



Desktop MQTT Client for Adafruit.io

Created by lady ada

MQTT.FX

Welcome to the home of MQTT.fx

A JavaFX based MQTT Client.

MQTT.fx is a MQTT Client written in Java based on [Eclipse Paho](#).

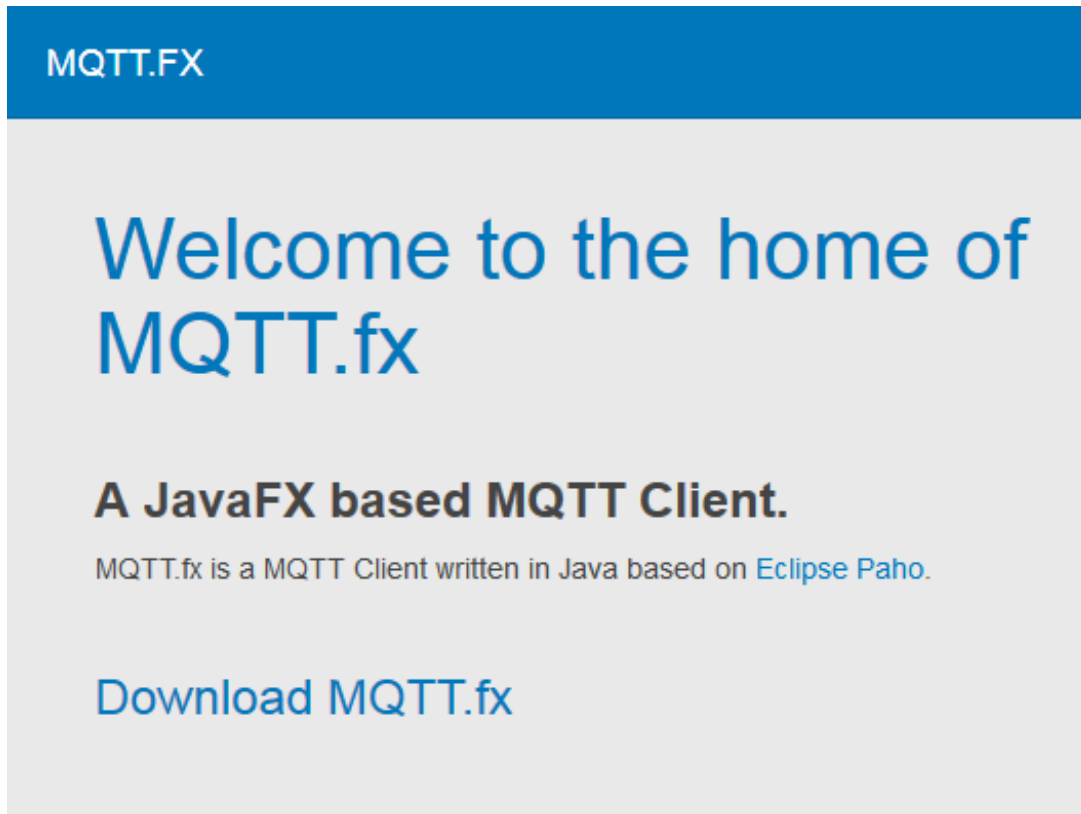
[Download MQTT.fx](#)

Last updated on 2017-07-14 05:58:50 AM UTC

Guide Contents

Guide Contents	2
Overview	3
Installing Software	4
Configuring	7
SSL Connections	9
Connecting & Use	11
Connecting	11
Publishing	11
Subscribing	12

Overview



The screenshot shows the MQTT.fx website. At the top, there is a blue header with the text "MQTT.FX" in white. Below the header, the main content area has a light gray background. It features the heading "Welcome to the home of MQTT.fx" in a large blue font. Underneath, it says "A JavaFX based MQTT Client." in bold black text, followed by a smaller line of text: "MQTT.fx is a MQTT Client written in Java based on [Eclipse Paho](#)." At the bottom of the main content area, there is a blue link that says "Download MQTT.fx".

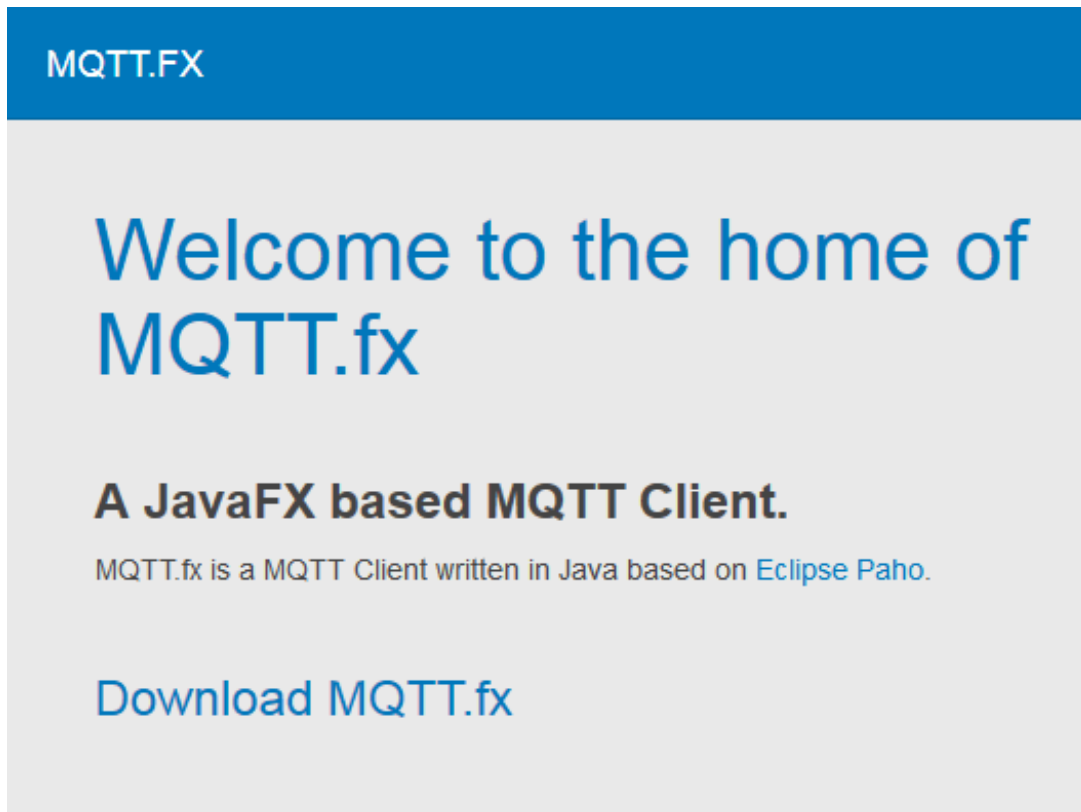
Even though adafruit.io (<http://adafruit.io/jcY>) is designed to be used with portable and wireless devices, it can also be used with desktop software! This makes it easy to control remote devices and debug your setup.

This tutorial will document how we installed some desktop software and used it to interact with our dashboard & feeds.

Installing Software

We tried out and really like MQTT.fx which is Open source under the Apache 2.0 license and has packages available for Windows, Mac and Linux (64bit deb)

Start by visiting <http://mqttfx.org/> (<http://adafru.it/xDY>)



The screenshot shows the MQTT.FX website. At the top, there is a blue header with the text "MQTT.FX" in white. Below the header, the main content area has a light gray background. The text "Welcome to the home of MQTT.fx" is displayed in a large, blue, sans-serif font. Below this, the text "A JavaFX based MQTT Client." is shown in a bold, black, sans-serif font. Underneath, a smaller line of text reads "MQTT.fx is a MQTT Client written in Java based on Eclipse Paho." with "Eclipse Paho." in blue. At the bottom of the main content area, there is a blue link that says "Download MQTT.fx".

and download the latest version, as of this writing, 0.0.16 is the latest

MQTT.FX HOME DOWNLOAD SCREEN

Download







Latest Release

[MQTT.fx Version 0.0.16](#) (more information)

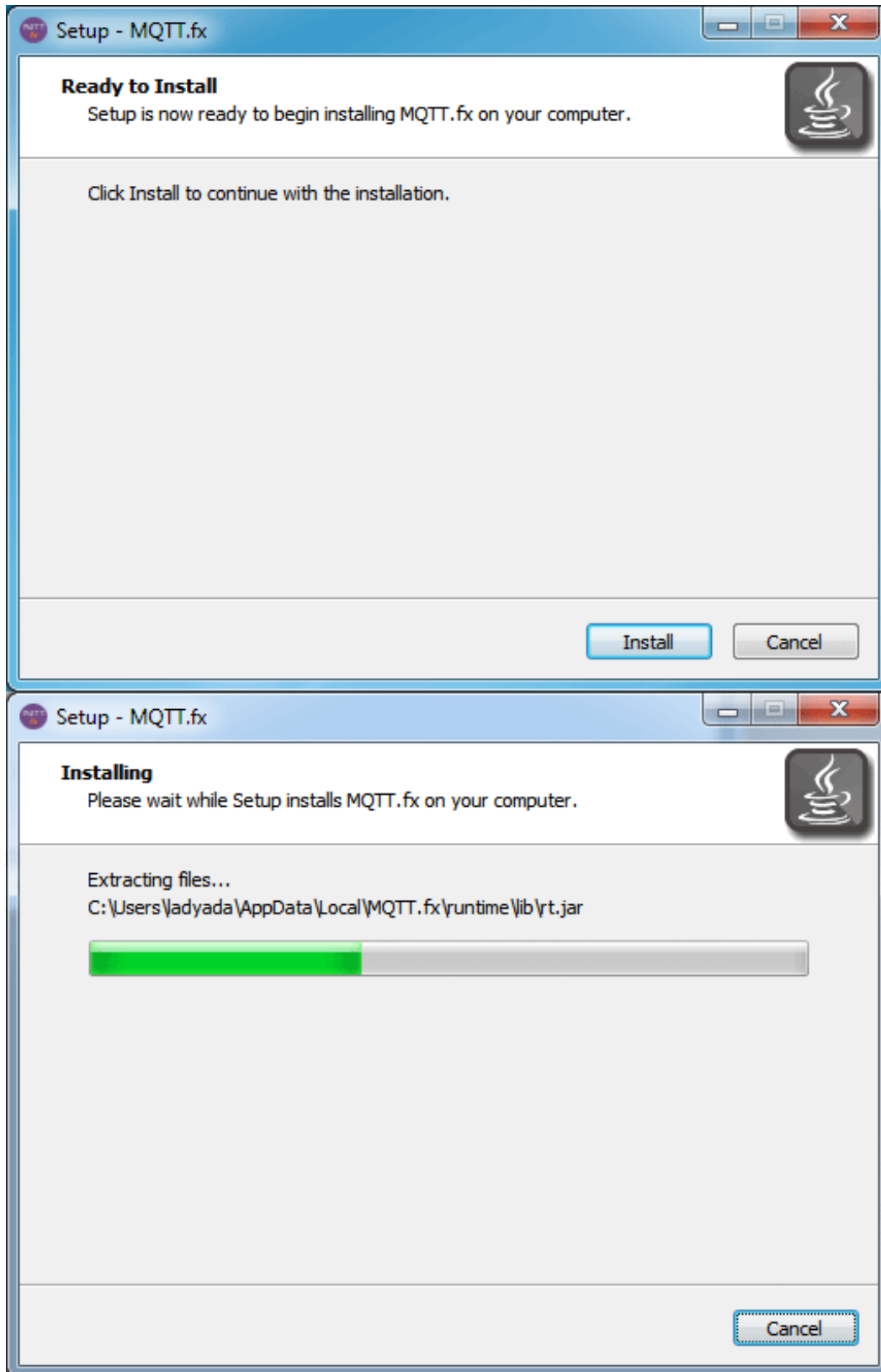
MQTT.fx 0.0.16

You might have to mouseover to see the links, we grabbed the .exe file for windows. The .dpkg file is for Macs, the .deb file is for Debian Linux.

Index of /apps/mqttfx/0.0.16

	Name	Last modified	Size	Description
	Parent Directory		-	
	MQTT.fx-0.0.16-64bit..>	29-Jun-2015 23:53	51M	
	MQTT.fx-0.0.16-64bit..>	29-Jun-2015 23:54	69M	
	MQTT.fx-0.0.16.dmg	29-Jun-2015 22:46	76M	
	MQTT.fx-0.0.16.pkg	29-Jun-2015 22:46	69M	
	mqtt.fx-0.0.16-64bit..>	01-Jul-2015 21:12	55M	

Go thru the installation process

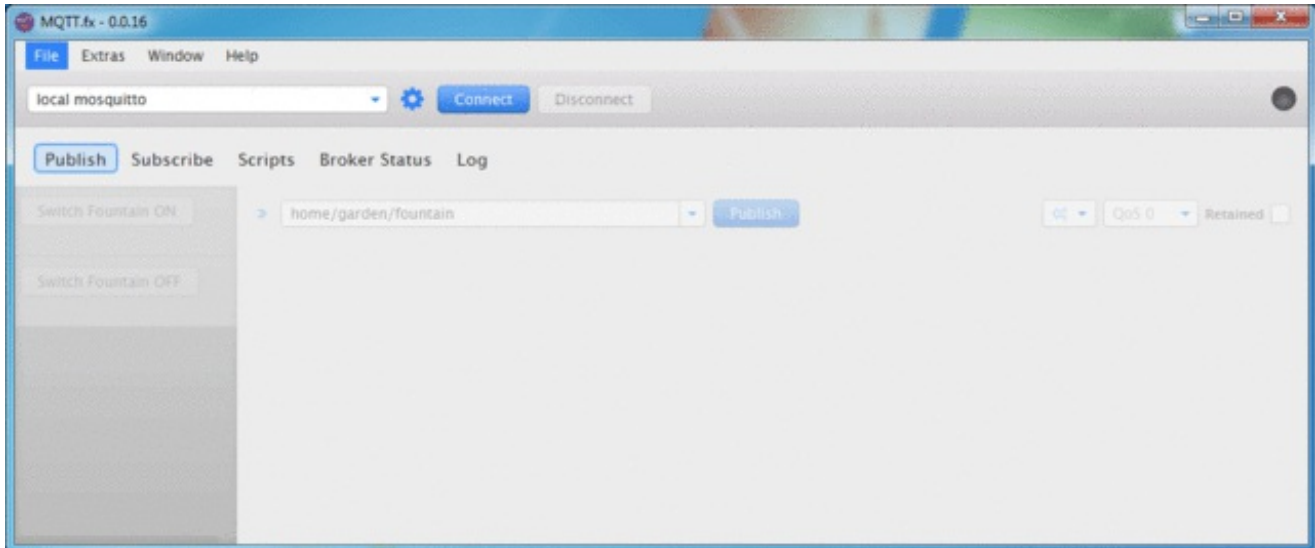


OK now run it and go to the next step!

Configuring

Once the software is installed, you'll need to configure it for adafruit.io

Start it up, this is the main window:



Click on the gear logo in the top middle area. It will take you to the config page.

1. The **Profile Name** is just the name we'll call this connection set by. I used adafruit.io but use whatever you like
2. The **Broker Address** must be io.adafruit.com - we don't recommend using an IP address in case the server moves
3. For the **Broker Port** use 1883 if you are not using SSL. For SSL connections, go with 8883 instead!
4. **Client ID** needs to be a unique value - you can use your AIO key or go to <http://www.danstools.com/md5-hash-generator/> (<http://adafru.it/fJt>) and type in a bunch of text to generate a unique MD5 hash

Then there's a set of sub-tabs. For the General Tab. You can set whatever timeout/keep alive you want. We recommend keeping Clean Session and Default MQTT version selected.

Connection Profile

Profile Name

Broker Address

Broker Port

Client ID

General

User Credentials

SSL/TLS

Proxy

Last Will and Testament

Connection Timeout

Keep Alive Interval

Clean Session

MQTT Version Use Default

Clear Publish History

Clear Subscription History

In the next tab you actually put in your credentials. Use your adafruit.com account user name for **User Name** and then your Password is the long alphanumeric key in your AIO settings

YOUR AIO KEY



Your Adafruit IO key should be kept in a safe place and treated with the same care as your Adafruit username and password. People who have access to your AIO key can view all of your data, create new feeds for your account, and manipulate your active feeds.

If you need to regenerate a new AIO key, all of your existing programs and scripts will need to be manually changed to the new key.

YOUR AIO KEY

afojwfeoiawjeofijaw90834294823oli4h234as98r2

REGENERATE AIO KEY

Connection Profile

Profile Name

Broker Address

Broker Port

Client ID

General

User Credentials

SSL/TLS

Proxy

Last Will and Testament

User Name

Password

SSL Connections

If you want to use SSL (and you should!) the next tab will let you set that up.

Connection Profile

Profile Name

Broker Address

Broker Port

Client ID

General User Credentials **SSL/TLS** Proxy Last Will and Testament

Enable SSL/TLS Protocol

CA signed server certificate
 Self signed server certificate
 Self signed certificates
 Self signed keystores

Select **CA signed server certificate** and for Protocol, try **TLSv1.2**

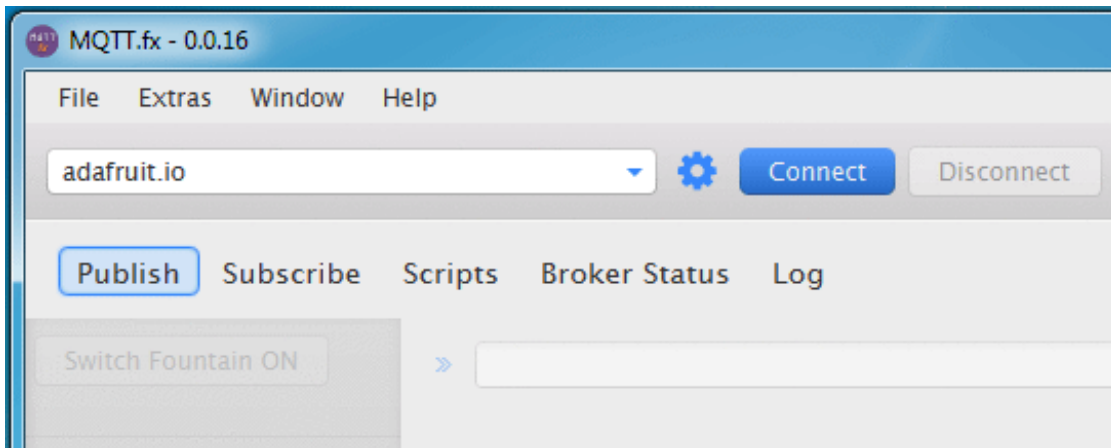
You'll also need to make sure the **Broker Port** above is **8883**!

Connecting & Use

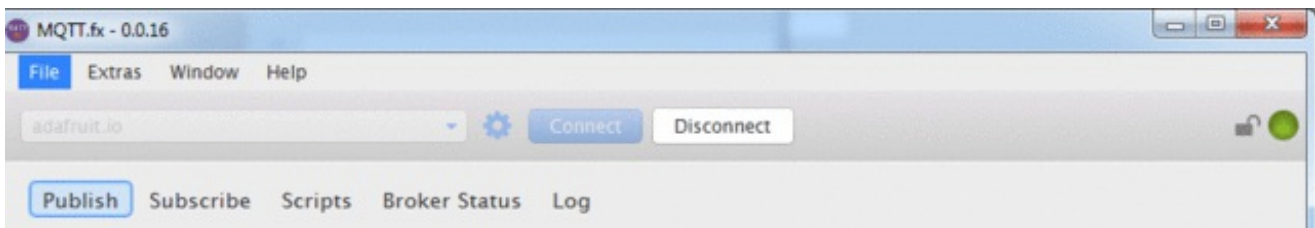
Connecting

Once done with the profile, click **OK** or **Apply** and **Cancel** to get back to the main screen.

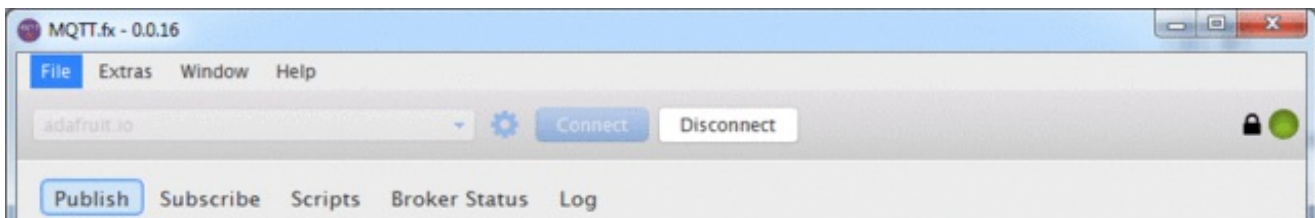
Select the new profile by name and click **Connect**



The **Connect** button will gray out and the **Disconnect** button will be clickable. On the very right hand side you'll see a green dot to tell you the connection is OK



If you are using SSL, you'll see a closed-lock as well

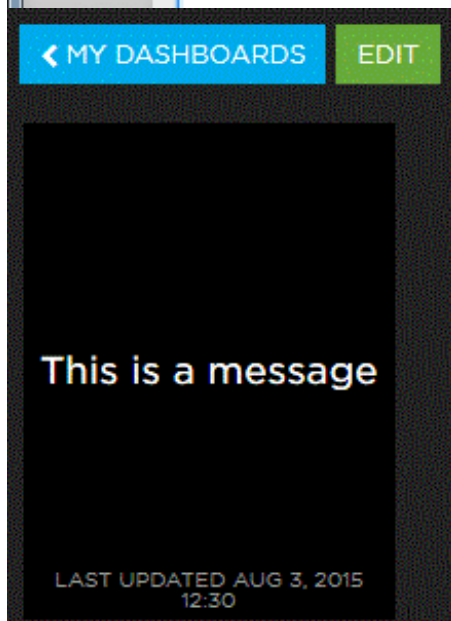
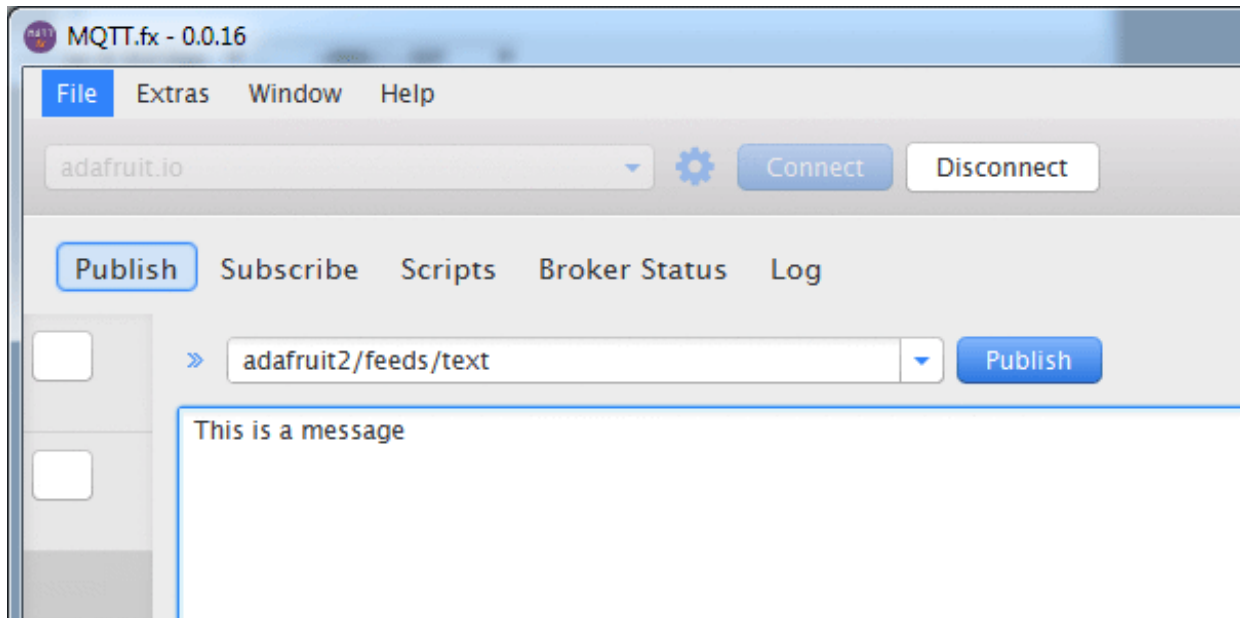


Publishing

You can publish to a feed by clicking on the **Publish** button in the 'tab menu'

Next to the >> there's a text field. Enter your feed name there, it will be in the format *username/feeds/feedname* where *username* is your adafruit account name, and *feedname* is the actual feed name you have in your feedlist

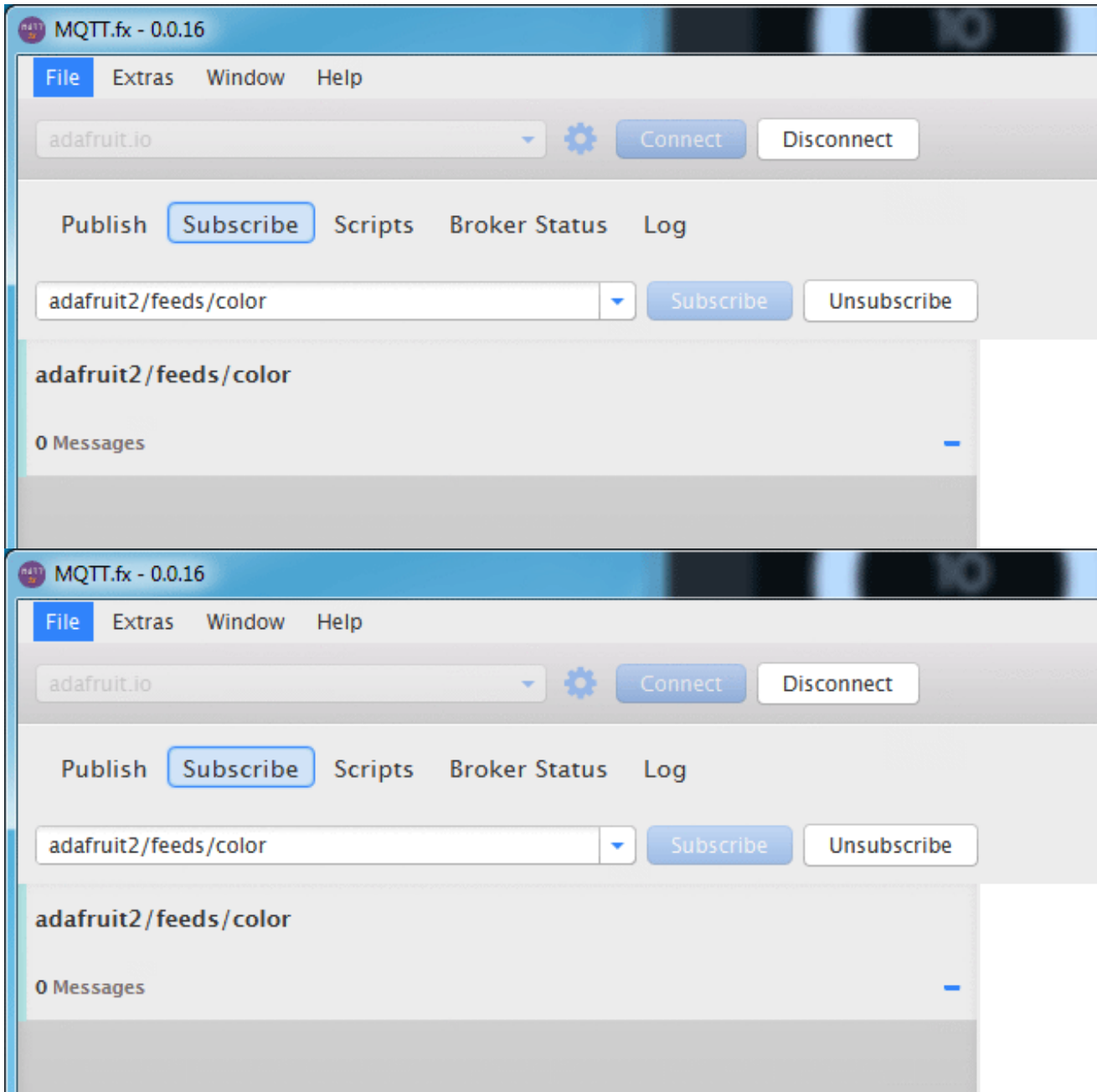
Type some text or data or numbers or whatever into the big box below and then click the **Publish** button to the right of the feed name entry box



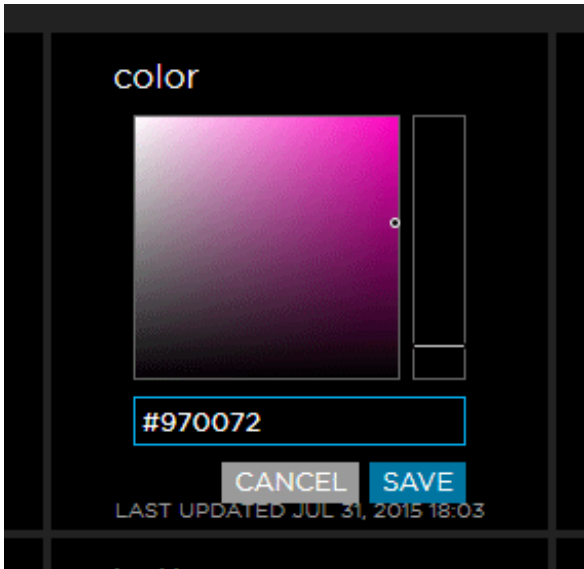
Subscribing

Subscribing to data feeds is pretty much the same, in the **Subscribe** tab, enter in the feed

name in the `username/feeds/feedname` format and click **Subscribe** to the right of the feed entry box



Then on your dashboard you can make a change



And it will be received in the desktop

