

Final Project

ISYS 540

Chris Bateman

The Problem

Several times in the course of my employment as a web developer, I have been required to retrieve a relatively massive list from the internet and either convert it into a XML document or insert it into a SQL database. Both situations have caused a fair amount of headache for me. My typical process involved highlighting the table, copying it and pasting it into Excel, and then spending 30 minutes coming up with some crazy set of Excel functions to generate the desired output.

Example: http://www.loc.gov/standards/iso639-2/php/code_list.php

Library of Congress >> Standards

ISO 639.2

Registration Authority

ISO 639-2 HOME - Code List - Changes to Codes

ISO 639 Joint Advisory Committee - ISO 639-1 Registration Authority

Codes for the Representation of Names of Languages

Codes arranged alphabetically by alpha-3/ISO 639-2 Code

Note: ISO 639-2 is the alpha-3 code in *Codes for the representation of names of languages-- Part 2*. There are 21 languages that have alternative codes for bibliographic or terminology purposes. In those cases, each is listed separately and they are designated as "B" (bibliographic) or "T" (terminology). In all other cases there is only one ISO 639-2 code. Multiple codes assigned to the same language are to be considered synonyms. ISO 639-1 is the alpha-2 code.

ISO 639-2 Code	ISO 639-1 Code	English name of Language	French name of Language
aar	aa	Afar	afar
abk	ab	Abkhazian	abkhaze
ace		Achinese	aceh
ach		Acoli	acoli
ada		Adangme	adangme
ady		Adyghe; Adygei	adyghé
afa		Afro-Asiatic languages	afro-asiatiques, langues
afh		Afrihili	afrihili
afr	af	Afrikaans	afrikaans
ain		Ainu	ainou
aka	ak	Akan	akan
akk		Akkadian	akkadien
alb (B)	sq	Albanian	albanais

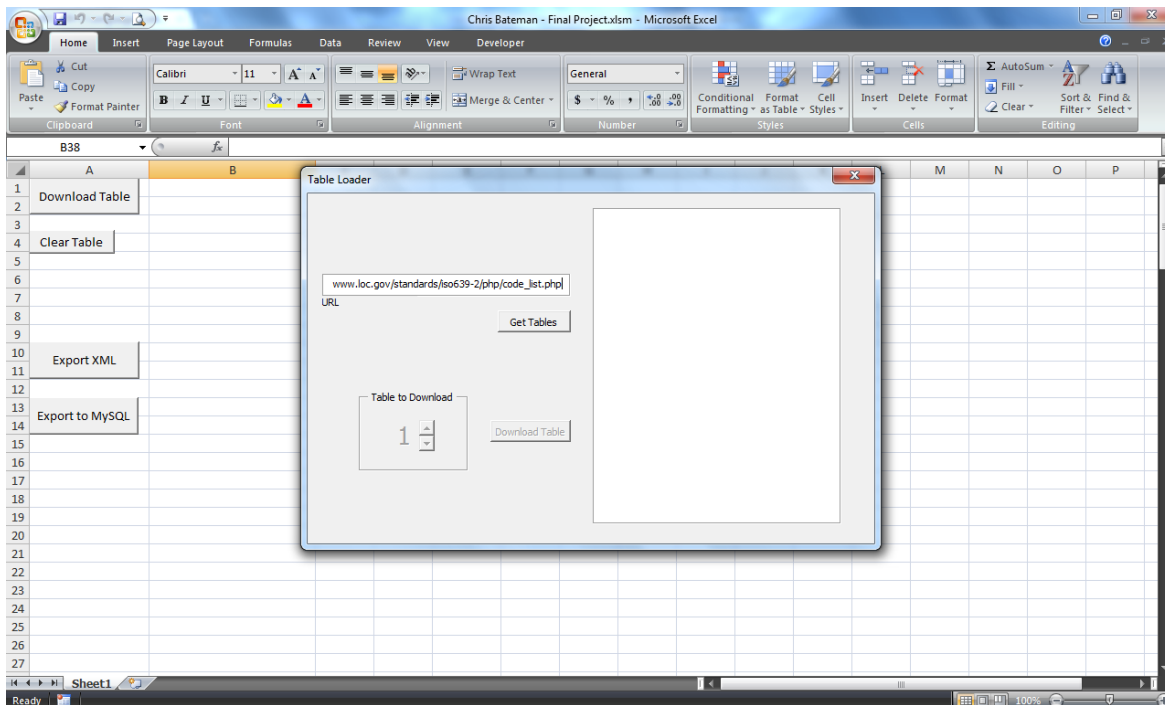
The Solution

Overview

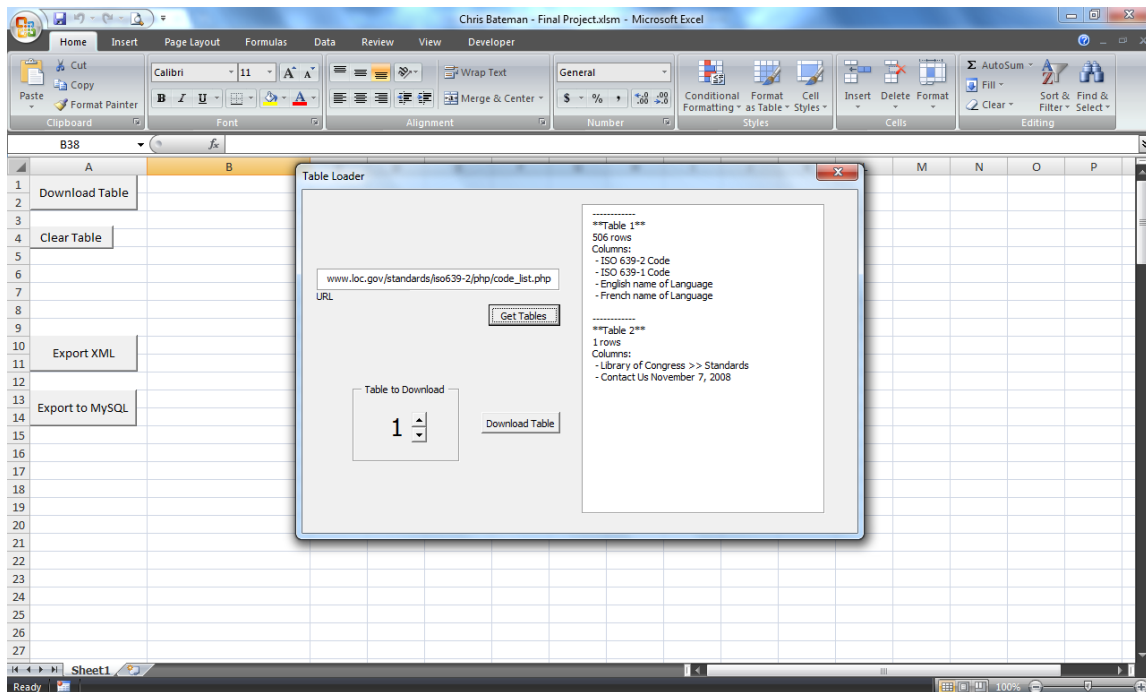
I created an Excel workbook and VBA script which will parse a HTML page on the internet and find all of the tables on the page. The user can choose which table to download. After downloading the table to the spreadsheet and making any desired adjustments, the user can choose to either export the data as XML or to insert the table into a MySQL database (given a MySQL server, database, username, and password).

The Process

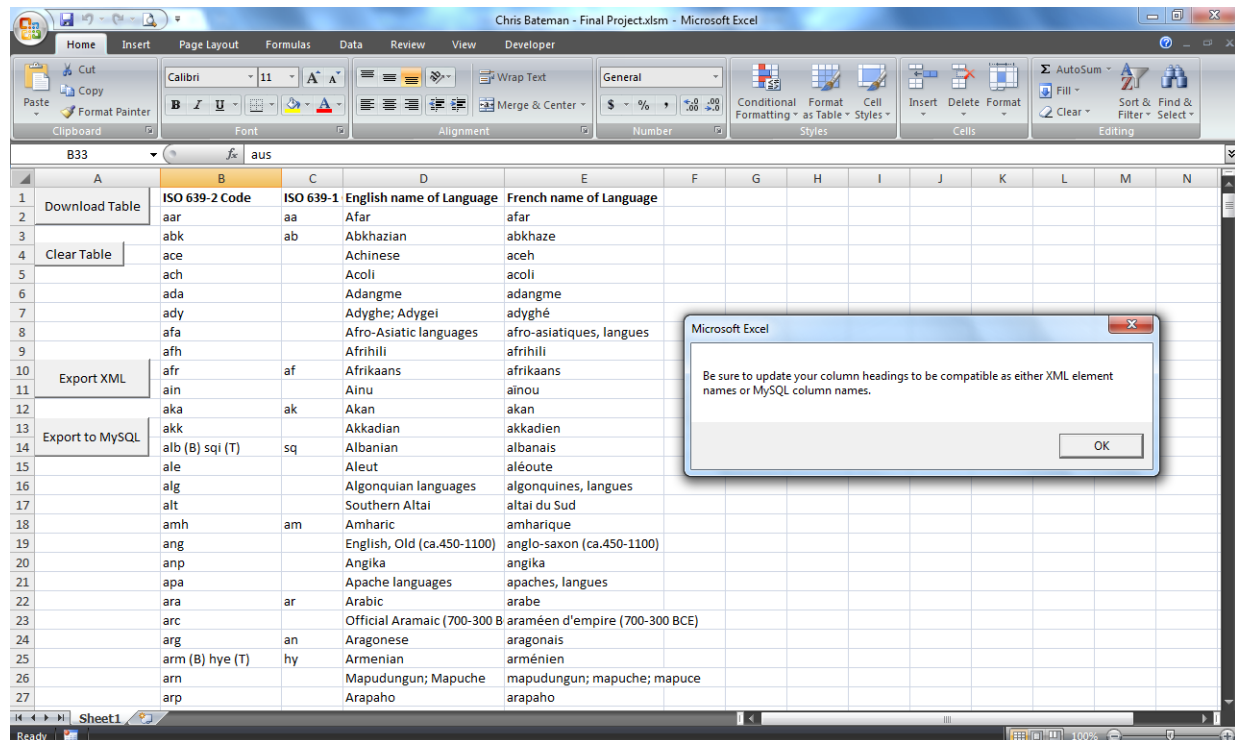
The user begins by clicking “Download Table”. This brings up a window in which the URL of the page to be scraped can be entered.



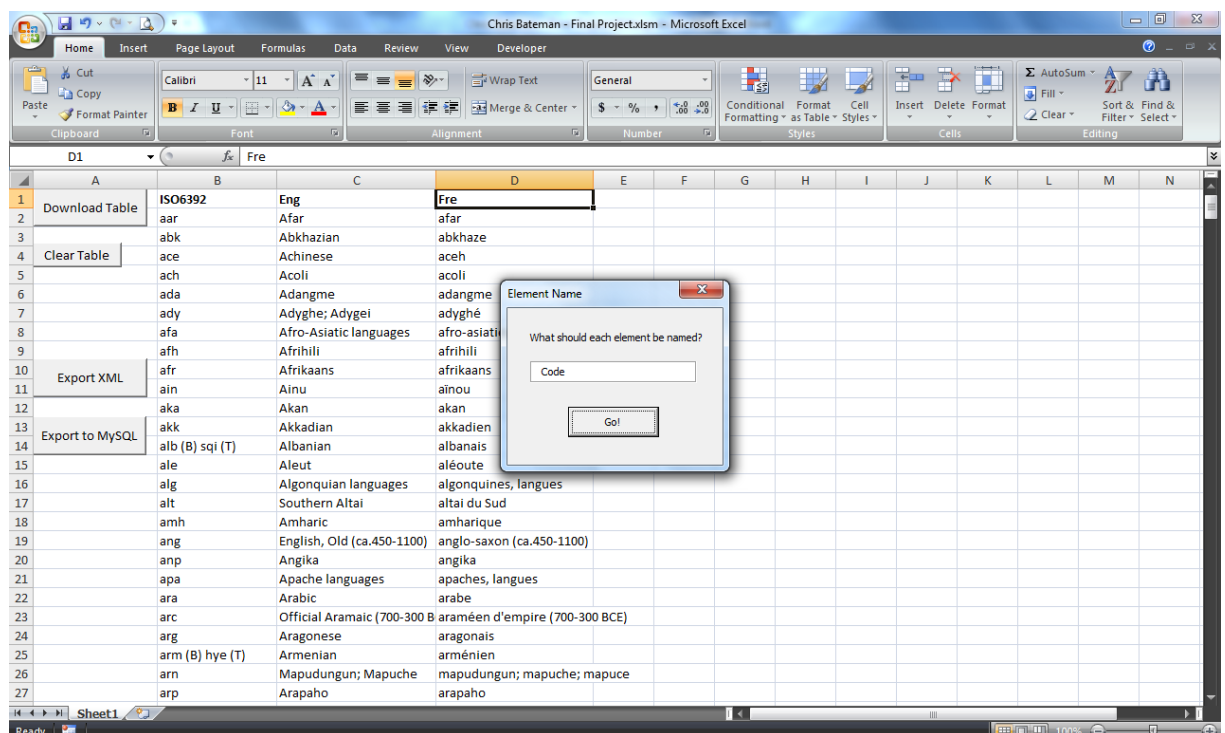
After clicking “Get Tables” a listing of the tables found on the page will appear. The number of rows in each table will be listed, along with the column names. A number spinner will be activated, which the user may use to select the number of the table he wants to download. Only valid table numbers will be able to be selected.



When the user clicks “Download Table” the table will be displayed in the spreadsheet, along with a warning that the user must change the column names so that they are compatible as either an XML element or a MySQL table name.



After updating the column names and removing any undesired columns, the user can choose to export the table to either XML or a MySQL database. When choosing “Export XML” the user will be prompted to enter a name for the XML element which will compose each item in the table.



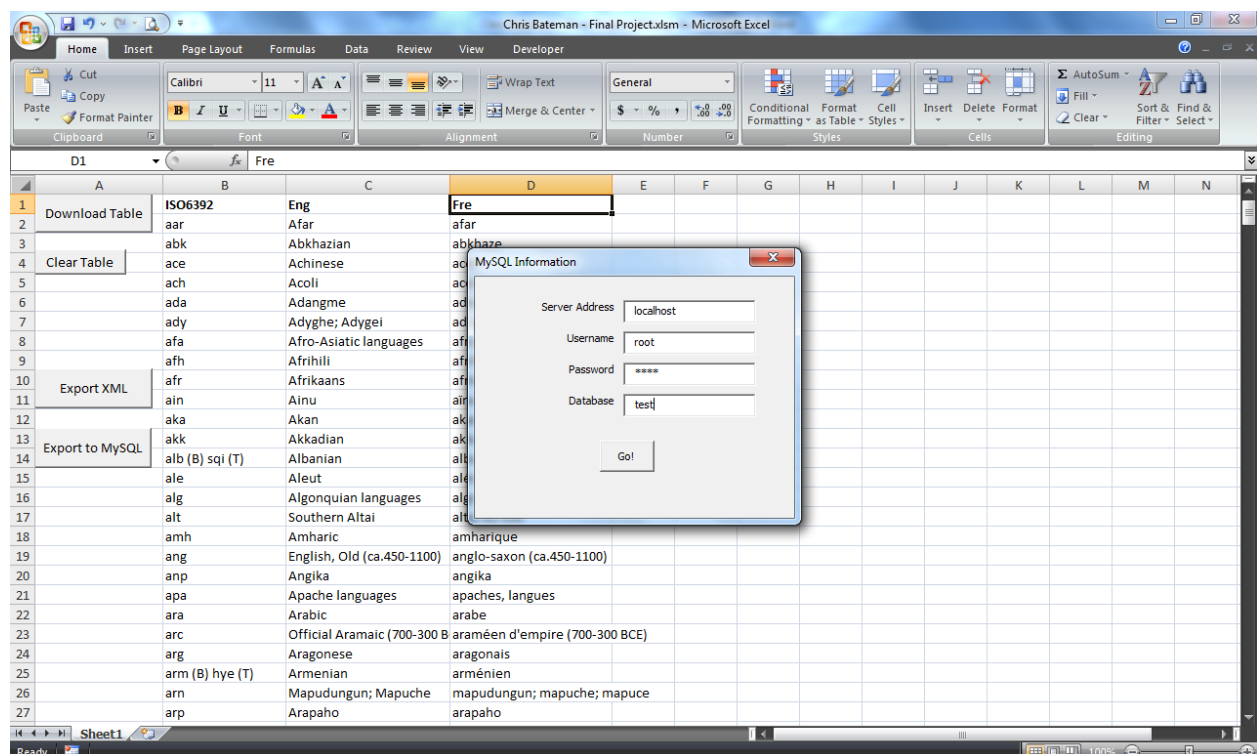
After clicking “Go!” the user will find the exported XML file in the same directory as the workbook. The XML file is formatted for readability.

```

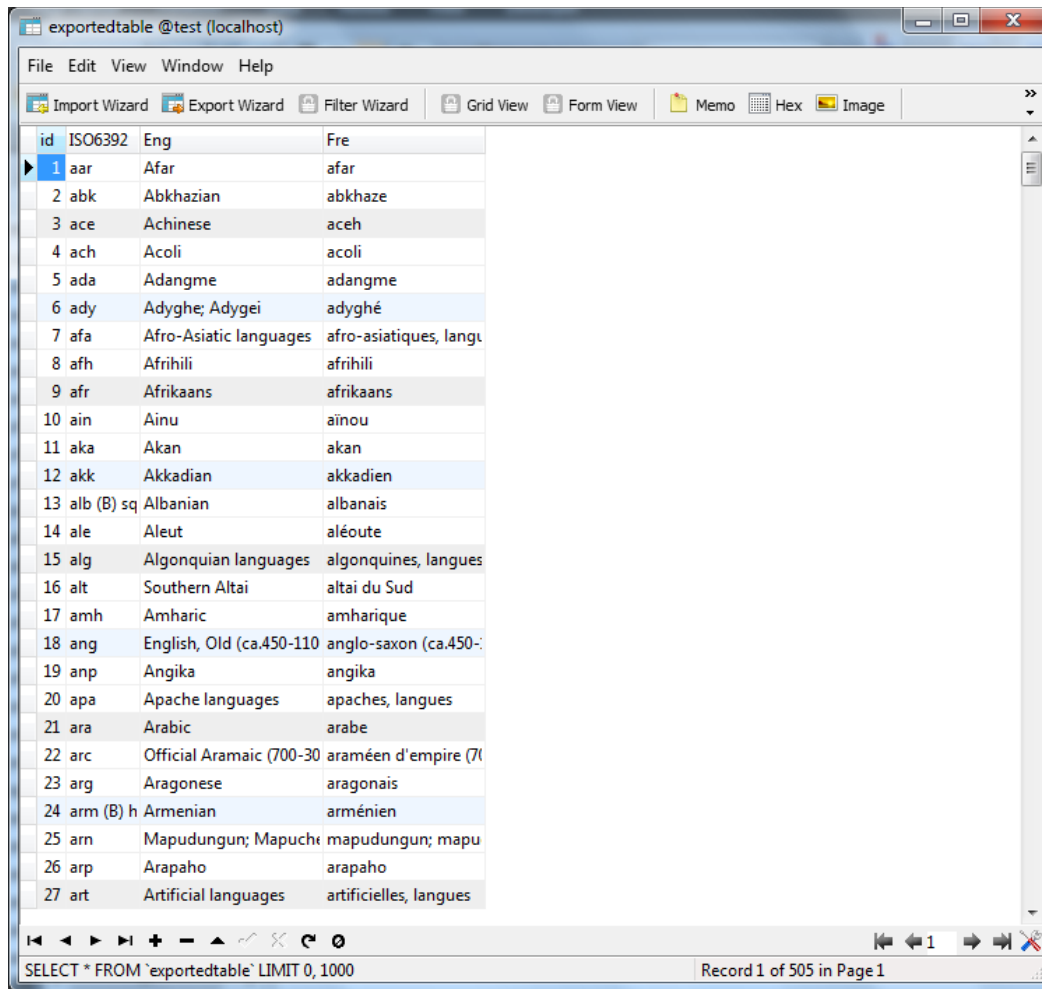
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <Root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
3   <Code>
4     <ISO6392>aar</ISO6392>
5     <Eng>Afar</Eng>
6     <Fre>afar</Fre>
7   </Code>
8   <Code>
9     <ISO6392>abk</ISO6392>
10    <Eng>Abkhazian</Eng>
11    <Fre>abkhaze</Fre>
12  </Code>
13  <Code>
14    <ISO6392>ace</ISO6392>
15    <Eng>Achinese</Eng>
16    <Fre>aceh</Fre>
17  </Code>
18  <Code>
19    <ISO6392>ach</ISO6392>
20    <Eng>Acoli</Eng>
21    <Fre>acoli</Fre>
22  </Code>
23  <Code>
24    <ISO6392>ada</ISO6392>
25    <Eng>Adangme</Eng>
26    <Fre>adangme</Fre>
27  </Code>
28  <Code>
29    <ISO6392>ady</ISO6392>
30    <Eng>Adyghe; Adyghe</Eng>
31    <Fre>adygh</Fre>
32  </Code>
33  <Code>
34    <ISO6392>afa</ISO6392>
35    <Eng>Afro-Asiatic languages</Eng>
36    <Fre>afro-asiatiques, langues</Fre>
37  </Code>
38  <Code>
39    <ISO6392>afh</ISO6392>
40    <Eng>Afrihili</Eng>
41    <Fre>afrihili</Fre>
42  </Code>
43  <Code>
44    <ISO6392>ain</ISO6392>
45    <Eng>Ainu</Eng>
46    <Fre>ainu</Fre>
47  </Code>
48  <Code>
49    <ISO6392>aka</ISO6392>
50    <Eng>Akan</Eng>
51    <Fre>akan</Fre>
52  </Code>
53  <Code>
54    <ISO6392>akk</ISO6392>
55    <Eng>Akkadian</Eng>
56    <Fre>akkadian</Fre>
57  </Code>
58  <Code>
59    <ISO6392>alb</ISO6392>
60    <Eng>Albanian</Eng>
61    <Fre>albanian</Fre>
62  </Code>
63  <Code>
64    <ISO6392>ale</ISO6392>
65    <Eng>Aleut</Eng>
66    <Fre>aleut</Fre>
67  </Code>
68  <Code>
69    <ISO6392>alg</ISO6392>
70    <Eng>Algonquian languages</Eng>
71    <Fre>algonquian, langues</Fre>
72  </Code>
73  <Code>
74    <ISO6392>alt</ISO6392>
75    <Eng>Southern Altai</Eng>
76    <Fre>altai</Fre>
77  </Code>
78  <Code>
79    <ISO6392>amh</ISO6392>
80    <Eng>Amharic</Eng>
81    <Fre>amharique</Fre>
82  </Code>
83  <Code>
84    <ISO6392>ang</ISO6392>
85    <Eng>English, Old (ca.450-1100)</Eng>
86    <Fre>anglo-saxon (ca.450-1100)</Fre>
87  </Code>
88  <Code>
89    <ISO6392>anp</ISO6392>
90    <Eng>Angika</Eng>
91    <Fre>angika</Fre>
92  </Code>
93  <Code>
94    <ISO6392>apa</ISO6392>
95    <Eng>Apache languages</Eng>
96    <Fre>apaches, langues</Fre>
97  </Code>
98  <Code>
99    <ISO6392>ara</ISO6392>
100   <Eng>Arabic</Eng>
101   <Fre>arabe</Fre>
102 </Code>
103 <Code>
104   <ISO6392>arc</ISO6392>
105   <Eng>Official Aramaic (700-300 BCE)</Eng>
106   <Fre>araméen d'empire (700-300 BCE)</Fre>
107 </Code>
108 <Code>
109   <ISO6392>arg</ISO6392>
110   <Eng>Aragonese</Eng>
111   <Fre>aragonais</Fre>
112 </Code>
113 <Code>
114   <ISO6392>arm</ISO6392>
115   <Eng>Armenian</Eng>
116   <Fre>arménien</Fre>
117 </Code>
118 <Code>
119   <ISO6392>arn</ISO6392>
120   <Eng>Mapudungun; Mapuche</Eng>
121   <Fre>mapudungun; mapuche; mapuce</Fre>
122 </Code>
123 <Code>
124   <ISO6392>arp</ISO6392>
125   <Eng>Arapaho</Eng>
126   <Fre>arapaho</Fre>
127 </Code>
128 </Root>

```

If the user chooses to export the table to a MySQL database, he will be presented with a dialog in which to enter the server address, username, password, and database.



The script will create a new SQL database on the chosen server and insert each item into it.



id	ISO6392	Eng	Fre
1	aar	Afar	afar
2	abk	Abkhazian	abkhaze
3	ace	Achinese	aceh
4	ach	Acoli	acoli
5	ada	Adangme	adangme
6	ady	Adyghe; Adygei	adyghé
7	afa	Afro-Asiatic languages	afro-asiatiques, langu
8	afh	Afrihili	afrihili
9	afz	Afrikaans	afrikaans
10	ain	Ainu	ainou
11	aka	Akan	akan
12	akk	Akkadian	akkadien
13	alb (B) sq	Albanian	albanais
14	ale	Aleut	aléoute
15	alg	Algonquian languages	algonquines, langues
16	alt	Southern Altai	altai du Sud
17	amh	Amharic	amharique
18	ang	English, Old (ca.450-110	anglo-saxon (ca.450-
19	anp	Angika	angika
20	apa	Apache languages	apaches, langues
21	ara	Arabic	arabe
22	arc	Official Aramaic (700-30	araméen d'empire (70
23	arg	Aragonese	aragonais
24	arm (B) h	Armenian	arménien
25	arn	Mapudungun; Mapuche	mapudungun; mapu
26	arp	Arapaho	arapaho
27	art	Artificial languages	artificielles, langues

That's it! The process was designed to have very few steps, allowing the user to get the desired information to its destination as quickly as possible.

Obstacles

Method of Accessing Web Page

IE Object vs. HTTP request: My initial preference was to use a HTTP request to retrieve web pages.

```
http.Open "GET", url, False  
http.Send
```

However, I quickly ran into problems, as I was without a good way to parse the HTML. I tried using regular expressions, but this is obviously a bad idea (see [this link](#)). The best option, then, was to use an internetexplorer object, and select all of the table elements like this:

```
Set tableList = ie.document.all.tags("TABLE")
```

Table Discovery

One of the challenges I ran into was the fact that many web pages unfortunately still use tables for page layout. This meant that on these pages the user was given lots of table options, many of which were nested and included other tables, including the one the user was looking for. In order to solve this problem, I simply only accept tables which contain no other table elements.

```
Set subTables = table.all.tags("TABLE")
If subTables.Length = 0 Then
...
```

Another problem was that some tables use the proper table header elements (<th>) while others used regular table elements (<td>) for the header. When I download the header names, I check first for <th> elements, and if there are none, I simply select the first row of <td> elements.

Undead IE Explorer processes

While I was developing and testing my application, I noticed my computer was getting slower. I opened up the task manager and saw that there were about 15 instances of "iexplore.exe". Here's what was happening: whenever I opened up the initial "Download Table" dialog box, an instance of Internet Explorer was initialized, and remained open until after the user had downloaded the table, at which point I called the "Quit" function. While I was testing, I would frequently open up that dialog and close it out before actually downloading the table, meaning that the "Quit" function never got called. In order to solve this problem, I had to discover the "Terminate" event. If the form is terminated in any way, I now call this line of code:

```
If IsNull(ie) Then ie.Quit
```

MySQL Connection

One of the most difficult parts of this project was figuring out how to connect to MySQL. In order to activate the ADODB class, I had to add a reference to "Microsoft ActiveX Data Objects 6.0 Library." Also, in order to connect to MySQL, I had to download "MySQL Connector / ODBC 5.1" from mysql.com. Another difficult part was inserting each item into the SQL table. I initially wrote the script so that it would string together all of the insert statements and then execute them all at once. This normally works just fine in MySQL, but for some reason, it didn't work coming from VBA. My solution was to execute each insert statement one at a time. This takes just a little bit longer, but it's not bad enough to be a problem.

Example Tables

These are some tables on the internet which demonstrate the usefulness of this VBA script:

- http://www.loc.gov/standards/iso639-2/php/code_list.php

- <http://www.ssa.gov/OACT/babynames/>
- <http://sports.espn.go.com/rpm/schedule?seriesId=1>
- http://www.iso.org/iso/support/country_codes/iso_3166_code_lists/iso-3166-1_decoding_table.htm
- <http://htmlhelp.com/reference/charset/iso160-191.html>
- <http://www.xe.com/iso4217.php>