Fact Sheet

**COM Interface**

A standard protocol for MS Windows called Component Object Model (COM) is available, which allows different applications – independent of a certain programming language – to intercommunicate with each other.

The SimulationX COM interface enables you to automate often recurring work processes, both within SimulationX itself as well as in co-operation with other software products, especially Microsoft Office world.

**COM Interface Applications**

- **Parametrizing**
  - Import of parameters
  - Parameter identification
  - Parameter studies

- **Simulating**
  - Variant Calculations
  - Optimization
  - Inverse Simulation
  - Software in the Loop

- **Modeling**
  - Model generation
  - Model administration
  - Alternative GUI

- **Evaluating**
  - Export of results
  - Graphic display
  - Validation

The script languages VBScript and JScript are available for the programming of the scripts within MS Windows. VBA (Visual Basic Application) and MS Visual Basic - integrated macro languages in the MS Office products provide a more comfortable access. Certainly you can use any other programming language which supports the access to COM.

**SimulationX** provides a number of objects, which enable the access to models (documents), elements as well as their parameters and result variables.
The following part of a VBA macro shows how the values of all recorded result variables of a SimulationX model can be imported into an MS-Excel Table.

```vba
col = 1
Cells.Clear
Cells.Item(1, col) = "Result variable:"
Cells.Item(2, col) = "Minimum:"
Cells.Item(3, col) = "Maximum:"
Cells.Item(4, col) = "Mean value:"
Cells.Item(5, col) = "RMS value:"
Columns(col).AutoFit
col = col + 1

For Each SimObject In Doc.SimObjects
For Each Result In SimObjects.Results
If Result.Protocol Then
    Cells.Item(1, col) = Result.Comment & " SimObject.Name & ")
    Cells.Item(2, col) = Result.MinValue
    Cells.Item(3, col) = Result.MaxValue
    Cells.Item(4, col) = Result.MeanValue
    Cells.Item(5, col) = Result.RMSValue
Columns(col).WrapText = False
Columns(col).AutoFit
col = col + 1
End If
Next
```

Moreover, you can take advantages of the possibilities of DCOM (Distributed COM). For example, complex parameter studies in SimulationX can be distributed among several computers within the network.