The **dingbat** package[*]

Scott Pakin

pakin@uiuc.edu

2001/04/27

Abstract

The dingbat package provides a LaTeX 2_ε interface to the `ark10.mf` and `dingbat.mf` fonts.

1   Usage

`\rightpointright`
`\leftpointright`
`\leftthumbsdown`
`\leftthumbsup`
`\rightpointleft`
`\rightthumbsdown`
`\rightthumbsup`

The dingbat package defines the following hand gestures:

☞	`\rightpointright`	Right hand pointing to the right
☞	`\leftpointright`	Left hand pointing to the right
	`\leftthumbsdown`	Left hand, thumbs down
	`\leftthumbsup`	Left hand, thumbs up
	`\rightpointleft`	Right hand pointing to the left
	`\rightthumbsdown`	Right hand, thumbs down
	`\rightthumbsup`	Right hand, thumbs up

(`\rightpointright` comes from `dingbat.mf`, while the rest come from `ark10.mf`, hence, the discrepency in style.)

`\squarewithdots`
`\filledsquarewithdots`
`\Sborder`
`\Zborder`

The following symbols are intended to be used to create fancy borders around a box or the entire page.

	`\squarewithdots`	Unfilled square with dots
	`\filledsquarewithdots`	Filled square with dots
	`\Sborder`	S-like border design
	`\Zborder`	Z-like border design

Note that if you *are* going to create fancy borders, I recommend using the niceframe package. niceframe is geared to drawing fancy borders and knows specifically about the border characters in `dingbat.mf`. The symbols defined by the dingbat package,

---

[*]This file has version number v1.00, last revised 2001/04/27.

`\squarewithdots`, `\filledsquarewithdots`, `\Sborder`, and `\Zborder`, are the only border characters that might be useful outside of a border (e.g., in a straight line or as a bullet in a fancy itemized list); the other, unnamed, characters defined in `dingbat.mf` represent a frame's corners and sides.

The following `dingbat` symbols don't fit into either of the previous categories:

	`\largepencil`	Large pencil
	`\anchor`	Anchor
	`\carriagereturn`	Carriage return
	`\checkmark`	Check mark
	`\eye`	Eye
	`\satellitedish`	DLA satellite dish
	`\smallpencil`	Small pencil

For some of these symbols, most notably `\largepencil`, the font metrics seem to be a bit off. As a result, the symbols tend to jut into previous or subsequent lines.

2 Implementation

There's nothing particularly interesting about the implementation. Section 2.1 defines the symbol commands that make up `dingbat.sty`, and Section 2.2 creates the `uark.fd` and `udingbat.fd` font definition files, which LaTeX loads when `dingbat.sty` selects font encoding `U` and font family `ark` or `dingbat`.

2.1 Symbol commands

1 ⟨∗package⟩

`\arkfamily` Define a macro, `\arkfamily`, which switches the font to `ark` and another macro,
`\ark@sym` `\ark@sym` which typesets a symbol in the `ark` font.

2 \newcommand{\arkfamily}{\fontencoding{U}\fontfamily{ark}\selectfont}
3 \newcommand{\ark@sym}[1]{{\arkfamily\symbol{#1}}}

`\carriagereturn` There are only ten symbols in the `ark` font. We define names individually for
`\leftthumbsdown` each of them. Note that `\largepencil`'s bounding box is a little too short, so
`\eye` `\largepencil` may jut into the previous line of text. I think some of the other
`\rightpointleft` bounding boxes may be the wrong size, as well, but I haven't yet tested that
`\smallpencil` carefully.
`\leftpointright`
`\leftthumbsup` 4 \newcommand{\carriagereturn}{\ark@sym{'103}}
`\largepencil` 5 \newcommand{\leftthumbsdown}{\ark@sym{'104}}
`\rightthumbsdown` 6 \newcommand{\eye}{\ark@sym{'105}}
`\rightthumbsup` 7 \newcommand{\rightpointleft}{\ark@sym{'114}}
8 \newcommand{\smallpencil}{\ark@sym{'120}}

2

```
 9 \newcommand{\leftpointright}{\ark@sym{'122}}
10 \newcommand{\leftthumbsup}{\ark@sym{'125}}
11 \newcommand{\largepencil}{\ark@sym{'127}}
12 \newcommand{\rightthumbsdown}{\ark@sym{'144}}
13 \newcommand{\rightthumbsup}{\ark@sym{'165}}
```

\dingbatfamily \dingbat@sym Define a macro, \dingbatfamily, which switches the font to dingbat and another macro, \dingbat@sym which typesets a symbol in the dingbat font.

```
14 \newcommand{\dingbatfamily}{\fontencoding{U}\fontfamily{dingbat}\selectfont}
15 \newcommand{\dingbat@sym}[1]{{\dingbatfamily\symbol{#1}}}
```

\checkmark \satellitedish \rightpointright \anchor The following are definitions for all the non-border characters in dingbat.mf.

```
16 \newcommand{\checkmark}{\dingbat@sym{'104}}
17 \newcommand{\satellitedish}{\dingbat@sym{'111}}
18 \newcommand{\rightpointright}{\dingbat@sym{'116}}
19 \newcommand{\anchor}{\dingbat@sym{'117}}
```

\squarewithdots \filledsquarewithdots \Sborder \Zborder The following are the border characters. There are many more, but I haven't yet made up LaTeX names for them.

```
20 \newcommand{\squarewithdots}{\dingbat@sym{'102}}
21 \newcommand{\filledsquarewithdots}{\dingbat@sym{'103}}
22 \newcommand{\Sborder}{\dingbat@sym{'123}}
23 \newcommand{\Zborder}{\dingbat@sym{'132}}
```

```
24 ⟨/package⟩
```

2.2 Font definition files

2.2.1 uark.fd

All we do here is define ark as a symbol font (U encoding) and tell LaTeX to scale ark10.mf to whatever font size is requested.

```
25 ⟨*uarkfd⟩
```

```
26 \DeclareFontFamily{U}{ark}{}
27 \DeclareFontShape{U}{ark}{m}{n}{<-> ark10}{}
```

```
28 ⟨/uarkfd⟩
```

2.2.2 udingbat.fd

Similarly, we define dingbat as another, U-encoded symbol font and instruct LaTeX to scale dingbat.mf to whatever font size is requested.

```
29 ⟨*udingbatfd⟩
```

```
30 \DeclareFontFamily{U}{dingbat}{}
31 \DeclareFontShape{U}{dingbat}{m}{n}{<-> dingbat}{}
```

```
32 ⟨/udingbatfd⟩
```

3   Credits

Just so it's clear who did what, here are the various components of the dingbat package and the associated copyright information:

ark10.mf   Copyright (c) 1988 by Arthur M. Keller
               The ARK font was initially done in MF79 by Scott
               Kim. It was converted to (new) MF by N.N. Billawala
               of Metamarks.
dingbat.dtx   Copyright (C) 2001 Scott Pakin
dingbat.ins   Copyright (C) 2001 Scott Pakin
dingbat.mf   Created 3-3-89 by Doug Henderson
uark.fd   Copyright (C) 2001 Scott Pakin

Index

Numbers written in italic refer to the page where the corresponding entry is described, the ones underlined to the code line of the definition, the rest to the code lines where the entry is used.