Message Transformations

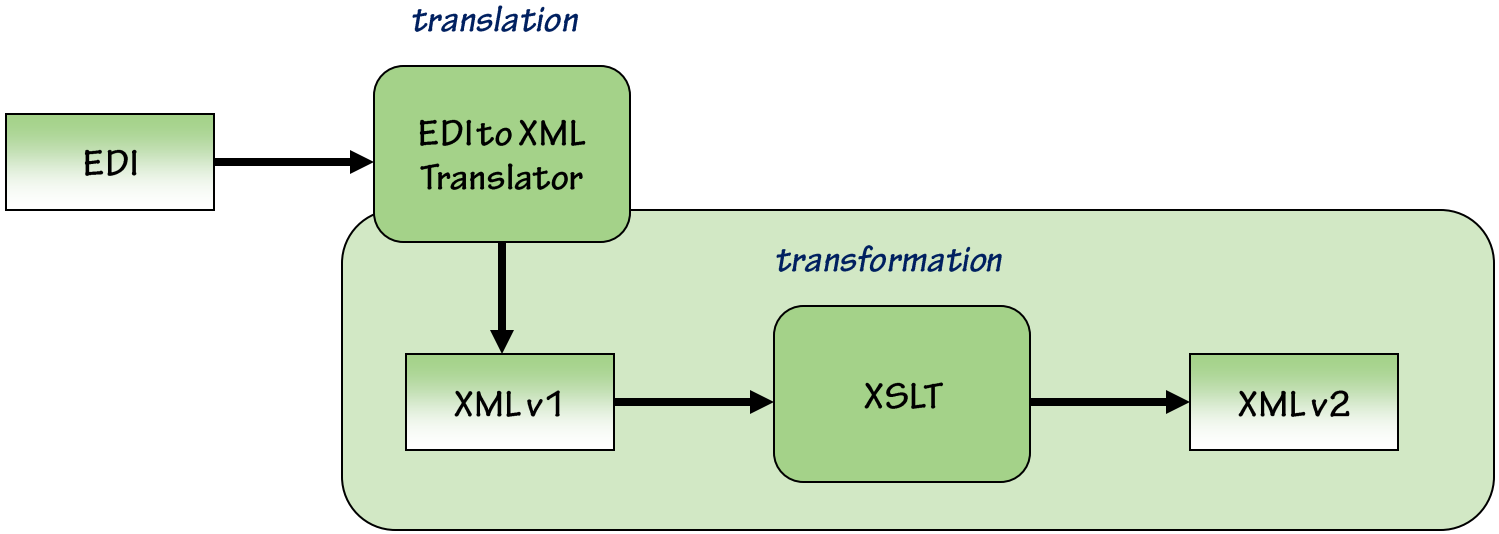
Moving between different message schemas

Outline

- BizTalk and XSLT
- BizTalk Mapper
- Links
- Functoids
- Scripting
- Custom functoids

Message transformations

- **Integration requires supporting multiple formats/schemas**
  - Applications won't always agree on same schemas
  - Message *translations* move between formats
  - Message *transformations* move between schemas
- ***XSLT* is a powerful XML transformation language**



XSLT

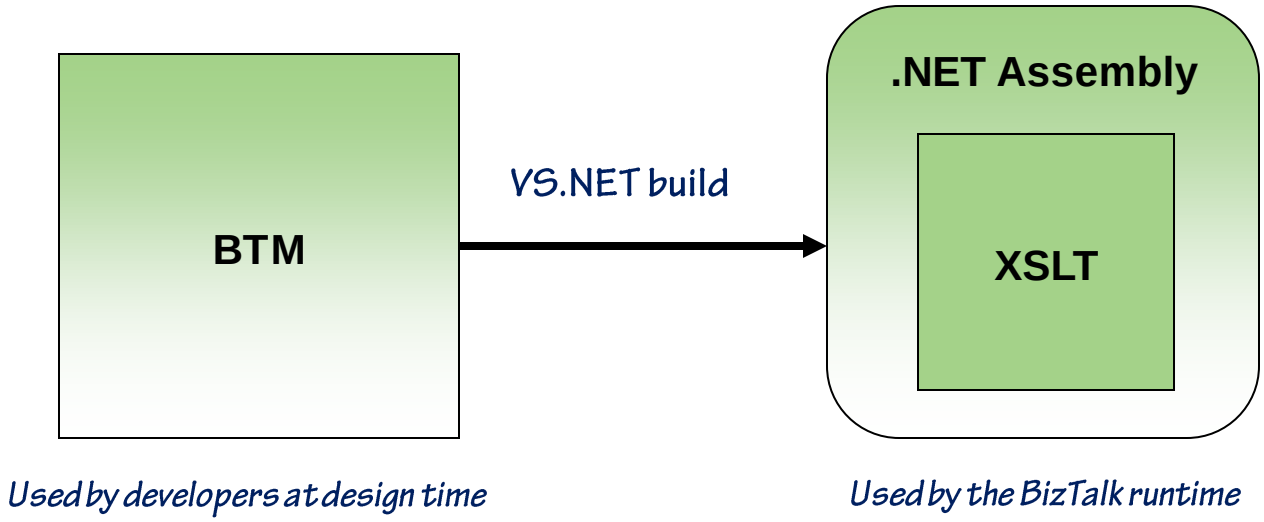
- **XSLT 1.0 is a standard XML transformation language**
  - Published as a W3C Recommendation
  - Accepts XML documents as input (must *translate* first)
  - Can produce XML or text-based output
- **The XSLT language is extremely powerful and flexible**
  - It's a functional language (not imperative)
  - It's *stateless* nature can complicate simple tasks
  - But complex *recursion* is a snap
  - Bottom line: it's not easy for most developers to master

BizTalk Mapper

- **The BizTalk Mapper takes the pain out of XSLT**
  - Maps a *source schema* to a *target schema*
  - Schemas visualized using instance-based tree views
  - Developer drags *links* between schemas
  - Simple 1-to-1 schema mappings become trivial
  - Properties offer control over most XSLT details
- **Advanced mappings possible through extensions**
  - A *functoid* is BizTalk's way of extending XSLT

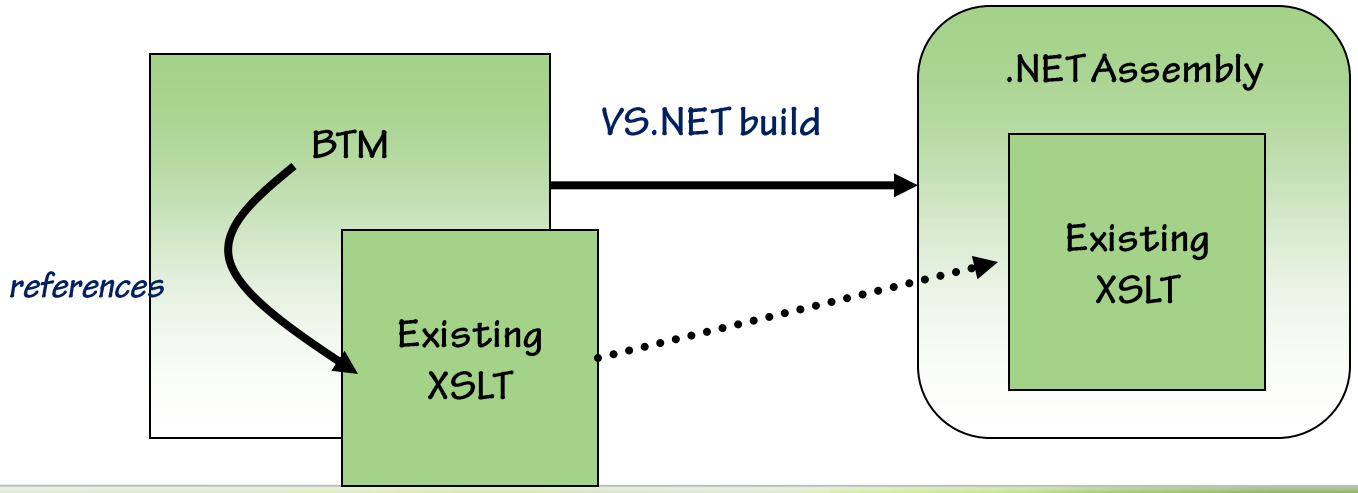
BTM vs. XSLT

- **The BizTalk Mapper stores map information in a *.btm* file**
  - The XSLT is generated from .btm when you build the map
  - XSLT stored within a .NET assembly in the GAC
  - Only the XSLT is used at runtime



Using existing XSLT transformations

- **Many companies already have existing XSLT transforms**
- **You can use existing XSLT transforms with a BTM file**
  - Set the *Custom XSL Path* to reference your file
- **Compiler builds the referenced XSLT into the assembly**
  - All map content is ignored in this case



Complex mapping issues

- **Simple 1-to-1 mappings can be handled with links alone**
- **More advanced mappings require more attention**
  - Conditional output, sums, database lookups, looping, etc.
- **BizTalk provides *functoids* to simplify these needs**

Functoids

- **A *functoid* represents an *extension function* for use in XSLT**
  - Functoids are available via the VS.NET toolbox
  - You place functoids on the map grid
  - Incoming links can specify input
  - Outgoing link specifies output
- **During build, the functoid translates to XSLT code**

Functoid categories

- **BizTalk Server 2006 ships with 80+ functoids**
  - They're organized in the following categories

Category	Description
String	String manipulation functions (size, find, trim, extract, etc)
Mathematical	Mathematical functions (add, subtract, multiply, divide, round, etc)
Logical	Logical functions (and, or, not, exists, greater than, less than, etc)
Date/Time	Date and time functions (date, time, add days, etc)
Conversion	Conversion between text formats (octal, hex, ascii, etc)
Scientific	Scientific functions (sine, cosine, tangent, logarithms, exponents, etc)
Cumulative	Cumulative functions (sum, average, max, min, concatenate, etc)
Database	Functions for looking up data in a database
Advanced	Functions for advanced mapping operations

Fundamental

- **The *string*, *date/time*, and *mathematical* functoids are core**
  - Use string functoids to manipulate text formats
  - Use date/time functoids to inject the current date/time into the target or to perform date-based operations
  - Use mathematical functoids to do basic math

Logical

- **The *Logical* functoids allow you to perform boolean operations**
  - Equality or inequality
  - Greater than or less than comparisons
  - Existence, type checks
  - You can combine tests with And, Or, Not
  - They return *true* or *false*
- **One primary use case is *conditional output***
  - Commonly used with *Value Mapping* and *Looping* functoids

Cumulative

- **Managing *state* while iterating over nodes is non-trivial in XSLT**
- **The *Cumulative* functoids make this easy**
  - You provide the source nodeset as input
  - Functoid takes care of the cumulative action
  - Supports: concatenate, sum, average, max, min

Database

- **The *Database* functoids extract data from a database for use in the target message**
  - Use *Database Lookup* to retrieve an ADO recordset row
  - You specify the connection string and table name
  - You specify a column and a lookup value for the column
  - Then use *Value Extractor* to extract values from row
  - You an use *Error Return* to harvest DB error information

Advanced functoids

- There are several advanced functoids, each with a purpose

Advanced Category	Functoids
Looping	Looping, Iteration, Record Count
Table Looping	Table Looping, Table Extractor
Value Mapping	Value Mapping, Value Mapping (Flattening)
Copying	Mass Copy
Scripting	Scripting
Misc.	Index, Assert, Nil Value

Scripting functoid

- **The *Scripting* functoid is the primary XSLT extensibility point**
  - Allows you to inject a custom "script" into the XSLT
- **Supports the following languages inline**
  - XSLT
  - C#, VB.NET, JScript.NET
- **Also supports invoking *external assemblies* in the GAC**

Writing custom functoids

- **If you want to package an extension for future reuse, write a *custom functoid***
  - Implemented in a .NET class and resource file
  - Derived from Microsoft.BizTalk.BaseFunctoid
  - Can supports inline code or external calls
- **Two general types of functoids: *regular* and *cumulative***
  - Regular implement one runtime-invoked function
  - Cumulative implements three runtime-invoked functions
  - Difference is *state management*

Summary

- BizTalk simplifies XSLT-based message transformations
- BizTalk Mapper shields developers from most XSLT details
- Links represent source-to-target mappings
- Functoids provide more advanced mapping functions
- BizTalk provides numerous canned functoids
- Scripting function offers additional extensibility
- You can write custom functoids to reuse your extensions

References

- **Using XML Schemas in BTS**
  - http://msdn.microsoft.com/library/default.asp?url=/library/en-us/BTS_2004WP/html/8c17983d-90b1-4e0a-9cc3-3bda0b1e7f55.asp
- **Message Transformations in BizTalk Server**
  - http://www.devx.com/enterprise/Article/29128
- **XSLT 1.0**
  - http://www.w3.org/TR/xslt
- **Essential XML Quick Reference (free download)**
  - http://www.theserverside.net/books/addisonwesley/EssentialXML/index.tss