The asapsym package

Paul D. Gessler
pdgessler@gmail.com

v1.0 from 2016/03/20

1 Introduction

The asapsym package provides convenience macros for using the symbols contained in the included *ASAP Symbol* font. The font is part of the ASAP family[1] of fonts. The font is designed by Omnibus-Type[2] and released under the SIL Open Font License (OFL)[3], version 1.1.

Development of the package is currently hosted on GitHub[4]. You are welcome to follow development, submit any issues you discover, or suggest improvements/enhancements using the web interface there.

2 Usage

To use the macros, you must use an OTF-capable engine with ε-TeX extensions (XeTeX or LuaTeX both fit the bill). To load the macros, choose the loading mechanism based on the format you are using:

```
\usepackage{asapsym}    % LaTeX
\input asapsym.code.tex % Plain TeX
```

Once the package is loaded, all macros are available for use throughout your document. Refer to sections 4 to 9 for a complete listing of macros and their resulting symbols, grouped by category.

3 Future Work

In future releases, I plan to add:

- color support for the 'not' sign overlays,
- switches/options for displaying male/female variants of the people symbols,

[1] http://www.omnibus-type.com/fonts/asap.php
[2] http://www.omnibus-type.com/index.php
[3] http://scripts.sil.org/OFL
[4] http://github.com/pdgessler/asapsym

- alias commands for custom user naming, and
- ConTEXt support.

4 Arrows

Macro	Symbol
\asapArrowLeft	←
\asapArrowUpLeft	↖
\asapArrowUp	↑
\asapArrowUpRight	↗
\asapArrowRight	→
\asapArrowDownRight	↘
\asapArrowDown	↓
\asapArrowDownLeft	↙
\asapArrowCircleOpenLeft	⭠
\asapArrowCircleOpenUpLeft	⭡
\asapArrowCircleOpenUp	⭡
\asapArrowCircleOpenUpRight	⭡
\asapArrowCircleOpenRight	⭢
\asapArrowCircleOpenDownRight	⭣
\asapArrowCircleOpenDown	⭣
\asapArrowCircleOpenDownLeft	⭣
\asapArrowCircleFillLeft	⭠
\asapArrowCircleFillUpLeft	⭡
\asapArrowCircleFillUp	⭡
\asapArrowCircleFillUpRight	⭡
\asapArrowCircleFillRight	⭢
\asapArrowCircleFillDownRight	⭣
\asapArrowCircleFillDown	⭣
\asapArrowCircleFillDownLeft	⭣

5 Elevators

Macro	Symbol
\asapElevator	
\asapStair	
\asapStairDown	
\asapStairUp	
\asapEscalator	
\asapEscalatorDown	
\asapEscalatorUp	

6 Objects

Macro	Symbol
\asapBook	
\asapEnvelope	
\asapGift	
\asapLocker	
\asapLostAndFound	
\asapMicroscope	
\asapCross	
\asapPhone	
\asapMobilePhone	
\asapTablet	
\asapMonitor	
\asapUtensils	
\asapMug	
\asapHanger	
\asapCigarette	

7 People

Macro	Symbol
\asapFemaleWithServiceAnimal	
\asapMaleWithServiceAnimal	
\asapFemaleWalkingDog	
\asapMaleWalkingDog	
\asapFemaleWalking	
\asapMaleWalking	
\asapFemaleWithLuggageWaving	
\asapMaleWithLuggageWaving	
\asapFemaleWithLuggageWaiting	
\asapMaleWithLuggageWaiting	
\asapFemaleDiscardingTrash	
\asapMaleDiscardingTrash	
\asapFemaleAtHelpDesk	
\asapMaleAtHelpDesk	
\asapFemaleHoldingInfant	
\asapMaleHoldingInfant	
\asapFemaleWalkingStroller	
\asapMaleWalkingStroller	
\asapFemaleWithChild	
\asapMaleWithChild	
\asapWalkingCane	
\asapWaitingSeated	
\asapFemaleAtDrinkingFountain	
\asapMaleAtDrinkingFountain	
\asapFemaleAdult	
\asapMaleAdult	
\asapFemaleChild	
\asapMaleChild	
\asapWheelchairStationary	
\asapWheelchairInMotion	
\asapPregnant	
\asapGroupMeeting	
\asapCycling	
\asapDog	
\asapInfant	

8 Signals

Macro	Symbol
\asapInformationSign	
\asapHospitalSign	
\asapHelpSign	
\asapDollarSign	
\asapEmergencySign	
\asapParkingSign	
\asapWalkSign	
\asapDogSign	
\asapMobilePhoneSign	
\asapCigaretteSign	
\asapNotSign	
\asapNotInformationSign	
\asapNotHospitalSign	
\asapNotHelpSign	
\asapNotDollarSign	
\asapNotEmergencySign	
\asapNotParkingSign	
\asapNotWalkSign	
\asapNotDogSign	
\asapNotMobilePhoneSign	
\asapNotCigaretteSign	

9 Transportation

Macro	Symbol
\asapBoat	
\asapHelicopter	
\asapAirplaneOverhead	
\asapAirplaneTakeoff	
\asapAirplaneLanding	
\asapBicycle	
\asapAutomobile	
\asapTaxi	
\asapAutomobileWithKey	
\asapBus	
\asapTrain	

10 Implementation

The implementation of the macros is divided into three parts:

- a LaTeX package wrapper file (`asapsym.sty`),
- a Plain TeX loader file (`asapsym.code.tex`), and
- the generic macro definition file (`asapsym-generic.tex`).

The third file is read in by each of the first two files after their own format-specific macros are defined.

10.1 LaTeX Package Wrapper

We use fontspec for loading OTF fonts and create a new font family to be used by the package:

```
\RequirePackage{fontspec}
\newfontfamily{\asapsym}{Asap Symbol}
```

The fontspec package takes care of requiring OTF-capable engines, producing a prominent error if one is not being used.

`\asapsym@raw` The `\asapsym@raw{`⟨*stylistic-set*⟩`}{`⟨*character*⟩`}` macro is used internally to typeset the actual symbol. The first argument is the stylistic set from which the symbol is to be taken. For simplicity, the leading zero must be present for stylistic sets 9 and below. The second argument is the character where symbol exists in the chosen stylistic set.

```
\def\asapsym@raw#1#2{\bgroup\asapsym%
  \addfontfeatures{StylisticSet=#1}#2\egroup}
```

The font is selected, the stylistic set applied, and the character is typeset within a group.

Having defined the only format-specific code, the generic code is `\input`:

```
\input{asapsym-generic}
```

10.2 Plain TeX Loader

We adopt the LaTeX convention of using `@` in protected macro names throughout the code. Therefore, we must first save the current category code of `@` for later restoration, and set its category code to 'letter'

```
\edef\asapsymatcode{\the\catcode`\@}
\catcode`\@=11
```

Next, we do some engine detection to ensure that a compatible one is used:

```
\ifdefined\XeTeXinterchartoks % we are in XeTeX
\else\ifdefined\directlua     % we are in LuaTex
```

For LuaTeX, we use `luaotfload.sty` so that the same font loading syntax may be used in XeTeX and LuaTeX:

```
    \input luaotfload.sty
```

If a compatible engine is not detected, we issue an error message and stop loading the remainder of the file:

```
\else  % not supported
  \errmessage{asapsym error: Not an OTF-capable engine! Use xetex or luatex}
  \expandafter\endinput
\fi
\fi
```

`\asapsym@raw` The `\asapsym@raw{`⟨*stylistic-set*⟩`}{`⟨*character*⟩`}` macro is used internally to typeset the actual symbol, similar to the LaTeX variant. Here, we use the `\font` primitive to select the font and stylistic set:

```
\def\asapsym@raw#1#2{\bgroup\font\asapsym =
  "[Asap-Symbol.otf]:+ss#1"\asapsym #2\egroup}
```

Having defined all the format-specific code, the generic code is `\input`:

```
\input asapsym-generic
```

Finally, we restore the category code of `@`:

```
\catcode`\@=\asapsymatcode
```

10.3 Generic Macro Code

Define a prefix to be used by all user-facing macros:

```
\def\asapsym@prefix{asap}
```

`\asapsym@set` The `\asapsym@set{`⟨*type*⟩`}{`⟨*stylistic-set*⟩`}` macro defines a helper macro for a given 'type' of symbol.

```
\def\asapsym@set#1#2{\expandafter\def%
  \csname\asapsym@prefix #1\endcsname##1{\asapsym@raw{#2}{##1}}}
```

Define a helper macro for each 'type' of symbol. The font is implemented such that each 'type' of symbol exists within a specific stylistic set:

```
\asapsym@set{arrow}{06}
\asapsym@set{elevator}{03}
\asapsym@set{object}{04}
\asapsym@set{people}{01}
\asapsym@set{signal}{05}
\asapsym@set{transport}{02}
```

`\asapsym@sym` The `\asapsym@sym{`⟨*base-name*⟩`}{`⟨*type*⟩`}{`⟨*char*⟩`}` defines a user-facing symbol macro. The resulting macro's `\csname` is comprised of the package prefix and `{`⟨*base-name*⟩`}`. The symbol is taken from character slot `{`⟨*char*⟩`}` of the stylistic set corresponding to `{`⟨*type*⟩`}`.

```
\def\asapsym@sym#1#2#3{\expandafter\def%
  \csname\asapsym@prefix#1\endcsname{\csname\asapsym@prefix#2\endcsname{#3}}}
```

asapsym@mka `\asapsym@mka{`⟨*base-name*⟩`}{`⟨*char*⟩`}` is used to create a user-facing arrow symbol macro. The macro name is comprised of the package prefix, a prefix `Arrow`, and `{`⟨*base-name*⟩`}`. The symbol is taken from character slot `{`⟨*char*⟩`}` of the arrows font.

```
\def\asapsym@mka#1#2{\asapsym@sym{Arrow#1}{arrow}{#2}}
```

All arrows are defined using the *make arrow* helper macro:

```
\asapsym@mka{Left}{A}
\asapsym@mka{UpLeft}{B}
\asapsym@mka{Up}{C}
\asapsym@mka{UpRight}{D}
\asapsym@mka{Right}{E}
\asapsym@mka{DownRight}{F}
\asapsym@mka{Down}{G}
\asapsym@mka{DownLeft}{H}
\asapsym@mka{CircleOpenLeft}{I}
\asapsym@mka{CircleOpenUpLeft}{J}
\asapsym@mka{CircleOpenUp}{K}
\asapsym@mka{CircleOpenUpRight}{L}
\asapsym@mka{CircleOpenRight}{M}
\asapsym@mka{CircleOpenDownRight}{N}
\asapsym@mka{CircleOpenDown}{O}
\asapsym@mka{CircleOpenDownLeft}{P}
\asapsym@mka{CircleFillLeft}{Q}
\asapsym@mka{CircleFillUpLeft}{R}
\asapsym@mka{CircleFillUp}{S}
\asapsym@mka{CircleFillUpRight}{T}
\asapsym@mka{CircleFillRight}{U}
\asapsym@mka{CircleFillDownRight}{V}
\asapsym@mka{CircleFillDown}{W}
\asapsym@mka{CircleFillDownLeft}{X}
```

asapsym@mke `\asapsym@mke{`⟨*base-name*⟩`}{`⟨*char*⟩`}` is used to create a user-facing elevator symbol macro. The macro name is comprised of the package prefix and `{`⟨*base-name*⟩`}`. The symbol is taken from character slot `{`⟨*char*⟩`}` of the elevators font.

```
\def\asapsym@mke#1#2{\asapsym@sym{#1}{elevator}{#2}}
```

All elevators are defined using the *make elevator* helper macro:

```
\asapsym@mke{Elevator}{A}
\asapsym@mke{Stair}{B}
\asapsym@mke{StairDown}{C}
\asapsym@mke{StairUp}{D}
\asapsym@mke{Escalator}{E}
\asapsym@mke{EscalatorDown}{F}
\asapsym@mke{EscalatorUp}{G}
```

asapsym@mko `\asapsym@mko{`⟨*base-name*⟩`}{`⟨*char*⟩`}` is used to create a user-facing object symbol macro. The macro name is comprised of the package prefix and `{`⟨*base-name*⟩`}`. The symbol is taken from character slot `{`⟨*char*⟩`}` of the objects font.

```
\def\asapsym@mko#1#2{\asapsym@sym{#1}{object}{#2}}
```

All objects are defined using the *make object* helper macro:

```
\asapsym@mko{Book}{A}
\asapsym@mko{Envelope}{B}
\asapsym@mko{Gift}{C}
\asapsym@mko{Locker}{D}
\asapsym@mko{LostAndFound}{E}
\asapsym@mko{Microscope}{F}
\asapsym@mko{Cross}{G}
\asapsym@mko{Phone}{H}
\asapsym@mko{MobilePhone}{I}
\asapsym@mko{Tablet}{J}
\asapsym@mko{Monitor}{K}
\asapsym@mko{Utensils}{L}
\asapsym@mko{Mug}{M}
\asapsym@mko{Hanger}{N}
\asapsym@mko{Cigarette}{O}
```

asapsym@mkp `\asapsym@mkp{`⟨*base-name*⟩`}{`⟨*char*⟩`}` is used to create a user-facing people symbol macro. The macro name is comprised of the package prefix and `{`⟨*base-name*⟩`}`. The symbol is taken from character slot `{`⟨*char*⟩`}` of the people font.

```
\def\asapsym@mkp#1#2{\asapsym@sym{#1}{people}{#2}}
```

asapsym@mkd `\asapsym@mkd{`⟨*base-name*⟩`}{`⟨*char*⟩`}` is used to create gender-specific user-facing people symbol macros. The macro names are comprised of the package prefix, `Male` or `Female` as the case may be, and `{`⟨*base-name*⟩`}`. The symbols are taken from character slot `{`⟨*char*⟩`}` of the people font.

```
\def\asapsym@mkd#1#2{
  \asapsym@mkp{Male#1}{\uppercase{#2}}
  \asapsym@mkp{Female#1}{\lowercase{#2}}}
```

All people are defined using the *make people* helper macro. Some symbols have the gender-specific variants, while others do not. `\asapsym@mkd` and `\asapsym@mkp` are used as appropriate.

```
\asapsym@mkd{WithServiceAnimal}{A}
\asapsym@mkd{WalkingDog}{B}
\asapsym@mkd{Walking}{C}
\asapsym@mkd{WithLuggageWaving}{D}
\asapsym@mkd{WithLuggageWaiting}{E}
\asapsym@mkd{DiscardingTrash}{F}
\asapsym@mkd{AtHelpDesk}{G}
\asapsym@mkd{HoldingInfant}{H}
\asapsym@mkd{WalkingStroller}{I}
\asapsym@mkd{WithChild}{J}
\asapsym@mkp{WalkingCane}{K}
\asapsym@mkp{WaitingSeated}{L}
\asapsym@mkd{AtDrinkingFountain}{M} % ??? is this correct ???
\asapsym@mkd{Adult}{N}
```

```
\asapsym@mkd{Child}{O}
\asapsym@mkp{WheelchairStationary}{P}
\asapsym@mkp{WheelchairInMotion}{p}
\asapsym@mkp{Pregnant}{Q}
\asapsym@mkp{GroupMeeting}{R}
\asapsym@mkp{Cycling}{S}
\asapsym@mkp{Dog}{T}
\asapsym@mkp{Infant}{U}
```

asapsym@mks `\asapsym@mks{`⟨*base-name*⟩`}{`⟨*char*⟩`}` is used to create user-facing sign symbol macros, along with 'negated' versions. The macro names are comprised of the package prefix, `Not` for negated versions, `{`⟨*base-name*⟩`}`, and `Sign`. The symbols are taken from character slot `{`⟨*char*⟩`}` of the people font. For the negated versions, the not sign is overlaid.

```
\def\asapsym@mks#1#2{
  \asapsym@sym{#1Sign}{signal}{#2}
  \asapsym@sym{Not#1Sign}{signal}{#2\llap{X}}}
```

All signals are defined using the *make signal* helper macro:

```
\asapsym@mks{Information}{A}
\asapsym@mks{Hospital}{B}
\asapsym@mks{Help}{C}
\asapsym@mks{Dollar}{D}
\asapsym@mks{Emergency}{E}
\asapsym@mks{Parking}{F}
\asapsym@mks{Walk}{G}
\asapsym@mks{Dog}{H}
\asapsym@mks{MobilePhone}{I}
\asapsym@mks{Cigarette}{J}
\asapsym@mks{Not}{X}
```

asapsym@mkt `\asapsym@mkt{`⟨*base-name*⟩`}{`⟨*char*⟩`}` is used to create a user-facing transporation symbol macro. The macro name is comprised of the package prefix and `{`⟨*base-name*⟩`}`. The symbol is taken from character slot `{`⟨*char*⟩`}` of the transportation font.

```
\def\asapsym@mkt#1#2{\asapsym@sym{#1}{transport}{#2}}
```

All transports are defined using the *make transportation* helper macro:

```
\asapsym@mkt{Boat}{A}
\asapsym@mkt{Helicopter}{B}
\asapsym@mkt{AirplaneOverhead}{C}
\asapsym@mkt{AirplaneTakeoff}{D}
\asapsym@mkt{AirplaneLanding}{E}
\asapsym@mkt{Bicycle}{F}
\asapsym@mkt{Automobile}{G}
\asapsym@mkt{Taxi}{H}
\asapsym@mkt{AutomobileWithKey}{I}
\asapsym@mkt{Bus}{J}
\asapsym@mkt{Train}{K}
```

Change History

v1.0

General: Initial 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.

A

\addfontfeatures 4
\asapsym . . . 2, 3, 16, 17
\asapsym@mka 31, 31, 32–55
\asapsym@mkd 81, 81, 84–93, 96–98
\asapsym@mke 56, 56, 57–63
\asapsym@mko 64, 64, 65–79
\asapsym@mkp 80, 80, 82, 83, 94, 95, 99–105
\asapsym@mks 106, 106, 109–119
\asapsym@mkt 120, 120, 121–131
\asapsym@prefix 20, 22, 30
\asapsym@raw . . 3, 16, 22
\asapsym@set . . 21, 23–28
\asapsym@sym 29, 31, 56, 64, 80, 107, 108, 120
\asapsymatcode . . . 6, 19

F

\font 16

L

\llap 108
\lowercase 83

N

\newfontfamily 2

U

\uppercase 82