

```

1  <?php
2
3  //  Linda Schield          revised June 2014    using new cfg
4
5  //  Linda Schield          July 2014
6  //                          Include processing of middle name
7
8  //  Linda Schield          September 2014
9  //                          1.  Passing data from peo50 via    $_SESSION
variable
10 //                          2.  Using fpdf2fileLKS for faster
processing w/ larger images
11
12 //  Linda Schield          9 November 2014 1.  Changed no photo
format to break to new page at 182 instead of 184
13 //                          2.  Pull DB snapshot when image
res is high
14
15 //  Linda Schield          March 2015  1.  Increased size of page
number
16 //                          2.  max length variable now
varies w/ location, as entries have varying indents.
17 //                          3.  Number of pre-household
data pages is now passed from peo50 as session variable.
18 //
19 //  Linda Schield          March 2018  Produce both high_res and
low_res documents in half sheet format.
20 //
21
22     function fit_text($cell_data,$max_width,$font_family,
$pdf_weight,$font_size,$pdf) {
23
24         //  passed variables: $cell_data is the text being
processed
25         //                          $max_space is the max length
allowed for the text
26         //                          $font_weight is B for bold, blank
for regular, I for italic, etc
27         //
28         //  returned variable: $d  the amount to reduce the
font by (note: must be used with the compressed version of the
font)
29
30     //      require_once (DOCUMENT_ROOT .
'wo_2/wo2_common/classlib/fpdf/fpdf.php');          //  fpdf
library for producing pdf file
31
32         $di = 0.5;          //  the incremental amount used
to DECREASE font-size each iteration
33         $pdf = new PDF('P', 'mm');
34
35         //  we only need the condensed versions of the font
36         $pdf->AddFont('ssp-c','B',
'SourceSansPro-SemiboldCondensed.php');
37         $pdf->AddFont('ssp-c','',
'SourceSansPro-RegularCondensed.php');
38

```

```

39         $di = 0.5;
40         $d = 0 - $di; // initialize to negative
value; for first time through $d + $di = 0
41
42         $pdf->SetFont($font_family . '-c', $font_weight,
$font_size ); // to start: use condensed
43         $string_width = $pdf->GetStringWidth($cell_data);
44
45         while ($string_width > $max_width) {
46             $d = $d + $di;
// first time
47             $d = 0 (This way we can return the successful font size.)
$pdf->SetFont($font_family . '-c', $font_weight,
$font_size - $d); // to start: use condensed; then decrement
if necessary
48             $string_width = $pdf->GetStringWidth($cell_data);
49         }
50
51         return $d; // return the amount to decrease
FONT_SIZE using the condensed version of the font.
52     }
53
54 // ===== END FUNCTION fit_text
=====
=====
55
56     define('DOCUMENT_ROOT',$_SERVER['DOCUMENT_ROOT'] . '/');
57     require_once (DOCUMENT_ROOT . 'wo_2/pages_cfg.php');
58
59     require_once (DOCUMENT_ROOT .
'wo_2/wo2_common/auth_login.php'); // verify
user has properly logged in
60
61     define ('ACCESS', 'p');
62 // the access code for this module
require_once (DOCUMENT_ROOT .
'wo_2/wo2_common/auth_access.php'); // verify user
has necessary access to use this module
63
64
65     require_once (DOCUMENT_ROOT .
'wo_2/wo2_common/classlib/fpdf/fpdf2fileLKS.php'); // fpdf
library for faster exec & rounded rectangles
66
67 // create DB snapshot. This can be used to compare to
current DB to see what changes have been made. NOT SURE WE
NEED THIS....LKS 3/1/2018
68 // require_once(DOCUMENT_ROOT .
'wo_2/wo2_pages/peo58_dir_snapshot.php');
69 // peo58_dir_snapshot($up1_path);
70
71     define('FPDF_FONTPATH',DOCUMENT_ROOT .
'wo_2/wo2_common/classlib/fpdf/font/'); // for custom
font SourceSansPro (ssp) and SourceSansProCondensed (ssp-c)
72
73     define("PAGE_WIDTH", 140);
74     define("PAGE_HEIGHT", 216);

```

```

75
76
77     define("LEFT_MARGIN", 15);
78     define("TOP_MARGIN", 10);
79     define("RIGHT_MARGIN", 15);
80     define("BOTTOM_MARGIN", 10);
81
82
83     define("TEXT_LEFT", LEFT_MARGIN + 3);
84
85     define("EFFECTIVE_PAGE_WIDTH", PAGE_WIDTH - LEFT_MARGIN
86 - RIGHT_MARGIN);
87
88     // x value for start of 2nd column of text (when there
89 is NO PHOTO)
90     define("COLUMN_2", LEFT_MARGIN+3 + (
91 EFFECTIVE_PAGE_WIDTH / 2));
92
93     define("IMAGE_HEIGHT", 37); // the height used for
94 all images. The appropriate width is calculated using the
95 width to height ratio.
96
97     define("FONT_SIZE", 10);
98     $font_size = FONT_SIZE; // need variable when
99 calling function fit_text
100
101     define("FONT_SIZE_PGNO", 10);
102     define("PADDING", 2); // PADDING is the distance
103 in mm between the photo and rounded rectangle
104     define("GAP1point5", 1.5); // GAP1point5 is the
105 vertical space between address and adults
106     define("GAP1", 1); // GAP1 is the vertical
107 space between household individuals
108
109     define("RR_RADIUS", 2); // RoundedRectangle Radius
110 (i.e. how round is the corner)
111
112     define("SPACES2", ' ');
113     define("SPACES3", ' ');
114     define("SPACES4", ' ');
115     define("SPACES5", ' ');
116
117     define("PATH_TO_HIGH_PHOTO", UP1_PATH .
118 '/dirImages/high-res/');
119     define("PATH_TO_LOW_PHOTO", UP1_PATH .
120 '/dirImages/low-res/');
121
122     define("FONT_FAMILY", 'ssp');
123     $font_family = FONT_FAMILY; // need variable when
124 calling function fit_text
125
126     $merged = array();
127     $merged = $_SESSION['peo50_merged'];
128     // FontFamily is SourceSansPro, an open source Adobe

```

```

font which comes in many styles. It is natively also quite
compressed, which
119      //      makes it very suitable for publishing the
directory. In addition, I have used the open source software
Font Forge to generate
120      //      regular and bold fonts which have additional
compression to help with getting long lines into a confined
space.

121
122
123  //      $pre_directory_pages = $_SESSION['pre_dir_pages'];
// The number of pre-directory pages which will print in the
physical booklet before the directory starts.
124      $wp_legend = '(work)  ';          // work phone legend
125      $res_legend = '(res)  ';          // residence address
legend
126      $mail_irdindicator = '(mail)  ';      // mailing
address legend
127      $d = 0;                          // variable returned
from function fit_text
128  //
=====
=====
129  //
130  // We need UTF-8, but it's a problem with fpdf. So, this next
command (array_walk_recursive) takes the $merged array and
converts
131  //      the entire array to ISO-8859-1, and now all is well!
132  //
133      array_walk_recursive($merged, function(&$value, $key) {
// note the & in &$value
134          if (is_string($value)) {
135              $value = iconv('UTF-8', 'ISO-8859-1', $value);
136          }
137      });
138
139      $res = array();
140      $res[1] = PATH_TO_LOW_PHOTO;
141      $res[2] = PATH_TO_HIGH_PHOTO;
142      $r = 1;                          // index to
control 1st time through for low res; 2nd time through for high
res.
143      while (isset ($res[$r])) {
144          $pdf = new FPDF2File('P', 'mm', array(PAGE_WIDTH,
PAGE_HEIGHT));
145          $pdf->SetMargins(LEFT_MARGIN, TOP_MARGIN, RIGHT_MARGIN
);
146          $pdf->Open('doc' . $r . '.pdf');
147          $pdf->AddPage();
148
149          $pdf->AddFont('ssp', '', 'SourceSansPro-Regular.php');
150
151          $pdf->AddFont('ssp', 'I', 'SourceSansPro-It.php');
152          $pdf->AddFont('ssp', 'B', 'SourceSansPro-Semibold.php'
);
153          $pdf->AddFont('ssp', 'BI',
'SourceSansPro-SemiboldIt.php');          // SSP regular

```

```

italic
153
154         $pdf->AddFont('ssp-c','',
'SourceSansPro-RegularCondensed.php'); // condensed font
as previously generated by FontForge
155         $pdf->AddFont('ssp-c','B',
'SourceSansPro-SemiboldCondensed.php'); // condensed
font as previously generated by FontForge
156
157
158         $pdf->SetLineWidth(0.2); // using
default 2mm
159         $pdf->SetDrawColor(0,0,0); // black
(0,0,0) for production
160         $pdf->SetAutoPageBreak(15); // 15mm from
page bottom
161
162         $pdf->SetY(TOP_MARGIN);
163         $top1 = TOP_MARGIN; // initialize
164
165         $i = 0;
166         while (isset ($merged[$i])) { // Begin main
loop
167
168
169         // count the number of fields in the $merged entry. If =
1, it is a cross reference entry, set $switch = 1.
170         // if > 1, it is a household entry. If no photo, set
switch = 2. If photo, set switch = 3.
171         // TO SUMMARIZE:
172         // switch_data = 1 Cross Reference entry
173         // switch_data = 2 Household entry without photo
174         // switch_data = 3 Household entry with photo
175         //
176         $format_check = count($merged[$i]);
177
178         if ($format_check == 1) { // if 1 field
in record, it is a xref record
179             $switch_data = '1';
180         }
181
182         if ($format_check > 1) { // if more
than 1 fields, check if there is a photo
183             if ( empty ($merged[$i]['h_photo'])) { //
if no photo, set switch to 2.
184                 $switch_data = '2';
185             } else {
186                 $switch_data = '3';
187             }
188         }
189
190         switch ($switch_data) {
191
192         // ===== CASE 1: CROSS REFERENCE ENTITY
=====
193         // ===== CASE 1: CROSS REFERENCE ENTITY

```

```

=====
194      // ===== CASE 1: CROSS REFERENCE ENTITY
=====
195
196      case '1':
197
198          $pdf->SetFont(FONT_FAMILY, 'B', FONT_SIZE
199      );
200      //          if ($top1 > 195) { //
201      if we can't fit this entry on this page
202          if ($top1 > 199) { //
203      if we can't fit this entry on this page
204          //Position at 1 cm from bottom
205          $pdf->SetY(-10);
206
207          $pdf->SetFont(FONT_FAMILY, 'BI',
208      FONT_SIZE_PGNO);
209          //Page number
210          $pdf->Cell(0,10,$pdf->PageNo(),0,0,
211      'C');
212          $pdf->AddPage();
213      //      add new page
214      //      reset
215      }
216
217          $pdf->SetFont(FONT_FAMILY, 'B', FONT_SIZE
218      );
219          // Take a line increase only if the
220      previous entity was not a xref entity. Otherwise, just set x
221      back to left margin.
222          $pdf->SetLeftMargin(LEFT_MARGIN);
223          if ($i > 0) {
224              if (count($merged[$i - 1] > 1)) {
225                  // if > 1, NOT a xref entity
226                  $pdf->Ln(1);
227              } else {
228                  $pdf->Ln(0);
229              }
230          }
231          // previous record was entity; don't
232      do line increase
233      }
234      }
235
236          $pdf->Cell(0,7,$merged[$i]['flexkey'],0,
237      1); // output xref listing
238
239          // next record is entity; don't do
240      line increase
241          if ((isset($merged[$i+1])) && count(
242      $merged[$i + 1] > 1)) { // if > 1, NOT a xref
243      entity
244          $pdf->Ln(1);
245          } else {
246          $pdf->Ln(0);
247          }
248      }

```

```

232         //          $text_bottom = $pdf->GetY();
233         //          $top1 = $text_bottom;
234
235         $top1 = $pdf->GetY();
236
237         break;
238
239         // ===== END CROSS REFERENCE ENTITY
===== START HOUSEHOLD ENTITY WITHOUT PHOTO
=====
240         // ===== END CROSS REFERENCE ENTITY
===== START HOUSEHOLD ENTITY WITHOUT PHOTO
=====
241         // ===== END CROSS REFERENCE ENTITY
===== START HOUSEHOLD ENTITY WITHOUT PHOTO
=====
242
243         // ===== CASE 2: HOUSEHOLD ENTRY WITHOUT
PHOTO
=====
244         // ===== CASE 2: HOUSEHOLD ENTRY WITHOUT
PHOTO
=====
245         // ===== CASE 2: HOUSEHOLD ENTRY WITHOUT
PHOTO
=====
246
247
248         case '2':
249
250             if ($top1 > 182) { //
if we can't fit this entry on this page
251                 //Position at 1 cm from bottom
252                 $pdf->SetY(-10);
253                 //Arial italic 8
254                 $pdf->SetFont(FONT_FAMILY, 'BI',
FONT_SIZE_PGNO);
255                 //Page number
256                 $pdf->Cell(0,10,$pdf->PageNo(),0,0,
'C');
257                 $pdf->AddPage();
//      add new page
258                 $pdf->SetY(TOP_MARGIN);
//      reset
259             }
260
261             $pdf->SetFont(FONT_FAMILY, 'B', FONT_SIZE
);
262             $pdf->SetLeftMargin(TEXT_LEFT);
263             $pdf->Ln(0); // establish x at left
margin
264
265             // household name, phone and address(es)
266             $top1 = $pdf->GetY();
267             $top2 = $top1 + 2 ; // 2mm border
from where rounded rectangle will be drawn
268             $pdf->SetY($top2);

```

```

269
270         if ((isset($merged[$i]['h_hpac']) && (
$merged[$i]['h_hpac'] > ''))) { // if landline data
present, we need hyphen
271             $hyphen = '-';
272         } else {
273             $hyphen = NULL;
274         }
275
276         $pdf->Cell(0,4,$merged[$i]['h_name'] .
SPACES5 . $merged[$i]['h_hpac'] . $hyphen . $merged[$i][
'h_hphone7'],0,1); // household name & phone
277
278         $pdf->SetFont(FONT_FAMILY, '', FONT_SIZE);
279
280         // address: If BOTH addr1 and addr2 are present, addr1
is street address and addr2 is mailing address. Label addr1
(res) and addr2 (mail).
281         // If only addr1 present, present without labeling.
282         // Addr2 w/out addr1 should not exist.
283
284         $address1 = $merged[$i]['h_addr1'] .
' ' . $merged[$i]['h_addr1b'];
285         $address2 = $merged[$i]['h_addr2'] .
' ' . $merged[$i]['h_addr2b'];
286
287         $pdf->Ln(0);
288
289         if ($merged[$i]['h_addr2'] > '') {
// if there is an address2, we need the (RES) and (MAIL)
designators.
290             $pdf->SetLeftMargin(TEXT_LEFT);
// don't indent extra cause we're using (RES)
291             $pdf->SetFont(FONT_FAMILY, 'B', 8);
// set font to smaller and bold
292             $pdf->Cell(9,4,'(RES) ',0,0);
// print (RES) in smaller, bold font; don't
advance to next line
293             $pdf->SetFont(FONT_FAMILY, '',
FONT_SIZE); // restore font to larger, not bold
294             } else {
295                 $pdf->SetLeftMargin(TEXT_LEFT + 4);
// indent address
296             }
297
298             $csz = $merged[$i]['h_city1'] . ', ' .
$merged[$i]['h_state1'] . SPACES2 . $merged[$i]['h_zip1'];
299             $pdf->Cell(0,4,$address1 . SPACES5 .
$csz,0,1); // address
300             if ($merged[$i]['h_addr_extra1'] > '')
{
301                 $pdf->Cell(0,4,$merged[$i][
'h_addr_extra1'],0,1); // extra address info
302             }
303
304             if ($merged[$i]['h_addr2'] > '') {
// if there is an address2, we need the (RES) and (MAIL)

```

designators.

```
305             $pdf->SetLeftMargin(TEXT_LEFT);
// don't indent extra cause we're using (RES)
306             $pdf->Ln(GAP1);           // gap 3
(between res and mail addresses)
307             $pdf->SetFont(FONT_FAMILY, 'B', 8);
// set font to smaller and bold
308             $pdf->Cell(9,4, '(MAIL) ', 0,0);
// print (RES) in smaller, bold font; don't
advance to next line
309             $pdf->SetFont(FONT_FAMILY, '',
FONT_SIZE); // restore font to larger, not bold
310             $csz = $merged[$i]['h_city2'] . ' ',
' . $merged[$i]['h_state2'] . SPACES2 . $merged[$i]['h_zip2'];
311             $pdf->Cell(0,4,$address2 . SPACES5
. $csz, 0,1); // address
312             if ($merged[$i]['h_addr_extra2'] >
'') {
313                 $pdf->Cell(0,4,$merged[$i][
'h_addr_extra2'],0,1); // extra address info
314             }
315         }
316
317             $pdf->Ln(GAP1point5); // gap 1
(between address and individuals)
318
319             // household adults
320             $j = 0;
321
322             while ($j < $merged[$i]['adult_count'])
{
323                 $pdf->SetFont(FONT_FAMILY, 'B',
FONT_SIZE); // set font to Bold
324                 if ( ! empty($merged[$i][$j][
'i_cpac'])) { // if cell phone data present, use a hyphen
325                     $hyphen = '-';
326                 } else {
327                     $hyphen = NULL;
328                 }
329
330                 // namefirst, namelast, clergy,
suffix: 19Feb2014 always use last name; append suffix if
present
331                 $middlename = ' ' . $merged[$i][$j][
'i_namemiddle'];
332                 $name = $merged[$i][$j][
'i_namefirst'] . $middlename;
333                 $name .= ' ' . $merged[$i][$j][
'i_namelast'];
334
335                 if (isset ($merged[$i][$j][
'i_clergy'])) {
336                     $name .= $merged[$i][$j][
'i_clergy'];
337                 }
338
339                 if ( ! empty($merged[$i][$j]['i_sfx']
```

```

340     ))) { // add suffix if present
           $name .= ' ' . $merged[$i][$j][
'i_sfx'];
341     }
342
343     if ($merged[$i]['adult_count'] > 1)
{ // if we have more than 1 adult (and therefore need both
columns)
344         $max_width = 50;
345     } else {
346         $max_width = 105;
347     }
348
349     $cell_data = (SPACES3 . $name .
SPACES2 . $merged[$i][$j]['i_cpac'] . $hyphen . $merged[$i][$j][
'i_cphone7']);
350
351     // calculate cell width of name +
cell phone number
352     $string_width = round($pdf->
GetStringWidth($cell_data));
353
354     if (($j + 2) % 2 == 1) { //
it's a 2nd column entry ( +2 added so that the division will
work for $j = 0 and 1)
355         $pdf->SetX(COLUMN_2);
356         $next_line = 1; //
we're filling 2nd col, so will need to advance
357     } else {
358         $pdf->SetX(TEXT_LEFT);
359         $next_line = 0;
360     }
361
362     if ($string_width > $max_width) {
363         $font_weight = 'B';
364         $d = fit_text($cell_data,
$max_width,$font_family,$font_weight,$font_size,$pdf); //
determine font size that will fit **with condensed font**
365         $pdf->SetFont(FONT_FAMILY . '-c'
,'B',FONT_SIZE - $d); // condensed, B, w/ [perhaps]
reduced font size ($d could equal zero)
366         $string_width = $pdf->
GetStringWidth($cell_data); // recalculate
367     } else {
368         $pdf->SetFont(FONT_FAMILY,'B',
FONT_SIZE); // same-old...
369     }
370
371     $pdf->Cell($string_width,4,
$cell_data,0,$next_line); // name & cell phone
372
373     $j++;
374 }
375
376     if (($j + 2) % 2 == 1) {
// remainder of 1 means even number of
adults (count starts w/ 0)

```

```

377         // $v_here = $pdf->GetY() + 4;
378 // increment 4mm
379         // $pdf->SetY($v_here);
380         $pdf->SetY($pdf->GetY() + 4);
381         // increment 4mm);
382     }
383     // adult email
384     $j = 0;
385     if ($merged[$i]['adult_email_count'] > 0
386 ) { // do these adults have email addresses?
387         while ($j < $merged[$i][
388 'adult_count']) {
389             $pdf->SetFont(FONT_FAMILY);
390             if (($j + 2) % 2 == 1) {
391                 // it's a 2nd column entry (+2 added so that the
392                 // division will work for $j = 0 and 1)
393                 $pdf->SetX(COLUMN_2 + 1);
394                 $next_line = 1;
395             } else {
396                 $pdf->SetX(TEXT_LEFT + 1);
397                 $next_line = 0;
398             }
399             $pdf->Cell(0, 4.5, SPACES4 .
400 $merged[$i][$j]['i_email'], 0, $next_line);
401             $j++;
402         }
403     }
404     if (($j + 2) % 2 == 1) {
405 // remainder of 1 means even number of adults (count starts
406 w/ 0)
407         // $v_here = $pdf->GetY() + 4;
408 // increment 4mm
409         // $pdf->SetY($v_here);
410         $pdf->SetY($pdf->GetY() + 4);
411         // increment 4mm);
412     }
413     // adult work phone
414     $j = 0;
415     if ($merged[$i]['adult_wphone_count'] >
416 0) {
417         while ($j < $merged[$i][
418 'adult_count']) {
419             $pdf->SetFont(FONT_FAMILY);
420             if (($j + 2) % 2 == 1) {
421                 // it's a 2nd column entry (+2 added so that the
422                 // division will work for $j = 0 and 1)
423                 $pdf->SetX(COLUMN_2 + 1);

```

```

419             $next_line = 1;
420         }
421     } else {
422         $pdf->SetX(TEXT_LEFT + 1);
423         $next_line = 0;
424     }
425     $hyphen = ''; // clear
426     $cell_data = ''; // clear
427
428
429     if ($merged[$i][$j]['i_wpac'] >
'' ) {
430         $hyphen = '-';
431         $cell_data = $merged[$i][$j]
['i_wpac'] . $hyphen . $merged[$i][$j]['i_wphone7'] . ' ' .
$merged[$i][$j]['i_wpext'];
432
433         $pdf->SetFontSize(FONT_SIZE-
2); // work phone legend in smaller font size
434         $string_width = $pdf->
GetStringWidth(SPACES4 . $wp_legend);
435         $pdf->Cell($string_width,4.5
, SPACES4 . $wp_legend,0,0);
436
437         $pdf->SetFontSize(FONT_SIZE
); // restore normal font size
438         $string_width = $pdf->
GetStringWidth($cell_data);
439         $pdf->Cell($string_width,4.5
,$cell_data,0,$next_line); // work phone
440     } else {
441         $pdf->Cell($string_width,4.5
,$cell_data,0,$next_line); // we don't have data, but we do
need the next line
442     }
443     $j++;
444 }
445 }
446
447 $pdf->SetFont(FONT_FAMILY);
448
449     if (($j + 2) % 2 == 1) {
// if there were an odd number of adults (count starts w/ 0)
450         $pdf->SetY($pdf->GetY() + 4);
// increment 4mm);
451     }
452
453     // collect all the children (while)
454     $pdf->Ln(GAP1point5);
455     $j = 0;
456     $cell_data = NULL;
457
458     while ( isset($merged[$i][$j][
'i_relationship'])) { // gather all children, comma
separated
459         if (($merged[$i][$j][
'i_relationship']) == 'C') { // if a child entry?

```

```

460 $j]['i_namemiddle']; $middlename = ' ' . $merged[$i][
461 $cell_data .= $merged[$i][$j][
'i_namefirst'] . $middlename . ', ' ; // now contains
trailing comma
462 }
463 $j++;
464 }
465
466 if (isset ($cell_data)) {
467     $strlen = strlen($cell_data);
468     $cell_data = substr_replace(
$cell_data, ' ', $strlen-2); // clear the final comma
469
470     $pdf->Ln(1);
471     $pdf->SetFont(FONT_FAMILY, 'BI');
472
473     $pdf->SetX(TEXT_LEFT + 2);
474
475     $pdf->Cell(0,4,$cell_data,0,1);
476     $pdf->SetFont(FONT_FAMILY, 'B');
477 }
478
479
480 $text_bottom = $pdf->GetY();
481
482 $text_height = $text_bottom - $top2;
483
484 // Rounded Rectangle: entity_height = text height (there
is no photo)
485 $entity_height = $text_height;
486
487 $rr_height = $entity_height + 4;
// the greater of text or image, plus 4mm
488 $pdf->RoundedRect(LEFT_MARGIN, $top1,
EFFECTIVE_PAGE_WIDTH, $rr_height, RR_RADIUS, 'D');
489
490 $pdf->SetY($top1 + $rr_height);
491 $top1 = $pdf->GetY(); //
start (top) of next rounded rectangle
492
493 break;
494
495 // ===== END HOUSEHOLD WITHOUT PHOTO
ENTITY ===== START HOUSEHOLD ENTITY WITH PHOTO
=====
496 // ===== END HOUSEHOLD WITHOUT PHOTO
ENTITY ===== START HOUSEHOLD ENTITY WITH PHOTO
=====
497 // ===== END HOUSEHOLD WITHOUT PHOTO
ENTITY ===== START HOUSEHOLD ENTITY WITH PHOTO
=====
498
499 case '3':
500 // if ($top1 > 164) { //
if we can't fit this entry on this page
501 if ($top1 > 168) { //

```

```

if we can't fit this entry on this page
502         //Position at 1 cm from bottom
503         $pdf->SetY(-10);
504         //Arial italic 8
505         $pdf->SetFont(FONT_FAMILY, 'BI',
FONT_SIZE_PGNO);
506         //Page number
507         $pdf->Cell(0,10,$pdf->PageNo(),0,0,
'C');
508         $pdf->AddPage();
//      add new page
509         $pdf->SetY(TOP_MARGIN);
//      reset
510     }
511
512     $pdf->SetFont(FONT_FAMILY, 'B', FONT_SIZE
);
513     $pdf->SetLeftMargin(TEXT_LEFT);
// for this kind of text, the left margin is indented a little
514     $pdf->Ln(0); // establish x at left
margin
515
516
517         $top1 = $pdf->GetY();
518         $top2 = $top1 + PADDING;
519         $pdf->SetY($top2);
520
521     // next section calculates beginning (left edge) of
photo. We don't want to run text into the photo! :)
522     $photo_filename = $res[$r] . $merged[$i
][ 'h_photo' ];
523     $image_size_array = getimagesize(
$photo_filename); // get image dimensions
524     $w_to_h_ratio = $image_size_array[0] /
$image_size_array[1]; // calculate ratio of width to
height
525     $image_width = $w_to_h_ratio *
IMAGE_HEIGHT; // calculate image width
needed for std/constant image height
526
527     $x_text_limit = PAGE_WIDTH -
RIGHT_MARGIN - $image_width - 2*PADDING; // text can't
extend beyond $x_text_limit
528     //echo '<br /><br />' . $x_text_limit;
529
530     $max_width = $x_text_limit - (TEXT_LEFT
+2); // calculate max length
allowed (mm)
531     //echo '<br />' . $max_width;
532     if ((isset($merged[$i][ 'h_hpac' ]) && (
$merged[$i][ 'h_hpac' ]> ''))) {
533         $hyphen = '-';
534     } else {
535         $hyphen = NULL;
536     }
537     // the for fit_text routine...
538     $font_weight = 'B';

```

```

539             $cell_data = $merged[$i]['h_name'] .
SPACES5 . $merged[$i]['h_hpac'] . $hyphen . $merged[$i][
'h_hphone7'];           // collect data to be printed
540             $string_width = $pdf->GetStringWidth(
$cell_data);
541             //echo '<br />' . $cell_data . ' 1=' . $image_width . '
2=' . $string_width . ' 3=' . $max_width;
542
543             if ($string_width > $max_width) {
544                 $d = fit_text($cell_data,$max_width,
$font_family,$font_weight,$font_size,$pdf); // determine font
size that will fit **with condensed font**
545                 $pdf->SetFont(FONT_FAMILY . '-c',
$font_weight,FONT_SIZE - $d);           // condensed, B, w/
[perhaps] reduced font size ($d could equal zero)
546                 $string_width = $pdf->GetStringWidth
($cell_data);           // recalculate
547             } else {
548                 $pdf->SetFont(FONT_FAMILY,
$font_weight,FONT_SIZE);           // same-old... (no
compression or font-size reduction)
549             }
550             // end for fit_text code
551             $pdf->Cell(0,4,$cell_data,0,1);           //
place household name and landline
552
553             $pdf->SetFont(FONT_FAMILY,'',FONT_SIZE);
// reset to default
554
555             // address: If BOTH addr1 and addr2 are present, addr1
is street address and addr2 is mailing address. Label addr1
(res) and addr2 (mail).
556             // If only addr1 present, present without labeling.
557             // Addr2 w/out addr1 should not exist.
558
559             $max_width = $x_text_limit - (TEXT_LEFT
+2) - 3;           // address has additional indent
560
561             $address1 = $merged[$i]['h_addr1'] .
' ' . $merged[$i]['h_addr1b'];
562
563             if ($merged[$i]['h_addr2'] > '') {
// if there is an address2, we need the (RES) and (MAIL)
designators.
564                 $pdf->Ln(GAP1);
// because there are two addresses, leave a little gap here
565                 $pdf->SetLeftMargin(TEXT_LEFT);
// don't indent extra cause we're using (RES)
566                 $pdf->SetFont(FONT_FAMILY,'B',8);
// set font to smaller and bold
567                 $pdf->Cell(9,4,'(RES)',0,0);
// print (RES) in smaller, bold font; don't advance to next
line
568                 $pdf->SetFont(FONT_FAMILY,'',
FONT_SIZE);           // restore font to larger, not bold
569             } else {
570                 $pdf->SetLeftMargin(TEXT_LEFT + 4);

```

```

        // no res/mail designator, so indent more
571     }
572
573     $cell_data = $address1;
574     $string_width = $pdf->GetStringWidth(
$cell_data);
575     $font_weight = '';
576     if ($string_width > $max_width) {
577         $d = fit_text($cell_data,$max_width,
$font_family,$font_weight,$font_size,$pdf); // determine font
size that will fit **with condensed font**
578         $pdf->SetFont(FONT_FAMILY . '-c',
$font_weight, FONT_SIZE - $d); // condensed, B, w/
[perhaps] reduced font size ($d could equal zero)
579         $string_width = $pdf->GetStringWidth
($cell_data); // recalculate
580     } else {
581         $pdf->SetFont(FONT_FAMILY,
$font_weight, FONT_SIZE); // same-old... (no
compression or font-size reduction)
582     }
583
584     $pdf->Cell(0,4,$cell_data,0,1); //
place address1
585
586     $csz1 = $merged[$i]['h_city1'] . ' '
. $merged[$i]['h_state1'] . SPACES2 . $merged[$i]['h_zip1'];
587     if ($csz1 > '') {
588         $cell_data = $csz1;
589         $string_width = $pdf->GetStringWidth
($cell_data);
590         $font_weight = '';
591         if ($string_width > $max_width) {
592             $d = fit_text($cell_data,
$max_width,$font_family,$font_weight,$font_size,$pdf); //
determine font size that will fit **with condensed font**
593             $pdf->SetFont(FONT_FAMILY . '-c'
,$font_weight, FONT_SIZE - $d); // condensed, B, w/
[perhaps] reduced font size ($d could equal zero)
594             $string_width = $pdf->
GetStringWidth($cell_data); // recalculate
595         } else {
596             $pdf->SetFont(FONT_FAMILY,
$font_weight, FONT_SIZE); // same-old... (no
compression or font-size reduction)
597         }
598
599         $pdf->Cell(0,4,$cell_data,0,1);
600     }
601
602     if ($merged[$i]['h_addr_extra1'] > '')
{
603         $extra1 = $merged[$i][
'h_addr_extra1'];
604         $cell_data = $extra1;
605         $string_width = $pdf->GetStringWidth
($cell_data);

```

```

606         $font_weight = '';
607         if ($string_width > $max_width) {
608             $d = fit_text($cell_data,
$max_width,$font_family,$font_weight,$font_size,$pdf); //
determine font size that will fit **with condensed font**
609             $pdf->SetFont(FONT_FAMILY . '-c'
,$font_weight,FONT_SIZE - $d); // condensed, B, w/
[perhaps] reduced font size ($d could equal zero)
610             $string_width = $pdf->
GetStringWidth($cell_data); // recalculate
611         } else {
612             $pdf->SetFont(FONT_FAMILY,
$font_weight,FONT_SIZE); // same-old... (no
compression or font-size reduction)
613         }
614
615         $pdf->Cell(0,4,$merged[$i][
'h_addr_extra1'],0,1); // extra address info
616     }
617
618
619         if ($merged[$i]['h_addr2'] > '') {
// if there is an address2, we need the (MAIL) designator for
address2.
620             $pdf->Ln(GAP1);
// leave a little gap between the two addresses
621             $pdf->SetLeftMargin(TEXT_LEFT);
// don't indent extra cause we're using (RES)
622             $pdf->SetFont(FONT_FAMILY,'B',8);
// set font to smaller and bold
623             $pdf->Cell(9,4,'(MAIL)',0,0);
// print (RES) in smaller, bold font; don't advance to next
line
624             $pdf->SetFont(FONT_FAMILY,'',
FONT_SIZE); // restore font to larger, not bold
625
626             $address2 = $merged[$i]['h_addr2']
. ' ' . $merged[$i]['h_addr2b'];
627             $cell_data = $address2;
628             $string_width = $pdf->GetStringWidth
($cell_data);
629             $font_weight = '';
630             if ($string_width > $max_width) {
631                 $d = fit_text($cell_data,
$max_width,$font_family,$font_weight,$font_size,$pdf); //
determine font size that will fit **with condensed font**
632                 $pdf->SetFont(FONT_FAMILY . '-c'
,$font_weight,FONT_SIZE - $d); // condensed, B, w/
[perhaps] reduced font size ($d could equal zero)
633                 $string_width = $pdf->
GetStringWidth($cell_data); // recalculate
634             } else {
635                 $pdf->SetFont(FONT_FAMILY,
$font_weight,FONT_SIZE); // same-old... (no
compression or font-size reduction)
636             }
637

```

```

638             $pdf->Cell(0,4,$cell_data,0,1);
639 // place address2
640             $csz1 = $merged[$i]['h_city2'] .
641 ' ' . $merged[$i]['h_state2'] . SPACES2 . $merged[$i]['
642 'h_zip2'];
641             if ($csz1 > '') {
642                 $cell_data = $csz1;
643                 $string_width = $pdf->
GetStringWidth($cell_data);
644                 $font_weight = '';
645                 if ($string_width > $max_width)
646 {
647                     $d = fit_text($cell_data,
648 $max_width,$font_family,$font_weight,$font_size,$pdf); //
649 determine font size that will fit **with condensed font**
650 $pdf->SetFont(FONT_FAMILY .
651 '-c',$font_weight,FONT_SIZE - $d); // condensed, B, w/
652 [perhaps] reduced font size ($d could equal zero)
653 $string_width = $pdf->
654 GetStringWidth($cell_data); // recalculate
655 } else {
656     $pdf->SetFont(FONT_FAMILY,
657 $font_weight,FONT_SIZE); // same-old... (no
658 compression or font-size reduction)
659 }
660             $pdf->Cell(0,4,$cell_data,0,1);
661         }
662         if ($merged[$i]['h_addr_extra2'] >
663 '') {
664             $extra1 = $merged[$i][
665 'h_addr_extra2'];
666             $cell_data = $extra1;
667             $string_width = $pdf->
668 GetStringWidth($cell_data);
669             $font_weight = '';
670             if ($string_width > $max_width)
671 {
672                 $d = fit_text($cell_data,
673 $max_width,$font_family,$font_weight,$font_size,$pdf); //
674 determine font size that will fit **with condensed font**
675 $pdf->SetFont(FONT_FAMILY .
676 '-c',$font_weight,FONT_SIZE - $d); // condensed, B, w/
677 [perhaps] reduced font size ($d could equal zero)
678 $string_width = $pdf->
679 GetStringWidth($cell_data); // recalculate
680 } else {
681     $pdf->SetFont(FONT_FAMILY,
682 $font_weight,FONT_SIZE); // same-old... (no
683 compression or font-size reduction)
684 }
685             $pdf->Cell(0,4,$merged[$i][
686 'h_addr_extra2'],0,1); // extra address info
687         }

```

```

671     }
672     $pdf->Ln(GAP1point5); //
leave a bigger gap between the address info and the individual
names
673
674     // household adults, name and cell phone
675
676     $j = 0;
677     $pdf->SetX(LEFT_MARGIN + 2);
678     $pdf->Ln(0);
679
680
681     while ($j < $merged[$i]['adult_count'])
{
682
683         if ( ! empty($merged[$i][$j][
'i_cpac'])) {
684             $hyphen = '-';
685         } else {
686             $hyphen = NULL;
687         }
688
689         // 19Feb2014 always use last name;
append clergy and suffix if present
690         $middlename = ' ' . $merged[$i][$j][
'i_namemiddle'];
691         $name = $merged[$i][$j][
'i_namefirst'] . $middlename;
692         $name .= ' ' . $merged[$i][$j][
'i_namelast'];
693         if (isset ($merged[$i][$j][
'i_clergy'])) {
694             $name .= $merged[$i][$j][
'i_clergy'];
695         }
696
697         if ( ! empty($merged[$i][$j]['i_sfx'
])) {
698             $name .= ' ' . $merged[$i][$j][
'i_sfx'];
699         }
700
701         $cell_data = SPACES3 . $name .
SPACES2 . $merged[$i][$j]['i_cpac'] . $hyphen . $merged[$i][$j][
'i_cphone7'];
702
703         $max_width = $x_text_limit - (
TEXT_LEFT +2) - 6; // individual data has additional indent
//
704         $name_cell_max_width = $max_width -
6; // it's less than $max_width due to indents. not
sure what the number should be ... :(
705         $string_width = $pdf->GetStringWidth
($cell_data);
706         $font_weight = 'B';
707         if ($string_width > $max_width) {
            $d = fit_text($cell_data,
$max_width,$font_family,$font_weight,$font_size,$pdf); //
determine font size that will fit **with condensed font**

```

```

708             $pdf->SetFont(FONT_FAMILY, '-c',
, $font_weight, FONT_SIZE - $d); // condensed, B, w/
[perhaps] reduced font size ($d could equal zero)
709             $string_width = $pdf->
GetStringWidth($cell_data); // recalculate
710         } else {
711             $pdf->SetFont(FONT_FAMILY,
$font_weight, FONT_SIZE); // same-old... (no
compression or font-size reduction)
712         }
713
714             $pdf->Cell(0,4,$cell_data,0,1);
715
716             $start_x = TEXT_LEFT;
717
718             $pdf->SetFont(FONT_FAMILY, '',
FONT_SIZE); // turn off Bold
719
720             // email address
721             if ($merged[$i][$j]['i_email'] > '')
{
722                 $cell_data = SPACES4 . $merged[$i
][$j]['i_email'];
723                 $string_width = $pdf->
GetStringWidth($cell_data);
724                 if ($string_width > $max_width)
{
725                     // max width allows enough space so won't run into
photo
726                     $font_weight = '';
$d = fit_text($cell_data,
$max_width, $font_family, $font_weight, $font_size, $pdf); //
determine font size that will fit **with condensed font**
727                     $pdf->SetFont(FONT_FAMILY,
'-c', '', FONT_SIZE - $d); // condensed, not Bold, w/
[perhaps] reduced font size ($d could equal zero)
728                     $string_width = $pdf->
GetStringWidth($cell_data); // recalculate
729                 } else {
730                     $pdf->SetFont(FONT_FAMILY, ''
, FONT_SIZE); // same-old...
731                 }
732
733                 $pdf->Cell($string_width, 4.5,
$cell_data, 0, 1); // print email address
734             }
735
736             // work phone
737
738             $hyphen = '';
739             $next_line = 1;
740             $pdf->SetFont(FONT_FAMILY, '',
FONT_SIZE);
741
742
743             if ($merged[$i][$j]['i_wpac'] > '')
{

```

```

744             $hyphen = '-';
745             $cell_data = $merged[$i][$j][
'i_wpac'] . $hyphen . $merged[$i][$j]['i_wphone7'] . ' ' .
$merged[$i][$j]['i_wpext'];
746
747             $pdf->SetFontSize(FONT_SIZE-2);
           // work phone legend in smaller font size
748             $string_width = $pdf->
GetStringWidth(SPACES4 . $wp_legend);
749             $pdf->Cell($string_width,4.5,
SPACES5 . $wp_legend,0,0);
750
751             $pdf->SetFontSize(FONT_SIZE);
           // restore normal font size
752             $string_width = $pdf->
GetStringWidth($cell_data);
753             $pdf->Cell($string_width,4.5,
$cell_data,0,$next_line); // work phone
754         }
755         $j++;
756
757         if ($j < $merged[$i]['adult_count'])
758         {           // if there are more adults to print
759             $pdf->Ln(GAP1);
760         }
761     }
762
763     // collect all the children (while)
764     $pdf->Ln(GAP1point5);
765     $j = 0;
766     $cell_data = NULL;
767
768     while ( isset($merged[$i][$j][
'i_relationship'])) {
769         if (($merged[$i][$j][
'i_relationship']) == 'C') {           // if a child entry?
770             $middlename = ' ' . $merged[$i][
$j]['i_namemiddle'];
771             $cell_data .= $merged[$i][$j][
'i_namefirst'] . ', ';
772         }
773         $j++;
774     }
775
776     if (isset ($cell_data)) {
777         $strlen = strlen($cell_data);
778         $cell_data = substr_replace(
$cell_data, ' ', $strlen-2);           // clear the final comma
779
780         $pdf->Ln(1);
781         $pdf->SetX(TEXT_LEFT + 2);
782         $pdf->SetFont(FONT_FAMILY, 'BI');
783         $pdf->Cell(0,4,$cell_data,0,1);
784
785         // position on NEXT LINE
786         $pdf->SetFont(FONT_FAMILY, 'B');

```

```

786
787
788         $text_bottom = $pdf->GetY();
789
790
791         $pdf->Image($photo_filename,
786 $x_text_limit+2,$stop2,0,IMAGE_HEIGHT);
792
793         $photo_bottom = $pdf->GetY();
794
795         $text_height = $text_bottom - $stop2;
796
797         // entity_height = the greater of
786 text_height, image_height
798         $pdf->SetLeftMargin(LEFT_MARGIN);
799         // restore
800         $entity_height = ($text_height >
786 IMAGE_HEIGHT) ? $text_height : IMAGE_HEIGHT;
801
802         $rr_height = $entity_height + 2*PADDING;
803         // the greater of text or image, plus 2 times the
804 PADDING value
805
806         $pdf->RoundedRect(LEFT_MARGIN, $stop1,
786 EFFECTIVE_PAGE_WIDTH, $rr_height, RR_RADIUS, 'D');
807
808         $pdf->SetY($stop1 + $rr_height);
809         $stop1 = $pdf->GetY();
810
811
812         break;
813     } // end switch
814
815     $i++;
816
817     } // end $merged array while
818
819 // ===== END HOUSEHOLD
820 WITH PHOTO ENTITY =====
821 // ===== END HOUSEHOLD
822 WITH PHOTO ENTITY =====
823 // ===== END HOUSEHOLD
824 WITH PHOTO ENTITY =====
825
826 // final page number
827 //Position at 1 cm from bottom
828 $pdf->SetY(-10);
829 //Arial italic 8
830 $pdf->SetFont(FONT_FAMILY,'BI',FONT_SIZE_PGNO);
831 //Page number
832 $pdf->Cell(0,10,$pdf->PageNo(),0,0,'C');
833 $pdf->Output();
834 $r ++; // increment res array
835
836 index
837     } // end low-res/high-res while
838
839     $url = BASE_URL . 'wo_2/wo2_pages/peo51b_upload.php';
840     header("Location: $url");
841

```

```
832         exit();
833
834     ?>
```