

SIMCA-Q

Embedded solution

Quick Tutorial SIMCA-Q

12 May 2023

This tutorial will demonstrate in different lessons how to use SIMCA®-Q embedded solution. It will use the COM interface, but the C interface is similar, with the difference that all functions starts with SQ_ and the object to perform the operation on is the first parameter. For example the COM function project.GetModel(1) corresponds to the C function SQ_GetModel(projectHandle,1).

Lessons

Lesson 1 – Get model information

This lesson will demonstrate how to open a project, get the first model and get the name of that model.

```
// Instantiate the SIMCA-Q object
SIMCAQ simcaq = new SIMCAQ();

// unlock the functionality of SIMCA-Q:
simcaq.SetOEMPassword("monkey123"); // the OEM password specified in your agreement (.$SQ license
file). Omit this step if no password is used

// Open the project
Project project = simcaq.OpenProject(projectPath, nullptr);

// The the number of the first model
int modelNumber = project.GetModelNumberFromIndex(1);

// Get the model object of the first model
Model model = project.GetModel(modelNumber);

// Get the name of the first model
String title = model.GetModelName();
```

Lesson 2 – Get extended model information

The second lesson will demonstrate how to retrieve model results from the first model.

```
// Instantiate the SIMCA-Q object
SIMCAQ simcaq = new SIMCAQ();

// Open the project
Project project = simcaq.OpenProject(projectPath, nullptr);

// The number of the first model
int modelNumber = project.GetModelNumberFromIndex(1);

// Get the model object of the first model
Model model = project.GetModel(modelNumber);

// Get number of components in the model
int numComponents = model.GetNumberOfComponents();
```

```
// Construct a vector with the components to get the values for
IntVector componentVector = simcaq.GetNewIntVector(2);

// Set first and last components. Note that the vector is 1-based.
componentVector.SetData(1, 1);
componentVector.SetData(2, numComponents);

// Get the scores vector
VectorData resultData = model.GetT(componentVector);

// Print the information
StringVector columnNames = resultData.GetColumnNames();
StringVector rowNames = resultData.GetRowNames();
FloatMatrix datamatrix = resultData.GetDataMatrix();
int numRows = datamatrix.GetNumberOfRows();
int numColumns = datamatrix.GetNumberOfCols();

// Print column names
for (int col = 1; col <= numColumns; ++col)
{
    Write(columnNames.GetData(col));
}

// Print the data
for (int row = 1; row <= numRows; ++row)
{
    Write(rowNames.GetData(row));
    for (int col = 1; col <= numColumns; ++col)
    {
        Write(datamatrix.GetData(row, column));
    }
    Write("\n"); // New line
}
}
```

Lesson 3 – Do a prediction

The third lesson will demonstrate how to make a prediction. The prediction set comes from somewhere else, for example read from file.

```
// Instantiate the SIMCA-Q object
SIMCAQ simcaq = new SIMCAQ();

// unlock the functionality of SIMCA-Q:
simcaq.SetOEMPassword("monkey123"); // the OEM password specified in your agreement (.$SQ license
file). Omit this step if no password is used

// Open the project
Project project = simcaq.OpenProject(projectPath, nullptr);

// The the number of the first model
int modelNumber = project.GetModelNumberFromIndex(1);

// Get the model object of the first model
Model model = project.GetModel(modelNumber);

// Get number of components in the model
int numComponents = model.GetNumberOfComponents();

// Construct a vector with the components to get the values for
IntVector componentVector = simcaq.GetNewIntVector(2);
```

```

// Set first and last components. Note that the vector is 1-based.
componentVector.SetData(1, 1);
componentVector.SetData(2, numComponents);

// Do a prediction
PreparePrediction prepPred = model.PreparePrediction();

// Get the variables needed for this prediction
VariableVector variableVector = prepPred.GetVariablesForPrediction();

// Set all data in the prediction set that should be used for the prediction
int numVariables = variableVector.GetSize();
for (int var = 1; var <= numVariables; ++var)
{
    for (int row = 1; row <= numRowsInPredictionset; ++row)
    {
        prepPred.SetQuantitativeData(row, var, externalPredictionSet[row-1][var-1]);
    }
}

// Do the actual prediction
Prediction prediction = prepPred.GetPrediction();

// Get the predicted scores vector from the prediction
VectorData resultData = prediction.GetTPS(componentVector);

// Print the information
StringVector columnNames = resultData.GetColumnNames();
StringVector rowNames = resultData.GetRowNames();
FloatMatrix datamatrix = resultData.GetDataMatrix();
int numRows = datamatrix.GetNumberOfRows();
int numColumns = datamatrix.GetNumberOfCols();

// Print column names
for (int col = 1; col <= numColumns; ++col)
{
    Write(columnNames.GetData(col));
}

// Print the data
for (int row = 1; row <= numRows; ++row)
{
    Write(rowNames.GetData(row));
    for (int col = 1; col <= numColumns; ++col)
    {
        Write(datamatrix.GetData(row, column));
    }
    Write("\n"); // New line
}

```

Lesson 4 – Batch projects

This lesson will demonstrate how to open a batch project, get the first batch level model, the first batch evolution model and get the aligned T from the batch evolution model.

```

// Instantiate the SIMCA-Q object
SIMCAQ simcaq = new SIMCAQ();

```

```
// unlock the functionality of SIMCA-Q:
simcaq.SetOEMPassword("monkey123"); // the OEM password specified in your agreement (.$SQ license
file). Omit this step if no password is used

// Open the project
Project project = simcaq.OpenProject(projectPath, nullptr);

// Test if it is a batch project
if (!project.GetIsBatchProject())
    write("In this lesson you'll need a batch project! :");

// Get first batch model
BatchModel batchModel = project.GetBatchModel(1);

// Get the first batch level model
int modelNumber = batchModel.GetBatchLevelModelNumber(1);
BatchLevelModel blModel = batchModel.GetBatchLevelModel(modelNumber);

// Get batch level model title
String title = blModel.GetModelName();

// Get the first batch evolution model
modelNumber = batchModel.GetBatchEvolutionModelNumber(1);
BatchEvolutionModel beModel = batchModel.GetBatchEvolutionModel(modelNumber);

// Construct a vector with the batches to get the values for
IntVector batchesVector = simcaq.GetNewIntVector(2);

// Set first and last batch. Note that the vector is 1-based.
int numBatches = beModel.GetModelNumberOfBatches();
batchesVector.SetData(1, 1);
batchesVector.SetData(2, numBatches);

// Get aligned scores vector
VectorData resultData = beModel.GetAlignedT(1, batchesVector);
```

Lesson 5 – Batch Prediction

The fifth lesson will explain how to do a prediction for a batch project.

```
// Instantiate the SIMCA-Q object
SIMCAQ simcaq = new SIMCAQ();

// unlock the functionality of SIMCA-Q:
simcaq.SetOEMPassword("monkey123"); // the OEM password specified in your agreement (.$SQ license
file). Omit this step if no password is used

// Open the project
Project project = simcaq.OpenProject(projectPath, nullptr);

// Test if it is a batch project
if (!project.GetIsBatchProject())
    write("In this lesson you'll need a batch project! :");

// Get first batch model
BatchModel batchModel = project.GetBatchModel(1);

// Get the first batch level model
int modelNumber = batchModel.GetBatchLevelModelNumber(1);
```

```

BatchLevelModel blModel = batchModel.GetBatchLevelModel(modelNumber);

// Start the prediction
PrepareBatchPrediction prepPred = blModel.GetPrepareBatchPrediction();
int numBEModels = batchModel.GetNumberOfBEM();
for (int phase = 1; phase <= numBEModels; ++phase)
{
    // Set the prediction set data from external prediction set
    VariableVector variableVector = prepPred.GetVariablesForBatchPrediction(phase);
    int numVariables = variableVector.GetSize();
    for (int row = 1; row <= numRowsInPredictionset; ++row)
    {
        for (int var = 1; var <= numVariables; ++var)
        {
            prepPred.SetQuantitativeBatchData(phase, row, var, externalPredictionSet[phase][row -
1][var - 1]);
        }
    }
}

// Do the actual prediction
BatchPrediction batchPrediction = prepPred.GetBatchPrediction();

// Get TPS from the batch level prediction
Prediction prediction = batchPrediction.GetBatchLevelPrediction();
VectorData resultData = prediction.GetTPS(nullptr);

// Get aligned TPS from the batch evolution prediction
modelNumber = batchModel.GetBatchEvolutionModelNumber(1);
BatchEvolutionPrediction bePrediction = batchPrediction.GetBatchEvolutionPrediction(modelNumber);
resultData = bePrediction.GetAlignedTPS(1);

```