



Software Version: 1.0.3.5

-Base on mobius-yt-r2-2.4.1

User Guide v1.2

Document Release Date: July 11, 2017

Software Release Date: July 11, 2017



Copyright (c) 2014, OCEAN
All rights reserved.

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) 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 OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Contents

1. Introduction.....	4
1.1. What is oneM2MBrowser	4
1.2. Supported platform.....	4
1.3. oneM2M Browser function	5
2. Architecture	5
2.1. System Architecture	5
2.2. Software Architecture	6
3. Development Environment.....	6
3.1. Download Source	6
3.2. Development Tool	7
3.3. Debug and Running.....	7
3.4. Reference Libraries	8
4. oneM2M Browser User Interface	9

1. Introduction

1.1. What is oneM2MBrowser

oneM2MBrowser is a tool for monitoring the oneM2M resources in the Mobius Yellow Turtle(Mobius-yt). It is working base on oneM2M HTTP RESTful API and MQTT Message. It was designed for helping developer to easily work with Mobius-yt.

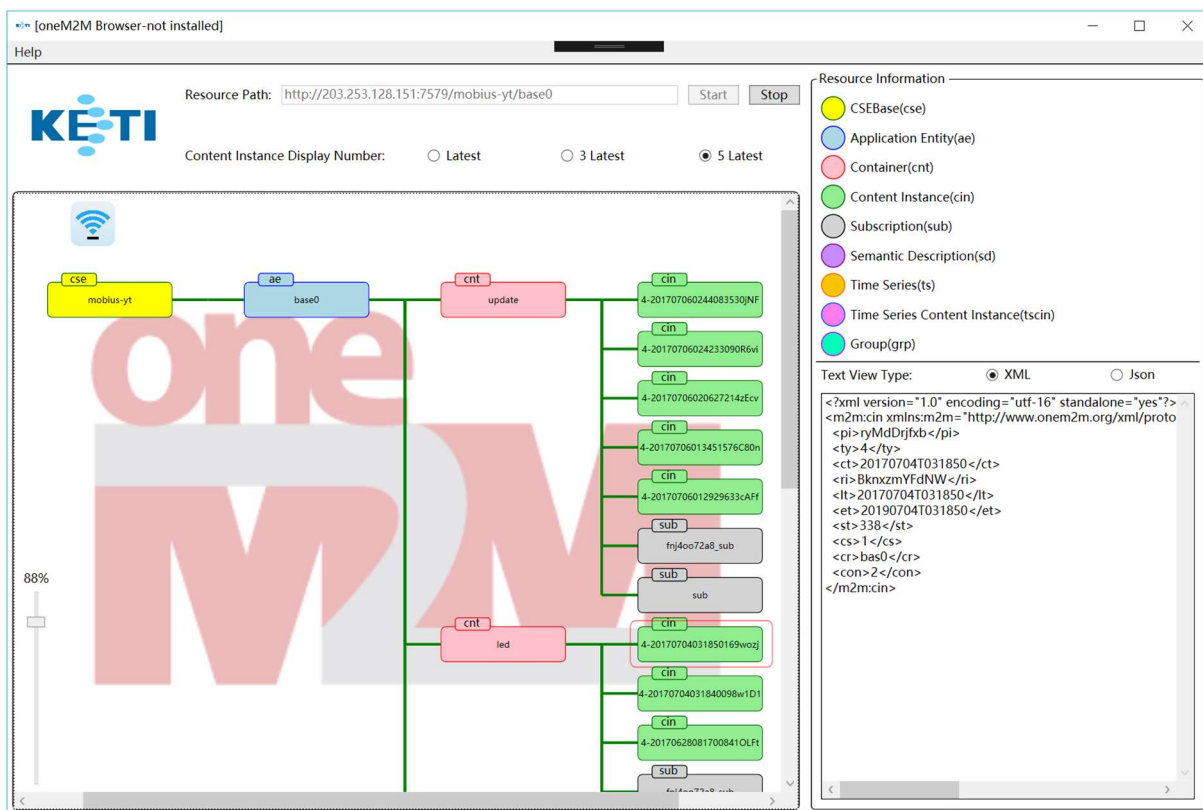


Figure 1 oneM2M Browser

1.2. Supported platform

It only works on Windows OS (windows XP, Vista, 7, 8, 10) and it also require the OS must be installed the .NET Framework.

1.3. oneM2M Browser function

1. Display the oneM2M resource as tree.
2. Retrieve the resource information.
3. Create new resource.
4. Real-time monitoring resource (delete, create).

2. Architecture

2.1. System Architecture

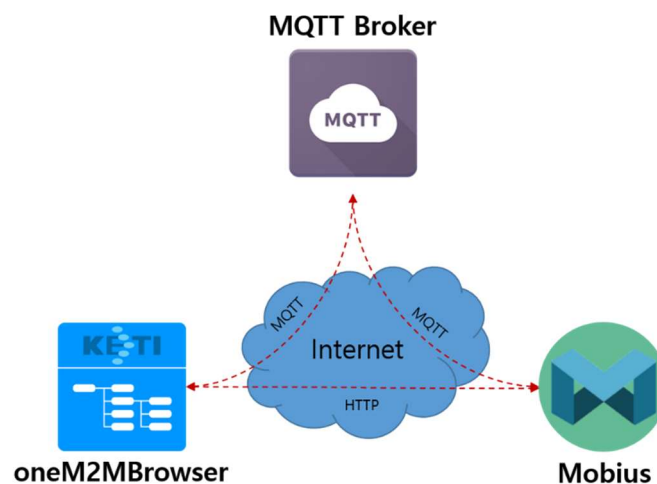


Figure 2 System Architecture

The oneM2MBrowser connects with Mobius IoT Server Platform and MQTT Broker. Mobius is a kind of IoT Server Platform collection all oneM2M (IoT Standard) Resources and provides a set of open API which binding with HTTP, COAP, MQTT protocol for managing resource inside. In oneM2MBrowser case, it retrieves the resource structure by discovery API and display it as GUI and create subscription for necessary resources for receiving resource creating or deleting notification message.

2.2. Software Architecture

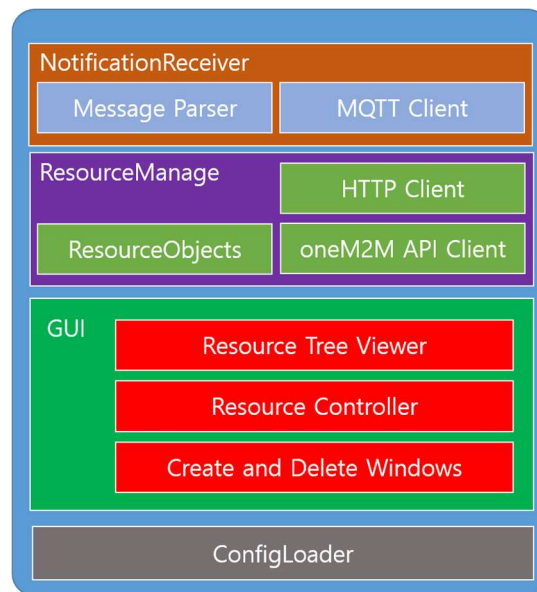


Figure 3 software Architecture

The oneM2MBrowser architecture can be divided into four big modules. "NotificationReceiver" can receive MQTT notification message and parse it then add or delete matched resource in "ResourceManager". "ResourceManager" provide a series of oneM2M API accessing function and it can discovery Mobius inside resource structure by these API then store it in the memory as objects. "GUI" provides some windows and dialogs for user interaction. "ConfigLoader" is used to store local application parameter.

3. Development Environment

3.1. Download Source

Download the source file from oneM2MBrowser github site:

<https://github.com/IoTKETI/oneM2MBrowser>

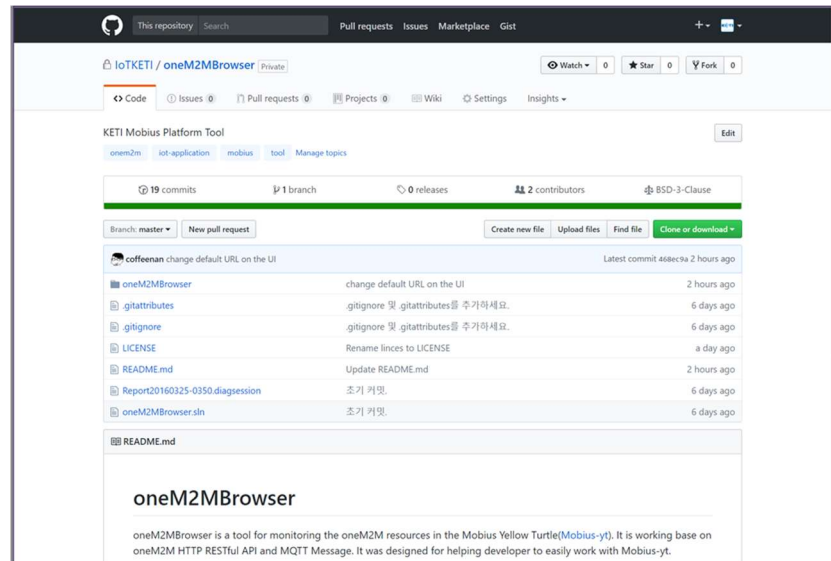


Figure 4 oneM2M Browser GitHub site

3.2. Development Tool

The oneM2MBrowser project is a Visual Studio project. So, it only can work on window PC. If you don't have a Windows PC that we suggest to install the VMware WorkStation Player and run a virtual machine on it.

1. Download the [Visual Studio](#).
2. Install the Visual Studio follow the guide.
3. Open the oneM2MBrowser home and double click oneM2MBrowser.sln file.

3.3. Debug and Running

Press the F5 key in Visual Studio click the start button on top of Visual Studio as Figure 5.

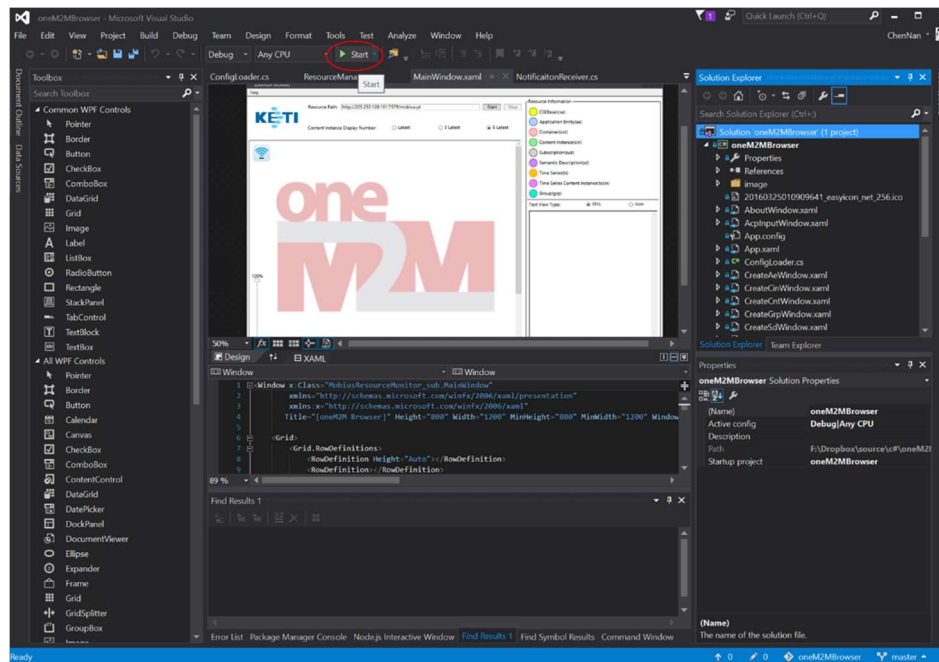


Figure 5 Debug the oneM2M Browser in visual studio

The oneM2MBrowser is developed base on [WPF application](#) which include in .net framework. So, the UI part is written with [XAML](#) language and logic part is written with C#. If you are not familiar with XAML language the Microsoft also provide [Microsoft Blend](#) for helping developer to edit UI XAML.

3.4. Reference Libraries

- [Newtonsoft.Json](#) - The JSON format parser.
- [M2Mqtt](#) - The MQTT message protocol used.
- [uHttpSharp](#) - The light HTTP server library.
- [WPFToolKit](#) - The extension WPF controller package.

These libraries have been contained in project so that dose not need to install it manually.

4. oneM2M Browser User Interface

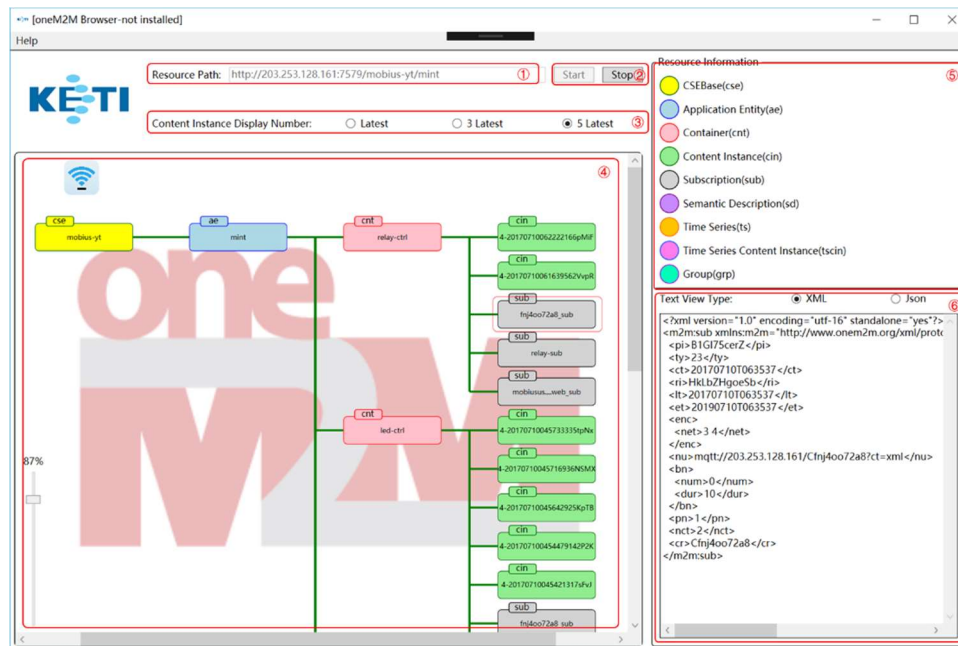


Figure 6 Main window

오류! 참조 원본을 찾을 수 없습니다. show main window of the oneM2M Browser. It is the core GUI in this project.

1. A text box for input resource path. In oneM2M standard, every resource should have a unique path.
2. Start and stop button for starting monitoring resource process or stop it.
3. Content Instance Display Number is used to limit the display content instance number. Too much content instance display on the view that maybe cause a crash in program. So, we only support to show latest 5, 3 or 1 content instance in application.
4. This is resource tree viewer of oneM2MBrowser. In here you can see all the child resources of the resource which you input in the ① area. If some new resource was

created or some resource was deleted then it will synchronously display on the viewer. In the version 1.0.3.5 we have already added the resource creation and deleting function. So, user can use the mouse to right click resource for create and delete resource directly.

5. Show the resource color of each supported resource type.
6. Show the resource information. In oneM2M standard, each type of resource has a series of attributes and it can be serialized to a JSON or XML string message.