

## Datenblatt

### COM-Schnittstelle

Mit dem Component Object Model (COM) steht unter Windows® eine universelle Schnittstelle zur Verfügung, die es den verschiedensten Anwendungen erlaubt, unabhängig von einer bestimmten Programmiersprache, miteinander zu kommunizieren.

Die COM-Schnittstelle von SimulationX gestattet es, häufig wiederkehrende Arbeitsabläufe zu automatisieren. Dies funktioniert sowohl innerhalb von SimulationX selbst, als auch in Zusammenarbeit mit anderen Applikationen wie Word, Excel sowie einer großen Anzahl anderer Programme aus den verschiedensten Anwendungsbereichen.

#### Möglichkeiten der COM-Schnittstelle

##### Parametrierung

- Import von Parametern
- Parameter-Identifikation
- Parameter-Studien

##### Simulation

- Variantenrechnung
- Optimierung
- Inverse Simulation
- Software in the Loop



##### Modellierung

- Anlegen von Modellen
- Modellverwaltung
- Alternative GUI

##### Evaluierung

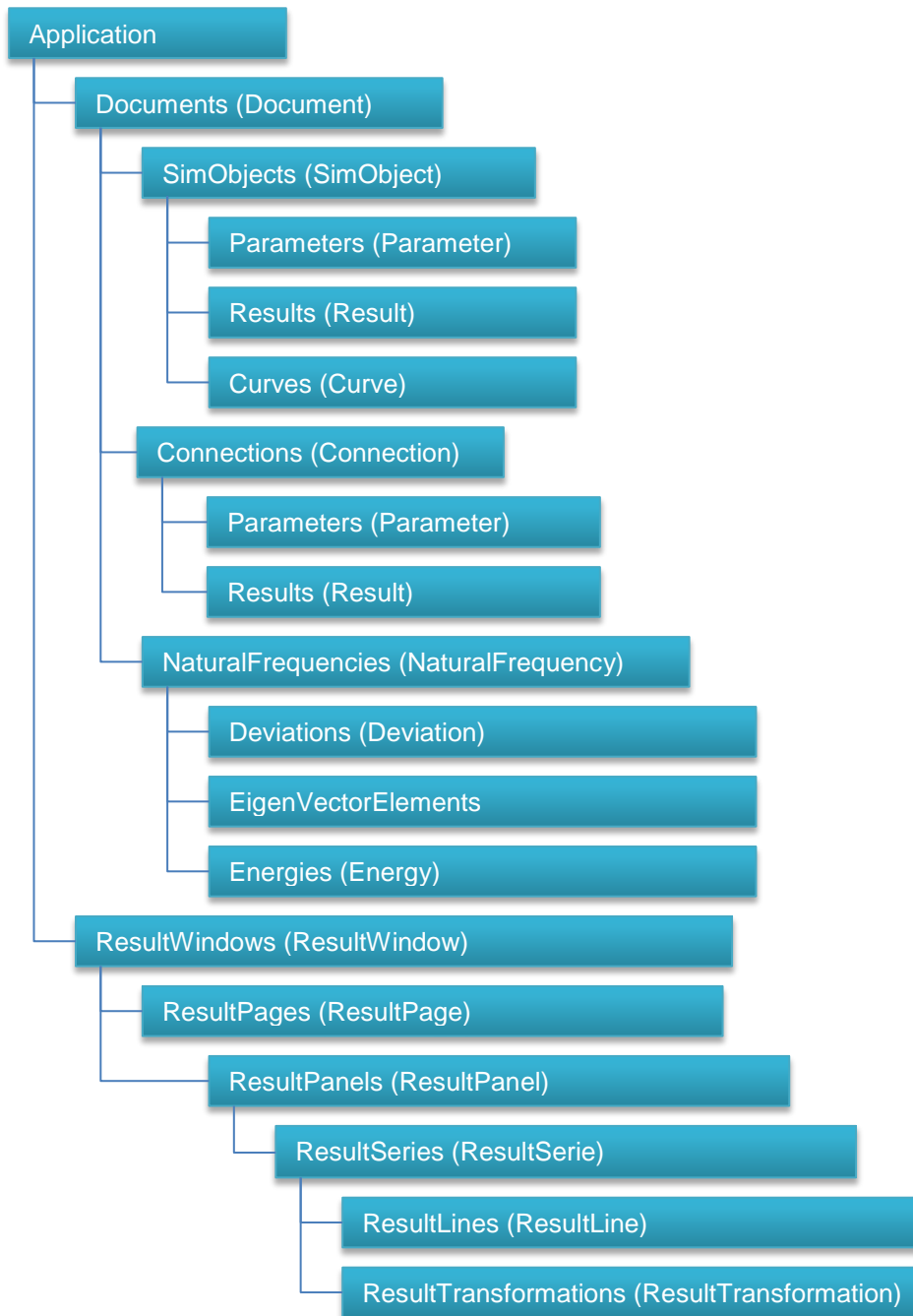
- Export von Ergebnissen
- Grafische Darstellung
- Validierung

- *Simulationsmodelle und Typen verwalten*
- *Zugriff auf alle Parameter eines Modells*
- *Steuerung der Simulationsrechnung*
- *Ergebnisse exportieren und automatisch aufbereiten*
- *Automatisierte Parameterstudien und Optimierung*

Zur Programmierung der zu automatisierenden Abläufe können Sie Script-Sprachen wie VBScript, JScript oder auch Python verwenden. Die in die Microsoft Office-Produkte integrierte Makrosprache VBA (Visual Basic for Applications) kann ebenso wie die aktuellen .Net-Sprachen VB.Net und C# zur Erstellung von SimulationX-Programmerweiterungen eingesetzt werden. Selbstverständlich lässt sich auch jede andere Programmiersprache verwenden, die den Zugriff auf COM unterstützt.

Über das COM-Interface stellt SimulationX eine Vielzahl von Objekten bereit, die den Zugriff auf Modelle und Komponenten sowie deren Parameter und Variablen ermöglichen. Simulationsrechnungen können sowohl im Zeit-, als auch im Frequenzbereich ausgeführt werden. Der vollständige Zugang zu den Simulationsergebnissen ermöglicht die automatisierte Durchführung von Parameterstudien und Optimierungsberechnungen. Zur Einführung in die

Programmierung mit dem SimulationX COM-Interface werden zahlreiche Beispiele mitgeliefert.



Der folgende Ausschnitt aus einem VBA-Makro zeigt die Übernahme von Berechnungsergebnissen eines SimulationX-Modells in eine Excel-Tabelle.

```

col = 1

Cells.Clear
Cells.Item(1, col) = "Ergebnisgröße:"
Cells.Item(2, col) = "Minimum:"
Cells.Item(3, col) = "Maximum:"
Cells.Item(4, col) = "Mittelwert:"
Cells.Item(5, col) = "Effektivwert:"

Columns(col).AutoFit
col = col + 1

For Each SimObject In Doc.SimObjects
  For Each Result In SimObject.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
Next

```

