Getting Started with Lab of Things

Contents

Overview

This guide covers everything you need to know to get up and running with Lab of Things. In particular you will learn how to set up a Home Hub to try out Lab of Things. For details about developing your own Lab of Things applications, see Lab of Things Developer Guide on Codeplex.

BACKGROUND

About Lab of Things and HomeOS

Lab of Things is a shared infrastructure designed to help researchers develop and evaluate technologies in the home environment. The short video at http://www.lab-of-things.com/ is the best overview. Lab of Things provides a common framework to write applications and has a set of capabilities beneficial to field deployments including logging application data from houses in cloud storage, remote monitoring of system health, and remote updating of applications if needed (e.g. to change to a new phase of the study by enabling new software, or to fix bugs).

Each household in Lab of Things runs HomeOS on dedicated computer, the Home Hub, which interacts with the in-home sensors and hosts the applications needed for the study (or studies) in which the household is participating. HomeOS provides a PC-like abstraction for in-home hardware and simplifies the tasks of writing applications and managing sensors.

Glossary

Lab of Things: A flexible platform for research that uses connected devices in homes (and in the future, other physical spaces such as commercial buildings). Lab of Things provides researchers with the ability to:

- Easily interconnect devices and implement application scenarios.
- Conduct field studies at scale through cloud services that can monitor and update experiments and provide easy access to collected data
- Share data, code, and participants, lowering the barrier to evaluating ideas in a diverse range of settings

It consists of two main technology areas: HomeOS and a set of cloud services.

HomeOS: A software platform that provides a PC-like abstraction for in-home hardware and simplifies the tasks of writing applications and managing sensors. In our documentation we often refer to HomeOS as just “**platform**”.

Dashboard: The user interface for the HomeOS. In the Dashboard you can install new applications, manage settings, and install and configure the sensors that will be connected to the Home Hub.

Home Hub: A dedicated computer that runs the HomeOS platform in a Lab of Things household.

OrgID: A unique identifier representing a Lab of Things participant. This can be either an individual or an organization.

Prerequisites for Lab of Things

To run HomeOS on a home hub:

- Windows 8/8.1 or Windows 7 with a wireless card
- Microsoft .NET Framework 4.5
- Microsoft Sync Framework 2.1 Redistributable Package (X86 only)
- For **development** you will need all of the above plus Visual Studio 2012 and Windows Azure SDK 2.1

Getting Started

Lab of Things is a research platform aimed at enabling field studies of connected devices more easily. We also welcome DIYers who like to work with devices around homes and beyond. You will need Visual Studio 2012 to build the source code before you can run it.

If you would like try running HomeOS without compiling code, you can go to the project's Downloads section on Codeplex, and then get the Release.zip. Unzip the file and run the code per the instructions (gettingstarted.text) included in the zip file.

To get the source code go to https://labofthings.codeplex.com/. The source control system for the project is Git. To Git Clone the Source Code, from a Git command prompt, clone https://git01.codeplex.com/labofthings. For more information about cloning, see Git Basics - Getting a Git Repository.

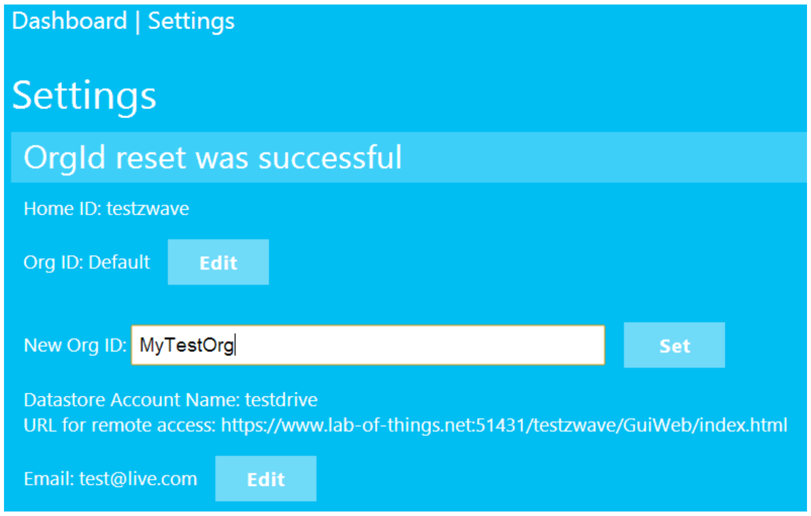
Get an Org ID

An Org ID is used to associate your hubs with your organization. You will need an Org ID in order to use the remote access feature and monitor your hubs. You can go to https://www.lab-of-things.net to register for your Org ID.

With an Org ID you'll be able to use the remote monitoring feature on our Remote Management Portal, which lets you view the statuses of your deployed hubs.

Important Note: Once you get your Org Id, you need to make sure your hub has it. You set your Org ID in one of the following two ways:

> In the Dashboard, go to Setting and click the **Edit** button next to **Org ID**. Change it to what you've successfully requested from the Management Portal (by default it is "Default")



Dashboard | Settings

Settings

OrgId reset was successful

Home ID: testzwave

Org ID: Default

New Org ID: MyTestOrg

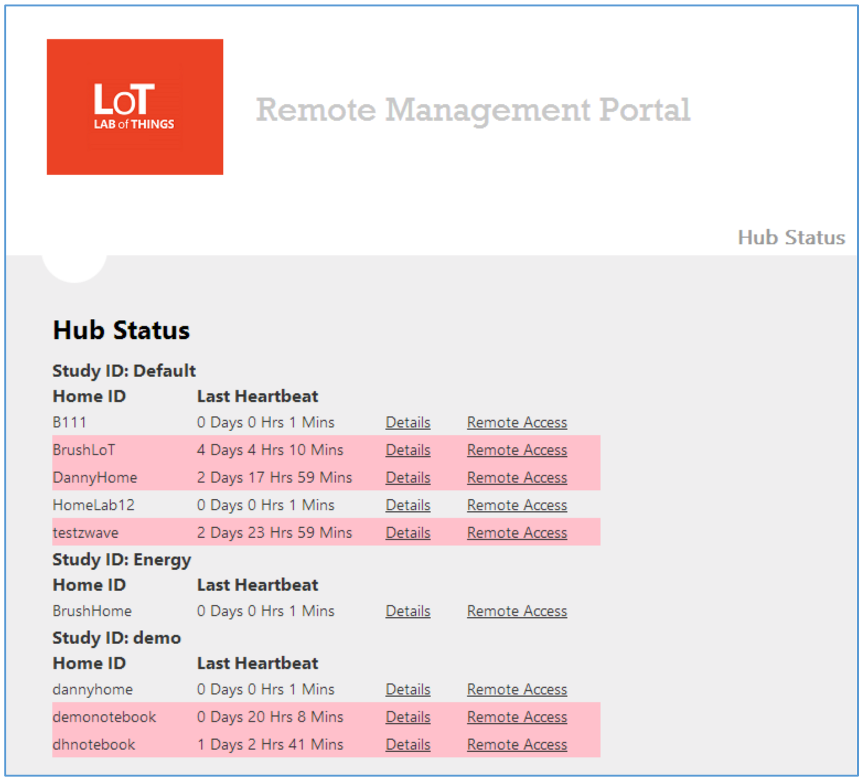
Datastore Account Name: testdrive
URL for remote access: https://www.lab-of-things.net:51431/testzwave/GuiWeb/index.html

Email: test@live.com

Or

You can put the Org ID in the settings.xml (Hub\output\configs\settings.xml). Edit the following entry: <Param Name="OrgId" Value="[your org id here]" />

LoT Remote Management Portal – it gives you a view of hubs you've deployed.



Hub Status

Study ID: Default

Home ID	Last Heartbeat		
B111	0 Days 0 Hrs 1 Mins	Details	Remote Access
BrushLoT	4 Days 4 Hrs 10 Mins	Details	Remote Access
DannyHome	2 Days 17 Hrs 59 Mins	Details	Remote Access
HomeLab12	0 Days 0 Hrs 1 Mins	Details	Remote Access
testzwave	2 Days 23 Hrs 59 Mins	Details	Remote Access

Study ID: Energy

Home ID	Last Heartbeat		
BrushHome	0 Days 0 Hrs 1 Mins	Details	Remote Access

Study ID: demo

Home ID	Last Heartbeat		
dannyhome	0 Days 0 Hrs 1 Mins	Details	Remote Access
demonotebook	0 Days 20 Hrs 8 Mins	Details	Remote Access
dhnotebook	1 Days 2 Hrs 41 Mins	Details	Remote Access

Build and Run HomeOS Solution

1. Start Visual Studio as Administrator

2. Load **Core.sln** from \homeos2\Hub\ and build the solution. Note, by default the binaries will be built to homeos2\Hub\output directory

   <you can skip step 1 & 2 if you downloaded the output binaries from Downloads on Codeplex>

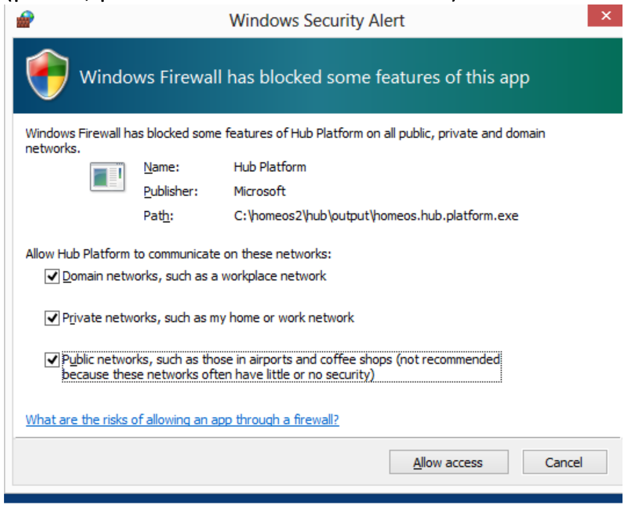
3. Open the Command Prompt as Administrator

4. Navigate to homeos2\Hub\output\

5. Run **startplatform.bat**.

   By default the platform is logging activities in \Hub\output\Data\platform\ homeos.log.

   Important Note:
   When you run platform for the first time, you will get a Windows Security Alert warning you that Windows Firewall is trying to block Hub Platform features. It's important to check all options (public, private and domain networks).



Windows Security Alert

Windows Firewall has blocked some features of this app

Windows Firewall has blocked some features of Hub Platform on all public, private and domain networks.

Name: Hub Platform
Publisher: Microsoft
Path: C:\homeos2\hub\output\homeos.hub.platform.exe

Allow Hub Platform to communicate on these networks:

[✓] Domain networks, such as a workplace network

[✓] Private networks, such as my home or work network

[✓] Public networks, such as those in airports and coffee shops (not recommended because these networks often have little or no security)

6. Now that the platform is running, you can set up your home hub by clicking on the Dashboard shortcut which is `located` in [root]\Hub\output\. This shortcut points to this url: http://localhost:51430/GuiWeb/index.html. From there you can start setting up your hub.

Run Platform in the debugger

1. Launch Visual Studio as an Administrator.
2. Open Core.sln.
3. Ensure that Platform (\Hub\Platform\Platform) is set as the startup project.
4. Press F5

Stop the Platform from the command prompt

At the console, type `exit` or `quit`.

(To get help for commands, type `help`.)

Reset the Platform

If the platform gets into a bad state, you can reset the information. This will delete all of the information that you entered the first time you can the platform.

1. Open an elevated command prompt.
2. Navigate to `homeos2\Hub\output\`.
3. Open an elevated command prompt.
4. Run the following command:
   ```
   reset.bat [name of config to be used]
   ```

If you run this command: `reset.bat Config`
The batch file will copy fresh copy of configuration files from Hub\Platform\Configs\Config to Hub\output\Configs\Config

Note: here's a list of useful batch files in `homeos2\Hub\output` to start, stop, and reset the platform:

- startplatform.bat: Starts the platform.
- reset.bat: Resets the platform to its default state. (you can achieve the same thing by deleting the entire output folder and then rebuild the solution again)
- starthostednetwork.bat: Run this to set up Gadgeteer devices.
- stophostednetwork.bat: Run this to stop the hosted network.

HomeOS Dashboard

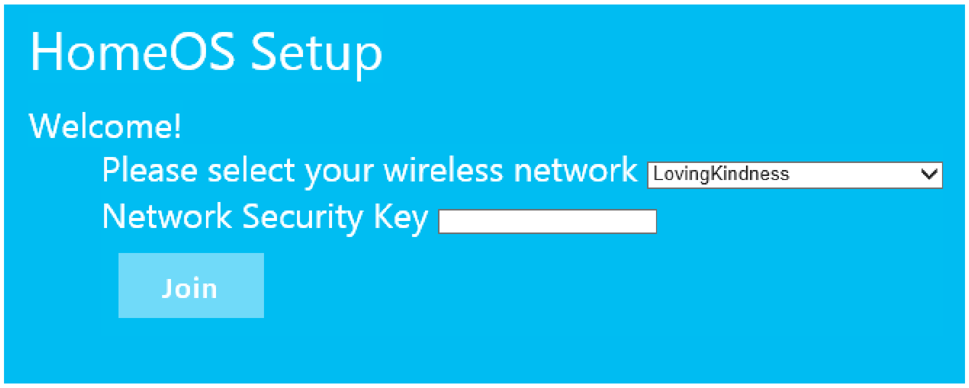
The Dashboard is the user interface for the Home Hub. In the Dashboard you can set up your hub and install new devices and applications. To access the dashboard, navigate to `http://localhost:51430/guiweb/` in your browser. (There's also a shortcut in the Hub/output that points to this url too)

Important Note: HomeOS **platform** must be running for the Dashboard to work.

HomeOS Setup in Dashboard

When you launch the dashboard for the first time, you will be prompted to select a wireless network and enter some information.

1. Choose a wireless network and enter the network's security key, then click **Join**.

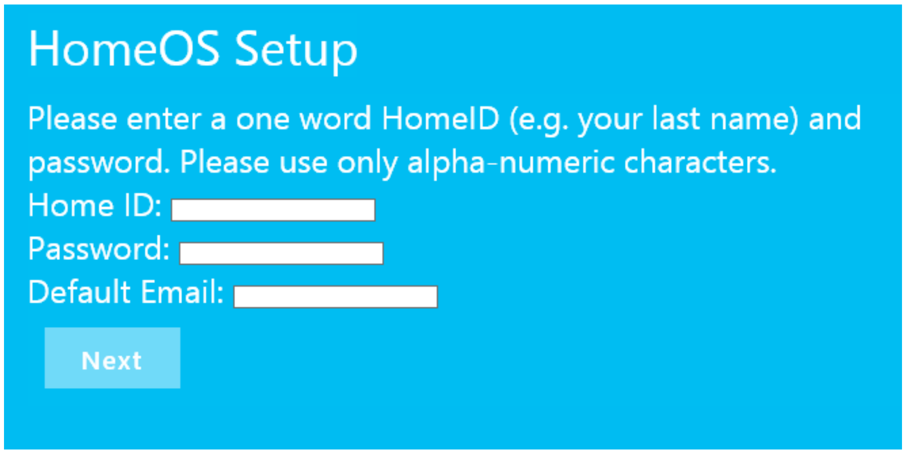

HomeOS Setup

Welcome!

Please select your wireless network LovingKindness

Network Security Key

2. Enter a HomeID, password, and the email address to use for contact information, then click **Next**.


HomeOS Setup

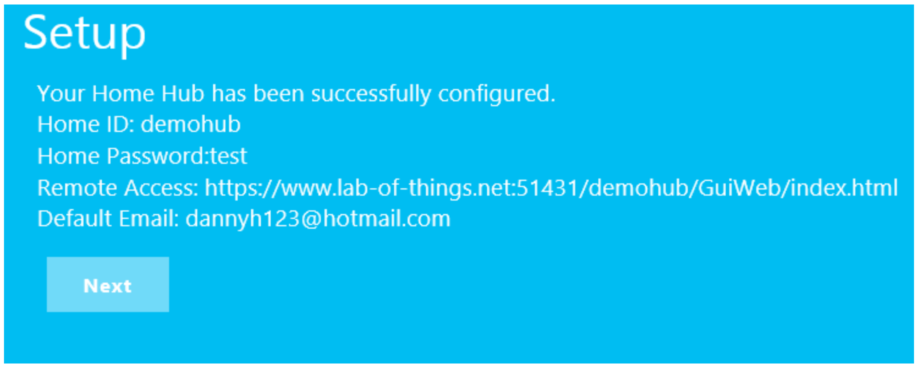
Please enter a one word HomeID (e.g. your last name) and password. Please use only alpha-numeric characters.

Home ID:

Password:

Default Email:

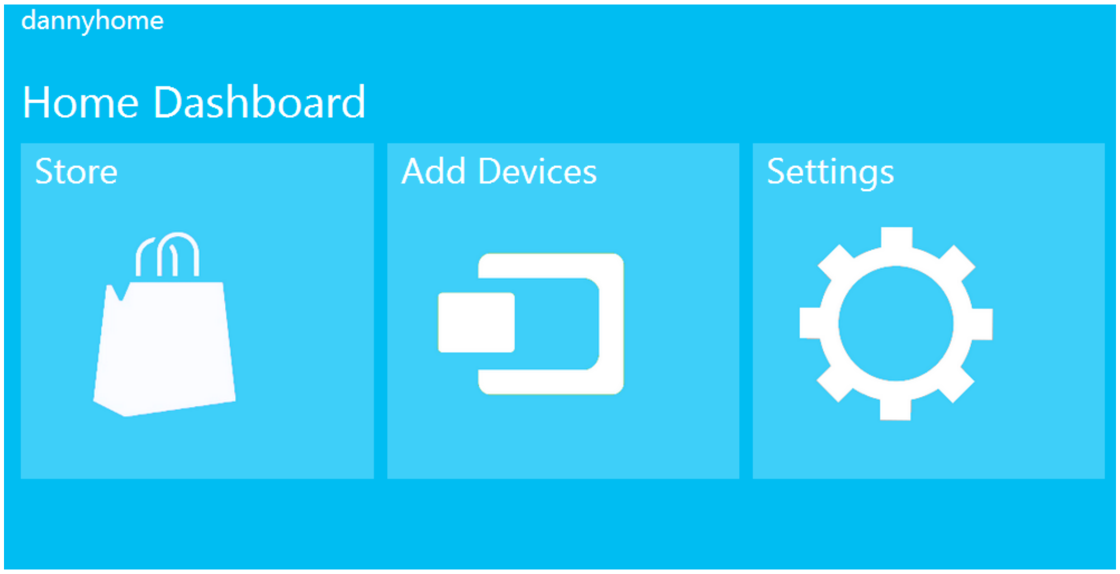
3. Your Dashboard information will be displayed. Copy this information for future reference.


Setup

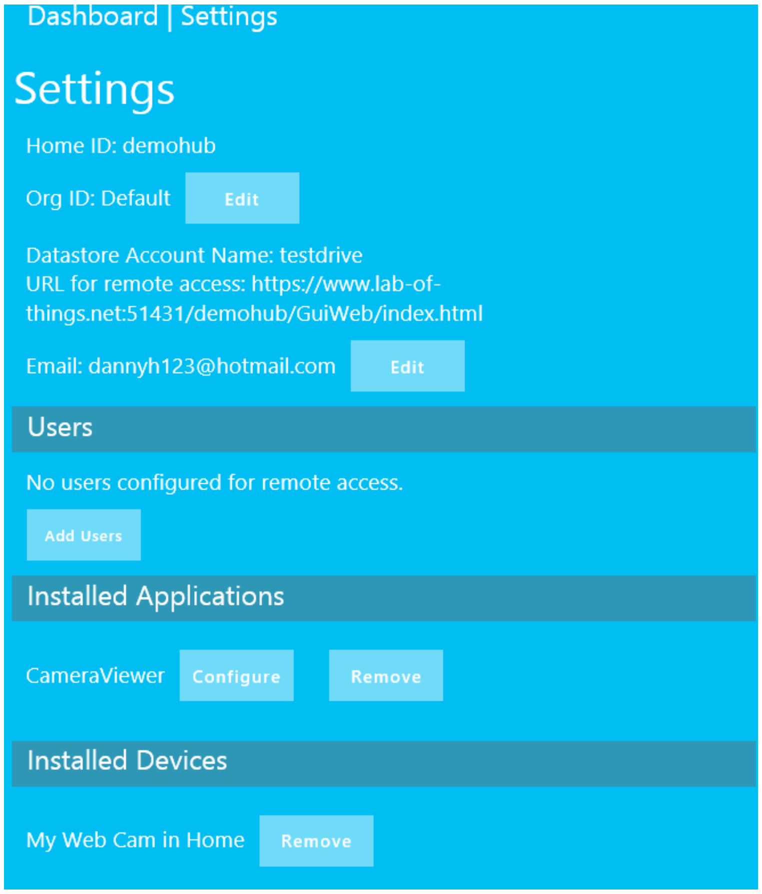
Your Home Hub has been successfully configured.
Home ID: demohub
Home Password:test
Remote Access: https://www.lab-of-things.net:51431/demohub/GuiWeb/index.html
Default Email: dannyh123@hotmail.com

4. Click **Next** to finish.

5. You will see the main screen for the Dashboard:



Settings


Dashboard | Settings

Settings

Home ID: demohub

Org ID: Default

Datastore Account Name: testdrive
URL for remote access: https://www.lab-of-things.net:51431/demohub/GuiWeb/index.html

Email: dannyh123@hotmail.com

Users

No users configured for remote access.

Installed Applications

CameraViewer

Installed Devices

My Web Cam in Home

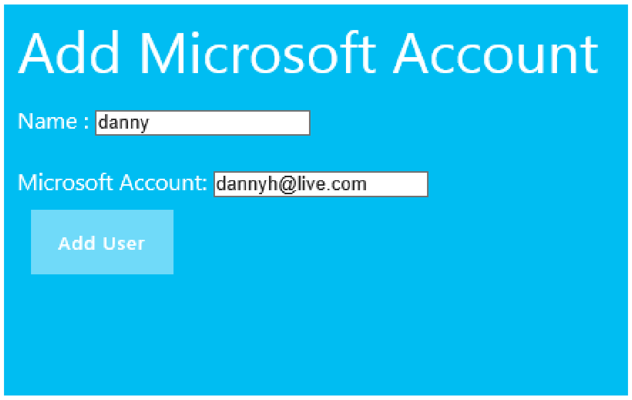
Setting page allows you to view and edit important configuration information about the hub:

- Enter or change your Org ID.

- See the Azure account being used for cloud data storage (the default is “testdrive” which is a test account and its contents are deleted regularly) – to change the account name and account key, you have to go to settings.xml in Hub/output/Configs/[your config – default is “Config”]/ Your remote access URL is listed here; before you can access the hub remotely, you will need to add a user (see below).

- Email listed here is the NotificationEmail which is the default notification email an app will use to send an email alert.

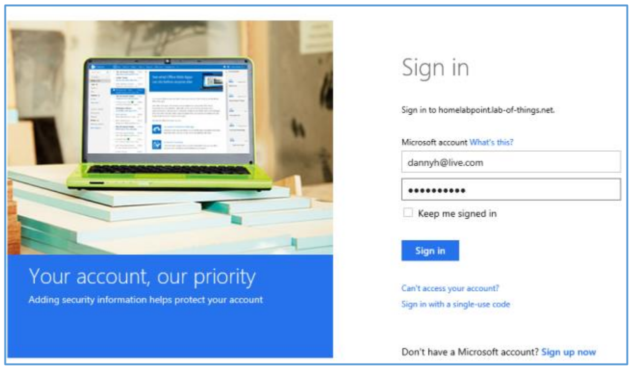
- Add/remove users. The users you add here will be able to remotely access the hub (Dashboard and apps) using the **URL for remote access** shown on this page.
  Important Note: the user you wish to add must be a valid Microsoft Account, and you will have to sign in and get authenticated by the Microsoft Account service to finish adding the user. See the screenshots below:



Add Microsoft Account

Name : danny

Microsoft Account: dannyh@live.com



Sign in

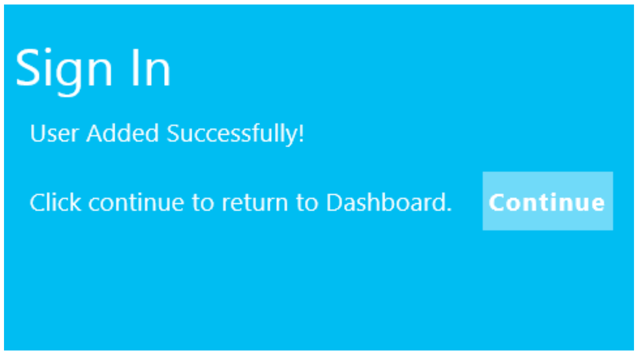
Sign in to homelabpoint.lab-of-things.net.

Microsoft account What's this?

dannyh@live.com

••••••••••

Keep me signed in
Sign in
Can't access your account?
Sign in with a single-use code
Don't have a Microsoft account? Sign up now
Your account, our priority
Adding security information helps protect your account


Sign In

User Added Successfully!

Click continue to return to Dashboard.

- See configure or remove a list of applications installed.

- See a list of installed devices

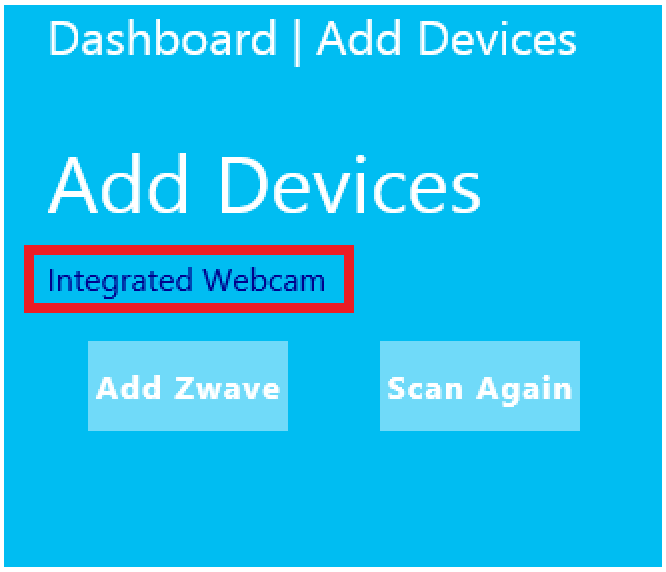
Add Devices

Currently four types of hardware devices are supported:

- Webcam
- Foscam Cameras
- Z-Wave Devices
- Microsoft .NET Gadgeteer

Adding Devices (Non Zwave)

- Going to the Add Devices page
- A list of unconfigured devices will be displayed. (When you run HomeOS for the first time, most likely the only device that will be discovered by the platform is a web cam, if your PC or notebook has one.)

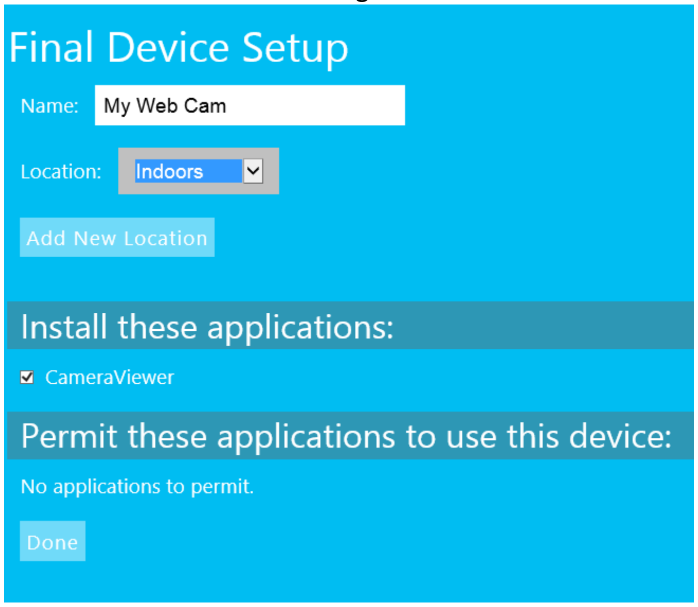

Dashboard | Add Devices

Add Devices

Integrated Webcam

- Click on the link of the discovered device; in this case “Integrated Webcam”
  Now configure the device:

  1. Enter a name for the device and select a location.
  2. Check the appropriate checkboxes to install application that is compatible with the device, and/or to provide permission for other applications to access the device. In our example here, the CameraViewer app shows up because it can use a webcam’s capability.
  3. Click **Done** to finish adding the device.


Final Device Setup

Name: My Web Cam

Location: Indoors

Install these applications:

☑ CameraViewer

Permit these applications to use this device:

No applications to permit.

Z-Wave Devices

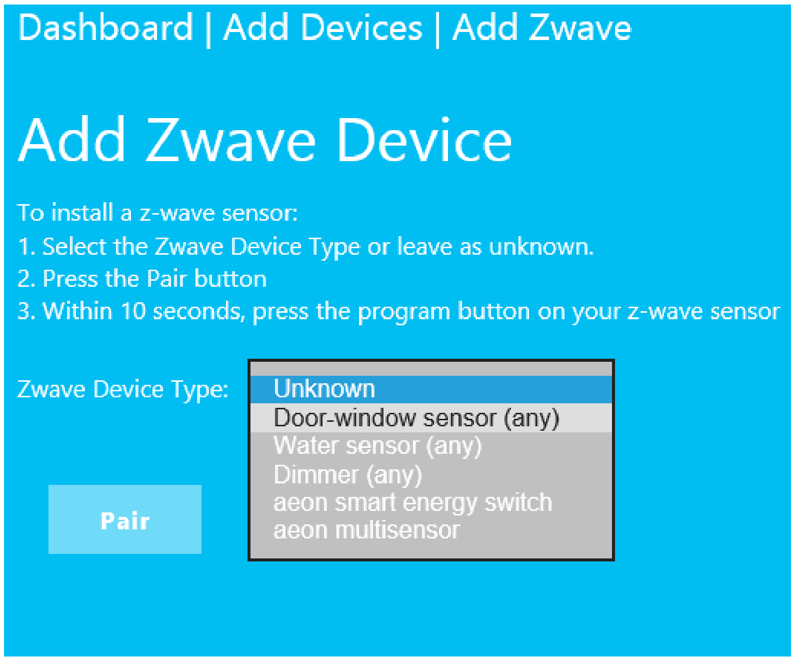
Z-Wave is a next-generation wireless ecosystem that lets all your home electronics talk to each other, and to you, via remote control. The process for adding a Z-Wave device involves pairing, so the steps are a bit different.

Important Note
By default the Z-Wave driver module for HomeOS is not included in the solution. Please contact lab-of-things@microsoft.com for more information on where to obtain the module.

Building Z-Wave Driver
Once you’ve acquired the driver, to get the driver running in the platform do the following:
- Place the Zwave driver folder under Hub\Drivers\
- In Visual Studio, add the .csproj file to the Drivers folder of the Core.sln solution
- Rebuild the solution

Adding Z-Wave Devices

1. On the Add Device Page, click on **Add ZWave**. You will see the Add Zwave Device page

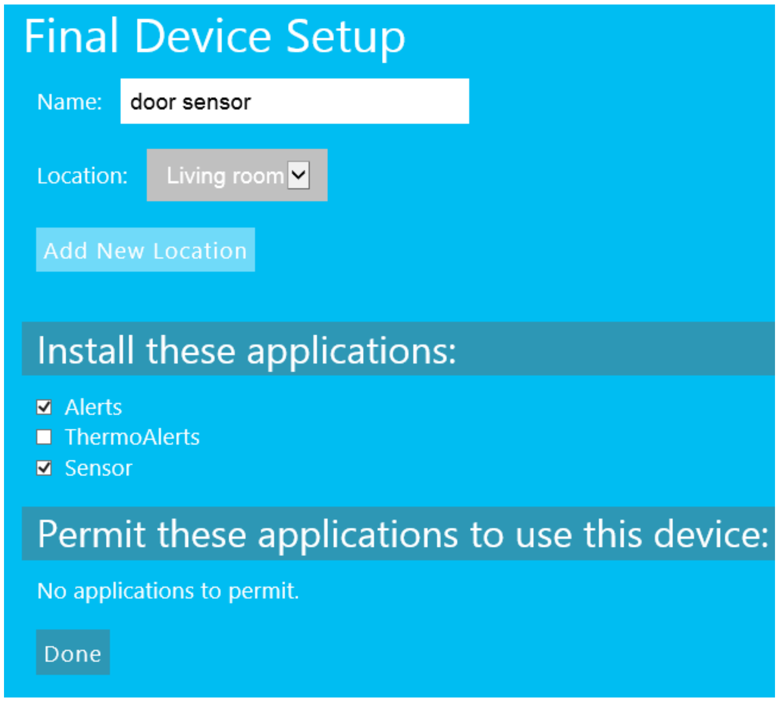

Dashboard | Add Devices | Add Zwave

Add Zwave Device

To install a z-wave sensor:
1. Select the Zwave Device Type or leave as unknown.
2. Press the Pair button
3. Within 10 seconds, press the program button on your z-wave sensor

2. You will first select the type of Zwave device you want to add. Currently we support the device types shown in the dropdown box in the screenshot.
3. Press Pair

4. Within 10 seconds, press and hold the **Program** button on the Z-Wave device. The console will confirm whether registration was successful. Refer to the device instructions for the location of the button, as well as other details such as whether a short or long press is required.
   If you receive an error (timeout, device not found), see the Z-Wave Device Won't Pair entry in the FAQ & Troubleshooting section.
5. On the **Configure your Z-Wave Device** page, enter a name and select the type and location for the device. If any applications are associated with the device you can provide permissions on this page.


Final Device Setup

Name: door sensor

Location: Living room

Add New Location

Install these applications:

- [✓] Alerts
- [ ] ThermoAlerts
- [✓] Sensor

Permit these applications to use this device:

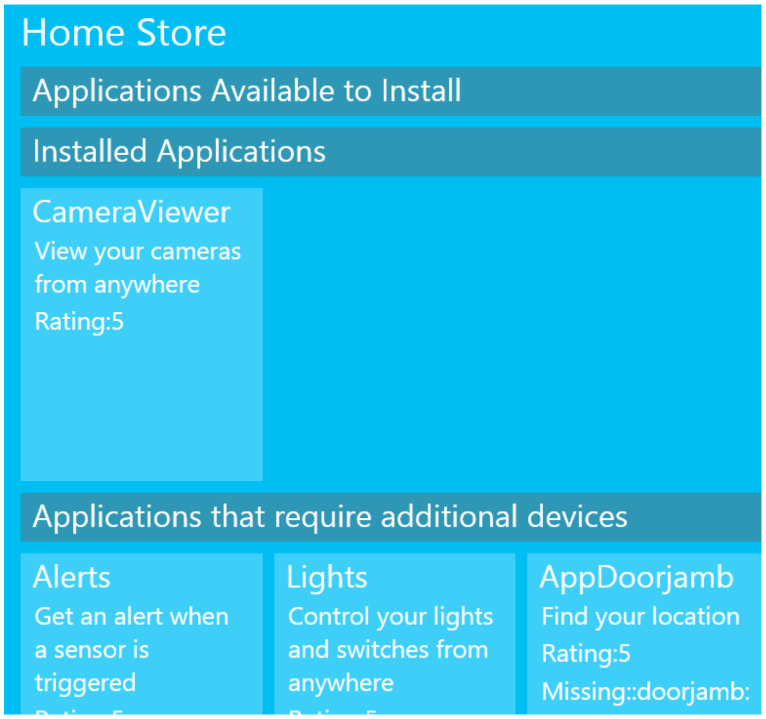
No applications to permit.

Done

6. Press **Done**. The console will confirm that the device was configured.

Home Store

The Home Store has a selection of applications that you can choose from. The list of applications and the devices needed for these applications comes from hub\output\homestore\moduledb.xml. More on this topic in the Development documentation.


Home Store

Applications Available to Install

Installed Applications

CameraViewer
View your cameras from anywhere
Rating:5

Applications that require additional devices

Alerts
Get an alert when a sensor is triggered

Lights
Control your lights and switches from anywhere

AppDoorjamb
Find your location
Rating:5
Missing::doorjamb:

Reference Hardware

The following is a list of hardware that has been tested and verified to work with Lab of Things.

Home Hub Systems

To minimize costs, we have tested an inexpensive netbook computer as a Home Hub, but similar models should work acceptably well: Acer Aspire One (tested model is AO722-0022).
Note: During initial development and testing, we strongly recommend using a more powerful laptop computer. We develop on several different laptops including the Lenovo X1 Carbon and the Dell Inspiron 15R. For more information about development systems, see Lab of Things Developer Guide.

Z-Wave Sensors

Z-Wave is a wireless communications protocol designed for home automation, specifically to remotely control applications in residential and light commercial settings.

- Z-Wave dongle (required for connecting to all Z-Wave-compatible sensors).
- Fortrezz water sensor
- Z-Wave door and window sensors

Note: Due to licensing restrictions, the Z-Wave driver modules for HomeOS are not shipped with Lab of Things. To request access to the Z-Wave modules, contact lab-of-things@microsoft.com.

Foscam

Foscam Wireless IP Cameras are designed to deliver live video and audio through the internet to a web browser, smartphone or third party recording application on the local network or anywhere in the world.

- Foscam wireless IP camera

Gadgeteer

Much more than just a sensor, Microsoft .NET Gadgeteer is an open-source toolkit for building small electronic devices using the .NET Micro Framework and Visual Studio/Visual C# Express. You can use Gadgeteer to build your own custom devices to include in your experiments. Purchase Gadgeteer components from GHI Electronics, and Seeed Studios.
If you want to use Gadgeteer-based sensors, you will also need to install a series of supporting SDKs. We provide source code for two example Gadgeteer-based devices: a "WindowCamera" which is a webcam with a screen to enable a simple custom user experience for HomeOS, and a "MoistureSensor" which senses moisture e.g. for leak detection. These are just examples; many other devices can be built. The hardware required for these examples is listed here:
WindowCamera: GHI FEZ Spider Mainboard, GHI WiFi RS21, GHI button, GHI multicolor LED, Seeed OLED display, GHI Camera, GHI USB Client SP.
MoistureSensor: GHI FEZ Spider Mainboard, GHI WiFi RS21, GHI button, GHI multicolor LED, GHI moisture sensor, GHI USB Client DP, GHI USB Serial.
We also provide 3D model files for the cases for the WindowCamera and MoistureSensor if you'd like to print your own cases. You can find the files here:

- \Gadgeteer\MoistureSensor\Case3DModel
- \Gadgeteer\WindowCamera\Case3DModel

Data Storage in the Cloud

By default, Lab of Things provides a pre-configured test storage account so you can see the config and log data being synced to the cloud.

In \Hub\output\Configs\Config\Settings.xml, you will see the entries for this default blob storage account:

```
<Param Name="DataStoreAccountName" Value="testdrive" />
<Param Name="DataStoreAccountKey"
Value="zRTT++dVryOWXJyAM7NM0TuQcu0Y23BgCQfkt7xh2f/Mm+r6c8/XtPTY0xxaF6t
PSACJiuACsjotDeNIVyXM8Q==" />
```

For your application data you can use Windows Azure Storage or Windows Azure SQL Database to write data to the cloud:

- How to work with Azure Storage: http://www.windowsazure.com/en-us/documentation/services/storage/
- How to work with Windows Azure SQL Database: http://msdn.microsoft.com/en-us/library/windowsazure/ee336279.aspx

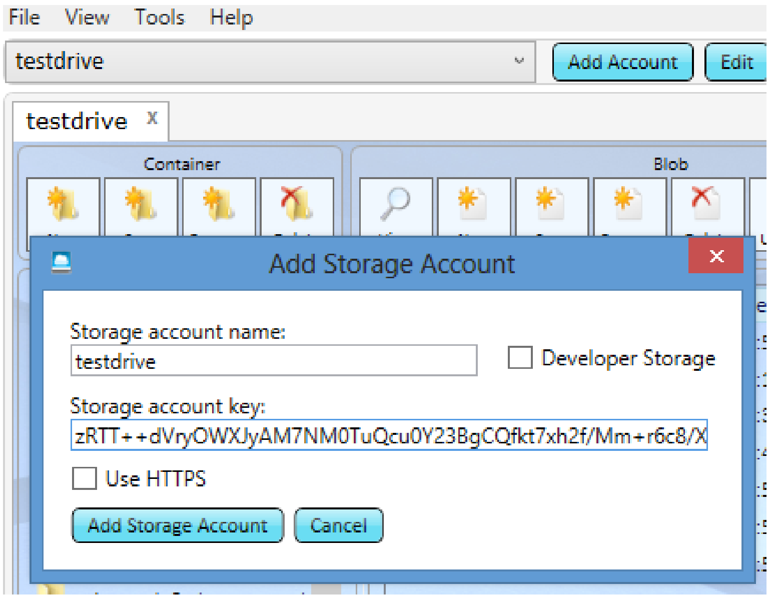
Microsoft Research has also been working on a research storage platform called Home Data Storage (HDS). HDS is designed to work with Lab of Things, but it is currently under development. A pre-alpha version of the API is included for testing purposes only, we will provide you with updates on HDS in the future.

View Contents of Cloud Data

To see the data 'synced to the cloud, download and install the Azure Storage Explorer utility.

After you download this utility, you need to enter the account name and the key to access the contents of your storage account'. Use the information for "testdrive" if you are just testing out:

- Storage account name: testdrive
- Storage account key: `zRTT++dVryOWXJyAM7NM0TuQcu0Y23BgCQfkt7xh2f/Mm+r6c8/XtPTY0xxaF6tPSACJiuACsjotDeNIVyXM8Q==`



Add Storage Account

Storage account name: testdrive

Developer Storage

Storage account key: zRTT++dVryOWXJyAM7NM0TuQcu0Y23BgCQfkt7xh2f/Mm+r6c8/X

Use HTTPS

Add Storage Account Cancel

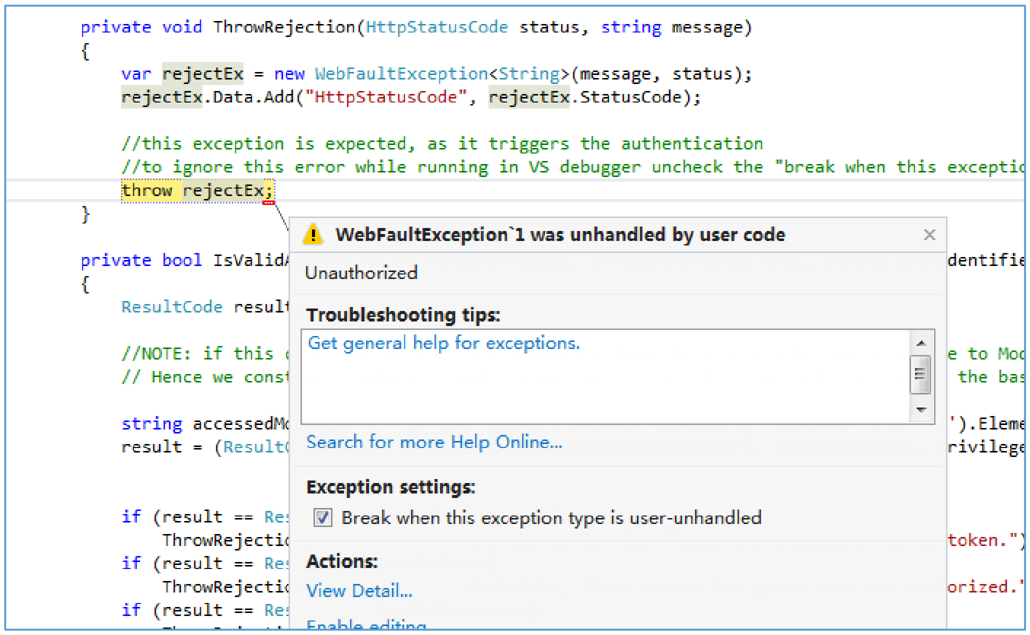
Once you provide the account name and key you will find a list of containers. Reminder: testdrive is a test account, everyone who's evaluating Lab of Things will have data written here.

FAQ & Troubleshooting

This section covers steps to take when common issues arise.

WebFaultException Thrown During Debugging

When you try to run Dashboard in Debug mode you may see the following exception:



```
        private void ThrowRejection(HttpStatusCode status, string message)
        {
            var rejectEx = new WebFaultException<String>(message, status);
            rejectEx.Data.Add("HttpStatusCode", rejectEx.StatusCode);

            //this exception is expected, as it triggers the authentication
            //to ignore this error while running in VS debugger uncheck the "break when this exceptic
            throw rejectEx;
        }

        private bool IsValidA                                                          dentifie
        {
            ResultCode resul

            //NOTE: if this                                                          e to Mod
            // Hence we const                                                         the bas

            string accessedMo                                                        ').Eleme
            result = (Result                                                         rivilege


            if (result == Re
                ThrowRejecti                                                         token.")
            if (result == Re
                ThrowRejecti                                                         orized.'
            if (result == Re
```

WebFaultException`1 was unhandled by user code

Unauthorized

Troubleshooting tips:

Get general help for exceptions.

Search for more Help Online...

Exception settings:

[✓] Break when this exception type is user-unhandled

Actions:

View Detail...

Enable editing

This exception is expected when running under the debugger. Throwing this exception is part of the authentication logic. Is OK to hit continue and if you don't want to see this exception repeatedly. You should uncheck that "Break when …" box.

Z-Wave Device Won't Pair

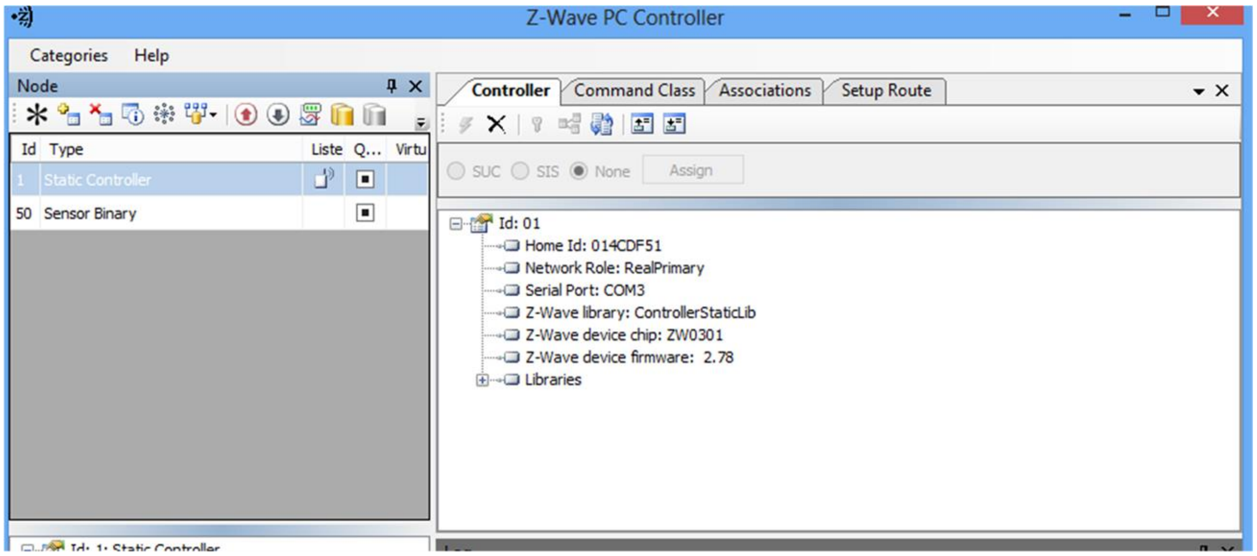
When attempting to add a Z-Wave device, the process times out or the error "Device not found" is displayed. Sometimes when a device is removed from the platform it can still remain registered on the system. Try these things to remove the Z-Wave device so that you can add it again.

Run removezwavenode

1. With the platform running, in the dashboard console enter the `removezwavenode` command.
2. Immediately press and hold the **Program** button on the Z-Wave device.
3. Confirmation will be displayed in the console.

Perform a hard reset

1. Stop the platform.
2. Navigate to `Hub\output\binaries\Pipeline\AddIns\HomeOS.Hub.Drivers.ZwaveZensys_4_55`.
3. Run ZWaveController.exe.
4. Under Node, click the icon with the red "x", then click the **Program** button on the device to unpair.



Id	Type	Liste	Q...	Virtu
1	Static Controller		■	
50	Sensor Binary		■	

```
Id: 01
    Home Id: 014CDF51
    Network Role: RealPrimary
    Serial Port: COM3
    Z-Wave library: ControllerStaticLib
    Z-Wave device chip: ZW0301
    Z-Wave device firmware:  2.78
    Libraries
```

Id: 1: Static Controller

Remove and reconnect the Z-Wave controller

- If a Z-Wave device is not being recognized, unplug the Z-Wave controller (dongle) and plug it back in.

Why isn't my driver running?

There could be a number of reasons, but be sure to check the following first as diagnostic steps.

Make sure that your scout for the driver is running – each device driver requires its scout to be running so that HomeOS platform can discover a new device (more on this in the Development documentation).

The only exception is the ZWave drive we provide, it doesn’t utilize scout for discovery.

- For your own driver, check that the scout component is running. Verify that Hub\output\binaries\Scouts\HomeOS.Hub.Scouts.YourScoutName exists
- Make sure that Scouts.xml (\Hub\output\Configs\Config) has an entry of the scout you want to run. For example, by default we ship with this entry in the Scouts.xml, so the web cam scout is always running:
  <Scout Name="HomeOS.Hub.Scouts.WebCam" DllName="HomeOS.Hub.Scouts.WebCam.dll"/>

The Platform is in a Weird State

If the platform becomes non-responsive or exhibits other unwanted behavior, you can reset the platform. Take these steps:

- Reset the platform by running reset command or delete output directory and then re-build the solution
- Clear your browser cache

Are there logs, and where are they?

By default in startplatform.bat, the platform is started without the **-l** switch:

```
binaries\Platform\HomeOS.Hub.Platform.exe -c %configDir%
```

This means that the log file will go to the default location: \Hub\output\Data\Platform\homeos.log
You can change it by changing the LogFile entry in settings.xml. You would have to restart the platform for this to take effect.

Known Issues Related to Cloud Data Storage

Dependencies

For remote storage of log and configuration, HomeOS uses Microsoft Sync Framework Azure technology to write this data to the cloud. So it is important that you have the Microsoft Sync Framework 2.1 Redistributable Packages (x86) installed. Otherwise you may encounter exceptions complaining about lack of StorageClient object or not finding Sync components.

Invalid HomeID causing invalid Container name

Unhandled Exception: Microsoft.WindowsAzure.StorageClient.StorageClientException: One of the request inputs is out of index.. System.Net.WebException: The remote server returned an error: (400) Bad Request.

You may see the above exception when a blob container name violates the following naming rules:

- Blob container names must start with a letter or number, and can contain only letters, numbers, and the dash (-) character.
- Every dash (-) character must be immediately preceded and followed by a letter or number

- All letters in a container name must be lowercase.
- Blob container names must be from 3 through 63 characters long.

When HomeOS logger is trying to create a container in the blob storage, but it's not happy with the container name. And the container name for the log would be "log-" + HomeId.ToLower().
If your HomeID (in settings.xml) has character(s) that might break the rules above the exception will throw.