SSIS: Create Folder If Missing, Then Copy Files (Idempotent Pattern)

A step-by-step, production-ready guide using Visual Studio 2022 / SQL Server Data Tools (SSDT).

# Real-world Scenario

Each day you ingest CSVs from C:\Drop\In. After successful processing, you archive the files to a date-stamped folder under C:\Drop\Archive (e.g., C:\Drop\Archive\20250810). The package must:  
• Create the archive folder only if it doesn’t exist (and not fail if it already exists),  
• Copy every \*.csv file from the source to the archive folder,  
• Be safe to re-run any number of times (idempotent).

# Prerequisites

• Visual Studio 2022 with SQL Server Data Tools (SSDT) and Integration Services Extension installed.

• Read/Write permissions to the source and archive folders.

# Step 1 — Define Package Variables

Create the following \*\*User\*\*-scope variables (SSIS → Variables):

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Value / Expression | Notes |
| SourceFolder | String | C:\Drop\In | Folder to read files from. |
| ArchiveRoot | String | C:\Drop\Archive | Base folder where date-named folders are created. |
| ArchiveFolder | String | EvaluateAsExpression = True | Final folder for this run (see expression below). |
| FileSpec | String | \*.csv | Pattern of files to process. |
| CurrentFile | String |  | Holds the fully qualified file name during the loop. |
| DestFile | String | EvaluateAsExpression = True | Destination file path computed from ArchiveFolder + file name. |

Use these expressions:

ArchiveFolder (Expression):

@[User::ArchiveRoot] + "\\"  
+ (DT\_WSTR,4)YEAR(GETDATE())   
+ RIGHT("0" + (DT\_WSTR,2)MONTH(GETDATE()),2)   
+ RIGHT("0" + (DT\_WSTR,2)DAY(GETDATE()),2)

DestFile (Expression):

@[User::ArchiveFolder] + "\\"   
+ REVERSE(   
 SUBSTRING(   
 REVERSE(@[User::CurrentFile]),   
 1,   
 FINDSTRING(REVERSE(@[User::CurrentFile]),"\\",1) - 1   
 )   
)

# Step 2 — Build the Control Flow

Drag the following onto the Control Flow surface in this order:

1) Script Task — \*\*Ensure Archive Folder\*\* (idempotent)  
2) Foreach Loop Container — iterate over source files  
 → inside it: File System Task — \*\*Copy File to Archive\*\*

# Step 3 — Script Task: Ensure Archive Folder

Configuration:

• Name: Ensure Archive Folder  
• ReadOnlyVariables: User::ArchiveFolder

Code (C#):

using System;  
using System.IO;  
using Microsoft.SqlServer.Dts.Runtime;  
  
public void Main()  
{  
 string folder = (string)Dts.Variables["User::ArchiveFolder"].Value;  
 // Idempotent: does not throw if folder already exists  
 Directory.CreateDirectory(folder);  
 Dts.TaskResult = (int)ScriptResults.Success;  
}

# Step 4 — Foreach Loop: Enumerate Source Files

Open the Foreach Loop editor and configure:

Collection tab:

• Enumerator: Foreach File Enumerator  
• Folder: set via Expressions → Directory = @[User::SourceFolder]  
• Files: either set directly to \*.csv or via Expressions → FileSpec = @[User::FileSpec]  
• Retrieve file name: Fully qualified

Variable Mappings tab:

• Index 0 → User::CurrentFile

# Step 5 — File System Task: Copy File

Inside the Foreach Loop, add a File System Task and configure:

• Operation: Copy file  
• IsSourcePathVariable: True → SourceVariable = User::CurrentFile  
• IsDestinationPathVariable: True → DestinationVariable = User::DestFile  
• OverwriteDestination: True (safe for re-runs)

# Why This Pattern Fixes Re-run Failures

• The Script Task uses Directory.CreateDirectory(), which is safe even if the folder already exists. Traditional “Create Directory” task fails on duplicates.  
• Copy task overwrites existing files, so a second run does not error.

# Optional (No-code Alternative with Pre-check)

If you prefer to keep the stock Create Directory task:

1) Add a small Script Task to set a Boolean variable User::DirExists = Directory.Exists(@ArchiveFolder).  
2) Use precedence constraints so Create Directory runs only when DirExists == False.  
3) Then run the Foreach loop and Copy task as above.

# Production Tips

• Log source, destination, and file counts to an audit table.  
• Wrap copy operations with a retry (e.g., OnError → Script Task with a short wait/retry) for transient locks.  
• Consider archiving with a subfolder per run ID or per file date if you need multiple reruns on the same day.

# Troubleshooting Checklist

• Permission denied: run the package under a credential that has read/write to both folders.  
• Expression typos: remember to escape backslashes in SSIS expressions as \\.  
• Empty loop: verify FileSpec and folder path; test using a File System Task configured to create a marker file.