

The **sdapsclassic** package*

Benjamin Berg
benjamin@sipsolutions.net

February 15, 2020

1 Documentation

Please refer to <https://sdaps.org/class-doc> for documentation.

2 Implementation

This package uses the L^AT_EX3 language internally, so we need to enable it.

```
1 % We need at least 2011-08-23 for \keys_set_known:nnN
2 \RequirePackage{expl3}[2011/08/23]
3 \% \RequirePackage{xparse}
4 \ExplSyntaxOn
5
6 \RequirePackage{sdapsbase}
7 \RequirePackage{sdapslayout}
8
9 % We use verbatim to grab the content of the questionnaire environment into a
10 % temporary file. This way we can input it multiple times for repeating the
11 % the content.
12
13 \RequirePackage{verbatim}
14
15
```

Now a lot of LaTeX 2e stuff that is basically just a copy of the old class. Would be nicer to redo this with xparse, but what the heck.

```
16
17 \RequirePackage{scrbbase}
18
19
20 %-----
```

```
21 % option processing
22 %-----
```

*This document corresponds to **sdapsclassic** v0.1, dated 2015/08/02.

```

23
24 % Whether the main SDAPS program will be used
25 \DeclareOption{disable_recognition}{\bool_gset_false:N\g_sdaps_recognition_bool}
26
27 % Whether sdaps should print the questionnaire id
28 \DeclareOption{no_print_questionnaire_id}{\bool_gset_false:N\g_sdaps_print_questionnaire_id_}
29 \DeclareOption{print_questionnaire_id}{\bool_gset_true:N\g_sdaps_print_questionnaire_id_bool}
30
31 % pass unknown options to scrartcl
32 \DeclareOption*{\PassOptionsToClass{\CurrentOption}{scrartcl}}
33
34 % We need this if right now as it is set by sdaps.opt.
35 \newif\if@sdaps@draft\@sdaps@drafttrue
36 \DeclareOption{final}{\@sdaps@draftfalse}
37
38 \tl_gset:Nn\g_sdaps_style_tl{qr}%
39
40
41 \DefineFamily{SDAPS}
42 \DefineFamilyMember{SDAPS}
43 \DefineFamilyKey{SDAPS}{sdaps_style}[qr]%
44 {%
45   \KOMA@set@ncmdkey{sdaps_style}{@tempa}{%
46     {code128}{0},%
47     {custom}{1},%
48     {qr}{2}%
49   }{#1}%
50   \ifcase \@tempa\relax
51     \tl_gset:Nn\g_sdaps_style_tl{code128}%
52   \or
53     \tl_gset:Nn\g_sdaps_style_tl{custom}%
54   \or
55     \tl_gset:Nn\g_sdaps_style_tl{qr}%
56   \fi
57 }
58
59 \DefineFamilyKey{SDAPS}{twoside_barcode}[both]%
60 {%
61   \KOMA@set@ncmdkey{twoside_barcode}{@tempa}{%
62     {both}{0},%
63     {front}{1},%
64     {back}{2}%
65   }{#1}%
66   \ifcase \@tempa\relax
67     \tl_gset:Nn\g_sdaps_twoside_barcode_tl{both}%
68   \or
69     \tl_gset:Nn\g_sdaps_twoside_barcode_tl{front}%
70   \or
71     \tl_gset:Nn\g_sdaps_twoside_barcode_tl{back}%
72   \fi

```

```

73 }
74
75 \DefineFamilyKey{SDAPS}{checkmode}[checkcorrect]%
76 {%
77   \KOMA@set@ncmdkey{checkmode}{@tempa}{%
78     {checkcorrect}{0},%
79     {check}{1},%
80     {fill}{2}%
81   }{#1}%
82   \ifcase \atempa\relax
83     \tl_gset:Nn\g_sdaps_checkmode_tl{checkcorrect}%
84   \or
85     \tl_gset:Nn\g_sdaps_checkmode_tl{check}%
86   \or
87     \tl_gset:Nn\g_sdaps_checkmode_tl{fill}%
88   \fi
89 }
90
91
92 \DefineFamilyKey{SDAPS}{globalid}[]
93 {
94   \tl_gset:Nn \g_sdaps_global_id_tl { #1 }
95 }
96
97 \DefineFamilyKey{SDAPS}{globalidlabel}[]
98 {
99   \tl_gset:Nn \g_sdaps_global_id_label_tl { #1 }
100 }
101
102 % Set default options for scrartcl. Is this done correctly?
103 \PassOptionsToClass{headings=small}{scrartcl}
104 \PassOptionsToClass{twoside}{scrartcl}
105 % Well, I don't think that this worked in the past ...
106 %\PassOptionsToClass{10pt}{scrartcl}
107
108 \seq_new:N \g_sdaps_questionnaire_ids_seq
109 \seq_gset_from_clist:Nn \g_sdaps_questionnaire_ids_seq {{}}
110
111 % process given options
112 \FamilyProcessOptions{SDAPS}\relax
113
114 %-----%
115 % load base-class
116 %-----%
117 \LoadClass{scrartcl}
118
119
120 %-----%
121 % required packages
122 %-----%

```

```

123 % geometry package
124 \RequirePackage{geometry}
125 % Corner mark positions plus 2mm, except for the header
126 \geometry{top=12mm}
127 \geometry{bottom=14mm}
128 \geometry{hmargin=12mm}
129 \geometry{includeheadfoot}
130 \geometry{headheight=\baselineskip}
131 \geometry{headsep=\baselineskip}
132 % QR code size plus 2mm
133 \geometry{footskip=12mm}
134
135 % ifthen package
136 \RequirePackage{ifthen}
137
138 % fontenc package
139 \RequirePackage[T1]{fontenc}
140
141 % color
142 \RequirePackage{color}
143
144 % Symbols (boxes)
145 \RequirePackage{amssymb}
146
147 % For writing out the page number of boxes
148 \RequirePackage{refcount}
149
150 % Defines the LastPage label
151 \RequirePackage{lastpage}
152
153 % Environment creation that gets the body as a macro
154 \RequirePackage{environ}
155
156 % headers and footers
157 \usepackage[scrlayer-scrpage]
158 \clearpairofpagestyles
159 \chead*{\@title}
160 \cfoot*{\sdapspagemark}
161
162 % hyperrefs
163 \RequirePackage{url}
164 \RequirePackage{hyperref}
165 \hypersetup{%
166   breaklinks,%
167   baseurl      = http://,%
168   pdfborder    = 0 0 0,%
169   pdfpagemode = UseNone,%
170   pdfcreator   = '\LaTeX{} with 'sdaps' class,%
171   pdfproducer  = '\LaTeX{}%
172 }

```

```

173
174 % graphics
175 \RequirePackage{graphicx}
176
177 % Section formatting
178 \RequirePackage{sectsty}
179
180 % table of fixed width
181 \RequirePackage{tabularx}
182
183 % Babel (needed for the translator)
184 \RequirePackage{babel}
185
186 % Translation
187 \RequirePackage{translator}
188 \usedictionary{translator-sdaps-dictionary}
189
190
191 % Execute sdaps.opt file to allow SDAPS to override any options
192 \InputIfFileExists{sdaps.opt}{}{%
193
194
195 %-----%
196 %           class definition
197 %-----%
198 % minimal base settings
199
200 \setlength{\lineskip}{1\p@}
201 \setlength{\normallineskip}{1\p@}
202 \renewcommand{\baselinestretch}{}
203 \setlength{\parindent}{0pt}
204 \setlength{\parskip}{1.0ex \oplus 1.5ex \ominus -0.25ex}
205 %\setlength{\parskip}{0pt}
206 \setlength{\columnsep}{10\p@}
207 \setlength{\columnseprule}{0\p@}
208 % Redefine the spacings for \subsection ie. \_sdaps_classic_question.
209 \renewcommand{\section}{\@startsection {section}{1}{\z@}%
210         {-\parskip}%
211         {\parskip}%
212         {\normalfont\Large\bfseries\SS@sectfont}}}
213 \renewcommand{\subsection{%
214     \@startsection{subsection}{2}{\z@}%
215     {0.5\parskip}%
216     {0.25\parskip}\% These are deleted again for questions
217     {}%
218 }%
219 \pagestyle{scrheadings}%
220 \pagenumbering{arabic}%
221 \raggedbottom%
222 \%flushbottom

```

```

223 \onecolumn
224 %
225 %
226 % Fonts for the "classic" class
227 %
228
229 %common font settings
230 \newkomafont{questionfont}{}
231 \newkomafont{choicefont}{}
232 %singlemark
233 \newkomafont{singlemarkquestionfont}{\usekomafont{questionfont}}
234 \newkomafont{singlemarkchoicefont}{\usekomafont{choicefont}}
235 %markgroup
236 \newkomafont{marklinequestionfont}{\usekomafont{questionfont}}
237 \newkomafont{marklinechoicefont}{\usekomafont{choicefont}}
238 %choicequestion
239 \newkomafont{choiceitemfont}{\usekomafont{choicefont}}
240 %choicegroup
241 \newkomafont{choicegroupchoicefont}{\usekomafont{questionfont}}
242 \newkomafont{choicegrouplinefont}{\usekomafont{choicefont}}
243
244

```

2.1 A bunch of old commands

```

245
246
247 \providetcommand{\addinfo}[2]{
248   \sdaps_info_write:x{\unexpanded{#1}=\unexpanded{#2}}
249 }
250
251 \newcommand\qid{\tl_use:N \g_sdaps_questionnaire_id_tl}
252
253 \def\sdaps_classic_question#1{%
254   \tl_if_empty:nTF{#1}{%
255     \refstepcounter{subsection}%
256     \par%
257   }{%
258     % #1 is nonempty
259     \subsection{\usekomafont{questionfont}\strut\ignorespaces#1}%
260     % This is extra spacing after the subsection is removed. By doing this we
261     % get exactly one \parskip
262     \nobreak%
263     \vspace{-0.25\parskip}%
264     \nobreak%
265   }%
266 }
267
268 \newenvironment{info}{%
269   \group_begin:

```

```

270      {
271          \par
272          % Prevent top skip glue from being inserted at the start of the page
273          \topskip=0pt
274          \noindent\hrule height 1pt%
275          \nobreak
276          \vspace{-\parskip}
277          \nobreak
278          \vspace{0.5ex}
279          \nobreak
280      }
281 }%
282 \par
283 \nobreak
284 \vspace{-\parskip}
285 \nobreak
286 \vspace{0.5ex}
287 \nobreak
288 \noindent\hrule height 1pt
289 \group_end:
290 }
291
292
293 \definecolor{sectionbgcolor}{gray}{0.8}
294 \definecolor{sectionfgcolor}{gray}{0.0}
295
296 \newcommand{\sectbox}[1]{%
297     \noindent\protect\colorbox{sectionbgcolor}{%
298         \tempdima=\hsize
299         \advance\tempdima by-2\fboxsep
300         \protect\parbox{\tempdima}{%
301             \smallskip
302             \raggedright % extra commands here
303             \color{sectionfgcolor}\usekomafont{section}{#1} \smallskip
304         }%
305     }%
306 }
307
308 \sectionfont{\sectbox}
309
310 \setkomafont{disposition}{\normalfont}
311 \addtokomafont{section}{\bfseries\sffamily}
312
313
314
315
316
317 % Not sure why we even had this ...
318 \def\smallskip{\vspace\smallskipamount}
319 \def\medskip{\vspace\medskipamount}

```

```

320 \def\bigskip{\vspace\bigskipamount}
321 \newskip\smallskipamount \smallskipamount=3pt plus 1pt minus 1pt
322 \newskip\medskipamount \medskipamount =6pt plus 2pt minus 2pt
323 \newskip\bigsipamount \bigsipamount =12pt plus 4pt minus 4pt
324
325
326
327 \cs_generate_variant:Nn \sdaps_textbox_hstretch:nnnn { nVVnn }
328 \cs_generate_variant:Nn \sdaps_textbox_hstretch:nnnn { VVVnn }
329 \cs_generate_variant:Nn \int_set:Nn { NV }
330
331

332
333
334 \bool_new:N \g_sdaps_classic_have_section
335 \bool_gset_false:N \g_sdaps_classic_have_section
336 \cs_new_eq:NN \sdapsclassic_origsection \section
337 \renewcommand{\section}[1]{
338   \bool_if:NT \g_sdaps_classic_have_section {
339     \sdaps_qobject_end:n { section }
340   }
341
342   \sdapsclassic_origsection{#1}
343   \bool_gset_true:N \g_sdaps_classic_have_section
344   \sdaps_qobject_begin:n { section }{ Head }{ #1 }
345 }
346
347 \cs_new_protected_nopar:Nn \sdaps_classic_ensure_section: {
348   \bool_if:NF \g_sdaps_classic_have_section {
349     % This is a bad hack to make the numbering start with zero
350     \int_gdecr:N \g_sdaps_object_id_int
351   \bool_gset_true:N \g_sdaps_classic_have_section
352   \sdaps_qobject_begin:nnn { section }{ Head }{ }
353     \sdaps_context_hook_end:n { \bool_gset_false:N \g_sdaps_classic_have_section }
354   }
355 }
356
357
358 \tl_new:N \l_sdaps_classic_textbox_var_tl
359 \tl_new:N \l_sdaps_classic_textbox_text_tl
360 \keys_define:nn { sdapsclassic / textbox }
361 {
362   var          .tl_set:N    = \l_sdaps_classic_textbox_var_tl,
363   text         .tl_set:N    = \l_sdaps_classic_textbox_text_tl,
364 }
365
366 % With star non-expanding, without start expanding

```

```

367 \providecommand{\textbox}{\@ifstar
368   {\bool_set_false:N \l_tmpa_bool \_sdaps_classic_textbox }
369   {\bool_set_true:N \l_tmpa_bool \_sdaps_classic_textbox }
370 }
371
372 \providecommand{\_sdaps_classic_textbox}[3] []
373 {
374   \keys_set:nn { sdapsclassic / textbox } { #1 }
375
376   \sdaps_classic_ensure_section:
377   \_sdaps_classic_question{#3}
378
379   \tl_if_empty:NTF \l_sdaps_classic_textbox_text_tl {
380     \sdaps_qobject_begin:nnn { textbox } { Text } { #3 }
381   } {
382     \sdaps_qobject_begin:nnV { textbox } { Text } \l_sdaps_classic_textbox_text_tl
383   }
384
385   \bool_if:NTF \l_tmpa_bool {
386     \sdaps_textbox_vhstretch:Vnn \l_sdaps_classic_textbox_var_tl { #2 } { 1 }
387   }{
388     \sdaps_textbox_vhstretch:Vnn \l_sdaps_classic_textbox_var_tl { #2 } { 0 }
389   }
390
391   \sdaps_qobject_end:n { textbox }
392 }
393
394 \newcounter{markcheckboxcount}
395 \setcounter{markcheckboxcount}{5}
396
397
398 \tl_new:N \l_sdaps_singlemark_var_tl
399 \int_new:N \l_sdaps_singlemark_count_int
400
401 \keys_define:nn { sdaps / singlemark }
402 {
403   var          .tl_set:N    = \l_sdaps_singlemark_var_tl,
404
405   % 0 is equivalent to using the markcheckboxcount counter
406   count        .int_set:N  = \l_sdaps_singlemark_count_int,
407   count        .initial:n = 0,
408 }
409
410 \cs_new_protected_nopar:Nn \_sdaps_classic_dummy_checkbox_prepare:NN
411 {
412   \group_begin:
413   \sdaps_context_begin_local:
414   \sdaps_context_disable_writing:
415   \bool_if:NT #1 {
416     \sdaps_context_append:nn { singlechoice } { draw_check=true }

```

```

417     \sdaps_context_append:nn { multichoice } { draw_check=true }
418 }
419 \bool_if:NT #2 {
420     \sdaps_context_append:nn { singlechoice } { fill=black }
421     \sdaps_context_append:nn { multichoice } { fill=black }
422 }
423 }
424
425 \cs_new_protected_nopar:Nn \_sdaps_classic_dummy_checkbox_single:
426 {
427     \sdaps_checkbox_set_type:n { singlechoice }
428     \sdaps_checkbox:nn { } { }
429 \group_end:
430 \ignorespaces
431 }
432
433 \cs_new_protected_nopar:Nn \_sdaps_classic_dummy_checkbox_multi:
434 {
435     \sdaps_checkbox_set_type:n { multichoice }
436     \sdaps_checkbox:nn { } { }
437 \group_end:
438 \ignorespaces
439 }
440
441 \providecommand{\checkbox}{%
442     \_sdaps_classic_dummy_checkbox_prepare:NN \c_false_bool \c_false_bool
443     \@ifstar
444         \_sdaps_classic_dummy_checkbox_single:
445         \_sdaps_classic_dummy_checkbox_multi:
446 }
447 \providecommand{\checkbox}{%
448     \_sdaps_classic_dummy_checkbox_prepare:NN \c_true_bool \c_false_bool
449     \@ifstar
450         \_sdaps_classic_dummy_checkbox_single:
451         \_sdaps_classic_dummy_checkbox_multi:
452 }
453 \providecommand{\filledbox}{%
454     \_sdaps_classic_dummy_checkbox_prepare:NN \c_false_bool \c_true_bool
455     \@ifstar
456         \_sdaps_classic_dummy_checkbox_single:
457         \_sdaps_classic_dummy_checkbox_multi:
458 }
459 \providecommand{\correctedbox}{%
460     \_sdaps_classic_dummy_checkbox_prepare:NN \c_true_bool \c_true_bool
461     \@ifstar
462         \_sdaps_classic_dummy_checkbox_single:
463         \_sdaps_classic_dummy_checkbox_multi:
464 }
465
466 \providecommand*{\singlemark}[4] []{%

```

```

467 \sdaps_classic_ensure_section:
468 \group_begin:
469 \keys_set:nn { sdaps / singlemark } { #1 }
470
471 \int_compare:nNnT { \l_sdaps_singlemark_count_int } = { 0 } {
472   \int_set:Nn \l_sdaps_singlemark_count_int { \themarkcheckboxcount }
473 }
474
475 \sdaps_classic_question{#2}%
476
477 \sdaps_qobject_begin:nnn { singlemark } { range } { #2 }
478
479 \sdaps_checkbox_set_type:n { singlechoice }
480
481 \tl_if_empty:NF \l_sdaps_singlemark_var_tl {
482   \sdaps_qobject_append_var:V \l_sdaps_singlemark_var_tl
483 }
484
485 \sdaps_range:nnn{lower}{0}{#3}
486 \sdaps_range:nnn{upper}{\int_use:N\l_sdaps_singlemark_count_int-1}{#4}
487
488 \begin{tabularx}{\linewidth}{X*\int_use:N\l_sdaps_singlemark_count_int{c}X}
489   {\hfill\usekomafont{singlemarkchoicefont}\strut\ignorespaces#3} &
490   \int_step_inline:nnnn { 1 } { 1 } { \int_use:N\l_sdaps_singlemark_count_int } { \sdaps
491     \usekomafont{singlemarkchoicefont}#4\hfill}\%
492 \end{tabularx}%
493
494 \sdaps_qobject_end:n { singlemark }
495 \group_end:
496
497 }
498
499
500
501 \providecommand*\singlemarkother[5][]{%
502   \sdaps_classic_ensure_section:
503
504   \group_begin:
505
506   \keys_set:nn { sdaps / singlemark } { #1 }
507
508   \int_compare:nNnT { \l_sdaps_singlemark_count_int } = { 0 } {
509     \int_set:Nn \l_sdaps_singlemark_count_int { \themarkcheckboxcount }
510   }
511
512   \sdaps_classic_question{#2}%
513
514   \sdaps_qobject_begin:nnn { singlemark } { range } { #2 }
515
516   \sdaps_checkbox_set_type:n { singlechoice }

```

```

517
518 \tl_if_empty:NF \l_sdaps_singlemark_var_tl {
519   \sdaps_qobject_append_var:V \l_sdaps_singlemark_var_tl
520 }
521
522 \sdaps_range:nnn{lower}{0}{#3}
523 \sdaps_range:nnn{upper}{\int_use:N\l_sdaps_singlemark_count_int-1}{#4}
524 \sdaps_answer:n{#5}
525
526 \begin{tabularx}{\linewidth}{X*\int_use:N\l_sdaps_singlemark_count_int}{c}XX}
527   {\hfill\usekomafont{\singlemarkchoicefont}\strut\ignorespaces#3} &
528   \int_step_inline:nnnn { 1 } { 1 } { \int_use:N\l_sdaps_singlemark_count_int } { \sdaps
529   {\usekomafont{\singlemarkchoicefont}}#4} &
530   {\usekomafont{\singlemarkchoicefont}\sdaps_checkbox:nn { _0 } { 0 } {} ~ #5\hfill}\\\%
531 \end{tabularx}%
532
533 \sdaps_qobject_end:n { singlemark }
534 \group_end:
535 }
536
537 \dim_new:N \l_sdaps_classic_choiceitem_pad_dim
538 \int_new:N \l_sdaps_classic_choiceitem_cols_int
539 \int_new:N \l_sdaps_classic_choiceitem_col_int
540 \dim_new:N \l_sdaps_classic_choicequestion_prevdepth_dim
541 \coffin_new:N \l_sdaps_classic_choicequestion_coffin
542
543 \msg_new:nnn { sdapsclassic } { choicequestion_wrong_mode } { Mode~should~always~be~vertical
544
545 \msg_new:nnn { sdapsclassic } { choicequestion_no_text } { Textboxes~cannot~be~used~in~singl
546 It~is~currently~not~supported~to~use~text~answers~in~single~choice~questions.~Until~support~
547 you~can~get~similar~results~by~temporarily~changing~the~checkbox~style. }
548
549 \msg_new:nnn { sdapsclassic } { choicequestion_unknown_key } { The~key~'#1'~is~unknown.\If
550
551 \tl_new:N \l_sdaps_choicequestion_type_tl
552 \tl_new:N \l_sdaps_choicequestion_var_tl
553 \tl_new:N \l_sdaps_choicequestion_text_tl
554 \int_new:N \l_sdaps_choicequestion_cols_int
555
556 \keys_define:nn { sdaps / choicequestion }
557 {
558   var .tl_set:N = \l_sdaps_choicequestion_var_tl,
559   text .tl_set:N = \l_sdaps_choicequestion_text_tl,
560   colsep .dim_set:N = \l_sdaps_choicequestion_colsep_dim,
561   colsep .initial:n = 6pt,
562   rowsep .dim_set:N = \l_sdaps_choicequestion_rowsep_dim,
563   rowsep .initial:n = 0pt,
564   cols .int_set:N = \l_sdaps_choicequestion_cols_int,
565   cols .initial:n = 3,
566

```

```

567 type      .choices:nn = { multichoice, singlechoice } { \tl_set:Nx \l_sdaps_choicquestio
568 type      .initial:n = { multichoice },
569
570 singlechoice .meta:n = { type=singlechoice },
571 multichoice .meta:n = { type=multichoice },
572
573 unknown .code:n = \msg_error:nnV { sdapsclassic } { choicequestion_unknown_key } \l_keys_k
574 }
575
576
577 \tl_new:N \l_sdaps_choicequestion_choice_var_tl
578 \tl_new:N \l_sdaps_choicequestion_choice_val_tl
579 \tl_new:N \l_sdaps_choicequestion_choice_text_tl
580
581 \keys_define:nn { sdaps / choicequestion / choice }
582 {
583   var      .tl_set:N = \l_sdaps_choicequestion_choice_var_tl,
584   val      .tl_set:N = \l_sdaps_choicequestion_choice_val_tl,
585   text     .tl_set:N = \l_sdaps_choicequestion_choice_text_tl,
586 }
587
588
589 \newenvironment{choicequestion}[2][]{%
590   \group_begin:
591
592   \sdaps_classic_ensure_section:
593
594   \tl_clear:N \l_sdaps_choicequestion_var_tl
595   \tl_clear:N \l_sdaps_choicequestion_text_tl
596
597   \keys_set:nn { sdaps / choicequestion } { #1 }
598
599   \sdaps_classic_question{#2}%
600
601   % Setup the context
602   \tl_if_eq:VnTF \l_sdaps_choicquestio_type_tl { multichoice } {
603     \tl_set:Nn \l_tmpa_tl { Choice }
604   } {
605     \tl_set:Nn \l_tmpa_tl { Option }
606   }
607   \tl_if_empty:NTF \l_sdaps_choicequestion_text_tl {
608     \sdaps_qobject_begin:nVn { choicequestion } \l_tmpa_tl { #2 }
609   } {
610     \sdaps_qobject_begin:nVV { choicequestion } \l_tmpa_tl \l_sdaps_choicequestion_text_tl
611   }
612
613   \sdaps_checkbox_set_type:V \l_sdaps_choicquestio_type_tl
614
615   \tl_if_empty:NF \l_sdaps_choicequestion_var_tl {
616     \sdaps_qobject_append_var:V \l_sdaps_choicequestion_var_tl

```

```

617   }
618
619 \dim_set:Nn \l__sdaps_classic_choiceitem_pad_dim { 1ex }
620
621 \int_set:NV \l__sdaps_classic_choiceitem_cols_int \l_sdaps_choicequestion_cols_int
622 \int_set:Nn \l__sdaps_classic_choiceitem_col_int { 0 }
623 \dim_set:Nn \l__sdaps_classic_choicequestion_prevdepth_dim { 1000pt }
624 \coffin_clear:N \l__sdaps_classic_choicequestion_coffin
625
626 % We have to be in vertical mode at this point.
627 \if_mode_vertical:
628   % Nothing
629 \else:
630   \msg_error:nn { sdapsclassic } { choicequestion_wrong_mode }
631 \fi:
632
633 % It is important to look like a paragraph, otherwise latex thinks there is
634 % nothing between the two question sections and doesn't insert a voluntary
635 % page break.
636 \the\everypar
637
638 % Paragraph like spacing
639 \vspace{\parskip}
640
641 \def\choicequestion_clubpenalty{\penalty\clubpenalty\def\choicequestion_clubpenalty{\relax
642 \def\choicequestion_widowpenalty{%
643 }%
644 \if_mode_vertical:
645   % Nothing
646 \else:
647   \msg_error:nn { sdapsclassic } { choicequestion_wrong_mode }
648 \fi:
649
650 \choicequestion_widowpenalty
651 \vbox:n { \skip_horizontal:N \@totalleftmargin \box_use:N \l__sdaps_classic_choicequestion
652 \prevdepth=\l__sdaps_classic_choicequestion_prevdepth_dim
653 \coffin_clear:N \l__sdaps_classic_choicequestion_coffin
654 \sdaps_qobject_end:n { choicequestion }%
655
656 \group_end:
657 \vspace{\parskip}%
658 }%
659
660 \newenvironment{optionquestion}[2][]{%
661   \choicequestion[singlechoice,#1]{#2}%
662 }{%
663   \endchoicequestion
664 }%
665
666 \cs_new_protected_nopar:Nn \_sdaps_classic_line_shipout_add:NNn

```

```

667 {
668   % We have to be in vertical mode at this point.
669   \if_mode_vertical:
670     % Nothing
671   \else:
672     \msg_error:nn { sdapsclassic } { choicequestion_wrong_mode }
673   \fi:
674
675   % Is linewidth the right thing here?
676   \int_compare:nT { \l_sdaps_classic_choiceitem_col_int + #3 > \l_sdaps_classic_choiceitem_cols_int }
677     % We can only typeset a coffin in vertical mode if we use the box function
678     \vbox:n { \skip_horizontal:N \@totalleftmargin \box_use:N \l_sdaps_classic_choicequestion_coffin
679       \prevdepth=\l_sdaps_classic_choicequestion_prevdepth_dim
680       \skip_vertical:N \l_sdaps_choicequestion_rowsep_dim
681       \coffin_clear:N \l_sdaps_classic_choicequestion_coffin
682       \dim_set:Nn \l_sdaps_classic_choicequestion_prevdepth_dim { 1000pt }
683       \choicequestion_clubpenalty
684       \int_set:Nn \l_sdaps_classic_choiceitem_col_int { 0 }
685     }
686
687   \dim_set:Nn \l_tmpa_dim { \linewidth / \l_sdaps_classic_choiceitem_cols_int + \l_sdaps_classic_choiceitem_col_int }
688   \dim_set:Nn \l_tmpa_dim { \l_sdaps_classic_choiceitem_col_int \l_tmpa_dim }
689
690   \dim_compare:nNnTF { \coffin_dp:N \l_sdaps_classic_choicequestion_coffin } > { \coffin_dp:N \l_sdaps_classic_choicequestion_coffin }
691     \dim_set:Nn \l_sdaps_classic_choicequestion_prevdepth_dim { \dim_min:nn { \l_sdaps_classic_choicequestion_coffin } { \l_sdaps_classic_choicequestion_coffin } }
692   }
693   \dim_set:Nn \l_sdaps_classic_choicequestion_prevdepth_dim { \dim_min:nn { \l_sdaps_classic_choicequestion_coffin } { \l_sdaps_classic_choicequestion_coffin } }
694 }
695
696 \coffin_join:NnnNnnnn \l_sdaps_classic_choicequestion_coffin { 1 } { H } #1 { 1 } { H } +
697 \int_add:Nn \l_sdaps_classic_choiceitem_col_int { #3 }
698 }
699
700 \providecommand*\choiceitem}[2][]{%
701   \tl_clear:N \l_sdaps_choicequestion_choice_var_tl
702   \tl_clear:N \l_sdaps_choicequestion_choice_val_tl
703   \tl_clear:N \l_sdaps_choicequestion_choice_text_tl
704
705   \keys_set:nn { sdaps / choicequestion / choice } { #1 }
706
707   \tl_if_empty:NTF \l_sdaps_choicequestion_choice_text_tl {
708     \sdaps_answer:n { #2 }
709   }
710   \sdaps_answer:V \l_sdaps_choicequestion_choice_text_tl
711 }
712
713 \dim_set:Nn \l_tmpa_dim { \linewidth / \l_sdaps_classic_choiceitem_cols_int - \l_sdaps_classic_choiceitem_col_int }
714
715 \hcoffin_set:Nn \l_tmpa_coffin {
716   \hbox_set:Nn \l_tmpa_box { \strut \sdaps_checkbox:VV \l_sdaps_choicequestion_choice_var }

```

```

717 \dim_set:Nn \l_tmpb_dim { \box_wd:N \l_tmpa_box }
718 \hskip \l_sdaps_choicequestion_colsep_dim
719 \hbox_unpack:N \l_tmpa_box
720 \begin{minipage}[t]{\dim_eval:n { \l_tmpa_dim - \l_tmpb_dim - 2\l_sdaps_choicequestion_c
721     \noindent \usekomafont{choiceitemfont} \tl_trim_spaces:n { #2 }\strut
722     \par
723     \dim_gset:Nn \g_tmpa_dim { \dim_max:nn { \box_dp:N \l_tmpa_box } { \prevdepth } }
724     \unskip
725 \end{minipage}
726 }
727
728 \_sdaps_classic_line_shipout_add:NNn \l_tmpa_coffin \g_tmpa_dim { 1 }
729 \ignorespaces
730 }
731
732 \providetcommand*\choicemulticolitem[3][]{%
733     \tl_clear:N \l_sdaps_choicequestion_choice_var_tl
734     \tl_clear:N \l_sdaps_choicequestion_choice_val_tl
735     \tl_clear:N \l_sdaps_choicequestion_choice_text_tl
736
737     \keys_set:nn { sdaps / choicequestion / choice } { #1 }
738
739     \tl_if_empty:NTF \l_sdaps_choicequestion_choice_text_tl {
740         \sdaps_answer:n { #3 }
741     } {
742         \sdaps_answer:V \l_sdaps_choicequestion_choice_text_tl
743     }
744
745     \dim_set:Nn \l_tmpa_dim { #2\linewidth / \l_sdaps_classic_choiceitem_cols_int - \l_sdaps_
746
747     \hcoffin_set:Nn \l_tmpa_coffin {
748         \hbox_set:Nn \l_tmpa_box { \strut \sdaps_checkbox:VV \l_sdaps_choicequestion_choice_var_
749         \dim_set:Nn \l_tmpb_dim { \box_wd:N \l_tmpa_box }
750         \hskip \l_sdaps_choicequestion_colsep_dim
751         \hbox_unpack:N \l_tmpa_box
752         \begin{minipage}[t]{\dim_eval:n { \l_tmpa_dim - \l_tmpb_dim - 2\l_sdaps_choicequestion_c
753             \noindent \usekomafont{choiceitemfont} \tl_trim_spaces:n { #3 }\strut
754             \par
755             \dim_gset:Nn \g_tmpa_dim { \dim_max:nn { \box_dp:N \l_tmpa_box } { \prevdepth } }
756             \unskip
757 \end{minipage}
758 }
759
760     \_sdaps_classic_line_shipout_add:NNn \l_tmpa_coffin \g_tmpa_dim { #2 }
761     \ignorespaces
762 }
763
764 \providetcommand*\choiceitemtext[4][]{%
765     \tl_clear:N \l_sdaps_choicequestion_choice_var_tl
766     \tl_clear:N \l_sdaps_choicequestion_choice_val_tl

```

```

767 \tl_clear:N \l_sdaps_choicequestion_choice_text_tl
768
769 \keys_set:nn { sdaps / choicequestion / choice } { #1 }
770 % TODO: Warn if val has been set here
771
772 % If not multichoice then warn
773 \tl_if_eq:VnF \l_sdaps_choicequestion_type_tl { multichoice } {
774   \msg_error:nn { sdapsclassic } { choicequestion_no_text }
775 }
776
777 \tl_if_empty:NTF \l_sdaps_choicequestion_choice_text_tl {
778   \sdaps_answer:n { #4 }
779 }
780   \sdaps_answer:V \l_sdaps_choicequestion_choice_text_tl
781 }
782
783 \dim_set:Nn \l_tmpa_dim { #3\linewidth / \l_sdaps_classic_choiceitem_cols_int - \l_sdaps_
784
785 \hcoffin_set:Nn \l_tmpa_coffin {
786   \hskip \l_sdaps_choicequestion_colsep_dim
787   \begin{minipage}[t]{\dim_eval:n { \l_tmpa_dim - 2\l_sdaps_choicequestion_colsep_dim}}
788     \usekomafont{choiceitemfont}
789     \dim_set:Nn \l_tmpa_dim { #2 }
790     \dim_set:Nn \l_tmpb_dim { #2 }
791     \dim_set:Nn \l_tmpa_dim { 0.5 \l_tmpa_dim - 0.8ex }
792     \dim_set:Nn \l_tmpb_dim { 0.5 \l_tmpb_dim + 0.8ex }
793     \strut\ignorespaces \tl_trim_spaces:n { #4 } ~
794     \sdaps_textbox_hstretch:VVVnn \l_sdaps_choicequestion_choice_var_tl \l_tmpa_dim \l_tmpb_dim
795     \par
796     \dim_gset:Nn \g_tmpa_dim { \prevdepth }
797     \unskip
798   \end{minipage}
799 }
800
801 \_sdaps_classic_line_shipout_add:NNn \l_tmpa_coffin \g_tmpa_dim { #3 }
802 \ignorespaces
803 }
804
805
806

```

2.3 The questionnaire environment

There is a bit of magic here. If (and only if) we need to render multiple pages of the questionnaire, then we write the whole questionnaire into a temporary file. This way we can input the file multiple times, which means that environments changing the parser (e.g. verbatim) work properly.

```

807 %
808
809 \iow_new:N \l_sdaps_questionnaire_iow

```

```

810 \bool_new:N \l__sdaps_questionnaire_parse_direct_bool
811
812 \bool_new:N \l__sdaps_questionnaire_info_bool
813
814 \keys_define:nn { sdaps / questionnaire }
815 {
816   info          .bool_set:N = \l__sdaps_questionnaire_info_bool,
817   info          .default:n  = true,
818   info          .initial:n = true,
819
820   noinfo        .meta:n   = { info=false },
821 }
822
823 \cs_new_protected_nopar:Nn \_sdaps_classic_show_info:
824 {
825   \bool_if:NT \l__sdaps_questionnaire_info_bool {
826     \begin{info}
827       \translate{infotext} \\[1ex]
828       \tl_if_eq:VnTF \g_sdaps_checkmode_tl { checkcorrect } {
829         \begin{tabularx}{\textwidth}{Xlll}
830           \checkbox & \strut \translate{info-multi} & \hspace{2em} \checkbox {} & \translate
831           \checkbox* & \strut \translate{info-single} & \hspace{2em} \checkbox* {} & \trans
832         \end{tabularx}
833     } {
834       \tl_if_eq:VnTF \g_sdaps_checkmode_tl { check } {
835         \begin{tabularx}{\textwidth}{Xlll}
836           \checkbox & \strut \translate{info-multi} & \hspace{2em} \checkbox {} & \trans
837           \checkbox* & \strut \translate{info-single} & \hspace{2em} \checkbox* {} & \tr
838         \end{tabularx}
839     } {
840       \tl_if_eq:VnTF \g_sdaps_checkmode_tl { fill } {
841         \begin{tabularx}{\textwidth}{Xlll}
842           \checkbox & \strut \translate{info-multi} & \hspace{2em} \filledbox {} & \tr
843           \checkbox* & \strut \translate{info-single} & \hspace{2em} \filledbox* {} & \tr
844         \end{tabularx}
845     } {
846       \PackageError{sdaps}{Sorry, there is no help text for the checkmode you have cho
847     }
848   }
849 }
850   \end{info}
851 }
852   \ignorespaces
853 }
854
855 \newenvironment { questionnaire } [ 1 ] []
856 {
857   \keys_set:nn { sdaps / questionnaire } { #1 }
858
859   \hypersetup{

```

```

860     pdfauthor      = \@author,
861     pdftitle       = \@title,
862     pdfsubject     = sdaps questionnaire \@title,
863     pdfkeywords    = sdaps questionnaire \@title
864 }
865
866 % If we only have one questionnaire ID, then parse the environment directly,
867 % otherwise write it into a temporary file and input it multiple times.
868 \group_begin:
869
870 \if@twoside
871   \bool_gset_true:N \g_sdaps_twoside_bool
872 \else
873   \bool_gset_false:N \g_sdaps_twoside_bool
874 \fi
875
876 % Enable all metadata writing by default
877 \sdaps_context_enable_writing:
878
879 \sdaps_info_write:x{Author=\exp_not:o{\@author}}
880 \sdaps_info_write:x{Title=\exp_not:o{\@title}}
881
882 \int_compare:nTF { \seq_count:N \g_sdaps_questionnaire_ids_seq <= 1 } {
883   \bool_set_true:N \l__sdaps_questionnaire_parse_direct_bool
884
885   % Set the questionnaire ID
886   \seq_gpop_left:NN \g_sdaps_questionnaire_ids_seq \l_tmpa_tl
887   \sdaps_set_questionnaire_id:V \l_tmpa_tl
888
889   % And, begin the questionnaire
890   \sdaps_begin:
891     \sdaps_classic_show_info:
892 } {
893   \bool_set_false:N \l__sdaps_questionnaire_parse_direct_bool
894
895   % Write content into a file, see "verbatim" documentation for more information.
896   % TODO: Allow multiple temporary files by postfixing with integer?
897   \iow_open:Nn \l__sdaps_questionnaire_iow { \c_sys_jobname_str . questionnaire }
898   \cs_set_eq:NN \do \makeother\dospecials
899   \catcode`\^^M\active
900   \def\verbatim@processline{
901     \iow_now:Nx \l__sdaps_questionnaire_iow {\the\verbatim@line}
902   }
903   \verbatim@
904 }
905 } {
906   \bool_if:NTF \l__sdaps_questionnaire_parse_direct_bool {
907     % Just end everything, nothing else to do.
908     % Make sure we always end the current paragraph.
909     \par

```

```

910      \sdaps_end:
911
912      \group_end:
913  } {
914      % We are done inputting the questionnaire
915      \iow_close:N \l__sdaps_questionnaire_iow
916      \group_end:
917
918      % Now cycle through all IDS and output it again.
919      \group_begin:
920      \seq_map_inline:Nn \g_sdaps_questionnaire_ids_seq {
921          \sdaps_set_questionnaire_id:n { ##1 }
922
923          % Reset a lot of global LaTeX counters
924          \Qifundefined{c@page}{}{\setcounter{page}{1}}
925          \Qifundefined{c@part}{}{\setcounter{part}{0}}
926          \Qifundefined{c@chapter}{}{\setcounter{chapter}{0}}
927          \Qifundefined{c@paragraph}{}{\setcounter{paragraph}{0}}
928          \Qifundefined{c@ subparagraph}{}{\setcounter{subparagraph}{0}}
929          \Qifundefined{c@ section}{}{\setcounter{section}{0}}
930          \Qifundefined{c@ subsection}{}{\setcounter{subsection}{0}}
931          \Qifundefined{c@ subsubsection}{}{\setcounter{subsubsection}{0}}
932          \Qifundefined{c@ equation}{}{\setcounter{equation}{0}}
933          \Qifundefined{c@ figure}{}{\setcounter{figure}{0}}
934          \Qifundefined{c@ table}{}{\setcounter{table}{0}}
935
936      \sdaps_begin:
937          \_sdaps_classic_show_info:
938          \input{ \c_sys_jobname_str . questionnaire }
939          % Make sure we always end the current paragraph.
940          \par
941      \sdaps_end:
942      \newpage
943
944      % Close the output file now (after the page has been shipped out)
945      % XXX: This is a bit of a hack, as disabling metadata writing has side effects
946      \iow_close:N \g_sdaps_infofile_iow
947
948  }
949  \group_end:
950 }
951 }
952
953 \def\sdapspagemark{
954     \sdaps_page_end:
955 }
956

```

2.4 The group environments

Alias for the group environments.

```
957 %
958
959 \tl_new:N \l__sdaps_classic_group_var_tl
960 \tl_new:N \l__sdaps_classic_group_text_tl
961 \tl_new:N \l__sdaps_classic_group_extra_tl
962 \keys_define:nn { sdapsclassic / group }
963 {
964     var          .tl_set:N   = \l__sdaps_classic_group_var_tl,
965     text         .tl_set:N   = \l__sdaps_classic_group_text_tl,
966 }
967
968 \newenvironment { choicegroup } [ 2 ] []
969 {
970     \group_begin:
971     \keys_set_known:nnN { sdapsclassic / group } { #1 } \l__sdaps_classic_group_extra_tl
972
973     \_sdaps_classic_question{#2}
974     \tl_if_empty:NTF \l__sdaps_classic_group_text_tl {
975         \sdaps_qobject_begin:nnn { choicegroup } { Head } { #2 }
976     } {
977         \sdaps_qobject_begin:nnV { choicegroup } { Head } \l__sdaps_classic_group_text_tl
978     }
979
980     \tl_if_empty:NF \l__sdaps_classic_group_var_tl {
981         \sdaps_qobject_append_var:V \l__sdaps_classic_group_var_tl
982     }
983
984     % Undefine the question (and choice) commands in local scope so that they
985     % can be redefined by choicearray without any issues.
986     % Note that \cs_undefine:N works in global scope, so we cannot use it here.
987     \cs_set_eq:NN \question \tex_undefined:D
988     \cs_set_eq:NN \choice \tex_undefined:D
989
990     \expandafter\choicearray\expandafter[\l__sdaps_classic_group_extra_tl]
991     % XXX: This is a bit of a hack, set in global scope because the choicearray
992     % environment does evil things with scopes.
993     \cs_gset_eq:NN \groupaddchoice \choice
994     \cs_gset_eq:NN \choiceline \question
995 }
996 {
997     \cs_gset_eq:NN \groupaddchoice \undefined
998     \cs_gset_eq:NN \choiceline \undefined
999
1000    \endchoicearray
1001
1002    \sdaps_qobject_end:n { choicegroup }
1003 \group_end:
```

```

1004 }
1005
1006 \newenvironment { optiongroup } [ 2 ] []
1007 {
1008   \group_begin:
1009     \keys_set_known:nnN { sdapsclassic / group } { #1 } \l__sdaps_classic_group_extra_tl
1010
1011   \_sdaps_classic_question{#2}
1012   \tl_if_empty:NTF \l__sdaps_classic_group_text_tl {
1013     \sdaps_qobject_begin:nnn { optiongroup } { Head } { #2 }
1014   } {
1015     \sdaps_qobject_begin:nnV { optiongroup } { Head } \l__sdaps_classic_group_text_tl
1016   }
1017
1018   \tl_if_empty:NF \l__sdaps_classic_group_var_tl {
1019     \sdaps_qobject_append_var:V \l__sdaps_classic_group_var_tl
1020   }
1021
1022   % Undefine the question (and choice) commands in local scope so that they
1023   % can be redefined by optionarray without any issues.
1024   % Note that \cs_undefine:N works in global scope, so we cannot use it here.
1025   \cs_set_eq:NN\question\tex_undefined:D
1026   \cs_set_eq:NN\choice\tex_undefined:D
1027
1028   \expandafter\optionarray\expandafter[\l__sdaps_classic_group_extra_tl]
1029     % XXX: This is a bit of a hack, set in global scope because the optionarray
1030     % environment does evil things with scopes.
1031     \cs_gset_eq:NN\groupaddchoice\choice
1032     \cs_gset_eq:NN\optionline\question
1033 }
1034 {
1035   \cs_gset_eq:NN\groupaddoption\undefined
1036   \cs_gset_eq:NN\optionline\undefined
1037
1038   \endoptionarray
1039   \sdaps_qobject_end:n { optiongroup }
1040   \group_end:
1041 }
1042
1043 \newenvironment { markgroup } [ 2 ] []
1044 {
1045   \group_begin:
1046     \keys_set_known:nnN { sdapsclassic / group } { #1 } \l__sdaps_classic_group_extra_tl
1047
1048   \_sdaps_classic_question{#2}
1049   \tl_if_empty:NTF \l__sdaps_classic_group_text_tl {
1050     \sdaps_qobject_begin:nnn { markgroup } { Head } { #2 }
1051   } {
1052     \sdaps_qobject_begin:nnV { markgroup } { Head } \l__sdaps_classic_group_text_tl
1053   }

```

```

1054
1055 \tl_if_empty:NF \l_sdaps_classic_group_var_tl {
1056     \sdaps_qobject_append_var:V \l_sdaps_classic_group_var_tl
1057 }
1058
1059 % Undefine the range command in local scope so that it can be redefined
1060 % by rangearray without any issues.
1061 % Note that \cs_undefine:N works in global scope, so we cannot use it here.
1062 \cs_set_eq:NN\range\tex_undefined:D
1063
1064 \expandafter\rangearray\expandafter[\l_sdaps_classic_group_extra_tl]
1065     % XXX: This is a bit of a hack, set in global scope because the choicearray
1066     % environment does evil things with scopes.
1067     \cs_gset_eq:NN\markline\range
1068 }
1069 {
1070     \cs_gset_eq:NN\markline\undefined
1071
1072     \endrangingarray
1073     \sdaps_qobject_end:n { markgroup }
1074     \group_end:
1075 }
1076
1077 \ExplSyntaxOff
1078

```

Change History

v0.1
 General: Initial version 1