

The sdapsclassic package*

Benjamin Berg
benjamin@sipsolutions.net

December 15, 2019

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 envorinment 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{scrkbase}
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}
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   \KOMASet@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   \KOMASet@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   \KOMACmdset@ncmdkey{checkmode}{@tempa}{%
78     {checkcorrect}{0},%
79     {check}{1},%
80     {fill}{2}%
81   }{#1}%
82   \ifcase \@tempa\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 postions 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{sclayer-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 \@plus 1.5ex \@minus -0.25ex}
205 %\setlength{\parskip}{0pt}
206 \setlength\columnsep{10\p@}
207 \setlength\columnseprule{0\p@}
208 % Redefine the spacings for \subsection ie. \questions.
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 \providecommand{\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\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\fbboxsep
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\bigskipamount \bigskipamount =12pt plus 4pt minus 4pt
324
325
326
327 \cs_generate_variant:Nn \sdaps_textbox_hstretch:nnnnn { nVVnn }
328 \cs_generate_variant:Nn \sdaps_textbox_hstretch:nnnnn { VVVnn }
329 \cs_generate_variant:Nn \int_set:Nn { NV }
330
331

```

2.2 The different question environments

```

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:nnn { 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   \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{\checkedbox}{%
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
469 \group_begin:
470
471 \keys_set:nn { sdaps / singlemark } { #1 }
472
473 \int_compare:nNnT { \l_sdaps_singlemark_count_int } = { 0 } {
474   \int_set:Nn \l_sdaps_singlemark_count_int { \themarkcheckboxcount }
475 }
476
477 \question{#2}%
478
479 \sdaps_qobject_begin:nnn { singlemark } { range } { #2 }
480
481 \sdaps_checkbox_set_type:n { singlechoice }
482
483 \tl_if_empty:NF \l_sdaps_singlemark_var_tl {
484   \sdaps_qobject_append_var:V \l_sdaps_singlemark_var_tl
485 }
486
487 \sdaps_range:nnn{lower}{0}{#3}
488 \sdaps_range:nnn{upper}{\int_use:N\l_sdaps_singlemark_count_int-1}{#4}
489
490 \begin{tabularx}{\linewidth}{X*{\int_use:N\l_sdaps_singlemark_count_int}{c}X}
491   {\hfill\usekomafont{singlemarkchoicefont}\strut\ignorespaces#3} &
492   \int_step_inline:nnnn { 1 } { 1 } { \int_use:N\l_sdaps_singlemark_count_int } { \sdaps
493     {\usekomafont{singlemarkchoicefont}#4\hfill}\%
494 \end{tabularx}%
495
496 \sdaps_qobject_end:n { singlemark }
497 \group_end:
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 \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 \coffin_new:N \l__sdaps_classic_choicequestion_coffin
541
542 \msg_new:nnn { sdapsclassic } { choicequestion_wrong_mode } { Mode~should~always~be~vertical
543
544 \msg_new:nnn { sdapsclassic } { choicequestion_no_text } { Textboxes~cannot~be~used~in~singl
545 It~is~currently~not~supported~to~use~text~answers~in~single~choice~questions.~Until~support
546 you~can~get~similar~results~by~temporarily~changing~the~checkbox~style. }
547
548 \msg_new:nnn { sdapsclassic } { choicequestion_unknown_key } { The~key~'#1'~is~unknown.\\If
549
550 \tl_new:N \l_sdaps_choicquestion_type_tl
551 \tl_new:N \l_sdaps_choicequestion_var_tl
552 \tl_new:N \l_sdaps_choicequestion_text_tl
553 \int_new:N \l_sdaps_choicequestion_cols_int
554
555 \keys_define:nn { sdaps / choicequestion }
556 {
557   var      .tl_set:N = \l_sdaps_choicequestion_var_tl,
558   text     .tl_set:N = \l_sdaps_choicequestion_text_tl,
559   colsep   .dim_set:N = \l_sdaps_choicequestion_colsep_dim,
560   colsep   .initial:n = 6pt,
561   rowsep   .dim_set:N = \l_sdaps_choicequestion_rowsep_dim,
562   rowsep   .initial:n = 0pt,
563   cols     .int_set:N = \l_sdaps_choicequestion_cols_int,
564   cols     .initial:n = 3,
565
566   type     .choices:nn = { multichoice, singlechoice } { \tl_set:Nx \l_sdaps_choicquestion

```

```

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

```

```

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

```

```

667 % Nothing
668 \else:
669 \msg_error:nn { sdapsclassic } { choicequestion_wrong_mode }
670 \fi:
671
672 % Is linewidth the right thing here?
673 \int_compare:nT { \l__sdaps_classic_choiceitem_col_int + #2 > \l__sdaps_classic_choiceitem
674 % We can only typeset a coffin in vertical mode if we use the box function
675 \vbox:n { \skip_horizontal:N \@totalleftmargin \box_use:N \l__sdaps_classic_choicequesti
676 \skip_vertical:N \l_sdaps_choicequestion_rowsep_dim
677 \coffin_clear:N \l__sdaps_classic_choicequestion_coffin
678 \choicequestion_clubpenalty
679 \int_set:Nn \l__sdaps_classic_choiceitem_col_int { 0 }
680 }
681
682 \dim_set:Nn \l_tmpa_dim { \linewidth / \l__sdaps_classic_choiceitem_cols_int + \l__sdaps_c
683 \dim_set:Nn \l_tmpa_dim { \l__sdaps_classic_choiceitem_col_int \l_tmpa_dim }
684
685 \coffin_join:NnnNnnnn \l__sdaps_classic_choicequestion_coffin { 1 } { H } #1 { 1 } { H } +
686 \int_add:Nn \l__sdaps_classic_choiceitem_col_int { #2 }
687 }
688
689 \providecommand*{\choiceitem}[2][\%
690 \tl_clear:N \l_sdaps_choicequestion_choice_var_tl
691 \tl_clear:N \l_sdaps_choicequestion_choice_val_tl
692 \tl_clear:N \l_sdaps_choicequestion_choice_text_tl
693
694 \keys_set:nn { sdaps / choicequestion / choice } { #1 }
695
696 \tl_if_empty:NTF \l_sdaps_choicequestion_choice_text_tl {
697 \sdaps_answer:n { #2 }
698 } {
699 \sdaps_answer:V \l_sdaps_choicequestion_choice_text_tl
700 }
701
702 \dim_set:Nn \l_tmpa_dim { \linewidth / \l__sdaps_classic_choiceitem_cols_int - \l__sdaps_c
703
704 \hcoffin_set:Nn \l_tmpa_coffin {
705 \hbox_set:Nn \l_tmpa_box { \strut \sdaps_checkbox:VV \l_sdaps_choicequestion_choice_var_
706 \dim_set:Nn \l_tmpb_dim { \box_wd:N \l_tmpa_box }
707 \hskip \l_sdaps_choicequestion_colsep_dim
708 \hbox_unpack_clear:N \l_tmpa_box
709 \begin{minipage}[t]{\dim_eval:n { \l_tmpa_dim - \l_tmpb_dim - 2\l_sdaps_choicequestion_c
710 \noindent \usekomafont{choiceitemfont} \tl_trim_spaces:n { #2 }
711 \end{minipage}
712 }
713
714 \_sdaps_classic_line_shipout_add:Nn \l_tmpa_coffin { 1 }
715 \ignorespaces
716 }

```

```

717
718 \providecommand*{\choicemulticolitem}[3][]{%
719   \tl_clear:N \l_sdaps_choicequestion_choice_var_tl
720   \tl_clear:N \l_sdaps_choicequestion_choice_val_tl
721   \tl_clear:N \l_sdaps_choicequestion_choice_text_tl
722
723   \keys_set:nn { sdaps / choicequestion / choice } { #1 }
724
725   \tl_if_empty:NTF \l_sdaps_choicequestion_choice_text_tl {
726     \sdaps_answer:n { #3 }
727   } {
728     \sdaps_answer:V \l_sdaps_choicequestion_choice_text_tl
729   }
730
731   \dim_set:Nn \l_tmpa_dim { #2\linewidth / \l__sdaps_classic_choiceitem_cols_int - \l__sdaps
732
733   \hcoffin_set:Nn \l_tmpa_coffin {
734     \hbox_set:Nn \l_tmpa_box { \strut \sdaps_checkbox:VV \l_sdaps_choicequestion_choice_var_
735     \dim_set:Nn \l_tmpb_dim { \box_wd:N \l_tmpa_box }
736     \hskip \l_sdaps_choicequestion_colsep_dim
737     \hbox_unpack_clear:N \l_tmpa_box
738     \begin{minipage}[t]{\dim_eval:n { \l_tmpa_dim - \l_tmpb_dim - 2\l_sdaps_choicequestion_c
739       \noindent \usekomafont{choiceitemfont} \tl_trim_spaces:n { #3 }
740     \end{minipage}
741   }
742
743   \_sdaps_classic_line_shipout_add:Nn \l_tmpa_coffin { #2 }
744   \ignorespaces
745 }
746
747 \providecommand{\choiceitemtext}[4][]{%
748   \tl_clear:N \l_sdaps_choicequestion_choice_var_tl
749   \tl_clear:N \l_sdaps_choicequestion_choice_val_tl
750   \tl_clear:N \l_sdaps_choicequestion_choice_text_tl
751
752   \keys_set:nn { sdaps / choicequestion / choice } { #1 }
753   % TODO: Warn if val has been set here
754
755   % If not multichoice then warn
756   \tl_if_eq:VnF \l_sdaps_choicquestion_type_tl { multichoice } {
757     \msg_error:nn { sdapsclassic } { choicequestion_no_text }
758   }
759
760   \tl_if_empty:NTF \l_sdaps_choicequestion_choice_text_tl {
761     \sdaps_answer:n { #4 }
762   } {
763     \sdaps_answer:V \l_sdaps_choicequestion_choice_text_tl
764   }
765
766   \dim_set:Nn \l_tmpa_dim { #3\linewidth / \l__sdaps_classic_choiceitem_cols_int - \l__sdaps

```



```

767
768 \hcoffin_set:Nn \l_tmpa_coffin {
769   \hskip \l_sdaps_choicequestion_colsep_dim
770   \begin{minipage}[t]{\dim_eval:n { \l_tmpa_dim - 2\l_sdaps_choicequestion_colsep_dim}}
771     \usekomafont{choiceitemfont}
772     \dim_set:Nn \l_tmpa_dim { #2 }
773     \dim_set:Nn \l_tmpb_dim { #2 }
774     \dim_set:Nn \l_tmpa_dim { 0.5 \l_tmpa_dim - 0.8ex }
775     \dim_set:Nn \l_tmpb_dim { 0.5 \l_tmpb_dim + 0.8ex }
776     \strut\ignorespaces \tl_trim_spaces:n { #4 } ~
777     \sdaps_textbox_hstretch:VVVnn \l_sdaps_choicequestion_choice_var_tl \l_tmpa_dim \l_tmpb_dim
778   \end{minipage}
779 }
780
781 \_sdaps_classic_line_shipout_add:Nn \l_tmpa_coffin { #3 }
782 \ignorespaces
783 }
784
785
786

```

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.

```

787 %
788
789 \iow_new:N \l__sdaps_questionnaire_iow
790 \bool_new:N \l__sdaps_questionnaire_parse_direct_bool
791
792 \bool_new:N \l__sdaps_questionnaire_info_bool
793
794 \keys_define:nn { sdaps / questionnaire }
795 {
796   info          .bool_set:N = \l__sdaps_questionnaire_info_bool,
797   info          .default:n  = true,
798   info          .initial:n   = true,
799
800   noinfo        .meta:n     = { info=false },
801 }
802
803 \cs_new_protected_nopar:Nn \_sdaps_classic_show_info:
804 {
805   \bool_if:NT \l__sdaps_questionnaire_info_bool {
806     \begin{info}
807       \translate{infotext} \[1ex]
808       \tl_if_eq:VnTF \g_sdaps_checkmode_tl { checkcorrect } {
809         \begin{tabularx}{\textwidth}{1X1111}

```

```

810         \checkbox & \strut \translate{info-multi} & \hspace{2em} \checkbox {} & \transla
811         \checkbox* & \strut \translate{info-single} & \hspace{2em} \checkbox* {} & \tran
812     \end{tabularx}
813 } {
814     \tl_if_eq:VnTF \g_sdaps_checkmode_tl { check } {
815         \begin{tabularx}{\textwidth}{lXl}
816             \checkbox & \strut \translate{info-multi} & \hspace{2em} \checkbox {} & \transla
817             \checkbox* & \strut \translate{info-single} & \hspace{2em} \checkbox* {} & \tran
818         \end{tabularx}
819     } {
820         \tl_if_eq:VnTF \g_sdaps_checkmode_tl { fill } {
821             \begin{tabularx}{\textwidth}{lXl}
822                 \checkbox & \strut \translate{info-multi} & \hspace{2em} \checkbox {} & \transla
823                 \checkbox* & \strut \translate{info-single} & \hspace{2em} \checkbox* {} & \tran
824             \end{tabularx}
825         } {
826             \PackageError{sdaps}{Sorry, there is no help text for the checkmode you have cho
827         }
828     }
829 }
830 \end{info}
831 }
832 \ignorespaces
833 }
834
835 \newenvironment { questionnaire } [ 1 ] []
836 {
837     \keys_set:nn { sdaps / questionnaire } { #1 }
838
839     \hypersetup{
840         pdfauthor      = \@author,
841         pdftitle       = \@title,
842         pdfsubject     = sdaps questionnaire \@title,
843         pdfkeywords    = sdaps questionnaire \@title
844     }
845
846     % If we only have one questionnaire ID, then parse the environment directly,
847     % otherwise write it into a temporary file and input it multiple times.
848     \group_begin:
849
850     \if@twoside
851         \bool_gset_true:N \g_sdaps_twoside_bool
852     \else
853         \bool_gset_false:N \g_sdaps_twoside_bool
854     \fi
855
856     % Enable all metadata writing by default
857     \sdaps_context_enable_writing:
858
859     \sdaps_info_write:x{Author=\exp_not:o{\@author}}

```

```

860 \sdaps_info_write:x{Title=\exp_not:o{\@title}}
861
862 \int_compare:nTF { \seq_count:N \g_sdaps_questionnaire_ids_seq <= 1 } {
863   \bool_set_true:N \l__sdaps_questionnaire_parse_direct_bool
864
865   % Set the questionnaire ID
866   \seq_gpop_left:NN \g_sdaps_questionnaire_ids_seq \l_tmpa_tl
867   \sdaps_set_questionnaire_id:V \l_tmpa_tl
868
869   % And, begin the questionnaire
870   \sdaps_begin:
871   \sdaps_classic_show_info:
872 } {
873   \bool_set_false:N \l__sdaps_questionnaire_parse_direct_bool
874
875   % Write content into a file, see "verbatim" documentation for more information.
876   % TODO: Allow multiple temporary files by postfixing with integer?
877   \iow_open:Nn \l__sdaps_questionnaire_iow { \c_sys_jobname_str . questionnaire }
878   \cs_set_eq:NN\do\@makeother\dospecials
879   \catcode'\^M\active
880   \def\verbatim@processline{
881     \iow_now:Nx \l__sdaps_questionnaire_iow {\the\verbatim@line}
882   }
883   \verbatim@
884 }
885 } {
886   \bool_if:NTF \l__sdaps_questionnaire_parse_direct_bool {
887     % Just end everything, nothing else to do.
888     % Make sure we always end the current paragraph.
889     \par
890     \sdaps_end:
891
892     \group_end:
893   } {
894     % We are done inputting the questionnaire
895     \iow_close:N \l__sdaps_questionnaire_iow
896     \group_end:
897
898     % Now cycle through all IDS and output it again.
899     \group_begin:
900     \seq_map_inline:Nn \g_sdaps_questionnaire_ids_seq {
901       \sdaps_set_questionnaire_id:n { ##1 }
902
903       % Reset a lot of global LaTeX counters
904       \@ifundefined{c@page}{}{\setcounter{page}{1}}
905       \@ifundefined{c@part}{}{\setcounter{part}{0}}
906       \@ifundefined{c@chapter}{}{\setcounter{chapter}{0}}
907       \@ifundefined{c@paragraph}{}{\setcounter{paragraph}{0}}
908       \@ifundefined{c@subparagraph}{}{\setcounter{subparagraph}{0}}
909       \@ifundefined{c@section}{}{\setcounter{section}{0}}

```

```

910     \ifundefined{c@subsection}{\setcounter{subsection}{0}}
911     \ifundefined{c@subsubsection}{\setcounter{subsubsection}{0}}
912     \ifundefined{c@equation}{\setcounter{equation}{0}}
913     \ifundefined{c@figure}{\setcounter{figure}{0}}
914     \ifundefined{c@table}{\setcounter{table}{0}}
915
916     \sdaps_begin:
917         \_sdaps_classic_show_info:
918         \input{ \c_sys_jobname_str . questionnaire }
919         % Make sure we always end the current paragraph.
920         \par
921     \sdaps_end:
922     \newpage
923
924     % Close the output file now (after the page has been shipped out)
925     % XXX: This is a bit of a hack, as disabling metadata writing has side effects
926     \iow_close:N \g_sdaps_infofile_iow
927
928     }
929 \group_end:
930 }
931 }
932
933 \def\sdapspagemark{
934     \sdaps_page_end:
935 }
936

```

2.4 The group environments

Alias for the group environments.

```

937 %
938
939
940 \newenvironment { choicegroup } [ 2 ] []
941 {
942     \group_begin:
943
944     \question{#2}
945
946     % Undefine the question (and choice) commands in local scope so that they
947     % can be redefined by choicearray without any issues.
948     % Note that \cs_undefine:N works in global scope, so we cannot use it here.
949     \cs_set_eq:NN\question\tex_undefined:D
950     \cs_set_eq:NN\choice\tex_undefined:D
951
952     \begin{choicearray}[#1]{#2}
953     % XXX: This is a bit of a hack, set in global scope because the choicearray
954     %       environment does evil things with scopes.

```

```

955     \cs_gset_eq:NN\groupaddchoice\choice
956     \cs_gset_eq:NN\choiceline\question
957 }
958 {
959     \cs_gset_eq:NN\groupaddchoice\undefined
960     \cs_gset_eq:NN\choiceline\undefined
961
962     \end{choicearray}
963     \group_end:
964 }
965
966 \newenvironment { optiongroup } [ 2 ] []
967 {
968     \group_begin:
969
970     \question{#2}
971
972     % Undefine the question (and choice) commands in local scope so that they
973     % can be redefined by optionarray without any issues.
974     % Note that \cs_undefine:N works in global scope, so we cannot use it here.
975     \cs_set_eq:NN\question\tex_undefined:D
976     \cs_set_eq:NN\choice\tex_undefined:D
977
978     \begin{optionarray}[#1]{#2}
979     % XXX: This is a bit of a hack, set in global scope because the optionarray
980     %       environment does evil things with scopes.
981     \cs_gset_eq:NN\groupaddchoice\choice
982     \cs_gset_eq:NN\optionline\question
983 }
984 {
985     \cs_gset_eq:NN\groupaddoption\undefined
986     \cs_gset_eq:NN\optionline\undefined
987
988     \end{optionarray}
989     \group_end:
990 }
991
992 \newenvironment { markgroup } [ 2 ] []
993 {
994     \group_begin:
995
996     \question{#2}
997
998     % Undefine the question (and choice) commands in local scope so that they
999     % can be redefined by choicearray without any issues.
1000     % Note that \cs_undefine:N works in global scope, so we cannot use it here.
1001     \cs_set_eq:NN\range\tex_undefined:D
1002
1003     \begin{rangearray}[#1]{#2}
1004     % XXX: This is a bit of a hack, set in global scope because the choicearray

```

```

1005      %      environment does evil things with scopes.
1006      \cs_gset_eq:NN\markline\range
1007 }
1008 {
1009      \cs_gset_eq:NN\markline\undefined
1010
1011      \end{rangearray}
1012      \group_end:
1013 }
1014
1015 \ExplSyntaxOff
1016

```

Change History

v0.1
General: Initial version 1