The Implementation of NTS

Karel Skoupý & Philip Taylor

NTS: The Structure

the implementation language of NTS is Java

the program code is encapsulated in classes

objects are instances of (sub)classes

classes are clustered in packages

The initialization

```
\def \initialisation
  {\nonstopmode

   \input init.inc

   \tracingcommands = 0 \tracingonline = 0 \tracingparagraphs = 0

   \time = 750 \showboxdepth = 100 \showboxbreadth = 1000000

   \baselineskip = 12pt \lineskiplimit = 0pt \lineskip = 1pt

   \def \NTS {{\tenit NTS}}

   \font \tenrm = cmr10  \font \tenit = cmti10 \tenrm
  }
```

A normal paragraph

```
\def \normalpar
  {\parindent = 0 pt
  %\adjdemerits = 10
   \pretolerance = 300
   \tolerance = 300
  }
```

A narrow paragraph

```
\def \narrowpar
  {\hsize = 0,5\hsize
   \tolerance = 9999
   \leftskip = 0.2\hsize
  }
```

A centered paragraph

```
\def \centeredpar
  {\leftskip = 0.5\hsize plus 1 fil
   \rightskip = \leftskip
   \parindent = 0 em
   \parfillskip = \parindent
   \hsize = 2\hsize
  }
```

A few rules

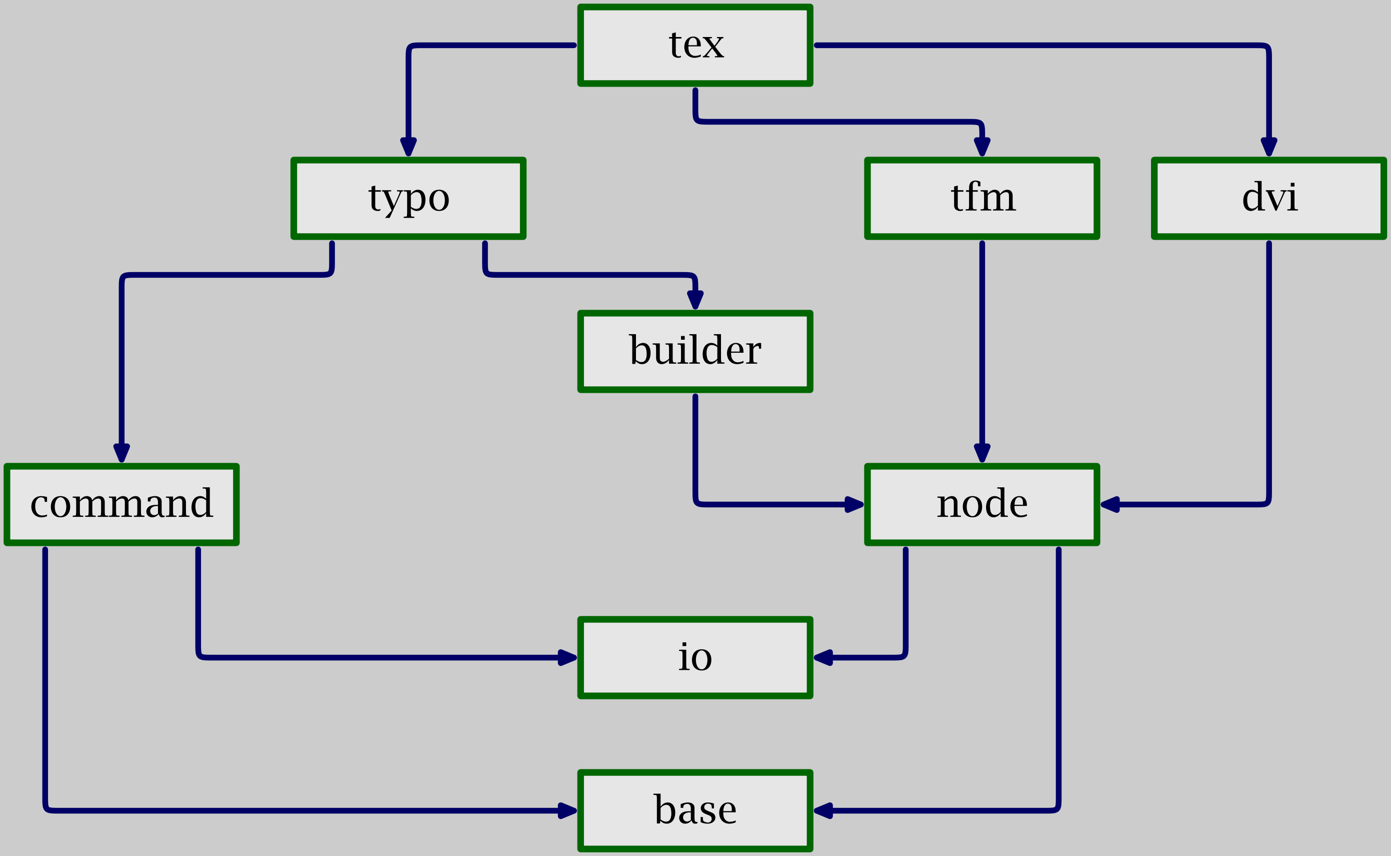
```
\def \divider #1%
  {\ifcase #1
     \message {No zeroth divider class}
   \or
     \par \hrule \par
   \or
     \par
     \vskip 10pt
     \hrule height 1 pt depth 1 pt
     \vskip 10pt
     \par
   \else
     \message {No divider class > 2}
   \fi
  }
```

The text

```
\def \text
  {Other authors in this series of papers on \NTS\ have
   explained the reasons for the creation of the \NTS~project
   (Joachim Lammarsch), the rationale behind the choice of
   programming language (Ji\v{r}\'{\i} Zlatu\v{s}ka), and
   future directions in which the project may develop (Hans
   Hagen). This paper addresses the rather more detailed area
   of the actual implementation itself, and is intended to
   provide the reader with as much detail as can reasonably be
   accommodated in a paper which is intended to appear in the
   Conference Proceedings. A considerably more detailed version
   of the paper will eventually be available as an accompaniment
   to (or possibly integrated in) the JavaDoc documentation
   which will accompany the released version of \NTS.
  }
```

The typesetting

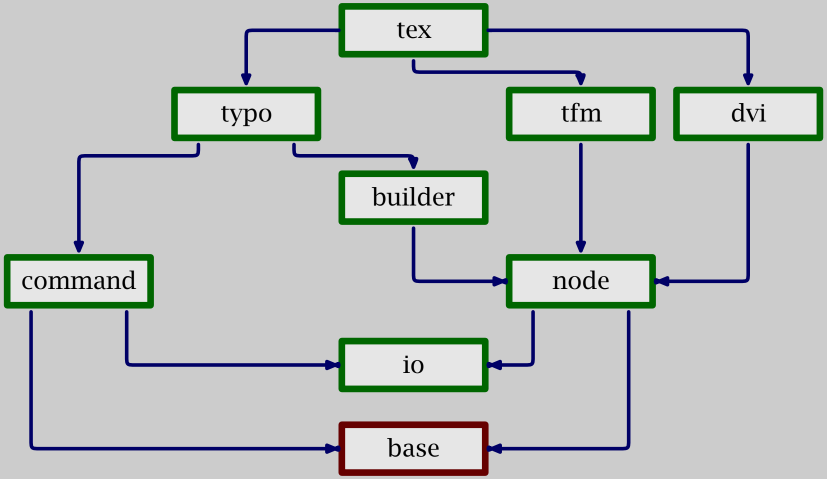
```
\setbox 1 = \hbox {\text}

\shipout \vbox
  {\normalpar
   \unhcopy 1
   \divider 1
   \narrowpar
   \unhcopy 1
   \divider 2
   \centeredpar
   \unhcopy 1
  }

\end
```



NTS Java packages

base

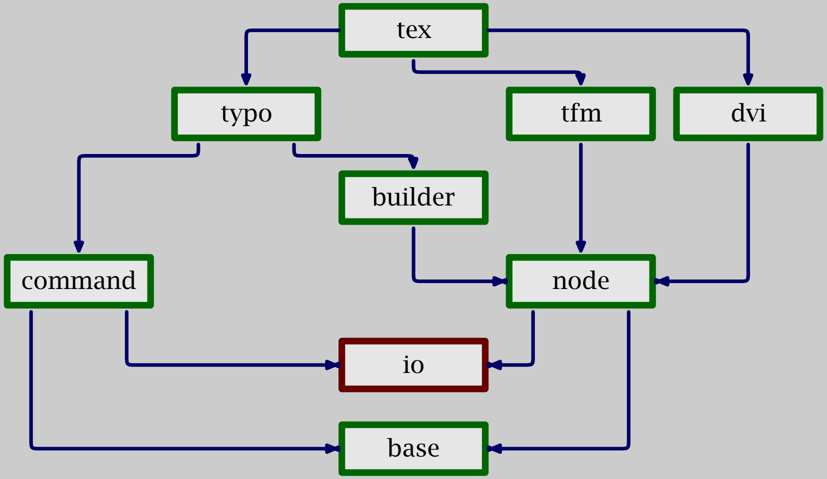
implements elementary data types



Dimen Glue Num LevelEqTable

io

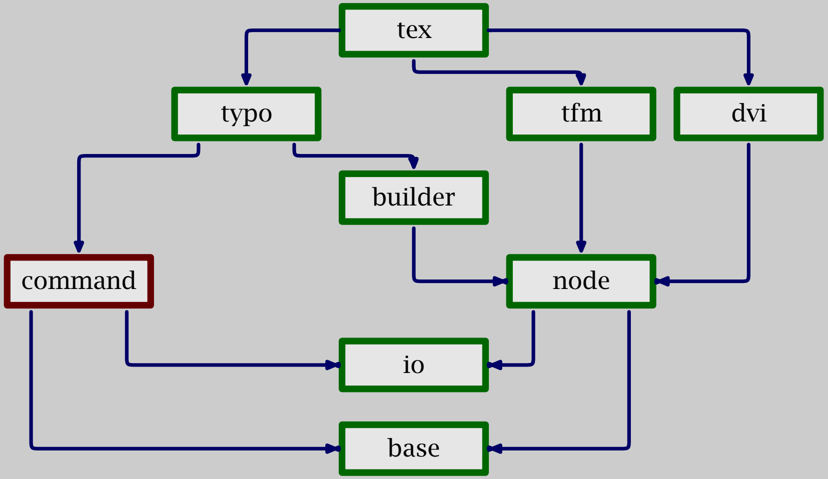
handles reading from input and writing to the log file



CharCode   Name   InputLine   Log   Loggable

command

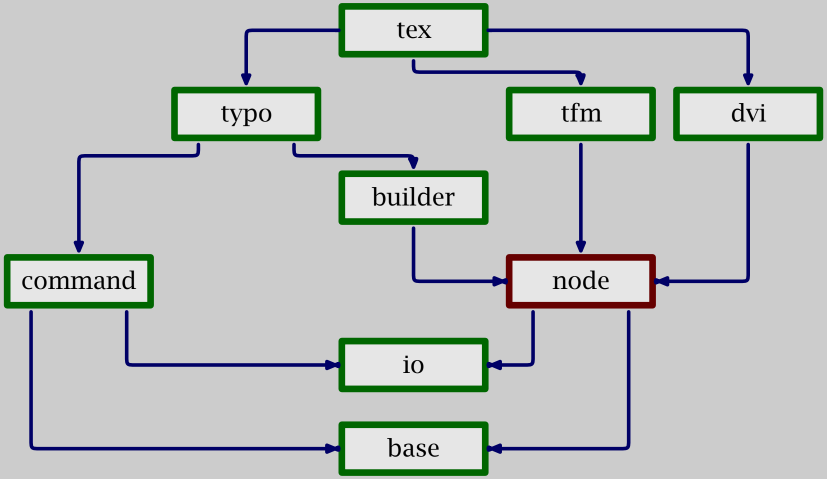
interprets the TEX input language



Token Tokenizer Command CommandBase

node

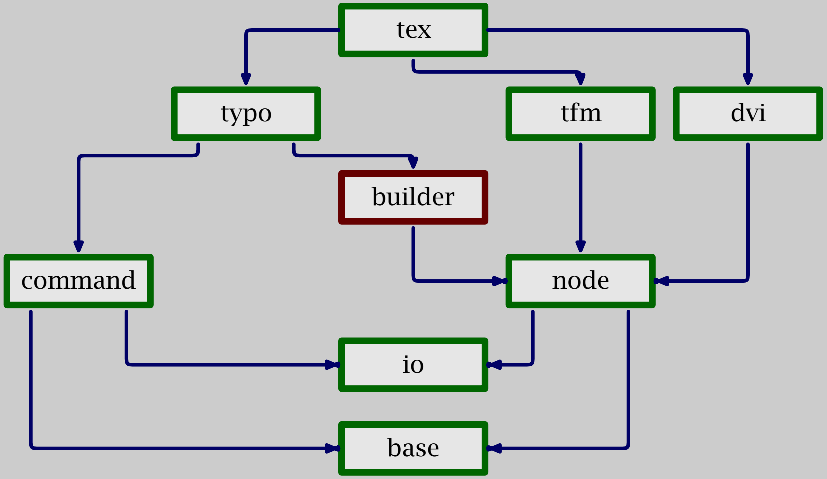
represents the material to be typeset



Node Packer FontMetric TypeSetter

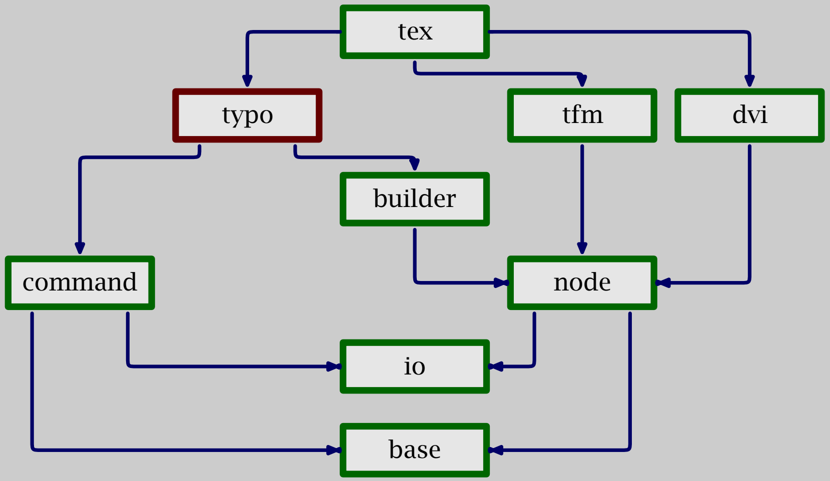
builder

takes care of mode-related things



typo

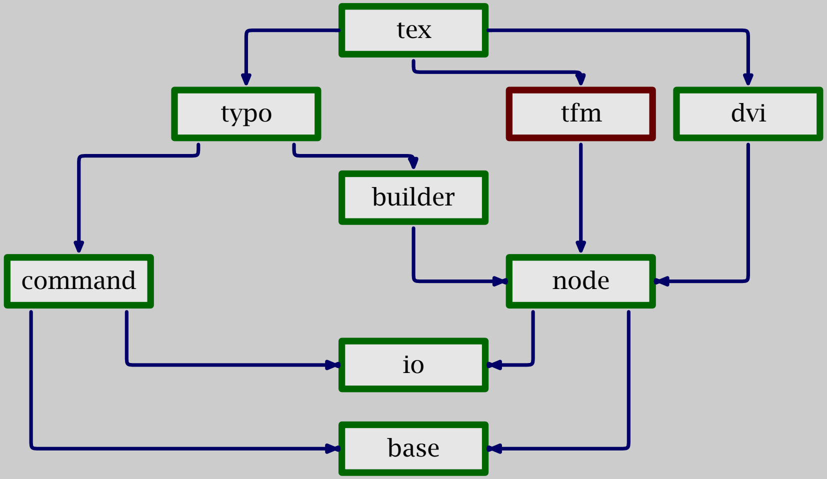
deals with typesetting



TypoCommand BuilderCommand Group

tfm

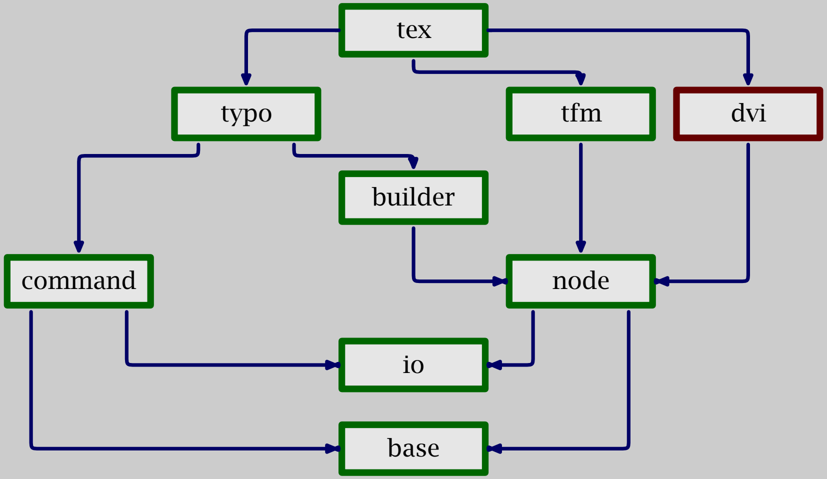
handles the natural TEX font metrics



TeXFm TeXFontMetric

dvi

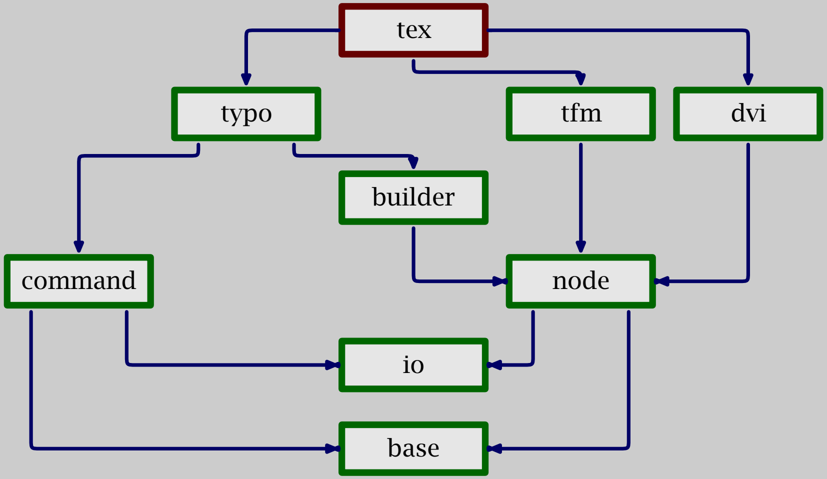
takes care of TeX's native output format



DviFormatSetter DviTypeSetter

tex

glues everything into a program



A few concluding remarks

compatibility with TeX is important to gain
user confidence and widespread acceptance

quite some time has been invested in obtaining strict compatibility with TeX

in spite of extensive documentation of TeX–the–
program, much in–depth study of the code was needed

Java still has kept its promise, but the heavy
use of objects will have a performance penalty