Emilija Dimikj

May 19, 2020

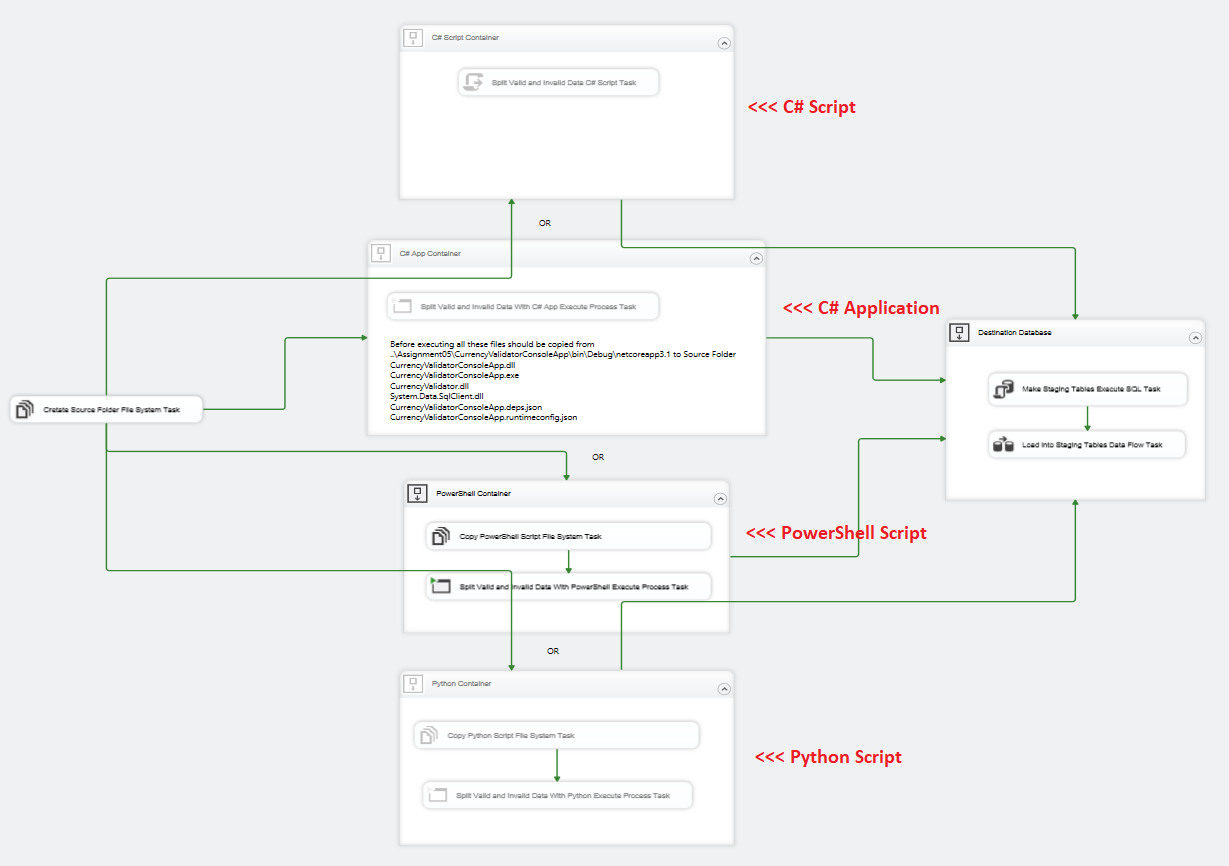
BIDD 220 B Sp 20: Data Migration Techniques (ETL Processing)

Assignment05

Performing ETL process using Non-SQL programming and SSIS

In this assignment we are performing ETL process using a variety of Non-SQL options like C# Application, C# script, Python and PowerShell script *(Figure 1)*. This is a very interesting assignment and to explore all options and possibilities and then to do it right I will need more than one week.

I have one SSIS project that contains 4 different ETL processes: C# script, C# console application, PowerShell script and Python script. SSIS project has one package. Whole solution contains that SSIS project, C# console application, C# class, plus code for PowerShell, C#, and Python that are put into subfolders.



*Figure 1: Non-Sql ETL Package*

# Introduction

This ETL process is starting with generating list of all valid world currencies, then checking the exchange rate data for validity and applying prices in currencies in Products data. For simplicity, I added just prices in USD and EUR. All four ETL processes are performing the same process:

*1: Extract source data*: My source data is in folder ***C:\CurrencyData*** in two files and database table

* ***CurrencyList.txt*** - List of all countries and their currencies.
* ***ExchangeRates.csv*** - Current exchange rates.
* ***SQL query*** that extracts Products with their Subcategories and categories from AdventureWorks\_Basics database.

*2. Transformation:*

* ***CurrencyList.txt*** is cleared of all rows that does not contains valid currencies and results are saved in two separate files: *validCurrencyList.txt* and *invalidCurrencyList.txt*. One currency is valid when it has three upper letter code and is checked with regex matching.
* ***ExchangeRates.csv*** is checked for valid and invalid currency rows, or it has three upper letter code and is contained in valid currency list produced with previous step. Results are saved in two separate files: *validExchangeRates.csv* and *invalidExchangeRates.csv*
* Results from the ***SQL query*** is cleared from null values and two new columns are added ProductPrice in USD and ProductPrice in EUR calculated with exchange rates from previous step. Result is saved in *ProductsForImport.csv* file

*3. Load:* Each result file is loaded in its own staging table in DWAdventureWorks\_Basics database

|  |  |  |
| --- | --- | --- |
| Source | Result file | Staging Table |
| CurrencyList.txt | validCurrencyList.txt | StagingValidCurrencies |
| invalidCurrencyList.txt | StagingInValidCurrencies |
| ExchangeRates.csv | validExchangeRates.csv | StagingExchangeRates |
| invalidExchangeRates.csv |  |
| SQL query | ProductsForImport.csv | StagingProducts |

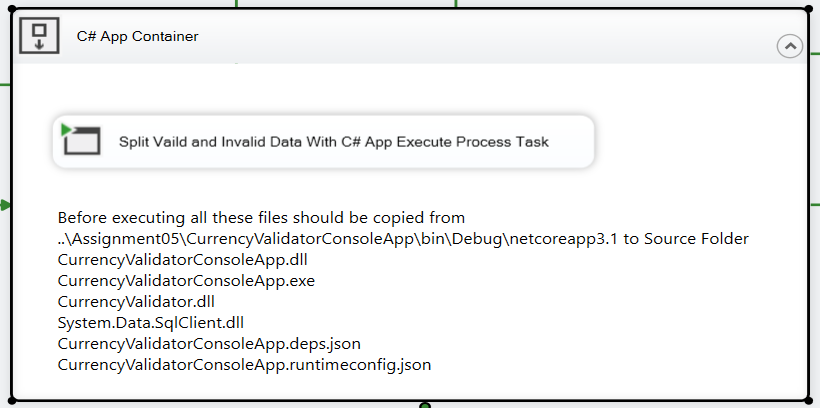
## C# script

The Script task integrated in SSIS provides code to perform functions that are not available in the built-in tasks and transformations that SSIS provides. The Script task can also combine functions in one script instead of using multiple tasks and transformations.

|  |  |
| --- | --- |
| Pros: | Cons: |
| * Integrated in SSIS * Usage of SSIS variables * Debugger * C# and VB programming languages * Everything is in the Package itself | * C# and VB programming languages * Slow debugging process |

## C# Application

|  |  |
| --- | --- |
| This is possible way of implementing existing code or code that was delivered in dll file to be used *(Figure 2)*. I am saying possible, but not easy to implement. It requires programming skills and very good implementation skills *(Figure 3)*. Not a lot of space to implement returning values other than saving files for next task to pick them up. | *Figure 2:C# Class and Console Application* |



*Figure 3: C# Application Execute Process Task*

|  |  |
| --- | --- |
| Pros: | Cons: |
| * Debugger * C# and VB programming languages * Usage of external libraries | * Difficult to implement * C# and VB programming languages * Implementation skills |

## PowerShell script

SSIS and PowerShell together offer a plenty of opportunities, and some shortcuts when having to import, export, or at times moving data. Doing same processing as Script Task with lines and lines of C# code that, done with PowerShell, could make maintaining that package much easier. PowerShell has a huge potential to save time of having to write any raw C# code for the packages. In the sense that it is built upon .NET framework itself, a lot of that code is accessible in much easier fashion.

|  |  |
| --- | --- |
| Pros: | Cons: |
| * Easy to implement * Easy maintenance | * Coding skills |

## Python script

Python is a very powerful programming language. Combined with SSIS, it can provide robust and flexible solutions. Python being a general-purpose, high-level programming language can support or perform a variety of tasks to enhance ETL development (not only SSIS). Perceived by many as easy to learn, super productive and with a generous Standard Library, Python can provide a great deal of functionality with hardly any code. SSIS functionality allows the developers to complete a vast plethora of tasks and further extend its capabilities through .NET languages implementation, however, Python’s sublime flexibility coupled with the ease of development not only provides somewhat gentler entry point into the programming realm but also allows for complex task execution with minimal development input.

|  |  |
| --- | --- |
| Pros: | Cons: |
| * Easy to implement * Easy maintenance | * Coding skills |

# Summary

For someone with good coding skills C# script is one of the best options. It is integrated and communication with SSIS Package is seamless. Maintenance is somewhat difficult, and debugger is very slow but on other hand everything is contained in one package.

PowerShell script is between C# script and Python. Coding part is more like C# and because is built upon .NET framework itself, a lot of that code is accessible in much easier fashion. Maintenance part is separated from Package itself and that makes it easier in case of changing functionality without touching the data flow.

For beginners in coding Python is best option. It is gaining in popularity so learning it is a good way to go. Python can provide a great deal of functionality with hardly any code, and it is easy to maintain and to implement.