

chessfss: A Chess Font Selection Scheme

Ulrike Fischer

June 14, 2006

Contents

1	Requirements	2
2	Changes of chessfss.sty	2
3	Introduction	4
3.1	Some history	4
3.2	Description of the fonts	5
3.3	Some remarks about nfss	5
3.4	For professionals: how to preserve kerning	6
3.5	Installing the skaknew fonts	7
3.6	User and package writers commands	8
3.7	Warnings	8
4	Using and changing the figurine font	8
4.1	Figurines without the package chessfss	8
4.2	Using symbols for pieces	9
4.3	Using normal text chars for the pieces	10
4.4	Commands that can switch between text and symbols	11
4.5	Kerning questions	12
4.6	Changing the look of the figurines	12
5	Using and changing the board font	13
5.1	Boards without the package chessfss	13
5.2	Simple command to use and change board fonts	14
5.3	Layers of Layers of Chars: Using and coloring composed chars	16
5.4	The logical layers	17
5.5	The predefined encodings	17
5.6	Coloring the layers	20

6	Using and changing the informant font	21
7	Other font changes	22
7.1	Using and changing the font on the side of the board	22
7.2	Changing the text font	22
7.3	Changing more than one font	23
8	Miscellaneous commands	23
9	Character and command tables of the encodings	23
	The figurines	23
	The board chars	24
	The informant symbols	25
10	Converting and installing a ttf-chessfont or a type1-chessfont for use with the package skak	26
10.1	Attention	26
10.2	The naming scheme	26
	10.2.1 Example	27
10.3	Testing of the fonts	27
10.4	Quick installation guide	28
10.5	A more complete installation guide	28
	10.5.1 Convert a ttf-font to a pfb-font.	28
	10.5.2 Installing the type1-fonts – the brute force method	29
	10.5.3 Update all databases and maps and test the new fonts	31
	10.5.4 Installing a type1 chess font with fontinst	31
10.6	chess-fig-example-enc.enc – an example	32
10.7	chess-board-example-enc.enc – an example	33
11	Fontnames and other informations	36

1 Requirements

The package chessfss uses some primitives of eTeX. It needs a recent version of xkeyval (key = value-syntax).

The newest version of the skaknew-fonts should be installed.

2 Changes of chessfss.sty

2006-06-14 Added commands `\shortcastling` and `\longcastling` for castling. Added `\chesscomment` as a copy of `\comment` (which clashes with the moreverb package).

Added another board encoding LSBC5.

2006-06-09 Changed default (“fallback”) chess font to skaknew. Disabled the package option skaknew as it has no sense anymore.

2006-06-07 Corrected faulty definition of `\chesssee`

2006-05-26 Replaced the experimental board encodings by the final ones and added color support.

2006-05-26 Changed and extended definitions for text figurines.

2006-05-16 Corrected an error in the description of .enc-files and in the description which caused errors with dvips.

2006-03-31 I changed the name of the informator symbol `\see` to `\chesssee` to avoid a clash with makeidx.

Fontencoding declaration are now in external .def-files. fd-informations of the standard fonts are in external .fd-files.

Changed the internal definition of the pieces. They now use `\DeclareTextCommand` etc. So they now longer execute font commands like `size!` The user definitions like `\WhiteEmptySquare` or `\king` are unchanged.

I added an LSBXA-encoding and a LSBXB-encoding. They can be used to color the back of board pieces differently to the piece itself. (This encodings have been replaced by the final ones and so don't exist anymore).

2005-12-15 Added two length (squarewidth, squareheight). Added the commands `\cfss@getsquaresize`, `\getsquaresize`. Changed `\cfss@setupboardsize` and `\setboardfontsize`, they now store the dimensions of a square of the board. Changed the definition of `\cfss@boardsymbol` to prevent the insertion of italic correction in boards (`\DeclareTextFontCommand` is no longer used).

2005-12-02 corrected missing % in definition of `\cfss@sidesfont`

2005-06-28 Changed definition of boardfont. The baselineskip is now set to the total height of a square. This corrects a problem with fonts, where the board squares are not really squares.

2005-05-09 Corrected the definitions of `\with` and `\without`. I confused them. Attention: also old versions of the package skak have mixed them up.

3 Introduction

This package offers commands to use and switch between chess fonts. It use the \LaTeX font selection scheme (`nfss`).

Using the package is easy: either load it with `\usepackage{chessfss}` or load a package like the new versions of `skak`, `chessboard` or `texmate` that will load the package `chessfss`. From this version on the package `chessfss` will not have any package option.

The package doesn't parse, format and print PGN (Portable Game Notation) input like e.g. the package `skak` or the package `texmate`.

The aim of the package is to offer writers of packages like the package `skak` a bundle of commands for fonts, so that they don't have to implement all these commands for themselves.

A normal user can use the package `chessfss` to print chesssymbols for decoration. But writing a complete game with the following commands is rather tedious. Luckily the newest version of the package `skak` and `texmate` now uses the package `chessfss`.

If you find errors in this text (this includes wrong english), you can write me a bugreport at `skak@nililand.de`. If you have questions ask them at the newsgroups `comp.text.tex` or `de.comp.text.tex`. I'm reading these groups regularly and I'm much better in answering there than in answering e-mails.

Remark

YAP (the previewer of Mik \TeX) doesn't like chess fonts. I had and have a lot of faulty diagramms with some pieces and/or fields missing. I also had figurines missing in a game. Some errors are only on screen, others are on screen and in the printed document. The same board at another place can be faultless. In all cases the `pdf` and the `ps` file were okay. So to save you from trouble you should better preview and print the `pdf` or `ps` file.

3.1 Some history

In the package `skak` the used chess fonts were hardwired in the code. The package `skak` always used the `skak-fonts` which exist only in the `mf` format. So even if one found other chess fonts (e.g. in `type1` format) and was able to install them for the use in \LaTeX it wasn't easy to use them with the package `skak`.

So I decided to change this and to implement in the package `skak` the `nfss` system for fonthandling. I added and changed a lot of commands in the package `skak` and sent my changes and suggestions to the author. Then I saw the package `texmate` and realized that all the commands should better go in a separate package so that all chess related styles can use them – and so the package `chessfss` was born.

The package `chessfss` can't deny that it had been written with the package `skak` in mind. Many of the commands and also the encodings are adapted to the needs of the package `skak`. But I'm quite willing to add commands for other needs. Two things come to my mind: they are a lot of chess fonts which have glyphs for the rules on the board side (with small letters and numbers). Up to now I ignored these glyphs as the package `skak` doesn't use them anyway.

3.2 Description of the fonts

While printing chessgames and chessboards we have to do with five different sets of characters:

- The figurine characters, which are used for the pieces while printing a game, e.g. `1 ♠f3 ♠f6`. In the families `skak` and `skaknew`, these characters exist in a medium and a bold version.
- The board characters, which are used for printing the board. While some characters of the board font look very similar to the figurines, their metrics are quite different. All characters of a board should have the same width and the pieces are a bit above the line:

Figurine: . Board: .

- The informant characters. these are symbols like \pm or \triangle . In the `skak` and `skaknew` fonts these symbols are stored in the same font file as the figurine characters. But this need not be the case with other chess families. A lot of fonts even doesn't provide these symbols.
- The notation characters used on the sides of a board. In the package `skak` they are taken from a text font and lifted. Other chess fonts often have small glyphs with a rule and a letter or number.
- Text characters, which are used for the notation of the moves, e.g. "e4". In the package `skak` the font of the document is used.

`chessfss` handles mostly the first three sets of characters.

3.3 Some remarks about `nfss`

In \LaTeX one change to another font by changing one or more property of the font – either with the low-level commands `\fontencoding`, `\fontfamily`, `\fontseries`, `\fontshape` and `\fontsize` or with various highlevel commands.

`nfss` uses a quite good substitution method: if a certain combination of properties doesn't point to an existing font, \LaTeX tries other combinations. As a last resort a default fall back font is used.

I wanted both capacities for the chess fonts too: the `\font...` commands should work where they make sense. And if a chess fonts doesn't exist on a system default chess fonts should be used. In previous versions this default fonts were the `skak-fonts`. I changed this in this version. Now the `skaknew-fonts` are the defaults.

During substitution the encoding is never changed. So if I wanted to make sure that the substitution ends at a chess font, I had to define a chess encoding. After some more thoughts I decided even that *three* chess encodings would be the best. Chess fonts with the full set of characters are quite rare. You can find fonts only with board symbols and fonts only with figurines. I didn't want to have to fill up such fonts via virtual fonts – that would have been possible but tedious.

So I declared three encodings: LSF for chess fonts with figurines, LSB for chess fonts with board symbols and LSI for fonts with the informant symbols:

```
\DeclareFontEncoding{LSF}{}{}
\DeclareFontSubstitution{LSF}{skaknew}{m}{n}
\DeclareFontEncoding{LSI}{}{}
\DeclareFontSubstitution{LSI}{skaknew}{m}{n}
\DeclareFontEncoding{LSB}{}{}
\DeclareFontSubstitution{LSB}{skaknew}{m}{n}
```

The names of the three encodings are stored in the macros `\cfss@figfontencoding`, `\cfss@boardfontencoding` and `\cfss@inffontencoding`. The font commands described on the following pages use these macros. So it is possible to change also the encodings of the chess fonts used. I only offer a user command to change the boardfont encoding as only persons who know what they do should change the others encodings.

3.4 For professionals: how to preserve kerning

Dividing the characters used in the notation of a game in the three logical fonts “figurines”, “text” and “informant symbols” makes it easy to combine e. g. the `skak-figurine` with every text font usable in \LaTeX . But the needed font changing commands (and the different `tfm` files) doesn't allow to define kernings between characters of the different sets.

If you want to be able to define kerning pairs, you will have to build for every combination of text font, figurines and symbol font that you want to use a virtual font where all symbols are in *one* font¹. The characters `a-h`, `1-8` and `!?.`, should be put at their standard ASCII positions. The figurines should be put at the positions `K,Q,R,B,N,p`

¹One font from the view of \TeX : the characters must be in one `tfm` file, the glyphs can come from various fonts.

```
\cfss@figfontencoding
\cfss@boardfontencoding
\cfss@inffontencoding
```

(if you don't want trouble with the package `skak`). For the remaining symbols you can choose the positions freely².

Then you should declare a new (local) encoding, e.g. `LSA`³ and make your new fonts accessible through this encoding. Make a copy of `lsienc.def` that you call `lsaenc.def`. Change in the file all occurrences `LSI` to `LSA`. You will have to adjust all the positions in the `\DeclareTextSymbol{<symbolname>}{LSA}{<position>}`. Copy also all definitions from `lsbenc.def` and `lsfenc.def`. Then you should redefine `\cfss@figfontencoding`, `\cfss@boardfontencoding` and `\cfss@inffontencoding`.

At last if you use a package that sets chess you will have to disable every font changing command it inserts during parsing and printing the notation and you will have to change to your font at the start of the chess notation.

If I get a good suggestion about how to arrange such a font and which characters it should contain I will try to incorporate the needed definitions in the package `chessfss` and to write more detailed instructions.

3.5 Installing the `skaknew`⁴ fonts

Get the folder `skaknew` from CTAN, and put all the files (`tfm`, `pfb`, `afm`, `map`) where they belong (for help look at the FAQ's at <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=wherfiles> and <http://www.dante.de/faq/de-tex-faq/html/makros1.html#7>). Renew the filename database. Add the map-file to your `psfonts.map` (e.g. via a local `updmap.cfg`).

You can ignore the `uskak.fd` files (the old from the package `skak` and the new that comes with `skaknew`).

I will now describe, how you can use and change each of these types of fonts.

I use in this documentation the `skaknew` fonts as default. The `maya` family is a local family of my own for which I installed a figurine and a board font.

²There is no need to try to preserve some positions of the informant symbols, as you will have to redefine them anyway to get rid of the double braces I put around them to avoid trouble with `\mainline` from the package `skak`.

³local `skak` all

⁴The `skaknew` fonts look very similar to `skak` but they are not identical. They are more or less the `type1` version of the original `skak`-fonts. There have been made some small changes and corrections during the conversion from the `mf` sources, so the author decided (and I agree with him) to give the new fonts another name.

3.6 User and package writers commands

As the package `chessfss` is intended mainly as a background package for chess packages many commands exist both in a user and a internal version. Sometimes there is no difference between both versions but it is also possible that they differ. Internal commands starts with `\cfss@`.

3.7 Warnings

When playing with the following commands, you will perhaps get warnings like

```
LaTeX Font Warning: Font shape 'LSI/maya/m/n' undefined (Font)
using 'LSI/skaknew/m/n' instead on input line 4.

LaTeX Font Warning: Font shape 'LSF/maya/bx/n' undefined (Font)
using 'LSF/maya/m/n' instead on input line 5.
```

these warnings show you that the substitution works like wanted: The first warning tells you that in the `maya` family there isn't an LSI-encoded font and that the informant font of the `skaknew` family is used instead. That is a sensible thing to do. The second warning tells you that there isn't a `bx` (bold) font in the `maya` figurine, and that the medium variant is used instead. That is a sensible thing to do, too. I didn't silenced the substitution because I like to know when a font is changed.

4 Using and changing the figurine font

4.1 Figurines without the package `chessfss`

♔a1, ♖b1, ♗c1, ♘a1, ♙b1, ♚c1

♕a1, ♜a1, ♞♟♠a1,

♛a1, ♝b1, ♞c1, ♟a1, ♠b1, ♡c1

```
\documentclass[parskip]{scrartcl}
\usepackage[LSF,T1]{fontenc}
\usepackage{bookman}
\pagestyle{empty}
\begin{document}
\makeatletter
\cfss@symking a1, \cfss@symqueen b1, \cfss@symrook
c1,
\cfss@symbishop a1, \cfss@symknight b1,
\cfss@sympawn c1

{\fontencoding{LSF}\fontfamily{pirat}\selectfont
K}a1,
{\fontencoding{LSF}\fontfamily{pirat}\selectfont
Q}a1,
{\fontencoding{LSF}\fontfamily{pirat}\selectfont
RBNp}a1,

\fontfamily{pirat}\selectfont
\cfss@symking a1, \cfss@symqueen b1, \cfss@symrook
c1,
\cfss@symbishop a1, \cfss@symknight b1,
\cfss@sympawn c1
\end{document}
```

As the above example shows you don't need the package `chessfss` to use chess fonts. You only need to load the encoding definition `LSF` as an option of `fontenc` (and you *must* load a text encoding as the last option of `fontenc`). Then you can use the internal definitions of the encoding definition or the font switching commands. In the examples it looks like as if `\cfss@symking` contains some font switching command but this isn't true: \LaTeX can't find a definition for `\cfss@symking` in the `T1` encoding and so switch to the default encoding for the command.

The above example also shows why you perhaps will want to use the package `chessfss`: the chess symbols and the text chars comes from different fontfamilies. If you try to change the fontfamily for the chesssymbols you must do it for each symbol if you don't want to affect your text font. (In the last two lines of the example document \LaTeX can't find a text font `pirat` and so switch back to the default `cmr`). So it would be quite useful to have commands that do the necessary font switches and commands to change only the chess fontfamily.

4.2 Using symbols for pieces

<code>\figfont</code>	<code>\figfont</code> and <code>\figsymbol{\langle text \rangle}</code> change the encoding to <code>LSF</code> and the fontfamily
<code>\figsymbol</code>	to <code>\cfss@figfontfamily</code> . <code>\figfont</code> is a font switch like e.g. <code>\bfseries</code> , while
<code>\cfss@figfontfamily</code>	<code>\figsymbol</code> takes like <code>\textbf</code> an argument. <code>\cfss@figfontfamily</code> can be changed
<code>\setfigfontfamily</code>	with the command <code>\setfigfontfamily{\langle family \rangle}</code> .

`\symking` For each piece there is a command to call the symbol directly. Font commands like `\bfseries` or `\large` work as expected – if the chess font has the font variants need. new 1.2
 E.g. there is a bold font in the skaknew family, but not in the maya family:

♔c1, ♕d1, ♖h8

```
\newcommand\PieceChar{K}
\figsymbol{K}c1, \figsymbol{\symbol{'113}}d1,
\figsymbol{\PieceChar}h8
```

♔♖♗♘♙♚♛♜♝♞♟♠♡♢♣♤♥♦♧♨♩

```
\figfont KQRBNp
\bfseries KQRBNp
```

♔e1, ♕d1, ♖h1,
 ♗e1, ♘d1, ♙h1
 ♚e1, ♛d1, ♜h1,
 ♝f1, ♞g1, ♟a2

```
\symking e1, \symqueen d1, \symrook h1, \
{\bfseries
\symking e1, \symqueen d1, \symrook h1}

\large\setfigfontfamily{maya}
\symking e1, \symqueen d1, \symrook h1, \
{\bfseries
\symbishop f1, \symknight g1, \sympawn a2}
```

♚♛♜♝♞♟♠♡♢♣♤♥♦♧♨♩

```
\setfigfontfamily{maya}
\figfont KQRBNp
```

`\cfss@symking` As already mentioned above there exist for each piece also an internal command. `\cfss@symking` and `\symking` are not equal. The first is a simple char declared through `\DeclareTextSymbol` in the encoding file. The second is a normal macro which first set the font family and then call the internal command. The difference can be seen in the following example. The internal command reacts only to the standard command to set the font family, while the external command reacts only to the command of the package `chessfss`.

♔, ♕, ♖, ♗
 ♘, ♙, ♚, ♛

```
\makeatletter
{\cfss@symking, \symking,
 \fontfamily{maya}\selectfont \cfss@symking,
 \symking}

{\cfss@symking, \symking,
 \setfigfontfamily{maya}\cfss@symking, \symking}
```

4.3 Using normal text chars for the pieces

Sometimes you don't want to use the symbols but normal chars for the pieces, e.g. if children should learn how to write the notation of a game. Unlike symbols text characters are not international. So some language support is needed. And it would be fine if kerning doesn't get destroyed: ♔e1 should be Te1 in german and not Te1.

`\settextfigchars` `\settextfigchars[language]{king}{queen}{rook}{bishop}{knight}{pawn}` changed 1.2
 defines text output characters for the pieces (e.g. the german abbreviations) for an (optional, default is english) language⁵. The name of the language can be chosen freely, but I suggest to use the name of babel options. I have predefined some language (english, german, spanish, finnish, french, interlingua, icelandic, italian, portuges) but apart from english and german I don't if they are correct.

`\settextfiglanguage` With this command you can switch to the text chars of one defined language. new 1.2

`\textfigsymbol` `\textfigsymbol{piecechar}` takes as argument the english abbreviation of *one* piece and prints the above defined character. There also exist internal and external commands for each piece – there is no difference between the two, package writer should use the internal commands as there is less danger that a user redefines it.

Q, Q

```
\newcommand\PieceChar{Q}
\textfigsymbol{Q}, \textfigsymbol{\PieceChar}
```

Ka1, Qg2, Re1, B, N, P
Kc1, Qg1, Rf3, B, N, P

```
\textking a1, \textqueen g2, \textrook e1,
 \textbishop, \textknight, \textpawn\
\bfseries
\textking c1, \textqueen g1, \textrook f3,
 \textbishop, \textknight, \textpawn\
```

Rc1, Dg1, Te4, A, C, P
 X, Y, Z, ü, ö, ä

```
\settextfiglanguage{italian}
\textking c1, \textqueen g1, \textrook e4,
 \textbishop, \textknight, \textpawn\
\settextfigchars[silly]{X}{Y}{Z}{ü}{ö}{ä}
\settextfiglanguage{silly}
\textking, \textqueen, \textrook, \textbishop,
 \textknight, \textpawn
```

4.4 Commands that can switch between text and symbols

`\textsymfigsymbol` `\textsymfigsymbol{piece}` and `\textsymking{piecechar}` use symbols or text depending on a switch that can be set with `\usetextfig` and `\usesymfig`.

`\cfss@textsymking` There is no difference between the different versions to set a single piece. In previous changed 1.2
`\king` versions `\king` did work like the new `\symking`, but then I decided that a user would prefer if the short commands can switch between text and symbols.
`\usetextfig`
`\usesymfig`

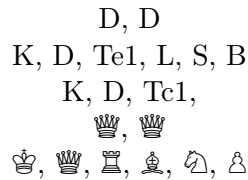
♔, ♕

```
\newcommand\PieceChar{Q}
\textsymfigsymbol{Q}, \textsymfigsymbol{\PieceChar}
```

⁵In the previous versions the command did have the name `\setfigtextchars`. I unified the naming but the older command will work too.



```
\textsymking, \textsymqueen, \textsymrook,
\textsymbishop, \textsymknight, \textsympawn\\
\king, \queen, \rook, \bishop, \knight, \pawn
```



```
\usetextfig\setttextfiglanguage{german}
\textsymfigsymbol{Q},
\textsymfigsymbol{\PieceChar}\\
\king, \queen, \rook e1, \bishop, \knight, \pawn\\
\textsymking, \textsymqueen, \textsymrook c1, \\

\usesymfig
\textsymfigsymbol{Q},
\textsymfigsymbol{\PieceChar}\\
\textsymking, \textsymqueen, \textsymrook,
\textsymbishop, \textsymknight, \textsympawn
```

4.5 Kerning questions

I have tried to write the text commands in such a way that kerning is possible. In most cases it works. But packages like the package `skak` are quite complicated, they have to parse the input and this can break kerning. To my surprise the main commands of the package `skak` preserved kerning, but it breaks with the simple `\wmove`. Sadly the new package `texmate` breaks kerning too:

Skak: 1 Sf3 Sf6 2 Tg1 Tg8
6 Tg1 Te8
Tg1Te8
Texmate: 1. Sf3 Sf6 2. Tg1 Tg8

```
Skak:\newgame
\usetextfig\setttextfiglanguage{german}
\mainline{1. Nf3 Nf6 2. Rg1 Rg8}\\
\variation{6. Rg1 Re8}\\
\wmove{Rg1 Re8}
Texmate: \newgame |1. Nf3 Nf6 2. Rg1 Rg8|
```

4.6 Changing the look of the figurines

`\setfigstyle` With `\setfigstyle{\font commands}`⁶ you can set the style of the pieces. The last command in the argument can be a command which needs one argument. You should be aware that using font commands around the pieces will destroy kernings, and it is always possible that a package like the package `skak` don't like the used font commands.

changed
1.2

⁶In older versions this command was called `\setttextfigstyle`. I changed the name as it can now also affect symbol pieces. The old name works too.

$\text{\textcircled{K}}e1, \text{\textcircled{Q}}d4, \text{\textcircled{R}},$
 $\underline{B}, \underline{N}, \underline{P}$
 $1 \text{\textcircled{N}}f3 \text{\textcircled{N}}f6 \text{\textcircled{R}}g1 \text{\textcircled{R}}g8$
 $6 \underline{R}g1 \underline{R}e8$
 $\underline{R}g1 \underline{R}e8$

```

\setfigstyle{\itshape\underline}
\king e1, \queen d4, \rook, \
\usetextfig
\bishop, \knight, \pawn

\newgame
\usesymfig
\mainline{1. Nf3 Nf6 2. Rg1 Rg8}\
\usetextfig
\variation{6. Rg1 Re8}\
\wmove{Rg1 Re8}

```

5 Using and changing the board font

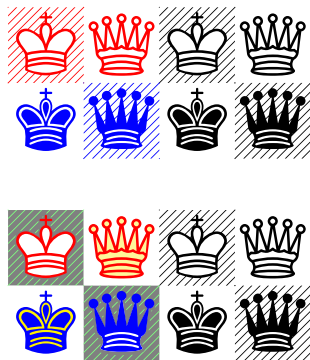
I don't think that a "normal" user would want to use the board fonts directly to print chess boards. It is a lot of tedious and errorprone work to make a board. The printing should be done with `\showboard` from the package `skak` or `\chessboard` from `chessboard.sty`. So the following examples and explanations are meant to be background informations and not real world examples.

The first version of the package `chessfss` handled board fonts with "PieceOnWhite"-chars $\text{\textcircled{K}}$ and "PieceOnBlack"-chars $\text{\textcircled{K}}$. Most of the free fonts for chessboards use such chars. But professional fonts often compose a field from parts like fieldmasks, piecemasks and pieces. In such fonts it is possible to color the different parts individually.

After the release of `chessboard` Ulrich Dirr added to the `skaknew` fonts a complete set of such basic chars and I added support for them. It is quite easy to implement in \LaTeX a method to use composed chars and to switch to another "composition" or to simple chars as it is already part of NFSS: in OT1 e.g. a \u is composed by putting dots above an "u", and you switch to an char "ü" simply by using another encoding like T1.

5.1 Boards without the package `chessfss`

As with the figurines you can use chessfonts without the package `chessfss`. You only must load the encoding definition(s). If you want to be able to color the fields and to use composed field chars you must use the commands: simple ASCII input like JQ can not be intercepted by the encoding commands.



```

\documentclass[parskip]{scrartcl}
\usepackage[LSB,LSBC3,T1]{fontenc}
\usepackage{xcolor}
\makeatletter
\newcommand\smalltestboard{%
{\par
\fontsize{20pt}{20pt}\selectfont
\cfss@WhiteKingOnBlack
\cfss@WhiteQueenOnWhite JQ\
\cfss@BlackKingOnWhite
\cfss@BlackQueenOnBlack kl\par}}

\def\cfss@whitepiececolor{\color{red}}
\def\cfss@whiteonwhitepiecemaskcolor{\color{yellow!50}}
\def\cfss@blackpiececolor{\color{blue}}
\def\cfss@blackonwhitepiecemaskcolor{\color{yellow}}
\def\cfss@blackfieldcolor{\color{green!50}}

\pagestyle{empty}
\begin{document}
\fontencoding{LSB}%
\smalltestboard

\fontencoding{LSBC3}%
\smalltestboard
\end{document}

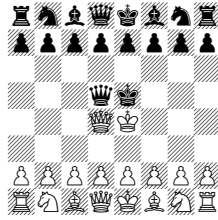
```

5.2 Simple command to use and change board fonts

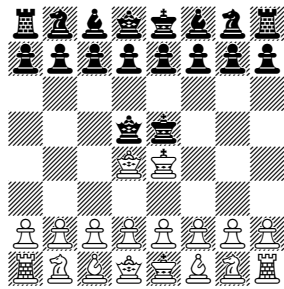
In most cases a user wants diagrams to have always the same size and look. So `\boardfont` and `\boardsymbol{<text>}` sets *all* font properties to the values stored in `\cfss@boardfontfamily`, `\cfss@boardfontseries`, `\cfss@boardfontsize` and `\cfss@boardfontencoding`.

The commands `\setboardfontfamily{<family>}`, `\setboardfontseries{<series>}`, `\setboardfontsize{<size>}` and `\setboardfontencoding` change these values. `\setboardfontsize{<size>}` sets the font size to the value `{<size>}` and the baselineskip to the total height of a square (there exists fonts where the square a not really square). The commands only set the values, they don't actually change the font. You need explicitly or implicitly a `\boardfont` command.

There isn't a `\setboardfontshape{<shape>}` command as I couldn't see what an `\itshape` board should be. But if the need arise I could add it.



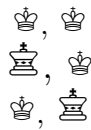
```
\setboardfontsize{10pt}%
\newcommand\testboard{
  \boardfont
  \noindent
  rmbkns\\
  opopop\\
  OZOZOZO\\
  ZOZqjOZO\\
  OZOLKZO\\
  ZOZOZOZO\\
  POPOPOPO\\
  SNAQJBMR\par}}
\testboard
```



```
\setboardfontsize{13pt}
\setboardfontfamily{maya}
\testboard
```

`\WhiteKingOnWhite` For each field of the board there is a command like `\WhiteKingOnWhite`. See table 2 on page 24 for all the names.

`\cfss@WhiteKingOnWhite` For each piece there is also an internal command. As in the case of the figurines `\cfss@WhiteKingOnWhite` and `\WhiteKingOnWhite` are not equal. The first is declared through `\DeclareTextSymbol` in the encoding file while the second is a normal macro which first set most font properties and then call the internal command. The difference can be seen in the following example.



```
\makeatletter
\setboardfontsize{11pt}
\cfss@WhiteKingOnWhite, \WhiteKingOnWhite

{\fontfamily{maya}\Large\selectfont
\cfss@WhiteKingOnWhite, \WhiteKingOnWhite}

{\setboardfontsize{14pt}%
\setboardfontfamily{maya}%
\cfss@WhiteKingOnWhite, \WhiteKingOnWhite}
```

`\getsquaresize` In a simple life, the squares of the board would be square and the fontsize would be also the size of one square. In real life you shouldn't take for granted that in every font a square of the board is really square (that is that the length and the total height are equal), that the length of the square is exactly equal to the nominal font size, and that the baseline of a square is at the bottom.

`\cfss@getsquaresize`

`\len@cfss@squarewidth`

`\len@cfss@squaretotalheight`

`\len@cfss@squaredepth`

The package `chessfss` allocates three length registers called `\len@cfss@squarewidth`, `\len@cfss@squaretotalheight` and `\len@cfss@squaredepth`.

`\getsquarewidth{<ReferencePiece>}` will store in this length registers the width, the totalheight (that is the sum of the height and the depth) and the depth of the object given by the argument. In most cases `{<ReferencePiece>}` should be the black empty square. `\setboardfontsize` e.g. calls `\getsquarewidth{\BlackEmptySquare}`.

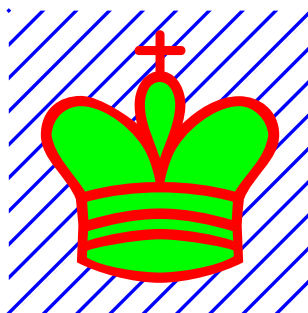
If you want to make sure, that a piece sits on the baseline, you should raise it e.g. with `\raisebox{\depth}{<PIECE>}`

5.3 Layers of Layers of Chars: Using and coloring composed chars

Coloring the simple board pieces doesn't work good as it colors everything: the field and the piece. To be able to use different colors one must compose a field char. I have defined different variants for such compositions. To be able to use them you must/should:








- use command names to insert the chars. The `\testboard` command used above will not work, or
- use a package like `chessboard` to print the board (`\showboard` of the package `skak` will not work)
- load a color package, as some of the masks are black by default and must get colors,
- have the newest version of the `skaknew` fonts (until now they are the only fonts that can use every encoding)
- load the encoding definition you want to use with `fontenc` (the package `chessfss` loads as default only the definition for the encoding `LSB`).

There are “good” compositions with `piecemasks` and `fieldmasks` and there are a bit dubious ones that use the normal chars of the `LSB`-encoding. I named the good ones that need an extended character set `LSBC1`, `LSBC2` etc. I named the other ones (that can be used by every font that works with `LSB`) `LSB1`, `LSB2`



5.4 The logical layers

I distinguish for different logical layers in the encodings.

<i>name of layer</i>	<i>examples</i>	<i>description/comment</i>
fieldmask	 	Solid and opaque. Blocks out the background. The default color of this char is black, and it should be used for white and black pieces and fields, so using correct colors is a must!
field	 	the transparent field with the small diagonals.
piecemask		solid and opaque, can be used to block out the background or field lines in the inner of a piece.
piece	 	font chars.

5.5 The predefined encodings











`\showchessboardencoding`

With `\showchessboardencoding[⟨fontfamily⟩]{⟨encoding⟩}` you get a small tabular that shows the characters used in the encoding and to which layer they belong. In the following tabulars of the predefined encodings the default colors of the basic chars are used. To see the effect of other colors use the command `\setboardfontcolors` described on page 20 before `\showchessboardencoding`.













As you see if you look at the “result” columns it is quite possible that you can’t distinguish the black and white pieces or the pieces on white fields and the pieces on the black fields. In such cases it is your duty to add colors to the pieces or the fields – either manually or by using e.g. the commands of the package `chessboard`.

LSB









The LSB encoding uses only one layer, but two color commands: first the field color than the piece color.

Encoding LSB					
Layer:	fieldmask	field	piecemask	piece	result
WhiteSquare		(color)			
BlackSquare		(color)			
WhiteOnWhite		(color)			
WhiteOnBlack		(color)			
BlackOnWhite		(color)			
BlackOnBlack		(color)			









LSB1

Encoding LSB1					
Layer:	fieldmask	field	piecemask	piece	result
WhiteSquare					
BlackSquare					
WhiteOnWhite					
WhiteOnBlack					
BlackOnWhite					
BlackOnBlack					

















LSB2

Encoding LSB2					
fields have no color!					
Layer:	fieldmask	field	piecemask	piece	result
WhiteSquare					
BlackSquare					
WhiteOnWhite					
WhiteOnBlack					
BlackOnWhite					
BlackOnBlack					




















LSB3

Encoding LSB3					
fields have no char or color!					
White and black pieces use the same char!					
Layer:	fieldmask	field	piecemask	piece	result
WhiteSquare					
BlackSquare					
WhiteOnWhite					
WhiteOnBlack					
BlackOnWhite					
BlackOnBlack					
























LSBC1

Encoding LSBC1					
Layer:	fieldmask	field	piecemask	piece	result
WhiteSquare					
BlackSquare					
WhiteOnWhite					
WhiteOnBlack					
BlackOnWhite					
BlackOnBlack					

















LSBC2

Encoding LSBC2					
Layer:	fieldmask	field	piecemask	piece	result
WhiteSquare					
BlackSquare					
WhiteOnWhite					
WhiteOnBlack					
BlackOnWhite					
BlackOnBlack					













LSBC3

Encoding LSBC3					
Layer:	fieldmask	field	piecemask	piece	result
WhiteSquare					
BlackSquare					
WhiteOnWhite					
WhiteOnBlack					
BlackOnWhite					
BlackOnBlack					

LSBC4

Encoding LSBC4					
Layer:	fieldmask	field	piecemask	piece	result
WhiteSquare					
BlackSquare					
WhiteOnWhite					
WhiteOnBlack					
BlackOnWhite					
BlackOnBlack					

LSBC5

Encoding LSBC5					
Layer:	fieldmask	field	piecemask	piece	result
WhiteSquare					
BlackSquare					
WhiteOnWhite					
WhiteOnBlack					
BlackOnWhite					
BlackOnBlack					

5.6 Coloring the layers

To each of the four layers I associated color commands that are used in the definition of the composed chars and which can be used to change the color of the char used in this layer:

<i>layer</i>	<i>color commands</i>	<i>default definition</i>
fieldmask	<code>\cfss@whitefieldmaskcolor</code>	<code>\color{white}</code>
	<code>\cfss@blackfieldmaskcolor</code>	<code>\color{gray}</code>
field	<code>\cfss@whitefieldcolor</code>	<code>{}</code>
	<code>\cfss@blackfieldcolor</code>	<code>{}</code>
piecemask	<code>\cfss@whiteonwhitepiecemaskcolor</code>	<code>\color{white}</code>
	<code>\cfss@whiteonblackpiecemaskcolor</code>	<code>\color{white}</code>
	<code>\cfss@blackonwhitepiecemaskcolor</code>	<code>\color{white}</code>
	<code>\cfss@blackonblackpiecemaskcolor</code>	<code>\color{white}</code>
piece	<code>\cfss@whitepiececolor</code>	<code>{}</code>
	<code>\cfss@blackpiececolor</code>	<code>{}</code>

`\setboardfontcolors`

With this command you can change the colors in the different layers. The command use a keyval syntax.



```

\makeatletter
\setboardfontsize{30pt}%
\setboardfontencoding{LSBC3}%
\setboardfontcolors{
  blackfieldmask=blue!30,
  whitefieldmask=yellow!50,
  blackfield=blue,
  whiteonwhitepiecemask=red!50,
  whitepiece=green,
  blackpiece=gray}
\newcommand\smalltestboard{{%
  \boardfont
  \noindent
  \cfss@BlackRookOnBlack
  \cfss@BlackKingOnWhite
  \cfss@BlackQueenOnBlack\
  \cfss@WhiteRookOnWhite
  \cfss@WhiteKingOnBlack
  \cfss@WhiteQueenOnWhite\par}}%
\makeatother
\smalltestboard

\bigskip
\setboardfontencoding{LSBC4}
\setboardfontfamily{lucena}
\smalltestboard

```

6 Using and changing the informant font

As for the figurines and for the board characters I declared an encoding for the informant symbols. But to tell the truth: this isn't really an encoding yet. You can't use it with other fonts as I put simply all the remainder chars of the skak fonts in it and so it contains symbols that are unique to the skak and skaknew font.

I plan to sort the symbols a bit in different sets (as textcomp is doing it). I will have to decide which symbols should be in this encoding. The two boardmarkers e.g. certainly aren't a must and should better be defined to pick up the characters directly from the skak fonts. On the other side symbols like ! and !? or \novelty (N) (which in the package skak is defined as \textsf{N}) should perhaps be part of the encoding.

In the meantime you should consider the whole thing as under construction.

<pre> \inffont \infsymbol \cfss@inffontfamily \setinffontfamily \wbetter \cfss@wbetter </pre>	<p>The “informant symbols” are symbols to comment a game. \inffont and \infsymbol{<i>text</i>} change the encoding to LSI and the fontfamily to \cfss@inffontfamily.</p> <p>\cfss@inffontfamily can be changed with the command \setinffontfamily{<i>family</i>}.</p> <p>For each piece there is a command to call the symbol directly, see table 3 on page 25.</p> <p>For each Symbol there is also an internal command. Please read the explications for the</p>
---	--

figurines and the boardsymbol to understand the difference between the internal and the external commands.

$\rightarrow\uparrow\triangle$, $\rightarrow\uparrow\triangle$, \rightarrow , \uparrow , \triangle

`{\inffont ACE}, \infsymbol{ACE},
\withattack, \withinit, \withidea`

Future plan: the NAG's

In PGN (the standard format to store chess games) some of the informant symbols can be coded with NAG's⁷. These consist of a dollar sign followed by a number. I plan to add to the package chessfss commands like `\@namedef{ $\$1$ }{!}` so that package writers can use `\@nameuse{ $\$1$ }` while parsing a pgn.

7 Other font changes

7.1 Using and changing the font on the side of the board

`\sidefont`
`\sidesymbol`
`\cfss@sidefontencoding`
`\cfss@sidefontfamily`
`\cfss@sidefontseries`
`\cfss@sidefontshape`
`\cfss@sidefontsize`
`\setsidefontencoding`
`\setsidefontfamily`
`\setsidefontseries`
`\setsidefontshape`
`\setsidefontsize`

The package skak uses the commands `\notationOn` and `\notationOff` to enable the letters and numbers on the side of a board. So at first I called the commands to control this font e.g. `\notationfont`. Then I changed the names because the font could be confused with the font for the notation of the game (and because the commands were to long to get in the margins of this documentation).

Like the board font this font should be a fix one. So every font property is set by `\sidefont` and `\sidesymbol{text}`. It's up to the package writers to use these commands in macros like `\showboard!`

Some board fonts don't set the squares on the baseline. Package writers should therefore offer commands to move the notations up and down to get around this deficiency.

7.2 Changing the text font

As a default the package skak and texmate.sty use for the notation of a game the normal text font which is active at the start of a game. Each package offer commands to change this font. I don't know yet if I should define text commands e.g. like `\chessifont`, `\chessiifont`.

7.3 Changing more than one font

`\setchessfontfamily`
`\setallchessfontfamily`

If you want to change both the board and the figurine font to the same new family, you can use `\setchessfontfamily{<family>}`. To change all three families use `\setallchessfontfamily{<family>}`.

8 Miscellaneous commands







The package `chessfss` defines some miscellaneous symbols commands. It is quite possible that they will in part go sometimes in the LSI encoding (and that symbols from LSI will go in this section).

`\castlingchar`
`\longcastling`
`\shortcastling`
`\novelty`
`\chesscomment`
`\various`

Symbol	commands
O	<code>\castlingchar</code>
O-O	<code>\shortcastling</code>
O-O-O	<code>\longcastling</code>
N	<code>\novelty</code>
RR	<code>\chesscomment</code> ⁸
R	<code>\various</code>

9 Character and command tables of the encodings

Table 1: The characters in the LSF-encoding (figurine)

Symbol	ASCII	commands
	K	<code>\symking</code> , <code>\cfss@symking</code>
	Q	<code>\symqueen</code> , <code>\cfss@symqueen</code>
	R	<code>\symrook</code> , <code>\cfss@symrook</code>
	B	<code>\symbishop</code> , <code>\cfss@symbishop</code>
	N	<code>\symknight</code> , <code>\cfss@symknight</code>
	p	<code>\sympawn</code> , <code>\cfss@sympawn</code>

⁷Numeric Annotation Glyphs

⁸A copy of this command exists under the name `\comment`. But this can give a clash with various verbatim packages.

Table 2: The characters in the LSB-encoding (board)


























Symbol	ASCII	Description	Command
	0	empty white square. A character that is invisible, but it's there. But it isn't really a square, it only has a width: —	<code>\WhiteEmptySquare</code>
	Z	empty black square.	<code>\BlackEmptySquare</code>
	K	white king on white	<code>\WhiteKingOnWhite</code>
	k	black king on white	<code>\BlackKingOnWhite</code>
	J	white king on black	<code>\WhiteKingOnBlack</code>
	j	black king on black	<code>\BlackKingOnBlack</code>
	Q	white queen on white	<code>\WhiteQueenOnWhite</code>
	q	black queen on white	<code>\BlackQueenOnWhite</code>
	L	white queen on black	<code>\WhiteQueenOnBlack</code>
	l	black queen on black	<code>\BlackQueenOnBlack</code>
	R	white rook on white	<code>\WhiteRookOnWhite</code>
	r	black rook on white	<code>\BlackRookOnWhite</code>
	S	white rook on black	<code>\WhiteRookOnBlack</code>
	s	black rook on black	<code>\BlackRookOnBlack</code>
	B	white bishop on white	<code>\WhiteBishopOnWhite</code>
	b	black bishop on white	<code>\BlackBishopOnWhite</code>
	A	white bishop on black	<code>\WhiteBishopOnBlack</code>
	a	black bishop on black	<code>\BlackBishopOnBlack</code>
	N	white knight on white	<code>\WhiteKnightOnWhite</code>
	n	black knight on white	<code>\BlackKnightOnWhite</code>
	M	white knight on black	<code>\WhiteKnightOnBlack</code>
	m	black knight on black	<code>\BlackKnightOnBlack</code>
	P	white pawn on white	<code>\WhitePawnOnWhite</code>
	p	black pawn on white	<code>\BlackPawnOnWhite</code>
	O	white pawn on black	<code>\WhitePawnOnBlack</code>
	o	black pawn on black	<code>\BlackPawnOnBlack</code>

Table 3: The characters in the LSI-encoding (informer symbols)^a

Symbol	ASCII	command,	Symbol	ASCII	command
+	+	<code>\checksymbol</code>	±	c	<code>\wupperhand</code>
-	-	<code>\castlinghyphen,</code> used only in O-O	⊙	d	<code>\doublepawns</code>
→	A	<code>\withattack</code>	⊖	e	<code>\bupperhand</code>
↑	C	<code>\withinit</code>	±	f	<code>\wbetter</code>
⊙	D	<code>\zugzwang</code>	⊖	g	<code>\bbetter</code>
△	E	<code>\withidea</code>	+−	h	<code>\wdecisive</code>
□	F	<code>\onlymove</code>	−+	i	<code>\bdecisive</code>
↗	G	<code>\diagonal</code>	=	j	<code>\equal</code>
↔	H	<code>\file</code>	∞	k	<code>\unclear</code>
⊕	I	<code>\centre</code>	—	l	<code>\chessee</code>
×	J	<code>\weakpt</code>	#	m	<code>\mate</code>
⊥	L	<code>\ending</code>	∞	n	<code>\compensation</code>
«	M	<code>\qside</code>	♠	o	<code>\opposbishops</code>
»	O	<code>\kside</code>	○●○	q	<code>\seppawns</code>
	P	<code>\etc</code>	♠	r	<code>\passedpawn</code>
>	S	<code>\morepawns</code>	♠	s	<code>\samebishops</code>
⊕	T	<code>\timelimit</code>	⊙	t	<code>\devadvantage</code>
○	U	<code>\moreroom</code>	∞	u	<code>\unitedpawns</code>
↔	V	<code>\counterplay</code>	┌	v	<code>\with</code>
×	X	<code>\capturesymbol</code>	└	w	<code>\without</code>
♠	a	<code>\bishoppair</code>	×	x	(boardmarker)
△	b	<code>\betteris</code>	○	y	(boardmarker)

^aRemark: You shouldn't rely on the fact that other informer fonts that you may find somewhere have exactly the same characters.

10 Converting and installing a ttf-chessfont or a type1-chessfont for use with the package skak

In the following I describe how to convert ttf-chessfont to true type. In the mean time I have put a lot of the files for chessfonts on CTAN. You find them in the folder `/fonts/chess/enpassant`. I wasn't able to reach every author of the fonts, so I couldn't each the `.pfb` of each font. For this fonts you will have to do the conversion yourself. But at least you don't have to write all the `.enc` and `.fd`-files.

10.1 Attention

1. The fonts come with various character sets. Some have board and figurine symbols in one file, some put them in several files, some have borders for the diagramm, some has some other symbols. I tried up to now neither to get rid of the symbols not needed in the package `skak`, nor to access the non-standard symbols.
2. Not every font worked for me. E.g. I wasn't able to convert an old font that came with `chessbase`.
3. I'm using Windows. I'm quite sure it should work on other OS's too, but I have no idea how.
4. I have no idea if the quality of the fonts I got is good and if all this couldn't be done better.
5. I'm using a rather long naming scheme (I'll describe it below). I did it because I would have gone mad with only 8 characters. If you need a 8.3-naming scheme, invent it.
6. I'm only describing how to get figurine and board characters. I hadn't had the time yet to look for other informantor fonts.
7. I converted mostly free fonts that I found at <http://www.enpassant.dk/>. I'm quite sure that it is no problem to use them in private documents. But I won't distribute the `pfb` without the consent of the authors. If you want to use the fonts for a commercial project, you should perhaps ask the authors first.

10.2 The naming scheme

I used a naming scheme with long filenames that tries more or less to follow the Karl Berry scheme.

All my fontnames start with `chess-` (the "foundry"). Then follows the name of the font family e.g. `alfonso-` and a string to describe the fontseries and the fontshape e.g. `bit-` for a bold and italic font (the defaults `m` (medium) and `n` (normal) are omitted).

Then follows a string for the encoding: `lsf`, `lsb` and `lsi` for the skak-encoded fonts. Fonts that have additional chars like piecemasks or fieldmask (or where chars from the standard LSB-set are missing) gets at the end a binary number: 1 indicates that char set exists in the font, 0 that is missing. The char sets are (in this order): solid field mask, transparent field mask, piecemask, pieces on black fields. So a standard LSB-font should get the endnumber -0001 (which I omit). The new skakfonts would get the ending 1111, the lucena fonts has the ending 1001, and a know of a fritz font with the characters 1000 (no pieces on black fields).

For the raw fonts I used the keyword `raw` together with some information about the character set, e.g. `board-fig-raw` for a font with board and figurines characters. All fonts that I converted are scalable so I didn't need to put a size information in the name, but if needed one should put it at the very end.

10.2.1 Example

The board font of the skak font (`skak10`) would have the name `chess-skak-lsb-10.tfm` in my naming scheme.

The fonts (`skakf10` and `skakf10b`) are a bit more difficult. They contain figurines and informant symbols, so from the point of view of the package `skak` each represent two fonts and should have two names: `chess-skak-lsf-10.tfm/chess-skak-lsi-10.tfm` and `chess-skak-b-lsf-10.tfm/chess-skak-b-lsi-10.tfm`. One could implement this by copying the `tfm` files, but I don't think one should do it only for the sake of the purity of the naming scheme. So I stuck on the original names of this family.

10.3 Testing of the fonts

To test the raw and the reencoded fonts, you should get acquainted with `nfssfont.tex` which you can find on your system. It is quite easy: Compile the document, it will ask you for a font, type the name of the `tfm`-file (without the ending) and then "enter". Then type `\table\bye` to get a fonttable.

The main problem with `nfssfont.tex` is that it doesn't show you invisible characters like the empty white field. To get small boxes around each character, save the file under another name. Then insert in the definition of `\def\:` an `\fbox{}` command:

```
\def\:{\setbox0=\hbox{\fbox{\char\n}}
```

When you install and test a new board font, you should make sure, that you test all characters. The `\testboard` from subsection 5 ist quite useful in this respect.

If you get errors you can use these rules of thump: If `LATEX` or `pdfLATEX` complain during the setting of the document about problems with the font, there is in most cases a problem with the `tfm` files. If `dvips` complains (or `pdfLATEX` at the end while inserting

the `pfb` files) there is a problem with the `map` files or the `pfb` files. If YAP complains, it doesn't find the `afm` files.

10.4 Quick installation guide

Basically you have to do:

1. Convert a `ttf`-font to a `pfb`-font.
2. Reencode the font, so that the figurine characters are at the correct position and/or reencode the font, so that the board characters are at the correct position.
3. Make reencoded `tfm`-files, and store everything where it belong,
4. Tell the drivers where to find the `pfb`-font (and how to reencode it),
5. Tell \LaTeX the encoding and the fontfamily name of the font(s).
6. Update all databases and maps and test the new font.

10.5 A more complete installation guide

Ok, let's start.

10.5.1 Convert a `ttf`-font to a `pfb`-font.

1. Get yourself a working `ttf2pt1`. All I needed to do was to put `ttf2pt1.exe`, `tl80ttf.dll`, `tl80freetype2.dll` in a folder. All files I found on CTAN in some binaries of `fptex`.
2. Get a free chess font. E.g. `alfonso.zip` from <http://www.enpassant.dk/>. `Alfonso` is a font with figurine and board characters, so we have to make two `tfm`-files out of it.
3. Unpack the `zip`-file and put the `ttf`-file `alfoches.ttf` into the folder with `ttf2pt1.exe`.
4. Open a command shell, and switch to this folder. Then use the command

```
ttf2pt1 -b alfoches.ttf chess-alfonso-board-fig-raw
```

If everything worked fine, you should get a `afm` and a `pfb`-file. (I also got a lot of messages that I ignored).

5. Continue with

```
afm2tfm chess-alfonso-board-fig-raw
```

`afm2tfm` is part of your \TeX -installation and so should work out of the box. You will get a `tfm`-file. `afm2tfm` will also tell you the internal fontname `Chess-Alfonso-X` of the font. You will need it for the `map`-file, so note this name somewhere.

10.5.2 Installing the type1-fonts – the brute force method

Reencode the fonts This is the tedious part of the installing process. You need a list that tells the drivers like `dvips` or `pdf \TeX` under which name they can find the requested piece in the fonts. Such lists are stored in `.enc`-files.

So you need two encoding files so that you can push the figurines and the board characters to the correct position for the package `skak`. Sadly the different authors of free chess fonts didn't use always the same internal glyph name for a character. Often they simply used the name of the ASCII-character which is on the position where they decided to put the piece – and different author used different positions. In the appendix you will find `chess-fig.enc` and `chess-board.enc`. You can use them for fonts from the author of our example font, Armando H. Marroquin. If they don't work for another font, you should open the `afm`-file and try to find out under which name the characters are stored and change the glyph names in the `enc`-files accordingly.

Some board fonts don't have a real empty white square. In the package `skak` you can define `\renewcommand\WhiteEmptySquare{\rule{\squarelength}{0pt}}` in your document as a workaround. But a better solution is to use a virtual font to repair the font.

Attention!!!

In `.enc`-files encodings are surrounded by the name declaration of the encoding:

```
/EncodingName[ ... ] def.
```

I have never tried if is possible to put more than one encoding definition in one `.enc`-file. But I have learn, that `dvips` don't like it, if you use the same encoding name in two files: `dvips` doesn't load the new file but simply reuse the old defined encoding. So you must make sure that every encoding has an unique name.

Make reencoded tfm-files Now you need to call (each command on one line)

```
afm2tfm chess-alfonso-board-fig-raw.afm
-T chess-fig.enc chess-alfonso-lsf.tfm
```

and

```
afm2tfm chess-alfonso-board-fig-raw.afm
-T chess-board.enc chess-alfonso-lsb.tfm
```

Then copy all the `tfm`, `afm` and `pfb`-files where they belong in your \TeX -system (see the FAQ-pointers in 3.5).

Change the map-files Open a new (if this is your first font) `texdocument` in your editor, write on three lines (ignore the line breaks)

```
chess-alfonso-board-fig-raw Chess-Alfonso-X
<chess-alfonso-board-fig-raw.pfb

chess-alfonso-lsf Chess-Alfonso-X " ChessFigEncoding ReEncodeFont "
<chess-fig.enc <chess-alfonso-board-fig-raw.pfb

chess-alfonso-lsb Chess-Alfonso-X " ChessBoardEncoding ReEncodeFont "
<chess-board.enc <chess-alfonso-board-fig-raw.pfb
```

in this document and save it, e.g. as `chess.map` to a folder where your \TeX -system looks for fontmaps. (`/localtexmf/dvips/...` is a good place).

Then add `chess.map` to your local `updmap.cfg`.

Tell \LaTeX about the new fonts Informations about fonts are stored normally in `<encoding><fontfamily>.fd` files. As the package `chessfss` use three different encodings which only one or two fonts in each family, you would get a lot of small `fd` files. To make life easier, the package `chessfss` inputs at the end of the package a file `chessfss.cfg` if it exists. An example `chessfss-ex.cfg` comes with the package.

So either open a new `texdocument` (or an existing `chessfss.cfg`), add the lines

```
%%% alfonso %%%
\DeclareFontFamily{LSF}{alfonso}{}
\DeclareFontShape{LSF}{alfonso}{m}{n}{<-> chess-alfonso-lsf}{}

\DeclareFontFamily{LSB}{alfonso}{}
\DeclareFontShape{LSB}{alfonso}{m}{n}{<-> chess-alfonso-lsb}{}

```

and save the document as `chessfss.cfg` e.g. in `localtexmf/tex/latex/chessfss/`. Or put the line in files named `lsfalfonso.fd` and `lsbalfonso.fd`.

If you want to use the composed encodings `LSB1`, `LSB2` and `LSB3` too, you should put the following lines (here for `LSB1`, adapt for the other encodings) in files `lsb1alfonso.fd`, `lsb2alfonso.fd` or `lsb3alfonso.fd`⁹:

⁹Don't put them in `chessfss.cfg`, the experimental encodings aren't declared by default in the package `chessfss`

```
\DeclareFontFamily{LSB1}{alfonso}{}  
\DeclareFontShape{LSB1}{alfonso}{m}{n}{<-> chess-alfonso-lsb}{}
```

10.5.3 Update all databases and maps and test the new fonts

Renew your filename database e.g. for MikTeX with `initexmf -u` and call `updmap`, e.g. with `initexmf --mkmaps`.

Try `nfssfnt.tex` on the fonts `chess-alfonso-board-fig-raw` and `chess-alfonso-lsb` to understand how the encoding vector changes the character set and the order of the fonts.

Try the new fontfamily by using `\setchessfontfamily{alfonso}` in a test document.

You will get a warning during the test, when you use `\mainline`. E.g.

```
LaTeX Font Warning: Font shape 'LSF/alfonso/bx/n' undefined (Font)  
using 'LSF/alfonso/m/n' instead on input line 9.
```

As already mentioned, there is nothing to worry about. If you don't like such warnings, learn (e.g. by reading `fntguide.tex` or the *L^AT_EX Companion*) which `\DeclareFontShape` command you need to get silence substitutions.

10.5.4 Installing a type1 chess font with fontinst

The brute force method has some faults: One can't correct missing glyph like the empty square. In some board fonts the square have other baselines than in the `skak-fonts` which can disturb board building commands. And perhaps you would like to adjust some glyphs e.g. make them a bit larger.

All this can be corrected by using virtual fonts and the package `fontinst` for the installation. One needs files `lsf.etx`, `lsb.etx` and `lsi.etx` that describes the encoding, `mtx`-files that describes the adjustments and then one should use a `tex` file with commands like

```
\installfont{<fontname>}{<fromafm>,<mtx>}{<enc>}{<enc>}{<family>}  
{<series>}{<shape>}{}
```

I haven't yet done it for a full font. And I will have to understand yet how I should handle the different names for the same glyph. So I will add more detailed instructions later.

10.6 chess-fig-example-enc.enc – an example

You must replace the asterixes by the name of the chars or `/.notdef`

```
/Chess***FigEncoding[
%0x
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%1x
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%2x
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%3x
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%4x
/.notdef /.notdef
%% Bishop
/***
/.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef
%% King
/*** /.notdef /.notdef
%% Knight
/*** /.notdef
%5x
/.notdef
%% Queen
/***
%%Rook
/*** /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%6x
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%7x Pawn
/*** /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%8x
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%9x
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%Ax
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%Bx
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%Cx
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
```



```

/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%Dx
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%Dx
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%Ex
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
] def

```

10.7 chess-board-example-enc.enc – an example

You must replace the asterixes by the name of the chars or /.notdef

```

/Chess***BoardEncoding[
%0x
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%1x
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%2x
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%3x
%WhiteSquare
/***
%solid king fieldmask
/***
%solid knight fieldmask
/***
%solid rook fieldmask
/***
%solid bishop fieldmask
/***
%solid queen fieldmask
/***
%solid pawn fieldmask
/***
/.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%4x
/.notdef
%WBishopOnBlack
/***
%WBishopOnWhite
/***
/.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef
%WKingOnBlack

```

```

/***
%WKingOnWhite
/***
%WQueenOnBlack
/***
%WKnightOnBlack
/***
%WKnightOnWhite
/***
%WPawnOnBlack
/***
%5x
%WPawnOnWhite
/***
%WQueenOnWhite
/***
%WRookOnWhite
/***
%WRookOnBlack
/***
%king piecemask
/***
%knight piecemask
/***
%rook piecemask
/***
%bishop piecemask
/***
%queen piecemask
/***
%pawn piecemask
/***
%BlackSquare
/***
/.notdef /.notdef /.notdef /.notdef /.notdef
%6x
/.notdef
%BBishopOnBlack
/***
%BBishopOnWhite
/***
/.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef
%BKingOnBlack
/***
%BKingOnWhite
/***
%BQueenOnBlack
/***
%BKnightOnBlack
/***
%BKnightOnWhite
/***
%BPawnOnBlack

```

```

/****
%7x
%BPawnOnWhite
/****
%BQueenOnWhite
/****
%BRookOnWhite
/****
%BRookOnBlack
/****
/.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef
%solid black fieldmask
/****
/.notdef /.notdef /.notdef /.notdef /.notdef
%8x
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%9x
/.notdef /.notdef /.notdef /.notdef /.notdef
%king fieldmask
/****
%knight fieldmask
/****
%rook fieldmask
/****
%bishop fieldmask
/****
%queen fieldmask
/****
%pawn fieldmask
/****
/.notdef /.notdef /.notdef /.notdef /.notdef
%Ax
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%Bx
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%Cx
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%Dx
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%Ex
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
] def

```

11 Fontnames and other informations

I here list every font I installed up to now, along with all names needed (the internal fontname, the name used in TeX), and the needed encoding vector. With the exception of the original font `skak` all are type1 fonts mostly converted from true type fonts.

Alfonso

source: <http://www.enpassant.dk/chess/download/alfonso.zip>
author: Armando H. Marroquin
characters: figurine and board symbols
familyname: alfonso
internal fontname: `Chess-Alfonso-X`
fonts: The following fonts for the package `chessfss` can be made from this source:

encoding	serie	tfm-name	reencoding command for <code>chess.map</code>
raw	-	<code>chess-alfonso-board-fig-raw</code>	none
LSF	m	<code>chess-alfonso-lsf</code>	" ChessFigEncoding ReEncodeFont " <chess-fig.enc
LSB	m	<code>chess-alfonso-lsb</code>	" ChessBoardEncoding ReEncodeFont " <chess-board.enc

Alpha

source: <http://www.enpassant.dk/chess/down1/alpha.zip>
author: Eric Bentzen
characters: figurine and board symbols, the font also has black figurines. I decided to put them under the fontshape (*not* fontseries!) bl. The boardfont doesn't have a square character, so you should redefine \WhiteEmptySquare or \cfss@WhiteEmptySquare to something sensible before using this font (or ignore the messages about the missing 0).
familyname: alpha
internal fontname: Chess-Alpha
fonts: The following fonts for the package chessfss can be made from this source:

encoding	serie	shape	tfm-name	reencoding command for chess.map
raw	-		chess-alpha-board-fig-raw	none
LSF	m	n	chess-alpha-lsf	" ChessFigEncoding ReEncodeFont " <chess-alpha-fig.enc
LSF	m	bl	chess-alpha-bl-lsf	" ChessFigEncoding ReEncodeFont " <chess-alpha-black-fig.enc
LSB	m	n	chess-alpha-lsb	" ChessBoardEncoding ReEncodeFont " <chess-alpha-board.enc

Aventurer

source: <http://www.enpassant.dk/chess/down1/adventur.zip>
author: Armando H. Marroquin
characters: figurine and board symbols
familyname: aventurer
internal fontname: ChessAventurer
fonts: The following fonts for the package chessfss can be made from this source:

encoding	serie	tfm-name	reencoding command for chess.map
raw	-	chess-aventurer-board-fig-raw	none
LSF	m	chess-aventurer-lsf	" ChessFigEncoding ReEncodeFont " <chess-fig.enc
LSB	m	chess-aventurer-lsb	" ChessBoardEncoding ReEncodeFont " <chess-board.enc

Berlin

source: <http://www.enpassant.dk/chess/down1/berlin.zip>
author: Eric Bentzen (see also font alpha)
characters: figurine and board symbols, the font also has black figurines. I decided to put them under the fontshape (*not* fontseries!) bl. The font also has characters for fairy chess. The boardfont doesn't have a square character, so you should redefine `\WhiteEmptySquare` or `\cfss@WhiteEmptySquare` to something sensible before using this font (or ignore the messages about the missing 0).
familyname: berlin
internal fontname: Chess-Berlin
fonts: The following fonts for the package chessfss can be made from this source:

encoding	serie	shape	tfm-name	reencoding command for chess.map
raw	–		chess-berlin-board-fig-raw	none
LSF	m	n	chess-berlin-lsf	" ChessFigEncoding ReEncodeFont " <chess-alpha-fig.enc
LSF	m	bl	chess-berlin-b-lsf	" ChessFigEncoding ReEncodeFont " <chess-alpha-black-fig.enc
LSB	m	n	chess-berlin-lsb	" ChessBoardEncoding ReEncodeFont " <chess-berlin-board.enc

Cases

source: <http://www.enpassant.dk/chess/down1/cases.zip>
author: Matthieu Leschemelle
characters: figurine and board symbols
familyname: cases
internal fontname: ChessCases
fonts: The following fonts for the package chessfss can be made from this source:

encoding	serie	tfm-name	reencoding command for chess.map
raw	-	chess-cases-board-fig-raw	none
LSF	m	chess-cases-lsf	" ChessFigEncoding ReEncodeFont " <chess-fig.enc
LSB	m	chess-cases-lsb	" ChessBoardEncoding ReEncodeFont " <chess-board.enc

Cheq

source: various, it's not a ttf- but a type1-font. (I don't remember where I got my version from)
author: Copyright (c) 1989 Adobe Systems Incorporated. All rights reserved.
characters: only board symbols
familyname: cheq
internal fontname: Cheq
fonts: The following fonts for the package chessfss can be made from this source:

encoding	serie	tfm-name	reencoding command for chess.map
raw	-	chess-cheq-board-raw	none
LSB	m	chess-cheq-lsb	" ChessBoardEncoding ReEncodeFont " <chess-cheq-board.enc

Condal

source: <http://www.enpassant.dk/chess/downl/condal.zip>
author: Armando H. Marroquin
characters: figurine and board symbols
familyname: condal
internal fontname: Chess-Condal
fonts: The following fonts for the package chessfss can be made from this source:

encoding	serie	tfm-name	reencoding command for chess.map
raw	-	chess-condal-board-fig-raw	none
LSF	m	chess-condal-lsf	" ChessFigEncoding ReEncodeFont " <chess-fig.enc
LSB	m	chess-condal-lsb	" ChessBoardEncoding ReEncodeFont " <chess-board.enc

Harlequin

source: <http://www.enpassant.dk/chess/downl/harlequi.zip>
author: Armando H. Marroquin
characters: figurine and board symbols
familyname: harlequin
internal fontname: Chess-Harlequin
fonts: The following fonts for the package chessfss can be made from this source:

encoding	serie	tfm-name	reencoding command for chess.map
raw	-	chess-harlequin-board-fig-raw	none
LSF	m	chess-harlequin-lsf	" ChessFigEncoding ReEncodeFont " <chess-fig.enc
LSB	m	chess-harlequin-lsb	" ChessBoardEncoding ReEncodeFont " <chess-board.enc

Kingdom

source: <http://www.enpassant.dk/chess/downl/kingdom.zip>
author: Armando H. Marroquin
characters: figurine and board symbols
familyname: kingdom
internal fontname: Chess-Kingdom
fonts: The following fonts for the package chessfss can be made from this source:

encoding	serie	tfm-name	reencoding command for chess.map
raw	-	chess-kingdom-board-fig-raw	none
LSF	m	chess-kingdom-lsf	" ChessFigEncoding ReEncodeFont " <chess-fig.enc
LSB	m	chess-kingdom-lsb	" ChessBoardEncoding ReEncodeFont " <chess-board.enc

Leipzig

source: <http://www.enpassant.dk/chess/downl/leipzig.zip>
author: Armando H. Marroquin
characters: figurine and board symbols
familyname: leipzig
internal fontname: Chess-Leipzig
fonts: The following fonts for the package chessfss can be made from this source:

encoding	serie	tfm-name	reencoding command for chess.map
raw	-	chess-leipzig-board-fig-raw	none
LSF	m	chess-leipzig-lsf	" ChessFigEncoding ReEncodeFont " <chess-fig.enc
LSB	m	chess-leipzig-lsb	" ChessBoardEncoding ReEncodeFont " <chess-board.enc

Line

source: <http://www.enpassant.dk/chess/downl/chesline.zip>
author: Armando H. Marroquin
characters: figurine and board symbols
familyname: line
internal fontname: **Chess-Line**
fonts: The following fonts for the package chessfss can be made from this source:

encoding	serie	tfm-name	reencoding command for chess.map
raw	-	chess-line-board-fig-raw	none
LSF	m	chess-line-lsf	" ChessFigEncoding ReEncodeFont " <chess-fig.enc
LSB	m	chess-line-lsb	" ChessBoardEncoding ReEncodeFont " <chess-board.enc

Lucena

source: <http://www.enpassant.dk/chess/downl/lucena.zip>
author: Armando H. Marroquin
characters: figurine and board symbols, the font has solid field masks (used in the last example).
familyname: lucena
internal fontname: **ChessLucena**
fonts: The following fonts for the package chessfss can be made from this source:

encoding	serie	tfm-name	reencoding command for chess.map
raw	-	chess-lucena-board-fig-raw	none
LSF	m	chess-lucena-lsf	" ChessFigEncoding ReEncodeFont " <chess-fig.enc
LSB,LSBC4	m	chess-lucena-lsb-1001	" ChessBoardEncoding ReEncodeFont " <chess-board.enc

Magnetic

source: <http://www.enpassant.dk/chess/downl/magnetic.zip>
author: Armando H. Marroquin
characters: figurine and board symbols
familyname: magnetic
internal fontname: Chess-Magnetic
fonts: The following fonts for the package chessfss can be made from this source:

encoding	serie	tfm-name	reencoding command for chess.map
raw	-	chess-magnetic-board-fig-raw	none
LSF	m	chess-magnetic-lsf	" ChessFigEncoding ReEncodeFont " <chess-fig.enc
LSB	m	chess-magnetic-lsb	" ChessBoardEncoding ReEncodeFont " <chess-board.enc

Mark

source: <http://www.enpassant.dk/chess/downl/chesmark.zip>
author: Armando H. Marroquin
characters: figurine and board symbols
familyname: mark
internal fontname: Chess-Mark
fonts: The following fonts for the package chessfss can be made from this source:

encoding	serie	tfm-name	reencoding command for chess.map
raw	-	chess-mark-board-fig-raw	none
LSF	m	chess-mark-lsf	" ChessFigEncoding ReEncodeFont " <chess-fig.enc
LSB	m	chess-mark-lsb	" ChessBoardEncoding ReEncodeFont " <chess-board.enc

Marroquin

source: <http://www.enpassant.dk/chess/downl/marroqui.zip>
author: Armando H. Marroquin
characters: figurine and board symbols
familyname: marroquin
internal fontname: Chess-Marroquin
fonts: The following fonts for the package chessfss can be made from this source:

encoding	serie	tfm-name	reencoding command for chess.map
raw	-	chess-marroquin-board-fig-raw	none
LSF	m	chess-marroquin-lsf	" ChessFigEncoding ReEncodeFont " <chess-fig.enc
LSB	m	chess-marroquin-lsb	" ChessBoardEncoding ReEncodeFont " <chess-board.enc

Maya

source: <http://www.enpassant.dk/chess/downl/chesmaya.zip>
author: Armando H. Marroquin
characters: figurine and board symbols
familyname: maya
internal fontname: Chess-Maya
fonts: The following fonts for the package chessfss can be made from this source:

encoding	serie	tfm-name	reencoding command for chess.map
raw	-	chess-maya-board-fig-raw	none
LSF	m	chess-maya-lsf	" ChessFigEncoding ReEncodeFont " <chess-fig.enc
LSB	m	chess-maya-lsb	" ChessBoardEncoding ReEncodeFont " <chess-board.enc

Mediaeval

source: http://www.enpassant.dk/chess/downl/medie_tt.zip
author: Armando H. Marroquin
characters: figurine and board symbols
familyname: mediaeval
internal fontname: Chess-Mediaeval
fonts: The following fonts for the package chessfss can be made from this source:

encoding	serie	tfm-name	reencoding command for chess.map
raw	-	chess-mediaeval-board-fig-raw	none
LSF	m	chess-mediaeval-lsf	" ChessFigEncoding ReEncodeFont " <chess-fig.enc
LSB	m	chess-mediaeval-lsb	" ChessBoardEncoding ReEncodeFont " <chess-board.enc

Merida

source: http://www.enpassant.dk/chess/downl/merid_tt.zip
author: Armando H. Marroquin
characters: figurine and board symbols
familyname: merida
internal fontname: Chess-Merida
fonts: The following fonts for the package chessfss can be made from this source:

encoding	serie	tfm-name	reencoding command for chess.map
raw	-	chess-merida-board-fig-raw	none
LSF	m	chess-merida-lsf	" ChessFigEncoding ReEncodeFont " <chess-fig.enc
LSB	m	chess-merida-lsb	" ChessBoardEncoding ReEncodeFont " <chess-board.enc

Millennia

source: <http://www.enpassant.dk/chess/downl/millenia.zip>
author: Armando H. Marroquin
characters: figurine and board symbols. From the description also the figurines should exist in a medium and bold version. But both fonts looks quite similar to me.
familyname: millennia
internal fontname: Chess-Millennia-L (medium) and Chess-Millennia-D (bold symbols)
fonts: The following fonts for the package chessfss can be made from this source:

encoding	serie	tfm-name	reencoding command for chess.map
raw	m	chess-millennia-board-fig-raw	none
raw	b	chess-millennia-b-board-fig-raw	none
LSF	m	chess-millennia-lsf	" ChessFigEncoding ReEncodeFont "<chess-fig.enc
LSB	m	chess-millennia-lsb	" ChessBoardEncoding ReEncodeFont " <chess-board.enc
LSB	b	chess-millennia-b-lsb	" ChessBoardEncoding ReEncodeFont " <chess-board.enc

Motif

source: <http://www.enpassant.dk/chess/downl/motif.zip>
author: Armando H. Marroquin
characters: figurine and board symbols
familyname: motif
internal fontname: Chess-Motif
fonts: The following fonts for the package chessfss can be made from this source:

encoding	serie	tfm-name	reencoding command for chess.map
raw	-	chess-motif-board-fig-raw	none
LSF	m	chess-motif-lsf	" ChessFigEncoding ReEncodeFont " <chess-fig.enc
LSB	m	chess-motif-lsb	" ChessBoardEncoding ReEncodeFont " <chess-board.enc

Pirat

source: <http://www.enpassant.dk/chess/downl/pirat.zip>
author: Klaus Wolf
characters: figurine and board symbols, the figurine fonts comes in bold and italic too and contains also some informant symbols and textcharacters.
familyname: pirat
internal fontname: ChessFigurinePirat (pirat.ttf), ChessFigurinePiratBold (piratf.ttf), ChessFigurinePiratItalic (piratk.ttf, piratkf.ttf), ChessDiagrammPirat (piratdia.ttf)
fonts: The following fonts for the package chessfss can be made from this source:

encoding	serie	shape	tfm-name	reencoding command for chess.map
raw	-	-	chess-pirat-fig-raw	none
raw	-	-	chess-pirat-b-fig-raw	none
raw	-	-	chess-pirat-it-fig-raw	none
raw	-	-	chess-pirat-bit-raw	none
LSF	m	n	chess-pirat-lsf	" ChessFigEncoding ReEncodeFont " <chess-pirat-fig.enc
LSF	b, bx	n	chess-pirat-b-lsf	" ChessFigEncoding ReEncodeFont " <chess-pirat-fig.enc
LSF	m	it	chess-pirat-it-lsf	" ChessFigEncoding ReEncodeFont " <chess-pirat-fig.enc
LSF	b, bx	it	chess-pirat-bit-lsf	" ChessFigEncoding ReEncodeFont " <chess-pirat-fig.enc
LSB	m	n	chess-pirat-lsb	" ChessBoardEncoding ReEncodeFont " <chess-pirat-board.enc

Skak

source: <ftp://ftp.dante.de/tex-archive/fonts/chess/skak/mf/>.
skak is mf-font. Over the year there have been made changes to the font, bugs have been corrected and characters added. So you should make sure you have newest version, and that no remains from older versions (**tfm**, **pk**) are on your system.

author: based on original work by Piet Tutelaers, with additions and changes from Torben Hoffman and Dirk Baechle

characters: figurine and board symbols and informant symbols, figurines and the informant symbols exist in medium and bold version.

familyname: skak

internal fontname: none (its not a type1 font).

fonts: The following fonts are in this family:

encoding	serie	tfm-name	reencoding command for chess.map
LSF, LSI	m	skakf10	–
LSF, LSI	b	skakf10b	–
LSB	m	skak10, skak15, skak20, skak30	–

Skaknew

source: <ftp://ftp.dante.de/tex-archive/fonts/chess/skaknew/>

author: Ulrich Dirr (he converted the skak-sources to type1)

characters: figurine and board symbols and informant symbols, figurines and the informant symbols exist in medium and bold version.

familyname: skaknew

internal fontname: SkakNew-Figurine, SkakNew-Figurine-Bold, SkakNew-Diagram, SkakNew-DiagramT

fonts: The following fonts are in this family:

encoding	serie	tfm-name	reencoding command for chess.map
LSF, LSI	m	SkakNew-Figurine	–
LSF, LSI	b	SkakNew-Figurine-Bold	–
LSB, LSBC1-4	m	SkakNew-Diagram SkakNew-DiagramT	– for small diagrams

Skak-k6

source: <http://www.enpassant.dk/chess/download/skak.zip>
author: Egon Madsen
characters: only board symbols. It's a type1-font. The font is too large. To prevent the notation of the board to be misplaced, one has to scaled it down in `chessfss.cfg`:

```
\DeclareFontShape{LSB}{skak-k6}{m}{n}{  
  {<-> s * [0.8] chess-skak-k6-lsb}{}  
}
```

`\ranklift` has to be changed too.
familyname: skak-k6
internal fontname: Skak
fonts: The following fonts are in this family:

encoding	serie	tfm-name	reencoding command for <code>chess.map</code>
raw	-	<code>chess-skak-k6-board-raw</code>	-
LSB	m	<code>chess-skak-k6-lsb</code>	" ChessBoardEncoding ReEncodeFont " < <code>chess-skak-k6-board.enc</code>

Utrecht

source: <ftp://ftp.pitt.edu/group/student-activities/chess/DTP/utrecht.zip>
author: Hans Bodlaender
characters: Board
familyname: utrecht
internal fontname: Chess-Utrecht
fonts: The following fonts for the package `chessfss` can be made from this source:

encoding	serie	tfm-name	reencoding command for <code>chess.map</code>
raw	-	<code>chess-utrecht-board-fig-raw</code>	none
LSB	m	<code>chess-utrecht-lsb</code>	" ChessUtrechtBoardEncoding ReEncodeFont " < <code>chess-utrecht-board.enc</code>
