

---

**Mini Pulp Process**

# **API Update Instructions**

**Version 1.2.0.0**

---

Eero Eriksson
Document state: Complete      Modified: 27.02.2025

## CONTENTS

---

1	Introduction .....	3
2	MppOpcUaClientLib .....	4
2.1	SDK Downloads .....	4
2.2	Porting the source code .....	4
2.2.1	Source code not available .....	4
2.3	Updated DLL .....	5
2.4	Documents .....	5
2.5	Full MppOpcUaClientLib .....	6
3	Node-OPCUA module .....	7
3.1	Getting the module .....	7
3.2	If the server breaks .....	7
3.2.1	Namespaces .....	7

---

**1**

---

**INTRODUCTION**

---

This document is a guide to updating the OPC UA API of the Mini Pulp Process (MPP). It contains instructions to both the client and server. Chapter 2 has the instructions for the MppOpcUaClientLib (Client), and chapter 3 has the instructions for the Node-OPCUA module (Server). While this might not be 100 % accurate in the future, it can act as guideline on what to do when updating the API.

---

**2**

---

**MPPOPCUACLIENTLIB**

---

In this chapter are the instructions for updating the client side of the OPC UA connection of the Mini Pulp Process simulator. It consists of updating the MppOpcUaClientLib library to a new version of .NET. Current version is .NET8.

**2.1****SDK Downloads**

You need to download the new version of the OPC UA SDK. To get the correct one, get the “.NET based OPC UA Client/Server SDK + PubSub Bundle” and run its setup. The needed .dll files should be in the bin folder of the installation. The files used are:

UnifiedAutomation.UaBase.dll

UnifiedAutomation.UaClient.dll

Current ones used in the software are evaluation version 4.1.0.556 with .NET8 support.

OPC UA SDK:

<https://www.unified-automation.com/downloads/opc-ua-development.html>

**2.2****Porting the source code**

The target framework of the library needs to be changed to the desired framework. It can be done manually but the easiest way to port the code is to use an upgrading assistant extension on Visual Studio. Microsoft provides one called “.NET Upgrade Assistant”. The software should be under the same .NET standard as future versions, so it shouldn't break from the upgrading. If something breaks, fix it manually.

.NET Upgrade Assistant:

<https://marketplace.visualstudio.com/items?itemName=ms-dotnettools.upgradeassistant>

**2.2.1****Source code not available**

If the source code isn't available, you can get it by decompiling an existing Tuni.MppOpcUaClientLib.dll file. Decompiling can be done by for example using JetBrains dotPeek decompiler or an appropriate Visual Studio extension. After the disassembly, the source code should look something like figure 1 below. Check for missing files or code.

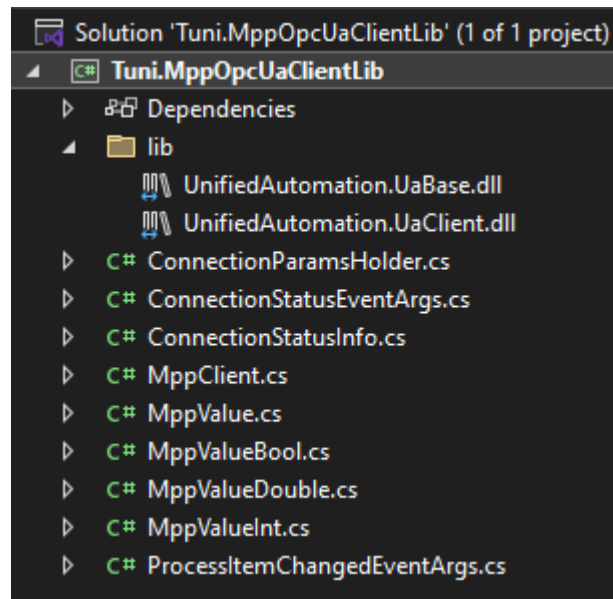


Figure 1: Source code files

dotPeek:

<https://www.jetbrains.com/decompiler/>

## 2.3 Updated DLL

Update the information in the project file before building the code. Make sure the updated source code references the updated UaClient and UaBase. Build the source code. The new .dll files should be in the bin folder.

## 2.4 Documents

Update the documentation provided with the MppOpcUaClientLib if necessary. The HTML documentation is generated with Doxygen. The XML documentation is generated when the class library is built. To enable the generation with Visual Studio, add a check to:

Project -> Properties -> Build -> Output -> “Generate a file containing API documentation”

Doxygen:

<https://www.doxygen.nl/download.html>

## 2.5

### Full MppOpcUaClientLib

The full MppOpcUaClientLib library package currently contains the following files/folders:

- Source code
- .dll files
  - UnifiedAutomation.UaBase.dll
  - UnifiedAutomation.UaClient.dll
  - Tuni.MppOpcUaClientLib.dll
- Documentation
  - README
  - MppOpcUaClientLib\_html\_doc (Doxygen)
  - Tuni.MppOpcUaClientLib.xml
  - API Update Instructions (This document)

---

**3****NODE-OPCUA MODULE**

---

In this chapter are the instructions for updating the server side of the OPC UA connection of the Mini Pulp Process simulator. It consists of updating the Node-OPCUA module to a new version.

**3.1****Getting the module**

Update the module used in the project folder of the NodeJs simulator with the command:

```
npm update node-opcua
```

The current version of the module is 2.148.0.

Node-OPCUA:

<https://github.com/node-opcua/node-opcua>

**3.2****If the server breaks**

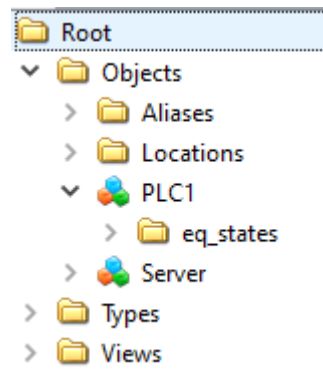
If the server breaks and the error message doesn't provide enough information, check the release notes of the module since the current version. They have listed all the breaking changes of each update, so most likely one of those is the cause for the server to break. Also, the sample server in the comments of the serve can help with the problem.

Node-OPCUA releases:

<https://github.com/node-opcua/node-opcua/releases>

**3.2.1****Namespaces**

If there is a problem with the namespaces, you can use UaExpert to see what the server exposes to the outside. The address space of the server should look like figure 2 with the server being index number 1 and the PLC1 being number 2. The actuators are in the eq\_states folder as variables. The Client checks the indexes, so they need to be correct to make the subscriptions work.



*Figure 2: Address space of the server*

UaExpert:

<https://www.unified-automation.com/products/development-tools/uaexpert.html>