

# 초간단 oblivoir v2.0 사용법

x-ob-liv-oir

2013년 10월

## 요 약

oblivoir 클래스 2.0 버전은 그 동안 별도의 브랜치로 개발되어 오던 xoblivoir와 oblivoir를 통합하여 완전히 동일한 클래스가 되었다. 이 문서는 oblivoir 즉 xoblivoir를 사용하는 방법을 간략히 기술한다.

## 차 례

제 1 절	oblivoir와 xoblivoir . . . . .	3
제 2 절	oblivoir와 memhangul . . . . .	4
제 3 절	한글 드라이버 . . . . .	4
제 4 절	옵션들 . . . . .	4
4.1	memoir 옵션 . . . . .	4
4.2	한글 드라이버의 옵션 . . . . .	5
4.3	oblivoir의 옵션 . . . . .	5
제 5 절	폰트 설정 방식에 대한 간단한 설명 . . . . .	9
5.1	ko.TEX-utf 엔진 . . . . .	9
5.2	X <sub>Y</sub> TEX-ko, LuaTEX-ko . . . . .	9
5.3	oblivoir의 폰트 설정 명령 . . . . .	11
5.4	디폴트 치환 . . . . .	13
5.5	ExternalLocation . . . . .	15
5.6	이탤릭, 기울임 . . . . .	16
제 6 절	그밖의 사항들 . . . . .	17

---

6.1	enumerate	17
6.2	graphicx, xcolor	17
6.3	참조 인용, 자동 조사	18
6.4	방점	18
제 7 절	샘플 문서	19
제 8 절	첨언	19

## 제 1 절 oblivoir와 xoblivoir

$ko\TeX$  2.0 (2013/09/30)의 등장<sup>1)</sup>으로  $ko\TeX$  패키지군은 이전의 텍 엔진<sup>2)</sup>과 새로운 엔진들,  $pdf\TeX$ ,  $X\TeX$ ,  $\LaTeX$ 에 모두 일관성있게 대응하도록 변모하였다. 이러한 변화에 발맞추어, 레거시 텍 엔진을 위한 oblivoir와 새로운 텍 엔진(주로  $X\TeX$ )을 위한 xoblivoir로 나누어져 있던 oblivoir 클래스도 체계를 정비하여 그 구별을 없애고 동작하는 엔진에 따라 동작 방식을 자동으로 대응하도록 고쳐졌다. 그러므로, 현재 oblivoir로 작성하는 문서는 (사용자가 몇 가지 주의깊게 엔진별 동작을 지정하기만 하면) 모든 텍 엔진에서 에러 없이 컴파일되고 유사한 결과를 얻을 수 있게 되었다.<sup>3)</sup>

그 동안 oblivoir는 비교적 복잡한 길을 거쳐왔다. 대강 정리하면,

- (1)  $H\TeX$  (나중의 kotex-euc) 한글을 memoir에서 쓰기 위하여 개발된 memhangul. 이 스타일은 더이상 사용할 수 없다.
- (2) dhucs (현재의 kotex-utf) 유니코드 한글을 memoir에서 쓰기 위하여 개발된 memhangul-ucs
- (3) memhangul-ucs를 바탕으로 memoir 클래스를 통하여 문서를 만드는 fake-article
- (4) fake-article을 oblivoir로 개명
- (5)  $X\TeX$ 을 위한 xoblivoir
- (6) xoblivoir에  $\LaTeX$  지원의 추가
- (7) xoblivoir와 oblivoir를 통합

이와 같이 발전하여 온 것이고, 이제 oblivoir와 xoblivoir는 완전히 동일한 클래스가 되었다.

이 문서는 oblivoir의 고유한 옵션과 폰트 설정 방식에 대해서만 설명한다. 실제로 oblivoir를 이용하여 문서를 작성할 때는 다음 세 층위의 명령이 모두 사용가능하다.

- (1) memoir 명령
- (2) 한글 엔진( $ko\TeX$ ,  $X\TeX$ -ko,  $\LaTeX$ -ko)의 명령
- (3) oblivoir 명령

이 각각의 명령에 대한 정보를 얻으려면, memoir 매뉴얼 (`texdoc memman`), 한글 패키지 매뉴얼 (예컨대, `texdoc kotex`, `texdoc xetexko`)을 읽어야 한다. 이 문서에

1) `texdoc kotex` 명령을 내리면 kotexdoc 문서를 읽을 수 있다.

2) 이른바 “레거시 텍”이라 하는  $\TeX$ ,  $\epsilon\text{-}\TeX$ ,  $pdf\TeX$ 을 가리킨다.

3) 폰트 사용 방식의 차이로 인해 “완전히 동일한” 결과를 보증하지는 않는다.

서는 위의 두 패키지 층위의 명령군에 대해 언급할 때, 여백에 `memoir`, `kotex` 등을 표시하겠다. 그리고 엔진별로 고유한 옵션과 명령에 대해서는 해당 사항을 본문에서 밝힌다.

위의 두 층위의 문서에서 설명하지 않는 oblivoir에 대한 정보를 이 문서에서 얻을 수 있다.

## 제 2 절 oblivoir와 memhangul

memhangul은 memoir를 한글 문서 작성에 사용할 수 있게 하기 위하여 개발된 스타일 패키지이다. 원래 독립된 스타일로서 개발되고 유지되어 왔지만, 현재는 더이상 독립적인 스타일로 사용되지 않으며 oblivoir의 핵심 기능을 정의한 서브스타일로서만 유지된다. 즉 oblivoir란 memhangul을 이용하는 한글 문서작성 클래스라고 할 수 있다.

## 제 3 절 한글 드라이버

oblivoir는 현재 실행되는 텍 엔진의 종류에 따라 한글 식자를 위하여 다음과 같은 한글 패키지를 부른다. 이 한글 패키지들을 (편의상) oblivoir의 한글 드라이버라고 한다.

oblivoir 옵션	텍 엔진	한글 식자 패키지
no option	pdf $\LaTeX$	ko $\TeX$ -utf
no option	X $\LaTeX$	X $\TeX$ -ko
no option	Lua $\LaTeX$	Lua $\TeX$ -ko

## 제 4 절 옵션들

### 4.1 memoir 옵션

원칙적으로 oblivoir는 memoir의 모든 옵션을 동일한 의미로 다 받아들인다. 여기에 해당하는 것으로 다음과 같은 것이 있다.

**용지 크기** a6paper, a5paper, a4paper, a3paper, b6paper, b5paper, b4paper, b3paper, mcrownvopaper, mlargecrownvopaper, mdemyvopaper, msmallroyalvopaper, dbillpaper, statementpaper, executivepaper, letterpaper, oldpaper, legalpaper,

ledgerpaper, broadsheetpaper, pottvopaper, foolscapvopaper, crownvopaper, postvopaper, largecrownvopaper, largepostvopaper, smalldemyvopaper, demyvopaper, mediumvopaper, smallroyalvopaper, royalvopaper, superroyalvopaper, imperialvopaper.

**본문 기본 글자 크기** 9pt, 10pt, 11pt, 12pt, 14pt, 17pt, 20pt, 25pt, 30pt, 36pt, 48pt, 60pt, \*pt, extrafont sizes

**프린팅 옵션** twoside, oneside, onecolumn, twocolumn, openright, openleft, openany, final, draft, ms, showtrims

**기타 옵션** leqno, fleqn, openbib, article, oldfontcommands

이상의 옵션의 의미와 효과에 대해서는 memoir 매뉴얼을 읽어보라.

## 4.2 한글 드라이버의 옵션

한글 식자를 위하여 로드되는 패키지에는 고유한 옵션들이 있다. oblivoir는 다음과 같은 옵션을 해당 한글 드라이버에 넘겨준다.

**ko.T<sub>E</sub>X-utf** hangul, hanja, nojosa, nonfrench, finemath, strictcharcheck

**X<sub>Y</sub>TeX-ko** hangul, hanja, unfonts.<sup>4)</sup>

**LuaTeX-ko** hangul, hanja, unfonts

**cjk-ko** hangul, hanja, nojosa, usedotemph, usecjk1font. usecjk1font 옵션은 uset1font로 입력해도 받아들인다.

이 가운데 [hangul] 옵션은 사실상 지정할 필요가 없다. hangul 옵션이 주어져도 예들 들어 절 숫자의 형식 같은 것은 oblivoir 방식이 유지되기 때문이다. 그러므로 밑줄 그은 옵션만이 oblivoir에서 의미가 있다고 할 것이다.

## 4.3 oblivoir의 옵션

**한글 드라이버를 지정하는 옵션**

다음 두 옵션은 한글 드라이버를 강제로 지정하는 옵션이다. 이 옵션을 쓸 때는 오직 pdfL<sub>A</sub>T<sub>E</sub>X만을 실행한다는 의미임을 기억해두자.

4) unfonts 옵션은 oblivoir에서 조금 다르게 동작하므로 아래 해당 부분을 참고할 것.

**cjk** *cjk-ko* 패키지로 한글을 표시한다. 이 패키지는 *memhangul*의 일부 기능과 충돌할 가능성이 있으며 *memhangul*은 원칙적으로 *cjk-ko*를 지원하지 않는다. 그러나 한글 표현만을 위해서라면 이 옵션으로 문서를 작성할 수 있다.

**dhucs** *ko.TEX-utf*로 한글을 표현한다. *pdfL<sup>A</sup>T<sub>E</sub>X*에서 디폴트이며 *memhangul-ucs*는 이 패키지를 의도하고 작성된 것이다.

따라서, 다음 옵션들은 해당하는 드라이버에서만 의미를 갖는다.

**strictcharcheck** *ko.TEX-utf*. 엄격한 문자 검사.

**finemath** *ko.TEX-utf*. 한글 간격 미세 조정.

**nofinemath** *ko.TEX-utf*. *finemath* 기능을 끄.

**usedotemph** *cjk-ko*. *\dotemph* 명령 사용 가능.

**uset1font** *cjk-ko*. 라틴문자도 *nanumtype1*으로 찍음.

**interwordHWP** *ko.TEX-utf*. 단어 간격을 조금 더 넓게 벌려준다.

**interworddefault** *ko.TEX-utf*. 단어 간격을 적당히 벌려준다.

## 레거시 텍과 관련된 옵션

다음 옵션은 *pdfT<sub>E</sub>X* 엔진 또는  $\epsilon$ -*T<sub>E</sub>X* 엔진에서만 의미를 갖는다. 이 옵션이 주어지고 새로운 텍 엔진이 운영될 때는 무시된다.

**dvips** *oblivoir*로 작성된 문서를 *latex*  $\rightarrow$  *dvips*  $\rightarrow$  *ps2pdf* 순으로 컴파일하려 할 때, 즉 *pstricks*를 이용할 때 이 옵션을 주어야 한다. 이 옵션을 준 문서에 대하여 *pdfL<sup>A</sup>T<sub>E</sub>X*을 실행하면 안 된다.

**romanfixed** 로마 글자의 크기와 위치를 미세조정하는 옵션으로서 *untype1*을 쓸 때 유용하다. 현재 상황에서는 의미가 크지 않음.

## 여러 가지 옵션

다음은 *oblivoir*의 고유한 옵션들이다.

**chapter** *\chapter* 명령을 제대로 쓸 수 있게 해준다. 이 옵션이 없으면 *oblivoir*는 *\section*부터 시작하는 문서라고 간주하고 식자하지만 *\chapter*에서 에러를 내지는 않는다.

**kosection** `\section`에 대하여 “제”와 “절”을 찍어주도록 하는 옵션이다.

**amsmath** `amsmath.sty`와 `amssymb.sty`를 미리 로드해주는 옵션이다.

**adjustmath** 한글 문서의 행간격이 벌어짐으로 인해서 `display` 수식의 전후에 너무 많은 공백이 생기는 것을 막아준다.

**arabicfront** `\frontmatter` 부분의 페이지 숫자를 아라비아 숫자로 찍는다. 기본값은 로마 숫자.

**footnote** 각주 번호와 숫자를 한국식으로 식자한다.

**figtabcapt** 그림과 표에 <그림 1>과 같은 방식으로 캡션을 단다.

**gremph** 글꼴 대체 강조 방식을 쓴다. 이것이 기본값이다.

**itemph** 기울인 글꼴 강조 방식을 쓴다.

**nonfrench** `nonfrenchspacing`.

**hangulpagestyle** 본문의 페이지 스타일을 `hangul` 양식으로 한다.

**nokorean** 사실상 `memoir`와 거의 같은 상태가 되게 한다. 즉 `pdf bookmark`도 만들지 않으며 한글도 찍히지 않는다.

**pdfbookmark** `nokorean` 옵션을 주면서도 북마크는 만들도록 `hyperref`을 로드해주는 역할을 한다. `nokorean` 옵션이 주어지지 않을 때는 무의미함.

**10.5pt** 본문 활자 크기를 10.5pt로 한다.

**quotespacing** `quote`, `quotation` 환경의 줄간격을 ‘좁은 줄간격’으로 줄인다.

다음 옵션들은 특별한 상황에서 의미를 가지는 것이다.

**lyxhyper** `LyX`에서 문서를 작성할 때 `LyX`이 강제로 `hyperref`을 로드하는 기능과 `oblivior`의 `hyperref` 로드 기능이 충돌하는 것을 방지하기 위한 것이다.

**tocentry** `chapter` 옵션과 같이 쓰여서 `toc`, `lof`의 엔트리를 조정해준다.

**microtype** 이 옵션이 지정되면 `pdfTeX`과 `LuaTeX`에서 `microtype` 패키지를 불러온다. 다만 `XYTeX`에서는 문장부호 끌어내기를 위한 `xetexko-hanging` 스타일을 로드한다.

**subfigure** subfig 패키지 대신 subfigure 패키지를 쓰기 위해서 충돌이 있는 코드 하나를 수정해준다.

**manualfontspec** fontspec 패키지를 자동으로 로드하지 않고 사용자가 직접 설정하고자 할 때

**fontspec** fontspec 패키지에 넘겨줄 옵션을 지정한다.

**xcolor** xcolor 패키지에 넘겨줄 옵션을 지정한다.

**hyperref** hyperref 패키지에 넘겨줄 옵션을 지정한다.

**moreverb** moreverb 패키지를 사용하려 할 때. 약간의 충돌을 해결해준다.

**preload** \documentclass가 시작되기 전에 로드해야 할 패키지를 지정한다.

**preloadoption** preload할 때 함께 넘겨줄 옵션을 쓴다.

**faht, fawd** 특별히 pdf 사이즈를 조절할 필요가 있을 때 사용한다. 특히 flowfram 패키지를 위해서 필요하다.

**noreserveinserts**  $\epsilon$ -TeX의 reserveinserts 확장 코드를 억제한다. 일반적으로 사용할 필요 없다.

**moreroom** pdfTeX에서 용량 부족으로 에러가 발생할 때 특별히 지정한다.

다음 옵션들은 현재 큰 의미를 지니지 않는 것들이다. 대부분  $ko$ TeX의 발전과 더불어 oblivoir에서 특별히 지정할 필요가 없어졌다.

**unfonts** 은 글꼴을 사용하게 한다. 레거시 텍에서는 untype1이고 새로운 텍 엔진에서는 은글꼴 트루타입이 기본 글꼴이 된다.

**nounfonts** 은 글꼴을 사용하지 않게 한다. 이것이 디폴트이다.

**latinquote** 따옴표를 라틴 글꼴에서 찍도록 강제하는 옵션이었다. 현재는 아무런 작용도 하지 않는다.

**oldhangul** 옛한글 식자를 위한 옵션이었다. 현재는 이 옵션이 없어도 옛한글을 잘 처리한다.

**nowinname** 은글꼴을 위해서 마련된 옵션이었으나 현재는 무의미하다.

## 제 5 절 폰트 설정 방식에 대한 간단한 설명

### 5.1 $ko\TeX$ -utf 엔진

글꼴 선택 명령  $\backslash\text{SetHangulFonts}$ ,  $\backslash\text{SetHanjaFonts}$ ,  $\backslash\text{SetAdhocFonts}$ 와 더불어,  $\text{gremph}$  옵션이 주어졌을 때  $\backslash\text{SetGremphFonts}$  등을 사용한다. 이 명령의 의미와 용법에 대해서는  $ko\TeX$  사용설명서를 참고하라.

아무런 지정도 없을 경우  $\text{nanumtype1}$ 으로 식자하고, 이것은  $ko\TeX$ -utf의 디폴트 상황과 동일하다.  $\text{gremph}$ 는  $ko\TeX$ -utf에서  $\backslash\text{usepackage}\{\text{dhucs-gremph}\}$ 를 선언한 경우와 동일하게 동작하므로 별도로 이 스타일을 엮을 필요는 없다.

### 5.2 $X\TeX$ -ko, $\text{Lua}\TeX$ -ko

한글 드라이버들은 라틴 문자 폰트와 한글/한자 폰트를 분리하는 것을 기본으로 하고 있다. 한글과 라틴 문자 글꼴을 분리하지 않으려 할 경우,  $X\TeX$ -ko 명령인  $\backslash\text{disablekoreanfonts}$ 를 선언한다. 이렇게 하면 한글 글꼴은 라틴 문자 글꼴을 따라가게 된다. 즉,  $\backslash\text{setmainfont}$  등으로 선언된 글꼴이 한글과 라틴 문자에 동시에 식자된다.<sup>5)</sup>

**라틴 문자** 라틴 문자 폰트는  $\text{fontspec}$  패키지의 방식을 따른다. 설정과 사용에 대해서는  $\text{fontspec}$  패키지 문서를 참고하라. 매우 방대하고 훌륭한 문서이다.

```
 $\backslash\text{setmainfont}\{\text{<Font Name>}\}$ 
 $\backslash\text{setsansfont}\{\dots\}$ 
 $\backslash\text{setmonofont}\{\dots\}$ 
```

이 명령의 옵션으로  $[\text{Ligatures=TeX}]$ 을 주면 ‘, ---와 같은 입력이 “, —와 같이 나타난다. 대부분의 텍 소스에는 이런 이른바 텍 리거처를 사용하는 것이 일반적이므로 이 옵션을 지정하는 것이 좋다.  $X\TeX$ 에서  $[\text{Mapping=tex-text}]$  옵션을 써도 좋는데 이 옵션은  $\text{Lua}\TeX$ 에서는 무의미하므로 양쪽에 통용되는  $[\text{Ligatures=TeX}]$ 을 쓰도록 하자. 따라서 만약  $\text{TeX Gyre Termes}$ 를 라틴 문자 세리프 글꼴로 쓰려 한다면,

5) 이외에도 한글-라틴 문자 글꼴의 분리에 따르는 여러 가지 옵션 정의에 관련된  $X\TeX$ -ko 매크로가 있으므로 이에 대해서는  $X\TeX$ -ko 매뉴얼을 참고하라.

```
\setmainfont[Ligatures=TeX]{TeX Gyre Termes}
```

와 같이 지정하면 된다.

**한글과 한자** 한글 폰트는 다음과 같이 설정한다. 자세한 사항은 매뉴얼을 참고하라.

```
\setmainhangulfont[Options]{FontName}  
\setsanshangulfont[Options]{FontName}  
\setmonohangulfont[Options]{FontName}  
\setmainhanjafont[Options]{FontName}  
\setsanshanjafont[Options]{FontName}  
\setmonohanjafont[Options]{FontName}
```

xoblivoir에서도 이 방식의 폰트 정의를 할 수 있다.

한글 폰트만을 이용해서 한글/라틴 문자에 모두 식자할 필요가 있을 수 있다. 이 경우 다음과 같이 하는 것이 한 가지 방법이다.

```
\disablekoreanfonts  
\setmainfont{NanumMyeongjo}
```

또는,

```
\setmainhangulfont{NanumMyeongjo}  
\hangulmarks
```

이렇게 하면 라틴 문자와 문장부호, 숫자를 나눔명조로 식자한다.

더 세세한 지정을 위하여 그리고 alphas, nums, parens, quotes, colons, hyphens, puncts, cjksymbols가 식자될 폰트의 영역 (latin, hangul, hanja, prevfont)을 정의하는 `\xetexkofontregime` 명령도 쓸 수 있다. 이에 대해서는 X<sub>Y</sub>TeX-ko 매뉴얼을 반드시 참조하라.

**임시 폰트 교체** 문장을 작성 중에 일시적으로 폰트를 교체하기 위한 명령으로

```
\adhochangulfont
\adhochanjafont
```

이 정의되어 있다. 이 명령들은 `\setmainhangulfont` 명령과 같은 방법으로 사용한다. 또한,

```
\hangulfontspec
\hanjafontspec
```

명령도 정의되어 있으며, `\fontspec` 명령이 적용되지 않는 한글과 한자 영역에 효력을 발휘하므로 필요한 대로 쓸 수 있다.

### 5.3 oblivoir의 폰트 설정 명령

#### 간단한 폰트 지정

한편, 한글 글꼴을 선언하는 데 있어서 `xoblivoir`에서는 약간 다른 방법 (더 편리한?)의 폰트 정의가 가능하다. 이것은 다음 세 명령으로 이루어져 있다.

```
\setkormainfont(<Bold>)(<Italic>){<Regular>}
\setkorsansfont(<Bold>)(<Italic>){<Regular>}
\setkormonofont{<Mono>}
```

이러한 폰트 정의 방식은 `xoblivoir` 클래스의 원래 의도인 ‘되도록 간단하게 필요한 것만’ 사용하자는 데서 나온 것이다. 그러므로 옛한글 설정이나 자간, 장평 등 복잡한 폰트 속성을 적용하려 한다면 당연히 `XgTeX-ko`의 (즉 `fontspec`의) 설정 방식을 이용하는 것이 좋다.

이후의 설명은 `\setkormainfont`에 대해서만 한다. 다른 두 명령의 사용법은 동일하다.

```
\setkormainfont{Font_Family_Name}
```

한글 글꼴 가운데 어떤 것은 — 모든 한글 글꼴이 그렇지는 않다는 데 주의하라 — 글꼴 가족을 이루고 있는 것이 있다. 예를 들면 “맑은 고딕”은 맥락에 따라 “맑은 고딕 Bold”를 글꼴 가족으로 인식한다. 이런 경우 예를 들어

```
\setsansfont{Malgun Gothic}
```

이렇게만 지정해도 굵은 글꼴이 요구되는 곳에서는 “맑은 고딕 Bold”를 식자해준다. 이렇게 잘 만들어진 글꼴을 이용하는 경우, 간단히 하나의 옵션으로 글꼴을 지정해주는 것만으로 굵은 글꼴을 사용하게 할 수 있다.

```
\setkormainfont[Options]{Font_Name}
```

만약 각괄호 ([])가 첫 인자로 나오면 이 명령의 사용법은 \setmainhangulfont와 동일하다. 즉 fontspec 명령을 그대로 쓸 수 있다.

```
\setkormainfont[Options]{Font_Name}[hnj_Options]{hnj_Font_Name}
```

다른 점은 위와 같이 한자 글꼴을 잇대어 정의할 수 있다는 것이다. 이 때 반드시 한자 글꼴 추가 정의가 시작된다는 뜻으로 각괄호 []를 써주어야 한다. 아무런 옵션을 지정하지 않더라도 이 부호는 있어야 한다.

## 폰트 가족의 지정

한글 폰트의 폰트 가족(font family)은 인위적으로 지정해주어야 할 때가 있다. 여러 한글 폰트를 임의로 묶어서 하나의 폰트 가족처럼 다루는 경우이다.

한글의 폰트 가족은 Regular/Bold/Italic 으로 이루어진다고 가정하고 있다.<sup>6)</sup> 폰트 가족을 지정하여 글꼴 설정을 할 때는 다음과 같이 한다.

```
\setkormainfont(<hgBd>)(<hgIt>){<Hangul>}
\setkorsansfont(<hgBd>)(<hgIt>){<Hangul>}
```

첫 두 인자를 둘러싼 괄호가 각괄호 []가 아니라 괄호 ()임에 주의하라.<sup>7)</sup> 아무튼 괄호 ()로 둘러싸인 부분은 생략가능하다. 첫 괄호 인자만 있고 이탤릭 글꼴을 지정하지 않으면 기본 글꼴로 치환된다.

한글 폰트와 별도의 한자 폰트를 설정할 필요가 있을 때는 다음과 같이 한다.

6) 원한다면 BoldItalic까지 포함할 수 있으나, xoblivoir는 BoldItalic을 별도로 설정하는 것을 기본으로 삼지 않았다. 지나치게 번거로울 뿐 아니라 BoldItalic에 특별한 의미를 부여하기 힘들었기 때문이다. 반면, Xe<sub>La</sub>TeX-ko는 BoldItalic을 별도로 설정할 수 있게 하고 있으므로 혹시라도 원한다면 이것은 사용자에게 맡겨져 있다.

7) 이것은 2008/11/09 버전의 변경사항이다. 하위호환성을 제공하지 않으므로 이전의 모든 소스를 수정해야 한다.

```
\setkormainfont(<hgBd>)(<hgIt>){<Hangul>}(<hjBd>)(<hjIt>){<Hanja>}
\setkorsansfont(<hgBd>)(<hgIt>){<Hangul>}(<hjBd>)(<hjIt>){<Hanja>}
```

한자를 한글과 별도의 폰트로 식자하도록 하는 것은 그 번거로움을 피할 수 없지만, 한글 폰트 가운데 한자 영역이 거의 없거나 있더라도 부실한 것이 너무 많기 때문에 현재의 한글 폰트의 사정을 돌아볼 때 부득이하고 불가피하다.

두 번째 선택 인자는 이탤릭체를 나타낸다. 이 인자를 지정하지 않으면 기본 글꼴로 치환된다. Regular 폰트를 이 자리에 쓰고 싶을 때가 있다. 이럴 경우 간단히

```
\setkormainfont(BOLD)(*){REGULAR}
```

와 같이 (\*)로 줄여 쓸 수 있다. 즉, 아예 두번째 인자가 없으면 REGULAR 폰트로 치환되고, (\*)로 지시되면 REGULAR 글꼴을 이탤릭으로 식자한다. 만약 itemph 옵션이 지시되어 있다면 REGULAR의 FakeSlanted 글자가 식자되고, gremph 옵션이 지시되어 있다면 이 자리에 선언된 글꼴이 굵은 모양으로 찍힐 것이다. 이 줄인 옵션은 두 번째 선택 인자, 즉 이탤릭 선언에만 동작하며, \setkormainfont와 \setkorsansfont에서 쓸 수 있다.

## 5.4 디폴트 치환

필요한 정보가 충분히 주어지지 않으면 폰트 설정은 기본값으로 치환된다. 아래 몇 가지 사용례를 보기로 하자.

```
\setkormainfont{HCR Batang LVT}
```

위의 명령을 실행하면 Regular와 Bold, Italic이 지정한 폰트로 바뀐다. 예시한 글꼴은 Bold 서체가 font family를 이루고 있으므로 굵은 글꼴을 자동으로 HCR Batang LVT Bold로 식자한다. 그러나 그렇지 않은 글꼴을 예를 들면,

```
\setmainfont{YJ BELLA Medium}
```

과 같이 하는 경우 양재벨라체가 굵은 글꼴 자리에도 똑같은 글꼴로 식자하게 된다.

```
\setkormainfont[BoldFont={NanumMyeongjo ExtraBold},%
  ItalicFont={NanumBarunGothic},%
  ItalicFeatures={FakeSlant={.17}}]{NanumMyeongjo}
```

위와 같이 fontspec 문법의 옵션을 지정할 수 있다.

```
\setkormainfont(NanumMyeongjo ExtraBold){NanumMyeongjo}
```

이 명령은 Regular와 Bold만 지정된다.

```
\setkormainfont(NanumMyeongjo ExtraBold)(*){NanumMyeongjo}
```

이 명령은 Italic에 NanumMyeongjo를 식자하게 한다. gremph에서는 upshape와 itshape가 똑같이 찍히게 되지만 itemph에서는 같은 글꼴이 기울어져 식자될 것이다.

```
\setkormainfont(* ExtraBold)(Malgun Gothic){NanumMyeongjo}
```

이것은 Regular, Bold, Italic을 한 가족으로 묶는다. BoldItalic이 만약 필요해지면 Bold체와 동일하게 설정된다.<sup>8)</sup> 한편 (\* ExtraBold)라는 표기는 NanumMyeongjo와 같은 이름이기 때문에 줄여 쓴 것이다.

한자 글꼴을 별도로 설정해야 할 때가 있다. 한자 글꼴을 설정하기 위해서는 위의 명령을 확장하여 다음과 같이 지정한다.

```
\setkormainfont(한글돋드)(한글이탤릭){한글레귤러}%
  (한자돋드)(한자이탤릭){한자레귤러}
```

()로 둘러싸인 부분은 모두 생략가능하지만, 이것이 한자 폰트를 설정하는 것이라는 점을 알리기 위해 한자 시작 부분에 ( 문자를 지정해야 한다. 즉,

```
\setkormainfont(){NanumMyeongjo}(){HCR Batang LVT}
```

8) BoldItalic은 위에서 설명한 (\*) 줄인 옵션의 영향을 받지 않는다.

위와 같이 하면 한글 영역에 나눔명조, 한자 영역에 함초롬 LVT 바탕을 식자할 수 있다. 이 때 볼드를 지정하지 않지만 최소한 ( )를 써주어야 한다는 점에 주의한다. 한글 폰트 정의에서 괄호 ( )를 사용했다면 한자 정의 확장에도 같은 괄호 ( )를 써야 한다. 한글 폰트 정의에서 각괄호 [ ]를 사용했다면 한자 정의 확장에도 같은 괄호 [ ]를 써야 한다.

```
\setmonoscale{<scale>}
```

한글 mono 폰트를 기본값(은 타자)이 아닌 글꼴로 지정할 경우 scaling이 필요할 때가 있다. 이를 위하여 위의 명령이 정의되어 있다. 이것은 한글 및 한자 mono 글꼴에만 영향을 미친다. 라틴 문자자는 fontspec 옵션으로 지정하면 될 것이다.

문장 중간에서 한글 및 한자 폰트를 일부 바꾸어 써야 할 때가 있다. 이 경우에는 Xe<sub>La</sub>TeX-k<sub>o</sub>의 \hangulfontspec 명령을 이용하면 매우 간편하게 이를 실현할 수 있다.

## 5.5 ExternalLocation

ExternalLocation으로 폰트를 찾도록 하는 설정은 특히 폰트 정보가 이상한 한글 폰트를 사용할 때를 위해서도 매우 중요하다. 굳이 폰트 캐싱을 할 필요가 줄어들기도 하므로, 이 방식을 선호하는 경우도 있다. xoblivoir에서는 “별표붙인 명령”으로 지정한다. 참고로 별표붙인 명령을 쓸 때는 fontspec 옵션을 지정하는 명령형식을 쓸 수 없다. 그 이유는 fontspec 명령 형식에서는 ExternalLocation 옵션을 줄 수 있기 때문이다.

```
\setkormainfont*(BOLD)(ITALIC){REGULAR}
```

TeX Live를 포함하여 대부분의 TeX 임플리멘테이션에서 시스템의 폰트 폴더를 kpathsearch로 찾을 수 있으므로 예컨대 Windows 폰트 폴더의 폰트들도 이 방식으로 호출할 수 있다. 한편, 한겨레결체의 경우 폰트 이름을 라틴 문자로 바꾸어두고 ExternalLocation으로 찾아야만 사용할 수 있다.<sup>9)</sup> 휴먼명조와 같은 글꼴도 마찬가지로 지여서 이 글꼴은 ExternalLocation 옵션으로 HMKMM.TTF를 직접 지시하는 것이 가장 안전하다.

한글 폰트 이름은 폰트 자체의 특성에 따라서 매우 복잡한 양상을 보인다. 이 글에서는 Windows에서 확인할 수 있는 이름을 기준으로 하고 있으나, 플랫폼에

9) 즉, 폰트 이름 정보가 잘못되어 있는 글꼴이다.

따라서 이름으로 폰트를 호출하기 위해서 다른 방법을 사용해야 할 수도 있다. 가장 안전한 방법은 ExternalLocation으로 폰트 파일의 이름을 부르는 것이다. 이때 몇 가지 이슈가 있다.

- (가) Windows에서는 한글 폰트 파일 이름을 부를 수 없다. 모든 폰트 파일 이름은 라틴 문자이어야만 한다. 그 이유는 한글 폰트 파일 이름이 윈도우즈 시스템 인코딩인 CP949로 불러져야 하는데 우리가 작성하는 tex 원본 파일은 UTF-8 인코딩이므로 한글 파일 이름을 호출하는 것이 윈도우즈에서 원천적으로 불가능하기 때문이다.
- (나) 반면, 시스템 로케일이 utf-8인 매킨토시나 리눅스에서는 한글 폰트 파일 이름도 ExternalLocation으로 호출할 수 있을 것이다. 리눅스에서는 모르겠으나 매킨토시에서는 이것이 가능했다.
- (다) texmf.cnf의 OSFONTDIR 변수를 수작업으로 수정해야 하는 경우가 있다. Windows의 KC2008이나 TeXLive 자체가 이것을 잘 설정하여 주므로 사용자가 신경 쓸 것이 없으나 경우에 따라 이 변수를 설정해야 할 필요가 있을 수 있다. 맥의 경우, /usr/local/texlive/2013/texmf.cnf 파일을 열어서<sup>10)</sup> 다음 내용을 자신의 환경에 맞게 써넣으면 될 것이다.

```
OSFONTDIR = {/Library/Fonts;~/Library/Fonts;/System/Library/Fonts;}
```

- (라) ExternalLocation으로 폰트를 호출한다는 것은 kpathsearch를 이용한다는 것이다. 그러므로 texmf 트리 아래에 해당 폰트를 가져다두고 mktexlsr해도 그 폰트에 접근할 수 있다.

폰트를 이름으로 호출하는 것은 폰트 자체에 정의된 이름과 파일이름이 다른 것이 너무 많아서 손쉽게 말하기가 어렵다. 대부분의 시스템에서(맥 제외) fc-list 명령은 XeTeX으로 부를 수 있는 폰트 이름을 확인하게 해준다. 한편 LuaTeX의 경우는 좀더 복잡하여 XeTeX에서 호출할 수 있는 것과 같을 때도 있고 다를 때도 있다. 파일 이름으로 호출(즉 ExternalLocation)하는 것이 가장 편리할 때도 있다.

## 5.6 이탤릭, 기울임

한글 글꼴에 이탤릭은 없다. 그러므로 강조를 위해 기울임으로 이탤릭을 대응하는 것은 그다지 권장하지 않는다. 예를 들어

10) /usr/local/texlive/2013/texmf/web2c/texmf.cnf를 수정해서는 안된다.

```
\setkormainfont(* Bold)(Gungsuh){Malgun Gothic}
```

으로 지정했을 때, 궁서체가 이탤릭에 해당하는 폰트로 설정된다. gremph가 디폴트이므로 궁서체는 곧은 모양으로 찍힌다.

그러나 디자인 상의 효과를 위해서나 다른 이유에서 이 서체를 기울이고 싶은 경우가 있을 것이다. 이 경우 [itemph] 클래스 옵션을 지정한다. 그 반대의 경우는 [gremph]이고 이것이 디폴트이다. 부분부분 기울이려 한다면 fontspec 명령을 써서 조작할 수 있으므로 별도로 명령을 만들어두거나 하지 않았다.

## 제 6 절 그밖의 사항들

### 6.1 enumerate

enumerate 패키지의 enumerate 아이템 항목 머리 설정은 다음과 같이 한다. 엔진과 무관하게 동작한다.

```
\begin{enumerate}[(㉠)] \tightlist
\item 첫째 항목
\item 둘째 항목
\end{enumerate}
```

(㉠) 첫째 항목

(㉡) 둘째 항목

paralist에서 위와 같은 방식으로 항목 머리를 설정하려면 레거시 텍을 위해서는 dhucs-paralist를, XeTeX과 LuaTeX을 위해서는 xob-paralist를 로드한다.<sup>11)</sup>

### 6.2 graphicx, xcolor

pdfL<sup>A</sup>T<sub>E</sub>X 문서는 dvi 드라이버로 어떤 것이 실행될지 모르기 때문에 graphicx 패키지의 로딩에 주의를 기울여야 한다. 즉

pdftex

11) 물론 paralist 자체는 그 이전에 부르거나 xob-paralist(dhucs-paralist)가 스스로 부르도록 할 것이고 xob-paralist(dhucs-paralist) 뒤에 paralist만 별도로 다시 부르면 안 된다.

```
\usepackage[<driver>]{graphicx}
```

과 같이 하는 것이 안전하다. 그러나 대부분의 경우 <driver>를 지정하지 않아도 된다.

xe(lua)tex

X<sub>Y</sub>TeX과 LuaTeX의 경우는 graphicx 패키지에 대한 명시적인 호출이 없어도 png, jpg, pdf 그림을 잘 불러온다. 즉 oblivoir가 알맞게 이 패키지를 호출하여준다.

### 6.3 참조 인용, 자동 조사

자동 조사는 ko<sub>Y</sub>TeX에서와 동일하다. 한글 label은 X<sub>Y</sub>TeX, LuaTeX에서 사용할 수 있다. 레거시 텍에서 label 자리에는 한글을 쓸 수 없다.

pdftex

pdfTeX:

“소적 \ref{sec:font}\를 보라.”

“소절 제 5 절을 보라.”

X<sub>Y</sub>TeX, LuaTeX:

“소적 \ref{sec:폰트}\를 보라.”

### 6.4 방점

ko<sub>Y</sub>TeX에서는 \dotemph 명령의 방점<sup>12)</sup>을 지원했다. X<sub>Y</sub>TeX-ko에서도 이 명령을 사용할 수 있게 되었고, xoblivoir에서 조금 확장했다. 기본인 \dotemph 외에 \circemph \useremph 두 개의 명령을 더 쓸 수 있다.

xe(lua)tex

xoblivoir에서 기원한 새로운 텍 엔진을 위한 oblivoir에서 마지막의 \useremph 명령은 하나의 선택 인자와 두 개의 인자를 요구한다. 즉,

```
\useremph[<height>]{<dotchar>}{text}
```

height는 방점으로 사용할 문자가 올라갈 길이이고 dotchar는 방점 문자이다. text는 방점을 찍을 텍스트. 단, 이 \useremph 명령은 X<sub>Y</sub>TeX-ko에서 제공되는 같은 이름의 명령과도 그 사용법이 다르고 레거시 텍을 위한 oblivoir에서의 사용법과도 다르므로 주의를 요한다.

12) 가로쓰기에서는 점을 글자 위에 찍으므로 傍점이 아니라 上점이 맞겠지만 관행적으로 방점이라 불러왔다. 이 문장부호의 정확한 명칭은 “드러냄표”이다.

## 제 7 절 샘플 문서

이 작은 안내서에 더하여 간단한 oblivoir 샘플 문서를 하나 제공한다. 이 문서에서 여러 가지 memoir와 oblivoir의 기능을 살펴볼 수 있을 것이다. oblivoir-test.tex을 컴파일해보고 소스를 검토해보시기 바란다.

## 제 8 절 첨언

xoblivoir 사용이 어느 정도로 쉬운가 하면, 나는 맨처음 이 문서를 LyX에서 작성하여 export한 다음, 두 줄 정도를 지우고 폰트 설정명령만을 써넣었다. 그래도 훌륭한 Xe<sub>La</sub>TeX 문서가 만들어졌던 것이다.

이 글을 쓰기 시작할 때만 해도 Xe<sub>La</sub>TeX-ko와 xoblivoir는 완성되어 있지 않았다. 그러나 지금은 일반적인 문서를 작성함에 있어서 불편이 없을 정도가 되었다.

돌이켜보면, 한글을 T<sub>E</sub>X 문서에 사용할 수 있다는 사실 자체가 신기했던 그 때로부터 20여년이 흘렀다. 본격적인 한글Xe<sub>La</sub>TeX 시스템들이 나오기 시작했던 1990년대 중반으로부터 헤아려도 십수 년, 이 기간 동안 한글이라는 문자 체계를 식자하기 위해 지불해야 했던 엄청난 노력과 자원을 생각하면 금석지감이 없지 않다.

LuaT<sub>E</sub>X과 Xe<sub>La</sub>TeX이라는 유니코드 텍 엔진의 등장은, 이러한 모든 노력들을 일시에 해결해버렸다. 이제 한글 문자의 식자는 더이상 문제가 되지 않는다. 그러나 한글 문서다운 한글 문서, 한글 문서의 타이포그래피의 완성을 위한 길은 아직도 요원하다. 단순히 “글자를 찍는” 문제가 해결되었다고 해서 모든 일이 끝난 것은 아닌 것이다. 단지 더 생산적인 문제를 더 잘 구현할 수 있는 바탕이 갖추어진 것일 뿐이라고 생각한다.