# Table of Contents

Abbreviations & Notations.................................................................................................................. 3
Introduction ........................................................................................................................................... 3
Purpose and functionality ....................................................................................................................... 3
Interfaces ............................................................................................................................................... 3
Compatibility ........................................................................................................................................ 4
License file ........................................................................................................................................... 4
Resource ID ......................................................................................................................................... 4
Error handling ..................................................................................................................................... 4
Model number ...................................................................................................................................... 4
Call sequence ....................................................................................................................................... 4
Performing a prediction .......................................................................................................................... 5
  Order of input data .............................................................................................................................. 5
Quantitative vs. qualitative functions ..................................................................................................... 5
Lagged variables ................................................................................................................................... 5
Filtered data ......................................................................................................................................... 6
Missing values ....................................................................................................................................... 6
Retrieving results .................................................................................................................................. 6
  VectorData .......................................................................................................................................... 6
  Component number ............................................................................................................................ 6
  Column ............................................................................................................................................... 6
Contributions ......................................................................................................................................... 7
  Score contributions ............................................................................................................................. 7
  Contribution for DModX ........................................................................................................................ 7
  Contribution for DModY ....................................................................................................................... 8
  Group Contributions ........................................................................................................................... 8
  Contribution scores SIMCA-Q vs. SIMCA .......................................................................................... 8
Control Charts ...................................................................................................................................... 9
  Shewhart Control Chart ..................................................................................................................... 9
  Cusum Control Chart .......................................................................................................................... 9
  EWMA Control Chart .......................................................................................................................... 9
Generated variables .............................................................................................................................. 9
Threading ............................................................................................................................................... 9
Samples .................................................................................................................................................. 10
Reference documents............................................................................................................................ 10
**Abbreviations & Notations**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIMCA-Q</td>
<td>Umetrics Q-series, includes SIMCA-QP, SIMCA-QP+ and SIMCA-QM</td>
</tr>
<tr>
<td>SIMCA-QP</td>
<td>The prediction functionality of SIMCA-Q for non-batch projects</td>
</tr>
<tr>
<td>SIMCA-QP+</td>
<td>The prediction functionality of SIMCA-Q for batch projects</td>
</tr>
<tr>
<td>SIMCA-QM</td>
<td>The modeling functionality of SIMCA-Q for non-batch projects</td>
</tr>
<tr>
<td>usp file</td>
<td>File created with any of Umetrics products, with the extension .usp</td>
</tr>
<tr>
<td>MVA</td>
<td>Multivariate data analysis</td>
</tr>
</tbody>
</table>

**Introduction**

The main purpose of this document is to describe the functionality and interfaces of SIMCA-QP. SIMCA-Q is not a “plug and play” product, but is meant to be encapsulated within another product. The SIMCA-Q component is a DLL file with a C interface and a COM interface.

Basic knowledge of MVA is necessary in order to understand the terminology used in SIMCA-Q. We therefore recommend attending a course in MVA before starting the development.

**Purpose and functionality**

SIMCA-QP is intended to serve as a calculation engine for non-batch data and to produce MVA predictions on request. It is event-driven, i.e. it only makes predictions upon request, and is otherwise idle. Predictions can only be requested for a non-batch project found in a usp file created with SIMCA or other Umetrics product. All communication with SIMCA-QP is handled through a given interface.

SIMCA-Q can, depending on licensing, handle predictions for multiple projects. The user is responsible for opening and closing projects.

**Interfaces**

SIMCA-Q has two main interfaces: a C interface and a COM interface.

The interface for SIMCA-QP consists of several functions. An overview of the functions can be found in R1 (C interface) and R2 (COM interface). Detailed descriptions are found in the generated help from the header files R3 (C interface) and R4 (COM interface).

Note: Necessary header files and import libraries are delivered with the SIMCA-Q DLL.
Compatibility
SIMCA-Q 13 is compatible only with projects created by SIMCA 13 or other Umetrics products of version 13. Projects created with earlier versions must first be converted to a version 13 project through SIMCA 13.

License file
SIMCA-Q cannot be used unless accompanied by a license file (UmetricsLicense.SSQP). The license file controls how SIMCA-Q may be used, e.g. the number of projects that can be loaded, etc.

The license file should normally reside in the same directory as the SIMCA-Q.dll file, but this can be overridden by a function call.

Resource ID
Resource IDs larger than 9999 are reserved by SIMCA-Q.

Error handling
All functions return an error code upon failure. An error message can be retrieved from the error code and all errors are logged in the SIMCA-Q internal log file. The log will contain information regarding where and why the error occurred.

By default the log file is named SIMCA-Q.log and will be located in the Application Data directory. A typical path is C:\Documents and Settings\UserName\Application Data\Roaming\Umetrics\SIMCA-Q\13. Both the name and location of the log file can be changed through a function call.

There is also a function available to disable the creation of the log file, as well as a function to supply SIMCA-Q with a function pointer (a callback method) to which the log information will be sent. The latter function is only available in the C interface.

Model number
The model number is required as input for some of the functions found in the interface. Note that the model number is not identical to the model index. For example, if one or more models have been deleted from the project, the model index and the model number will differ. The model number is the number of the model in SIMCA, displayed in the “No.” column of the project window.

In the following example we have four models with the model numbers 1, 3, 5 and 6 since model numbers 2 and 4 have been deleted. Model number 6 has not yet been fitted.

Call sequence
The call sequence for SIMCA-QP is as follows:
1. Open a project, with the full path to the usp file as input
2. Load a model, with the model number as input
3. Get info from the model, e.g. T, DModX, etc.
4. Set data for prediction
5. Perform the prediction
6. Get results from prediction, e.g. TPS, DModXPS, etc.
7. Close the project
Steps 4-6, as well as steps 2-6, may be repeated as desired.
Performing a prediction
Predictions can be performed on one observation at a time, or on several observations at once. If performance is of importance, the prediction speed per observation will be much faster if several observations are bundled together.

Order of input data
The order of the data must be the same as that returned by the function `GetVariablesForPrediction`.

Quantitative vs. qualitative functions
There are two different functions to specify data with, one for quantitative data and one for qualitative data. Both functions can be used for both kinds of data, as illustrated in the following example.

Model M1 contains the following variables: Var1, Var2, Var3 (a Qualitative variable with settings A1, B2, and C3) and Var4.
This is the data to use as predictionset:

<table>
<thead>
<tr>
<th>Primari</th>
<th>2</th>
<th>3</th>
<th>Qualitat</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Var1</td>
<td>Var2</td>
<td>Var3</td>
<td>Var4</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1.1</td>
<td>B2</td>
<td>5.3</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1.4</td>
<td>9.5</td>
<td>A1</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>1.2</td>
<td>9.3</td>
<td>C3</td>
</tr>
</tbody>
</table>

Figure 1

`GetVariablesForPrediction` will return Var1, Var2, Var3, and Var4.
There are three different ways to specify the data:
1. Set quantitative data as quantitative data and qualitative as qualitative.
   This is the most straightforward approach. Note that when setting data the index of the variable should always be the index returned by `GetVariablesForPrediction`. Quantitative data will be set for variable index 1, 2 and 4 and qualitative data for variable index 3.
2. Set all as quantitative data.
   Even if there are qualitative data, one can specify this as quantitative. For the first observation we have the setting B2 for the qualitative variable. B2 has the index 2 among the qualitative settings for Var3 (all settings in order can be retrieved via the function `GetQualitativeSettings` for the Var3 variable). To set B2 as a quantitative value, use the index, in this case 2.
3. Set all as qualitative.
   The quantitative data can be set as qualitative simply by making it into a string, e.g. the value 1.1 is represented by the string “1.1”. Note that this approach will significantly slow down performance.

Lagged variables
If the model contains lagged variables, you have the option of setting data for these. If only one or a few observations are to be predicted at a time, it may be necessary to send data for the lagged variables. However, if many observations are to be predicted at once, leaving the lags as missing is not a problem. When lags are not supplied, they will be set as missing for the first time points.

If the data are to be specified, this is done with the functions `SetQuantitativeLagData` and `SetQualitativeLagData`. Specify data for each variable up to the largest lag step.
Example: Assume that the model from Figure 1 above contains three variables that have been lagged: Var1.L1, Var1.L3 and Var2.L2. In order to set data for these variables there must exist three historical values for Var1 and two historical values for Var2, as illustrated in the following table:
Filtered data
If the model contains filtered data, SIMCA-QP will automatically apply the same filter to the predictions. GetVariablesForPrediction will return the original variables that the filter was created from. The user should treat these projects like any other regular project.

Missing values
If a variable lacks data, or has invalid data, it should be given the value that represents a missing value. This must be done for each observation that partially lacks data. However, an observation that lacks data entirely should not be set at all.

The value representing a missing value in SIMCA-QP is accessible through the function GetMissingValue.

Note that if the data to be predicted is dominated by missing values, the prediction will be very time consuming.

Retrieving results
Most results can be requested both from the model and from the prediction. For example, to retrieve scores from the model, use the GetT function from the Model object. To get scores from the prediction, use the GetTPS function from the Prediction object. All vectors from the prediction are identified by a ‘PS’ in the name.

VectorData
All vectors returned from SIMCA-QP will be in the form of a VectorData object. The VectorData object consists of the data matrix plus the column and row names. For example, the following score vector will have a vector data object that contains a matrix (2 columns, 5 rows), column names (M4.t[1], M4.t[2]) and row names (1, 2, 3, 4, and 5).

If the data comes from a prediction, e.g. TPS, the observations will be named in the following pattern: From Q_1_Obs1, From Q_2_Obs2, and so on.

Component number
When requesting data for a vector, you may sometimes need to send the component number as input. Some functions are valid even if there are no components in the model. All other functions will fail for component number zero (0). For an OPLS model, some vectors are only valid for the last predictive component. Whether or not a function is valid for a zero component model is documented in the header files and in the detailed function documentation.

Column
Qualitative variables will be expanded in the VectorData matrix to comprise one column for each setting. For example, when requesting XVar from the model described in Figure 1 on page 5, the VectorData will appear as follows:
Look at columns 4, 5 and 6 above. Notice how the qualitative variable, Var3, has been expanded into three columns for A1, B2 and C3 respectively. These columns will contain a 0 for settings not set, and a 1 for the active setting. To get the names, etc. of the expanded columns, use the functions beginning with GetColumn from the model: GetColumnXSize, GetColumnXNameByIndex, etc.

**Contributions**

Contributions can be requested from both the models and the predicted data.

When you request a contribution, SIMCA-QP requires the weights, component number and type of contribution (Score, DModX, and DModY) as input.

To be able to interpret the result returned from a request for contribution, e.g. to identify which variable each returned column is connected with, the column functions should be used.

**Score contributions**

There are two types of contributions that can be requested for scores: Scores Single Weight or Scores Multi Weight. For batch projects combined contributions can also be constructed.

**Scores Single Weight**

This contribution cannot be requested for a zero component model.

- **Weight**: If the model is a PCA model the weight must be Normalized, Raw, RX or P. For a PLS model the weight must be NoWeight, RX, P, WStar, VIP, CoeffCS or CoeffCSRaw. If the weight is either CoeffCS or CoeffCSRaw, the y-variable must be given as input.

- **Component**: This contribution can only be calculated for a single component.

- **Observations**: To obtain the contribution showing the difference between two observations, the index of the reference observation and the index of the second observation must be provided. The contribution relative to the average of the X variables as reference can be retrieved by setting the index of the reference observation to zero. If only one observation exists in the prediction, the index values will be ignored. In this case, the average of the X variables is used as reference.

**Scores Multi Weight**

This contribution cannot be requested for a zero component model.

- **Weight**: If the model is a PCA model the weight must be P. For a PLS model the weight must be P or WStar.

- **Component list**: This contribution can be calculated for one or more components. A list containing the selected components is given as input.

- **Observations**: To obtain the contribution showing the difference between two observations, the index of the reference observation and the index of the second observation must be provided. The contribution relative to the average of the X variables as reference can be retrieved by setting the index of the reference observation to zero. If only one observation exists in the prediction, the index values will be ignored. In this case, the average of the X variables is used as reference.

**Contribution for DModX**

This contribution can be requested for a zero component model.
- **Weight**: If the model is a PCA model the weight must be Normalized or RX. For a PLS model the weight must be Normalized, RX, VIP or CoeffCS. If the weight is CoeffCS, the y-variable must be given as input. Note! For a zero component model the only valid weight is Normalized, regardless of model type.

- **Component**: This contribution is the observation residuals, multiplied by the selected weight, after the specified number of components.

- **Observations**: Specifies the indexes of the observations in the model or predictions.

### Contribution for DModY

This contribution can only be requested for a PLS model with at least one component.

- **Weight**: The weight must be Normalized or RY.

- **Component**: This contribution is the observation residuals, multiplied by the selected weight, after the specified number of components.

- **Observations**: Specifies the indexes of the observations in the model or predictions.

### Group Contributions

Contributions can be requested for a group of observations. The only difference in the interface is that a list of observations is requested as input instead of a single observation.

### Contribution scores SIMCA-Q vs. SIMCA

The following table describes the relationship between contributions in SIMCA-Q and SIMCA.

<table>
<thead>
<tr>
<th>Contribution type</th>
<th>Weight</th>
<th>Contribution in SIMCA-P</th>
</tr>
</thead>
<tbody>
<tr>
<td>ScoresSingleWeight</td>
<td>Normalized</td>
<td>Contribution Scores with weight Normalized.</td>
</tr>
<tr>
<td>ScoresSingleWeight</td>
<td>Raw</td>
<td>Contribution Scores with weight Raw.</td>
</tr>
<tr>
<td>ScoresSingleWeight</td>
<td>P</td>
<td>Contribution Scores with weight p</td>
</tr>
<tr>
<td>ScoresSingleWeight</td>
<td>WStar (PLS only)</td>
<td>Contribution Scores with weight w*</td>
</tr>
<tr>
<td>ScoresSingleWeight</td>
<td>RX (PCA)</td>
<td>Contribution Scores with weight RX</td>
</tr>
<tr>
<td>ScoresSingleWeight</td>
<td>RX (PLS)</td>
<td>No equivalent in SIMCA but not illegal</td>
</tr>
<tr>
<td>ScoresSingleWeight</td>
<td>CoeffCS (PLS only)</td>
<td>Contribution Scores with weight CoeffCS</td>
</tr>
<tr>
<td>ScoresSingleWeight</td>
<td>CoeffCSRaw (PLS only)</td>
<td>Contribution Y-predicted</td>
</tr>
<tr>
<td>ScoresSingleWeight</td>
<td>VIP</td>
<td>Contribution Scores with weight VIP</td>
</tr>
<tr>
<td>ScoresMultiWeight</td>
<td>P with 2 components</td>
<td>Contribution Scores with weight pp</td>
</tr>
<tr>
<td>ScoresMultiWeight</td>
<td>P with more than 2 consecutive components</td>
<td>Contribution Scores with weight PRange</td>
</tr>
<tr>
<td>ScoresMultiWeight</td>
<td>P with more than 2 non-consecutive components</td>
<td>No equivalent in SIMCA but not illegal</td>
</tr>
<tr>
<td>ScoresMultiWeight</td>
<td>WStar with 2 components (PLS only)</td>
<td>Contribution Scores with weight w<em>w</em></td>
</tr>
<tr>
<td>ScoresMultiWeight</td>
<td>WStar with more than 2 consecutive components (PLS only)</td>
<td>Contribution Scores with weight W*Range</td>
</tr>
<tr>
<td>ScoresMultiWeight</td>
<td>WStar with more than 2 non-consecutive</td>
<td>No equivalent in SIMCA but not illegal</td>
</tr>
</tbody>
</table>
Control Charts
Control charts are used to observe the data measured on a process or product over time and thereby detect process upsets, shifts, trends, etc. With all control charts, the target and standard deviation can either be estimated or entered by the user. Unlike SIMCA, there is no need to select the data, item, number and component. The values to be used will be put into a vector.

Shewhart Control Chart
Two types of Shewhart charts are available in SIMCA-QP: Mean/Range and Mean/StdDev. The sample size must be smaller than the number of observations, and between 2 and 25.

Cusum Control Chart
This chart is used to detect a deviation from the target. The sample size must be smaller than the number of observations, and between 2 and 25.

\[ \text{GetHighCuSum}() \] Retrieves the cumulative sum on the high-side difference and is used to detect a deviation from the target on the high side.

\[ \text{GetLowCuSum}() \] Retrieves the cumulative sum on the low-side difference and is used to detect a deviation from the target on the low side.

EWMA Control Chart
EWMA stands for Exponentially Weighted Moving Averages. The sample size must be smaller than the number of observations, and between 2 and 25. The lambda can either be estimated or entered by the user. When not entered by the user, it will be estimated to a value that will minimize the error sum of squares.

Generated variables
Generated variables can be a part of the project using SIMCA-QP. However, there are some limitations concerning which generated variables can be handled by SIMCA-QP. The following generated variables will not be handled by SIMCA-QP:

Generated variables that have been lagged, generated variables with phase, and model “options” (e.g. score).

If the variables were created using a plug-in DLL, the path to the directory containing the plug-in DLL must be provided in the SetPluginPath function. The function must be called before the first call to OpenProject. If the function is not called, the plug-in DLL is assumed to be in the same directory as the SIMCA-Q DLL.

Threading
SIMCA-Q can be used in different threads per project. This means that one thread can call OpenProject with one usp file and another thread can call OpenProject with a different usp file. The two threads will then work in parallel on the different projects. Performing operations on the same project from different threads is not guaranteed to be thread safe.

SIMCA-Q uses different threads internally to perform certain calculations. This can be turned on or off by a call to UseMultiThreading. Multi-threading is turned off by default and should remain off for small projects where the overhead needed to create several threads is too large.
Samples
A number of samples illustrating how to use SIMCA-QP from different environments are available in the samples folder. Some of these, such as VB, C# and C++, encapsulate the C/COM interface in a higher level interface called EzQ, demonstrating how a higher level interface can be created.

Reference documents

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Name</th>
<th>Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>SIMCA-QP C Interface Description.pdf</td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>SIMCA-Q COM Interface Description.pdf</td>
<td></td>
</tr>
<tr>
<td>R3</td>
<td>SIMCA-QP C Interface.chm</td>
<td></td>
</tr>
<tr>
<td>R4</td>
<td>SIMCA-QP COM Interface.chm</td>
<td></td>
</tr>
</tbody>
</table>