



ConnectOPC-UA

Software Design Document

For the specific implementation of OPC-UA Protocol

Bhupendra Kumar Mishra
Electronics Maintenance (EMX)
Bharat Heavy Electricals Limited (BHEL)

Revision History: 1.0.0.2349

Bhopal
August 2019

Contents

1.	Introduction.....	7
1.1	Purpose.....	8
1.2	Scope	8
1.3	Definitions, Acronyms and Abbreviations.....	8
1.4	Intended Audience	10
1.5	List of Components/Modules Used.....	11
1.6	References	11
1.7	Content Summary.....	11
2.	System Overview.....	12
2.1	System Architecture	13
2.2	Infrastructure Services.....	15
3.	System Design.....	16
3.1	System Requirements.....	16
3.2	Setting up development environment.....	17
3.2.1	Installing Node Js.....	17
3.2.2	Installing NPM	17
3.2.3	Installing Node-OPCUA.....	18
3.2.4	Installing Async	18
3.2.5	Installing json2xls.....	19
3.3	System Components.....	19
4.	Modules.....	23
4.1	Configure Machine	23
4.2	Connect Machine.....	29
4.3	Browse Address Space	33
4.4	Read/Monitor Variables	37
4.5	Write variables.....	38
4.6	Pub-Sub Notifications.....	41
4.7	Multiple Pub-sub Notifications.....	47
4.8	Creating OPCUA Server.....	50
4.9	Settings.....	55
4.10	Help.....	55
4.11	Menus.....	56
4.12	Method Call.....	59
4.13	Event Notifier.....	63
4.14	File Transfer	66
4.15	Favourites	69
4.16	Node Crawler	71
5.	Deploying the Application.....	73
5.1	Deployment using electron-packager.....	73
5.2	Deployment using electron-builder.....	75
5.3	Installing software on Mac OS.....	76
5.4	Installing Software in Windows.....	78
5.6	Installing Software in Linux.....	80
6.	Case Study with Siemens Sinumerik 828D	81
6.1	Siemens 828D CNC Controller at a glance	81

6.2	Setting up the OPCUA in Siemens 828D	87
6.3	Overview.....	89
6.4	Features.....	92
6.5	Simulation of OPC-UA Server With Sinutrain	95
6.5.1	Activating OPC-UA in Sinutrain	96
6.5.2	Connecting with Sinutrain OPC-UA Server	103
6.5.3	Browsing Address Space of Sinutrain's Configured Machine	105
6.5.4	Monitoring PLC Variables	107
6.5.5	Monitoring NC and other variables variables.....	110
6.5.6	Subscription of Variables.....	120
6.5.7	File Transfer	123
7.	Case Study with ConnectOPC-UA Server.....	128
8.	Source Code	131
8.1	Main.js.....	131
8.2	Configure Machine	192
8.3	Connect Machine.....	197
8.4	Browse Address Space.....	203
8.5	Monitor/Read Variables	212
8.6	Write Variables to the Server.....	216
8.7	Create Pub-Sub Notifications	220
8.8	Multiple Pub-Sub Notifications	232
8.9	Method Call.....	242
8.10	Event Notifier.....	250
8.11	File Transfer	255
8.12	Create OPC-UA Server.....	260
8.13	Settings.....	264
8.14	Help	270
8.15	Favoutites	279
8.16	View Favourites.....	282
8.17	Package.json	289
8.18	Project Dependencies	290
Appendix.....		325
A.1	Installation Files required in Windows 64Bit Environment.....	325
A.2	Installation Files for Linux OS (Ubuntu, Fedora, Debian, Raspberry Pi3)	327
A3	Installation files for Mac OS	335

LIST OF FIGURES

FIGURE 1 OPC UA ARCHITECTURE.....	7
FIGURE 2 OPCUA TARGET APPLICATIONS	13
FIGURE 3 APP DEVELOPMENT WITH ELECTRON JS.....	19
FIGURE 4 COMPONENT DIAGRAM.....	21
FIGURE 5 ACTIVITY DIAGRAM.....	22
FIGURE 6 OPC-UA SECURITY ARCHITECTURE CLIENT-SERVER.....	23
FIGURE 7 CONFIGURATION SETUP MODE1	24
FIGURE 8 ROLES IN OPCUA	25
FIGURE 9 CONFIGURATION SETUP MODE2.....	25
FIGURE 10 CONFIGURATION SETUP MODE3	26
FIGURE 11 CONFIGURATION SETUP MODE4	28
FIGURE 12 MACHINE CONFIGURATION	28
FIGURE 13 CONFIGURE MODULE	29
FIGURE 14 CONNECT MACHINE	30
FIGURE 15 DISCONNECT MACHINE	31
FIGURE 16 DISCOVERY PROCESS	31
FIGURE 17 CONNECT MACHINE MODULE	32
FIGURE 18 MULTIPLEXING USERS ON A SESSION.....	33
FIGURE 19 NODE OPCUA CLASS.....	34
FIGURE 20 ADDRESS SPACE MODEL	34
FIGURE 21 BROWSE ADDRESS SPACE IN SOFTWARE	35
FIGURE 22 SAMPLE OF DATA SHOWING IN BROWSE OPTION	36
FIGURE 23 BROWSE_ADDRESS_SPACE MODULE	36
FIGURE 24 READ VARIABLE SCREEN	37
FIGURE 25 READ VARIABLE	38
FIGURE 26 READ VARIABLE MODULE.....	38
FIGURE 27 WRITE VARIABLE SCREENSHOT	39
FIGURE 28 VALUE OF BOOLEAN VARIABLE IS WRITTEN TO THE SERVER	40
FIGURE 29 IMPLEMENTATION OF WRITE VARIABLE.....	41
FIGURE 30 SETTINGS FOR SUBSCRIPTION AND MONITORED ITEMS.....	42
FIGURE 31 SUBSCRIBE VARIABLES SCREEN	44
FIGURE 32 STARTING SUBSCRIPTION OF SUBSCRIBE VARIABLE SCREEN	45
FIGURE 33 MONITORING OF A VARIABLE FROM SUBSCRIPTION.....	46
FIGURE 34 IMPLEMENTATION OF PUB-SUB.....	47
FIGURE 35 SUBSCRIPTION START IN MULTIPLE PUB-SUB	48
FIGURE 36 MONITORING OF VARIABLE IN MULTIPLE PUB-SUB NOTIFICATION.....	48
FIGURE 37 IMPLEMENTATION OF MULTIPLE PUB-SUB	49
FIGURE 38 DOWNLOADING OF SUBSCRIBED DATA	50
FIGURE 39 CREATION OF OPCUA SERVER SCREEN.....	52
FIGURE 40 CREATION OF OPCUA SERVER	53
FIGURE 41 ALLOW NETWORK CONNECTION ON THE SPECIFIED PORT AND APPLICATION.....	54
FIGURE 42 SHUTDOWN THE SERVER.....	54
FIGURE 43 SETTINGS SCREEN.....	55
FIGURE 44 HELP SCREEN	56
FIGURE 45 OPTIONS AVAILABLE VIA MENU.....	57
FIGURE 46 EDIT OPTIONS VIA MENU.....	57
FIGURE 47 CONNECTOPC-UA OPTIONS VIA MENU.....	58
FIGURE 48 SETTINGS OPTIONS VIA MENU.....	58
FIGURE 49 METHOD CALL	59
FIGURE 50 METHOD CALL WITH 2 ARGUMENTS AND NO OUTPUT	60
FIGURE 51 PROPERTIES OF NODE-ID SHOWING A NODECLASS METHOD.....	61
FIGURE 52 SHOWING THE DESCRIPTION AND DATATYPE OF INPUT ARGUMENTS	62
FIGURE 53 SHOWING THE DESCRIPTION AND DATATYPE OF OUTPUT ARGUMENT	62

FIGURE 54 METHOD CALL WITH 2 INPUT ARGUMENTS AND 1 OUTPUT ARGUMENT	63
FIGURE 55 BASE CONDITION STATE MODEL	64
FIGURE 56 ACKNOWLEDGEABLECONDITIONS STATE MODEL.....	65
FIGURE 57 EVENT NOTIFIER	65
FIGURE 58 ALARM MONITORING IN THE DEMO SERVER ON THE 'SERVER' NODE	66
FIGURE 59 FILE TRANSFER	67
FIGURE 60 BROWSING FILEREAD NODE ID.....	68
FIGURE 61 READ FILE FROM THE FILEREAD NODE ID.....	68
FIGURE 62 FAVOURITES WINDOW	69
FIGURE 63 MENU FOR FAVOURITES	69
FIGURE 64 ADD TO FAVOURITES OPTION	70
FIGURE 65 ADDING NODE TO THE FAVOURITES LIST	70
FIGURE 66 REJECTION OF DUPLICATE NODE ID IN FAVOURITES.....	70
FIGURE 67 NODE CRAWLER OPTION	71
FIGURE 68 USE OF CONTINUATION POINTS WHILE BROWSING LARGE NODES.....	71
FIGURE 69 HIERARCHY OF NODE CRAWLER VIEW	72
FIGURE 70 DEPLOYMENT FOR LINUX APPLICATION	74
FIGURE 71 DEPLOYMENT FOR MAC OS APPLICATION	75
FIGURE 72 DEPLOYMENT FOR WINDOWS APPLICATION	75
FIGURE 73 DEPLOYMENT OF ELECTRON APP ON MAC OS USING ELECTRON-BUILDER.....	76
FIGURE 74 DEPLOYMENT OF ELECTRON APP IN WINDOWS 10 (64 BIT) USING ELECTRON-BUILDER....	76
FIGURE 75 FILES OF BINARIES IN MAC OS	76
FIGURE 76 INSTALLATION OF SOFTWARE IN MAC.....	77
FIGURE 77 CONNECTOPC-UA APPLICATION IN LAUNCHPAD IN MACOS	77
FIGURE 78 INSTALLATION FILES OF SOFTWARE IN WINDOWS (64 BIT).....	78
FIGURE 79 INSTALLATION OF SOFTWARE IN WINDOWS 10 (64 BIT)	78
FIGURE 80 LISTING OF CONNECTOPC_UA IN ALL APPS OF WINDOWS 10 (64 BIT).....	79
FIGURE 81 CONNECTOPCUA IN CONTROL PANEL	79
FIGURE 82 INSTALLATION FILES OF SOFTWARE IN LINUX (64 BIT ENVIRONMENT).....	80
FIGURE 83 SCREENSHOT OF APPLICATION RUNNING IN LINUX PLATFORM (UBUNTU 14.04 LTS)	80
FIGURE 84 SIEMENS 828D CNC CONTROLLER, DRIVES & MOTORS.....	82
FIGURE 85 FAMILY LINE OF SIEMENS 828D CNC CONTROLLERS	83
FIGURE 86 PPU OF SIEMENS 828D.....	83
FIGURE 87 PPU FEATURES 1.....	84
FIGURE 88 PPU FEATURE2	85
FIGURE 89 SINAMICS DRIVES AND SIMOTICS SERVOMOTORS.....	86
FIGURE 90 SIEMENS 828D CNC MACHINING STANDARDS	87
FIGURE 91 SETTING THE OPCUA OPTION IN SIEMENS 828D.....	87
FIGURE 92 SETUP OPTIONS IN OPCUA DIALOG	88
FIGURE 93 SECURITY MESSAGE FOR OPENING A TCP PORT.....	89
FIGURE 94 BROWSING ADDRESS SPACE IN SIEMENS 828D	91
FIGURE 95 APPLICATION SCENARIO	93
FIGURE 96 ACCESSIBILITY OF THE SERVER	93
FIGURE 97 SINUTRAIN SOFTWARE PACKAGE.....	95
FIGURE 98 SINUTRAIN MAIN SCREEN	97
FIGURE 99 CREATION OF NEW LATHE MACHINE IN SINUTRAIN.....	98
FIGURE 100 MACHINE SCREEN IN SINUTRAIN.....	99
FIGURE 101 RUNNING OF PART PROGRAM IN SINUTRAIN	100
FIGURE 102 LICENSES IN SINUTRAIN.....	101
FIGURE 103 SETTING THE OPTION OPCUA IN SINUTRAIN	102
FIGURE 104 RUNNING OF OPC-UA SERVER IN SINUTRAIN.....	102
FIGURE 105 MACHINE CONFIGURATION	103
FIGURE 106 CONFIGURATION SAVED	104
FIGURE 107 SUCCESSFUL CONNECTION OF THE SOFTWARE WITH SINUTRAIN.....	105
FIGURE 108 BROWSING ADDRESS SPACE OF SINUTRAIN CONFIGURED MACHINE	106
FIGURE 109 PLC INPUT I0.0 STATUS.....	108

FIGURE 110 PLC I0.0 VALUE FROM MACHINE	109
FIGURE 111 MB10 VALUE AS READ BY THE SOFTWARE.....	109
FIGURE 112 STATUS OF MB10 ON THE MACHINE	110
FIGURE 113 READING NUMBER OF ENCODERS NC VARIABLE	110
FIGURE 114 NC-MD 30200 (NUMBER OF ENCODERS).....	111
FIGURE 115 ACTUAL PART NUMBER.....	111
FIGURE 116 FEEDOVERRIDE FROM MACHINE	112
FIGURE 117 SPINDLE OVERRIDE FROM MACHINE	112
FIGURE 118 DRY RUN FEED	113
FIGURE 119 SPINDLE ACTUAL SPEED	113
FIGURE 120 ALL AXES STOPPED SIGNAL.....	114
FIGURE 121 AXES ACCELERATION	114
FIGURE 122 AXIS KV FACTOR.....	115
FIGURE 123 CURRENT ACTIVE PROGRAM NAME.....	115
FIGURE 124 ACTUAL PARTS MACHINED	116
FIGURE 125 ACTUAL POSITION X AXIS.....	116
FIGURE 126 DRIVE READY SIGNAL	117
FIGURE 127 AXIS CURRENT	117
FIGURE 128 VALUE OF R0 PARAMETER IN CHANNEL 1.....	118
FIGURE 129 SELECTED PROGRAM NUMBER	118
FIGURE 130 SINGLE BLOCK ACTIVE STATUS.....	119
FIGURE 131 PERMISSION FOR READING ALL NODES IN SINUMERIK	119
FIGURE 132 GRANTING WRITE PERMISSION TO ALL NODES IN SINUMERIK TO USER 'OPCUACLIENT'	120
FIGURE 133 MONITORING OF SPINDLE OVERRIDE.....	121
FIGURE 134 SPINDLE SPEED VARIATION WRT TO SPINDLE OVERRIDE AND PART PROGRAM	122
FIGURE 135 SPINDLE ACTUAL SPEED VS SPINDLE OVERRIDE.....	123
FIGURE 136 COPYFILEFROMSERVER & COPYFILETOSERVER METHODS	124
FIGURE 137 FILE TRANSFER METHODS TO/FROM SERVER.....	126
FIGURE 138 CONNECTOPC-UA SERVER	128
FIGURE 139 ENDPOINT OF THE SERVER.....	128
FIGURE 140 CONNECTION ESTABLISHED WITH THE SERVER.....	129
FIGURE 141 BROWSING NODES	129
FIGURE 142 FILE READ.....	130

1. Introduction

OPC Unified Architecture (OPCUA) is a machine to machine communication protocol for industrial automation developed by OPC Foundation. It is an open source protocol freely available and implementable under GPL 2.0 License. The OPC UA information modeling framework turns data into information. With complete object-oriented capabilities, even the most complex multi-level structures can be modeled and extended. Data-types and structures are defined in profiles. For example, the existing OPC Classic specifications were modeled into UA profiles which can also be extended by other organizations. Fig-1 shows the detailed architecture of OPC-UA Protocol.

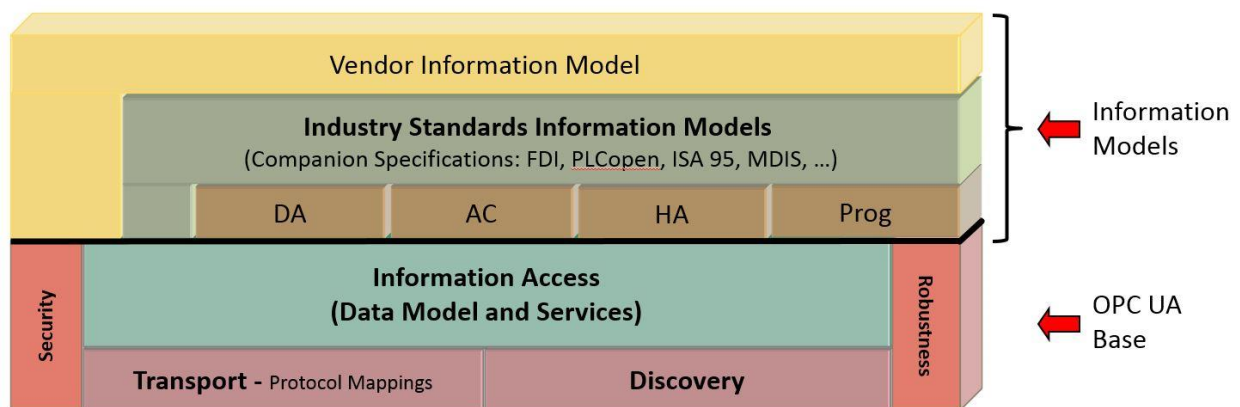


Figure 1 OPC UA Architecture

Details of layers and protocol are not in the scope of this document and are available from <https://opcfoundation.org/about/opc-technologies/opc-ua/> for reference. With the advent of Industry 4.0, OPC-UA become the backbone of interconnected communication in production, warehouse and transport logistics. A central challenge posed by IIoT (Industrial Internet of things) and Industry 4.0 is the secure and standardized exchange of data and information between devices, machines, sensors and services in the Industry. The OPC UA protocol is a powerful data-hub communication protocol because it is able to integrate a wide range of machines, systems and devices via its powerful server-client-based protocol

In November 2016, the Industry 4.0 Platform published a checklist for classifying and advertising products as Industry 4.0 "Basic," "Ready," or "Full". To comply with the "Industry 4.0 communication" criterion, even the lowest category requires the product to be addressable over the network via TCP/UDP or IP and to integrate at least the OPC UA information model. As a result, any product being advertised as "Industry 4.0-enabled" must be OPC UA-capable (either integrated or via a gateway). The checklist also stresses the information modeling property of OPC UA.

With the strong Server-client approach and versatile information model, huge volume data can be fetched from multiple machine, devices and services. Data analysis on these models can be a key for taking critical business decisions which may give a substantial competitive edge over the competitors. With this view, this document describes the implementation of various services provided by the OPC-UA protocol, for fetching the data from the machines (Primarily Siemens and Fanuc CNC Controllers) and devices and how to use the data for further business intelligence.

1.1 Purpose

Machines, devices and services are most valuable resources for any organization. All efforts are made in order to optimize the usage of these resources. For machines, various parameters are monitored for determining the usage patterns like OEE (Overall Equipment Effectiveness), % breakdown, MTTR (Mean Time to Repair), MTBF (Mean Time Between Failure) etc. Quality of data used for the calculation of these parameters is very important and critical for the further decision making. Hence it is very important that the data should be unbiased and not tempered in any way. In addition to the above, Industry 4.0 is the need of hour for all the industries now and as discussed above it cannot be implemented without OPC-UA. The purpose of this document is to implement the OPC-UA protocol for fetching the data from various machines, sensors, devices and services and move toward the implementation of Industry 4.0 compliance. Further this document will explore the possibility of data analysis using R/Python and integrate the data analysis part with the OPC-UA functionality.

1.2 Scope

This document describes the software implementation of OPC-UA Protocol. Few services of OPC-UA protocol are not implemented in the software and are indented for the future revisions. This document will also serve as the user manual of the software 'ConnectOPC-UA', however the details of the OPC-UA protocol are not the scope of this document.

1.3 Definitions, Acronyms and Abbreviations

Following abbreviations are used in this documents :-

OPC UA	OPC Unified Architecture
AS	Address Space
A&E	Alarms and Events
API	Application Programming Interface
DA	Data Access
DCS	Distributed Control System
DX	Data Exchange
HDA	Historical Data Access
HMI	Human Machine Interface
LDAP	Lightweight Directory Access Protocol
MES	Manufacturing Execution System
PLC	Programmable Logic Controllers
SCADA	Supervisory Control and Data Aquisition
SOAP	Simple Object Access Protocol
UDDI	Universal Description, Discovery and Integration
UML	Unified Modelling Language
WSDL	Web Services Definition Language
XML	Extensible Markup Language

In the Scope of this document and as per OPCUA Specifications, following definitions apply.

- i) **AddressSpace** : The collection of information that an OPC UA Server makes visible to its Clients
- ii) **Alarm** : A type of Event associated with a state condition that typically requires acknowledgement

- iii) **Attribute** : A primitive characteristic of a Node. All Attributes are defined by OPC UA, and may not be defined by Clients or Servers. Attributes are the only elements in the AddressSpace permitted to have data values.
- iv) **Certificate** : A digitally signed data structure that describes capabilities of a Client or Server.
- v) **Client** : A software application that sends Messages to OPC UA Servers conforming to the Services specified in this set of specifications.
- vi) **Condition** : A generic term that is an extension to an Event. A Condition represents the conditions of a system or one of its components and always exists in some state.
- vii) **Communication Stack** : A layered set of software modules between the application and the hardware that provides various functions to encode, encrypt and format a Message for sending, and to decode, decrypt and unpack a Message that was received.
- viii) **Complex Data** : Data that is composed of elements or more than one primitive data type, such as a structure.
- ix) **Discovery** : The process by which OPC UA Client obtain information about OPC UA Servers, including endpoint and security information.
- x) **Event** : A generic term used to describe an occurrence of some significance within a system or system component.
- xi) **Event Notifier** : A special Attribute of a Node that signifies that a Client may subscribe to that particular Node to receive Notifications of Event occurrences.
- xii) **Information Model** : An organizational framework that defines, characterizes and relates information resources of a given system or set of systems. The core address space model supports the representation of Information Models in the AddressSpace.
- xiii) **Message** : The data unit conveyed between Client and Server that represents a specific Service request or response.
- xiv) **Method** : A callable software function that is a component of an Object.
- xv) **MonitoredItem** : A Client-defined entity in the Server used to monitor Attributes or Event Notifiers for new values or Event occurrences and generate Notifications for them.
- xvi) **Node(OPCUA)** : The fundamental component of an AddressSpace
- xvii) **NodeClass** : The class of a Node in an AddressSpace. Node Classes define the metadata for the components of the OPC UA Object Model. They also define constructs, such as Views, that are used to organize the AddressSpace.
- xviii) **Notification** : The generic term for data that announces the detection of an Event or of a changed Attribute value. Notifications are sent in NotificationMessages.
- xix) **NotificationMessage** : A Message published from a Subscription that contains one or more Notifications.
- xx) **Object** : A Node that represents a physical or abstract element of a system. Objects are modeled using the OPC UA Object Model. Systems, subsystems and devices are examples of Objects. An Object may be defined as an instance of an ObjectType.
- xxi) **ObjectInstance** : A synonym for Object. Not all Objects are defined by ObjectTypes.
- xxii) **ObjectType** : A Node that represents the type definition for an Object.
- xxiii) **Profile** : A specific set of capabilities, to which a Server may claim conformance. Each Server may claim conformance to more than one Profile.
- xxiv) **Program** : An executable Object that, when invoked, immediately returns a response to indicate that execution has started, and then returns intermediate and final results through Subscriptions identified by the Client during invocation.
- xxv) **Reference** : An explicit relationship (a named pointer) from one Node to another. The Node that contains the Reference is the source Node, and the referenced Node is the target Node. All References are defined by ReferenceTypes.

- xxvi) **ReferenceType** : A Node that represents the type definition of a Reference. The ReferenceType specifies the semantics of a Reference. The name of a ReferenceType identifies how source Nodes are related to target Nodes and generally reflects an operation between the two, such as "A Contains B".
- xxvii) **RootNode** : The beginning or top Node of a hierarchy.
- xxviii) **Server** : A software application that implements and exposes the Services specified in this set of specifications.
- xxix) **Service** : A Client-callable operation in an OPC UA Server. A Service is similar to a method call in a programming language or an operation in a Web services WSDL contract.
- xxx) **Service Set** : A group of related Services
- xxxi) **Session** : A logical long-running connection between a Client and a Server. A Session maintains state information between Service calls from the Client to the Server.
- xxxii) **Subscription** : A Client-defined endpoint in the Server, used to return Notifications to the Client. Generic term that describes a set of Nodes selected by the Client (1) that the Server periodically monitors for the existence of some condition, and (2) for which the Server sends Notifications to the Client when the condition is detected.
- xxxiii) **Variable** : A Variable is a Node that contains a value.
- xxxiv) **View** : A specific subset of the AddressSpace that is of interest to the Client.

The following is a list of definitions for this template based on the UML approach to system design:

Class Diagram	Describes the structure of a system
Object Diagram	Expresses possible object combinations of a specific Class Diagram
State chart Diagram	Expresses possible states of a class (or a system)
Activity Diagram	Describes activities and actions taking place in a system
Sequence Diagram	Shows one or several sequences of messages sent among a set of objects
Collaboration Diagram	Describes a complete collaboration among a set of objects
Use-case Diagrams	Illustrates the relationships between use cases
Component Diagram	A special case of a Class Diagram used to describe components within a software system
Deployment Diagram	A special case of a Class Diagram used to describe hardware within the overall system architecture
System Block diagram	A diagram showing the major components of the system with its interconnections and external interfaces

1.4 Intended Audience

The intended audience for this document is engineers/researchers who are interested in Industry 4.0 / OPC-UA. As the software is implemented using Node.js and Electron.js, the reader should have a working knowledge of JavaScript, Node.js. This document also provides a diagnostic tool for the machines is useful for those readers who are looking after the maintenance of CNC/PLC based machines.

1.5 List of Components/Modules Used

Various Libraries and modules are used for the designing of this software. Few important ones are listed below. Please ensure, different versions of libraries may have compatibility issues and may have to resolved separately.

Node.js	v10.16.0 LTS
npm	v6.7.4
Electron.js	v5.0.8
Async	v3.1.0
NodeOPC-UA	v0.7.4
Electron Builder	v21.1.5

1.6 References

Following are the references for the modules that are primarily used in this project

Node.js	https://nodejs.org/en/
npm	https://www.npmjs.com/
Electron.js	https://electronjs.org/releases/stable
Async	https://www.npmjs.com/package/async
NodeOPC-UA	https://node-opcua.github.io/
Electron Builder	https://www.electron.build/
OPC-UA Protocol	https://opcfoundation.org/about/opc-technologies/opc-ua/
JSON2XLS	https://www.npmjs.com/package/json2xls
Path	https://www.npmjs.com/package/path

1.7 Content Summary

Chapter-1 deals with the introduction of the projects, objective and scope of the project and this documents, it provides the references of the modules used in the project. **Chapter-2** describes the system overview, what are the alternatives present for the application development, which one is the best alternative, what is the system architecture and infrastructure used in the development of the software. **Chapter-3** deals with the system requirements for the development of the software, it also describes how to setup the environment for the developemnt of the software along system block diagram, sequence diagram and activity chart diagram. **Chapter-4** deals with the description of modules implemented in this software. **Chapter-5** deals with the deployment of application in Windows, Linux and Mac OS Platform. **Chapter-6** deals with the Case study of using this software for monitoring the CNC Machines equipped with Siemen make Sinumerik 828D CNC Controller.**Chapter-7** deals with another case study of using this software for the in-build OPCUA Server. **Chapter-8** deals with the Source-code of the software

2. System Overview

OPC-UA is a platform-independent standard through which various kinds of systems and devices can communicate by sending request and response Messages between Clients and Servers or Network Messages between Publishers and Subscribers over various types of networks. It supports robust, secure communication that assures the identity of OPC UA Applications and resists attacks.

In the Client Server model OPC UA defines sets of Services that Servers may provide, and individual Servers specify to Clients what Service sets they support. Information is conveyed using OPC UA-defined and vendor-defined data types and Servers define object models that Clients can dynamically discover. Servers can provide access to both current and historical data, as well as Alarms and Events to notify Clients of important changes. OPC UA can be mapped onto a variety of communication protocols and data can be encoded in various ways to trade off portability and efficiency.

In addition to the Client Server model, OPC UA defines a mechanism for Publishers to transfer the information to Subscribers using the PubSub model.

OPC UA provides a consistent, integrated AddressSpace and service model. This allows a single Server to integrate data, Alarms and Events, and history into its AddressSpace, and to provide access to them using an integrated set of Services. These Services also include an integrated security model.

OPC UA also allows Servers to provide Clients with type definitions for the Objects accessed from the AddressSpace. This allows information models to be used to describe the contents of the AddressSpace. OPC UA allows data to be exposed in many different formats, including binary structures and XML or JSON documents. The format of the data may be defined by OPC, other standard organizations or vendors. Through the AddressSpace, Clients can query the Server for the metadata that describes the format for the data. In many cases, Clients with no pre-programmed knowledge of the data formats will be able to determine the formats at runtime and properly utilize the data.

OPC UA adds support for many relationships between Nodes instead of being limited to just a single hierarchy. In this way, a Server may present data in a variety of hierarchies tailored to the way a set of Clients would typically like to view the data. This flexibility, combined with support for type definitions, makes OPC UA applicable to a wide array of problem domains. As illustrated in **Figure 2**, OPC UA is not targeted at just the SCADA, PLC and DCS interface, but also as a way to provide greater interoperability between higher level functions.

In order to access data from the machines, it is necessary to implement OPCUA Client and Pub-Sub service. There are 2 ways to achieve it, one is to write the complete stack for it and the second way is to use the available stack and build the system over the stack. We decided to use open source NodeOPC-UA stack. NodeOPC-UA is an OPC UA stack fully written in JavaScript and NodeJS. NodeOPC-UA takes advantage of the asynchronous nature of node.js, creating highly responsive applications.

Node.js is based on javascript and is able to run on web servers easily, in order to have platform independent desktop applications Electron.js module is used. Electron is a framework for creating native applications with web technologies like JavaScript, HTML, and CSS. It is an open source project maintained by GitHub and is able to generate cross platform (i.e. MAC, Windwos 32/64 bit, and Linux) native applications.

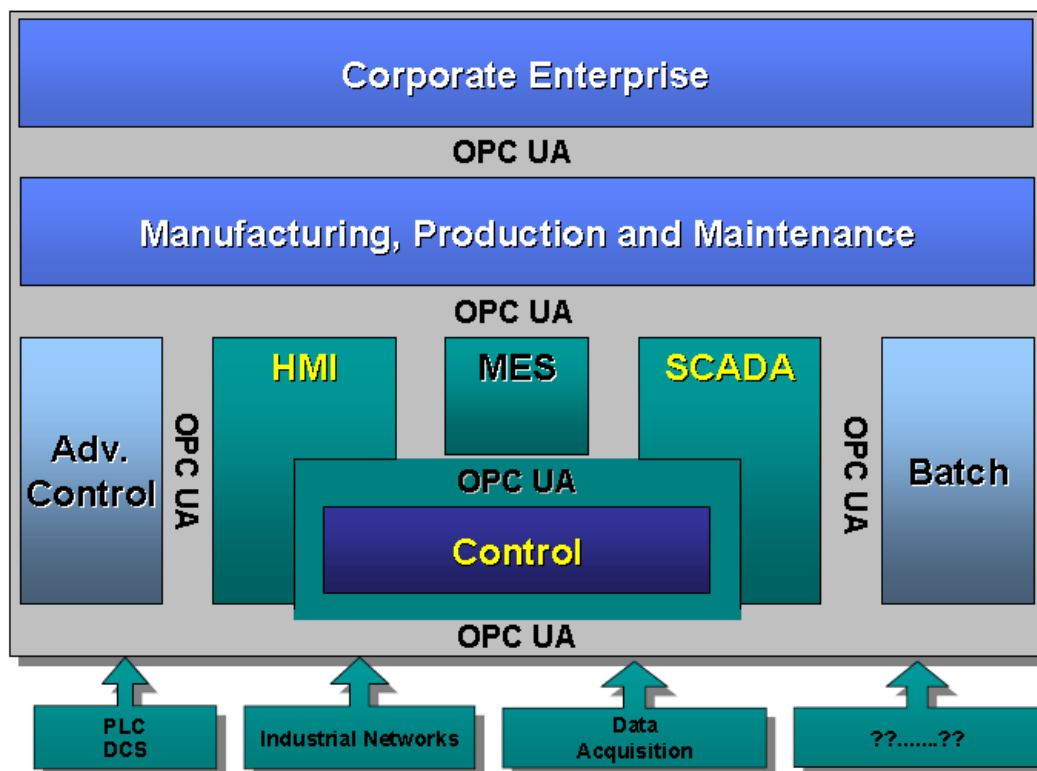


Figure 2 OPCUA Target Applications

2.1 System Architecture

Software is designed on client-server architecture and Pub-Sub model. OPC UA is designed to provide robustness of published data. A major feature of all OPC servers is the ability to publish data and Event Notifications. OPC UA provides mechanisms for Clients to quickly detect and recover from communication failures associated with these transfers without having to wait for long timeouts provided by the underlying protocols.

OPC UA is designed to support a wide range of Servers, from plant floor PLCs to enterprise Servers. These Servers are characterized by a broad scope of size, performance, execution platforms and functional capabilities. Therefore, OPC UA defines a comprehensive set of capabilities, and Servers may implement a subset of these capabilities. To promote interoperability, OPC UA defines subsets, referred to as Profiles, to which Servers may claim conformance. Clients can then discover the Profiles of a Server, and tailor their interactions with that Server based on the Profiles.

ConnectOPC-UA Software is divided into 7 main modules which implements various services provided by the OPCUA.

Sno.	Modules	Description
01	Server Discovery, Connection and session establishment between client and server	OPCUA provides a security model, which can be selected and configured to meet the security need of a given installation. Application level security relies on a secure communication channel that is active for the duration of the application Session and ensures the integrity of all Messages that are exchanged. This means users need to

		be authenticated only once, when the application Session is established. When a Session is established, the Client and Server applications negotiate a secure communications channel. Digital (X.509) Certificates are utilized to identify the Client and Server.
02	Browsing Address Space	The set of Objects and related information that the Server makes available to Clients is referred to as its AddressSpace. The OPC UA AddressSpace represents its contents as a set of Nodes connected by References. To promote interoperability of Clients and Servers, the OPC UA AddressSpace is structured hierarchically with the top levels the same for all Servers. Although Nodes in the AddressSpace are typically accessible via the hierarchy, they may have References to each other, allowing the AddressSpace to represent an interrelated network of Nodes.
03	Read Variables from Server	The COM UA Proxy converts all the ItemIds in the Read into valid NodeIds by replacing the '-' with '=' and calls the OPC UA Read Service for the Value Attribute. If the Read Service call is successful then DataValue for each node is mapped to the VQT for each item If the Read Service call fails or If there are errors for some of the Nodes, then the StatusCodes of these Nodes are mapped to the error code by the COM UA Proxy.
04	Write Variables to Server	The COM UA Proxy converts all the ItemIds in the Write into valid NodeIds by replacing the '-' with '='. It converts the Value, Quality and Timestamp (VQT) to a DataValue structure and calls the OPC UA Write Service for the Value Attribute. If the Write Service call fails or if there are errors for some of the Nodes, then the Status Codes of these Nodes are mapped to the error code by the COM UA Proxy.
05	Subscription (Pub-Sub Model)	The Subscription Service Set is used by the Client to create and maintain Subscriptions. Subscriptions are entities that periodically publish Notification Messages for the Monitored Item assigned to them (see 7.9). The Notification Message contains a common header followed by a series of Notifications. The format of Notifications is specific to the type of item being monitored (i.e. Variables, Attributes, and Event Notifiers). Once created, the existence of a Subscription is independent of the Client's Session with the Server. This allows one Client to create a Subscription, and a second, possibly a redundant Client, to receive Notification Messages from it. To protect against non-use by Clients, Subscriptions have a configured lifetime that Clients periodically renew. If any Client fails to renew the lifetime, the lifetime expires and the Subscription is closed by the Server. When a Subscription is closed, all Monitored Items assigned to the Subscription are deleted.

06	Subscriptions with large no of variables(Pub-Sub Model)	This modules supports all the features mentioned in module-5, the only difference is module-5 supports monitoring of 5 variables only at a time, while in this module you can monitor maximum number of variables that is set by the connected server
07	Creation of a Simple OPCUA Server	Here a simple OPCUA Server with a fixed Address Space is created for the diagnostics of the software and the client part. It should be used for the training and diagnostic purposes only. Full Functional OPCUA Server with variable Address Space is out of scope for the current version of software and may be released in later versions.
08	Settings and Help	This section deals with various parameters required for Module-1 to Module-07. So depending upon the type of OPCUA Server and the type of service appropriate parameter settings should be set in this section. Apart from settings it also provide the help sections which describe the software version, designer details and modules of OPC-UA protocol implemented in this software.

2.2 Infrastructure Services

In order to build secure connection between client and server, both anonymous and authentication using username and password are provided. In addition to above additional security options are provided through the software where every data exchange is encrypted using Basic256 and Basic128RSA15. Every module of the software is equipped with its error handling procedures which present the detailed info to the user about the source of error. Various settings are provided to control the lifecycle of the session and the subscription services, changing these settings will change the behavior of the application. OPCUA Server is included as a diagnostics tool for the application, although it is fully functional and importing of address space while configuring the server will be included in the future update of this application.

3. System Design

Electron.js and Node.js are the building blocks for this app development. In order to implement OPC-UA services Node.js module NodeOPC-UA has been used.

Node.js is asynchronous event driven JavaScript Runtime. It is designed to build scalable network applications. This is in contrast to today's more common concurrency model where OS threads are employed. Thread-based networking is relatively inefficient and very difficult to use. Furthermore, users of Node are free from worries of dead-locking the process, since there are no locks. Almost no function in Node directly performs I/O, so the process never blocks. Because nothing blocks, scalable systems are very reasonable to develop in Node.

Electron.js is an open source library developed by GitHub for building cross-platform desktop applications with HTML, CSS, and JavaScript. Electron accomplishes this by combining Chromium and Node.js into a single runtime and apps can be packaged for Mac, Windows, and Linux.

In electron.js there are 2 process types Main process and Renderer Process.

Main Process : The process that runs package.json's main script is called the main process. The script that runs in the main process can display a GUI by creating web pages. An Electron app always has one main process, but never more.

Renderer Process : Since Electron uses Chromium for displaying web pages, Chromium's multi-process architecture is also used. Each web page in Electron runs in its own process, which is called the renderer process.

The main process creates web pages by creating **BrowserWindow** instances. Each **BrowserWindow** instance runs the web page in its own renderer process. When a **BrowserWindow** instance is destroyed, the corresponding renderer process is also terminated.

The main process manages all web pages and their corresponding renderer processes. Each renderer process is isolated and only cares about the web page running in it.

In web pages, calling native GUI related APIs is not allowed because managing native GUI resources in web pages is very dangerous and it is easy to leak resources. If you want to perform GUI operations in a web page, the renderer process of the web page must communicate with the main process to request that the main process perform those operations.

3.1 System Requirements

Electron.js and node.js supports multiple OS platform and require a very small RAM footprints. Due to the Chromium minimum requirements, it could not be run in older Windows and Mac OS. Full System requirement list is provided below.

OS	Min RAM Required	Processor	OS Architecture	Electron.js/Node.js Support
Windows 7 & Above	512 MB	Intel Pentium 4 processor or later that's SSE2 capable	x86 & x64	Supported
Windows XP and Olders Versions	-	-	-	Not Supported
MacOS 10.10	512 MB	Intel 64 bit processor	x64	Supported

(Yosemite).				
MacOs older than Yosemite	-	-	-	Not Supported
Ubuntu 12.04/12.04LTS	512MB	An Intel Pentium 4 processor or later that's SSE2 capable	ia32/x64	Supported
Fedora 21				
Debian 8				
Other linux distribution with ARM processors (Raspbian with Raspberry Pi3 only)	512 MB	ARM Processor	ARM v7 with hard-float ABI and NEON for Debian Wheezy	Supported

3.2 Setting up development environment

Before start writing the code, proper environment must be set. Following are the steps need to be executed for setting up the development environment.

3.2.1 Installing Node Js

Node.js can be installed for multiple platforms from <https://nodejs.org/en/download/>. For windows platform .exe x86/x64 file must be selected depending upon the 32/64 bit installed Windows OS version. For Mac OS .pkg/.tar.gz file must be selected. In Linux (for Intel Processors), Nodejs can be installed from the source or can be installed from Linux Binaries). Nodejs Binaries is also available for ARM v6/v7/v8 processors.

3.2.2 Installing NPM

npm (Node Package Manager) is distributed with Node.js, which means that when you download Node.js, you automatically get **npm** installed on your computer.

To check if you have Node.js installed, run this command in your terminal:

```
node -v
```

To confirm that you have npm installed you can run this command in your terminal:

```
npm -v
```

npm is a separate project from Node.js, and tends to update more frequently. As a result, even if you've just downloaded Node.js (and therefore npm), you'll probably need to update your npm. Luckily, npm knows how to update itself! To update your npm, type this into your terminal:

```
npm install npm@latest -g
```

3.2.3 Installing Node-OPCUA

Node-opcua package can be installed from the Node Package Manager (npm) by the following command

```
$ npm install node-opcua
```

Installing node-opcua from source

running the demo server from source

```
$ git clone https://github.com/node-opcua/node-opcua.git
$ cd node-opcua
$ npm install
$ node packages/node-opcua-samples/bin/simple_server
```

running the demo client from source

```
$ git clone https://github.com/node-opcua/node-opcua.git
$ cd node-opcua
$ npm install
$ node packages/node-opcua-samples/bin/simple_client.js -e
"opc.tcp://opcserver.mAutomation.net:4841" -n="ns=1;s=EVR2.sys"
```

3.2.4 Installing Async

Async is a utility module which provides straight-forward, powerful functions for working with asynchronous JavaScript. Although originally designed for use with Node.js and installable via npm and it can directly be used in the browser also

```
npm install async
```

Async is also installable via:

```
yarn: yarn add async
bower: bower install async
```

Async provides around 70 functions that include the usual 'functional' suspects (map, reduce, filter, each...) as well as some common patterns for asynchronous control flow (parallel, series, waterfall...). All these functions assume you follow the Node.js convention of providing a single callback as the last argument of your asynchronous function -- a callback which expects an Error as its first argument -- and calling the callback once.

3.2.5 Installing json2xls

json2xls is the utility to convert json to a excel file, based on Node-Excel-Export, it can be installed via a npm

```
npm install json2xls
```

3.3 System Components

As discussed in chapter-2, software is divided into 10 modules as shown below. Each module has a web page and its associated rendered process which is communicating with the Electron's main application process (**main.js**) via IPC Communication model. Electron provides 2 IPC (Inter Process Communication) modules called **ipcMain** and **ipcRenderer**.

SNo.	Module Name	Web page	Symbolic representation
i)	Configure machine	index.html	M1
ii)	Connect machine	connect.html	M2
iii)	Browse Address Space	browse.html	M3
iv)	Monitor/Read Variables	monitor.html	M4
v)	Write Variables to the server	write.html	M5
vi)	Create Pub-sub Notifications	pub_sub.html	M6
vii)	Multiple Pub-sub Notofications	pubsub_mult.html	M7
viii)	OPCUA Server	server.html	M8
ix)	Settings	settings.html	M9
x)	Help	help.html	M10

The **ipcMain** module is used to communicate asynchronously from the main process to renderer processes. When used in the main process, the module handles asynchronous and synchronous messages sent from a renderer process (web page). The messages sent from a renderer will be emitted to this module.

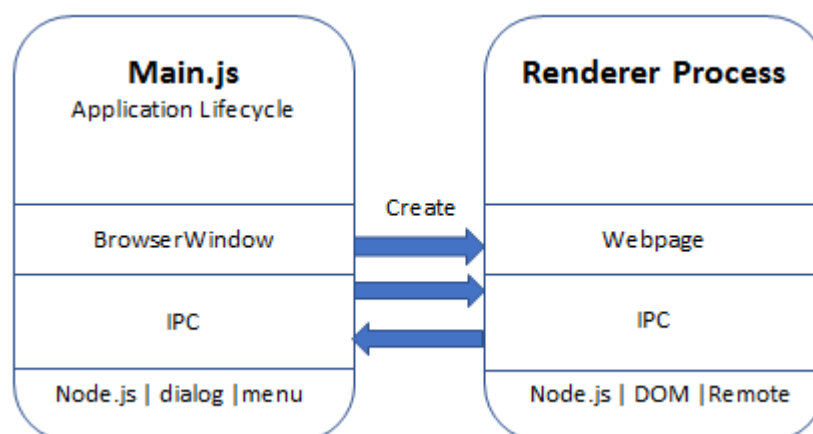


Figure 3 App Development with Electron JS

The **ipcRenderer** module is used to communicate asynchronously from a renderer process to the main process. It provides a few methods so you can send synchronous and asynchronous messages from the renderer process (web page) to the main process. You can also receive replies from the main process.

It is clear from the above component diagram that some processes are independent of other processes like Process help.html and index.html are independent of all other processes while browse.html, monitor.html, write.html, pub_sub.html, pub_sub_mult.html is depending on connect.html process and connect.html process is itself dependent on index.html. This means without the error free and successful completion of connect.html, none of the connected processes can execute successfully.

In order to connect a Device/Machine, machine must be configured. OPC Url address of the machine/device must be provided along with the port number. Depending on the security profiles of the machine/device proper settings and authentication must be provided while configuring the machine option. After the success configuring of the machine, next step is to connect the machine.

If machine/device is connected successfully, then the address space provided by the OPCUA protocol can be accessed using browse option. The value of any node available in address space can be read/write. Using Pub-Sub model any variable can be monitored for the value change. Activity diagram of the software is shown in figure-4, it is evident from the diagram that in case of any error, machine will be disconnected and the secure session between the client and the server will become invalid, hence machine/device should be reconnected using Connect machine activity.

Download service is provided for downloading the address space and monitored variable values. Please remember the variables subscribed will be monitored till the subscription is active, as soon as subscription ends either due to the stop command or due to invalid session, monitoring of values will also stop, although all the historical data remain in the memory and can be downloaded from the download service. Currently data is download in Excel format only which can be viewed using Microsoft Excel or any compatible software that supports viewing of .xls/.xlsx file. Downloading of data in .pdf format will be made available in future version of this software. One advantage of having the data in excel format is that the downloaded data can be directly accessed by R,python,SAS statistical softwares for the further data analysis.

Few options can be accessed without configuring and connecting the machine/Device like OPCUA Server can be created and another instance of this software or any other client can be connected to the OPCUAServer this software has created and access the address space of this software. Similarly help and settings can also be accessed. Help option will show the current version of the software, contact details of the designer and the options available with the current version of the software. Settings provides the option to control the lifecycle of the connection, server settings, subscription and read/write settings and misc settings. Please remember any changes in the settings will restart the application by itself, and all the settings will get activated from this instance.

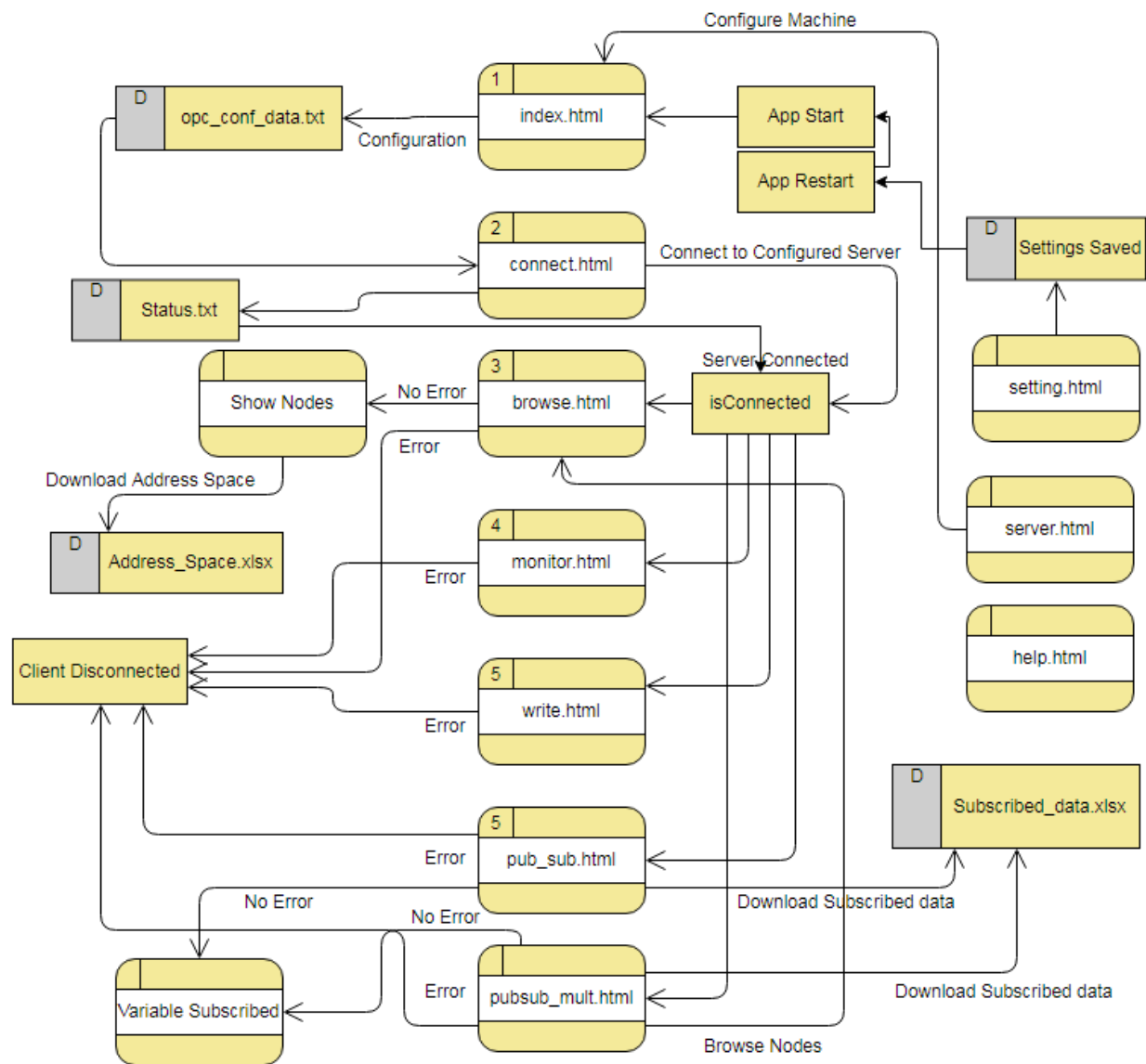


Figure 4 Component Diagram

It must be carefully noted that for every new instance of the software(i.e. every time software starts) machine has to be connected using 'connect machine' option. However user need not to reconfigure the machine every time and last machine/device configuration done by the user remain valid till it is again changed by the user. If the user is connecting to the different machine with different OPC URL address and different Server Profile, then new configuration is again required.

Total number of signals that can be subscribed is limited by the server, any request of accessing the node value beyond the limit of maximum number of simultaneous subscribed connection, error Bad – 'Monitored Item Id Invalid' will be generated. Subscribing variables for monitoring needs server resources which may affect the core functionalities of the machine, hence the simultaneous subscription should be strictly limited.

Error handling and logging is carefully done at each modules, so that the user can understand the issues and could resolve the problems. Input validation in the user form is present with suitable error message to avoid errors due to typographical mistakes.

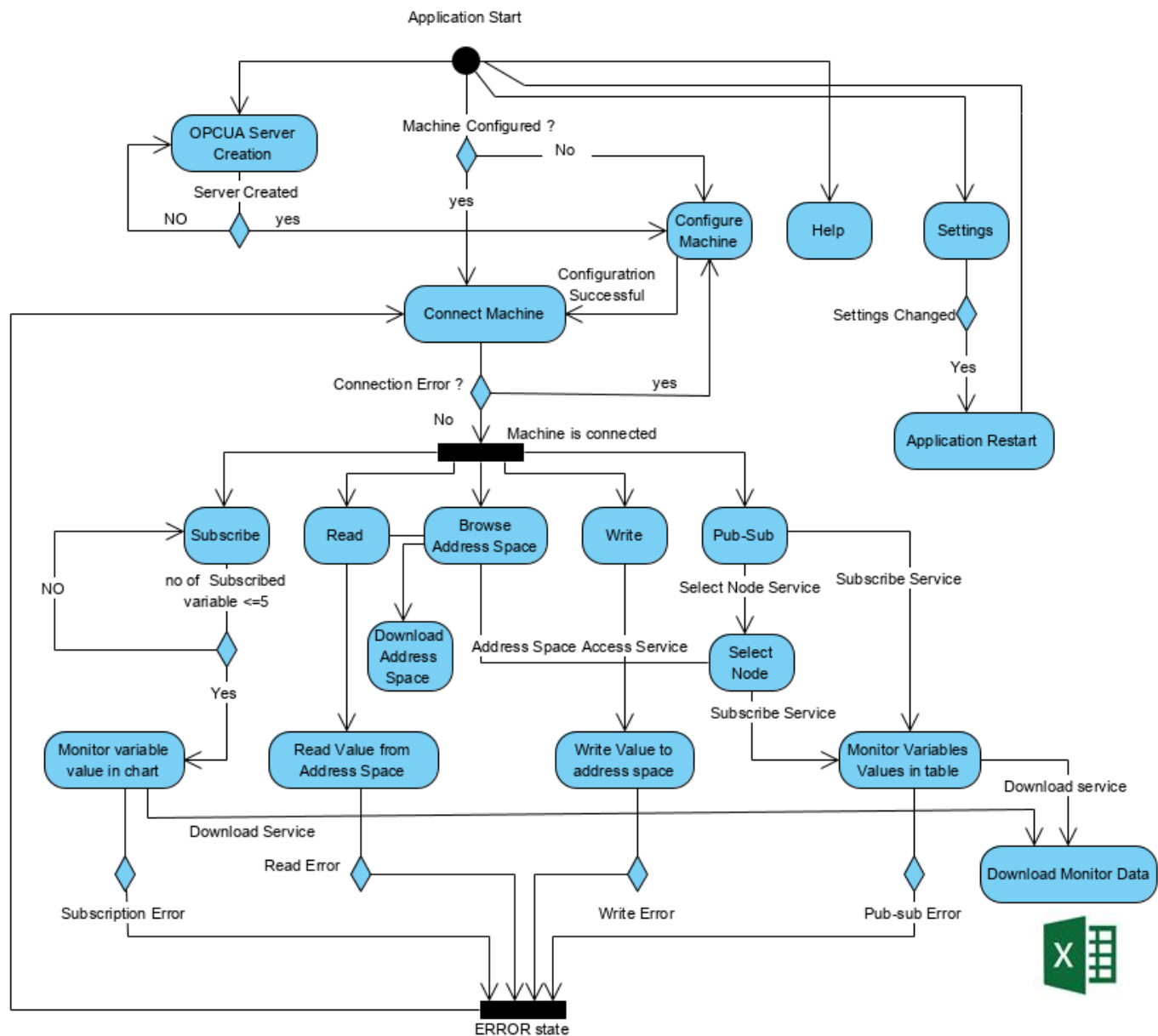


Figure 5 Activity Diagram

4. Modules

In chapter-3, the development of the software is discussed along with the system requirement, dependencies and the components. This chapter deals with the details of all the components of the software as discussed in Chapter-3. This chapter actually deals with the deployment of all the modules in the software and their interconnections with each other as shown in activity diagram (Figure-5)

4.1 Configure Machine

As discussed in chapter-2 and chapter-3, in order to connect to a machine/device proper configuration needs to be done. It is necessary that the server profiles and proper authentication (Username and password) must be known in prior of using this option. Following are the screenshots from the software that need to be filled properly in order to connect with the device/machine successfully.

The OPC UA security architecture is a generic solution that allows implementation of the required security features at various places in the OPC UA Application architecture. Depending on the different mappings described in Part 6, the security objectives are addressed at different levels. The OPC UA security architecture, for Client / Server communication is structured in an Application Layer and a Communication Layer atop the Transport Layer as shown in **Figure 6**.

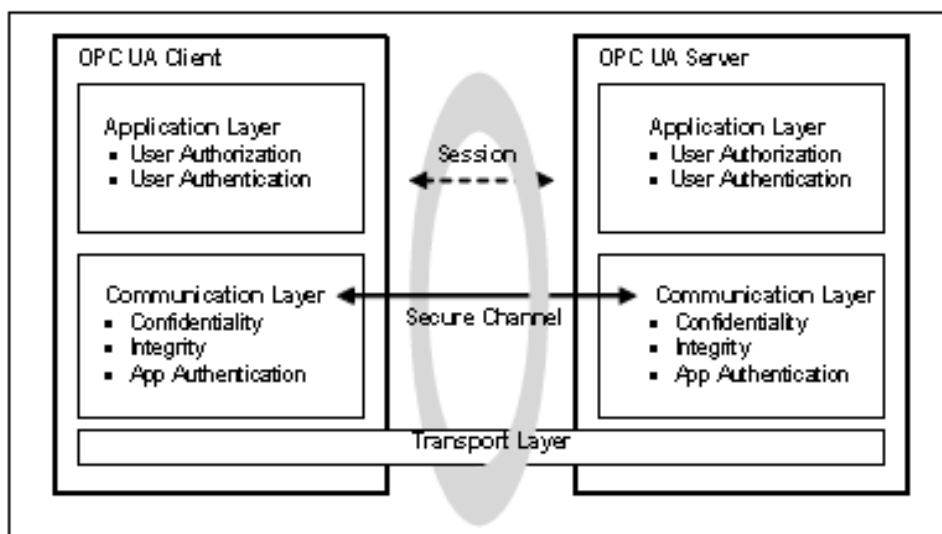


Figure 6 OPC-UA Security Architecture Client-Server

Machine can be configured with no authentication i.e. Anonymous connection to the server, provided server supports anonymous connections as shown in Figure 7.

Figure 7 Configuration setup mode1

As per OPCUA Protocol User Authentication is achieved when the Client passes user credentials to the Server as specified via Session Services. The Server can authenticate the user with these credentials. The owner (user) of a Session can be changed using the ActivateSession Service in order to meet needs of the application. User Authentication is not directly part of the Publish-Subscribe communication pattern but is used as part of the SKS associated with this communication pattern. Sample of configuration required for this type of machines/devices are shown in Fig-9.

OPC UA provides user authorization based on the authenticated user. OPC UA Applications may determine in their own way what data is accessible and what operations are authorized or they may use Roles. Profiles exist to indicate the support of user credentials to restrict or control access to the address space.

OPC UA provides standard approach for implementing role based security. Servers may choose to implement none, part or all of mechanisms defined in Part 5 of OPCUA Protocol. The OPC UA approach assigns Permissions to Roles. Clients are then granted Roles based on connection information. Roles might be restricted by User Authentication, Application Authentication, Security Modes, or Transports. The assignment of Roles and restrictions is application specific.

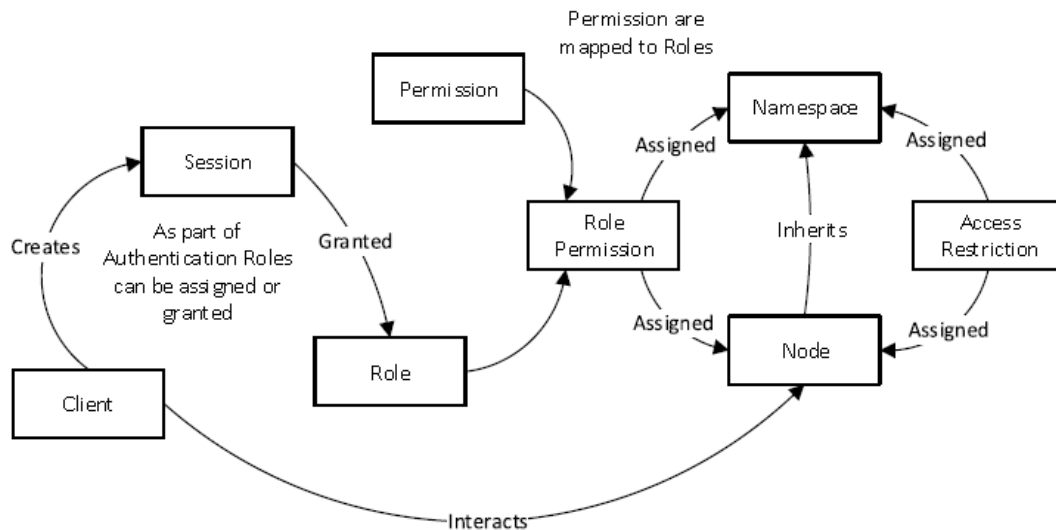


Figure 8 Roles in OPCUA

The screenshot shows the ConnectOPC-UA application window. The title bar includes an Apple logo, the text 'connectopc_ua', and menu items 'Edit', 'Options', and 'Settings'. The main window has a sidebar on the left with a list of options: 'Configure Machine' (highlighted in red), 'Connect Machine', 'Browse Address Space', 'Monitor variables', 'Write variables to the server', 'Create Pub-Sub/Notifications', 'Multiple Pub-Sub/Notifications', 'Create OPC-UA Server', 'Settings', and 'Help'. The main area is titled 'Configure Machine' and contains a form with the following fields:

Machine Name	OPC Machine
Machine Number	123
Machine Location	BLK2BAY1
OPC URL ADDRESS	opc.tcp://localhost:4840/opcua
Anonymous Connection	☐
Username	cnc
Password	
Security Policy	None
Message Security Mode	None

At the bottom of the form is a green 'Save' button.

Figure 9 Configuration setup mode2

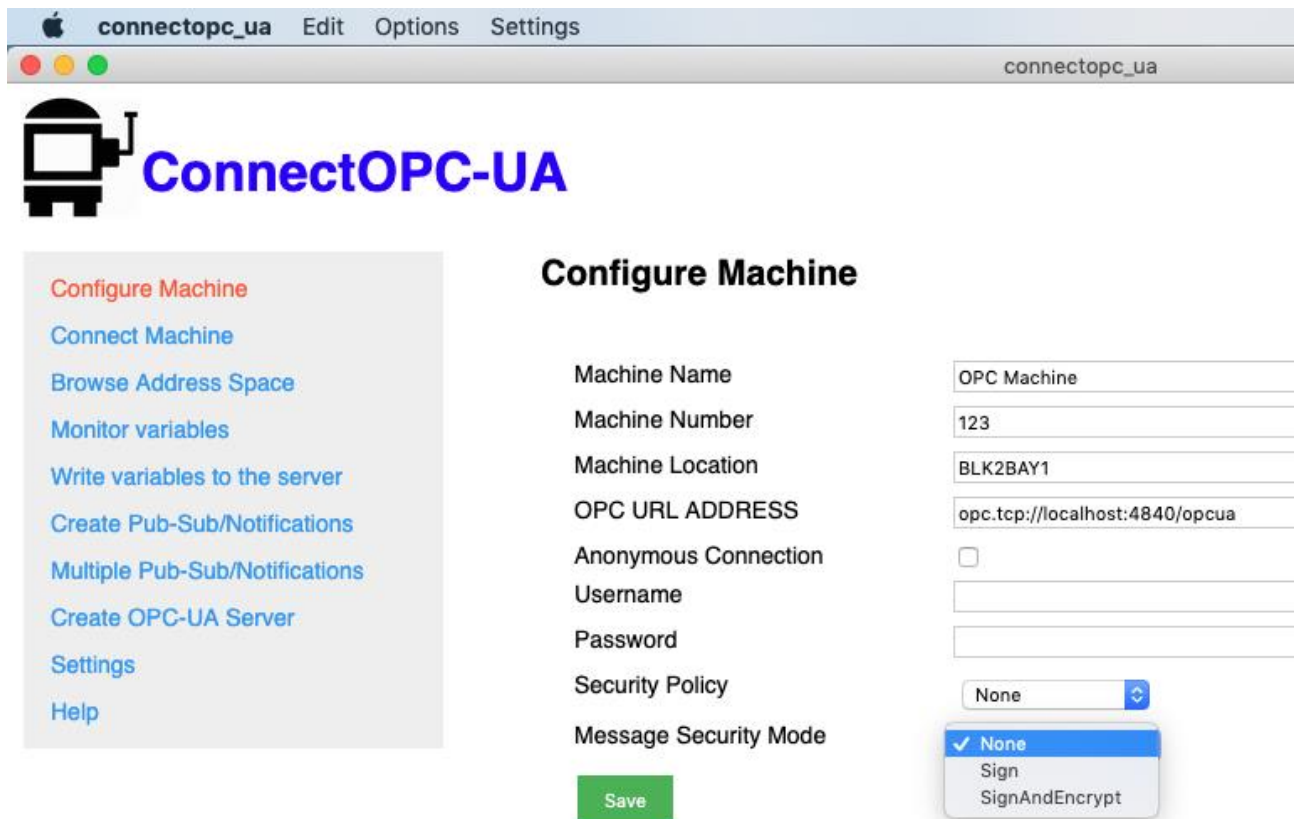


Figure 10 Configuration Setup mode3

The OPC UA Security Services are a group of abstract service definitions specified in Part 4 of OPCUA Specifications that are used for applying various security mechanisms to communication between OPC UA Clients and Servers.

The Discovery Service Set (specified in Part 4 of OPCUA Specifications) defines services used by an OPC UA Client to obtain information about the security policies (see 4.6) and the Certificates of specific OPC UA Servers.

The services of the Secure Channel Service Set (specified in Part 4 of OPCUA Specifications) are used to establish a Secure Channel which is responsible for securing Messages sent between a Client and a Server. The challenge of the Secure Channel establishment is that it requires the Client and the Server to securely exchange cryptographic keys and secret information in an insecure environment, therefore a specific Key Exchange Algorithm (similar to SSL Handshake protocol defined in SSL/TLS) is applied by the communication participants.

The OPC UA Client retrieves the security policies and Certificates of the OPC UA Server by the above mentioned discovery services. These Certificates contain the Public Keys of the OPC UA Server.

The OPC UA Client sends its Public Key in a Certificate and secret information with the OpenSecureChannel service Message to the Server. This Message is secured by applying Asymmetric Encryption with the Server's Public Key and by generating Asymmetric Signatures with the Client's Private Key. However, the Certificate is sent unencrypted so that the receiver can use it to verify the Asymmetric Signature.

The Server decrypts the Message with its Private Key and verifies the Asymmetric Signature with the Client's Public Key. The secret information of the OPC UA Client together with the secret information of the OPC UA Server is used to derive a set of cryptographic keys that are

used for securing all further Messages. Furthermore, all other service Messages are secured with Symmetric Encryption and Symmetric Signatures instead of the asymmetric equivalents. The Server sends its secret information in the service response to the Client so that the Client can derive the same set of cryptographic keys.

Since Clients and Servers have the same set of cryptographic keys they can communicate securely with each other.

These derived cryptographic keys are changed periodically so that attackers do not have unlimited time and unrestricted sequences of Messages to use to determine what the keys are.

For PubSub communications, the security related definitions are specified in Part 14 and provide a description of how to secure messages and also how to obtain the security keys required for message security.

The Publisher will utilize the keys provided to secure the message. It will encrypt the body of the message and sign the entire message. Subscribers will utilize the keys to decrypt and verify the signature of the messages.

To obtain the required keys, the Publisher or Subscriber make use of Client – Server communication. The keys may also be obtained using session-less method calls.

Fig-10 and Fig-11 shows the configurable options for security policy and message security mode

In the 'Configure Machine' option OPC URL address field cannot be left blank and is mandatory field. As per OPCUA Specifications Address must start with "opc.tcp://" followed by the ip address or name of the server and then port number must be specified followed by the resource name (if any). It must be noted that firewall must allow network connection on the specified port number else connection could not be established successfully. As shown in Fig-12 after the necessary details entered into the screen and submitting it a confirmation box will be opened suggesting now to use 'connect machine' option

Fig-13 shows the block diagram of the module implemented in Electron App. Please note the convention that all the data available from the IPC Communication are having a text color of Red. Any action performed in javascript/Node.js are denoted in blue color. Functions or modules under implementation are shown inside the yellow boxes. The GUI presented to the user is shown inside the grey box. So when user opens the app by default browser window will open 'configure machine' web page by loading 'index.html' web page, same option can be opened by clicking the 'Configure Machine' option from sidebar or option menu. After the index.html page is presented to the user, the main process will wait for the 'save button' pressed by the user. After the 'save' button is pressed the form data is converted into JSON array and through ipc communication this json array is sent to main.js through identifier 'conf_data'. In main.js this data is received via 'conf_data' identifier, the data is validated and saved to opc_conf_data.xml. Error/success message in above process is again transmitted from main process to renderer process via 'conf_data_reply' identifier, once the renderer process receives the data through this identifier it will show the message in alert box. Same convention is followed throughout this documentation and will not be discussed again.

Configure Machine

Machine Name: OPC Machine

Machine Number: 123

Machine Location: BLK2BAY1

OPC URL ADDRESS: opc.tcp://localhost:4840/opcua

Anonymous Connection: ☐

Username:

Password:

Security Policy: **None** (Basic128RSA15, Basic256)

Message Security Mode:

Save

Figure 11 Configuration Setup mode4

Configure Machine

Machine Name: OPC Machine

Machine Number: 123

Machine Location: BLK2BAY1

OPC URL ADDRESS: opc.tcp://localhost:4840/opcua

Anonymous Connection: ☐

Username:

Password:

Security Policy: None

Message Security Mode: None

Save

New Machine successfully Configured
Use Connect option to communicate with the machine

OK

Figure 12 Machine configuration

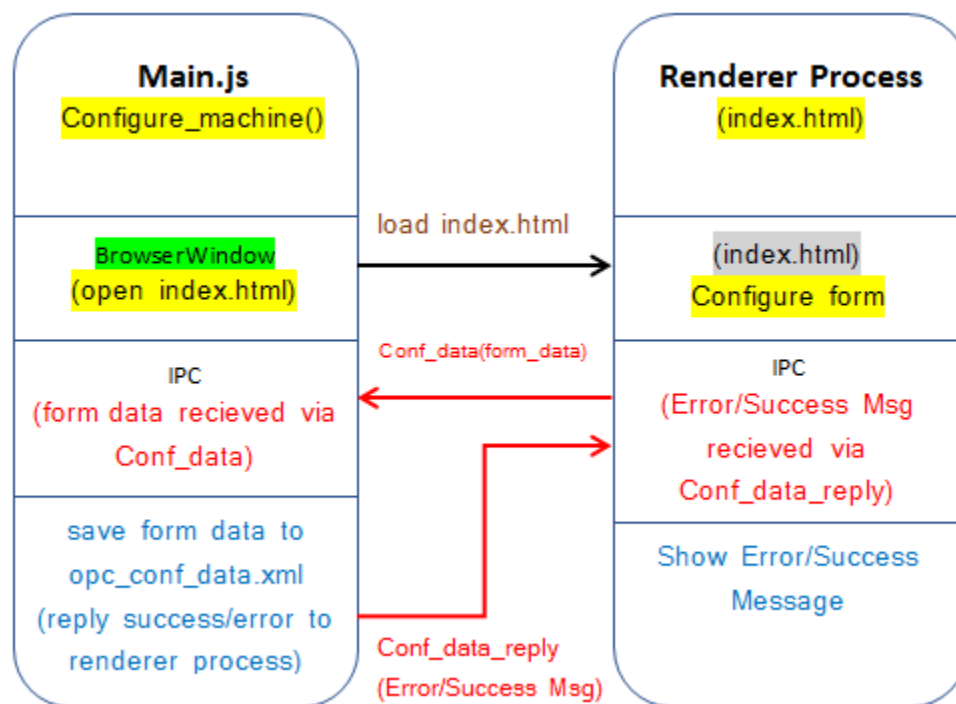


Figure 13 Configure Module

4.2 Connect Machine

After the successful configuration of the machine, now it is possible to get connected with the machine provided all the settings done in the configuration are correct with respect to the selected machine and machine is powered on with OPCUA Enabled features. Fig-14 & 15 shows 'Connect Machine' screen of the software, it must be noted that if no machine is connected then the action button will show connect option and configured machine name is shown inside the 'red' box as shown in Fig-14, while on the other hand if the configured machine is already connected then it shows 'disconnect' button and the configured machine name in 'green' box as shown in Fig-15.

Connect machine process involves 3 main steps

- 1) Discovery of the Machine/Devices
- 2) Connecting to the Machine/Devices
- 3) Session creation.

As per OPCUA specifications discovery of the server is Services used to discover the Endpoints implemented by a Server and to read the security configuration for those Endpoints. The Discovery Services are implemented by individual Servers and by dedicated Discovery Servers. Part 12 of OPCUA Specifications describes how to use the Discovery Services with dedicated Discovery Servers.

Every Server shall have a DiscoveryEndpoint that Clients can access without establishing a Session. This Endpoint may or may not be the same Session Endpoint that Clients use to establish a SecureChannel. Clients read the security information necessary to establish a SecureChannel by calling the GetEndpoints Service on the DiscoveryEndpoint.

In addition, Servers may register themselves with a well-known Discovery Server using the RegisterServer Service. Clients can later discover any registered Servers by calling the FindServers Service on the Discovery Server.

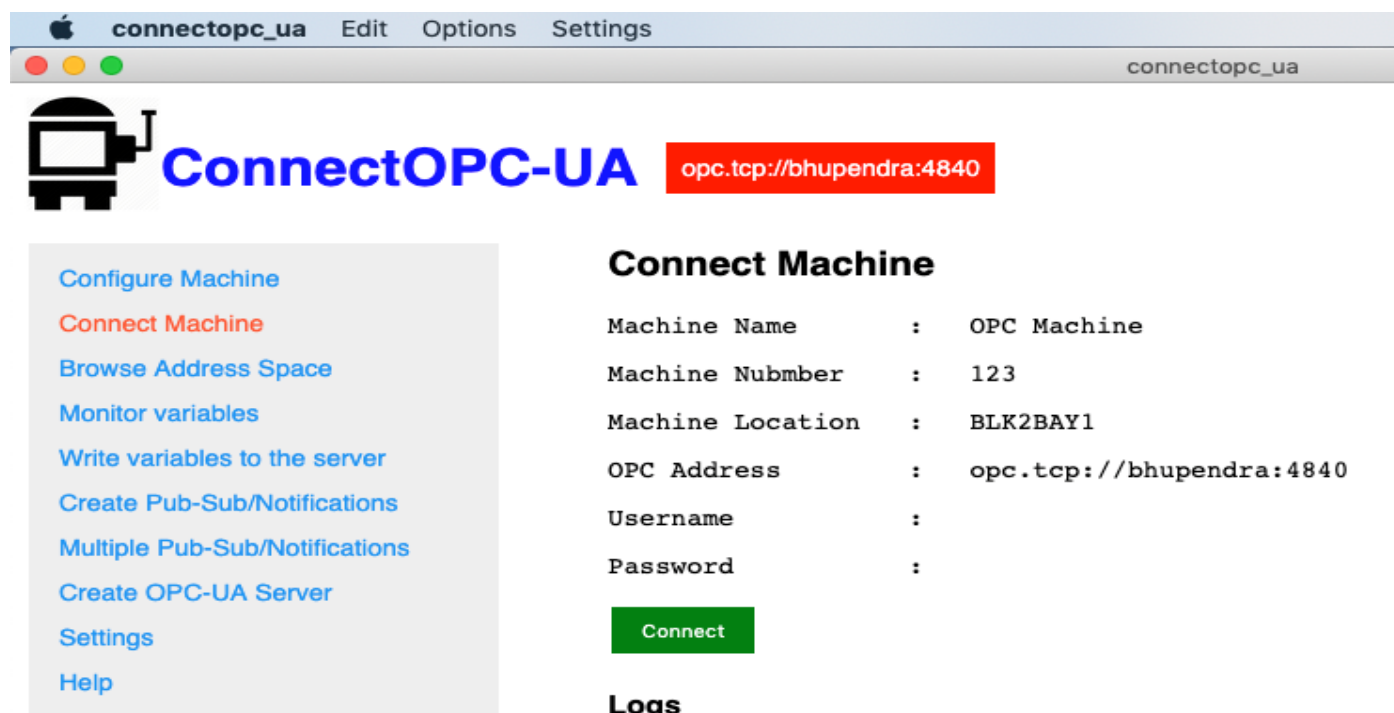


Figure 14 Connect machine

Fig-15 shows the discovery process of OPCUA. The URL for a DiscoveryEndpoint shall provide all of the information that the Client needs to connect to the DiscoveryEndpoint.

Once a Client retrieves the Endpoints, the Client can save this information and use it to connect directly to the Server again without going through the discovery process. If the Client finds that it cannot connect then the Server configuration may have changed and the Client needs to go through the discovery process again.

DiscoveryEndpoints shall not require any message security, but it may require transport layer security. In production systems, Administrators may disable discovery for security reasons and Clients shall rely on cached EndpointDescriptions. To provide support for systems with disabled Discovery Services Clients shall allow Administrators to manually update the EndpointDescriptions used to connect to a Server. Servers shall allow Administrators to disable the DiscoveryEndpoint.

After the successful discovery and connection to the machine (Server), a session is created that will be alive in the complete lifecycle of the application. Log as shown in Fig-14 & 15 will show all the events (Success or Error) in the connection attempt and session creation. The configured machine is connected or not will be shown in all the screens except 'Create OPCServer' option. In case of any error, or timeout the session will be closed and all the references to the objects will become invalid. In such cases 'Connect Machine' option need to be used again to ceated a new session.

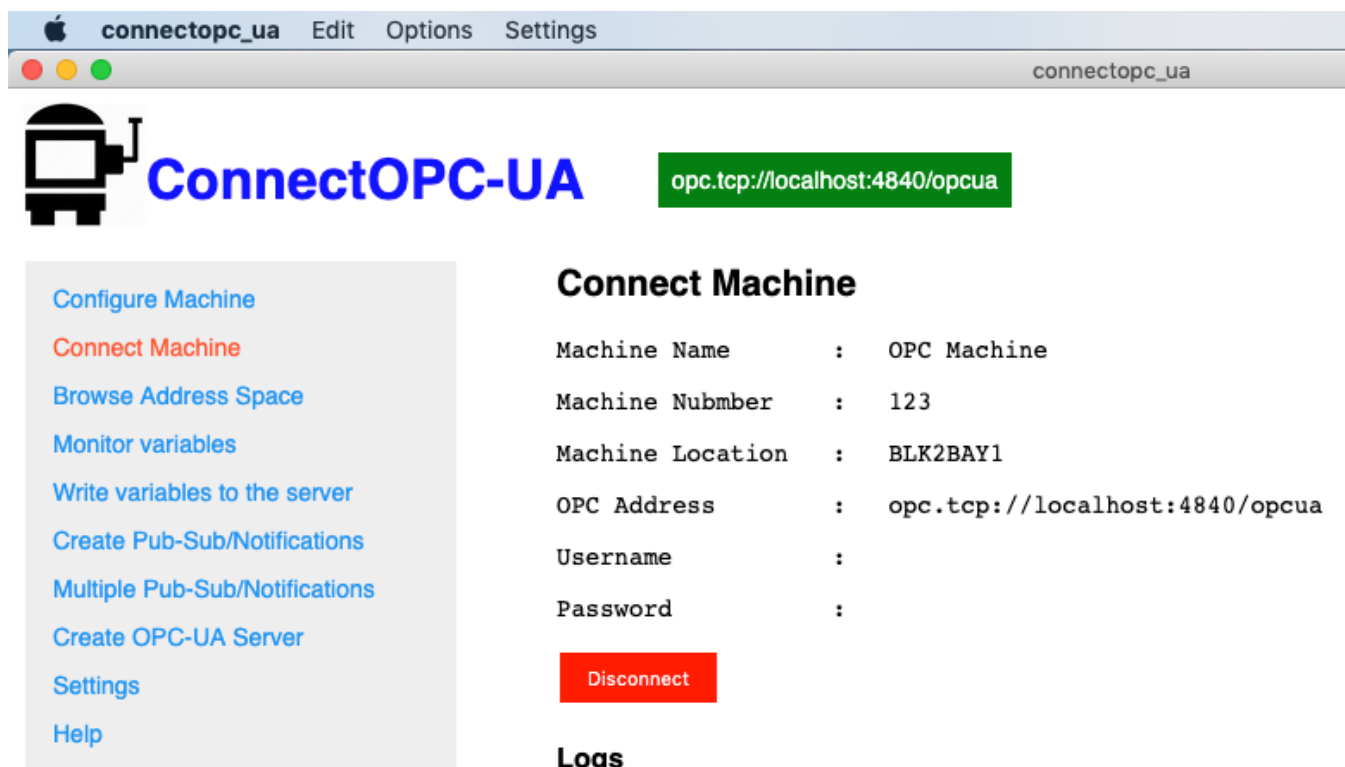


Figure 15 Disconnect Machine

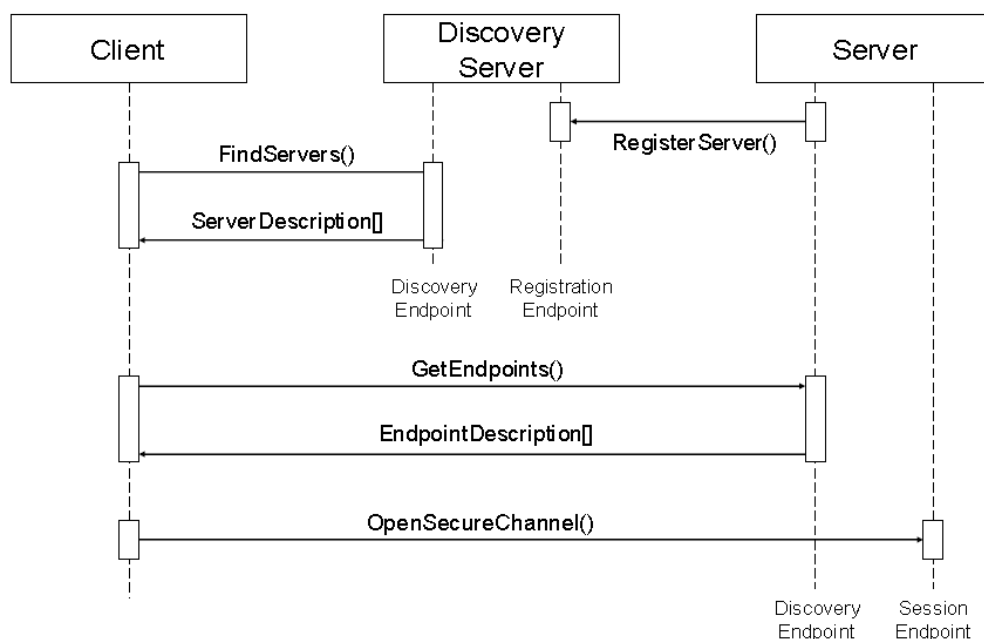


Figure 16 Discovery Process

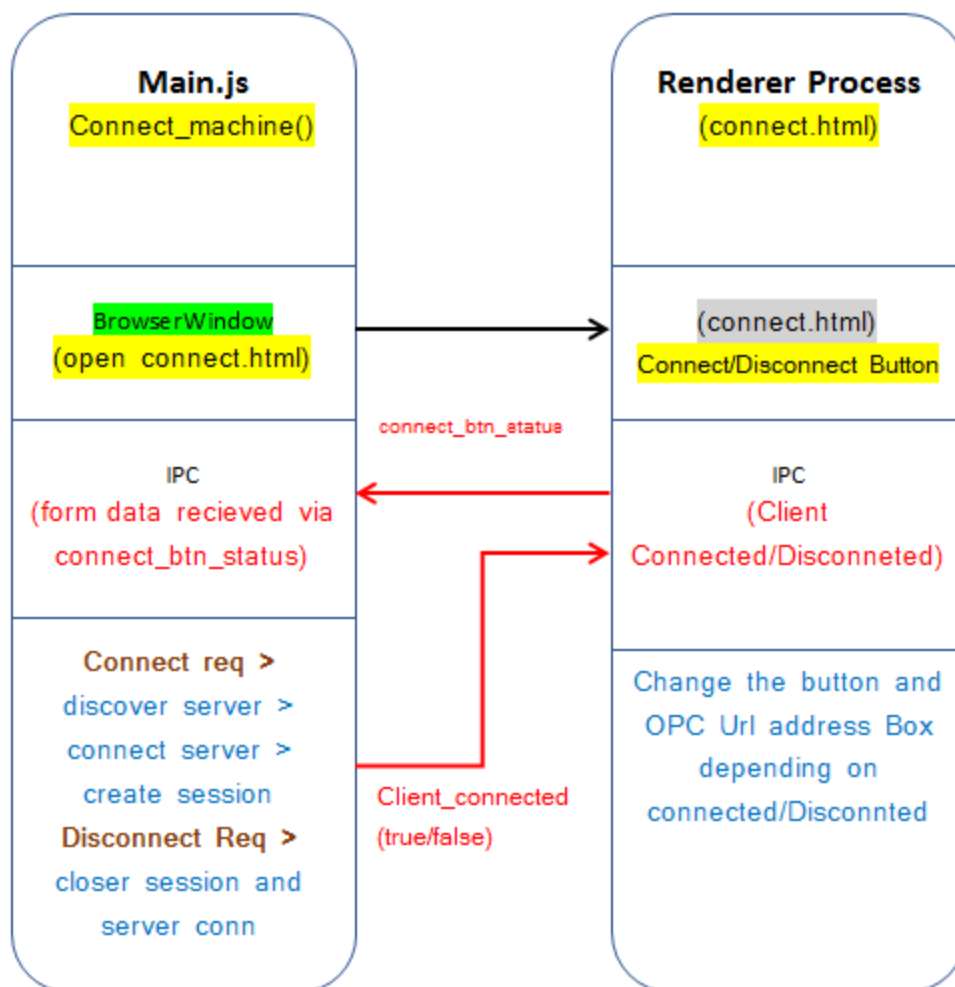


Figure 17 Connect Machine Module

When a Session is created, the Server adds an entry for the Client in its SessionDiagnosticsArray Variable. See Part 5 of OPCUA Specifications for a description of this Variable.

Sessions are created to be independent of the underlying communications connection. Therefore, if a communications connection fails, the Session is not immediately affected. The exact mechanism to recover from an underlying communication connection error depends on the SecureChannel mapping as described in Part 6 of OPCUA Specifications.

Sessions are terminated by the Server automatically if the Client fails to issue a Service request on the Session within the timeout period negotiated by the Server in the CreateSession Service response. This protects the Server against Client failures and against situations where a failed underlying connection cannot be re-established. Clients shall be prepared to submit requests in a timely manner to prevent the Session from closing automatically. Clients may explicitly terminate Sessions using the CloseSession Service.

When a Session is terminated, all outstanding requests on the Session are aborted and Bad_SessionClosed StatusCodes are returned to the Client. In addition, the Server deletes the entry for the Client from its SessionDiagnosticsArray Variable and notifies any other Clients who were subscribed to this entry.

If a Client invokes the CloseSession Service then all Subscriptions associated with the Session are also deleted if the deleteSubscriptions flag is set to TRUE. If a Server terminates a Session for any other reason, Subscriptions associated with the Session, are not deleted. Each Subscription has its own lifetime to protect against data loss in the case of a Session

termination. In these cases, the Subscription can be reassigned to another Client before its lifetime expires.

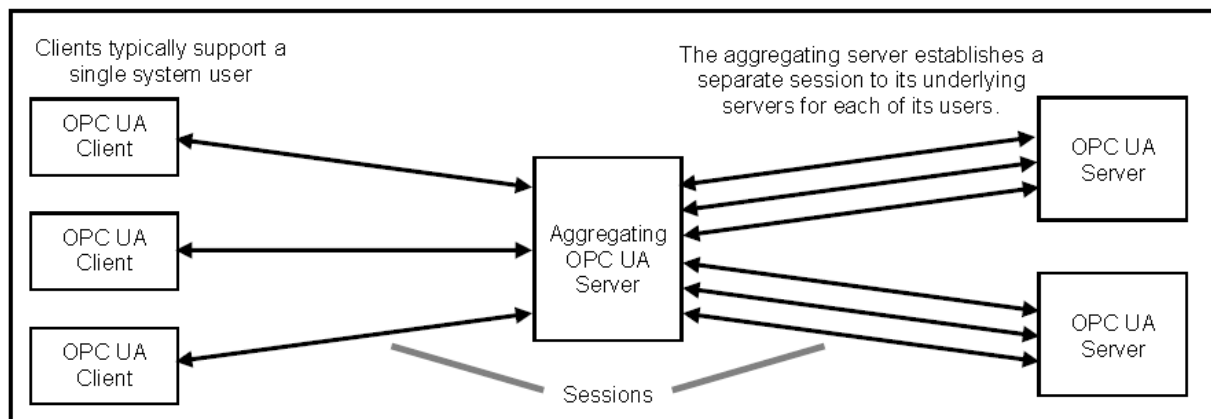


Figure 18 Multiplexing Users on a session

Some Servers, such as aggregating Servers, also act as Clients to other Servers. These Servers typically support more than one system user, acting as their agent to the Servers that they represent. Security for these Servers is supported at two levels.

First, each OPC UA Service request contains a string parameter that is used to carry an audit record id. A Client, or any Server operating as a Client, such as an aggregating Server, can create a local audit log entry for a request that it submits. This parameter allows the Client to pass the identifier for this entry with the request. If the Server also maintains an audit log, then it can include this id in the audit log entry that it writes. When the log is examined and the entry is found, the examiner will be able to relate it directly to the audit log entry created by the Client. This capability allows for traceability across audit logs within a system. See Part 2 for additional information on auditing. A Server that maintains an audit log shall provide the information in the audit log entries via event Messages defined in this standard. The Server may choose to only provide the Audit information via event Messages. The Audit EventType is defined in Part 3.

Second, these aggregating Servers may open independent Sessions to the underlying Servers for each Client that accesses data from them. Figure-18 illustrates this concept.

4.3 Browse Address Space

In OPC Unified Architecture, the address space is composed of nodes. The nodes are interconnected by means of references. The references do not necessarily have to be hierarchical; this means that the address space does not form a tree, but it is generally an interconnected "mesh" of nodes. Node classes are defining attributes and references for different nodes. OPC UA defines eight node classes. Each node in the address space is an instance of one of these node classes. Clients and servers are not allowed to define additional node classes or extend the definitions of these node classes (e.g. the list of attributes for a node class). OPC UA Node Classes provides more details on the different node classes.

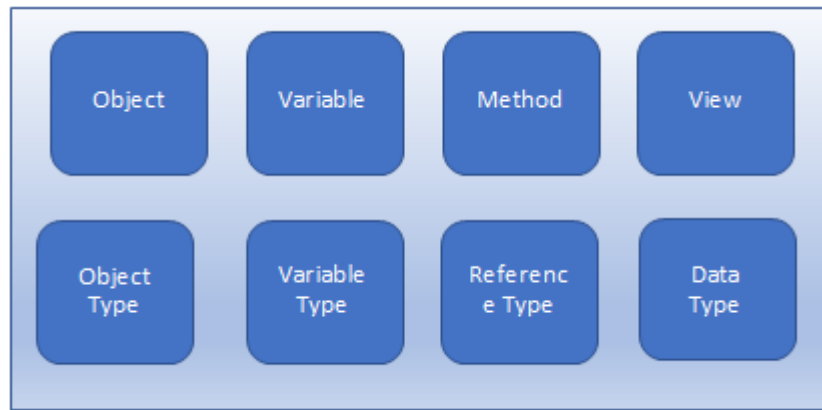


Figure 19 Node OPCUA Class

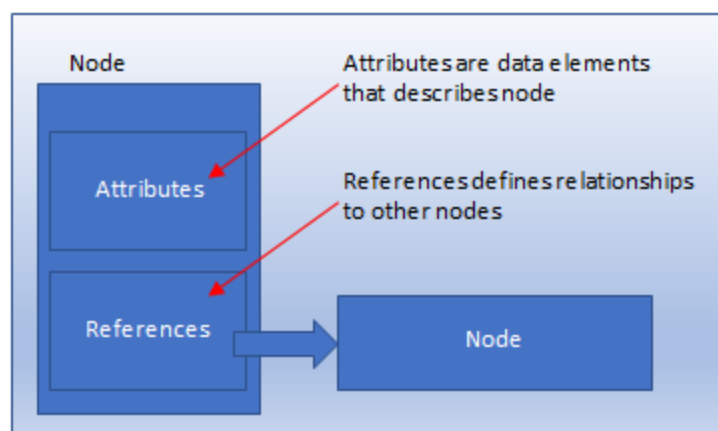


Figure 20 Address Space Model

Attributes are data elements that describe nodes. Clients can access attribute values using Read, Write, Query, and Subscription/MonitoredItem services.

Attributes are elementary components of node classes. Attribute definitions are part of the node class definitions and are therefore known to clients and are not directly visible in the address space.

Each attribute definition consists of an attribute id, a name, a description, a data type and a mandatory/optional indicator. The set of attributes defined for each node class cannot be extended by clients or servers. When a node is instantiated in the address space, the values of the mandatory node class Attributes must be provided.

References are used to relate nodes to each other. They can be accessed using the browsing and querying services.

Like attributes, they are defined as fundamental components of nodes. Unlike attributes, references are defined as instances of ReferenceType nodes. ReferenceType nodes are visible in the address space and are defined using the ReferenceType node class.

The node that contains the reference is referred to as the source node and the node that is referenced is referred to as the target node. The combination of the source node, the ReferenceType and the target node are used in OPC UA services to uniquely identify references. Thus, each node can reference another node with the same ReferenceType only once.

The target node of a reference may be in the same address space or in the address space of another OPC UA server. Target nodes located in other servers are identified in OPC UA services using a combination of the remote server name and the identifier assigned to the Node by the remote server.

This software gives you methods to traverse through the address space information and obtain the information available there. It is also possible to filter the returned nodes by criteria such as the node classes of interest, or reference types to follow.

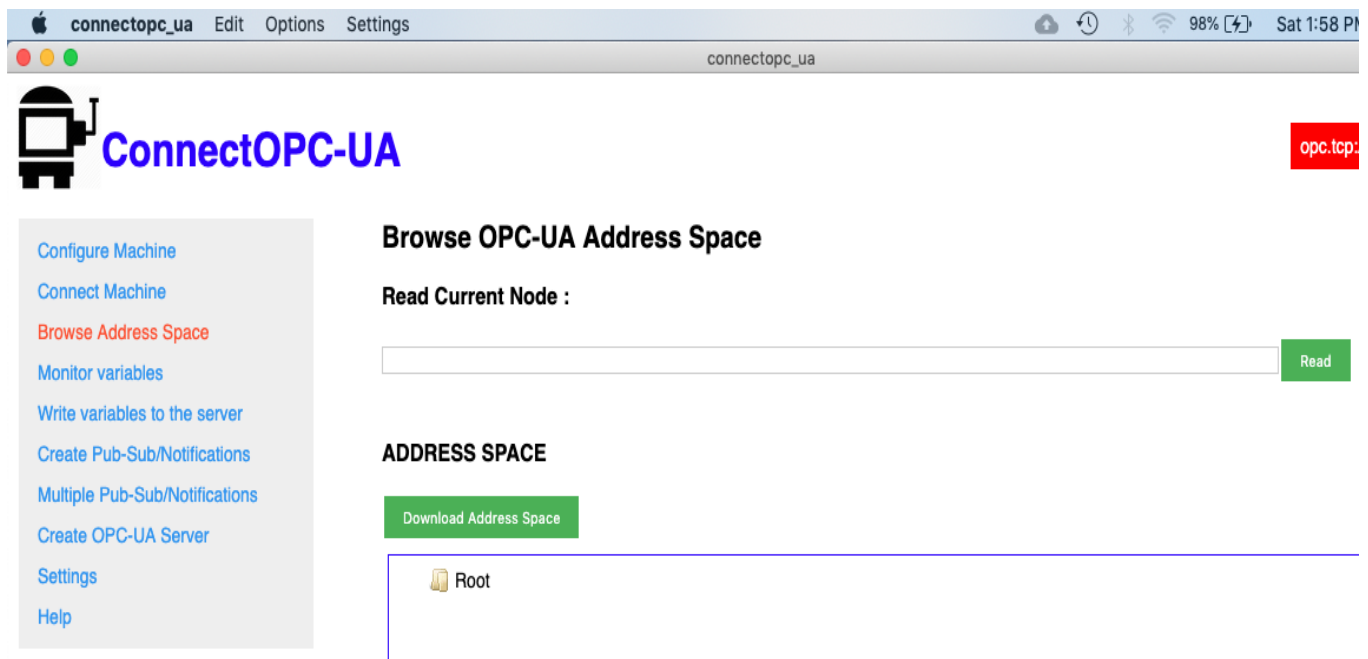


Figure 21 Browse Address space in software

Browse Address space shows only the 'Root' Folder at the top as shown in Fig-21, and when the root node is clicked then a list of nodes listed under the root nodes will be shown. On the same note clicking on any particular node, list of child nodes connected to this parent node will be shown.

Whenever any node is clicked, its nodeId gets visible in the 'read current node' option. With the read option, the value of the node selected can be read along with the server stamp values as shown in Fig-22.

The explored Address space can be downloaded via 'Download Address Space'. Please note 'Download Address Space' will not download the complete address space, it will download only the address space explored by the user in that session.

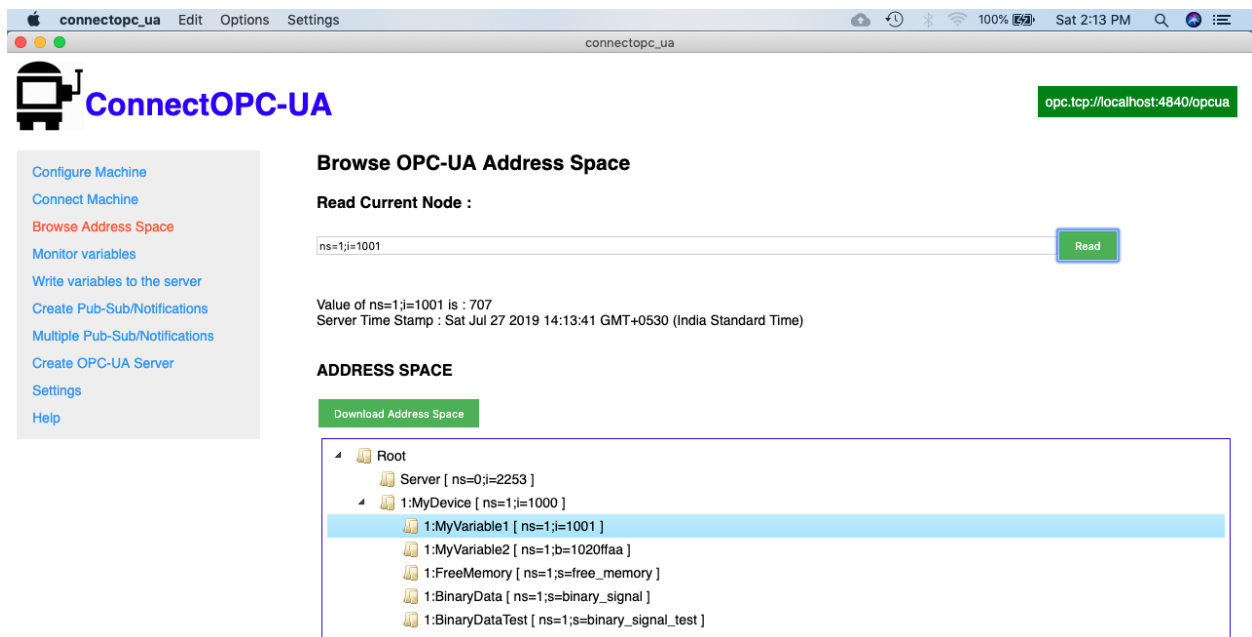


Figure 22 Sample of data showing in Browse Option

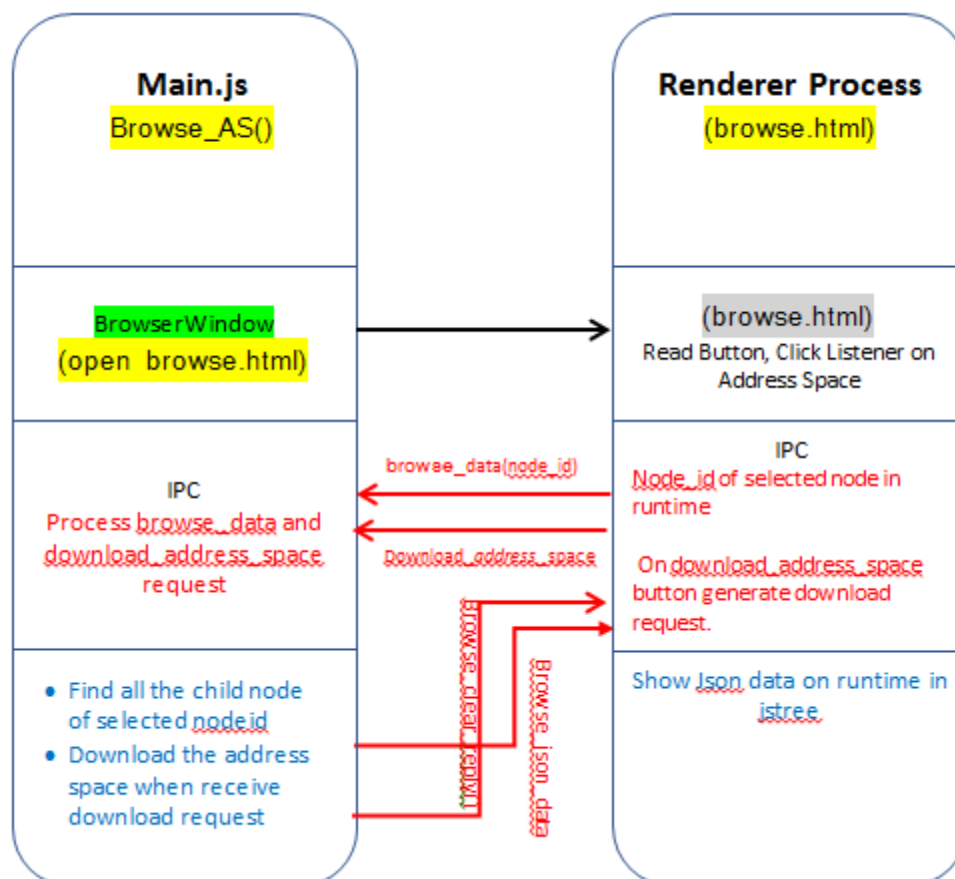


Figure 23 browse_address_space module

4.4 Read/Monitor Variables

Read variable can be used to read one or more attributes of a node. The maxAge parameter is used to direct the Server to access the value from the underlying data source, such as a device, if its copy of the data is older than that which the maxAge specifies. If the Server cannot meet the requested maximum age, it returns its "best effort" value rather than rejecting the request. A screenshot from the software is shown in Fig-24. Value of one of the variables and the server timestamp is shown in Fig-25.

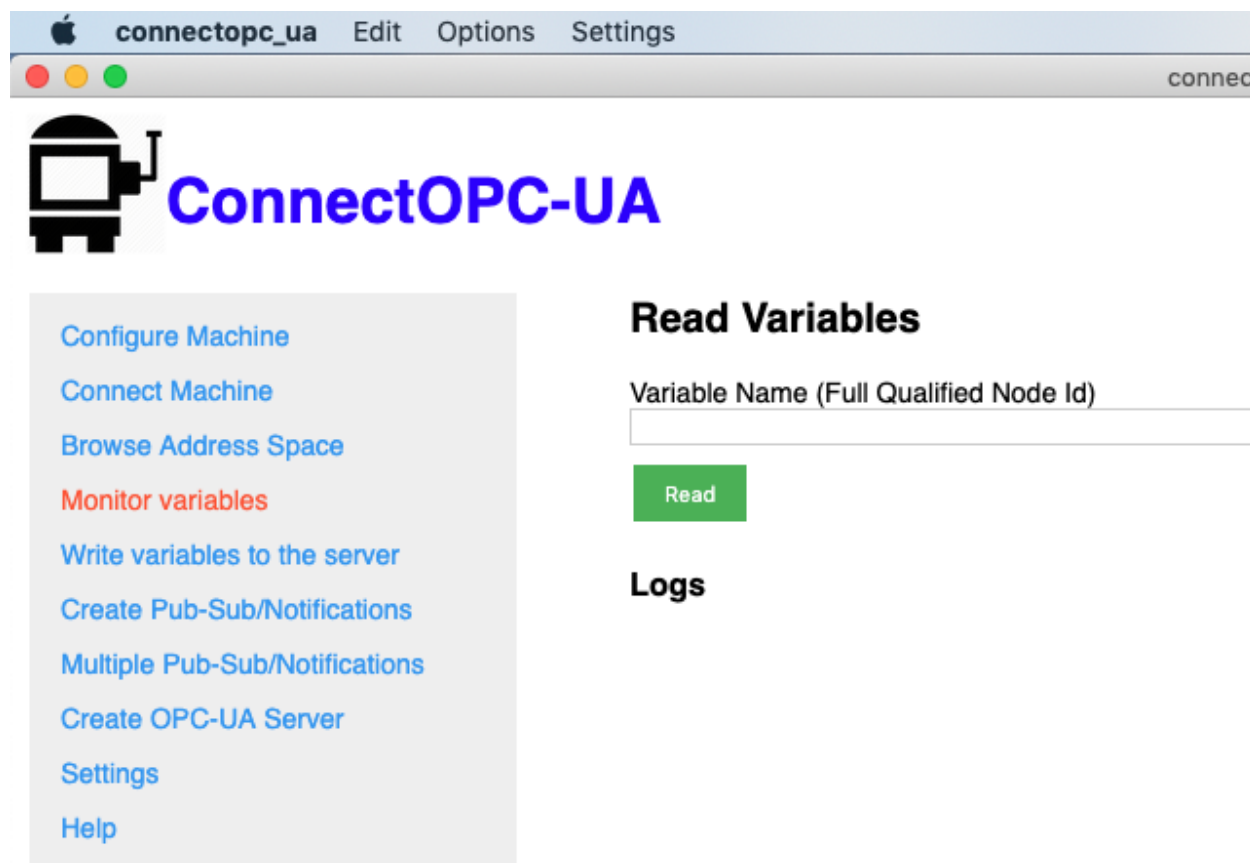


Figure 24 Read Variable Screen

You may define multiple read instructions for the OPC UA: Read Node, and you must define at least one. Each instruction has the following fields:

Namespace Index (ns) (Required): A string template or positive integer.

Identifier (Required): A string template or string for the node identifier. The identifier string should begin with one of the following types:

i=: for unsigned integer (UINT).

s=: for string.

g=: for global unique identifier (GUID).

b=: for byte string.

Result Key (Required): The key at which to store the result of this read operation. This key will exist on the Destination Path defined below the instructions. This can resolve to any string except errors, since that key is reserved for any errors that occur during reads

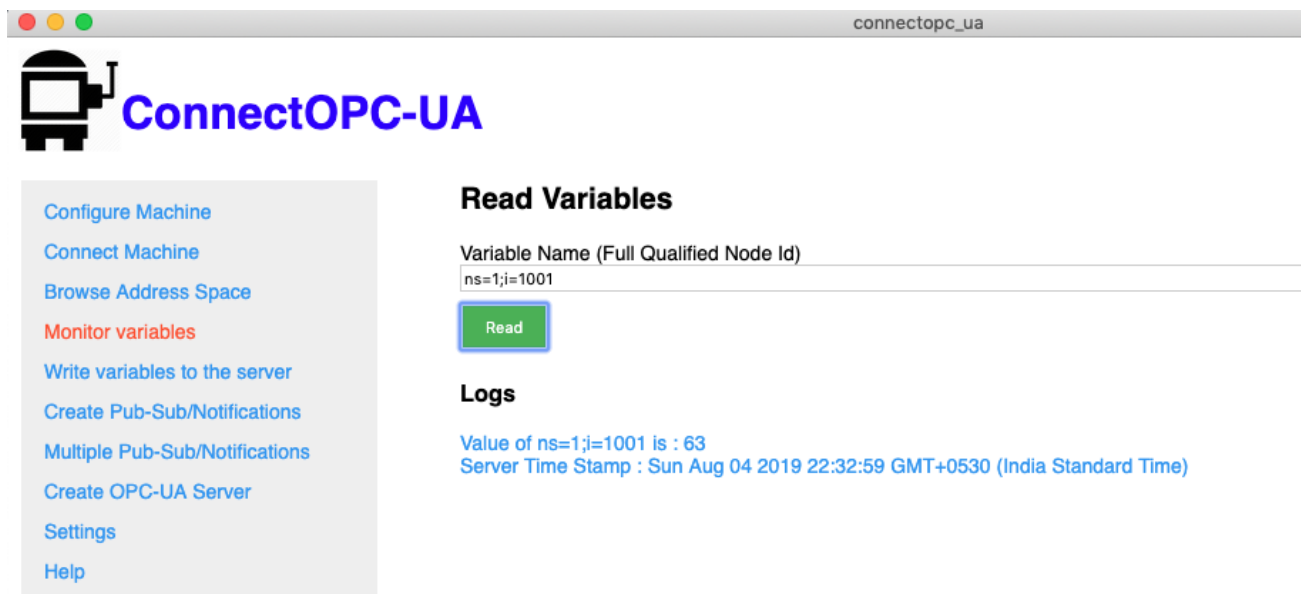


Figure 25 Read variable

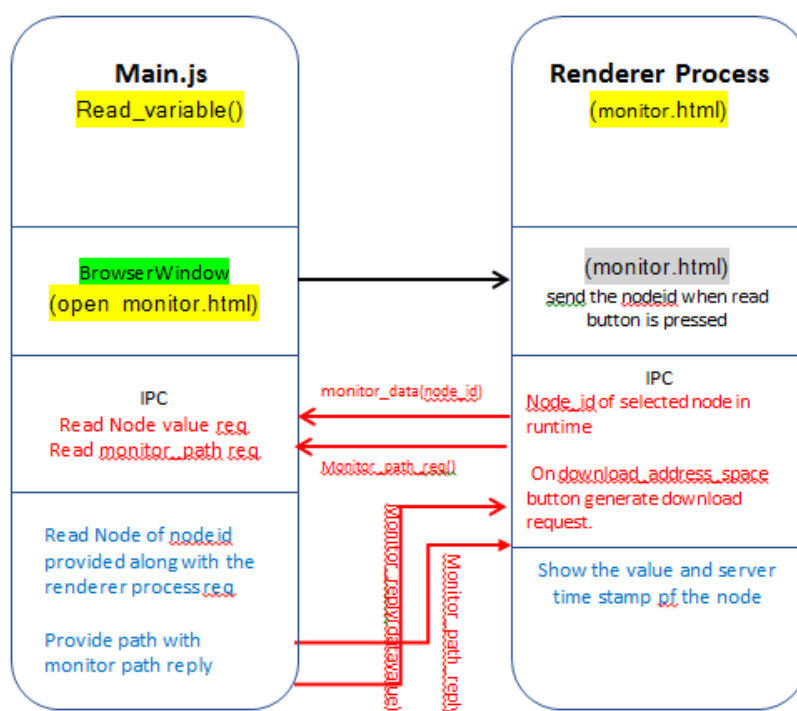


Figure 26 Read Variable Module

4.5 Write variables

The OPC UA: Write Node allows you to write to multiple new node values to your OPC UA Server

You may define multiple write instructions for the OPC UA: Write Node, and you must define at least one. Each instruction has the following fields:

Modules

Namespace Index (ns): (Required) A string template or positive integer.

Identifier: (Required) A string template or string for the node identifier. The identifier string should begin with one of the following types:

i=: for unsigned integer (UINT).

s=: for string.

g=: for global unique identifier (GUID).

b=: for byte string.

Value String Template/Payload Path: (Required) The value can be a string template or from a payload path. Note: If your node value is an array you will need to provide the array values on a payload path.

The success or failure of each write instruction will be placed in an object at the Destination Path (a payload path) (optional) under the write key. If the write is successful the write value will be success, and conversely if the write failed the write value will be fail.

If the write failed there will be a list of errors at the Destination Path under the key errors. Because the nodes are written as a group one failed write for a node may cause the other writes to fail.

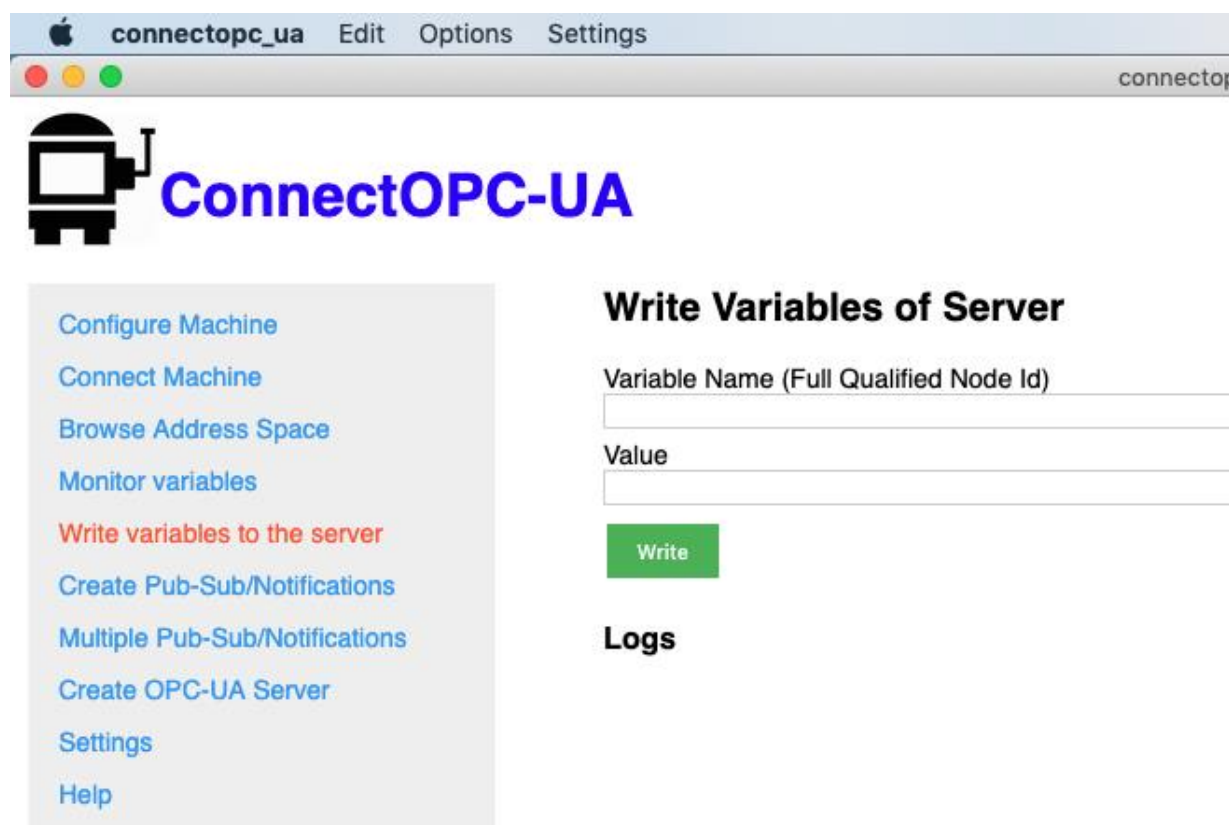


Figure 27 Write Variable Screenshot

Screenshot of write variable is shown in Fig-27 and how the value of Boolean variable is changed and written to the server is shown in Fig-28

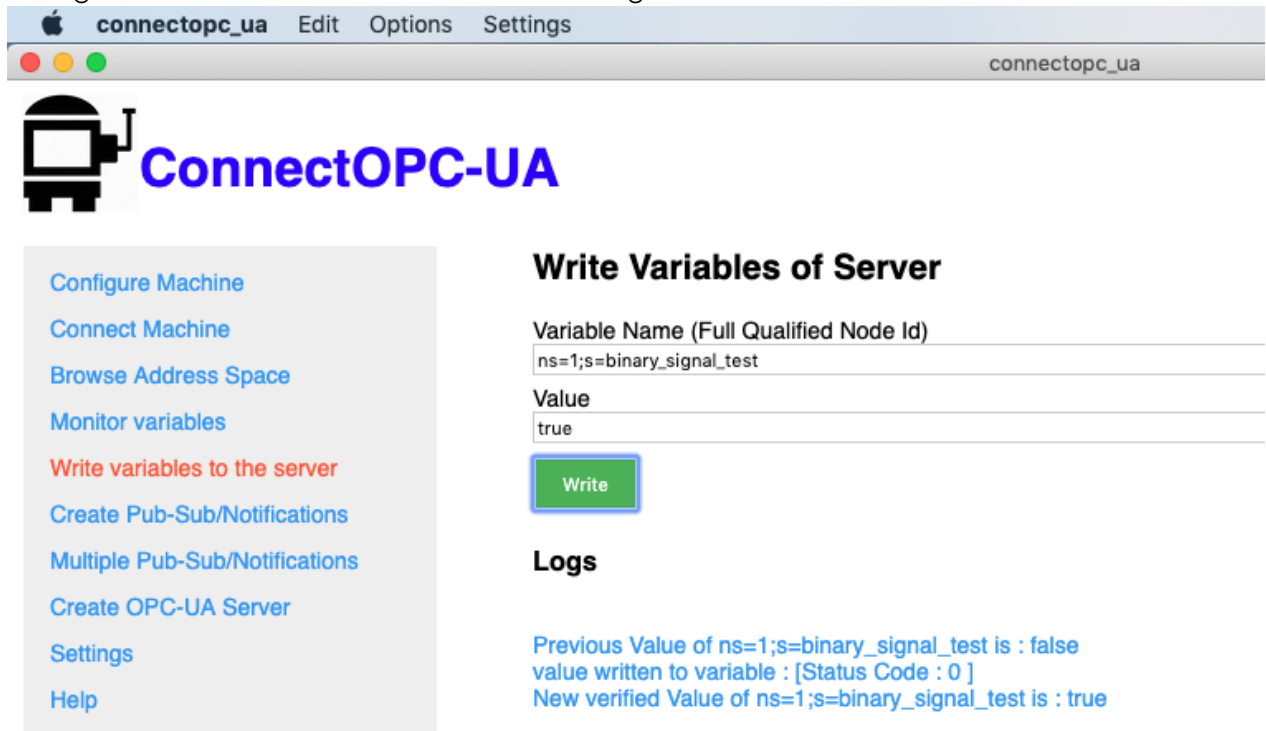


Figure 28 Value of Boolean variable is written to the server

In order to validate the value has been written to the server correctly, after writing to the server value of the variable again read and verify that the value actually has been written to the server as shown in fig-28.

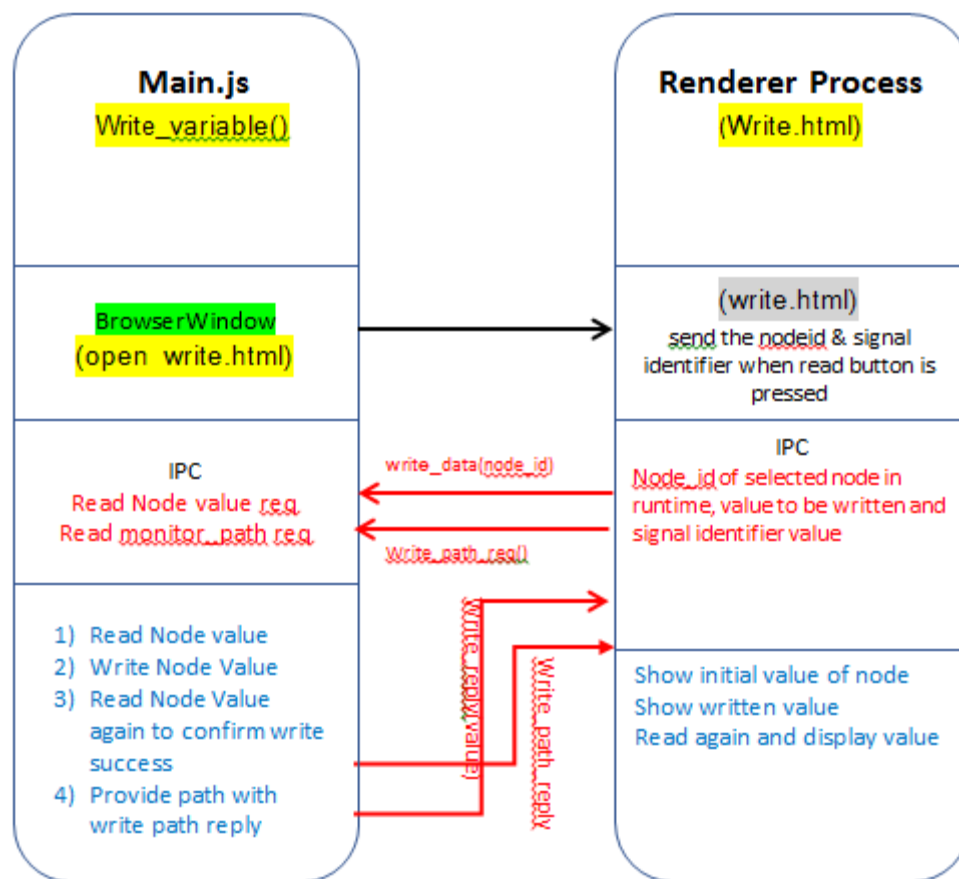


Figure 29 Implementation of write Variable

4.6 Pub-Sub Notifications

In contrast to permanently reading information (polling), OPC UA provides a more elegant functionality called Subscription. A UA Client can subscribe to a selection of Nodes of interest and let the server monitor these items. Only in case of changes, e.g. to their values, the server notifies the client about such changes. This mechanism reduces the amount of transferred data immensely. Besides the reduction of bandwidth this mechanism introduces further advantages and is the recommended mechanism to “read” information from a UA Server.

A client can subscribe to different types of information provided by an OPC UA server. The purpose of a Subscription is to group these sources of information, called Monitored Items, together, forming a piece of information called a Notification.

The following figure-30 shows the services which are involved when a client subscribes to data changes and events.

A Subscription consists of at least one Monitored Item, has to be created within the context of a Session and can be transferred to another Session. To create a Session, a Secure Channel between the Client and the Server has to be established.

There are three different types of “changes” a Client can subscribe to when adding Monitored Items to the Subscription:

- 1) Subscribe to data changes of Variable Values (Value attribute of a Variable),
- 2) Subscribe to Events of Objects (EventNotifier attribute of an Object & EventFilter Set),

- 3) Subscribe to aggregated Values, which are calculated in client-defined time intervals, based on current Variable Values.

- 4)

The settings available for Monitored Items and the Subscription are shown in the following figure-30.

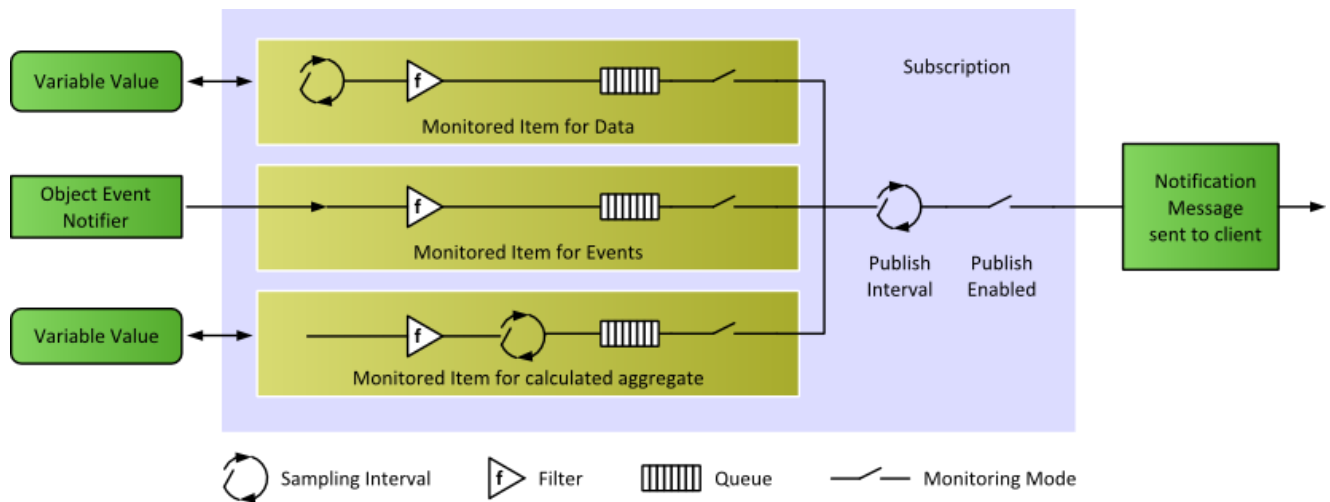


Figure 30 Settings for Subscription and monitored items

The sampling interval defines, individually for each Monitored Item in the subscription, a time interval. This is the rate at which the server checks the data source for changes. The Sampling can be set much faster than the notification to the Client, in which case the Server may queue the samples and publish the complete queue (see below). The server will revise the fastest possible sampling interval and by this “protect” the data source from extensive load that may be caused by the sampling itself. In the case of Aggregate Monitored Items the Sampling Interval denotes the interval for which the aggregates are calculated.

Filters contain several criteria to distinguish which data changes or events should be reported and which should be blocked. In the case of Aggregate Monitored Items it also contains the Raw Data Sampling Interval which defines the rate at which values are sampled from the data source.

A UA server may support queuing of data samples or events. The queue size, i.e. the maximum number of values which can be queued, can be configured for each monitored item. When the data gets delivered to the Client (Publish) the queue gets emptied. Each queue has a discard policy (e.g. discard oldest) in case an overflow occurs.

By setting the monitoring mode, the data sampling and reporting can be enabled or disabled.

There are two settings which affect the Subscription itself. After each Publishing Interval notifications collected in the queues are delivered to the client in a Notification Message (Publish Response). The Client must insure that the server has received enough Publish Tokens (Publish Requests), so that whenever the Publish Interval elapsed and a notification is ready to send, the server uses such a token and sends the data within a Publish Response. In case that there is nothing to report (e.g. no values have changed) the server will send a KeepAlive notification to the Client, which is an empty Publish, to indicate that the server is still alive. The sending of the Notification Message can be enabled or disabled by changing the Publish Enabled setting.

The most important parameters, `PublishingInterval` and `SamplingInterval` are used to define data sampling, but there are also parameters to define subscriptions status monitoring.

1) `PublishingInterval`

`PublishingInterval` defines how often the server checks if there are notification packages (publish responses) for a Subscription to be sent back to the client. A typical value is 1000 (ms). It should not be -1 nor 0.

Instead of using very small `PublishingInterval`, consider using a smaller `SamplingInterval`. The server may also increase the `PublishingInterval` to a default minimum.

2) `SamplingInterval` and `QueueSize`

If you wish to record samples at a faster rate than `PublishingInterval`, you should use the `SamplingInterval` of the monitored items. You will also need to reserve a longer `QueueSize`, to be able to keep all the samples in the server. The server can send all the samples in the queue in one notification – which is sent every `PublishingInterval`.

3) `KeepAliveCount`

If there is no data to send after the next `PublishingInterval`, the server will skip it. But `KeepAlive` defines how many intervals may be skipped, before an empty notification is sent anyway: to give the client a hint that the subscription is still alive in the server and that there just has not been any data arriving to the client.

4) `LifeTimeCount`

The client's responsibility is to send `PublishRequests` to the server, in order to enable the server to send `PublishResponses` back. The `PublishResponses` are used to deliver the notifications: but if there are no `PublishRequests`, the server cannot send a notification to the client.

The server will also verify that the client is alive by checking that new `PublishRequests` are received – `LifeTimeCount` defines the number of `PublishingIntervals` to wait for a new `PublishRequest`, before realizing that the client is no longer active. The Subscription is then removed from the server.

5) `Republish`

The notification packets are identified with a `SequenceNumber`. The `UaClient` can use these to detect if it has missed some of the notifications. If that happens, the client can make a `RepublishRequest` to request notifications that were missed during a communication break, for example. The server must keep the sent notifications to be able to resend, until the client has acknowledged that they were received.

6) `PublishRequestFactor`

`UaClient` will send several `PublishRequests` to the server when the first subscription is created. The server can use these, to fill in `PublishResponses` with new notifications. When the client receives the responses, it will send out new `PublishRequests`.

`PublishRequestFactor` determines how many requests are sent to the server: it is multiplied by the number of subscriptions to give the final number. The default value 2.

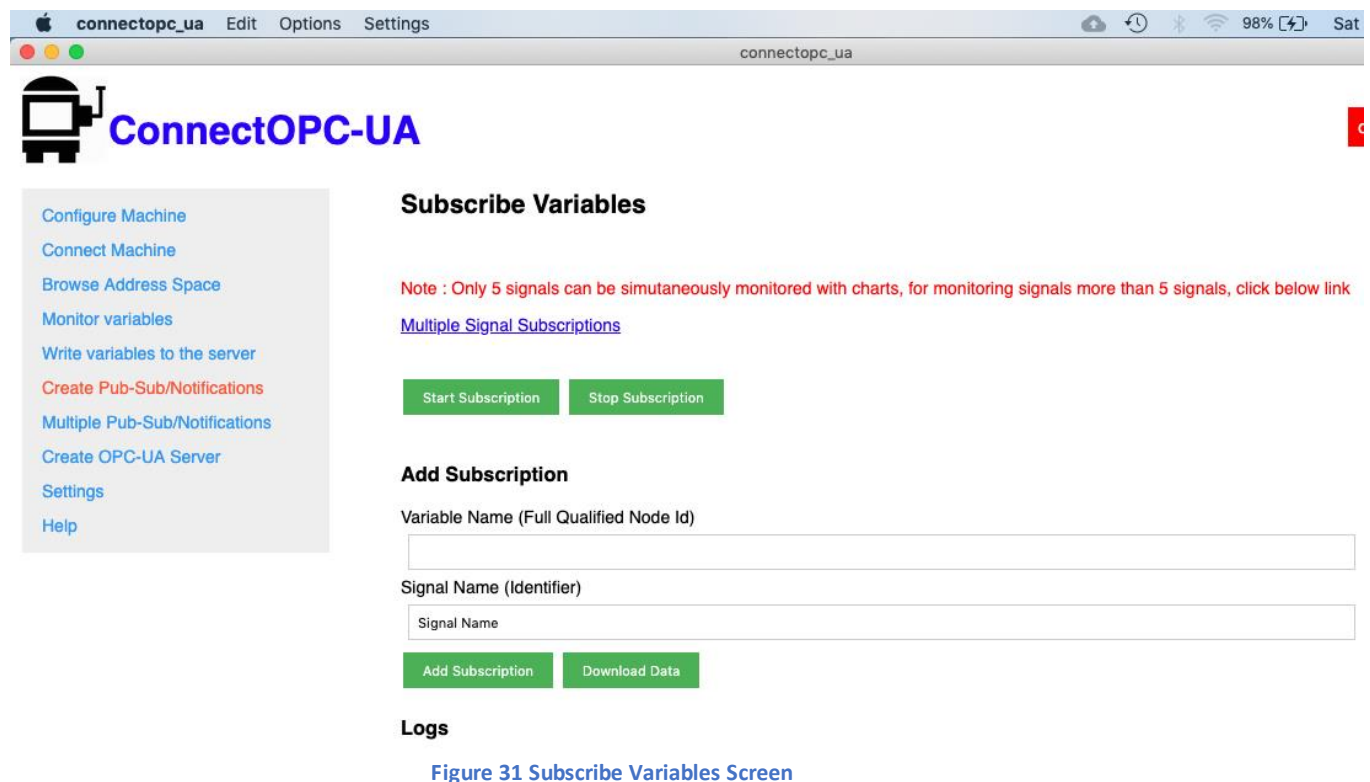


Figure 31 Subscribe Variables Screen

Fig-31 shows the screenshot of subscribe variable module from the application. In order to monitor a variable, subscription must be started at first, for starting the subscription 'Start Subscription' button must be pressed. It must be noted that in order to start a subscription software should be connected to the server and should have a valid session. Fig-32 shows that the subscription has been started indicating the subscription ID as well. After the successful creation of subscription, now any variable can be added for the monitoring. Fig-33 shows the monitoring of free_memory and chart shows the live values of free memory with respect to the server timestamp.

Server timestamp should be synchronized with the client software, in order to do that the time zone correction if needed can be done on the software. If server (Machine) time stamping are not correct then it should be corrected prior to the start of the monitoring the variables.

With this software option you can only monitor the live data of 5 variables simultaneously, any request after the 5th variable will be rejected. All the data being monitored with this screen at any time can be downloaded with 'Download Data' option. It must be noted that the data is available for that active session only, if due to some error session closes then no data will be available for the download.

Chart.js library is used for showing the live data interactively through the time charts, however same chart can not be downloaded as picture format, but downloaded data in form of excel file can generate the charts by using excel chart options. Further data analysis can also be done using any statistical softwares like R, Python, SAS etc.

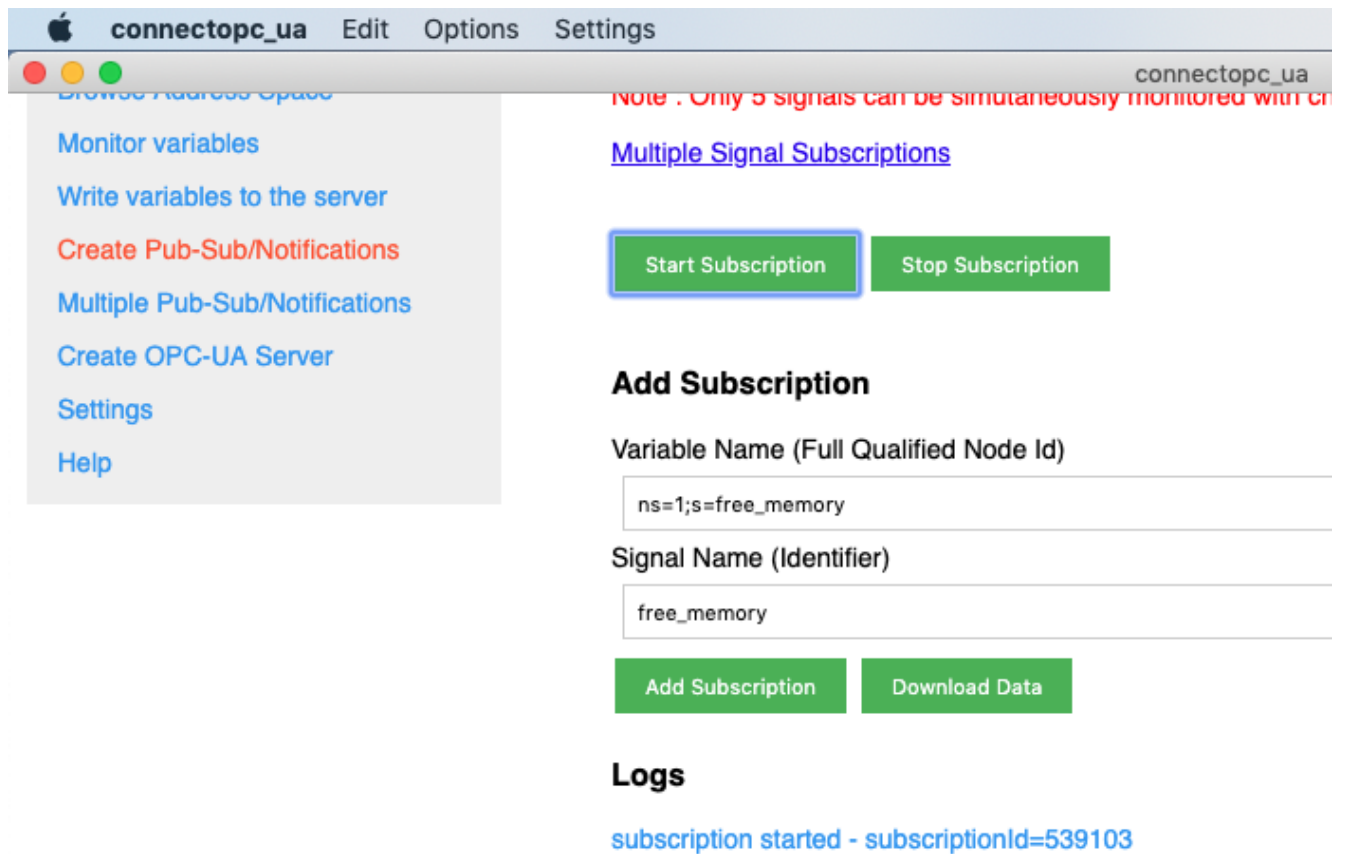


Figure 32 Starting subscription of Subscribe Variable Screen

Add Subscription

Variable Name (Full Qualified Node Id)

Signal Name (Identifier)

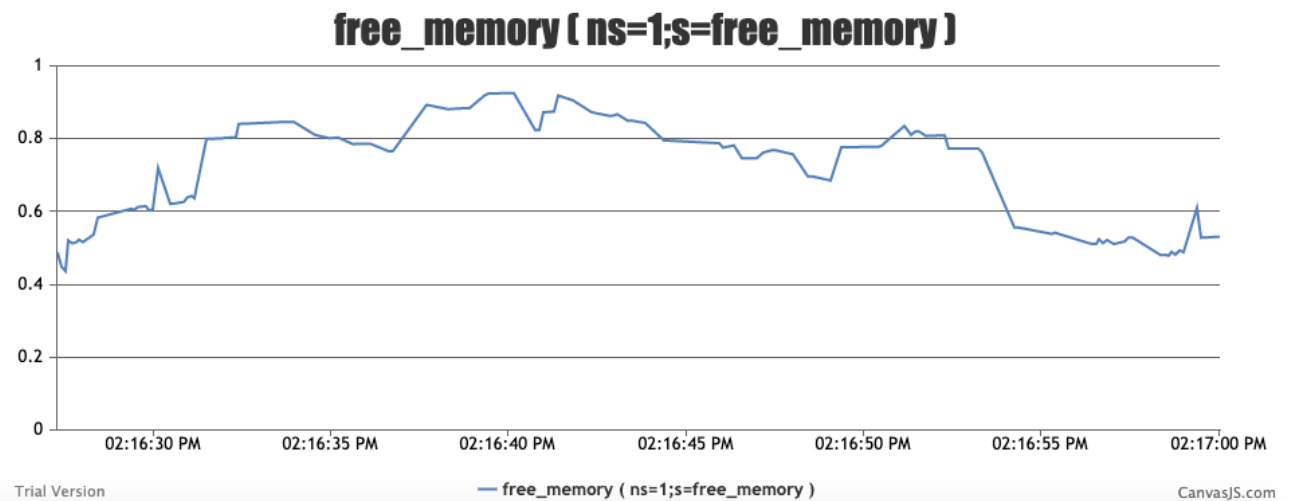
[Add Subscription](#)[Download Data](#)**Logs**[subscription started - subscriptionId=539103](#)

Figure 33 Monitoring of a variable from subscription

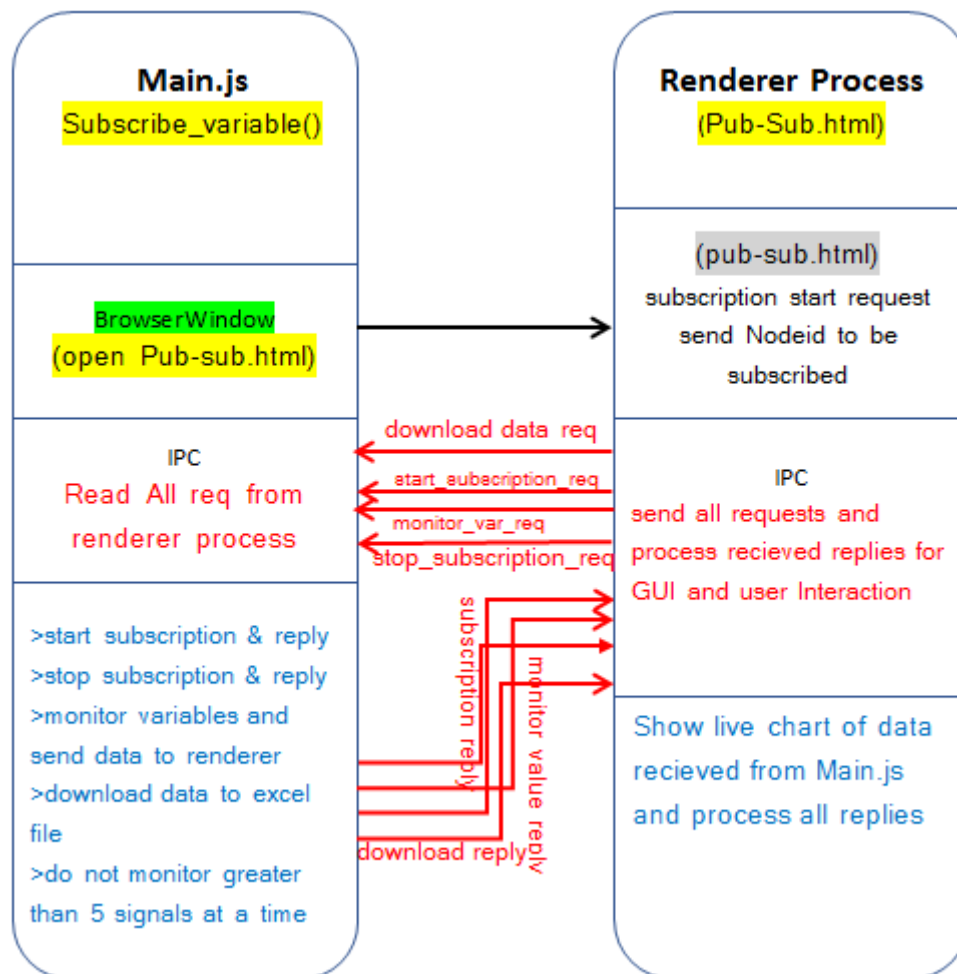


Figure 34 Implementation of Pub-sub

4.7 Multiple Pub-sub Notifications

Multiple pub-sub notifications is similar to pub-sub notifications, the only difference is there is no limit in the number of monitoring variables from the software side, although any limit from the server side is still valid, which is usually very large. Here tables are used in place of charts for the display of values of monitoring signals. All other options as discussed in section 4.6 is valid here as well. Address space is provided in this option where and node selected can be subscribed for the monitoring. Fig-35 shows the start of subscription and Fig-36 shows the monitoring of variable in the table.

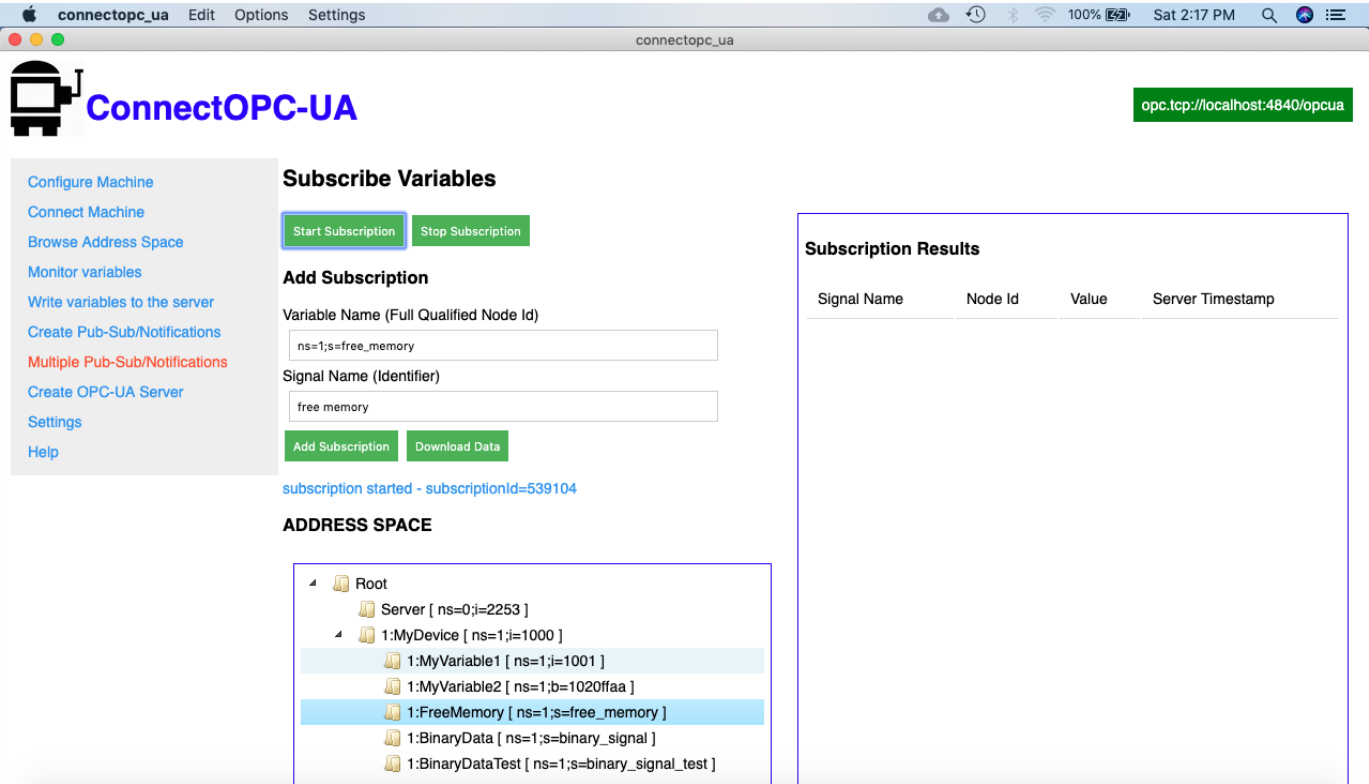


Figure 35 Subscription Start in Multiple Pub-Sub

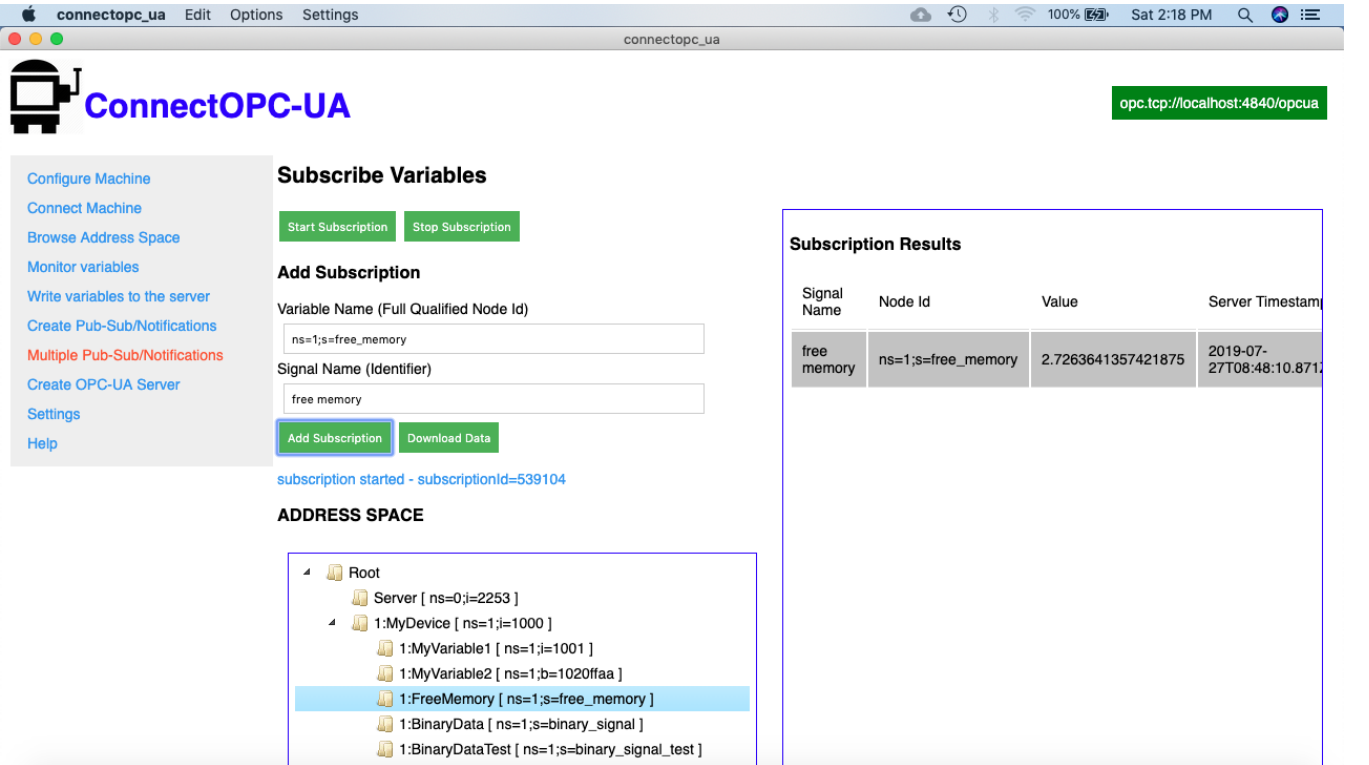


Figure 36 Monitoring of variable in Multiple Pub-sub Notification

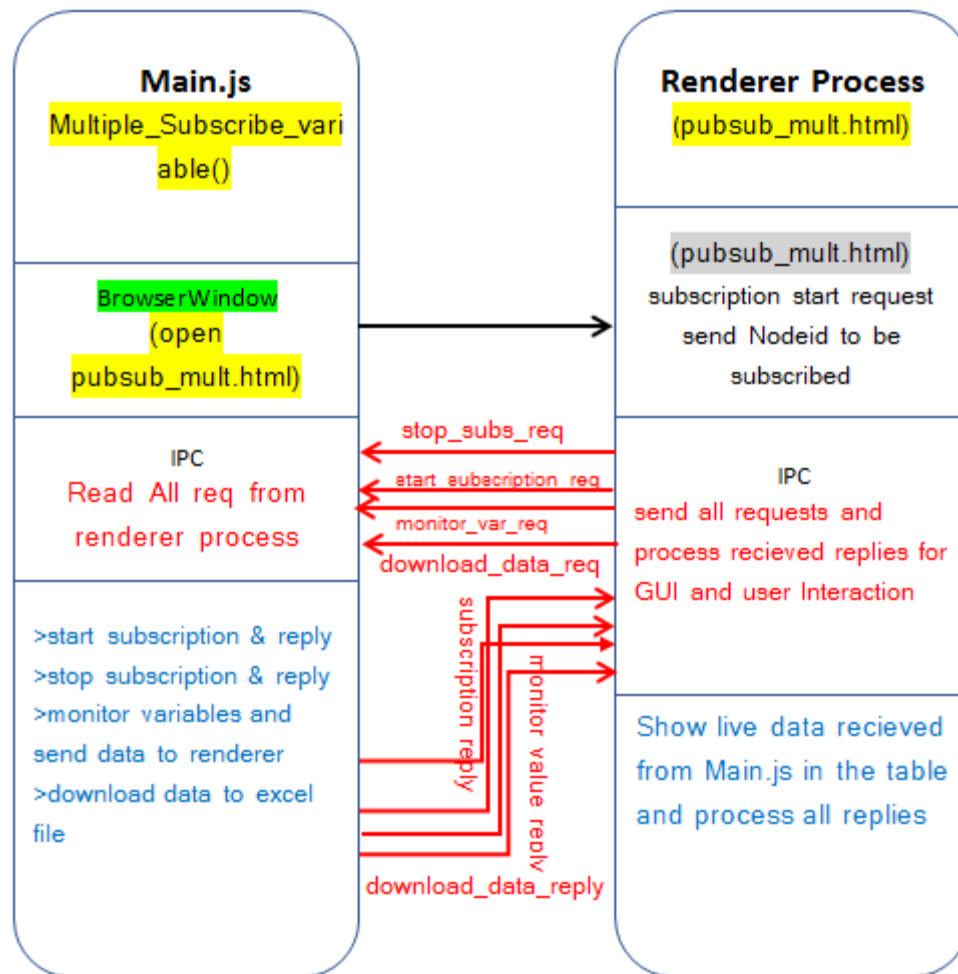


Figure 37 Implementation of Multiple Pub-Sub

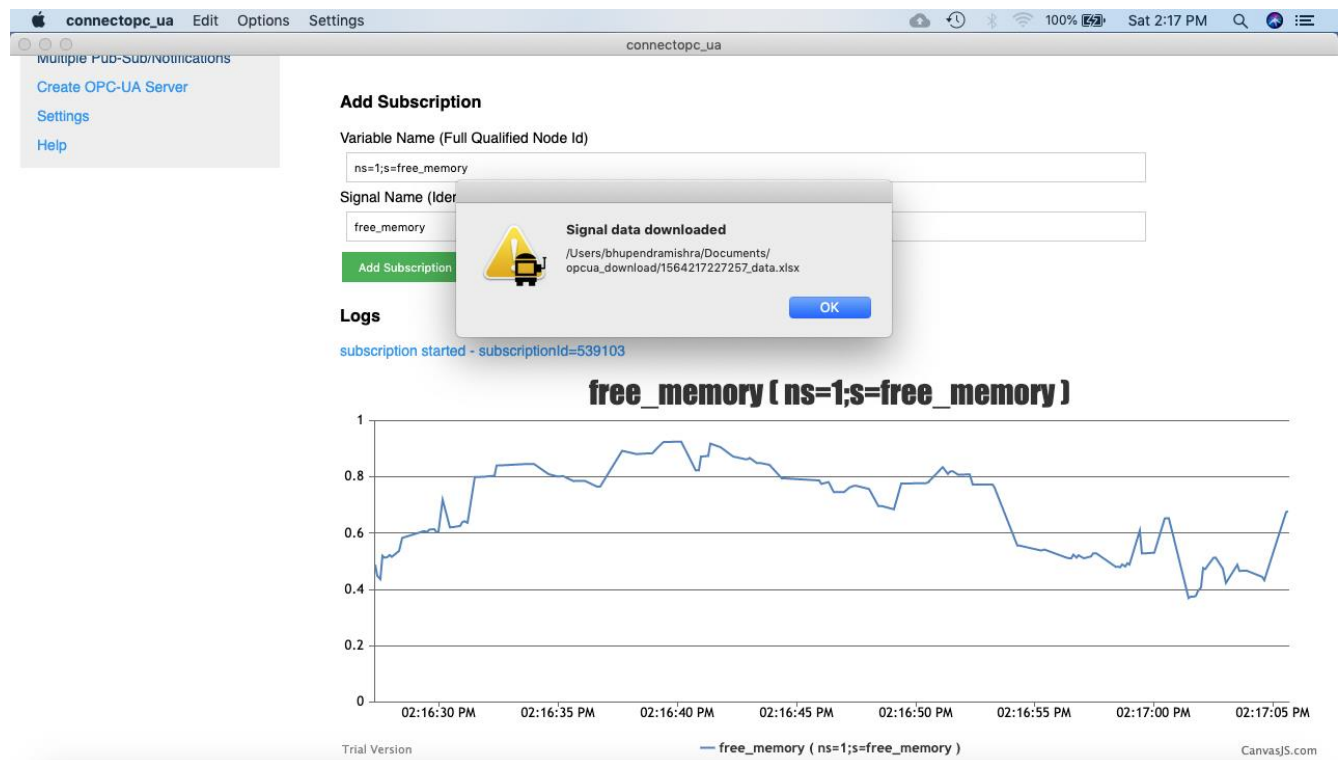


Figure 38 Downloading of Subscribed data

4.8 Creating OPCUA Server

It must be noted that the server implemented in the software is for training and diagnostics purpose only and do not provide any industrial use. In the same instance of the software OPCUA Server can be created and it can be connected with the software client as a machine, session can be created and software can access the Address Space of this server. So it can act as a good diagnostic tools to access the clients for any errors.

Creation of the OPCUA server, its types and capabilities are defined in part-5 of OPCUA specifications. Few Important points from the specifications are included here for the reference. All the capabilities provided by the OPCUA Server are defined in ServerCapabilitiesType Object, elements of this objects is described below which is used in this software.

ServerProfileArray lists the Profiles that the Server supports. See Part 7 for the definitions of Server Profiles. This list should be limited to the Profiles the Server supports in its current configuration.

LocaleIdArray is an array of LocaleIds that are known to be supported by the Server. The Server might not be aware of all LocaleIds that it supports because it may provide access to underlying servers, systems or devices that do not report the LocaleIds that they support.

MinSupportedSampleRate defines the minimum supported sample rate, including 0, which is supported by the Server.

MaxBrowseContinuationPoints is an integer specifying the maximum number of parallel continuation points of the Browse Service that the Server can support per session. The value specifies the maximum the Server can support under normal circumstances, so there is no

guarantee the Server can always support the maximum. The client should not open more Browse calls with open continuation points than exposed in this Variable. The value 0 indicates that the Server does not restrict the number of parallel continuation points the client should use.

MaxQueryContinuationPoints is an integer specifying the maximum number of parallel continuation points of the QueryFirst Services that the Server can support per session. The value specifies the maximum the Server can support under normal circumstances, so there is no guarantee the Server can always support the maximum. The client should not open more QueryFirst calls with open continuation points than exposed in this Variable. The value 0 indicates that the Server does not restrict the number of parallel continuation points the client should use.

MaxHistoryContinuationPoints is an integer specifying the maximum number of parallel continuation points of the HistoryRead Services that the Server can support per session. The value specifies the maximum the Server can support under normal circumstances, so there is no guarantee the Server can always support the maximum. The client should not open more HistoryRead calls with open continuation points than exposed in this Variable. The value 0 indicates that the Server does not restrict the number of parallel continuation points the client should use.

SoftwareCertificates is an array of SignedSoftwareCertificates containing all SoftwareCertificates supported by the Server. A SoftwareCertificate identifies capabilities of the Server. It contains the list of Profiles supported by the Server. Profiles are described in Part 7.

MaxArrayLength Property indicates the maximum length of a one or multidimensional array supported by Variables of the Server. In a multidimensional array it indicates the overall length. For example, a three-dimensional array of 2x3x10 has the array length of 60. The Server might further restrict the length for individual Variables without notice to the client. Servers may use the Property MaxArrayLength defined in Part 3 on individual DataVariables to specify the size on individual values. The individual Property may have a larger or smaller value than MaxArrayLength.

MaxStringLength Property indicates the maximum number of bytes in Strings supported by Variables of the Server. Servers may override this setting by adding the MaxStringLength Property defined in Part 3 to an individual DataVariable. If a Server does not impose a maximum number of bytes or is not able to determine the maximum number of bytes this Property shall not be provided.

MaxByteStringLength Property indicates the maximum number of bytes in a ByteString supported by Variables of the Server. It also specifies the default maximum size of a FileType Object's read and write buffers. Servers may override this setting by adding the MaxByteStringLength Property defined in Part 3 to an individual DataVariable or FileType Object. If a Server does not impose a maximum number of bytes or is not able to determine the maximum number of bytes this Property shall not be provided.

OperationLimits is an entry point to access information on operation limits of the Server, for example the maximum length of an array in a read Service calls.

ModellingRules is an entry point to browse to all ModellingRules supported by the Server. All ModellingRules supported by the Server should be able to be browsed starting from this Object.

AggregateFunctions is an entry point to browse to all AggregateFunctions supported by the Server. All AggregateFunctions supported by the Server should be able to be browsed starting from this Object. AggregateFunctions are Objects of AggregateFunctionType.

RoleSet Object is used to publish all Roles supported by the Server. The RoleSetType is specified in F.2

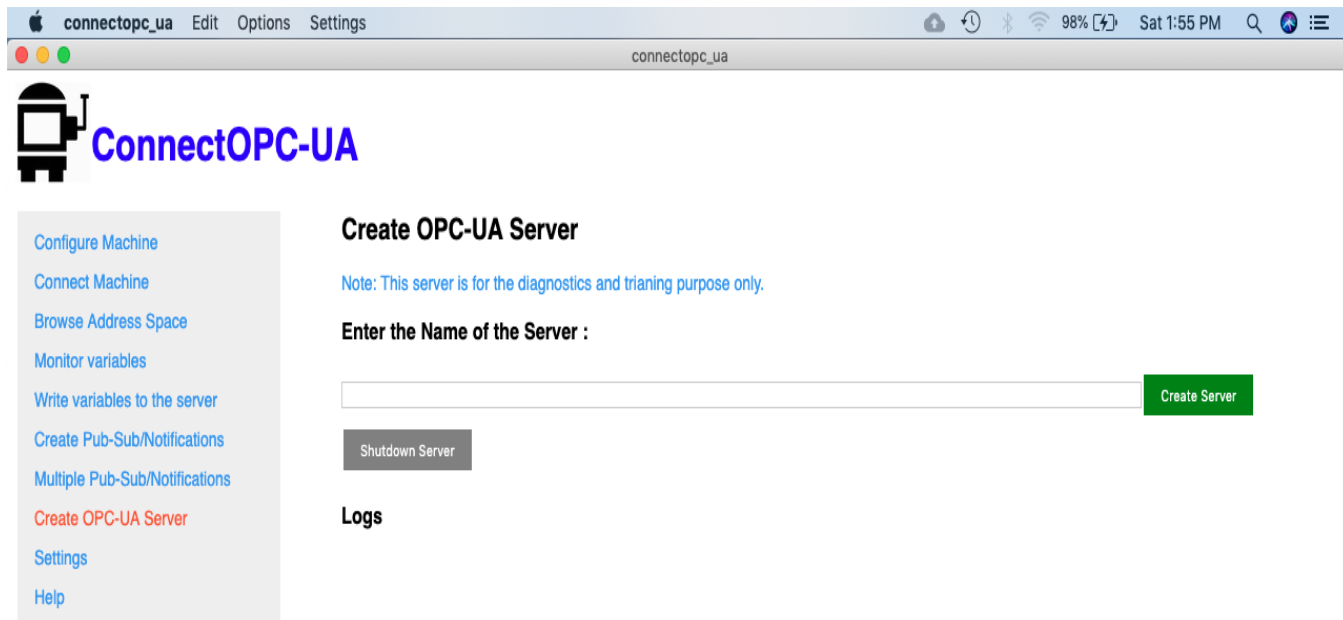


Figure 39 Creation of OPCUA Server Screen

Screen for the creation of the server is shown in fig-39, after pressing the button 'Create Server', OPCUA Server will be created and will receive and transmit the network connections on the specified port set in the settings option. Fig-40 shows the successful creation of the server and the OPC URL of the server. Fig-41 shows the permission asked by the OS to give access to the application for receive and transmit data through the network on the specified port. When shutdown server button is clicked server will be stopped and all the connections established will be refused. Any further request of the connection to this server will be refused. The server implemented through this software is having its fixed address space and its node can be accessed through the node id. Following are the details of the address space of the server.

Variable name	NodeId	Data type	Remark
MyVariable1	ns=1;i=1001	int32	
MyVariable2	ns=1;b=1020ffaa	int32	
FreeMemory	ns=1;s=free_memory	int32	
BinaryData	ns=1;s=binary_signal	boolean	
BinaryDataTest	ns=1;s=binary_data_test	boolean	
FileRead	ns=1;s=File_Read	ByteString	It will read the file present at the specified download folder in the settings bt the name 'read.txt'
N-Root	ns=1;i=1007	Method Call	2 Input Argument 1st is number 2nd is nth root and out the nth

			root of the number
Simulated_Variable	ns=1;s=Simulated_variable	Double	It produces Sine Wave WRT Time
EventNotifier	ns=1;s=Simulated-Event	Event Notifier	

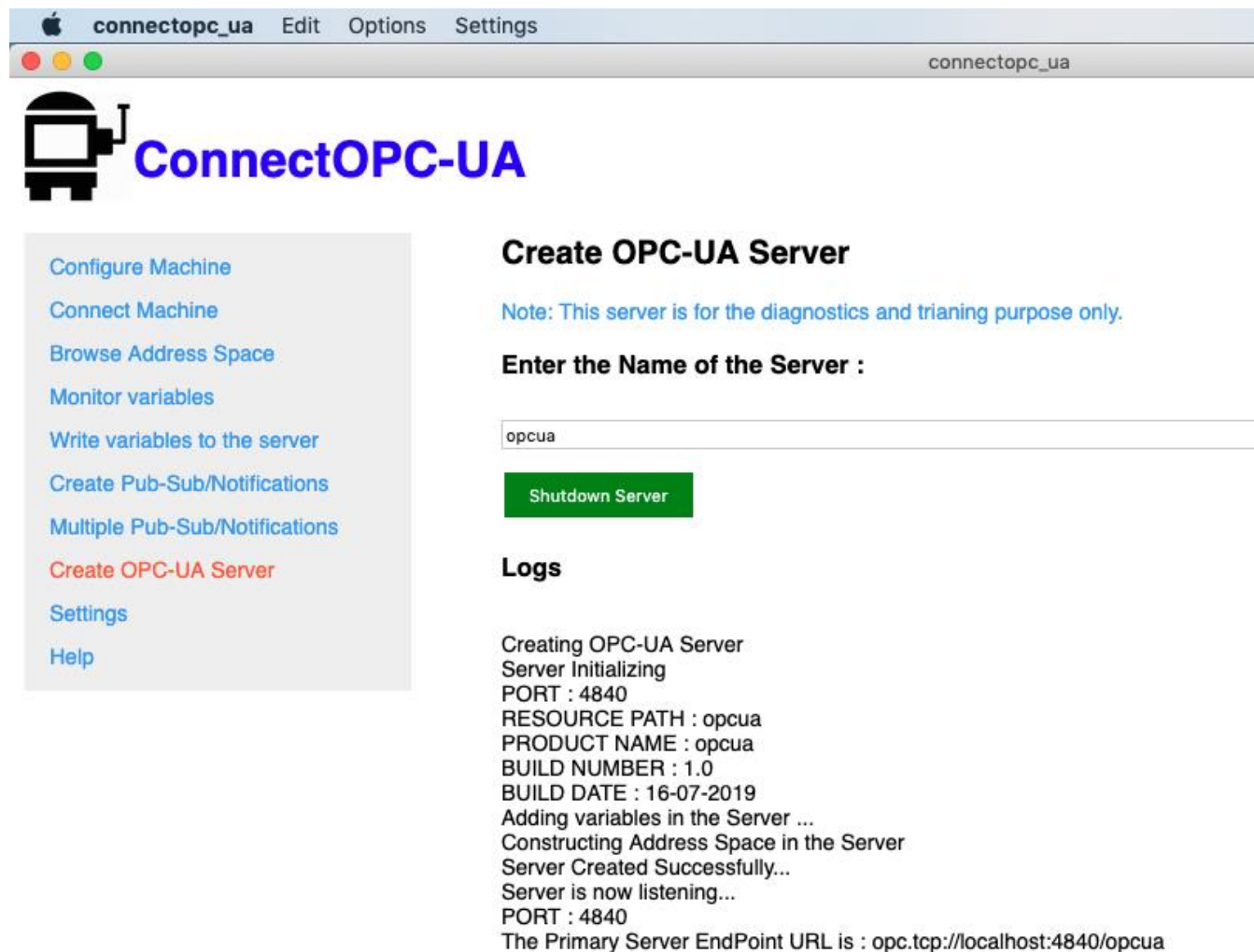


Figure 40 Creation of OPCUA Server

MyVariable1 (ns=1;i=1001) is a counter increasing by 1 after every 500ms, MyVariable2 (ns=1;i=1020ffaa) is having a fixed value of 10 but any other value can be written to the server using write variable option. Free Memory (ns=1;s=free_memory) is calculating some free memory ratio. BinaryDataTest (ns=1;s=binary_data_test) is a variable where boolean data can be written to the server.

So the variables of above address space can be used for writing to the server, reading the value from it, browsing the address space, subscribing the variables and analysing the data from the live charts and monitoring multiple values from the subscription.

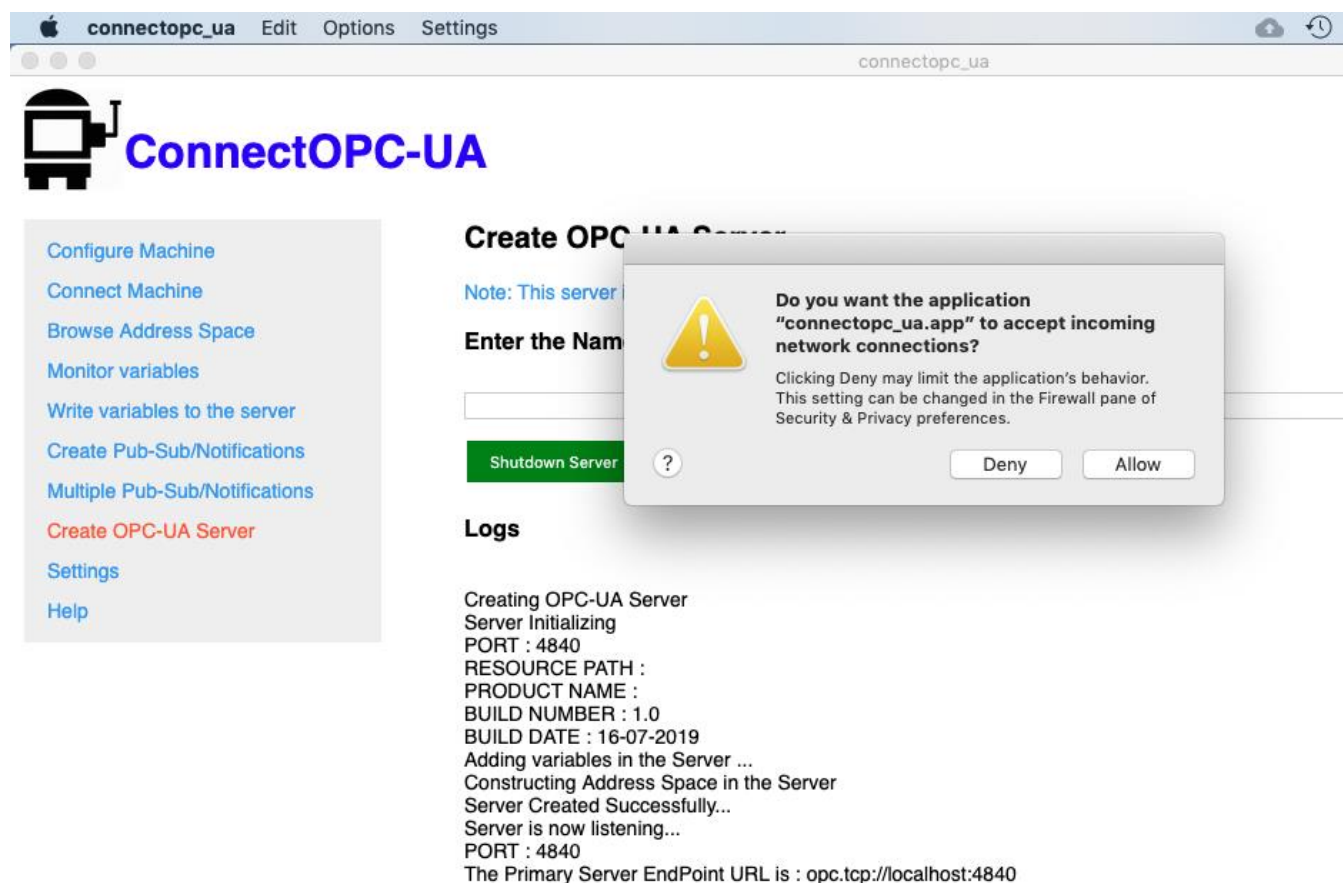


Figure 41 Allow network connection on the specified port and application



Figure 42 Shutdown the Server

4.9 Settings

There are various options of server creation, connecting to the server, session creation and creating a subscription. These parameters are present in the settings with best default settings are available on application load. Fig-43 shows all the settings parameters that are used in the software. Please note any wrong setting of parameter here can change the lifecycle of session creation, and subscriber settings and may malfunction. Fig-43 is showing the best default setting for running the application trouble free. 'Download Folder' setting must be done in the first startup of the application, this is the folder inside which all the downloaded data from the application resides. After any changes to the settings, application will restart itself and all the settings will be effective from that new instance of the application.

Settings

Connection Timeout (in Seconds)	100
Maximum Read/Write/Subscribe Operations	100
Secure Channel Re-authentication Enabled	☒
Failover Enabled	☒
Failover Endpoint	opc.tcp://192.168.1.7:4840
Server Port	4840
Minimum Sampling Interval	100
Downloads Folder	/Users/bhupendramishra/Documents/opcu
Connection Strategy [Maximum Retries]	2
Connection Strategy [Enable Keep Session Alive]	☒
Connection Strategy [Endpoint must exist for session]	☐
Subscribe Settings [Requested Publishing interval]	1000
Subscribe Settings [Requested Lifetime Count]	10
Subscribe Settings [Requested Max Keep Alive Count]	2
Subscribe Settings [Maximum Notification per publish]	10
Subscribe Settings [Publishing Enabled]	☒
Subscribe Settings [Priority]	10
Node Id of parent node (OPCUA Client Only)	ns=0;i=85

Figure 43 Settings Screen

4.10 Help

This section describes the options of OPCUA implemented in this software, along with the details of packages or modules used in the development of this software. Fig-44 shows the screen for the help section.



Figure 44 Help Screen

4.11 Menus

All the software options are also accesible via menu of the application. Fig-45, Fig-46, Fig-47 and Fig-48 shows how to access these options via menus.

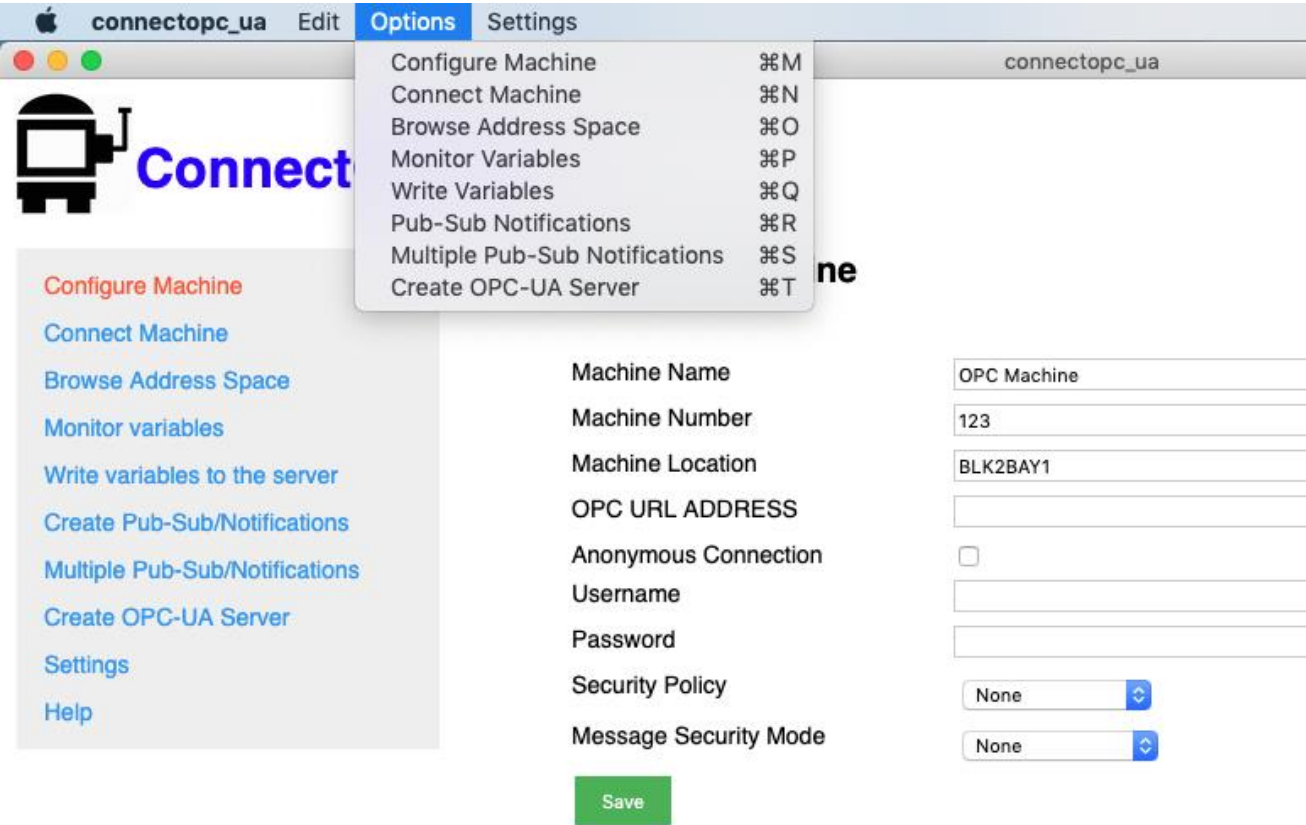


Figure 45 options available via menu

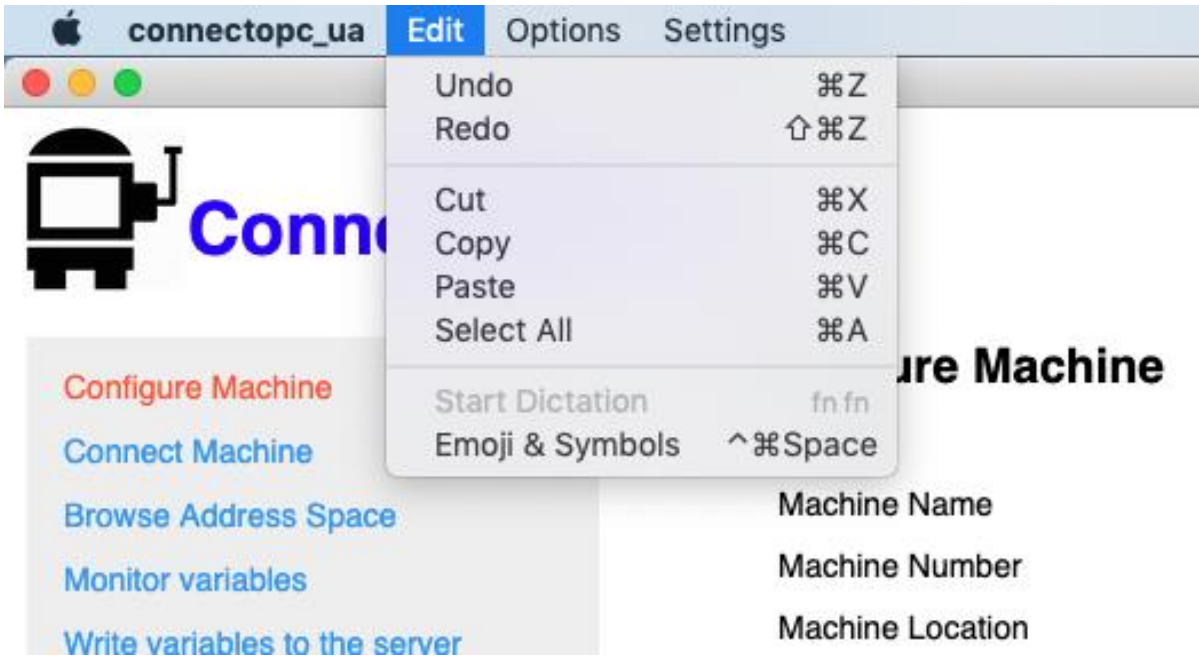


Figure 46 Edit Options via Menu

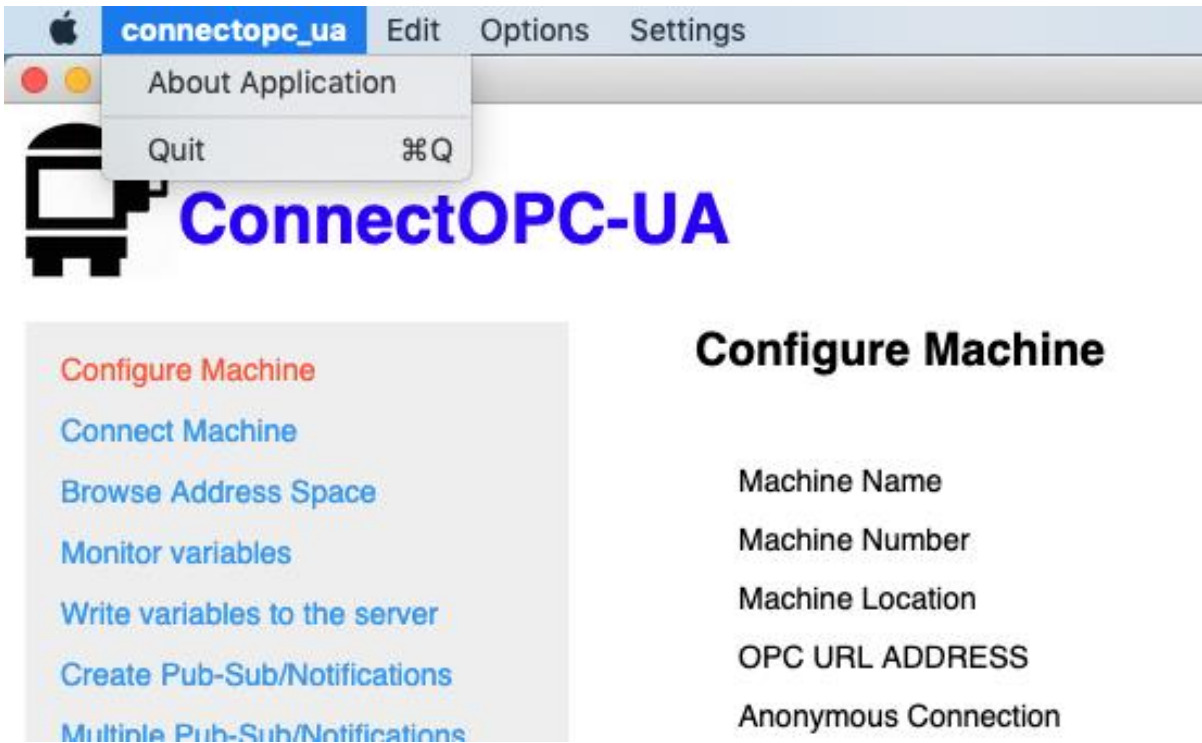


Figure 47 ConnectOPC-UA Options via Menu

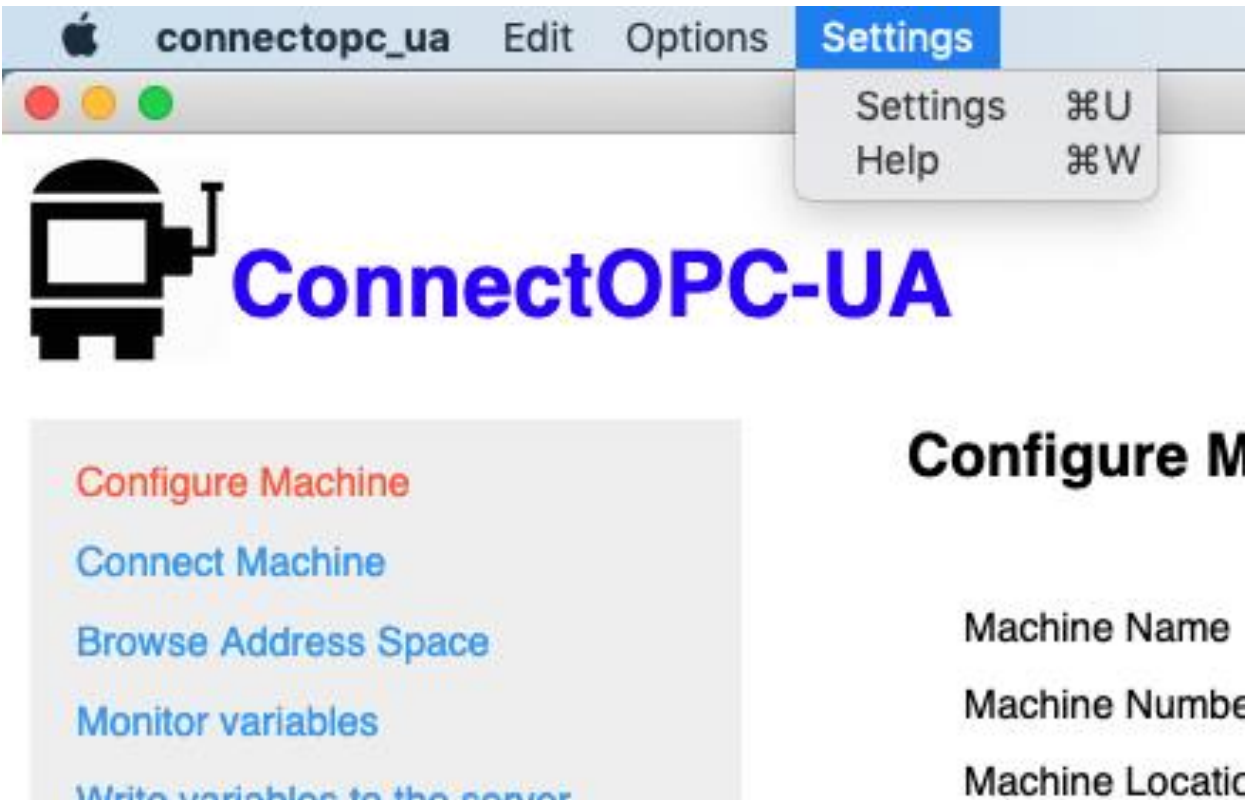


Figure 48 Settings options via Menu

4.12 Method Call

In OPC-UA, Methods represent the function calls of Objects. Methods are called (invoked) and return after completion, whether successful or unsuccessful. Execution times for Methods may vary, depending on the function they are performing. A Method is always a component of an Object.

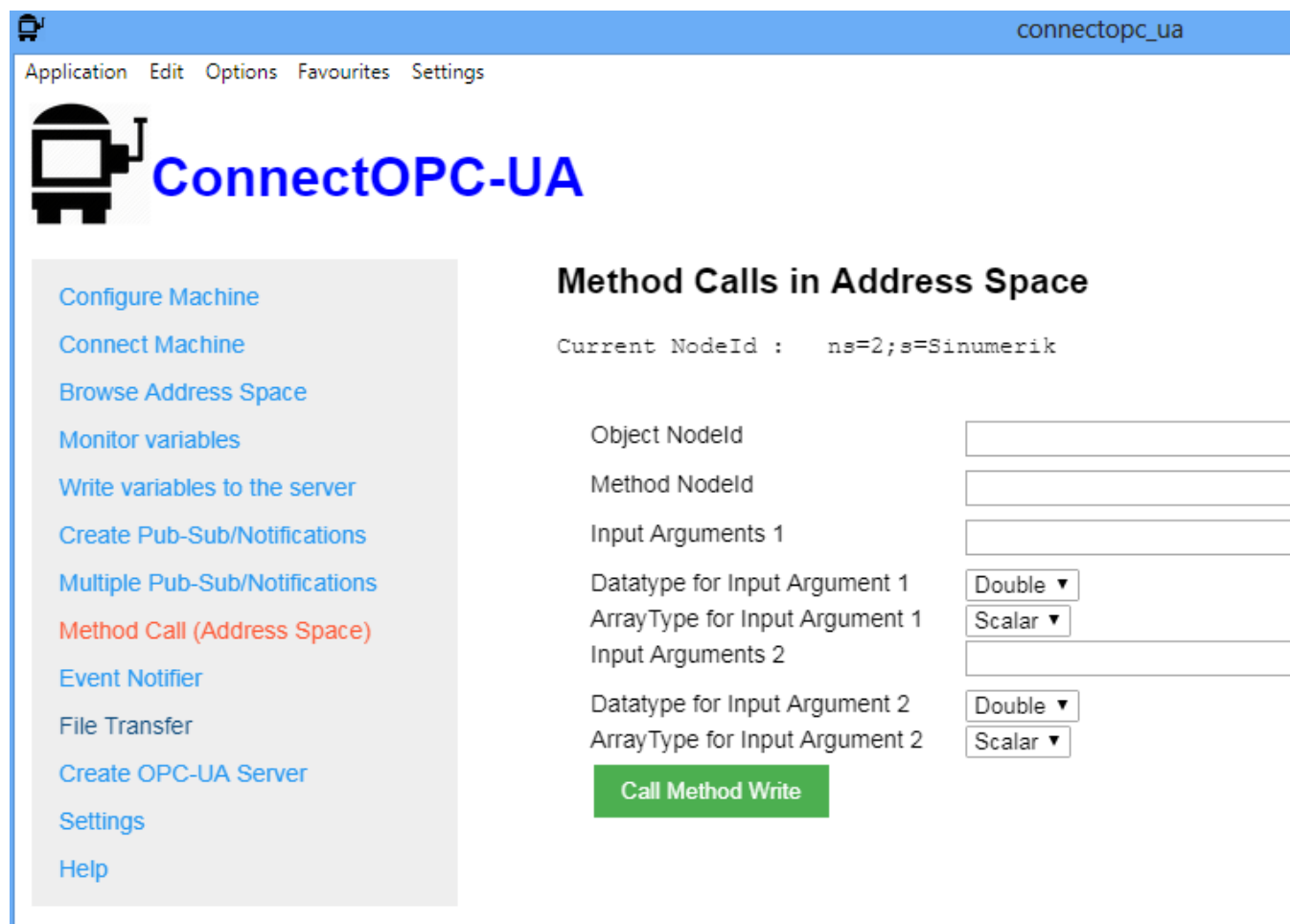


Figure 49 Method Call

The Method Service Set defines the means to invoke Methods. A Method is always a component of an Object. Discovery is provided through the browse and query Services. Clients discover the Methods supported by a Server by browsing for the owning Objects that identify their supported Methods.

Because Methods may control some aspect of plant operations, method invocation may depend on environmental or other conditions. This may be especially true when attempting to re-invoke a Method immediately after it has completed execution. Conditions that are required to invoke the Method may not yet have returned to the state that permits the Method to start again. In addition, some Methods may be capable of supporting concurrent invocations, while others may have a single invocation executing at a given time.

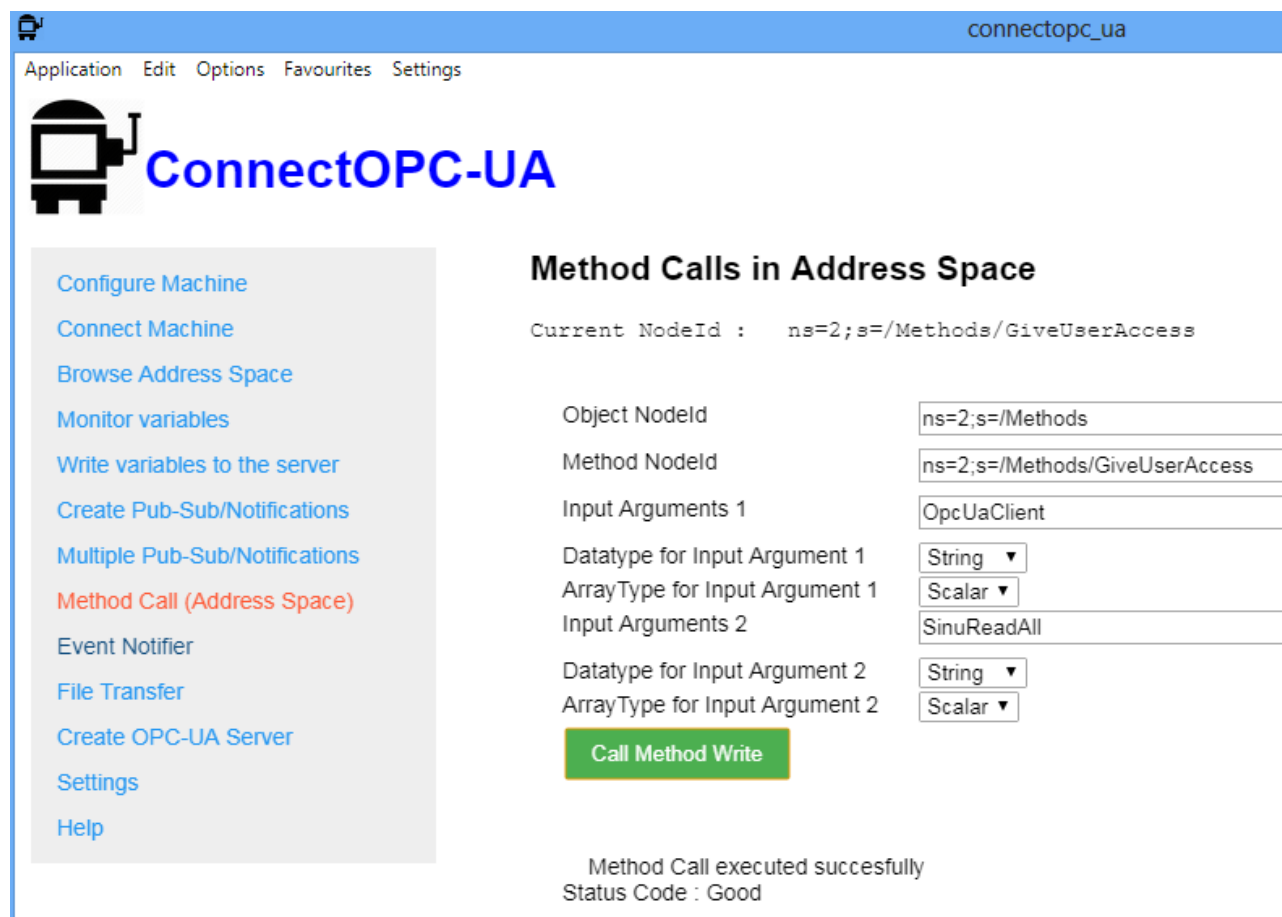


Figure 50 Method Call with 2 Arguments and no output

While Methods may affect the state of the owning Object, they have no explicit state of their own. In this sense, they are stateless. Methods can have a varying number of input arguments and return resultant arguments. Each Method is described by a Node of the Method NodeClass. This Node contains the metadata that identifies the Method's arguments and describes its behavior.

Call is used to call (invoke) a list of Methods. This Service provides for passing input and output arguments to/from a Method. These arguments are defined by Properties of the Method. If the Method is invoked in the context of a Session and the Session is terminated, the results of the Method's execution cannot be returned to the Client and are discarded. This is independent of the task actually performed at the Server.

The order the operations are processed in the Server is not defined and depends on the different tasks and the internal Server logic. If a Method is contained in more than one operation, the order of the processing is undefined. If a Client requires sequential processing the Client needs separate Service calls.

Fig-49 shows the screenshot from the software that shows 2 input arguments, in order to initiate call method, a valid Method call enabled Node-id must be selected which can be identified by browsing the node id as shown in Fig 50. Atleast 1 input arguments must be provided as per the specification of the method provided by the opc-ua server. Type of input arguments required by also be found out by the discovery of the variables by browsing the node-id as shown in Fig-51.

As shown from the Fig-52 and Fig-53, it is clear that N-Root nodeid is a method type taking 2 input arguments, both are integer types, First input argument is the number withose nth (Second Input Agument) root is to be calculated, and the result will be stored in the output

argument is returning the Nth root of the number. Fig-54 shows the successful execution of the method call on N-Root Node-id with 1st Input Argument as 1000 and 2nd Input Argument as 15 and the result shown is the 15th root of 1000. Status Code : Good is shown indicating that the method call is completed successfully without any errors.

Browse OPC-UA Address Space

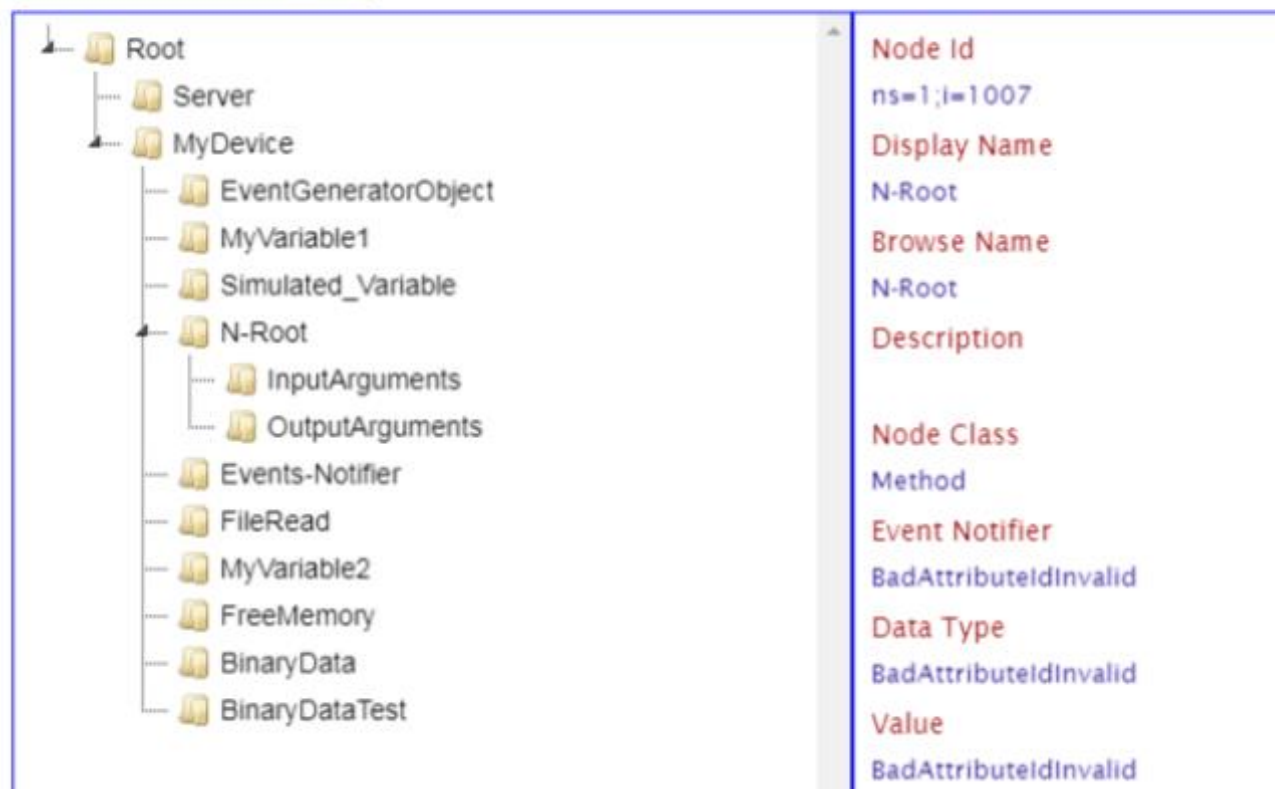
Read Current Node :

ns=1;i=1007

Add to Favourites

ADDRESS SPACE

Download Address Space



Node Id	Display Name	Browse Name	Description	Node Class	Event Notifier	Data Type	Value
ns=1;i=1007	N-Root	N-Root		Method	BadAttributeIdInvalid	BadAttributeIdInvalid	BadAttributeIdInvalid

Figure 51 Properties of Node-id showing a Nodeclass Method

```

Value of ns=1;i=1008 is : { /*Argument*/
name /* UAString */: Number
dataType /* NodeId */: ns=0;i=6
valueRank /* Int32 */: -1
arrayDimensions /* UInt32 [] */: [ 0] (l=1)
description /* LocalizedText */: locale=null text=specifies the number whose nth root is to be calculated
};{ /*Argument*/
name /* UAString */: root_level
dataType /* NodeId */: ns=0;i=6
valueRank /* Int32 */: -1
arrayDimensions /* UInt32 [] */: [ 0] (l=1)
description /* LocalizedText */: locale=null text=specifies the root level, example 2 is square root 3 is cube root
};
Server Time Stamp : Wed Aug 21 2019 14:13:15 GMT+0530 (India Standard Time)

```

Figure 52 Showing the Description and datatype of Input Arguments

ns=1;i=1009

Add to Favourites

```

Value of ns=1;i=1009 is : { /*Argument*/
name /* UAString */: Nth-root
dataType /* NodeId */: ns=0;i=11
valueRank /* Int32 */: 1
arrayDimensions /* UInt32 [] */: [ 0] (l=1)
description /* LocalizedText */: locale=null text=Nth root of a number
};
Server Time Stamp : Wed Aug 21 2019 14:13:15 GMT+0530 (India Standard Time)

```

Figure 53 Showing the Description and datatype of Output Argument

Similarly Fig-50 shows the method call implemented in Siemens 828D CNC Controller where Object Id is ns=2;s=/Methods and Method Node Id is ns=2;s=/Methods/GiveUserAccess. Here Both the input Arguments are of String type, 1st Input Argument is the Username and 2nd input argument is the access right realm input that should be given to the user. In the Fig-50 'OpcUaClient' is the username and 'SinuReadAll' is the permission to read all the nodes in the address space. No output argument is generated here, but the StatusCode :Good indicates the successful execution of the method call.

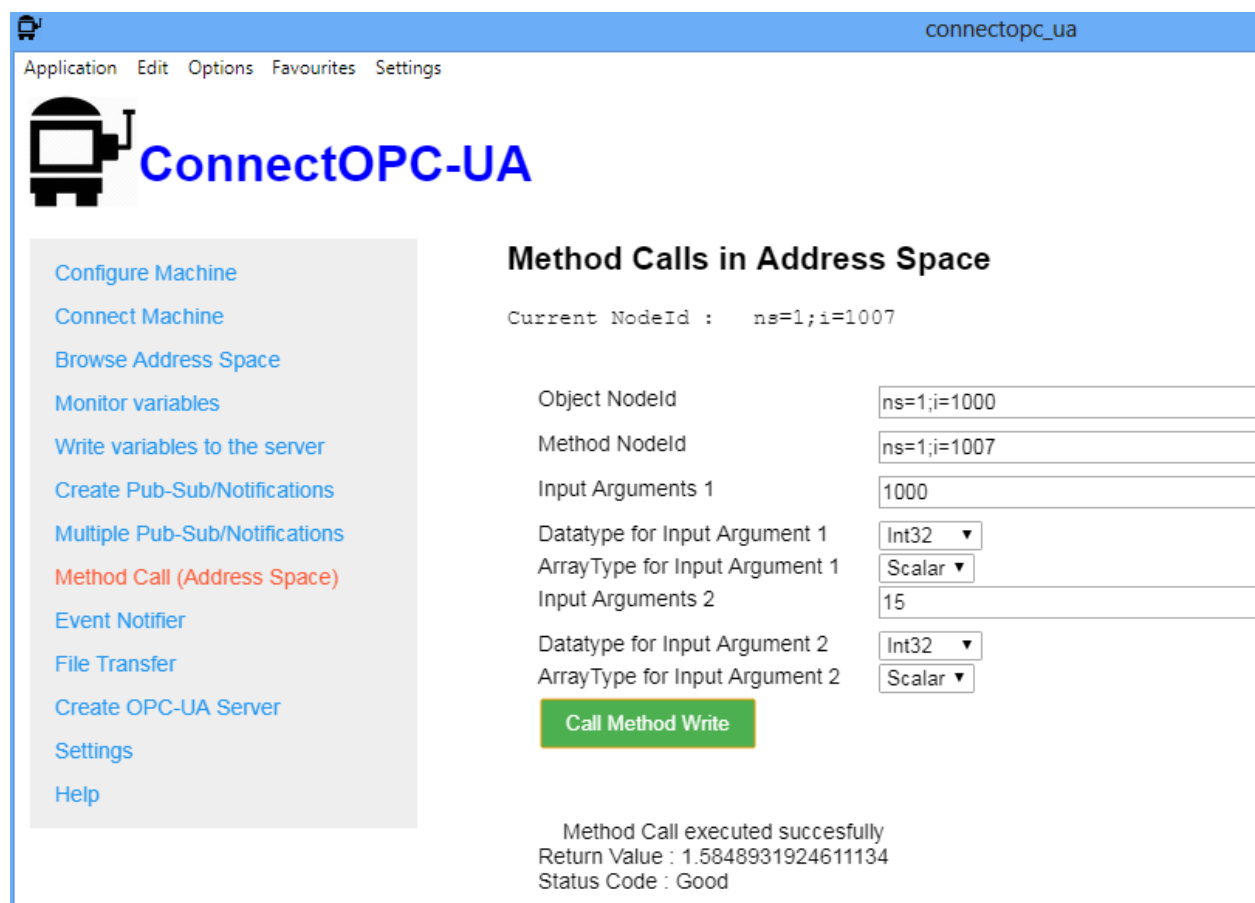


Figure 54 Method call with 2 Input Arguments and 1 Output Argument

4.13 Event Notifier

Conditions are used to represent the state of a system or one of its components. Some common examples are:

- A temperature exceeding a configured limit
- A device needing maintenance
- A batch process that requires a user to confirm some step in the process before proceeding

Each Condition instance is of a specific ConditionType. The ConditionType and derived types are sub-types of the BaseEventType (see Part 3 and Part 5). This part defines types that are common across many industries. It is expected that vendors or other standardization groups will define additional ConditionTypes deriving from the common base types defined in this part. The ConditionTypes supported by a Server are exposed in the AddressSpace of the Server.

Condition instances are specific implementations of a ConditionType. It is up to the Server whether such instances are also exposed in the Server's AddressSpace. Clause 4.10 provides additional background about Condition instances. Condition instances shall have a unique identifier to differentiate them from other instances. This is independent of whether they are exposed in the AddressSpace.

As mentioned above, Conditions represent the state of a system or one of its components. In certain cases, however, previous states that still need attention also have to be maintained. ConditionBranches are introduced to deal with this requirement and distinguish current state

and previous states. Each ConditionBranch has a BranchId that differentiates it from other branches of the same Condition instance. The ConditionBranch which represents the current state of the Condition (the trunk) has a NULL BranchId. Servers can generate separate Event Notifications for each branch. When the state represented by a ConditionBranch does not need further attention, a final Event Notification for this branch will have the Retain Property set to False. Clause 4.4 provides more information and use cases. Maintaining previous states and therefore the support of multiple branches is optional for Servers.

Conceptually, the lifetime of the Condition instance is independent of its state. However, Servers may provide access to Condition instances only while ConditionBranches exist.

The base Condition state model is illustrated in Figure-55. It is extended by the various Condition subtypes defined in this standard and may be further extended by vendors or other standardization groups. The primary states of a Condition are disabled and enabled. The Disabled state is intended to allow Conditions to be turned off at the Server or below the Server (in a device or some underlying system). The Enabled state is normally extended with the addition of sub-states.

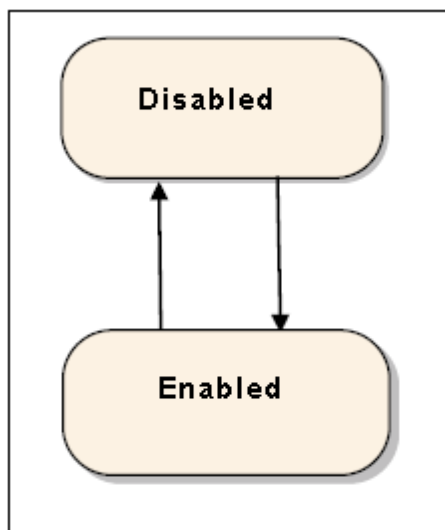


Figure 55 Base Condition state model

A transition into the Disabled state results in a Condition Event however no subsequent Event Notifications are generated until the Condition returns to the Enabled state.

When a Condition enters the Enabled state, that transition and all subsequent transitions result in Condition Events being generated by the Server.

If Auditing is supported by a Server, the following Auditing related action shall be performed. The Server will generate AuditEvents for Enable and Disable operations (either through a Method call or some Server / vendor – specific means), rather than generating an AuditEvent.

AcknowledgeableConditions are sub-types of the base ConditionType. AcknowledgeableConditions expose states to indicate whether a Condition has to be acknowledged or confirmed.

An AckedState and a ConfirmedState extend the EnabledState defined by the Condition. The state model is illustrated in Figure-56. The enabled state is extended by adding the AckedState and (optionally) the ConfirmedState.

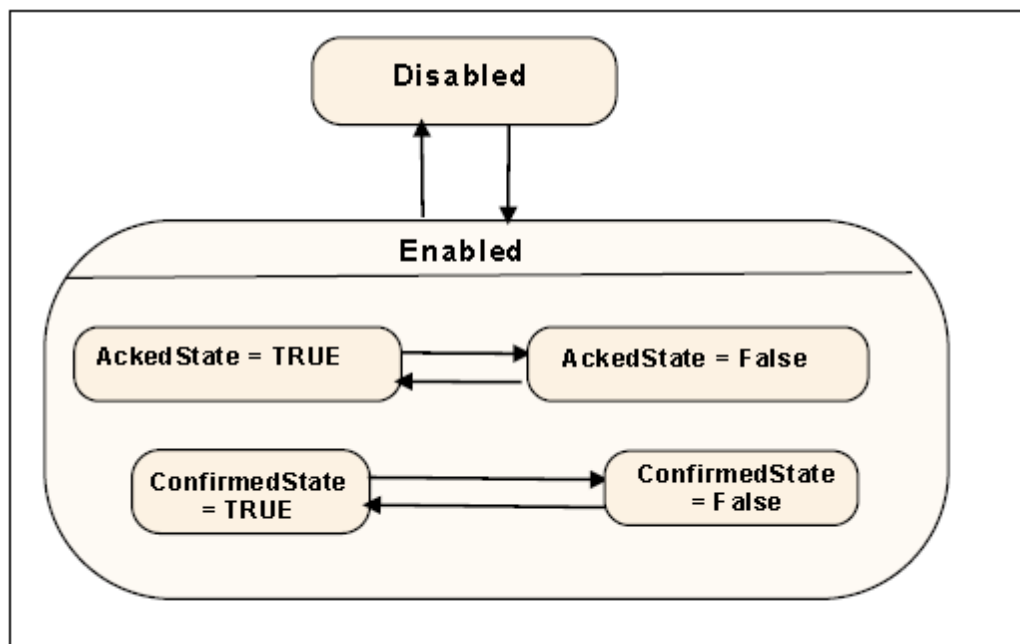


Figure 56 AcknowledgeableConditions State Model

Acknowledgment of the transition may come from the Client or may be due to some logic internal to the Server. For example, acknowledgment of a related Condition may result in this Condition becoming acknowledged, or the Condition may be set up to automatically acknowledge itself when the acknowledgeable situation disappears.

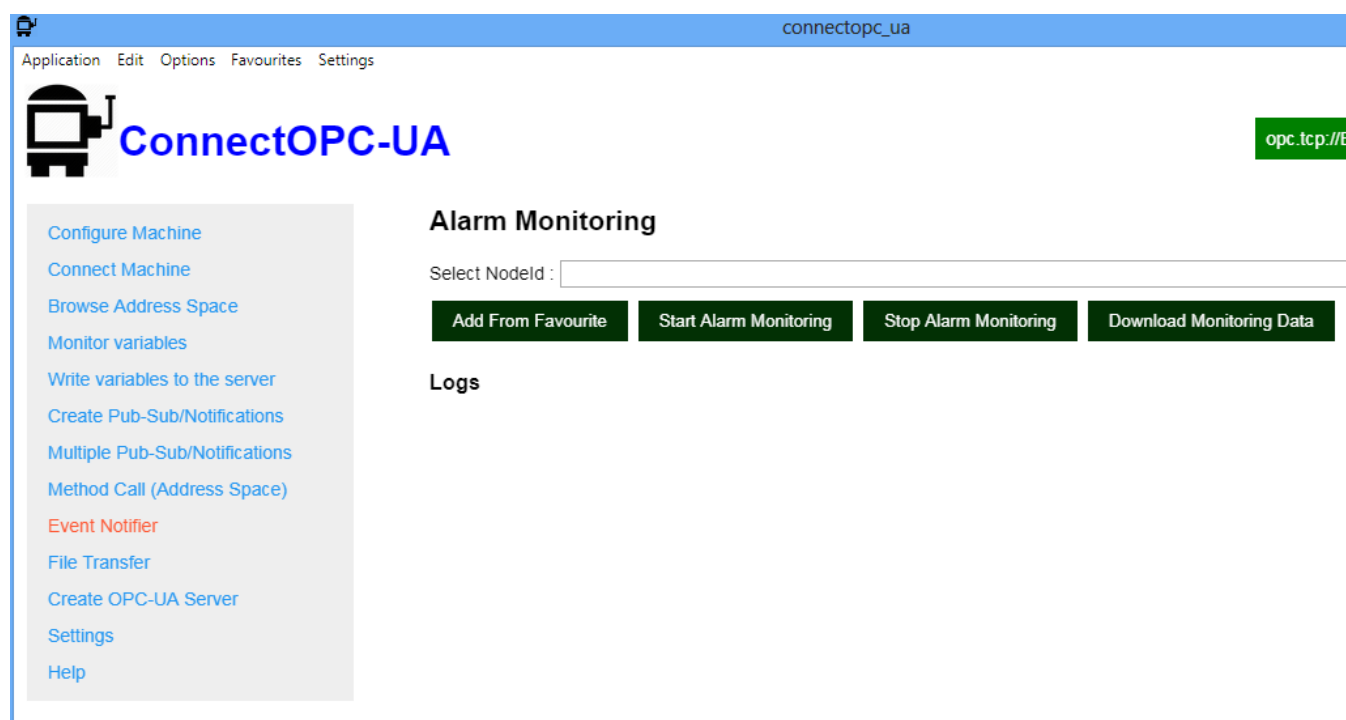


Figure 57 Event Notifier

Alarms and conditions are usually provided in the 'Server' Object, but it may be provided in any node available, user must refer to the server specifications for listening events/alarms/conditions. Fig-__ shows the alarm monitoring in the Demo device on the server node. Selecting 'Stop Alarm Monitoring' will stop the event notifier and any changes in

variable in this duration will not be monitored. 'Download Monitoring Data' is used for downloading the complete Alarm with timestamps for a particular session. If session becomes invalid and application instance is still valid, then creating a new session will again start the monitoring automatically and the previous data will also be available for the downloading.

Alarm Monitoring

Select NodeId :

Add From Favourite

Start Alarm Monitoring

Stop Alarm Monitoring

Download Monitoring Data

Logs

```

locale=null text=The alarm was deactivated by the system. &#x000D;N&#x000A;j Wed Aug 21 2019 19:52:56 GMT+0530 (India Standard Time) Wed
Aug 21 2019 19:52:56 GMT+0530 (India Standard Time) ns=2;s=1:Metals/WestTank?Gold
locale=null text=The alarm was deactivated by the system. &#x000A;* O&#x000A;&#x000A; Wed Aug 21 2019 19:52:56 GMT+0530 (India Standard Time) Wed
Aug 21 2019 19:52:56 GMT+0530 (India Standard Time) ns=2;s=1:Colours/NorthMotor?Yellow
locale=null text=The alarm was deactivated by the system. ^E-^0@&#x000A;[HILJJ&#x000A; Wed Aug 21 2019 19:52:56 GMT+0530 (India Standard Time) Wed
Aug 21 2019 19:52:56 GMT+0530 (India Standard Time) ns=2;s=1:Colours/EastTank?Green
locale=null text=Events Raised &#x000A;&#x000A;&#x000A;O&#x000A;(BY&#x000A;7&#x000A; Wed Aug 21 2019 19:52:56 GMT+0530 (India Standard Time) Wed Aug 21 2019 19:52:56
GMT+0530 (India Standard Time) null
locale=null text=Raising Events &#x000A;&#x000A;
^E&#x000A;Co&#x000A;y Wed Aug 21 2019 19:52:56 GMT+0530 (India Standard Time) Wed Aug 21 2019 19:52:56 GMT+0530 (India Standard Time) null
locale=null text=Events Raised &#x000A;&#x000A;LK&#x000A;&#x000A;&#x000A;a Wed Aug 21 2019 19:52:55 GMT+0530 (India Standard Time) Wed Aug 21 2019 19:52:55
GMT+0530 (India Standard Time) null
locale=null text=Raising Events z ,&#x000A;3-&#x000A;@&#x000A;L&#x000A;&#x000A; Wed Aug 21 2019 19:52:55 GMT+0530 (India Standard Time) Wed Aug 21 2019 19:52:55
GMT+0530 (India Standard Time) null
locale=null text=The alarm is active. &#x000A;&#x000A;I @&#x000A;I&#x000A;-&#x000A;&#x000A; Wed Aug 21 2019 19:52:54 GMT+0530 (India Standard Time) Wed Aug 21 2019
19:52:54 GMT+0530 (India Standard Time) ns=2;s=1:Metals/SouthMotor?Silver

```

Figure 58 Alarm Monitoring in the demo server on the 'server' node

4.14 File Transfer

BLOB (Binary Large Object Block) – "BLOBs" provide a way to transfer data that has no OPC UA data definition. Normally, all OPC UA data is referenced by some sort of data definition which explains the data format. BLOB data is used when the application wishes to transfer data that has no OPC UA definition. BLOB data is user defined and could be anything: video, audio, data files or anything else.

In OPC UA, servers can stick to the standard types defined in the OPC UA specifications, or they can define their own types. The "non-standard" data types can be completely server-specific, or can conform to some industry standard. Usage of these data types is commonly referred to as OPC UA complex data. With complex data, it is possible that the data type is not even known at compile time; instead, it can be discovered, constructed or processed during runtime, without prior knowledge of it.

Use of OPC UA complex data poses a new range of interoperability problems. When the server uses some new data type, how can the client know what the data type is, and how is the data encoded? The data is basically a binary blob that is transferred as a block between the server and the client, and who knows what it means? In some cases, the client can be hard-coded with upfront knowledge of what this binary blob of data (called extension object) contains and how it is encoded. But that is only when the client can be written with specific knowledge about the server it connects to, and even then, decoding of the data might be a tedious programming task. Thankfully, OPC UA specifications provide ways for servers to describe the data types and their encoding to the clients, using information that can reside in the server itself, using so-called data type dictionaries. Client can then extract

the data type descriptions from these dictionaries, and be written in a generic way that adjusts to the data types that exist in the server.

Retrieving the associated metadata from the server's information model, interpreting the data type dictionaries, and decoding and encoding data according to data type descriptions is quite complicated. ConnectOPC-UA can do it for you, with the help of the OPC UA Complex Data extension, described in this section. With the OPC UA Complex Data extension, you do not have to deal with binary blobs; instead, you work with meaningful structures that represent the data types and the data itself logically.

Read File From the Machine (Server)

Variable Name (Full Qualified Node Id)

Add From Favourite

Read

Download File

Write File to the Machine (Server)

Variable Name (Full Qualified Node Id)

Add From Favourite

Write

Upload File

Figure 59 File Transfer

Fig-59 shows the screenshot of the File transfer function. 'FileType' node id is needed for reading/writing data from/to the server. While reading the file from the server, file can be downloaded using 'Download File' option. While writing the file to the server, file to be written must be uploaded in the software using 'Upload File' function. File transfer is particularly useful in transferring the NC Part Programs remotely in the CNC Controllers using OPC-UA Servers. A Sample OPCUA Server implemented in the software for the diagnostics purpose have a Node Id by the Description 'FileRead' which will read the file 'read.txt' from the server and presented in the screen. 'FileRead' node id is shown in the Fig-60. Fig-61 shows the content of the file from the node id. Writing of the File is not yet tested but can be implemented in the same way.

Browse OPC-UA Address Space

Read Current Node :

ns=1;s=File_Read

Read

Add to Favourites

ADDRESS SPACE

Download Address Space

Root

Server

MyDevice

EventGeneratorObject

MyVariable1

Simulated_Variable

N-Root

Events-Notifier

FileRead

MyVariable2

FreeMemory

BinaryData

BinaryDataTest

Node Id

Display Name

Browse Name

Description

Node Class

Event Notifier

Data Type

Value

Access Level

Historizing

User Access Level

Write Mask

Value Rank

ns=1;s=File_Read

FileRead

FileRead

Read File

Variable

BadAttributeldInvalid

ByteString

BadInternalError

3

false

3

0

-1

Figure 60 Browsing FileRead Node Id

Read Variables

Variable Name (Full Qualified Node Id)

ns=1;s=File_Read

Add From Favourite

Read

Logs

Value of ns=1;s=File_Read is : OPC Unified Architecture (OPC UA) is a machine to machine communication protocol for industrial automation developed by the OPC Foundation. Distinguishing characteristics are:

- 1) Focus on communicating with industrial equipment and systems for data collection and control
- 2) Open - freely available and implementable under GPL 2.0 license [1]
- 3) Cross-platform - not tied to one operating system or programming language
- 4) Service-oriented architecture (SOA)
- 5) Inherent complexity - the specification consists of 1250 pages in 14 documents
- 6) Robust security
- 7) Integral information model, which is the foundation of the infrastructure necessary for information integration where vendors and organizations can model their complex data into an OPC UA namespace to take advantage of the rich service-oriented architecture of OPC UA.
- 8) There are over 35 collaborations with the OPC Foundation currently. Key industries include pharmaceutical, oil and gas, building automation, industrial robotics, security, manufacturing and process control.

Server Time Stamp : Wed Aug 21 2019 21:22:00 GMT+0530 (India Standard Time)

Figure 61 Read File from the FileRead Node Id

4.15 Favourites

It is very tedious to find the node id from large address space, for example Siemens 828D CNC Controllers has address space of more than 10000 nodes. It is very important to make a list of important node id(s), so the smaller list can be used for monitoring, subscribing, writing and transferring the files.

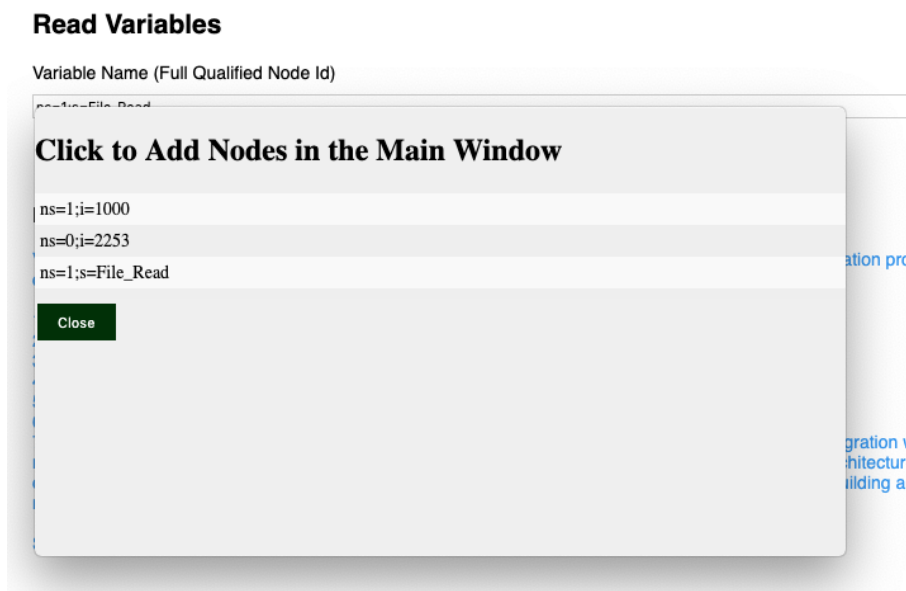


Figure 62 Favourites Window

'Favourite Window' can be invoked in any function using the command 'Add from Favourites'. Node id to be selected must be clicked in the favourite window and favourite window will be closed automatically and the node id is automatically selected in the required input field. Node id can be added to the favourites using the 'Add to Favourites' option which is available in 'Browse Address Space' Option as Shown in the Fig-64.

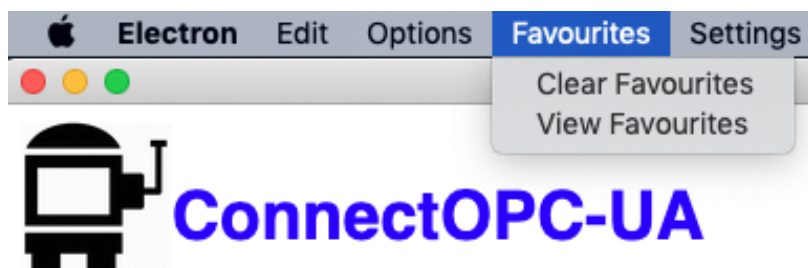


Figure 63 Menu for favourites

All the added favourites can be viewed separately for a selected configured machine will be shown in the Menu Option 'View Favourites'. All the favourites can be cleared using 'Clear Favourites'. Please note that the user cannot the old favourites list after the clear option, and used need to reselect all the node ids one by one again.

Browse OPC-UA Address Space

Read Current Node :

Read

Add to Favourites

ADDRESS SPACE

Download Address Space

Root

- Server
 - MyDevice
 - EventGeneratorObject
 - MyVariable1

Node Id

Display Name

Browse Name

Description

Node Class

ns=0;i=85

Objects

Objects

The browse entry point when looking for c

Object

Figure 64 Add to Favourites Option

Favourites are managed for a particular machine/device, i.e. the favourites added by the user are available for that particular configuration of the machine. So the newly configured machine, initially the favourites are empty and user has to make the fresh selections for it. Duplicate node id cannot be added to the favourites and the confirmation of the same is provided to the user.

A

Node Id : ns=1;i=1006 is added to favourites

OK

Browse OPC-UA Address Space

Read Current Node :

Add to Favourites

Figure 65 Adding Node to the favourites list

This Node Id is already added in Favorites

OK

owse OPC-UA Address Space

ad Current Node :

Add to Favourites

Figure 66 Rejection of Duplicate Node Id in Favourites

4.16 Node Crawler

Node Crawler provides better and efficient way of browsing node. Server can made serious restriction on number of nodes that can be browsed in a single instance and provide the continuation points for the further browsing from the same position where it left earlier. Conventional browse method will browse the maximum number of nodes as restricted by the server and rest of the node will not be browsed. Node Crawler can be invoked by clicking on the 'Menu Bar → Options' and then 'Node Crawler'. Fig-67 Shows the option Node Crawler.

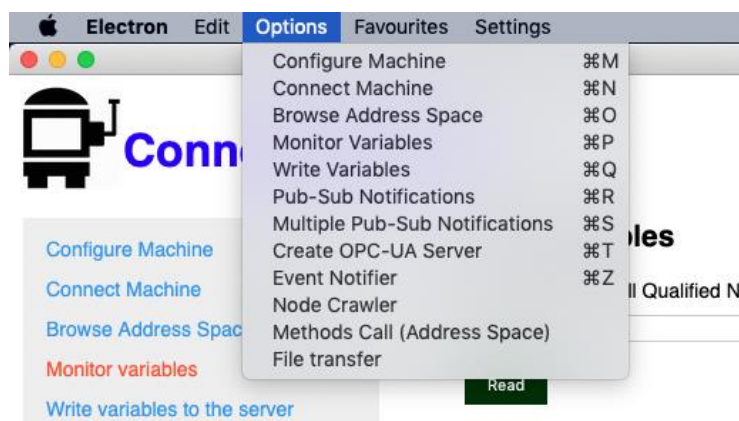


Figure 67 Node Crawler Option

For Example, Siemens 828D OPCUA Server provides 50 Maximum number of nodes that can be browsed in a Single instance, but in parent nodes more than 500 nodes are present. So, in order to browse the complete node continuation points must be used. In Connect OPC-UA, a maximum of 2500 nodes can be read in a single instance which is practically very high value, which ensures no node id can be left while browsing the particular node. Use of continuation points are shown in the Fig-68

```

}
results                                     /* BrowseResult      [] */: [
{ /*0*/
  statusCode                               /* StatusCode          */: Good (0x000000)
  continuationPoint                       /* ByteString           */
  BUFFER{00000000: 09 00 00 40 7b 7b 00 00
  {{..
}

references                                 /* ReferenceDescript[] */: [
{ /*0*/
  referenceTypeId                         /* NodeId               */: ns=0;i=47
  isForward                              /* UABoolean            */: true
  nodeId                                 /* ExpandedNodeId        */: ns=2;s=/Channel/State/protAreaCounter
  browseName                             /* QualifiedName         */: 2:/Channel/State/protAreaCounter
  displayName                             /* LocalizedText         */: locale=en_us text=protAreaCounter
  nodeClass                              /* NodeClass             */: 2 ( 2)
  typeDefinition                         /* ExpandedNodeId        */: ns=0;i=63

```

Figure 68 Use of Continuation Points while browsing large nodes

The view of the Node Crawler is divided into 2 sections , First section shows the address space of the sever, any node clicked will initiate the browsing process and shows the node inside the clicked parent folder, second view shows the attributes of the clicked node. The View Hierarchy of the node is shown in the Fig-69

Browse OPC-UA Address Space

Read Current Node :

ns=1;i=1007

Add to Favourites

ADDRESS SPACE

Download Address Space

BuildDate

SecondsTillShutdown

ShutdownReason

ServiceLevel

Auditing

EstimatedReturnTime

LocalTime

ServerCapabilities

ServerDiagnostics

VendorServerInfo

ServerRedundancy

Namespaces

GetMonitoredItems

ResendData

SetSubscriptionDurable

RequestServerStateChange

Dictionaries

ServerConfiguration

Node Id

Display Name

Browse Name

Description

Node Class

Event Notifier

Data Type

Value

Access Level

Historizing

User Access Level

Write Mask

Value Rank

Access Restrictions

ns=1;i=1007

N-Root

N-Root

Method

BadAttributeInvalid

BadAttributeInvalid

BadAttributeInvalid

BadAttributeInvalid

BadAttributeInvalid

BadAttributeInvalid

BadAttributeInvalid

0

BadAttributeInvalid

undefined

Figure 69 Hierarchy of Node Crawler View

5. Deploying the Application

To distribute your app with Electron, you need to package and rebrand it. The easiest way to do this is to use one of the following third-party packaging tools:

- 1) electron-forge
- 2) electron-builder
- 3) electron-packager

These tools will take care of all the steps you need to take to end up with distributable Electron applications, such as packaging your application, rebranding the executable, setting the right icons and optionally creating installers.

You can also choose to manually get your app ready for distribution. The steps needed to do this are outlined below.

To distribute your app with Electron, you need to download Electron's prebuilt binaries. Next, the folder containing your app should be named `app` and placed in Electron's resources directory as shown in the following examples. Note that the location of Electron's prebuilt binaries is indicated with `electron/` in the examples below.

On macOS:

```
electron/Electron.app/Contents/Resources/app/
├── package.json
├── main.js
└── index.htmlCopy
```

On Windows and Linux:

```
electron/resources/app
├── package.json
├── main.js
└── index.htmlCopy
```

Then execute `Electron.app` (or `electron` on Linux, `electron.exe` on Windows), and Electron will start as your app. The `electron` directory will then be your distribution to deliver to final users.

Apart from shipping your app by copying all of its source files, you can also package your app into an `asar` archive to avoid exposing your app's source code to users.

To use an `asar` archive to replace the `app` folder, you need to rename the archive to `app.asar`, and put it under Electron's resources directory like below, and Electron will then try to read the archive and start from it.

On macOS:

```
electron/Electron.app/Contents/Resources/
└── app.asarCopy
```

On Windows and Linux:

```
electron/resources/
└── app.asar
```

5.1 Deployment using electron-packager

Installing electron-packager

We're going to use a package called *electron-packager* to create the OS-specific builds of our Electron app. We first have to install it at the command line within the project folder:

```
$ npm install electron-packager --save-dev
```

Modifying our Package.json File

The packager uses information such as the name, version, description, etc.. in the `package.json` file in the root directory.

We're also going to create 3 different *scripts*, each for packaging the app for the 3 different operating systems (Win, Mac and Linux).

```
{
  // Other properties removed for brevity

  "scripts": {
    "start": "electron .",
    "package-mac": "electron-packager . --overwrite --platform=darwin --arch=x64 --icon=assets/icons/mac/icon.icns --prune=true --out=release-builds",
    "package-win": "electron-packager . electron-tutorial-app --overwrite --asar=true --platform=win32 --arch=ia32 --icon=assets/icons/win/icon.ico --prune=true --out=release-builds --version-string.CompanyName=CE --version-string.FileDescription=CE --version-string.ProductName=\"CryptoApp\"",
    "package-linux": "electron-packager . electron-tutorial-app --overwrite --asar=true --platform=linux --arch=x64 --icon=assets/icons/png/1024x1024.png --prune=true --out=release-builds"
  },

  // Other properties removed for brevity
}
}
```

Once you've updated your `package.json` file and created the necessary icon files, you're now ready to create a build of your app!

In the console, type:

```
$ npm run package-win (or package-mac, package-linux)
```

If it all goes smoothly, there will be a new folder within your project called `release-builds`.

Fig-70,71,72 shows the successful deployment of the application for Linux, Mac OS and Windows platform.

```
[Bhupendras-MacBook-Pro:opc_ua_machines bhupendramishra$ npm run package-linux

> connectopc_ua@1.0.0 package-linux /Users/bhupendramishra/Documents/projects/nodejs/opc_ua_machines
> electron-packager . ConnectOPC-UA-app --overwrite --platform=linux --arch=x64 --prune=true --out=release-builds

Packaging app for platform linux x64 using electron v5.0.7
Wrote new app to release-builds/ConnectOPC-UA-app-linux-x64
Bhupendras-MacBook-Pro:opc_ua_machines bhupendramishra$ █
```

Figure 70 Deployment for Linux Application

Deploying the Application

```
[Bhupendras-MacBook-Pro:opc_ua_machines bhupendramishra$ npm run package-mac

> connectopc_ua@1.0.0 package-mac /Users/bhupendramishra/Documents/projects/nodejs/opc_ua_machines
> electron-packager . --overwrite --platform=darwin --arch=x64 --prune=true --out=release-builds

Packaging app for platform darwin x64 using electron v5.0.7
Wrote new app to release-builds/connectopc_ua-darwin-x64
Bhupendras-MacBook-Pro:opc_ua_machines bhupendramishra$
```

Figure 71 Deployment for Mac OS Application

```
[Bhupendras-MacBook-Pro:opc_ua_machines bhupendramishra$ npm run package-win

> connectopc_ua@1.0.0 package-win /Users/bhupendramishra/Documents/projects/nodejs/opc_ua_machines
> electron-packager . ConnectOPCUA-app --overwrite --platform=win32 --arch=ia32 --prune=true --out=release-builds

Packaging app for platform win32 ia32 using electron v5.0.7
█
```

Figure 72 Deployment for Windows Application

5.2 Deployment using electron-builder

Create a directory build in the root of the project and save a background.png (macOS DMG background), icon.icns (macOS app icon) and icon.ico (Windows app icon) into it. The Linux icon set will be generated automatically based on the macOS. Ex. zulip-electron/build

Add electron-builder to your app devDependencies by running:

```
npm install electron-builder --save-dev
```

In electron-builder you could use two package.json structure as well as single package.json. We'll be using single package.json structure since it's easier to maintain. Our package.json is shown below where "dist" entry is added to the script. In order to generate binaries, following command should be run in the terminal or command prompt. It must be noted that it only generate binaries of the platform on which it is running.

```
npm run dist
"scripts": {
  "start": "electron .",
  "pack": "electron-builder --dir",
  "package-mac": "electron-packager . --overwrite --platform=darwin --arch=x64 --prune=true --out=release-builds",
  "package-win": "electron-packager . ConnectOPCUA-app --overwrite --platform=win32 --arch=ia32 --prune=true --out=release-builds",
  "package-linux": "electron-packager . ConnectOPCUA-app --overwrite --platform=linux --arch=x64 --prune=true --out=release-builds",
  "dist": "electron-builder"
},
```

Deploying the Application

```
[Bhupendras-MacBook-Pro:opc_ua_machines bhupendramishra$ npm run dist

> connectopc_ua@1.0.0 dist /Users/bhupendramishra/Documents/projects/nodejs/opc_ua_machines
> electron-builder

  • electron-builder version=21.0.15 os=18.6.0
  • loaded configuration file=package.json ("build" field)
  • writing effective config file=dist/builder-effective-config.yaml
  • rebuilding native dependencies dependencies=deasync@0.1.15 platform=darwin arch=x64
  • packaging platform=darwin arch=x64 electron=5.0.7 appOutDir=dist/mac
  • skipped macOS application code signing reason=cannot find valid "Developer ID Application" signing allIdentities=
    1) D69EB92A0FC8BD96C5913DB11EB4565A3D718D5E
ERR_TP_NOT_TRUSTED)
    1 identities found
    Valid identities only
    0 valid identities found
  • building target=macOS zip arch=x64 file=dist/connectopc_ua-1.0.0-mac.zip
  • building target=DMG arch=x64 file=dist/connectopc_ua-1.0.0.dmg
  • building block map blockMapFile=dist/connectopc_ua-1.0.0.dmg.blockmap
  • building embedded block map file=dist/connectopc_ua-1.0.0-mac.zip
Bhupendras-MacBook-Pro:opc_ua_machines bhupendramishra$
```

Figure 73 Deployment of Electron App on Mac OS using electron-builder

```
> connectopc_ua@0.0.1 dist E:\nodejs\opc_ua_machines
> electron-builder

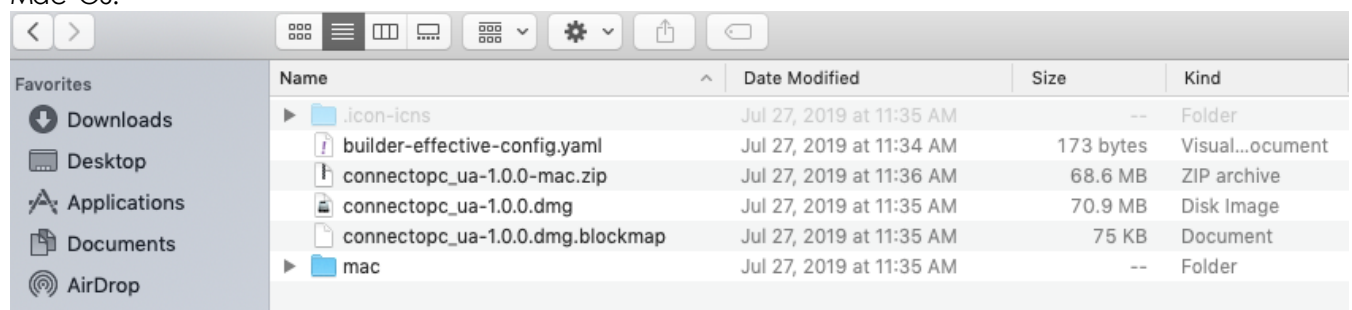
  • electron-builder version=21.1.1 os=10.0.10586
  • loaded configuration file=package.json ("build" field)
  • writing effective config file=dist\builder-effective-config.yaml
  • rebuilding native dependencies dependencies=deasync@0.1.15 platform=win32 arch=x64
  • packaging platform=win32 arch=x64 electron=5.0.7 appOutDir=dist\win-unpacked
  • building target=nsis file=dist\connectopc_ua Setup 0.0.1.exe archs=x64 oneClick=true perFile
  • building block map blockMapFile=dist\connectopc_ua Setup 0.0.1.exe.blockmap

E:\nodejs\opc_ua_machines>
```

Figure 74 Deployment of Electron App in Windows 10 (64 Bit) using electron-builder

5.3 Installing software on Mac OS

Following are the files present for the binaries that is required for the installation of Software in Mac OS.



Name	Date Modified	Size	Kind
▶ .icon-icns	Jul 27, 2019 at 11:35 AM	--	Folder
▶ builder-effective-config.yaml	Jul 27, 2019 at 11:34 AM	173 bytes	Visual...ocument
▶ connectopc_ua-1.0.0-mac.zip	Jul 27, 2019 at 11:36 AM	68.6 MB	ZIP archive
▶ connectopc_ua-1.0.0.dmg	Jul 27, 2019 at 11:35 AM	70.9 MB	Disk Image
▶ connectopc_ua-1.0.0.dmg.blockmap	Jul 27, 2019 at 11:35 AM	75 KB	Document
▶ mac	Jul 27, 2019 at 11:35 AM	--	Folder

Figure 75 Files of binaries in Mac OS

For the installation, we have to double click on "connectopc_ua-1.0.0.dmg". Screen as shown in Fig-54 will appear, now just drag the connectopc_ua icon to Applications to move

Deploying the Application

the software in Applications directory. Installed application can be found in Launchpad of the MacOS as shown in Fig-55

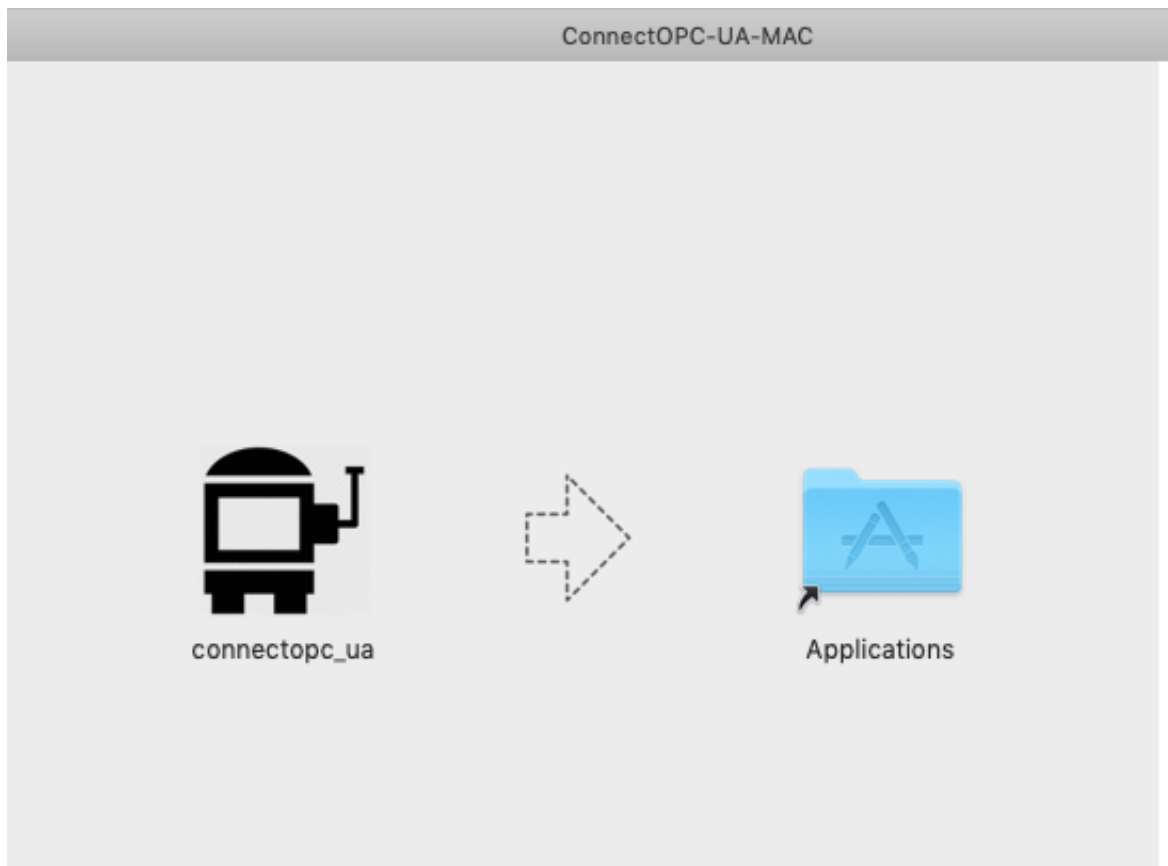


Figure 76 Installation of software in Mac

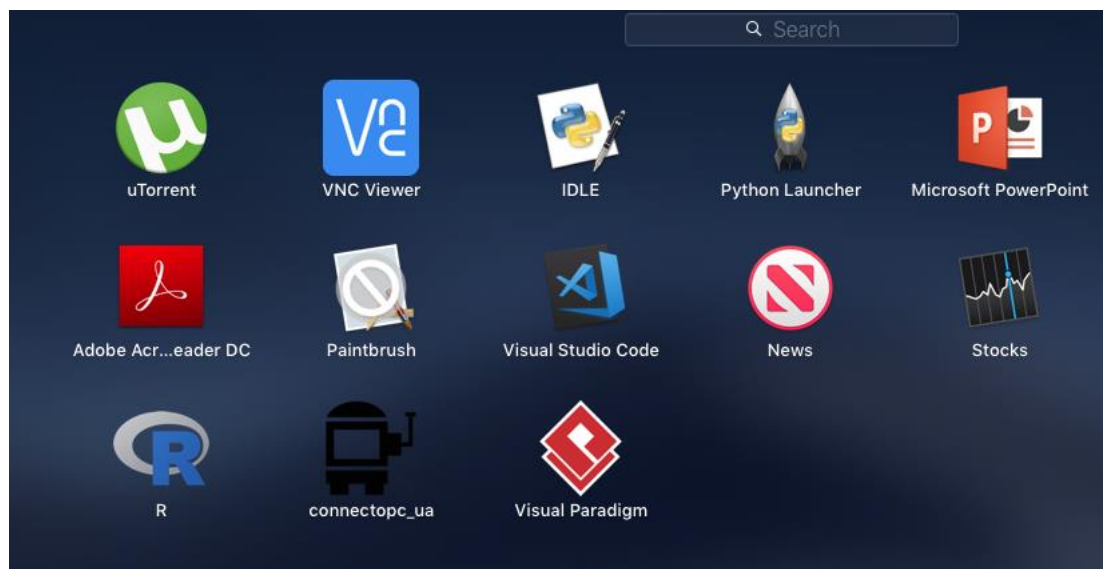
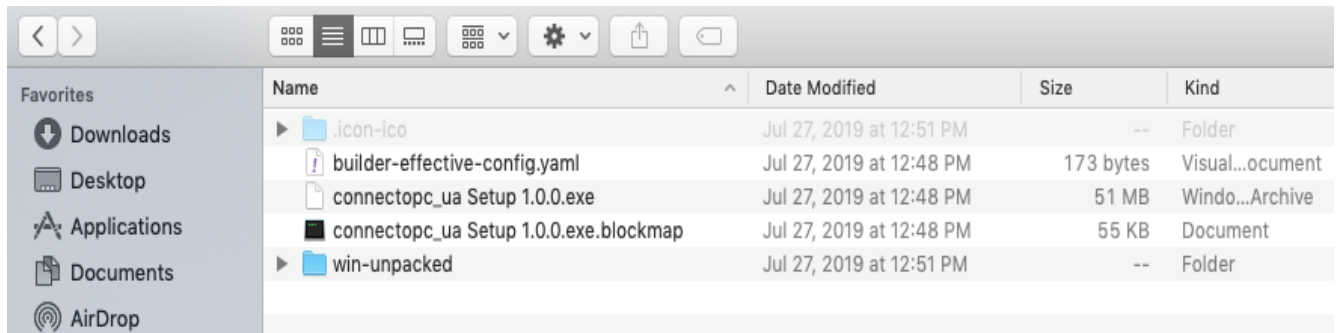


Figure 77 Connectopc-ua Application in Launchpad in MacOS

5.4 Installing Software in Windows

Following are the files present for the binaries that is required for the installation of Software in Windows 10 (64 bit Environment).



Name	Date Modified	Size	Kind
▶ .icon-ico	Jul 27, 2019 at 12:51 PM	--	Folder
builder-effective-config.yaml	Jul 27, 2019 at 12:48 PM	173 bytes	Visual...ocument
connectopc_ua Setup 1.0.0.exe	Jul 27, 2019 at 12:48 PM	51 MB	Windo...Archive
connectopc_ua Setup 1.0.0.exe.blockmap	Jul 27, 2019 at 12:48 PM	55 KB	Document
▶ win-unpacked	Jul 27, 2019 at 12:51 PM	--	Folder

Figure 78 Installation Files of Software in Windows (64 bit)

For the installation, we have to double click on "connectopc_ua Setup 1.0.0.exe". Screen as shown in Fig-56 will appear, after the successful installation, application can be opened from the shortcut available at Desktop or Application is also available from "All Apps" as shown in Fig-57.

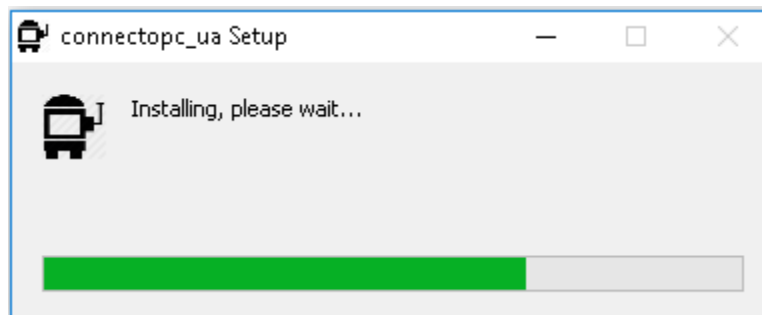


Figure 79 Installation of Software in Windows 10 (64 Bit)

Deploying the Application

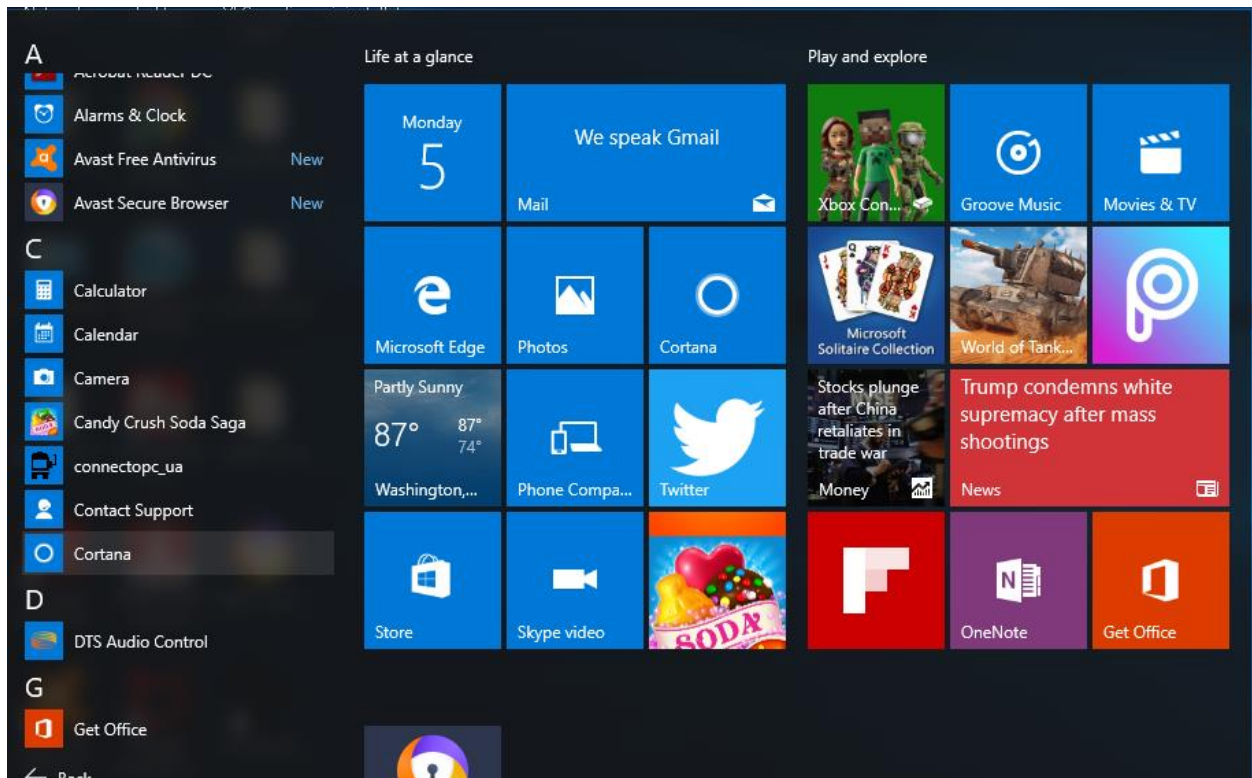


Figure 80 Listing of Connectopc_ua in All Apps of Windows 10 (64 bit)

Uninstall or change a program

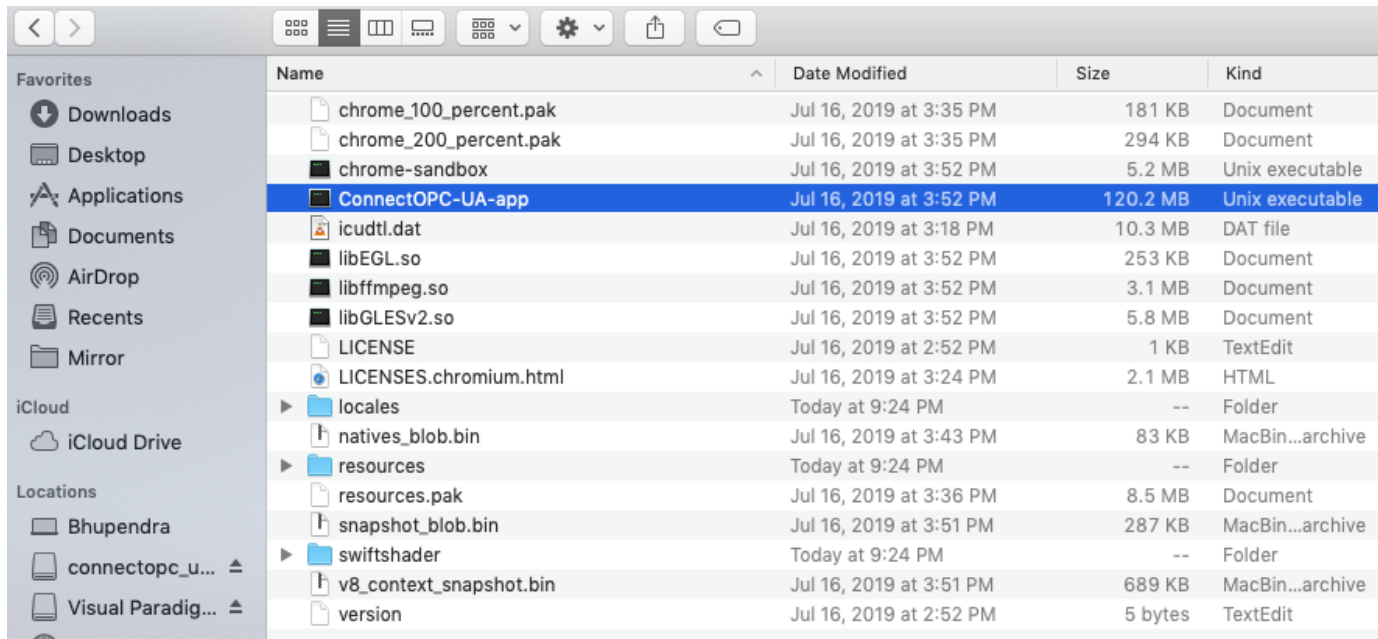
To uninstall a program, select it from the list and then click Uninstall, Change, or Repair.

Organize ▾				
Name	Publisher	Installed On	Size	Version
7-Zip 17.01 beta	Igor Pavlov	13-11-2017	3.53 MB	17.01 beta
Adobe Reader 9.3.4	Adobe Systems Incorporated	05-04-2015	211 MB	9.3.4
Apple Mobile Device Support	Apple Inc.	18-09-2015	28.0 MB	9.0.0.26
Apple Software Update	Apple Inc.	18-09-2015	2.39 MB	2.1.4.131
Avast Free Antivirus	AVAST Software	20-05-2019		19.5.2378
Bonjour	Apple Inc.	18-09-2015	2.05 MB	3.1.0.1
Cisco EAP-FAST Module	Cisco Systems, Inc.	27-11-2018	1.53 MB	2.2.14
Cisco LEAP Module	Cisco Systems, Inc.	27-11-2018	632 KB	1.0.19
Cisco PEAP Module	Cisco Systems, Inc.	27-11-2018	1.22 MB	1.1.6
connectopc_ua 1.0.0	Bhupendra Kumar Mishra	22-08-2019	264 MB	1.0.0

Figure 81 ConnectOPCUA in Control Panel

5.6 Installing Software in Linux

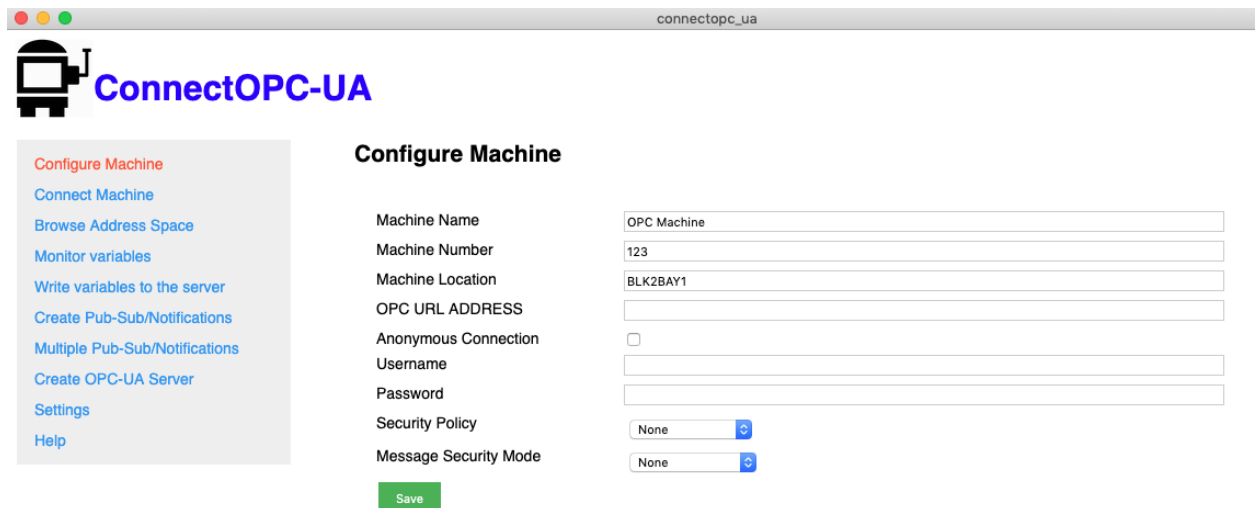
Following are the files present for the binaries that is required for the installation of Software in Linux (64 bit Environment).



Name	Date Modified	Size	Kind
chrome_100_percent.pak	Jul 16, 2019 at 3:35 PM	181 KB	Document
chrome_200_percent.pak	Jul 16, 2019 at 3:35 PM	294 KB	Document
chrome-sandbox	Jul 16, 2019 at 3:52 PM	5.2 MB	Unix executable
ConnectOPC-UA-app	Jul 16, 2019 at 3:52 PM	120.2 MB	Unix executable
icudtl.dat	Jul 16, 2019 at 3:18 PM	10.3 MB	DAT file
libEGL.so	Jul 16, 2019 at 3:52 PM	253 KB	Document
libffmpeg.so	Jul 16, 2019 at 3:52 PM	3.1 MB	Document
libGLESv2.so	Jul 16, 2019 at 3:52 PM	5.8 MB	Document
LICENSE	Jul 16, 2019 at 2:52 PM	1 KB	TextEdit
LICENSES.chromium.html	Jul 16, 2019 at 3:24 PM	2.1 MB	HTML
locales	Today at 9:24 PM	--	Folder
natives_blob.bin	Jul 16, 2019 at 3:43 PM	83 KB	MacBin...archive
resources	Today at 9:24 PM	--	Folder
resources.pak	Jul 16, 2019 at 3:36 PM	8.5 MB	Document
snapshot_blob.bin	Jul 16, 2019 at 3:51 PM	287 KB	MacBin...archive
swiftshader	Today at 9:24 PM	--	Folder
v8_context_snapshot.bin	Jul 16, 2019 at 3:51 PM	689 KB	MacBin...archive
version	Jul 16, 2019 at 2:52 PM	5 bytes	TextEdit

Figure 82 Installation files of Software in Linux (64 Bit Environment)

Software can be started by clicking on ConnectOPC-UA-app.



ConnectOPC-UA

- Configure Machine
- Connect Machine
- Browse Address Space
- Monitor variables
- Write variables to the server
- Create Pub-Sub/Notifications
- Multiple Pub-Sub/Notifications
- Create OPC-UA Server
- Settings
- Help

Configure Machine

Machine Name	OPC Machine
Machine Number	123
Machine Location	BLK2BAY1
OPC URL ADDRESS	
Anonymous Connection	☐
Username	
Password	
Security Policy	None
Message Security Mode	None

Save

Figure 83 Screenshot of Application running in Linux Platform (Ubuntu 14.04 LTS)

6. Case Study with Siemens Sinumerik 828D

In this section, we will test the software with Siemens make 828D CNC Controller, the test has been carried out in 2-Axes CNC Lathe machine. Before using the software in this CNC Controller we will look into the CNC Controller it features at a glance. There after we will setup the CNC Controller to make it ready for OPCUA Communication and configure the machine with the OPCURL. The study of the CNC Machines, CNC Controllers, PLC are not the scope of this documentation, but where it is necessary to understand things in context of software, we will bring out some basic concepts at a glance.

6.1 Siemens 828D CNC Controller at a glance

With their unique CNC performance, the SINUMERIK 828D and SINUMERIK 828D BASIC CNCs set benchmarks when it comes to milling and turning on mid-range machines. Siemens premium CNC technology opens up unique possibilities for even more productive workpiece milling and turning. With its SINUMERIK 828 controls, Siemens Machine Tool Systems offers compact CNC systems for standard milling and turning machines.

With its technology-specific system software, the range of applications extends from vertical and basic horizontal machining centers — naturally also for mold making applications — up to turning machines with counter spindle, driven tools and Y-axis. Rugged hardware architecture and intelligent control algorithms, along with premium drive and motor technology, ensure the highest dynamic response and precision when machining. The intuitive SINUMERIK Operate graphical user interface facilitates efficient machine operation. With their outstanding performance, SINUMERIK 828 controls master all of the demands for standard milling and turning machines, and are supplemented by a range of integrated IT solutions.



Figure 84 Siemens 828D CNC Controller, Drives & Motors

Rugged and maintenance-free

With an operator panel front manufactured out of die-cast magnesium, a panel-based CNC design with well-conceived interfaces along with the high IP65 degree of protection make the SINUMERIK 828 a reliable CNC, even in harsh environments. The SINUMERIK 828D and SINUMERIK 828D BASIC are completely maintenance-free CNC controls as they have no fan and no hard disk — and are equipped with NV-RAM memory technology without buffer battery. Two different panel layouts are available — horizontal and vertical — a feature that further enhances their versatility.

User-friendly

Equipped with a full QWERTY CNC keyboard with shortcut keys and a high resolution 8.4" / 10.4" TFT color display, SINUMERIK 828 controls can be simply operated. CNC data can be transferred quickly and easily using USB, Compact Flash (CF) card and RJ45 interfaces at the operator panel front. The fully graphical user interface and the structure of the CNC direct keys facilitate fast operation with just a few keystrokes. Turning and milling machines are operated in an identical fashion.

A CNC portfolio for the global machine tool market

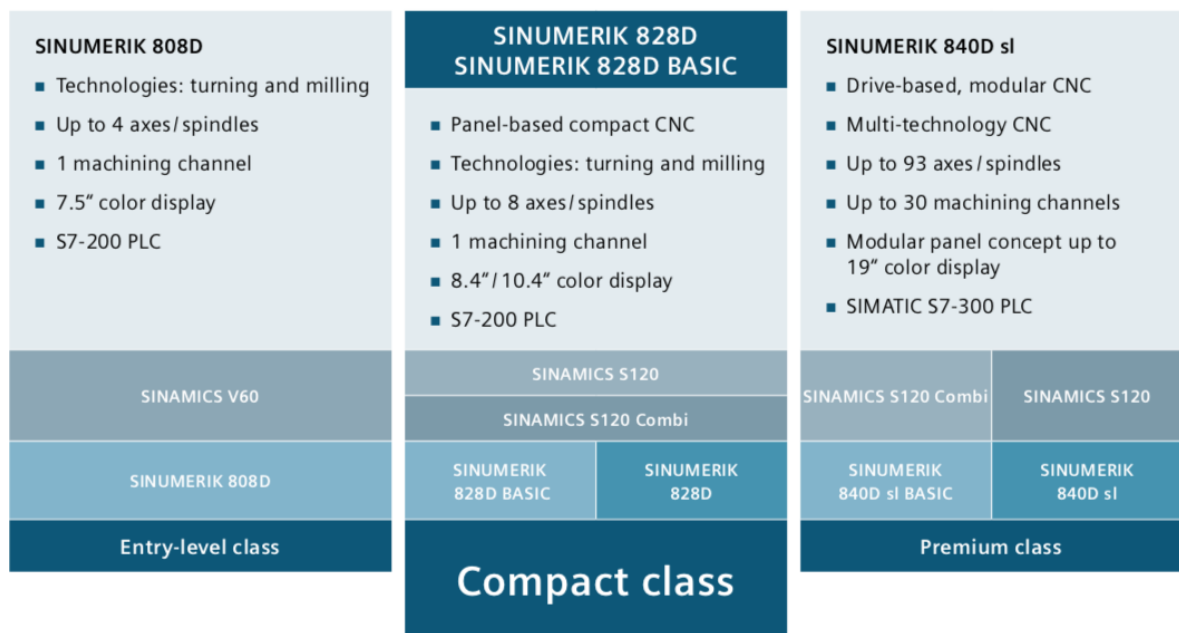


Figure 85 Family Line of Siemens 828D CNC Controllers

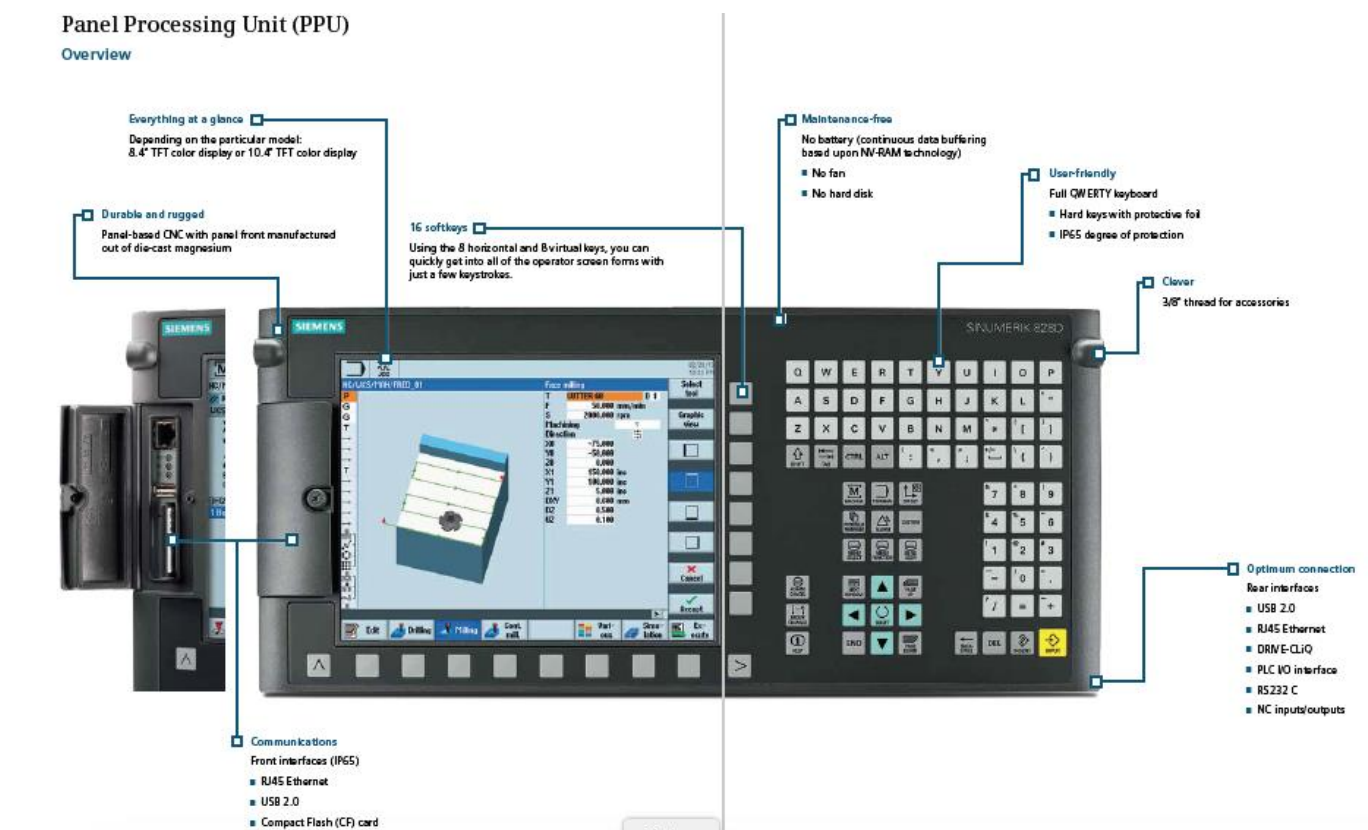


Figure 86 PPU of Siemens 828D

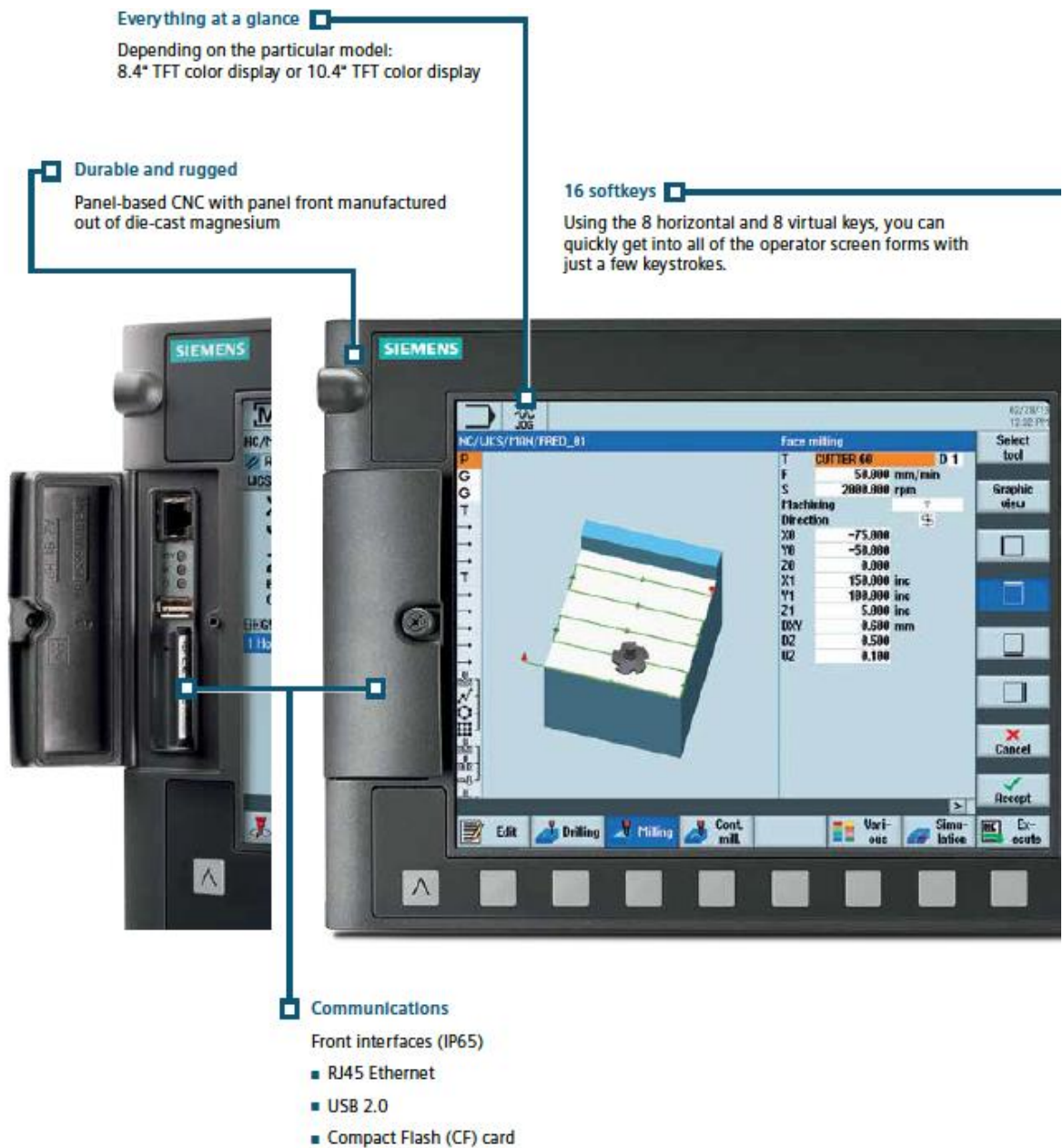


Figure 87 PPU Features 1

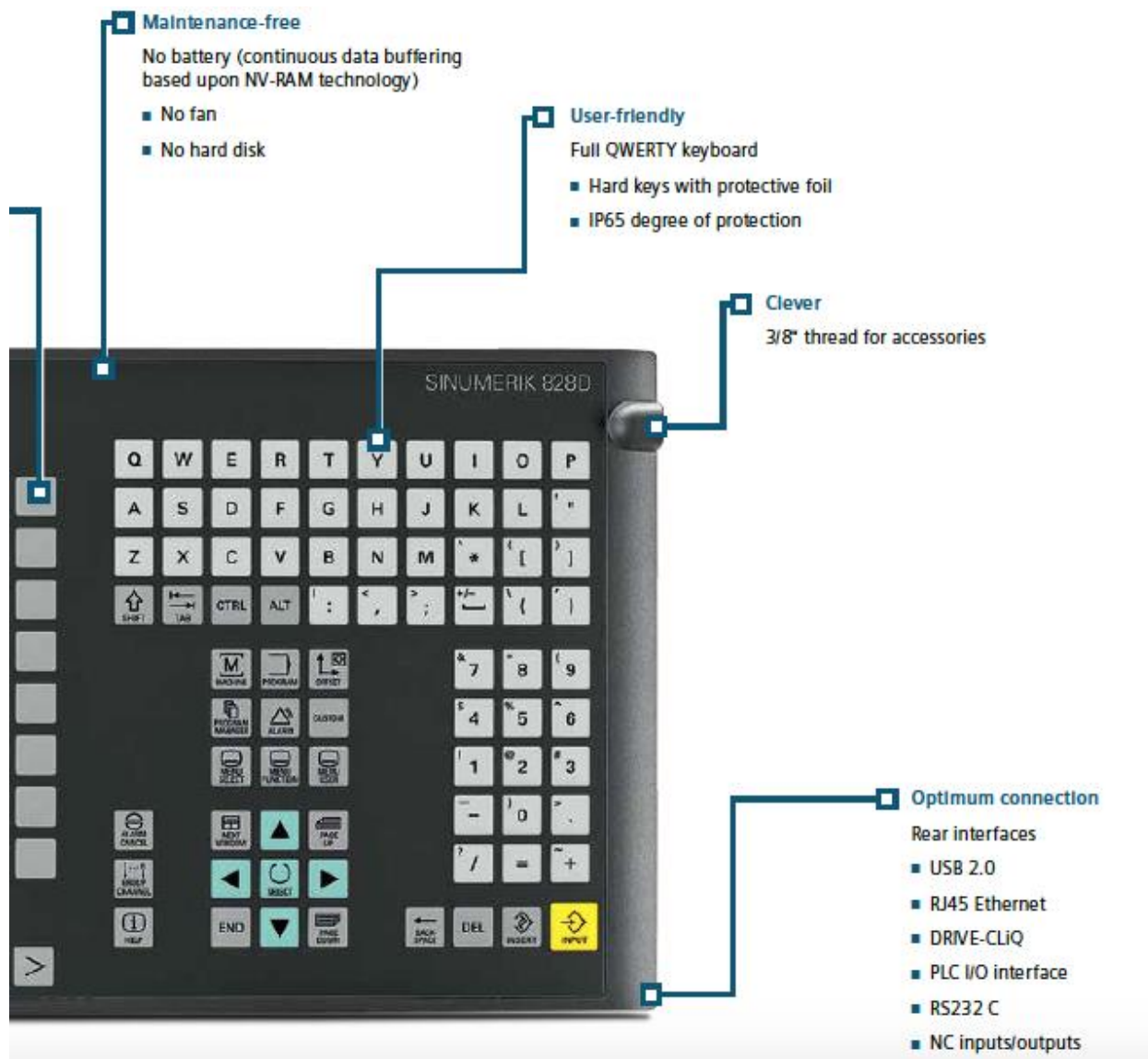


Figure 88 PPU Feature2

SINAMICS S120 is synonymous with performance and flexibility when it comes to CNC machines. In addition to a wide range of motor modules up to a power rating of 300 kW, there is also an in feed unit with a controlled DC link. This ensures the shortest spindle acceleration times and facilitates perfect reactive power compensation for the complete machine ($\cos \varphi = 1$). This is complemented by DSC (Dynamic Servo Control), which represents a unique position control technique to achieve the highest dynamic performance of feed and spindle motors.



Figure 89 SINAMICS DRIVES AND SIMOTICS SERVOMOTORS

High standstill torques, the fastest speeds and smooth-running characteristics make SIMOTICS servomotors the optimal feed drive for CNC machines. A high degree of protection, strong bearings and a vibration-free design mean that these synchronous servomotors have outstanding reliability. High-quality magnetic materials result in a very high-power density — and therefore very small motor dimensions. As a result, these motors can be installed in extremely tight spaces.

The highest precision

SINUMERIK CNCs and SINAMICS drives compute with high-performance 80 bit NANOP accuracy. This eliminates rounding errors and results in an extremely high internal computational accuracy in the complete controller circuit. Feed forward control ensures that errors are almost completely compensated — and jerk limitation reduces stress on the mechanical system when axes accelerate. Thanks to Dynamic Servo Control, SINAMICS drives offer additional position control — and as an additional feature, disturbance resistance for the machine control.

Maximum speed

When machining many CNC blocks in the shortest time, for example, free-form surfaces, the machining process itself no longer defines the speed, but the performance of the CNC system. With the Advanced Surface feature, SINUMERIK CNC offers you the ideal solution. Advanced Surface stands for state-of-the-art control algorithms, such as Look Ahead or the dynamic compression of linear and circular blocks in fifth degree polynomials (NURBS). This means that machines equipped with SINUMERIK controls can be operated at their physical limit.

The shortest idle times

Especially in large serial production, idle times, where the machine is no longer productive, represent a critical productivity-inhibiting factor. Here, with its synchronous architecture and intelligent functions, such as synchronized actions or asynchronous sub-programs, SINUMERIK provides the optimal solution. For example, the loading of equipment can be implemented without having to make time-consuming interventions in the PLC.

The machining standard

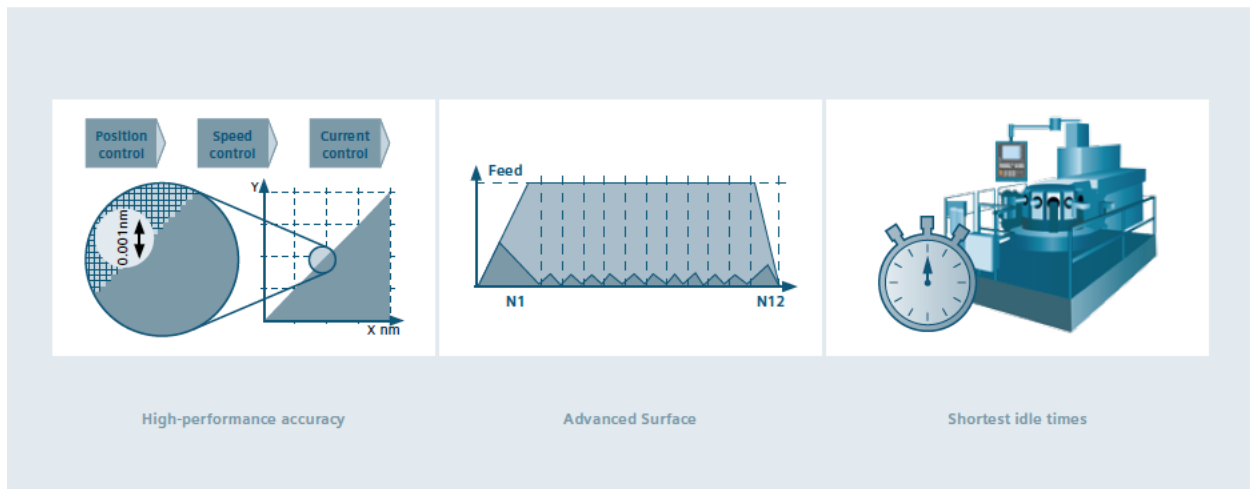


Figure 90 Siemens 828D CNC Machining standards

6.2 Setting up the OPCUA in Siemens 828D

- 1) Set the "Access MyMachine / OPC UA" option via the "Startup > Licenses" operating area.

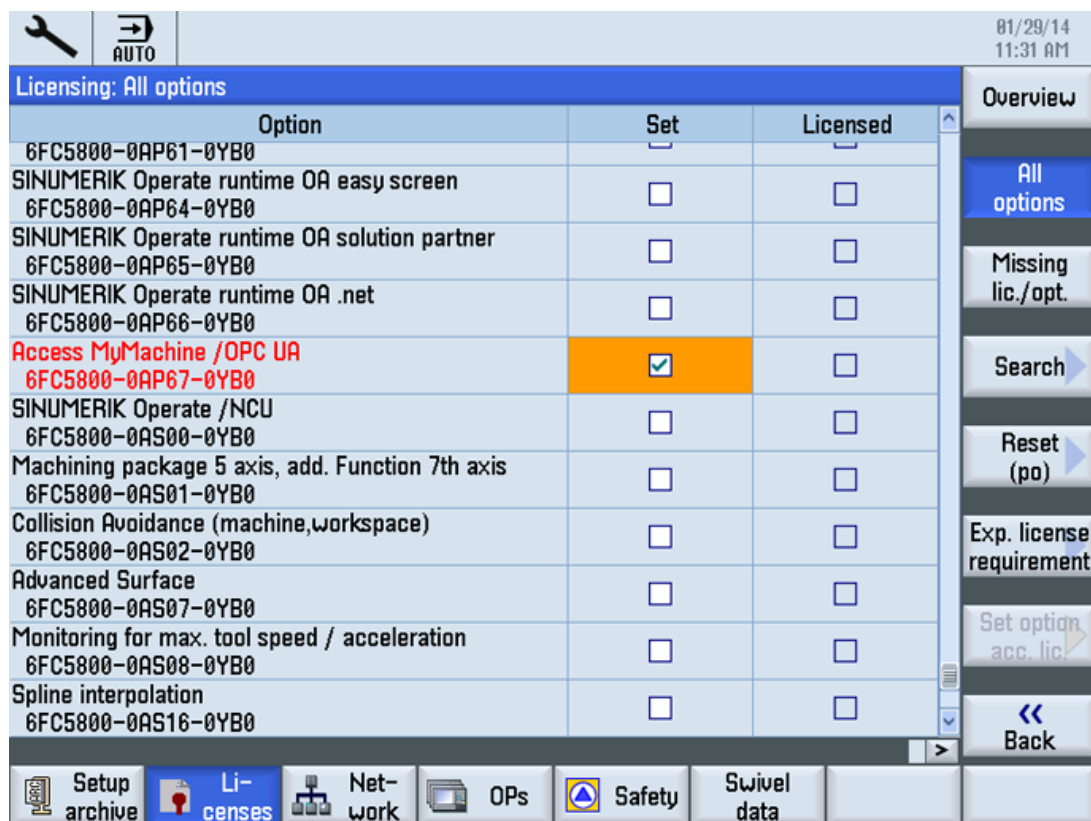


Figure 91 Setting the OPCUA Option in Siemens 828D

- 2) Restart the CNC Controller, step 1 will enable the OPCUA in the next startup
- 3) Start the OPCUA Configuration dialog via "Startup→ Network"

- 4) Press the OPCUA Softkey
- 5) Press the "Setup" softkey. The Settings dialog will appear. Then press the "Change" softkey. Make the necessary settings for connection, authentication and activation.

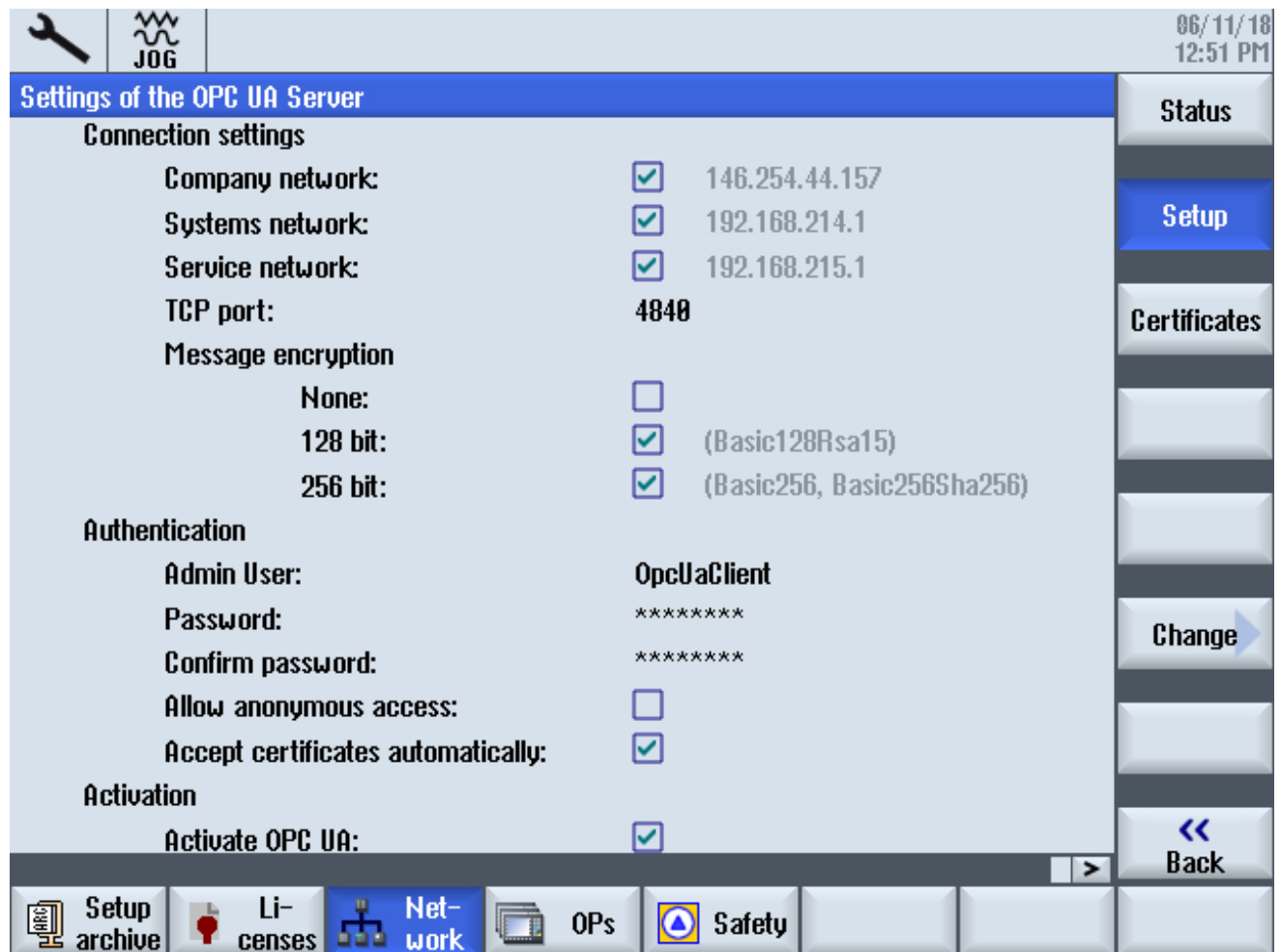


Figure 92 Setup Options in OPCUA Dialog

Group	Setting	Description
Connection Settings	Company Network	The available network connections (IP address) on a specific target system (828D, 840D sl, PCU, IPC) are shown.
	Machine Network	
	Service Network	
	TCP PORT	TCP port at which the OPC UA server should be available. Standard configuration: 4840 Note! The port must also be open in the firewall. For PPU/NCU this happens automatically. With PCU/IPC the port must be opened manually in the firewall.
	Message Encryption	It can be chosen which security endpoints should be offered from the server
	Setting	Standard Configuration
	None	Deactivated

		128 Bit	Activated
		256 Bit	Activated
Authentication	Admin Username	User name of the administrator. The administrator can add or delete users and assign or delete user authorizations.	
	Password	Password of the user	
	Confirm Password	Confirm the password of the user	
Activation	Activate OPCUA	Place the checkmark to activate OPC UA and remove the checkmark to deactivate it.	

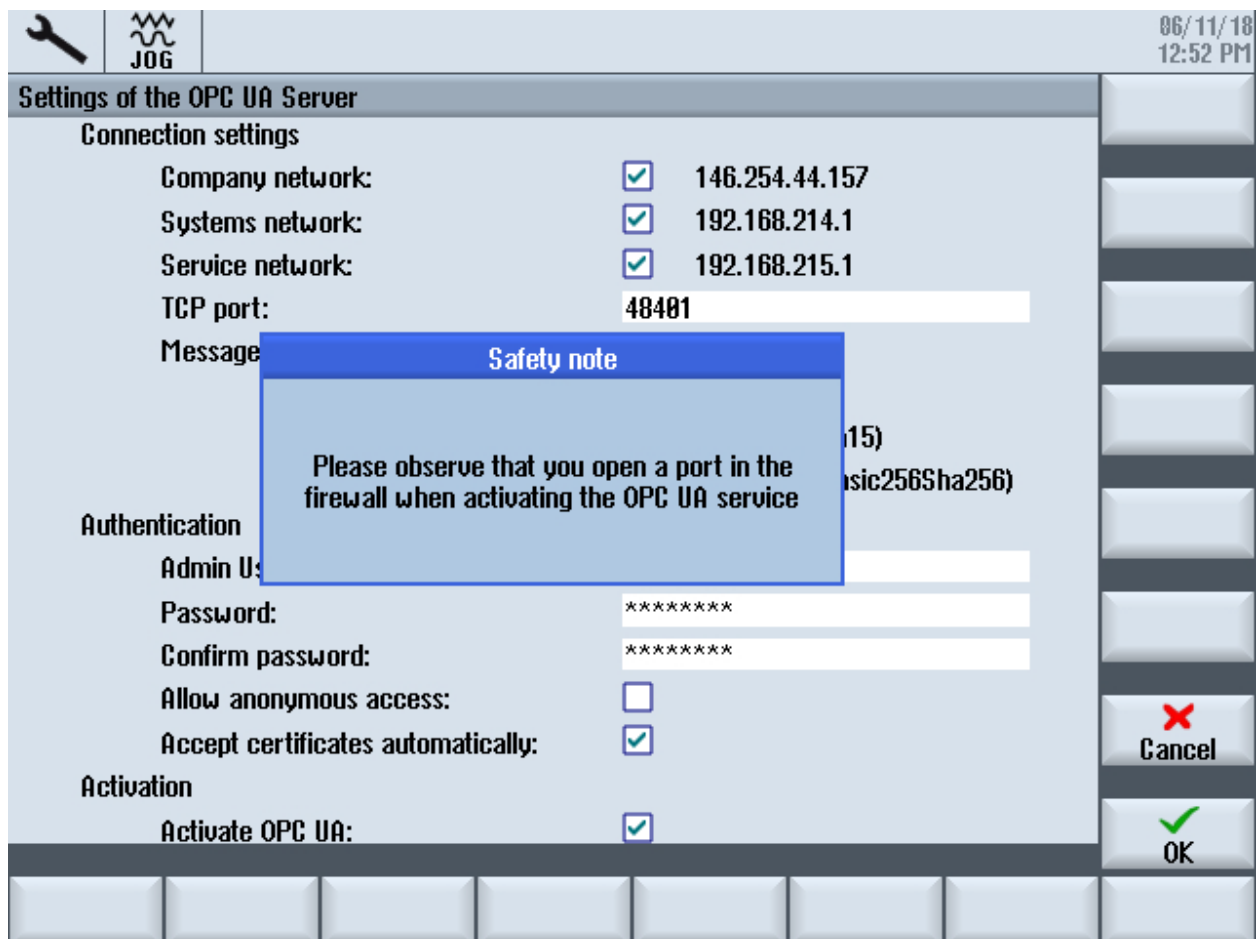


Figure 93 Security message for opening a TCP Port

- 6) If settings are all done, restart is necessary to activate the new settings. Perform a hardware restart on the target systems NCU and PPU. A restart of the SINUMERIK Operate is necessary on the PCU 50/IPC.

6.3 Overview

The OPC UA server provides the possibility to communicate with SINUMERIK via OPC UA.

The following functionalities of the OPC UA specification are supported by the server:

- 1) Data Access: Read, write and subscribe to SINUMERIK variables (NC, PLC)
- 2) Alarms & Conditions: Event based provision of SINUMERIK alarms and messages from HMI, NC and PLC

3) Methods: User management, file transfer, tool management and program selection

This section describes the address space of the OPC UA server and gives further information how to address some SINUMERIK specific values. Especially since a lot of SINUMERIK values are stored in arrays or matrices. Furthermore you can find description on the SINUMERIK alarm object and how to get the alarms from the server.

If the OPC UA server is browsed, the available address space is mapped under the "Sinumerik" node. Global User Data (GUD) can be found under the "/Sinumerik/GUD" node. The PLC blocks (inputs, outputs, bit memory, data blocks) can be found under the "/Sinumerik/Plc" node. Machine data can be found under the node "/Sinumerik/TEA". Setting data can be found under the node "/Sinumerik/SEA".

Observe the following while browsing:

1) In the address space of the NC, the displayed variables always represent only the first parameter of the corresponding unit.

Example:

The R parameters can be found under "Sinumerik > Channel > Parameter > R". The corresponding identifier is called "/Channel/Parameter/R", which is finally mapped to "/Channel/Parameter/R[u1, 1]". If you want to access other parameters, you need to specify the corresponding index in brackets, e. g. "/Channel/Parameter/R[u2,56]".

2) In the address space of the PLC, the displayed variables represent the access format that has to be extended accordingly.

Example:

The variable "/Plc/MB" is in the address space and is mapped to "/Plc/MB0". For accessing further bytes this variable must be extended by the appropriate byte number, e.g. "/Plc/MB6".

3) The address space of the NC also contains variables that are not available in a corresponding machine configuration. These variables return "BadAttributeIdInvalid" as value.

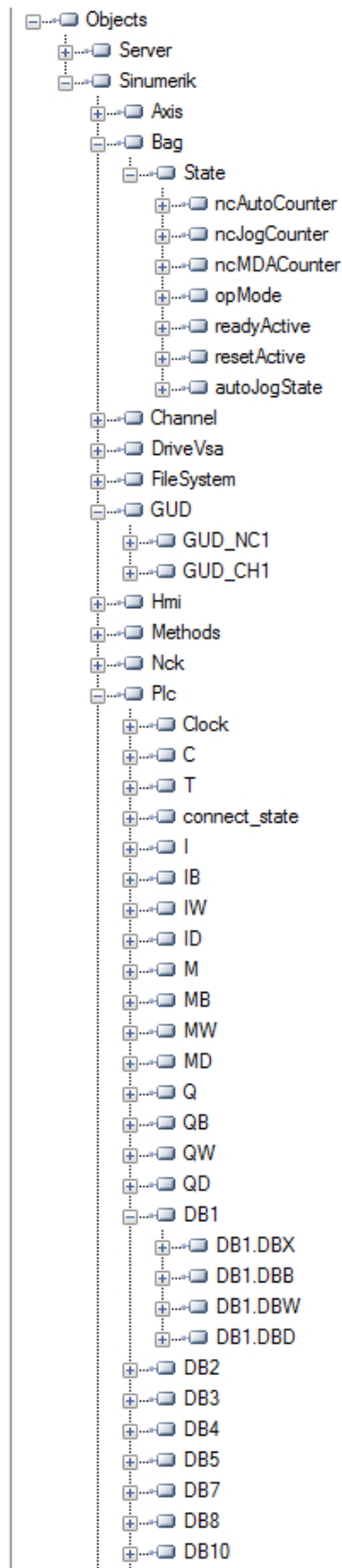


Figure 94 Browsing Address Space in Siemens 828D

The variable paths for NC access are stored in the address space of the SINUMERIK Operate OPC UA server. The R parameters are found under the identifier `"/Channel/Parameter/R"`, which is eventually mapped to `"/Channel/Parameter/R[u1, 1]"`. If you want to access other

parameters, you must correspondingly extend the identifier, for example "/Channel/Parameter/R[u2, 56]".

Variable Path	Description
/Channel/Parameter/R[u1,10]	R parameter 10 in channel 1
/Channel/Parameter/R[u1,1,5]	R parameter array
/Channel/Parameter/R[u1,1,#5]	R parameters 1 to 5 in channel 1
/Channel/GeometricAxis/name[u2,3]	Name of the 3rd axis in channel 2
/Channel/GeometricAxis/actToolBasePos[u1,3]	Position of the 3rd axis in channel 1
/Plc/M5.0	Memory bit 0 at byte offset 5
/Plc/DB5.DBW2	Word (16-bit) at byte offset 2 in data block 5
/Plc/DB8.DBB2	STRING UTF8 string beginning at byte offset 2 in data block 8
/Plc/DB8.DBW2[10]	Array of 10 words beginning at byte offset 2 in data block 8
/Plc/DB100.DBB1	Byte at byte offset 1 in data block 100
/Plc/DB2.DBDO:REAL[10]	Array of 10 double words (32-bit) beginning at byte offset 0 in data block 2, which are formatted as a floating-point number
/NC/_N_CH_TEA_ACX/\$MC_CHAN_NAME	Channel name of channel 1
/NC/_N_CH_TEA_ACX/\$MC_CHAN_NAME[u2]	Channel name of channel 2

6.4 Features

The SINUMERIK OPC UA server provides the possibility to communicate with SINUMERIK via OPC UA. The following functionalities of the OPC UA specification are supported by the server.

- 1) Read, write and subscribe to SINUMERIK variables (NC, PLC)
- 2) Transfer of part programs (see chapter File system)
- 3) Support for File and Folder Objects
- 4) Event based provision of SINUMERIK alarms and messages from HMI, NC and PLC
- 5) Methods for selection of part programs from the NC file system and tool management
- 6) Multi language support for the alarm and warning messages.

➔ Security settings

The server provides the possibility to communicate in an unencrypted or encrypted way. The following options are possible:

- 1) None
- 2) 128 Bit - Sign (Basic128Rsa15)
- 3) 128 Bit - Sign & Encrypt (Basic128Rsa15)
- 4) 256 Bit - Sign (Basic256Sha256)
- 5) 256 Bit - Sign (Basic256)
- 6) 256 Bit - Sign & Encrypt (Basic256Sha256)
- 7) 256 Bit - Sign & Encrypt (Basic256)

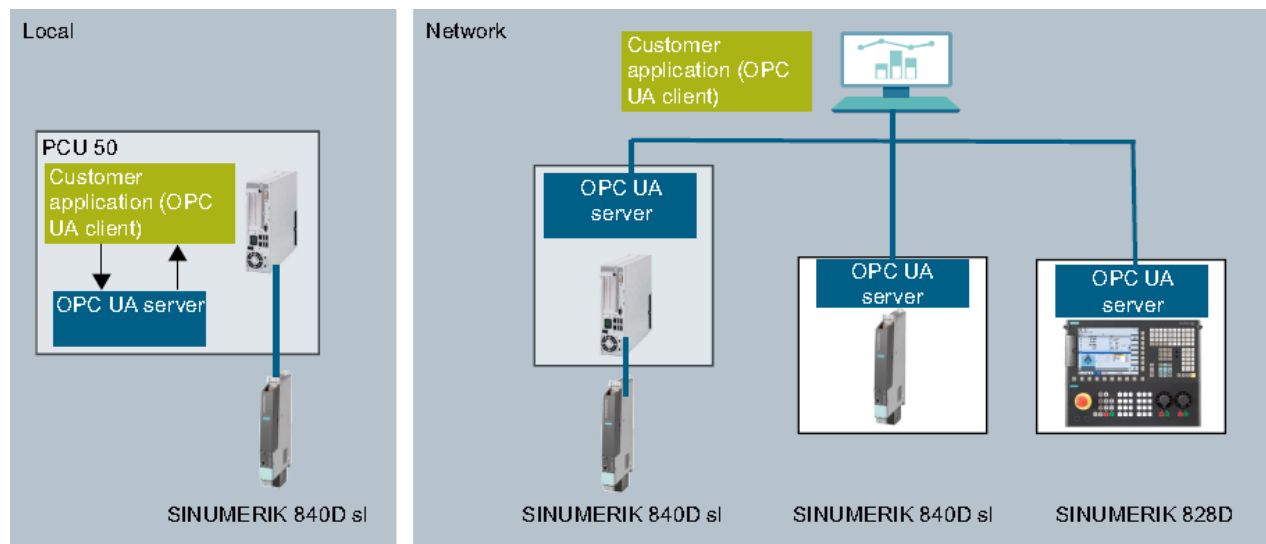


Figure 95 Application scenario

SINUMERIK systems	Accessiblity	
SINUMERIK 828D	After successful licensing and activation the OPC UA server is accessible via the X130 interface.	
SINUMERIK 840D sl	The OPC UA server needs SINUMERIK Operate and runs on the same place as SINUMERIK Operate. For this reason, system setup depends on whether a Thin Client is used (SINUMERIK Operate runs on NCU) or a PCU / IPC with Windows operating system. If a Windows operating system is used, the OPC UA server is also accessible as LocalHost.	
	Thin Client	If a Thin Client is used, the OPC UA server is accessible after successful licensing and activation via X130 interface of the NCU.
	PCU / IPC	If a PCU / IPC is used, the OPC UA server is accessible after successful licensing and activation via "eth1" (X1) interface of the PCU / IPC. In this case the OPC UA server is not accessible via the X130 interface of the NCU.

Figure 96 Accessibility of the server

The admin can add/delete users and rights via OPC UA methods provided by the server. Therefore a connection with a generic client must be established, using the admin credentials. Users and rights can then be assigned using the following OPC UA server methods:

- 1) Add users (**AddUser, AddCertificateUser**)
- 2) Delete users (**DeleteUser**)
- 3) List users (**GetUserList**)
- 4) Change password (**ChangeMyPassword**)
- 5) Give access rights (**GiveUserAccess**)
- 6) Remove access rights (**DeleteUserAccess**)
- 7) List access rights (**GetMyAccessRights, GetUserAccessRights**)

A new user created with the "AddUser" or "AddCertificateUser" function has no rights at all. The user administrator has the responsibility for the user management and the associated rights. All users must use a secure password.

Methods	Description
AddUser	Create a new user for accessing OPC UA
	Input Arguments :
	Username Username
	Initially password of new user is username, password must be changed using change my password
Add Certificate User	Add a new user via certificate authentication
	Input Arguments :
	Username User to Whom certificate is issued
	Certificate Certificate (.der) as bytestring
Delete User	Delete the User from accessing OPCUA
	Input Arguments :
	Username Username to be deleted
	User setup while configuring OPCUA cannot be deleted
Change My Password	Changes the password of a connected User
	Input Arguments :
	Oldpw1 Old Password of user
	NewPw New password of the user
	NewPw1 Confirm New Password of the user
GiveUserAccess	Give User Access Rights
	Input Arguments :
	User Username to whom right to be granted
	Rights Rights represented in realm string
Some Possible Realm Strings for Access Rights	
"StateRead" → Status data - NC, channel, axis, read access "StateWrite" → Status data - NC, channel, axis, write access "FrameRead" → Zero offsets, read access "FrameWrite" → Zero offsets, write access "SeaRead" → Setting data, read access "SeaWrite" → Setting data, write access "TeaRead" → Machine data, read access "TeaWrite" → Machine data, write access "ToolRead" → Tool and magazine data, read access "ToolWrite" → Tool and magazine data, write access, Tool management methods "DriveRead" → Drive data, read access "DriveWrite" → Drive data, write access "GudRead" → User data, read access "GudWrite" → User data, write access "FsRead" → File system, read access "FsWrite" → File system, write access "PlcRead" → PLC, read access "PlcWrite" → PLC, write access "AlarmRead" → Allows to subscribe to alarms "RandomRead" → Random (and ReadVar method), read access "RandomWrite" → Random (and WriteVar method), write access "SinuReadAll" → All of the read access operations mentioned "SinuWriteAll" → All of the write access operations mentioned "ApWrite" → Allowed to call method Select	

6.5 Simulation of OPC-UA Server With Sinutrain

The **SinuTrain** NC programming station brings SINUMERIK Operate including animated machine operator panel to the PC. It therefore facilitates smooth know-how transfer from training into practical environments. This powerful tool allows offline programming at the PC and NC programs can be transferred directly to the CNC. Using SINUMERIK Operate and the original SINUMERIK CNC kernel, all operating and programming operations as well as the execution of CNC programs can be realized without any restrictions.



Figure 97 Sinutrain Software Package

The intuitive PC-based SinuTrain solution comprises software that is identical with that of the control system. This perfectly addresses learning and programming as required in practice, and supports all of the CNC programming methods established worldwide. SinuTrain for

SINUMERIK Operate can be individually adapted to the axis configurations of various machines. This in turn permits maximum compatibility to be achieved between the CNC programs generated offline and the machines on the factory floor.

We have used Sinutrain V4.7, the software is installed on Windows 8 Pro platform, we have to create the machine and then OPC-UA will be activated on that machine. Fig-98 shows the main screen of Sinutrain Software. Fig-99 shows the creation of new CNC Machine using template. In our Example we created 2-Axis (X & Z Axes) Lathe Machines with 1 Spindle. Fig-100 shows the machine screen just configured in the Sinutrain. Fig-101 shows the execution of part Program in Sinutrain

6.5.1 Activating OPC-UA in Sinutrain

Activating OPC-UA require different steps in Sinutrain. Options must be first activated by clicking Setup → License → All Options → Search for OPCUA → Set that OPCUA Option → Restart the Machine. The above steps are shown in Fig ____.

Copy ALL files from folder

```
C:\Siemens\SinuTrain\SINUMERIK CNC-SW 840D sl 4.7 SP3 HF1\hmi\siemens\sinnerik\hmi\
template\cfg\miniweb\
```

to folder

```
C:\Siemens\SinuTrain\SINUMERIK CNC-SW 840D sl 4.7 SP3
HF1\hmi\user\sinnerik\hmi\miniweb\cfg\
```

Add file UserDataBase.xml (from the .zip archive) to directory

```
C:\Siemens\SinuTrain\SINUMERIK
CNC-SW 840D sl 4.7 SP3 HF1\hmi\siemens\sinnerik\hmi\miniweb\System\UserDataBase\
```

File UserDataBase.xml contains entries for OPC UA user 'OpcUaClient' with Password 'OpcUaClient'. This user is administrator of the OPC UA server and has the right to read and write all items at the controller.

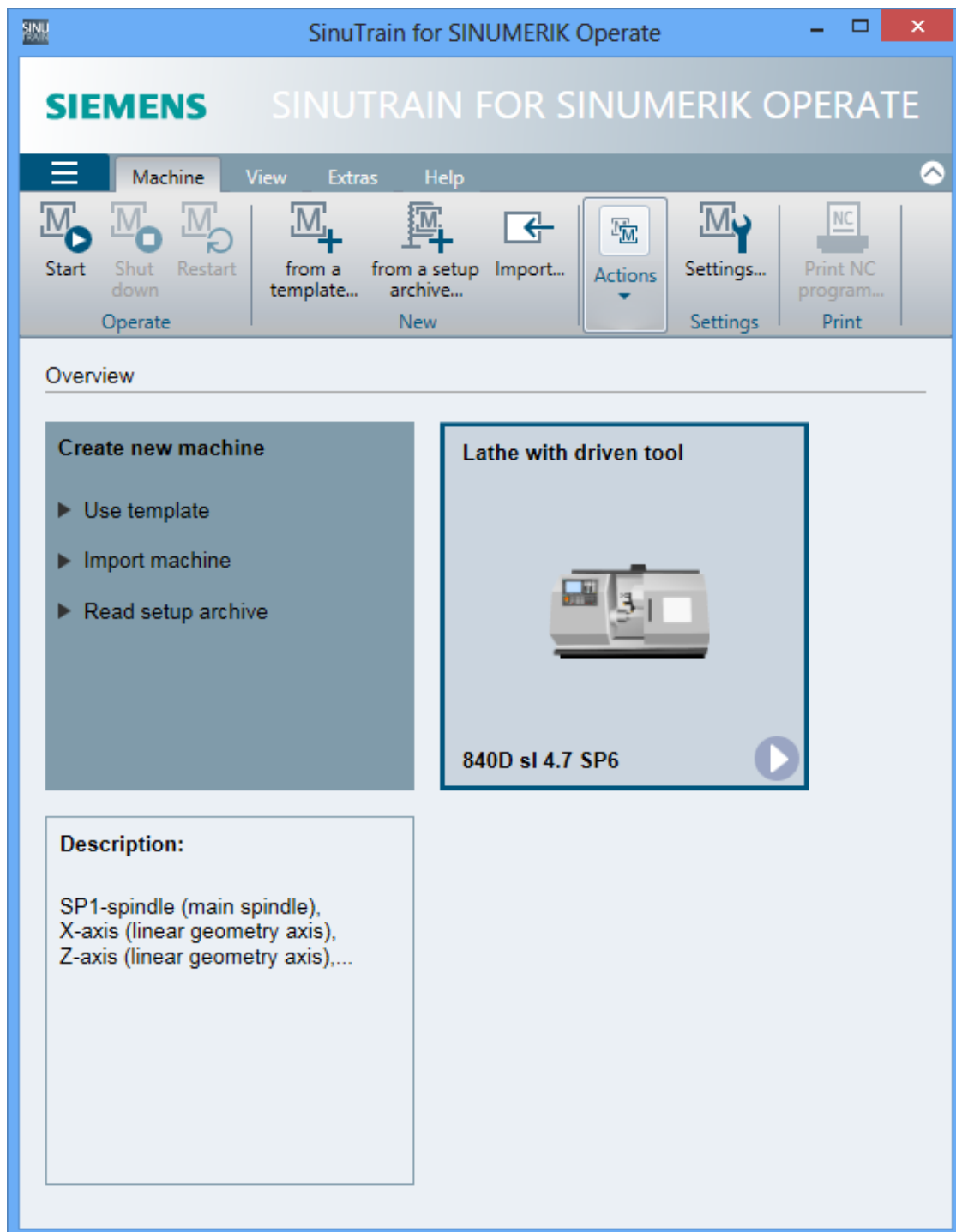


Figure 98 Sinutrain Main Screen

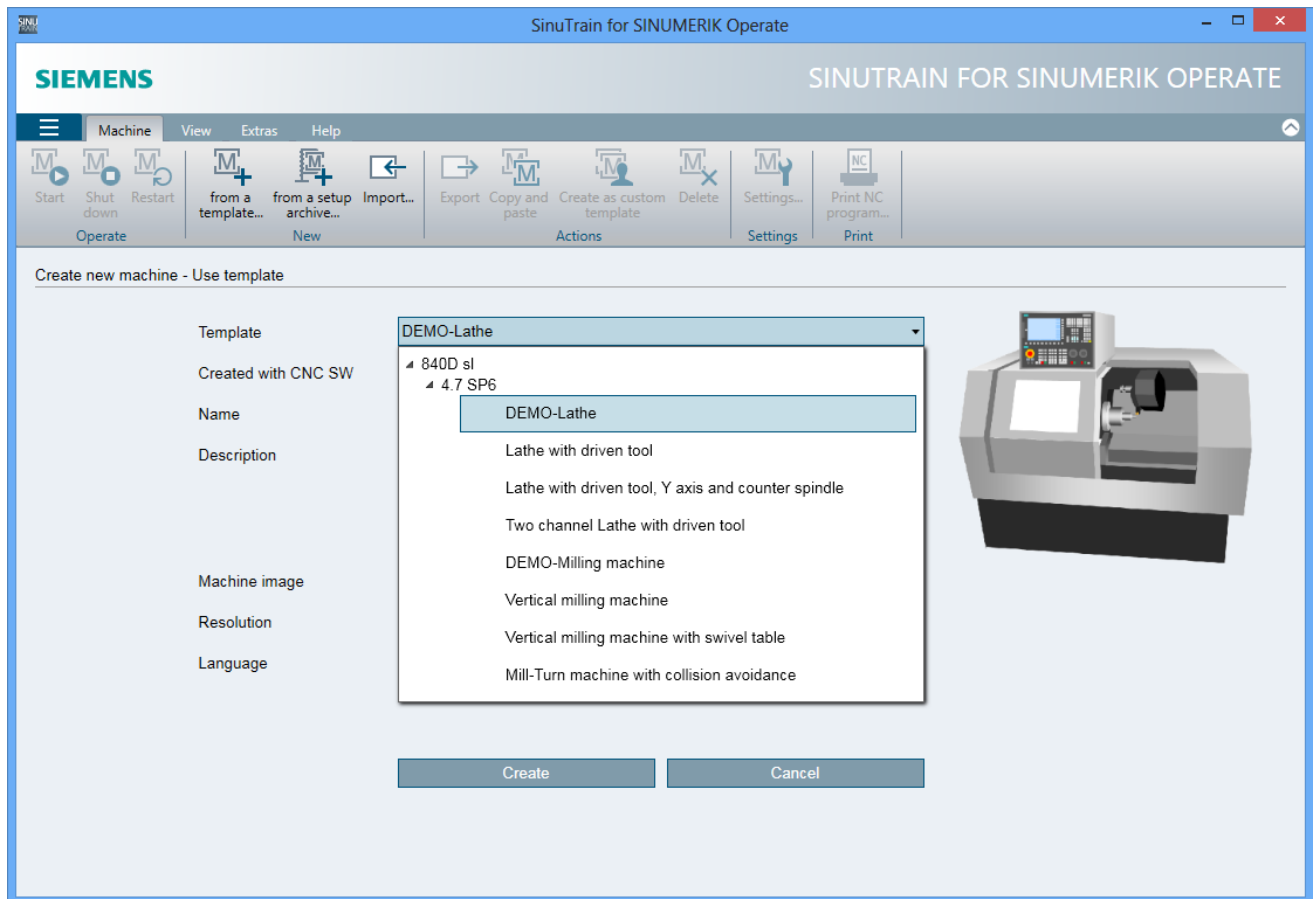


Figure 99 Creation of new Lathe Machine in Sinutrain

Start SinuTrain, minimize it.

Open file C:\Siemens\SinuTrain\SINUMERIK CNC-SW 840D sl 4.7 SP3

HF1\hmi\user\sinnerik\hmi\cfg\systemconfiguration.ini with a text editor.

At the end add:

[processes]

PROC100= image:="C:\Siemens\SinuTrain\SINUMERIK CNC-SW 840D sl 4.7 SP3

HF1\hmi\siemens\sinnerik\hmi\miniweb\release\miniweb.exe", process:=MiniWebServer,

cmdline:="..\System ..\WWWRoot", startupTime:=afterServices,

workingdir:="C:\Siemens\SinuTrain\SINUMERIK CNC-SW 840D sl 4.7 SP3

HF1\hmi\siemens\sinnerik\hmi\miniweb\release"

Restart the machine from the SinuTrain Workbench and make sure that systemconfiguration.ini still contains the additional process.

If configured correctly on startup of the machine now an additional command line window opens that shows details about the activities of miniweb.exe. Do NOT close this window!

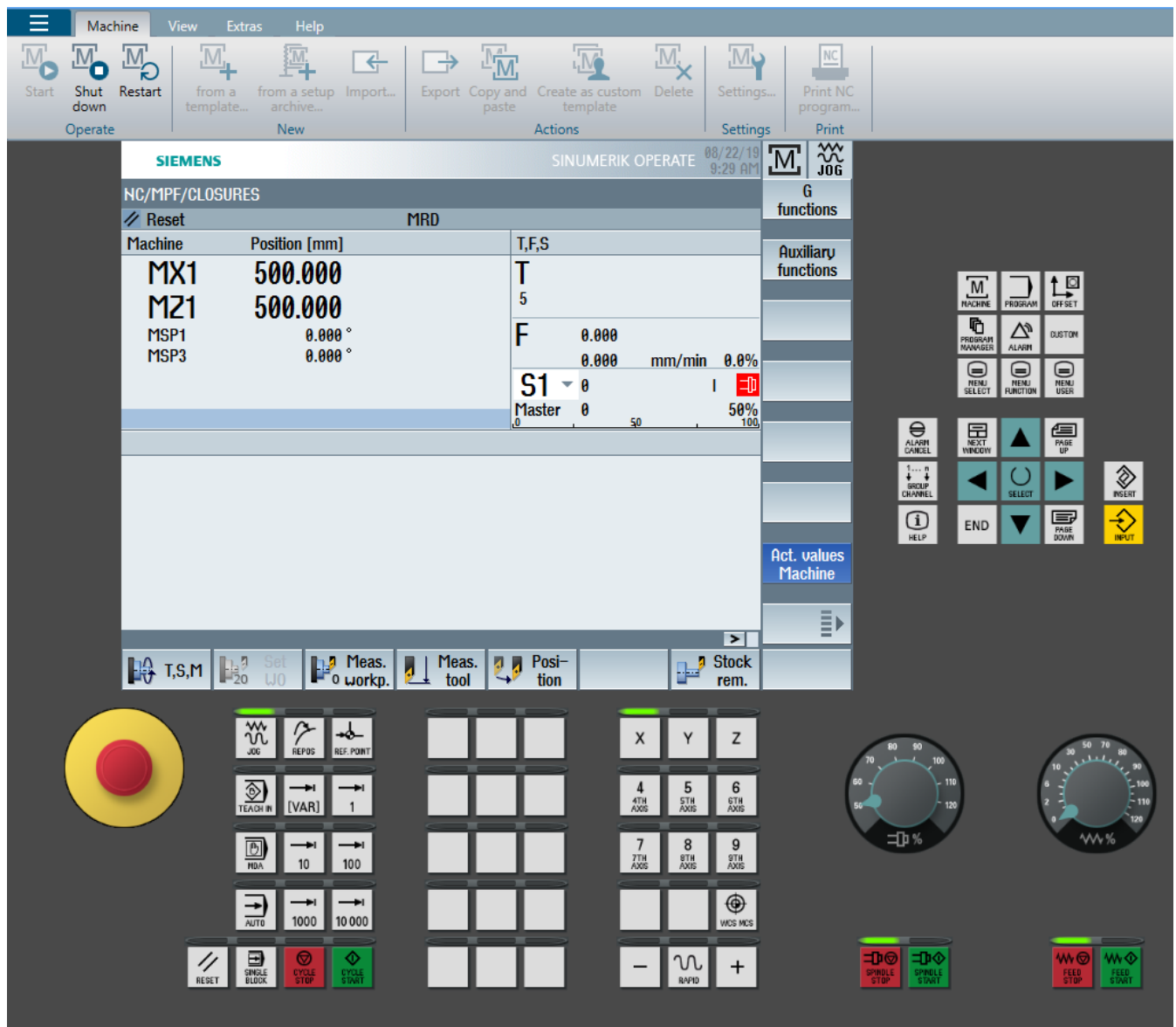


Figure 100 Machine Screen in Sinutrain

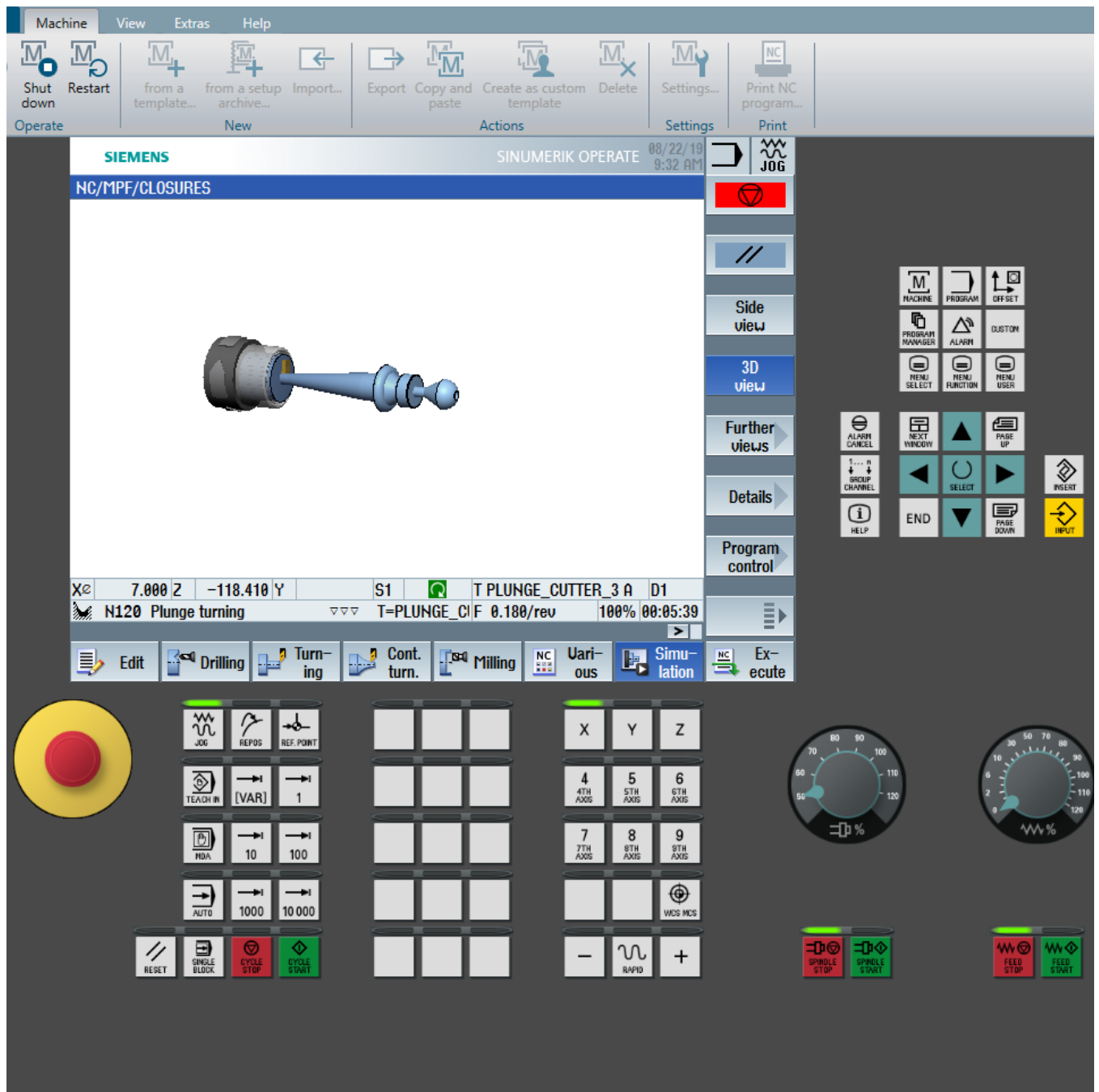


Figure 101 Running of Part Program in Sinutrain

SIEMENS SINUMERIK OPERATE 08/22/19 9:39 AM   JOG

Licensing: All options

Option	Set	Licensed
Additional 1 axis/spindle 6FC5800-0AA00-0YB0	0	255
Additional 1 positioning axis/auxiliary spindle 6FC5800-0AB00-0YB0	0	255
Additional 1 operation mode modul (BAG) 6FC5800-0AC00-0YB0	0	255
Additional 1 channel 6FC5800-0AC10-0YB0	0	255
SI axis/spindle, add. 1 axis/spindle 6FC5800-0AC70-0YB0	0	255
Add. 2 Mbyte CNC user memory 6FC5800-0AD00-0YB0	0	255
Add. 128 KB PLC user memory 6FC5800-0AD10-0YB0	0	255
Safety Integrated plus /SI axis/spindle 6FC5800-0AK00-0YB0	0	255
Travel to a fixed stop with force control 6FC5800-0AM01-0YB0	☐	☒
Synchronous axes pair (GANTRY) 6FC5800-0AM02-0YB0	☐	☒
Master/slave for drives 6FC5800-0AM03-0YB0	☐	☒

Overview

All options

Missing lic./opt.

Search

Reset (po)

Exp. license requirement

Set option acc. lic.

Back

Li-censes

Swivel data

Figure 102 Licenses in Sinutrain

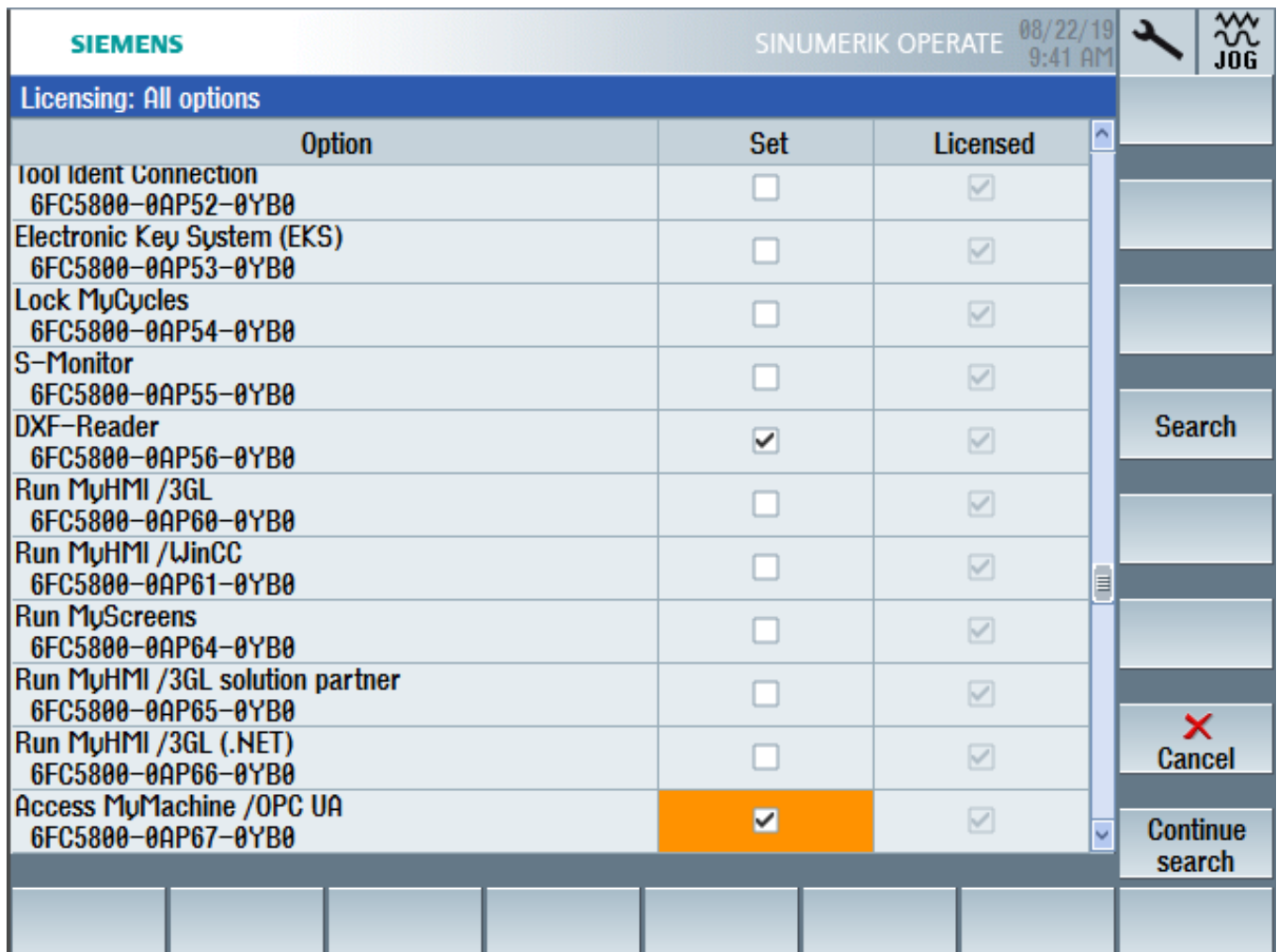


Figure 103 Setting the Option OPCUA in Sinutrain

```

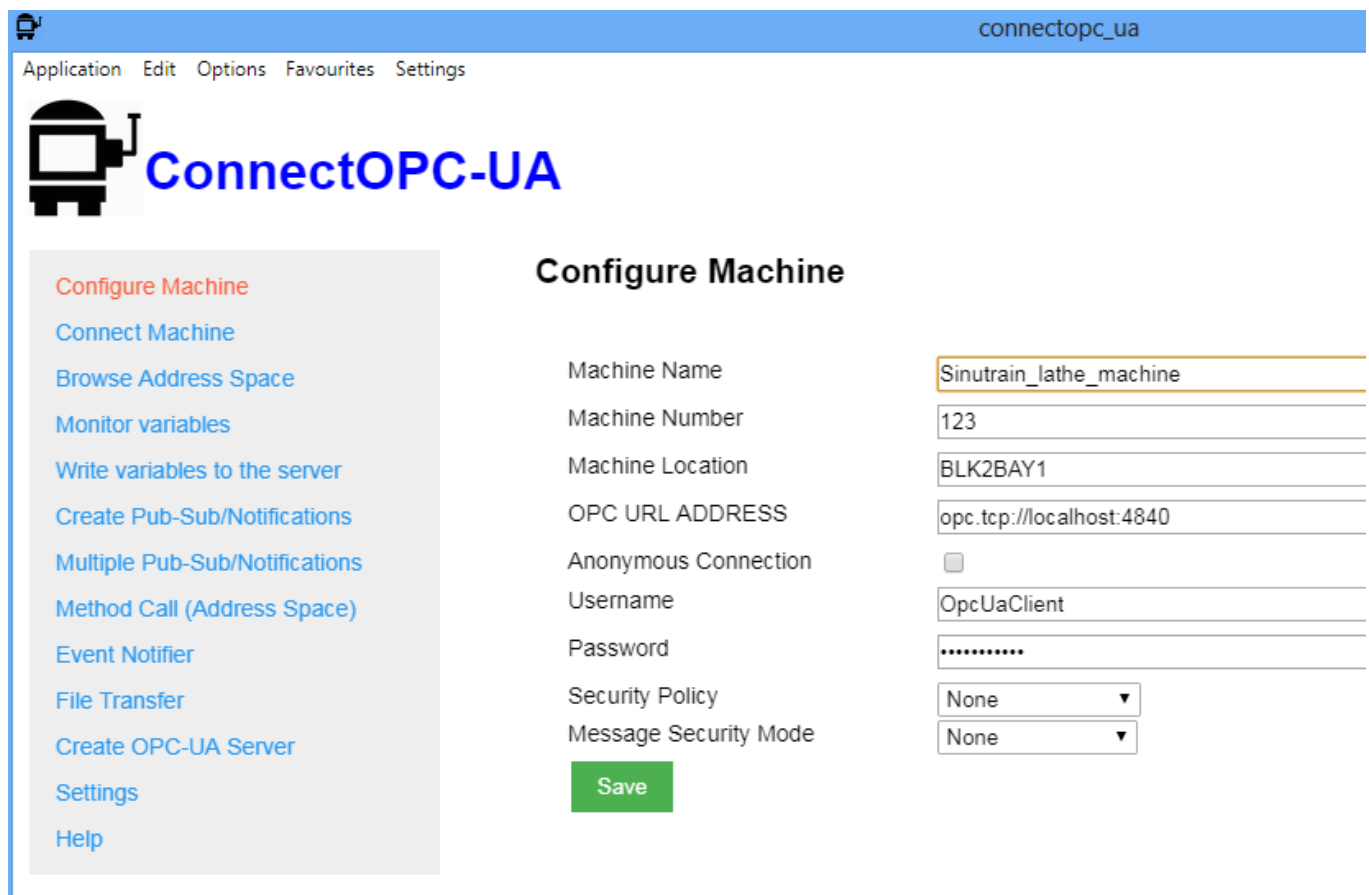
C:\Siemens\SinuTrain\SINUMERIK CNC-SW 840D sl 4.7 SP6\hmi\siemens\sinum...
Initialisation of static modules completed!
Initialisation of static modules completed!
Parameter-file for AddressModeller not yet available, wait another 5 sec
Parameter-file for AddressModeller not yet available, wait another 5 sec
Parameter-file for AddressModeller not yet available, wait another 5 sec
Parameter-file for AddressModeller not yet available, wait another 5 sec
Parameter-file for AddressModeller not yet available, wait another 5 sec
Parameter-file for AddressModeller not yet available, wait another 5 sec
Parameter-file for AddressModeller not yet available, wait another 5 sec
Parameter-file for AddressModeller not yet available, wait another 5 sec
Parameter-file for AddressModeller not yet available, wait another 5 sec
Parameter-file for AddressModeller not yet available, wait another 5 sec
Parameter-file for AddressModeller not yet available, wait another 5 sec
Parameter-file for AddressModeller not yet available, wait another 5 sec
Parameter-file for AddressModeller not yet available, giving up!
Try loading Service dll 'Dll\MWEB_TraceApplication_Dll.dll':
"MWEB_TRACEAPP_CloseAndReopenTracefile" BAILED OUT (Code: E6890002): Cannot open
trace file: /Traces/MiniWeb.log
"MWEB_TRACE_ConfigChanged" BAILED OUT with code: E6890051: Failed: Error openin
g Trace file, giving up.
Trace output will be redirected to TraceApp...
OpcUa Server started successfully on 127.0.0.1:4840.
Ups no public Port :-<.

```

Figure 104 Running of OPC-UA Server in Sinutrain

6.5.2 Connecting with Sinutrain OPC-UA Server

If the installation of Sinutrain software and ConnectOPC-UA is on the same PC then OPC Url will be 'opc.tcp://localhost:4840' and if the installation are on different PC then 'localhost' must be replaced by the I.P. Address of the PC having Sinutrain Installed. In case of accessing the Sinutrain on different PC then Open UAApplication.xml with a text editor and replace all 'localhost' entries in this file with the IPv4 address of your PC. Fig-105 shows the sample machine configuration required for accessing the OPCUA server of Sinutrain. In this Configuration of Sinutrain Admin Username is 'OpcUaClient' and password is 'OpcUaClient', security policy and message security mode is selected as 'None' and the configuration need to be saved, before making it active for the further connection. Fig-106 shows the confirmation that the configuration of the machine is saved successfully.



The screenshot shows the 'connectopc_ua' application window. The title bar includes 'connectopc_ua'. The menu bar contains 'Application', 'Edit', 'Options', 'Favourites', and 'Settings'. The main window features a logo on the left and a 'Configure Machine' section on the right. A sidebar on the left lists various options: 'Configure Machine' (highlighted in red), 'Connect Machine', 'Browse Address Space', 'Monitor variables', 'Write variables to the server', 'Create Pub-Sub/Notifications', 'Multiple Pub-Sub/Notifications', 'Method Call (Address Space)', 'Event Notifier', 'File Transfer', 'Create OPC-UA Server', 'Settings', and 'Help'. The 'Configure Machine' section contains the following fields and controls:

Field/Control	Value
Machine Name	Sinutrain_lathe_machine
Machine Number	123
Machine Location	BLK2BAY1
OPC URL ADDRESS	opc.tcp://localhost:4840
Anonymous Connection	☐
Username	OpcUaClient
Password	
Security Policy	None
Message Security Mode	None

A green 'Save' button is located below the configuration fields.

Figure 105 Machine Configuration

After successful configuration of the machine, attempt should be made to connect with it by accessing the 'Connect Machine' option. 'Connect' button must be pressed, in order to create a connection and open a secure session between the client and Sinutrain's OPC-UA Server. User will be notified the status of the connection in 'Logs' section as shown in Fig-106. Server discovery is automatically attempted on the unsuccessful connection with the client, and correct settings for security policy and message security mode is provided, but the correct settings need to be configured again and fresh attempt for connection need to be made thereafter.

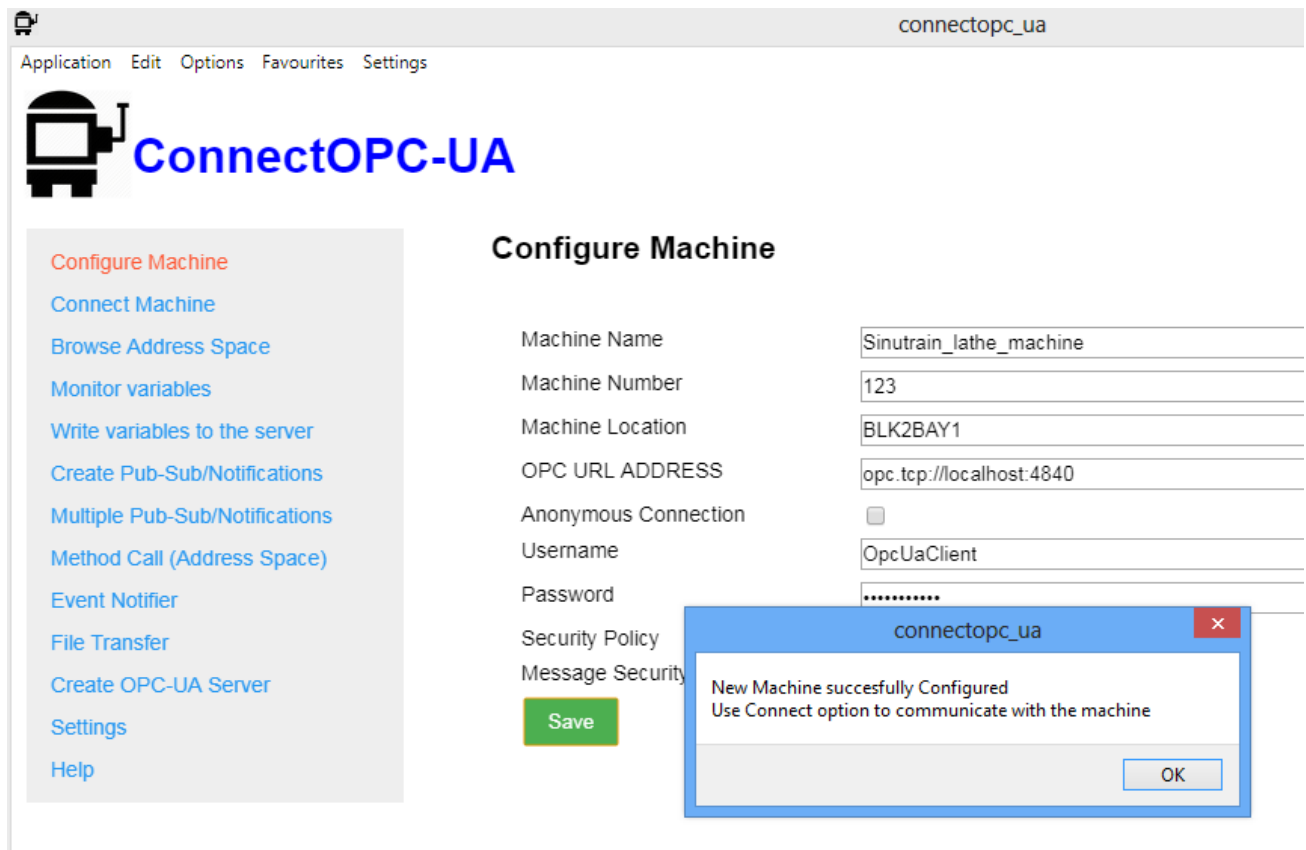


Figure 106 Configuration Saved

Anonymous support is available in the newest version of Siemens OPCUA Server, but it should be avoided for any production release application, anonymous connection can be used for the prototyping and connection check only. Any anonymous connection is prone to IT security threats and may hamper machine functioning adversely.

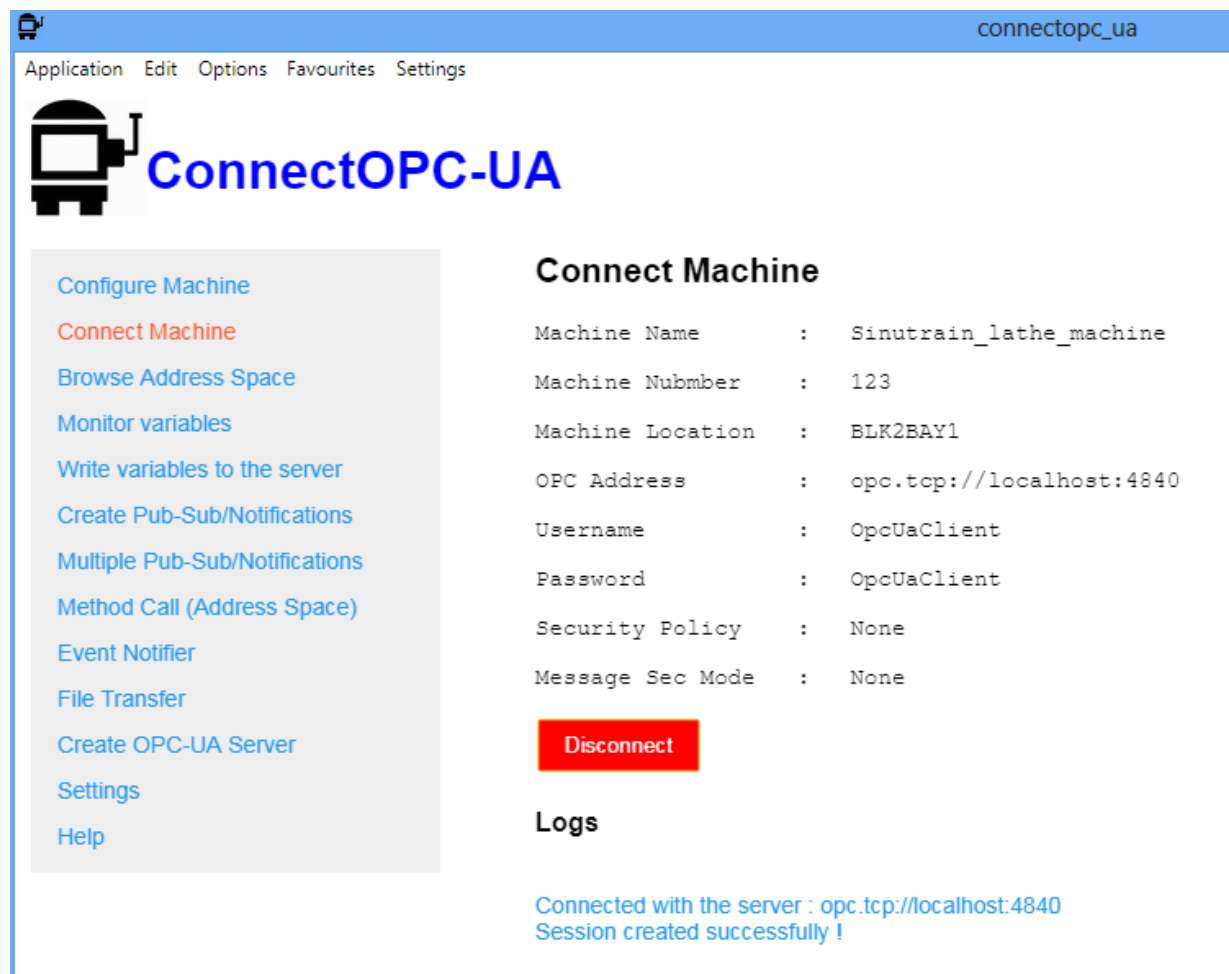


Figure 107 Successful connection of the software with Sinutrain

6.5.3 Browsing Address Space of Sinutrain's Configured Machine

After the successful connection with the configured machine in Sinutrain's software, the address space of the machine can be browsed as shown in Fig-108 .

If the OPC UA server is browsed, the available address space is mapped under the "Sinumerik"

node. Global User Data (GUD) can be found under the "/Sinumerik/GUD" node.

The PLC blocks (inputs, outputs, bit memory, data blocks) can be found under the "/Sinumerik/

Plc" node. Machine data can be found under the node "/Sinumerik/TEA". Setting data can be found under the node "/Sinumerik/SEA".

connectopc_ua

JA

Browse OPC-UA Address Space

Read Current Node :

Add to Favourites

ADDRESS SPACE

Download Address Space

The screenshot shows the OPC-UA Address Space browser interface. On the left, a tree view displays the hierarchy: Root > Server > Sinumerik. Under Sinumerik, there are several sub-nodes: Axis, Bag, Channel, DriveVsa, GUD, Hmi, Methods, Nck, Plc, Random, SEA, TEA, and Tool. The 'Channel' node is selected. On the right, the properties of the selected node are displayed in a table-like format:

Node Id	ns=2;s=Sinumerik
Display Name	Sinumerik
Browse Name	Sinumerik
Description	An object with several features of s
Node Class	Object
Event Notifier	0
Data Type	BadAttributeIdInvalid
Value	BadNotReadable
Access Level	

Figure 108 Browsing Address Space of Sinutrain Configured Machine

Observe the following while browsing:

- ➔ In the address space of the NC, the displayed variables always represent only the first parameter of the corresponding unit.

Example:

The R parameters can be found under "Sinumerik > Channel > Parameter > R". The corresponding identifier is called "/Channel/Parameter/R", which is finally mapped to "/Channel/Parameter/R[u1,1]". If you want to access other parameters, you need to specify the corresponding index in brackets, e. g. "/Channel/Parameter/R[u2,56]".

- ➔ In the address space of the PLC, the displayed variables represent the access format that has to be extended accordingly
Example: The variable "/Plc/MB" is in the address space and is mapped to "/Plc/MB0". For accessing further bytes this variable must be extended by the appropriate byte number, e.g. "/Plc/MB6".
- ➔ The address space of the NC also contains variables that are not available in a corresponding machine configuration. These variables return "BadAttributeIdInvalid" as value.

The variable paths for NC access are stored in the address space of the SINUMERIK Operate OPC UA server.

You can obtain additional information from the List Manual for 840D sl and 828D "NC variables and interface signals"

(<https://support.industry.siemens.com/cs/de/de/view/109748365/en>).

Any OPC UA client supporting Alarms & Conditions connected to the SINUMERIK OPC UA server can subscribe to alarms to get the notifications of alarms.

All OPC UA Clients that have subscribed for SINUMERIK alarms will be provided with an alarm as soon as it becomes active. Also if the alarm becomes inactive, the status of the corresponding alarm/s will be updated automatically.

Alarms and Conditions support subscription of all the pending and active alarms of the SINUMERIK system. Part program messages are not supported as part of Alarms and Conditions, but can be received using data access. The OPC UA Server provides all alarms that will be provided by the SINUMERIK AlarmService:

- ➔ HMI alarms
- ➔ NCK alarms including drive alarms
- ➔ Diagnostic buffer alarms
- ➔ PLC alarms (FC10)
- ➔ Alarm_S(Q) alarms (SFC17/18, PDiag, HiGraph, S7-Graph) with results of criteria analysis.

Multi language support for the alarms and warnings messages are supported and the required alarm language can be selected during session creation in OPC UA Client. If the desired language is not supported in the operate, the default English language is supported.

The SINUMERIK Alarm object is of the "CNCAAlarmType" which is defined in the Companion Specification "OPC UA Information Model for CNC Systems" (<http://opcfoundation.org/UA/CNC/>).

6.5.4 Monitoring PLC Variables

PLC variables can be found in the OPC UA server under the "/Sinumerik/Plc" node.

In the address space of the PLC, the displayed variables represent the access format that has to be extended accordingly.

Example

Syntax of the PLC variable is as follows: "/Plc/MB"

This variable must be extended by the appropriate byte number, e.g. to "/Plc/MB6".

Note: On SINUMERIK 828D, you can only access the freely definable customer data blocks from DB9000.

Area	Address	Permissible Data Type	OPCUA Data Type
Output Image	Qx.y	Bool	Boolean
	QBx	Byte, Char, String	UInt32, String
	QWx	Word, Char, Int	UInt32, Int32
	QDx	DWord, DInt, Real	UInt32, Int32, Double
Data Block	DBz.DBx.y	Bool	Boolean
	DBx.DBBy	Byte, Char, String	UInt32, String
	DBx.DBWy	Word, Char, Int	UInt32, Int32
	Dbx.DB Dy	DWord, DInt, Real	UInt32, Int32, Double
Input Image	Ix.y	Bool	Boolean
	IBx	Byte, Char, String	UInt32, String
	IWx	Word, Char, Int	UInt32, Int32
	IDx	DWord, DInt, Real	UInt32, Int32, Double
Bit Memory	Mx.y	Bool	Boolean
	MBx	Byte, Char, String	UInt32, String
	MWx	Word, Char, Int	UInt32, Int32
	MDx	DWord, DInt, Real	UInt32, Int32, Double
Counters	Cx	-	Byte
Timers	Tx	-	UInt32
PLC Time	Clock	-	UInt16

Read Variables

Variable Name (Full Qualified Node Id)

ns=2;s=/Plc/I0.0

Read

Logs

Value of ns=2;s=/Plc/I0.0 is : true

Server Time Stamp : Thu Aug 22 2019 14:51:25 GMT+0530 (India Standard Time)

Figure 109 PLC Input I0.0 status

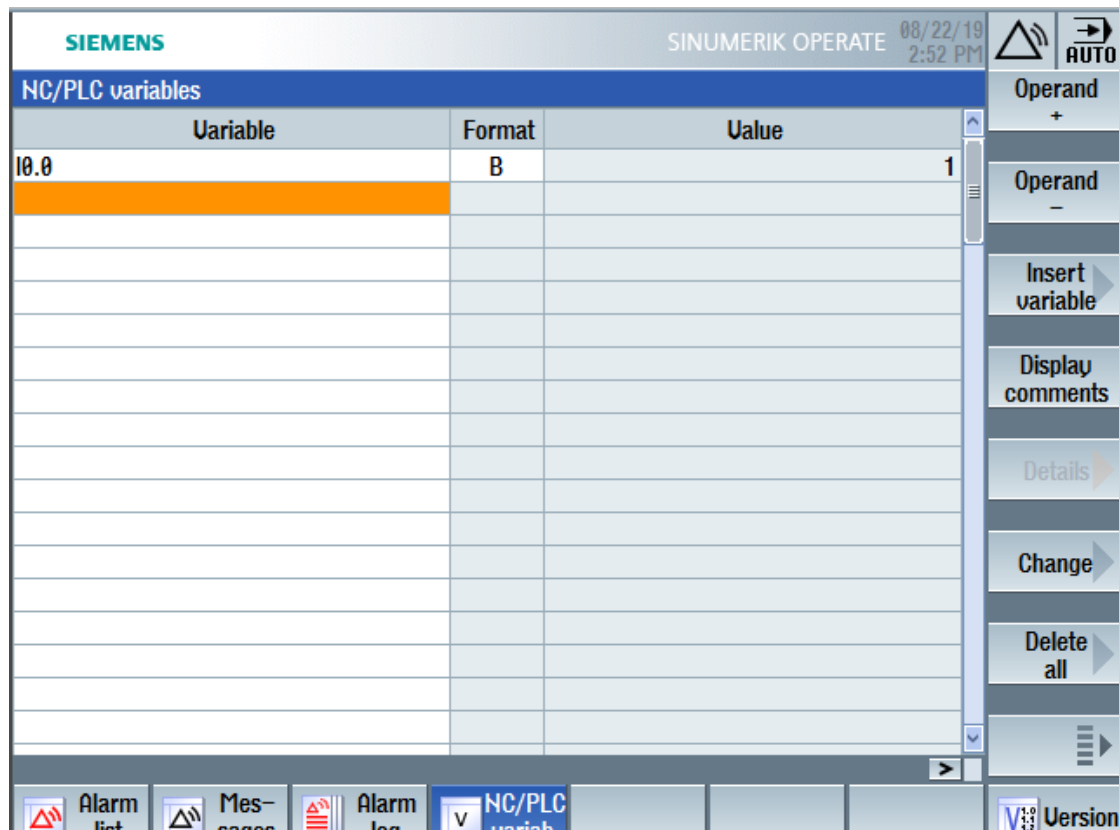


Figure 110 PLC IO.0 Value from Machine

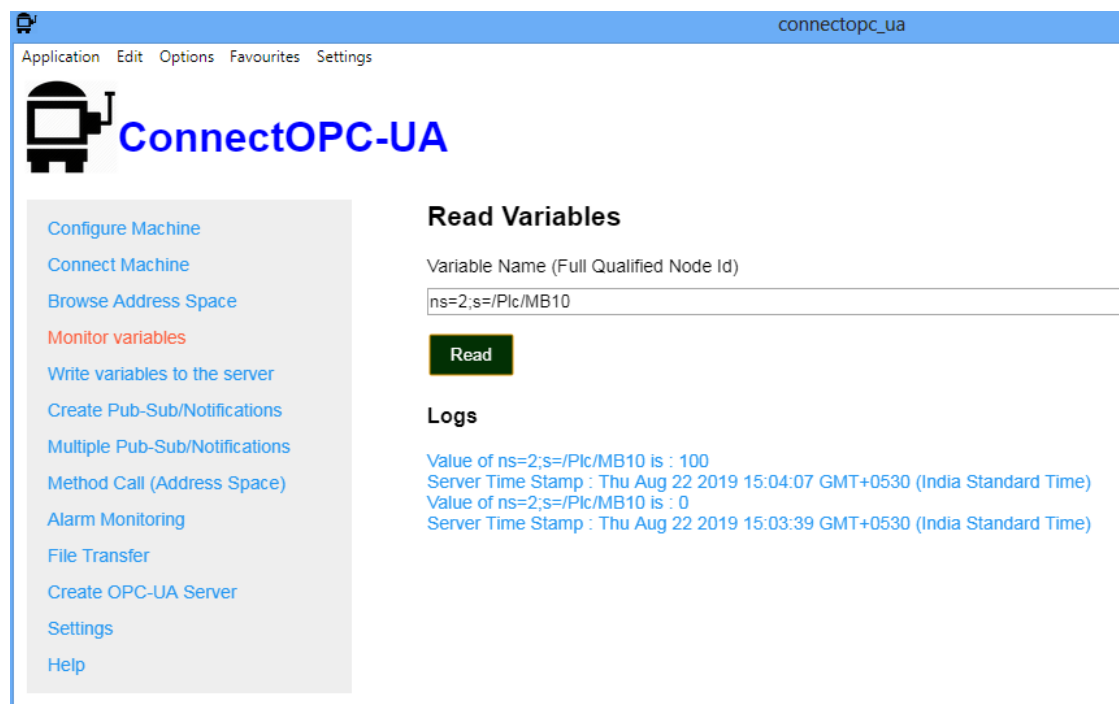


Figure 111 MB10 Value as read by the Software

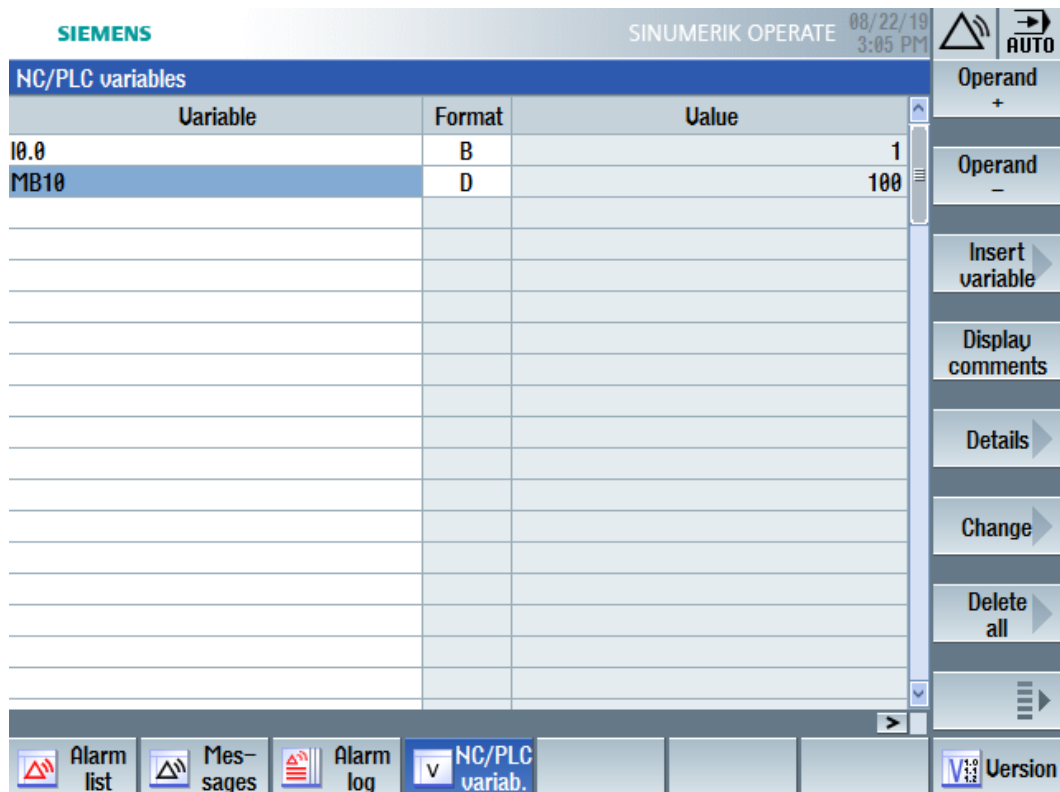


Figure 112 Status of MB10 on the Machine

6.5.5 Monitoring NC and other variables variables

NC variables are distributed in various nodes in Siemens OPC-UA server. Machine data can be found under the node "/Sinumerik/TEA". Setting data can be found under the node "/Sinumerik/SEA".

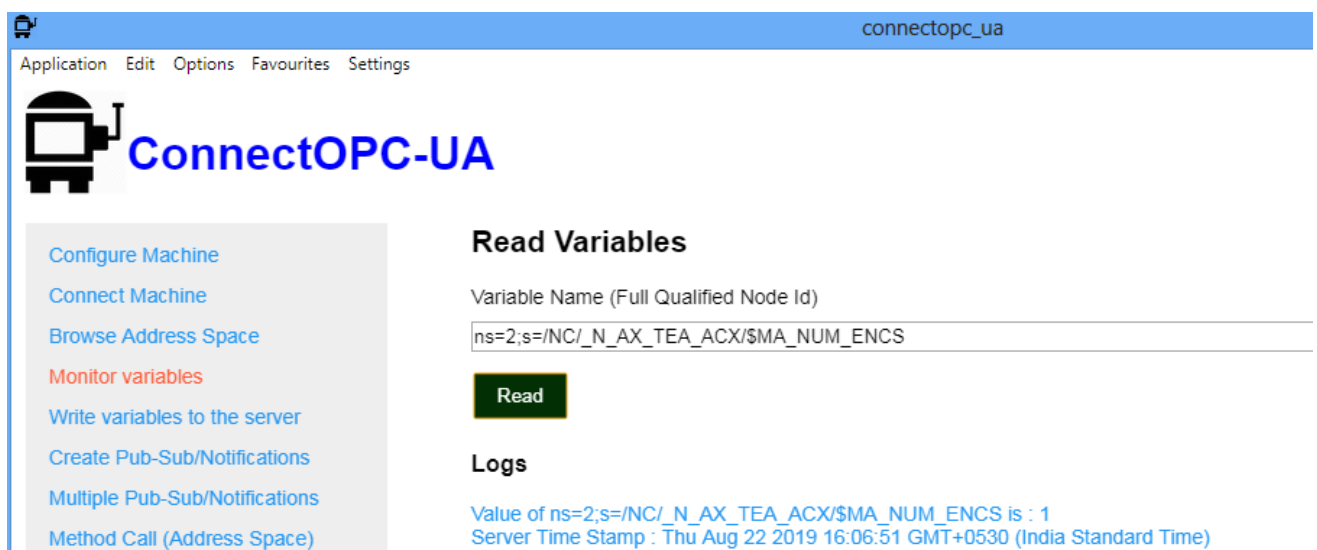


Figure 113 Reading Number of Encoders NC Variable

SIEMENS

SINUMERIK OPERATE

08/22/19
4:08 PM

Axis machine data

AX1:MX1

30110[0]	\$MA_CTRLOUT_MODULE_NR	1	po		Axis +
30120[0]	\$MA_CTRLOUT_NR	1	po		
30130[0]	\$MA_CTRLOUT_TYPE	0	po		Axis -
30132[0]	\$MA_IS_VIRTUAL_AX	0	po		
30134[0]	\$MA_IS_UNIPOLAR_OUTPUT	0	po		Select axis
30200	\$MA_NUM_ENCS	1	po		

Figure 114 NC-MD 30200 (Number of Encoders)

SinuTrain for SINUMERIK Operate - Lathe with driven tool (840D sl 4.7 SP6)

Machine View Extras Help

Add Node from Favourites

Application Edit Options Favourites Settings

ConnectOPC-UA

opc.tcp://localhost:4840

- Configure Machine
- Connect Machine
- Browse Address Space
- Monitor variables
- Write variables to the server
- Create Pub-Sub/Notifications
- Multiple Pub-Sub/Notifications
- Method Call (Address Space)
- Alarm Monitoring
- File Transfer
- Create OPC-UA Server
- Settings
- Help

Read Variables

Variable Name (Full Qualified Node Id)

ns=2;s=/Channel/ProgramInfo/actPartProgram

Read

Logs

Value of ns=2;s=/Channel/ProgramInfo/actPartProgram is :
N10 F_HEAD(349175807,28.1,-150,2,2,10,2,249.8,375.772,0.5,4000.,71,1,101023,-120.,20,6,0,0,0,0,1,0,".....0,0,0,0,0,0,.....");"RO"
;Example by CNC4YOU (www.cnc4you.siemens.com)
N2
Server Time Stamp : Fri Aug 23 2019 07:43:30 GMT+0530 (India Standard Time)

Figure 115 Actual Part Number

SinuTrain for SINUMERIK Operate - Lathe with driven tool (84)

Machine View Extras Help

Add Node from Favourites

Application Edit Options Favourites Settings

ConnectOPC-UA

Configure Machine
Connect Machine
Browse Address Space
Monitor variables
Write variables to the server
Create Pub-Sub/Notifications
Multiple Pub-Sub/Notifications
Method Call (Address Space)
Alarm Monitoring
File Transfer

Read Variables

Variable Name (Full Qualified Node Id)

ns=2;s=/Channel/State/feedRateIpoOvr

Read

Logs

Value of ns=2;s=/Channel/State/feedRateIpoOvr is : 114.99999761581421
Server Time Stamp : Fri Aug 23 2019 07:48:12 GMT+0530 (India Standard Time)
Value of ns=2;s=/Channel/State/feedRateIpoOvr is : 0
Server Time Stamp : Fri Aug 23 2019 07:47:52 GMT+0530 (India Standard Time)

Figure 116 Feedoverride from Machine

Add Node from Favourites

Application Edit Options Favourites Settings

ConnectOPC-UA

Configure Machine
Connect Machine
Browse Address Space
Monitor variables
Write variables to the server
Create Pub-Sub/Notifications
Multiple Pub-Sub/Notifications
Method Call (Address Space)
Alarm Monitoring
File Transfer
Create OPC-UA Server
Settings

Read Variables

Variable Name (Full Qualified Node Id)

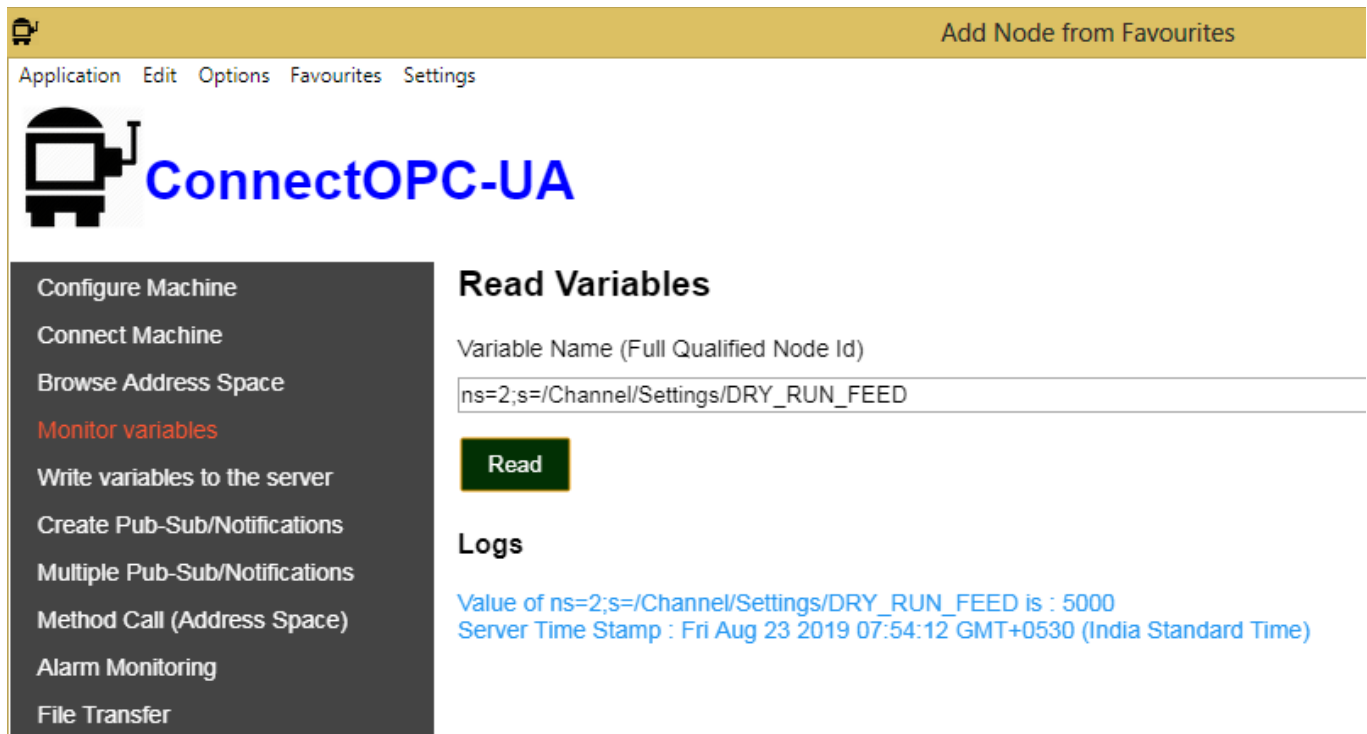
ns=2;s=/Channel/Spindle/speedOvr

Read

Logs

Value of ns=2;s=/Channel/Spindle/speedOvr is : 50
Server Time Stamp : Fri Aug 23 2019 07:50:26 GMT+0530 (India Standard Time)

Figure 117 Spindle Override from Machine



Application Edit Options Favourites Settings

ConnectOPC-UA

- Configure Machine
- Connect Machine
- Browse Address Space
- Monitor variables**
- Write variables to the server
- Create Pub-Sub/Notifications
- Multiple Pub-Sub/Notifications
- Method Call (Address Space)
- Alarm Monitoring
- File Transfer

Read Variables

Variable Name (Full Qualified Node Id)

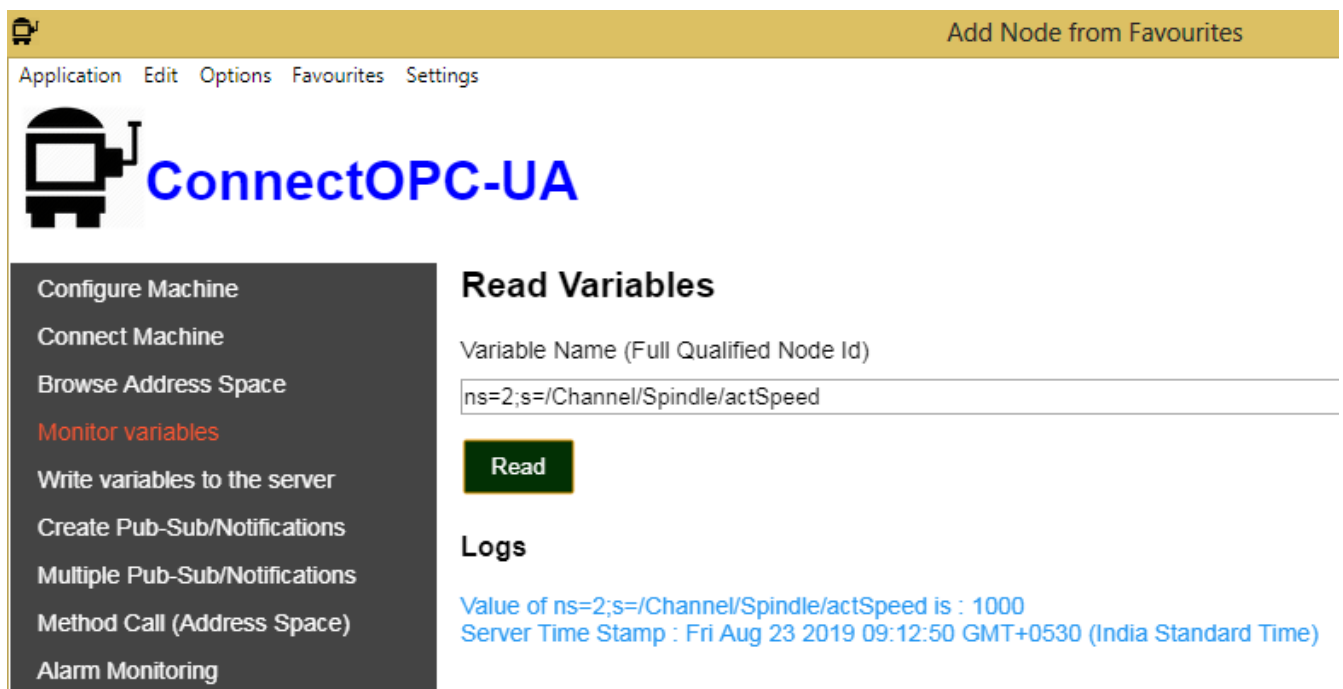
ns=2;s=/Channel/Settings/DRY_RUN_FEED

Read

Logs

Value of ns=2;s=/Channel/Settings/DRY_RUN_FEED is : 5000
Server Time Stamp : Fri Aug 23 2019 07:54:12 GMT+0530 (India Standard Time)

Figure 118 Dry Run Feed



Application Edit Options Favourites Settings

ConnectOPC-UA

- Configure Machine
- Connect Machine
- Browse Address Space
- Monitor variables**
- Write variables to the server
- Create Pub-Sub/Notifications
- Multiple Pub-Sub/Notifications
- Method Call (Address Space)
- Alarm Monitoring

Read Variables

Variable Name (Full Qualified Node Id)

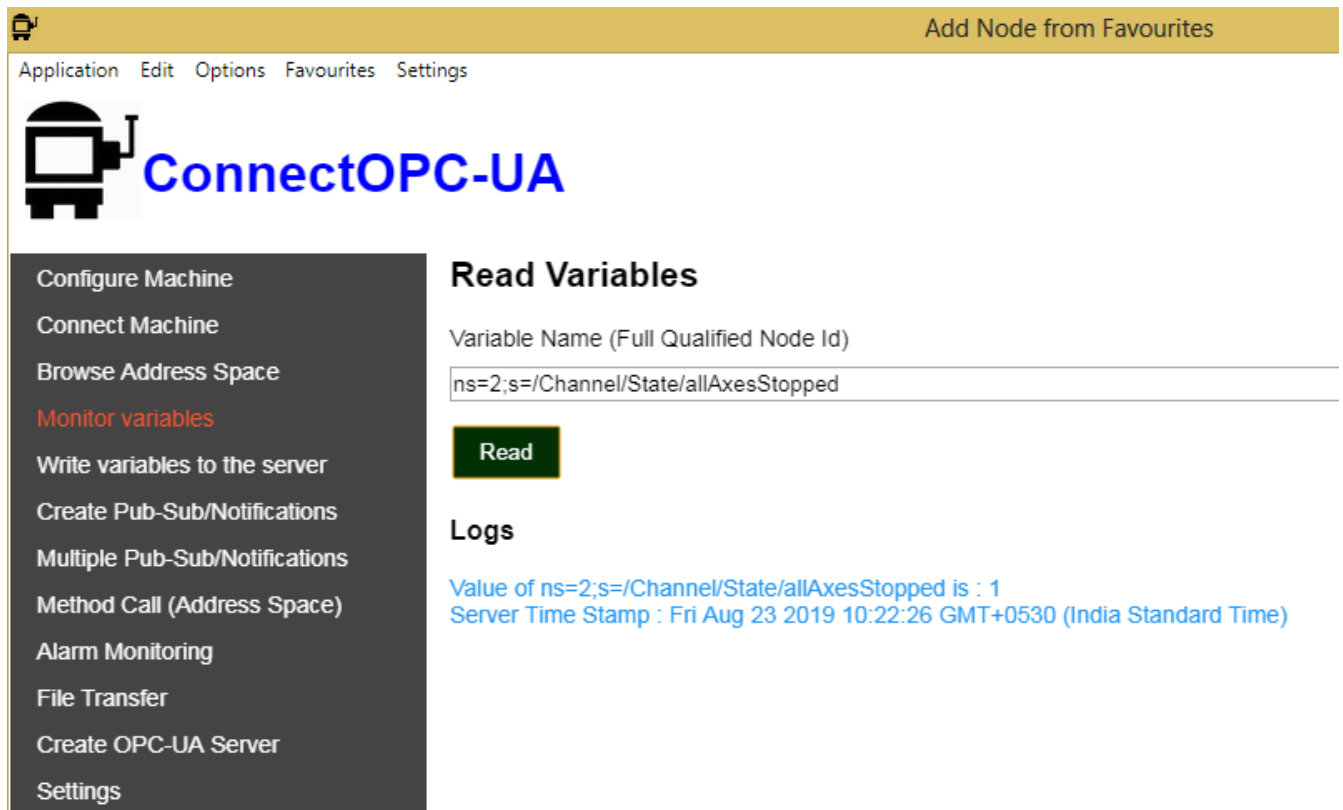
ns=2;s=/Channel/Spindle/actSpeed

Read

Logs

Value of ns=2;s=/Channel/Spindle/actSpeed is : 1000
Server Time Stamp : Fri Aug 23 2019 09:12:50 GMT+0530 (India Standard Time)

Figure 119 Spindle Actual Speed



Application Edit Options Favourites Settings

ConnectOPC-UA

Configure Machine
Connect Machine
Browse Address Space
Monitor variables
Write variables to the server
Create Pub-Sub/Notifications
Multiple Pub-Sub/Notifications
Method Call (Address Space)
Alarm Monitoring
File Transfer
Create OPC-UA Server
Settings

Read Variables

Variable Name (Full Qualified Node Id)

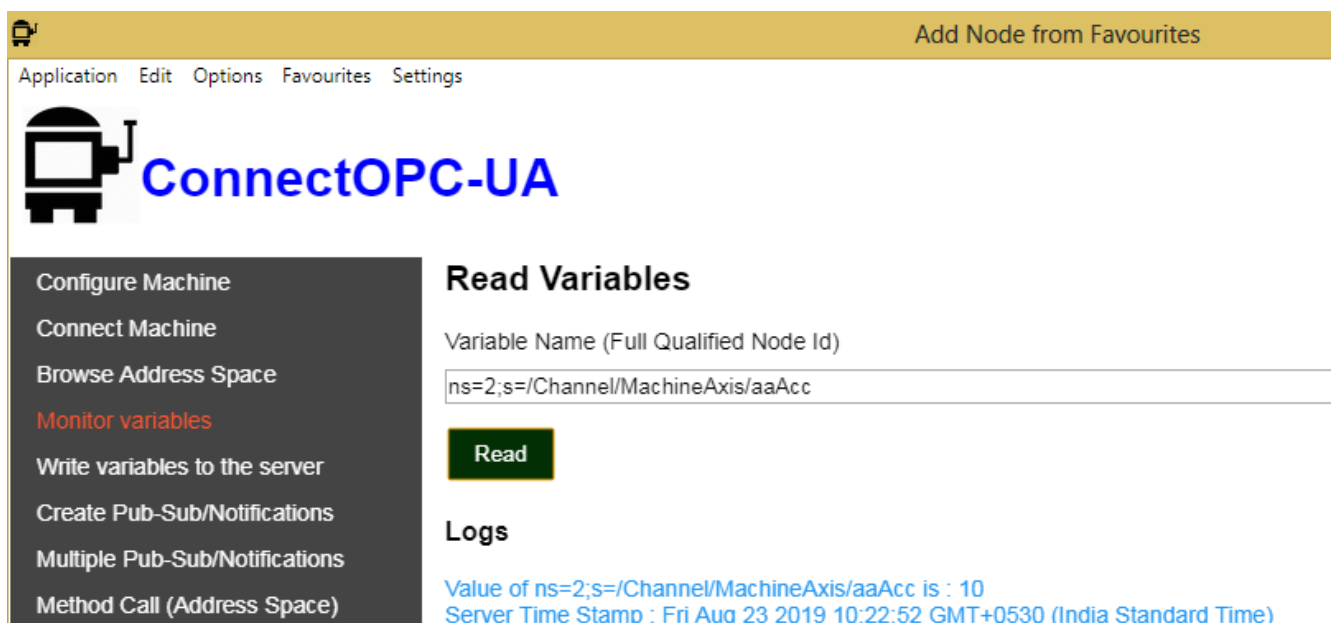
ns=2;s=/Channel/State/allAxesStopped

Read

Logs

Value of ns=2;s=/Channel/State/allAxesStopped is : 1
Server Time Stamp : Fri Aug 23 2019 10:22:26 GMT+0530 (India Standard Time)

Figure 120 All Axes Stopped Signal



Application Edit Options Favourites Settings

ConnectOPC-UA

Configure Machine
Connect Machine
Browse Address Space
Monitor variables
Write variables to the server
Create Pub-Sub/Notifications
Multiple Pub-Sub/Notifications
Method Call (Address Space)

Read Variables

Variable Name (Full Qualified Node Id)

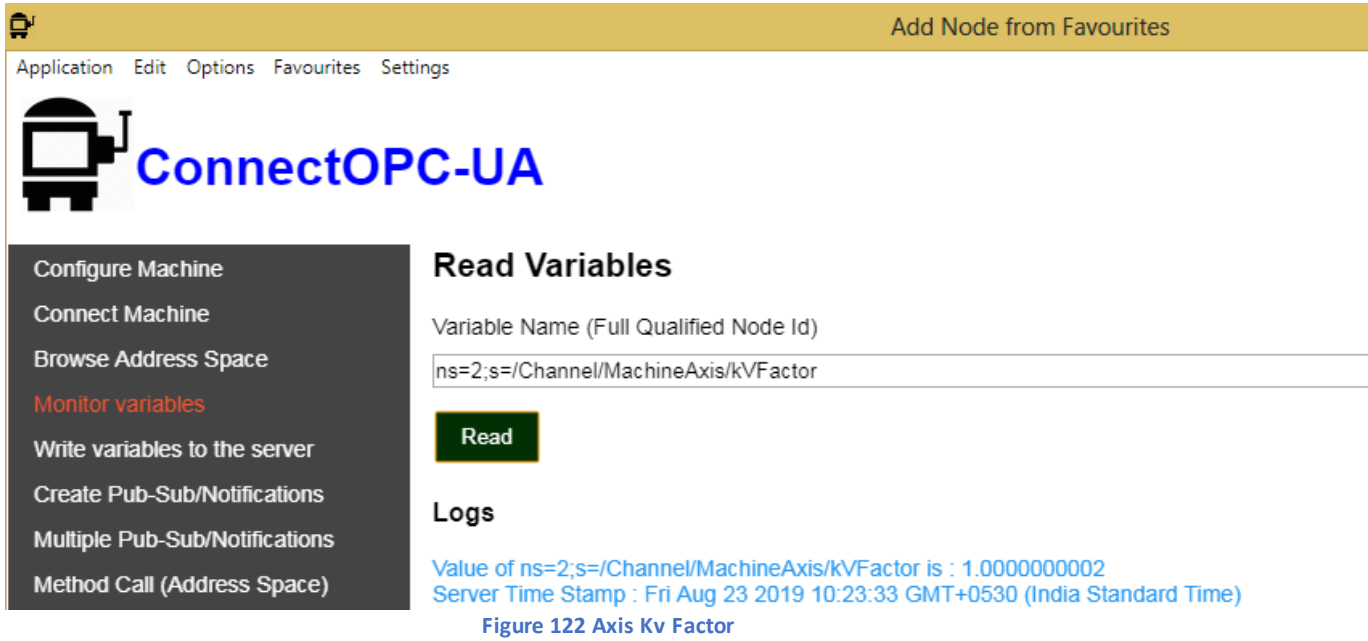
ns=2;s=/Channel/MachineAxis/aaAcc

Read

Logs

Value of ns=2;s=/Channel/MachineAxis/aaAcc is : 10
Server Time Stamp : Fri Aug 23 2019 10:22:52 GMT+0530 (India Standard Time)

Figure 121 Axes Acceleration



ConnectOPC-UA interface showing the 'Read Variables' section. The variable name is set to 'ns=2;s=/Channel/MachineAxis/kVFactor'. The 'Read' button is highlighted. The logs show the value of the variable is 1.0000000002, with a server time stamp of Fri Aug 23 2019 10:23:33 GMT+0530 (India Standard Time).

Application Edit Options Favourites Settings

ConnectOPC-UA

- Configure Machine
- Connect Machine
- Browse Address Space
- Monitor variables**
- Write variables to the server
- Create Pub-Sub/Notifications
- Multiple Pub-Sub/Notifications
- Method Call (Address Space)

Read Variables

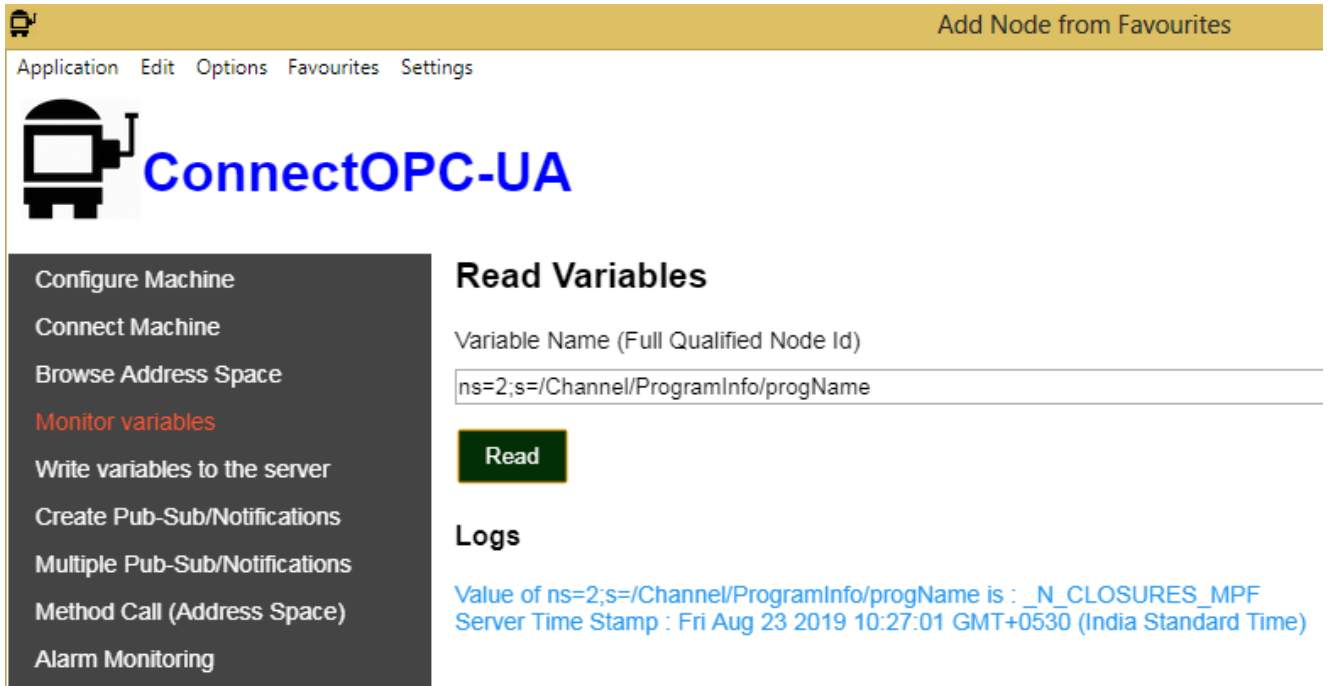
Variable Name (Full Qualified Node Id)

ns=2;s=/Channel/MachineAxis/kVFactor

Read

Logs

Value of ns=2;s=/Channel/MachineAxis/kVFactor is : 1.0000000002
 Server Time Stamp : Fri Aug 23 2019 10:23:33 GMT+0530 (India Standard Time)
 Figure 122 Axis Kv Factor



ConnectOPC-UA interface showing the 'Read Variables' section. The variable name is set to 'ns=2;s=/Channel/ProgramInfo/progName'. The 'Read' button is highlighted. The logs show the value of the variable is '_N_CLOSURES_MPF', with a server time stamp of Fri Aug 23 2019 10:27:01 GMT+0530 (India Standard Time).

Application Edit Options Favourites Settings

ConnectOPC-UA

- Configure Machine
- Connect Machine
- Browse Address Space
- Monitor variables**
- Write variables to the server
- Create Pub-Sub/Notifications
- Multiple Pub-Sub/Notifications
- Method Call (Address Space)
- Alarm Monitoring

Read Variables

Variable Name (Full Qualified Node Id)

ns=2;s=/Channel/ProgramInfo/progName

Read

Logs

Value of ns=2;s=/Channel/ProgramInfo/progName is : _N_CLOSURES_MPF
 Server Time Stamp : Fri Aug 23 2019 10:27:01 GMT+0530 (India Standard Time)

Figure 123 Current Active Program Name

Application Edit Options Favourites Settings

 **ConnectOPC-UA**

Configure Machine
Connect Machine
Browse Address Space
Monitor variables
Write variables to the server
Create Pub-Sub/Notifications
Multiple Pub-Sub/Notifications
Method Call (Address Space)

Read Variables

Variable Name (Full Qualified Node Id)

ns=2;s=/Channel/State/actParts

Read

Logs

Value of ns=2;s=/Channel/State/actParts is : 6
Server Time Stamp : Fri Aug 23 2019 10:27:21 GMT+0530 (India Standard Time)
Figure 124 Actual Parts Machined

Application Edit Options Favourites Settings

 **ConnectOPC-UA**

Configure Machine
Connect Machine
Browse Address Space
Monitor variables
Write variables to the server
Create Pub-Sub/Notifications
Multiple Pub-Sub/Notifications
Method Call (Address Space)

Read Variables

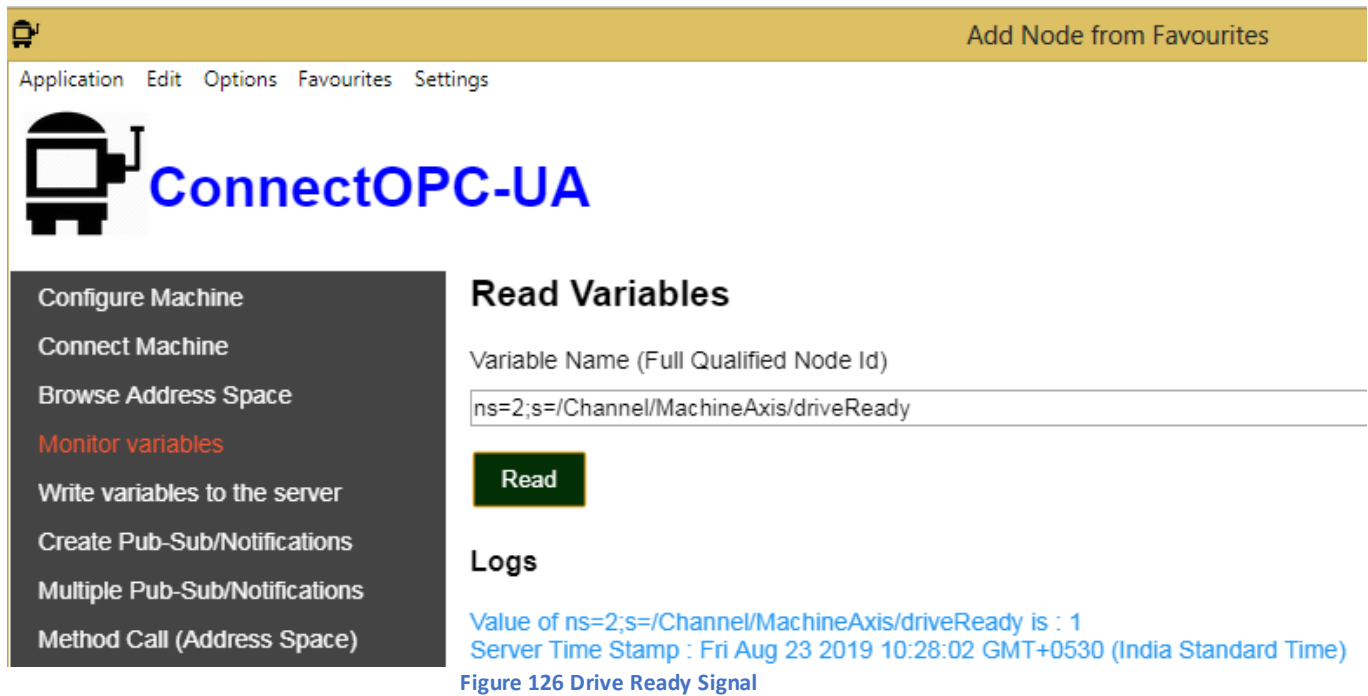
Variable Name (Full Qualified Node Id)

ns=2;s=/Channel/MachineAxis/aalm

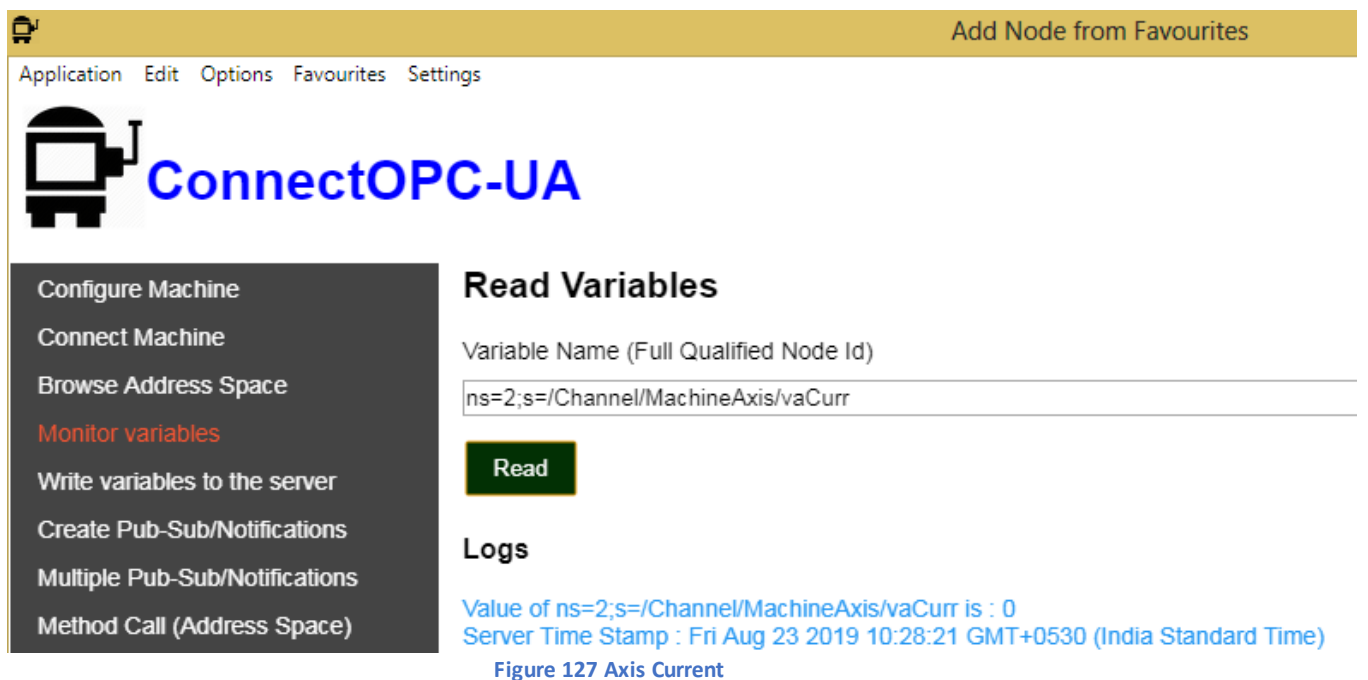
Read

Logs

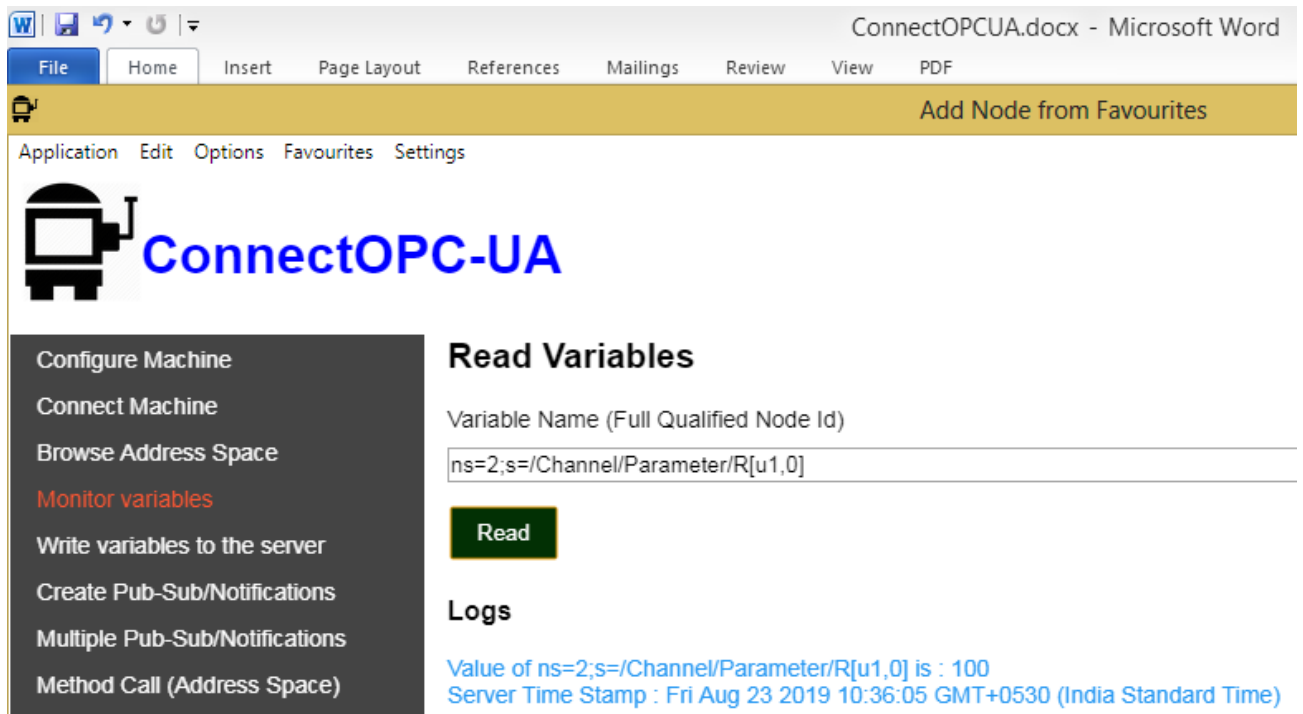
Value of ns=2;s=/Channel/MachineAxis/aalm is : 249.8
Server Time Stamp : Fri Aug 23 2019 10:27:45 GMT+0530 (India Standard Time)
Figure 125 Actual Position X Axis



This screenshot shows the ConnectOPC-UA application interface. The top bar is yellow with a robot icon and the text 'Add Node from Favourites'. Below it is a menu bar with 'Application', 'Edit', 'Options', 'Favourites', and 'Settings'. The main header area features a robot icon and the title 'ConnectOPC-UA' in blue. On the left is a dark grey sidebar with a list of options: 'Configure Machine', 'Connect Machine', 'Browse Address Space', 'Monitor variables' (highlighted in orange), 'Write variables to the server', 'Create Pub-Sub/Notifications', 'Multiple Pub-Sub/Notifications', and 'Method Call (Address Space)'. The main content area is titled 'Read Variables' and contains a text input field for 'Variable Name (Full Qualified Node Id)' with the value 'ns=2;s=/Channel/MachineAxis/driveReady'. Below the input is a green 'Read' button. Under the 'Logs' section, it displays 'Value of ns=2;s=/Channel/MachineAxis/driveReady is : 1' and 'Server Time Stamp : Fri Aug 23 2019 10:28:02 GMT+0530 (India Standard Time)', followed by the caption 'Figure 126 Drive Ready Signal'.



This screenshot shows the ConnectOPC-UA application interface, similar to the first one. The top bar is yellow with a robot icon and the text 'Add Node from Favourites'. Below it is a menu bar with 'Application', 'Edit', 'Options', 'Favourites', and 'Settings'. The main header area features a robot icon and the title 'ConnectOPC-UA' in blue. On the left is a dark grey sidebar with a list of options: 'Configure Machine', 'Connect Machine', 'Browse Address Space', 'Monitor variables' (highlighted in orange), 'Write variables to the server', 'Create Pub-Sub/Notifications', 'Multiple Pub-Sub/Notifications', and 'Method Call (Address Space)'. The main content area is titled 'Read Variables' and contains a text input field for 'Variable Name (Full Qualified Node Id)' with the value 'ns=2;s=/Channel/MachineAxis/vaCurr'. Below the input is a green 'Read' button. Under the 'Logs' section, it displays 'Value of ns=2;s=/Channel/MachineAxis/vaCurr is : 0' and 'Server Time Stamp : Fri Aug 23 2019 10:28:21 GMT+0530 (India Standard Time)', followed by the caption 'Figure 127 Axis Current'.



ConnectOPCUA.docx - Microsoft Word

File Home Insert Page Layout References Mailings Review View PDF

Add Node from Favourites

Application Edit Options Favourites Settings

ConnectOPC-UA

- Configure Machine
- Connect Machine
- Browse Address Space
- Monitor variables**
- Write variables to the server
- Create Pub-Sub/Notifications
- Multiple Pub-Sub/Notifications
- Method Call (Address Space)

Read Variables

Variable Name (Full Qualified Node Id)

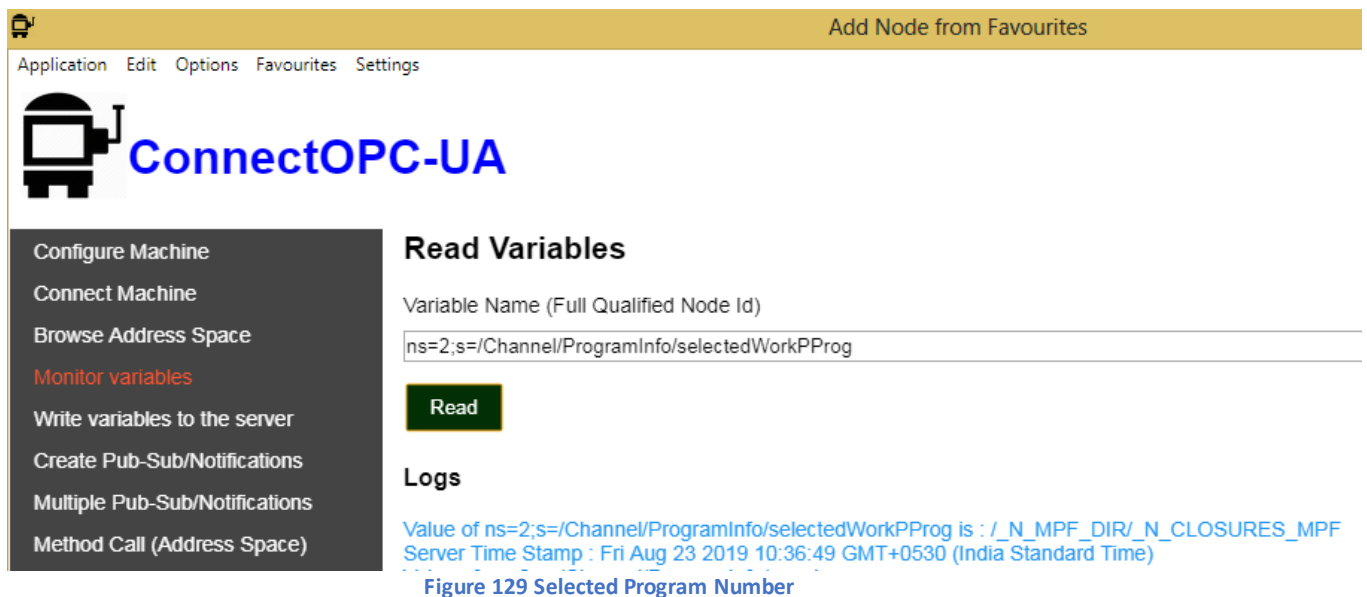
ns=2;s=/Channel/Parameter/R[u1,0]

Read

Logs

Value of ns=2;s=/Channel/Parameter/R[u1,0] is : 100
 Server Time Stamp : Fri Aug 23 2019 10:36:05 GMT+0530 (India Standard Time)

Figure 128 Value of R0 parameter in Channel 1



Add Node from Favourites

Application Edit Options Favourites Settings

ConnectOPC-UA

- Configure Machine
- Connect Machine
- Browse Address Space
- Monitor variables**
- Write variables to the server
- Create Pub-Sub/Notifications
- Multiple Pub-Sub/Notifications
- Method Call (Address Space)

Read Variables

Variable Name (Full Qualified Node Id)

ns=2;s=/Channel/ProgramInfo/selectedWorkPPProg

Read

Logs

Value of ns=2;s=/Channel/ProgramInfo/selectedWorkPPProg is : /_N_MPF_DIR/_N_CLOSURES_MPF
 Server Time Stamp : Fri Aug 23 2019 10:36:49 GMT+0530 (India Standard Time)

Figure 129 Selected Program Number

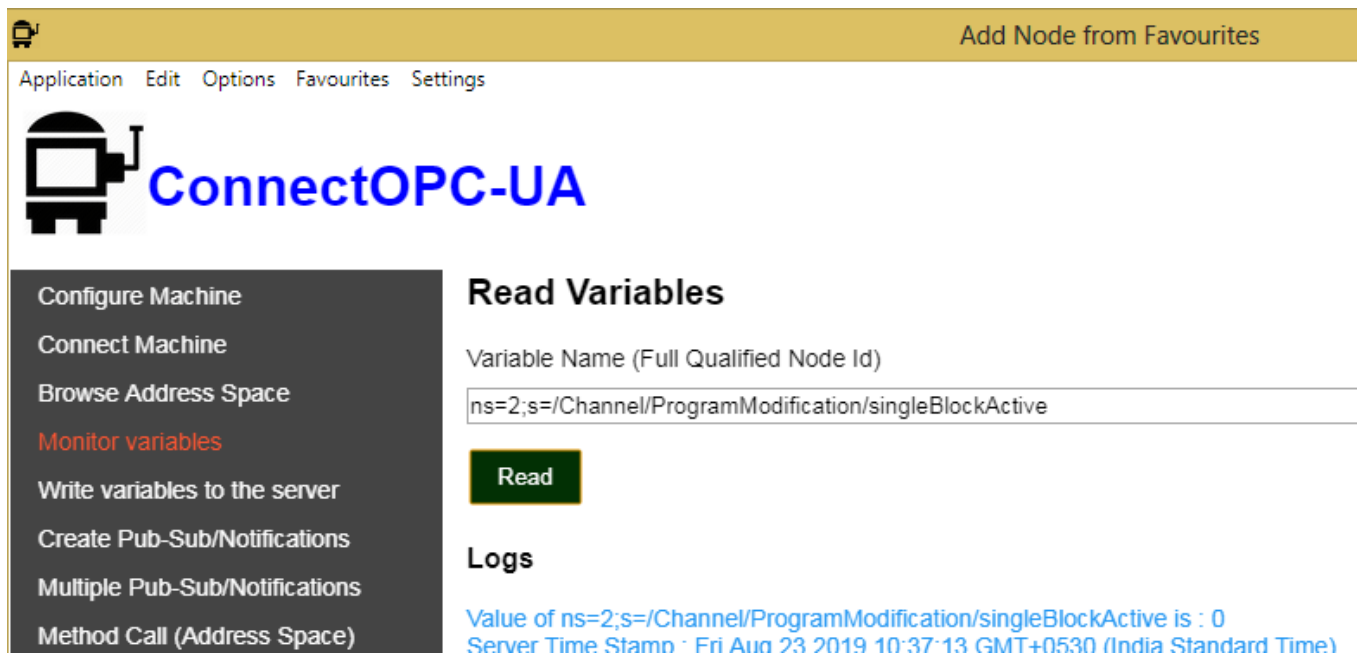


Figure 130 Single Block Active status

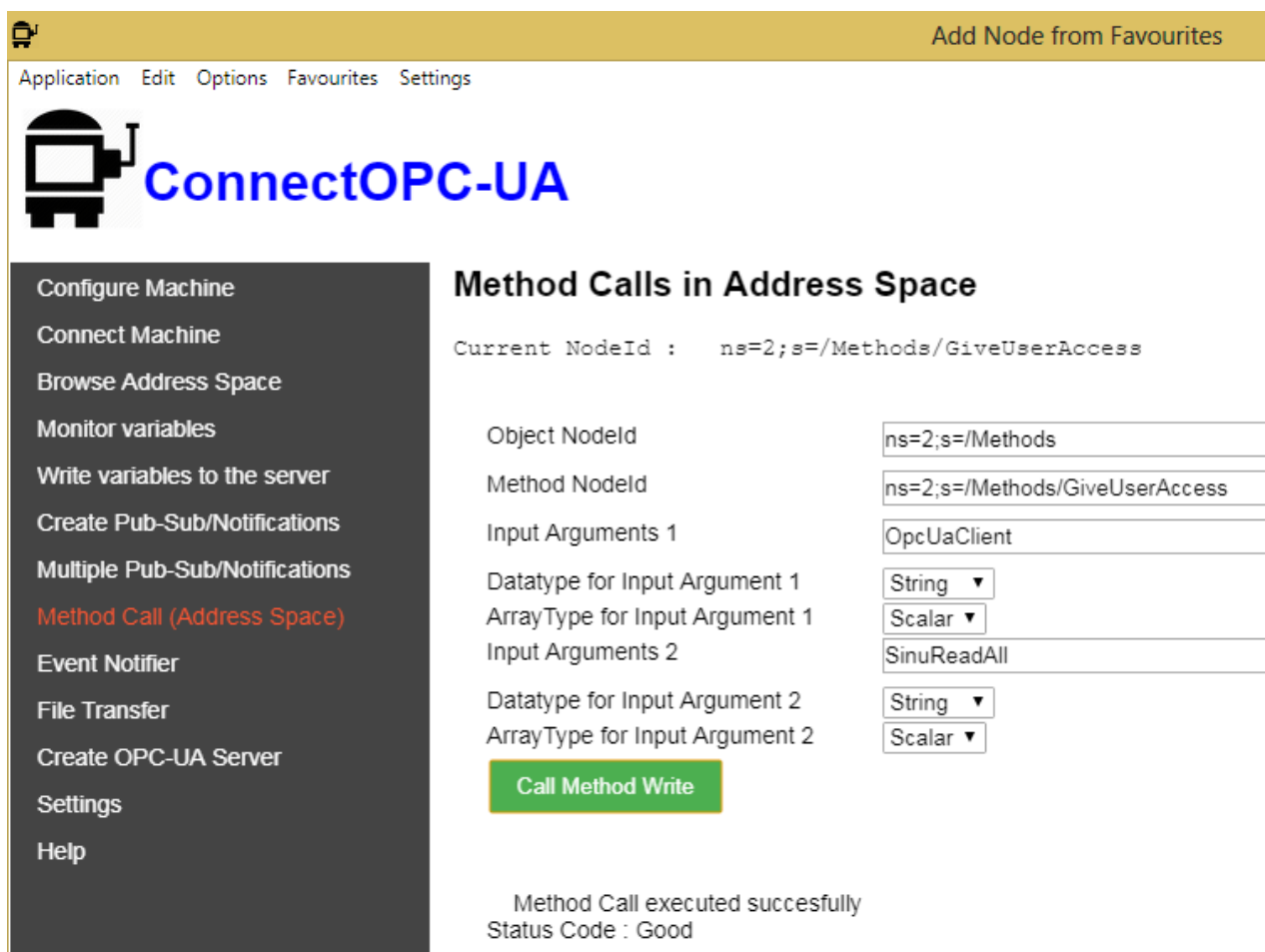


Figure 131 Permission for reading all nodes in Sinumerik

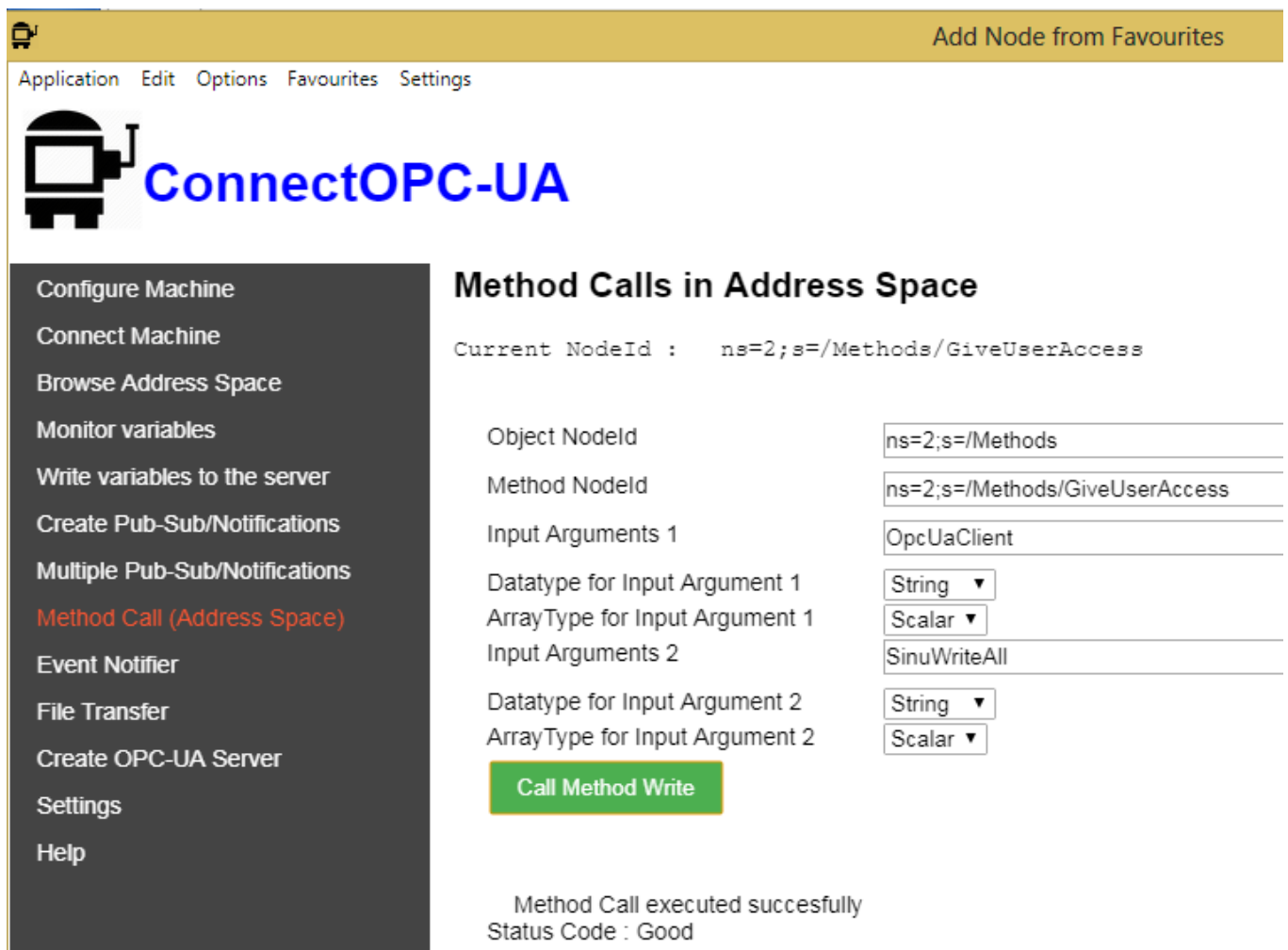


Figure 132 Granting write permission to all nodes in Sinumerik to user 'OpcUaClient'

6.5.6 Subscription of Variables

An OPC UA client can subscribe to a selection of nodes of interest and let the server monitor these items. Only in case of changes, e.g. to their values, the server notifies the client about such changes. This mechanism reduces the amount of transferred data immensely. Besides the reduction of bandwidth this mechanism introduces further advantages and is the recommended mechanism to “read” information from a UA server.

A client can subscribe to different types of information provided by an OPC UA server. The purpose of a subscription is to group these sources of information, called monitored items, together, forming a piece of information called a notification.

A subscription consists of at least one monitored item, which has to be created within the context of a session and can be transferred to another session. To create a session, a secure channel between the client and the server has to be established.

There are two different types of “changes” a client can subscribe to when adding monitored items to the subscription:

- Subscribe to data changes of Variable Values (Value attribute of a Variable)
- Subscribe to Events of Objects (EventNotifier attribute of an Object)

Fig-__ shows the spindle override for channel 1

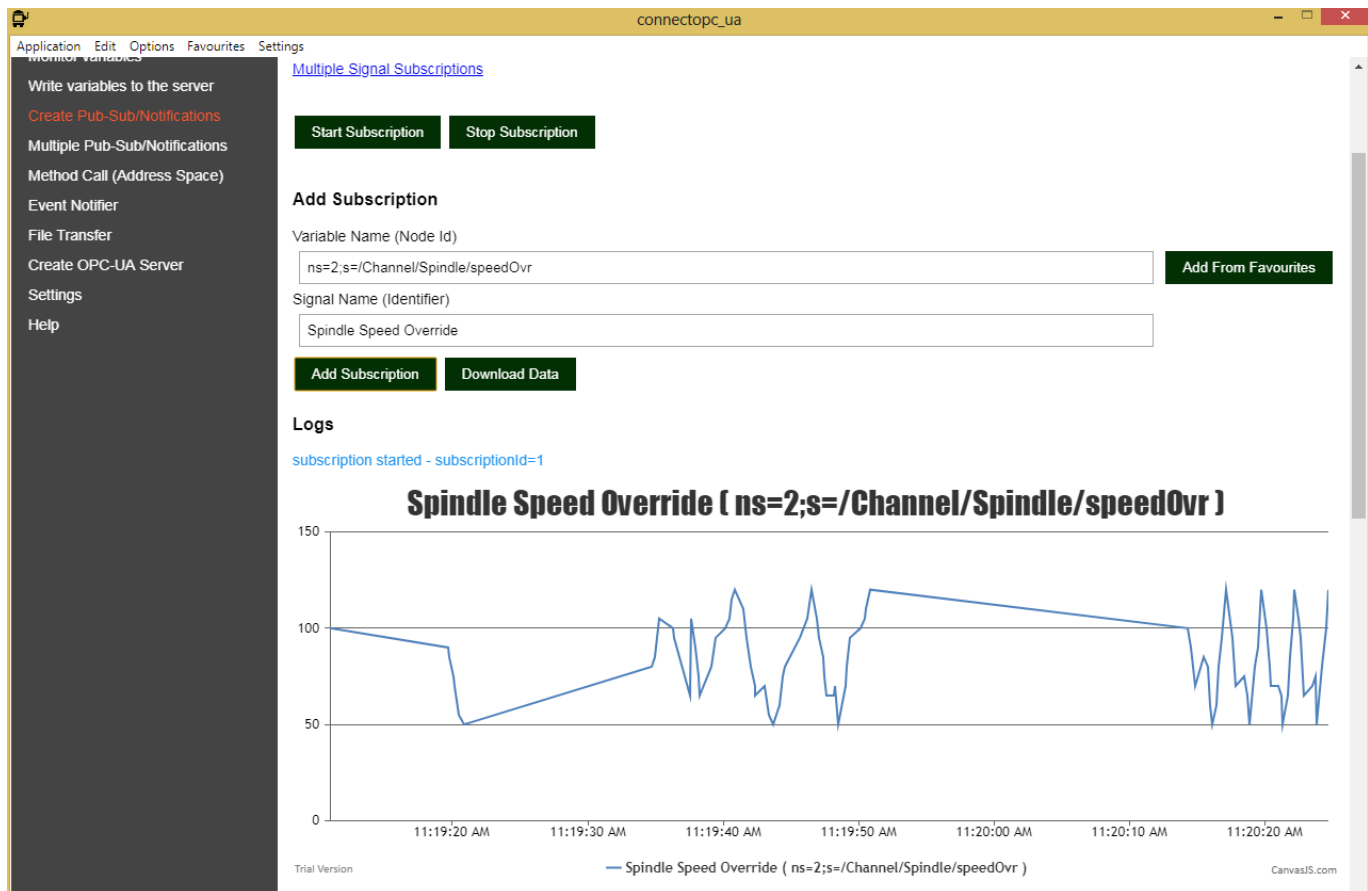


Figure 133 Monitoring of Spindle Override

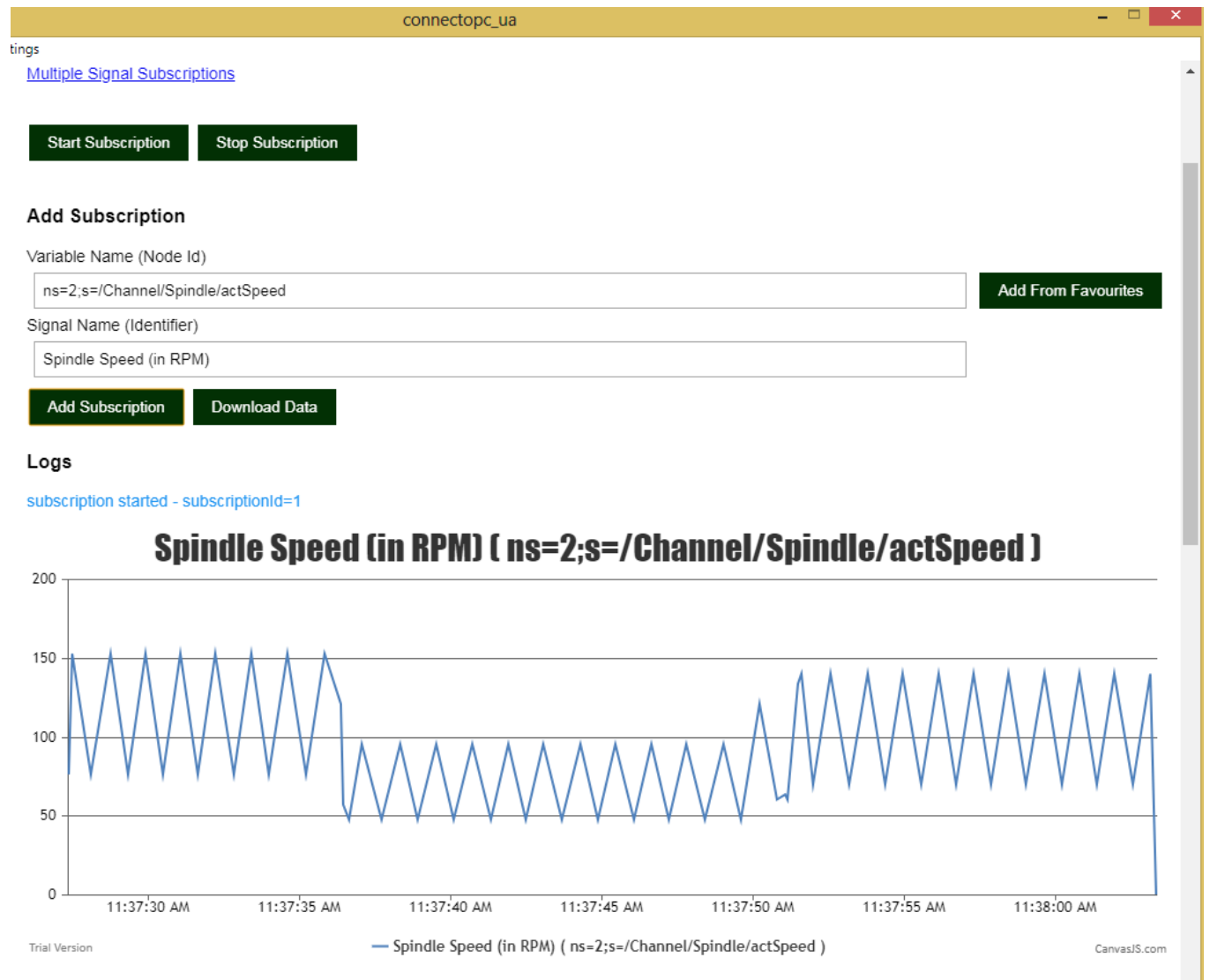


Figure 134 Spindle Speed Variation wrt to Spindle Override and Part Program

Part Program :

```
block1:
M03 S100
G4F23
M03 S200
G4F23
GOTOB block1
```

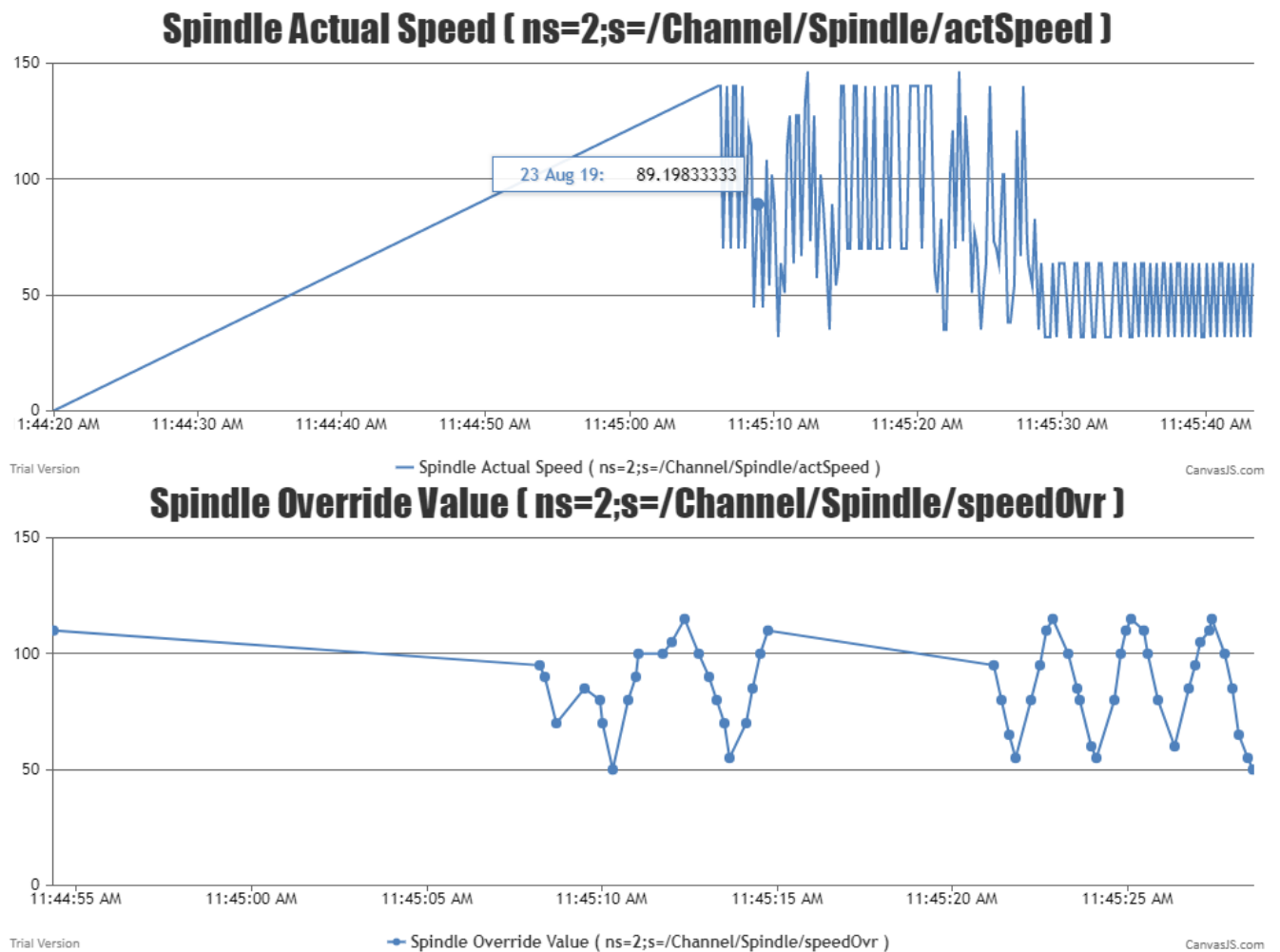


Figure 135 Spindle Actual Speed vs Spindle Override

6.5.7 File Transfer

The OPC UA server allows the OPC UA client to support the transfer of files between the client and the server.

The two methods `CopyFileToServer` and `CopyFileFromServer` and also standard file system methods need the following access rights:

- `FsRead`
- `FsWrite`

As a user, you will require user access rights to access these files from the server. The access rights are provided using the "GiveUserAccess" method. The following access rights can be provided for the file system.

- `FsRead` for the standard file system methods like `Open`, `GetPosition`, `Read`.
- `FsWrite` for the standard file system methods like `CreateDirecotry`, `CreateFile`, `Delete`, `MoveOrCopy`, `Write`, `SetPosition`, `Close`.

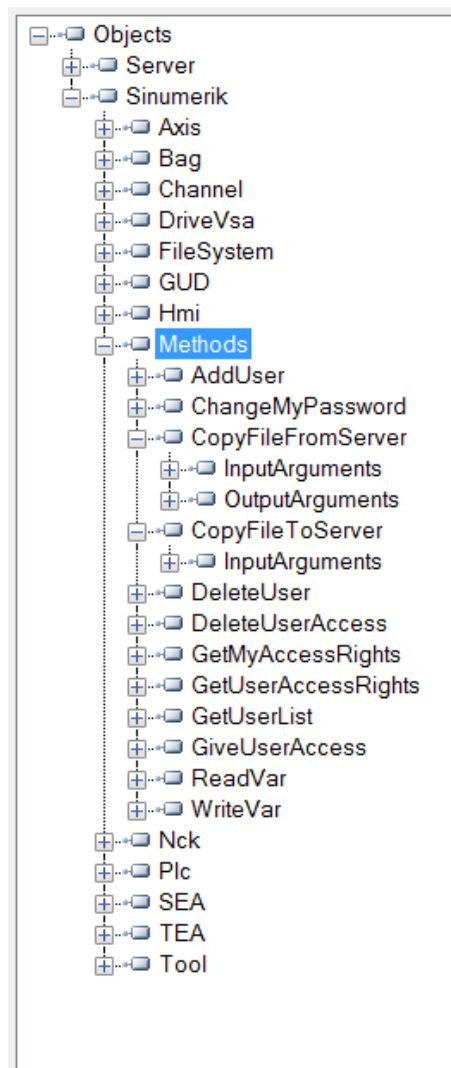


Figure 136 CopyFileFromServer & CopyFileToServer Methods

The SINUMERIK OPC UA server supports the "FileType"/"FolderType" as described in the OPC UA Specification Part 5, which allows manipulating files and folders via OPC UA. The folders, "Part Programs", "Sub Programs" and "Work Pieces" are of the "FolderType" type, which contain the following methods:

Methods/Attributes	Description
Create Directory	To Create New Folder under parent folder
Create File	To create new file under parent folder
Delete File	Delete a folder
Move or Copy	copy or move files from source to destination

You can create, delete, move or copy folders and files using the above methods. When you create a new folder using "CreateDirectory" , a new node will be created with "FolderType" and name provided by the user in OPC UA client. This folder contains all methods and attributes specified in above table.

The node in the address space, under which the "CreateDirectory" method is called, is the "parent" node of the new folder node. Here, the files are available with the extensions .mpf, .spf and .wfp respectively under Part Program, Sub Programs and Work Pieces folders of SINUMERIK. Each of these files will be of "FileType" type and consists of following methods and properties:

Methods/Attributes	Description
Open	Open the file in read/write mode
Read	Read the content of the file
Write	Write data to the file
Close	Close the file (if file opened)
GetPosition	Gets the position of current position of file pointer while file read/write operation.
SetPosition	Sets the position of current position of file pointer while file read/write operation.
OpenCount	Gives the number of file open instances.
Size	Gives the file size details.
Userwritable	Set to true if current user has access to modify the content of the file.
Writable	Set to false if the file is read only.

Whenever the user creates a new file using the method "CreateFile", a new node will be created with "FileType" type with a user provided name. This file again contains all methods and attributes specified in the table above. The node in address space, under which the "CreateFile" method is called, is the "parent" node of the new file node.

In addition to the standard file system, two additional methods are provided to transfer files from server to client and vice versa as shown in Fig-137.

- 1) CopyFileFromServer
- 2) CopyFileToServer

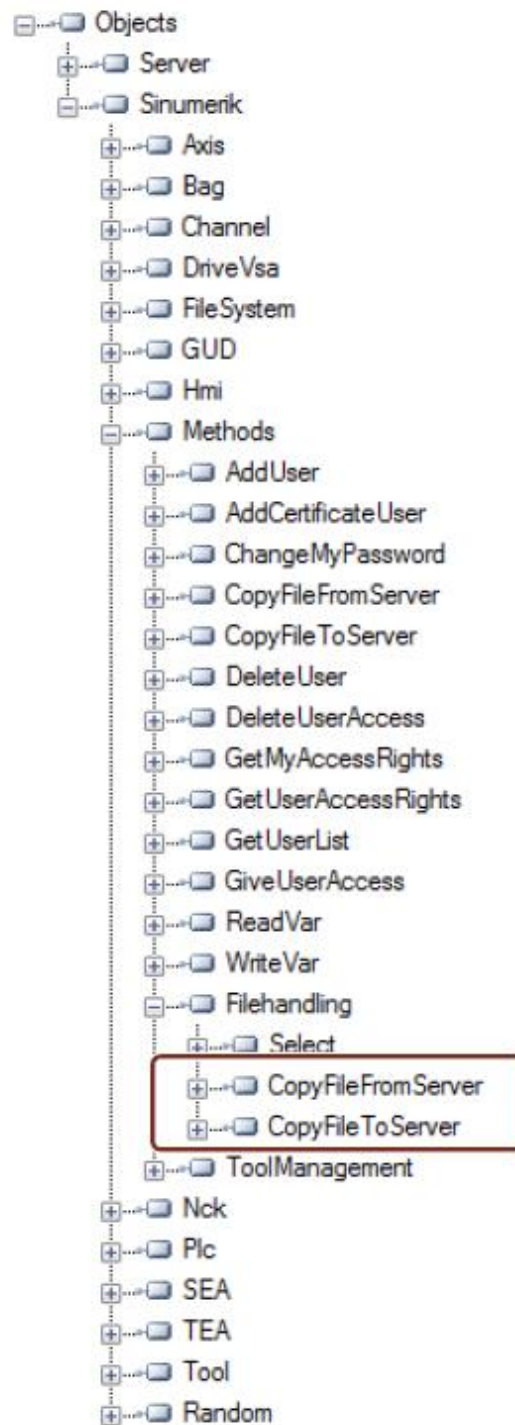


Figure 137 File Transfer Methods to/from Server

1. CopyFileFromServer:

- Allows copying file from SINUMERIK OPC UA server to client location.
- The user shall provide the name of the file with full path to be copied.
- On completion of the file transfer, an appropriate message will be displayed

2. CopyFileToServer:

- Allows copying a client file to a specified SINUMERIK NC memory location.
- The user shall select the file to be transferred and specify the location on server.

Out of security reasons only the following folders are accessible:

- Part Programs
- Sub Programs
- Work Pieces.

For example:

The complete path of the files can be provided as below:

- Sinumerik/FileSystem/Part Program/partprg.mpf
- Sinumerik/FileSystem/Sub Program/subprg.spf
- Sinumerik/FileSystem/Work Pieces/wrkprg.wpf

The maximum subscription load is calculated from the load imposed to the system by the sample rate of all monitored items from all subscriptions of all active sessions.

Max number of monitored items = Systemload / Updates per second

Updates per second = 1 / Sampling rate (in seconds)

Systemload SINUMERIK 840D sl = 1000 items/s

Systemload SINUMERIK 828D = 500 items/s

Almost every variable that is available with the CNC Controller is available in the address space and hence can be accessed by the software through OPC-UA. Because the number of nodes is very large, an exhaustive list of all the variables is not possible to cover in this document. However depending upon the application suitable favourites list can be maintained by the user, so quick selection of nodes can be made while subscribing and monitoring of nodes. For example if you are the part programmer of the machine then you would be primarily interested in 'Sinumerik/Channel/ProgramInfo' node, as this node have all the features regarding the flow and execution of part program. Similarly if one wants to access PLC I/O then the selected node should be 'Sinumerik/Plc'.

No exhaustive list is provided by the Siemens for the description of address space variables; hence the user must have domain knowledge of the CNC Controllers. They may also refer the Siemens List Manual for 840D sl and 828D "NC variables and interface signals" (<https://support.industry.siemens.com/cs/de/de/view/109748365/en>).

7. Case Study with ConnectOPC-UA Server

Software is equipped with in-build OPC-UA server that can be very useful for the self-diagnostics of the software and various features of the OPC-UA protocol can be checked. OPC-UA server can be created by using 'Create OPC-UA Server' option. Any specific resources can be added while creation of the server, this field can be left blank. The screenshot of OPC-UA server creation is shown in Fig-138.

Create OPC-UA Server

Note: This server is for the diagnostics and trianing purpose only.

Enter the Name of the Server :

Create Server

Shutdown Server

Logs

Figure 138 ConnectOPC-UA Server

After successful creation of the server, endpoint of the server is provided, this is useful for the connection of the client with this server. 'opc.tcp://EMX-6047742-01.bhelbpl.co.in:48400' is the endpoint which must be specified while configuring the machine as shown in the Fig-139.

Create OPC-UA Server

Note: This server is for the diagnostics and trianing purpose only.

Enter the Name of the Server :

Create Server

Shutdown Server

Logs

```

Creating OPC-UA Server
Server Initializing
PORT : 4840
RESOURCE PATH :
PRODUCT NAME :
BUILD NUMBER : 1.0
BUILD DATE : 16-07-2019
Adding variables in the Server ...
Constructing Address Space in the Server
Server Created Successfully...
Server is now listening...
PORT : 48400
The Primary Server EndPoint URL is : opc.tcp://EMX-6047742-01.bhelbpl.co.in:48400
  
```

Figure 139 Endpoint of the server

Connect Machine

Machine Name : OPC Machine
 Machine Nubmber : 123
 Machine Location : BLK2BAY1
 OPC Address : opc.tcp://EMX-6047742-01.bhelbpl.co.in:48400
 Username :
 Password :
 Security Policy : None
 Message Sec Mode : None

Disconnect

Logs

Connected with the server : `opc.tcp://EMX-6047742-01.bhelbpl.co.in:48400`
 Session created successfully !

Figure 140 Connection Established with the Server

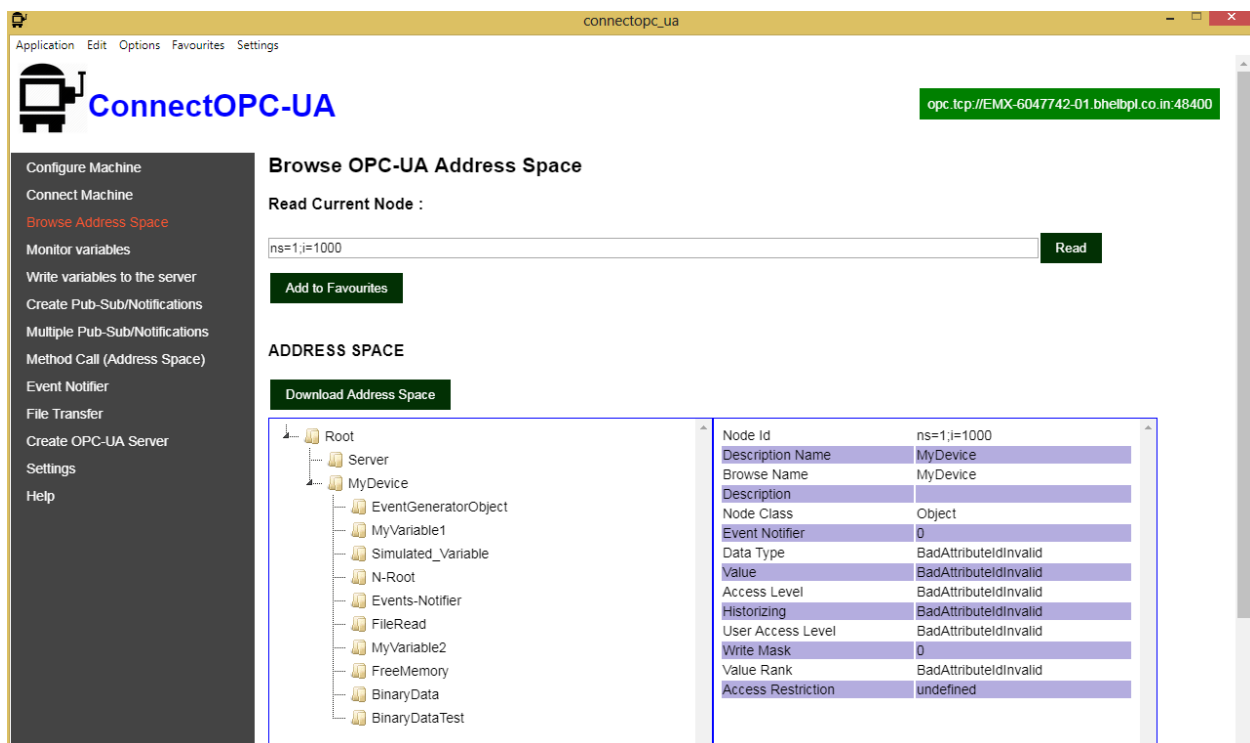


Figure 141 Browsing Nodes

Connect Machine

Browse Address Space

Monitor variables

Write variables to the server

Create Pub-Sub/Notifications

Multiple Pub-Sub/Notifications

Method Call (Address Space)

Event Notifier

File Transfer

Create OPC-UA Server

Settings

Help

Read Current Node :

ns=1;s=File_Read

Read

Add to Favourites

Value of ns=1;s=File_Read is : OPC Unified Architecture (OPC UA) is a machine to machine communication protocol for industrial automation developed by the OPC Foundation. Distinguishing characteristics are:

Focus on communicating with industrial equipment and systems for data collection and control

Open - freely available and implementable under GPL 2.0 license [1]

Cross-platform - not tied to one operating system or programming language

Service-oriented architecture (SOA)

Inherent complexity - the specification consists of 1250 pages in 14 documents

Robust security

Integral information model, which is the foundation of the infrastructure necessary for information integration where vendors and organizations can model their complex data into an OPC UA namespace to take advantage of the rich service-oriented architecture of OPC UA. There are over 35 collaborations with the OPC Foundation currently. Key industries include pharmaceutical, oil and gas, building automation, industrial robotics, security, manufacturing and process control.

Server Time Stamp : Fri Aug 23 2019 14:51:49 GMT+0530 (India Standard Time)

ADDRESS SPACE

Download Address Space

Root

Server

MyDevice

EventGeneratorObject

MyVariable1

Simulated_Variable

N-Root

Node Id	ns=1;s=File_Read
Description Name	FileRead
Browse Name	FileRead
Description	Read File
Node Class	Variable
Event Notifier	BadAttributeIdInvalid
Data Type	ByteString
Value	[object Object]

Figure 142 File Read

8. Source Code

8.1 Main.js

```

const electron = require('electron')
const {electron1} = require('electron')
const {app, BrowserWindow, Menu} = require('electron')
const makeNodeId = require("node-opcua-nodeid").makeNodeId;
const Variant = require("node-opcua-variant").Variant;
const add_eventGeneratorObject = require("node-opcua-address-space").add_eventGeneratorObject;
const assert = require("node-opcua-assert").assert;
const DataValue = require("node-opcua-data-value").DataValue;
const shell = require('electron').shell
const {ipcMain} = require('electron')
var json2xls = require('json2xls');
const { dialog } = require('electron')
const { dialog1 } = require('electron')
var path3 = require('path')
var treeify = require('treeify')
let fs = require('fs');
let fs1 = require('fs');
let fs2 = require('fs');
let opcua = require("node-opcua");
let UAFileType = require('node-opcua')
let async = require("async");
let subscribed_nodes = [];
let clientFile
let mode
let {ClientFile, OpenFileMode, installFileType } = require('node-opcua-file-transfer');

var count_nc = 0;
const os = require('os');
//const storage = require('electron-json-storage');
//storage.setDataPath(os.tmpdir());
//const dataPath = storage.getDataPath();
let
connection_timeout, max_oper, failover_endpoint, server_port, min_sampling, download_folder, sec_con
n, failover_enabled, cs_max_rt, cs_session_alive, cs_endpoint_exist, pub_interval, lifetime_count, keep_al
ive_count, max_not, publishing_enabled, priority, p_nodeid
let opc_url_address
let opc_login_username
let opc_login_password
let server;

let client_connected;
let userIdentity;
let connect_log;
let endpointUrl;

let server_parameters = "";
let server_json_reply;
let subscribed_data = [];
let machine_name, machine_no, machine_location;

```

```

let signal_identifier;
let session_created,client_connect;
let the_session,the_subscription;
let opcua_connected;
let status_data;
let session_id;
let client
let server_status=false;
var VariantArrayType= opcua.VariantArrayType;
var DataType      = opcua.DataType;
var StatusCodes = opcua.StatusCodes;

let win,win1

const fs_status = require('fs')

const path1 = app.getPath('userData')+"/"+ "status_data.txt"

try {
  if (!fs_status.existsSync(path1)) {
    status_data={endpointurl:'endpointUrl','status':'Disconnected'}
    try { fs_status.writeFileSync(app.getPath('userData')+"/"+ "status_data.txt",
JSON.stringify(status_data), 'utf-8'); }
    catch(e) { alert('Failed to save the file!'); }
  }
} catch(err) {
  console.error(err)
}

const fs_setting = require('fs')

const path = app.getPath('userData')+"/"+ "settings_data.txt"

try {
  if (!fs_setting.existsSync(path)){
    connection_timeout=100
    max_oper=100
    failover_endpoint="opc.tcp://192.168.1.7:4840"
    server_port="4840"
    min_sampling="100"
    download_folder=app.getAppPath('userData')
    sec_conn=true
    failover_enabled=true
    cs_max_rt=2
    cs_session_alive=true
    cs_endpoint_exist=false
    pub_interval=1000
    lifetime_count=10
    keep_alive_count=2
    max_not=10
    publishing_enabled=true
    priority=10
    p_nodeid="ns=0;i=85"
    //alert(username);
  }
}

```



```

//var config_data={}
//var key='opc_connect_config_data'
//config_data[key]=[];
var
data={'connection_timeout':connection_timeout,'failover_endpoint':failover_endpoint,'max_oper':max
_oper,'sec_conn':sec_conn,'failover_enabled':failover_enabled,'server_port':server_port,'min_samplin
g':min_sampling,'download_folder':download_folder,'cs_max_rt':cs_max_rt,'cs_session_alive':cs_sessi
on_alive,'cs_endpoint_exist':cs_endpoint_exist,'pub_interval':pub_interval,'lifetime_count':lifetime_co
unt,'keep_alive_count':keep_alive_count,'max_not':max_not,'publishing_enabled':publishing_enabled,'
priority':priority,'p_nodeid':p_nodeid};

    try { fs_setting.writeFileSync(app.getPath('userData')+"/"+'settings_data.txt', JSON.stringify(data),
'utf-8'); }
    catch(e) { alert('Failed to save the file!'); }
  }
} catch(err) {
  console.error(err)
}

/*let client= new opcua.OPCUAClient({
  keepSessionAlive: true,
  endpoint_must_exist: false
});*/

//let client= OPCUAClient(options);

try {
  const fileContents10 = fs2.readFileSync(app.getPath('userData')+"/"+'settings_data.txt', 'utf8')

  const data = JSON.parse(fileContents10)
  //alert(JSON.stringify(data))
  connection_timeout=data.connection_timeout
  max_oper=data.max_oper
  failover_endpoint=data.failover_endpoint
  server_port=data.server_port
  min_sampling=data.min_sampling
  download_folder=data.download_folder
  sec_conn=data.sec_conn
  failover_enabled=data.failover_enabled
  cs_max_rt=data.cs_max_rt
  cs_session_alive=data.cs_session_alive
  cs_endpoint_exist=data.cs_endpoint_exist
  pub_interval=data.pub_interval
  lifetime_count=data.lifetime_count
  keep_alive_count=data.keep_alive_count
  max_not=data.max_not
  publishing_enabled=data.publishing_enabled
  priority=data.priority
  p_nodeid=data.p_nodeid

} catch(err) {

```

```

//console.error(err)
dialog.showErrorBox("Settings not defined","Before using the app please save your settings first")

}

let browse_json=[{ "id" : p_nodeid, "parent" : "#", "text" : "Root" }];

/*var options = {
  connectionStrategy: {
    maxRetry: cs_max_rt,
    keepSessionAlive: cs_session_alive,
    endpoint_must_exist: cs_endpoint_exist
  }
};

client = opcua.OPCUAClient.create(options);*/

function createWindow () {
  // Create the browser window.
  const { width, height } = electron.screen.getPrimaryDisplay().workAreaSize
  win = new BrowserWindow({
    width: width,
    height: height,
    resizable: false,
    icon: path3.join(__dirname, 'linux/64x64.png'),
    webPreferences: {
      nodeIntegration: true
    }
  })
  win.loadFile('connect.html')
  console.log("app started")

  // Emitted when the window is closed.
  win.on('closed', () => {
    // Dereference the window object, usually you would store windows
    // in an array if your app supports multi windows, this is the time
    // when you should delete the corresponding element.
    status_data={endpointurl: endpointUrl, 'status': 'Disconnected'}
    try { fs.writeFileSync(app.getPath('userData')+"/"+"status_data.txt", JSON.stringify(status_data), 'utf-8'); }
    catch(e) {
      //alert('Failed to save the file!');
    }
    win = null
    app.quit()
  })
  createMenu()
}

```

```

}

function createfavouriteWindow () {
  // Create the browser window.
  const { width, height } = electron.screen.getPrimaryDisplay().workAreaSize
  win1 = new BrowserWindow({
    width: width/2,
    height: height/2,
    resizable:false,
    frame:false,
    backgroundColor: '#EFEFEF',
    icon: path3.join(__dirname, 'linux/64x64.png'),
    webPreferences: {
      nodeIntegration: true
    }
  })
  win1.loadFile('favourites.html')
  win1.show()
  //console.log("app started")

  // Emitted when the window is closed.
  win1.on('closed', () => {

    win1 = null
    //app.quit()

  })
  //createMenu_favourite()

}
app.on('ready', createWindow)

// Quit when all windows are closed.
app.on('window-all-closed', () => {
  // On macOS it is common for applications and their menu bar
  // to stay active until the user quits explicitly with Cmd + Q
  if (process.platform !== 'darwin') {
    app.quit()
  }
})

app.on('activate', () => {
  // On macOS it's common to re-create a window in the app when the
  // dock icon is clicked and there are no other windows open.
  if (win === null) {
    createWindow()
  }
})

```

```

function createMenu() {
  const application = {
    label: "Application",
    submenu: [
      {
        label: "About Application",
        selector: "orderFrontStandardAboutPanel:"
      },
      {
        type: "separator"
      },
      {
        label: "Quit",
        accelerator: "Command+Q",
        click: () => {
          app.quit()
        }
      }
    ]
  }
}

```

```

const edit = {
  label: "Edit",
  submenu: [
    {
      label: "Undo",
      accelerator: "CmdOrCtrl+Z",
      selector: "undo:"
    },
    {
      label: "Redo",
      accelerator: "Shift+CmdOrCtrl+Z",
      selector: "redo:"
    },
    {
      type: "separator"
    },
    {
      label: "Cut",
      accelerator: "CmdOrCtrl+X",
      selector: "cut:"
    },
    {
      label: "Copy",
      accelerator: "CmdOrCtrl+C",
      selector: "copy:"
    },
    {
      label: "Paste",
      accelerator: "CmdOrCtrl+V",
      selector: "paste:"
    },
    {
      label: "Select All",
      accelerator: "CmdOrCtrl+A",

```

```

    selector: "selectAll:"
  }
]
}

const options = {
  label: "Options",
  submenu: [
    {
      label: "Configure Machine",
      accelerator: "CmdOrCtrl+M",
      click: () => {
        win.loadFile('index.html')
      }
    },
    {
      label: "Connect Machine",
      accelerator: "CmdOrCtrl+N",
      click: () => {
        win.loadFile('connect.html')
      }
    },
    {
      label: "Browse Address Space",
      accelerator: "CmdOrCtrl+O",
      click: () => {
        win.loadFile('browse.html')
      }
    },
    {
      label: "Monitor Variables",
      accelerator: "CmdOrCtrl+P",
      click: () => {
        win.loadFile('monitor.html')
      }
    },
    {
      label: "Write Variables",
      accelerator: "CmdOrCtrl+Q",
      click: () => {
        win.loadFile('write.html')
      }
    },
    {
      label: "Pub-Sub Notifications",
      accelerator: "CmdOrCtrl+R",
      click: () => {
        win.loadFile('pubsub.html')
      }
    },
    {
      label: "Multiple Pub-Sub Notifications",
      accelerator: "CmdOrCtrl+S",
      click: () => {
        win.loadFile('pubsub_mult.html')
      }
    }
  ]
}

```

```

    }
  },
  {
    label: "Create OPC-UA Server",
    accelerator: "CmdOrCtrl+T",
    click: () => {
      win.loadFile('opcuaserver.html')
    }
  },
  {
    label: "Event Notifier",
    accelerator: "CmdOrCtrl+Z",
    click: () => {
      win.loadFile('alarm_monitoring.html')
    }
  },
  {
    label: "Node Crawler",
    click: () => {
      win.loadFile('nodecrawler.html')
    }
  },
  {
    label: "Methods Call (Address Space)",
    click: () => {
      win.loadFile('method_call.html')
    }
  },
  {
    label: "Filetransfer",
    click: () => {
      win.loadFile('filetransfer.html')
    }
  }
]
}

const setting_menu = {
  label: "Settings",
  submenu: [
    {
      label: "Settings",
      accelerator: "Command+U",
      click: () => {
        win.loadFile('settings.html')
      }
    },
    {
      label: "Status Codes",
      click: () => {
        win.loadFile('opcua_status_codes.html')
      }
    },
    {
      label: "Help",
      accelerator: "Command+W",

```

```

        click:()=> {
            win.loadFile('help.html')
        }
    }

    ]

}

const favourite_menu = {
    label: "Favourites",
    submenu: [
        {
            label: "Clear Favourites",
            click:()=> {
                clear_favourites()
            }
        },
        {
            label: "View Favourites",
            click:()=> {
                win.loadFile('favourites1.html')
            }
        },
    ],
}

]

}

const template = [
    application,
    edit,
    options,
    favourite_menu,
    setting_menu
]

Menu.setApplicationMenu(Menu.buildFromTemplate(template))
}

function createMenu_favourite() {
    const application = {
        label: "Close",
        click:()=> {
            win1.close()
        }
    }
}

```

```

const template = [
  application,

]

Menu.setApplicationMenu(Menu.buildFromTemplate(template))
}

function clear_favourites()
{
  try{
    const fsmr1 = require('fs')
    const mrfileContents1 =
fsmr1.unlinkSync(app.getPath('userData')+"/"+machine_name+"_favourites.txt")
  }
  catch(e)
  {
    dialog.showErrorBox("Error while clearing configuration file", "Error : "+e)
  }
}

ipcMain.on('conf_data', (event, arg) => {
  console.log(arg);
  try { fs1.writeFileSync(app.getPath('userData')+"/"+'opc_conf_data.txt', arg, 'utf-8');
    //dialog.showErrorBox("New Machine succesfully Configured", "Use Connect option to
communicate with the machine")
    event.sender.send('conf_data_reply', "New Machine succesfully Configured\nUse Connect option
to communicate with the machine")
    //win.loadURL('connect.html')
  }
  catch(e) {
    dialog.showErrorBox("Machine configuration failed", "Error : "+e)
    //alert('Failed to save the file!');
  }
});

ipcMain.on('connect_status', (event, arg) => {
  console.log(arg);
  //try { fs.writeFileSync('opc_conf_data.txt', arg, 'utf-8'); }
  //catch(e) { alert('Failed to save the file!'); }
});

ipcMain.on('upload_file_ft', (event, arg) => {
  let options = {
    // See placeholder 1 in above image
    title : "Select File",

    // See placeholder 2 in above image
    defaultPath : "D:\\electron-app",

    // See placeholder 3 in above image

```



```

    buttonLabel : "Open",

    // See placeholder 4 in above image
    filters :[
      {name: 'Text File',extensions: ['.jpg', 'png', 'gif','txt']},
      {name: 'All Files',extensions:['*']}
    ],
    properties: ['openFile']
  }

  dialog.showOpenDialog(win,options,(filePaths)=>{
    console.log(filePaths)
    try
    {
      var fow=require('fs')
      const mrfileContents1 = fow.readFileSync(filePaths.toString(),'utf8')
      event.sender.send('upload_data_content',mrfileContents1)
    }catch(e)
    {
      console.log(e)
    }

  })
  // let filePaths = dialog.showOpenDialog(win,options)
  //console.log(filePaths)
  //try { fs.writeFileSync('opc_conf_data.txt', arg, 'utf-8'); }
  //catch(e) { alert('Failed to save the file!'); }
  });

  ipcMain.on('favourite_data_node', (event, arg) => {
    console.log(arg);
    win.webContents.send('nc_clear_reply', arg);
    // win.loadFile('pubsub_mult.html')
    // event.sender.send('nc_clear_reply',arg)
    win1.close();
  });

  ipcMain.on('setting_app_path_req', (event, arg) => {
    event.sender.send('setting_app_path_rep',app.getPath('userData')+"/")
  });

  ipcMain.on('connect_path_req', (event, arg) => {
    event.sender.send('connect_path_rep',app.getPath('userData')+"/")
  });

  ipcMain.on('browse_path_req', (event, arg) => {
    event.sender.send('browse_path_rep',app.getPath('userData')+"/")
  });

  ipcMain.on('help_path_req', (event, arg) => {
    event.sender.send('help_path_rep',app.getPath('userData')+"/")
  });
  ipcMain.on('monitor_path_req', (event, arg) => {
    event.sender.send('monitor_path_rep',app.getPath('userData')+"/")
  });

```

```

ipcMain.on('pubsub_path_req',(event, arg) => {
    event.sender.send('pubsub_path_rep',app.getPath('userData')+"/")
});

ipcMain.on('setting_data',(event, arg) => {
    console.log(arg);
    app.relaunch();
    app.quit();
    //try { fs.writeFileSync('opc_conf_data.txt', arg, 'utf-8'); }
    //catch(e) { alert('Failed to save the file!'); }
});

ipcMain.on('connect_data',(event, arg) => {

    console.log(arg);
    var data=JSON.parse(arg)
    endpointUrl=data.url_address
    opc_login_username=data.username
    opc_login_password=data.password
    machine_name=data.machine_name
    machine_no=data.machine_no
    machine_location=data.machine_location
    security_policy_string=data.security_policy
    security_mode_string=data.msg_sec_mode
    var security_mode,security_policy
    if(opc_login_username!="" && opc_login_password!="")
    {

        userIdentity = { userName: opc_login_username, password:opc_login_password };
    }
    else{
        userIdentity=null
    }
    if(security_policy_string=="None")
    {
        security_policy=opcua.SecurityPolicy.None
    }
    else if(security_policy_string=="Basic256RSA15")
    {
        security_policy=opcua.SecurityPolicy.None
    }
    else if(security_policy_string=="Basic256")
    {
        security_policy=opcua.SecurityPolicy.None
    }
    if(security_mode_string=="None")
    {
        security_mode=opcua.MessageSecurityMode.None
    }
}

```

```

else if(security_mode_string=="Sign")
{
    security_mode=opcua.MessageSecurityMode.Sign
}
else if(security_mode_string=="SignAndEncrypt")
{
    security_mode=opcua.MessageSecurityMode.SignAndEncrypt
}

var button_status=data.connect_status

var options = {
    connectionStrategy: {
        maxRetry: cs_max_rt,
        keepSessionAlive: cs_session_alive,
        endpoint_must_exist: cs_endpoint_exist,
        securityMode:security_mode,
        securityPolicy:security_policy
    }
};

client= opcua.OPCUAClient.create(options);
//event.sender.send('connect_reply', 'Attempting to connect with the server : '+endpointUrl)

async.series([

// step 1 : connect to
function(callback) {

    if(button_status=="Connect")
    {
        client.connect(endpointUrl,function (err) {
            //const {ipcRenderer}= require('electron')
            if(err){
                console.log(" cannot connect to endpoint : " , endpointUrl );
                event.sender.send('connect_reply', 'Cannot Connect with the server :
'+endpointUrl+"\n"+"Error : "+err);
                client_connected=false;
                status_data={'endpointurl':endpointUrl,'status':'Disconnected'}
                event.sender.send('status_reply', JSON.stringify(status_data));
                client_connect=false;
                try { fs.writeFileSync(app.getPath('userData')+"/"+'+status_data.txt', JSON.stringify(status_data),
'utf-8'); }
                catch(e) { alert('Failed to save the file!'); }
                //win.webContents.send('asynchronous-message', {'SAVED': 'File Saved'});

            } else {
                console.log("connected !");
                event.sender.send('connect_reply', 'Connected with the server : '+endpointUrl);
                client_connected=true;
                client_connect=true;
            }
        });
    }
}

```

```

        status_data={'endpointurl':endpointUrl,'status':'Connected'}
        event.sender.send('status_reply', JSON.stringify(status_data));
        try { fs.writeFileSync(app.getPath('userData')+"/"+'+status_data.txt',
JSON.stringify(status_data), 'utf-8'); }
        catch(e) { alert('Failed to save the file!'); }

        event.sender.send('client_connected', client_connected);
        browse_json=[{"id": "ns=0;i=85", "parent": "#", "text": "Root" }];
        //win.webContents.send('asynchronous-message', {'SAVED': 'File Saved'});
    }
    callback(err);
});

client.on("timed_out_request ", function () {
    console.log("timed_out_request ");
    event.sender.send('connect_reply', "timed_out_request ");
    status_data={'endpointurl':endpointUrl,'status':'Disconnected'}
    try { fs.writeFileSync(app.getPath('userData')+"/"+'+status_data.txt', JSON.stringify(status_data),
'utf-8'); }
    catch(e) { alert('Failed to save the file!'); }
    event.sender.send('status_reply', JSON.stringify(status_data));
});

client.on("start_reconnection", function () {
    console.log("start_reconnection not working so aborting");
    event.sender.send('connect_reply', "start_reconnection not working so aborting");
    status_data={'endpointurl':endpointUrl,'status':'Disconnected'}
    try { fs.writeFileSync(app.getPath('userData')+"/"+'+status_data.txt', JSON.stringify(status_data),
'utf-8'); }
    catch(e) { alert('Failed to save the file!'); }
    event.sender.send('status_reply', JSON.stringify(status_data));
});

client.on("connection_reestablished", function () {
    console.log("connection_reestablished ");
    event.sender.send('connect_reply', "connection_reestablished ");
    status_data={'endpointurl':endpointUrl,'status':'Connected'}
    try { fs.writeFileSync(app.getPath('userData')+"/"+'+status_data.txt', JSON.stringify(status_data),
'utf-8'); }
    catch(e) { alert('Failed to save the file!'); }
    event.sender.send('status_reply', JSON.stringify(status_data));
});

client.on("close", function () {
    console.log("close and abort");
    event.sender.send('connect_reply', "close and abort");
    status_data={'endpointurl':endpointUrl,'status':'Disconnected'}
    try { fs.writeFileSync(app.getPath('userData')+"/"+'+status_data.txt', JSON.stringify(status_data),
'utf-8'); }
    catch(e) { alert('Failed to save the file!'); }
    event.sender.send('status_reply', JSON.stringify(status_data));
    //opcuaServerUp = false;
});

client.on("backoff", function (nb, delay) {
    event.sender.send('connect_reply', "connection failed for the "+ nb+" time ... We will retry in
"+delay+ " ms");
    console.log(" connection failed for the", nb," time ... We will retry in ", delay, " ms");
    status_data={'endpointurl':endpointUrl,'status':'Disconnected'}

```

```

        try { fs.writeFileSync(app.getPath('userData')+"/"+'+status_data.txt', JSON.stringify(status_data),
'utf-8'); }
        catch(e) { alert('Failed to save the file!'); }
        event.sender.send('status_reply', JSON.stringify(status_data));
        //opcuaServerUp = false;
    });
}
else
{
    client.disconnect()
    status_data={'endpointurl':endpointUrl,'status':'Disconnected'}
    try { fs.writeFileSync(app.getPath('userData')+"/"+'+status_data.txt', JSON.stringify(status_data), 'utf-
8'); }
    catch(e) { alert('Failed to save the file!'); }
    event.sender.send('status_reply', JSON.stringify(status_data));
}

},

function(callback){

    client.createSession(userIdentity,function(err,session){
        if(!err) {
            the_session= session;
            console.log("session created")
            event.sender.send('connect_reply', 'Session created successfully!');
            session_id=session.sessionId.toString();
            session_created=true
        }
        else
        {
            event.sender.send('connect_reply', 'Session creation failed!');
            console.log("session creation failed")
            session_created=false
        }
        callback(err);
    });
},

],
function(err){
    if(err){
        console.log("failure",err);
        client.disconnect(function({});
    } else {
        console.log("done!")
    }
    //client.disconnect(function({});
})
}

```

```

    })

    ipcMain.on('monitor_data_value_node', (event, arg) => {

        async.series([

            function(callback) {
                var maxAge = 0;
                var nodeToRead = { nodeId: arg, attributeld: opcua.Attributelds.Value };
                the_session.read(nodeToRead, maxAge, function(err, dataValue) {
                    if (!err) {
                        console.log(arg, dataValue.toString());
                        //read_value=dataValue.toString().split(":");
                        //console.log(read_value)

                        //event.sender.send('monitor1_reply', arg+" : " +dataValue.toString()+"\n");
                        //event.sender.send('monitor1_reply', "VariableValue"+" : " +read_value[3].split("")[0]+" \n");
                        event.sender.send('browse_node_reply', "\n\nValue of "+arg+" is : "
+dataValue.value.value+"\n"+"Server Time Stamp : "+dataValue.serverTimestamp+"\n\n");

                    }
                    callback(err);
                });
            },

            function(err) {
                if (err) {
                    console.log(" failure ", err);
                    dialog.showErrorBox("Reading failed", "Error : "+err)
                    //client.disconnect(function(){});
                } else {
                    console.log("done!")
                }
                //client.disconnect(function(){});
            }
        ]);

        ipcMain.on('monitor_data', (event, arg) => {

            async.series([

                function(callback) {
                    var maxAge = 0;
                    var nodeToRead = { nodeId: arg, attributeld: opcua.Attributelds.Value };
                    the_session.read(nodeToRead, maxAge, function(err, dataValue) {

```

```

        if(!err) {
            console.log(arg,dataValue.toString());
            read_value=dataValue.toString().split(":");
            console.log(read_value)

            //event.sender.send('monitor1_reply', arg+" : "+dataValue.toString()+"\n");
            //event.sender.send('monitor1_reply', "VariableValue"+" : "+read_value[3].split("")[0]+" \n");
            event.sender.send('monitor1_reply', "Value of "+arg+" is : "
+dataValue.value.value+"\n"+"Server Time Stamp : "+dataValue.serverTimestamp);

        }
        callback(err);
    });

},

],
function(err){
    if(err) {
        console.log(" failure",err);
        //client.disconnect(function(){});
        dialog.showErrorBox("Error", "Error : "+err)
    } else {
        console.log("done!")
    }
    // client.disconnect(function(){});
})

});

ipcMain.on('browse_data', (event, arg) => {

    async.series([

        function(callback){
            try{

                the_session.browse(arg, function(err, browseResult) {
                    if(!err) {
                        console.log('Node Id recieved is '+arg)
                        event.sender.send('browse_clear_reply', arg);
                        console.log(arg)
                        console.log(browseResult.toString());
                        browseResult.references.forEach( function(reference) {
                            //if(!err)
                            // {
                                console.log( reference.browseName.toString());

```

```

        //event.sender.send('browse_reply', reference.browseName.toString()+" [ Node id :
        "+reference.nodeId.toString()+" ]");

        //var data1=JSON.stringify(browse_json)
        var match=false;
        for(var i=0; i<browse_json.length; i++){
            if(browse_json[i].text == reference.browseName.toString()){
                match=true
                // console.log("match detected",browse_json[i].id )
                break;
            }
            else
            {
                //console.log("nor equal : "+browse_json[i].id+reference.nodeId.toString())
            }
        }
        if(!match)
        {
            var
            data={"id":reference.nodeId.toString(),"parent":arg,"text":reference.browseName.toString()+ " [
            "+reference.nodeId.toString()+" ]"};
            browse_json.push(data);
            //console.log("adding",data)
        }
        else
        {
            console.log("data not added to json String")
        }
        //browse_json.push(data);
        event.sender.send('browse_json_reply', browse_json);
        //event.sender.send('browse_reply', browse_json);
        // }
        // else{
        //     console.log('Error in Browsing'+err);
        //     event.sender.send('connect_reply', 'Error in Browsing'+err);
        // }

    });

}

    callback(err);
});
}catch(e)
{
    dialog.showErrorBox("BrowseError","Check server connection : "+e)
}
},

```



```

    ],
    function(err){
        if(err) {
            console.log(" failure ",err);
            //client.disconnect(function({});
            dialog.showErrorBox("Error", "Error : "+err)
        } else {
            console.log("done!")
        }
        //client.disconnect(function({});
    })

});

ipcMain.on('node_crawler_data', (event, arg) => {
    var continuation_points=null;
    event.sender.send('nc_clear_reply', arg);
    // "ns=2;s=/Channel/State"

    async.series([

        function(callback){
            var maxAge = 0;
            var nodeToRead = [{ nodeId: arg, attributeId: opcua.AttributeIds.Value },
                { nodeId: arg, attributeId: opcua.AttributeIds.AccessLevel },
                { nodeId: arg, attributeId: opcua.AttributeIds.Description },
                { nodeId: arg, attributeId: opcua.AttributeIds.DataType },
                { nodeId: arg, attributeId: opcua.AttributeIds.DisplayName },
                { nodeId: arg, attributeId: opcua.AttributeIds.BrowseName },
                { nodeId: arg, attributeId: opcua.AttributeIds.EventNotifier },
                { nodeId: arg, attributeId: opcua.AttributeIds.NodeId },
                { nodeId: arg, attributeId: opcua.AttributeIds.Historizing },
                { nodeId: arg, attributeId: opcua.AttributeIds.NodeClass },
                { nodeId: arg, attributeId: opcua.AttributeIds.UserAccessLevel },
                { nodeId: arg, attributeId: opcua.AttributeIds.WriteMask },
                { nodeId: arg, attributeId: opcua.AttributeIds.ValueRank },
                { nodeId: arg, attributeId: opcua.AttributeIds.AccessRestrictions },

            ];

            the_session.read(nodeToRead, maxAge, function(err,dataValue){
                if(!err) {
                    count=1
                    var value_node,access_level,desc,dt,dn,bn,en,nd,his,nc,ual,wm,vr,ar
                    //console.log(arg, dataValue.toString());
                    dataValue.forEach(function(datavalue1){
                        if(count==1)
                        {
                            if(datavalue1.statusCode==opcua.StatusCodes.Good)
                            {

```

```

        value_node=datavalue1.value.value
    }
    else
    {
        value_node= datavalue1.statusCode._name
    }
}

else if(count==2)
{
    if(datavalue1.statusCode==opcua.StatusCodes.Good)
    {

        access_level=datavalue1.value.value
    }
    else
    {
        access_level= datavalue1.statusCode._name
    }
}
else if(count==3)
{
    if(datavalue1.statusCode==opcua.StatusCodes.Good)
    {

        desc=datavalue1.value.value.text
    }
    else
    {
        desc= datavalue1.statusCode._name
    }
}

else if(count==4)
{
    if(datavalue1.statusCode==opcua.StatusCodes.Good)
    {
        if(datavalue1.value.value.toString()=="ns=0;i=0")
        {
            dt="Null"
        }
        else if(datavalue1.value.value.toString()=="ns=0;i=1")
        {
            dt="Boolean"
        }
        else if(datavalue1.value.value.toString()=="ns=0;i=2")
        {
            dt="Signed Byte"
        }
        else if(datavalue1.value.value.toString()=="ns=0;i=3")
        {
            dt="Unsigned Byte"
        }
        else if(datavalue1.value.value.toString()=="ns=0;i=4")
        {

```

```

        dt="Int16"
    }
    else if(datavalue1.value.value.toString()=="ns=0;i=5")
    {
        dt="UInt16"
    }
    else if(datavalue1.value.value.toString()=="ns=0;i=6")
    {
        dt="Int32"
    }
    else if(datavalue1.value.value.toString()=="ns=0;i=7")
    {
        dt="UInt32"
    }
    else if(datavalue1.value.value.toString()=="ns=0;i=8")
    {
        dt="Int64"
    }
    else if(datavalue1.value.value.toString()=="ns=0;i=9")
    {
        dt="UInt64"
    }
    else if(datavalue1.value.value.toString()=="ns=0;i=10")
    {
        dt="Float"
    }
    else if(datavalue1.value.value.toString()=="ns=0;i=11")
    {
        dt="Double"
    }
    else if(datavalue1.value.value.toString()=="ns=0;i=12")
    {
        dt="String"
    }
    else if(datavalue1.value.value.toString()=="ns=0;i=13")
    {
        dt="DateTime"
    }
    else if(datavalue1.value.value.toString()=="ns=0;i=14")
    {
        dt="Guid"
    }
    else if(datavalue1.value.value.toString()=="ns=0;i=15")
    {
        dt="ByteString"
    }
    else if(datavalue1.value.value.toString()=="ns=0;i=16")
    {
        dt="XmlElement"
    }
    else if(datavalue1.value.value.toString()=="ns=0;i=17")
    {
        dt="NodeId"
    }
    else if(datavalue1.value.value.toString()=="ns=0;i=18")
    {

```

```

        dt="ExpandedNodeId"
    }
    else if(datavalue1.value.value.toString()=="ns=0;i=19")
    {
        dt="StatusCode"
    }
    else if(datavalue1.value.value.toString()=="ns=0;i=20")
    {
        dt="QualifiedName"
    }
    else if(datavalue1.value.value.toString()=="ns=0;i=21")
    {
        dt="LocalizedText"
    }
    else if(datavalue1.value.value.toString()=="ns=0;i=22")
    {
        dt="ExtentionObject"
    }
    else if(datavalue1.value.value.toString()=="ns=0;i=23")
    {
        dt="DataValue"
    }
    else if(datavalue1.value.value.toString()=="ns=0;i=24")
    {
        dt="Variant"
    }
    else if(datavalue1.value.value.toString()=="ns=0;i=25")
    {
        dt="DiagnosticInfo"
    }
    else{
        dt=datavalue1.value.value
    }

}
else
{
    dt= datavalue1.statusCode._name
}
}
else if(count==5)
{
    if(datavalue1.statusCode==opcua.StatusCodes.Good)
    {

        dn=datavalue1.value.value.text
    }
    else
    {
        dn= datavalue1.statusCode._name
    }
}
}
else if(count==6)
{

```

```

    if(datavalue1.statusCode==opcua.StatusCodes.Good)
    {
        bn=datavalue1.value.value.name
    }
    else
    {
        bn= datavalue1.statusCode._name
    }
}
else if(count==7)
{
    if(datavalue1.statusCode==opcua.StatusCodes.Good)
    {
        en=datavalue1.value.value
    }
    else
    {
        en= datavalue1.statusCode._name
    }
}
else if(count==8)
{
    if(datavalue1.statusCode==opcua.StatusCodes.Good)
    {
        nd=datavalue1.value.value
    }
    else
    {
        nd= datavalue1.statusCode._name
    }
}
else if(count==9)
{
    if(datavalue1.statusCode==opcua.StatusCodes.Good)
    {
        his=datavalue1.value.value
    }
    else
    {
        his= datavalue1.statusCode._name
    }
}
else if(count==10)
{
    if(datavalue1.statusCode==opcua.StatusCodes.Good)
    {
        nc=datavalue1.value.value
        if(datavalue1.value.value==64)
        {
            nc="Data Type"
        }
    }
}

```

```

        else if(datavalue1.value.value==4294967295)
        {
            nc="Invalid"
        }
        else if(datavalue1.value.value==4)
        {
            nc="Method"
        }
        else if(datavalue1.value.value==1)
        {
            nc="Object"
        }
        else if(datavalue1.value.value==8)
        {
            nc="Object Type"
        }
        else if(datavalue1.value.value==32)
        {
            nc="Reference Type"
        }
        else if(datavalue1.value.value==0)
        {
            nc="Unspecified"
        }
        else if(datavalue1.value.value==2)
        {
            nc="Variable"
        }
        else if(datavalue1.value.value==16)
        {
            nc="Variable Type"
        }
        else if(datavalue1.value.value==128)
        {
            nc="View"
        }
    }
    else
    {
        nc= datavalue1.statusCode._name
    }
}

else if(count==11)
{
    if(datavalue1.statusCode==opcua.StatusCodes.Good)
    {
        ual=datavalue1.value.value
    }
    else
    {
        ual= datavalue1.statusCode._name
    }
}
}

```

```

        else if(count==12)
        {
            if(datavalue1.statusCode==opcua.StatusCodes.Good)
            {

                wm=datavalue1.value.value
            }
            else
            {
                wm= datavalue1.statusCode._name
            }
        }
        else if(count==13)
        {
            if(datavalue1.statusCode==opcua.StatusCodes.Good)
            {

                vr=datavalue1.value.value
            }
            else
            {
                vr= datavalue1.statusCode._name
            }
        }

        else if(count==10)
        {
            if(datavalue1.statusCode==opcua.StatusCodes.Good)
            {

                ar=datavalue1.value.value
            }
            else
            {
                ar= datavalue1.statusCode._name
            }
        }

        count++
    })

    value_data={"v":value_node,"al":access_level,"desc":desc,"dt":dt,"dn":dn,"bn":bn,"en":en,"nd":nd,"his":his,"nc":nc,"ual":ual,"wm":wm,"vr":vr,"ar":ar}
    // console.log(value_data)
    event.sender.send('node_attributes',JSON.stringify(value_data))
    //read_value=dataValue.toString().split(":");
    // console.log(read_value)

    //event.sender.send('monitor1_reply', arg+" : " +dataValue.toString()+"\n");
    //event.sender.send('monitor1_reply', "Variable Value"+" : " +read_value[3].split(" ")[0]+" \n");
    //event.sender.send('monitor1_reply', "Value of "+arg+" is : "+dataValue.value.value+"\n"+"Server Time Stamp : "+dataValue.serverTimestamp);

}
callback(err);

```

```

});

},

function(callback){

    const browseDesc = {
        nodeId: arg,
        referenceTypeId:null,
        nodeClassMask:0,
        browseDirection: opcua.BrowseDirection.Forward,
        resultMask:63
    };

    const browseRequest1 = new opcua.BrowseRequest({
        view: null,/{ viewId: 'ns=0;i=85'},
        requestedMaxReferencesPerNode: 45,
        nodesToBrowse: [browseDesc]
    });
    the_session.performMessageTransaction(browseRequest1, function (err, response) {
        if(!err) {

            console.log(response.toString());

            response.results[0].references.forEach(function(reference) {
                //if(!err)
                //{
                // console.log( reference.browseName.toString());

                // if(reference.referenceTypeId != "ns=0;i=40" && reference.referenceTypeId != "ns=0;i=45" &&
                reference.referenceTypeId != "ns=0;i=63" && reference.referenceTypeId != "ns=0;i=61")
                    if(reference.referenceTypeId != "ns=0;i=40")
                    {

                        var match=false;
                        for(var i=0; i<browse_json.length; i++){
                            if(browse_json[i].id == reference.nodeId.toString()){
                                match=true
                                // console.log("match detected",browse_json[i].id )
                                break;
                            }
                        }

                        if(!match )
                        {
                            var
                            data={"id":reference.nodeId.toString(),"parent":arg,"text":reference.displayName.text.toString()};
                            browse_json.push(data);

```



```

    }
    else
    {
        console.log("data not added to json String")
    }

    event.sender.send('nc_json_reply', browse_json);
    count_nc++
    console.log("count",count_nc)
}
})

if(response.results[0].continuationPoint!=null)
{
    continuation_points=response.results[0].continuationPoint
}
else
{
    continuation_points=null
}
callback();
}
});
},

function (callback){
    if(continuation_points!=null)
    {
        const browseNextRequest = new opcua.BrowseNextRequest({
            continuationPoints:[continuation_points]
        });
        the_session.performMessageTransaction(browseNextRequest, function (err, response) {
            console.log(response.toString());
            response.results[0].references.forEach( function(reference) {
                //if(!err)
                // {
                //console.log( reference.browseName.toString());

                //event.sender.send('browse_reply', reference.browseName.toString()+" [ Node id :
                "+reference.nodeId.toString()+" ]");

                //var data1=JSON.stringify(browse_json)
                if(reference.referenceTypeId != "ns=0;i=40")
                {
                    var match=false;
                    for(var i=0; i<browse_json.length; i++){
                        if(browse_json[i].id == reference.nodeId.toString()){
                            match=true
                            // console.log("match detected",browse_json[i].id )
                            break;

```

```

    }
    else
    {
        //console.log("nor equal : "+browse_json[i].id+reference.nodeId.toString())
    }
}
if(!match)
{
    var
data={"id":reference.nodeId.toString(),"parent":arg,"text":reference.displayName.text.toString()};
    browse_json.push(data);
    //console.log("adding",data)
}
else
{
    console.log("data not added to json String")
}
//browse_json.push(data);
event.sender.send('nc_json_reply', browse_json);
}
})
if(response.results[0].continuationPoint!=null)
{
    continuation_points=response.results[0].continuationPoint
}
else
{
    continuation_points=null
}
// response.responseHeader.serviceResult.should.equal(StatusCodes.BadNothingToDo);
callback();
});
},
function (callback){
    if(continuation_points!=null)
    {
        const browseNextRequest = new opcua.BrowseNextRequest({
            continuationPoints:[continuation_points]
        });
        the_session.performMessageTransaction(browseNextRequest, function (err, response) {
            console.log(response.toString());
            response.results[0].references.forEach( function(reference) {
                //if(!err)
                //{
                console.log( reference.browseName.toString());

                //event.sender.send('browse_reply', reference.browseName.toString()+" [ Node id :
"+reference.nodeId.toString()+" ]");

                //var data1=JSON.stringify(browse_json)
                if(reference.referenceTypeId != "ns=0;i=40")
                {
                    var match=false;

```

```

        for(var i=0; i<browse_json.length; i++){
            if(browse_json[i].id == reference.nodeId.toString()){
                match=true
                // console.log("match detected",browse_json[i].id )
                break;
            }
            else
            {
                //console.log("nor equal : "+browse_json[i].id+reference.nodeId.toString())
            }
        }
        if(!match)
        {
            var
data={"id":reference.nodeId.toString(),"parent":arg,"text":reference.displayName.text.toString()};
            browse_json.push(data);
            //console.log("adding",data)
        }
        else
        {
            console.log("data not added to json String")
        }
        //browse_json.push(data);
        event.sender.send('nc_json_reply', browse_json);
    }
    })
    if(response.results[0].continuationPoint!=null)
    {
        continuation_points=response.results[0].continuationPoint
    }
    else
    {
        continuation_points=null
    }
    // response.responseHeader.serviceResult.should.equal(StatusCodes.BadNothingToDo);
    callback();
    });
    },

function (callback){
    if(continuation_points!=null)
    {
        const browseNextRequest = new opcua.BrowseNextRequest({
            continuationPoints:[continuation_points]
        });
        the_session.performMessageTransaction(browseNextRequest, function (err, response) {
            console.log(response.toString());
            response.results[0].references.forEach( function(reference) {
                //if(!err)
                //{
                    console.log( reference.browseName.toString());

                    //event.sender.send('browse_reply', reference.browseName.toString()+" [ Node id :
"+reference.nodeId.toString()+" ]");

```

```

        //var data1=JSON.stringify(browse_json)
        if(reference.referenceTypeId != "ns=0;i=40")
        {
            var match=false;
            for(var i=0; i<browse_json.length; i++){
                if(browse_json[i].id == reference.nodeId.toString()){
                    match=true
                    // console.log("match detected",browse_json[i].id )
                    break;
                }
                else
                {
                    //console.log("nor equal : "+browse_json[i].id+reference.nodeId.toString())
                }
            }
            if(!match)
            {
                var
                data={"id":reference.nodeId.toString(),"parent":arg,"text":reference.displayName.text.toString()};
                browse_json.push(data);
                //console.log("adding",data)
            }
            else
            {
                console.log("data not added to json String")
            }
            //browse_json.push(data);
            event.sender.send('nc_json_reply', browse_json);
        }
    })
    if(response.results[0].continuationPoint!=null)
    {
        continuation_points=response.results[0].continuationPoint
    }
    else
    {
        continuation_points=null
    }
    // response.responseHeader.serviceResult.should.equal(StatusCodes.BadNothingToDo);
    callback();
    });
}
},

function (callback){
    if(continuation_points!=null)
    {
        const browseNextRequest = new opcua.BrowseNextRequest({
            continuationPoints: [continuation_points]
        });
    };
    the_session.performMessageTransaction(browseNextRequest, function (err, response) {
        console.log(response.toString());
        response.results[0].references.forEach( function(reference) {
            //if(!err)
            // {

```

```

        console.log( reference.browseName.toString());

        //event.sender.send('browse_reply', reference.browseName.toString()+" [ Node id :
"+reference.nodeId.toString()+" ]");

        //var data1=JSON.stringify(browse_json)
        if(reference.referenceTypeId != "ns=0;i=40")
        {
            var match=false;
            for(var i=0; i<browse_json.length; i++){
                if(browse_json[i].id == reference.nodeId.toString()){
                    match=true
                    // console.log("match detected",browse_json[i].id )
                    break;
                }
                else
                {
                    //console.log("nor equal : "+browse_json[i].id+reference.nodeId.toString())
                }
            }
            if(!match)
            {
                var
data={"id":reference.nodeId.toString(),"parent":arg,"text":reference.displayName.text.toString()};
                browse_json.push(data);
                //console.log("adding",data)
            }
            else
            {
                console.log("data not added to json String")
            }
            //browse_json.push(data);
            event.sender.send('nc_json_reply', browse_json);
        }
    })
    if(response.results[0].continuationPoint!=null)
    {
        continuation_points=response.results[0].continuationPoint
    }
    else
    {
        continuation_points=null
    }
    // response.responseHeader.serviceResult.shouldEqual(StatusCodes.BadNothingToDo);
    callback();
});
}
},

function (callback) {
    if(continuation_points!=null)
    {
        const browseNextRequest = new opcua.BrowseNextRequest({
            continuationPoints: [continuation_points]
        });
    }
}

```

```

the_session.performMessageTransaction(browseNextRequest, function (err, response) {
    console.log(response.toString());
    response.results[0].references.forEach( function(reference) {
        //if(!err)
        //{
            console.log( reference.browseName.toString());

            //event.sender.send('browse_reply', reference.browseName.toString()+" [ Node id :
            "+reference.nodeId.toString()+" ]");

            //var data1=JSON.stringify(browse_json)
            if(reference.referenceTypeId != "ns=0;i=40")
            {
                var match=false;
                for(var i=0; i<browse_json.length; i++){
                    if(browse_json[i].id == reference.nodeId.toString()){
                        match=true
                        // console.log("match detected",browse_json[i].id )
                        break;
                    }
                    else
                    {
                        //console.log("nor equal : "+browse_json[i].id+reference.nodeId.toString())
                    }
                }
                if(!match)
                {
                    var
                    data={"id":reference.nodeId.toString(),"parent":arg,"text":reference.displayName.text.toString()};
                    browse_json.push(data);
                    //console.log("adding",data)
                }
                else
                {
                    console.log("data not added to json String")
                }
                //browse_json.push(data);
                event.sender.send('nc_json_reply', browse_json);
            }
        })
        if(response.results[0].continuationPoint!=null)
        {
            continuation_points=response.results[0].continuationPoint
        }
        else
        {
            continuation_points=null
        }
        // response.responseHeader.serviceResult.should.equal(StatusCodes.BadNothingToDo);
        callback();
    });
}
},

function (callback){

```

```

if(continuation_points!=null)
{
const browseNextRequest = new opcua.BrowseNextRequest({
continuationPoints:[continuation_points]
});
the_session.performMessageTransaction(browseNextRequest, function (err, response) {
console.log(response.toString());
response.results[0].references.forEach( function(reference) {
//if(!err)
//{
console.log( reference.browseName.toString());

//event.sender.send('browse_reply', reference.browseName.toString()+" [ Node id :
"+reference.nodeId.toString()+" ]");

//var data1=JSON.stringify(browse_json)
if(reference.referenceTypeId != "ns=0;i=40")
{
var match=false;
for(var i=0; i<browse_json.length; i++){
if(browse_json[i].id == reference.nodeId.toString()){
match=true
// console.log("match detected",browse_json[i].id )
break;
}
else
{
//console.log("nor equal : "+browse_json[i].id+reference.nodeId.toString())
}
}
if(!match)
{
var
data={"id":reference.nodeId.toString(),"parent":arg,"text":reference.displayName.text.toString()};
browse_json.push(data);
//console.log("adding",data)
}
else
{
console.log("data not added to json String")
}
//browse_json.push(data);
event.sender.send('nc_json_reply', browse_json);
}
})
if(response.results[0].continuationPoint!=null)
{
continuation_points=response.results[0].continuationPoint
}
else
{
continuation_points=null
}
// response.responseHeader.serviceResult.should.equal(StatusCodes.BadNothingToDo);
callback();

```

```

});
}
},

function (callback) {
    if(continuation_points!=null)
    {
        const browseNextRequest = new opcua.BrowseNextRequest({
            continuationPoints:[continuation_points]
        });
        the_session.performMessageTransaction(browseNextRequest, function (err, response) {
            console.log(response.toString());
            response.results[0].references.forEach( function(reference) {
                //if(!err)
                // {
                console.log( reference.browseName.toString());

                //event.sender.send('browse_reply', reference.browseName.toString()+" [ Node id :
                "+reference.nodeId.toString()+" ]");

                //var data1=JSON.stringify(browse_json)
                if(reference.referenceTypeId != "ns=0;i=40")
                {
                    var match=false;
                    for(var i=0; i<browse_json.length; i++){
                        if(browse_json[i].id == reference.nodeId.toString()){
                            match=true
                            // console.log("match detected",browse_json[i].id )
                            break;
                        }
                        else
                        {
                            //console.log("not equal : "+browse_json[i].id+reference.nodeId.toString())
                        }
                    }
                    if(!match)
                    {
                        var
                        data={"id":reference.nodeId.toString(),"parent":arg,"text":reference.displayName.text.toString()};
                        browse_json.push(data);
                        //console.log("adding",data)
                    }
                    else
                    {
                        console.log("data not added to json String")
                    }
                    //browse_json.push(data);
                    event.sender.send('nc_json_reply', browse_json);
                }
            })
            if(response.results[0].continuationPoint!=null)
            {
                continuation_points=response.results[0].continuationPoint
            }
            else

```



```

    {
        continuation_points=null
    }
    // response.responseHeader.serviceResult.should.equal(StatusCodes.BadNothingToDo);
    callback();
    });
}
},

function (callback) {
    if(continuation_points!=null)
    {
        const browseNextRequest = new opcua.BrowseNextRequest({
            continuationPoints:[continuation_points]
        });
        the_session.performMessageTransaction(browseNextRequest, function (err, response) {
            console.log(response.toString());
            response.results[0].references.forEach( function(reference) {
                //if(!err)
                // {
                console.log( reference.browseName.toString());

                //event.sender.send('browse_reply', reference.browseName.toString()+" [ Node id :
                "+reference.nodeId.toString()+" ]");

                //var data1=JSON.stringify(browse_json)
                if(reference.referenceTypeId != "ns=0;i=40")
                {
                    var match=false;
                    for(var i=0; i<browse_json.length; i++){
                        if(browse_json[i].id == reference.nodeId.toString()){
                            match=true
                            // console.log("match detected",browse_json[i].id )
                            break;
                        }
                    }
                    else
                    {
                        //console.log("nor equal : "+browse_json[i].id+reference.nodeId.toString())
                    }
                }
                if(!match)
                {
                    var
                    data={"id":reference.nodeId.toString(),"parent":arg,"text":reference.displayName.text.toString()};
                    browse_json.push(data);
                    //console.log("adding",data)
                }
                else
                {
                    console.log("data not added to json String")
                }
                //browse_json.push(data);
                event.sender.send('nc_json_reply', browse_json);
            }
        })
    }
}

```

```

    if(response.results[0].continuationPoint!=null)
    {
        continuation_points=response.results[0].continuationPoint
    }
    else
    {
        continuation_points=null
    }
    // response.responseHeader.serviceResult.should.equal(StatusCodes.BadNothingToDo);
    callback();
});
}
},

function (callback) {
    if(continuation_points!=null)
    {
        const browseNextRequest = new opcua.BrowseNextRequest({
            continuationPoints: [continuation_points]
        });
        the_session.performMessageTransaction(browseNextRequest, function (err, response) {
            console.log(response.toString());
            response.results[0].references.forEach( function(reference) {
                //if(!err)
                // {
                console.log( reference.browseName.toString());

                //event.sender.send('browse_reply', reference.browseName.toString()+" [ Node id :
                "+reference.nodeId.toString()+" ]");

                //var data1=JSON.stringify(browse_json)
                if(reference.referenceTypeId != "ns=0;i=40")
                {
                    var match=false;
                    for(var i=0; i<browse_json.length; i++){
                        if(browse_json[i].id == reference.nodeId.toString()){
                            match=true
                            // console.log("match detected",browse_json[i].id )
                            break;
                        }
                    }
                    else
                    {
                        //console.log("nor equal : "+browse_json[i].id+reference.nodeId.toString())
                    }
                }
                if(!match)
                {
                    var
                    data={"id":reference.nodeId.toString(),"parent":arg,"text":reference.displayName.text.toString()};
                    browse_json.push(data);
                    //console.log("adding",data)
                }
                else
                {
                    console.log("data not added to json String")
                }
            }
        });
    }
}

```

```

    }
    //browse_json.push(data);
    event.sender.send('nc_json_reply', browse_json);
  }
})
if(response.results[0].continuationPoint!=null)
{
  continuation_points=response.results[0].continuationPoint
}
else
{
  continuation_points=null
}
// response.responseHeader.serviceResult.should.equal(StatusCodes.BadNothingToDo);
callback();
});
}
},

function (callback) {
  if(continuation_points!=null)
  {
    const browseNextRequest = new opcua.BrowseNextRequest({
      continuationPoints: [continuation_points]
    });
    the_session.performMessageTransaction(browseNextRequest, function (err, response) {
      console.log(response.toString());
      response.results[0].references.forEach( function(reference) {
        //if(!err)
        //{
          console.log( reference.browseName.toString());

          //event.sender.send('browse_reply', reference.browseName.toString()+" [ Node id :
"+reference.nodeId.toString()+" ]");

          //var data1=JSON.stringify(browse_json)
          if(reference.referenceTypeId != "ns=0;i=40")
          {
            var match=false;
            for(var i=0; i<browse_json.length; i++){
              if(browse_json[i].id == reference.nodeId.toString()){
                match=true
                // console.log("match detected",browse_json[i].id )
                break;
              }
            }
            else
            {
              //console.log("nor equal : "+browse_json[i].id+reference.nodeId.toString())
            }
          }
          if(!match)
          {
            var
            data={"id":reference.nodeId.toString(),"parent":arg,"text":reference.displayName.text.toString()};
            browse_json.push(data);

```

```

        //console.log("adding",data)
    }
    else
    {
        console.log("data not added to json String")
    }
    //browse_json.push(data);
    event.sender.send('nc_json_reply', browse_json);
}
})
if(response.results[0].continuationPoint!=null)
{
    continuation_points=response.results[0].continuationPoint
}
else
{
    continuation_points=null
}
// response.responseHeader.serviceResult.should.equal(StatusCodes.BadNothingToDo);
callback();
});
}
},

function (callback) {
    if(continuation_points!=null)
    {
        const browseNextRequest = new opcua.BrowseNextRequest({
            continuationPoints:[continuation_points]
        });
        the_session.performMessageTransaction(browseNextRequest, function (err, response) {
            console.log(response.toString());
            response.results[0].references.forEach( function(reference) {
                //if(!err)
                //{
                console.log( reference.browseName.toString());

                //event.sender.send('browse_reply', reference.browseName.toString()+" [ Node id :
                "+reference.nodeId.toString()+" ]");

                //var data1=JSON.stringify(browse_json)
                if(reference.referenceTypeId != "ns=0;i=40")
                {
                    var match=false;
                    for(var i=0; i<browse_json.length; i++){
                        if(browse_json[i].id == reference.nodeId.toString()){
                            match=true
                            // console.log("match detected",browse_json[i].id )
                            break;
                        }
                    }
                    else
                    {
                        //console.log("nor equal : "+browse_json[i].id+reference.nodeId.toString())
                    }
                }
            }
        }
    }
}

```

```

        if(!match)
        {
            var
data={"id":reference.nodeId.toString(),"parent":arg,"text":reference.displayName.text.toString()};
            browse_json.push(data);
            //console.log("adding",data)
        }
        else
        {
            console.log("data not added to json String")
        }
        //browse_json.push(data);
        event.sender.send('nc_json_reply', browse_json);
    }
    })
    if(response.results[0].continuationPoint!=null)
    {
        continuation_points=response.results[0].continuationPoint
    }
    else
    {
        continuation_points=null
    }
    // response.responseHeader.serviceResult.should.equal(StatusCodes.BadNothingToDo);
    callback();
});
}
},

],
function(err){
    if(err) {
        console.log(" failure ",err);
        //client.disconnect(function({});
        dialog.showErrorBox("Session Error, Click Again, We will try to reconnect you", "Error : "+err)
        client.createSession(userIdentity,function(err, session) {
            if(!err) {
                the_session = session;
                console.log("session created")
                event.sender.send('connect_reply', 'Session created successfully !');
                session_id=session.sessionId.toString();
                session_created=true
            }
            else
            {
                event.sender.send('connect_reply', 'Session creation failed !');
                console.log("session creation failed")
                session_created=false
            }
            callback(err);
        });
    } else {

```

```

        console.log("done!")
    }
    //client.disconnect(function(){});
})

});

ipcMain.on('download_address_space', (event, arg) => {
    console.log(arg);
    var xls = json2xls(browse_json);
    var time_identifier = (new Date).getTime();
    //fs.writeFileSync('data.xlsx', xls, 'binary');
    try {
        fs1.writeFileSync(download_folder+"/"+machine_name+"_"+machine_no+"_"+machine_location+"_"+
            address_space.xlsx', xls, 'binary');
        dialog.showErrorBox("Address space downloaded",
            download_folder+"/"+machine_name+"_"+machine_no+"_"+machine_location+"_"+time_identifier+"_"
            +'address_space.xlsx')
        //alert('Data saved as : data.xlsx')
    }

    catch(e) {
        console.log("Error : "+e)
        const response = dialog.showMessageBox("Data cannot be saved : "+e);
    }
});

ipcMain.on('create_server', (event, arg) => {

    data=JSON.parse(arg)
    console.log("Creating Server")
    event.sender.send('create_server_reply', "Creating OPC-UA Server");
    server_parameters=server_parameters+"\nCreating OPC-UA Server"
    const opcua = require("node-opcua");

    if(data.type=="connect")
    {
        server = new opcua.OPCUAServer({
            port: server_port,
            timeout:1000000,
            maxAllowedSessionNumber:20,
            resourcePath: data.variable_name, // this path will be added to the endpoint resource name
            buildInfo:{
                productName: data.variable_name,
                buildNumber: "1.0",
                buildDate: new Date(2019,7,16)
            }
        })
    }
});

function post_initialize() {
    console.log("initialized");
    event.sender.send('create_server_reply', "Server Initializing\n PORT : "+server_port+"\nRESOURCE
    PATH : "+data.variable_name+"\nPRODUCT NAME : "+data.variable_name+"\nBUILD NUMBER : 1.0\n
    BUILD DATE : 16-07-2019");
}

```

```

server_parameters=server_parameters+"\nServer Initializing\n PORT : "+server_port+"\nRESOURCE
PATH : "+data.variable_name+"\nPRODUCT NAME : "+data.variable_name+"\nBUILD NUMBER : 1.0\n
BUILD DATE : 16-07-2019"
function construct_my_address_space(server) {
    event.sender.send('create_server_reply', "Adding variables in the Server ...");
    server_parameters=server_parameters+"\nAdding variables in the Server ..."

    const addressSpace = server.engine.addressSpace;
    const namespace = addressSpace.getOwnNamespace();

    // declare a new object
    const device = namespace.addObject({
        organizedBy: addressSpace.rootFolder.objects,
        browseName: "MyDevice"
    });

    add_eventGeneratorObject(namespace, device);

    // add some variables
    // add a variable named MyVariable1 to the newly created folder "MyDevice"
    let variable1 = 1;

    // emulate variable1 changing every 500 ms
    setInterval(function() { variable1+=1; }, 500);

    namespace.addVariable({
        componentOf: device,
        attributeIds: opcua.AttributeIds.Value,
        browseName: "MyVariable1",
        dataType: "Double",
        value: {
            get: function () {
                return new opcua.Variant({dataType: opcua.DataType.Double, value: variable1 });
            }
        }
    });

    const variable6 = 10.0;

    server.nodeVariable1 = namespace.addVariable({
        componentOf: device,
        nodeId: "s=Simulated_variable",
        browseName: "Simulated_Variable",
        dataType: "Double",
        value: {
            get: () => {
                const t = new Date() / 10000.0;
                const value = variable6 + 10.0 * Math.sin(t);
                return new opcua.Variant({dataType: opcua.DataType.Double, value: value});
            }
        }
    });
};

```

```

const method = namespace.addMethod(device,{

  browseName: "N-Root",

  inputArguments: [
    {
      name:"Number",
      description:{ text: "specifies the number whose nth root is to be calculated" },
      dataType: opcua.DataType.Int32
    },{
      name:"root_level",
      description:{ text: "specifies the root level, example 2 is square root 3 is cube root" },
      dataType: opcua.DataType.Int32
    }
  ],

  outputArguments: [{
    name:"Nth-root",
    description:{ text: "Nth root of a number" },
    dataType: opcua.DataType.Double ,
    valueRank: 1
  }]
});

// optionally, we can adjust userAccessLevel attribute
method.outputArguments.userAccessLevel = opcua.makeAccessLevelFlag("CurrentRead");
method.inputArguments.userAccessLevel = opcua.makeAccessLevelFlag("CurrentRead");

method.bindMethod((inputArguments,context,callback) => {

  const number_val = inputArguments[0].value;
  const root_val = inputArguments[1].value;

  // console.log("Hello World ! I will bark ",nbBarks," times");
  //console.log("the requested volume is ",volume,"");
  //const sound_volume = Array(volume).join("!");

  const barks = Math.pow(number_val, 1/root_val);

  const callMethodResult = {
    statusCode: opcua.StatusCodes.Good,
    outputArguments: [{
      dataType: opcua.DataType.Double,
      arrayType: opcua.VariantArrayType.Scalar,
      value:barks
    }]
  };
  callback(null,callMethodResult);
});

const triggerNode = namespace.addVariable({
  browseName: "Events-Notifier",
  nodeId: "s=Simulated-Event",
  componentOf: device,
  dataType: "Double",

```



```

typeDefinition: makeNodeId(68)
});

let value = 100.0;
const getFunc = function () {
    return new Variant({
        dataType: DataType.Double,
        value: value
    });
};
const setFunc = function (variant) {

    value = variant.value;

    const server1 = addressSpace.rootFolder.objects.server;

    server1.raiseEvent("1:MyEventType", {
        message: {
            dataType: DataType.LocalizedText,
            value: {text: "Hello World"}
        },
        severity: {
            dataType: DataType.UInt32,
            value: 32
        }
    });
};

const options = {
    get: getFunc,
    set: setFunc
};
triggerNode.bindVariable(options);

const fullPath = download_folder + "/" + 'read_file.txt';
const fs_node_read = require('fs')
// fs.writeFileSync(my_data_filename, "some content", "utf8");
// fileType = addressSpace.findObjectType("FileType");

// var opcuaFile = fileType.instantiate({
//     nodeId: "s=MyFile",
//     browseName: "MyFile",
//     organizedBy: addressSpace.rootFolder.objects,
// })

const imageNode = namespace.addVariable({
    componentOf: device,
    browseName: "FileRead",
    description: {locale: "en", text: "Read File"},
    nodeId: "s=File_Read",
    dataType: opcua.DataType.ByteString,
});

```

```

const options111={
  refreshFunc: function (callback){
    fs_node_read.readFile(fullpath,function (err, data) {
      if(err) {
        return callback(null,new DataValue({
          statusCode: StatusCodes.BadInternalError,
          value: {dataType: "ByteString", value: null}

          }));
        console.log("File read data : "+data)
      }
      assert(data instanceof Buffer);
      callback(null, new DataValue({value: {dataType: "ByteString", value: data}}));
      //callback(null, new DataValue({value: {dataType: "ByteString", value: data}}));
    });
  }
};
imageNode.bindVariable(options111,/*overwrite=*/false);

/* installFileType(opcuaFile, {
  filename: my_data_filename
});*/

// add a variable named MyVariable2 to the newly created folder "MyDevice"
let variable2 = 10.0;

namespace.addVariable({

  componentOf: device,
  attributId: opcua.AttributIds.Value,
  nodeId: "ns=1;b=1020FFAA", // some opaque NodeId in namespace 4

  browseName: "MyVariable2",

  dataType: "Double",

  value: {
    get: function () {
      return new opcua.Variant({dataType: opcua.DataType.Double, value: variable2 });
    },
    set: function (variant) {
      variable2 = parseFloat(variant.value);
      return opcua.StatusCodes.Good;
    }
  }
});
const os = require("os");
/**
 * returns the percentage of free memory on the running machine
 * @return {double}
 */
function available_memory() {
  // var value = process.memoryUsage().heapUsed / 1000000;
  const percentageMemUsed = os.freemem() / os.totalmem() * 100.0;
  return percentageMemUsed;
}

```

```

    }

    function available_memory_binary() {
        // var value= process.memoryUsage().heapUsed / 1000000;
        const percentageMemUsed = os.freemem() / os.totalmem() * 100.0;
        if (percentageMemUsed > 0.5)
        {
            return true;
        }
        else
        {
            return false;
        }
    }

    namespace.addVariable({

        componentOf: device,
        attributId: opcua.AttributIds.Value,
        nodeId: "s=free_memory", // a string nodeId
        browseName: "FreeMemory",
        dataType: "Double",
        value: {
            get: function () {return new opcua.Variant({dataType: opcua.DataType.Double, value:
available_memory_binary() });}
        }
    });

    namespace.addVariable({

        componentOf: device,
        attributId: opcua.AttributIds.Value,
        nodeId: "s=binary_signal", // a string nodeId
        browseName: "BinaryData",
        dataType: "Boolean",
        value: {
            get: function () {return new opcua.Variant({dataType: opcua.DataType.Boolean, value:
available_memory_binary() });}
        }
    });

    let variable3 = false;

    namespace.addVariable({

        componentOf: device,
        attributId: opcua.AttributIds.Value,
        nodeId: "s=binary_signal_test", // a string nodeId
        browseName: "BinaryDataTest",
        dataType: "Boolean",
        value: {
            get: function () {return new opcua.Variant({dataType: opcua.DataType.Boolean, value: variable3
});},
            set: function (variant) {
                variable3 = variant.value
                console.log("variant value :", variant.value)
            }
        }
    });

```

```

        return opcua.StatusCodes.Good;
    }
}
});

}
construct_my_address_space(server);
server.start(function() {
    event.sender.send('create_server_reply', "Constructing Address Space in the Server");
    console.log("Server is now listening...( press CTRL+C to stop)");
    console.log("port", server.endpoints[0].port);
    const endpointUrl = server.endpoints[0].endpointDescriptions()[0].endpointUrl;
    console.log("the primary server endpoint url is ", endpointUrl);
    event.sender.send('create_server_reply', "Server Created Successfully...\nServer is now
listening...\nPORT: "+server.endpoints[0].port+"\n The Primary Server EndPoint URL is :
"+endpointUrl);
    server_parameters=server_parameters+"\nConstructing Address Space in the Server\nServer
Created Successfully...\nServer is now listening...\nPORT: "+server.endpoints[0].port+"\n The Primary
Server EndPoint URL is : "+endpointUrl
    //event.sender.send('create_server_reply1', "disconnect");
    server_status=true;
    server_json_reply={"text":server_parameters,"status":server_status}
    event.sender.send('server_enquiry_reply',JSON.stringify(server_json_reply))

});
}
server.initialize(post_initialize);
}
else{
    //server_parameters="";
}

});

ipcMain.on('server_enquiry',(event, arg) => {
    //console.log(arg);

    //console.log("Server parameter length : "+server_parameters.length)
    server_json_reply={"text":server_parameters,"status":server_status}

    event.sender.send('server_enquiry_reply',JSON.stringify(server_json_reply))
});

ipcMain.on('shut_down_server', (event, arg) => {
    //event.sender.send('setting_app_path_rep',app.getPath('userData')+"/")
    server.shutdown(function(err)
    {
        if(!err)
        {
            event.sender.send('create_server_reply', "Server Stopped");
            server_status=false

```

```

        server_json_reply={"text":"Server Stopped","status":server_status}
        event.sender.send('server_enquiry_reply',JSON.stringify(server_json_reply))
    }
}
)
});

ipcMain.on('subscribe_data',(event, arg)=> {

    var data=JSON.parse(arg)
    signal_identifier=data.signal_identifier

    if(data.type=="stop_subscription")
    {
        try{
            the_subscription.terminate();
            subscribed_nodes=[];
            //subscribed_data=[];
        }
        catch(e)
        {

        }
    }

    else

    {

        //subscribed_data.push({"Machine Name ":"machine_name","machine_no":machine_no})
        //subscribed_data.push({"Machine Location ":"machine_location","Signal Name":machine_no})

        async.series([

        function(callback){
            try{
                the_subscription=opcua.ClientSubscription.create(the_session,{
                    requestedPublishingInterval: pub_interval,
                    requestedLifetimeCount: lifetime_count,
                    requestedMaxKeepAliveCount: keep_alive_count,
                    maxNotificationsPerPublish: max_not,
                    publishingEnabled: publishing_enabled,
                    priority: priority
                });

                the_subscription.on("started",function(){
                    console.log("subscription started for 2 seconds - subscriptionId=",the_subscription.subscriptionId);
                    event.sender.send('subscribe_reply',"subscription started -
subscriptionId="+the_subscription.subscriptionId)
                }).on("keepalive",function(){
                    //event.sender.send('subscribe_reply',"keepalive")

```

```

        console.log("keepalive");
    }).on("terminated",function(){
        event.sender.send('subscribe_reply',"Subscription Terminated")
    });
}
catch(e)
{
    dialog.showErrorBox("Subscription cannot be started","Error in subscription service")
}

/*setTimeout(function(){
    the_subscription.terminate(callback);
},1000000);*/

// install monitored item
/* var monitoredItem = the_subscription.monitor({
    nodeId: opcua.resolveNodeId(data.data),
    attributeId: opcua.AttributeIds.Value
},
{
    samplingInterval: 100,
    discardOldest: true,
    queueSize: 10
},
opcua.read_service.TimestampsToReturn.Both
);
console.log("-----");

monitoredItem.on("changed",function(dataValue){
    console.log(" % free mem = ",dataValue.value.value);
    // event.sender.send('subscribe_reply',dataValue.value.value+"\n"+dataValue.serverTimestamp)
    data={"x":dataValue.serverTimestamp,"y":dataValue.value.value}
    data1={"Signal Name":signal_identifier,"Server Time
Stamp":dataValue.serverTimestamp,"Value":dataValue.value.value}
    subscribed_data.push(data1)
    event.sender.send('subscribed_value',JSON.stringify(data))
});*/

},

],

function(err){
    if(err){
        console.log(" failure ",err);
        //client.disconnect(function({});
        dialog.showErrorBox("Error", "Error : "+err)
    } else {
        console.log("done!")
    }
}
// client.disconnect(function({});

```

```

    })

    }

});

ipcMain.on('subscription_start_data',(event, arg)=> {

    try
    {

        var data_rx=JSON.parse(arg)
        subscribed_nodes.push(data_rx);
        console.log("count is : "+data_rx.count)

        // var itemtomonitor={nodeId: opcua.resolveNodeId(data_rx.data),attributeId:
opcua.AttributeIds.Value}
        //var requestedParameters={samplingInterval: 100,discardOldest: true,queueSize: 10}
        // var monitoredItem=await the_subscription.monitor(itemtomonitor,requestParameters)

        var monitoredItem = opcua.ClientMonitoredItem.create(the_subscription,{
            nodeId: opcua.resolveNodeId(data_rx.data),
            attributeId: opcua.AttributeIds.Value
        },
        {
            samplingInterval: 100,
            discardOldest: true,
            queueSize: 10,
            //timestampsToReturn:opcua.TimestampsToReturn.Both
        },
        opcua.TimestampsToReturn.Both
        );
        console.log("-----");

        monitoredItem.on("changed",function(dataValue){
            console.log(data_rx.data,dataValue.toString());
            if(dataValue.value.value==true)
            {
                dataValue.value.value=1;
            }
            else if (dataValue.value.value==false)
            {
                dataValue.value.value=0;
            }
        })
        // event.sender.send('subscribe_reply',dataValue.value.value+"\n"+dataValue.serverTimestamp)
        var
        data={"x":dataValue.serverTimestamp,"y":dataValue.value.value,"count":data_rx.count,"node_id":data
_rx.data,"signal_name":data_rx.signal_identifier}
        var data1={"Node Id":data_rx.data,"Signal Name":data_rx.signal_identifier,"Server Time
Stamp":dataValue.serverTimestamp,"Value":dataValue.value.value}
        subscribed_data.push(data1)
        event.sender.send('subscribed_value',JSON.stringify(data))
    });
});

```

```

}catch(e)
{
    //var data1={"Node Id":data_rx.data,"Signal Name":data_rx.signal_identifier,"Server Time Stamp":dataValue.serverTimestamp,"Value":dataValue.value.value}
    //subscribed_data.push(data1)
    var data={"x":"","y":"","count":"0"}
    event.sender.send('subscribed_value',JSON.stringify(data))
    dialog.showErrorBox("Subscription cannot be added", 'Subscription not started :'+e)

}

});

ipcMain.on('enquire_subscribed_data',(event, arg) => {
    event.sender.send('enquire_subscribed_data_reply',subscribed_nodes)
});

ipcMain.on('favourite_data',(event, arg) => {
    createfavouriteWindow();
});

ipcMain.on('download_subscribed_data',(event, arg) => {
    console.log(arg);
    var xls = json2xls(subscribed_data);
    var time_identifier = (new Date).getTime();
    //fs.writeFileSync('data.xlsx',xls,'binary');
    try { fs1.writeFileSync(download_folder+"/"+time_identifier+"_"+data.xlsx', xls,'binary');
    dialog.showErrorBox("Signal data downloaded",download_folder+"/"+time_identifier+"_"+data.xlsx')
    //alert('Data saved as : data.xlsx')
    }

    catch(e) {
        console.log("Error : "+e)
        dialog.showErrorBox("File can not be saved",e)

        //const response= dialog.showMessageBox("Data cannot be saved : "+e);
    }
});

ipcMain.on('write_data',(event, arg) => {
    var write_data=JSON.parse(arg)
    var data_type

    async.series([

        function(callback){
            var maxAge = 10;
            var nodeToRead = { nodeId: write_data.nodeid, attributId: opcua.AttributIds.Value };
            the_session.read(nodeToRead, maxAge, function(err,dataValue) {
                if (!err) {
                    event.sender.send('write_reply', "Previous Value of "+write_data.nodeid+" is : "
+dataValue.value.value);
                    data_type=dataValue.value.dataType
                }
            })
        }
    ], callback)
});

```



```

    }
    callback(err);
  });

},

function(callback){
  let nodeToWrite
  console.log("Node Id to be written",write_data.nodeid)
  if(write_data.value==="true" && data_type==opcua.DataType.Boolean)
  {
    console.log("True is selected")

    nodeToWrite = {
      nodeId: write_data.nodeid,
      attributeld: opcua.Attributelds.Value,
      value: /* DataValue */ {
        sourceTimestamp: new Date(),
        value: /* Variant */ {
          dataType: data_type,
          value: true
        }
      }
    },
  };
}

else if(write_data.value==="false" && data_type==opcua.DataType.Boolean)
{
  console.log("false is selected")
  nodeToWrite = {
    nodeId: write_data.nodeid,
    attributeld: opcua.Attributelds.Value,
    value: /* DataValue */ {
      sourceTimestamp: new Date(),
      value: /* Variant */ {
        dataType: data_type,
        value: false
      }
    }
  },
};
}

else{

  nodeToWrite = {
    nodeId: write_data.nodeid,
    attributeld: opcua.Attributelds.Value,
    value: /* DataValue */ {
      sourceTimestamp: new Date(),
      value: /* Variant */ {
        dataType: data_type,
        value: write_data.value
      }
    }
  },
};
};

```

```

    }

    the_session.write(nodeToWrite, function(err,statusCode) {
        if(!err)
        {
            console.log(nodeToWrite)
            event.sender.send('write_reply', "value written to variable:["+"Status Code : "+statusCode+" ]");
            console.log("Data type",data_type)
        }
        else
        {
            event.sender.send('write_reply', "value written to variable failed");
            console.log("Data type",data_type)
        }

        callback(err);
    });

},

function(callback){
    var maxAge = 0;
    var nodeToRead = { nodeId: write_data.nodeid, attributeld: opcua.Attributelds.Value };
    the_session.read(nodeToRead, maxAge, function(err,dataValue){
        if(!err) {
            event.sender.send('write_reply', "New verified Value of "+write_data.nodeid+" is : "
+dataValue.value.value+"\n");
            data_type=dataValue.value.dataType
        }
        callback(err);
    });

},

],

function(err){
    if(err) {
        console.log(" failure ",err);
        //client.disconnect(function(){});
        dialog.showErrorBox("Error", "Error : "+err)
    } else {
        console.log("done!")
    }
    // client.disconnect(function(){});
}
});

```

```

ipcMain.on('alarm_monitoring_start',(event, arg) => {
  console.log("Alarm Monitoring Started")
  if(arg=="stop_monitoring")
  {
    the_subscription.terminate();
  }
  else
  {
    async.series([

function(callback){
  try{
    the_subscription=opcua.ClientSubscription.create(the_session,{
      requestedPublishingInterval: pub_interval,
      requestedLifetimeCount: lifetime_count,
      requestedMaxKeepAliveCount: keep_alive_count,
      maxNotificationsPerPublish: max_not,
      publishingEnabled: publishing_enabled,
      priority: priority,
      localeIds:['en-us']

    });

    the_subscription.on("started",function(){
      console.log("subscription started for 2 seconds - subscriptionId=",the_subscription.subscriptionId);
      event.sender.send('alarm_monitoring_reply',"subscription started -
subscriptionId="+the_subscription.subscriptionId)
    }).on("keepalive",function(){
      //event.sender.send('subscribe_reply',"keepalive")
      console.log("keepalive");
    }).on("terminated",function(){
      event.sender.send('alarm_monitoring_reply',"Subscription Terminated")
    });
  }
  catch(e)
  {
    dialog.showErrorBox("Subscription cannot be started","Error in subscription service")
  }

  try{

var fields = [
  "EventId",
  "EventType",
  "SourceNode",
  "SourceName",
  "Time",
  "ReceiveTime",
  "Message",
  "Severity",
  "LocalTime",

```

```

        //ConditionType
        "ConditionClassId",
        "ConditionClassName",
        "ConditionName",
        "BranchId",
        "Retain",
        "Quality",
        "LastSeverity",

        // AcknowledgeConditionType
        "AkedState",
        "ConfirmedState",

        // AlarmConditionType
        "AlarmIdentifier",
        "AuxParameters"

    ];

    var fields1 = [
        "Message",
        "EventId",
        "Time",
        "ReceiveTime",
    ];

    var fields2 = [];

    //var baseEventTypeId = "i=2041";
    //var eventFilter = opcua.constructEventFilter(fields1);
    var eventFilter = new opcua.constructEventFilter(fields1,[opcua.resolveNodeId("ConditionType")]);

    event.sender.send('alarm_monitoring_reply',"Monitoring NodeId : "+arg)
    try
    {
        var event_monitoringItem = opcua.ClientMonitoredItem.create(the_subscription,
        {
            //nodeId: arg,
            nodeId:opcua.resolveNodeId(arg),
            attributeId: opcua.AttributeIds.EventNotifier,
            indexRange: null,
            dataEncoding: {namespaceIndex:0,name:null}
        },
        {
            queueSize: 100000,
            filter: eventFilter,
            discardOldest: true
        }, opcua.TimestampsToReturn.Neither
    );
    var data_event=""

```

```

    }catch(e)
    {
        console.log("Error in starting of monitoring : "+e)
    }

    event_monitoringItem.on("changed",function(dataValue){
        console.log("on changed ran")
        data_event=""
        dataValue.forEach(function (variant, index) {
            data_event=data_event+" "+variant.value

        });
        event.sender.send('alarm_monitoring_reply',data_event)

        console.log("value_changed : ", dataValue.toString())
    });

    event_monitoringItem.on("err", function (err_message) {
        console.log("event_monitoringItem ", err_message);

        event.sender.send('alarm_monitoring_reply',"Error in event_monitoringItem : "+err_message)
    });
}
catch(e)
{
    dialog.showErrorBox("Error in subscription service:" ,e)
}

},

],
function(err){
    if(err){
        console.log(" failure ",err);
        //client.disconnect(function(){});
        dialog.showErrorBox("Error", "Error : "+err)
    } else {
        console.log("done!")
    }
    // client.disconnect(function(){});
})
}

});

```

```
ipcMain.on('method_call', (event, arg) => {
  var data = JSON.parse(arg)
  var object_nodeid = data.object_nodeid
  var method_nodeid = data.method_nodeid
  var value = data.value1
  var data_type = data.datatype1
  var array_type = data.arraytype1
  var DataType = opcua.DataType;
  var datatype = DataType.Double;

  var value2 = data.value2
  var data_type2 = data.datatype2
  var array_type2 = data.arraytype2
  var DataType2 = opcua.DataType;
  var datatype2 = DataType2.Double;

  var arrayType = opcua.VariantArrayType;
  var arraytype = arrayType.Scalar;
  var arrayType2 = opcua.VariantArrayType;
  var arraytype2 = arrayType2.Scalar;

  if(data_type === "double")
  {
    datatype = DataType.Double
  }
  else if(data_type === "string")
  {
    datatype = DataType.String
  }
  else if(data_type === "float")
  {
    datatype = DataType.Float
  }
  else if(data_type === "int16")
  {
    datatype = DataType.Int16
  }
  else if(data_type === "int32")
  {
    datatype = DataType.Int32
  }
  else if(data_type === "int64")
  {
    datatype = DataType.Int64
  }
  else if(data_type === "uint16")
  {
    datatype = DataType.UInt16
  }
  else if(data_type === "uint32")
  {

```

```
        datatype=DataType.UInt32
    }
    else if(data_type=="uint64")
    {
        datatype=DataType.UInt64
    }

    if(data_type2=="double")
    {
        datatype2=DataType2.Double
    }
    else if(data_type2=="string")
    {
        datatype2=DataType2.String
    }
    else if(data_type2=="float")
    {
        datatype2=DataType2.Float
    }
    else if(data_type2=="int16")
    {
        datatype2=DataType2.Int16
    }
    else if(data_type2=="int32")
    {
        datatype2=DataType2.Int32
    }
    else if(data_type2=="int64")
    {
        datatype2=DataType2.Int64
    }
    else if(data_type2=="uint16")
    {
        datatype2=DataType2.UInt16
    }
    else if(data_type2=="uint32")
    {
        datatype2=DataType2.UInt32
    }
    else if(data_type2=="uint64")
    {
        datatype2=DataType2.UInt64
    }

    if(array_type=="scalar")
    {
        arraytype=arrayType.Scalar;
    }
    else if(array_type=="array")
    {
```

```

    arraytype=arrayType.Array;
}

if(array_type2=="scalar")
{
    arraytype2=arrayType2.Scalar;
}
else if(array_type2=="array")
{
    arraytype2=arrayType2.Array;
}

if(data_type2=="None" | array_type2=="None")
{
    console.log("executing single argument",data_type2+arrayType2)
    async.series([

        function (callback) {

            var methodsToCall = [];
            methodsToCall.push({
                objectId: object_nodeid,
                methodId: method_nodeid,
                inputArguments: [{
                    dataType: datatype,
                    arrayType: arraytype,
                    value: value}
                ] //OK
            });
            the_session.call(methodsToCall,function(err,results){
                try
                {
                    if(results[0].statusCode===StatusCodes.Good){
                        if(results[0].outputArguments[0]!=null)
                        {
                            final_result="Return Value : "+results[0].outputArguments[0].value+"\nStatus Code : Good"
                            event.sender.send("method_call_reply","Method Call executed succesfully\n"+final_result)
                        }
                        else{
                            final_result="Status Code : Good"
                            event.sender.send("method_call_reply","Method Call executed succesfully\n"+final_result)
                        }
                    }
                    else
                    {
                        event.sender.send("method_call_reply","Method Call Failed Status Code : 
"+results[0].statusCode)
                    }
                }catch(e)
                {
                    if(results[0].statusCode===StatusCodes.Good){
                        dialog.showErrorBox("Error", "Method call successful, could not read output arguments")
                    }
                    else
                    {

```



```

        dialog.showErrorBox("Error", "Method call failed")
    }
}

    callback(err);
});
},
],
function(err){
    if(err) {
        console.log(" failure ",err);
        //client.disconnect(function({}));
        dialog.showErrorBox("Error", "Error : "+err)
    } else {
        console.log("done!")
    }
    // client.disconnect(function({}));
})

}

else
{
    console.log("executing double argument",data_type2+arrayType2)
    async.series([

        function (callback){

            var methodsToCall = [];
            methodsToCall.push({
                objectId: object_nodeid,
                methodId: method_nodeid,
                inputArguments: [{
                    dataType: datatype,
                    arrayType: arraytype,
                    value: value},
                    {
                        dataType: datatype2,
                        arrayType: arraytype2,
                        value: value2 }
                ] //OK
            });
            the_session.call(methodsToCall,function(err,results){
                try
                {
                    if(results[0].statusCode===StatusCodes.Good){
                        if(results[0].outputArguments[0]!==null)
                        {
                            final_result="Return Value: "+results[0].outputArguments[0].value+"\nStatus Code : Good"
                            event.sender.send("method_call_reply","Method Call executed succesfully\n"+final_result)
                        }
                    }
                    else{
                        final_result="Status Code : Good"
                        event.sender.send("method_call_reply","Method Call executed succesfully\n"+final_result)
                    }
                }
            })
        }
    ])
}

```

```

        }
        else
        {
            event.sender.send("method_call_reply","Method Call Failed Status Code :
"+results[0].statusCode)
        }
    }catch(e)
    {
        if(results[0].statusCode===StatusCodes.Good){
            dialog.showErrorBox("Error", "Method call successful, could not read output arguments")
        }
        else
        {
            dialog.showErrorBox("Error", "Method call failed")
        }
    }

    callback(err);
    });
},
],
function(err) {
    if (err) {
        console.log(" failure ",err);
        //client.disconnect(function(){});
        dialog.showErrorBox("Error", "Error : "+err)
    } else {
        console.log("done!")
    }
    // client.disconnect(function(){});
})

}

console.log("datatypes are ",data)

});

ipcMain.on('write_to_server', (event, arg) => {

    async.series([

        function(callback) {
            try{
                clientFile = new ClientFile(the_session, arg);
                mode = OpenFileMode.ReadWriteAppend;
                clientFile.open(mode);
                callback()
            }catch(e)

```

```

    {
        dialog.showErrorBox("File Mode Not Supported to this node id", "Error : "+e)
    }

},

function(callback){
    try{
        const dataToWrite = Buffer.from("Some data");
        clientFile.write(dataToWrite);
        callback()
    }catch(e)
    {
        dialog.showErrorBox("Data writing failed to thid node id", "Error : "+e)
    }

},

function(callback){
    try{
        clientFile.close();
        callback()
    }catch(e)
    {
        dialog.showErrorBox("File Closing failed", "Error : "+e)
    }

},

],

function(err){
    if(err) {
        console.log(" failure ",err);
        dialog.showErrorBox("Writing to server failed", "Error : "+err)
        //client.disconnect(function(){});
    } else {
        console.log("done!")
    }
    //client.disconnect(function(){});
})

});

ipcMain.on('read_from_server',(event, arg) => {

    async.series([

        function(callback){
            try{
                clientFile= new ClientFile(the_session,arg);
                mode = OpenFileMode.ReadWriteAppend;
                clientFile.open(mode);
                callback()
            }

```

```

    }catch(e)
    {
        dialog.showErrorBox("File Mode Not Supported to this node id", "Error : "+e)
    }

},

function(callback){
    try{
        // const dataToWrite= Buffer.from("Some data");
        const data = clientFile.read(200);
        callback()
    }catch(e)
    {
        dialog.showErrorBox("Data writing failed to thid node id", "Error : "+e)
    }

},

function(callback){
    try{
        // clientFile.close();
        callback()
    }catch(e)
    {
        dialog.showErrorBox("File Closing failed", "Error : "+e)
    }

},

],

function(err){
    if(err) {
        console.log(" failure ",err);
        dialog.showErrorBox("Writing to server failed", "Error : "+err)
        //client.disconnect(function({});
    } else {
        console.log("done!")
    }
    //client.disconnect(function({});
})

});

```

8.2 Configure Machine

```

<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="content-type" content="text/html; charset=windows-1252">

```

```

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>
  body {
    font-family: "Lato", sans-serif;
  }

  h1 {color:blue;
    position:absolute;
    left:80px}

  .sidenav{
    width: 250px;
    position:absolute;
    z-index: 1;
    top: 100px;
    left: 10px;
    background: #eee;
    overflow-x: hidden;
    padding: 8px 0;
  }

  p {
    color:#2196F3;
  }

  select{
    padding:1px;
  }

  .container {
    border-radius:5px;
    padding: 20px;
  }

  /* Floating column for labels: 25% width */
  .col-25 {
    float: left;
    width: 25%;
    margin-top: 2px;
  }

  /* Floating column for inputs: 75% width */
  .col-75 {
    float: left;
    width: 75%;
    margin-top: 2px;
  }

  .sidenav{
    width: 250px;

```

```

position: absolute;
z-index: 1;
top: 100px;
left: 1px;
background: #444444;
overflow-x: hidden;
padding: 1px 0;
height: 100%;
}

.active, .sidenav a {
padding: 6px 8px 6px 16px;
text-decoration: none;
font-size: 14px;
color: #FFFFFF;
display: block;
}

.sidenav a: hover {
color: #C2C2C2;
}

.main {
margin-left: 255px; /* Same width as the sidebar + left position in px */
font-size: 14px; /* Increased text to enable scrolling */
padding: 0px 2px;
}

input[type="text"] {
width: 80%;
margin-bottom : 6px;
}

input[type="password"] {
width: 80%;
margin-bottom : 6px;
}

input[type="checkbox"] {
margin-bottom : 6px;
}

input[type="button"], input[type="submit"], input[type="reset"] {
background-color: #4CAF50;
border: none;
color: white;
padding: 8px 16px;
text-decoration: none;
margin: 4px 2px;
cursor: pointer;
margin-bottom : 6px;
}

```

```

}

.vl {
border-right: 1px solid green;
height: 100%;
top: 100px;
position: absolute;
left: 270px;
margin-left: -3px;
}

@media screen and (max-height: 450px) {
.sidenav {padding-top: 15px;}
.sidenav a {font-size: 18px;}
}

hr {
position: absolute;
width: 100%;
left: 2px;
}
</style>

</head>
<body> 
<h1 style="text-align: center; margin-top: -50px;"> ConnectOPC-UA </h1>

<div id="main_menu" class="sidenav"><a href="index.html"><font color="#FF5733">Configure
Machine</font></a><a href="connect.html">Connect Machine</a> <a href="browse.html">Browse
Address Space</a> <a href="monitor.html">Monitor variables</a><a href="write.html">Write
variables to the server</a><a href="pubsub.html">Create
Pub-Sub/Notifications</a><a href="pubsub_mult.html">Multiple Pub-Sub/Notifications</a><a
href="method_call.html">Method Call (Address Space)</a><a href="alarm_monitoring.html">Event
Notifier</a><a href="filetransfer.html">File Transfer</a> <a href="opcuaserver.html">Create OPC-UA
Server</a><a href="settings.html">Settings</a><a href="help.html">Help</a></div>
<div class="main">

<h2>Configure Machine</h2>
<div class="container">

<div class="row">
<div class="col-25"><label for="fname">Machine Name</label> </div>
<div class="col-75"><input name="machine_name" id="machine_name" type="text"></div>
</div>

<div class="row">
<div class="col-25"><label for="fname">Machine Number</label> </div>
<div class="col-75"><input name="machine_no" id="machine_no" type="text"></div> </div>

<div class="row">
<div class="col-25"><label for="fname">Machine Location</label></div>

```

```

        <div class="col-75"><input name="machine_location" id="machine_location"
type="text"></div> </div>

        <div class="row">
            <div class="col-25"><label for="fname">OPC URL ADDRESS</label> </div>
            <div class="col-75"><input name="url_address" id="url_address" type="text"></div> </div>

        <div class="row">
            <div class="col-25"><label for="fname">Anonymous Connection</label> </div>
            <div class="col-75"><input name="connection_type" value="male" unchecked=""
type="checkbox"></div> </div>

        <div class="row">
            <div class="col-25"><label for="fname">Username</label> </div>
            <div class="col-75"><input name="username" id="username" type="text"></div> </div>

        <div class="row">
            <div class="col-25"><label for="fname">Password</label></div>
            <div class="col-75"><input name="password" id="password" type="password"></div> </div>

        <div class="row">
            <div class="col-25"><label for="fname">Security Policy</label></div>
            <div class="col-75"><select id="security_policy">
                <option value="None">None</option>
                <option value="Basic128RSA15">Basic128RSA15</option>
                <option value="Basic256">Basic256</option>
            </select></div> </div>

        <div class="row">
            <div class="col-25"><label for="fname">Message Security Mode</label> </div>
            <div class="col-75"><select id="msg_sec_mode">
                <option value="None">None</option>
                <option value="Sign">Sign</option>
                <option value="SignAndEncrypt">SignAndEncrypt</option>
            </select></div> </div>
        <input type="button" value="Save" id="connect" onclick="myFunction();">

    </div>
</div>

<script>
const {ipcRenderer} = require('electron')
document.getElementById('machine_name').value="OPC Machine"
document.getElementById('machine_no').value="123"
document.getElementById('machine_location').value="BLK2BAY1"

//alert("I am an alert box!");
function myFunction(){
    var username=document.getElementById('username').value
    var password=document.getElementById('password').value
    var url_address=document.getElementById('url_address').value
    var machine_name=document.getElementById('machine_name').value
    var machine_no=document.getElementById('machine_no').value

```



```

var machine_location=document.getElementById('machine_location').value
var sec_policy_element=document.getElementById('security_policy')
var security_policy=sec_policy_element.options[sec_policy_element.selectedIndex].value
var msg_sec_mode_element=document.getElementById('msg_sec_mode')
var
msg_sec_mode=msg_sec_mode_element.options[msg_sec_mode_element.selectedIndex].value

//alert(username);
var config_data={}
var key='opc_connect_config_data'
config_data[key]=[];
var
data={'machine_name':machine_name,'machine_no':machine_no,'machine_location':machine_location,'url_address':url_address,'username':username,'password':password,'security_policy':security_policy,'msg_sec_mode':msg_sec_mode};
config_data[key].push(data)

ipcRenderer.send('conf_data',JSON.stringify(data))

};

ipcRenderer.on('conf_data_reply', (event, arg) => {
    alert(arg)
})

</script>
</body>
</html>

```

8.3 Connect Machine

```

<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=windows-1252">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <style>
    body {
      font-family: "Lato", sans-serif;
    }

    h1 {color:blue;
      position:absolute;
      left:80px}

    .sidenav{
      width: 250px;
      position:absolute;

```

```

z-index: 1;
top: 100px;
left: 1px;
background: #444444;
overflow-x: hidden;
padding: 1px 0;
height: 100%;
}

.active, .sidenav a {
padding: 6px 8px 6px 16px;
text-decoration: none;
font-size: 14px;
color: #FFFFFF;
display: block;
}

.sidenav a: hover {
color: #C2C2C2;
}

.main {
margin-left: 255px; /* Same width as the sidebar + left position in px */
font-size: 14px; /* Increased text to enable scrolling */
padding: 0px 2px;
}

.status_div {
right: 0;
font-size: 14px;
position: absolute;
padding: 8px;
margin-top: -50px;
}

p {
color: #2196F3;
}

.sidenav {
width: 250px;
position: absolute;
z-index: 1;
top: 100px;
left: 1px;
background: #444444;
overflow-x: hidden;
padding: 1px 0;
height: 100%;
}

.active, .sidenav a {
padding: 6px 8px 6px 16px;
text-decoration: none;
font-size: 14px;

```

```

    color: #FFFFFF;
    display: block;
}

.sidenav a: hover {
    color: #C2C2C2;
}

.main {
    margin-left: 255px; /* Same width as the sidebar + left position in px */
    font-size: 14px; /* Increased text to enable scrolling */
    padding: 0px 2px;
}

input[type="text"] {
    width: 80%;
    margin-bottom : 6px;
}

input[type="password"] {
    width: 80%;
    margin-bottom : 6px;
}

input[type="checkbox"] {
    margin-bottom : 6px;
}

input[type="button"], input[type="submit"], input[type="reset"] {
    background-color: #4CAF50;
    border: none;
    color: white;
    padding: 8px 16px;
    text-decoration: none;
    margin: 4px 2px;
    cursor: pointer;
    margin-bottom : 6px;
}

.vl {
    border-right: 1px solid green;
    height: 100%;
    top: 100px;
    position: absolute;
    left: 270px;
    margin-left: -3px;
}

span {
    padding : 8px;
    margin : 8px;
}

```

```

    @media screen and (max-height: 450px){
      .sidenav {padding-top: 15px;}
      .sidenav a {font-size: 18px;}
    }

</style>

</head>
<body>


<div id="s_div" class="status_div">
  <span id="endpointurl"></span>
</div>
<h1 style="text-align: center; margin-top: -50px;"> ConnectOPC-UA </h1>

  <div class="sidenav"><a href="index.html">Configure Machine</a><a href="connect.html"><font
color="#FF5733">Connect Machine</font></a> <a href="browse.html">Browse Address Space</a> <a
href="monitor.html">Monitor variables</a><a href="write.html">Write variables to the server</a><a
href="pubsub.html">Create
  Pub-Sub/Notifications</a><a href="pubsub_mult.html">Multiple Pub-Sub/Notifications</a><a
href="method_call.html">Method Call (Address Space)</a><a href="alarm_monitoring.html">Event
Notifier</a><a href="filetransfer.html">File Transfer</a> <a href="opcuaserver.html">Create OPC-UA
Server</a><a href="settings.html">Settings</a><a href="help.html">Help</a></div>
  <div class="main">
    <h2>Connect Machine</h2>

    <form id="connect_form">
      <pre>Machine Name    : <span id='machine_name'></span></pre>
      <pre>Machine Nubmber  : <span id='machine_no'></span></pre>
      <pre>Machine Location : <span id='machine_location'></span></pre>
      <pre>OPC Address     : <span id='url_address'></span></pre>
      <pre>Username        : <span id='username'></span></pre>
      <pre>Password         : <span id='password'></span></pre>
      <pre>Security Policy  : <span id='security_policy'></span></pre>
      <pre>Message Sec Mode : <span id='msg_sec_mode'></span></pre>
      <input type="button" value="Connect" id="connect" onclick="myFunction();">
    </form>

    <h3>Logs</h3>
    <p id='logs'></p>
  </div>

<script>
  const {ipcRenderer} = require('electron')
  let apppath
  ipcRenderer.send('connect_path_req', apppath)

  ipcRenderer.on('connect_path_rep', (event, arg) => {
    apppath=arg
    document.getElementById('connect').style.color = "white";

    try {

```

```

const fs = require('fs')
const fs1 = require('fs')
const fileContents = fs.readFileSync(apppath+'opc_conf_data.txt', 'utf8')
const fileContents1 = fs1.readFileSync(apppath+'status_data.txt', 'utf8')
const data = JSON.parse(fileContents)
const data1 = JSON.parse(fileContents1)
//alert(JSON.stringify(data))
document.getElementById('machine_name').innerText=data.machine_name
document.getElementById('machine_no').innerText=data.machine_no
document.getElementById('machine_location').innerText=data.machine_location
document.getElementById('url_address').innerText=data.url_address
document.getElementById('username').innerText=data.username
document.getElementById('password').innerText=data.password
document.getElementById('endpointurl').innerText=data1.endpointurl
document.getElementById('security_policy').innerText=data.security_policy
document.getElementById('msg_sec_mode').innerText=data.msg_sec_mode
if(data1.status.toString()=="Connected")
{
  document.getElementById('endpointurl').style.backgroundColor = "green";
  document.getElementById('endpointurl').style.color = "white";
  document.getElementById('connect').style.backgroundColor = "red";
  document.getElementById('connect').value = "Disconnect";
  // ipcRenderer.send('connect_button_status',JSON.stringify("Disconnected"))

}
else{
  document.getElementById('endpointurl').style.backgroundColor = "red";
  document.getElementById('endpointurl').style.color = "white";
  document.getElementById('connect').style.backgroundColor = "green";
  document.getElementById('connect').value = "Connect";
  //ipcRenderer.send('connect_button_status',JSON.stringify("Connected"))
}

} catch(err) {
  console.error(err)
}
})

//alert("I am an alert box!");
function myFunction(){
  document.getElementById('logs').innerText="Attempting to connect the OPC SERVER..."
  document.getElementById('logs').innerText="";
  var button_status=document.getElementById('connect').value
  var
connect_data={url_address':document.getElementById('url_address').innerText,'username':document.
getElementById('username').innerText,'password':document.getElementById('password').innerText,'co
nnect_status':button_status,'machine_name':document.getElementById('machine_name').innerText,'
machine_no':document.getElementById('machine_no').innerText,'machine_location':document.getEle
mentById('machine_location').innerText,'security_policy':document.getElementById('security_policy').i
nnerText,'msg_sec_mode':document.getElementById('msg_sec_mode').innerText}

```

```

ipcRenderer.send('connect_data',JSON.stringify(connect_data))

}
//const {ipcRenderer} = require('electron')
ipcRenderer.on('connect_reply', (event, arg) => {
  var old_text=document.getElementById('logs').innerText
  var new_text=old_text+'\n'+arg
  document.getElementById('logs').innerText=new_text
  ipcRenderer.send('connect_log',new_text)
  //alert(arg)
})

ipcRenderer.on('status_reply', (event, arg) => {
  var data=JSON.parse(arg)
  // document.getElementById("endpointurl").innerText=data.endpointurl;
  //document.getElementById("status").innerText=data.status;
  if(data.status.toString()=="Connected")
  {
    //document.getElementById('status').innerText="Connected"
    document.getElementById('endpointurl').style.backgroundColor = "green";
    document.getElementById('endpointurl').style.color = "white";
    document.getElementById('connect').style.backgroundColor = "red";
    document.getElementById('connect').value= "Disconnect";
    // ipcRenderer.send('connect_button_status',JSON.stringify("Disconnected"))
  }
  else
  {
    document.getElementById('endpointurl').style.backgroundColor = "red";
    document.getElementById('endpointurl').style.color = "white";
    document.getElementById('connect').style.backgroundColor = "green";
    document.getElementById('connect').value = "Connect";
    // ipcRenderer.send('connect_button_status',JSON.stringify("Connected"))
  }
})

/* ipcRenderer.on('client_connected', (event, arg) => {
  if(arg)
  {
    document.getElementById("connect").innerText="Connected"
    document.getElementById("connect").disabled = true;
    //property.style.backgroundColor = "#d2d2d2"
  }
  else{
    document.getElementById("connect").innerText="Connect"
    document.getElementById("connect").disabled = false;
    //property.style.backgroundColor = "#d2d2d2"
  }
}
})*/
</script>

```

```

</body>
</html>

```

8.4 Browse Address Space

```

<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=windows-1252">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="jtree-dist/themes/default/style.min.css" />
  <script>
    $=require('jquery')
  </script>
  <style>
body {
  font-family: "Lato", sans-serif;
}

h1 {color:blue;
  position:absolute;
  left:80px}

.sidenav{
width: 250px;
position:absolute;
z-index: 1;
top: 100px;
left: 1px;
background: #444444;
overflow-x: hidden;
padding: 1px 0;
height:100%;
}

.active,.sidenav a {
padding: 6px 8px 6px 16px;
text-decoration: none;
font-size: 14px;
color: #FFFFFF;
display: block;
}

.sidenav a: hover {
color: #C2C2C2;
}

.main {
margin-left: 255px; /* Same width as the sidebar + left position in px */
font-size: 14px; /* Increased text to enable scrolling */
padding: 0px 2px;
}
p {

```

```

    color:#2196F3;
}

label {
    color:#9E0909;
    font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode', Geneva,
    Verdana, sans-serif
}

span{
    padding: 6px 6px 1px 6px;
}

input[type="text"] {
    width: 90%;
    margin-bottom : 2px;
    border:none;
    color:#201096;
    font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode', Geneva,
    Verdana, sans-serif
}

.input_text_class{
    width: 80%;
    margin-bottom : 6px;
}

input[type="password"] {
    width: 80%;
    margin-bottom : 6px;
}

input[type="checkbox"] {
    margin-bottom : 6px;
}

input[type=button], input[type=submit], input[type=reset] {
    background-color:rgb(2, 48, 4);
    border: none;
    color:white;
    padding: 8px 16px;
    text-decoration: none;
    margin: 4px 2px;
    cursor:pointer;
    margin-bottom : 6px;
}

.vl {
    border-right: 1px solid green;
    height: 100%;
    top: 100px;
    position: absolute;
    left: 270px;
    margin-left: -3px;
}

```



```
}

.tree_style
{
    border : 1px solid blue;
    padding : 5px 5px;
    overflow: scroll;
    height: 500px;
    margin : 5px;
}

.col-50-border {
    border : 1px solid blue;
    float: left;
    width: 45%;
    margin-top: 2px;
    overflow: scroll;
    height: 870px;
    padding : 6px;
}

.col-20 {
    float: left;
    width: 140px;
    margin-top: 2px;
    padding : 2px;
    position:static;
}

.col-75 {
    float: left;
    width: 250px;
    margin-top: 2px;
    padding : 2px;
    position:static;
}

.status_div{
    right: 0;
    font-size: 14px;
    position: absolute;
    padding : 8px;
    margin-top : -50px;
}

span{
    padding : 8px;
    margin : 8px;
```

```

}

.span_border111{

width : 800px;
word-wrap: break-word;
word-break: break-all;

}

@media screen and (max-height: 450px){
.sidenav {padding-top: 15px;}
.sidenav a {font-size: 18px;}
}

hr {
position : absolute;
width: 100%;
left: 2px;
}

tr:nth-child(even) {background-color: #b4addf;}
th, td {

padding:1px;
}

table {
width: 100%;
}
</style> </head>
<body> 
<div id="s_div" class="status_div">
<span id="endpointurl"></span>
</div>
<h1 style="text-align: center; margin-top: -50px;"> ConnectOPC-UA </h1>

<div class="sidenav"><a href="index.html">Configure Machine</a><a href="connect.html">Connect
Machine</a> <a href="browse.html"><font color="#FF5733">Browse Address Space</font></a> <a
href="monitor.html">Monitor variables</a><a href="write.html">Write variables to the server</a><a
href="pubsub.html">Create
Pub-Sub/Notifications</a><a href="pubsub_mult.html">Multiple Pub-Sub/Notifications</a><a
href="method_call.html">Method Call (Address Space)</a><a href="alarm_monitoring.html">Event
Notifier</a> <a href="filetransfer.html">File Transfer</a><a href="opcuaserver.html">Create OPC-UA
Server</a><a href="settings.html">Settings</a><a href="help.html">Help</a></div>
<div class="main">
<h2>Browse OPC-UA Address Space</h2>
<h3>Read Current Node : </h3>
<input name="mlogs" class="input_text_class" id="mlogs"><input type = "button" value="Read"
id="read" onclick="myFunction();"><input type = "button" value="Add to Favourites"
id="add_favourite" onclick="add_favourite();">

<div class="span_border111">
<span id='monitor_value'></span>

```

```

    </div>
    <h3>ADDRESS SPACE</h3>
    <input type = "button" value="Download Address Space" id="read"
onclick="download_address_space();"><br>
    <div class="col-50-border">
        <div id="browse-tree"></div>
    </div>
    <div class="col-50-border">
        <table id=table_data>
    </div>

</div>

</div>

<script src="jtree-dist/jstree.min.js"></script>

<script>
    var process=false;
    const {ipcRenderer} = require('electron')
    let apppath
    let download_folder
    ipcRenderer.send("browse_path_req", "apppath")
    ipcRenderer.on('browse_path_rep', (event, arg) => {
        apppath=arg
        let fs2 = require('fs');
        const fileContents10 = fs2.readFileSync(apppath+'settings_data.txt', 'utf8')
        var p_nodeid;
        var datamr, datamr_mc

        try {

            const data = JSON.parse(fileContents10)
            //alert(JSON.stringify(data))
            p_nodeid=data.p_nodeid;
            download_folder=data.download_folder

        } catch(err) {
            console.error(err)
        }

        var data1=[{ "id" : p_nodeid, "parent" : "#", "text" : "Root" }]

        var node_sel
        // document.getElementById('mlogs').innerText="collecting logs"

```

```

$(function () {
    $.jstree.defaults.core.data=true;
    // $('#browse-tree').jstree();

    $('#browse-tree').on("changed.jstree", function (e, data) {
        node_sel=data.selected.toString();
        console.log("selected node id is : "+node_sel.toString()+" process is : "+process)
        if(!process)
        {
            console.log('Data Selected is : '+ data.selected.toString());
            ipcRenderer.send('node_crawler_data',data.selected.toString())
        }
    });

    $('#browse-tree').jstree({ 'core' : {
        'data' : data1,

        'themes': {
            'theme': 'default',
            'responsive':false
        }
    },

    // "plugins":["state", "search", "ui" ],

    });

    $('#browse-tree').on("refresh.jstree", function (e, data)
    {

        // $('#browse-tree').jstree("toggle_node",node_sel);
        $(this).jstree("open_all");
        process=false;
        //$('#tree').jstree("open_all",node_sel)
        //$(this).jstree(data.node);
    })

    });

    const fs1 = require('fs')

    const fileContents1 = fs1.readFileSync(apppath+'status_data.txt', 'utf8')
    try {

        const data1 = JSON.parse(fileContents1)
        //alert(JSON.stringify(data))

        document.getElementById('endpointurl').innerText=data1.endpointurl
        if(data1.status.toString()=="Connected")
        {

```

```

        document.getElementById('endpointurl').style.backgroundColor = "green";
        document.getElementById('endpointurl').style.color = "white";

    }
    else{
        document.getElementById('endpointurl').style.backgroundColor = "red";
        document.getElementById('endpointurl').style.color = "white";

    }
    } catch(err) {
    console.error(err)
    }
})

function add_favourite(){
    fs_favourite=require('fs')
    try {

        const fsmr = require('fs')
        const mrfileContents = fsmr.readFileSync(apppath+'opc_conf_data.txt', 'utf8')
        var datamr = JSON.parse(mrfileContents)
        var datamr_mc=datamr.machine_name
        if(document.getElementById("mlogs").value !=="" && document.getElementById("mlogs").value
        !=null)
        {
            var fav_match=false;
            const fsmr1 = require('fs')
            if (fsmr1.existsSync(apppath+datamr_mc+"_"+"favourites.txt")) {
                const mrfileContents1 = fsmr1.readFileSync(apppath+datamr_mc+"_"+"favourites.txt", 'utf8')
                var data_fav=mrfileContents1.split("\n")
                for(i=1;i<data_fav.length;i++)
                {
                    if(data_fav[i]==document.getElementById("mlogs").value)
                    {
                        fav_match=true
                        break;
                    }
                }
            }

            if(!fav_match)
            {
                favourite_write_stream=fs_favourite.createWriteStream(apppath+datamr_mc+"_"+"favourites.txt",
                {'flags':'a'});
                favourite_write_stream.write(document.getElementById("mlogs").value+"\n")
                favourite_write_stream.end()
                alert("Node Id : "+document.getElementById("mlogs").value+" is added to favourites")
            }
        }
    }
}

```

```

        else
        {
            alert('This Node Id is already added in Favorites')
        }
    }
    else{
        alert("Node Id is not available")
    }
}
catch(e) {
    alert('Failed to save the file! Error : '+e);
}
}

//const {ipcRenderer} = require('electron')

//alert("I am an alert box!");

function myFunction(){
    var variable_name=document.getElementById('mlogs').value
    ipcRenderer.send('monitor_data_value_node',variable_name)
}

function download_address_space(){
    // var variable_name=document.getElementById('mlogs').value
    ipcRenderer.send('download_address_space',"download_address_space")
}

ipcRenderer.on('node_attributes', (event, arg) => {

    data_rx_na=JSON.parse(arg)
    var table = document.getElementById("table_data");
    table.innerHTML = "";
    var row=table.insertRow(0)
    var cell=row.insertCell(0)
    cell.innerText="Node Id"
    var cell=row.insertCell(1)
    cell.innerText=data_rx_na.nd

    var row=table.insertRow(1)
    var cell=row.insertCell(0)
    cell.innerText="Description Name"
    var cell=row.insertCell(1)
    cell.innerText=data_rx_na.dn

    var row=table.insertRow(2)
    var cell=row.insertCell(0)
    cell.innerText="Browse Name"
    var cell=row.insertCell(1)
    cell.innerText=data_rx_na.bn

```

```
var row=table.insertRow(3)
var cell=row.insertCell(0)
cell.innerText="Description"
var cell=row.insertCell(1)
cell.innerText=data_rx_na.desc

var row=table.insertRow(4)
var cell=row.insertCell(0)
cell.innerText="Node Class"
var cell=row.insertCell(1)
cell.innerText=data_rx_na.nc

var row=table.insertRow(5)
var cell=row.insertCell(0)
cell.innerText="Event Notifier"
var cell=row.insertCell(1)
cell.innerText=data_rx_na.en

var row=table.insertRow(6)
var cell=row.insertCell(0)
cell.innerText="Data Type"
var cell=row.insertCell(1)
cell.innerText=data_rx_na.dt

var row=table.insertRow(7)
var cell=row.insertCell(0)
cell.innerText="Value"
var cell=row.insertCell(1)
cell.innerText=data_rx_na.v

var row=table.insertRow(8)
var cell=row.insertCell(0)
cell.innerText="Access Level"
var cell=row.insertCell(1)
cell.innerText=data_rx_na.al

var row=table.insertRow(9)
var cell=row.insertCell(0)
cell.innerText="Historizing"
var cell=row.insertCell(1)
cell.innerText=data_rx_na.his

var row=table.insertRow(10)
var cell=row.insertCell(0)
cell.innerText="User Access Level"
var cell=row.insertCell(1)
cell.innerText=data_rx_na.ual

var row=table.insertRow(11)
var cell=row.insertCell(0)
cell.innerText="Write Mask"
var cell=row.insertCell(1)
cell.innerText=data_rx_na.wm

var row=table.insertRow(12)
var cell=row.insertCell(0)
```

```

        cell.innerText="Value Rank"
        var cell=row.insertCell(1)
        cell.innerText=data_rx_na.vr

        var row=table.insertRow(13)
        var cell=row.insertCell(0)
        cell.innerText="Access Restriction"
        var cell=row.insertCell(1)
        cell.innerText=data_rx_na.ar

    })

ipcRenderer.on('nc_json_reply', (event, arg) => {
    $('#browse-tree').jstree(true).settings.core.data = arg;

    $('#browse-tree').jstree(true).refresh(true);
    process=true
    // $('#browse-tree').jstree.redraw(true);
    // $('#browse-tree').jstree("toggle_node",node_sel);
    // $('#browse-tree').jstree(true).refresh(true);
    // $('#browse-tree').jstree(true).open_node(node_sel);
    // $('#browse-tree').jstree(true)
})

ipcRenderer.on('browse_reply', (event, arg) => {
    //var old_text=document.getElementById('mlogs').value
    //var new_text=old_text+"\n"+JSON.stringify(arg)
    //document.getElementById('mlogs').innerText=JSON.stringify(arg)
})

ipcRenderer.on('nc_clear_reply', (event, arg) => {
    document.getElementById('mlogs').value=arg
})

ipcRenderer.on('browse_node_reply', (event, arg) => {
    document.getElementById('monitor_value').innerText=arg
})

</script>
</body>
</html>

```

8.5 Monitor/Read Variables

```

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=windows-1252">
<meta name="viewport" content="width=device-width, initial-scale=1">

```



```

<style>
  body {
    font-family: "Lato", sans-serif;
  }

  h1 {color:blue;
    position:absolute;
    left:80px}

  .sidenav{
width: 250px;
position:absolute;
z-index: 1;
top: 100px;
left: 1px;
background: #444444;
overflow-x: hidden;
padding: 1px 0;
height:100%;
}

.active,.sidenava {
padding: 6px 8px 6px 16px;
text-decoration: none;
font-size: 14px;
color: #FFFFFF;
display: block;
}

.sidenava: hover {
color: #C2C2C2;
}

.main {
margin-left: 255px; /* Same width as the sidebar + left position in px */
font-size: 14px; /* Increased text to enable scrolling */
padding: 0px 2px;
}

p {
color: #2196F3;
}

input[type="text"] {
width: 80%;
margin-bottom : 6px;
}

input[type="password"] {
width: 80%;
margin-bottom : 6px;
}

```

```

input[type="checkbox"] {
    margin-bottom : 6px;
}

input[type=button], input[type=submit], input[type=reset] {
    background-color:rgb(2,48, 4);
    border: none;
    color:white;
    padding:8px 16px;
    text-decoration: none;
    margin: 4px 2px;
    cursor:pointer;
    margin-bottom : 6px;
}

.vl {
    border-right: 1px solid green;
    height: 100%;
    top: 100px;
    position:absolute;
    left: 270px;
    margin-left: -3px;
}

@media screen and (max-height: 450px){
    .sidenav {padding-top: 15px;}
    .sidenav a {font-size: 18px;}
}

.status_div{
    right: 0;
    font-size: 14px;
    position: absolute;
    padding : 8px;
    margin-top : -50px;
}

span{
    padding : 8px;
    margin : 8px;
}

</style>

</head>
<body> 
<div id="s_div" class="status_div">
    <span id="endpointurl"></span>
</div>
<h1 style="text-align: center; margin-top: -50px;"> ConnectOPC-UA </h1>

```

```

<div class="sidenav"><a href="index.html">Configure Machine</a><a href="connect.html">Connect
Machine</a> <a href="browse.html">Browse Address Space</a> <a href="monitor.html"><font
color="#FF5733">Monitor variables</font></a> <a href="write.html">Write variables to the
server</a><a href="pubsub.html">Create
  Pub-Sub/Notifications</a><a href="pubsub_mult.html">Multiple Pub-Sub/Notifications</a><a
href="method_call.html">Method Call (Address Space)</a><a href="alarm_monitoring.html">Alarm
Monitoring</a> <a href="filetransfer.html">File Transfer</a> <a href="opcua_server.html">Create OPC-
UA Server</a><a href="settings.html">Settings</a><a href="help.html">Help</a></div>
<div class="main">
  <h2>Read Variables</h2>
  <form id="connect_form">
    Variable Name (Full Qualified Node Id)<br> <input name="variable_name" id="variable_name"
type="text"> <input type="button" value="Add From Favourite" id="aff" onclick="add_frm_fav1();">
<br>
    <input type="button" value="Read" id="read" onclick="myFunction();">
  </form>

  <h3>Logs</h3>
  <p id='mlogs'></p>
</div>

<script>
  const {ipcRenderer} = require('electron')
  ipcRenderer.send('monitor_path_req','apppath')
  ipcRenderer.on('monitor_path_rep', (event, arg) => {
    const fs1 = require('fs')

    const fileContents1 = fs1.readFileSync(arg+'status_data.txt', 'utf8')
    try {

      const data1 = JSON.parse(fileContents1)
      //alert(JSON.stringify(data1))

      document.getElementById('endpointurl').innerText=data1.endpointurl
      if(data1.status.toString()=="Connected")
      {
        document.getElementById('endpointurl').style.backgroundColor = "green";
        document.getElementById('endpointurl').style.color = "white";

      }
      else{
        document.getElementById('endpointurl').style.backgroundColor = "red";
        document.getElementById('endpointurl').style.color = "white";

      }
    } catch(err) {
      console.error(err)
    }

  })

  //document.getElementById('mlogs').innerText="collecting logs"
</script>

```

```

<script>

ipcRenderer.on('nc_clear_reply', (event, arg) => {
    document.getElementById('variable_name').value=arg
})

//alert("I am an alert box!");
function myFunction(){
    var variable_name=document.getElementById('variable_name').value
    ipcRenderer.send('monitor_data',variable_name)
}

function add_frm_fav1(){

    ipcRenderer.send('favourite_data',"start_favourite_data")
}

ipcRenderer.on('monitor1_reply', (event, arg) => {
var old_text=document.getElementById('mlogs').innerText
var new_text=arg+"\n"+old_text
document.getElementById('mlogs').innerText=new_text
})

</script>
</body>
</html>

```

8.6 Write Variables to the Server

```

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=windows-1252">
<meta name="viewport" content="width=device-width, initial-scale=1">

<style>
body {
    font-family: "Lato", sans-serif;
}

h1 {color:blue;
    position:absolute;
    left:80px}

.sidenav{
    width: 250px;
    position:absolute;
    z-index: 1;

```

```

    top: 100px;
    left: 10px;
    background: #eee;
    overflow-x: hidden;
    padding: 8px 0;
}

p {
    color: #2196F3;
}

.sidenav {
width: 250px;
position: absolute;
z-index: 1;
top: 100px;
left: 1px;
background: #444444;
overflow-x: hidden;
padding: 1px 0;
height: 100%;
}

.active, .sidenav a {
padding: 6px 8px 6px 16px;
text-decoration: none;
font-size: 14px;
color: #FFFFFF;
display: block;
}

.sidenav a:hover {
color: #C2C2C2;
}

.main {
margin-left: 255px; /* Same width as the sidebar + left position in px */
font-size: 14px; /* Increased text to enable scrolling */
padding: 0px 2px;
}

input[type="text"] {
width: 80%;
margin-bottom : 6px;
}

input[type="password"] {
width: 80%;
margin-bottom : 6px;
}

input[type="checkbox"] {
margin-bottom : 6px;
}

```

```

}

input[type=button], input[type=submit], input[type=reset] {
    background-color:rgb(2,48, 4);
    border: none;
    color:white;
    padding:8px 16px;
    text-decoration: none;
    margin: 4px 2px;
    cursor:pointer;
    margin-bottom : 6px;
}

.vl {
    border-right: 1px solid green;
    height: 100%;
    top: 100px;
    position:absolute;
    left: 270px;
    margin-left: -3px;
}

}

@media screen and (max-height: 450px){
    .sidenav {padding-top: 15px;}
    .sidenav a {font-size: 18px;}
}

.status_div{
    right: 0;
    font-size: 14px;
    position: absolute;
    padding : 8px;
    margin-top : -50px;
}

}

span{
    padding : 8px;
    margin : 8px;
}

</style>

</head>
<body> 
<div id="s_div" class="status_div">
    <span id="endpointurl"></span>
</div>
<h1 style="text-align: center; margin-top: -50px;"> ConnectOPC-UA </h1>

    <div class="sidenav"><a href="index.html">Configure Machine</a><a href="connect.html">Connect
Machine</a> <a href="browse.html">Browse Address Space</a>  <a href="monitor.html">Monitor

```

```

variables</a><a href="write.html"><font color="#FF5733">Write variables to the server</font></a><a
href="pubsub.html">Create
  Pub-Sub/Notifications</a><a href="pubsub_mult.html">Multiple Pub-Sub/Notifications</a><a
href="method_call.html">Method Call (Address Space)</a><a href="alarm_monitoring.html">Event
Notifier</a><a href="filetransfer.html">File Transfer</a><a href="opcuaserver.html">Create OPC-UA
Server</a><a href="settings.html">Settings</a><a href="help.html">Help</a></div>
<div class="main">
  <h2>Write Variables of Server</h2>
  <form id="connect_form">
    Variable Name (Full Qualified Node Id)<br> <input name="variable_name" id="variable_name"
type="text"> <input type="button" value="Add from Favourites" id="readaff"
onclick="add_frm_fav1();"> <br>
    Value<br> <input name="value_name" id="value_name" type="text"> <br>
    <input type="button" value="Write" id="read" onclick="myFunction();">
  </form>

  <h3>Logs</h3>
  <p id='mlogs'></p>
</div>

<script>
  const {ipcRenderer} = require('electron')
  ipcRenderer.send('monitor_path_req', 'apppath')
  ipcRenderer.on('monitor_path_rep', (event, arg) => {
    const fs1 = require('fs')

    const fileContents1 = fs1.readFileSync(arg+'status_data.txt', 'utf8')
    try {

      const data1 = JSON.parse(fileContents1)
      //alert(JSON.stringify(data))

      document.getElementById('endpointurl').innerText=data1.endpointurl
      if(data1.status.toString()=="Connected")
      {
        document.getElementById('endpointurl').style.backgroundColor = "green";
        document.getElementById('endpointurl').style.color = "white";

      }
      else{
        document.getElementById('endpointurl').style.backgroundColor = "red";
        document.getElementById('endpointurl').style.color = "white";

      }
    } catch(err) {
      console.error(err)
    }
  })

  ipcRenderer.on('nc_clear_reply', (event, arg) => {
    document.getElementById('variable_name').value=arg
  })

  function add_frm_fav1(){

```

```

        ipcRenderer.send('favourite_data',"start_favourite_data")
    }

    //document.getElementById('mlogs').innerText="collecting logs"
</script>

<script>

//alert("I am an alert box!");
function myFunction(){
    var variable_name=document.getElementById('variable_name').value
    var value_name=document.getElementById('value_name').value
    var data={"nodeid":variable_name,"value":value_name}
    ipcRenderer.send('write_data',JSON.stringify(data))
}

ipcRenderer.on('write_reply', (event, arg) => {
    var old_text=document.getElementById('mlogs').innerText
    var new_text=old_text+"\n"+arg
    document.getElementById('mlogs').innerText=new_text
})

</script>
</body>
</html>

```

8.7 Create Pub-Sub Notifications

```

<!DOCTYPE html>
<html>
<head>

<meta http-equiv="content-type" content="text/html; charset=windows-1252">
<meta name="viewport" content="width=device-width, initial-scale=1">

<style>
body {
    font-family: "Lato", sans-serif;
}

h1 {color:blue;
    position:absolute;
    left:80px}

```



```

    p {
        color: #2196F3;
    }

    .p_alarm {
        color: #ff0000;
    }

    .sidenav {
        width: 250px;
        position: absolute;
        z-index: 1;
        top: 100px;
        left: 1px;
        background: #444444;
        overflow-x: hidden;
        padding: 1px 0;
        height: 100%;
    }

    .active, .sidenav a {
        padding: 6px 8px 6px 16px;
        text-decoration: none;
        font-size: 14px;
        color: #FFFFFF;
        display: block;
    }

    .sidenav a:hover {
        color: #C2C2C2;
    }

    .main {
        margin-left: 255px; /* Same width as the sidebar + left position in px */
        font-size: 14px; /* Increased text to enable scrolling */
        padding: 0px 2px;
    }

    input[type="text"] {
        width: 80%;
        margin: 6px;
        padding: 6px;
    }

    input[type="password"] {
        width: 80%;
        margin-bottom: 6px;
    }

    input[type="checkbox"] {
        margin-bottom: 6px;
    }

```

```

input[type=button], input[type=submit], input[type=reset] {
    background-color: rgb(2,48, 4);;
    border: none;
    color: white;
    padding: 8px 16px;
    text-decoration: none;
    margin: 4px 2px;
    cursor: pointer;
    margin-bottom : 6px;
}

.vl {
    border-right: 1px solid green;
    height: 100%;
    top: 100px;
    position: absolute;
    left: 270px;
    margin-left: -3px;
}

}

@media screen and (max-height: 450px){
    .sidenav {padding-top: 15px;}
    .sidenav a {font-size: 18px;}
}

.status_div{
    right: 0;
    font-size: 14px;
    position: absolute;
    padding : 8px;
    margin-top : -50px;
}

}

span{
    padding : 8px;
    margin : 8px;
}

</style>

</head>
<body> 
<div id="s_div" class="status_div">
    <span id="endpointurl"></span>
</div>
<h1 style="text-align: center; margin-top: -50px;"> ConnectOPC-UA </h1>

    <div class="sidenav"><a href="index.html">Configure Machine</a><a href="connect.html">Connect
Machine</a> <a href="browse.html">Browse Address Space</a>  <a href="monitor.html">Monitor
variables</a><a href="write.html">Write variables to the server</a><a href="pubsub.html"><font
color="#FF5733">Create

```

```

    Pub-Sub/Notifications</font></a> <a href="pubsub_mult.html">Multiple Pub-
Sub/Notifications</a> <a href="method_call.html">Method Call (Address Space)</a> <a
href="alarm_monitoring.html">Event Notifier</a> <a href="filetransfer.html">File Transfer</a> <a
href="opcuaserver.html">Create OPC-UA Server</a> <a href="settings.html">Settings</a> <a
href="help.html">Help</a></div>
<div class="main">
    <h2>Subscribe Variables</h2><br>
    <p class="p_alarm">Note : Only 5 signals can be simultaneously monitored with charts, for
monitoring signals more than 5 signals, click below link</p>
    <a href="pubsub_mult.html">Multiple Signal Subscriptions</a><br><br><br>
    <input type = "button" value="Start Subscription" id="subscribe" onclick="subscribe();">
    <input type = "button" value="Stop Subscription" id="stop_subscribe" onclick="myFunction1();">

    <br><br>
    <h3>Add Subscription</h3>
    <form id="connect_form">
        Variable Name (Node Id)<br> <input name="variable_name" id="variable_name" type="text">
<input type = "button" value="Add From Favourites" id="add_from_favourite"
onclick="add_from_favourite1();"><br>
        Signal Name (Identifier)<br> <input name="var_identifier_name" id="var_identifier_name"
type="text"> <br>
        <input type = "button" value="Add Subscription" id="subscribe" onclick="myFunction();">
        <input type = "button" value="Download Data" id="download" onclick="download_data();">
    </form>
    <h3>Logs</h3>
    <p id='mlogs'></p>
    <div id="chart" style="height: 370px; width: 100%;"></div>
    <div id="chart2" style="height: 370px; width: 100%;"></div>
    <div id="chart3" style="height: 370px; width: 100%;"></div>
    <div id="chart4" style="height: 370px; width: 100%;"></div>
    <div id="chart5" style="height: 370px; width: 100%;"></div>

</div>
<script>
const {ipcRenderer} = require('electron')
var CanvasJS = require('./canvasjs.min.js');
var $ =require('jquery')
var chart,chart2,chart3,chart4,chart5;
var datapoints=[];
var datapoints2=[];
var datapoints3=[];
var datapoints4=[];
var datapoints5=[];
var count=1;

ipcRenderer.send('enquire_subscribed_data','fetch_subscriptions')
ipcRenderer.on('enquire_subscribed_data_reply',(event, arg) => {
    fetch_subscription(arg);
    //alert(arg)
})

function add_from_favourite1()
{
    ipcRenderer.send('favourite_data','start_favourite_data')
    // alert('favourite called')
}

```

```

ipcRenderer.on('nc_clear_reply', (event, arg) => {
    document.getElementById('variable_name').value=arg
})

function subscribe()
{
    data={"type":"subscription","data":variable_name,"signal_identifier":"start_subscription"}
    ipcRenderer.send('subscribe_data',JSON.stringify(data))
}

function fetch_subscription(data)
{
    for (var i in data)
    {
        var var_identifier_name=data[i].signal_identifier
        ipcRenderer.send('subscription_start_data',JSON.stringify(data[i]))
        //ipcRenderer.send('subscription_start_data',JSON.stringify(data))

        if(data[i].count<=5)
        {
            if(count==1)
            {

                chart= new CanvasJS.Chart("chart",{
                    theme: "light", // "light2", "dark1", "dark2"
                    animationEnabled:true, // change to false
                    title:{
                        text: var_identifier_name
                    },
                    data: [
                        {
                            type: "line",
                            xValueType: "dateTime",
                            showInLegend: true,
                            name: var_identifier_name,
                            dataPoints: datapoints
                        }
                    ],

                });
                // chart.render();
            }

            else if(count==2)
            {
                //var new_chart2=document.createElement('div');
                //new_chart2.setAttribute('id',variable_name);
                //document.body.appendChild(new_chart2);

                chart2 = new CanvasJS.Chart("chart2",{
                    theme: "light", // "light2", "dark1", "dark2"

```

```

        animationEnabled:true, // change to false
        title:{
            text: var_identifier_name
        },
        data: [
            {
type: "line",
xValueType: "dateTime",
                showInLegend: true,
                name: var_identifier_name,
                dataPoints: datapoints2
            }
        ]
    });
    // chart2.render();
}

else if(count==3)
{

    //var new_chart3=document.createElement('div');
    //new_chart3.setAttribute('id',variable_name);
    //document.body.appendChild(new_chart3);
    chart3 = new CanvasJS.Chart("chart3",{
        theme: "light", // "light2", "dark1", "dark2"
        animationEnabled:true, // change to false
        title:{
            text: var_identifier_name
        },
        data: [
            {
type: "line",
xValueType: "dateTime",
                showInLegend: true,
                name: var_identifier_name,
                dataPoints: datapoints3
            }
        ]
    });
    // chart3.render();
}

else if(count==4)
{

    //var new_chart4=document.createElement('div');
    //new_chart4.setAttribute('id',variable_name);
    //document.body.appendChild(new_chart4);
    chart4 = new CanvasJS.Chart("chart4",{
        theme: "light", // "light2", "dark1", "dark2"
        animationEnabled:true, // change to false
        title:{
            text: var_identifier_name
        },
        data: [
            {

```

```

        type: "line",
        xValueType: "dateTime",
            showInLegend: true,
            name: var_identifier_name,
            dataPoints: datapoints4
        }
    ]
});
// chart4.render();
}

else if(count==5)
{

    //var new_chart5=document.createElement('div');
    //new_chart5.setAttribute('id',variable_name);
    //document.body.appendChild(new_chart5);
    chart5 = new CanvasJS.Chart("chart5",{
        theme: "light", // "light2", "dark1", "dark2"
        animationEnabled: true, // change to false
        title:{
            text: var_identifier_name
        },
        data: [
            {
                type: "line",
                xValueType: "dateTime",
                    showInLegend: true,
                    name: var_identifier_name,
                    dataPoints: datapoints5
                }
            ]
        });
    //chart5.render();
}

count++
}

else{
    count=1;
}

}
}
/* window.onload = function () {

    <div id="chartContainer" style="height: 370px; width: 100%;"></div>

```

```

chart= new CanvasJS.Chart("chartContainer",{
    theme: "light", // "light2", "dark1", "dark2"
    animationEnabled:true, // change to false
    title:{
        text: "Real Time Subscribed Node Value"
    },
    data: [
        {
            type: "line",
            xValueType: "dateTime",
            showInLegend: true,
            name: "Real Time Node ID Variable data",
            dataPoints: datapoints
        }
    ]
});
chart.render();
}*/
//var updateInterval = 2000;

ipcRenderer.on('subscribed_value', (event, arg) => {
    var data=JSON.parse(arg)
    var count_rx=data.count
    if(count_rx==0)
    {
        count=1;
    }
    console.log("count is "+count_rx)
    console.log("Data X : "+data.x)
    time=data.x
    console.log("time : "+time)
    var d = new Date(data.x);
    console.log("date conversion is : "+d);

    if(count_rx==1)
    {
        datapoints.push({
            x: d,
            y: data.y
        });
        chart.render();
        //chart.update();
    }

    else if(count_rx==2)
    {
        datapoints2.push({
            x: d,
            y: data.y
        });

        chart2.render();
        //chart2.update();
    }
    else if(count_rx==3)

```

```

    {
      datapoints3.push({
        x: d,
        y: data.y
      });
      chart3.render();
      //chart3.update();
    }
    else if(count_rx==4)
    {
      datapoints4.push({
        x: d,
        y: data.y
      });
      chart4.render();
      //chart4.update();
    }
    else if(count_rx==5)
    {
      datapoints5.push({
        x: d,
        y: data.y
      });
      chart5.render();
      //chart5.update();
    }
  })

</script>

<script>
  document.getElementById('var_identifier_name').value="Signal Name"
  ipcRenderer.send('pubsub_path_req','apppath')

  ipcRenderer.on('pubsub_path_rep', (event, arg) => {
    const fs1 = require('fs')

    const fileContents1 = fs1.readFileSync(arg+'status_data.txt', 'utf8')
    try {

      const data1 = JSON.parse(fileContents1)
      //alert(JSON.stringify(data))

      document.getElementById('endpointurl').innerText=data1.endpointurl
      if(data1.status.toString()=="Connected")
      {
        document.getElementById('endpointurl').style.backgroundColor = "green";
        document.getElementById('endpointurl').style.color = "white";

      }
      else{
        document.getElementById('endpointurl').style.backgroundColor = "red";
        document.getElementById('endpointurl').style.color = "white";
      }
    }
  })

```



```

    }
    } catch(err) {
    console.error(err)
    }

    })

    //document.getElementById('mlogs').innerText="collecting logs"
</script>

<script>

function chart1_download()
{
    alert('ready for download')
    var img1=chart.toBase64Image()
}

//alert("I am an alert box!");
function myFunction(){
    if(count<=5)
    {
        var variable_name=document.getElementById('variable_name').value
        var var_identifier_name=document.getElementById('var_identifier_name').value+" (
"+document.getElementById('variable_name').value+" )"

data={"type":"subscription","data":variable_name,"signal_identifier":var_identifier_name,"count":count}

        ipcRenderer.send('subscription_start_data',JSON.stringify(data))

        if(count==1)
        {

            chart= new CanvasJS.Chart("chart",{
                theme: "light", // "light2", "dark1", "dark2"
                animationEnabled:true, // change to false
                title:{
                    text: var_identifier_name
                },
                data: [
                    {
                        type: "line",
                        xValueType: "dateTime",
                        showInLegend: true,
                        name: var_identifier_name,
                        dataPoints: datapoints
                    }
                ]
            });
        }
    }
}

```

```

    }
    ],

    });
    //chart.render();
}

else if(count==2)
{
    //var new_chart2=document.createElement('div');
    //new_chart2.setAttribute('id',variable_name);
    //document.body.appendChild(new_chart2);

    chart2 = new CanvasJS.Chart("chart2",{
        theme: "light", // "light2", "dark1", "dark2"
        animationEnabled:true, // change to false
        title:{
            text: var_identifier_name
        },
        data: [
            {
                type: "line",
                xValueType: "dateTime",
                showInLegend: true,
                name: var_identifier_name,
                dataPoints: datapoints2
            }
        ]
    });
    //chart2.render();
}

else if(count==3)
{
    //var new_chart3=document.createElement('div');
    //new_chart3.setAttribute('id',variable_name);
    //document.body.appendChild(new_chart3);
    chart3 = new CanvasJS.Chart("chart3",{
        theme: "light", // "light2", "dark1", "dark2"
        animationEnabled:true, // change to false
        title:{
            text: var_identifier_name
        },
        data: [
            {
                type: "line",
                xValueType: "dateTime",
                showInLegend: true,
                name: var_identifier_name,
                dataPoints: datapoints3
            }
        ]
    });
    //chart3.render();
}

```

```

else if(count==4)
{
//var new_chart4=document.createElement('div');
//new_chart4.setAttribute('id',variable_name);
//document.body.appendChild(new_chart4);
chart4 = new CanvasJS.Chart("chart4",{
    theme: "light", // "light2", "dark1", "dark2"
    animationEnabled:true, // change to false
    title:{
        text: var_identifier_name
    },
    data: [
        {
type: "line",
xValueType: "dateTime",
            showInLegend: true,
            name: var_identifier_name,
            dataPoints: datapoints4
        }
    ]
});
//chart4.render();
}

else if(count==5)
{
//var new_chart5=document.createElement('div');
//new_chart5.setAttribute('id',variable_name);
//document.body.appendChild(new_chart5);
chart5 = new CanvasJS.Chart("chart5",{
    theme: "light", // "light2", "dark1", "dark2"
    animationEnabled:true, // change to false
    title:{
        text: var_identifier_name
    },
    data: [
        {
type: "line",
xValueType: "dateTime",
            showInLegend: true,
            name: var_identifier_name,
            dataPoints: datapoints5
        }
    ]
});
//chart5.render();
}

count++
}
else{
    alert('Due to memory constraints only 5 Signals can be monitored with chart. In order to monitor
multiple signals please use multiple subscription screen')
}

```

```

    }

    function myFunction1(){
        try{
            count=1

        }catch(e)
        {
            alert('Error : '+e)
        }
        data={"type":"stop_subscription","data":"Empty data"}
        ipcRenderer.send('subscribe_data',JSON.stringify(data))
        window.location.reload();
    }

    function download_data(){

        ipcRenderer.send('download_subscribed_data',"download")

    }

    ipcRenderer.on('subscribe_reply',(event, arg)=> {
        var old_text=document.getElementById('mlogs').innerText
        var new_text=arg+"\n"+old_text
        document.getElementById('mlogs').innerText=new_text
    })

</script>
</body>
</html>

```

8.8 Multiple Pub-Sub Notifications

```

<!DOCTYPE html>
<html>
<head>

    <meta http-equiv="content-type" content="text/html; charset=windows-1252">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="jtree-dist/themes/default/style.min.css" />
    <script>
    $=require('jquery')
    </script>

    <style>
    body {
        font-family: "Lato", sans-serif;
    }

```

```

    }

    h1 {color:blue;
        position:absolute;
        left:80px}

    p {
        color:#2196F3;
    }

    .p_alarm{
        color:#ff0000;
    }

    .sidenav{
width: 250px;
position:absolute;
z-index: 1;
top: 100px;
left: 1px;
background: #444444;
overflow-x: hidden;
padding: 1px 0;
height:100%;
    }

    .active,.sidenava {
padding: 6px 8px 6px 16px;
text-decoration: none;
font-size: 14px;
color: #FFFFFF;
display: block;
    }

    .sidenava: hover {
color: #C2C2C2;
    }

    .main{
margin-left: 255px; /* Same width as the sidebar + left position in px */
font-size: 14px; /* Increased text to enable scrolling */
padding: 0px 2px;
    }

    input[type="text"] {
        width: 80%;
        margin : 6px;
        padding : 6px;
    }

```

```

input[type="password"] {
    width: 80%;
    margin-bottom : 6px;
}

input[type="checkbox"] {
    margin-bottom : 6px;
}

input[type=button], input[type=submit], input[type=reset] {
    background-color: rgb(2, 48, 4);;
    border: none;
    color: white;
    padding: 8px 8px;
    text-decoration: none;
    margin: 2px 2px;
    cursor: pointer;
    margin-bottom : 3px;
}

.vl {
    border-right: 1px solid green;
    height: 100%;
    top: 100px;
    position: absolute;
    left: 270px;
    margin-left: -3px;
}

}

@media screen and (max-height: 450px){
    .sidenav {padding-top: 15px;}
    .sidenav a {font-size: 18px;}
}

.status_div{
    right: 0;
    font-size: 14px;
    position: absolute;
    padding: 8px;
    margin-top : -50px;
}

}

span{
    padding : 8px;
    margin : 8px;
}

.col-25 {
    float: left;
    width: 25%;
    margin-top: 2px;
}

}

```

```

/* Floating column for inputs: 75% width */
.col-75 {
    float: left;
    width: 75%;
    margin-top: 2px;
}

.col-50 {
    float: left;
    width: 50%;
    margin-top: 2px;
}

.col-45 {
    float: left;
    width: 48%;
    margin-top: 2px;
    height : 300px;
}

.col-50-border {
    border : 1px solid blue;
    float: left;
    width: 50%;
    margin-top: 2px;
    overflow: scroll;
    height: 870px;
    padding : 6px;
}

.col-border {
    border : 1px solid blue;
    width : 90%;
    float: left;
    margin: 10px;
    overflow: scroll;
    height: 575px;
    padding : 6px;
}

.log_style
{
    border : 1px solid blue;
    overflow: scroll;
    float: left;
    width: 50%;
    margin-top: 2px;
}

tr:nth-child(even) {background-color: #c2c2c2;}
th, td {
border-bottom: 1px solid #ddd;

```

```

padding:10px;
}

table {
    width: 100%;
}

</style>

</head>
<body> 
<div id="s_div" class="status_div">
    <span id="endpointurl"></span>
</div>
<h1 style="text-align: center; margin-top: -50px;"> ConnectOPC-UA </h1>

<div class="sidenav"><a href="index.html">Configure Machine</a><a href="connect.html">Connect
Machine</a> <a href="browse.html">Browse Address Space</a> <a href="monitor.html">Monitor
variables</a><a href="write.html">Write variables to the server</a><a href="pubsub.html">Create
Pub-Sub/Notifications</a><a href="pubsub_mult.html"><font color="#FF5733">Multiple Pub-
Sub/Notifications</font><a href="method_call.html">Method Call (Address Space)</a><a
href="alarm_monitoring.html">Event Notifier</a><a href="filetransfer.html">File Transfer</a> </a><a
href="opcuaserver.html">Create OPC-UA Server</a><a href="settings.html">Settings</a><a
href="help.html">Help</a></div>
<div class="main">
    <h2>Subscribe Variables</h2>
    <div class="col-45">

        <input type = "button" value="Start Subscription" id="subscribe" onclick="subscribe();">
        <input type = "button" value="Stop Subscription" id="stop_subscribe" onclick="myFunction1();">

        <h3>Add Subscription</h3>
        <form id="connect_form" >
            Variable Name (Full Qualified Node Id)<br> <input name="variable_name" id="variable_name"
            type="text"> <br>
            Signal Name (Identifier)<br> <input name="var_identifier_name" id="var_identifier_name"
            type="text"> <br>
            <input type = "button" value="Add Subscription" id="subscribe" onclick="myFunction();">
            <input type = "button" value="Download Data" id="download" onclick="download_data();">
            <input type = "button" value="Add From Favourites" id="add_from_favourite"
            onclick="add_from_favourite1();">
        </form>
        <p id="mlogs"></p>
        <h3>ADDRESS SPACE</h3>
        <div id="browse-tree" class="col-border"></div>
        </div>
        <div class="col-50-border">
            <h3>Subscription Results</h3>
            <table id=table_data>

            </table>

        </div>

```



```

</div>

<script src="jtree-dist/jstree.min.js"></script>
<script>
var process=false;
const {ipcRenderer} = require('electron')
var $ =require('jquery')
var table = document.getElementById("table_data");
var row=table.insertRow(-1)
var cell1 = row.insertCell(0);
var cell2 = row.insertCell(1);
var cell3 = row.insertCell(2);
var cell4 = row.insertCell(3);
var apppath,nodeid;
cell1.innerHTML="Signal Name"
cell2.innerHTML="Node Id"
cell3.innerHTML="Value"
cell4.innerHTML="Server Timestamp"

var count=1;

ipcRenderer.send('enquire_subscribed_data','fetch_subscriptions')
ipcRenderer.on('enquire_subscribed_data_reply',(event, arg) => {
    //fetch_subscription(arg);
    //alert(arg)
})

ipcRenderer.send('enquire_subscribed_data','fetch_subscriptions')
ipcRenderer.on('enquire_subscribed_data_reply',(event, arg) => {
    //fetch_subscription(arg);
    //alert(arg)
})

document.getElementById('var_identifier_name').value="Signal Name"
ipcRenderer.send('pubsub_path_req',apppath)

ipcRenderer.on('pubsub_path_rep', (event, arg) => {
    const fs1 = require('fs')
    apppath=arg
    const fileContents1 = fs1.readFileSync(arg+'status_data.txt', 'utf8')
    try {

        const data1 = JSON.parse(fileContents1)
        //alert(JSON.stringify(data))

        document.getElementById('endpointurl').innerText=data1.endpointurl
        if(data1.status.toString()=="Connected")
        {
            document.getElementById('endpointurl').style.backgroundColor = "green";
            document.getElementById('endpointurl').style.color = "white";

        }

    }
}

```

```

        else{
            document.getElementById('endpointurl').style.backgroundColor = "red";
            document.getElementById('endpointurl').style.color = "white";
        }

        let fs2 = require('fs');
const fileContents10 = fs2.readFileSync(apppath+'settings_data.txt', 'utf8')
var p_nodeid;

try {

    const data = JSON.parse(fileContents10)
    //alert(JSON.stringify(data))
    p_nodeid=data.p_nodeid;

} catch(err) {
    console.error(err)
}

var data2=[{ "id" : p_nodeid, "parent" : "#", "text" : "Root" }]

var node_sel
// document.getElementById('mlogs').innerText="collecting logs"

$(function () {
    $.jstree.defaults.core.data=true;
    // $('#browse-tree').jstree();

    $('#browse-tree').on("changed.jstree", function (e, data){
        node_sel=data.selected.toString();
        console.log("selected node id is : "+node_sel.toString()+" process is : "+process)
        if(!process)
        {
            console.log('Data Selected is : '+ data.selected.toString());
            ipcRenderer.send('node_crawler_data',data.selected.toString())
        }

    });

    $('#browse-tree').jstree({ 'core' : {
        'data' : data2,

        'themes': {
            'theme': 'default',
            'responsive':false
        }
    },

    'plugins':[ "state", "search", "wholerow", "ui" ],

});

```

```

        $('#browse-tree').bind("refresh.jstree", function(e, data)
        {
            // $('#browse-tree').jstree("toggle_node",node_sel);
            $(this).jstree("open_all");
            process=false;
            //$('#tree').jstree("open_all",node_sel)
            //$(this).jstree(data.node);
        })

    });

    } catch(err) {
    console.error(err)
    }

    })

    ipcRenderer.on('nc_json_reply', (event, arg) => {
        $('#browse-tree').jstree(true).settings.core.data = arg;

        $('#browse-tree').jstree(true).refresh(true);
        process=true

    })

    ipcRenderer.on('nc_clear_reply', (event, arg) => {
        document.getElementById('variable_name').value=arg
    })

    function add_from_favourite1()
    {
        ipcRenderer.send('favourite_data',"start_favourite_data")
        // alert('favourite called')
    }

    function subscribe()
    {
        data={"type":"subscription","data":{"variable_name","signal_identifier":"start_subscription"}}
        ipcRenderer.send('subscribe_data',JSON.stringify(data))
    }

    function fetch_subscription(data)
    {
        for (var i in data)
        {
            var var_identifier_name=data[i].signal_identifier

```

```

ipcRenderer.send('subscription_start_data',JSON.stringify(data[i]))
//ipcRenderer.send('subscription_start_data',JSON.stringify(data))

}
}
/* window.onload = function () {

<div id="chartContainer" style="height: 370px; width: 100%;"></div>

chart= new CanvasJS.Chart("chartContainer",{
  theme: "light", // "light2", "dark1", "dark2"
  animationEnabled: true, // change to false
  title:{
    text: "Real Time Subscribed Node Value"
  },
  data: [
    {
type: "line",
xValueType: "dateTime",
      showInLegend: true,
      name: "Real Time Node ID Variable data",
      dataPoints: datapoints
    }
  ]
});
chart.render();
}*/
//var updateInterval = 2000;

ipcRenderer.on('subscribed_value',(event, arg) => {
  var data=JSON.parse(arg)
  var count_rx=data.count
  if(count_rx==0)
  {
    count_rx=count-1;
    count--;
  }
  console.log("Count Rx : ",count_rx)
  console.log("Table row length : ",table.rows.length)
  table.rows[count_rx].cells[0].innerHTML=data.signal_name.split("(")[0];
  table.rows[count_rx].cells[1].innerHTML=data.node_id;
  table.rows[count_rx].cells[2].innerHTML=data.y;
  table.rows[count_rx].cells[3].innerHTML=data.x;
  // Insert new cells (<td> elements) at the 1st and 2nd position of the "new" <tr> element:

})

```

```

/* document.getElementById('var_identifier_name').value="Signal Name"
ipcRenderer.send('pubsub_path_req','apppath')

ipcRenderer.on('pubsub_path_rep', (event, arg) => {
  const fs1 = require('fs')

  const fileContents1 = fs1.readFileSync(arg+'status_data.txt', 'utf8')
  try {

    const data1 = JSON.parse(fileContents1)
    //alert(JSON.stringify(data))

    document.getElementById('endpointurl').innerText=data1.endpointurl
    if(data1.status.toString()=="Connected")
    {
      document.getElementById('endpointurl').style.backgroundColor = "green";
      document.getElementById('endpointurl').style.color = "white";

    }
    else{
      document.getElementById('endpointurl').style.backgroundColor = "red";
      document.getElementById('endpointurl').style.color = "white";

    }
  } catch(err) {
    console.error(err)
  }

})*/

//document.getElementById('mlogs').innerText="collecting logs"

//alert("I am an alert box!");
function myFunction(){

  var row=table.insertRow(-1)
  var cell1 = row.insertCell(0);
  var cell2 = row.insertCell(1);
  var cell3 = row.insertCell(2);
  var cell4 = row.insertCell(3);
  var variable_name=document.getElementById('variable_name').value
  var var_identifier_name=document.getElementById('var_identifier_name').value+" (
"+document.getElementById('variable_name').value+" )"

  data={"type":"subscription","data":variable_name,"signal_identifier":var_identifier_name,"count":count}

```

```

        ipcRenderer.send('subscription_start_data',JSON.stringify(data))
        count++;

    }

    function myFunction1(){

        data={"type":"stop_subscription","data":"Empty data"}
        ipcRenderer.send('subscribe_data',JSON.stringify(data))
        window.location.reload();

    }

    function download_data(){

        ipcRenderer.send('download_subscribed_data',"download")

    }

    ipcRenderer.on('subscribe_reply',(event, arg)=> {
    var old_text=document.getElementById('mlogs').innerText
    var new_text=arg+"\n"+old_text
    document.getElementById('mlogs').innerText=new_text
    })

</script>
</body>
</html>

```

8.9 Method Call

```

<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=windows-1252">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="jtree-dist/themes/default/style.min.css" />
  <script>
    $=require('jquery')
  </script>
  <style>
body {
  font-family: "Lato", sans-serif;
}

h1 {color:blue;
  position:absolute;
  left:80px}

```

```

.sidenav{
width: 250px;
position: absolute;
z-index: 1;
top: 100px;
left: 1px;
background: #444444;
overflow-x: hidden;
padding: 1px 0;
height: 100%;
}

.active,.sidenav a {
padding: 6px 8px 6px 16px;
text-decoration: none;
font-size: 14px;
color: #FFFFFF;
display: block;
}

.sidenav a: hover {
color: #C2C2C2;
}

.main {
margin-left: 255px; /* Same width as the sidebar + left position in px */
font-size: 14px; /* Increased text to enable scrolling */
padding: 0px 2px;
}

p {
color: #2196F3;
}

.container {
border-radius: 5px;
padding: 20px;
}

/* Floating column for labels: 25% width */
.col-25 {
float: left;
width: 25%;
margin-top: 2px;
}

/* Floating column for inputs: 75% width */
.col-75 {
float: left;
width: 75%;
margin-top: 2px;
}

span{

```

```

padding: 6px 6px 1px 6px;
}

input[type="text"] {
width: 80%;
margin-bottom : 6px;
}

input[type="password"] {
width: 80%;
margin-bottom : 6px;
}

input[type="checkbox"] {
margin-bottom : 6px;
}

input[type=button], input[type=submit], input[type=reset] {
background-color: #4CAF50;
border: none;
color: white;
padding: 8px 16px;
text-decoration: none;
margin: 4px 2px;
cursor: pointer;
margin-bottom : 6px;
}

.vl {
border-right: 1px solid green;
height: 100%;
top: 100px;
position: absolute;
left: 270px;
margin-left: -3px;
}

.tree_style
{
border : 1px solid blue;
padding : 5px 5px;
overflow: scroll;
height: 500px;
margin : 5px;
}

.status_div{
right: 0;
font-size: 14px;
position: absolute;

```



```

padding : 8px;
margin-top : -50px;

}

span{
padding : 8px;
margin : 8px;
}

.span_border111{

width : 800px;
word-wrap: break-word;
word-break: break-all;

}

@media screen and (max-height: 450px) {
.sidenav {padding-top: 15px;}
.sidenav a {font-size: 18px;}
}

hr {
position : absolute;
width: 100%;
left: 2px;
}
</style> </head>
<body> 
<div id="s_div" class="status_div">
<span id="endpointurl"></span>
</div>
<h1 style="text-align: center; margin-top: -50px;"> ConnectOPC-UA </h1>

<div class="sidenav"><a href="index.html">Configure Machine</a><a href="connect.html">Connect
Machine</a> <a href="browse.html">Browse Address Space</a> <a href="monitor.html">Monitor
variables</a><a href="write.html">Write variables to the server</a><a href="pubsub.html">Create
Pub-Sub/Notifications</a><a href="pubsub_mult.html">Multiple Pub-Sub/Notifications</a><a
href="method_call.html"><font color="#FF5733">Method Call (Address Space)</font></a><a
href="alarm_monitoring.html">Event Notifier</a><a href="filetransfer.html">File Transfer</a> <a
href="opcuaserver.html">Create OPC-UA Server</a><a href="settings.html">Settings</a><a
href="help.html">Help</a></div>
<div class="main">
<h2>Method Calls in Address Space</h2>
<pre>Current NodeId : <span id='mlogs'></span></pre>
<div class="container">

<div class="row">
<div class="col-25"><label for="fname">Object NodeId</label> </div>
<div class="col-75"><input name="object_nodeid" id="object_nodeid" type="text"></div>
</div>

<div class="row">
<div class="col-25"><label for="fname">Method NodeId</label> </div>

```

```

        <div class="col-75"><input name="method_nodeid" id="method_nodeid"
type="text"></div> </div>

        <div class="row">
            <div class="col-25"><label for="fname">Input Arguments 1</label></div>
            <div class="col-75"><input name="value_write1" id="value_write1" type="text"></div>
        </div>

        <div class="row">
            <div class="col-25"><label for="fname">Datatype for Input Argument 1</label></div>
            <div class="col-75"><select id="data_type1">
                <option value="double">Double</option>
                <option value="float">Float</option>
                <option value="string">String</option>
                <option value="uint16">UInt16</option>
                <option value="uint32">UInt32</option>
                <option value="uint64">UInt64</option>
                <option value="int16">Int16</option>
                <option value="int32">Int32</option>
                <option value="int64">Int64</option>
            </select></div> </div>

            <div class="row">
                <div class="col-25"><label for="fname">ArrayType for Input Argument 1</label>
        </div>

            <div class="col-75"><select id="array_type1">
                <option value="scalar">Scalar</option>
                <option value="array">Array</option>
            </select></div> </div>

            <div class="row">
                <div class="col-25"><label for="fname">Input Arguments 2</label></div>
                <div class="col-75"><input name="value_write2" id="value_write2"
type="text"></div> </div>

            <div class="row">
                <div class="col-25"><label for="fname">Datatype for Input Argument 2</label>
        </div>

            <div class="col-75"><select id="data_type2">
                <option value="double">Double</option>
                <option value="float">Float</option>
                <option value="string">String</option>
                <option value="uint16">UInt16</option>
                <option value="uint32">UInt32</option>
                <option value="uint64">UInt64</option>
                <option value="int16">Int16</option>
                <option value="int32">Int32</option>
                <option value="int64">Int64</option>
                <option value="None">None</option>
            </select></div> </div>

            <div class="row">
                <div class="col-25"><label for="fname">ArrayType for Input Argument
2</label></div>

                <div class="col-75"><select id="array_type2">
                    <option value="scalar">Scalar</option>

```

```

        <option value="array">Array</option>
        <option value="None">None</option>
    </select></div> </div>

</div>

<input type = "button" value="Call Method Write" id="write" onclick="myFunction();">
</div>
<div class="container">
<div class="span_border111">
    <span id='monitor_value'></span>

</div>
<h3>ADDRESS SPACE</h3>
<input type = "button" value="Download Address Space" id="read"
onclick="download_address_space();">
<div class="tree_style">
    <div id="browse-tree"></div>
</div>

</div>

<script src="jtree-dist/jstree.min.js"></script>

<script>
    const {ipcRenderer} = require('electron')
    let apppath
    var process=false;
    ipcRenderer.send("browse_path_req", "apppath")
    ipcRenderer.on('browse_path_rep', (event, arg) => {
        apppath=arg
        let fs2 = require('fs');
        const fileContents10 = fs2.readFileSync(apppath+'settings_data.txt', 'utf8')
        var p_nodeid;

        try {

            const data = JSON.parse(fileContents10)
            //alert(JSON.stringify(data))
            p_nodeid=data.p_nodeid;

        } catch(err) {
            console.error(err)
        }

        var data1=[{ "id" : p_nodeid, "parent" : "#", "text" : "Root" }]

        var node_sel
        // document.getElementById('mlogs').innerText="collecting logs"

        $(function () {
            $.jstree.defaults.core.data=true;

```

```

// $('#browse-tree').jstree();

$('#browse-tree').on("changed.jstree", function (e, data) {
    node_sel=data.selected.toString();
    console.log("selected node id is : "+node_sel.toString()+" process is : "+process)
    if(!process)
    {
        console.log('Data Selected is : '+ data.selected.toString());
        ipcRenderer.send('node_crawler_data',data.selected.toString())
    }

});

$('#browse-tree').jstree({ 'core' : {
    'data' : data1,

    'themes': {
        'theme': 'default',
        'responsive': false
    }
},

'plugins':[ "state", "search", "wholerow", "ui" ],

});

$('#browse-tree').bind("refresh.jstree", function (e, data)
{
    // $('#browse-tree').jstree("toggle_node",node_sel);
    $(this).jstree("open_all");
    process=false;
    //$('#tree').jstree("open_all",node_sel)
    //$(this).jstree(data.node);
})

});

const fs1 = require('fs')

const fileContents1 = fs1.readFileSync(apppath+'status_data.txt', 'utf8')
try {

    const data1 = JSON.parse(fileContents1)
    //alert(JSON.stringify(data))

    document.getElementById('endpointurl').innerText=data1.endpointurl
    if(data1.status.toString()=="Connected")
    {
        document.getElementById('endpointurl').style.backgroundColor = "green";
        document.getElementById('endpointurl').style.color = "white";

    }
    else{
        document.getElementById('endpointurl').style.backgroundColor = "red";
    }
}

```

```

        document.getElementById('endpointurl').style.color = "white";

    }
    } catch(err) {
    console.error(err)
    }
})

</script>

<script>
    //const {ipcRenderer} = require('electron')

    //alert("I am an alert box!");

    function myFunction(){
        var variable_name=document.getElementById('mlogs').innerText
        var object_nodeid=document.getElementById('object_nodeid').value
        var method_nodeid=document.getElementById('method_nodeid').value
        var value1=document.getElementById('value_write1').value
        var datatype_element1=document.getElementById('data_type1')
        var datatype1=datatype_element1.options[datatype_element1.selectedIndex].value
        var arraytype_element1=document.getElementById('array_type1')
        var arraytype1=arraytype_element1.options[arraytype_element1.selectedIndex].value

        var value2=document.getElementById('value_write2').value
        var datatype_element2=document.getElementById('data_type2')
        var datatype2=datatype_element2.options[datatype_element2.selectedIndex].value
        var arraytype_element2=document.getElementById('array_type2')
        var arraytype2=arraytype_element2.options[arraytype_element2.selectedIndex].value

        data={ 'object_nodeid':object_nodeid,'method_nodeid':method_nodeid,'value1':value1,'datatype1':datatype1,'arraytype1':arraytype1,'value2':value2,'datatype2':datatype2,'arraytype2':arraytype2}
        ipcRenderer.send('method_call',JSON.stringify(data))
    }

    function download_address_space(){
        // var variable_name=document.getElementById('mlogs').value
        ipcRenderer.send('download_address_space',"download_address_space")
    }

    ipcRenderer.on('nc_json_reply', (event, arg) => {
        $('#browse-tree').jstree(true).settings.core.data = arg;

        $('#browse-tree').jstree(true).refresh(true);
        process=true
        //$("#browse-tree").jstree("toggle_node",node_sel);
        // $('#browse-tree').jstree(true).refresh(true);
        // $('#browse-tree').jstree(true).open_node(node_sel);
        //$('#browse-tree').jstree(true)

```

```

    })

    ipcRenderer.on('browse_reply', (event, arg) => {
        //var old_text=document.getElementById('mlogs').value
        //var new_text=old_text+"\n"+JSON.stringify(arg)
        //document.getElementById('mlogs').innerText=JSON.stringify(arg)
    })

    ipcRenderer.on('nc_clear_reply', (event, arg) => {
        document.getElementById('mlogs').innerHTML=arg
    })

    ipcRenderer.on('method_call_reply', (event, arg) => {
        //data=JSON.parse(arg)
        console.log('recieved data is :',arg)
        document.getElementById('monitor_value').innerText=arg
    })

</script>
</body>
</html>

```

8.10 Event Notifier

```

<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="content-type" content="text/html; charset=windows-1252">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <style>
        body {
            font-family: "Lato", sans-serif;
        }

        h1 {color:blue;
            position:absolute;
            left:80px}

        .sidenav{
            width: 250px;
            position:absolute;
            z-index: 1;
            top: 100px;
            left: 1px;
            background: #444444;
            overflow-x: hidden;
            padding: 1px 0;
            height:100%;
        }
    </style>

```

```

.active,.sidenav a {
  padding: 6px 8px 6px 16px;
  text-decoration: none;
  font-size: 14px;
  color: #FFFFFF;
  display: block;
}

.sidenav a:hover {
  color: #C2C2C2;
}

.main {
  margin-left: 255px; /* Same width as the sidebar + left position in px */
  font-size: 14px; /* Increased text to enable scrolling */
  padding: 0px 2px;
}

.status_div {
  right: 0;
  font-size: 14px;
  position: absolute;
  padding: 8px;
  margin-top: -50px;
}

p {
  color: #2196F3;
}

input[type="text"] {
  width: 80%;
  margin-bottom: 6px;
}

input[type="password"] {
  width: 80%;
  margin-bottom: 6px;
}

input[type="checkbox"] {
  margin-bottom: 6px;
}

input[type="button"], input[type="submit"], input[type="reset"] {
  background-color: rgb(2, 48, 4);
  border: none;
  color: white;
  padding: 8px 16px;
  text-decoration: none;
  margin: 4px 2px;
}

```

```

        cursor:pointer;
        margin-bottom : 6px;
    }

    .vl {
        border-right: 1px solid green;
        height: 100%;
        top: 100px;
        position:absolute;
        left: 270px;
        margin-left: -3px;
    }

    span{
        padding : 8px;
        margin : 8px;
    }

    @media screen and (max-height: 450px){
        .sidenav {padding-top: 15px;}
        .sidenav a {font-size: 18px;}
    }

</style>

</head>
<body>


<div id="s_div" class="status_div">
    <span id="endpointurl"></span>
</div>
<h1 style="text-align: center; margin-top: -50px;"> ConnectOPC-UA </h1>

<div class="sidenav"><a href="index.html">Configure Machine</a><a href="connect.html">Connect
Machine</a> <a href="browse.html">Browse Address Space</a> <a href="monitor.html">Monitor
variables</a><a href="write.html">Write variables to the server</a><a href="pubsub.html">Create
Pub-Sub/Notifications</a><a href="pubsub_mult.html">Multiple Pub-Sub/Notifications</a><a
href="method_call.html">Method Call (Address Space)</a><a href="alarm_monitoring.html"><font
color="#FF5733">Event Notifier</font><a href="filetransfer.html">File Transfer</a></a><a
href="opcua_server.html">Create OPC-UA Server</a><a href="settings.html">Settings</a><a
href="help.html">Help</a></div>
<div class="main">
    <h2>Alarm Monitoring</h2>

    <form id="connect_form">
        Select Nodet : <input name="node_id" id="node_id" type="text"><input type = "button"
value="Add From Favourite" id="aff" onclick="add_frm_fav1();">
        <input type = "button" value="Start Alarm Monitoring" id="start" onclick="myFunction();">
        <input type = "button" value="Stop Alarm Monitoring" id="stop" onclick="myFunction1();">

```



```

    <input type = "button" value="Download Monitoring Data" id="download_data"
    onclick="download_data1();">

    </form>

    <h3>Logs</h3>
    <p id='logs'></p>
</div>

<script>
    let path_string,download_folder
    var new_text
    count=0;
    const {ipcRenderer} = require('electron')
    ipcRenderer.send('monitor_path_req','apppath')
    ipcRenderer.on('monitor_path_rep', (event, arg) => {
        const fs1 = require('fs')

        path_string=arg
        const fileContents1 = fs1.readFileSync(arg+'status_data.txt', 'utf8')
        try {

            const data1 = JSON.parse(fileContents1)
            //alert(JSON.stringify(data))

            document.getElementById('endpointurl').innerText=data1.endpointurl
            if(data1.status.toString()=="Connected")
            {
                document.getElementById('endpointurl').style.backgroundColor = "green";
                document.getElementById('endpointurl').style.color = "white";

            }
            else{
                document.getElementById('endpointurl').style.backgroundColor = "red";
                document.getElementById('endpointurl').style.color = "white";

            }
        } catch(err) {
            console.error(err)
        }

        const fs3 = require('fs')
        try {
            const fileContents10 = fs3.readFileSync(path_string+"/"+'settings_data.txt', 'utf8')

            const data = JSON.parse(fileContents10)
            //alert(JSON.stringify(data))

            download_folder=data.download_folder

```

```

    } catch(err) {
    console.error(err)
    //dialog.showErrorBox("Settings not defined","Before using the app please save your settings first")

    }

    })

    ipcRenderer.on('nc_clear_reply', (event, arg) => {
        document.getElementById('node_id').value=arg
    })

    function add_frm_fav1(){

        ipcRenderer.send('favourite_data',"start_favourite_data")
    }

    function myFunction(){
        var nodeid=document.getElementById('node_id').value
        ipcRenderer.send('alarm_monitoring_start',nodeid)
    }

    function myFunction1(){
        var nodeid=document.getElementById('node_id').value
        ipcRenderer.send('alarm_monitoring_start',"stop_monitoring")
    }

    ipcRenderer.on('alarm_monitoring_reply', (event, arg) => {
        if(count<=20)
        {
            var old_text=document.getElementById('logs').innerText
            new_text=arg+"\n"+old_text
            document.getElementById('logs').innerText=new_text
            count++
        }
        else{
            count=0;
            var old_text=document.getElementById('logs').innerText
            new_text=arg+"\n"+old_text
            document.getElementById('logs').innerText=""
            document.getElementById('logs').innerText=arg
        }

    })

    function download_data1()
    {
        const fs2 = require('fs')
        var time_identifier = (new Date).getTime();

```

```

//fs.writeFileSync('data.xlsx',xls,'binary');
    try { fs.writeFileSync(download_folder+"/"+time_identifier+".txt", new_text, 'binary');
        alert('Data saved as : '+download_folder+"/"+time_identifier+".txt")
    }

catch(e) {
    console.log("Error : "+e)
    alert("Data cannot be saved : "+e)
    //const response = dialog.showMessageBox("Data cannot be saved : "+e);
}
}

</script>
</body>
</html>

```

8.11 File Transfer

```

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=windows-1252">
<meta name="viewport" content="width=device-width, initial-scale=1">

<style>
body {
    font-family: "Lato", sans-serif;
}

h1 {color:blue;
    position:absolute;
    left:80px}

.sidenav{
    width: 250px;
    position:absolute;
    z-index: 1;
    top: 100px;
    left: 10px;
    background: #eee;
    overflow-x: hidden;
    padding: 8px 0;
}

p {

```

```

        color:#2196F3;
    }

    .sidenav{
width: 250px;
position: absolute;
z-index: 1;
top: 100px;
left: 1px;
background: #444444;
overflow-x: hidden;
padding: 1px 0;
height:100%;
    }

.active,.sidenav a {
padding: 6px 8px 6px 16px;
text-decoration: none;
font-size: 14px;
color: #FFFFFF;
display: block;
}

.sidenav a: hover {
color: #C2C2C2;
}

.main {
margin-left: 255px; /* Same width as the sidebar + left position in px */
font-size: 14px; /* Increased text to enable scrolling */
padding: 0px 2px;
}

input[type="text"] {
width: 80%;
margin-bottom : 6px;
}

input[type="password"] {
width: 80%;
margin-bottom : 6px;
}

input[type="checkbox"] {
margin-bottom : 6px;
}

input[type=button], input[type=submit], input[type=reset] {
background-color: rgb(2, 48, 4);
border: none;
color: white;
padding: 8px 16px;
text-decoration: none;
margin: 4px 2px;
cursor: pointer;
}

```

```

        margin-bottom : 6px;
    }

    .vl {
        border-right: 1px solid green;
        height: 100%;
        top: 100px;
        position: absolute;
        left: 270px;
        margin-left: -3px;
    }

    @media screen and (max-height: 450px) {
        .sidenav {padding-top: 15px;}
        .sidenav a {font-size: 18px;}
    }

    .status_div{
        right: 0;
        font-size: 14px;
        position: absolute;
        padding : 8px;
        margin-top : -50px;
    }

    span{
        padding : 8px;
        margin : 8px;
    }

</style>

</head>
<body> 
<div id="s_div" class="status_div">
    <span id="endpointurl"></span>
</div>
<h1 style="text-align: center; margin-top: -50px;"> ConnectOPC-UA </h1>

    <div class="sidenav"><a href="index.html">Configure Machine</a><a href="connect.html">Connect
Machine</a> <a href="browse.html">Browse Address Space</a> <a href="monitor.html">Monitor
variables</a><a href="write.html">Write variables to the server</a><a href="pubsub.html">Create
Pub-Sub/Notifications</a><a href="pubsub_mult.html">Multiple Pub-Sub/Notifications</a><a
href="method_call.html">Method Call (Address Space)</a><a href="alarm_monitoring.html">Event
Notifier</a> <a href="filetransfer.html"><font color="#FF5733">File Transfer</font></a> <a
href="opcuaserver.html">Create OPC-UA Server</a><a href="settings.html">Settings</a><a
href="help.html">Help</a></div>
    <div class="main">
        <h2>Read File From the Machine (Server)</h2>
        <form id="connect_form">

```

```

    Variable Name (Full Qualified Node Id)<br> <input name="variable_name" id="variable_name"
type="text"> <input type = "button" value="Add From Favourite" id="aff" onclick="add_frm_fav1();">
<br>
    <input type = "button" value="Read" id="read" onclick="read_from_server();">
    <input type = "button" value="Download File" id="download_file" onclick="myFunction();">
</form>

<h2>Write File to the Machine (Server)</h2>
<form id="connect1_form">
    Variable Name (Full Qualified Node Id)<br> <input name="variable_name1" id="variable_name1"
type="text"> <input type = "button" value="Add From Favourite" id="aff" onclick="add_frm_fav1();">
<br>
    <input type = "button" value="Write" id="write" onclick="write_to_server2();">
    <input type = "button" value="Upload File" id="upload" onclick="upload_file1();">
</form>

<h3>Logs</h3>
<p id='mlogs'></p>
</div>

<script>
    const {ipcRenderer} = require('electron')
    ipcRenderer.send('monitor_path_req','apppath')
    ipcRenderer.on('monitor_path_rep', (event, arg) => {
        const fs1 = require('fs')

        const fileContents1 = fs1.readFileSync(arg+'status_data.txt', 'utf8')
        try {

            const data1 = JSON.parse(fileContents1)
            //alert(JSON.stringify(data))

            document.getElementById('endpointurl').innerText=data1.endpointurl
            if(data1.status.toString()=="Connected")
            {
                document.getElementById('endpointurl').style.backgroundColor = "green";
                document.getElementById('endpointurl').style.color = "white";

            }
            else{
                document.getElementById('endpointurl').style.backgroundColor = "red";
                document.getElementById('endpointurl').style.color = "white";

            }
        } catch(err) {
            console.error(err)
        }

    })

    //document.getElementById('mlogs').innerText="collecting logs"
</script>

```

```

<script>

ipcRenderer.on('nc_clear_reply', (event, arg) => {
    document.getElementById('variable_name').value=arg
})

ipcRenderer.on('upload_data_content', (event, arg) => {
    document.getElementById('mlogs').innerHTML=arg
})

//alert("I am an alert box!");
function myFunction(){
    var variable_name=document.getElementById('variable_name').value
    ipcRenderer.send('monitor_data',variable_name)
}

function read_from_server(){
    var variable_name=document.getElementById('variable_name').value
    ipcRenderer.send('read_from_server',variable_name)
}

function upload_file1(){
    ipcRenderer.send('upload_file_ft',"oo")
}

function add_frm_fav1(){

    ipcRenderer.send('favourite_data',"start_favourite_data")
}

function write_to_server2(){
    ipcRenderer.send("write_to_server",document.getElementById('variable_name1').value)
    //alert('write to server function called')
}

ipcRenderer.on('monitor1_reply', (event, arg) => {
    var old_text=document.getElementById('mlogs').innerText
    var new_text=arg+"\n"+old_text
    document.getElementById('mlogs').innerText=new_text
})

</script>
</body>
</html>

```

8.12 Create OPC-UA Server

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=windows-1252">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <script>
    $=require('jquery')
  </script>
  <style>
body {
  font-family: "Lato", sans-serif;
}

  h1 {color:blue;
    position:absolute;
    left:80px}

  .sidenav {
width: 250px;
position: absolute;
z-index: 1;
top: 100px;
left: 1px;
background: #444444;
overflow-x: hidden;
padding: 1px 0;
height:100%;
}

.active,.sidenav a {
padding: 6px 8px 6px 16px;
text-decoration: none;
font-size: 14px;
color: #FFFFFF;
display: block;
}

.sidenav a: hover {
color: #C2C2C2;
}

.main {
margin-left: 255px; /* Same width as the sidebar + left position in px */
font-size: 14px; /* Increased text to enable scrolling */
padding: 0px 2px;
}

p {
color: #2196F3;
}

span{
```



```

padding: 6px 6px 1px 6px;
}

input[type="text"] {
width: 80%;
margin-bottom : 6px;
}

input[type="password"] {
width: 80%;
margin-bottom : 6px;
}

input[type="checkbox"] {
margin-bottom : 6px;
}

input[type=button], input[type=submit], input[type=reset] {
background-color: #4CAF50;
border: none;
color: white;
padding: 8px 16px;
text-decoration: none;
margin: 4px 2px;
cursor: pointer;
margin-bottom : 6px;
}

.vl {
border-right: 1px solid green;
height: 100%;
top: 100px;
position: absolute;
left: 270px;
margin-left: -3px;
}

.tree_style
{
border : 1px solid blue;
padding : 5px 5px;
overflow: scroll;
height: 500px;
margin : 5px;
}

.status_div{
right: 0;
font-size: 14px;
position: absolute;

```

```

padding : 8px;
margin-top : -50px;

}

span{
padding : 8px;
margin : 8px;
}

.span_border111{

width : 800px;
word-wrap: break-word;
word-break: break-all;

}

@media screen and (max-height: 450px) {
.sidenav {padding-top: 15px;}
.sidenav a {font-size: 18px;}
}

hr {
position : absolute;
width: 100%;
left: 2px;
}
</style> </head>
<body> 

<h1 style="text-align: center; margin-top: -50px;"> ConnectOPC-UA </h1>

<div class="sidenav"><a href="index.html">Configure Machine</a><a href="connect.html">Connect
Machine</a> <a href="browse.html">Browse Address Space</a> <a href="monitor.html">Monitor
variables</a><a href="write.html">Write variables to the server</a><a href="pubsub.html">Create
Pub-Sub/Notifications</a><a href="pubsub_mult.html">Multiple Pub-Sub/Notifications</a><a
href="method_call.html">Method Call (Address Space)</a><a href="alarm_monitoring.html">Event
Notifier</a> <a href="filetransfer.html">File Transfer</a> <a href="opcuaserver.html"><font
color="#FF5733">Create OPC-UA Server</font></a><a href="settings.html">Settings</a><a
href="help.html">Help</a></div>
<div class="main">
<h2>Create OPC-UA Server</h2>
<p>Note: This server is for the diagnostics and trianing purpose only.</p>
<h3>Enter the Name of the Server : </h3>
<input name="mlogs" id="mlogs" type="text"><input type = "button" value="Create Server"
id="server" onclick="myFunction();"><input type = "button" value="Shutdown Server"
id="shutdownserver" onclick="myFunction1();">
<h3>Logs</h3>
<div class="span_border111">
<span id='monitor_value'></span>

</div>

```

```

</div>

<script>

</script>

<script>
// document.getElementById('server').disabled="false"

const {ipcRenderer} = require('electron')
function myFunction(){
  var variable_name=document.getElementById('mlogs').value

  data={"variable_name":variable_name,"type":"connect"}
  ipcRenderer.send('create_server',JSON.stringify(data))
  document.getElementById('monitor_value').innerText=""

}

function myFunction1(){
  var variable_name=document.getElementById('mlogs').value
  ipcRenderer.send('shut_down_server',"shut_down")

}

ipcRenderer.send('server_enquiry',"status")
ipcRenderer.on('server_enquiry_reply', (event, arg) => {
data=JSON.parse(arg)
if(data.status)
{
  console.log('Disconnect server : '+ data.text)
  document.getElementById('monitor_value').innerText=data.text
  document.getElementById('server').disabled=true
  document.getElementById('shutdownserver').disabled=false
  document.getElementById('shutdownserver').style.backgroundColor = "green";
  document.getElementById('server').style.backgroundColor = "grey";
}
else
{
  //document.getElementById('monitor_value').innerText=arg
  console.log('Disconnect server')
  document.getElementById('monitor_value').innerText=data.text
  document.getElementById('server').disabled=false
  document.getElementById('shutdownserver').disabled=true
  document.getElementById('shutdownserver').style.backgroundColor = "grey";
  document.getElementById('server').style.backgroundColor = "green";
  //document.getElementById('server').disabled="false"
  //document.getElementById('server').style.backgroundColor = "green";
  //document.getElementById('server').value="Connect Server"

```

```

    }

    })

    /*ipcRenderer.on('create_server_reply1', (event, arg)=> {

    if(arg=="disconnect")
    {
        document.getElementById('monitor_value').innerText=data.text
        document.getElementById('server').value="Disconnect Server"
        document.getElementById('server').disabled="true"
        document.getElementById('server').style.backgroundColor = "grey";

    }
    else
    {
        //document.getElementById('monitor_value').innerText=arg
        document.getElementById('server').value="Connect Server"
        document.getElementById('server').disabled="false"
        document.getElementById('server').style.backgroundColor = "green";
    }

    })*/

    ipcRenderer.on('create_server_reply', (event, arg)=> {
    var old_text=document.getElementById('monitor_value').innerText
    var new_text=old_text+"\n"+arg
    document.getElementById('monitor_value').innerText=new_text
    })

</script>
</body>
</html>

```

8.13 Settings

```

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=windows-1252">
<meta name="viewport" content="width=device-width, initial-scale=1">

<style>
body {
    font-family: "Lato", sans-serif;
}

h1 {color:blue;

```

```

        position: absolute;
        left: 80px;

        .sidenav {
            width: 250px;
            position: absolute;
            z-index: 1;
            top: 100px;
            left: 1px;
            background: #444444;
            overflow-x: hidden;
            padding: 1px 0;
            height: 100%;
        }

        .active, .sidenav a {
            padding: 6px 8px 6px 16px;
            text-decoration: none;
            font-size: 14px;
            color: #FFFFFF;
            display: block;
        }

        .sidenav a: hover {
            color: #C2C2C2;
        }

        .main {
            margin-left: 255px; /* Same width as the sidebar + left position in px */
            font-size: 14px; /* Increased text to enable scrolling */
            padding: 0px 2px;
        }

        p {
            color: #2196F3;
        }

        select {
            padding: 5px;
            margin: 5px;
        }

        .container {
            border-radius: 5px;
            padding: 20px;
        }

        /* Floating column for labels: 25% width */
        .col-25 {
            float: left;
            width: 50%;
            margin-top: 2px;
        }
    }

```

```

/* Floating column for inputs: 75% width */
.col-75 {
    float: left;
    width: 50%;
    margin-top: 2px;
}

input[type="text"] {
    width: 50%;
    margin: 2px;
}

input[type="password"] {
    width: 50%;
    margin-bottom: 6px;
}

input[type="checkbox"] {
    margin: 2px;
}

input[type=button], input[type=submit], input[type=reset] {
    background-color: #4CAF50;
    border: none;
    color: white;
    padding: 8px 16px;
    text-decoration: none;
    margin: 4px 2px;
    cursor: pointer;
    margin-bottom: 6px;
}

.vl {
    border-right: 1px solid green;
    height: 100%;
    top: 100px;
    position: absolute;
    left: 270px;
    margin-left: -3px;
}

.input_border{
    border: 1px solid blue;
    margin: 10px;
    padding: 10px;
}

@media screen and (max-height: 450px){
    .sidenav {padding-top: 15px;}
    .sidenav a {font-size: 18px;}

```

```

    }

    hr {
        position : absolute;
        width: 100%;
        left: 2px;
    }
</style>

</head>
<body> 
    <h1 style="text-align: center; margin-top: -50px;"> ConnectOPC-UA </h1>

    <div class="sidenav"><a href="index.html">Configure Machine</a><a href="connect.html">Connect
Machine</a> <a href="browse.html">Browse Address Space</a> <a href="monitor.html">Monitor
variables</a><a href="write.html">Write variables to the server</a><a href="pubsub.html">Create
    Pub-Sub/Notifications</a><a href="pubsub_mult.html">Multiple Pub-Sub/Notifications</a><a
href="method_call.html">Method Call (Address Space)</a><a href="alarm_monitoring.html">Event
Notifier</a><a href="filetransfer.html">File Transfer</a> <a href="opcuaserver.html">Create OPC-UA
Server</a><a href="settings.html"><font color="#FF5733">Settings</font></a><a
href="help.html">Help</a></div>
    <div class="main">
        <h2>Settings</h2>

        <div class="container">
            <form id="connect_form">

                <div class="row">
                    <div class="col-25"><label for="fname">Connection Timeout (in Seconds)</label></div>
                    <div class="col-75"><input id="conn_timeout" type="text"> </div> </div>

                <div class="row">
                    <div class="col-25"><label for="fname">Maximum Read/Write/Subscribe
Operations</label></div>
                    <div class="col-75"><input id="max_oper" type="text"> </div> </div>

                <div class="row">
                    <div class="col-25"><label for="fname">Secure Channel Re-authentication Enabled
</label></div>
                    <div class="col-75"><input id="sec_conn" value="male" unchecked=""
type="checkbox"> </div> </div>

                <div class="row">
                    <div class="col-25"><label for="fname">Failover Enabled </label></div>
                    <div class="col-75"><input name="connection_type" id="failover_enabled"
value="male" unchecked="" type="checkbox"></div> </div>

                <div class="row">
                    <div class="col-25"><label for="fname">Failover Endpoint </label></div>
                    <div class="col-75"><input name="machine_no" id="failover_endpoint" type="text">
</div> </div>

                <div class="row">
                    <div class="col-25"><label for="fname">Server Port </label></div>
                    <div class="col-75"><input name="machine_no" id="server_port" type="text"> </div>
</div>

```

```

        <div class="row">
            <div class="col-25"><label for="fname">Minimum Sampling Interval </label></div>
            <div class="col-75"><input name="machine_no" id="min_sampling" type="text"> </div>
        </div>

        <div class="row">
            <div class="col-25"><label for="fname">Downloads Folder </label></div>
            <div class="col-75"><input name="machine_no" id="download_folder" type="text">
        </div> </div>

        <div class="row">
            <div class="col-25"><label for="fname">Connection Strategy [Maximum Retries]
        </label> </div>
            <div class="col-75"><input name="machine_no" id="cs_max_rt" type="text"> </div>
        </div>

        <div class="row">
            <div class="col-25"><label for="fname">Connection Strategy [Enable Keep Session
        Alive] </label></div>
            <div class="col-75"><input name="connection_type" id="cs_session_alive"
        value="male" unchecked="" type="checkbox"> </div> </div>

        <div class="row">
            <div class="col-25"><label for="fname">Connection Strategy [Endpoint must exist for
        session] </label></div>
            <div class="col-75"><input name="connection_type" id="cs_endpoint_exist"
        value="male" unchecked="" type="checkbox"> </div> </div>

        <div class="row">
            <div class="col-25"><label for="fname">Subscribe Settings [Requested Publishing
        interval] </label></div>
            <div class="col-75"><input id="pub_interval" type="text"> </div> </div>

        <div class="row">
            <div class="col-25"><label for="fname">Subscribe Settings [Requested Lifetime Count]
        </label> </div>
            <div class="col-75"><input id="lifetime_count" type="text"> </div> </div>

        <div class="row">
            <div class="col-25"><label for="fname">Subscribe Settings [Requested Max Keep Alive
        Count] </label></div>
            <div class="col-75"><input id="keep_alive_count" type="text"> </div> </div>

        <div class="row">
            <div class="col-25"><label for="fname">Subscribe Settings [Maximum Notification per
        publish] </label></div>
            <div class="col-75"><input id="max_not" type="text"> </div> </div>

        <div class="row">
            <div class="col-25"><label for="fname">Subscribe Settings [Publishing Enabled]</label>
        </div>
            <div class="col-75"><input name="connection_type" id="publishing_enabled"
        value="male" unchecked="" type="checkbox"> </div> </div>

        <div class="row">

```



```

        <div class="col-25"><label for="fname"> Subscribe Settings [Priority] </label></div>
        <div class="col-75"><input id="priority" type="text"> </div> </div>

        <div class="row">
            <div class="col-25"><label for="fname"> Node Id of parent node (OPCUA Client Only)
</label> </div>
            <div class="col-75"><input id="p_nodeid" type="text"> </div> </div>
            <input type = "submit" value="Save" id="connect" onclick="myFunction();">
        </form>
    </div>
</div>

<script>
    let apppath
    const {ipcRenderer} = require('electron')
    ipcRenderer.send('setting_app_path_req',"apppath")

    ipcRenderer.on('setting_app_path_rep', (event, arg) => {
        apppath=arg
        let fs = require('fs');
        let fs1 = require('fs');
        const fileContents1 = fs1.readFileSync(arg+'settings_data.txt', 'utf8')
        try {

            const data = JSON.parse(fileContents1)
            //alert(JSON.stringify(data))
            document.getElementById('conn_timeout').value=data.connection_timeout
            document.getElementById('max_oper').value=data.max_oper
            document.getElementById('failover_endpoint').value=data.failover_endpoint
            document.getElementById('server_port').value=data.server_port
            document.getElementById('min_sampling').value=data.min_sampling
            document.getElementById('download_folder').value=data.download_folder
            document.getElementById('sec_conn').checked=data.sec_conn
            document.getElementById('failover_enabled').checked=data.failover_enabled
            document.getElementById('cs_max_rt').value=data.cs_max_rt
            document.getElementById('cs_session_alive').checked=data.cs_session_alive
            document.getElementById('cs_endpoint_exist').checked=data.cs_endpoint_exist
            document.getElementById('pub_interval').value=data.pub_interval
            document.getElementById('lifetime_count').value=data.lifetime_count
            document.getElementById('keep_alive_count').value=data.keep_alive_count
            document.getElementById('max_not').value=data.max_not
            document.getElementById('publishing_enabled').checked=data.publishing_enabled
            document.getElementById('priority').value=data.priority
            document.getElementById('p_nodeid').value=data.p_nodeid

        } catch(err) {
            console.error(err)
        }

    })

    //document.getElementById('machine_name').value="OPC Machine"
    //document.getElementById('machine_no').value="123"
    //document.getElementById('machine_location').value="BLK2BAY1"

```

```

//alert("I am an alert box!");
function myFunction(){
  var conn_timeout=document.getElementById('conn_timeout').value
  var max_oper=document.getElementById('max_oper').value
  var failover_endpoint=document.getElementById('failover_endpoint').value
  var server_port=document.getElementById('server_port').value
  var min_sampling=document.getElementById('min_sampling').value
  var download_folder=document.getElementById('download_folder').value
  //alert(username);
  //var config_data={}
  //var key='opc_connect_config_data'
  //config_data[key]=[];
  var
data={'connection_timeout':conn_timeout,'failover_endpoint':failover_endpoint,'max_oper':max_oper,
'sec_conn':document.getElementById('sec_conn').checked,'failover_enabled':document.getElementByI
d('failover_enabled').checked,'server_port':server_port,'min_sampling':min_sampling,'download_folde
r':download_folder,'cs_max_rt':document.getElementById('cs_max_rt').value,'cs_session_alive':docum
ent.getElementById('cs_session_alive').checked,'cs_endpoint_exist':document.getElementById('cs_end
point_exist').checked,'pub_interval':document.getElementById('pub_interval').value,'lifetime_count':do
cument.getElementById('lifetime_count').value,'keep_alive_count':document.getElementById('keep_ali
ve_count').value,'max_not':document.getElementById('max_not').value,'publishing_enabled':documen
t.getElementById('publishing_enabled').checked,'priority':document.getElementById('priority').value,'p
_nodeid':document.getElementById('p_nodeid').value};
  //config_data[key].push(data)
  let fs = require('fs');
  try { fs.writeFileSync(apppath+'settings_data.txt',JSON.stringify(data), 'utf-8');

    alert("Settings Saved : app will be closed, reopen the app")
    ipcRenderer.send('setting_data',JSON.stringify(data))
    //app.quit()
  }
  catch(e) { alert('Failed to save the file ! : '+e); }

};

</script>
</body>
</html>

```

8.14 Help

```

<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=windows-1252">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <style>

```

```

body {
  font-family: "Lato", sans-serif;
}

h1 {color:blue;
  position:absolute;
  left:80px}

p {
  color:#2196F3;
}

.sidenav{
width: 250px;
position:absolute;
z-index: 1;
top: 100px;
left: 1px;
background: #444444;
overflow-x: hidden;
padding:1px 0;
height:100%;
}

.active,.sidenav a {
padding: 6px 8px 6px 16px;
text-decoration: none;
font-size: 14px;
color: #FFFFFF;
display: block;
}

.sidenav a: hover {
color: #C2C2C2;
}

.main {
margin-left: 255px; /* Same width as the sidebar + left position in px */
font-size: 14px; /* Increased text to enable scrolling */
padding: 0px 2px;
}

input[type="text"] {
width: 80%;
margin-bottom : 6px;
}

input[type="password"] {
width: 80%;
margin-bottom : 6px;
}

```

```

input[type="checkbox"] {
    margin-bottom : 6px;
}

input[type=button], input[type=submit], input[type=reset] {
    background-color:#4CAF50;
    border: none;
    color: white;
    padding: 8px 16px;
    text-decoration: none;
    margin: 4px 2px;
    cursor: pointer;
    margin-bottom : 6px;
}

.vl {
    border-right: 1px solid green;
    height: 100%;
    top: 100px;
    position: absolute;
    left: 270px;
    margin-left: -3px;
}

@media screen and (max-height: 450px){
    .sidenav {padding-top: 15px;}
    .sidenav a {font-size: 18px;}
}

.status_div{
    right: 0;
    font-size: 14px;
    position: absolute;
    padding : 8px;
    margin-top : -50px;
}

span{
    padding : 8px;
    margin : 8px;
}

.center1 {
    display: block;
    margin-left: auto;
    margin-right: auto;
}

.p12{
    margin : 1px;
    padding : 1px;
    font-family: "Times New Roman", Times, serif;
    font-size: 20px;
}

```

```

    }

    img{
        padding:10px;
    }

    tr:nth-child(even) {background-color:#c2c2c2;}
    th, td {
        border-bottom: 1px solid #ddd;
        padding:10px;
    }

    table {
        width: 50%;
    }

</style>

</head>
<body> 
<div id="s_div" class="status_div">
    <span id="endpointurl"></span>
</div>
<h1 style="text-align: center; margin-top: -50px;"> ConnectOPC-UA </h1>

    <div class="sidenav"><a href="index.html">Configure Machine</a><a href="connect.html">Connect
Machine</a> <a href="browse.html">Browse Address Space</a> <a href="monitor.html">Monitor
variables</a><a href="write.html">Write variables to the server</a><a href="pubsub.html">Create
    Pub-Sub/Notifications</a><a href="pubsub_mult.html">Multiple Pub-Sub/Notifications</a><a
href="method_call.html">Method Call (Address Space)</a><a href="alarm_monitoring.html">Event
Notifier</a> <a href="filetransfer.html">File Transfer</a> <a href="opcuaserver.html">Create OPC-UA
Server</a><a href="settings.html">Settings</a><a href="help.html"><font
color="#FF5733">Help</font></a></div>
    <div class="main">
        <h2> About the Software </h2>
        <p class="p12">ConnectOPC-UA v4.2.8</p>
        <p class="p12">Developed by : </p>
        <p class="p12">Bhupendra Kumar Mishra </p>
        <p class="p12">Dy Manager (EMX) </p>
        <p class="p12">BHEL Bhopal</p>
        <p class="p12">Phone No : +91-9406903375</p>
        <p class="p12">Email : bhupendramishra@bhel.in, bhupendramishr@gmail.com</p>
        <br><br><h2>Used Packages</h2>
        
        <br><h2>Supported Features</h2>
        <table >

            <td>Find Servers</td>
            <td>Yes</td>
        </tr>
        <tr>

            <td>Get Endpoints</td>

```

```

        <td>Yes</td>
    </tr>

    <tr>
        <td>Register Servers</td>
        <td>No</td>
    </tr>
    <tr>
        <td>Find Servers on Network</td>
        <td>No</td>
    </tr>

    <tr>
        <td>Open Secure Channel</td>
        <td>Yes</td>
    </tr>

    <tr>
        <td>Close Secure Channel</td>
        <td>Yes</td>
    </tr>

    <tr>
        <td>Create Session</td>
        <td>Yes</td>
    </tr>

    <tr>
        <td>Close Session</td>
        <td>Yes</td>
    </tr>

    <tr>
        <td>Activate Session</td>
        <td>Yes</td>
    </tr>

    <tr>
        <td>Browse</td>
        <td>Yes</td>
    </tr>

    <tr>
        <td>Browse Next</td>
        <td>Yes</td>
    </tr>

    <tr>
        <td>Translate Browse Path to Nodeids</td>
        <td>Yes</td>
    </tr>

    <tr>
        <td>Register Nodes</td>
        <td>Yes</td>
    </tr>

```

```

<tr>
  <td>Unregister Nodes</td>
  <td>Yes</td>
</tr>

<tr>
  <td>Attribute Read</td>
  <td>Yes</td>
</tr>

<tr>
  <td>Attribute Write</td>
  <td>Yes</td>
</tr>

<tr>
  <td>History Read</td>
  <td>No</td>
</tr>

<tr>
  <td>History Update</td>
  <td>No</td>
</tr>

<tr>
  <td>Create Monitored Items</td>
  <td>No</td>
</tr>

<tr>
  <td>Modify Monitored Items</td>
  <td>No</td>
</tr>

<tr>
  <td>Set Monitoring Mode</td>
  <td>No</td>
</tr>

<tr>
  <td>Set Triggerring</td>
  <td>Yes</td>
</tr>

<tr>
  <td>Delete Monitored Items</td>
  <td>Yes</td>
</tr>

<tr>
  <td>Create Subscription</td>
  <td>Yes</td>
</tr>

```

	<td><td>Modify Subscription</td></td>	<td>Modify Subscription</td>
	<td><td>Yes</td></td>	<td>Yes</td>
	<td></tr></td>	</tr>
	<td><tr></td>	<tr>
	<td><td>Delete Subscription</td></td>	<td>Delete Subscription</td>
	<td><td>Yes</td></td>	<td>Yes</td>
	<td></tr></td>	</tr>
	<td><tr></td>	<tr>
	<td><td>Publish</td></td>	<td>Publish</td>
	<td><td>Yes</td></td>	<td>Yes</td>
	<td></tr></td>	</tr>
	<td><tr></td>	<tr>
	<td><td>Republish</td></td>	<td>Republish</td>
	<td><td>Yes</td></td>	<td>Yes</td>
	<td></tr></td>	</tr>
	<td><tr></td>	<tr>
	<td><td>Transfer</td>	<td>Transfer
Subscription</td>	<td><td>Yes</td></td>	<td>Yes</td>
	<td></tr></td>	</tr>
	<td><tr></td>	<tr>
Binary (OPC.TCP)</td>	<td><td>UA-TCP UA-SC UA</td>	<td>UA-TCP UA-SC UA
	<td><td>Yes</td></td>	<td>Yes</td>
	<td></tr></td>	</tr>
	<td><tr></td>	<tr>
[NONE]</td>	<td><td>Security Policy</td>	<td>Security Policy
	<td><td>Yes</td></td>	<td>Yes</td>
	<td></tr></td>	</tr>
	<td><tr></td>	<tr>
[Basic128Rsa15]</td>	<td><td>Security Policy</td>	<td>Security Policy
	<td><td>Yes</td></td>	<td>Yes</td>
	<td></tr></td>	</tr>
	<td><tr></td>	<tr>
[Basic256]</td>	<td><td>Security Policy</td>	<td>Security Policy
	<td><td>Yes</td></td>	<td>Yes</td>
	<td></tr></td>	</tr>
	<td><tr></td>	<tr>
[Basic256Sha256]</td>	<td><td>Security Policy</td>	<td>Security Policy
	<td><td>Yes</td></td>	<td>Yes</td>
	<td></tr></td>	</tr>
	<td><tr></td>	<tr>


```

<td>Authentication [Anonymous]</td>
<td>Yes</td>
</tr>

<tr>

<td>Authentication [Username and Password]</td>
<td>Yes</td>
</tr>

<tr>

<td>Authentication [X509 Certificate]</td>
<td>No</td>
</tr>

<tr>
<td>Client Behaviour</td>
<td>Base
</tr>

<tr>
<td>Address Space Lookup</td>
<td>Yes</td>
</tr>

<tr>
<td>Attribute Read</td>
<td>Yes</td>
</tr>

<tr>
<td>Data Change Subscription</td>
<td>Yes</td>
</tr>

<tr>
<td>Data Access</td>
<td>Yes</td>
</tr>

```

```

<tr>

<td>Discovery</td>

<td>Yes</td>

</tr>

<tr>

<td>Pub-Sub</td>

<td>Yes</td>

</tr>
</table>
</div>

<script>
    const {ipcRenderer} = require('electron')
    ipcRenderer.send('help_path_req','apppath')
    ipcRenderer.on('help_path_rep', (event, arg) => {
        const fs1 = require('fs')

const fileContents1 = fs1.readFileSync(arg+'status_data.txt', 'utf8')
try {

    const data1 = JSON.parse(fileContents1)
    //alert(JSON.stringify(data))

    document.getElementById('endpointurl').innerText=data1.endpointurl
    if(data1.status.toString()=="Connected")
    {
        document.getElementById('endpointurl').style.backgroundColor = "green";
        document.getElementById('endpointurl').style.color = "white";

    }
    else{
        document.getElementById('endpointurl').style.backgroundColor = "red";
        document.getElementById('endpointurl').style.color = "white";

    }
    } catch(err) {
        console.error(err)
    }

    })

    //document.getElementById('mlogs').innerText="collecting logs"
</script>

<script>

```

```
//alert("I am an alert box!");

</script>
</body>
</html>
```

8.15 Favoutites

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Add Node from Favourites</title>
  <style>
    body {
      margin: 0;
      min-width: 250px;
    }

    /* Include the padding and border in an element's total width and height */
    * {
      box-sizing: border-box;
    }

    /* Remove margins and padding from the list */
    ul {
      margin: 0;
      padding: 0;
    }

    /* Style the list items */
    ul li {
      cursor: pointer;
      position: relative;
      padding: 4px;
      list-style-type: none;
      background: #eee;
      font-size: 14px;
      transition: 0.2s;

      /* make the list items unselectable */
      -webkit-user-select: none;
      -moz-user-select: none;
      -ms-user-select: none;
      user-select: none;
    }
  </style>

```

```

/* Set all odd list items to a different color (zebra-stripes) */
ul li:nth-child(odd) {
    background: #f9f9f9;
}

/* Darker background-color on hover */
ul li:hover {
    background: #ddd;
}

/* When clicked on, add a background color and strike out text */
ul li.checked {
    background: #888;
    color: #fff;
    text-decoration: line-through;
}

/* Add a "checked" mark when clicked on */
ul li.checked::before {
    content: "";
    position: absolute;
    border-color: #fff;
    border-style: solid;
    border-width: 0 2px 2px 0;
    top: 10px;
    left: 16px;
    transform: rotate(45deg);
    height: 15px;
    width: 7px;
}

/* Style the close button */
.close {
    position: absolute;
    right: 0;
    top: 0;
    padding: 12px 16px 12px 16px;
}

.close:hover {
    background-color: #f44336;
    color: white;
}

/* Style the header */
.header {
    background-color: #f44336;
    padding: 30px 40px;
    color: white;
    text-align: center;
}

/* Clear floats after the header */
.header:after {
    content: "";

```

```

        display: table;
        clear: both;
    }

    /* Style the input */
    input[type=button], input[type=submit], input[type=reset] {
background-color: rgb(2, 48, 4);
border: none;
color: white;
padding: 8px 16px;
text-decoration: none;
margin: 4px 2px;
cursor: pointer;
margin-bottom : 6px;
    }

    /* Style the "Add" button */
    .addBtn {
padding: 10px;
width: 25%;
background: #d9d9d9;
color: #555;
float: left;
text-align: center;
font-size: 16px;
cursor: pointer;
transition: 0.3s;
border-radius: 0;
    }

    .addBtn:hover {
background-color: #bbb;
    }
</style>
</head>
<body>
    <h2>Click to Add Nodes in the Main Window</h2>
    <ul id="dynamic-list" style="list-style-type:none;"></ul>
    <input type = "button" value="Close" id="close" onclick="close2();">
    <script>
        let apppath
        const {ipcRenderer} = require('electron')
        let fs2 = require('fs');
        let fs3 = require('fs');

        function close2()
        {
            ipcRenderer.send("favourite_data_node", "")
        }

        ipcRenderer.send("browse_path_req", apppath)
        ipcRenderer.on('browse_path_rep', (event, arg) => {
            apppath=arg

            const mrfileContents = fs2.readFileSync(apppath+'opc_conf_data.txt', 'utf8')
            var datamr = JSON.parse(mrfileContents)

```

```

var datamr_mc=datamr.machine_name

const mrfileContents1 = fs3.readFileSync(apppath+datamr_mc+"_"+"favourites.txt", 'utf8')
var datamr1 = mrfileContents1
var data_line=datamr1.split("\n")
var ul = document.getElementById("dynamic-list");
for(i=0;i<data_line.length;i++)
{
    var li = document.createElement("li");
    li.setAttribute('id',data_line[i]);
    li.appendChild(document.createTextNode(data_line[i]));
    ul.appendChild(li);
}
})
var list= document.querySelector('ul');
list.addEventListener('click', function(ev) {
    if (ev.target.tagName === 'LI') {
        ipcRenderer.send("favourite_data_node", ev.target.id)
    }
}, false);

</script>
</body>
</html>

```

8.16 View Favourites

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="content-type" content="text/html; charset=windows-1252">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Add Node from Favourites</title>
    <style>

body {
    font-family: "Lato", sans-serif;
}

h1 {color:blue;
    position:absolute;
    left:80px}

.sidenav{
width: 250px;
position:absolute;
z-index: 1;

```

```

top: 100px;
left: 1px;
background: #444444;
overflow-x: hidden;
padding: 1px 0;
height: 100%;
}

.active, .sidenav a {
padding: 6px 8px 6px 16px;
text-decoration: none;
font-size: 14px;
color: #FFFFFF;
display: block;
}

.sidenav a: hover {
color: #C2C2C2;
}

.main {
margin-left: 255px; /* Same width as the sidebar + left position in px */
font-size: 14px; /* Increased text to enable scrolling */
padding: 0px 2px;
}

p {
color: #2196F3;
}

label {
color: #9E0909;
font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode', Geneva,
Verdana, sans-serif
}

span {
padding: 6px 6px 1px 6px;
}

input[type="text"] {
width: 90%;
margin-bottom : 2px;
border: none;
color: #201096;
font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode', Geneva,
Verdana, sans-serif
}

.input_text_class {
width: 80%;
margin-bottom : 6px;
}

```

```

input[type="password"] {
    width: 80%;
    margin-bottom : 6px;
}

input[type="checkbox"] {
    margin-bottom : 6px;
}

input[type=button], input[type=submit], input[type=reset] {
    background-color:rgb(2,48, 4);
    border: none;
    color:white;
    padding:8px 16px;
    text-decoration: none;
    margin: 4px 2px;
    cursor:pointer;
    margin-bottom : 6px;
}

.vl {
    border-right: 1px solid green;
    height: 100%;
    top: 100px;
    position:absolute;
    left: 270px;
    margin-left: -3px;
}

.tree_style
{
    border : 1px solid blue;
    padding : 5px 5px;
    overflow: scroll;
    height: 500px;
    margin : 5px;
}

.col-50-border {
    border : 1px solid blue;
    float: left;
    width: 45%;
    margin-top: 2px;
    overflow: scroll;
    height: 870px;
    padding: 6px;
}

.col-20 {
    float: left;
    width: 25%;
    margin-top: 2px;
}

```



```

        padding : 2px;

    }

.col-75 {
    float: left;
    width: 70%;
    margin-top: 2px;
    padding : 2px;

}

.status_div{
    right: 0;
    font-size: 14px;
    position: absolute;
    padding : 8px;
    margin-top : -50px;

}

span{
    padding : 8px;
    margin : 8px;
}

.span_border111{

    width : 800px;
    word-wrap: break-word;
    word-break: break-all;

}

@media screen and (max-height: 450px) {
    .sidenav {padding-top: 15px;}
    .sidenav a {font-size: 18px;}
}

hr {
    position : absolute;
    width: 100%;
    left: 2px;
}

/* Include the padding and border in an element's total width and height */
* {
    box-sizing: border-box;
}

/* Remove margins and padding from the list */
ul {
    margin: 0;

```

```

padding: 0;
}

/* Style the list items */
ul li {
  cursor: pointer;
  position: relative;
  padding: 4px;
  list-style-type: none;
  background: #eee;
  font-size: 14px;
  transition: 0.2s;

  /* make the list items unselectable */
  -webkit-user-select: none;
  -moz-user-select: none;
  -ms-user-select: none;
  user-select: none;
}

/* Set all odd list items to a different color (zebra-stripes) */
ul li:nth-child(odd) {
  background: #f9f9f9;
}

/* Darker background-color on hover */
ul li:hover {
  background: #ddd;
}

/* When clicked on, add a background color and strike out text */
ul li.checked {
  background: #888;
  color: #fff;
  text-decoration: line-through;
}

/* Add a "checked" mark when clicked on */
ul li.checked::before {
  content: "";
  position: absolute;
  border-color: #fff;
  border-style: solid;
  border-width: 0 2px 2px 0;
  top: 10px;
  left: 16px;
  transform: rotate(45deg);
  height: 15px;
  width: 7px;
}

/* Style the close button */
.close {
  position: absolute;
  right: 0;
  top: 0;

```

```

        padding: 12px 16px 12px 16px;
    }

    .close:hover {
        background-color: #f44336;
        color: white;
    }

    /* Style the header */
    .header {
        background-color: #f44336;
        padding: 30px 40px;
        color: white;
        text-align: center;
    }

    /* Clear floats after the header */
    .header:after {
        content: "";
        display: table;
        clear: both;
    }

    /* Style the input */
    input {
        margin: 0;
        border: none;
        border-radius: 0;
        width: 75%;
        padding: 10px;
        float: left;
        font-size: 16px;
    }

    /* Style the "Add" button */
    .addBtn {
        padding: 10px;
        width: 25%;
        background: #d9d9d9;
        color: #555;
        float: left;
        text-align: center;
        font-size: 16px;
        cursor: pointer;
        transition: 0.3s;
        border-radius: 0;
    }

    .addBtn:hover {
        background-color: #bbb;
    }
</style>
</head>
<body bgcolor="#E6E6FA">

```

```

<div id="s_div" class="status_div">
  <span id="endpointurl"></span>
</div>
<h1 style="text-align: center; margin-top: -50px;"> ConnectOPC-UA </h1>

<div class="sidenav"><a href="index.html">Configure Machine</a><a
href="connect.html">Connect Machine</a> <a href="browse.html"><font color="#FF5733">Browse
Address Space</font></a> <a href="monitor.html">Monitor variables</a><a href="write.html">Write
variables to the server</a><a href="pubsub.html">Create
  Pub-Sub/Notifications</a><a href="pubsub_mult.html">Multiple Pub-Sub/Notifications</a><a
href="method_call.html">Method Call (Address Space)</a><a href="alarm_monitoring.html">Event
Notifier</a><a href="filetransfer.html">File Transfer</a> <a href="opcuaserver.html">Create OPC-UA
Server</a><a href="settings.html">Settings</a><a href="help.html">Help</a></div>
<div class="main">
<h2>Favourites : </h2>
<ul id="dynamic-list" style="list-style-type:none;"></ul>
</div>
<script>
  let apppath
  const {ipcRenderer} = require('electron')
  let fs2 = require('fs');
  let fs3 = require('fs');
  ipcRenderer.send("browse_path_req", "apppath")
  ipcRenderer.on('browse_path_rep', (event, arg) => {
    apppath=arg

    const mrfileContents = fs2.readFileSync(apppath+'opc_conf_data.txt', 'utf8')
    var datamr = JSON.parse(mrfileContents)
    var datamr_mc=datamr.machine_name

    const mrfileContents1 = fs3.readFileSync(apppath+datamr_mc+"_"+"favourites.txt", 'utf8')
    var datamr1 = mrfileContents1
    var data_line=datamr1.split("\n")
    var ul = document.getElementById("dynamic-list");
    for(i=0;i<data_line.length;i++)
    {
      var li = document.createElement("li");
      li.setAttribute('id',data_line[i]);
      li.appendChild(document.createTextNode(data_line[i]));
      ul.appendChild(li);
    }
  })
  var list=document.querySelector('ul');
  /* list.addEventListener('click', function(ev) {
    if (ev.target.tagName === 'LI') {
      ipcRenderer.send("favourite_data_node", ev.target.id)
    }
  }, false);*/
</script>
</body>
</html>

```

8.17 Package.json

```

{
  "name": "connectopc_ua",
  "version": "1.0.0",
  "description": "Implementation of OPC-UA for machines (CNC and PLC based machines)",
  "main": "main.js",
  "scripts": {
    "start": "electron .",
    "pack": "electron-builder --dir",
    "package-mac": "electron-packager . --overwrite --platform=darwin --arch=x64 --prune=true --out=release-builds",
    "package-win": "electron-packager . ConnectOPCUA-app --overwrite --platform=win32 --arch=ia32 --prune=true --out=release-builds",
    "package-linux": "electron-packager . ConnectOPC-UA-app --overwrite --platform=linux --arch=x64 --prune=true --out=release-builds",
    "dist": "electron-builder"
  },
  "repository": {
    "type": "git",
    "url": "opc_ua_machines"
  },
  "keywords": [
    "opc-ua",
    "CNC",
    "PLC",
    "Automation",
    "Industry",
    "4.0"
  ],
  "author": "Bhupendra Kumar Mishra",
  "license": "ISC",
  "dependencies": {
    "async": "^3.1.0",
    "body-parser": "^1.19.0",
    "canvasjs": "^1.8.1",
    "ejs": "^2.6.2",
    "electron-json-storage": "^4.1.7",
    "express": "^4.17.1",
    "jquery": "^3.4.1",
    "json2xls": "^0.1.2",
    "node-opcua": "^2.1.5",
    "node-opcua-file-transfer": "^2.1.5",
    "nodemailer": "^6.3.0",
    "request": "^2.88.0",
    "treeify": "^1.1.0"
  },
  "devDependencies": {
    "electron": "^6.0.3",
    "electron-builder": "^21.2.0",
    "electron-packager": "^14.0.4"
  },
  "build": {
    "appId": "connectopc-ua.id",
    "mac": {
      "category": "public.app-category.productivity"
    }
  }
}

```

```

    },
    "icon": "linux/512X512.png"
  }
}

```

8.18 Project Dependencies

```

connectopc_ua@1.0.0 E:\opc_ua_machines
+-- async@3.1.0
+-- body-parser@1.19.0
| +-- bytes@3.1.0
| +-- content-type@1.0.4
| +-- debug@2.6.9
| | `-- ms@2.0.0
| +-- depd@1.1.2
| +-- http-errors@1.7.2
| | +-- depd@1.1.2 deduped
| | +-- inherits@2.0.3
| | +-- setprototypeof@1.1.1 deduped
| | +-- statuses@1.5.0 deduped
| | `-- toidentifier@1.0.0
| +-- iconv-lite@0.4.24
| | `-- safer-buffer@2.1.2
| +-- on-finished@2.3.0
| | `-- ee-first@1.1.1
| +-- qs@6.7.0
| +-- raw-body@2.4.0
| | +-- bytes@3.1.0 deduped
| | +-- http-errors@1.7.2 deduped
| | +-- iconv-lite@0.4.24 deduped
| | `-- unpipe@1.0.0
| `-- type-is@1.6.18
|   +-- media-typer@0.3.0
|   `-- mime-types@2.1.24 deduped
+-- canvasjs@1.8.1
+-- ejs@2.6.2
+-- electron@6.0.3
| +-- UNMET DEPENDENCY @types/node@10.14.15
| +-- electron-download@4.1.1
| | +-- debug@3.2.6
| | | `-- ms@2.1.2
| | +-- env-paths@1.0.0
| | +-- fs-extra@4.0.3
| | | +-- graceful-fs@4.1.15 deduped
| | | +-- jsonfile@4.0.0 deduped
| | | `-- universalify@0.1.2 deduped
| | +-- minimist@1.2.0
| | +-- nugget@2.0.1
| | | +-- debug@2.6.9 deduped
| | | +-- minimist@1.2.0 deduped
| | | +-- pretty-bytes@1.0.4
| | | +-- get-stdin@4.0.1

```

```

| | | | `-- meow@3.7.0
| | | | +-- camelcase-keys@2.1.0
| | | | | +-- camelcase@2.1.1
| | | | | `-- map-obj@1.0.1 deduped
| | | | +-- decamelize@1.2.0 deduped
| | | | +-- loud-rejection@1.6.0
| | | | | +-- currently-unhandled@0.4.1
| | | | | | `-- array-find-index@1.0.2
| | | | | `-- signal-exit@3.0.2 deduped
| | | | +-- map-obj@1.0.1
| | | | +-- minimist@1.2.0 deduped
| | | | +-- normalize-package-data@2.5.0 deduped
| | | | +-- object-assign@4.1.1
| | | | +-- read-pkg-up@1.0.1
| | | | | +-- find-up@1.1.2
| | | | | | +-- path-exists@2.1.0
| | | | | | | `-- pinkie-promise@2.0.1 deduped
| | | | | | `-- pinkie-promise@2.0.1
| | | | | | | `-- pinkie@2.0.4
| | | | | `-- read-pkg@1.1.0
| | | | | | +-- load-json-file@1.1.0
| | | | | | | +-- graceful-fs@4.1.15 deduped
| | | | | | | | +-- parse-json@2.2.0 deduped
| | | | | | | | +-- pify@2.3.0
| | | | | | | | +-- pinkie-promise@2.0.1 deduped
| | | | | | | | `-- strip-bom@2.0.0
| | | | | | | | `-- is-utf8@0.2.1
| | | | | | +-- normalize-package-data@2.5.0 deduped
| | | | | | `-- path-type@1.1.0
| | | | | | | +-- graceful-fs@4.1.15 deduped
| | | | | | | +-- pify@2.3.0 deduped
| | | | | | `-- pinkie-promise@2.0.1 deduped
| | | | +-- redent@1.0.0
| | | | | +-- indent-string@2.1.0
| | | | | | `-- repeating@2.0.1
| | | | | | `-- is-finite@1.0.2
| | | | | | | `-- number-is-nan@1.0.1 deduped
| | | | | `-- strip-indent@1.0.1
| | | | | `-- get-stdin@4.0.1 deduped
| | | | `-- trim-newlines@1.0.0
| | | +-- progress-stream@1.2.0
| | | +-- speedometer@0.1.4
| | | `-- through2@0.2.3
| | | | +-- readable-stream@1.1.14
| | | | | +-- core-util-is@1.0.2 deduped
| | | | | +-- inherits@2.0.3 deduped
| | | | | +-- isarray@0.0.1
| | | | | `-- string_decoder@0.10.31
| | | | `-- xtend@2.1.2
| | | | | `-- object-keys@0.4.0
| | | +-- request@2.88.0 deduped
| | | +-- single-line-log@1.1.2
| | | | `-- string-width@1.0.2
| | | | +-- code-point-at@1.1.0
| | | | +-- is-fullwidth-code-point@1.0.0
| | | | | `-- number-is-nan@1.0.1

```

```

| | | `-- strip-ansi@3.0.1
| | | `-- ansi-regex@2.1.1
| | `-- throttleit@0.0.2
| | +- path-exists@3.0.0
| | +- rc@1.2.8
| | +- deep-extend@0.6.0
| | +- ini@1.3.5
| | +- minimist@1.2.0 deduped
| | `-- strip-json-comments@2.0.1
| +- semver@5.7.0
| `-- sumchecker@2.0.2
| `-- debug@2.6.9 deduped
|-- extract-zip@1.6.7
| +- concat-stream@1.6.2
| | +- buffer-from@1.1.1
| | +- inherits@2.0.3 deduped
| | +- readable-stream@2.3.6
| | | +- core-util-is@1.0.2
| | | +- inherits@2.0.3 deduped
| | | +- isarray@1.0.0
| | | +- process-nexttick-args@2.0.1
| | | +- safe-buffer@5.1.2 deduped
| | | +- string_decoder@1.1.1
| | | | `-- safe-buffer@5.1.2 deduped
| | | `-- util-deprecate@1.0.2
| | `-- typedarray@0.0.6
| +- debug@2.6.9 deduped
| +- mkdirp@0.5.1 deduped
| `-- yauzl@2.4.1
| `-- fd-slicer@1.0.1
| `-- pend@1.2.0
+-- electron-builder@21.2.0
| +- app-builder-lib@21.2.0
| | +- 7zip-bin@5.0.3
| | +- @develar/schema-utils@2.1.0
| | | +- ajv@6.10.0 deduped
| | | | `-- ajv-keywords@3.4.1
| | | +- async-exit-hook@2.0.1
| | | +- bluebird-lst@1.0.9 deduped
| | | +- builder-util@21.2.0 deduped
| | | +- builder-util-runtime@8.3.0 deduped
| | | +- chromium-pickle-js@0.2.0
| | | +- debug@4.1.1
| | | | `-- ms@2.1.2
| | | +- ejs@2.6.2 deduped
| | | +- electron-publish@21.2.0
| | | | +- bluebird-lst@1.0.9 deduped
| | | | +- builder-util@21.2.0 deduped
| | | | +- builder-util-runtime@8.3.0 deduped
| | | | +- chalk@2.4.2 deduped
| | | | +- fs-extra@8.1.0
| | | | | +- graceful-fs@4.2.2
| | | | | +- jsonfile@4.0.0 deduped
| | | | | | `-- universalify@0.1.2 deduped
| | | | | +- lazy-val@1.0.4 deduped
| | | | | `-- mime@2.4.4

```



```

| | +-- fs-extra@8.1.0
| | | +-- graceful-fs@4.2.2
| | | +-- jsonfile@4.0.0 deduped
| | | `-- universalify@0.1.2 deduped
| | +-- hosted-git-info@2.7.1
| | +-- is-ci@2.0.0 deduped
| | +-- isbinaryfile@4.0.2
| | +-- js-yaml@3.13.1
| | | +-- argparse@1.0.10
| | | | `-- sprintf-js@1.0.3
| | | `-- esprima@4.0.1
| | +-- lazy-val@1.0.4 deduped
| | +-- minimatch@3.0.4
| | | `-- brace-expansion@1.1.11
| | | +-- balanced-match@1.0.0
| | | `-- concat-map@0.0.1
| | +-- normalize-package-data@2.5.0
| | | +-- hosted-git-info@2.7.1 deduped
| | | +-- resolve@1.11.0 deduped
| | | +-- semver@5.7.0 deduped
| | | `-- validate-npm-package-license@3.0.4
| | | +-- spdx-correct@3.1.0
| | | | +-- spdx-expression-parse@3.0.0 deduped
| | | | | `-- spdx-license-ids@3.0.4
| | | | `-- spdx-expression-parse@3.0.0
| | | | +-- spdx-exceptions@2.2.0
| | | | `-- spdx-license-ids@3.0.4 deduped
| | +-- read-config-file@5.0.0 deduped
| | +-- sanitize-filename@1.6.2
| | | `-- truncate-utf8-bytes@1.0.2 deduped
| | +-- semver@6.3.0
| | `-- temp-file@3.3.4
| | +-- async-exit-hook@2.0.1 deduped
| | `-- fs-extra@8.1.0
| |   +-- graceful-fs@4.2.2
| |   +-- jsonfile@4.0.0 deduped
| |   `-- universalify@0.1.2 deduped
| +-- bluebird-lst@1.0.9
| | `-- bluebird@3.5.5
| +-- builder-util@21.2.0
| | +-- 7zip-bin@5.0.3 deduped
| | +-- @types/debug@4.1.5
| | +-- app-builder-bin@3.4.3
| | +-- bluebird-lst@1.0.9 deduped
| | +-- builder-util-runtime@8.3.0 deduped
| | +-- chalk@2.4.2 deduped
| | +-- debug@4.1.1
| | | `-- ms@2.1.2
| | +-- fs-extra@8.1.0
| | | +-- graceful-fs@4.2.2
| | | +-- jsonfile@4.0.0 deduped
| | | `-- universalify@0.1.2 deduped
| | +-- is-ci@2.0.0 deduped
| | +-- js-yaml@3.13.1 deduped
| | +-- source-map-support@0.5.13
| | | +-- buffer-from@1.1.1 deduped

```

```

| | | `-- source-map@0.6.1
| | +- stat-mode@0.3.0
| | `-- temp-file@3.3.4 deduped
| +- builder-util-runtime@8.3.0
| | +- debug@4.1.1
| | | `-- ms@2.1.2
| | `-- sax@1.2.4
| +- chalk@2.4.2
| | +- ansi-styles@3.2.1
| | | `-- color-convert@1.9.3
| | |   `-- color-name@1.1.3
| | +- escape-string-regexp@1.0.5
| | `-- supports-color@5.5.0
| |   `-- has-flag@3.0.0
| +- dmg-builder@21.2.0
| | +- app-builder-lib@21.2.0 deduped
| | +- bluebird-lst@1.0.9 deduped
| | +- builder-util@21.2.0 deduped
| | +- fs-extra@8.1.0
| | | +- graceful-fs@4.2.2
| | | +- jsonfile@4.0.0 deduped
| | | `-- universalify@0.1.2 deduped
| | +- iconv-lite@0.5.0
| | | `-- safer-buffer@2.1.2 deduped
| | +- js-yaml@3.13.1 deduped
| | `-- sanitize-filename@1.6.2
| |   `-- truncate-utf8-bytes@1.0.2 deduped
| +- fs-extra@8.1.0
| | +- graceful-fs@4.2.2
| | +- jsonfile@4.0.0
| | | `-- graceful-fs@4.1.15 deduped
| | `-- universalify@0.1.2
| +- is-ci@2.0.0
| | `-- ci-info@2.0.0
| +- lazy-val@1.0.4
| +- read-config-file@5.0.0
| | +- dotenv@8.1.0
| | +- dotenv-expand@5.1.0
| | +- fs-extra@8.1.0
| | | +- graceful-fs@4.2.2
| | | +- jsonfile@4.0.0 deduped
| | | `-- universalify@0.1.2 deduped
| | +- js-yaml@3.13.1 deduped
| | +- json5@2.1.0
| | | `-- minimist@1.2.0 deduped
| | `-- lazy-val@1.0.4 deduped
| +- sanitize-filename@1.6.2
| | `-- truncate-utf8-bytes@1.0.2
| |   `-- utf8-byte-length@1.0.4
| +- update-notifier@3.0.1
| | +- boxen@3.2.0
| | | +- ansi-align@3.0.0
| | | | `-- string-width@3.1.0
| | | |   +- emoji-regex@7.0.3 deduped
| | | |   +- is-fullwidth-code-point@2.0.0
| | | |   `-- strip-ansi@5.2.0

```

```

| | | | `-- ansi-regex@4.1.0
| | | +- camelcase@5.3.1
| | | +- chalk@2.4.2 deduped
| | | +- cli-boxes@2.2.0
| | | +- string-width@3.1.0
| | | +- emoji-regex@7.0.3 deduped
| | | +- is-fullwidth-code-point@2.0.0
| | | `-- strip-ansi@5.2.0
| | | `-- ansi-regex@4.1.0
| | | +- term-size@1.2.0
| | | `-- execa@0.7.0
| | | +- cross-spawn@5.1.0
| | | | +- lru-cache@4.1.5
| | | | | +- pseudomap@1.0.2
| | | | | `-- yallist@2.1.2
| | | | +- shebang-command@1.2.0
| | | | | `-- shebang-regex@1.0.0
| | | | `-- which@1.3.1
| | | | `-- isexe@2.0.0
| | | +- get-stream@3.0.0
| | | +- is-stream@1.1.0
| | | +- npm-run-path@2.0.2
| | | | `-- path-key@2.0.1
| | | +- p-finally@1.0.0
| | | +- signal-exit@3.0.2 deduped
| | | `-- strip-eof@1.0.0
| | +- type-fest@0.3.1
| | `-- widest-line@2.0.1
| | `-- string-width@2.1.1
| | +- is-fullwidth-code-point@2.0.0
| | `-- strip-ansi@4.0.0
| | `-- ansi-regex@3.0.0
| +- chalk@2.4.2 deduped
| +- configstore@4.0.0
| | +- dot-prop@4.2.0
| | | `-- is-obj@1.0.1
| | +- graceful-fs@4.1.15 deduped
| | +- make-dir@1.3.0
| | | `-- pify@3.0.0
| | +- unique-string@1.0.0
| | | `-- crypto-random-string@1.0.0
| | +- write-file-atomic@2.4.3 deduped
| | `-- xdg-basedir@3.0.0 deduped
| +- has-yarn@2.1.0
| +- import-lazy@2.1.0
| +- is-ci@2.0.0 deduped
| +- is-installed-globally@0.1.0
| | +- global-dirs@0.1.1
| | | `-- ini@1.3.5 deduped
| | | `-- is-path-inside@1.0.1
| | | `-- path-is-inside@1.0.2
| | +- is-npm@3.0.0
| | +- is-yarn-global@0.3.0
| | +- latest-version@5.1.0
| | | `-- package-json@6.5.0
| | | +- got@9.6.0 deduped

```

```

| | | +-- registry-auth-token@4.0.0
| | | | +-- rc@1.2.8 deduped
| | | | `-- safe-buffer@5.1.2 deduped
| | | +-- registry-url@5.1.0
| | | | `-- rc@1.2.8 deduped
| | | `-- semver@6.3.0
| | +-- semver-diff@2.1.0
| | | `-- semver@5.7.0 deduped
| | `-- xdg-basedir@3.0.0
| `-- yargs@13.3.0
|   +-- cliui@5.0.0
|   | +-- string-width@3.1.0
|   | | +-- emoji-regex@7.0.3 deduped
|   | | +-- is-fullwidth-code-point@2.0.0
|   | | | `-- strip-ansi@5.2.0 deduped
|   | | +-- strip-ansi@5.2.0
|   | | | `-- ansi-regex@4.1.0
|   | | | `-- wrap-ansi@5.1.0
|   | | +-- ansi-styles@3.2.1 deduped
|   | | +-- string-width@3.1.0
|   | | | +-- emoji-regex@7.0.3 deduped
|   | | | +-- is-fullwidth-code-point@2.0.0
|   | | | | `-- strip-ansi@5.2.0 deduped
|   | | | | `-- strip-ansi@5.2.0
|   | | | | `-- ansi-regex@4.1.0
|   | +-- find-up@3.0.0
|   | | `-- locate-path@3.0.0
|   | | +-- p-locate@3.0.0
|   | | | `-- p-limit@2.2.0
|   | | | | `-- p-try@2.2.0
|   | | | `-- path-exists@3.0.0 deduped
|   +-- get-caller-file@2.0.5
|   +-- require-directory@2.1.1
|   +-- require-main-filename@2.0.0
|   +-- set-blocking@2.0.0
|   +-- string-width@3.1.0
|   | +-- emoji-regex@7.0.3
|   | +-- is-fullwidth-code-point@2.0.0
|   | | `-- strip-ansi@5.2.0
|   | | | `-- ansi-regex@4.1.0
|   +-- which-module@2.0.0
|   +-- y18n@4.0.0
|   `-- yargs-parser@13.1.1 deduped
+-- electron-json-storage@4.1.7
| +-- async@2.6.3
| | `-- lodash@4.17.15 deduped
| +-- lockfile@1.0.4
| | `-- signal-exit@3.0.2
| +-- lodash@4.17.15
| +-- mkdirp@0.5.1
| | `-- minimist@0.0.8
| +-- rimraf@2.6.3
| | `-- glob@7.1.4
| | +-- fs.realpath@1.0.0
| | +-- inflight@1.0.6
| | | +-- once@1.4.0 deduped

```

```

| | | `-- wrappy@1.0.2 deduped
| | | +-- inherits@2.0.3 deduped
| | | +-- minimatch@3.0.4 deduped
| | | +-- once@1.4.0 deduped
| | | `-- path-is-absolute@1.0.1
| | `-- write-file-atomic@2.4.3
| | +-- graceful-fs@4.1.15
| | +-- imurmurhash@0.1.4
| | `-- signal-exit@3.0.2 deduped
+-- electron-packager@14.0.4
| +-- @electron/get@1.4.0
| | +-- debug@4.1.1
| | | `-- ms@2.1.2
| | +-- env-paths@2.2.0
| | +-- fs-extra@8.1.0
| | | +-- graceful-fs@4.2.1
| | | +-- jsonfile@4.0.0 deduped
| | | `-- universalify@0.1.2 deduped
| | +-- got@9.6.0
| | | +-- @sindresorhus/is@0.14.0
| | | +-- @szmarczak/http-timer@1.1.2
| | | | `-- defer-to-connect@1.0.2
| | | +-- cacheable-request@6.1.0
| | | | +-- clone-response@1.0.2
| | | | | `-- mimic-response@1.0.1 deduped
| | | | +-- get-stream@5.1.0
| | | | | `-- pump@3.0.0 deduped
| | | | +-- http-cache-semantics@4.0.3
| | | | +-- keyv@3.1.0
| | | | | `-- json-buffer@3.0.0
| | | | +-- lowercase-keys@2.0.0
| | | | +-- normalize-url@4.3.0
| | | | `-- responselike@1.0.2
| | | | `-- lowercase-keys@1.0.1 deduped
| | | +-- decompress-response@3.3.0
| | | | `-- mimic-response@1.0.1 deduped
| | | +-- duplex3@0.1.4
| | | +-- get-stream@4.1.0
| | | | `-- pump@3.0.0
| | | | +-- end-of-stream@1.4.1
| | | | | `-- once@1.4.0 deduped
| | | | `-- once@1.4.0 deduped
| | | +-- lowercase-keys@1.0.1
| | | +-- mimic-response@1.0.1
| | | +-- p-cancelable@1.1.0
| | | +-- to-readable-stream@1.0.0
| | | | `-- url-parse-lax@3.0.0
| | | | `-- prepend-http@2.0.0
| | | `-- sumchecker@3.0.0
| | `-- debug@4.1.1 deduped
+-- asar@2.0.1
| +-- chromium-pickle-js@0.2.0 deduped
| +-- commander@2.20.0
| +-- cuint@0.2.2
| +-- glob@7.1.4 deduped
| +-- minimatch@3.0.4 deduped

```

```

| | +-- mkdirp@0.5.1 deduped
| | `-- tmp-promise@1.1.0
| | +-- bluebird@3.5.5 deduped
| | `-- tmp@0.1.0
| | `-- rimraf@2.6.3 deduped
| +-- cross-zip@2.1.5
| | `-- rimraf@2.6.3 deduped
| +-- debug@4.1.1
| | `-- ms@2.1.2
| +-- electron-notarize@0.1.1
| | +-- debug@4.1.1
| | | `-- ms@2.1.2
| | `-- fs-extra@8.1.0
| | +-- graceful-fs@4.2.1
| | +-- jsonfile@4.0.0 deduped
| | `-- universalify@0.1.2 deduped
| +-- electron-osx-sign@0.4.11
| | +-- bluebird@3.5.5 deduped
| | +-- compare-version@0.1.2
| | +-- debug@2.6.9 deduped
| | +-- isbinaryfile@3.0.3
| | | `-- buffer-alloc@1.2.0
| | | +-- buffer-alloc-unsafe@1.1.0
| | | `-- buffer-fill@1.0.0
| | +-- minimist@1.2.0 deduped
| | `-- plist@3.0.1 deduped
| +-- fs-extra@8.1.0
| | +-- graceful-fs@4.2.1
| | +-- jsonfile@4.0.0 deduped
| | `-- universalify@0.1.2 deduped
| +-- galactus@0.2.1
| | +-- debug@3.2.6
| | | `-- ms@2.1.2
| | +-- flora-colossus@1.0.1
| | | +-- debug@4.1.1
| | | | `-- ms@2.1.2
| | | `-- fs-extra@7.0.1
| | | +-- graceful-fs@4.1.15 deduped
| | | +-- jsonfile@4.0.0 deduped
| | | `-- universalify@0.1.2 deduped
| | `-- fs-extra@4.0.3 deduped
| +-- get-package-info@1.0.0
| | +-- bluebird@3.5.5 deduped
| | +-- debug@2.6.9 deduped
| | +-- lodash.get@4.4.2
| | `-- read-pkg-up@2.0.0
| | +-- find-up@2.1.0
| | | `-- locate-path@2.0.0
| | | +-- p-locate@2.0.0
| | | | `-- p-limit@1.3.0
| | | | `-- p-try@1.0.0
| | | `-- path-exists@3.0.0 deduped
| | `-- read-pkg@2.0.0
| | +-- load-json-file@2.0.0
| | | +-- graceful-fs@4.1.15 deduped
| | | +-- parse-json@2.2.0

```

```

| | | | `-- error-ex@1.3.2
| | | | `-- is-arrayish@0.2.1
| | | | +-- pify@2.3.0
| | | | `-- strip-bom@3.0.0
| | | +-- normalize-package-data@2.5.0 deduped
| | | `-- path-type@2.0.0
| | | `-- pify@2.3.0 deduped
| +-- junk@3.1.0
| +-- parse-author@2.0.0
| | `-- author-regex@1.0.0
| +-- plist@3.0.1
| | +-- base64-js@1.3.0
| | +-- xmlbuilder@9.0.7
| | `-- xmldom@0.1.27
| +-- rcredit@2.0.0
| +-- resolve@1.11.0
| | `-- path-parse@1.0.6
| +-- sanitize-filename@1.6.1
| | `-- truncate-utf8-bytes@1.0.2 deduped
| +-- semver@6.3.0
| `-- yargs-parser@13.1.1
|   +-- camelcase@5.3.1
|   `-- decamelize@1.2.0
+-- express@4.17.1
| +-- accepts@1.3.7
| | +-- mime-types@2.1.24 deduped
| | `-- negotiator@0.6.2
| +-- array-flatten@1.1.1
| +-- body-parser@1.19.0 deduped
| +-- content-disposition@0.5.3
| | `-- safe-buffer@5.1.2 deduped
| +-- content-type@1.0.4 deduped
| +-- cookie@0.4.0
| +-- cookie-signature@1.0.6
| +-- debug@2.6.9 deduped
| +-- depd@1.1.2 deduped
| +-- encodeurl@1.0.2
| +-- escape-html@1.0.3
| +-- etag@1.8.1
| +-- finalhandler@1.1.2
| | +-- debug@2.6.9 deduped
| | +-- encodeurl@1.0.2 deduped
| | +-- escape-html@1.0.3 deduped
| | +-- on-finished@2.3.0 deduped
| | +-- parseurl@1.3.3 deduped
| | +-- statuses@1.5.0 deduped
| | `-- unpipe@1.0.0 deduped
| +-- fresh@0.5.2
| +-- merge-descriptors@1.0.1
| +-- methods@1.1.2
| +-- on-finished@2.3.0 deduped
| +-- parseurl@1.3.3
| +-- path-to-regexp@0.1.7
| +-- proxy-addr@2.0.5
| | +-- forwarded@0.1.2
| | `-- ipaddr.js@1.9.0

```

```

| +- qs@6.7.0 deduped
| +- range-parser@1.2.1
| +- safe-buffer@5.1.2
| +- send@0.17.1
| | +- debug@2.6.9 deduped
| | +- depd@1.1.2 deduped
| | +- destroy@1.0.4
| | +- encodeurl@1.0.2 deduped
| | +- escape-html@1.0.3 deduped
| | +- etag@1.8.1 deduped
| | +- fresh@0.5.2 deduped
| | +- http-errors@1.7.2 deduped
| | +- mime@1.6.0
| | +- ms@2.1.1
| | +- on-finished@2.3.0 deduped
| | +- range-parser@1.2.1 deduped
| | `-- statuses@1.5.0 deduped
| +- serve-static@1.14.1
| | +- encodeurl@1.0.2 deduped
| | +- escape-html@1.0.3 deduped
| | +- parseurl@1.3.3 deduped
| | `-- send@0.17.1 deduped
| +- setprototypeof@1.1.1
| +- statuses@1.5.0
| +- type-is@1.6.18 deduped
| +- utils-merge@1.0.1
| `-- vary@1.1.2
+- jquery@3.4.1
+- json2xls@0.1.2
| `-- excel-export@0.3.11
| `-- node-zip@1.1.1
| `-- jszip@2.5.0
| `-- pako@0.2.8
+- node-opcua@2.1.5
| +- chalk@2.4.2 deduped
| +- node-opcua-address-space@2.1.5
| | +- async@3.1.0 deduped
| | +- dequeue@1.0.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-client-dynamic-extension-object@2.1.5
| | | +- chalk@2.4.2 deduped
| | | | +- node-opcua-assert@2.0.0 deduped
| | | | | +- node-opcua-binary-stream@2.1.0 deduped
| | | | | +- node-opcua-data-model@2.1.5 deduped
| | | | | +- node-opcua-data-value@2.1.5 deduped
| | | | | +- node-opcua-debug@2.1.0 deduped
| | | | | | +- node-opcua-extension-object@2.1.5 deduped
| | | | | | +- node-opcua-factory@2.1.5 deduped
| | | | | | +- node-opcua-nodeid@2.1.5 deduped
| | | | | | +- node-opcua-pseudo-session@2.1.5 deduped
| | | | | | +- node-opcua-schemas@2.1.5 deduped
| | | | | | +- node-opcua-service-browse@2.1.5 deduped
| | | | | | +- node-opcua-status-code@2.1.5 deduped
| | | | | | `-- node-opcua-variant@2.1.5 deduped
| | | +- node-opcua-common@2.1.5 deduped

```



```

| | +- node-opcua-constants@2.1.5 deduped
| | +- node-opcua-crypto@1.1.2 deduped
| | +- node-opcua-data-access@2.1.5 deduped
| | +- node-opcua-data-model@2.1.5 deduped
| | +- node-opcua-data-value@2.1.5 deduped
| | +- node-opcua-date-time@2.1.5
| | | +- long@4.0.0
| | | +- node-opcua-assert@2.0.0 deduped
| | | `-- underscore@1.9.1 deduped
| | +- node-opcua-debug@2.1.0 deduped
| | +- node-opcua-enum@2.1.0 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-numeric-range@2.1.5 deduped
| | +- node-opcua-object-registry@2.1.0
| | | +- node-opcua-assert@2.0.0 deduped
| | | +- node-opcua-debug@2.1.0 deduped
| | | `-- underscore@1.9.1 deduped
| | +- node-opcua-pseudo-session@2.1.5
| | | +- async@3.1.0 deduped
| | | +- node-opcua-data-model@2.1.5 deduped
| | | +- node-opcua-data-value@2.1.5 deduped
| | | +- node-opcua-nodeid@2.1.5 deduped
| | | +- node-opcua-service-browse@2.1.5 deduped
| | | +- node-opcua-service-call@2.1.5 deduped
| | | +- node-opcua-service-read@2.1.5 deduped
| | | +- node-opcua-service-translate-browse-path@2.1.5 deduped
| | | +- node-opcua-status-code@2.1.5 deduped
| | | +- node-opcua-utils@2.1.5 deduped
| | | +- node-opcua-variant@2.1.5 deduped
| | | `-- underscore@1.9.1 deduped
| | +- node-opcua-schemas@2.1.5
| | | +- node-opcua-assert@2.0.0 deduped
| | | +- node-opcua-binary-stream@2.1.0 deduped
| | | +- node-opcua-data-model@2.1.5 deduped
| | | +- node-opcua-debug@2.1.0 deduped
| | | +- node-opcua-enum@2.1.0 deduped
| | | +- node-opcua-extension-object@2.1.5 deduped
| | | +- node-opcua-factory@2.1.5 deduped
| | | +- node-opcua-nodeid@2.1.5 deduped
| | | +- node-opcua-variant@2.1.5 deduped
| | | `-- node-opcua-xml2json@2.1.5 deduped
| | +- node-opcua-service-browse@2.1.5 deduped
| | +- node-opcua-service-call@2.1.5 deduped
| | +- node-opcua-service-filter@2.1.5 deduped
| | +- node-opcua-service-history@2.1.5 deduped
| | +- node-opcua-service-session@2.1.5 deduped
| | +- node-opcua-service-translate-browse-path@2.1.5 deduped
| | +- node-opcua-service-write@2.1.5 deduped
| | +- node-opcua-status-code@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | +- node-opcua-utils@2.1.5 deduped
| | +- node-opcua-variant@2.1.5 deduped
| | +- node-opcua-xml2json@2.1.5
| | | +- bomstrip@0.1.4
| | | +- ltx@2.8.1

```

```

| | | |-- inherits@2.0.3 deduped
| | | +-- node-opcua-assert@2.0.0 deduped
| | | +-- node-opcua-debug@2.1.0 deduped
| | | +-- node-opcua-utils@2.1.5 deduped
| | | |-- underscore@1.9.1 deduped
| | | +-- object.values@1.1.0
| | | +-- define-properties@1.1.3
| | | |-- object-keys@1.1.1
| | | +-- es-abstract@1.13.0
| | | +-- es-to-primitive@1.2.0
| | | | +-- is-callable@1.1.4 deduped
| | | | +-- is-date-object@1.0.1
| | | | |-- is-symbol@1.0.2
| | | | |-- has-symbols@1.0.0
| | | | +-- function-bind@1.1.1 deduped
| | | | +-- has@1.0.3 deduped
| | | | +-- is-callable@1.1.4
| | | | +-- is-regex@1.0.4
| | | | |-- has@1.0.3 deduped
| | | | |-- object-keys@1.1.1
| | | | +-- function-bind@1.1.1
| | | | |-- has@1.0.3
| | | |-- function-bind@1.1.1 deduped
| | | +-- set-prototype-of@1.0.0
| | | +-- underscore@1.9.1
| | |-- xml-writer@1.7.0
| +-- node-opcua-address-space-for-conformance-testing@2.1.5
| | +-- node-opcua-address-space@2.1.5 deduped
| | +-- node-opcua-assert@2.0.0 deduped
| | +-- node-opcua-basic-types@2.1.5 deduped
| | +-- node-opcua-data-access@2.1.5 deduped
| | +-- node-opcua-data-model@2.1.5 deduped
| | +-- node-opcua-data-value@2.1.5 deduped
| | +-- node-opcua-factory@2.1.5 deduped
| | +-- node-opcua-nodeid@2.1.5 deduped
| | +-- node-opcua-status-code@2.1.5 deduped
| | +-- node-opcua-variant@2.1.5 deduped
| |-- underscore@1.9.1 deduped
| +-- node-opcua-assert@2.0.0
| | +-- better-assert@1.0.2
| | |-- callsite@1.0.0
| | |-- chalk@2.4.2 deduped
| +-- node-opcua-basic-types@2.1.5
| | +-- node-opcua-assert@2.0.0 deduped
| | +-- node-opcua-binary-stream@2.1.0
| | | +-- colors@1.3.3 deduped
| | | +-- node-opcua-assert@2.0.0 deduped
| | | +-- node-opcua-buffer-utils@2.1.0 deduped
| | |-- underscore@1.9.1 deduped
| | +-- node-opcua-buffer-utils@2.1.0
| | |-- node-opcua-assert@2.0.0 deduped
| | +-- node-opcua-date-time@2.1.5 deduped
| | +-- node-opcua-enum@2.1.0 deduped
| | +-- node-opcua-guid@2.1.0
| | |-- node-opcua-assert@2.0.0 deduped
| | +-- node-opcua-nodeid@2.1.5 deduped

```

```

| | +-- node-opcua-status-code@2.1.5 deduped
| | +-- node-opcua-utils@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +-- node-opcua-certificate-manager@2.1.5
| | +-- @types/mkdirp@0.5.2
| | | `-- @types/node@12.7.1
| | +-- async@3.1.0 deduped
| | +-- chalk@2.4.2 deduped
| | +-- delayed@2.0.0
| | +-- env-paths@2.2.0
| | +-- mkdirp@0.5.1 deduped
| | +-- node-opcua-assert@2.0.0 deduped
| | +-- node-opcua-basic-types@2.1.5 deduped
| | +-- node-opcua-buffer-utils@2.1.0 deduped
| | +-- node-opcua-common@2.1.5 deduped
| | +-- node-opcua-constants@2.1.5 deduped
| | +-- node-opcua-crypto@1.1.2 deduped
| | +-- node-opcua-debug@2.1.0 deduped
| | +-- node-opcua-pki@1.5.3 deduped
| | +-- node-opcua-status-code@2.1.5 deduped
| | +-- node-opcua-utils@2.1.5 deduped
| | +-- once@1.4.0
| | | `-- wrappy@1.0.2
| | +-- thenify@3.3.0
| | | `-- any-promise@1.3.0
| | `-- underscore@1.9.1 deduped
| +-- node-opcua-client@2.1.5
| | +-- @types/async@3.0.1
| | +-- @types/once@1.4.0
| | +-- @types/underscore@1.9.2
| | +-- async@3.1.0 deduped
| | +-- callbackify@1.1.0
| | +-- chalk@2.4.2 deduped
| | +-- delayed@2.0.0 deduped
| | +-- node-opcua-assert@2.0.0 deduped
| | +-- node-opcua-basic-types@2.1.5 deduped
| | +-- node-opcua-buffer-utils@2.1.0 deduped
| | +-- node-opcua-certificate-manager@2.1.5 deduped
| | +-- node-opcua-client-dynamic-extension-object@2.1.5 deduped
| | +-- node-opcua-common@2.1.5 deduped
| | +-- node-opcua-constants@2.1.5 deduped
| | +-- node-opcua-crypto@1.1.2 deduped
| | +-- node-opcua-data-model@2.1.5 deduped
| | +-- node-opcua-data-value@2.1.5 deduped
| | +-- node-opcua-date-time@2.1.5 deduped
| | +-- node-opcua-debug@2.1.0 deduped
| | +-- node-opcua-extension-object@2.1.5
| | | +-- chalk@2.4.2 deduped
| | | +-- node-opcua-assert@2.0.0 deduped
| | | +-- node-opcua-basic-types@2.1.5 deduped
| | | +-- node-opcua-binary-stream@2.1.0 deduped
| | | +-- node-opcua-debug@2.1.0 deduped
| | | +-- node-opcua-factory@2.1.5 deduped
| | | | `-- node-opcua-nodeid@2.1.5 deduped
| | | +-- node-opcua-hostname@2.1.0 deduped
| | | +-- node-opcua-nodeid@2.1.5 deduped

```

```

| | +- node-opcua-object-registry@2.1.0 deduped
| | +- node-opcua-pki@1.5.3 deduped
| | +- node-opcua-pseudo-session@2.1.5 deduped
| | +- node-opcua-schemas@2.1.5 deduped
| | +- node-opcua-secure-channel@2.1.5 deduped
| | +- node-opcua-service-browse@2.1.5 deduped
| | +- node-opcua-service-call@2.1.5 deduped
| | +- node-opcua-service-discovery@2.1.5 deduped
| | +- node-opcua-service-endpoints@2.1.5 deduped
| | +- node-opcua-service-filter@2.1.5 deduped
| | +- node-opcua-service-history@2.1.5 deduped
| | +- node-opcua-service-query@2.1.5 deduped
| | +- node-opcua-service-read@2.1.5 deduped
| | +- node-opcua-service-register-node@2.1.5 deduped
| | +- node-opcua-service-secure-channel@2.1.5 deduped
| | +- node-opcua-service-session@2.1.5 deduped
| | +- node-opcua-service-subscription@2.1.5 deduped
| | +- node-opcua-service-translate-browse-path@2.1.5 deduped
| | +- node-opcua-service-write@2.1.5 deduped
| | +- node-opcua-status-code@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | +- node-opcua-utils@2.1.5 deduped
| | +- node-opcua-variant@2.1.5 deduped
| | +- once@1.4.0 deduped
| | +- thenify@3.3.0 deduped
| | `-- underscore@1.9.1 deduped
+- node-opcua-client-crawler@2.1.5
| | +- async@3.1.0 deduped
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-common@2.1.5 deduped
| | +- node-opcua-constants@2.1.5 deduped
| | +- node-opcua-data-model@2.1.5 deduped
| | +- node-opcua-data-value@2.1.5 deduped
| | +- node-opcua-debug@2.1.0 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-service-browse@2.1.5 deduped
| | +- node-opcua-status-code@2.1.5 deduped
| | +- node-opcua-utils@2.1.5 deduped
| | +- node-opcua-variant@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
+- node-opcua-client-proxy@2.1.5
| | +- async@3.1.0 deduped
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-client@2.1.5 deduped
| | +- node-opcua-constants@2.1.5 deduped
| | +- node-opcua-data-model@2.1.5 deduped
| | +- node-opcua-data-value@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-service-browse@2.1.5 deduped
| | +- node-opcua-service-call@2.1.5 deduped
| | +- node-opcua-service-read@2.1.5 deduped
| | +- node-opcua-service-subscription@2.1.5 deduped
| | +- node-opcua-service-write@2.1.5 deduped
| | +- node-opcua-status-code@2.1.5 deduped
| | +- node-opcua-utils@2.1.5 deduped

```

```

| | +- node-opcua-variant@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +- node-opcua-common@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-crypto@1.1.2 deduped
| | +- node-opcua-data-model@2.1.5 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-service-endpoints@2.1.5 deduped
| | +- node-opcua-service-secure-channel@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +- node-opcua-constants@2.1.5
| +- node-opcua-crypto@1.1.2
| | +- @types/lorem-ipsu@1.0.2
| | +- better-assert@1.0.2 deduped
| | +- chalk@2.4.2 deduped
| | +- hexy@0.3.0
| | +- jsrsasign@8.0.12
| | +- ssh-key-to-pem@0.11.1
| | | +- asn1@0.1.11
| | | | `-- ctype@0.5.2
| | | `-- underscore@1.9.1 deduped
| +- node-opcua-data-access@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-data-model@2.1.5 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +- node-opcua-data-model@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-enum@2.1.0 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-status-code@2.1.5 deduped
| | +- node-opcua-utils@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +- node-opcua-data-value@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-data-model@2.1.5 deduped
| | +- node-opcua-date-time@2.1.5 deduped
| | +- node-opcua-enum@2.1.0 deduped
| | +- node-opcua-extension-object@2.1.5 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-status-code@2.1.5 deduped
| | +- node-opcua-utils@2.1.5 deduped
| | +- node-opcua-variant@2.1.5 deduped

```

```

| | `-- underscore@1.9.1 deduped
| +-- node-opcua-debug@2.1.0
| | +-- chalk@2.4.2 deduped
| | +-- hexy@0.3.0 deduped
| | +-- node-opcua-assert@2.0.0 deduped
| | +-- node-opcua-buffer-utils@2.1.0 deduped
| | `-- underscore@1.9.1 deduped
| +-- node-opcua-enum@2.1.0
| | `-- node-opcua-assert@2.0.0 deduped
| +-- node-opcua-factory@2.1.5
| | +-- chalk@2.4.2 deduped
| | +-- node-opcua-assert@2.0.0 deduped
| | +-- node-opcua-basic-types@2.1.5 deduped
| | +-- node-opcua-binary-stream@2.1.0 deduped
| | +-- node-opcua-enum@2.1.0 deduped
| | +-- node-opcua-guid@2.1.0 deduped
| | +-- node-opcua-nodeid@2.1.5 deduped
| | +-- node-opcua-status-code@2.1.5 deduped
| | +-- node-opcua-utils@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +-- node-opcua-hostname@2.1.0
| | `-- node-opcua-assert@2.0.0 deduped
| +-- node-opcua-nodeid@2.1.5
| | +-- node-opcua-assert@2.0.0 deduped
| | +-- node-opcua-constants@2.1.5 deduped
| | +-- node-opcua-enum@2.1.0 deduped
| | +-- node-opcua-guid@2.1.0 deduped
| | `-- underscore@1.9.1 deduped
| +-- node-opcua-nodesets@2.1.5
| +-- node-opcua-numeric-range@2.1.5
| | +-- node-opcua-assert@2.0.0 deduped
| | +-- node-opcua-basic-types@2.1.5 deduped
| | +-- node-opcua-binary-stream@2.1.0 deduped
| | +-- node-opcua-factory@2.1.5 deduped
| | +-- node-opcua-status-code@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +-- node-opcua-packet-analyzer@2.1.5
| | +-- chalk@2.4.2 deduped
| | +-- node-opcua-assert@2.0.0 deduped
| | +-- node-opcua-basic-types@2.1.5 deduped
| | +-- node-opcua-binary-stream@2.1.0 deduped
| | +-- node-opcua-debug@2.1.0 deduped
| | +-- node-opcua-factory@2.1.5 deduped
| | +-- node-opcua-utils@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +-- node-opcua-secure-channel@2.1.5
| | +-- backoff@2.5.0
| | | `-- precondition@0.2.3
| | | +-- chalk@2.4.2 deduped
| | | +-- node-opcua-assert@2.0.0 deduped
| | | +-- node-opcua-basic-types@2.1.5 deduped
| | | +-- node-opcua-binary-stream@2.1.0 deduped
| | | +-- node-opcua-certificate-manager@2.1.5 deduped
| | | +-- node-opcua-chunkmanager@2.1.5
| | | | +-- chalk@2.4.2 deduped
| | | | +-- node-opcua-assert@2.0.0 deduped

```

```

| | | +- node-opcua-basic-types@2.1.5 deduped
| | | +- node-opcua-binary-stream@2.1.0 deduped
| | | +- node-opcua-buffer-utils@2.1.0 deduped
| | | +- node-opcua-factory@2.1.5 deduped
| | | +- node-opcua-packet-assembler@2.0.0 deduped
| | | `-- underscore@1.9.1 deduped
| | +- node-opcua-common@2.1.5 deduped
| | +- node-opcua-crypto@1.1.2 deduped
| | +- node-opcua-debug@2.1.0 deduped
| | +- node-opcua-enum@2.1.0 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-object-registry@2.1.0 deduped
| | +- node-opcua-packet-analyzer@2.1.5 deduped
| | +- node-opcua-service-secure-channel@2.1.5 deduped
| | +- node-opcua-status-code@2.1.5 deduped
| | +- node-opcua-transport@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | +- node-opcua-utils@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +- node-opcua-server@2.1.5
| +- async@3.1.0 deduped
| +- bonjour@3.5.0
| | +- array-flatten@2.1.2
| | +- deep-equal@1.0.1
| | +- dns-equal@1.0.0
| | +- dns-txt@2.0.2
| | | `-- buffer-indexof@1.1.1
| | +- multicast-dns@6.2.3
| | | +- dns-packet@1.3.1
| | | | +- ip@1.1.5
| | | | `-- safe-buffer@5.1.2 deduped
| | | `-- thunky@1.0.3
| | `-- multicast-dns-service-types@1.1.0
| +- colors@1.3.3 deduped
| +- dequeue@1.0.5 deduped
| +- node-opcua-address-space@2.1.5 deduped
| +- node-opcua-assert@2.0.0 deduped
| +- node-opcua-basic-types@2.1.5 deduped
| +- node-opcua-certificate-manager@2.1.5 deduped
| +- node-opcua-client@2.1.5 deduped
| +- node-opcua-client-dynamic-extension-object@2.1.5 deduped
| +- node-opcua-common@2.1.5 deduped
| +- node-opcua-constants@2.1.5 deduped
| +- node-opcua-crypto@1.1.2 deduped
| +- node-opcua-data-model@2.1.5 deduped
| +- node-opcua-data-value@2.1.5 deduped
| +- node-opcua-date-time@2.1.5 deduped
| +- node-opcua-debug@2.1.0 deduped
| +- node-opcua-enum@2.1.0 deduped
| +- node-opcua-extension-object@2.1.5 deduped
| +- node-opcua-factory@2.1.5 deduped
| +- node-opcua-hostname@2.1.0 deduped
| +- node-opcua-nodeid@2.1.5 deduped
| +- node-opcua-nodesets@2.1.5 deduped
| +- node-opcua-numeric-range@2.1.5 deduped

```

```

| | +- node-opcua-object-registry@2.1.0 deduped
| | +- node-opcua-pki@1.5.3 deduped
| | +- node-opcua-secure-channel@2.1.5 deduped
| | +- node-opcua-service-browse@2.1.5 deduped
| | +- node-opcua-service-call@2.1.5 deduped
| | +- node-opcua-service-discovery@2.1.5 deduped
| | +- node-opcua-service-endpoints@2.1.5 deduped
| | +- node-opcua-service-filter@2.1.5 deduped
| | +- node-opcua-service-history@2.1.5 deduped
| | +- node-opcua-service-node-management@2.1.5 deduped
| | +- node-opcua-service-query@2.1.5 deduped
| | +- node-opcua-service-read@2.1.5 deduped
| | +- node-opcua-service-register-node@2.1.5 deduped
| | +- node-opcua-service-secure-channel@2.1.5 deduped
| | +- node-opcua-service-session@2.1.5 deduped
| | +- node-opcua-service-subscription@2.1.5 deduped
| | +- node-opcua-service-translate-browse-path@2.1.5 deduped
| | +- node-opcua-service-write@2.1.5 deduped
| | +- node-opcua-status-code@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | +- node-opcua-utils@2.1.5 deduped
| | +- node-opcua-variant@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
+- node-opcua-server-discovery@2.1.5
| | +- @types/bonjour@3.5.5
| | | `-- @types/node@12.7.1 deduped
| | +- bonjour@3.5.0 deduped
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-certificate-manager@2.1.5 deduped
| | +- node-opcua-common@2.1.5 deduped
| | +- node-opcua-debug@2.1.0 deduped
| | +- node-opcua-hostname@2.1.0 deduped
| | +- node-opcua-pki@1.5.3 deduped
| | +- node-opcua-secure-channel@2.1.5 deduped
| | +- node-opcua-server@2.1.5 deduped
| | +- node-opcua-service-discovery@2.1.5 deduped
| | +- node-opcua-service-endpoints@2.1.5 deduped
| | +- node-opcua-status-code@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
+- node-opcua-service-browse@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-data-model@2.1.5 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-service-secure-channel@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
+- node-opcua-service-call@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped

```



```

| | +- node-opcua-service-secure-channel@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | +- node-opcua-variant@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +- node-opcua-service-discovery@2.1.5
| | +- @types/bonjour@3.5.5 deduped
| | +- bonjour@3.5.0 deduped
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-data-model@2.1.5 deduped
| | +- node-opcua-debug@2.1.0 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-service-endpoints@2.1.5 deduped
| | +- node-opcua-service-secure-channel@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +- node-opcua-service-endpoints@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-data-model@2.1.5 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-service-secure-channel@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +- node-opcua-service-filter@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-constants@2.1.5 deduped
| | +- node-opcua-data-model@2.1.5 deduped
| | +- node-opcua-debug@2.1.0 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-numeric-range@2.1.5 deduped
| | +- node-opcua-service-translate-browse-path@2.1.5 deduped
| | +- node-opcua-status-code@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | +- node-opcua-variant@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +- node-opcua-service-history@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-data-model@2.1.5 deduped
| | +- node-opcua-data-value@2.1.5 deduped
| | +- node-opcua-extension-object@2.1.5 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-service-filter@2.1.5 deduped
| | +- node-opcua-service-secure-channel@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | `-- underscore@1.9.1 deduped

```

```

| +-- node-opcua-service-node-management@2.1.5
| | +-- node-opcua-assert@2.0.0 deduped
| | +-- node-opcua-basic-types@2.1.5 deduped
| | +-- node-opcua-binary-stream@2.1.0 deduped
| | +-- node-opcua-data-model@2.1.5 deduped
| | +-- node-opcua-extension-object@2.1.5 deduped
| | +-- node-opcua-factory@2.1.5 deduped
| | +-- node-opcua-nodeid@2.1.5 deduped
| | +-- node-opcua-service-secure-channel@2.1.5 deduped
| | +-- node-opcua-types@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +-- node-opcua-service-query@2.1.5
| | +-- node-opcua-assert@2.0.0 deduped
| | +-- node-opcua-basic-types@2.1.5 deduped
| | +-- node-opcua-binary-stream@2.1.0 deduped
| | +-- node-opcua-data-model@2.1.5 deduped
| | +-- node-opcua-factory@2.1.5 deduped
| | +-- node-opcua-nodeid@2.1.5 deduped
| | +-- node-opcua-numeric-range@2.1.5 deduped
| | +-- node-opcua-service-browse@2.1.5 deduped
| | +-- node-opcua-service-filter@2.1.5 deduped
| | +-- node-opcua-service-secure-channel@2.1.5 deduped
| | +-- node-opcua-service-subscription@2.1.5 deduped
| | +-- node-opcua-service-translate-browse-path@2.1.5 deduped
| | +-- node-opcua-types@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +-- node-opcua-service-read@2.1.5
| | +-- node-opcua-assert@2.0.0 deduped
| | +-- node-opcua-basic-types@2.1.5 deduped
| | +-- node-opcua-binary-stream@2.1.0 deduped
| | +-- node-opcua-data-model@2.1.5 deduped
| | +-- node-opcua-data-value@2.1.5 deduped
| | +-- node-opcua-factory@2.1.5 deduped
| | +-- node-opcua-nodeid@2.1.5 deduped
| | +-- node-opcua-numeric-range@2.1.5 deduped
| | +-- node-opcua-service-secure-channel@2.1.5 deduped
| | +-- node-opcua-types@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +-- node-opcua-service-register-node@2.1.5
| | +-- node-opcua-assert@2.0.0 deduped
| | +-- node-opcua-basic-types@2.1.5 deduped
| | +-- node-opcua-binary-stream@2.1.0 deduped
| | +-- node-opcua-factory@2.1.5 deduped
| | +-- node-opcua-nodeid@2.1.5 deduped
| | +-- node-opcua-service-secure-channel@2.1.5 deduped
| | +-- node-opcua-types@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +-- node-opcua-service-secure-channel@2.1.5
| | +-- node-opcua-assert@2.0.0 deduped
| | +-- node-opcua-basic-types@2.1.5 deduped
| | +-- node-opcua-binary-stream@2.1.0 deduped
| | +-- node-opcua-data-model@2.1.5 deduped
| | +-- node-opcua-extension-object@2.1.5 deduped
| | +-- node-opcua-factory@2.1.5 deduped
| | +-- node-opcua-nodeid@2.1.5 deduped
| | +-- node-opcua-types@2.1.5 deduped

```

```

| | `-- underscore@1.9.1 deduped
| +- node-opcua-service-session@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-extension-object@2.1.5 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-service-endpoints@2.1.5 deduped
| | +- node-opcua-service-secure-channel@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +- node-opcua-service-subscription@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-data-value@2.1.5 deduped
| | +- node-opcua-extension-object@2.1.5 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-service-filter@2.1.5 deduped
| | +- node-opcua-service-read@2.1.5 deduped
| | +- node-opcua-service-secure-channel@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | +- node-opcua-variant@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +- node-opcua-service-translate-browse-path@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-constants@2.1.5 deduped
| | +- node-opcua-data-model@2.1.5 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-service-secure-channel@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +- node-opcua-service-write@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-data-value@2.1.5 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-numeric-range@2.1.5 deduped
| | +- node-opcua-service-secure-channel@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +- node-opcua-status-code@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-constants@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +- node-opcua-transport@2.1.5
| | +- chalk@2.4.2 deduped
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped

```

```

| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-buffer-utils@2.1.0 deduped
| | +- node-opcua-chunkmanager@2.1.5 deduped
| | +- node-opcua-debug@2.1.0 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-packet-assembler@2.0.0
| | | +- node-opcua-assert@2.0.0 deduped
| | | `-- underscore@1.9.1 deduped
| | +- node-opcua-status-code@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | +- node-opcua-utils@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +- node-opcua-types@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-buffer-utils@2.1.0 deduped
| | +- node-opcua-data-model@2.1.5 deduped
| | +- node-opcua-data-value@2.1.5 deduped
| | +- node-opcua-date-time@2.1.5 deduped
| | +- node-opcua-debug@2.1.0 deduped
| | +- node-opcua-enum@2.1.0 deduped
| | +- node-opcua-extension-object@2.1.5 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-generator@2.1.5
| | | +- chalk@2.4.2 deduped
| | | +- node-opcua-assert@2.0.0 deduped
| | | +- node-opcua-basic-types@2.1.5 deduped
| | | +- node-opcua-buffer-utils@2.1.0 deduped
| | | +- node-opcua-constants@2.1.5 deduped
| | | +- node-opcua-date-time@2.1.5 deduped
| | | +- node-opcua-debug@2.1.0 deduped
| | | +- node-opcua-enum@2.1.0 deduped
| | | +- node-opcua-factory@2.1.5 deduped
| | | +- node-opcua-guid@2.1.0 deduped
| | | +- node-opcua-nodeid@2.1.5 deduped
| | | +- node-opcua-numeric-range@2.1.5 deduped
| | | +- node-opcua-schemas@2.1.5 deduped
| | | +- node-opcua-status-code@2.1.5 deduped
| | | +- node-opcua-utils@2.1.5 deduped
| | | +- node-opcua-xml2json@2.1.5 deduped
| | | +- prettier@1.18.2
| | | +- typescript@3.5.3
| | | `-- underscore@1.9.1 deduped
| | +- node-opcua-guid@2.1.0 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-numeric-range@2.1.5 deduped
| | +- node-opcua-packet-analyzer@2.1.5 deduped
| | +- node-opcua-schemas@2.1.5 deduped
| | +- node-opcua-status-code@2.1.5 deduped
| | +- node-opcua-utils@2.1.5 deduped
| | +- node-opcua-variant@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +- node-opcua-utils@2.1.5
| | +- node-opcua-assert@2.0.0 deduped

```

```

| | `-- underscore@1.9.1 deduped
| +-- node-opcua-variant@2.1.5
| | +-- node-opcua-assert@2.0.0 deduped
| | +-- node-opcua-basic-types@2.1.5 deduped
| | +-- node-opcua-data-model@2.1.5 deduped
| | +-- node-opcua-enum@2.1.0 deduped
| | +-- node-opcua-extension-object@2.1.5 deduped
| | +-- node-opcua-factory@2.1.5 deduped
| | +-- node-opcua-nodeid@2.1.5 deduped
| | +-- node-opcua-utils@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +-- node-opcua-vendor-diagnostic@2.1.5
| | +-- humanize@0.0.9
| | +-- node-opcua-address-space@2.1.5 deduped
| | +-- node-opcua-assert@2.0.0 deduped
| | +-- node-opcua-constants@2.1.5 deduped
| | +-- node-opcua-server@2.1.5 deduped
| | +-- node-opcua-status-code@2.1.5 deduped
| | `-- node-opcua-variant@2.1.5 deduped
| `-- semver@6.3.0
+-- node-opcua-file-transfer@2.1.5
| +-- colors@1.3.3
| +-- node-opcua-address-space@2.1.5
| | +-- async@3.1.0 deduped
| | +-- dequeue@1.0.5 deduped
| | +-- node-opcua-assert@2.0.0 deduped
| | +-- node-opcua-basic-types@2.1.5 deduped
| | +-- node-opcua-client-dynamic-extension-object@2.1.5
| | | +-- chalk@2.4.2 deduped
| | | +-- node-opcua-assert@2.0.0 deduped
| | | +-- node-opcua-binary-stream@2.1.0 deduped
| | | +-- node-opcua-data-model@2.1.5 deduped
| | | +-- node-opcua-data-value@2.1.5 deduped
| | | +-- node-opcua-debug@2.1.0 deduped
| | | +-- node-opcua-extension-object@2.1.5 deduped
| | | +-- node-opcua-factory@2.1.5 deduped
| | | +-- node-opcua-nodeid@2.1.5 deduped
| | | +-- node-opcua-pseudo-session@2.1.5 deduped
| | | +-- node-opcua-schemas@2.1.5 deduped
| | | +-- node-opcua-service-browse@2.1.5 deduped
| | | +-- node-opcua-status-code@2.1.5 deduped
| | | `-- node-opcua-variant@2.1.5 deduped
| | +-- node-opcua-common@2.1.5
| | | +-- node-opcua-assert@2.0.0 deduped
| | | +-- node-opcua-basic-types@2.1.5 deduped
| | | +-- node-opcua-crypto@1.1.2 deduped
| | | +-- node-opcua-data-model@2.1.5 deduped
| | | +-- node-opcua-factory@2.1.5 deduped
| | | +-- node-opcua-nodeid@2.1.5 deduped
| | | +-- node-opcua-service-endpoints@2.1.5 deduped
| | | +-- node-opcua-service-secure-channel@2.1.5 deduped
| | | +-- node-opcua-types@2.1.5 deduped
| | | `-- underscore@1.9.1 deduped
| | +-- node-opcua-constants@2.1.5
| | +-- node-opcua-crypto@1.1.2 deduped
| | +-- node-opcua-data-access@2.1.5

```

```

| | | +-- node-opcua-assert@2.0.0 deduped
| | | +-- node-opcua-basic-types@2.1.5 deduped
| | | +-- node-opcua-binary-stream@2.1.0 deduped
| | | +-- node-opcua-data-model@2.1.5 deduped
| | | +-- node-opcua-factory@2.1.5 deduped
| | | +-- node-opcua-nodeid@2.1.5 deduped
| | | +-- node-opcua-types@2.1.5 deduped
| | | `-- underscore@1.9.1 deduped
| | +-- node-opcua-data-model@2.1.5 deduped
| | +-- node-opcua-data-value@2.1.5
| | | +-- node-opcua-assert@2.0.0 deduped
| | | +-- node-opcua-basic-types@2.1.5 deduped
| | | +-- node-opcua-binary-stream@2.1.0 deduped
| | | +-- node-opcua-data-model@2.1.5 deduped
| | | +-- node-opcua-date-time@2.1.5 deduped
| | | +-- node-opcua-enum@2.1.0 deduped
| | | +-- node-opcua-extension-object@2.1.5 deduped
| | | +-- node-opcua-factory@2.1.5 deduped
| | | +-- node-opcua-nodeid@2.1.5 deduped
| | | +-- node-opcua-status-code@2.1.5 deduped
| | | +-- node-opcua-utils@2.1.5 deduped
| | | +-- node-opcua-variant@2.1.5 deduped
| | | `-- underscore@1.9.1 deduped
| | +-- node-opcua-date-time@2.1.5
| | | +-- long@4.0.0 deduped
| | | +-- node-opcua-assert@2.0.0 deduped
| | | `-- underscore@1.9.1 deduped
| | +-- node-opcua-debug@2.1.0 deduped
| | +-- node-opcua-enum@2.1.0 deduped
| | +-- node-opcua-factory@2.1.5
| | | +-- chalk@2.4.2 deduped
| | | +-- node-opcua-assert@2.0.0 deduped
| | | +-- node-opcua-basic-types@2.1.5 deduped
| | | +-- node-opcua-binary-stream@2.1.0 deduped
| | | +-- node-opcua-enum@2.1.0 deduped
| | | +-- node-opcua-guid@2.1.0 deduped
| | | +-- node-opcua-nodeid@2.1.5 deduped
| | | +-- node-opcua-status-code@2.1.5 deduped
| | | +-- node-opcua-utils@2.1.5 deduped
| | | `-- underscore@1.9.1 deduped
| | +-- node-opcua-nodeid@2.1.5 deduped
| | +-- node-opcua-numeric-range@2.1.5
| | | +-- node-opcua-assert@2.0.0 deduped
| | | +-- node-opcua-basic-types@2.1.5 deduped
| | | +-- node-opcua-binary-stream@2.1.0 deduped
| | | +-- node-opcua-factory@2.1.5 deduped
| | | +-- node-opcua-status-code@2.1.5 deduped
| | | `-- underscore@1.9.1 deduped
| | +-- node-opcua-object-registry@2.1.0 deduped
| | +-- node-opcua-pseudo-session@2.1.5 deduped
| | +-- node-opcua-schemas@2.1.5
| | | +-- node-opcua-assert@2.0.0 deduped
| | | +-- node-opcua-binary-stream@2.1.0 deduped
| | | +-- node-opcua-data-model@2.1.5 deduped
| | | +-- node-opcua-debug@2.1.0 deduped
| | | +-- node-opcua-enum@2.1.0 deduped

```

```

| | | +- node-opcua-extension-object@2.1.5 deduped
| | | +- node-opcua-factory@2.1.5 deduped
| | | +- node-opcua-nodeid@2.1.5 deduped
| | | +- node-opcua-variant@2.1.5 deduped
| | | `-- node-opcua-xml2json@2.1.5 deduped
| | +- node-opcua-service-browse@2.1.5
| | | +- node-opcua-assert@2.0.0 deduped
| | | +- node-opcua-basic-types@2.1.5 deduped
| | | +- node-opcua-binary-stream@2.1.0 deduped
| | | +- node-opcua-data-model@2.1.5 deduped
| | | +- node-opcua-factory@2.1.5 deduped
| | | +- node-opcua-nodeid@2.1.5 deduped
| | | +- node-opcua-service-secure-channel@2.1.5 deduped
| | | +- node-opcua-types@2.1.5 deduped
| | | `-- underscore@1.9.1 deduped
| | +- node-opcua-service-call@2.1.5 deduped
| | +- node-opcua-service-filter@2.1.5
| | | +- node-opcua-assert@2.0.0 deduped
| | | +- node-opcua-basic-types@2.1.5 deduped
| | | +- node-opcua-binary-stream@2.1.0 deduped
| | | +- node-opcua-constants@2.1.5 deduped
| | | +- node-opcua-data-model@2.1.5 deduped
| | | +- node-opcua-debug@2.1.0 deduped
| | | +- node-opcua-factory@2.1.5 deduped
| | | +- node-opcua-nodeid@2.1.5 deduped
| | | +- node-opcua-numeric-range@2.1.5 deduped
| | | +- node-opcua-service-translate-browse-path@2.1.5 deduped
| | | +- node-opcua-status-code@2.1.5 deduped
| | | +- node-opcua-types@2.1.5 deduped
| | | +- node-opcua-variant@2.1.5 deduped
| | | `-- underscore@1.9.1 deduped
| | +- node-opcua-service-history@2.1.5
| | | +- node-opcua-assert@2.0.0 deduped
| | | +- node-opcua-basic-types@2.1.5 deduped
| | | +- node-opcua-binary-stream@2.1.0 deduped
| | | +- node-opcua-data-model@2.1.5 deduped
| | | +- node-opcua-data-value@2.1.5 deduped
| | | +- node-opcua-extension-object@2.1.5 deduped
| | | +- node-opcua-factory@2.1.5 deduped
| | | +- node-opcua-nodeid@2.1.5 deduped
| | | +- node-opcua-service-filter@2.1.5 deduped
| | | +- node-opcua-service-secure-channel@2.1.5 deduped
| | | +- node-opcua-types@2.1.5 deduped
| | | `-- underscore@1.9.1 deduped
| | +- node-opcua-service-session@2.1.5
| | | +- node-opcua-assert@2.0.0 deduped
| | | +- node-opcua-basic-types@2.1.5 deduped
| | | +- node-opcua-binary-stream@2.1.0 deduped
| | | +- node-opcua-extension-object@2.1.5 deduped
| | | +- node-opcua-factory@2.1.5 deduped
| | | +- node-opcua-nodeid@2.1.5 deduped
| | | +- node-opcua-service-endpoints@2.1.5 deduped
| | | +- node-opcua-service-secure-channel@2.1.5 deduped
| | | +- node-opcua-types@2.1.5 deduped
| | | `-- underscore@1.9.1 deduped
| | +- node-opcua-service-translate-browse-path@2.1.5 deduped

```

```

| | +- node-opcua-service-write@2.1.5
| | | +- node-opcua-assert@2.0.0 deduped
| | | +- node-opcua-basic-types@2.1.5 deduped
| | | +- node-opcua-binary-stream@2.1.0 deduped
| | | +- node-opcua-data-value@2.1.5 deduped
| | | +- node-opcua-factory@2.1.5 deduped
| | | +- node-opcua-nodeid@2.1.5 deduped
| | | +- node-opcua-numeric-range@2.1.5 deduped
| | | +- node-opcua-service-secure-channel@2.1.5 deduped
| | | +- node-opcua-types@2.1.5 deduped
| | | `-- underscore@1.9.1 deduped
| | +- node-opcua-status-code@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | +- node-opcua-utils@2.1.5
| | | +- node-opcua-assert@2.0.0 deduped
| | | `-- underscore@1.9.1 deduped
| | +- node-opcua-variant@2.1.5 deduped
| | +- node-opcua-xml2json@2.1.5
| | | +- bomstrip@0.1.4 deduped
| | | +- ltx@2.8.1 deduped
| | | +- node-opcua-assert@2.0.0 deduped
| | | +- node-opcua-debug@2.1.0 deduped
| | | +- node-opcua-utils@2.1.5 deduped
| | | `-- underscore@1.9.1 deduped
| | +- object.values@1.1.0 deduped
| | +- set-prototype-of@1.0.0 deduped
| | +- underscore@1.9.1 deduped
| | `-- xml-writer@1.7.0 deduped
| +- node-opcua-assert@2.0.0 deduped
| +- node-opcua-basic-types@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-buffer-utils@2.1.0 deduped
| | +- node-opcua-date-time@2.1.5 deduped
| | +- node-opcua-enum@2.1.0 deduped
| | +- node-opcua-guid@2.1.0 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-status-code@2.1.5 deduped
| | +- node-opcua-utils@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +- node-opcua-client@2.1.5
| | +- @types/async@3.0.1 deduped
| | +- @types/once@1.4.0 deduped
| | +- @types/underscore@1.9.2 deduped
| | +- async@3.1.0 deduped
| | +- callbackify@1.1.0 deduped
| | +- chalk@2.4.2 deduped
| | +- delayed@2.0.0 deduped
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-buffer-utils@2.1.0 deduped
| | +- node-opcua-certificate-manager@2.1.5
| | | +- @types/mkdirp@0.5.2 deduped
| | | +- async@3.1.0 deduped
| | | +- chalk@2.4.2 deduped
| | | +- delayed@2.0.0 deduped

```



```

| | | +-- env-paths@2.2.0
| | | +-- mkdirp@0.5.1 deduped
| | | +-- node-opcua-assert@2.0.0 deduped
| | | +-- node-opcua-basic-types@2.1.5 deduped
| | | +-- node-opcua-buffer-utils@2.1.0 deduped
| | | +-- node-opcua-common@2.1.5 deduped
| | | +-- node-opcua-constants@2.1.5 deduped
| | | +-- node-opcua-crypto@1.1.2 deduped
| | | +-- node-opcua-debug@2.1.0 deduped
| | | +-- node-opcua-pki@1.5.3 deduped
| | | +-- node-opcua-status-code@2.1.5 deduped
| | | +-- node-opcua-utils@2.1.5 deduped
| | | +-- once@1.4.0 deduped
| | | +-- thenify@3.3.0 deduped
| | | `-- underscore@1.9.1 deduped
| | +-- node-opcua-client-dynamic-extension-object@2.1.5 deduped
| | +-- node-opcua-common@2.1.5 deduped
| | +-- node-opcua-constants@2.1.5 deduped
| | +-- node-opcua-crypto@1.1.2 deduped
| | +-- node-opcua-data-model@2.1.5 deduped
| | +-- node-opcua-data-value@2.1.5 deduped
| | +-- node-opcua-date-time@2.1.5 deduped
| | +-- node-opcua-debug@2.1.0 deduped
| | +-- node-opcua-extension-object@2.1.5
| | +-- chalk@2.4.2 deduped
| | +-- node-opcua-assert@2.0.0 deduped
| | +-- node-opcua-basic-types@2.1.5 deduped
| | +-- node-opcua-binary-stream@2.1.0 deduped
| | +-- node-opcua-debug@2.1.0 deduped
| | +-- node-opcua-factory@2.1.5 deduped
| | | `-- node-opcua-nodeid@2.1.5 deduped
| | +-- node-opcua-hostname@2.1.0 deduped
| | +-- node-opcua-nodeid@2.1.5 deduped
| | +-- node-opcua-object-registry@2.1.0 deduped
| | +-- node-opcua-pki@1.5.3 deduped
| | +-- node-opcua-pseudo-session@2.1.5 deduped
| | +-- node-opcua-schemas@2.1.5 deduped
| | +-- node-opcua-secure-channel@2.1.5 deduped
| | +-- node-opcua-service-browse@2.1.5 deduped
| | +-- node-opcua-service-call@2.1.5 deduped
| | +-- node-opcua-service-discovery@2.1.5
| | | +-- @types/bonjour@3.5.5 deduped
| | | +-- bonjour@3.5.0 deduped
| | +-- node-opcua-assert@2.0.0 deduped
| | +-- node-opcua-basic-types@2.1.5 deduped
| | +-- node-opcua-binary-stream@2.1.0 deduped
| | +-- node-opcua-data-model@2.1.5 deduped
| | +-- node-opcua-debug@2.1.0 deduped
| | +-- node-opcua-factory@2.1.5 deduped
| | +-- node-opcua-nodeid@2.1.5 deduped
| | +-- node-opcua-service-endpoints@2.1.5 deduped
| | +-- node-opcua-service-secure-channel@2.1.5 deduped
| | +-- node-opcua-types@2.1.5 deduped
| | | `-- underscore@1.9.1 deduped
| | +-- node-opcua-service-endpoints@2.1.5
| | +-- node-opcua-assert@2.0.0 deduped

```

```

| | | +- node-opcua-basic-types@2.1.5 deduped
| | | +- node-opcua-binary-stream@2.1.0 deduped
| | | +- node-opcua-data-model@2.1.5 deduped
| | | +- node-opcua-factory@2.1.5 deduped
| | | +- node-opcua-nodeid@2.1.5 deduped
| | | +- node-opcua-service-secure-channel@2.1.5 deduped
| | | +- node-opcua-types@2.1.5 deduped
| | | `-- underscore@1.9.1 deduped
| | +- node-opcua-service-filter@2.1.5 deduped
| | +- node-opcua-service-history@2.1.5 deduped
| | +- node-opcua-service-query@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-data-model@2.1.5 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-numeric-range@2.1.5 deduped
| | +- node-opcua-service-browse@2.1.5 deduped
| | +- node-opcua-service-filter@2.1.5 deduped
| | +- node-opcua-service-secure-channel@2.1.5 deduped
| | +- node-opcua-service-subscription@2.1.5 deduped
| | +- node-opcua-service-translate-browse-path@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | | `-- underscore@1.9.1 deduped
| | +- node-opcua-service-read@2.1.5 deduped
| | +- node-opcua-service-register-node@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-service-secure-channel@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | | `-- underscore@1.9.1 deduped
| | +- node-opcua-service-secure-channel@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-data-model@2.1.5 deduped
| | +- node-opcua-extension-object@2.1.5 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | | `-- underscore@1.9.1 deduped
| | +- node-opcua-service-session@2.1.5 deduped
| | +- node-opcua-service-subscription@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-data-value@2.1.5 deduped
| | +- node-opcua-extension-object@2.1.5 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-service-filter@2.1.5 deduped
| | +- node-opcua-service-read@2.1.5 deduped

```

```

| | | +- node-opcua-service-secure-channel@2.1.5 deduped
| | | +- node-opcua-types@2.1.5 deduped
| | | +- node-opcua-variant@2.1.5 deduped
| | | `-- underscore@1.9.1 deduped
| | +- node-opcua-service-translate-browse-path@2.1.5 deduped
| | +- node-opcua-service-write@2.1.5 deduped
| | +- node-opcua-status-code@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | +- node-opcua-utils@2.1.5 deduped
| | +- node-opcua-variant@2.1.5 deduped
| | +- once@1.4.0 deduped
| | +- thenify@3.3.0 deduped
| | `-- underscore@1.9.1 deduped
| +- node-opcua-data-model@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-enum@2.1.0 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-status-code@2.1.5 deduped
| | +- node-opcua-utils@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +- node-opcua-debug@2.1.0 deduped
| +- node-opcua-nodeid@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-constants@2.1.5 deduped
| | +- node-opcua-enum@2.1.0 deduped
| | +- node-opcua-guid@2.1.0 deduped
| | `-- underscore@1.9.1 deduped
| +- node-opcua-nodesets@2.1.5
| +- node-opcua-pki@1.5.3
| | +- async@3.1.0 deduped
| | +- better-assert@1.0.2 deduped
| | +- byline@5.0.0
| | +- chalk@2.4.2 deduped
| | +- cli-spinner@0.2.10
| | +- cli-table@0.3.1
| | | `-- colors@1.0.3
| | +- del@5.0.0
| | | +- globby@10.0.1
| | | | +- @types/glob@7.1.1
| | | | | +- @types/events@3.0.0
| | | | | +- @types/minimatch@3.0.3
| | | | | `-- @types/node@12.7.1 deduped
| | | | +- array-union@2.1.0
| | | | +- dir-glob@3.0.1
| | | | | `-- path-type@4.0.0
| | | | +- fast-glob@3.0.4
| | | | | +- @nodelib/fs.stat@2.0.1
| | | | | +- @nodelib/fs.walk@1.2.2
| | | | | | +- @nodelib/fs.scandir@2.1.1
| | | | | | | +- @nodelib/fs.stat@2.0.1 deduped
| | | | | | | `-- run-parallel@1.1.9
| | | | | | | `-- fastq@1.6.0
| | | | | | | `-- reusify@1.0.4

```

```

| | | | +- glob-parent@5.0.0
| | | | | `-- is-glob@4.0.1 deduped
| | | | +- is-glob@4.0.1
| | | | | `-- is-extglob@2.1.1
| | | | +- merge2@1.2.4 deduped
| | | | | `-- micromatch@4.0.2
| | | | +- braces@3.0.2
| | | | | `-- fill-range@7.0.1
| | | | | `-- to-regex-range@5.0.1
| | | | | `-- is-number@7.0.0
| | | | | `-- picomatch@2.0.7
| | | +- glob@7.1.4 deduped
| | | +- ignore@5.1.2
| | | +- merge2@1.2.4
| | | | `-- slash@3.0.0
| | +- is-path-cwd@2.2.0
| | +- is-path-in-cwd@2.1.0
| | | `-- is-path-inside@2.1.0
| | | `-- path-is-inside@1.0.2 deduped
| | +- p-map@2.1.0
| | | `-- rimraf@2.6.3 deduped
| +- node-opcua-crypto@1.1.2 deduped
| +- progress@2.0.3
| +- source-map-support@0.5.13 deduped
| +- thenify@3.3.0 deduped
| +- underscore@1.9.1 deduped
| +- walk@2.3.14
| | `-- foreachasync@3.0.0
| +- wget-improved@3.0.2
| | +- minimist@1.2.0 deduped
| | | `-- tunnel@0.0.6
| +- yargs@13.3.0 deduped
| | `-- yauzl@2.10.0
| | +- buffer-crc32@0.2.13
| | | `-- fd-slicer@1.1.0
| | | `-- pend@1.2.0 deduped
| +- node-opcua-pseudo-session@2.1.5
| | +- async@3.1.0 deduped
| | +- node-opcua-data-model@2.1.5 deduped
| | +- node-opcua-data-value@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-service-browse@2.1.5 deduped
| | +- node-opcua-service-call@2.1.5 deduped
| | +- node-opcua-service-read@2.1.5 deduped
| | +- node-opcua-service-translate-browse-path@2.1.5 deduped
| | +- node-opcua-status-code@2.1.5 deduped
| | +- node-opcua-utils@2.1.5 deduped
| | +- node-opcua-variant@2.1.5 deduped
| | | `-- underscore@1.9.1 deduped
| +- node-opcua-secure-channel@2.1.5
| | +- backoff@2.5.0 deduped
| | +- chalk@2.4.2 deduped
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-certificate-manager@2.1.5 deduped

```

```

| | +- node-opcua-chunkmanager@2.1.5
| | | +- chalk@2.4.2 deduped
| | | +- node-opcua-assert@2.0.0 deduped
| | | +- node-opcua-basic-types@2.1.5 deduped
| | | +- node-opcua-binary-stream@2.1.0 deduped
| | | +- node-opcua-buffer-utils@2.1.0 deduped
| | | +- node-opcua-factory@2.1.5 deduped
| | | +- node-opcua-packet-assembler@2.0.0 deduped
| | | `-- underscore@1.9.1 deduped
| | +- node-opcua-common@2.1.5 deduped
| | +- node-opcua-crypto@1.1.2 deduped
| | +- node-opcua-debug@2.1.0 deduped
| | +- node-opcua-enum@2.1.0 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-object-registry@2.1.0 deduped
| | +- node-opcua-packet-analyzer@2.1.5
| | | +- chalk@2.4.2 deduped
| | | +- node-opcua-assert@2.0.0 deduped
| | | +- node-opcua-basic-types@2.1.5 deduped
| | | +- node-opcua-binary-stream@2.1.0 deduped
| | | +- node-opcua-debug@2.1.0 deduped
| | | +- node-opcua-factory@2.1.5 deduped
| | | +- node-opcua-utils@2.1.5 deduped
| | | `-- underscore@1.9.1 deduped
| | +- node-opcua-service-secure-channel@2.1.5 deduped
| | +- node-opcua-status-code@2.1.5 deduped
| | +- node-opcua-transport@2.1.5
| | | +- chalk@2.4.2 deduped
| | | +- node-opcua-assert@2.0.0 deduped
| | | +- node-opcua-basic-types@2.1.5 deduped
| | | +- node-opcua-binary-stream@2.1.0 deduped
| | | +- node-opcua-buffer-utils@2.1.0 deduped
| | | +- node-opcua-chunkmanager@2.1.5 deduped
| | | +- node-opcua-debug@2.1.0 deduped
| | | +- node-opcua-factory@2.1.5 deduped
| | | +- node-opcua-nodeid@2.1.5 deduped
| | | +- node-opcua-packet-assembler@2.0.0 deduped
| | | +- node-opcua-status-code@2.1.5 deduped
| | | +- node-opcua-types@2.1.5 deduped
| | | +- node-opcua-utils@2.1.5 deduped
| | | `-- underscore@1.9.1 deduped
| | +- node-opcua-types@2.1.5 deduped
| | +- node-opcua-utils@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +- node-opcua-server@2.1.5
| +- async@3.1.0 deduped
| +- bonjour@3.5.0 deduped
| +- colors@1.3.3 deduped
| +- dequeue@1.0.5 deduped
| +- node-opcua-address-space@2.1.5 deduped
| +- node-opcua-assert@2.0.0 deduped
| +- node-opcua-basic-types@2.1.5 deduped
| +- node-opcua-certificate-manager@2.1.5 deduped
| +- node-opcua-client@2.1.5 deduped
| +- node-opcua-client-dynamic-extension-object@2.1.5 deduped

```

```

| | +- node-opcua-common@2.1.5 deduped
| | +- node-opcua-constants@2.1.5 deduped
| | +- node-opcua-crypto@1.1.2 deduped
| | +- node-opcua-data-model@2.1.5 deduped
| | +- node-opcua-data-value@2.1.5 deduped
| | +- node-opcua-date-time@2.1.5 deduped
| | +- node-opcua-debug@2.1.0 deduped
| | +- node-opcua-enum@2.1.0 deduped
| | +- node-opcua-extension-object@2.1.5 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-hostname@2.1.0 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-nodesets@2.1.5 deduped
| | +- node-opcua-numeric-range@2.1.5 deduped
| | +- node-opcua-object-registry@2.1.0 deduped
| | +- node-opcua-pki@1.5.3 deduped
| | +- node-opcua-secure-channel@2.1.5 deduped
| | +- node-opcua-service-browse@2.1.5 deduped
| | +- node-opcua-service-call@2.1.5 deduped
| | +- node-opcua-service-discovery@2.1.5 deduped
| | +- node-opcua-service-endpoints@2.1.5 deduped
| | +- node-opcua-service-filter@2.1.5 deduped
| | +- node-opcua-service-history@2.1.5 deduped
| | +- node-opcua-service-node-management@2.1.5
| | | +- node-opcua-assert@2.0.0 deduped
| | | +- node-opcua-basic-types@2.1.5 deduped
| | | +- node-opcua-binary-stream@2.1.0 deduped
| | | +- node-opcua-data-model@2.1.5 deduped
| | | +- node-opcua-extension-object@2.1.5 deduped
| | | +- node-opcua-factory@2.1.5 deduped
| | | +- node-opcua-nodeid@2.1.5 deduped
| | | +- node-opcua-service-secure-channel@2.1.5 deduped
| | | +- node-opcua-types@2.1.5 deduped
| | | `-- underscore@1.9.1 deduped
| | +- node-opcua-service-query@2.1.5 deduped
| | +- node-opcua-service-read@2.1.5 deduped
| | +- node-opcua-service-register-node@2.1.5 deduped
| | +- node-opcua-service-secure-channel@2.1.5 deduped
| | +- node-opcua-service-session@2.1.5 deduped
| | +- node-opcua-service-subscription@2.1.5 deduped
| | +- node-opcua-service-translate-browse-path@2.1.5 deduped
| | +- node-opcua-service-write@2.1.5 deduped
| | +- node-opcua-status-code@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | +- node-opcua-utils@2.1.5 deduped
| | +- node-opcua-variant@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +- node-opcua-service-call@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-service-secure-channel@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | +- node-opcua-variant@2.1.5 deduped

```

```

| | `-- underscore@1.9.1 deduped
| +- node-opcua-service-read@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-data-model@2.1.5 deduped
| | +- node-opcua-data-value@2.1.5 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-numeric-range@2.1.5 deduped
| | +- node-opcua-service-secure-channel@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +- node-opcua-service-translate-browse-path@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-constants@2.1.5 deduped
| | +- node-opcua-data-model@2.1.5 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-nodeid@2.1.5 deduped
| | +- node-opcua-service-secure-channel@2.1.5 deduped
| | +- node-opcua-types@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +- node-opcua-status-code@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-constants@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| +- node-opcua-types@2.1.5
| | +- node-opcua-assert@2.0.0 deduped
| | +- node-opcua-basic-types@2.1.5 deduped
| | +- node-opcua-binary-stream@2.1.0 deduped
| | +- node-opcua-buffer-utils@2.1.0 deduped
| | +- node-opcua-data-model@2.1.5 deduped
| | +- node-opcua-data-value@2.1.5 deduped
| | +- node-opcua-date-time@2.1.5 deduped
| | +- node-opcua-debug@2.1.0 deduped
| | +- node-opcua-enum@2.1.0 deduped
| | +- node-opcua-extension-object@2.1.5 deduped
| | +- node-opcua-factory@2.1.5 deduped
| | +- node-opcua-generator@2.1.5
| | | +- chalk@2.4.2 deduped
| | | +- node-opcua-assert@2.0.0 deduped
| | | +- node-opcua-basic-types@2.1.5 deduped
| | | +- node-opcua-buffer-utils@2.1.0 deduped
| | | +- node-opcua-constants@2.1.5 deduped
| | | +- node-opcua-date-time@2.1.5 deduped
| | | +- node-opcua-debug@2.1.0 deduped
| | | +- node-opcua-enum@2.1.0 deduped
| | | +- node-opcua-factory@2.1.5 deduped
| | | +- node-opcua-guid@2.1.0 deduped
| | | +- node-opcua-nodeid@2.1.5 deduped
| | | +- node-opcua-numeric-range@2.1.5 deduped
| | | +- node-opcua-schemas@2.1.5 deduped
| | | +- node-opcua-status-code@2.1.5 deduped
| | | +- node-opcua-utils@2.1.5 deduped

```

```

| | | +-- node-opcua-xml2json@2.1.5 deduped
| | | +-- prettier@1.18.2 deduped
| | | +-- typescript@3.5.3 deduped
| | | `-- underscore@1.9.1 deduped
| | +-- node-opcua-guid@2.1.0 deduped
| | +-- node-opcua-nodeid@2.1.5 deduped
| | +-- node-opcua-numeric-range@2.1.5 deduped
| | +-- node-opcua-packet-analyzer@2.1.5 deduped
| | +-- node-opcua-schemas@2.1.5 deduped
| | +-- node-opcua-status-code@2.1.5 deduped
| | +-- node-opcua-utils@2.1.5 deduped
| | +-- node-opcua-variant@2.1.5 deduped
| | `-- underscore@1.9.1 deduped
| `-- node-opcua-variant@2.1.5
|   +-- node-opcua-assert@2.0.0 deduped
|   +-- node-opcua-basic-types@2.1.5 deduped
|   +-- node-opcua-data-model@2.1.5 deduped
|   +-- node-opcua-enum@2.1.0 deduped
|   +-- node-opcua-extension-object@2.1.5 deduped
|   +-- node-opcua-factory@2.1.5 deduped
|   +-- node-opcua-nodeid@2.1.5 deduped
|   +-- node-opcua-utils@2.1.5 deduped
|   `-- underscore@1.9.1 deduped
+-- nodemailer@6.3.0
+-- request@2.88.0
| +-- aws-sign2@0.7.0
| +-- aws4@1.8.0
| +-- caseless@0.12.0
| +-- combined-stream@1.0.8
| | `-- delayed-stream@1.0.0
| +-- extend@3.0.2
| +-- forever-agent@0.6.1
| +-- form-data@2.3.3
| | +-- asynckit@0.4.0
| | +-- combined-stream@1.0.8 deduped
| | `-- mime-types@2.1.24 deduped
| +-- har-validator@5.1.3
| | +-- ajv@6.10.0
| | | +-- fast-deep-equal@2.0.1
| | | +-- fast-json-stable-stringify@2.0.0
| | | +-- json-schema-traverse@0.4.1
| | | `-- uri-js@4.2.2
| | |   `-- punycode@2.1.1
| | `-- har-schema@2.0.0
| +-- http-signature@1.2.0
| | +-- assert-plus@1.0.0
| | +-- jsprim@1.4.1
| | | +-- assert-plus@1.0.0 deduped
| | | +-- extsprintf@1.3.0
| | | +-- json-schema@0.2.3
| | | `-- verror@1.10.0
| | |   +-- assert-plus@1.0.0 deduped
| | |   +-- core-util-is@1.0.2 deduped
| | |   `-- extsprintf@1.3.0 deduped
| | `-- sshpk@1.16.1
| |   +-- asn1@0.2.4

```



```

| | | `-- safer-buffer@2.1.2 deduped
| | +-- assert-plus@1.0.0 deduped
| | +-- bcrypt-pbkdf@1.0.2
| | | `-- tweetnacl@0.14.5 deduped
| | +-- dashdash@1.14.1
| | | `-- assert-plus@1.0.0 deduped
| | +-- ecc-jsbn@0.1.2
| | | +-- jsbn@0.1.1 deduped
| | | `-- safer-buffer@2.1.2 deduped
| | +-- getpass@0.1.7
| | | `-- assert-plus@1.0.0 deduped
| | +-- jsbn@0.1.1
| | +-- safer-buffer@2.1.2 deduped
| | `-- tweetnacl@0.14.5
| +-- is-typedarray@1.0.0
| +-- isstream@0.1.2
| +-- json-stringify-safe@5.0.1
| +-- mime-types@2.1.24
| | `-- mime-db@1.40.0
| +-- oauth-sign@0.9.0
| +-- performance-now@2.1.0
| +-- qs@6.5.2
| +-- safe-buffer@5.1.2 deduped
| +-- tough-cookie@2.4.3
| | +-- psl@1.1.32
| | | `-- punycode@1.4.1
| +-- tunnel-agent@0.6.0
| | `-- safe-buffer@5.1.2 deduped
| `-- uuid@3.3.2
`-- treeify@1.1.0

```

Appendix

A.1 Installation Files required in Windows 64Bit Environment

```

Folder PATH listing for volume New Volume
Volume serial number is 9885-ACB9
E:.
| builder-effective-config.yaml
| connectopc_ua Setup 1.0.0.exe
| connectopc_ua Setup 1.0.0.exe.blockmap
| tree.txt
|
|---.icon-ico
|   icon.ico
|
|---win-unpacked
|   chrome_100_percent.pak
|   chrome_200_percent.pak
|   connectopc_ua.exe

```

- | d3dcompiler_47.dll
- | ffmpeg.dll
- | icudtl.dat
- | libEGL.dll
- | libGLESv2.dll
- | LICENSE.electron.txt
- | LICENSES.chromium.html
- | natives_blob.bin
- | resources.pak
- | snapshot_blob.bin
- | v8_context_snapshot.bin

- | — locales
 - | am.pak
 - | ar.pak
 - | bg.pak
 - | bn.pak
 - | ca.pak
 - | cs.pak
 - | da.pak
 - | de.pak
 - | el.pak
 - | en-GB.pak
 - | en-US.pak
 - | es-419.pak
 - | es.pak
 - | et.pak
 - | fa.pak
 - | fi.pak
 - | fil.pak
 - | fr.pak
 - | gu.pak
 - | he.pak
 - | hi.pak
 - | hr.pak
 - | hu.pak
 - | id.pak
 - | it.pak
 - | ja.pak
 - | kn.pak
 - | ko.pak
 - | lt.pak
 - | lv.pak
 - | ml.pak
 - | mr.pak
 - | ms.pak
 - | nb.pak
 - | nl.pak
 - | pl.pak
 - | pt-BR.pak
 - | pt-PT.pak
 - | ro.pak
 - | ru.pak
 - | sk.pak
 - | sl.pak
 - | sr.pak

```

| sv.pak
| sw.pak
| ta.pak
| te.pak
| th.pak
| tr.pak
| uk.pak
| vi.pak
| zh-CN.pak
| zh-TW.pak
|
|---resources
|   app.asar
|   electron.asar
|   elevate.exe
|
|---swiftshader
|   libEGL.dll
|   libGLSv2.dll

```

A.2 Installation Files for Linux OS (Ubuntu, Fedora, Debian, Raspberry Pi3)

Folder PATH listing for volume New Volume

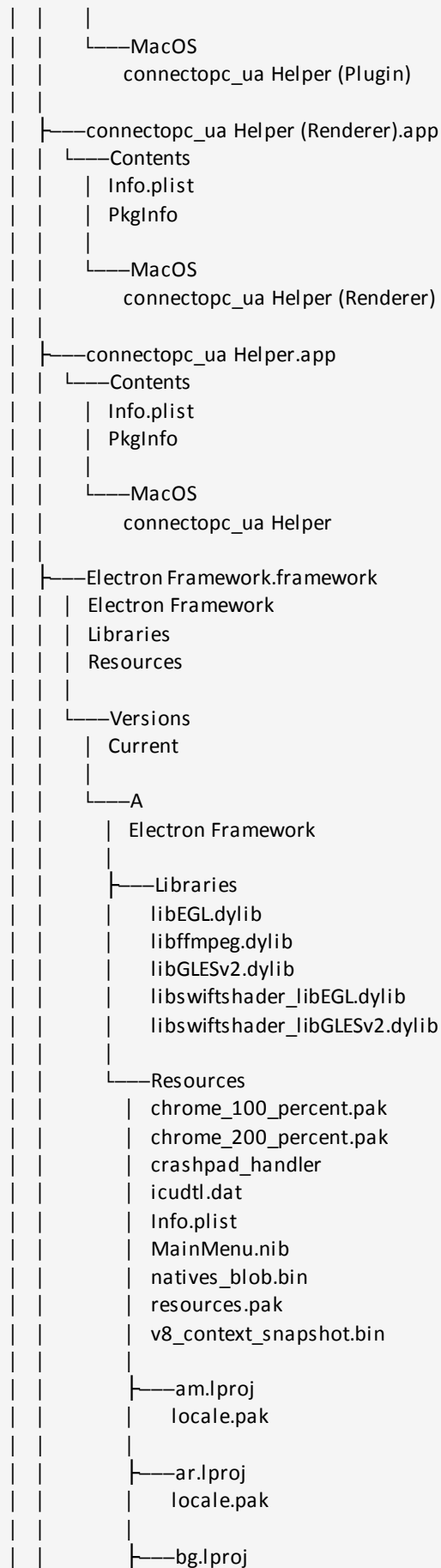
Volume serial number is 9885-ACB9

E:.

```

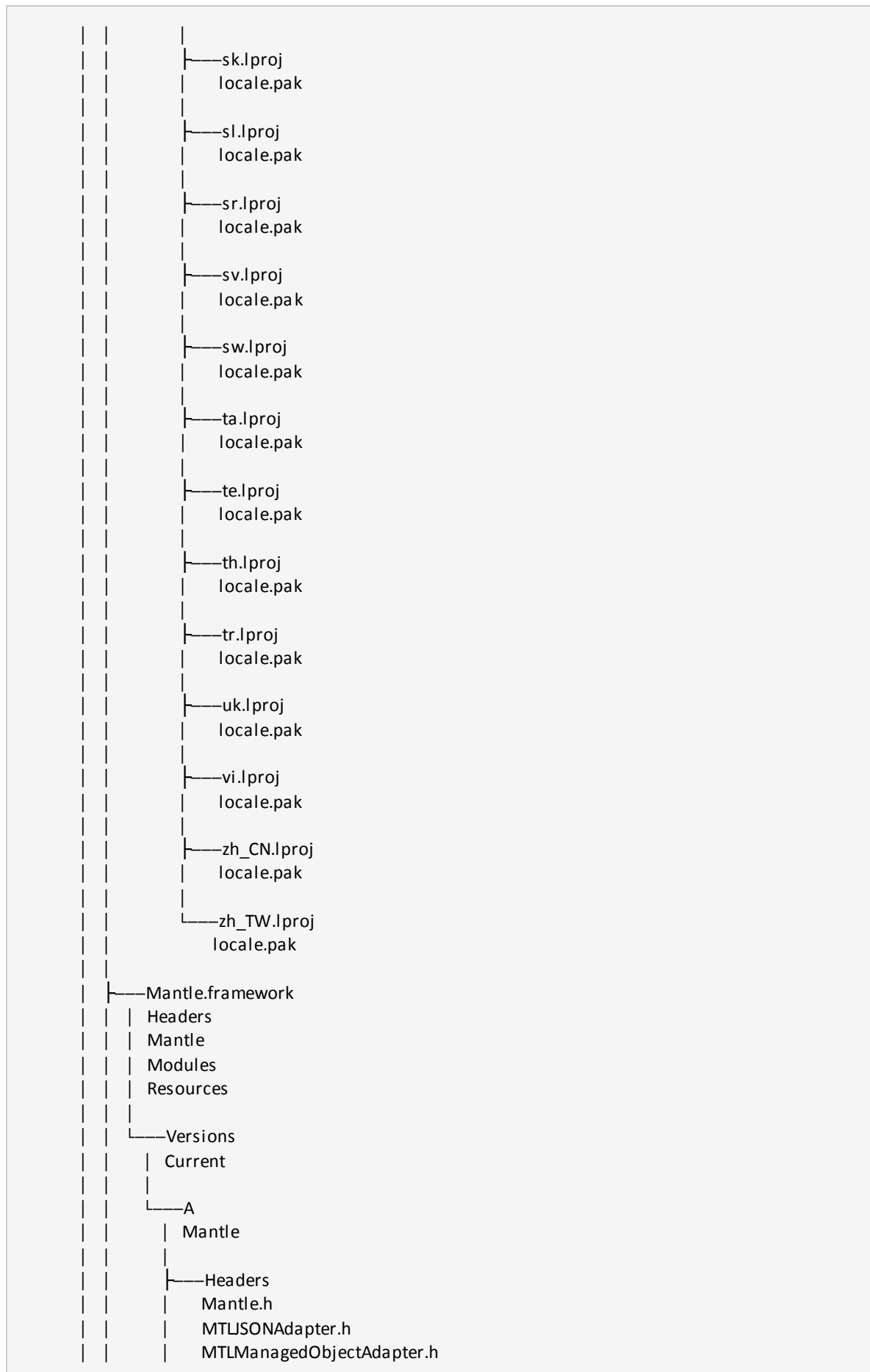
| builder-effective-config.yaml
| connectopc_ua-1.0.0-mac.zip
| connectopc_ua-1.0.0.dmg
| connectopc_ua-1.0.0.dmg.blockmap
| tree.txt
|
|---.icon-icns
|   icon.icns
|
|---mac
|   |---connectopc_ua.app
|       |---Contents
|           | Info.plist
|           | PkgInfo
|           |
|           |---Frameworks
|               |---connectopc_ua Helper (GPU).app
|                   |---Contents
|                       | Info.plist
|                       | PkgInfo
|                       |
|                       |---MacOS
|                           connectopc_ua Helper (GPU)
|               |---connectopc_ua Helper (Plugin).app
|                   |---Contents
|                       | Info.plist
|                       | PkgInfo

```



		locale.pak
		└─bn.lproj locale.pak
		└─ca.lproj locale.pak
		└─cs.lproj locale.pak
		└─da.lproj locale.pak
		└─de.lproj locale.pak
		└─el.lproj locale.pak
		└─en.lproj locale.pak
		└─en_GB.lproj locale.pak
		└─es.lproj locale.pak
		└─es_419.lproj locale.pak
		└─et.lproj locale.pak
		└─fa.lproj locale.pak
		└─fi.lproj locale.pak
		└─fil.lproj locale.pak
		└─fr.lproj locale.pak
		└─gu.lproj locale.pak
		└─he.lproj locale.pak
		└─hi.lproj locale.pak

		hr.lproj
		locale.pak
		hu.lproj
		locale.pak
		id.lproj
		locale.pak
		it.lproj
		locale.pak
		ja.lproj
		locale.pak
		kn.lproj
		locale.pak
		ko.lproj
		locale.pak
		lt.lproj
		locale.pak
		lv.lproj
		locale.pak
		ml.lproj
		locale.pak
		mr.lproj
		locale.pak
		ms.lproj
		locale.pak
		nb.lproj
		locale.pak
		nl.lproj
		locale.pak
		pl.lproj
		locale.pak
		pt_BR.lproj
		locale.pak
		pt_PT.lproj
		locale.pak
		ro.lproj
		locale.pak
		ru.lproj
		locale.pak



```

| | | MTLModel+NSCoding.h
| | | MTLModel.h
| | | MTLValueTransformer.h
| | | NSArray+MTLManipulationAdditions.h
| | | NSDictionary+MTLManipulationAdditions.h
| | | NSObject+MTLComparisonAdditions.h
| | | NSValueTransformer+MTLInversionAdditions.h
| | | NSValueTransformer+MTLPredefinedTransformerAdditions.h
| | |
| | | └─ Modules
| | |     module.modulemap
| | |
| | | └─ Resources
| | |     Info.plist
| | |
| └─ ReactiveCocoa.framework
|   | Headers
|   | Modules
|   | ReactiveCocoa
|   | Resources
|   |
|   └─ Versions
|       | Current
|       |
|       └─ A
|           | ReactiveCocoa
|           |
|           └─ Headers
|               | EXTKeyPathCoding.h
|               | EXTScope.h
|               | metamacros.h
|               | NSArray+RACSequenceAdditions.h
|               | NSControl+RACCommandSupport.h
|               | NSControl+RACTextSignalSupport.h
|               | NSData+RACSupport.h
|               | NSDictionary+RACSequenceAdditions.h
|               | NSEnumerator+RACSequenceAdditions.h
|               | NSFileHandle+RACSupport.h
|               | NSIndexSet+RACSequenceAdditions.h
|               | NSNotificationCenter+RACSupport.h
|               | NSObject+RACAppKitBindings.h
|               | NSObject+RACDeallocating.h
|               | NSObject+RACLifting.h
|               | NSObject+RACPropertySubscribing.h
|               | NSObject+RACSelectorSignal.h
|               | NSOrderedSet+RACSequenceAdditions.h
|               | NSSet+RACSequenceAdditions.h
|               | NSString+RACSequenceAdditions.h
|               | NSString+RACSupport.h
|               | NSText+RACSignalSupport.h
|               | NSURLConnection+RACSupport.h
|               | NSUserDefaults+RACSupport.h
|               | RACBacktrace.h
|               | RACBehaviorSubject.h
|               | RACChannel.h
|               | RACCommand.h

```



```

| | | RACCompoundDisposable.h
| | | RACDisposable.h
| | | RACEvent.h
| | | RACGroupedSignal.h
| | | RACKVOChannel.h
| | | RACMulticastConnection.h
| | | RACQueueScheduler+Subclass.h
| | | RACQueueScheduler.h
| | | RACReplaySubject.h
| | | RACSchedulder+Subclass.h
| | | RACSchedulder.h
| | | RACScopedDisposable.h
| | | RACSequence.h
| | | RACSerialDisposable.h
| | | RACSignal+Operations.h
| | | RACSignal.h
| | | RACStream.h
| | | RACSubject.h
| | | RACSubscriber.h
| | | RACSubscribingAssignmentTrampoline.h
| | | RACTargetQueueScheduler.h
| | | RACTestScheduler.h
| | | RACTuple.h
| | | RACUnit.h
| | | ReactiveCocoa.h
| | |
| | | └── Modules
| | |     module.modulemap
| | |
| | | └── Resources
| | |     Info.plist
| |
| └── Squirrel.framework
|     Headers
|     Modules
|     Resources
|     Squirrel
|
| └── Versions
|     Current
|     |
|     └── A
|         Squirrel
|         |
|         └── Headers
|             NSBundle+SQLVersionExtensions.h
|             NSProcessInfo+SQLVersionExtensions.h
|             SQLDownloadedUpdate.h
|             SQLUpdate.h
|             SQLUpdater.h
|             Squirrel.h
|
|         └── Modules
|             module.modulemap
|
|         └── Resources

```

```
|      Info.plist
|      Shiplt
|
|----- MacOS
|      connectopc_ua
|
|----- Resources
|      app.asar
|      connectopc_ua.icns
|      electron.asar
|
|----- am.lproj
|----- ar.lproj
|----- bg.lproj
|----- bn.lproj
|----- ca.lproj
|----- cs.lproj
|----- da.lproj
|----- de.lproj
|----- el.lproj
|----- en.lproj
|----- en_GB.lproj
|----- es.lproj
|----- es_419.lproj
|----- et.lproj
|----- fa.lproj
|----- fi.lproj
|----- fil.lproj
|----- fr.lproj
|----- gu.lproj
|----- he.lproj
|----- hi.lproj
|----- hr.lproj
|----- hu.lproj
|----- id.lproj
|----- it.lproj
|----- ja.lproj
|----- kn.lproj
|----- ko.lproj
|----- lt.lproj
|----- lv.lproj
|----- ml.lproj
|----- mr.lproj
|----- ms.lproj
|----- nb.lproj
|----- nl.lproj
|----- pl.lproj
|----- pt_BR.lproj
|----- pt_PT.lproj
|----- ro.lproj
|----- ru.lproj
|----- sk.lproj
|----- sl.lproj
|----- sr.lproj
|----- sv.lproj
|----- sw.lproj
```

```

├── ta.lproj
├── te.lproj
├── th.lproj
├── tr.lproj
├── uk.lproj
├── vi.lproj
├── zh_CN.lproj
└── zh_TW.lproj

```

A3 Installation files for Mac OS

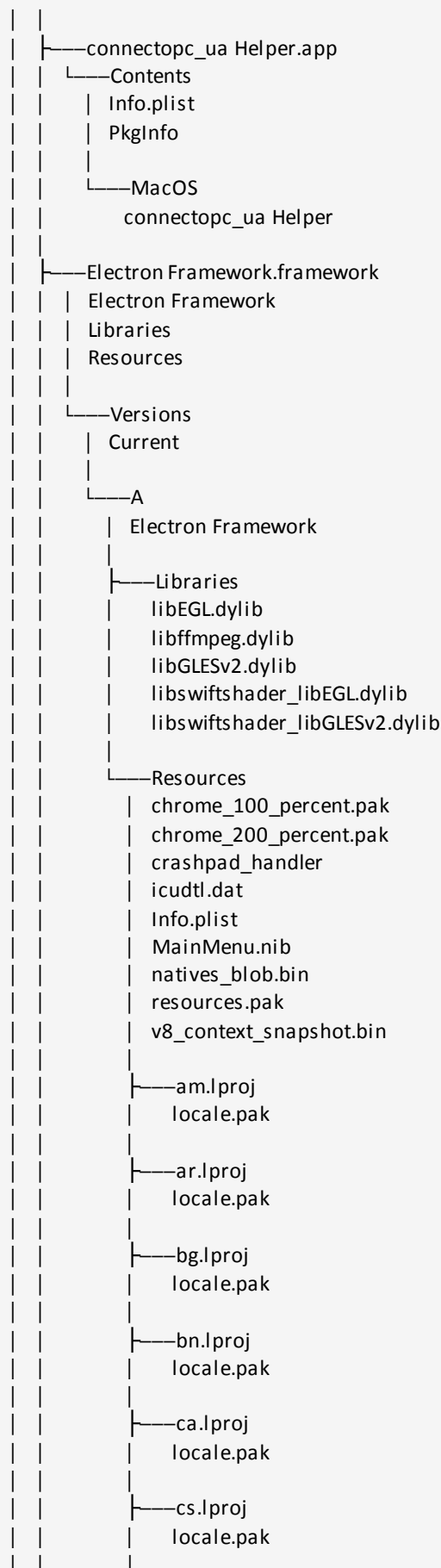
Folder PATH listing for volume New Volume

Volume serial number is 9885-ACB9

E:.

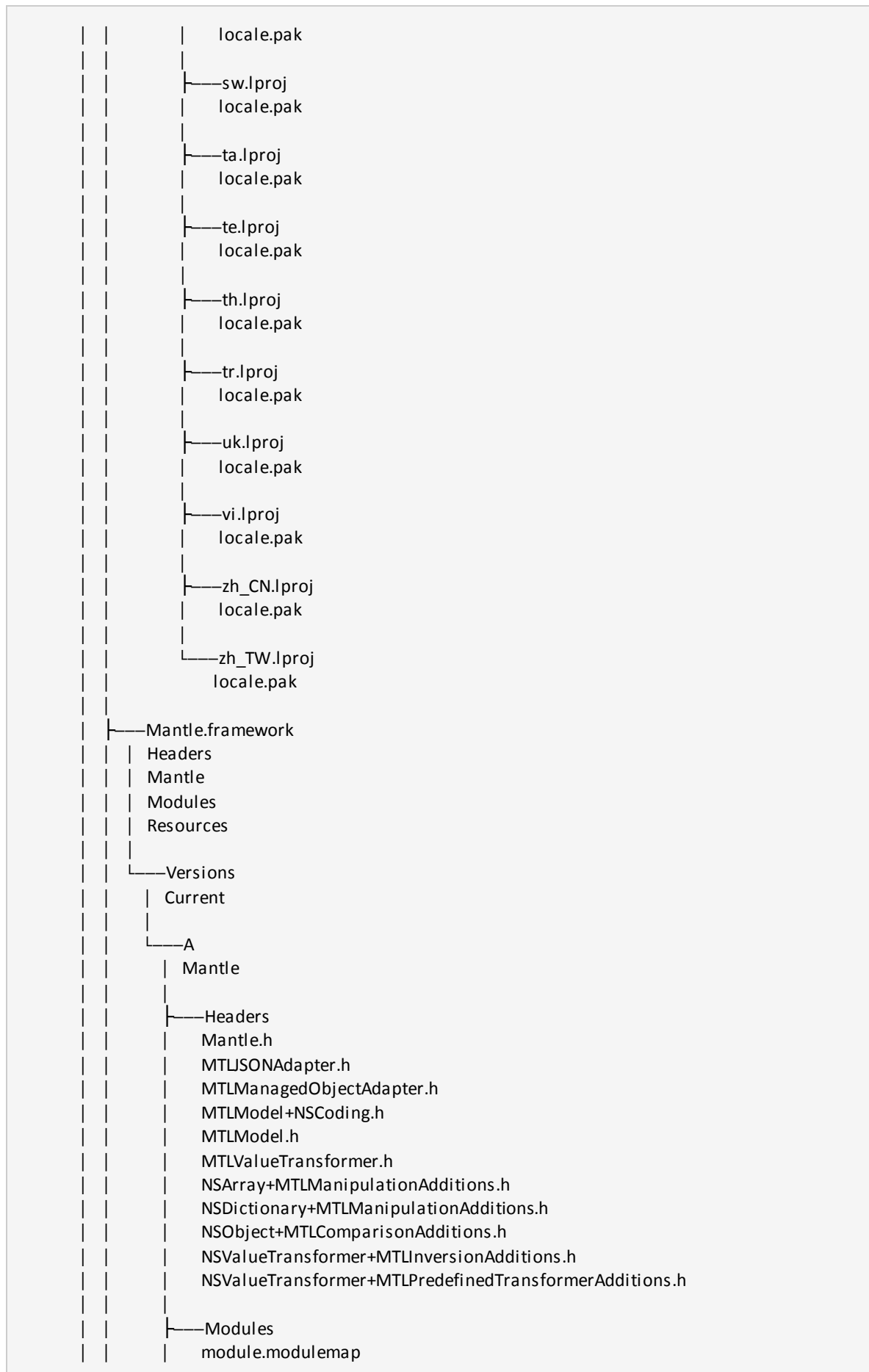
```

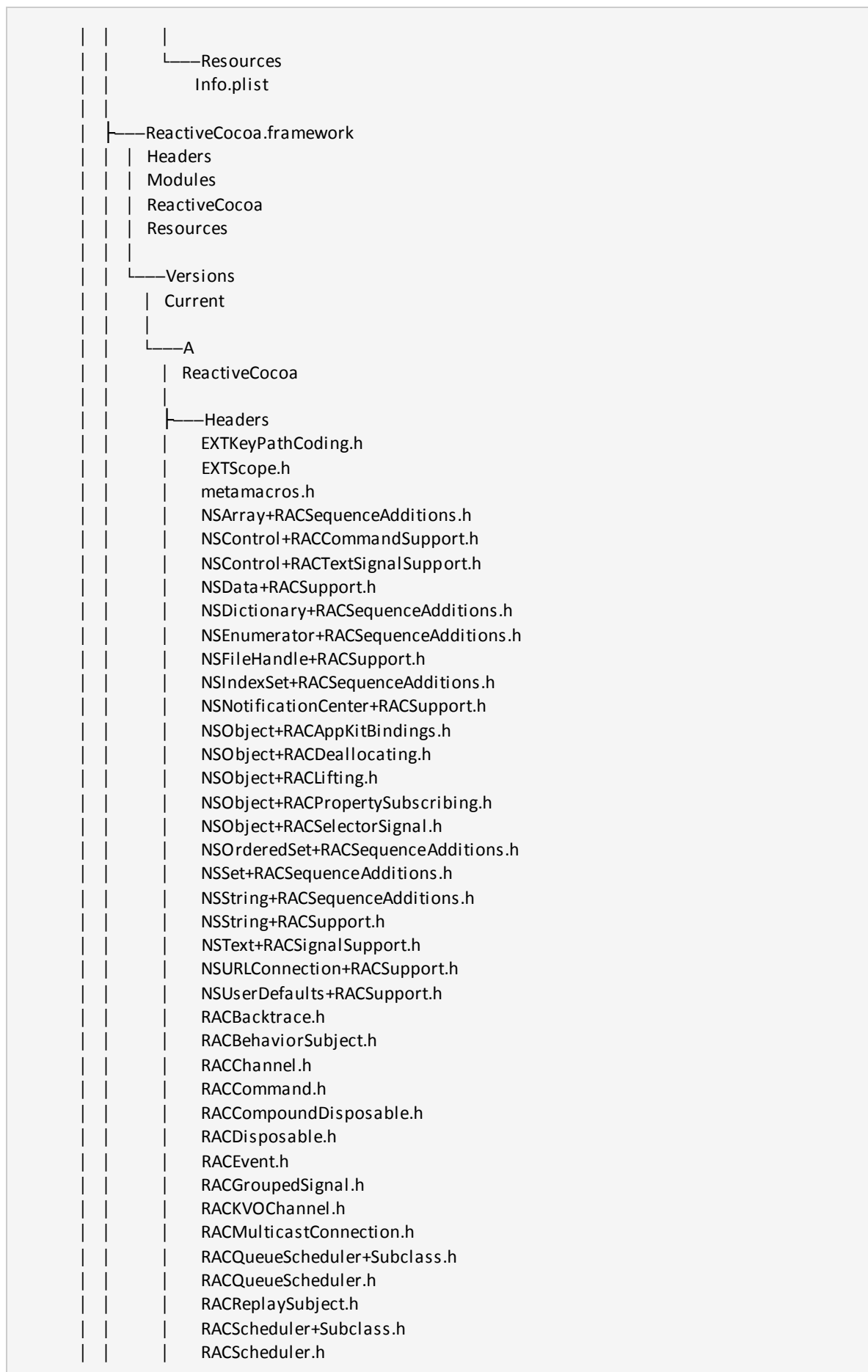
| builder-effective-config.yaml
| connectopc_ua-1.0.0-mac.zip
| connectopc_ua-1.0.0.dmg
| connectopc_ua-1.0.0.dmg.blockmap
| tree.txt
|
├── .icon-icns
|   └── icon.icns
|
└── mac
    ├── connectopc_ua.app
    │   ├── Contents
    │   │   ├── Info.plist
    │   │   └── PkgInfo
    │   │
    │   └── Frameworks
    │       ├── connectopc_ua Helper (GPU).app
    │       │   ├── Contents
    │       │   │   ├── Info.plist
    │       │   │   └── PkgInfo
    │       │   └── MacOS
    │       │       connectopc_ua Helper (GPU)
    │       ├── connectopc_ua Helper (Plugin).app
    │       │   ├── Contents
    │       │   │   ├── Info.plist
    │       │   │   └── PkgInfo
    │       │   └── MacOS
    │       │       connectopc_ua Helper (Plugin)
    │       ├── connectopc_ua Helper (Renderer).app
    │       │   ├── Contents
    │       │   │   ├── Info.plist
    │       │   │   └── PkgInfo
    │       │   └── MacOS
    │       │       connectopc_ua Helper (Renderer)
    
```



		da.lproj
		locale.pak
		de.lproj
		locale.pak
		el.lproj
		locale.pak
		en.lproj
		locale.pak
		en_GB.lproj
		locale.pak
		es.lproj
		locale.pak
		es_419.lproj
		locale.pak
		et.lproj
		locale.pak
		fa.lproj
		locale.pak
		fi.lproj
		locale.pak
		fil.lproj
		locale.pak
		fr.lproj
		locale.pak
		gu.lproj
		locale.pak
		he.lproj
		locale.pak
		hi.lproj
		locale.pak
		hr.lproj
		locale.pak
		hu.lproj
		locale.pak
		id.lproj
		locale.pak
		it.lproj
		locale.pak

		ja.lproj
		locale.pak
		kn.lproj
		locale.pak
		ko.lproj
		locale.pak
		lt.lproj
		locale.pak
		lv.lproj
		locale.pak
		ml.lproj
		locale.pak
		mr.lproj
		locale.pak
		ms.lproj
		locale.pak
		nb.lproj
		locale.pak
		nl.lproj
		locale.pak
		pl.lproj
		locale.pak
		pt_BR.lproj
		locale.pak
		pt_PT.lproj
		locale.pak
		ro.lproj
		locale.pak
		ru.lproj
		locale.pak
		sk.lproj
		locale.pak
		sl.lproj
		locale.pak
		sr.lproj
		locale.pak
		sv.lproj






```

| | | RACScopedDisposable.h
| | | RACSequence.h
| | | RACSerialDisposable.h
| | | RACSignal+Operations.h
| | | RACSignal.h
| | | RACStream.h
| | | RACSubject.h
| | | RACSubscriber.h
| | | RACSubscribingAssignmentTrampoline.h
| | | RACTargetQueueScheduler.h
| | | RACTestScheduler.h
| | | RACTuple.h
| | | RACUnit.h
| | | ReactiveCocoa.h
| | |
| | | └─ Modules
| | |     module.modulemap
| | |
| | | └─ Resources
| | |     Info.plist
| | |
| └─ Squirrel.framework
|   | Headers
|   | Modules
|   | Resources
|   | Squirrel
|   |
|   └─ Versions
|       | Current
|       |
|       └─ A
|           | Squirrel
|           |
|           └─ Headers
|               | NSBundle+SQRLVersionExtensions.h
|               | NSProcessInfo+SQRLVersionExtensions.h
|               | SQRLDownloadedUpdate.h
|               | SQRLUpdate.h
|               | SQRLUpdater.h
|               | Squirrel.h
|               |
|               └─ Modules
|                   module.modulemap
|               |
|               └─ Resources
|                   Info.plist
|                   Shiplt
|
| └─ MacOS
|     connectopc_ua
|
| └─ Resources
|     app.asar
|     connectopc_ua.icns
|     electron.asar

```

- └─am.lproj
- └─ar.lproj
- └─bg.lproj
- └─bn.lproj
- └─ca.lproj
- └─cs.lproj
- └─da.lproj
- └─de.lproj
- └─el.lproj
- └─en.lproj
- └─en_GB.lproj
- └─es.lproj
- └─es_419.lproj
- └─et.lproj
- └─fa.lproj
- └─fi.lproj
- └─fil.lproj
- └─fr.lproj
- └─gu.lproj
- └─he.lproj
- └─hi.lproj
- └─hr.lproj
- └─hu.lproj
- └─id.lproj
- └─it.lproj
- └─ja.lproj
- └─kn.lproj
- └─ko.lproj
- └─lt.lproj
- └─lv.lproj
- └─ml.lproj
- └─mr.lproj
- └─ms.lproj
- └─nb.lproj
- └─nl.lproj
- └─pl.lproj
- └─pt_BR.lproj
- └─pt_PT.lproj
- └─ro.lproj
- └─ru.lproj
- └─sk.lproj
- └─sl.lproj
- └─sr.lproj
- └─sv.lproj
- └─sw.lproj
- └─ta.lproj
- └─te.lproj
- └─th.lproj
- └─tr.lproj
- └─uk.lproj
- └─vi.lproj
- └─zh_CN.lproj
- └─zh_TW.lproj