



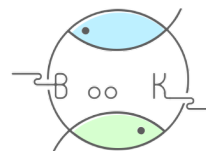
L^AT_EX 入门

作者：刘海洋

项目：GitHub [lihui588211/BookRewriting](#)

时间：2023/10/2

书籍模板：ElegantBook



重写书籍只为学习，请勿商用

目录

第一章 熟悉 L ^A T _E X	1
1.1 让 L ^A T _E X 跑起来	1
1.1.1 L ^A T _E X 的发行版及其安装	1
1.1.1.1 C T _E X 套装	2
1.1.1.2 T _E X Live	2
1.1.2 编辑器与周边工具	3
1.1.2.1 编辑器举例——TeXworks	3
1.1.2.2 PDF 阅读器	6

第一章 熟悉 L^AT_EX

L^AT_EX 是一种基于 T_EX 的文档排版系统。T_EX 只这么交错起伏的几个字母，便道出了“排版”二字的几分意味：精确、复杂、注重细节和品味。而 L^AT_EX 则为了减轻这种写作、排版一肩挑的负担，把大片排版的格式隐藏在若干样式之后，以内容的逻辑结构统帅纷繁的格式，使之成为现在最流行的科技写作——尤其是数学写作的工具之一。

无论你是因为心慕 L^AT_EX 漂亮的输出结果，还是因为要写论文投稿被逼上梁山，都不得不面对一个事实：L^AT_EX 是一种并不简单的计算机语言，不能只点点鼠标就弄好一篇漂亮的文章，也不是一两个小时的泛泛了解就尽能对付得过去。^[1]

L^AT_EX 的读音和写法

T_EX 一名源自 technology 的希腊词根 τεχ_η，T_EX 之父高德纳教授近乎固执地要求它的发音必须是（按国际音标）[tɛx]，尽管英语中它常被读作 [tɛk]。（同样，高德纳教授也近乎固执地要求别人说他的姓 Knuth 时不要丢掉 K，叫他 Ka-NOOTH，尽管在英语环境他时常会变成 Nooth 教授。）对比汉语，T_EX 的发音近似于“泰赫”。

L^AT_EX 这个名字则是把 L^AT_EX 之父 Lamport 博士的姓和 T_EX 混合得到的。所以 L^AT_EX 大约应该读成“拉泰赫”。不过人们仍然按照自己的理解和拼写发音习惯去读它：[ˈla:tɛk]、[ˈlei:tɛk] 或是 [laːˈtɛk]，甚至不怎么合理的 [ˈleɪtɛks]^[2]。好在 Lamport 并不介意 L^AT_EX 到底被读做什么。“读音最好由习惯决定，而不是法令。”——Lamport 如是说。

两个创始人对于名称和读音的不同态度或许多少说明了这样一个事实：L^AT_EX 相对原始的 T_EX 更少关注排版的细节，因此 L^AT_EX 在很多时候并不充当专业排版软件的角色，而只是一个文档编写工具。而当人们在 L^AT_EX 中也抱以追求完美的态度并用到一些平时不大使用的命令时，通常总说这是在 T_EX 层面排版——尽管 L^AT_EX 本身正是运行于 T_EX 之上的。

类似地，T_EX 和 L^AT_EX 字母错位的排版也体现出一种面向排版的专业态度，即使在字符难以错位的场合，也应该按大小写交错写成 TeX 和 LaTeX。

现在我们使用的 L^AT_EX 格式版本为 2_ε，意思是超出了第 2 版，接近却没有达到第 3 版，因此写成 L^AT_EX 2_ε。在只能使用普通字符的场合，一般写成 LaTeX2_ε。

1.1 让 L^AT_EX 跑起来

学习 L^AT_EX 的第一步就是上手试一试，让 L^AT_EX 跑起来。首先安装 T_EX 系统及其他一些必要的软件，然后跑一个测试的例子。下面的几节包含了一大堆具体软件安装和使用的内容，虽然比较烦琐，但这是使用 L^AT_EX 进行写作的必要前提。如果你早已做好这些准备，或者在读书之前就已经迫不及待地做了不少尝试的话，可以直接跳到开始一个实际的例子。

1.1.1 L^AT_EX 的发行版及其安装

T_EX/L^AT_EX 并不是单独的程序，现在的 T_EX 系统都是复杂的软件包，里面包含各种排版的引擎、编译脚本、格式转换工具、管理界面、配置文件、支持工具、字体及数以千计的宏包和文档。一个 T_EX 发行版就是把这样的部件都集合起来，打包发布的软件。

^[1]不过现代人好像都追求快，沉不下心来。这也是我通过此项目强迫自己学习的理由

^[2]额。我的读法...

尽管内容复杂,但现在的 T_EX 发行版的安装还是非常方便的。下面将介绍两个最为流行的发行版,一是 1.1.1.1 的 C_T_EX 套装,二是 1.1.1.2 的 T_EX Live。前者是 Windows 系统下的软件,后者则可以用在各种常用的桌面操作系统上。对 Windows 用户来说,两个发行版并没有显著的优劣之分,你可以任选一个安装使用^[3]。

1.1.1.1 C_T_EX 套装

已过时,跳过。

1.1.1.2 T_EX Live

T_EX Live 是由 TUG (T_EX User Group, T_EX 用户组) 发布的一个发行版。T_EX Live 可以在类 Unix/Linux、macOS 和 Windows 系统等不同操作系统下安装使用,并且提供相当可靠的工作环境。T_EX Live 可以安装到硬盘上运行,也可以经过便携(portable)安装刻录在光盘上直接运行(故有“Live”之称)。

不同操作系统下安装设置 T_EX Live 的方式基本一样,这里以 Windows 操作系统为例进行演示。^[4]

T_EX Live 一般以安装镜像的方式在互联网上发布。光盘镜像文件可以从官网上下载^[5]。载入镜像后,执行 `install-tl-windows.bat` 进行安装。只要选好安装的位置,不断单击“下一步”就可以安装 T_EX Live 了,如图所示。

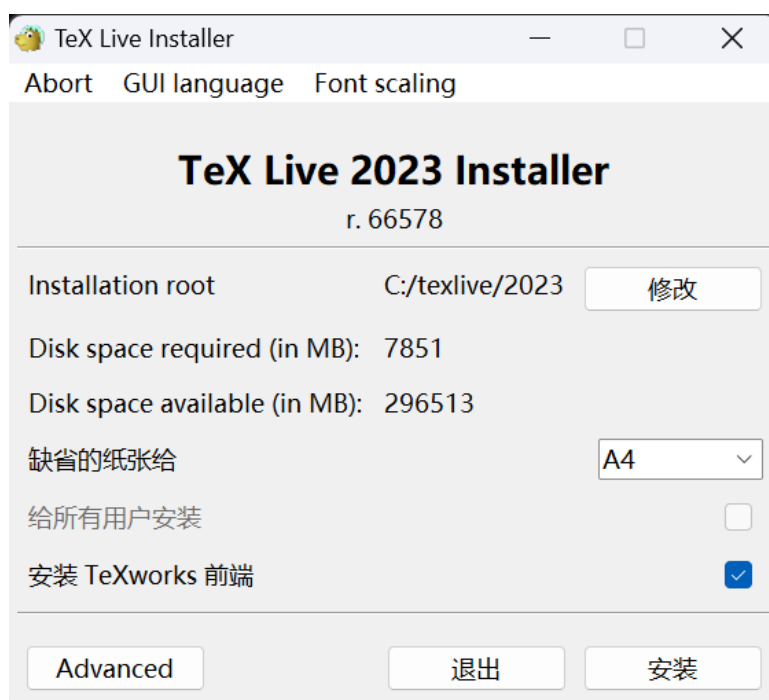


图 1.1: 在 Windows 11 上安装 T_EX Live2023

程序安装好了之后,会在开始菜单栏增加 T_EX Live 文件夹图标。其中包含的内容比较简单,它包含以下项目:

- **TeXworks editor**: 这是 T_EX Live 预装的一个 T_EX 文件编辑器,简单方便。大部分工作都可以在这个编辑器中完成。
- **DVIOUT DVI viewer**: 这是一个 DVI 文件预览器。我们很少用到它。

^[3]并不是。由于 C_T_EX 早已停止维护及历史原因,现在 T_EX Live 是更好的选择。于是我选择跳过 1.1.1.1

^[4]原书中还包含部分 Linux 系统上的操作讲解,这里删去。

^[5]<https://mirrors.tuna.tsinghua.edu.cn/CTAN/systems/texlive/Images/>

- **TeX Live command-line** : 它打开 Windows 的命令提示符, 并设置好必要的环境变量, 可以在其中使用命令行编译处理 T_EX 文档。
- **TeX Live documentation** : 这是一个 HTML 页面的链接, 里面是 T_EX Live 系统中所有 PDF 或 HTML 格式的文档列表。在首页你可以找到几种语言 (包括简体中文) 的 T_EX Live 发行版文档, 以及到近 4000 份各种文档的列表的链接——这份有一公里长的列表多少说明了 T_EX Live 是一个多么复杂的系统, 以及它在安装时为什么占用了这么大的空间。当然, 你不需要读完里面的所有文档才能学会 L^AT_EX, 不过你会发现工作中总需要时不时地查看里面的东西。
- **TLShell TeX Live Manager** : 这是 T_EX Live 管理工具的图形界面, 简称 tlmgr。管理工具也可以在命令行下用 tlmgr 命令运行, 用 tlmgr gui 可以在命令行下打开图形界面。

1.1.2 编辑器与周边工具

1.1.2.1 编辑器举例——TeXworks

像其他计算机语言一样, L^AT_EX 使用纯文本描述、因而任何能编辑纯文本的编辑器都能编辑 L^AT_EX 文档, 如 Windows 系统的记事本、写字板, Linux 下的 VI、GEdit。不过, 使用专门为 L^AT_EX 设计或配置的编辑器, 进行语法高亮、命令补全、信息提示、文档排版等工作, 会使工作方便很多。

L^AT_EX 代码编辑器有很多, 大致可以分为两类: 一种主要为 L^AT_EX/T_EX 代码编辑而专门设计的编辑器, 二是可以为 L^AT_EX/T_EX 代码编辑配置或安装插件的通用代码编辑器。前者如 WinEdt、TeXworks、TexMaker、Kile, 后者如 Emacs、VIM、Eclipse、SciTE 等。通常前一种编辑器配置和使用更简单一些, 下面主要以 TeXworks 为例说明编辑器的一些简单配置。其他大部分编辑器在基本功能和设置上都大同小异, 不难举一反三。

TeXworks 的界面非常简洁 (见图 1.2): 它分为两个部分, 左侧是 T_EX 源文件的编辑窗口, 右侧是生成的 PDF 文件的预览窗口。左边的编辑器窗口最上面是标题栏和标准菜单栏, 接着是工具栏, 中间最大的编辑区, 最下面是显示行列号的状态栏。右边的预览窗口把编辑区换成了 PDF 预览区。

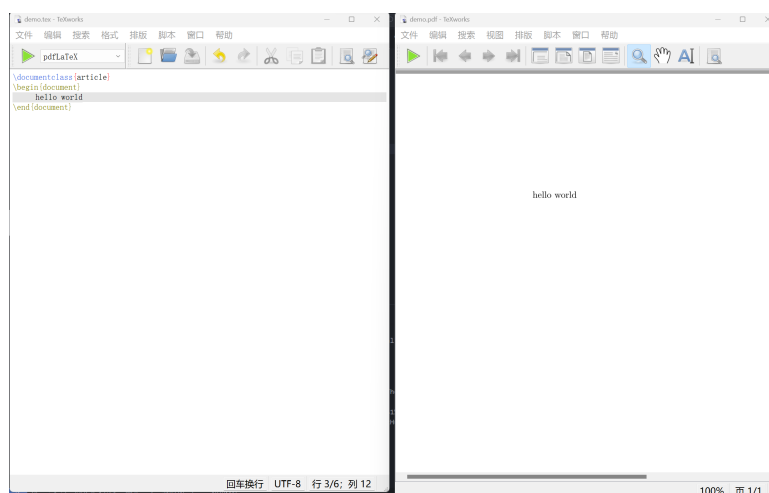


图 1.2: TeXworks 界面

除了文本编辑区, 编辑器窗口中最常用的是工具栏。工具栏的最左边的按钮是整个编辑器最为重要的“排版”按钮, 它调用具体的命令把输入的 T_EX 源文件编译为对应的 PDF 结果, 刷新右边 PDF 文件的显示。紧靠排版按钮右边的下拉菜单中用来选择排版时所使用的命令, 通常对应一条单一的命令, 但也可以配置为好几条命令的复合。通常我们使用最多的排版命令是“XeLaTeX”或“PDFLaTeX”, 视具体情况而定。使用排版按钮时, 未保存的文档会自动保存。工具栏剩下的按钮则是一系列常见的标准按钮: 新建、打开、保存; 撤销、重做; 剪切、复制、粘贴; 查找和替换, 不必多说。

PDF 预览窗口的工具栏也是一排按钮。最前面的排版按钮与编辑区的功能一样。右面是 4 个向前后翻页的按钮; 而后是显示比例的按钮; 再后面是放大工具、滚屏工具; 最后是 PDF 文本查找工具。

使用 TeXworks 也很简单：

1. 在编辑区输入 T_EX 源文件
2. 单击“保存”按钮，给源文件起名并保存在指定位置
3. 在排版按钮旁的下拉菜单中选择“XeLaTeX”，单击排版按钮，查看结果。

编译时在文本编辑区下方的“控制台输出”面板中会显示编译进度和信息。如果编译过程有错误或提示输入、程序会停下来等待处理。如果编译结束无误，控制台输出面板会自动关闭，而在预览窗口会显示新的 PDF 结果。

在文本编辑区或 PDF 预览区用鼠标左键单击可以从源文件跳转到 PDF 文件中的对应位置；或反过来从 PDF 跳转到源文件中的对应位置。这个功能称为 T_EX 文档的**反向查找**，对编写长文档特别有用。正反向查找是由 SyncTeX 机制实现的，需要源代码编辑器、PDF 阅读器和 T_EX 输出程序的共同参与，一些旧的发行版或程序可能并不支持。

TeXworks 支持**自动补全**功能。输入一个助记词或命令的一部分，再按 Tab 键，则 TeXworks 会根据配置补全整个命令或是环境；连续按 Tab 键可以切换补全的不同形式。例如，输入 \doc 再按 Tab 键，会补全命令 \documentclass{}；使用 beq 补全则可以得到公式环境：

```
\begin{equation}
|
\end{equation} *
```

光标在环境中央等待输入，再次按下 Ctrl + Tab 组合键可以跳转到后面的圆点处继续下面内容的输入，而不需要使用方向键。

下面来看看 TeXworks 中一些常见的配置。

刚刚安装的 TeXworks 通常会使用很小的字体，而且可能没有语法高亮等功能，给编辑工作带来诸多不便。在 TeXworks 的“格式”菜单中，“字体”项可以用来临时更改显示的字体，而“语法高亮显示”项可以临时控制如何进行语法高亮。要使字体和语法高亮的设置对所有文档生效，则应该修改 TeXworks 的默认选项。单击 TeXworks “编辑”菜单的最后一项“首选项”，将弹出 TeXworks 首选项窗口（见图1.3）。在“编辑器”选项卡中，可以设置编辑器默认的字体和字号，下面则有语法高亮、自动缩进等格式。

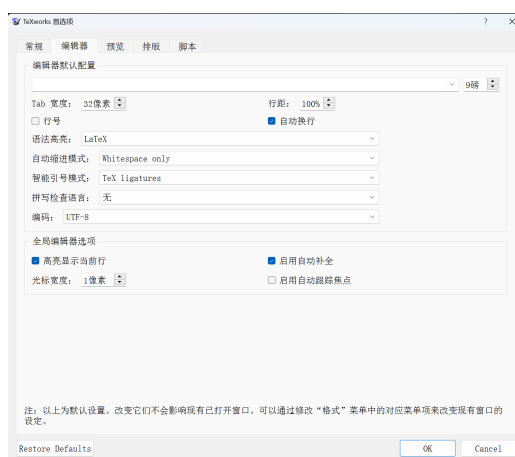


图 1.3: TeXworks 编辑器首选项设置

TeXworks 支持多种语言界面和多种文字编码。TeXworks 默认的界面会与操作系统的默认语言（Locale 设置）一致，可以在首选项设置窗口的“一般”（General）选项卡中设置程序的界面语言为中文。在“编辑器”选项卡中则有“编码”选项，一般应选择 TeXworks 的默认值，即 UTF-8 编码，编辑器保存和打开文件将默认使用此编码。

TeXworks 首选项窗口的“排版”选项卡可以用来设置 TeXworks 的“排版”按钮所执行的命令。选择对应的处理工具，单击“编辑...”按钮，就可以在弹出的窗口中设置对应的命令及参数。参数中使用的变量，可参见

TeXworks 的帮助文档。

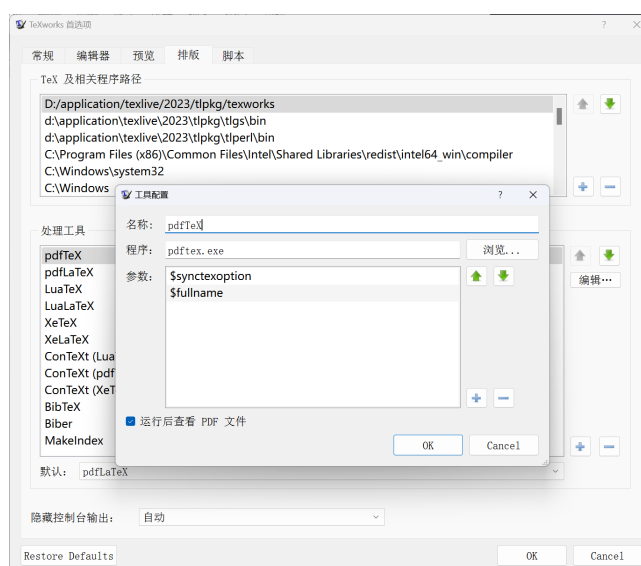


图 1.4: TeXworks 排版首选项设置

文本编码与 Unicode

在使用 T_EX 编辑器时，必须要注意文档保存的文字编码。如果编码使用错误，轻则遇到“乱码”，重则导致程序运行错误。我们提到的“UTF-8”编码，就是现在最为常用的编码之一。

文字在计算机内部都是以数字的形式表示、储存和运输的，人们圈定一些在计算机中使用的字符，称为字符集（character set），一个字符通常就用它在字符集中的序号来表示。不过由于在计算机中数字的二进制表示也有不同的格式，因而相同的字符集也可能有不同的二进制表示方式，也就是字符编码（character encoding）。IBM 公司以前给自己系统中每一种编码编一个号，即所谓代码页（code page），后来其他计算机厂商如微软、Oracle 都把自己的字符编码用代码页的方式给出，不过使用的代码页编号都不一样。我们通常见到的代码页，都是微软公司的编号。字符集、字符编码、代码页这些概念，在很多时候都不加区分，可以混用。

ISO 于 1990 年推出了通用字符集（Universal Character Set, UCS）标准 ISO 10646，包括 UCS-2 和 UCS-4 两种长度的编码；1991 年一个叫做通用码协会（Unicode Consortium）的组织发布了 Unicode 1.0 标准。两个组织都打算把全世界所有文字的符号都用一套字符集和编码统一起来。后来，两个组织协作起来，从 Unicode 2.0 起，Unicode 就符合 ISO 10646 了。2012 年 1 月发布的 Unicode 6.1 已经定义了 110181 个字符，包括世界上 100 种文字，而且还在不断修订扩充之中。^[6]Unicode 已逐渐成为字符编码的新方向，包括 GB 18030 也可以看成是 Unicode 字符集的一种编码格式。除了 UCS-2 和 UCS-4，Unicode 标准还提出了多种编码形式，称为 Unicode 交换格式（Unicode Transformation Format, UTF）：主要包括变长的 UTF-8、UTF-16 和定长的 UTF-32 编码。UTF-8 编码与 ASCII 编码向下兼容，因而最为常用。

T_EX 系统原本只支持 ASCII 编码。但只要设置好超过 127 的数字对应的符号，所有拓展 ASCII 编码都能正确排版，如 ISO 8859 的各种标准。汉字编码 GB 2312、GBK 和 UTF-8 都是兼容 ASCII 的多字节编码，因而在 L^AT_EX 中通过 CJK 宏包也可以通过特殊的方式，把多个字符对应到一个汉字上，支持中文的排版。

CJK 宏包这种支持多字节编码的方法其实是一种黑客手段，后来 T_EX 的新实现 XeTeX 和 LuaTeX 都直接支持 UTF-8 编码，新的中文排版方式也自 2007 年起随着这两种新排版引擎应运而生。LuaLaTeX 的本地化支持目前还暂处于起步阶段，本书将着重介绍基于 XeTeX 的方式。

^[6]目前最新版本的 Unicode 13.0 中，包含的字符总数已达 143859 个。

1.1.2.2 PDF 阅读器

T_EX Live 已经预装了 DVI 文件和 PostScript 文件的阅读器。然而却没有安装最重要的 PDF 格式阅读器。现在使用的 T_EX 系统基本上最终都输出 PDF 格式的结果，而且 T_EX 系统中的大部分文档也都是 PDF 格式的，因此一个 PDF 阅读器是不可或缺的。

PostScript 阅读器 GSview 和 PS_View 也都可以当做 PDF 阅读器来使用，不过效果不是很好。我们使用 PDF 阅读器有两个目的，一是在编辑文档过程中随时查看编译的效果，这对于编辑复杂公式、插图以及幻灯片来说都非常重要；二是为了阅读 PDF 格式的文档资料，或查看自己编写文档的最后效果。这两个目的各有不同的要求，前者要求快捷方便，最好还能在 PDF 的效果和 T_EX 源文件之间方便地切换检索；后者则要求显示准确美观、功能全面。

要满足第一个要求，编辑器 TeXworks 内置的显示功能最为方便。TeXworks 把源代码和 PDF 结果左右排开，对照显示，T_EX 代码编译后右边的 PDF 文件就会更新。而且可以用 Ctrl+鼠标左键单击进行从源文件到 PDF 文件或 PDF 文件到源文件的正反向索引。

要满足第二个要求，我们建议使用 Adobe Reader 最新的版本^[7]（Linux 系统中通常称为 arcoread）。Adobe Reader 是官方免费提供的 PDF 格式阅读器，通常它的显示结果最好，支持全面的 PDF 特性（如 JavaScript 脚本、动画、3D 对象等，而且这些很可能在 L^AT_EX 制作的幻灯片中用到。）其他一些常见的 PDF 阅读器，如 Foxit Reader，则可能在一些功能上有欠缺。

如果你还没有安装 Adobe Reader，可以直接从 Adobe 的官方，或几乎任何的下载站点中得到并安装。不必抱怨它占用上百兆的安装空间：除了一些高级功能的插件，Adobe Reader 还提供了许多种高质量的 OpenType 西文字体，这些都可以在你未来的排版中用到。

PS 格式和 PDF 格式

PS 是 PostScript 的简称。PostScript 是 Adobe 公司于 1984 年发布的一种页面描述语言，自 1985 年苹果公司的 LaserWriter 打印机开始，此后的很多高档激光打印机都带有 PostScript。于是，PostScript 逐渐成为电子与桌面出版的标准格式，并一直延伸到整个出版业，风靡一时。就连国内最大的北大方正公司的排版系统也是以变形的 PostScript 格式输出，并沿用至今。

“PostScript”这个名字多少体现了这门语言的特点：它是一种基于后缀表达式和栈操作的解释型计算机语言。例如，表达式 $1+2$ 就被写成 `1 2 add`。而 `0 0 moveto ; 100 100 lineto` 则是在描述从坐标 (0, 0) 到坐标 (100, 100) 的直线路径。使用这种后缀语法原本是为了方便计算机芯片高效解释 PostScript 这种复杂的语言，大部分 PostScript 代码也都是由其他计算机程序自动生成的。不过，富有经验的老手可以就凭借着这种看起来有些怪异的语法直接画出图来，这种技艺也一直延伸到将要讲到的 PSTricks 宏包中。

PostScript 拥有强大的图形能力，可以用一段 PostScript 语言的代码表示很复杂的图形。然而，作为一门完整的计算机语言，PostScript 过于复杂，因而出现了所谓封装的 PostScript（Encapsulated PostScript）格式，即 EPS 格式。EPS 格式的文件也是一段 PostScript 代码，但只能表示一页，而且加上了诸多限制，成为一种专门用来存储可以嵌入其他应用的图形格式。T_EX 的许多输出引擎都支持这种图形格式。

由于在电子出版领域的地位，PostScript 一度成为 T_EX 最重要的输出格式，至今也能在网络上见到大量 T_EX 系统产生的 PostScript 格式的书籍和文章，一些期刊也一直要求以能生成 PostScript 格式的 T_EX 文档投稿。然而随着新一代廉价的喷墨打印机的出现，需要复杂解释芯片的 PostScript 打印机逐渐式微；而网络技术的发展进一步催生了电子文档交换的需求，PDF——Portable Document Format（可移植文档格式）便应运而生。PDF 由 Adobe 公司于 1993 年发布，它是 Adobe Acrobat 系列产品的原生文件格式，并随着文件格式的公开和阅读器 Adobe Reader 免费的发放，迅速风靡起来。

PDF 和 PostScript 使用相同的 Adobe 图形模型，可以得到与 PostScript 相同的输出效果，而在程序语言

^[7]全名为 Adobe Acrobat Reader

方面则比 PostScript 大为削减，并增强文档格式结构化，可以迅速地由计算机处理。尽管 PDF 最初只是 PostScript 削减功能适应电子文档处理的结果，但 PDF 转而在电子文档交互式表单、多媒体嵌入等方面大下功夫，并不断进行各方面的扩充，最终成为一种比 PostScript 还复杂的格式。PDF 也继 PostScript 之后成为现在新一代的电子出版业的事实标准。

现代的 T_EX 输出引擎几乎都以 PDF 为输出格式。同时 PDF 格式也可以像 EPS 格式一样作为图形格式被 T_EX 和其他软件使用。现在能够输出 PDF 图形的软件和支持嵌入 PDF 图形的 T_EX 引擎比 EPS 格式的还要多些，PDF 也成为现在 T_EX 系统中最重要的图形格式。