**How to Automate Daily Newsletter by RPA – Keyword Driven Approach**

*Requirement – Build Automation solution for daily newsletter which can work as both attended and unattended without hampering the work in hand. We will not make use of any browser, drivers. This will be the headless browser automation with the help of Fizzler and HtmlAgilityPack.*

For this example, to extract news, we will consider following websites –

<https://www.financialexpress.com>

<https://economictimes.indiatimes.com>

<https://www.thehindubusinessline.com>

<https://www.livemint.com>

<https://indianexpress.com>

<https://www.business-standard.com>

**Input file required with details of sheets –**

****

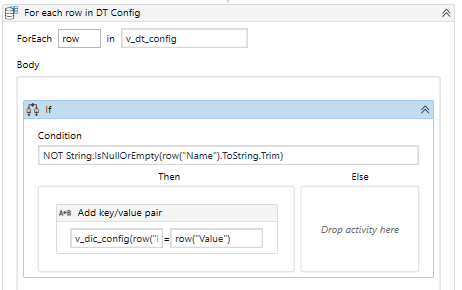
* ***Newspaper Input*** – Contains the name of newspaper website, URL for each tab of the website, tab name, parent node, base URL (main URL of the website), highlight node, and highlight node type. (Nodes part will be explained in the next part of the document)
* ***Config*** – Contains from, to, cc, bcc email ids whom we need to send newsletter, SMTP details if required, email subject, header sequence, repository path of the output files.
* ***Keywords*** – Contains valid keyword, invalid keywords, and classification of headers. Valid and excluded columns are independent of each other.

**Step 1: Make your environment clean [Follow the best practices]**

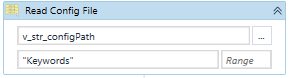
1. Close excel, other unwanted applications. Make sure it won’t impact user’s work in case of attended robot. Check output repository is created or not, If not create it.
2. Read Input file mentioned above. It contains 3 sheets.
   1. Read Newspaper Input sheet as datatable. [Output variable - *v\_dt\_inputDT*]



* 1. Read Config sheet as dictionary. [Output variable - *v\_dic\_config*]

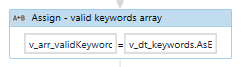


* 1. Read Keywords file as Datatable. [Output variable - *v\_dt\_keywords*]



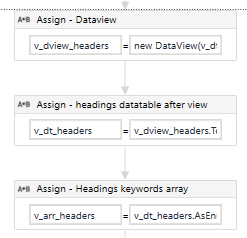
**Step 2: Collect all the required data from input sheet, Initialize datatable wherever required. [Keep variable scope to complete project]**

1. Take valid keywords in array of strings from keyword sheet.



***(Array of String)*** *v\_arr\_validKeywords = v\_dt\_keywords.AsEnumerable().Select(Function(r) r.Field(Of String)(0)).ToArray().Where(Function(r) Not String.IsNullOrWhiteSpace(r)).ToArray* ***where 0 stands for index***

1. Take unique headings in array of strings from keyword sheet.



***(Dataview)*** *v\_dview\_headers = new DataView(v\_dt\_keywords)*

***(Datatable)*** *v\_dt\_headers = v\_dview\_headers.ToTable(true, "Headers")*

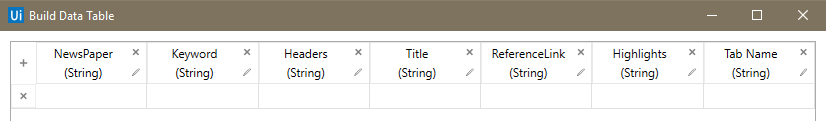
***(Array of String)*** *v\_arr\_headers = v\_dt\_headers.AsEnumerable().Select(Function(r) r.Field(Of String)(0)).ToArray().Where(Function(r) Not String.IsNullOrWhiteSpace(r)).ToArray*

1. Take invalid keywords in array of strings from keyword sheet.



***(Array of String)*** *v\_arr\_invalidWords = v\_dt\_keywords.AsEnumerable().Select(Function(r) r.Field(Of String)(2)).ToArray().Where(Function(r) Not String.IsNullOrWhiteSpace(r)).ToArray* ***where 2 stands for index***

1. Build datatable to store the data extracted. [Output variable - *v\_dt\_outputDT*]



1. Check if master file exists – In this master file, we can append daily data. Hence at the first time when we will run this automation solution, this file will be created. Purpose of this file is to not send repeated news. On each day, we will refer to this file against each news extracted from website. If it’s already there, we will not sent it over the email newsletter.
   1. *If file available in directory* – Read the file in datatable [Output variable - *v\_dt\_prevData*]
   2. *If not available* – Just build a datatable with name *v\_dt\_prevData* similar to point no. 4 datatable schema.

**Step 3: Extraction of data from newspaper websites.**

**[Complete step 3 is in for each row of Newspaper Input sheet]**

1. Create 7 string variables to assign a value of each column inside for each row.

*v\_str\_newspaperName = row("NewsPaper").ToString*

*v\_str\_newspaperURL = row("URL").ToString*

*v\_str\_parentNode = row("Parent Node").ToString*

*v\_str\_baseWebsite = row("Base").ToString*

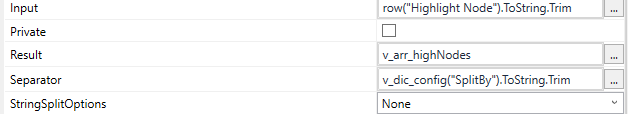
*v\_str\_highlightNode = row("Highlight Node").ToString*

*v\_str\_highlightNodeType = row("Highlight Node Type").ToString*

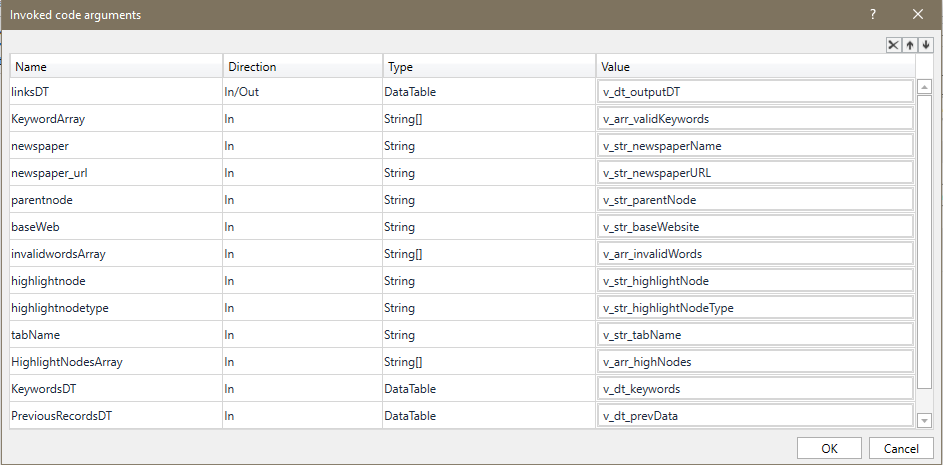
*v\_str\_tabName = row("Tab Name").ToString*

1. Split v*\_str\_highlightNode* by “Hnsplit” String. [Output - *v\_arr\_highNodes*] Purpose for this will be explained in the next step.

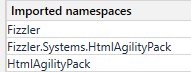


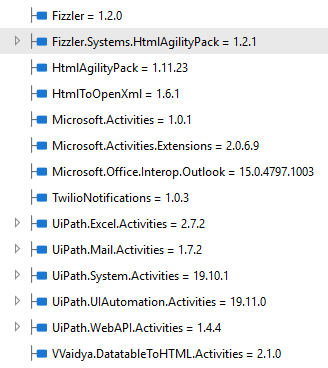


1. Invoke VB.net code with below arguments.



Import the required namespaces/packages in uipath project –





**Why we are using these packages?**

***Fizzler*** - Fizzler is a W3C Selectors parser and generic selector framework for document hierarchies. [<https://www.nuget.org/packages/Fizzler/>]

***HtmlAgilityPack*** - HAP is an HTML parser written in C# to read/write DOM and supports plain XPATH or XSLT. [<https://html-agility-pack.net/>]

Using these packages, we can load the website page, get the html content of the page and play with the selectors. We need not open the browser or install anything different.

**What important input parameters are required for VB code?**

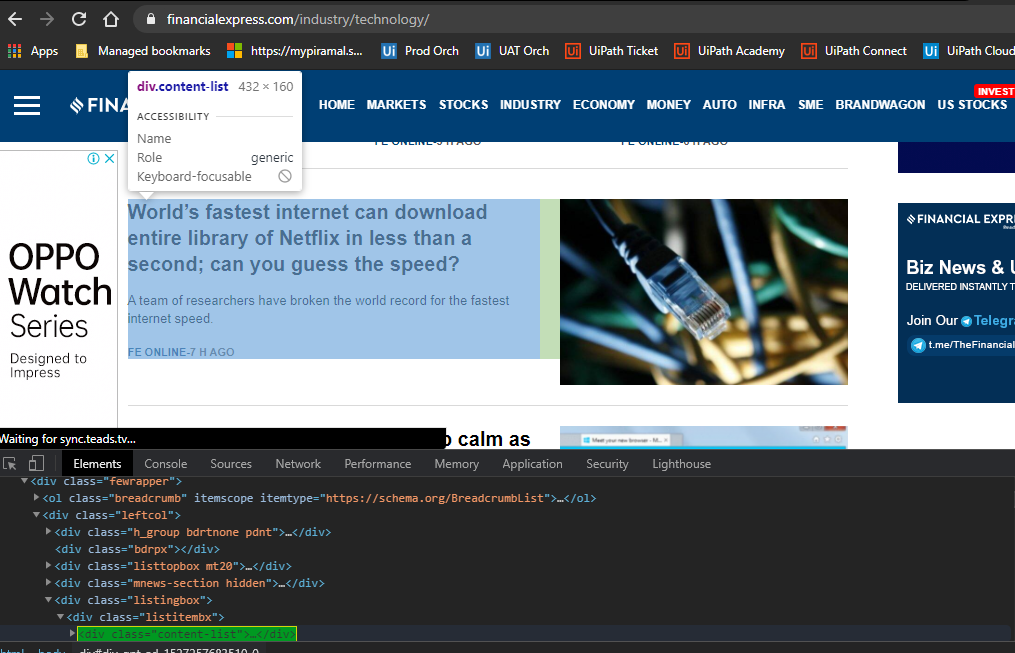
**Newspaper name** (*v\_str\_newspaperName*) – To know news is extracted from which newspaper website.

**URL** (*v\_str\_newspaperURL*) – URL of specific tab of the newspaper website. We need to navigate to this URL, extract the news title on the basis of keyword, parent node and highlight node.

**Tab name** (*v\_str\_tabName*) – Tab name of the above URL. Just for reference and to keep in our master file.

**Parent node** (*v\_str\_parentNode*) – Parent node is the node where we will find all news title from the website. [To get this parent node, open developer’s tool bar (Press F12). Inspect 4-5 news title, analyze which node is common and put that node in excel sheet.]

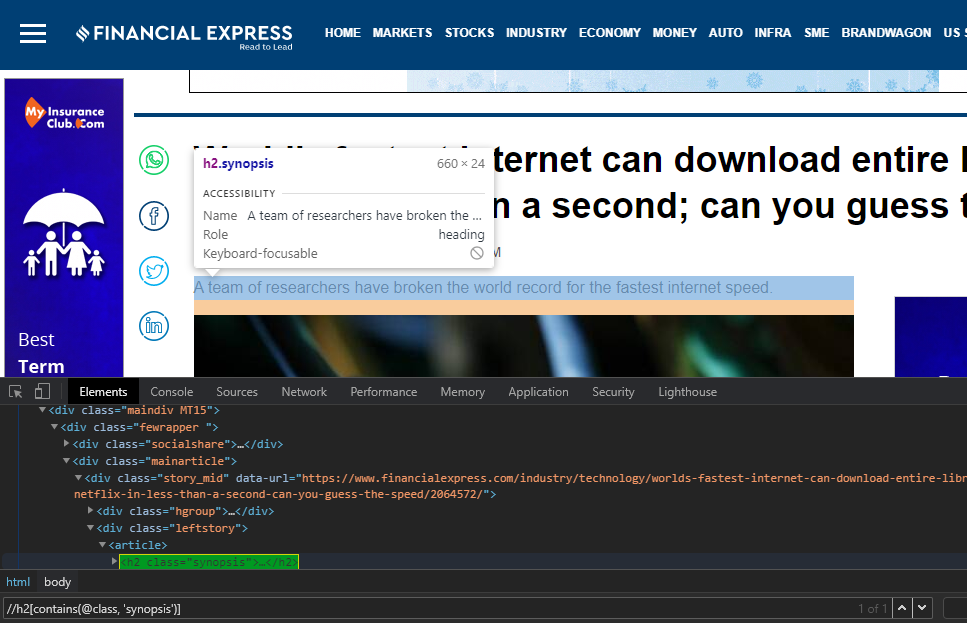
Below example, refer to the first record from Newspaper input sheet.



**Valid Keywords** (*v\_arr\_validKeywords*) – the news title which we will get from parent nodes, we will check if that title contains any keyword from this list.

**Invalid Keywords** (*v\_arr\_invalidWords*) – This is for second level of check. From the filtered news, we can check if there are any invalid keywords from this list.

**Highlight node** (*v\_str\_highlightNode*) – To get the summary of the news (1st paragraph/ highlights of the news) we need this highlight node. [To get this highlight node, open developer’s tool bar (Press F12). Inspect 4-5 news summary of highlights of the news, analyze which node is common and put that node in excel sheet.] There is chance that we get multiple selectors combination for highlights of the news. So we keep all selectors with separator “Hnsplit”. Below example, refer to the first record from Newspaper input sheet.



**Highlight Node Type** (*v\_str\_highlightNodeType*) – While extracting highlights of the news, possibility that there is highlights with more than one point. In such case, we mention Multiple in excel sheet, otherwise it is Single. Please analyze the pattern of highlights for this, we need to be sure on this.

**Previous data from Master file** (*v\_dt\_prevData*) – To check if extracted news is already available in master file.

**VB Code attached here with line by line comments** – (Open in notepad++ with visual basic language)



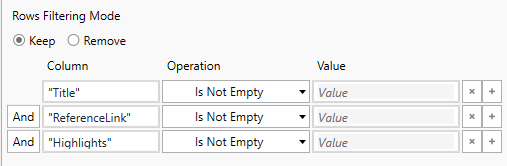
**Step 4: Data massaging on the output datatable from above VB code.**

1. Remove empty rows on below column conditions. [Output variable same as input variable - *v\_dt\_outputDT*]

Title

ReferenceLink

Highlights



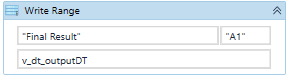
1. Remove duplicate rows [Output variable same as input variable - *v\_dt\_outputDT*]



1. Remove duplicate titles from output datatable [Output variable same as input variable - *v\_dt\_outputDT*]

*v\_dt\_outputDT = v\_dt\_outputDT.AsEnumerable().GroupBy(function(x) x.Field(Of String)("Title")).Select(function(y) y.First()).CopyToDataTable()*

1. Write this datatable in output repository with todays date [Give your own naming convention\_todays date]



1. Append same data in Master file.
   1. Condition 1 – If master file is not created, create master file and write data.
   2. Condition 2 – If master file is already created, append data

**Step 5: If output datatable contains any data then only go ahead with step 5, or else send exception email saying new news not available.**

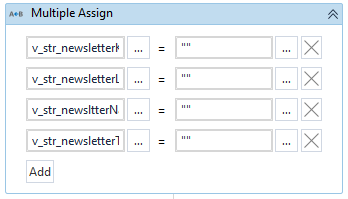
1. Create 4 String variables [these variables are used for storing the data and pass it to html string to create the newsletter]

*v\_str\_newsletterKeyword –* To store the keyword matched

*v\_str\_newsletterLink –* To store the link of the news

*v\_str\_newsltterName –* To store the newsletter name

*v\_str\_newsletterTitle –* To store the title of the news



1. Initialize/create the html string variables with default values. Please note – I have picked the html source of one of the newsletter and split it into parts. Purpose is to make the content of it dynamic using for each row loop on our datatable. Attaching a text file here, which contains all the default values of the html string variable.



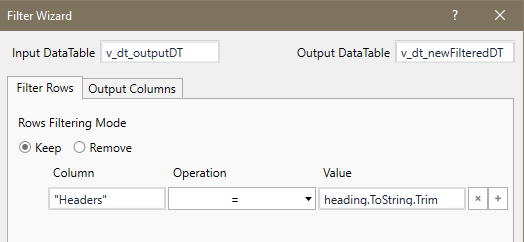
1. For each loop – [For each heading in ***v\_dic\_config("Headers Sequence").ToString.Split(new char(){","c})*]**

For Each No. 1

[Note - If you refer the config sheet of input file, by using “Headers Sequence” we can define the priority like which header news we want on the top. Note – these header names and headers in Keywords should be exactly same.]

Inside this for each loop –

1. Create a new datatable. *v\_dt\_newFilteredDT = new datatable().*
2. Filter Data Table.



This will filter the news from output datatable on the basis of one by one headings available in “Headers Sequence”.

For example –



So this for each loop with iterate all the headings one by one, and filter it on main datatable (*v\_dt\_outputDT*) and store the result in *v\_dt\_newFilteredDT*

1. If loop - v*\_dt\_newFilteredDT.Rows.Count>0* [To check if we have news related to current heading in the loop]

Then Part below –

Assign -> *v\_str\_newsletterKeyword = heading.ToString*

Assign -> v*\_str\_finalHtmlString = v\_str\_finalHtmlString+v\_str\_htmlKeywordHeaderString.Replace("{NewsletterKeyword}",v\_str\_newsletterKeyword.ToString.ToUpper)*

For each row in *v\_dt\_newFilteredDT* – [This is for each No. 2 inside above for each]

Inside this for each part below –

Use Assign activities with values like below –

***(String)*** *v\_str\_newsltterName = row("NewsPaper").ToString*

***(String)*** *v\_str\_newsletterTitle = row("Title").ToString*

***(String)*** *v\_str\_newsletterLink = row("ReferenceLink").ToString*

***(String)*** *v\_str\_newsHighlight = row("Highlights").ToString*

***(String)*** *v\_str\_finalHtmlString = v\_str\_finalHtmlString + v\_str\_newsKeywordHeaderString.Replace("{NewsletterLink}",v\_str\_newsletterLink).Replace("{NewsletterTitle}",v\_str\_newsletterTitle).Replace("{NewsletterName}", v\_str\_newsltterName).Replace("{NewsHighlights}",v\_str\_newsHighlight)*

***v\_str\_finalHtmlString*** now contains the complete newsletter html string with our news data excluding the end part.

End of For each no. 2

Assign -> *v\_str\_finalHtmlString = v\_str\_finalHtmlString + v\_str\_endString*

Now we have complete htmlstring content of newsletter with end part.

End of For each no. 1

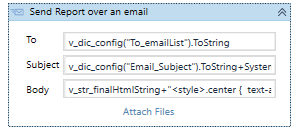
1. Switch case for deciding whether to send newsletter over an email by outlook or SMTP. This will be the value in config sheet.



If Case type is Outlook – Use Send Outlook Mail Message activity. Pass below string in body. Make sure IsBodyHtml is checked in properties.

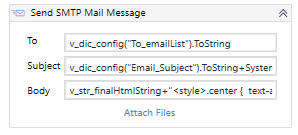
*v\_str\_finalHtmlString+"<style>.center { text-align: center;}</style><div class='center'> <p>For any doubts or queries, Please write to "+v\_dic\_config("COE\_emailID").ToString+"</p></div>"*

Pass to, cc, bcc, from email ids, subject from config file.



If Case type is SMTP - Use Send SMTP Mail Message activity. Pass above string in body. Make sure IsBodyHtml is checked in properties.

Pass to, cc, bcc, from email ids, subject, SMTP details from Config file.



**Step 6: Keep the complete code in try catch, use the same switch logic for sending customized exception email in catch block.**

Outputs are:

1. The newsletter which will be sent over an email. (Attached separately.)
2. Daily newsletter file which will be created Daily –



1. Master Data file in which we will append daily data.



So, that’s about the newsletter automation. It hardly takes 5 minutes for execution to navigate through all URL’s and extract the news. Do try it ☺

**Advantages:**

* Headless Browser Automation
* No different installations needed
* In order to categorize whether its sports related, economy related we can also use AI fabric with this project.
* Can run attended as well as unattended.
* We can add as much as keywords, URLs, Newspapers in config file, it will still work without even touching the code. However, efforts for finding parent node and summary nodes and keep that in excel sheet is still there, but that’s the minimal effort we will need.