sartorius stedim biotech

User Guide

BioPAT® Spectro

SimApi

www.umetrics.com

Sartorius Stedim Data Analytics AB, patents and trademarks

US-7151976, US-7809450, US-8494798, US-8271103, US-8244498, US-9069345, US-8086327, US-8412356, US-8577480, US-8725469, US-9429939, US-9541471, US-7523384, US-7465417.

COMPUTER-IMPLEMENTED SYSTEMS AND METHODS FOR GENERATING GENERALIZED FRACTIONAL DESIGNS, Pending

Trademark	Descriptor
Umetrics™	Suite of Data Analytics Solutions
MODDE®	Design of Experiments Solution
SIMCA®	Multivariate Data Analysis Solution
OPLS®	Method for improved regression analysis
O2PLS®	Method for data integration
O2PLS-DA®	Method for exhaustive discriminant analysis
OPLS-DA®	Method for group separation
PLS-TREE®	Top down clustering
S-PLOT®	Highlighting discriminatory variables
EZinfo®	Embedded Waters solution
VALUE FROM DATA®	We are value providers

ID #2079

Guide edition date: October 4, 2017

sartorius stedim biotech

Sartorius Stedim Data Analytics AB

Tvistevägen 48

SE-907 36 Umeå

Sweden

Phone: +46 (0)90 18 48 00

Contents

1 Introduction

This document is the user guide for the **BioPAT® Spectro SimApi** from Sartorius Stedim Data Analytics.

It connects to an ODBC (Open Database Connectivity) data source. For the SimApi to be able to interpret the data the ODBC data source needs to be structured in a particular way as described in this document.

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

This SimApi can be used by Easy Analytics, SIMCA, or SIMCA-online.

For more information on available SimApi's, see umetrics.com/simapi.

1.1 Features

- Reading process data (current and historical) and batch data.
- Support for Write back from SIMCA-online for batch evolution and batch data to the ODBC data source.
- Support for multiple batch nodes.
- Support for a discrete node and reading discrete data from this node.

1.1.1 Synthetic process batch id

In a batch project, the process data has to have a batch id tag (column) that is specified on the **Phase conditions** section of the project configuration in SIMCA-online. This tag is matched against the batch node to know if a phase should execute.

If the process data doesn't have a batch id tag, the ODBC SimApi feature **synthetic process batch id** can be used. It generates the process data batch id using data from the batch node.

To use this feature, go to the **Phase conditions** section of the project configuration in the SIMCA-online client and configure the **Batch identifier tag** to be the batch id of **the batch node**. Whenever the server reads the batch id for the process data, the synthetic batch id from the batch node will be returned ensuring that the unit will execute.

Note that this feature does **not** work with concurrent (parallel) batches. Thus for any given time there must be only one batch active in the batch node.

Note that performance suffer a bit when synthetic batch id tags are used, especially during catch-up and repredict.

1.1.2 Discrete data

Discrete data is infrequently measured data which have no logical values in between measurements. Usually a sample is taken on each batch at semi-regular intervals (such as once every day). This sample is then sent to a lab which performs analysis on the sample and at a later stage returned with a report on the sample for the required variables. This is then entered in the database in the discrete data table.

1.1.3 Synthetic batch age tags for discrete data nodes

For discrete data nodes there are four synthetic tags named $BatchAge(d), $BatchAge(h), $BatchAge(m), $BatchAge(s). When reading their values they will return the batch age as a floating point number for each sample in four different magnitudes: days, hours, minutes and seconds respectively. These tags can be used as maturity in the SIMCA model, reducing the need to explicitly adding and populating such tags to the discrete data tables.

2 Prerequisites

For this SimApi to work there are requirements that must be fulfilled before the installation.

2.1 ODBC Drivers

In order for the BioPAT® Spectro SimApi to work with your database you must install ODBC drivers for your database. You obtain drivers from the manufacturer of the database. Drivers for Microsoft SQL Server are often installed on most Windows computers.

2.1.1 Configure ODBC Data Sources

The data source should be configured as a System DSN in the ODBC Data Sources control panel in Windows.

Note that there are two versions of this tool on 64-bit Windows: one for 32-bit applications and one for 64-bit[1].

In the ODBC configuration wizard you can configure a user name to use to connect to the database if needed. (Otherwise the account used to run the SimApi will be used to connect to the data base – this typically mean LocalSystem for the SIMCA-online service or a user account for SIMCA if the SimApi is used from SIMCA).

Use the Test Data Source button at the end of the ODBC configuration wizard to verify connectivity to the database.

2.2 Data Source Prerequisites

A database can of course contain almost any data with an arbitrary structure. The SimApi however assumes it can read current or historical data for a limited number of tags[2].

Thus, for the ODBC SimApi to work with a database, a certain predefined structure needs to be applied to the database. In many cases this can be done by creating a set of database views that re-arrange existing data into the required structure.

The BioPAT® Spectro SimApi requires that the database has a certain structure in order for the SimApi to be able to use it. The following views or tables are required:

- One or more views containing data.
- One or more **Batch Nodes**. Contains information about batches such as start time, stop time and optional batch conditions. This is view is optional if you only have process (non-batch) data.

Notes

- When **view** is mentioned above it refers to a database view that aggregates data from existing tables. This is often the simplest solution; creating views that present existing data in a format that the SimApi requires. Database **tables** can be used too instead of views.
- It is important that all views or tables used has a unique primary key and indexing is used on date/time columns so that the performance of the queries by the SimApi won't suffer.

The sub sections describe the database views in detail.

2.2.1 Views (or Tables)

The views (or tables) for the data contains both historical and current data.

Every column in the view corresponds to a tag that will be transposed through the SimApi.

Each row in the views represent an observation with values for each tag.

The views should have the following columns

- DateTime – The historical time for the tag values (PK, datetime, not null).
- [Tag Name] – There should be one column for each tag. Contains the data for the tag (float for numerical values or varchar for string data).

Note that for each column, the above description also states which rows should be primary key and the data type for each column.

---

[1] On 64-bit Windows you can start the 32-bit ODBC Data Sources program by launching it manually from the SysWow64-folder, typically C:\Windows\SysWOW64. If you just start ODBC Data Sources from the start menu in 64-bit Windows 7 it will launch the 64-bit version.

[2] For more information about SimApis, see the SIMCA-online Technical Guide at http://www.umetrics.com/kb/simca-online-technical-guide.

The **names** of the columns are arbitrary in the database since the names are specified in the configuration file.

The following columns are not mandatory, but are useful to add if batches are modelled with multiple phases and there are several units in the process

- UnitBatchID – One tag per unit that contains the batch ID within a certain unit (varchar). This tag can be used in the **Batch identifier tag** field for that particular unit in the Phase conditions section of the configuration of this project in SIMCA-online.
- PhaseID – One tag per unit that holds the phase info for the unit (int, float or varchar). This tag can be used in logical expression in the **Phase condition** field in the Phase conditions section of the configuration of this project in SIMCA-online.

The maximum allowed number of tags (columns) fetched is 255.

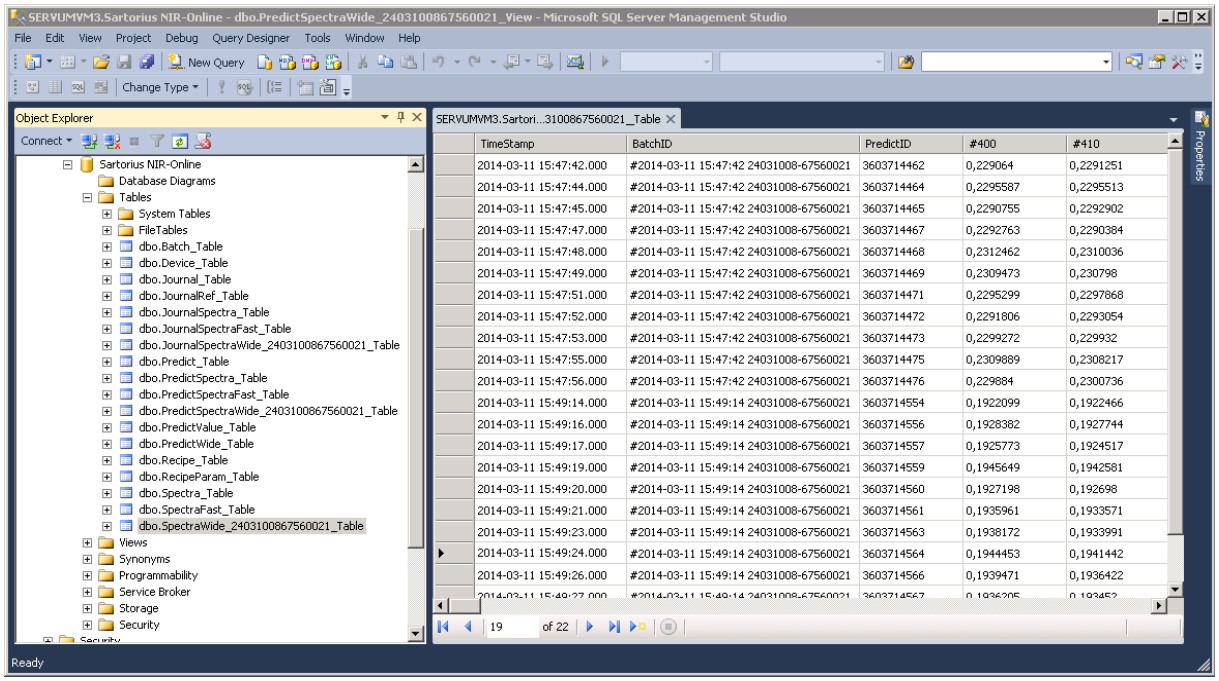

Figure 1. Spectra Wide Table example.

2.2.2 Batch Nodes (or Batch Tables)

The batch nodes (or -tables or -views) contain information about batches such as start time, stop time, and also optionally batch conditions. A batch node is required by SIMCA-online to analyze batch data, but is optional if only continuous data is used.

Each row in a batch node represents one batch.

A batch node needs to have the following columns:

- BatchID – The name of the batch (PK, varchar, not null).
- BatchStart – The start time of the batch when it first enters the entire process (not when it starts in a unit (part of) in the process) (datetime, not null).
- BatchStop – The time when the whole batch is completed in the system (not in a unit), null if not completed (datetime).

Note that the BatchID column should be the primary key.

In addition there can be optional columns as follows, for each batch condition:

- [Batch condition name] – One column for each batch condition. Column name is the tag name, which contains the data (float).
- UnitID – The name of the unit to which the batch is associated with (varchar).

The **names** of the batch node and columns are arbitrary in the database since the names are specified in the configuration file.

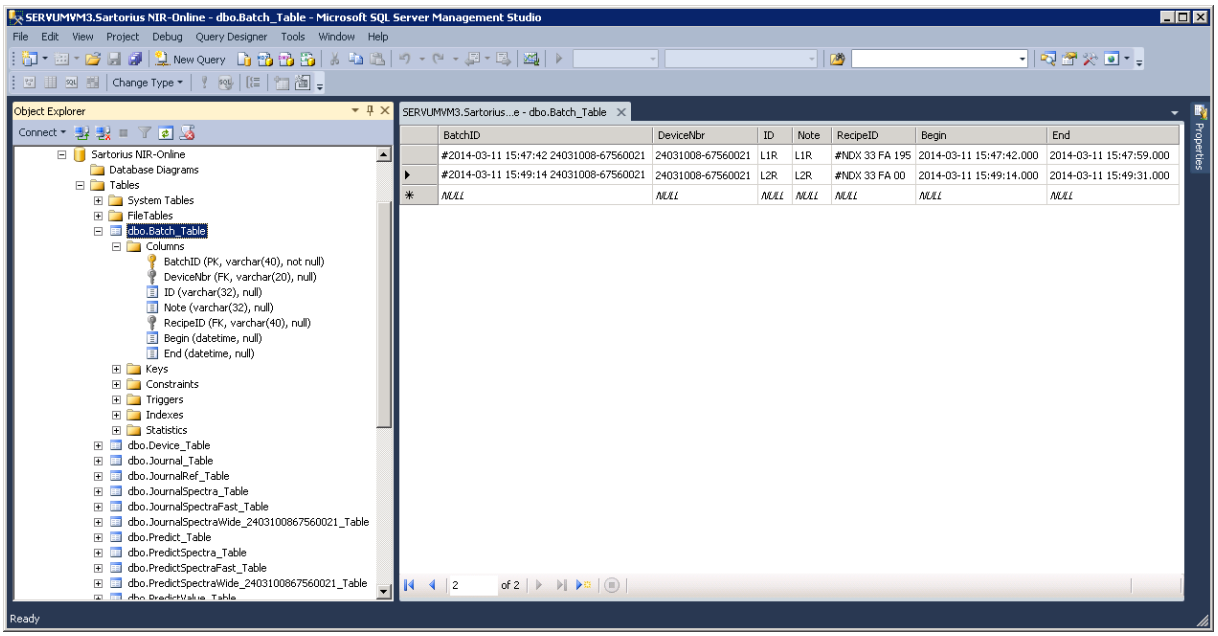

Figure 2. Batch node example.

2.2.3 Discrete Node (or Table)

The optional discrete node (or -table or -view) contain discrete data measurements. A discrete node is required by SIMCA-online if it should be able to analyze discrete data.

Each row in a discrete node represents one measurement for a batch and tag at a given time.

A discrete node needs to have the following columns:

- BatchID – The name of the batch (varchar, not null).
- TagName – The name of the tag (varchar, not null).
- Time – The time when the sample was taken (datetime, not null).
- Value – The measurement value (float).

Note that the combination of BatchID+TagName+Time should be the primary key.

The **name** of the discrete node and columns are arbitrary in the database since the names are specified in the configuration file.

Discrete node example with two tags sampled three times (at roughly 24 hour intervals) per batch.

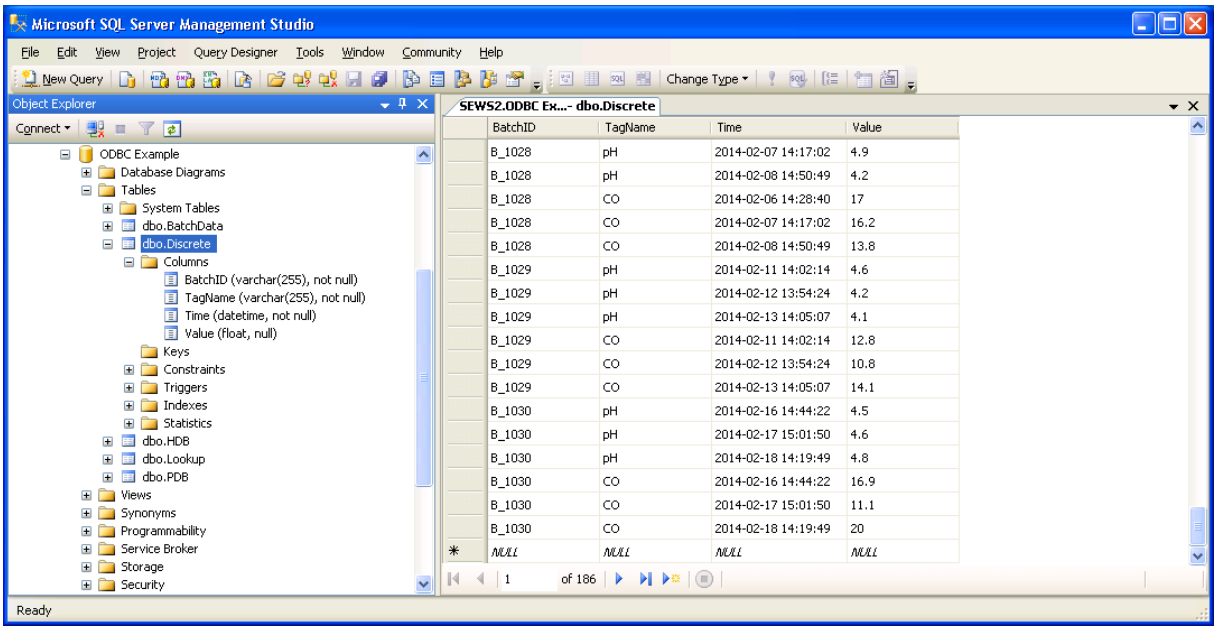

Figure 3. Discrete node example with two tags sampled three times (at roughly 24 hour intervals) per batch.

2.2.3.1 Discrete Tag Lookup Table

This optional table/view is used to query the discrete tags in the system. It defines which discrete tags should be available.

Use for performance optimizations when loading the SimApi or if you want to control what tags are exposed from the SimApi. For instance if you want to expose tags that doesn't have any measurements.

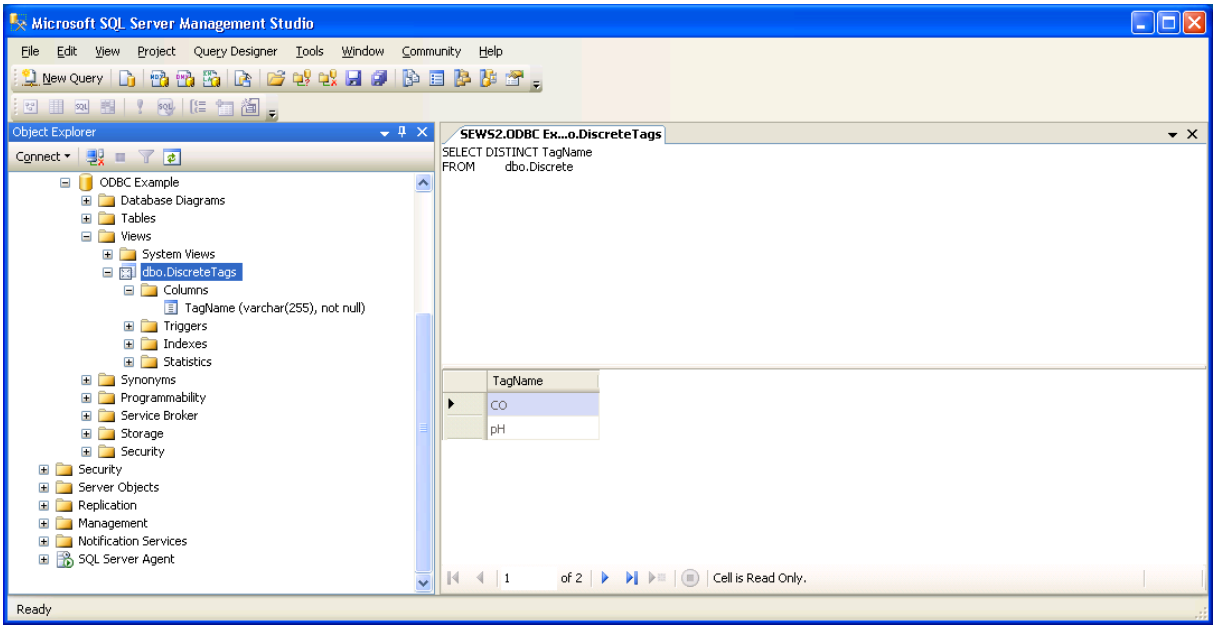

Figure 4. Simple discrete tag lookup view which use the discrete table.

2.2.3.2 Discrete data as seen by SIMCA-online

The following is how the discrete data in **Error! Reference source not found.** will look in SIMCA-online when combined with the

Discrete Extracted Data (at time 2014-05-27 11:39:48)

	1	2	3	4	5	6	7	8
1	BATCH	OBSERVATION	$BatchAge(d)	$BatchAge(h)	$BatchAge(m)	$BatchAge(s)	CO	pH
76	B_1024	2	3,03539	72,8494	4370,97	262258	17,9	4,7
77	B_1025	0	1,02872	24,6892	1481,35	88881	10,8	4,3
78	B_1025	1	2,03596	48,8631	2931,78	175907	11,3	4,1
79	B_1025	2	3,02333	72,56	4353,6	261216	14,3	4,6
80	B_1026	0	1,02106	24,5056	1470,33	88220	10,5	4,9
81	B_1026	1	2,02056	48,4933	2909,6	174576	16,8	4,8
82	B_1026	2	3,04142	72,9942	4379,65	262779	16	4,6
83	B_1027	0	1,04109	24,9861	1499,17	89950	13,7	4,1
84	B_1027	1	2,00284	48,0681	2884,08	173045	13,2	4,6
85	B_1027	2	3,02736	72,6567	4359,4	261564	16,9	4,2
86	B_1028	0	1,01748	24,4194	1465,17	87910	17	4
87	B_1028	1	2,0094	48,2256	2893,53	173612	16,2	4,9
88	B_1028	2	3,03286	72,7886	4367,32	262039	13,8	4,2
89	B_1029	0	1,01223	24,2936	1457,62	87457	12,8	4,6
90	B_1029	1	2,00679	48,1631	2889,78	173387	10,8	4,2
91	B_1029	2	3,01424	72,3417	4340,5	260430	14,1	4,1
92	B_1030	0	1,02598	24,6236	1477,42	88645	16,9	4,5
93	B_1030	1	2,03811	48,9147	2934,88	176093	11,1	4,6
94	B_1030	2	3,00894	72,2144	4332,87	259972	20	4,8

Figure 5. Discrete data as seen by SIMCA-online. Note that the generated batch age tags represent the age of the batch for each observation/sample.

batches in **Error! Reference source not found.**.

3 Installation and setup

Refer to the **SimApi Overview User Guide** located at umetrics.com/simapi for installation and setup within the different softwares in the Umetrics Suite.

3.1 XML Configuration File

The XML configuration file is a text file, it can be edited with for example Notepad. It has the following settings

Log file specific settings	**Explanation**
LogFileSize	The maximum allowed size of the log file before the file is truncated.
LogLevel	The higher the value the more information is printed to the log file. Maximum value is 4 and minimum value is 0. (0=Critical, 1=Error, 2=Warning, 3=Information, 4=Debug).
Connection specific settings	
DSN	Data Source Name as set up in the Windows **ODBC Administrator** control panel.
Credentials	The user name and password for the database (if applicable) encrypted. Both are configured in the Credentials configuration dialog that is displayed when configuring the SimApi. If the user and password is already set up in the ODBC driver, this value can be empty.
SQLDialect	The dialect to use. Could be any of these values: **standard**, **postgresql**, **db2**, **mssql**, **mysql**, **oracle**, **access**, **denodo**. If left blank then the **standard** value will be used as default.
QueryTimeout	The time before a query or connection to the database will time out and fail.
View specific settings	
TimeField	The name of the date/time field (column) in the views (or tables).
Batch node specific settings	
BatchTable	The name of the view or table that contains the batch data. Multiple batch tables can be specified by separating their names a pipe character (|). For example: BT1|BT2|BT3. The following fields (columns) have to be identical in all views.
BatchIDField	The field (column) name of the batch ID in the batch node.
StartTimeField	The field (column) name of the **start** time for the batch.
StopTimeField	The field (column) name of the **stop** time for the batch.
BatchIDUnitField	The field (column) name of the unit ID in the batch node. This field can be used to generate synthetic batch id process tags filtered by unit id.
Discrete view specific settings	
DiscreteTable	The name of the view/table that contains the discrete data.
DiscreteLookupTable	The name of the view/table that contains the discrete tags. If left blank the DiscreteTable will be used.
DiscreteTimeField	The field (column) name of the time of the measurement.

DiscreteBatchIDField	The field (column) name of the identity of the batch that was measured.
DiscreteTagNameField	The field (column) name of the name of the tag that was measured.
DiscreteValueField	The field (column) name of the value of the measurement.

4 Support

umetrics_support-SIMCA-online@sartorius-stedim.com