OPC UA SimApi User Guide

February 21, 2022

Contents

1 Introduction

This document is the user guide for the OPC UA SimApi from Sartorius Stedim Data Analytics.

A SimApi is the connection between the Umetrics® Suite and external data sources.

The OPC UA SimApi described in this document is the connector between an OPC UA (Unified Architecture) Server and SIMCA®-online Solution and SIMCA® Multivariate Data Analysis Solution.

For a detailed list of changes in different versions of this SimApi, see the **Version Info.txt** file that comes with the installation.

For more information on SimApis, see sartorius.com/umetrics-simapi.

1.1 Terms

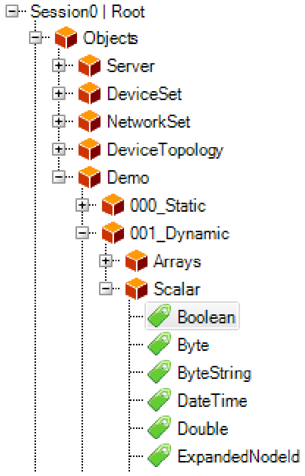
OPC UA name	Comment
OPC UA node	Can correspond to a node or tag in SIMCA-online depending on the OPC UA node class.
OPC UA node class	The type of node. Can be Object, Variable or other types.
Object (an OPC node of node class Object)	Exposed as a node by the SimApi
Variable (an OPC node of node class Variable)	Exposed as a tag by the SimApi

1.2 Features

The SimApi implements the following SimApi features; Refer to sartorius.com/umetrics-simapi to learn more about the general SimApi features.

- Connect to an OPC UA server (using the OPC UA binary protocol), trying the best available security policy automatically creating a client certificate sent to the server for administrator approval on the UA server.
- Support for numerical values (analog and discrete in the OPC terminology) and text values
- Hierarchical view of the OPC UA nodes.
- Setting to specify which OPC UA nodes to expose in the SimApi. Can be used to only show a subset to shorten the startup time of the SimApi.
- Option to control which OPC references are followed when enumerating tags at startup.
- Use provided credentials, or anonymous access
- Current and historical continuous data for OPC UA nodes of Node Class **Variable**
- Optional use of OPC UA HistoryRead for current data, for use when the UA server does not support CurrentRead.
- Optional support for OPC UA Variables with array data for current data only (no support for historical data)
- Multiple instances of the SimApi can be run on the same SIMCA-online server
- The SimApi automatically re-establishes lost connections to the UA server if

1.2.1 Browsing the OPC UA node hierarchy



The SimApi browses the node hierarchy of the server on startup. This can take time on a big server, or slow network connection. You can make it quicker using two settings:

- by specifying **Top level Node IDs** to browse only certain branches of the server's hierarchy.
- change to browse only references of type **Organizes.** Learn more in 3.1.

1.2.1.1 On how the SimApi exposes the OPC UA node hierarchy

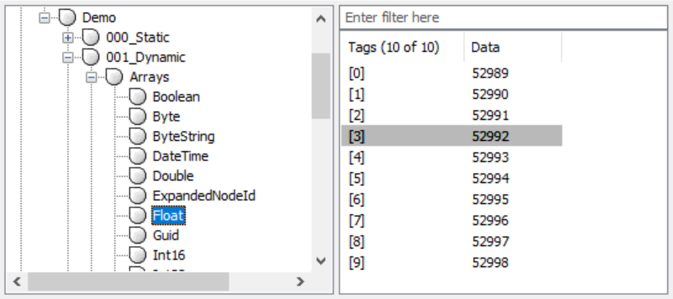
OPC UA Nodes have different node classes.

- The Object node class is used to represent systems, system components, real-world objects, and software objects. This corresponds to Nodes (containers) in the SimApi.
  - If a node is referenced in many places in the server OPC UA hierarchy, it will be exposed only in **one** place by the SimApi. This means a node can appear to be missing in the SimApi, when in fact it has been added but in a different place. Tip: if you want it to be added somewhere specific, use the setting Top Level Node IDs setting to point to that specific parent node. You can use UAExpert , https://www.unified-automation.com/products/development-tools/uaexpert.html, to get the NodeId.
- The Variable node class is used to represent the content of an Object. A variable is exposed as a tag in the SimApi.
  - OPC UA variables of the classes BaseVariableType, BaseDataVariableType, or DataItemType and its subclasses are exposed as tags, and are used to read data for in SIMCA-online or SIMCA.
  - OPC UA variables that contains/reference other variables are exposed as nodes by the SimApi. It will not be possible to read data for the variable that is exposed as a node, only for the contained other variable which are exposed as tags.

1.2.2 Array data

Array data consist of more than one value per observation. An example of array data is a measurement made at many wavelengths. To handle array data the SimApi translates each element in the array to a tag which can be used in SIMCA-online.

For example, if the array data for the Variable **Float** is [52090, 52990, 52991, ..] the SimApi would create tags [0],[1], [2], ... inside a SimApi **node** named **Float.** To use obtain the values, SIMCA-online connects the tags [0], [1], [2] to individual variables in a project.



Tags (10 of 10)	Data
[0]	52989
[1]	52990
[2]	52991
[3]	52992
[4]	52993
[5]	52994
[6]	52995
[7]	52996
[8]	52997
[9]	52998

Limitations: only current data is supported for array data. Sartorius has yet to find an OPC UA server that supports historical data for array data. Single dimension arrays are supported (no matrices).

Array data is optional in the SimApi. To use it, turn it on in the configuration utility.

When the SimApi starts, it queries the UA server for the size of the array data. The sizes of the arrays are fixed once the SimApi has started. This process adds additional time to the startup process because each variable is queried for array data. To make startup faster, use the 'Top Level NodeIDs' setting in the configuration.

1.2.3 Current and Historical data access is strongly recommended

Important: to work well in SIMCA-online, nodes/tags used must be able to provide both current data and historical data to SIMCA-online when it asks for it. In SIMCA-online you cannot use different tags for current and historical data. Read more in the SimApi Guide.

1.2.4 Optional use of OPC UA HistoryRead for current data, for use when the UA server does not support CurrentRead.

The SimApi normally uses OPC UA CurrentRead read current data. If this is not supported by the UA server, an option in the SimApi configuration can be changed to use HistoryRead instead. Then the SimApi will ask for historical data for the timestamp of the request and ask for one datapoint and use that as the current data. Note that this currently does not work for Array data because it is not supported for HistoryRead.

One such server that we tested is the Siemens SIMATIC Process Historian 2014 SP3.

1.2.5 Reading large blocks of data

A maximum array length is negotiated automatically between the OPC UA server and client when establishing the connection, e.g. 65.536. This can be viewed in the log of the server.

If ReadHistoricalData found more data points than this number, an error will occur, and no data would be returned.

To avoid this error, a maximum number of values can be set in the SimApi settings (see 3.1). For "Number of values per read operation", a number should therefore be entered below this maximum array length, e.g. 30.000. If more data is now available for the specified period, the server sends it in several portions and the error is avoided.

A value of 0 means no limitation, but this means that the automatic negotiation described above takes place resulting in a limit anyway.

1.3 OPC UA Compatibility

OPC UA is a large specification. https://en.wikipedia.org/wiki/OPC_Unified_Architecture. This SimApi supports a small subset of functionality as defined earlier.

It has been developed for and tested with:

- **Siemens WinCC** SCADA system. Versions V7.4SP1 and TIA V15SP1 version 7.x.
- **Siemens SIMATIC S7-1500** controller. Version V20.08.01.
- **Siemens SIMATIC Process Historian** (PH-OPC-UA server). Version 2014 SP3.
- **KEPServerEx** version 6 and 5 (optional component of Ambr 250)
- **MFCS 4 OPC UA Server.**
- **Unified Automation UaAnsiCServer**, version 1.9.1. https://www.unified-automation.com/
- **Prosys OPC UA Simulation server.** Version 5.0.4. https://prosysopc.com/
- **OPC Foundation demo server** opc.tcp://opcua.demo-this.com:51210/UA/SampleServer

2 Prerequisites

To use the SimApi on a computer, it must have the following software installed:

- The Microsoft Visual C++ Redistributable for Visual Studio 2015-2019. Often these are already installed on a computer (for example they are installed automatically by SIMCA or SIMCA-online), but if the SimApi fails to start because of this, download and install the latest version from https://support.microsoft.com/en-us/help/2977003/the-latest-supported-visual-c-downloads

2.1 OPC UA server security

The OPC UA SimApi uses the following concepts that control access to an OPC UA server:

- **SimApi Users** are users known on the OPC UA server. The username and password for a SimApi connection is configured in the configuration utility of the SimApi. It is passed to the OPC UA server when establishing the connection.
- **Certificates** determine which computers can connect to the OPC UA server. Certificates are always used except when the UA server is using the security policy **None**. Certificates are managed by an administrator on the UA server.

In addition, networking firewalls also must allow the SimApi to connect to remote servers.

These topics are explained in detail below.

2.2 OPC UA certificates and usage by the SimApi

When connecting to an OPC UA server, the SimApi first tries to use the **best available** security policy. Unless **None** is the only supported policy, this means that a certificate is always used automatically by the SimApi:

When connecting to the OPC UA server the first time, the OPC UA SimApi creates a certificate and sends this to the UA server for approval. An administrator of the UA server must then mark the certificate as trusted:

2.2.1 Administrator approval of SimApi server certificate is needed

How the administrator approves the certificate varies between servers: Some servers have certain directories, where trusted certificates are deposited. Consult your OPC server documentation to learn more.

For example, on WinCC, the OPC UA server automatically places it in the directory for untrusted certificates. If the certificate is then moved from there to the directory for trusted certificates, access to the OPC UA server is granted the next time the connection is established.

2.3 Firewalls between SimApi and OPC UA server

A networking firewall between the SimApi running in SIMCA, or SIMCA-online server and the OPC UA server can restrict network traffic so that the SimApi doesn't work. The TCP port used by your OPC server, as specified in the connection string, needs to be open so that traffic from the SimApi can reach the OPC server.

2.3.1 Testing network connectivity

You can use the **Test-NetConnection** command in **PowerShell** to test connectivity from the SimApi computer to the OPC UA server computer. If this test is not successful you know that there are network issues such as a firewall that blocks the traffic.

For example, to verify connectivity to the OPC server **opc.tcp://uademo.prosysopc.com:53530**, open a PowerShell console, and then type the following (notice how the name is specified without opc.tcp:// and that the port is provided as its own parameter). The last line shows that the test was successful:

```
Test-NetConnection -ComputerName uademo.prosysopc.com -Port 53530
ComputerName     : uademo.prosysopc.com
RemoteAddress    : 212.68.19.35
RemotePort       : 53530
InterfaceAlias   : Wi-Fi
SourceAddress    : 192.168.1.210
TcpTestSucceeded : True
```

The following test fails, because the port has changed to a port which is not in use on the server:

```
Test-NetConnection -ComputerName uademo.prosysopc.com -Port 12345
WARNING: TCP connect to (212.68.19.35 : 12345) failed
WARNING: Ping to 212.68.19.35 failed with status: TimedOut

ComputerName           : uademo.prosysopc.com
RemoteAddress          : 212.68.19.35
RemotePort             : 12345
InterfaceAlias         : Wi-Fi
SourceAddress          : 192.168.1.210
PingSucceeded          : False
PingReplyDetails (RTT) : 0 ms
TcpTestSucceeded       : False
```

3 Installation and setup

Refer to the **SimApi Guide** located at sartorius.com/umetrics-simapi for general step by step instructions that apply when installing a SimApi. That document also contains many troubleshooting and testing tips when installing a SimApi.

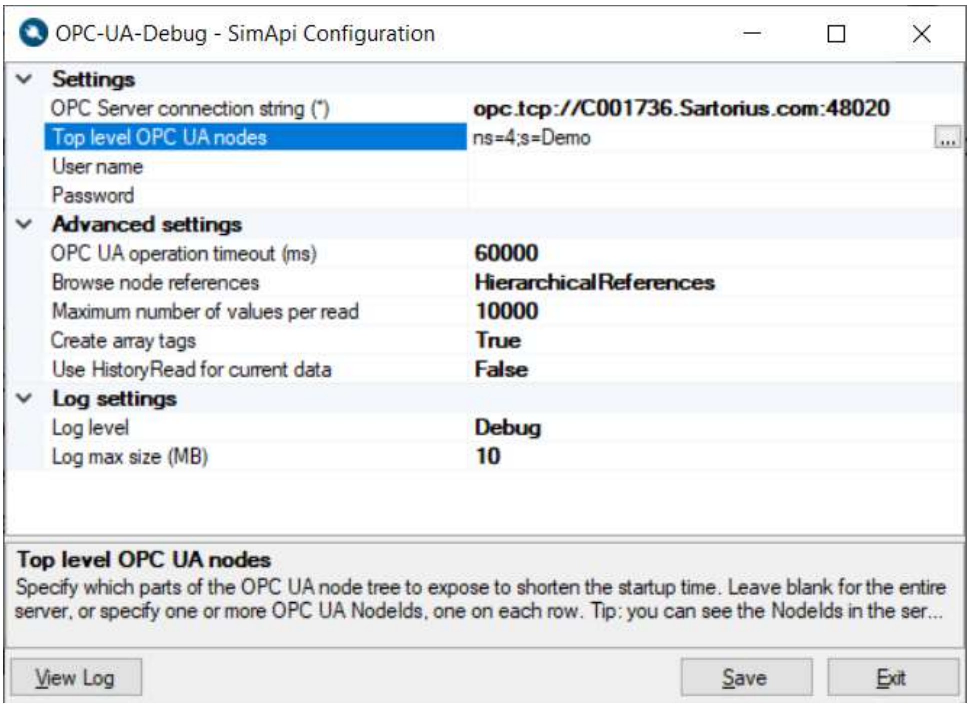
3.1 Configuring SimApi settings

To change settings for the SimApi in SIMCA-online; launch the **Server Options** utility, and on the SimApi tab, click **Configure...** for the SimApi instance you want to configure. The same guidelines apply to SIMCA, although all screenshots and examples below are for SIMCA-online.

The following dialog is displayed. Configure the settings you require for you environment.

After saving and exiting, the SIMCA-online server service needs to be restarted for the changes to be effective.

If the connection for a configuration is made for the first time, you must ensure that the certificate is classified as trusted by the OPC UA server (see 2.1 and 2.2). The log file always contains detailed information on errors.



Settings	
OPC Server connection string (*)	**opc.tcp://C001736.Sartorius.com:48020**
Top level OPC UA nodes	ns=4;s=Demo
User name	
Password	
Advanced settings	
OPC UA operation timeout (ms)	**60000**
Browse node references	**HierarchicalReferences**
Maximum number of values per read	**10000**
Create array tags	**True**
Use HistoryRead for current data	**False**
Log settings	
Log level	**Debug**
Log max size (MB)	**10**

Top level OPC UA nodes

Specify which parts of the OPC UA node tree to expose to shorten the startup time. Leave blank for the entire server, or specify one or more OPC UA NodeIds, one on each row. Tip: you can see the NodeIds in the ser...

The following settings are available.

Setting	Explanation
OPC Server connection string	The connection string to the target OPC server, ending with the port number (1234). For example, opc.tcp://server.example.com:1234
Top level OPC UA nodes	Specify which parts of the OPC UA node tree to expose to shorten the startup time. Leave blank for the entire server, or specify one or more OPC UA NodeIds, one on each row. Tip: you can see the NodeIds in the server log after starting it with this setting left empty. Click the ... button to specify NodeIds. Here is an example of a NodeId: **ns=4;s=Demo.History**

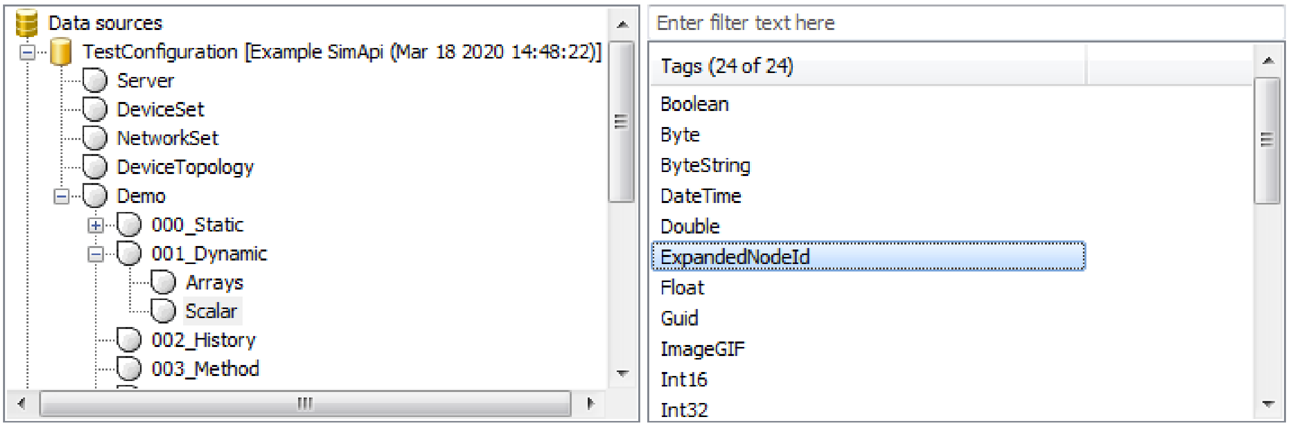
Username	The credentials used to connect to the OPC UA server provided above. Leave blank to use anonymous access.
Password	The credentials used to connect to OPC UA server provided above. Leave blank to use anonymous access.
OPC UA operation time out (ms)	Sets the timeout in milliseconds, before a OPC UA api call is canceled.
Browse node references	Controls which nodes are exposed by filtering OPC nodes on their reference type. Organizes - fewer nodes and faster startup. HierarchicalReferences - more nodes and variables-in-variables.
Maximum number of values per read operation	Sets the maximum number of values per node in each read operation. If more values exist, the read operation is automatically split into multiple reads. 0 means that there is no maximum.
Create array tags	If OPC variables should be analyzed during startup to determine if they provide array data. For such a variable one tag for each array element is created. See 1.2.2 above.
Use HistoryRead for current data	When the UA server does not support current data (CurrentRead), change this option to True to use HistoryRead to obtain the latest historical values as current values. Note: does not work for array tags.
Log Level	Controls how much information is written to the log file. (Debug, Information, Warning, Error, Critical). Debug is helpful for troubleshooting issues with the SimApi.
Log Max Size (MB)	Controls the max size of the log file before creating a new. Setting this to 0 means infinite size.

In the picture above, you can see an example of a configuration:

- We have one server, **c001736**, whose endpoint URL is entered in OPC server connection string field.
- **Top level OPC UA nodes** is set to **ns=2;s=Demo** so that only that node is exposed by the SimApi. This makes the SmiApi faster to start because the entire server need not be browsed for nodes and variables.
- **Create array tags is True**. The server will create array tags for all applicable OPC UA Variables of the server. See 1.2.2 above.
- The **Log Level is set to "Debug**", so debug entries will be written into the log file. Tip: use Debug when configuring the SimApi initially but change it to Information if the log file grows in size too quickly when you use the SimApi in production.

3.1.1 How this looks like in SIMCA-online

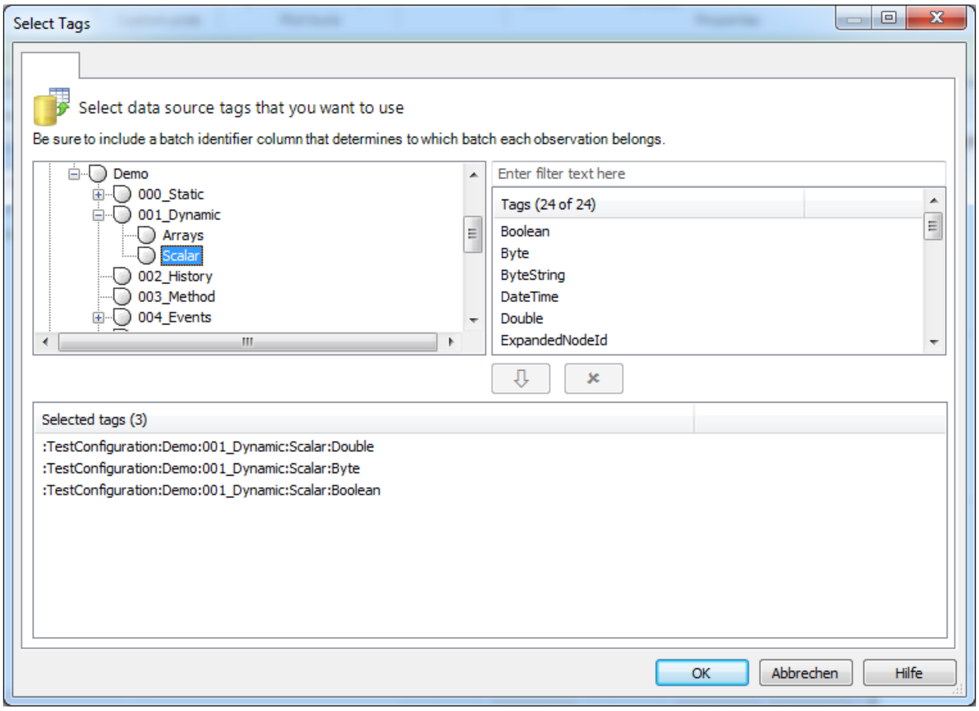
This is what the above settings result in in SIMCA-online:



Data sources
TestConfiguration [Example SimApi (Mar 18 2020 14:48:22)]
Server
DeviceSet
NetworkSet
DeviceTopology
Demo
000_Static
001_Dynamic
Arrays
Scalar
002_History
003_Method

Tags (24 of 24)
Boolean
Byte
ByteString
DateTime
Double
ExpandedNodeId
Float
Guid
ImageGIF
Int16
Int32

The OPC UA nodes of the OPC UA server are visible under the top-level node. You can select OPC UA nodes now as tags in SIMCA-online:



Select Tags

Select data source tags that you want to use

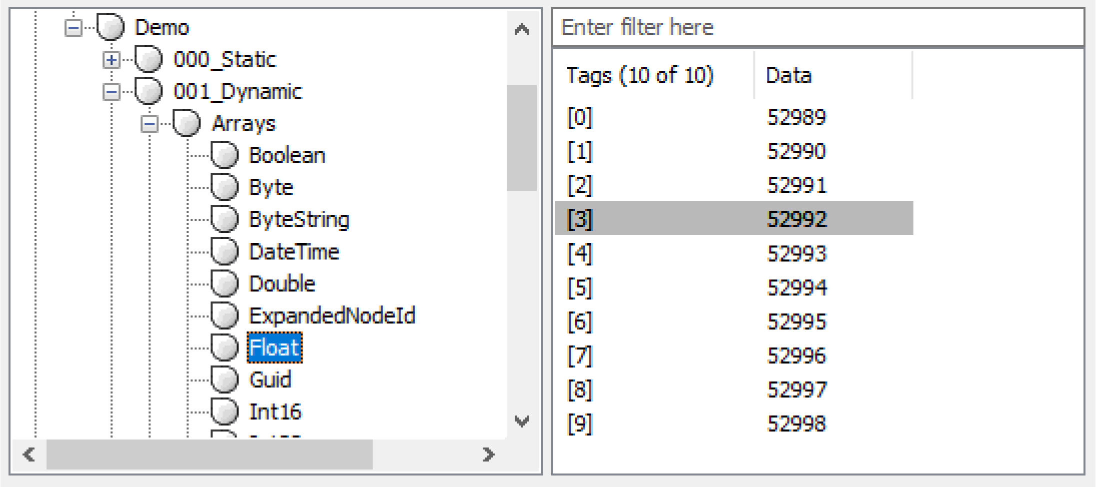
Be sure to include a batch identifier column that determines to which batch each observation belongs.

Demo
000_Static
001_Dynamic
Arrays
Scalar
002_History
003_Method
004_Events

Tags (24 of 24)
Boolean
Byte
ByteString
DateTime
Double
ExpandedNodeId

Selected tags (3)
:TestConfiguration:Demo:001_Dynamic:Scalar:Double
:TestConfiguration:Demo:001_Dynamic:Scalar:Byte
:TestConfiguration:Demo:001_Dynamic:Scalar:Boolean

The tags that are created for array data in an OPC Variable **Float** looks like this:



Demo
000_Static
001_Dynamic
Arrays
Boolean
Byte
ByteString
DateTime
Double
ExpandedNodeId
Float
Guid
Int16

Tags (10 of 10)	Data
[0]	52989
[1]	52990
[2]	52991
[3]	52992
[4]	52993
[5]	52994
[6]	52995
[7]	52996
[8]	52997
[9]	52998

4 Support

This SimApi is developed by Sartorius Data Analytics. For support, please visit sartorius.com/umetrics-support