

The `xepersian-hm` package Fixing kashida in xepersian Source code documentation

Hossein Movahhedian*

Released 2020-08-05 v1.0d

1 `xepersian-hm` Implementation

1.1 File: `xepersian-hm.sty`

```
1 < *xepersian-hm-sty >
2 \NeedsTeXFormat{LaTeX2e}
3 \RequirePackage{expl3}[2020-03-06]
4 \@ifpackagelater{expl3}{2020-03-06}
5 {}
6 {%
7   \PackageError{xepersian-hm}{Support package l3kernel too old}
8   {%
9     Please install an up to date version of l3kernel\MessageBreak
10    using your TeX package manager or from CTAN.\MessageBreak
11    \MessageBreak
12    Loading xtemplate will abort!%
13   }%
14   \endinput
15 }
16 \RequirePackage{l3keys2e}
17 \RequirePackage{graphicx}[2019-11-30]
18 \RequirePackage{zref-savepos}[2020-03-03]
19 \RequirePackage{xepersian}
20 \ExplSyntaxOn
21 \ProvidesExplPackage {xepersian-hm} {2020-08-05} {1.0d} { Fixes-issues-in-xepersian-package }
22
23 \box_new:N \l_xephm_ksh_box
24
25 \int_const:Nn \c_xephm_ksh_int {"0640} % kashida
26 \int_const:Nn \c_xephm_lrm_int {"200E} % left-right-mark
27 \int_const:Nn \c_xephm_zwj_int {"200D} % zero-width joiner
```

*E-mail: dma8hm1334@gmail.com

```

28
29 \int_const:Nn \c_xephm_two_int {2} % 2
30 \int_const:Nn \c_xephm_four_int {4} % 4
31
32 \tl_const:Nn \c_xephm_skip_a_tl { 0 em plus 0.5 em }
33 \tl_const:Nn \c_xephm_skip_b_tl { 0.14 em plus 5.5 em }
34
35 \int_new:N \l_xephm_counter_int
36
37 \int_new:N \l_xephm_kashida_slot_int
38
39 \int_new:N \l_xephm_line_break_penalty_int
40
41 \int_new:N \l_xephm_min_penalty_int
42 \int_new:N \l_xephm_low_penalty_int
43 \int_new:N \l_xephm_med_penalty_int
44 \int_new:N \l_xephm_high_penalty_int
45 \int_new:N \l_xephm_max_penalty_int
46
47 \tl_new:N \l_xephm_line_break_tl
48
49 \tl_new:N \l_xephm_main_font_full_tl
50 \tl_new:N \l_xephm_main_font_name_tl
51
52 \tl_new:N \l_xephm_font_full_tl
53 \tl_new:N \l_xephm_font_name_tl
54
55 \tl_new:N \l_xephm_skip_default_tl
56
57 \tl_new:N \l_xephm_active_ligs_tl
58
59 \tl_new:N \l_xephm_gap_filler_tl
60
61 \dim_new:N \l_xephm_diff_pos_dim
62
63 \bool_set_false:N \l_xephm_kashida_hm_fix_bool
64 \bool_set_false:N \l_xephm_ligature_bool
65 \bool_set_false:N \l_xephm_linebreakpenalty_bool
66
67 \int_set:Nn \l_xephm_min_penalty_int { 0 }
68 \int_set:Nn \l_xephm_low_penalty_int { 8 }
69 \int_set:Nn \l_xephm_med_penalty_int { 15 }
70 \int_set:Nn \l_xephm_high_penalty_int { 25 }
71 \int_set:Nn \l_xephm_max_penalty_int { 10000 }
72
73 \tl_set:Nn \l_xephm_stretch_glyph_tl { glyph }
74 \tl_set:Nn \l_xephm_stretch_leaders_glyph_tl { leaders+glyph }
75 \tl_set:Nn \l_xephm_stretch_leaders_hrule_tl { leaders+hrule }
76 \tl_set:Nn \l_xephm_stretch_off_tl { Off }
77 \tl_set:Nn \l_xephm_stretch_on_tl { On }

```

78

```
79 \tl_set:Nn \l_xephm_fnt_kayhan_tl      { kayhan }
80 \tl_set:Nn \l_xephm_fnt_kayhannavaar_tl { kayhannavaar }
81 \tl_set:Nn \l_xephm_fnt_kayhanpook_tl  { kayhanpook }
82 \tl_set:Nn \l_xephm_fnt_kayhansayeh_tl { kayhansayeh }
83 \tl_set:Nn \l_xephm_fnt_khoramshahr_tl { khoramshahr }
84 \tl_set:Nn \l_xephm_fnt_khorramshahr_tl { khorrasmshahr }
85 \tl_set:Nn \l_xephm_fnt_niloofar_tl    { niloofar }
86 \tl_set:Nn \l_xephm_fnt_paatch_tl      { paatch }
87 \tl_set:Nn \l_xephm_fnt_riyaz_tl       { riyaz }
88 \tl_set:Nn \l_xephm_fnt_roya_tl        { roya }
89 \tl_set:Nn \l_xephm_fnt_shafigh_tl     { shafigh }
90 \tl_set:Nn \l_xephm_fnt_shafighKurd_tl { shafighKurd }
91 \tl_set:Nn \l_xephm_fnt_shafighUzbek_tl { shafighUzbek }
92 \tl_set:Nn \l_xephm_fnt_shiraz_tl      { shiraz }
93 \tl_set:Nn \l_xephm_fnt_sols_tl        { sols }
94 \tl_set:Nn \l_xephm_fnt_tabriz_tl      { tabriz }
95 \tl_set:Nn \l_xephm_fnt_titr_tl        { titr }
96 \tl_set:Nn \l_xephm_fnt_titre_tl       { titre }
97 \tl_set:Nn \l_xephm_fnt_traffic_tl     { traffic }
98 \tl_set:Nn \l_xephm_fnt_vahid_tl       { vahid }
99 \tl_set:Nn \l_xephm_fnt_vosta_tl       { vosta }
100 \tl_set:Nn \l_xephm_fnt_yaghut_tl      { yaghut }
101 \tl_set:Nn \l_xephm_fnt_yagut_tl       { yagut }
102 \tl_set:Nn \l_xephm_fnt_yas_tl         { yas }
103 \tl_set:Nn \l_xephm_fnt_yekan_tl       { yekan }
104 \tl_set:Nn \l_xephm_fnt_yermook_tl     { yermook }
105 \tl_set:Nn \l_xephm_fnt_zar_tl         { zar }
106 \tl_set:Nn \l_xephm_fnt_ziba_tl        { ziba }
107 \tl_set:Nn \l_xephm_fnt_default_tl     { default }
108 \tl_set:Nn \l_xephm_fnt_noskip_tl      { noskip }
109
110 \tl_set:Nn \l_xephm_lig_aalt_tl         { aalt } % Access All Alternatives
111 \tl_set:Nn \l_xephm_lig_ccmp_tl         { ccmp } % Glyph Composition/Decomposition
112 \tl_set:Nn \l_xephm_lig_dlig_tl         { dlig } % Discretionary Ligatures
113 \tl_set:Nn \l_xephm_lig_fina_tl         { fina } % Final (Terminal) Forms
114 \tl_set:Nn \l_xephm_lig_init_tl         { init } % Initial Forms
115 \tl_set:Nn \l_xephm_lig_locl_tl         { locl } % Localized Forms
116 \tl_set:Nn \l_xephm_lig_medi_tl         { medi } % Medial Forms
117 \tl_set:Nn \l_xephm_lig_rlig_tl         { rlig } % Required Ligatures
118 \tl_set:Nn \l_xephm_lig_default_tl     { default }
119
120 \clist_set:Nn \l_xephm_lig_aalt_clist    { } % Access All Alternatives
121 \clist_set:Nn \l_xephm_lig_ccmp_clist    { } % Glyph Composition/Decomposition
122 \clist_set:Nn \l_xephm_lig_dlig_clist    { FDF2 = , FDF3 = , FDFB = { % Discretionary Ligatures
123 \clist_set:Nn \l_xephm_lig_fina_clist    { } % Final (Terminal) Forms
124 \clist_set:Nn \l_xephm_lig_init_clist    { } % Initial Forms
125 \clist_set:Nn \l_xephm_lig_locl_clist    { } % Localized Forms
126 \clist_set:Nn \l_xephm_lig_medi_clist    { } % Medial Forms
127 \clist_set:Nn \l_xephm_lig_rlig_clist    { } % Required Ligatures
```

```

128 \clist_set:Nn \l_xephm_lig_default_clist { }
129
130 \clist_set:Nn \l_xephm_lig_names_clist
131 {
132   \l_xephm_lig_aalt_tl , { \l_xephm_lig_aalt_clist } ,
133   \l_xephm_lig_ccmp_tl , { \l_xephm_lig_ccmp_clist } ,
134   \l_xephm_lig_dlig_tl , { \l_xephm_lig_dlig_clist } ,
135   \l_xephm_lig_fina_tl , { \l_xephm_lig_fina_clist } ,
136   \l_xephm_lig_init_tl , { \l_xephm_lig_init_clist } ,
137   \l_xephm_lig_locl_tl , { \l_xephm_lig_locl_clist } ,
138   \l_xephm_lig_medi_tl , { \l_xephm_lig_medi_clist } ,
139   \l_xephm_lig_rlig_tl , { \l_xephm_lig_rlig_clist } ,
140 }
141
142 \msg_new:nnn { xepersian-hm } { error-kashida-character-is-not-available-in-the-main-font }
143 {
144   Sorry,~ kashida~ character~ is~ not~ available~ in~ the~ main~ font~#1!
145 }
146
147 \msg_new:nnn { xepersian-hm } { error-value-not-available-for-kashida-option }
148 {
149   Sorry,~ value~ `#1'~ is~ not~ available~ for~ `Kashida'~ option~ yet~!
150 }
151
152 \msg_new:nnn { xepersian-hm } { error-specify-value-for-kashida-option }
153 {
154   Sorry,~ you~ must~ specify~ a~ value~ for~ `Kashida'~ option~ yet~!
155 }
156
157 \msg_new:nnn { xepersian-hm } { warning-experimental-feature }
158 {
159   Please~ note~ that~ the~ feature~ `#1'~ is~ still~ experimental~
160   and~ is~ not~ regarded~ as~ stable.
161 }
162
163 \msg_new:nnn { xepersian-hm } { hm-series-font-not-found }
164 {
165   Either~ the~ font~`#1'~ is~ not~ installed~ on~ your~ system~ or~ does~ not~
166   belong~ to~ HM~Series~fonts.~
167   Please~ note~ that~ the~ option~ `Kashida=leaders+glyph'~ is~ currently~ only~
168   supported~ by~ HM~Series~fonts.~
169   If~ you~ know~ of~ any~ other~ font~ that~ supports~ this~ option,~ please~
170   let~ me~ know~ to~ add~ it~ to~ the~ list~ of~ corresponding~ fonts.~
171 }
172
173 \keys_define:nn { xepersian-hm }
174 {
175   Kashida .code:n =
176   {
177     \tl_set:Nn \l_tmpa_tl { #1 }

```

```

178     \tl_case:NnTF \l_tmpa_tl
179     {
180         \l_xephm_stretch_glyph_tl
181         {
182             \msg_warning:nnn { xepersian-hm } { warning-experimental-feature } { Kashida=glyph }
183             \tl_set:Nx \l_xephm_gap_filler_tl { \l_xephm_stretch_glyph_tl }
184             \AtBeginDocument
185             {
186                 \tl_set:Nx \l_xephm_main_font_full_tl { \tex_fontname:D \tex_the:D \tex_font:D }
187                 \tl_set:Nx \l_xephm_main_font_name_tl { \l_xephm_main_font_full_tl }
188                 \regex_replace_once:nnN { ^"([~/]+)/.* } { \1 } \l_xephm_main_font_name_tl
189                 \int_set:Nn \l_xephm_kashida_slot_int { \XeTeXcharglyph \c_xephm_ksh_int }
190                 \int_compare:nT { \l_xephm_kashida_slot_int = \c_zero_int }
191                 {
192                     \msg_error:nxx { xepersian-hm } { error-kashida-character-is-not-available-in }
193                 }
194             }
195             \bool_set_true:N \l_xephm_kashida_hm_fix_bool
196         }
197         \l_xephm_stretch_leaders_glyph_tl
198         {
199             \tl_set:Nx \l_xephm_gap_filler_tl { \l_xephm_stretch_leaders_glyph_tl }
200             \bool_set_true:N \l_xephm_kashida_hm_fix_bool
201         }
202         \l_xephm_stretch_leaders_hruler_tl
203         {
204             \tl_set:Nx \l_xephm_gap_filler_tl { \l_xephm_stretch_leaders_hruler_tl }
205             \bool_set_true:N \l_xephm_kashida_hm_fix_bool
206         }
207         \l_xephm_stretch_off_tl
208         {
209             \tl_set:Nx \l_xephm_gap_filler_tl { \l_xephm_stretch_off_tl }
210             \bool_set_false:N \l_xephm_kashida_hm_fix_bool
211         }
212         \l_xephm_stretch_on_tl
213         {
214             \tl_set:Nx \l_xephm_gap_filler_tl { \l_xephm_stretch_leaders_glyph_tl }
215             \bool_set_true:N \l_xephm_kashida_hm_fix_bool
216         }
217     } { } { \tl_set:Nx \l_xephm_gap_filler_tl { #1 } }
218     \tl_if_empty:NT \l_xephm_gap_filler_tl { \msg_error:nn { xepersian-hm } { error-specify-value } }
219 } ,
220
221 linebreakpenalty .code:n =
222 {
223     \int_set:Nn \l_tmpa_int { #1 }
224     \tl_case:nnTF \l_tmpa_int
225     {
226         \l_xephm_min_penalty_int { \int_set:Nn \l_xephm_line_break_penalty_int { \l_xephm_min_penalty_int } }
227         \l_xephm_low_penalty_int { \int_set:Nn \l_xephm_line_break_penalty_int { \l_xephm_low_penalty_int } }

```

```

228     \l_xephm_med_penalty_int { \int_set:Nn \l_xephm_line_break_penalty_int { \l_xephm_med_p
229     \l_xephm_high_penalty_int { \int_set:Nn \l_xephm_line_break_penalty_int { \l_xephm_high_p
230     \l_xephm_max_penalty_int { \int_set:Nn \l_xephm_line_break_penalty_int { \l_xephm_max_p
231   } { } { \int_set:Nn \l_xephm_line_break_penalty_int { #1 } }
232   \bool_set_true:N \l_xephm_linebreakpenalty_bool
233 } ,
234
235 kashidastretch .code:n =
236 {
237   \tl_set:Nn \l_tmpa_tl { #1 }
238   \tl_case:NnTF \l_tmpa_tl
239   {
240     \l_xephm_fnt_kayhan_tl      { \tl_set:Nn \l_xephm_skip_default_tl { 0.14 em plus 0.5
241     \l_xephm_fnt_kayhannavaar_tl { \tl_set:Nn \l_xephm_skip_default_tl { 0.129 em plus 0.5
242     \l_xephm_fnt_kayhanpook_tl  { \tl_set:Nn \l_xephm_skip_default_tl { 0.133 em plus 0.5
243     \l_xephm_fnt_kayhansayeh_tl { \tl_set:Nn \l_xephm_skip_default_tl { 0.135 em plus 0.5
244     \l_xephm_fnt_khoramshahr_tl { \tl_set:Nn \l_xephm_skip_default_tl { 0.128 em plus 0.5
245     \l_xephm_fnt_khorramshahr_tl { \tl_set:Nn \l_xephm_skip_default_tl { 0.13 em plus 0.5
246     \l_xephm_fnt_niloofar_tl   { \tl_set:Nn \l_xephm_skip_default_tl { 0.132 em plus 0.5
247     \l_xephm_fnt_paatch_tl     { \tl_set:Nn \l_xephm_skip_default_tl { 0.127 em plus 0.5
248     \l_xephm_fnt_riyaz_tl      { \tl_set:Nn \l_xephm_skip_default_tl { 0.125 em plus 0.5
249     \l_xephm_fnt_roya_tl       { \tl_set:Nn \l_xephm_skip_default_tl { 0.142 em plus 0.5
250     \l_xephm_fnt_shafigh_tl    { \tl_set:Nn \l_xephm_skip_default_tl { 0.143 em plus 0.5
251     \l_xephm_fnt_shafighKurd_tl { \tl_set:Nn \l_xephm_skip_default_tl { 0.126 em plus 0.5
252     \l_xephm_fnt_shafighUzbek_tl { \tl_set:Nn \l_xephm_skip_default_tl { 0.123 em plus 0.5
253     \l_xephm_fnt_shiraz_tl     { \tl_set:Nn \l_xephm_skip_default_tl { 0.122 em plus 0.5
254     \l_xephm_fnt_sols_tl       { \tl_set:Nn \l_xephm_skip_default_tl { 0.124 em plus 0.5
255     \l_xephm_fnt_tabriz_tl     { \tl_set:Nn \l_xephm_skip_default_tl { 0.119 em plus 0.5
256     \l_xephm_fnt_titr_tl       { \tl_set:Nn \l_xephm_skip_default_tl { 0.12 em plus 0.5
257     \l_xephm_fnt_titre_tl      { \tl_set:Nn \l_xephm_skip_default_tl { 0.121 em plus 0.5
258     \l_xephm_fnt_traffic_tl    { \tl_set:Nn \l_xephm_skip_default_tl { 0.124 em plus 0.5
259     \l_xephm_fnt_vahid_tl      { \tl_set:Nn \l_xephm_skip_default_tl { 0.134 em plus 0.5
260     \l_xephm_fnt_vosta_tl     { \tl_set:Nn \l_xephm_skip_default_tl { 0.136 em plus 0.5
261     \l_xephm_fnt_yaghut_tl     { \tl_set:Nn \l_xephm_skip_default_tl { 0.138 em plus 0.5
262     \l_xephm_fnt_yagut_tl      { \tl_set:Nn \l_xephm_skip_default_tl { 0.137 em plus 0.5
263     \l_xephm_fnt_yas_tl        { \tl_set:Nn \l_xephm_skip_default_tl { 0.126 em plus 0.5
264     \l_xephm_fnt_yekan_tl     { \tl_set:Nn \l_xephm_skip_default_tl { 0.141 em plus 0.5
265     \l_xephm_fnt_yermook_tl    { \tl_set:Nn \l_xephm_skip_default_tl { 0.139 em plus 0.5
266     \l_xephm_fnt_zar_tl        { \tl_set:Nn \l_xephm_skip_default_tl { 0.116 em plus 0.5
267     \l_xephm_fnt_ziba_tl       { \tl_set:Nn \l_xephm_skip_default_tl { 0.119 em plus 0.5
268     \l_xephm_fnt_default_tl    { \tl_set:Nn \l_xephm_skip_default_tl { 0.14 em plus 0.5
269     \l_xephm_fnt_noskip_tl     { \tl_set:Nn \l_xephm_skip_default_tl { 0 em plus 0.5
270   } { } { \tl_set:Nn \l_xephm_skip_default_tl { #1 } }
271 } ,
272 kashidastretch .default:n = \tl_set:Nn \l_xephm_skip_default_tl { 0 em plus 0.5 em } ,
273
274 ligatures .code:n =
275 {
276   \tl_set:Nn \l_tmpa_tl { #1 }
277   \tl_case:NnTF \l_tmpa_tl

```

```

278     {
279     \l_xephm_lig_aalt_tl   { \tl_set:Nx \l_xephm_active_ligs_tl { \l_xephm_lig_aalt_tl } }
280     \l_xephm_lig_ccmp_tl  { \tl_set:Nx \l_xephm_active_ligs_tl { \l_xephm_lig_ccmp_tl } }
281     \l_xephm_lig_dlig_tl  { \tl_set:Nx \l_xephm_active_ligs_tl { \l_xephm_lig_dlig_tl } }
282     \l_xephm_lig_fina_tl  { \tl_set:Nx \l_xephm_active_ligs_tl { \l_xephm_lig_fina_tl } }
283     \l_xephm_lig_init_tl  { \tl_set:Nx \l_xephm_active_ligs_tl { \l_xephm_lig_init_tl } }
284     \l_xephm_lig_locl_tl  { \tl_set:Nx \l_xephm_active_ligs_tl { \l_xephm_lig_locl_tl } }
285     \l_xephm_lig_medi_tl  { \tl_set:Nx \l_xephm_active_ligs_tl { \l_xephm_lig_medi_tl } }
286     \l_xephm_lig_rlig_tl  { \tl_set:Nx \l_xephm_active_ligs_tl { \l_xephm_lig_rlig_tl } }
287     \l_xephm_lig_default_tl { \tl_set:Nx \l_xephm_active_ligs_tl { \l_xephm_lig_default_tl } }
288     } { } { \tl_set:Nn \l_xephm_active_ligs_tl { #1 } }
289     \bool_set_true:N \l_xephm_ligature_bool
290   } ,
291   ligatures .default:n = \tl_set:Nn \l_xephm_active_ligs_tl { \l_xephm_lig_default_tl } ,
292 }
293
294 \ProcessKeysOptions { xepersian-hm }
295
296 \bool_if:NT \l_xephm_kashida_hm_fix_bool
297 {
298   \tex_input:D { kashida-xepersian-hm.def }
299 }
300
301 \ExplSyntaxOff
302 \endinput
303 </xepersian-hm-sty>

```

1.2 File: kashida-xepersian-hm.def

```

304 <(*kashida-xepersian-hm-def)
305 \ExplSyntaxOn
306 \ProvidesExplFile {kashida-xepersian-hm.def} {2020-08-05} {1.0d} { Fixes-implementation-of-Kashida-~
307
308 \newXeTeXintercharclass \c_xephm_d_charclass % dual-joiner class
309 \newXeTeXintercharclass \c_xephm_l_charclass % lam
310 \newXeTeXintercharclass \c_xephm_r_charclass % right-joiner
311 \newXeTeXintercharclass \c_xephm_a_charclass % alef
312 \newXeTeXintercharclass \c_xephm_y_charclass % yeh
313
314 \bool_if:NT \l_xephm_ligature_bool
315 {
316   \clist_new:N \l_xephm_ligatures_clist
317   \int_new:N \l_xephm_lig_names_len_int
318   \int_set:Nn \l_xephm_lig_names_len_int { \clist_count:N \l_xephm_lig_names_clist }
319   \int_step_inline:nnnn { 1 } { 2 } { \l_xephm_lig_names_len_int }
320   {
321     \int_set:Nn \l_tmpa_int { #1 }
322     \int_set:Nn \l_tmpb_int { \int_eval:n { \l_tmpa_int + 1 } }
323     \tl_set:Nf \l_tmpa_tl { \clist_item:Nn \l_xephm_lig_names_clist { \l_tmpa_int } }
324     \clist_set:Nx \l_tmpa_clist { { \clist_item:Nn \l_xephm_lig_names_clist { \l_tmpb_int } } }

```

```

325     \bool_if:nT { \tl_if_eq_p:NN \l_xephm_active_ligs_tl \l_tmpa_tl || \tl_if_eq_p:NN \l_xephm_
326         {
327             \clist_put_left:Nx \l_xephm_ligatures_clist { \l_tmpa_clist }
328         }
329     }
330 \clist_map_inline:Nn \l_xephm_ligatures_clist
331     {
332         \seq_set_split:Nnn \l_tmpa_seq { = } { #1 }
333         \seq_pop_left:NN \l_tmpa_seq \l_tmpa_tl { } { }
334         \seq_pop_left:NN \l_tmpa_seq \l_tmpb_tl { } { }
335         \tl_const:cx { \tl_use:N \l_tmpb_tl } { \char" \l_tmpa_tl \ }
336     }
337 }
338
339 \bool_if:NT \l_xephm_linebreakpenalty_bool
340 {
341     %% Partly adapted from LaTeX2e source
342     \cs_new:Nn \xephm_line_break: {
343         \if_mode_vertical:
344             \GenericError{
345                 \space\space\space\space\space\space\space\space\space\space\space\space\space\space\space
346             }{
347                 LaTeX Error: There's no line here to end
348             }{
349                 See the LaTeX manual or LaTeX Companion for explanation.
350             }{
351                 Your command was ignored.\MessageBreak
352                 Type \space I <command> <return> \space to replace it-
353                 with another command,\MessageBreak
354                 or \space <return> \space to continue without it.}
355         \else:
356             \l_tmpa_skip \tex_lastskip:D
357             \tex_unskip:D
358             \tex_penalty:D -\l_xephm_line_break_penalty_int
359             \dim_compare:nT { \l_tmpa_skip > \c_zero_skip }
360                 { \skip_horizontal:N \l_tmpa_skip \tex_ignorespaces:D }
361         \fi:
362     }
363
364 \NewDocumentCommand { \discouragebadlinebreaks } { 0{\l_xephm_line_break_penalty_int} 0{\c_xephm_s
365     {
366         \IfNoValueF {#1}
367             { \int_set:Nn \l_xephm_line_break_penalty_int {#1} }
368         \IfNoValueF {#2}
369             { \tl_set:Nn \l_xephm_skip_default_tl {#2} }
370         \xephm_put_line_breaks:n { #3 }
371     }
372
373 \cs_new_protected:Nn \xephm_put_line_breaks:n
374     {

```



```

375     \tl_set:Nn \l_xephm_line_break_tl { #1 }
376     \regex_replace_all:nnN { +([                ]) { { \ \0 \ \c{xephm_line_break:}\ } \l_xephm
377     \tl_use:N \l_xephm_line_break_tl
378   }
379 }
380
381 %% Partly adapted from the code provided by David Carlisle in:
382 %% https://tex.stackexchange.com/questions/356709/how-to-know-the-width-and-fill-the-glue-space-between
383 \cs_new:Npn \xephm_kashida_glyph #1
384 {
385   \bool_if:NT \l_xephm_kashida_hm_fix_bool
386   {
387     \c_xephm_lrm_int\text_penalty:D 10000
388     \mode_leave_vertical:
389     \tex_global:D \tex_advance:D \l_xephm_counter_int \c_one_int
390
391     \tl_set:Nx \l_xephm_pos_tl { p\text_romannumeral:D \l_xephm_counter_int }
392     \tl_set:Nx \l_xephm_zref_tl { z\text_romannumeral:D \l_xephm_counter_int }
393
394     \zsaveposx{x_i_\l_xephm_zref_tl}
395     \tl_set:Nx \l_tmpa_tl
396     {
397       \iow_now:cx { @auxout }
398       {
399         \token_to_str:N \gdef \exp_after:wN \token_to_str:N \cs:w xi\l_xephm_pos_tl \cs_end: { \zpe
400       }
401     }
402     \l_tmpa_tl
403     \skip_horizontal:n { #1 }
404     \zsaveposx{x_f_\l_xephm_zref_tl}
405     \tl_set:Nx \l_tmpa_tl
406     {
407       \iow_now:cx { @auxout }
408       {
409         \token_to_str:N \gdef \exp_after:wN \token_to_str:N \cs:w xf\l_xephm_pos_tl \cs_end: { \zpe
410       }
411     }
412     \l_tmpa_tl
413     \exp_after:wN
414     \if_meaning:w
415     \cs:w xi\l_xephm_pos_tl \cs_end: \text_relax:D
416     \else:
417     \dim_set:Nn \l_xephm_diff_pos_dim
418     {
419       \dim_eval:n { \cs:w xi\l_xephm_pos_tl \cs_end: sp - \cs:w xf\l_xephm_pos_tl \cs_end: sp }
420     }
421     \dim_compare:nTF { \l_xephm_diff_pos_dim == 0sp }
422     { }
423     { \llap { \resizebox { \l_xephm_diff_pos_dim \text_relax:D } { \height } { \c_xephm_ksh_int }
424     \fi:

```

```

425 }
426 }
427
428 \cs_new:Npn \xephm_kashida_leaders #1
429 {
430   \bool_if:NT \l_xephm_kashida_hm_fix_bool
431   {
432     \tl_if_eq:NNTF \l_xephm_gap_filler_tl \l_xephm_stretch_leaders_glyph_tl
433     {
434       \tl_set:Nx \l_xephm_font_full_tl { \tex_fontname:D \tex_the:D \tex_font:D }
435       \tl_set:Nx \l_xephm_font_name_tl { \l_xephm_font_full_tl }
436       \tl_set:Nx \l_xephm_font_init_tl { \l_xephm_font_name_tl }
437       \regex_replace_once:nnN { ^"\[?(HM)[\ \ ](X|F).* } { \1\2 } \l_xephm_font_init_tl\relax
438       \tl_set:Nn \l_tmpa_tl { HMF }
439       \tl_set:Nn \l_tmpb_tl { HMX }
440       \bool_if:nTF { \str_if_eq_p:NN { \l_xephm_font_init_tl } { \l_tmpa_tl } || \str_if_eq_p:NN
441         {
442           \hbox_set:Nn \l_xephm_ksh_box { \XeTeXglyph\XeTeXglyphindex"kashida" }
443           \c_xephm_zwj_int \tex_penalty:D 10000
444           \tex_leaders:D \copy\l_xephm_ksh_box \skip_horizontal:n { #1 }
445           \c_xephm_zwj_int
446         }
447         {
448           \msg_error:nx { xepersian-hm } { hm-series-font-not-found } { \l_xephm_font_name_tl }
449         }
450       }
451     }
452     %% Partly adapted from the code provided by Jonathan Kew in:
453     %% https://tug.org/pipermail/xetex/2009-February/012307.html.
454     %% Somebody notified me that the code in 'kashida-xepersian.def' from xepersian
455     %% package is an exact copy of Jonathan Kew's code. Being unaware of this, in
456     %% the earlier versions of this package I made a mistake and acknowledged
457     %% Vafa Khalighi instead of Jonathan Kew. A sincere thank you to Jonathan Kew
458     %% for his excellent code.
459     \c_xephm_lrm_int\c_xephm_zwj_int\tex_penalty:D 10000
460     \tex_leaders:D \tex_hrulerule:D height \XeTeXglyphbounds \c_xephm_two_int
461     \int_use:N \XeTeXcharglyph \c_xephm_ksh_int depth \XeTeXglyphbounds \c_xephm_four_int
462     \int_use:N \XeTeXcharglyph \c_xephm_ksh_int \skip_horizontal:n { #1 }
463     \c_xephm_zwj_int
464   }
465 }
466 }
467
468 \XeTeXinterchartokenstate = 1
469
470 \clist_set:Nn \l_xephm_a_clist { 0622,0623,0625,0627 } %
471 \clist_map_inline:Nn \l_xephm_a_clist
472 {
473   \XeTeXcharclass "#1 \c_xephm_a_charclass
474 }

```

```

475
476 \clist_set:Nn \l_xephm_d_clist { 0626,0628,062A,062B,062C,062D,062E,0633,0634,0635,0636,0637,0638,0639 }
477 \clist_map_inline:Nn \l_xephm_d_clist
478 {
479   \XeTeXcharclass "#1 \c_xephm_d_charclass
480 }
481
482 \clist_set:Nn \l_xephm_l_clist { 0644 } %
483 \clist_map_inline:Nn \l_xephm_l_clist
484 {
485   \XeTeXcharclass "#1 \c_xephm_l_charclass
486 }
487
488 \clist_set:Nn \l_xephm_r_clist { 0624,0629,062F,0630,0631,0632,0648,0698 } % , , , , , ,
489 \clist_map_inline:Nn \l_xephm_r_clist
490 {
491   \XeTeXcharclass "#1 \c_xephm_r_charclass
492 }
493
494 \clist_set:Nn \l_xephm_y_clist { 0649,064A,06CC } % , ,
495 \clist_map_inline:Nn \l_xephm_y_clist
496 {
497   \XeTeXcharclass "#1 \c_xephm_y_charclass
498 }
499
500 \tl_if_eq:NNTF \l_xephm_gap_filler_tl \l_xephm_stretch_glyph_tl {
501   \XeTeXinterchartoks \c_xephm_y_charclass \c_xephm_y_charclass = {
502     \bool_if:NTF \l_xephm_kashida_hm_fix_bool
503     { \c_xephm_zwj_int \xephm_kashida_glyph \l_xephm_skip_default_tl \c_xephm_zwj_int }
504     { \c_xephm_zwj_int \xephm_kashida_glyph \c_xephm_skip_a_tl \c_xephm_zwj_int }
505   }
506   \XeTeXinterchartoks \c_xephm_d_charclass \c_xephm_y_charclass = {
507     \bool_if:NTF \l_xephm_kashida_hm_fix_bool
508     { \c_xephm_zwj_int \xephm_kashida_glyph \l_xephm_skip_default_tl \c_xephm_zwj_int }
509     { \c_xephm_zwj_int \xephm_kashida_glyph \c_xephm_skip_a_tl \c_xephm_zwj_int }
510   }
511   \XeTeXinterchartoks \c_xephm_y_charclass \c_xephm_d_charclass = { \c_xephm_zwj_int \xephm_kashida_glyph }
512   \XeTeXinterchartoks \c_xephm_d_charclass \c_xephm_d_charclass = { \c_xephm_zwj_int \xephm_kashida_glyph }
513   \XeTeXinterchartoks \c_xephm_l_charclass \c_xephm_d_charclass = { \c_xephm_zwj_int \xephm_kashida_glyph }
514   \XeTeXinterchartoks \c_xephm_d_charclass \c_xephm_l_charclass = { \c_xephm_zwj_int \xephm_kashida_glyph }
515   \XeTeXinterchartoks \c_xephm_l_charclass \c_xephm_l_charclass = { \c_xephm_zwj_int \xephm_kashida_glyph }
516   \XeTeXinterchartoks \c_xephm_d_charclass \c_xephm_r_charclass = { \c_xephm_zwj_int \xephm_kashida_glyph }
517   \XeTeXinterchartoks \c_xephm_d_charclass \c_xephm_a_charclass = { \c_xephm_zwj_int \xephm_kashida_glyph }
518   \XeTeXinterchartoks \c_xephm_l_charclass \c_xephm_r_charclass = { \c_xephm_zwj_int \xephm_kashida_glyph }
519   \XeTeXinterchartoks \c_xephm_l_charclass \c_xephm_a_charclass = { }
520 }
521 {
522   \bool_if:NTF {
523     \tl_if_eq_p:NN \l_xephm_gap_filler_tl \l_xephm_stretch_leaders_glyph_tl ||
524     \tl_if_eq_p:NN \l_xephm_gap_filler_tl \l_xephm_stretch_leaders_hrule_tl

```

```

525 }
526 {
527 \XeTeXinterchartoks \c_xephm_y_charclass \c_xephm_y_charclass = {
528 \bool_if:NTF \l_xephm_kashida_hm_fix_bool
529 { \xephm_kashida_leaders \l_xephm_skip_default_tl }
530 { \xephm_kashida_leaders \c_xephm_skip_a_tl }
531 }
532 \XeTeXinterchartoks \c_xephm_d_charclass \c_xephm_y_charclass = {
533 \bool_if:NTF \l_xephm_kashida_hm_fix_bool
534 { \xephm_kashida_leaders \l_xephm_skip_default_tl }
535 { \xephm_kashida_leaders \c_xephm_skip_a_tl }
536 }
537 \XeTeXinterchartoks \c_xephm_y_charclass \c_xephm_d_charclass = { \xephm_kashida_leaders \c_xephm_skip_a_tl }
538 \XeTeXinterchartoks \c_xephm_d_charclass \c_xephm_d_charclass = { \xephm_kashida_leaders \c_xephm_skip_a_tl }
539 \XeTeXinterchartoks \c_xephm_l_charclass \c_xephm_d_charclass = { \xephm_kashida_leaders \c_xephm_skip_a_tl }
540 \XeTeXinterchartoks \c_xephm_d_charclass \c_xephm_l_charclass = { \xephm_kashida_leaders \c_xephm_skip_a_tl }
541 \XeTeXinterchartoks \c_xephm_l_charclass \c_xephm_l_charclass = { \xephm_kashida_leaders \c_xephm_skip_a_tl }
542 \XeTeXinterchartoks \c_xephm_d_charclass \c_xephm_r_charclass = { \xephm_kashida_leaders \c_xephm_skip_a_tl }
543 \XeTeXinterchartoks \c_xephm_d_charclass \c_xephm_a_charclass = { \xephm_kashida_leaders \c_xephm_skip_a_tl }
544 \XeTeXinterchartoks \c_xephm_l_charclass \c_xephm_r_charclass = { \xephm_kashida_leaders \c_xephm_skip_a_tl }
545 \XeTeXinterchartoks \c_xephm_l_charclass \c_xephm_a_charclass = { }
546 }
547 {
548 \msg_error:nx { xepersian-hm } { error-value-not-available-for-kashida-option } { \l_xephm_gap_tl }
549 }
550 }
551
552 \NewDocumentCommand \KashidaHMFixOn {} { \bool_set_true:N \l_xephm_kashida_hm_fix_bool }
553 \NewDocumentCommand \KashidaHMFixOff {} { \bool_set_false:N \l_xephm_kashida_hm_fix_bool }
554
555 \tex_let:D \KashidaOn \KashidaHMFixOn
556 \tex_let:D \KashidaOff \KashidaHMFixOff
557
558 \bool_if:NTF \l_xephm_kashida_hm_fix_bool
559 {
560 \tl_if_empty:NT \l_xephm_skip_default_tl { \tl_set:Nn \l_xephm_skip_default_tl { 0.14 em plus 0 }
561 }
562 {
563 \tl_set:NV \l_xephm_skip_default_tl \c_xephm_skip_a_tl
564 }
565
566 \ExplSyntaxOff
567
568 \makeatletter
569 \newif\if@Kashida@on
570 %% Becuase Vafa Khalighi has copied the above code (injecting the character uni+200E) in xepersian-2
571 %% (https://tug.org/svn/texlive/trunk/Master/texmf-dist/tex/xelatex/xepersian/kashida-xepersian.def?)
572 %% the following line of code is not needed in xepersian anymore.
573 %% % \newif\if@Kashida@XB@fix
574 \makeatother

```

```
575
576 \endinput
577 </kashida-xepersian-hm-def>
```

Acknowledgements

Todo

Change History

Todo

References

Todo

Index

Todo