

# adfsymbols

Clea F. Rees\*

v1.4 (SVN Rev: 10985) 2025/03/31

## Abstract

Hirwen Harendal, Arkandis Digital Foundry (ADF) has produced Symbols ADF. This guide outlines the  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  support provided with version 1.001 of the fonts in postscript type 1 format.

## 1 Introduction

This document explains how to use the  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  support included with version 1.001 of the Symbols ADF font collection in postscript type 1 format. The fonts were developed by Hirwen Harendal of the Arkandis Digital Foundry (ADF), and information about the fonts themselves, together with copies of the fonts in opentype format, can be found at <http://pagesperso-orange.fr/arkandis/ADF/tugfonts.htm>. The fonts are released under the GPL. For details, see README, NOTICE and COPYING.

The  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  support package consists of all files listed in manifest.txt and these files are released under the  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  Project Public Licence as explained in the included licensing notices and README. Please let me know of any problems so that I can solve them if I can. If you can correct the problems and send me the fix, that would be even better. Unlike the fonts themselves, the  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  support is somewhat experimental.

adfsymbols includes a copy of the fonts in type 1 format, documentation and support files for  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  including two  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  package files, `adfarrows.sty` and `adfbullets.sty`.

## 2 The support packages

adfsymbols provides access to the symbols in ArrowsADF and BulletsADF in  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  through two packages, `adfarrows` and `adfbullets`.

### 2.1 adfarrows

`adfarrows` (*pkg.*) `adfarrows` provides access to ArrowsADF. The package supports a single option to

---

\*Bug tracker: [codeberg.org/cfr/nfssext/issues](https://codeberg.org/cfr/nfssext/issues) | Code: [codeberg.org/cfr/nfssext](https://codeberg.org/cfr/nfssext) |  
Mirror: [github.com/cfr42/nfssext](https://github.com/cfr42/nfssext)

scale the fonts.

`scale (opt.) = <scaling factor>`

Scale the font by *<scaling factor>*, which should be a positive integer or simple decimal such as 2 or 1.2. This option is intended for cases where the fonts should be scaled to match other fonts used in the document e.g. for consistency with the size of regular text or superscript markers.

Initially empty, which is equivalent to 1 but more efficient.

`adfarrows` provides the command `\adfarrow{}` which takes a single numerical argument. There are 52 arrows in `ArrowsADF` which can be produced by feeding the relevant number between 1 and 52 to `\adfarrow{}`.

`\adfarrow {<number>}`

Where *<number>* is a positive integer between 1 and 52 inclusive<sup>1</sup>.

1: ↗	14: ↙	27: ↖	40: ↘
2: ↖	15: ←	28: ↗	41: ←
3: ↗	16: ↖	29: ↘	42: ↖
4: ↘	17: ↑	30: ↗	43: ↑
5: ↓	18: ↗	31: ↓	44: ↗
6: ↗	19: →	32: ↘	45: →
7: ←	20: ↘	33: ←	46: ↖
8: ↘	21: ↓	34: ↖	47: ↓
9: ↑	22: ↗	35: ↑	48: ↗
10: ↗	23: ←	36: ↘	49: ←
11: →	24: ↖	37: →	50: ↖
12: ↖	25: ↑	38: ↖	51: ↑
13: ↓	26: ↗	39: ↓	52: ↗

For example, `\adfarrow{5}\adfarrow{9}` produces: ↘↑.

### 2.1.1 Alternative commands

To make things a little more convenient, additional commands are provided to access the various arrows. The effect is to typeset one of the arrows show above but it is not necessary to look up or remember the correct numerical argument.

`\adfhalfarrowright` First, table 1 lists the four commands provided to access the half arrows. In each case, the number of the arrow is given first. This may be used directly with the `\adfarrow{}` command as explained above. The alternative command is given next. This command may be used to typeset the same arrow. For example both `\adfarrow{1}` and `\adfhalfarrowright` produce ∞. Finally, the arrow produced by the two commands is typeset to their right.

The remaining arrows consist of six families each containing eight arrows — one for each of the eight directions of the compass. These may be accessed in two ways, in addition to using `\adfarrow{}`.

<sup>1</sup>The argument 0 will simply typeset a space and should be avoided as using it may interfere with TeX's spacing algorithms. The problem is that TeX will not recognise it as a space and so will treat it instead as a character.

Table 1: Commands for half arrows

No.	Command	No.	Command
1	<code>\adfhalfarrowright</code>	2	<code>\adfhalfarrowleft</code>
27	<code>\adfhalfarrowrightsolid</code>	28	<code>\adfhalfarrowleftsolid</code>

Table 2: Directional commands

Direction	Command	Example usage
north	<code>\adfarrown</code>	<code>\adfarrown1</code> ↑
northeast	<code>\adfarrowne</code>	<code>\adfarrowne2</code> ↗
east	<code>\adfarrowe</code>	<code>\adfarrowe3</code> →
southeast	<code>\adfarrowse</code>	<code>\adfarrowse4</code> ↘
south	<code>\adfarrows</code>	<code>\adfarrows5</code> ↓
southwest	<code>\adfarrowsw</code>	<code>\adfarrowsw6</code> ↙
west	<code>\adfarroww</code>	<code>\adfarroww1</code> ←
northwest	<code>\adfarrownw</code>	<code>\adfarrownw3</code> ↖

`\adfarrown`  $\{\langle number \rangle\}$  First, eight commands are provided (table 2). Each command takes `\adfarrowne` a single numerical argument,  $\langle number \rangle$ , which must be a positive integer in the `\adfarrowe` range 1–6 inclusive. The argument corresponds to one of the six families of arrows. `\adfarrowse` So using the same number with the different commands will typeset arrows from `\adfarrows` the same family pointing in different directions.

`\adfarrowsw` Second, a further command is provided which allows you to specify both the family `\adfarroww` and direction as separate arguments. This is in fact the base command `\adfarrow` `\adfarrownw` again. Above, we used the command with just one argument: `\adfarrow{}`. In effect, we left the optional argument empty: `\adfarrow[]{}.`

`\adfarrow`  $\{\langle number \rangle\} [\langle family \rangle] \{\langle direction \rangle\}$

`\adfarrow` Where  $\langle number \rangle$  may be any positive integer between 1 and 52 (as above),  $\langle family \rangle$  may be any integer between 1 and 6 (table 3) and  $\langle direction \rangle$  may be any of the eight standard compass directions (table 4).  $\langle family \rangle$  may also be the name of the ‘family’ of arrows.  $\langle direction \rangle$  may also be given in an abbreviated form.

When  $\langle family \rangle$  is given, the second argument specifies the arrow’s direction. *Note that you must specify a family if you specify a direction.* If the optional argument is omitted, the command expects the numerical argument corresponding to the arrow you wish to typeset as listed earlier.

The arrow’s direction may be specified in either a long or an abbreviated form.

The different possibilities are illustrated table 5 where each row consists of a selection of equivalent commands which may be used to produce identical output in different ways. In each case, the number of the arrow is given first. This may be used directly with the `\adfarrow{}` command as explained above. One of the eight commands from the previous section follows. Two additional uses of `\adfarrow` are given next using the `\adfarrow[family]{direction}` form described in this section. Finally, the arrow each of these commands typesets is displayed to their right.

Table 3: `\adfarrow`: ‘family’ names and numbers for first argument

No.	Name
1	opentail
2	plain
3	comic
4	solidtail
5	thick
6	tail

Table 4: `\adfarrow`: direction names for second argument

Direction	Name & abbreviation	
north	north	n
northeast	northeast	ne
east	east	e
southeast	southeast	se
south	south	s
southwest	southwest	sw
west	west	w
northwest	northwest	nw

Table 5: `\adfarrow`: examples

No.	Commands equivalent to <code>\adfarrow{⟨no.⟩}</code>			Result
4	<code>\adfarrowse1</code>	<code>\adfarrow[1]{southeast}</code>	<code>\adfarrow[opentail]{se}</code>	↘
51	<code>\adfarrown6</code>	<code>\adfarrow[tail]{north}</code>	<code>\adfarrow[6]{n}</code>	↖
42	<code>\adfarrownw5</code>	<code>\adfarrow[thick]{nw}</code>	<code>\adfarrow[5]{northwest}</code>	↙
15	<code>\adfarroww2</code>	<code>\adfarrow[2]{w}</code>	<code>\adfarrow[plain]{west}</code>	←
31	<code>\adfarrows4</code>	<code>\adfarrow[solidtail]{south}</code>	<code>\adfarrow[4]{s}</code>	↓
22	<code>\adfarrowsw3</code>	<code>\adfarrow[comic]{sw}</code>	<code>\adfarrow[3]{southwest}</code>	↙

## 2.2 adfbullets

`adfbullets` (*pkg.*) `adfbullets` provides access to `BulletsADF`. The package supports a single option to scale the fonts.

`scale` (*opt.*) =  $\langle$ *scaling factor* $\rangle$

Scale the font by  $\langle$ *scaling factor* $\rangle$ , which should be a positive integer or simple decimal such as 2 or 1.2. This option is intended for cases where the fonts should be scaled to match other fonts used in the document e.g. for consistency with the size of regular text or superscript markers.

Initially empty, which is equivalent to 1 but more efficient.

`adfbullets` provides the command `\adfbullet{}` which takes a single numerical argument. There are 52 bullets in `BulletsADF` which can be produced by feeding the relevant number between 1 and 52 to `\adfbullet{}`.

`\adfbullet`  $\{\langle$ *number* $\rangle\}$

Where  $\langle$ *number* $\rangle$  is a positive integer between 1 and 52 inclusive<sup>2</sup>.

1: ❖	14: ❄	27: •	40: ▶
2: ❁	15: ❁	28: •	41: •
3: ❁	16: ❁	29: ■	42: •
4: ❁	17: ❁	30: ◆	43: •
5: ❁	18: ❁	31: ◀	44: •
6: ❁	19: ❁	32: ▶	45: ◦
7: ❁	20: ◦	33: ▲	46: ■
8: ❁	21: ❁	34: ▼	47: ■
9: ❁	22: ❁	35: ◀	48: ❁
10: ❁	23: ❁	36: ▶	49: ◆
11: ❁	24: ❁	37: ◀	50: ◆
12: ❁	25: *	38: ▶	51: ❖
13: ❁	26: ❁	39: ◀	52: ◦

For example, `\adfbullet{17}\adfbullet{19}\adfbullet{23}` produces: ❁❁❁.

## 3 Usage Examples

`enumitem` allows you to easily change the format of lists:

```
\begin{itemize}[label=\adfbullet{25}]
  \item sealing was,
  \item cabbages;
  \item kings.
\end{itemize}
```

\* sealing was,

<sup>2</sup>Again, the argument 0 will simply typeset a space and should be avoided as using it may interfere with TeX's spacing algorithms.

- \* cabbages;
- \* kings.

Refer to the package documentation for further details.

`adfarrows` and `adfbullets` can be used in beamer presentations to produce lists with custom bullet markers; as icons and markers in `pgf` diagrams; with `sectsty`, `titlesec` and/or `fancyhdr` to typeset custom headings, headers and footers. For example, the equivalent of,

```
\pagestyle{fancy}
\fancyhf[ch]{}
\fancyhf[lf]{}
\fancyhf[rf]{}
\fancyhf[lh]{}
\fancyhf[rh]{}
\fancyhf[ch]{%
\itshape adfsymbols\hspace*{1.5em}{\Large\adfbullet{14}}\hspace*{1.5em}\filedate}
\fancyhf[cf]{%
\itshape {\large\adfbullet{39}} \thepage-\ofname-\lastpage %
{\large\adfbullet{40}}}
\renewcommand{\headrulewidth}{0pt}
```

was used to customise this document's headers and footers with `fancyhdr`.

## A Implementation

You do not need to read the remainder of this document in order to install or use the fonts.

### A.1 Encoding

Both `ArrowsADF` and `BulletsADF` use a single encoding. The only reason to reencode the fonts is to ensure consecutive slot numbers, which makes the user interface a bit nicer.

```
1 /SymbolsADFEncoding [
2 /space
3 /A
4 /B
5 /C
6 /D
7 /E
8 /F
9 /G
10 /H
11 /I
12 /J
13 /K
```

14 /L  
15 /M  
16 /N  
17 /O  
18 /P  
19 /Q  
20 /R  
21 /S  
22 /T  
23 /U  
24 /V  
25 /W  
26 /X  
27 /Y  
28 /Z  
29 /a  
30 /b  
31 /c  
32 /d  
33 /e  
34 /f  
35 /g  
36 /h  
37 /i  
38 /j  
39 /k  
40 /l  
41 /m  
42 /n  
43 /o  
44 /p  
45 /q  
46 /r  
47 /s  
48 /t  
49 /u  
50 /v  
51 /w  
52 /x  
53 /y  
54 /z  
55 /.notdef  
56 /.notdef  
57 /.notdef  
58 /.notdef  
59 /.notdef  
60 /.notdef  
61 /.notdef  
62 /.notdef  
63 /.notdef  
64 /.notdef  
65 /.notdef  
66 /.notdef  
67 /.notdef

68 /.notdef  
69 /.notdef  
70 /.notdef  
71 /.notdef  
72 /.notdef  
73 /.notdef  
74 /.notdef  
75 /.notdef  
76 /.notdef  
77 /.notdef  
78 /.notdef  
79 /.notdef  
80 /.notdef  
81 /.notdef  
82 /.notdef  
83 /.notdef  
84 /.notdef  
85 /.notdef  
86 /.notdef  
87 /.notdef  
88 /.notdef  
89 /.notdef  
90 /.notdef  
91 /.notdef  
92 /.notdef  
93 /.notdef  
94 /.notdef  
95 /.notdef  
96 /.notdef  
97 /.notdef  
98 /.notdef  
99 /.notdef  
100 /.notdef  
101 /.notdef  
102 /.notdef  
103 /.notdef  
104 /.notdef  
105 /.notdef  
106 /.notdef  
107 /.notdef  
108 /.notdef  
109 /.notdef  
110 /.notdef  
111 /.notdef  
112 /.notdef  
113 /.notdef  
114 /.notdef  
115 /.notdef  
116 /.notdef  
117 /.notdef  
118 /.notdef  
119 /.notdef  
120 /.notdef  
121 /.notdef



122 /.notdef  
123 /.notdef  
124 /.notdef  
125 /.notdef  
126 /.notdef  
127 /.notdef  
128 /.notdef  
129 /.notdef  
130 /.notdef  
131 /.notdef  
132 /.notdef  
133 /.notdef  
134 /.notdef  
135 /.notdef  
136 /.notdef  
137 /.notdef  
138 /.notdef  
139 /.notdef  
140 /.notdef  
141 /.notdef  
142 /.notdef  
143 /.notdef  
144 /.notdef  
145 /.notdef  
146 /.notdef  
147 /.notdef  
148 /.notdef  
149 /.notdef  
150 /.notdef  
151 /.notdef  
152 /.notdef  
153 /.notdef  
154 /.notdef  
155 /.notdef  
156 /.notdef  
157 /.notdef  
158 /.notdef  
159 /.notdef  
160 /.notdef  
161 /.notdef  
162 /.notdef  
163 /.notdef  
164 /.notdef  
165 /.notdef  
166 /.notdef  
167 /.notdef  
168 /.notdef  
169 /.notdef  
170 /.notdef  
171 /.notdef  
172 /.notdef  
173 /.notdef  
174 /.notdef  
175 /.notdef

176 /.notdef  
177 /.notdef  
178 /.notdef  
179 /.notdef  
180 /.notdef  
181 /.notdef  
182 /.notdef  
183 /.notdef  
184 /.notdef  
185 /.notdef  
186 /.notdef  
187 /.notdef  
188 /.notdef  
189 /.notdef  
190 /.notdef  
191 /.notdef  
192 /.notdef  
193 /.notdef  
194 /.notdef  
195 /.notdef  
196 /.notdef  
197 /.notdef  
198 /.notdef  
199 /.notdef  
200 /.notdef  
201 /.notdef  
202 /.notdef  
203 /.notdef  
204 /.notdef  
205 /.notdef  
206 /.notdef  
207 /.notdef  
208 /.notdef  
209 /.notdef  
210 /.notdef  
211 /.notdef  
212 /.notdef  
213 /.notdef  
214 /.notdef  
215 /.notdef  
216 /.notdef  
217 /.notdef  
218 /.notdef  
219 /.notdef  
220 /.notdef  
221 /.notdef  
222 /.notdef  
223 /.notdef  
224 /.notdef  
225 /.notdef  
226 /.notdef  
227 /.notdef  
228 /.notdef  
229 /.notdef

```
230 /.notdef
231 /.notdef
232 /.notdef
233 /.notdef
234 /.notdef
235 /.notdef
236 /.notdef
237 /.notdef
238 /.notdef
239 /.notdef
240 /.notdef
241 /.notdef
242 /.notdef
243 /.notdef
244 /.notdef
245 /.notdef
246 /.notdef
247 /.notdef
248 /.notdef
249 /.notdef
250 /.notdef
251 /.notdef
252 /.notdef
253 /.notdef
254 /.notdef
255 /.notdef
256 /.notdef
257 /.notdef
258 ] def
```

# adfsymbols: adfarrows

Clea F. Rees\*

v1.4 (SVN Rev: 10985) 2025/03/31

```
259 \NeedsTeXFormat{LaTeX2e}
260 \RequirePackage{svn-prov}
261 \ProvidesPackageSVN[\filebase.sty]{$Id: adfarrows.dtx 10985 2025-03-31 05:35:43Z
  cfrees $}[v1.4 \revinfo ArrowsADF]
262 \DefineFileInfoSVN[adfarrows]
263 \newif\if@adfarrows@digonnew
```

Copied verbatim, excepting format and modulo package/module name from Joseph Wright's `siunitx.sty` under LPPL

```
264 \@ifundefined{ExplLoaderFileDate}{%
265   \IfFileExists{expl3.sty}{%
266     \RequirePackage{expl3}%
267   }{%
268     \@adfarrows@digonnewfalse
269   }%
270 }{\@adfarrows@digonnewtrue}
```

`scale` (*opt.*) `scale` takes a factor by which to scale the fonts. This is empty by default, which is equivalent to 1, but more efficient.

```
271 \if@adfarrows@digonnew
272 \ExplSyntaxOn
273 \keys_define:nn { adfarrows }
274 {
275   scale .tl_set:N = \adfarrows@scale,
276   scale .initial:V = \@empty,
277 }
278 \else
279 \let\adfarrows@scale\@empty
280 \fi
```

Provide `\ProcessKeyOptions`, `\IfFormatAtLeastTF` on older kernels. Joseph Wright: from `siunitx.sty`; <https://chat.stackexchange.com/transcript/message/64327823#64327823>

```
281 %%%%%%%%%%%
282 \providecommand \IfFormatAtLeastTF { \@ifl@t@r \fmtversion }
```

---

\*Bug tracker: [codeberg.org/cfr/nfssect/issues](https://codeberg.org/cfr/nfssect/issues) | Code: [codeberg.org/cfr/nfssect](https://codeberg.org/cfr/nfssect) |  
Mirror: [github.com/cfr42/nfssect](https://github.com/cfr42/nfssect)

```

283 \IfFormatAtLeastTF { 2022-06-01 }
284 {
285   \ProcessKeyOptions [ adfarrows ]
286 }{
287   \RequirePackage { l3keys2e }
288   \ProcessKeysOptions { adfarrows }
289 }
290 %%%%%%%%%%%
291 \ExplSyntaxOff

```

\adfarrows@style

```

292 \DeclareRobustCommand{\adfarrows@style}{%% do NOT break line below!
293   \not@math@alphabet\adfarrows@style\relax
294   \fontencoding{U}\fontfamily{ArrowsADF}\fontseries{m}\fontshape{n}\selectfont
295 }

```

```

296 \ExplSyntaxOn

```

\l\_\_adfarrows\_base\_ot\_int

```

297 \int_new:N \l__adfarrows_base_ot_int
298 \int_set:Nn \l__adfarrows_base_ot_int {1}

```

\l\_\_adfarrows\_base\_p\_int

```

299 \int_new:N \l__adfarrows_base_p_int
300 \int_set:Nn \l__adfarrows_base_p_int {2}

```

\l\_\_adfarrows\_base\_c\_int

```

301 \int_new:N \l__adfarrows_base_c_int
302 \int_set:Nn \l__adfarrows_base_c_int {3}

```

\l\_\_adfarrows\_base\_st\_int

```

303 \int_new:N \l__adfarrows_base_st_int
304 \int_set:Nn \l__adfarrows_base_st_int {4}

```

\l\_\_adfarrows\_base\_th\_int

```

305 \int_new:N \l__adfarrows_base_th_int
306 \int_set:Nn \l__adfarrows_base_th_int {5}

```

\l\_\_adfarrows\_base\_t\_int

```

307 \int_new:N \l__adfarrows_base_t_int
308 \int_set:Nn \l__adfarrows_base_t_int {6}

```

\l\_\_adfarrows\_dir\_e\_int

```

309 \int_new:N \l__adfarrows_dir_e_int
310 \int_set:Nn \l__adfarrows_dir_e_int {0}

```

\l\_\_adfarrows\_dir\_east\_int

```
311 \int_new:N \l__adfarrows_dir_east_int
312 \int_set:Nn \l__adfarrows_dir_east_int {0}
```

\l\_\_adfarrows\_dir\_se\_int

```
313 \int_new:N \l__adfarrows_dir_se_int
314 \int_set:Nn \l__adfarrows_dir_se_int {1}
```

\l\_\_adfarrows\_dir\_southeast\_int

```
315 \int_new:N \l__adfarrows_dir_southeast_int
316 \int_set:Nn \l__adfarrows_dir_southeast_int {1}
```

\l\_\_adfarrows\_dir\_s\_int

```
317 \int_new:N \l__adfarrows_dir_s_int
318 \int_set:Nn \l__adfarrows_dir_s_int {2}
```

\l\_\_adfarrows\_dir\_south\_int

```
319 \int_new:N \l__adfarrows_dir_south_int
320 \int_set:Nn \l__adfarrows_dir_south_int {2}
```

\l\_\_adfarrows\_dir\_sw\_int

```
321 \int_new:N \l__adfarrows_dir_sw_int
322 \int_set:Nn \l__adfarrows_dir_sw_int {3}
```

\l\_\_adfarrows\_dir\_southwest\_int

```
323 \int_new:N \l__adfarrows_dir_southwest_int
324 \int_set:Nn \l__adfarrows_dir_southwest_int {3}
```

\l\_\_adfarrows\_dir\_w\_int

```
325 \int_new:N \l__adfarrows_dir_w_int
326 \int_set:Nn \l__adfarrows_dir_w_int {4}
```

\l\_\_adfarrows\_dir\_west\_int

```
327 \def\adfarrows@west{west}%
328 \int_new:N \l__adfarrows_dir_west_int
329 \int_set:Nn \l__adfarrows_dir_west_int {4}
```

\l\_\_adfarrows\_dir\_nw\_int

```
330 \int_new:N \l__adfarrows_dir_nw_int
331 \int_set:Nn \l__adfarrows_dir_nw_int {5}
```

\l\_\_adfarrows\_dir\_northwest\_int

```
332 \int_new:N \l__adfarrows_dir_northwest_int
333 \int_set:Nn \l__adfarrows_dir_northwest_int {5}
```

`\l__adfarrows_dir_n_int`

```
334 \int_new:N \l__adfarrows_dir_n_int
335 \int_set:Nn \l__adfarrows_dir_n_int {6}
```

`\l__adfarrows_dir_north_int`

```
336 \int_new:N \l__adfarrows_dir_north_int
337 \int_set:Nn \l__adfarrows_dir_north_int {6}
```

`\l__adfarrows_dir_ne_int`

```
338 \int_new:N \l__adfarrows_dir_ne_int
339 \int_set:Nn \l__adfarrows_dir_ne_int {7}
```

`\l__adfarrows_dir_northeast_int`

```
340 \int_new:N \l__adfarrows_dir_northeast_int
341 \int_set:Nn \l__adfarrows_dir_northeast_int {7}
```

`\g__adfarrows_base_int`

```
342 \int_new:N \g__adfarrows_base_int
```

`\g__adfarrows_add_int`

```
343 \int_new:N \g__adfarrows_add_int
```

`\l__adfarrows_base_opentail_int`

```
344 \int_new:N \l__adfarrows_base_opentail_int
345 \int_set:Nn \l__adfarrows_base_opentail_int {3}
```

`\l__adfarrows_base_plain_int`

```
346 \int_new:N \l__adfarrows_base_plain_int
347 \int_set:Nn \l__adfarrows_base_plain_int {11}
```

`\l__adfarrows_base_comic_int`

```
348 \int_new:N \l__adfarrows_base_comic_int
349 \int_set:Nn \l__adfarrows_base_comic_int {19}
```

`\l__adfarrows_base_solidtail_int`

```
350 \int_new:N \l__adfarrows_base_solidtail_int
351 \int_set:Nn \l__adfarrows_base_solidtail_int {29}
```

`\l__adfarrows_base_thick_int`

```
352 \int_new:N \l__adfarrows_base_thick_int
353 \int_set:Nn \l__adfarrows_base_thick_int {37}
```

`\l__adfarrows_base_tail_int`

```
354 \int_new:N \l__adfarrows_base_tail_int
355 \int_set:Nn \l__adfarrows_base_tail_int {45}
```

`\l__adfarrows_arrow_int`

```
356 \int_new:N \l__adfarrows_arrow_int
```

I don't know why somebody would use these fonts with a Unicode engine, but, just in case, map for that as well as pdf $\TeX$ .

Lua $\TeX$  manual page 49.

```
357 \bool_if:nT { \sys_if_engine luatex_p: }
358 {
359   \protected\def\pdfglyphtounicode {\pdfextension glyphtounicode }
360 }
361 \bool_if:nT { \sys_if_engine luatex_p: || \sys_if_engine pdftex_p: }
362 {
```

`\__adfarrows_glyphtounicode_seq` This seems ... insane?

It would be more efficient to just set everything directly, but this is easier to set up and only read once. First, a sequence to hold glyph names.

```
363 \seq_new:N \__adfarrows_glyphtounicode_seq
364 \seq_set_from_clist:Nn \__adfarrows_glyphtounicode_seq
365 {
```

outlines

```
366 A, %% A right arrow top half 21C0
367 B, %% B left arrow top half 21BC
```

outline shaft/tail with solid tip

```
368 C, %% C → 2192
369 D, %% D ↘ 2198
370 E, %% E ↓ 2193
371 F, %% F ↙ 2199
372 G, %% G ← 2190
373 H, %% H ↖ 2196
374 I, %% I ↑ 2191
375 J, %% J ↗ 2197
```

solid in various styles

```
376 K, %% K → 2192
377 L, %% L ↘ 2198
378 M, %% M ↓ 2193
379 N, %% N ↙ 2199
380 O, %% O ← 2190
381 P, %% P ↖ 2196
382 Q, %% Q ↑ 2191
383 R, %% R ↗ 2197
```



```

384 S, %% S → 2192
385 T, %% T ↘ 2198
386 U, %% U ↓ 2193
387 V, %% V ↙ 2199
388 W, %% W ← 2190
389 X, %% X ↖ 2196
390 Y, %% Y ↑ 2191
391 Z, %% Z ↗ 2197
392 a, %% a right arrow top half 21C0
393 b, %% b left arrow top half 21BC
394 c, %% c → 2192
395 d, %% d ↘ 2198
396 e, %% e ↓ 2193
397 f, %% f ↙ 2199
398 g, %% g ← 2190
399 h, %% h ↖ 2196
400 i, %% i ↑ 2191
401 j, %% j ↗ 2197
402 k, %% k → 2192
403 l, %% l ↘ 2198
404 m, %% m ↓ 2193
405 n, %% n ↙ 2199
406 o, %% o ← 2190
407 p, %% p ↖ 2196
408 q, %% q ↑ 2191
409 r, %% r ↗ 2197
410 s, %% s → 2192
411 t, %% t ↘ 2198
412 u, %% u ↓ 2193
413 v, %% v ↙ 2199
414 w, %% w ← 2190
415 x, %% x ↖ 2196
416 y, %% y ↑ 2191
417 z, %% z ↗ 2197
418 }

```

`\l__adfarrows_tounicode_seq` A sequence to hold Unicode targets. These are not incredibly detailed, but hopefully more useful than PUA.

```

419 \seq_new:N \l__adfarrows_tounicode_seq
420 \seq_set_from_clist:Nn \l__adfarrows_tounicode_seq
421 {

```

outlines

```

422 21C0, %% A right arrow top half 21C0
423 21BC, %% B left arrow top half 21BC

```

outline shaft/tail with solid tip

```

424 2192, %% C → 2192
425 2198, %% D ↘ 2198
426 2193, %% E ↓ 2193
427 2199, %% F ↙ 2199
428 2190, %% G ← 2190

```

429 2196, %% H ↖ 2196  
 430 2191, %% I ↑ 2191  
 431 2197, %% J ↗ 2197

solid in various styles

432 2192, %% K → 2192  
 433 2198, %% L ↘ 2198  
 434 2193, %% M ↓ 2193  
 435 2199, %% N ↙ 2199  
 436 2190, %% O ← 2190  
 437 2196, %% P ↖ 2196  
 438 2191, %% Q ↑ 2191  
 439 2197, %% R ↗ 2197  
 440 2192, %% S → 2192  
 441 2198, %% T ↘ 2198  
 442 2193, %% U ↓ 2193  
 443 2199, %% V ↙ 2199  
 444 2190, %% W ← 2190  
 445 2196, %% X ↖ 2196  
 446 2191, %% Y ↑ 2191  
 447 2197, %% Z ↗ 2197  
 448 21C0, %% a right arrow top half 21C0  
 449 21BC, %% b left arrow top half 21BC  
 450 2192, %% c → 2192  
 451 2198, %% d ↘ 2198  
 452 2193, %% e ↓ 2193  
 453 2199, %% f ↙ 2199  
 454 2190, %% g ← 2190  
 455 2196, %% h ↖ 2196  
 456 2191, %% i ↑ 2191  
 457 2197, %% j ↗ 2197  
 458 2192, %% k → 2192  
 459 2198, %% l ↘ 2198  
 460 2193, %% m ↓ 2193  
 461 2199, %% n ↙ 2199  
 462 2190, %% o ← 2190  
 463 2196, %% p ↖ 2196  
 464 2191, %% q ↑ 2191  
 465 2197, %% r ↗ 2197  
 466 2192, %% s → 2192  
 467 2198, %% t ↘ 2198  
 468 2193, %% u ↓ 2193  
 469 2199, %% v ↙ 2199  
 470 2190, %% w ← 2190  
 471 2196, %% x ↖ 2196  
 472 2191, %% y ↑ 2191  
 473 2197, %% z ↗ 2197  
 474 }

\\_adfarrows\_tounicode:nn TFM-specific mapping.

pdfTEX manual page 33.

475 \cs\_new\_nopar:Npn \\_adfarrows\_tounicode:nn #1#2

```
476 {
477   \pdfglyphtounicode { tfm:ArrowsADF/#1 } { #2 }
478 }
```

Generate the actual mappings.

```
479 \seq_map_pairwise_function:NNN \l__adfarrows_glyphtounicode_seq
480   \l__adfarrows_tounicode_seq \__adfarrows_tounicode:n
481 }
```

`\__adfarrows_arrow:n`

```
482 \cs_new_nopar:Nn \__adfarrows_arrow:n
483 {
484   \int_if_exist:cTF { l__adfarrows_base_#1_int }
485   {
486     \int_gset_eq:Nc \g__adfarrows_base_int { l__adfarrows_base_#1_int }
487   }{ % some kind of error check needed here
488     \int_gset:Nn \g__adfarrows_base_int { #1 }
489   }
490   \int_if_exist:cTF { l__adfarrows_dir_#2_int }
491   {
492     \int_gset_eq:Nc \g__adfarrows_add_int { l__adfarrows_dir_#2_int }
493   }{
494     \PackageError{adfarrows}{#2 not a valid direction. Setting east }
495     \int_gzero:N \g__adfarrows_add_int
496   }
497   \int_set:Nn \l__adfarrows_arrow_int { \g__adfarrows_base_int + \g__adfarrows_add_int
498   }
499   \int_compare:nNnTF { \l__adfarrows_arrow_int } < { 53 }
500   { % \int_compare:nNnTF { \l__adfarrows_arrow_int } > { 0 }
501     {
502       \expandafter\adfarrows@style\expandafter\char \int_to_arabic:n {
503         \l__adfarrows_arrow_int
504       }
505       \PackageError{adfarrows}{\textbackslash l__adfarrows_arrow_int must
506         be greater than 0 but is \int_to_arabic:n {\l__adfarrows_arrow_int}}%
507     }{
508       \PackageError{adfarrows}{\textbackslash l__adfarrows_arrow_int must
509         be less than than 53 but is \int_to_arabic:n {\l__adfarrows_arrow_int}}%
510     }
511 }
```

`\__adfarrows_arrow:n`

```
511 \cs_new_nopar:Nn \__adfarrows_arrow:n
512 {
513   \adfarrows@style\char#1
514 }
```

`\adfarrows`

```

515 \NewDocumentCommand \adfarrow { o m }
516 {
517   \group_begin:
518   \IfValueTF { #1 }
519   {
520     \__adfarrows_arrow:nn { #1 } { #2 }
521   }{
522     \__adfarrows_arrow:n { #2 }
523   }
524   \group_end:
525 }

526 \ExplSyntaxOff

```

`\adhalfarrowright`

```
527 \newcommand*{\adhalfarrowright}{\adfarrow{1}}
```

`\adhalfarrowleft`

```
528 \newcommand*{\adhalfarrowleft}{\adfarrow{2}}
```

`\adhalfarrowrightsolid`

```
529 \newcommand*{\adhalfarrowrightsolid}{\adfarrow{27}}
```

`\adhalfarrowleftsolid`

```
530 \newcommand*{\adhalfarrowleftsolid}{\adfarrow{28}}
```

`\adfarrowe`

```

531 \gdef\adfarrowe#1{%
532   \ifcase #1 \relax
533   \or \adfarrow{3}%
534   \or \adfarrow{11}%
535   \or \adfarrow{19}%
536   \or \adfarrow{29}%
537   \or \adfarrow{37}%
538   \or \adfarrow{45}%
539   \fi}

```

`\adfarrowse`

```

540 \gdef\adfarrowse#1{%
541   \ifcase #1 \relax
542   \or \adfarrow{4}%
543   \or \adfarrow{12}%
544   \or \adfarrow{20}%
545   \or \adfarrow{30}%
546   \or \adfarrow{38}%
547   \or \adfarrow{46}%
548   \fi}

```

`\adfarrows`

```

549 \gdef\adfarrows#1{%
550 \ifcase #1 \relax
551 \or \adfarrows{5}%
552 \or \adfarrows{13}%
553 \or \adfarrows{21}%
554 \or \adfarrows{31}%
555 \or \adfarrows{39}%
556 \or \adfarrows{47}%
557 \fi}

```

`\adfarrowsw`

```

558 \gdef\adfarrowsw#1{%
559 \ifcase #1 \relax
560 \or \adfarrowsw{6}%
561 \or \adfarrowsw{14}%
562 \or \adfarrowsw{22}%
563 \or \adfarrowsw{32}%
564 \or \adfarrowsw{40}%
565 \or \adfarrowsw{48}%
566 \fi}

```

`\adfarrows`

```

567 \gdef\adfarrows#1{%
568 \ifcase #1 \relax
569 \or \adfarrows{7}%
570 \or \adfarrows{15}%
571 \or \adfarrows{23}%
572 \or \adfarrows{33}%
573 \or \adfarrows{41}%
574 \or \adfarrows{49}%
575 \fi}

```

`\adfarrowsw`

```

576 \gdef\adfarrowsw#1{%
577 \ifcase #1 \relax
578 \or \adfarrowsw{8}%
579 \or \adfarrowsw{16}%
580 \or \adfarrowsw{24}%
581 \or \adfarrowsw{34}%
582 \or \adfarrowsw{42}%
583 \or \adfarrowsw{50}%
584 \fi}

```

`\adfarrows`

```

585 \gdef\adfarrows#1{%
586 \ifcase #1 \relax
587 \or \adfarrows{9}%
588 \or \adfarrows{17}%

```

```
589 \or \adfarrow{25}%
590 \or \adfarrow{35}%
591 \or \adfarrow{43}%
592 \or \adfarrow{51}%
593 \fi}
```

\adfarrowne

```
594 \gdef\adfarrowne#1{%
595 \ifcase #1 \relax
596 \or \adfarrow{10}%
597 \or \adfarrow{18}%
598 \or \adfarrow{26}%
599 \or \adfarrow{36}%
600 \or \adfarrow{44}%
601 \or \adfarrow{52}%
602 \fi}

603 %% end adfarrows.sty
```

## A.2 Font Definitions

uarrowsadf.fd (*fd.*) Font declarations for ArrowsADF font

```
604 \ProvidesFile{uarrowsadf.fd}[v1.3 2024/10/01 font definitions for U/ArrowsADF.]
```

addaswyd o t1phv.fd (dyddiad y ffeil fd: 2020-03-25)

```
605 \expandafter\ifx\csname adfarrows@scale\endcsname\relax
606 \let\adfarrows@@scale\@empty
607 \else
608 \edef\adfarrows@@scale{s*\csname adfarrows@scale\endcsname}%
609 \fi
610 \DeclareFontFamily{U}{ArrowsADF}{}
611 \DeclareFontShape{U}{ArrowsADF}{m}{n}{
612 <-> \adfarrows@@scale ArrowsADF
613 }{}
614 \DeclareFontShape{U}{ArrowsADF}{m}{sc}{<->ssub * ArrowsADF/m/n}{}
615 \DeclareFontShape{U}{ArrowsADF}{m}{it}{<->ssub * ArrowsADF/m/sc}{}
616 \DeclareFontShape{U}{ArrowsADF}{m}{sl}{<->ssub * ArrowsADF/m/it}{}
617 \DeclareFontShape{U}{ArrowsADF}{m}{si}{<->ssub * ArrowsADF/m/sl}{}
618 \DeclareFontShape{U}{ArrowsADF}{m}{scit}{<->ssub * ArrowsADF/m/si}{}
619 \DeclareFontShape{U}{ArrowsADF}{m}{scsl}{<->ssub * ArrowsADF/m/scit}{}
620 \DeclareFontShape{U}{ArrowsADF}{b}{n}{<->ssub * ArrowsADF/m/scsl}{}
621 \DeclareFontShape{U}{ArrowsADF}{b}{sc}{<->ssub * ArrowsADF/b/n}{}
622 \DeclareFontShape{U}{ArrowsADF}{b}{it}{<->ssub * ArrowsADF/b/sc}{}
623 \DeclareFontShape{U}{ArrowsADF}{b}{sl}{<->ssub * ArrowsADF/b/it}{}
624 \DeclareFontShape{U}{ArrowsADF}{b}{si}{<->ssub * ArrowsADF/b/sl}{}
625 \DeclareFontShape{U}{ArrowsADF}{b}{scit}{<->ssub * ArrowsADF/b/si}{}
626 \DeclareFontShape{U}{ArrowsADF}{b}{scsl}{<->ssub * ArrowsADF/b/scit}{}
627 \DeclareFontShape{U}{ArrowsADF}{bx}{n}{<->ssub * ArrowsADF/b/scsl}{}
628 \DeclareFontShape{U}{ArrowsADF}{bx}{sc}{<->ssub * ArrowsADF/bx/n}{}
629 \DeclareFontShape{U}{ArrowsADF}{bx}{it}{<->ssub * ArrowsADF/bx/sc}{}
630 \DeclareFontShape{U}{ArrowsADF}{bx}{sl}{<->ssub * ArrowsADF/bx/it}{}

```

```
631 \DeclareFontShape{U}{ArrowsADF}{bx}{si}{<->ssub * ArrowsADF/bx/sl}{}
632 \DeclareFontShape{U}{ArrowsADF}{bx}{scit}{<->ssub * ArrowsADF/bx/si}{}
633 \DeclareFontShape{U}{ArrowsADF}{bx}{scsl}{<->ssub * ArrowsADF/bx/scit}{}

634 \DeclareUnicodeCharacter{21C0}{right arrow top half}
635 \DeclareUnicodeCharacter{21BC}{left arrow top half}
636 \DeclareUnicodeCharacter{2192}{\textrightarrow}
637 \DeclareUnicodeCharacter{2198}{\searrow}
638 \DeclareUnicodeCharacter{2193}{\textdownarrow}
639 \DeclareUnicodeCharacter{2199}{\swarrow}
640 \DeclareUnicodeCharacter{2190}{\textleftarrow}
641 \DeclareUnicodeCharacter{2196}{\nwarrow}
642 \DeclareUnicodeCharacter{2191}{\textuparrow}
643 \DeclareUnicodeCharacter{2197}{\nearrow}
```

# adfsymbols: adfbullets

Clea F. Rees\*

v1.4 (SVN Rev: 10985) 2025/03/31

```
644 \NeedsTeXFormat{LaTeX2e}
645 \RequirePackage{svn-prov}
646 \ProvidesPackageSVN[\filebase.sty]{$Id: adfbullets.dtx 10985 2025-03-31 05:35:43Z
    cfrees $}[v1.4 \revinfo]
647 \DefineFileInfoSVN[adfbullets]
648 \newif\if@adfbullets@digonnew
```

Copied verbatim, excepting format and modulo package/module name from Joseph Wright's `siunitx.sty` under LPPL

```
649 \@ifundefined{ExplLoaderFileDate}{%
650   \IfFileExists{expl3.sty}{%
651     \RequirePackage{expl3}%
652   }{%
653     \@adfbullets@digonnewfalse
654   }%
655 }{\@adfbullets@digonnewtrue}
```

`scale` (*opt.*) `scale` takes a factor by which to scale the fonts. This is empty by default, which is equivalent to 1, but more efficient.

```
656 \if@adfbullets@digonnew
657 \ExplSyntaxOn
658 \keys_define:nm { adfbullets }
659 {
660   scale .tl_set:N = \adfbullets@scale,
661   scale .initial:V = \@empty,
662 }
663 \else
664 \let\adfbullets@scale\@empty
665 \fi
```

Provide `\ProcessKeyOptions`, `\IfFormatAtLeastTF` on older kernels. Joseph Wright: from `siunitx.sty`; <https://chat.stackexchange.com/transcript/message/64327823#64327823>

```
666 %%%%%%%%%%%
667 \providecommand \IfFormatAtLeastTF { \@ifl@t@r \fmtversion }
```

---

\*Bug tracker: [codeberg.org/cfr/nfssext/issues](https://codeberg.org/cfr/nfssext/issues) | Code: [codeberg.org/cfr/nfssext](https://codeberg.org/cfr/nfssext) |  
Mirror: [github.com/cfr42/nfssext](https://github.com/cfr42/nfssext)



```

668 \IfFormatAtLeastTF { 2022-06-01 }
669 {
670   \ProcessKeyOptions [ adfbullets ]
671 }{
672   \RequirePackage { l3keys2e }
673   \ProcessKeysOptions { adfbullets }
674 }
675 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
676 \ExplSyntaxOff

```

`\adfbullets@style`

```

677 \DeclareRobustCommand{\adfbullets@style}{%% do NOT break line below!
678   \not@math@alphabet\adfbullets@style\relax
679   \fontencoding{U}\fontfamily{BulletsADF}\fontseries{m}\fontshape{n}\selectfont
680 }

```

I don't know why somebody would use these fonts with a Unicode engine, but, just in case, map for that as well as pdfTeX.

LuaTeX manual page 49.

```

681 \ExplSyntaxOn
682 \bool_if:nT { \sys_if_engine luatex_p: }
683 {
684   \protected\def\pdfglyphtounicode { \pdfextension glyphtounicode }
685 }
686 \bool_if:nT { \sys_if_engine luatex_p: || \sys_if_engine pdftex_p: }
687 {

```

`_adfbullets_glyphtounicode_seq` This seems ... insane?

It would be more efficient to just set everything directly, but this is easier to set up and only read once. First, a sequence to hold glyph names.

```

688 \seq_new:N \l__adfbullets_glyphtounicode_seq
689 \seq_set_from_clist:Nn \l__adfbullets_glyphtounicode_seq
690 {
691   A, %% A
692   B, %% B
693   C, %% C
694   D, %% D
695   E, %% E
696   F, %% F
697   G, %% G balloon 4-pointed asterisk 2724
698   H, %% H
699   I, %% I ☒ 2720 filled
700   J, %% J ☒ 2720 open
701   K, %% K
702   L, %% L
703   M, %% M
704   N, %% N
705   O, %% O
706   P, %% P
707   Q, %% Q

```

```

708 R, %% R
709 S, %% S
710 T, %% T
711 U, %% U
712 V, %% V
713 W, %% W
714 X, %% X
715 Y, %% Y 8-pointed rectilinear star 2737
716 Z, %% Z
717 a, %% a . filled 25CC
718 b, %% b . filled 25CC
719 c, %% c ■ 2B1B
720 d, %% d ◇ 2BC1
721 e, %% e ◁ 2BC7
722 f, %% f ▷ 2BC8
723 g, %% g triangle up 2BC5
724 h, %% h triangle down 2BC6
725 i, %% i arrowhead left top highlighted 2B98
726 j, %% j arrowhead right top highlighted 2B9A
727 k, %% k
728 l, %% l
729 m, %% m arrowhead left top highlighted 2B98 larger/darker
730 n, %% n arrowhead right top highlighted 2B9A larger/darker
731 o, %% o
732 p, %% p ellipse 2B2C
733 q, %% q dot large 25CE
734 r, %% r dot 00B7
735 s, %% s circled bullet 29BF circled bullet
736 t, %% t
737 u, %% u ■ 2BC0
738 v, %% v cusp 2BCC small
739 w, %% w cusp 2BCC med
740 x, %% x cusp 2BCC large
741 y, %% y cusp open 2BCE
742 z, %% z . open 25CB
743 }

```

`\l__adfbullets_tounicode_seq` A sequence to hold Unicode targets. These are not incredibly detailed, but hopefully more useful than none.

```

744 \seq_new:N \l__adfbullets_tounicode_seq
745 \seq_set_from_clist:Nn \l__adfbullets_tounicode_seq
746 {
747 0 , %% A
748 0 , %% B
749 0 , %% C
750 0 , %% D
751 0 , %% E
752 0 , %% F
753 2724 , %% G
754 0 , %% H
755 2720 , %% I
756 2720 , %% J
757 0 , %% K

```

```

758 0 , %% L
759 0 , %% M
760 0 , %% N
761 0 , %% O
762 0 , %% P
763 0 , %% Q
764 0 , %% R
765 0 , %% S
766 0 , %% T
767 0 , %% U
768 0 , %% V
769 0 , %% W
770 0 , %% X
771 2737 , %% Y
772 0 , %% Z
773 25CC , %% a
774 25CC , %% b
775 2B1B , %% c
776 2BC1 , %% d
777 2BC7 , %% e or 25C0 etc.?
778 2BC8 , %% f
779 2BC5 , %% g
780 2BC6 , %% h
781 2B98 , %% i
782 2B9A , %% j
783 0 , %% k
784 0 , %% l
785 2B98 , %% m
786 2B9A , %% n
787 0 , %% o
788 2B2C , %% p
789 25CE , %% q
790 00B7 , %% r
791 29BF , %% s
792 25B0 , %% t
793 2BC0 , %% u
794 2BCC , %% v
795 2BCC , %% w
796 2BCC , %% x
797 2BCE , %% y
798 25CB , %% z
799 }

```

`\_adfbullets_tounicode:nm` TFM-specific mapping.

pdfTeX manual page 33.

```

800 \cs_new_nopar:Npn \_adfbullets_tounicode:nm #1#2
801 {
802   \str_compare:nNnTF { #2 } = { 0 }
803   {

```

Map to bullet if nothing better.

```

804   \pdfglyphtounicode { tfm:BulletsADF/#1 } { 2022 }

```

```

805     } {
806     \pdfglyphtounicode { tfm:BulletsADF/#1 } { #2 }
807     }
808 }

```

Generate the actual mappings.

```

809 \seq_map_pairwise_function:NNN \l__adfbullets_glyph_tounicode_seq
810 \l__adfbullets_tounicode_seq \l__adfbullets_tounicode:nn
811 }
812 \ExplSyntaxOff

```

`\adfbullet`

```

813 \newcommand*\adfbullet[1]{\adfbullets@style\char#1}
814 %% end adfbullets.sty

```

### A.3 Font Definitions

`ubulletsadf.fd` (*fd.*) Font declarations for BulletsADF font

```

815 \ProvidesFile{ubulletsadf.fd}[v1.3 2024/10/01 font definitions for U/BulletsADF.]

```

addaswyd o t1phv.fd (dyddiad y ffeil fd: 2020-03-25)

```

816 \expandafter\ifx\csname adfbullets@scale\endcsname\relax
817 \let\adfbullets@@scale\@empty
818 \else
819 \edef\adfbullets@@scale{s*\csname adfbullets@scale\endcsname}
820 \fi
821 \DeclareFontFamily{U}{BulletsADF}{}
822 \DeclareFontShape{U}{BulletsADF}{m}{n}{
823 <-> \adfbullets@@scale BulletsADF
824 }{}
825 \DeclareFontShape{U}{BulletsADF}{m}{sc}{<->ssub * BulletsADF/m/n}{}
826 \DeclareFontShape{U}{BulletsADF}{m}{it}{<->ssub * BulletsADF/m/sc}{}
827 \DeclareFontShape{U}{BulletsADF}{m}{sl}{<->ssub * BulletsADF/m/it}{}
828 \DeclareFontShape{U}{BulletsADF}{m}{si}{<->ssub * BulletsADF/m/sl}{}
829 \DeclareFontShape{U}{BulletsADF}{m}{scit}{<->ssub * BulletsADF/m/si}{}
830 \DeclareFontShape{U}{BulletsADF}{m}{scsl}{<->ssub * BulletsADF/m/scit}{}
831 \DeclareFontShape{U}{BulletsADF}{b}{n}{<->ssub * BulletsADF/m/scsl}{}
832 \DeclareFontShape{U}{BulletsADF}{b}{sc}{<->ssub * BulletsADF/b/n}{}
833 \DeclareFontShape{U}{BulletsADF}{b}{it}{<->ssub * BulletsADF/b/sc}{}
834 \DeclareFontShape{U}{BulletsADF}{b}{sl}{<->ssub * BulletsADF/b/it}{}
835 \DeclareFontShape{U}{BulletsADF}{b}{si}{<->ssub * BulletsADF/b/sl}{}
836 \DeclareFontShape{U}{BulletsADF}{b}{scit}{<->ssub * BulletsADF/b/si}{}
837 \DeclareFontShape{U}{BulletsADF}{b}{scsl}{<->ssub * BulletsADF/b/scit}{}
838 \DeclareFontShape{U}{BulletsADF}{bx}{n}{<->ssub * BulletsADF/b/scsl}{}
839 \DeclareFontShape{U}{BulletsADF}{bx}{sc}{<->ssub * BulletsADF/bx/n}{}
840 \DeclareFontShape{U}{BulletsADF}{bx}{it}{<->ssub * BulletsADF/bx/sc}{}
841 \DeclareFontShape{U}{BulletsADF}{bx}{sl}{<->ssub * BulletsADF/bx/it}{}
842 \DeclareFontShape{U}{BulletsADF}{bx}{si}{<->ssub * BulletsADF/bx/sl}{}
843 \DeclareFontShape{U}{BulletsADF}{bx}{scit}{<->ssub * BulletsADF/bx/si}{}

```

```

844 \DeclareFontShape{U}{BulletsADF}{bx}{scsl}{<->ssub * BulletsADF/bx/scit}{}
845 \DeclareUnicodeCharacter{2022}{\textbullet}
846 %%%% \DeclareUnicodeCharacter{2724}{balloon 4-pointed asterisk}
847 % \DeclareUnicodeCharacter{2720}{\maltese$}
848 %%%% \DeclareUnicodeCharacter{2737}{8-pointed rectilinear star}
849 % \DeclareUnicodeCharacter{25CC}{\circle{ } filled}
850 \DeclareUnicodeCharacter{2B1B}{\blacksquare$}
851 \DeclareUnicodeCharacter{2BC1}{\diamond$}
852 \DeclareUnicodeCharacter{2BC7}{\triangleleft$}
853 \DeclareUnicodeCharacter{2BC8}{\triangleright$}
854 \DeclareUnicodeCharacter{2BC5}{triangle up}
855 \DeclareUnicodeCharacter{2BC6}{triangle down}
856 \DeclareUnicodeCharacter{2B98}{arrowhead left top highlighted}
857 \DeclareUnicodeCharacter{2B9A}{arrowhead right top highlighted}
858 %% \DeclareUnicodeCharacter{2B2C}{ellipse}
859 \DeclareUnicodeCharacter{25CE}{dot large}
860 \DeclareUnicodeCharacter{00B7}{dot}
861 \DeclareUnicodeCharacter{29BF}{circled bullet}
862 % \DeclareUnicodeCharacter{2BC0}{\blacksquare$}
863 \DeclareUnicodeCharacter{2BCC}{cusp}
864 \DeclareUnicodeCharacter{2BCE}{cusp open}
865 \DeclareUnicodeCharacter{25CB}{\circle{ } open}

```

## Change History

v1.2a	Remove cack-handed dependency on fp. . . . .	13
General: Fix lack of localisation bug. . . . .	Was \adfarrows@fam{<>}{<>}{<>}{<>}. . . . .	16
v1.2b	\adfarrows: Remove pifont dependency. . . . .	19
General: Include both PDF and TFM. . . . .	\adfbullet: Remove pifont dependency. . . . .	28
v1.3	\l__adfarrows_dir_west_int: Try to make west arrows point west. . . . .	14
General: May as well use expl3 here. The alternative would be rewriting the code to use TeX counts, but for symbols like these there does not seem to be much reason to avoid the overhead of expl3. (Certainly almost anything would be an improvement over the current implementation, I suppose.) . . . . .	scale: . . . . .	12, 24
Belated update for (N)NFSS (probably unneeded. Try switching to DTX/INS. . . . .	uarrowsadf.fd: Support for scaling. . . . .	22
Drop dependencies on pifont and fp. . . . .	ubulletsadf.fd: Support for scaling. . . . .	28
	v1.4	
	General: Add /ToUnicode values (adfarrows). . . . .	16
	Add /ToUnicode values (adfbullets). . . . .	25
	v??	
	General: First public release. . . . .	1

## Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

<b>Symbols</b>	
<code>\@adfarrows@digonnewfalse</code> . . . . .	<i>268</i>
<code>\@adfarrows@digonnewtrue</code> . . . . .	<i>270</i>
<code>\@adfbullets@digonnewfalse</code> . . . . .	<i>653</i>
<code>\@adfbullets@digonnewtrue</code> . . . . .	<i>655</i>
<code>\@empty</code> . . . . .	<i>276, 279, 606, 661, 664, 817</i>
<code>\@ifl@t@r</code> . . . . .	<i>282, 667</i>
<code>\@ifundefined</code> . . . . .	<i>264, 649</i>
<code>\__adfarrows_arrow:n</code> . . . . .	<i>511</i>
<code>\__adfarrows_arrow:n</code> . . . . .	<i>511, 522</i>
<code>\__adfarrows_arrow:nn</code> . . . . .	<i>482, 520</i>
<code>\__adfarrows_tounicode:nn</code> . . . . .	<i>475, 480</i>
<code>\__adfbullets_tounicode:nn</code> . . . . .	<i>800, 810</i>
<b>A</b>	
<code>\adfarrows</code> <i>2, 3, 3, 515, 527, 528, 529,</i> <i>530, 533, 534, 535, 536, 537,</i> <i>538, 542, 543, 544, 545, 546,</i> <i>547, 551, 552, 553, 554, 555,</i> <i>556, 560, 561, 562, 563, 564,</i> <i>565, 569, 570, 571, 572, 573,</i> <i>574, 578, 579, 580, 581, 582,</i> <i>583, 587, 588, 589, 590, 591,</i> <i>592, 596, 597, 598, 599, 600, 601</i>	
<code>\adfarrows_e</code> . . . . .	<i>3, 531</i>
<code>\adfarrows_n</code> . . . . .	<i>3, 585</i>
<code>\adfarrows_w</code> . . . . .	<i>3, 594</i>
<code>\adfarrows_nw</code> . . . . .	<i>3, 576</i>
<code>\adfarrows</code> . . . . .	<i>3, 549</i>
<code>adfarrows (pkg.)</code> . . . . .	<i>1</i>
<code>\adfarrows@scale</code> . . . . .	<i>606, 608, 612</i>
<code>\adfarrows@scale</code> . . . . .	<i>275, 279</i>
<code>\adfarrows@style</code> . . . . .	<i>292, 501, 513</i>
<code>\adfarrows@west</code> . . . . .	<i>327</i>
<code>\adfarrows_e</code> . . . . .	<i>3, 540</i>
<code>\adfarrows_w</code> . . . . .	<i>3, 558</i>
<code>\adfarrows_nw</code> . . . . .	<i>3, 567</i>
<code>\adfbullet</code> . . . . .	<i>5, 813</i>
<code>adfbullets (pkg.)</code> . . . . .	<i>5</i>
<code>\adfbullets@scale</code> . . . . .	<i>817, 819, 823</i>
<code>\adfbullets@scale</code> . . . . .	<i>660, 664</i>
<code>\adfbullets@style</code> . . . . .	<i>677, 813</i>
<code>\adfhalfarrowleft</code> . . . . .	<i>2, 528</i>
<code>\adfhalfarrowleftsolid</code> . . . . .	<i>2, 530</i>
<code>\adfhalfarrowright</code> . . . . .	<i>2, 527</i>
<code>\adfhalfarrowrightsolid</code> . . . . .	<i>2, 529</i>
<b>B</b>	
<code>\blacksquare</code> . . . . .	<i>850, 862</i>
<code>\bool_if:nT</code> . . . . .	<i>357, 361, 682, 686</i>
<b>C</b>	
<code>\char</code> . . . . .	<i>501, 513, 813</i>
<code>\circle</code> . . . . .	<i>849, 865</i>
<code>\cs_new_nopar:Nn</code> . . . . .	<i>482, 511</i>
<code>\cs_new_nopar:Npn</code> . . . . .	<i>475, 800</i>
<code>\csname</code> . . . . .	<i>605, 608, 816, 819</i>
<b>D</b>	
<code>\DeclareFontFamily</code> . . . . .	<i>610, 821</i>
<code>\DeclareFontShape</code> . . . . .	<i>611, 614, 615, 616, 617,</i> <i>618, 619, 620, 621, 622, 623,</i> <i>624, 625, 626, 627, 628, 629,</i> <i>630, 631, 632, 633, 822, 825,</i> <i>826, 827, 828, 829, 830, 831,</i> <i>832, 833, 834, 835, 836, 837,</i> <i>838, 839, 840, 841, 842, 843, 844</i>
<code>\DeclareRobustCommand</code> . . . . .	<i>292, 677</i>
<code>\DeclareUnicodeCharacter</code> . . . . .	<i>634, 635, 636, 637, 638, 639,</i> <i>640, 641, 642, 643, 845, 846,</i> <i>847, 848, 849, 850, 851, 852,</i> <i>853, 854, 855, 856, 857, 858,</i> <i>859, 860, 861, 862, 863, 864, 865</i>
<code>\def</code> . . . . .	<i>327, 359, 684</i>
<code>\diamond</code> . . . . .	<i>851</i>
<b>E</b>	
<code>\edef</code> . . . . .	<i>608, 819</i>
<code>\else</code> . . . . .	<i>278, 607, 663, 818</i>
<code>\endcsname</code> . . . . .	<i>605, 608, 816, 819</i>
<code>\expandafter</code> . . . . .	<i>501, 605, 816</i>
<b>F</b>	
<code>\fi</code> . . . . .	<i>280, 539, 548, 557, 566,</i> <i>575, 584, 593, 602, 609, 665, 820</i>
<code>\fmtversion</code> . . . . .	<i>282, 667</i>
font definitions:	
<code>uarrowsadf.fd</code> . . . . .	<i>604</i>
<code>ubulletsadf.fd</code> . . . . .	<i>815</i>
<code>\fontencoding</code> . . . . .	<i>294, 679</i>
<code>\fontfamily</code> . . . . .	<i>294, 679</i>

<code>\fontseries</code> .....	294, 679	<code>\l__adfarrows_base_t_int</code> .....	307
<code>\fontshape</code> .....	294, 679	<code>\l__adfarrows_base_tail_int</code> ...	354
<b>G</b>			
<code>\g__adfarrows_add_int</code> .....		<code>\l__adfarrows_base_th_int</code> .....	305
.....	343, 492, 495, 497	<code>\l__adfarrows_base_thick_int</code> ..	352
<code>\g__adfarrows_base_int</code> .....		<code>\l__adfarrows_dir_e_int</code> .....	309
.....	342, 486, 488, 497	<code>\l__adfarrows_dir_east_int</code> .....	311
<code>\gdef</code> .....	531,	<code>\l__adfarrows_dir_n_int</code> .....	334
540, 549, 558, 567, 576, 585, 594		<code>\l__adfarrows_dir_ne_int</code> .....	338
<code>\group_begin:</code> .....	517	<code>\l__adfarrows_dir_north_int</code> ...	336
<code>\group_end:</code> .....	524	<code>\l__adfarrows_dir_northeast_int</code>	340
<b>I</b>			
<code>\if@adfarrows@digonnew</code> ....	263, 271	<code>\l__adfarrows_dir_northwest_int</code>	332
<code>\if@adfbullets@digonnew</code> ...	648, 656	<code>\l__adfarrows_dir_nw_int</code> .....	330
<code>\ifcase</code> .....	532,	<code>\l__adfarrows_dir_s_int</code> .....	317
541, 550, 559, 568, 577, 586, 595		<code>\l__adfarrows_dir_se_int</code> .....	313
<code>\ifFileExists</code> .....	265, 650	<code>\l__adfarrows_dir_south_int</code> ...	319
<code>\ifFormatAtLeastTF</code> 282, 283, 667, 668		<code>\l__adfarrows_dir_southeast_int</code>	315
<code>\ifValueTF</code> .....	518	<code>\l__adfarrows_dir_southwest_int</code>	323
<code>\ifx</code> .....	605, 816	<code>\l__adfarrows_dir_sw_int</code> .....	321
<code>\int_compare:nNnTF</code> .....	498, 499	<code>\l__adfarrows_dir_w_int</code> .....	325
<code>\int_gset:Nn</code> .....	488	<code>\l__adfarrows_dir_west_int</code> ...	327
<code>\int_gset:eq:Nc</code> .....	486, 492	<code>\l__adfarrows_dir_west_int</code> ...	327
<code>\int_gzero:N</code> .....	495	<code>\l__adfarrows_glyphtounicode_seq</code>	
<code>\int_if_exist:cTF</code> .....	484, 490	.....	363, 479
<code>\int_new:N</code> 297, 299, 301, 303, 305,		<code>\l__adfarrows_tounicode_seq</code> 419, 480	
307, 309, 311, 313, 315, 317, 319,		<code>\l__adfbullets_glyphtounicode_seq</code>	
321, 323, 325, 328, 330, 332,		.....	688, 809
334, 336, 338, 340, 342, 343,		<code>\l__adfbullets_tounicode_seq</code> 744, 810	
344, 346, 348, 350, 352, 354, 356		<code>\let</code> .....	279, 606, 664, 817
<code>\int_set:Nn</code> ... 298, 300, 302, 304,		<b>M</b>	
306, 308, 310, 312, 314, 316,		<code>\maltese</code> .....	847
318, 320, 322, 324, 326, 329,		<b>N</b>	
331, 333, 335, 337, 339, 341,		<code>\nearrow</code> .....	643
345, 347, 349, 351, 353, 355, 497		<code>\newcommand</code> ..	527, 528, 529, 530, 813
<code>\int_to_arabic:n</code> .....	501, 505, 508	<code>\NewDocumentCommand</code> .....	515
<b>K</b>			
<code>\keys_define:mn</code> .....	273, 658	<code>\newif</code> .....	263, 648
<b>L</b>			
<code>\l__adfarrows_arrow_int</code> .....		<code>\not@math@alphabet</code> .....	293, 678
.....	356, 497, 498, 499, 502, 505, 508	<code>\narrow</code> .....	641
<code>\l__adfarrows_base_c_int</code> .....	301	<b>O</b>	
<code>\l__adfarrows_base_comic_int</code> ..	348	options:	
<code>\l__adfarrows_base_opentail_int</code>	344	scale .....	2, 5, 271, 656
<code>\l__adfarrows_base_ot_int</code> .....	297	<code>\or</code> .....	533, 534, 535, 536, 537,
<code>\l__adfarrows_base_p_int</code> .....	299	538, 542, 543, 544, 545, 546,	
<code>\l__adfarrows_base_plain_int</code> ..	346	547, 551, 552, 553, 554, 555,	
<code>\l__adfarrows_base_solidtail_int</code>	350	556, 560, 561, 562, 563, 564,	
<code>\l__adfarrows_base_st_int</code> .....	303	565, 569, 570, 571, 572, 573,	
		574, 578, 579, 580, 581, 582,	
		583, 587, 588, 589, 590, 591,	
		592, 596, 597, 598, 599, 600, 601	

<b>P</b>	
<code>\PackageError</code> . . . . .	494, 505, 508
packages:	
<code>adfarrows</code> . . . . .	1
<code>adfbullets</code> . . . . .	5
<code>\pdfextension</code> . . . . .	359, 684
<code>\pdfglyphtounicode</code> . . . . .	359, 477, 684, 804, 806
<code>\ProcessKeysOptions</code> . . . . .	288, 673
<code>\protected</code> . . . . .	359, 684
<code>\providecommand</code> . . . . .	282, 667
<code>\ProvidesFile</code> . . . . .	604, 815
<b>R</b>	
<code>\relax</code> . . . . .	293, 532, 541, 550, 559, 568, 577, 586, 595, 605, 678, 816
<b>S</b>	
<code>scale (opt.)</code> . . . . .	2, 5, 271, 656
<code>\searrow</code> . . . . .	637
<code>\selectfont</code> . . . . .	294, 679
<code>\seq_map_pairwise_function:NNN</code> . . . . .	479, 809
<code>\seq_new:N</code> . . . . .	363, 419, 688, 744
<code>\seq_set_from_clist:Nn</code> . . . . .	364, 420, 689, 745
<code>\str_compare:nNnTF</code> . . . . .	802
<code>\swarrow</code> . . . . .	639
<code>\sys_if_engine_luatex_p:</code> . . . . .	357, 361, 682, 686
<code>\sys_if_engine_pdftex_p:</code> . . . . .	361, 686
<b>T</b>	
<code>\textbackslash</code> . . . . .	505, 508
<code>\textbullet</code> . . . . .	845
<code>\textdownarrow</code> . . . . .	638
<code>\textleftarrow</code> . . . . .	640
<code>\textrightarrow</code> . . . . .	636
<code>\textuparrow</code> . . . . .	642
<code>\triangleleft</code> . . . . .	852
<code>\triangleright</code> . . . . .	853
<b>U</b>	
<code>uarrwsadf.fd (fd.)</code> . . . . .	604
<code>ubulletsadf.fd (fd.)</code> . . . . .	815