

SchüCal COM-Schnittstelle SchüCal COM Interface

Interface Developer Guide

Schüco Digital Products

Der Inhalt dieses Dokumentes ist nur für den internen Gebrauch der Schüco Digital GmbH bestimmt. Jede externe Verwendung bedarf einer schriftlichen Erlaubnis der Schüco Digital GmbH. Eine Weitergabe von Inhalt oder Informationen an Dritte, ist ohne Zustimmung der Schüco Digital GmbH nicht erlaubt.

Eventuell im Dokument verwendete Markennamen, Warenbezeichnungen, Gebrauchsnamen etc. unterliegen, auch wenn nicht besonders gekennzeichnet, marken- bzw. patentrechtlichem Schutz.

The content of this document is only permitted for the internal purpose of Schüco Digital GmbH. Any other usage requires the written permission of Schüco Digital GmbH. All Data based on the content of these document are not allowed to be circulated.

Trademarks and copyrights used within this document are not explicitly marked. However, we point out that usage of those trademarks and copyrights is protected by respective laws and patents.

Inhalt / Content

SchüCal & COM	4
1. Ein Projekt inkl. Position in SchüCal anlegen / Creating a project with items in SchüCal	4
1.1. Ein Wohnbau Projekt in SchüCal anlegen / Creating a residential project in SchüCal	5
2. Projekte und Positionen aus SchüCal einlesen / Reading projects and items from SchüCal	5
3. Fertigungslose / Production batches.....	6
3.1. Fertigungslose erzeugen / creating production batches	6
3.2. Fertigungslose ermitteln / Determine production batches	6
4. Projekte / Projects.....	7
4.1. UniqueID.....	7
4.2. Dokument Ein Projekt sperren / Locking a project.....	8
4.3. Ein Projekt entsperren / Unlocking a project.....	8
4.4. Angehängte Dokumente abfragen / Query attached documents.....	8
4.5. Klassifizierung nach EN 1090 / Classification in accordance with EN 1090	10
5. Positionen / Items	10
5.1. Eine Position sperren / Locking an item	10
5.2. Eine Position entsperren / Unlocking an item	10
5.3. Dateianhänge einer Position ermitteln / Determine attachments of an item	11
5.4. Positions-Varianten / Item Variants	11
5.5. Positionsskizze exportieren / Exporting item drawing	12
5.6. Angebotstext einer Position lesen / Reading quotation text of an item	12
6. Virtuelle Fertigungslose / virtual production batches.....	13
6.1. Fertigungsloserzeugung / Creating production batches	13
6.2. Materialtopf über ein Fertigungslos erzeugen / Creating a materialpot by the means of a production batch	13
7. Artikeldaten ermitteln / Determine article data	14
7.1. Allgemeine Artikeldaten / General article data.....	14
7.2. CTB (Concealed Toughened Blind)	17

7.3. CCB (Concealed Composite Blind)	18
7.4. VarioAir	20
7.5. VentoTherm Twist.....	21
7.6. Glas- und Paneeldaten einer Position / Glass and panel data of an item.....	28
7.7. Glas- und Paneeldaten (Einzelmengen) einer Position (Ermittlung der Feldinfo) / Glass and panel data (individual quantity) of an item (determining field info)	31
7.8. Eckvulkanisierte Dichtungsrahmen einer Position / Corner vulcanised gasket frames of an item.....	34
7.9. Türfüllungen / Door infills	36
7.10. Integralmaster	36
7.11. Oberflächendaten einer Position lesen / Reading surface data of an item.....	37
7.12. Bezeichnungen für Oberflächen/Farben.....	39
8. Kalkulation / Cost data	40
8.1. Kalkulatorische Daten einer Position lesen / Reading cost data of an item	40
8.2. Kalkulatorische Daten von zusätzlichen Positionen lesen (Ab SchüCal 2017 R1) / Reading cost data of additional items (from SchüCal 2017 R1)	43
8.3. Beschichtungskosten gesamt / Total coating costs	44
8.4. Beschichtungskosten pro Oberfläche / Coating costs per surface.....	45
9. CE-Kennzeichnung / CE marking.....	45
9.1. CE-Kennzeichnung für Fassaden-Rahmenwerke / CE marking for facade framework	47
9.2. CE-Kennzeichnung für Öffnungen / CE marking for openings	48
9.3. CE-Kennzeichnung für Sonnenschutz / CE marking for solar shading	49
10. Hinweisliste / Message list.....	50
11. Zusätzliche Kosten / Additional costs.....	50
12. Kalkulationsgrunddaten / Cost data block	51
13. Erstellen der XML-Datei für die Fertigungszeiten / Generate xml-file for the fabrication time	52

SchüCal & COM

SchüCal ist eine komponentenbasierte Softwarelösung und stellt im Sinne des Microsoft Component Object Modell (COM) einen ActiveX Server und Container dar.

Das Programm SchüCal stellt aus allen Modulen die wichtigsten Projekte mit ihren Eigenschaften und Methoden über eine COM-Schnittstelle zur Verfügung und hält gleichzeitig Verbindung zu Projekten anderer COM-Server.

Dieses Dokument bietet dem Programmierer eine Hilfe bei der Erstellung von SchüCal-Schnittstellen anhand von VB-Beispielcode an. Es werden alle relevanten Schritte für einen Datenaustausch gezeigt und so für externe Programmierer nachvollziehbar gemacht.

Diese Beispiele sind lediglich als Quellcode-Beispiele zu betrachten. Für ihre Korrektheit wird keine Gewährleistung übernommen.

SchüCal is a component based software solution. For the purpose of Microsoft Component Object Modell (COM) the application SchüCal is an ActiveX Server and Container.

Of all modules the program SchüCal provides the most important projects with their features and methods via a COM-interface and at the same time the program is connected to other projects of different COM-servers.

This document provides the software engineer help for creating SchüCal -interfaces, by means of VB-sample code. In this document all the relevant steps for a data exchange will be demonstrated and accordingly made clear to the software engineer.

The examples are merely to be seen as source code examples without guarantee for their correctness.

1. Ein Projekt inkl. Position in SchüCal anlegen / Creating a project with items in SchüCal

```
s_oApp = New Schuecal2000TLB.Schuecal2000App
'Projekt / Project
Dim strJob As String
'Bearbeiter / User
Dim strWorker As String

'Projekt-Positionen-manager-modul anlegen / Creating project-items manager
module
Dim oJM As JobMaster
oJM = s_oApp.JobMaster

'Container über alle Projekte anlegen / Creating containers over all
projects
Dim oJobs As Jobs
oJobs = oJM.Jobs
'neues Projekt anlegen / Creating a project
Dim newJob As Job
newJob = s_oApp.JobMaster.Jobs.GetNewJob(strJob)

'Position / Item
Dim strItem As String
'Positionsbezeichnung / Item name
Dim strDescription As String

'Container über alle Positionen erstellen / creating container over all
items
Dim oItems As Items
```

```
oItems = newJob.Items

'neue Position anlegen / Creating new item
Dim newItem As Item
newItem = oItems.GetNewItem(strItem)

'Positionsbezeichnung setzen / Setting item name
newItem.Description = strDescription
'Bearbeiter setzen / set worker
newItem.Worker = strWorker
'Stückzahl festlegen / Specify count
newItem.ItemCount = 3
newItem.Save(True)
End Sub
```

1.1. Ein Wohnbau Projekt in SchüCal anlegen / Creating a residential project in SchüCal

```
s_oApp = New Schuecal2000TLB.Schuecal2000App
'Projekt / Project
Dim strJob As String

Dim oJobs As Jobs2
oJobs = s_oApp.JobMaster.Jobs

'neues Projekt anlegen / Creating a project
Dim newJob As Job
newJob = oJobs.GetNewResidentialJob(strJob)

Dim bIsResidentialJob As Boolean
'ist wahr / is true
bIsResidentialJob = newJob.Residential
```

2. Projekte und Positionen aus SchüCal einlesen / Reading projects and items from SchüCal

```
Dim lmaxJob As Long
Dim lmaxItem As Long
Dim oItems As Items
Dim ljob As Long
Dim oJobs As Jobs
Dim lItem As Long
Dim bearbeiter As String
Dim datum1 As Date
Dim datum2 As Date
Dim datum As Date
Dim lZelle As Integer
Dim oJob As Job4

'SchüCal -Applikation anlegen / Creating SchüCal application
Dim SApp As New Schuecal2000App
'Container auf alle Projekte ablegen / Store container on all projects
Set oJobs = SApp.JobMaster.Jobs
'Anzahl der Projekte wird ermittelt / Number of projects will be determined
lJobs = SApp.JobMaster.Jobs.count
```

```
'Alle Projekte durchgehen / Going through all projects
For ljob = 0 To lJobs - 1
'Das Projekt aus SchüCal holen / Fetching project from SchüCal
    Set oJob = SApp.JobMaster.Jobs.Item(ljob)
    'die Projekt-Id einlesen / Load project-Id
    Dim strJob As String
    strJob = oJob.JobID
    'Projektbezeichnung einlesen / Load project name
    Dim strJobsName As String
    strJobsName = oJob.ObjectName
    'Die Positionen aus SchüCal holen / Fetching items from SchüCal
    Set oItems = oJob.Items
    'Anzahl der Positionen wird ermittelt / Number of items is determined
    lmaxItem = oItems.count
    'Alle Positionen durchgehen / Going through all items
    For lItem = 0 To lmaxItem - 1
        Dim oItem As Item
        'Die Position aus SchüCal holen / Fetching item from SchüCal
        Set oItem = oItems.Item(lItem)
        'Erstellungsdatum einlesen / Load creation date
        Dim oTime As Date
        oTime = oItem.CreationDate
        'Positions-Id einlesen / Load item-Id
        Dim strItem As String
        strItem = oItem.ItemID
        'Bearbeiter einlesen / Load user
        Dim strWorker As String
        strWorker = oItem.Worker
    Next
Next
```

Existierten Wohnbau- und reguläre Projekte mit dem gleichen Namen, so wird über ‚GetNewJob‘ immer das reguläre Projekt bevorzugt. Um das Wohnbauprojekt zu erhalten muss ‚GetNewJob‘ mit der UniqueID des gewünschten Projektes aufgerufen werden.

If residential and regular projects with the same name exist, the regular project will be returned by 'GetNewJob'. To guarantee that the residential project is returned, 'GetNewJob' must be called with the UniqueID of the desired project.

3. Fertigungslose / Production batches

3.1. Fertigungslose erzeugen / creating production batches

```
s_oApp = New Schuecal2000TLB.Schuecal2000App
Dim oFactory As Object
oFactory = s_oApp.Factory
Dim LotFileIndex As Schuecal2000TLB.JimLotFileIndex
LotFileIndex = oFactory.GetClassObject("JimLotFileIndex")
LotFileIndex.CreateNewLot("FL1")
```

3.2. Fertigungslose ermitteln / Determine production batches

```
s_oApp = New Schuecal2000TLB.Schuecal2000App
Dim oFactory As Object
oFactory = s_oApp.Factory
Dim LotFileIndex As Schuecal2000TLB.JimLotFileIndex
```

```

LotFileIndex = oFactory.GetClassObject("JimLotFileIndex")
Dim oAllLots As Schuecal2000TLB.JimLots
'Den Container auf alle Fertigungslose / Create container on all production
batches
oAllLots = LotFileIndex.AllLots
Dim lLotsCount As Long
lLotsCount = oAllLots.Count

Dim lRun As Long
For lRun = 0 To lLotsCount - 1
    Dim oLot As Schuecal2000TLB.JimLot
    oLot = oAllLots.Item(lRun)
    Dim strLotname As String
    strLotname = oLot.LotName(oLot.UniqueID)
    Dim lItemRun As Long
    Dim oLotItems As Schuecal2000TLB.JimLotItems
    oLotItems = oLot.GetLotItems(False)
    Dim lItemsCount As Long
    lItemsCount = oLotItems.Count
    For lItemRun = 0 To lItemsCount - 1
        Dim LotItem As Schuecal2000TLB.JimLotItem
        LotItem = oLotItems.Item(lItemRun)
        'Position-ID / Item-ID
        Dim stritem As String
        stritem = LotItem.ItemID
        'Projekt-ID / Project-ID
        Dim strJob As String
        strJob = LotItem.Job
        Dim lCount As Long
        lCount = LotItem.ItemCount
    Next
Next

```

4. Projekte / Projects

4.1. UniqueID

In SchüCal ist es möglich, auf Projekte über deren UniqueID zuzugreifen. Außerdem kann jede in diesem Dokument aufgeführte Funktion, die als Parameter die JobID des Projekts erwartet, sowohl mit der JobID als auch mit der UniqueID aufgerufen werden.

In SchüCal, it is possible to access projects by their UniqueID. Also, each function mentioned in this document that expects the JobID of the project as a parameter, can be called both with the JobID or the UniqueID.

Die UniqueIDs der Projekte können in SchüCal wie folgt abgefragt werden:

The UniqueIDs of projects can be retrieved from SchüCal as follows:

```

s_oApp = New Schuecal2000TLB.Schuecal2000App

Dim oJobDatas As Schuecal2000TLB.JimJobDatas
oJobDatas = s_oApp.JobMaster.JobData
Dim lJobDatas As Long
lJobDatas = oJobDatas.Count

For lJobDatas = 0 To lJobDatas - 1
    Dim oJobData As Schuecal2000TLB.JimJobData
    oJobData = oJobDatas.Item(lJobDatas)

```

```
Dim strUniqueID As String
strUniqueID = oJobData.UniqueID
Next
```

Oder wenn das Objekt des Projektes schon zur Hand ist:

Or in case that the job object is already present:

```
s_oApp = New Schuecal2000TLB.Schuecal2000App
Dim oJob As Job7
oJob = ...
oJob.UniqueID
```

4.2. Ein Projekt sperren / Locking a project

```
'Projekt-ID / JobID
Dim strJobID As String

'Projekt anlegen / Create project
Dim oJob2 As Job4
oJob2 = ...
'Projekt sperren / lock project
oJob2.LockJob(strJobID)
```

4.3. Ein Projekt entsperren / Unlocking a project

```
'Projekt-ID / Project-ID
Dim strJobID As String

'Projekt anlegen / Create project
Dim oJob2 As Job4
Job2 = ...
'Projekt entsperren / unlock project
oJob2.UnlockJob(strJobID)
```

4.4. Angehängte Dokumente abfragen / Query attached documents

In SchüCal besteht die Möglichkeit, einem Projekt Dateien anzuhängen. Diese werden über COM wie folgt ermittelt:

In SchüCal, it is possible to attach files to a project. These files are determined by COM as follows:

```
Dim oJobs As Jobs
Dim oJob As Job4

s_oApp = New Schuecal2000TLB.Schuecal2000App

'Container auf alle Objekte ablegen / Store container on all projects
oJobs = s_oApp.JobMaster.Jobs
'Es wird festgelegt für welches Projekt die Dokumente ermittelt werden
sollen / Set the project for which documents are to be determined
oJob = s_oApp.JobMaster.Jobs.GetNewJob(strJob)
'Es wird die Klasse der Anhänge initialisiert / Class of attachments is
initialised
Dim strErrorMessage As String
Dim oJobAttachments As Schuecal2000TLB.JimJobAttachments
oJobAttachments = oJob.AttachmentsOfJob(oJob.JobID, True, strErrorMessage,
True)
```



```
'Anzahl der angehängten Dokumente / Number of attached documents
Dim lAttCount As Long
lAtt = oJobAttachments.Count

Dim lRun As Long
For lRun = 0 To lAtt - 1
    'jedes einzelne angehängte Dokument / each attached document
    Dim oAttachment As Schuecal2000TLB.JimJobAttachment
    oAttachment = oJobAttachments.Item(lRun)
    'Dateiname / filename
    Dim strFilename As String
    strFilename = oAttachment.FileName
    'Titel / title
    Dim strFiletitle As String
    strFiletitle = oAttachment.FileTitle
    'Pfad / path
    Dim strPath As String
    strPath = oAttachment.Path
    'Flag, das ermittelt ob das Dokument versteckt ist / Flag that
    determines if the document is hidden
    Dim bIsHidden As Boolean
    bIsHidden = oAttachment.IsHidden
    'Dateityp / file type
    Dim FileType As Schuecal2000TLB.EAttachmentType
    FileType = oAttachment.Type
    'Folgende Dateitypen sind in dem enum EAttachmentType enthalten:
    / The following file types are included in the enum EAttachmentType:
    ' eFile (0) = simple file
    ' eDOP (1) = declaration of product
    ' eCE (2) = CE-marking
    ' eOfferDocument (3) = document from offermanagement
Next
```

4.5. Klassifizierung nach EN 1090 / Classification in accordance with EN 1090

Bestellungen für Projekte können ab R2016 bei Bedarf gemäß Ausführungsklasse EXC3 (EN1090) erfolgen. In der Projekt-Eigenschaftsliste können Sie unter der Gruppe 'Klassifizierung nach EN 1090' den Eintrag 'EXC3' wählen. Dieser kann über COM wie folgt gesetzt und abgefragt werden:

As of version R2016, order for projects can be carried out in accordance with Execution Class EXC3 (EN1090) if required. In the project properties list, you can select the option "EXC3" in the group "Classification in accordance with EN 1090". These options are determined and set by COM as follows:

```
Dim lConstructionTyp As Schuecal2000TLB.EEN1090ConstructionType
'To get the classification
lConstructionTyp = oJob.EN1090ConstructionType
'To set the classification
oJob.EN1090ConstructionType = Schuecal2000TLB.EEN1090ConstructionType.eEXC3
oJob.Save()
, wobei Schuecal2000TLB.EEN1090ConstructionType ein enum für folgende Werte
ist:
eNoneConstType, ohne Klassifizierung / without classification
eEXC1, nach EXC1, / in EXC1
eEXC2, nach EXC2, / in EXC2
eEXC3, nach EXC3 / in EXC3
```

5. Positionen / Items

5.1. Eine Position sperren / Locking an item

```
'Projekt-ID / JobID
Dim strJobID As String
'Position-ID / Item-ID
Dim strItemID As String

'Position anlegen / Create item
Dim oItem3 As Item3
Set oItem3 = ...

'Position sperren / lock item
oItem3.LockItem(strJobID, strItemID)
```

5.2. Eine Position entsperren / Unlocking an item

```
'Projekt-ID / Project-ID
Dim strJobID As String
'Position-ID / Item-ID
Dim strItemID As String

'Position anlegen / Create item
Dim oItem3 As Item3
Set oItem3 = ...

'Position entsperren / Unlock item
oItem3.UnlockItem(strJobID, strItemID)
```

5.3. Dateianhänge einer Position ermitteln / Determine attachments of an item

In SchüCal ist es möglich, über die neue Schnittstelle "Item5" die Dateianhänge einer Position zu ermitteln. Folgende Vorgehensweise:

/ In SchüCal, it is possible to determine the attachments of an item by the means of the new "Item5" interface. This is how it works:

```
'Projekt-ID / JobID
Dim strJobID As String
'Position-ID / ItemID
Dim strItemID As String
Dim oJob2 As Job2
oJob2 = s_oApp.JobMaster.Jobs.GetNewJob(strJobID)

'Position / Item
Dim oItem3 As Schuecal2000TLB.Item5
oItem3 = oJob2.Items.GetNewItem(strItemID)

'Klasse der Dateianhänge / Class of attachments
Dim oItemAttachments As Schuecal2000TLB.JimAttachments

oItemAttachments = oItem3.Attachments
Dim lItemAtt As Long
lItemAtt = oItemAttachments.Count
Dim lRun1 As Long
For lRun1 = 0 To lItemAtt - 1
    Dim oAttachment As Schuecal2000TLB.JimAttachment
    oAttachment = oItemAttachments.Item(lRun1)
    Dim strFilename As String
    strFilename = oAttachment.FileName
    Dim strFiletitle As String
    strFiletitle = oAttachment.FileTitle
    Dim strPath As String
    strPath = oAttachment.Path
    Dim strFileText As String
    strFileText = oAttachment.FileExtension
    Dim FileType As Schuecal2000TLB.EItemAttachmentType
    FileType = oAttachment.Type
Next
```

5.4. Positions-Varianten / Item Variants

In SchüCal ist es möglich, ab Release 2016R2, über die neue Schnittstelle Item5 zu ermitteln, ob es sich bei der Position um eine Variante handelt und zu welcher Basis diese Variante gehört. Folgende Vorgehensweise:

/ In SchüCal it is possible (Release 2016R2 and later) to determine if the item is a variant and to which basis this variant belongs. The new interface "Item5" makes this procedure possible and it works like this:

```
Dim oItem As Item5
...
'Es wird ermittelt, ob es sich bei der Position um eine Variante handelt
/Determine if the item is a variant
Dim bIsVariant As Boolean
bIsVariant = oItem.IsVariant
```

```
'Die UniqueID der Position wird ermittelt. Diese wird benötigt um die
Verbindung Variante zu Basis herzustellen
/Determine the UniqueID. This is important, to connect the variant with the
base
Dim strUniqueID As String
strUniqueID = oItem.UniqueID

'UniqueID der Basis Position
/UniqueId of the base
Dim strUniqueIDofBase As String
strUniqueIDofBase = oItem.UniqueIDofBase
```

5.5. Positionsskizze exportieren / Exporting item drawing

```
Dim oItem As Item4
...
Dim strSketch as string
strSketch = oItem.ItemSketch
```

Die Skizze wird durch obigen Aufruf im Temp-Ordner hinterlegt. Zu beachten ist, dass der Temp-Ordner nach dem Beenden von SchüCal gelöscht wird.

By the means of this function call the sketch will be saved in the temp folder. Please note that the temp folder is deleted after you quit SchüCal.

5.6. Angebotstext einer Position lesen / Reading quotation text of an item

```
s_oApp = New Schuecal2000TLB.Schuecal2000App
Projektbezeichnung / Project name
Dim strJob As String

'Positionsbezeichnung / Item name
Dim strItem As String

'Flag für die Optimierung / Flag for the optimisation
Dim bOptimized As Boolean
bOptimized = True
'Benutzerrechte ermitteln / Determine user rights
Dim oJob As Job
Dim oBatch As JimVirtualItems
'Factory ist eine spezielle Klasse, von der Klassen mit statischen Methoden
'abgeleitet werden können / Factory is a special class from which classes
with statistical methods can be derived
Dim oFactory As Object
oFactory = s_oApp.Factory
Dim oItem As Item
oJob = s_oApp.JobMaster.Jobs.GetNewJob(strJob)
oItem = oJob.Items.GetNewItem(strItem)
Dim matpot As MatpotNode 'Materialtopf / matpot
'Stückzahl ermitteln / Determining number of units
Dim lCount As Long
lCount = oItem.ItemCount
'Fertigungslos erzeugen / Create production batch
oBatch = oFactory.GetClassObject("JimVirtualItems")
'Fertigungslos füllen / Fill production batch
oBatch.AddItem(strJob, strItem, lCount, False)
'Ermittlungsmodul anlegen / Creating ascertainment module
```

```
Dim oAscertainment As TopAscertainment
oAscertainment = s_oApp.TopAscertainment
'Den Container auf alle Materialdaten anlegen und alle Materialdaten für
'diese Position erzeugen / Create container on all material data and create
all material data for this item.
matpot = s_oApp.TopAscertainment.GetMaterialOfProductionBatch(oBatch)

'Angebotstext wird ermittelt: Der erste Parameter ist die Position und der
'zweite Parameter ist der Container auf alle Materialdaten der Position
'(Materialtopf)
/ Determine quotation text: the first parameter is the item and the second
is the materialpot
Dim strOfferText As String
strOfferText = oAscertainment.GetOfferText(oItem, matpot)
```

6. Virtuelle Fertigungslose / virtual production batches

6.1. Fertigungsloserzeugung / Creating production batches

```
//In SchüCal gibt es die Möglichkeit, Fertigungslose zu generieren. Durch
//Fertigungslose können mehrere Projekte bzw. Positionen zusammengefasst
//werden, um einen gemeinsamen Materialtopf zu erzeugen und Ausgaben zu
//generieren.
In SchüCal, there exist several possibilities to generate production
batches. By generating production batches, several projects/items can be
combined to form and create one collective materialpot and to generate
output.

'Container für die virtuellen Positionen anlegen und mit den Positionen
'initialisieren lassen / Create containers for the virtual items and
intialise these with the items
'Da die Methode Fertigungslose (Positionsgruppen) behandelt, ist es
möglich, 'den Container auch mit mehreren Positionen zu initialisieren
Since the method deals with productions batches (item groups), it is also
possible to initialise the container with several items.
Dim oBatch As JimVirtualItems
'Factory ist eine spezielle Klasse, von der Klassen mit statischen Methoden
'abgeleitet werden können / Factory is a special class from which classes
with statistical methods can be derived
Dim oFactory As Object
Set oFactory = SApp.Factory

'Neues Fertigungslos erzeugen / Creating new production batch
Set oBatch = oFactory.GetClassObject("JimVirtualItems")
'Fertigungslos füllen / Fill production batch
oBatch.AddItem strJob, strItem, lCount, False
```

6.2. Materialtopf über ein Fertigungslos erzeugen / Creating a materialpot by the means of a production batch

Hier wird demonstriert, wie der Materialtopf von Fertigungslosen ermittelt wird. Nachdem der Materialtopf vorhanden ist, können folgende Informationen ermittelt werden:

This demonstrates how the material list of production batches (ProductionBatchItems) is determined. When the material list is available, then the following information can be determined:

- [Artikel­daten ermitteln](#)
Determine article data
- [Glas- und Paneel­daten einer Position](#)
Glass and panel data of an item
- [Glas-, und Paneel­daten \(Einzel­mengen\) einer Position \(Ermit­tlung der Feld­info\)](#)
Glass and panel data (single volume) of a item (determing field info)

7. Artikel­daten ermitteln / Determine article data

7.1. Allgemeine Artikel­daten / General article data

```
s_oApp = New Schuecal2000TLB.Schuecal2000App
'Projektbezeichnung / Project name
Dim strJob As String
'Positionsbezeichnung / Item name
Dim strItem As String
'Flag für die Optimierung / Flag for the optimisation
Dim bOptimized As Boolean
bOptimized = True
'Benutzerrechte ermitteln / Determine userrights
Dim oJob As Job
Dim oBatch As JimVirtualItems
'Factory ist eine spezielle Klasse, von der Klassen mit statischen Methoden
'abgeleitet werden können / Factory is a special class from which classes
with statistical methods can be derived
Dim oFactory As Object
oFactory = s_oApp.Factory
Dim oItem As Item

oJob = s_oApp.JobMaster.Jobs.GetNewJob(strJob)
oItem = oJob.Items.GetNewItem(strItem)
Dim matpot As MatpotNode 'Materialtopf / materialpot
'Stückzahl ermitteln / Determining number of units
Dim lCount As Long
lCount = oItem.ItemCount
'Fertigungslos erzeugen / Create production batch
oBatch = oFactory.GetClassObject("JimVirtualItems")
'Fertigungslos füllen / Fill production batch
oBatch.AddItem(strJob, strItem, lCount, False)
'Ermittlung-modul anlagen / Creating ascertainment module
Dim oAscertainment As TopAscertainment
oAscertainment = s_oApp.TopAscertainment
'Den Container auf alle Materialdaten anlegen und alle Materialdaten für
'diese Position erzeugen / Create container on all material data and create
all material data for this item
matpot = s_oApp.TopAscertainment.GetMaterialOfProductionBatch(oBatch)
'Modul zur Erstellung der Ausgabedaten anlegen / Create module for the
creation of output data
Dim oGenerate As OutpGenerateBulklist
oGenerate = s_oApp.GenerateBulklist
'Die Preise immer ermitteln / Always determine the prices
oGenerate.SetPrintPriceCOM(True)

'Generieren der Ausgabedaten. Erster Parameter ist der Container auf alle
'Materialdaten
```

```
'(Materialtopf), der zweite und der dritte Parameter sind Flags, die
'aussagen, ob die Ermittlung mit oder ohne Optimierung durchgeführt werden
'soll.
/ Generate output data. First parameter is the container for all material
data (material pot), the second and the third parameters are flags that
specify whether the determination shall be carried out with or without
optimisation
oGenerate.GenerateOptimizedBulklist(matpot, bOptimized, True)
'Die generierten Ausgabedaten werden als ein Container (OutpBulklistResult)
'zurückgegeben
/ The output data is returned as container(OutpBulklistResult)
Dim oResult As OutpBulklistResult5
oResult = oGenerate.Result
'Die Ausgabedaten geben den Container der Artikeldaten (OutpBulklistPrices)
zurück. Der Container beinhaltet die Datensätze, die nach Artikelnummer,
Oberfläche und Farbe gruppiert sind.
/ The output data return the container of the article data
(OutpBulklistPrices). This contains the records that are grouped according
to item number, surface and colour.
Dim oPrices As OutpBulklistPrices
oPrices = oResult.COMEntries
'Anzahl der Datensätze wird ermittelt / Number of data sets is determined
Dim lPrices As Long
lPrices = oPrices.Count
'Schleife durch alle Datensätze / Loop through all data records
Dim lRun As Long
For lRun = 0 To lPrices - 1
    Dim oPrice As OutpBulklistPrice5 (*ab SchueCal2015)
    oPrice = oPrices.Item(lRun)
    'Menge in Stück oder Meter / Quantity in piece or meter
    Dim dAmount As Double
    dAmount = oPrice.Amount

    'Ermittlung, ob es sich bei dem Artikel um einen EVT-Artikel handelt
    / Determining whether the article is an EVT-Article
    Dim lInstallationType As Long
    Dim bIsEVT As Boolean
    lInstallationType = oPrice.InstallationTypeEVT
    If(lInstallationType = 1)
        bIsEVT = True 'Gangflügel
    ElseIf(lInstallationType = 2)
        bIsEVT = True 'Standflügel
    Else
        bIsEVT = False

    'umlaufende Oberfläche [m] / Circumfering surface [m]
    Dim dAreaSurface As Double
    dAreaSurface = oPrice.AreaSurface
    'Artikelnummer / Article number
    Dim strArticleNumber As String
    strArticleNumber = oPrice.ArticleNumber
    'Bezeichnung / Name
    Dim strDescription As String
    strDescription = oPrice.Description
    'EDV-Code / DP-Code
    Dim dEDVCode As Double
    dEDVCode = oPrice.EdvCode
    'Farbe innen (bei getrennten Halbschalen) / Colour inside
    Dim strInnerColour As String
    strInnerColour = oPrice.InnerColour
```

```

'Oberfläche innen / Surface inside
Dim binsidesurface As Boolean
binsidesurface = oPrice.HasInsideSurface
'Artikel in EXC3-Ausführung, falls lExc3NormID > 0
/ if lExc3NormID > 0, then is the article an in EXC3 classification
Dim lExc3NormID As Long
lExc3NormID = oPrice.Exc3NormID
Dim lInnerSurface As Long
lInnerSurface = -1
If (binsidesurface = True) Then
    lInnerSurface = oPrice.InnerSurface
    'Flag, das bei der Optimierung gesetzt wird / Flag which is set
    during optimisation
    Dim bOptimizeFlag As Boolean
    bOptimizeFlag = oPrice.OptimizeFlag
    'Farbe aussen / Colour outside
    Dim strOuterColour As String
    strOuterColour = oPrice.OuterColour
    'Oberfläche aussen / Surface outside
    Dim lOuterSurface As Long
    lOuterSurface = -1
    Dim boutsidesurface As Boolean
    boutsidesurface = oPrice.HasOutsideSurface
    If (boutsidesurface = True) Then
        lOuterSurface = oPrice.OuterSurface
    End If
    'Farbe mitte bei 3-schaligen Profilen / Colour middleside
    Dim strMiddleColour As String
    strMiddleColour = oPrice.MiddleColour
    'Oberfläche mitte / Surface middleside
    Dim lmiddleSurface as Long
    lmiddleSurface = oPrice.MiddleSurface

    'Polierte Oberfläche / Surface polished
    Dim dPolishSurface As Double
    dPolishSurface = oPrice.PolishSurface
'Preis für die benötigte Menge / Price for the required quantity
Dim dPrice As Double
dPrice = oPrice.Price
'tatsächlich benötigte Menge der Stangen / actually required quantity
of bars
Dim dOffer As Double
dOffer = oPrice.Supply
'Einheit (Stangen, VE, Stück) / Unit (bars, PU, piece)
Dim lOfferType As Long
lOfferType = oPrice.SupplyType
'Anwendungsfall / Usecase
Dim lUseCase As Long
lUseCase = oPrice.UseCase
Dim bIsDCSDecor As Boolean
If (lUseCase = Schuecal2000TLB.EUseCase.eDCSDecor) Then
    bIsDCSDecor = True
End If
'Gewicht [kg] / Weight [kg]
Dim dWeight As Double
dWeight = oPrice.Weight
'Flag, das bei Sonderlängen gesetzt wird / Flag which is set
for special lengths
Dim bFixLengthFlag As Boolean
bFixLengthFlag = oPrice.FixLengthFlag
'Länge nur bei Sonderlängen [mm] / Length only in case of

```



```

        extraordinary lengths
        Dim dFixLength As Double
        dFixLength = oPrice.FixLength
        'Flag, das bei Isolierstegen gesetzt wird / Flag, which is
        set for insulating bars
        Dim bPolyamidFlag As Boolean
        bPolyamidFlag = oPrice.PolyamidFlag
        'Flag, das bei Überlängen gesetzt wird / Flag, which is set
        for excess lengths
        Dim bOverLengthFlag As Boolean
        bOverLengthFlag = oPrice.OverLengthFlag
        'Anzahl der Artikel beim zusätzlichen Material / Amount of
        additional material articles
        Dim sCount As Single
        sCount = oPrice.Count
        'Flag, das bei Lagerartikeln gesetzt wird / Flag indicating
        stock
        Dim bStockArticle As Boolean
        bStockArticle = oPrice.StockArticle
    End If
Next

```

7.2. CTB (Concealed Toughened Blind)

```

'Die Ausgabedaten geben den Container der Artikeldaten
'(OutpBulklstCTBDatas)zurück. Der Container beinhaltet die Datensätze.
The output data return the container of the article data
(OutpBulklstCTBDatas). The container contains the data sets.
Dim oCTBDatas As OutpBulklstCTBDatas
Set oCTBDatas = oResult.COMCTBs

'Anzahl der Datensätze der CTBs wird ermittelt / Number of data records of
CTB data is determined
Dim lCTBDatas As Long
lCTBDatas = oCTBDatas.Count

'Durch alle Datensätze / Through all data sets
Dim lRun As Long
For lRun = 0 To lCTBDatas - 1
    Dim oCTBData As OutpBulklstCTBData
    Set oCTBData = oCTBDatas.Item(lRun)
    'ID des CTB / ID of CTB
    Dim lCTBID As Long
    Set lCTBID = oCTBData.CTBID
    'Artikelnummer des CTB / Article number of CTB
    Dim strCTBArticle As String
    Set strCTBArticle = oCTBData.CTBArticle
    'Bestellhöhe HB / Order Height HB
    Dim dOrderHeightHB As Double
    Set dOrderHeightHB = oCTBData.OrderHeightHB
    'Bestellbreite BB / Order Width BB
    Dim dOrderWidthBB As Double
    Set dOrderWidthBB = oCTBData.OrderWidthBB
    'Bestellbreite BRa / Order Width BRa
    Dim dOrderWidthBRa As Double
    Set dOrderWidthBRa = oCTBData.OrderWidthBRa
    'Bestellbreite BB1 / Order Width BB1
    Dim dOrderWidthBB1 As Double
    Set dOrderWidthBB1 = oCTBData.OrderWidthBB1
    'Bestellbreite BB2 / Order Width BB2

```

```

Dim dOrderWidthBB2 As Double
Set dOrderWidthBB2 = oCTBData.OrderWidthBB2
'Oberfläche BB / surface BB
Dim lSurfaceBB As Long
Set lSurfaceBB = oCTBData.SurfaceBB
'Farbton BB / Colour BB
Dim strColourBB As String
Set strColourBB = oCTBData.ColourBB
'Oberfläche BB2 / Surface BB2
Dim lSurfaceBB2 As Long
Set lSurfaceBB2 = oCTBData.SurfaceBB2
'Farbton BB2 / Colour BB2
Dim strColourBB2 As String
Set strColourBB2 = oCTBData.ColourBB2
'Anzahl des CTB / Count of CTB
Dim lCount As Long
Set lCount = oCTBData.Count
'Projekt-ID / Object ID
Dim strJobID As String
Set strJobID = oCTBData.JobID
'Positions-ID / Item ID
Dim strItemID As String
Set strItemID = oCTBData.ItemID
'Preis des CTB / Price of cTB
Dim dPrice As Double
Set dPrice = oCTBData.Price
'Positionsnummer für Bestellung / Item number for order
Dim lItemNumber As Long
Set lItemNumber = oCTBData.ItemNumber
'Sonnenschutz-ID / sun shading ID
Dim lSunShadingID As Long
Set lSunShadingID = oCTBData.SunShadingID

```

Next

7.3. CCB (Concealed Composite Blind)

```

s_oApp = New Schuecal2000TLB.Schuecal2000App
'Objektbezeichnung / Project name
Dim strJob As String
strJob = "COMTest"
'Positionsbezeichnung / Item name
Dim strItem As String
strItem = "01"
'Flag für die Optimierung / Flag for the optimisation
Dim bOptimized As Boolean
bOptimized = True
'Benutzerrechte ermitteln / Determine userrights

Dim oJob As Job
Dim oBatch As JimVirtualItems
'Factory ist eine spezielle Klasse, von der Klassen mit statischen Methoden
'abgeleitet werden können / Factory is a special class from which classes
'with statistical methods can be derived

Dim oFactory As Object
oFactory = s_oApp.Factory
Dim oItem As Item

oJob = s_oApp.JobMaster.Jobs.GetNewJob("COMTest")
oItem = oJob.Items.GetNewItem("01")

```

```

Dim matpot As MatpotNode 'Materialtopf

'Anzahl der Elemente ermitteln / Determining number of units
Dim lCount As Long
lCount = oItem.ItemCount

'Fertigungslos erzeugen / Create production batch
oBatch = oFactory.GetClassObject("JimVirtualItems")
'Fertigungslos füllen / Fill production batch
oBatch.AddItem(strJob, strItem, lCount, False)

'Ermittlung-modul anlagen / Creating ascertainment module
Dim oAscertainment As TopAscertainment
oAscertainment = s_oApp.TopAscertainment

'Den Container auf alle Materialdaten anlegen und alle Materialdaten für
'diese Position erzeugen / Create container on all material data and create
all material data for this item.
matpot = s_oApp.TopAscertainment.GetMaterialOfProductionBatch(oBatch)

'Modul zur Erstellung der Ausgabedaten anlegen / Create module for the
creation of output data
Dim oGenerate As OutpGenerateBulklist
oGenerate = s_oApp.GenerateBulklist

'Die Preise immer ermitteln / Always determine the prices
oGenerate.SetPrintPriceCOM(True)
oGenerate.GenerateOptimizedBulklist(matpot, bOptimized, True)
Dim oResult As OutpBulklistResult5
oResult = oGenerate.Result

Dim oCTBDatas As Schuecal2000TLB.OutpBulklistCTBDatas
oCTBDatas = oResult.COMCCBEntries

'Anzahl der Datensätze der CTBs wird ermittelt / Number of data records of
CTB data are determined
Dim lCTBDatas As Long
lCTBDatas = oCTBDatas.Count

'Durch alle Datensätze / Through all data sets
Dim lRun As Long
For lRun = 0 To lCTBDatas - 1
    Dim oCTBData As OutpBulklistCTBData
    oCTBData = oCTBDatas.Item(lRun)
    'ID des CTB / ID of CTB
    Dim lCTBID As Long
    lCTBID = oCTBData.CTBID
    'Artikelnummer des CTB / Article number of CTB
    Dim strCTBArticle As String
    strCTBArticle = oCTBData.CTBArticle
    'Bestellhöhe HB / Order Height HB
    Dim dOrderHeightHB As Double
    dOrderHeightHB = oCTBData.OrderHeightHB
    'Bestellbreite BB / Order Width BB
    Dim dOrderWidthBB As Double
    dOrderWidthBB = oCTBData.OrderWidthBB
    'Farbton BB / Colour BB
    Dim strColourBB As String
    strColourBB = oCTBData.ColourBB
Next

```

7.4. VarioAir

```

'Objektbezeichnung / Project name
Dim strJob As String
'Positionsbezeichnung / Item name
Dim strItem As String
'Flag für die Optimierung / Flag for the optimisation
Dim bOptimized As Boolean
bOptimized = True
'Benutzerrechte ermitteln / Determine userrights

Dim oJob As Job
Dim oBatch As JimVirtualItems
'Factory ist eine spezielle Klasse, von der Klassen mit statischen Methoden
'abgeleitet werden können / Factory is a special class from which classes
with statistical methods can be derived

Dim oFactory As Object
oFactory = s_oApp.Factory
Dim oItem As Item

oJob = s_oApp.JobMaster.Jobs.GetNewJob("COMTest")
oItem = oJob.Items.GetNewItem("01")

Dim matpot As MatpotNode 'Materialtopf

'Anzahl der Elemente ermitteln / Determining number of units
Dim lCount As Long
lCount = oItem.ItemCount

'Fertigungslos erzeugen / Create production batch
oBatch = oFactory.GetClassObject("JimVirtualItems")
'Fertigungslos füllen / Fill production batch
oBatch.AddItem(strJob, strItem, lCount, False)

'Ermittlung-modul anlagen / Creating ascertainment module
Dim oAscertainment As TopAscertainment
oAscertainment = s_oApp.TopAscertainment

'Den Container auf alle Materialdaten anlegen und alle Materialdaten für
'diese Position erzeugen / Create container on all material data and create
all material data for this item.

matpot = s_oApp.TopAscertainment.GetMaterialOfProductionBatch(oBatch)

'Modul zur Erstellung der Ausgabedaten anlegen / Create module for the
creation of output data

Dim oGenerate As OutpGenerateBulklist
oGenerate = s_oApp.GenerateBulklist

'Die Preise immer ermitteln / Always determine the prices
oGenerate.SetPrintPriceCOM(True)

oGenerate.GenerateOptimizedBulklist(matpot, bOptimized, True)

Dim oResult As OutpBulklistResult5
oResult = oGenerate.Result

Dim oVarioAirDatas As Schuecal2000TLB.OutpBulklistVarioAirDatas
oVarioAirDatas = oResult.COMVarioAirs

```

```
'Anzahl der Datensätze der VarioAirs wird ermittelt / Number of data
records of VarioAir data are determined

Dim lVarioAirDatas As Long
lVarioAirDatas = oVarioAirDatas.Count

'Durch alle Datensätze / By all data sets
Dim lRun As Long
For lRun = 0 To lVarioAirDatas - 1
    Dim oVarioAirData As Schuecal2000TLB.OutpBulklistVarioAirData3
    oVarioAirData = oVarioAirDatas.Item(lRun)
    'Artikelnummer /Article number
    Dim strArt As String
    strArt = oVarioAirData.VarioAirArticle
    'Oberfläche Innen /Inner surface
    dim lSurfIn as Long
    lSurfIn = oVarioAirData.SurfaceInside
    'Oberfläche außen /Outer surface
    Dim lSurfOut As Long
    lSurfOut = oVarioAirData.SurfaceOutside
    'Farbe Innen /Inner color
    Dim strColourIn As String
    strColourIn = oVarioAirData.ColourInside
    'Farbe außen /Outer color
    Dim strColourOut As String
    strColourOut = oVarioAirData.ColourOutside
    'Wetterschenkel /Weather bar
    Dim lWeather as Long
    lWeather = oVarioAirData.WeatherbarType
    'Endkappen-Farbe, 0:Schwarz, 1: Weiß
    /Colour of end cap, 0:Black, 1:White
    dim lEndCap as Long
    lEndCap = oVarioAirData.EndCapColour
Next
```

7.5. VentoTherm Twist

```
'Objektbezeichnung / Project name
Dim strJob As String
'Positionsbezeichnung / Item name
Dim strItem As String
'Flag für die Optimierung / Flag for the optimisation
Dim bOptimized As Boolean
bOptimized = True
'Benutzerrechte ermitteln / Determine userrights

Dim oJob As Job
Dim oBatcch As JimVirtualItems
'Factory ist eine spezielle Klasse, von der Klassen mit statischen Methoden
'abgeleitet werden können / Factory is a special class from which classes
with statistical methods can be derived

Dim oFactory As Object
oFactory =s_oApp.Factory
Dim oItem As Item

oJob = s_oApp.JobMaster.Jobs.GetNewJob("COMTest")
oItem = oJob.Items.GetNewItem("ventothermtwist")
```

```

Dim matpot As MatpotNode 'Materialtopf

'Anzahl der Elemente ermitteln / Determining number of units
Dim lCount As Long
lCount = oItem.ItemCount

'Fertigungslos erzeugen / Create production batch
oBatch = oFactory.GetClassObject("JimVirtualItems")
'Fertigungslos füllen / Fill production batch
oBatch.AddItem(strJob, strItem, lCount, False)

'Ermittlung-modul anlagen / Creating ascertainment module
Dim oAscertainment As TopAscertainment
oAscertainment = s_oApp.TopAscertainment

'Den Container auf alle Materialdaten anlegen und alle Materialdaten für
'diese Position erzeugen / Create container on all material data and create
all material data for this item.

matpot = s_oApp.TopAscertainment.GetMaterialOfProductionBatch(oBatch)

'Modul zur Erstellung der Ausgabedaten anlegen / Create module for the
creation of output data

Dim oGenerate As OutpGenerateBulklist
oGenerate = s_oApp.GenerateBulklist

'Die Preise immer ermitteln / Always determine the prices
oGenerate.SetPrintPriceCOM(True)

oGenerate.GenerateOptimizedBulklist(matpot, bOptimized, True)

Dim oResult As OutpBulklistResult7
oResult = oGenerate.Result

Dim oVarioAirDatas As Schuecal2000TLB.OutpBulklistVentoThermTwistDatas
oVentoThermTwistDatas = oResult.COMVentoThermTwistEntries

'Anzahl der Datensätze der VentoThermTwist wird ermittelt / Number of data
records of VentoThermTwist data are determined

Dim lVentoThermTwistDatas As Long
lVentoThermTwistDatas = oVentoThermTwistDatas.Count

'Durch alle Datensätze / By all data sets
Dim lRun As Long
For lRun = 0 To lVentoThermTwistDatas - 1
    Dim oData As OutpBulklistVentothermTwistData
    'Artikelnummer / article number
    Dim strArticleNr As String
    strArticleNr = oData.ArticleNr
    'Projektname / job name
    Dim strJobFileID As String
    strJobFileID = oData.JobFileID
    'Positionsname / item name
    Dim strItemFileID As String
    strItemFileID = oData.ItemFileID
    'Lüfterlänge / vent length
    Dim dLength As Double
    dLength = oData.Length
    'Anzahl / count

```

```

Dim lCount As Long
lCount = oData.Count
'Kabelausgang / position of cable exit
Dim lCablePosition As Long
lCablePosition = oData.CablePosition
'Sichtblende / screen
Dim lCoverPanelVariant As Long
lCoverPanelVariant = oData.CoverPanelVariant
'I/O Kabel / I/O cable
Dim lIOCable As Long
lIOCable = oData.IOCable
'Filter / filter
Dim lFilterVariant As Long
lFilterVariant = oData.FilterVariant
'Beschichtung mit Voranodisation / coating with pre anodization
Dim lSeaClimate As Long
lSeaClimate = oData.SeaClimate
'Farbe der Endkappe / colour of the end cap
Dim lEndCapColour As Long
lEndCapColour = oData.EndCapColour
'Oberfläche innen / inside surface
Dim lSurfaceInside As Long
lSurfaceInside = oData.SurfaceInside
'Oberfläche außen / outside surface
Dim lSurfaceOutside As Long
lSurfaceOutside = oData.SurfaceOutside
'Farbe innen / inside colour
Dim strColourInside As String
strColourInside = oData.ColourInside
'Farbe außen / outside colour
Dim strColourOutside As String
strColourOutside = oData.ColourOutside
'Glanzgrad innen / inside gloss level
Dim lInnerColourType As Long
lInnerColourType = oData.InnerColourType
'Glanzgrad außen / outside gloss level
Dim lOuterColourType As Long
lOuterColourType = oData.OuterColourType
'Bautiefe / installation depth
Dim dInstallationDepth As Double
dInstallationDepth = oData.InstallationDepth
'Preis / price
Dim dPrice As Double
dPrice = oData.Price

```

Next

7.6. ZDS

```
'Objektbezeichnung / Project name
Dim strJob As String
'Positionsbezeichnung / Item name
Dim strItem As String
'Flag für die Optimierung / Flag for the optimisation
Dim bOptimized As Boolean
bOptimized = True
'Benutzerrechte ermitteln / Determine userrights

Dim oJob As Job
Dim oBatch As JimVirtualItems
'Factory ist eine spezielle Klasse, von der Klassen mit statischen Methoden
'abgeleitet werden können / Factory is a special class from which classes
with statistical methods can be derived

Dim oFactory As Object
oFactory = s_oApp.Factory
Dim oItem As Item

oJob = s_oApp.JobMaster.Jobs.GetNewJob("COMTest")
oItem = oJob.Items.GetNewItem("zds")

Dim matpot As MatpotNode 'Materialtopf

'Anzahl der Elemente ermitteln / Determining number of units
Dim lCount As Long
lCount = oItem.ItemCount

'Fertigungslos erzeugen / Create production batch
oBatch = oFactory.GetClassObject("JimVirtualItems")
'Fertigungslos füllen / Fill production batch
oBatch.AddItem(strJob, strItem, lCount, False)

'Ermittlung-modul anlagen / Creating ascertainment module
Dim oAscertainment As TopAscertainment
oAscertainment = s_oApp.TopAscertainment

'Den Container auf alle Materialdaten anlegen und alle Materialdaten für
'diese Position erzeugen / Create container on all material data and create
all material data for this item.

matpot = s_oApp.TopAscertainment.GetMaterialOfProductionBatch(oBatch)

'Modul zur Erstellung der Ausgabedaten anlegen / Create module for the
creation of output data

Dim oGenerate As OutpGenerateBulklist
oGenerate = s_oApp.GenerateBulklist

'Die Preise immer ermitteln / Always determine the prices
oGenerate.SetPrintPriceCOM(True)

oGenerate.GenerateOptimizedBulklist(matpot, bOptimized, True)

Dim oResult As OutpBulklistResult8
oResult = oGenerate.Result

Dim oZSDSdatas As Schuecal2000TLB.OutpBulklistZDSEntries
oZSDSdatas = oResult.COMZDSEntries
```


'Anzahl der Datensätze der ZDS wird ermittelt / Number of data records of ZDS data are determined

```
Dim lZSDSdatas As Long
lZSDSdatas = oZSDSdatas.Count
```

'Durch alle Datensätze / By all data sets

```
Dim lRun As Long
For lRun = 0 To lZSDSdatas - 1
    Dim oData As OutpBulklistZDSData
    oData = oZSDSdatas.Item(lRun)
```

```
    Dim lZDSID As Long
    lZDSID = oData.GetZDSID();
```

' near to facade (3), far from facade (4)

```
    Dim lZDSType As Long
    lZDSType = oData.GetZDSType()
```

```
    ' Serge 600 (101)
    ' Serge 1 Percent (102)
    ' Santine 5500 (103)
    ' Soltis 86 (201)
    ' Soltis 92 (202)
    ' Soltis B92 (301)
    ' Orchestra (302)
    ' Vuscreen FR (303)
    ' Vuscreen ALU (304)
```

```
    Dim lTextileProperty As Long
    lTextileProperty = oData.GetTextileProperty()
```

```
    ' Flexible (1)
    ' Rigid (2)
    ' Special (3)
    ' Special Textile (4)
```

```
    Dim lTextileType As Long
    lTextileType = oData.GetTextileType()
```

```
    ' Serge600GreyCoal (101)
    ' Serge600CoalCoal (102)
    ' Serge600GreyGrey (103)
    ' Serge600PearlgreyPearlgrey (104)
    ' Serge600GreyWhite (105)
    ' Serge600GreyWhitepearlgrey (106)
    ' Serge600WhitePearlgrey (107)
    ' Serge600Oystershell (108)
    ' Serge600LinenWhite (109)
    ' Serge600Softclay (110)
    ' Serge600WhiteWhite (111)
    ' Serge600LinenPearlgrey (112)
    ' Serge600CoalBronze (113)
    ' Serge600GreySand (114)
    ' Serge600BronzeBronze (115)
    ' Serge600SandBronze (116)
    ' Serge600GreyAzureblue (117)
    ' Serge600LinenGreygold (118)
    ' Serge600WhiteWhitepearlgrey (119)
    ' Serge600SandWhite (120)
    ' Serge600PearlgreyWhite (121)
    ' Serge600LinenLinen (122)
```

```
' Serge600Wetsand (123)
' Serge600Sandstone (124)
' Serge600LinenBordeaux (125)
' Serge600PearlgreyAzureblue (126)
' Serge600Palm (127)
' Serge600Moss (128)
' Serge600Maroon (129)
' Serge600GreyYellowgreen (130)
' Santine5500Greycharcoal (201)
' Santine5500GreyTurquoise (202)
' Santine5500Pearls (203)
' Santine5500RAL9005 (204)
' SergelPercentCoalCoal (301)
' SergelPercentGreyGrey (302)
' SergelPercentPearlgreyPearlgrey (303)
' SergelPercentGreyCoal (304)
' SergelPercentWhitePearlgrey (305)
' Soltis86Anthrazite (401)
' Soltis86Pebble (402)
' Soltis86Bamboo (403)
' Soltis86Concrete (404)
' Soltis86Nutmeg (405)
' Soltis86Champagne (406)
' Soltis86Pepper (407)
' Soltis86Red (408)
' Soltis86Cocoa (409)
' Soltis86Caramel (410)
' Soltis86Midnightblue (411)
' Soltis86Buttercupyellow (412)
' Soltis92Nutmeg (501)
' Soltis92Pebble (502)
' Soltis92Anthrazite (503)
' Soltis92AluAlu (504)
' Soltis92Bronze (505)
' Soltis92Beton (506)
' Soltis92Hemp (507)
' Soltis92Champagne (508)
' Soltis92Cocoa (509)
' Soltis92Gold (510)
' Soltis92Navyblue (511)
' Soltis92Pinegreen (512)
' SoltisB92Pebble (601)
' SoltisB92Sandbeige (602)
' SoltisB92Black (603)
' OrchestraSlate (701)
' OrchestraMousegrey (702)
' OrchestraStone (703)
' VuscreenFRDarkgrey (801)
' VuscreenFRWhiteGrey (802)
' VuscreenFRWhiteBrightgrey (803)
' VuscreenALUDarkgrey (901)
' VuscreenALUBeige (902)
' VuscreenALUWhite (903)
Dim lTextileColour As Long
lTextileColour = oData.GetTextileColour()
Dim lTextileColourSide1 As Long
lTextileColourSide1 = oData.GetTextileColourSide1()
Dim lTextileColourSide2 As Long
lTextileColourSide2 = oData.GetTextileColourSide2()

' Beschichtungs ID / surface id
```

```

Dim lGuidebarSurface As Long
lGuidebarSurface = oData.GetGuidebarSurface()

' Farb Id / colour id
Dim strGuidebarColour As String
strGuidebarColour = oData.GetGuidebarColour()

' left (1), right (2)
Dim lMotorPosition As Long
lMotorPosition = oData.GetMotorPosition()

Dim strSpecialTextile As String
strSpecialTextile = oData.GetSpecialTextile()

Dim dOrderWidth As Double
dOrderWidth = oData.GetOrderWidth()

Dim dOrderHeight As Double
dOrderHeight = oData.GetOrderHeight()

Dim dHeightBetweenUnits As Double
dHeightBetweenUnits = oData.GetHeightBetweenUnits()

Dim dCurtainHeight As Double
dCurtainHeight = oData.GetCurtainHeight()

Dim dCurtainWidth As Double
dCurtainWidth = oData.GetCurtainWidth()

Dim dCurtainHeightOffset As Double
dCurtainHeightOffset = oData.GetCurtainHeightOffset()

Dim dStitchingPosition As Double
dStitchingPosition = oData.GetStitchingPosition()

Dim bVerticalStitching As Boolean
bVerticalStitching = oData.GetVerticalStitching()

Dim bHorizontalStitching As Boolean
bHorizontalStitching = oData.GetHorizontalStitching()

Dim strJobFileID As String
strJobFileID = oData.GetJobFileID()

Dim strItemFileID As String
strItemFileID = oData.GetItemFileID()

Dim lCount As Long
lCount = oData.GetCount()

Dim strArticleNr As String
strArticleNr = oData.GetArticleNr()

Dim dPrice As Double
dPrice = oData.GetPrice()

Dim dWeight As Double
dWeight = oData.GetWeight()
Next

```

7.7. Glas- und Paneeldaten einer Position / Glass and panel data of an item

```
'Projektbezeichnung / Project name
Dim strJob As String
'Positionsbezeichnung / Item name
Dim strItem As String
'Flag für die Optimierung / Flag for the optimisation
Dim bOptimized As Boolean

'Benutzerrechte ermitteln / Determine userrights

'Fertigungslos erzeugen / Creating production batch

'Material ermitteln / Calculate additional material

'Modul zur Erstellung der Glas-Ausgabedaten anlegen / Creating module for
the creation of glass output data
Dim oGlasslist As OutpGenerateGlasslist
Set oGlasslist = SApp. GenerateGlasslist
'Die Preise immer ermitteln / Always determine the prices
oGlasslist.SetPrintPrice( True )
'Generiert die Glas-Ausgabedaten. Parameter sind der Container der
'Materialdaten / Generates the glass output data. Parameters are the
containers of the material data.
oGlasslist.GenerateGlasslist(matpot)

'Die generierten Glas-Ausgabedaten werden als ein Container zurückgegeben /
The generated glass output data will be returned as a container
Dim oGlasslistResult As OutpGlasslistResult
Set oGlasslistResult = oGlasslist.Result

'Die generierten Glas-Ausgabedaten als ein COM-Container zurückgeben
'Der Container beinhaltet die einzelnen Glasdaten (OutpGlasslistPrice), die
gruppiert sind nach dem Glastyp, Breite, Höhe, Stärke und nach dem
'Projekt und Position, zu der sie gehören
Return the generated glass output data as a COM-container
The container contains the particular glass data (OutpGlasslistPrice),
'which are grouped by glass type, width, height, thickness and by the
'project and items they belongs to
Dim oGlasslistPrices As OutpGlasslistPrices
Set oGlasslistPrices = oGlasslistResult.COMEntries
'Anzahl wird ermittelt / Number is determined
Dim lGlasses As Long
lGlasses = oGlasslistPrices.Count
'Schleife durch alle Datensätze / Loop through all data records
Dim lRun As Long
For lRun = 0 To lGlasses - 1
    Dim oGlass As OutpGlasslistPrice4
    Set oGlass = oGlasslistPrices.Item( lRun )
    'Dieses Flag sagt aus, ob es ein Glas (False) oder Paneel (True) ist.
    This flag tells whether it is a glass or panel. True means that it
    'is panel and false means it is glass
    Dim bIsPaneel As Boolean
    bIsPanel = oGlass.IsPanel
    'Glas- oder Paneeltypbezeichnung / Glass or panel type name
    Dim strArticle As String
    strArticle = oGlass.Description
    'Anzahl der Scheiben/Paneele / Number of panes/panels
    Dim dAmount As Double
```

```

dAmount      = oGlass.Amount
'Breite der Scheibe/Paneel [cm] / Width of the pane/panel [cm]
Dim dMaxWidth As Double
dMaxWidth    = oGlass.MaxWidth
'Höhe der Scheibe/Paneel [cm] / Height of the pane/panel [cm]
Dim dMaxHeight As Double
dMaxHeight   = oGlass.MaxHeight
'Preis pro Scheibe oder Paneel / Price per pane or panel
Dim dGlassPrice As Double
dGlassPrice  = oGlass.GlassPrice
'Stärke / Thickness
Dim dThickness As Double
dThickness   = oGlass.Thickness
'Projekt zu dem die Scheibe oder das Paneel gehört wird ermittelt
Project to which the pane or the panel belongs will be determined.
Dim strActJobID As String
strActJobID  = oGlass.ActJobID
'Position zu der die Scheibe oder das Paneel gehört /
'Item to which the pane or the panel belongs
Dim strActItemID As String
strActItemID = oGlass.ActItemID
'Glastyp
Dim lGlasstype As long
lGlasstype   = oGlass.Glasstype
'Ermitteln ob Rechteckscheibe oder Modelltyp vorhanden ist
Dim bRectGlass As bool
bRectGlass   = oGlass.IsRectGlass
'Ermittlung des Modelltyps
Dim lModelType As long
If (bRectGlass=false) Then
    lModelType = oGlass.ModellType
End If
'Ermittlung, ob Monoglas oder Isolierglas
Dim bMonoGlass As bool
bMonoGlass   = oGlass.IsMonoGlass
'Nettopreis / net price
Dim dNettoPrice As Double
dNettoPrice  = oGlass.NettoPrice
'Energiezuschlag / energy charge
Dim dEnergyCharge As Double
dEnergyCharge = oGlass.EnergyCharge
'Mautzuschlag / road toll charge
Dim dTollCharge As Double
dTollCharge  = oGlass.TollCharge
'Artikelnummer /articlenumber
Dim strArt As String
strArt       = oGlass.ArticleNumber
'Kurzbezeichnung des Glases / short text of glass
Dim strShortText As String
strShortText = oGlass.ShortText

'Hersteller des Abstandhalters / producer of spacer
Dim strProducerOfSpacer As String
strProducerOfSpacer = oGlass.ProducerOfSpacer
'Werkstoff des Abstandhalters / material of spacer
Dim strMaterialOfSpacer as String
strMaterialOfSpacer = oGlass.MaterialOfSpacer
'Isolierglasssprossel / sashbar1
Dim strIsoSashbar1ArtNr As String
strIsoSashbar1ArtNr = oGlass.IsoSashbar1ArtNr
'Material der Isolierglassprosse 1 / material of sashbar1

```

```

Dim strIsoSashbar1Material As String
strIsoSashbar1Material = oGlass.IsoSashbar1Material
'Beschreibung der Isolierglasssprosse1 / description of sashbar1
Dim strIsoSashbar1Description As String
strIsoSashbar1Description = oGlass.IsoSashbar1Description
'Farbe der Isolierglasssprosse1 / color of sashbar1
Dim strIsoSashbar1Colour As String
strIsoSashbar1Colour = oGlass.IsoSashbar1Colour
'Isolierglasssprosse2...
Dim strIsoSashbar2ArtNr As String
strIsoSashbar2ArtNr = oGlass.IsoSashbar2ArtNr
'Material der Isolierglasssprosse2
Dim strIsoSashbar2Material As String
strIsoSashbar2Material = oGlass.IsoSashbar2Material
'Beschreibung der Isolierglasssprosse2
Dim strIsoSashbar2Description As String
strIsoSashbar2Description = oGlass.IsoSashbar2Description
'Farbe der Isolierglasssprosse2
dim strIsoSashbar2Colour as String
strIsoSashbar2Colour = oGlass.IsoSashbar2Colour

' Ermittlung für Ganzglasecken, Stufengläser und flügelüberdeckende
Türfüllungen
/ Determination of all-glass corners, stepped glazing and leaf-
enclosing door infills
'Größte Breite der außenliegenden Scheibe
/ Maximum width of the outer pane
Dim dMaxWidthSG As Double
dMaxWidthSG = oGlass.MaxWidthSG
'Größte Höhe der außenliegenden Scheibe
/ Maximum height of the outer pane
Dim dMaxHeightSG As Double
dMaxHeightSG = oGlass.MaxHeightSG
'Größte Breite der innenliegenden Scheibe
/ Maximum width of the inner pane
Dim dMaxWidthIP As Double
dMaxWidthIP = oGlass.MaxWidthInnerPanel
'Größte Höhe der innenliegenden Scheibe
/ Maximum height of the inner pane
Dim dMaxHeightIP As Double
dMaxHeightIP = oGlass.MaxHeightInnerPanel
'Abzugsmaß für die Bestellung
/ Deduction for the order
Dim dDiffMeasure As Double
dDiffMeasure = oGlass.DiffMeasure
' Hinweis zum Abzugsmaß
/ Note on deduction
dim strDeffmeasureText as String
strDeffmeasureText = oGlass.DiffMeasureText
' Hinweis zur Bestellung z.B. FWS 35PD
/ Note on the order (e.g. FWS 35PD)
Dim strordertext As String
strordertext = oGlass.OrderInformationText

```

Next

7.8. Glas- und Paneeldaten (Einzelmengen) einer Position (Ermittlung der Feldinfo) / Glass and panel data (individual quantity) of an item (determining field info)

```
Dim oFactory As Object
Dim oBatch As Schuecal2000TLB.IJimVirtualItems
Dim oAscertainment As Schuecal2000TLB.ITopAscertainment
Dim matpot As Schuecal2000TLB.IMatpotNode
Dim oGlasslist As Schuecal2000TLB.IOutpGenerateGlasslist
Dim oGlasslistResult As Schuecal2000TLB.IOutpGlasslistResult
Dim oGlasslistPrices As Schuecal2000TLB.IOutpGlasslistPrices
Dim lGlasses As Long

oFactory = s_oApp.Factory
oBatch = oFactory.GetClassObject("JimVirtualItems")
oBatch.AddItem(strJobID, strItemID, lItemCount, False)
oAscertainment = s_oApp.TopAscertainment
matpot = s_oApp.TopAscertainment.GetMaterialOfProductionBatch(oBatch)
'Modul zur Erstellung der Glas-Ausgabedaten anlegen / Creating module for
the creation of glass output data
oGlasslist = s_oApp.GenerateGlasslist
'Die Preise immer ermitteln / Always determine the prices
oGlasslist.SetPrintPrice(True)
'Durch diese Methode (GenerateGlasslistCOM) werden die Glas- und
'Paneeldatendaten unabhängig von den eingestellten Optionen in SchüCal
'ausgegeben. / Because of the methode (GenerateGlasslistCOM) the glass, and
panel data are given out autonomous from the made options in SchüCal.

'Die Methode ist wie folgt definiert:
'GeterateGlasslistCOM(matpot,
    a_bLong,
    a_lSelection,
    a_bScetchModelType,
    a_bScetchStepGlas,
    a_lMeasureFormat), wobei
' „matpot“ der Materialtopf ist,
'a_bLong, Flag das festlegt, ob Lang- oder Kurzform ausgegeben wird. Bei
'Ausgabe in Langform erfolgt feldweise Ermittlung, wobei unter anderem die
'Feldnummer ermittelt wird.
'a_lSelection, long-Wert, der die Werte 0,1,2 annehmen kann.
'0: Ermittlung von Glas und Panel
'1: nur Glas wird ermittelt
'2: nur Panel
'a_bScetchModelType, Flag, um die Skizzen für Modellscheiben zu ermitteln
'a_bScetchStepGlas, Flag um die Skizzen für Stufenisoliererglas zu bekommen
'a_lMeasureFormat, long-Wert der die Bemassung festlegt. Folgende Werte
können 'angenommen werden:
' 0: Bemaßung in mm auf 1mm genau
' 1: Bemaßung in cm auf 1mm genau
' 2: Bemaßung in mm auf 1/10 mm genau
' 3: Bemaßung in cm auf 1/10 mm genau (2 Nachkommastellen)
' unten dargestelltes Beispiel gibt die Ausgabedaten in Langform und Glas-
und 'Paneeldaten aus.
' Dieses Bsp. Zeigt auch wie die Feldinfo ermittelt werden kann.
' (oGlass.FieldNumber()) /
'The method is defined as following:
'GeterateGlasslistCOM(matpot,
    a_bLong,
```

```

    a_lSelection,
    a_bScetchModelType,
    a_bScetchStepGlas,
    a_lMeasureFormat), in which
' „matpot“ is the matpot,
'a_bLong, flag that determins whether the long or the short form is given
out. 'At 'output in long form the determining happens by field, in which
among 'other things the field number is determined.
'a_lSelection, long-Wert, that can assume the value 0,1,2.
'0: Determining of glass and panel
'1: only determining the glass
'2: only panel
'a_bScetchModelType, Flag, to determining the drawing of model panes
'a_bScetchStepGlas, Flag to determining the drawing of step double glazing
'a_lMeasureFormat, long-Wert that determines the dimensioning. The
following 'values can be assumed:
' 0: dimensioning in mm at exactly 1mm
' 1: dimensioning in cm at exactly 1mm
' 2: dimensioning in mm at exactly 1/10 mm (2 decimal places)
' 3: dimensioning in cm at exactly 1/10 mm
'The given example shows the output data in long form with glass and panel.
'The given example shows also how the field info can be determined.
'(oGlass.FieldNumber())

oGlasslist.GenerateGlasslistCOM(matpot, True, 0, True, True, 0)
'Die generierten Glas-Ausgabedaten werden als ein Container zurückgegeben /
The generated glass output data will be returned as a container
oGlasslistResult = oGlasslist.Result
oGlasslistPrices = oGlasslistResult.COMEntries
'Anzahl wird ermittelt / Number is determined
lGlasses = oGlasslistPrices.Count
'Schleife durch alle Datensätze / Loop through all data records
Dim lRun As Long
For lRun = 0 To lGlasses-1
    Dim oGlass As Schuecal2000TLB.IOutpGlasslistPrice
    oGlass = oGlasslistPrices.Item(lRun)
    Dim bIsPanel As Boolean
    bIsPanel = oGlass.IsPanel
    'Glas- oder Paneeltypbezeichnung / Glass or panel type name
    Dim strArticle As String
    strArticle = oGlass.Description
    'Anzahl der Scheiben/Paneele / Number of panes/panels
    Dim dAmount As Double
    dAmount = oGlass.Amount
    'Breite der Scheibe/Paneel [cm] / Width of the pane/panel [cm]
    Dim dMaxWidth As Double
    dMaxWidth = oGlass.MaxWidth
    'Höhe der Scheibe/Paneel [cm] / Height of the pane/panel [cm]
    Dim dMaxHeight As Double
    dMaxHeight = oGlass.MaxHeight
    'Preis pro Scheibe oder Paneel / Price per pane or panel
    Dim dGlassPrice As Double
    dGlassPrice = oGlass.GlassPrice
    'Stärke / Thickness
    Dim dThickness As Double
    dThickness = oGlass.Thickness
    Dim lFrameID As Long
    lFrameID = oGlass.FrameID
    'Feldinfo / field info
    Dim strFieldnr As String
    strFieldnr = oGlass.FieldNumber

```



```

'Dateiname bzw. Pfad der Grafik-Datei der Skizze / File name resp. file
path of the sketch graphic file
Dim strFilename As String
strFilename = oGlass.FileName
'U-Wert / u-value
Dim dUValue As Double
dUValue = oGlass.Uvalue
'Gewicht pro Qm / Weight per qm
Dim dGlasWeightperSqm As Double
dGlasWeightperSqm = oGlass.GlassWeightPerSqm
'Dicke Glas1 (Die Nummerierung der Gläser erfolgt von außen nach innen)
/ Thickness Glas1 (Numbering of the glasses is carried out from the
outside to the inside)
Dim lThicknessGlass1 As Long
lThicknessGlass1 = oGlass.ThicknessGlass1
'Dicke Glas2 / Thickness Glass2
Dim lThicknessGlass2 As Long
lThicknessGlass2 = oGlass.ThicknessGlass2
' Dicke Glas3 / Thickness Glass3
Dim lThicknessGlass3 As Long
lThicknessGlass3 = oGlass.ThicknessGlass3
'Glaszwischenraum (Abstand zwischen Glas1 und Glas2) / air space
(space between Glass1 and Glass2)
Dim lAirDistance1 As Long
lAirDistance1 = oGlass.AirDistance1
'Glaszwischenraum (Abstand zwischen Glas2 und Glas3) / air space
(space between Glass2 and Glass3)
Dim lAirDistance2 As Long
lAirDistance2 = oGlass.AirDistance2
'Abfrage, ob Stufenglas vorhanden ist / Query whether a stepped glazing
exists
Dim bIsStepGlas As Boolean
bIsStepGlas = oGlass.IsStepGlass
' Abfrage, ob Glas mit Isoliersprossen vorhanden ist / Query whether
glass with insulating sash bars exists
Dim bIsSashBarGlas as Boolean
BIsSashBarGlas = oGlass.IsIsoSashbarGlas
Next

'Um allgemeine Informationen über das Glas, den Aufbau und den technischen
Daten zu erhalten, wird wie folgt ermittelt:
To retrieve general information about the glass, the glass composition and
technical specifications, the procedure is as follows:
Dim oGlassInfo As Schuecal2000TLB.OutpGlasslistGlasses
oGlassInfo = oGlasslist.Glasses(matpot)

'Anzahl wird ermittelt / Number is determined
Dim lGlassesInfo As Long
lGlassesInfo = oGlassInfo.Count
'Schleife durch alle Datensätze / Loop through all data records
Dim lInfos As Long
For lInfos = 0 To lGlassesInfo - 1
    Dim oInfo As OutpGlasslistGlass
    oInfo = oGlassInfo.Item(lInfos)
    'Abfrage, ob Panel oder Glas / Query whether glass or panel
    Dim bIsPanel as Boolean
    bIsPanel = oInfo.IsPanel
    'Beschreibung / Description
    Dim strDescription as String
    strDescription = oInfo.Description
    'Kurzbeschreibung / short Description

```

```

Dim strShortDescription as String
strShortDescription = oInfo.ShortText
'Dicke außen / outer thickness
Dim lThicknessOuter as long
lThicknessOuter = oInfo.GetThicknessGlass1
Dim lThicknessMiddle as long
Dim lThicknessInner as long

If (oInfo.GetThicknessGlass3() == 0) Then
'Dicke mitte / middle thickness
lThicknessMiddle = 0
'Dicke innen / inner thickness
lThicknessInner = oInfo.ThicknessGlass2
Else
'Dicke mitte / middle thickness
lThicknessMiddle = oInfo.ThicknessGlass2
'Dicke innen / inner thickness
lThicknessInner = oInfo.ThicknessGlass3
End If
...
Next

```

7.9. Eckvulkanisierte Dichtungsrahmen einer Position / Corner vulcanised gasket frames of an item.

```

'Projektbezeichnung / Project name
Dim strJob As String
'Positionsbezeichnung / Item name
Dim strItem As String
'Flag für die Optimierung / Flag for the optimisation
Dim bOptimized As Boolean

'Benutzerrechte ermitteln / Determine user rights

'Stückzahl ermitteln / Determining number of units
Dim lCount As Long
lCount = oItem.ItemCount

'Fertigungslos erzeugen / Creating production batch

'Material ermitteln / Calculate additional material

'Modul zur Erstellung der Ausgabedaten anlegen / Creating module for the
creation of the output data
Dim oGenerate As OutpGenerateBulklist
Set oGenerate = SApp.GenerateBulklist

'Die Preise immer ermitteln / Always determine the prices
oGenerate.SetPrintPriceCOM( True )

'Generieren der Ausgabedaten. Erster Parameter ist der Container auf alle
'Materialdaten (Materialtopf), der zweite und der dritte Parameter sind
'Flags, die aussagen, ob die Ermittlung mit oder ohne Optimierung
'durchgeführt werden soll.
Generating the output data. First parameter is the container on all
'material data (material list), the second and third parameter are flags,
'which tell, if the determination shall take place with or without
'optimisation.
oGenerate.GenerateOptimizedBulklist matpot, bOptimized, bOptimized

```

```

'Die generierten Ausgabedaten werden als ein Container (OutpBulklistResult)
'zurückgegeben
The generated output data are returned as a container '
(OutpBulklistResult)
Dim oResult As OutpBulklistResult2
Set oResult = oGenerate.Result

'Die Ausgabedaten liefern den Container für die eckvulkanisierten
'Dichtungsrahmen (OutpBulklistVulcanizedFrames). Der Container beinhalten
die 'Dichtungsrahmen, die nach Artikelnummer, Breite, Höhe, Projekt und
Position, 'zu der sie gehören, gruppiert sind.
The output data provide the container with corner vulcanised gasket frames
(OutpBulklistVulcanizedFrames). The containers contain the gasket frames,
which are grouped by article number, width, height, project and item to
'which they belong.
Dim oVulcanizedFrames As OutpBulklistVulcanizedFrames
Set oVulcanizedFrames = oResult.COMVulcanizedFrameEntriesPerItem

'Anzahl der Datensätze für die eckvulkanisierten Dichtungsrahmen wird
'ermittelt (OutpBulklistVulcanizedFrames). /
The number of data sets for the corner vulcanised gasket frames will be
determined.
Dim lFramesData As Long
lFramesData = oVulcanizedFrames.Count
Dim lRun As Long
For lRun = 0 To lFramesData - 1
    Dim oFrame As OutpBulklistVulcanizedFrame
    Set oFrame = oVulcanizedFrames.Item(lRun)
    'Anzahl der eckvulkanisierten Dichtungsrahmen / Number of corner
    vulcanized gasket frames
    Dim dAmount As Double
    dAmount = oFrame.Amount
    'Preis für einen eckvulkanisierten Dichtungsrahmen / Price for a
    corner vulcanised gasket frame
    Dim dPrice As Double
    dPrice = oFrame.Price* dAmount
    'Breite des eckvulkanisierten Dichtungsrahmen / Width of the corner
    vulcanised gasket frame
    Dim dWidth As Double
    dWidth = oFrame.Width
    'Höhe des eckvulkanisierten Dichtungsrahmen / Height of the corner
    vulcanised gasket frame
    Dim dHeight As Double
    dHeight = oFrame.Height
    'Artikelnummer des eckvulkanisierten Dichtungsrahmen / Article number
    of the corner vulcanised gasket frame
    Dim strArticleNr As String
    strArticleNr = oFrame.ArticleNumber
    'Projekt zu dem der eckvulkanisierte Dichtungsrahmen gehört /
    Project, to which the corner vulcanised gasket frame belongs
    Dim strJobID As String
    strJobID = oFrame.JobID
    'Position zu der der eckvulkanisierte Dichtungsrahmen gehört / Item
    the corner vulcanised gasket frame belongs to.
    Dim strItemID As String
    strItemID = oFrame.ItemID
Next

```

7.10. Türfüllungen / Door infills

In SchüCal ist es möglich, Türfüllungen festzulegen. Um für den Bestellprozess das Bestellblatt für die Türfüllung zu erhalten, wird wie folgt vorgegangen:

SchüCal allows you to define door infills. To get the order of the door infill for the ordering process, the procedure is as follows:

```
...
'Modul zur Erstellung der Ausgabedaten anlegen / Create module for the
creation of output data

Dim oGenerate As OutpGenerateBulklist
oGenerate = s_oApp.GenerateBulklist
oGenerate.SetCreateOrderFile(True)

'Die Preise immer ermitteln / Always determine the prices
oGenerate.SetPrintPriceCOM(True)

oGenerate.GenerateOptimizedBulklist(matpot, bOptimized, True)

Dim oResult As OutpBulklistResult6
oResult = oGenerate.Result

'Die Daten der Türfüllungen / door infill data
Dim oDoorPanels As OutpBulklistDoorPanelOrderDatas
oDoorPanels = oResult.DoorPanelOrdersCOM

'Anzahl der Datensätze wird ermittelt / Number of data sets is determined
Dim lDoorPanelsCount As Long
lDoorPanelsCount = oDoorPanels.Count

'Schleife durch alle Datensätze / Loop through all data records
Dim lRun As Long
For lRun = 0 To lDoorPanelsCount - 1
    Dim oDoorPanel As Schuecal2000TLB.OutpBulklistDoorPanelOrderData
    oDoorPanel = oDoorPanels.Item(lRun)
    Dim strJobFileID As String
    strJobFileID = oDoorPanel.JobFileID
    Dim strItemFileID As String
    strItemFileID = oDoorPanel.ItemFileID
    Dim strFieldNo As String
    strFieldNo = oDoorPanel.FieldNo
    'link für das Bestellblatt / link for the order form
    Dim strLink As String
    strLink = oDoorPanel.DoorPanelOrderLink
    Dim strSupplier As String
    strSupplier = oDoorPanel.Supplier
Next
```

7.11. Integralmaster

```
Dim matpot As MatpotNode 'Materialtopf
oBatch = oFactory.GetClassObject("JimVirtualItems")
oBatch.AddItem(strJob, strItem, lItemCount, False)
Dim oAscertainment As TopAscertainment
oAscertainment = s_oApp.TopAscertainment
```

```

matpot = s_oApp.TopAscertainment.GetMaterialOfProductionBatch(oBatch)
Dim oGenerate As OutpGenerateBulklist
oGenerate = s_oApp.GenerateBulklist
oGenerate.SetPrintPriceCOM(True)
oGenerate.GenerateOptimizedBulklist(matpot, bOptimized, True)
Dim oResult As OutpBulklistResult6
oResult = oGenerate.Result

Dim oIntMasterDatas As Schuecal2000TLB.OutpBulklistIntegralmasterDatas
oIntMasterDatas = oResult.COMIntegralmasterEntries

'Anzahl der Datensätze des Integralmasters wird ermittelt / Number of data
records of Integralmaster data are determined
Dim lDatas As Long
lDatas = oIntMasterDatas.Count

'Durch alle Datensätze / By all data sets
Dim lRun As Long
For lRun = 0 To lDatas - 1
    Dim oData As Schuecal2000TLB.OutpBulklistIntegralmasterData
    oData = oIntMasterDatas.Item(lRun)
    Dim strArticle As String
    strArticle = oData.ArticleNumber
    Dim lCount As Long
    lCount = oData.Count
    Dim strField As String
    strField = oData.FieldNo
    Dim dHeight As Double
    dHeight = oData.Height
    'Öffnungstyp / Openingtype
    Dim iOpenType As Integer
    iOpenType = oData.OpeningType
    Dim iMode As Integer
    iMode = oData.OperatingMode
    Dim iSide As Integer
    iSide = oData.OperatingSide
    Dim dPrice As Double
    dPrice = oData.Price
    Dim dWidth As Double
    dWidth = oData.Width
Next

```

7.12. Oberflächendaten einer Position lesen / Reading surface data of an item

```

'Projektbezeichnung / Project name
Dim strJob As String
'Positionsbezeichnung / Item name
Dim strItem As String
'Flag für die Optimierung / Flag for the optimisation
Dim bOptimized As Boolean

'Benutzerrechte ermitteln / Determine userrights

'Stückzahl ermitteln / Determining number of units
Dim lCount As Long
lCount = oItem.ItemCount

'Fertigungslos erzeugen / Creating production batch

```

```
'Material ermitteln / Calculate additional material

'Modul zur Erstellung der Ausgabedaten anlegen / Create module for the
creation of output data
Dim oGenerate As OutpGenerateBulklist
Set oGenerate = SApp.GenerateBulklist

'Die Preise immer ermitteln / Always determine the prices
oGenerate.SetPrintPriceCOM( True )

'Generieren der Ausgabedaten. Erster Parameter ist der Container auf alle
'Materialdaten (Materialtopf), der zweite und der dritte Parameter sind
Flags, 'die aussagen, ob die Ermittlung mit oder ohne Optimierung
durchgeführt wird
Generating output data. First parameter is the container on all material
'data (material list), the second and third parameter are flags, which
'tell, whether the determination shall take place with or without
'optimisation.
oGenerate.GenerateOptimizedBulklist matpot, bOptimized, bOptimized

'Die generierten Ausgabedaten werden als Container (OutpBulklistResult)
'zurückgegeben
Generated output data is handed back as a container (OutpBulklistResult)
Dim oResult As OutpBulklistResult2
Set oResult = oGenerate.Result

'Die Ausgabedaten liefern den Container für die Oberflächendaten
'(OutpBulklistSurfacePrices). Der Container beinhaltet die einzelnen
'Oberflächendaten
The output data provide the container for surface data
(OutpBulklistSurfacePrices). The container contains the particular surface
data
Dim oSurfacePrice As OutpBulklistSurfacePrices
Set oSurfacePrices = oResult.COMSurfaceEntries

'Anzahl der Datensätze der Oberflächendaten wird ermittelt / Number of data
records of surface data are determined
Dim lPrices As Long
lPrices = oSurfacePrices.Count

'Durch alle Datensätze / By all data sets
Dim lRun As Long
For lRun = 0 To lPrices - 1
    Dim oPrice As OutpBulklistSurfacePrice2
    Set oPrice = oSurfacePrices.Item(lRun)
    'Oberflächennummer / Surface number
    Dim lSurface As Long
    lSurface = oPrice.Surface
    'Farbton (z.B. RAL-Nr.) / Colour (e.g. RAL-No)
    Dim strColour As String
    strColour = oPrice.Colour
    'Preis für die umlaufende Oberfläche / Price for the circumfering
    surface
    Dim dPriceSurface As Double
    dPriceSurface = oPrice.PriceSurfaceArea
    'Fläche der umlaufenden Oberfläche [qm] / Area of the circumfering
    surface [m²]
    Dim dSurfaceArea As Double
    dSurfaceArea = oPrice.SurfaceArea
    'Preis für die mechanisch bearbeitete Oberfläche / Price for the
```

```

mechanically processed surface
Dim dPriceOfPolish As Double
dPriceOfPolish = oPrice.PriceLengthOfPolish
'Fläche der mechanisch bearbeiteten Oberfläche [qm] / Area of the
mechanically processed surface [m²]
Dim dLenghtOfPolish As Double
dLengthOfPolish = oPrice.LengthOfPolish
Next

```

7.13. Bezeichnungen für Oberflächen/Farben

DabSurfaceManager2 ist die Klasse, die als Schnittstelle dient, um die Bezeichnungen für Oberfläche und Farben zu ermitteln: */ DabSurfaceManager2 is the interface to ascertain the surface and color description and information*

```

Dim pSurface As Schuecal2000TLB.DabSurfaceManager2
pSurface = s_oApp.Factory.GetClassObject("SurfaceManager")

Dim oPrice As OutpBulklistPrice6
Dim strColour as String
strColour = oPrice.InnerColour
Dim strColourname As String
strColourname = pSurface.NameOfColour(strColour)
Dim binsidesurface As Boolean
binsidesurface = oPrice.HasInsideSurface
If (binsidesurface = True) Then
    lSurface = oPrice.InnerSurface
    'Farbe aussen / Colour outside
    strOuterColour = oPrice.OuterColour
    'Oberfläche aussen / Surface outside
    Dim boutsidesurface As Boolean
    boutsidesurface = oPrice.HasOutsideSurface
    If (boutsidesurface = True) Then
        lOuterSurface = oPrice.OuterSurface
        Dim strOuterColourname As String
        strOuterColourname = pSurface.NameOfColour(strOuterColour)
    End If
    'Oberfläche mitte / Surface middle
    Dim lmiddleSurface As Long
    lmiddleSurface = oPrice.MiddleSurface

    'Farbe mitte / color middle
    Dim strmiddleColour As String
    strmiddleColour = oPrice.MiddleColour

    'Farbbezeichnung/ color name
    Dim strmiddleColourname As String
    strmiddleColourname = pSurface.NameOfColour(strmiddleColour)
End If

'Oberflächenbezeichnung aus dem Projekt / surface description from the
project/job
Dim treattext As String
treattext = pSurface.Coating(lSurface, oJob.JobID), wobei lSurface die
innere bzw mittlere oder äußere Oberfläche sein kann / where lSurface is
the inner- or middle- or outer surface

'Liefert Farbcode + Bezeichnung zurück / returned colorcode + description
Dim strcolor As String
strcolor = pSurface.Colour(strColour)

```

8. Kalkulation / Cost data

8.1. Kalkulatorische Daten einer Position lesen / Reading cost data of an item

(Versiegelungskosten, Glaskosten, Paneelkosten, Verglasungskosten, Kosten fürs Verbinden der Profile, Sonderkosten, Fertigungskosten, Montagekosten, Kosten fürs Schneiden, Kosten fürs Kanten, Kosten für die konstruktive Bearbeitung).

(Sealing costs, glass costs, panel costs, glazing costs, costs for connecting the profiles, special costs, fabrication costs, assembly costs, costs for cutting, costs for edging, costs for project design).

```
s_oApp = New Schuecal2000TLB.Schuecal2000App
'Projektbezeichnung / Project name
Dim strJob As String
'Positionsbezeichnung / Item name
Dim strItem As String
strItem = "01"
'Flag für die Optimierung / Flag for the optimisation
Dim bOptimized As Boolean
bOptimized = True
Dim oJob As Job
Dim oBatch As JimVirtualItems
'Factory ist eine spezielle Klasse, von der Klassen mit statischen Methoden
'abgeleitet werden können / Factory is a special class from which classes
with statistical methods can be derived
Dim oFactory As Object
oFactory = s_oApp.Factory
Dim oItem As Item

oJob = s_oApp.JobMaster.Jobs.GetNewJob(strJob)
oItem = oJob.Items.GetNewItem(strItem)

Dim matpot As MatpotNode 'Materialtopf

'Stückzahl ermitteln / Determining number of units
Dim lCount As Long
lCount = oItem.ItemCount

'Fertigungslos erzeugen / Create production batch
oBatch = oFactory.GetClassObject("JimVirtualItems")
'Fertigungslos füllen / Fill production batch
oBatch.AddItem(strJob, strItem, lCount, False)
...
'Ermittlung-modul anlagen / Creating ascertainment module
Dim oAscertainment As TopAscertainment
oAscertainment = s_oApp.TopAscertainment

'Den Container auf alle Materialdaten anlegen und alle Materialdaten für
'diese Position erzeugen / Create container on all material data and create
all material data for this item
matpot = s_oApp.TopAscertainment.GetMaterialOfProductionBatch(oBatch)
'Kalkulation starten / Starting calculation
Dim oCalcCalculation As CalcCalculation
oCalcCalculation = s_oApp.GetCalculation
oCalcCalculation.SetOfferCalculation(True)
'Kalkulationstöpfe ermitteln / Determine calculation pot
```



```

Dim oCalcCalcPot As CalcCalcPot
oCalcCalcPot = oCalcCalculation.Calculate(matpot)
'Anzahl der Kalkulationstöpfe wird ermittelt / Number of calculation pots
is determined
Dim lPots As Long
lPots = oCalcCalcPot.NumberofPots

Dim oCalcCalcPotData As CalcCalcPotData8 (erst ab SchüCal 2017 R1)
oCalcCalcPotData = oCalcCalcPot.GetCalcPotData(strJob, strItem)
'Kosten der Versiegelung für die Versiegelung pro Meter / Sealing costs per
meter
Dim dSealingCosts As Double
dSealingCosts = lCount * oCalcCalcPotData.SealingCosts
'Versiegelungslänge [m] / Sealing length [m]
Dim dSealingAmount As Double
dSealingAmount = lCount * oCalcCalcPotData.SealingAmount

'Versiegelungskosten pro h / sealing costs per hour
Dim dSealingH As Double
dSealingH = oCalcCalcPotData.SealingCostsperH
'Versiegelung pro h (Menge)
Dim dsealingAmountH As Double
dsealingAmountH = oCalcCalcPotData.SealingAmountperH

'Kosten für alle Paneele Die Daten der einzelnen Paneele sind in den
'Glasdaten zu finden. / Cost for all panels. The data for each of the
panels can be found in the 'glass data
Dim test1 As Double
test1 = oCalcCalcPotData.GlassEnergyChargeCosts
'Preis Profile inkl. kalkulatorischer Aufschlag / Price of profiles
(calculated supplement included)
Dim dPWithS As Double
dPWithS = oCalcCalcPotData.PriceProfilesWithScrap
'Preis Profile ohne kalk. Aufschlag / Price of profiles without calculated
supplement

Dim dPWithoutS As Double
dPWithoutS = oCalcCalcPotData.PriceProfilesWithoutScrap
'Preis Oberflächen inkl. Kalk. Aufschlag / surface price (calculated
supplement included)
Dim dPSwithS As Double
dPSwithS = oCalcCalcPotData.PriceSurfaceWithScrap
'Preis Oberflächen ohne kalk. Aufschlag / surface price without calculated
supplement

Dim dPSWithoutS As Double
dPSWithoutS = oCalcCalcPotData.PriceSurfaceWithoutScrap
Dim dPanelCosts As Double
dPanelCosts = lCount * oCalcCalcPotData.PanelCosts
'Fläche aller Paneele [qm] / Area of all panels [m²]
Dim dPanelArea As Double
dPanelArea = lCount * oCalcCalcPotData.PanelArea
'Kosten für alle Gläser. Die Daten der einzelnen Gläser sind in den
'Glasdaten zu finden. / Costs for all glass. The data of each glass can be
found in the glass 'data.
Dim dGlassCosts As Double
dGlassCosts = lCount * oCalcCalcPotData.GlassCosts 'Material
'Fläche aller Gläser [qm] / Area of all glasses [m²]
Dim dGlassArea As Double
dGlassArea = lCount * oCalcCalcPotData.GlassArea

```

```

'Verglasungskosten / Glazing costs
Dim dGlassingCosts As Double
dGlassingCosts = lCount * oCalcCalcPotData.GlassingCosts 'Lohn / costs
'Verglasungsfläche [qm] / Glazing area [m²]
Dim dGlassingArea As Double
dGlassingArea = lCount * oCalcCalcPotData.GlassingArea
'Verbindungskosten / Connection costs
Dim dJoinOfPCosts As Double
dJoinOfPCosts = lCount * oCalcCalcPotData.JoinOfProfileCosts
'Die zu verbindende Länge [m] / The length to be connected [m]
Dim dJoinOfPAmount As Double
dJoinOfPAmount = lCount * oCalcCalcPotData.JoinOfProfileAmount
'Sonderkosten aus dem Zubehör / Special costs from the accessories
Dim dSpecialCosts As Double
dSpecialCosts = lCount * oCalcCalcPotData.SpecialCosts
'Fertigungskosten / Fabrication costs
Dim dFabricationCosts As Double
dFabricationCosts = lCount * oCalcCalcPotData.FabricationCosts
'Fertigungszeit [h] / Fabrication time [h]
Dim dFabricationTime As Double
dFabricationTime = lCount * oCalcCalcPotData.FabricationTime
'Montagekosten / Assembly costs
Dim dInstallCosts As Double
dInstallCosts = lCount * oCalcCalcPotData.InstallationCosts
'Montagezeit [h] / Assembly time [h]
Dim dInstallTime As Double
dInstallTime = lCount * oCalcCalcPotData.InstallationTime
'Kosten für das Schneiden / Costs for cutting
Dim dCuttingCosts As Double
dCuttingCosts = lCount * oCalcCalcPotData.CuttingCosts
'Zeit für das Schneiden [h] / Time for cutting [h]
Dim dCuttingAmount As Double
dCuttingAmount = lCount * oCalcCalcPotData.CuttingAmount
'Kosten für das Kanten / Costs for edging
Dim dEdgingCosts As Double
dEdgingCosts = lCount * oCalcCalcPotData.EdgingCosts
'Zeit für das Kanten [h] / Time for edging [h]
Dim dEdgingAmount As Double
dEdgingAmount = lCount * oCalcCalcPotData.EdgingAmount
'Kosten für die konstruktive Bearbeitung / Costs for the project design
Dim dConstructiveEditCosts As Double
dConstructiveEditCosts = lCount * oCalcCalcPotData.ConstructiveEditCosts
'Zeit für die konstruktive Bearbeitung / Time for the constructive
processing
Dim dConstructiveEditAmount As Double
dConstructiveEditAmount = lCount * oCalcCalcPotData.ConstructiveEditAmount
'Materialeinzelkosten / Individual material costs
Dim dTotalMaterialPrice As Double
dTotalMaterialPrice = oCalcCalcPotData.TotalDirectMaterialCosts
'Positionspreis / Item price
Dim dPrice As Double
dPrice = oCalcCalcPotData.Price

(Ab SchüCal 2014 R3)
'Verwaltungs- und Vertriebsgemeinkosten / Sales administration overheads
Dim dAdministrativeCosts As Double
dAdministrativeCosts = oCalcCalcPotData.AdminAndMarketingCosts
'Risiko und Gewinn / Profit and loss
Dim dRiskProfit As Double
dRiskProfit = oCalcCalcPotData.RiskProfit
'Provision / Commission

```

```
Dim dProvision As Double
dProvision = oCalcCalcPotData.Provision
'Teuerungszuschlag / Surcharge
Dim dExtraCharge As Double
dExtraCharge = oCalcCalcPotData.ExtraCharge
'Verteilkosten / Distribution costs
Dim dDistributedCosts As Double
dDistributedCosts = oCalcCalcPotData.DistributedCosts
'Mehrwertsteuer Bezeichnung / VAT Description
Dim strVATDescription As String
strVATDescription = oCalcCalcPotData.VATDescription
'Mehrwertsteuer Prozentsatz / VAT in Percent
Dim dVATValue As Double
dVATValue = oCalcCalcPotData.VATValue
'Mehrwertsteuer Text für Listenende / VAT Long Description for end of list
Dim strVATLongDescription As String
strVATLongDescription = oCalcCalcPotData.VATLongDescription
```

8.2. Kalkulatorische Daten von zusätzlichen Positionen lesen (Ab SchüCal 2017 R1) / Reading cost data of additional items (from SchüCal 2017 R1)

Rüstkosten Beschichtung, Diff. Mindestauftragswert Beschichtung und Zusätzliche Projektkosten, können im Projekt auf einzelne Positionen verteilt oder als zusätzliche Positionen ermittelt werden. Als zusätzliche Positionen ermittelte Kosten werden bei der Kalkulation automatisch in den Kalkulationstopf eingetragen. *Coating setup costs, differences to minimum order values for coating and additional job costs, can be distributed to single items or can be ascertained as additional items. Items, which has been ascertained as additional items, will be added to the calculation pot automatically.*

```
s_oApp = New Schuecal2000TLB.Schuecal2000App
'Projektbezeichnung / Project name
Dim strJob As String
'Positionsbezeichnung / Item name
Dim strItem As String
strItem = "01"
'Flag für die Optimierung / Flag for the optimisation
Dim bOptimized As Boolean
bOptimized = True
Dim oJob As Job
Dim oBatch As JimVirtualItems
'Factory ist eine spezielle Klasse, von der Klassen mit statischen Methoden
'abgeleitet werden können / Factory is a special class from which classes
with statistical methods can be derived
Dim oFactory As Object
oFactory = s_oApp.Factory
Dim oItem As Item

oJob = s_oApp.JobMaster.Jobs.GetNewJob(strJob)
oItem = oJob.Items.GetNewItem(strItem)

Dim matpot As MatpotNode 'Materialtopf

'Stückzahl ermitteln / Determining number of units
Dim lCount As Long
lCount = oItem.ItemCount
```

```
'Fertigungslos erzeugen / Create production batch
oBatch = oFactory.GetClassObject("JimVirtualItems")
'Fertigungslos füllen / Fill production batch
oBatch.AddItem(strJob, strItem, lCount, False)
...
'Ermittlung-modul anlagen / Creating ascertainment module
Dim oAscertainment As TopAscertainment
oAscertainment = s_oApp.TopAscertainment

'Den Container auf alle Materialdaten anlegen und alle Materialdaten für
'diese Position erzeugen / Create container on all material data and create
all material data for this item
matpot = s_oApp.TopAscertainment.GetMaterialOfProductionBatch(oBatch)
'Kalkulation starten / Starting calculation
Dim oCalcCalculation As CalcCalculation
oCalcCalculation = s_oApp.GetCalculation
oCalcCalculation.SetOfferCalculation(True)
'Kalkulationstöpfe ermitteln / Determine calculation pot
Dim oCalcCalcPot As CalcCalcPot
oCalcCalcPot = oCalcCalculation.Calculate(matpot)
'Anzahl der Kalkulationstöpfe wird ermittelt / Number of calculation pots
is determined
Dim lPots As Long
lPots = oCalcCalcPot.NumberofPots

'Schleife durch die Kalkulationstöpfe / loop through the calculation pots
Dim lPot As Integer
For lPot = 0 To lPots - 1
    Dim oCalcCalcPotData As CalcCalcPotData8 (erst ab SchüCal 2017 R1)
    oCalcCalcPotData = oCalcCalcPot.GetCalcPotData1(lPot)

    Dim eItemType As Schuecal2000TLB.ECalcedItemType
    eItemType = oCalcCalcPotData.ItemType

    If eItemType = eItemType.eItem Then
        'Daten der Position abfragen / Read data of item
    ElseIf eItemType = eItemType.eSurfacePlatingSetupCosts
        'Daten der zusätzlichen Position für Rüstkosten Beschichtung
        lesen / Read data of additional item of coating setup costs
    ElseIf eItemType = eItemType.eSurfacePlatingMinOrderCosts
        'Daten der zusätzlichen Positionen für Diff.
        Mindestauftragswert Beschichtung lesen / Read data of additional item of
        differences to minimum order values for coating
    ElseIf eItemType = eItemType.eJobAdditionalCosts
        'Daten der zusätzlichen Position für zusätzliche Projektkosten
        lesen / Read data of additional item of additional job costs
    End If
Next
```

8.3. Beschichtungskosten gesamt / Total coating costs

```
'Oberflächenpreise / Surface costs
Dim dPriceSurface As Double
DPriceSurface = lCount * oCalcCalcPotData.SurfaceSurroundingCosts
Dim dAreaSurface As Double
DAreaSurface = lCount * oCalcCalcPotData.SurfaceSurroundingAmount
Dim dPriceOfPolish As Double
DPriceOfPolish = lCount * oCalcCalcPotData.SurfacePolishedCosts
Dim dAreaPolish As Double
```

```
DAreaPolish = lCount * oCalcCalcPotData.SurfacePolishedAmount
```

8.4. Beschichtungskosten pro Oberfläche / Coating costs per surface

```
Dim oBatch As Schuecal2000TLB.IJimVirtualItems
Dim oFactory As Object
oFactory = s_oApp.Factory
oBatch = oFactory.GetClassObject("JimVirtualItems")
oBatch.AddItem(strJob, strItem, lItemCount, False)
Dim oAscertainment As Schuecal2000TLB.TopAscertainment
oAscertainment = s_oApp.TopAscertainment
Dim matpot As Schuecal2000TLB.IMatpotNode
matpot = s_oApp.TopAscertainment.GetMaterialOfProductionBatch(oBatch)
Dim oGenerate As Schuecal2000TLB.OutpPrintNormalBulklist
oGenerate = s_oApp.PrintNormalBulklist
Dim oResult As Schuecal2000TLB.OutpBulklistResult2
oResult = oGenerate.COMGenerateBulklist(matpot)
Dim oSurfacePrices As Schuecal2000TLB.IOutpBulklistSurfacePrices
oSurfacePrices = oResult.COMSurfaceEntries
Dim lPrices As Long
lPrices = oSurfacePrices.Count
Dim lRun As Long
For lRun = 0 To lPrices - 1
    Dim oPrice As Schuecal2000TLB.OutpBulklistSurfacePrice2
    oPrice = oSurfacePrices.Item(lRun)
    Dim lSurface As Long
    lSurface = oPrice.Surface
    'Farbton (z.B. RAL-Nr.) / Colour (e.g. RAL-No)
    Dim strColour As String
    strColour = oPrice.Colour
    'Preis / Price
    Dim dPriceSurface As Double
    dPriceSurface = oPrice.PriceSurfaceArea
    'Fläche / Surface
    Dim dSurfaceArea As Double
    dSurfaceArea = oPrice.SurfaceArea
    'Preis für die mechanisch bearbeitete Oberfläche / Price for
    'the mechanically processed surface
    Dim dPriceOfPolish As Double
    dPriceOfPolish = oPrice.PriceLengthOfPolish
    'Fläche der mechanisch bearbeiteten Oberfläche [qm] / Area of the
    'mechanically processed surface [m²]
    Dim dLenghtOfPolish As Double
    dLenghtOfPolish = oPrice.LengthOfPolish
    'Stückkosten / Unit cost
    Dim dPricePiece As Double
    dPricePiece = oPrice.PricePiece
    'Stück / Number
    Dim dPieceCount As Double
    dPieceCount = oPrice.PieceCount
Next
```

9. CE-Kennzeichnung / CE marking

```
s_oApp = New Schuecal2000TLB.Schuecal2000App
Dim oFactory As Object
Dim oBatch As Schuecal2000TLB.JimVirtualItems
```

```

Dim oAscertainment As Schuecal2000TLB.TopAscertainment
Dim matpot As Schuecal2000TLB.MatpotNode
Dim oCEList As Schuecal2000TLB.OutpGenerateCEList
'CE-Kennzeichnung für Fassaden-Rahmenwerke / CE marking for façade
framework
Dim oCEListDatas As Schuecal2000TLB.OutpCEListDatas
'CE-Kennzeichnung für Öffnungen und Festfelder / CE marking for openings
and fixed lights
Dim oCEListOpeningDatas As Schuecal2000TLB.OutpCEListOpeningDatas
Dim lCEListDatas As Long
Dim lCEListOpeningDatas As Long
Dim strJobI As String
'Positionsbezeichnung / Item name
Dim strItemID As String

oFactory = s_oApp.Factory
oBatch = oFactory.GetClassObject("JimVirtualItems")
oBatch.AddItem(strJobID, strItemID, 1, False)
oAscertainment = s_oApp.TopAscertainment
matpot = s_oApp.TopAscertainment.GetMaterialOfProductionBatch(oBatch)
'Modul zur Erstellung der CE-Kennzeichnung anlegen / Creating module for
the creation of CE markings data
oCEList = s_oApp.GenerateCEList
oCEList.GenerateCEList(matpot)

Dim oCEListResult As Schuecal2000TLB.OutpCEListResult
'Die generierten Daten der CE-Kennzeichnung werden als ein Container
zurückgegeben / The generated CE marking data will be returned as a
container
oCEListResult = oCEList.Result

```

9.1. CE-Kennzeichnung für Fassaden-Rahmenwerke / CE marking for facade framework

```

oCEListDatas = oCEListResult.COMCEItems

'Anzahl wird ermittelt / Number is determined
lCEListDatas = oCEListDatas.Count
'Schleife durch alle Datensätze / Loop through all data records
Dim lRun As Long
For lRun = 0 To lCEListDatas - 1
    Dim oData As Schuecal2000TLB.OutpCEListData2
    oData = oCEListDatas.Item(lRun)
    'Projekt-ID
    strJobID = oData.JobID
    'Item ID
    strItemID = oData.ItemID
    'Projekt-Beschreibung / Project description
    Dim strJobDescription As String
    strJobDescription = oData.JobDescription
    'Positions-Beschreibung / Item description
    Dim strItemDescription As String
    strItemDescription = oData.ItemDescription
    'System-ID
    Dim lSystemID As Long
    lSystemID = oData.SystemID
    'Profilsystem / Profile system
    Dim strProfileSystem As String
    strProfileSystem = oData.ProfileSystem
    'Beschreibung / Description
    Dim strRemark As String
    strRemark = oData.Remark
    'Brandverhalten / Fire behaviour
    Dim lFireBehavior As Long
    lFireBehavior = oData.FireBehavior
    'Feuerwiderstand / Fire resistance
    Dim lFireResistance As Long
    lFireResistance = oData.FireResistance
    'Brandausbreitung / Spread of fire
    Dim strFireExpansion As String
    strFireExpansion = oData.FireExpansion
    'Schlagregendichtheit / Watertightness
    Dim lRainPerformance As Long
    lRainPerformance = oData.RainPerformance
    'Widerstand gegen Eigenlast / Resistance to dead load
    Dim dWeightResistance As Double
    dWeightResistance = oData.WeightResistance
    'Widerstand gegen Windlast / Resistance to wind load
    Dim dWindResistance As Double
    dWindResistance = oData.WindResistance
    'Stoßfestigkeit innen / Impact resistance inside
    Dim lHitInside As Long
    lHitInside = oData.HitInside
    'Stoßfestigkeit aussen / Impact resistance outside
    Dim lHitOutside As Long
    lHitOutside = oData.HitOutside
    'Temperaturwechselbeständigkeit / Resistance to change in temperature
    Dim strTemperatureChange As String
    strTemperatureChange = oData.TemperatureChange
    'Widerstand gegen Horizontallasten / Resistance to horizontal loads
    Dim dHorizontalResistance As Double

```

```

dHorizontalResistance = oData.HorizontalResistance
'Wärmedurchgang / Thermal transmission
Dim dHeatTranmission As Double
dHeatTranmission = oData.HeatTransmission
'Luftdurchlässigkeit / Air permeability
Dim lAirTransmission As Long
lAirTransmission = oData.AirTransmission
'Luftschalldämmung / Airbourne sound insulation
Dim dSoundProtection As Double
dSoundProtection = oData.SoundProtection
'Widerstand gegen Eigenlast / Resistance to dead load
dWeightResistance = oData.WeightResistance
'Widerstand gegen Windlast / Resistance to wind load
dWindResistance = oData.WindResistance
'Nummer der Zertifizierungsstelle / ID number of the certifying body
Dim lCertifier As Long
lCertifier = oData.Certifier
'Zertifikat / Certificate
Dim strCertificate As String
strCertificate = oData.Certificate

```

Next

9.2. CE-Kennzeichnung für Öffnungen / CE marking for openings

```

oCEListOpeningDatas = oData.COMCEOpeningDatas

'Anzahl wird ermittelt / Number is being determined
lCEListOpeningDatas = oCEListOpeningDatas.Count
'Schleife durch alle Datensätze / Loop through all data records
Dim lRunOpenings As Long
For lRunOpenings = 0 To lCEListOpeningDatas - 1
    Dim oOpenData As Schuecal2000TLB.OutpCEListOpeningData
    oOpenData = oCEListOpeningDatas.Item(lRunOpenings)
    'Projekt-ID
    strJobID = oOpenData.JobID
    'Item ID
    strItemID = oOpenData.ItemID
    'ID der Fläche / plane ID
    Dim lPlaneNo As Long
    lPlaneNo = oOpenData.PlaneNo
    'ID des Einselementes / ID number of the insert unit
    Dim lInsertElementNo As Long
    lInsertElementNo = oOpenData.InsertElementNo
    'Feld-Nummer / Surface number
    Dim lFieldNo As Long
    lFieldNo = oOpenData.FieldNo
    'System-ID / system ID
    Dim lSystemID1 As Long
    lSystemID1 = oOpenData.SystemID
    'Öffnungsbeschreibung / Opening description
    Dim strOpeningDescription As String
    strOpeningDescription = oOpenData.OpeningDescription
    'Beschlag / Fitting
    Dim strFitting As String
    strFitting = oOpenData.Fitting
    'Feld / Surface
    Dim strField As String
    strField = oOpenData.Field
    'Breite (nur für Türen) / Width (only for doors)
    Dim dWidth As Double

```



```

dWidth = oOpenData.Width
'Höhe (nur für Türen) / Height (only for doors)
Dim dHeight As Double
dHeight = oOpenData.Height
'Luftschalldämmung / Airbourne sound insulation
Dim dSoundProtection1 As Double
dSoundProtection1 = oOpenData.SoundProtection
'Stoßfestigkeit / Impact resistance
Dim lHitPerformance As Long
lHitPerformance = oOpenData.HitPerformance
'Tragfähigkeit von Sicherheitsvorrichtungen / Load-bearing
capacity of safety devices
Dim strSafety As String
strSafety = oOpenData.Safety
'Wärmedurchgangskoeffizient / U-value
Dim dUValue As Double
dUValue = oOpenData.UValue
'Widerstandsfähigkeit gegen Windlast / Resistance to wind load
Dim dWindPressure As Double
dWindPressure = oOpenData.WindPressure
'Schlagregendichtheit / Watertightness
Dim lRainPerformancel As Long
lRainPerformancel = oOpenData.RainPerformance
'Luftdurchlässigkeit / Air permeability
Dim lAirTransmission1 As Long
lAirTransmission1 = oOpenData.AirTransmission

```

Next

9.3. CE-Kennzeichnung für Sonnenschutz / CE marking for solar shading

```

Dim oCEListSun As Schuecal2000TLB.OutpCEListSunShadingDatas
oCEListSun = oData.COMCESunShadingDatas
'Anzahl wird ermittelt / Number is being determined
Dim lCEListSunDatas As Long
lCEListSunDatas = oCEListSun.Count
'Schleife durch alle Datensätze / Loop through all data records
Dim lSunDatas As Long
For lSunDatas = 0 To lCEListSunDatas - 1
    Dim oSunData As Schuecal2000TLB.OutpCEListSunShadingData
    oSunData = oCEListSun.Item(lSunDatas)
    'Job-ID
    strJobID = oSunData.JobID
    'Item ID
    strItemID = oSunData.ItemID
    'Artikelnummer / Article number
    Dim strArticleSun As String
    strArticleSun = oSunData.ArticleNumber
    Dim lDopNoOption As Long
    lDopNoOption = oSunData.DopNoOption
    Dim strDopNo As String
    strDopNo = oSunData.DopNo
    Dim strUseCase As String
    strUseCase = oSunData.UseCase
    Dim lWindLoad As Long
    lWindLoad = oSunData.WindLoad

```

Next

10. Hinweisliste / Message list

Durch die Ausgabe der Fertigungslisten in SchüCal wird beim Ermitteln einer Position oder eines Projektes auch eine Fehlerliste erzeugt. Die Fehlerliste ist ein Dokument, das während der Ermittlung erzeugt wird. Dieses Dokument enthält Hinweise, die wichtig sind für die Ermittlung und das weitere Vorgehen. Um Konflikte zu vermeiden, bietet sich die Möglichkeit, über COM sowohl den Inhalt dieser Fehlerliste (Hinweistext) zu ermitteln, wie auch den Pfad des Dokumentes.

With the output of fabrication lists in SchüCal, message lists are created while the project is calculated. A message list is a WORD document that is created during the calculation. The document contains information which is important for the calculation and the further processing. To avoid conflicts, COM provides the possibility to determine the content of the message list (additional information) as well as the path of the document.

```
s_oApp = New Schuecal2000TLB.Schuecal2000App
Dim oBatch As Schuecal2000TLB.IJimVirtualItems
Dim oFactory As Object
oFactory = s_oApp.Factory
oBatch = oFactory.GetClassObject("JimVirtualItems")
oBatch.AddItem(strJob, strItem, ItemCount, False)
Dim oAscertainment As Schuecal2000TLB.ITopAscertainment
oAscertainment = s_oApp.TopAscertainment
Dim matpot As Schuecal2000TLB.IMatpotNode
matpot =
s_oApp.TopAscertainment.GetMaterialOfProductionBatch(oBatch)
Dim pError As Schuecal2000TLB.OutpPrintErrorlist
pError = s_oApp.PrintErrorlist
`Fehlertext /error text
Dim strErrorText As String
strErrorText = pError.GetErrorTextProm(oBatch)
`Pfad, in dem die Fehlerliste abgelegt wird / Path in which the error list
is stored
Dim strErrorPath As String
strErrorPath = pError.PrintListToFileProm(oBatch)
MsgBox(strErrorPath)
```

11. Zusätzliche Kosten / Additional costs

In SchüCal ist es möglich, innerhalb des Elementes zusätzliche Lohnkosten zu definieren (z. B. für das Kanten, das Schneiden,...). Diese Kosten sind über die Schnittstelle wie folgt zu ermitteln:

In SchüCal, it is possible to define additional labour costs (e.g. edging, cutting, ...) within an unit. These costs are to be determined via the interface, as to be seen here:

```
s_oApp = New Schuecal2000TLB.Schuecal2000App
Dim JimVal As Schuecal2000TLB.JimCostcenterValues
JimVal = s_oApp.CostcenterValues
Dim ovalues As Schuecal2000TLB.CostcenterValues
ovalues = JimVal.getCostcenterValues("strJob", "strItem")
Dim lvalues As Long
lvalues = ovalues.Count
Dim p As Long
For p = 0 To lvalues - 1
    Dim ovalue As CostcenterValue
    ovalue = ovalues.Item(p)
    Dim strcostcenter As String
```

```

        strcostcenter = ovalue.Costcenter
        Dim eType As Schuecal2000TLB.ETimeType
        eType = ovalue.Type
        Dim dCostvalue As Double
        dCostvalue = ovalue.Timevalue
    Next

```

, wobei / where:
 Strcostcenter: die Id der Kostenstelle ist, / Cost center ID
 eType: ein enum für den Lohnkostentyp ist. / Enum for the labour costs
 Folgende Werte sind möglich: / Possible values are:
 eFAT = Fertigung / Fabrication
 eMAT = Montage, / Assembly
 eEdging = Kanten, / Edging
 eConstructiveEdit = konstruktive Zeit, / Constructive edit
 eCutting = Schneiden, / Cutting
 eSealingStrip = Versiegelung nach Zeit, / Sealing by time
 eSealingStrip2 = Versiegelung nach Meter, / Sealing by metre
 eFreeInputCosts = Frei Eingabe (Kosten pro Element), / Free input (costs per unit)
 eFreeInputTime = Frei Eingabe (Zeit pro Element), / Free input (time per unit)
 eGlazingCosts = Verglasungskosten, / Glazing costs
 eSpecialCosts = Sonderkosten / Special costs
 und dCostvalue der hinterlegte Wert / The stored value

12. Kalkulationsgrunddaten / Cost data block

Der Zugriff auf die projektspezifischen Kalkulationsgrunddaten, erfolgt in SchüCal wie folgt:

Access to the project-specific cost data block in SchüCal takes place as shown here:

```

Dim pCodes As Schuecal2000TLB.CalcEdvCodes3
pCodes = s_oApp.Factory. GetClassObject("CalcEdvCodes")
Dim pCalcAttributes As Schuecal2000TLB.IExportCalcAttribute
pCalcAttributes = s_oApp.Factory.GetClassObject("ExportCalcAttribute")
Dim dPercent As Double
dPercent = pCodes.GetEDVCodeValueFromJob(
pCalcAttributes.CALC_GENERAL_FABRICATION_COSTS,
Schuecal2000TLB.ECodeType.eMaterialAndWages,
Schuecal2000TLB.EValueType.ePercent,
strJob)
, strJob: JobID oder UniqueID des Projektes

Dim pCodes As Schuecal2000TLB.CalcEdvCodes3
pCodes = s_oApp.Factory. GetClassObject("CalcEdvCodes")
Dim pCalcAttributes As Schuecal2000TLB.IExportCalcAttribute
pCalcAttributes = s_oApp.Factory.GetClassObject("ExportCalcAttribute")
Dim dPercent As Double
dPercent = pCodes.GetEDVCodeValueFromBaseData(
pCalcAttributes.CALC_GENERAL_FABRICATION_COSTS,
Schuecal2000TLB.ECodeType.eMaterialAndWages,
Schuecal2000TLB.EValueType.ePercent)

```

13. Erstellen der XML-Datei für die Fertigungszeiten **/ Generate xml-file for the fabrication time**

```
Dim oFatXml As Schuecal2000TLB.OutOutputlistPrintFabricationtime  
oFatXml = oFactory.GetClassObject("PrintFabricationtime")  
Dim strpath As String  
strpath = oFatXml.PrintListToXML(matpot)
```