Tutorial for custom plugin development for ALTUI

Alexis Mermet, Sept 4th 2015

ALTUI has a plugin architecture enabling dynamic addition of javascript modules in order to customize the drawing of several area

1. The device dashboard on the device page
2. The device icon display on the device page
3. A custom tab in the device control panel page
4. Some css styles specific for this device

All these are optional; you only define and code what you want to overide. ALTUI allways provide a default implementation. Especially for ICONS, by default ALTUI is implementing the same rules as UI5 and UI7 using device json descriptions, but it can do some more fine grained/sophisticated icon selection using javascript code to override the default device icon display behavior. An example is given below.

I am detailing here after what the plugin author must do :

1. Declare the plugin files and object.method he has written into the ALTUI vera device so that the code is dynamically loaded.

2. Writes the code/methods using ALTUI apis and CSS/Names convention

Declaration

ALTUI VERA device has a configuration variable which is a JSON object of a hashtable indexed by the device type and containing the various object.method javascript name.

There are 3 ways to declare a plugin :

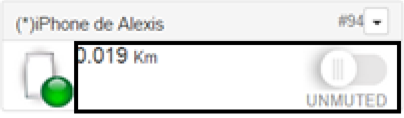
- Manually by editing the ALTUI device « PluginConfig » variable by hand or using the settings screen of the VERA device. It must be a valid JSON syntax so it is recommended to use a json validator
- Automatically, by the default configuration of ALTUI which is forced every time a new version of the L_ALTUI.lua file is distributed or when the user presses the “Default configuration” button on the settings tab of the device
- Dynamically , by calling the UPNP method of hte ALTUI device. Cf action « RegisterPlugin » in the S_ALTUI.xml service description.

The configuration variable example in JSON is :

```
{
  "urn:schemas-micasaverde-com:device:PowerMeter:1": {
    "DeviceDrawFunc": "ALTUI_PluginDisplays.drawPowerMeter",
    "ScriptFile": "J_ALTUI_plugins.js"
  },
  "urn:schemas-rts-services-com:device:ProgramLogicEG:1": {
    "DeviceDrawFunc": "ALTUI_PluginDisplays.drawPLEG",
    "ScriptFile": "J_ALTUI_plugins.js"
  },
  "urn:schemas-upnp-org:device:IPX800:1": {
    "DeviceDrawFunc": "ALTUI_IPhoneLocator.drawIPX",
    "ScriptFile": "J_ALTUI_iphone.js"
  },
  "urn:schemas-micasaverde-com:device:SmokeSensor:1": {
    "DeviceDrawFunc": "ALTUI_PluginDisplays.drawSmoke",
    "ScriptFile": "J_ALTUI_plugins.js"
  },
  "urn:schemas-a-lurker-com:device:InfoViewer:1": {
    "DeviceDrawFunc": "ALTUI_PluginDisplays.drawInfoViewer",
    "ScriptFile": "J_ALTUI_plugins.js"
  },
  "urn:schemas-upnp-org:device:IPhoneLocator:1": {
    "StyleFunc": "ALTUI_IPhoneLocator.getStyle",
    "DeviceDrawFunc": "ALTUI_IPhoneLocator.drawIPhone",
    "ScriptFile": "J_ALTUI_iphone.js"
  }
}
```

NOTE

- functions are using the javascript module pattern so are like « objectname ». « method »
- ScriptFiles J_xx.js must be in the same folder as ALTUI device files on the Vera

	Purpose	Input parameters	Output
DeviceDrawFunc	Display the device small dashboard on the Device page	• device : a ALTUI device object. device.altuiid is the unique ID for ALTUI and is needed to uniquely identify a device , even in the case of a multicontroller setup.	A string, in HTML format for the display of the content of the small dashboard panel as shown here in the black box. ALTUI is drawing everything else. 
StyleFunc	Register some specific CSS style for this device drawing	none	Returns a CSS valid string. See ALTUI_PluginDisplays.getStyle() for example.
ControlPanelFunc	Display a full custom device control panel page	(device, domparent) • device : the ALTUI device • domparent ; the jquery HTML DOM object	No output. The function is supposed to insert the control panel HTML content into the domparent. For instance , using jquery : $(domparent).append(html);
DeviceIconFunc	Display a device icon	• device : a ALTUI device object.	A HTML string containing the <img></img> object to display. Only the yellow part should change, the rest should be kept as in this example. Example "<img class='altui-device-icon pull-left img-rounded' src='"+newsrc+"' alt='"+conditionGroup+"' onerror='UIManager.onDeviceIconError(\""+ device.altuiid+"\")' ></img>"

Development Rules

- ALTUI apis
  - Use only the MultiBox object public api, it should cover almost all possible actions on a box and supports multiple box
  - **do not query devices by yourself using user_data** or equivalent, use MutliBox.getDevices() & others apis
  - device , scene, room etc are object, similar to what MIOS user_data provides, with a few extra ALTUI internal fields like altuiid
  - device.id is a single box identificator, so most of the time, MultiBox object uses device.altuiid which is a universally unique id accross the multiple boxes , or the device object itself

- Development framework
  - Use Jquery / JqueryUI which are already loaded by ALTUI web page.
  - Use Bootstrap, especially its grid system (row, col-xx-nn) and css classes to insure the responsiveness (auto adjustment to client device screen size) of the user interface
  - You can debug easily using chrome or firefox javascript debugger
  - It is recommended to have //# sourceURL=xxxxxxxx.js (where xxxxx is the file name) at the first line of the source file such that file names appear in the debugger

Step by Step example for the virtual switch

- Step1 : declaring the plugin
  - We are going to use the default configuration method, so we go into L_ALTUI.lua and search for getDefaultConfig(). The tbl object here is the lua version of the ALTUI configuration table.
  - We are going to reuse the J_ALTUI_plugins.js file but one could use a specific new file
  - We add the following entry in the table

```
        tbl["urn:schemas-upnp-org:device:VSwitch:1"]= {
                ["ScriptFile"]="J_ALTUI_plugins.js",
                ["DeviceDrawFunc"]="ALTUI_PluginDisplays.drawVswitch",
        }
```

- Step2: implement the drawVswitch method of the object ALTUI_PluginDisplays
  - The drawVswitch is a public method of the ALTUI_PluginDisplays object.
  - Using MultiBox.getStatus() api, we get the current status of the vswitch device and the two lines of text kept in the vswitch device variable Text1, Text2
  - Using a helper method of ALTUI_PluginDisplays, we create a OnOff button with a htmlid of “altui-vswitch-<devicealtuiid>”
  - We add a <script> object in the html page to handle the click event on the onoff button. touchend is an event for iPad/touch screen devices. When a click is received, we use the `ALTUI_PluginDisplays.toggleVswitch()` method to toggle the vswitch device status. `ALTUI_PluginDisplays.toggleButton(altuiid, htmlid,service,variable,callback) is a helper method`
    - `Altuiid : the altuiid of the device`
    - `Htmlid : the htmlid of the button`
    - `Service, variable : the variable of the VERA device which has the state of the button`
    - `Callback (id,newval) : a callback function which receives the complementary value for the button ( OFF if it was ON , ON if it was OFF )`
  - In the `ALTUI_PluginDisplays.toggleButton` callback function, we call `MultiBox.runActionByAltuiID( altuiid, service, actionname, action parameters table )` to change the vswitch button status.
  -

```
var ALTUI_PluginDisplays= ( function( window, undefined ) {


function _drawVswitch( device ) {
        var html ="";
        var status = parseInt(MultiBox.getStatus( device, 'urn:upnp-org:serviceId:VSwitch1', 'Status' ));
        html += ALTUI_PluginDisplays.createOnOffButton( status,"altui-vswitch-"+device.altuiid,
_T("OFF,ON") , "pull-right");
        $.each( ['Text1','Text2'],function(i,v) {
                var dl1 = MultiBox.getStatus( device, 'urn:upnp-org:serviceId:VSwitch1', v );
                if (dl1 != null)
                        html += $("<div class='altui-vswitch-text'></div>").text(dl1).wrap( "<div></div>"
).parent().html()
        });
        // on off
        html += "<script type='text/javascript'>";
        html += "$('div#altui-vswitch-{0}').on('click touchend', function() {
ALTUI_PluginDisplays.toggleVswitch('{0}','div#altui-vswitch-{0}'); } );".format(device.altuiid);
        html += "</script>";
        return html;
}
```

```
  // explicitly return public methods when this object is instantiated
  return {
        //----------------------------------------------------------
        // PUBLIC  functions
        //----------------------------------------------------------
        drawVswitch         : _drawVswitch,
        toggleVswitch: function (altuiid, htmlid) {
                ALTUI_PluginDisplays.toggleButton(altuiid, htmlid, 'urn:upnp-org:serviceId:VSwitch1',
'Status', function(id,newval) {
                        MultiBox.runActionByAltuiID( altuiid, 'urn:upnp-org:serviceId:VSwitch1',
'SetTarget', {newTargetValue:newval} );
                });
        },
  };
})( window );
```

Other examples

- Styles
    - If your device code needs some CSS style to be dynamically loaded in the page to support your display code, you can implement the “`StyleFunc`” function. An example is given in J_ALTUI_plugins.js with the getStyle() function.

```
function _getStyle() {
                var style="";
                style += ".altui-watts, .altui-volts, .altui-dimmable, .altui-countdown 
{font-size: 16px;}";
                style += ".altui-temperature  {font-size: 16px;}";
                style += ".altui-humidity, .altui-light  {font-size: 18px;}";
                style += ".altui-motion {font-size: 22px;}";
                style += ".altui-weather-text, .altui-lasttrip-text, .altui-vswitch-text 
{font-size: 11px;}";
                style += ".altui-red { color:red;}";
                style += ".altui-blue { color:blue;}";
                style += ".altui-orange { color:darkorange;}";
                style += ".altui-magenta { color:magenta;}";
                style += ".altui-multiswitch-container { position:absolute; left:58px; 
right:16px; } .altui-multiswitch-container .row { padding-top:1px; padding-bottom:1px; 
margin-left:0px; margin-right:0px;} .altui-multiswitch-container .col-xs-3 { padding-
left:1px; padding-right:1px; }  .altui-multiswitch-open { white-space: nowrap; overflow: 
hidden; text-overflow: ellipsis; padding-left:0px; padding-right:0px; margin-left:0px; 
margin-right:0px; width: 100%; max-width: 100% }";
                style += ".altui-heater-container { position:absolute; left:71px; 
right:16px; } .altui-heater-container .row { padding-top:1px; padding-bottom:1px; margin-
left:0px; margin-right:0px;} .altui-heater-container .col-xs-3 { padding-left:1px; 
padding-right:1px; text-align:center;}  .altui-heater-btn { white-space: nowrap; overflow: 
hidden; text-overflow: ellipsis; padding-left:0px; padding-right:0px; margin-left:0px; 
margin-right:0px; width: 100%; max-width: 100% }";
                style += ".altui-heater-container select.input-sm { height:22px; 
padding:0;}";
                style += ".altui-cyan { color:cyan;}";
                style += ".altui-dimmable-slider { margin-left: 60px; }";
                style += ".altui-infoviewer-log,.altui-window-btn,.altui-datamine-open { 
margin-top: 10px; }";
                style += "div.altui-windowcover button.btn-sm { width: 4em; }";
                return style;
}
```

- Control panel
    - The _drawCanaplusControlPanel(device, domparent) in J_ALTUI_iphone.js shows an example of a complete custom tab for a device. This device is a SAT TV set box and provides a kind of remote user interface.
    - Note how this function injects directly in the domparent the html layout of the user interface using the code

```
$(domparent).append(html);
```

    - Note how this function intercepts the click events on the button and trigger the device actions using the code

```
        $(".altui-cplus-button").click( function() {
                var id = $(this).prop('id');
                MultiBox.runAction( device, 'urn:upnp-org:serviceId:cplus1', 'SendKey', 
{keyStream:id} );
        });
```

- Device Icon
    - The _drawWeatherIcon (device) in J_ALTUI_plugins.js shows an example of a dynamic icon selection by the plugin code.

```
function _drawWeatherIcon( device) {
        var html ="";
        var conditionGroup = MultiBox.getStatus( device, 'urn:upnp-micasaverde-
com:serviceId:Weather1', 'ConditionGroup');
        var newsrc = (conditionGroup!=null) ?
"http://icons.wxug.com/i/c/i/"+conditionGroup+".gif" : defaultIconSrc;
        return "<img class='altui-device-icon pull-left img-rounded' src='"+newsrc+"'
alt='"+conditionGroup+"' onerror='UIManager.onDeviceIconError(\""+device.altuiid+"\")'
></img>";
};
```