
Typesetting with T_EX / L_AT_EX

Part I: Basic Components and Essential L_AT_EX

F. C. Langbein

School of Computer Science
Cardiff University





Overview

- Introduction to the $\text{T}_{\text{E}}\text{X}$ / $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ typesetting system
 - Components and software tools
 - The $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ typesetting language for technical documents
- **Part I:** basic components and essential $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$
- Part II: formatting and layout
- Part III: figures and tables
- Part IV: basic mathematics and $\text{AMS}_{\text{L}}\text{A}_{\text{T}}\text{E}_{\text{X}}$
- Part V: $\text{PDF}_{\text{L}}\text{A}_{\text{T}}\text{E}_{\text{X}}$ and slides
- Part VI: $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$ and MakeIndex
- Part VII: useful things...

What is T_EX / L_AT_EX?

- T_EX is a computer **typesetting system**
(created by D. Knuth while writing “The Art of Computer Programming”)
 - T_EX is really a complete **programming language** aimed at creating documents
 - Basic T_EX language can be expanded by macros
 - **Formats** are large macro sets for layout, formatting, etc.
- L_AT_EX is a particular T_EX format created by Leslie Lamport
 - Set of macros to specify documents on a high level
 - In many ways L_AT_EX is similar to SGML/XML/HTML
 - Really **L_AT_EX 2_ε**, an intermediate step to L_AT_EX3
- Other formats: plain tex, extended plain tex, ...



Why L^AT_EX?

- You can easily create beautiful, long, complex documents
 - It **knows** a lot about typesetting
 - It produces **exact** rather than approximate results
(no approximate WYSIWYG system)
 - Easy to **modify and expand** using macros
 - A set of **well-designed tools** to prepare documents
(no big expensive software package that often fails in mysterious ways)
- Supported on (nearly) all computer platforms
 - Free and commercial versions available for every platform
 - T_EX document sources are in **plain text** (ASCII)



TEX / L^AT_EX Resources

- The course web site at
<http://www.langbein.org/teaching/latex/>
- Comprehensive TEX Archive Network (CTAN) at
<http://www.ctan.org/>
- TEX Users Group (TUG) at
<http://www.tug.org/>
- TEX newsgroup at `comp.text.tex`
- M. Gossens, F. Mittelbach, A. Samarin, The L^AT_EX Companion. Addison Wesley, 1994
- H. Kopka, P. Daly, A Guide To LaTeX. Addison Wesley, 2003.

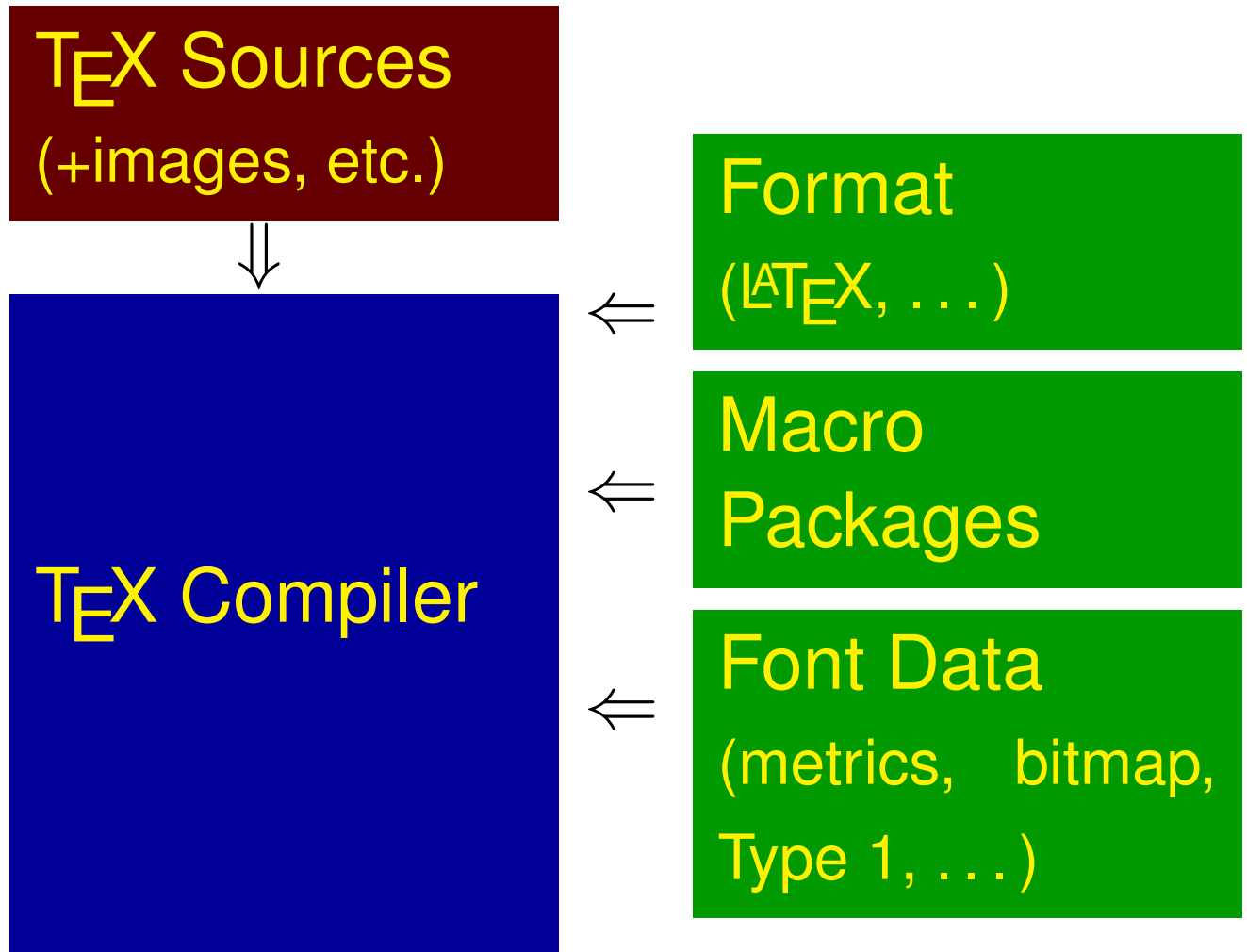
Components of the T_EX System

T_EX Sources
(+images, etc.)

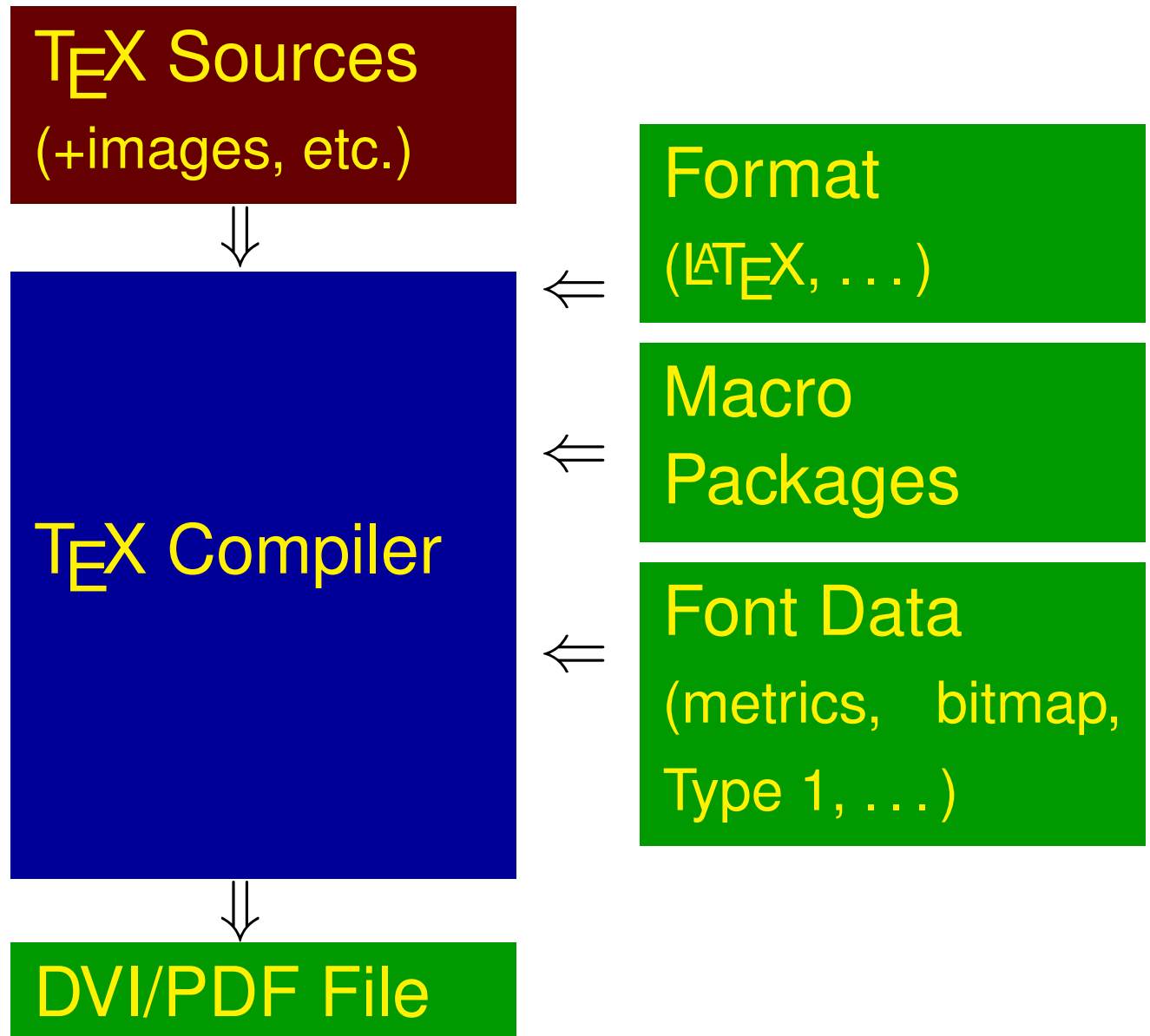


T_EX Compiler

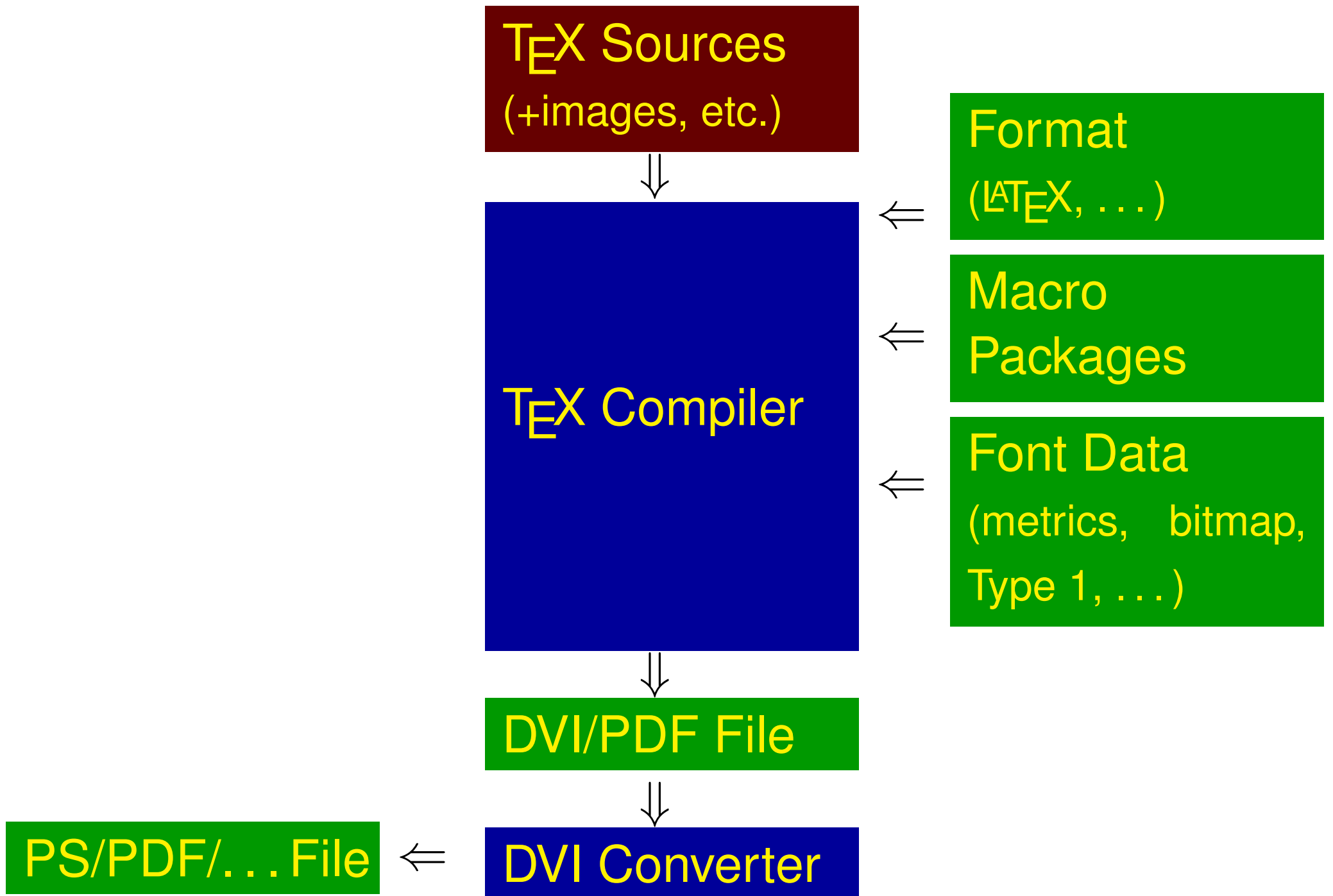
Components of the T_EX System



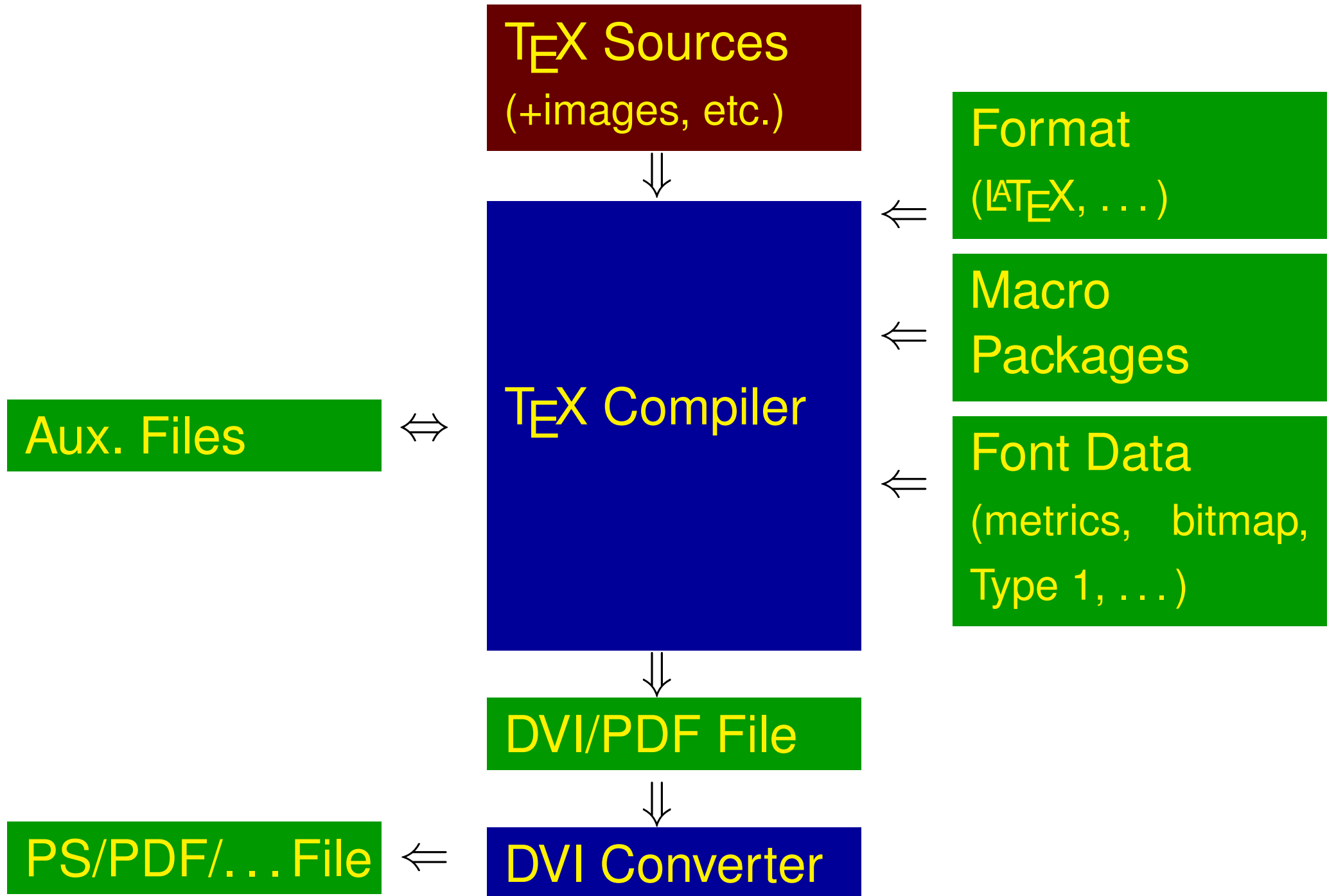
Components of the T_EX System



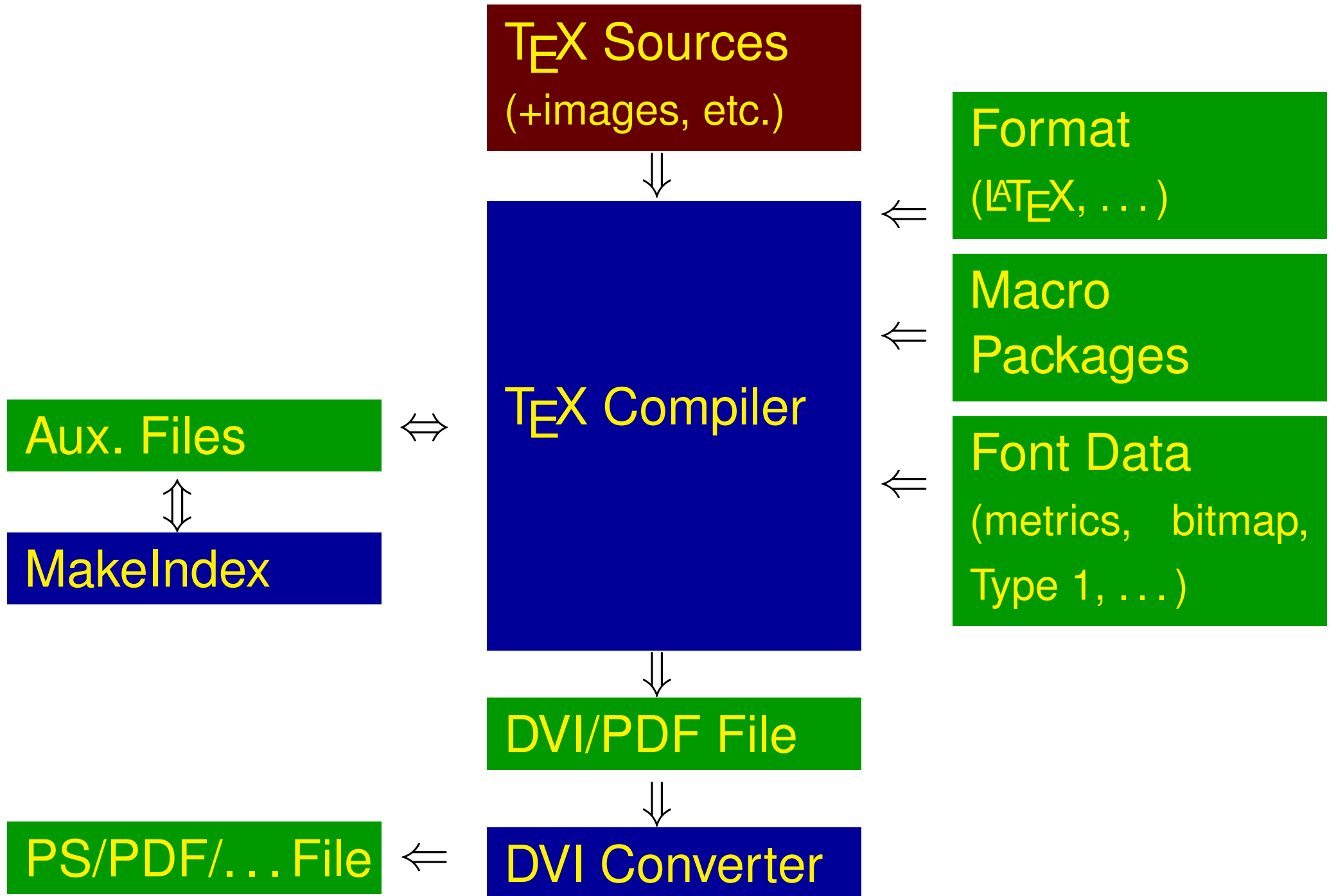
Components of the T_EX System



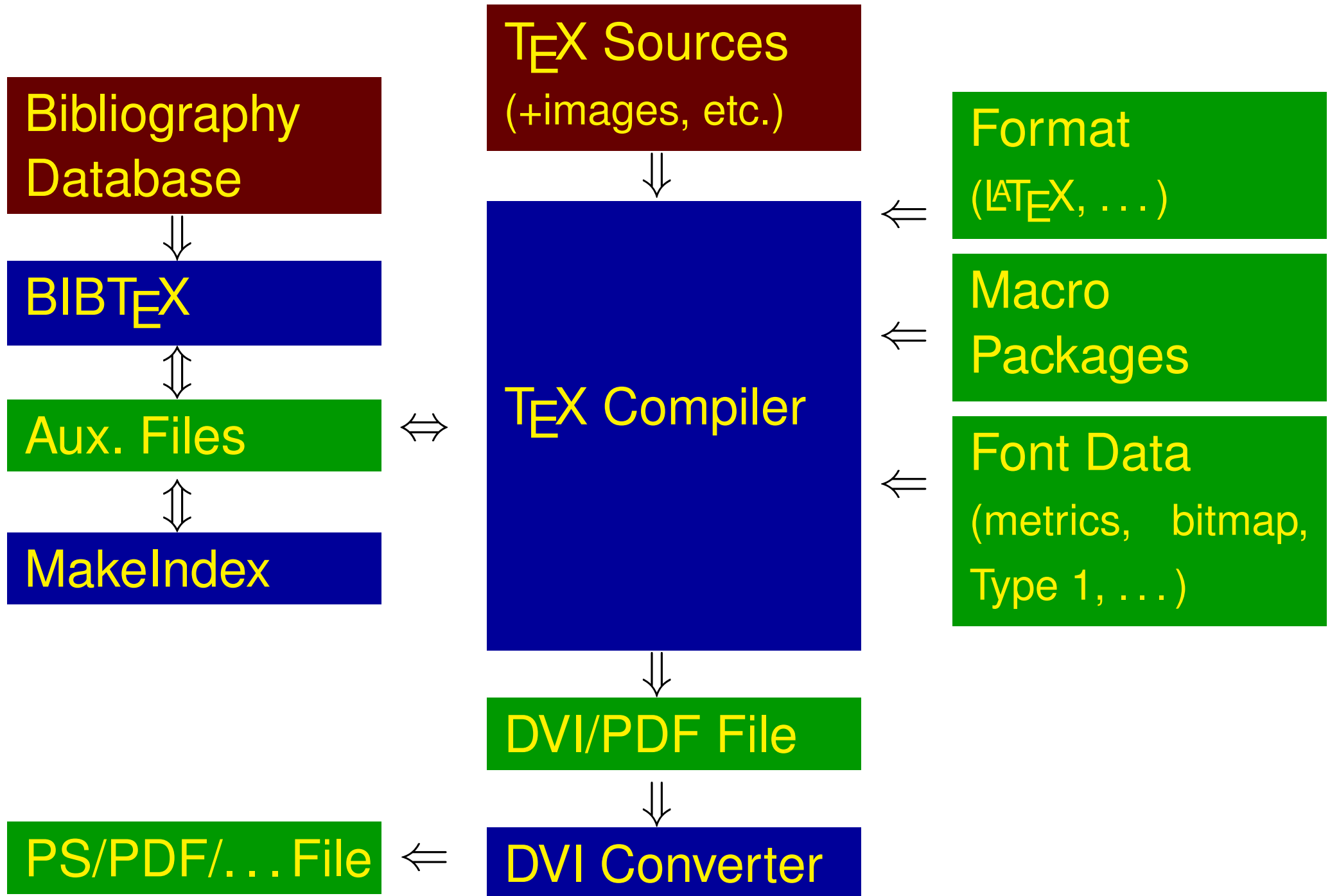
Components of the T_EX System



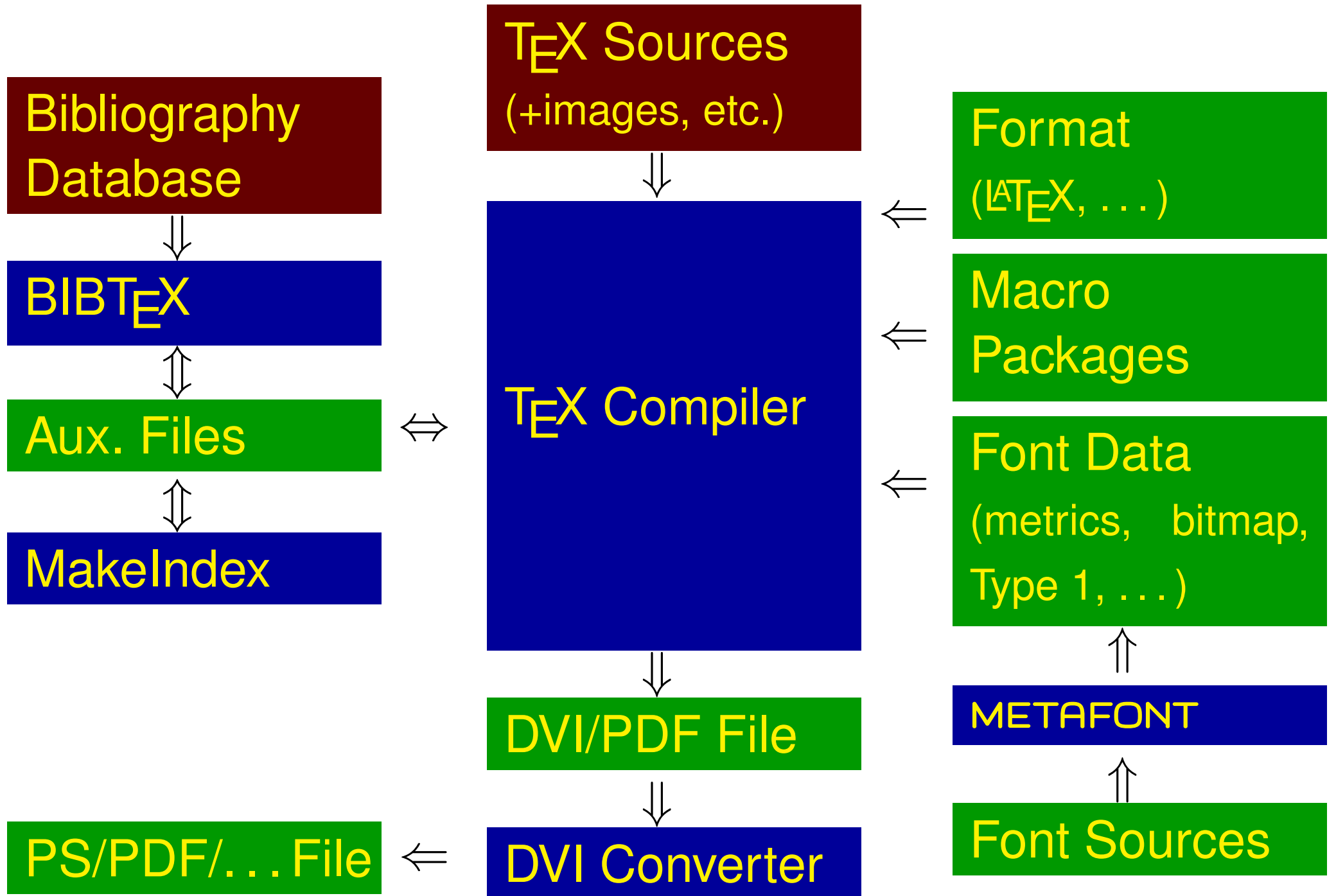
Components of the T_EX System



Components of the T_EX System



Components of the T_EX System



Basic Structure and Syntax

Overall Document Structure

➤ Every \LaTeX document has the following structure:

```
\documentclass [options] {class}  
preamble  
\begin{document}  
    document body  
\end{document}
```

- **class:** **document type**, e.g. article, report, book
- **options:** optional list of document type **modifications**
- **preamble:** contains **formatting, layout, etc.** information and inclusion of additional macro packages

- document body: actual contents of the document

L^AT_EX Commands

➤ Three versions of commands:

- **Non-letter characters** &, \$, %, ~, -, {, }, #, ^ have special meaning telling L^AT_EX to do something
- **Backslash ** followed by **single non-letter character**
(e.g. any of the above characters can be produced by adding \)
- **Backslash ** followed by **one or more letters**
- Letter/non-letter characters are defined explicitly!
(i.e. they may be redefined sometimes)

➤ Commands generally have the structure:

```
\COMMAND [optarg] {mandarg}
```

- There may be none, one or multiple optional / mandatory arguments

Environments

- An **environment groups segments** of code
 - The body of an environment is treated differently from the “outside”
 - General syntax:

```
\begin{environment}  
    body of environment  
\end{environment}
```

- For example, a `center` environment centring its body:

```
\begin{center}  
    centred \scshape  
text  
\end{center}
```

```
text before environment  
    centred TEXT  
text after environment
```

- Changes inside an environment are usually local

Grouping

➤ Text can be **grouped** using { **some text** }

- Changes inside group are **local**

- For example, to emphasise the text in the group:

```
This is {\em important}
text
```

This is *important* text

- Effect of command (or declaration) `\em` ends with end of group

➤ An argument to a command is similar to a group, but the command is outside, e.g.

```
This is \emph{important} text
```

- Yields the same result, but the *different* command `\emph` takes the text as argument

Characters, Words, Paragraphs

- \LaTeX regards **groups of characters separated by spaces** (even multiple spaces) or newlines as **words**
- A **blank line** (or multiple blank lines together) tells \LaTeX to begin a **new paragraph** (or use command `\par`)
- `%` indicates a **comment** and everything until the end of line is ignored (incl. newline character)

```
This is an      example  
paragraph.
```

```
The next % A comment  
parag% Comment 2  
raph starts here.
```

```
This is an example para-  
graph.
```

```
The next paragraph starts  
here.
```

Sentences

➤ Sentences end with . ? !

● L^AT_EX inserts **extra space** after these symbols

➤ L^AT_EX tries to be smart in that if the letter before a period is upper case, it treats it as an abbreviation:

```
A. Foo sent a proposal  
to the CRSPE. How nice.
```

```
A. Foo sent a proposal to  
the CRSPE. How nice.
```

➤ Causes problems when sentence ends in upper case letter

● Fix this by adding \@:

```
A. Foo sent a proposal  
to the CRSPE\@. How  
nice.
```

```
A. Foo sent a proposal to  
the CRSPE. How nice.
```

● The additional space can be turned off by \frenchspacing

Spaces

➤ A **forced space** is inserted with `\`

● To force a standard space after `.` `!` `?` etc.:

E.g. `\` consider this
example

E.g. consider this example

➤ A **non-breaking space** is inserted with `~`:

`Do~not~break~these~spaces.`

➤ There is no space after a command without argument:

`\TeX pert`

`TEXpert`

● Insert empty group or forced space to get a space:

`\TeX{} pert`, `\TeX\` pert

`TEX pert`, `TEX\` pert

New Lines

- **New line** can be forced with `\\ [length]`:

```
First\\  
Second\\ [5mm]  
Third
```

First
Second
Third

- `\linebreak [length]` breaks line, but **keeps text justified**:

```
A short \linebreak  
justified paragraph.
```

A short justified paragraph.

- `\nolinebreak` **prevents line break**:

```
A short fully  
justified\nolinebreak  
paragraph.
```

A short fully justified paragraph.

Page breaks

- To force a **ragged page break**:

```
\newpage
```

- To force a **vertically justified page break**:

```
\pagebreak[n]
```

- Optional argument $n = 0, \dots, 4$ makes page break a request (the higher, the more insistent)

- To **prevent a page break**:

```
\nopagebreak[n]
```

- To force a page break, and process all unprocessed floats (tables, figures):

```
\clearpage, \cleardoublepage
```

Quotes, Hyphens and Dashes

➤ Almost never use " character. Instead use:

‘single quotes’

‘‘double quotes’’

‘single quotes’

“double quotes”

➤ Depending on context **different types of hyphens** are used:

- **hyphen** (-) used for compound words and end-of-line hyphenation
- **en dash** (- -) used for ranges such as in pp. 5–10
- **em dash** (- - -) used in punctuation—similar, though not identical, to a colon
- **minus sign** (\$-\$) used in math mode, as in $x - y$