

Proslys OPC UA Simulation Server – User Manual

Version: 2.2.2

Table of Contents

Installing the Application	2
Uninstalling the Application	2
About the Server	3
Objects	3
Types	3
Views	3
User Interface	4
Application Menu	4
Status	4
Endpoints	5
Registering to a Local Discovery Server	5
Users	6
Sessions	7
Certificates	8
Connection Log	9
Address Space	10
Simulation	11
Debug Log	12
Reg/Res Log	13
File locations	14

Proslys OPC UA Simulation Server – User Manual

OPC UA Simulation Server is an OPC UA server application, which provides simulated data. You can use it in place of OPC UA servers that provide online production data, for example, to test connections from different OPC UA client applications or to help you with your OPC UA system or application development.

Installing the Application

The installation includes a complete “embedded” Java Runtime Environment (JRE). This ensures that you don’t need to install Java in your computer, although the application is running in a Java environment, and you don’t need to worry about the Java updates. The embedded Java is only used for this application.

The application install packages are available from <http://www.prosysopc.com> upon request. You should get the correct package, depending on your target environment.

On Windows, run the installer executable and follow the instructions. By default, the application is installed with normal user privileges under your home directory. If you want to install it to, for example, Program Files, you need to run the installer as user which has *administrator* privileges.

On OS X, you can just install it normally from the dmg-package. Just note that the application is not signed, so you need to accept it at the first startup. Since OSX 10.8 (Mountain Lion), this requires that you open the application using the right-click menu - Open. You can then accept the application to be run, although it is not signed. After the first startup, you can run it normally from the Launch Pad as well.¹

On Debian-based Linux (such as Ubuntu), use

```
sudo dpkg -i prosys-opc-ua-simulation-server-x.x.x-x.deb
```

On RPM-based Linux (such as Fedora), use use

```
sudo rpm -i prosys-opc-ua-simulation-server-x.x.x-x.x86_64.rpm
```

(If the rpm install fails with "error: failed dependencies" then you are most likely running a Debian-based Linux distribution.)

Uninstalling the Application

On Windows the application can be uninstalled through the Control Panel, or with the uninstaller (in the installation folder). Remember to stop the application before uninstalling (some files cannot be removed while the application is running and must be removed manually if uninstalling is done while the application is running).

On OS X you can just remove the application from the Applications folder.

On Debian-based Linux use

```
sudo dpkg -r prosys-opc-ua-simulation-server
```

On RPM-based Linux use

```
sudo rpm -e prosys-opc-ua-simulation-server
```

¹ See <http://www.wikihow.com/Install-Software-from-Unsigned-Developers-on-a-Mac> for more information about the Apple Gatekeeper options.

About the Server

The Simulation Server is an OPC UA server that contains simulation signals for demonstration and testing purposes.

As most OPC UA servers, the address space of the Simulation Server has three main folders: *Objects*, *Types* and *Views*. *Objects* include instances of OPC UA nodes, defined in *Types* folder. *Views* can include subgroups of OPC UA address space

The address space can be explored with the Address Space view. It is also available for all client applications, such as Proslys OPC UA Java Client.

See the Address Space part for more about the address space and how to browse it in the server.

Objects

The Simulation Server defines the following objects:

- *Server* object is a standard object that provides the status and capability information about the server
- *Simulation* folder contains all simulation signals defined in the Simulation page
- *MyObjects* contains a sample object, *MyDevice* that simulates a simple device with 4 components
 - *MySwitch* is a simple switch variable (DataType=Boolean) that can be turned on and off
 - *MyLevel* is a variable (DataType=Double) for simulated level measurement
 - *MyLevelAlarm* is an alarm object corresponding to *MyLevel*. New events are sent from the server when the value of *MyLevel* exceeds the alarm limits defined in the alarm object.
 - *MyMethod* is a sample method that can be called from the client applications
- *MyBigNodeManager* represents an object including a large number of data items
- *StaticData* object includes variables of various data types that change only when they are written to

Types

The OPC UA servers provide complete type information for the objects also in the address space. These can be found from the Types folder. The Simulation Server includes the following types:

- *ObjectTypes* includes type definitions for all the objects in the server address space
- *VariableTypes* includes type definitions for all the variables in the server address space
- *DataTypes* defines all the data types that are used in the server address space
- *ReferenceTypes* defines all the reference types that appear between nodes
- *EventTypes* define all the events that appear in server address space

Views

The Simulation Server does not define any *Views*.

User Interface

The user interface of the Simulation Server consists of several views which are organized into tabbed pages.

Application Menu

The Application Menu has just a couple of functions, you can shut down the server with File->Shutdown (or you can just close the window). Help->About lists information about the application (e.g. version numbers).

Status

The status page (Figure 1) displays short information about the current server status. If everything is ok, the status displays “Running”. The view also displays the current and starting time of the server. If remote connections are used, you should ensure that the computers are running with synchronized clocks. Especially the secure connections will not work if the clocks are too much out of sync. The Server Status and these times are also available for client application from the Server object.

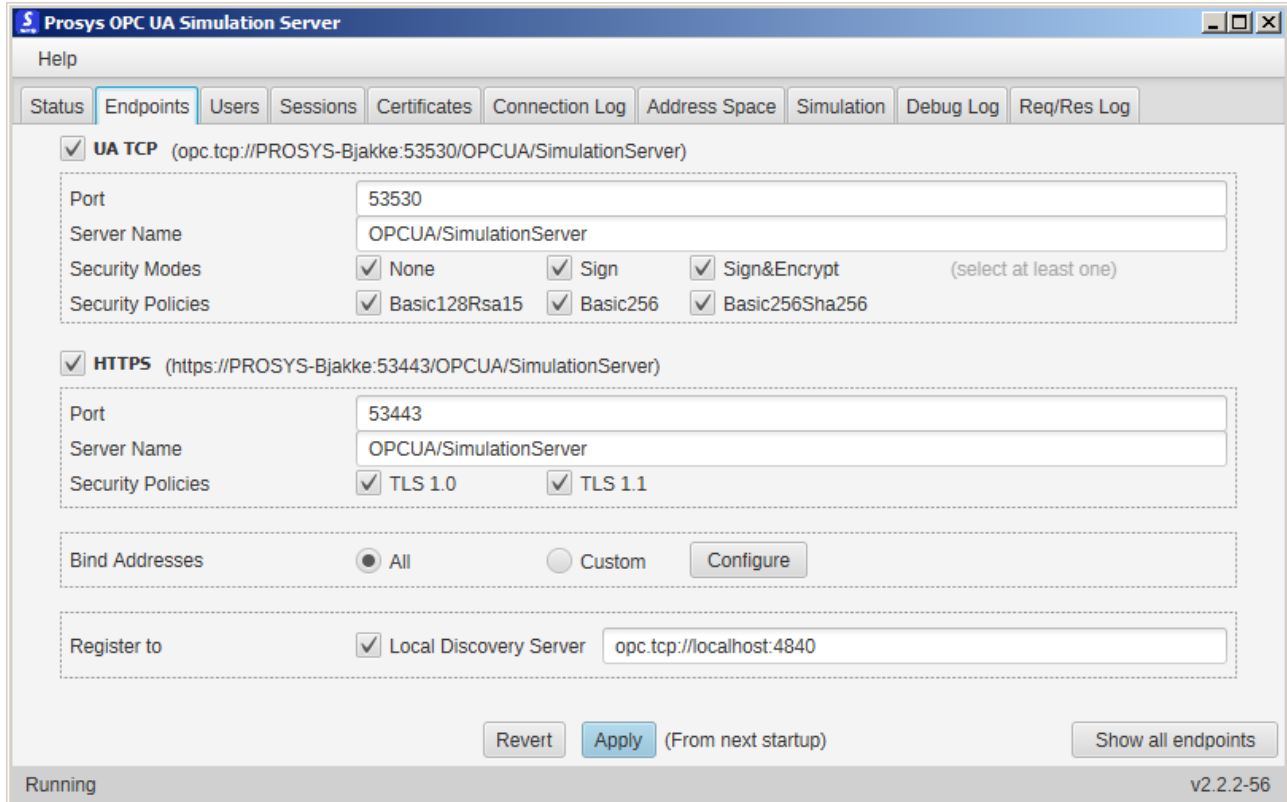
The Connection Addresses show which addresses (URLs) the client applications can use to connect to the server. Note that the Simulation Server supports both *opc.tcp* and *https* protocols. Most client applications only support *opc.tcp*, since *https* was defined recently for OPC UA 1.02. The Simulation Server does not support the soap protocol, which is also an alternative for some (mostly .NET based) OPC UA applications. You can define the available addresses in the Endpoints page.



Figure 1: The Status tab displays some basic info about the server

Endpoints

The Endpoints page (Figure 2) allows you to configure all endpoint related settings and view all endpoints that the server has. By clicking “Show all endpoints” button, you see a pop-up window, which displays all the currently defined endpoints. You can edit the settings and save them by clicking “Apply”. The settings are applied the next time the application is started.



The screenshot shows the 'Proslys OPC UA Simulation Server' application window with the 'Endpoints' tab selected. The window has a menu bar with 'Help' and a tab bar with 'Status', 'Endpoints', 'Users', 'Sessions', 'Certificates', 'Connection Log', 'Address Space', 'Simulation', 'Debug Log', and 'Req/Res Log'. The 'Endpoints' tab contains two main sections for configuring endpoints: 'UA TCP' and 'HTTPS'. Each section has fields for 'Port', 'Server Name', and 'Security Modes/Security Policies'. The 'UA TCP' section is for 'opc.tcp://PROSYS-Bjake:53530/OPCUA/SimulationServer' and the 'HTTPS' section is for 'https://PROSYS-Bjake:53443/OPCUA/SimulationServer'. Below these sections are 'Bind Addresses' (All/Custom) and 'Register to' (Local Discovery Server) options. At the bottom, there are 'Revert', 'Apply (From next startup)', and 'Show all endpoints' buttons. The status bar at the bottom indicates 'Running' and 'v2.2.2-56'.

Proslys OPC UA Simulation Server

Help

Status Endpoints Users Sessions Certificates Connection Log Address Space Simulation Debug Log Req/Res Log

☒ **UA TCP** (opc.tcp://PROSYS-Bjake:53530/OPCUA/SimulationServer)

Port: 53530

Server Name: OPCUA/SimulationServer

Security Modes: ☒ None ☒ Sign ☒ Sign&Encrypt (select at least one)

Security Policies: ☒ Basic128Rsa15 ☒ Basic256 ☒ Basic256Sha256

☒ **HTTPS** (https://PROSYS-Bjake:53443/OPCUA/SimulationServer)

Port: 53443

Server Name: OPCUA/SimulationServer

Security Policies: ☒ TLS 1.0 ☒ TLS 1.1

Bind Addresses: ☒ All ☐ Custom

Register to: ☒ Local Discovery Server

(From next startup)

Running v2.2.2-56

Figure 2: The Endpoints page allows you to configure the endpoints of the server

Registering to a Local Discovery Server

The endpoints page (Figure 2) also has controls for registering the server to a Local Discovery Server. Please see the OPC Unified Architecture specification Part 12 for more information about Local Discovery Server. The view allows enabling/disabling the registration and changing the connection address for the Local Discovery Server. Please note that the registration requires a secure connection therefore the discovery server needs to trust Simulation Server's certificate.

Users

The Users page (Figure 3) displays user authentication settings. OPC UA has multiple options for authentication. Currently only Anonymous and Username/Password configurations are supported. (All user certificates and tokens are actually accepted.). **Note!** that the changes in User Authentication Methods apply only after restarting the server.

You can use the user list to define user names and passwords that are allowed access to the server. Currently all users get equal access rights to the server.

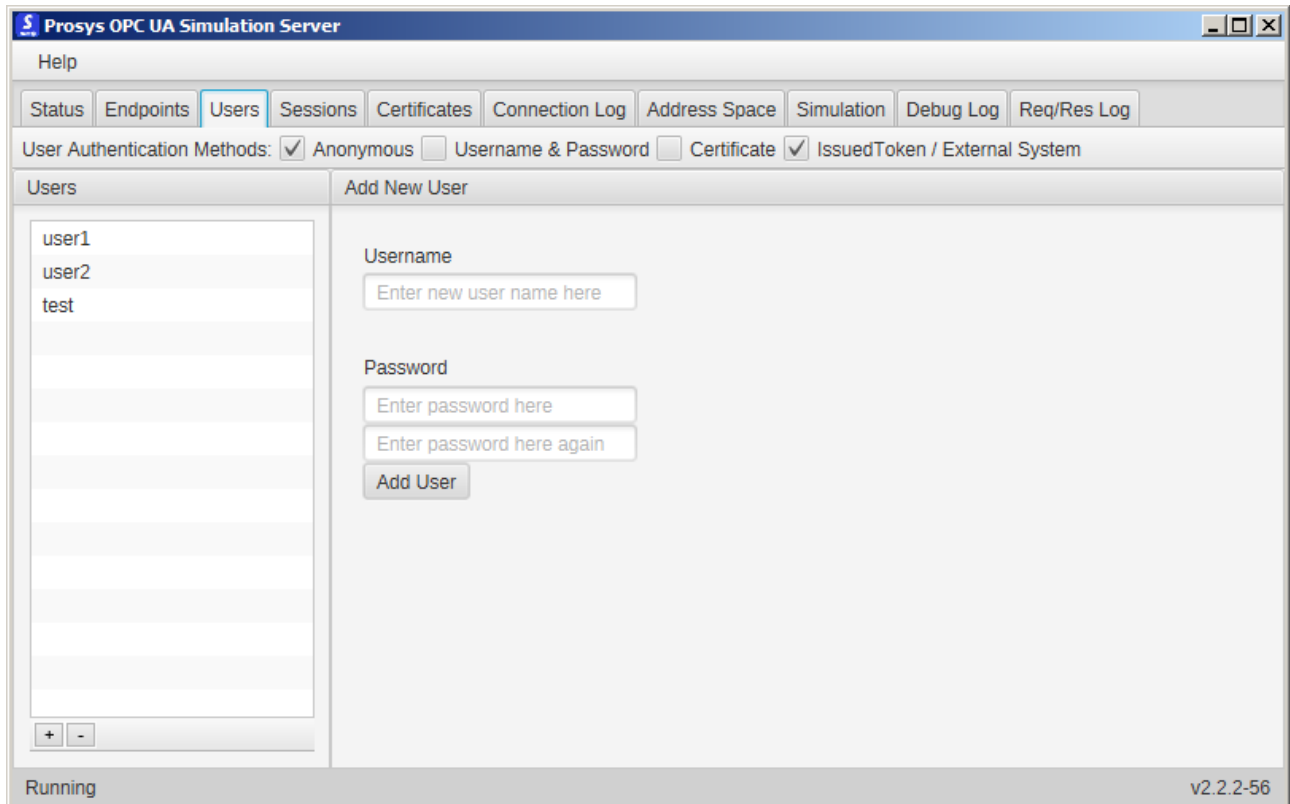


Figure 3: Users page

Sessions

The Sessions page (Figure 4) displays the current sessions opened by the client application. The Session list on the left shows the session names and the detail view on the right displays available information about the selected session.

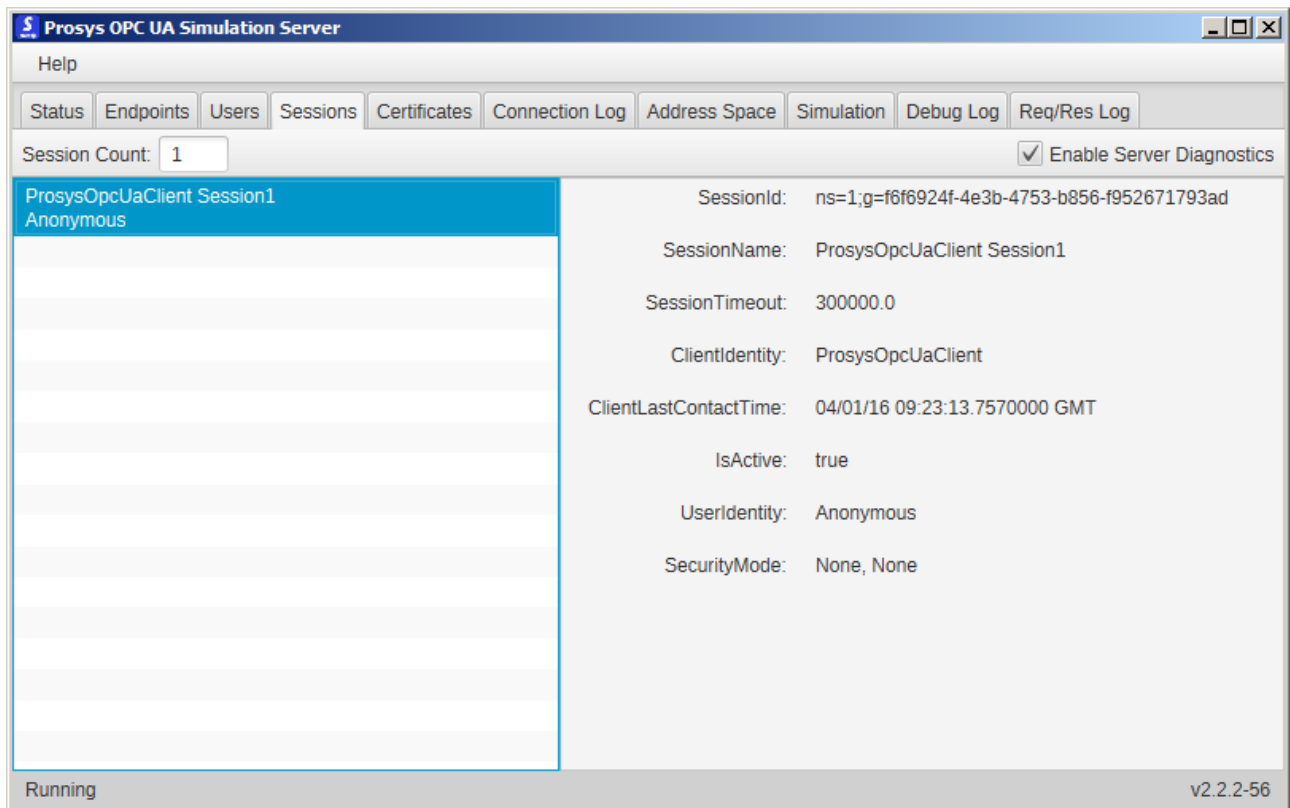


Figure 4: Session tab displays all current Sessions in the server

Certificates

OPC UA applications use *application instance certificates* (not to be confused with user certificates) to authenticate the applications that are allowed to make secure connections (for unsecure connections, the certificates are not used).

The Certificates page (Figure 5) allows you to define which application certificates you trust. When a new client application connects, its certificate will be added to the certificate list as “Rejected”. This is done by right-clicking the Certificate in the left list and selecting Trust from the menu. You can use the same to later reject a certificate that has been trusted previously.

If your operating system has a default application for displaying certificates, you can use that by clicking “Open Certificates in OS viewer”. The certificates are stored in a Certificate Store (PKI folder) in the application settings (See File locations at the end of the manual).

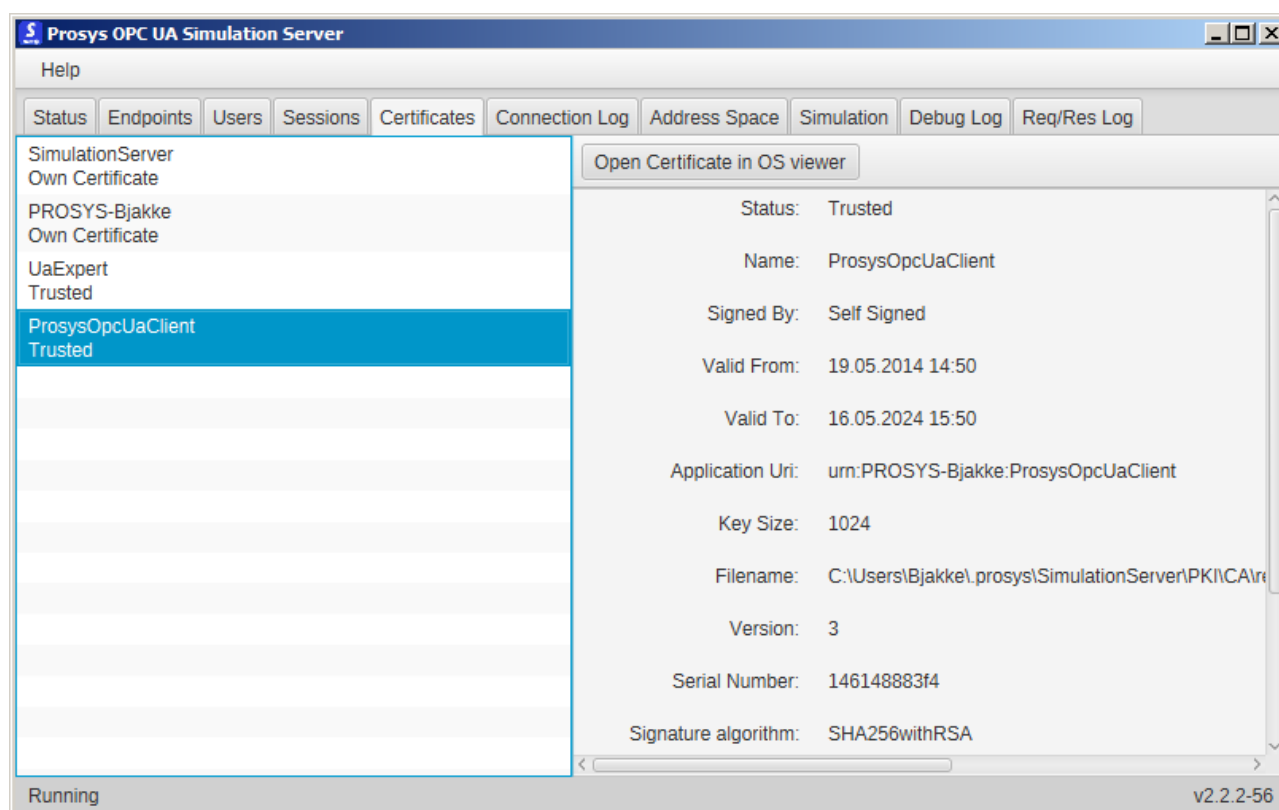
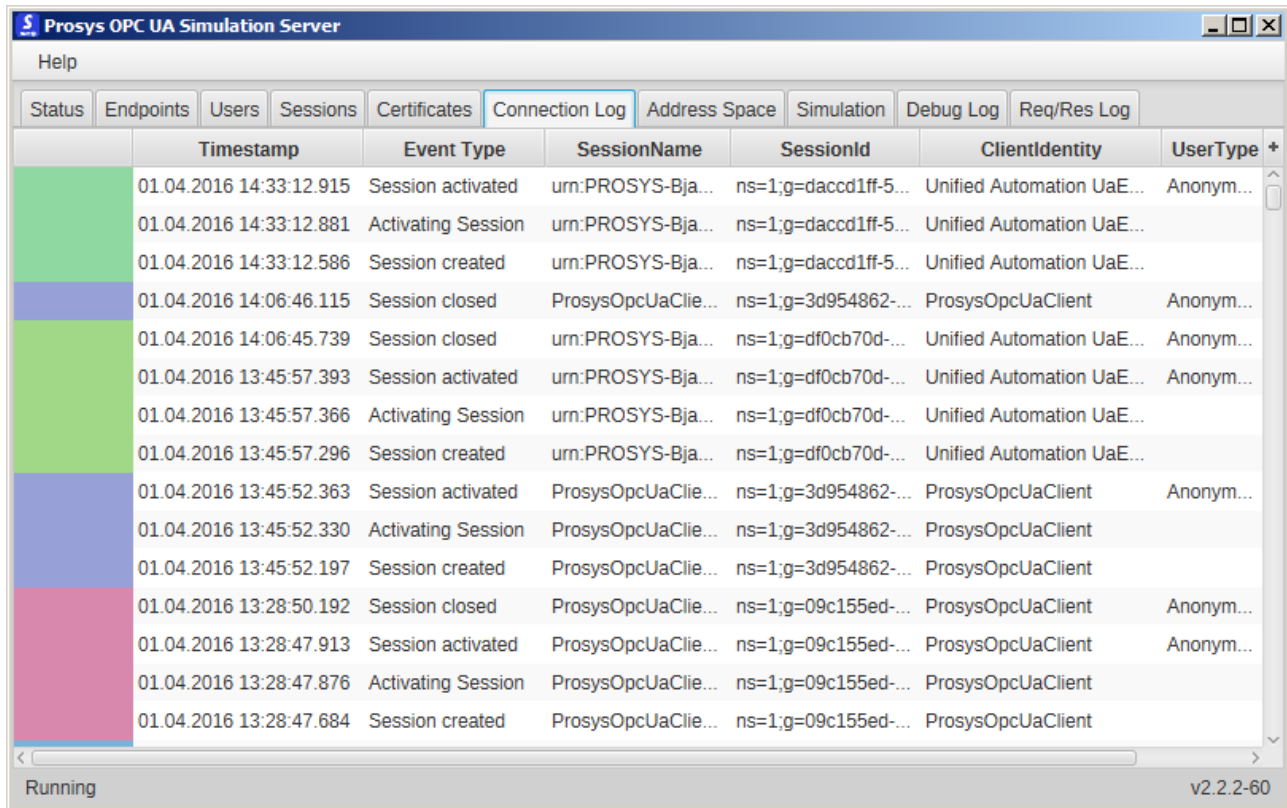


Figure 5: Certificates tab allows you to define which certificates you trust.

Connection Log

The Connection Log page (Figure 6) displays a history of client connections. The log is stored in an internal database, so it will also show past connections.



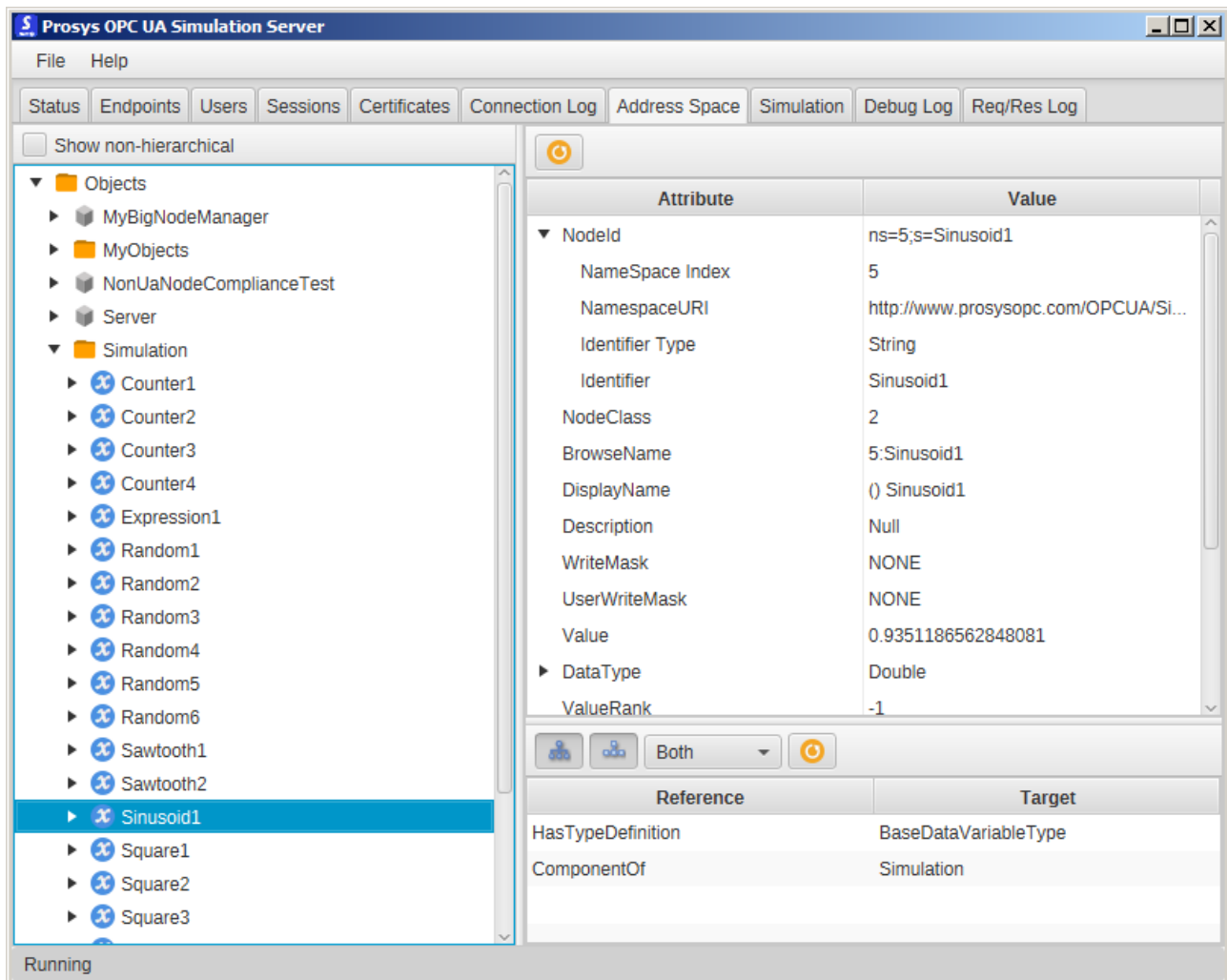
	Timestamp	Event Type	SessionName	SessionId	ClientIdentity	UserType
Green	01.04.2016 14:33:12.915	Session activated	urn:PROSYS-Bja...	ns=1;g=daccd1ff-5...	Unified Automation UaE...	Anonym...
	01.04.2016 14:33:12.881	Activating Session	urn:PROSYS-Bja...	ns=1;g=daccd1ff-5...	Unified Automation UaE...	
	01.04.2016 14:33:12.586	Session created	urn:PROSYS-Bja...	ns=1;g=daccd1ff-5...	Unified Automation UaE...	
Blue	01.04.2016 14:06:46.115	Session closed	ProslysOpcUaClie...	ns=1;g=3d954862-...	ProslysOpcUaClient	Anonym...
Green	01.04.2016 14:06:45.739	Session closed	urn:PROSYS-Bja...	ns=1;g=df0cb70d-...	Unified Automation UaE...	Anonym...
	01.04.2016 13:45:57.393	Session activated	urn:PROSYS-Bja...	ns=1;g=df0cb70d-...	Unified Automation UaE...	Anonym...
	01.04.2016 13:45:57.366	Activating Session	urn:PROSYS-Bja...	ns=1;g=df0cb70d-...	Unified Automation UaE...	
Green	01.04.2016 13:45:57.296	Session created	urn:PROSYS-Bja...	ns=1;g=df0cb70d-...	Unified Automation UaE...	
	01.04.2016 13:45:52.363	Session activated	ProslysOpcUaClie...	ns=1;g=3d954862-...	ProslysOpcUaClient	Anonym...
	01.04.2016 13:45:52.330	Activating Session	ProslysOpcUaClie...	ns=1;g=3d954862-...	ProslysOpcUaClient	
Blue	01.04.2016 13:45:52.197	Session created	ProslysOpcUaClie...	ns=1;g=3d954862-...	ProslysOpcUaClient	
	01.04.2016 13:28:50.192	Session closed	ProslysOpcUaClie...	ns=1;g=09c155ed-...	ProslysOpcUaClient	Anonym...
	01.04.2016 13:28:47.913	Session activated	ProslysOpcUaClie...	ns=1;g=09c155ed-...	ProslysOpcUaClient	Anonym...
Pink	01.04.2016 13:28:47.876	Activating Session	ProslysOpcUaClie...	ns=1;g=09c155ed-...	ProslysOpcUaClient	
	01.04.2016 13:28:47.684	Session created	ProslysOpcUaClie...	ns=1;g=09c155ed-...	ProslysOpcUaClient	

Figure 6: Connection Log tab displays the connection history

Address Space

The address space of a server includes the data and types that are available for client applications. The address space consists of *nodes*, which can be objects, variables, various types or views. The nodes are connected to each other with *references*. The *hierarchical references* (such as Organizes, HasComponent, HasProperty, etc.) are used to compose the tree-structure, which is typically used when displaying the address space. In addition, there can be various non-hierarchical references (such as HasTypeDefinition, HasCondition, etc.).

The Address Space page (Figure 7) can be used to explore the Address Space of the server. You may be familiar with a similar view in some OPC UA Client applications (such as Proslys OPC UA Client or UaExpert). The left part shows the Nodes in a tree view and the right-hand side shows the Attributes and References of the selected node.



The screenshot shows the 'Proslys OPC UA Simulation Server' interface. The 'Address Space' tab is active. On the left, a tree view shows the hierarchy: Objects > MyObjects > Sinusoid1. The right pane displays the attributes and references for the selected 'Sinusoid1' node.

Attribute	Value
NodeId	ns=5;s=Sinusoid1
Namespace Index	5
NamespaceURI	http://www.prosysopc.com/OPCUA/Si...
Identifier Type	String
Identifier	Sinusoid1
NodeClass	2
BrowseName	5:Sinusoid1
DisplayName	() Sinusoid1
Description	Null
WriteMask	NONE
UserWriteMask	NONE
Value	0.9351186562848081
DataType	Double
ValueRank	-1

Reference	Target
HasTypeDefinition	BaseDataVariableType
ComponentOf	Simulation

Figure 7: Address Space of the server. The attributes and references of the selected node are displayed on the right.

Simulation

The Simulation page (figure 8) enables configuring custom signals with various simulation waveforms.

“Run Simulation” is used to enable/disable simulation. Interval defines how often the signals are updated. Simulation Time shows the timestamp corresponding to the last update. Simulation can be run in the current time only.

The defined signals are displayed on the left. The parameters of the selected signal are configured with the editor on the right-hand side. The parameters depend on the type of the signal.

You can add and remove signals using the “+” and “-”-buttons below the signal list. The signal names must be unique. The name is also used to define the NodeId, BrowseName and DisplayName of the signal.

The graph on the bottom of the page can be used to visualize the selected signals (the ones that have their Visualize-checkbox checked in the list).

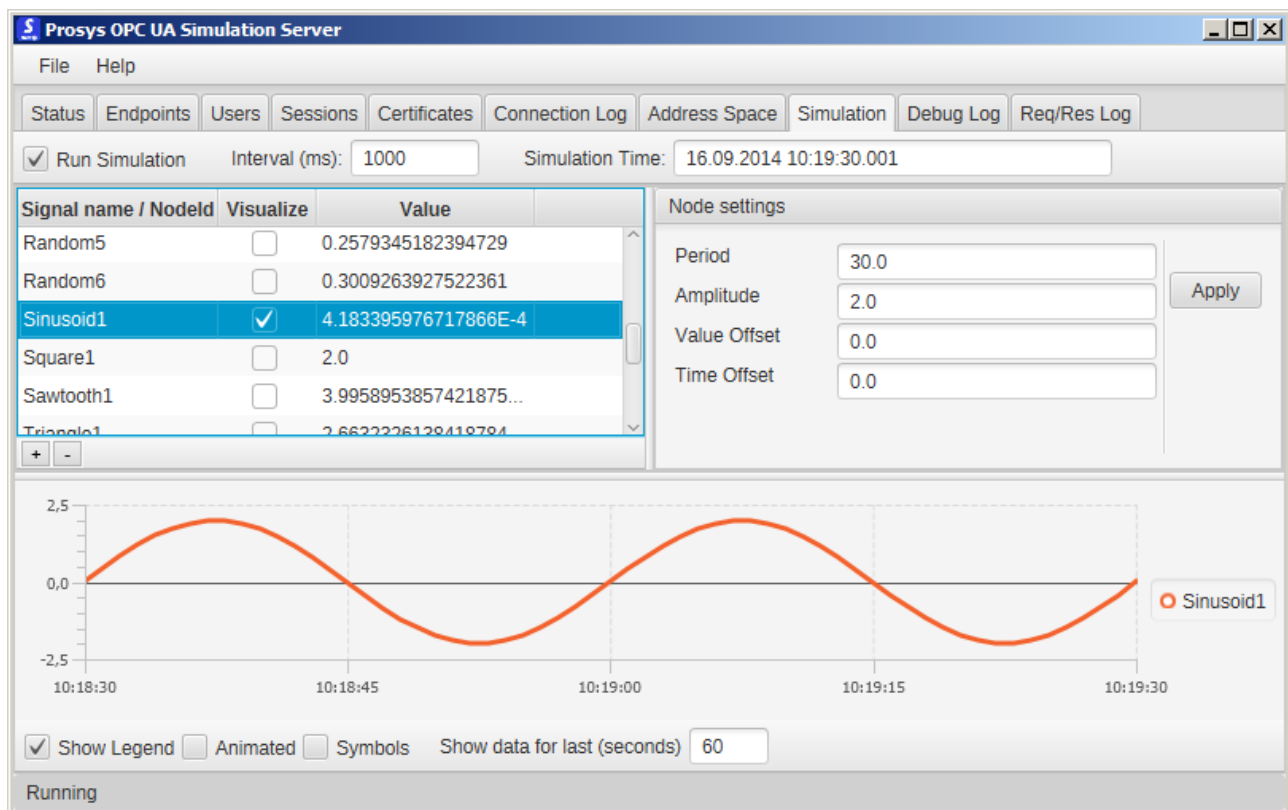


Figure 8: Simulation view

Expression signals can be defined with a custom mathematical *expression*. The expression may also refer to other signals, which are defined with the Inputs parameter. Expression signals cannot be used as inputs for other expressions.

Debug Log

The Debug Log page displays logging from the application's internal behavior. This can be used for examining the application behavior in detail, e.g. if the application is unable to start the server.

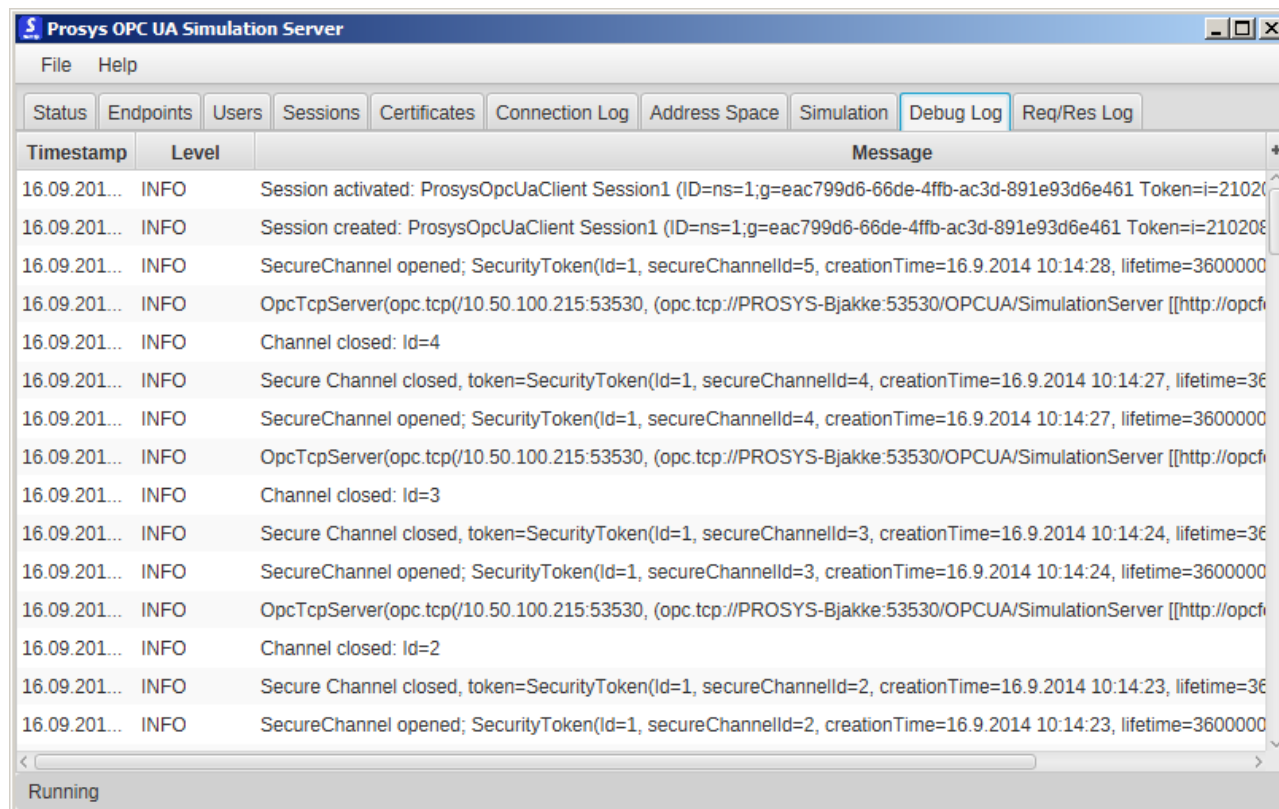


Figure 9: Debug Log view

Reg/Res Log

The Ref/Res Log page can be used to record every request and response to the server (except opening and closing of the secure channel). This functionality is by default off, check the Active checkbox in order to start recording. Note that the recording is kept in memory until cleared by pressing Clear button or shutting down the application, therefore if kept on for very long time, it is possible to run out of memory.

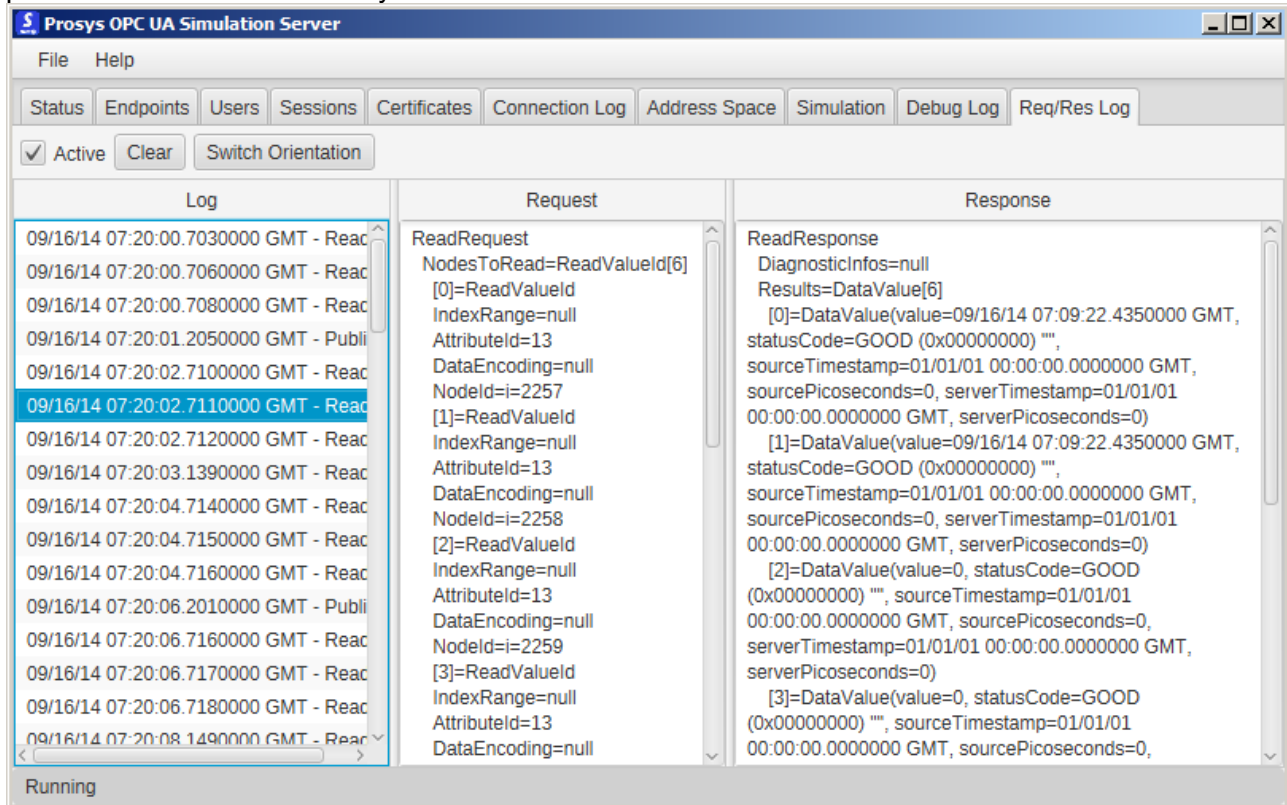


Figure 10: Request-Response Log view

File locations

The application saves its settings to a folder in path “(user.home)/.prosys/SimulationServer”. The (user.home) is the location of the current user’s home folder. If you want to reset to default settings, just delete the settings file in the folder. Settings are saved automatically when the application closes. The certificates used in OPC UA communication are saved to PKI –subfolder in the same location.