using System;

using System.Data;

using System.IO;

using Microsoft.SqlServer.Dts.Runtime;

using System.Text.RegularExpressions;

[Microsoft.SqlServer.Dts.Tasks.ScriptTask.SSISScriptTaskEntryPoint]

public partial class ScriptMain : Microsoft.SqlServer.Dts.Tasks.ScriptTask.VSTARTScriptObjectModelBase

{

public void Main()

{

try

{

// Example: Reading data from a flat file (you can modify this for database connections)

string filePath = Dts.Variables["User::InputFilePath"].Value.ToString();

string[] fileLines = File.ReadAllLines(filePath);

// Validate each line (Assuming CSV with CustomerID, Email, Age)

foreach (string line in fileLines)

{

string[] columns = line.Split(',');

// Check for null or empty CustomerID

if (string.IsNullOrWhiteSpace(columns[0]))

{

Dts.Events.FireInformation(0, "Validation Error", $"Missing CustomerID in line: {line}", string.Empty, 0, ref false);

}

// Check for valid email format

if (!Regex.IsMatch(columns[1], @"^[^@\s]+@[^@\s]+\.[^@\s]+$"))

{

Dts.Events.FireInformation(0, "Validation Error", $"Invalid Email in line: {line}", string.Empty, 0, ref false);

}

// Check for valid age

if (!int.TryParse(columns[2], out int age) || age <= 0)

{

Dts.Events.FireInformation(0, "Validation Error", $"Invalid Age in line: {line}", string.Empty, 0, ref false);

}

}

// Success

Dts.TaskResult = (int)ScriptResults.Success;

}

catch (Exception ex)

{

// Log the exception and fail the task

Dts.Events.FireError(0, "Script Task", ex.Message, string.Empty, 0);

Dts.TaskResult = (int)ScriptResults.Failure;

}

}

// Enum for task result status

enum ScriptResults

{

Success = Microsoft.SqlServer.Dts.Runtime.DTSExecResult.Success,

Failure = Microsoft.SqlServer.Dts.Runtime.DTSExecResult.Failure

}

}