The `dndicons` package

A set of high-quality icons made with Ti*k*Z for use in notes for tabletop role-playing games

Jasper Habicht *

Version 1.6.0, released on 15 March 2024

---

1 Introduction

The `dndicons` package provides a set of high-quality icons made with Ti*k*Z for use in notes for tabletop role-playing games. The icons are meant to be used in the body text, but they can also be used in other contexts such as graphics or diagrams.

Since the commands to typeset the icons use `tikzpicture` environments, these commands should not be used inside another `tikzpicture`. However, because the package defines the icons as Ti*k*Z shapes, it is possible to use the icons in `tikzpicture` environments directly.

Apart from that, as of version 1.1.0, the package provides a way to define custom commands to typeset the icons as boxed material which is safe to use in a `tikzpicture` context. As of version 1.3.1, the package provides an additional way to use the icons as Ti*k*Z pics. As of version 1.4.0, a variant of the package provides an alternative way to typeset the icons using the `l3draw` packagee.

2 Loading the package

The package can be installed by copying the file `dndicons.sty` into the working directory or into the `texmf` directory. After the package has been installted, the `dndicons` package is loaded by calling `\usepackage{dndicons}` in the preamble of the document.

```
pics
```

The package provides the option `pics`. If the package is loaded with this option, every icon is also available as Ti*k*Z pic. On the use of pics, see section 3.5 below.

2.1 Dependencies

The package loads the `tikz` package which in turn loads the `xcolor` package. If you want to make use of certain options these packages provide, you need to load the packages with the relevant options beforehand or use, for example, `\PassOptionsToPackage{svgnames}{xcolor}`.

* E-mail: mail@jasperhabicht.de

3 Usage

Once loaded, the package provides a set of commands that can be used to print icons inline. The package also defines a set of node shapes that can be used inside a `tikzpicture` environment.

3.1 Global settings

```
dnd icons
```

All icons share the TikZ style `dnd icons` that has no options per default but can be used to style all icons at once. For example, if `\tikzset{dnd icons/.append style={draw=red}}` is placed at the beginning of the document, all icons will be drawn in red. Per default, the icons are drawn in the color of the surrounding text.

Note that it may be necessary to add the TikZ option `transform shape` when applying transformations to the icons, because the icons are realised as TikZ nodes which are not affected by some transformations per default.

```
dnd icons/background color
```

Some icons can be used with a negative color scheme where the icon is drawn negatively inside a filled shape. Per default, the icons are drawn in white in such cases, but it might be desirable that the icons are in the same color as the background. To this end, the color can be changed using the TikZ option `dnd icons/background color` in the following way:

```
\colorbox{blue!50}{%
    \ability[negative]{charisma}
        [scale=2, transform shape]%
}

\tikzset{
    dnd icons/background color={blue!50}
}
\colorbox{blue!50}{%
    \ability[negative]{charisma}
        [scale=2, transform shape]%
}
```

This feature can, of course, also be used to change the color of the icon independently from the color of the background.

```
dnd icons/before sep
dnd icons/after sep
dnd icons/baseline
```

The TikZ options `dnd icons/before sep` and `dnd icons/after sep` are used to define the width of the space that is added before and after the icons respectively. The default value of both lengths is 0.05 em. For example, setting the space before icons to 1 cm can be achieved as follows:

```
                                Roll\die{eightside}{}a die!

Roll a die!                     \tikzset{
Roll      a die!                    dnd icons/before sep={1cm}
                                }
                                Roll\die{eightside}{}a die!
```

The option `baseline` can be used to adjust the baseline of the icons. A larger value for the baseline will shift the icon downwards relative to the baseline of the surrounding text. The default value of the baseline is –3.5 pt.

3.2 Icons

Because the package defines the icons as Ti*k*Z shapes, every command can actually be used together with every shape. However, the combinations of shapes and commands as described in the following subsections are preferable.

```
\dndiconsdie[‹style›]{‹shape›}[‹options›]{‹integer›}
\dndiconsability[‹style›]{‹shape›}[‹options›]
\dndiconssaving[‹style›]{‹shape›}[‹options›]
\dndiconsspell{‹shape›}[‹options›]
\dndiconsspellschool[‹style›]{‹shape›}[‹options›]
\dndiconsdamage{‹shape›}[‹options›]
\dndiconsattack{‹shape›}[‹options›]
\dndiconscondition{‹shape›}[‹options›]
```

The package defines a set of alternative commands which can be used in cases where another package defines one of the basic commands `\die`, `\ability`, `\saving`, `\spell`, `\spellschool`, `\damage`, `\attack` or `\condition` which are described in the following in detail. These alternative commands are exact copies of the relevant basic commands and take the same set or arguments.

3.2.1 Icon `\die`

```
\die[‹style›]{‹shape›}[‹options›]{‹integer›}
```

The command `\die` prints an icon to depict a die with a certain number of sides. Two special icons exist for a two-sided die (which would be equivalent to a coin) and for a hundred-sided die (which typically comes in the shape of a sphere). There is also a special icon for a fudge die.

For the six-sided die, nine additional shapes exists representing the values one to nine as pips. Also, additional shapes exist representing the plus or minus side of a fudge die.

The command takes two mandatory commands, the first of which describes the shape (see the lists below) and the second can take an integer that is placed in front of the shape. For example, `\die{eightside}{2}` results in 2 (meaning two eight-sided dice are rolled).

The command also takes two optional arguments, the second of which can take arbitrary Ti*k*Z options to style the icon. The options affect the shape, not the integer when it is printed before the icon. As an example, `\die{eightside}[blue, thick]{2}` results in 2.

The first optional argument can take the value `normal` or `large`, `normal` being the default value. With `large` given as argument, the icon is drawn larger and the additional integer is printed inside of the shape instead of before it. As an example, `\die[large]{eightside}{2}` results in 2. Note that the integer will always be printed on top of the shape, even if the shape does not have an open center as in the case of the `fudge` shapes or the shapes featuring pips.

Command	Icon	Shape
`\die`	⬭	`twoside`
	△	`fourside`
	□	`sixside`
	⯃	`eightside`
	◇	`tenside`
	⬠	`twelveside`
	⯂	`twentyside`
	◍	`hundredside`
	⊞	`fudge`
	⚀	`sixside one`
	⚁	`sixside two`
	⚂	`sixside three`
	⚃	`sixside four`
	⚄	`sixside five`
	⚅	`sixside six`
	🎲	`sixside seven`
	🎲	`sixside eight`
	🎲	`sixside nine`
	⊞	`fudge plus`
	⊟	`fudge minus`

3.2.2 Icons `\ability` and `\saving`

\ability[‹style›]{‹shape›}[‹options›]

The command `\ability` prints icons depicting an ability of a character. The abilities are represented by animal-like shapes. The relevant shape should be given as mandatory argument to the command. The second optional argument can take arbitrary Ti*k*Z options to style the icon.

The first optional argument can take the value `positive` or `negative`, `positive` being the default value. With `negative` given as argument, the icon is drawn negative inside a circle. As an example, `\ability[negative]{charisma}` results in .

\saving[‹style›]{‹shape›}[‹options›]

The command `\saving` prints the icons that can be typeset using the `\ability` command inside a small shield. It can take the same values for the mandatory argument as the `\ability` command. The optional argument can take arbitrary Ti*k*Z options to style the icon.

The first optional argument can take the value `normal` or `empty`, `normal` being the default value. With `empty` given as argument, the icon inside the shield is not printed. In this case, the mandatory argument can be left empty. As an example, `\saving[empty]{}` results in ⛉.

Command	Icon	Shape
`\ability`		`strength`
		`dexterity`
		`dexterity alt`
		`constitution`
		`intelligence`
		`wisdom`
		`charisma`
		`resilience`

Command	Icon	Shape
		`sanity`
		`perception`
		`luck`
		`armor`
		`proficiency`
`\saving`		`strength`
		`dexterity`
		`dexterity alt`
		`constitution`
		`intelligence`
		`wisdom`
		`charisma`
		`resilience`
		`sanity`
		`perception`
		`luck`
		`armor`
		`proficiency`

3.2.3 Icon `\spell`

```
\spell{‹shape›}[‹options›]
```

The command `\spell` prints icons depicting the effect of a spell or how it is to be effected. The optional argument can take arbitrary Ti*k*Z options to style the icon.

Command	Icon	Shape
`\spell`		`linear`
		`conic`
		`quadratic`
		`cubic`
		`spheric`
		`cylindric`
		`verbal`
		`somatic`
		`material`
		`focus`

3.2.4 Icon `\spellschool`

```
\spellschool[‹style›]{‹shape›}[‹options›]
```

The command `\spellschool` prints icons that represent the school a spell belongs to. The second optional argument can take arbitrary Ti*k*Z options to style the icon.

The first optional argument can take the value `negative` or `positive`, `negative` being the default value. Per default the icon is drawn in white inside a filled escutcheon. With `positive` given as argument, the icon as well as the escutcheon are drawn in the currently selected color. As an example, `\spellschool[positive]{evocation}` results in .

Command	Icon	Shape
`\spellschool`		`abjuration`
		`conjuration`
		`divination`
		`enchantment`
		`evocation`
		`illusion`
		`necromancy`
		`transmutation`

3.2.5 Icons `\damage`, `\attack`, and `\condition`

`\damage`{‹shape›}[‹options›]

The command `\damage` prints icons depicting the damage of an attack. The icon is printed inside a circle. The optional argument can take arbitrary Ti*k*Z options to style the icon.

`\attack`{‹shape›}[‹options›]

The command `\attack` prints icons depicting the kind of an attack. The optional argument can take arbitrary Ti*k*Z options to style the icon.

`\condition`{‹shape›}[‹options›]

The command `\condition` prints icons depicting a condition of a character. The optional argument can take arbitrary Ti*k*Z options to style the icon.

Command	Icon	Shape
`\damage`		`acid`
		`bludgeoning`
		`cold`
		`fire`
		`force`
		`lightning`
		`necrotic`
		`piercing`
		`poison`
		`psychic`
		`radiant`
		`slashing`
		`thunder`
		`healing`
`\attack`		`melee`
		`ranged`
		`magic`
		`singlehanded`
		`doublehanded`
`\condition`		`buff`
		`blinded`
		`charmed`

Command	Icon	Shape
		`deafened`
		`exhausted`
		`frightened`
		`grappled`
		`incapacitated`
		`invisible`
		`paralyzed`
		`petrified`
		`poisoned`
		`prone`
		`restrained`
		`stunned`
		`unconscious`
		`hearing`
		`seeing`

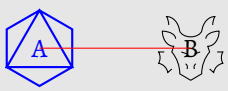
3.3 Icon styles

```
dnd icons/every die
dnd icons/every ability
dnd icons/every saving
dnd icons/every spell
dnd icons/every spellschool
dnd icons/every damage
dnd icons/every attack
dnd icons/every condition
dnd icons/every ‹shape›
```

Using Ti*k*Z styles, all instances of a certain command or a certain shape can be styled at once. These styles all follow the pattern `dnd icons/every` followed by a space and the name of the command or the shape. For example, `\tikzset{dnd icons/every die/.append style={red}}` can be used to draw in red all icons created using the `\die` command. To draw every instace of the `charisma` shape in red, `\tikzset{dnd icons/every charisma/.append style={red}}` can be used.

3.4 Direct use of shapes

Because the icons are defined as Ti*k*Z shapes, they can directly be applied to Ti*k*Z nodes. However, the shapes don't have a shape border and no anchors except for the `center` anchor that sits exactly in the center of the shape. Therefore, if nodes with these shapes are connected using edges, the `center` anchor will be used to connect the nodes. If nodes with these shapes are being positioned, only the `center` anchor is available. Text content of these nodes is simply printed on top of the center of the node. Compare the following example.

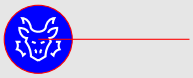



```
\begin{tikzpicture}
    \node[eightside, blue, thick]
        at (0,0) (A) {A};
    \node[charisma] at (2,0) (B) {B};
    \draw[red] (A) -- (B);
\end{tikzpicture}
```

3.5 Icons as pics

If the package is loaded with the option `pics`, every icon is also available as Ti*k*Z pic. The names of the pic always start with `dnd icons` followed by a space and the name of the relevant icon (see the lists above). For abilities, savings, spellschools and damages, additional pics exists where the name has the suffixes `ability`, `saving`, `spellschool`, and `damage` respectively.

The icon is embedded as a node in the pic which has the name `-node`. Thus, it is possible to name the pic and refer to the node inside. Due to the fact that the icon is a node, the option 'transform shape' has to be used if transformations on the pic are to affect the node as well. It is easily possible to apply styles to the node using the Ti*k*Z option `every node` as shown in the following example.



```
\begin{tikzpicture}
    \pic[
        transform shape,
        scale=2,
        fill=blue,
        draw=red,
        every node/.append style={
            white,
            thick
        }
    ] (p) {dnd icons charisma ability};
    \draw[red] (p-node) -- +(2,0);
\end{tikzpicture}
```

```
dnd icons/create pic from shape
dnd icons/create pic from ability shape
dnd icons/create pic from saving shape
dnd icons/create pic from spellschool shape
dnd icons/create pic from damage shape
dnd icons/create every style
```

The package defines five Ti*k*Z keys that are used to create pics using the relevant node shapes. Another key is defined to create keys that can be used to style all instances of a command or shape. In normal circumstances, it is not necessary to use these keys. They are mentioned here only for reference.

3.6 Boxing of icons

Because the icons cannot simply be used inside `tikzpicture` environments, the package provides a workaround to place icons inside of boxes for later use. Icons that are boxed this way can safely used inside `tikzpicture` environments. This might be necessary, if an icon should be used in inline text that sits inside a node.

\provideprotecteddndicon{‹command›}[‹style›]{‹shape›}[‹options›]{‹box name›}

The command `\provideprotecteddndicon` creates a box containing the icon that would be created using one of the regular commands this package provides.

`\provideprotecteddndicon{die}[large]{eightside}[blue, thick]{mybox}`, for example, stores the icon of an eight-sided die with the relevant style and Ti*k*Z options in a new box named `mybox`. Note that no integer can be added to the `die` command in this context.

\useprotecteddndicon{‹box name›}

Using the command `\useprotecteddndicon`, the previously defined box can be used to place the relevant icon. With the above definition, `\useprotecteddndicon{mybox}` would result in  .

Having created a boxed icon, it is safe to use it, for example, inside a Ti*k*Z node:


```
\begin{tikzpicture}
    \node[circle, draw, align=center] {
        \useprotecteddndicon{mybox} \\
        Roll a die!
    };
\end{tikzpicture}
```

4 `l3draw` package variant

A variant of the package that uses the `l3draw` package instead of the `tikz` package is loaded by calling `\usepackage{dndicons-l3draw}` in the preamble of the document after having installed the file `dndicons-l3draw.sty`. The `l3draw` package is an experimental package that provides only basic drawing functionality and therefore loads quicker than Ti*k*Z. The `l3draw` variant thus only supports a certain set of option for styling the icons.

`edge`

The `l3draw` package is experimental and subject to ongoing updates. Therefore, the `l3draw` variant of the `dndicons` package provides a fallback to established and more stable macro definitions (being available since the last major TeX Live release) which is active per default. Loading the `l3draw` variant of the package with the option `edge` will use functions that are only available in the most recent version of the `l3draw` package.

`opacity`

With the `opacity` option, the `l3draw` variant of the `dndicons` package supports transparency using the `l3opacity` package. Since transparency requires management of the PDF settings, it is necessary to call `\DocumentMetadata{}` before loading a `\documentclass`. The `l3opacity` package is experimental.

4.1 Icon commands

The commands of the main variant of the `dndicons` package, `\die`, `\ability`, `\saving`, `\spell`, `\spellschool`, `\damage`, `\attack` and `\condition`, which are described above have the same functionality and can be used the same way as in the main variant of the package. However, the available options are different and described in the following in detail.

\DndIconsUseIcon[‹options›][‹integer›]{‹shape›}
\DndIconsUseIcon*[‹options›][‹integer›]{‹shape›}

`\DndIconsUseIcon` is the primary command to typeset icons using the `l3draw` variant of the package. The commands `\die`, `\ability`, `\saving`, `\spell`, `\spellschool`, `\damage`, `\attack` and `\condition` are based on this command.

The `\DndIconsUseIcon` command has a starred version and two optional arguments as well as one mandatory argument. The mandatory argument holds the shape of the icon. All shapes that are described above for the main variant of the package are available. The second optional argument can be used to add an integer when used with shapes for dice.

The starred version of the command is used to fill a frame with color instead of drawing its outline. Frames can be put around the shape via the relevant `frame` option.

```
\DndiconsDie[‹style›]{‹shape›}[‹options›]{‹integer›}
\DndiconsAbiliy[‹style›]{‹shape›}[‹options›]
\DndiconsSaving[‹style›]{‹shape›}[‹options›]
\DndiconsSpell{‹shape›}[‹options›]
\DndiconsSpellschool[‹style›]{‹shape›}[‹options›]
\DndiconsDamage{‹shape›}[‹options›]
\DndiconsAttack{‹shape›}[‹options›]
\DndiconsCondition{‹shape›}[‹options›]
```

If that the `dndicons` package is to be loaded together with the `dndicons-l3draw` package or some other package that defines one of the above described basic commands `\die`, `\ability`, `\saving`, `\spell`, `\spellschool`, `\damage`, `\attack` or `\condition`, this set of alternative commands can be used. These commands are exact copies of the relevant basic commands and take the same set or arguments.

4.2 Icon options

```
frame
stroke
fill
text
color
background
stroke opacity
fill opacity
text opacity
opacity
background opacity
line width
scale
scale inner
rotate
```

The `\DndIconsUseIcon` command and the commands `\die`, `\ability`, `\saving`, `\spell`, `\spellschool`, `\damage`, `\attack` and `\condition` can be used with certain options that each consist of a key-value pair and can be combined. When used with the `\DndIconsUseIcon` command or the other commands based on this command, these options should be used directly without wrapping them inside the `style` option.

For example, `\die{eightside}[color=blue, line width=0.8pt]{2}` would result in 2.

With the `frame` option, one of four different frames can be selected that are drawn around the shape of the icon. The values `ability` and `damage` draw a circle around the shape. The value `saving` draws a rounded shield and the value `spellschool` draws an angular shield around the shape. The commands `\ability`, `\saving`, `\spellschool` and `\damage` make use of the relevant frame.

The `background` option sets the color of the shape when it is printed over a filled frame which can be achieved by setting the `negative` option for the `\ability` or the `\spellschool` com-

mand or using the starred version of the `\DndIconsUseIcon` command.

The `color` option sets the color of strokes, fills and text in general while the `stroke` option, the `fill` option and the `text` option set the color only for strokes, fills or text respectively. Similarly, the `opacity` macro sets the opacity generally, while the options `stroke opacity`, `fill opacity` and `text opacity` allow for setting the opacity of strokes, fill and text separately. The opacity styles are only available when loading the package with the `opacity` option (see the . The `line width` sets the line width for strokes. Using the `scale` and `rotate` options, the shape can be scaled and rotated.

The `scale inner` option can be used to change the scaling of the icon when placed inside a frame when using the `\ability`, `\saving`, `\spellschool` and `\damage` macros. The default value is 0.675.

```
accessible
accessible = basic
accessible = full
accessible = false
```

Using the `accessible` option with no value or the value `basic`, a replacement text to the relevant icon can be added which can be copied to the clipboard. The value `full` additionally places an `\Alt` mark in the PDF which means that the shape of the relevant icon is read by screen readers. For this to work, is necessary to call `\DocumentMetadata{}` before loading a `\documentclass`, preferably with tagging enabled to make use of the `accessible` option. The value `full` can be used to disable the function explicitly. Note that depending on the compiler and PDF viewer, there might be line breaks or no spaces before and after the replacement text.

```
every die
every ability
every saving
every spell
every spellschool
every damage
every attack
every condition
every ‹shape›
```

Styles following the pattern `dnd icons/every` followed by a space and the name of the command or the shape can be used to apply styles to every instance of this command or shape. For example, `\dndiconsset{every die={color={red}}}` can be used to draw in red all icons created using the `\die` command. Calling `\dndiconsset{every charisma={color={red}}}` will draw every instace of the `charisma` shape in red.

4.3 Setting options globally

`\dndiconsset`

Apart from setting the options to the commands directly, it is also possible to set them globally using the `\dndiconsset` command. Globally set options are overridden by options that are selected directly.

```
\dndiconsset{
    color=blue
}

\ability{charisma}
\ability{charisma}[color=red]
\ability{charisma}
```

```
before sep
after sep
baseline
```

Similar to the settings of the main variant of the package, the spacing before and after the icons can be set using the options `before sep` and `after sep`. The option `baseline` can be used to adjust the baseline of the icons. These options can also be applied to the icon commands directly.

```
Roll\die{eightside}{}a die!

\dndiconsset{
    before sep={1cm}
}
Roll\die{eightside}{}a die!
```

Roll a die!
Roll a die!

5 Changes

v1.1.0 (2023/08/15)
First public release.

v1.1.1 (2023/11/15)
Fudge dice icon added.

v1.1.2 (2023/11/16)
Bug fixed that caused wrong spacing when using dice icons without quantifier.

v1.2.0 (2023/11/20)
Corrections in the manual. Icons for six-sided dice with one to nine pips, plus sign and minus sign added.

v1.2.1 (2023/11/20)
Documentation of lengths for spacing added.

v1.2.2 (2023/11/21)
Corrections in the manual.

v1.3.0 (2023/11/21)
Option to set background color added. Renamed global option.

v1.3.1 (2024/02/18)
Correction of initializing code. Correction of default value of after sep. Addition of pics.

v1.3.2 (2024/02/19)
Reformatting of the manual.

v1.3.4 (2024/02/20)
Minor corrections of some shapes.

v1.4.0 (2024/02/21)
`l3draw` variant added.

v1.4.1 (2024/02/21)
Corrections in `l3draw` variant.

v1.4.2 (2024/02/21)
Alternative set of commands in `l3draw` variant defined.

v1.4.3 (2024/02/22)
Added option to `l3draw` variant to enable upward compatibility.

v1.4.4 (2024/02/24)
Added styles for every instance of command or shape, correction of recent `l3draw` code.

v1.5.0 (2024/02/25)
Alternative set of commands defined, added support of styles in pics.

v1.5.1 (2024/02/28)
Addition of opacity to `l3draw` variant.

v1.5.2 (2024/02/29)
Unification of scaling, minor corrections.

v1.5.3 (2024/03/03)
Optimisation of code in `l3draw` variant.

v1.5.4 (2024/03/06)
Correction of baseline settings in `l3draw` variant, added accessibililty support for `l3draw` variant.

v1.5.5 (2024/03/07)
Optimsation of accessibililty support for `l3draw` variant.

v1.6.0 (2024/03/15)
Four attribute icons added, minor correction of styles.