



# Effective Control Flow Execution with 7 Different Types of Foreach Loop Enumerator

Reza Rad

SQL Server MVP, Author, DW / BI Architect

# About Me: Reza Rad

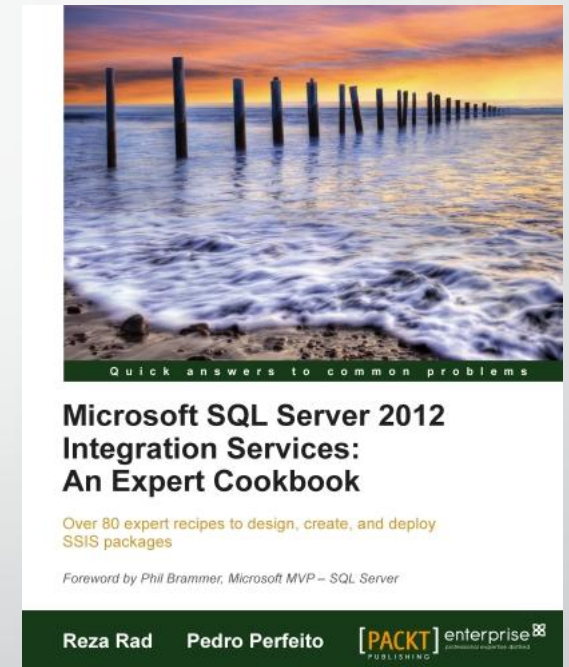
SQL Server MVP

Author of some SQL Server BI books

DW / BI Architect

Trainer

Author of SSIS Webcast Series



# Containers in SSIS

Sequence  
Container

For Loop  
Container

Foreach  
Loop  
Container

# Foreach Loop Container

- Loop through items of an array, data table, object collection, xml content, files, SQL Server objects, metadata objects.
- Benefits;
  - Reducing number of packages and tasks >> Lower maintenance cost

# Seven Types of Enumerators

- From Variable Enumerator
- Item Enumerator
- ADO Enumerator
- ADO.NET Schema Rowset Enumerator
- Nodelist Enumerator
- SMO Enumerator
- File Enumerator

# Dynamism in SSIS

- Variables
- Expression Language
  - Data Flow -> Derived Column, Conditional Split Transformation
  - Control Flow -> Expression Task, Precedence Constraint
- Expression Property
  - Setting property values dynamically



# Foreach File Enumerator



# Foreach File Enumerator

- Looping through files of a directory
- File mask can be used, such as "Customers?.csv"
- Fully qualified file path, or file name only (with or without extension) can be fetched
- Ability to traverse subfolders



# File Enumerator: DEMO

- Customer records exists in multiple CSV files in a source folder
- A loop structure will loops through CSV files, and;
  - Extract data from each file, and load it into the Customer database table
  - Move CSV files to Archive folder
  - Log an entry with file name and datetime into SourceFileLog table



# Foreach From Variable Enumerator



## Foreach From Variable Enumerator

- Looping through arrays of objects
- Array can be loaded with .NET scripts through Array or ArrayList or similar collection structures
- Array should be loaded into an OBJECT type variable in the package

# From Variable Enumerator: Sample

- SalesData CSV files coming from a source directory
- A loop structure should only pick those files that does not exists in the Archive folder. Load them with file name and date time into a database table, and move them to archive
- We use Script Task to find new files and then we fill them into an ArrayList, the ArrayList will be used in the Foreach Loop.

# Considerations

- Best practice to use From Variable is; where you can fetch an array (much easier) in a .NET script, and then use that array in the Foreach Loop.
- There are many situations that few lines of .NET script do same thing that more than dozens of SSIS tasks do. This enumerator impressively helps in those scenarios.



# Foreach ADO.NET Schema Rowset Enumerator

# ADO.NET Schema Rowset Enumerator

- Looping through schema and metadata of the data source
- Examples;
  - All tables under specific database
  - Views of Sales schema under the database
  - All system objects
- List of items can be filtered
- INFORMATION\_SCHEMA views in SQL Server

Columns  
Columns Domain Usage  
Constraint Column Usage  
Constraint Table Usage  
Foreign Keys  
Indexes  
Key Column Usage  
Primary Keys  
Procedure Columns  
Procedure Parameters  
Procedures  
Provider Types  
Referential Constraints  
Schemata  
Statistics  
Table Constraints  
Table Privileges  
Tables  
Translations  
Usage Privileges  
View Column Usage  
Views  
View Table Usage  
SQL Languages  
DB Info Keywords  
DB Info Literals  
Check Constraints by Table  
Table Statistics  
Tables Info  
Trustees

# ADO.NET Schema Rowset Enumerator: Sample

- Loop through all tables under AdventureWorks2012 database, with Sales Schema.
- Export the data of each table into a csv file with name of same table
- Structure of tables are different (Data flow cannot be used)
  - BCP for export from SQL Server to CSV





# Considerations

- Best for scenarios that looping through schema and metadata is required.
- Output for each schema view differs based on the information provided.
- Loop can be narrowed using filters in schema view's columns.



# Foreach Nodelist Enumerator

# Foreach Nodelist Enumerator

- Looping through elements and attributes of an xml data
- XML can be
  - DirectInput -> Static loop
  - File Connection -> Configurable loop
  - Variable -> Dynamic loop

# XPath

- Standard query language for XML content
- Examples:

List of all books:

`/bookstore/book`

Title of the specific book

`/bookstore/book[1]/title`

Author of books with price greater than 30:

`/bookstore/book[price>=30]/author`

More here:

<http://www.w3schools.com/xpath/default.asp>

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
```

# Xpath enumeration types

- Navigagor: Looping based on .NET XPathNavigator
- Node: Looping based on Xpath provided and returns Nodes
- NodeText: Looping based on Xpath provided and returns Node Texts
- ElementCollection: looping through elements provided through OuterXpath, and fetch data based on InnerXpath



# Considerations

- Using Nodelist enumerator with Web Service Task empowers the ETL scenario
- XPATH; more profession you gain in this query language, more power you'll have in Nodelist enumerators

# Nodelist Enumerator : DEMO

- List of teams that played in FIFA 2010 fetched from a web service
- A loop structure required to loop through team names, and call another web service to get full details of each team based on another web service.
- The second web service accepts team name as an input parameter
- Results of inner web services should be written into xml files named based on each time, such as Brazil.xml



# Foreach Item Enumerator





## Foreach Item Enumerator

- Loops through Static data table defined in the enumerator configuration.
- Column data types can be different, but not names.
- Useful for defining loop structure for multi column arrays in the packages

# Foreach Item Enumerator: Sample

- There are two database servers for Employee info, one for each department; Sales, and Production. Name of databases, and name of tables are different.
- Loop structure required to loop through two different servers with different db name and table names, and load data into a single destination database table on a third server.



# Considerations

- Good for looping through arrays with multiple columns (each column can have different data type).



# Foreach ADO Enumerator



## Foreach ADO Enumerator

- The most useful type of enumerator
- Loops through a data table that loaded into an object type variable
- Data table can be loaded from Execute SQL Task with Full Result Set
- Data table can be loaded from Data Flow Task with RecordSet Destination



## Advantages of ADO Enumerator

- Data Table can be loaded from any source through the Data Flow (Excel, Oracle, DB2, Flat File...)
- Data Table can be transformed through any transformation through the Data Flow
- Loop structure will be dynamic, because the source table that the data table comes from can be changed

# ADO Enumerator: DEMO

- Using ADO Enumerator to implement a simple package execution framework.
- Packages with their execution order entered into a database table, there is also an enable/disable tag.
- Loop structure will iterates through packages in the execution order and execute them one by one
- A log entry will be written back into a log table



# SMO Enumerator



# Foreach SMO Enumerator

- SMO = SQL Server Management Objects
- List of .NET libraries for managing SQL Server objects.
- Examples;
  - Looping through SQL Server jobs, Logins
  - Linked Servers

# Foreach SMO Enumerator: Sample

- There are two database servers for customer info; North Island, and South Island. Two Linked Server created to connect to each of these servers.
- Loop structure required to loop through these linked servers (**and any future linked servers with customer info**), and load Customer data from CustomerInfo table into an integrated database.

# Summary

- Foreach Loop empowers ETL design
  - Reduces redundant work
  - Reduces maintenance and support costs
  - Improves consistency and standardization
- Empower Foreach Loop with Expressions and Variables

# Tutorial Videos to Study More

- [ADO Enumerator](#)
- [Item Enumerator](#)
- [SMO Enumerator](#)
- [ADO.NET Schema Rowset Enumerator](#)
- [From Variable Enumerator](#)
- [Nodelist Enumerator](#)
- [File Enumerator](#)
- [For Loop Container](#)

**RADACAD**

# Questions?



[@Rad\\_Reza](#)



[a.raad.g@gmail.com](mailto:a.raad.g@gmail.com)



<http://www.rad.pasfu.com>



[nz.linkedin.com/in/rezarad/](https://nz.linkedin.com/in/rezarad/)



Thank you