MQTT Client provider

(Publisher)

Version 1.3.1

User's guide

August 30 2021

[Remarks]

[Revision history]

Version	Date	Content
1.0.0	2016-12-15	First edition.
1.1.0	2017-07-19	"Type option" was added to the AddVariable command.
1.2.0	2018-05-29	Added exclusive control to initialization and termination of Mosquitto library. Supports SSL communication.
1.2.1	2018-07-31	"AddController" was corrected.
1.2.2	2019-07-24	CodePage support for topics and messages.
1.3.0	2020-05-21	Change client authentication for secure communication arbitrarily.
1.3.0	2020-08-31	Added cause and solution for error code 0x8010000F.
1.3.1	2021-08-30	Internal code optimization.

[Hardware]

Model	Version	Notes

Content

1. Introduction

This document is a user's guide of a CAO provider that works as a MQTT client to publish messages for MTQQ Broker.

CAO provider described in this document is called PUBLISHER provider (CaoProvMQTT_PUBLISHER.dll).

Chapter2 describes the overview of PUBLISHER provider, and details about variables.

PUBLISHER provider uses an open source Mosquitto library to use MQTT protocol for communication.

For details about MQTT protocol and Mosquitto library, refer to the following URL.

The operation of MQTT Client as SUBSCRIBER is implemented in the SUBSCRIBER provider (CaoProvMQTT_SUBSCRIBER.dll) that is another MQTT Client provider.

For details about SUBSCRIBER provider, refer to SUBSCRIBER provider user's guide.

In order to use this provider, "Visual C++ Redistributable for Visual Studio 2015" is required.

[Reference site links]

http://mqtt.org/

http://mosquitto.org/

2. Outline of this provider

2.1. Outline

PUBLISHER provider is a CAO provider that publishes messages to MQTT Broker with Mosquitto library. The file format is DLL (Dynamic Link Library) and it is dynamically uploaded from the CAO engine. To use PUBLISHER provider, you need to install ORiN2SDK, or, complete registration manually with the procedure shown below.

Table 2-1 PUBLISHER provider

File name	CaoProvMQTT_PUBLISHER.dll
ProgID	CaoProv.MQTT_PUBLISHER
Registration	regsvr32 CaoProvMQTT_PUBLISHER.dll
De-registration	regsvr32 /u CaoProvMQTT_PUBLISHER.dll

2.2. Method and Properties

2.2.1. CaoWorkspace::AddController method

PUBLISHER provider establishes communication by referring to the connection parameters for communication with MQTT Broker when AddController is executed. (It functions as a MQTT Client.)

Syntax AddController(<bstrCtrlName:BSTR>,<bstrProvName:BSTR>,
<bstrPCName:BSTR>,<bstrOption:BSTR>))

bstrCtrlName : [in] Controller name
bstrProvName : [in] Provider name. Fixed to " CaoProv.MQTT_PUBLISHER"
bstrPcName : [in] Computer name where the provider runs.
bstrOption : [in] Option string

The following shows a list of option string items.

Table 2-2 Option strings of CaoWorkspace::AddController

Option [1]	Description
Conn=<host>[:<port>[:<keepalive>]]]	Required. Connection parameter. (Refer to 2.2.1.1)
ID[=<ID string>]	ID string of a Client (When omitting this, it generates a unique ID based on GUIDE instead.)
User[=<User name>[:<Password>]]	User/Password authorization setting (This entry remains empty if it is omitted.)
MqttProtocol[=<version>]	MQTT protocol version used for communication. (Refer to 2.2.1.2)
Will[=<topic>:<payload>[:<qos>[:<retain>]]]	Will settings (Refer to 2.2.1.3)
ConnTimeout[=<Timeout period >]	Timeout period for connection (millisecond) (Default : 500)
Timeout[=<Timeout period >]	Timeout period at the request sending (millisecond) (Default : 500)
CA[=<CA certificate file name >][2]	Specify a certificate file name of CA (certificate

[1] Items enclosed with square brackets ("[]") are omittable. Underlined part shows the default value when the option is not specified.
[2] Secure communication is performed when the CA option is specified. To perform client authentication, specify Certificate, PrivateKey, PKPassword.

	authority). (Default: null)
Certificate[=<Certificate file name >][2]	Specify a certificate file name. (Default: null)
PrivateKey[=<Private key file name >][2]	Specify a private key file name. (Default: null)
PKPassword[=<Password>][2]	Specify a password of private key
Insecure[=<True/False>]	Specify whether to skip the server's certificate and server consistency check when connecting secure communications. True: Skip the check. False: Do not skip check (default)

2.2.1.1. Conn option

The following shows connection parameter strings for Conn option.

"Conn=<host>[:<port>[:<keepalive>]]]"

<host>	:	Connection destination host name or IP address Example : "localhost", "192.168.0.1"
<port>	:	Connection port number 1883, 8883, … any number (Default: 1883)
<keepalive>	:	Keep arrive value that is set to MQTT Broker. (sec) (Default: 60)

2.2.1.2. MqttProtocol option

The following shows connection parameter strings for MqttProtocol option.

"MqttProtocol[=<version>]"

<version>	:	MQTT protocol version used for communication. Either V3.1 or V3.1.1 is available. v31, v311 (Default: v311)

2.2.1.3. Will option

The following shows connection parameter strings for Will option.

Will[=<topic>:<payload>[:<qos>[:<retain>]]]

<topic>	:	Topic
<payload>	:	Text message in Will.
<qos>	:	QoS. 0, 1, 2. (Default : 0)
<retain>	:	Retain. True, False. (Default: False)

2.2.2. CaoController::AddVariable method

AddVariable method of CaoController class is a method to create variable objects that enables each provider to publish.

To execute Publish, use the variable name as a topic. (Refer to 2.2.2.1)

Syntax AddVariable(<bstrVariableName:VT_BSTR>[,<bstrOption:VT_BSTR>])

<bstrVariableName> : [in] Variable Name

<bstrOption> : [in] Option character string

The following shows a list of option character string items.

Table 2-3 Option strings of CaoController::AddVariable

Option [3]	Description
QoS[=<0/1/2>]	QoS at Publish. 0,1,2
Retain[=<True/False>]	Retain at Publish. True, False
Type[=<0/1>]	Data type at Publish. 0 (VT_BSTR), 1 (VT_UI1|VT_ARRAYY)
CodePage[=<Code page>]	Code page of the message to publish. 0 or higher (Refer to 2.2.2.2)

[3] Items enclosed with square brackets ("[]") are omittable. Underlined part shows the default value when the option is not specified..

2.2.2.1. Specifying a topic with a variable name

PUBLISHER provider uses a variable name as a topic.

A topic has a format of layer using "/" (slash) as a delimiter. It also can be specified with wildcards, such as "#" (sharp) or "+" (plus). (You can combine such wildcards to use.)

/ : Delimiter of layer structure

: All topics in this layer and lower layers will be the target.

+ : Treat this layer as a wildcard

Example) Office/# All topics in "Office" and the lower layer will be the target.

Match : Office/Tokyo

Match : Office/Osaka

Match : Office/Nagoya/4F

Do not match : House/Tokyo/1F

Example) Office/+/3F If the first and the third layer matches, any topic is allowed in the second layer.

Match : Office/Tokyo/3F

Match : Office/Osaka/3F

Do not match : Office/Nagoya/4F

Do not match : Office/Tokyo/1F

2.2.2.2. Code page specification

Specify code page or constant.

2.2.2.2.1. Code page

Example)

932	:	Japanese(Shift JIS)
1200	:	Unicode
65001	:	Unicode(UTF-8)

2.2.2.2.2. Constant

0	:	The system default Windows ANSI code page.
1	:	The current system OEM code page.
2	:	The current system Macintosh code page.
3	:	The Windows ANSI code page for the current thread.
42	:	Symbol code page.

2.2.3. CaoVariable:put_Value property

Execute Publish with a variable name as a topic, and a string specified by Value (VT_BSTR) as a Payload.

2.2.4. CaoVariable:get_Value property

Return a value (VT_BSTR) that has executed Publish the last.

2.2.5. CaoController::OnMessage event

The following cause triggers an OnMessage event of CaoController class.

Table 2-4 Message type

Message type		Trigger
1	Disconnection detected	This message occurs when a client disconnects from MQTT Broker by any causes.

2.2.5.1. Disconnection detected

Data type obtained by the disconnection detection message is as follows.

Number : Message type (1)

Value : Response code of the cause of disconnection (VT_I4)

DateTime : Time stamp

Destination : Null

Source : Connection destination host name or IP address

Description : Text description of the response code

2.3. Variable list

2.3.1. CaoController class

Table 2-5 CaoController class User variable list

Variable name	Data type	Description	Attribute	
			get	put
Arbitrary (used as a topic)	VT_BSTR	Execute "Publish" by "put_Value".	✓	✓

Table 2-6 CaoController class System variable list

Variable name	Data type	Description	Attribute	
			get	put
@VERSION	VT_BSTR	Provider version information	✓	—
@MQTT_LIB_VERSION	VT_BSTR	Mosquitto library Version information	✓	—

2.4. Error code

In PUBLISHER provider, the following original error codes are defined.

For about ORiN2 common errors, refer to the error code section of ORiN2 Programming guide.

Table 2-7 Specific error code list

Error name	Error number	Description
E_MOSQ_CONN_PENDING	0x80100001	Error code of Mosquitto library MOSQ_ERR_CONN_PENDING occurred.
E_MOSQ_NOMEM	0x80100002	Error code of Mosquitto library MOSQ_ERR_NOMEM occurred.
E_MOSQ_PROTOCOL	0x80100003	Error code of Mosquitto library MOSQ_ERR_PROTOCOL occurred.
E_MOSQ_INVAL	0x80100004	Error code of Mosquitto library MOSQ_ERR_INVAL occurred.
E_MOSQ_NO_CONN	0x80100005	Error code of Mosquitto library MOSQ_ERR_NO_CONN occurred.
E_MOSQ_CONN_REFUSED	0x80100006	Error code of Mosquitto library MOSQ_ERR_CONN_REFUSED occurred.
E_MOSQ_NOT_FOUND	0x80100007	Error code of Mosquitto library MOSQ_ERR_NOT_FOUND occurred.
E_MOSQ_CONN_LOST	0x80100008	Error code of Mosquitto library MOSQ_ERR_CONN_LOST occurred.
E_MOSQ_TLS	0x80100009	Error code of Mosquitto library MOSQ_ERR_TLS occurred.
E_MOSQ_PAYLOAD_SIZE	0x8010000A	Error code of Mosquitto library MOSQ_ERR_PAYLOAD_SIZE occurred.
E_MOSQ_NOT_SUPPORTED	0x8010000B	Error code of Mosquitto library MOSQ_ERR_NOT_SUPPORTED occurred.
E_MOSQ_AUTH	0x8010000C	Error code of Mosquitto library MOSQ_ERR_AUTH occurred.
E_MOSQ_ACL_DENIED	0x8010000D	Error code of Mosquitto library MOSQ_ERR_ACL_DENIED occurred.
E_MOSQ_UNKNOWN	0x8010000E	Error code of Mosquitto library MOSQ_ERR_UNKNOWN occurred.

E_MOSQ_ERRNO	0x8010000F	Error code of Mosquitto library MOSQ_ERR_ERRNO occurred. The MQTT Broker of the connection destination may not be running. Start MQTT Broker and reconnect.
E_MOSQ_EAI	0x80100010	Error code of Mosquitto library MOSQ_ERR_EAI occurred.
E_MOSQ_PROXY	0x80100011	Error code of Mosquitto library MOSQ_ERR_PROXY occurred.

Appendix A. Terms of MQTT protocol

This chapter describes terms used in the MQTT protocol.

- **Broker**

  A server of the MQTT protocol communication.

  A Broker receives a topic message from a Publisher and then provides it to an interested Subscriber.

- **Subscriber**

  A Client on the MQTT protocol communication (message receiving side).

  A Subscriber registers at a Broker to subscribe a desired topic, and then receive messages.

- **Publisher**

  A Client on the MQTT protocol communication (message sending side).

  A Publisher sends messages to a Broker with specifying a topic. There is no need of thinking about Subscriber.

- **QoS**

  QoS defines how difficult the Broker/Client tries to assure that a message is received.

  QoS0 : The Broker/Client delivers the message at most once.

  QoS1 : The Broker/Client delivers the message at least once.

  QoS2 : The Broker/Client delivers the message exactly once.

- **Will**

  When a Client connects to a Broker, the Client can inform the Broker that it has a Will.

  If the Client disconnects from Broker unexpectedly, a topic and a message specified by Will are sent to a Subscriber which connects to the Broker.

- **Retain**

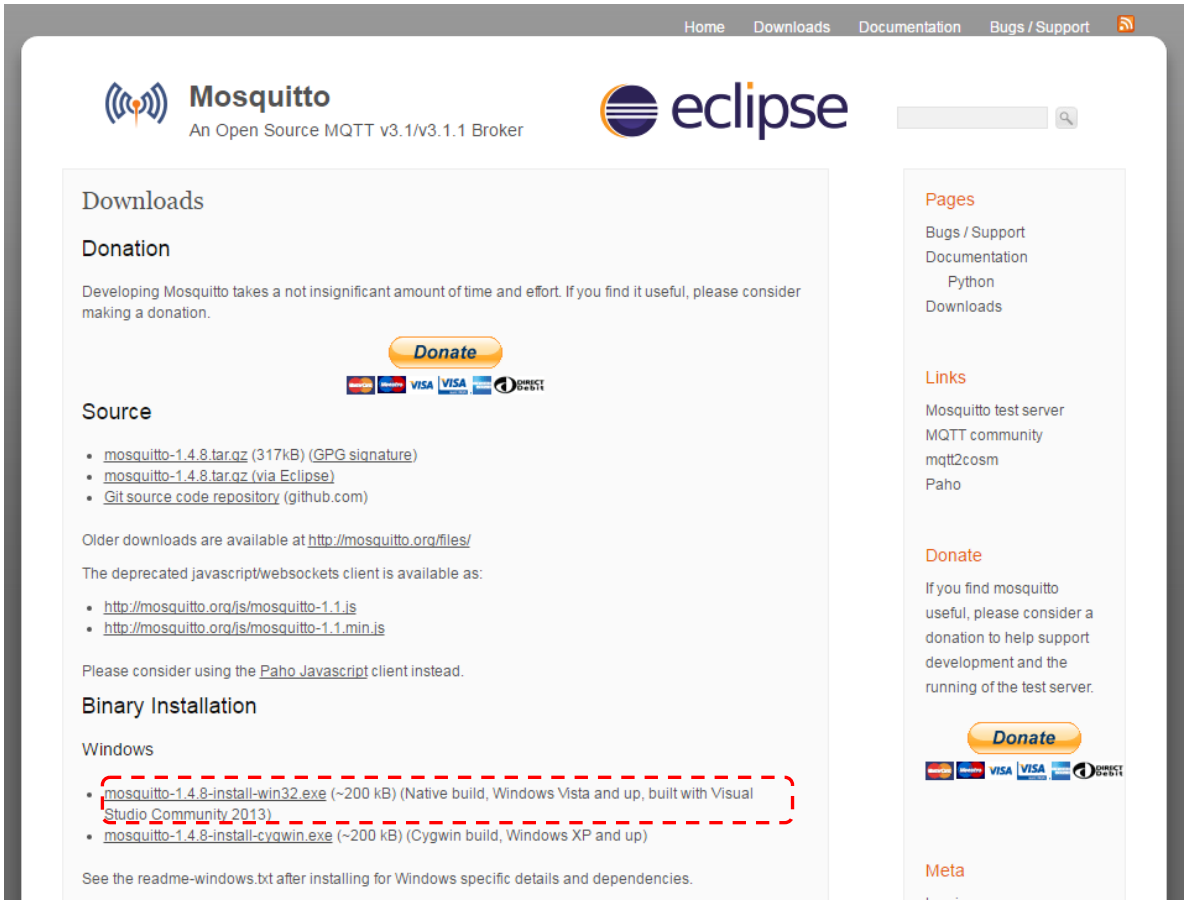
  "Retain" is a function that Broker retains the latest message in each topic after publishing it. If a new subscription is made that matches the topic of the retained message, the message is sent to the subscriber.

Appendix B. Implementation of MQTT Broker

This chapter explains how to implement MQTT Broker with Mosquitto that is an open source message broker.

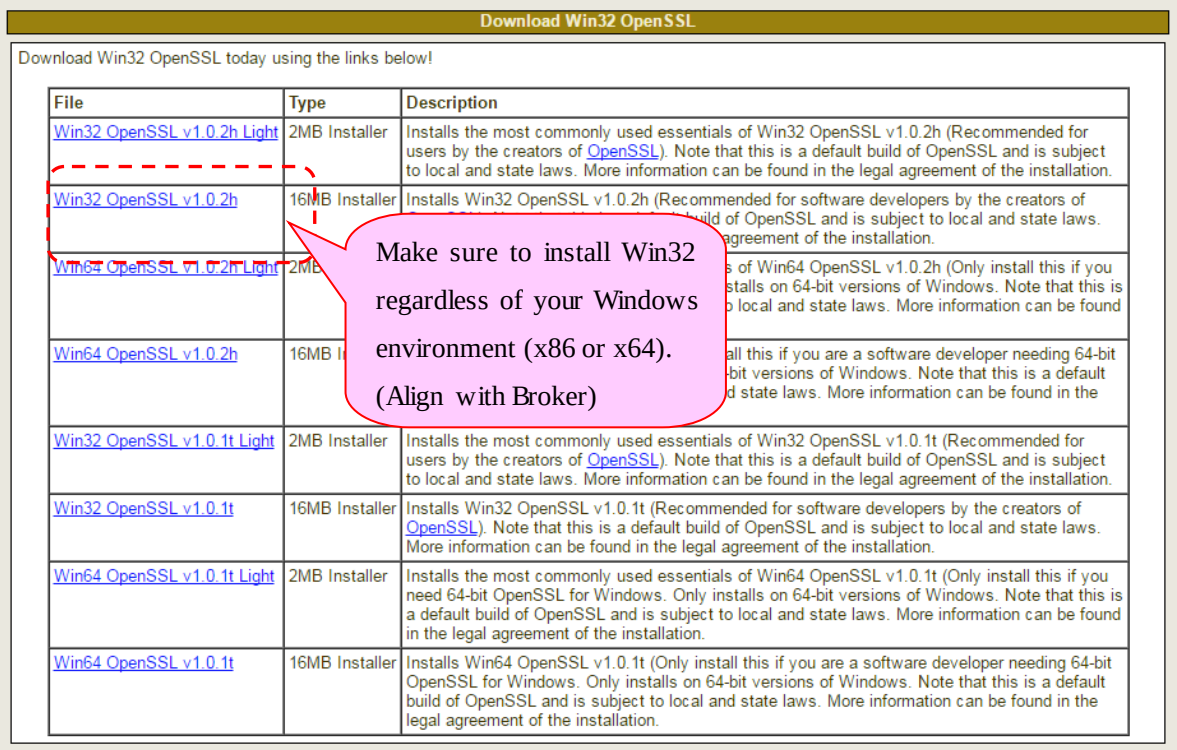
[For Windows]

1. Download an installer from http://mosquitto.org/download/ .

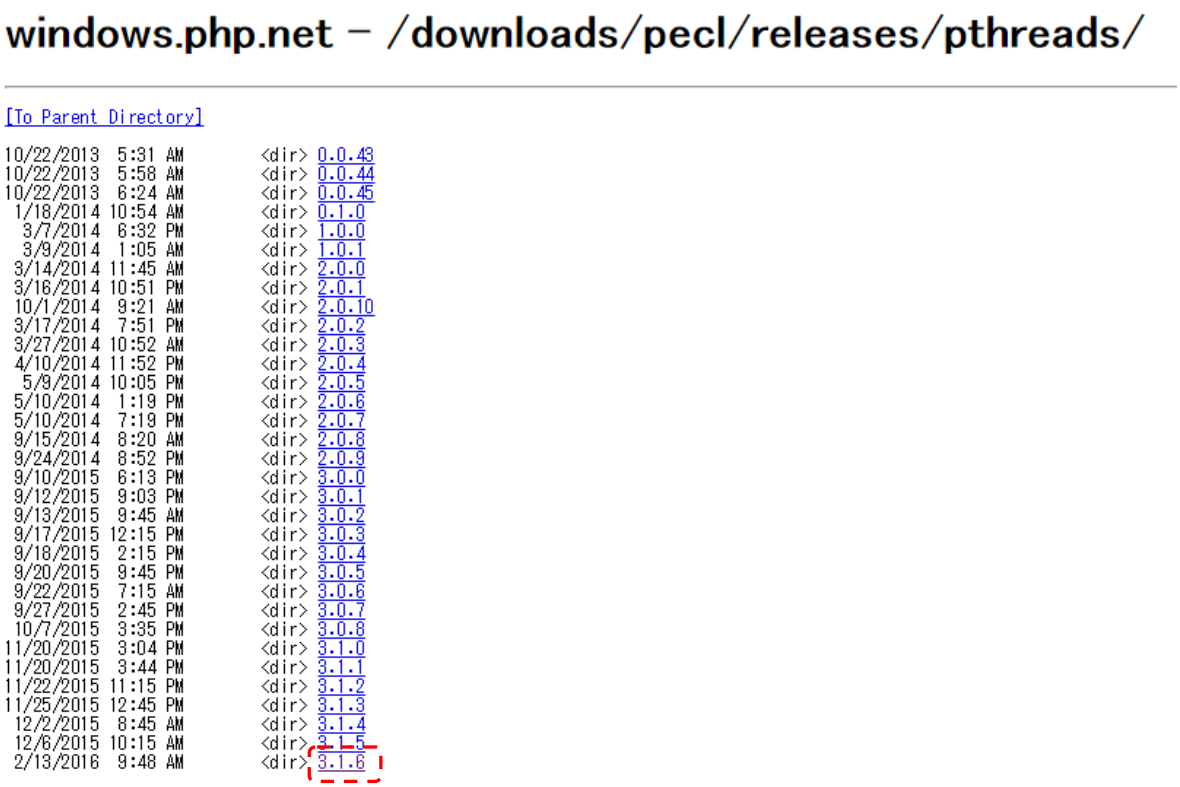


2. Execute the downloaded installer.
3. You will find a message that asks you to install OpenSSL (libeay32.dll, ssleay32.dll) and pthreads (pthreadVC2.dll) manually though, ignore this message in this step and go next. (Details are explained later.
4. Leave the components as default settings.
5. Change the install destination to "C:¥mosquitt" from "C:¥Program Files (x86)¥mosquitto", which is the default setting. (This is because the default path includes spaces that makes difficult to later settings.)
6. Complete the installation.

7. Download the Win32 installer (the latest version) of OpenSSL that have been compiled from http://slproweb.com/products/Win32OpenSSL.html. (Leave as a default.)



8. Download an compressed file (the latest version) of pthreads from http://windows.php.net/downloads/pecl/releases/pthreads/ and then expand it to any directory.



9. Add the path to OpenSSL and pthreads.
10. Restart your computer.

Now you can use Mosquitto Broker. In default setting, Mosquitto Broker starts as a service at the startup of your computer. You need to close a running service if you start an option by specifying with a command line or read a conf file which setting has been modified.

[For Linux (Ubuntu)]

1. Start Terminal, and then execute the following commands.
2. Update a package list.

```
$ sudo apt-get update
```

3. Implement a Broker.

```
$ sudo apt-get install mosquitto
```

4. Implement a Client.

```
$ sudo apt-get install mosquitto-clients
```

If any dependent packages such as OpenSSL are not installed, you need to install it separately.

If mosquitto does not appear in the package list, add a repository with the following command.

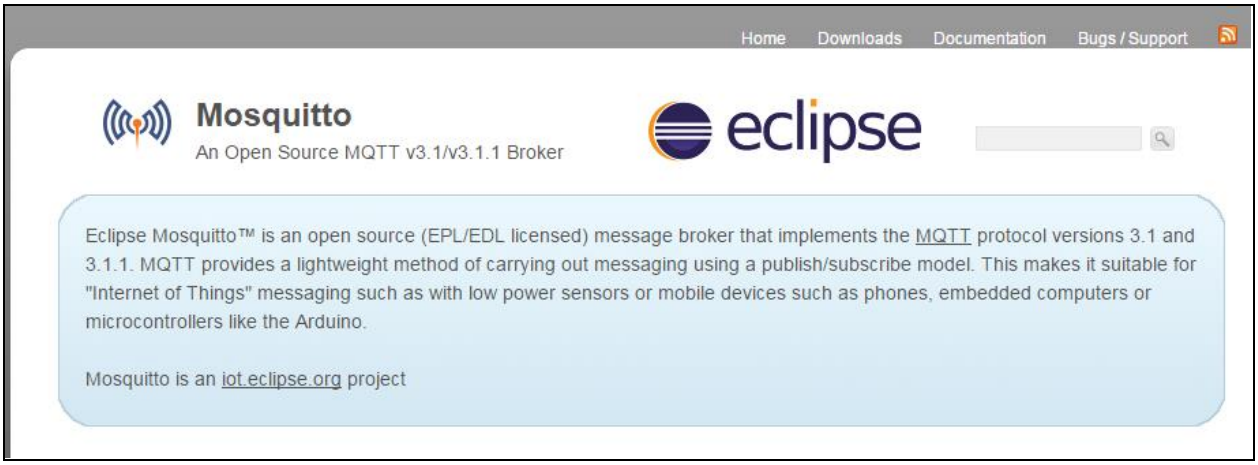
```
$ sudo add-apt-repository ppa:mosquitto-dev/mosquitto-ppa
```

Once the Mosquitto package has been implemented, Broker becomes a service-state.

Just like the Windows environment, you need to close a running service if you start an option by specifying with a command line or read a conf file which setting has been modified.

Appendix C. Mosquitto

URL: https://mosquitto.org/



EPL

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes

performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement , including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

EDL

Eclipse Distribution License - v 1.0

Copyright (c) 2007, Eclipse Foundation, Inc. and its licensors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Appendix D. Pthread_win32

URL: http://sources.redhat.com/pthreads-win32

Open Source
POSIX Threads for Win32

Current release: 2.9.1 (2012-05-27)
See the ANNOUNCEMENT and NEWS for details.

Pthreads-w32 release 1.11.0 (2005-06-04) was the last series 1.x.x version!
(Version 1.11.0 is a back-port of the 2.7.0 functionality and bug fixes. See the NEWS file inside the package for more information.)

What is this project about?

The POSIX 1003.1-2001 standard defines an application programming interface (API) for writing multithreaded applications. This interface is known more commonly as *pthreads*. A good number of modern operating systems include a threading library of some kind: Solaris (UI) threads, Win32 threads, DCE threads, DECthreads, or any of the draft revisions of the pthreads standard. The trend is that most of these systems are slowly adopting the pthreads standard API, with application developers following suit to reduce porting woes.

Win32 does not, and is unlikely to ever, support pthreads natively. This project seeks to provide a freely available and high-quality solution to this problem.

Various individuals have been working on independent implementations of this well-documented and standardised threading API, but most of them never see the light of day. The tendency is for people to only implement what they personally need, and that usually does not help others. This project attempts to consolidate these implementations into one implementation of pthreads for Win32.

License

This implementation is free software, distributed under the GNU Lesser General Public License (LGPL).

LGPL

```
        pthreads-win32 - a POSIX threads library for Microsoft Windows


This file is Copyrighted
------------------------

    This file is covered under the following Copyright:

        Copyright (C) 2001,2006 Ross P. Johnson
        All rights reserved.
```

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Pthreads-win32 is covered by the GNU Lesser General Public License
---

Pthreads-win32 is open software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation version 2.1 of the License.

Pthreads-win32 is several binary link libraries, several modules, associated interface definition files and scripts used to control its compilation and installation.

Pthreads-win32 is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

A copy of the GNU Lesser General Public License is distributed with pthreads-win32 under the filename:

COPYING.LIB

You should have received a copy of the version 2.1 GNU Lesser General Public License with pthreads-win32; if not, write to:

Free Software Foundation, Inc.
59 Temple Place
Suite 330
Boston, MA 02111-1307
USA

The contact addresses for pthreads-win32 is as follows:

Web: http://sources.redhat.com/pthreads-win32

Email: Ross Johnson
Please use: Firstname.Lastname@homemail.com.au

Pthreads-win32 copyrights and exception files

With the exception of the files listed below, Pthreads-win32
is covered under the following GNU Lesser General Public License
Copyrights:

Pthreads-win32 - POSIX Threads Library for Win32
Copyright(C) 1998 John E. Bossom
Copyright(C) 1999,2006 Pthreads-win32 contributors

The current list of contributors is contained
in the file CONTRIBUTORS included with the source
code distribution. The current list of CONTRIBUTORS
can also be seen at the following WWW location:
http://sources.redhat.com/pthreads-win32/contributors.html

Contact Email: Ross Johnson
Please use: Firstname.Lastname@homemail.com.au

These files are not covered under one of the Copyrights listed above:

COPYING
COPYING.LIB
tests/rwlock7.c

This file, COPYING, is distributed under the Copyright found at the
top of this file. It is important to note that you may distribute
verbatim copies of this file but you may not modify this file.

The file COPYING.LIB, which contains a copy of the version 2.1
GNU Lesser General Public License, is itself copyrighted by the

Free Software Foundation, Inc. Please note that the Free Software Foundation, Inc. does NOT have a copyright over Pthreads-win32, only the COPYING.LIB that is supplied with pthreads-win32.

The file tests/rwlock7.c is derived from code written by Dave Butenhof for his book 'Programming With POSIX(R) Threads'. The original code was obtained by free download from his website http://home.earthlink.net/~anneart/family/Threads/source.html and did not contain a copyright or author notice. It is assumed to be freely distributable.

In all cases one may use and distribute these exception files freely. And because one may freely distribute the LGPL covered files, the entire pthreads-win32 source may be freely used and distributed.

General Copyleft and License info

For general information on Copylefts, see:

http://www.gnu.org/copyleft/

For information on GNU Lesser General Public Licenses, see:

http://www.gnu.org/copyleft/lesser.html
http://www.gnu.org/copyleft/lesser.txt

Why pthreads-win32 did not use the GNU General Public License
---

The goal of the pthreads-win32 project has been to provide a quality and complete implementation of the POSIX threads API for Microsoft Windows within the limits imposed by virtue of it being a stand-alone library and not

linked directly to other POSIX compliant libraries. For example, some functions and features, such as those based on POSIX signals, are missing.

Pthreads-win32 is a library, available in several different versions depending on supported compilers, and may be used as a dynamically linked module or a statically linked set of binary modules. It is not an application on it's own.

It was fully intended that pthreads-win32 be usable with commercial software not covered by either the GPL or the LGPL licenses. Pthreads-win32 has many contributors to it's code base, many of whom have done so because they have used the library in commercial or proprietry software projects.

Releasing pthreads-win32 under the LGPL ensures that the library can be used widely, while at the same time ensures that bug fixes and improvements to the pthreads-win32 code itself is returned to benefit all current and future users of the library.

Although pthreads-win32 makes it possible for applications that use POSIX threads to be ported to Win32 platforms, the broader goal of the project is to encourage the use of open standards, and in particular, to make it just a little easier for developers writing Win32 applications to consider widening the potential market for their products.