

Getting Started with Lab of Things

Contents

Overview	2
About Lab of Things and HomeOS	2
Glossary.....	2
Prerequisites for Home Hub	3
Reference Hardware	3
Home Hub Systems	3
Sensors.....	3
Download Source and Binaries	4
To Git Clone the Source Code	4
Lab of Things Workflows.....	4
Set up a Home Hub for Evaluation.....	5
Pre-Setup	5
Set up a Hub using the MSI	5
Set up a Home Hub Manually for Development & Testing.....	5
Run the Platform.....	6
Run the Platform from the command prompt	6
Stop the Platform from the command prompt	6
Reset the Platform	6
Update the Platform	7
Batch Files	7
Running Watchdog.....	7
Running the Dashboard for the First Time.....	8
Manage Settings	10
Home Store	11
Install and Configure Hardware	12
Add or Remove a Device	13
Add a Z-Wave Device	13
Cloud Storage for Your Data.....	14
Prerequisite	14
Connect to Azure Blob Storage	14

View Contents of Cloud Data.....	15
Console Commands	16
Open an Elevated Command Prompt	17
Management Portal	17
Troubleshooting.....	17
Z-Wave Device Won't Pair	18
The Platform is in a Weird State	18

Overview

This guide covers everything you need to know to get up and running with Lab of Things and HomeOS. In particular you will learn how to set up a Home Hub to try out Lab of Things. For details about setting up a development environment and developing your own Lab of Things applications, see [Lab of Things Developer Guide](#).

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. 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).

Lab of Things uses the HomeOS software system. HomeOS provides a PC-like abstraction for in-home hardware and simplifies the tasks of writing applications and managing sensors. 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.

Glossary

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

HomeOS: A software platform that provides a PC-like abstraction for in-home hardware and simplifies the tasks of writing applications and managing sensors. Each household participating in Lab of Things runs HomeOS on a dedicated computer.

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

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

Lab of Things study: A research application (e.g. energy monitoring, occupancy sensing, adaptive heating) running in multiple homes.

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

Watchdog: A Windows service running on a Home Hub that monitors the software platform and ensures that it is always running. For example, if the Home Hub gets rebooted, watchdog will kick in and restart the platform automatically.

Prerequisites for Home Hub

- [Windows 8](#) or Windows 7 with a wireless card
- Microsoft .NET Framework 4.5 installed
- [Silicon Labs USB to UART driver](#) (needed for Z-Wave dongles).

Note: for development purposes you can use a wired computer for your Home Hub. For more information, see [Lab of Things Developer Guide](#).

Reference Hardware

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

Home Hub Systems

Each household participating in Lab of Things needs to have a Home Hub. 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](#).

Sensors

This section lists the sensors we have tested with.

Z-Wave

[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 issues, the Z-Wave modules do not ship 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`

Download Source and Binaries

Both source code and binaries are available to download at <https://labofthings.codeplex.com/>. You can copy the binaries to a Home Hub to run immediately. If you want to make modifications or extend the platform you can also download the full source code. For more information about development, see the [Lab of Things Developer Guide](#).

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](#).

Lab of Things Workflows

This guide covers two specific workflows:

- Set up a Home Hub for Evaluation
- Deploy a Lab of Things study

The procedures in this guide can be applied to more than one workflow.

Set up a Home Hub for Evaluation

This section explains how to set up a Home Hub for evaluation.

Pre-Setup

Before deploying to a hub, you first need to enter your organization ID (OrgID) in `Hub\output\configs\settings.xml`. To obtain an OrgID, email a request to Lab-of-things@microsoft.com. The OrgID is used to associate your Home Hubs with your organization.

Set up a Hub using the MSI

To try out Lab of Things without compiling the source code you can run the .msi installer <http://www.lab-of-things.com/LabOfThings.msi>. It will install the output binaries and the watchdog service. It also makes the appropriate configuration settings to the hub. Note: Install [Silicon Labs USB to UART driver](#) (only applicable if you are using Z-Wave sensors).

- Once you've installed the MSI to access the dashboard, go to <http://localhost:51430/guiweb/>.
- See Dashboard UI section in this document to set up the hub and configure devices.
- If your hub is connected to the internet, go to the Management Portal to see <http://homelab.cloudapp.net/>

Set up a Home Hub Manually for Development & Testing

These steps represent the modifications that the .msi makes on the computer, and are presented here for reference purposes.

1. Enable [Remote Access](#).
2. Open port 51430 in Windows Firewall. For more information, see [Open a Port in Windows Firewall](#).
3. Turn off Windows Update.
4. Copy `Hub\output` files to the hub computer.
5. Install and run Watchdog.
6. From `homeos2\Hub\output\binaries\platform\run`
`HomeOS.hub.platform.exe -c ..\..\Configs\Config.`

(Use `..\..\Configs\DummyConfig` for testing and evaluation.)

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).



Run the Platform

During development you can use the console to perform tasks such as starting, stopping, and resetting the platform.

Run the Platform from the command prompt

Demo mode will enable you to run the Home Hub environment without starting the watchdog service.

1. [Open an elevated command prompt.](#)
2. Navigate to `homeos2\Hub\output\`.
3. Run `startplatform.bat`. The console will display messages for the various services, as well as errors (if there are any). When the platform is ready, you will see the message `Waiting for commands`.
4. To exit out of demo mode, type `exit`.
5. To get help for commands, type `help`.

Stop the Platform from the command prompt

- At the console, type `exit`.

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 used]
```

Update the Platform

From time to time you may want to push updates out to your Home Hubs. You can use the Platform Packager tool to prepare these updates. PlatformPackager generates output files that can be copied to a remote location and then used to update all of your hubs.

To run PlatformPackager

1. [Open an elevated command prompt](#)
2. Navigate to `homeos2\Hub\Tools\PlatformPackager\bin\Debug\`.
3. Run `PlatformPackager.exe`. The resulting files will be output to a `tmp` folder in `Debug`.
4. Open the `tmp` folder and copy the `.config` and `.zip` files to a web-accessible location (must be accessible using `http`).
5. On each hub, edit `watchdog.txt`, pasting the URL for the web location in the previous step. Your updates will be automatically applied.

Batch Files

Use the batch files located 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.
- `starthostednetwork.bat`: Run this to set up Gadgeteer devices.
- `stophostednetwork.bat`: Run this to stop the hosted network.

Running Watchdog

Starting platform in command prompt is useful during development and testing. However, during deployment, you would want to enable watchdog to make sure that platform is always running.

To install Watchdog

1. Build the Hub project.
2. [Open an elevated command prompt](#)
3. `C:\Windows\Microsoft.NET\Framework\v4.0.30319\installutil.exe [Root]homeos2\Hub\output\Watchdog\HomeOS.Hub.Watchdog.exe`
4. Start the watchdog service using the following command:

```
net start "HomeOS Hub Watchdog"
```

To stop WatchDog

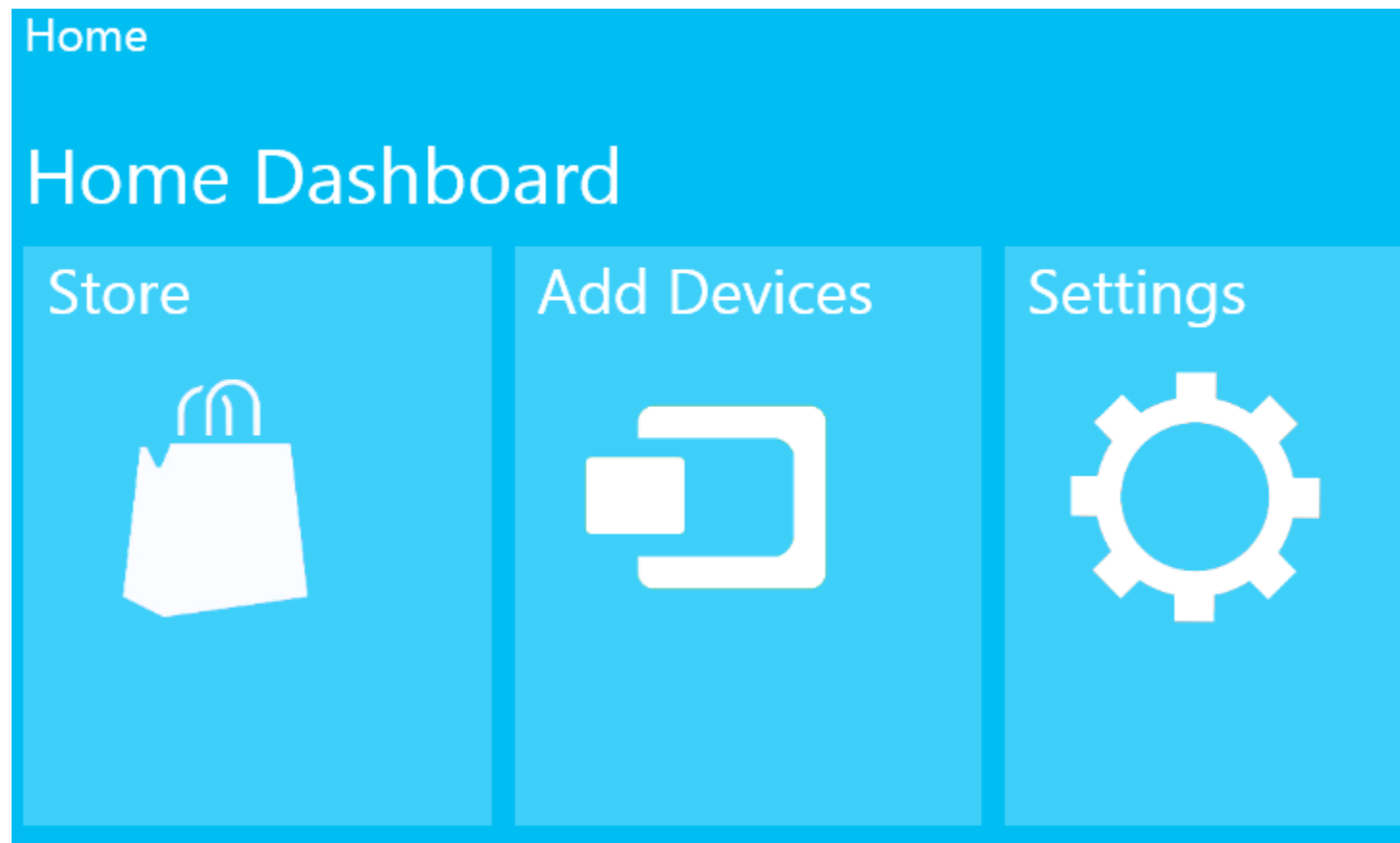
- Go to Service in either MMC or Task Manager.
- Highlight HomeOS Hub Watch Dog service, and then right-click and select Stop.Dashboard User Interface.

Note: If you have previously started the platform manually or by using `startplatform.bat`, the log file from that configuration will be used when watchdog is started (instead of the default `output\Data\Platform\homeos.log`). To restore the default log file, take one of the following steps:

- [Reset the platform](#)
- Add a new log file entry in Settings.xml:

```
<Param Name="???" Value="output\Data\Platform\homeos.log"/>
```

The Dashboard is the user interface for the Home Hub. In the Dashboard you can installed applications, install new applications, manage settings, and install and configure the sensors that will be connected to the Home Hub. To access the dashboard, navigate to <http://localhost:51430/guiweb/>.



Running the Dashboard for the First Time

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

Network Security Key

Join

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:

Next

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

HomeOS Setup

Your HomeOS Hub has been successfully configured.

Home ID: willsplace

Home Password:berlinerstrasse

Remote Access:

<http://homelab.cloudapp.net:51431/willsplace/GuiWeb/in>

Default Email: w@123.org

Next

4. Click **Next** to finish.

Manage Settings

In settings, you can add/remove users that can remotely access your Home Hub, and see which applications and devices are installed. Your remote access URL is listed here as well.

Settings

Home ID:

URL for remote access: <http://homelab.cloudapp.net:51431//GuiWeb/index.html>

Email:

[Edit](#)

Users

No users configured for remote access.

[Add Users](#)

Installed Applications

No applications currently installed

Installed Devices

No devices currently installed

[Home Store](#)

The Home Store has a selection of applications that you can choose from.

[Dashboard](#) | [Store](#)

Home Store

Applications Available to Install

Installed Applications

Applications that require additional devices

Alerts

Get an alert when a sensor is triggered

Rating:5

Missing:(sensor)

AlertsTS

Typescript version of Alerts

Rating:5

Missing:(sensor)

Sm

View
cam

Rat

Mis

AppDummy by

HomeTemperature

Mo

Install and Configure Hardware

Currently four types of hardware devices are supported:

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

Add or Remove a Device

Follow this procedure to add new sensors and other types of hardware devices to the Home Hub.

To add a new device

1. Ensure that the platform is running (open the command window and type `help`, you should see a list of commands).
2. Launch the dashboard (<http://localhost:51430/guiweb/>).
3. Click **Add Devices**. A list of unconfigured devices will be displayed.
4. Under **Click on device name to configure**, choose the device you want to add from the list of devices.
5. Enter a name for the device and select a location.
6. Check the appropriate checkboxes to install software that is associated with the device, and/or to provide permission for other applications to access the device.
7. Click **Done** to finish adding the device.

To remove a device

1. Ensure that the platform is running (open the command window and type `help`, you should see a list of commands).
2. Launch the dashboard (<http://localhost:51430/guiweb/>).
3. Click **Settings**, then scroll down to **Installed Devices**.
4. Locate the device in the list, then click **Remove**.

Add a Z-Wave Device

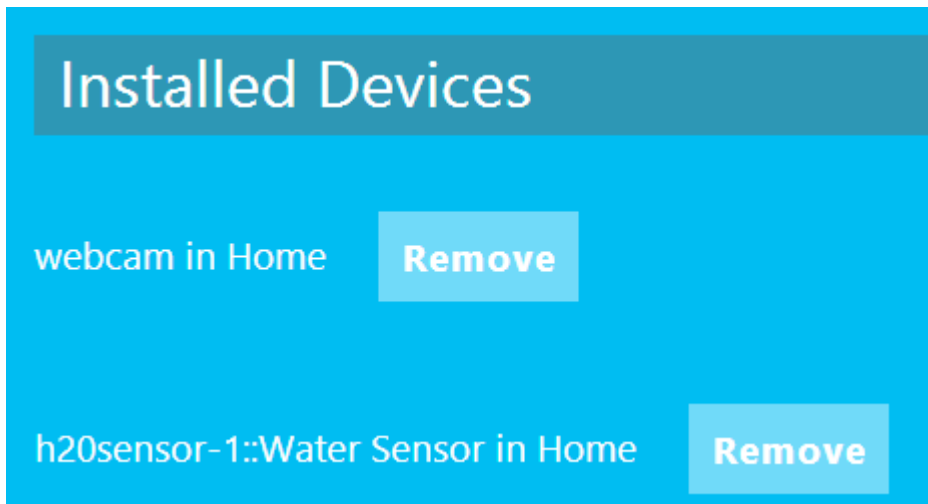
By default the Z-Wave module is not included in the solution. Please contact lab-of-things@microsoft.com for more information on where to obtain the module.

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.

To add a Z-Wave device

1. Ensure that the platform is running (open the command window and type `help`, you should see a list of commands).
2. Launch the dashboard (<http://localhost:51430/guiweb/>).
3. Click **Add Devices**. A list of unconfigured devices will be displayed.
4. Click **Add ZWave**.
5. In the dashboard, press the **Pair** button.
6. 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 [Z-Wave Device Won't Pair](#).
7. 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.

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



Cloud Storage for Your Data

By default, Lab of Things provides a pre-configured test storage account so you can see 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==" />
```

Any time you run the platform under Config, data from your hub or development computer will automatically be pushed to the "testdrive" Azure storage account.

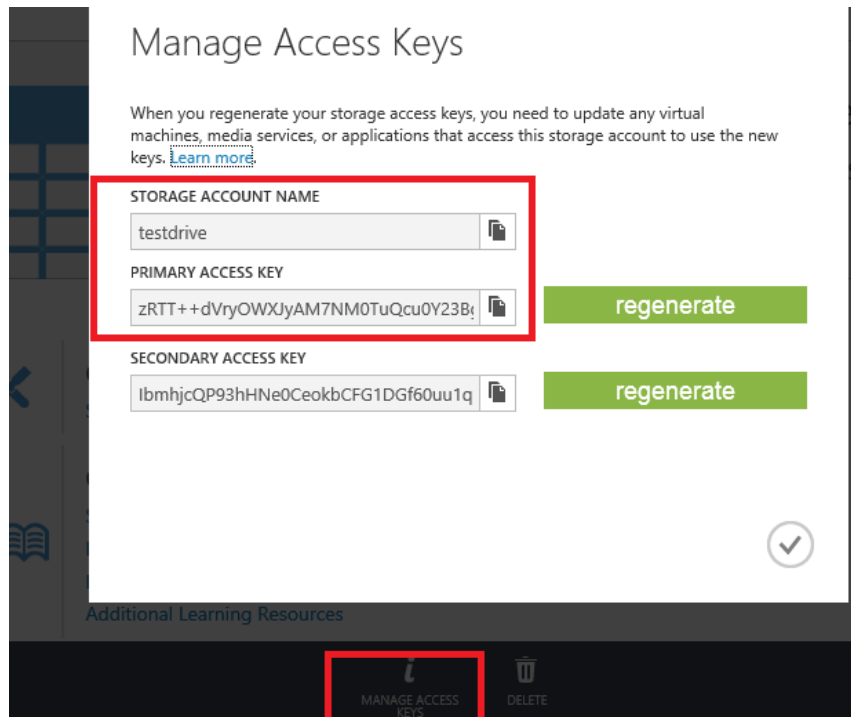
Prerequisite

- Install the latest Azure SDK (go to <http://www.windowsazure.com/en-us/downloads/>, choose the .NET SDK for VS2012).
- For development: Install Microsoft Sync Framework 2.1 SDK: <http://www.microsoft.com/en-us/download/details.aspx?id=23217> (**Select the x86 version**)
- On hubs you want to deploy to: Install Microsoft Sync Framework 2.1 Redistributable Package: Install the Microsoft Sync Framework 2.1 Redistributable Package: <http://www.microsoft.com/en-us/download/details.aspx?id=19502>
 - Download and install both **ProviderServices-v2.1-x86-ENU.msi** and **Synchronization-v2.1-x86-ENU.msi**. **Select the x86 version for both.**

Connect to Azure Blob Storage

A Lab of Things application can sync data to the cloud by using the Storage System API, as long as the Storage System stream is created as a remoteable stream. When a stream gets 'synced to the cloud, it's written to an Azure blob storage account. Here are the steps to enable your application to sync data to the cloud

1. Get an Azure subscription (more about what a subscription is and how it works [here](#)).
2. Once you have a subscription, create a [Storage Account](#).
3. From the storage account, you need two important pieces of information:
 - Blob account name
 - One of the blob account keys – you can find the key info by clicking on the MANAGE ACCESS KEYS button on the bar at the bottom of your Storage account screen.



4. Paste the Account Name and the Account Key of the blob account in the Settings.xml in the configuration folder you want to run (e.g. DemoConfig for testing, Config for deployment)
The entries in Settings.xml should look something like this:

```
<Param Name="DataStoreAccountName" Value="MyAccountName" />
<Param Name="DataStoreAccountKey" Value="YourKeyString" />
```

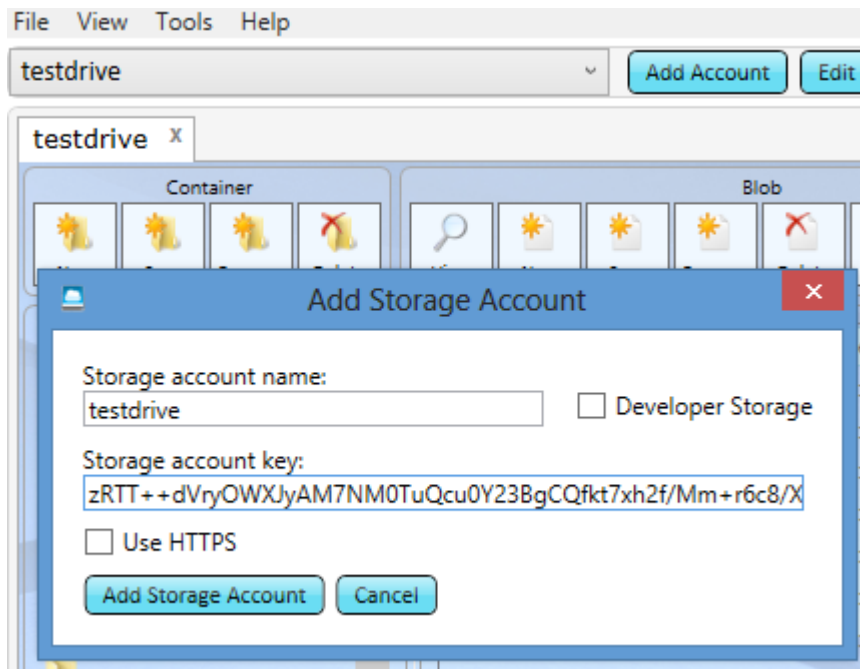
View Contents of Cloud Data

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 the DemoConfig:

- Storage account name: testdrive

- Storage account key:
zRTT++dVryOWXJyAM7NM0TuQcu0Y23BgCQfkt7xh2f/Mm+r6c8/XtPTY0xxaF6tPS
ACJiuACs jotDeNIVyXM8Q==



Once you provide the account name and key, you will see the containers whatever the app you are running that is enable for syncing to the cloud.

The data synced to the cloud is serialized by the Storage subsystem in .dat format. For binary files such as images, right now you can't just change the extension to view them, because the serialization process has added some additional information to the file. We will be releasing an API soon to allow you to deserialize and view data stored in the cloud.

Console Commands

When the platform is running, you can run the following commands by typing them at the console prompt within the console or PowerShell window that you used to launch the platform.

Command	Description
show alladdins	To show all AddIns that are available, loaded, and ready to start as modules. AddInToken names are same as binary names.
startmodule {friendlyName} {binaryName}	Command to start a module from binary called binaryName with a friendly name assigned as friendlyName

show modules	Command to display currently running modules and their secrets.
show modulesstates	Command to display currently what state running modules are in.
show resourceusage	Command to display CPU and memory consumption of running modules.
show ports	Command to display ports exported by currently running modules.
show wifinets	Command to display currently available wifi networks.
stopmodule {secret}	Command to stop module with given secret.
Exit	Command to quit all modules and HomeOS platform.

Run `homeos.hub.platform.exe -?` to get a list of switches.

Switch	Description
-?	Display the help message.
-l	Set log file name ('-' for stdout).
-c	Set the configuration directory to use.
-r	Running Mode.
-p false	Turn off policy enforcement.

Open an Elevated Command Prompt

1. Open Start by swiping in from the right edge of the screen (or if you're using a mouse, pointing to the upper-right corner of the screen and moving the mouse pointer down), and then tapping or clicking **Start**.
2. Type `cmd`, then right-click or press and hold on **Command Prompt**.
3. At the bottom of the screen, click **Run as administrator**.

Or

- If you're using a keyboard with Windows 8, you can open an elevated command prompt from the Power User Menu. Just press Windows + X and then click on **Command Prompt (Admin)**. Click **Yes** in the User Account Control message that appears.

Management Portal

Once your hub is running, you can see the hub status and heartbeat on this web location:

Troubleshooting

This section covers steps to take when common issues arise.

Z-Wave Device Won't Pair

Z-Wave Device Won't Pair

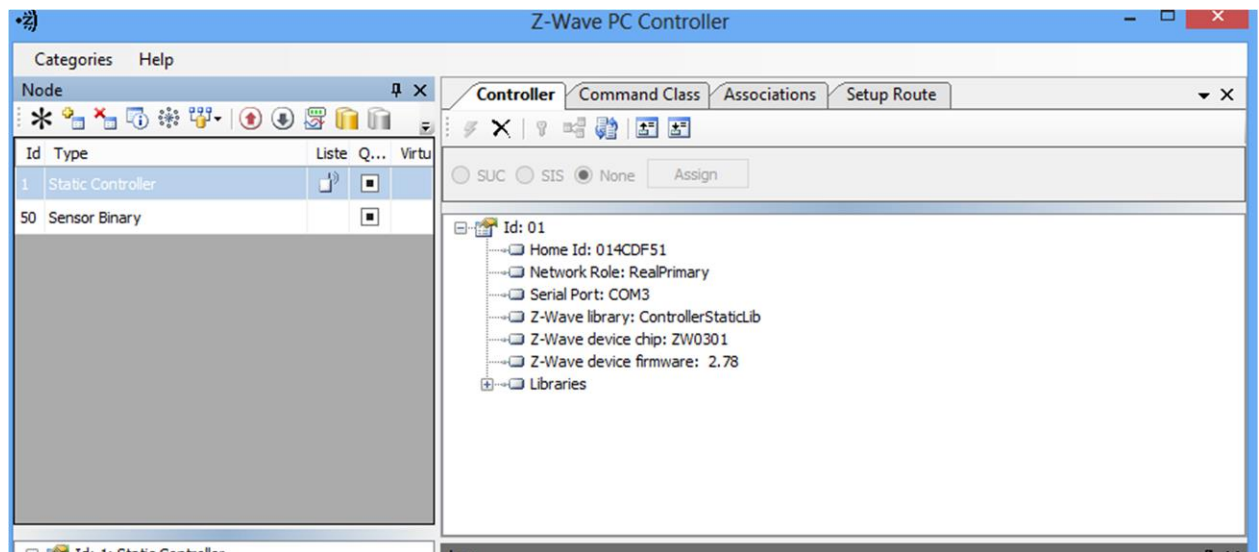
When attempting to add a Z-Wave device, the process times out or the error "Device not found" is displayed. Sometimes when a device is removed from the platform it can still remain registered on the system. Try these things to remove the Z-Wave device so that you can add it again.

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.



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.

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](#)

- Clear your browser cache