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 nss ................................ 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
1 Requirements

The package chessfss uses some primitives of e\TeX. 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 informatorsymbol \see to \chessssee 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@sidefont

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 \MikTeX) 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 used them anyway.

### 3.2 Description of the fonts

While printing chess games 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:
  

- The informator characters. these are symbols like ♟ or ♙. 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 and 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 fallback 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 informator symbols:

\begin{verbatim}
\DeclareFontEncoding{LSF}{}{} \\
\DeclareFontSubstitution{LSF}{skaknew}{m}{n} \\
\DeclareFontEncoding{LSI}{}{} \\
\DeclareFontSubstitution{LSI}{skaknew}{m}{n} \\
\DeclareFontEncoding{LSB}{}{} \\
\DeclareFontSubstitution{LSB}{skaknew}{m}{n}
\end{verbatim}

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 “informator 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 font\(^1\). 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

---

\(^1\)One font from the view of \TeX: the characters must be in one tfm file, the glyphs can come from various fonts.
(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. LSA³ 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 skaknew⁴ 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=wherefiles 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 informator 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

\begin{verbatim}
 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.
\end{verbatim}

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 informator 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
\documentclass{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
\end{document}

As the above example shows you don’t need the package \texttt{chessfss} to use chess fonts. You only need to load the encoding definition LSF as an option of \texttt{fontenc} (and you \emph{must} load a text encoding as the last option of \texttt{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 \texttt{\cfss@symking} contains some font switching command but this isn’t true: \LaTeX{} can’t find a definition for \texttt{\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 \texttt{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.

\subsection*{4.2 Using symbols for pieces}

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

\begin{flushright}
June 14, 2006
\end{flushright}
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. E.g. there is a bold font in the skaknew family, but not in the maya family:

\symking Work as expected – if the chess font has the font variants need.

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

\figfont KQRBNp
\bfseries KQRBNp

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.

\cfss@symking Work as expected – if the chess font has the font variants need.

\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: Re1 should be Te1 in german and not Te1.
\settextfigchars{\langle language\rangle}{\langle king\rangle}{\langle queen\rangle}{\langle rook\rangle}{\langle bishop\rangle}{\langle knight\rangle}{\langle pawn\rangle}
defines text output characters for the pieces (e.g. the german abbreviations) for an (optional, default is english) language. 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.

With this command you can switch to the text chars of one defined language.

\textfigsymbol{\langle piecechar\rangle} 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.

\begin{verbatim}
Q, Q
\textfigsymbol{Q}, \textfigsymbol{\textfigsymbol{Q}}
\end{verbatim}

\begin{verbatim}
Ka1, Qg2, Re1, B, N, P
Kc1, Qg1, Rf3, B, N, P
\textking a1, \textqueen g2, \textrook e1, \textbishop, \textknight, \textpawn\textbfseries\\
\textfigsymbol{\textfigsymbol{Q}}, \textfigsymbol{\textfigsymbol{Q}}
\end{verbatim}

\begin{verbatim}
Rc1, Dg1, Te4, A, C, P
X, Y, Z, ü, ö, ä
\settextfiglanguage{italian}
\textking c1, \textqueen g1, \textrook e4, \textbishop, \textknight, \textpawn
\settextfigchars{\langle silly\rangle}{\langle silly\rangle}
\textking, \textqueen, \textrook, \textbishop, \textknight, \textpawn
\end{verbatim}

4.4 Commands that can switch between text and symbols

\textsymfigsymbol{\langle piece\rangle} and \textsymking{\langle piecechar\rangle} use symbols or text depending on a switch that can be set with \usetextfig and \usesymfig. There is no difference between the different versions to set a single piece. In previous versions \textfigking did work like the new \textsymking, but then I decided that a user would prefer if the short commands can switch between text and symbols.

\begin{verbatim}
\\ newcommand\PieceChar{Q}\\ \textsymfigsymbol{Q}, \textsymfigsymbol{\textfigsymbol{Q}}
\end{verbatim}

\begin{verbatim}
\settextfigchars{\langle silly\rangle}{\langle silly\rangle}
\textking, \textqueen, \textrook, \textbishop, \textknight, \textpawn
\end{verbatim}

\begin{verbatim}
\newcommand\PieceChar{Q}\\ \textsymfigsymbol{Q}, \textsymfigsymbol{\textfigsymbol{Q}}
\end{verbatim}

\footnote{In the previous versions the command did have the name \textfigtextchars. I unified the naming but the older command will work too.}
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 breaks 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

4.6 Changing the look of the figurines

\setfigstyle With \setfigstyle{(font commands)} 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.

\footnote{In older versions this command was called \settextfigstyle. I changed the name as it can now also affect symbol pieces. The old name works too.}

June 14, 2006 12

chessfss Version 1.2
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 error-prone 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 $\text{K}$ and “PieceOnBlack”-chars $\text{J}$. 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 ü 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 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.
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 \texttt{\boardfont} and \texttt{\boardsymbol{(text)}} sets all font properties to the values stored in \texttt{\cfss@boardfontfamily}, \texttt{\cfss@boardfontseries}, \texttt{\cfss@boardfontsize} and \texttt{\cfss@boardfontencoding}.

The commands \texttt{\setboardfontfamily{(family)}}, \texttt{\setboardfontseries{(series)}}\hspace{1em}, \texttt{\setboardfontsize{(size)}} and \texttt{\setboardfontencoding} change these values. \texttt{\setboardfontsize{(size)}} sets the font size to the value \texttt{(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 \texttt{\boardfont} command.

There isn’t a \texttt{\setboardfontshape{(shape)}} command as I couldn’t see what an \texttt{\itshape} board should be. But if the need arise I could add it.
For each field of the board there is a command like \WhiteKingOnWhite. See table 2 on page 24 for all the names.

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.

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

\texttt{\textbackslash getsquarewidth(\langle ReferencePiece \rangle)} 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 \texttt{\langle ReferencePiece \rangle} should be the black empty square. \texttt{\textbackslash setboardfontsize} e.g. calls \texttt{\textbackslash getsquarewidth(\texttt{\textbackslash BlackEmptySquare})}.

If you want to make sure, that a piece sits on the baseline, you should raise it e.g. with \texttt{\textbackslash raisebox\{\textbackslash depth\}\{\langle PIECE \rangle\}}

### 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 \texttt{\textbackslash testboard} command used above will not work, or
- use a package like \texttt{chessboard} to print the board (\texttt{\textbackslash showboard} of the package \texttt{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 \texttt{skaknew} fonts (until now they are the only fonts that can use every encoding)
- load the encoding definition you want to use with \texttt{fontenc} (the package \texttt{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 \texttt{LSBC1}, \texttt{LSBC2} etc. I named the other ones (that can be used by every font that works with LSB) \texttt{LSB1}, \texttt{LSB2} . . . .
5.4 The logical layers

I distinguish for different logical layers in the encodings.

<table>
<thead>
<tr>
<th>name of layer</th>
<th>examples</th>
<th>description/comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>fieldmask</td>
<td>🟩⬛⬛</td>
<td>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!</td>
</tr>
<tr>
<td>field</td>
<td>🟩⬛⬛</td>
<td>the transparent field with the small diagonals.</td>
</tr>
<tr>
<td>piecemask</td>
<td>🟩⬛⬛</td>
<td>solid and opaque, can be used to block out the background or field lines in the inner of a piece.</td>
</tr>
<tr>
<td>piece</td>
<td>🟩⬛⬛</td>
<td>font chars.</td>
</tr>
</tbody>
</table>

5.5 The predefined encodings

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.

<table>
<thead>
<tr>
<th>Encoding</th>
<th>LSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer:</td>
<td>fieldmask</td>
</tr>
<tr>
<td>WhiteSquare</td>
<td>(color)</td>
</tr>
<tr>
<td>BlackSquare</td>
<td>(color)</td>
</tr>
<tr>
<td>WhiteOnWhite</td>
<td>(color)</td>
</tr>
<tr>
<td>WhiteOnBlack</td>
<td>(color)</td>
</tr>
<tr>
<td>BlackOnWhite</td>
<td>(color)</td>
</tr>
<tr>
<td>BlackOnBlack</td>
<td>(color)</td>
</tr>
</tbody>
</table>
### LSB1 Encoding LSB1

<table>
<thead>
<tr>
<th>Layer:</th>
<th>fieldmask</th>
<th>field</th>
<th>piecemask</th>
<th>piece</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>WhiteSquare</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackSquare</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WhiteOnWhite</td>
<td></td>
<td></td>
<td>K</td>
<td>K</td>
<td></td>
</tr>
<tr>
<td>WhiteOnBlack</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackOnWhite</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackOnBlack</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### LSB2 Encoding LSB2

fields have no color!

<table>
<thead>
<tr>
<th>Layer:</th>
<th>fieldmask</th>
<th>field</th>
<th>piecemask</th>
<th>piece</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>WhiteSquare</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackSquare</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WhiteOnWhite</td>
<td></td>
<td></td>
<td>K</td>
<td>K</td>
<td></td>
</tr>
<tr>
<td>WhiteOnBlack</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackOnWhite</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackOnBlack</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### LSB3 Encoding LSB3

fields have no char or color!
White and black pieces use the same char!

<table>
<thead>
<tr>
<th>Layer:</th>
<th>fieldmask</th>
<th>field</th>
<th>piecemask</th>
<th>piece</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>WhiteSquare</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackSquare</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WhiteOnWhite</td>
<td></td>
<td></td>
<td>K</td>
<td>K</td>
<td></td>
</tr>
<tr>
<td>WhiteOnBlack</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackOnWhite</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackOnBlack</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### LSBC1

**Encoding LSBC1**

<table>
<thead>
<tr>
<th>Layer:</th>
<th>fieldmask</th>
<th>field</th>
<th>piecemask</th>
<th>piece</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>WhiteSquare</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackSquare</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WhiteOnWhite</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WhiteOnBlack</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackOnWhite</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackOnBlack</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### LSBC2

**Encoding LSBC2**

<table>
<thead>
<tr>
<th>Layer:</th>
<th>fieldmask</th>
<th>field</th>
<th>piecemask</th>
<th>piece</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>WhiteSquare</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackSquare</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WhiteOnWhite</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WhiteOnBlack</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackOnWhite</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackOnBlack</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### LSBC3

**Encoding LSBC3**

<table>
<thead>
<tr>
<th>Layer:</th>
<th>fieldmask</th>
<th>field</th>
<th>piecemask</th>
<th>piece</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>WhiteSquare</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackSquare</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WhiteOnWhite</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WhiteOnBlack</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackOnWhite</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackOnBlack</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### LSBC4

**Encoding LSBC4**

<table>
<thead>
<tr>
<th>Layer</th>
<th>fieldmask</th>
<th>field</th>
<th>piecemask</th>
<th>piece</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>WhiteSquare</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackSquare</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WhiteOnWhite</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WhiteOnBlack</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackOnWhite</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackOnBlack</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### LSBC5

**Encoding LSBC5**

<table>
<thead>
<tr>
<th>Layer</th>
<th>fieldmask</th>
<th>field</th>
<th>piecemask</th>
<th>piece</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>WhiteSquare</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackSquare</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WhiteOnWhite</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WhiteOnBlack</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackOnWhite</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BlackOnBlack</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### 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:

<table>
<thead>
<tr>
<th>layer</th>
<th>color commands</th>
<th>default definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>fieldmask</td>
<td>\cfss@whitefieldmaskcolor</td>
<td>\color{white}</td>
</tr>
<tr>
<td></td>
<td>\cfss@blackfieldmaskcolor</td>
<td>\color{gray}</td>
</tr>
<tr>
<td>field</td>
<td>\cfss@whitefieldcolor</td>
<td>{}</td>
</tr>
<tr>
<td></td>
<td>\cfss@blackfieldcolor</td>
<td>{}</td>
</tr>
<tr>
<td>piecemask</td>
<td>\cfss@whiteonwhitepiecemaskcolor</td>
<td>\color{white}</td>
</tr>
<tr>
<td></td>
<td>\cfss@whiteonblackpiecemaskcolor</td>
<td>\color{white}</td>
</tr>
<tr>
<td></td>
<td>\cfss@blackonwhitepiecemaskcolor</td>
<td>\color{white}</td>
</tr>
<tr>
<td></td>
<td>\cfss@blackonblackpiecemaskcolor</td>
<td>\color{white}</td>
</tr>
<tr>
<td>piece</td>
<td>\cfss@whitepiececolor</td>
<td>{}</td>
</tr>
<tr>
<td></td>
<td>\cfss@blackpiececolor</td>
<td>{}</td>
</tr>
</tbody>
</table>

With this command you can change the colors in the different layers. The command use a keyval syntax.
6 Using and changing the informator font

As for the figurines and for the board characters I declared an encoding for the informator 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 \texttt{skak} fonts in it and so it contains symbols that are unique to the \texttt{skak} and \texttt{skaknew} font.

I plan to sort the symbols a bit in different sets (as \texttt{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 \texttt{skak} fonts. On the other side symbols like ! and !? or \texttt{\textsf{novelty}} (\texttt{N}) (which in the package \texttt{skak} is defined as \texttt{\textsf{N}}) should perhaps be part of the encoding.

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

The “informator symbols” are symbols to comment a game. \texttt{\textsf{inffont}} and \texttt{\textsf{infsymbol}}\{(text)\} change the encoding to LSI and the fontfamily to \texttt{\cfss@inffontfamily}. \texttt{\cfss@inffontfamily} can be changed with the command \texttt{\setinffontfamily\{(family)\}}.

For each piece there is a command to call the symbol directly, see table 3 on page 25.

For each Symbol there is also an internal command. Please read the explications for the
figurines and the boardsymbol to understand the difference between the internal and the external commands.

$$\neg \Delta, \neg \nabla, \neg i, \Delta$$

Future plan: the NAG's

In PGN (the standard format to store chess games) some of the informator symbols can be coded with NAG’s\(^7\). These consist of a dollar sign followed by a number. I plan to add to the package \texttt{chessfss} commands like \texttt{@namedef\{\$1\}\{!\}} so that package writers can use \texttt{@nameuse\{\$1\}} while parsing a pgn.

7 Other font changes

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

The package \texttt{skak} uses the commands \texttt{\notationOn} and \texttt{\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. \texttt{\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 \texttt{\sidefont} and \texttt{\sidesymbol\{⟨text⟩\}. It’s up to the package writers to use these commands in macros like \texttt{\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 \texttt{skak} and \texttt{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 \texttt{\chessifont}, \texttt{\chessiifont}.
7.3 Changing more than one font

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).

\begin{tabular}{ll}
Symbol & commands \\
\hline
O & \castlingchar \\
O-O & \shortcastling \\
O-O-O & \longcastling \\
N & \novelty \\
RR & \chesscomment \\
R & \various \\
\end{tabular}

9 Character and command tables of the encodings

\begin{table}[h]
\centering
\begin{tabular}{lll}
Symbol & ASCII & commands \\
\hline
\K & K & \symking, \cfss@symking \\
\Q & Q & \symqueen, \cfss@symqueen \\
\R & R & \symrook, \cfss@symrook \\
\B & B & \symbishop, \cfss@symbishop \\
\N & N & \symknight, \cfss@symknight \\
\p & p & \sympawn, \cfss@sympawn \\
\end{tabular}
\caption{The characters in the LSF-encoding (figurine)}
\end{table}

\textsuperscript{7} Numeric Annotation Glyphs

\textsuperscript{8} A copy of this command exists under the name \comment. But this can give a clash with various verbatim packages.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>ASCII</th>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>empty white square. A character that is invisible, but it’s there. But it isn’t really a square, it only has a width: _</td>
<td>\WhiteEmptySquare</td>
</tr>
<tr>
<td>Z</td>
<td>Z</td>
<td>empty black square.</td>
<td>\BlackEmptySquare</td>
</tr>
<tr>
<td>K</td>
<td>k</td>
<td>white king on white</td>
<td>\WhiteKingOnWhite</td>
</tr>
<tr>
<td>J</td>
<td>j</td>
<td>black king on black</td>
<td>\BlackKingOnWhite</td>
</tr>
<tr>
<td>Q</td>
<td>q</td>
<td>white queen on white</td>
<td>\WhiteQueenOnWhite</td>
</tr>
<tr>
<td>L</td>
<td>l</td>
<td>black queen on black</td>
<td>\BlackQueenOnBlack</td>
</tr>
<tr>
<td>R</td>
<td>r</td>
<td>white rook on white</td>
<td>\WhiteRookOnWhite</td>
</tr>
<tr>
<td>S</td>
<td>s</td>
<td>black rook on white</td>
<td>\BlackRookOnWhite</td>
</tr>
<tr>
<td>B</td>
<td>b</td>
<td>white bishop on white</td>
<td>\WhiteBishopOnWhite</td>
</tr>
<tr>
<td>A</td>
<td>a</td>
<td>black bishop on black</td>
<td>\BlackBishopOnBlack</td>
</tr>
<tr>
<td>N</td>
<td>n</td>
<td>white knight on white</td>
<td>\WhiteKnightOnWhite</td>
</tr>
<tr>
<td>M</td>
<td>m</td>
<td>black knight on black</td>
<td>\BlackKnightOnBlack</td>
</tr>
<tr>
<td>P</td>
<td>p</td>
<td>white pawn on white</td>
<td>\WhitePawnOnWhite</td>
</tr>
<tr>
<td>O</td>
<td>o</td>
<td>black pawn on black</td>
<td>\BlackPawnOnBlack</td>
</tr>
<tr>
<td>Symbol</td>
<td>ASCII</td>
<td>command,</td>
<td>Symbol</td>
</tr>
<tr>
<td>--------</td>
<td>-------</td>
<td>---------------</td>
<td>--------</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
<td>\checksymbol</td>
<td>±</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>\castlinghyphen, used only in O-O</td>
<td>©</td>
</tr>
<tr>
<td>→</td>
<td>A</td>
<td>\withattack</td>
<td>±</td>
</tr>
<tr>
<td>†</td>
<td>C</td>
<td>\withinit</td>
<td>÷</td>
</tr>
<tr>
<td>⊙</td>
<td>D</td>
<td>\zugzwang</td>
<td>++</td>
</tr>
<tr>
<td>△</td>
<td>E</td>
<td>\withidea</td>
<td>−+</td>
</tr>
<tr>
<td>□</td>
<td>F</td>
<td>\onlymove</td>
<td>=</td>
</tr>
<tr>
<td>↘</td>
<td>G</td>
<td>\diagonal</td>
<td>∞</td>
</tr>
<tr>
<td>⇔</td>
<td>H</td>
<td>\file</td>
<td>−</td>
</tr>
<tr>
<td>♡</td>
<td>I</td>
<td>\centre</td>
<td>±</td>
</tr>
<tr>
<td>×</td>
<td>J</td>
<td>\weakpt</td>
<td>☺</td>
</tr>
<tr>
<td>⊥</td>
<td>L</td>
<td>\ending</td>
<td>☞</td>
</tr>
<tr>
<td>≪</td>
<td>M</td>
<td>\qside</td>
<td>o-0</td>
</tr>
<tr>
<td>≫</td>
<td>O</td>
<td>\kside</td>
<td>δ</td>
</tr>
<tr>
<td>≦</td>
<td>P</td>
<td>\etc</td>
<td>♠</td>
</tr>
<tr>
<td>&gt;</td>
<td>S</td>
<td>\morepawns</td>
<td>☺</td>
</tr>
<tr>
<td>⊕</td>
<td>T</td>
<td>\timelimit</td>
<td>∞</td>
</tr>
<tr>
<td>○</td>
<td>U</td>
<td>\moreroom</td>
<td>⊱</td>
</tr>
<tr>
<td>≡</td>
<td>V</td>
<td>\counterplay</td>
<td>⊱</td>
</tr>
<tr>
<td>×</td>
<td>X</td>
<td>\capturesymbol</td>
<td>×</td>
</tr>
<tr>
<td>☳</td>
<td>a</td>
<td>\bishoppair</td>
<td>O</td>
</tr>
<tr>
<td>☳</td>
<td>b</td>
<td>\betteris</td>
<td></td>
</tr>
</tbody>
</table>

*Remark: You shouldn’t rely on the fact that other informator fonts that you may find somewhere have exactly the same characters.*
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 informator 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: \texttt{lsf}, \texttt{lsb} and \texttt{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 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 \texttt{raw} together with some information about the character set, e.g. \texttt{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.

\subsection{Example}

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

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

\subsection{Testing of the fonts}

To test the raw and the reencoded fonts, you should get acquainted with \texttt{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 \texttt{tfm}-file (without the ending) and then “enter”. Then type \texttt{\textbackslash table\textbackslash bye} to get a fonttable.

The main problem with \texttt{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 \texttt{\def\:\} an \texttt{\fbox{}} command:

\begin{verbatim}
\def\:\{\setbox0=\hbox{\fbox{\char\n}}}
\end{verbatim}

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

If you get errors you can use these rules of thump: If \texttt{\LaTeX} or pdf\texttt{\LaTeX} complain during the setting of the document about problems with the font, there is in most cases a problem with the \texttt{tfm} files. If \texttt{dvips} complains (or pdf\texttt{\LaTeX} at the end while inserting
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 \url{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 pdftex 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

June 14, 2006 29 chessfss Version 1.2
Then copy all the \texttt{tfm}, \texttt{afm} and \texttt{pfb}-files where they belong in your \TeX{}-system (see the FAQ-pointers in \cite{3.5}).

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

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

chess-alfonso-lsf Chess-Alfonso-X " ChessFigEncoding ReEncodeFont "
\<chess-lsfalfonso-fig-raw.pfb

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

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

Then add \texttt{chess.map} to your local \texttt{updmap.cfg}.

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

So either open a new textdocument (or an existing \texttt{chessfss.cfg}), add the lines

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

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

and save the document as \texttt{chessfss.cfg} e.g. in \texttt{localtexmf/tex/latex/chessfss/}. Or put the line in files named \texttt{lssfalfonso.fd} and \texttt{lssfalfonso.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 \texttt{lsb1alfonso.fd}, \texttt{lsb2alfonso.fd} or \texttt{lsb3alfonso.fd}:

```latex
9Don't put them in \texttt{chessfss.cfg}, the experimental encodings aren't declared by default in the package \textit{chessfss}
```
Update all databases and maps and test the new fonts

Renew your filename database e.g. for Miktex with \texttt{initexmf -u} and call \texttt{updmap}, e.g. with \texttt{initexmf --mkmaps}.

Try \texttt{nffsfont.tex} on the fonts \texttt{chess-alfonso-board-fig-raw} and \texttt{chess-alfonso-1sb} to understand how the encoding vector changes the character set and the order of the fonts.

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

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

\begin{quote}
\LaTeX\  Font  Warning:  Font  shape '\texttt{LSF/alfonso/bx/n}'  undefined  (Font) using  '\texttt{LSF/alfonso/m/n}'  instead  on  input  line  9.
\end{quote}

As already mentioned, there is nothing to worry about. If you don’t like such warnings, learn (e.g. by reading \texttt{fntguide.tex} or the \LaTeX\ Companion) which \texttt{\DeclareFontShape} command you need to get silence substitutions.

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 \texttt{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 \texttt{fontinst} for the installation. One needs files \texttt{lsf.etx}, \texttt{lsb.etx} and \texttt{lsi.etx} that describes the encoding, \texttt{mtx}-files that describes the adjustions and then one should use a \texttt{tex} file with commands like

\begin{quote}
\texttt{\installfont{<fontname>}{<fromafm>,<mtx>}{<enc>}{<enc>}{<family>}{<series>}{<shape>}{}}
\end{quote}

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`

```plaintext
/*Chess***FigEncoding*
%0z
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%1z
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%2z
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%3z
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
/.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef /.notdef
%4z
/.notdef /.notdef
%% Bishop
/*** */
/.notdef /.notdef /.notdef /.notdef /
/.notdef /.notdef /
%% King
/*** */
/.notdef /
%% Knight
/*** */
/.notdef /
%% Queen
/*** */
%% Rook
/*** */
/.notdef /.notdef /.notdef /.notdef /
/.notdef /.notdef /.notdef /.notdef /
%% Pawn
/*** */
/.notdef /.notdef /.notdef /.notdef /
/.notdef /.notdef /
```

June 14, 2006 32 chessfss Version 1.2
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
  /KKingOnBlack

June 14, 2006 33 chessfss Version 1.2
.piece

**WKingOnWhite**
**WQueenOnBlack**
**WKnightOnBlack**
**WKnightOnWhite**
**WPawnOnBlack**
**WPawnOnWhite**
**WQueenOnWhite**
**WRookOnWhite**
**WRookOnBlack**

**BKingOnBlack**
**BKingOnWhite**
**BBishopOnBlack**
**BBishopOnWhite**

**king piecemask**
**knight piecemask**
**rook piecemask**
**bishop piecemask**
**queen piecemask**
**pawn piecemask**
**BlackSquare**

/*
.
*/
/*
*/

June 14, 2006 34 chessfss Version 1.2
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/downl/alfonso.zip](http://www.enpassant.dk/chess/downl/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:

<table>
<thead>
<tr>
<th>encoding</th>
<th>serie</th>
<th>tfm-name</th>
<th>reencoding command for chess.map</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td></td>
<td>chess-alfonso-board-fig-raw</td>
<td>none</td>
</tr>
<tr>
<td>LSF</td>
<td>m</td>
<td>chess-alfonso-lsf</td>
<td>&quot; ChessFigEncoding ReEncodeFont &quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&lt;chess-fig.enc&gt;</td>
</tr>
<tr>
<td>LSB</td>
<td>m</td>
<td>chess-alfonso-lsb</td>
<td>&quot; ChessBoardEncoding ReEncodeFont &quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&lt;chess-board.enc&gt;</td>
</tr>
</tbody>
</table>

June 14, 2006 36  chessfss Version 1.2
Alpha

code:  
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 senseful before using this font (or ignore the messages about the missing 0).

code:  
familyname: alpha
internal fontname: Chess-Alpha
fonts: The following fonts for the package chessfss can be made from this source:

<table>
<thead>
<tr>
<th>encoding</th>
<th>serie</th>
<th>shape</th>
<th>tfm-name</th>
<th>reencoding command for chess.map</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td></td>
<td></td>
<td>chess-alpha-board-fig-raw</td>
<td>none</td>
</tr>
<tr>
<td>LSF</td>
<td>m</td>
<td>n</td>
<td>chess-alpha-lsf</td>
<td>&quot;ChessFigEncoding ReEncodeFont &quot; &lt;chess-alpha-fig.enc</td>
</tr>
<tr>
<td>LSF</td>
<td>m</td>
<td>bl</td>
<td>chess-alpha-bl-lsf</td>
<td>&quot;ChessFigEncoding ReEncodeFont &quot; &lt;chess-alpha-black-fig.enc</td>
</tr>
<tr>
<td>LSB</td>
<td>m</td>
<td>n</td>
<td>chess-alpha-lsb</td>
<td>&quot;ChessBoardEncoding ReEncodeFont &quot; &lt;chess-alpha-board.enc</td>
</tr>
</tbody>
</table>

June 14, 2006 37 chessfss Version 1.2
Aventurer

source: [http://www.enpassant.dk/chess/downl/adventur.zip](http://www.enpassant.dk/chess/downl/adventur.zip)

author: Armando H. Marroquin

characters: figurine and board symbols

familyname: aventurer

internal fontname: ChessAdventurer

fonts: The following fonts for the package chessfss can be made from this source:

<table>
<thead>
<tr>
<th>encoding</th>
<th>serie</th>
<th>tfm-name</th>
<th>reencoding command for chess.map</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>-</td>
<td>chess-aventurer-board-fig-raw</td>
<td>none</td>
</tr>
<tr>
<td>LSF</td>
<td>m</td>
<td>chess-aventurer-lsf</td>
<td>&quot; ChessFigEncoding ReEncodeFont &quot; &lt;chess-fig.enc</td>
</tr>
<tr>
<td>LSB</td>
<td>m</td>
<td>chess-aventurer-lsb</td>
<td>&quot; ChessBoardEncoding ReEncodeFont &quot; &lt;chess-board.enc</td>
</tr>
</tbody>
</table>
Berlin

source: [http://www.enpassant.dk/chess/downl/berlin.zip](http://www.enpassant.dk/chess/downl/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 senseful 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:

<table>
<thead>
<tr>
<th>encoding</th>
<th>serie</th>
<th>shape</th>
<th>tfm-name</th>
<th>reencoding command for chess.map</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td></td>
<td></td>
<td>chess-berlin-board-fig-raw</td>
<td>none</td>
</tr>
<tr>
<td>LSF</td>
<td>m</td>
<td>n</td>
<td>chess-berlin-lsf</td>
<td>&quot;ChessFigEncoding ReEncodeFont &quot; &lt;chess-alpha-fig.enc</td>
</tr>
<tr>
<td>LSF</td>
<td>m</td>
<td>bl</td>
<td>chess-berlin-b-lsf</td>
<td>&quot;ChessFigEncoding ReEncodeFont &quot; &lt;chess-alpha-black-fig.enc</td>
</tr>
<tr>
<td>LSB</td>
<td>m</td>
<td>n</td>
<td>chess-berlin-lsb</td>
<td>&quot;ChessBoardEncoding ReEncodeFont &quot; &lt;chess-berlin-board.enc</td>
</tr>
</tbody>
</table>

June 14, 2006 39 chessfss Version 1.2
Cases

source: [http://www.enpassant.dk/chess/downl/cases.zip](http://www.enpassant.dk/chess/downl/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:

<table>
<thead>
<tr>
<th>encoding</th>
<th>serie</th>
<th>tfm-name</th>
<th>reencoding command for chess.map</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>–</td>
<td>chess-cases-board-fig-raw</td>
<td>none</td>
</tr>
<tr>
<td>LSF</td>
<td>m</td>
<td>chess-cases-lsf</td>
<td>&quot;ChessFigEncoding ReEncodeFont&quot; &lt;chess-fig.enc</td>
</tr>
<tr>
<td>LSB</td>
<td>m</td>
<td>chess-cases-lsb</td>
<td>&quot;ChessBoardEncoding ReEncodeFont&quot; &lt;chess-board.enc</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>encoding</th>
<th>serie</th>
<th>tfm-name</th>
<th>reencoding command for chess.map</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>–</td>
<td>chess-cheq-board-raw</td>
<td>none</td>
</tr>
<tr>
<td>LSB</td>
<td>m</td>
<td>chess-cheq-lsb</td>
<td>&quot;ChessBoardEncoding ReEncodeFont&quot; &lt;chess-cheq-board.enc</td>
</tr>
</tbody>
</table>
**Condal**

source: [http://www.enpassant.dk/chess/downl/condal.zip](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:

<table>
<thead>
<tr>
<th>encoding</th>
<th>serie</th>
<th>tfm-name</th>
<th>reencoding command for chess.map</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td></td>
<td>chess-condal-board-fig-raw</td>
<td>none</td>
</tr>
<tr>
<td>LSF</td>
<td>m</td>
<td>chess-condal-lsf</td>
<td>&quot;ChessFigEncoding ReEncodeFont &quot;&lt;chess-fig.enc</td>
</tr>
<tr>
<td>LSB</td>
<td>m</td>
<td>chess-condal-lsb</td>
<td>&quot;ChessBoardEncoding ReEncodeFont &quot;&lt;chess-board.enc</td>
</tr>
</tbody>
</table>

**Harlequin**

source: [http://www.enpassant.dk/chess/downl/harlequi.zip](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:

<table>
<thead>
<tr>
<th>encoding</th>
<th>serie</th>
<th>tfm-name</th>
<th>reencoding command for chess.map</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td></td>
<td>chess-harlequin-board-fig-raw</td>
<td>none</td>
</tr>
<tr>
<td>LSF</td>
<td>m</td>
<td>chess-harlequin-lsf</td>
<td>&quot;ChessFigEncoding ReEncodeFont &quot;&lt;chess-fig.enc</td>
</tr>
<tr>
<td>LSB</td>
<td>m</td>
<td>chess-harlequin-lsb</td>
<td>&quot;ChessBoardEncoding ReEncodeFont &quot;&lt;chess-board.enc</td>
</tr>
</tbody>
</table>

June 14, 2006 41  chessfss Version 1.2
### Kingdom

source: [http://www.enpassant.dk/chess/downl/kingdom.zip](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:

<table>
<thead>
<tr>
<th>encoding</th>
<th>serie</th>
<th>tfm-name</th>
<th>reencoding command for chess.map</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>-</td>
<td>chess-kingdom-board-fig-raw</td>
<td>none</td>
</tr>
<tr>
<td>LSF</td>
<td>m</td>
<td>chess-kingdom-lsf</td>
<td>&quot;ChessFigEncoding ReEncodeFont &lt;chess-fig.enc&quot;</td>
</tr>
<tr>
<td>LSB</td>
<td>m</td>
<td>chess-kingdom-lsb</td>
<td>&quot;ChessBoardEncoding ReEncodeFont &lt;chess-board.enc&quot;</td>
</tr>
</tbody>
</table>

### Leipzig

source: [http://www.enpassant.dk/chess/downl/leipzig.zip](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:

<table>
<thead>
<tr>
<th>encoding</th>
<th>serie</th>
<th>tfm-name</th>
<th>reencoding command for chess.map</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>-</td>
<td>chess-leipzig-board-fig-raw</td>
<td>none</td>
</tr>
<tr>
<td>LSF</td>
<td>m</td>
<td>chess-leipzig-lsf</td>
<td>&quot;ChessFigEncoding ReEncodeFont &lt;chess-fig.enc&quot;</td>
</tr>
<tr>
<td>LSB</td>
<td>m</td>
<td>chess-leipzig-lsb</td>
<td>&quot;ChessBoardEncoding ReEncodeFont &lt;chess-board.enc&quot;</td>
</tr>
</tbody>
</table>

June 14, 2006 42 chessfss Version 1.2
**Line**


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:

<table>
<thead>
<tr>
<th>encoding</th>
<th>serie</th>
<th>tfm-name</th>
<th>reencoding command for chess.map</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td></td>
<td>chess-line-board-fig-raw</td>
<td>none</td>
</tr>
<tr>
<td>LSF</td>
<td>m</td>
<td>chess-line-lsf</td>
<td>&quot;ChessFigEncoding ReEncodeFont &quot;&lt;chess-fig.enc</td>
</tr>
<tr>
<td>LSB</td>
<td>m</td>
<td>chess-line-lsb</td>
<td>&quot;ChessBoardEncoding ReEncodeFont &quot;&lt;chess-board.enc</td>
</tr>
</tbody>
</table>

**Lucena**

source: [http://www.enpassant.dk/chess/downl/lucena.zip](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:

<table>
<thead>
<tr>
<th>encoding</th>
<th>serie</th>
<th>tfm-name</th>
<th>reencoding command for chess.map</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td></td>
<td>chess-lucena-board-fig-raw</td>
<td>none</td>
</tr>
<tr>
<td>LSF</td>
<td>m</td>
<td>chess-lucena-lsf</td>
<td>&quot;ChessFigEncoding ReEncodeFont &quot;&lt;chess-fig.enc</td>
</tr>
<tr>
<td>LSB,LSBC4</td>
<td>m</td>
<td>chess-lucena-lsb-1001</td>
<td>&quot;ChessBoardEncoding ReEncodeFont &quot;&lt;chess-board.enc</td>
</tr>
</tbody>
</table>
### Magnetic

source: [http://www.enpassant.dk/chess/downl/magnetic.zip](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:

<table>
<thead>
<tr>
<th>encoding</th>
<th>serie</th>
<th>tfm-name</th>
<th>reencoding command for chess.map</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>-</td>
<td>chess-magnetic-board-fig-raw</td>
<td>none</td>
</tr>
<tr>
<td>LSF</td>
<td>m</td>
<td>chess-magnetic-lsf</td>
<td>&quot;ChessFigEncoding ReEncodeFont&quot;&lt;chess-fig.enc</td>
</tr>
<tr>
<td>LSB</td>
<td>m</td>
<td>chess-magnetic-lsb</td>
<td>&quot;ChessBoardEncoding ReEncodeFont&quot;&lt;chess-board.enc</td>
</tr>
</tbody>
</table>

### Mark


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:

<table>
<thead>
<tr>
<th>encoding</th>
<th>serie</th>
<th>tfm-name</th>
<th>reencoding command for chess.map</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>-</td>
<td>chess-mark-board-fig-raw</td>
<td>none</td>
</tr>
<tr>
<td>LSF</td>
<td>m</td>
<td>chess-mark-lsf</td>
<td>&quot;ChessFigEncoding ReEncodeFont&quot;&lt;chess-fig.enc</td>
</tr>
<tr>
<td>LSB</td>
<td>m</td>
<td>chess-mark-lsb</td>
<td>&quot;ChessBoardEncoding ReEncodeFont&quot;&lt;chess-board.enc</td>
</tr>
</tbody>
</table>
### Marroquin

source: [http://www.enpassant.dk/chess/downl/marroqui.zip](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:

<table>
<thead>
<tr>
<th>encoding</th>
<th>serie</th>
<th>tfm-name</th>
<th>reencoding command for chess.map</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td></td>
<td>chess-marroquin-board-fig-raw none</td>
<td></td>
</tr>
<tr>
<td>LSF</td>
<td>m</td>
<td>chess-marroquin-lsf              &quot; ChessFigEncoding ReEncodeFont &quot; &lt;chess-fig.enc</td>
<td></td>
</tr>
<tr>
<td>LSB</td>
<td>m</td>
<td>chess-marroquin-lsb              &quot; ChessBoardEncoding ReEncodeFont &quot; &lt;chess-board.enc</td>
<td></td>
</tr>
</tbody>
</table>

### Maya

source: [http://www.enpassant.dk/chess/downl/chesmaya.zip](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:

<table>
<thead>
<tr>
<th>encoding</th>
<th>serie</th>
<th>tfm-name</th>
<th>reencoding command for chess.map</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td></td>
<td>chess-maya-board-fig-raw none</td>
<td></td>
</tr>
<tr>
<td>LSF</td>
<td>m</td>
<td>chess-maya-lsf              &quot; ChessFigEncoding ReEncodeFont &quot; &lt;chess-fig.enc</td>
<td></td>
</tr>
<tr>
<td>LSB</td>
<td>m</td>
<td>chess-maya-lsb              &quot; ChessBoardEncoding ReEncodeFont &quot; &lt;chess-board.enc</td>
<td></td>
</tr>
</tbody>
</table>

June 14, 2006 45  chessfss Version 1.2
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:

<table>
<thead>
<tr>
<th>encoding</th>
<th>serie</th>
<th>tfm-name</th>
<th>reencoding command for chess.map</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td></td>
<td>chess-mediaeval-board-fig-raw</td>
<td>none</td>
</tr>
<tr>
<td>LSF</td>
<td>m</td>
<td>chess-mediaeval-lsf</td>
<td>&quot; ChessFigEncoding ReEncodeFont &quot; &lt;chess-fig.enc</td>
</tr>
<tr>
<td>LSB</td>
<td>m</td>
<td>chess-mediaeval-lsb</td>
<td>&quot; ChessBoardEncoding ReEncodeFont &quot; &lt;chess-board.enc</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>encoding</th>
<th>serie</th>
<th>tfm-name</th>
<th>reencoding command for chess.map</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td></td>
<td>chess-merida-board-fig-raw</td>
<td>none</td>
</tr>
<tr>
<td>LSF</td>
<td>m</td>
<td>chess-merida-lsf</td>
<td>&quot; ChessFigEncoding ReEncodeFont &quot; &lt;chess-fig.enc</td>
</tr>
<tr>
<td>LSB</td>
<td>m</td>
<td>chess-merida-lsb</td>
<td>&quot; ChessBoardEncoding ReEncodeFont &quot; &lt;chess-board.enc</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>encoding</th>
<th>serie</th>
<th>tfm-name</th>
<th>reencoding command for chess.map</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>m</td>
<td>chess-millennia-board-fig-raw</td>
<td>none</td>
</tr>
<tr>
<td>raw</td>
<td>b</td>
<td>chess-millennia-b-board-fig-raw</td>
<td>none</td>
</tr>
<tr>
<td>LSF</td>
<td>m</td>
<td>chess-millennia-lsf</td>
<td>&quot;ChessFigEncoding ReEncodeFont &lt;chess-fig.enc&quot;</td>
</tr>
<tr>
<td>LSB</td>
<td>m</td>
<td>chess-millennia-lsb</td>
<td>&quot;ChessBoardEncoding ReEncodeFont &lt;chess-board.enc&quot;</td>
</tr>
<tr>
<td>LSB</td>
<td>b</td>
<td>chess-millennia-b-lsb</td>
<td>&quot;ChessBoardEncoding ReEncodeFont &lt;chess-board.enc&quot;</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>encoding</th>
<th>serie</th>
<th>tfm-name</th>
<th>reencoding command for chess.map</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>–</td>
<td>chess-motif-board-fig-raw</td>
<td>none</td>
</tr>
<tr>
<td>LSF</td>
<td>m</td>
<td>chess-motif-lsf</td>
<td>&quot;ChessFigEncoding ReEncodeFont &lt;chess-fig.enc&quot;</td>
</tr>
<tr>
<td>LSB</td>
<td>m</td>
<td>chess-motif-lsb</td>
<td>&quot;ChessBoardEncoding ReEncodeFont &lt;chess-board.enc&quot;</td>
</tr>
</tbody>
</table>

June 14, 2006 47 chessfss Version 1.2
Pirat

source: [http://www.enpassant.dk/chess/downl/pirat.zip](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 informator 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:

<table>
<thead>
<tr>
<th>encoding</th>
<th>serie</th>
<th>shape</th>
<th>tfm-name</th>
<th>reencoding command for chess.map</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td></td>
<td></td>
<td>chess-pirat-fig-raw</td>
<td>none</td>
</tr>
<tr>
<td>raw</td>
<td></td>
<td></td>
<td>chess-pirat-b-fig-raw</td>
<td>none</td>
</tr>
<tr>
<td>raw</td>
<td></td>
<td></td>
<td>chess-pirat-it-fig-raw</td>
<td>none</td>
</tr>
<tr>
<td>raw</td>
<td></td>
<td></td>
<td>chess-pirat-bit-raw</td>
<td>none</td>
</tr>
<tr>
<td>LSF</td>
<td>m</td>
<td>n</td>
<td>chess-pirat-lsf</td>
<td>&quot;ChessFigEncoding ReEncodeFont &quot; &lt;chess-pirat-fig.enc</td>
</tr>
<tr>
<td>LSF</td>
<td>b, bx</td>
<td>n</td>
<td>chess-pirat-b-lsf</td>
<td>&quot;ChessFigEncoding ReEncodeFont &quot; &lt;chess-pirat-fig.enc</td>
</tr>
<tr>
<td>LSF</td>
<td>m</td>
<td>it</td>
<td>chess-pirat-it-lsf</td>
<td>&quot;ChessFigEncoding ReEncodeFont &quot; &lt;chess-pirat-fig.enc</td>
</tr>
<tr>
<td>LSF</td>
<td>b, bx</td>
<td>it</td>
<td>chess-pirat-bit-lsf</td>
<td>&quot;ChessFigEncoding ReEncodeFont &quot; &lt;chess-pirat-fig.enc</td>
</tr>
<tr>
<td>LSB</td>
<td>m</td>
<td>n</td>
<td>chess-pirat-lsb</td>
<td>&quot;ChessBoardEncoding ReEncodeFont &quot; &lt;chess-pirat-board.enc</td>
</tr>
</tbody>
</table>
Skak


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 informator symbols, figurines and the informator 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:

<table>
<thead>
<tr>
<th>encoding serie</th>
<th>tfm-name</th>
<th>reencoding command for chess.map</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSF, LSI m</td>
<td>skakf10</td>
<td>–</td>
</tr>
<tr>
<td>LSF, LSI b</td>
<td>skakf10b</td>
<td>–</td>
</tr>
<tr>
<td>LSB m</td>
<td>skak10, skak15, skak20, skak30</td>
<td>–</td>
</tr>
</tbody>
</table>

Skaknew


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

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

familyname: skaknew

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

fonts: The following fonts are in this family:

<table>
<thead>
<tr>
<th>encoding serie</th>
<th>tfm-name</th>
<th>reencoding command for chess.map</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSF, LSI m</td>
<td>SkakNew-Figure</td>
<td>–</td>
</tr>
<tr>
<td>LSF, LSI b</td>
<td>SkakNew-Figure-Bold</td>
<td>–</td>
</tr>
<tr>
<td>LSB, LSBC1-4 m</td>
<td>SkakNew-Diagram</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>SkakNew-DiagramT</td>
<td>for small diagrams</td>
</tr>
</tbody>
</table>
Skak-k6

source: http://www.enpassant.dk/chess/downl/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:

<table>
<thead>
<tr>
<th>encoding</th>
<th>serie</th>
<th>tfm-name</th>
<th>reencoding command for chess.map</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>–</td>
<td>chess-skak-k6-board-raw</td>
<td>--</td>
</tr>
<tr>
<td>LSB</td>
<td>m</td>
<td>chess-skak-k6-lsb</td>
<td>&quot; ChessBoardEncoding ReEncodeFont &quot; &lt;chess-skak-k6-board.enc</td>
</tr>
</tbody>
</table>

Utrecht

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:

<table>
<thead>
<tr>
<th>encoding</th>
<th>serie</th>
<th>tfm-name</th>
<th>reencoding command for chess.map</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>–</td>
<td>chess-utrecht-board-fig-raw</td>
<td>none</td>
</tr>
<tr>
<td>LSB</td>
<td>m</td>
<td>chess-utrecht-lsb</td>
<td>&quot; ChessUtrechtBoardEncoding ReEncodeFont &quot; &lt;chess-utrecht-board.enc</td>
</tr>
</tbody>
</table>

June 14, 2006 50 chessfss Version 1.2