

# How to validate email addresses in SSIS using C# (Advanced ETL)

Programming By: Chris Singleton

**NewPatientsVisitsAddDateEmailValidatorETL - Microsoft Visual Studio**

File Edit View Project Build Debug Team Format SSIS Tools Architecture Test Analyze Window Help

Developi Any CPU Start SQL SQL

**SSIS Toolbox**

- Favorites
  - Data Flow Task
  - Execute SQL Task
- Common
  - Analysis Services Processing...
  - Bulk Insert Task
  - Data Profiling Task
  - Execute Package Task
  - Execute Process Task
  - Expression Task
  - File System Task
  - FTP Task
  - Hadoop File System Task
  - Hadoop Hive Task
  - Hadoop Pig Task
  - Script Task
  - Send Mail Task
  - Web Service Task
  - XML Task
- Azure
- Containers
  - For Loop Container
  - Foreach Loop Container
  - Sequence Container
- Other Tasks
  - Analysis Services Execute D...
  - Back Up Database Task
  - CDC Control Task
  - Check Database Integrity T...
  - Data Mining Query Task
  - Execute SQL Server Agent J...
  - Execute T-SQL Statement T...
  - History Cleanup Task
  - Maintenance Cleanup Task
  - Message Queue Task
  - Notify Operator Task
  - Rebuild Index Task
  - Reorganize Index Task
  - Shrink Database Task

**NewPatientsVisitsA...orETL.dtsx [Design]**

Control Flow Data Flow Parameters Event Handlers Package Explorer

of data files exist, then Split them into Valid, Invalid files and folders.

**Scripts Split Valid and InValid Emails**

Split Bellevue Valid and Invalid Emails C# Script Task

Split Kirkland Valid and Invalid Emails C# Script Task

Split Redmond Valid and Invalid Emails C# Script Task

Copy Data From Flat File to Patients Database.

even if there are no files to be processed in other areas.

**Context Menu:**

- Edit...
- Disable
- Add Precedence Constraint
- Group
- Autosize
- Execute Task
- Edit Breakpoints...
- Zoom
  - Cut Ctrl+X
  - Copy Ctrl+C
  - Paste Ctrl+V
  - Delete Del
  - Rename
  - Select All Ctrl+A
  - Properties Alt+Enter

**Connection Managers**

- 183.BellevueVisitsDerivedDTSourcesFlatConn
- 183.KirklandVisitsDerivedDTSourcesFlatConn
- 183.RedmondVisitsDerivedDTSourcesFlatConn
- 1.BellevueEmailToValidateSourceFlatConn
- 1.BellevueNPOriginEmailSourceFlatFileConn
- 1.KirklandNPOriginEmailSourceFlatFileConn
- 1.RedmondNPOriginEmailSourceFlatFileConn
- 2.BellevueParseFlatEmailDestConn
- 2.BellevueVisitsDerivedDTDestFlatConn

**Error List**

Entire Solution 0 Errors 0 Warnings 0 Messages Build + IntelliSense Search Error List

**Solution Explorer**

Solution 'NewPatientsVisitsAddD'

- Data Modeling Designs
  - DoctorsSchedules (OLTP)
  - DWCorporateReports (OL
  - Patients (OLTP)
- Database BackUps
- Documentation
  - BI Solutions WorkSheet
  - LessonsLearned.docx
  - TechnicalSupportOverview
- DWCorporateReportsDB (OL
- DWCorporateReports.bak
- DWCorporateReports.mdt
- Error Log
  - ErrorLog.Log
- New Patients Data Date Time
  - Bellevue\_VisitsDTMerged.
  - Kirkland\_VisitsDTMerged.
  - Redmond\_VisitsDTMerged.
- Original NewPatients Data
  - Bellevue
  - Kirkland
  - Redmond
- Patients Doctors Clinics DBs (
- DoctorsSchedules.mdf
- Patients.mdf
- T-SQL Scripts
  - DWCorporateReportsDB.s
  - DWCorporateReportsETL.
  - PatientsTEST\_DB.sql
- Validated Invalid Emails
  - Bellevue\_InValidEmailData
  - Kirkland\_InValidEmailData
  - Redmond\_InValidEmailDa
- Validated Valid Emails
  - Bellevue\_ValidEmailData.c
  - Kirkland\_ValidEmailData.c
  - Redmond\_ValidEmailData
- NewPatientsAddDateEmail
- Data Sources
- SSIS Packages
  - NewPatientsVisitsAdd

Tab... Solu... Tea... Clas... Noti...

Ready 4:43 PM 3/28/2017

**NewPatientsVisitsAddDateEmailValidatorETL - Microsoft Visual Studio**

File Edit View Project Build Debug Team Format SSIS Tools Architecture Test Analyze Window Help

Develop... Any CPU Start SQL SQL

**SSIS Toolbox**

- Favorites
  - Data Flow Task
  - Execute SQL Task
- Common
  - Analysis Services Processin...
  - Bulk Insert Task
  - Data Profiling Task
  - Execute Package Task
  - Execute Process Task
  - Expression Task
  - File System Task
  - FTP Task
  - Hadoop File System Task
  - Hadoop Hive Task
  - Hadoop Pig Task
  - Script Task
  - Send Mail Task
  - Web Service Task
  - XML Task
- Azure
- Containers
  - For Loop Container
  - Foreach Loop Container
  - Sequence Container
- Other Tasks
  - Analysis Services Execute D...
  - Back Up Database Task
  - CDC Control Task
  - Check Database Integrity T...
  - Data Mining Query Task
  - Execute SQL Server Agent J...
  - Execute T-SQL Statement T...
  - History Cleanup Task
  - Maintenance Cleanup Task
  - Message Queue Task
  - Notify Operator Task
  - Rebuild Index Task
  - Reorganize Index Task
  - Shrink Database Task

**Task List**

- 183.BellevueVisitsDerivedDTSourFlatConn
- 1.BellevueEmailToValidateSourceFlatConn
- 1.RedmondNPorginEmailSourceFlatFileConn

**Script Task Editor**

Access Microsoft Visual Studio Tools for Applications (VSTA) to write scripts using the Visual Basic 2015 or Visual C# 2015, and configure the task's properties.

**Script**

ScriptLanguage: Microsoft Visual C# 2015

EntryPoint: Main

ReadOnlyVariables:

ReadWriteVariables:

**ScriptLanguage**

Specifies the programming language used by the script.

**Edit Script...**

Click Button

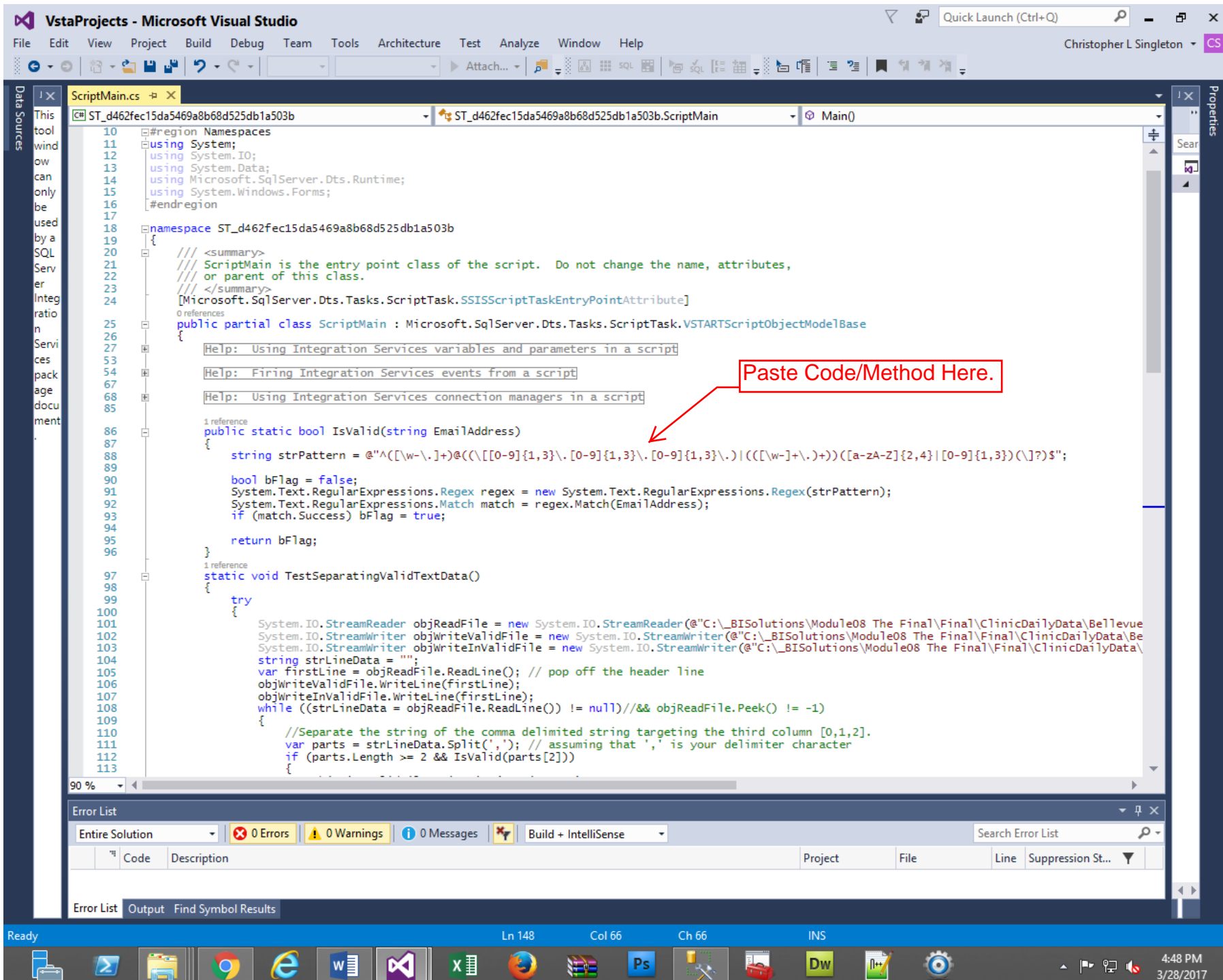
**Information...**

Server Explorer SSIS Toolbox

Ready

4:44 PM 3/28/2017

Note: When you hit the edit button, a new instance of Visual Studio comes up with a C# window.



Splits Valid/Invalid emails to other CSV files from the original CSV Flat File.

```
public static bool IsValid(string EmailAddress)
{
    string strPattern = @"^([\w-\.]*)@((\[[0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\. )|(([\w-]+\.)+))([a-zA-Z]{2,4}|[0-9]{1,3})(\ )?$";

    bool bFlag = false;
    System.Text.RegularExpressions.Regex regex = new System.Text.RegularExpressions.Regex(strPattern);
    System.Text.RegularExpressions.Match match = regex.Match(EmailAddress);
    if (match.Success) bFlag = true;

    return bFlag;
}

static void TestSeparatingValidTextData()
{
    try
    {
        System.IO.StreamReader objReadFile = new System.IO.StreamReader(@"+
        C:\_BISolutions\Module08 The Final\Final\ClinicDailyData\Bellevue\
        Bellevue_EmailsToValidate\Bellevue_EmailToValidateData.csv");

        System.IO.StreamWriter objWriteValidFile = new System.IO.StreamWriter(@"+
        C:\_BISolutions\Module08 The Final\Final\ClinicDailyData\Bellevue\
        Bellevue_ValidEmail\Bellevue_ValidEmailData.csv", false);

        System.IO.StreamWriter objWriteInvalidFile = new System.IO.StreamWriter(@"+
        C:\_BISolutions\Module08 The Final\Final\ClinicDailyData\Bellevue\
        Bellevue_InvalidEmail\Bellevue_InvalidEmailData.csv", false);

        string strLineData = "";
        var firstLine = objReadFile.ReadLine(); // pop off the header line.
        objWriteValidFile.WriteLine(firstLine);
        objWriteInvalidFile.WriteLine(firstLine);
        while ((strLineData = objReadFile.ReadLine()) != null)
        {
            //Separate the string of the comma delimited string targeting the third column [0,1,2].
            var parts = strLineData.Split(','); // assuming that ',' is your delimiter character
            if (parts.Length >= 2 && IsValid(parts[2]))
            {
                objWriteValidFile.WriteLine(strLineData);
            }
        }
    }
}
```

RegEx Pattern that validates the given email address in the CSV Flat File.

Instantiated Object that Reads the file.

Location of File to Read.

Loop through each row of line data.

Targets the third column (third string). Columns start at 0.

Write the whole row (line) of data to the file named "ValidEmail Data".

```

        else
        {
            objWriteInvalidFile.WriteLine(strLineData);
        }
    }
    objReadFile.Close();
    objWriteValidFile.Close();
    objWriteInvalidFile.Close();
}
catch (System.Exception objException)
{
    System.Console.WriteLine(objException.ToString());
    throw objException; //Must add this for the Main method to catch this exception.
}
}
/// <summary>
/// This method is called when this script task executes in the control flow.
/// Before returning from this method, set the value of Dts.TaskResult to indicate success or failure.
/// To open Help, press F1.
/// </summary>
public void Main()
{
    try
    {
        TestSeparatingValidTextData();
        Dts.TaskResult = (int)ScriptResults.Success;
    }
    catch (Exception objException)
    {
        string strMessage = @"On Error, please check the file is in C:\_BISolutions\"+
            "Module08 The Final\Final\ClinicDailyData\Bellevue\ Folder";
        Dts.Events.FireError(0, strMessage, objException.ToString(), string.Empty, 0);
        Dts.TaskResult = (int)ScriptResults.Failure;
    }
}

```

← If false, then writes to InvalidEmailData file.

Close Methods, Close Read/Write objects.

← Call Method

← Handle Errors

Message to show when an error happens.

```
#region ScriptResults declaration
/// <summary>
/// This enum provides a convenient shorthand within the scope of this class for setting the
/// result of the script.
///
/// This code was generated automatically.
/// </summary>
enum ScriptResults
{
    Success = Microsoft.SqlServer.Dts.Runtime.DTSExecResult.Success,
    Failure = Microsoft.SqlServer.Dts.Runtime.DTSExecResult.Failure
};
#endregion
```

Added the code again for re-usability.

```
public static bool IsValid(string EmailAddress)
{
    string strPattern = @"^([\w-\.]*)@((\[[0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\. )|(([\w-]+\.)+))([a-zA-Z]{2,4}|[0-9]{1,3})(\ [])?$";

    bool bFlag = false;
    System.Text.RegularExpressions.Regex regex = new System.Text.RegularExpressions.Regex(strPattern);
    System.Text.RegularExpressions.Match match = regex.Match(EmailAddress);
    if (match.Success) bFlag = true;

    return bFlag;
}
static void TestSeparatingValidTextData()
{
    try
    {
        System.IO.StreamReader objReadFile = new System.IO.StreamReader(@"+
            C:\_BISolutions\Module08 The Final\Final\ClinicDailyData\Bellevue\
            Bellevue_EmailsToValidate\Bellevue_EmailToValidateData.csv");
        System.IO.StreamWriter objWriteValidFile = new System.IO.StreamWriter(@"+
            C:\_BISolutions\Module08 The Final\Final\ClinicDailyData\Bellevue\
            Bellevue_ValidEmail\Bellevue_ValidEmailData.csv", false);
        System.IO.StreamWriter objWriteInvalidFile = new System.IO.StreamWriter(@"+
            C:\_BISolutions\Module08 The Final\Final\ClinicDailyData\Bellevue\
            Bellevue_InvalidEmail\Bellevue_InvalidEmailData.csv", false);

        string strLineData = "";
        var firstLine = objReadFile.ReadLine(); // pop off the header line.
        objWriteValidFile.WriteLine(firstLine);
        objWriteInvalidFile.WriteLine(firstLine);
        while ((strLineData = objReadFile.ReadLine()) != null)
        {
            //Separate the string of the comma delimited string targeting the third column [0,1,2].
            var parts = strLineData.Split(','); // assuming that ',' is your delimiter character
            if (parts.Length >= 2 && IsValid(parts[2]))
            {
                objWriteValidFile.WriteLine(strLineData);
            }
        }
    }
}
```

```

        else
        {
            objWriteInValidFile.WriteLine(strLineData);
        }
    }
    objReadFile.Close();
    objWriteValidFile.Close();
    objWriteInValidFile.Close();
}
catch (System.Exception objException)
{
    System.Console.WriteLine(objException.ToString());
    throw objException; //Must add this for the Main method to catch this exception.
}
}

/// <summary>
/// This method is called when this script task executes in the control flow.
/// Before returning from this method, set the value of Dts.TaskResult to indicate success or failure.
/// To open Help, press F1.
/// </summary>
public void Main()
{
    try
    {
        TestSeparatingValidTextData();
        Dts.TaskResult = (int)ScriptResults.Success;
    }
    catch (Exception objException)
    {
        string strMessage = @"On Error, please check the file is in C:\_BISolutions\"+
                            "Module08 The Final\Final\ClinicDailyData\Bellevue\ Folder";
        Dts.Events.FireError(0, strMessage, objException.ToString(), string.Empty, 0);
        Dts.TaskResult = (int)ScriptResults.Failure;
    }
}
}

```



```
#region ScriptResults declaration
/// <summary>
/// This enum provides a convenient shorthand within the scope of this class for setting the
/// result of the script.
///
/// This code was generated automatically.
/// </summary>
enum ScriptResults
{
    Success = Microsoft.SqlServer.Dts.Runtime.DTSExecResult.Success,
    Failure = Microsoft.SqlServer.Dts.Runtime.DTSExecResult.Failure
};
#endregion
```