

adforn<sup>\*</sup>

Clea F. Rees<sup>†</sup>

24<sup>th</sup> September, 2019

### Abstract

Hirwen Harendal, Arkandis Digital Foundry (ADF) has produced Ornements ADF. This guide outlines the  $\text{\TeX}/\text{\LaTeX}$  support provided with version 1.001 of the font in postscript type 1 format.

## §1 Introduction

This document explains how to use the  $\text{\TeX}/\text{\LaTeX}$  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  $\text{\TeX}/\text{\LaTeX}$  support package consists of all files listed in `manifest.txt` and these files are released under the  $\text{\LaTeX}$  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  $\text{\TeX}/\text{\LaTeX}$  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  $\text{\TeX}/\text{\LaTeX}$  including a  $\text{\LaTeX}$  package file, `adforn.sty`.

## §2 The support package

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<sup>1</sup>.

---

<sup>\*</sup>Version 1.1

<sup>†</sup>ReesC21 <at> cardiff <dot> ac <dot> uk

<sup>1</sup>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  $\text{\TeX}$  in any case and it is difficult to see why anybody would want to.

## §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{}`<sup>2</sup>:

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: §

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.

---

BASIC SYMBOLS & SHAPES				
74 \adfs	§	75 \adfgee		§
14 \adfdiamond	◊	71 \adfsquare		□
73 \adfbullet	•			

---

<sup>2</sup>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.

---

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 \adffhalfleftarrow	◀	72 \adffhalfrightarrow	▶	
42 \adffleftarrowhead	◀	43 \adffrightarrowhead	▶	
1 \adffhalfleftarrowhead	◀	2 \adffhalfrightarrowhead	▶	
FLOURISHES				
20 \adffflourishleft	↙	48 \adffflourishright	↘	
21 \adffflourishleftdouble	↙↙	49 \adffflourishrightdouble	↘↘	
17 \adffopenflourishleft	↖	45 \adffopenflourishright	↗	
18 \adffclosedflourishleft	↖↖	46 \adffclosedflourishright	↗↗	
22 \adffsingleflourishleft	↙	50 \adffsingleflourishright	↘	
19 \adffdoubleflourishleft	↙↙	47 \adffdoubleflourishright	↘↘	
26 \adfftripleflourishleft	↙↙↙	54 \adfftripleflourishright	↘↘↘	
23 \adffsharpflourishleft	———	51 \adffsharpflourishright	———	
24 \adffdoublesharpflourishleft	————	52 \adffdoublesharpflourishright	————	
25 \adffsickleflourishleft	↙↙	53 \adffsickleflourishright	↘↘	
16 \adffwavesleft	————	44 \adffwavesright	————	
FLOWERS				
60 \adffflowerleft	✿	32 \adffflowerright	✿	
LEAVES				
66 \adffleafleft	⌚	38 \adffleafright	⌚	
59 \adffsolidleafleft	⌚⌚	31 \adffsolidleafright	⌚⌚	
13 \adffhalfleafleft	⌚⌚	15 \adffhalfleafright	⌚⌚	
58 \adffoutlineleafleft	⌚⌚	30 \adffoutlineleafright	⌚⌚	
68 \adffsmallleafleft	⌚⌚	40 \adffsmallleafright	⌚⌚	
64 \adffflatleafleft	⌚⌚	36 \adffflatleafright	⌚⌚	
57 \adffflatleafoutlineleft	⌚⌚	29 \adffflatleafoutlineright	⌚⌚	
65 \adffflatleafsolidleft	⌚⌚	37 \adffflatleafsolidright	⌚⌚	
67 \adffdownleafleft	⌚⌚	39 \adffdownleafright	⌚⌚	
61 \adffdownhalfleafleft	⌚⌚	33 \adffdownhalfleafright	⌚⌚	
55 \adffflatdownhalfleafleft	⌚⌚	27 \adffflatdownhalfleafright	⌚⌚	
56 \adffflatdownoutlineleafleft	⌚⌚	28 \adffflatdownoutlineleafright	⌚⌚	
35 \adffhangingleafleft	⌚⌚	63 \adffhangingleafright	⌚⌚	
69 \adffsmallhangingleafleft	⌚⌚	41 \adffsmallhangingleafright	⌚⌚	
62 \adffhangingflatleafleft	⌚⌚	34 \adffhangingflatleafright	⌚⌚	

---

So,

```
\adffflourishleftdouble\quad\adffast9\quad\adffflourishrightdouble
```

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

∞ \* ∞

∞ 4 of 4 ∞