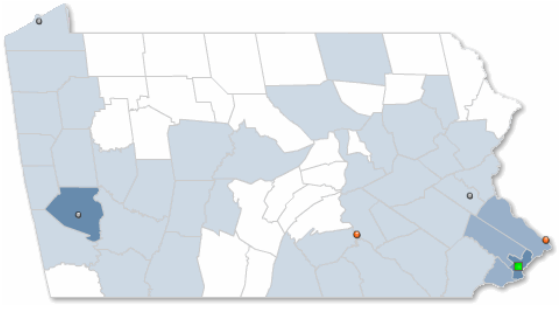
- Level 1: World.
- Level 2: Continent.
- Level 3: Country.
- Level 4: USA state or Canada provinces.
- Level 5: USA County.



2.1.10. AreaSelector USA Single State and Counties

AreaSelector USA Single State and Counties is made up of two levels:

- Level 1: USA state.
- Level 2: USA counties.



2.2. Files

AreaSelector comes with a set of pre-defined files which are listed and shortly described in the following table:

Files	
fmASEngine.swf	AreaSelector's engine.
fmASEngine.xml	AreaSelector's configuration.
fmASIconLibrary.fla	Icon library's flash source.
fmASIconLibrary.swf	Icon library.
fmASToolbar.swf	AreaSelector's toolbar.
fmASUSA.swf	USA map.
fmASUSA.xml	USA theme's configuration.
areas/	
states.xml (in case of AreaSelector World: continents.xml)	Area configuration.
statesCategories.xml (AS World: continentsCategories.xml)	Area categories.
statesColors.xml (AS World: continentsColors.xml)	Area color palettes.
statesSummary.xml (AS World: continentsSummary.xml	Area additional information.
pois/	
pois.xml	POIs (Points of Interest).
poisCategories.xml	POI categories.
poisSummary.xml	POI additional information.

Legend with the color-coding of the files:

Files	
In red	Files you should not change.
In green	Files that you may change, but you'll always need.
In blue	Files provided as samples.

2.3. Theme

AreaSelector works with themes which provide the necessary information for map representation with properties and events.

A theme contains: a map which is a SWF Flash movie; a set of areas to interact with; and a set of POIs which can be shown on the map. You can load AreaSelector with various types of themes corresponding to different information which can be shown in a dynamic way.

Every theme is defined by a XML file which contains the configuration specifying where to obtain the area and POI information.

2.3.1. Areas

Zooming in upon an area is the main interaction that can be preformed upon specific areas.

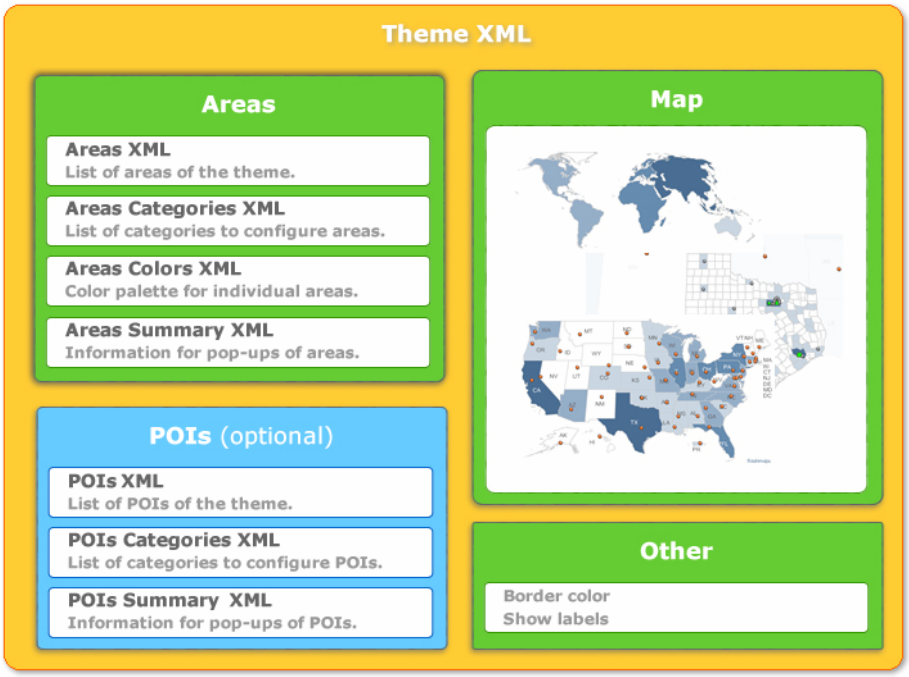
These areas have a set of properties which can be configured dynamically, like changing their colors, displaying area's tooltip or adding events to execute on them.

All the areas of a theme are grouped in categories to ease the application of their properties and events.

Inside the theme you have to specify the set of areas and their categories. For instance, in AreaSelector USA, by default, there is the theme containing the states and a category for them all defining their colors and events.

But, instead of setting colors and events to a full category, you can set them to each area individually.

All the information about the areas, their categories, colors and so on, are stored in static XML files or dynamic pages that output the corresponding XML format.



2.3.2. POIs (Points of Interest)

POIs (Points of Interest) let you show any kind of locations on the map, like company offices, stores, event sites, etc.

You can display the same set of POIs on all views, or make AreaSelector load a set of POIs when you enter an area, for example, when you click and zoom into a state. This is specially useful when you have an extremely large number of POIs displayed at global level.

A POI is defined by its latitude and longitude of its exact coordinates. By default, the icon corresponding to POI's category is normally used, however, you can also set a specific icon up for a single POI.

You can set POI events, like: showing a pop-up by clicking the POI; calling a JavaScript function by placing the cursor over the POI, and so on.

The POIs are grouped in categories which indicate the icon to represent them on the map and the actions to be executed for the members of a category.

In the same way as with the areas, it is possible to add an informative pop-up window responding to the cursor on an event.

All details about a set of POIs, its categories, the color scheme or additional pop-up information are stored in static XML files or dynamic pages that output the corresponding XML format.

2.3.3. Icons

On the map, a POI is represented by an icon which is normally defined by the POI's category.

You may use external icons or members of the icon library.

The main advantage of external icons is that the user does not need to modify the whole library to add a new icon. JPG and SWF files are allowed. The main handicap is a slower performance when displaying many POIs, since Internet Explorer loads an icon file for each of the POIs. Use external icons if you need to display very few POIs, or if you want a different icon for each POI.

Members of the icon library have to be inserted and modified inside that fmDBMPOIs.fla, using Macromedia Flash MX 2004 or higher. This increases performance considerably, since icons do not have to be loaded for each POI. We highly recommend using these icons, unless you need to display only a few POIs or you want a different icon for each POI. If you do not have Flash MX 2004 or higher, or would rather we take care of adding new icons or modifying existing ones for you, please forward said request to support@flashmaps.com

3. XML reference

3.1. fmASEngine.xml

fmASEngine.xml is AreaSelector's general configuration file. It has to be in the same directory as fmASEngine.swf.

Modifications in this file could affect the correct operation of the system, so make a backup copy before making any changes.

This is an example of what fmASEngine.xml looks like:

```
<fmEngine
  version="002.005.001"
  mapSize="63074184-36925814-24792479"
  serialNumber="fmas-0014-6267"
  theme="fmUSA/fmASUSA.xml"
  centerLat="35.12000" centerLon="-96.50000"
  mapWidth="500" mapHeight="300"
  mapScale="30000000" mapScaleMax="1600"
  initialScale="125" initialX="2" initialY="20"
  levels="2" selectDivider="," selectColor="0xFFF000"
  otherAreasTransparency="30" showLabel="true"
  POIs_SendParameters="true" >

  <fmPOIsLoad level="1" load="false" />
  <fmPOIsLoad level="2" load="false" />

  <fmMoveAndScaleArea
    minOriginLon="-172.15000" minOriginLat="52.26000"
    maxOriginLon="-129.60000" maxOriginLat="72.00000"
    centerTargetLon="-113.5000" centerTargetLat="28.0000"
    scale="0.33" />
  <fmMoveAndScaleArea
    minOriginLon="-160.75000" minOriginLat="18.72000"
    maxOriginLon="-154.60000" maxOriginLat="22.52000"
    centerTargetLon="-106.0000" centerTargetLat="28.0000"
    scale="1.2" />
  <fmMoveAndScaleArea
    minOriginLon="-68.00000"  minOriginLat="17.50000"
    maxOriginLon="-65.00000"  maxOriginLat="19.00000"
    centerTargetLon="-87.0000" centerTargetLat="27.50000"
    scale="1.2" />
</fmEngine>
```

Some attributes may be changed to configure AreaSelector's behavior:

Attributes

theme	Initial theme to load.
selectDivider	Character used to separate the

	elements in the selected area list.
selectColor	Color for selected areas.
otherAreasTransparency	Transparency percentage (100 means solid) for parent area when zooming in loads the next level.
showLabel	Show area's label in the top left corner ("true" or "false"). The default value is "true".
POIs_SendParameters	Add the POI parameters to URL when click over a POI ("true" or "false")

Within this configuration file you can also specify which map levels of AreaSelector the POIs should be shown.

You may also configure AreaSelector to load POIs statically (same POIs in every level) or dynamically (POIs are loaded depending on the area being focused upon).

Tags

<fmPOIsLoad>	Loading the POIs dynamically or not.

The corresponding attributes are the following:

Attributes

level	AreaSelector level.
load	State whether you are loading POIs dynamically or not ("true" or "false"). The default value is "false".

If one or more levels are configured to load the POIs dynamically, AreaSelector calls the specified URL to load new POIs every time it enters these levels, sending as parameters: the level, the area ID, and the sub area, as in the following example:

…/pois.asp?level=2&area=US&subarea=

The parameters are:

Parameters

Level	Level.
area	Area that has been clicked.
subarea	Parent area.

3.2. theme.xml

AreaSelector can display on its map specific sets of areas on which sets of POIs can be displayed. Under said conditions, themes are used.

A theme consists of a map associated with a set of areas and POIs. All of them with their own data, properties, events, customization, etc.

Theme's general configuration is stored inside an XML file, which has the same name as the theme's main map, except that the "swf" extension is replaced by "xml". For example, default AreaSelector USA State's main map is fmASUSA.swf, so the corresponding configuration file has to be named fmASUSA.swf.

Please see the following example of a theme configuration file:

```
<theme
  id="USA"
  mapFile="fmASUSA.swf"
  mapBorder="0xCCCCCC"
  areas="areas/states.xml"
  areasColors="areas/statesColors.xml"
  areasCategories="areas/statesCategories.xml"
  areasSummary="areas/statesSummary.xml"
  pois="pois/pois.xml"
  poisCategories="pois/poisCategories.xml"
  poisSummary="pois/poisSummary.xml">
</theme>
```

As you can see, it consists on a single XML element:

Tags

<theme></theme>	Start/end tag of theme.

The theme node contains a set of attributes similar to the areas and their properties or the set of POIs and its properties.

Attributes

id	Unique ID of the theme.
mapFile	Map's URL.
mapBorder	Map's border color.
areas	URL containing the list of areas (XML format).
areasCategories	URL containing the area categories (XML format).
areasColors (optional)	URL containing the list of area colors (XML format).
areasSummary (optional)	URL containing the additional information of the areas to show in the summary pop-up (XML format).
pois (optional)	URL containing the set of POIs to load on the map (XML format).
poisCategories (optional)	URL containing the POI categories (XML format).
poisSummary (optional)	URL containing the additional information of the POIs to show in

	the summary pop-up (XML format).
showTexts (optional)	States whether the area labels will be shown (“true”) or not (“false”) when you hover over them. The default value is “true”.

Some of the attributes are optional, depending on the configuration you want to set.

If you want to set a given color for one specific area, you have to define this inside areasColors.xml. Otherwise, each area is colored as specified in its category, and you do not need the aforementioned areasColors.xml.

A theme can or can not have a set of POIs. If it does not, the attributes concerning the POI files remain empty. If you display POIs, the ‘pois’ and ‘poisCategories’ attributes need to have valid values.

Also optional are the attributes referring to the additional information which can be shown in pop-up windows when you hover over an area (‘areasSummary’) or a POI (‘poisSummary’). If you need to show this pop-up information, you have to also make sure to define the ‘catSummary’ attribute inside the areasCategories.xml or poisCategories.xml respectively.

3.3. areas.xml

This XML file contains the list of areas in the current theme, which will be loaded inside AreaSelector.

Each area corresponds to one movie clip and has a unique ID. Also, each area belongs to a category and has its own configuration file.

The following is an example of a areas.xml file:

```
<areas>
  <area areaId="AL" areaLabel="AL" areaCategory="state" />
  <area areaId="AK" areaLabel="AK" areaCategory="state" />
  <area areaId="AZ" areaLabel="AZ" areaCategory="state" />
  <area areaId="AR" areaLabel="AR" areaCategory="state" />
  ……
</areas>
```

It contains the following XML tags:

Tags	
<areas>	The collection of areas.
<area>	Each of the areas.

Each area contains the following attributes:

Attributes

areaId	Unique ID of the area.
areaCategory	Area's category.
areaLabel	Area's label, to be displayed when you hover over it.
areaMovie (optional)	If the area has children areas (e.g. the collection of states inside USA), map URL.
areaEnabled (optional)	Set if this area is enabled ("true") to be clicked or disabled ("false"). The default value is "true".
areaTooltip (optional)	Show tooltip ("true") or do not ("false"). The tooltip displays the area label. The default value is "true". (Note: Tooltips will not be shown if the summary pop-up is used).
areaZoom (optional)	Set if you can zoom in this area ("true") or not ("false"). The default value is true.

If you want to load children areas when clicking on one, you need to set 'areaMovie' attribute, which specifies the map to loada flash file. The same path should also contain the corresponding XML configuration file.

In this example, California ('areaID="CA"') is configured in such a way that, in case of clicking the state, the counties inside ('counties/CA.swf') will be loaded:

Example

```
<area areaId="CA" areaLabel="CA" areaCategory="state"
areaMovie="counties/CA.swf" />
```

3.4. areasCategories.xml

Area categories contain a set of properties and events which apply to all areas in the same category.

The areasCategories.xml file summarizes the categories with their properties and events to simplify the process of modifying a set of areas.

Each category has a unique identifier ('catID') corresponding to the attribute 'areaCategory' of each area (see chapter 3.3. areas.xml).

An example of an areasCategories.xml file is given here:

```
<areasCategory>
  <areaCategory catId="state" catToolTip="true"
   catEnabled="true" catZoom="true"
   catSummary="popup/summary.swf" >
     <color colorNormal="0xFFFFFF" colorOver="0xFEDF74"
      colorPress="0xFECD25" colorText="0x000000" />
     <link lnkEvent=”onRelease” lnkTarget=”_JS”
      lnkUrl=”areaEvent” />
     <link lnkEvent=”onRollOver” lnkTarget=”_JS”
      lnkUrl=”areaEvent” />
  </areaCategory>
</areasCategory>
```

It contains the following XML tags:

Tags

<areasCategory>	The full set of area categories.
<areaCategory>	Each area category.

Each category contains a set of attributes defining all areas of this category. Here is a list of the attributes:

Attributes

catId	Unique category ID.
catToolTip (optional)	Display area tooltip (“true”) or do not (“false”). The tooltip displays the corresponding area’s label. The default value is “true”. (Note: Tooltips will not be shown if the summary pop-up for areas is used.)
catEnabled (optional)	Set if every area in this category is enabled (“true”) or disabled (“false”). The default value is “true” or enabled.
catZoom (optional)	Set if every area in this category will zoom-in when clicked (“true”) or will not (“false”). The default value is “true”.
catSummary (optional)	Set the flash pop-up to show if the areas of this category have summary information.

To display a pop-up window when hovering over this area, you have to specify the path of the pop-up flash movieclip file. This has to be done with the attribute ‘catSummary’. Note: You have to specify this path also in the theme.xml file by setting the attribute ‘areasSummary’ (see theme.xml in chapter 3.2.).

Within the categories of the areas you can set the colors applying to the areas of a category, as well as the events to be launched. This can be done by the following XML elements:

Tags	
<color>	Colors for the areas of this category.
<link>	Events for the areas of this category.

Having a closer look at these elements:

3.4.1. Color tag

Within the color tag you define the colors for all areas in a category. An example is given in the following:

```
<color colorNormal="0xFFFFFF" colorOver="0xFEDF74"
       colorPress="0xFECD25" colorText="0x000000" />
```

The attributes defining the color scheme will be explained in the following table:

Attributes	
colorNormal	Color of the area in normal state (mouse cursor is outside the area).
colorOver	Color of the area when hovering over the area.
colorPress	Color of the area when clicking the area.
colorText	Text color.

3.4.2. Link tag

The link tag lets you specify events assigned to the areas of a category. An example is given here:

```
<link lnkEvent=”onRelease” lnkTarget=”_blank”
lnkUrl=”http://www.flashmaps.com” />
<link lnkEvent=”onRollOver” lnkTarget=”_JS”
lnkUrl=”areaEvent” />
<link lnkEvent=”onRollOut” lnkTarget=”_JS”
lnkUrl=”areaEvent” />
```

The attributes defining an event are the following:

Attributes	
lnkEvent	Event when action will be launched. Can have the following values: - onRelease: invoked when clicking over an area. - onRollOver: invoked when hovering over an area. - onRollOut: invoked when the mouse leaves the area.
lnkTarget	Type of action to launch. Can have the following values:

	- _MC: load a movieclip. - _JS: call a JavaScript function. - _AS: call the fmFromASArea function in the Flash container (if any). - _blank, _self, _top: open a web page on a new browser window.
lnkUrl	Action to launch. The value depends on the type of action. - _MC: path to the movieclip to load. - _JS: name of the JavaScript function to call. The event and area's ID are sent as parameters. - _AS: Extra parameter. - _blank, _self, _top: web page's URL. The ID is sent as poi_id parameter.

You can define several actions for one event by inserting several <link> elements.

3.5. areasColors.xml

areasColors.xml lets you configure the colors for each area individually, in case you want different colors in areas of the same category.

Each color element has an area ID attribute ('areaId'), corresponding with the area ID in the areas.xml file (see chapter 3.3. areas.xml). An example of the areasColors.xml is given here:

```
<areasColor>
  <areaColor areaId="AL" colorNormal="0xFFE6E6"
    colorOver="0xFEDF74" colorPress="0xFECD25"
    colorText="0x000000" />
  <areaColor areaId="AZ" colorNormal="0xFFC8C8"
    colorOver="0xFEDF74" colorPress="0xFECD25"
    colorText="0x000000" />
  <areaColor areaId="AR" colorNormal="0xFFE6E6"
    colorOver="0xFEDF74" colorPress="0xFECD25"
    colorText="0x000000" />
  ……
</areasColor>
```

It includes the following XML elements:

Tags	
<areasColor>	Collection of area colors.

<areaColor>	Area color scheme tag.

Each color scheme has a set of attributes, which are the following:

Attributes

areaId	Unique area ID.
colorNormal	Color of the area in normal state (mouse cursor is outside the area).
colorOver	Color of the area when hovering over the area.
colorPress	Color of the area when clicking on the area.
colorText	Text color.

3.6. areasSummary.xml

You can display pop-up windows with customizable information when hovering over an area.

To do so, you have to specify the URL of the Flash movieclip file (swf) in the areasCategories.xml file (see 3.4. areasCategories.xml). The area-specific information to show must be defined here.

This XML file stores the additional information of the areas you want to show in the summary pop-up window. Note that you can include as much information as you like. For example:

```
<areasSummary>
  <areaSummary areaId="AL" population="4,447,100" />
  <areaSummary areaId="AK" population="626,932" />
  <areaSummary areaId="AZ" population="5,130,632" />
  <areaSummary areaId="AR" population="2,673,400" />
  ……
</areasSummary>
```

It includes the following XML elements:

Tags

<areasSummary>	Collection of area summaries.
<areaSummary>	Each area summary.

The only obligatory attribute is the area ID (‘areaID’).

Attributes

areaId	Area ID.

3.7. pois.xml

Use pois.xml to specify the collection of POI (Points of Interest) URL to be loaded.

Each POI has a unique ID ('poiID') and belongs to one category ('poiCategory') which is normally represented by a specific icon. Latitude and longitude coordinates ('poiLat' and 'poiLon') are needed as well in order to properly place the POIs on the map.

An example of a pois.xml file is given here:

```
<POIs>
  <POI poiId="1" poiLabel="Montgomery" poiCategory="city"
    poiLon="-86.27911" poiLat="32.361538" />
  <POI poiId="2" poiLabel="Phoenix" poiCategory="city"
    poiLon="-112.0763" poiLat="33.52837" />
  <POI poiId="3" poiLabel="Little Rock" poiCategory="city"
    poiLon="-92.33112" poiLat="34.736009" />
  <POI poiId="4" poiLabel="Sacramento" poiCategory="city"
    poiLon="-121.46892" poiLat="38.555605" />
  ……
</POIs>
```

It includes the following XML elements:

Tags

<POIs>	Collection of POIs.
<POI>	Each POI.

Each POI element contains the following attributes:

Attributes

poiId	Unique POI ID.
poiCategory	POI category.
poiLabel	POI label.
poiLon (numeric)	POI longitude (decimal degrees).
poiLat (numeric)	POI latitude (decimal degrees).

3.8. poisCategories.xml

POIs are grouped by categories. Each category has a set of properties and events which are applied to all the POIs inside their respective category. That information is inside poisCategories.xml.

The following is an example of a poisCategories.xml file:

```
<POIsCategory>
  <POICategory catId="city" catUrl="city"
   catEmbedded="true" catSummary="popup/city.swf">
    <link lnkEvent="onRelease" lnkTarget="_JS"
     lnkUrl="poiEvent" />
    <link lnkEvent="onRollOver" lnkTarget="_JS"
     lnkUrl="poiEvent" />
    <link lnkEvent="onRollOut" lnkTarget="_JS"
     lnkUrl="poiEvent" />
```

```
  </POICategory>
  ……
</POIsCategory>
```

It includes the following XML elements:

Tags

<POIsCategory>	Collection of POI categories.
<POICategory>	Each POI category.

Each POI category contains a set of attributes to be applied to every POI inside of its respective category. These attributes are as follows:

Attributes

catId	POI category ID.
catUrl	POI category's icon URL (JPG or SWF allowed) or name of the icon if from the library.
catEmbedded (optional)	Defines if the icon is in the icon library ("true") or it is a stand-alone file ("false"). The default value is "false".
catType (optional)	Type of icon to show if it is not in the icon library. Always "plain".
catWidth (optional)	Width of the icon if the icon is not in the icon library. The default value is "4".
catHeight (optional)	Height of the icon if the icon is not in the icon library. The default value is "4".
catSummary (optional)	Flash movieclip pop-up URL to show if POIs of this category have summary information.

If you want to display a summary pop-up window when you hover over a POI, you have to specify the URL in 'catSummary' attribute and also in the theme.xml file, as well as in the 'poisSummary' attribute.

Within the POI category you can customize events for user interaction through the followingXML link element:

Tags

<link>	POI event definition.

Here is a closer look at this element:

3.8.1. Link tag

The link tag allows specifying one or various events assigned to a POI category, as seen in the following example:

<link lnkEvent="onRelease" lnkTarget="_blank"

```
lnkUrl="http://www.flashmaps.com" />
<link lnkEvent="onRollOver" lnkTarget="_JS"
lnkUrl="poiEvent" />
<link lnkEvent="onRollOut" lnkTarget="_JS"
lnkUrl="poiEvent" />
```

The attributes defining an event are the following:

Attributes	
lnkEvent	Event when action will be launched. Can have the following values: - onRelease: invoked when clicking over an area. - onRollOver: invoked when hovering over an area. - onRollOut: invoked when the mouse leaves the area.
lnkTarget	Type of action to launch. Can have the following values: - _MC: pop-up a Flash movieclip. - _JS: call a JavaScript function. - _AS: call an AreaSelector internal function. - _blank, _self, _top: open a web page.
lnkUrl	Action to launch. The value depends of the type of the action. - _MC: path to the movieclip to load. - _JS: name of the JavaScript function to call. The event and area's ID are sent as parameters. - _AS: Action name to be launched. - _blank, _self, _top: web page's URL. The ID is sent as poi_id parameter.

You can also use as many events as you need to combine different event actions.

3.9. poisSummary.xml

You can display pop-up windows with customizable information when hovering over an area.

To do so, you have to specify the URL of the Flash movieclip file (swf) in the poisCategories.xml file (see 3.4. poisCategories.xml). The poi-specific information to be displayed, must be defined here.

This XML file stores the additional information of the POIs you want to display in the summary pop-up window. You can include all of the information you'd like, for example:

```
<POIsSummary>
  <POISummary poiId=”1" population="201.568" />
  <POISummary poiId=”2” population="1.321.045" />
  <POISummary poiId=”3” population="183.133" />
  <POISummary poiId="4" population="407.018" />
  ……
</POIsSummary>
```

It includes the following XML elements:

Tags

<POIsSummary>	Collection of POI summaries.
<POISummary>	Each POI summary.

The only obligatory attribute is the POI ID ('poiId').

Attributes

poiId	POI ID.

4. API reference

AreaSelector can be embedded into the HTML as an <object> or into a parent Flash movie clip (a flash container). This section contains the description of the functions available in the API (Application Programming Interface) for both environments.

4.1. JavaScript API reference

4.1.1. fmThemeLoad

Load a theme.

Function
```
fmThemeLoad(theme_str)
```
Parameter
```
theme_str (string)  XML URL containing the theme to load.
                    Can be an XML file or a dynamic page.
```
Example
```
Load the USA theme:
<a href="JavaScript:fmThemeLoad('USA.xml');">Load USA</a>
```

4.1.2. fmThemeReloadAreas

Reload a collection of areas.

Function
```
fmThemeReloadAreas(areas_xml)
```
Parameter
```
areas_xml (string)  XML URL containing the areas to reload.
                    Can be an XML-file or a dynamic page.
```
Example
```
Reload the areas of USA:
<a href="JavaScript:fmThemeReloadAreas('states.asp');">
Reload states</a>
```

4.1.3. fmInitialView

Display initial view.

Function
```
fmInitialView()
```
Example
```
Initial view
<a href="JavaScript:fmInitialView();">Initial view</a>
```

4.1.4. fmMapBackLevel

Go back one level. For example, if displaying one state with its counties, get back to country map.

Function
```
fmMapBackLevel()
```
Example
```
Get back one level:
<a href="JavaScript:fmMapBackLevel();">Back one level</a>
```

4.1.5. fmAreaCenter

Focus on a certain area in the current theme. Depending on the current level, AreaSelector focuses with or without the need to zoom.

Function
```
fmAreaCenter(area_str)
```
Parameter
```
area_str (string)   ID of the area to focus on.
```
Example
```
Focus on California:
<a href="JavaScript:fmAreaCenter('CA');">California</a>
```

4.1.6. fmAreaZoomIn

Dynamically zoom in a certain area running through higher levels.

Function
```
fmThemeReloadAreas(areas_array)
```
Parameter
```
areas_array         Array of area IDs to run through while
(array)             zooming in and center.
```
Example
```
Dinamycally zoom in from North America to California,
through United States:
<a href="JavaScript:fmAreaZoomIn(('NA','US','CA'));">
California</a>
```

4.1.7. fmAreaCenterLatLon

Focus on a point, inside a certain area, at a specific scale.

Function
```
fmAreaCenterLatLon(area_str, lat, lon, scale)
```
Parameter
```
area_str (string)   ID of the area to focus on.
lat (numeric)       Latitude in decimal degrees.
lon (numeric)       Longitude in decimal degrees.
scale (numeric)     Scale percentage. The higher value, the
                    closer view.
```
Example
```
Center over a point inside NY
<a href="JavaScript:fmAreaCenterLatLon('NY', 40.742272,
-73.987907, 2000);">New York Location</a>
```

4.1.8. fmAreaEnabled

Enable or disable specific areas. When areas are disabled, they can not be selected.

Function
```
fmAreaEnabled(area_str, enabled_str)
```
Parameter
```
area_str (string)   ID of the area to enable/disable.
enabled_str         Enable or disable area (true or false)
(string)
```
Example
```
Disable California:
<a href="JavaScript:fmAreaEnabled('CA', 'false');">
Disable California</a>
```

4.1.9. fmAreaColor

Change the colors of a specific area.

Function
```
fmAreaColor(area_str, colorNormal, colorOver, colorPress,
colorText)
```
Parameter
```
area_str (string)   ID of the area to change colors.
colorNormal         Area color in normal state (mouse
(string)            pointer outside the area).
colorOver (string)  Area color when hovering over it.
colorPress          Area color when clicking it.
(string)
colorText (string)  Text color.
```
Example
```
Set new colors to Oregon:
<a href="JavaScript:fmAreaColor('OR', '0x4a6e93',
'0xFF0000', '0xFFF000', '0x000000');">Color Oregon</a>
```

4.1.10. fmMapModeZoom

Set AreaSelector to zoom mode. When a user selects an area, the system zooms in on it. This is the default mode, howeveryou normally want to get back to it after using the Select Mode.

Function
```
fmMapModeZoom()
```
Example
```
Enter zoom mode:
<a href="JavaScript:fmMapModeZoom();">Mode Zoom</a>
```

4.1.11. fmMapModeSelect

Set AreaSelector to selection mode. When a user selects an area, it is either selected or deselected.

Function
```
fmMapModeSelect()
```
Example
```
Enter select mode:
<a href="JavaScript:fmMapModeSelect();">Mode Select</a>
```

4.1.12. fmMapModeExportListAreas

Get a list of selected areas, sorted by levels.

Function
```
fmMapModeExportListAreas():String
```
Return
```
areas_str (string)  List of selected areas.
```
Example
```
Get the list of selected areas:
<a
href="JavaScript:alert(fmMapModeExportListAreas());">Get
Selected Areas</a>
```

4.1.13. fmMapModeCleanAreas

Clear selected areas.

Function
```
fmMapModeCleanAreas()
```
Example
```
Clear selected areas:
<a href="JavaScript:fmMapModeCleanAreas();">Clear</a>
```

4.1.14. fmPOIsShowCategory

Show POIs pertaining to one or every category.

Function
```
fmPOIsShowCategory(category_str)
```
Parameter
```
category_str        ID of the category to show, or "*" for
(string)            every one.
```
Example
```
Show all categories
<a href="JavaScript:fmPOIsShowCategory('*');">Show all</a>
```

4.1.15. fmPOIsHideCategory

Hide POIs pertaining to one or every category.

Function
```
fmPOIsHideCategory(category_str)
```
Parameter
```
category_str        ID of the category to hide, or "*" for
(string)            every one.
```
Example

```
Hide hotels:
<a href="JavaScript:fmPOIsShowCategory('hotel');">
Hide Hotels</a>
```

4.1.16.fmPOIAddEvent

Execute a concrete event.

Function

```
fmPOIAddEvent(event_str, target_str, url_str, id_str)
```

Parameter

event_str (string)	Event when action will be launched. Can have the following values: - onRelease: invoked when clicking over an area. - onRollOver: invoked when hovering over an area. - onRollOut: invoked when the mouse leaves the area.
target_str (string)	Type of action to launch. Can have the following values: - _MC: pop-up a Flash movieclip. - _JS: call a JavaScript function. - _AS: call an AreaSelector. internal function. - _blank, _self, _top: open a web page.
url_str (string)	Action to launch. The value depends on the type of action. - _MC: path to the movieclip to load. - _JS: name of the JavaScript function to call. The event and area's ID are sent as parameters. - _AS: Action name to be launched. - _blank, _self, _top: web page's URL. The ID is sent as poi_id parameter.
Id_str (string)	POI ID.

Example

```
Open a popup
<a href="JavaScript:fmPOIAddEvent('onRelease','_MC',
'fmPOI.swf', '12');">Show popup</a>
```

4.1.17.fmShowAlert

Show an alert window with a custom message over AreaSelector.

Function

```
fmShowAlert(title_str, text_str)
```

Parameter

title_str (string)	Title of the alert window.
text_str (string)	Text inside the alert window.

Example

```
Show alert Window:
<a href="JavaScript:fmShowAlert('Error','Please select a
state');">Show Alert</a>
```

4.1.18.fmHideAlert

Hide the alert window.

Function
```
fmHideAlert()
```
Example
```
Hide alert window:
<a href="JavaScript:fmHideAlert();">Hide Alert</a>
```

4.2. Flash API reference

4.2.1. fmThemeLoad

Load a theme.

Function
```
fmThemeLoad(theme_str)
```
Parameter
```
theme_str (string)  XML URL containing the theme to load.
                    Can be an XML file or a dynamic page.
```
Example
```
container_mc.fmThemeLoad("fmUSA.xml");
```

4.2.2. fmThemeReloadAreas

Reload a collection of areas.

Function
```
fmThemeReloadAreas(areas_str)
```
Parameter
```
Areas_str (string)  XML URL containing the areas to load.
                    Can be an XML file or a dynamic page.
```
Example
```
container_mc.fmThemeReloadAreas("fmStates.asp");
```

4.2.3. fmInitialView

Display initial view.

Function
```
fmInitialView()
```
Example
```
container_mc.fmInitialView();
```

4.2.4. fmMapBackLevel

Go back one level. For example, if displaying a state with its counties, get back to country map.

Function
```
fmMapBackLevel()
```
Example
```
container_mc.fmMapBackLevel();
```

4.2.5. fmAreaCenter

Focus on a certain area in the current theme. Depending on the current view level, AreaSelector focuses with or without the need to zoom.

Function
```
fmAreaCenter(area_str)
```
Parameter
```
area_str (string)   ID of the area to focus on.
```
Example
```
container_mc.fmAreaCenter("CA");
```

4.2.6. fmAreaZoomIn

Dynamically zoom on a certain area running through higher levels.

Function
```
fmAreaZoomIn(areas_array)
```
Parameter
```
areas_array         Array of area IDs to run through while
(array)             zooming in and center.
```
Example
```
var areas_array = new Array("NA", "US", "CA");
container_mc.fmAreaZoomIn(areas_array);
```

4.2.7. fmAreaCenterLatLon

Focus on a point inside a certain area at a specific scale.

Function
```
fmAreaCenterLatLon(area_str, lat, lon, scale)
```
Parameter
```
area_str (string)   ID of the area to focus on.
lat (number)        Latitude in decimal degrees.
lon (number)        Longitude in decimal degrees.
scale (number)      Scale percentage. The higher value, the
                    closer view.
```
Example
```
container_mc.fmAreaCenterLatLon("NY", 40.7422, -73.987,
2000);
```

4.2.8. fmAreaEnabled

Enable or disable a specific area. When disabled, areas are not selectable.

Function
```
fmAreaEnabled(area_str, bEnabled)
```
Parameter
```
Area_str (string)   ID of the area to enable/disable.
bEnabled (boolean)  Enable or disable area (true or false)
```
Example
```
container_mc.fmAreaEnabled("CA", false);
```

4.2.9. fmAreaColor

Change the colors of a specific area.

Function
```
fmAreaColor(area_str, colorUp, colorOver, colorDown,
colorText)
```
Parameter
```
area_str (string)   ID of the area.
colorUp (string)    Area color in normal state (mouse
                    pointer outside the area).
colorOver (string)  Area color when hovering over it.
colorDown (string)  Area color when clicking it.
colorText (string)  Text color.
```
Example
```
container_mc.fmAreaColor("OR", "0x4a6e93", "0xFF0000",
"0xFFF000", "0x000000");
```

4.2.10. fmMapModeActivate

Set zoom or selection mode.

Function
```
fmMapModeActivate(mode_str)
```
Parameter
```
mode_str (string)   Mode to set. The options are:
                       - "zoom": activate zoom mode.
                       - "select": activate select mode.
```
Example
```
container_mc.fmMapModeActivate("select");
```

4.2.11. fmMapModeExport

Get the list of selected areas sorted by level.

Function
```
fmMapModeExport():string
```
Return
```
(string)            List of selected areas.
```
Example

```
var areas_str = container_mc.fmMapModeExport();
```

4.2.12.fmMapModeClean

Clear selected areas.

Function
```
fmMapModeClean()
```
Example
```
container_mc.fmMapModeClean();
```

4.2.13.fmPOIsShowCategory

Show POIs pertaining to one or every category.

Function
```
fmPOIsShowCategory(category_str)
```
Parameter
```
category_str        ID of the category to show or * for
(string)            every one.
```
Example
```
container_mc.fmPOIsShowCategory(“hotel”);
container_mc.fmPOIsShowCategory(“*”);
```

4.2.14.fmPOIsHideCategory

Hide POIs pertaining to one or every category.

Function
```
fmPOIsHideCategory(category_str)
```
Parameter
```
category_str        ID of the category to hide or “*” for
(string)            every one.
```
Example
```
container_mc.fmPOIsHideCategory(“hotel”);
container_mc.fmPOIsHideCategory(“*”);
```

4.2.15.fmPOIAddEvent

Execute a specific event.

Function
```
fmPOIAddEvent(event_str, target_str, url_str, id_str)
```
Parameter
```
event_str (string)  Event when action will be launched. Can
                    have the following values:
                       - onRelease: invoked when clicking
                         over an area.
                       - onRollOver: invoked when hovering
                         over an area.
                       - onRollOut: invoked when the mouse
                         leaves the area.
target_str          Type of action to launch. Can have the
```

```
(string)            following values:
                       - _MC: pop-up a Flash movieclip.
                       - _JS: call a JavaScript function.
                       - _AS: call an AreaSelector.
                         internal function.
                       - _blank, _self, _top: open a web
                         page.
url_str (string)    Action to launch. The value depends on
                    the type of action.
                       - _MC: path to the movieclip to
                         load.
                       - _JS: name of the JavaScript
                         function to call. The event and
                         area's ID are sent as parameters.
                       - _AS: Action name to be launched.
                       - _blank, _self, _top: web page's
                         URL. The ID is sent as poi_id
                         parameter.
Id_str (string)     POI ID.
```

Example

```
container_mc.fmPOIAddEvent("onRelease", "_blank",
"http://www.flashmaps.com", "12");
```

4.2.16. fmShowAlert

Show an alert window with a custom message over AreaSelector.

Function

```
fmShowAlert(title_str, text_str)
```

Parameter

```
title_str (string)  Title of the alert window.
text_str (string)   Text inside the alert window.
```

Example

```
container_mc.fmShowAlert("Error", "Error loading xml.");
```

4.2.17. fmHideAlert

Hide the alert window.

Function

```
fmHideAlert()
```

Example

```
container_mc.fmHideAlert();
```

4.2.18. fmFromASPOIsLoaded

This function is called after a set of POIs has been loaded.

Event

```
fmFromASPOIsLoaded()
```

Example

```
container_mc.fmHideAlert();
```

4.2.19.fmFromASArea

This function is called when a zoom in, zoom out or new level load are executed.

It is also called when an area event type “_AS” is launched.

Function
```
fmFromASArea(event_str, id_str)
```
Parameter
```
event_str (string)  Event.
Id_str (string)     Area ID.
```

4.2.20.fmFromASPOI

This function is called when a POI event type “_AS” is launched.

Function
```
fmFromASPOI(event_str, id_str)
```
Parameter
```
event_str (string)  Event.
Id_str (string)     POI ID.
```

5. Object reference

AreaSelector object can receive a set of parameters when referenced from within the HTML.

5.1. Load an initial theme

Set the initial them.

Parameter

```
customTheme (string)  Theme XML URL.
```

Example

```
<param name=”movie”
value=”fmASToolbar.swf?customTheme=fmUSA.xml” />
```

5.2. Focus on an initial area

Focus initially on a certain area.

Parameter

```
customArea (string)   Area ID.
```

Example

```
<param name=”movie” value=”fmASToolbar.swf?customArea=CA”
/>
```

6. How to

6.1. Insert AreaSelector into your HTML

This is the way you should insert your AreaSelector Flash object inside your HTML code:

```
<object id="fmASEngine" width="500" height="300"
viewastext
   type="application/x-shockwave-flash" BASE=”.”
   data="fmASMap/fmASToolbar.swf">
   <param name="movie" value="fmASMap/fmASToolbar.swf" />
   <param name="BASE" value="." />
   <param name="quality" value="high" />
   <param name="play" value="true" />
   <param name="bgcolor" value="#FFFFFF" />
   <param name="allowScriptAccess" value="sameDomain" />
   <param name="menu" value="false" />
</object>
```

Note that the movie clip URL (`fmASMap/fmASToolbar.swf`) may be changed to send parameters (`like customTheme, etc.`).

Be sure to include those changes both in the data attribute, and in the movie parameter tag.

6.2. Insert AreaSelector into a Flash container

If you add the AreaSelector into a Flash container, you should change the parameter fmASMcPath in the JavaScript initial variables library (fmASInitVars.js) with the name of the AreaSelector container movie clip.

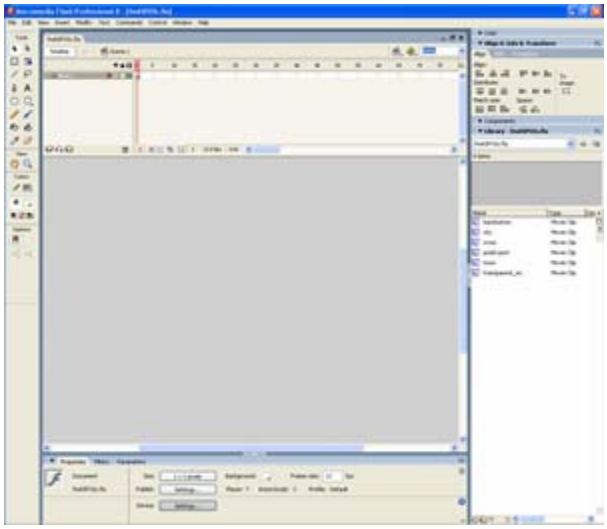
```
Parameter
fmASMcPath            AreaSelector container movieclip.
Example
fmASMcPath = “container_mc.”
```

6.3. Insert or modify an icon in the library

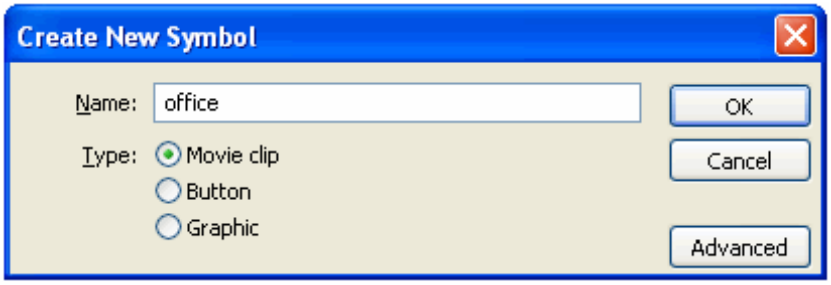
To add or modify POI icons in the icon library, you must edit the fmASIconLibrary.fla with Macromedia Flash 8.

The steps to add an icon are the following:

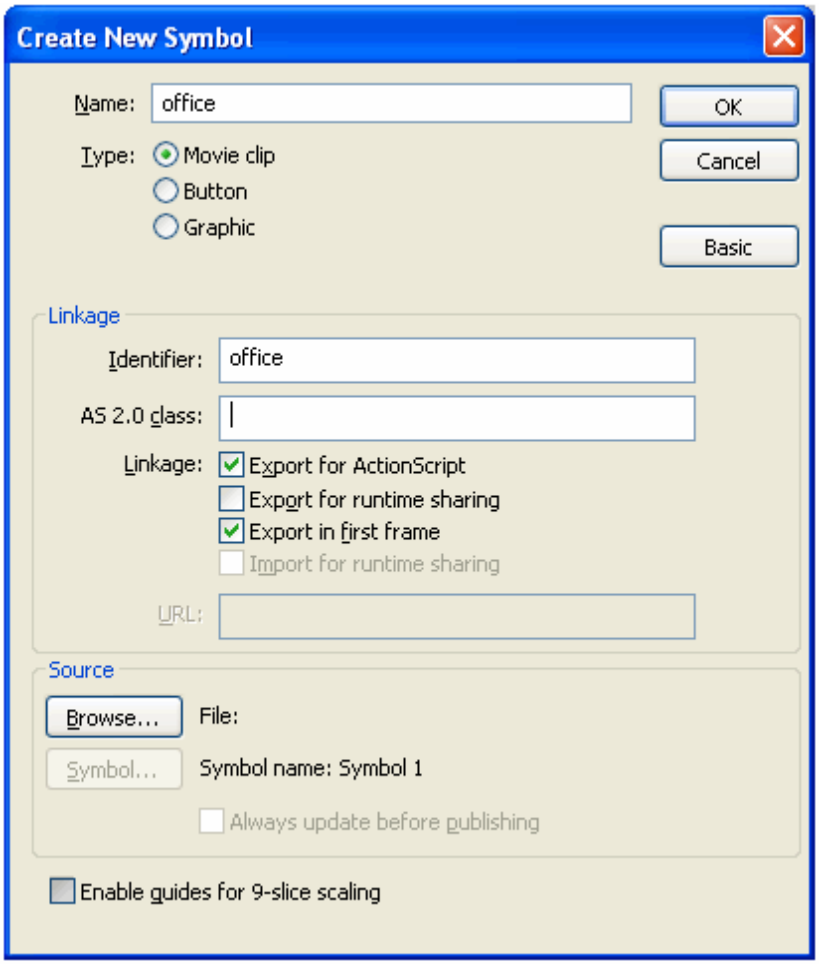
- Open fmASIconLibrary.fla with Macromedia Flash 8 editor.



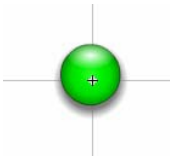
- Insert a new symbol (Menu Insert → New Symbol) and name the new icon. Select the type: Movie clip.



- Press the Advanced button and check “Export for Action Script” and “Export in first frame” boxes. Put the same icon name in the Identifier textbox.



- Draw the icon in this area. Note that the icon will be centered in the position 0, 0, where the cross is displayed.



- Once you compile your movie clip (Ctrl + Enter) your new icon will be inserted in your icon library, and ready to use. Feel free to add as much icons as you need.