

APX UI

New UI

1. Footprint reduction
2. Multi-theme(run-time) support.
3. Ease of rebranding – logos etc. should be changeable without changing backend or front end code.

The Basics from UI's point of view

1. JavaScript – which makes the browser work like an application.
2. jQuery – a JS library. Simpler, faster and more powerful.
3. AJAX – technology that allows JS functions to request information from the server.

APX UI – Communication with Backend

- The backend uses CGIs that communicate with the frontend using predefined XMLs.
- Getting the Configuration –
HTTP GET is used to get configuration information specific to a module in an XML format.

Request:

```
GET /cgi/xml_action.cgi?method=get&module=duster&file=wlan_basic_settings
HTTP/1.1
```

Response for this would be:

```
<?xml version="1.0" encoding="US-ASCII" ?>
<RGW>
  <wlan_settings>
    <mac>00:50:43:20:cb:2a</mac>
    <channel>0</channel>
    <bandwidth></bandwidth>
  </wlan_settings>
</RGW>
```

APX UI – Communication with Backend

- Changing the configuration

HTTP POST is used to set/change the configuration.

Request:

```
POST /cgi/xml_action.cgi?method=set&module=duster&file=wlan_basic_settings
HTTP/1.1
```

Response for this would be:

```
<?xml version="1.0" encoding="US-ASCII" ?>
<RGW>
  <wlan_settings>
    <channel>2</channel>
    <bandwidth>2</bandwidth>
  </wlan_settings>
</RGW>
```

APX UI –Backend Interface

- All the XML GET/POST are forwarded by the webserver(mongoose) to the CGI task. The CGI task invoke the duster_parse to dispatch the XML GET/POST request to the back-end modules.
- The XML document hierarchy mirrors the hierarchy maintained by the persistent storage manager(PSM)
- For get request will retrivers variables from PSM and returns values in an XML format and invoke registered GET API.
- For set request will save the value to PSM and invoke registered POST API.

APX UI - Overview

The screenshot displays the APX UI interface. At the top left is the Marvell logo. A navigation bar contains tabs: Dashboard, Internet, Home Network (selected), Wireless, Storage, and Router. A 'Quick Links' section in the top right contains 'Welcome', 'Quick Setup | Help | Log Out'. A left sidebar lists menu items: DHCP Settings (highlighted), Connected Devices, Network Activity, Access Control, Firewall Settings, and Application and Gaming. The main content area shows 'DHCP Settings' with fields for DHCP Range (192.168.1.X), Router LAN IP (192.168.1.1), DHCP Server (Enabled), DHCP Start Address (192.168.1.100), DHCP End Address (192.168.1.200), and DHCP Lease Time (86400 seconds). A 'Save' button is at the bottom right. Annotations include 'The Main TABS' pointing to the navigation bar, 'Quick Links' pointing to the top right, 'The Menu Items within a Tab.' pointing to the sidebar, and 'The Panel working Area' pointing to the DHCP settings content.

The Main TABS

Quick Links

Welcome
Quick Setup | Help | Log Out

Dashboard | Internet | **Home Network** | Wireless | Storage | Router

DHCP Settings

Connected Devices

Network Activity

Access Control

Firewall Settings

Application and Gaming

DHCP Settings

DHCP Range: 192.168.1.X

Router LAN IP: 192.168.1.1

DHCP Server: Enabled Disabled

DHCP Start Address: 192.168.1.100

DHCP End Address: 192.168.1.200

DHCP Lease Time: 86400 (in Seconds)

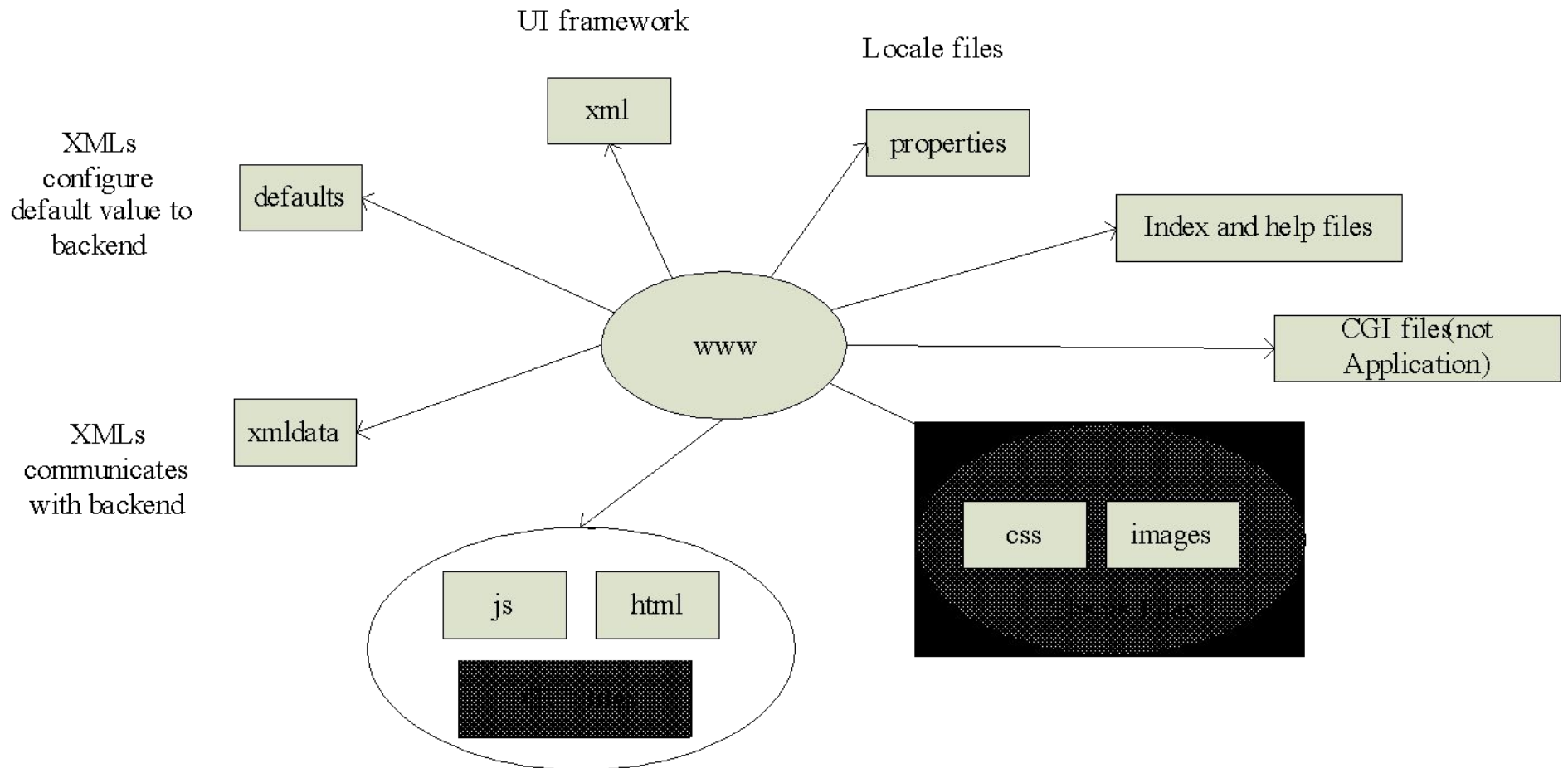
The Panel working Area

The Menu Items within a Tab.

Common Footer

Copyright 2010, All Rights Reserved, Marvell International Ltd. **MARVELL** APX v1.4.44

APX UI- Directory Structure



APX UI Key Files

1. www/xml/ui.xml

Tab definitions. Top level user interface layout.

2. www/index.html

New javascript files need to be included in the index.html to load by the client. HTML files are loaded by the javascript files when the javascript is invoked.

3. www/properties/Messages_<locale>.properties

Locale(e.g en) can be used for development. New property strings need to be translated and added to other supported locales as well.

4. www/js/panel/<newapp>/<function>.js

The source code layout convention is to put javascript code for each application in a separate directory with a separate javascript file for each major feature.

5. www/html/<newapp>/<function>.html

Any static html markup associated with the tan panel needs to be created.

6. www/help_<locale>.html

Add sections to the help file for each new application feature. It need translated and added to each locale specific help files.

APX UI - Sample Screen

1. Entry in App Framework(ui.xml)

```
</Tab>
<Tab Name='tHome_Network' type='submenupresent'>
  <Menus>
    <Menu id='mDHCP_Settings' implFunction='objDHCP_Settings' xmlName='lan' />
    <Menu id='mConnected_Devices' implFunction='objConnectedDev' xmlName='device_management_all' />
    <Menu id='mAccess_Logs' implFunction='objAccess_Logs' xmlName='detailed_log' />
    <Menu id='mAccess_Control' implFunction='objAccess_Control' xmlName='internet_access_control' />
    <Menu id='mFirewall_Settings' implFunction='objFirewall_Settings' xmlName='firewall' />
    <Menu id='mApp_Gaming' implFunction='objApp_Gaming' xmlName='port_forward_trigger' />
  </Menus>
</Tab>
<Tab Name='tWireless' type='submenupresent'>
  <Menus>
    <Menu id='mWire_Set' implFunction='objWire_Set' xmlName='wlan_basic_settings' />
  </Menus>
</Tab>
```

Annotations in the code block:

- A red circle highlights the `id` attribute in the first menu element, with an arrow pointing to a box labeled "Resource ID".
- A red circle highlights the `implFunction` attribute in the first menu element, with an arrow pointing to a box labeled "jquery class which implements the panel".
- A red circle highlights the `xmlName` attribute in the first menu element, with an arrow pointing to a box labeled "name of the xml".

2. The HTML file of the module
3. JS file – the jQuery class implementation.
4. Entry in properties file.

APX UI – Additional Stuff

* Locales Support

1. Properties file corresponding to each locale
2. One place modification for changing label text, popup text etc.

* Multiple Themes

1. Changing the theme doesn't require any code change in HTML or in JS
2. "css" and "images" folders corresponding to each theme
3. The "theme" duster module for communicating to the backend the change of theme.

APX UI – Adding a module

- 1) Depending on where you want to include the new panel, add an entry in ui.xml inside the corresponding “Tab” and “Menus” tags.
- 2) Create the html file in specific folder within the html directory.
- 3) Create the js file in specific folder.
Responsibilities of each js class ->
 1. Implement the onLoad fun that populates the DOM elements as per the XML fetched.
 2. Localization, validation etc as per the screen requirement.
 3. Implement the onPost which fetches modified field values and invokes the base API to send data to backend.
 4. Implement the setXMLName fun which is called by the
- 4) Include all the label texts, button texts etc. in the properties files of both the locales.
- 5) Don't forget to include the newly created js file in index.html using script tag.

APX UI – Adding a module

- * Commonly required APIs in the jQuery code –
 - `getData(xmlName)` – invokes the AJAX call for a GET from backend on the specified xml name.
 - `lableLocalization()`, `buttonLocalization()` – pass an array of elements returned from `document.getElementById`
 - `putMapElement(2-D array, index, path-of-xml-node, value-of-that-node)`
- * The jQuery class must implement the `loadHTML` function. In it the `callProductHTML` API needs to be used to specify the html file corresponding to this particular panel.

APX UI – Dashboard

The screenshot displays the APX UI Dashboard with the following components and annotations:

- Quick Links:** Located at the top right, containing "Welcome", "Quick Setup", "Help", and "Log Out".
- Main Status Indicators:** A central navigation bar with tabs for "Dashboard", "Internet", "Home Network", "Wireless", "Storage", and "Router". Below this is a status bar with icons for Internet (cloud), Router (server rack), and Home Network (laptop), each with a corresponding status indicator (green, orange, and red respectively).
- Module Info Widget:** A callout box pointing to the "Internet Connection" section in the Internet module.
- Status Indicator:** A callout box pointing to a red status indicator in the "Connected Devices" section of the Home Network module.
- Common Footer:** Located at the bottom, containing "Copyright 2010, All Rights Reserved, Marvell International Ltd.", the "MARVELL" logo, and "APX.v1.4.44".

Module	Section	Item	Status	
Internet	Internet Connection	WAN Link Status	Connected (Green)	
		Connection Type	DHCP	
		IP Address	10.31.130.29	
		Default Gateway	10.31.130.1	
Internet	Traffic Statistics	Sent Packets	1676113	
		Received Packets	1782100	
Router	Router LAN IP	Router LAN IP	192.168.1.1	
		Firewall	Enabled (Green)	
	Router	USB File Sharing	USB File Sharing	Disabled (Red)
			Dynamic DNS	Disabled (Red)
Router	Software Information	Device Model	Wireless Media Gateway	
		Software Version	REL_MRVL_WMGM-88_v1.4.44	
Home Network	Home Network	Connected Devices	0 (Red)	
		Wireless		
	Home Network	Primary Wireless Network	Marvell-WMG-88-salil (Security: WPA2-PSK)	Enabled (Green)
			Other Wireless Network	Salil-Guest (Security: None)
	Home Network	Channel Number	11	
Home Network	WiFi Protected Setup	Enabled (Green)		
Home Network	DHCP Server	Enabled (Green)		

APX UI – Quick Access Links

* Quick Setup

1. Behavior after Log-In
2. Skip setup option
3. Functionality change in Quick setup behavior than previous UI.

* Help Page

1. Opens in new Tb.
2. Context specific help for each panel/screen.

* Log Out

APX UI – Commonly offered widgets

- A bunch of commonly required widgets are provided in the controls/ directory within the js code.
- For example – enabled_disabled control, ip_address control etc.
- These widgets expose a set of APIs which can be directly used.