

IPHONE INTERFACE FOR MCONTROL

A Design Project Report

Presented to the Engineering Division of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering (Electrical)

by

Aakanksha Singh

Project Advisor: Bruce Land

Degree Date: 24th May, 2009

Abstract

Master of Electrical Engineering Program

Cornell University

Design Project Report

Project Title: Design and Implementation of iPhone interface for mControl [1]

Author: Aakanksha Singh

Abstract: An interface was designed such that mControl[1] could be connected to the iPhone. The mControl is the software for digital home developed by Embedded Automation. It allows us to control all aspects of our home from Media Center PC, locally or remotely via Internet Explorer or from a Windows Mobile Device. The main objective behind this was to control the various house hold devices like lights, fan, dimmers etc through iPhone via mControl. In a fully automated solar powered house we can use iPhone or iPod touch to remotely control the appliances. This gives us an alternate and better method in addition to LCD Panels for the control of devices. Also it is more convenient as the interfacing does not require any extra hardware support in the iPhone or the system on which the mControl server is running other than the Smart Linc router.

My work here has been to design and code the interface application. This javascript code (Appendix II) enables the client running at the iPhone to contact the server at Windows Home Server and both issue commands to change the status of an Insteon device or to monitor it. In addition to coding the interface, I have also documented the steps of installation (Appendix I) of both the mControl Server and Smart Linc Router so that anyone starting from scratch can make changes in the code. The code as of now can control and monitor the lights, dimmers and the fan in the solar powered house. Progressively, the irrigation module, security module, thermostat and other Insteon devices can also be added to it without much effort.

Report Approved by

Project Advisor: _____ Date: _____

EXECUTIVE SUMMARY

The 2009 Cornell University Solar Decathlon (CUSD) consists of several teams of students that work together to design and build a fully functioning house that is completely solar-powered. The Engineering Team is responsible for designing and creating a control system that addresses issues such as power management, electrical installation, and home automation and control. The task of the Engineering Team is two-fold: to design and build a model of a control system that will function properly; and to develop a standard, easy-to-follow protocol that allows for the flexibility to easily install electrical components to the control system of the solar house.

The complete solar powered house system is self sustainable and it is being designed to be fully automated. This automation is provided by the Embedded Automations mControl software. This centralized system can be used to control the appliances connected to the Insteon devices.

The Controls team is a sub-part of the Engineering team of the Solar Decathlon Project. I am working as a team member in this Controls team. I did the design and coding for the interface between the mControl[3] software and iPhone/iPod Touch. In addition to controlling the devices in the solar powered house using the LCD panels or Windows home server, there is yet another convenient and smart method to control the devices like fans, lights, irrigation pumps, dimmers etc. The alternative is to use the iPhone/ iPod Touch for this purpose. My work in this project is to enable this functionality.

The main aim of this design document is to explain the design and the process of interfacing the iPhone/ iPod Touch to the mControl[3]. This solution enables the user to use iPhone/iPod Touch as a remote control whether he is in his house or away. This is a powerful tool in the world of home automation and is expected to facilitate the better use of mControl.

TABLE OF CONTENTS

Executive Summary	3
Introduction.....	5
Overview of the system.....	5
Interface Design Objectives.....	6
Implementation.....	7
Results And Conclusions.....	14
Constraints and Extensibility	14
References and Bibliography.....	14
Appendices	15

INTRODUCTION

In the fully automated solar powered house, the top level user interface is provided by the software mControl. This is the software for digital home developed by Embedded Automation. It allows us to control all aspects of our home from Media Center PC, locally or remotely via Internet Explorer or from a Windows Mobile Device. It can be interfaced to plug and play with Insteon and ELK products which will be detailed later on.

The mControl Mobile Client, though initially developed to support Windows Mobile devices (e.g., Windows Smartphones, Pocket PCs), can also be used for other popular mobile devices/phones including iPhone and iPod Touch devices.

This document underlines how we can use such devices to control the mControl mobile client.

OVERVIEW OF THE SYSTEM

The overall house system consists of broadly of three units – energy generation unit, energy distribution unit and control unit. Here we are concerned with the control unit of the system. The energy production and consumption performed on a per branch circuit basis is done by Computerized Electricity System (CES). The Heating, Ventilating and Air Conditioning (HVAC) control is handled by automated logic. It has both hardware and software components. The software component determines the functionality of the hardware component which can be suitably customized to HVAC system requirements.

The top level user interface mControl is a single user interface for controls as well as media and entertainment. It runs on Windows media center and is accessible through PC, mPanels and iPhones/iPod Touch. The mPanels are low power interactive touch screens supposed to be present in every room. They are powered by power over ethernet which essentially eliminates the need for extra set of wiring for power.

The Insteon products are used for the lighting and other basic sensor controls of the home. The protocols used by Insteon devices uses power lines to send communication signals to control lights and appliances. If we require complete control of the lighting in the house, every light switch has to be an Insteon light switch. This would enable it to interpret the communication signals sent over power lines and hence perform actions accordingly.

The ELK products are used for controlling additional things like pumps. The ELK products can be in turn interfaced with Insteon and it allows control of relays and standard mechanical home equipments.

The block diagram for the complete system is shown below:

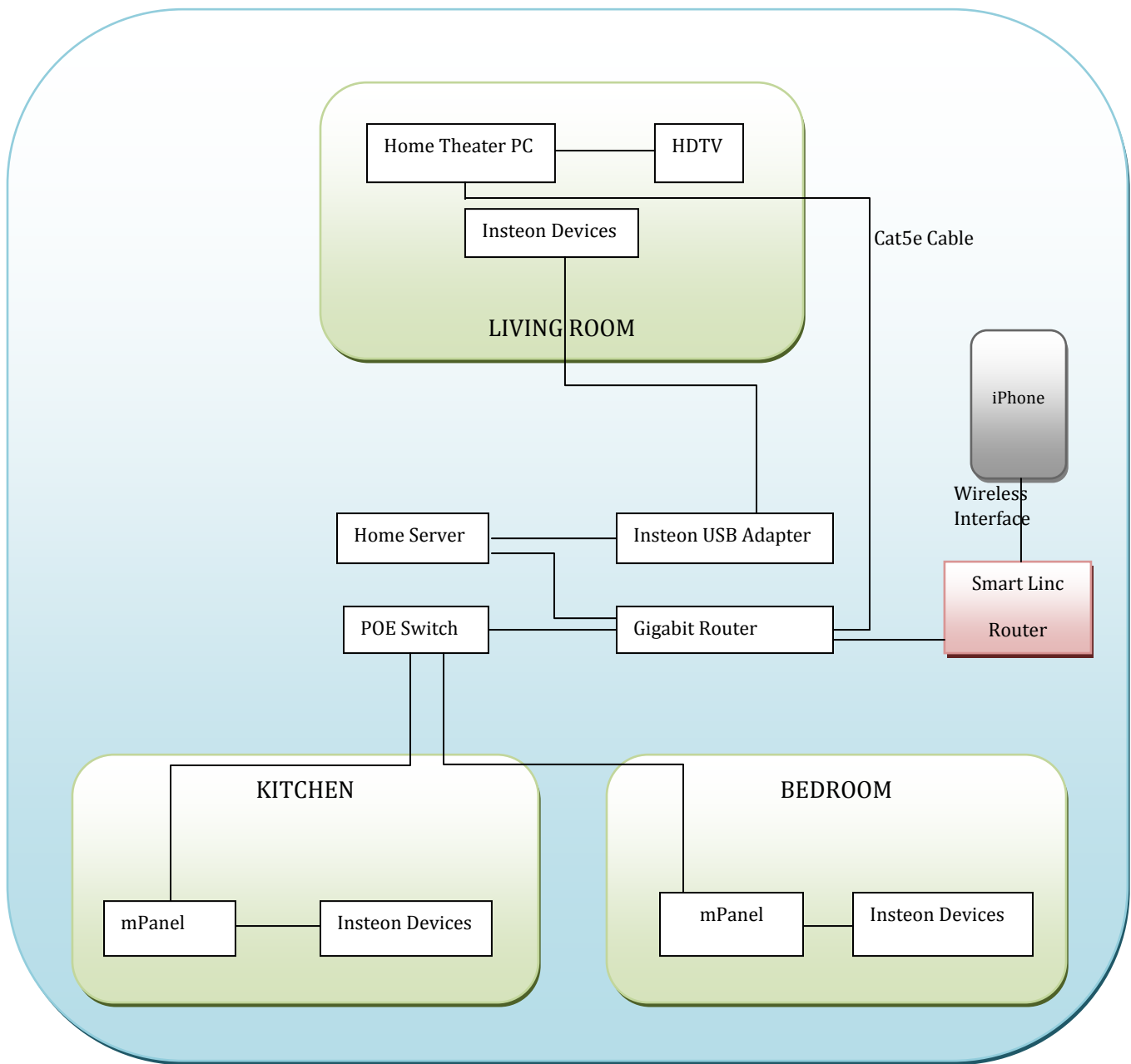


Fig1. System Diagram

My part has been to provide the iPhone interface as shown towards the right side of the system diagram. The iPhone connects to the Gigabit router and hence the home server using the Smart Linc[2] router.

INTERFACE DESIGN OBJECTIVES

The iPhone interface is supposed to provide the remote access to mControl software to control the devices connected to the mControl server. The iPhone acts as a unified control portal for the digital home. With iPhone Interface the homeowners can control digital music, lights, fan and all other devices from distance and also manage and control the surveillance and security settings.

The software interface between the mControl server and the client on iPhone is the mobile.aspx file. This file is accessed when a client such as iPhone wants to access and control the devices connected to the home server.

IMPLEMENTATION

In order to use the iPhone interface, configurations have to be made at the server and also at the router. The basic block diagram showing the interconnections between the iPhone, server and the wireless router are shown in the following diagram. My work in this project was the understanding of the code and making changes to the javascript files so that the iPhone could be interfaced to the system. Initially a large emphasis was on understanding the code i.e. how the files are linked together and how the control passes on from one class to the other when accessed. This has been shown in **Appendix III**.

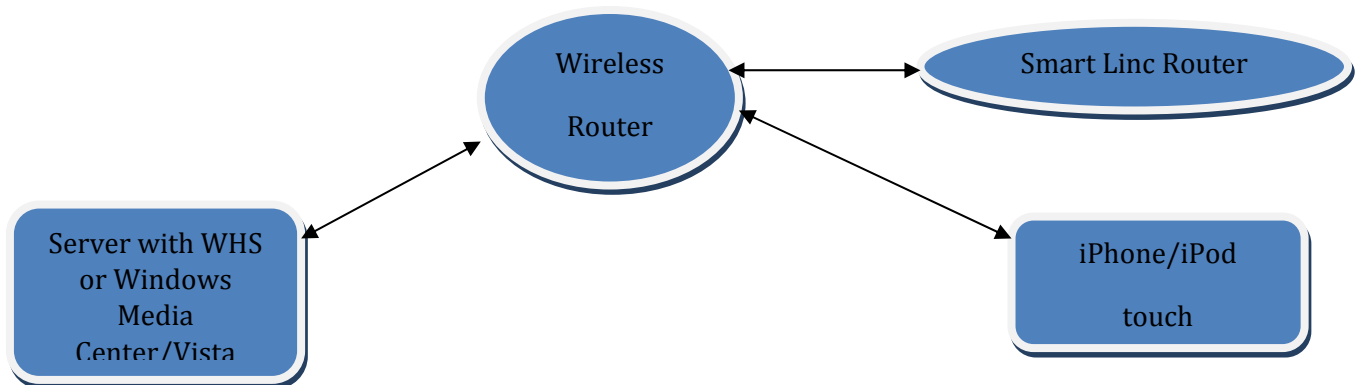


Fig2. iPhone Interface Block Diagram

Scripts and Configuration

The access point for the client where mControl is running is the ASP.NET file mobile.aspx. The source of the scripts is **mobileBasicFunctions.js**, **mobileEventsLogic.js**, **mobile_logic.js**, **mobileCommon.js**, **mobileLightObject.js**, **mobileCameraObject.js**, **mobileThermostatObject.js**, **mobileSecurityObject.js**, **mobileMainStatus.js**. The script files are located in the **C:\Program Files\Embedded Automation\mControl\scripts** (or if you are using Windows Home Server, in **C:\inetpub\wwwroot\mControl\scripts**) directory. To make changes to the script files, you must use a text or HTML editor. Changes take effect after you save the file's changes and access the Mobile page. (You may need to refresh the screen to view the changes as scripts may be resident in your browser's cache.) To recover changes, please delete the script file and then perform a

Repair Installation, which will reload the original script file. Other source files can be added to control different devices in the system like for irrigation pumps etc.

The details of these scripts is as under:

mobileBasicFunctions.js - This implements the basic functions of a mobile like getTimeString, retrieve matching query variable, get module's basename ID number, search for devices, zones etc using the ID numbers, create device objects, create data objects for triggers, macros, actions and conditions etc.

mobileEventsLogic.js - This implements the initialization of http request, create http request, processes events request, extract xml docs from string, process event change.

mobile_logic.js - This is used to populate zones, populate devices, build devices, switches, dimmers etc., view locations, populate macros and update camera images.

mobileCommon.js - This is used to browse, get or set text and send command.

mobileLightObject.js - This obtains switch status image and camera status image. We can enhance its capability by adding functions to it.

mobileSecurityObject.js - This is to get and set security status and implement if some additional functionality is desired.

mobileMainStatus.js - This implements the steps to be taken when there is a change in status of any device. It updates its status, gets macro status image and updates macro status image.

In addition to adding functionality to control the devices, we can optimize various features of the mobile client of mControl.

Style Optimizations

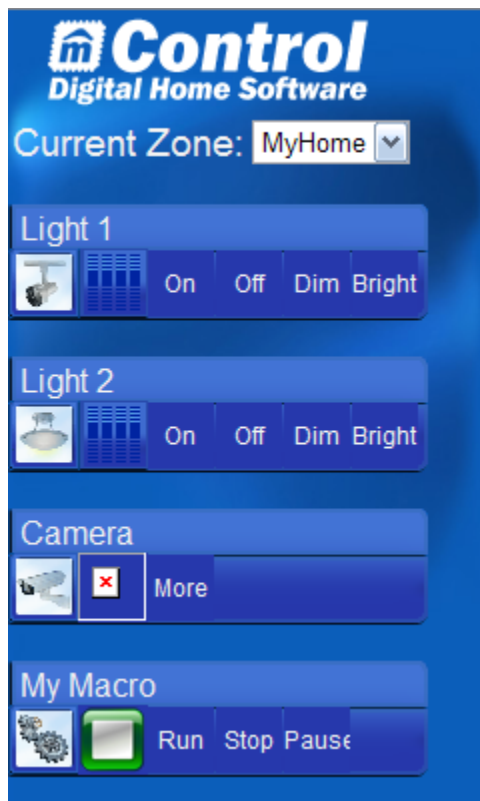
The following changes relate to the Mobile Client for mControl's style sheet. The style sheet is defined with the **blue_static.css** file, located in the **C:\Program Files\Embedded Automation\mControl\styles\mobile** (or if you are using Windows Home Server, in **C:\Inetpub\wwwroot\mControl\styles\mobile**) directory.

To make changes to the style sheet, you must use a text or HTML editor. Changes take effect after you save the file's changes and access the Mobile page. (You may need to refresh the screen to view the changes as images may be resident in your browser's cache.) To recover changes, please delete the **blue_static.css** file and then perform a Repair Installation, which will reload the original style sheet.

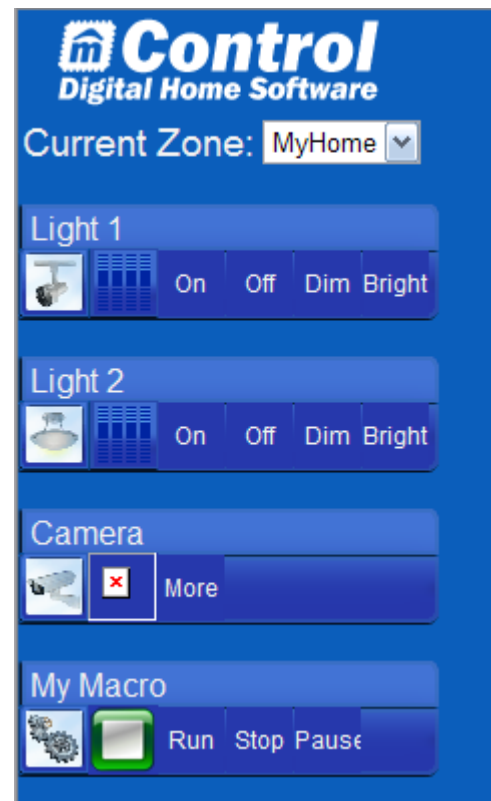
Modifying the Background Image

Removing the background image for the Mobile Client for mControl may improve loading performance and reduce memory usage. To remove the loading of background image, delete the highlighted line from the style sheet (~line 4):

```
.bodyClass
{
    background-color:#0B5EBB;
    background-image: url(../images/mobile/blue_static/Background.jpg);
    position:absolute;
    background-repeat: no-repeat; MARGIN: 0px; overflow: hidden;
    background-attachment: fixed;
    font-family: arial;
}
```



Before



After

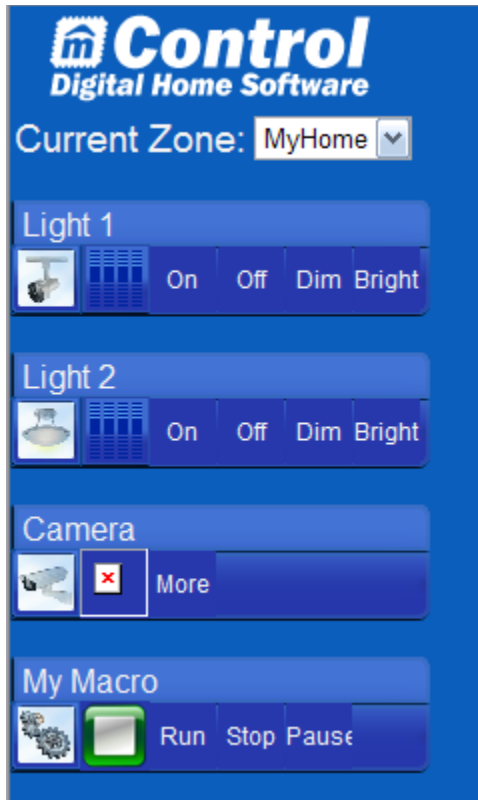
Similarly, you may consider adding your own custom background image by adjusting this line.

Modifying the Device Panel Image(s)

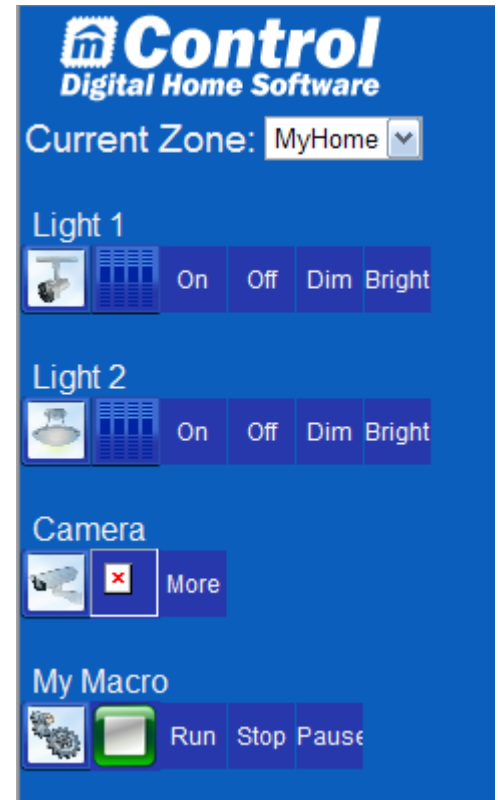
Removing the device panel image(s) for the Mobile Client for mControl may improve loading performance and reduce memory usage. To remove the loading of device panel image(s), delete the highlighted line from the style sheet (~line 48):

```
.mainDevicePanel
```

```
{
width:210; height:58;
background-image:url(../images/mobile/blue_static/Device.gif);
}
```



Before



After

Similarly, you may consider adding your own device panel image by adjusting this line.

Modifying the Device Button Text Width

Modifying the device button text width for the Mobile Client for mControl may improve legibility and aesthetics. To modify the device button text width, adjust the width setting to the desired size (~lines 67 and 74) – please note, the width of the `mainDeviceButtonTd` should be greater than `mainDeviceButton` (for example, we used 2 pixels in our example):

```
.mainDeviceButtonTd
{
width: 42;
}

.mainDeviceButton
{
background-color: #2638AC;
font-size: 9pt; color: #f2f2f2;
width:40; height:33;
border: 0; position:relative; overflow:hidden;
```

```
}

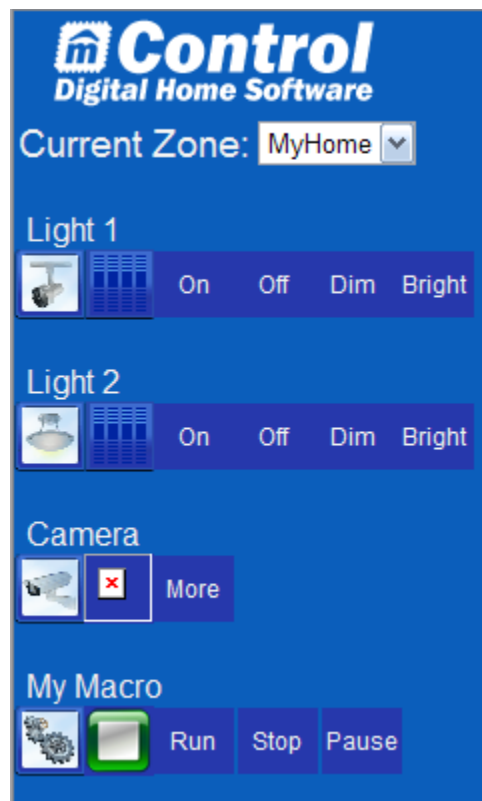
```

Similarly, you may consider adding your own device panel image by adjusting this line.

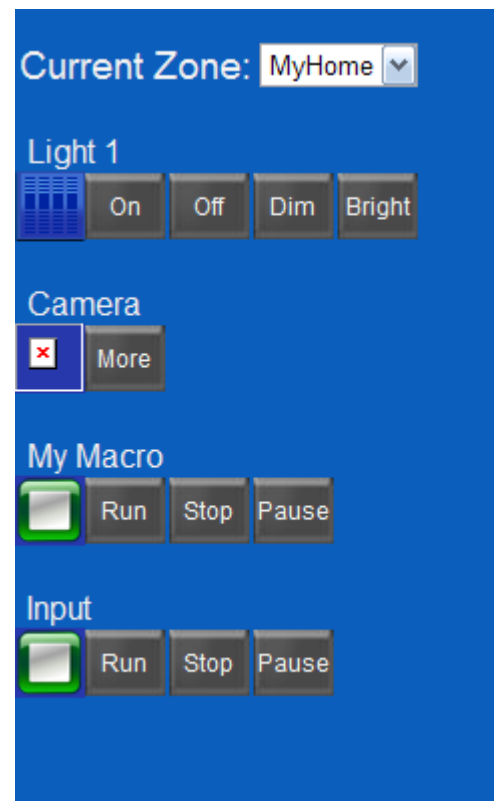
Modifying the Device Buttons

Modifying the device button to use a button image for the Mobile Client for mControl may improve legibility and aesthetics. To modify the device button to use a button image, add a button image to the `\mControl\images\mobile\blue_static` directory and then add a reference to the `mainDeviceButton` class: (for example, we created `DeviceButton.jpg` in our example):

```
.mainDeviceButton
{
background-image: url(../images/mobile/blue_static/DeviceButton.jpg);
/*background-color: #2638AC;*/
font-size: 9pt; color: #f2f2f2;
width:40; height:33;
border: 0; position:relative; overflow:hidden;
}
```



Before



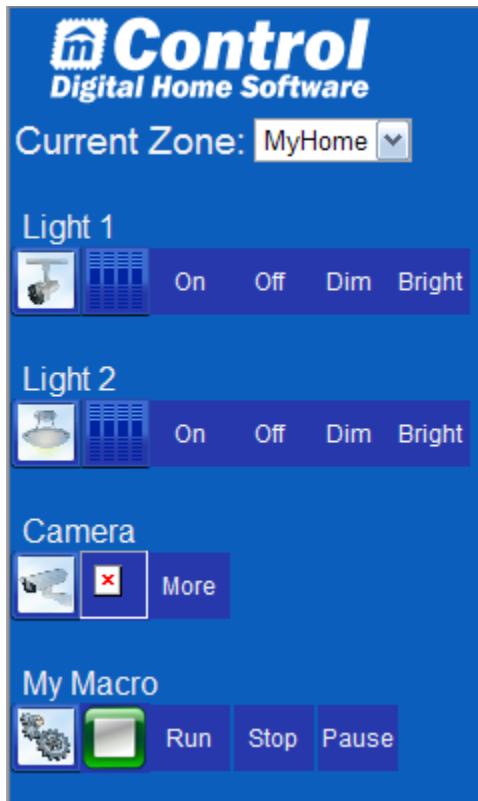
After

Modifying the Device Icon Image

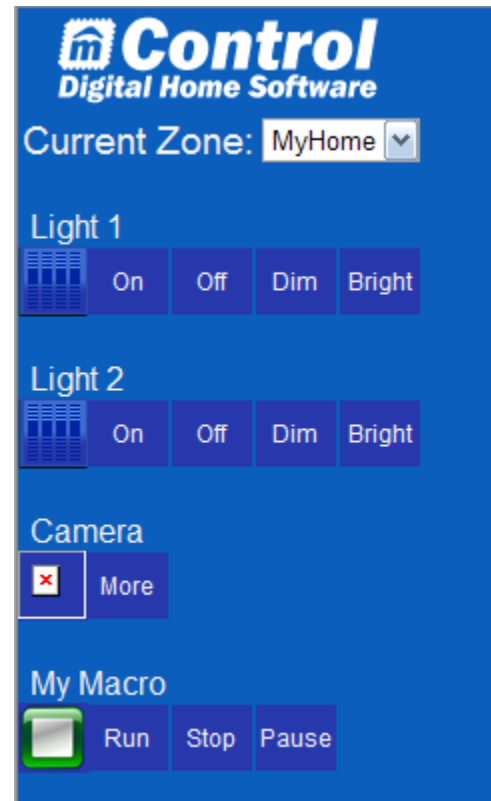
Removing the device icon image for the Mobile Client for mControl may improve loading performance and reduce memory usage. To remove the loading of device icon image, open the [mobile_logic.js](#) file and comment out the highlighted line (~line 205):

```
// This is the line which includes the device icon as part of the device's
// You can comment out this line by adding "//" to the beginning of the line, as shown:

//html += "<td class='mainDeviceIcon'><div class='mainDeviceImage' style='background-image:
url(" + url+
");'/></td>";
```



Before



After

Similarly, you may consider adding your own custom background image by adjusting this line.

HTML (ASP.NET) Optimizations

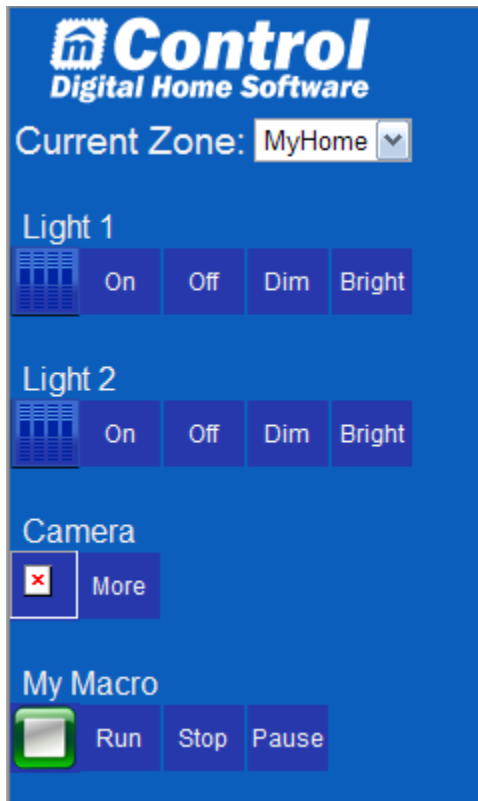
The following changes relate to the Mobile Client for mControl's HTML (ASP.NET) files. These HTML (ASP.NET) files are located in the [C:\Program Files\Embedded Automation\mControl](#) (or if you are using Windows Home Server, in [C:\Inetpub\wwwroot\mControl](#)) directory. To make changes to the HTML (ASP.NET) files, you must use a text or HTML editor. Changes take effect after you save the file's changes and access the Mobile page. (You may need to refresh the screen to view the changes as HTML (ASP.NET) information may be resident in your browser's cache.)

To recover changes, please delete the HTML (ASP.NET) file and then perform a Repair Installation, which will reload the original HTML (ASP.NET) file.

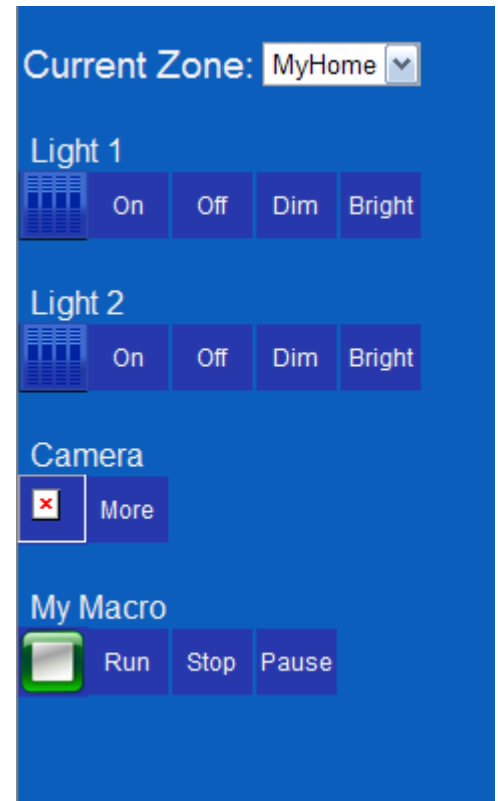
Modifying the Logo Image

Removing the logo image for the Mobile Client for mControl may improve loading performance and reduce memory usage. To remove the loading of device icon image, open the [mobile.aspx](#) file and delete the highlighted line (~line 24):

```
<body class="bodyClass" id="body" onload='bodyLoad()>
<img src='images/vendor/vendor1.png' alt = 'mControl Home' />
```



Before



After

Similarly, you may consider adding your own custom background image by adjusting this line.

Smart Linc Router configuration

Smart Linc[2] is an internet web server interface to the INSTEON products optimized for mobile devices. There are some steps which need to be followed to set up and install this so that mobile devices can be used for controlling Insteon devices. This assigns an IP and a port to the client running at the iPhone/iPod Touch. Similarly, an IP and a port is also assigned to the server running at the Windows system in the house. When we access <http://SERVERIP:PORT/mobile.aspx>, we are able to connect to the server. It is the Smart Linc router that helps us to connect to the

Gigabit router/ Wireless router and hence the Windows Server. The Steps for installation of the smart Linc router are documented in the Appendix I.

RESULTS AND CONCLUSIONS

The iPhone can be used to as an interface for home automation now with Smart Linc router if already connected. Currently, the server and the client scripts were run on the same machine due to the unavailability of the Smart Linc router. The client could access the zones successfully.

Apparently, any new device that needs to be controlled through iPhone has to be interfaced at two places in the code. Firstly, the logic corresponding to its implementation has to be added in the “scripts” folder as a .js file. Secondly, the link for this source script has to be provided in the mobile.aspx file and we are all set to control it.

The implementation of this interface lends huge advantage to the solar decathlon house as a whole. It has increased the convenience of the resident of the house by enabling him to control devices at his home using the iPhone.

CONSTRAINTS AND EXTENSIBILITY

This control of the Insteon devices using the iPhone/iTouch interface has some limitations. We cannot control the Irrigation module, the security module, the camera module as of now. But in future we can add this implementation and use iPhone for this purpose as well. The extension to control these modules does not involve major design level changes. We just have to add code in the corresponding .js files.

REFERENCES AND BIBLIOGRAPHY

- [1] mControl Embedded Automation- <http://www.embeddedautomation.com/EACONewsPR071105-01.htm>
- [2] SmartLinc Insteon Central Controller - <http://www.smarthome.com/2412N/SmartLinc-INSTEON-Central-Controller/p.aspx>
- [3] mControl Application Note 06-0001-A
- [4] mControl Application Note 06-0002-A
- [5] mControl v2 User Manual

APPENDICES

APPENDIX I

Installation of mControl Clients (Vista) to talk to mControl Server (WHS)

A recommended system for mControl operation utilizes the mControl server on a dedicated machine, while access/operation of mControl happens from other machines on the network.



Further, we recommend the Windows Home Server (WHS) platform as the mControl server because:

- (a) WHS is operational 24-7, allowing the mControl system to be used whenever required. This provides flexibility in the macro operation - they can run anytime.
- (b) WHS provides a console for general administration - mControl Editor and mControl Manager have been designed for this console.
- (c) WHS provides remote access capability. Microsoft even provides a free DDNS service.
- (d) Of course, WHS provides storage and back-up features.

Many people are also using a Windows Media Center on their network also.

Now here are the guidelines on how to setup a mControl client/server system.

The key steps are:

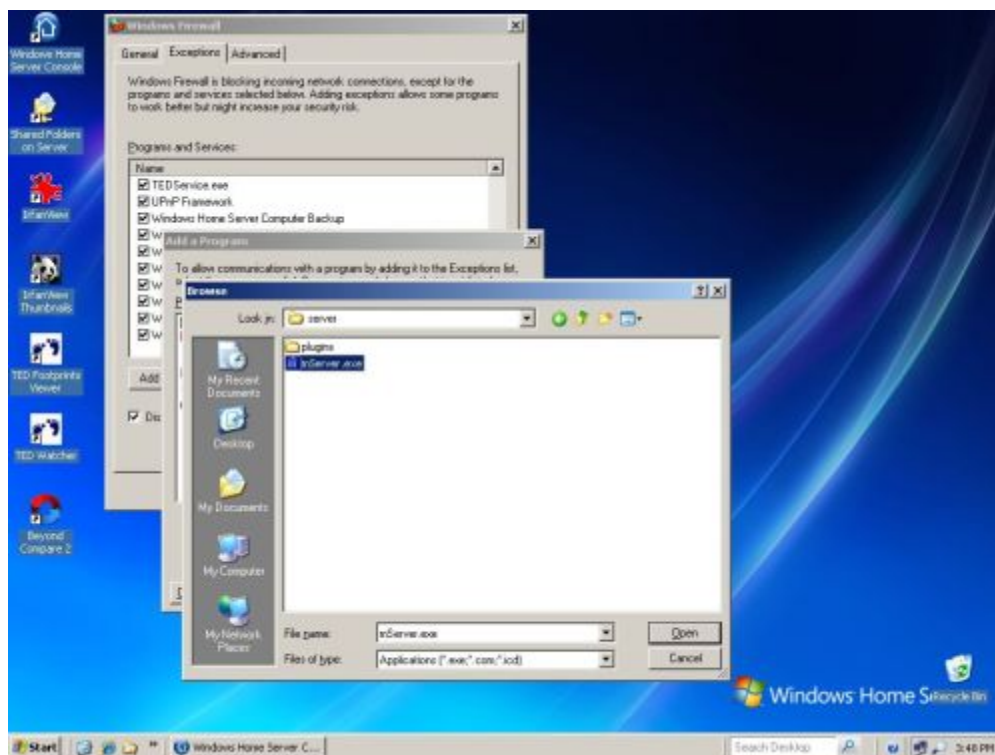
1. Install mControl on the Server Machine
2. Setup the Server Machine's Firewall (to allow the mControl server to send/receive messages with mControl clients).
3. Setup the Client Machine's Firewall (to allow the mControl client to send/receive messages with the mControl server)
4. Install mControl on the Client Machine
5. Allow Firewall exceptions for mControl Clients.

1. Install mControl on the Server Machine:

Use the instructions provided in the "*mControl Installation*" section of the mControl v2 User Manual. Please note, the installer and installation method differ if you installing on a Vista or XP machine, as compared to a WHS server machine.

2. Setup the Server Machine's Firewall

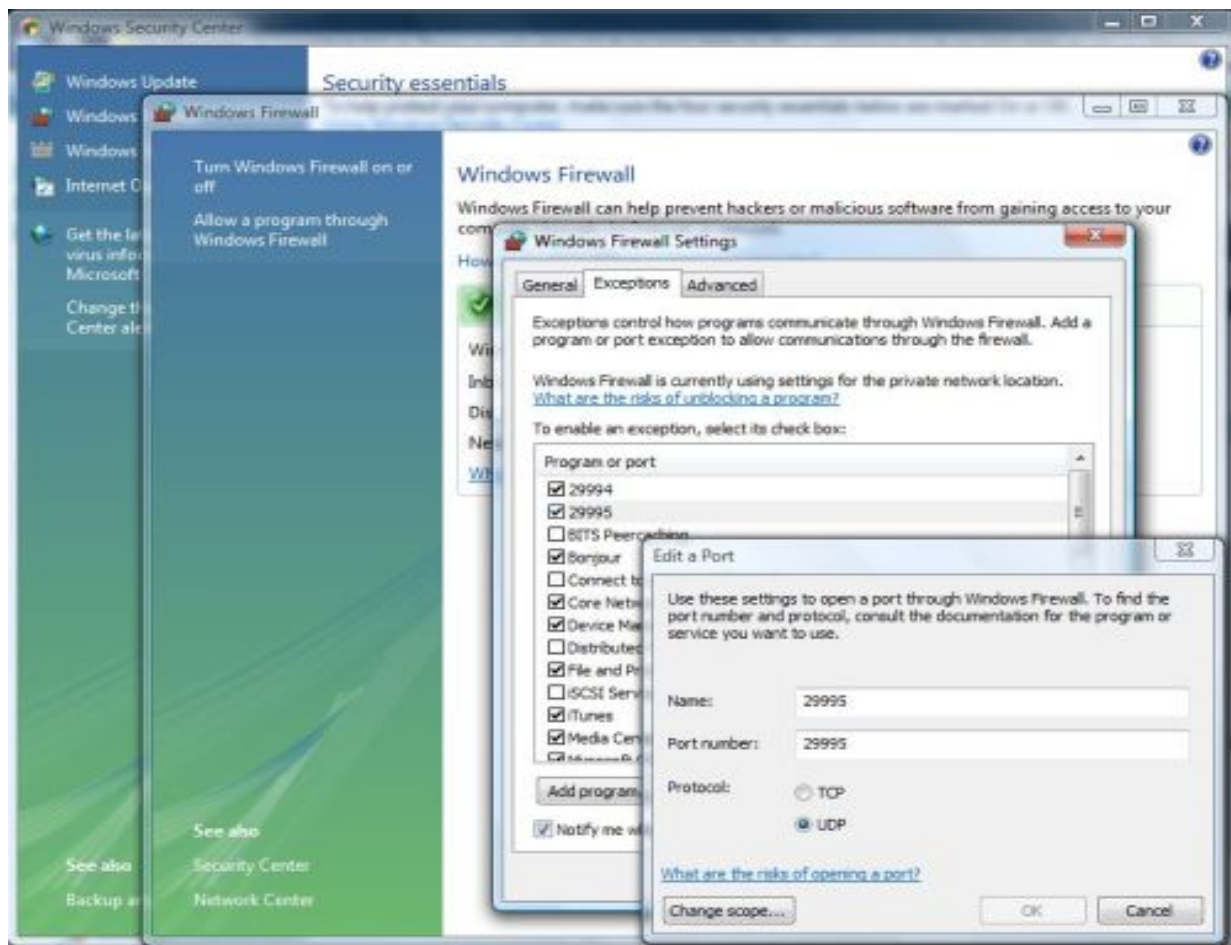
If you are using a Firewall (which is recommended), you will need to except the mControl Server - so that it can send/receive messages with mControl clients.



To do this, open your Firewall and select the "Exceptions" tab. Click on the "Programs ..." button and browse to mServer.exe. On your WHS machine, mServer.exe is located in the [C:\inetpub\wwwroot\mControl\server](#) directory. On Vista/XP machines, mServer.exe is located in the [C:\Program Files\Embedded Automation\mControl\server](#) directory.

3. Setup the Client Machine's Firewall (to allow the mControl client to send/receive messages with the mControl server)

If you are using a Firewall (which is recommended), you will need to except a few ports used by the mControl clients - so that it can send/receive messages with mControl server.



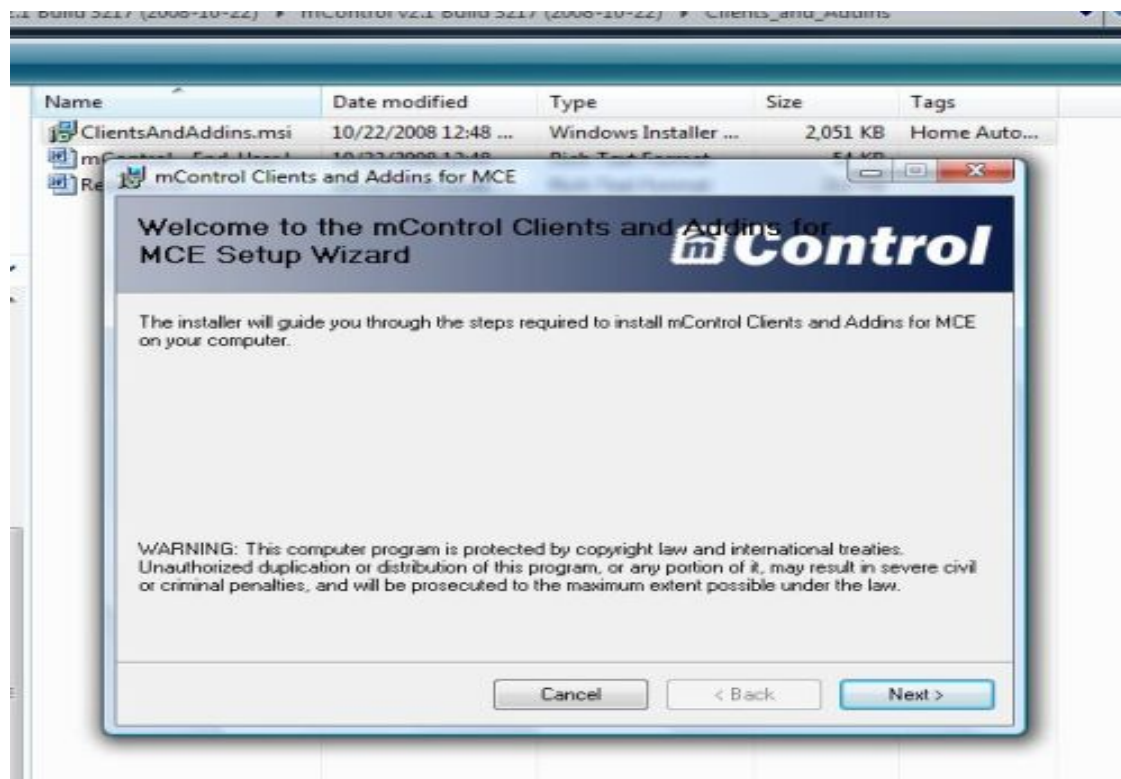
Specifically, the following ports need to be excepted to allow proper operation:

- Port 29990 (TCP) - used by mControl's internal web server and browser/media center clients
- Port 29994 (UDP) - used by mControl service to send messages (e.g. status messages)
- Port 29995 (UDP) - used by mControl clients to send messages

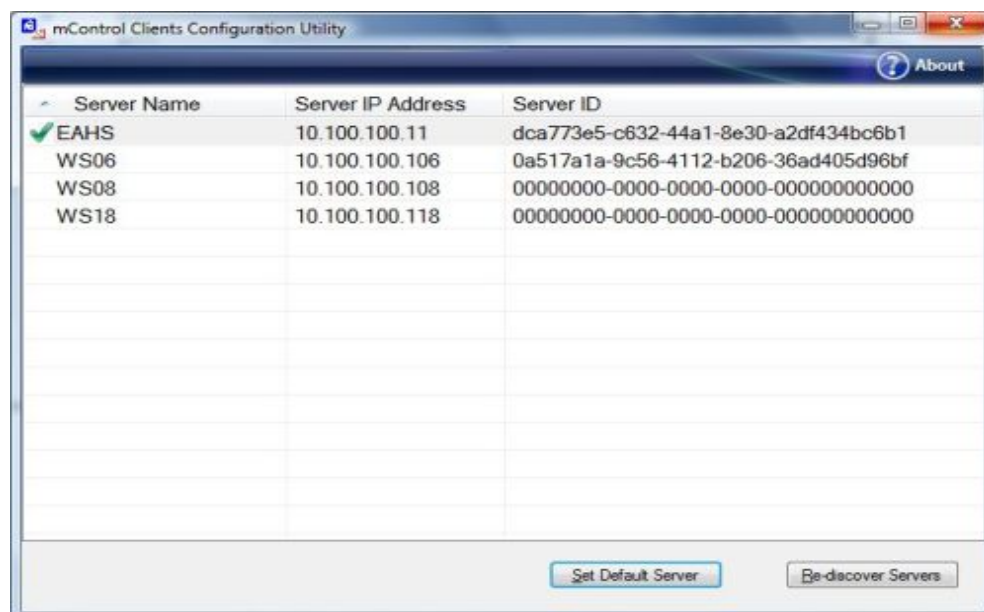
4. Install mControl on the Client Machine

You don't need to install the full mControl on your client machines - use the mControl Client Installer to install the necessary software:

<http://www.embeddedautomation.com/fo...ead.php?t=1129>

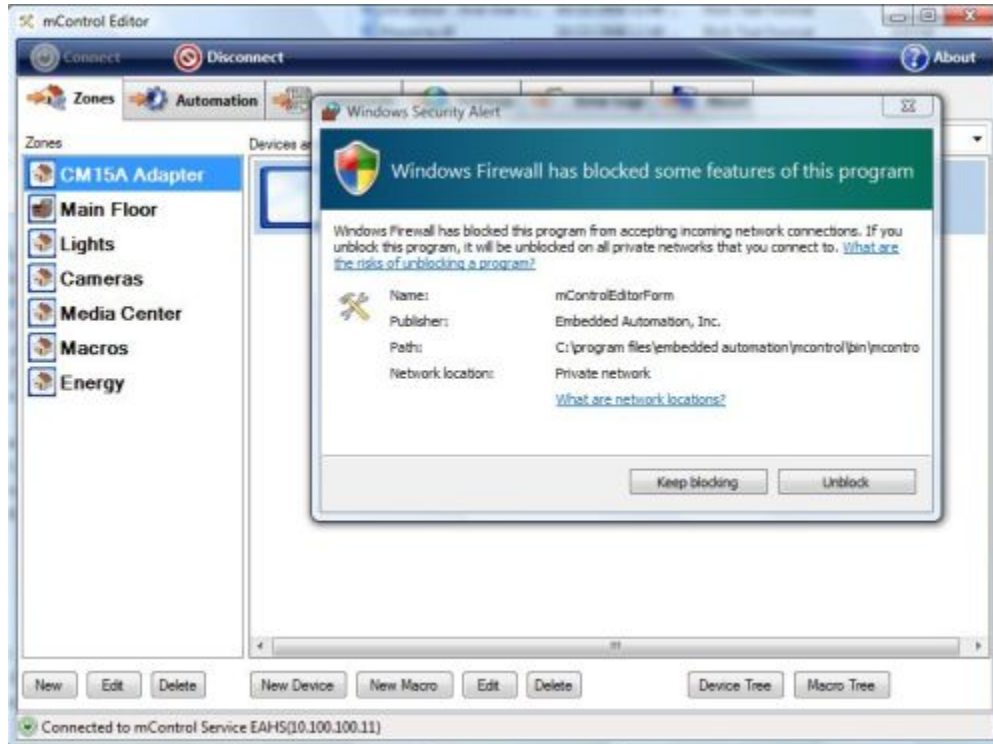


Towards the end of the installation, the installer will provide a list of mControl servers to use – if you haven't done step 3, you may not see any mControl servers. Select the default server.

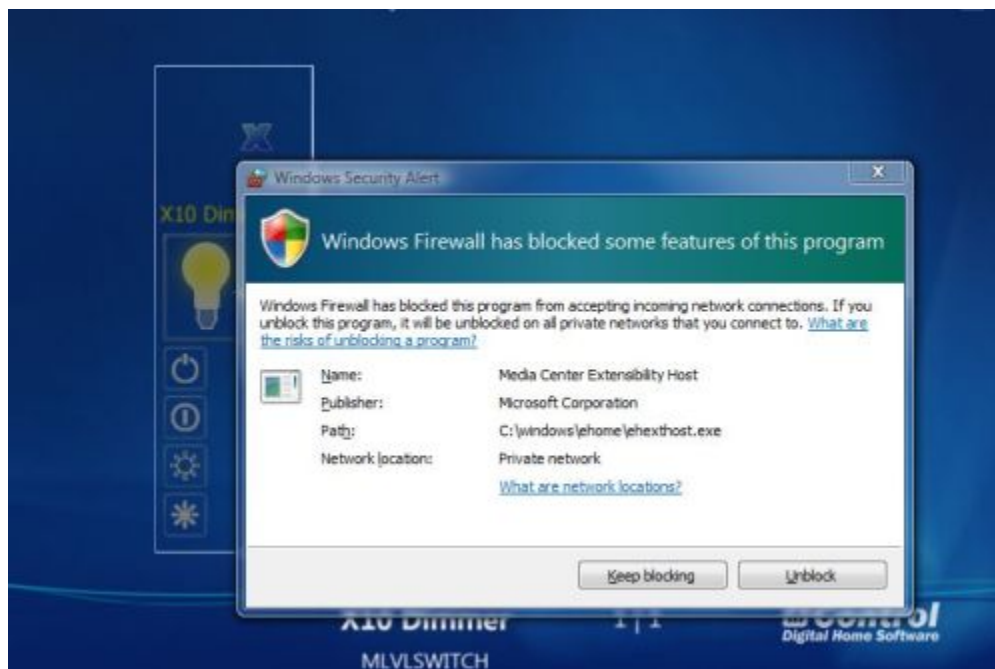


5. Allow Firewall exceptions for mControl Clients

When you run the various mControl clients, your Firewall will ask if you would like to except the program/ports used by the client - select "Unblock" to allow the exception. mControl Editor (mControlEditorForm.exe):



mControl Vista Media Center Client (Media Center Extensibility Host):



Using the mControl Client and Addins installer

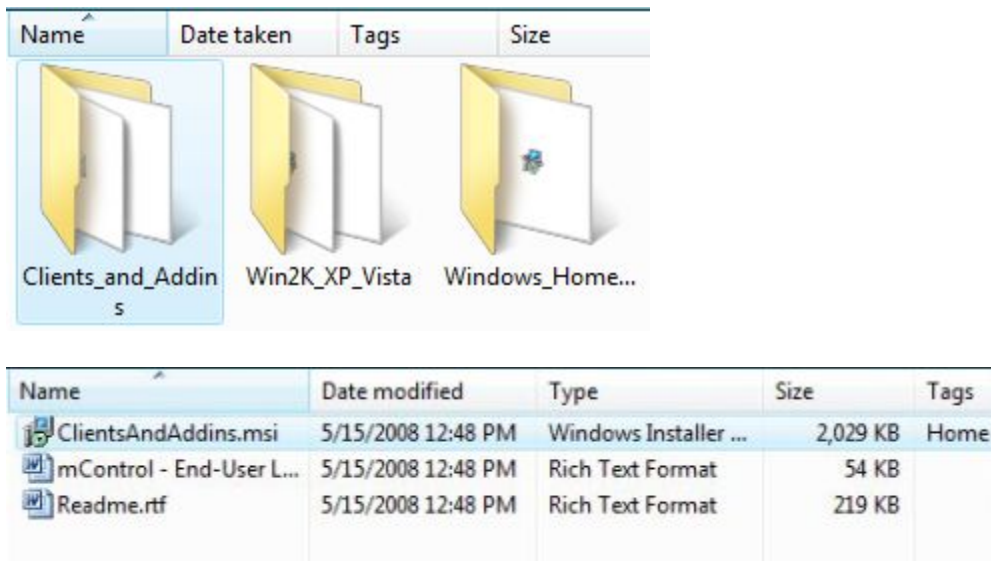
Instead of doing a full install of mControl, sometimes we just need to install client. A client install is practical when your automation server is hosted on another machine. It allows you to discover mControl servers and set the default server without editing any configuration files.

When you do a client install, the following programs and utilities are installed:

1. mControl Editor
2. mControl for MCE
3. Client Connection Configurator
4. OneVoice to mControl Bridge

Due to possible resource conflicts, it is recommended to only do one type of install on your machine. You will also need to configure your firewall on each system that is either a client or service host to allow exceptions for ports 29990 to 29995.

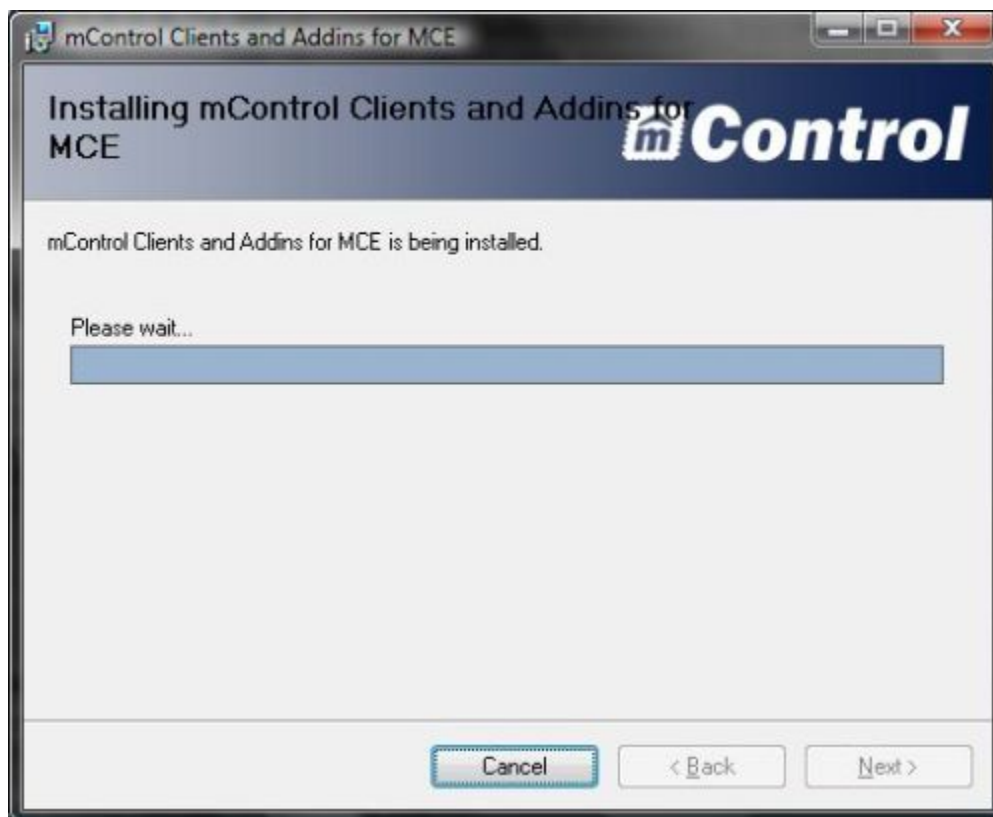
After downloading and extracting the folder with the latest build of mControl, run ClientsAndAddins.msi, located in the Clients_And_Addins folder.



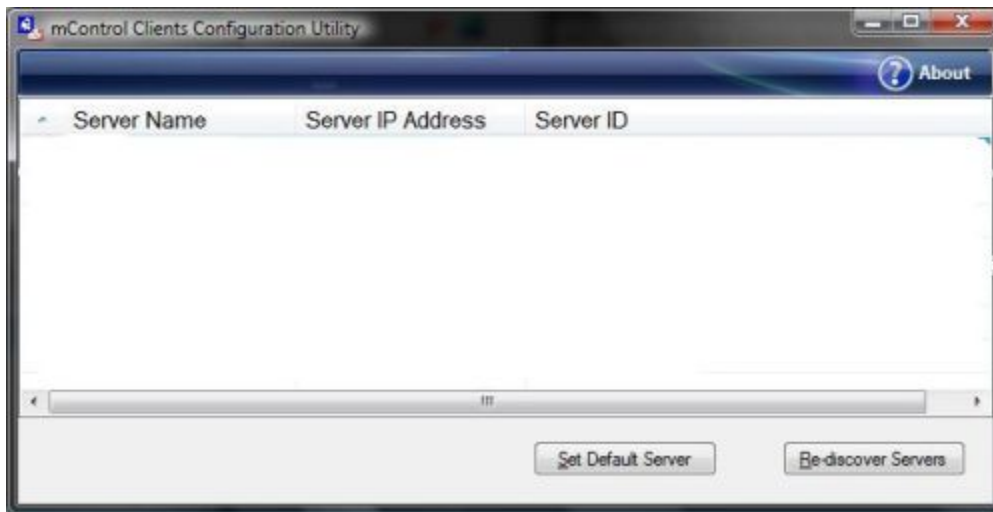
The following screen should appear:



Press next, and agree to the license.

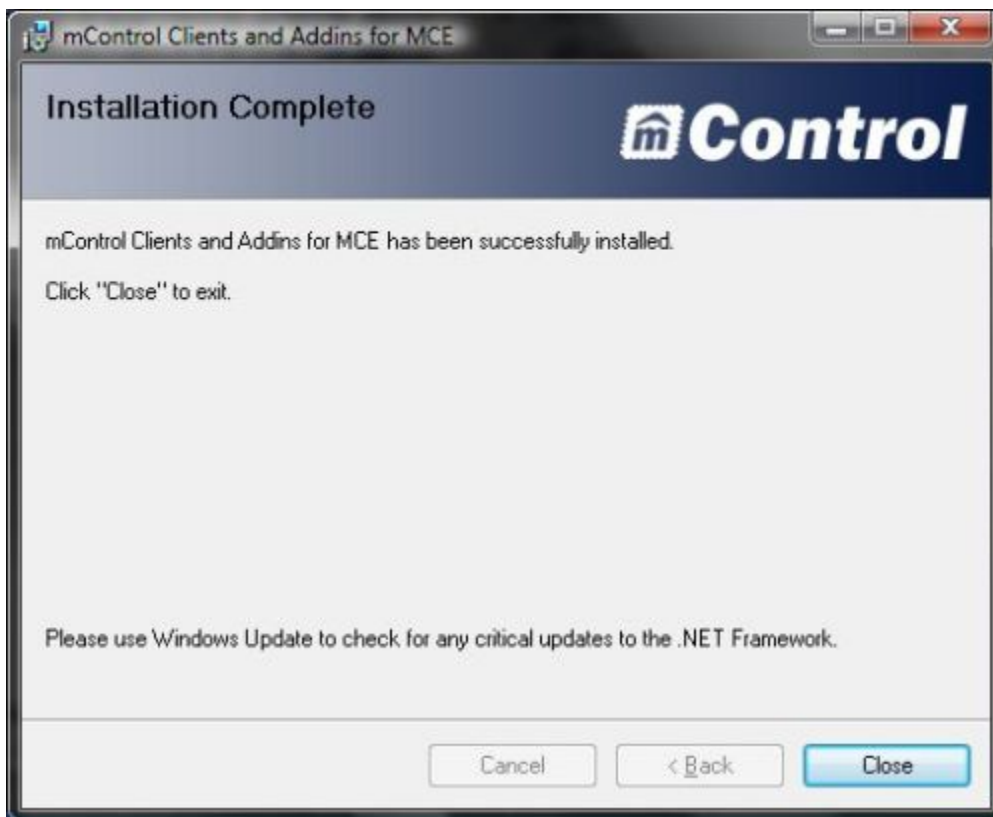


The following screen should show up when the install is almost complete.



Here, you can select from a list of mControl servers on your network. Click on Re-Discover Servers and set the server you want to use as the default. Now close this window.

An info screen will be displayed. Press next, and installation will finish.



While opening an mControl client, such as mControl Editor, the client will now connect to the

server chosen as the default. The default server can be changed at any time by using the Client Connection Configurator tool.

SmartLinc Router Installation

1 Plug included Ethernet cable into an open LAN port on your router.

2 Plug the other end of Ethernet cable into SmartLinc's Ethernet jack.

3 Plug SmartLinc into an AC outlet

SmartLinc's white LED should turn on dimly

4 Verify proper connection between SmartLinc and your router:

*Router's LED for the port SmartLinc is plugged into should be on
Green LED on SmartLinc's Ethernet jack should be on*

Logging In to Smart Linc from Home

5 Open a browser on your PC, PDA, or mobile phone (while connected to the Internet) (Java Script support required for default web server application, for devices that do not support Java Script, see step 8's note below)

6 Type <http://smartlinc.smarthome.com>

7 Click on "I'm at home"

8 Click on the link that begins with <http://>

SmartLinc's Home Page will appear

"You should check the clock" text will disappear once you set the clock

Setting the Time, Day and House Name

9 Click on the Settings icon (Wrench, bottom center) (or get here the first time opening SmartLinc, by clicking on "You should check the clock")

10 Type in the correct time and click "Set Time"

11 Click on the correct day of week and click "Set Day"

12 If desired, type in a name for your house and click "Set House Name"

13 Click on the Home icon (upper left) to return to the homepage.

Set up a Room, A Device and Timer

14 SmartLinc comes with samples of rooms, scenes/devices and timer events. Feel free to navigate around now and get familiar with where things are setup and stored. If you wish to clear all the samples out of the unit, Click on the Settings icon (wrench) and go to step 15. Otherwise, proceed to step 17.

15 Click on Authentication at the bottom of the Settings page:

SmartLinc's Set Login screen will appear

16 Click on:

- a. Clear all Labels
- b. Clear all Timers
- c. Clear all URLs
- d. Reset all Checkboxes

17 Click the Room Settings icon (lower left corner of homepage)

18 Type "Frontyard" over the first "room" label (room indicates that no name had been assigned yet)

19 Click the "Show Room" Checkbox to the right of the room name

20 Click Save and Return

21 Click on "Frontyard"

22 Click the Scene Settings icon (lower left)

23 Type "Front Lights" over the first "scene" label

24 Click the "Show" Checkbox to the right of the scene name

25 Click Save and return

26 Click on "Settings" (far right)

27 Click on "Add"

The Add/Remove Status will change to "waiting".....

28 Press & Hold the Set button on the Front Lights SwitchLinc (or other INSTEON device such as KeypadLinc, LampLinc, etc.) for 5 seconds (or per that devices Quick Start Guide linking instructions)

The Add/Remove Status will change to "Done" – and display the INSTEON ID of the device you added (e.g. 0D425B).

29 Click the "ON" and "OFF" icons

Your lights will turn on and off

30 Type 07:00 into the "On Time box" and Click PM

31 Type 11:00 into the "Off Time box" and Click PM

32 Click on Sat and Sun (to uncheck these days)

33 Click on Save and return

Smart Linc Setup away from Home

These instructions are also for in-home use if the iPod Touch device does not support Wi-Fi. These instructions assume the user is familiar with networking setup and firewall management (you will need to refer to your router's instruction manual to determine how to perform the following steps on your specific router). User needs to be aware that there are always risks associated with configuring devices to make them accessible to the Internet, User takes full responsibility for network management and security. Smartlabs takes no responsibility for your network security. You must be on your local network (same as your SmartLinc) when Registering at <http://smartlinc.smarthome.com> (final steps below).

- 1) Log onto your router
- 2) Determine what your router's range of DHCP addresses is and pick an IP address outside this range (e.g. 192.168.0.101)
- 3) Open the "port forwarding" (sometimes called virtual servers and other things) feature on your router
- 4) Forward the IP address (and we advise a random port of your choice to move the server off the default port 80)
- 5) Log onto your SmartLinc while at home via <http://smartlinc.smarthome.com> and choosing the "I'm at home" option
- 6) Click the Settings icon (wrench, bottom middle)
- 7) Click Authentication (bottom of screen)
- 8) Enter your username and passwords of choice (10 character max., letters, numbers and "@" only)
- 9) Click "Save User Name and Password"
- 10) Click "Back" on your browser
 - a. You may need to log into SmartLinc at this point
- 11) Click "Change Network Settings"
- 12) Change the IP address to match the IP address you forwarded in your router (above)
- 13) Change the Port to match the port you forwarded in your router (above)
- 14) Click on Disabled for DHCP
- 15) Click on "*Save"
- 16) Unplug your SmartLinc from the electrical outlet for at least 10 seconds, then plug it back in
- 17) Enter your SmartLinc's new IP address and port (if applicable) into your browser for example <http://192.168.0.20:8080>

- 18) Your SmartLinc homepage should reappear
 - 19) Log onto <http://smartlinc.smarthome.com>
 - 20) Click on “New User Registration”
 - 21) Type in your username
 - 22) Type in your password (twice)
 - 23) Type in the INSTEON ID of your SmartLinc (8 characters, including dots)
 - 24) Click “Submit”
 - a. You will need to enter your SmartLinc username and password again if you have Authentication turned on as advised above
- Log into SmartLinc while away from Home (Optional)**
- 1) Log onto <http://smartlinc.smarthome.com>
 - 2) Click on “I’m not at home”
 - 3) Type in your username
 - 4) Type in your password
 - 5) Click “Log In”
- Your SmartLinc Home Page will appear*

APPENDIX II

Source Code

1. Source file: mobile_logic.js

```
//=====
/// <copyright from='2005' to='2008' company='Embedded Automation, Inc.'>
///
/// mControlWeb
///
/// Copyright (c) Embedded Automation, Inc. All rights reserved.
///
/// </copyright>
//
// $Id: mobile_logic.js,v 1.21 2008/04/22 19:25:05 george Exp $
//
//=====

var _selectedZoneArrayIndex = 0; // Currently selected zone (JS array
index)
var _selectedZoneId; // Database zone id
var _cameraImageIds = new Array();

function PopulateZones()
{
    var selectOptions = "";

    var selectedZoneArrayIndex = -1;
    var selectedZoneId = getQueryVariable("Zone")
    if (selectedZoneId != FAILED)
    {
        selectedZoneArrayIndex = searchForID(SystemZones,
selectedZoneId);
    }
    if(selectedZoneArrayIndex < 0)
    {
        selectedZoneId = searchForOrder(SystemZones);
        selectedZoneArrayIndex = searchForID(SystemZones,
selectedZoneId);
    }
    _selectedZoneId = selectedZoneId;
    _selectedZoneArrayIndex = selectedZoneArrayIndex;
    var zoneArrayIndexes = new Array(); // Array of VALID
Zones, indexed by each zones order
    for (var i = 0; i < SystemZones.length; i++)
    {
        var order = SystemZones[i].getOrder();
        if (order >= 0)
            zoneArrayIndexes[order] = i;
    }

    for(var j = 0; j < zoneArrayIndexes.length; j++)
    {
        var zone = SystemZones[zoneArrayIndexes[j]];
        selectOptions += "<option value='" + zone.getID() +
```

```

        "'>" + zone.getName() + "</option>"
    }

    return selectOptions;
}

function cmbZone_SelectedIndexChanged()
{
    var path = location.pathname;
    var p = path.lastIndexOf("/");
    if (p >= 0) { path = path.substring(0, p); }
    location.href = path + "/Mobile.aspx?Zone=" + cmbZones.value;
}

function PopulateDevices()
{
    InitXMLHttpRequest();
    var deviceList = new Array();
    var orders = new Array();
    var deviceHTML = "";
    var n = 0;
    for (var i = 0; i < SystemDevices.length; i++)
    {
        device = SystemDevices[i];
        if (device.getZone() == _selectedZoneId)
        {
            var order = device.getOrder();
            orders[n++] = order;
            deviceList[order] = device;
        }
    }
    orders.sort(_SortNumber);

    for (var i = 0; i < n; i++)
    {
        var index = orders[i];
        deviceHTML += _BuildOneDeviceHtml(deviceList[index]);
    }

    return deviceHTML;
}

function _BuildOneDeviceHtml(device)
{
    var deviceHTML = "";
    var moduleIndex = findIndexOf(SystemModules, device.getModule());
    var adapterIndex = findIndexOf(SystemAdapters, device.getAdapter());
    var stat = device.getStatus();
    if (moduleIndex >= 0 && adapterIndex >= 0) {
        switch (SystemModules[moduleIndex].getModuleClass())
        {
            case "BINSWITCH":
                var level = parseInt(stat);
                deviceHTML = _BuildSwitchHTML(device, level);
                break;
            case "MLVLSWITCH":
                var level = parseInt(stat);

```

```

        deviceHTML = _BuildDimmerHTML(device, level);
        break;
    case "IPCAMERA":
        deviceHTML = _BuildCameraHTML(device);
        break;
    case "THERMOSTAT":
        deviceHTML = _BuildThermostatHTML(device, stat);
        break;
    case "ALARMSYS":
        deviceHTML = _BuildSecurityHTML(device, stat);
        break;
    }
}
else {
    deviceHTML = _BuildDisabledDeviceHTML(device);
}
return deviceHTML;
}

function _BuildDisabledDeviceHTML(currentDevice)
{
    var html = "<td class='mainDeviceStatus'><img id='Device" +
currentDevice.getID() + "' " +
        "src='images/mobile/blue_static/Status.-1.gif' /></td>";
    html += "<td class='mainDeviceDisabledText'>Disabled</td>";

    return _BuildDeviceHtml(currentDevice.getName(), currentDevice.getImage(),
currentDevice.getID(),
        false, html, null, null);
}

function _BuildSwitchHTML(currentDevice, level)
{
    var switchHTML = "<td class='mainDeviceStatus'><img
id='mainDeviceStatus" + currentDevice.getID() + "' " +
        "src='" + GetSwitchStatusImage(level) + "' /></td>";
    switchHTML += "<td class='mainDeviceButtonTd'><input type='button'
class='mainDeviceButton' value='On' onclick=\"SendCommand('AutomationCommand', "
+ currentDevice.getID() + ", 'On')\" " + " /></td>";
    switchHTML += "<td><input type='button' class='mainDeviceButton'
value='Off' onclick=\"SendCommand('AutomationCommand', " +
currentDevice.getID() + ", 'Off')\" " + " /></td>";

    return _BuildDeviceHtml(currentDevice.getName(), currentDevice.getImage(),
currentDevice.getID(),
        false, switchHTML, null, null);
}

function _BuildDimmerHTML(currentDevice, level)
{
    var switchHTML = "<td class='mainDeviceStatus'><img
id='mainDeviceStatus" + currentDevice.getID() + "' " +
        "src='" + GetDimmerStatusImage(level) + "' /></td>";
    switchHTML += "<td class='mainDeviceButtonTd'><input type='button'
class='mainDeviceButton' value='On' onclick=\"SendCommand('AutomationCommand', "
+ currentDevice.getID() + ", 'On')\" " + " /></td>";

```

```

        switchHTML += "<td class='mainDeviceButtonTd'><input type='button'
class='mainDeviceButton' value='Off' onclick=\"SendCommand('AutomationCommand',
" + currentDevice.getID() + ", 'Off')\" + \"></td>";
        switchHTML += "<td class='mainDeviceButtonTd'><input type='button'
class='mainDeviceButton' value='Dim' onclick=\"SendCommand('AutomationCommand',
" + currentDevice.getID() + ", 'Dim')\" + \"></td>";
        switchHTML += "<td><input type='button' class='mainDeviceButton'
value='Bright' onclick=\"SendCommand('AutomationCommand', " +
currentDevice.getID() + ", 'Bright')\" + \"></td>";

        return _BuildDeviceHtml(currentDevice.getName(), currentDevice.getImage(),
currentDevice.getID(),
            false, switchHTML, null, null);
    }

function DirectToCameraView(devId)
{
    window.location='MobileCameraView.aspx?dev=' + devId;
}

function _BuildCameraHTML(currentDevice)
{
    var id = currentDevice.getID();
    SendCommand('AutomationCommand', id, "StreamOneShot")
    var html = "<img class='mainCameraImg' id='mCamImg' + id + \" />";
    _cameraImageIds[_cameraImageIds.length] = "mCamImg" + id;
    return _BuildDeviceHtml(currentDevice.getName(), currentDevice.getImage(),
currentDevice.getID(),
        true, html, "mainDeviceStatus", "DirectToCameraView");
}

function DirectToThermostatView(devId)
{
    window.location='MobileThermostatView.aspx?dev=' + devId;
}

function _BuildThermostatHTML(currentDevice, stat)
{
    var temprature = GetThermostatTemperature(stat, true);
    return _BuildDeviceHtml(currentDevice.getName(), currentDevice.getImage(),
currentDevice.getID(),
        true, temprature, "mainDeviceStatus", "DirectToThermostatView");
}

function DirectToSecurityView(devId)
{
    window.location='MobileSecurityView.aspx?dev=' + devId;
}

function _BuildSecurityHTML(currentDevice, stat)
{
    var status = GetSecurityState(stat);
    return _BuildDeviceHtml(currentDevice.getName(), currentDevice.getImage(),
currentDevice.getID(),
        true, status, "mainDeviceStatusText2", "DirectToSecurityView");
}

```

```

function _BuildDeviceHtml(name, image, id, link, status, statusClass,
onClickFun)
{
    var html = "";

    html += "<table class='mainDevicePanel' border='0' cellpadding='0'
cellspacing='0'>";
    html += "<tr><td class='mainDeviceName'>" + name +
"</td></tr>";
    html += "<tr><td>";
        html += "<table class='mainDeviceSubPanel' border='0'
cellpadding='0' cellspacing='0'>";
            html += "<tr>"
                var url =
"images/mobile/blue_static/device/" + image;
                html += "<td class='mainDeviceIcon'><div
class='mainDeviceImage' style='background-image: url(" + url+ ");'/></td>";
                if(link)
                {
                    html += "<td id='mainDeviceStatus"
+ id + "' class='" + statusClass + "'>" + status + "</td>";
                    var link = onClickFun + "(" +
id + ")";
                    html += "<td><input type='button'
class='mainDeviceButton' value='More' onclick='" + link + "' /></td>";
                }
                else
                {
                    html += status;
                }
            html += "</tr>";
        html += "</table>";
    html += "</td></tr>";
    html += "</table><br>";
    return html;
}

function PopulateMacros()
{
    var macroHTML = "";
    var macros = new Array();
    var orders = new Array();
    var n = 0;
    for (var i = 0; i < SystemMacros.length; i++)
    {
        var macro = SystemMacros[i];
        if (macro.getZoneID() == _selectedZoneId &&
macro.getShowInZone())
        {
            var order = macro.getOrder();
            orders[n++] = order;
            macros[order] = macro;
        }
    }
    orders.sort(_SortNumber);

    for(var i = 0; i < n; i++)

```

```

    {
        var index = orders[i];
        var macro = macros[index];
        if(macro.getEnabled()) {
            macroHTML += _BuildMacroHTML(macro);
        }
        else {
            macroHTML += _BuildDisabledDeviceHTML(macro);
        }
    }

    return macroHTML;
}

function _SortNumber(a,b)
{
    return a - b;
}

function _BuildMacroHTML(macro)
{
    var id = "mainMacroStatus" + macro.getID();
    var src = GetMacroStatusImage(macro.getState());
    var html = "<td class='mainDeviceStatus'><img id='" + id + "' " +
"src='" + src + "' /></td>";
    var run = "SendCommand('MacroCommand', " + macro.getID() + ",
'RunMacro')";
    var stop = "SendCommand('MacroCommand', " + macro.getID() + ",
'StopMacro')";
    var pause = "SendCommand('MacroCommand', " + macro.getID() + ",
'PauseMacro')";
    html += "<td class='mainDeviceButtonTd'><input type='button'
class='mainDeviceButton' value='Run' onclick=\"" + run + "\" /></td>";
    html += "<td class='mainDeviceButtonTd'><input type='button'
class='mainDeviceButton' value='Stop' onclick=\"" + stop + "\" /></td>";
    html += "<td><input type='button' class='mainDeviceButton'
value='Pause' onclick=\"" + pause + "\" /></td>";
    return _BuildDeviceHtml(macro.getName(), macro.getImage(), macro.getID(),
false, html, null, null);
}

function bodyLoad()
{
    //InitSoap();
    var n = _cameraImageIds.length;
    if(n > 0)
    {
        if(n > 3) n = 3;
        setTimeout("_UpdateCameraImages();", n * 1000);
    }
    else
    {
        _cameraImageIds = null;
    }
    //StartTimer(processEventChange);
    //InitXMLHttpRequest();
    setTimeout("QueryProcDeviceStatus();", 1000);
}

```

```

}

function _UpdateCameraImages()
{
    var n = _cameraImageIds.length;
    if(n > 0) {
        for(var i = 0; i < n; i++) {
            var imageId = _cameraImageIds[i];
            var deviceId = imageId.substr(7);
            //var deviceIndex = searchForID(SystemDevices, deviceId);
            //if(deviceIndex >= 0) {
                var img = GetCameraImage(deviceId, "S", 35,
35);
                eval(imageId).src=img;
            //}
        }
    }
    _cameraImageIds = null;
}

```

2. Source file: mobileBasicFunctions.js

```

//=====
/// <copyright from='2005' to='2008' company='Embedded Automation, Inc.'>
///
/// mControlWeb
///
/// Copyright (c) Embedded Automation, Inc. All rights reserved.
///
/// </copyright>
//
// $Id: mobileBasicFunctions.js,v 1.2 2008/04/22 19:25:05 george Exp $
//
//=====

var FAILED = -999; // Represents the failed sentinel value.

function getTimeString()
{
    var now = new Date();
    var dateString = now.getHours() + now.getMinutes() +
now.getSeconds();
    return dateString;
}

////////////////////////////////////
// Description: JS function to retrieve the matching query
variable
// Returns -999 when not found.
function getQueryVariable(variable) {
    var query = window.location.search.substring(1);
    var vars = query.split("&");
    for (var i=0;i<vars.length;i++) {
        var pair = vars[i].split("=");
        if (pair[0] == variable) {

```

```

        return pair[1];
    }
}
//alert('Query Variable ' + variable + ' not found');
return FAILED;
}

//gets the module baseName ID Number
function findIndexOf(searchArray, searchString)
{
    try
    {
        searchString = searchString.toUpperCase();
        for (k = 0; k < searchArray.length; k++)
        {
            if (searchArray[k].getBaseName().toUpperCase() ==
searchString)
            {
                return k;
            }
        }
        return FAILED;
    }
    catch (e) {}
    return FAILED;
}

////////////////////////////////////
// Description:      Searches a systemArray (Devices, Zones, etc..)
//                  using a database ID, and returns the
equivalent
//                  systemArray ID
function searchForID(searchArray, searchID)
{
    for (i = 0; i < searchArray.length; i++)          // Used to get
the location of current device in the xArray
    {
        if (searchArray[i].getID() == searchID)
            return i;
    }
    return -1;
}

////////////////////////////////////
// Description:      Searches a systemArray (Devices, Zones, etc..)
//                  using a database ID, and returns the
order 1 object
function searchForOrder(searchArray)
{
    for (i = 0; i < searchArray.length; i++)          // Used to get
the location of current device in the xArray
    {
        if (searchArray[i].getOrder() == 0)
            return searchArray[i].getID();
    }
}

```

```

function myDevice(id, zone, name, address, splitAddress, module, mprotocol,
image,
    notifyFlags, visible, adapter, status, order, options, extraData)
{
    var _id          = id;
    var _zone        = zone;
    var _name        = name;
    var _address     = address;
    var _splitAddress = splitAddress;
    var _module      = module;
    var _mprotocol   = mprotocol;
    var _image       = image;
    var _dimBy       = 10;
    var _notifyFlags = notifyFlags;
    var _visible     = visible;
    var _adapter     = adapter;
    var _status      = status;
    var _order       = order;
    var _extraData   = extraData;
    var _options     = options.split("|");
    var _port, _uname, _pwd, _anonymous;
    try {
        _port          = _options[0];
        _uname         = _options[1];
        _pwd           = _options[2];
        _anonymous     = _options[3];
    }
    catch (e) {
        _port = 80;
        _uname = _pwd = '';
        _anonymous = false;
    }
    _options = options;

    this.setID          = function (x) { _id
= x; }
    this.setZone        = function (x) { _zone
x; }
    this.setName        = function (x) { _name
x; }
    this.setAddress     = function (x) { _address
    }
    this.setSplitAddress = function (x) { _splitAddress = x; }
    this.setModule      = function (x) { _module
    }
    this.setMprotocol   = function (x) { _mprotocol
    }
    this.setImage       = function (x) { _image
x; }
    this.setPort        = function (x) { _port
x; }
    this.setDimBy       = function (x) { _dimBy
x; }
    this.setNotifyFlags = function (x) { _notifyFlags = x; }
    this.setVisible     = function (x) { _visible
    }
    this.setAdapter     = function (x) { _adapter
    }
}

```

```

        this.setStatus          = function (x) { _status          = x;
    }
        this.setOrder          = function (x) { _order          =
x; }
        this.setUname          = function (x) { _uname          =
x; }
        this.setPwd            = function (x) { _pwd
= x; }
        this.setAnonymous      = function (x) { _anonymous      = x; }
        this.setOptions        = function (x) { _options        = x;
}

        this.getID            = function () { return _id; }
        this.getZone           = function () { return _zone; }
        this.getName           = function () { return _name; }
        this.getAddress        = function () { return _address; }
        this.getSplitAddress    = function () { return _splitAddress; }
        this.getModule         = function () { return _module; }
        this.getMprotocol       = function () { return _mprotocol; }
        this.getImage          = function () { return _image; }
        this.getPort           = function () { return _port; }
        this.getDimBy          = function () { return _dimBy; }
        this.getNotifyFlags    = function () { return _notifyFlags; }
        this.getVisible        = function () { return _visible; }
        this.getAdapter        = function () { return _adapter; }
        this.getStatus          = function () { return _status; }
        this.getOrder          = function () { return _order; }
        this.getUname          = function () { return _uname; }
        this.getPwd            = function () { return _pwd; }
        this.getAnonymous      = function () { return _anonymous; }
        this.getOptions        = function () { return _options; }
        this.getExtraData      = function () { return _extraData; }
}

////////////////////////////////////
//  Data objects for zones
////////////////////////////////////
function myZone(id, name, order, image)
{
    var _id          = id
    var _name        = name;
    var _order       = order;

    this.setID       = function (x) { _id          = x; }
    this.setName     = function (x) { _name        = x; }
    this.setOrder    = function (x) { _order       = x; }

    this.getID       = function () { return _id; }
    this.getName     = function () { return _name; }
    this.getOrder    = function () { return _order; }
    this.getImage    = function () { return image; }
}

////////////////////////////////////
//  Data objects for macros
////////////////////////////////////

```

[illegible]

```

////////////////////////////////////
function myCondition(id, name, trigID, condType, condValStr)
{
    var _id          = id;
    var _name        = name;
    var _trigID      = trigID;
    var _condType    = condType;
    var _condValStr  = condValStr;

    this.setID       = function (x) { _id = x; }
    this.setTrigID   = function (x) { _trigID = x; }
    this.setCType    = function (x) { _condType = x; }
    this.setCondStr  = function (x) { _condValStr = x; }

    this.getID       = function () { return _id; }
    this.getName     = function () { return _name; }
    this.getTrigID   = function () { return _trigID; }
    this.getCType    = function () { return _condType; }
    this.getCondStr  = function () { return _condValStr; }
}

////////////////////////////////////
// Data objects for actions
////////////////////////////////////
function myAction(actionID, macroID, aname, atype,
                  command, enabled, aorder)
{
    var _actionID    = actionID;
    var _macroID     = macroID;
    var _aname       = aname;
    var _atype       = atype;
    var _command     = command;
    var _enabled     = enabled;
    var _aorder      = aorder;

    this.setID       = function (x) { _actionID = x; }
    this.setMacroID  = function (x) { _macroID = x; }
    this.setAname    = function (x) { _aname = x; }
    this.setAtype    = function (x) { _atype = x; }
    this.setCommand  = function (x) { _command = x; }
    this.setEnabled  = function (x) { _enabled = x; }
    this.setOrder    = function (x) { _aorder = x; }

    this.getID       = function () { return _actionID; }
    this.getMacroID  = function () { return _macroID; }
    this.getAname    = function () { return _aname; }
    this.getAtype    = function () { return _atype; }
    this.getCommand  = function () { return _command; }
    this.getEnabled  = function () { return _enabled ? 1 : 0; }
    this.getOrder    = function () { return _aorder; }
}

////////////////////////////////////
// Data objects for actions' commands
////////////////////////////////////
function myActionCommand(id, name, code, triggersrc, order)
{

```

```

    var _id          = id;
    var _name        = name;
    var _code        = code;
    var _triggersrc  = triggersrc;
    var _order       = order;

    this.setID       = function (x) { _id          = x; }
}

    this.setName     = function (x) { _name        = x; }
    this.setCode     = function (x) { _code        = x; }
    this.setTriggersrc = function (x) { _triggersrc = x; }
    this.setOrder    = function (x) { _order       = x; }

    this.getID       = function () { return _id; }
    this.getName     = function () { return _name; }
    this.getCode     = function () { return _code; }
    this.getTriggersrc = function () { return _triggersrc; }
    this.getOrder    = function () { return _order; }
}

////////////////////////////////////
// Data objects for mServer loaded adapters
////////////////////////////////////
function myAdapterRecord(name, displayas, description, aprotocol, port,
    ports, supports, canEnroll, opts)
{
    var _options = opts
    this.getBaseName     = function () { return name; }
    this.getDisplayas    = function () { return displayas; }
    this.getDescription = function () { return description; }
    this.getAprotocol    = function () { return aprotocol; }
    this.getPort         = function () { return port; }
    this.getPorts        = function () { return ports; }
    this.getSupports     = function () { return supports.split("|"); }
    this.getCanEnroll    = function () { return canEnroll; }
    this.getOptions      = function () { return _options; }
    this.setOptions      = function (x) { _options = x; }
}

////////////////////////////////////
// Data objects for known modules to mServer
////////////////////////////////////

// This is a readonly object so we don't implement any setters in it
function myModuleRecord(baseName, moduleClass, displayName, description,
    moduleProtocol, defaultImage, capabilityFlags, groupMin, groupMax,
    fieldGroup, conditionGroup, cmdUiBtnGrp)
{
    var _capsFlag = capabilityFlags;

    this.getBaseName     = function () { return baseName; }
    this.getModuleClass  = function () { return moduleClass; }
    this.getDisplayName  = function () { return displayName; }
    this.getDescription = function () { return description; }
    this.getModuleProtocol = function () { return moduleProtocol; }
    this.getDefaultImage = function () { return defaultImage; }
    this.getGroupMin     = function () { return groupMin; }

```

```

        this.getGroupMax          = function () { return groupMax; }
        this.getFieldGroup       = function () { return fieldGroup; }
        this.getConditionGroup   = function () { return conditionGroup; }
        this.getCmdGroup         = function () { return cmdUiBtnGrp; }

        // bit-field retrieval functions
        this.getIsDimmable       = function () { return _capsFlag &
BIT_IsDimmable; }
        this.getPresetOn        = function () { return _capsFlag &
BIT_HasPresetOn; }
        this.getRampRate        = function () { return _capsFlag &
BIT_HasRampRate; }
        this.getIsGroupable     = function () { return _capsFlag &
BIT_HasGrpTrigger; }
        this.getCanDoCondition  = function () { return _capsFlag &
BIT_CanDoCondition; }
        this.getUsesActiveX     = function () { return _capsFlag &
BIT_UsesActiveX; }
        this.getDetectsMotion   = function () { return _capsFlag &
BIT_DetectsMotion; }
        this.getHasAudio        = function () { return _capsFlag &
BIT_HasAudio; }
        this.getCanPan          = function () { return _capsFlag &
BIT_CanPan; }
        this.getCanTilt         = function () { return _capsFlag &
BIT_CanTilt; }
        this.getCanZoom         = function () { return _capsFlag &
BIT_CanZoom; }
        this.getCanJPeg         = function () { return _capsFlag &
BIT_CanJPeg; }
        this.getCanMJPEG        = function () { return _capsFlag &
BIT_CanMJPEG; }
    }

    ///////////////////////////////////////////////////////////////////
    //  Data objects for modules' UI buttons in mServer
    ///////////////////////////////////////////////////////////////////

    // This is a readonly object so we don't implement any setters in it
    function myModuleCmdRecord(grpName, command, description, uiclass, urlPage)
    {
        this.getGrpName = function () { return grpName; }
        this.getCommand = function () { return command; }
        this.getDescription = function () { return description; }
        this.getUIClass = function () { return uiclass; }
        this.getUrlPage = function () { return urlPage; }
    }

    ///////////////////////////////////////////////////////////////////
    //  Data objects for condition property in mServer
    ///////////////////////////////////////////////////////////////////

    // This is a readonly object so we don't implement any setters in it
    function myCondPropRecord(groupname, propertyname, propType, allowedOps,
allowedVals, isChangeProp, isStatusProp)
    {

```

```

        this.getName                = function () { return groupname; }
    }
    this.getProperty                = function () { return propertyname; }
    }
    this.getType                   = function () { return propType; }
    }
    this.getOperation               = function () { return allowedOps; }
    this.getValue                  = function () { return allowedVals; }
    }
    this.getValidForChange         = function () { return isChangeProp; }
    this.getValidForStatus         = function () { return isStatusProp; }
    }

    //////////////////////////////////////
    //  Data objects for modules group of fields holder (address parts
    holder)
    //////////////////////////////////////

    // This is a readonly object so we don't implement any setters in it
    function myModuleFieldGroup(grpname, name, addressJoiner, enrollTimeout,
    unenrollTimeout)
    {
        this.getGrpname            = function () { return grpname; }
    }
        this.getName                = function () { return name; }
    }
        this.getAddressJoiner       = function () { return addressJoiner; }
    }
        this.getEnrollTimeout        = function () { return enrollTimeout; }
    }
        this.getUnenrollTimeout      = function () { return unenrollTimeout; }
    }
    }

    //////////////////////////////////////
    //  Data objects for modules fields (address and advanced options)
    //////////////////////////////////////

    // This is a readonly object so we don't implement any setters in it
    function myModuleComponentRecord (label, type, values, errmsg, defaultval)
    {
        this.getLabel              = function () { return label; }
        this.getType               = function () { return type; }
        this.getValues             = function () { return values; }
        this.getErrMsg             = function () { return errmsg; }
        this.getDefault            = function () { return defaultval; }
    }
    }

    //////////////////////////////////////
    //  Data objects for a group of condition property in mServer
    //////////////////////////////////////

    // This is a readonly object so we don't implement any setters in it
    function myCondGrpRecord(name, propsArray)
    {
        this.getName               = function () {return name; }
        this.getProperty           = function () {return propsArray; }
    }

```

```

}

// This is a readonly object so we don't implement any setters in it
function myLanguageRecord (_id, _name, _local)
{
    this.getID          = function () { return _id; }
    this.getName        = function () { return _name; }
    this.getLocal       = function () { return _local; }
}

////////////////////////////////////
// System Users objects
////////////////////////////////////
function myUserAcct(id, name, image)
{
    var _name          = name;
    var _image          = image;

    this.setName       = function (x) { _name      = x; }
    this.setImage      = function (x) { _image     = x; }

    this.getID         = function () { return id; }
    this.getName       = function () { return _name; }
    this.getImage      = function () { return _image; }
}

////////////////////////////////////
// Data objects for a protocols definitions in mServer
////////////////////////////////////

// This is a readonly object so we don't implement any setters in it
function myProtocolRecord(name, canEnroll, forcedAddressList)
{
    this.getName              = function () {return name;
}
    this.getCanEnroll         = function () {return canEnroll;
}
    this.forcedAddressList    = function () {return
forcedAddressList; }
}

```

3. Source file: mobileCommon.js

```

//=====
/// <copyright from='2005' to='2008' company='Embedded Automation, Inc.'>
/// mControlWeb
/// Copyright (c) Embedded Automation, Inc. All rights reserved.
/// </copyright>
/// $Id: mobileCommon.js,v 1.8 2008/04/02 17:02:25 roger Exp $
//
//=====

var _deviceIndex;
var _deviceId;

function BrowseBack()
{

```

```

        //window.location.href=("Mobile.aspx");
        window.location.href = "Mobile.aspx?Zone=" +
SystemDevices[_deviceIndex].getZone();
    }

function setComboBoxValue(comboBox, value)
{
    var options = comboBox.options;
    var index = -1;
    for(var i = 0; i < options.length; i++)
    {
        if(options[i].value == value)
        {
            index = i;
            break;
        }
    }
    comboBox.selectedIndex = index;
}

function GetElementInnerText(element)
{
    var text;
    if(typeof(element.innerText) == "undefined")
    {
        text = element.innerHTML;
    }
    else
    {
        text = element.innerText;
    }
    return text;
}

function SetElementInnerText(element, text)
{
    if(typeof(element.innerText) == "undefined")
    {
        element.innerHTML = text;
    }
    else
    {
        element.innerText = text;
    }
}

function SendCommand(commandType, devid, commandStr)
{
    if(commandType == "AutomationCommand")
    {
        doServiceRequest(_actionObjHttp, commandType, "function=" +
escape(commandStr) + "<deviceId>" + devid + "</deviceId>" + "<command>" +
commandStr + "</command>" );
    }
    else if(commandType == "MacroCommand")
    {

```

```

        doServiceRequest(_actionObjHttp, commandStr, "<macroId>" + devid
+ "</macroId>");
    }
}

```

4. Source file: mobileEventsLogic.js

```

//=====
/// <copyright from='2005' to='2008' company='Embedded Automation, Inc.'>
///
///    mControlWeb
///
///    Copyright (c) Embedded Automation, Inc.    All rights reserved.
///
/// </copyright>
//
//    $Id: mobileEventsLogic.js,v 1.3 2008/04/22 19:25:05 george Exp $
//
//=====

var _soapServer = "";
var _soapPath = "";
var _timerObjHttp = false;
var _actionObjHttp = false;
var _xmlDocEvent = false;

function InitXMLHttpRequest()
{
    if (document.implementation && document.implementation.createDocument) {
        // Mozilla, create a new DOMParser and document by prsing
the XML string
        _xmlDocEvent=document.implementation.createDocument("", "", null);
        //xmlDoc = new DOMParser().parseFromString(xmlString,
"text/xml");
        //alert("EVENTS_LOGIC: This browser is not supported");
        //return;
    }
    else if (window.ActiveXObject) {
        // Internet Explorer, create a new XML document using
ActiveX
        _xmlDocEvent = new ActiveXObject("Microsoft.XMLDOM")
        _xmlDocEvent.async="false";
    }
    _timerObjHttp = CreateXMLHttpRequest();
    _actionObjHttp = CreateXMLHttpRequest();
    _soapServer = location.host;
    var soapPath = location.pathname;
    var p = soapPath.lastIndexOf("/");
    if (p >= 0) {
        soapPath = soapPath.substring(0, p);
    }
    _soapPath = soapPath + "/mServer.asmx";
}

function QueryProcDeviceStatus()

```

```

{
    _OnTimerEvent();
    setTimeout("QueryProcDeviceStatus();", 1000);
}

function _OnTimerEvent() {
    try
    {
        if (doServiceRequest(_timerObjHttp, "GetQueuedEvents", null)) {
            _ProcessEventsRequest();
        }
    } catch (e) {
        // server might be down and we are unable to connect to
it
    }
}

function doServiceRequest(objXmlHttp, method, paramStr) {
    if (objXmlHttp) {
        try {
            var soapStr =
                '<?xml version="1.0" encoding="utf-8"?>' +
                '<soap:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">' +
                '<soap:Body>' +
                '<' + method + '
xmlns="http://embeddedautomation.com/webservices/mControl">' +
                ' + ((paramStr == null || paramStr
== "") ? '' : paramStr) +
                '</' + method + '>' +
                '</soap:Body>' +
                '</soap:Envelope>';
            // call the web service
            objXmlHttp.open("POST", location.protocol + "/" +
_soapServer + _soapPath, false); // true/false for asynchronous
            objXmlHttp.setRequestHeader ("Man", "POST " + _soapPath
+ " HTTP/1.1");
            objXmlHttp.setRequestHeader ("Host", _soapServer);
            objXmlHttp.setRequestHeader("Content-Type", "text/xml;
charset=utf-8");
            objXmlHttp.setRequestHeader("SOAPAction",
"http://embeddedautomation.com/webservices/mControl/" + method);
            objXmlHttp.send(soapStr);

            while (objXmlHttp.readyState != 4) {
                // should spend at most 5 seconds here -
otherwise exit
            }
            if (objXmlHttp.status == 200) {
                return true;
            } else {
                /*** - has to be changed to a proper notification
for MCE and IE
                soapServer + "/" + method + "?" + paramStr
                /* alert("There was a problem with the service:\n" +

```

```

        + "\n\n" + objXmlHttp.statusText +
"\n\t- " + objXmlHttp.responseText);
    }
    } catch (e) {
        setErrorInfo('mServer is down. Restart it & refresh
the page.', '#f00000');
    }
}

return false;
}

function _ProcessEventRequest() {
    // retrieve the string returned by the web service (as XML
string, not as node)
    // then process the response parsing the string as XML document
    var xmlstr =
_timerObjHttp.responseXML.getElementsByTagName("GetQueuedEventsResult")[0].firstChild.data;
    // to speed up processing, we will rely on the length of
returned string
    // an empty events element will be
"<QueuedEvents>\n</QueuedEvents>\n"
    // while a non-event will be much longer, so if length is less
than
    // length("<QueuedEvents>\n</QueuedEvents>\n") [approx. 40 chars
// (including some extra margin)] then our string has no events.
// This way is faster than processing the whole document as xml,
etc.
    if (xmlstr == null || xmlstr.length < 40) {
        return
    }

    if (_GetXmlDocumentFromString(xmlstr))
    {
        _ProcessEventChange(_xmlDocEvent);
    }
}

function _GetXmlDocumentFromString(xmlString) {
    var ok = false;
    try
    {
        if (_xmlDocEvent)
        {
            if (window.ActiveXObject)
            {
                _xmlDocEvent.loadXML(xmlString); // Use loadXML
as a DOM parser.
                if (_xmlDocEvent.parseError.errorCode == 0)
                    ok = true;
            }
            else
            {
                _xmlDocEvent = new
DOMParser().parseFromString(xmlString, "text/xml");
                ok = true;
            }
        }
    }
}

```

```

        }
    }
    catch (e) { }
    return ok;
}

function _ProcessEventChange(xmlDoc) {
    try
    {
        var objNodeList = xmlDoc.getElementsByTagName("event-change");
        var objNodeList2 = xmlDoc.getElementsByTagName("macro-change");

        if (objNodeList)
        {
            for (var i = 0; i<objNodeList.length; i++)
            {
                var chgArray =
objNodeList[i].firstChild.data.split("#");
                eventChangeStatus(chgArray[0], chgArray[1]);
            }
        }

        if (objNodeList2)
        {
            for (var i = 0; i<objNodeList2.length; i++)
            {
                var chgArray =
objNodeList2[i].firstChild.data.split("#");
                macroChangeStatus(chgArray[0], chgArray[1]);
            }
        }
    }
    catch(e) { }
}

function CreateXMLHttpRequest()
{
    if(window.XMLHttpRequest)
    {
        return new XMLHttpRequest();
    }
    else if(window.ActiveXObject)
    {
        var msxmls = new Array(
            'Microsoft.XMLHTTP',
            'Msxml2.XMLHTTP.5.0',
            'Msxml2.XMLHTTP.4.0',
            'Msxml2.XMLHTTP.3.0',
            'Msxml2.XMLHTTP');

        for(var i = 0; i < msxmls.length; i++)
        {
            try
            {
                return new ActiveXObject(msxmls[i]);
            }
        }
    }
}

```

```

        catch (e) { }
    }

    throw new Error("Could not instantiate XMLHttpRequest");
}

```

5. Source file: mobileLightObject.js

```

//=====
/// <copyright from='2005' to='2008' company='Embedded Automation, Inc.'>
///
///     mControlWeb
///
///     Copyright (c) Embedded Automation, Inc.    All rights reserved.
///
/// </copyright>
//
//     $Id: mobileLightObject.js,v 1.1 2008/04/02 00:00:36 roger Exp $
//
//=====

function GetSwitchStatusImage(level)
{
    return GetDimmerStatusImage((level > 0) ? 100 : 0);
}

function GetDimmerStatusImage(level)
{
    if(level < 0) level = 0;
    if(level > 100) level = 100;
    var levelRounded = Math.round(level / 10) * 10;
    if(level < 0) levelRounded = -1;

    return "images/mobile/blue_static/Status." + levelRounded + ".gif";
}

```

6. Source file: mobileMainStatus.js

```

//=====
/// <copyright from='2005' to='2008' company='Embedded Automation, Inc.'>
///
///     mControlWeb
///
///     Copyright (c) Embedded Automation, Inc.    All rights reserved.
///
/// </copyright>
//
//     $Id: mobileMainStatus.js,v 1.1 2008/04/02 00:00:36 roger Exp $
//
//=====

function eventChangeStatus(deviceId, status)
{
    try

```

```

    {
        var deviceIndex = searchForID(SystemDevices, deviceId);
        var device = SystemDevices[deviceIndex];
        device.setStatus(status);

        if (device.getZone() == _selectedZoneId)
        {
            _UpdateDeviceStatus(device);
        }
    }
    catch (Exception) { }
}

function _UpdateDeviceStatus(device)
{
    var deviceHtmlObj = eval("mainDeviceStatus" + device.getID());
    if(!deviceHtmlObj) return;

    var modID = findIndexOf(SystemModules, device.getModule());
    switch (SystemModules[modID].getModuleClass().toUpperCase())
    {
        case "BINSWITCH":
            var level = parseInt(device.getStatus());
            deviceHtmlObj.src = GetSwitchStatusImage(level);
            break;
        case "MLVLSWITCH":
            var level = parseInt(device.getStatus());
            deviceHtmlObj.src = GetDimmerStatusImage(level);
            break;
        case "THERMOSTAT":
            SetElementInnerText(deviceHtmlObj,
GetThermostatTemperature(device.getStatus(), false));
            break;
        case "ALARMSYS":
            SetElementInnerText(deviceHtmlObj,
GetSecurityState(device.getStatus()));
            break;
    }
}

function macroChangeStatus(id, state)
{
    try
    {
        var macroIndex = searchForID(SystemMacros, id);
        var macro = SystemMacros[macroIndex];
        macro.setState(state);
        if (macro.getZoneID() == _selectedZoneId)
        {
            _UpdateMacroStatus(macro);
        }
    }
    catch (Exception) { }
}

function GetMacroStatusImage(state)
{

```

```

        return "images/mobile/blue_static/Mac." + state + ".png";
    }

    function _UpdateMacroStatus(macro)
    {
        var macroHtmlObj = eval("mainMacroStatus" + macro.getID());
        if(!macroHtmlObj) return;
        macroHtmlObj.src = GetMacroStatusImage(macro.getState());
    }

```

APPENDIX III

THE main server generated web page uses ActiveX scripting and the corresponding file is mobile.aspx as stated before. The control flow is as shown:

