OPC UA SimApi User Guide

2024-12-05

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. Also exposed as a node if it has other nodes below it in the hierarchy.

1.2 Features

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

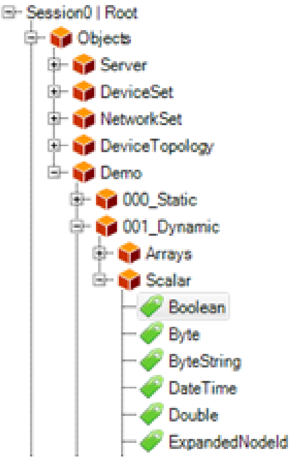
- Connect to an OPC UA server (using the OPC UA binary protocol), trying the best available security policy automatically creating a client certificate sent to the server for administrator approval on the UA server.
- Support for numerical values (analog and discrete in the OPC terminology) and text values.
- Hierarchical view of the OPC UA nodes.
- Setting to specify which OPC UA nodes to expose in the SimApi. Use this to reduce the startup time by selecting only nodes that are needed.
- 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**, using the OPC UA Data Access and Historical Access features respectively.
- As for all SimApis, data for a specific time point is the most recent data available for that time.
- Data are returned for OPC UA quality of Good and Uncertain, but Bad are returned as missing.
- Optional use of historical data for current data when the UA server does not support current data natively.
- Optional support for OPC UA Variables with array data for current data only (no support for historical data).
- Batch node functionality for Sartorius BioPAT MFCS servers.
- Write back of current values using OPC UA Write. 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.

1.2.1 Browsing the OPC UA Node Hierarchy

The SimApi browses the node hierarchy of the server on startup to create the list of nodes and tags you see in SIMCA-online. 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 Transforming the OPC UA Node Tree to the SimApi Node and Tag Tree

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.
  - OPC UA variables of types that provide numeric data are exposed as tags and are used to read data for in SIMCA-online or SIMCA. Some types of OPC UA variables are binary data or video, or audio and these are not exposed by the SimApi. If this happens it is logged in the SimApi log file.

- If a Variable node has children, a SimApi node will also be added as well as the tag. To make the name unique its name will be prefixed by $, since SIMCA-online cannot handle a node and tag of the same name inside a parent node.

1.2.2 Handling of OPC UA Quality Good, Uncertain and Bad

As for all SimApis, data for a specific time point is the most recent data available for that time.

Data with OPC UA quality Good are always used.

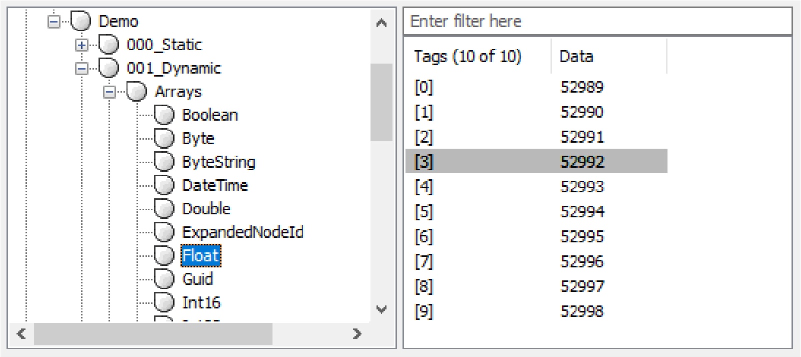
For a value with OPC UA quality **Uncertain**, the data is returned by the SimApi, but a warning is logged.

For a value with OPC UA quality **Bad**, missing is returned by the SimAPi, and a warning is logged.

1.2.3 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.

1.2.4 Current and Historical Data Access are Required

To work in desktop SIMCA historical data from a SimApi is required.

To work in SIMCA-online for batch processes, OPC UA 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. Learn more in the SimApi Guide.

1.2.5 Optional: Use Historical Data as Current Data

The SimApi normally uses OPC UA Data Access Read to read the current data. If this is not supported by the UA server, an option in the SimApi configuration can be changed to use historical data 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 does not work for Array data.

One server which only supports historical data is the Siemens SIMATIC Process Historian 2014 SP3.

1.2.6 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.

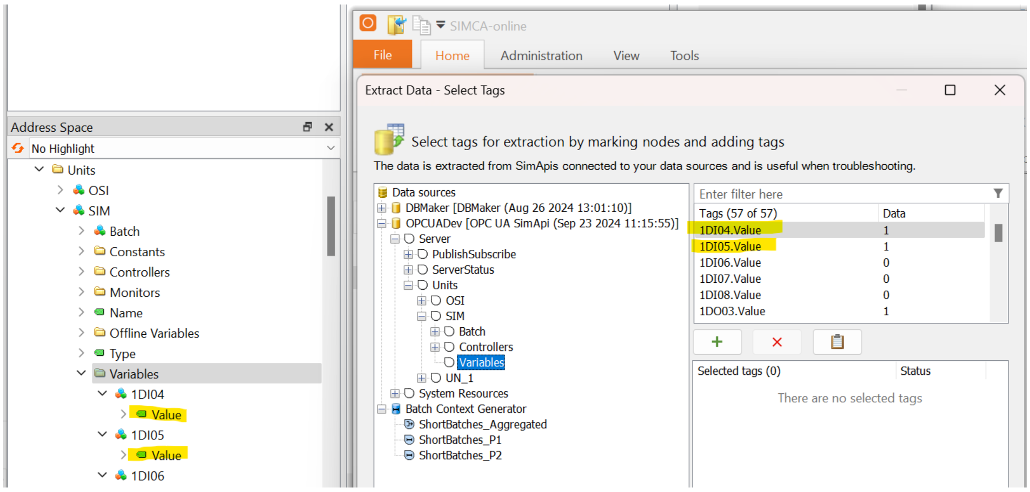
1.2.7 Sartorius BioPAT MFCS Servers

This SimApi can connect to a Sartorius BioPAT MFCS system for reading and writing data and executing batch projects in real-time in SIMCA-online.

MFCS version 4.14 or later is required for historical data and batch node functionality. The MFCS server needs a license that enabled the OPC UA server feature.

Each unit on an MFCS server is published in the OPC UA node tree under /Unit/[UnitName]. See the left part of the screenshot below.

The SimApi translates the UA nodes to SimApi nodes and tags to be used in SIMCA or SIMCA-online. Tags correspond to variables that can be used to read data from or write values to. The screenshot shows the OPC UA nodes to the left and the resulting SimApi node and tag-structure to the right:



Select tags for extraction by marking nodes and adding tags

The data is extracted from SimApis connected to your data sources and is useful when troubleshooting.

Tags (57 of 57)	Data
1DI04.Value	1
1DI05.Value	1
1DI06.Value	0
1DI07.Value	0
1DI08.Value	0
1DO03.Value	1

Selected tags (0)	Status

There are no selected tags

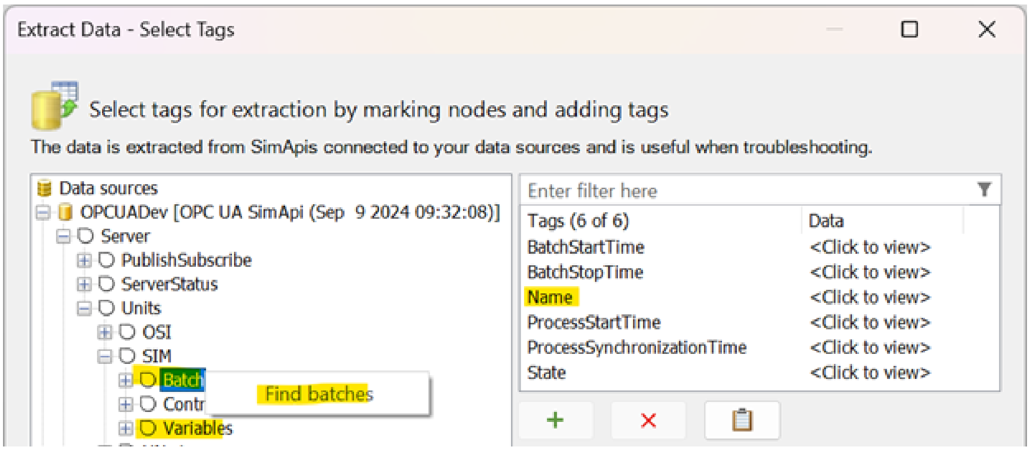
MFCS process variables are listed under the /**Variables** OPC UA node. On the MFCS server, each variable has a Value sub node, but the SimApi doesn't show these, but instead flattens the node structure so that you select the Variables node and then the variables themselves are listed as tags, but with a suffix '.Value'.

Learn more about batch node functionality and write back to the MFCS server in the following sections.

1.2.8 Per-Unit Batch Node for Sartorius BioPAT MFCS Servers

The SimApi implements **batch node** functionality for a Sartorius BioPAT MFCS system. A batch node is required in SIMCA and SIMCA-online to define when batches start and end. An MFCS server 4.14 or later is required.

Each unit in MFCS has its own batch node, defining the batches and their lifespan in that unit. The node is always called **Batch** and lies under each unit of the server. The following screenshot from SIMCA-online shows the batch node, and that the user has right-clicked to show the **Find batches** command which can be used to test a batch node.

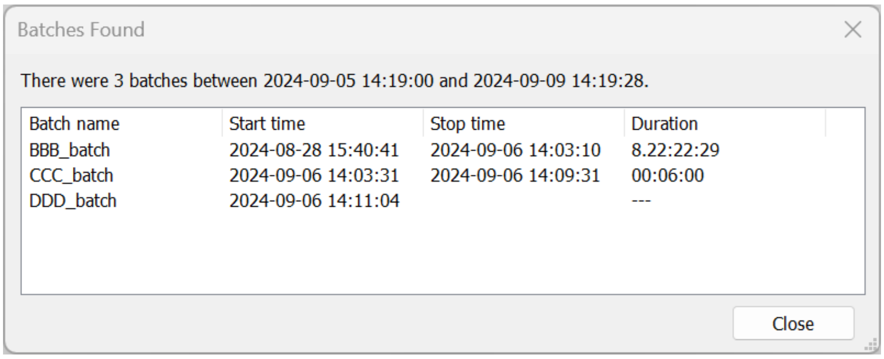


Select tags for extraction by marking nodes and adding tags

The data is extracted from SimApis connected to your data sources and is useful when troubleshooting.

Tags (6 of 6)	Data
BatchStartTime	<Click to view>
BatchStopTime	<Click to view>
Name	<Click to view>
ProcessStartTime	<Click to view>
ProcessSynchronizationTime	<Click to view>
State	<Click to view>

The results of doing find batches in a time range looks like this:

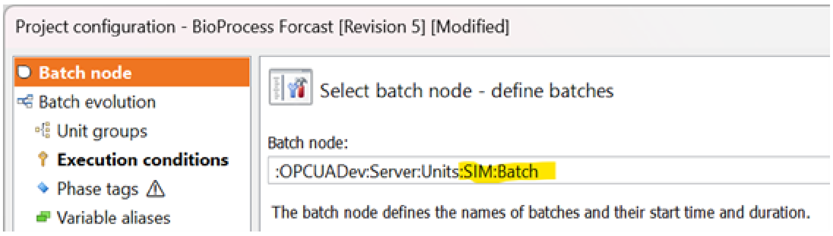


There were 3 batches between 2024-09-05 14:19:00 and 2024-09-09 14:19:28.

Batch name	Start time	Stop time	Duration
BBB_batch	2024-08-28 15:40:41	2024-09-06 14:03:10	8.22:22:29
CCC_batch	2024-09-06 14:03:31	2024-09-06 14:09:31	00:06:00
DDD_batch	2024-09-06 14:11:04		---

To use the batch node in desktop SIMCA, select the **Batch** node when importing data to find the batches.

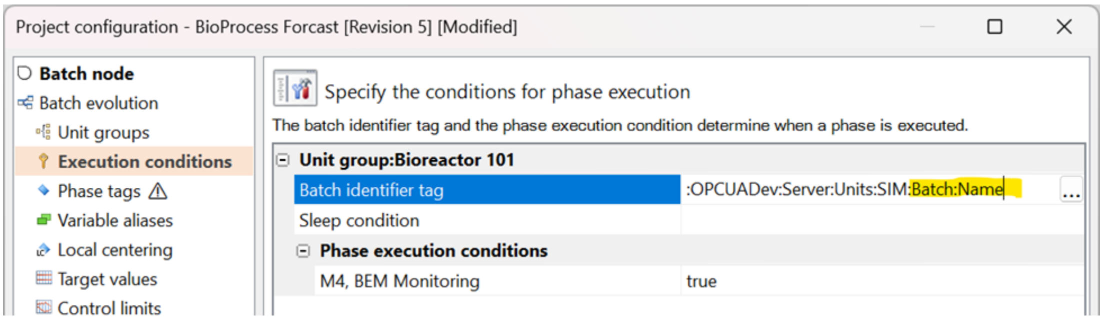
To use the batch node in SIMCA-online for a project configuration, select the **Batch** node on the Batch node project configuration page, and the node's **Name** tag on the Execution conditions page as illustrated below. This enables SIMCA-online to pick up batches and execute the models in the project with data from MFCS in real-time and by predicting past batches manually.



Select batch node - define batches

Batch node: :OPCUADev:Server:Units:SIM:Batch

The batch node defines the names of batches and their start time and duration.



Specify the conditions for phase execution

The batch identifier tag and the phase execution condition determine when a phase is executed.

Unit group:Bioreactor 101	
Batch identifier tag	:OPCUADev:Server:Units:SIM:Batch:Name
Sleep condition	
Phase execution conditions	
M4, BEM Monitoring	true

Tip: If you have several units in MFCS, you can use Batch Context Generator in SIMCA-online and its **Aggregation** feature if you want a batch in SIMCA-online to span the units in MFCS.

Troubleshooting missing Name tag: If the Name tag in the Batch node doesn't show up in the SimApi, make sure the MFCS server settings for the OPC Server component are to expose variables with 'Variable type' set to DataItem (not set to Property).

Note: The Batch node on the OPC UA server also has other tags and sub nodes, but they are not needed by SIMCA or SIMCA-online. Using the BatchStartTime and BatchStopTime tags directly is not recommended since the data in these tags are raw data that are of little use as process data. The purpose of the batch node functionality for MFCS is to translate the raw data into a format SIMCA and SIMCA-online can use.

1.2.9 Write Back of Current Values

The SimApi supports Write back of current values to OPC UA nodes using OPC UA Write.

The target use-case is writing back advised set points from SIMCA-online Control Advisor to, for example, Sartorius BioPAT MFCS server. To learn to which OPC UA nodes write-back is supported, consult the OPC UA server's documentation.

Note that the SimApi does not implement full support for Write-back which requires supporting writing a block of more than one observation of historical data. Instead the SimApi always just writes the last observation in such a block using OPC UA Write without the SourceTimeStamp set.

If the last observation in a block is older than two times the interval between observations as warning will be logged to the SimApi log.

Before using the values written back, it is advisable to use other ways to judge if the data written represents 'now' and should be used. For example, for SIMCA-online Control Advisor, the value of ValidUntilMaturity can be used. Learn more in SIMCA-online help.

The data from the SimApi (single-precision float or string missing) is cast to the DataType of the target OPC UA node. Missing data are written back as the default value of the DataType of the target node, with StatusCode GoodNoData. Scalar data types are supported, not array or matrix data.

If write fails, an error is returned and the details logged to the SimApi log file.

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)
- **Sartorius BioPAT MFCS 4.14 OPC UA Server.**
- **Unified Automation UaAnsiCServer**, version 1.9.3. https://www.unified-automation.com/
- **Prosys OPC UA Simulation server**, version 5.4.6. 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 2015-2022. 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 computer.

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

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 the SimApi Certificate

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:

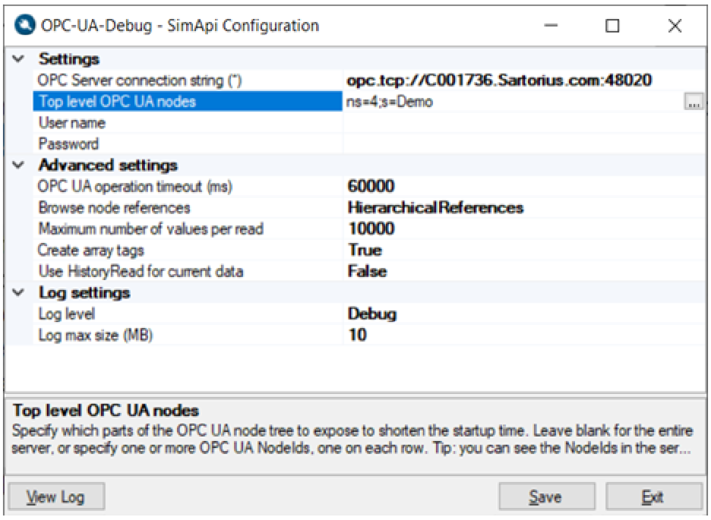
```
Test-NetConnection -ComputerName uademo.prosysopc.com -Port 12345
WARNING: TCP connect to (212.68.19.35 : 12345) failed
WARNING: Ping to 212.68.19.35 failed with status: TimedOut

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.

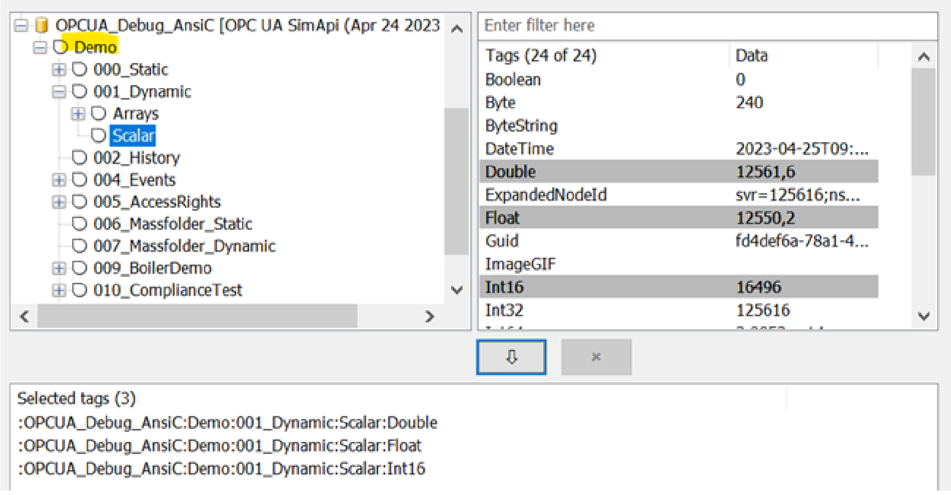
Setting	Explanation
Password	The credentials used to connect to OPC UA server provided above. Leave blank to use anonymous access.
OPC UA operation time out (ms)	Sets the timeout in milliseconds, before a OPC UA api call is canceled.
Browse node references	Controls which nodes are exposed by filtering 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 Historical Access for current data	If the UA server does not support current data (Data Access), change this option to True to use the most recent historical values as current values (Historical Access). 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. The name of this node in SIMCA-online will be the display name and not the full NodeId.
- **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 Appearance in SIMCA-online

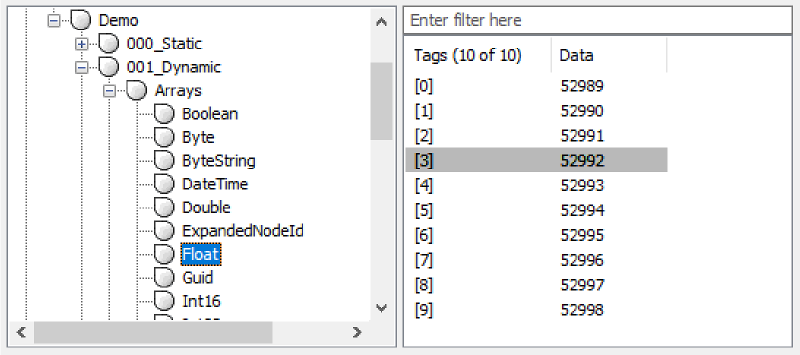
This is what the above settings result in in SIMCA-online. The OPC UA nodes of the OPC UA server are visible under the top-level node with name **Demo** (the shorter display name from OPC UA is used rather than the full NodeId you specified in the configuration).



Tags (24 of 24)	Data
Boolean	0
Byte	240
ByteString	
DateTime	2023-04-25T09:...
Double	12561,6
ExpandedNodeId	svr=125616;ns...
Float	12550,2
Guid	fd4def6a-78a1-4...
ImageGIF	
Int16	16496
Int32	125616

Selected tags (3)
:OPCUA_Debug_AnsiC:Demo:001_Dynamic:Scalar:Double
:OPCUA_Debug_AnsiC:Demo:001_Dynamic:Scalar:Float
:OPCUA_Debug_AnsiC:Demo:001_Dynamic:Scalar:Int16

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

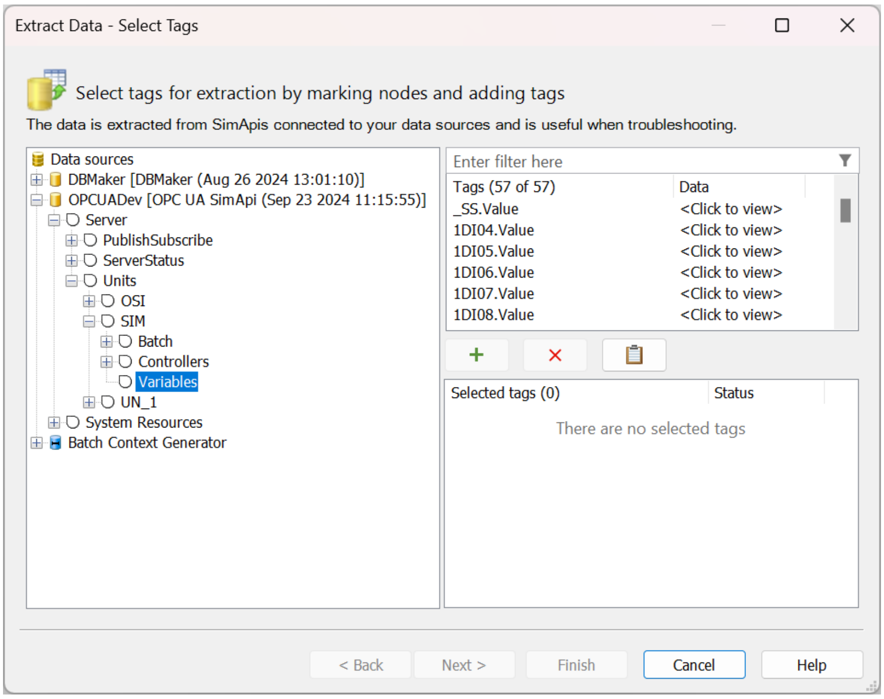


Tags (10 of 10)	Data
[0]	52989
[1]	52990
[2]	52991
[3]	52992
[4]	52993
[5]	52994
[6]	52995
[7]	52996
[8]	52997
[9]	52998

3.2 Testing the SimApi in SIMCA-online

Here is some information on how to test the SimApi:

Log in to the server in the SIMCA-online client, and navigate to Extract on the File tab. Extract helps you test the SimApi by obtaining data through it:



Extract Data - Select Tags

Select tags for extraction by marking nodes and adding tags

The data is extracted from SimApis connected to your data sources and is useful when troubleshooting.

Tags (57 of 57)	Data
_SS.Value	<Click to view>
1DI04.Value	<Click to view>
1DI05.Value	<Click to view>
1DI06.Value	<Click to view>
1DI07.Value	<Click to view>
1DI08.Value	<Click to view>

Selected tags (0)	Status

There are no selected tags

- The nodes ("folders") of the SimApi are displayed in the top-left box. Tags for the selected node are displayed top-right.
- Current data can be tested quickly simply by clicking <Click to view> on tags that provide continuous process data (see the screenshot)
- Right-click on a batch node and do Find batches for a time range. This shows batches in that node. Note that only the nodes /Unit/[UnitName]/Batch are batch nodes (learn more earlier in this document)
- Select tags in Extract and click Next and then finish the wizard to obtain data using the different modes of data retrieval that are supported: current data and historical data.

Learn more about SimApi testing in the SimApi Guide, chapter 6. sartorius.com/umetrics-simapi

3.3 Troubleshoot SimApi Problems Using the SimApi Log

If the server does not start, the SimApi doesn't work as expected or extract fails, you need to consult the SimApi log file which tells you what the problem is. Navigate to %programdata%\Umetrics\SimApi to Windows File Explorer to find the log file.

Tip: Enable debug-level logging in the SimApi configuration (if not already set) log to get full details.

4 Support

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

Sartorius Stedim Data Analytics AB
Östra Strandgatan 24
903 33 Umeå
Sweden

Phone: +46 90-18 48 00
www.sartorius.com

The information and figures contained in these instructions correspond to the version date specified below. Sartorius reserves the right to make changes to the technology, features, specifications and design of the equipment without notice. Masculine or feminine forms are used to facilitate legibility in these instructions and always simultaneously denote all genders.