The vexillology package*

Karl D. Hammond

2026/01/15

Abstract

This package implements symbols used by vexillologists (people who study flags) to indicate certain aspects of flags, such as where they are used, who uses them, and what they look like. The package uses Ti*k*Z to draw the symbols, whose heights scale with the font size.

1 Introduction

In the 1970s, Dr. Whitney Smith, vexillologist and founder of the North American Vexillogical Association, created a series of symbols for use in his books on flags that identify how flags are used. These symbols include whether the particular image being shown is used as the civilian flag rather than for state or military purposes, as well as whether they are officially designed by law or other government policy or simply commonly used, how they appear when hung vertically, and whether the flag is currently used or has been/was officially adopted.

The symbols were proposed (though never officially adopted) by the International Federation of Vexillogical Associations (FIAV) in the late 1970s, and the symbol set has been supplemented on occasion. In particular, Željko Heimer added the symbols ▭ and ▨ to differentiate flags that were currently in use from historical, no-longer-in-use flags. The package author also added the ◹ symbol for clarity.[1]

A proposal exists [A. Pandey, "Proposal to Encode Vexillology Symbols in UNICODE," https://www.unicode.org/L2/L2024/24245-vexillology-symbols.pdf, 2024] to encode the symbols in this document in UNICODE. However, at the time of writing, this has not been done. If this proposal is accetped, a package option will be introduced to use the UNICODE characters directly.

*This document corresponds to vexillology v1.0.0, dated 2026/01/15.

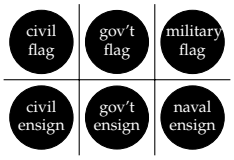
[1]The author has asked in various places what the difference between 🏴 and 🏳 is, but it is not entirely clear. The interpretations herein represent his understanding at the time of the most recent update.

2 Status and Design Symbols

The first set of symbols describes the state of recognition for a flag and its design. The symbols are included in Table 1. These symbols are typically used in tandem with the symbols in section 3, so further discussion of them will be given in a later section for those that are not explained adequately in the table.

3 Usage Symbols

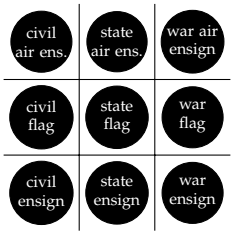
In a second set of symbols, dots in a 3×2 grid denote the use of a flag. The top row represents land-based use, while the bottom shows maritime use. The first column denotes civilian ("civil") use, the second column government ("state") use, and the final column military ("war") use; that is,



used on land

used on water

Occasionally, an "extended" usage symbol is used, which has three rows: air, land, and water:



used in the air

used on land

used on water

\vexusage These symbols are generated by the **\vexusage** command. The argument to **\vexusage** is a six-bit whole number (or, in the case of the extended usage symbol, a nine-bit whole number), with the least-significant bit denoting use as the civil flag. In the extended symbol, the three most significant bits are the air usage bits.

For example, **\vexusage**{23} expands to  , which indicates a flag used as the civil flag (bit 0, or $2^0 = 1$), the state flag (bit 1, or $2^1 = 2$), the war flag (bit 2, or $2^2 = 4$), and the state ensign (bit 5, or $2^5 = 16$), for a total of $2^0 + 2^1 + 2^2 + 2^5 = 23$.

If the argument to **\vexusage** is 64 (i.e., 2^6) or more, the extended symbol is assumed. For example, **\smash**{**\vexusage**{66}} produces  . To force a three-row symbol for an argument of 63 or less, use the starred form: **\smash**{**\vexusage***{23}} expands to  .

The argument to **\vexusage** is parsed for addition and other operations, so it is valid to say **\vexusage**{1+2+4+16} instead of **\vexusage**{23}. To avoid having to remember which bit is which—or do powers of 2 in one's head—several macros are defined to make it easy to set only one or two bits. These macros are listed in Table 2. Using those macros, **\vexusage**{**\vexcivilensign**+**\vexstateensign**}, equivalent to **\vexusage**{24}, yields  .

Table 1. Symbols used to denote the design and/or official status of a flag or ensign.

Symbol	Macro	Meaning
	`\vexnormal`	Ordinary (*de jure*) version of the flag, obverse side
	`\vexproposed`	Design was proposed, but was never officially adopted
	`\vexreconstruction`	Design is a reconstruction based on observations or descriptions
	`\vexreverse`	Design shown is the reverse side
	`\vexvariant`	Design shown is an acceptable variant of the ordinary design
	`\vexalternative`	Design shown is one of several alternative versions that can be displayed
	`\vexdefacto`	Design shown is the *de facto* flag, widely used but not officially adopted
	`\vexreverseisdifferent`	Flag has different designs on the obverse and the reverse sides
	`\vexsinisterhoist`	Obverse side is meant to be hoisted with the pole on the observer's right.
	`\vexauthorized`	Design is officially authorized by the applicable government (unnecessary when ▭ is used)
	`\vexhistorical`	Design was used in the past, but is now abandoned.
	`\vexreverseismirror`	Reverse side is a mirror image of the obverse
	`\vexreverseiscongruent`	Reverse side is identical to the obverse
	`\vexnormalvertical`	Flag can be hung vertically by hanging with an ordinary pole, then rotating 90° counterclockwise (reverse is face-up)
	`\vexobversevertical`	Flag can be hung vertically by hanging with an ordinary pole, then rotating 90° clockwise (obverse is face-up)[a]
	`\vexrotatevertical`	Design does not rotate with the flag when hanging vertically
	`\vexunknownvertical`	Information is not given or is not known for vertical display
	`\vexonlyvertical`	Flag [as shown] can only be hoisted vertically
	`\vexnorotating`	Design has no elements that can be rotated
	`\vexunauthorized`	Flag is not officially authorized to represent the group or country

[a]This symbol is non-standard, but it is used here to differentiate, for example, the flags of Canada and the United States, which show their reverse when hung vertically, from Brazil or Nicaragua, which show the obverse.

Table 2. Shortcut macros to represent each bit in the **\vexusage** command argument.

Macro	Expansion	Result of **\vexusage**{*macro*}
Single-Bit Shortcuts		
\vexcivilflag	1	
\vexstateflag	2	
\vexwarflag	4	
\vexcivilensign	8	
\vexstateensign	16	
\vexwarensign	32	
\vexcivilairensign	64	
\vexstateairensign	128	
\vexwarairensign	256	
Multi-Bit Shortcuts		
\vexnationalflag	7	
\vexnationalensign	56	
\vexnationalflagandensign	63	
\vexnationaleverything	511	

4 Using the Symbols

As an example of the use of these symbols, the flags of Barbados, China, and the Dominican Republic are shown below:


Barbados


China


Dominican Republic

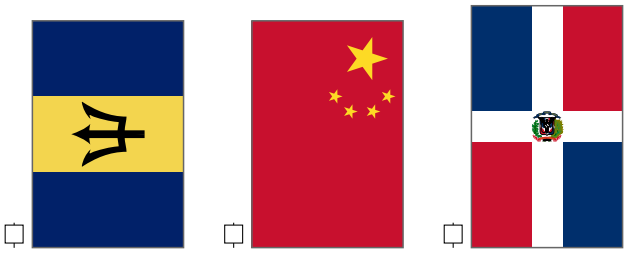
The symbols indicate that Barbados uses the flag variant shown for all purposes except its navy; China uses this flag variant for all civilian and state purposes, but not for military purposes; and the Dominican Republic uses the design shown for state and military use but *not* for civilian use (the civil flag and ensign lack the emblem).

The flag of Barbados has mirror symmetry, so the reverse is both a mirror image of the obverse *and* congruent to it. China and the Dominican Republic, in contrast, look different on the reverse, so the symbol specifies that the reverse of the flag looks like the mirror image of the obverse:



where indicates that the back side of the flag is being shown.

These three flags all have different specifications for *vertical* display: Barbados uses "normal" vertical display, meaning the flag pole is rotated 90° and the flag then flips down and the reverse is shown; China rotates the flag 270°, meaning the *obverse* of the flag is shown in the vertical orientation. The Dominican Republic, in contrast, adopts a modified design for vertical banners, in this case with the base design displayed normally but the emblem rotated so it remains upright.



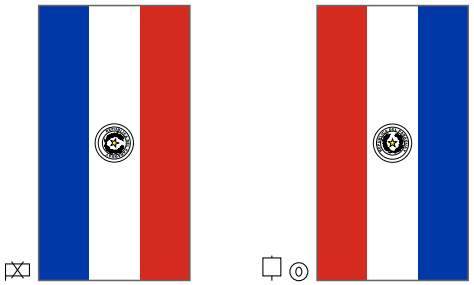
Another example would be a listing for Paraguay, which might look something like this:


Paraguay

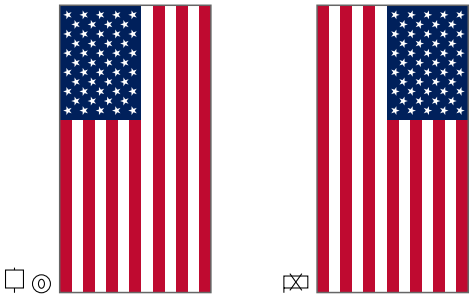

Paraguay (reverse)

The symbols on the left-hand image indicate that the flag is used both as the national flag and the national ensign (i.e., for all uses on both land and sea). The ▭ symbol indicates that the flag is established by law and the view shown is the obverse. The symbol ≠ indicates that the reverse side has a different design, which is the image on the right-hand side. On the right side, the symbol ▭→ indicates that the image shown is the reverse side of the flag; the symbol ⓪ is used here to denote that the image shows is part of the statutory description of the flag[2]

The symbol ℝ indicates that part—but not all—of the design should be rotated when displaying the flag vertically. In this case, it dictates that the correct vertical display of Paraguay's flag is the image on the right, *not* the image on the left:



The difference between ◰ and ◳ is the difference between the two images below:



The *correct* way to hang the flag of the USA vertically is the version on the left, meaning the base design would have the symbol ◰:

USA ⁘ ▭ ◰

If the correct vertical version was the one on the right, the symbol ◳ would be used instead (as it is for the flag of China earlier in this section).

[2]The symbol ▭ implies that the flag is established by law and/or officially adopted somehow, so the ⓪ symbol is not needed if ▭ is present.

5 Symbol Size

The vexillogical symbols should scale properly with font size:

\tiny text N

\scriptsize text N

\footnotesize text N

\small text N

\normalsize text N

\large text N

\Large text N

\LARGE text N

\huge text N

\Huge text N

6 Implementation

The `extended` option causes *all* **\vexusage** commands to produce an "extended" symbol (3×3 grid).

```
\newif\ifvexallextended
\DeclareOption{extended}{\vexallextendedtrue}
\ProcessOptions
```

This package requires Ti*k*Z and its `arrows.meta` library.

```
\RequirePackage{tikz}
\usetikzlibrary{arrows.meta}
```

We scale each symbol relative to the height of the letter x in the current font so they scale with text size. The height of the 2×3 grid is intended to match the height of the baseline–capline distance (the height of most capital letters).

```
\newcommand*{\vexxscale}{0.8ex}
\newcommand*{\vexyscale}{0.8ex}
```

The **\vexgrid** macro draws the ordinary 2×3 grid; the **\vexgridextended** macro draws the "extended" grid (3×3).

```
\newcommand*{\vexgrid}{%
  \draw (0,0) -- (3,0)
    (1,-1) -- (1,1)
    (2,-1) -- (2,1);
}
\newcommand*{\vexgridextended}{%
  \draw (0,0) -- (3,0)
    (0,1) -- (3,1)
```

```
    (1,-1) -- (1,2)
    (2,-1) -- (2,2);
}
```

These commands actually draw the dots.

```
\newcommand*{\vexdotone}{%
  \fill (0.5,0.5) circle (0.35);
}
\newcommand*{\vexdottwo}{%
  \fill (1.5,0.5) circle (0.35);
}
\newcommand*{\vexdotthree}{%
  \fill (2.5,0.5) circle (0.35);
}
\newcommand*{\vexdotfour}{%
  \fill (0.5,-0.5) circle (0.35);
}
\newcommand*{\vexdotfive}{%
  \fill (1.5,-0.5) circle (0.35);
}
\newcommand*{\vexdotsix}{%
  \fill (2.5,-0.5) circle (0.35);
}
\newcommand*{\vexdotseven}{%
  \fill (0.5,1.5) circle (0.35);
}
\newcommand*{\vexdoteight}{%
  \fill (1.5,1.5) circle (0.35);
}
\newcommand*{\vexdotnine}{%
  \fill (2.5,1.5) circle (0.35);
}
```

The **\vexusage** takes a whole-number argument $(0, 1, \ldots)$. Numbers will be interpreted modulo 512 (e.g., if 513 is passed to this function, it will be interpreted as $513 \mod 512 = 1$), which is accomplished by repeated integer division modulo 2.

```
\NewDocumentCommand{\vexusage}{s m}{%
\begin{tikzpicture}[x=\vexxscale,y=\vexyscale]
  \IfBooleanTF {#1}
    {\vexgridextended}
    {
      \ifvexallextended
        \vexgridextended
      \else
        \pgfmathparse{int(#2)}
        \ifnum\pgfmathresult>63\relax
          \vexgridextended
        \else
          \vexgrid
```

```
        \fi
      \fi
    }
  \pgfmathparse{int(mod(#2,2))}
  \ifnum\pgfmathresult=1\relax
    \vexdotone
  \fi
  \pgfmathparse{int(mod(div(#2,2),2))}
  \ifnum\pgfmathresult=1\relax
    \vexdottwo
  \fi
  \pgfmathparse{int(mod(div(#2,4),2))}
  \ifnum\pgfmathresult=1\relax
    \vexdotthree
  \fi
  \pgfmathparse{int(mod(div(#2,8),2))}
  \ifnum\pgfmathresult=1\relax
    \vexdotfour
  \fi
  \pgfmathparse{int(mod(div(#2,16),2))}
  \ifnum\pgfmathresult=1\relax
    \vexdotfive
  \fi
  \pgfmathparse{int(mod(div(#2,32),2))}
  \ifnum\pgfmathresult=1\relax
    \vexdotsix
  \fi
  \pgfmathparse{int(mod(div(#2,64),2))}
  \ifnum\pgfmathresult=1\relax
    \vexdotseven
  \fi
  \pgfmathparse{int(mod(div(#2,128),2))}
  \ifnum\pgfmathresult=1\relax
    \vexdoteight
  \fi
  \pgfmathparse{int(mod(div(#2,256),2))}
  \ifnum\pgfmathresult=1\relax
    \vexdotnine
  \fi
\end{tikzpicture}}
```

These user-facing macros are intended as arguments to the **\vexusage** macro. Ideally, we would bit-wise AND the argument, but because most of these do not share bits, it is sufficient to add them.

```
\newcommand*{\vexcivilflag}{1}
\newcommand*{\vexstateflag}{2}
\newcommand*{\vexwarflag}{4}
\newcommand*{\vexcivilensign}{8}
```

```
\newcommand*{\vexstateensign}{16}
\newcommand*{\vexwarensign}{32}
\newcommand*{\vexcivilairensign}{64}
\newcommand*{\vexstateairensign}{128}
\newcommand*{\vexwarairensign}{256}
\newcommand*{\vexnationalflag}{7}
\newcommand*{\vexnationalensign}{56}
\newcommand*{\vexnationalflagandensign}{63}
\newcommand*{\vexnationaleverything}{511}
```

These are drawn based on the images in Smith's book as well as the images on the Wikipedia.

```
\newcommand*{\vexnormal}{%
\begin{tikzpicture}[x=\vexxscale,y=\vexyscale]
  \draw (0,-0.4) -- (0,1) -- (2,1) -- (2,0) -- (0,0);
\end{tikzpicture}}
\newcommand*{\vexproposed}{%
\begin{tikzpicture}[x=\vexxscale,y=\vexyscale]
  \draw (0,-0.4) -- (0,1) -- (2,1) (2,0) -- (0,0);
\end{tikzpicture}}
\newcommand*{\vexreconstruction}{%
\begin{tikzpicture}[x=\vexxscale,y=\vexyscale]
  \draw (0,-0.4) -- (0,1) -- (2,1) -- (2,0) -- (0,0);
%  \node at (1,0.5) {\resizebox{!}{0.5ex}{?}};% FIXME?
  \fill (0.9875,0.2) circle (0.075*\vexxscale);
  \draw [line width=0.125*\vexyscale,line cap=round]
    (0.9875,0.381) .. controls +(0,0.25) and +(0,-0.12) .. (1.125,0.70)
    .. controls +(0,0.09) and +(0.05,0) .. (1,0.8125)
    .. controls +(-0.09,0) and +(0,0.065) .. (0.85,0.702);
\end{tikzpicture}}
\newcommand*{\vexreverse}{%
\begin{tikzpicture}[x=\vexxscale,y=\vexyscale]
  \draw (0,-0.4) -- (0,1) -- (2,1) -- (2,0) -- (0,0);
  \draw [-{Straight Barb[length=0.25ex,width=0.25ex]}] (1,0.5) -- (2.75,0.5);
\end{tikzpicture}}
\newcommand*{\vexvariant}{%
\begin{tikzpicture}[x=\vexxscale,y=\vexyscale]
  \draw (0,-0.4) -- (0,1) -- (2,1) -- (2,0) -- (0,0);
  \draw (-0.2,-0.5) .. controls +(-0.4,0.3) and +(-0.4,-0.3) .. (-0.2,1.1);
  \draw (2.2,-0.5) .. controls +(0.4,0.3) and +(0.4,-0.3) .. (2.2,1.1);
\end{tikzpicture}}
\newcommand*{\vexalternative}{%
\begin{tikzpicture}[x=\vexxscale,y=\vexyscale]
  %\draw[green] (0,-0.4) -- (0,1) -- (2,1) -- (2,0) -- (0,0);
  \draw (0,-0.4) -- (0,1) -- (2,1) -- (2,0.5)
    (0.5,0) -- (0,0);
  \draw [xshift=0.4ex,yshift=-0.4ex]
    (0,-0.4) -- (0,1) -- (2,1) -- (2,0) -- (0,0);
```

```
\end{tikzpicture}}
\newcommand*{\vexdefacto}{%
\begin{tikzpicture}[x=\vexxscale,y=\vexyscale]
  \draw (0,-0.4) -- (0,1) -- (2,1) -- (2,0) -- (0,0);
  \draw (0.5,-0.2) -- (1.5,1.2);
\end{tikzpicture}}
\newcommand*{\vexreverseisdifferent}{%
\begin{tikzpicture}[x=\vexxscale,y=\vexyscale,
    >={Straight Barb[length=0.25ex,width=0.25ex]}]
  \draw [->] (0,0) -- (2,0);
  \draw [->] (1.5,0.5) -- (-0.5,0.5);
  \draw (0.25,-0.5) -- (1.25,1);
\end{tikzpicture}}
\newcommand*{\vexsinisterhoist}{%
\begin{tikzpicture}[x=\vexxscale,y=\vexyscale]
  \draw (0,0) -- (0,1) -- (2,1) -- (2,-0.4) -- (2,0) -- cycle;
\end{tikzpicture}}
\newcommand*{\vexhistorical}{%
\begin{tikzpicture}[x=\vexxscale,y=\vexyscale,line cap=rect]
  \draw (0,-0.4) -- (0,1) -- (1.2,1) -- (0.6,0) -- (0,0);
  \draw (2,-0.4) -- (2,0.7) -- (1.5,0.7) -- (0.8,-0.4) -- cycle;
\end{tikzpicture}}
\newcommand*{\vexreverseismirror}{%
\begin{tikzpicture}[x=\vexxscale,y=\vexyscale,line cap=rect,
    >={Straight Barb[length=0.25ex,width=0.35ex]}]
  \draw (2,-0.5) -- (2,1) -- (0,1) -- (0,0) -- (2,0);
  \draw [->] (0.25,-0.125) .. controls +(0.25,-0.20) and +(-0.25,0) .. (1.75,-0.25);
  \draw [line cap=round] (0.75,0.15) -- ++(0.25,0.35)
    (1.25,0.15) -- (1.25,0.85)
    -- (0.85,0.85) .. controls (0.70,0.8) and (0.70,0.6) .. (0.85,0.5)
    -- (1.25,0.5);
\end{tikzpicture}}
\newcommand*{\vexreverseiscongruent}{%
\begin{tikzpicture}[x=\vexxscale,y=\vexyscale,line cap=rect,
    >={Straight Barb[length=0.25ex,width=0.35ex]}]
  \draw (2,-0.5) -- (2,1) -- (0,1) -- (0,0) -- (2,0);
  \draw [->] (0.25,-0.125) .. controls +(0.25,-0.20) and +(-0.25,0) .. (1.75,-0.25);
  \draw (1.25,0.15) -- (0.75,0.15) -- (0.75,0.85) -- (1.25,0.85)
    (0.75,0.5) -- (1.15,0.5);
\end{tikzpicture}}
\newcommand*{\vexnormalvertical}{%
\begin{tikzpicture}[x=\vexxscale,y=\vexyscale,line cap=rect]
  \draw (2.125,1.25) -- (-0.125,1.25) -- (-0.125,-0.25)
    -- (1.125,-0.25) -- (1.125,1.25);
  \begin{scope}
    \clip (0.1,0.1) rectangle (0.9,0.9);
    \draw (0.85,0.15) -- (0.15,0.15) -- (0.85,0.85) -- (0.15,0.85);
```

```
  \end{scope}
\end{tikzpicture}}
\newcommand*{\vexobversevertical}{%
\begin{tikzpicture}[x=\vexxscale,y=\vexyscale,line cap=rect]
  \draw (-0.125,1.25) -- (-0.125,-0.25)
    -- (1.125,-0.25) -- (1.125,1.25) -- (-1.125,1.25);
  \begin{scope}
    \clip (0.1,0.1) rectangle (0.9,0.9);
    \draw (0.85,0.15) -- (0.15,0.15) -- (0.85,0.85) -- (0.15,0.85);
  \end{scope}
\end{tikzpicture}}
\newcommand*{\vexrotatevertical}{%
\begin{tikzpicture}[x=\vexxscale,y=\vexyscale,line cap=rect]
  \draw (2.125,1.25) -- (-0.125,1.25) -- (-0.125,-0.25)
    -- (1.125,-0.25) -- (1.125,1.25);
  \draw [line cap=round] (0.20,0.1) -- (0.20,0.9) -- (0.75,0.9)
    .. controls (0.90,0.7625) and (0.90,0.5875) .. (0.75,0.5) -- (0.20,0.5)
    (0.55,0.5) -- (0.82,0.1);
\end{tikzpicture}}
\newcommand*{\vexunknownvertical}{%
\begin{tikzpicture}[x=\vexxscale,y=\vexyscale,line cap=rect]
  \draw (2.125,1.25) -- (-0.125,1.25) -- (-0.125,-0.25)
    -- (1.125,-0.25) -- (1.125,1.25);
%  \node at (0.5,0.5) {\resizebox{!}{0.5ex}{?}};% FIXME?
  \fill (0.49375,0.2) circle (0.075*\vexxscale);
  \draw [line width=0.125*\vexyscale,line cap=round]
    (0.49375,0.381) .. controls +(0,0.25) and +(0,-0.12) .. (0.64125,0.70)
    .. controls +(0,0.09) and +(0.05,0) .. (0.50625,0.8125)
    .. controls +(-0.09,0) and +(0,0.065) .. (0.35625,0.702);
\end{tikzpicture}}
\newcommand*{\vexnorotating}{%
\begin{tikzpicture}[x=\vexxscale,y=\vexyscale,line cap=rect]
  %\draw (2,1) -- (0,1) -- (0,0) -- (1,0) -- (1,1);
  \draw (2.125,1.25) -- (-0.125,1.25) -- (-0.125,-0.25)
    -- (1.125,-0.25) -- (1.125,1.25);
\end{tikzpicture}}
\newcommand*{\vexunauthorized}{%
\begin{tikzpicture}[x=\vexxscale,y=\vexyscale]
  \draw (0,-0.4) -- (0,1) -- (2,1) -- (2,0) -- (0,0);
  \draw (0.5,-0.2) -- (1.5,1.2);
  \draw (1.5,-0.2) -- (0.5,1.2);
\end{tikzpicture}}
\newcommand*{\vexonlyvertical}{%
\begin{tikzpicture}[x=\vexxscale,y=\vexyscale]
  \draw (0,0) rectangle (1.5,1.5);
  \draw (0.75,0) -- ++(0,-0.4);
  \draw (0.75,1.5) -- ++(0,0.2);
```

```
\end{tikzpicture}}
\newcommand*{\vexauthorized}{%
\begin{tikzpicture}[x=\vexxscale,y=\vexyscale]
  \draw (0,0) circle (0.75);
  \draw (0,0) ellipse (0.25 and 0.375);
\end{tikzpicture}}
```

Change History

v1.0.0

General: Initial public release 1

Index

Numbers written in italics refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition.