

```
1  #!/usr/bin/perl -w
2
3  # This Script Generates The Callfiles
4
5  use strict;
6  use warnings;
7  use CGI;
8  use DBI;
9  use File::Basename;
10 use Text::CSV;
11 use File::Copy;
12
13
14 my $cgi = CGI->new();
15 my @qsnames;
16 my $qsindex=0;
17 my @currentcampaignarray;
18 my @startcampaignlist;
19 my @stopcampaignlist;
20 my @deletecampaignlist;
21 my $campaign;
22
23
24 open ( LOGFILE, ">/tmp/qs-set-campaign-list.log" ) or die "$!";
25 print LOGFILE "Logfile Begins: \n";
26
27 print $cgi->header('text/html');
28 print $cgi->start_html('Activating Campaigns');
29
30
31 @qsnames = $cgi->param;
32
33 ## Print Out Query String Elements
34 foreach(@qsnames)
35 {
36     # print "Element $qsindex: $_ = " . $cgi->param($_) . "<br>";
37     print LOGFILE "Element $qsindex: $_ = " . $cgi->param($_) . "\n";
38     print "Element $qsindex: $_ = " . $cgi->param($_) . "<br>";
39     # print $cgi->param($_);
40
41
42     if ( ( $_ =~ m/ControlCampaign_*/ ) && ( $cgi->param($_) =~ m/start/ ) )
43     {
44         @currentcampaignarray=split('_', $_);
45
46         print LOGFILE "Started Campaign: $currentcampaignarray[1]\n";
47         push(@startcampaignlist, $currentcampaignarray[1]);
48     }
49
50     if ( ( $_ =~ m/ControlCampaign_*/ ) && ( $cgi->param($_) =~ m/stop/ ) )
51     {
52         @currentcampaignarray=split('_', $_);
53
54         print LOGFILE "Stopped Campaign: $currentcampaignarray[1]\n";
55         push(@stopcampaignlist, $currentcampaignarray[1]);
56     }
57
58
59     if ( ( $_ =~ m/ControlCampaign_*/ ) && ( $cgi->param($_) =~ m/delete/ ) )
60     {
61         @currentcampaignarray=split('_', $_);
62
63         print LOGFILE "Deleted Campaign: $currentcampaignarray[1]\n";
64         push(@deletecampaignlist, $currentcampaignarray[1]);
65     }
66
67
68
69 $qsindex ++;
```

```

70 }
71 # End of Query String Printout
72
73 print LOGFILE "End of Query String Printout \n\n";
74
75
76 print("<h2>Call Files Generated:</h2>");
77 # my $dailystoptime=$cgi->param("DailyStopTime");
78
79 #####
80
81 ## Begin Callfile Generation
82 # my @timezones=("pacific","mountain","central","eastern");
83 # my $timezone="";
84
85 # foreach $timezone (@timezones)
86 # {
87 # system("mkdir -p ./${timezone}");
88 # }
89 # system("mkdir -p $callfiledir");
90 # my $database;
91
92
93 my $callfiledir="/tmp/callfiles";
94 my $dropsequence;
95 my $lastdropsequence;
96 my $callfiledata;
97 my $campaignid;
98 my $record;
99 my $database;
100
101
102 system("mkdir -p $callfiledir");
103
104 # Connect to DB
105 $database = DBI->connect('DBI:mysql:asterisk', 'asterisk', '') or die "Couldn't connect to database: " . DBI->errstr;
106
107 ## Get Callee Data (Phone Number, etc)
108 foreach $campaign (@startcampaignlist)
109 {
110     print LOGFILE "Campaign to Be Activated: $campaign\n";
111     my $campaignid=$campaign;
112     my $dropsequence="";
113     my @lastdroparray="";
114     my $lastdrop="";
115
116     # Get Last Drop Sequence.
117     my $lastdropsequence = $database->prepare("select DropSequence from Campaign_{$campaignid} order by 1 desc limit 1" ) or die "Couldn't prepare statement: " . $database->errstr;
118     $lastdropsequence->execute() or die "Couldn't execute statement: " . $callfiledata->errstr;
119
120     @lastdroparray = $lastdropsequence->fetchrow_array();
121
122
123     if (! @lastdroparray )
124     {
125         $dropsequence=1;
126         print LOGFILE "LastDrop: None -- This Is First Drop\n";
127     }
128     else
129     {
130         $dropsequence=$lastdroparray[0];
131         print LOGFILE "LastDrop: $dropsequence\n";
132     }
133
134     $lastdropsequence->finish();

```

```

135
136
137
138     # Get Phone Number Etc.
139     # my @callfilearray="";
140     my $callfiledata = $database->prepare("select * from Callee_Data where Camp
aignList like '%" . $campaignid . "' ) or die "Couldn't prepare statement: " . $database-
>errstr;
141     $callfiledata->execute() or die "Couldn't execute statement: " . $callfiled
ata->errstr;
142
143
144     my @callfiledataresult="";
145
146     ## CallFile Generation Loop
147     while ( @callfiledataresult = $callfiledata->fetchrow_array()
148     # foreach $record ( @callfiledataresult = $callfiledata->fetchrow_array()
149     {
150     # $record = @callfiledataresult[5];
151     # print @campaignnameresult . "\n";
152     # print $callfiledataresult[5] . "\n";
153     # $campaignname=$campaignnameresult[0];
154     #     if (! my $campaignname)
155     #     {
156     #         $campaignname="Test";
157     #     }
158
159
160     foreach $record (@callfiledataresult)
161     {
162
163     $/="\r\n";
164
165     if (! $record )
166     {
167     next;
168     }
169     else
170     {
171     print $record . "&nbsp;&nbsp;&nbsp;";
172     }
173
174
175
176     # chomp $record;
177
178     # my @linearray=split(',',$record);
179
180
181     my $firstname=$callfiledataresult[3];
182     my $lastname=$callfiledataresult[4];
183     my $address=$callfiledataresult[6];
184     my $city=$callfiledataresult[7];
185     my $state=$callfiledataresult[8];
186     my $zip=$callfiledataresult[9];
187     my $NUMBERCALLED='1' . $callfiledataresult[5];
188     # my $debt=$callfiledataresult[16];
189     my $campaign_name="Call Files Generated From SQL";
190     # print $NUMBERCALLED;
191
192
193     # my $campaignid=00001;
194     # my $dropsequence=7;
195     my $calltime="either";
196     my $callfile="$callfiledir/$calltime-$NUMBERCALLED.call";
197     my $dropfile="/tmp/$calltime-$NUMBERCALLED.call";
198     my $ani="000-111-2345";
199     my $account="$NUMBERCALLED";
200

```

```

201         open (OUTFILE, ">$callfile");
202
203 # Channel: SIP/1$NUMBERCALLED@talkinip
204 print OUTFILE <<END_CALLFILE;
205 Channel: IAX2/rwl/$NUMBERCALLED
206 Context: customer-contact
207 Extension: s
208 Priority: 1
209 CallerID: CALL 4 YOU <$ani>
210 WaitTime: 30
211 Set: SpeechText="Call For You"
212 Set: NUMBERCALLED=$NUMBERCALLED
213 Set: DESTINATION="Custom Data Here"
214 Set: DESTINATIONDIGIT=1
215 Set: REPEATDIGIT=2
216 Set: REMOVEDIGIT=9
217 Set: CAMPAIGNID=$campaignid
218 Set: DROPSEQUENCE=$dropsequence
219 Set: CAMPAIGN_NAME=$campaign_name
220 Account: $account
221 END_CALLFILE
222 close(OUTFILE);
223
224 my $targetfile="/var/spool/asterisk/outgoing/";
225 move($callfile, $targetfile);
226
227     }
228     print "<br>";
229 }
230 ## CallFile Generation Loop
231 $callfiledata->finish();
232
233
234 }
235 # End Get Callee Data
236 print LOGFILE "End of Get Callee Data \n\n";
237 $database->disconnect();
238 # End of CallFile Generation Loop
239
240
241
242
243
244 # Connect to DB For Stopping Campaigns
245 $database = DBI->connect('DBI:mysql:asterisk', 'asterisk', '') or die "Couldn't connect to database: " . DBI->errstr;
246
247 ## Stop Each Campaign
248 foreach $campaign (@stopcampaignlist)
249 {
250     print LOGFILE "Campaign to Be Stopped: $campaign\n";
251     my $campaignid=$campaign;
252
253     # Set Stop Sequence.
254     my $stopcampaign = $database->prepare("update Campaign_Master set CallingStatus='stopped' where CampaignID=$campaignid") or die "Couldn't prepare statement: " . $database->errstr;
255     $stopcampaign->execute() or die "Couldn't execute statement: " . $stopcampaign->errstr;
256 }
257 # End Stop Campaign
258 print LOGFILE "End of Stop Campaign \n\n";
259 $database->disconnect();
260 # End of Stop Campaign Loop
261
262
263
264
265 # Connect to DB For Deleting Campaigns

```

```
266 $database = DBI->connect('DBI:mysql:asterisk', 'asterisk', '') or die "Couldn't con
nect to database: " . DBI->errstr;
267
268 ## Stop Each Campaign
269 foreach $campaign (@deletecampaignlist)
270 {
271     print LOGFILE "Campaign to Be Stopped: $campaign\n";
272     my $campaignid=$campaign;
273
274     # Set Stop Sequence.
275     my $deletecampaign = $database->prepare("drop table Campaign_{$campaignid}")
or die "Couldn't prepare statement: " . $database->errstr;
276     $deletecampaign->execute() or die "Couldn't execute statement: " . $deletec
ampaign->errstr;
277
278     $deletecampaign = $database->prepare("delete from Campaign_Master where Cam
paignID='{$campaignid}'") or die "Couldn't prepare statement: " . $database->errstr;
279     $deletecampaign->execute() or die "Couldn't execute statement: " . $deletec
ampaign->errstr;
280
281     $deletecampaign = $database->prepare("delete from Callee_Data where Callee_
Data.CampaignList = 'CampaignID-{$campaignid}'") or die "Couldn't prepare statement:
" . $database->errstr;
282     $deletecampaign->execute() or die "Couldn't execute statement: " . $deletec
ampaign->errstr;
283
284 }
285 # End Delete Campaign
286 print LOGFILE "End of Stop Campaign \n\n";
287 $database->disconnect();
288 # End of Delete Campaign Loop
289
290
291
292
293 close LOGFILE;
294
295
296 print <<END_REDIRECT;
297 <meta http-equiv="refresh" content="3;URL=/tgdialer" >
298 END_REDIRECT
299
300
301
302
303
304 # exit(99);
305
306
307
308
309
310
311
312 close LOGFILE;
313
314
315 print <<END_REDIRECT;
316 <meta http-equiv="refresh" content="3;URL=/tgdialer" >
317 END_REDIRECT
318
319
320
321
322
323 # exit(99);
```