**Building**

<https://github.com/coolwanglu/pdf2htmlEX/wiki/Building>

Fontforge 설치

<http://fontforge.org/>

<https://github.com/fontforge/fontforge>

cygwin에서 설치 :   
<http://www.useragentman.com/blog/2011/02/27/installing-cygwin-and-fontforge-for-windows/>   
<http://darkcheat.tistory.com/16>

window 설치파일 : <http://www.mpetroff.net/software/fontforge-windows/>

git clone git://github.com/fontforge/fontforge.git

cd fontforge

README-MS.html 및 INSTALL 파일 내용을 참고하여 설치를 진행한다.

<http://darkcheat.tistory.com/16>

<http://fontforge.org/ms-install.html>

<http://sourceforge.net/projects/fontforge/files/fontforge-executables/>

bunzip2 fontforge\_cygwin-\*.tar.bz2

$ tar xf fontforge\_cygwin-\*.tar

$ cd fontforge

$ ./doinstall

## [윈도우즈에서 FontForge 20080203 버전 설치 및 실행하기](http://darkcheat.tistory.com/16)

카테고리: [컴퓨터](http://darkcheat.tistory.com/category/컴퓨터)

작성일: 2012/11/03 04:21

폰트포지(FontForge)는 프리웨어 폰트 에디터 중에서 가장 좋으며 모든 운영체제에서 실행 가능하다. 헌데 FontForge의 20080203 이후 버전에서 2바이트를 사용하는 한글폰를 가지고 Generate Font 해버리면 죽어버리는 현상이 발생한다.

최근버전의 FontForge는 윈도우즈에서 실행 가능하도록 셋팅이되어 배포되고 있지만 20080203 버전은 그렇지 않아서 직접 윈도우즈에서 돌아가도록 설치해야하는데, 여기서 그 방법을 설명한다.

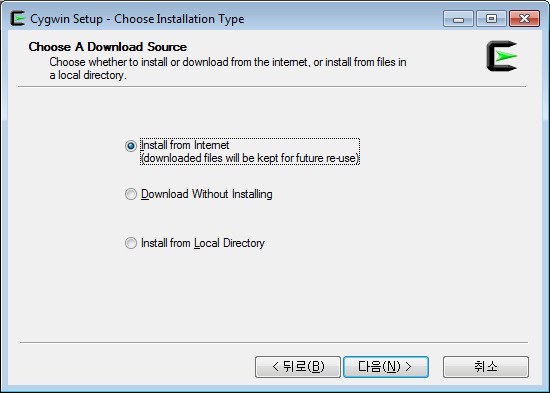
FontForge는 UNIX X-Windows 프로그램이기 때문에 Windows에서 실행시키려면 UNIX 명령어 쉘 유사환경을 만들어주는 Cygwin이 필요하다. 그래서 우선 Cygwin부터 설치하도록 해보자.

## 1. Cygwin 설치하기

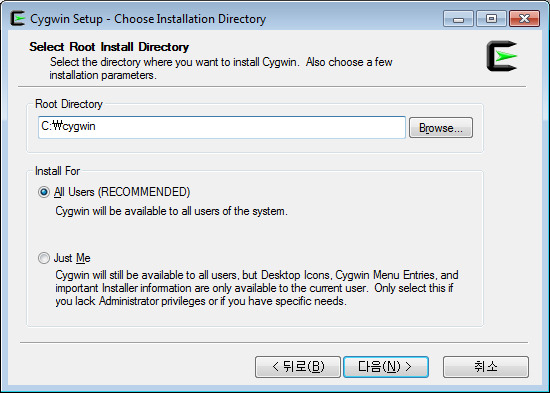
1. Cygwin 홈페이지(<http://cygwin.com/>)에 들어가서 [setup.exe](http://cygwin.com/setup.exe) 링크를 클릭하여 다운받는다.



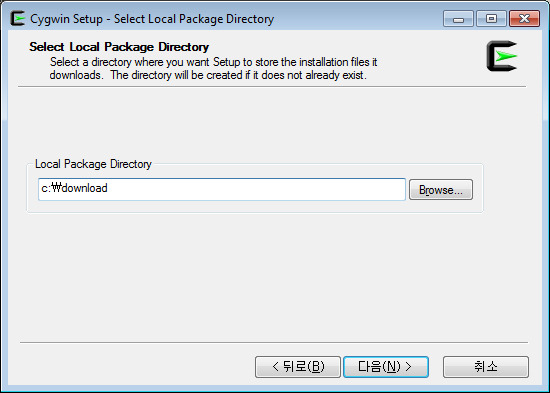
1. setup.exe를 실행시키면 위와 같은 화면이 뜨는데, '다음' 버튼을 눌러 다음 화면으로 넘어가자.



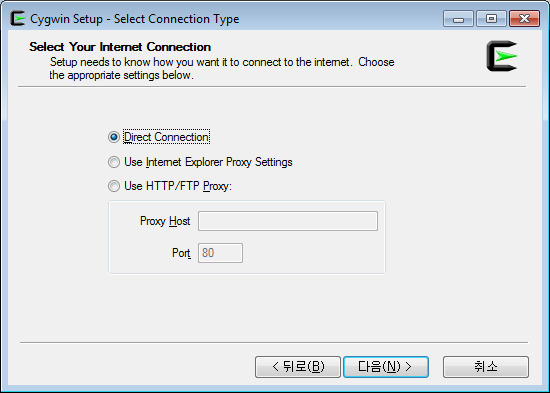
1. 'Install from Internet'를 선택한고 다음 화면으로 넘어간다.



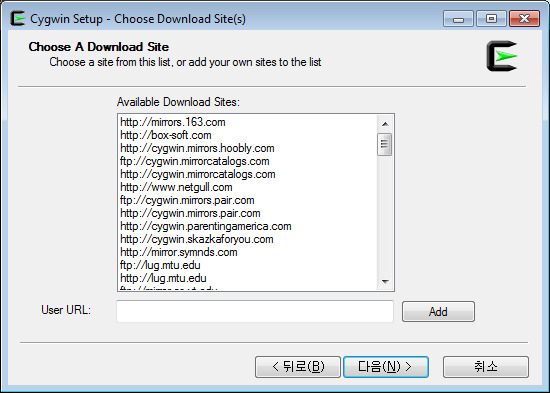
1. 설치 경로는 디폴트 경로인 c:\cygwin에다가 설치하는 것이 좋다. 경로 지정하는 부분의 아래쪽 'Install for'도 관리자 권한을 가지고 있는 이상 디폴트값 그대로 둔후 다음 화면으로 넘어간다.



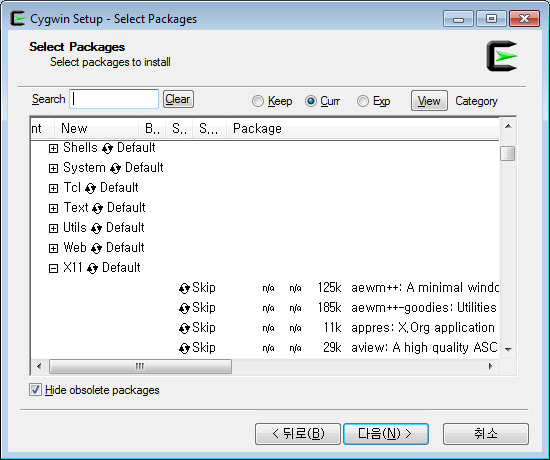
1. 패키지를 받을 디렉토리를 설정해준다. 패키지는 Cygwin 프로그램의 부품이라고 생각하면 된다. 나중에 패키지를 다 받고나면 이 패키지들을 압축 해제하여 4번에서 설정해준 경로에 설치하는 것이다.



1. 인터넷 연결을 설정하는 부분인데, 특별한 경우가 아니라면 'Direct Connection'를 선택해 준다.

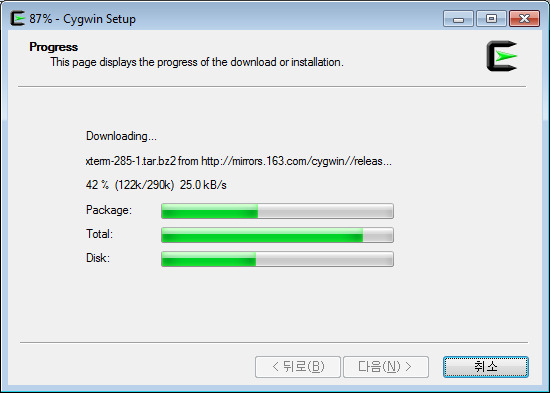


1. 다운로드 받을 사이트를 선택해 준후 다음화면으로 넘어간다.

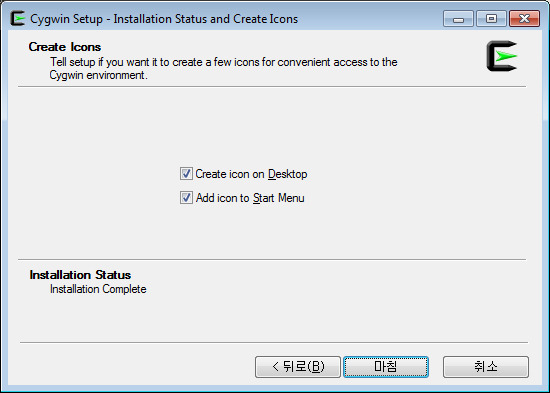


1. 다운로드하고 설치 해야할 패키지를 선택하는 부분인데, 아래에 나열된 패키지명을 선택해주면 된다. 설치 화면에서 Skip 부분을 클릭하여 버전으로 바뀌게되면 선택되어진 것이다. 그리고 왼쪽 위에 Search 입력란 부분에 찾으려는 패키지명을 입력하면 훨씬 빠르게 찾을 수 있다.
   * **X11** 카테고리 아래에:
     + X-start-menu-icons
     + xinit
     + xorg-docs
     + xorg-server
     + libcairo2
     + libXi6
     + libXi-devel
   * **Gnome** 카테고리 아래에:
     + libxml2
     + libpango1.0\_0
   * **Graphics** 카테고리 아래에:
     + giflib
     + libpng
     + jpeg
     + libtiff (현재 시점에서 libtiff의 최신버전은 libtiff5이다)
   * **Editors** 카테고리 아래에:
     + vim
   * **Web** 카테고리 아래에:
     + wget
   * **Shells** 카테고리 아래에:
     + rxvt (rxvt-unicode-\*가 아닌 rxvt 임을 주의)

패키지의 버전이 따라 패키지명이나 부모 카테고리가 조금씩 달라질 수 있는데, 이때는 설치해야할 패키지명과 비슷한 것을 찾아서 선택해주면 된다. 모두 선택이 완료되었으면 다음 화면으로 넘어간다.



1. 그러면 위와 같이 다운로드가 시작되고 다운로드가 완료되면 자동으로 설치도 진행된다. 만약 이 화면이 뜨지 않고 "Connecting" 이라는 메시지만 나타난다면 7번 화면으로 돌아가서 다른 사이트를 선택해보자.



1. 정상적으로 완료가 되었으면 위와 같은 화면이 뜨게 된다. 바탕화면과 시작메뉴에 바로가기 아이콘 추가를 체크해주고 '완료' 버튼을 눌러서 종료한다.

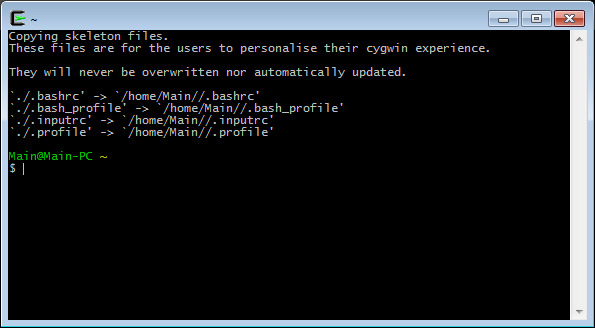
## 2. FontForge 설치 및 실행하기

FontForge 중에서 Cygwin에서 구동되도록 이미 컴파일된 버전이 있기 때문에, 우리가 컴파일까지 해야한다는 걱정을 할 필요 없다. [SourceForge](http://sourceforge.net/projects/fontforge/files/) 사이트의 fontforge-executables 폴더에 가보면 FontForge 이름뒤에 \_cygwin이라고 붙어있는 파일이 있는데, 이것이 Cygwin에서 돌아가도록 컴파일된 버전이다. 아래에는 미리 컴파일된 20080203 버전의 FontForge를 첨부해 놓았다.

[[http://i1.daumcdn.net/cfs.tistory/v/110706133414/blog/image/extension/unknown.gif](http://darkcheat.tistory.com/attachment/cfile22.uf@18085A35509417D7270E97.bz2)fontforge\_cygwin-20080203.tar.bz2](http://darkcheat.tistory.com/attachment/cfile22.uf@18085A35509417D7270E97.bz2)

fontforge\_cygwin-20080203.tar.bz2 파일을 받았으면 이것을 C:\cygwin\home\<유저명> (유저명은 윈도우즈에 로그인 할 때 사용된 계정 이름이다)에다 옮겨준후 다음 절차를 따라가자.

1. 바탕화면에 Cygwin Terminal 아이콘을 클릭하여 이전에 설치한 Cygwin을 실행시키자.

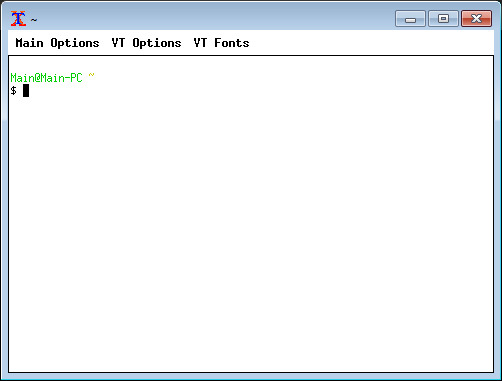


1. Cygwin이 실행되면 위와 같이 검은 창이 뜨고 초록색으로 된 자신의 계정이 보일 것이다. $ 옆에 캐럿이 깜빡일 것인데 여기다가 아래의 명령어를 차례대로 입력해주자. (아래의 코드에서 $ 는 프롬프트를 흉내기기 위한 것일 뿐 명령어의 일부분이 아니므로 타이핑 하면 안된다.)

$ bunzip2 fontforge\_cygwin-20080203.tar.bz2  
$ tar xf fontforge\_cygwin-20080203.tar  
$ cd fontforge  
$ ./doinstall

1. 그리고 아래의 명령어를 입력하여 xterm 창을 띄우자.

$ startxwin



1. 위의 그림이 xterm 창이 띄워진 상태다. 여기에 다음 명령어를 입력하면 FontForge가 실행된다.

$ fontforge

FontForge를 실행시킬 때 'Open Font'창을 띄우기 싫다면 다음과 같이 명령어 뒤에 -new를 붙여주면 된다.

$ fontforge -new

fontforge 명령어를 쳤는데 FontForge가 뜨지않고 cygpng12.dll이 없다는 에러 메시지가 뜬다면 cygpng12.dll 파일을 다운받아 C:\cygwin\bin 폴더에 복사해주면 된다.

[[http://i1.daumcdn.net/cfs.tistory/v/0/blog/image/extension/dll.gif](http://darkcheat.tistory.com/attachment/cfile7.uf@16075335509417D628DEE7.dll)cygpng12.dll](http://darkcheat.tistory.com/attachment/cfile7.uf@16075335509417D628DEE7.dll)

FontForge가 제대로 실행된다면 아무 한글 폰트를 열어서 Generate Font를 해보자. 이후 버전의 FontForge와 달리 죽어버리 않는다는 것을 알 수 있다.

* **poppler** >= 0.20.0 with xpdf headers (compile with **--enable-xpdf-headers**)
  + Install **libpng** (and headers) BEFORE you compile poppler if you want background images generated
  + Install **poppler-data** if your want CJK support

<http://blog.alivate.com.au/poppler-windows/>

* [Optional] **ttfautohint**
  + run pdf2htmlEX with **--external-hint-tool=ttfautohint** to enable it

<http://www.freetype.org/ttfautohint>

NAME

pdf2htmlEX - converts PDF to HTML without losing text and format.

USAGE

pdf2htmlEX [options] <input-filename> [<output-filename>]

DESCRIPTION

pdf2htmlEX is a utility that converts PDF files to HTML files.

pdf2htmlEX tries its best to render the PDF precisely, maintain proper styling, while retaining text and optimizing for Web.

Fonts are extracted form PDF and then embedded into HTML (Type 3 fonts are not supported). Text in the converted HTML file is usually selectable and

copyable.

Other objects are rendered as images and also embedded.

OPTIONS

Pages

-f, --first-page <num> (Default: 1)

Specify the first page to process

-l, --last-page <num> (Default: last page)

Specify the last page to process

Dimensions

--zoom <ratio>, --fit-width <width>, --fit-height <height>

--zoom specifies the zoom factor directly; --fit-width/height specifies the maximum width/height of a page, the values are in pixels.

If multiple values are specified, the minimum one will be used.

If none is specified, pages will be rendered as 72DPI.

--use-cropbox <0|1> (Default: 1)

Use CropBox instead of MediaBox for output.

--hdpi <dpi>, --vdpi <dpi> (Default: 144)

Specify the horizontal and vertical DPI for images

Output

--embed <string>

--embed-css <0|1> (Default: 1)

--embed-font <0|1> (Default: 1)

--embed-image <0|1> (Default: 1)

--embed-javascript <0|1> (Default: 1)

--embed-outline <0|1> (Default: 1)

Specify which elements should be embedded into the output HTML file.

If switched off, separated files will be generated along with the HTML file for the corresponding elements.

--embed accepts a string as argument. Each letter of the string must be one of `cCfFiIjJoO`, which corresponds to one of the --embed-\*\*\*

switches. Lower case letters for 0 and upper case letters for 1. For example, `--embed cFIJo` means to embed everything but CSS files and out‐

lines.

--split-pages <0|1> (Default: 0)

If turned on, the content of each page is stored in a separated file.

--page-filename may be used to specify the format for the filenames for individual pages. A %d placeholder may be included to indicate where

the page number should be placed. The placeholder supports a limited subset of normal numerical placeholders, including specified width and

zero padding.

If --page-filename does not contain a placeholder for the page number, the page number will be inserted directly before the file extension. If

the filename does not have an extension, the page number will be placed at the end of the file name.

If --page-filename is not specified, <input-filename> will be used for the output filename, replacing the extension with .page and adding the

page number directly before the extension.

This switch is useful if you want pages to be loaded separately & dynamically -- a supporting server might be necessary.

Examples

pdf2htmlEX --split-pages 1 foo.pdf

Yields page files foo1.page, foo2.page, etc.

pdf2htmlEX --split-pages 1 foo.pdf bar.baz

Yields page files bar1.baz, bar2.baz, etc.

pdf2htmlEX --split-pages 1 foo.pdf page%dbar.baz

Yields page files page1bar.baz, page2bar.baz, etc.

pdf2htmlEX --split-pages 1 foo.pdf bar%03d.baz

Yields page files bar001.baz, bar002.baz, etc.

--dest-dir <dir> (Default: .)

Specify destination folder.

--css-filename <filename> (Default: <none>)

Specify the filename of the generated css file, if not embedded.

If it's empty, the file name will be determined automatically.

--page-filename <filename> (Default: <none>)

Specify the filename template for pages. This is only useful when --split-pages is 1

If it's empty, a default one will be used, see description of --split-pages

--outline-filename <filename> (Default: <none>)

Specify the filename of the generated outline file, if not embedded.

If it's empty, the file name will be determined automatically.

--process-nontext <0|1> (Default: 1)

Whether to process non-text objects (as images)

--process-outline <0|1> (Default: 1)

Whether to show outline in the generated HTML

--printing <0|1> (Default: 1)

Enable printing support. Disabling this option may reduce the size of CSS.

--fallback <0|1> (Deafult: 0)

Output in fallback mode, for better accuracy and browser compatibility, but the size becomes larger.

Fonts

--embed-external-font <0|1> (Default: 1)

Specify whether the local matched fonts, for fonts not embedded in PDF, should be embedded into HTML.

If this switch is off, only font names are exported such that web browsers may try to find proper fonts themselves, and that might cause issues

about incorrect font metrics.

--font-suffix <suffix> (Default: .ttf)

Specify the suffix of fonts extracted from the PDF file.

--decompose-ligature <0|1> (Default: 0)

Decompose ligatures. For example 'fi' -> 'f''i'.

--auto-hint <0|1> (Default: 0)

If set to 1, hints will be generated for the fonts using fontforge.

This may be preceded by --external-hint-tool.

--external-hint-tool <tool> (Default: <none>)

If specified, the tool will be called in order to enhanced hinting for fonts, this will precede --auto-hint.

The tool will be called as '<tool> <in.suffix> <out.suffix>', where suffix will be the same as specified for --font-suffix.

--stretch-narrow-glyph <0|1> (Default: 0)

If set to 1, glyphs narrower than described in PDF will be stretched; otherwise space will be padded to the right of the glyphs

--squeeze-wide-glyph <0|1> (Default: 1)

If set to 1, glyphs wider than described in PDF will be squeezed; otherwise it will be truncated.

Text

--heps <len>, --veps <len> (Default: 1)

Specify the maximum tolerable horizontal/vertical offset (in pixels).

pdf2htmlEX would try to optimize the generated HTML file moving Text within this distance.

--space-threshold <ratio> (Default: 0.125)

pdf2htmlEX would insert a whitespace character ' ' if the distance between two consecutive letters in the same line is wider than ratio \*

font\_size.

--font-size-multiplier <ratio> (Default: 4.0)

Many web browsers limit the minimum font size, and many would round the given font size, which results in incorrect rendering.

Specify a ratio greater than 1 would resolve this issue, however it might freeze some browsers.

For some versions of Firefox, however, there will be a problem when the font size is too large, in which case a smaller value should be speci‐

fied here.

--space-as-offset <0|1> (Default: 0)

If set to 1, space characters will be treated as offsets, which allows a better optimization.

For PDF files with bad encodings, turning on this option may cause losing characters.

--tounicode <-1|0|1> (Default: 0)

A ToUnicode map may be provided for each font in PDF which indicates the 'meaning' of the characters. However often there is better "ToUnicode"

info in Type 0/1 fonts, and sometimes the ToUnicode map provided is wrong. If this value is set to 1, the ToUnicode Map is always applied, if

provided in PDF, and characters may not render correctly in HTML if there are collisions.

If set to -1, a customized map is used such that rendering will be correct in HTML (visually the same), but you may not get correct characters

by select & copy & paste.

If set to 0, pdf2htmlEX would try its best to balance the two methods above.

--optimize-text <0|1> (Deafult: 0)

If set to 1, pdf2htmlEX will try to reduce the number of HTML elements used for text. Turn it off if anything goes wrong.

PDF Protection

-o, --owner-password <password>

Specify owner password

-u, --user-password <password>

Specify user password

--no-drm <0|1> (Default: 0)

Override document DRM settings

Misc.

--clean-tmp <0|1> (Default: 1)

If switched off, intermediate files won't be cleaned in the end.

--data-dir <dir> (Default: /usr/share/pdf2htmlEX)

Specify the folder holding the manifest and other files (see below for the manifest file)`

--css-draw <0|1> (Default: 0)

Experimental and unsupported CSS drawing

--debug <0|1> (Default: 0)

Print debug information.

Meta

-v, --version

Print copyright and version info

--help Print usage information

MANIFEST and DATA-DIR

When split-pages is 0, the manifest file describes how the final html page should be generated.

By default, pdf2htmlEX will use the manifest in the default data-dir (run `pdf2htmlEX -v` to check), which gives a simple demo of its syntax.

You can modify the default one, or you can create a new one and specify the correct data-dir in the command line.

When single-html is 1, all files referred by the manifest must be located in the data-dir.

EXAMPLE

pdf2htmlEX /path/to/file.pdf

Convert file.pdf into file.html

pdf2htmlEX --clean-tmp 0 --debug 1 /path/to/file.pdf

Convert file.pdf and leave all intermediate files.

pdf2htmlEX --dest-dir out --single-html 0 /path/to/file.pdf

Convert file.pdf into out/file.html and leave font/image files separated.