

Jeff Dredge
MBA 614

Executive Summary

The purpose of this project was to allow BYU students such as myself to easily access the Wall Street Journal from the Factiva website. BYU has a subscription to Factiva's online content, which allows all students to access the content including the Wall Street Journal. Unfortunately, the service is difficult to browse to and through.

By the click of one button, anyone who has a BYU NetID can easily download the Journal into Excel. The program I wrote accesses the internet, downloads each article and places each one into a separate worksheet. The program then places the text of the article into a textbox, formats the article for readability. Finally, the program creates a table of contents with hyperlinks to each article and hyperlinks in each article that can be used to return to the table of contents. When the user exits the file, the workbook is restored to its original state, and is ready for use again.

Implementation

Button:

The code starts with a macro ("Button") that is run when the user clicks a button that shows up when the user opens the document. The macro uses a "For Each" loop to make all tabs visible and then uses another "For Each" loop to delete all tabs other than the tab titled "WSJ", which is the tab that has the button used to retrieve the Wall Street Journal. Finally, the macro shows a form ("OnCampus") asking the user if you are on campus or not.

frmOnCampus:

The "OnCampus" form asks the user, "Are you on Campus?" If the user is not on campus, a second user form is displayed when the "Yes" button is clicked. This second user form ("frmPassword") allows the user to enter a password and NetID, which is used to login and give the user access to Wall Street Journal via Factiva. If the user is on campus, a NetID and password are not necessary to access Factiva data. After the user clicks either "Yes" on the "OnCampus" form or clicks "Sign In" on the Password form, a macro titled GetData is initiated.

GetData:

I have actually used two separate macros to access the internet and download information—"GetDataOnCampus" and "GetDataOffCampus". Both of these macros do essentially the same thing; the only difference is that the "GetDataOffCampus" macro uses the NetID and password provided by the user form to log in if the user is off campus.

The "GetData" macros use Agent1 provided by Professor Allen to access the HBLL library page with a link to Factiva. Using the "followLinkByHref" function, the agent follows the link to the Factiva website. Next, using the "followLinkByText" function, the agent follows a link on the Factiva website link to go to

After the macro is finished the program returns to the “GetData” macro, which promptly runs another macro titled “GetLinks”.

GetLinks

The “GetLinks” macro imports the text file containing the HTML code for the web page shown previously in **Exhibit 1**. This data is used to pick out the links from the page, which are used to access articles.

I had some difficulties accessing the drop down menus that allow the user to choose different dates and different pages of the Wall Street Journal. I went to Professor Allen to get help using the agent to follow links in the drop down menus; however, after some time he was unable to help me. As such, I was unable to download all of the day’s articles. I was only able to download the “Front Page” articles that are displayed as a default on the web page.

The “GetLinks” macro uses Excel’s built in functionality to import the “source2” text file saved earlier and paste it into cell “A1”. Using xlDown and xlUp to go to the last row containing text; I then set the row number of the cell as an integer variable labeled “y”. I use the “y” variable in a “For x = 1 to y” loop to iterate through each row of the HTML in search of references to pages that include the letters “J000”, which is the beginning of a 25 digit numbering system used to label each Wall Street Journal article (see code in **Exhibit 3**).

Exhibit 3

```
For x = 1 To y
    z = InStr(1, ActiveCell.Value, "J000")
    If z <> 0 Then
        Range("b1").Offset(x, 0) = Mid(ActiveCell.Value, z, 25)
        ActiveCell.Offset(1, 0).Select
    Else
        ActiveCell.Offset(1, 0).Select
    End If
Next
```

If the row in question contains a reference to an article’s number, the code pastes the number in a temporary holding spot (the first blank cell in row “B” of the same page) to be used by “Agent1” later. Before the macro is finished running, the active cell is set to the top of the list of references (cell “B1”).

At this point, the “GetLinks” macro is finished running, and the program returns to the “GetData” macro. The “GetData” macro then uses “Agent1” to access the web page containing the article referenced by the 25 digit code at the top of the list of references found. I noticed that all of the article web pages began with the same prefix and ended with the 25 digit reference code. So I pass the “Agent1”

“followLinkByHref” function a variable (“x”) that contains the standard prefix combined with the 25 digit reference code selected.

Once the page containing the specified article is loaded, I pass variable “y” (y is the 25 digit code) to the “Agent1” “importPage” function. “Agent1” then imports the pages content into a new tab titled by the 25 digit code. The program then continues on to run a macro titled “TextBox”

TextBox:

The purpose of the “TextBox” macro is to create a textbox, past the article in the textbox, and format the sheet to look nice. The reason I used a textbox is because each cell can only show a certain number of characters, and the contents of one article cannot be shown in just one cell.

First, this macro creates a text box, positions it, names it, and changes the font style and size. The rest of the macro deals with locating the article contents and pasting it into the text box.

When pasted into excel, each article has a similar layout. Each article has the text “Copyright” located near the top of the article, and the unique 25 digit article code near the bottom of the article. I use these to identify where the body and title of the article are in the imported data.

Using the search function of Excel, I locate the name of the article and past it into the “Table of Contents” tab . To do this, I used a “For loop” to ensure the link was placed in the first empty cell in the “Table of Contents tab starting at cell “E7”. After the link has been created, the macro continues on to paste the article into a textbox.

When the web content is pasted into the spreadsheet, each paragraph of the article is pasted into a separate cell that is two rows below the previous paragraph. In order to gather the contents of the article and paste them into the textbox, I wrote a “For loop” to cycle through the lines of the article and copy the lines with article contents into the textbox one line below the data already in the text box (if any). Once the entire article has been pasted into the textbox the macro auto sizes the textbox to show the entire article.

Once the textbox contains the article, the macro continues on to run a macro titled “Format”

Format:

The “Format” macro makes the tab look similar to **Exhibit 1** only the text box is shown below the heading. To write the formatting codes, I merely recorded myself formatting the sheet.

Once the “Format” macro is finished running, the program returns to the “TextBox” macro. Once the first article has been placed into a new tab and formatted, the “TextBox” macro returns to the “GetData” macro which continues its “Do loop” to move onto the next article listed in the list of articles identified in the HTML. When all of the articles have been placed into separate tabs, the program returns a final time to “GetData” macro. The “GetData” macro finishes up by moving the “Table of Contents” tab to the first tab position and activates it.

Auto_Close

The final macro in the program is initiated when the Excel file is closed by the user. The macro uses two separate “For each” loops. The first loop makes all of the sheets in the workbook visible, and the second loop deletes all of the tabs except for the original tab displayed when the workbook was opened—the page that contains the button that runs the entire program.

Learning and Conceptual Difficulties

While writing this program, I learned a number of new skills and struggled through several things, which I will discuss briefly in the next paragraphs.

One of the first difficulties I ran into was using “Agent1” to navigate through the internet. Before this project, the only thing I knew how to do was open a web page and use the web query to download that particular page. I had to read the code in “Agent1” to discover the “followLinkByText” and “followLinkByHref” functions. Once I discovered these functions, I was able to successfully browse through the internet.

The biggest issue I ran into was reading and understanding HTML. As mentioned previously, even with the professor’s help, I was unable to access the drop down menus on the web page that would have allowed me to download articles from the various sections of the Wall Street Journal and from various days. Unfortunately, the program only downloads the articles that are shown on the front page of the Journal; although, I had originally planned on downloading the entire Journal.

Another issue I ran into was working with textboxes in excel. When a text box is inserted into a worksheet, Excel assigns it a numerical name. Because this name changes with each new textbox added, I had to learn how to name the textbox so I could reference it in my code. This took some searching on the internet, but I was able to learn how to name a textbox by setting “ShapeRange.Name” equal to a specified name.

Previously, I had never used the “For each” loop; however, I had to learn how to use these types of loops to cycle through worksheets in the workbook.

I also learned how to initiate macros on the opening and closing of the work book by using the “Auto_Close” and “Workbook_Open”. Although I didn’t end up using “Workbook_Open” in the code, I’m sure this will be useful in the future.

Overall, this project was a very beneficial learning experience for me and it gave me many useful tools that I hope to use in the future.