/\*

Microsoft SQL Server Integration Services Script Task

Write scripts using Microsoft Visual C# 2008.

The ScriptMain is the entry point class of the script.

\*/

using System;

using System.Data;

using Microsoft.SqlServer.Dts.Runtime;

using System.Windows.Forms;

namespace ST\_fcd56ec20b8646dfb3261487b129ff89.csproj

{

[System.AddIn.AddIn("ScriptMain", Version = "1.0", Publisher = "", Description = "")]

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

{

#region VSTA generated code

enum ScriptResults

{

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

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

};

#endregion

/\*

The execution engine calls this method when the task executes.

To access the object model, use the Dts property. Connections, variables, events,

and logging features are available as members of the Dts property as shown in the following examples.

To reference a variable, call Dts.Variables["MyCaseSensitiveVariableName"].Value;

To post a log entry, call Dts.Log("This is my log text", 999, null);

To fire an event, call Dts.Events.FireInformation(99, "test", "hit the help message", "", 0, true);

To use the connections collection use something like the following:

ConnectionManager cm = Dts.Connections.Add("OLEDB");

cm.ConnectionString = "Data Source=localhost;Initial Catalog=AdventureWorks;Provider=SQLNCLI10;Integrated Security=SSPI;Auto Translate=False;";

Before returning from this method, set the value of Dts.TaskResult to indicate success or failure.

To open Help, press F1.

\*/

public void Main()

{

// TODO: Add your code here

int error\_count = 0;

string error\_detail = "";

string excelFilePath = Dts.Variables["User::File\_Name"].Value.ToString();

string strCn = "Provider=Microsoft.ACE.OLEDB.12.0;" + "Data Source=" + excelFilePath + ";Excel 12.0;HDR=YES";

DataTable dtTables = new DataTable(); ;

System.Data.OleDb.OleDbConnection cn = new System.Data.OleDb.OleDbConnection(strCn);

cn.Open();

System.Data.OleDb.OleDbCommand command = new System.Data.OleDb.OleDbCommand();

command.Connection = cn;

command.CommandText = "SELECT \* FROM [Sheet1$]";

System.Data.OleDb.OleDbDataAdapter adapter = new System.Data.OleDb.OleDbDataAdapter();

adapter.SelectCommand = command;

adapter.Fill(dtTables);

string str = string.Empty;

if (dtTables.Columns.Count != 45)

{

error\_count += 1;

error\_detail = "Mis-match in the columns count" + "\r\n";

}

if (!dtTables.Columns.Contains("SSN"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name SSN is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Borrower FirstName"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Borrower FirstName is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Borrower LastName"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Borrower LastName is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Borrower Email"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Email is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Home Phone"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Home Phone is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Cell Phone"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Cell Phone is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Marital Status"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Marital Status is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Date of Birth"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Date of Birth is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Current Street Address"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Current Street Address is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("City"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name City is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("State"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name State is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Zip"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Zip is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("YearsAtThisAddress"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name YearsAtThisAddress is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("MonthlyIncome"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name MonthlyIncome is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Bonuses"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Bonuses is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Commission"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Commission is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("OtherIncome"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name OtherIncome is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Rent or Own"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Rent or Own is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Purpose of Loan"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Purpose of Loan is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Property Usage"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Property Usage is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("LoanAmount"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name LoanAmount is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Purchase Price"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Purchase Price is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Number of Units"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Number of Units is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Property City"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Property City is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Property State"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Property State is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Property Zip"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Property Zip is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Sex"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Sex is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Ethnicity"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Ethnicity is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Race"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Race is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("CreditCardAuthorization"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name CreditCardAuthorization is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Checking"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Checking is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Savings"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Savings is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("RetirementFund"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name RetirementFund is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("MutualFund"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name MutualFund is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Referral"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Referral is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("RealEstateAgentName"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name RealEstateAgentName is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("RealEstateAgentPhone"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name RealEstateAgentPhone is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("RealEstateAgentEmail"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name RealEstateAgentEmail is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Loan\_ID"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Loan\_ID is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Property\_ID"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Property\_ID is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Co-Borrower SSN"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Co-Borrower SSN is mis-spelled or not exists in the file. " + "\r\n";

}

if (!dtTables.Columns.Contains("Loan Date"))

{

error\_count += 1;

error\_detail = error\_detail + " Column name Loan to Date is mis-spelled or not exists in the file. " + "\r\n";

}

Dts.Variables["User::ErrorsInSource"].Value = error\_count;

Dts.Variables["User::Body"].Value = error\_detail;

cn.Close();

}

// Dts.TaskResult = (int)ScriptResults.Success;

}

}