

# IP21 Batch SimApi User Guide

## 1. Introduction

This document is the user guide for the *IP21 Batch SimApi* from Umetrics.

The SimApi uses the SDK for IP21 Batch to connect to one or more IP21 servers.

For a detailed list of changes in different versions of this SimApi, see the *IP21 Batch SimApi Version Info.txt* file that comes with the installation.

This SimApi can be used by SIMCA-online or SIMCA.

For more information on available SimApis, see [www.umetrics.com/simapi](http://www.umetrics.com/simapi).

### 1.1 Features

This SimApi implements;

- Can connect to several IP21 servers.
- Configurable set of *areas* to expose through the SimApi.
- Batch node functionality; read information on batches in the system.
- Batch data –batch conditions or quality attributes. One observation for each batch.
- Synthetic process tag for PhaseID, each phase has one or more configured steps.
- Synthetic process tag for BatchID.
- Synthetic process tag for FilterID.

Note that this SimApi does not read continuous (process) data except for the synthetic tags. For continuous data you can use the IP21 SimApi.

### 1.2 Prerequisites

The following packages need to be present on the computer where you want to run the SimApi. You don't need to install the software if you already have it, but if not we provide download locations for publically available software:

.Net Framework 4 or higher is needed. You can download it from here if you don't already have it: <http://www.microsoft.com/en-US/download/details.aspx?id=42642>

Visual C++ Redistributable for Visual Studio 2012 and 2010 is needed. Download locations: <http://www.microsoft.com/en-us/download/details.aspx?id=30679> and <http://www.microsoft.com/en-us/download/details.aspx?id=5555>

SDK for IP21 Batch, this also installs the *AtBatch21ApplicationInterface.dll* that this SimApi uses. This SDK is obtained from AspenTech, Umetrics cannot provide this DLL.

An ODBC data source that connects to the IP21Batch server needs to be configured since the *AtBatch21ApplicationInterface.dll* uses this to communicate with the server. Make sure the correct version, 32-bit or 64-bit, is configured so it matches the installed SimApi version.

### 1.3 Installation

The SimApi comes in a 32-bit (x86) and a 64-bit (x64) version. Use the one that matches the program (for example SIMCA-online or SIMCA) you will use the SimApi in.

1. (Make sure the software packages are installed and that the ODBC connection is in place as mentioned in 1.2 above.)

2. Run the SimApi setup program to install the SimApi in a folder with read and write permissions since the log file is in the same folder as the SimApi.

3. Register the AtIP21BatchApplicationInterface.dll by opening a command prompt with administrator privileges, go to the folder where the AtIP21BatchApplicationInterface.dll is located and type:

```
regsvr32.exe AtIP21BatchApplicationInterface.dll
```

A confirmation dialog will open and state if the registration was successful or not.

If the 64-bit version of this SimApi is used register both versions. The 64-bit version of the *AtIP21BatchApplicationInterface.dll* is usually located in C:\Program Files\Common Files\AspenTech Shared.

4. Configure the SimApi as described in 2.3 below.

When the SimApi is installed the following files should be available:

- Umetrics.IP21BatchSimApi.dll (the SimApi)
- Umetrics.IP21Batch.dll
- Umetrics.IP21Batch.xml will be created after the first use of the SimApi dll and needs to be configured manually by editing the file with a text editor (see 2.3 below).
- AtIP21BatchApplicationInterface.dll (Installed by the SDK for IP21 Batch)

## 1.4 Troubleshooting

Use the SimApi log file and the SIMCA-online server log files to determine what is happening. See <http://www.umetrics.com/kb/making-use-simca-online-server-logging-finding-problems> for more information.

Also see the [SIMCA-online technical guide](#) for more information on SimApi troubleshooting, including issues that can occur when trying to use a 32-bit SimApi on a 64-bit windows computer.

## 1.5 Configuration file and log file locations

The SimApi stores its settings in a XML settings file in the same directory as the SimApi:

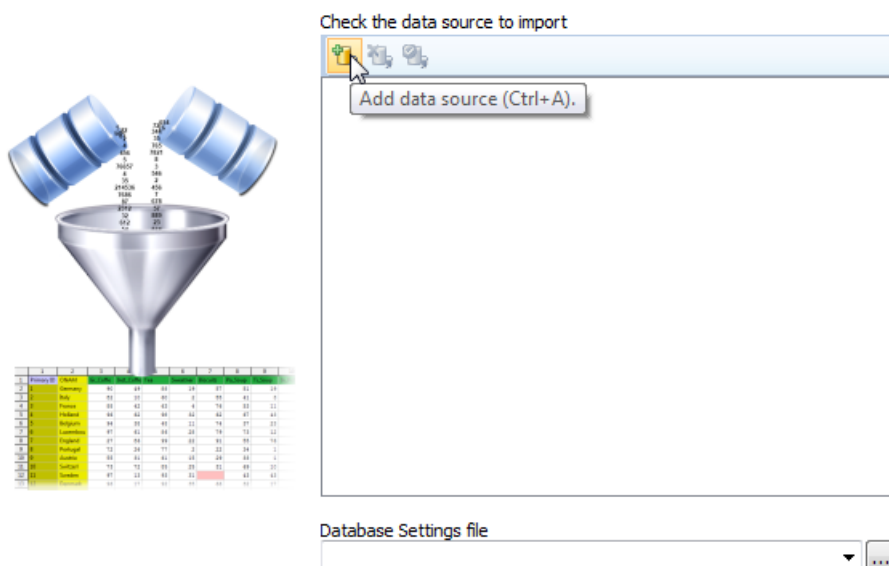
The SimApi log file will also be written to this directory. This log file is useful for troubleshooting.

## 2. Setting up

In order for SIMCA or SIMCA-online to be able to use the SimApi you need to configure it as described here.

### 2.1 Setting up the SimApi for use in SIMCA

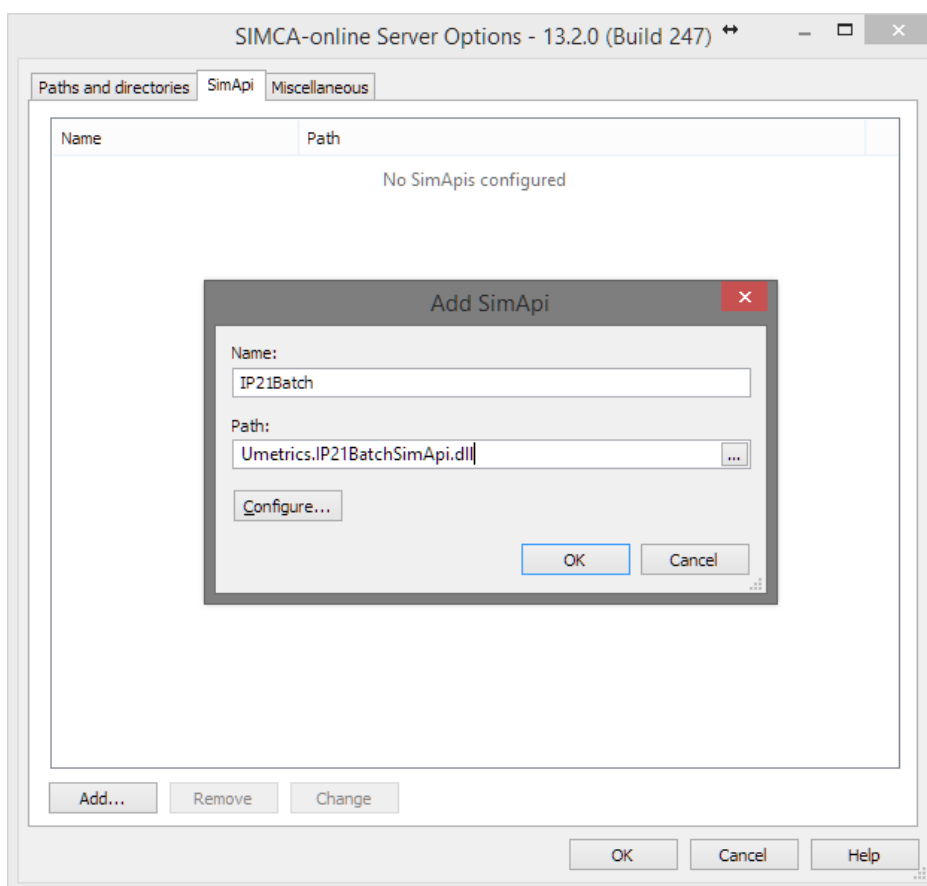
Start SIMCA-import by *File->New Regular Project->New Batch Project*. If the database import wizard is not opened automatically, open it from *File->New Spreadsheet->From Database*. Click on *Add data source*.



Select *SimApi* as connection type, find the installed *Umetrics.IP21BatchSimApi.dll*. Continue to **Error! Reference source not found..**

## 2.2 Setting up the SimApi for use in SIMCA-online

Start the SIMCA-online Server Options utility from the start menu. Go to the SimApi tab and click *Add...*



Enter a name to use for the SimApi in SIMCA-online, for example *IP21Batch*, and then click the *...*-button to browse to and select the *Umetrics.IP21BatchSimApi.dll*. All settings for this SimApi is done in the automatically generated XML file called *Umetrics.IP21Batch.xml*. Note that this file is

generated upon first use of the dll so you might need to start the SIMCA-online server first in order to get a default copy of the file that you can edit, unless you have a configuration file to start with.

## 2.3 The XML Configuration File

Below a sample XML file is shown. This example connects to two different IP21 Batch Servers, note that the setting StatusValueRunning is using a string type for one server and an int type for the other.

```
<?xml version="1.0" encoding="utf-8"?>
<Configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <LogFileSize>20</LogFileSize>
  <Servers>
    <ServerConfig>
      <IP21ServerName>Example server name 1</IP21ServerName>
      <Dialect>Standard</Dialect>
      <Nodes>
        <NodeConfig>
          <Name>example area name 1</Name>
          <BatchID>BatchID</BatchID>
          <Status>STATUS</Status>
          <StatusValueRunning xsi:type="xsd:string">ACTIVE</StatusValueRunning>
          <StatusValueFinished xsi:type="xsd:string">FINISHED</StatusValueFinished>
          <SubLevel>0</SubLevel>
          <Instance>0</Instance>
          <SyntheticBatchIDTag>true</SyntheticBatchIDTag>
        </NodeConfig>
      </Nodes>
    </ServerConfig>
    <ServerConfig>
      <IP21ServerName>Example server name2</IP21ServerName>
      <Dialect>Standard</Dialect>
      <Nodes>
        <NodeConfig>
          <Name>example area name 2</Name>
          <BatchID>BatchID</BatchID>
          <Status>STATUS</Status>
          <StatusValueRunning xsi:type="xsd:int">1</StatusValueRunning>
          <StatusValueFinished xsi:type="xsd:int">0</StatusValueFinished>
          <SubLevel>0</SubLevel>
          <Instance>0</Instance>
          <SyntheticBatchIDTag>>false</SyntheticBatchIDTag>
        </NodeConfig>
      </Nodes>
    </ServerConfig>
  </Servers>
</Configuration>
```

Below are the available settings in the XML file. These settings are marked bold below, and described in the sub sections.

### 2.3.1 Connection Specific Settings

- **LogFileSize** – The log file size in MB.
- **Servers** – Several ServerConfigs can be configured under this section.
  - **ServerConfig** – Contains the settings for the server.
    - **IP21ServerName** – The computer name or IP address that the SimApi should connect to.
    - **Dialect**– There are two different dialects, Standard and ExitStatus. Default is Standard.
    - **Nodes** – Several NodeConfigs can be configured under this section.
  - **NodeConfig**– This can be an area or a node under an area. One area must be configured. Any nodes under the area should be configured under SubNodes.

Each NodeConfig configured as an area will be exposed as a node through the SimApi.

- **Name** – Name of area or node in IP21.
- **Alias** – Name that is exposed through the SimApi to SIMCA-online for instance.
- **BatchID** – Name of the characteristic that contains the batch id.
- **Status** – Name of the characteristic that contains the status for the batch.
- **StatusValueRunning** – Value of the status characteristic indicating the batch is running. This can be either a string or a number, see the sample XML file above on how to configure this.
- **StatusValueFinished** – Value of the status characteristic indicating the batch is finished. This can be either a string or a number, see the sample XML file above on how to configure this.
- **StatusValueInterrupted** – Value of the status characteristic indicating the batch was interrupted. This can be either a string or a number. Configure in a similar way as StatusValueRunning and StatusValueFinished.
- **SubLevel** – Sub level that this node belongs to, area is level 0. First level under area = 1, second level under area = 2 and so on.
- **Instance** – Instance number, can have several instances of the same unit. Default this is not used and is set to 0.
- **FilterTag** – Name of the characteristic that contains the filter name. This can be used if a node should be filtered against, for instance, a certain product and then the product is specified in the specified characteristic.
- **FilterCompareValue** – Value that should be filtered against.
- **SyntheticAreaFilterValueIDTag** – Synthetic tag called FilterID that returns the value from the FilterTag characteristic. Default is false, i.e. the synthetic tag will not be exposed through the SimApi. Note setting can only be used for an Area.
- **SyntheticBatchIDTag** – Synthetic tag called BatchID that returns the value from the BatchID characteristic. Default is false, i.e. the synthetic tag will not be exposed through the SimApi.
- **SubNodes** – Nodes under the NodeConfig are configured here.
- **PhaseValues** – Several PhaseValueConfigs can be configured under this section. When configured a synthetic PhaseID tag will become available through the SimApi.
  - **PhaseValueConfig** – Represents a phase.
    - **Name** – Name of the phase.
    - **StepConfigs** – Several StepConfigs can be configured under this section.
      - **StepConfig** – Represents a step.
        - **Name** – Name of the step.
        - **Instance** – Instance number, can have several instances of the same step name.
        - **UseEndTimeAsStartTime** – The stop time will be used to describe when this step is active in this phase. This step will be active until the next step becomes active. This can be useful when the synthetic phase tag is used.

### **3. Support**

See <http://www.umetrics.com/support>