

W32TEX

Web2C-7.5.7 Kpathsearch-3.5.7

2009 年 10 月 5 日



日本語化は株式会社アスキーによるものです。また、NTT JTEX は千葉大学の桜井さんによるものです。もしおかしな振る舞いをする場合、株式会社アスキーや、桜井さんによるオリジナルのものとは無関係で、Win32 への移植によるものかもしれませんので、株式会社アスキー、あるいは桜井さんに直接問いあわせることは避けてください。この文書の最後のアドレスに email でご連絡をお願いいたします。

目次

1	はじめに	2
1.1	フォーマットファイルの作成 (fmtutil)	2
1.2	perl スクリプトなどの実行 (runscr)	3
1.3	Kpathsearch 機構についての注意	3
1.4	多重 TEXMF trees	3
1.5	MetaPost における troff モード	4
1.6	Source specials について	4
1.7	Windows 95/98 での使用	5
1.8	プロセス間通信	5
2	インストール法	7
3	アンインストール法	10
4	個々のファイルの詳細	10
5	日本語サポート Ghostscript	34
6	その他のファイル	35
7	ファイル sh.exe について	37

1 始めに

株式会社アスキーで開発された、pTeX 3.1.10、および千葉大学の桜井さんによって開発された NTT JTeX 2.1 を Win32 (x86) console applications として Microsoft VC++ で make したバイナリです。ベースとする TeX のバージョンは 3.1415926 としています。デフォルト漢字コードはシフト JIS としております。ただし、upTeX の仲間ではデフォルトエンコーディングは utf8 です。いずれの場合も pTeX の仲間に関しては土村さんによる pTeXENC ライブラリを使用していますので、エンコーディングは sjis, jis, euc, utf8 のうち、任意のものを使用することができます。日本語化されていない TeX および関連する各種応用プログラムも含まれています。

1.1 フォーマットファイルの作成 (fmtutil)

新たにフォーマットファイルを作成する場合には、コマンド `fmtutil` をサポートしていますので、MS-DOS プロンプト (Windows 95/98)、あるいはコマンドプロンプト (Windows NT/2000/XP/Vista) で

```
fmtutil --missing
```

とか

```
fmtutil --all
```

などとするだけでよくなっています。fmtutil.exe は

```
fmtutil --help
```

とすると、簡単な説明を表示します。fmtutil が使用するデータベースは `texmf/web2c/fmtutil.cnf` です。これには既に多くのフォーマットファイル用のデータが記述してあります。fmtutil によって、メタフォント用の `mf.base`、メタポスト用の `mpost.mem`、`mfplain.mem`、`metafun.mem` を作成することもできます。また、日本語メタポスト (松山道夫さんにより最初手がけられ、現在鈴木秀幸さんによってメンテナンスされています) も同梱してあり、`jmpost.mem` も `fmtutil` から作成できるようにしてあります。fmtutil.cnf の書き方については、デフォルトでもあるサンプル `texmf/web2c/fmtutil.cnf` をご覧ください。もし `platex` などの実行時に、`platex.fmt` などが無かったら、プログラム `mktexfmt` が自動的に呼ばれ、これから更に `fmtutil` が呼ばれることによってフォーマットファイルが自動作成され、処理がとまらずに続行されます。

1.2 perl スクリプトなどの実行 (runscr)

runscr.exe なる実行ファイルは perl スクリプトなどを走らせるためのものです。例えば foo.pl なる名前の perl スクリプトがあるとします。これを runscr.exe と同じディレクトリに置きます。あるいは texmf/scripts 以下においてもよいです。次に

```
ln runscr.exe foo.exe      (NT/2000/XP with NTFS)
```

```
fsutil hardlink create foo.exe runscr.exe (XP with NTFS)
```

あるいは

```
copy runscr.exe foo.exe   (FAT)
```

として foo.exe なるコマンドを作ります。そうすると、foo とするだけで目的の perl スクリプト foo.pl を実行できます。Windows XP with NTFS で標準サポートされている、fsutil コマンドでは、リソースキットの ln コマンドと比べて、NewName と ExistingName の順番が逆なので、注意して下さい。texmf/doc/misc/runscr にあるソースを見たらわかるように、foo.exe ... を実行すると、同じディレクトリ、あるいは texmf/scripts 以下に foo があれば perl foo ..., foo.pl があれば perl foo.pl ... foo.rb があれば ruby foo.pl ... などを実行しているだけです。

1.3 Kpathsearch 機構についての注意

オリジナルの Kpathsearch では、カレントワーキングディレクトリのパス名に文字 '{', '}', ',', '}' が含まれていると、PK フォント作成の際にエラーになることがありますが、W32TeX ではエラーにならないように変更してあります。

1.4 多重 TEXMF trees

デフォルトでは texmf.cnf に

```
TEXMF = {$TEXMFLOCAL,$TEXMFMAIN}
```

としており、TEXMF trees は二個にしています。ユーザ作成のパッケージなどを入れておくのは \$TEXMFLOCAL です。ただし、構造は \$TEXMFMAIN と同じにしなければいけません。配布パッケージでは、常に \$TEXMFLOCAL は空ディレクトリなので、ユーザ側がインストールしているものを上書きすることはありません。具体的には

```
TEXMFMAIN = InstallDir/share/texmf
TEXMFLOCAL = InstallDir/share/texmf-local
```

としています。デフォルトでそうしているように、多重 TEXMF trees を使う場合には TEXMF の記述において、braces ({ と }) が必要であることに注意して下さい。下さい。また TEXMFDBS は mktexlsr が参照する変数であり、データベース ls-R を作成したい場所を記述しておきます (最大 16 個まで)。デフォルトでは次のように設定してあり、通常変更する必要はありません:

```
TEXMFDBS = $TEXMF
```

1.5 MetaPost における troff モード

MetaPost における troff モードは、groff によって実現してあります。従って groff がインストールしてあれば troff モードを使うことができます。コマンド gsoelim.exe, geqn.exe と gtroff.exe が必要です。また次の作業をしておかなければいけません:

1. texmf.cnf の TRFONTS なる変数に groff の PS フォントのディレクトリ名を設定して下さい。

例: TRFONTS=c:/w32tex/share/groff/font/devps//

1.6 Source specials について

tex, etex, omega, aleph, ptex, jtex をエンジンとするものについては source specials をサポートしています。これは dviout for Windows Ver. 3.11 以降と共に用いるとき便利なものです。pdftex をエンジンとするものについては、DVI 出力時 (\pdfoutput=0) のとき source specials をサポートしています。デフォルトの PDF (Adobe Portable Document Format) 出力時には source specials は意味を持たないので、Warning を出力して無視します。source specials を埋め込むには

```
platex --src=auto,par,math,vbox foo.tex
```

などとします。dviout for Windows で preview 時に、指定した行近辺にエディタのカーソルを移動させることができます。方法については dviout for Windows の help を参照して下さい。なお、dvips(k) は source specials を無視するようになっていまして、source specials があってもパフォーマンスが落ちることはありません。source specials を埋め込むと、標準の dvi では無くなるため、場合によっては、埋め込まない場合と出力が違うことがあります。最終出力では source specials を埋め込まない dvi を作って下さい。

1.7 Windows 95/98 での使用

Windows series で大抵使用可能なはずですが、稀に Windows 95/98 でエラーになることがあるかも知れません。Windows NT/2000/XP では特に不具合の報告はありません。今までに報告のあった、Windows 95/98 で使用する場合の不具合について述べておきます。

- tex testfont が Win95 で処理不能である。
これについては、tex の代わりに ptex を使うと処理できるようです。うまく行かない場合には、plain TeX の色々なバージョンを試して下さい。全ての可能性をあげると
 1. tex testfont
 2. ptex testfont
 3. ptexold testfont
 4. jtex testfont
 5. etex testfont
 6. omega testfont
 7. aleph testfont
 8. pdftex -output-format=dvi testfont

となります。

その他の不具合報告はありません。

1.8 プロセス間通信

TeX における IPC (InterProcess Communication) は、NeXT 上の TeXView 専用に T. Rokicki によって実装されたもので、Shamim Mohamed が Web2C に適用したものです。コントロール情報と共に、DVI 自身もファイルに書き出されると同時に、TeXView にも送られ、イン

クリメンタルに表示できるようになっているようです。NeXT 専用なので、ほとんど IPC が使えるように Make されることはありません。ただ、DVI ファイル書き出しがバッファリングされなくなるので、それを利用するアプリケーションのために IPC enabled として Make されることがあります。サーバ (TeXView) が無ければ単に DVI ファイルがバッファリングされずに書き出されるという効果だけになります。通常使用では、バッファリングなしで書き出される場合、(体感できるほどではないでしょうが) 若干パフォーマンスが悪くなると思われます。

W32TeX も IPC enabled として Make しています。従ってオプション `-ipc` と `-ipc-start` を使用することができます。サーバの名前は `texview.exe` に固定されていて変更はできません。 `texview.exe` が無ければ、単に DVI をバッファリング無しで書き出すようになるだけです。 `texview.exe` が存在して通信できる場合も、コントロール情報を送るだけで、DVI 自身は送りません。通信は TCP/IP でアドレスはローカルホスト (127.0.0.1) に固定しています。使用ポートも 4242 に固定しています。送られるコントロール情報は

```
struct msg {
    int    namelength;
    int    eof;
    char   fname[256];
}
```

の形式をしており、`eof` は既に書き出された DVI ファイルのバイト数に近い数です。 `namelength` がゼロでない場合には、`fname` に DVI ファイルのフルパス名が入っており、その文字列の長さが `namelength` です。ただし C 文字列ではないので、文字列の直後が `'\0'` になっている保証はないので注意して下さい。 `texmf/doc/ipc` に、あまり役に立たないサンプル `texview.c` と、それからつくった実行ファイル `texview.exe` があります。 `texview.exe` は手動で `bin` ディレクトリにコピーしない限りインストールはされません。 `dviout` は DVI ファイルが最後まで書かれていない段階で、途中まで読ませてもだめなので、サンプルの `texview.exe` は コントロール情報のファイル名だけを使用して、DVI ファイルがすべて書き出された頃にも `dviout` が立ち上がっていなかったら、自動的に `dviout` を立ち上げて完成した DVI を開きます。既に `dviout` が立ち上がっている場合には、その `dviout` を使用して DVI を開きます。あまり参考にはなりません。詳細は `texview.c` を参照して下さい。 W32TeX における IPC の特徴をまとめると次のようになります：

- オプション `-ipc` の場合

既に `texview.exe` が実行されていた場合、`connection` に成功したら DVI ファイルを書きながら、コントロール情報を `texview.exe` に送ります。 `connection` に失敗したり、 `texview.exe` が実行されていない場合には、DVI ファイル書き出しがバッファリングされないだけで、通常と同じです。

- オプション `-ipc-start` の場合

`texview.exe` を立ち上げて通信しようと試みます。成功したら DVI ファイルを書きながら、コントロール情報を `texview.exe` に送ります。`texview.exe` の立ち上げや通信に失敗した場合は、DVI ファイルの書き出しがバッファリングされないだけで、通常と同じです。

サンプルとして入れている `texview.exe` を使う場合、次のような使い方が考えられます:

```
platex test
platex test
platex -ipc-start test
```

この場合、2回は通常のように `platex` を実行し、3回目はコンパイルが完成することを予想して、終わったら `dviout` でプレビューしよう、という意図を持っていることとなります。ただし `dviout.exe` と `texview.exe` が `PATH` から見つけられないと、`-ipc-start` オプションの効果は無くなります。

2 インストール法

念のために記しますが、以下に述べる作業は、Windows 95/98 ならば MS-DOS プロンプト、Windows NT/2000/XP/Vista ならば コマンドプロンプトにて行います。

- (1) $\text{T}_{\text{E}}\text{X}$ をインストールするドライブと、トップディレクトリ (フォルダ) の名前を決めて、そのディレクトリが存在しない場合には作成して下さい。たとえば `c:/w32tex` など。このトップディレクトリを今後**インストールディレクトリ**と呼びます。以下ではインストールディレクトリを `c:/w32tex` として説明しますので、実際と違う場合には読みかえて下さい。トップディレクトリの名前は何でも良いですが、`c:/Program Files/tex` のように、空白を含む名前は避けて下さい。一見正常に動作しているように見えても、何かの拍子におかしなことになる可能性があります。 $\text{T}_{\text{E}}\text{X}$ のソースファイル名に空白を含めることができるように機能拡張されていますが、関連するすべてのプログラムで空白を含むファイル名を使えることが保証されているわけではありません。この機能はできるだけ使用しないことをお勧めします。

```
c:
md \w32tex
```

- (2) パッケージをダウンロードするディレクトリ (フォルダ) の名前を決めて下さい。ここでは `c:/temp` として話を進めます。従って実際のディレクトリが違う場合は読みかえて下さい。

- (3) 最初に `texinst757.zip` という名前のファイルをダウンロードし、それをインストールディレクトリに展開して下さい:

```
c:
cd \w32tex
unzip c:/temp/texinst757.zip
```

- (4) 次に、以下に掲げる必要ファイルをダウンロードし、カレントディレクトリをインストールディレクトリとして、以下のようにするとパッケージが展開されます:

```
c:
cd \w32tex
texinst757 c:/temp
```

`texinst757` コマンドの引数は、ダウンロードしたアーカイブファイルが存在するディレクトリのフルパス名です。展開が終わると最後に `PATH` に追加すべきものが表示されますから、Windows 95/98 の場合にはファイル `AUTOEXEC.BAT` を編集することにより、また Windows NT/2000/XP の場合にはコントロールパネル等で定義してください。必要なのはバイナリディレクトリ (たとえば `c:\w32tex\bin`) を `PATH` に追加することだけです。`TEXMFMAIN` は、インストールしたディレクトリから自動決定されるので、定義する必要はありません。もし、他のプログラムとの連携などの必要性から、`TEXMFMAIN` を環境変数で定義した場合には、環境変数の方が優先されます。`TEXMFCNF` は `Kpathsearch` 機構の内部で決定されるので、特別の理由がある場合を除いて、環境変数では定義しないで下さい。エディタについての

```
TEXEDIT=c:/w32tex/bin/mule +%d %s
MFEDIT=c:/w32tex/bin/mule +%d %s
MPEDIT=c:/w32tex/bin/mule +%d %s
```

のようなものは、各自で好きなエディタの約束に従って定義して下さい。これらは `texmf.cnf` で定義しておいても、`TEX` 関係のプログラムから見えます。他のプログラムで必要なら、環境変数で定義して下さい。`%d` には行番号が、`%s` にはファイル名が入ります。実体が `mule` for Win32 であって、名前が `mule.exe` あるいは `mule.bat` なるエディタコマンドが `PATH` 上にあれば `TEXEDIT`、`MFEDIT` および `MPEDIT` を定義する必要はありません。なお、Windows 95/98 で `AUTOEXEC.BAT` に記述するときは、以下のようにパーセント記号 `%` を二重にする必要があります:

```
set TEXEDIT=c:/w32tex/bin/mule +%d %s
set MFEDIT=c:/w32tex/bin/mule +%d %s
set MPEDIT=c:/w32tex/bin/mule +%d %s
```

環境変数では、最後のスペースやタブも無視されないので、余分なスペースやタブを行末に入れないように注意して下さい。行末に余分なスペースやタブがついていると、わかりにくいエラーが生じる原因になります。

(5) 必要ファイルの例を 3 例以下に掲げます。組み合わせはこれに限りません。後に述べる各ファイルの説明を参考にして適当なシステムを構築して下さい。

- **最小インストール。**

1. [latex.tar.bz2](#)
2. [mftools.tar.bz2](#)
3. [platex.tar.bz2](#)
4. [ptex-3.1.10-w32.tar.bz2](#)
5. [web2c-2009-lib.tar.bz2](#)
6. [web2c-2009-w32.tar.bz2](#)

- **標準インストール。最小インストールに以下を付け加えます。**

1. [dviPDFm-w32.tar.bz2](#)
2. [dvipsk-w32.tar.bz2](#)
3. [jtex-2.1-w32.tar.bz2](#)
4. [ltxpkgs.tar.bz2](#)
5. [luatex-w32.tar.bz2](#)
6. [makeindex-w32.tar.bz2](#)
7. [manual.tar.bz2](#)
8. [oldformat.tar.bz2](#)
9. [oldinputs.tar.bz2](#)
10. [pdftex-w32.tar.bz2](#)
11. [t1fonts.tar.bz2](#)
12. [timesnew.tar.bz2](#)
13. [txpx-pazofonts.tar.bz2](#)
14. [vf-a2bk.tar.bz2](#)

- **フルインストール。更に以下を付け加えます。**

1. [context-doc.tar.bz2](#)
2. [context.tar.bz2](#)
3. [cweb-w32.tar.bz2](#)
4. [dvi2ps-4.1j-w32.tar.bz2](#)
5. [dvitools-w32.tar.bz2](#)
6. [lcdf-typetools-w32.tar.bz2](#)
7. [omega-aleph-w32.tar.bz2](#)
8. [omegaj-w32.tar.bz2](#)
9. [plain2-2.54-w32.tar.bz2](#)
10. [psutils-w32.tar.bz2](#)
11. [sam2p-w32.tar.bz2](#)
12. [t1utils-w32.tar.bz2](#)
13. [tex4htk-w32.tar.bz2](#)
14. [texinfotools-w32.tar.bz2](#)
15. [tiff2png-w32.tar.bz2](#)
16. [ttf2pk-w32.tar.bz2](#)
17. [ttf2pt1-w32.tar.bz2](#)
18. [txtutil.tar.bz2](#)
19. [ums.tar.bz2](#)
20. [uptex-w32.tar.bz2](#)
21. [utf.tar.bz2](#)
22. [vf-n2bk.tar.bz2](#)

3 アンインストール法

レジストリは全然さわっておりませんので、コピーしたファイル、新たに作成したディレクトリを全て削除し、環境変数をもとに戻せば終わりです。

4 個々のファイルの詳細

(01) [latex.tar.bz2](#)

L^AT_EX 2_ε <2005/12/01> patch level 0 のマクロおよびフォーマットファイルです。

(02) [mftools.tar.bz2](#)

mktxmf.exe, mktextfm.exe, mktexpk.exe および ps2pk.exe です。詳細な設定をする

には `texmf/doc/mftools/INSTALL` を読んで、それに従って下さい。これは $\text{T}_{\text{E}}\text{X}$ や、種々の DVI ドライバから自動呼出しされますので、必ずインストールしておいて下さい。dviout for Windows でフォント生成に `mktexpk` を使用するように設定しておく、`psnfss` 等を使って作成した dvi を preview する場合に要求される PK フォントは、殆どの場合 `ps2pk` で作成されます。必要な Type1 フォントは `t1fonts.tar.bz2` の中に入れてあります。Adobe 純正の base14 Type1 フォントから作成したい場合には Acrobat Reader 3.0 (注。Acrobat Reader 4.0 以降ではない) 付属の Type1 フォント (.PFB) を、`texmf/fonts/type1/adobe/base` なるディレクトリを作成してそこに置いておいて下さい。そうして `pspk supp-acrofont-base.map` を `pspk supp-base.map` に上書きコピーしておいて下さい。これらのファイルは `texmf/fonts/map/ps2pk/updmap` にあります。デフォルトの `pspk supp-base.map` は `pspk supp-urw-base.map` と同じものになっています。この変更を有効にするには `updmap` コマンドを実行します。Acrobat Reader 3.0 から Acrobat Reader 4.0 以降にバージョンアップするときは、3.0 を消去する前に Type1 フォントを上記述べたディレクトリにコピーしておくといいでしょう。`txpx-pazofonts.tar.gz` をインストールしておけば、`ps2pk` 経由で、dviout でも Type1 フォントパッケージ TX fonts および PX fonts を使用できるようにしています。

(03) [platex.tar.bz2](#)

株式会社アスキーによる $\text{p}_{\text{L}}\text{A}_{\text{T}}\text{E}_{\text{X}}2_{\epsilon}$ のマクロおよびフォーマットファイルです。

(04) [ptex-3.1.10-w32.tar.bz2](#)

$\text{T}_{\text{E}}\text{X}-3.141592$ をベースにして、株式会社アスキーによって日本語化されたバイナリです。必ず入手する必要があります。この $\text{p}_{\text{T}}\text{E}_{\text{X}}-3.1.10$ では `source specials` をサポートしています。フォントなどの必須サポートファイルも含まれます。日本語仮想フォントを作成する tool `makejvf.exe` も含んでいます。Win32 用の `makejvf.exe` は `kpathsearch 5.0.0` に対応させていますので、`jfm` ファイルは `TEXMF tree` から見つけることができます。従ってカレントディレクトリに `jfm` を置いておく必要はありません。 $\text{p}_{\text{T}}\text{E}_{\text{X}}$ ではファイル名として日本語を使うこともできますが、それは `--jobname` オプションでベースファイル名を指定した場合に限ります。例えば

```
platex --jobname=表題 表題
```

とすると、正しく `表題.dvi` が出力されますが、単に

```
platex 表題
```

とした場合には、ファイル名が崩れてしまいます。dviware 等は、日本語ファイル名をサポートしていないものもありますから、可能な限りファイル名には日本語を使用せず、アスキー文字に限ることをお勧めします。

(05) [web2c-2009-lib.tar.bz2](#)

TeX 用フォントなどのサポートファイル群です。必須です。

(06) [web2c-2009-w32.tar.bz2](#)

オリジナルの TeX および関連する実行ファイル群です。METAFONT, MetaPost, その他多量の実行ファイルが含まれています。必須です。TeX は source specials をサポートしています。

松山道夫さんによって、最初に手がけられ、現在鈴木秀幸さんによってメンテナンスされている日本語 MetaPost (jmpost 0.04) も同梱しています。texmf/doc/jmetapost 以下に簡単な Documents が入っていますので、参考にして下さい。

(07) [dvipdfm-w32.tar.bz2](#)

Mark Wicks による dvipdfm 0.13.2c を Win32 用に Microsoft VC++ 5.0 で Make したものです。DVI を PDF に変換するドライバです。圧縮および Virtual fonts に対応しています。texmf/doc/dvipdfm/base/dvipdfm.pdf がマニュアルです。これは dvipdfm 自身によって作成した PDF です。サポートファイルとして、pdftex-w32.tar.bz2, t1fonts.tar.bz2 をインストールしておいて下さい。texmf.cnf における T1FONTS なる変数を適切に定義しておくことも重要です。JPEG, EPDF などのイメージファイルを埋め込む場合、texmf.cnf に TEXPICTS なる変数が定義してあると、それに従ってファイル検索をします。定義しなかったらカレントディレクトリ以下をサーチします。バージョン 0.11 から PNG 画像にも対応しています。txpx-pazofonts.tar.gz をインストールしておく、Type1 フォントパッケージ TX fonts, PX fonts を使用できるようにしています。なお、この dvipdfm は、九州大学の平田さんによる日本語サポート拡張を取り込んだものです (0.13.3TL-j-p1d)。従って、platex の出力 dvi から、素晴らしい日本語 PDF を作成することができます。縦書きにも対応しており、日本語 MetaPost にも対応しています。おまけとして、hyperref パッケージと共に使う、日本語しおり作成ツール out2uni.exe も入れています。foo.tex がソースとすると、platex foo のコンパイルが終了したあと、out2uni foo とします。この後更に一回だけ platex foo とします。こうして出来た dvi に対して dvipdfm foo とすると、日本語しおりを含んだ PDF ができます。hyperref パッケージは

```
\usepackage[dvipdfm]{color}
\usepackage[dvipdfm,bookmarks=true,bookmarksnumbered=true,%
            bookmarkstyle=toc]{hyperref}
```

のようにして読み込んでおきます。

使用法

```
foo.dvi がカレントディレクトリに存在すれば
dvipdfm foo
とすると、うまく行けば foo.pdf ができます。
dvipdfm
とすると、使用法の説明が表示されます。
```

このパッケージには通常版日本語 dvipdfm (dvipdfm.exe) に加えて、dvipdfmx も同梱してあります。これは、TrueType フォントや CFF/OpenType CID フォントを埋め込むことができる素晴らしいものです。通常版と区別するため、実行ファイル名は dvipdfmx.exe としております。また、コンフィギュレーション設定ファイル名は、通常版のものが dvipdfm.cfg であるのに対して、dvipdfmx 用のものは dvipdfmx.cfg としています。またフォント設定用マップファイル名は、通常版のものが cid.map であるのに対して、dvipdfmx 用のものは cid-x.map としています。CMap の設定が必要ですが、最小限必要なものはパッケージ自体に含めていますから、パッケージを展開するだけで、使用可能になります。より完全な CMap の設定法は以下で述べます。フォントとして、Ryumin-Light, GothicBBB-Medium, HeiseiMin-W3, HeiseiKakuGo-W5 を用いたものは、Acrobat Reader が理解できる predefined フォントなので、埋め込みません。配布パッケージの cid-x.map では、rml に Ryumin-Light、gbm に GothicBBB-Medium を対応させています。TrueType フォント “MS 明朝”、“MS ゴシック” の埋め込みをテストするには、

```
dvipdfmx -f msembed.map dviname
```

としてみてください。

[PDF 暗号化]

dvipdfmx では、オプション -S によって、PDF 暗号化が可能になっています。コマンドラインから、Owner Password と User Password を入力します。dvipdfmx.cfg の K セクションで暗号化キービット数 (40 or 128)、を設定できます。dvipdfmx.cfg の P セクションについては、dvipdfmx.cfg 自身に記述してある説明を読んで下さい。

[CMap のインストール法]

最小限必要なものは、既にインストール済みなので、通常使用には困りませんが、より沢山の CMap をインストールし、dvipdfmx が見つけることができるようにする方法を述べます。gs-7.07 などインストールしている場合には既に沢山の CMap をインストールしてあるはずですが、この場合は、そのリソースディレクトリを texmf.cnf の変数 CMAPFONTS に記述するだけで dvipdfmx から見えるようになります。

```
CMAPFONTS = d:/Resource/CMap//
```

のように記述します。最新の texmf.cnf には変数 CMAPFONTS が既に定義してありますから、これを探して値を自分のパソコンに合致するように変更して下さい。gs-7.07 などインストールしていない場合には、GS-CJK のオフィシャルサイト

```
ftp://ftp.gyve.org/pub/gs-cjk/
```

あるいは、

```
http://www.ring.gr.jp/pub/text/TeX/ptex-win32/gs/
```

から二つのファイル adobe-cmaps-200204.zip と acro5-cmaps-2001.zip を get し、これを適当なリソースディレクトリ、例えば d:/Resource をカレントディレクトリとして展開します。そうして上で述べたように texmf.cnf に

```
CMAPFONTS = d:/Resource/CMap//
```

のように記述します。或いはもっと簡単な方法として、Acrobat Reader をインストールしている場合には十分な CMap がありますから、texmf.cnf の変数 CMAPFONTS に

```
CMAPFONTS = c:/acro5/Resource/CMap//
```

と記述すれば OK です。dvipdfmx は CMap を探す場合、CMAPFONTS に記述してある場所を探します。パッケージに同梱の最小限のファイルは texmf/fonts/cmap にあり、texmf.cnf で適切に CMAPFONTS が定義してありますから使うことができます。

[TrueType フォントの埋め込み法]

上に述べたように、パッケージをインストールしただけで、“MS明朝”と“MSゴシック”を埋め込むように設定してありますが、自分で多様なフォントを埋め込みたい場合のために、典型的な方法を述べておきます。まず、 \TeX で使うときの tfm フォント名を決めます。ここでは myfnt.tfm として説明します。次に rml.tfm に対応する tfm 名 (dvips-tfm 名) を決めます。ここではこれを psmvfnt.tfm として説明します。埋め込む TrueType フォント名が somefnt.ttc (TrueType フォントコレクション) であるとします。

1. texmf/fonts/tfm/ptex/jextra なるディレクトリをつくり、そこに texmf/fonts/tfm/ptex にある jis.tfm を myfnt.tfm なる名前で作ります。
2. texmf/fonts/vf/ptex/jextra なるディレクトリをつくり、コマンドプロンプト、或いは MS-DOS プロンプトで、カレントディレクトリを texmf/fonts/vf/ptex/jextra にします。そうして makejvf myfnt psmvfnt というコマンドを実行します。そうすると myfnt.vf および psmvfnt.tfm ができるので、myfnt.vf はそのままにし、psmvfnt.tfm を texmf/fonts/tfm/dvips なるディレクトリに移動しておきます。
3. texmf/fonts/map/dvipdfm/base にあるファイル cid-x.map を編集し、

```
psmvfnt H :0:somefnt
```

なるエントリを作成しておきます。TrueType フォントコレクション (somefnt.ttc) の場合、使用するフレームをコロンの間に指定します。第一フレームは 0, 第二フレームは 1 を指定します。フレームが一個しかないもの (コレクションではないもの: somefnt.ttf) の場合は必ず 0 を指定して下さい。このように設定しておくと、 \TeX で myfnt なる名前で作成したフォントに対して somefnt なる TrueType フォントを埋め込むことができます。なお、

```
psmvfnt H :0:!somefnt
```

のように、フォント名の直前に記号 ! を付けておくことによって、フォントを埋め込まないように指定することもできます。

[日本語しおり等の作成法]

通常日本語版 dvipdfm.exe では、日本語しおり、日本語文書情報、日本語テキストアノテーションを作成するには、外部プログラム out2uni.exe の助けを必要としますが、

dvipdfmx の実行ファイル dvipdfmx の場合には、dvipdfmx 自身だけで日本語しおりなどを作成することができるようになっています。このためにはソースの始めのほうを、pL^AT_EX の場合は

```
\documentclass[12pt]{jarticle}
\usepackage{atbegshi}
\AtBeginShipoutFirst{\special{pdf:tounicode 90ms-RKSJ-UCS2}}
... ..
```

のようにしておきます。

NTT-jL^AT_EX の場合は、デフォルトのエンコーディングが Shift-JIS の場合でも、PDF 文字列は EUC に決めてありますので、

```
\documentclass[12pt]{j-article}
\usepackage{atbegshi}
\AtBeginShipoutFirst{\special{pdf:tounicode EUC-UCS2}}
... ..
```

のように書き出しておきます。また pL^AT_EX の場合も encoding を Shift-JIS ではなくして使用する場合には、やはり PDF 文字列は EUC になりますので、NTT-jL^AT_EX の場合と同じように CMap EUC-UCS2 を使った記法にしてください。

(08) ***dvipsk-w32.tar.bz2***

dvipsk-5.96 に株式会社アスキーによって作成された日本語化パッチ jpatch p1.7a をあてて Win32 (x86) 用に make したものです。dvi → ps ドライバです。jpatch p1.7a では eclepsf.sty および epsbox.sty にも対応しています。kpathsea のバージョンは T_EX と同じで 5.0.0 です。日本語を扱うには下記の vf-a2bk.tar.bz2 も必要です。下記に示す mftools.tar.bz2 を入手して mktexpk をインストールしておくことも必要です。-SJIS なるオプションで、Shift JIS encoding の出力を得ることができます。このオプションなしのときは JIS encoding となります。説明書が texmf/doc/dvips 以下に入ります。gsftopk 1.19.2 も同梱してあります。mftools.tar.bz2 内の mktexpk を dviout で使用する場合、Ghostscript の助けを借りて、texmf/fonts/map/dvips/base/psfonts.map 等に記述してある PS フォントを使用した DVI ファイルも表示できます。また、VFlib+FreeType を利用する版も同梱していますが、これらについては texmf/doc/dvips 以下の文書をご覧ください。latex, hyperref, dvipsk および distiller を利用して、PDF 文書を作成するとき、日本語を含む bookmarks (しおり) を作成するためのツール

bkmk2uni.exe も入れてあります。これは標準入力から PostScript を読み込んで、distiller に処理させるための PostScript を標準出力に出力します。もとの PostScript の /Title (string), /Subject (string), /Author (string), /Keywords (string), /T (string), /Subj (string), /Contents (string) なる部分の string を bigendian Unicode に変換しています。string に含まれる日本語は Shift-JIS コード (デフォルト) あるいは euc-japan (オプション -e) でなければいけません。pTeX で処理した DVI を dvipsk で処理すると Shift-JIS になります。ただし、NTT-jTeX で処理した DVI の場合には、この日本語文字列はソースが Shift-JIS であっても EUC コードになります。使用法は次の通りです:

```
dvipsk -Ppdf -z -f foo | bkmk2uni >foo.ps
```

あるいは NTT-jTeX の場合は (vf-n2bk.tar.bz2 が必要)

```
dvipsk -Ppdf -z -f foo | bkmk2uni -e >foo.ps
```

オプション -f は、dvipsk の出力を標準出力にするためのものです。こうしてできた foo.ps を distiller で処理して下さい。hyperref パッケージは ltxpkgs.tar.bz2 に入れています。Bookmarks (しおり) は、\section などの情報から自動的に作成されますが、Acrobat Reader の文書情報に表示される内容を埋め込むには、hyperref package のオプションで次のように記述しておきます:

```
\usepackage[dvips, pdftitle={タイトル},%
pdfauthor={鈴木太郎}, pdfsubject={サブタイトル},%
pdfkeywords={TeX, 文書処理, メタフォント}, other keys, ...]{hyperref}
```

数式モードのものを bookmarks にすることはできませんから、数式モードを \section などに使用するときは

```
\section{\texorpdfstring{TeX string}{PDF string}}
```

として、PDF string には数式モードを使わない、代替用の文字列を与えておいて下さい。また、text annotation を挿入するには次のような special をソースに記述しておきます:

```
\special{ps: SDict begin [
/Contents (注釈の本文)
/Rect [70 600 170 700]
/Open false
/Color [0 1 1]
/Title (注釈の題名)
/Subtype /Text
/ANN pdfmark end}
```

さて、dvipsk で埋め込める画像ファイルは、eps 形式だけです。そこで、種々のビットマップ画像を eps に変換する高機能なユーティリティ bmeps.exe を Win32 用に make して同梱してあります。bmeps は Dirk Krause によるものです。bmeps の全機能を dvipsk に取り込んで、dvipsk をコンパイルしなおすこともできますが、dvipsk をあまり複雑にするのは好ましくないと判断し、そのようにはしていません。bmeps.exe を、dvipsk を補助する外部プログラムとして使用するという立場を選択しています。libpng, libjpeg, zlib, libpnm/ppm/pgm/pbm とリンクしていますから、png, jpeg, pnm, ppm, pgm, pbm 形式のファイルを eps に変換することができます。dvipsk を insecure mode (オプション-R0) で使用すれば、上記画像を dvipsk の実行時に自動的に eps に変換しながら埋め込むこともできます。詳細は dvipsk-w32.txt を読んで下さい。

このファイルに、Adobe から入手できる Euro type1 フォントと、簡単なマクロパッケージ euro-p.sty を含めています。`\usepackage{euro-p}` としておけば、`\euro`、`\euromono`、`\eurosans`、`\euroserif` によって、ユーロ通貨記号を出力することができます。`\euro` は `\euromono` と同じです。

また、Antonis Tsolomitis (University of the Aegean) によるギリシャ語用 type1 フォント (kerkis)、及びそれを使用する L^AT_EX パッケージも入れています。kerkis フォントは、本来はギリシャ語エンコーディング LGR で使用するものですが、OT1, T1 でも使用できるように作られています。

```
\usepackage{deflgr,kerkis,kmath}
```

とすることにより、Bookman old style のフォントを使用することができます。コマンド `\LaTeX` などが再定義されているので、`\section` 中でエラーになりますが、これは `\protect` で避けることができます。日本語環境で kerkis フォントを使用すると

```
I search font definition file:
```

なるメッセージが沢山出て、処理速度が著しく遅くなる場合があります。このような場合には `ls-R` を作成しておく、処理が速くなります。`ls-R` を使用する場合には、`texmf trees` に何か新しいものを入れたら、必ず `mktexlsr` コマンドを実行するのを忘れないようにすることが大切です。なお、データベース `ls-R` を使用したく無くなった場合、消去するには `deltexlsr` コマンドを実行します。

最後に、同梱してある実行ファイル `udvips.exe` は、JIS コード、Unicode、CID コードが混在する PostScript ファイルを出力できるもので、齋藤修三郎さんによる `utf` パッケージを使った場合の `dvi` を `ps` に変換する場合に用いるものです。稲垣淳さんによる `ums` パッケージを拙作の仮想フォント経由で使用する場合の出力 `dvi` に対しても `udvips` を使います。

(09) [***jtex-2.1-w32.tar.bz2***](#)

千葉大学の桜井さんによる NTT J_TE_X `jtex-2.1` の本体です。必要フォントおよびマクロを含みます。この NTT J_TE_X は `source specials` をサポートしています。

(10) [***ltxpkgs.tar.bz2***](#)

AMS-LaTeX, graphics, mfnfss, psnfss, tools, hyperref の各パッケージその他を入れています。

(11) [***luatex-w32.tar.bz2***](#)

Taco Hoekwater, Hans Hagen, Haltmut Henkel さんによる Lua_TE_X です。Lua_TE_X は、スクリプト言語 Lua が埋め込まれていて、Lua による low level 設定ができるようになっています。TeX, e-TeX, Omega, Aleph の主要部分がマージされ、更に種々の機能追加がされています。まだ完成版ではなくて、発展途上のエンジンです。

(12) [***makeindex-w32.tar.bz2***](#)

インデックス作成プログラムです。`makeindex` はオリジナルの `makeindex` バージョン 2.14 です。これは日本語には対応していません。日本語対応の `jmakeindex` は、旧アスキー j_TE_X 付属のものを Win 32 (x86) 用に `make` したものです。`kpathsea 5.0.0` に対応しています。`mendex` はアスキー (株) によるものであり、`kpathsea 5.0.0` に対応しているとともに、読みに関する辞書が使えるように改善されたものです。

(13) [**manual.tar.bz2**](#)

マニュアルです。PDF に変換したのものも含んでいます。

(14) [**oldformat.tar.bz2**](#)

L^AT_EX 2.09, pL^AT_EX 2.09 および 旧 アスキー pT_EX 用のフォーマットファイルを集めたものです。(p)L^AT_EX 2.09 のソースファイルも大抵は (p)L^AT_EX 2_ε の 2.09 互換モードとしてタイプセットできますが、たまに、どうしてもだめなものもあります。そのような場合に備えて 2.09 の環境も作成しておくとなので提供するものです。このパッケージにあるフォーマットファイルは次のようなものです:

latex209.fmt	: 欧文用 LaTeX 2.09
ptexold.fmt	: 旧日本語 plain pTeX
amstex.fmt	: 日本語対応 plain AMSTeX
platex209.fmt	: 縦書き対応日本語 LaTeX 2.09
qlatex209.fmt	: 横書き用日本語 LaTeX 2.09
alatex209.fmt	: 日本語対応 AMS-LaTeX based on LaTeX 2.09

qlatex209 は jlatex209 としたいところですが、NTT J_TE_X 用に jlatex209 なる名前を使用しているため、このような名前にしています。旧 LaTeX 2.09 と語呂あわせをして連想するとよいでしょう。ptexold.fmt は何のためにあるのかというと、`\newfam` 命令を使用するとき、ptex.fmt ではスペース不足になることがありますので、そのような場合に利用して下さい。コマンドの名前が気に入らない場合には、実行ファイルの名前とフォーマットファイルの名前を対にして変更して下さい。例えば alatex209 を amsltx11 に変更するには

alatex209.exe → amsltx11.exe

alatex209.fmt → amsltx11.fmt

とします。勿論この場合、texmf.cnf も書き換える必要があります。

(15) [**oldinputs.tar.bz2**](#)

oldformat.tar.bz2 と共に使用する input files です。(p)L^AT_EX 2_ε の packages としても使用可能なものがあると思います。

(16) [**pdftex-w32.tar.bz2**](#)

pdf_TE_X version 1.40.10-2.2 です。ただしエンジン名は pdftex です。過去には pdftex と pdfetex という違うものがありましたが、現在では一種類のみで、e-TeX 拡張されたものだけです。フォーマットファイル pdftex.fmt, pdflatex.fmt, もあります。pdfetex は pdftex と全く同じであり、pdfelatex は pdflatex と全く同じです。(17) に Blue Sky Research による type 1 fonts を入れておきましたので、(16), (17) をインストールすれ

ば即 $\text{T}_\text{E}\text{X}$ ソースから直接 PDF ファイルを作成できます。 $\text{T}_\text{E}\text{X}$ や $\text{L}^{\text{A}}\text{T}_\text{E}\text{X} 2_\epsilon$ のソースからデフォルトで直接 PDF を出力しますが、はじめに `\pdfoutput=0\relax` としておけば、出力は PDF ではなく DVI となります。マニュアルは `texmf/doc/pdftex/base` にある `pdftex-a.pdf` です。スクリーン表示用は `pdftex-s.pdf` です。ソースは `pdftex-t.tex` ですが、これをタイプセットするには `ConTEXt` をインストールしておく必要があります。また同ディレクトリにある `PDFTEX-W32.txt` というファイルを読んで下さい。用紙のサイズは A4 がデフォルトになっています。PNG に加えて JPEG, EPDF も取り込めるようになっていました。EPDF とは一ページだけの PDF と思えばよいでしょう。(正確には左下座標が原点の `/MediaBox` 文のあるもの)。EPSF を EPDF に変換する perl スクリプトを同梱してあります。 $\text{L}^{\text{A}}\text{T}_\text{E}\text{X} 2_\epsilon$ の graphics package では、pdftex オプションにより、PNG(.png), JPEG(.jpg), EPDF(.pdf) と MetaPost 出力の EPSF (拡張子を .mps にすること) をとり込むことができます。

pdf $\text{T}_\text{E}\text{X}$ -1.40.10-2.2 では、バージョン 0.13d 以下の場合と比べて、プリミティブが大きく変更され、改善されています。従って例えば graphics package を使用するとき、バージョン 1.40.10-2.2 に対応していないものを使用すると、うまくゆきません。この場合、最新の graphics package (当サイトにある `ltxpkgs.tar.bz2` でも良い) をインストールして下さい。また、hyperref package を使用する場合も、古いものは pdf $\text{T}_\text{E}\text{X}$ -1.40.10-2.2 では使用できません。バージョン 1.40.10-2.2 に対応したものは `ltxpkgs.tar.bz2` にあります。DVI 出力時には source specials が有効です。デフォルトの PDF 出力時には source specials は意味を持ちませんので、warning が出ます。pdftex.map で、Type1 フォントパッケージ TX fonts および PX fonts を使用できるようにしています。ただし `txpx-pazofonts.tar.gz` をインストールしておかなければいけません。

このパッケージには、xpdf の作者 Derek B. Noonburg さんによるユーティリティを 5 個入れてあります。このうちの pdftotext で日本語を扱う方法について述べておきます。まず Ghostscript で CID フォントを扱うようにしてあれば、既に CMap ファイル等を含んだ Resource 用ディレクトリがあると思います。Resource 用ディレクトリが無ければ新たに作成します。これが `D:/Resource` であるとして説明します。違う場合には読み換えて下さい。まず pdf $\text{T}_\text{E}\text{X}$ の Documents が入っているディレクトリに `pdftotext-suppl.zip` なるファイルがありますから、これを `D:/Resource` をカレントディレクトリとして展開します。

```
unzip -o pdftotext-suppl.zip
```

次に $\text{T}_\text{E}\text{X}$ のバイナリディレクトリに `xpdfrc` なるテキストファイルがあるはずですから、これを編集します。Resource 用ディレクトリが実際に `D:/Resource` である場合には編集しなくても良いようにしてあります。要するに `D:/Resource` となっている部分を、実際のディレクトリに書き直すだけです。そうすると例えば

```
pdftotext -enc Shift-JIS foo.pdf foo.txt
```

とすることによって、日本語 PDF `foo.pdf` からテキストを取り出して、テキストファイル `foo.txt` を作成することができます。

おまけとして、`pdfLATEX` で日本語 PDF を作成するためのサポートファイル `topdftex.exe` も同梱してあります。これを使用するための必要条件、および使用方法に関しては `PDFTEX-W32.txt` に詳しく記述してあります。

更に、pdf encryption に関して、Paulo Soares さんによる、java によるプログラムを入れ、これを簡単に使用するための `pdfcrypt.exe` を作成して入れています。Java runtime さえあれば実行できます。

```
pdfcrypt [-K keybit] [-P permission] old.pdf new.pdf [info]
```

のようにします。keybit と permission のデフォルト値は、`texmf/pdftocrypt/pdftocrypt.cfg` に記述しておきます。オプションの `[info]` は、文字列のペアであり、

```
Title      "Title string"
Subject    "Subject string"
Author     "Author string"
Keywords   "Keywords string"
```

の四個が指定可能です。これを指定して、文書情報を設定、変更できます。ただし使用可能文字は ASCII 文字に限られます。なお、owner password と user password の入力が必要ですので、キーボードから入力して下さい。詳細については `texmf/pdftocrypt` にある文書を読んで下さい。以前の pdf_LA_TE_X では、pdf encryption が primitive レベルでサポートされていましたが、新しい pdf_LA_TE_X では、この機能は削除されています。

(17) [*tlfonts.tar.bz2*](#)

Blue Sky Research による Type 1 CM fonts、およびその他の Type1 fonts を入れておきました。Type 1 フォントは重要なので、このパッケージはぜひインストールしておいて下さい。

(18) [*timesnew.tar.bz2*](#)

Acrobat Reader 4.0以降には、MonotypeのTimesNewRomanシリーズと、ArialシリーズのType1フォントが入っています。Windows版にはPFBとPFMが入っています。TeXで使用するフォントを作成するには、AFMファイルが必要なのですが、plainの本田さんにより、Ghostscriptのpf2afmの出力と、ユーティリティ pfm2afm の出力をマージすれば、さしあたり使用可能なAFMが作成できることが指摘されました。この方法で作成したフォントと、簡単なLaTeX packageをtimesnew.tar.bz2に入れています。また、本田さんによる、フォントを簡単に変更するパッケージ chfont.sty を同梱させてもらっています。これについては少しあとで説明します。

使用するには、以下のようにして下さい:

1. texmf/fonts/type1にmonotypeなるディレクトリを作成し、Acrobat Reader 4.0以降のディレクトリ.../Resource/FontにあるPFBファイルのうち、次の8個をtexmf/fonts/type1/monotypeにコピーします:

```
_a_____.pfb
_ab____.pfb
_abi____.pfb
_ai____.pfb
_eb____.pfb
_ebi____.pfb
_ei____.pfb
_er____.pfb
```

他のファイルはAdobe純正PFBであり、既にAcrobat Reader 3.0からコピーしてあるでしょうから、コピーしないようにして下さい。

2. timesnew.tar.bz2を、TeXをインストールしたトップディレクトリ(デフォルトとしているのはc:/w32tex)をカレントディレクトリとして展開します:

```
c:
cd \w32tex
tar zxvf timesnew.tar.bz2
```

以上で準備完了となります。`\usepackage{timesnewp}` とすると、`\usepackage{times}` とした場合のTimesがTimesNewRomanに置き換えられ、HelveticaがArialに置き換えられます。数式がある場合、`\usepackage{mathmnsx}` とすると、`\usepackage{mathptmx}` とした場合のTimesがTimesNewRomanに置き換えられます。ただし全てが置き換えられるわけではありません。zptmcm7m, zptmcm7t,

zptmcm7v, zptmcm7y はそのまま用いられます。`\usepackage{mathmnsxx}`とすると、`\usepackage{mathptmx}`とした場合の Times が TimesNewRoman に置き換えられます。この場合には全てが置き換えられます。zptmcm7m, zptmcm7t, zptmcm7v, zptmcm7y のかわりに ymnscm7m, ymnscm7t, ymnscm7v, ymnscm7y が使用されます。(これらの名前は y で始まり、Karl Berry の命名規則とちょっと違いますが、他のパッケージとフォント定義ファイルの名前が衝突するのを避けるためにこのようにしています。) また本田さんによる、chfont.sty は次のように使用します。

```
\usepackage[rm=cmr,sf=cmss,tt=cmtt]{chfont}
```

とすると、通常のデフォルトとなります。つまり、こうしなかった場合と同じです。

```
\usepackage[rm=ptm,sf=phv,tt=pcr]{chfont}
```

とすれば、PSNFSS の `\usepackage{times}` とした場合と同じになります。また

```
\usepackage[rm=mns,sf=mal,tt=pcr]{chfont}
```

とすると、`\usepackage{timesnewp}` とした場合と同じになります。これ以外に

```
\usepackage[rm=mns,sf=phv,tt=cmtt]{chfont}
```

など自由に組み合わせを変更して使用できる点が便利です。詳細は texmf/doc/timesnew にある chfont.dtx を L^AT_EX でコンパイルし chfont.dvi を preview してみてください。

[注意]

1. dviout, dvips(k), pdf(e)tex, dvipdfm(x) で使用するには、texmf/web2c/updmap.cfg なるファイルで、

```
#Map monotype-mns.map
```

となっている行のコメント文字 `#` を消して

```
Map monotype-mns.map
```

に変更した後、`updmap` コマンドを実行して下さい。あるいは、これら全ての作業のかわりに

```
updmap --add monotype-mns.map
```

とするだけでもよいです。これによって、マップファイルに必要事項が書き込まれます。

2. `dvi2ps` から利用するには、最新の `dvi2ps-4.1j-w32.tar.bz2` をインストールしておいて下さい。 `fontdesc` として、 `bikanurwt1` あるいは `vf*` を使う場合、 `Type1` を埋め込みます。

(19) [***txpx-pazofonts.tar.bz2***](#)

Young U. Ryu による `Type1` フォントパッケージ `TX fonts` と `PX fonts` および Diego Puga による `mathpazo` です。まず、 `texmf/doc/txfonts`, `texmf/doc/pxfonts` の文書をお読み下さい。 `dvips(k)`, `dvi2ps`, `dvipdfm`, `pdfTeX`, `dviout` でこれらのフォントを使用できるように、該当するパッケージで実現してあります。

(20) [***vf-a2bk.tar.bz2***](#)

`dvipsk-w32.tar.bz2` あるいは `dvi2ps-4.1j-w32.tar.bz2` と共に用いる `virtual fonts` などです。株式会社アスキーで新たに開発された日本語 `virtual font` 作成ツール `makejvf-1.1a` で作成したものです。 `vftool-1.2` で作成したものよりも、日本語用に最適化されていると言われています。

(21) [***context-doc.tar.bz2***](#)

`ConTeXt` に関する `Documents` です。非常にサイズが大きいのので注意して下さい。 `Documents` は殆ど `PDF` 形式であり、 `texmf/doc/context` 以下に入ります。入れ替えていないので、古すぎるものしか無いかもしれません。

(22) [***context.tar.bz2***](#)

`ConTeXt` です。最低 `pdfTeX` と `perl` と `ruby` をインストールしておかないといけません。展開するだけで基本的にインストール終了となります。使用できるエンジンは、 `pdftex`, `xetex`, `luatex (mark iv)`、です。これに加えて、 `W32TeX` では `eptex`, `euptex` も使用できるように拡張しています。フォーマットファイルは

```
fmtutil --byfmt cont-en
```

とすることによって作成されます。

```
texexec --help
```

で簡単なメッセージが表示されます。foo.tex (ConTeXt に従って作成されたソース) を処理するには

```
texexec foo      --- (PDF 出力)
```

とします。ConTeXt コマンドを表示するための texshow コマンドを実行するためには、perl の Tk module が必要です。

```
fmtutil --byfmt mptopdf
```

とすると、MetaPost の出力を、PDF に変換できるようになります。

```
mptopdf foo.3
```

などとします。ここで foo.3 は MetaPost の出力ファイル名の例です。ただし pdftex をインストールしておく必要があります。

(23) [cweb-w32.tar.bz2](#)

D. E. Knuth と S. Levy による CWEB の最新版 (Version 3.64) です。Win32 版は Kpathsea 5.0.0 に対応させています。T_EX のインストールディレクトリで cweb-w32.tar.bz2 を展開してから、以下の作業をして下さい：

1. mule あるいは Meadow 用の cweb.el は X/share/mule2/lisp に入りますから、mule 或いは meadow のインストールディレクトリが異なっている場合、これを該当ディレクトリに移動してから、不要なディレクトリ X/share/mule2 以下を削除して下さい。ここで X は T_EX をインストールしたトップディレクトリのフルネームです。デフォルトでは X = c:/w32tex です。
2. X/share/texmf/tex/plain/misc/cwebmac.tex は pdfT_EX にも対応しています。

このパッケージには、Julian Gilbey による ctie.exe も入れています。ctie.exe は、複数の change ファイルから一つの change ファイルを作成したり、マスター cweb ファイルを作成するためのものです。これも Kpathsea 5.0.0 に対応しています。web における対応物は tie.exe です。

(24) **[dvi2ps-4.1j-w32.tar.bz2](#)**

千葉大学の桜井さんによる、dvi2ps-4.1j を Win32 用に make したものです。dvi2dvi.exe も含んでいます。texmf/web2c/texmf.cnf に次のようなエントリがありますが、これは dvi2ps によって利用されます:

```
%%-----
% special variables for dvi2ps ( Win32 )
% -----
DVI2PATH = $TEXMFMAIN/dvi2ps
DVI2LIB  = $DVI2PATH
DVI2FONTDESC = fontdesc
%%-----
```

(25) **[dvitools-w32.tar.bz2](#)**

dvi ファイルを編集するためのいくつかのコマンド、dvidvi, dv2dt, dt2dv, dvibook, dviconcat, dviselect, dvitodvi です。また dvi.png も入れています。texmf/doc/dvitools 以下の文書を読んで下さい。

(26) **[lcdf-typetools-w32.tar.bz2](#)**

Eddie Kohler さんによる lcdf typetools です。cftot1, mmafm, mmpfb, otinfo, otftotfm, tldotlessj, tllint, tlreencode, tltestpage, ttftotype42 を含みます。このうち otftotfm が Kpathsea に対応しています。texmf/doc/lcdf-typetools/base/ にある文書を読んで下さい。

(27) **[omega-aleph-w32.tar.bz2](#)**

Unicode によって多言語対応を目標としている Omega-1.23.2.3 です。T_EX 3 で 8 ビットであった諸量が 16 ビットに拡張されています。通常の使用では T_EX (omega), L^AT_EX 2_ε (lambda) と思ってよいでしょう。実行ファイルとライブラリを含みます。更に、e-TeX と Omega をマージして作成された Aleph と、関連するフォーマットファイル (aleph.fmt と lamed.fmt) も入れてあります。Aleph とは、e-TeX primitive の大部分を使用できる Omega と思えばよいと思います。コマンド名は omega にあたるものが aleph, lambda にあたるものが lamed です。Omega 用の dvips である odvips.exe

と、二つのフォーマットファイル `omega.fmt`, `lambda.fmt` も入れてあります。 `dvipsk-w32.tar.bz2` は別にインストールしておいて下さい。 `odvips.exe` は、鈴木秀幸さんによるパッチを適用して、level 1 ofm に対応しています。詳細については `texmf/doc/omega` 以下にある文書をご覧下さい。 `torture.tex` と `torture.ps` は、アラビア語についての説明とサンプルになっています。 `source specials` をサポートしています。

(28) [omegaj-w32.tar.bz2](#)

Matt Gushee さんによる、Omega-j を Win32 環境で使用できるようにしたものです。フォントは `omsmin` と `msgoth` を使用できるようにしています。Omega-j は Omega によって日本語をタイプセットする試みです。オリジナルの `injis.ocp` は不完全だったので、東京理科大学の松田さんによって改良されたものを同梱させてもらっています。また同じく松田さんによる `ineucjp.ocp` も同梱させてもらっています。Windows では、Shift-JIS を `input` とすることができると便利なので、`insjis.ocp` と `insjisknj.ocp` を作成して入れてあります。 `insjisknj.ocp` は、下記に述べるフォーマットファイル作成時に `preload` させるためのもので、汎用のものではありません。フォーマットファイルは日本語 plain Omega, 日本語 L^AT_EX 2.09, 日本語 L^AT_EX 2_ε の全てを作成できるようにソースを同梱しています (それぞれ `omegaj`, `lmj209`, `lambdaj`)。 (`omegaj`, `lmj209`, `lambdaj` の制限事項: `\verb` コマンド、および `\verbatim` 環境の内側で日本語を使用することはできません。) GT フォントを `lambdaj` から使用することもできます。詳細は `texmf/doc/omegaj` 以下にある文書 `OMEGA-J-W32.txt` を読んで下さい。 TrueType フォント `mmincho.ttc`, `msgothic.ttc`, `gt200001.ttf`, `gt200002.ttf`, `gt200003.ttf`, `gt200004.ttf`, `gt200005.ttf`, `gt200006.ttf`, `gt200007.ttf`, `gt200008.ttf`, `gt200009.ttf`, `gt200010.ttf`, `gt200011.ttf`, `gt2000k1f.ttf`, `gt2000k2f.ttf` から Type1 フォントを作成するスクリプトも入れてあります。

`omegaj`, `lambdaj`, `lmj209` は Shift-JIS インプットを前提としたものです。一方 Omega 自身は多様なインプットに対応しています。そこで色々なエンコーディングで行分割、禁則処理を実現するため、`inunijapan.ocp` と、lambda 用のパッケージ `unijapan.sty` を同梱してあります。これを利用すると、Shift-JIS、JIS、EUC-JAPAN、big endian UCS2、little endian UCS2、utf-8 の全ての場合に `lambdaj` と同等の出力を得ることができます。 `lambdaj` ではなくて、オリジナルの `lambda` を使うことにご注意下さい。GT フォントも、`\UMS` コマンドも `lambdaj` と同じように使うことができます。それぞれの場合の、ソースの `preamble` の書き方は `texmf/doc/omegaj` 以下に入れてある `OMEGA-J-W32.txt` を読めばわかるようにしています。なお、以前は `notepad` や `wordpad` で utf-8 などとしてセーブしたものは、ファイル先頭の 2 あるいは 3 バイト (エンコーディングの種類を示すバイト) のため、`omega` がエラーになっていましたが、新しいものではこれらのバイトを読みとばすようにしていますから、エラーにはなりません。

(29) [plain2-2.54-w32.tar.bz2](#)

内田さんによる plain2-2.54 を Windows 用にコンパイルしたものです。plain2 はプレーンテキストファイルから、L^AT_EX 用ソースや HTML 等を出力するアプリケーションです。

(30) [**psutils-w32.tar.bz2**](#)

Angus Duggan さんによる、PostScript ユーティリティです。一部のものは perl を必要とします。texmf/doc/psutils/base に文書がありますのでお読み下さい。

(31) [**sam2p-w32.tar.bz2**](#)

Szabó Péter さんによる、bitmap ファイルを eps や pdf 等に変換するアプリケーションです。texdoc sam2p として文書をお読み下さい。

(32) [**t1utils-w32.tar.bz2**](#)

Type1 フォント用ユーティリティ を Win32 用にコンパイルしたものです。どのコマンドも --help オプションによって、簡単な説明を表示します。t1ascii.exe は PFB 形式のものを PFA 形式の Type1 フォントに変換します。

```
t1ascii foo.pfb foo.pfa
```

とすれば、存在する foo.pfb から foo.pfa が作成されます。t1binary.exe は逆に PFA から PFB への変換をします。

```
t1binary foo.pfa foo.pfb
```

のようにします。t1disasm.exe は Type1 フォント (PFA, PFB どちらでもかまいません) を、人間が読める形式のテキストファイルに変換します。

```
t1disasm foo.pfb foo.psf
```

のようにします。foo.psf はテキストファイルですから、コマンド類が理解できる場合には編集して変更することができます。このようなテキストファイルから PFB 形式ファイルや PFA 形式ファイルに変換するには、t1asm.exe を使います。

```
t1asm -b foo.psf foo.pfb
```

あるいは

```
t1asm -a foo.psf foo.pfa
```

のようにします。オプション `-b` はデフォルトとなっています。t1unmac.exe は Macintosh PostScript フォントを、PFB あるいは PFA 形式ファイルに変換するものです。

```
t1unmac -b input foo.pfb
```

あるいは

```
t1unmac -a input foo.pfa
```

のようにします。t1mac.exe は PFB あるいは PFA 形式の Type1 フォントを Macintosh PostScript フォントに変換するものです。

```
t1mac foo.pfa output
```

のようにします。詳細に関しては `texmf/doc/t1utils` にある Documents をお読み下さい。

(33) [**tex4htk-w32.tar.bz2**](#)

TeX4ht は Eitan M. Gurari さんによるもので、TeX, LaTeX ソースから、html, xhtml, xml などを出力するものです。日本語を扱うには、NTT-jTeX の jtex, jlatex を使用します。NTT-jTeX は日本語を 256 以内ずつにわけて、サブフォントとして実現してあるので、実際上欧文 TeX とほぼ同じで TeX4ht に手を加える必要がないわけです。ただし、TeX4ht が利用する形で、フォントの準備をする必要があります。この作業をさ

れたのは行木孝夫さんです。tex4htk-w32.tar.bz2 には、行木さんによる、Unicode の dnp-htf-fonts.tar.gz および、EUC-JP の dnp-htf-fonts-euc.tar.gz を Shift-JIS に変換したものを予め含めさせてもらっています。pTeX, pL^AT_EX に依存するソースは不可であることにくれぐれもご注意ください。4ht ファイルが準備してあるものでないと、変換がうまく行きませんから、クラスファイルやパッケージ類は欧文のものを使用するようにして下さい。欧文のものであれば、大抵のものに 4ht ファイルが準備されています。tex4htk-w32 で配布する TeX4ht は、欧文の場合も日本語の場合も jtex, jlatex を使うようにしています。また Kpathsearch 5.0.0 に対応させています。更に、通常 Unix 系ではスクリプト、Windows ではバッチファイルとして実現してある各種コマンドは、全て exe 形式の実行ファイルとしてあります。これらは Fabrice Popineau さんによる runht.c を修正して作成した runht.exe のコピーです。(ht.exe, htlatex.exe, 等々)。自分自身の名前から、振る舞いを決定するようになっています。オプション `--help` でヘルプメッセージを表示します。単に tex4htk-w32.tar.bz2 を展開しただけではインストールしたことになりません。詳細については、texmf/doc/tex4ht/win32 に入っている tex4htk-w32.txt を読んで下さい。

(34) [**texinfotools-w32.tar.bz2**](#)

Texinfo-4.13 中で、Win32 環境でも有用と思われる部分をパッケージに含めました。Texinfo 用インデックス作成プログラム `texindex.exe` は森下 お代官様 Ma N M O S さんによるパッチ (texindex-japanese-patch) によって Shift JIS コード用に日本語化してあります。また `makeinfo.exe` も Shift JIS コード用に日本語化したつもりです。Info ファイル作成プログラム `makeinfo.exe` は emacs buffer 内で Info ファイルを作成する場合よりもエラーチェックが厳しいのでご注意ください。--force オプションを指定すると、エラーがあっても出力します。`makeinfo.exe` は --html オプションにより、HTML を出力することもできます。日本語の場合も HTML を出力できると思います。ただし入出力は Shift JIS コードとなっていますので、出力はあとで JIS に変換したほうが良いでしょう。日本語の場合、`@setfilename` の次の行に `@documentlanguage ja` と記述しておく、HTML は `<html lang="ja">` で始まります。シェルスクリプト `texi2dvi` は、web2c-2009-w32.tar.bz2 の documents 用ディレクトリに同梱してある Korn shell `sh.exe` (Windows NT/2000/XP 専用、Windows 98 等では使用不可) および Cygnus GNU-Win32 の `bash` で動作すると思います。TEMP 環境変数が必須ですが、Cygnus `bash` の場合にも `TEMP=//c/tmp` などとしてはだめで、`TEMP=c:/tmp` などとしなければいけません。`TEMP=c:\tmp` は OK です。`TEMP=//c/tmp` としていけない理由は、`ptex //c/tmp/foo.texi` などとできないからです。`TEMP=c:\tmp` などの場合にはスクリプトの中で必要に応じて `\` を `/` に変換しています。`texi2dvi` はいろいろな GNU tools を呼び出すので、GNU tools をインストールしていないと使用できません。`texi2dvi` が使用する TeX コマンドは `ptex` をデフォルトにしていますが、環境変数 `TEX` で変更できます。また、ソース `foo.txi` が欧文だけなら

```
texi2dvi --pdf foo.txi
```

とすると、pdfTeX によって PDF を出力します。その他についてはスクリプト自身を参照して下さい。texi2dvi が使用できない環境で GNU マニュアルなどをタイプセットする場合は

```
ptex foo.txi
texindex foo.??
ptex foo.txi
ptex foo.txi
```

のようにします。クロスリファレンスがまだ解決できないというメッセージが出るようなら、適当にコマンドを繰り返して下さい。texinfo.tex は pdfTeX に対応していますが (Version 0.13d 以下及びバージョン 1.40.10-2.2 にも対応)、pTeX と NTT-jTeX にも対応させています。その他の詳細は texmf/doc/texinfo 中のドキュメント類をご覧ください。

(35) [**tiff2png-w32.tar.bz2**](#)

pdfTeX では PNG format の画像もとり込めるようになっています。これは TIFF から PNG へ画像フォーマットを変換するものです。引数なしで実行すると使用方法を表示します。pdfTeX は TIFF を直接埋めませんから、このプログラムで PNG に変換して埋め込んで下さい。

(36) [**ttf2pk-w32.tar.bz2**](#)

FreeType 1.4 のライブラリをリンクした ttf2pk と ttf2tfm です。ttfdump 0.5.5 も入れています。TeX のインストールディレクトリで展開してから、texmf.cnf に

```
TTF2TFMINPUTS = .;$TEXMF/ttf2pk//
TTF2PKINPUTS = .;$TEXMF/ttf2pk//
```

なるエントリを作成すると、インストールされます。最新の texmf.cnf にはこれらが既に記述してあります。これらは、先述 (29) の Omega-j を使用するとき必須となります。ttf2pk についての詳細は、東京理科大学の松田さんの Web サイトにあります。

(37) [**ttf2pt1-w32.tar.bz2**](#)

TrueType フォントから Type1 フォントを作成する `ttf2pt1.exe` です。`omegaj-w32.tar.bz2` に入っている、Type1 フォント作成用スクリプトを実行するためには必須です。FreeType 2 のライブラリとリンクしており、Kpathsea にも対応させています。`texmf/doc/misc/ttf2pt1` に文書がありますのでお読み下さい。

(38) [**txtutil.tar.bz2**](#)

テキストファイルの行末コードを各オペレーティングシステム用に変更するだけの簡単なプログラムです。入力ファイルはどの形式でも可です。0x0d (Mac) にするもの、0x0a (UNIX) にするもの、0x0d0x0a (DOS, Windows) にするもの、があります。テキストファイル以外には決して使用しないで下さい。ほぼ確実にファイルが破壊されます。TeX が書き出すテキストファイルの行末は 0x0a になるようにしています。`notepad.exe` などで読む場合はご面倒ですが `todos.exe` で 0x0d0x0a に変更して下さい。通常のエディタなら無変更で読めるでしょう。

(39) [**ums.tar.bz2**](#)

稲垣淳さんによる pL^AT_EX 用のパッケージ `ums.sty` および関連するサポートファイルを入れたものです。これは稲垣さんのご厚意により配布するものです。元々、Unicode (subfont) で作成した `msmin???.tfm`, `msgoth???.tfm` を pL^AT_EX で比較的単純に利用し、Windows 標準の `msmincho.ttc`, `msgothic.ttc` なるフォントを利用しようというものでしたが、これらのフォントが存在しない環境でも、Unicode の範囲内にある多くの文字を利用できるようにしてあります。このために便利な dvipdfmx 用の仮想フォントも同梱しています。`texmf/doc/misc/ums` に文書がありますのでお読み下さい。

(40) [**uptex-w32.tar.bz2**](#)

Tanaka Takuji さんによって内部エンコーディングを Unicode にされた pTeX, すなわち upTeX です。入力エンコーディングのデフォルトは utf8 です。他に関連する一連のアプリケーションを含みます。

(41) [**utf.tar.bz2**](#)

齋藤修三郎さんによる pL^AT_EX 用のパッケージ `utf.sty` および関連するサポートファイルを入れたものです。これは `jfm` と `jvf` の技術で、UTF16 の範囲内にある広大な文字種を使用可能にするものです。更に、CID コードで文字を指定することもできるようになっています。これにより、ヒラギノ (Pro 版) など、Adobe-Japan1-5 のコード範囲をサポートするフォントが持つ、全てのグリフを pL^AT_EX で処理し、`udvips + distiller` などで出力することが可能になっています。齋藤さんのご厚意で再配布することが可能になりました。`texmf/doc/misc/utf` に文書がありますのでお読み下さい。

(42) [**vf-n2bk.tar.bz2**](#)

NTT J_TE_X 用の virtual fonts です。これをインストールすれば、NTT J_TE_X の DVI ファイルを dvipsk によって PostScript に変換することができます。

(43) [**xetex-w32.tar.bz2**](#)

Jonathan Kew さんによる XeTeX です。内部で拡張 dvi フォーマット (xdv) を経由して、dvi_{pdf}mx を XeTeX 用に拡張した xdv_{pdf}mx ドライバによって pdf に変換します。xdv ファイルを出力したあと、xdv_{pdf}mx によって pdf 変換することも可能です。OpenType フォントを ICU ライブラリで layout するのが特徴です。詳細については付属のドキュメントや、WEB 情報を参照してください。

(44) [**texinst757.zip**](#)

Web2C-2009 の簡易インストーラです。最初にインストールディレクトリにこれを展開します。実体は texinst757.exe, tar.exe, gzip.exe です。おまけとして、圧縮効率の高い bzip2.exe、関連ツール bzip2recover.exe、および ln.exe も入れています。PATH がバイナリディレクトリに通った後では、これらの実行ファイルをバイナリディレクトリ (デフォルトは c:/w32tex/bin) に移動しておくといいでしょう。なおここに入れている tar.exe は、圧縮フィルタとして gzip に加えて bzip2 も使用できるようにしています。オプション文字は j を採用しています。長い形式は --bzip2 です。従って

```
tar jcf foo.tar.bz2 bar      (作成)
tar jxvf foo.tar.bz2        (展開)
```

のような使い方ができます。Windows 95/98 の場合には、展開の際ディレクトリ作成日付けの変更ができないので、Warning が出ますが無視して下さい。

ln.exe は、Windows 2000/XP/NT の NTFS 専用で、ハードリンクを作るプログラムです。リソースキットには同じ役割をするものが含まれています。ここに提供する ln.exe は、三浦重喜さんによるもので、リソースキットを持っていない場合のためのものです。

5 日本語サポート Ghostscript

T_EX を使う場合、Ghostscript はいろいろな場面で役に立ちます。日本語サポート Ghostscript の Windows 版に関するいくつかのファイルを配布していますので、以下に簡単に説明します。

(01) [**gs707w32full.zip**](#)

gnu Ghostscript 7.07 は、gs-cjk team による成果が採り入れられていて、TrueType フォントを CID keyed font として使用できるようになっています。従ってそのままの形で Windows 上でも日本語を表示印刷できます。また、日本語 pdf 作成もできます。

これは Windows 上における gnu Ghostscript 7.07 のバイナリです。WINAPI を使って表示する付加機能により、OpenType フォントなど、OS が認識している任意の日本語フォントで高品質の表示が得られるようになります。WINAPI によるドライバは、浅山和典さんによるものです。ここで使用している WINAPI ドライバには、Bitmap フォントとしてアクセスする機能が付加してあり、これをデフォルトとして設定してあります。

なお、WINAPI ドライバが有効になるのは `-dWINKANJI` オプションが使用された場合だけであり、その他の場合はオリジナルの gnu Ghostscript 7.07 そのものとして動作します。よって、gnu Ghostscript 7.07 の良いところを殺すことなく、機能を付加できていると期待できます。GSview において、Options → Advanced Configure... → Ghostscript Options エディットボックスに `-dWINKANJI` と記述しておく、GSview で通常の PostScript ファイルを扱う場合、WINAPI ドライバを使用するようになります。このように設定しておいて下さい。

(02) [gs864w32full-gpl.zip](#)

GPL Ghostscript 8.64 の Windows 用バイナリのフルセットです。オリジナルのバイナリがあるにもかかわらず、配布する理由は

1. `-dWINKANJI` オプションをつけると、OS が認識している日本語 OpenType フォントや、日本語 TrueType フォントを WIN32API によって使用することができるよう拡張していること。
2. gsftopk に関して、Paul Vojta さんのパッチをあてていること。

です。インストールするには、`gs864w32full-gpl.zip` を適当な作業ディレクトリで展開してから、そのディレクトリに存在する `setupgs.exe` なるインストール専用プログラムを実行します。その際、環境変数 `TEMP` の指すディレクトリ名に日本語が含まれていると、ほぼ 100% インストールに失敗しますから、環境変数 `TEMP` が `c:\temp` など日本語を含まない名前のディレクトリを指すように定義しなおしておいて下さい。また、`setupgs.exe` を実行して表示される画面で、

Use Windows TrueType fonts for Chinese, Japanese and Korean

をチェックしておくのを忘れないようにして下さい。これによって Windows のフォントディレクトリに存在する CJK フォントを探して、`cidfmap` なるファイルに適切なエントリを作成します。なお、GSview の Ghostscript options 編集ボックスには `-dWINKANJI` なるオプションを追加しておいて下さい。これを追加しなくても、オリジナルの機能で日本語を表示することができますが、`-dWINKANJI` を設定しない場合は、特に縦書きにおいて信頼性が落ちるようです。

6 その他のファイル

(01) **[textrace-w32.zip](#)**

Szabo Peter <pts@fazekas.hu> による T_EXtrace を Windows 上で実行できるようにしたものです。T_EXtrace は METAFONT ソース、より一般的には PK フォントから Type1 フォントを生成するツールです。これはどこかのディレクトリにインストールするのではなくて、適当なディレクトリに展開して、そのディレクトリで作業するように作られています。詳細な使用方法に関しては文書 README-W32.txt を読んで下さい。汎用に使用できますが、特に lambda_j で使用する msmin??、msgoth?? フォントの type1 バージョンを生成するバッチファイルを含めています。perl と Ghostscript が必要です。

(02) **[gtftex.lzh](#)**

これは本田知亮さん、および稲垣淳さんとの共同によるもので、日本学術振興会未来開拓学術研究推進事業「マルチメディア通信システムにおける多国語処理の研究」プロジェクトで収集された、69263(このうち、重複のため廃止された番号がいくつかあります)個に及ぶ漢字を、株式会社アスキーによる pL_AT_EX で使用するためのパッケージです。プリアンブルで `\usepackage{gtftex}` としておくと、GT フォント用コマンドが使用できるようになります。なお、奥村さんによる新ドキュメントクラス jsarticle にも対応しています。

使用できるコマンド書式は三つあり、例えば

```
\GT{000205}
\GT{GT2000-02 8956}
\GT{GT2000-02 鰻}
```

は全て同じ文字“𪛗”を表しています。最初のものは `\GT{gtfontnumber}` のように使い、gtfontnumber は 000001 から 069263 までの十進数で、GT フォント整理番号と呼ばれる番号を入力します。整理番号と実際のフォントの対応をつけるために virtual fonts を使用しています。二番目のものは `\GT{フォント名 ShiftJIScode}` のようにします。フォント名には GT2000-01, GT2000-02, ..., GT2000-11, および GT2000-k1, GT2000-k2 を指定します。ShiftJIS コードは大文字の 16 進数で指定します。三番目のものは ShiftJIS コードに対応する、通常の encoding における漢字を入力する方法です。この方法は代理文字形式とよばれます。詳細に関しては、インストールしたあと、texmf/ptex/platex/gtftex にある *.dtx ファイルを platex でタイプセットしてお読み下さい。

7 ファイル sh.exe について

Korn shell のサブセット sh.exe を texmf/doc/ksh のなかに入れております。sh.exe は Windows NT/2000/XP では shell として使用できますが、Windows 98 等では使用できません。従って bin ディレクトリではなくて、documents 用ディレクトリに入れてあります。sh.exe は起動時に copyright notice を表示すればバイナリ配布が許されているものです。sh.txt を参照してください。勿論これを快適に shell として使うには GNU utilities が不可欠ですが、相性のよいものは ftp://microlib.cc.utexas.edu/microlib/nt/gnu/gnubin.tar.Z です。配布する sh.exe は Win32 CUI としてコンパイルし直したもので、コマンドラインの長さ制限をオリジナルの 1024 から、24576 に大きくしてあります。初期化ファイルのフルパスを ENV 環境変数で例えば ENV=c:/users/default/.profile のように指定しておくと、便利な shell として使用することもできます。ドライブ名は普通のように使用できます。ディレクトリセパレータは UNIX と同じで、/ ですが、パスセパレータはコロン:ではなくてセミコロン;です。セミコロンは shell の特殊文字でもあるので、初期化ファイル .profile で PATH を設定するときは、引用符が必要です。

```
例： PATH="c:/winnt/system32;c:/w32tex/bin;."
```

Korn shell の殆どの機能は実現してあるようで、非常に便利で強力な shell です。cmd.exe 用の foo.bat, foo.cmd, foo.vbs, foo.js ファイルを実行するために、.profile で COMSPEC=cmd.exe としておいてください。PATH 上に extension なし、.sh, .ksh, .sub, .pl, .lg の拡張子をもつもの(例 foo, foo.sh, foo.ksh, foo.sub, foo.pl, foo.lg)があると、script として実行できません(foo とするだけ)。最初の行を

```
#! c:/bin/sh   とか   #! c:/w32tex/bin/perl
```

で始めておけば、望みの処理をしてくれます。Korn shell については市販の書籍を参考にしてください。sh.exe を標準の shell として使用しない方は 7 に記した面倒なことは気にする必要はありません。最後に .profile の例を付けておきます。

```
# -----
# example of .profile
PATH="c:/bin;c:/w32tex/bin;c:/winnt/system32;c:/winnt;"
PATH=${PATH}"c:/usr/bin;."
#
alias which='whence -p' # 'which x' will tell you if executable x
                        # exists and if so give its full path
alias sl='ls -F'        # short list
alias l='ls -alF'      # long list
alias del='rm -i'      # interactive remove
alias h=history         # list history
alias r='fc -e -'      # run history number
alias rd=rmdir         # remove directory
alias md=mkdir         # make directory
alias cd=chd           # change directory and print cwd
alias b='cmd /c'       # for invoking cmd
alias clear='cmd /c cls'
PS1="(hello)# "
#
TERM='pcbios' # may be needed for Congruent's version of vi
SHELL=c:/bin/sh
TEMP=c:/temp
TMP=$TEMP
TMPDIR=$TEMP
HISTSIZ=300
SH_PATH=c:/bin/sh.exe
export PATH TERM TEMP TMP TMPDIR SHELL HISTSIZ SH_PATH
#
COMSPEC=cmd.exe
TZ=JST-9
export COMSPEC TZ
# function chd
chd ()
{
    builtin cd "${1+"$@"} && echo $PWD
}
# return to the previous directory
rpd ()
```

```
{  
chd $OLDPWD  
}  
set -o emacs  
set -o ignoreeof  
# end of example
```

Akira Kakuto < kakuto@fuk.kindai.ac.jp >