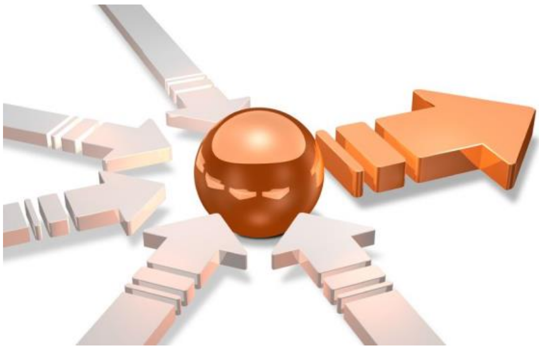
Guide

SimApi

Ensure Manufacturing Success



Guide edition date: May 18, 2021

Sartorius Stedim Data Analytics AB
Östra Strandgatan 24
SE-903 33 Umeå
Sweden
Phone: +46 (0)90 18 48 00
Email: umetrics@sartorius.com

Contents

1 Introduction to SimApis

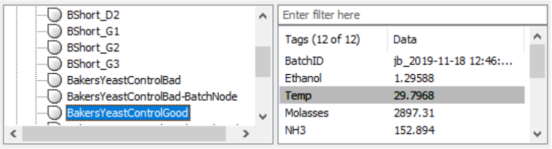
A SimApi is a software interface between the Umetrics Suite software and a data source. The primary purpose of a SimApi is to provide data to SIMCA®-online or SIMCA®.

Sartorius Stedim Data Analytics AB develops SimApis for many different data sources, such as process historians and general-purpose databases.

This document shows what a SimApi is, and how it is used in Umetrics Suite products. You'll learn how to plan for, and install a SimApi, how to troubleshoot and how to test your installation. The final chapter is about technical details of SimApis aimed at developers.

1.1 SimApi purpose: provide data to Umetrics Suite products

The primary purpose of a SimApi is to provide data to SIMCA®-online or SIMCA® from a data source. The data source is not part of SIMCA-online but can be a process historian or other system that keeps and manages the data.



BShort_D2
BShort_G1
BShort_G2
BShort_G3
BakersYeastControlBad
BakersYeastControlBad-BatchNode
BakersYeastControlGood

Enter filter here

Tags (12 of 12)	Data
BatchID	jb_2019-11-18 12:46:...
Ethanol	1.29588
Temp	29.7968
Molasses	2897.31
NH3	152.894

A SimApi exposes a hierarchy of nodes, corresponding to folders in a file system. Each node can contain other nodes, or tags. A tag corresponds to a variable. For these tags Umetrics Suite products can obtain data. The picture shows a tag, **Temp**, selected in the node **BakersYeastControlGood** in a data source in SIMCA-online.

1.2 SimApi usage in the Umetrics Suite

SIMCA can use a SimApi to retrieve data for project creation and model building as the following picture illustrates.

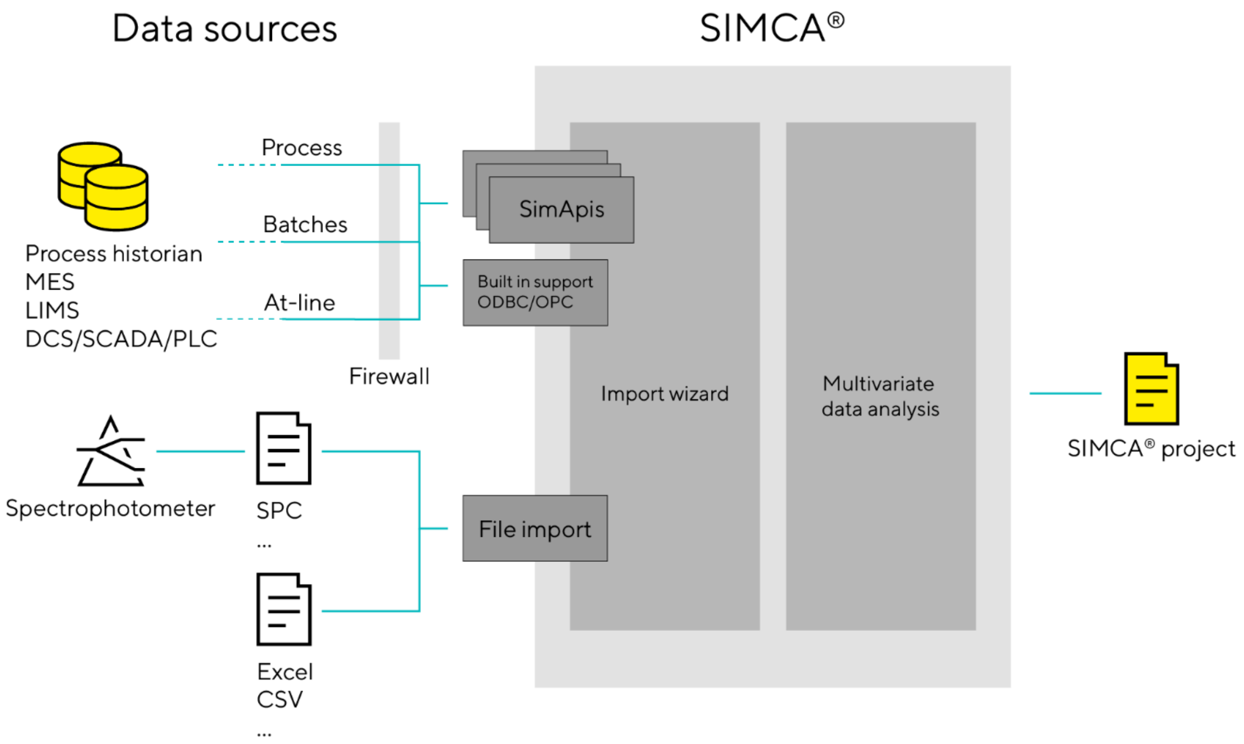


Figure 1. SIMCA used to obtain data from a data source through a SimApi.

SIMCA-online uses SimApis to obtain data in real-time for monitoring and control, as well as write back of data to the data source. The following picture shows where the SimApi is in a system consisting of a data source, SIMCA-online server, and clients.

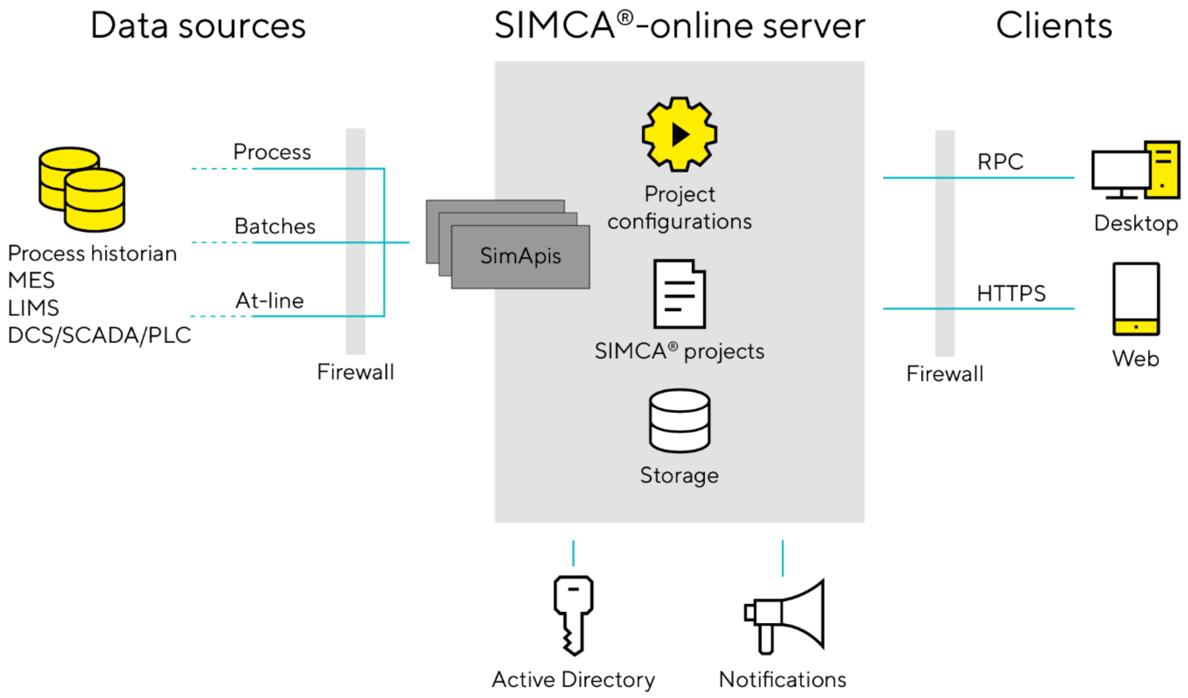


Figure 2. A SIMCA-online installation with a server with SimApis connecting to data sources, and SIMCA-online clients working with data on the SIMCA-online server.

1.3 Commonly used SimApis

The most widely used SimApis are:

- The SimApis for connecting to OSIsoft PI:
  - the **PI AF SimApi** developed by Sartorius Stedim Data Analytics
  - the old **PI Batch SimApi** developed by OSIsoft
- The **ODBC SimApi** - for general access to databases such as SQL Server or Oracle

The full list of available SimApis are listed together with their features in paragraph 3. We recommend that customers use one of these tried and tested SimApis that provides full functionality with SIMCA-online or SIMCA.

1.4 The DBMaker SimApi for simulation data

DBMaker is an application provided with the SIMCA-online server installation. It simulates a data source, such as a process historian, by using a preloaded data table with several observations which are provided one by one to SIMCA-online through the DBMaker SimApi.

DBMaker is only used for demonstration purposes and cannot be used in production with live data from a data source[1].

See the built-in help to learn more about DBMaker.

[1] You can publish your own data with DBMaker for testing SIMCA-online or demoing it with your own data. See the following knowledgebase article for more information: umetrics.com/kb/testing-simca-online-real-world-data-without-simapi

1.5 Additional documentation

The document you are reading now is one of a set of related documents, each with different focus and target audience:

Source	What	Where
SIMCA-online web page	Introductory information and downloads	sartorius.com/umetrics-simca-online
SIMCA-online ReadMe and Installation.pdf	Installation and how to get started with SIMCA-online demo data	In the installation zip file
SIMCA-online Implementation Guide	Outlines SIMCA-online functionality, puts it in context with other Umetrics Suite software, describes requirements and best practices for successful deployment, and step-by-step installation instructions.	sartorius.com/umetrics-simca-online
SimApi Guide	Preparing for and performing SimApi installations, including troubleshooting. Also contains technical details on SimApis for developers.	sartorius.com/umetrics-simapi
SimApi User Guides	One for each published SimApi. Lists the features, installation instructions, and configuration specifics of each individual SimApi.	sartorius.com/umetrics-simapi
SIMCA-online Technical Guide	Technical reference for SIMCA-online server installation planning, troubleshooting, and in-depth how SIMCA-online works.	sartorius.com/umetrics-simca-online
SIMCA-online help	How to use SIMCA-online and how SIMCA-online works.	In the software itself, and on sartorius.com/umetrics-simca-online
SIMCA-online Web Client Installation Guide	Describes the installation of the SIMCA-online Web Client.	sartorius.com/umetrics-simca-online
Umetrics knowledge base	Searchable database with articles about each released software version, technical articles, and known issues on Umetrics Suite products.	sartorius.com/umetrics-kb
SIMCA help / user guide	How to use SIMCA for creating projects.	In SIMCA and on sartorius.com/umetrics-simca

1.6 Technical support

Sartorius **online support team** answers technical questions about SimApis and can also forward requests for enhancement of SimApis to the appropriate people. Learn more at sartorius.com/umetrics-support.

2 Obtaining SimApis

We provide documentation for SimApis, and links to SimApis which are offered from partners, at sartorius.com/umetrics-simapi.

All SimApis built in-house can be acquired for free in the web shop on that page.

Consult the feature matrix to learn more about SimApi functionality and which SimApi that supports which feature.

For details about a specific SimApi, see its User Guide. The individual SimApi user guide and this document have complementing information when it comes to SimApi planning, installation, and troubleshooting.

3 SimApi features

Not all data sources are alike. A SimApi need not implement all functions in the specification. For these reasons, different SimApis offer different functionality. The following matrix lists available SimApis and their features.

	BatchTableWrapper	BUCHINIR-Online	CSV	Discoverant	IP21	IP21Batch	JUNE5	MFCS Access	MFCS4	ODBC	OPCDA	OPCHDA	OPCUA	OPC/ODBC	PI AF	SIPAT	SynTQ	Wonderware
Current data	o	o	o	o	o		o	o	o	o	o	o	o	o	o	o	o	o
Historical continuous data	o	o		o	o		o	o	o	o		o	o	o	o	o	o	o
Discrete data		o		o						o								
Batch data	o	o		o		o				o				o	o	o		
Batch context node	o	o		o		o		o	o	o				o	o	o		
Write back - continuous data	o	o			o					o	o	o		o	o			o
Write back - discrete data		o								o								
Write back - batch data	o	o								o				o	o			
Node hierarchy						o	o	o	o		o	o	o		o			
Array tag expansion					o						o	o	o					
Multiple data sources	o	o			o	o	o	o	o	o			o		o			o
Connection resiliency		o	o				o	o	o	o			o		o			o
Developed in-house	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o			o

The below table explains each feature. Notice that the table has separate columns to show which features are available in SIMCA-online and SIMCA respectively.

Feature	Purpose	SIMCA-online usage	SIMCA usage
Current data	Read the most recent value from the data source.	Real-time normal execution	
Historical data	Read historical data from the data source.	Catch-up and predict of past data, create projects using File > New	Database Import Wizard to import process data for model creation.
Discrete data	Read laboratory/IPC data from the data source. Discrete data can be used for batch configurations when data needs to be re-read during the batch evolution. Many observations per batch.	For batch projects with phases configured for discrete data retrieval.	

Feature	Purpose	SIMCA-online usage	SIMCA usage
Batch data	Read batch conditions and final quality attributes (or other MES type data). One observation per batch.	Batch conditions or local centering.	Database Import Wizard to read batch conditions for batch level model creation.
Batch node	Specify the start time and the end time (empty for an active batch) for a specific batch. Enumerate all batches that existed in a time range.	Required for execution of batch configurations.	Database Import Wizard to select batches to import.
Write back - continuous data	Write continuous data, such as predictions, back to the data source.	Write back for batch configurations for the batch evolution level or for continuous configurations	
Write back - discrete	Write discrete data, such as predictions, back to the data source.	Write back for batch configurations at the batch evolution level for phases configured for discrete data retrieval	
Write back - batch data	Write back batch level data, such as predictions or final quality attributes, to the data source.	Write back for batch configuration at the batch level	
Node hierarchy	The SimApi supports a hierarchy of nodes, similarly to a file system. Each node can contain tags and other nodes. The hierarchy makes is easier to manage a large number of nodes and tags.	Supported in all places where tags are used.	
Array tag expansion	An array tag stores multiple values. The SimApi expands the array tag to many individual tags, one for each value in the array.	Supported where tags are used for continuous data.	
Multiple data sources	The SimApi can connect to more than a single data source or supports multiple instances of itself with individual settings and log files for each instance.	Connect to several different data sources of the same kind. See umetrics.com/kb/multiple-instances-same-simapi-simca-online-server	
Connection resiliency	If the SimApi becomes disconnected from the data source, it will try to reestablish the connection.	The SIMCA-online server doesn't have to be restarted to reestablish connections to the data source.	

Feature	Purpose	SIMCA-online usage	SIMCA usage
Developed in-house	The SimApi is developed, provided and supported by Sartorius Stedim Data Analytics AB.		

3.1 Only current data, without historical data, is not recommended

Some SimApis, notably OPC DA, only supports reading current data, and not historical data.

A SimApi that only supports current data cannot be used in SIMCA, because it won't be able to read historical data on which to build the models.

For SIMCA-online, we strongly recommend a data source and SimApi that provide not only current data for real-time execution, but also *historical* data to be able to predict and catch-up past data. SIMCA-online automatically switches between real-time data and historical data as needed.

A data source that only provides current data but not historical data can work for continuous projects in SIMCA-online, but for batch projects historical data is required.

4 Preparing for a SimApi installation

This section describes important information for a successful installation of a SimApi.

4.1 32- or 64-bit SimApis

There are 32-bit and 64-bit versions of each SimApi. Which one to use depends on the architecture of the program you want to use it with. For example, a 64-bit SIMCA-online server requires a 64-bit SimApi DLL, and a 32-bit SIMCA desktop software requires a 32-bit SimApi DLL.

4.2 Location for log file and settings

A SimApi stores its log files in the hidden **Program Data** folder[2]:

%programdata%\Umetrics\SimApi, where %programdata% maps to the actual folder on your computer. It defaults to C:\ProgramData.

Each SimApi typically uses its own log file, which similarly to the SIMCA-online server log file will contain more or less data depending on a log level setting. This file is useful for troubleshooting. The log file is named <simapi>.log where <simapi> is the SimApi that you are installing, for example PIAFSimApi. Also see the next section for SIMCA-online SimApi instance names.

This folder also contains the SimApi settings in an XML file named <simapi>.xml.

Most SimApis have graphical user interfaces that change the settings in the xml file, but for some you enter the changes directly in the XML file with a text editor, such as Notepad. See the user guide for each SimApi for more information.

4.3 File names when named instances are used with SIMCA-online

With SIMCA-online 13.1 or later, each SimApi instance get its own configuration file and log file in order to support multiple instances of each SimApi. The names of these files are suffixed by the name of the instance as given on the SimApi tab in the SIMCA-online Server Options dialog. The following example shows the naming of these files, where <simapi> needs to be replaced with the SimApi name.

Configuration name given when the instance is added: **OmegaServer**

Configuration file name: <simapi>**OmegaServer**.xml

[2] This folder is hidden in Windows by default. In order to see it in Windows Explorer you configure File Explorer to show hidden files. Note that you can navigate to a hidden folder by copying and pasting the folder path to File Explorer's address bar.

Log file name: <simapi>**OmegaServer**.log

Note that the generic file <simapi>.log file is still created. This log file contains entries that for technical reasons cannot be directed to the log file of the instances.

Note that SIMCA does not support multiple instances of the SimApi, and therefore uses the names without instance name as described above.

4.4 Network planning

You should locate the SIMCA-online server close to the data source in the network. This ensures a fast connection between SIMCA-online and its data source.

Networking equipment may interfere with the connection between SIMCA-online and the data source; firewalls can for example drop connections after a period of inactivity resulting in problems in SIMCA-online where the data source becomes disconnected in a SimApi.

4.5 User accounts and data source permissions

Data sources typically control access to their data. This is usually done with usernames and passwords but IP-address- or DNS-based restrictions can also be used (for example PI Trusts in OSIsoft PI).

The username and password can be provided to the data source in different ways:

- A SimApi is run as the Windows user of the user running SIMCA or the SIMCA-online service account on the server computer. The SimApi can connect to the data source using this account. This is how the OPC HDA, and the OSIsoft PI SimAPi work, and ODBC if you don't provide credentials when configurating it.
- For generic ODBC you can use the ODBC Data Sources Administrator application found on Start in Windows.
- Some database providers provide their own drivers and tools for their databases. Oracle databases, for example, use the Oracle Data Access Components (ODAC).
- Some SimApis, such as PI AF and ODBC, have configuration dialogs that store the encrypted credentials in the SimApi XML configuration file.
- OPC DA and HDA use DCOM as the transport between data source and SimApi. DCOM is configured with the Component Services tool (DCOMCNFG.EXE) in Windows and uses Windows authentication.
- For the OSIsoft PI SimApi (not the newer AF SimApi), the OSIsoft AboutPI-SDK application (PISDKUtility.exe) is used to set up the connection to the PI server.
- PI also has various security options available in the PI System Management Tools on the PI server computer. Read more in the PI AF SimApi User Guide. This guide is helpful even if you use the older OSIsoft PI SimApi.

4.6 Verifying data source connectivity

When you want to install a SimApi on a computer it can be useful to verify the connectivity from that computer to the data source with another tool:

- **Matrikon OPC Explorer** for DA or HDA (these are separate tools) can be used to test OPC connectivity, and **Matrikon OPC Analyzer** can be used to diagnose the OPC connectivity issues. Download these free tools from https://www.matrikonopc.com/products/opc-desktop-tools/index.aspx
- **OPC Rescue** (for DA and HDA) from the OPC Training Institute's web site "enables users to easily diagnose communication and security problems, and repair them instantly with the push of a button. All this can be done without ever having to learn to configure DCOM"
- **ODBC Data Sources** in Windows is used to configure and test generic ODBC. Note that there are two versions of this tool on 64-bit Windows: one for 32-bit applications and one for 64-bit[3]. Use the Test Data Source button at the end of the ODBC configuration wizard to verify connectivity to the database. We recommend that you configure your data sources as **System** DSNs.

[3] In Windows, use Start search to find both the 32- and 64-bit ODBC Data Sources tools. In 64-bit Windows 7 the start menu only lists the 64-bit version, but you can start the 32-bit ODBC Data Sources program by launching it manually from the SysWow64-folder, typically C:\Windows\SysWOW64.

- **A database specific connection tool** from the provider of the database, such as the Oracle Data Access Components.
- **PI System Explorer can be used to test connectivity to the PI AF server. It is part of the PI AF Client which is a pre-requisite for the PI AF SimApi.**
- **AboutPI-SDK application** (PISDKUtility.exe) can be used to test connectivity and to view any error messages that might have been logged when SIMCA-online tries to connect to the PI server. This is only used for the older OSIsoft SimApi, not PIAF.
- **PI System Management Tools** are used on the PI server computer for troubleshooting from that side. For example, to look for security issues preventing access from the SIMCA-online server. Learn more on PI system troubleshooting in this YouTube video.
- **Excel** can be used to obtain data from an ODBC connection and most other systems when a suitable plugin is installed.

5 Installing a SimApi

Here is how to install a SimApi on a computer:

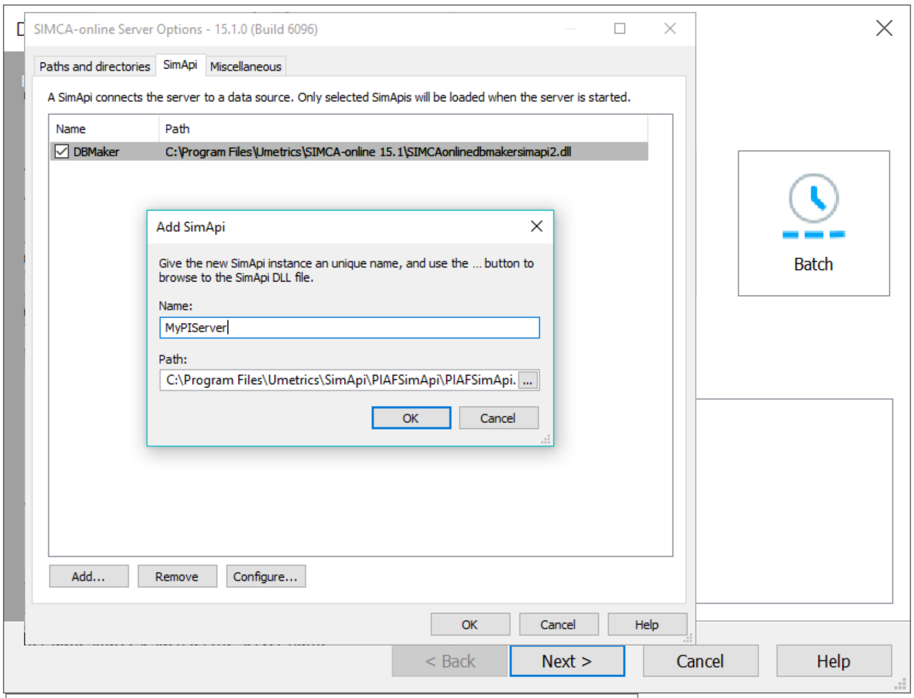
1. Read the **User Guide** for the SimApi you are installing. It contains specifics for that SimApi that complement the general instructions you are reading now.
2. Install and configure any **prerequisites** mentioned in the SimApi User Guide (for example database drivers or SDKs)
3. Run the **setup program to install the SimApi**. Install the 64-bit (x64) or the 32-bit (x86) version that matches the software you will run it in.
4. **Configure the SimApi** in SIMCA-online or SIMCA as described in the following sections and refer to the user guide of the SimApi for descriptions of available settings.
5. **Start the SIMCA-online server**. Note that this can take time, because when the SimApi is initialized, it will enumerate all tags in the data source.
6. Test the SimApi by obtaining some data. For SIMCA-online, you can use **File > Extract** as described in 6.1.
7. If the SimApi fails to work as expected, refer to the **SimApi log files** to help troubleshooting, and to the SimApi user guide.

5.1 Setting up the SimApi for use in SIMCA

Here's how to use the SimApi in SIMCA:

1. Start the database import in one of the following ways:
    a. **In SIMCA: File > New Regular Project** or **New Batch Project**. If the database import wizard is not opened automatically, open it from **File > New Spreadsheet > From Database**.
    b. **In SIMCA: Import Dataset** on the Data tab of an open SIMCA project.

2. Click **Add new data source**



SIMCA-online Server Options - 15.1.0 (Build 6096)

A SimApi connects the server to a data source. Only selected SimApis will be loaded when the server is started.

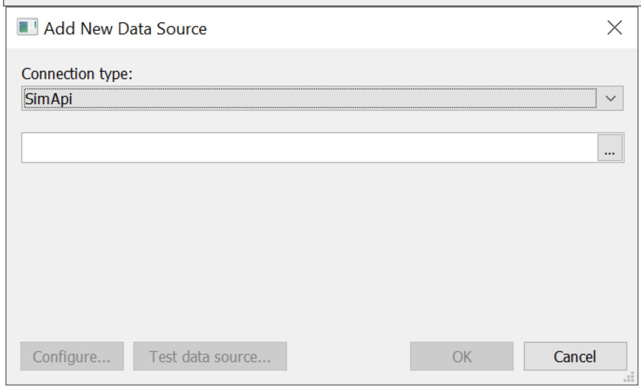
Name	Path
☑ DBMaker	C:\Program Files\Umetrics\SIMCA-online 15.1\SIMCAonlinedbmakersimapi2.dll

Add SimApi

Give the new SimApi instance an unique name, and use the ... button to browse to the SimApi DLL file.

Name: MyPIServer

Path: C:\Program Files\Umetrics\SimApi\PIAFSimApi\PIAFSimApi.



Add New Data Source

Connection type: SimApi

3. Select **SimApi** as the connection type, click the **...**-button and locate the <simapi>.dll in the installation folder, and click **Open**.
4. Click **Configure** and refer to the individual SimApi User Guide how to make the settings.
5. Click the **Test data source** connection to verify that you can connect to the database.
6. Click **OK** to complete the configuration.
7. Refer to the SIMCA help for how to import data.

5.2 Setting up the SimApi for use in SIMCA-online

Important: To be able to use a SimApi, a SIMCA-online server license is required. A demo installation of SIMCA-online does not allow SimApis to be used.

Here’s how to set up the SimApi in SIMCA-online:

1. On the server computer, start the **SIMCA-online Server Options** utility from Start in Windows. Go to the SimApi tab and click **Add...**
2. Give this instance of the SimApi a name. Use a name that distinguishes it from other data sources, such as the server name and type (for example MyPIServer).
3. Click the **...**-button to browse to and select the <simapi>.dll located in the installation folder.
4. Click **OK** to add the new SimApi and answer **Yes** to the question about launching the SimApi configuration program.
5. Make the settings for the SimApi instance. Refer to the **User Guide** of the SimApi for information about the available settings.
6. When done, close SIMCA-online Server Options and start the SIMCA-online server.

To configure multiple instances of this SimApi, repeat the above steps and use unique names for each instance. Read more about the different log and configuration files for the instances above.

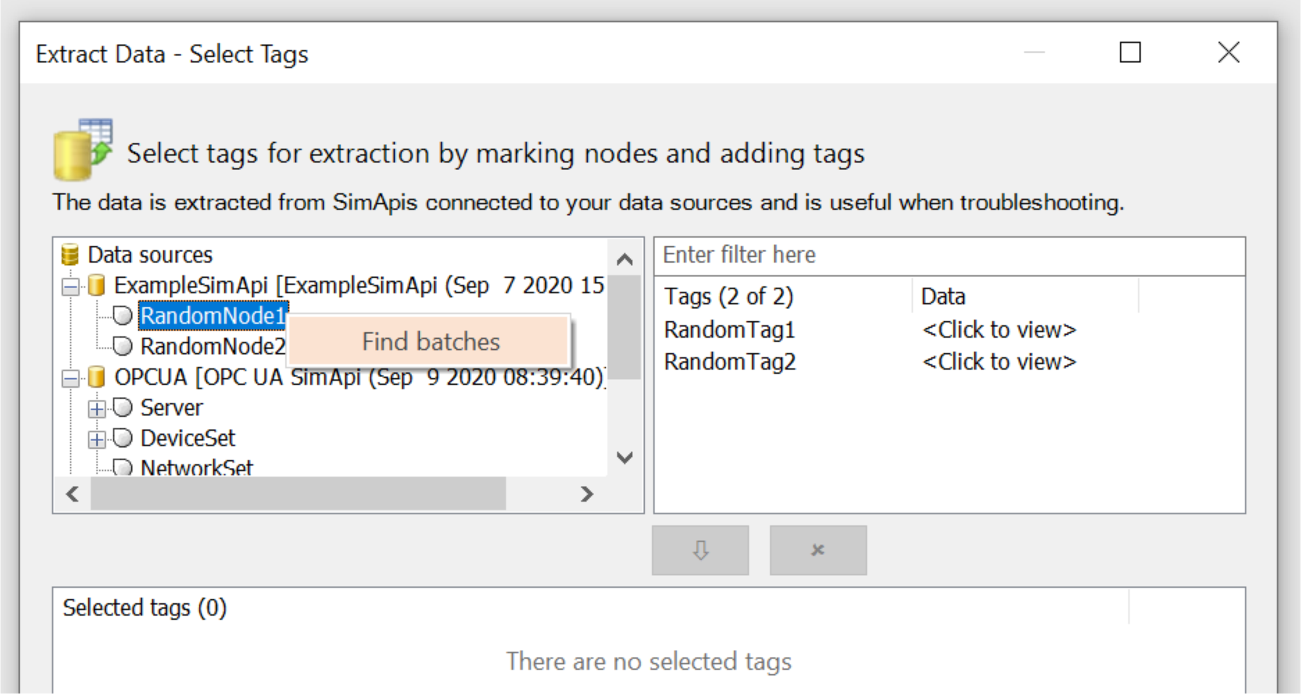
6 Testing and troubleshooting a SimApi installation

This chapter is about testing and troubleshooting a SimApi installation.

6.1 Testing a SimApi from SIMCA-online

Once the SIMCA-online server has been started successfully you can test your SimApi in SIMCA-online (if the server does not start, see 6.2):

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
Data sources
ExampleSimApi [ExampleSimApi (Sep 7 2020 15
RandomNode1
RandomNode2
Find batches
OPCUA [OPC UA SimApi (Sep 9 2020 08:39:40)
Server
DeviceSet
NetworkSet
Enter filter here
⇩
×

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 (2 of 2)	Data
RandomTag1	<Click to view>
RandomTag2	<Click to view>

Selected tags (0)

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 node to **Find batches** within a time range. The node must be a batch node that knows about batches.
- Select tags in Extract and click Next and finish the wizard to obtain data using the different modes of data retrieval: current-, **historical**-, **batch**- and **discrete data**.

Compare the extracted data with what you see in your data source tools.

6.2 Troubleshoot SimApi problems using the SimApi log file

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. Enable Debug-level logging in the SimApi log to get full details. See 4.2.

Note: the SIMCA-online server logs[4] are not so useful here. They will show how the SimApi was loaded and initialized by the server, but the SimApi specific details are in *its* log file.

[4] Learn more on the server log at https://umetrics.com/kb/making-use-simca-online-server-logging-finding-problems

6.3 Use the right SIMCA-online service account

When you are testing access to the external data source, remember that you are logged in as a specific user on the server computer (typically your own user account in a Windows domain), but that the SIMCA-online server **service account** is a different account, by default LocalSystem, which has **different** access rights compared to your user account.

For this reason, it is not uncommon that tests work when run as your account, but that SIMCA-online fails to connect to the data source.

To solve this issue, access must be granted for the account that is used by the SIMCA-online server service. Typically, you change LocalSystem to a specific domain service account, and grant rights to this account.

7 Technical details on SimApis

This chapter gives technical details on how a SimApi works. It is mainly aimed at developers that want to understand SimApis for the purpose of implementing a SimApi for a data source.

Developers should also read the earlier parts of this document for an introduction to SimApis and to the high-level descriptions of features.

7.1 When to consider developing a SimApi and when not to?

Before considering developing a SimApi for a data source:

1. Investigate if there already is a SimApi that you can use. Perhaps you can enable some feature in your data source to use one of the existing SimApis, such as OPC UA.
2. Carefully go through this document and its references and investigate if your data source fulfills necessary requirements: for example, it needs to be fast enough, provide not just current data, but also historical data (in almost all cases).

For these reasons, we don’t recommend developing a SimApi that connects to low level hardware or instruments. It is better to connect those instruments to a process historian such as OSIsoft PI, and let it obtain data from the instrument, and historize it. Then the PIAF SimApi can be used to obtain data from PI to the Umetrics product.

7.2 SimApi development and the SimApi specification

The SimApi specification, **SimApi-v2**, contains documentation for all C-functions in the SimApi that a SimApi DLL needs to implement as well as some guidance for how to develop a SimApi. This implementation is a C/C++ implementation at a very low level, making directly using it as the basis of an implementation unnecessarily hard.

The recommended, and easier, way to implement a SimApi is to base it on the **ExampleSimApi** source code that we provide. It is an example SimApi implementation written in .NET that handles the transition from a C interface to .NET. It also has framework code for logging, settings, configuration GUI and other framework code, making a SimApi implementation in .NET relatively straight forward for an experienced developer.

A SimApi can be developed by people with knowledge in .NET, C, or C++, and good knowledge of the data source that the SimApi should connect to.u

Contact your sales representative to obtain the SimApi Software Development kit.

7.3 Reading or writing data

A SimApi has the main task of providing data from a data source. This is referred to as **reading** data.

Most SimApi implementations also support **writing** data. This means writing back data through the SimApi to the data source. Writing data is an optional feature in SIMCA-online.

7.4 Tags and Nodes

A **tag** is an identifier of a column or “variable” in a data source. A tag’s name is used to identity the tag. Names within node must be unique.

A **node** is a container of tags. A node can also contain other nodes, similarly to how a file system has folders in folders.

Like in a file system, the node and tag names can be combined to a full path that uniquely identifies a tag. The tag paths are used in SIMCA-online or SIMCA when selecting tags to use. A tag path starts with a SimApi instance name followed by the node-structure, and ending with the tag name, each item separated with a colon (:). For example ":ODBCSQLServer:Node:SensorTag1".

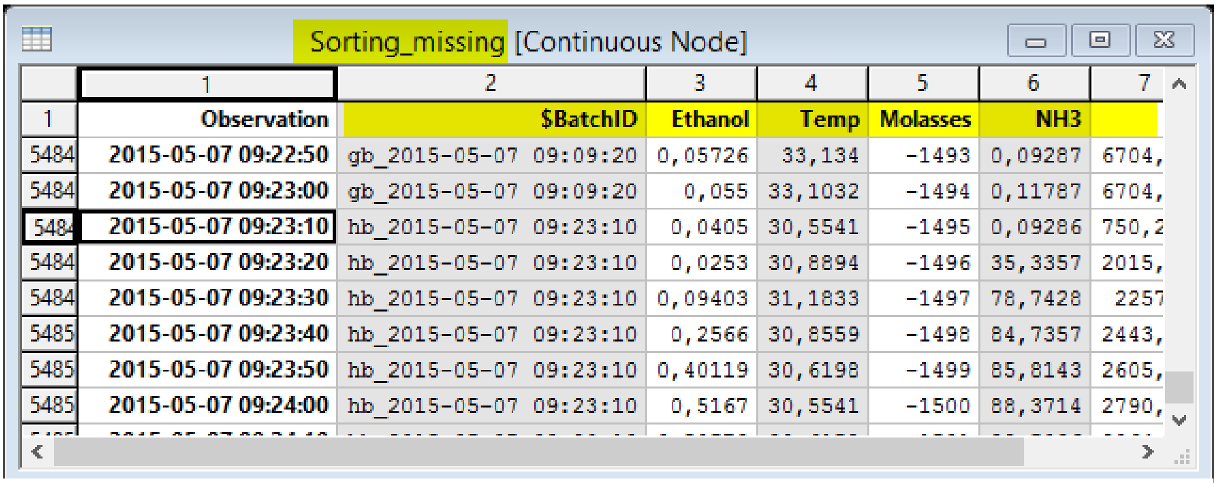
7.4.1 Case sensitivity of tag- and node names

Tag names and node names are **case sensitive**.

Thus, a tag called "tag1" is **not** the same as "Tag1" because of the different case of the "T". We recommend against using tags or node names that differ only in case.

7.4.2 Continuous process node

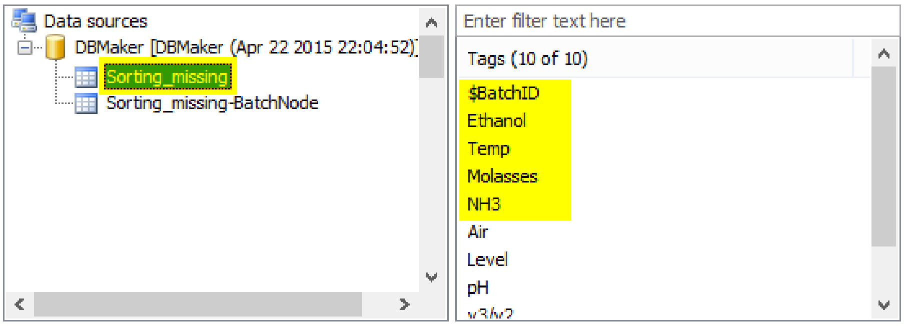
If a node contains tags with continuous process data, it can be referred to as a **process node**. The following two screenshots show a tabular representation of a process node with data followed by a picture showing how the node looks when selecting tags in SIMCA-online.



Sorting_missing [Continuous Node]

	1	2	3	4	5	6	7
1	Observation	$BatchID	Ethanol	Temp	Molasses	NH3	
5484	2015-05-07 09:22:50	gb_2015-05-07 09:09:20	0,05726	33,134	-1493	0,09287	6704,
5484	2015-05-07 09:23:00	gb_2015-05-07 09:09:20	0,055	33,1032	-1494	0,11787	6704,
548	2015-05-07 09:23:10	hb_2015-05-07 09:23:10	0,0405	30,5541	-1495	0,09286	750,2
5484	2015-05-07 09:23:20	hb_2015-05-07 09:23:10	0,0253	30,8894	-1496	35,3357	2015,
5484	2015-05-07 09:23:30	hb_2015-05-07 09:23:10	0,09403	31,1833	-1497	78,7428	2257
5485	2015-05-07 09:23:40	hb_2015-05-07 09:23:10	0,2566	30,8559	-1498	84,7357	2443,
5485	2015-05-07 09:23:50	hb_2015-05-07 09:23:10	0,40119	30,6198	-1499	85,8143	2605,
5485	2015-05-07 09:24:00	hb_2015-05-07 09:23:10	0,5167	30,5541	-1500	88,3714	2790,

Figure 3. An example of a node with data. The node name is **Sorting_missing**. The first column contains the time stamps of the observations. The remaining column headers are tags. Each column contains measurements of the tags. Each row constitutes one observation.



Data sources
- DBMaker [DBMaker (Apr 22 2015 22:04:52)]
  - Sorting_missing
  - Sorting_missing-BatchNode

Enter filter text here

Tags (10 of 10)

$BatchID
Ethanol
Temp
Molasses
NH3
Air
Level
pH

Figure 4: This is how the same node looks from SIMCA-online. You see the node name to the left, and the tags in that node listed to the right. The full tag path to the Ethanol tag in this screenshot is **:DBMaker:Sorting_missing:Ethanol**.

7.4.3 Continuous process nodes must be independent of batches, runs or time

To work well in a SimApi a node must be **independent** of batches, runs, or time. Having a node that contains data for a specific batch or time range would not work well in SIMCA-online because the project configuration then could only read data for that batch and not be used for other batches.

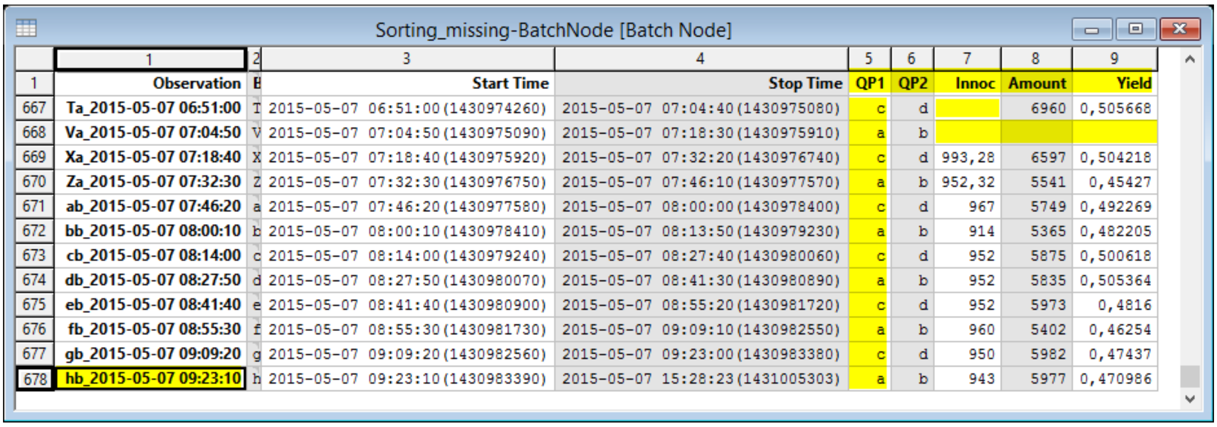
Instead, a node should be mapped to one or more physical units in the process where measurements are performed.

7.4.4 Batch node

A **batch node is a node that keeps track of batches**; their batch identifiers, start times, and end times. It is a requirement for batch project execution in SIMCA-online. If you don't have a batch node in your data source, you can use the Batch Context Generator in SIMCA-online. See the built-in help.

A batch node can also contain **batch data**; data for which there is only one observation for the whole batch. Note that tags with batch data need not be in a node that has the full functionality of a batch node. It is enough that the SimApi supports reading batch data for the tags. Learn more on batch data in 7.6.

Here is an example of a batch node:



Sorting_missing-BatchNode [Batch Node]

	1	2	3	4	5	6	7	8	9
1	Observation	B	Start Time	Stop Time	QP1	QP2	Innoc	Amount	Yield
667	Ta_2015-05-07 06:51:00	T	2015-05-07 06:51:00(1430974260)	2015-05-07 07:04:40(1430975080)	c	d		6960	0,505668
668	Va_2015-05-07 07:04:50	V	2015-05-07 07:04:50(1430975090)	2015-05-07 07:18:30(1430975910)	a	b			
669	Xa_2015-05-07 07:18:40	X	2015-05-07 07:18:40(1430975920)	2015-05-07 07:32:20(1430976740)	c	d	993,28	6597	0,504218
670	Za_2015-05-07 07:32:30	Z	2015-05-07 07:32:30(1430976750)	2015-05-07 07:46:10(1430977570)	a	b	952,32	5541	0,45427
671	ab_2015-05-07 07:46:20	a	2015-05-07 07:46:20(1430977580)	2015-05-07 08:00:00(1430978400)	c	d	967	5749	0,492269
672	bb_2015-05-07 08:00:10	b	2015-05-07 08:00:10(1430978410)	2015-05-07 08:13:50(1430979230)	a	b	914	5365	0,482205
673	cb_2015-05-07 08:14:00	c	2015-05-07 08:14:00(1430979240)	2015-05-07 08:27:40(1430980060)	c	d	952	5875	0,500618
674	db_2015-05-07 08:27:50	d	2015-05-07 08:27:50(1430980070)	2015-05-07 08:41:30(1430980890)	a	b	952	5835	0,505364
675	eb_2015-05-07 08:41:40	e	2015-05-07 08:41:40(1430980900)	2015-05-07 08:55:20(1430981720)	c	d	952	5973	0,4816
676	fb_2015-05-07 08:55:30	f	2015-05-07 08:55:30(1430981730)	2015-05-07 09:09:10(1430982550)	a	b	960	5402	0,46254
677	gb_2015-05-07 09:09:20	g	2015-05-07 09:09:20(1430982560)	2015-05-07 09:23:00(1430983380)	c	d	950	5982	0,47437
678	hb_2015-05-07 09:23:10	h	2015-05-07 09:23:10(1430983390)	2015-05-07 15:28:23(1431005303)	a	b	943	5977	0,470986

Figure 5. A sample batch node named **Sorting_missing-BatchNode**. The first column contains batch identifiers, followed by the start time and end time of the batch. The last five columns are tags with batch data (batch conditions). This example also shows the different data types which are defined in 7.5: QP1 and QP2 are qualitative (text) tags the remaining tags contain numerical data. Also notice the missing data highlighted in yellow in some cells.

Note: The above screenshot is taken from DBMaker, bundled with SIMCA-online. To see this yourself in DBMaker, click the View Data button on the Bakers Yeast database to display two windows, one of which is the batch node, and the other the process data.

7.5 Data types: numerical data, text data and missing data

For each tag, a SimApi can support three types of data: numerical, text and missing:

- **Numerical** data are typically real values of process parameters, for example 6.5. The SimApi interface as currently defined can only handle 32-bit floating point values. All other numerical data types in a data source should be converted to float.
- **Text/string** data are used for batch IDs, phase execution conditions or for qualitative variables. The values for text tag data are case sensitive. This means that the value "running" is not the same as "RUNNING". Datetime variables are not supported directly by the SimApi, but they can be returned as a string formatted as YY-MM-DD HH:MM (for example "2020-09-07 13:45").
- **Missing values** means there is no value to return, i.e. no data.

What type is **returned** by a SimApi is up to the SimApi implementation. A SimApi knows about the data in the data source and should return the data type that fits best.

7.6 Three modes of data retrieval: Continuous, Batch and Discrete

The SimApi specification defines three modes of retrieval for data, i.e. three different ways the SimApi can provide data from tags in a data source (or in the other direction: write data to tags in a data source).

Continuous data retrieval – this refers to data read continuously, and sequentially, observation per observation as the batch or process evolves. Data is read for a specific time, or range, at a regular interval between observations.

Batch data retrieval – this refers to a single observation with data for an entire batch (**not** associated with a specific maturity or time point). Batch attributes and local centering data are read as batch data in SIMCA-online. Batch conditions are normally read as batch data too (unless they are configured for discrete data retrieval).

Discrete data retrieval – discrete data can consist of several observations for many maturities. But unlike continuous data, discrete data is not read sequentially but rather all data at once for a specific phase of a batch. Data need not be spaced with regular intervals of the maturity variable. *All* data is re-read each time the data is requested, at the configured interval.

For any given tag data can be requested in any of the three modes, but typically a SimApi will only support one of these modes for an individual tag. Likewise, it is theoretically allowed to mix tags within a node, but typically all tags within a specific node support the same mode of data retrieval.

For continuous data (but not for batch- or discrete data[5]), requests can be made for current data or historical data which is the topic of the next section.

Not all SimApis support all modes. See the feature matrix above and the SimApi web page for details.

7.7 Current and Historical continuous data through a SimApi

7.7.1 Current data

Reading **current data** means asking the data source for values of one or more tags at the time of asking. Notice that the time of the external data source is **not** used here. The SimApi reports the latest values from the data source to the SIMCA-online server.

The data read as current data is what SIMCA-online will show as live data. For this reason, it is important that there are **no unnecessary delays in the data source**. Current data should be as recent as possible to work well in SIMCA-online.

7.7.2 Historical data

Reading **historical data** means asking the data source for values of one or more tags for a specific time range with a specific interval between observations. Notice that here it **is** the external data source's local time that is used to find the data. Therefore, time synchronization between data source and servers are important.

Historical data consists of a matrix of data. It is up to the SimApi implementation to request the data from the data source, and sample it at the specified interval and construct the matrix of data to return:

- Sometimes the data source itself has aggregation functions to return processed data, or sampling functions, that can be used to return the right data.
- For other data sources the SimApi has to request all data in the time range (typically also requesting the observation **before** the start of the time range to be able to know the initial value), and then manually sample the right observations.

Note: SIMCA-online typically does not request more than hundreds of observations in one call during normal project execution. When doing extract in SIMCA-online, or when running SIMCA, larger requests of data can be made. These can take a long time which is perfectly normal.

7.7.3 Current data and historical data must match

Sometimes there can be **differences when data is read as real-time current data or historical data**. This causes problems in SIMCA-online because the server automatically switches between current and historical data as needed.

Read more relating to time differences across the system, and interpolation and compression in the data source in the SIMCA-online Technical Guide.

7.7.4 More real-time requirements

Read more on requirements for real-time data sources in the SIMCA-online Implementation Guide in the section about Data Requirements.

7.8 Data can be read for any time T

When SIMCA-online asks for a value of a tag for time T it will receive the value from the data source from time T, **or** the latest observation in the data source before time T, **or** an interpolated value for time T. Thus, the server will always

[5] As you saw above, all discrete data is read each time data is requested, and thus it doesn't make sense talking about current or historical data here.

get a value at each time it asks for, even though and observation for this exact time point might not exist in the data source.

Timestamps in the SimApi are always UTC.

SIMCA-online clients and SIMCA present the time as local time.

7.9 Threading

The SimApi is called by a single thread by the user of the SimApi. Thus, the SimApi need not worry about multi-threading.

7.10 Error handling

When a SimApi cannot fulfill a request from the data source it can handle this problem in one of two ways; by returning missing values (no data) or by signaling a SimApi error:

1. Returning missing values to the caller and signaling success allows the caller to continue as normal (but of course without any data). This is a recommended practice for partial errors such as when data for could be obtained for some, but not all, tags in a request.
2. Signaling a SimApi error allows the caller (for example the SIMCA-online server) to see this immediately and to act. This is a recommended practice for requests that fails completely and cannot return any data at all.

SIMCA-online handles missing values or error codes differently, as is described in the SIMCA-online Technical Guide.