

LyX の高度な機能

第 2.4.x 版

LyX プロジェクトチーム*

2024 年 5 月 14 日

*このファイルは主に Richard Kimberly Heck によって管理されています。コメントをされたい場合や誤りを発見された場合には、LyX 文書化メーリングリスト <lyx-docs@lists.lyx.org> に英文にてお知らせ下さい。この文書の翻訳は、当初人見光太郎氏が行った貢献に基づいています。

目次

1	はじめに	1
2	LyX と L^AT_EX	3
2.1	LyX は L ^A T _E X をどう使っているか	3
2.2	L ^A T _E X ファイルを LyX に読み込む	4
2.3	LyX 文書に T _E X コードを挿入する	4
2.4	LyX と L ^A T _E X プリアンブル	6
2.4.1	L ^A T _E X プリアンブルについて	6
2.4.2	プリアンブルの変更	7
2.4.3	例	8
2.4.3.1	例 1: オフセット	8
2.4.3.2	例 2: ラベル	8
2.4.3.3	例 3: 段落の字下げ	9
2.4.3.4	例 4: この文書	9
2.5	LyX および L ^A T _E X のエラー	9
3	文書クラス	15
3.1	コレクション	15
3.1.1	AMS L ^A T _E X(米国数学会)	15
3.1.1.1	これらのレイアウトが提供するもの	16
3.1.2	追加フォント寸法	19
3.1.3	ヘブライ語	19
3.1.4	日本語(標準クラス)	19
3.1.5	日本語(JS シリーズ)	20
3.1.6	日本語(BX シリーズ)	20
3.1.7	日本語(JLReq クラス)	20
3.1.8	Koma-Script	21
3.1.8.1	概要	21

目次

3.1.8.2	<i>KOMA-Script Article</i> ・ <i>KOMA-Script Report</i> ・ <i>KOMA-Script Book</i>	22
3.1.8.3	新しい letter クラス：KOMA-Script Letter (KOMA-Script V.2)	25
3.1.8.4	問題点	26
3.1.9	ポーランド語 M. W. コレクション	27
3.1.10	Tufte コレクション	27
3.2	論文	27
3.2.1	Astronomy & Astrophysics	28
3.2.1.1	はじめに	28
3.2.1.2	最初の一步	28
3.2.1.3	ヘッダ部	28
3.2.1.4	概要	29
3.2.1.5	サポートされる環境	30
3.2.1.6	LyX でサポートされていないコマンド	30
3.2.1.7	図フロートおよび表フロート	30
3.2.1.8	Referee レイアウト	31
3.2.1.9	例示用論文	31
3.2.2	AAST _E X	31
3.2.2.1	はじめに	32
3.2.2.2	新しい論文を書き始める	32
3.2.2.3	論文を書き終わったら	32
3.2.2.4	特定のコマンドに関するコメント	33
3.2.2.5	FAQ とヒントと技とその他の考察	35
3.2.2.6	最後に	37
3.2.3	チェス	37
3.2.4	Elsevier	37
3.2.5	Paper	37
3.2.6	RevT _E X4	37
3.2.6.1	導入	38
3.2.6.2	プレアンブルに関すること	38
3.2.6.3	レイアウト	38
3.2.6.4	重要な注意	39
3.2.7	Springer 学術誌	39

3.3	書籍	39
3.3.1	Memoir	39
3.3.1.1	概観	39
3.3.1.2	基本的な機能と制約	40
3.3.1.3	追加された機能	41
3.3.2	Recipe Book	42
3.4	履歴書	43
3.4.1	Europass (2013)	43
3.4.2	Europe CV	43
3.4.3	Modern CV	43
3.4.4	Simple	43
3.5	書簡	43
3.5.1	DIN-Brief	43
3.5.2	フランス語 Letter (frletter)	44
3.5.3	フランス語 Letter (lettre)	44
3.5.4	G-Brief (V.2)	44
3.6	プレゼンテーション	44
3.6.1	Beamer	44
3.6.2	Foil \TeX	44
3.6.2.1	はじめに	45
3.6.2.2	最初の一步	45
3.6.2.3	サポートされる環境	47
3.6.2.4	スライドの組を作る	48
3.6.2.5	サポートされない Foil \TeX 機能	50
3.6.3	Powerdot	51
3.6.4	Seminar	51
3.6.5	Slides (別名 SLI \TeX)	51
3.6.5.1	はじめに	52
3.6.5.2	使い始めるには	52
3.6.5.3	段落環境	53
3.6.5.4	スライド・オーバーレイ・注釈で発表用のスライドを作る	54
3.6.5.5	スライドクラスのひな型ファイル	58
3.7	レポート	58
3.7.1	Report	58

目次

3.8	台本	59
3.8.1	Broadway	59
3.8.1.1	はじめに	59
3.8.1.2	特別な問題	59
3.8.1.3	特別の機能	59
3.8.1.4	用紙寸法と余白	59
3.8.1.5	環境	60
3.8.2	Hollywood (Hollywood スペック・スクリプト)	61
3.8.2.1	はじめに	61
3.8.2.2	特別な問題	61
3.8.2.3	特別な機能	61
3.8.2.4	用紙寸法と余白	61
3.8.2.5	環境	61
3.8.2.6	脚本で使われる特別な用語	62
4	モジュール	65
4.1	特定の学術分野	65
4.1.1	化学：危険有害性情報・注意書き	65
4.1.2	化学：リスクと安全性に関する表示	65
4.1.3	言語学	65
4.2	アクセシビリティ	65
4.2.1	点字	65
4.3	記註及び改訂	66
4.3.1	FiXme 註釈	66
4.3.2	PDF コメント	66
4.3.3	PDF フォーム	66
4.3.4	ルビ(振り仮名)	66
4.3.5	TODO 註釈	66
4.4	書誌情報	67
4.4.1	Natbib を使用した APA 様式	67
4.5	ボックス	67
4.5.1	色付き装飾ボックス	67
4.5.2	画像ボックス	67
4.5.3	節ボックス	68
4.5.4	可変幅ミニページ	68

4.6	修正とハック	69
4.6.1	Computer Modern フォントの修復	69
4.6.2	L ^A T _E X カーネル修正 (廃版)	69
4.6.3	簡素形差込枠	69
4.6.4	表題とプリアンブルのハック	69
4.7	フロート及びキャプション	69
4.7.1	Algorithm2e フロート	69
4.7.2	二言語キャプション (多言語キャプション)	70
4.7.3	図番号を節毎に振る	70
4.7.4	表番号を節毎に振る	70
4.8	脚註及び後註	70
4.8.1	後註 (基本)	70
4.8.2	後註 (拡張)	70
4.8.3	脚註を後註に (基本)	71
4.8.4	脚註を後註に (拡張)	71
4.9	レジャー・スポーツ・音楽	71
4.9.1	チェス盤	71
4.9.2	Lilypond 記譜	71
4.10	高度な簡条書き	72
4.10.1	調整可能な簡条書き	72
4.10.1.1	調整可能な連番簡条書き	72
4.10.1.2	Report における連番段落	73
4.10.1.3	連番の復帰	73
4.10.1.4	簡条書きの余白	74
4.10.1.5	さらなる調整	74
4.10.2	段落簡条書き	75
4.11	文芸的プログラミング	75
4.11.1	Noweb	75
4.11.2	Rnw (knitr)	76
4.11.3	Sweave	76
4.12	数式	76
4.12.1	AMS 定理	76
4.12.2	AMS 定理 (拡張)	77
4.12.3	AMS 定理 (拡張・章毎に種別連番)	77
4.12.4	AMS 定理 (拡張・種別連番)	77

目次

4.12.5	AMS 定理 (章毎に種別連番)	77
4.12.6	AMS 定理 (種別連番)	78
4.12.7	数式番号を節毎に振る	78
4.12.8	標準定理	78
4.12.9	標準定理 (命名可)	78
4.12.10	標準定理 (章毎連番)	78
4.12.11	標準定理 (節毎連番)	79
4.12.12	標準定理 (章毎に種別連番)	79
4.12.13	標準定理 (節毎に種別連番)	79
4.12.14	標準定理 (種別連番)	79
4.12.15	標準定理 (連番なし)	80
4.12.16	数式群 (subequations)	80
4.13	ページレイアウト	80
4.13.1	ヘッダ・フッタ文の設定	80
4.13.1.1	装飾的なヘッダ・フッタ	81
4.13.2	文中横向き部分	82
4.13.3	多段組	83
4.13.3.1	基本	83
4.13.3.2	段組中の段組	85
4.13.3.3	応用例	85
4.13.3.4	これは序文の小々節見出しです	86
4.14	段落様式	90
4.14.1	任意の段落形	90
4.14.1.1	既定の形	90
4.14.1.2	任意の形	95
4.14.2	ぶら下げ段落	97
4.14.3	頭文字 (ドロップキャップ)	97
4.15	テキストマークアップ	97
4.15.1	ハイフネート可能なテキストマークアップ (Soul)	97
4.15.2	論理マークアップ	97
5	書誌情報	99
5.1	他の引用様式	99
5.2	区分された書誌情報	100
5.3	複数の書誌情報	101

6	ブリット	103
6.1	はじめに	103
6.2	外見	103
6.3	使用法	104
7	補完ツール	107
7.1	複合文書	107
7.1.1	一般的な操作	107
7.1.2	ファイル間の相互参照	108
7.1.3	すべての子文書での書誌情報	109
7.2	LyX 書庫	109
8	LyX とインターネット網	111
8.1	XHTML における数式出力	113
8.2	書誌情報と引用	114
8.3	索引	115
8.4	記号一覧および用語集	116
9	DocBook 出力	117
10	LyX サーバー	119
10.1	はじめに	119
10.2	LyX サーバーを開始するには	119
10.3	通常の通信	120
10.3.1	AppleScript (Mac OS X)	121
10.4	通知	121
10.5	単純な LyX サーバプロトコル	122
10.6	DVI と PDF の原文検索	122
10.6.1	自動有効化	123
10.6.2	手動有効化	123
10.6.3	特定のビューアを設定して使うには	125
10.7	出力先検索	128
10.8	TeX のチェック	129
10.8.1	はじめに	129
10.8.2	使いかた	130
10.8.3	微調整のしかた	131

目次

10.9	LyXでのバージョン管理	134
10.9.1	はじめに	134
10.9.2	LyXにおけるRCSコマンド	135
10.9.2.1	登録	135
10.9.2.2	変更をチェックインする	135
10.9.2.3	チェックアウトして編集	135
10.9.2.4	リポジトリのバージョンに戻す	136
10.9.2.5	コピー	136
10.9.2.6	最後のチェックインを取り消す	136
10.9.2.7	履歴を表示	136
10.9.2.8	改訂情報	136
10.9.3	LyXにおけるCVSコマンド	137
10.9.3.1	CVSの一般的な利用法	137
10.9.3.2	登録	138
10.9.3.3	変更のチェックイン	138
10.9.3.4	変更のチェックアウト	139
10.9.3.5	レポジトリ版に戻す	139
10.9.3.6	コピー	139
10.9.3.7	ファイル名変更	139
10.9.3.8	ローカルディレクトリチェックアウトのレポジトリか らの更新	140
10.9.3.9	履歴を表示	141
10.9.3.10	改訂情報	141
10.9.4	LyXにおけるSVNコマンド	141
10.9.4.1	登録	141
10.9.4.2	変更をチェックイン	141
10.9.4.3	チェックアウトして編集	142
10.9.4.4	リポジトリ版に戻す	142
10.9.4.5	コピー	142
10.9.4.6	ファイル名変更	142
10.9.4.7	レポジトリを使ったローカルディレクトリの同期 ¹	143
10.9.4.8	履歴を表示	143
10.9.4.9	ファイルロック	143
10.9.4.10	自動ロック特性	144

¹このコマンドは、`subversion` ≥ 1.5 でのみ動作することに注意してください。

10.9.4.11 文書の改訂情報	145
10.9.5 SVN と Windows 環境	146
10.9.5.1 準備	146
10.9.5.2 文書を Subversion 管理下に置くには	146
10.9.5.3 Windows 下で SVN と共に SSH トンネルを使うには	147
10.9.5.4 行末変換	148
10.9.6 LyX における GIT コマンド	149
10.9.6.1 登録	149
10.9.6.2 変更をチェックイン	150
10.9.6.3 リポジトリ版に戻す	150
10.9.6.4 ファイル名変更	150
10.9.6.5 履歴を表示	150
10.9.6.6 改訂情報	150
10.9.7 さらに調整	151
10.9.8 バージョン管理と文書比較	151
10.10 文芸プログラミング	151
10.10.1 はじめに	151
10.10.2 文芸プログラミング	152
10.10.2.1 書誌情報	152
10.10.3 LyX と Noweb を使った文芸プログラミング	153
10.10.3.1 ドキュメントとコードの生成 (weaving および tan- gling)	153
10.10.3.2 LyX の設定	157
10.10.3.3 デバッグ機能拡張	157
10.10.3.4 ツールバー拡張	158
10.10.3.5 色の変更	159
10.10.4 LyX と knitr/Sweave	159
索引	161

1 はじめに

いまお読みの取扱説明書『 $\text{L}_\text{Y}\text{X}$ 拡張機能篇』は、基本的には『ユーザーの手引き』の第二部です。これらの文書を別部とした理由は、単純に『ユーザーの手引き』がすでに大部となっていることと、『ユーザーの手引き』が、ほとんどの文書を作成する上で必要な基本機能をすべて説明してしまっているためです。一方で、 $\text{L}_\text{Y}\text{X}$ プロジェクトチームは、 $\text{L}_\text{Y}\text{X}$ を各設定ファイルと外部パッケージによって拡張可能な状態にしておくことを長期的な目標にしています。つまり、ある $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ パッケージをサポートした場合に、 $\text{L}_\text{Y}\text{X}$ 自身に変更を加えることなく、当該パッケージ用のレイアウトファイル（ないしモジュール）を作成すればいいようにしているのです。すでにこのような形で、いくつもの新機能がユーザーによって追加されています。この取扱説明書は、そのような形での貢献を文書化する冊子として位置付けられています。

この取扱説明書では、ファックスのサポート・バージョン管理のような、正しく機能するためには追加的なソフトウェアを必要とする特殊機能についても説明しています。さらに、 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ の強力な機能を直接使って文書を美しく仕上げるための、 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ ツールやテクニックを述べた章も収録しています。つきつめれば、 $\text{L}_\text{Y}\text{X}$ とはあくまで WYSIWYM たることを企図しているため、 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ の全機能の一部へのインタフェースは用意するものの、すべてを用意することはないでしょう。

もし、まだ『はじめの一步』をお読みでなかったら、この取扱説明書から読みはじめるのは誤りです。『はじめの一步』には、全取扱説明書に共通の表記法と書式が説明されていますので、まずそれからお読みください。さらに、この取扱説明書を読もうとする前に、『ユーザーの手引き』をよくお読みになって、 $\text{L}_\text{Y}\text{X}$ の全基本機能を習得してください。

この取扱説明書の取り上げられている話題は、 $\text{L}_\text{Y}\text{X}$ が $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ にどう関わっているかに強く関係していますので、まず次章では、 $\text{L}_\text{Y}\text{X}$ の内部動作がどうなっているか、 $\text{L}_\text{Y}\text{X}$ に希望する $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ コードを正確に生成させるにはどうしたらいいかを収録しています。本章は、あきらかに熟練 $\text{L}_\text{Y}\text{X}$ ユーザー向けの章となっています。

2 L_YX と L^AT_EX

2.1 L_YX は L^AT_EX をどう使っているか

本章は、T_EX 熟練者と L^AT_EX に興味のある方向けの章です。ここでは、L_YX と L^AT_EX が、印刷可能な出力を生成するのに、どのように協働しているのかを説明します。全取扱説明書中で本章だけは、L^AT_EX のことを多少なりともご存知であることを前提とします。

かつて L_YX は、「L^AT_EX の WYSIWYM なフロントエンド」と呼ばれていましたが、いまではこれは正しくありません。L^AT_EX のフロントエンドは、いくつも出回っています¹。これらは、基本的に L^AT_EX を走らせる機能を有し、編集時のファイルのエラーを表示する機能を持ったテキストエディタです。L_YX も、エディタであり、L^AT_EX を走らせる機能を有し、ファイル中のエラーを指摘しますが、それ以上にもっともっと多くのことをしてくれます。ひとつには、L_YX を効果的に使うのに、L^AT_EX の知識が必要とされないことが挙げられます。また L_YX は、それ自身、いくつかの拡張を L^AT_EX に加えています。時間のあるときにでも、いちど以下の手順を試してみてください。まずファイルメニューからエクスポート▷LaTeX（あるいは表示▷ソースを閲覧）を選択し、そこから出力される .tex ファイルのプリアンブルをご覧ください。L_YX によって特別に定義された新しいマクロが複数定義されていることに気づかれるはずです。これらのマクロは、文書中で使用する機能に応じて、自動的に定義されます。

自動的に L^AT_EX を実行するコマンドが、以下のようにいくつかあります。

- 表示▷書式
- 表示▷更新▷書式

これらは、最後に L^AT_EX を実行してからファイルに変更が加えられた場合のみ、L^AT_EX を実行します。

L_YX が編集時のファイルを L^AT_EX にかけるとき、L_YX は以下の段階をふみます。

¹有名なものとしては、Linux 上の T_EXmaker と kile、OSX 上の T_EXshop などがあります。もちろん vi と emacs の L^AT_EX モードも含まれます。

2 LyX と L^AT_EX

1. 文書を L^AT_EX に変換し、ファイルの拡張子を .lyx から .tex に変えて保存.
2. この .tex ファイルに対して (必要に応じて複数回) L^AT_EX を実行し、L^AT_EX ファイルをコンパイルするのに必要な他のコマンド (bibtex や makeindex など) を実行..
3. もしエラーがあれば、エラーログを表示.

L^AT_EX を表示▷DVI を使用して実行した場合には、LyX はこの後、DVI ビューアを実行して DVI ファイルを表示します. 表示▷PostScript を使用した場合には、LyX はさらに以下の段階をふみます.

- dvips を実行して、DVI ファイルを PostScript に変換.
- ghostview などの PostScript ビューアを実行して、PostScript ファイルを表示.

LyX は、他の書式を表示したりエクスポートしたりする際にも、同様のことを行います.

2.2 L^AT_EX ファイルを LyX に読み込む

LyX でファイル▷インポート▷LaTeX コマンドを使用すると、L^AT_EX ファイルを LyX にインポートすることができます. このコマンドは、tex2lyx というプログラムを呼び出し、foo.tex などというファイルから foo.lyx を生成します. LyX は、この変換したファイルを開きます².

tex2lyx は、文法的に正しい L^AT_EX ファイルのほとんどを変換することができますが、すべてを変換できるわけではありません. tex2lyx は、理解不能な部分は T_EX コードボックスの形で残しますので、ファイルを tex2lyx で変換したのち、T_EX コードボックスを探して、手動で正しく編集してください.

T_EX コードボックスがなにかご存じなければ、次節をお読みください.

2.3 LyX 文書に T_EX コードを挿入する

どの LyX 文書にもつねに T_EX コードを挿入することが可能であるというきわめて単純な理由によって、L^AT_EX にできることは、つねに LyX でも行うことができます. 一方、LyX は、すべての L^AT_EX 構成物を LyX 内に表示できるわけではなく、将来的にもそうならないでしょう. かりに L^AT_EX コマンドを LyX 文書に挿入する必要が出てき

²tex2lyx は、もちろんコマンドラインから実行することもできます.

た場合には、T_EX コードボックスを使用してください。これは、挿入▷T_EX コード若しくはショートカットキーで、文書中に挿入することができます。

以下は、L^AT_EX コマンドを L_AT_EX 文書中に挿入する例です。コードは以下のようになっているとしましょう。

```
\begin{tabular}{l}
\begin{minipage}{5cm}
これは、ミニページ環境の例です。
これには、
（フロートでない）図表を
含め、ほとんど何でも
入れることができます。
\end{minipage}
&
\begin{minipage}{5cm}
\begin{verbatim}
\begin{minipage}{5cm}
これは...
\end{minipage}
\end{verbatim}
\end{minipage}
\end{tabular}
```

上記のテキストを含んだ T_EX コードボックスが、この段落の直後にあります。この取扱説明書を L_AT_EX 上でご覧になっていれば、T_EX コード差込枠のみが見えるはずです。印刷した取扱説明書をご覧であれば、実際の結果が見れるはずです。

```
これは、ミニページ環境の
例です。これには、（フロー \begin{minipage}{5cm}
トでない）図表を含め、ほと これは...
んど何でも入れることがで \end{minipage}
きます。
```

T_EX コードを使用する方法に加えて、複雑な L^AT_EX コードを含む部分を別ファイルとして作成しておき、そのファイルをインクルードするのに挿入▷ファイル▷子文書を使用する方法もあります（取り込みの型として Input を選択しなくてはなりません）。この方法をとる場合には、.tex ファイルが正しく動作することが確認できて

2 LyX と L^AT_EX

いる場合のみに限った方がよいでしょう。さもないと、L^AT_EX エラーを追跡するたいへんな作業を行わなくてはならなくなります。

最後に、いくつか強調しておくべきことがあります。

- LyX は、あなたの L^AT_EX コードが正しいかどうかをチェックしません。
- 車輪の再発明をしないようご注意ください。

二番目の点に関して、LyX はひじょうにたくさんの機能を装備しており、その数は増えつづけています。ご自身で L^AT_EX コードを書かなくてはならないと思いつく前に、LyX がこれこれの機能をもっていないことを、取扱説明書で確実に確かめてください。さらに世の中には、封筒のラベルから美しい複数ページにわたる表を作るものまで、ありとあらゆることを行う膨大な数の L^AT_EX パッケージが存在します。詳細については、[CTAN](#) をご覧ください。

もし、文書中で例外的に手の込んだことをなさりたいければ、かならず良い L^AT_EX 関連書籍を参考とするようにしてください。『ユーザーの手引き』の書誌情報に多くの L^AT_EX 関連書籍が引用されています。

2.4 LyX と L^AT_EX プリアンブル

2.4.1 L^AT_EX プリアンブルについて

すでに L^AT_EX についてご存知であれば、ここでプリアンブル（前置文）が何に向いているか説明する必要はないでしょう。もしご存じなければ、以下の説明で大まかな理解は得られるはずです。詳しくは L^AT_EX 関連書籍をご参照ください。いずれにしても、以下で LyX 文書の L^AT_EX プリアンブルで何ができるか、または何をすることがないか説明をしますので、以下の各要点はお読みになった方が良いでしょう。

L^AT_EX プリアンブルは、L^AT_EX 文書のいちばん始め、本文の前の部分に来ます。ここでは以下のことを行います。

- 文書クラス（ドキュメントクラス）の宣言。
これは LyX がすでに行ってくれています。もしあなたが熟達した L^AT_EX 使いであり、自作の文書クラスを使いたい場合には、取扱説明書『カスタマイズ篇』をご覧ください。LyX に自作文書クラスへの仲介を行わせるにはどうすればよいかの情報がありません。
- パッケージ使用の宣言。
L^AT_EX パッケージは、パッケージがプリアンブルで宣言された場合のみ、文書

中で利用可能となる、追加コマンド群を提供します。パッケージをインクルードすると、特別なコマンドが提供されることに加え、文書に変更を加えることができます。例えば、`indentfirst` パッケージは、すべての段落の最初の行を字下げします。他にも、ラベルや宛名書き、余白設定等の様々なパッケージがあります。

- カウンタ・変数・長さ・幅の設定。
L^AT_EX のカウンタと変数の中には、プリアンブルでグローバルに設定しない**かぎり**、望んだ効果を持たないものがあります。(文書中で設定したり再設定したりすることのできる変数もあります。) 余白は、プリアンブルで設定しなくてはならない変数の代表例です。他には、箇条書きに付けるラベル書式が挙げられます。じつは、この設定は文書中どこでもできるのですが、プリアンブルでいちどだけ設定するのが望ましい方法です。
- ユーザー定義コマンドの宣言 (`\newcommand` または `\renewcommand` を使用)。
ユーザー定義コマンドは、文書中で頻出する L^AT_EX コマンドの短縮形です。このようなコマンドを宣言するには、プリアンブルが適した場所ですが、どこで宣言することも**可能**です(もちろんコマンドが初めて使用される場所よりも前です)。あまりないことではありますが、お使いの文書中に生の L^AT_EX コードがあふれている場合には便利でしょう。

L_YX は、その生成する `.tex` ファイルのプリアンブルに、それ自身の定義を追加します。これによって、L_YX の生成する L^AT_EX ファイルの可搬性を高めています。

2.4.2 プリアンブルの変更

L_YX が L^AT_EX ファイルのプリアンブルに追加するコマンドは、つねに同じです。これは、L_YX 自体にパッチを当てない**かぎり**変更できないようになっています。しかしながら、文書▷設定ダイアログの L^AT_EX プリアンブルの項から、プリアンブルに追加をすることはできます。L_YX は、プリアンブルダイアログ内のすべてのものを、L_YX 埋込みのプリアンブルに付け加えます。ご自身の宣言をプリアンブルに追加する前に、あなたがしようとしていることを L_YX がすでにサポートしていないかどうか確認してください(車輪の再発明をしないように、とはすでに申しあげたことです)。また、**プリアンブルコードが正しいことを確認してください**。L_YX は、プリアンブルをチェックしません。もしエラーがあれば、「Missing `\begin{document}`」といったエラーが出ることでしょう。このようなエラーが出たときは、プリアンブルを再確認してください。

2.4.3 例

以下は、プリアンブルに追加できるものとその効果を示す例です。

2.4.3.1 例 1：オフセット

L^AT_EX には、`\hoffset` と `\voffset` というページ位置を制御する 2 つの変数があります。その機能は名前から明らかなおりで。これらの変数は、ラベルを印刷するときなどに便利です。往々にして、印刷用紙の寸法とそのうち実際に印字可能な範囲は同じではありません。ここに `\hoffset` と `\voffset` が役立つわけです。

`\hoffset` と `\voffset` の既定値はともに 0 ポイント、つまりページをシフトしません。残念ながら、DVI ドライバによってはつねにページをシフトすることがあるようです。我々には、どうしてそのようなことをするのか、そしてシスアドがどうしてそれを修正しないのか理解できませんが、もし個人的に管理できないシステムで LyX を使用していて、シスアドがお馬鹿さんである場合には、`\hoffset` と `\voffset` が役に立ちます。かりに、上余白と左余白がつねに 0.5 インチずつ大きすぎるものとしましょう。その場合は、以下をプリアンブルに追加するとよいでしょう。

```
\setlength{\hoffset}{-0.5 in}
\setlength{\voffset}{-0.5 in}
```

これで余白が修正されます。

2.4.3.2 例 2：ラベル

いま、たくさんの宛名ラベルを印刷をしたいものとしましょう。ラベルシートを印刷するために便利な `labels.sty` という L^AT_EX パッケージが、お近くの CTAN アーカイブで入手可能です。お使いのシステムには、既定ではこのパッケージは導入されていないかもしれません。ご自身でお確かめください。また、このパッケージの取扱説明書を読みたいかもしれませんが、それもご自身にお任せします。これは使用例ですので、このパッケージをどう使うかの例だけお見せすることにします。

まず、`article` 文書クラスを使用していることをご確認ください。次に、以下のような設定をプリアンブルに追加しなければなりません。

```
\usepackage{labels}
\LabelCols=3
\LabelRows=7
\LeftBorder=8mm
```

```
\RightBorder=8mm
\TopBorder=9mm
\BottomBorder=2mm
```

これは Avery ラベルシートのストック 5360 番を使用する場合の設定です。これで、各宛名を `\begin{labels}` と `\end{labels}` で囲めば、宛名ラベルの印刷をすることができます。本機能およびその他の機能に関しては、`labels.sty` の取扱説明書に説明されています。

そのうち誰かが、このパッケージを直接サポートする L_YX レイアウトファイルを作ってくれるかもしれません。その誰かはあなたかもしれません。

2.4.3.3 例 3：段落の字下げ

アメリカ人は、すべての段落の最初の行を字下げするように教育されています。そのために彼らの他の変な習慣と同じく、いつものようにすべての段落の最初の行を下げるまで、アメリカ人は ご ち ゃ ご ち ゃ と文句を言いつづけるのです。(これはユーモアです。(ほんと?) ユーモアですってば。)

もちろん、これは標準的な組版ルールではありません。書籍では、典型的には、他の段落のあとにくる段落のみ字下げを行ないます。一行めを字下げをするそもそもの目的は、隣接する段落から区別するためです。したがって、図のあとや節の最初の段落など、先行する段落がない場合には、字下げをする必要はないのです。

けれどもあなたが典型的なアメリカ人なら (ユーモアですって!)、そのような小難しいことはかまわずに、いつもどおりの字下げをせずにはいられないでしょう! プリアンブルに次の行を加えて下さい。

```
\usepackage{indentfirst}
```

もしお使いの T_EX 頒布版が脳死状態でなければ、このパッケージが入っているはずなので、建国の父たちが意図したように、すべての段落が字下げされます。

2.4.3.4 例 4：この文書

この文書のプリアンブルを見てみると、どのような高度な設定ができるか、おおまかな理解が得られるでしょう。

2.5 L_YX および L^AT_EX のエラー

L_YX が L^AT_EX を呼び出す際、L_YX は L^AT_EX に、かりにエラーがあったとしても楽観的に無視して先に進むように指示します。その後 L_YX は、L^AT_EX の実行ログファイルを

2 LyX と L^AT_EX

事後処理用に使います。ログファイルを分析したのち、LyX はエラーを一覧表示するダイアログを表示します。エラーのうちどれかをクリックすると、LyX ファイル中、そのエラーの発生した場所に表示を移動します³。

人によっては、ログファイルを直接ご覧になりたいかもしれません。これは、文書▷LaTeX ログで見ることができます。エラーメッセージや警告には、よく見かけるものがいくつかありますので、それらをここに収録しておきます。ここにはないエラーや警告に関しては、LaTeX 関連の良書を参照してください。

- LaTeX Warning

これで始まるメッセージは、L^AT_EX コード自体を「デバッグ」するための警告メッセージです。相互参照や書誌情報を追加したり変更した場合に、もう一度 L^AT_EX を実行する必要があることを知らせるために、このようなメッセージが表示されます。このメッセージは、だいたいにおいて無視しても大丈夫です。

- LaTeX Font Warning

L^AT_EX がフォントを見つけられなかったと言う警告メッセージです。多くの場合、このメッセージの後に、L^AT_EX が使用した代替フォントに関する情報が続きます。これも無視して構いません。

- Overfull \hbox

L^AT_EX はほぼどんなときでもこの警告を吐き出します。どんな文書を書いたとしても、このメッセージは、少なくとも1つは出てくるようです。これは、一行が長すぎて右余白にはみ出してしまったという警告であり、L^AT_EX が適切な改行方法を見つけられなかった時に生じます（タイプライタ体では自動改行を許されないため頻発することが知られています）。ほとんどの場合、最終出力を見ても判別できないくらい（せいぜい1ポイントか2ポイント）です。しかしながら、時には目に見えて余白にはみ出してしまうこともあるので、その場合には、最終出力では修正したいと思うことでしょう。

全体的に overfull 行の数をできるだけ少なくする方法はいくつかあります。文書▷設定▷フォントのマイクロタイポグラフィ拡張を有効にするをクリックすると、大幅な改善が見込めます。さらに、文書▷設定▷プレアンブルに以下のような L^AT_EX コードを書いても、overfull 行の数は、通常、大幅に減少します（この文書のプレアンブルにも使用しています）。

³正確にいうと、通常は、です。ログファイルの分析は難しい仕事ですので、LyX はつねに正しい行に連れて行ってくれるとは限りません。ときには、L^AT_EX がある行にエラーがあると報告したとしても、実際のエラーはそれよりも前であることもあります。これは、プログラムで閉じ括弧を忘れたときに似ていなくもありません。エラーは表示されますが、実際にエラーのある場所よりも後ろでエラーが報告されるのです。

```

\tolerance 1414
\hbadness 1414
\emergencystretch 1.5em
\hfuzz 0.3pt

```

しかしながら、場合によっては、改行が正しく行われるように、文章そのものを書き換えなくてはならない場合もあります⁴。

何れにしても、少なくとも文書の完成版では、各行の確認を行うべきです。はみ出過ぎているものがあれば、目で見ればわかるでしょう⁵。

- Underfull \hbox

これは上の警告と違ってあまり出てきません。これも、L^AT_EX が適切な改行方法を見つけられずに、間延びした行を生成せざるを得なかった時に起こります。overfull 行と同様、外見上の問題がないかどうか確認してください。

- Overfull \vbox と Underfull \vbox

ページ区切りに問題があったという警告です。この場合も、出力を目で確認してください。何か問題があれば、見てすぐわかります。

- LaTeX Error: File 'Xxxx' not found

ファイル「Xxxx」が、システムに導入されていません。これは通常、文書が必要としている L^AT_EX パッケージに、導入済でないものがある場合に現れます。プリアンブルに変更を加えておらず、`\usepackage{}` コマンドも使用していないとすれば、LyX が読み込もうしているパッケージに存在しないものがあるということです。ヘルプ▷L^AT_EX の設定を使用して、LyX が認識しているパッケージの一覧を得てください。このファイルは、(ツール▷再設定を使用して) LyX の再設定を行うたびに更新され、どのパッケージが検出されたか、それらのパッケージが何をするものかを表示します。

`\usepackage{}` コマンドを使っていて、肝心のパッケージが導入されていない場合には、ご自身で導入する必要があります。

- LaTeX Error: Unknown option

このように始まるエラーメッセージは、パッケージに誤ったあるいは未定義のオプションが指定されていることを伝えようとするものです。パッケージの取扱説明書をご確認ください。

⁴詳細は、<http://www.tex.ac.uk/FAQ-overfull.html> を参照してください。

⁵文書▷設定で「draft」オプションを有効にすることもできます。すると、overfull 行の余白には、L^AT_EX が黒い四角を描画してくれます。

- Undefined control sequence

お使いの文書に L^AT_EX コードを挿入した際、打ち間違いをしてしまうと、このようなメッセージが表示されます。あるいは、パッケージを読み込ませるのを忘れたのかもしれませんが。いずれにしても、このメッセージは、あなたが未定義のコマンドを使用したことを示しています。

これ以外にも、エラーメッセージや警告メッセージがあります。一般に、読んで意味がすぐにわかるものは、L^AT_EX のメッセージです。何を言っているのかさっぱりわからないものもありますが、通常これらは、T_EX のメッセージであり、それが何を意味しているのか、どのように解釈すればいいのか、まったく手がかりがありません。誰もわからないのです。

エラーメッセージが表示された場合は、つねに以下のチェックリストにしたがった方がよいでしょう。

1. ご自分で挿入した L^AT_EX コードに打ち間違いがないかどうか確かめる。
2. 打ち間違いがなければ、コマンドを正しく使ったかどうか確かめる。
3. 文書のもっともはじめの部分にエラーボックスが多数積み上がった場合には—特に Missing `\begin{document}` が表示されているならば—、プリアンブルにエラーがあることを意味します。プリアンブルのデバックを行ってください。
4. プリアンブルに何も追加せず、文書に L^AT_EX コードをまったく挿入しなかった場合には、L^AT_EX 頒布版自体をまず疑ってください。導入されていないパッケージを探しだして、導入してください。
5. 導入されていないパッケージはなかったものとしましょう。LyX を微調整するオプションを使用していますか。特に、手動で非改行空白や改行、改頁などをたくさん挿入するなど、微調整オプションを悪用していませんか。適切な段落環境を使用する代わりに、これらを使用してその場をしのごうとしていませんか。
6. 微調整オプションはまったく使用しておらず、規則に乗っ取って作業していたものとしましょう。何か曲芸のようなことをしようとはしていませんか。数式や表中で、たとえば表のセルに図を入れるなど、おかしなことをしようとしていませんか。
7. L^AT_EX が改行位置を見つけられないほど長い一節がありませんか。既定では、改行するために単語間に空白をたくさん追加することを、L^AT_EX はかなり嫌います。可能であれば、問題が生じないように段落を修正するのが望ましいでしょう。

8. 入れ子の階層を深くしすぎていませんか. L_YXは、(今のところ) 環境の入れ子の階層が制限内にあるかチェックしません. かりに、環境を17段の入れ子にしたりしていたら、それが原因です (L^AT_EXの上限は5段です).
9. エラーメッセージは発生しないけれども、出力がひどいものとしましょう. 図表が幅が広すぎたり長すぎてページに収まらない場合には,
 - a) 図がページに合うように大きさを変える,
 - b) 表がページに合うように削る,などの手段を採る必要があります.
10. 他に出力におかしいところがあり、曲芸をやろうとしたり微調整オプションでしのごうとしたりはしていないとすれば、何がおかしいのか確定的なことは言えません.

これらすべてが役に立たないとすれば、ひょっとしたらL_YXのバグを発見したのかもしれない.....

3 文書クラス

『ユーザーの手引き』で解説したとおり、 \LaTeX はもともと article と report, book および letter の 4 つの標準文書クラスを提供していましたが、個人や組織、なかでも初期においては米国数学会が、文書クラスの幅の拡張に貢献してきました。本章では、外部的にメンテナンスされている主要なクラスについて、簡単に説明します。文書設定ダイアログの文書クラスプルダウンの一覧に「利用不能:」とマークされているものがありますが、これらを使用するためには、取扱説明書『カスタマイズ篇』の「文書クラスを新規に導入する」の章にある説明にしたがって導入する必要があります。

3.1 コレクション

3.1.1 AMS \LaTeX (米国数学会)

\LaTeX がサポートする文書クラス article (AMS) と book (AMS) は、学会によってメンテナンスされています。これらの機能の使用法については、数式篇説明書と、学会ウェブサイト <http://www.ams.org/publications/authors/tex/amslatex> に説明があります。以下の要約は、当初 DAVID JOHNSON によって提供され、RICHARD KIMBERLY HECK と \LaTeX チームが改訂したものです。

AMS \LaTeX レイアウトは、American Mathematical Society (アメリカ数学会) の刊行物に投稿する数学論文用のスタイルです。レイアウトは、特定の学術誌向けに仕立てられてはいませんが、簡単にそうすることができます。各学術誌毎の指示については、AMS の説明書をご参照ください (通常は、 \TeX 出力中の一行を変更する必要があるだけです)。AMS の説明書は、ウェブ Web 上の <http://www.ams.org> か、FTP で <ftp://ftp.ams.org/pub/tex/amslatex/> から入手することができます。これらのレイアウトは、数学的な文章を書くのであれば、どんなときでも適切かつ有用であるはずで

基本となる AMS \LaTeX レイアウトには、以下の三つがあります。

3 文書クラス

- amsart：標準 AMS article 書式. ファイル▷用例を開く▷論文▷American Mathematical Society (AMS) を参照.
- amsbook：標準 AMS book (実際はモノグラフ) 書式. ファイル▷用例を開く▷書籍▷American Mathematical Society (AMS) を参照.
- amsproc：標準 AMS 学会報書式.

L_AT_EX は、これらの最初の2つをネイティブにサポートしています. amsproc を利用したい場合には、取扱説明書『カスタマイズ篇』の「文書クラスを新規に導入する」の章を参照してください.

これらのレイアウトには、それ自身では、AMS クラスを使用するのに必要最低限のものしか含まれていません. 特に、定理や補題などを設定するのに使用される「theorem」環境は、何一つ含まれていません. これらは、代わりに AMS 定理モジュールに収録されており、AMS クラスのいずれかを選択すると、このモジュールが既定で読み込まれるようになっていきます (これは別のクラスでも使用することができ、別のものに変えたいときは取り除くこともできます). 少し使用頻度の低いものとしては、AMS 定理 (拡張) モジュールがあり、これは手動で読み込む必要があります.

既定では、定理類は、文書を通して連番が振られますが、これは標準定理 (節毎連番) モジュールを読み込めば変更することができ、book (AMS) を使用している場合には標準定理 (章毎連番) を用いても変更することができます. これらを用いると、出力は $n.m$ のようになり、最初の数字は節番号 (または章番号)、次の数字はそれまで同節内 (章内) に現れた出力の全数を表します. 大半の環境は、連番なしでも用いることができます. 連番なしのものは、環境名の後ろにアスタリスクを付けたもので表されます. 連番なしの出力のみが欲しい場合には、標準定理 (連番なし) モジュールを用います.

これらのモジュールは、AMS クラスで使用する必要はないことに注意してください. AMS 定理モジュールや上述の他のモジュールは、他のクラス、たとえば Article (標準クラス)・Report (標準クラス)・KOMA-script Book などでも問題なく使用することができます.

3.1.1.1 これらのレイアウトが提供するもの

これらのレイアウトが提供している環境を列挙すると、長い一覧となります. AMS- \LaTeX では、実のところ、「theorem」環境の派生型を際限なく定義していくことが可能です. しかし、AMS は、L_AT_EX で利用可能になっている環境のみを使用することを推奨しています.

article (AMS) と book (AMS) には, —— 節・書誌情報・タイトル・著者・日付のような標準環境の他に —— 以下の環境があります.

住所 これは著者の恒久的な住所です.

現在の住所 住所と別に指定されているときには, 著者の投稿時の仮初めの住所です.

電子メール 著者の電子メールアドレス

URL 著者のウェブアドレス. 任意.

キーワード 論文中で論じられているトピックを識別するためのキーワードやキーワード.

分野分類 *Mathematical Reviews* に説明が公表されている AMS Subject Classifications の分類番号を指します. 上述した AMS のウェブサイトでも, オンラインで見ることができます.

謝辞

献呈

翻訳者

以下の環境は, 標準定理モジュールと AMS 定理モジュールの両方で提供され, 後者の場合には, 星付き (連番なし) 版と星なし (連番) 版の両方が提供されます. 標準定理 (連番なし) モジュールでは, 同環境の星付き版だけが提供されます.

定理 1. これは, 通常, 得られた主要な結果を叙述するのに使用されます.

系. これは直前の叙述から比較的直接に導き出せる結果を叙述するのに使います. これも主要な結果であることがあります.

補題 2. 他の叙述を証明するために必要とされる, 比較的小さな結果です.

命題 3. 議論されている一般的な理論に (希望的には) 何かを付け加えるような, 主要ではない結果です.

予想 4. 正当化を与えずに述べられた叙述で, 著者には証明法がわからないけれども, 正しいと (少なくとも著者にとっては) 思われるものです.

定義. これは自明でしょう. この環境のフォントは, 前述の環境とは異なるフォントが使用されます.

3 文書クラス

例. 証明された結果の例を叙述するのに使用されます。

問題 5. これが何を指すかは必ずしも特定されていません。ご自分で解明してください。

演習. この環境の説明を記述せよ。

註釈 6. この環境も theorem 型、通常は重要性の低い所見に用いられます。

主張. より大きな結果の証明を与える途中で、しばしば用いられます。

ケース 1. 一般的に、これらはある条件の特定の場合を取り上げて、長い議論を区切るのに用いられます。

ケース 2. ケースの連番の枠組みは、それ自身で独立していて、他の連番の叙述とは区別されます。

証明. この環境の最後には、証了記号（通常は四角ですがスタイルによって変わります）が置かれます。この環境の中に別の環境を入れたい——たとえばケース環境など——場合で、証了記号は証明環境の終わりにのみ現れるようにしたい場合には、他の環境は証明環境の中に入れ子にしなければなりません。入れ子に関する情報については、『ユーザーの手引き』の「環境を入れ子にする」の節をご覧ください。 □

事実 7. 命題と同様の用いられ方をしますが、だいたい長くないものに使われるようです。

そして、以下は AMS 定理 (拡張) で提供されるものです。

基準. 必要とされる条件です。

アルゴリズム. 使用する一般的手順です。

公理. 議論されている系内部で真と仮定される性質や命題です。

条件. 議論の現在の文脈で仮定されている条件を述べるのに、よく用いられます。

註釈. 注記と同様です。

記法. 表記法の説明に用いられます。

要約 8. 本当に説明する必要がありますか？

謝辞. 謝辞 (acknowledgement) です。

結論. 長い議論の終わりに時々用いられます。

仮定. 仮定

問. 問い

数式モジュールには、(...種類別連番)の形のオプションで提供されているものを含み、他に10個のモジュールがあります。

さらに、AMSクラスは、AMS L^AT_EX パッケージと AMS フォントパッケージを自動的に提供します。これらの環境を使用するためには、お使いのシステム上で利用可能となっている必要があります。

3.1.2 追加フォント寸法

「追加フォント寸法」コレクションは、Article(フォント寸法を追加した標準クラス)と Book(フォント寸法を追加した標準クラス)、Letter(フォント寸法を追加した標準クラス)、Report(フォント寸法を追加した標準クラス)の各文書クラスを提供しています。これらは、それぞれ `article.cls` および `book.cls`, `letter.cls`, `report.cls` の各文書クラスを使用しますが、文書▷設定▷フォント ダイアログにある基本寸法の選択肢として 8・9・14・17・20 が追加されます。

3.1.3 ヘブライ語

文書クラスのヘブライ語 Article とヘブライ語 Letter は、`article.cls` と `letter.cls` 文書クラスを用いて、L^AX でヘブライ語の利用を可能とします。

3.1.4 日本語(標準クラス)

L^AX には、日本語文書の組版を可能にするためのいくつかのシリーズが収められています。これらのシリーズは、異なる時期に開発されたため、サポートする組版エンジンや機能が異なります。このコレクションは最も古いものです。これは、横書きと縦書きの日本の組版要求に合うように、L^AT_EX 標準クラスを調整したものです。このコレクションには、日本語 Article(標準クラス)、日本語 Article(標準クラス:縦書き)、日本語 Book(標準クラス)、日本語 Book(標準クラス:縦書き)、日本語 Report(標準クラス)、日本語 Report(標準クラス:縦書き)が収録されており、すべて日本語用に仕立てられた「古典的」エンジンである pL^AT_EX と upL^AT_EX で動作します。

より近代的な LuaL^AT_EX 組版エンジンで使用するために、コレクションには、`luatexja` L^AT_EX パッケージによって提供される、専用の日本語 Article(LuaTeX 用標準クラス)、日本語 Article(LuaTeX 用標準クラス:縦書き)、日本語 Book(LuaTeX 用標準クラス)、

3 文書クラス

日本語 Book (LuaTeX 用標準クラス：縦書き), 日本語 Report (LuaTeX 用標準クラス), 日本語 Report (LuaTeX 用標準クラス：縦書き) が納められています。

Lua \LaTeX クラス用のひな型が, ファイル▷新規 (ひな型使用) の論文・書籍・レポートの各カテゴリにあります。

3.1.5 日本語 (JS シリーズ)

このコレクションには, 日本語 (標準クラス) に収録されているクラスの改善版が収められています (第 3.1.4 節参照)。これには, 日本語 Article (JS シリーズ) と日本語 Book (JS シリーズ), 日本語 Report (JS シリーズ) が収録されています。Report クラスは, 日本語 Book (JS シリーズ) を使用して, 文書▷設定▷文書クラス▷クラスオプション▷詳細設定で report オプションを指定してください。これらのクラスは, すべて日本語向けの「古典的」エンジンである p \LaTeX と up \LaTeX で動作します。

Lua \LaTeX クラス用のひな型が, ファイル▷新規 (ひな型使用) の論文・書籍・レポートの各カテゴリにあります。

3.1.6 日本語 (BX シリーズ)

このシリーズには, 日本語文書のためのもう一つの文書クラスである, 日本語 Article (BX シリーズ) 文書クラス・日本語 Book (BX シリーズ) 文書クラス・日本語 Report (BX シリーズ) 文書クラスが含まれています。さらに, プレゼンテーション・スライド用の日本語 Slide (BX シリーズ) 文書クラスも収録されています。素の状態では p \LaTeX と up \LaTeX のみをサポートする標準クラスや JS シリーズと違って (第 3.1.4 節および第 3.1.5 節参照; Lua \LaTeX は luatexja \LaTeX パッケージで追加されるクラスのみを通じてサポート), このシリーズのクラスは, 日本語組版能力を提供する適切なパッケージを使用して, pdf \LaTeX ・Xe \LaTeX ・Lua \LaTeX も直接サポートします。

3.1.7 日本語 (JLReq クラス)

このシリーズは, World Wide Web コンソーシアム (W3C, <https://www.w3.org/TR/jlreq/?lang=en> を参照) が設定した「日本語組版処理の要件」の標準にしたがった jlreq クラスに依拠しています。p \LaTeX と up \LaTeX だけでなく, Lua \LaTeX でも使用できます。

L \LaTeX では, jlreq \LaTeX クラスにクラスオプションを指定して構成された日本語 Article (JLReq クラス)・日本語 Book (JLReq クラス)・日本語 Report (JLReq クラス) の各レイアウトを提供しています。

用例 Article 文書が、ファイル▷用例を開く▷論文▷日本語 Article (JLReq クラス) にあり、英語版解説文書が <http://mirrors.ctan.org/macros/jptex/latex/jlreq/jlreq.pdf> にあります (日本語版は <https://ftp.kddilabs.jp/CTAN/macros/jptex/latex/jlreq/jlreq-ja.pdf>) .

3.1.8 Koma-Script

BERND RELLERMEYER 原文 ; J URGEN SPITZM ULLER および L^AT_EX チームにより改訂

3.1.8.1 概要

KOMA-Script 文書クラスコレクションは、1990 年代初めに Frank Neukam によって作られたスクリプトクラスの成果です。公式には、1994 年に letter クラスの追加を以て開始されました。Frank Neukam のクラスは、植字の原則によって触発され、それは KOMA-Script クラスのデザインに完全に統合されてきました。

L^AT_EX の文書クラス KOMA-Script Article • KOMA-Script Report • KOMA-Script Book • KOMA-Script Letter は、それぞれ Koma-Script ファミリーの L^AT_EX 文書クラス `scrartcl.cls` • `scrreprt.cls` • `scrbook.cls` • `scrletter.cls` に対応しています。これらは、標準文書クラス `article.cls` • `report.cls` • `book.cls` • `letter.cls` の代替物であり、これらよりも多くの点でヨーロッパに於ける活版印刷の慣例に合致します。

- KOMA-Script クラスを選択した際の基本文字寸法は、KOMA-Script Article • KOMA-Script Report • KOMA-Script Book で 11pt, KOMA-Script Letter で 12pt です。
- 見出しや箇条書き (記述) 環境のラベル、および KOMA-Script Letter 文書クラスの要素の多くでは、ボールド体サンセリフフォントが使われます¹。章見出しの連番は、節見出しの連番と同様に行われ、別行立てで「Chapter...」という行が付きません。さらに、多くのオプションによって、見出しの外観を修正することができます (L^AT_EX では、文書▷設定ダイアログのクラスオプションフィールドで指定します)。これらのオプションの独語での詳細な説明が、Koma-Script の取扱説明書 *scrguide* にあります。

¹ボールド体サンセリフフォントには、古い cm フォントと新しい ec フォントの間に大きな違いがあり、特に見出しの表示の違いが際立ちます。比較すると、ec ボールドサンセリフフォントの方が、少し細く見えます。ec フォントを使用する際に「通常の」外観が欲しい場合には、WALTER SCHMIDT 作の L^AT_EX パッケージ `cmsd.sty` が役立ちます。

3 文書クラス

- KOMA-Script のページレイアウトは、ページを帯状に分割するか、円を描くかの2つの伝統的な植字慣習のいずれかに従います。既定では、ページが水平及び垂直方向の帯に分割されていると考え、帯を余白に割り当て、残りの使われていない部分を本文領域に割り当てます。余白の大きさは、ページを帯に分割する因数を変更することによって、変えることができます。A4の既定値は9で、この因数を増加させると、より狭い帯になるので、余白が狭くなります。

因数がどのような値であっても、2つの帯が両面文書の外側と下部の余白に割り当てられ、1つの帯が内側と上部の余白に割り当てられ、本文領域がページの真ん中の周りに残されます。ほとんどの両面文書は綴られるので、これは内側の余白が狭すぎるようにしてしまう可能性があります。そこで、帯の大きさが計算される前に、ページの綴り補正分を差し引くことができます。

Koma-Script 文書クラスで、本文領域をデザインする主な方法は、文書▷設定▷文書クラス▷クラスオプション▷詳細設定にBCORオプションまたはDIVオプションを入力する方法です。

この文書では、綴り補正は7.5mmであり、クラスオプション▷詳細設定ダイアログにBCOR7.5mmと入力しています。たとえば、因数を11に変えるには、詳細設定クラスオプションにコンマ区切りの項目リストにDIV=11を書き加えます。A4以外の用紙寸法におけるこの因数の既定値がほしければ、DIV=calcと書き加えます。

ページレイアウトの計算に円方式を採用したい場合には、代わりにDIV=classicとしてください。

- Koma-Script ファミリーのL^AT_EX 文書クラスには、多くの追加コマンドが定義されています。それらのうち、L^AT_EX で意味を持つものは、対応する段落環境として実装されています。

Koma-Script ファミリーのL^AT_EX 文書クラスに関する独語の詳細な説明が、Koma-Script 取扱説明書 *scrguide* にあります²。以下の各節では、L^AT_EX に関係する側面のみ説明致します。

3.1.8.2 KOMA-Script Article • KOMA-Script Report • KOMA-Script Book

文書クラス *KOMA-Script Article* • *KOMA-Script Report* • *KOMA-Script Book* は、それぞれレイアウトファイル `scrartcl.layout` • `scrreprt.layout` • `crbook.layout` によって実装されています。これらは、対応する標準文書クラス *article* • *report* • *book*

²英語翻訳版の *scrnggu* もありますが、完訳ではありません。

の段落環境を、若干の修正を加えた上ですべて含んでいます。L^AT_EX 固有の箇条書き (一覧) 環境は例外であり、同じ機能を持つ新しいラベリング環境で置き換えられています。このラベリング環境以外にも、新しい段落環境が多く追加されています。これらは、*letter (koma-script)* には含まれていないのでご注意ください。

- 部 (addpart)・章 (addchap)・節 (addsec)：これらは、それぞれ部*・章*・節*と同等の働きをしますが、加えて目次に項目が表示されるようになります。部 (addpart) と章 (addchap) は、*article (koma-script)* には収録されていません。
- 章 (addchap*)・節 (addsec*)：これらは、章 (addchap) と節 (addsec) とまったく同等に動作しますが、加えてランニングヘッダを取り除きます。章 (addchap*) は、*article (koma-script)* には収録されていません³。
- 小見出し (minisec)：これは文書構造に現れない形で、続く段落の直上に標準文字寸法の見出しを付けます。
- 上部キャプションと下部キャプションは、ある要素の上部ないし下部にキャプションを配置する際、位置に応じて正しく余白取りを行う特別なキャプションです (組版規則に厳密に従うならば、表キャプションは、つねに表の上に置くべきでしょう)。また、クラスオプション `tablecaptionsabove` を用いれば、キャプションを指定すると、表には上部キャプションが、図には下部キャプションが適用されるようになります。これを使用するためには、Koma-Script 第 2.8q 版以上が必要です。
- 格言 (dictum)：これは、章頭などに警句を置くのに用いることができます。オプション引数 (挿入▷格言著者) を用いれば、格言の作者を入れることができます。格言と作者の間は、線で区切られます。これを使用するためには、Koma-Script 第 2.8q 版以上が必要です。格言 (dictum) は、*article (koma-script)* には収録されていません。

以下の各環境は、標準のタイトル・著者・日付とともに文書のタイトル領域を形成します。これらは、最初の「通常の」段落よりも前に置かれる必要があります⁴。これらの環境が 2 度以上使用された場合には、後に使用されたものが前のものを上書きします。つまり最後に使用したもののみが有効になります。しかしながら、タイトルや著者、日付のような異なる環境の順序を変更しても、生成された文書には何の効果も生じません。

³*book (koma-script)* と *report (koma-script)* には、`\addpart*` コマンドもありますが、これは Part* と同一なので、L^AT_EX では実装されていません。

⁴つまり、対応する L^AT_EX コマンドが `\maketitle` コマンドよりも前に現れなくてはなりません。

3 文書クラス

- 主題：通常のタイトル部（タイトル・著者・日付）の上に、文書の主題用の中央揃えの段落を生成します。
- 出版社：通常のタイトル部（タイトル・著者・日付）の下に、出版社名用の中央揃えの段落を生成します。
- 献呈：*report (koma-script)* と *book (koma-script)* では、タイトル頁の次に、献辞用の中央揃えの段落を独立した頁として生成します。*article (koma-script)* の場合は、通常のタイトル部（タイトル・著者・日付）の下に献辞用の中央揃えの段落を生成します。
- タイトル頭書き：通常のタイトル部（タイトル・著者・日付）の上に、文書頭書き用の左揃えの段落を生成します。
- 扉裏上部：*report (koma-script)* や *book (koma-script)* の両面印刷において、扉裏上部に左揃え段落を生成します。片面印刷や、*article (koma-script)* では何の効果も生じません。
- 扉裏下部：*report (koma-script)* や *book (koma-script)* の両面印刷において、扉裏下部に左揃え段落を生成します。片面印刷や、*article (koma-script)* では何の効果も生じません。
- 追加タイトル：実際の文書の前に、整形を行わない段落から成る、特殊な「汚い」ページを生成します。

KOMA-Script は、文書▷設定▷文書クラス▷クラスオプション▷詳細設定ダイアログにおいて、幅広い詳細設定クラスオプションを提供しており、文書全体に適用することができます。主要なものを挙げると、

draft=true 組版エラーの起こった全ての行末に小さな黒い四角を付記した PDF を生成します

headings= big/normal/small の値を取り、big と small は、既定値がふさわしくない場合に、ページ寸法を考慮に入れながら見出しの大きさを調整します

numbers= auto(既定値)/enddot/noenddot の値を取り、auto は章や節番号等の後にピリオドを付けるかどうかを KOMA-Script に委ね、enddot は強制的にピリオドを付け、noenddot はピリオドを抑制します

また数式オプションとしては

leqno 数式番号を右ではなく左側に付けます

fleqn 数式を中央揃えではなく左揃えにします

KOMA-Script は、標準 L^AT_EX コマンドやパッケージとは別に、文書の出力に影響するコマンドも提供します。たとえば、キャプションをボールド体にしたければ、文書▷設定▷L^AT_EX プリアンブルに

```
\setkomafont{captionlabel}{\bfseries}
```

と書き加えれば実現できます。

書籍を執筆中であれば、かりそめのページ番号をローマ数字で入れ、この部分の章に番号を振らないことは普通にあります。章*環境を用いてしまうと、巻頭言と序文は目次に現れません。目次に現れるようにするには、章環境を用いて、書籍の先頭に (L^AT_EX プリアンブルではなく) T_EX コードで

```
\frontmatter
```

と書き、本文の第 1 章の前に

```
\mainmatter
```

と書きます。あるいは、章* (TOC) を使うこともできます。

また、巻末の付録の後に、付番していない章環境が欲しいことがあるかもしれません⁵。章*環境を用いてしまうと、それらは目次には現れませんので、これらの最初の見出しの前に T_EX コードで

```
\backmatter
```

と書き、章環境を用います。L^AT_EX での付番は、これらのコマンドを使っても通常通り為されますが、PDF 出力においては、見出しが付番なしで現れ、目次にも表示されません。

文書クラス *article (koma-script)*・*report (koma-script)*・*book (koma-script)* の各レイアウトファイルは、`scrmacros.inc` ファイルを読み込みます。このファイルは、ユーザ環境を定義する場所として想定されています。 `scrmacros.inc` を、個人用レイアウトディレクトリにコピーして編集してください！

3.1.8.3 新しい letter クラス：KOMA-Script Letter (KOMA-Script V.2)

by JÜRGEN SPITZMÜLLER

Koma-Script 第 2.8 版では、新しい letter クラス `scrlettr2` が導入されたため、`scrlettr` は廃止されてサポート対象外となりました。これは—L^AT_EX 側で—全く新しいインターフェイスを持っており、旧クラスとは互換性がありません。した

⁵付録は、通常、文字を使って「付番」されます。

3 文書クラス

がって、LyX は両者をサポートしていますが、新しい方のクラスを使用することを推奨します。

このクラスは、*letter (koma-script)* と同一の機能に加えて、新しい機能をいくつかサポートしています。基本となる項目は、住所（受取人の住所、旧レイアウトの書簡と同じ）・頭語・結語です。次の住所は別の書簡を新しく開始します（つまり一文書あたり複数の書簡を書くことができます）。新しく付け加えられた項目は、送り主の電子メール・URL・ファックス・銀行、およびヘッダに（挿入▷図で）ロゴを入れることができるようになった点です。

しかしながら、もっとも大きな改善は、書簡のレイアウトを、ほとんどどんなニーズにも合わせて設定できるようになった点でしょう。これはプリアンブルか、クラスオプションとして読み込まれる特別なスタイルファイル（「レタークラスオプション」、拡張子*.lco）によって設定することができます⁶。用例としては、ファイル▷新規（ひな型使用）...▷書簡▷KOMA-Script Letter (V.2) のひな型をご覧ください。詳細な解説は、Koma-Script 取扱説明書 (*scrguide*) にあります。

3.1.8.4 問題点

Koma-Script 文書クラスを LyX 上に表示する上で、LyX の内部処理が若干の問題を生じます。

- 章環境の章番号は、章見出しと同じ行ではなく、その上に表示されます。これは、レイアウトファイル中の Counter_Chapter ラベル型に対する LyX の内部処理が原因です。
- 章 (addchap) と節 (addsec) 環境の見出しは、「本当の」L^AT_EX の目次には追加されますが、LyX の目次（移動メニュー）には追加されません。
- *letter* 文書クラスでの段落は、垂直スペースによる段落区切りで表示され、字下げはされません。これが標準的な挙動となっていて、特別な L^AT_EX コマンドは必要ありません。しかし、文書▷設定ダイアログでは、対応するラジオボタンは字下げに設定されています。垂直スペースは、隙間を空けるために、文書に L^AT_EX コマンドをつねに追加するようになっていますが、この文書クラスの場合には、そのような挙動が望まれないためです。

⁶KOMA パッケージには、既定の*.lco ファイルがいくつか収録されています。たとえば、ドイツの組版規則に則った DIN.lco や、旧 scrletter クラスの既定レイアウトを提供する KOMAold.lco などです。たとえば後者は、文書▷設定▷文書クラス▷クラスオプションフィールドにクラスオプション KOMAold を加えることで読み込むことができます。

3.1.9 ポーランド語 M.W. コレクション

TOMASZ LUCZAK 文

L^AT_EX 文書クラスポーランド語 Article (MW Bundle)・ポーランド語 Report (MW Bundle)・ポーランド語 Book (MW Bundle) は、それぞれ L^AT_EX 文書クラス `mwart.cls`・`mwrep.cls`・`mwbk.cls` に対応しています。これらは、標準の `article.cls`・`report.cls`・`book.cls` を代替するものであり、多くの点で、ポーランド語の組版慣習によく適合しています。

基本的な違いは、次の点です。

- 付番のない見出し（節*などの星付き）も目次に加えられます。

- 以下のページスタイルの追加

uheadings 分離線付きヘッダ

myheadings `\markright` コマンドおよび `\markboth` コマンドで設定するカスタムヘッダ

myuheadings 分離線付きカスタムヘッダ

outer ページの外側にページ番号

- オプション

rmheadings セリフ体見出し — 既定

sfheadings サンセリフ体見出し

authortitle タイトルページは、著者名・タイトルの順で配置 — 既定

titleauthor タイトルページは、タイトル・著者名の順で配置

withmarginpar ページ上に余白のための場所を確保

3.1.10 Tufte コレクション

文書クラスの Tufte Book と Tufte ハンドアウトは、`tufte-booke.cls` と `tufte-handout.cls` 文書クラスを用います。これらの文書クラスの詳細は、ファイル▷用例を開く▷書籍▷Tufte Book と ファイル▷用例を開く▷ハンドアウト▷Tufte ハンドアウトにあります。

3.2 論文

コレクション中の文書クラスも参照してください (第 3.1 節参照)。

3 文書クラス

3.2.1 Astronomy & Astrophysics

PETER SÜTTERLIN 原文；L_YX チーム改訂

3.2.1.1 はじめに

この節では、文書クラス `aa.cls`(これについての情報は <https://www.aanda.org/for-authors/latex-issues/technical-background-information> にあります) 第 9.1 版を使用して科学雑誌 *Astronomy and Astrophysics* (『天文学と宇宙物理学』) に投稿する論文を執筆するための L_YX の使用法を説明します (`aa-package`).

このパッケージには取扱説明書がついていますので、取扱説明書『カスタマイズ篇』の「文書クラスを新規に導入する」の節にしたがってこの文書クラスを導入する前に参照してください。

この学術誌の出版元は、2001 年 1 月 1 日以降、Springer 社から EDP Science 社に変更されていることにご注意ください。この出版社の変更によって、`thesaurus` コマンドの廃止など、スタイルファイルに若干の変更が加わっています。もし古い版が導入されている場合には、更新してください。古い (第 4 版) レイアウトに対する互換性については、`LYXDir/layouts/aapaper.layout` のコメントをご参照ください。

3.2.1.2 最初の一步

ファイル▷新規(ひな型使用)▷論文▷Astronomy & Astrophysics メニューにあるひな型を土台として、書き始めることをお勧めします。ひな型を使用しない場合には、以下の設定にしてください。

- 文書▷設定▷文書クラスダイアログで Astronomy & Astrophysics を選択してください (まあ、これは自明ですね)。
- ページレイアウト ダイアログのページ様式オプションは変更しないで、既定値のままに設定しておいてください。全体のレイアウトはマクロが行うので、何も変更してはいけません。

3.2.1.3 ヘッダ部

最初に入力すべきものは、ヘッダ情報です。ヘッダ情報には、以下の 7 つの項目があり、うち 1 つは必須ではありません。

- タイトル：[必須]
- サブタイトル：[非必須]

- 著者：[必須]
- 住所：[必須]
- 抜き刷り：[非必須] 連絡と抜き刷りを受け取る人を指定
- 郵便：[非必須] 連絡用の郵便住所
- 日付：[必須] 推奨されるフォーマットは、Received: <日付>; Accepted <日付>です。

`\maketitle` コマンドは、`LATEX` がヘッダ終了時に自動的に発行するので、手動で発行する必要はありません。各項目の順序は関係ありませんが、見た目のためと、実際の文書のレイアウトに合わせるために、上記の順序を維持しておくとい良いでしょう。

現住所の付記などのために、ヘッダ部に脚注を挿入したい場合には、挿入▷脚注を使用して標準の脚注を挿入してください。この場合には、`LATEX` は自動的に`\thanks{}`を使用します。

編集▷特別文辞様式に表題付けに関連した2つの差込枠があります。

- 所属機関は、連絡担当著者と所属機関の組をマークします。所属機関には、住所フィールドに現れるのと同じ順序で連番が振られますので、各著者に標識を付けてください。
- 電子メールは、至急連絡用の電子メールアドレスを付するものです。

これら以外に、このマクロは、`LATEX` が対応していない以下の`LaTeX` コマンドを追加的に使用できます。

- 複数の著者名あるいは所属名を区切るための`\and`

`LATEX` 中では`TeX` コードとして適切なコマンドを入力する必要があります。ひな型中の用例とファイル▷新規(ひな型使用)▷論文▷Astronomy & Astrophysicsにある用例を参照のこと。

3.2.1.4 概要

概要はヘッダ部の直後に置きます。第5版から概要環境はコマンドに変更され、1段落だけに限るといった制限が付きまして。加えて、キーワード項目が含まれていなくてはなりません。これは、段落様式キーワードで為されます。用例論文を参照してください。

3 文書クラス

3.2.1.5 サポートされる環境

A&A 論文用レイアウトでは、文章を構造化するために、以下の環境がサポートされています。

- 標準
- 節
- 小節
- 小々節
- 箇条書き (記号)
- 箇条書き (連番)
- 箇条書き (記述)
- キャプション
- 概要
- 謝辞 (Acknowledgment)
- 書誌情報
- LaTeX

3.2.1.6 LyX でサポートされていないコマンド

LyX の Astronomy & Astrophysics レイアウトでは、まだサポートされていないコマンドがあります。そのうちいくつかは既に紹介しましたが、網羅するために、以下にすべて列挙しておきます。

- `\and`
- `\authorrunning`
- `\object{}`
- `\titlerunning{}`

これらのコマンドのいずれかを使いたい場合には、自分で入力する必要があります。挿入▷`TeX` コードを使用することを忘れないで下さい！

3.2.1.7 図フロートおよび表フロート

LyX は、必要なフロート環境 `figure`・`figure*`・`table`・`table*`をサポートしていますので、ここでは詳しい説明はしません。ユーザーの手引きをご覧ください。ただ、表は左揃えにしなくてはならないことに注意して下さい。そうするためには、表を選択して、編集▷段落設定で配置を変更してください。

一つだけ、特別なものがあります。それは、図横にキャプションを配置する図です。この図を作成するには、以下のようにしなくてはなりません。

1. 挿入▷フロート▷図を使って、幅広の図フロートを作成します。それから、フロート内で編集▷フロートの設定メニューを開き、フロートダイアログで段をまたぐを有効にします。
2. キャプション文を入力します。
3. キャプション上にカーソルを移動し、Return を押します。
4. 図を挿入します。
5. カーソルを図の後ろに置いて、挿入▷整形▷横方向の空白を選択して空白ダイアログを開き、空白コンテキストメニューから水平フィルを選択します。
6. M-c t と押して L^AT_EX モードに切り替えます。
7. T_EX 差込枠中に `\parbox[b]{55mm}{` と入力します。ここでは波括弧を閉じないでください。
8. カーソルをキャプション文の後ろに移動し、M-c t } と押して、T_EX モードで閉じ括弧を入力します。

ファイル▷用例を開く▷論文▷Astronomy & Astrophysicsにあるファイルの図も参照してください。

3.2.1.8 Referee レイアウト

投稿用には、論文を特別なダブルスペースレイアウトに整形しなくてはなりません。このためには、文書クラスに referee オプションを文書▷設定▷文書クラスダイアログの任意設定クラスオプションフィールドに入力する必要があります。

3.2.1.9 例示用論文

ファイル▷用例を開く▷論文▷Astronomy & Astrophysics は L_YX で書かれています。これは、[aa-package](#)にある、元のマクロパッケージ aa.dem から採用した例示用論文ですが、古い構造化されていない概要型を使用して L_YX に翻訳されたものです。これはヒントを得るために使用したり、aa.dem の L^AT_EX コードと L_YX での書き方の比較に使ったりしてください。

3.2.2 AAST_EX

MIKE RESSLER 文

3 文書クラス

3.2.2.1 はじめに

AAST_EX は、American Astronomical Society (アメリカ天文学会) が発行する *Astrophysical Journal* (宇宙物理学雑誌) 同誌の Letters と Supplement を含む)・*Astronomical Journal* (天文学雑誌)・*Publications of the Astronomical Society of the Pacific* (太平洋天文学会誌) 三誌への電子投稿を可能にするために、同学会が作成したマクロ集です。L_YX は、これらの文書を生成する上で、なかでも数式・引用・画像処理能力において、優れたツールであることが認識されています。L_YX は、同マクロ第 5.0 版以降を必要としており、できればここで説明されている第 6.0 版以降が望まれます。第 5.0 版未満の版は、L^AT_EX 2.09 で使用するよう企図されているため、根本的に L_YX とは互換性がありません。AAST_EX パッケージは、以下の AAST_EX ウェブサイトからダウンロードできるでしょう。

<https://aas.org/aastex/aastex-downloads>

パッケージ中には、完備したユーザーの手引きが収録されていますので、L_YX で論文を書き始める前に、その中身を完全に理解しておく必要があります。L_YX は、AAST_EX の全コマンドを理解する必然性をなくすものではなく、すべてを手入力する苦勞を減らすだけです。仕上がりの L^AT_EX 文書が、論文を投稿しようとしているジャーナルの要求事項に完全に合致していることを確認するのは、あなたの責任です。

3.2.2.2 新しい論文を書き始める

論文を書き始めるには、AAST_EX ひな型ファイルを土台に始めることを強くお勧めします。ファイル▷新規(ひな型使用)▷論文▷American Astronomical Society をクリックしてください。これには、論文原稿のほとんどに共通して使われるフィールドが書き込まれています。書き込まれている文字列(括弧<>を含む)を正しい情報で上書きしてください。AAST_EX コマンド・AAST_EX 環境の多くは、L_YX 中で直接実行することができますが、できないものもあります。このようなコマンドは、直接 L^AT_EX コードを入力して、それを T_EX コードとしてマークしなければなりません。このようなコマンドは、T_EX コードあるいは Evil Red Text (邪悪な赤い文)と呼ばれます。AAST_EX 文書に必要とされる T_EX コードを最小にするようには努力しました。

3.2.2.3 論文を書き終わったら

論文を満足する水準で書き終え、プレビューや印刷が正しく為されているとしても、まだ学術誌に投稿する前にしなくてはならない「後処理」が残されています。

1. 論文を L^AT_EX ファイルとして書き出します (ファイル▷書き出し▷LaTeX) .
2. 出力した .tex ファイルを好きなエディタで編集します.
 - a) `\documentclass` コマンドよりも前のコメント行を削除します.
 - b) `\usepackage...{fontenc}` という行があれば (通常は `\documentclass` の直後にあります) 削除します. また, `\secnumdepth` 行があれば, これも削除します.
 - c) L^AT_EX プリアンブル中に意図的に置いたコマンド (.tex ファイル中, 「User specified L^AT_EX commands」というコメントの直後にあります) を除いて, `\makeatletter` と `\makeatother` の間のもの (これらのコマンドを含む) をすべて削除します.
3. 編集後のファイルを L^AT_EX にかけて, 正しく処理されることを確認します.
4. 学術誌の要求事項を再読して, ファイル名と書式が正しいことを確認します.
5. 投稿してください.

3.2.2.4 特定のコマンドに関するコメント

AAS_TE_X ユーザーの手引き (aasguide.tex) に, 各 AAS_TE_X コマンドの詳しい説明があるので, ここでは, 各コマンドの使用法の詳細な説明は行いません. そこで, ここではユーザーの手引きに現れるとおりに一覧を列挙して, 必要なところだけコメントを加えることにしましょう.

期待通りに動作するもの これらは期待通りに動作するので, ここでは列挙するだけにして, コマンドの説明されている節を挙げておきます. `\documentclass` (2.1.1) ・ `\begin{document}` (2.2) ・ `\title` (2.3) ・ `\author` (2.3) ・ `\affil` (2.3) ・ `\abstract` (2.4) ・ `\keywords` (2.5) ・ `\section` (2.7) ・ `\subsection` (2.7) ・ `\subsubsection` (2.7) ・ `\paragraph` (2.7) ・ `\facility` (2.10) ・ `\begin{displaymath}` (2.12) ・ `\begin{equation}` (2.12) ・ `\begin{eqnarray}` (2.12) ・ `\begin{mathletters}` (2.12) ・ `\begin{thebibliography}` (2.13.1) ・ `\bibitem` (2.13.2) ・ 全引用コマンドとその派生 (2.13.2) ・ 一般的な graphicx 図コマンド (2.14.1) ・ `\begin{table}` (2.15.4) ・ `\begin{tabular}` (2.15.4) ・ `\caption` (2.15.4) ・ `\label` (他にもあるが特に 2.15.4) ・ `\tablerefs` (2.15.5) ・ `\tablecomments` (2.15.5) ・ `\url` (2.17.4) ・ `\end{document}` (2.18).

3 文書クラス

以下のスタイルオプションも正しく動作します。longabstract (2.4)・preprint (3.2.1)・preprint2 (3.2.2)・eqsecnum (3.3)・flushrt (3.4)。これらは、文書▷設定にある、クラスオプションの任意設定ボックスに入力してください。

動作するが、若干説明が必要なもの 以下のコマンドは動作しますが、若干の説明が必要です。

- 以下の項目は、学術誌の編集者向けに取ってあるものですが、どうしても必要ならば、L^AT_EX プリアンブルに入れることができます。received・revised・accepted・ccc・copyright (すべて 2.1.3)。
- 以下の項目は、L^AT_EX のプリアンブルに置くことができるもので、これらは、ひな型ファイルには、空の引数と共に含まれています。slugcomment(2.1.4)・shorttitle(2.1.5)・shortauthors(2.1.5)。
- email(2.3)—「単独で」使うことはできますが、段落の途中で使うことはできません。埋め込む必要がある場合には、T_EX コードを使用してください。
- and(2.3)—後ろに{}が付記されます。これはエラーの原因にはなりません。
- notetoeditor(2.6)—「単独で」使うことはできますが、段落の途中で使うことはできません。埋め込む必要がある場合には、T_EX コードを使用してください。
- placetable(2.8)—相互参照用タグを挿入することはできません。手動でタグ名を入力する必要があります。
- placefigure(2.8)—placetable と同様。
- acknowledgements(2.9)—後ろに{}が付記されます。これはエラーの原因にはなりません。
- appendix(2.11)—後ろに{}が付記されます。これはエラーの原因にはなりません。
- figcaption(2.14.2)—文章のはじめにカーソルを置いて挿入▷短縮タイトルを選択すれば、非必須のファイル名引数を挿入することができます。「短縮タイトル」とありますが、figcaptionが必要とする型の非必須引数を挿入します。望むらくはいつの日か改名されることでしょう。

- `\objectname(2.17.1)`—非必須パラメータとしてカタログ ID を取る以外は、`\figcaption` と同様.
- `\dataset(2.17.1)`—非必須パラメータとしてカタログ ID を取る以外は、`\figcaption` と同様.

実装されていないため、 $\text{T}_{\text{E}}\text{X}$ コードを使用するもの `\eqnum(2.12)`・`\setcounter{equation}(2.12)`・`学術誌名略語(2.12.4)`・`\figurenum(2.14.1)`・`\epsscale(2.14.1)`・`\plotone(2.14.1)`・`\plottwo(2.14.1)`・`\tablenum(2.15.4)`・`\tableline(2.15.4`: 入れたい場所の後ろに、左側セルの最初の要素として挿入する. 表中では、 $\text{L}_{\text{Y}}\text{X}$ の方法は使わないでください)・`\tablenotemark(2.15.5)`・`\tablenotetext(2.15.5)`・補遺(2.17)の大半(`\objectname`・`\dataset`・`\url`・`\email`を除く. 上記参照)・`\singlespace(3.1)`・`\doublespace(3.1)`・`\onecolumn(3.2)`・`\twocolumn(3.2)`

実装できないもの `references(2.12.3)` 環境と `deluxetable(2.14)` 環境は、少なくとも意味のあるような形で実装することができないので、無視してください. もし、絶対に、どうしても、`deluxetable` を使う必要があるならば、テキストエディタで別ファイルとして編集し、挿入▷子文書を使用して、 $\text{L}_{\text{Y}}\text{X}$ ファイルに読み込むことを推奨します. この実例は、ファイル▷新規(ひな型使用)▷論文▷American Astronomical Society をご参照ください.

3.2.2.5 FAQ とヒントと技とその他の考察

$\text{L}_{\text{Y}}\text{X}$ を $\text{AAST}_{\text{E}}\text{X}$ と協調させる $\text{L}_{\text{Y}}\text{X}$ に、新しいレイアウトと文書クラスを認識させるには若干コツが要ります. もしうまく行かなかったら、次のようにしてみてください.

1. $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ が、 $\text{AAST}_{\text{E}}\text{X}$ を検出できることを確認してください. $\text{AAST}_{\text{E}}\text{X}$ 頒布版に含まれる `sample.tex` (あるいは `table.tex` でも可) を、 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ や $\text{AAST}_{\text{E}}\text{X}$ とまったく関係のないディレクトリにコピーして、`sample.tex` を $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ にかけてください.
2. `aastex.layout` が、 $\text{L}_{\text{Y}}\text{X}$ の `layouts` フォルダにあることを確認してください.
3. $\text{L}_{\text{Y}}\text{X}$ 中から ツール▷環境構成を再度実行し、 $\text{L}_{\text{Y}}\text{X}$ を再起動してください.
4. ひな型ではなく、通常の新規ファイルを開いてください. American Astronomical Society ($\text{AAST}_{\text{E}}\text{X}$ V. 6) が、文書▷設定のクラス一覧に表示されていますか?

3 文書クラス

既存の AAST_EX 文書を処理する際に、AAST_EX レイアウトを見つけられないとか、「タイトルレイアウトと通常のレイアウトを混在させないでください」といった警告が出るようであれば、正しい導入が行われていません。

表を処理する際の L_AT_EX エラー 既定では、L_YX は、表のキャプションやタイトルを中央揃えにしようと試みます。これは AAST_EX と悪い相互干渉を発生させるようですので、キャプションやタイトルのどこかをクリックして、編集▷段落設定を選択し、配置を両端揃えに設定してください。私のところではこれで解決しました。

書誌情報 若干の注意があります。

1. 文献項目の本文中の空白取りが少しおかしくなることがあるようです。書誌情報を入力する際には、最後の著者と、著作年を入れる括弧の間にスペースを**入れ**ないようにしてください。たとえば、Ressler(1992) はいいですが、Ressler (1992) ではだめです。
2. 書誌情報の入力の仕方は、必ずしも自明ではありません。もっとも簡単な方法は、文書の最後に、最初の書誌情報をまず書き込み、それを書誌情報環境としてマークすることです。すると、入力したものの前に、小さな灰色のボックスが現れます。このボックスをクリックして、残りの情報を入力してください。新しい書誌情報を追加するには、既存の書誌情報の後ろでリターンを押します。すると、新しいボックスの付いた新規行が出てくるので、後は同様にしてください。

EPS ファイルの読み込み AAST_EX には、自前の図関連コマンド（たとえば `\plotone`）がありますが、私は（既定の `graphicx` を使った）L_AT_EX の標準図コマンドの方がずっと好きです。望めば、図フロートボックスに、T_EX コードで `\plotone` 等のコマンドを挿入することはできますが、私には正しいレイアウトを得ることができた試しがありません。標準の画像コマンドを使えば、L_YX は、`\usepackage{graphics}` コマンドを L_AT_EX プリアンブルに挿入し、図を L_AT_EX 2_ε の標準的な方法で取り扱って、文章の中にちりばめます。現在、ApJ は、ちょうどこのようにして作成した図を受け入れています。AJ はまだ「すべてを末尾に積み上げる」方法を使用しているかもしれません。

可能だったが、実装しなかったもの 「素敵な」ものの中には、実装することも可能だったけれども、あえて実装しなかったものがあります。たとえば、紙の原稿ではダブルスペースであったとしても、L_YX ウィンドウ中で、文章をダブルスペースにする

意味はないと考えました。また、preprint 様式と preprint2 様式を、独立したレイアウトにすることもしませんでした。いずれにしても、執筆時間のほとんどを、平の原稿モードで使うのがほとんどですから、これらの機能でディスクスペースを消費しないことにしました。

3.2.2.6 最後に

これまでに進んできた道のりは、長い人も短い人もいることでしょう。私には、ApJ と AJ に発表した論文がありますが、その努力の 98% は LyX 上で為されました。残りの 2% が、L^AT_EX での後処理と若干の後片付けです。投稿の過程では何のトラブルもなく、これらの学術誌が通常の投稿原稿と違う点に気づくことはなかったでしょう。ですから、がんばって発表してください！

3.2.3 チェス

チェス文書クラスは、チェスゲームの叙述を可能にするために、標準 article 文書クラスを lyxskak パッケージとともに使用します。ファイル▷用例を開く▷論文▷チェスを開くとゲーム 1 とゲーム 2 がありますので、それらを参照してください。

3.2.4 Elsevier

Elsevier 文書クラスは、Elsevier 出版社の学術誌用に、elsarticle L^AT_EX クラスのサポートを提供します。これは、以下の各環境を追加します：表題脚註・著者脚註・連絡担当著者・住所・電子メール・キーワード。ひな型はファイル▷新規(ひな型使用)▷論文▷Elsevier メニューで利用可能です。これは、さらに多くの環境を設定していて、必要な情報を全て含みます。しかしながら、ドキュメンテーションは [CTAN](#) でも利用可能です。

3.2.5 Paper

Paper (標準クラス) 文書クラスは Article (標準クラス) 文書クラスの代替となるものです。これは同様の機能を提供しますが、サンセリフ体の節や見出し等を持つこちらのレイアウトの方が気に入るかも知れません。

3.2.6 RevT_EX4

by AMIR KARGER

3 文書クラス

REVTeX (V. 4) テキストクラスは、アメリカ物理学会の RevTeX 4.0 (1999 年 5 月の β リリース) クラスで動作します。

L^AT_EX には、RevTeX 3.1 で動作する REVTeX (廃版) テキストクラスもあります。しかしながら、第 3.1 版は L^AT_EX 2.09 で動作するものであり、基本的に時代遅れなものになっています。L^AT_EX は L^AT_EX 2_ε を必要とすることから、第 3.1 版は L^AT_EX とうまく協調しないことを意味しますが、動作するようには調整してあります。RevTeX 4.0 は、L^AT_EX 2_ε で素直に動作するようにデザインされているので、L^AT_EX と REVTeX (V. 4) テキストクラスの組で使用するの、たいへん簡単です。

この取扱説明書は、RevTeX 4.0 の説明書の補完として使用されることを前提としています。したがって、ここでは RevTeX 固有マクロの説明はせず、必要な場合にはプリアンブルに何と書けば良いかを知っているものとして説明します。

3.2.6.1 導入

RevTeX 4 パッケージの README ファイルにあるとおり、ともかく RevTeX 4 を導入しなくてはなりません。パッケージは、The RevTeX 4 Web Site <http://publish.aps.org/revtex4/> から入手することができます。これを L^AT_EX が検出することのできる場所に導入してください。それから、適当に選んだディレクトリ (もといクラスファイルを導入したのとは異なるディレクトリ) で簡単な RevTeX 4 文書を L^AT_EX でコンパイルできるかどうか確認してください。この後、L^AT_EX を環境構成すると、クラスファイルを検出し、RevTeX4 テキストクラスが使用できるようになるはずですが。

使い始めるには、tex2lyx を使って RevTeX 4 文書を読み込むか、またはファイル▷新規 (ひな型使用)▷Article にある Revtex 4 のひな型を使うのが、おそらく一番簡単な方法でしょう。

3.2.6.2 プリアンブルに関すること

`\documentclass` に与える「preprint」や「aps」のような非必須引数は、他のクラス同様、文書▷設定▷文書クラスのクラスオプションフィールドに入れます。RevTeX では、少なくとも一つの非必須引数が必要なことに注意してください。

`\draft` など他のプリアンブルは、他のクラス同様、文書▷設定▷L^AT_EX プリアンブルに入れてください。

3.2.6.3 レイアウト

レイアウトは、基本的に RevTeX4.0 のコマンドに対応しています。たとえば、Email レイアウトは、`\email{}` に対応しています。(少なくとも RevTeX 4.0 Beta 現在では)

住所と所属はまったく同じものなので、両方を使う必要はありません⁷。

3.2.6.4 重要な注意

RevTeX 4 には、バグを引き起こしがちなために、気をつけなくてはならない特殊な面がいくつかあるため、LYX ではさらに気をつけなくてはなりません。

RevTeX では、`\thanks` コマンドは、`\author` コマンドの外側に来なくてはなりません。LYX では、独立した感謝レイアウトがこの働きをします。直接、著者レイアウトに脚注を付けてしまうと、おかしいことが起こる可能性があります。詳細については、RevTeX 4 の取扱説明書をご覧ください。

また、著者電子メール・著者 URL・感謝 (thanks) レイアウトは、著者レイアウトと住所 (または同等の所属) レイアウトの間に置かなければなりません。感謝 (thanks) を住所の後に置いてしまうと、L^AT_EX はコンパイルに失敗します。

3.2.7 Springer 学術誌

すべての `svj*` クラスは、完全に新しい `sn-jnl.cls` に置き換えられました。情報については、<https://www.springernature.com/gp/authors/campaigns/latex-author-support> をご覧ください。LYX のネイティブサポートはまだ為されていません。

3.3 書籍

コレクション (第 3.1 節) にある文書クラスも参照してください。

3.3.1 Memoir

JÜRGEN SPITZMÜLLER 文

3.3.1.1 概観

Memoir は、非常に強力で、着実に改良が加えられている文書クラスであり、フィクションとノンフィクションの文学作品用に設計されています。その目的は、ユーザーに、文書の組版に関して最大限の裁量を得させることにあります。Memoir は、標準

⁷ ついでながら、`tex2lyx` は両方とも対応しているので、`\address` と `\affiliation` のどちらも翻訳することができます。

3 文書クラス

book クラスに基づいていますが、article クラスをエミュレートすることもできます (以下参照)。

Memoir 開発者である Peter Wilson は、L^AT_EX の世界での、多くの便利なパッケージの作者として知られていますが、それらのほとんどは、Memoir に統合されました。このため、章のデザインや、目次や付録などのレイアウトを、はるかに簡単に行うことができます。しかしながら、L_YX は、これらすべての素晴らしい機能をネイティブでサポートしているわけではありません。これらのうちいくつかは、将来のリリースで追加されるかもしれませんが⁸、L_YX のフレームワークが持つ制約のために、おそらく決して追加されることのないと考えられるものも多数あります。もちろん、ネイティブな L^AT_EX コマンド (T_EX コード⁹) の助けを借りれば、全機能を使用できます。この節では、L_YX がネイティブにサポートしている機能についてのみ、列挙するものとします。詳細な説明 (および他の機能) については、Memoir クラスの詳しい取扱説明書¹⁰ をご覧になることをお勧めします。この取扱説明書は、本クラスのユーザーの手引きであるのみに留まらず、良い組版に関する包括的な解説であり、それ自身が良い組版の傑出した用例となっています。

3.3.1.2 基本的な機能と制約

Memoir は、基本的に、標準 book クラスの全機能をサポートしています。しかしながら、以下に示すような若干の違いがあります。

フォント寸法: Memoir では、9・10・11・12・14・17 と、より広い範囲のフォント寸法が使用可能です。

ページ様式: Memoir パッケージと fancyhdr パッケージとのコマンドの衝突のため (両パッケージは同じ名前のコマンドを定義しているため、L^AT_EX が混乱させられます)、fancy ページ様式は使用できません。その代わりに、Memoir には、多くの独自のページ様式が付属してきます (文書▷設定▷ページレイアウト参照)。これらを章題ページに使用したい場合には、コマンド `\chapterstyle` を、本文またはプリアンブルに記述する必要があります (例えば、`\chapterstyle{companion}` など)。

章立て: 標準クラスでは、章立てコマンド (章・節・小節等) は、非必須引数を一つ取ることができます。これを用いて、目次やヘッダ用に別のタイトルを指定することができます (たとえば、タイトルが長すぎる場合などに用います)。L_YX で

⁸もし提案があれば、ぜひ lyx-devel@lists.lyx.org までお送り下さい。

⁹【参考】詳細は、第 2.3 節参照。

¹⁰【参考】CTAN:/macros/latex/memoir/memman.pdf

は、章や節の始めで挿入▷短縮タイトルを使えば、これを指定することができます。一方、Memoirでは、第2の非必須引数を取ることができ、目次用のタイトルとヘッダ用のタイトルの指定を分離しています。L^AT_EXでは、挿入メニューから短縮タイトルの特定の派生形の形で利用できます。

目次・図一覧・表一覧: 標準クラス（および他の多くのクラス）では、目次と図一覧および表一覧は、自動的に新規頁に置かれます。Memoirは、これとは別の方針を採ります。改頁を行いたい場合は、手動で頁区切りを入れなくてはなりません。

タイトル頁: 理由は不明ですが、Memoirはタイトルページに丁付けを行います（標準クラスでは、タイトルページは「empty」様式）。タイトル頁に丁付けを行いたくない場合には、プリアンブルに`\aliaspagestyle{title}{empty}`と入力してください。

Article: クラスオプション *article* を用いる（文書▷設定▷文書クラス▷クラスオプション▷詳細設定で入力）と、*article* 様式をエミュレートできます。すなわち、（脚注・図・表等の）カウンタは、章毎にリセットされず、次の章は頁を改めず（ただし、「本物の」*article* クラスとは違って、頁を改めることも可）、部は *book* クラスと同様に、独立の頁を使用します。

Oldfontcommands: L^AT_EX 旧 2.09 版で使用されていた古いフォントコマンド（`\rm` や `\it` など）は、Memoirにおいては、既定では使用することができません。Memoirがこれらのコマンドを検知すると、エラーを発生してL^AT_EXを停止させます。しかし、クラスオプション *oldfontcommands* を指定すると、これらのコマンドが容認され、エラーの代わりに警告を発生するのみとなります（これによって、少なくともL^AT_EXは停止しなくなります）。まだ、多くのパッケージと、就中多くのBib_TE_Xスタイルファイルが、これらのコマンドを使用しているので、我々はこのオプションを既定で使うこととしました。

3.3.1.3 追加された機能

L^AT_EXでサポートされる機能（今のところそれほど多くはありません）を簡単に説明しましょう。詳しくはMemoirの取扱説明書¹¹を見てください。

概要: 概要が、追加された機能として扱われているのを不思議に思われるかもしれませんが、*book* クラスとしては追加された機能なのです。通常 *book* クラスには

¹¹CTAN:/macros/latex/memoir/memman.pdf.

3 文書クラス

概要はないのですが，memoirにはあります。これは，どこでも何回でも自由に使うことができます。

章要約: この古い組版様式はご覧になったことがあるかも知れません。章の内容が，見出しの下と目次に要約されているものです（たとえば，「我らが英雄がトロヤに到着する。彼は友人を失うが，新たな友人を得る。」など）。章要約 (chapterprecis) はこれを行います。したがって，章要約は章の下でのみ意味を持ちます。

題辞: 題辞 (epigraph) は章頭の，気の利いた標語やモットーです。題辞環境は，そのような標語を組む洗練された方法を提供します。標語本体 (本文) とその著者 (出典: 非必須) は，短い線で区切られます。著者 (出典) は，挿入▷題辞出典で入力できます。

詩題: Memoirには（複雑で隠喩的なものも含めて）詩を組版するための多くの機能がありますが，L_AT_EXは，そのうちいくつかしかサポートしていません。詩題はその一つです。これは詩用の中央揃えタイトルで，目次にも表示されます（詩句が詩の標準環境です。Memoirには，詩句の拡張版もありますが，これらを使うためにはT_EXコードを使用しなくてはなりません。これらの環境は通常の詩句環境の中に入れ子にしなくてはならないのですが，L_AT_EXではそれができないためです）。

詩題*: 目次に現れない他は詩題と同じです。

3.3.2 Recipe Book

RecipeBook 文書クラスは，KOMAScript Book 文書クラスを使用しますが，以下の2つの環境を追加します。

レシピ 二重水平線を引いた中央揃えボールド体の見出しを生成する小々節階層の連番節環境

材料 材料が記述される記述環境で，環境の終わりに水平線が引かれます

ファイル▷用例を開く▷書籍▷Recipe Bookに使用法が描かれています。

このクラスは，1ページに一つか二つのレシピを組版するようにデザインされています。ハイパーリンク化された目次とページ番号によって，レシピをめくるのが便利になっています。

3.4 履歴書

3.4.1 Europass (2013)

Europass (2013) 文書クラスは、2013 年欧州委員会推奨の「Europass CV」の非公式実装である `europasscv.cls` 文書クラスを提供します。ファイル▷用例を開く▷履歴書▷Europass (2013) が使用法の案内をします。ドキュメンテーションは [CTAN](#) から利用可能です。

3.4.2 Europe CV

Europe CV 文書クラスは、2002 年欧州委員会推奨の「Europass CV」の非公式実装である `europcv.cls` 文書クラスを提供します。ファイル▷用例を開く▷履歴書▷Europe CV が使用法の案内をします。ドキュメンテーションは [CTAN](#) から利用可能です。

3.4.3 Modern CV

Modern CV 文書クラスは、`moderncv.cls` 文書クラスを提供します。これによって、カスタマイズ可能な履歴書を生成することができます。ファイル▷用例を開く▷履歴書▷Modern CV が使用法の案内をします。ドキュメンテーションは [CTAN](#) から利用可能です。

3.4.4 Simple

Simple CV 文書クラスは、元々 L^AT_EX で使用するために開発された `simplecv.cls` 文書クラスを提供します。ファイル▷用例を開く▷履歴書▷Simple CV が使用法の案内をします。ドキュメンテーションは [CTAN](#) から利用可能です。

3.5 書簡

KOMA-Script Letter (V. 2) (第 3.1.8.3 節) および Letter (フォント寸法を追加した標準クラス) (第 3.1.2 節) 文書クラスも参照してください。

3.5.1 DIN-Brief

DIN-Brief 文書クラスは、ドイツ規格協会の標準 (*Deutsches Institut für Normung* : DIN) に則った書簡を書くのに用いる `dinbrief.cls` L^AT_EX クラスへのサポートを提

3 文書クラス

供します。ファイル▷新規(ひな型使用)▷書簡▷DIN-Briefメニューが使用法の案内をします。ドキュメンテーションは [CTAN](#) から利用可能です。

3.5.2 フランス語 Letter (frletter)

フランス語 Letter (frletter) 文書クラスは、フランス語の慣習に則った書簡を書くのに用いる `frletter.cls` L^AT_EX クラスへのサポートを提供します。ファイル▷新規(ひな型使用)▷書簡▷フランス語 Letter (frletter)メニューが使用法の案内をします。

3.5.3 フランス語 Letter (lettre)

フランス語 Letter (lettre) 文書クラスは、フランス語の書簡やファックス、封筒を書くのに用いる `lettre.cls` L^AT_EX クラスへのサポートを提供します。ファイル▷新規(ひな型使用)▷書簡▷フランス語 Letter (lettre)メニューが使用法の案内をします。ドキュメンテーションは [CTAN](#) から利用可能です。

3.5.4 G-Brief (V.2)

G-Brief (V.2) 文書クラスは、ドイツ語の慣習に則った書簡を書くのに用いる `g-brief2.cls` L^AT_EX クラスへのサポートを提供します。ファイル▷新規(ひな型使用)▷書簡▷G-Brief (V.2)メニューが使用法の案内をします。ドキュメンテーションは [CTAN](#) から利用可能です。

3.6 プレゼンテーション

3.6.1 Beamer

Beamer 文書クラスは、プレゼンテーションを作成するのに用いる `beamer.cls` L^AT_EX クラスを使用します。ファイル▷新規(ひな型使用)▷プレゼンテーション▷Beamerメニューにあるファイルか、ヘルプ▷用途別説明書▷Beamerプレゼンテーションが使用法の案内をします。ドキュメンテーションは [CTAN](#) から利用可能です。

3.6.2 FoilT_EX

ALLAN RAE 原文 ; L^yX チーム改訂

3.6.2.1 はじめに

FoilTeX 文書クラスは、OHP スライドを作成するのに用いる `foils.cls` 文書クラスを提供します。これを行う 2 つの文書クラス、`slides` 文書クラス (第 3.6.5 節参照) と FoilTeX スライドクラスとがあります。2023 年現在、前者のメンテナンスは続けられていますが、FoilTeX は 2008 年以降メンテナンスされていません。この節は後者の記載を行います。お使いのコンピューターに FoilTeX 文書クラスが導入されていない場合には、おそらく `slides` 文書クラスを使わなくてはなりません。`foils.cls` 文書クラスを導入したい場合には、[CTAN](#) からダウンロード可能です。また、取扱説明書『カスタマイズ篇』の「文書クラスを新規に導入する」の章をご一読ください。

3.6.2.2 最初の一步

いうまでもなく、この文書クラスを使うには、文書▷設定▷文書クラスメニューの文書クラスドロップダウンで「FoilTeX」を選択する必要があります。また、文書設定ダイアログ中には、以下のように、このクラスに特有の設定がいくつかあることを知っておかなくてはなりません。

- 文書設定ダイアログ中の、両面文書オプションと二段組文書オプションは変更しないでください。これらは `foils` クラスでは無視されます。
- 既定のフォント寸法は 20pt であり、17pt・25pt・30pt を選択することも可能です。
- 既定のフォントは sans serif ですが、数式ではすべて、通常のローマン体が表示されます。
- FoilTeX は、35 mm スライド用の特殊寸法他、A4 とレター大の用紙寸法をサポートします。A5・B5・リーガル・エグゼクティブ大の用紙寸法はサポートしていません。
- フロートの配置の設定は無視されますので、わざわざ変更する必要はありません。フロートはすべて、本文中で定義された位置に表示されます。
- このクラスのページレイアウトの設定は、若干異なった挙動を示します。FoilTeX では、ユーザ定義のロゴをはじめ、フッタとヘッダに広範な設定ができます。詳細については第 3.6.2.4 節をご覧ください。タイトルページは、文書中の他のページとは異なった取扱がなされ、ページ番号はつねに付けられず、(ロゴが定義されていれば) ロゴは、つねにページ下部中央に表示されます。使用できるページ様式とその挙動は、以下の通りです。

3 文書クラス

- empty** 最終出力には、ページ番号をはじめ、ヘッダ・フッタが出力されません（もちろん脚注は除きます）。
- plain** 最終出力には、ページ番号がページ下部の中央に出力されますが、他のヘッダ・フッタは出力されません（脚注除く）。
- foilheadings** ページ番号が右下隅に出力されます。追記したヘッダ・フッタも出力されます。これが既定値です。
- fancy** これは fancyheadings パッケージを使えるようにします。ただし、FoilTeX パッケージの作者は、ページレイアウトが崩れる可能性があるため、これを使用することは勧めないと言及しています。

任意設定クラスオプション 以下のオプションは文書設定ダイアログの任意設定クラスオプションボックスで使うことができます。

- 35mmSlide** これは、ページレイアウトを縦 7.33 インチ横 11 インチに設定します。これは 35 mm スライドの縦横比にほぼ等しく、このメディアを作る作業が若干易しくなります。
- headrule** タイトルページを除くすべてのページのヘッダの下に罫線を入れます。
- footrule** タイトルページを除くすべてのページのフッタの上に罫線を入れます。
- dvips** これは、新規 foils 文書を作成する度に自動的に設定されます。このオプションは、横向きに設定されたページを回転させるのに、dvips ドライバを使うよう FoilTeX に指示します。
- landscape** このオプションは、用紙寸法を横向きページ大に変更するだけで、回転は行いません。したがって、このオプションを使う場合は、各ページを回転させる外部プログラムを使用する、プリンタに用紙を横向きに給紙する必要があります。また、このオプションは、Foilhead 環境と Rotatefoilhead 環境の役割を逆転させることに注意してください（これらの環境については次節で説明します）。
- leqno** 数式番号を左側に付けます。
- fleqn** 数式を左揃えにします。

3.6.2.3 サポートされる環境

他のクラスで標準的にサポートされている環境は、FoilTeX クラスでもサポートされています。それ以外に、FoilTeX が提供している環境と、LyX が付け加えた環境があります。以下の環境は、他のクラスと共通の環境です。

- 標準
- 筒条書き (記号)
- 筒条書き (連番)
- 筒条書き (記述)
- 筒条書き (リスト)
- LyX コード
- 詩句
- 引用 (字下げあり)
- 引用 (字下げなし)
- タイトル
- 著者
- 日付
- 概要
- 書誌情報
- 住所
- 右寄せ住所

つまり、節区切りを除けば、主な環境のすべてです。一枚のスライドは、基本的にタイトルと本文を備えた自己完結した節なので、FoilTeX は新しいスライドを始める以下のようなコマンドを備えています。

- スライドタイトル (Foilhead)
- 横置きスライドタイトル (Rotatefoilhead)

LyX は、以下のような、これらを若干修正した環境も用意しています。

- スライドタイトル (小)(ShortFoilhead)
- 横置きスライドタイトル (小)(ShortRotatefoilhead)

これらの違いは次節で説明します。

スライドは、アイデアを説明したり、新しい定理を説明したりするのにしばしば使われるので、FoilTeX はそれらを説明するのに便利な環境を幅広く用意しています。

- 定理
- 系
- 補題
- 命題

3 文書クラス

- 定義
- 証明
- 定理*
- 補題*
- 系*
- 命題*
- 定義*

星付き版は連番が振られず、星なし版は連番が振られます。他に、L_AT_EXで追加された以下の2つの箇条書き環境があります。

- 箇条書き (チック)
- 箇条書き (クロス)

FoilT_EXには、強力なヘッダ機能とフッタ機能があり、これらはプリアンブルで設定するのが最良ですが、文章中のどこでも設定することができます。もし文書中でこれらの設定を変更したい場合には、スライドの一番上、foilheadの直後で行うのが良いでしょう。

この目的のためには、以下のコマンドスタイルがあります (MARTIN VERMEER)。

- グラフィックスロゴ
- 配布制限
- 右フッタ
- 右ヘッダ
- 左ヘッダ

FoilT_EXが提供しているコマンドで、L_AT_EXが直接サポートしていないコマンドもありますが、これらの説明と使い方については、第3.6.2.5節で行います。

3.6.2.4 スライドの組を作る

この節は、様々な環境を使ってスライドの組を作る方法を簡単に紹介します。スライド組の例をご覧になりたい場合は、ファイル▷用例を開く▷プレゼンテーション▷Foilsをご覧ください。

タイトルページを作る タイトル・著者・日付・概要環境のある他のクラスとは異なり、FoilT_EXは独立したページにタイトルを作成します。日付環境を使用しなかった場合には、L_AT_EXは(出力を生成し直したときにはいつも)その時点の日付を挿入します。

新しいスライドを開始する 前述のように、新しいスライドを開始するには、4つの方法があります。縦向きのスライドを開始するには、スライドタイトル (Foilhead) やスライドタイトル (小) (ShortFoilhead) を使用します。両環境の違いは、スライドタイトル (foilhead) とスライド本文の間の余白の違いです。

横向きのスライドは、横向きスライドタイトル (Rotatefoilhead) か横向きスライドタイトル (小) (ShortRotatefoilhead) 環境を使用して生成します。ここでも両環境の違いは、タイトルと本文の間の余白のみです。どちらの場合も、(小) と記された版は、タイトルと本文の間の余白が、0.5 インチ (約 1.27cm) 小さくなります。

横向きのスライドをサポートする際、一つ問題となるのは、スライドを回転させようと思うと、PostScript 出力を生成するのに dvips ドライバを使用することを要求してしまうことです。もしスライドをプリンタに横に給紙することができれば、dvips ドライバを持っていなくても、横向きのスライドをつくることができます;-)

定理・補題・証明など これらの環境の同じ型をお互いに直接続けて置くことはできません。通常の段落開業を用いると、2つの環境をマージしたかのように、前の環境が延長されるだけです。そうではなく、いわゆる分離線を挿入する必要があります。『ユーザーの手引き』の「入れ子の分離」の節を参照してください。

箇条書き 他のクラスでも共通にサポートされている箇条書き様式すべてに加えて、二つの新しい様式を使用することができます。ここでは、その新しい様式のみを説明することにします。他の箇条書き環境について詳しく知りたい場合には、ユーザーの手引きをご覧ください。記号による箇条書きを使用したい場合には、上記第6節の箇条書き (記号) ブリットの指定の節も、読まれることが望ましいでしょう。

箇条書き (チック) と箇条書き (クロス) の二つの新しい様式は、チェック印や × 印を箇条書きのラベルに使用する専用の環境を提供することで、予定一覧や正誤一覧を作りやすくするように設計されたものです。これらの箇条書きは、箇条書き (記号) の派生型を専用に用意したものに過ぎません。ただし、これらは、psnfss パッケージが導入済であることを要求します。

図と表 FoilTeX では、図表フロートは、本文中の挿入箇所にこれらが正確に表示されるように再定義されており、ページの上部やユーザー指定場所に押しつけたりはしないようになっています。たとえフロート配置の設定に変更を加えたとしても、単に無視されます。

ページヘッダおよびページフッタ グラフィックスロゴと配布制限の二つは、左フッタ文を制御するものです。前者は、グラフィックスのロゴをスライドに入れるための

3 文書クラス

もので、既定値は「-Typeset by FoilTeX-」となっています。後者は、「機密事項」などのように、聴衆に秘密区分を提示することを意図しています。既定値では、何も設定されていません。

スライドの残り三隅には、右フッタ（既定ではページ番号）・右ヘッダ（右上）・左ヘッダ（左上）を使って文字を入れることができます。

3.6.2.5 サポートされない FoilTeX 機能

以下で言及されているコマンドは、 \LaTeX 環境内で設定されるか、他の環境中の \TeX コードとして設定される必要があります。

長さ 長さは、すべて `\setlength{長さ名}{新規長さ}` コマンドで調節されます。ここで **長さ名** は、変更したい長さに付けられている名前置き換え、**新規長さ** は、その長さの値で置き換えます。長さはすべて、インチ (in)・ミリメートル (mm)・ポイント (pt) のような長さ単位か、`\textwidth` のように文書比あるいはフォント基準の長さで指定する必要があります。

`\foilheadskip` で指定される長さを調節することによって、スライドタイトルとスライド本文の間の余白を変更することができます。たとえば、スライドタイトルをすべて 0.5 in だけスライド本文に近づけるには、プレアンブルに `\setlength{\foilheadskip}{-0.5in}` と書き込みます。

フロート周囲の余白は、以下の長さを設定することで調整することができます。

`\abovefloatskip` 本文とフロート上辺との間隔

`\abovecaptionskip` フロートとキャプションとの間隔

`\belowcaptionskip` キャプションとそれに続く本文との間隔

`\captionwidth` この長さを調整することでキャプションを、周囲の本文より短くすることができます。`\textwidth` に対する相対的な長さで調整するのが最も良いでしょう。

以下のようなタイトルページ関連の長さもあり、タイトルが長い場合や、著者が複数いる場合に使うと便利かもしれません。

`\abovetitleskip` ヘッダとタイトルとの間隔

`\titleauthorskip` タイトル環境と著者環境との間隔

`\authorauthorskip` 著者が複数いる場合の著者行の間隔

`\authordateskip` 著者と日付との間隔

`\dateabstractskip` 日付と概要との間隔

最後に紹介する長さ関連コマンドは、箇条書き環境すべてに使用できるものです。`\zerolistvertdimens` を箇条書き環境の内側に置くと、項目間の垂直余白がすべて取り除かれます。これは長さではなくコマンドなので、上述のもののように`\setlength` を使う必要がないことに注意してください。

ヘッダとフッタ `\LogoOn` および `\LogoOff` は、`MyLogo` で定義したロゴを特定のページに入れるかどうかを制御します。`\LogoOff` をプリアンブルに入れると、全スライドからロゴが取り除かれます。特定のスライドにのみロゴを付けたくない場合は、当該ページのスライドタイトルの直後に直接`\LogoOff` を置き、次のスライドタイトルの直後に直接`\LogoOn` を置いてください。

文書レイアウトダイアログで fancy ページ様式を使用することを選択した場合には、横置きページのヘッダとフッタが回転後も正しく配置されるように、プリアンブルに`\let\headwidth\textwidth` を加える必要があるでしょう。これは、`fancyheadings` パッケージが提供するページレイアウトと `FoilTeX` クラスの提供するページレイアウトの間に衝突が発生するためです。

3.6.3 Powerdot

Powerdot 文書クラスは、プレゼンテーションを作成するのに用いる `powerdot.cls` L^AT_EX クラスを提供します。ファイル▷用例を開く▷プレゼンテーション▷Powerdot メニューが使用法の案内をします。ドキュメンテーションは [CTAN](#) から利用可能です。

3.6.4 Seminar

Seminar 文書クラスは、プレゼンテーションを作成するのに用いる `seminar.cls` L^AT_EX クラスを提供します。ファイル▷用例を開く▷プレゼンテーション▷Seminar メニューが使用法の案内をします。ドキュメンテーションは [CTAN](#) から利用可能です。

3.6.5 Slides (別名 `SlitEX`)

JOHN WEISS 原文；L^yX チーム改訂

3 文書クラス

3.6.5.1 はじめに

本節では、オーバーヘッドプロジェクタ用のスライドをつくるのに、LyXをどのように使えばよいかを説明します。これを行う2つの文書クラス、スライド (Slides) 文書クラスと FoilTeX 文書クラスとがあります。本節は前者について説明を行います。FoilTeX の文書をお探しの場合は、第 3.6.2 節をご参照ください。2023 年現在、Slides のメンテナンスは続けられていますが、FoilTeX は 2008 年以降メンテナンスされていません。

3.6.5.2 使い始めるには

当然ながら、この文書クラスを使用するには、文書▷設定ダイアログのクラスリストから「スライド」を選択しなくてはなりません。このクラスについて、以下のように、特に知っておいた方が良いことがいくつかあります。

- 用紙方向と二段組文書の各オプションは、あえて変更する必要はありません。これらのオプションは、スライドクラスではサポートされていないので、効果がありません。
- ヘディング様式オプションは、このクラスでは少し異なった振る舞いをします。各選択肢とその機能は、以下の通りです。

プレーン (plain) 最終出力の右下隅にページ番号が付きます。

設定 (headings) プレーン (plain) と同様ですが、それ以外にあなたが指定したタイムマーカ―をすべて出力します。これが既定値です。

空 最終出力には、ページ番号もタイムマーカ―も位置揃えマーカ―も出力されません。

- スライドクラスには、もう一つ clock というオプションがあります。これを使用するには、クラスオプションに「clock」と入力してください。

このオプションを使うと、注釈にタイムマーカ―を加えることができます。詳細は、第 3.6.5.4 節をご覧ください。

slides クラスを用いた文書を自動的にセットアップするには、ひな型ファイルを使用することもできます（新規ファイルを開く際にファイル▷新規(ひな型使用)▷プレゼンテーション▷Slides を使用してください）。ひな型ファイルには、このクラスが使用する特別な段落環境の用例も含まれています。これらについては、以下で説明します。

3.6.5.3 段落環境

サポートされている環境 新しくスライド文書を開いたときにまず気付くのは、フォント寸法とフォント族の違いでしょう。ここでは、サンセリフ体フォントの極大寸法が用いられており、出力でもこのフォントが使用されます。これは、現在スライドを作成中であることの「目印」だと考えてください。出力スライドは大きなフォントを使用するため、書ける場所は少なくなります。もちろん、既定の大きな画面フォントは WYSIWYG ではなく、スライドであること目印に過ぎません。

次に気付くのは、(ツールバー左端の) 段落環境プルダウンボックスが変わっていることでしょう。通常見かける環境の多くがなくなって、代わりに5つの新しい環境が入っています。これは、Slides クラスが、標準段落環境のうち以下のものしかサポートしていないためです。

- 標準
- 筒条書き (記号)
- 筒条書き (連番)
- 筒条書き (記述)
- 筒条書き (一覧)
- 引用 (字下げあり)
- 引用 (字下げなし)
- 詩句
- キャプション
- L_AT_EX コード

これ以外の標準的な環境、例えば節見出し用の環境などは slides クラスでは使えません。

一方、以下の新しい環境があります。

- スライド
- オーバーレイ
- 注釈
- 不可視文
- 可視文

これら5つの環境は、L_AT_EX の「機能」のために使い方に癖があります。ご存じのように、L_AT_EX では、空の段落環境に他の環境を入れることができません。それ自体は素晴らしいデザインなのですが、これは、スライドを平文以外では書き始めることができないことを意味します。これを取り扱うためには、ちょっとした「L_AT_EX マジック」が必要になります。

新環境の癖 5つの新しい環境にはすべて、L_AT_EX 現行版の制約に由来する癖があります。いま述べたように、L_AT_EX では、環境を別の環境で開始することが禁止されていま

3 文書クラス

す。これを回避するために、スライド環境は『ユーザーの手引き』で述べられているような段落環境ではなくなっています。

スライド・オーバーレイ・注釈は「疑似環境」だとお考えください。これらは節見出しや「キャプション」のように見えますが、実際には（必要なら前段落を終了して）新しい段落環境を開始します。同様に、不可視文と可視文も「疑似コマンド」とお考えください。この2つは特定のアクションを行います。

スライド・オーバーレイ・注釈・不可視文・可視文の5環境に共通しているのは、長めのラベルを持つことです。このラベルに続く文章—通常は段落環境の内容になるはずですが—is、スライド・オーバーレイ・注釈・不可視文・可視文では、まったく意味がなく、 L_X はこの部分を完全に無視します。これらの5つの環境の内容は、完全に空のままにしておいて構いません。

長めのラベルの後に文章を入れる必要はまったくないとはいえ、何か書いておきたいこともあるかもしれません。たとえば、スライドの内容に関する短い説明などがそうでしょう。この場合、いつもどおり説明用のコメントを入力して Return を押してください。

3.6.5.4 スライド・オーバーレイ・注釈で発表用のスライドを作る

スライド環境を使う もし本節が、実際のプレゼンテーションの作り方を教えてくれるものと期待されているとすれば、大いにがっかりされるかもしれません。当然ながら、プレゼンテーション用の資料を準備するうえでスライドクラスを役立てる方法を詳しく説明はしますが、内容をどうするかはあなた次第です（そしてこれも L_X の哲学です）。

スライド環境を選択すれば（方法は第3.6.5.3節参照のこと）、 L_X は新規スライドを開始します。この環境ないし「疑似環境」のラベルは、鮮やかな青で「新規スライド:」と表示されます。この環境の後のテキスト環境や段落環境は、すべて新規スライドに入ります。簡単ですよ。

Slidesは、おそらく L_X の中で強制的に改頁をしなくてはならない唯一のクラスです（これは段落レイアウトダイアログで指定できます）。実際、一つのスライドの内容を入力し終わったら、改頁したくなるはずですが、仮に一つのスライドに物理的に入りきらない文章を入力したとすると、入りきらない部分は、次の新規スライドにまわされます。しかしながら、入りきらない部分のスライドにはページ番号が付きませんので、こうするのはお勧めしません。その上、一枚に収まらないスライドでは、オーバーレイを使用すると干渉する恐れがあります。

オーバーレイ環境と注釈環境も、スライド環境と同様の動作を示します。これらは、青の代わりに鮮やかなマゼンタで、それぞれ「新規オーバーレイ:」ないし「新

規注釈:」というラベルを表示します。ラベルのフォントも、3つの環境でそれぞれ異なるものが使用されます。

スライドと同様に、オーバーレイや注釈の内容が、一つのスライドないしページに物理的に収まりきらない場合には、余分は新規スライドに回されます。この場合には、注釈やオーバーレイの本来の目的が達せられなくなりますので、ここでもこのような状態は避けるべきでしょう。

スライドでオーバーレイを使用する オーバーレイの背後にある考え方は、別のスライドの上にスライドを載せようとするものです。メインのスライドに図があるとき、関連した文章を表示する前に、図について議論を進めたいと思うことがあるかもしれませんが。これを行うひとつの方法は、スライドで後に表示したい部分に黒い紙の蓋を貼り付けてしまうことですが、この方法は、たとえばグラフを他のグラフに重ねたい場合などには、うまく行きません。この場合には、二つのグラフを揃えるために、重ねる二つのスライドを、喋りながら手探りで揃えなくてはなりません。これらのような場合には、オーバーレイ環境を使えば、もっと楽になります。

各オーバーレイは、「親」スライドのページ番号を引継ぎ、その後「-a」を付け加えます¹²。当然のことながら、スライドとオーバーレイの内容は両方とも、1枚のスライドに物理的に収まるようになっていなければなりません。オーバーレイはスライドの「一部」であると考えるのがよいでしょう。L_AT_EXでは、オーバーレイのラベルをスライドのラベルよりも下げることで、このことを視覚的に表しています。また、印刷可能な出力を生成すると、オーバーレイと親スライド両方の四隅に、揃えマーカーがあることに気づかれるはずです。これは、2枚のスライドを物理的に重ね合わせるための補助となるものです。

2つのスライドをオーバーレイさせる際の大きな問題は、2枚のスライドの内容を揃えることです。2枚目のスライドのグラフには、どれだけの余白を残しておくべきなのでしょうか？さらにややこしく、2枚目のスライドにグラフと文章を入れたいときに、その間に割り込んでしまうような文章がメインスライドにある場合には、どうでしょうか？正しい寸法の垂直スペースを挿入してみることもできるでしょう。しかし、もっと良い方法として、不可視文と可視文を使用する方法があります。

これらの名前が意味するとおり、不可視文と可視文は、それぞれ続く文章すべてを見えなくしたり見えるようにしたりするコマンド型の段落環境です。しかしながら、第3.6.5.3節で述べたように、この2つの環境の中には何も置くことができません。不可視文を作成すると、「<以下不可視文>」と書かれた水色のラベルが、ページに中央

¹²想像できるように、複数のオーバーレイがあれば、「-a」「-b」「-c」などが親スライドのページ番号に付け加えられます。

3 文書クラス

揃えで表示されます。このラベルに続く段落は、それが置かれるべきスライド（あるいはオーバーレイ；どちらでも同じことです）部分に、代わりに空白の部分が置かれます。

可視文の場合には、中央揃えのラベルは、鮮やかな緑の「<以下可視文>」となります。このラベル以降の段落は、通常通りの挙動を示します。新規のスライド・オーバーレイ・注釈を開始すると、不可視文は自動的に終了されることに注意してください。したがって、スライドの終わりでは、可視文を使用する必要はありません。

ここまでで、スライドやオーバーレイ上で不可視文と可視文を適切に使用して、スライドをオーバーレイさせる方法が明らかになったことと思います。

1. メインのスライドかオーバーレイかに関わらず、スライド上に現れるすべてを含んだスライドを作成します。
2. オーバーレイのみに表示するすべての図や段落の前に、不可視文環境を挿入し、必要ならば、オーバーレイのみの文章の後に可視文環境を挿入します。
3. スライドの直後から、オーバーレイを開始します。
4. このスライドの内容をオーバーレイにコピーします。
5. オーバーレイ内部の不可視文行をすべて不可視文に変更し、可視文は不可視文に変更します。

これだけです。これでオーバーレイができあがりました。

L_YX スライドクラスをデザインする上で、ひとつ残された問題があります。段落途中の文章を不可視にしたり、不可視段落の途中の文章を再度可視にしたりすることはできないのです。これを実現するためには、L^AT_EX コードを埋め込む必要があります¹³。

スライドで注釈を使う オーバーレイと同様、注釈も「親」スライドに関連付けられています。ここでも L_YX スライドクラスは視覚的な効果を与えています。注釈のラベルはスライドよりも短く（オーバーレイのラベルよりも長いですが）、オーバーレイのラベルと同様、派手な赤紫色です。加えて、印刷した注釈には、「親」スライド

¹³このコマンドは、

- `{\invisible ... }`
- `{\visible ... }`

... のような形となり、T_EX としてマークする必要があります。「可視性」を変更したい文章は、波括弧のあいだ（そして `\invisible` ないし `\visible` コマンドの後ろ）に入れます。T_EX としてマークする方法がわからない場合は、ユーザーの手引きをご参照ください。

のページ番号に「-1」「-2」「-3」等を付けたページ番号が付けられます。1つのスライドには、複数の注釈を関連付けることができるので、スライドとオーバーレイと同じように、長い注釈は、1枚の紙に収まるように分割したいと思われるに違いありません。

注釈の目的は明らかです。スライドに関して付け加えたいことを書き留めておくことです。特定のスライドについての覚書として使うこともできます。後者の場合には、タイムマーカを利用したいこともあるかもしれません。現在のところ、 $\text{L}_\text{Y}\text{X}$ のスライドクラスは、 $\text{S}_\text{L}\text{T}_\text{E}\text{X}$ の機能であるタイムマーカを「ネイティブには」サポートしていません。したがって、 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ コードを使用することに頼らなくてはなりません。

タイムマーカを使用するには、クラスオプション「clock」を追加指定する必要があります（第3.6.5.2節参照）。このオプションはタイムマーカ機能を有効にして、生成したすべての注釈の左下にタイムマーカを表示します。タイムマーカに表示するものを設定するには、 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ コマンド「`\settime{}`」および「`\addtime{}`」を使用します。両コマンドは、秒で測った時間を引数にとります。「`\settime{}`」は、タイムマーカを特定の時間に設定し、「`\addtime{}`」は、タイムマーカを指定した分だけ進めます。このようにタイムマーカと注釈を用いれば、特定のスライドにどれだけの時間をかけるのか忘れないでいられます。

最後に、説明しておくべき機能がもう一つあります。当然ながら、スライドとオーバーレイはすべてスライドに、注釈はすべて普通の紙に印刷したいはずですが、しかしながら、注釈は関連付けられた親スライドの直後に来なくてはなりません。ユーザーはどうしたら良いのでしょうか。

幸いなことに、どれを印刷するか選択することを可能にする $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ コマンドが、2種類あります。どちらも文書のプリアンブルに記述されなくてはなりません。コマンド「`\onlyslides{\slides}`」は、出力にスライドとオーバーレイのみが含まれるようにするものです。同様に、「`\onlynotes{\notes}`」は注釈以外の出力をすべて抑制します。私のアドバイスとしては、両コマンドをプリアンブルに書いておいて、最初は両方ともコメントアウトしておくのが良いでしょう。そうすれば、執筆時にはプレゼンテーション全体をプレビューすることができます。いったん書き終えたならば、二つのうち一方のコメントを外して、印刷したい方を選ぶことができます。私ならば、まず`\onlyslides{\slides}`のコメントを外して、現ファイル名に「*-slides」を付けた新たなファイルに出力を行ない、今度はこの行をふたたびコメントアウトするとともに、「`\onlynotes{\notes}`」のコメントを外して、これを「*-notes.ps」というファイルに出力します。そうすれば、必要に応じて、スライドまたは通常の紙をプリンタにセットし、一方のファイルをプリンタに送ればよいのです。

「`\onlyslides{}`」および「`\onlynotes{}`」コマンドには、他の引数を与えるこ

3 文書クラス

ともできます。詳しくは、 \LaTeX の良書をご覧ください。

3.6.5.5 スライドクラスのひな型ファイル

スライドクラス用のひな型ファイルが提供されています。これを使用するには、新しいプレゼンテーションを作り始める際に、ファイル▷新規(ひな型使用)▷プレゼンテーション▷スライドメニューを選択してください。すると新規ファイルには、スライド—オーバーレイ—註釈の組が、例として含まれており、スライドとオーバーレイには、さらに不可視文と可視文の使用例も含まれています。また、プリアンブルには、次のように書き込まれています。

```
% Uncomment to print out only slides and overlays
%
%\onlyslides{\slides}

% Uncomment to print out only notes
%
%\onlynotes{\notes}
```

最後に、私はこの文書クラスを、 $\text{\LaTeX} 2_{\epsilon}$ の組み込みクラスである「 \LaTeX エミュレーション」クラスをサポートする目的で作成しました。私も \LaTeX チームの他のメンバーも、この組み込みスライドクラスを使用することに賛成したり反対したりするものではありません。これは単に、ユーザーがこれを使いたかったり必要としたときのためにあるものです。プレゼンテーションを作成するためには、 $\text{\text{FoilTeX}}$ (第 3.6.2 節参照) や $\text{\text{Seminar}}$ (第 3.6.4 節参照) のように、他の $\text{\LaTeX} 2_{\epsilon}$ クラスも存在します。他にどのようなことができるのか試してみてください。

3.7 レポート

3.7.1 Report

Report クラスは、いわば Book クラスと Article クラスの相の子です。これは Book 文書クラスのように部・章・節を提供しますが、前付け・中付け・後付けは提供しません。また Article 文書クラスのように概要段落様式を提供し、既定では片面文書です。また(両面モードでも)、新しい章を右側ページから始めることはしません。

外部でメンテナンスされていて、 \LaTeX が公式にサポートしている Report 文書クラスについては、コレクション(第 3.1 節)に説明があります。

- 日本語 Report (標準クラス) および日本語 Report (標準クラス：縦書き) については、第 3.1.4 節を参照してください。
- KOMA-Script Report については、第 3.1.8 節を参照してください。
- Report (フォント寸法を追加した標準クラス) については、第 3.1.2 節を参照してください。
- ポーランド語 Report (MW シリーズ) については、第 3.1.9 節を参照してください。

3.8 台本

台本カテゴリでは、演劇や映画の台本を執筆を援助する文書クラスをまとめています。このカテゴリには、講演台本のような他の台本にも開かれています。まだ公式に L^AT_EX がサポートしているクラスはありません。

3.8.1 Broadway

GARST REESE 文

3.8.1.1 はじめに

Broadway は演劇を書くためのものです。書式は Hollywood (第 3.8.2 節参照) よりも装飾的で、それほど標準化されていません。この書式はワークショップに向いているでしょう。

3.8.1.2 特別な問題

Hollywood と同様です (第 3.8.2.2 節参照)。

3.8.1.3 特別の機能

話者の名前をラベルとして挿入し、名前を挿入する際には、このラベルを相互参照してください。相互参照ダイアログには、登場人物の現在の配役が表示されます。

3.8.1.4 用紙寸法と余白

US レター寸法で、余白は左 1.6 インチ、右 0.75 インチ、上 0.5 インチ、下 0.75 インチです。

3 文書クラス

3.8.1.5 環境

以下の環境を使うことができます。右に記してあるキー割当ては、broadway.bindを使った場合のものです。

- **標準**
これは使わなくていいはずですが、他の環境が当てはまらない場合のためにここに挙げておきます。
- **ト書き** M-z n
舞台設定と動作の説明に用います。最初に話者名をすべて大文字で書いておきます。
- **幕** M-z a
自動的に連番が振られます。連番は、画面上ではアラビア数字ですが、出力ではローマ数字です。
- **幕*** M-z S at
幕の副題です。単に中央揃えの文になります。
- **場面** M-z S-S
自動的に連番は振られませんので、手動で番号を付けてください。これは私がどうすればいいか分からなかったためです。
- **幕開きに：** M-z S-R
ト書きの特殊な場合で、カーテンが上がるときの舞台設定と動作を説明します。
- **話者** M-z s
話者（役者）の配役です。すべて大文字で中央揃えに置かれます。
- **括弧付き** M-z p
話者への指示です。括弧は自動的に挿入されます。画面上には左括弧（のみが表示されますが、印字出力には両方の括弧が表示されます。この環境は、対話の中だけで使用されます。
- **対話** M-z d
話者の言う内容。
- **幕切れ** M-z S-C
幕が降ります。

- タイトル M-z S-T
- 著者 M-z S-A
- 右寄せ住所 M-z r

3.8.2 Hollywood (Hollywood スペック・スクリプト)

GARST REESE 文

3.8.2.1 はじめに

脚本を正しい書式で書くことは、ハリウッドにおける「通過儀礼」です。ハリウッドの脚本は、読み手が内容に集中できるようにし、俳優にとっても読みやすくかつ親しみやすいように設計されています。脚本の1ページは、映画の1分でなければならず、銀幕上で見えたり聞こえたりしないものは、脚本に混入してはなりません。また、全体を通して Courier 12 pt フォントを使わなければならず、イタリック体は用いてはいけません。

3.8.2.2 特別な問題

話者の行は、文の途中で改行することは決してあってはなりません。もし話者行の塊が、次頁にまたがる場合には、話者名を再度書き、その後に (Cont'd) と加えます。

3.8.2.3 特別な機能

話者名はラベルとして挿入し、話者名を挿入するには、そのラベルを相互参照してください。相互参照ダイアログは、登場人物の現在の配役を表示することになります。これは、ト書きに話者名を挿入するのにも使用することができます。

3.8.2.4 用紙寸法と余白

US レター寸法で、左余白 1.6 インチ・右余白 0.75 インチ・上余白 0.5 インチ・下余白 0.75 インチです。

3.8.2.5 環境

以下の環境が利用可能です。右に示されているキー割当てを使用するには、hollywood.bind を使用してください。

3 文書クラス

- **一般**
他の環境が動作しない場合に使用しますが、これを使うことは避けるようにしてください。
- **フェイドイン:** M-z S-I
通常、この後には「Sally は目を覚ますと」のように続きます。
- **屋内:** M-z i
新しい屋内 (INTERIOR) カメラの設定を指定します。この後には、かならず DAY や NIGHT などを指定して、必要な照明を定義します。この行はすべて大文字で書きます。
- **屋外:** M-z e
屋外 (EXTERIOR) カメラの設定を指定します。この行はすべて大文字で書きます。
- **話者** M-z s
話す役。
- **括弧付き** M-z p
話者への指示。自動的に括弧 () が挿入されます。
- **対話** M-z d
話者が話す内容。
- **画面遷移** M-z t
カメラの動きの指示。例：CUT TO:.
- **フェイドアウト:** M-z S-I
- **著者** M-z S-A
- **タイトル** M-z S-T
- **右寄せ住所** M-z r

3.8.2.6 脚本で使われる特別な用語

- (O.S) — off screen
- (V.O) — voice over

- b.g. — background
- C.U. — close-up
- PAN — camera movement
- INSERT — cut to close-up of

4 モジュール

4.1 特定の学術分野

4.1.1 化学：危険有害性情報・注意書き

このモジュールは、化学物質の危険有害性情報・注意書きの番号と説明文を組版するための、2つの差込枠と段落様式を提供します。詳細については、[ファイル▷用例を開く▷モジュール▷危険有害性情報・注意書き](#)および[ヘルプ▷用途別説明書▷危険有害性情報・注意書き](#)をご覧ください。

4.1.2 化学：リスクと安全性に関する表示

このモジュールは、[挿入▷特別差込枠](#)から入力することのできるR-S番号とR-S文の2つの差込枠と、化学物質のリスクと安全性に関する表示の番号と説明文を組版するための1つの環境を提供します。

4.1.3 言語学

このモジュールは、言語学に便利な環境(付番用例や語句注解, セマンティック・マークアップ, OT 絵画フロート)を付け加えます。このモジュールについての情報は、[ヘルプ▷用途別説明書▷言語学](#)および[ファイル▷用例を開く▷モジュール▷言語学](#)で入手できます。

4.2 アクセシビリティ

4.2.1 点字

このモジュールは点字筆記をサポートし、 \LaTeX パッケージの `braille.sty` を必要とします。パッケージと関連文書は [CTAN](#) から入手できます。ヘルプ▷用途別説明書▷点字および[ファイル▷用例を開く▷モジュール▷点字](#)をご覧ください。

4.3 記註及び改訂

4.3.1 FiXme 註釈

このモジュールは、文書改訂のための「FiXme」(余白) 註釈を提供します。「FIXME 一覧」段落様式を用いれば、註釈一覧を生成することができます。これらの註釈は LaTeX プリアンブルで設定を変更できます。[CTAN](#) で入手できる `fixme` 解説文書をご覧ください。

註： 既定状態では、註釈は「原稿」モードでのみ表示されます(文書▷設定▷文書クラス▷クラスオプション▷詳細設定で「draft」オプションを指定した時のみ)。註釈をつねに表示するには、文書▷設定▷プリアンブルに`\fxsetup{draft}`と指定してください。

4.3.2 PDF コメント

このモジュールは、PDF 出力向けに各種註釈を提供します。ヘルプ▷用途別説明書▷PDF コメントおよびファイル▷用例を開く▷モジュール▷PDF コメントをご覧ください。

4.3.3 PDF フォーム

PDF フォーム用のフィールドとボタンを提供します。[hyperref 解説文書](#)およびヘルプ▷用途別説明書▷PDF フォーム およびファイル▷用例を開く▷モジュール▷PDF フォームをご覧ください。

4.3.4 ルビ (振り仮名)

このモジュールは、漢字の読み(ルビ・振り仮名)を組版する差込枠を定義します。(TeX エンジンによって) `okumacro`・`luatexja-ruby`・`ruby` パッケージあるいは避難的定義を使用します。

4.3.5 TODO 註釈

このモジュールは、文書中に TODO 項目を入れ込む特別差込枠を提供し、「TODO 一覧」を生成するための段落様式も提供します。文書▷文書クラス▷クラスオプション▷詳細設定に「final」を渡せば、TODO 註釈は出力されません。[todonotes 解説文書](#)を参照してください。

4.4 書誌情報

4.4.1 Natbib を使用した APA 様式

このモジュールは、apacite と共に natbib を使用できるようにします (書誌情報様式は apacite である必要はありません — natbib と apacite の両パッケージで動作する様式であれば、apacite や apacitex あるいは他の様式でも構いません)。

4.5 ボックス

4.5.1 色付き装飾ボックス

このモジュールは、tcolorbox パッケージを使って色付きボックスをサポートする特別差込枠を追加します。詳細については、ヘルプ▷用途別説明書▷色付きボックスおよびファイル▷用例を開く▷モジュール▷色付き装飾ボックス、[tcolorbox 解説文書](#)をご覧ください。

4.5.2 画像ボックス

これは、ボックスの中身を伸縮したり回転したりする反転ボックス・寸法変更ボックス・回転ボックス・伸縮ボックスの各特別差込枠を提供します。

反転ボックスは、下記のように、特段の設定なく簡単に文を反転させることができます。

```
Great Western Railway
```

(これらの例を見るためには、Ctrl-R を押す必要があります。)

寸法変更ボックスを使うと、文や画像の寸法を指定することができます。使用することのできる単位は、em・ex・in・pt・pc・cm・mm・dd・cc・nd・nc・bp・sp です。下記のように高さに ! を入力すると、幅の因数で寸法を指定できます。

Great Western Railway グレートウェスタン鉄道

回転ボックスを使えば、ボックスの参照点を中心に内容物を回転させることができます。入れることができます。異なる原点の周りに回転させたい場合は、ボックスの中にカーソルを置いて、挿入▷原点でを選択してください。origin は c・l・r・b・t あるいはこれらの組み合わせで指定できます。また、回転角度は、反時計回りの角度で表現されます。たとえば、

Great Western Railway グレートウェスタン鉄道

また、下記のように、ボックスを組み合わせることも可能です。

Great Western Railway グレートウェスタン鉄道

伸縮ボックスは内容を伸縮します。垂直因子を追加するには、挿入▷垂直因子を選択してください。垂直因子を使わない場合には、水平因子が使われます。水平因子と垂直因子の値が異なる場合には、下記のように歪みが生まれます。

Great Western Railway グレートウェスタン鉄道

負の水平因子は文を水平方向に逆にし、

Great Western Railway グレートウェスタン鉄道

負の垂直因子は文字を垂直方向にひっくり返します。

Great Western Railway グレートウェスタン鉄道

4.5.3 節ボックス

このモジュールは、節見出し付きのボックスを定義します。主に SciPoster 文書クラス向けです。

4.5.4 可変幅ミニページ

このモジュールは、 \LaTeX パッケージの `varwidth` を使用するミニページ (可変幅) 差込枠を挿入▷特別差込枠に追加します。`varwidth` パッケージは、幅が (指定された上限幅を越えない限り) 内容幅と同じになるような、可変幅のミニページを提供します。この差込枠には2つの非必須引数があり、カーソルをミニページの内側に置いて、挿入▷垂直調整 (`c|t|b`) か挿入▷最大幅 (既定値は `\linewidth`) を使用します。ファイル▷用例を開く▷モジュール▷可変幅ミニページをご覧ください。

4.6 修正とハック

4.6.1 Computer Modern フォントの修復

このモジュールは `fix-cm` パッケージを使用して、Computer Modern フォントの外観を改善し、任意の寸法で使用できるようにするものです。 [fix-cm 解説文書](#) を参照。

4.6.2 L^AT_EX カーネル修正 (廃版)

このモジュールは、L^AT_EX のいくつかのバグ修正を含む、L^AT_EX パッケージの `fixltx2e` を読み込みます。このモジュールを使って文書の処理を行うと、`fixltx2e` のバージョンの違いによって、出力文書の外観が異なるかもしれません。

註： 最近の LaTeX カーネルは `fixltx2e` の機能を取り込んでいるので (2015/01/01 現在), 新しい LaTeX 頒布では、`fixltx2e` モジュールは遺物となっています。

4.6.3 簡素形差込枠

このモジュールは、いくつかの差込枠 (索引・派生枝・URL) を簡素形に再定義します。

4.6.4 表題とプリアンプルのハック

このモジュールは、以下の2つの段落様式を新たに提供します。

1. プリアンプル文様式では、そこに入力されたものは全てプリアンプルにそのまま置かれます。これは、プリアンプルコードを L^AX 文書本体に置いておきたいときに使うことができます。
2. 表題文様式では、その中身が、L^AT_EX 文書本体の、`\maketitle` が現れるよりも前の部分に置かれます。これは、表題に関連した部分に派生枝や注釈を用いたいときに役立ちます (もしこれらを標準レイアウトにおいてしまうと、L^AX がその前に `\maketitle` を出力してしまうので、早すぎることになってしまいます)。

4.7 フロート及びキャプション

4.7.1 Algorithm2e フロート

このモジュールは、アルゴリズムフロートに、L^AX 固有のアルゴリズムフロートではなく `algorithm2e` パッケージを使用します。アルゴリズムを入力して字下げするには、

4 モジュール

アルゴリズム段落様式を使用してください。 [algorithm2e 解説文書](#)を参照。

4.7.2 二言語キャプション(多言語キャプション)

このモジュールは、二言語キャプションを組むための段落様式キャプション設定を提供します。この段落様式の中で挿入▷言語とすると、babel言語名を使って望む第二言語を入力することができます。使用法について、より詳しくは、ファイル▷用例を開く▷モジュール▷多言語キャプションおよびヘルプ▷用途別説明書▷多言語キャプションを参照。

4.7.3 図番号を節毎に振る

このモジュールは、図番号を節の開始毎にリセットし、「図 2.1」のように節番号を図番号の前に付します(既定では、book および report 文書クラスでは章ごとに番号を振り、article 文書クラスでは一続きとなります)。

4.7.4 表番号を節毎に振る

このモジュールは、表番号を節の開始毎にリセットし、「表 2.1」のように節番号を表番号の前に付します(既定では、book および report 文書クラスでは章ごとに番号を振り、article 文書クラスでは一続きとなります)。

4.8 脚註及び後註

4.8.1 後註(基本)

このモジュールは脚註差込枠に加え、後註差込枠を追加します。これはendnotesパッケージを使用します。このパッケージにはいくつか制約がありますが、古いL^AT_EX 頒布版でも動作します。後註挿入箇所に挿入▷特別差込枠▷後註を実行し、後註本体を表示させたいところに挿入▷一覧/目次/参照▷後註として後註一覧を挿入します。ヘルプ▷埋込オブジェクト篇の第4.2節「脚註」を参照。

4.8.2 後註(拡張)

このモジュールは脚註差込枠に加え、後註差込枠を追加します。これはenotezパッケージを使用します。このパッケージはendnotesパッケージよりも強力かつ柔軟ですが、かなり新しいL^AT_EX 頒布版(L^AT_EX3)を必要とします。後註挿入箇所に挿入▷特

別差込枠▷後註を実行し、後註本体を表示させたいところに挿入▷一覧/目次/参照▷後註として後註一覧を挿入します。ヘルプ▷埋込オブジェクト篇の第4.2節「脚註」を参照。

4.8.3 脚註を後註に (基本)

このモジュールは、すべての脚註を後註にします。これはendnotesパッケージを使用します。このパッケージにはいくつか制約がありますが、古いL^AT_EX頒布版でも動作します。後註本体を表示させたいところに挿入▷一覧/目次/参照▷後註として後註一覧を挿入します。ヘルプ▷埋込オブジェクト篇の第4.2節「脚註」を参照。

4.8.4 脚註を後註に (拡張)

このモジュールは、すべての脚註を後註にします。これはenotezパッケージを使用します。このパッケージはendnotesパッケージよりも強力かつ柔軟ですが、かなり新しいL^AT_EX頒布版(L^AT_EX3)を必要とします。後註本体を表示させたいところに挿入▷一覧/目次/参照▷後註として後註一覧を挿入します。ヘルプ▷埋込オブジェクト篇の第4.2節「脚註」を参照。

4.9 レジャー・スポーツ・音楽

4.9.1 チェス盤

このモジュールは、チェスゲームを印刷するために、chessboardパッケージをサポートします。ファイル▷用例を開く▷論文▷チェスを開くとゲーム1とゲーム2がありますので、それらと[chessboard 解説文書](#)を参照してください。

4.9.2 Lilypond 記譜

このモジュールは、LilyPond楽譜エディタ用のコードを挿入▷特別差込枠▷LilyPondにて入力する差込枠を提供します。これは出力時に処理されます。ヘルプ▷用途別説明書▷LilyPondおよびファイル▷用例を開く▷モジュール▷LilyPond Bookを参照。これは、Rnw(knitr)またはSweaveが読み込まれているときには、使用できません。

4.10 高度な箇条書き

4.10.1 調整可能な箇条書き

このモジュールは、`enumitem` パッケージを読み込んで、各種箇条書きの調整を可能にします。

4.10.1.1 調整可能な連番箇条書き

連番箇条書きの番号の振り方の既定値は、箇条書きの各階層における最初の項目に、非必須引数を加えることで(挿入▷箇条書きオプション)変更することができます。そこに、`TeX` コード(ショートカット:)

```
label=\roman{enumi}
```

というコマンドを書き加えてください。ここで、`enumi` は、第1階層の連番のカウンタです。箇条書きの下位階層の連番を変更するには、コマンド中の「`i`」を、階層番号の小文字ローマ数字で置き換えてください(`enumi`, `enumii`, `enumiii`, `enumiv`)。

コマンド`\roman`は、このカウントを、小文字のローマ数字で出力します。大文字のローマ数字を使うには、上記コマンド中の`\roman`を`\Roman`に置き換えてください。アラビア数字にするには、`\arabic`を使用します。項目に、大文字もしくは小文字のラテン文字で「連番」を付けたい場合には、それぞれ`\Alph`もしくは`\alph`を使用してください。

[註] 連番は1文字に限られているので、ラテン文字では26項目までしか連番を振ることはできません。

下記は、調整した連番箇条書きの例です。

#A# 第1階層

A.1 第2階層

A.2 第2階層

1 第3階層

i) 第4階層

上記リストでは、以下のようなコマンドが使用されています。

```
label=\#\Alph{enumi}\#
```

```
label=\Alph{enumi}.\arabic{enumii}
```

```
label=\bfseries{\arabic{enumiii}}
```

```
label=\emph{\roman{enumiv}}}
```

ここで、コマンド`\emph{}`は、ラベルを強調文字にし、`\bfseries{}`はボールド体にします。

4.10.1.2 Report における連番段落

1. 公的なレポートには、連番を振った段落がよく用いられます。この目的を達するには、第6.3節で要約したような様々な方法があります。各コマンドの `item` を `enum` に変えてください。
2. これらの連番を振った段落は、節の開始前に、`TEX` コードで


```
\renewcommand{\labelenumi}{\thesection.\arabic{enumi}}
```

 と入力することで生成されます。`\thesection` は、節の `LATEX` カウンタを使用し、数字の最初の部分を生成するのに使われています。
3. このように連番を振った段落では、小節が使用されることはあまりないので、この例では小節見出しは無視しました。

4.10.1.3 連番の復帰

以下のように、あいだに段落をはさんで、連番を復帰させることができます。

1. 1つめ
2. 2つめ

通常のテキスト

3. 復帰した連番

連番を復帰させるには、スタイル箇条書き (連番)-復帰を使用してください。

[註] 前に復帰すべき連番がない場合には、`LATEX` エラーが発生します。

箇条書きの連番を復帰させる際、次の番号ではない番号に復帰させたかったり、新しい連番を、指定した番号から始めたいことがあるかもしれません。これも、通常の連番箇条書きの最初の項目に、非必須引数を加えることで実現できます。非必須引数として、

`start=番号`

というコマンドを入力してください。ここで、「番号」は箇条書きを開始したい番号です。例：

4 モジュール

1. 第1項目
2. 第2項目

指定した番号で始める連番：

4. この連番は4から始まります

4.10.1.4 箇条書きの余白

ときには、箇条書きの項目間の垂直方向の余白を、調整したい場合があります。例えば、以下の例の既定の余白が、大きすぎると感じたとしましょう。

- 標準余白の
- 記号付き箇条書き

上記箇条書きの最初の項目に、非必須引数を加えることで余白を詰めることができます。下記の例のように、箇条書きに追加される余白をなくすには、非必須引数にコマンド `nolistsep` を加えます。

- 箇条書きに追加される
- 垂直余白をなくした
- 記号付き箇条書き

余白を広げるには、 \LaTeX パッケージの `enumitem` で提供されている、いくつかのコマンドを使用することができます。詳しくは、取扱説明書 [enumitem 解説文書](#) をご覧ください。

水平方向の余白や行頭下げを変更するためにも、多くのコマンドが使えます。下記の例は、行頭下げを、文書中の段落の行頭下げと同じにし、ラベル分離幅を `2cm` にして、番号がページ余白に置かれるようにしたものです。

1. 負の行頭下げを持つ
2. 連番箇条書き

4.10.1.5 さらなる調整

記述箇条書きのスタイルを変更することも可能です。コマンド

`font=definition`

は、記述ラベルのフォントを変更し、コマンド

`style=definition`

は、箇条書きのスタイルを設定します。

下記は、コマンド

```
font=\itshape, style=nextline
```

を使用した例です。

電離放射線:

電離放射線は、電子を原子や分子から分離させるに十分なエネルギーを持つ、素粒子や電磁波から成ります。

参照カウント:

コンピュータ科学において、参照カウントは、オブジェクト・メモリブロック・ディスク容量・その他のリソースへの、参照やポインタやハンドルの数を保管しておくための技術です。

L^AT_EX パッケージの `enumitem` で提供されるコマンドや機能は、他にもたくさんあります。詳細は、取扱説明書 [enumitem documentation](#) をご覧ください。

4.10.2 段落箇条書き

このモジュールは、`paralist` パッケージを使用し、9つの新しい箇条書き環境を提供します。記号・連番箇条書きを、段落内に組版したり、独立段落や簡素形として組版したりすることができます。大半の環境には、ラベルを整形するためのオプションの引数があります。さらに、LaTeX 環境の記号・連番箇条書きも、同様のオプション引数を使えるように拡張します。詳細については、[ファイル▷用例を開く▷モジュール▷段落箇条書き \(paralist\)](#) および [ヘルプ▷用途別説明書▷Paralist](#)、さらに [paralist 解説文書](#) を参照してください。

4.11 文芸的プログラミング

4.11.1 Noweb

このモジュールは、Noweb を文芸的プログラミングツールとして使用することを可能にします。Noweb のファイルと解説文書は、[CTAN](#) で入手可能です。文芸的プログラミングについては、第 10.10 節に解説があり、[ファイル▷用例を開く▷モジュール▷Noweb](#)、[ファイル▷用例を開く▷モジュール▷Noweb](#)、[ファイル▷用例を開く▷モジュール▷Noweb Listerrors](#)、[ファイル▷用例を開く▷モジュール▷Noweb2LyX](#) の 3 つの用例ファイルがあります。

4.11.2 Rnw (knitr)

このモジュールは、knitr と includernw パッケージを使用します。このモジュールを機能させるには、動的レポートを生成する R の knitr パッケージを、`install.packages('knitr')` として導入する必要があります。

註：

- これは R \geq 2.14.1 に依存します。詳細については <http://yihui.name/knitr> をご覧ください。
- includernw パッケージも導入されていなくてはなりません。
- LilyPond Book または Sweave が読み込まれている場合には、使用できません。

情報と文芸的プログラミングについては、ヘルプ▷用途別説明書▷knitr、ファイル▷用例を開く▷モジュール▷Rnw (knitr) および第 10.10 節をご覧ください。

4.11.3 Sweave

このモジュールを使うと、統計言語 S または R を文芸的プログラミングツールとして使用できるようになります。includernw および fancyvrb パッケージが必要です。includernw と fancyvrb はともに CTAN から入手できます。LilyPond Book が読み込まれている場合には、使用できません。情報と文芸的プログラミングについては、ヘルプ▷用途別説明書▷Sweave、ファイル▷用例を開く▷モジュール▷Sweave および第 10.10 節をご覧ください。

4.12 数式

4.12.1 AMS 定理

このモジュールは、拡張 AMS 機構を使う定理環境と証明環境を定義します。連番付きと連番なしの両方の型が提供されます。既定では、定理は文書を通じて連番が付されます。これは、AMS 定理 (... 毎... 連番) モジュールのうちの一つを読み込むことによって変更することができます。標準定理モジュールや標準定理 (連番なし) モジュールが読み込まれているときには、使用することができません。

4.12.2 AMS 定理 (拡張)

このモジュールは、AMS 定理パッケージを使う定理環境をいくつか追加定義します (AMS 定理モジュールは読み込まれていなくてはなりません). これには、基準・アルゴリズム・公理・条件・註釈・記法・要約・結論・事実・仮定・場合・問題が、連番ありと連番なしの両方の形で含まれています.

4.12.3 AMS 定理 (拡張・章毎に種類別連番)

このモジュールは、AMS 定理 (章毎に種類別連番) モジュールを使う定理環境をいくつか追加定義します (AMS 定理 (章毎に種類別連番) モジュールは読み込まれていなくてはなりません). 中には、基準・アルゴリズム・公理・条件・註釈・記法・要約・結論・仮定・場合が、連番ありと連番なしの両方の形で含まれています. AMS 定理 (拡張) モジュールとは異なり、このモジュールでは定理の種類毎に別々のカウンタを有し、章毎にリセットされます (例: 基準 1 → 基準 2 → 公理 3 → 仮定 4 → ... ではなく、基準 1.1 → 基準 1.2 → 公理 1.1 → 仮定 1.1 → #基準 2.1 → 基準 2.2 → 公理 2.1 → ... となります).

4.12.4 AMS 定理 (拡張・種類別連番)

このモジュールは、AMS 定理 (種類別連番) パッケージを使う定理環境をいくつか追加定義します (AMS 定理 (種類別連番) モジュールは読み込まれていなくてはなりません). 中には、基準・アルゴリズム・公理・条件・註釈・記法・要約・結論・事実・仮定・場合が、連番ありと連番なしの両方の形で含まれています. AMS 定理 (拡張) モジュールとは異なり、このモジュールでは定理の種類毎に別々のカウンタを有します (例: 基準 1 → 基準 2 → 公理 3 → 仮定 4 → ... ではなく、基準 1 → 基準 2 → 公理 1 → 仮定 1 → 基準 3 → ... となります).

4.12.5 AMS 定理 (章毎に種類別連番)

このモジュールは、拡張 AMS 機構を使う定理環境と証明環境を定義します. 連番付きと連番なしの両方の型が提供されます. AMS 定理モジュールとは異なり、ここに含まれる定理型は、種類毎に別々のカウンタを有します (例: 定理 1 → 定理 2 → 補題 3 → 命題 4 → ... ではなく、定理 1.1 → 定理 1.2 → 補題 1.1 → 命題 1.1 → 定理 1.3 → 補題 1.2 → ... となります). 連番は章毎にリセットされます (定理 1.1 → 定理 2.1 → ...). AMS 定理や標準定理, 標準定理 (種類別連番), 標準定理 (連番なし) の各モジュールが読み込まれているときには、使用することができません.

4.12.6 AMS 定理 (種類別連番)

このモジュールは、拡張 AMS 機構を使う定理環境と証明環境を定義します。連番付きと連番なしの両方の型が提供されます。AMS 定理モジュールとは異なり、このモジュールでは定理の種類毎に別々のカウンタを有します (例：定理 1 → 定理 2 → 補題 3 → 命題 4 → ... ではなく、定理 1 → 定理 2 → 補題 1 → 命題 1 → 定理 3 → 補題 2 → ... となります)。連番の及ぶ範囲は文書全体です。章毎または節毎の連番をお望みであれば、「章毎に...連番」または「節毎に...連番」とあるモジュールを選択してください。AMS 定理や標準定理、標準定理 (種類別連番)、標準定理 (連番なし) の各モジュールが読み込まれているときには、使用することができません。

4.12.7 数式番号を節毎に振る

このモジュールを使うと、数式番号が節の開始毎にリセットされ、数式番号の前に「(2.1)」のように節番号が付きます。

4.12.8 標準定理

このモジュールは、非 AMS クラス用にいくつか定理環境を定義します。既定では、定理は文書を通じて連番が付されます。これは、標準定理 (... 連番) モジュールのうちの一つを読み込むことによって変更することができます。AMS 定理や標準定理 (連番なし) のモジュールが読み込まれているときには、使用することができません。

4.12.9 標準定理 (命名可)

このモジュールは、定理名付き定理を使用できるようにします。定理名は挿入▷定理見出し文で入れることができます。

4.12.10 標準定理 (章毎連番)

このモジュールは、定理などを章ごとに付番します (つまり章が始まるたびにカウンタがリセットされます)。このモジュールは、章環境を提供するレイアウトとのみ使用してください。AMS 定理 (種類別連番) または標準定理モジュールのどちらかが読み込まれていなくてはなりません。標準定理 (節毎連番) のモジュールが読み込まれているときには、使用することができません。

4.12.11 標準定理 (節毎連番)

このモジュールは、定理などを節ごとに付番します (つまり節が始まるたびにカウンタがリセットされます)。AMS 定理 (種類別連番) または標準定理モジュールのどちらかが読み込まれていなくてはなりません。標準定理 (章毎連番) のモジュールが読み込まれているときには、使用することができません。

4.12.12 標準定理 (章毎に種類別連番)

このモジュールは、非 AMS クラス用にいくつか定理環境を定義します。AMS 定理モジュールや標準定理モジュールとは違って、このモジュールでは定理の種類毎に別々のカウンタが設定されています (例: 定理 1 → 定理 2 → 補題 3 → 命題 4 → 定理 5 → 補題 6... のようにはならないで、定理 1 → 定理 2 → 補題 1 → 命題 1 → 定理 3 → 補題 2... のように進行します)。連番は、章の開始時にリセットされます。AMS 定理 (種類別連番) または標準定理 (種類別連番) モジュールのどちらかが読み込まれていなくてはなりません。AMS 定理や標準定理、標準定理 (連番なし)、標準定理 (節毎連番)、標準定理 (節毎に種類別連番) の各モジュールが読み込まれているときには、使用することができません。

4.12.13 標準定理 (節毎に種類別連番)

このモジュールは、非 AMS クラス用にいくつか定理環境を定義します。AMS 定理モジュールや標準定理モジュールとは違って、このモジュールでは定理の種類毎に別々のカウンタが設定されています (例: 定理 1 → 定理 2 → 補題 3 → 命題 4 →... のようにはならないで、定理 1 → 定理 2 → 補題 1 → 命題 1 → 定理 3 → 補題 2 →... のように進行します)。連番は、節の開始時にリセットされます。AMS 定理 (種類別連番) または標準定理 (種類別連番) モジュールのどちらかが読み込まれていなくてはなりません。AMS 定理や標準定理、標準定理 (連番なし)、標準定理 (章毎連番)、標準定理 (章毎に種類別連番) の各モジュールが読み込まれているときには、使用することができません。

4.12.14 標準定理 (種類別連番)

このモジュールは、非 AMS クラス用にいくつか定理環境を定義します。AMS 定理モジュールや標準定理モジュールとは違って、このモジュールでは定理の種類毎に別々のカウンタが設定されています (例: 定理 1 → 定理 2 → 補題 3 → 命題 4 →... のようにはならないで、定理 1 → 定理 2 → 補題 1 → 命題 1 → 定理 3 → 補題 2 →... のように進

4 モジュール

行します)。連番の及ぶ範囲は文書全体です。章毎または節毎の連番をお望みであれば、「章毎に...連番」または「節毎に...連番」とあるモジュールを選択してください。AMS 定理や標準定理、標準定理(連番なし)、AMS 定理(種類別連番)の各モジュールが読み込まれているときには、使用することができません。

4.12.15 標準定理(連番なし)

このモジュールは、拡張 AMS 機構を用いる、番号なし定理環境と証明環境のみ定義します。AMS 定理(種類別連番)または標準定理モジュールのどちらかが読み込まれていなくてはなりません。

4.12.16 数式群(subequations)

このモジュールは、LyX 中で数式群をまとめる直感的な方法を提供します。ヘルプ▷数式▷細目番号をご覧ください。

4.13 ページレイアウト

4.13.1 ヘッダ・フッタ文の設定

このモジュールは、文書▷設定ページ▷ページレイアウトダイアログのページ様式オプションを fancy に設定する必要があります。これは、fancyhdr パッケージをサポートする文書クラスに以下の 6 つの環境を追加します。

- 左ヘッダ
- 中央ヘッダ
- 右ヘッダ
- 左フッタ
- 中央フッタ
- 右フッタ

4.13.1.1 装飾的なヘッダ・フッタ

既定のページレイアウトは、どちらかといえば素っ気ないものです。Article (標準クラス) 文書クラスの場合には、ページの下の真ん中にページ番号がつくだけです。この取扱説明書は KOMA-Script Book クラスなので、すこしは凝ってますが、文書▷設定ページ▷ページレイアウトダイアログのページ様式オプションを「fancy」に設定すれば、ページヘッダが「左」「中央」「右」と命名された3つのフィールドに分割されます。フッタも同様に3つのフィールドに分割されます。これらのフィールドをもっとも簡単に設定する L^AT_EX コマンドは、`\lhead`・`\chead`・`\rhead`・`\lfoot` などです。かりに、各ページの左上端に自分の名前を入れたいものとしましょう。次のようなコマンドをプリアンブルに入れるだけです。

```
\lhead{物書 はじめ}
```

これで自分の名前が左上に現れます。フィールドに取り除きたい既定の内容がある場合（往々にして中央のフッタにはページ番号が入ります）には、空の引数を持つコマンドを入れるだけです。すなわち、

```
\cfoot{}
```

とします。

しかしながら、ちょっとした変更を加えるのにもっと簡単な方法があります。文書▷設定▷モジュール▷ページレイアウト▷ヘッダ・フッタ文の設定モジュールを読み込めば、上記の6つのフィールドに対応する新しい環境が6つ加えられます。

もっと美しくしてみましょう。節番号を「第..節」という言葉といっしょに（第3節のように）左上に入れ、ページ番号を（4ページのように）右上に、自分の名前を左下に、日付を右下に入れます。新しく使えるようにした環境を使って、以下の項目を加えてください。

左ヘッダ：第`\thesection` 節

中央ヘッダ：

右ヘッダ：`\thepage` ページ

左フッタ：物書 はじめ

中央フッタ：

右フッタ：`\today`

`\thesection`・`\thepage`・`\today`の各コマンドは、T_EX コードのコマンドとして入力するようにしてください。コマンド`\thesection`と`\thepage`は、それぞれ L^AT_EX の節カウンタおよびページカウンタを取得するもので、これによって現在の節番号とページ番号を印字します。`\today`は、今日の日付を印字するものです。

4 モジュール

ヘッダ下またはフッタ上に描かれる罫線の太さも変更することができます。もし罫線が必要なければ、太さを0にしてください。ヘッダ罫線は、既定では0.4ptの太さであり、フッタ罫線は0ptです。太さを変更するには、

```
\renewcommand{\headrulewidth}{0.4pt}
```

```
\renewcommand{\footrulewidth}{0.4pt}
```

のようなコマンドを使用してください。

`\thispagestyle{empty}`・`\thispagestyle{plain}`・`\thispagestyle{fancy}`のようなコマンドを使えば、各ページごとにヘッダ・フッタ設定を有効にしたり無効にしたりすることができます。設定を変更したいページに、これらのコマンドをTeXコードとして挿入してください。実は、タイトルページは、その後のページは大域的に装飾的 (fancy) に設定されていれば装飾的 (fancy) になるにもかかわらず、既定でプレーン (plain) になるように設定されているのです。

ヘッダやフッタを両面文書の外側にしたいときには、

```
\fancyfoot{}
```

```
\fancyfoot [LE,RO] {\thepage}
```

のようにします。最初のコマンドは、新しいコマンドと干渉するのを避けるために、既存のフッタをすべて消去しています。

これらのコマンドは、簡単なものは上記で新しく作った環境の中に、複雑なものはTeXコードコマンドとして、文書中の任意の場所に入れることができ、その次のページから効力を持ちます。

最後の例として、ヘッダやフッタに画像を挿入できることを取り上げましょう。会社のロゴをページ左上に入れたいものとします。たとえば、文書▷設定▷LaTeXプリアンブルに

```
\thead{\resizebox{1in}{!}{\includegraphics{logo.png}}}
```

と書けば、これを実現できます (文書中に他に画像を挿入していない場合には、`\usepackage{graphics}`をプリアンブルに宣言する必要があります)。

詳しい解説は、[CTAN](#)にある `fancyhdr` の解説文書を参照してください。

4.13.2 文中横向き部分

このモジュールは、挿入▷特別差込枠▷横向きか挿入▷特別差込枠▷横向き (フロート) を使って、文書の一部を横向きにします。

4.13.3 多段組

このモジュールは、multicol パッケージを使用するもので、文書▷設定▷本文レイアウトにある二段組文書とは別の機能です。文書全体に二段組を行いたい場合には、二段組文書オプションを利用することをお勧めします。それ以外の場合には、本モジュールをご使用ください。

多段組中の脚注は、ページ下部に置かれ、各段の下ごとには置かれませんが、フロートに関して、編集▷フロート設定で段をまたぐオプションを有効にする必要があるという制限がある他は、各段において通常の使用をすることができます。

4.13.3.1 基本

文章中で二段組を使用したい場合には、段組を始める場所で挿入▷特別差込枠▷多段組メニューを使用して、多段組差込枠を挿入してください。二段組で印字されるべき文章は、全てこの差込枠の中に書いてください。

下記はその用例です。

The Adventure of the Empty House
by Sir Arthur Conan Doyle

It was in the spring of the year 1894 that all London was interested, and the fashionable world dismayed, by the murder of the Honourable Ronald Adair under most unusual and inexplicable circumstances. The public has already learned those particulars of the crime which came out in the police investigation, but a good deal was suppressed upon that occasion, since the case for the prosecution was so overwhelmingly strong that it was not necessary to bring forward all the facts. Only now, at the end of nearly ten years, am I allowed to supply those missing links which make up the whole of that remarkable chain. The crime was of interest in itself, but that interest was as nothing

to me compared to the inconceivable sequel, which afforded me the greatest shock and surprise of any event in my adventurous life. Even now, after this long interval, I find myself thrilling as I think of it, and feeling once more that sudden flood of joy, amazement, and incredulity which utterly submerged my mind. Let me say to that public, which has shown some interest in those glimpses which I have occasionally given them of the thoughts and actions of a very remarkable man, that they are not to blame me if I have not shared my knowledge with them, for I should have considered it my first duty to do so, had I not been barred by a positive prohibition from his own lips, which was only withdrawn upon the third of last month.

三段以上にするには、カーソルを多段組差込枠の内部において、挿入▷段数メニューを使用してください。この差込枠の中に、欲しい段数を書き入れます（三段組の場合

4 モジュール

には「3」と書きます)。
下記は三段組の例です。

It can be imagined that my close intimacy with Sherlock Holmes had interested me deeply in crime, and that after his disappearance I never failed to read with care the various problems which came before the public. And I even attempted, more than once, for my own private satisfaction, to employ his methods in their solution, though with indifferent success. There was none, however, which appealed to me like this tragedy of Ronald Adair. As I read the evidence at the inquest, which led up to a verdict of willful murder against some person or persons unknown, I realized more clearly than I had ever done the loss which the community had sustained by the death of Sherlock Holmes. There were points about this strange business which would, I was sure, have specially appealed to him, and the efforts of the police would have been supplemented, or more probably anticipated, by the trained observation and the alert mind of the first criminal agent in Europe. All day, as I drove upon my round, I turned over the case in my mind and found no explanation which appeared to me to be adequate. At the risk of telling a twice-told tale, I will recapitulate the facts as they were known to the public at the conclusion of the inquest.

お望みならば10段組まで組むことができますが、文書の読者にとっては、あまり気持ちのいいものではないでしょう。

4.13.3.2 段組中の段組

また、段中に段を組むこともできます。

The Honourable Ronald Adair was the second son of the Earl of Maynooth, at that time governor of one of the Australian colonies. Adair's mother had returned from Australia to undergo the operation for cataract, and she, her son Ronald, and her daughter Hilda were living together at 427 Park Lane.

The youth moved rest {sic} the man's life in the best society – moved in a narrow and had, so far as was conventional circle, for known, no enemies and his habits were quiet no particular vices. He and his nature unemotional. Yet it was upon Miss Edith Woodley, this easy-going young of Carstairs, but the aristocrat that death engagement had been came, in most strange broken off by mutual and unexpected form, consent some months between the hours of before, and there was ten and eleven-twenty no sign that it had left on the night of March any very profound feeling behind it. For the

Ronald Adair was fond of cards – playing continually, but never for such stakes as would hurt him. He was a member of the Baldwin, the Cavendish, and the Bagatelle card clubs. It was shown that, after dinner on the day of his death, he had played a rubber of whist at the latter club. He had also played there in the afternoon. The evidence of those who had played with him – Mr. Murray, Sir John Hardy, and Colonel Moran – showed that the game was whist, and that there was a fairly equal fall of the cards. Adair might have lost five pounds, but not more. His fortune was a considerable one, and such a loss could not in any way affect him. He had played nearly every day at one club or other, but he was a cautious player, and usually rose a winner. It came out in evidence that, in partnership with Colonel Moran, he had actually won as much as four hundred and twenty pounds in a sitting, some weeks before, from Godfrey Milner and Lord Balmoral. So much for his recent history as it came out at the inquest.

4.13.3.3 応用例

本節の用例は、多段組のさらに特別な機能を示すためのものです。

多段組機能についてさらに知りたい場合には、`LATEX` パッケージ `multicol` [解説文書](#) をご覧ください。

序文 多段組に序文を追記するには、カーソルを多段組差込枠内部に置き、挿入▷序文メニューを使用してください。序文は、この差込枠内に書いてください。序文のある文章の例を掲げます。

And the story continues and continues and continues and continues...

On the evening of the crime, he returned her daughter. Desiring to say good-night, from the club exactly at ten. His mother and she attempted to enter her son's room. The sister were out spending the evening with door was locked on the inside, and no answer could be got to their cries and knocking. The servant deposed that she heard him enter the front room on the second floor, generally used as his sitting-room. The unfortunate young man was found lying near the table. His head had been horribly mutilated by an expanding revolver bullet, She had lit a fire there, and as it smoked but no weapon of any sort was to be found she had opened the window. No sound was heard from the room until eleven-twenty, the hour of the return of Lady Maynooth and in the room.

最初の $\text{T}_{\text{E}}\text{X}$ コードの後に、節宣言コマンドを $\text{T}_{\text{E}}\text{X}$ コードとして使用すれば、節見出しを前書きとして使用することができます。例えば、

```
\subsection{小節見出し}
```

というコマンドは、小節を生成します。以下の例では、前書きが小々節になっています。

4.13.3.4 これは序文の小々節見出しです

A minute examination of the circumstances which separated the house from the road. served only to make the case more complex. Apparently, therefore, it was the young man In the first place, no reason could be given himself who had fastened the door. But how why the young man should have fastened the door upon the inside. There was the possibility that the murderer had done this, and did he come by his death? No one could had afterwards escaped by the window. The have climbed up to the window without leaving traces. Suppose a man had fired through the window, he would indeed be a remarkable drop was at least twenty feet, however, and able shot who could with a revolver inflict so a bed of crocuses in full bloom lay beneath. deadly a wound. Again, Park Lane is a frequented thoroughfare; there is a cab stand Neither the flowers nor the earth showed any sign of having been disturbed, nor were there within a hundred yards of the house. No one any marks upon the narrow strip of grass had heard a shot.

多段組を開始する部分に、必要とされる6行を確保出来るだけの垂直スペースがページに残されていない場合、多段組の前に改行が挿入されます。前書きの行数如何では、この必要行数を変更したいかもしれません。これは、カーソルを多段組差込枠

内部の（もしあれば）序文の後に置き、挿入▷改頁前の余白メニューを使用することで設定できます。この差込枠に、余白の大きさを「5cm」などと書き込んでください。

以下の例では、`7\baselineskip`を用いて、必要行数を7行に指定しています（コマンド`\baselineskip`は \TeX コードとして入力する必要があります）。

On the evening of the crime, he returned her daughter. Desiring to say good-night, from the club exactly at ten. His mother and she attempted to enter her son's room. The sister were out spending the evening with door was locked on the inside, and no answer could be got to their cries and knocking. The servant deposed that she heard him enter the front room on the second floor, generally used as his sitting-room. The unfortunate young man was found lying She had lit a fire there, and as it smoked near the table. His head had been horribly she had opened the window. No sound was mutilated by an expanding revolver bullet, heard from the room until eleven-twenty, the but no weapon of any sort was to be found hour of the return of Lady Maynooth and in the room.

周囲の余白 多段組の前後の余白は、長さ`\multicolsep`を変えることで変更することができます。例えば、 \TeX コードで書かれたコマンド

```
\setlength{\multicolsep}{3cm}
```

は、その値を3cmに変更します。この変更は、多段組が始まる前に指定しなくてはなりません。既定値は13ptです。

以下の例では、`\multicolsep`は2.5cmに指定されています。

All day I turned these facts over in my mind, endeavouring to hit upon some theory which could reconcile them all, and to find that line of least resistance which my poor friend had declared to be the starting-point of every investigation. I confess that I made little progress. In the evening I strolled across the Park, and found myself about six o'clock at the Oxford Street end of Park Lane. A group of loafers upon the pavements, all staring up at a particular window, directed me to the house which I had come to see. A tall, thin man with coloured glasses, whom I strongly suspected of being a plain-clothes detective, was pointing out some theory of his own, while the others crowded

4 モジュール

round to listen to what he said. I got as near him as I could, but his observations seemed to me to be absurd, so I withdrew again in some disgust. As I did so I struck against an elderly, deformed man, who had been behind me, and I knocked down several books which he was carrying.

[註] `\setlength` で指定した値は、再度変更されるまでは、以降のすべての多段組で使用されます。

改段 改段は、文章中の階段を行いたい位置に、`TEX` コードで、コマンド `\columnbreak{}` を挿入すれば、強制的に行うことができます。ただし、多くの場合、文章中に空白が挿入されることに注意してください。以下が用例です。

“You’re surprised to see me, sir,” said he, in a strange, croaking voice. I acknowledged that I was. “Well, I’ve a conscience, sir, and when I chanced to see you go into this house, as I came hobbling after you, I thought to myself, I’ll just step in and see that kind gentleman, and tell him that if I was a bit gruff in my manner there was not any harm meant, and that I am much obliged to him for picking up my books.” “You make too much of a trifle,” said I. “May I ask how you knew who I was?” AFTER THIS SENTENCE THE COLUMN BREAK IS FORCED.	“Well, sir, if it isn’t too great a liberty, I am a neighbour of yours, for you’ll find my little bookshop at the corner of Church Street, and very happy to see you, I am sure. Maybe you collect yourself, sir. Here’s BRITISH BIRDS, and CATULLUS, and THE HOLY WAR – a bargain, every one of them. With five volumes you could just fill that gap on that second shelf. It looks untidy, does it not, sir?”
---	---

段間隔 段幅は自動的に計算されますが、段の間の余白は指定することができます。これは、長さ `\columnsep` を変更することで実現できます。既定値は 10 pt です。以下は `\columnsep` を 3 cm に設定した例です。

My observations of No. 427 Park Lane did little to clear up the problem in which I was interested. The house was separated from the street by a low wall and railing, the whole not more than five feet high. It was perfectly easy, therefore, for anyone to get into the garden, but the window was entirely inaccessible, since there was no water pipe or anything which could help the most active man to climb it. More puzzled than

ever, I retraced my steps to Kensington. I had not been in my study five minutes when the maid entered to say that a person desired to see me. To my astonishment it was none other than my strange old book collector, his sharp, wizened face peering out from a frame of white hair, and his precious volumes, a dozen of them at least, wedged under his right arm.

垂直線 段間には、長さ`\columnseprule`の幅を持つ罫線が描かれます。罫線幅を0ptにすると（これが既定値です）、罫線は出力されません。以下の例では、段を分かつ線は2pt幅に設定されています。

“You’re surprised to see me, sir,” said he, in a strange, croaking voice.

I acknowledged that I was.

“Well, I’ve a conscience, sir, and when I chanced to see you go into this house, as I came hobbling after you, I thought to myself, I’ll just step in and see that kind gentleman, and tell him that if I was a bit gruff in my manner there was not any harm meant, and that I am much obliged to him for picking up my books.”

“You make too much of a trifle,” said I. “May I ask how you knew who I was?”

“Well, sir, if it isn’t too great a liberty, I am a neighbour of yours, for you’ll find my little bookshop at the corner of Church Street, and very happy to see you, I am sure. Maybe you collect yourself, sir. Here’s BRITISH BIRDS, and CATULLUS, and THE HOLY WAR – a bargain, every one of them. With five volumes you could just fill that gap on that second shelf. It looks untidy, does it not, sir?”

罫線は、コマンド`\columnseprulecolor`を再定義すれば、着色することができます。これは、多段組差込枠の前に $\text{T}_{\text{E}}\text{X}$ コードで以下のコマンドを挿入すれば、実現できます。

```
\renewcommand{\columnseprulecolor}{\color{red}}
```

このコマンド中の`\red`をお好みの色に変更してください。既定色とユーザー定義色の全てを使用することができます。既定色とユーザー定義色についての詳細な情報

は、取扱説明書『埋込オブジェクト篇』の「色付き表」の節をご覧ください。既定色に戻すには、以下のコマンドを挿入してください。

```
\renewcommand{\columnseprulecolor}{\normalcolor}
```

以下は、シアン色の罫線と 1 cm の段間隔を持つ例です。

“You’re surprised to see me, sir,” said he, in a strange, croaking voice.

I acknowledged that I was.

“Well, I’ve a conscience, sir, and when I chanced to see you go into this house, as I came hobbling after you, I thought to myself, I’ll just step in and see that kind gentleman, and tell him that if I was a bit gruff in my manner there was not any harm meant, and that I am much obliged to him for picking up my books.”

“You make too much of a trifle,” said I.

“May I ask how you knew who I was?”

“Well, sir, if it isn’t too great a liberty, I am a neighbour of yours, for you’ll find my little bookshop at the corner of Church Street, and very happy to see you, I am sure. Maybe you collect yourself, sir. Here’s BRITISH BIRDS, and CATULLUS, and THE HOLY WAR – a bargain, every one of them. With five volumes you could just fill that gap on that second shelf. It looks untidy, does it not, sir?”

4.14 段落様式

4.14.1 任意の段落形

L^AT_EX は、他のテキストプロセッサと同様に、既定で長方形の段落を使用します。しかし、例えば、ポスターや招待状、詩などの特殊な場合には、段落の形を好きなように変更することができます。このモジュールは、`shapepar` パッケージを使用して、1 ダースを超える段落形を提供するだけでなく、好みの形を定義するコマンドを提供します。

4.14.1.1 既定の形

`shapepar` パッケージは、以下の形を提供しています。

名称	説明	備考
CDlabel	丸い穴の開いたドーナツ型 (CD/DVD 大)	伸縮不可。多くの文字を入れられないことに注意。

名称	説明	備考
Circle	円	脆弱. 計算に失敗することも.
Diamond	菱形 (シンボリックに “diamond” と表記)	-
Heart	ハート型	-
Hexagon	六角形	-
Nut	ナット型 (丸い穴の開いた六角形)	-
Square	正方形	-
Star	五芒星	-
Candle	燃えるろうそく	-
Drop down/up	雫型/逆雫型	脆弱. 計算に失敗することも.
TEX	TEX ロゴ	-
Triangle up/down/ left/right	各方向の三角形	-

段落に形を適用するには、LyX ツールバー中の環境プルダウンボックスから形を選んでください。

4 モジュール

以下は、ナット型の段落の例です。

 Lorem ipsum dolor sit amet,
 consetetur sadipscing elitr, sed
 diam nonumy eirmod tempor invidunt
 ut labore et dolore magna aliquyam erat,
sed diam volup- tua. At vero eos
et accusam et justo duo do-
lores et ea re- bum. Stet clita
kasd gubergren, no sea takimata
sanctus est Lo- rem ipsum do-
lor sit amet. Lo- rem ipsum dolor
sit amet, consetetur sadipscing elitr, sed
diam nonumy eirmod tempor invidunt
ut labore et dolore magna aliquyam
erat, sed diam voluptua.

【註】 shapepar 段落は改頁にまたがることはできません。

shapepar は、段落中にどれだけ文章が含まれようとも、常に形を維持しようとして、したがって、段落の大きさは、含まれている文章量によって変化しますので、段落の大きさがページ余白にはみ出してしまふことがあります。寸法が伸縮することを示すために、文章量を二倍にした例を見てみましょう。

Lorem ipsum dolor sit amet, consetetur
 sadipscing elitr, sed diam nonumy eirmod
 tempor invidunt ut labore et dolore magna ali-
 quyam erat, sed diam voluptua. At vero eos et ac-
 cusam et justo duo dolores et ea rebum. Stet clita kasd
 gubergren, no sea takimata sanctus est Lorem ipsum
 dolor sit amet. Lorem ipsum dolor sit amet,
 consetetur sadipscing elitr, sed diam nonu-
 my eirmod tempor invidunt ut labore et
 dolore magna aliquyam erat, sed diam volup-
 tua. Lorem ipsum dolor sit amet, consete-
 tur sadipscing elitr, sed diam nonumy eir-
 mod tempor invidunt ut labore et dolore
 magna aliquyam erat, sed diam voluptua. At
 vero eos et accusam et justo duo dolores et ea rebum.
 Stet clita kasd gubergren, no sea takimata sanctus est
 Lorem ipsum dolor sit amet. Lorem ipsum dolor sit
 amet, consetetur sadipscing elitr, sed diam nonu-
 my eirmod tempor invidunt ut labore et dolore
 magna aliquyam erat, sed diam voluptua.

ただし、CDlabel 形は例外です。これは、CD/DVD の寸法に収まらなくてはならないので、伸縮しません。したがって、形に収まる文章量は限られています。

drop 形と circle 形は脆弱で、文章量によっては寸法計算が失敗する可能性があります。この場合、“Arithmetic overflow” という L^AT_EX エラーが発生します。この時は、文章の一部を追加したり削除したりするか、どれもうまくいかなければ別の形を使用してください。

先端のある形では、先端部の文字列で、L^AT_EX のハイフネーションルーチンが失敗することがあります。したがって、この部分の文章には、ハイフネーション位置 () を指定することが往々にして必要になります。

Shapepar 段落は、常に中央揃えか、左の縁が左ページ余白に接するように配置されます。したがって、これらの段落を揃えるのに段落ダイアログを使用することはできません。解決方法としては、段落をミニページか parbox の中において、このボックスを揃えることです。ここでの問題は、ボックスの正しい幅を見出すことです。もしボックスの幅が広すぎれば、ボックスを中央揃えにしたとしても、ボックスの左側が付いてしまうので中央揃えになりません。そこで、正しい幅を見出すまでは、若干

4 モジュール

試行錯誤しなくてはならないかもしれません。以下は、二つのハート形の段落です。一つ目は揃え指定がなく、二つ目は右揃えにしています。

Lorem ip- sum dolor
 sit amet, conse- tetur sadipscing
 elitr, sed diam nonumy eirmod tempor
 invidunt ut labore et dolore magna ali-
 quyam erat, sed diam voluptua. At vero
 eos et accusam et justo duo dolores et
 ea rebum. Stet clita kasd guber-
 gren, no sea takimata sanctus
 est Lorem ipsum do-
 lor sit amet.



Lorem ip- sum dolor
 sit amet, conse- tetur sadipscing
 elitr, sed diam nonumy eirmod tempor
 invidunt ut labore et dolore magna ali-
 quyam erat, sed diam voluptua. At vero
 eos et accusam et justo duo dolores et
 ea rebum. Stet clita kasd guber-
 gren, no sea takimata sanctus
 est Lorem ipsum do-
 lor sit amet.



4.14.1.2 任意の形

ユーザーは、どのような形であれ、定義することができます。これを手動で行うのは、全ての座標を指定しなくてはならないので、かなりの労力を必要とします。しかし、以下のように、コンピューターに座標を計算させる方法もあります。

1. プログラム [Jpgfdraw](#) を導入します。
2. Jpgfdraw 内で、TeX/LaTeX▷Settings▷Set Normal Size メニューで文書中で使用しているフォント寸法を選択します。例えば、この文書は 12 pt を使用しています。
3. 形を描きます。

4 モジュール

4. TeX/LaTeX▷Shapepar メニューを使用して、形の座標をエクスポートします。現れるダイアログで、形の外枠を座標に使うか、パス自体を使うかを指定します。

1

座標は TeX ファイルに書き込まれます。これを LyX 文書で使用するには、

1. 段落の最初の文字の前（もしくは空の段落内）にカーソルを置きます。
2. LyX の挿入▷ファイル▷子文書メニューでファイルを選択し、取り込みの型として Input を選択してください。

【註】 変更した形は、現在の段落にのみ適用されます。次の段落では全て通常に戻ります。したがって、段落定義ファイルは、形を変えたい段落全てで読み込まれなくてはなりません。

以下の用例をご覧ください。

Lorem ipsum dolor sit amet,
sed diam nonumy eirmod
labore et dolore magna
diam voluptua. At ve-
et justo duo dolores
clita kasd gubergren,
sanctus est Lorem
amet. Lorem ipsum
consetetur sadipscing
nonumy eirmod tempor
et dolore magna aliquyam

consetetur sadipscing elit,
tempor invidunt ut
aliquyam erat, sed
ro eos et accusam
et ea rebum. Stet
no sea takimata
ipsum dolor sit
dolor sit amet,
elit, sed diam
invidunt ut labore
erat, sed diam voluptua.

【ヒント】 Jpgfdraw で複雑な形を描く場合、まず形を好きなプログラムで描き、出来上がった画像を Jpgfdraw に読み込む方法があります。この場合、画像の外枠に沿って、直線またはスプラインを描くだけですみます。

任意の段落形モジュールは、shapepar と Shapepar の二つのコマンドも提供していて、任意の形を座標毎に定義するのに使用することができます。² これらのスタイルがどのように使われていて、どのような設定ができるのかについての詳しい情報は、shapepar パッケージの**解説文書**をご覧ください。

¹詳細については、<http://www.dickimaw-books.com/apps/jpgfdraw/manual/shapepar.html> を参照してください。

²これらのコマンドは、上述の全ての形で内部的に使用されています。

4.14.2 ぶら下げ段落

このモジュールは、ぶら下げ段落様式、すなわち段落の第1行以外はすべて字下げする段落様式を追加します。これは、LyX 文書中には反映しない可能性もありますが、出力には反映されます。

4.14.3 頭文字 (ドロップキャップ)

➤ のモジュールは、ドロップキャピタル段落様式である頭文字段落環境を追加します。これを用いれば、望む場所をドロップキャピタルにすることができます。すると、挿入メニューに以下の3つの項目が追加されます。

頭文字 ドロップキャピタル用の差込枠

頭文字の残り ドロップキャピタル化する単語の残り部分

オプション ドロップキャピタルの整形オプション

様式の見た目を調整するには、ヘルプ▷埋込オブジェクト篇▷本文回り込みオブジェクト▷頭文字を参照してください。

4.15 テキストマークアップ

4.15.1 ハイフネート可能なテキストマークアップ (Soul)

このモジュールは、soul パッケージを使って、ハイライトしたり均等配置したり、打消線や下線を引いたり、大文字化/スモールキャップ化したりするための文辞様式を定義します。

文辞特性ダイアログで提供されるマークアップと違い、soulでマークアップされる単語はハイフネートされます。soul 解説文書を参照してください。

4.15.2 論理マークアップ

このモジュールは、論理マークアップ用にいくつかの文字様式(名詞・強調・ストロング・コード)を定義します。これらは、編集▷特別文辞様式でアクセスできます。

5 書誌情報

BibTeX あるいは Biblatex の基本的な使用法は、『ユーザーの手引き』の「文献データベース」の節に説明があります。以下の各小節では、LyX でサポートされている書誌情報の特別な機能について説明することとします。

5.1 他の引用様式

標準の BibTeX は、引用された著作を参照するのに、([12] のように) 数字を使用します。しかし、多くの科学分野で、これとは異なる引用様式が用いられています。もっともよく見られるのは、著者-出版年様式 (例: Knuth 1984a) です。LyX は、この様式を提供する 3 つのパッケージ `biblatex`・`natbib`・`jurabib` をサポートしています。これらのパッケージには、それぞれ長所短所がありますが、ここでは詳細に立ち入る紙幅がありません。単純な著者-出版年様式 (あるいは著者-番号) 様式が欲しい場合や、無数にある `natbib` 用スタイルファイルのうちどれかを使用したい場合には、定評のある `natbib` パッケージがおそらくよい選択でしょう。法学でよく用いられる引用様式を探している場合には、`jurabib` パッケージの利用を考えるとよいかもしれません。整形を完全に制御したい場合や、`ibidem`・脚註引用・タイトル全引用・400 BC などの) 複雑な日時形式・英語以外への完全な現地語化のような複雑な機能を使用したい場合には、機能面で他のパッケージを凌駕している `biblatex` をぜひとも考えるべきでしょう。ただし、同時に `biblatex` はたいへん重く、リソース消費量も多いことに注意してください。

LyX での各パッケージの取扱い方は、ほぼ同じです。文書▷設定メニューを開き、書誌情報項目の様式整形子において基本 (BibTeX)・Biblatex・Biblatex (Natbib モード)・Natbib (BibTeX)・Jurabib (BibTeX) のいずれかを選択してください (各選択肢についての詳細は、ユーザーの手引きの第 6.5.3 節を参照のこと)。これらのパッケージによって、引用ダイアログに機能が追加され、参照の様式が選択できるようになります (「Knuth 1984」, 「Knuth (1984)」, 「Knuth, 1984」, 「1984」など)。各パッケージは、専用にデザインされたスタイルファイルを必要とすることに注意してください。これらは全てそれ自身のスタイルファイルを同梱していますが、他にもたくさん

の追加されたスタイルファイルがありますし、natbibに至っては、対話的なスタイルファイルビルダーさえ存在します¹。

5.2 区分された書誌情報

場合によっては、書誌情報をいくつかの節に分ける必要があることがあります。たとえば、あなたが歴史家であるとすれば、出典と学術文献を分ける機能は、おそらく「必需品」でさえあります。残念ながら、BibTeX 自身ではこのようなことができません。しかし、適切な L^AT_EX パッケージの助けを借りると、BibTeX をこのような必要性を満たすように拡張することができます。

LyX は、このようなパッケージの一つである bibtopic² をそのままサポートします。このパッケージの長所は、(multibib のような他のパッケージと比較すると) 新規の引用コマンドを定義する必要がないことです。そのかわりに、書誌情報の違う節には、その節向けの項目を収録した別の文献データベースを用意する必要があります。たとえば、書誌情報を「出典」の節と「学術文献」の節に分けたい場合には、それぞれの文献データベース、たとえば sources.bib と scientific.bib とを作成する必要があります。

文書▷設定を開いて書誌情報の下の区分された書誌情報をチェックしてください。すると、書誌情報の各節に一つずつ、複数の書誌情報を挿入することができます。上記の例では、BibTeX 書誌情報 sources.bib と二つめの文献データベース scientific.bib を挿入します。各節に同じスタイルを用いるか異なるスタイルを用いるかは自由です。くわえて、書誌情報の節に、指定したデータベースの「引用されたすべての参照」を載せる（これが既定です）か、「引用されなかったすべての参照」を載せるか、あるいは「すべての参照」を載せるかを選択することができます。これは、もし書誌情報を「本文に引用された出典」・「本文に引用されていない出典」・「学術文献」の三つの節に分割したい場合に、便利でありましょう。各節の見出しは、通常の節や小節として入れることができます。bibtopic は、書誌情報用の見出しを取り除いてしまいますので、手動でこれを追加しなくてはなりません（たとえば章*や節*などをお使いください）。

Biblatex の場合には、手順が多少異なります。bibtopic は Biblatex では動作しませんので、Biblatex 使用中は前述の区分された書誌情報オプションは無効になります。しかしながら、Biblatex は区分された書誌情報を生成する独自の方法を提供しています。書誌情報項目の keywords 項目オプションにキーワードを書き込み、書誌情報セ

¹<ftp://ctan.tug.org/tex-archive/macros/latex/contrib/custom-bib/> をご参照ください。

²<ftp://ctan.tug.org/tex-archive/macros/latex/contrib/bibtopic/> から入手可能です。

クッションをキーワードでフィルターに掛けることもできますし (BIBLATEX が生成した書誌情報ボタンを右クリックすると開くダイアログで、オプションフィールドに keyword=指定キーワードと書き込みます), あるいは, 上述のオプションフィールドに, 例えば type=book や nottype=collection と入力することで, (book や article などの) 項目型によりフィルターを掛けることもできます. あるいは, 各項目を割り当てるための, いわゆる「書誌情報カテゴリ」を作ることもできます. 詳細については, Biblalex 取扱説明書の「区分された書誌情報」の節をご参照ください.

5.3 複数の書誌情報

各部・章・節・小節・子文書ごとの書誌情報など, 複数の書誌情報も L_YX ではサポートされています. これを有効にするためには, 文書▷設定▷書誌情報を開き, 複数の書誌情報コンボボックスで必要な単位 (「節毎」など) を選択してください. それから, 執筆中文書の (節などの) 各単位に, Bib(la)T_EX 書誌情報を加えてください. 出力時に, 現在の単位に含まれる文献参照のみが含まれた, 分割された書誌情報が生成されます.

BibT_EX を使用する場合には, 複数の書誌情報を生成するのに, chapterbib パッケージか bibtopic パッケージが使用されます. 「子文書毎」を選択し, 区分された書誌情報にチェックを入れない場合には, 前者のパッケージが使用されますが, その他の場合には, すべて bibtopic パッケージが使用されます³. bibtopic パッケージが使用される場合, L_YX は特定の単位を生成するために, その単位を bibtopic の `\begin{btUnit}` と `\end{btUnit}` で囲みます. このアプローチには, いくつかの制限があることに注意してください. まず, 全ての文献参照はいずれかの btUnit の中になくってはなりません. そのため, 定義された単位の外にある引用 (たとえば「節毎」を使用した場合の, 第1節の前) は, 「???' と表示されてしまいます. (このような制限を回避するなどのため) `\begin{btUnit}` と `\end{btUnit}` を T_EX コードで入力して, 自前の単位を生成することもできます. ただし, btUnit は入れ子にすることができないことに注意してください. 第二の制限は, bibtopic は, 引用のないものも含めて, つねに書誌情報データベースの参照すべてを保持してしまうということです. これにより奇妙なラベルが付されることがあります (例えば, データベース中に Miller という著者の 2014 年の著作が 2 つあり, その 1 つだけが引用されている場合, ラベルは「Miller 2014b」などとなってしまいます).

Biblalex を使用する場合には, refsection オプションが代わりに使用されます.

³別のアプローチとして, もし T_EX コードを使っても良いという気があるならば (第 2.3 節参照), bibunits パッケージを使用する方法があります.

5 書誌情報

特定の単位が必要な時には、それらの開始時に $\text{T}_{\text{E}}\text{X}$ コードで `\newrefsection` または `\newrefsegment` と書きます。単位の終了箇所も制御したい場合には、代わりに `\begin{refsection} ... \end{refsection}` 環境を使ってください。ただし、`refsection` 関連は入れ子にできないことに注意してください。現行単位毎の書誌情報に加えて、`Biblatex` はすべての書誌情報を、後で一箇所（例えば文末）に出力する簡単な方法も提供しています。そのためには、`Bib(la)TeX` 差込枠ダイアログの内容コンボボックスで「文献参照単位全て」を選択してください。この話題についての詳細は、`Biblatex` の取扱説明書を参照してください。

6 ブリット

ALLAN RAE 文

6.1 はじめに

LyXには、ブリットの形が216個準備されていて、簡単なダイアログから呼び出すことができます。このダイアログを使えば、箇条書き(記号)の各階層で使うブリットの形を変更することができます。この設定は文書全体に共通なので、段落ごとに別のブリットを指定することはできません¹。

6.2 外見

メニュー項目の文書▷設定を選択してダイアログを開き、**ブリットタブ**を選択してください。

このダイアログは、ブリット形の一覧表を提示します。表左のボタン列で、6つの各ブリット形パネルを切り替えることができます。上部のボタン列では、変更するブリット階層を選択できます。

任意のブリットを選択すると、表下のテキスト入力欄が有効となり、そこにブリットの形の \LaTeX 表記を入力することができます。このテキストを変更すると、同時に、必要となるパッケージを \LaTeX プリアンブルで指定する必要があります。

6つのパネルは、それぞれが必要とするパッケージによって分類されています。下記は、フォント名と \LaTeX パッケージの対応表です。

¹正確には可能ですが、手動で行う必要があります。

フォント	必要とするパッケージ
標準	L ^A T _E X 本体
数式	amssymb.sty
絵文字 1	pifont.sty
絵文字 2	pifont.sty
絵文字 3	pifont.sty
絵文字 4	pifont.sty

L^AX は、ユーザーが導入されていないパッケージのブリットを使用したとしても、警告を発生しません。ファイルを開覧したり印刷したりしようとしたときに L^AT_EX エラーが出るようならば、パッケージが導入されていない可能性が大です²。

6.3 使用法

変更したいブリット階層を選んで、ブリット形と寸法を選択してください。何らかの変更を加えたとしても、L^AX 上には反映されませんが、文書の表示を行えば確認されます。

ブリット形は、適切な階層のブリットボタンをマウスで右クリックすれば、簡単に既定値に初期化することができます。

もし本当に段落ごとに異なるブリットの形を使いたいならば、T_EX コードを使って少し手を汚す必要があります。この場合も、ブリット選択ダイアログを使えば、さまざまな形の L^ATeX コードを確認することができるので手助けとなるでしょう。任意に設定した段落をつくるには、以下のような選択肢があります。

- ‡ 特定の階層の新しいブリット形を指定するには、L^AT_EX コマンド `\renewcommand{}{}` を使用します。あとで現在のブリット形に戻すことができるように、現在のブリット形を保存しておく必要もあります。ここで使っている簡条書きでは、第 1 階層のブリットを変更するために、以下の L^AT_EX コードが使用されています。

```
\let\savelabelitemi=\labelitemi
\renewcommand\labelitemi[0]{\small\(\sharp\)}
```

- ‡ 簡条書き (記号) の階層は、`\labelitem` コマンドの一部にローマ数字で指定されていることに注意してください。

²L^AX が、パッケージがなくてもブリットを使用することを許すのは、編集をここで行ったとしても、別のところへエクスポートする可能性があるからです。

- ★ 各項目のブリットを指定するには、 \TeX コードとして入力したブリット形を、行頭に「ユーザ設定」項目差込枠（挿入▷ユーザ設定項目）に入れて挿入してください。たとえば、この項目の行頭には、 $\backslash(\star\backslash)$ が指定されています。

文書全体で使用するブリット形が有効であるようにするために、`labelitem` を元の設定に戻す必要があります。そのためには、以下のようにします。

```
\renewcommand\labelitemi[0]{\savelabelitemi}
```


7 補完ツール

7.1 複合文書

7.1.1 一般的な操作

たくさんの節を含んだ大きなファイルで作業をする場合、文書をいくつかのファイルに分割した方が往々にして便利です。場合によっては、表部分にはしょっちゅう変更が加えられるけれども、その前の文章はつねに同じであるようなケースもあるでしょう。このような場合には、複合文書を利用することを、まずご検討ください。たとえば、科学論文は通常、導入部・所見・結果・議論・結論の五つの節から成ります。これら各節を別々の L_AT_EX ファイルにして、タイトル・著者・概要・書誌情報等と、これら五つのインクルード¹ファイルを持つ、もう一つ別の「マスター」ファイルを作成するようにすることができます。これらの各ファイルは、単にマスターファイルにインクルードされるだけでなく、単独でも出力したり印刷したりすることのできる完全な L_AT_EX ファイルであるように注意することが重要です。しかも、これらの各ファイルは、同じ文書クラスでなくてはなりません。article クラスと book クラスを混在させるようなことはなさないでください。L^AT_EX ファイルをインクルードすることもできますが、このファイルにはプリアンブルがあってはなりません (`\begin{document}`行とその前のすべて、および `\end{document}`行は削除しておかなければなりません)。さもないと、DVI ファイルを作成する際にエラーが発生するでしょう。

L_AT_EX では、挿入 ▢ ファイル ▢ 子文書 で簡単にファイルをインクルードすることができます。このメニューを選んでクリックすると、カーソルの現在位置に小さな枠が挿入されます。この枠をクリックすると、ダイアログが現れるので、そこでインクルードするファイルとインクルード方法を選択することができます。

ファイル選択ボックスは、いまや自明でしょう。インクルード方法には、「include」・「input」・「verbatim」の三つがあります。「include」と「input」は多くの点で類似していますが、いくつかの大きな違いがあります。

¹ (訳註) 「取り込む」の意です。原語「include」に対する訳語として、プログラミング用語の「インクルードする」以外に適語がないので、ここではプログラミング用語をそのまま使用します。

7 補完ツール

1. 「include」されたファイルは新規頁から組版されるのに対し、「input」されたファイルは現在のページから組版されます。
2. 「include」されたファイルは、それら自身がさらに他のファイル（「孫ファイル」）を「include」することはできません。一方で、「input」は無限に下層 input を持つことができます。
3. 「include」は選ばれた「include」ファイルの出力しか許しませんが、実際の（ページ割りなどの）カウンタや参照は保持されます（詳細については埋込オブジェクト説明書の子文書の節を参照のこと）。したがって、子文書にアウトソースされた章については「include」が好まれる方式ですが、任意のファイル継承がある場合には「input」の方が良いでしょう。

「verbatim」インクルードファイルでは、ファイルの内容がそのまま（つまり「verbatim」モードで）固定幅タイプライタフォントで組版されます。通常、このファイルの空白は、通常の L_AT_EX での空白の取扱いとは違って、連続した複数の空白がそのまま出力されるものの、単なる見えない空白です。ただし、出力の空白に印付けをするチェックボックスを有効にすると、空白が存在することを明確にするために空白の印が出力されます。

一般的に、親ファイルは、組版前に完全な L_AT_EX ファイルに変換されますが、インクルードされるファイルはプリアンブル情報を全く持たない L_AT_EX ファイルに変換されます。

7.1.2 ファイル間の相互参照

この節の内容は、すでに古くなっています。既定のマスター文書についてと、マスターが開いているときに子文書をどう開くかについて記述する必要があります。
[[要修正]]

異なるファイル間に相互参照を設定することも可能です。まず、すべての関係するファイルを開いてください。例として、2つのファイルを扱う場合を考え、各ファイルを A・B と呼びます。B が A にインクルードされるファイルだとします。A にラベルを入れて、B で参照したいとしましょう。文書 B で相互参照ダイアログを開いて、使用する「バッファ」を選んでください。

7.1.3 すべての子文書での書誌情報

子文書を使って作業する場合、全体の書誌情報を文末に一つだけ置きたいのだけでも、子文書単独で出力する時に限り、子文書分を抽出した書誌情報が付けられるようにしたいことがあります。以下にその方法を説明します。

親文書には、全体の書誌情報が出力されるべき場所（親ファイルでも子ファイルでも）に、書誌情報差込枠を挿入するだけです。書誌情報差込枠を親ファイルに置いた場合、文献参照はすべての子文書に継承されるので、各子文書の引用ダイアログで使用できるようになります。

子文書限定の書誌情報については、子文書内の、子文書が独立にコンパイルされた時に書誌情報が出力されるべき場所に、書誌情報差込枠を挿入します。その際、この差込枠を、たとえば「子文書のみ」などと名付けた派生枝の中に置く（挿入▷派生枝▷新規派生枝を挿入...）のが肝要です。そして、子文書の中でこの派生枝を有効にし（文書▷設定...▷派生枝）親文書で無効にします（文書▷設定...▷派生枝）。こうすることで、子文書中の書誌情報は、親文書中では無視され、子文書中では有効になります。

複数の書誌情報が必要な場合（親文書中でも子文書ごとに書誌情報を出力するなど）は、第 5.3 節を参照してください。

7.2 LyX 書庫

しばしばユーザーは、出版社に送ったり、共著者とやりとりするために、LyX ファイルを、その依存する全画像（及び他のファイル）と「同梱」する必要に迫られることがあります。LyX には、この過程を自動化する Python スクリプト (`lyxpak.py`) が含まれています。これを使うには、`zipfile python` モジュールまたは `tarfile python` モジュールが、システムに導入済みであることが必要です。既定設定では、Unix 系システムでは、`gzip` 圧縮 `tar` ファイル形式が優先され、Windows では `zip` 形式が優先されます。

LyX の環境構成時には、「LyX 書庫」を書き出すことができるようにスクリプトが準備され、ファイル▷書き出しの下で利用可能となります。

ツール▷設定▷ファイル処理▷変換子の LyX->LyX 書庫 変換子に、`-t` スイッチや `-z` スイッチを加えることで、プラットホームとは独立に、`tar` 形式や `zip` 形式を強制することができます（スイッチは、スクリプト名の前ではなく、後ろに加えてください）。

8 L_YX とインターネット網

L_YX は、これまでも長い間、HTML を始めとするウェブフレンドリーな文書諸形式への書き出しをサポートしてきました。しかしながら、第 2.0 版以前では、HTML への書き出しは、常に外部変換子を用いることで実現されていました¹。これらは 2 つのグループに大別できます。1 つは、`htlatex`・`html2latex`・`plastex` のように、L_AT_EX を中間形式として用い、その L_AT_EX を生成するのに L_YX を用いるもの。2 つめは、`elyxer` のように、直接 L_YX ファイルに働きかける変換子です。これらは、それぞれに長所・短所を有しています。

L_AT_EX ベースの変換子は、原則として、L_AT_EX が書き出す文書についてのすべてを知っているという長所があります。この種の変換子は、例えば、特定のコードのくぐりや、L_YX によって生成されたのか、T_EX コードで書かれたのかについては気にしません。これらの変換子は、`aux` ファイルやカウンタや参照を取り扱うことができ、`BibTEX` が生成する `bb1` ファイルも利用することがよくあります。一方で、L_AT_EX は解析の非常に難しい言語であり—T_EX 自身だけが T_EX を理解することができると言われていたりすることがあります—、L_AT_EX ベースの変換子は、非常に簡単な文法に見えるものでさえ飲み込まず、複雑なものになると全く歯が立たないこともあります。例えば、この現行を書いている時点では、取扱説明書『埋込オブジェクト篇』は、他の変換子では書き出しができるのに、`htlatex` では行うことができません。他には、数式をどう取り扱うかという問題があります。これらの変換子の典型的なものは、数式を小さな画像に変換して、HTML 文書からこれらの画像へのリンクを張ります。これらの画像は、実は L_AT_EX によって生成されているので、正確です。しかしながら、これらはうまく伸縮できないので、実際の文書の中で表現されるように—フォント寸が凡そ正しくなるように—見せようとする、大変な苦勞を伴います。

Alex Fernandez の `elyxer`(<http://pinchito.es/elyxer/>) は、これらの問題の一部を解決します。特に、これは数式を画像に変換するのではなく、HTML と CSS を組み合わせて数式を生成するので、生成される数式は完璧な大きさに伸縮されて表示されます。しかしながら、ちょっとだけ複雑な数式でさえ、`htlatex` よりも質が落ちます。

¹L_YX で外部変換子を使うことについての詳細は、取扱説明書『カスタマイズ篇』の「変換子」の節をご覧ください。

これはHTMLの限界を示しています²。さらに重要なことは（少なくとも第1.1.1版においては）、eLyXerは数式マクロを一部しかサポートしておらず、ユーザ定義の段落様式や文字様式をサポートしていません。これらの制約によって、LyXユーザが生成する文書の多くにとっては、eLyXerは不向きなものになっています。もちろん、原理的に、これらの問題を解決することもできたのですが、LyX開発陣は別の方法を採ることとし、LyXがL^AT_EXやDocBook、平文を書き出すことができるのと同様に、LyX自身にXHTMLを書き出す能力を身につけさせることとしたのです。

この原稿執筆時点では、XHTML出力は開発途上であり、おそらく「実験的」機能として解されるべきでしょう³。それでもこのアプローチは、他の2つの方法よりも、潜在的な利点がいくつもあるので、開発陣はこの方法を選択したのです。

これらの利点は、主に、XHTML出力ルーチンがLyXの一部となっているため、LyXが出力する文書についてLyXが関知しているところのものは、すべて知り得るところから来ています⁴。つまり、XHTML出力ルーチンは、(文書構造に表示されるような形での)目次や、それぞれの段落様式に関連付けられたカウンタや、ユーザ定義の様式についても知っているのです。XHTML出力ルーチンは、国際化についてLyXが関知しているものも知っているのです。ちょうどその時に有効になっている言語に依存して、「Chapter 1」と出力されたり、「Kapitel 1」と出力されたりするのです。

汎く一般的に、出力ルーチンは、LyXが文書レイアウトについて関知しているものは知っており、したがって、文書が画面上にどのように描画されるかも知っています。文書をXHTMLに出力するときには、この情報を利用します。特に、LyXは、文書を画面上に描画するとき使用するレイアウト情報に対応したCSSスタイル情報を自動的に生成します。もし節見出しが、LyX中でサンセリフ体かつボールド体であるとすると、(既定では)ウェブブラウザに表示した時にも、サンセリフでボールドになります。これは、節のような定義済み様式だけでなく、ユーザ定義のものも含めて、すべての様式でそうです。実際、XHTML出力ルーチンは、ユーザ定義の段落・文字様式と、LyX自身の定義済み様式とを区別しません。どちらの場合も、LyXが様式について関知しているところのものは、全てレイアウトファイルに含まれています。これは、脚注のような定義済み差込枠や、各種ユーザ設定差込枠でもそうです。

²とはいえ、eLyXerでは、数式用にjsMathとMathJaxを使用することもできます。ただし、この設定はグローバルにしか設定することができず、バックエンドで実行させるサーバへのアクセスが利用できる必要があります。

³LyXソースツリーもしくは[オンラインでのアクセス](#)にあるdevelopment/HTML/HTML.notesファイルには、通常、XHTML出力の現状についての最新の情報が収録されています。[バグ追跡ページ](#)にあるXHTML関連バグ一覧も参照のこと。

⁴他には、これらのルーチンがLyX内部にあるため、LyXファイル形式や既存の差込枠の文法に変更があっても、影響を受けずに済むという利点もあります。

結果として、XHTML 出力は、レイアウトファイルやモジュールを通して、 \LaTeX 出力と全く同じように、ユーザが設定したり拡張したりすることができるのです。詳細については、取扱説明書『カスタマイズ篇』の第5章をごらんください。

本章の残りでは、XHTML 出力についての、さらに詳細な情報について取り扱い、その制約や、制約の回避の仕方を取り扱います。

8.1 XHTML における数式出力

\LaTeX では、数式描画に4つの方法を選択できます。以下に示すように、それぞれに利点と欠点があります。

- MathML

MathML は、ウェブ上で数学を用いるために、特にデザインされた XML の一方言であり、これをサポートするブラウザ上では、通常、特に美しい表現が得られます。欠点は、すべてのブラウザが MathML をサポートしているわけではないという事と、Firefox のような Gecko ベースのブラウザでも、サポートが完璧ではないという事です。

例えば、数式が `xymatrix` パッケージを使用していたり、 \TeX コードを使用しているなどで、もし \LaTeX が数式を MathML として描画できないとすると、 \LaTeX は数式を代わりに画像として出力します。

- HTML

前述のとおり、`elyxer` は、数式を CSS で装飾した HTML として出力します⁵。簡単な数式では、これは非常にうまく行きますが、少し複雑な数式になると、`elyxer` は歯が立たなくなります。それでも、この方法は非常に広くサポートされているという利点があり、かなり簡単な数式を少しだけ含んでいるような文書に適しているでしょう。

例えば、数式が `xymatrix` パッケージを使用していたり、 \TeX コードを使用しているなどで、もし \LaTeX が数式を HTML として描画できないとすると、 \LaTeX は数式を代わりに画像として出力します。

- 画像

`htlatrix` のように、 \LaTeX は数式を画像としても出力できます。この画像は、実は、自動プレビューで使用されているものと全く同じです⁶。この方法の利点は、画像が他ならぬ \LaTeX によって生成されているので、非常に正確であるこ

⁵ \LaTeX は、`elyxer` からの HTML 出力のために、CSS の一部を拝借しています。

⁶ただし、画像を出力に使うために、自動プレビューを入りにしておく必要はありません。

とです。欠点は、前述したとおり、画像がビットマップなので、伸縮が全くうまく行かなかったり、コピーが出来なかったりといったことがあります。

画像の寸法は、文書▷設定▷形式の「数式画像の伸縮」パラメータを変更することで制御できます⁷。

もしLyXが、何らかの理由によって数式の画像を生成できない（例えば、必要なL^AT_EXパッケージが導入済みでないなど）ときは、回避策として、生のL^AT_EXを出力します。

- L^AT_EX

最後に、LyXは、数式をL^AT_EXとしても出力することができます。この方法は、最終手段としての出力方法である一方で、HTML文書に埋め込まれたL^AT_EXを描画するのにJavaScriptを使用する、jsMathのようなツールとともに使用することもできます。LyXは、jsMathで必要とされているのと同じように、L^AT_EX部分を、span（行内数式用）またはdiv（別行立て数式用）にclass='math'を指定したもので、包んで出力します。

これらの出力方法のいずれかを文書▷設定▷形式で選択しなくてはなりません。既定では、LyXはMathMLで出力を行います。これは、文書全体に適用される設定となります。

最後に、LyXでは、例えば、MathMLではあまりうまく出力されないような差込枠に対して、別の出力方法を選択できる選択肢が用意されています⁸。

8.2 書誌情報と引用

XHTML出力は、書誌情報と引用を完全にサポートしています。

引用ラベルは、LyX画面上のラベルを生成するのと同じ機構で生成されているので、出力中のラベルは、LyX中の外見と同質またはそれよりも上質な仕上がりとなっています。連番引用を使用している場合には、LyXは、画面上で表示されるような各カッコに囲まれた引用キーを単純に表示するのではなく、[1]や[17]などの連番ラベルを出力します。著者-出版年形式の引用を使用している場合で、同じ著者-出版年の組み合わせが複数ある場合、LyXは、BibT_EXが行うように出版年に小文字を付します。ラベルは、書誌情報項目とともに印字されます。現在のところ、角括弧を丸括弧に変更するなど、ラベルの外見を変更する手段はありませんのでご注意ください。

⁷必要な方のために述べておくと、これは、画像の解像度をdpi（1インチあたりドット数）で制御するためのもので、75dpiを基準としています。

⁸その他にも、LyXは、差込枠にT_EXコードが含まれていると画像出力に回帰するので、数式中に $a = b$ のような無害なT_EXコードを挿入することによって、画像出力を強制することができます。

書誌情報出力は、引用ダイアログで文献情報表示を取り扱うのと同じ機構で取り扱われているので、XHTML出力は、引用ダイアログで特定の文献項目を表示させた時のものに非常によく似ています。整形の仕方は、レイアウトファイルや望ましくはモジュールでカスタマイズすることができます。詳細については、取扱説明書『カスタマイズ篇』をご覧ください。

現在のところ、主だった瑕疵は、相互参照された情報が、関連した項目の全てに印字されてしまうことです。つまり、以下のような表示が為されます。

Jason Stanley, “Context and Logical Form”, in *Language in Context: Selected Essays* (Oxford: Oxford University Press, 2007), pp. 30–68.

Jason Stanley, “Semantics in Context”, in *Language in Context: Selected Essays* (Oxford: Oxford University Press, 2007), pp. 201–30.

これは、遠くないうちに修正されるでしょう。

現在のところ、節ごとの書誌情報出力はサポートされていません。複数の書誌情報がある場合には、LyXは同じ情報を繰り返し出力してしまいます。

8.3 索引

LyXは、索引をXHTMLとして書き出すことができますが、現在のところ、若干の制約があります。

索引の書き出しは、ユーザーの手引きの索引の節に書かれているような手の込んだ方法を使おうとしないほうが、信頼性が保てます⁹。当該節の小節見出しに沿って、それらがどう処理されるかを説明しましょう。

- 索引項目のグループ化（下層項目）：LyXはこれらをサポートしようと努めていますが、項目は「！」（感嘆符の周りに空白）で分離しなくてはなりません。これは、そうしないと数式などで使われているエスケープされた感嘆符をチェックするのが非常に難しくなるからです。
- ページ範囲：ページ範囲については、XHTMLでは意味がないので、全くサポートされません。その代わりに、引用項目のはじめと終わりの頁それぞれに、2つの索引項目が得られるだけです。

⁹ここでの主な問題は、これらの操作の際に、LyX自身は何もしていないという点です。つまり、生のL^AT_EXで入力しなくてはならないようなものをそのまま入力しているのです。

- 相互参照: 相互参照のサポートはありません. LyXが「|see...」という下だりを持つ項目を検出すると, その部分はただ無視されて, 残りの部分が通常の索引項目だとみなされます.
- 索引項目の整序: LyXは整序順の変更をサポートします. LyXは, 最初に検出した「@」の前の部分を取り出して, それを整序に用い, 最初の「@」の後の部分を実際の項目として取り出します. 現在のところ, LyXはエスケープされた「@」を取り扱わないので, 電子メールアドレスを索引に使わないでください.
- 索引項目のレイアウト: 項目は, テキスト様式ダイアログを使うなり, LyX内で使用可能な方法を用いることで, 索引項目を成形することができます. 「項目|textbf」のような文法はサポートされません. LyXは, 項目内にパイプ記号「|」を検出すると, パイプ記号とその後に続くものすべてを削除してしまいます.
- 複数の索引: 複数の索引はサポートされていません. すべての項目は, ひとつの大きな索引として印字されます. 索引が何度も印字されることを防ぐため, 中心となる索引だけを印字しますので, その索引は必ず置くようにしてください.

8.4 記号一覧および用語集

現在のところ, 用語集のサポートはありません. この機能の追加は, 比較的容易だと思われるので, あなたの貢献をお待ちしております.

9 DocBook 出力

L^AT_EX は、HTML 以外に DocBook XML で書かれた文書を生成することができます。第 2.3 版以前では、DocBook 文書 (第 4 版のみ) を生成するのに限られたひな型しか使えませんでした。L^AT_EX 第 2.4 版からは、ほとんどの L^AT_EX レイアウトが、有効な DocBook 文書 (第 5 版のみ) を生成することができます。この機能は L^AT_EX に組み込まれ、外部ツールを使用する必要はありません。

結果として、DocBook 出力は、レイアウトファイルとモジュールを通じて、L^AT_EX 出力とまったく同じように調整し拡張することができます。詳細については、取扱説明書『カスタマイズ篇』第 5 章を参照してください。

数式出力のような、ほとんどの L^AT_EX の機能は DocBook でサポートされています。

- 数式出力は、ほとんど MathML で処理されます (主要な例外には L^AT_EX でサポートされない構成物、すなわち ERT があります)。すべての数式は raw T_EX として DocBook 出力に渡し、さらなる処理に回すこともできます。
- 書誌情報と引用は、外部 BibT_EX ファイルの使用を含めて実装されています。文献項目は、詳細な情報がわかるとき表示はされませんが、標準的な DocBook 構成物として出力はされます。項目の表示は、外部 DocBook プロセッサが行うものと想定されています。
- 索引は、複数索引を含めてサポートされています。グループ化 (!使用)・ページ範囲・相互参照 (|see) は、エスケープすることも含めてサポートされています。項目順序 (@) やレイアウト (|mathbf) はサポートされていません。索引項目のエスケープはサポートされていません。
- 用語集は実装されています。

以下の 2 つの主要パラメーターを文書レベルで設定することができます。

- 表の形式：L^AT_EX は、ユーザーの要請に応じて、表を HTML (既定値) もしくは CALS として生成することができます。ほとんどの DocBook プロセッサは、両方の形式を受け付けます。

- MathML プレフィックス：DocBook では、HTML と違って、MathML が自身のネームスペース内に包摂されています。すなわち、MathML 標準に従うタグを示さなくてはならないということです。以下の 3 つの選択肢が提供されています。
 - インライン：MathML ネームスペースは（各数式の xmlns 属性を使用して）各数式用に定義されます
 - m プレフィックス使用（既定値）：MathML ネームスペースは（ルート要素で xmlns:m 属性を使用して）文書レベルで定義されます。各 MathML は m:math のように m プレフィックスを使用します。
 - mml プレフィックス使用：m オプションに類似ですが mml を使用しますこのオプションは、L^AT_EX を個人的なスタイルに合わせたい場合や、MathML 用の一つのプレフィックスしか受け付けないバグ付きソフトウェアを使用する際に、とくに役に立ちます。

10 LyXサーバー

10.1 はじめに

「LyXサーバー」を使用すると、他のプログラムが、LyXと通信したり、LyXコマンドを実行させたり、LyXの内部状態に関する情報を取得したりすることができるようになります。これは熟練したユーザー向けの機能ですが、便利なはずです。たとえば、LyXサーバーに書き込むことによって、JabRefのような書誌情報管理ソフトが、LyXに文献引用を「押し込む」ことが可能となります。

10.2 LyXサーバーを開始するには

LyXサーバーは、名前をつけたパイプの対を利用することで機能します。パイプは、通常UserDirの下に作られ（Windowsを除く。Windowsでは、名前を付けたローカルのパイプは`\\.pipe`に特別なオブジェクトとして置かれます）、「`.lyxpipe.in`」および「`.lyxpipe.out`」という名前がついています。外部プログラムは、`.lyxpipe.in`に書き込みを行い、戻りデータを`.lyxpipe.out`から読み出します。パイプ名の語幹は、ツール▷設定から、たとえば「`/home/myhome/lyxpipe`」あるいはWindowsでは「`\\.pipe\lyxpipe`」などに設定することができます（ここで、`lyxpipe`以外の任意のファイルパス、例えば「`\\.pipe\my\lyx\pipe`」なども使用できます）。サーバーを開始するためには、これを手動で設定しなくてはなりません。

LyXは、パイプを生成するに際して、「`.in`」および「`.out`」を付け加えます。パイプがすでに存在していれば、LyXは他のLyXプロセスがすでに実行中であると判断して、サーバーを開始しません。POSIX（Unix型）システムでは、なんらかの理由で、LyXが終了する際に使用されていない「古い」パイプが削除されずに残ったりすると、LyXはこれを削除しようとします。何らかの理由でこれが失敗した場合、手動でパイプを削除したのちに、LyXを再起動する必要があります。Windowsでは、プログラムが終了したりクラッシュしたりすると、OSがパイプを削除するので、「古い」パイプが残ることはありません。

サーバーを使用するLyXプロセスを複数起動するには、それぞれのプロセスに、別

のユーザーディレクトリを使用させたり、別々の preferences ファイルを使用させるなど、別々の設定を使用する必要があります。

クライアントプログラムを開発しようとしているのであれば、LyX サーバーからのデバッグ情報を有効にすると便利かもしれません。そのためには、LyX を `lyx -dbg lyxserver` として起動してください。

C++ 言語で書かれた完全なサンプルクライアントが、ソース頒布版の `development/lyxserver/server_monitor.cpp` にあります。

もう一つの便利なツールとして、`src/client/lyxclient` にあるコマンドラインベースのクライアントがあります。

10.3 通常の通信

LyX コールを発行するには、クライアントは ASCII テキスト行を入力パイプに書き込みまなくてはなりません。この行は、以下の書式を持ちます。

LYXCMD: クライアント名: 関数: 引数

クライアント名 は、クライアントが自由に選べる名前です。これが用いられるのは、クライアントが他の要求クライアントから結果を見分けることができるように、LyX が返答をする際、クライアント名をエコーするときだけです。

関数 は、LyX に実行させたい関数です。これは、ミニバッファで使用するコマンドと同じです。

引数 は、一部の関数のみに効力を持つオプションの引数です（たとえば、「自己挿入」LFUN は、現在のカーソル位置に引数を文章として挿入します）。

LyX からの返答は、出力パイプに送り出され、

INFO: クライアント名: 関数: データ

の書式をしています。ここで**クライアント名**と**関数**は、コマンド要求からそのままエコーされますが、**データ**はコマンド実行の結果に依存した、多少なりとも有用な情報です。「font-state」のような一部のコマンドは、LyX の内部状態に関する情報を返しますが、空のデータ応答を返すものもあります。空のデータ返答は、コマンド実行が無事終わったことを意味します。

エラーが発生した場合の LyX の返答は

ERROR: クライアント名: 関数: エラーメッセージ

の書式をしており、**エラーメッセージ**には、なぜコマンドが失敗したかの説明が入っているはずです。

(例)

```
echo "LYXCMD:test:beginning-of-buffer:" >~/lyxpipe.in
echo "LYXCMD:test:get-xy:" >~/lyxpipe.in
read a <~/lyxpipe.out
echo $a
```

10.3.1 AppleScript (Mac OS X)

LyX 2.1 以降、LyX は、コマンド実行を通じた通常のコミュニケーションのために、AppleScript を使った基本的なやりとりをサポートしています。このコマンドは、直接引数 (実行する **function**) とオプション引数を取ります。これは、関数の出力を返すか、エラーをトリガーして、エラーメッセージとコードを表示します。

以下は例です。

```
tell application "LyX"
  try
    -- Stores the current file name into f
    set f to (run "server-get-filename" with argument "")
    on error the error_message number the error_number
      display dialog "Error: " & the error_number & ". " -
        & the error_message buttons {"OK"} default button 1
    end try
  end tell
```

10.4 通知

LyX は、発生しているイベントを、非同期的にクライアントに通知することができます。いまのところ、ユーザーがあるキー列を関数「notify」に割り当てているときのみ、LyX はこの通知を行います。LyX が送る文字列の書式は

NOTIFY: キー列

となっていて、**キー列**は、ユーザーが実際に打鍵したキー列の印字表現になります。

この機構は、LyX のコマンド群を拡張してマクロを実装するのに使用することができます。なんらかのキー列を「notify」に割り当ててください。それから出力パイ

プで待機するクライアントを起動し、キー列に応じたコマンドを読み込み、一つないし複数の L_YX コマンドを発行する L_YX 要求と L_YX コールを使用する関数を開始するのです。

10.5 単純な L_YX サーバープロトコル

L_YX は、セッション管理に使用することができる単純なプロトコルを実装しています。メッセージは、すべて

LYXSRV: クライアント名: プロトコルメッセージ

の書式をしており、プロトコルメッセージは「hello」ないし「bye」です。クライアントから「hello」を受け取ると、L_YX はメッセージを聴いていることをクライアントに伝えるために返答をし、L_YX が終了することをクライアントに伝えるのには、L_YX が「bye」を送信します。

10.6 DVI と PDF の原文検索

DVI/PDF ビューア¹には、原文検索 (reverse search ないし inverse search と呼ばれます) 機能を提供しているものがあります。これを使えば、DVI/PDF 出力の特定の場所をクリックすることによって、L_YX の文書中で対応する行にカーソルを移動させることができます。これを実現するためには、ビューアは L_YX と情報交換を行う必要があります。これは、命名パイプ (*lyxpipe*) を使用するか、L_YX が作業ディレクトリに生成する UNIX ドメインソケット (*lyxsocket*) を使用する (これは *lyxclient* プログラムが L_YX と通信するのに使用している方法です) ことで、L_YX サーバを経由して実現することができます。場合によっては、ビューアと L_YX を橋渡しする補助スクリプトを必要とすることがあるかもしれませんが、それ以外は、ビューアは L_YX と直接通信することができます。これは、選んだビューアと基本ソフトに依存します。また、必要とされるビューアの設定法と原文検索が実際に行われる方法とも依存します。したがって、下記では個々のビューア毎の原文検索の設定の仕方を説明することとします。その前に、DVI/PDF 出力で原文検索を有効にするためには、一般に何が必要とされるかを説明します。

¹PDF の原文検索機能を提供しているビューアには、KDE/Linux 上の Okular と Qpdfview on Unix と Mac OS X 上の Skim、Windows 上の SumatraPDF があります。

10.6.1 自動有効化

ツールバーでボタンを押せば、ほとんどの場合、LyXがすべてをうまくこなしてくれます。あるいは、文書▷設定▷形式メニューで出力と同期にチェックを入れることで有効にすることもできます。その場合、LyXは、自動的に、必要な SyncTeX マクロ (PDF の場合) や、srcltx パッケージ (DVI の場合) を挿入してくれます。このオプションは、表示/更新ツールバーからも容易にアクセスすることができます。

もし、これとは別の設定が必要な場合には、文書▷設定▷出力▷出力と同期の直下にあるドロップダウンボックスから、自製マクロを選択ないし入力することができます。

方式として `\synctex=1` を用いると、gzip 圧縮が有効となることに注意してください。使用しているビューアが gzip 圧縮をサポートしていない場合には、代わりに `\synctex=-1` を指定してください。

また、srcltx パッケージや src-specials を取り込むと、組版上、不具合が生じることがあるのでご注意ください。したがって、srcltx パッケージや src-specials を使用している場合には、最終版を組版する際には、出力同期を無効にするべきです。

自動設定でカバーされている範囲よりも、さらに特殊なものが必要な場合には、出力同期を手動で設定する仕方を述べた次節をお読みください。自動設定で事足りる場合には、自動設定・手動設定ともに必要となるビューア上の設定を述べた、第 10.6.3 節に進むことができます。

10.6.2 手動有効化

L^AT_EX は、原文検索のために複数の方法を提供しています。ある方法は latex/pdflatex プログラムに組み込まれており、他の方法は外部パッケージとして提供されています。どちらを選択するかは、お使いの L^AT_EX 頒布版が既に特定の方法を提供しているか (組み込みによる方法は比較的新しいものです) と、お使いのビューアがそれに対応しているかに依存します。使用可能な方法を以下に述べます。

src-special 経由の DVI 組み込み検索 (DVI のみ)

この方法は、DVI ファイルに原文検索に必要な情報を追加するものです。これは、かなり前の L^AT_EX から利用可能となっており (比較的最近の L^AT_EX 頒布版はすべて対応しています)、安定して動作します。これを有効にするには、設定▷ファイル処理▷変換子の LaTeX (plain)->DVI ないし LaTeX (plain)->DraftDVI 変換子を latex-src-specials \$\$\$i に変更します。もしこれが動作しないときは、お使いの L^AT_EX

エンジンが違うオプションを必要としていないか確認してください（頒布版によっては文法の異なることがあります）。

外部パッケージ（PDFSync および srcltx）

pdfsync および *srcltx* パッケージは、PDF 出力（*pdflatex* 経由）および DVI 出力での原文検索を提供します。これを有効にするには、LyX プリアンブルで以下のようにパッケージを読み込んでください。

- PDF の原文検索には `\usepackage{pdfsync}`
- DVI の原文検索には `\usepackage[active]{srcltx}`

と書きます。DVI と PDF の両方の原文検索を実行できるようにするためには、プリアンブルに以下の行を挿入する方法があります。

```
\usepackage{ifpdf}
\ifpdf
  \usepackage{pdfsync}
\else
  \usepackage[active]{srcltx}
\fi
```

このようにすれば、DVI でも PDF（*pdflatex*）でも正しいパッケージを使いながら、ファイルを閲覧することができます。

なお、PDFSync は、文書の出力レイアウトに干渉する可能性があるので注意してください。したがって、仕上がり文書では PDFSync を無効にすることをお勧めします。

SyncTeX を使った組み込み原文検索（DVI および PDF）

(pdf)latex の最近のバージョンは、PDF および DVI 両方の原文検索を組み込みでサポートしています。このいわゆる *SyncTeX* 機能は、主に、PDFSync パッケージを *pdftex* プログラムへ統合したことと、それを *src-specials* アプローチへ取り込んだことの成果です。これを使用するためには、少なくとも TeXLive 2008 か最近の MikTeX 頒布版が必要です。また、既に SyncTeX サポートを提供している PDF ビューアは少ししかありません（UNIX 上の Okular と Qpdfview, Mac 上の Skim, 及び Windows 上の SumatraPDF など。詳細は次節を参照のこと）。

DVI 出力で SyncTeX を有効にするには、設定▷ファイル処理▷変換子の L^A-T_EX (PLAIN) -> DVI ないし L^AT_EX (PLAIN) -> DRAFTDVI 変換子を latex

`-synctex=1 $$i` に変更します。PDF 出力用には、`LATEX` (`PDFLATEX`) -> PDF (`PDFLATEX`) 変換子を `pdflatex -synctex=1 $$i` に変更してください。お使いのビューアが下記に記述されていない場合や、万が一うまくいかない場合にはまた、ビューアを `SyncTEX` で使うのに設定を行う必要があるかどうか、お使いのビューアの取扱説明書をご確認ください。²

10.6.3 特定のビューアを設定して使うには

Xdvi (全プラットフォーム)

LyX は、`lyxclient` プログラムを自動的に使うのに必要なフックを、既に提供しているので、`xdvi` を使用する場合には、DVI 原文検索を実行するのに他にすべきことはありません。使用中の文書を上記で述べたように設定するだけです（原文検索は、Ctrl-クリックか、Mac OS X 上では Alt-クリックで始動します）。

しかし、何らかの理由で、LyX と通信するのにソケットではなく命名パイプを使用したい場合には、設定▷ファイル処理▷ファイル形式の DVI ビューアを `xdvi -editor "lyxeditor.sh %f %l"` (`lyxeditor.sh` は適切なスクリプト名) に変更してください³。たとえば、最低限のシェルスクリプトは

```
#!/bin/sh
LYXPIPE="/path/to/lyxpipe"
COMMAND="LYXCMD:revdvi:server-goto-file-row:$1 $2"
echo "$COMMAND" > "${LYXPIPE}.in || exit
read TMP < "${LYXPIPE}.out || exit
```

のようになります。ここで `/path/to/lyxpipe` は、設定▷パスで指定した LyXServer パイプのパスを指定してください。⁴

MacDviX (Mac OS X)

`/Applications/MacDviX_Folder/calleditor.script` の終わりに

```
/Applications/LyX.app/Contents/MacOS/lyxeditor "$2" $1
exit 1
```

²`-synctex=1` オプションは、`gzip` 圧縮を有効にします。お使いのビューアが `gzip` 圧縮をサポートしていない場合には、代わりに `-synctex=-1` を使用してください。

³Mac OS X 上では、`DISPLAY=:0.0 xdvi -editor "lyxeditor.sh %f %l"` としなくてはなりません。

⁴ソース頒布版の `development/tools` フォルダには、ユーザが設定した `lyxpipe` を探し出す `lyxeditor` スクリプトがあります。

10 LyX サーバー

という行を付け加えてください。Applications フォルダ以外の場所に LyX を導入した場合には、それに従って上記の行を修正してください。

原文検索は、Alt-クリック (OPTION-クリック) で始動します。

Skim (Mac OS X)

設定▷ファイル処理▷ファイル書式▷PDF (pdflatex) のビューア設定に `open -a Skim.app $$i` と入力し、Skim▷設定▷Sync で LyX を選択してください。

原文検索は、COMMAND-SHIFT-クリックで始動します。

Evince (GNOME)

Evince では原文検索はそのままでは動きませんが、サードパーティのヘルパースクリプトを使用すれば、利用可能になります。詳細については、<https://wiki.lyx.org/LyX/SyncTeX> を参照してください。

Okular (KDE)

設定▷Okular を設定▷エディタで「Lyx クライアント」をエディタとして選択してください。これによって適切なコマンド (`lyxclient -g %f %l`) が挿入されます。

原文検索は、SHIFT-クリックで始動します。これは、Okular が「ブラウズ」モード (すなわち手のマークをクリックしたとき) にあるときのみ動作することに注意してください。

Qpdfview (UNIX)

Qpdfview は、第 0.3.5 版以降、SyncTeX をサポートしています。Edit▷Settings▷Behavior を開いて「Source editor」項目の入力フィールドをクリックし、`lyxclient -g %1 %2` というコマンドを入力してください。

原文検索は、ダブルクリックで始動します。

YAP (Windows)

yap を起動し、View▷Options メニューを選択し、「Inverse DVI Search」タブを選択します。「New...」ボタンをクリックして開くウィンドウの「Name:」フィールドに「LyX Editor」(または自分の好きな名称) と入力してください。「...」というラベルのボタンをクリックしてファイルダイアログを開き、バッチファイル `lyxeditor.bat` (以下参照) のあるディレクトリを見つけます。`lyxeditor.bat` を選択してプログラ

ムの引数を%f %1と指定してください。lyxeditor.batは、lyxpipeを通じてLyXと通信するためのラップとして使用され、以下のようにになっています。

```
@echo off
echo LYXCMD:revdvi:server-goto-file-row:%~1 %2> \\.\pipe\lyxpipe.in
type \\.\pipe\lyxpipe.out
```

LyXで指定したLyXServerパイプパスが、\\.\pipe\lyxpipeとなっていることを確認するか、lyxeditor.batラップの方を適切に修正してください。

yapでは、原文検索はダブルクリックで始動します。

SumatraPDF (Windows)

SumatraPDFを原文検索で使用するためには、設定▷ファイル処理▷ファイル書式▷PDF (pdflatex)のビューア設定でSumatraPDF -inverse-search "lyxeditor.bat "%f" %1" (lyxeditor.batは上記のラップ)と入力してください。SumatraPDF.exeがコマンドPATHにない場合、フルファイル名を使用してください。

原文検索はダブルクリックで始動します。

YAP (Cygwin)

まずはじめに、yapがWindows環境での既定のDVIビューアになっていることを確認してから、yapを起動し、View▷Optionsメニューを選択し、「Inverse DVI Search」タブを選択します。「New...」ボタンをクリックして開くウィンドウの「Name:」フィールドに「LyX Editor」(または自分の好きな名称)と入力してください。「...」というラベルのボタンをクリックしてファイルダイアログを開き、バッチファイルlyxeditor.exeプログラム(これは、Cygwinでは既定でLyX実行ファイルとともに導入されます)のあるディレクトリを見つけます。lyxeditor.exeを選択してプログラムの引数を-g %f %1と指定してください。これによって、LyXとの通信にlyxsocketが使用されるようになります。何らかの理由でlyxpipeを使用したい場合には、-gオプションを削除して、LyX設定でLyXServerパイプパスを指定することを忘れないでください。

yapでは、原文検索はダブルクリックで始動します。

SumatraPDF (Cygwin)

SumatraPDFを原文検索で使用するためには、設定▷ファイル処理▷ファイル書式▷PDF (pdflatex)のビューア設定でSumatraPDF.shと入力してください。ここでSumatraPDF.shは以下のようなスクリプトです(お使いのコマンドパスに置きます。/usr/local/binがもっとも良い選択肢でしょう)。

```
#!/bin/bash
cd $(dirname $1)
SumatraPDF -inverse-search "lyxeditor -g %f %1" $(basename $1)
```

SumatraPDF は、Windows ネイティブのアプリケーションであり、LyX の Cygwin 版が使用する POSIX パスを理解しないので、このラップスクリプトが必要となります。SumatraPDF.exe がお使いのコマンドパスにない場合は、上記のスクリプトで POSIX のフルパスを使用してください。-g は *lyxsocket* 経由の通信を有効にします。ここでも、*lyxpipe* を使用したい場合には、-g オプションを削除して、LyX 設定で LyXServer パイプパスを指定することを忘れないでください。

原文検索はダブルクリックで始動します。

10.7 出力先検索

出力先検索 (forward search) とは、英文名が示すとおり、いわば原文検索 (reverse search) の「逆」を意味します。すなわち、LyX 内部から、ビューアに特定の箇所を表示させることを可能にします。(以下に説明するように) もし出力先検索の準備ができていれば、カーソルを LyX 文書の好きなところに置いて、移動▷出力先検索を選択すると (あるいはマウスの右クリックで表示される文脈依存メニューから出力先検索を選択してもよい)、ビューアも同じ箇所に移動します。もちろん、使用しているビューアが、この機能をサポートしていません。

出力先検索を有効にするためには、生成された PDF/DVI 出力ファイルに、TeX ソースに関する情報を追加する必要があります。これは、第 10.6.1 節に述べられている方法で実行することができます。

次に、ツール▷設定▷出力▷一般で自分の使うビューアを利用するように、LyX を設定する必要があります。いくつかのビューアについては、試験済みの設定が前もって準備されており、ドロップダウンリストから選択することができます。これらの設定の何れも意にそぐわない場合には、ご自身で適切な設定を見つけ出して入力しなくてはなりません。定義文法は、以下の変数を使用します。

- `$$n`: 行番号
- `$$t`: 書き出された (一時) .tex ファイル名 (パスを除く)
- `$$f`: 書き出された (一時) .tex ファイル名 (パスを含む)
- `$$o`: 書き出された出力ファイル名 (どちらが一時ディレクトリに存在するかに依存して、dvi あるいは pdf)。

yap・xdvi・okular⁵・qpdfview・Sumatra PDF⁶など、一部のビューアでしか、完全な出力先検索機能は提供されていないことに注意してください⁷。Evince⁸など他のビューアには、出力先検索を使用するのに、追加のツールを必要とするものもあります。広く使われているPDFビューアの多く（特にAdobe Reader）は、出力先検索を全く提供していませんが、他の一部のビューア（たとえばxpdf）では、少なくとも、文書を読み込み直したり、ファイルの特定ページに移動することはできますので、少なくとも「近く」には移動することができます。この後者の機能は、syncTeXの外部コールで提供されています（事前に用意されている用例をご覧ください）。

出力先検索は、DVI出力とPDF出力の双方で動作します。LyXは、過去にあなたが使用したプレビュー形式をチェックして（つまり、一時ディレクトリにどちらの形式が存在するかをチェックして）、その形式に適切な設定を選択します。

10.8 T_EXのチェック

ASGER ALSTRUP 文

10.8.1 はじめに

chktexプログラムが導入されていると⁹、ツールメニューの下に、T_EX構文チェックという項目が表示されます。chktexは、CTAN <https://www.ctan.org/tex-archive/help/Catalogue/entries/chktex.html> から入手することができます。

ChkT_EXパッケージは、L^AT_EXの構成物には直感的でないものもあり、すぐ忘れてしまいがちであるのに不満を感じたJENS T. BERGER THIELEMANNが書いたプログラムです。このプログラムは、L^AT_EXファイルを走査して、ファイルの整合性をチェックし、犯しがちな誤りを見つけ出します。別の技術用語では、LaTeXにおけるlintであるとも言えます。

さて、LyXはつねに正しいL^AT_EXファイルを生成するはずなのに、文法チェッカーはLyXで何をするというのでしょうか。答えは簡単です。lintがC言語プログラム

⁵ ツール▷設定▷ファイル処理▷ファイル形式でokular --uniqueと設定すると良いでしょう。

⁶ SumatraPDFは、外部プログラムCMCDDE (<http://www.istri.fr/spip/zip/CMCDDE.zip> からダウンロード可能)を通じてDDEコマンドを使用することもできます。

⁷ SumatraPDFも、外部プログラムCMCDDEを通じて、DDEコマンドを使用することができます。CMCDDEは、<http://www.istri.fr/spip/zip/CMCDDE.zip> からダウンロードすることができます。

⁸ Evinceでの出力先検索は、サードパーティのヘルパースクリプトを使用すれば、利用可能になります。詳細については、<https://wiki.lyx.org/LyX/SyncTeX> を参照してください。

⁹ L^AT_EX頒布版としてMiK_T_EXを使用している場合には、chktexはまだ利用可能ではありません。

の文法をチェックするだけでなく、打ち間違いによる意味上のチェックも行うように、ChkTeX は、文法エラーの他に犯しがちな写植上の誤りも検出してくれるからです。ChkTeX は、以下のような誤りを検出することができます。

- 省略記号の検出：
...ではなく...を使わなくてはなりません
- 括弧の前後にスペースを置かない：
(wrong spacing)：誤った空白
- よく使われる省略形の後ろは通常スペースにする：
e.g. is too wide spacing.：広すぎる空白
- 直前の文章が大文字で終わっているときに、文末スペースに直す：
This is a TEST. And this is wrong spacing.：誤った空白
- ラベルや類似コマンドの前のスペース：
ラベルが別のページに行ってしまうように、ラベルは文章に接していません。 ¹⁰ このラベルは離れすぎです。
- 参照の前にハードスペースではなくスペース：
運が悪ければ、参照文字列と参照番号のあいだでちょうど改行されてしまい、残念な結果となります。第 10.8.1 節をご覧ください。
- 数字のあいだで ×ではなく「x」が使われている：
2x2 は、2 × 2 よりも安っぽく見えます。

などなど... ChkTeX は、印刷する前に文書を「仕上げる」上で、たいへん価値のあるツールです。必須のスペルチェックを終えた直後、組版の微調整をする前には、これを実行するべきです。

10.8.2 使いかた

このプログラムを導入したならば、使うのはツール▷TeX 構文チェックを選択するだけなので簡単です。こうすると、LyX は使用中の文書の L^AT_EX ファイルを生成して、ChkTeX を起動し、当該ファイルを検査しますが、ChkTeX から警告があった場合には、それを含んだ「エラーボックス」を LyX 中に挿入します。警告は、誤りのある箇所のそばに置かれるので、メニュー項目の移動▷エラーを選択するか、既定の cua

¹⁰この脚注は、正しくないページに行ってしまう恐れがあります。

キー設定ファイルならばショートカットキー C-g を押せば、簡単に警告の箇所を見出すことができます。エラーボックスを開けるには、マウスでボックスをクリックするか、cua キー設定の場合にはショートカットキー C-i, emacs キー設定の場合はショートカットキー C-o を押します。警告を読み、それが誤りを示していれば修正します。警告が何を意味しているか理解しにくい場合には、無視しても問題はありません。画面上の文書と、ChkT_EX を起動する際に行なわれる実際の技術的な操作のあいだには、隠された層があり、この隔たりのせいで、一部の警告が不可解なものになったり、明らかに馬鹿げたものになる場合があります。

本文書自体、この機能を確認する良い材料になりますので、実行してみてください。すると、かなりの量の警告が出るはずです。これはコンピュータが敏感すぎるせいですので、警告のほとんどは実際には警告に値しません。

10.8.3 微調整のしかた

場合によっては、ChkT_EX は、あなたが望むよりも煩すぎると思うかもしれません。そのような場合には、必要と思うまで ChkT_EX を無効にするか、あなたにもう少し合うように ChkT_EX を調整することができます。

ChkT_EX は調整や拡張がひじょうに柔軟なのですが、L_AT_EX における ChkT_EX 関連の問題が、それですべて解決すると思うべきではありません。L_AT_EX は、ChkT_EX 出力¹¹ から得た行番号と内部構造を一致させるため、やや特殊な L_AT_EX ファイルを生成しなくてはならないので、警告の一部は正しく表示されていないように見えるでしょう。これに対処するためには、以下の2つの方法があります。

- ツール▷設定▷出力▷L_AT_EX▷CheckT_EX コマンド中の ChkT_EX 起動コマンドラインを調整するか、グローバルな ChkT_EX の設定ファイル（通常は chktexrc ファイル）を調整します。コマンドラインからどの警告を有効・無効にできるかは以下をご覧ください。
- ファイル▷書き出し▷L_AT_EX を使って文書を L_AT_EX ファイルに変換し、そのファイルに対して chktex を手動で実行します。このようにすると、L_AT_EX 文書中の対応箇所を見つけるのがやや面倒ですが、少し我慢すれば見つけることができます。

以下は、設定で有効化・無効化できる警告メッセージです。警告を無効にするには -n#, 有効にするには -w# を用います。イタリック表示の項目は、既定で無効となっています。既定値は "chktex -n1 -n3 -n6 -n9 -n22 -n25 -n30 -n38" です。

¹¹chktex の実際の出力は、chktex を実行した直後に **文書▷L_AT_EX ログ** を使えば見ることができます。

L_AT_EX は、`chktex` と通信するために、他のコマンドラインパラメータを特定の方法で使用しているため、メッセージを有効化・無効化する以外のオプションを使うべきではありません。

1. *Command terminated with space.*
2. Non-breaking space (“~”) should have been used.
3. *You should enclose the previous parenthesis with “{ }”.*
4. Italic correction (“\”) found in non-italic buffer.
5. Italic correction (“\”) found more than once.
6. *No italic correction (“\”) found.*
7. Accent command “`cmd`” needs use of “`cmd`”.
8. Wrong length of dash may have been used.
9. “`%s`” *expected, found “%s”.*
10. Solo “`%s`” found.
11. You should use “`%s`” to achieve an ellipsis.
12. Inter-word spacing (“\ ”) should perhaps be used.
13. Inter-sentence spacing (“\@”) should perhaps be used.
14. Could not find argument for command.
15. No match found for “`%s`”.
16. Math mode still on at end of L_AT_EX file.
17. Number of “`char`” doesn’t match the number of “`char`”.
18. You should use either “ ” or “ ” as an alternative to “ ”.
19. You should use “ ’ ” (ASCII 39) instead of “ ” (ASCII 180).
20. User-specified pattern found.
21. This command might not be intended.
22. *Comment displayed.*

23. Either "\,' or '\,' will look better.
24. Delete this space to maintain correct page references.
25. *You might wish to put this between a pair of “{}”.*
26. You ought to remove spaces in front of punctuation.
27. Could not execute L^AT_EX command.
28. Don't use \/ in front of small punctuation.
29. $\$ \times \$$ may look prettier here.
30. *Multiple spaces detected in output.*
31. This text may be ignored.
32. Use " to begin quotation, not '.
33. Use ' to end quotation, not ".
34. Don't mix quotes.
35. You should perhaps use “cmd” instead.
36. You should put a space in front of/after parenthesis.
37. You should avoid spaces in front of/after parenthesis.
38. *You should not use punctuation in front of/after quotes.*
39. Double space found.
40. You should put punctuation outside inner/inside display math mode.
41. You ought to not use primitive T_EX in L^AT_EX code.
42. You should remove spaces in front of “%s”
43. “%s” is normally not followed by “%c”.

L_YX の将来の版では、このツール（およびその縮小版 lacheck）の力を最大限引き出すように、より完備したインターフェイスを提供できればと考えています。しかし、現状でも十分に役立つものとなっています。ある程度長さがある既存文書に使用してみただけであれば、きっと驚かれることでしょう。

10.9 LyXでのバージョン管理

LARS BULLIK BJØNNES および PAVEL SANDA 文

10.9.1 はじめに

LyX は、今ではもっとも基本的な RCS/CVS/SVN/GIT コマンドをサポートしています。もしもっと洗練されたことを行いたい場合には、ターミナル上もしくは好みのクライアントから手動で行う必要があります。

また、CVS のサポートは、subversion のサポートほど良くないので、CVS よりも SVN を使うことをお勧めします。Subversion を学び始めるには、SVN Book¹² が良いでしょう。RCS の場合は、「rsintro」をお読みください（これは man ファイルです。man rcsintro で読めます）。このファイルは、RCS の基本機能のすべてを説明しています。特に、RCS ディレクトリに関するコメントと、マスター RCS ファイル（、v で終わるファイルです）の概念に注意を払ってください。

LyX で実際にバージョン管理機能を使い始める前に、RCS/CVS/SVN/GIT の使用方法に習熟しておかなくてはなりません。LyX で LyX への実装においては、GNU RCS または CVS/SVN パッケージの最新版を仮定しています。旧版への保証はありません。ログメッセージのほとんどは、現在のところ操作後には表示されません。確認したい場合には、メッセージ面で確かめてください。バージョン管理を常用するユーザーには、VC ツールバーが便利でしょう。これは表示▷ツールバー▷バージョン管理で有効にできます。

自製の外部コマンドを導入したい場合には、取扱説明書『LyX 関数』の vc-command を確認してください。

バージョン管理機能を使用している場合には、文書を非圧縮形式で保存をすることを強く推奨します（文書▷圧縮にチェックが入っていれば外してください）。非圧縮 LyX 文書はテキストファイルであるため、バージョン管理システムが、二つのバージョンをマージすることができます。一方、圧縮 LyX ファイルはバイナリファイルであるため、バージョン管理システムはマージすることができません。また、不必要なマージ衝突を避けるために、複数のユーザーが同一文書の作業を行う場合には、文書▷設定▷出力▷一時特性を保存を無効にすることをお勧めします。

¹²<http://svnbook.red-bean.com/>

10.9.2 LyXにおけるRCSコマンド

以下の節ではLyXでサポートされているRCSコマンドを説明します。それらのコマンドはファイル▷バージョン管理のサブメニューにあります。LyXは、RCS 5.7/5.8/5.10に対して試験されています。

10.9.2.1 登録

文書がバージョン管理の下にない場合には、これがメニューにあらわれる唯一の項目です。文書がバージョン管理下であれば、項目登録は不可視になります。

このコマンドは、文書をRCSに登録します（CVSが管理しているディレクトリ下でない場合）。すると対話的に、文書の最初の説明を入力するように尋ねられます。すると文書は読み込み専用モードになるので、変更を加えるためにはチェックアウトして編集を実行する必要があります。バージョン管理下にある文書は、ミニバッファのファイル名の後ろに「[RCS:<バージョン> <ロック>]」という項目が表示されます。

実行されるRCSコマンドは：

```
ci -q -u -i -t-"<初期説明文>" <ファイル名>
```

スイッチについて知りたい場合は、`man ci`を読んでください。

10.9.2.2 変更をチェックインする

ファイルの編集が終わると、その変更をチェックインします。このとき、変更についての説明文を求められます。これは履歴ログに記録されます。バージョン番号は繰り上がり、変更がマスターRCSファイルに適用され、文書がアンロックされて読み込み専用モードになります。

- RCSコマンド：`ci -q -u -m"<説明文>" <ファイル名>`

10.9.2.3 チェックアウトして編集

これは、文書をロックして編集できるようにします。そしてあなただけが文書を読み書きできるようにします。通常は、しばらくの間、文書を編集し、それが終わると変更をチェックインします。あなたがファイルをロックしたことを反映して、ステータスラインが変わります。

- RCSコマンド：`co -q -l <ファイル名>`

10.9.2.4 リポジトリのバージョンに戻す

これは、最終チェックイン以降、文書に加えられた変更をすべて破棄します。変更が破棄される前に、警告が発せられます。

- RCS コマンド：`co -f -u<バージョン> <ファイル名>`

10.9.2.5 コピー

これは、現行文書のコピーを生成します。RCS は、コピー操作を自前ではサポートしませんので、バージョン履歴には保存されず、新コピーは新規ファイルとして追加されます。最後のチェックイン以降、変更が加えられていないサラの文書であることが要求されます。ファイル名とコピー操作の説明を入力するように促され、その後、新コピーがローカルとリポジトリの両方に生成されます。コピーされた文書と元の文書の親ディレクトリが異なる場合には、コピーにインクルードされたファイルの相対パスは、(ファイル名前をつけて保存...と同様に) 全て調整されます。最後に、元の文書の代わりに、コピーされたファイルが読み込まれます。

RCS コマンド：

```
"ファイル名"を"新ファイル名"にコピーする場合
ci -q -u -i "<新ファイル名>"
```

10.9.2.6 最後のチェックインを取り消す

これは、最後のチェックインがあたかも為されなかったかのようになります。LyX に読み込まれた文書に変更は加えられませんが、マスター RCS ファイルから最後のバージョンが削除されます。

- RCS コマンド：`rcs -o<バージョン> <ファイル名>`

10.9.2.7 履歴を表示

RCS 文書の完全な履歴を表示します。ブラウザ中に、`rlog <ファイル名>`の出力が表示されます。詳細は `man rlog` をご覧ください。

10.9.2.8 改訂情報

LyX は、RCS バージョン番号情報 (のみ) をサポートしています。詳細については、10.9.4.11 を参照。

10.9.3 LyXにおけるCVSコマンド

LyXは、現在CVSを一部サポートしています。そのコマンドは、ファイルのバージョン管理サブメニューにあります。バージョン管理システムSVNの方が強力なので、可能ならばCVSではなく、SVNを使用してください。

10.9.3.1 CVSの一般的な利用法

CVSを初めて使いたい場合には、外部ツールを使用して自分のレポジトリを作成し、作業コピーをチェックアウトする必要があります。クライアント=サーバーモデルを使用している場合には、最初のレポジトリチェックアウトを行う前にログインする必要がありますがあるかもしれません。

改訂管理下にある文書のレポジトリを他人も利用している場合に、同一文書の同一箇所に別々の変更を加えてしまうと問題が発生します。標準的なCVSレポジトリは、ファイルロック機構を有しません。これは驚くべきことかもしれませんが、コンフリクトは、文書の同じ箇所の特定の内容について、人々が意見を異にしている場合にのみ発生します。したがって、共著者同士がコンスタントにコミュニケーションを交わしていれば、このようなコンフリクトは滅多に発生しません。もしコミュニケーションがなければ、いずれにしても根本的な問題があるわけです。それにもかかわらず、いわゆる「予備付きチェックアウト」を使用したいと思う人々もいます。この場合、最初にチェックアウトした際に、全ファイルの作業コピーは読み込み専用にされ、ユーザーは、作業コピーを書き込み可能にする特別なコマンドを使用した後、編集を開始することになります。変更をチェックインすると、作業コピーは読み込み専用状態に戻ります。LyXで予備付きチェックアウトを利用したい場合には、`.cvsrc` ファイルを編集して、`cvs -r` という行を書き加える必要があります。この方法の利点は、文書のうち、書き込み可能コピーを誰が使用しているかを見ることができることです。ただし、一人のユーザーだけがコピーを書き込み可能にしているとは限りません。

LyXは、あなたが予備付きチェックアウトを使用しているか、予備なしチェックアウトを利用しているか、推測しようと試みます。使用中の作業コピーが読み込み専用になっているか、あるいはそれが書き込み可能であって、CVS/Baseサブディレクトリに使用中文書のコピーが存在している場合には、予備付きチェックアウトを使用しているものと仮定し、そうでない場合には予備なしチェックアウトを使用しているものと仮定します。予備付きチェックアウトが検出された場合には、作業コピーが読み込み専用の場合に書き込み可能にするためには、チェックアウトメニューを使用しなくてはなりません。その後、チェックインメニューが利用可能になるので、自分が加

えた変更をリポジトリに移した後、作業コピーを再び読み込み専用にする際には、これを使用します。

CVS の詳細な情報は、<http://www.nongnu.org/cvs> と <http://ximbiot.com/cvs> にあります。

以下に言及するサブコマンドとスイッチを理解するには、`man cvs` をお読みください。

10.9.3.2 登録

お使いの文書が改訂管理下に置かれていなければ、メニューにはこの項目だけが表示されます。もし改訂管理が有効ならば、登録項目は表示されません。

本コマンドは、既に文書ディレクトリを CVS 管理下に置いている場合にのみ（すなわち `CVS/Entries` ファイルが存在する場合にのみ）、お使いの文書を CVS に登録します。つまり、外部ツールを使用して、アーカイブを自分で作成ないしチェックアウトしなくてはならないことを意味します（このステップを忘れてしまうと、LyX は文書を RCS に登録します）。

すると、文書の最初の説明文を入力するように促されます。登録したファイルは、まだチェックインされたわけではないことに注意してください。

実行される CVS コマンドは、`cvs -q add -m"<入力した説明文>" "<ファイル名>"` です。

上記および他のすべての CVS コマンドにおける "`<ファイル名>`" は、「カレント作業ディレクトリをファイルの場所に移動し、パスコンポーネントを除いたファイル名を引数として使用する」という意味になります。

10.9.3.3 変更のチェックイン

ファイルの編集を終えたら、変更をコミットします。このとき、文書に変更が加えられていれば、変更の説明文を入れるよう促されます。その後、変更がレポジトリに書き込まれます。文書を変更せず、予備付きチェックアウトが検出された場合、チェックアウトで作られた予備は取り消されます。

CVS コマンド：

```
cvs -q commit -m"<入力した説明文>" "<ファイル名>"あるいは
cvs -q unedit "<ファイル名>"
```

10.9.3.4 変更のチェックアウト

レポジトリを他人と共有している場合、他人の加えた変更を自らの作業コピーに取り込まなくてはならないことがあります。

CVS コマンド：`cvs -q update "<ファイル名>"`

読み込み専用チェックアウトが検出された場合、作業コピーを書き込み可能にし予備を作成します。

CVS コマンド：`cvs -q edit "<ファイル名>"`

10.9.3.5 レポジトリ版に戻す

これは、最後のチェックイン以降、文書に加えられた変更をすべて廃棄します。変更が廃棄される前に警告が表示されます。最初にファイルが削除され、次に CVS 更新コマンドが実行されます。

CVS コマンド：`cvs -q update "<ファイル名>"`

予備付きチェックアウトが検出され、作業コピーに変更がない場合、予備のみが取り消されます。

CVS コマンド：`cvs -q unedit "<ファイル名>"`

10.9.3.6 コピー

これは、現行文書のコピーを生成します。CVS は、コピー操作を自前ではサポートしませんので、バージョン履歴には保存されず、新コピーは新規ファイルとして追加されます。最後のチェックイン以降、変更が加えられていないサラの文書であることが要求されます。ファイル名とコピー操作の説明を入力するように促され、その後、新コピーがローカルとレポジトリの両方に生成されます。コピーされた文書と元の文書の親ディレクトリが異なる場合には、コピーにインクルードされたファイルの相対パスは、(ファイル名をつけて保存...と同様に) 全て調整されます。最後に、元の文書の代わりに、コピーされたファイルが読み込まれます。

CVS コマンド：

"<ファイル名>"を"<新ファイル名>"にコピーする場合
`cvs -q add "<新ファイル名>"`

10.9.3.7 ファイル名変更

これは、現行文書のファイル名を変更します。CVS は、ファイル名変更操作を自前ではサポートしませんので、バージョン履歴には保存されず、名称変更後のファイル

は、新規ファイルとして追加されます。最後のチェックイン以降、変更が加えられていないサラの文書であることが要求されます。ファイル名とコピー操作の説明を入力するように促され、その後、ローカルとリポジトリの両方でファイル名が変更されます。新旧ファイル名の文書の親ディレクトリが異なる場合には、コピーにインクルードされたファイルの相対パスは、(ファイル名をつけて保存...と同様に) 全て調整されます。最後に、新ファイル名を使ってファイルが再度読み込まれます。

CVS コマンド：

```
"<ファイル名>"を"<新ファイル名>"に変更する場合
cvs -q add "<新ファイル名>"
cvs -q remove "<ファイル名>"
```

10.9.3.8 ローカルディレクトリチェックアウトのレポジトリからの更新

子文書や図を含んだり、外部.tex ファイルを取り込むなど、文書が複雑になるにつれ、バージョン管理の利用は複雑になります。LyXは、文書が存在するツリー全体の更新をサポートします。これは、CVSの使用法について詳しい知識のない人々や、レポジトリに追加素材をコミットする野望のない人々と共同作業をするような場合に、特に便利です。ファイル構造は、すべての外部ファイルが、文書と同じディレクトリかそのサブディレクトリに来るように組織する必要があります。いずれにしても、複合文書を新しいディレクトリを建てて収容するのは、望ましい習慣です。

コマンドレポジトリを使ってローカル辞書を更新は、ディレクトリ全体を更新します。ローカルの変更が検出された場合には、更新を開始する前にユーザーに警告が発せられます。マージコンフリクトが発生した場合には、コンフリクトを起こしている部分の新旧バージョンが、両方とも最終文書に置かれますので、マージの結果を見直し、修正する必要があります。コンフリクトは、<<<<<<と>>>>>>の対に囲まれ、====で区切られて表示されます。最初の部分は、文書名に続いて、更新操作前のバージョンが置かれています。第2の部分は、>記号列の後にバージョン番号に続いて、レポジトリ版が置かれます。

CVS コマンド：

```
cd $path; cvs diff "." (変更が検出された場合には知らせる)
cd $path; cvs -q update "."
```

ここで\$pathは文書パスを表します。

10.9.3.9 履歴を表示

これは、CVS 文書の全履歴を表示します。 `cvs log "<ファイル名>"` の出力がブラウザに表示されます。

10.9.3.10 改訂情報

LyX は、CVS 版数・著者名・最終コミットの日時をサポートします。これらの情報は、すべて `cvs log -r <ファイル名>` によって抽出されます。詳細については、第 10.9.4.11 節を参照してください。

10.9.4 LyX における SVN コマンド

LyX は、現在 SVN を一部サポートしています。そのコマンドは、ファイル▷バージョン管理メニューにあります。レポジトリへのアクセスに、ssh 経由のパスワード保護アクセスを使用している場合には、端末ウィンドウでの入力を促されます。LyX は、SVN 1.5・1.6・1.7・1.8 に対して試験されています¹³。

10.9.4.1 登録

お使いの文書が改訂管理下に置かれていなければ、メニューにはこの項目だけが表示されます。もし改訂管理が有効ならば、登録項目は表示されません。

本コマンドは、既に文書のあるディレクトリを SVN 管理下に置いている場合にのみ（すなわち `.svn/entries` ファイルが存在する場合にのみ）、お使いの文書を SVN に登録します。つまり、アーカイブはご自分でチェックアウトしなくてはなりません。

すると、文書の最初の説明文を入力するように促されます。登録したファイルは、まだコミットされたわけではないことに注意してください。

実行される SVN コマンドは、`svn add -q "<ファイル名>"` です。

スイッチの説明は `man svn` をお読みください。

10.9.4.2 変更をチェックイン

ファイルの編集を終えたら、変更をコミットします。このとき、変更の説明文を入れるよう促されます。その後、変更がコミットされます。

SVN コマンド¹⁴：`svn commit -q -m"<入力した説明文>" "<ファイル名>"`

¹³ほとんどのコマンドは、1.4 でも動作します。10.9.4.7 を参照してください。SVN 1.8 では、ファイルがロックされている時にパーミッション変更点を戻してしまうという現時点では未解決の問題が存在しています。

¹⁴ロックが有効になっていない場合です。第 10.9.4.9 節をご覧ください。

10.9.4.3 チェックアウトして編集

このファイルの変更点をレポジトリから取得して更新します。すべての干渉点は、手動で解決しなくてはならないので、この機能を使う前にSVNのマージ法と干渉解決法を理解しておくようにしてください。

SVN コマンド¹⁵ : `svn update --non-interactive "<ファイル名>"`

10.9.4.4 リポジトリ版に戻す

これは、最後のチェックイン以降、文書に加えられた変更をすべて廃棄します。変更が廃棄される前に警告が表示されます。

SVN コマンド : `svn revert -q "<ファイル名>"`

10.9.4.5 コピー

これは、現行文書のコピーを生成します。最後のチェックイン以降、変更が加えられていないサラの文書であることが要求されます。ファイル名とコピー操作の説明を入力するように促され、その後、新コピーがローカルとリポジトリの両方に生成されます。コピーされた文書と元の文書の親ディレクトリが異なる場合には、コピーにインクルードされたファイルの相対パスは、(ファイル名をつけて保存...と同様に) 全て調整されます。最後に、元の文書の代わりに、コピーされたファイルが読み込まれます。

SVN コマンド :

```
svn copy -q "<ファイル名>" "<新ファイル名>"
svn commit
```

10.9.4.6 ファイル名変更

これは、現行文書のファイル名を変更します。最後のチェックイン以降、変更が加えられていないサラの文書であることが要求されます。ファイル名とコピー操作の説明を入力するように促され、その後、ローカルとリポジトリの両方でファイル名が変更されます。新旧ファイル名の文書の親ディレクトリが異なる場合には、コピーにインクルードされたファイルの相対パスは、(ファイル名をつけて保存...と同様に) 全て調整されます。最後に、新ファイル名を使ってファイルが再度読み込まれます。

¹⁵同上.

SVN コマンド：

```
svn move -q add "<ファイル名>" "<新ファイル名>"
svn commit
```

10.9.4.7 レポジトリを使ったローカルディレクトリの同期¹⁶

上記の各コマンドはすべて、共通の短所を持っています。すなわち、これらのコマンドは現在の文書しか取り扱わないのです。もし、作業中の文書に図を含めたり、外部 .tex ファイルを取り込んだりすれば、管理はずっと複雑になります。そこで LyX は、文書のおいてあるツリー全体の更新をサポートします¹⁷。これは、subversion 管理を知らなかったりレポジトリに新しい素材をコミットする勇気のない人たちと共同作業をする羽目になったときに、ひじょうに役に立ちます。

ローカル辞書をレポジトリを使って更新コマンドは、ディレクトリ全体を更新し、マージに衝突が発生した場合には、ローカル版は保存されるので、意図しないデータロスが発生しません。ローカルな変更が発見された場合には、更新が開始される前にユーザーに警告が与えられます。

SVN コマンド：

```
svn diff $path (変更が発見された場合には通知します)
svn update --accept mine-full $path
```

ここで \$path は文書のパスを示します。

10.9.4.8 履歴を表示

これは、SVN 文書の全履歴を表示します。svn log "<ファイル名>"の出力がブラウザに表示されます。

10.9.4.9 ファイルロック

別々の改訂管理システム間でファイルを交換しようとする時、二人のユーザーが同じ文書（の一部）を編集している場合、マージが衝突する問題を引き起こします。そのような衝突が発生した場合、手動で解決しなくてはなりません。その合理的な解決法としては、与えられた時間に一人のユーザーだけがファイルを編集することが許されるように保証するための、何らかのロック機構を導入することが考えられます。

¹⁶このコマンドは、subversion ≥ 1.5 でのみ動作することに注意してください。

¹⁷すべての外部ファイルが、文書と同じディレクトリかそれ以下のサブディレクトリに来るようにファイル構造を構成する必要があります。

SVNには、このように相互に排他的なファイルアクセスを提供する機構が二つあります。ロックと、`svn:needs-lock`svn ファイル特性に依拠した書き込み権限の自動設定（第 10.9.4.10 節を参照）です¹⁸。ある文書について、この特性が検出されたときには、LyX は自動的に文書編集に SVN ロックを使用するようになり、チェックイン・チェックアウト機構は RCS と同じ枠組みに切り替わります。これは特に、LyX でのファイルの使われ方に以下の二つのモードがあることを意味します。

- 非ロック状態。読み込まれたファイルは、読み込み専用モードになります。編集するためには、チェックアウトする必要があります。チェックアウトを行うと、レポジトリからの更新と、書き込みロックの獲得が行われます。ロックの獲得ができないと、非ロック状態から出ることができません。
- ロック状態。読み込まれたファイルは、「通常の」編集モードになります。他のユーザには、このファイルを編集する許可が与えられません。チェックインを行うと、変更のコミットと書き込みロックの解放が行われます。文書に変更が加えられなかった場合には、コミットは実行されず¹⁹、書き込みロックの解放のみが行われます。

SVN コマンド：

```
チェックイン： svn commit -q -m"<説明>" "<ファイル名>"
                svn unlock "<ファイル名>"
```

```
チェックアウト： svn update "<ファイル名>"
                  svn unlock "<ファイル名>"
```

10.9.4.10 自動ロック特性

上述したような `.lyx` ファイルに対する書き込み権限の自動設定は、ファイルバージョン管理のロック特性を使用で行うことができます。このコマンドは、SVN サーバでファイルがロックされていないときのみ使用可能状態になります（つまり、使用するにはチェックアウトが必要です）。

SVN コマンド：

```
チェックイン： svn propset svn:needs-lock ON "<ファイル名>"
```

```
チェックアウト： svn propdel svn:needs-lock "<ファイル名>"
```

¹⁸<http://svnbook.red-bean.com/en/1.2/svn.advanced.locking.html>

¹⁹コミットは実行されないのに、コミットメッセージを入力するよう促されますが、驚かないでください。

10.9.4.11 文書の改訂情報

文書中で改訂情報を有効化するにはいくつかの方法があります。

- LyX による直接サポート
 - ツリー改訂情報 (`vcs-tree-revision`). `svnversion` コマンドの出力を返します. 下記の表が, 結果の読み方のヒントを与えるでしょう.

出力	意味
4123:4168	改訂版の混合した作業コピー
4168M	修正された作業コピー
4123S	スイッチされた作業コピー
4123P	チェックアウト頻度が低かったことによる不完全な作業コピー
4123:4168MS	改訂版が混合し, 修正され, スイッチされた作業コピー

- ファイル改訂情報. `svn info --xml file.lyx` の出力を解析したものが返されます. サポートされるフラグは
 - * 最終コミットのバージョン番号 (`vcs-revision`)
 - * 最終コミットの著者 (`vcs-author`)
 - * 最終コミットの日付 (`vcs-date`)
 - * 最終コミットの時間 (`vcs-time`)

これらの情報は, 情報差込枠で得ることができます (b). これらの情報は, ファイルを `svn` 管理下に置いたときのみ (すなわち使用中文書の `.svn` ディレクトリがあるときのみ) 利用可能です.

- もう一つの—ハッカー的な—方法は, SVN キーワードを使用する方法です²⁰. 簡単に言えば, ファイルキーワード特性を設定して (例: `svn propset svn:keywords 'Rev' file.lyx`), 文書中にキーワード `TeX` コード²¹ タグ (例: `Rev`) を貼り付けるのです. このようにして, SVN クライアントは, 各更新やコミットの後に自動的に改訂番号を代入することができるのです (例: `$Rev: 59 $`). このアプローチにはいくつか問題があります. 第一に, `TeX` の世界では「\$」文字は数式を表すのに使われるので, LyX 中の `Rev` という数式表現が, すべて `Rev : 59` になってしまいます. その他の `Id`・`Date`・`Author` などといったキーワードも

²⁰<http://svnbook.red-bean.com/en/1.4/svn.advanced.props.special.keywords.html>

²¹これは, LyX がキーワードタグの途中で改行してしまわないことを確実にするための, 簡単な方法です.

同様です。第二に、SVN 出力は使用中のロケールに依存するので、Date が用いられると、SVN は、問題を引き起こすような文字列を容易に生成してしまいます。第三に、文書中には、番号だけでなく「Rev: 59」という文字列全体が表示されてしまいます。Subversion がユーザ定義のキーワードを実装するまでは、このアプローチを信頼性を持って使用したり、LyX がこれを直接サポートすることは、難しそうです。

10.9.5 SVN と Windows 環境

ユーザが、かなり早いうちに自分自身でコマンドライン操作を理解できないのであれば、TortoiseSVN を使った方が賢明でしょう。—P. A. ルービン

10.9.5.1 準備

LyX を導入して、Subversion レポジトリへのアクセス権を持っている他に、ユーザは Subversion クライアントプログラムを導入する必要があります。クライアントプログラムの Windows 用インストーラは [CollabNet](#) にあります。ユーザは [TortoiseSVN](#) も導入することを考えるでしょう。これは Subversion の操作を Windows Explorer のコンテキスト（右クリック）メニューに統合します。LyX 外部で行う操作は、一般的に Explorer コンテキストメニューを使った方が便利です。TortoiseSVN は、LyX 自身が使用するクライアントプログラムの代わりとは成りませんので、注意してください。

10.9.5.2 文書を Subversion 管理下に置くには

LyX 文書を Subversion のバージョン管理下に置く前に、その親ディレクトリをバージョン管理下に置かなくてはなりません。文書を、既にレポジトリに存在するプロジェクトに追加する際には、新規文書が置かれることになるディレクトリへ、プロジェクトをチェックアウトすることによって実現します。プロジェクト自体がバージョン管理下でない場合（たとえばこの文書が新規プロジェクトの最初のファイルである場合）には、このディレクトリをレポジトリにインポートしなくてはなりません。これは、LyX の外部で行います。インポートもチェックアウトも共に、TortoiseSVN を使えば、Explorer のコンテキストメニューから簡単に終わることができます。あるいは、DOS プロンプトでコマンドライン・クライアントを使って成し遂げることもできます。TortoiseSVN を使ったプロジェクトのインポート手続きを下記で説明します。レポジトリは既に存在し、新規プロジェクトを C:\new project で開始するものとし

まず、Subversion クライアントプログラムを使うための情報を得るには、DOS シェルで `svn --help` を実行してください。

1. Windows Explorer で `C:\new project` を見つけ出して右クリックし、TortoiseSVN▷Repo-browser を選択してください。必要であれば、レポジトリの URL を調整し、OK をクリックしてください。
2. 新規プロジェクトフォルダを置くレポジトリレベル（多くの場合最上位レベル）を右クリックし、Create folder... をクリックしてください。プロジェクトフォルダの名称を入力して OK をクリックしてください。必要であればログファイル用のメッセージを入力し、再び OK をクリックしてください。新規プロジェクトフォルダがレポジトリに現れるはずですが、最後に OK を再度クリックして、レポジトリブラウザを終了してください。
3. 再度 `C:\new project` を右クリックし、今回は SVN Checkout... を選択してください。レポジトリにたった今作成した、プロジェクトフォルダの URL を選択し、チェックアウトディレクトリを `C:\new project` に設定してから、OK をクリックしてください。フォルダが空でないことを警告されますが、OK をクリックして進んでください。すると、`C:\new project` の下に `.svn` ディレクトリが作成されます。
4. LyX で文書を新規作成するか開くかして、ファイル▷バージョン管理▷登録をクリックします。ログメッセージを入力して OK をクリックし、文書をバージョン管理にコミットしてください。

爾後、ファイル▷バージョン管理メニューのすべての機能が使用できるようになります。Windows Explorer の TortoiseSVN コンテキストメニューか、コマンドプロンプトでの Subversion クライアントプログラムを使用して、文書をチェックイン・チェックアウトしたり、履歴を見たりといったことができるようになります。

10.9.5.3 Windows 下で SVN と共に SSH トンネルを使うには

Linux に較べて、Windows で SVN クライアントを SSH 経由で通信するようにさせるのは、かなり厄介な仕事です。どのようにクライアント側を設定すればよいか、少なくとも幾分かのヒントをお示しするつもりですが、ssh と Windows コマンドラインの知識が前もって必要であり、それと共にかなりの量のフラストレーションが溜まることを覚悟しておいてください。

1. 前節で述べたように、Windows 用の SVN クライアントを入手してください。これが新規の導入である場合には、何らかの SVN コマンド（たとえば `svn --version`）を実行して、`config` ファイル群を生成させてください。これらは後で変更を加える必要があります。
2. Windows 用の ssh クライアントを選んでください。選択肢はたくさんありますが、ここでは Putty tools²² のものを使用することとしましょう。クライアントが接続の度にパスワードを訊いてこないように、接続を自動化する必要があります。簡単にするために、追加的なパスワード保護などを加えないキーのみを使用することとします。
 - a) `puttygen` を使ってキーを生成します。ppk プライベートキーファイルを保存し、パブリックキーファイルはサーバ側に置きます。SVN サーバが Linux で走っている場合、このパブリックキーの書式は Linux の `openssh` と互換性がないために、「Public key for pasting into OpenSSH authorized_keys file:」編集フィールドから、直接サーバの `~/.ssh/authorized_keys` または `authorized_keys2` ファイルにコピー&ペーストを行わなくてはならないことに注意してください。
 - b) Putty の `plink` を入手してください。SVN の `config` ファイル²³ 中の `[tunnels]` セクションに `ssh` コマンドを設定してください。例：
`ssh=c:/path/plink.exe -i c:/path/private_key.ppk24`
3. SVN アーカイブをチェックアウトしてください。例：
`svn co svn+ssh://user@server/repository_path`

10.9.5.4 行末変換

共著者同士が Linux と Windows を混在させているとき、LyX は .lyx ファイル中で異なった行末記号を使用することになります。これは、LyX の機能に関する限り問題になりませんが、コミット差分は巨大になり、マージではコンフリクトを引き起こしがちです。幸いにして、SVN は、.lyx ファイルを `native` モードに切り替える際²⁵,

²²<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

²³正確なパスは Windows のバージョンに依りますが、通常は `c:\Documents and Settings\user\Application Data\Subversion\config` や `AppData\Roaming\Subversion` 辺りにあります。

²⁴通常、正確なコマンドを機能させるには長い時間がかかり、それは多くの物事に依存します。たとえば、リモートサーバが Putty などの既定セッションとして保存されないようにしてください。うまく行かない場合には、まず SVN なしで `plink` 経由の接続を試してください。

²⁵手短には `svn propset svn:eol-style native FILE_NAME`

CR/LF問題をどのように取り扱えばいいかを知っているので大丈夫です²⁶。

10.9.6 LyXにおけるGITコマンド

LyXで、最小限のGITコマンドがサポートされるようになりました。これらのコマンドは、ファイル▷バージョン管理にあります。リポジトリのアクセスに、パスワード保護されたsshを使用している場合には、ターミナルウィンドウでの操作をするように促されます。LyXは、GIT 1.7/2.30に対して試験を行っています²⁷。

GITと他のバージョン管理システムとの大きな相違点は、GITの配布形態にあります。伝統的なバージョン管理システムでは、リポジトリをホストする、中心となるサーバーがあります。ユーザーは変更点をサーバーにコミットし、他のユーザーが行った更新分をサーバーから受け取ります。一方、GITでは、ユーザーは、ローカルのリポジトリにコミットを行います。ローカルリポジトリは、GITコマンドのpullまたはpushを使って、1つ以上のリモートリポジトリと同期されます。LyXは、リモートのGITリポジトリとは、一切関わりません。CVSやSVNの集権的リポジトリに対して行うのと同じことを、ローカルのリポジトリに対してのみ行います。リモートのGITリポジトリを使用している場合には、お好みのGITクライアントを使用して、pullまたはpushの操作を行う必要があります。

10.9.6.1 登録

お使いの文書が改訂管理下に置かれていなければ、メニューにはこの項目だけが表示されます。もし改訂管理が有効ならば、登録項目は表示されません。

本コマンドは、既に文書のあるディレクトリをGIT管理下に置いている場合にのみ（すなわち.git/indexファイルが存在する場合にのみ²⁸）、お使いの文書をGITに登録します。つまり、アーカイブはご自分でチェックアウトしなくてはなりません。

すると、文書の最初の説明文を入力するように促されます。登録したファイルは、まだコミットされたわけではないことに注意してください。

実行されるGITコマンドは、`git add "<ファイル名>"`です。

スイッチの説明は`man git`をお読みください。

²⁶<http://svnbook.red-bean.com/en/1.4/svn.advanced.props.file-portability.html>

²⁷Windowsユーザーはgitにパスが通っていることを確認する必要があります。これは、LyX中からツール▷設定▷パス▷PATH接頭辞で設定できます。

²⁸このためには、`git init`「と」初回リポジトリコミット（あるいは少なくとも`git add`）を手動で行わなくてはなりません。

10.9.6.2 変更をチェックイン

ファイルの編集を終えたら、変更をコミットします。このとき、変更の説明文を入れるよう促されます。その後、変更がコミットされます。

GIT コマンド：`git commit -q -m "<入力した説明文>" "<ファイル名>"`

10.9.6.3 リポジトリ版に戻す

これは、最後のチェックイン以降、文書に加えられた変更をすべて廃棄します。変更が廃棄される前に警告が表示されます。

GIT コマンド：`git checkout -q "<ファイル名>"`

10.9.6.4 ファイル名変更

これは、現行文書のファイル名を変更します。最後のチェックイン以降、変更が加えられていないサラの文書であることが要求されます。ファイル名とコピー操作の説明を入力するように促され、その後、ローカルとリポジトリの両方でファイル名が変更されます。新旧ファイル名の文書の親ディレクトリが異なる場合には、コピーにインクルードされたファイルの相対パスは、(ファイル▷名前をつけて保存...と同様に) 全て調整されます。最後に、新ファイル名を使ってファイルが再度読み込まれます。

GIT コマンド：

```
git mv "<ファイル名>" "<新ファイル名>"
git commit
```

10.9.6.5 履歴を表示

これは、GIT 文書の全履歴を表示します。`git log "<ファイル名>"`の出力がブラウザに表示されます。

10.9.6.6 改訂情報

LyX は、ファイルの最終コミットの GIT ハッシュ番号・その短縮形・著者名・最終コミットの日時をサポートします。これらの情報は、すべて `git log -n 1 --pretty=format:%H%n%h%n%an%n%ai` によって抽出されます。ツリーバージョン情報は、`git describe --abbrev --dirty --long` で取得されます。詳細については、第 10.9.4.11 節を参照してください。

10.9.7 さらなる調整

最近になって `vc-command` 機構が付け加えられたおかげで、`LyX` のパワーユーザは、改訂管理のための自製コマンドを作ることができるようになりました。

たとえば、以下のようにして、二つの `TortoiseSVN` コマンドを直接統合することができます。

コミット：`vc-command DR "." "TortoiseProc /command:commit /path:$p"`

元に戻す：`vc-command DR "." "TortoiseProc /command:revert /path:$p"`

10.9.8 バージョン管理と文書比較

バージョン管理は、通常 `diff` ダンプを生成することによって、改訂版の間の変更点を検査するのが、典型的な使用法です。これは平文ファイルには有効なのですが、複雑な構造を持つ `LyX` ファイルの場合には、あまり便利ではありません。そこで、`LyX` は独自の文書比較機能を提供しています。この機能呼び出すには2つの方法があります。1つは、`LyX` の `vc-compare` 関数を直接呼び出す方法であり（詳細は `LyX` 関数説明書を参照）、もう1つはツールバーアイコンによる方法、さらにはメニュー項目を使用する方法です。当該文書のうち、任意の2つの改訂版を比較することもできますし、編集時の現行版を古い改訂版と比較することもできます（このとき改訂版遡及数「0」は、編集時のファイルと最後にコミットした改訂版との比較を意味します）。

この機能は、`SVN` と `RCS` でサポートされていますが、`RCS` は、より複雑なバージョン付けスキームを持っているため、制約があります。ダイアログ中で改訂版を指定する際の数字は、つねに `RCS` 改訂番号の最後の数を指し示します（例：「1.2.35」版のときには「35」）。現在のところ、改訂の `GIT` 表記はサポートされておらず、`HEAD~X` の形で表記される `X` ステップ前の改訂版を、編集中文書と比較することのみが可能です。

10.10 文芸プログラミング

KAYVAN SYLVAN (kayvan@sylvan.com) 改訂 EDMAR WIENSKOSKI JR. (edmar-w-jr@technologist.com) 原文

10.10.1 はじめに

この節の主な目的は、文芸的プログラミング (literate programming) のために、`LyX` をどう使えばよいかをお見せすることです。ここでは、あなたがこのプログラミング

技術をよくご存じで、「tangling」や「weaving」の意味するところを知っているものとして説明します。そうでない場合には、以下の各節にあるウェブリンクをたどってください。そこには、かつての開発の歴史から最新のツールチップまで、たくさんの良質なドキュメンテーションがあります。

また、あなたが、LyXの設定を変更することを厭わない程度までLyXに習熟しており、Xリソースファイルにも親しんでいることを前提にしています。そうでない場合は、あなたの必要とするトピックを扱っている、LyXの他の説明書で調べてください。

10.10.2 文芸プログラミング

以下は、文芸プログラミング FAQからの抜粋です。

文芸プログラミングは、人間が読むのに適した形でのドキュメンテーションとソースの組み合わせです。実際、文芸的プログラムは楽しい読み物であるべきであり、さらには参加したいとさえ思わせるべきなのです！（ボブ、ごめん。言わずにはいれなかった。）一般的に、文芸プログラムは、ソースとドキュメンテーションを同一ファイル上に結合したものです。そして、文芸プログラミングツールは、ファイルを解析して、読むことのできるドキュメンテーションか、あるいはコンパイル可能なソースを生成します。文芸プログラミングのWEBスタイルは、D. E. Knuthが、T_EX組版ソフトウェアを開発中に作り出したものです。

他のエキスパートの発言です。

文芸プログラミングは、冗長なコメント付けと何が違いますか？

3つの際立った特徴があります。重要な順に列挙すれば、

- 推敲の順序が柔軟
- ブラウジングの自動サポート
- 特に図式と数式を含んだドキュメンテーションの組版

などがあります。

あなたの興味に火を付けたところで書誌情報を見てください。

10.10.2.1 書誌情報

完全な文芸プログラミング FAQは

Literate Programming FAQ <http://www.literateprogramming.com/lpfaq.pdf>

にあります。

このFAQには、23（二十三です!）の文芸プログラミング用ツールが列挙されています。一部のツールは、特定のプログラミング言語用に特化したものですが、一般的な対象を持つものもあります。私は自分で使うのに、いくつかの理由から NOWEB を選びました。

- 説明文書を L^AT_EX でも HTML でも出力できる。
- オープンアーキテクチャを持っているので、新しいフィルタをプラグインとして加えたり²⁹、自分が必要な特別な処理を実行することが容易である。
- すでに利用可能となっているフィルタの良質なコレクションがある（HTMLはその一つです）。
- 無料である。

Noweb のウェブページは

Noweb ホームページ <https://www.cs.tufts.edu/~nr/noweb/>

にあります。ここから始めれば、たくさんの興味深いリンクや、文芸プログラミングの用例にたどり着くことができます。

10.10.3 LyX と Noweb を使った文芸プログラミング

LyX における文芸プログラミングのサポートは、LyX の汎用変換機構を使用して実現しています。このサポートは「Noweb 非依存」的な方法で提供されているので、LyX の設定を変更するだけで、自分の好きな他の文芸プログラミングツールを使って、この新しい LyX の機能を利用することができます。

10.10.3.1 ドキュメントとコードの生成 (weaving および tangling)

noweb モジュールの使用 Noweb と LyX の導入に成功したならば、新規文書を開いたときにはいつも、文書クラスを選択した後、文書▷設定メニューで「noweb」モジュールを追加するようにしてください。Noweb が正しく導入されていれば、「モジュール」

²⁹フィルタは、与えられたデータストリームを読み込み、操作されたデータストリームを出力するプログラムです。このようにして、(文芸コードを含んだ) WEB ファイルを、C プログラムコードや L^AT_EX コードのみを含むファイルに変換することができます。

リンクをクリックした際、利用可能リストに「noweb」モジュールが表示され、文書に追加できるようになっています。

コードの入力 LyX では、CHUNK という名前の特別差込枠を使えば、コードを入力することができます。Noweb は次のように chunks を区切ります。

```
<<My code>>=
code
more code
even more code
@
```

問題は<<と@の間にあるものを完全にそのままの形で扱わなければならないことです。つまり、LyX は、この間に書かれたものを特別に解釈することは決してあってはなりません。これも CHUNK によって取り扱われ、通常のテキスト差込枠のように見えますが、自由に空白を入れることができます。

特殊な注釈として、chunk 内で Noweb の「%def」を使えば、Noweb の identifier の相互参照に項目を加えることができます。次の例を見てください。

```
<<My chunk>>=
def some_function(args):
    "This is the doc string for this function."
    print "My args: ", args
@ %def some_function
```

これをどのように使って、その結果相互参照がどうなるかは *LIBDIR/examples/listerrors.lyx* 中の文芸 python プログラムを見るとよくわかります。

説明文書の生成 この時点で、適切な文書クラスを持ち、いくつかのコードとテキストが入った新規文書ファイルが作成されているはずです。これをどうやって印刷したらいいのでしょうか。答えは簡単で、通常の文書で行うように文書▷表示等を選ぶだけです。特別な手続きは必要ありません。

理解を助けるために、LyX の内部で何が起っているかを説明しておきましょう。

1. 文書▷更新が選択されると、L^AT_EX ファイルがつくられます。

文書が文芸プログラミング用のクラスであった場合には、拡張子に（設定パネルで設定した）文芸プログラム用のものをつけられます。それ以外は.tex が拡張子に使われます。

2. 違いがファイル名だけなのに注意してください。LyXによる特別な処理は必要ありません。CHUNK 差込枠でコードを入力するのだけがポイントです。
3. 文書が何らかの文芸プログラムクラスなら、LyXは、LyXからNowebへの内部変換子を用い、それからNowebからL^AT_EXへの変換子³⁰を使ってL^AT_EXファイルを生成します。
それ以外の場合はこのステップを飛ばします。
4. 最後にL^AT_EXが実行され、通常の文書と同じ後処理が行われます。

様々な変換子が実行するコマンドを変更すれば、簡単に、特定の「文芸プログラムツール」から変更することができます。

コードの生成 ビルドメニューが選ばれた場合や、それに対応するツールバーのボタンが押された場合、上のステップ1と同様にL^AT_EXファイルが生成されます。それからLyXはNoweb→Programのコンバータを起動します。通常は、このコンバータは(他のコンバータと同じように)2つの部分からできています。

1. 変換プログラム。このプログラムは、一つのフォーマットから他のフォーマットへの変換を行います(この場合は、Nowebフォーマットからプログラムの疑似フォーマットへ変換します)。
2. エラーログ解析子。これはエラーメッセージをLyXが理解できる形へ書き直します。これによって、LyXはファイルバッファの正しい位置へエラーボックスを配置することができます。

冒頭部、「変換子」設定は

```
build-script $$i $$r
```

と指定します。これは基本的に、LyXが”build-script”(プログラムかスクリプト名)を(一般にはLyXの一時ファイル用のディレクトリにつくられる)、Nowebファイルの名前と元のLyXファイルのディレクトリパスを使って呼び出すことを意味しています。

以下に「build-script」の一例を書いておきます。パスが通っているディレクトリに置いて使います。

³⁰これらの変換子は、ツール▷設定パネルの変換子タブに定義されています。変換子の一般的な情報については、取扱説明書『カスタマイズ篇』の「変換子」の節をご覧ください。

10 LyX サーバー

```
#!/bin/sh
#
notangle -Rbuild-
script $1 | env NOWEB_SOURCE=$1 NOWEB_OUTPUT_DIR=$2 sh
```

変換子設定の次の部分は「フラグ」で、これは

```
parselog=$$s/scripts/listerrors
```

に設定されます。

これは、”build-script” プロセスが生成したエラーを”listerrors” プログラムを介して処理するものです。

ビルドは、通常 LyX の一時ディレクトリで行われるので、変換によって生成されたファイルはそのディレクトリに置かれます。LyX は、主要ファイルとみなされるものをコピーして取り出しますが、Noweb → プログラム変換は複数のファイルを生成することがあるので、LyX を閉じる際に大半が削除される可能性があります。これが、NOWEB_OUTPUT_DIR 環境変数を渡して、build-script chunk が生成したファイルをその場所に置かれるようにする理由です。

文章中でのプログラムの構築操作の説明 LyX と Noweb の統合の最後の部分は、“build-script” chunk です。一般に、あなたのプログラムの構築操作の説明それ自身を chunk に入れて文章中に含めなければなりません。上の例の noweb 用の build-script では、chunk を見つけるのに notangle コマンドを使い、sh を使って起動します。

以下に典型的な chunk がどのようなものかを書いておきます。

```
<<build-script>>=
#!/bin/sh

if [ -z "${NOWEB_SOURCE}" ]
then
NOWEB_SOURCE=myfile.nw
fi
[... code to extract files ... use NOWEB_OUTPUT_DIR here ...]
[... code to compile files ...]
@
```

ファイル▷用例を開く▷モジュール▷Noweb Listerrors またはファイル▷用例を開く▷モジュール▷Noweb2LyX をご覧ください。このファイルには異なる実装の「listerrors」

プログラムが含まれていて、これまで説明してきた各々の部分が全体としてどのようになるかをわかってもらうための例です。面白いことに、これらのファイルには、Python と Perl が各々使われていて、L_YX の文芸プログラミングのサポートがプログラム言語に依存しないことを示しています。

10.10.3.2 L_YX の設定

すべての文芸プログラムのための設定は、ツール▷設定パネルの「ファイル処理」タブですることができます。重要な部分を見ておきましょう。

「NoWeb」形式 ファイル形式タブで設定します。ここで、Noweb 特有の設定を行います。GUI 名を NoWeb に、拡張子を .nw に設定してください。これによって、L_YX に変換プロセスの第一段階でつくられるファイルに .nw という拡張子をつけることを指示します。

プログラム形式 これは、変換の終着点を示すためだけの、空の形式です（これがあ
ることによって、変換子を作ることができます）。

NoWeb->L_AT_EX この変換子は、文芸的文書の「weaving」を行います。Noweb の場合、「noweave -delay -index \$\$i > \$\$o」に設定されます。

NoWeb->Program これは「tangling」を行います。上述のように、変換子は「build-script \$\$i \$\$r」、フラグは「parselog=\$\$s/scripts/listerrors」となります。

10.10.3.3 デバッグ機能拡張

ddd/gdb または他のデバッガで使える「server-goto-file-row」関数が新たに L_YX サーバに実装されました。

ddd/gdb でデバッグしているとき、一つのキーストロークで現在実行している位置でテキストエディタを起動できます。既定の ddd の設定では Shift-Ctrl-V になっています。この設定は ddd でツール▷設定▷ファイル書式ウインドウを表示して、編集プログラムエントリを変えると変えることができます。

新しい L_YX サーバの関数と、この ddd の機能を使うために編集プログラムを次のように設定してください。

```
echo "LYXCMD:monitor:server-goto-file-  
row:@FILE@ @LINE@" >~/lyxpipe.in
```

10 LyX サーバー

こうしておくことで ddd を使っていてプログラムの編集したい場所があったとき、(ddd のウィンドウで) Shift-Ctrl-V を押すだけで ddd は LyX サーバに情報を送り、LyX は同じファイルを表示し、カーソルは ddd が示した場所へあらわれます。これでもうデバッグの最中に推測しながらスクロールして場所を探す必要がなくなります。

この機能が動くには LyX サーバが利用可能でなければなりません (既定では LyX サーバーは使えないようになっています)。

第 10.2 節にその方法が説明されています。

10.10.3.4 ツールバー拡張

ツールバーに加えることのできる 6 つの新しいボタンがあります。そのうちの 4 つのボタンは標準・節・ \LaTeX ・LyX コードの段落環境へのショートカットであり、1 つは特別差込枠 Chunk へのショートカットです。最後の一つはファイルメニュー項目の「Build Program」へのショートカットです。

LyX はツールバーで使うためのいくつかのボタンを用意しています。私のツールバーはこの 6 つのショートカットに文書▷更新▷DVI と 文書▷表示▷DVI を組み合わせたものになっています。

```
Toolbar
  Layouts
  Icon "layout Standard"
  Icon "layout Section"
  Icon "layout latex"
  Icon "layout LyX-Code"
  Icon "flex.insert Chunk"
  Separator
  Icon "buffer-view"
  Icon "buffer-typeset"
  Icon "build-program"
  Separator
  .
  .
  .
End
```

10.10.3.5 色の変更

LyXではいくつかの色を**設定**で変えることができます。人々が困りがちなのは、L^AT_EX フォント色でしょう。これは、既定では赤になっていますが、chunk は L^AT_EX フォントを使うことになっていて、文芸プログラミング文書には多くの chunk が現れるために、すべて赤で見ていると疲れてしまうかもしれません。これは、操作性▷色タブで変更することができます。

もう一つは、改行文字が画面に表示されていることでしょう。改行文字の色を背景の色と同じにすることで改行文字を見えなくできます。私は改行文字を背景の色に似ているが少し違う色にすることを勧めます。こうすると、改行文字があるのはわかりますが、じゃまにならないでしょう。

10.10.4 LyX と knitr/Sweave

knitr と Sweave へのサポートについては、ヘルプ▷用途別説明書▷K_ni_tr およびファイル▷用例を開く▷モジュール▷R_nw (knitr)、ヘルプ▷用途別説明書▷S_we_av_e およびファイル▷用例を開く▷モジュール▷S_we_av_e に説明があります。

索引

LaTeX パッケージ

enumitem, 74, 75

箇条書き

調整, 74

余白, 74