

```

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  $_SESSION variable
10 //                      2.  Using fpdf2fileLKS for faster processing w/ larger images
11
12 //  Linda Schield      9 November 2014 1.  Changed no photo format to break to new page at
13 //                      182 instead of 184
14 //                      2.  Pull DB snapshot when image res is high
15 //  Linda Schield      March 2015  1.  Increased size of page number
16 //                      2.  max length variable now varies w/ location, as entries
17 //                      have varying indents.
18 //                      3.  Number of pre-household data pages is now passed from
19 //                      peo50 as session variable.
20 //  Linda Schield      March 2018  Produce both high_res and low_res documents in half
21 //                      sheet format.
22 //
23 //
24 //  function fit_text($cell_data,$max_width,$font_family,$font_weight,$font_size,$pdf) {
25 //      //  passed variables: $cell_data is the text being processed
26 //      //                      $max_space is the max length allowed for the text
27 //      //                      $font_weight is B for bold, blank for regular, I for italic,
28 //      etc
29 //      //
30 //      //  returned variable:  $d  the amount to reduce the font by (note: must be used
31 //      with the compressed version of the font)
32 //
33 //      require_once (DOCUMENT_ROOT . 'wo_2/wo2_common/classlib/fpdf/fpdf.php');
34 //      fpdf library for producing pdf file
35
36 //
37 //
38 //
39 //
40 //
41 //
42 //
43 //
44 //
45 //
46 //
47 //
48 //
49 //
50 //
51 //
52 //
53 //
54 //
55 //
56 //
57 //
58 //
59 //
60 //
61 //
62 //
63 //
64 //
65 //
66 //
67 //
68 //
69 //
70 //
71 //
72 //
73 //
74 //
75 //
76 //
77 //
78 //
79 //
80 //
81 //
82 //
83 //
84 //
85 //
86 //
87 //
88 //
89 //
90 //
91 //
92 //
93 //
94 //
95 //
96 //
97 //
98 //
99 //
100 //
101 //
102 //
103 //
104 //
105 //
106 //
107 //
108 //
109 //
110 //
111 //
112 //
113 //
114 //
115 //
116 //
117 //
118 //
119 //
120 //
121 //
122 //
123 //
124 //
125 //
126 //
127 //
128 //
129 //
130 //
131 //
132 //
133 //
134 //
135 //
136 //
137 //
138 //
139 //
140 //
141 //
142 //
143 //
144 //
145 //
146 //
147 //
148 //
149 //
150 //
151 //
152 //
153 //
154 //
155 //
156 //
157 //
158 //
159 //
160 //
161 //
162 //
163 //
164 //
165 //
166 //
167 //
168 //
169 //
170 //
171 //
172 //
173 //
174 //
175 //
176 //
177 //
178 //
179 //
180 //
181 //
182 //
183 //
184 //
185 //
186 //
187 //
188 //
189 //
190 //
191 //
192 //
193 //
194 //
195 //
196 //
197 //
198 //
199 //
200 //
201 //
202 //
203 //
204 //
205 //
206 //
207 //
208 //
209 //
210 //
211 //
212 //
213 //
214 //
215 //
216 //
217 //
218 //
219 //
220 //
221 //
222 //
223 //
224 //
225 //
226 //
227 //
228 //
229 //
230 //
231 //
232 //
233 //
234 //
235 //
236 //
237 //
238 //
239 //
240 //
241 //
242 //
243 //
244 //
245 //
246 //
247 //
248 //
249 //
250 //
251 //
252 //
253 //
254 //
255 //
256 //
257 //
258 //
259 //
260 //
261 //
262 //
263 //
264 //
265 //
266 //
267 //
268 //
269 //
270 //
271 //
272 //
273 //
274 //
275 //
276 //
277 //
278 //
279 //
280 //
281 //
282 //
283 //
284 //
285 //
286 //
287 //
288 //
289 //
290 //
291 //
292 //
293 //
294 //
295 //
296 //
297 //
298 //
299 //
300 //
301 //
302 //
303 //
304 //
305 //
306 //
307 //
308 //
309 //
310 //
311 //
312 //
313 //
314 //
315 //
316 //
317 //
318 //
319 //
320 //
321 //
322 //
323 //
324 //
325 //
326 //
327 //
328 //
329 //
330 //
331 //
332 //
333 //
334 //
335 //
336 //
337 //
338 //
339 //
340 //
341 //
342 //
343 //
344 //
345 //
346 //
347 //
348 //
349 //
350 //
351 //
352 //
353 //
354 //
355 //
356 //
357 //
358 //
359 //
360 //
361 //
362 //
363 //
364 //
365 //
366 //
367 //
368 //
369 //
370 //
371 //
372 //
373 //
374 //
375 //
376 //
377 //
378 //
379 //
380 //
381 //
382 //
383 //
384 //
385 //
386 //
387 //
388 //
389 //
390 //
391 //
392 //
393 //
394 //
395 //
396 //
397 //
398 //
399 //
400 //
401 //
402 //
403 //
404 //
405 //
406 //
407 //
408 //
409 //
410 //
411 //
412 //
413 //
414 //
415 //
416 //
417 //
418 //
419 //
420 //
421 //
422 //
423 //
424 //
425 //
426 //
427 //
428 //
429 //
430 //
431 //
432 //
433 //
434 //
435 //
436 //
437 //
438 //
439 //
440 //
441 //
442 //
443 //
444 //
445 //
446 //
447 //
448 //
449 //
450 //
451 //
452 //
453 //
454 //
455 //
456 //
457 //
458 //
459 //
460 //
461 //
462 //
463 //
464 //
465 //
466 //
467 //
468 //
469 //
470 //
471 //
472 //
473 //
474 //
475 //
476 //
477 //
478 //
479 //
480 //
481 //
482 //
483 //
484 //
485 //
486 //
487 //
488 //
489 //
490 //
491 //
492 //
493 //
494 //
495 //
496 //
497 //
498 //
499 //
500 //
501 //
502 //
503 //
504 //
505 //
506 //
507 //
508 //
509 //
510 //
511 //
512 //
513 //
514 //
515 //
516 //
517 //
518 //
519 //
520 //
521 //
522 //
523 //
524 //
525 //
526 //
527 //
528 //
529 //
530 //
531 //
532 //
533 //
534 //
535 //
536 //
537 //
538 //
539 //
540 //
541 //
542 //
543 //
544 //
545 //
546 //
547 //
548 //
549 //
550 //
551 //
552 //
553 //
554 //
555 //
556 //
557 //
558 //
559 //
560 //
561 //
562 //
563 //
564 //
565 //
566 //
567 //
568 //
569 //
570 //
571 //
572 //
573 //
574 //
575 //
576 //
577 //
578 //
579 //
580 //
581 //
582 //
583 //
584 //
585 //
586 //
587 //
588 //
589 //
590 //
591 //
592 //
593 //
594 //
595 //
596 //
597 //
598 //
599 //
600 //
601 //
602 //
603 //
604 //
605 //
606 //
607 //
608 //
609 //
610 //
611 //
612 //
613 //
614 //
615 //
616 //
617 //
618 //
619 //
620 //
621 //
622 //
623 //
624 //
625 //
626 //
627 //
628 //
629 //
630 //
631 //
632 //
633 //
634 //
635 //
636 //
637 //
638 //
639 //
640 //
641 //
642 //
643 //
644 //
645 //
646 //
647 //
648 //
649 //
650 //
651 //
652 //
653 //
654 //
655 //
656 //
657 //
658 //
659 //
660 //
661 //
662 //
663 //
664 //
665 //
666 //
667 //
668 //
669 //
670 //
671 //
672 //
673 //
674 //
675 //
676 //
677 //
678 //
679 //
680 //
681 //
682 //
683 //
684 //
685 //
686 //
687 //
688 //
689 //
690 //
691 //
692 //
693 //
694 //
695 //
696 //
697 //
698 //
699 //
700 //
701 //
702 //
703 //
704 //
705 //
706 //
707 //
708 //
709 //
710 //
711 //
712 //
713 //
714 //
715 //
716 //
717 //
718 //
719 //
720 //
721 //
722 //
723 //
724 //
725 //
726 //
727 //
728 //
729 //
730 //
731 //
732 //
733 //
734 //
735 //
736 //
737 //
738 //
739 //
740 //
741 //
742 //
743 //
744 //
745 //
746 //
747 //
748 //
749 //
750 //
751 //
752 //
753 //
754 //
755 //
756 //
757 //
758 //
759 //
760 //
761 //
762 //
763 //
764 //
765 //
766 //
767 //
768 //
769 //
770 //
771 //
772 //
773 //
774 //
775 //
776 //
777 //
778 //
779 //
780 //
781 //
782 //
783 //
784 //
785 //
786 //
787 //
788 //
789 //
790 //
791 //
792 //
793 //
794 //
795 //
796 //
797 //
798 //
799 //
800 //
801 //
802 //
803 //
804 //
805 //
806 //
807 //
808 //
809 //
810 //
811 //
812 //
813 //
814 //
815 //
816 //
817 //
818 //
819 //
820 //
821 //
822 //
823 //
824 //
825 //
826 //
827 //
828 //
829 //
830 //
831 //
832 //
833 //
834 //
835 //
836 //
837 //
838 //
839 //
840 //
841 //
842 //
843 //
844 //
845 //
846 //
847 //
848 //
849 //
850 //
851 //
852 //
853 //
854 //
855 //
856 //
857 //
858 //
859 //
860 //
861 //
862 //
863 //
864 //
865 //
866 //
867 //
868 //
869 //
870 //
871 //
872 //
873 //
874 //
875 //
876 //
877 //
878 //
879 //
880 //
881 //
882 //
883 //
884 //
885 //
886 //
887 //
888 //
889 //
890 //
891 //
892 //
893 //
894 //
895 //
896 //
897 //
898 //
899 //
900 //
901 //
902 //
903 //
904 //
905 //
906 //
907 //
908 //
909 //
910 //
911 //
912 //
913 //
914 //
915 //
916 //
917 //
918 //
919 //
920 //
921 //
922 //
923 //
924 //
925 //
926 //
927 //
928 //
929 //
930 //
931 //
932 //
933 //
934 //
935 //
936 //
937 //
938 //
939 //
940 //
941 //
942 //
943 //
944 //
945 //
946 //
947 //
948 //
949 //
950 //
951 //
952 //
953 //
954 //
955 //
956 //
957 //
958 //
959 //
960 //
961 //
962 //
963 //
964 //
965 //
966 //
967 //
968 //
969 //
970 //
971 //
972 //
973 //
974 //
975 //
976 //
977 //
978 //
979 //
980 //
981 //
982 //
983 //
984 //
985 //
986 //
987 //
988 //
989 //
990 //
991 //
992 //
993 //
994 //
995 //
996 //
997 //
998 //
999 //
1000 //

```

```

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 $d
= 0 (This way we can return the successful font size.)
47     $pdf->SetFont($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');          // the access code for
this module

```

```

62     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 - RIGHT_MARGIN);
86
87 //  x value for start of 2nd column of text (when there is NO PHOTO)
88     define("COLUMN_2", LEFT_MARGIN+3 + (EFFECTIVE_PAGE_WIDTH / 2));
89
90
91
92     define("IMAGE_HEIGHT", 37);           //  the height used for all images.  The
appropriate width is calculated using the width to height ratio.
93
94     define("FONT_SIZE", 10);
95     $font_size = FONT_SIZE;           //  need variable when calling function fit_text

```

```

96
97     define("FONT_SIZE_PGNO", 10);
98     define("PADDING", 2);          // PADDING is the distance in mm between the photo and
rounded rectangle
99     define("GAP1point5", 1.5);      // GAP1point5 is the vertical space between
address and adults
100    define("GAP1", 1);              // GAP1 is the vertical space between household
individuals
101
102    define("RR_RADIUS", 2);          // RoundedRectangle Radius (i.e. how round is the corner)
103
104    define("SPACES2", ' ');
105    define("SPACES3", ' ');
106    define("SPACES4", ' ');
107    define("SPACES5", ' ');
108
109
110    define("PATH_TO_HIGH_PHOTO", UP1_PATH . '/dirImages/high-res/');
111    define("PATH_TO_LOW_PHOTO", UP1_PATH . '/dirImages/low-res/');
112
113    define("FONT_FAMILY", 'ssp');
114    $font_family = FONT_FAMILY;      // need variable when calling function fit_text
115
116    $merged = array();
117    $merged = $_SESSION['peo50_merged'];
118    // 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    $imail_irdndicator = '(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     $pdf->AddFont('ssp', 'I', 'SourceSansPro-It.php');
151     $pdf->AddFont('ssp', 'B', 'SourceSansPro-Semibold.php');
152     $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
170             // entry, set $switch = 1.
171             // if > 1, it is a household entry. If no photo, set switch = 2. If photo, set
172             // switch = 3.
173             // TO SUMMARIZE:
174             //     switch_data = 1      Cross Reference entry
175             //     switch_data = 2      Household entry without photo
176             //     switch_data = 3      Household entry with photo
177             //
178             $format_check = count($merged[$i]);
179
180             if ($format_check == 1) {          // if 1 field in record, it is a xref
181 record
182                 $switch_data = '1';
183             }
184
185             if ($format_check > 1) {           // if more than 1 fields, check if there
186 is a photo
187                 if ( empty ($merged[$i]['h_photo'])) { // if no photo, set switch to
188 2.
189                     $switch_data = '2';
190                 } else {
191                     $switch_data = '3';           // otherwise, (there is a
192 photo), set switch to 3.
193                 }
194             }
195
196             switch ($switch_data) {

```

```

192      // ===== CASE 1: CROSS REFERENCE ENTITY
=====
193      // ===== CASE 1: CROSS REFERENCE ENTITY
=====
194      // ===== CASE 1: CROSS REFERENCE ENTITY
=====
195
196          case '1':
197
198              $pdf->SetFont(FONT_FAMILY, 'B', FONT_SIZE);
199
200              // if we can't fit this entry
on this page if ($top1 > 195) {
201              // if we can't fit this entry
on this page if ($top1 > 199) {
202                  //Position at 1 cm from bottom
203                  $pdf->SetY(-10);
204
205                  $pdf->SetFont(FONT_FAMILY, 'BI', FONT_SIZE_PGNO);
206                  //Page number
207                  $pdf->Cell(0,10,$pdf->PageNo(),0,0,'C');
208                  $pdf->AddPage(); // add new page
209                  $pdf->SetY(TOP_MARGIN); // reset
210              }
211
212              $pdf->SetFont(FONT_FAMILY, 'B', FONT_SIZE);
213              // Take a line increase only if the previous entity was not a xref
entity. Otherwise, just set x back to left margin.
214              $pdf->SetLeftMargin(LEFT_MARGIN);
215              if ($i > 0) {
216                  if (count($merged[$i - 1] > 1)) { // if > 1, NOT a xref
entity
217                      $pdf->Ln(1);
218                  } else {
219                      $pdf->Ln(0); // previous record was
entity; don't do line increase
220                  }
221              }
222

```

```

223         $pdf->Cell(0,7,$merged[$i]['flexkey'],0,1);           // output xref
listing
224
225         // next record is entity; don't do line increase
226         if ((isset($merged[$i+1])) && count($merged[$i + 1] > 1)) { //
if > 1, NOT a xref entity
227             $pdf->Ln(1);
228         } else {
229             $pdf->Ln(0);
230         }
231
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     $stop1 = $pdf->GetY();
267     $stop2 = $stop1 + 2 ; // 2mm border from where rounded rectangle
will be drawn
268     $pdf->SetY($stop2);
269
270     if ((isset($merged[$i]['h_hpac']) && ($merged[$i]['h_hpac'] > '')))
271     { // if landline data present, we need hyphen
272         $hyphen = '-';
273     } else {
274         $hyphen = NULL;
275     }
276
277     $pdf->Cell(0,4,$merged[$i]['h_name'] . SPACES5 . $merged[$i][
'h_hpac'] . $hyphen . $merged[$i]['h_hphone7'],0,1); // household name & phone
278
279     $pdf->SetFont(FONT_FAMILY, '', FONT_SIZE);
280
281     // address: If BOTH addr1 and addr2 are present, addr1 is street address and addr2
is mailing address. Label addr1 (res) and addr2 (mail).
282     // If only addr1 present, present without labeling.
283     // Addr2 w/out addr1 should not exist.
284
285     $address1 = $merged[$i]['h_addr1'] . ' ' . $merged[$i]['h_addr1b'
];
286
287     $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'])) { // add
suffix if present
340             $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; // increment 4mm
378         // $pdf->SetY($v_here);
379
380         $pdf->SetY($pdf->GetY() + 4); // increment 4mm);
381     }
382     // adult email
383     $j = 0;
384
385     if ($merged[$i]['adult_email_count'] > 0) { // do these adults
have email addresses?
386         while ($j < $merged[$i]['adult_count']) {
387             $pdf->SetFont(FONT_FAMILY);
388
389             if (($j + 2) % 2 == 1) { // it's a 2nd column
entry (+2 added so that the division will work for $j = 0 and 1)
390                 $pdf->SetX(COLUMN_2 + 1);
391                 $next_line = 1;
392
393             } else {
394                 $pdf->SetX(TEXT_LEFT + 1);
395                 $next_line = 0;
396             }
397             $pdf->Cell(0,4.5,SPACES4 . $merged[$i][$j]['i_email'],0,
$next_line);
398             $j++;
399         }
400     }
401
402
403     if (($j + 2) % 2 == 1) { // remainder of 1 means
even number of adults (count starts w/ 0)
404     // $v_here = $pdf->GetY() + 4; // increment 4mm
405     // $pdf->SetY($v_here);
406
407     $pdf->SetY($pdf->GetY() + 4); // increment 4mm);
408 }

```

```

409
410 // adult work phone
411 $j = 0;
412
413 if ($merged[$i]['adult_wphone_count'] > 0) {
414     while ($j < $merged[$i]['adult_count']) {
415         $pdf->SetFont(FONT_FAMILY);
416
417         if (($j + 2) % 2 == 1) { // it's a 2nd column
entry (+2 added so that the division will work for $j = 0 and 1)
418             $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         $cell_data = ''; // clear
427
428         if ($merged[$i][$j]['i_wpac'] > '') {
429             $hyphen = '-';
430             $cell_data = $merged[$i][$j]['i_wpac'] . $hyphen .
431 $merged[$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             $middlename = ' ' . $merged[$i][$j]['i_namemiddle'];
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         $pdf->Cell(0,4,$cell_data,0,1);
475         $pdf->SetFont(FONT_FAMILY,'B');
476     }
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     $stop1 = $pdf->GetY();
518     $stop2 = $stop1 + PADDING;
519     $pdf->SetY($stop2);
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);    // place address2
639
640         $csz1 = $merged[$i]['h_city2'] . ' ' . $merged[$i][
'h_state2'] . SPACES2 . $merged[$i]['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                 $d = fit_text($cell_data,$max_width,$font_family,
$font_weight,$font_size,$pdf); // determine font size that will fit **with condensed font**
647                 $pdf->SetFont(FONT_FAMILY . '-c',$font_weight,FONT_SIZE

```

```

- $d);          // condensed, B, w/ [perhaps] reduced font size ($d could equal zero)
648             $string_width = $pdf->GetStringWidth($cell_data);
             // recalculate
649             } else {
650             $pdf->SetFont(FONT_FAMILY,$font_weight,FONT_SIZE);
             // same-old... (no compression or font-size reduction)
651             }
652
653             $pdf->Cell(0,4,$cell_data,0,1);
654         }
655
656         if ($merged[$i]['h_addr_extra2'] > '') {
657             $extra1 = $merged[$i]['h_addr_extra2'];
658             $cell_data = $extra1;
659             $string_width = $pdf->GetStringWidth($cell_data);
660             $font_weight = '';
661             if ($string_width > $max_width) {
662                 $d = fit_text($cell_data,$max_width,$font_family,
$font_weight,$font_size,$pdf); // determine font size that will fit **with condensed font**
663                 $pdf->SetFont(FONT_FAMILY . '-c',$font_weight,FONT_SIZE
- $d);          // condensed, B, w/ [perhaps] reduced font size ($d could equal zero)
664                 $string_width = $pdf->GetStringWidth($cell_data);
             // recalculate
665             } else {
666             $pdf->SetFont(FONT_FAMILY,$font_weight,FONT_SIZE);
             // same-old... (no compression or font-size reduction)
667             }
668
669             $pdf->Cell(0,4,$merged[$i]['h_addr_extra2'],0,1); //
extra address info
670         }
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             $max_width = $x_text_limit - (TEXT_LEFT +2) - 6; //
individual data has additional indent
703             // $name_cell_max_width = $max_width - 6; // it's less than
$max_width due to indents. not sure what the number should be ... :(
704             $string_width = $pdf->GetStringWidth($cell_data);
705             $font_weight = 'B';
706             if ($string_width > $max_width) {
707                 $d = fit_text($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);
712             // same-old... (no compression or font-size reduction)
713         }
714         $pdf->Cell(0,4,$cell_data,0,1);
715         $start_x = TEXT_LEFT;
716
717         $pdf->SetFont(FONT_FAMILY, '', FONT_SIZE); // turn off Bold
718
719         // email address
720         if ($merged[$i][$j]['i_email'] > '') {
721             $cell_data = SPACES4 . $merged[$i][$j]['i_email'];
722             $string_width = $pdf->GetStringWidth($cell_data);
723
724             if ($string_width > $max_width) { // max width
725                 allows enough space so won't run into photo
726                 $font_weight = '';
727                 $d = fit_text($cell_data,$max_width,$font_family,
728                 $font_weight,$font_size,$pdf); // determine font size that will fit **with condensed font**
729                 $pdf->SetFont(FONT_FAMILY . '-c', '', FONT_SIZE - $d);
730                 // condensed, not Bold, w/ [perhaps] reduced font size ($d could equal zero)
731                 $string_width = $pdf->GetStringWidth($cell_data);
732                 // recalculate
733             } else {
734                 $pdf->SetFont(FONT_FAMILY, '', FONT_SIZE);
735                 // same-old...
736             }
737             $pdf->Cell($string_width,4.5,$cell_data,0,1); //
738             print email address
739         }
740
741         // work phone
742
743         $hyphen = '';
744         $next_line = 1;
745         $pdf->SetFont(FONT_FAMILY, '', FONT_SIZE);

```

```

742
743         if ($merged[$i][$j]['i_wpac'] > '') {
744             $hyphen = '-';
745             $cell_data = $merged[$i][$j]['i_wpac'] . $hyphen . $merged[
746 $i][$j]['i_wphone7'] . ' ' . $merged[$i][$j]['i_wpext'];
747
748             $pdf->SetFontSize(FONT_SIZE-2);           // work
749             phone legend in smaller font size
750             $string_width = $pdf->GetStringWidth(SPACES4 . $wp_legend);
751             $pdf->Cell($string_width,4.5,SPACES5 . $wp_legend,0,0);
752
753             $pdf->SetFontSize(FONT_SIZE);           // restore normal
754             font size
755             $string_width = $pdf->GetStringWidth($cell_data);
756             $pdf->Cell($string_width,4.5,$cell_data,0,$next_line); //
757             work phone
758         }
759         $j++;
760         if ($j < $merged[$i]['adult_count']) {      // if there are
761             more adults to print
762             $pdf->Ln(GAP1);
763         }
764         // collect all the children (while)
765         $pdf->Ln(GAP1point5);
766         $j = 0;
767         $cell_data = NULL;
768         while ( isset($merged[$i][$j]['i_relationship'])) {
769             if (($merged[$i][$j]['i_relationship']) == 'C') {      // if
770                 a child entry?
771                 $middlename = ' ' . $merged[$i][$j]['i_namemiddle'];
772                 $cell_data .= $merged[$i][$j]['i_namefirst'] . ', ';
773             }
774             $j++;

```

```

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);           // position on NEXT LINE
784             $pdf->SetFont(FONT_FAMILY, 'B');
785         }
786
787
788         $text_bottom = $pdf->GetY();
789
790
791         $pdf->Image($photo_filename, $x_text_limit+2,$stop2,0,IMAGE_HEIGHT);
792
793         $photo_bottom = $pdf->GetY();
794
795         $text_height = $text_bottom - $stop2;
796
797         // entity_height = the greater of text_height, image_height
798         $pdf->SetLeftMargin(LEFT_MARGIN);           // restore
799         $entity_height = ($text_height > IMAGE_HEIGHT) ? $text_height :
IMAGE_HEIGHT;
800
801         $rr_height = $entity_height + 2*PADDING;           // the greater of
text or image, plus 2 times the PADDING value
802         $pdf->RoundedRect(LEFT_MARGIN, $stop1, EFFECTIVE_PAGE_WIDTH,
$rr_height, RR_RADIUS, 'D');
803
804         $pdf->SetY($stop1 + $rr_height);
805         $stop1 = $pdf->GetY();
806
807
808         break;
809     } // end switch

```

```

810
811         $i++;
812
813     }    // end $merged array while
814
815 // ===== END HOUSEHOLD WITH PHOTO ENTITY
=====
816 // ===== END HOUSEHOLD WITH PHOTO ENTITY
=====
817 // ===== END HOUSEHOLD WITH PHOTO ENTITY
=====
818
819     // final page number
820     //Position at 1 cm from bottom
821     $pdf->SetY(-10);
822     //Arial italic 8
823     $pdf->SetFont(FONT_FAMILY, 'BI', FONT_SIZE_PGNO);
824     //Page number
825     $pdf->Cell(0,10,$pdf->PageNo(),0,0,'C');
826     $pdf->Output();
827     $r ++;    // increment res array index
828 }    // end low-res/high-res while
829
830 $url = BASE_URL . 'wo_2/wo2_pages/peo51b_upload.php';
831 header("Location: $url");
832 exit();
833
834 ?>

```