

# Windows 上の日本語 TeX 環境について

阿部 紀行

東京大学大学院数理科学研究科

2010 年 3 月 23 日

# TeX

- Knuth 氏による組版ソフト .
- 数式に強い .
- 数学者には欠かせない .
- 古いソフト ( Ver 0 の公開が 1982 年.....ほとんど同じ年です ) .
- 古さ故な問題もある.....気がする .

## Windows 上の日本語 T<sub>E</sub>X

日本語の T<sub>E</sub>X はアスキー社による pT<sub>E</sub>X が主流．Windows 上のバイナリで動くのは以下の二つ．

W32TeX (角藤氏) 近畿大の角藤氏によりメンテナンスされている．幅広いパッケージをフォロー．一番よく使われている．

Cygwin 上でうごくもの (黒木氏) Cygwin (Unix ツールの Windows への移植の集合体) 上で pTeXlive をコンパイルしたもの．実行に Cygwin が必要．

日本人が Windows で T<sub>E</sub>X を使っていますと言えば，ほぼ間違いなく角藤氏の W32TeX

## W32TeX のインストール

- たくさんのファイルをダウンロード .
- コマンドプロンプトで作業 .
- 環境変数も追加しなきゃ .
- W32TeX 以外にも dviout , GhostScript , GSView..... .
- .....

めんどくさっ！

と思ったので，楽しようと思いました．

## TeX インストーラ 3

- W32TeX をダウンロード 解凍 PATH の設定を自動化 .
- dviout , GhostScript , GSView のインストールもつけた .
- ウィザードを利用 「次へ」を押していけばインストールできる .
- インストールファイルの情報をインターネットから取得 各ソフトのアップデートに自動的に対応 (こちらがさぼれる！)
- 「よくはまる問題」への対処も入れたつもり (うまく動いているかはあまり知りませんが.....)
- ほかのソフトも , プラグインでインストール可能 (DLL : Ver 0.02 , Ruby : Ver 0.50 から .)

ところで.....

TeX インストーラ 3 って.....

1 と 2 もありました .

## インストーラの動き

- (1) インストール先，接続先，proxy などを設定．
- (2) サーバに接続し，ファイル情報（更新時刻，サイズ等）を取得．
- (3) 取得した情報をもとに，一覧表示．手元にあるファイルの更新時刻と比較し，新しいか古いかも表示．
- (4) ファイルをダウンロード．
- (5) W32TeX のファイルを解凍．
- (6) GhostScript と GSView のインストーラを起動．
- (7) dviout のファイルを解凍．レジストリを設定．
- (8) プラグインによるインストールを実行．



## ファイルの情報取得

- 付属インストーラ `texinst***.exe` のソースファイルから W32TeX ファイルの一覧を取得 .
- フルインストール以外はできません .
- ダウンロード先からファイル情報を取得 ( HTTP : index から , FTP : ls して .)
- W32TeX 以外は , ファイル一覧から 「それっぽいファイル」を指定 . ( 「それっぽさ」は正規表現で与える .)

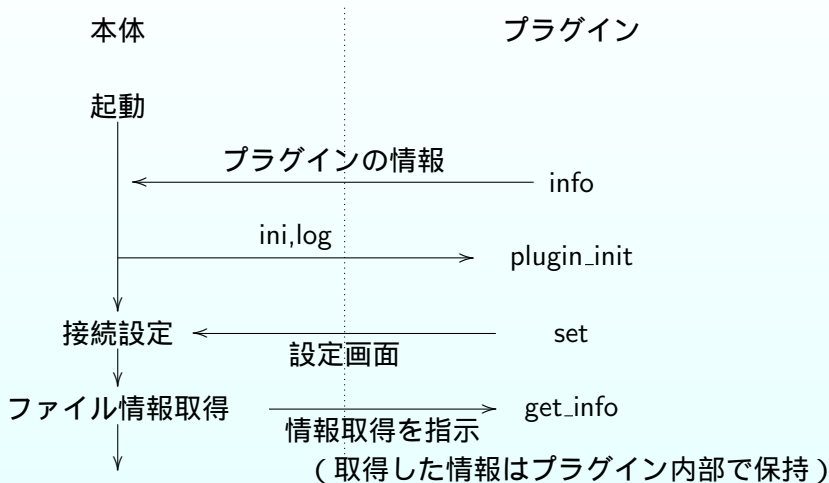
# アップグレード

- W32TeX は頻繁にアップグレードされる .
- 手元のファイルの更新時刻とインターネットにあるファイルの更新時刻を比較し , 手元のファイルが三日以上古ければ新しいものが出たと判断 .
- インストール ( 解凍 ) の前にアンインストールをする .
- W32TeX は解凍したファイル一覧をとっておいて , そこにあるファイルを削除 .
- 各種ファイルはアンインストーラを実行 .
- 兼宗氏が昔の W32TeX を公開してくれているので , あのころにも戻れる .

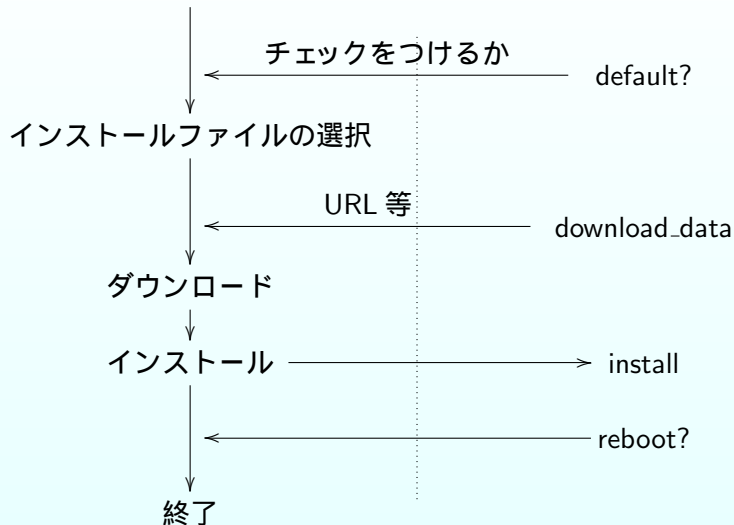
# プラグイン

- 「標準的でない」ソフトのインストールのために，プラグインが使える．
- 新ドキュメントクラス（現在は W32TeX にもある），Xy-pic，ispell，WinShell を公開中．
- Ruby で作れるので，比較的簡単に作れる（と思う）．
- プラグインの作り方を説明してみようと思います．

# 処理の流れ



## 処理の流れ（続き）



## 実際のプラグイン

ひな形は次のようになります。

### Sample.rb

```
class SampleCls
  def initialize
    初期化
  end

  各種関数
end

Sample = SampleCls.new
```

以下，<http://hoge hoge.net/soft/soft.zip> をダウンロードして解凍するプラグインを作ってみます。

## 実装すべき関数

最近の開発版から一部変更されています。

- `plugin_init` : 初期化 .
- `info` : インストールするソフトの情報を返す .
- `set?` : 設定項目があるか ?
- `set` : 設定画面を表示する .
- `get_info` : ファイルの情報を取得する .
- `download_data` : `GetInfo` で取得した情報をもとに , URL やファイルサイズなどを返す .
- `default?` : デフォルトでインストールするか ( `true` を返すとチェックがつく ) .
- `install` : インストールを行う .
- `reboot?` : 再起動が必要かどうか .

## SampleCls の初期化

```
def initialize
  @size = 0
  @success = false
end
```

変数の初期化です。@size にはファイルサイズを，@success には情報取得ができたかを入れる予定です。



## plugin\_init

```
def plugin_init(ini, log)
  @ini = ini # 本体の設定ファイル
  @log = log # ログの出力先
end
```

@ini に本体の設定ファイル（デフォルトでは abtexinst.ini ），@log にログの出力先（デフォルトでは abtexinst\_log.txt ）を格納します  
以下の処理は，そのログを@log にはき出すべきですが，以下の例では省略します．

## info

```
def info
  return [ # 配列
    {"softname" => "sample",
     "info" => "サンプルです"}
  ]
end
```

インストールするソフトウェアの情報を返します。連想配列で、softname がソフトウェアの名前、info が情報を返します。配列で返します。一つのプラグインで複数のソフトウェアをインストールすることもできます。複数のソフトウェアに対応している場合は、各々のソフトウェアを"softname"で識別します。

set?

```
def set?(soft)
  return false
end
```

設定画面は出しません .

## set

```
def set (soft ,hwnd)  
    return  
end
```

設定画面は出しません .

## get\_info

```
def get_info(hwnd)
  # 本体のダイアログに情報表示
  AbTeXInst::add_message(hwnd, "情報取得中\r\n")
  Net::HTTP.start('hoge.hoge.net', 80) { |http|
    res = http.head("/soft/soft.zip")
  }
  if res == nil then # 接続に失敗
    AbTeXInst::add_message(hwnd, "失敗\r\n")
    return false
  end
  AbTeXInst::add_message(hwnd, "成功\r\n")
  @size = res['Content-Length'].to_i # サイズ取得
  @success = true
  return true
end
```

## download\_data

```
def download_data(soft)
  return {
    # get_infoが成功したかどうか
    "success" => @success,
    # ダウンロードファイルの情報
    "download_files" => [{
      "url" => "http://hogegege.net/soft/soft.zip",
      # 保存ファイル名
      "file" => "soft.zip",
      "size" => @size }]]}
end
```

サイズはおおよそで問題ありません。

## default?

```
def default?(soft)
  return true
end
```

とりあえず常にインストールするようにします (本当はファイルの更新時刻を比較し, 更新されているならば true を返すとよいでしょう.)

# install

```
def install(hwnd, soft, dlresult)
  dlr = dlresult["http://hogehe.net/soft/soft.zip"];
  if dlr == nil or dlr != AbTeXinst::SUCCESS then
    return false
  end
  dir = AbTeXinst::Setting::install_dir + "share\\texmf"
  file = AbTeXinst::Setting::hizon_dir + "soft.zip"
  AbTeXinst::add_message(hwnd, "インストール中.....")
  # soft.zipをtexmfツリーに展開する
  AbTeXinst::zipdlg(hwnd, file, dir)
  AbTeXinst::add_message(hwnd, "終了\r\n")
  return true
end
```



## reboot?

```
def reboot?(soft)
  return false
end
```

ほとんどの場合再起動をする必要はないと思います。

## 作ったものの問題点

そのままだと実用に耐えません．

- エラー処理がほぼゼロ．
- ログを一切はき出さない．
- 本体の設定にかかわらずネットワークに接続する．
- proxy 設定も無視．

これらの対策を施すと，だいたい Jsclasses プラグインになります．

## 今後

- とりあえずのインストーラとしては完成したと思っています。
- インストール CD を作るのにも使えるように開発版でしてみました  
が、いくつか違和感があります。
  - CD からインストールするのにも関わらず、ネットワークの設定画面が表示される。
  - 「アップデート」機能がうまく動かない（解凍したファイル一覧を保持できないため。）
  - そもそも統一するものじゃないかもしれない。
- プラグイン
  - そもそも仕様がやっつけ？
  - 何か標準的な仕様があればそれにのっかるけど（ネットインストーラの動きも含めて。）
- そもそも何かツールキットを活用して作った方がよかったんじゃないか：もう遅い？
- なかなか更新できないでいますが、ぼちぼちやりますのでよろしく  
お願いします。