adforn

Clea F. Rees*

v1.4 (SVN Rev: 11696) 2026/02/25

Abstract

Hirwen Harendal, Arkandis Digital Foundry (ADF) has produced Ornements ADF. This guide outlines the TEX/LATEX support provided with version 1.001 of the font in postscript type 1 format.

§1 Introduction

This document explains how to use the TEX/LATEX support included with version 1.001 of Ornements ADF in postscript type 1 format. The font was developed by Hirwen Harendal of the Arkandis Digital Foundry (ADF), and information about the font itself, together with a copy of the font in opentype format, can be found at http://pagesperso-orange.fr/arkandis/ADF/tugfonts.htm. The font is released under the GPL. For details, see README, NOTICE and COPYING.

The TEX/LATEX support package consists of all files listed in manifest.txt and these files are released under the LATEX 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 font itself, the TEX/LATEX support is somewhat experimental.

adforn includes a copy of the font in type 1 format (`OrnementsADF.pfb`, `OrnementsADF.pfm` and `OrnementsADF.afm`), documentation and support files for TEX/LATEX including a LATEX package file, `adforn.sty`.

§2 The support package

`adforn` (*pkg.*) `\usepackage[`⟨*options*⟩`]{`⟨*adforn*⟩`}`

*Bug tracker: codeberg.org/cfr/nfssext/issues | Code: codeberg.org/cfr/nfssext | Mirror: github.com/cfr42/nfssext

adforn provides access to the ornaments and symbols in OrnementsADF via two sets of commands. First, it provides a single command which takes a range of arguments. The different arguments determine which ornament is typeset. Second, it provides a separate command for each ornament. The choice of command determines which ornament is typeset. The two mechanisms are equivalent[1].

The package supports a lonely option to 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.

§2.1 One command; many arguments

adforn provides the command `\adforn{}` which takes a single numerical argument. There are 75 ornaments in the font which can be produced by feeding the relevant number between 1 and 75 to `\adforn{}`[2]:

1: ❮	16: ❀	31: ❧	46: ❀	61: ❀
2: ❯	17: ❀	32: ❀	47: ❀	62: ❀
3: ❁	18: ❀	33: ❀	48: ❀	63: ❀
4: ❁	19: ❀	34: ❀	49: ❀	64: ❀
5: ❁	20: ❀	35: ❀	50: ❀	65: ❀
6: ❁	21: ❀	36: ❀	51: ❀	66: ❀
7: ❁	22: ❀	37: ❀	52: ❀	67: ❀
8: ❁	23: ❀	38: ❀	53: ❀	68: ❀
9: ❁	24: ❀	39: ❀	54: ❀	69: ❀
10: ❁	25: ❀	40: ❀	55: ❀	70: ❀
11: ❁	26: ❀	41: ❀	56: ❀	71: ❀
12: ❁	27: ❀	42: ❮	57: ❀	72: ❯
13: ❧	28: ❀	43: ❯	58: ❀	73: •
14: ❖	29: ❀	44: ❀	59: ❧	74: §
15: ❧	30: ❀	45: ❀	60: ❀	75: §

[1] The only difference is that the first allows you to typeset a space by passing it the argument `0` whereas there is no command to typeset the space in the second set. For all practical purposes, this difference is irrelevant since you should not use such a command to typeset a space in TEX in any case and it is difficult to see why anybody would want to.

[2] As mentioned above, 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.

For example,

```
%  \adforn{21}\quad\adforn{11}\quad\adforn{49}
%
```

produces:

§2.2 Many commands; no arguments

In addition to the numerical interface, a number of additional commands are provided as an alternative means of accessing the various symbols and ornaments. The following list groups them roughly according to kind. In each case, the number of the ornament is given first. This may be used directly with the `\adforn{}` command as explained above. The alternative command is given next. This command may be used to typeset the same ornament. For example both `\adforn{14}` and `\adfdiamond` produce ⋄. Finally, the ornament produced by the two commands is typeset to their right.

no.	**command**		**no.**	**command**
BASIC SYMBOLS & SHAPES				
74	§	`\adfS`	§	
75	¶	`\adfgee`	¶	
14	⋄	`\adfdiamond`	⋄	
71	▫	`\adfsquare`	▫	
73	•	`\adftextbullet`	•	
FANCY ASTERISKS & BULLETS				
3	✵	`\adfast1`	✵	
4	✳	`\adfast2`	✳	
5	✳	`\adfast3`	✳	
6	✵	`\adfast4`	✵	
7	✵	`\adfast5`	✵	
8	✵	`\adfast6`	✵	
9	✵	`\adfast7`	✵	
10	✵	`\adfast8`	✵	
11	✵	`\adfast9`	✵	
12	✵	`\adfast{10}`	✵	
ARROWS & ARROWHEADS				
70	↼	`\adfhalfleftarrow`	↼	
72	⇀	`\adfhalfrightarrow`	⇀	

BASIC SYMBOLS & SHAPES CONT.			
42	◂	\adfleftarrowhead	◂
43	▸	\adfrightarrowhead	▸
1	◁	\adfhalfleftarrowhead	◁
2	▷	\adfhalfrightarrowhead	▷
FLOURISHES			
20		\adfflourishleft	
48		\adfflourishright	
21		\adfflourishleftdouble	
49		\adfflourishrightdouble	
17		\adfopenflourishleft	
45		\adfopenflourishright	
18		\adfclosedflourishleft	
46		\adfclosedflourishright	
22		\adfsingleflourishleft	
50		\adfsingleflourishright	
19		\adfdoubleflourishleft	
47		\adfdoubleflourishright	
26		\adftripleflourishleft	
54		\adftripleflourishright	
23		\adfsharpflourishleft	
51		\adfsharpflourishright	
24		\adfdoublesharpflourishleft	
52		\adfdoublesharpflourishright	
25		\adfsickleflourishleft	
53		\adfsickleflourishright	
16		\adfwavesleft	
44		\adfwavesright	
FLOWERS			
60		\adfflowerleft	
32		\adfflowerright	
LEAVES			
66		\adfleafleft	
38		\adfleafright	
59		\adfsolidleafleft	
31		\adfsolidleafright	
13		\adfhalfleafleft	
15		\adfhalfleafright	
58		\adfoutlineleafleft	
30		\adfoutlineleafright	

BASIC SYMBOLS & SHAPES CONT.			
68		\adfsmallleafleft	
40		\adfsmallleafright	
64		\adfflatleafleft	
36		\adfflatleafright	
57		\adfflatleafoutlineleft	
29		\adfflatleafoutlineright	
65		\adfflatleafsolidleft	
37		\adfflatleafsolidright	
67		\adfdownleafleft	
39		\adfdownleafright	
61		\adfdownhalfleafleft	
33		\adfdownhalfleafright	
55		\adfflatdownhalfleafleft	
27		\adfflatdownhalfleafright	
56		\adfflatdownoutlineleafleft	
28		\adfflatdownoutlineleafright	
35		\adfhangingleafleft	
63		\adfhangingleafright	
69		\adfsmallhangingleafleft	
41		\adfsmallhangingleafright	
62		\adfhangingflatleafleft	
34		\adfhangingflatleafright	

So,

```
\adfflourishleftdouble\quad\adfast9\quad\adfflourishrightdouble
```

will produce the same output as the example code given in the previous section:

§A Implementation

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

§A.1 Package

Simple wrappers.

```
\NeedsTeXFormat{LaTeX2e}
\RequirePackage{svn-prov}
\ProvidesPackageSVN[\filebase.sty]{$Id: adforn.dtx 11696 2026-02-25 05:39:12Z cfrees $}[v1.4 \
\DefineFileInfoSVN[adforn]
\newif\if@adforn@digonnew
```

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

```
\@ifundefined{ExplLoaderFileDate}{%
  \IfFileExists{expl3.sty}{%
    \RequirePackage{expl3}%
  }{%
    \@adforn@digonnewfalse
  }%
}{\@adforn@digonnewtrue}
```

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

```
\if@adforn@digonnew
\ExplSyntaxOn
\keys_define:nn { adforn }
{
  scale .tl_set:N = \adforn@scale,
  scale .initial:V = \@empty,
}
\else
  \let\adforn@scale\@empty
\fi
```

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

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\providecommand \IfFormatAtLeastTF { \@ifl@t@r \fmtversion }
\IfFormatAtLeastTF { 2022-06-01 }
{
  \ProcessKeyOptions [ adforn ]
}{
  \RequirePackage { l3keys2e }
  \ProcessKeysOptions { adforn }
}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\ExplSyntaxOff

\RequirePackage{fixtounicode}
```

\adforn@style

```
35 \DeclareRobustCommand{\adforn@style}{%% do NOT break line below!
36   \not@math@alphabet\adforn@style\relax
37   \fontencoding{U}\fontfamily{OrnementsADF}\fontseries{m}\fontshape{n}\selectfont
38 }
```

\adforn

```
39 \newcommand*\adforn[1]{{\adforn@style\char#1}}
```

alfleftarrowhead

```
40 \newcommand*{\adfhalfleftarrowhead}{\adforn{1}}
```

lfrightarrowhead

```
41 \newcommand*{\adfhalfrightarrowhead}{\adforn{2}}
```

\adfast

```
42 \gdef\adfast#1{%
43  \ifcase #1  \relax
44   \or \adforn{3}%
45   \or \adforn{4}%
46   \or \adforn{5}%
47   \or \adforn{6}%
48   \or \adforn{7}%
49   \or \adforn{8}%
50   \or \adforn{9}%
51   \or \adforn{10}%
52   \or \adforn{11}%
53   \or \adforn{12}%
54  \fi}
```

\adfhalfleafleft

```
55 \newcommand*{\adfhalfleafleft}{\adforn{13}}
```

\adfdiamond

```
56 \newcommand*{\adfdiamond}{\adforn{14}}
```

adfhalfleafright

```
57 \newcommand*{\adfhalfleafright}{\adforn{15}}
```

\adfwavesleft

```
58 \newcommand*{\adfwavesleft}{\adforn{16}}
```

openflourishleft

```
59 \newcommand*{\adfopenflourishleft}{\adforn{17}}
```

osedflourishleft

```
60 \newcommand*{\adfclosedflourishleft}{\adforn{18}}
```

ubleflourishleft

```
61 \newcommand*{\adfdoubleflourishleft}{\adforn{19}}
```

\adfflourishleft

```
62 \newcommand*{\adfflourishleft}{\adforn{20}}
```

ourishleftdouble

```
63 \newcommand*{\adfflourishleftdouble}{\adforn{21}}
```

ngleflourishleft

```
64 \newcommand*{\adfsingleflourishleft}{\adforn{22}}
```

harpflourishleft

```
65 \newcommand*{\adfsharpflourishleft}{\adforn{23}}
```

harpflourishleft

```
66 \newcommand*{\adfdoublesharpflourishleft}{\adforn{24}}
```

ckleflourishleft

```
67 \newcommand*{\adfsickleflourishleft}{\adforn{25}}
```

ipleflourishleft

```
68 \newcommand*{\adftripleflourishleft}{\adforn{26}}
```

ownhalfleafright

```
69 \newcommand*{\adfflatdownhalfleafright}{\adforn{27}}
```

outlineleafright

```
70 \newcommand*{\adfflatdownoutlineleafright}{\adforn{28}}
```

leafoutlineright

```
71 \newcommand*{\adfflatleafoutlineright}{\adforn{29}}
```

outlineleafright

```
72 \newcommand*{\adfoutlineleafright}{\adforn{30}}
```

dfsolidleafright

```
73 \newcommand*{\adfsolidleafright}{\adforn{31}}
```

\adfflowerright

```
74 \newcommand*{\adfflowerright}{\adforn{32}}
```

ownhalfleafright

```
75 \newcommand*{\adfdownhalfleafright}{\adforn{33}}
```

ingflatleafright

```
76 \newcommand*{\adfhangingflatleafright}{\adforn{34}}
```

fhangingleafleft

```
77 \newcommand*{\adfhangingleafleft}{\adforn{35}}
```

adfflatleafright

```
78 \newcommand*{\adfflatleafright}{\adforn{36}}
```

atleafsolidright

```
79 \newcommand*{\adfflatleafsolidright}{\adforn{37}}
```

\adfleafright

```
80 \newcommand*{\adfleafright}{\adforn{38}}
```

adfdownleafright

```
81 \newcommand*{\adfdownleafright}{\adforn{39}}
```

dfsmallleafright

```
82 \newcommand*{\adfsmallleafright}{\adforn{40}}
```

hangingleafright

```
83 \newcommand*{\adfsmallhangingleafright}{\adforn{41}}
```

adfleftarrowhead

```
84 \newcommand*{\adfleftarrowhead}{\adforn{42}}
```

dfrightarrowhead

```
85 \newcommand*{\adfrightarrowhead}{\adforn{43}}
```

\adfwavesright

```
86 \newcommand*{\adfwavesright}{\adforn{44}}
```

penflourishright

```
87 \newcommand*{\adfopenflourishright}{\adforn{45}}
```

sedflourishright

```
88 \newcommand*{\adfclosedflourishright}{\adforn{46}}
```

bleflourishright

```
89 \newcommand*{\adfdoubleflourishright}{\adforn{47}}
```

adfflourishright

```
90 \newcommand*{\adfflourishright}{\adforn{48}}
```

urishrightdouble

```
91 \newcommand*{\adfflourishrightdouble}{\adforn{49}}
```

gleflourishright

```
92 \newcommand*{\adfsingleflourishright}{\adforn{50}}
```

arpflourishright

```
93 \newcommand*{\adfsharpflourishright}{\adforn{51}}
```

arpflourishright

```
94 \newcommand*{\adfdoublesharpflourishright}{\adforn{52}}
```

kleflourishright

```
95 \newcommand*{\adfsickleflourishright}{\adforn{53}}
```

pleflourishright

```
96 \newcommand*{\adftripleflourishright}{\adforn{54}}
```

downhalfleafleft

```
97 \newcommand*{\adfflatdownhalfleafleft}{\adforn{55}}
```

noutlineleafleft

```
98 \newcommand*{\adfflatdownoutlineleafleft}{\adforn{56}}
```

tleafoutlineleft

```
99 \newcommand*{\adfflatleafoutlineleft}{\adforn{57}}
```

foutlineleafleft

```
100 \newcommand*{\adfoutlineleafleft}{\adforn{58}}
```

adfsolidleafleft

```
101 \newcommand*{\adfsolidleafleft}{\adforn{59}}
```

\adfflowerleft

```
102 \newcommand*{\adfflowerleft}{\adforn{60}}
```

downhalfleafleft

```
103 \newcommand*{\adfdownhalfleafleft}{\adforn{61}}
```

gingflatleafleft

```
104 \newcommand*{\adfhangingflatleafleft}{\adforn{62}}
```

hangingleafright

```
105 \newcommand*{\adfhangingleafright}{\adforn{63}}
```

\adfflatleafleft

```
106 \newcommand*{\adfflatleafleft}{\adforn{64}}
```

latleafsolidleft

```
107 \newcommand*{\adfflatleafsolidleft}{\adforn{65}}
```

\adfleafleft

```
108 \newcommand*{\adfleafleft}{\adforn{66}}
```

\adfdownleafleft

```
109 \newcommand*{\adfdownleafleft}{\adforn{67}}
```

adfsmallleafleft

```
110 \newcommand*{\adfsmallleafleft}{\adforn{68}}
```

lhangingleafleft

```
111 \newcommand*{\adfsmallhangingleafleft}{\adforn{69}}
```

adfhalfleftarrow

```
112 \newcommand*{\adfhalfleftarrow}{\adforn{70}}
```

\adfsquare

```
113 \newcommand*{\adfsquare}{\adforn{71}}
```

dfhalfrightarrow

```
114 \newcommand*{\adfhalfrightarrow}{\adforn{72}}
```

\adftextbullet

```
115 \newcommand*{\adftextbullet}{\adforn{73}}
116 \newcommand*{\adfS}{\adforn{74}}
```

\adfgee

```
117 \newcommand*{\adfgee}{\adforn{75}}
```

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.

```
118 \ExplSyntaxOn
119 \bool_if:nT { \sys_if_engine_luatex_p: || \sys_if_engine_pdftex_p: }
120 {
```

yphtounicode_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.

```
\seq_new:N \l__adforn_glyphtounicode_seq
\seq_set_from_clist:Nn \l__adforn_glyphtounicode_seq
{
  parenleft, %% parenleft
  parenright, %% parenright
  zero, %% zero
  one, %% one
  two, %% two
  three, %% three
  four, %% four
  five, %% five
  six, %% six
  seven, %% seven
  eight, %% eight
  nine, %% nine
  less, %% less
  equal, %% equal
  greater, %% greater
  A, %% A
  B, %% B
  C, %% C
  D, %% D
  E, %% E
  F, %% F
  G, %% G
  H, %% H
  I, %% I
  J, %% J
  K, %% K
  L, %% L
  M, %% M
  N, %% N
  O, %% O
  P, %% P
  Q, %% Q
  R, %% R
  S, %% S
  T, %% T
  U, %% U
  V, %% V
  W, %% W
  X, %% X
  Y, %% Y
  Z, %% Z
  bracketleft, %% bracketleft
```

```
    bracketright, %% bracketright
    a, %% a
    b, %% b
    c, %% c
    d, %% d
    e, %% e
    f, %% f
    g, %% g
    h, %% h
    i, %% i
    j, %% j
    k, %% k
    l, %% l
    m, %% m
    n, %% n
    o, %% o
    p, %% p
    q, %% q
    r, %% r
    s, %% s
    t, %% t
    u, %% u
    v, %% v
    w, %% w
    x, %% x
    y, %% y
    z, %% z
    braceleft, %% braceleft
    bar, %% bar
    braceright, %% braceright
    bullet, %% bullet
    section, %% section
    paragraph, %% paragraph
  }
```

`rn_tounicode_seq` A sequence to hold Unicode targets. These are, for the most part, simply arbitrary. I frankly have no idea what these should map to. I don't know whether there are corresponding Unicode points. If there are, I don't know where they are or how to find them. If I can find them, I have no idea whether to count most of these are the same symbol or which of several codepoints to choose.

```
\seq_new:N \l__adforn_tounicode_seq
\seq_set_from_clist:Nn \l__adforn_tounicode_seq
{
  2B98 , %% 1 hightlight left arrowhead
  2B9A , %% 2 hightlight right arrowhead
  273F , %% 3 BLACK FLORETTE (Dingbats)
  2748 , %% 4 heavy sparkle
  2747 , %% 5 sparkle
```

```
    274A , %% 6 8 teardrop-spoked propeller asterisk
    274B , %% 7 heavy 8 teardrop-spoked propeller asterisk
    2747 , %% 8 sparkle
    2748 , %% 9 heavy sparkle
    274A , %%10 8 teardrop-spoked propeller asterisk
    273F , %%11 BLACK FLORETTE (Dingbats)
    273F , %%12 BLACK FLORETTE (Dingbats)
    2619 , %%13 REVERSED ROTATED FLORAL HEART BULLET
    2B26 , %%14 White Medium Diamond? or 25C7? or 1F754?
    2767 , %%15 rotated floral heart bullet?
```

These are not characters. No idea what to do with them. Ref.: David Carlisle https://github.com/latex3/tagging-project/issues/1181#issuecomment-3783742565.

```
    0020, %%16 <Not A Character>
    0020, %%17 <Not A Character>
    0020, %%18 <Not A Character>
    0020, %%19 <Not A Character>
    0020, %%20 <Not A Character>
    0020, %%21 <Not A Character>
    0020, %%22 <Not A Character>
    0020, %%23 <Not A Character>
    0020, %%24 <Not A Character>
    0020, %%25 <Not A Character>
    0020, %%26 <Not A Character>
```

Arbitrary, but back to characters, at least.

```
    1F654, %%27 Turned North West Pointing Leaf
    1F654, %%28 Turned North West Pointing Leaf  white
    1F65B, %%29 South East Pointing Vine Leaf
    2767 , %%30 rotated floral heart bullet? white
    2767 , %%31 rotated floral heart bullet? black
    1F660, %%32 NORTH WEST POINTING BUD
    2766 , %%33 FLORAL HEART (Dingbats)
    2767 , %%34 rotated floral heart bullet?
    2766 , %%35 FLORAL HEART (Dingbats)
    2767 , %%36 rotated floral heart bullet?
    2767 , %%37 rotated floral heart bullet?
    2767 , %%38 rotated floral heart bullet?
    2766 , %%39 FLORAL HEART (Dingbats)?
    2767 , %%40 rotated floral heart bullet?
    2767 , %%41 rotated floral heart bullet?
    2B9C , %%42 Black Leftwards Equilateral Arrowhead
    2B9E , %%43 black rightwards equilateral arrowhead
```

Not characters. See above.

```
    0020, %%44 <Not A Character>
    0020, %%45 <Not A Character>
    0020, %%46 <Not A Character>
```

```
    0020, %%47 <Not A Character>
    0020, %%48 <Not A Character>
    0020, %%49 <Not A Character>
    0020, %%50 <Not A Character>
    0020, %%51 <Not A Character>
    0020, %%52 <Not A Character>
    0020, %%53 <Not A Character>
    0020, %%54 <Not A Character>
```

Arbitrary, but back to characters, at least.

```
    1F651, %%55 southwest pointing leaf
    1F651, %%56 southwest pointing leaf
    1F658, %%57 North West Pointing Vine Leaf white
    2619 , %%58 REVERSED ROTATED FLORAL HEART BULLET white
    2619 , %%59 REVERSED ROTATED FLORAL HEART BULLET
    1F662, %%60 NORTH EAST POINTING BUD
    2766 , %%61 FLORAL HEART (Dingbats)
    2619 , %%62 REVERSED ROTATED FLORAL HEART BULLET
    2766 , %%63 FLORAL HEART (Dingbats)
    2619 , %%64 REVERSED ROTATED FLORAL HEART BULLET
    2619 , %%65 REVERSED ROTATED FLORAL HEART BULLET
    2619 , %%66 REVERSED ROTATED FLORAL HEART BULLET
    2619 , %%67 REVERSED ROTATED FLORAL HEART BULLET
    2619 , %%68 REVERSED ROTATED FLORAL HEART BULLET
    2619 , %%69 REVERSED ROTATED FLORAL HEART BULLET
    1F66C, %%70 Leftwards Rocket
    274F , %%71 shadowed square
    1F66E, %%72 Rightwards Rocket
    2022 , %%73 bullet
    00A7 , %%74 section
    2761 , %%75 curved stem paragraph ornament?
}
```

Generate the actual mappings.

```
  \fixtounicode_tounicode:nNN { OrnementsADF } \l__adforn_glyphtounicode_seq
    \l__adforn_tounicode_seq
}
\ExplSyntaxOff

%% end adforn.sty
```

The remaining files are not used directly, but are required to generate the files which allow TEX and LATEX to use the fonts. The sources use `fontinst` as explained in the (sparse) comments. While you can install these files into a TEX tree, they are not required for typesetting.

§A.2 Driver

The file does all the initial setup of the fonts. It organises the fonts into families, defines shapes and reencodes as required.

```
/OrnementsADFEncoding [
/space
/parenleft
/parenright
/zero
/one
/two
/three
/four
/five
/six
/seven
/eight
/nine
/less
/equal
/greater
/A
/B
/C
/D
/E
/F
/G
/H
/I
/J
/K
/L
/M
/N
/O
/P
/Q
/R
/S
/T
/U
/V
/W
/X
/Y
/Z
/bracketleft
```

```
328 /bracketright
329 /a
330 /b
331 /c
332 /d
333 /e
334 /f
335 /g
336 /h
337 /i
338 /j
339 /k
340 /l
341 /m
342 /n
343 /o
344 /p
345 /q
346 /r
347 /s
348 /t
349 /u
350 /v
351 /w
352 /x
353 /y
354 /z
355 /braceleft
356 /bar
357 /braceright
358 /bullet
359 /section
360 /paragraph
361 /.notdef
362 /.notdef
363 /.notdef
364 /.notdef
365 /.notdef
366 /.notdef
367 /.notdef
368 /.notdef
369 /.notdef
370 /.notdef
371 /.notdef
372 /.notdef
373 /.notdef
374 /.notdef
375 /.notdef
376 /.notdef
```

```
377 /.notdef
378 /.notdef
379 /.notdef
380 /.notdef
381 /.notdef
382 /.notdef
383 /.notdef
384 /.notdef
385 /.notdef
386 /.notdef
387 /.notdef
388 /.notdef
389 /.notdef
390 /.notdef
391 /.notdef
392 /.notdef
393 /.notdef
394 /.notdef
395 /.notdef
396 /.notdef
397 /.notdef
398 /.notdef
399 /.notdef
400 /.notdef
401 /.notdef
402 /.notdef
403 /.notdef
404 /.notdef
405 /.notdef
406 /.notdef
407 /.notdef
408 /.notdef
409 /.notdef
410 /.notdef
411 /.notdef
412 /.notdef
413 /.notdef
414 /.notdef
415 /.notdef
416 /.notdef
417 /.notdef
418 /.notdef
419 /.notdef
420 /.notdef
421 /.notdef
422 /.notdef
423 /.notdef
424 /.notdef
425 /.notdef
```

```
426 /.notdef
427 /.notdef
428 /.notdef
429 /.notdef
430 /.notdef
431 /.notdef
432 /.notdef
433 /.notdef
434 /.notdef
435 /.notdef
436 /.notdef
437 /.notdef
438 /.notdef
439 /.notdef
440 /.notdef
441 /.notdef
442 /.notdef
443 /.notdef
444 /.notdef
445 /.notdef
446 /.notdef
447 /.notdef
448 /.notdef
449 /.notdef
450 /.notdef
451 /.notdef
452 /.notdef
453 /.notdef
454 /.notdef
455 /.notdef
456 /.notdef
457 /.notdef
458 /.notdef
459 /.notdef
460 /.notdef
461 /.notdef
462 /.notdef
463 /.notdef
464 /.notdef
465 /.notdef
466 /.notdef
467 /.notdef
468 /.notdef
469 /.notdef
470 /.notdef
471 /.notdef
472 /.notdef
473 /.notdef
474 /.notdef
```

475 /.notdef
476 /.notdef
477 /.notdef
478 /.notdef
479 /.notdef
480 /.notdef
481 /.notdef
482 /.notdef
483 /.notdef
484 /.notdef
485 /.notdef
486 /.notdef
487 /.notdef
488 /.notdef
489 /.notdef
490 /.notdef
491 /.notdef
492 /.notdef
493 /.notdef
494 /.notdef
495 /.notdef
496 /.notdef
497 /.notdef
498 /.notdef
499 /.notdef
500 /.notdef
501 /.notdef
502 /.notdef
503 /.notdef
504 /.notdef
505 /.notdef
506 /.notdef
507 /.notdef
508 /.notdef
509 /.notdef
510 /.notdef
511 /.notdef
512 /.notdef
513 /.notdef
514 /.notdef
515 /.notdef
516 /.notdef
517 /.notdef
518 /.notdef
519 /.notdef
520 /.notdef
521 /.notdef
522 /.notdef
523 /.notdef

```
/.notdef
/.notdef
/.notdef
/.notdef
/.notdef
/.notdef
/.notdef
/.notdef
/.notdef
/.notdef
/.notdef
/.notdef
/.notdef
/.notdef
/.notdef
/.notdef
] def
```

§A.3 Font Definitions

`ementsadf.fd` (*fd.*) Font declarations for OrnementsADF font

```
\ProvidesFile{uornements.fd}[v1.2 2024/09/29 font definitions for U/OrnementsADF.]
```

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

```
  \expandafter\ifx\csname adforn@scale\endcsname\relax
    \let\adforn@@scale\@empty
  \else
    \edef\adforn@@scale{s*[\csname adforn@scale\endcsname]}%
  \fi
\DeclareFontFamily{U}{OrnementsADF}{}
\DeclareFontShape{U}{OrnementsADF}{m}{n}{
  <-> \adforn@@scale OrnementsADF
}{}
\DeclareFontShape{U}{OrnementsADF}{m}{sc}{<->ssub * OrnementsADF/m/n}{}
\DeclareFontShape{U}{OrnementsADF}{m}{it}{<->ssub * OrnementsADF/m/sc}{}
\DeclareFontShape{U}{OrnementsADF}{m}{sl}{<->ssub * OrnementsADF/m/it}{}
\DeclareFontShape{U}{OrnementsADF}{m}{si}{<->ssub * OrnementsADF/m/sl}{}
\DeclareFontShape{U}{OrnementsADF}{m}{scit}{<->ssub * OrnementsADF/m/si}{}
\DeclareFontShape{U}{OrnementsADF}{m}{scsl}{<->ssub * OrnementsADF/m/scit}{}
\DeclareFontShape{U}{OrnementsADF}{b}{n}{<->ssub * OrnementsADF/m/scsl}{}
\DeclareFontShape{U}{OrnementsADF}{b}{sc}{<->ssub * OrnementsADF/b/n}{}
\DeclareFontShape{U}{OrnementsADF}{b}{it}{<->ssub * OrnementsADF/b/sc}{}
\DeclareFontShape{U}{OrnementsADF}{b}{sl}{<->ssub * OrnementsADF/b/it}{}
\DeclareFontShape{U}{OrnementsADF}{b}{si}{<->ssub * OrnementsADF/b/sl}{}
\DeclareFontShape{U}{OrnementsADF}{b}{scit}{<->ssub * OrnementsADF/b/si}{}
\DeclareFontShape{U}{OrnementsADF}{b}{scsl}{<->ssub * OrnementsADF/b/scit}{}
```

```
\DeclareFontShape{U}{OrnementsADF}{bx}{n}{<->ssub * OrnementsADF/b/scsl}{}
\DeclareFontShape{U}{OrnementsADF}{bx}{sc}{<->ssub * OrnementsADF/bx/n}{}
\DeclareFontShape{U}{OrnementsADF}{bx}{it}{<->ssub * OrnementsADF/bx/sc}{}
\DeclareFontShape{U}{OrnementsADF}{bx}{sl}{<->ssub * OrnementsADF/bx/it}{}
\DeclareFontShape{U}{OrnementsADF}{bx}{si}{<->ssub * OrnementsADF/bx/sl}{}
\DeclareFontShape{U}{OrnementsADF}{bx}{scit}{<->ssub * OrnementsADF/bx/si}{}
\DeclareFontShape{U}{OrnementsADF}{bx}{scsl}{<->ssub * OrnementsADF/bx/scit}{}
```

Change History

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.