Level Up Your Biml

Best Practices and Coding Techniques

Cathrine Wilhelmsen

@cathrinew

Data Community

SQL Day

GOLD SPONSORS

elitmind think bright

TECHNOLOGY INNOVATION DATA KNOWLEDGE tidk

VOLVO

Volvo Group IT

SILVER SPONSORS

dbWatch DATABASE CONTROL

it KONTRAKT®

UBS

VEEAM

BRONZE SPONSOR

KSIĄŻKI IT Z CAŁEGO ŚWIATA

NOVATECH

www.novatech.com.pl

STRATEGIC PARTNER

Microsoft

Session Description

Is your Biml solution starting to remind you of a bowl of tangled spaghetti code? Good! That means you are solving real problems while saving a lot of time. The next step is to make sure that your solution does not grow too complex and confusing – you do not want to waste all that saved time on future maintenance!

Attend this session for an overview of Biml best practices and coding techniques. Learn how to centralize and reuse code with Include files and the CallBimlScript method. Make your code easier to read and write by utilizing LINQ (Language-Integrated Queries). Share code between files by using Annotations and ObjectTags. And finally, if standard Biml is not enough to solve your problems, you can create your own C# helper classes and extension methods to implement custom logic.

Start improving your code today and level up your Biml in no time!

Cathrine Wilhelmsen

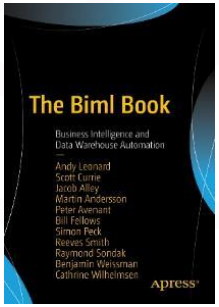
Business Intelligence Consultant

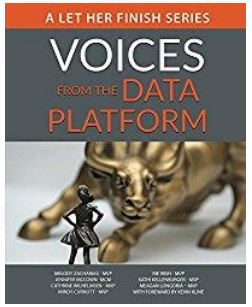
@cathrinew

cathrinew.net

Microsoft Most Valuable Professional

BimlHero CERTIFIED EXPERT





You…

Know basic Biml and BimlScript

Created a staging environment using Biml

Created a metadata-driven Biml framework

…?

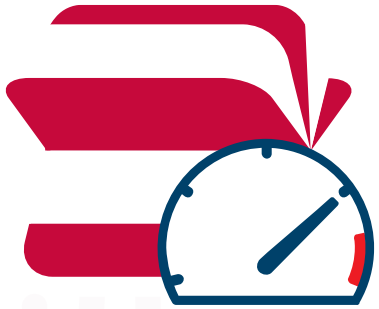
…the next 60 minutes…

Code Management

Practical Biml Coding

C# Classes and Methods

Biml Tools



BimlExpress

BimlStudio

Code Management

Don't Repeat Yourself

Centralize and reuse code

Update once in one file

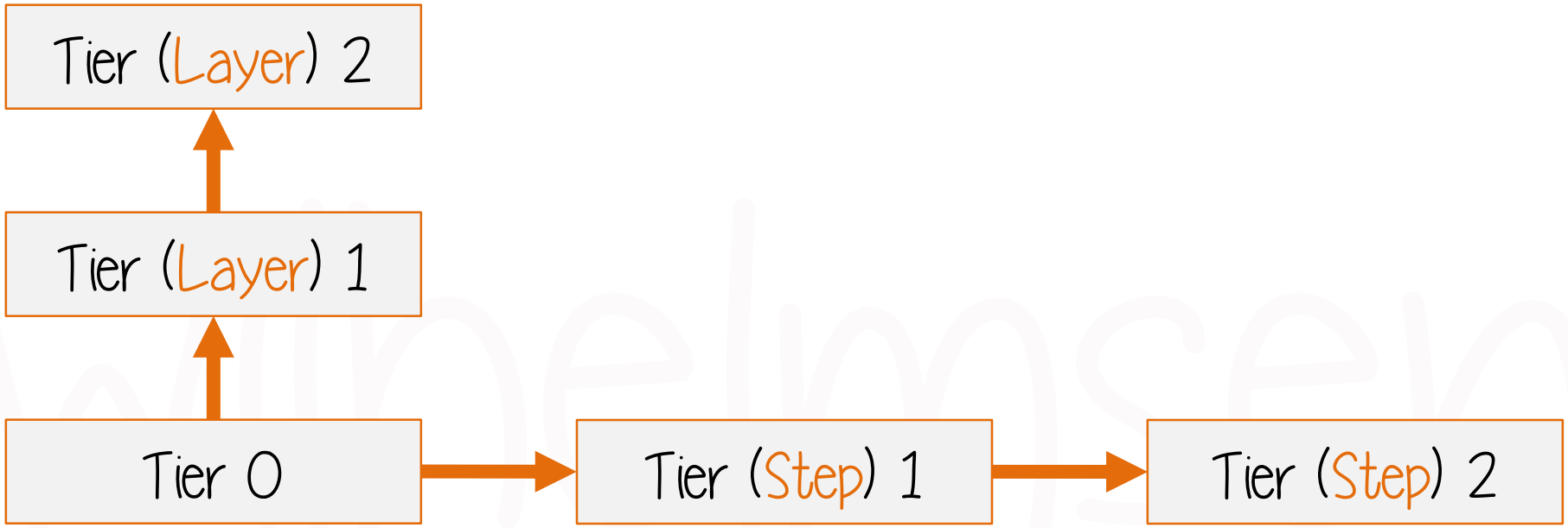
1. Tiered Biml Files
2. Include Files
3. CallBimlScript

Tiered Biml Files

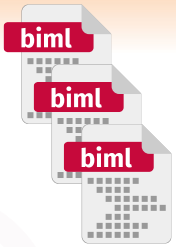
Multiple Biml Files Working Together

What are Tiered Biml Files?

Think of tiers as **stacked layers** or **sequential steps**



Tiered Biml Files



Split Biml code in multiple files to:

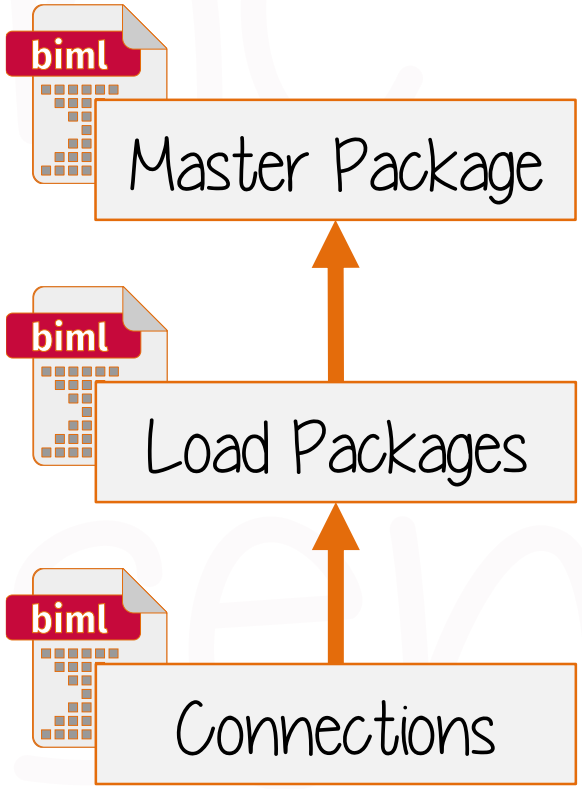
- Solve logical dependencies
- Build solutions in multiple steps behind the scenes

Specify the tier per file by using the template directive:

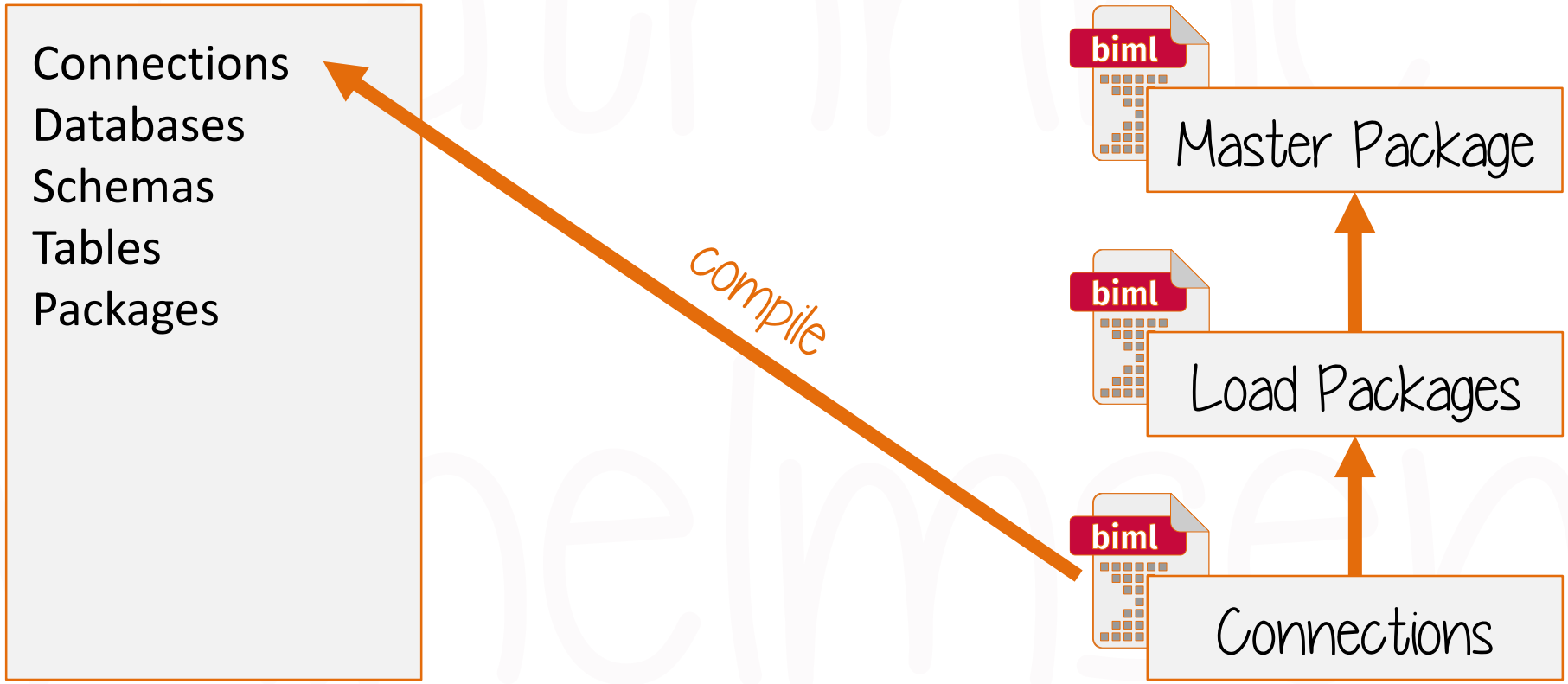
```
<#@ template tier="2" #>
```

Tiered Biml Files: Behind the Scenes

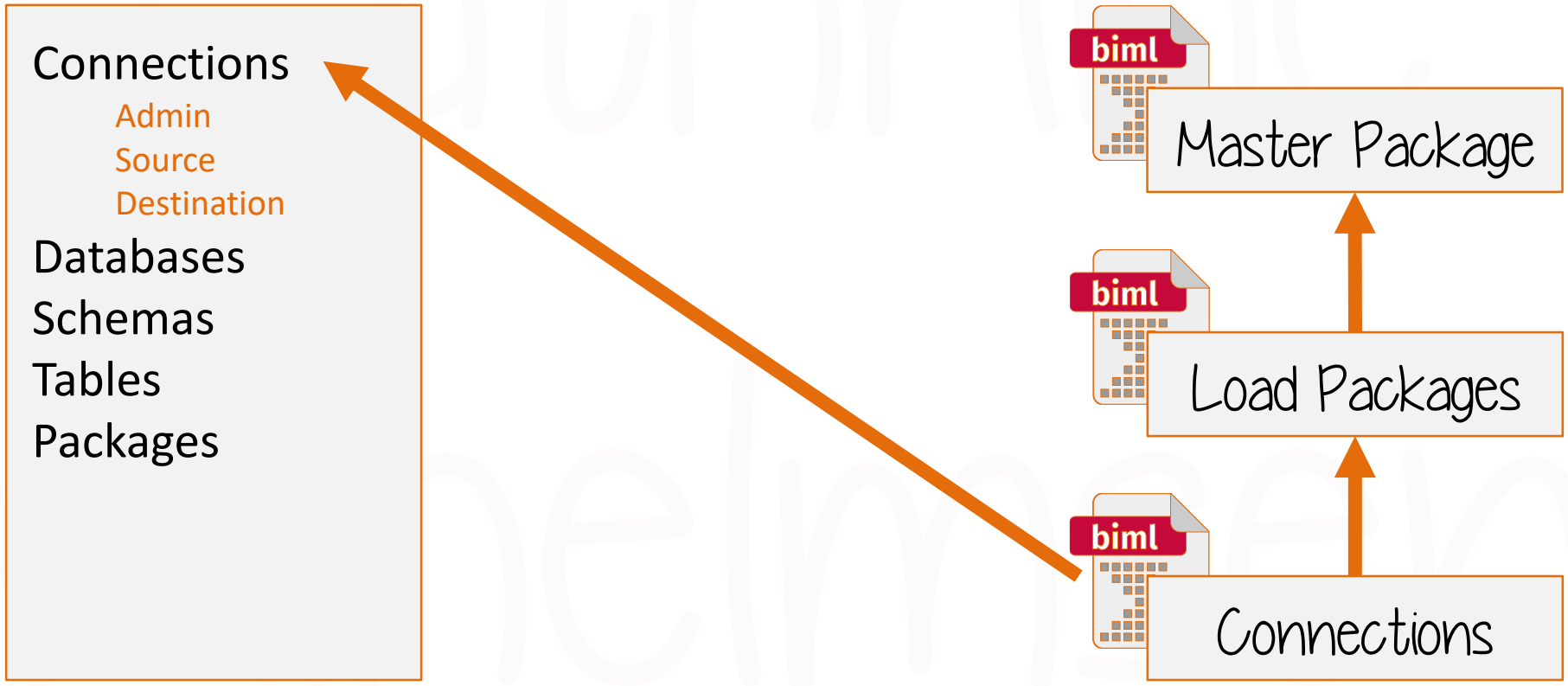
Connections
Databases
Schemas
Tables
Packages



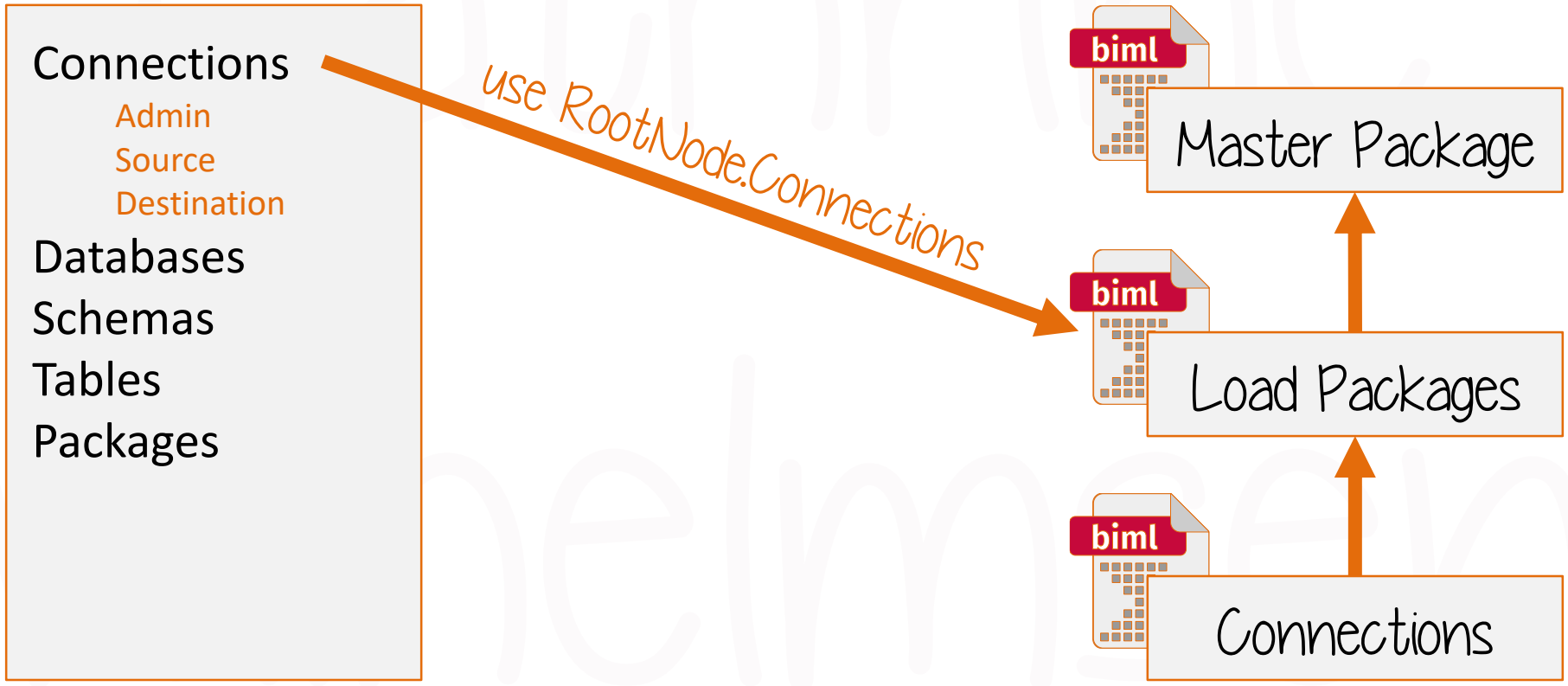
Tiered Biml Files: Behind the Scenes



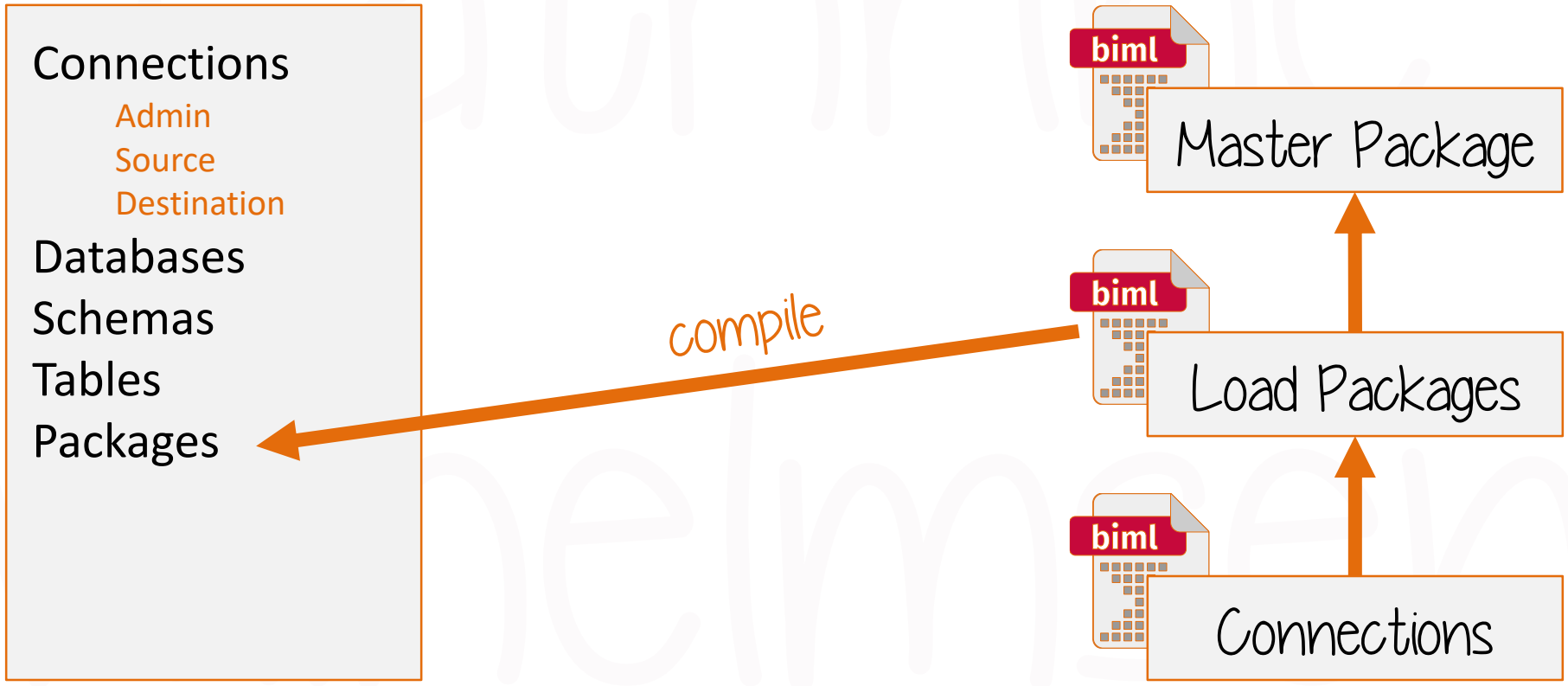
Tiered Biml Files: Behind the Scenes



Tiered Biml Files: Behind the Scenes

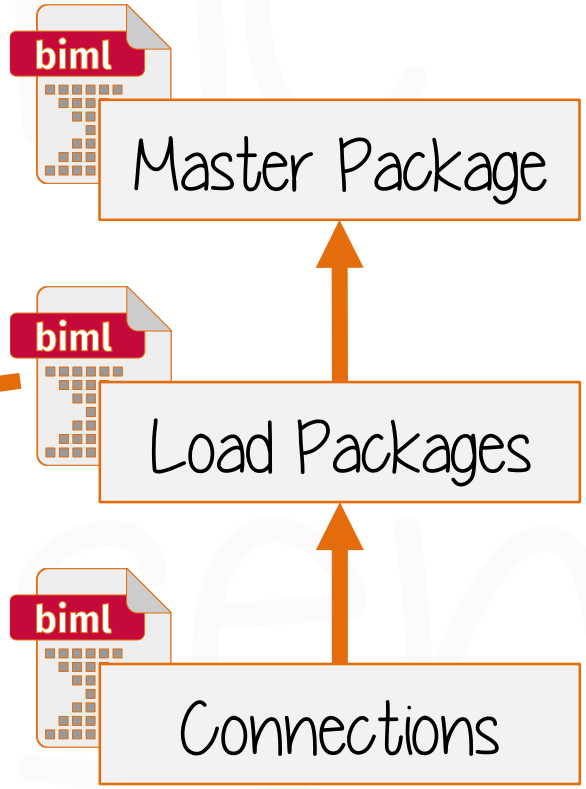


Tiered Biml Files: Behind the Scenes



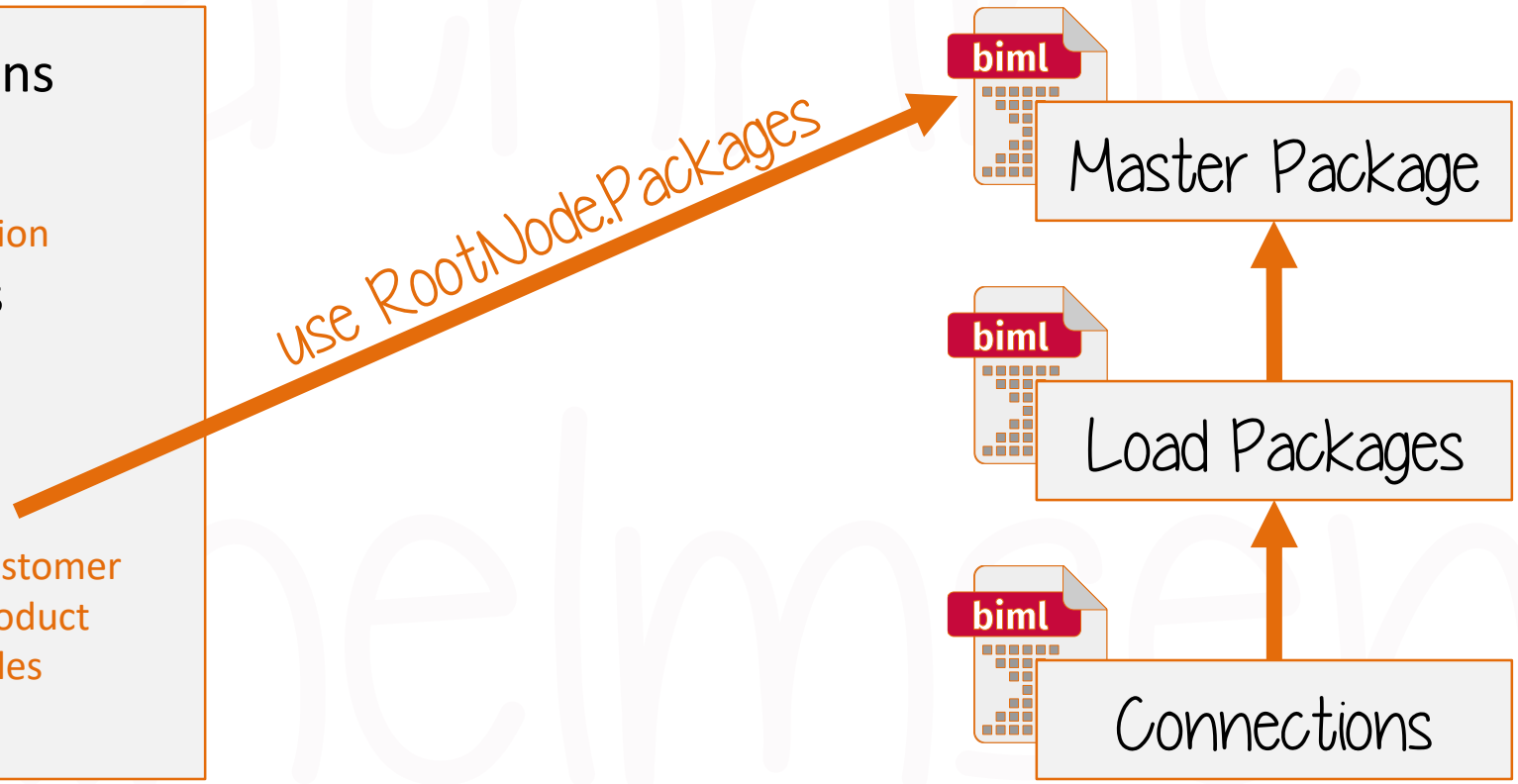
Tiered Biml Files: Behind the Scenes

- Connections
  - Admin
  - Source
  - Destination
- Databases
- Schemas
- Tables
- Packages
  - Load_Customer
  - Load_Product
  - Load_Sales

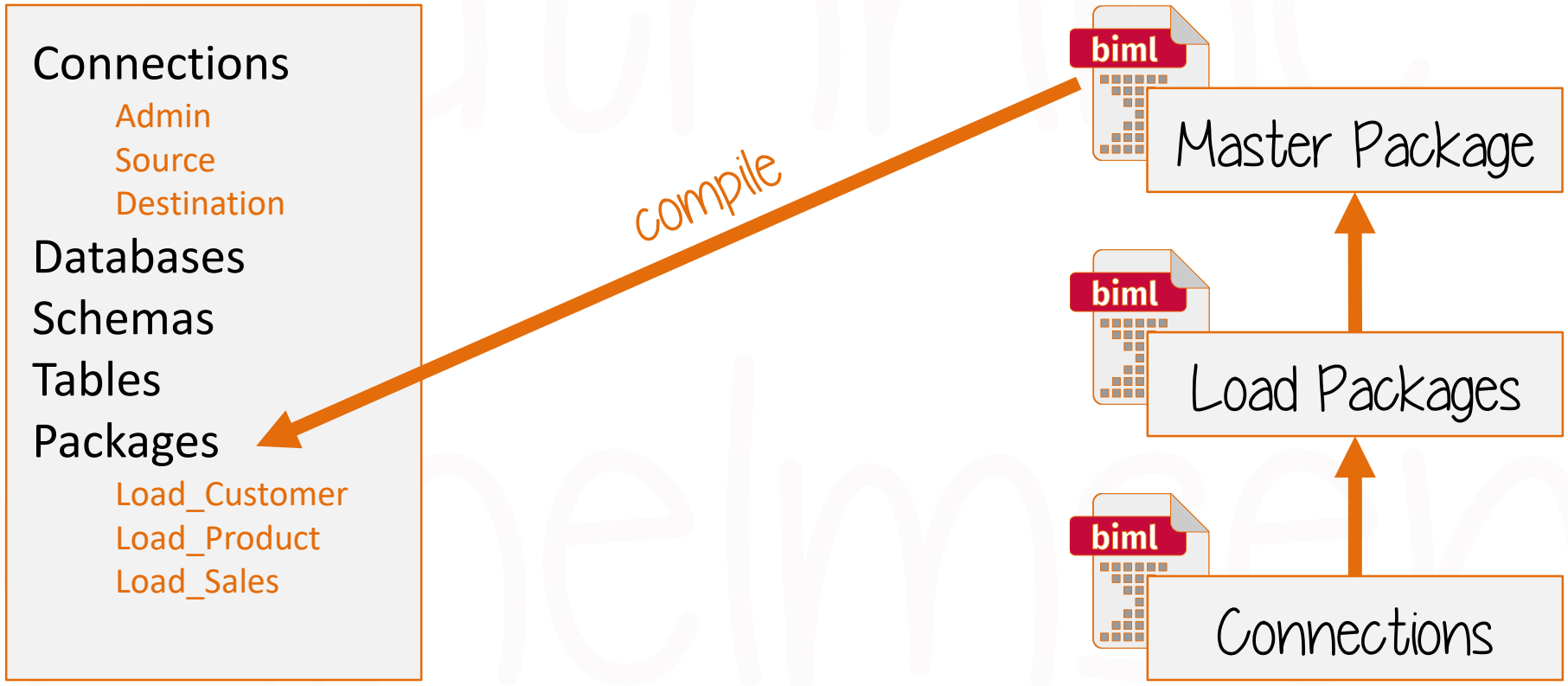


Tiered Biml Files: Behind the Scenes

- Connections
  - Admin
  - Source
  - Destination
- Databases
- Schemas
- Tables
- Packages
  - Load_Customer
  - Load_Product
  - Load_Sales

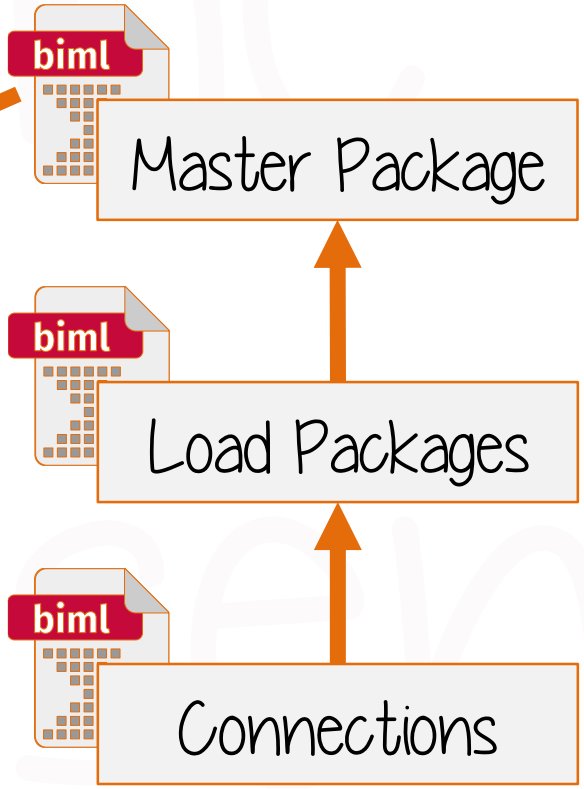


Tiered Biml Files: Behind the Scenes



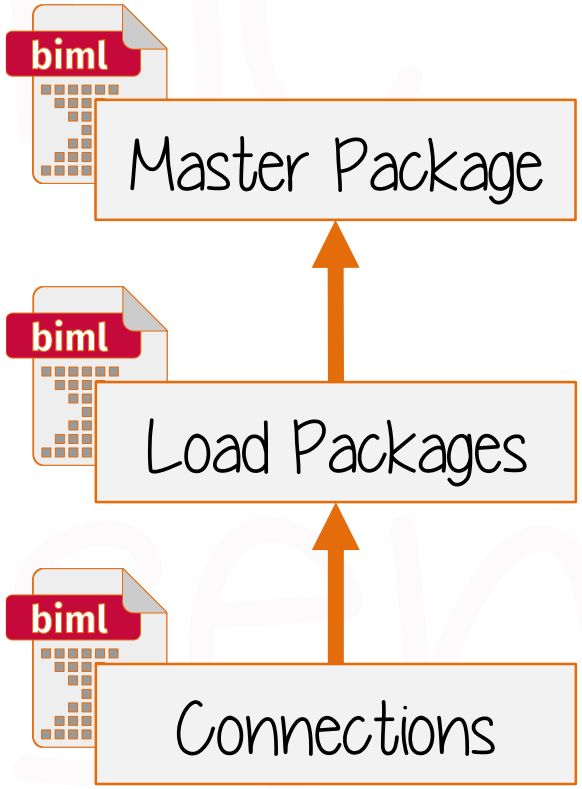
Tiered Biml Files: Behind the Scenes

- Connections
  - Admin
  - Source
  - Destination
- Databases
- Schemas
- Tables
- Packages
  - Load_Customer
  - Load_Product
  - Load_Sales
  - Master



Tiered Biml Files: Behind the Scenes

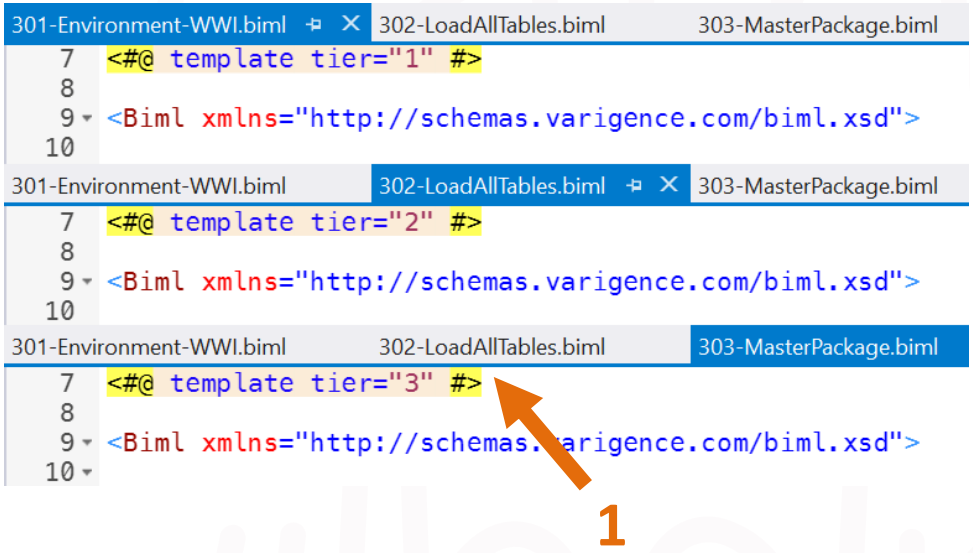
- Connections
  - Admin
  - Source
  - Destination
- Databases
- Schemas
- Tables
- Packages
  - Load_Customer
  - Load_Product
  - Load_Sales
  - Master

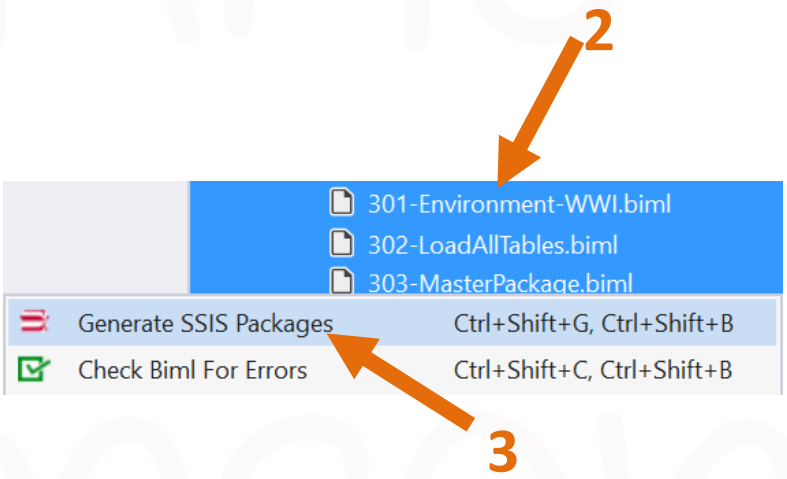


Tiered Biml Files: Behind the Scenes

- Connections
  - Admin
  - Source
  - Destination
- Databases
- Schemas
- Tables
- Packages
  - Load_Customer
  - Load_Product
  - Load_Sales
  - Master

How do you use Tiered Biml Files?



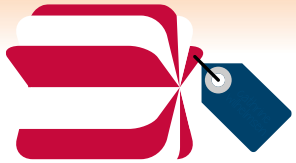


1. Create Biml files with specified tiers
2. Select all the tiered Biml files
3. Right-click and click Generate SSIS Packages

Annotations and ObjectTags

Store and Pass Metadata Between Biml Files

Annotations and ObjectTags



Annotations and ObjectTags are Key / Value pairs

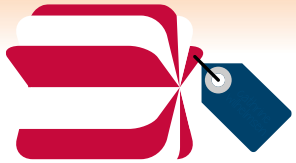
Annotations: String / String

ObjectTags: String / Object

Store metadata by attaching tags to Biml objects

Higher tier files can get tags from lower tier files

Annotations



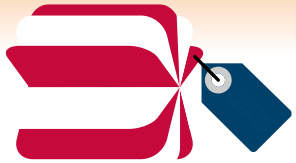
Create annotations:

```
<OleDbConnection Name="Dest">
  <Annotations>
    <Annotation Tag="Schema">stg</Annotation>
  </Annotations>
</OleDbConnection>
```

Use annotations:

```
RootNode.OleDbConnections["Dest"].GetTag("Schema");
```

ObjectTags



Create ObjectTags:

```
RootNode.OleDbConnections["Dest"].ObjectTag["Filter"]
= new List<string>{"Product","ProductCategory"};
```

Use ObjectTags:

```
RootNode.OleDbConnections["Dest"].ObjectTag["Filter"];
```

Include Files

Automated Copy & Paste

Include Files

Include common code in multiple files and projects

Can include many file types: .biml .txt .sql .cs

Use the include directive

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

The directive will be replaced by the included file

Works like an automated Copy & Paste

Include Files

Include Files

Include Files

Global Include Files

Automated Copy & Paste Everywhere

Global Include Files

Use the global directive

```
<#@ global #>
```

Use instead of adding include directive to all files to

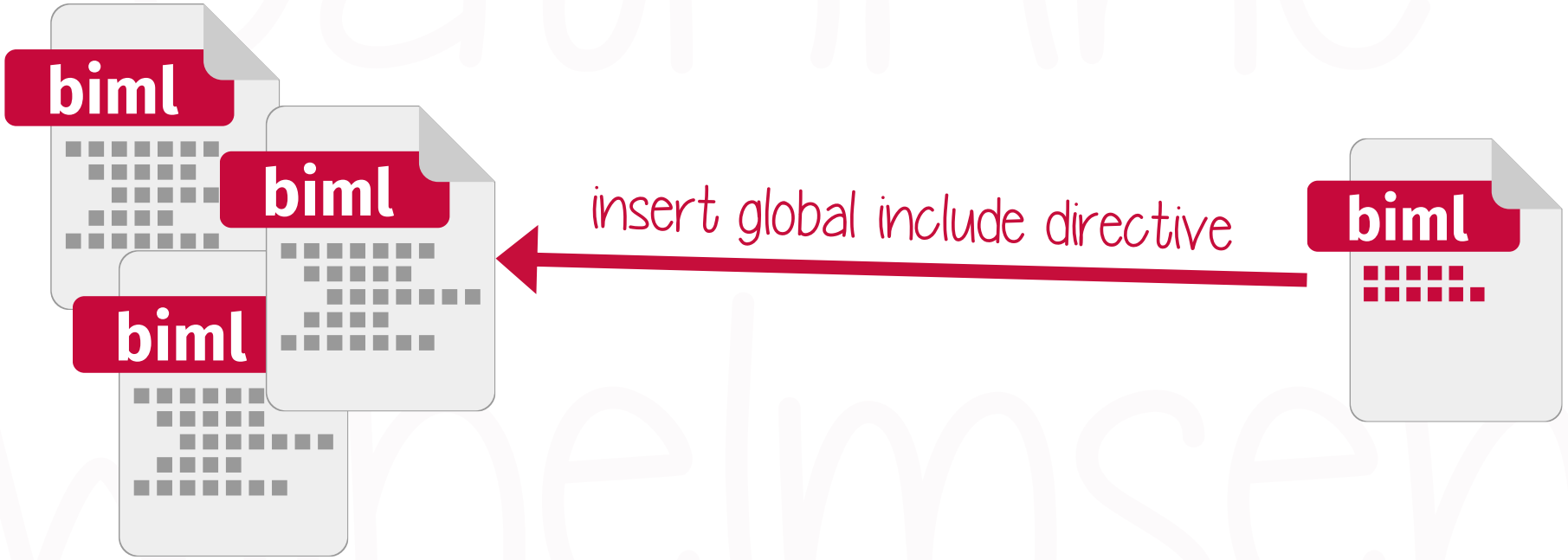
- Create global variables
- Include code files
- Make VB default language

Global Include Files

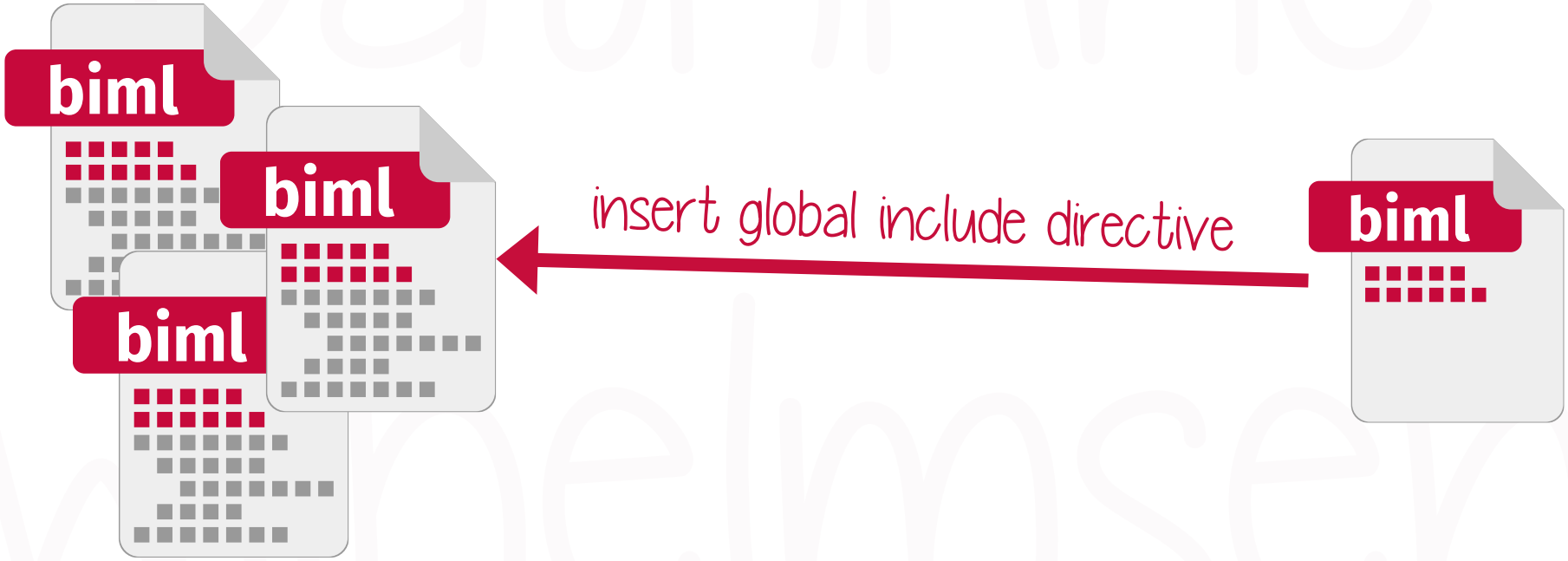
Specify attributes to control global include files:

```
<#@ global
    order="1"
    location="top"
    active="true" #>
```

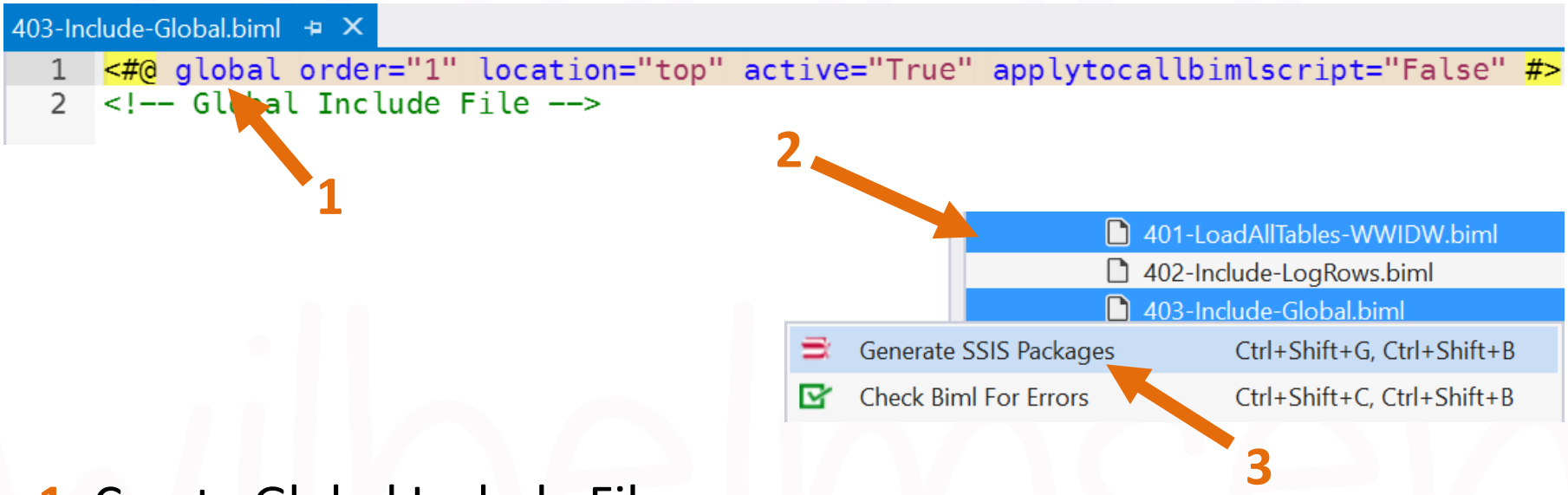
Global Include Files



Global Include Files



How do you use Global Include Files?



1. Create Global Include File
2. Select Biml files *and* Global Include File
3. Right-click and click Generate SSIS Packages

CallBimlScript

Parameterized Control over Returned Code

CallBimlScript

Control and limit the code returned

Works like a parameterized include (*or stored procedure*)

CallBimlScript file specifies accepted parameters:

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

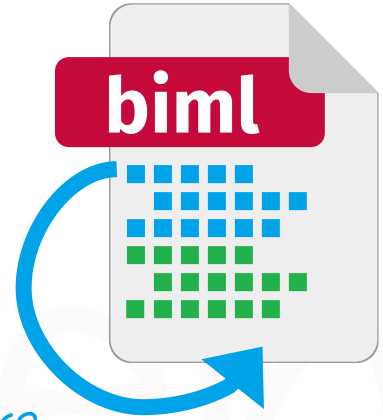
Main file calls and passes parameters:

```
<#=CallBimlScript("CommonCode.biml", Parameter)#>
```

CallBimlScript

CallBimlScript

CallBimlScript



logic based
on parameters

CallBimlScript

CallBimlScript

CallBimlScriptWithOutput

Parameterized Control over Returned Code and Objects

CallBimlScriptWithOutput

CallBimlScript file specifies custom output object:

```
<# CustomOutput.BimlComment = "Comment"; #>
```

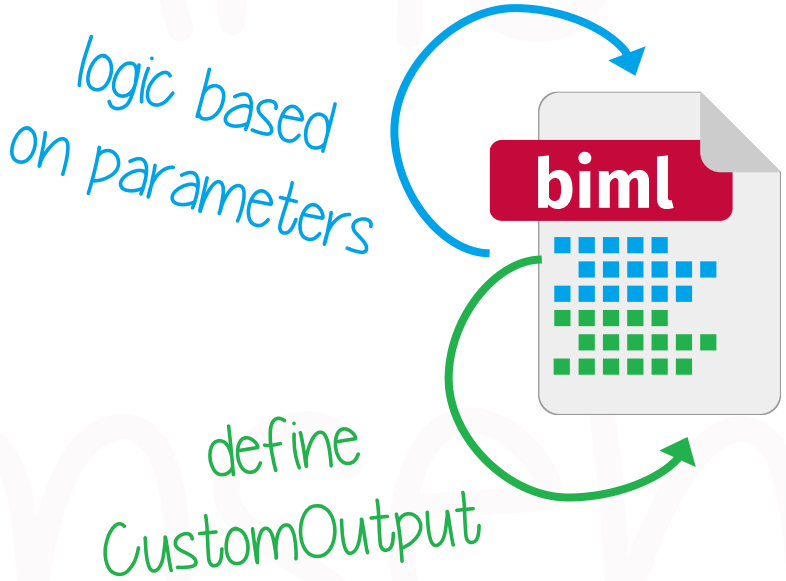
Main file defines, passes and uses output object:

```
<# dynamic outObj; #>
<#=CallBimlScriptWithOutput("CommonCode.biml",
   out outObj)#>
<#=outObj.BimlComment#>
```

CallBimlScriptWithOutput

CallBimlScriptWithOutput

CallBimlScriptWithOutput



CallBimlScriptWithOutput

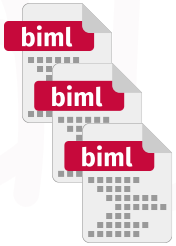
CallBimlScriptWithOutput

Wait! When do you use what?

In larger projects, you often see a combination of Tiered Biml Files, Include Files, and CallBimlScript

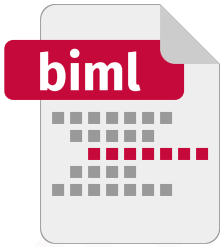
Rule of Thumb:

If you reuse code more than *3 times*, refactor so you *Don't Repeat Yourself*



Tiered Biml Files

Solve Dependencies
Multi-Step Builds



Include Files

Automated Copy & Paste
Code Included As-Is

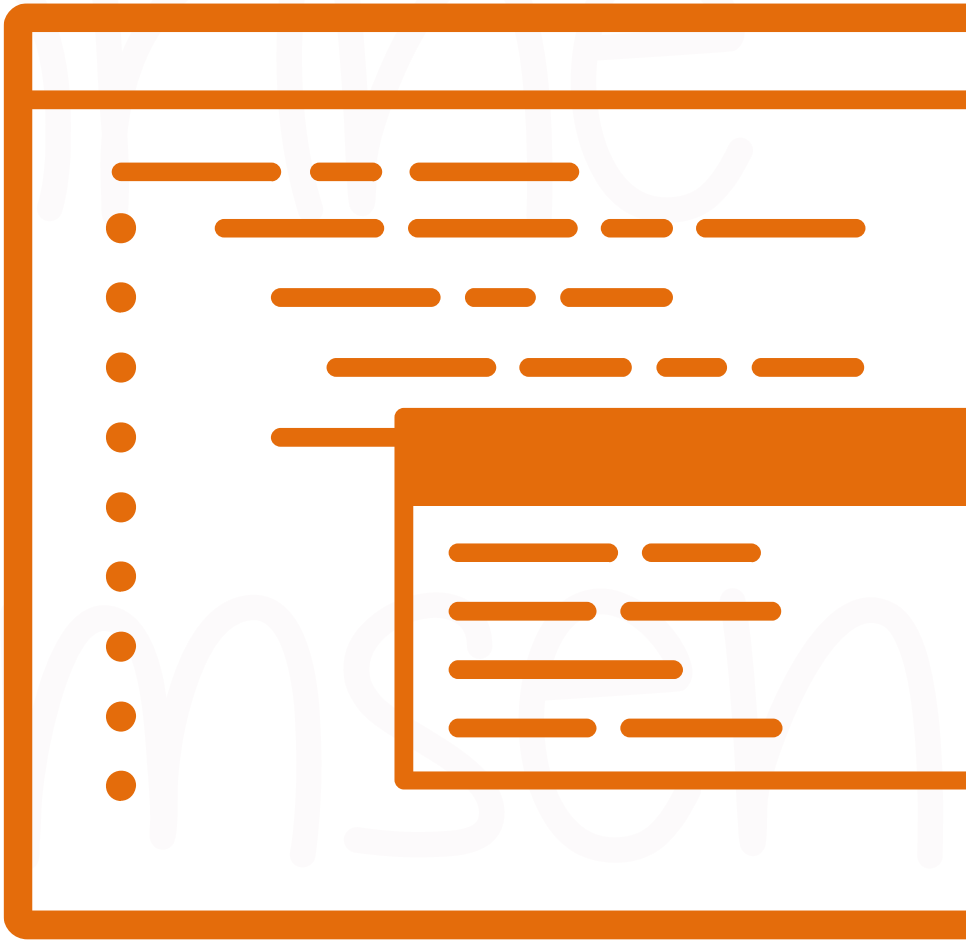
CallBimlScript

Parameterized
Control Returned Code

DEMO

Code Management

Practical Biml Coding



LINQ

LINQ (Language-Integrated Query)

One language to query:

SQL Server Databases

Datasets

Collections

Two ways to write queries:

SQL-ish Syntax

Extension Methods

LINQ Extension Methods

Filter	Sort	Aggregate	Group
Where, OfType	OrderBy, ThenBy	Count, Sum	GroupBy

Check Collections	Get Elements	Project Collections
All, Any, Contains	First, Last, ElementAt	Select, SelectMany

LINQ Extension Methods Example

```
var numConnections = RootNode.Connections.Count()

foreach (var table in RootNode.Tables.Where(…))

    if (RootNode.Packages.Any(…))
```

LINQ Extension Methods Example

But what do you put in here?

```
foreach (var table in RootNode.Tables.Where(…))

    if (RootNode.Packages.Any(…))
```

Lambda Expressions

"A lambda expression is an anonymous function that you can use to create delegates or expression tree types"

Lambda Expressions

...huh? o_O

Lambda Expressions

```
table => table.Name == "Product"
```

Lambda Expressions

```
table => table.Name == "Product"
```

The arrow is the lambda operator

Lambda Expressions

```
table => table.Name == "Product"
```

Input parameter is on the left side

Lambda Expressions

```
table => table.Name == "Product"
```

Expression is on the right side

LINQ and Lambda

Chain LINQ Methods and use Lambda Expressions for simple and powerful querying of collections:

```
.Where(table => table.Schema.Name == "Production")
        .OrderBy(table => table.Name)
```

LINQ: Filter collections

`Where()`

Returns the filtered collection with all elements that meet the criteria

```
RootNode.Tables.Where(t => t.Schema.Name == "Production")
```

`OfType()`

Returns the filtered collection with all elements of the specified type

```
RootNode.Connections.OfType<AstExcelOleDbConnectionNode>()
```

LINQ: Sort collections

`OrderBy()`

Returns the collection sorted by key…

```
RootNode.Tables.OrderBy(t => t.Name)
```

`ThenBy()`

…then sorted by secondary key

```
RootNode.Tables.OrderBy(t => t.Schema.Name)
               .ThenBy(t => t.Name)
```

LINQ: Sort collections

`OrderByDescending()`

Returns the collection sorted by key…

```
RootNode.Tables.OrderByDescending(t => t.Name)
```

`ThenByDescending()`

…then sorted by secondary key

```
RootNode.Tables.OrderBy(t => t.Schema.Name)
               .ThenByDescending(t => t.Name)
```

LINQ: Sort collections

`Reverse()`

Returns the collection sorted in reverse order

```
RootNode.Tables.Reverse()
```

LINQ: Group collections

`GroupBy()`

Returns a collection of key-value pairs where each value is a new collection

```
RootNode.Tables.GroupBy(t => t.Schema.Name)
```

LINQ: Aggregate collections

```
Count()
```

Returns the number of elements in the collection

```
RootNode.Tables.Count()
RootNode.Tables.Count(t => t.Schema.Name == "Production")
```

LINQ: Aggregate collections

`Sum()`

Returns the sum of the (numeric) values in the collection

```
RootNode.Tables.Sum(t => t.Columns.Count)
```

`Average()`

Returns the average value of the (numeric) values in the collection

```
RootNode.Tables.Average(t => t.Columns.Count)
```

LINQ: Aggregate collections

`Min()`

Returns the minimum value of the (numeric) values in the collection

```
RootNode.Tables.Min(t => t.Columns.Count)
```

`Max()`

Returns the maximum value of the (numeric) values in the collection

```
RootNode.Tables.Max(t => t.Columns.Count)
```

LINQ: Check collections

`All()`

Returns true if all elements in the collection meet the criteria

```
RootNode.Databases.All(d => d.Name.StartsWith("A"))
```

`Any()`

Returns true if any element in the collection meets the criteria

```
RootNode.Databases.Any(d => d.Name.Contains("DW"))
```

LINQ: Check collections

`Contains()`

Returns true if collection contains element

```
RootNode.Databases.Contains(AdventureWorks2014)
```

LINQ: Get elements

`First()`

Returns the first element in the collection (that meets the criteria)

```
RootNode.Tables.First()
RootNode.Tables.First(t => t.Schema.Name == "Production")
```

`FirstOrDefault()`

Returns the first element in the collection or default value (that meets the criteria)

```
RootNode.Tables.FirstOrDefault()
RootNode.Tables.FirstOrDefault(t => t.Schema.Name ==
"Production")
```

LINQ: Get elements

`Last()`

Returns the last element in the collection (that meets the criteria)

```
RootNode.Tables.Last()
RootNode.Tables.Last(t => t.Schema.Name == "Production")
```

`LastOrDefault()`

Returns the last element in the collection or default value (that meets the criteria)

```
RootNode.Tables.LastOrDefault()
RootNode.Tables.LastOrDefault(t => t.Schema.Name ==
"Production")
```

LINQ: Get elements

`ElementAt()`

Returns the element in the collection at the specified index

```
RootNode.Tables.ElementAt(42)
```

`ElementAtOrDefault()`

Returns the element in the collection or default value at the specified index

```
RootNode.Tables.ElementAtOrDefault(42)
```

LINQ: Project collections

`Select()`

Creates a new collection from one collection

A list of table names:

```
RootNode.Tables.Select(t => t.Name)
```

A list of table and schema names:

```
RootNode.Tables.Select(t => new {t.Name, t.Schema.Name})
```

LINQ: Project collections

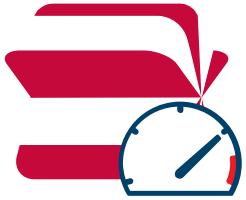
`SelectMany()`

Creates a new collection from many collections and merges the collections

A list of all columns from all tables:

```
RootNode.Tables.SelectMany(t => t.Columns)
```

The power is in the…

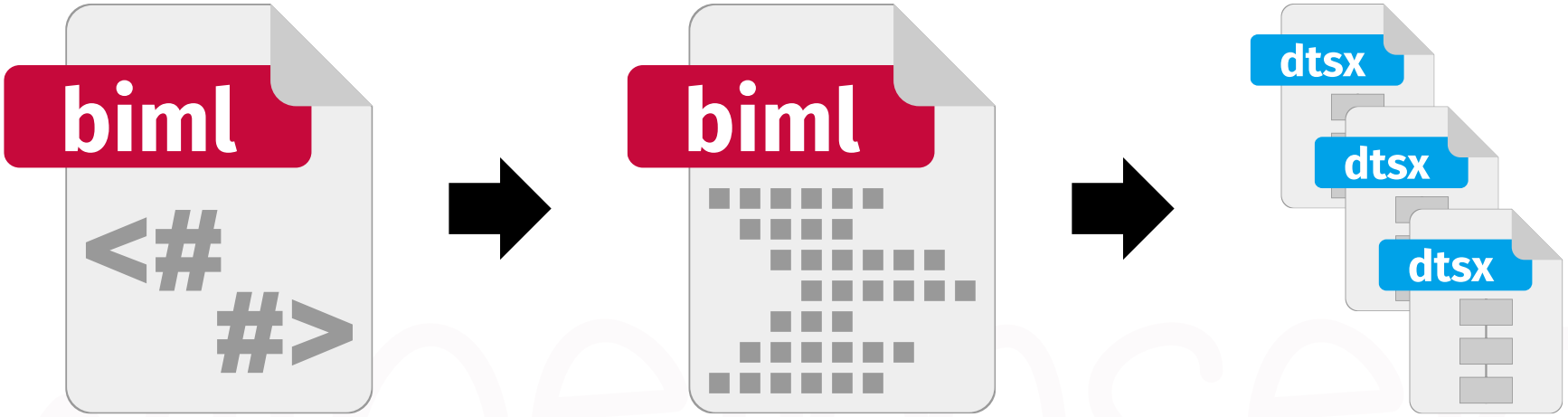


BimlExpress

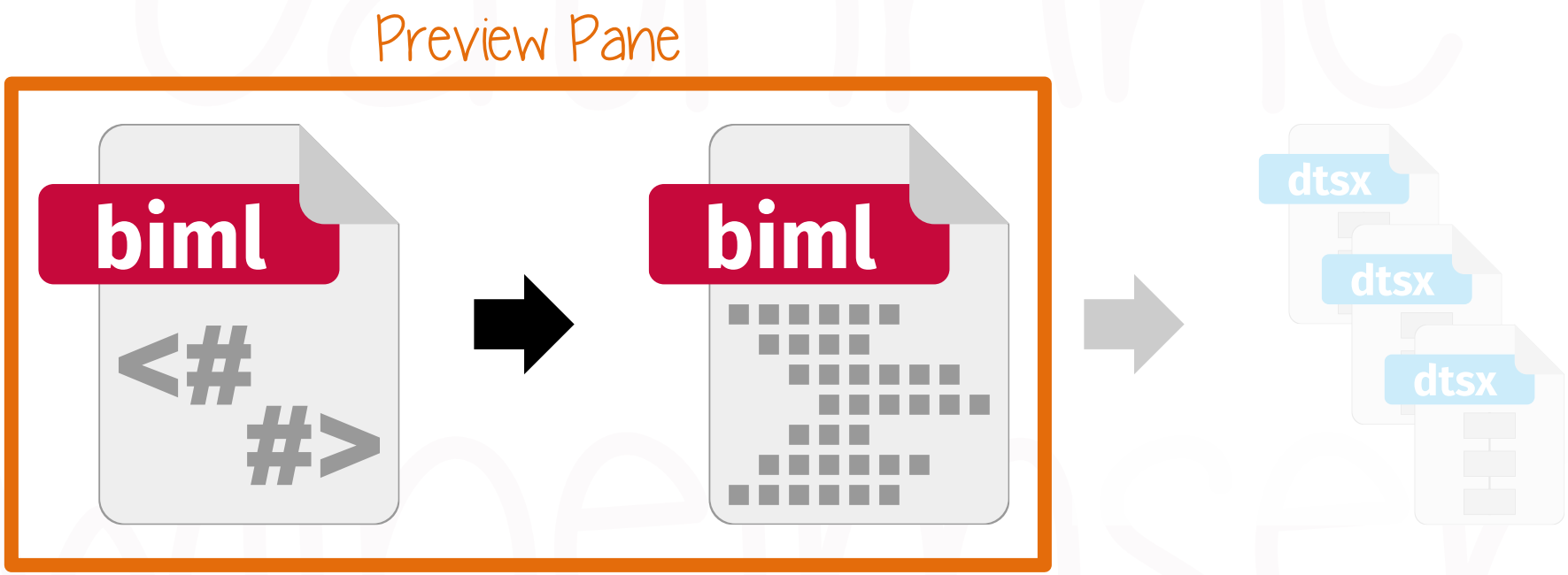
Preview Pane

BimlExpress Preview Pane

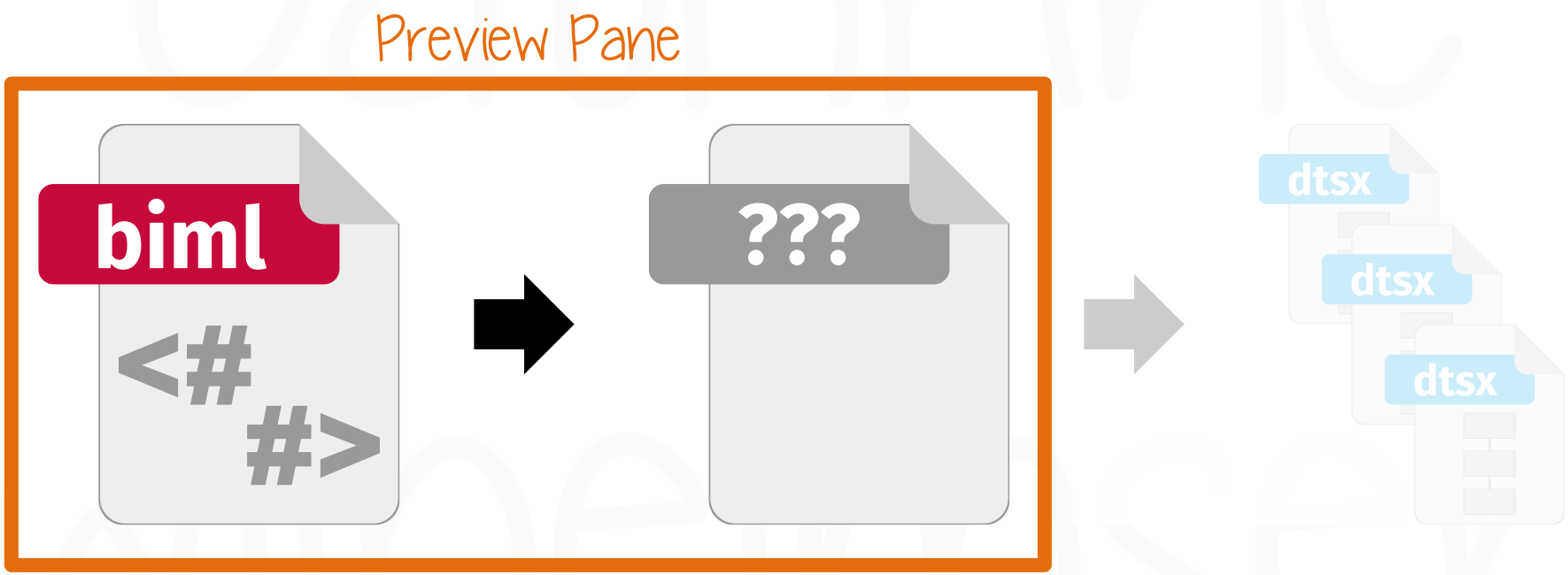
BimlScript to Biml to SSIS



BimlScript to Biml



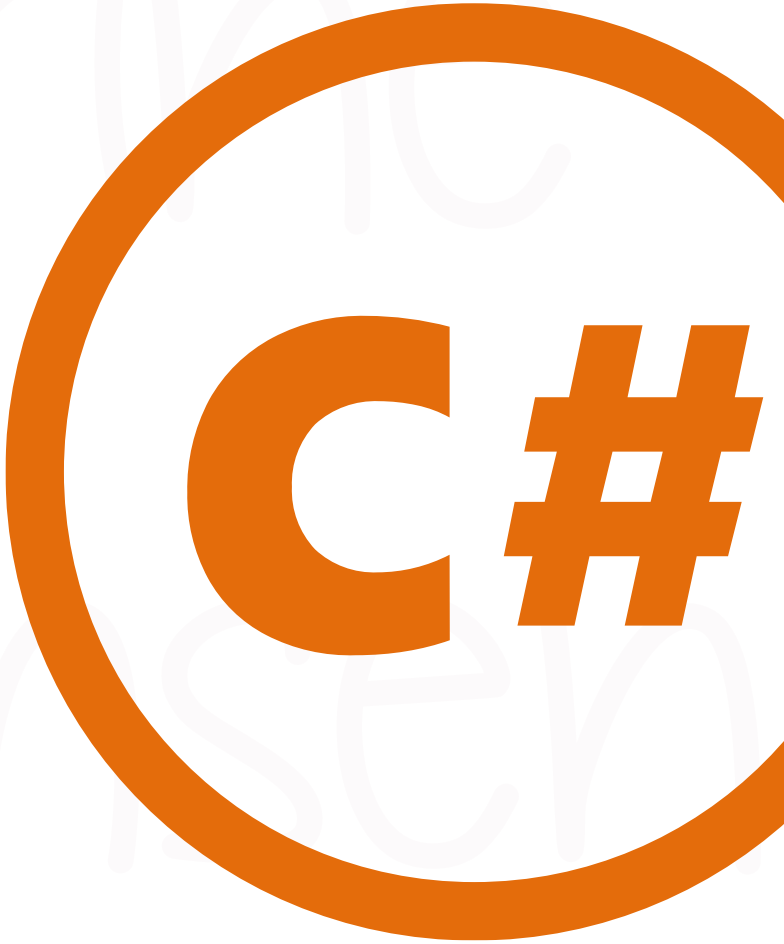
BimlScript to …?



DEMO

Practical Biml Coding

C# Classes and Methods



C# Classes and Methods

BimlScript and LINQ not enough?

Need to reuse C# code?

Create your own classes and methods!

C# Classes and Methods

```
public static class Utilities {
  public static string GetName(AstTableNode table) {
    return table.Schema.Name.ToUpper()
           + "_" + table.Name.ToUpper();
  }
}
```

C# Classes and Methods: Class

```
public static class Utilities {
    public static string GetName(AstTableNode table) {
        return table.Schema.Name.ToUpper()
            + "_" + table.Name.ToUpper();
    }
}
```

C# Classes and Methods: Method

```
public static class Utilities {
    public static string GetName(AstTableNode table) {
        return table.Schema.Name.ToUpper()
            + "_" + table.Name.ToUpper();
    }
}
```

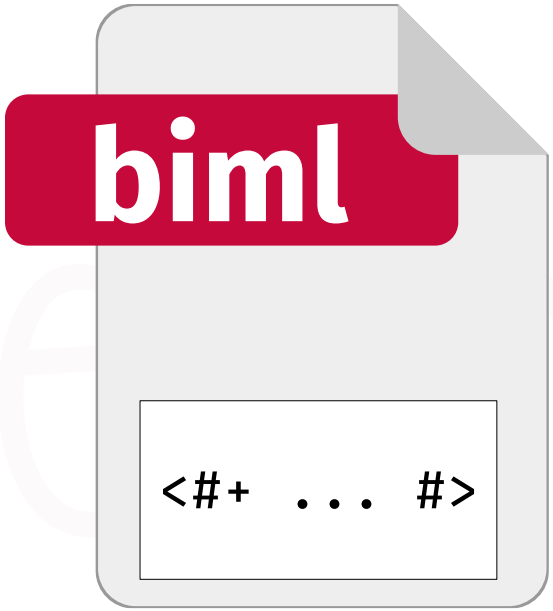
C# Classes and Methods: Logic

```
public static class Utilities {
  public static string GetName(AstTableNode table) {
    return table.Schema.Name.ToUpper()
           + "_" + table.Name.ToUpper();
  }
}
```

C# Code Management

Inline Code Blocks

```
<#+ ... #>
```



Included Biml Files with Code Blocks

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

Code Files

```
<#@ code file="Code.cs" #>
```

Code Blocks vs. Code Files

Code Blocks

- Used when logic is specific to one file
- Reuse by including files with code blocks

Code Files

- Use code editor of choice
- Create extension methods

C# Extension Methods

"Make it look like the method belongs to an object instead of a helper class"

Extension Methods: From this…

```
<#@ code file="HelperClass.cs" #>

<Biml xmlns="http://schemas.varigence.com/biml.xsd">
  <# foreach (var table in RootNode.Tables) { #>
    <# if (HelperClass.AnnotationTagExists(table, "SourceSchema")) { #>
      ...
    <# } #>
  <# } #>
</Biml>
```

```
public static class HelperClass {
  public static bool AnnotationTagExists(AstNode node, string tag) {
    return (node.GetTag(tag) != "") ? true : false;
  }
}
```

Extension Methods: ...to this

```
<#@ code file="HelperClass.cs" #>

<Biml xmlns="http://schemas.varigence.com/biml.xsd">
  <# foreach (var table in RootNode.Tables) { #>
    <# if (HelperClass.AnnotationTagExists(table, "SourceSchema")) { #>
      ...
    <# } #>
  <# } #>
</Biml>
```

```
public static class HelperClass {
  public static bool AnnotationTagExists(this AstNode node, string tag) {
    return (node.GetTag(tag) != "") ? true : false;
  }
}
```

Extension Methods: …to this

```
<#@ code file="HelperClass.cs" #>

<Biml xmlns="http://schemas.varigence.com/biml.xsd">
  <# foreach (var table in RootNode.Tables) { #>
    <# if (table.AnnotationTagExists("SourceSchema")) { #>
      ...
    <# } #>
  <# } #>
</Biml>
```

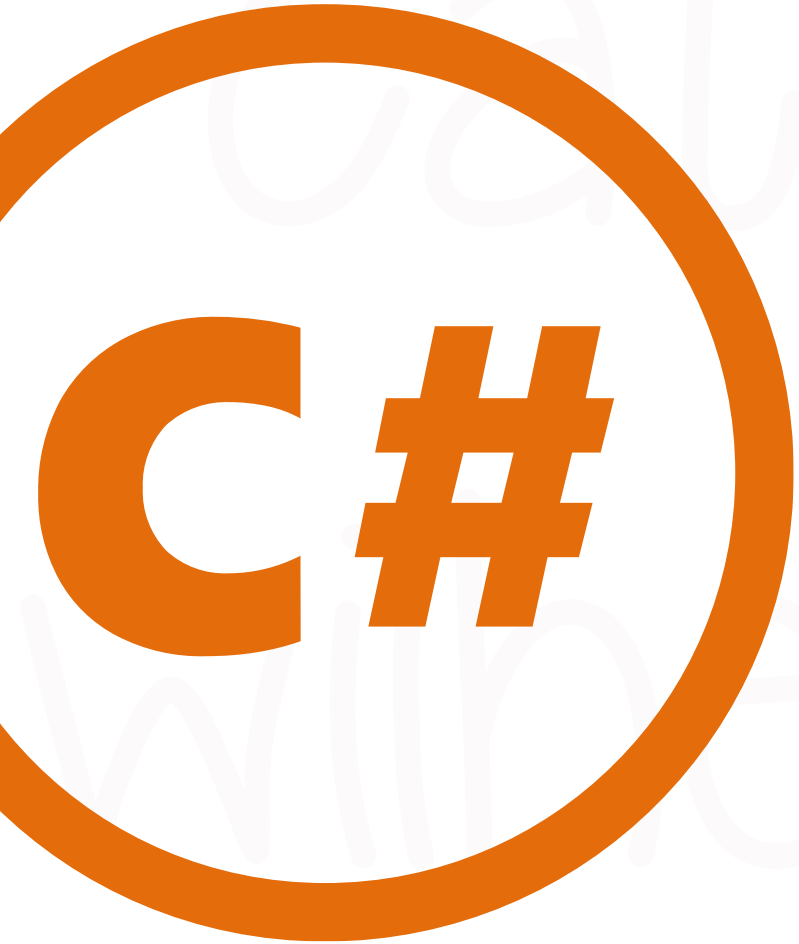
```
public static class HelperClass {
  public static bool AnnotationTagExists(this AstNode node, string tag) {
    return (node.GetTag(tag) != "") ? true : false;
  }
}
```

Extension Methods: ...to this :)

```
<#@ code file="HelperClass.cs" #>

<Biml xmlns="http://schemas.varigence.com/biml.xsd">
  <# foreach (var table in RootNode.Tables.Where(t =>
                                t.AnnotationTagExists("SourceSchema")) { #>
      ...
    <# } #>
  <# } #>
</Biml>
```

```
public static class HelperClass {
  public static bool AnnotationTagExists(this AstNode node, string tag) {
    return (node.GetTag(tag) != "") ? true : false;
  }
}
```



DEMO

C# Classes and Methods

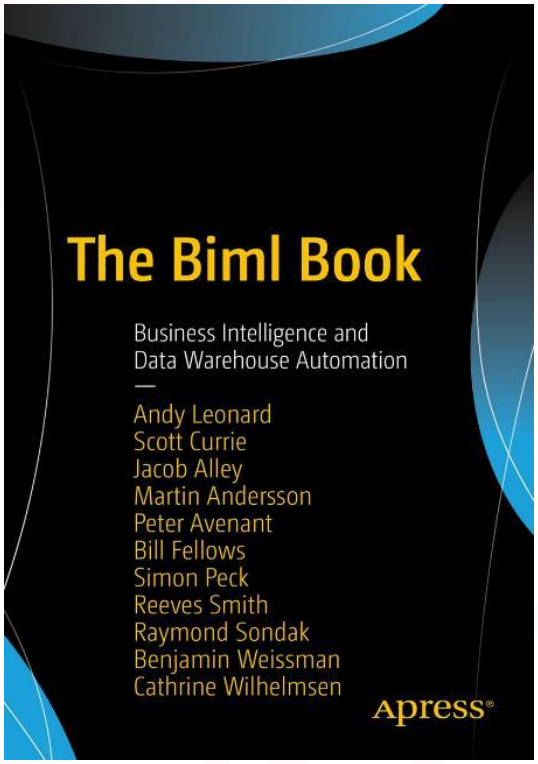
…the past 60 minutes…

Code Management

Practical Biml Coding

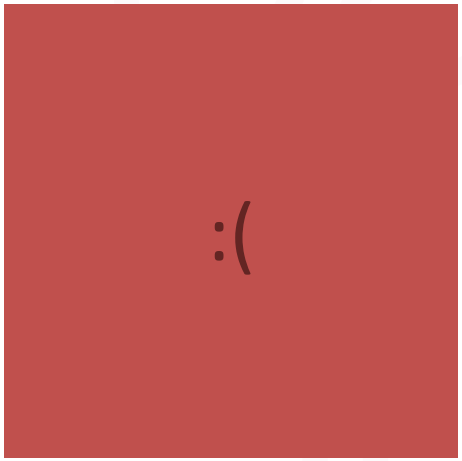
C# Classes and Methods

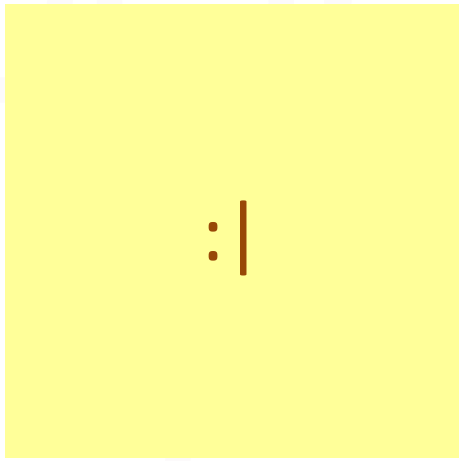
Where can you learn more?

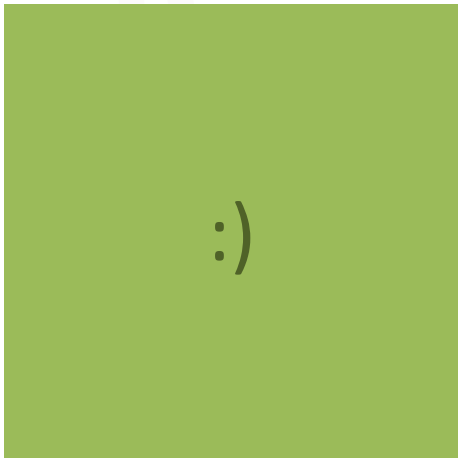


cathrinew.net/BimlBook

Evaluations







What can we do better? How can we improve?

Get things done

Start small

Start simple

Start with ugly code

Keep going

Expand

Improve

Deliver often

Biml resources and demo files:

cathrinew.net/biml

hi@cathrinew.net

@cathrinew

cathrinew.net