noventum
IT MANAGEMENT CONSULTING

MENSCHEN
PROJEKTE
QUALITÄT

Freitag, 8. Mai 2015

Einführung in BIML, metadatengetriebene DWH-Entwicklung

noventum
IT Management Consulting

Felix Möller

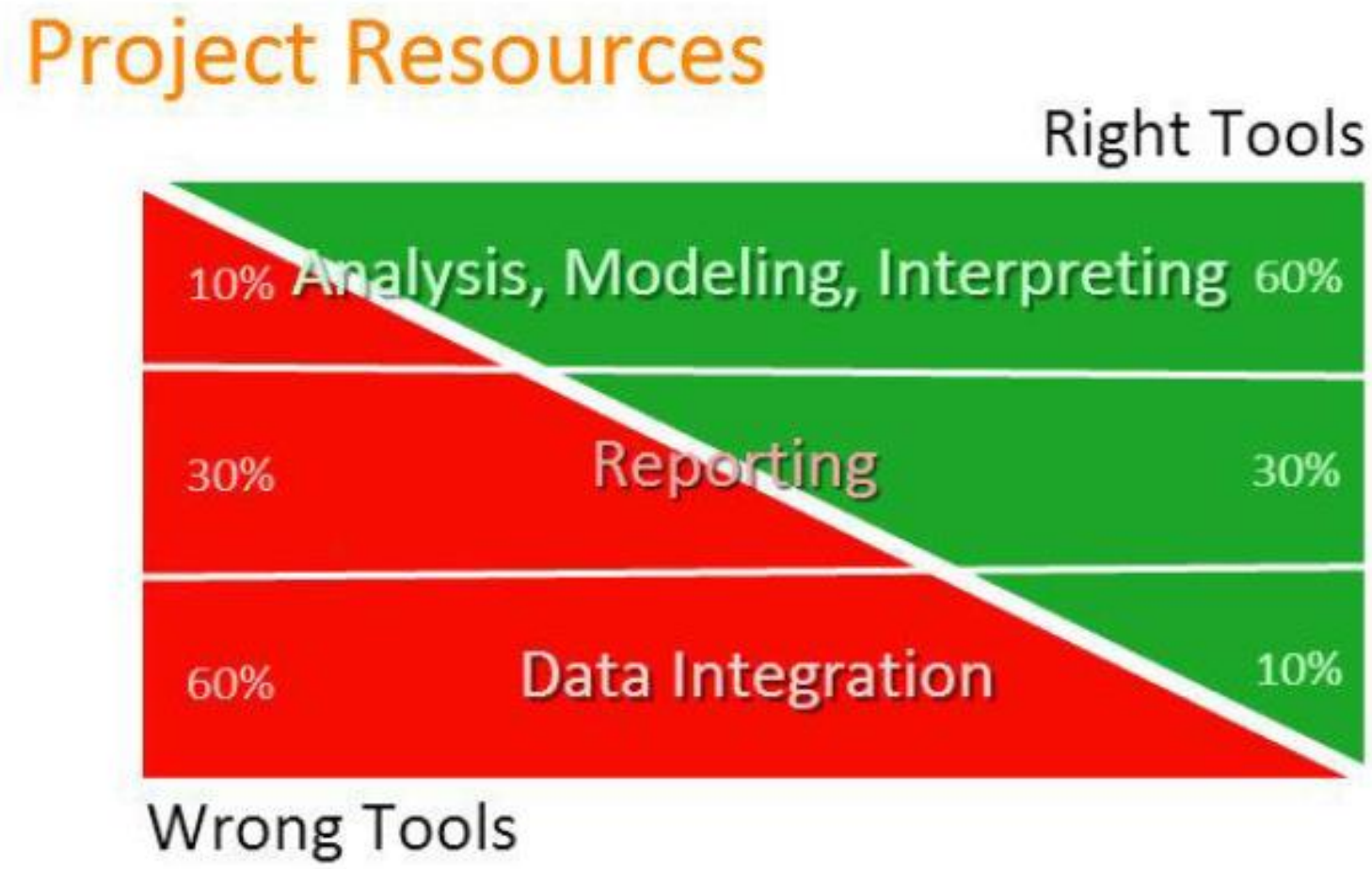
noventum
IT MANAGEMENT CONSULTING

Agenda

- Wieso BIML und was ist BIML?
- Erste Schritte
  - Biml
  - BimlScript
- Fortgeschrittene Schritte
  - Modularisierung
  - Arbeiten mit Metadaten
  - Integration C#
- Noventum Framework
- Wie fange ich an?

Wieso BIML? Herausforderungen bei DWH-Entwicklung mit SSIS (I)

- Problematische Wiederverwendbarkeit
- Wiederkehrende Patterns werden immer wieder erstellt
- Review von Unterschieden quasi unmöglich
- Einhalten von Namenskonventionen zeitraubend
- Unternehmensweite Standards zur Historisierung, Zeilenmetadaten, Logging, etc. müssen manuell eingehalten werden
- Nachträglich Verbesserungen an Patterns nur schwer möglich
- Anbinden von Datenquellen sehr repetitiv



Vgl. Schulungsmaterial Varigence Session 3

Was ist BIML?

- BIML ist eine XML-basierte Beschreibungssprache für SSIS-Pakete
- Alles was man mit SSDT BI machen kann, kann auch mit BIML gemacht werden
- Es gibt zwei Entwicklungsumgebungen für BIML
  - Mist kommerziell IDE von Varigence
  - Kostenlos BIML in BIDShelper
- BIML erlaubt sinnhafte Versionierung von SSIS-Paketen

Historie

- 2007 Projekt Vulcan innerhalb von Microsoft entstanden
- Als CodePlex-Projekt veröffentlicht https://vulcan.codeplex.com/
- 2008 Scott Currie gründet Varigence das Unternehmen hinter BIML
- 2011 der BIML Compiler ist kostenlos als Teil von BIDShelper 1.5.0 verfügbar
- 2015 mit BIDShelper 1.7.0 erscheint ein signifikantes Update auf dem Stand von Mist 4.0
  - Eigene C#-Klassen können eingebunden werden
  - Support für SQL Server 2014
  - Wesentlich schnellere Extraktion der Meta-Daten

Biml™

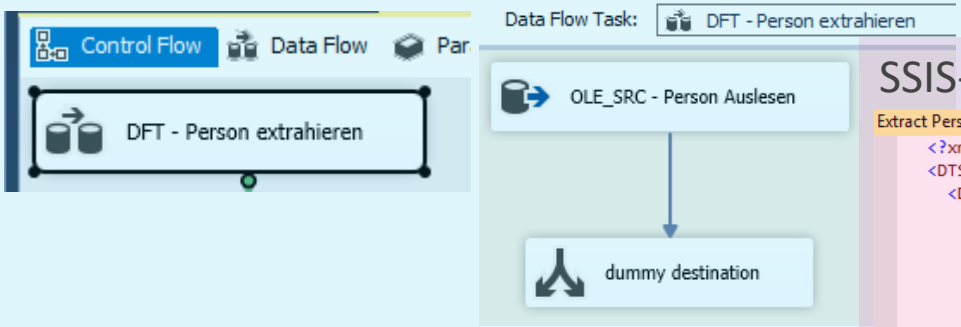
Varigence™ BUSINESS INTELLIGENCE

BI BIDS Helper

BimlScript

BIML vs SSIS-Quelltext

SSIS-Designer

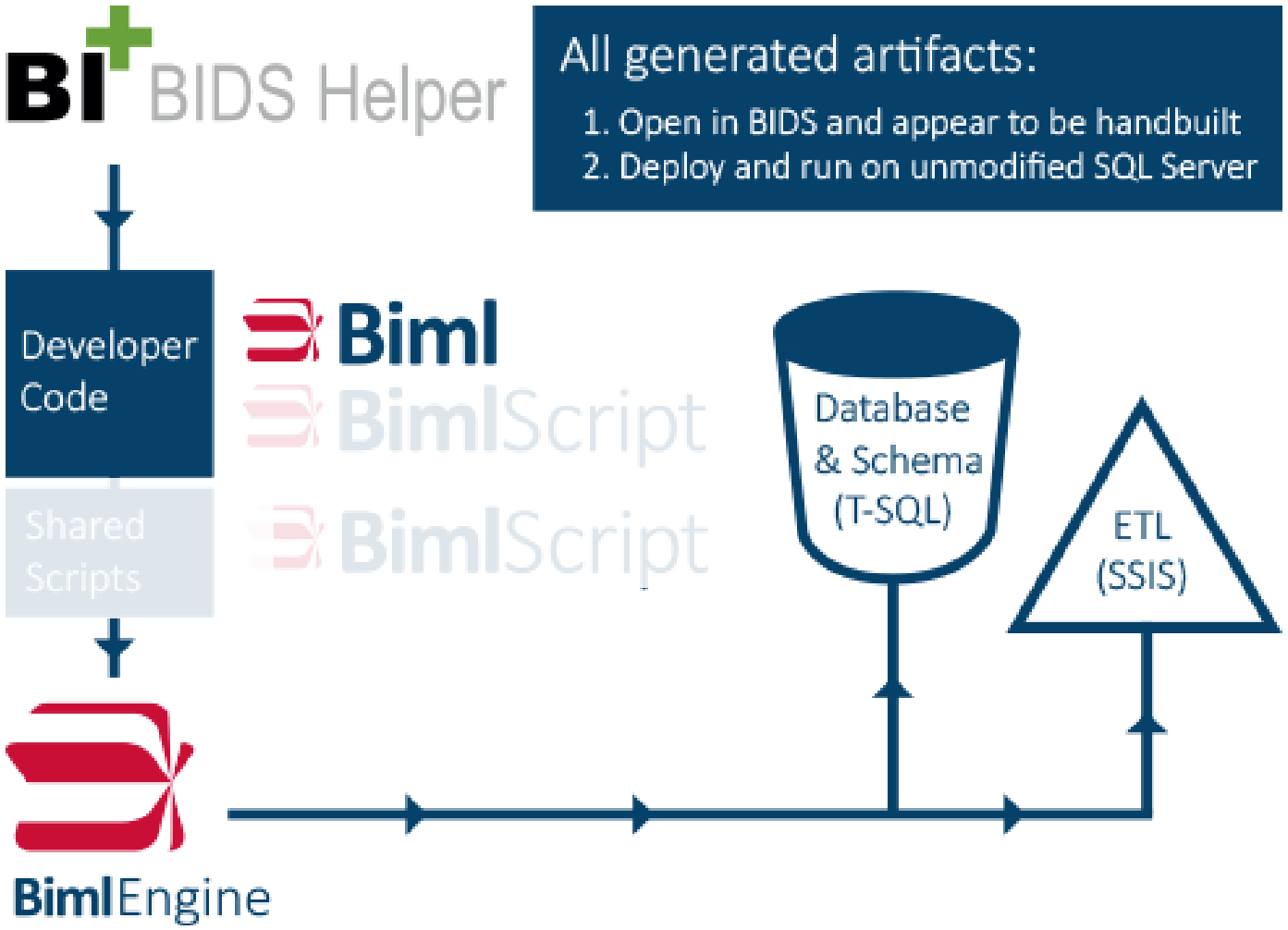



```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
  <Connections>
    <OleDbConnection Name="AdventureWorks" ConnectionString="D
  </Connections>
  <Packages>
    <Package Name="Extract Person" ProtectionLevel="EncryptSer
      <Tasks>
        <Dataflow Name="DFT - Person extrahieren">
          <Transformations>
            <OleDbSource Name="OLE_SRC - Person Auslesen" Conn
              <DirectInput>SELECT * FROM [Person].[Person]</Di
            </OleDbSource>
            <Multicast Name="dummy destination"></Multicast>
          </Transformations>
        </Dataflow>
      </Tasks>
    </Package>
  </Packages>
</Biml>
```

SSIS-Quelltext

Extract Person.dtsx [XML] | Extract Person.dtsx [Design] | bidibiml01ms.Adve...2014.conmgr [X

```
<?xml version="1.0" encoding="utf-8"?>
<DTS:Executable DTS:CreationDate="05/02/2015 15:32:02" DTS:CreationName="Mic
  <DTS:ConnectionManagers>
    <DTS:ConnectionManager DTS:CreationName="OLEDB" DTS:DTSID="{6BF277EA-AD2
      <DTS:ObjectData>
        <DTS:ConnectionManager DTS:ConnectionString="Data Source=bidibiml01m
        </DTS:ConnectionManager>
      </DTS:ObjectData>
    </DTS:ConnectionManager>
  </DTS:ConnectionManagers>
  <DTS:Executables>
    <DTS:Executable DTS:CreationName="Microsoft.Pipeline" DTS:DTSID="{95DD95
      <DTS:ObjectData>
        <pipeline BLOBTempStoragePath="" bufferTempStoragePath="" version="1
          <components>
            <component componentClassID="Microsoft.OLEDBSource" contactInfo="
              <connections>
                <connection connectionManagerID="{6BF277EA-AD26-401E-A002-53
                </connection>
              </connections>
              <outputs>
                <output name="Output" refId="Package\DFT - Person extrahieren
                  <externalMetadataColumns isUsed="True">
                    <externalMetadataColumn dataType="i4" name="BusinessEnti
                    </externalMetadataColumn>
                    <externalMetadataColumn dataType="wstr" length="2" name="
                    </externalMetadataColumn>
                    <externalMetadataColumn dataType="bool" name="NameStyle"
                    </externalMetadataColumn>
                    <externalMetadataColumn dataType="wstr" length="8" name="
                    </externalMetadataColumn>
                    <externalMetadataColumn dataType="wstr" length="50" name=
```



Vgl. http://bimlscript.com/GetStarted/AboutBimlScript

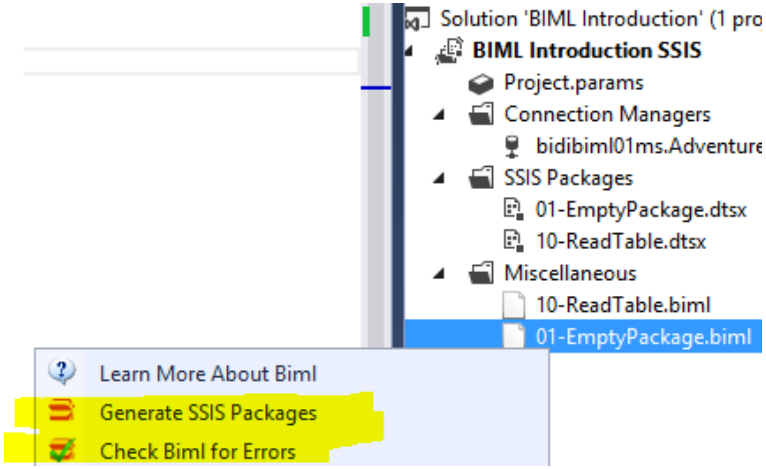
Live Demo – Erstellen des ersten Pakets

- Erstes Paket ohne jegliche Funktion

10-ReadTable.dtsx [Design] | 01-EmptyPackage.dtsx [Design] | 01-EmptyPackage.biml

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
  <Packages>
    <Package Name="01-EmptyPackage" ProtectionLevel="EncryptAllWithUserKey">
    </Package>
  </Packages>
</Biml>
```

- BIDShelper erzeugt zwei Einträge im Kontextmenü für BIML



Live Demo – Auslesen einer Tabelle

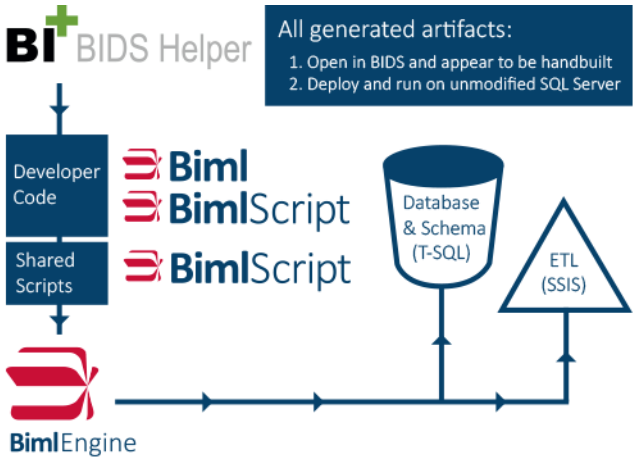
- Beim Erstellen des SSIS-Pakets mit BIDShelper muss die Zieltabelle schon existieren, damit die Datentypen etc erkannt werden können

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
  <Connections>
    <OleDbConnection Name="AdventureWorks" ConnectionString="Data Source=bidibiml01ms;Initial Ca
    <OleDbConnection Name="BIMLTarget" ConnectionString="Data Source=bidibiml01ms;Initial Catalog
  </Connections>
  <Packages>
    <Package Name="10-ReadTable" ProtectionLevel="EncryptSensitiveWithUserKey">
      <Tasks>
        <Dataflow Name="DFT - Person extrahieren">
          <Transformations>
            <OleDbSource Name="OLE_SRC - Person Auslesen" ConnectionName="AdventureWorks">
              <DirectInput>SELECT * FROM [Person].[Person]</DirectInput>
            </OleDbSource>
            <OleDbDestination Name="OLE_DST - dbo_Person" ConnectionName="BIMLTarget">
              <ExternalTableOutput Table="dbo.Person" />
            </OleDbDestination>
          </Transformations>
        </Dataflow>
      </Tasks>
    </Package>
  </Packages>
</Biml>
```

Einstieg in BimlScript

- BimlScript erlaubt es, in BIML-Dateien zu programmieren
  - Einsetzten von Variablen
  - If-Bedingungen, Foreach-Schleifen, etc.

- Syntax
  - Directives
    ```
    <#@ template tier = "10" #>
    ```
  - Control Blocks
    ```
    <# foreach(var t in ts) { #>
    ```
  - Expressions
    ```
    <#= table #>
    ```
  - Inline Code
    ```
    <#+ public IR R { ….} #>
    ```



Live Demo – Mehrere Tabellen übertragen

```
<Connections>
  <OleDbConnection Name="AdventureWorks" ConnectionString="Data Source=bidibiml01ms;Initial Cata
  <OleDbConnection Name="BIMLTarget" ConnectionString="Data Source=bidibiml01ms;Initial Catalog=
</Connections>
<Packages>
  <# var tables = new List<string> {"Person", "Address"}; #>
  <# foreach(var table in tables) { #>
  <Package Name="11-ReadTable_<#= table #>" ProtectionLevel="EncryptSensitiveWithUserKey">
    <Tasks>
      <Dataflow Name="DFT - <#= table #> extrahieren">
        <Transformations>
          <OleDbSource Name="OLE_SRC - <#= table #> Auslesen" ConnectionName="AdventureWorks">
            <DirectInput>SELECT * FROM [Person].[<#= table #>]</DirectInput>
          </OleDbSource>
          <OleDbDestination Name="OLE_DST - dbo_<#= table #>" ConnectionName="BIMLTarget">
            <ExternalTableOutput Table="dbo.<#= table #>" />
          </OleDbDestination>
        </Transformations>
      </Dataflow>
    </Tasks>
  </Package>
  <# } #>
</Packages>
</Biml>

<#@ template tier = "10" #>
```

Live Demo – Master-Paket erzeugen

- Wenn mehrere Pakete in SSDT BI markiert sind, werden diese in Reihenfolge der Markierung von der BimlEngine ausgewertet.
- Mit der Directive „template tier“ lässt sich dies beeinflussen

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
  <Packages>
    <Package Name="12-ReadTable_Master" ProtectionLevel="EncryptSensitiveWithUserKey">
      <Tasks>
        <# foreach (var package in RootNode.Packages) { #>
        <ExecutePackage Name="EPT - <#= package.SsisSafeName #>">
          <ExternalProjectPackage Package="<#=package.PackageFileName #>" />
        </ExecutePackage>
        <# } #>
      </Tasks>
    </Package>
  </Packages>
</Biml>

<#@ template tier = "20" #>
```

Live Demo – Modularisierung: Parameter in SSIS

- Es können Parameter verwendet werden. Es stehen auch die Standard SSIS Attribute e.g. System::PackageName zur Verfügung

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
  <Connections>
    <OleDbConnection Name="BIMLTarget" ConnectionString="Data Source=bidibiml01ms;Initial Catalog=BIMLTarget;Provi
  </Connections>
  <Packages>
    <Package Name="20-StarOfLogging" ProtectionLevel="EncryptSensitiveWithUserKey">
      <Tasks>
        <ExecuteSQL ConnectionName="BIMLTarget" Name="SQL - Ich war hier">
          <DirectInput>
            <![CDATA[INSERT INTO dbo.SsisLogging (message, package, tst) VALUES ('Ich war hier', ?, getdate())]]>
          </DirectInput>
          <Parameters>
            <Parameter Name="0" DataType="String" VariableName="System.PackageName"/>
          </Parameters>
        </ExecuteSQL>
      </Tasks>
    </Package>
  </Packages>
</Biml>
```

Live Demo – Modularisierung: CallBimlScript() (I)

- Es lassen sich Fragmente aus einem Biml-Dokument heraustrennen und so modularisieren
  - Caller

```
    </Dataflow>
    <#= CallBimlScript("21-StartOfLogging-Callee.biml") #>
</Tasks>
```

  - Callee

```
<ExecuteSQL ConnectionName="BIMLTarget" Name="SQL - Ich war hier">
  <DirectInput>
    <![CDATA[INSERT INTO dbo.SsisLogging (message, package, tst) VALUES ('Ich war hier', ?, getdate())]]>
  </DirectInput>
  <Parameters>
    <Parameter Name="0" DataType="String" VariableName="System.PackageName"/>
  </Parameters>
</ExecuteSQL>
```

Live Demo – Modularisierung: CallBimlScript() (II)

- Es lassen sich mit CallBimlScript auch Parameter übergeben.
- Diese müssen im Callee über die Directive „Property“ deklariert werden.

```
<ExecuteSQL ConnectionName="BIMLTarget" Name="SQL - Ich war hier">
  <DirectInput>
    <![CDATA[INSERT INTO dbo.SsisLogging (message, package, tst) VALUES ('<#= message #>', ?, getdate())]]>
  </DirectInput>
  <Parameters>
    <Parameter Name="0" DataType="String" VariableName="System.PackageName"/>
  </Parameters>
</ExecuteSQL>

<#@ property name="message" type="String" #>
```

Metadatenabzug

- Die BimlEngine kommt mit mehreren Funktionen zum Extrahieren von Metadaten (jeweils mit etlichen Signaturen)
  - ImportDB – importiert eine ganze Datenbank
  - ImportTableNodes – importiert mehrere Tabellen
  - GenerateTableNodes – importiert eine Tabellle
  - ImportSchema (seit 1.7.0) – kombiniert alle Möglichkeiten. Laut Ankündigung bis zu 500× schneller (1)

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
  <Connections>
    <OleDbConnection Name="AdventureWorks" ConnectionString="Data Source=bidibiml01m
  </Connections>
  <Packages>
    <#
      var sourceConnection = SchemaManager.CreateConnectionNode("AdventureWorks", "D
      var table = sourceConnection.GenerateTableNode("Person", "Person", ImportOptio
    #>
    <Package Name="30-2-SelectWithMetadata" ProtectionLevel="EncryptSensitiveWithUse
      <Tasks>
        <Dataflow Name="DFT - Person extrahieren">
          <Transformations>
            <OleDbSource Name="OLE_SRC - Person Auslesen" ConnectionName="AdventureW
              <DirectInput>SELECT <#= table.GetColumnList() #> FROM <#= table.Schema
            </OleDbSource>
            <Multicast Name="dummy destination"></Multicast>
          </Transformations>
        </Dataflow>
      </Tasks>
    </Package>
  </Packages>
</Biml>

<#@ import namespace="Varigence.Biml.CoreLowerer.SchemaManagement" #>
<#@ import namespace="System.Data" #>

<#@ template language="C#" #>
```

1: https://www.youtube.com/watch?v=dWhn67pTFBw

Metadaten in MessageBox

- Zum Debuggen ist es sehr hilfreich, während des Kompilierens Informationen auszugeben
  - MessageBox.Show()
  - Es kann nur auf Informationen aus vorherigen „template tiers“ zugegriffen werden

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
  <Connections>
    <OleDbConnection Name="AdventureWorks" ConnectionString="Data Source=bidibiml01ms;Initial Catalog=Adventure
  </Connections>
  <#
      var sourceConnection = SchemaManager.CreateConnectionNode("AdventureWorks", "Data Source=bidibiml01ms;Ini
      var results = sourceConnection.GetDatabaseSchema(new List<string> {"Person"}, new List<string> {"Person",
            ImportOptions.ExcludeCheckConstraint | ImportOptions.ExcludeColumnDefault | ImportOptions.ExcludePr
      var tables = results.TableNodes;
      foreach (var table in tables) {
          MessageBox.Show(table.GetBiml());
      }
   #>
</Biml>

<#@ import namespace="Varigence.Biml.CoreLowerer.SchemaManagement" #>
<#@ import namespace="System.Data" #>

<#@ template language="C#" #>
<#@ assembly name="C:\\Windows\\Microsoft.NET\\Framework\\v2.0.50727\\System.Windows.Forms.dll" #>
<#@ import namespace="System.Windows.Forms" #>
```

Metadaten persistieren

- Für die metadatengetriebene Entwicklung lassen sich die Metadaten automatisch abziehen und dann anreichern
- Diese Metadaten lassen sich in XML-Form wegschreiben

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
    <#
        XmlDocument xmlDoc = new XmlDocument();
        xmlDoc.LoadXml(RootNode.GetBiml());
        string sFile = @"C:\Users\Administrator\Desktop\biml\BIML Introduction\BIML Introduction\4
        xmlDoc.Save(sFile);
        #>
    </Biml>

<!--Directives:-->
<#@ template tier="9" #>
<#@ import namespace="Varigence.Biml.CoreLowerer.SchemaManagement" #>
<#@ import namespace="System.Xml" #>
```

Persistierte Metadaten Grundlage für metadatengetriebene Entwicklung

- Mit Annotation lassen sich beliebige Informationen an alle BIML-Elemente hängen.
- Diese können später in BimlScript ausgewertet werden.

```
<Tables>
  <Table Name="Address" SchemaName="AdventureWorks2014.Person">
    <Annotations>
      <Annotation Tag="TargetSchema">DIL</Annotation>
      <Annotation Tag="SourceSystem">AdventureWorks2014</Annotation>
      <Annotation Tag="DataType">md</Annotation>
      <Annotation Tag="targetTableName">Adressen</Annotation>
      <Annotation Tag="Layer">SRC</Annotation>
    </Annotations>
    <Columns>
      <Column Name="AddressID">
        <Annotations>
          <Annotation Tag="targetColumnName">Adressen_Key</Annotation>
          <Annotation Tag="SCDType">Key</Annotation>
        </Annotations>
      </Column>
      <Column Name="AddressLine1" DataType="String" Length="60">
        <Annotations>
          <Annotation Tag="targetColumnName">Adresszeile1</Annotation>
          <Annotation Tag="SCDType">2</Annotation>
        </Annotations>
      </Column>
```

Integration von C#-Dateien

- Seit BIDShelper 1.7.0 gibt es die „code file“ directive.
- Hiermit lassen sich C#-Dateien einbinden.

```
            <OleDbSource Name="OLE_SRC - Person Auslesen" ConnectionName="AdventureWorks">
              <DirectInput><#= BimlCSClass.getSelectStatement(table, 5) #></DirectInput>
            </OleDbSource>
            <Multicast Name="dummy destination"></Multicast>
          </Transformations>
        </Dataflow>
      </Tasks>
    </Package>
  </Packages>
</Biml>

<#@ import namespace="Varigence.Biml.CoreLowerer.SchemaManagement" #>
<#@ import namespace="System.Data" #>
<#@ code file = "51-BimlCSClass.cs" #>
```

Visual Studio für C#-Projekt

- Vollständiges Syntaxhightlighting und Autovervollständigung erfordert C#-Projekt
  - Es lässt sich Visual Studio für Windows Desktop verwenden
  - Projekte können sich überlappen

```
1  using Varigence.Languages.Biml.Table;
2  using Varigence.Biml.Extensions.SchemaManagement;
3
4  public static class BimlCSClass {
5      /// <summary>
6      /// Returns the number of specified rows from a table
7      /// </summary>
8      /// <param name="baseTable">Table for the from clause of the SELECT statement</param>
9      /// <param name="rows">Number of rows to return</param>
10     /// <returns></returns>
11     public static string getSelectStatement(AstTableNode baseTable, int rows) {
12         return string.Format("SELECT TOP {0} {1} FROM {2}", rows, baseTable.GetColumnList(
13     }
14 }
```

Demo – Teil des unseres Frameworks

- Streng nach Schichtenarchitektur
- Namenskonventionen
- Historisierung
  - Type1
  - Type2
  - Bitemporal
- Staging
  - Delta für Oracle-Datenbanken

Weitere Informationen

- Andy Leonard, Matt Masson, Tim Mitchell, Jessica Moss, Michelle Ufford: SQL Server 2012 Integration Services Design Patterns
- Blogs
  - Paul Te Braak: 25 Posts zu BIML https://paultebraak.wordpress.com/2015/01/01/biml-xv-foreach-record-in-the-dataset-control-flow/
  - Catherine Wilhelmsen: http://www.cathrinewilhelmsen.net/biml/
  - Andy Leonard: Stairway to BIML http://www.sqlservercentral.com/stairway/100550/
- Dokumentation
  - Sprach- und API-Dokumentation https://www.varigence.com/Documentation/Language/Index
  - Community Webseite: http://bimlscript.com/Develop/Resources
- Foren:
  - StackOverflow: http://stackoverflow.com/questions/tagged/biml

Software

- Beispiel SQL Server 2014
  - SQL Server 2014 – https://www.microsoft.com/en-us/evalcenter/evaluate-sql-server-2014
  - SQL Server Data Tools – Business Intelligence für Visual Studio 2013 – http://www.microsoft.com/de-de/download/details.aspx?id=42313
  - BIDShelper möglichst 1.7.0 – https://bidshelper.codeplex.com/
  - Visual Studio Express 2013 für Windows Desktop – https://www.visualstudio.com/en-us/products/visual-studio-express-vs.aspx
  - Tortoise – http://tortoisesvn.net/index.de.html

Alternativen

- Mist
  - Kommerzielle Entwicklungsumgebung mit Support für SSIS und SSAS
  - Live Editing, Transformers, uvm.
  - https://www.varigence.com/Mist (derzeit nur Internet Explorer)
- EzAPI
  - https://sqlsrvintegrationsrv.codeplex.com/releases/view/82369
  - Letzte Version mit Installer für 2008, Repository seit zwei Jahren ohne Veränderungen.
- …

Tricks für Syntax Highlighting in Visual Studio

- Direktiven immer ans Ende der Datei
- Code Nuggets immer als letzte Parameter
- Für CallBimlScript-Callees die Tags außerhalb des inkludierten Scopes auskommentieren
- In Visual Studio unter Tools -> Options -> Text Editor -> XML -> Formatting -> Auto Reformat die automatische Formatierung deaktivieren
- Ansonsten http://bimlscript.com/Develop probieren, oder die MistCloud Application installieren

VIELEN DANK FÜR IHRE AUFMERKSAMKEIT.

Fragen? Gerne!

Kontaktdaten

noventum
IT MANAGEMENT CONSULTING

FELIX MÖLLER
Senior Consultant

noventum consulting GmbH
Münsterstraße 111
D-48155 Münster

fon +49 2506 9302-0
fax +49 2506 9302-23
mobile +49 151 55113794
felix.moeller@noventum.de