

```

1  #!/usr/bin/perl -w
2
3  use strict;
4  use warnings;
5  use CGI;
6  use File::Basename;
7  use Text::CSV;
8  use DBI;
9
10
11 my $finalcolumn;
12 my @finalcolumns;
13 my $customcolumn;
14 my @customcolumns;
15
16 my $upload_dir = "/var/www/htdocs/tgdialer/tmp";
17
18 my $query = CGI->new();
19 my @qsnames = $query->param;
20 my $qsindex=0;
21
22 open ( LOGFILE, ">/tmp/qs-process-field-assignments.log" ) or die "$!";
23 print LOGFILE "Logfile Begins: \n";
24
25 # query string items here:
26 # address by name used in html form
27
28 my $insertcolumns="CampaignList,";
29 my $valuelist="" . $query->param("CampaignList") . ",";
30 # my $valuelist="$campaignname,";
31 my $campaignname=$query->param("CampaignList");
32 my $customdata="";
33 my $customcolname="";
34 my $queryparam="";
35
36 foreach(@qsnames)
37 {
38     if ($query->param($_) eq "")
39     {
40         $_ = "NullColumnName$qsindex";
41         # print LOGFILE "Element $qsindex: $_ = " . $query->param($_) . "\n";
42         # next;
43         # $queryparam="NULL";
44         # $queryparam=$query->param($_);
45     }
46
47     if ( ($_ =~ /Selecto/ ) && ($query->param($_) !~ /NullSelect/ ) )
48     {
49         push(@finalcolumns, $query->param($_).",".$qsindex);
50         # $queryparam=$query->param($_);
51     }
52
53
54     if ( ($_ =~ /Custom/) && ($query->param($_) !~ /NullSelect/ ) )
55     {
56         $customcolname="Custom:" . $query->param($_);
57         # push(@customcolumns, $customcolname . "," . $qsindex);
58         push(@finalcolumns, $customcolname . "," . $qsindex);
59         print LOGFILE $query->param($_) . "\n";
60         $customdata=$customdata . "," . $query->param($_);
61         $customdata =~ s/^,;//;
62         # $queryparam=$query->param($_);
63     }
64
65
66 $queryparam=$query->param($_);
67 if (! $queryparam )
68 {
69     $queryparam="NULL";

```

```

70 }
71 print LOGFILE "Element $qsindex: $_ = " . $queryparam . "\n";
72 # print LOGFILE "Element $qsindex: $_ = " . $query->param($_) . "\n";
73 # print "$_ <br>";
74 # print $cgi->param($_);
75 $qsindex ++;
76 }
77
78 print LOGFILE "$customdata \n";
79
80 print $query->header('text/html');
81 print $query->start_html('Processing Field Assignments');
82
83
84 # Process echosen columns into SQL
85 print "Predefined Columns: <br><br>";
86 foreach $finalcolumn (@finalcolumns)
87 {
88 print "$finalcolumn<br>";
89 }
90 print "<br>";
91
92
93 print "----<br>";
94
95 print "Custom Columns: <br><br>";
96 # foreach $customcolumn (@customcolumns)
97 # {
98 # print "$customcolumn<br>";
99 # }
100 print "$customdata<br>";
101 print "-----<br><br>";
102
103
104 print LOGFILE $query->param('NewFileName') . "\n";
105
106
107 my $csv = Text::CSV->new();
108 my $i = 0;
109 my $colcount = 0;
110 my $rowcount = 0;
111 my $nameindex = 0;
112 my $rowindex = 0;
113 my $filename = "";
114 my $newfilename=$query->param('NewFileName');
115 my $headstat=$query->param('headerline');
116 my $custominsert;
117 my $customvalue="";
118
119 if ( ! ($headstat) )
120 {
121 $headstat='off';
122 }
123
124 # Connect to DB
125 my $database = DBI->connect('DBI:mysql:asterisk', 'asterisk', '') or die "Couldn't
connect to database: " . DBI->errstr;
126
127 #open ( UPLOADFILE, "<$upload_dir/$filename" ) or die "$!";
128 open ( UPLOADFILE, "<$upload_dir/$newfilename" ) or die "$!";
129 while (<UPLOADFILE>)
130 {
131
132
133
134 if ($csv->parse($_))
135 {
136 my $insertcolumns="CampaignList,";
137 my $valuelist="" . $query->param("CampaignList") . ",";

```

```

138
139     my @columns = $csv->fields();
140     $colcount = $#columns + 1;
141
142     if ($. == 1)
143     {
144         my $rowindex = $.;
145
146
147         for ($i = 0; $i <= $colcount; $i++)
148         {
149             $nameindex = $i - 1;
150             if ($i == 0)
151             {
152                 next;
153             }
154
155
156         }
157
158     }
159
160     if ($headstat eq 'on')
161     {
162         $headstat='turned-off';
163         $rowindex = $. - 1;
164         next;
165     }
166
167     if ($headstat eq 'turned-off')
168     {
169         $headstat='turned-off';
170         $rowindex = $. - 1;
171     }
172     else
173     {
174         $rowindex = $.;
175     }
176
177
178     $rowcount = $rowindex;
179     # print "$rowindex:";
180
181     ### Various columns Processed here:
182     # print LOGFILE "Element $qsindex: $_ = " . $query->param($_) . "\n";
183
184     # for ($i = 0; $i < $colcount; $i++)
185     my $finalcolcount = ($#finalcolumns + 1);
186 # print LOGFILE "$finalcolcount \n";
187 my @lastcolinfo=split(',', $finalcolumns[$#finalcolumns]);
188 # print LOGFILE "@lastcolinfo \n";
189 my $lastcol=$lastcolinfo[1];
190 # print LOGFILE "$lastcol \n";
191
192     foreach $finalcolumn (@finalcolumns)
193     {
194         # print $finalcolumn."<br>";
195         my @colstosave=split(',', $finalcolumn);
196
197         if ($columns[$colstosave[1]] eq "")
198         {
199             $columns[$colstosave[1]] = "-^-" ;
200         }
201
202         if ( ($finalcolumn =~ /Custom:/) )
203         {
204             # print "fuckoff <br>";
205             # print $colstosave[0] . "-" . $columns[$colstosave[1]] . "
<br>";

```

```

206             $custominsert = $colstosave[0];
207             $custominsert =~ s/Custom:;//;
208             $customvalue = $customvalue . $custominsert . ":" . $column
ns[$colstosave[1]] . ",";
209             next;
210         }
211
212         if ($colstosave[1] == $lastcol)
213         {
214             # print "$colstosave[0],";
215             # print "$columns[$colstosave[1]] <br>";
216             $insertcolumns=$insertcolumns . $colstosave[0];
217             $columns[$colstosave[1]] =~ tr/$//d;
218             $columns[$colstosave[1]] =~ tr/,//d;
219             $columns[$colstosave[1]] =~ tr/'//d;
220             $valuelist=$valuelist . "'" . $columns[$colstosave[1]] . "'";
221         }
222         else
223         {
224             # print "$columns[$colstosave[1]],";
225             $insertcolumns=$insertcolumns . $colstosave[0] . ",";
226             $columns[$colstosave[1]] =~ tr/$//d;
227             $columns[$colstosave[1]] =~ tr/,//d;
228             $columns[$colstosave[1]] =~ tr/'//d;
229             $valuelist=$valuelist . "'" . $columns[$colstosave[1]] . "'";
230         }
231
232 # Insert Record Into Callee_Data;
233 # $database->do("insert into Callee_Data ($colstosave[0]) values ('$columns[$colsto
save[1]]')") or die "Couldn't prepare statement: " . $database->errstr;
234
235     }
236
237     $customvalue =~ s/,,$//;
238     print $customvalue . "<br>";
239     print "<br>";
240     $insertcolumns =~ s/,,$//;
241     $valuelist =~ s/,,$//;
242
243     # print "$insertcolumns";
244     # print "<br>";
245
246     # print "$valuelist";
247     # print "<br>";
248
249     $database->do("insert into Callee_Data ($insertcolumns, CustomInputData) values ($v
aluelist, '$customvalue')") or die "Couldn't prepare statement: " . $database->err
str;
250
251     $customvalue="";
252
253
254     # if ($rowindex >= 30)
255     # {
256     # last;
257     # }
258
259
260     # print LOGFILE "$finalcolcount\n";
261 }
262 else
263 {
264     my $err = $csv->error_input;
265     print "Failed to parse line: $err";
266 }
267
268 }
269 ## End of CSV File Parse
270

```

```
271
272 close UPLOADFILE;
273 close LOGFILE;
274 $database->disconnect();
275
276 print <<END_REDIRECT;
277 <h1>Database Successfully Loaded</h1>
278 <meta http-equiv="refresh" content="3;URL=/tgdialer" >
279 END_REDIRECT
280
281 print $query->end_html;
282
283
284
```