

릭 (LyX) 소개

by 릭 팀 (the LyX Team)*

김강수 (웁김)†

2007년 8월 6일

차 례

차 례	i
1 릭 (LyX)의 철학	1
1.1 릭 (LyX)이란 무엇인가?	1
1.2 릭 (LyX)과 워드 프로세서	3
1.3 레이텍 (L ^A T _E X)이란 무엇인가?	5
2 도움말문서 둘러보기	7
2.1 매뉴얼의 형태	8
2.2 매뉴얼에서 사용되는 단위	10
2.3 매뉴얼의 구성	10
3 릭 프로젝트에 기여하기	12
3.1 릭에 기여하기	12
3.2 문서화에 기여하기	13

*코멘트 또는 에러 수정은 릭 문서화 메일링리스트 (lyx-docs@lists.lyx.org)로 보내주십시오.

†번역상의 잘못에 대한 지적은 info@mail.ktug.or.kr로 해주십시오.

1 릿 (L_YX) 의 철학

1.1 릿 (L_YX) 이란 무엇인가?

릿(L_YX)은 문서 작성 시스템이다. 주로 수식, 상호참조, 참고문헌, 색인 등이 갖추어진 복잡한 기술 과학 문헌을 작성하는 데 탁월하다. 길이가 길든 짧든, 예를 들어 자동 색서닝과 페이지네이션, 스펠체크 등 프로세싱 능력이 필요한 문서에 특히 좋다. 그밖에 엄마한테 편지를 쓴다든가 하는데도 사용할 수 있다. 그런 일이라면 더 쉽게 하는 간단한 프로그램이 있기가 하겠지만. 그러나, 배너, 전단지, 광고지 등을 제작하는 데는, 좀 고생하면 불가능하지는 않을 거라 해도, 전혀 적당한 툴이 아니다. 그 이유는 나중에 설명하겠다. 실제 사용례를 보면, 메모, 편지, 박사 석사 논문, 단행본, 학술지 논문, 연극이나 영화 대본, 사업계획서, 프레젠테이션... 등이 있다.

릿(L_YX)은 컴퓨터로 글쓰기에 있어서 전통적인 낡은 “타이프라이터 개념”과 결별하고 현대적인 마크업 언어 패러다임을 채택한 프로그램이다. 저자들이 조판의 전문가가 될 필요없이 최소한의 노력으로 전문적인 출력을 얻을 수 있도록 설계되었다. 조판 자체는 저자가 아니라 컴퓨터에 의하여 이루어진다. 릿(L_YX)을 사용함으로써 저자는 자신이 쓰고자 하는 내용에 집중할 수 있게 된다.

릿 사용상의 첫번째 어려움은 사고의 전환을 요구한다는 점이다. 한때 우리는 모두 문서작성을 위해 타자기를 사용했는데, 타자기의 한계를 극복하기 위한 특정한 트릭들을 배워야만 했다. “_” 문자 겹쳐찍기로 구현했던 밑줄긋기는 텍스트를 강조하는 방법이었다. 포 작성을 위해선 사전에 칼럼 사이즈와 탭스톱을 미리 알고 그것을 설정해두어야 했다. 마찬가지로 기법이 오른쪽 정렬된 텍스트에도 적용되었으며, 행 끝의 하이픈처리를 위해서는 행 끝에 가서 어떤 일이 일어날지를 미리 짐작하는 예리한 눈이 필요했다.

달리 말하면, 무슨 문자가 어디 가서 찍힐 것이냐 하는 세밀한 사항에 대해 끊임없이 주의를 기울이도록 훈련받았던 것이다. 그 결과, 거의 모든 워드 프로세서들이 이런 사고방식을 벗어나지 못했다. 여전히 공백 삽입에 탭스탑을 사용한다. 아직도 어느 페이지에 무엇이 찍힐 것인지 신경쓰게

된다. 텍스트를 강조하는 것은 폰트를 바꾸는 것으로 인식된다. 마치 타자기의 휠을 교체하는 것과 마찬가지로 생각하는 것이다. 이런 것이 이른바 WYSIWYG 워드 프로세서의 바탕에 깔려 있는 철학이다. 즉, “눈에 보이는 것이 출력 결과물이다(What You See Is What You Get).” 불행히도 이 접근방법이 가져오는 결과는 “보이는 것밖엔 출력할 수 없다(What You See Is All You Get)”는 것이 되고 만다.

릭(LyX)이 일반 워드 프로세서와 다른 점이 이것이다. 어느 글자가 어디에 찍힐 것인지는 신경쓰지 말자. 단지 자신이 의도하는 것을 릭에게 알려주기만 하면 릭이 스타일이라고 불리는 일련의 규칙¹⁾을 따라서 나머지를 알아서 처리할 것이다. 간단한 보기를 들어 보자.

레포트를 쓴다고 가정하자. 먼저 “서론”이라는 절(section)로 레포트를 시작하려 한다. 그래서 폰트 사이즈를 바꾸는 워드 프로세서의 메뉴를 찾아서 폰트 사이즈를 새로 설정하였다. 그리고 서체를 볼드체로 바꾸었다. 그런 다음 “제 1 절 서론”이라고 타이프했다고 하자. 나중에 이 부분이 문서의 다른 위치로 가야 하게 된다면, 이 절 앞에 새로운 절을 하나 새로 끼워넣기로 한다면, 당연히 이 절과 그 이후의 모든 절의 번호를 바꾸어야 하고 목차에 나타난 숫자도 다 교체해주어야 한다.

릭(LyX)에서는, 버튼 바의 왼쪽 상단 풀다운 메뉴를 끌어내려서 Section을 선택한 다음 “서론”이라고 입력한다.

이걸로 다 되었다. 이 섹션을 잘라서 다른 곳에 붙여도 그 상황에 맞게 번호가 자동으로 붙는다. 이 절에 상호참조 태그를 붙여두면 이 절을 참조할 때 릭이 자동으로 참조번호를 다 업데이트해주기 때문에 위치가 바뀌었다고 해서 번호를 다시 입력해야 하거나 할 필요가 전혀 없다.

일관성이라는 측면에 대해 생각해보자. 닷새가 지난 후에, 레포트 원고를 다시 열어서 제4절을 작성하게 되었다. 그런데 전에 제1절을 입력할 때 16포인트를 사용했던 것을 잊어버리고 이번엔 18포인트를 사용했다고 하자. 이런 문제가 릭에서는 발생하지 않는다. 절표제에 몇 포인트 문자를

1) 공평을 기하기 위해 언급하자면 최근의 유명한 오피스 스위트 프로그램들은 스타일 시트나 마크업 방식과 유사한 도구를 채택하고 있다. 그러나 이런 논리적 스타일 틀이 실제로 사용되는 경우가 매우 희박함을 경험적으로 알고 있다.

사용할 것인가는 컴퓨터가 관장하는 문제이지 저자의 소관이 아니다. 그런 일은 컴퓨터가 더 잘 기억하는 법이다.

다른 보기 하나 더. 목록(list)을 작성하는 경우이다. 워드 프로세서에서 목록(리스트)은 탭스탑과 개행(改行)으로 이루어진 문단이다. 각 목록 아이템에 레이블은 어떻게 붙여야 할지, 레이블의 모양은 어떠해야 할지, 아이템 사이에 엔터를 몇 번 쳐야 할지 등등에 대해서 미리 염두에 두고 있어야 한다. 그러나 릭에서는 딱 두 가지만 신경쓰면 된다. 하나는 리스트의 종류가 무엇이나 하는 것이고, 다른 하나는 그 리스트 안에 올 내용이 무엇이나 하는 것이다. 그걸로 충분하다.

그러므로, 릭(LyX)의 기본 아이디어는 다음과 같다. 자신이 무엇을 하고자 하는가가 중요하지 어떻게 할 것인가는 중요하지 않다. WYSIWYG이라는 개념 대신 릭이 표방하는 모델은 “What You See Is What You Mean”, 즉 WYSIWYM이다. 이것은 매우 강력한 아이디어로서, 글쓰기 기제(機制)를 현저히 단순화한다. 동시에 릭이 포스터나 전단지를 만드는데 그다지 좋은 툴이 되지 못하는 이유이기도 하다. 이런 경우에 중요한 것은 실제로 특정한 글자나 그림이 어느 위치에 배열되는가를 결정하는 것이지, 문단이나 절(section)과 같은 논리적·기능적 단위가 사용되는 것이 아니기 때문이다. 이것이 릭에 무슨 기능적 결함이 있다는 말은 아니다. 그저, 그런 식의 작업에 적합한 툴이 아니라는 말일 뿐. 닭잡는 데 소잡는 칼을 쓰지는 않잖아.²⁾

1.2 릭과 다른 워드 프로세서의 차이점

릭(LyX)에는 없는 것들이다.

- 자(ruler)
- 탭스탑
- 추가공백(Enter 나 Space를 두 번 이상 누르는 것)

2) 원문은 you don't use a screwdriver to drive in nails.

자와 탭스탑은 페이지 상의 특정 위치에 무엇이 오는가를 보여주기 위한 것인데, 릭에서는 쓸모없다. 페이지 배열은 프로그램 소관이 지 글쓰는 이 가 상관할 것이 아니다. 추가공백도 비슷하다. 문맥상 필요하다면 릭이 추가할 것이다. 문단 사이에 엔터를 쳐서 빈 줄을 삽입할 수 없는 것이 처음에는 귀찮게 여겨질지 모르지만, WYSIWYM 방식에 익숙해지면 수 공이 같 것이다.

다음 몇 가지는 릭에도 있지만 흔히 생각하는 그런 것과 좀 다르게 보이는 것들이다.

- 문단 들여쓰기
- 페이지 나누기
- 행간격 (단행, 복행간격 등)
- 수평공백과 수직공백
- 폰트 및 폰트 사이즈
- 타입페이스 (굵게, 기울임, 밑줄 등)

이런 기능들이 릭에도 있지만 보통 사용자가 자주 쓸 일이 없다. 릭이 스스로 글쓰는 이의 의도에 맞추어 조절한다. 장절 표제와 본문 등 문서 요소들의 타입페이스와 폰트 사이즈는 자동적으로 설정된다. 문단 들여 쓰기도 맥락에 의존한다. 문단 유형이 다르면 들여쓰기 방법도 달라진다. 페이지 나누기는 자동적으로 이루어진다. 일반적으로, 행간격과 단어간격, 문단간 간격 등은 릭이 결정한다.³⁾

마지막으로, 우리가 생각하기에 릭(및 L^AT_EX)이 많은 워드 프로세서보다 우월한 영역이 몇 가지 있다.

- 하이픈처리

3) L^AT_EX에 대한 지식이 필요하다라는 문제가 있기는 하지만, 이런 설정값을 조절하는 방법도 있다. 문서 전체에 대해서 설정값을 바꿀 수도 있고 일부 문단만 그리할 수도 있다. 자세한 것은 *User's Guide*와 *Extended Features* 안내서를 참조하라.

- 나열식 문단 유형
- 수식
- 표 (table)
- 교차참조

당연히, 요즘 워드 프로세서들은 수학기호, 표, 하이픈처리 등을 잘 처리하며, 많은 부분에서 스타일 정의 방식과 WYSIWYM 개념을 채택하는 쪽으로 가고 있다. 그러나, 그런 변화는 최근에 와서 향상된 기능인 데 비해, 릭은 처음부터 L^AT_EX 문서처리 시스템에 기반하고 있다. L^AT_EX은 20년전부터 쓰여왔고, 현재도 동작한다.

1.3 레이텍 (L^AT_EX) 이란 무엇인가?

레이텍 (L^AT_EX)은 1985년 레슬리에 램포트가 설계한 문서작성 시스템이다.⁴⁾ 이것은 다널드 커누스가 1984년에 만든 T_EX이라는 이름의 조판 언어로부터 파생된 것이다. 텍 (T_EX)은 ASCII 파일의 스크립트로 쓰여진 조판 명령의 계열을 받아들여 그 명령을 수행한다. 인쇄업에서 통용되던 많은 “기법”들이 커누스에 의해 컴퓨터 알고리즘으로 모델링되어 텍에 통합 구현되었는데, 그 결과 훌륭한 인쇄 품질을 얻을 수 있게 되었다. 텍으로부터 출력되는 형식은 *pdf* 라는 portable document format 이거나 소위 “장치독립적” 파일인 *dvi* 이다. *dvi* 포맷은 미리보이게 주로 쓰이고 최종적으로 PostScript 와 같은 출력형식으로 변환되는 중간단계 파일로 쓰이기도 한다.

텍 (T_EX)은 조판 엔진에 그치는 것이 아니고 매크로 정의를 가능하게 한다. 텍을 사용하는 많은 사람들이 많은 조판 세부사항을 일일이 지정할 필요가 없도록 커누스가 제작한 매크로 패키지를 실제로 사용하고 있다. 레슬리에 램포트가 등장하는 지점이 이곳이다. 그는 보다 사용자-친화적이고 타입세팅에 신경을 덜 쓰게하는 매크로 패키지를 제작하려 하였다.

4) 이 절의 내용은 Helmut Kopka와 Patrick Daly가 쓴 *A Guide to L^AT_EX 2_ε*에서 가져왔다. *User's Guide*의 참고문헌에 이 책의 항목이 있다.

섹션, 표, 수학적식을 일관되고 통합된 형식으로 작성하는 일련의 명령을 정의하려 하였던 것이다. 이것이 레이텍이 탄생한 사연이다.

이제 L^AT_EX이 발전하고 성장함에 따라 여러 사람들이 자신의 텍 매크로 패키지를 제작하기 시작하였다. 그 중에는 슬라이드 만들기, 수학 저널용 논문 작성 등등이 포함된다. 원래의 텍 기능을 이용하는 사람도 있고 레이텍을 수정하는 경우도 있었다. 이런 혼란상을 정리하기 위해서 1980년대 후반에 L^AT_EX 전문가(L^AT_EX-nician)들이 L^AT_EX 2_ε를 만들었는데, 이것이 현재 버전이다. 이 새 버전의 L^AT_EX에는 텍 매크로 제작 명령을 더 쉽게 사용할 수 있도록 하는 것, 새로운 폰트 선택 스킴을 도입하는 것 등이 포함되었다. 사실 레이텍은 이미 그 자체로 하나의 확장된 언어이다! 전세계 많은 사용자들이 표준 레이텍을 보충하는 자신의 레이텍 추가 스타일들을 제작해왔다.

레이텍을 확장하는 데는 두 가지 형식이 있다. 클래스와 스타일이 그것이다. 클래스라는 것은 book이나 article과 같은 어떤 새로운 문서 형식을 기술하는 레이텍 매크로 집합이다. 슬라이드를 위한 클래스, 물리 수학 저널을 위한 클래스... 등이 있을 수 있으며 많은 대학교들은 논문 형식을 위한 자체 클래스를 제공하고 있기도 하다! 스타일이란 새로운 문서 유형을 정의하는 것이 아니라서 클래스와 구별된다. 스타일이 하는 일은 어떤 문서가 사용하는 특정한 형식의 동작방식을 정의하는 것이다. 예컨대 릭은 용지의 여백과 행간격을 제어하기 위해서 그러한 목적으로 제작된 두 개의 서로 다른 레이텍 스타일을 사용한다. 스타일 파일의 종류는 매우 다양해서, 봉투의 라벨 인쇄, 들여쓰기 재조정, 새로운 폰트의 추가, 그림의 제어, 페이지 면주의 장식, 참고문헌 목록의 변화, 각주·표·그림의 위치와 모양의 변경, 목록 모양의 조정 등 여러 영역의 다양한 기능에 걸친다.

요약하자.

텍 (T_EX): 매크로 기능을 갖춘 조판 언어.

레이텍 (L^AT_EX): 텍을 조판엔진으로 사용하는 매크로 패키지.

클래스 (classes): 레이텍에서 문서 유형을 기술함.

스타일 (styles): 레이텍의 표준 동작을 어떤 식으로든 변경함.

릭(LyX): 레이텍을 사용하여 출력을 얻는 비주얼한 WYSIWYM 워드 프로세서.

릭이 다른 워드 프로세서와 구별되는 것이 레이텍 때문이다. 릭은 레이텍을 사용하기 위한 그림 사용자 인터페이스(GUI) 프론트엔드 프로그램이다. 릭의 목적은 레이텍의 매크로 명령에 대해서 모르더라도 레이텍을 이용하여 문서를 조판할 수 있게 하려는 것이다. 그러나 릭이 레이텍의 수많은 명령을 모두 시각적으로 활용할 수 있게 해줄 수는 없지만, 그래도 레이텍이 할 수 있는 것은 릭이 모두 할 수 있다.

레이텍과 마찬가지로 릭이 추구하는 것은 글쓰기의 맥락, 즉 저자가 쓰고 있는 내용이다. 그 글이 어떤 모양으로 출력될 것인가 하는 문제는 컴퓨터에게 맡기는 것이다.

2 도움문서 둘러보기

사용자의 질문에 쉽게 답하고 릭의 모든 기능을 기술하기 위해 도움문서들은 몇 개의 파일로 구분되어 있다. 각각의 파일들은 아래 기술할 것과 같은 나름의 목적을 가지고 있다. 이 파일들을 디벼보기 전에 먼저 이 장을 주의깊게 읽어야 한다. 알아두면 시간을 절약할 수 있는 많은 유용한 정보와 언급이 기록되어 있기 때문이다.

릭은 지속적으로 발전하고 있기 때문에, 따라잡으려고 노력은 하지만, 문서가 불충분하거나 낡은 내용일 수 있다. 릭의 많은 부분이 그러하듯이 설명서도 자신의 “일상적 직업”을 가지고 가족을 돌보는 많은 자원봉사자들의 작업이다. 독자도 어떤 도움을 주고 싶다면 이 문서 마지막에 있는 3절을 읽어보도록 하시라.

또한, 부탁할 것이 있는데, 이 매뉴얼의 어떤 부분이 혼란스럽거나 명확하지 않거나 잘못되어 있다면 주저없이 우리에게 알려달라! 현재의 문서 관리자들은 lyx-docs@lists.lyx.org로 연락할 수 있다. 문서를 읽어도 분명히 이해가 되지 않는 질문이 있고 신속한 답변을 원한다면, 사용자 메일링리스트가 활동중이다. lyx-users@lists.lyx.org로 연락하면 된다.

2.1 매뉴얼의 형태

매뉴얼을 인쇄해서 볼 사람도 있고 릭 화면으로 읽을 사람도 있을 것이다. 릭 파일과 인쇄된 출력물 사이에 약간의 차이점이 있다. 먼저, 타이틀이 단순히 문서 상단에 나와 있을 뿐이고, 인쇄본처럼 별면으로 표현되지 않는다. 그리고 각주와 목차 전체가 바로 다 보이지 않는다. **foot** 라고 생긴 부분을 마우스 왼쪽 버튼으로 클릭하면 각주 내용이 열린다. 목차는 tableofcontents 회색 박스를 클릭하거나 Navigate 메뉴를 클릭하면 자동적으로 내용이 펼쳐져 보인다.

인쇄본에서는 모든 교차참조가 각 장, 절, 소절 등에 해당하는 실제 숫자로 나타난다. 그러나 릭 화면에서는 모든 교차참조가 **Ref. sec:Contrib** 와 같은 밝은 회색 박스로 표현된다. 이 박스를 마우스 왼쪽 클릭하면, 문서 전체에 걸친 모든 교차참조자 목록이 들어 있는 대화상자가 나타난다. 이 문서에서는 딱 두 가지밖에 없는데 그 중 하나가 “sec:Contrib” 이다. 참조되는 절로 바로 이동하려면 상자를 마우스 오른쪽 클릭하거나 대화상자에서 Go to Label 버튼을 누른다. 원래 위치로 돌아가는 것은 매우 쉬운데, Go back 을 클릭하면 된다.

이 문서의 온라인 판과 인쇄본의 차이점을 이해했다면, 문서의 구성에 대해 살펴보기로 하자. 가끔 색다른 폰트를 사용한 것을 볼 수 있을 것이다.

- **강조 스타일 (Emphasized Style)**은 일반적인 강조, 인자, 서명, 다른 매뉴얼의 섹션 명칭, 그리고 저자의 주석을 나타내는 데 쓰였다.
- **타입라이터 서체 (Typewriter)**는 프로그램과 파일 이름, 릭 코드와 릭 함수 이름을 나타내는 데 쓰였다.
- **산세리프 서체 (Sans Serif)**는 메뉴, 버튼, 대화상자 이름, 키보드 키 이름에 쓰였다.
- **고유명사 스타일 (NOUN STYLE)**은 사람 이름에 쓰였다.
- **볼드체 (Bold)**는 레이텍 코드를 나타낼 때 쓰였다.

메뉴 단축키와 키매김에 대해 “확장기능 안내서”(Extended.lyx 파일)의 *Key Bindings* 절을 참조하면 된다. 단축키에 대해 언급할 필요가 있을 때는 다음과 같은 규칙을 따랐다.

- “C-”는 Control- 키를 의미한다.
- “S-”는 Shift- 키를 의미한다.
- “M-”은 Meta- 키를 의미하며, 일부 키보드에서 Alt- 키이다.
- “F1” ... “F12” 들은 기능키이다.
- “Esc”는 이스케이프 키이다.
- “Left” “Right” “Up” “Down”: 화살표 키들이다.
- “Insert” “Delete” “Home” “End” “PageUp” “PageDown”: 대부분의 피씨 키보드에서 커서 키 위에 놓인 여섯 개의 키들이다. “PageUp”과 “PageDown”은 어떤 키보드에서는 “Prior”와 “Next”라고 부르기도 한다.
- Return과 Enter 키는 둘 다 같은 키를 가리킨다. “Return”이라고 명칭을 붙인 키보드도 있고 “Enter”라고 하는 것도 있다. 릭은 두 키를 동일하게 취급한다.

가끔 “(‘Extended’를 보라)”는 표현을 만날 것이다. 각 기능에 대한 키매김 기본값을 “Extended Features Manual”에 요약해두었으므로 그 곳을 참고하라는 뜻이다. 릭 사용자들 사이에 가장 많이 쓰이는 키매김에 두 종류가 있는데 하나는 피씨 사용자들에게 친숙한 “CUA” 스타일이고, 다른 하나는 유닉스 시스템과 Emacs 에디터 사용자 사이에서 “성장”하여 널리 쓰이는 “emacs” 스타일이다. 특별히 언급이 없으면 안내문서의 키매김은 기본 CUA 맵에 의한 것이다. Emacs를 선호한다면 이 문서를 읽고 자신의 키매김을 스스로 찾아낼 수 있을 정도로 실력이 있는 사용자일 거라고 생각한다.

표 1: 단위

단위	명칭 / 설명
mm	밀리미터 (millimeter)
cm	센티미터 (centimeter)
in	인치 (inch)
pt	포인트 point (72.27 pt = 1 in)
pc	파이카 pica (1 pc = 12 pt)
sp	scaled point (65536 sp = 1 pt)
bp	big point (72 bp = 1 in)
dd	디도 didot (72 dd \approx 37.6 mm)
cc	시세로 cicero (1 cc = 12 dd)
Scale%	% of original image width
text%	% of text width
col%	% of column width
page%	% of paper width
line%	% of line width
theight%	% of text height
pheight%	% of paper height
ex	height of letter x in current font
em	width of letter M in current font
mu	math unit (1 mu = 1/18 em)

2.2 매뉴얼에서 사용되는 단위

이 문서에서 기술된 단위를 이해하려면, 릭의 모든 단위를 설명하는 표 1을 보라.

2.3 매뉴얼의 구성

다음 목록은 도움말 문서 각 파일의 내용을 요약한 것이다.

소개 (Introduction) 이 파일.

튜토리얼 (Tutorial) 릭을 처음 사용하는 사람으로서 이전에 레이텍조차도 사용해본 적이 없다면 이 문서에서부터 시작한다. 레이텍을 사용

해온 사람이라면 “레이텍 사용자를 위한 릭”이라는 절에서 시작해도 된다. 문서의 나머지 부분을 건너뛰어도 무방할 것이다.

사용자 안내서 (User’s Guide) 주요 문서이다. 이 문서에서 릭의 기본적인 동작과 기능을 거의 대부분 설명한다. 튜토리얼을 이미 읽었다고 가정한다.

확장사용안내서-넣기 (Extended Insets) 사용자 안내서의 확장이다. 표, 그래픽, 떠다니는 개체, 주석, 박스 등을 다루는 방법을 상세히 설명한다. 레이텍 도사를 위한 여러 가지 트릭도 소개한다.

확장사용안내서-기능 (Extended Features) 사용자 안내서의 확장이다. 레이텍 명령을 직접 사용하는 방법, 추가 레이아웃, 특별한 편집 기능 등을 다루고 있다.

설정변경 (Customization) 고급 릭 기능에 대한 설명이다. 릭의 전체 동작을 변경하는 방법, 키매김, 다국어, 설정파일 등에 대해 다루고 있다.

레이텍 설정 (L^AT_EX configuration) 릭은 인스톨 시 사용자의 시스템을 탐색한다. 이 파일은 릭이 설치시에 어떤 정보를 얻었는지에 대한 내용을 포함하고 있다. 시스템에 뭔가 부족한 것이 있다고 생각되면 읽어보라.

이 파일들은 서로를 참조하고 있다. 예를 들면 사용자 안내서는 인스톨과 설정변경에 대한 몇 가지 정보를 포함하는데, 이를 위해 설정변경 안내서를 읽도록 요청하는 식이다.

중요한 점 하나를 다시 지적해두자.

만약 릭을 처음 사용한다면, 튜토리얼을 읽으시오. 당장.

이걸 생략하면 계속 혼란스러울 수 있다. 릭은 워드 프로세서로 하는 일을 다 하지만, 전혀 다른 접근방법을 택하고 있기 때문이다.

3 릿 프로젝트에 기여하기

3.1 릿에 기여하기

릿은 대부분 C++로 쓰여졌다(레이텍 변환기는 Python으로 쓰였다). 이것은 큰 규모의 프로젝트로서 버그로부터 자유로울 수 없으며, 소스 코드의 개선이 필요하기도 하다.

3.1.1 버그 보고

릿을 사용하면서 버그라고 생각되는 동작을 발견할 수 있다. 드물기는 하나 프로그램이 죽는 경우도 있다. 사용자 인터페이스 문제를 릿 팀은 중요한 버그라고 간주한다. 릿 인터페이스에서 혼동되거나 명확하지 않은 부분을 지적해주는 것은 특히 도움이 된다.

릿은 버그 추적 시스템을 가지고 있다. <http://bugzilla.lyx.org/>가 그것이다. 버그로 보고하기 전에 이미 보고된 버그가 아닌지 이 버그 추적기를 체크해보도록 하라. 알려진 버그에 대한 코멘트나 새로운 버그 보고를 희망한다면 버그 추적기를 사용하거나 개발자 메일링리스트 lyx-devel@lists.lyx.org로 메일을 보내달라. 이 메일링리스트의 아카이브가 릿 웹사이트에 링크되어 있다: <http://www.lyx.org/>.

버그 보고가 유용한 것이 되려면 최소한 문제를 발견한 릿의 버전을 포함해야 한다. 정확하고 세밀한 묘사가 필요하다. 개발자가 버그 소스를 잡아내는 데 투자하는 시간이 길어질수록 다른 개발에 할애할 시간이 줄어든다. 릿을 실행한 시스템과 그 버전을 언급하라. 시스템에 설치한 라이브러리의 버전도 지적해주어야 한다. 그리고 릿이 사용한 외부 프로그램의 버전을 알려주는 것이 좋다. 컴파일 또는 설정 문제라면 `config.log` 파일을 첨부하고 사용한 컴파일러에 대해 지적해달라.

3.1.2 수정과 새 기능 추가

릿 소스를 수정하였고 그것이 릿의 일부가 되어야 한다고 생각한다면 그 변경사항을 `diff` 파일 형태로(단일 형식) 앞서 언급한 개발자 리스트에 보내달라. `change log`와 자신의 `patch`에 대한 설명이 포함되어야 한다.

3.2 문서화에 기여하기

릭 문서는 방대하지만 릭이 여전히 개발중에 있고 새로 릴리스될 때마다 새로운 기능이 추가되고 있어서, 문서에 변경을 가해야 할 필요를 느낄 수 있다. 여기서는 문서에서 오류를 발견했거나 개선을 위한 제안을 하고자 할 때 어떻게 할 것인지 설명한다.

3.2.1 매뉴얼 상의 오류 보고

문서에서 문제점을 발견했다면 메일링리스트 `lyx-docs@lists.lyx.org` 로 메시지를 보내달라. 문서화 팀이 적절한 수정을 가할 것이다.

3.2.2 문서화 팀의 일원이 되기

릭 프로젝트의 다른 부분도 그렇지만 문서화 프로젝트는 언제나 도움을 필요로 한다! 만약 문서화 프로젝트에 기여할 의사가 있으면, 다음 절차를 따라 하라.

1. 릭 소스 코드를 얻는다.

<http://www.lyx.org/trac/browser/lyx-devel/trunk/lib/doc>
디렉토리 안에 `DocStyle.lyx` 이라는 파일이 있다. 이것을 읽어보라.
이 파일은 문서의 스타일시트이다.

2. 그 다음으로 사용자 안내서와 튜토리얼을 읽는다.

이것을 읽어보면 아이디어가 떠오를 것이다. 튜토리얼과 사용자 안내서는 가장 최신의 문서이다. 매뉴얼이 어떤 모양이고 어떻게 읽혀 지기를 원하는지 알아차릴 수 있을 것이다.

3. 문서화 팀에 접촉한다.

`lyx-docs@lists.lyx.org`
이 메일링리스트에서 자신이 원하는 것을 논의하고 다른 사람들로부 터 피드백을 받을 수 있을 것이다.

자신의 관심이 텍스트의 의미를 명료하게 하는 것부터 문서의 주요 구조를 재편성하는 데 이르기까지 다양한 영역에 걸쳐 있을 것이다. 어떤 변경과 개선이든 기쁘게 받아들인다.