

Alle startklar?

Sind folgende „Tools“ bei allen installiert?

- SQL Server
- Visual Studio mit BimlExpress
- BimlStudio (mit gültigem Key)
- AdventureWorks

PASS Essential

Rom wurde nicht an einem Tag erbaut!
(aber die hatten auch kein Biml)



Wer ist heute hier?

Was habt ihr mit Biml gemacht?

(hoffentlich nichts – ist ja ein Essential 😊!)

Programmiererfahrungen?

SSIS Erfahrungen?

Warum seid ihr heute hier?


Agenda

- Wer ist heute hier?
- Was ist Biml?
- Erstellen und Beladen einer einfachen Staging Umgebung
- Wie kann mein Biml Code besser werden?
 - CallBimlScript
 - Include
 - Codefiles
- Diskussion: Metadaten-Modelle
 - Was sollte hinein?
 - Wie kann man sie abbilden?
 - Wie kann man sie warten?
- Steuerung einer Solution über Metadaten – „Mein Biml Framework“

Agenda – Auf der Tonspur

- Und was ist mit Flatfiles?
- Wie sieht das für SSAS aus?
- Build Automation
- Dokumentation
- Source Control
- Weitere Ansatzpunkte mit T-SQL
- Reverse Engineering
- Fragen (die sind aber auch zwischendurch erlaubt 😊)

Wer bin ich?

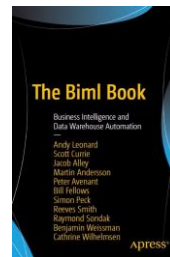
- Ben Weissman, Solisyon, Nürnberg
-  @bweissman
- b.weissman@solisyon.de
- SQL Server seit Version 6.5
- Data Passionist



Certified Data Vault Modeler



Data Science
Big Data
Artificial Intelligence
Data Analysis



Microsoft
CERTIFIED
Solutions Associate
Machine Learning

Microsoft
CERTIFIED
Solutions Expert
Data Management and Analytics



PASS Essential



Der Data Platform Podcast mit Biml Ben, Mr. T und Angry Frank



Adaptive Query Processing ADF Azure Azure Data Studio **Azure Notebooks** Azure Stack Big Data Clusters **Biml** Black Panther Business Application Summit 2018 Data Platform **Data Platform Summit** dbatools **Docker** Flensburger Radler Alkoholfrei GDPR **Git Hub** Ignite Jupyter Notebooks Kubernetes **Las Vegas** Lissabon Microsoft Professional Program MPP **PASS Camp** PASS Deutschland e.V. PASS Essentials PASS Summit **Power BI** PowerShell Query Folding Regionalgruppen Solo SQL Management Studio 18 - Preview SQL Operations Studio **SQL Saturday** SQL Server 2019 Tabular Tomb Raider **tSQLt** TugalT Visual Studio Code **WDC**



Ben Weissman
Biml Ben



Tillmann Eitelberg
Mr. T



Frank Geisler
Angry Frank



11

Episoden



2396

Downloads



1147

Sendeminuten



12

Gäste

<https://www.pleasetalkdatatome.de>

Was ist eigentlich Biml?

- Biml ist eine Markup Sprache – also: XML
- Erfunden/Entwickelt von Varigence
- So sieht Biml aus:

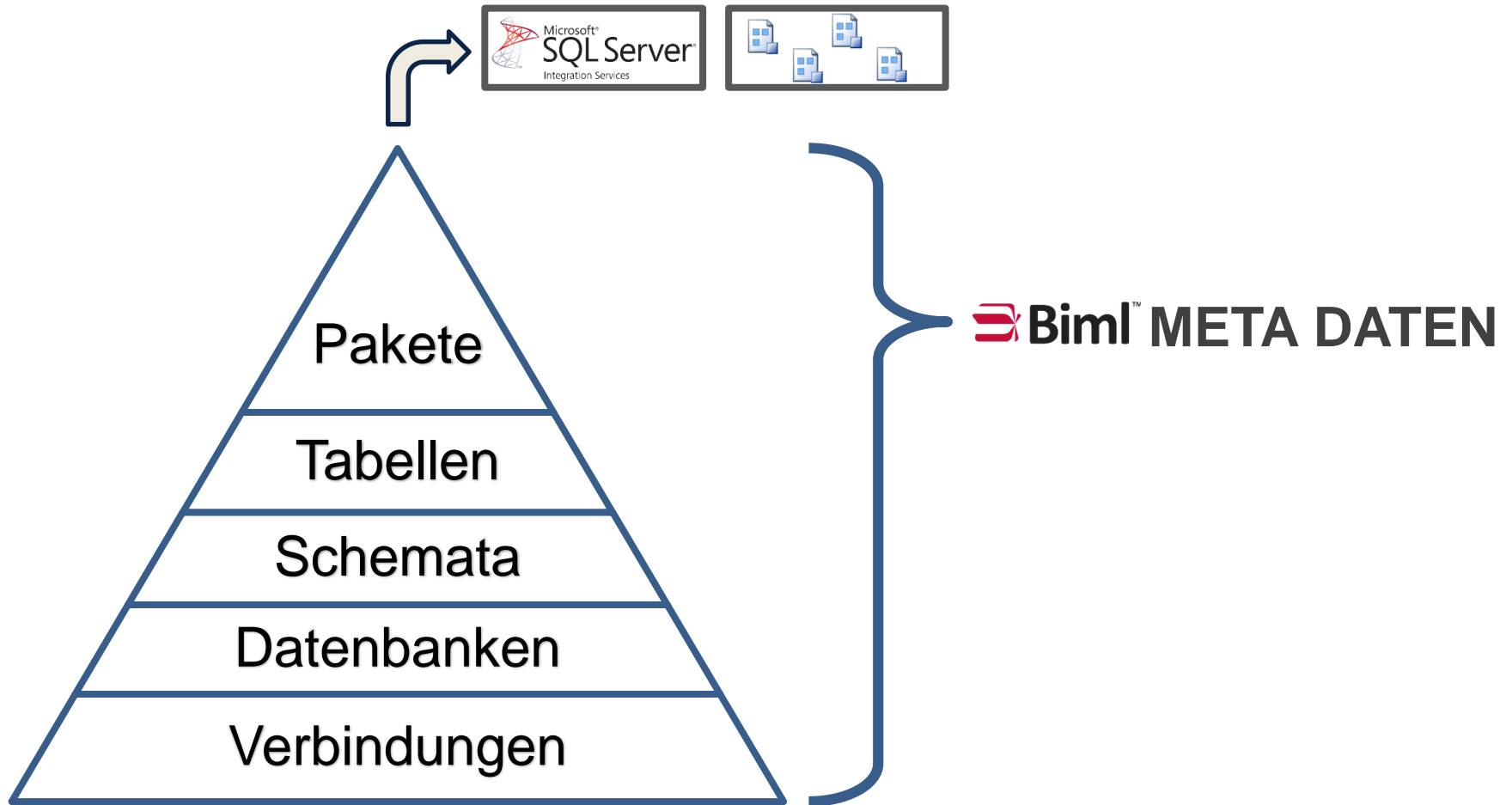
```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">  
  <Packages>  
    <Package Name="HelloBiml"/>  
  </Packages>  
</Biml>
```
- Wird erst in Verbindung mit BimlScript/APIs mächtig
- Kann neben SSIS auch für T-SQL und SSAS genutzt werden
- Verschiedenste Frontends

Das heißt so viel, wie...?

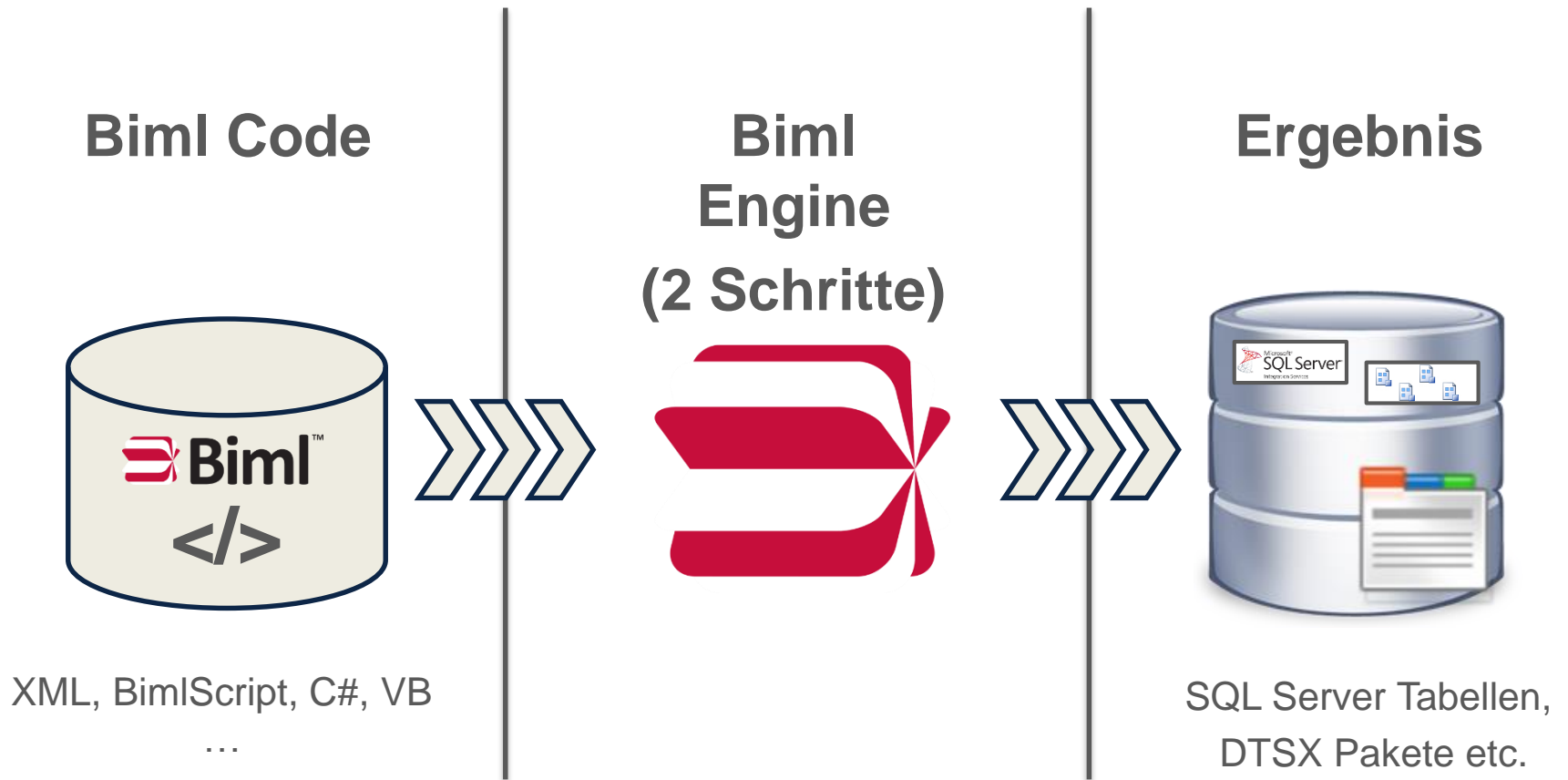
- Kompatibilität zwischen SQL Versionen wird belanglos
- Gesteigerte Produktivität durch Zeitersparnis bei “dummen” Datenbeladungen – Suche nach Mustern!
- Je besser ein Task standardisierbar ist (z.B. Beladung einer Staging Umgebung), desto höher der Zeitgewinn
- Biml ist der “3D Drucker für SSIS”



Basis Konstrukt



Wie funktioniert das dann?



Frontends

 **BimlOnline**

 **BimlExpress**

 **BimlStudio**

Biml Code

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
```

```
  <Packages>
```

```
    <Package Name="Mein_Paket"></Package>
```

```
    <Package Name="Mein_anderes_Paket"/>
```

```
  </Packages>
```

```
</Biml>
```

Biml Code: Das Root Element

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
```

```
  <Packages>
```

```
    <Package Name="Mein_Paket"></Package>
```

```
    <Package Name="Mein_anderes_Paket"/>
```

```
  </Packages>
```

```
</Biml>
```

Biml Code: Collections

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
```

```
  <Packages>
```

```
    <Package Name="Mein_Paket"></Package>
```

```
    <Package Name="Mein_anderes_Paket"/>
```

```
  </Packages>
```

```
</Biml>
```

Biml Code: Objekte

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
```

```
  <Packages>
```

```
    <Package Name="Mein_Paket"></Package>
```

```
    <Package Name="Mein_anderes_Paket"/>
```

```
  </Packages>
```

```
</Biml>
```


Biml Code: Attribute

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
```

```
  <Packages>
```

```
    <Package Name="Mein_Paket"></Package>
```

```
    <Package Name="Mein_anderes_Paket"/>
```

```
  </Packages>
```

```
</Biml>
```

Biml Code: Kurze und lange Syntax

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
```

```
  <Packages>
```

```
    <Package Name="Mein_Paket"></Package>
```

```
    <Package Name="Mein_anderes_Paket"/>
```

```
  </Packages>
```

```
</Biml>
```

Let's write some basic Biml...

CODING!!

Spannend wirds mit BimlScript!

- `<# Code Block #>`
- `<#+ Module Level Code Block #>`
- `<#= Inline Code Block #>`
 - Ruft die .NET ToString() Methode auf der im Block enthaltenen Expression auf
 - Kann kein Statement, void oder NULL sein
- `<#@ Directive #>`

Was wir uns schon einmal zusätzlich merken...

- Alle **Biml Objekte** folgen dem Namensschema Ast----Node
- Rootnode = Biml Meta Store
- Nicht das Tiering vergessen!

- Wichtige Methoden/Funktionen:
 - GetBiml
 - GetTag
 - ImportDB
 - GetDatabaseSchema
 - GenerateTableNodes

BimlScript

```
<#@ Template Language="VB" #>

<Biml xmlns="http://schemas.varigence.com/biml.xsd">
  <Packages>
    <# for each table as AstTableNode in RootNode.Tables #>
      <Package Name="Populate <#=table.Name#>"></Package>
    <# next #>
  </Packages>
</Biml>
```

BimlScript: Directive

```
<#@ Template Language="VB" #>
```

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
```

```
<Packages>
```

```
<# for each table as AstTableNode in RootNode.Tables #>
```

```
<Package Name="Populate <#=table.Name#>"></Package>
```

```
<# next #>
```

```
</Packages>
```

```
</Biml>
```

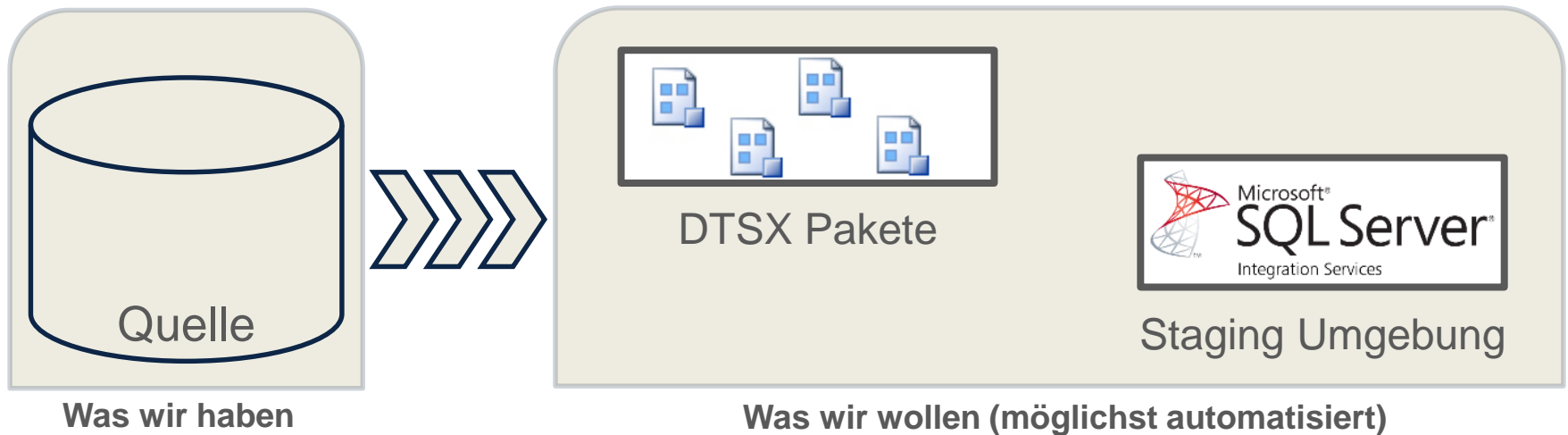
BimlScript: Control Nugget

```
<#@ Template Language="VB" #>  
  
<Biml xmlns="http://schemas.varigence.com/biml.xsd">  
  <Packages>  
    <# for each table as AstTableNode in RootNode.Tables #>  
      <Package Name="Populate <#=table.Name#>"></Package>  
    <# next #>  
  </Packages>  
</Biml>
```


BimlScript: Text Nugget

```
<#@ Template Language="VB" #>  
  
<Biml xmlns="http://schemas.varigence.com/biml.xsd">  
  <Packages>  
    <# for each table as AstTableNode in RootNode.Tables #>  
      <Package Name="Populate <#=table.Name#>"></Package>  
    <# next #>  
  </Packages>  
</Biml>
```

Erstellen und beladen einer einfachen Staging Umgebung



Erstellen und beladen einer einfachen Staging Umgebung

- Ziel: Übertragen / Erstellen aller Tabellen der Adventureworks Datenbank in einer neuen Datenbank AdventureStage
- Alle Zieltabellen bleiben im dbo Schema
- Der alte Schemaname wandert in den Tabellennamen
- Wir laden per Truncate / Full Load
- Code-Bausteine zum „merken“:
 - ImportOptions.ExcludeIdentity Or ImportOptions.ExcludePrimaryKey Or ImportOptions.ExcludeUniqueKey Or ImportOptions.ExcludeColumnDefault Or ImportOptions.ExcludeIndex Or ImportOptions.ExcludeCheckConstraint Or ImportOptions.ExcludeForeignKey
 - <OleDbConnection Name="Target"
ConnectionString="Provider=SQLNCLI11;Server=localhost;Initial Catalog=PASSEssential_Destination;Integrated Security=SSPI;" >

Wie kann man das besser machen?

- Was wir uns anschauen:
 - Include Files
 - Code Files
 - CallBimlScript
 - Macht mehr aus Annotations!
- „Auf der Tonspur“
 - Extension Methods

Warum immer das Rad neu erfinden: Include Files

- Denkt an „Suchen/Ersetzen“ – Biml ersetzt einfach 1:1 die Include Anweisung mit dem Inhalt der referenzierten Datei
- Möglicher Anwendungsfall: Globale Variablen-Datei
- Resultiert in keinerlei Output (sofern die referenzierte Datei keinen Output generiert)
- Akzeptiert keine Parameter
- Syntax:

```
<#@ include file="IncludedFile.biml" #>
```

Warum immer das Rad neu erfinden: CallBimlScript

- Quasi eine Funktion – Aufrufe mit Parameter resultiert in entsprechendem Rückgabecode (auch Rückgabe von Werten wäre möglich!)
- Praktisch wenn der gleiche Task mit verschiedensten Variablen oder in verschiedenen Projekten benötigt wird
- Resultiert immer in einem Output
- Akzeptiert Parameter
- Syntax:

```
<#=CallBimlScript("CallMeMaybe.biml", Parameter1, ...)#>
```

Warum immer das Rad neu erfinden: CallBimlScript

- Hinzufügen einer Extended Property zu einer Tabelle
- Tip: designerbimlpath deklarieren für Intellisense
- Syntax:

<#=CallBimlScript("Add_EP.biml", table,"BimlInitialized","True") #>

Add_EP.biml:

```
1 <#@ template language="VB" designerbimlpath="Biml/Packages/Package/Tasks" #>
2 <#@ property name="table" type="asttablenode" #>
3 <#@ property name="EPName" type="string" #>
4 <#@ property name="EPValue" type="string" #>
5 <ExecuteSQL ConnectionName="Target">
6   <DirectInput>
7     GO
8     IF EXISTS ( SELECT objname FROM fn_listextendedproperty(NULL, 'schema', '<#= table.schema #>', 'table', DEFAULT, NULL, NULL)
9       WHERE name = '<#= EPName #>' AND objtype = 'Table' and objname = N'<#= table.name #>')
10     EXEC sys.sp_dropextendedproperty @name=N'<#= EPName #>', @level0type=N'SCHEMA',
11     @level0name=N'<#= table.schema #>', @level1type=N'TABLE',@level1name=N'<#= table.name #>'
12     GO
13     EXEC sys.sp_addextendedproperty @name=N'<#= EPName #>', @value=N'<#= EPValue #>',
14     @level0type=N'SCHEMA',@level0name=N'<#= table.schema #>', @level1type=N'TABLE',@level1name=N'<#= table.name #>'
15   </DirectInput>
16 </ExecuteSQL>
```

Warum immer das Rad neu erfinden: Code Files

- Entwicklung von eigenen Klassen/Methoden/Funktionen in VB/C# mit direktem Kompilieren in der Biml Umgebung
- Unendlich viele Anwendungsfälle... Zum Beispiel:
 - Handling globaler Variablen (ja, hatten wir schon ☺)
 - Umsetzung von Namenskonventionen für Tabellen
 - ...
- Ergibt zunächst keinen Output – es handelt sich um managed Code, es macht also genau das, was man ihm sagt...
- Syntax:

```
<#@ code file="ThisIsNotTheCSharpYoureLookingFor.vb" #>
```


Warum immer das Rad neu erfinden: Code Files - Beispiel

- Naming Convention
- 3 Parameter:
 - Connection
 - Schema
 - Tabellenname
- Ergebnis:
 - Wenn das Schema „DBO“ ist:
 <Connection>_<Tabellenname>
 - Sonst:
 <Connection>_<Schema>_<Tabellenname>

Wie kann man das besser machen?

- Macht mehr aus Annotations!
- Extension Methods

Diskussion: Metadaten-Modelle

- Was sollte hinein?
- Wie kann man sie abbilden?
- Wie kann man sie warten?

Let's get meta driven...

Jetzt bauen wir all das in unsere Solution ein!

Anforderung:

- Meta-Daten Steuerung
 - Connections (Nur SQL Server Quellen)
 - Tabellen
- Naming Convention für Tabellen
- Beibehalten von Indizes
- Nur geänderte Tabellen deployen

Etwas Hilfestellung

```
Dim tmplist As New List(Of String)
Dim DT As DataTable =
ExternalDataAccess.GetDataTable(Conn.ConnectionString, SQL)
For Each dr As DataRow In dt.rows
    tmplist.add(dr.item(0).ToString())
Next
```

Flatfiles

- Jedes Flat File benötigt ein Format und eine Verbindung
- Die Verbindung ist der Verweis auf eine Datei
- Das Format beschreibt den Inhalt der Datei
- Mehrere Verbindungen können das gleiche Format nutzen vv.
- Dies ist besonders hilfreich, wenn man versucht über mehrere gleichartige Dateien zu loopen
- Tip: Versucht durch eine Format-Datei (XML/FMT) den Metadatenprozess zu automatisieren

SSAS

- Aufbau und Design von Cubes
 - Syntaxelemente
 - Nachteile
 - Metadaten
 - Pro-Feature
- Processing von Cubes
 - Syntaxelemente
 - Automatisierung - Heute bau ich mir eine Enterprise Edition

Build Automation (Pro Feature!)

- Variante 1: msbuild
- Variante 2: bimlc
 - ACHTUNG: --cleanoutputfolder- !!!
 - bimlc
 - cleanoutputfolder-
 - buildDocumentation-
 - v 2014
 - t C:\MyProject\output
 - s C:\MyProject\01_Environment.biml
 - s C:\MyProject\02_BuildTableMetadata.biml
 - s C:\MyProject\03_BuildStaging.biml

What else...

- Dokumentation
 - Manuell
 - In BimlStudio
- Source Control
- Weitere Ansatzpunkte für T-SQL etc.
- Reverse Engineering
 - Packages
 - Metadaten

Mehr

- Biml Tutorials
 - <https://www.bimlscript.com/>
- Andy Leonard
 - http://sqlblog.com/blogs/andy_leonard/archive/2016/06/02/so-you-want-to-learn-more-about-biml.aspx
- Solisyon Biml Blog
 - <http://Biml-Blog.de>

Und: das Biml Buch!

Gibts es noch...

Fragen?



Gerne auch im Nachgang
per Mail (b.weissman@solisyon.de)

oder

Twitter  (@bweissman)