

http://www.ascii.co.jp/EWB/

この書籍はEWB(Editor's Work Bench)によって組版されています。 EWBに関する詳しい情報は、上記のURLをご覧ください。

はじめに

昨今のパソコンおよびインターネットの普及にともない、出版の電子化は避けて通れないものとなっています。マルチメディアが台頭し始めた頃の CD-ROM に始まり、インターネット上でのHTML、さらには書籍イメージを再現する PDF 等、さまざまな形式への電子化が試みられています。

株式会社アスキーでは書籍の電子化へ対応するため、DTP の普及以前から EWB (Editor's Work Bench) というシステムを開発してきました。

EWB の開発は 1986 年からはじまりました (EWB バージョン 1)。このバージョンは、編集部内の作業工程を崩さないことに主眼をおき、文字原稿の棒打ち出力のみをサポートしました。この中に、プリンタを使った写植エミュレーション出力と呼ばれる機能があり、社内校正やラフレイアウトに貢献し、大幅な工程短縮とコスト減を達成しました。

1990年にEWB バージョン 2 というシステムへ発展し、組版エンジンに"日本語 TeX (pTeX)"を採用しました。この日本語 TeX は、スタンフォード大学のクヌース教授が開発した TeX を、アスキーが日本語化し縦組対応したものです。さらに、日本語 TeX で作成した組版結果を写研などの写植機で印画紙出力ができるようにし、出版物に耐え得る品質に向上させました。

1996年のEWBバージョン3で、PostScript言語を採用して画像を扱えるようにしました。これにより印画紙への貼り込み作業がなくなり、また面付けデータを作成して直接フィルム出力ができるようになったため、さらなるコスト削減と時間短縮が可能になりました。

その後、pTEX の改良にともなってバージョン 3.1 が開発され、1999 年 10 月にフリーソフトとして公開されました。 さらに PDF への対応を強化したバージョン 3.2 をリリースし、また 3.3 では体裁作成機能を強化しました。

現在、EWB はアスキーの書籍の半分以上で使用されています。プラットフォームが UNIX 系 OS であったり、画面上で組版イメージを直接操作できないといった敷居はありますが、慣れてしまうと手放せなくなるような軽快さをもっています。書籍の体裁によっては大きな省力化が実現できますので、編集 / 制作の手段のひとつとして EWB を使っていただけると幸いです。

2002 年 1 月 株式会社アスキー 出版技術グループ

本書の構成

本書は、利用目的に応じて4つの部分に分けられており、各部は以下のような構成になっています。

第1部基礎編

第1部では、電子編集の最も基本的な概念と EWB の特徴や表現力について解説を行います。 はじめて EWB に接するユーザーあるいは今後 EWB を導入しようと考えている方は、まず第1部から読みはじめることをお勧めします。

第2部 実践編

第2部では、実際に EWB を使う上で必要となる操作方法やコマンド、さらに EWB を効率的に使いこなすための知識やノウハウを集中的に解説します。原稿入力からデータ入稿までの作業手順を追って、具体例を示します。

第3部応用編

第3部は、標準では扱えない機能を、どうのようにして扱えるようにするのかという話題を中心に解説を行います。EWBの中核となる"トリガ"や"日本語 T_EX"を拡張するために必要な情報を提供しています。

テクニカルな情報を掲載していますので、マクロ拡張には欠かせない内容となっています。

Appendix

Appendix では、トリガー覧や書体一覧、コマンドリファレンスなど、EWB を運用している時に必要あるいは有益な情報を一覧表形式でまとめて掲載しています。

目 次

はじめに・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・				
	本	書の構成・・・・・・・・・・・・・・・・・・・・・・・・・・・・4		
第一	1	部 基礎編		
1	章	電子編集とは・・・・・・・・・・・・・・・・・・・・13		
	1.1	電子化をすると13		
	1.2	電子編集システムの特色は・・・・・・・・・・・・・・・・・15		
	1.3	うまい電子化をするために ・・・・・・・・・・・・・・・・・・・・・・・・ 18		
2	章	EWB のプロローグ · · · · · · · 21		
	2.1	現場から誕生した EWB · · · · · · · · · · · · · 21		
	2.2	EWB の特徴は何? · · · · · · · · · · · · · · · · · · ·		
	2.3	EWB の作業工程を見る · · · · · · · · · · · · · · · · · · ·		
3	章	トリガの役割り29		
	3.1	トリガとは何か? ・・・・・・・・・・・29		
	3.2	トリガ入力の真の意味は・・・・・・・・・・・・・・・・・34		
	3.3	トリガ入力とコミュニケーション・・・・・・・37		
4	章	EWB の表現力 · · · · · · · · · · · · · · · · · · ·		
	4.1	一般知識		

4.2	文字42
4.3	判型 / 基本組 · · · · · · · · · 46
4.4	ページスタイル・・・・・・・・・・・・・・・・・・・・・・・・48
4.5	見出し・・・・・・・・50
4.6	小組······54
4.7	表組 · · · · · · · · · · · · · · · · · · ·
4.8	脚注 · · · · · · · · · · · · · · · · · · ·
4.9	箇条書62
4.10	キャプション・・・・・・・・・・・・・・・・・・・・・・・・64
4.11	プログラム · · · · · · · · 67
4.12	キートップ・・・・・・・・・・・・・・・・・・・・・・・・69
4.13	罫線 · · · · · · · · · · · · · · · · · · ·
4.14	囲み罫72
4.15	ルビ・・・・・・・・・・・・・・・・・・・・・・73
4.16	傍点 · · · · · · · · · · · · · · · · · · ·
4.17	ハイパーリンク・・・・・・・・77
4.18	索引77
章	レイアウト・・・・・・・・・・・・・・・・81
5.1	便利なオートレイアウト・・・・・・・・81
5.2	思い通りにレイアウトする84
5.3	ワークサイクル・・・・・・・・・・・・・・・・・・・・・・・・・88

穿	3 2	部 実践編 89
1	章	実作業をはじめる前に ・・・・・・・・・・・・・・・ 91
	1.1	実作業の流れ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
	1.2	EWB の作業環境 · · · · · · · · · · · · · · · · · · ·
2	章	原稿入力 · · · · · · · · · · · · · · · · · · ·
	2.1	原稿の文字コード・・・・・・・・・・・・・・・・・・・・・・95
	2.2	原稿の管理・・・・・・・・・・・・・・・・・・・・・・・96
	2.3	文字の入力規則・・・・・・・・・・・・・・・・・・・96
	2.4	編集トリガの入力規則・・・・・・・・・・・・・・・・・100
3	章	体裁入力 · · · · · · · · · · · · · · · · · · ·
	3.1	体裁入力の作業 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
	3.2	体裁入力ファイルの変換 · · · · · · · · · · · · · · · · · · ·
	3.3	体裁入力ファイルの基礎 · · · · · · · · · · · · · · · · · · ·
	3.4	本文用体裁入力ファイルの詳細・・・・・・・・・・・125
	3.5	索引用体裁入力ファイルの詳細・・・・・・・・・・・・152
	3.6	guisted による体裁編集 · · · · · · · · · · · · · · · · · · ·
4	章	EWB の操作 設定から簡単な出力まで179
	4.1	EWB-Shelf とは・・・・・・・179
	4.2	EWB-Shelf のインストール・・・・・・182
	4.3	EWB-Shelf の操作・・・・・・・183

4.4	エラーリカバリー ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
5 章	EWB の操作 レイアウトから索引作成まで ······ ₁₉
5.1	図表類の管理・・・・・・・・・・・・・・・・・・・・・・・19년
5.2	レイアウト修正 · · · · · · · · 200
5.3	目次の生成・・・・・・・・・・・214
5.4	索引作成 · · · · · · · · · · · · 215
5.5	エラー・・・・・・・・・・・・・・・・・・・・・・・・22
6 章	仕上がりファイルの作成 225
6.1	最終チェック・・・・・・・・・・225
6.2	面付け PostScript ファイルの作成 · · · · · · · · · · · · 228
6.3	入稿後の作業・・・・・・・・・・・・232
6.4	PDF 用 PostScript ファイルの作成 · · · · · · · · 233
6.5	原稿のバックアップ・・・・・・・・・・・・・・・・・・238
7 章	GUI シェルによる EWB の操作 · · · · · · · · · · · · 237
7.1	GUI シェルの準備 · · · · · · · · · · · · · 237
7.2	GUI シェルによる作業 · · · · · · · · 238
第 3	部 応用編25
1音	図表ファイルの作成 · · · · · · · · 25
·早 1.1	EWB で扱える図表・・・・・・・・・・25%
	図表ファイルと本文とのリンク・・・・・・・・254
1.4	豊成ファール C字入 Cツフファ

2 章	標準機能の拡張 · · · · · · · · · · · · · · · · · · ·
2.1	EWB の表現力を広げる · · · · · · · · · · · · · · · · · 259
2.2	約物修正の一括処理・・・・・・・・260
2.3	トリガの表現力を広げる
2.4	トリガ定義の拡張・・・・・・・・・263
2.5	マクロの拡張・・・・・・・270
2.6	書体の追加・・・・・・・・・・・・・・・・・・324
3 章	EWB でのトラブル・・・・・・・・・・329
3.1	出力時のトラブル・・・・・・・・・・・329
3.2	組版時のトラブル・・・・・・・・・・・330
3.3	EPS ファイルの作成ミス · · · · · 331
3.4	その他のトラブル・・・・・・・・・・・332
Арр	endix · · · · · · · · 333
Appe	idix A EWB のツール群・・・・・・・・・・・・335
Appe	ıdix B 標準編集トリガー覧表 · · · · · · · · · · · · · · 337
Appe	ıdix C 組版トリガー覧表 · · · · · · · · · · · · · · · · · · 341
Appe	ndix D 書体一覧表······343
D.1	T _E X フォント
D.2	PS 和文フォント 345
D.3	PostScript 従属欧文フォント(プロポーショナルピッチ) · · · · · 347

	D.4	Post	cript 従属欧文フォント(コンスタントピッチ)·····348
	D.5	Post	cript 欧文フォント 350
Α	ppendi	хE	約物 / 記号類一覧表 · · · · · · · · · · · · · · · 359
A	ppendi	x F	「EX / LATEX 書籍一覧表 ············361
Α	ppendi	x G	コマンドリファレンス・・・・・・・363
	G.1	EWB	羽部コマンド一覧363
	G.2	補助コ	マンド一覧??

第 1 部

基礎編



電子編集とは

電子化という言葉を聞くと、どんなイメージを想像しますか? 多少、複雑な作業を行うように感じますが、コンピュータをツールとして使うことにより、便利になったり、仕事の効率がアップするようなことだといえます。たとえば、ワープロを利用することも電子化の1つとして考えられるわけです。しかし、いざ本格的に電子化をしようと思った時に、システムの内容や作業工程などといった要素を加味して行わなければ、せっかくやった電子化もなかなか効果が現れにくいものです。

本章では、EWB (Editor's Work Bench)のお話をする前に、一般的な電子編集システムと呼ばれるものはどのようなものなのか、電子化をどう考えたらいいのかについて、その概要を解説し、基本的なポイントを抑えます。

1.1 電子化をすると

一般に、電子編集システムを使用すると、本を作る作業工程のどの部分が電子化され、どのような効果が期待できるのでしょうか。

1.1.1 どんなことができるのか?

本を作る作業工程は出版社/編集部により多少異なりますが、大きくみると編集/制作/版下作成/出力という4つの部分に分かれています。電子編集システムと呼ばれるものは、これらの工程をほぼカバーしたもので、図1-1のグレーの部分にあたります。図を見るとわかる通り、ほとんどの作業工程がコンピュータのみによってできてしまうわけです。

それでは、電子編集システムのそれぞれの機能をもう少し詳しく見ていくことにしましょう。

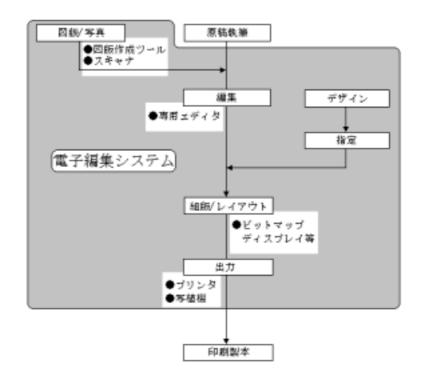


図 1-1 電子編集システムの作業工程

編集/図版作成

専用のワープロが用意されており、システムによってさまざまな特徴を持っています。システム以外のワープロやエディタで作成した文書も扱うことができます。また、専用の図形作成ツールなどが用意されているシステムもあり、簡単に図版を作成できます。

デザイン

コンピュータ上で装丁や体裁に関することが指定できます。本を作る上で大切な台紙を自由自在に作成でき、罫線などのデザインに必要な表現力も提供されています。

指定

書体、文字サイズ、行送りなどの必要な組版指定がすべてできるようになっており、すぐに確認することも可能です。しかし、書体や単位系などはシステムに依存することが多いため、注意が必要です。

組版 / レイアウト

作成した文書は、即座に組版し確認できます。組版ルールは、システムによって多種多様であり、好みにあったチューニングをすることができるものもあります。また、レイアウトは図版類

を同時に扱えるものが多く、試行錯誤しながら行う時に威力を発揮します。

出力

作成した版下は、ディスプレイやプリンタで確認できます。特に、プリンタによる出力は、校正用として扱えるため大変便利です。また、ほとんどの電子編集システムは、写植出力を意識して開発されているため、高品位の出力が得られるようになっています。

1.1.2 どんな効果があるんだろう

電子編集システムを使うことによって、従来の手法ではありえなかった効果が期待できます。 ここでは、代表的な効果をあげてみます。

コスト減と時間短縮

社内で即座にプリンタによるゲラ出力が可能になるために、初校、再校といった作業工程を省 くことができ、コンピュータ上で版下作成をダイレクトにできるため、コスト減、時間短縮につ ながります。

再利用が可能に

入稿時の版下イメージがコンピュータ上にデータとして存在するため、増刷りや再版時の作業が効率的に行えます。さらに、体裁情報が電子化されているので、シリーズ化されている出版物の作業が激減します。

意志伝達のミスがなくなる

ページ体裁や書体指定などの情報がすべて電子化されていることにより、作業中における意志 伝達のミスがなくなります。

マルチメディアへの発展性

文字情報や図形情報がすべて電子化されているため、ペーパー以外のメディアを使った新しい 出版体系に対応しやすくなります。

1.2 電子編集システムの特色は

電子編集システムの特長は、システム設計者が本自体および本作りをどのようにとらえているかによって、左右されるものです。

ここでは、本というものがどのようなものであるか、という視点で電子編集システムの考え方 を見ていきます。

1.2.1 よく本を見てみると

本を手にして眺めてみると、ページの中には、見出し、箇条書、脚注、図、写真などのさまざまな要素があることがわかります。これらの要素の関係は、見方を変えることにより次のように分けられます(図 1-2)。

- ・各要素が階層的な関係にあり、内容に従って図表類が流れていくような内容重視型のもの
- ・各要素がほとんど階層的な関係になく、デザイン優先で図表類を固定してしまうようなビジュ アル重視型のもの

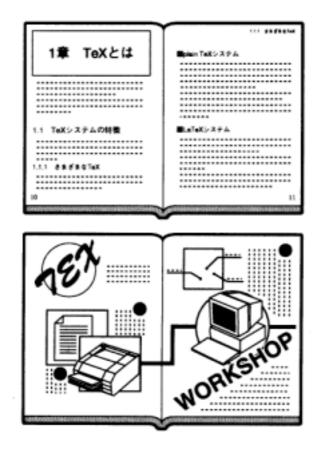


図 1-2 内容重視とデザイン重視

これらの2つの分類は、電子編集システムと密接な関係があり、システムを選択する上で重要

なポイントとなります。

1.2.2 電子編集システムの 2 つの顔

電子編集システムには、「本の各要素は階層的な関係にある」という考え方に基づいたバッチ方式と、「本はデザインにより決まる」という考え方のWYSIWYG(What You See Is What You Get)方式があります。これらの方式は、異なる特徴を持っているため目的に応じて使い分けが必要です。

わかりやすい WYSIWYG

WYSIWYG とは画面で作成した版下が、そのままプリンタや写植機から出力されるということです。ユーザーは、画面上で各要素の割り付けをすると、その結果がリアルタイムに組版やレイアウトに反映されます。また、WYSIWYG の方式は画面上でページごとに各要素を直接入力して、版下を作成していくため会話方式とも呼ばれています。よく耳にする PageMaker や QuarkXPress などの DTP ソフトは、この方式をとっています。

WYSIWYGでは、各ページごとに、思った通りのイメージをディスプレイ上に展開することができるため、頻繁にデザインが変わる雑誌などに向いています。特に、最近では、写真やカラー図版も扱えるようになり、そういった意味でもデザイン重視の出版物にお勧めです。しかし、この方式では文書中の各要素に対して、自由に体裁を指定できるというメリットがあるのですが、その反面、まとめて書体を変えるといった一括処理が苦手なので、作業に対する考慮が必要となります。

決め事に強いバッチ

バッチ方式とは、「文書は各要素が階層的な関係にある」ということを利用して一括でページを 組み上げていくという手法のことをいいます。各要素の意味付けをするために、タグと呼ばれる 印を付け(これをマークアップと呼ぶ) 各要素の体裁をあらかじめ決めておき、すべてのページ をまとめて作成してしまうわけです。

このような方式では、各要素を一括してレイアウトを行うため、要素があらかじめ決っているような定型の書籍に向いており、特に、大量ページのものに威力を発揮します。さらに、要素指定と体裁指定を別々に行い、最後のフェーズで組版をするため、作業の切り分けや同時進行が可能になります。しかし、バッチ方式は、要素をまとめて扱うため、すべてを処理しなければイメージが確認できず、不定型の出版物に対する対応が難しいという短所が存在する事実もあります。

1.3 うまい電子化をするために

出版物が多種多様なように、編集部にもそれぞれ違ったやり方があります。各編集部ごとに、 ニーズと作業工程の分析を行い、最適な手法やシステムを選択しなければなりません。

ここでは、無理なく電子化をするためには、どのような点に注意をしたらよいのかをまとめて おきましょう。

1.3.1 どんな本を作るの?

まず、電子編集システムを使って、どのような出版物を作ろうとしているのかを明確にしましょう。デザインが複雑でレイアウトがページごとに変わるような場合は WYSIWYG 方式を、要素が定型的に決っているような場合にはバッチ方式を選択するのがよいでしょう。

1.3.2 まずは作業工程の分析を

新しくシステムを導入することで、作業工程に少なからず変化が生じてしまいます。この変化が大きすぎると、編集部内で作業上の混乱がおきてしまい、効率の低下を導きだします。この混乱を避けるために、編集部の作業工程の分析を行い、システム導入後に作業がどのように変化するかをシミュレーションしてみましょう。あまり変化が大きくなるようなシステムの導入は避けるべきです。一般的に、バッチ方式は、編集者と制作者の作業の切り分けを容易に行えるため、作業工程をあまり崩さないシステム導入ができるといえます。

1.3.3 無理は禁物

編集部内に、電子編集システムを導入することにより、従来、印刷所や版下屋さんなどの外部に委託していた作業を内部で行うことになります。この作業を誰が行うかをあらかじめ決め、工程管理をしっかりと行うことが重要です。

工程の一部分のみを電子化するという部分的な運用も有効な手段といえます。たとえば、本文のみを電子化してみるというようなことも1つのやり方です。編集部の作業を全面的に電子化する前段階として、部分的な運用をしてみるものよいのではないでしょうか。

Column:電子編集システムの種類

電子編集システムを規模や価格の順に分類すると、CTS(電算写植システム) EP(電子組版システム) DTP(デスクトップパブリッシング)の3つに分けられます。

CTS は、大規模の印刷所に導入されているシステムであり、商業印刷物に要求される高い品質に対応するために、たくさんの機能を持っています。そのため、操作には訓練された専用オペレータが必要です。EP は、小規模の印刷所や企業内印刷を行っている一般企業などに導入されています。CTS ほどの機能は持っていませんが、操作面に優れており、多少の専門知識があれば、使いこなせるシステムです。DTP は、個人や部署レベルで利用可能なシステムです。操作性がきわめてよく、報告書程度の文書を体裁よく出力するぐらいであれば、一般のユーザーでも簡単に使用できます。

ここ数年で、コンピュータのコストパフォーマンスが格段に進歩し、高品位フォントなどの供給によって、EPや DTPを使用しても、CTS に劣らないような品質の出力を手に入れられるようになりました。現在、このようなシステムが出版社や印刷所にかなりの勢いで普及しています。



EWB のプロローグ

電子編集システムを編集部にいきなり導入することはなかなか難しいものです。アスキーでは、電子化が容易にできるように、作業工程を考慮に入れて開発した EWB と呼ばれるシステムを用意しました。

この章では EWB の特徴や、EWB を導入することで編集部の作業がどのように変化するかという説明をしていきます。 EWB の基本を理解して、効率的な導入を行ってください。

2.1 **現場から誕生した** EWB

EWB は、書籍やマニュアルなどの電子化のために開発されたシステムです。コンピュータの特長をうまく生かし、出版物を作成する時の便利なツールとなるように設計されたものです。

2.1.1 EWB **のコンセプトは**

EWBの開発コンセプトは、編集部がいかに無理なく電子化できるかです。編集部の声をダイレクトに聞けるという社内開発のメリットを最大限に生かして、EWBは開発されました。特に、開発時に意識したポイントは、

- ・作業工程をなるべく変更しない
- ・作業分担もできるだけ変えない
- ・作業の効率をアップさせる
- できる限りコストを下げる
- ・既存の環境を利用する
- ・必ずプロの品質に耐えられること

などがあげられます。

そして、EWB ではこれらの要求を満たすために、次のような答えを選択したのです。

バッチ方式の採用

EWB は、基本的に書籍やマニュアルの制作を目的としたシステムであるため、各要素の関係を うまく利用するバッチ方式を採用しています。この方式により、内容重視の大量文書を無理なく 作成することができ、作業工程や作業分担に対する影響を最小限に抑えられます。

EWB を UNIX 上で動かす

いろいろな要求に対して、満足に対応できるシステムはなかなかないものです。このことは、EWB にもいえることです。UNIX の世界では、自分が必要とする機能を持ったツールがない時に、既存のツールを組み合わせてその機能を実現するといった考え方があります。EWB を UNIX 上で動作させることによって、満足いかない機能を他のツールと組み合わせることで充実させることができます。また、EWB 自体は多数のツールによって構成されているため、必要に応じて一部の機能のみを使うこともできます。このように、EWB が UNIX という環境を利用することで、拡張性や柔軟性を得ることができ、いろいろな目的に対応できるわけです(図 2-1)。

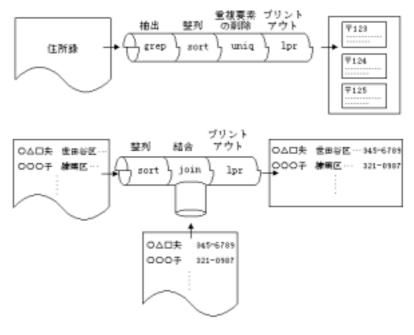


図 2-1 UNIX におけるツールの組み合せ

さらに、UNIX にはネットワーク機能があります。EWB をインストールした UNIX マシンと PostScript プリンタが 1 台あれば、ユーザーは自分のデスクトップマシンからネットワークを介し

て EWB を利用できます (図 2-2)。

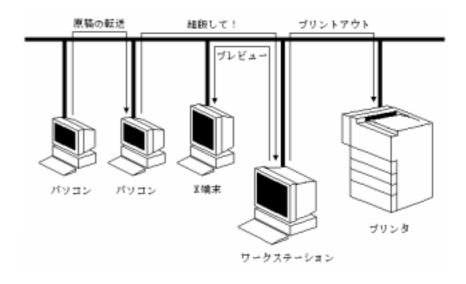


図 2-2 UNIX ネットワークと EWB

高機能な組版エンジン

電子化をすることで、出力品質が低下するようでは、たとえどんなに利用しやすいシステムでもプロの編集者には使ってもらえません。このため、EWBでは組版エンジンに TeX を採用し、高品質な組版を実現しています。

 T_{EX} というと身構えてしまう方もいるとは思いますが、基本的に EWB のユーザーは T_{EX} を意識する必要はありません。あくまでも組版エンジンとして、EWB の内部で動いているものですから御安心ください (図 2-3)。

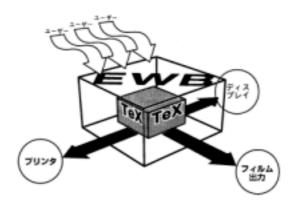


図 2-3 EWB と T_EX の関係

2.2 EWB **の特徴は何?**

こうして開発された EWB を使用することによって、どんなことができるのでしょうか。ここでは、EWB の機能面から探ってみることにします。

2.2.1 こんな機能がある

EWB では、入力した原稿を効率よく扱うことで、無理のない作業を行うことができます。

文字/組版

ワープロやエディタで入力した文字は、確実に出力に反映されます。 書体は、和文 21 書体、欧文約 200 書体を扱うことができます。

組版に関しては、縦組と横組の両者をサポートしています。文字の追い出しや追い込みなどは 段落単位で行いますので、大変きれいに仕上がります。また、英単語のハイフネーション処理も ほぼ自動的に行ってくれます。

オートナンバリング

ちょっと章の構成を変えたり、図の位置が変わったりすると、それに関連する章番号や図番号を全部付け直さなければなりません。何度も番号を付け変えていると、つい間違って後になって、あわててしまうことがあります。このため、EWBには、見出しや図表、さらには脚注の番号を組版時に自動的に付けてくれる機能があります。また、番号の数字をアルファベットにしたり、ローマ数字にしたりといったこともできます。このため、編集者は煩わしい番号管理から解放されます。

また、EWB はページアップシステムですから、柱やノンブルも自動的に振られます。柱は該当する見出しを EWB が判断し、左右ページで異なる柱を付けることができます。

オートレイアウト

EWB は自動的にレイアウトを行う機能を備えています。あらかじめ制作者が指示した規則に従って、EWB はページを作っていきます。特に、図表類の少ない出版物には、かなりの威力を発揮します。

しかし、コンピュータが規則に基づいてレイアウトするものですから、場合によっては、満足 した結果が得られません。このような場合は、手動で対処する必要があります。

社内校正/フィルム出力

EWB では、文字校正や版下校正をプリンタ出力によって、手軽に行うことができます。また、X Window System によるプレビュー機能を用いると、版下イメージが即座に見られるため、レイアウトの確認などは、より効率的に作業が行えます。

図 2-4を比較してみてください。解像度や書体は多少違いますが、組版結果や出力位置は、まったく一緒のはずです。このように EWB では、プリンタやプレビューで確認したものは、必ずフィルムの出力に反映されるのです。

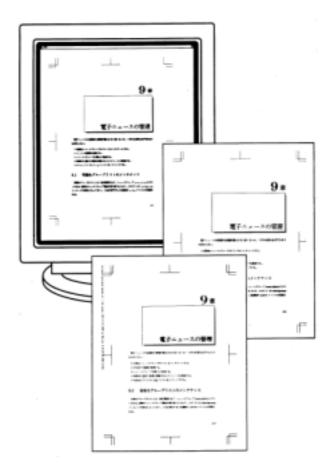


図 2-4 デバイスに依存しない出力

2.3 EWB **の作業工程を見る**

ここでは、EWB を作業工程の点から見てみることにしましょう。EWB を導入すると、作業がどのように変化するのでしょうか。

2.3.1 工程は変わるのか?

従来、編集者は制作上の指示を、紙と鉛筆を使って行っていましたが、EWBでは、それと同じことをコンピュータの操作を通して行います。このため、手作業にくらべてはるかに早く、そして正確に作業を行えます。図 2-5は、EWBの導入前と導入後の作業を表しており、グレーの部分は社内で行う作業を示しています。

この図からわかるように編集や指定などの工程は従来と変更がありません。変化があるのは、そこから先の作業です。従来の方法では、写植入稿後の初校や再校の作業がありますが、これが EWB では、プリンタ出力を使った社内校正にとって代わっています。EWB では、レイアウトは コンピュータが行うので、版下貼り込みや版下校正いう工程がなくなり、その代わりに、自動で 行われたレイアウトの修正をするという作業が発生します。

2.3.2 同時進行が可能に

今度は作業工程の変化をスケジュール面から見てみましょう。

図 2-6を見てみると従来の方法では、写植入稿とそれに伴う校正、さらに版下貼り込みと版下校正があります。EWB ではこのような工程が一切なくなり、それに伴う待ち時間も少なくなっていることがわかります。さらに、EWB では、プリンタによる出力が可能なため、編集者がこの出力を使って校正している間に、体裁の追加や変更を制作者が行うなどといった作業が並行して行えます。

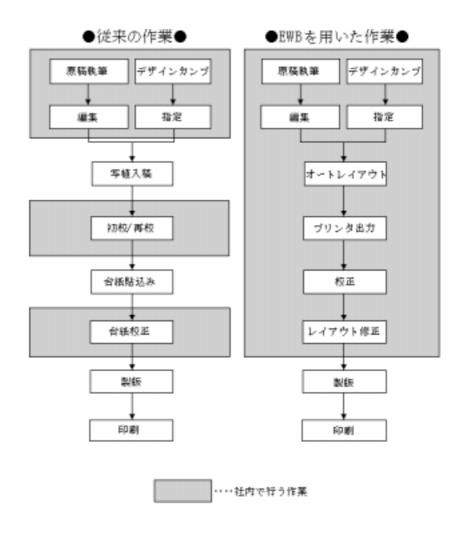


図 2-5 従来の作業と EWB を用いた作業

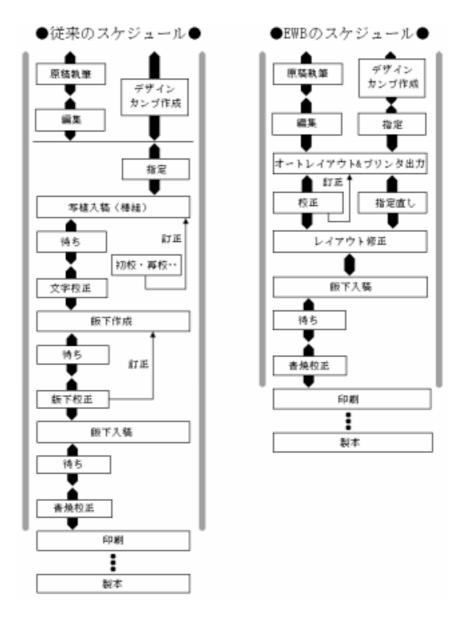


図 2-6 従来のスケジュールと EWB のスケジュール



トリガの役割り

従来の方式では、編集や組版上の指定を紙と鉛筆で行ってきました。同じことを EWB では、原稿に電子的な印を付けることで実現します。この印のことを EWB では、トリガ (trigger)と呼びます。この trigger という単語には "引き金"という意味がありますが、トリガと呼ぶ本当の意味付けは、単なる組版の命令ではなく、著者や編集者の思考の助けとなり、コミュニケーションの "引き金"になるということから由来しています。

たとえば、著者が編集者に対して、強調したい単語などに印を付け意思表示をする場合に、独 自で決めた記号では、なかなか意味を理解するのは難しいのものです。トリガの本来の目的は、 このような時に、間違いなくコミュニケーションができるように記号を規格化し、意志伝達をス ムースにできるようにと用意されたものなのです。

ここでは、トリガとは一体どのようなものなのかを明らかにし、EWB の本質を理解していただきます。

3.1 **トリガとは何か?**

コンピュータ上で、マーカーの代わりとなるトリガは、どのような決まり事があり、どのような効果が期待できるのでしょうか。

3.1.1 トリガは約束ごと

EWBのトリガは、各要素を直接マークアップできるように原稿中に入力します。そのため、どんなワープロやエディタを使っていても入力しやすいように、トリガは特殊な記号ではなく数個の記号と英数字からなる文字列で構成されています。たとえば、節見出しのトリガは、リスト 3-1のように入力します。

いずれの電子メールシステムにおいても、何をどのように行うかについては、その都度 指示が与えられる。

//ii UUCPの混雑

他のサイトにニュースを転送しているサイトの場合、隣接システムが1つダウンしてし\ ま

うと、UUCPが著しく渋滞しかねない。ダウンしているシステムが復旧しないかぎり、記事は、UUCPにスプールされ続けたまま、転送されることがないからである。

リスト 3-1 節見出しトリガの入力

リストの中にある"//"は、「これからトリガが始まります」という意味を表す記号です。これを、トリガ開始文字列と呼んでいます。さらに、それに続く"ii"がトリガの本体にあたる部分で内容を示しています。

トリガは、過去のアスキーの出版物を基準に要素を洗い出し、11 種類のトリガをあらかじめ用意してあります。表 3-1は、EWB で標準で使えるトリガの一覧です。

種別	代表的なトリガ	機能
見出し	//i	見出しを作成します
小組	//c1{ ~ //c1}	小組を作成します
箇条書	//k1{ ~ //k1}	箇条書を作成します
脚注	//ky 番号	脚注を定義し、本文中から参照します
図表	//f 番号	図表のキャプションを定義し、本文中から参照します
強調	//g{ ~ //g}	原稿中の文字列を強調します
プログラム	//list1{ ~ //list1}	トリガで囲まれた部分をプログラムリストとして出力します
表組	//table1 { ~ //table1}	タブを使って表を作成します
罫線	//kei	罫線を発生します
キートップ	//key1{ ~ //key1}	トリガで囲まれた文字を罫線で囲みます
囲み罫	//w1{ ~//w1}	トリガで囲まれた文章を罫線で囲みます
ルビ	//rb1{ ~ //rb1}	文字列にルビを振ります
傍点	//bou1{ ~ //bou1}	文字列に傍点を振ります
索引	//in 番号	索引に入れたい項目に印を付けます

表 3-1 標準トリガー覧表

この表を見てみると、トリガの種類が少ないと感じるかもしれません。しかし、実際に出版されている大半の書籍において、これらのトリガでも多いぐらいなのです。見出しや強調、さらに図表など数種類のトリガで作成されてしまった例もあります。

トリガ開始文字列やトリガの本体は、状況に合わせて自由に変更することができます。また、標準トリガで対応できない場合は、トリガの種類を増やす仕掛けも用意してあります。しかし、あまり無計画にトリガを改変したりすると、可読性が低下し、混乱の元になりがちなので注意が必要です。

一般的に、トリガの種類をいたずらに増やすよりも、本の各要素をなるべく絞って表現した方が、読みやすいものになるといえるかもしれません。どうしてもトリガの種類が増えてしまうような場合は、「EWB 向きではない」と判断した方がよいでしょう。

3.1.2 トリガの入った原稿を覗いてみる

原稿にトリガを入力していくことを"EWB コーディング"と呼びます。ここでは、EWB コーディングした原稿を使って、トリガ入力の方法を簡単に説明していきます。まずは、トリガ入力というものがどのようなものなのかということを覚えてください。EWB のコーディング例を眺めてみると、トリガ入力が意外に簡単で、原稿の構成がはっきりと見えてくることがわかります(リスト 3-2)。

//i 接続の手順

UUCPネットワークの土台となるのが、物理的な通信リンクである。リンクを開設しないかぎり、何もできない。

通信リンクには、つぎの3種類がある。

//k1{

ダイレクトリンク(RS-232による直結)

モデムリンク(電話)

ローカルエリアネットワーク

//k1}

ダイレクトリンクとモデムリンクの場合には、自分のシステムのシリアル回線(RS-232)を通信媒体として使うことになる。ダイレクトリンクではシステムとシステムとを直接ケーブルで接続し、モデムリンクではケーブルをモデムに接続する。

ローカルエリアネットワーク(LAN)には数多くの種類があり、それぞれ設定条件が異なる。ローカルエリアネットワークの物理的な設定方法は、本書では扱わない(既存のLANを利用するときのUUCPの設定方法については、次章で述べる)。したがって、詳細についてはシステムに添付されているマニュアルを参照してほしい。

//ii シリアルケーブルの作成

Electronic Industries Association (EIA;米国電子工業会)によるRS-232C規格(ふつうは単にRS-232と呼ぶ)では、コンピュータや端末とモデムを接続するシリアルケーブルについて規定している。

RS-232規格で規定されている信号の多くは、ほとんど使用されていない。本書において重要な信号を//f1000に示す。

//f1000 RS-232の信号とその機能

RS-232ケーブルは最大25線で構成される。各線が固有の機能をもち、別々の信号を搬送

リスト 3-2 EWB コーディングした原稿

見出しのトリガ

見出しには、" //i " と " //ii " というトリガが付いています。このトリガは、それぞれ、章見出しと節見出しを表します。" 第 1 章 " や " 1.1 " などといった文字列は、EWB が自動的に付けてくれますので、入力する必要はありません。

図表のトリガ

" //f1000 " というトリガが、2 箇所に入力されています。これは、図のトリガを表しています。最初に入力されているトリガが図の参照を、次のトリガが図のキャプションを意味しています。" //f " の後にある番号がその図の定義と参照関係を表しています。この番号は順番に並んでいる必要はありません。原稿の中で他の番号と重なっていなければよいのです。表やリストなどといったトリガの入力も同様です。

箇条書

"/k1"は箇条書のトリガを表しています。箇条書や小組のように範囲を指定するものは、トリガの後に中括弧を付けて囲みます。

3.1.3 EWB システムによる出力

図 3-1は、リスト 3-2の原稿を EWB システムによって、出力したものです。見出しや図表の番号が正確に付いており、きれいに組版されていることがわかります。

では、なぜ、EWB コーディングされた原稿があるとこのようなきれいな出力が得られるのでしょうか? これは、EWB システムにちょっとした仕掛けがしてあるからです。概念的に表すと図 3-2のような流れになります。

この図にあるように、EWBシステムでは、EWBコーディングされた原稿ファイル以外に"トリガ定義ファイル"と"スタイルファイル"と呼ばれるものを必要とします。このトリガ定義ファイルとは、トリガの意味を EWBシステムに伝える役目を行い、スタイルファイルは組版の指示を行うものなのです。EWBシステムでは、これらの3つのファイルを読み込んでトリガを解釈し、組版して出力します。これらのファイルについては、第2部で詳しく説明しますので、ここでは、このような仕掛けになっているのだということを覚えておいてください。

EWB システムは、このような仕掛けになっているため、原稿には手を入れずに、出力結果を変えることができます。たとえば、極端な例になりますが、横組の書籍を縦組に変更するなどと

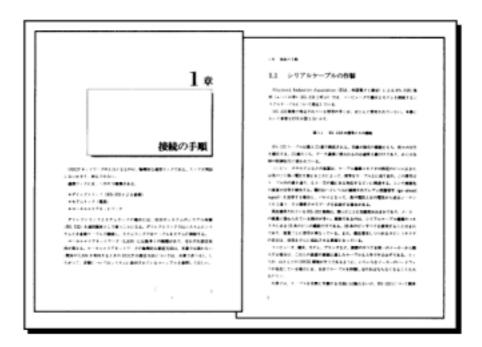


図 3-1 EWB システムによる出力

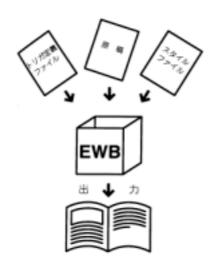


図 3-2 EWB システムの概念図

いった変更も簡単に対応できます。それでは、実際に、トリガ定義ファイルとスタイルファイルを他のものに入れ換えて、出力してみましょう(図3-3)。

このように、同じ原稿を出力したにも関わらず、図 3-1とは全く違ったイメージになります。この 2 つの出力を比較することから、トリガとは、組版命令とは異なったものであることがわかります。あえて、組版との関係を考えてみると、「ここから組み方が変わります」という印としての

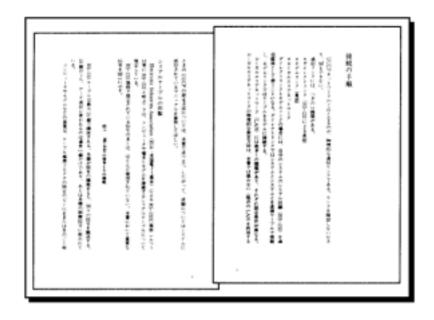


図 3-3 スタイルファイルを変更して出力

意味しかありません。EWBシステムにとって、どのような組版をするかということは、トリガ定義ファイルとスタイルファイルの2つで決まるわけです。

3.2 トリガ入力の真の意味は

前節で述べたように、トリガは組版命令ではありません。では、トリガ入力は何のために行う のでしょうか。また、誰がトリガを入力していくものなのでしょうか。ここでは、トリガという ものを、もう少し掘り下げてみましょう。

3.2.1 本を作る作業とは

本を作る作業を大きく分けるとしたら、編集と制作という2つの作業からなります。編集は、 企画意図に基づいて、全体の構成を考えて、原稿を依頼したり、原稿の内容を煮詰めたりします。 一方、制作は、その内容をより効果的に伝えるために、書体や装丁を工夫します。

ここで、本の中の要素、たとえば、節見出しを作る場合を考えてみることにします。編集者は、「この一文は節のレベルの見出しである」というような意味上の論理的な処理を行います。また、制作者は「節見出しには、新ゴBの20級1歯詰め」のような実体に結びつける物理的な処理を行います。これらの2つの処理があって、はじめてページの中に、節見出しが出来上がるわけです。このように、本を作る作業を"論理"と"物理"というように分けて考えることにより、編

集者と制作者の作業分担が、自然に見えてきます。

従来の編集の作業においては、マーカーと紙を使って論理的な処理を行っていましたが、EWBではこの処理をトリガを使って行います。トリガを使用することで、コンピュータに編集者の意志が通じるわけです。また、制作者は、このトリガに関する体裁をコンピュータ上のスタイルファイルというものを利用して表現します。このように、EWBでは、トリガとスタイルファイルと呼ばれるものを使うことで、作業の切り分けを行うことができるのです。

3.2.2 文書の論理構造

さらに、一歩考えを進めて、本全体を眺めてみましょう。ページの中にある見出しや箇条書などといった要素は、ランダムに並んでるわけではありません。章の中に節があり、節の中に本文や図があるといったように、論理的な入れ子(あるいは階層)構造になっています(図 3-4)。このような構造を、一般的に、文書の論理構造と呼びます。また、見出しは本文より大きな文字で組まれているとか、脚注はページの下に位置しているなどといった見た目の構造のことを物理構造と呼びます(図 3-5)。

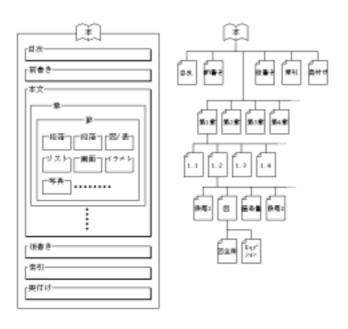


図 3-4 文書の論理構造

文書の論理構造という言葉を使えば、編集者の仕事は論理構造を記述することであり、制作者の仕事は物理構造の記述であるといえます。EWBでは、この論理構造を、トリガを使って記述することになるわけです(図 3-6)。

「トリガによる文書の論理構造の記述」。 これが EWB のトリガの真髄なのです。



図 3-5 文書の物理構造

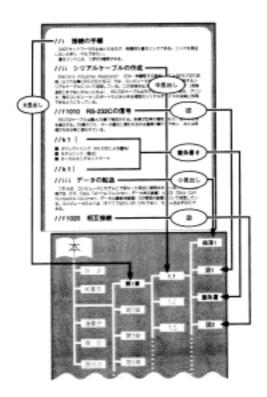


図 3-6 トリガを使った文書構造記述

Column:編集トリガと組版トリガ

EWB では、見出しや箇条書などの要素をマークアップするトリガを編集トリガと呼び、強制改ページやアキ調整などのレイアウトに関係するトリガを組版トリガと呼んでいます。

これらのトリガは、作業工程上において、同時に入力されることはありません。 執筆や編集作業時には編集トリガが使用され、レイアウト作業時には組版トリガが入力されます。 もし、同時に使われる場面が現われてしまったら、各作業工程の流れを確認してみましょう。

3.3 トリガ入力とコミュニケーション

いろいろと面倒な話をしてきましたが、EWB を使って仕事を進めるためには、トリガ入力が不可欠です。「EWB は、トリガなんてものを入力しなきゃいけないなんて面倒だ!」と思うかもしれません。しかし、数種類の小さい文字列を入力するだけですから、慣れれば本一冊、数時間で済んでしまうのです。

ここでは、より積極的に EWB を利用していただくために、トリガはコミュニケーションをスムースにする触媒の役目があるというお話をしていきます。

3.3.1 筆者と編集者

原稿執筆の段階から筆者にトリガを入力してもらうのもよい方法です。編集者のトリガ入力の 負担も軽くなり、トリガによって筆者の意図が間違いなく伝わります。特に、標準トリガは効果 抜群です。さらに、原稿全体の構成が見やすくなるというメリットもあります。

3.3.2 編集者と編集者

編集者と編集者間のコミュニケーションでも、トリガは役に立ちます。たとえば、複数の編集者が一冊の本を担当しているなどといった場合です。トリガの入力規則を決めておけば、編集者間で原稿をやり取りしながら、容易に編集作業を進めることができます。

3.3.3 編集者と制作者

一冊の本を作るためには、編集者と制作者間の打ち合せが頻繁に行われます。EWB では、トリガをキーワードにすることで、打ち合せをスムースに行うことができます。たとえば、従来の方法では、箇条書がたくさんあった場合に、「この部分の箇条書の問答はどうします?」といったように一つ一つ確認しながら、打ち合せをしなければなりませんでした。しかし、EWB では、「箇条書"//k2"の問答は?」というように、トリガによって表される要素ごとの打ち合せだけで済みます。



EWB **の表現力**

前章では、EWBをトリガという視点から解説してみました。トリガの意味付けや指定できる要素などといったことを理解していただけたのではないでしょうか。しかし、実際に本を作成していく上で、トリガ指定された要素が、どのような表現力を持っているかを知っていなければ、作業がなかなか進んでいきません。

そこで、この章では、トリガで指定された各要素が紙面上でどのように表現されていくのかということを、具体的な図を示しながら解説していきます。ここで、述べる事柄は、基本的に EWB が標準でサポートしているものに限っています。もし、これ以上の表現力を必要とする場合は、トリガなどを拡張しなければならないので、注意が必要です。

4.1 一般知識

4.1.1 体裁入力ファイル

これまで説明したように、EWBでは編集者が原稿にトリガを入力することで、見出しや箇条書という要素を特定していきます。そして、制作者は、このトリガによって表現される各要素の体裁を指定します。EWBでは、この体裁を各要素ごとに"体裁入力ファイル"と呼ばれるテキストファイルに記述します(リスト 4-1)。体裁入力ファイルは、基本的に、本ごとに作成するものです。シリーズ化された出版物などの場合には、すでに作成されているファイルを流用することで、効率アップできます。

また、このファイルには、各要素の体裁以外に、判型や版面の大きさ、柱やノンブルの位置な どといった指定も行います。

```
# 体裁入力ファイル
その他
  書名
                "新規の書籍"
                横 # 指定項目:縦/横
  組方向
                1
  ページスタイル
                         # デフォルトのページ体裁番号
  トリガ定義ファイル "Styles/triger.def"
  スタイルファイル "Styles/ewb2latex.cls"
  PDFスタイルファイル "Styles/PDFstyle"
その他終り
頁サイズ
  仕上がりサイズ縦 257mm #単位:mm/歯/pt
仕上がりサイズ横 182mm #単位:mm/歯/pt
トップマージン 24.75mm #単位:mm/歯/pt
のど側のマージン 23mm #単位:mm/歯/pt
頁サイズ終り
禁則
                       # 指定項目:有り/無し
               無し
  禁則解除
  和欧文間スペース 有り
                        # 指定項目:有り/無し
禁則終り
本文タブ
                        # 半角字数 タブ幅を指定
  スペース
本文タブ終り
基本組
                       # 和文書体
  書体
             Ryumin-Light
                         # 指定項目:
  変形
              正体
                         # 正体/かな/
                         # 長体1/長体2/長体3/長体4/
                         # 平体1/平体2/平体3/平体4/
                         # 斜体1/斜体2/斜体3
            130
  級数
                         # 単位: Q/pt
             13歯
                         # 単位:歯/pt/mm
  字送り
            GothicBBB-Medium # 和文書体
  強調書体
  強調変形
             正体
                        # 指定項目:
                         # 正体/かな/
                         # 長体1/長体2/長体3/長体4/
                         # 平体1/平体2/平体3/平体4/
                        # 斜体1/斜体2/斜体3
                        # 単位:Q/pt
  強調級数
            130
             13歯 # 単位:歯/pt/mm
  強調字送り
            PCTimes-Roman # 欧文書体
  欧文書体
  欧文級数
                        # 単位:Q/pt
             13Q
  欧文ベースライン 0.03字
  欧文強調書体 PCHelvetica # 欧文書体
```

リスト 4-1 体裁入力ファイル

4.1.2 基本的な体裁指定

各要素に共通な書体や級数などの文字関連に対して、どのような指定ができるのかを見てみましょう(図 4-1)。

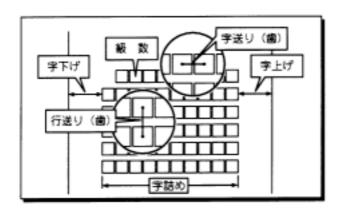


図 4-1 基本的な体裁指定

書体

現在、モリサワやアドビの PostScript 書体が和文 21 書体、欧文 200 書体以上が同時に使用でき、PostScript 形式の書体であればさらに追加することができます。この書体指定には PostScript フォント名 (Ryumin-Light や Times-Roman)を使います。さらに、PostScript 書体以外に TEX の標準的な書体も同時に使用できます。EWB で使用可能な書体に関しては、Appendix D の 書体一覧表を参照してください。

変形

和文書体に対しては、正体、長体、平体、斜体、かなつめを指定できます。変形の範囲は長体 1 (10%) ~ 長体 4 (40%)、平体 1 (10%) ~ 平体 4 (40%)、斜体 1 (10度) ~ 斜体 3 (30度)、かな (かなつめ)です。欧文の変形については標準ではサポートされません。

級数(文字サイズ)

文字サイズを級(Q)またはポイント(pt)で指定します。

字送り / 行送り

字送りや行送りは、歯(H) ポイント(pt) ミリ(mm) で指定します。小数点で指定することもできます。

字詰め/字下げ/字上げ

字詰め(1行の文字数)を字数で指定します。

また、字下げや字上げを指定する場合は、版面の両端からの値(絶対値)を字数/歯/ポイント/ミリで入力します。

ベースライン補正

EWB で使用する文字は、それぞれ異なる高さ、深さ、幅を持っており(図 4-2)、ベースライン(参照点を含む水平線)に従って組むと、文字がきれいに並んで見えるようになっています。一般的な和文書体とそれに伴う従属欧文との組み合せの場合は、きれいに並んで見えますが、字形の違った書体の組み合せを行うと並び方が不揃いになります。このような場合には、欧文のベースラインの位置を垂直方向(縦組の場合は水平方向)にずらしてきれいに見えるように補正をします(図 4-3)。この補正値は、全角文字の高さを 1 として "0~1"の小数、または歯 / ポイント / ミリで値を設定します。この値は、大きくすると欧文文字が下にずれます(縦組では左にずれます)。

文字色

文字色の指定です。シアン、マゼンタ、イエロー、ブラックの各色成分について 0.0 ~ 1.0 の数値で指定します。本文基本文字の色は指定できません(常に黒です)。

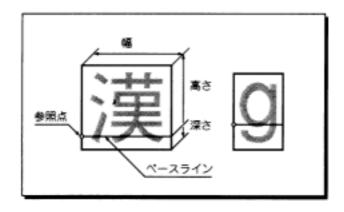


図 4-2 文字ボックス

4.2 文字

EWB では、基本的に、パソコンなどで入力できるような JIS の第 1 水準および第 2 水準の漢字を扱うことができます。ただし、半角カナ文字は使用することができません。

4.2.1 半角/全角

全角で入力した文字は、すべて全角の送りになり、半角で入力した数字やアルファベットは欧文送り(プロポーショナルピッチ)になります(図 4-4)。このため、数字や欧文は半角で入力した

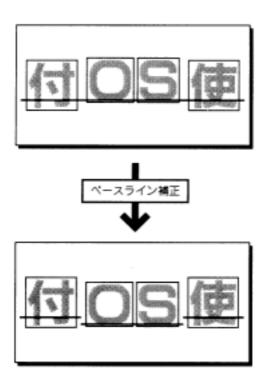


図 4-3 和文と欧文のベースライン補正

方がきれいに出力されるのでお勧めです。また、縦組の場合には、半角で入力したアルファベットなどの文字は横転して出力されます。

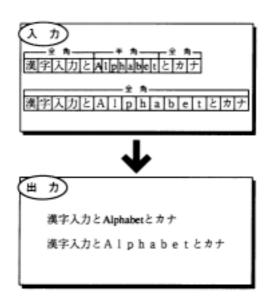


図 4-4 半角 / 全角入力

4.2.2 禁則

行頭および行末禁則

EWB では、通常の組版処理に従い、行頭および行末禁則処理を自動的に行います。この時の禁則対象文字は、括弧類や句読点、そして拗促音などがあげられます。約物類が多重になった場合も、適切な処理を行うように考慮されています。

また、EWBでは、字詰めが狭い場合にも対処できるように、拗促音(ア、っなど)や音引き(一)を禁則対象から外すことができるようになっています。

段落に対する禁則

EWB は、段落を組む時に、最後の1行が1文字だけにならないように処理を行います。もし、このようなことが起こりそうな場合には、1文字をどうにかして2文字にしたり、その1文字を前の行に追い込んだりして処理します。

4.2.3 和欧文間四分アキ

和文と欧文の間に、自動的に四分アキを入れるかあるいは入れないかを指定できるようになっています(図 4-5)。

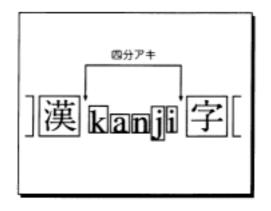


図 4-5 和欧文間四分アキ

4.2.4 約物や英単語などに対するスペーシング

"("や"、"などの約物は、通常全角の幅を持っています。しかし、約物が行頭や行末などに遭遇した場合には、禁則処理が行われて、約物の幅が半角から全角の間で変動します。また、英単語の追い出しや追い込みに伴う行中のスペース調整は、段落単位で行われるため、通常の組版処

理の結果よりも優れたものが得られます(図4-7)。

4.2.5 ハイフネーション

EWB では、英単語が行末に遭遇した場合、可能な限り追い込みや追い出しを試みます。しかし、この処理が成功しなかった時には、その単語をハイフネーションの規則に従って、2 つに分けることを行います(行末にハイフンは付かない:図 4-6)。ただし、特殊な単語(技術用語など)は、この規則には適用できず行末からはみ出してしまうことがあります。

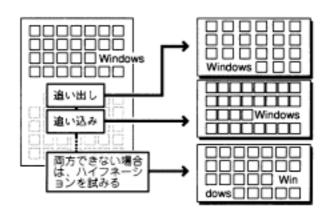


図 4-6 段落単位のスペーシングとハイフネーション

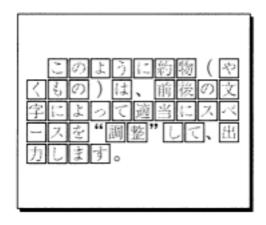


図 4-7 約物の組版

4.3 判型/基本組

EWBでは、本の基本的な体裁を判型と基本組により決めます。また、版面はこの基本組、すなわち本文の字詰めや行数などの指定で決定します。

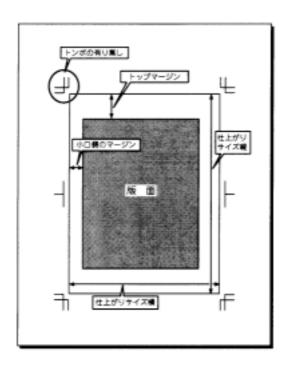


図 4-8 判型

4.3.1 判型

ミリ/歯/ポイント単位で、自由なサイズを指定できます(図4-8)。

マージン

仕上がり線から版面までの上下左右のアキを表します。"地"のアキと"のど"のアキは、"トップマージン"と"小口側のマージン"という2つの指定項目により自動的に計算されます。

4.3.2 基本組

本文の字上げや字下げもサポートしています。

強調

本文で使用する書体以外に、強調書体の指定もできます。強調したい単語を、" //g{ " と " //} " で囲みます。

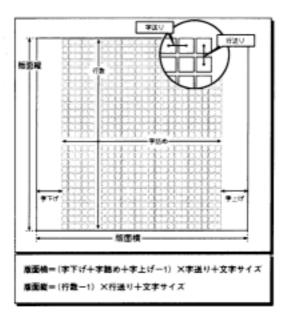




図 4-9 判型と基本組の出力

4.4 ページスタイル

左右ページごとに、柱とノンブルの付き方や出力位置などを指定できます(図 4-10)。さらに、本文用や章扉用などといった複数のページスタイルを用意することができます。

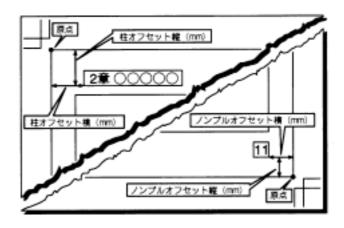


図 4-10 ページスタイル

4.4.1 補足説明

原点

出力位置の基準点を、ページの"右上、右下、左上、左下"の4点から選べます。 オフセット横/オフセット縦

原点からの左右、上下の位置をミリ/歯/ポイント単位で表します(図 4-11)。 シンボル

柱やノンブルは、自動的に発生させることができ、いろいろな表現で出力することが可能です(図 4-12)。

修飾

柱やノンブルを罫線や EPS 画像で飾ることができます。各修飾要素の縦横サイズと原点からのオフセットを指定します。

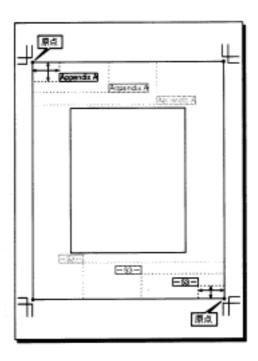


図 4-11 柱とノンブルの位置指定

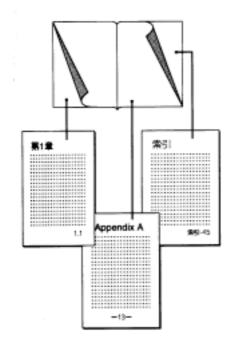


図 4-12 柱とノンブルのシンボル

4.5 見出し

EWB では、複数の見出しレベルをサポートしています。各見出しは出現順にカウントされ、番号付けを自動的に行います(第1章、1.2.1 など)。各レベルの見出しに対して上位の見出しが設定でき、上位見出しの出現に伴って見出し番号や図表番号が自動的に初期化される仕掛けになっています。

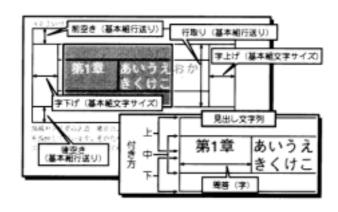


図 4-13 見出しの体裁

4.5.1 入力方法

見出しトリガ"//i"などの後にスペースを入れ、見出しの内容を入力します(リスト 4-2)。この時、"第1章"などと入力する必要はなく、見出しの内容のみを入れます。また、見出しは1行で書くようにしてください。

EWBでは見出しの前後にある改行はすべて無視するようになっており、出力時には反映されません。このため、前後のアキなどを意識せずに入力を行うことができます。さらに、見出しの前後に空行を入れられるため、可読性が高められます。

そこで、この章では、トリガで指定された各要素が紙面上でどのように表現されていくのかということを、具体的な図を示しながら解説していきます。ここで、述べる事柄は、基本的にEWBが標準でサポートしているものに限っています。

//iii 体裁入力ファイル

これまで説明したように、EWBでは編集者が原稿にトリガを入力することで、見出しや 箇条書という要素を特定していきます。そして、制作者は、このトリガによって表現され る各要素の体裁を指定します。

リスト 4-2 見出しの入力

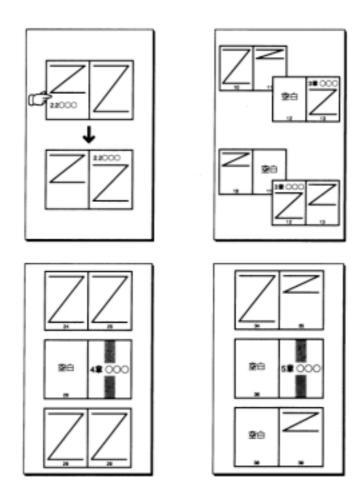


図 4-14 見出しによるページ体裁の変更

4.5.2 補足説明

前空き / 行取り / 後空き

見出しの前後のアキは、"前空き、行取り、後空き"を上手に組み合わせて指定します。この中の"前空き"は、見出しがページの先頭に位置すると消えてしまいます(図4-13)。一方、"行取り"を使って指定されたアキは、どのような場合でも、必ず一定のアキが取られます。ページ体裁

各見出しに基づいて、柱や章扉などといったページ体裁を変更することができます(図 4-14)。 見出し禁則

見出しを出力する時に見出しがページ末にきて、本文が1行も入らなくなってしまった場合には、次のページに移動します(図 4-15)。

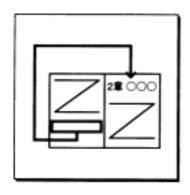


図 4-15 見出し禁則

EPS ファイル

見出しに EPS 画像を使用することができます。使用する EPS ファイル中に" @n@ "や" @str@ " といった文字列を入れておけば、その部分がそれぞれ見出し番号、見出し文字列に置き換わって表示されます (\boxtimes 4–16)。

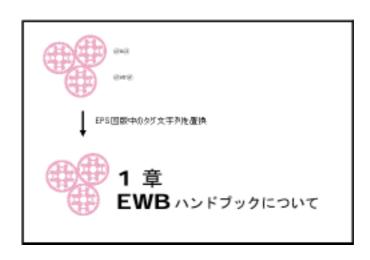


図 4-16 EPS 画像を使用した見出し

4.5.3 用語説明

問答

見出しが複数行になってしまい、2 行目以降のみを字下げしたい場合に用いるものです (図 4-17)。この時の指定は、全角文字の文字数/歯/ポイント/ミリで行います。行揃えが"左"のときのみ有効です。

問答あり 第1章 Windowsプログ ラミングの基礎

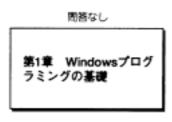


図 4-17 問答の利用

シンボル

見出しの番号付けに使われる文字は、"アラビア数字(算用数字)、ローマ数字(大文字/小文字)、アルファベット(大文字/小文字)、の中から選ぶことができます(図 4-18)。

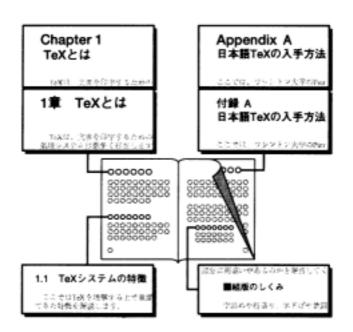


図 4-18 見出しのシンボル

4.5.4 目次

EWB の目次は、見出しから抽出した情報を集めて、指定した体裁で組まれます(図 4-19)。

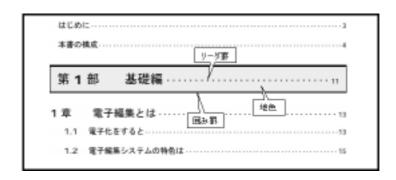


図 4-19 目次の体裁

リーダ罫

見出し文字列と見出しページ番号の間の体裁を指定できます。

囲み罫

目次項目を罫線で囲むことができます。

背景色

罫線の内側の色を指定できます。

4.6 小組

コラムやリードなどといった本文とは異なる組み方をする場合に使います。小組の種類は必要 に応じて増やすことができます。

4.6.1 入力方法

小組トリガの入力は、範囲指定で行います。小組にしたい段落の前に開きのトリガ"//c1{"を、小組にしたい最後の段落の後に閉じのトリガ"//c1}"を入力してください(リスト 4-3)。この時、閉じのトリガには、"//}"という省略形を使用することもできます。なお、小組の中に見出しを含むことはできませんので注意が必要です。

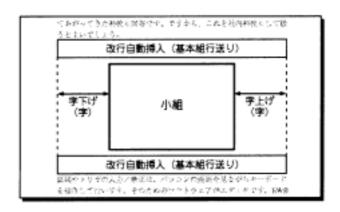


図 4-20 小組の体裁

文書の論理構造という言葉を使えば、編集者の仕事は論理構造を記述することであり、制作者の仕事は物理構造の記述であるといえます。EWBでは、この論理構造を、トリガを使って記述することになるわけです。

//c2{

EWBでは、見出しや箇条書などの要素をマークアップするトリガを編集トリガと呼び、 強制改ページやアキ調整などのレイアウトに関係するトリガを組版トリガと呼んでいます。 これらのトリガは、作業工程上において、同時に入力されることはありません。執筆や 編集作業時には編集トリガが使用され、レイアウト作業時には組版トリガが入力されます。 もし、同時に使われる場面が現われてしまったら、各作業工程の流れを確認してみまし よう。 //}

いろいろと面倒な話をしてきましたが、EWBを使って仕事を進めるためには、トリガ入力が不可欠です。「EWBは、トリガなんてものを入力しなきゃいけないなんて面倒だ!」と思うかもしれません。しかし、数種類の小さい文字列を入力するだけですから、慣れれば本一冊、数時間で済んでしまうのです。

リスト 4-3 小組の入力

4.6.2 用語説明

改行挿入

EWB では、小組の前後に、統一的な決まったアキを入れることを改行挿入と呼んでいます (図 4-20)。 改行挿入を指定すると、小組の前後に入力された改行が無視され、出力時に体裁 入力ファイルで指定されたアキが反映されます。また、改行挿入が指定されていない場合に は、小組の前後に入力された改行は、そのまま出力に反映されます。

囲み罫

全体を罫線で囲んだ小組を作ることが出来ます(図4-21)。 定義されている囲み罫のトリガ名を指定して使用します。

文書の論理構造という言葉を使えば、編集者の仕事は論理構造を記述することであり、制作者の仕事は物理構造の記述であるといえます。EWBでは、この論理構造を、トリガを使って記述することになるわけです。

EWB では、見出しや箇条書などの要素をマークアップするトリガを編集トリガと呼び、強制改ページやアキ調整などのレイアウトに関係するトリガを組版トリガと呼んでいます。

これらのトリガは、作業工程上において、同時に入力されることはありません。執筆や編集作業時には編集トリガが使用され、レイアウト作業時には組版トリガが入力されます。もし、同時に使われる場面が現われてしまったら、各作業工程の流れを確認してみましょう。

いろいろと面倒な話をしてきましたが、EWB を使って仕事を進めるためには、トリガ入力が不可欠です。「EWB は、トリガなんてものを入力しなきゃいけないなんて面倒だ!」と思うかもしれません。しかし、数種類の小さい文字列を入力するだけですから、慣れれば本一冊、数時間で済んでしまうのです。

図 4-21 罫線で囲んだ小組の出力

4.7 表組

表の項目が単純に並んでいるような体裁の表を作成することができます(各項目が連結しているような表は対象外)。表組の種類は必要に応じて増やすことができます。

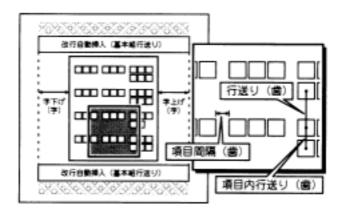


図 4-22 表組の体裁

4.7.1 入力方法

表組トリガの入力は、範囲指定で行います。表組にしたい部分の前に開きのトリガ"//table1{"を、表組にしたい部分の後に閉じのトリガ"//table1}"を入力してください。この時、閉じのトリガには、"//}"という省略形を使用することもできます(リスト 4-4、図 4-23)。

表の項目は、タブを利用して区切っていきます。通常、各項目は先頭の行に入力されたタブの 位置を基準にして、それに続く行はすべてその位置に揃います。

さらに、便利な使い方として、表組トリガのオプション指定というものが用意してあります (リスト 4-5、図 4-24)。これを利用すると、各項目の幅を数値で指定したり、項目の揃え(左/中/右)を指定したりすることができます。この場合、タブの役割は単純に各項目の区切り記号の意味しか持ちません。

```
//table1{
//g{デバイス
          信号
               意味//g}
コンピュータ
               別のシステムを呼び出します。準備はいいですか?
          DTR+
モデム
               準備できました。発信してください。
          DSR+
               リモートシステムと接続しました。
モデム
          DCD+
コンピュータ
         RTS+
              データを送ってもいいですか?
               結構です。どうぞ。
モデム
         CTS+
          TxD... データを送信します。
...RxD データを受信しました
コンピュータ
モデム
モデム
               ちょっと待ってください!
          CTS-
モデム
          CTS+ 大丈夫です。再開してください。
//table1}
```

リスト 4-4 表組の入力

```
//table1[1(8w)1(8w)c][1(8w)1(8w)1]{
//g{デバイス
          信号
                意味//g}
コンピュータ
                別のシステムを呼び出します。準備はいいですか?
          DTR+
          DSR+
                準備できました。発信してください。
モデム
               リモートシステムと接続しました。
モデム
          DCD+
コンピュータ
              データを送ってもいいですか?
          RTS+
モデム
          CTS+
               結構です。どうぞ。
          TxD... データを送信します。
...RxD データを受信しました
コンピュータ
モデム
モデム
          CTS-
                ちょっと待ってください!
モデム
          CTS+
                大丈夫です。再開してください。
//table1}
```

リスト 4-5 表組の入力 (オプション指定)

```
デバイス 信号
          意味
コンピュー DTR+ 別のシステムを呼び出します。準備はいいですか?
タ
モデム
     DSR+ 準備できました。発信してください。
モデム
     DCD+ リモートシステムと接続しました。
コンピュー RTS+ データを送ってもいいですか?
タ
モデム CTS+ 結構です。どうぞ。
コンピュー TxD... データを送信します。
モデム
     ...RxD データを受信しました。
モデム
     CTS- ちょっと待ってください!
モデム
     CTS+ 大丈夫です。再開してください。
```

図 4-23 表組の出力

デバイス	信号	意味
コンピュータ	DTR+	別のシステムを呼び出します。準備はいいですか?
モデム	DSR+	準備できました。発信してください。
モデム	DCD+	リモートシステムと接続しました。
コンピュータ	RTS+	データを送ってもいいですか?
モデム	CTS+	結構です。どうぞ。
コンピュータ	TxD	データを送信します。
モデム	RxD	データを受信しました。
モデム	CTS-	ちょっと待ってください!
モデム	CTS+	大丈夫です。再開してください。

図 4-24 表組の出力(オプション指定)

4.7.2 補足説明

タブ

表組トリガでは、基本的に、タブを利用して表の項目を区切っていきます(図 4-25)。項目の幅指定をしない場合には、指定されたタブ(ハードタブ)の値を使って項目を揃えていきます。この時、タブの値の指定は、表組の文字サイズ(正確には字送り)の倍数となります。項目内行送り

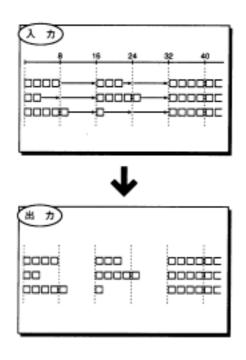


図 4-25 タブによる表作成

1 つの項目が複数行になってしまった時に使用される行送りのことを示します(図 4-22)。項目間隔

隣合う項目間のアキを指定するものです(図 4-22)。

表組の改ページ

表組トリガで囲まれた部分が途中でページ末に遭遇した場合に、EWB は改ページ可能な位置を探し、表を分割します(図 4-26)。このとき、改ページが可能な位置は各項目の境目で、項目の途中では改ページすることはできません。

囲み罫

小組と同様に、囲み罫を使用した表を作ることが出来ます(図4-27)。

4.8 脚注

脚注トリガには、参照部(本文中にある上付き文字に相当)と定義部(ページ末にある脚注の内容に相当)があります。この両者の関係を EWB システムに伝えるため、脚注トリガに続いて同じ番号を入力します。ここで入力する番号は原稿整理上で使用するものであり、実際の脚注番号とは無関係なものです。原稿中で、この番号が重ならなければ。どんな順序であっても構いま

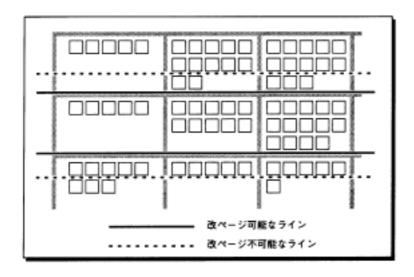


図 4-26 表組の改ページ



図 4-27 罫線を使った表組

せん。EWB では、この番号を使って参照と定義の関係を解決しながら、自動的に脚注の番号付けをして出力してくれます(脚注の番号付けは見出しと同様に自由)。

4.8.1 入力方法

参照部には、実際に脚注のマーク(上付き文字)が付く場所に、" //ky 数字 " を入力します。また、定義部は開きのトリガ " //ky { " と閉じのトリガ " //ky }、または// } " の中に脚注の内容を記述します。脚注の内容は、" //ky 数字 " というトリガの後にスペースを入れて入力します(リスト 4-6)。

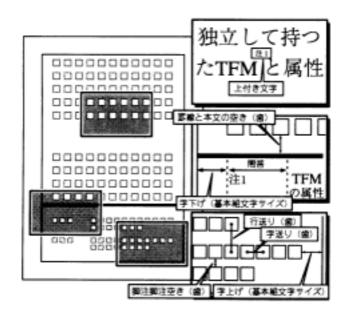


図 4-28 脚注の体裁

このトリガは、複数行になっても問題なくページ末に出力されます(図 4-29)。

ネットに参加するためのもう1つの方法は、UUNETに登録することである。もともとはUU NETは、UNIXの技術者グループであるUsenixによって出資された、UUCPとUsenetの非営利的商用フックアップ//ky100である。UUNETに登録すれば、CompuServeとTymnetの公衆データ網を通して、ネットにアクセスできる。これらのデータ網には、大半の大都市にローカルアクセス電話番号がある。ピーク時以外(夜間と週末)のTymnetの接続料は、1時間5ドルとけっこう妥当な値段である。CompuServeでは、時間帯を問わず1時間5ドルである。UUNETでは、全ニュースを流すための月間経費見積もりを250ドルとしており、これに月額35ドルのUUNETメンパー料金が加算される。UUNETの長所は、パップボーンサイトから1ホップで接続されることである。また、無料で再配布が可能なUNIXソースアーカイブも、数多く提供されている。

//ky{
//ky100 訳注 Usenetは、管理、運営の母体となる組織をもたないので、UUNETという非営利の組織が、有償でUsenetへの接続サービスを行っている。

//ky}

リスト 4-6 脚注の入力

4.8.2 補足説明

罫線本文空き

ページ末に現れる脚注と本文との最小限のアキを指定する値です(図 4-28)。本文と脚注で使われる罫線のアキが、この値より小さくなることはありません。

キットに参加するためのも今1つの方法は、UUNETに登録することである。も ともとは UUNETは、UNIX の批案カデループである Userdx によって負責された。UUNETは、UNIXの批案カデループである。UUNETに登録すれた。UUNETに登録すれば、CompuServe と Tyranetの企業をデータ構造地して、キットにアクセスできる。これなのデータ構造は、大学の大都市にローカルアクセス電調番号がある。ビーク特は、所「機関を選出 の Tyranet の機能報は、1時間5ドルとけっこう場合を開放である。ビーク特は、所属を選出 の Tyranet の機能報は、1時間5ドルとけっこう場合を開放である。CompuServe では、時間番を開わず1時間5ドルとけっこう場合を開放である。CompuServe では、時間番を開わず1時間5ドルとけっこう場合を開放するの。UUNET の上に対し、これに月期35ドルの UUNET メンバー料金が加算される。UUNETの批解は、バッテボーンサイトから1カップで構成されることである。また、無料で判断者が可能な UNIX ソースアーカイブも、数多く提供されている。

図 4-29 脚注の出力

4.9 **箇条書**

本文の中に現れる箇条書を指定するトリガです。箇条書の種類は必要に応じて増やすことができます。EWBでは、箇条書項目に対して自動的に番号付けを行いませんので注意が必要です。

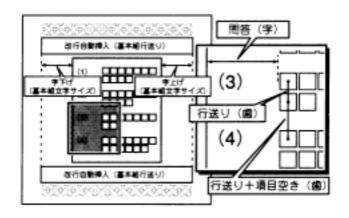


図 4-30 箇条書の体裁

4.9.1 入力方法

箇条書トリガの入力は、範囲指定で行います。箇条書にしたい段落の前に開きのトリガ"//k1{"を、箇条書にしたい段落の後に閉じのトリガ"//k1}"を入力します(リスト 4-7、図 4-31)。このとき、閉じのトリガには"//}"という省略形も使用できます。

また、箇条書トリガの前後にある改行は、小組と同様な処理を施すことができます。

//k1{
・//|EWB-Shelfが作成した"Work "と"Targets "という2つのディレクトリは、勝手に消さない
・//|原稿ファイルやディレクトリのパーミッションを読み込み不可に変更しない
・//|原稿ファイル名やディレクトリ名を勝手に変更しない。変更した場合には、book
コマンドでEWB-Shelfに対して変更したことを伝える
//}

リスト 4-7 箇条書の入力

- ・EWB-Shelf が作成した"Work"と"Targets"という2つのディレクトリは、勝手に消さない
- ・原稿ファイルやディレクトリのパーミッションを読み込み不可に変更しない
- ・原稿ファイル名やディレクトリ名を勝手に変更しない。変更した場合には、book コマンドで EWB-Shelf に対して変更したことを伝える

図 4-31 箇条書の出力

4.9.2 補足説明

問答

箇条書の1つの項目が複数行になってしまい、2行目以降のみを字下げしたい場合に用いる ものです(図4-30)。この時の指定は、全角文字の文字数/歯/ポイント/ミリで行います。

項目空き

"項目空き"とは、箇条書の各項目の行間を指定するための指定項目です(図 4-30)。ここで指定されたアキは、箇条書内で指定された行送りにプラスされます。

区切りトリガ

箇条書項目として用いる記号が種類によっては、指定された問答値と揃わない場合があります。このような時には、区切りトリガ"//|"を箇条書項目で用いる記号の後に入力することで対処します(図 4-32)。

囲み罫

小組と同様に、囲み罫を使用した箇条書を作ることが出来ます。

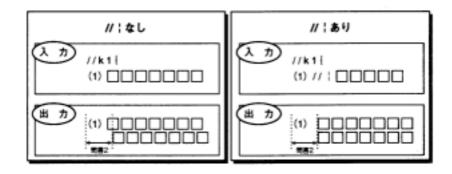


図 4-32 区切りトリガの利用

4.10 キャプション

図表類のトリガは、キャプションを定義する部分とその図表類の番号を参照する部分からなります。この両者の関係を EWB システムに伝えるために、図表トリガに続いて同じ番号を入力します。ここで入力する番号は原稿整理上で使用するものであり、実際の図表類の番号とは無関係なものです。この番号は重複していなければ、どのような順番であっても構いません。 EWB では、この番号を使って参照と定義の関係を解決しながら、自動的に図表類の番号付けをして出力してくれます。さらに、この図表類の番号には、"図 1-1(a)"などといった枝番号を振ることもできるようになっています。

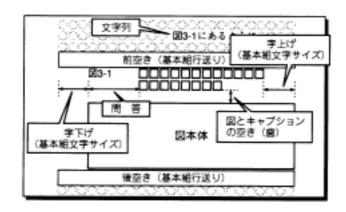


図 4-33 キャプションの体裁

4.10.1 入力方法

参照部は、実際に図表類の番号を参照する場所に、" //f 数字 "などを入力してください(リスト 4-8、 図 4-34)。また、定義部は図表類を入れたい場所に対して、" //f 数字 " などの後にスペースを入れて、キャプションの内容を記述します。この定義部は 1 行になるように入力してください。もし、枝番号を付ける場合は、" //f100-a " などといったように半角 " - " に続いて半角英数字を入れます(図 4-35)。

図表トリガの前後の改行は出力に反映されませんので、原稿が見やすいように改行をうまく利用することができます。

UUCPによる直接接続を行う場合、4番ピンと5番ピンを必ずクロスさせて接続し、2つのD TEインターフェイスが正しく会話できるようにするべきである(いうまでもなく、どちらかのコンピュータにDCEインターフェイスがあればこのかぎりでない。その場合はストレートケーブルを使用する)。 ヌルモデムケーブルのピン接続方法を//f0205に示す。

//f0205 ヌルモデムケーブルの接続

この20番ピンと6番ピンおよび8番ピンの接続は、ヌルモデムケーブルの場合にしか使えない。これは、モデムとの接続においてはDTR/DCDのハンドシェイク寝具尾が本当にとり

リスト 4-8 キャプションの入力



図 4-34 キャプションの出力

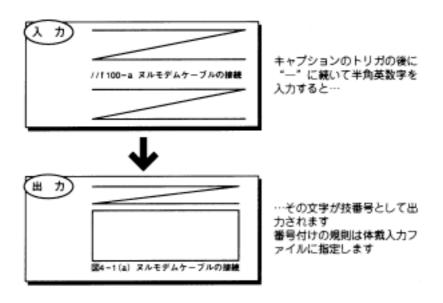


図 4-35 枝番号

4.10.2 補足説明

強調の有無

本文中に現れる図表の参照番号を強調するかどうかの指定を行います。"強調する"、"強調しない"という項目の他に、最初に出てきた参照部を強調し、それ以後は強調しないという "先頭のみ"という指定ができます。

文字列 / 枝付き文字列

キャプションの番号付けの方法を記述します。 枝番号が付く場合と付かない場合を別々に指定することができます。

前後空き

図表と本文(あるいはキャプション)のアキを指定します(図 4-33)。

図とキャプションの空き

図表とキャプションのアキを指定します。個々のキャプションに対して、指定することもできます(図 4-33)。

図表とキャプションの位置

キャプションの位置は図に対して"上右、上中、上左、下右、下中、下左、右上、右中、右下、左上、左中、左下"の12種類の中から選ぶことができます。ただし、表やプログラムリストといったテキスト類を図表として扱う場合、右上、右中、右下、左上、左中、左下にキャプションを付けることはできません。この指定は、トリガや体裁入力ファイルで行うのではなく、"物理属性エディタ"と呼ばれるツールを用いて行います(図 4-36)。



図 4-36 物理属性エディタ

4.11 プログラム

プログラムリストや画面キャラクタイメージを出力するために用意されたものです。欧文はコンスタントピッチ(各文字が等幅なもの)書体を使い、タブやスペースを認識するようになっているため、入力した通りの出力を得ることができます。また、プログラムトリガは、標準で1つ用意されていますが、必要に応じて増やすことができます。



図 4-37 プログラムの体裁

4.11.1入力方法

プログラムトリガの入力は、範囲指定で行います。プログラムにしたい段落の前に開きのトリガ"//list{"を、プログラムにしたい段落の後に閉じのトリガ"//list}"を入力します。この時、閉じのトリガには"//}"という省略形を使用することもできます(リスト 4-9、図 4-38)。また、プログラムトリガの前後にある改行は、小組と同様な処理を施すことができます。

```
//list1{
/* 英数文字列を返す */
char *read_token(str)
char *str;
 char *s;
                          //cmt{静的なバッファ//cmt}
 static char buf[MAX];
                           //cmt{英文字だったら//cmt}
 if (isalpha(*str)) {
  for (s = str + 1 ; *s && isalnum(*s) ; s++)
  strncpy(buf, str, s - str);
                          //cmt{円マークになっている//cmt}
  buf[s - str] = '\0';
   return buf;
                          //cmt{その他の場合はNULLを返す//cmt}
 } else
   return NULL;
//list1}
```

リスト 4-9 プログラムの入力

```
1: /* 英数文字列を返す */
2: char *read_token(str)
3: char *str;
4: {
5: char *s;
6: static char buf[MAX];
                             静的なバッファ
7:
8: if (isalpha(*str)) {
                              英文字だったら
9:
    for (s = str + 1; *s && isalnum(*s); s++)
10:
11:
     strncpy(buf, str, s - str);
     buf[s - str] = '\foots 0';
12:
                             円マークになっている
13:
     return buf;
                              その他の場合は NULL を返す
14: } else
15:
     return NULL;
16: }
```

図 4-38 プログラムの出力

4.11.2 補足説明

インデント

プログラムリストのタブストップ値が設定できます(図4-37)。

行番号

行番号を自動的に発生することができます。行番号に使用する書体は、任意に設定できます。 コメント

プログラムリスト中に、その行の意味を説明するためにコメントを入れることがあります。 EWB では、これを指定するためのコメントトリガを用意しました。コメントにしたい部分を "//cmt{ "と "//cmt} "で囲んでください。このトリガでは、書体や級数の指定もできます(図 4-37)。

用み罫

小組と同様に、囲み罫を使用したプログラムリストを作ることが出来ます。

4.12 **キートップ**

本文中に現れるキーボードのキートップを指定するために用意されたものです。キートップにしたい文字列を罫で囲みます。キートップの角は"直角"と"角丸"から選ぶことができ、2段で出力することもできます(図 4-39)。キートップトリガは、標準で3つ用意してありますが、必要に応じて増やすことができます。

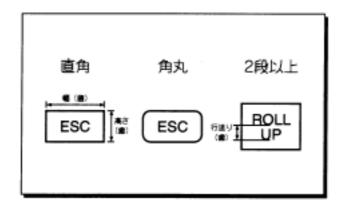


図 4-39 キートップの体裁

4.12.1 入力方法

キートップにしたい文字列を"//key1{"と"//key1}"を使って囲みます。この時、閉じのトリガには省略形"//}"を使うこともできます。また、2段のキートップを指定したい場合には、"//key1{ROLL//|UP//}"といったように、区切りトリガ"//|"を使って上下に置く文字列を区切ります(リスト 4–10、図 4–40)。

キートップトリガは、文字列の長さに応じて幅が変化しません。多種類のキートップがある場合には、事前に複数のキートップを設計することをお勧めします。

入力モードで画面をテキストで埋めていくと、しまいには画面の一番下の行に達する。最下行にきて//key1{RETURN//key1}を押すと、一番上の行が巻き取とられたように見えなくなり、画面の一番下に空行が現れて新しいテキストを入れられるようになる。この動作をスクロールという。その代わりzlは、数値引数を行番号として解釈し、それをカレント行として利用する。たとえば//key2{ROLL//|UP//key2}とするとカレント行が画面の一番上に表示される。

リスト 4-10 キートップの入力

入力モードで画面をテキストで埋めていくと、しまいには画面の一番下の行に達する。最下行にきて(RETURN)を押すと、一番上の行が巻き取とられたように見えなくなり、画面の一番下に空行が現れて新しいテキストを入れられるようになる。この動作をスクロールという。その代わり z は、数値引数を行番号として解釈し、それをカレント行として利用する。たとえば $\binom{\mathsf{ROLL}}{\mathsf{UP}}$ とするとカレント行が画面の一番上に表示される。

図 4-40 キートップの出力

4.13 罫線

水平方向の罫線を発生するものです(図 4-41)。プログラムリストや表組などといった要素と 組み合わせて使用すると有効的です。罫線トリガは必要に応じて増やすことができます。

4.13.1 入力方法

罫線を入れたい場所に、"//kei"を入力します(リスト 4-11、図 4-42)。前後に入力した改行は出力に反映されますので、罫線トリガのアキを活用したい時は、改行を取り除いてください。

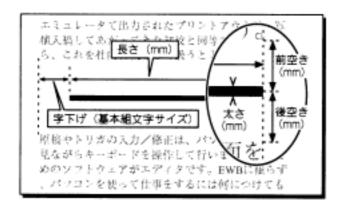


図 4-41 罫線の体裁

上位サイトを見つけようとしても、どうにもならないことがある。最も簡単なのは、メッセージを投稿して相手を探すことのはずだ。つまり、すでにネットワークに属していて、こちらと接続してくれるサイトが近くにあるかどうかをたずねればよい。しかし残念ながら、なんらかの関係がないと、この最も簡単な手段されとれない。

ほかのUNIXユーザーを知っていれば、そのなかの誰かがすでにネットに属していないかを聞いてまわるとよい。ニュースが膨大な量にのぼるため、市内電話料金の場合でもかなりの請求額となってしまうから、できるだけ近くの人を探したいだろう。運悪く、遠距離のユーザーしか知らない場合でも、その人にニュースの供給元を探すための最初のメッセージだけをポストしてもらうことはできる。

リスト 4-11 罫線の入力

上位サイトを見つけようとしても、どうにもならないことがある。最も簡単なのは、メッセージを投稿して相手を探すことのはずだ。つまり、すでにネットワークに属していて、こちらと接続してくれるサイトが近くにあるかどうかをたずねればよい。しかし残念ながら、なんらかの関係がないと、この最も簡単な手段されとれない。

ほかの UNIX ユーザーを知っていれば、そのなかの誰かがすでにネットに属していないかを聞いてまわるとよい。 ニュースが膨大な量にのぼるため、市内電話料金の場合でもかなりの請求額となってしまうから、できるだけ近くの 人を探したいだろう。運悪く、遠距離のユーザーしか知らない場合でも、その人にニュースの供給元を探すための最 初のメッセージだけをポストしてもらうことはできる。

図 4-42 罫線の出力

4.14 囲み罫

文章を罫線で囲みます。長い文章の場合は、途中で改ページさせることも可能です。囲み罫ト リガは必要に応じて増やすことができます。

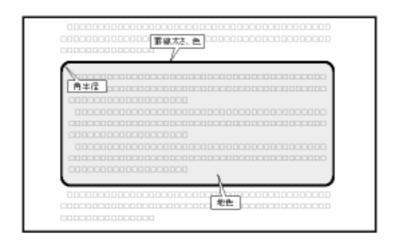


図 4-43 囲み罫の体裁

4.14.1 入力方法

//w1}

囲み罫は、範囲指定で行います。罫線で囲みたい段落の前に開きのトリガ"//w1{"を、囲みたい段落の後に閉じのトリガ"//w1}"を入力します(リスト 4-12、図 4-44)。この時、閉じのトリガには"//}"という省略形を使用することもできます。

この表を見てみると、トリガの種類が少ないと感じるかもしれません。しかし、実際に出版されている大半の書籍において、これらのトリガでも多いぐらいなのです。見出しや強調、さらに図表など数種類のトリガで作成されてしまった例もあります。 //w1{

トリガ開始文字列やトリガの本体は、状況に合わせて自由に変更することができます。また、標準トリガで対応できない場合は、トリガの種類を増やす仕掛けも用意してあります。しかし、あまり無計画にトリガを改変したりすると、可読性が低下し、混乱の元になりがちなので注意が必要です。

一般的に、トリガの種類をいたずらに増やすよりも、本の各要素をなるべく絞って表現した方が、読みやすいものになるといえるかもしれません。どうしてもトリガの種類が増えてしまうような場合は、「EWB向きではない」と判断した方がよいでしょう。

リスト 4-12 囲み罫の入力

この表を見てみると、トリガの種類が少ないと感じるかもしれません。しかし、実際に出版されている大半の書籍において、これらのトリガでも多いぐらいなのです。見出しや強調、さらに図表など数種類のトリガで作成されてしまった例もあります。

トリガ開始文字列やトリガの本体は、状況に合わせて自由に変更することができます。また、標準トリガで対応できない場合は、トリガの種類を増やす仕掛けも用意してあります。しかし、あまり無計画にトリガを改変したりすると、可読性が低下し、混乱の元になりがちなので注意が必要です。

一般的に、トリガの種類をいたずらに増やすよりも、本の各要素をなるべく絞って表現した方が、読みやすいものになるといえるかもしれません。どうしてもトリガの種類が増えてしまうような場合は、「EWB 向きではない」と判断した方がよいでしょう。

図 4-44 囲み罫の出力

4.14.2 補足説明

角半径

囲み罫の四隅を丸める場合の半径を指定します(図4-43)。0を設定すれば直角になります。

罫線太さ

上下左右の罫線の太さをそれぞれ指定します(図4-43)。

罫線色

罫線の色を CMYK 形式で指定します。

背景色

罫線で囲まれた範囲の地の色を CMYK 形式で指定します。

ページ分割

囲まれた文章がページからはみ出る場合に、途中で改ページして分割するかどうかを指定します。

4.15ルビ

文章に読み仮名等のルビを振ります。ルビトリガは必要に応じて増やすことができます。



図 4-45 ルビの体裁



図 4-46 ルビの左右揃え

4.15.1 入力方法

漢字部と読み仮名部を"//rb1{"と"//rb1}"を使って囲みます。このとき、漢字部分と読み仮名を"//rb1{漢字,かんじ//rb1}のように","で区切って入力します(リスト 4-13、図 4-47)。閉じのトリガには省略形"//}"を使うこともできます。

1番ピンは//rb1{保安用,フレーム//rb1}接地である。コンピュータとモデムを接続する場合はホスト側に、また、コンピュータ同士を直に接続する場合はいずれか一方の側にこのピンを接続し、ケーブルのもう一方の側は接続しないままとするべきである。RS-232の正しい実装は、ポートの1番ピンをシステムのフレーム内部でアースに接続するというものである。RS-232D規格では、両側とも接続し、モデム内部で//rb1{断線,ブレーク//rb1}することになっている。RS-232D規格では、上記のように、DTE側だけで接続する。ほとんどすべてのメーカーが信号用接地とフレーム用接地をモデム内部で相互に接続させているが、これは誤りである。

リスト 4-13 ルビの入力

1番ピンは保安用接地である。コンピュータとモデムを接続する場合はホスト側に、また、コンピュータ同士を直に接続する場合はいずれか一方の側にこのピンを接続し、ケーブルのもう一方の側は接続しないままとするべきである。RS-232の正しい実装は、ポートの1番ピンをシステムのフレーム内部でアースに接続するというものである。実際の規格では、両側とも接続し、モデム内部で断線することになっている。RS-232D規格では、上記のように、DTE側だけで接続する。ほとんどすべてのメーカーが信号用接地とフレーム用接地をモデム内部で相互に接続させているが、これは誤りである。

図 4-47 ルビの出力

4.15.2 補足説明

本文ルビ空き

本文とルビ文字の空きを指定します(図4-45)

ルビ級数比率

本文文字に対するルビ文字の大きさを比率で指定します(図 4-45)。

揃え

本文文字とルビ文字をどこで揃えるかを指定します。左/中/右が指定できます(図 4-46)。 食い込み

ルビが本文より長い場合に、ルビ文字を前後の本文のどこまで重ねるかを指定します。 食い 込み量は本文の文字数で指定します。

4.16 傍点

文章に1文字ずつ傍点を振ります。傍点トリガは必要に応じて増やすことができます。

4.16.1入力方法

傍点を振りたい部分を"//bou1{"と"//bou1}"を使って囲みます(リスト 4-14、図 4-49)。閉じのトリガには省略形"//}"を使うこともできます。

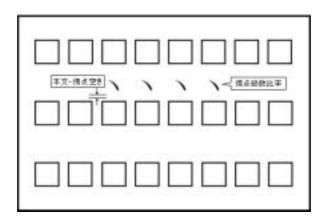


図 4-48 傍点の体裁

入力フィールドには文字列を入力するフィールドと//bou1{いくつかの候補//bou1}から選択するフィールドの2種類があります。入力の方法は、まず上下の矢印キーを使い、入力したいフィールドにカーソルを合わせます。このフィールドが候補から選択するフィールドであれば、//bou1{左右の矢印キー//bou1}を使い必要な項目を選択します。また、文字列を入力するフィールドであれば、文字入力をはじめる前にスペースキーを1回押します。文字入力が終ったらリターンキーか下矢印キーを押し、フィールドの内容を確定します。

リスト 4-14 傍点の入力

入力フィールドには文字列を入力するフィールドと $\dot{\mathbf{n}}$ く $\dot{\mathbf{o}}$ が $\dot{\mathbf{o}}$ 候 補から選択するフィールドの 2 種類があります。入力の方法は、まず上下の矢印キーを使い、入力したいフィールドにカーソルを合わせます。このフィールドが候補から選択するフィールドであれば、 $\dot{\mathbf{c}}$ 右 $\dot{\mathbf{o}}$ 矣 $\dot{\mathbf{n}}$ $\dot{\mathbf{r}}$ $\dot{\mathbf{e}}$ き $\dot{\mathbf{e}}$ 他の決定な項目を選択します。また、文字列を入力するフィールドであれば、文字入力をはじめる前にスペースキーを 1 回押します。文字入力が終ったらリターンキーか下矢印キーを押し、フィールドの内容を確定します。

図 4-49 傍点の出力

4.16.2 補足説明

傍点文字

傍点として出力する文字を指定します。

本文傍点空き

本文と傍点文字の空きを指定します(図4-48)

傍点級数比率

本文文字に対する傍点文字の大きさを比率で指定します(図4-48)。

傍点位置補正

傍点の位置を字送り方向に調整します。

4.17 **ハイパー**リンク

WEB サイト等への URL を記述します。最終出力を PDF へ変換するとリンクが有効になり、URL 文字列をクリックするとリンク先にジャンプします。

4.17.1 入力方法

リンク先として記述する URL を " //url { " と " //url } " を使って囲みます。閉じのトリガには省略 形 " // } " を使うこともできます。このトリガで囲まれた部分は自動改行ができません。プレビュー 等で行末の状態を確認し、改行が必要であれば、トリガ中の改行したい場所に " @ @ break() " を入れて強制的に改行させてください。

4.18 索引

EWB には、索引を自動生成する機能があります。EWB は本文を組んだ結果を元に、索引に入れる項目とページ番号を自動的に抽出し、索引を作成します。

この索引を作成する作業は、通常の本文ページを作成する作業とは異なり、独立した工程になります。

4.18.1 EWB **の索引作成**

まず原稿ファイル中で、索引項目としてピックアップしたい単語に対して、"索引トリガ"を入力します。この索引トリガとは、「この単語を索引にしたい」という印になるわけです。そして、実際の索引項目や読み入力などは、"索引入力エディタ"と呼ばれる専用ツールを用いて行います。この入力結果は、"索引ファイル"と呼ばれる原稿ファイルとは異なるファイルに収められます(図 4-50)。

また、索引の体裁に関しては、本文と異なる"索引体裁入力ファイル"に記述します。このファイルの内容は、段間のアキやリーダー罫の種類などといった索引固有の項目が、通常の体裁入力ファイルに付加されたものです。

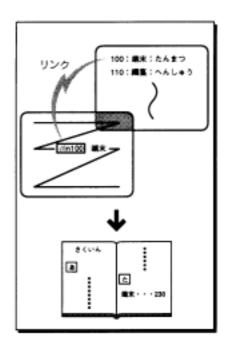


図 4-50 EWB の索引作成

EWB では、これら3つのファイルを読み込んで、索引ページを自動的に作成します(図 4-51)。

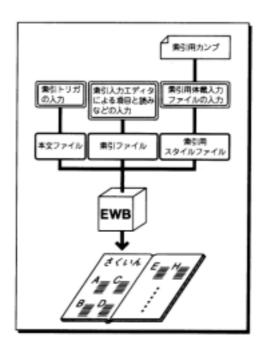


図 4-51 EWB の索引体裁作成

4.18.2 索引トリガ

索引トリガは、原稿中に索引項目としてあげたい単語に対して、" //in 数字 " を入力します。このトリガの入力規則は、図表トリガなどと同様です。

4.18.3 索引入力エディタ

実際の索引項目やソートに使う読みなどは、索引入力エディタを使用して入力します(図 4-52)。このエディタは原稿ファイルを読み込んで、入力された索引トリガを元に、会話式に必要なデータを入力していくもので、原稿の修正などは一切できないようになっています。



図 4-52 索引入力エディタ

4.18.4 索引の体裁

段数

2段組などの多段組ができるようになっています。

カラム

多段組にした場合のカラム間のアキ(段間のアキ)が指定できるようになっています(図 4-53)。 また、カラム間に引く罫線の太さ(段間罫)も指定できます。

ページ番号

索引項目に対するページ番号は、通常に使用する書体の他に、ボールド、イタリックの指定 もできます。

レベル

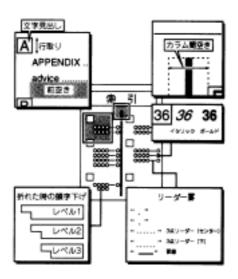


図 4-53 索引の体裁

索引項目のレベルを3種類設定できます。それぞれのレベルに対して、異なる体裁を指定できるようになっています。



レイアウト

これまでは、トリガはどんな役割りをして、どのような表現力を持っているかなどといった原稿の要素指定を中心に説明してきました。EWBでは、このようなことがわかってしまえば、後は、簡単なことばかりです。というのも、これから版下作成までの作業はあなたの代わりにコンピュータがほとんどやってくれるからなのです。

この章では、用意されたトリガ入り原稿をどのようにレイアウトし、版下作成を行っていくかといったことについて触れてみます。バッチ処理を採用した EWB の特徴をよく理解して効率的な版下作成を行ってください。

5.1 **便利なオートレイアウト**

ここでは、EWB における版下作成の様子を見てみることにします。トリガ入り原稿がコンピュータ上でどのように扱われて版下になっていくのでしょうか。また、どのような部分が自動化され、効率が上がるのでしょうか。

5.1.1 指定した要素を流し込む

EWBではトリガを使って原稿中に文書の論理的な記述を行い、そのトリガの物理的意味付けは体裁入力ファイルに記述するという説明をしてきました。しかし、これらの情報がばらばらでは目的の版下を得ることはできません。2つの情報が一緒になることではじめて紙面上の表現ができます。EWBは、これらの情報が組版処理を行う時に結合される仕掛けになっており、バッチ処理の特徴がこの時点で現われます。

まず、編集者が EWB に対して組版指示を行うと、EWB はトリガのはいった原稿と各トリガの体裁を読み込みます。原稿の先頭から段落ごとにトリガで指定された通りに文字を組んでいき、

組み上がった段落を順番に仮想的な巻紙の上に置いていきます(図 5-1)。そして、ある程度出力できる要素が集まったら、仮想的な巻紙をページ単位に切っていきます。この時に、脚注や図表類のように出力する位置が固定しないものは、本文とは別の巻紙に置いていき、出力時に最適な場所に挿入されるようになっています。また、柱やノンブルも指定された体裁で割り付けられます。このような処理を自動的に繰り返して、必要な版下が出来上がるわけです。

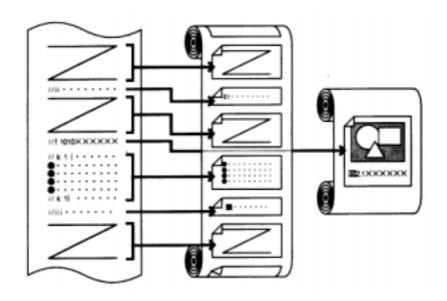


図 5-1 仮想的な巻紙

5.1.2 **各要素の自動配置**

ワープロやエディタで作成した文書を EWB で処理すると、事前に決められている版面に従って、自動的に文字が流れていき、ページ上に収まるようになっています。その際、組版処理に伴うさまざまな作業、たとえば、どういった禁則処理を行い、どの単語にどのようなハイフネーション処理を施すかなどといった処理は、すべて EWB が自動的に行ってくれます。特別な作業を行う必要はありません。

また、トリガで指定された各要素も決められた規則に従って、自動的に配置されていきます。 見出しがページ末に1行だけ残ってしまうような場合に改ページを自動的に行って次のページに 送ったり、複数の脚注を出力する場合にいくつまで入るかどうかを自動的に判断してページ末に 配置するなどといったページアップに必要な処理を行います。このように EWB では、各要素に 対して試行錯誤しながら適切な位置を判断し、コンピュータがいちばん美しく見えるように組み 上げてくれるのです。

5.1.3 図表類の位置を決める

版面上にさまざまな要素をレイアウトしていく上で、特に重要なものは図表類の扱いです。図表類と内容は大変深い関係があり、一般に読み易さを優先するため図表類の位置指定は、多くの労力を必要とします。EWBでは、図表類が自動的にページ上の適切な位置にくるように工夫を凝らしています。

フロートしない図表類

プログラムリストや長い表などのように、複数ページにまたがって出力して欲しい図表類を、 EWB では"フロート(浮動)しないもの"と呼びます。この分類に属する図表類は、原稿中に指 定した位置に必ず出力されるようになっています。指定された図表類が複数ページに分かれる場 合は、本文の組版と同様に適切な位置で自動的に改ページされます(図 5-2)。

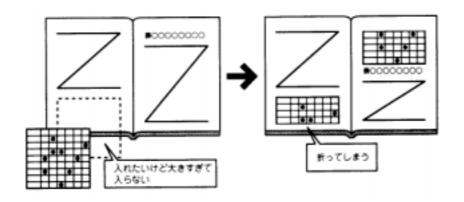


図 5-2 フロートしない図表類

フロートする図表類

キャプションの付いた図や表などが、本文中にある参照番号の近くに位置して欲しいような図表類を、EWB では"フロートするもの"と呼びます。この分類に指定された図表類は複数に分かれてしまうことはなく、自動的に本文の参照番号と一緒に流れていき、参照番号が存在するページ内、もしくはその数ページ以内に出力されるようになっています(図5-3)。

"フロートするもの"として指定された図表類は、それぞれに対して"here、top、bottom、page"という4つの位置指定を組み合わせて、レイアウト規則を記述することができます(図 5-4)。たとえば、"here、top、bottom"の順序で指定された場合、まず図表トリガの書かれている位置に図表の挿入を試みます。もし、きちんと収まるのであれば、その部分に図表を挿入します。この時、うまくその位置に入らない場合には、次のページの先頭に入るかどうかを試してみます。そして、

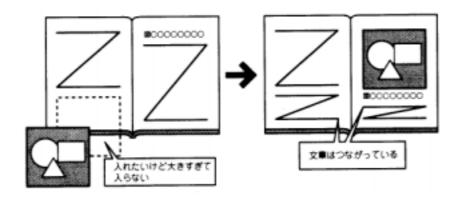


図 5-3 フロートする図表類

それでもうまく入らない場合は、ページ末に挿入しようとし、最適な位置を見つけ出します。 このように、EWB では指定されたレイアウト規則に基づいて、図表の大きさと挿入位置の整合性をチェックしていき、適切な場所を自動的に捜していくのです。

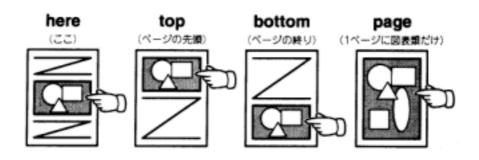


図 5-4 図表類の位置指定

5.2 思い通りにレイアウトする

組版やレイアウトに関連した作業は、すべて EWB が受け持って処理してくれます。しかしながら、コンピュータは規則に基づいて処理を行うだけですから、私たちにとって満足のいくものを提供してくれるとは限りません。もちろん EWB も同様で、EWB が適切な位置を判断して各要素を出力したとしても、思った通りのレイアウトが得られない場合もあるわけです。

そこで、EWB には、自動的に組み上げられたレイアウトを、手作業で思い通りの位置に修正するための機能が備わっています。

5.2.1 組版トリガとは

自動的に組み上げられたレイアウトを手作業で修正する場合にも、やはりトリガを使用します。 手作業で、行のアキ調整や改ページなどといったことを行うトリガは、"編集トリガ"と対比させて、"組版トリガ"と呼びます(図 5-5)。

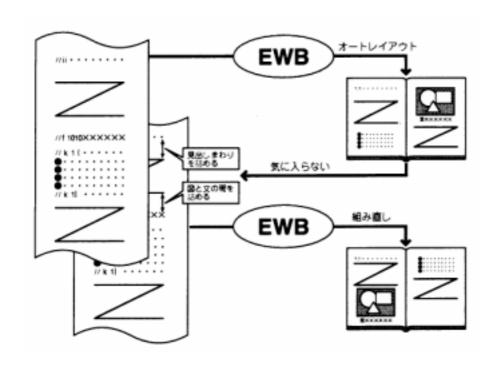


図 5-5 組版トリガによるレイアウト修正

組版トリガの種類としては、大きく分けて次の3つがあります。

- ・改行
- ・改ページ
- ・縦/横方向のアキ調整

この他にも、"字下げ"や"揃え"などといった組版トリガも用意されており、柔軟に対応することができます。

5.2.2 強制改行

行中にきれいに収まらなかった単語や文字に対して、任意の位置で強制改行させることができます(図 5-6)。

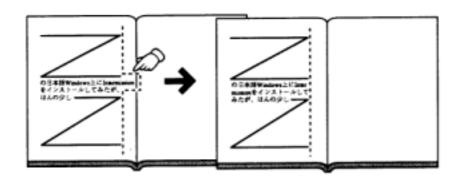


図 5-6 強制改行

英字の製品名や技術用語には、どこでハイフネーションを行ったらよいのかわからない場合も数多くあります。EWBでは、このような英単語を見つけた場合には、自動的に分けるのをやめてしまい、版面からはみ出して出力されるようになっています。このような場面に遭遇した時に、強制改行の組版トリガを用いて対処します。また、これと反対に、自動的に改行されてしまうような英単語などを分割禁止にすることもできます。

もちろん英単語だけではなく、複数行になってしまう見出しなども、思い通りの位置で改行することができます。

5.2.3 強制改ページ

指定した位置で強制的に改ページを行わせることができます。段落と段落の間だけでなく、文書中の任意の位置で改ページさせることができます(図 5-7)。

重要な見出しや文書がページ末にあり、読み易さを損なうような場合には、強制的に改ページをさせて、次のページに送ることができます。

5.2.4 アキの調整

文字間や行間を必要な量だけ、一時的に広げたり詰めたりすることができます(図 5-8)。図表類や見出しの前後のアキ調整を行ったり、文字間が大きく空いている場所を詰めたりする時に用いる組版トリガです。

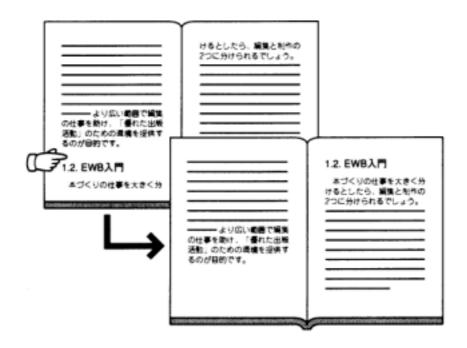


図 5-7 強制改ページ

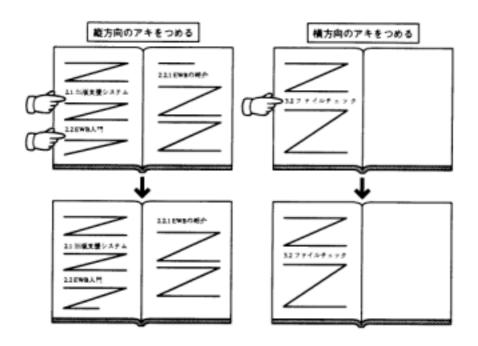


図 5-8 縦/横方向のアキ調整

5.3 ワークサイクル

EWB で組版した結果は、プリンタに印刷したり、画面上に表示させて確認することが簡単にできます。プリンタや画面で組版結果を確認し、気に入らなければもう一度、記述されたトリガの位置を変更したり、トリガの種類を変えたりして、EWB に処理をさせます。この作業を、納得の行く結果が得られるまで繰り返し、最終的に出来上がったものをフィルムや印画紙に出力するのです。EWB システムでの版下作業のサイクルは、このような"修正&確認"という作業の繰り返しになっています(図 5-9)。

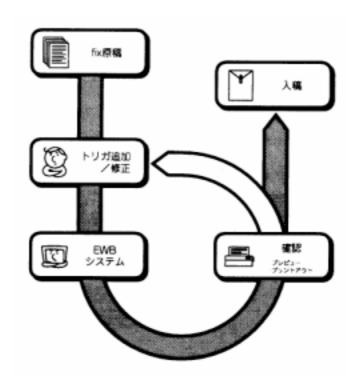


図 5-9 EWB のワークサイクル

もちろん、コンピュータが完璧なものを仕上げていれば、このようなレイアウト修正の作業は必要ありません。単に、トリガの入力ミスがないか、書き忘れはないかなどといったことを確認するだけで済みます。

しかし、コンピュータが規則に基づいて組み上げたレイアウトであるため、どうしても、私たちの目で見て感じ、そして納得した美しさとは、比較することができません。そのため、EWBでは、すべてをコンピュータにまかせるのではなく、自動的な組版処理と手作業によるレイアウト処理を部分的に組み合わせて、より品質の高い版下を得ることができるようになっています。

第 2 部

実践編



実作業をはじめる 前に

第1部の基礎編では、EWB の考え方やトリガの表現力などの電子化をする上で重要なポイントを中心に、解説をしてきました。本編は、これらの知識を踏まえて、EWB の操作を含む実作業の面から述べていきます。

そこでまず、この章では EWB の実作業をスムースに行えるように、体裁入力や組版などといった各フェーズの流れとそれらの関係を見てみることにします。実作業をはじめる前に、どのような手順で作業を進めていくのかをしっかりと把握してください。

1.1 実作業の流れ

EWB を操作して写植出力を手にするまでには、大きく分けて 7 つのフェーズから成り立ちます。1 つ 1 つのフェーズで、必要なファイルを作成していき、最終的にこれらのファイルが有機的に結び付きます。各フェーズにはそれぞれの専用ツールがありますが、特に、レイアウト処理をするような場合には、簡単に操作ができるように "EWB-Shelf"と呼ばれるインターフェイスプログラムが用意されています。このプログラムを利用することで、必要なファイルをすべて管理することができるため、各フェーズで効率的な作業を行うことができます(図 1-1)。

1.1.1 原稿入力

筆者から受け取った原稿ファイルは、" 章単位 " に分けて管理を行い、文字入力の統一を図ります。原稿中の文字や約物は、推奨する入力仕様に従って再入力していきます。 さらに、編集トリガを使用して文書の論理的な意味付けを行います。 EWB には、専用のエディタがありませんので、入力作業は好みのエディタやワープロを使用してください。

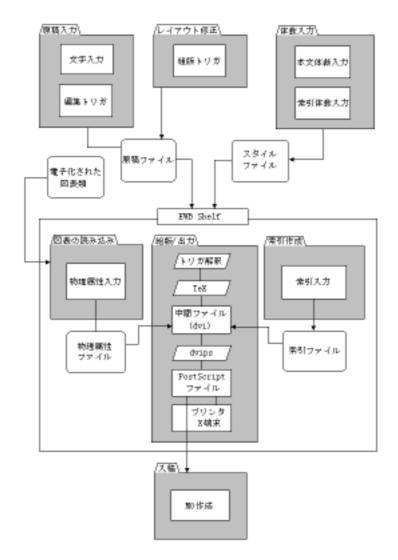


図 1-1 EWB の各フェーズ

1.1.2 体裁入力

レイアウト指定を元に、判型や基本組、そしてトリガによって表現される要素の体裁をコンピュータ上で数値入力する作業を行います。この数値は、"体裁入力ファイル"と呼ばれるものに記述していきます。最終的には、このファイルから組版処理に必要な"スタイルファイル"というものに変換して作業を行っていきます。

また、作成中の本の中には、さまざまな体裁が存在する場合もありますが、このような時には、 それぞれのスタイルファイルを作成します。特に、体裁が特殊な索引は本文とは別に、"索引体裁 入力ファイル"と呼ばれるものが用意されています。

1.1.3 図表の読み込み

コンピュータ上で作成した図表ファイルと原稿中にある図表トリガとのリンクや貼り込み用のアタリ罫などの情報を設定する作業を行います。これらの情報は、EWB-Shelf 上にある"物理属性エディタ"と呼ばれる専用ツールを用いて入力や修正を行い、"物理属性ファイル"に保存します。このファイルには、図表やキャプションの位置、アタリサイズなどのレイアウト情報が記述できるようになっています。

ただし、この作業は原稿中で図表トリガを使用しなかった場合には、必要ありません。

1.1.4 組版/出力

トリガ入力と体裁入力が済み、図表類のレイアウト情報の設定が終ったら、組版と出力のフェーズに移ります。ここではじめて、PS プリンタや X 端末 (X-Window が使用できる端末)を使用して、版下イメージを確認できるわけです。

組版や出力は、すべて EWB-Shelf が行ってくれます。まず、原稿中のトリガを解析して文法のミスがあるかどうかをチェックします。正しくトリガが入力されていれば、組版プログラム(日本語 TeX)により、組版とレイアウトが行われます。そして、この組版結果は PS (PostScript)ファイルに保存されます。最後に、プリンタや X 端末などといったデバイスを選択し、この PS ファイルを出力することによって目的の版下を得ることができるのです。

1.1.5 レイアウト修正

EWBのオートレイアウト機能によって作成されたページを手直しする作業を行います。思い通りにレイアウトされていないページは、原稿中に組版トリガを入力して対処します。また、図表の出力位置を変更する場合は、物理属性エディタを使って位置修正を行います。

もし、レイアウト修正を頻繁に行うような事が起きてしまったら、プリンタを使用せずに、X端末などのディスプレイで確認した方が効率アップにつながります。

1.1.6 索引作成

本文の版下が出来上がったら、索引作成を行います。原稿中に入れた索引トリガに対して、索引として入れる項目とその読みの情報を入力します。これらの情報は、"索引入力エディタ"と呼ばれる専用ツールで入力や修正を行い、"索引ファイル"に保存します。このファイルと本文の組版結果からページ情報を取得し、アルファベットや五十音順に並べ替えられます。そして、索引も本文と同様に、スタイルファイルに基づいて組版が行われます。

1.1.7 最終出力

EWB-Shelf によって生成された PS ファイルを面付けし、入稿してフィルム出力を行います。 PS ファイルはファイルサイズが大きいので入稿には MO を使用します。

PDF にしてホームページに掲載という場合は、面付けしない PS ファイルを作成し、Windows または Macintosh 上で Acrobat Distiller を使って PDF に変換します。

1.2 EWB **の作業環境**

ここでは、EWB を使うにあたってどのような環境が必要なのかを説明します。まずは、作業をはじめる前に、自分の環境を確かめてみましょう。

1.2.1 EWB マシン

EWB システムは、いままで説明してきましたように、UNIX 系 OS 上ですべて動作するものです。PC を用意して、FreeBSD または Linux をインストールし、EWB システムを稼動させることになります。EWB マシンが用意できたらユーザーはそのマシンにログインして EWB を使用することになります。

1.2.2 各自のマシン

各自の机に必要なマシンは、EWB マシンの端末として telnet が使用できるものであれば、どんなマシンでもかまいません。また組版結果を画面で見るために X 端末 (または X エミュレーションソフト)があると大変便利です。

1.2.3 プリンタ

校正のために出力するプリンタには、PostScript プリンタを使用します。プリンタには必要な PostScript フォントをインストールしてあり、また UNIX の lpr コマンドで使用できる状態にして ある必要があります。



原稿入力

EWB では原稿ファイルの作成、すなわち文字や約物、そしてトリガをいかに効率よく入力していくかということが、各フェーズの中で一番のポイントとなります。

ここでは、文字やトリガを入力する際に気を付けなければいけない基本的な規則を中心に解説をします。編集者だけでなく、筆者などといった原稿ファイルの作成に携わる方々は、大きな作業の効率アップにつながりますので、ぜひ一読してください。

2.1 原稿の文字コード

EWB 原稿では漢字コードとして EUC または Shift-JIS を使用します。漢字コードが JIS の文書である場合はあらかじめ漢字コードを変換しておく必要があります。

改行コードは UNIX 形式、DOS 形式、Macintosh 形式のいずれでも扱うことができますが、UNIX のエディタで扱うことを考えてあらかじめ UNIX 形式に変換しておいたほうがいいでしょう。

漢字および改行コード変換には tounix というコマンドを使用します。tounix は漢字コードを EUC、改行コードを UNIX 形式に変換します。

% tounix 原稿ファイル名

ファイル名を複数書いて一括変換することもできます。

2.2 原稿の管理

筆者から受け取った原稿を、まず整理しなければなりません。EWB では、基本的に、" 章単位 " で処理を行っていきます。したがって、原稿のファイルは、必ず章単位で扱ってください。この 時、ファイル名は " chap01.ewb " " appendix.ewb " などといったようにわかりやすいものにすると 原稿管理に役立ちます。

また、作成するファイルが何章にあたるかという情報を、EWBに伝えなければなりません。これは、ファイルの先頭に"chapter"という文字列を使って表します。たとえば、第2章の原稿ファイルであれば、"chapter 2"などといった具合いに記述するわけです。

2.3 文字の入力規則

EWBでは、パソコンのキーボードから入力した文字は必ず出力されます。たとえば、全角文字で"12、000US\$"と入力しても、半角で"12,000US\$"と入力しても問題なく出力は行われます。しかし、この場合に、内容的な違いはありませんが、出力される文字の表現にちょっとした差が現れます。

ここでは、きれいな出力を得るためには、文字や約物をどのような規則で入力していったらいいのかという"コツ"について説明します。

2.3.1 入力した文字はそのまま出力

一般に、ワープロなどで漢字やひらがな類を入力する場合は全角のみとなりますが、英数字類は半角と全角の2種類で入力することができます。EWBでは、入力された文字の種類によって、出力形式が異なってきます。

まずここでは、全角と半角の入力による違いを見てみましょう。

英数字

英数字を、全角で入力した場合と半角で入力した場合では、書体と字送りに違いが現れてきます。全角で入力した文字は和文書体を使って"全角送り"で出力され、半角で入力した文字は欧文書体を使って"欧文送り(プロポーショナルピッチ)"になります。

また、縦組の場合には、この違いが顕著に現れます。英数字などを全角で入力すると漢字と同じ方向の"立った状態"に、半角で入力すると 90 度回転した"寝た状態"に出力されます(図 2-1)。 縦組の英数字入力は、意識的に文字を打ち分ける必要があるため注意が必要です。

全角で入力したMacintoshを 256 倍使うための本

図 2-1 縦組における全角文字 と半角文字

約物/記号

約物や記号に関しても、英数字と同様に、全角と半角では出力結果が異なってきます。そのため、状況によって全角と半角をきちんと打ち分けなければなりません。たとえば、パーレンを全角と半角で入力してみると、半角で入力したパーレンは隣合った和文に比べて少し下がって出力されます(図 2-2)。これは、パーレンが欧文ベースラインの位置に合ってしまうためで、このような場合に半角を用いるのは、不適切といえます。しかし、このことがすべての約物や記号類にいえるとも限りません。"-"や"*"などといった記号は、半角の方が適切な場合が多いのです。このように約物や記号は入力の仕方によって、見え方がさまざまに変化するため、まずは、文字入力する前に、各自で統一的な入力仕様を決めることをお勧めします。Appendix E (359ページ)に、推奨する入力仕様をあげておきますので、参考にしてください。

全角で入力した(パーレン)です。

半角で入力した (パーレン)です。

図 2-2 パーレンの入力

改行

EWB では、行末に入力された1つの"改行"によって段落の終りを認識します。そして、ワープロなどと同様に、改行のみの行は空行を意味することになり、改行を複数個入力することで段落と段落の間を大きく空けることができます(図 2-3)。

また、ワープロ画面上で行を揃えるために、段落の途中で改行を入れてしまうと、EWB は改行 以降の文書を違う段落と見なして組版してしまうため注意が必要です。

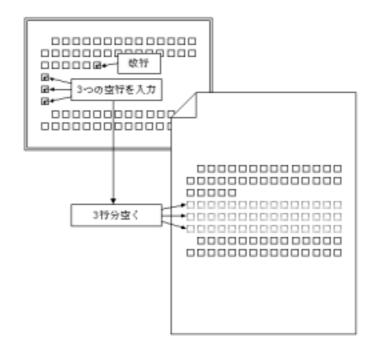


図 2-3 改行で段間を空ける

スペース

ワープロで入力できる"スペース"には、全角スペースと半角スペースがあります。全角スペースは、全角幅のアキを空けるのに使え、文字幅はいつも一定に保たれます。たとえば、段落先頭の1文字をインデントさせる時や、語と語の区切りなどに用いることができます。

一方、半角スペースは欧文スペースとして扱われるため、英単語の区切りなどに使うことができます。この欧文スペースは、組版によって文字幅が変動し、行末にくると消滅してしまいます。 ハイフネーションができないような長い単語を行末で分ける場合に、この半角スペースを入力しておくと、その位置で改行ができるという効果もあります。

タブ

タブは、単語や文書のカラム揃え(桁揃え)に使用することができます(図2-4)。本文中によくある簡単な表やリストなどは、タブを用いてカラムを揃えておくと、簡単に縦の方向に揃って出力されます。この時、カラム間のアキは、タブストップの値で決ってきます。ワープロ上のタブストップの値を変更して入力してください。

src 実際の電子ニュースプログラムのソースコード。
doc インストレーションマニュアルを含むマニュアル。
man 電子ニュースプログラムのそれぞれについて解説したオンラインマニュアル。
Misc 何かの様に役に立つ投に立つアクセサリブログラム(ただし、電子ニュースの基幹部分ではない)。
uscp 電子ニュース同に UUCPの最適化を行う各種ソースコードバッチ。UNIX ソースライセンスがない場合は、使う必要がない。
cvt 田パージョンのニュースを変換するための各種スクリプト。

図 2-4 タブによる桁揃え

コントロールコード

EWB では、タブ以外のコントロールコードは、すべて無視されます。ワープロなどで使用される改ページ " ^L " などは、原稿中に入力されていても、組版には影響しません。

2.3.2 EWB **で出力できない文字**

この章のはじめで、「EWB は、キーボードから入力した文字は必ず出力されます」と述べました。しかし、ワープロなどで入力できる一部の文字は、出力デバイスのシステムとの違い(文字コード体系の違いなど)で、場合によっては多少の工夫をしないと出力されないことがあります。そこで、EWB では、すべてのコンピュータで確実に文字が扱えるように、入力文字を"半角 ASCII と JIS で定められた全角文字"と規定しています。

EWB で扱えない文字

EWB では、半角カナとメーカーが独自に用意した JIS 以外の全角文字を使用することができません。たとえば、NEC の PC-9801 を例にとると

- ・半角カナ(0xa1~0xdf)
- ・NEC 外字 (JIS:0x2921 ~ 0x2d7f と JIS:0x7920 ~ 0x7c7f)

の範囲の文字は使用できないことになります。しかし、この NEC 外字と呼ばれる中には、よく使

われる"1~10までの丸付き数字"と"ローマ数字"もはいっています。EWBでは、これらの文字に限って標準で扱えるようにしてあります。

もし、半角カナ文字が原稿中に入力されていると、

:sample.doc(34行):TeXでは扱えない文字です .終了しますsample.doc:エラー発生:pass1

のようなエラーメッセージが組版時に出力され処理を中止します。また、NEC 外字が原稿中にあると、

:sample.doc(32行): TeXでは扱えない文字です.

のような警告メッセージが組版時に出力され、その文字は正常に出力されません。

2.4 編集トリガの入力規則

トリガ入力は、EWB では避けては通れない作業です。なぜなら編集者の編集意図は、トリガによって、EWB システムに伝えられるからです。ここでは、正しく、そして効率よくトリガを入力するために、知っておかなければならないことを解説していきます。

2.4.1 トリガの基礎知識

トリガには、"編集トリガ"と"組版トリガ"の2種類があり、原稿入力時に入れるものは、このうち編集トリガを使用します。編集トリガは、執筆者や編集者の意志を表現するもので、原稿に対して"見出し"や"箇条書"などといった論理的な意味付けをするものです。トリガは、"トリガ開始文字列"と呼ばれる半角記号とそれに続く数個の半角英数字から構成される文字列で、文書中に埋め込まれます。通常、このトリガ開始文字列は、"//"を使用します。もし、原稿中でこの開始文字列を使いたい場合は、"///"のように記述してください。

また、トリガは各要素の種類によって入力方法が、多少異なってきます。たとえば、要素の範囲指定をするトリガの場合は、開始を表す"{"と終了を表す"}"を用いたり、図表キャプションや脚注の場合には、定義と参照関係を解決するための番号を入力したりします。このように、各トリガは要素によって、入力規則が決められているのです。

2.4.2 編集トリガのグループ分け

編集トリガは、トリガの入力方法やその性質などによって、8 つのグループに分類することができます。

単純トリガグループ

罫線トリガ ("//kei") が、このグループに属します。1 つのトリガが単純に1 つの機能に置き換えられるため、このような呼び方をしています。

見出しトリガグループ

見出しトリガ("//i")が、このグループに属します。

グループトリガグループ

小組(" //c1{ ") / 表組(" //table1{ ") / リスト(" //list1{ ") / 囲み罫(" //w1{ ") のトリガが、このグループに属します。開始のトリガと終了のトリガによって囲まれた部分に対して作用します。このトリガは、行の途中に入力することができません。

強調トリガグループ

強調文字(" //g{ ") /ルビ(" //rb1{ ") / 傍点(" //bou1{ ") のトリガが、このグループに属します。グループトリガと同様に、開始と終了のトリガによって囲まれた部分に作用します。グループトリガとの違いは、文章のどこの位置にでも入力できるという点です。

図表トリガグループ

図表トリガ (" //f " など) が、このグループに属します。このトリガには、定義部と参照部があります。

箇条書トリガグループ

箇条書トリガ (" //k1{ ") が、このグループに属します。グループトリガと同様に、開始のトリガと終了のトリガがあります。

脚注トリガグループ

脚注トリガ (" //ky ") が、このグループに属します。このトリガにも図表トリガと同様に、定義部と参照部があります。

非展開トリガグループ

強調トリガと同様に、開始と終了のトリガによって囲まれた部分に作用します。強調トリガとの違いは、囲まれた部分が EWB により展開されずにそのまま TFX に渡されるという点です。

ラベルトリガグループ、参照トリガグループ

ラベルトリガ ("//label { ") や参照トリガ ("//pageref { " など) が、このグループに属します。これらのトリガはラベルと参照を組み合わせて使用します。見出しの相互参照のためのラベルをラベルトリガで作り、参照トリガを使用してページ番号、文字列を参照します。参照トリガは見出しだけでなく図表に対しても使用できます。

URL トリガグループ

URL トリガ (" //url { ") が、このグループに属します。開始のトリガと終了のトリガによって 囲まれた部分に対して作用します。このトリガは WEB ページ等の URL に対して使用します。 組み込みトリガ

EWB に、最初から組み込まれているトリガです。

2.4.3 トリガの入力方法

それでは、各グループごとにトリガの入力方法を詳しく見ていくことにしましょう。

単純トリガグループ

単純トリガは、その用途に合わせて適切な位置に入力してください。ただし、単純トリガの直後に英文字がある場合は、トリガとその英文字との区別ができなくなってしまうため、トリガの後に半角スペースを1つ入力してください。

//kei

見出しトリガグループ

見出しトリガは、必ず行頭から入力します。見出し文字列(内容)は、トリガの後にスペース を入れ、それに続いて入力します。見出しトリガの有効範囲は、トリガが入力されている行だけ なので、見出し文字列の中には改行を入れないでください。

また、見出しの前後に入力した空行は、EWBシステムによって無視されるため、出力には反映しません。

//i 見出し文字列

グループトリガグループ

グループトリガには、開始を表すトリガと終了を表すトリガがあります。開始を表すトリガは、必ず行頭に入力してください。また、終了を表すトリガの後には、文字やトリガは入力できません。必ず、終了を表すトリガの後には改行を1つ入力してください。この終了を表すトリガは、"//}"という省略形を使用することもできます。

```
//c1{......開始を表すトリガ
//c1}......終了を表すトリガ
```

このグループトリガには、改行自動挿入という便利な機能が備わっています。開始トリガの前と終了トリガの後に、統一したアキを自動的に入れる仕掛けです。この指定がされている場合には、開始トリガの前の空行と終了トリガの後の空行を無視するようになっています。

さて、このグループトリガの中で1つだけ特殊なトリガがあります。それは、表組トリガで、オプション指定という機能です。開始トリガ("/table1")と"{"との間に、オプション文字列と呼ばれるものが指定できるようになっています。このオプション文字列は、ブラケットを使って記述するもので、表のカラム幅や揃え(左、中、右)の体裁指定ができます。

この表組トリガの入力仕様を詳しく記述すると、次のようになります。

たとえば、次のようなトリガ指定を考えてみましょう。

```
//table1[c(5w)r(8w)1]{......開始を表すトリガ
//c1}.....終了を表すトリガ
```

この指定では、表の第1カラム目は5文字分の幅でセンター揃えをし、第2カラム目は8文字

分の幅で右詰めを行い、第3カラム目の幅はなりゆきで左揃えになります。

強調トリガグループ

強調トリガもグループトリガと同様に、開始を表すトリガと終了を表すトリガがあります。この時、終了を表すトリガは、"//}"という省略形が使えます。

//g{強調したい文字列//g}

このトリガグループで、多少特殊な使い方をするのがルビトリガです。ルビを振りたい漢字と 読み仮名を"."で区切って記述します。

//rb1{漢字,かんじ//rb1}

傍点トリガは通常の強調と同様に、傍点を振りたい文字列をトリガで囲むだけです。

//bou1{傍点を振る文字列//bou1}

図表トリガグループ

図表トリガは、トリガに続いて番号を入力します。枝番号を付ける場合には、番号に続いて半角"-"を入力し、その後に半角英数字を1文字入力してください。

また、図表トリガには、"参照"と"定義"と呼ばれる役割りの違うトリガがあります。このトリガの違いを EWB では、次のように判断しています。

- ・図表トリガが行頭からはじまっていて、トリガに続く番号、もしくは枝番号を表す半角文字 の後にスペースがある場合は、図表トリガの定義と見なす。このスペースに続く文字列が、 図表のキャプションとなる。
- ・それ以外の本文などにある図表トリガは、図表の参照を表す。

参照部に相当する図表トリガの後に半角数字や半角"-"が続く場合には、トリガとの区別ができなくなるため、その後に半角スペースを1つ入力してください。定義部の前後にある空行は無視されるため、出力には影響しません。

//f101010 図の参照

//f101010 図のキャプション図の定義

```
//f101020-a
//f101020-b
図の参照
//f101020-a 図のキャプション ........図の定義
//f101020-b 図のキャプション
```

箇条書トリガグループ

箇条書トリガにも、開始を表すトリガと終了を表すトリガが存在します。開始を表すトリガは必ず行頭に入力してください。終了を表すトリガには、" //} "という省略形を使用することができます。また、このトリガの後には文字やその他のトリガは書けません。必ず、改行を1つ入力してください。

//k1{......開始を表すトリガ

- ・箇条書の文その1
- ・箇条書の文その2
- ・箇条書の文その3

//k1}.....終了を表すトリガ

箇条書項目の記号によっては、指定された問答に揃わない場合もあります。そのような場合には、"// "というトリガを使います。

//k1{......開始を表すトリガ (1)//|箇条書の文その1 (2)//|箇条書の文その2 (3)//|箇条書の文その3 //k1}......終了を表すトリガ

さらに、箇条書グループのトリガはグループトリガと同様に、改行自動挿入という機能を使って、箇条書の前後のアキを統一することができます。この指定がされている場合には、開始トリガの前の空行と終了トリガの後の空行が無視されます。

脚注トリガグループ

脚注トリガにも、図表トリガと同様に参照と定義があります。参照部は、脚注トリガに続いて番号を入力し、定義部は、開始を表すトリガと終了を表すトリガの間に入力します。開始を表すトリガは、必ず行頭に入力してください。また、終了を表すトリガには、"//}"という省略形が使えます。このトリガの後には、文字やその他のトリガは書けませんので、必ず改行を1つ入力してください。

脚注の文字列(内容)は、この定義部の中に記述するもので、トリガと番号の後にスペースを 空けてから入力します。この時、脚注は複数行になっても構いません。

> //ky100 //ky200 脚注の参照

//ky{......定義部の開始を表すトリガ //ky100 脚注の文字列 //ky200 脚注の文字列 //ky}.....定義部の終了を表すトリガ

非展開トリガグループ

強調トリガなどで "-5mm " や " fig_01.eps " といった文字列を T_EX で処理するパラメータとして T_EX に渡そうとするとこれらの文字列を EWB が "--5mm "" fig\us01.eps " という形で T_EX に渡 してしまいます。これをそのままの形で T_EX に渡すようにするのが非展開トリガです。

このトリガはグループトリガと同様に、開始を表すトリガと終了を表すトリガがあります。この時、終了を表すトリガは、"//}"という省略形が使えます。

//ICON{50mm,0mm,0,icon_01.eps//ICON}

ラベルトリガグループ、参照トリガグループ

ラベルトリガおよび参照トリガは、見出しや図表を別の場所から参照するのに使用します。見出しの直後にラベルトリガを書いておくと、別の場所からその見出しの番号、見出しの文字列、見出しのあるページ番号を参照トリガでひろうことができます。また、図表トリガを参照トリガで参照すると図表番号、図表のキャプション、図表のあるページ番号をひろうことができます。

強調トリガと同様に、開始、終了を表すトリガがあり、終了トリガには " // } " という省略形が使えます。ラベルにする文字列は英字および数字が使用できます。

ラベルのある箇所と参照する箇所は同じファイル中でなくともかまいません。図表番号は、文中に図表トリガをそのまま記述する方法でも参照できますが、その場合は同じ文書ファイル中か

らしか参照できません。

URL トリガグループ

URL トリガは文中の URL に対して使用します。このトリガで記述された箇所は、EWB の組版 結果を PDF へ変換するとハイパーリンクとして機能するようになります。

このトリガはグループトリガと同様に、開始を表すトリガと終了を表すトリガがあります。この時、終了を表すトリガは、"//}"という省略形が使えます。囲まれた部分は自動改行の対象とはならなくなるので使用時には注意が必要です。途中で改行させたい場合は、その箇所に"@@break()"や"@@hbreak()"といった組版トリガを入れる必要があります。

また、" http://www.abc.def... " の " // " の部分は EWB のトリガ開始文字列と同じなので、原稿では " //// " と記述しなければなりません。。

```
//url{http:///www.abc.def/ghi//url}
//url{http:////www.abc.def/@@break()ghi//url}
途中で改行させる場合
```

組み込みトリガ

組み込みトリガは、いままで説明したどのグループにも属さないものです。このトリガは、EWB に最初から組み込まれており、チューニングなどはできません。この組み込みトリガには、

・コメントトリガ

- ・索引トリガ
- ・インクルードトリガ
- ・LaTeX トリガ

の4つがあります。

コメントトリガは、プログラムトリガの中で使えるトリガです。プログラムリストに説明を付け加えるような場合に使用します。終了を表すトリガには"//}"という省略形も使えます。このトリガは、開始のトリガから終了のトリガまでを同一行内で記述してください。

索引トリガの入力は、図表トリガと同様にトリガに続いて番号を入力します。番号の後に半角数字が続く場合は、トリガと区別ができなくなってしまうため、半角スペースを1つ入力してください。なお、索引トリガは索引作成時のみに使用されるトリガで、本文の出力にはまったく影響しない唯一のトリガです。

```
//in100索引に入れたい文字列
```

残りの2つのトリガは、基礎編では説明しなかったものです。これらのトリガは、通常ではあまり使用されません。

まず、インクルードトリガですが、これは文書中に他のファイルを読み込む時に使います。このトリガは、行頭に記述してください。そして、トリガの後に半角スペースを入れ、読み込みたいファイル名を入力します。このファイル名は、"絶対パス名"か作業ディレクトリからの"相対パス名"を指定してください。

```
//inc ファイル名
```

また LaTeX トリガは、T_EX のコードを書くために用意されたものです。通常の入力では手に負えない数式などを表現したい時に利用します。入力規則は、強調トリガグループと同じです。

 $//LaTeX{T_{FX} \supset -F//LaTeX}$

2.4.4 編集トリガの強さ

"編集トリガの強さ"とは、前後に空行を持ったトリガが隣接してしまった場合に、出力にどのような影響があるのかということです。ここでは、具体的な例を見ながら説明していきましょう。

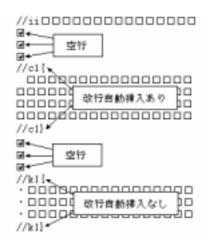


図 2-5 隣合うトリガ

図 2-5を見てみると、" //ii " と " //c1 { "の間、および " //c1 } " と " //k1 { "の間に空行があります。これらの空行は、出力に反映されるのでしょうか、それとも無視されてしまうのでしょうか。このような状況に遭遇すると、EWB では次のような考え方で処理を行っていきます。

- ・トリガに隣合う空行は、トリガが所有する
- ・トリガの所有する空行が出力されるか、無視されるかはトリガの性質によって決る
- ・トリガが隣合う場合、強い方のトリガが空行を所有する
- ・同じ強さのトリガが隣合う場合は、先に出てきたトリガが空行を所有する

そして、トリガの強さは、次のような順序になっています。

- 1. 見出しトリガ
- 2. 改行自動挿入が指定されていないグループトリガ、箇条書トリガ
- 3. 改行自動挿入が指定されているグループトリガ、箇条書トリガ、図表トリガ

この例では、見出しトリガが一番強いので、トリガの後にある空行は見出しが所有し、この空

行は消滅します。そして、小組と箇条書の間にある空行は、小組の方が強いため小組の所有となり、この空行は出力されることになります。このように、トリガの強さと出てくる順番によって、空行の扱いが違いますので注意が必要です。

2.4.5 トリガのネスト

脚注の中にプログラムリストを入れたり、小組の中に表組を入れたりと、各要素同士が混ざり合うような場面があります。このように、あるトリガの有効範囲内に他のトリガを入力することを、トリガのネストと呼んでいます。トリガのネストには規則が存在しており、すべてのトリガの中に、すべてのトリガがネストできるわけではありません。たとえば、箇条書の中にコメントトリガが入力できなかったり、グループトリガの範囲には見出しが書けなかったりといった具合いです。さらに、トリガ以外のタブ入力にもこのような制限があります。表 2-1は、トリガとタブのネストの可否を表しています。トリガ入力を行うときの参考にしてください。

2.4.6 編集トリガの入力のコツ

トリガ入力は、大変そうに見えますが、基本的な知識を理解してしまえば、恐れることはありません。ここでは、効率よくトリガが入力できるように、ワンポイントアドバイスをします。

使い慣れたワープロやエディタを使いましょう

EWB には、専用のエディタがありません。文字入力やトリガ入力には自分の使い慣れたワープロなどを使ってください。UNIX 上には"vi"や"emacs"などといったエディタがありますが、これらのエディタは UNIX 上でプログラムを書くために作られたものですから、必ずしも原稿の編集に向いてるとは限りません。UNIX に慣れていない方は、これらのエディタを無理に使って効率を下げるよりも、Windows 上の使い慣れたワープロやエディタを使いましょう。

ただし、1 つだけ注意しなければならないことがあります。"一太郎"や"Word"などといった 印刷機能を持っているワープロで作成した原稿ファイルには、印刷時に使用する特殊なコードが はいっている場合があります。EWB 用の原稿ファイルを作成する時には、必ず、"テキストファイル形式"で保存してください。また、UNIX へ転送後、tounix コマンドで文字コードを変換する ことを忘れないでください。

空行を入れよう

これまで説明したように、見出しや図表キャプションなどといったトリガの前後には、複数の 空行を入れることができます。ワープロ上で原稿を読みやすくするために、適切な場所に空行を 入れましょう。

番号の付け方

脚注や図表トリガの後に入力する番号は、原稿中で昇順になっている必要はないという説明をしました。しかし、昇順になっている方が、原稿がわかりやすく図表の管理が楽になります。また、後からトリガ挿入を行いやすいように、この番号を10番あるいは100番おきにしておいた方が便利です。さらに、統一した番号付けの方法を決めておきましょう。たとえば、「第2章3節の4番目の図を、//f0203040と表す」と仮に決めておけば、後で読んだ時にわかりやすくなります。

ネストするトリガ

				1	- X F 9	21.7	, ,,						
ネストされる トリガ	7 料服	44) Will	類類						**************************************	toles	かり	< **
見出し	×	×	×	×	×	×	×	×	×	×	×	×	×
グループ	×				×		×		×	×	×	×	×
表組	×				×		×		×	×	×	×	×
強調										×	×		×
図表 (参照)	×				×				×	×	×	×	×
図表 (定義)	×	*1			×		×		×	×	×	×	×
箇条書	×				×		×		×	×	×	×	×
脚注 (参照)	×				×		×	×	×	×	×	×	×
脚注 (定義)	×				×		×	×	×	×	×	×	×
プログラム	×			×	×		×			×	×	×	×
非展開										×	×		×
単純										×	×	×	×
インクルード	×				×		×	×	×	×	×	×	×
LaTeX	×								×	×	×	×	×
コメント	×	×	×	×	×	×	×	×		×	×	×	×
URL										×	×		×
索引										×	×	×	×
タブ	×				×	×				×	×	×	×

表 2-1 トリガ/タブのネスト

^{*1} 囲み罫を使用したグループの中ではフロートする図表は使用できません。また、ページ分割できる囲み罫はフロートする図表として使用できません。



体裁入力

この章では、作成する本の体裁情報を EWB システムに伝えるために行う"体裁入力"という作業について説明します。この体裁入力は従来、制作者がデザインカンプを使って編集者や制作プロダクションなどに紙面イメージを伝えていた作業を、コンピュータに対して行うものです。コンピュータはあまり融通が効かないため、「ちょっと空けて」とか「太い罫線を引いて」などといったあいまいな表現では伝わりません。しかし、体裁データを正しく入力しておけば、必ず正確な紙面イメージが得られ、効率アップを図ることができます。

3.1 **体裁入力の作業**

本を作成する時には、まず、デザインカンプを作成します。デザインカンプは、版面、柱やノンブルの位置、書体、級数などの紙面を形作るすべての要素の値が書き込まれています。EWB の体裁入力とは、このデザインカンプの値を"体裁入力ファイル"と呼ばれるファイルに書き写す作業なのです。

3.1.1 用途に合わせた体裁入力

1冊の本には前付け、本文、索引などといったように内容によって、体裁が異なるような場合、それぞれのデザインカンプを用意するものです。 EWB でも同様に、それぞれに対して体裁入力を行います。たとえば、前付けと本文の体裁が異なっていれば、別々の体裁入力ファイルを作成しなければならないわけです(図 3-1)。

また、索引の体裁には"索引項目とページ番号間のリーダー罫"などと、本文関連の体裁とは 多少異なる部分があります。そこで、前付けや本文などで使用するものを"本文用体裁入力ファ イル"と、索引で使用するものを"索引用体裁入力ファイル"と呼んで区別しています。

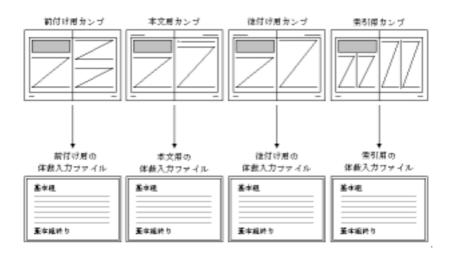


図 3-1 用途に合わせた体裁入力

3.1.2 体裁入力ファイルの作成

体裁入力ファイルは、すでに雛型が用意されているため、はじめから1つ1つ入力する必要はありません。この雛型は、標準的に EWB がインストールされていれば"/usr/local/ewb/3.3"というディレクトリに、"pssted.dat(本文用)"、"psidxsted.dat(索引用)"という名前で置いてあります。体裁入力する場合には、これらのファイルを作業ディレクトリにコピーし、必要な箇所の値を変更して、体裁入力ファイルの作成を行ってください。

また、体裁入力ファイルは特殊なものではなく、簡単なテキストファイルです。このファイルの入力には、UNIX上の vi や Windows上の使い慣れたワープロなどで行ってください。

Column:GUI による体裁編集

X Window System が使用できる環境であれば、体裁入力ファイルを作成するためのツールとして guisted というプログラムが用意されています。テキストエディタで体裁入力ファイルを作成するよりも楽に体裁が作成できます。

このプログラムは

guisted

で起動します。詳しくは「3.6 guistedによる体裁編集」を参照してください。

3.2 体裁入力ファイルの変換

体裁入力ファイルの作成が終ったら、そのファイルを EWB が理解できる形式に変換しなくてはなりません。実は、体裁入力ファイルと呼ばれるものは、EWB が直接使用するものではなく、体裁データをスムースに入力できるようにと用意されたファイルなのです。

3.2.1 スタイルエディタ

体裁入力ファイルを EWB が理解できる形式に変換するには、スタイルエディタと呼ばれるプログラムを使用します。このプログラムは、本文用と索引用の2つが用意されており、それぞれ "pssted"と "psidxsted"といいます。

本文用の pssted は、本文用体裁入力ファイルに基づいて、3 つのファイルを生成します(表 3-1)。まず、1 つ目は、"スタイルファイル"と呼ばれるファイルです。このファイルには、EWB が理解できる形式で判型や版面、そして各要素の体裁情報が収められており、組版時に用いられます。2 つ目は、"トリガ定義ファイル"と呼ばれるファイルです。このファイルは、原稿中に埋め込まれたトリガの意味を EWB が解釈するために使われます。3 つ目は、PDF 用リンク箇所の色等を設定する"PDF スタイルファイル"です。PDF スタイルファイルは PDF 用 PS ファイルを生成するときのみ使用されます。

また、複数の体裁入力を行う場合は、スタイルファイルの名前を別々に管理してください。標準では、"ewb2latex.cls"という名前で生成されますが、本文以外にもさまざまなスタイルファイルを作成した時には、"preface.cls"や"appendix.cls"などといったわかりやすい名前にしましょう。

ファイル	通常の名前	用途
スタイルファイル	ewb2latex.cls	本の体裁を EWB に伝える
トリガ定義ファイル	triger.def	原稿に入力したトリガの意味を解釈する時に使われる
PDF スタイルファイル	PDFstyle	PDF 用 PS ファイル生成時に、リンク箇所の色等を指定する

表 3-1 pssted によって生成されるファイル

さらに、索引用のpsidxsted は、索引用体裁入力ファイルに基づいて、2 つのファイルを生成します(表 3-2)。 やはり、pssted と同様に、EWB が理解できる形式の"スタイルファイル"が作成されます。もう 1 つのファイルは、"インデックススタイル"と呼ばれるファイルです。このファイルは索引作成時に行うソーティングなどに用いる情報ファイルで、直接には関係がありません。

ファイル	通常の名前	用途
スタイルファイル	index.cls	索引ページの体裁を EWB に伝える
インデックススタイル	indexsty	EWB システムが内部で使用するファイル

表 3-2 psidxsted によって生成されるファイル

3.2.2 pssted と psidxsted の使い方

それでは、具体的な生成方法を見ていくことにしましょう。

まず、本文用体裁入力ファイルを作成したら、pssted と呼ばれるプログラムを使用して、EWB が理解できるファイルに変換します。この pssted の使い方は、次のようになります。

```
pssted [-h] [-s] [-t] [-p] 体裁入力ファイル
```

```
オプション説明-hヘルプ。簡単な使い方が表示されます-sスタイルファイルを出力しません-tトリガ定義ファイルを出力しません-pPDFスタイルファイルを出力しません
```

pssted は、次に示すように本文用体裁入力ファイルを指定して起動します。

```
% pssted pssted.dat
% ls
PDFstyle ewb2latex.cls pssted.dat triger.def
%
```

本文用体裁入力ファイルに、記述的な誤りがなければ、画面上には何も表示されずに、UNIXのプロンプトが返ってきます。これで、必要なファイルが生成されているはずです。もし、すでに pssted によって、生成されたファイルが作業ディレクトリにある場合には、古いファイルを "ewb2latex.cls" などのようにファイル名の最後に "~"を付けて保存します。

```
% ls
PDFstyle ewb2latex.cls pssted.dat triger.def
% pssted pssted.dat
% ls
PDFstyle ewb2latex.cls pssted.dat PDFstyle~
triger.def~ ewb2latex.cls~ triger.def
%
```

また、索引に関しても、本文と同じようなやり方で行います。索引用体裁入力ファイルを作成したら、psidxsted というプログラムを使用して、EWB が理解できるファイルに変換します。この

psidxsted には、pssted のような起動時オプションはありません。次のように索引用体裁入力ファイルを指定して起動します。

```
% psidxsted psidxsted.dat
% ls
psidxsted.dat index.cls indexsty
%
```

索引用体裁入力ファイルに、記述的な誤りがなければ、画面上には何も表示されずに、プロンプトが返ってきます。psidxsted も pssted と同様に、すでに古いファイルが存在する場合には、ファイル名の最後に "~"を付けて保存します。

```
% ls
psidxsted.dat index.cls indexsty
% psidxsted psidxsted.dat
% ls
psidxsted.dat index.cls indexsty
index.cls indexsty
%
```

3.3 体裁入力ファイルの基礎

ここからは、具体的に本文用体裁入力ファイルと索引用体裁入力ファイルの作成方法を解説していきます。まずは、これらのファイル形式や基本的な指定項目を取り上げてみましょう。

3.3.1 体裁入力ファイルの形式

体裁入力ファイルは、見出しや箇条書といった各要素の体裁が順番に並んでいる、簡単なテキストファイルです。たとえば、1つの箇条書の体裁は、"箇条書"と"箇条書終り"という文字列に囲まれたプロック内に、級数や書体などの項目が、1行ずつ記述されています。各行の先頭に指定項目名があり、それに続いて項目の値を入力するようになっています。それぞれの項目は、"スペース"あるいは"タブ"で区切ります。もし、項目にスペースを入力したい場合には、ダブルクォーテーションを用いて文字列を囲んでください。また、コメントを記述したい場合には、"#"を使ってください。この記号から、行末までがコメントとして扱われます。

3.3.2 体裁入力ファイルで使用される変数

見出しや図表などの番号付けの方法を指定する際に、"%ii 番号%"などといった項目を指定することがあります。体裁入力ファイルでは、"%"で囲まれた文字列を変数として扱っています。これらの変数は、常に見出し番号や図表番号などを保持しており、これらの変数の組み合せでさまざまな番号付けができるように工夫されています。

3.3.3 書体などの基本的な指定

書体や級数などは、ほとんどの要素で指定しなければならない基本的なものです。

```
書体
                Ryumin-Light
                          # 和文書体
変形
                正体
                     # 指定項目:
                正体/かな/
                長体1/長体2/長体3/長体4/
          #
          #
                平体1/平体2/平体3/平体4/
                斜体1/斜体2/斜体3
                     # 単位:Q/pt
級数
                130
                     # 単位:歯/pt/mm
字误 ()
                13歯
強調書体
                               # 和文強調書体
                GothicBBB-Medium
                     # 指定項目:
強調変形
                正体
          #
                正体/かな/
                長体1/長体2/長体3/長体4/
                平体1/平体2/平体3/平体4/
          #
          #
                斜体1/斜体2/斜体3
強調級数
                     # 単位:Q/pt
                130
欧文書体
                          # 欧文書体
                PCTimes-Roman
欧文級数
                     # 単位:Q/pt
                130
欧文ベースライン
                0字
                     # 単位:字/歯/pt/mm
          # 値を大きくするほど欧文が下がります
文字色シアン
                     # CMYKでシアン成分を0.0~1.0で指定
               0.0
文字色マゼンタ
                0.0
                     # CMYKでマゼンタ成分を0.0~1.0で指定
文字色イエロー
                0 0
                     # CMYKでイエロー成分を0.0~1.0で指定
文字色ブラック
                     # CMYKでブラック成分を0.0~1.0で指定
                1.0
欧文強調書体
                          # 欧文強調書体
               PCHelvetica
欧文強調級数
               13Q # 単位:Q/pt
欧文強調ベースライン 0字
                     # 単位:字/歯/pt/mm
          # 値を大きくするほど欧文が下がります
強調文字色シアン
               0.0 # CMYKでシアン成分を0.0~1.0で指定
強調文字色マゼンタ
               0.0
                    # CMYKでマゼンタ成分を0.0~1.0で指定
強調文字色イエロー
               0.0
                    # CMYKでイエロー成分を0.0~1.0で指定
強調文字色ブラック
                    # CMYKでブラック成分を0.0~1.0で指定
               1 0
```

書体

和文書体を指定します。"Ryumin-Light"などの書体コード名を入力してください。指定可能な和文書体は、Appendix D に掲載してあります。参考にしてください。

变形

和文書体の変形を指定します。変形しない場合は正体を指定します。長体 1 ~ 長体 4 を指定した場合は $10\% \sim 40\%$ だけ字幅を細くします。平体 1 ~ 平体 4 を指定した場合は $10\% \sim 40\%$ だけ高さを低くします。

級数

和文の級数を指定します。単位は級数 (Q) / ポイント (pt) が使用できます。

字送り

和文文字の字送りを指定します。単位は歯数 (歯) / ポイント (pt) / ミリ (mm) が使用できます。

欧文書体

欧文書体を指定します。"Times-Roman"などの書体コード名を入力してください。指定可能な欧文書体は、Appendix D に掲載してあります。参考にしてください。

欧文級数

欧文の級数を和文とは別に指定できます。

欧文ベースライン

和文と欧文のベースライン補正値が指定できます。単位は字/歯数(歯)/ポイント(pt)/ミリ(mm)が使用できます。単位を字で指定した場合は、全角文字の高さを"1"として、小数で指定します。横組の場合は、大きな値を入力するほど欧文が下にずれ、縦組の場合は、大きな値ほど欧文が左にずれます。

強調書体

原稿中で、" //g{ " と " //g} " で囲まれた部分の書体や級数などを指定します。基本組や小組などそれぞれの要素に対して、強調書体を指定できます。

文字色

文字の色を CMYK 形式の数値で指定します。数値は 0.0~1.0 の範囲です。

3.3.4 行送りなどの指定

行送りや字上げ、字下げなどの指定は、縦や横方向のアキを決める重要な体裁です。

字詰め 43字 # 単位:字 行送り 24歯 # 単位:歯/pt/mm 字下げ 0字 # 単位:字/歯/pt/mm 字上げ 0字 # 単位:字/歯/pt/mm 問答 3字 # 単位:字/歯/pt/mm

字詰め

1 行の文字数を指定します。この項目があるのは基本組と見出しのみです。

行送り

行送りを指定します。単位は歯数 (歯) / ポイント (pt) / ミリ (mm) が使用できます。

字下げ

版面に対する字下げ (左アキ) を指定します。単位は字数 (字) / 歯数 (歯) / ポイント (pt) / ミリ (mm) が使用できます。字数で指定した場合は、基本組の字送りの倍数になります。

字上げ

版面に対する字上げ (右アキ) を指定します。単位は字数 (字) / 歯数 (歯) / ポイント (pt) / ミリ (mm) が使用できます。字数で指定した場合は、基本組の字送りの倍数になります。

問答

箇条書や見出しなどにある指定です。行が複数に分かれてしまった時に、2 行目以降の字下げ量を指定します。単位は字数 (字) / 歯数 (歯) / ポイント (pt) / ミリ (mm) が使用できます。

3.3.5 改行自動挿入

小組や箇条書などの前後に、統一的なアキを自動的に入れる指定です。

半改行自動挿入 有り # 指定項目:有り/無し

改行挿入 0.5行 # 単位:行

改行挿入

小組や箇条書などの前後に統一的なアキを挿入できます。

この指定には、"半改行自動挿入"と"改行挿入"の2つがあります。

半改行自動挿入を指定する場合は、"有り"あるいは"無し"という文字列を入力します。半改行自動挿入を"有り"にすると、原稿中の開始トリガの前と終了トリガの後に入力した改行が無視されて、指定されたアキが必ず出力されます。半改行自動挿入を"無し"にすると、原稿中の開始トリガの前と終了トリガの後に入力した改行が出力時に反映されます。上の例のように、改行挿入の項目を"#"によってコメントアウトしてある(改行挿入を指定していない)場合は、改行挿入として自動的に 0.5 行分の空きが前後に入ります。改行挿入を指定した場合には、その行数分の空きが前後に入ります。

3.3.6 禁則/和欧文間スペース

組版規則(文字組み)に対する指定です。禁則の制御と和欧文間のアキを指定することができます。

禁則

禁則解除 無し # 指定項目:有り/無し 和欧文間スペース 有り # 指定項目:有り/無し

禁則終り

禁則解除

拗促音や音引き(つ、ャ、ーなど)を、行頭禁則の対象にするかどうかを指定します。"有り"を指定すると、拗促音や音引きを禁則対象文字からはずします。"無し"を指定すると、これらを禁則対象文字にします。

和欧文間スペース

和欧文間に対して、自動的に四分アキを入れるかどうか指定します。"有り"を指定すると四分アキがはいります。"無し"を指定すると四分アキを入れません。

3.3.7 ページサイズ

判型の大きさや版面までのマージンを指定します。

頁サイズ

仕上がりサイズ縦257mm# 単位:mm/歯/pt仕上がりサイズ横182mm# 単位:mm/歯/ptトップマージン24.75mm# 単位:mm/歯/ptのど側のマージン23mm# 単位:mm/歯/pt

頁サイズ終り

仕上がりサイズ縦

仕上がリサイズの縦の長さを指定します。単位はミリ (mm) / 歯数 (歯) / ポイント (pt) が使用できます。

仕上がりサイズ横

仕上がリサイズの横の長さを指定します。単位はミリ (mm) / 歯数 (歯) / ポイント (pt)\ が使用できます。

トップマージン

天側のアキを指定します。単位はミリ (mm) / 歯数 (歯) / ポイント (pt) が使用できます。地側のアキは、この"トップマージン"と基本組で設定される版面のサイズから、自動的に計算されます。

のど側のマージン

のど側のアキを指定します。単位はミリ (mm) / 歯数 (歯) / ポイント (pt) が使用できます。

小口側のアキは、この"のど側のマージン"と基本組で設定される版面のサイズから、自動的に計算されます。

3.3.8 ページ体裁

版面の外側にある柱やノンブルの体裁、位置、書体などの指定を行います。左ページおよび右ページを、それぞれ別に設定します(図 3-2)。

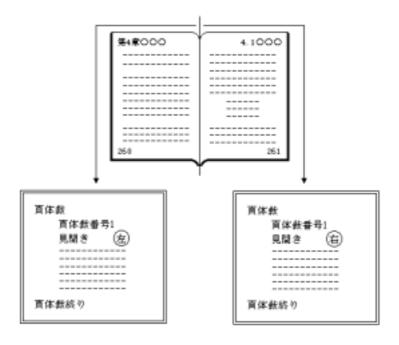


図 3-2 左右ページの体裁

また、ページ体裁は1冊の本に対して、複数の体裁が指定できます。これらの体裁は、"頁体裁番号"と呼ばれる数字で区別されるようになっています。この頁体裁番号は、見出しで参照され、各見出しごとに柱やノンブルの体裁を変えることができるようになっています。

```
    頁体裁
    1

    頁牌き
    左
    # 指定項目:左/右

    柱原点
    左上
    # 指定項目:

    性原点
    左上/右上

    # 左上/右上
    # 左下/右下

    柱オフセット横
    15mm
    # 単位:mm/歯/pt

    柱オフセット縦
    14mm
    # 単位:mm/歯/pt
```

```
柱行揃え
                 左
                           # 指定項目:左/中/右
                            # 和文書体
  柱書体
                 GothicBBB-Medium
  柱級数
                 11Q # 単位:Q/pt
  柱欧文書体
                 Helvetica
                           # 欧文書体
                           # 単位:Q
  柱欧文級数
                 110
  柱ベースライン
                 0.03字
                           # 単位:字/歯/pt/mm
              # ベースライン補正値
               # 大きな値にするほど欧文が
               # 下がります。
                     # CMYKでシアン成分を0.0~1.0で指定
  柱文字色シアン
                 0.0
  柱文字色マゼンタ
                 0.0
                     # CMYKでマゼンタ成分を0.0~1.0で指定
  柱文字色イエロー
                0.0
                     # CMYKでイエロー成分を0.0~1.0で指定
                     # CMYKでブラック成分を0.0~1.0で指定
  柱文字色ブラック
                1.0
  柱字送り
                 11歯
                           # 単位:歯/pt/mm
  柱文字列
                 "%ii番号% %ii文字列%"
  柱修飾
                 # 指定項目:罫線/EPS
        要素
           EPS
        オフセット横 15.0mm # 単位:mm/歯/pt
        オフセット縦 16.0mm # 単位:mm/歯/pt
        EPSファイル名 hashira1.eps
        EPS幅 0.0mm
                 # 単位:mm/歯/pt
        EPS高さ 0.0mm
                 # 単位:mm/歯/pt
        EPS倍率 10
                 # 倍率をパーセントで指定
  柱修飾終り
  ノンブル原点
                           # 指定項目:
                 左下
                   左上/右上
               #
                   左下/右下
               #
  ノンブルオフセット横 15mm
                           # 単位:mm/歯/pt
  ノンブルオフセット縦
                 13.5mm
                          # 単位:mm/歯/pt
  ノンブル行揃え
                          # 指定項目:左/中/右
                左
  ノンブル書体
                 11.11
                          # 和文書体
  ノンブル級数
                12Q
                          # 単位:Q/pt
  ノンブル欧文書体
                Times-Roman
                         # 欧文書体
  ノンブル欧文級数
                120
                          # 単位:Q/pt
                 0.03字
  ノンブルベースライン
                           # 単位:字/歯/pt/mm
              # ベースライン補正値
                    # CMYKでシアン成分を0.0~1.0で指定
  ノンブル文字色シアン
                 0.0
                     # CMYKでマゼンタ成分を0.0~1.0で指定
  ノンブル文字色マゼンタ 0.0
  ノンブル文字色イエロー 0.0 # CMYKでイエロー成分を0.0~1.0で指定
  ノンブル文字色ブラック 1.0
                     # CMYKでブラック成分を0.0~1.0で指定
  ノンブル字送り
                 12 歯
                           # 単位:歯/pt/mm
  ノンブル文字列
                 "%頁%"
  ノンブル修飾
            罫線 # 指定項目:罫線/EPS
       要素
       オフセット横 10.0mm # 単位:mm/歯/pt
       オフセット縦 10.0mm #単位:mm/歯/pt
       罫線幅 50.0mm # 単位:mm/歯/pt
                 1.0mm # 単位:mm/歯/pt
       罫線高さ
                     # CMYKでシアン成分を0.0~1.0で指定
       罫線色シアン
                 0.00
       罫線色マゼンタ 0.00
                     # CMYKでシアン成分を0.0~1.0で指定
       罫線色イエロー 0.00
                     # CMYKでシアン成分を0.0~1.0で指定
       罫線色ブラック 1.00
                     # CMYKでシアン成分を0.0~1.0で指定
  ノンブル修飾終り
頁体裁終り
```

頁体裁番号

頁体裁番号を指定します。

見開き

右ページに対しての指定なのか、左ページに対しての指定なのかを入力します。どちらかのページに柱やノンブルが付かない場合でも、必ず両者の体裁が必要です。

柱 / ノンブル原点

柱やノンブルの付く位置を指定するための原点を決めます。"右上、右下、左上、左下"のいずれかを指定します。

柱 / ノンブルオフセット

柱やノンブルの付く位置を指定します。原点からの縦と横の長さを指定してください。単位はミリ (mm) / 歯数 (歯) / ポイント (pt) が使用できます。

柱/ノンブル行揃え

柱やノンブルが付く位置に対して、文字列の揃え方を指定します。" 左、中、右 " のいずれかを指定してください。

柱文字列

柱文字列の出力方法を指定します。"%<見出しトリガ>番号%(トリガの見出し番号を出力)%<見出しトリガ>文字列%(トリガの見出し文字列を出力)"のいずれかの変数を入力してください。この変数や固定文字列を記述する時は、必ず、半角のダブルクォーテーションで囲んでください。たとえば、"%ii 番号%%ii 文字列%"といったようにです。もし、柱を出力したくない場合は、""と入力してください。

ノンブル文字列

ノンブル文字列の出力方法を指定します。"%頁%(ページ番号を出力)"という変数や固定文字列などを入力してください。この文字列は、必ず、半角のダブルクォーテーションで囲んでください。たとえば、"-%頁%-"といった具合いです。もし、ノンブルを出力したくない場合は、""と入力してください。

柱 / ノンブル修飾

柱またはノンブルに飾りを付けます。飾りとして罫線または EPS 画像が使用できます。飾りを使わない場合はこの項目以下は必要ありません。

要素

柱/ノンブルの修飾要素を罫線にするか EPS 画像にするかを指定します。

オフセット

修飾要素の位置を原点からの距離で指定します。罫線の場合は罫線の基準位置は柱/ノンブルの原点と同じで、原点が右上なら罫線の右上までの距離になります。EPS 画像の場合、画像の基準位置は左右は柱/ノンブルの原点と同じですが、上下は原点の上下に関わらず画像の下までの距離になります。

罫線幅/高さ

修飾要素を罫線にした場合に、罫線のサイズを指定します。

罫線色

修飾要素を罫線にした場合に、罫線色を CMYK で指定します。

EPS ファイル名

修飾要素を EPS にした場合に、EPS ファイル名を指定します。

EPS幅/高さ/倍率

修飾要素を EPS にした場合に、EPS 画像のサイズを指定します。倍率が 0 でない場合は倍率が優先されます。倍率が 0 の場合は、幅と高さで指定されたサイズで出力されます。幅または高さが 0 の場合、縦横比を保ったまま 0 でない方のサイズに合わせて拡大 / 縮小されます。いずれも 0 の場合は、原寸大で出力されます。

3.4 本文用体裁入力ファイルの詳細

ここでは、本文用体裁入力ファイルの内容を、指定項目別に細かく見ていくことにしましょう。

3.4.1 その他

体裁指定にはあまり関係しませんが、さまざまな初期設定値を入力する部分です。

その他 書名 "pssted sample file" 組方向 構 # 指定

組方向 横 # 指定項目:縦/横 ページスタイル 2 # デフォルトの頁体裁番号

トリガ定義ファイル ./triger.def スタイルファイル ./ewb2latex.cls PDFスタイルファイル ./PDFstyle

その他終り

書名

作成する本の書名を入力してください。書名にスペースが含まれる場合には、半角のダブルクォーテーションで囲ってください。

組方向

縦組および横組を選択します。"縦、横"のいずれかを入力してください。

ページ番号

最初に出力されるページ番号を設定します。なお、ページ番号は組版時にも設定できます。

ページスタイル

通常のページに使うページスタイルの"頁体裁番号"を指定します。

トリガ定義文字

トリガ開始文字列を定義します。通常は、" // " にしておきます (" // " 以外の文字列は完全にはサポートされていません)。

章定義文字

章番号を指定する文字列を設定します。通常は、"chapter"にしておきます。

トリガ定義ファイル

pssted によって生成される"トリガ定義ファイル"の名前を設定します。

スタイルファイル

pssted によって生成される"スタイルファイル"の名前を設定します。1つの作業ディレクトリで、複数の体裁入力を行う場合には、この名前を変更してください。

ベーススタイル

基本となる EWB のベーススタイルファイルを設定します。通常は、"ewbbase3_3.sty"にしておきます。

3.4.2 基本組

本の基本的な組を指定します。ここで入力する値によって、版面の大きさが決定されます。基本組は、必ず入力してください。

```
基本組
  書体
                   Ryumin-Light
                              # 和文書体
  变形
                   正体
                         # 指定項目:
                   正体/かな/
             #
                   長体1/長体2/長体3/長体4/
             #
                   平体1/平体2/平体3/平体4/
                   斜体1/斜体2/斜体3
  級数
                         # 単位:Q/pt
                   130
  字送り
                         # 単位:歯/pt/mm
                   13歯
                                    # 和文強調書体
  強調書体
                   GothicBBB-Medium
  強調変形
                   正体
                         # 指定項目:
             #
                   正体/かな/
             #
                   長体1/長体2/長体3/長体4/
                   平体1/平体2/平体3/平体4/
             #
                   斜体1/斜体2/斜体3
  強調級数
                         # 単位:Q/pt
                   13Q
                         # 単位:歯/pt/mm
  強調字送り
                   13歯
  欧文書体
                   PCTimes-Roman
                              # 欧文書体
  欧文級数
                   13Q
                         # 単位:Q/pt
  欧文ベースライン
                   0字
                         # 単位:字/歯/pt/mm
             # 値を大きくするほど欧文が下がります
  欧文強調書体
                              # 欧文強調書体
                   PCHelvetica
                        # 単位:Q/pt
  欧文強調級数
                   13Q
  欧文強調ベースライン
                         # 単位:字/歯/pt/mm
                   0字
```

```
# 値を大きくするほど欧文が下がります
  強調文字色シアン
                       # CMYKでシアン成分を0.0~1.0で指定
                  0.0
  強調文字色マゼンタ
                       # CMYKでマゼンタ成分を0.0~1.0で指定
                  0.0
                       # CMYKでイエロー成分を0.0~1.0で指定
  強調文字色イエロー
                  0.0
                       # CMYKでブラック成分を0.0~1.0で指定
  強調文字色ブラック
                  1.0
  字詰め
                  43字
                      # 単位:字
                      # 単位:歯/pt/mm
  行送り
                  24歯
                      # 単位:字/歯/pt/mm
  字下げ
                  0字
  字上げ
                  0字
                      # 単位:字/歯/pt/mm
  行数
                     #1ページの行数
                  35行
基本組終り
```

版面の大きさ

ここで入力した項目の値によって、版面のサイズが決ります。EWB では、版面の大きさの計算は、次のような式で行われます。

版面の横のサイズ = (字下げ+字詰め+字上げ-1)×字送り+和文級数 版面の縦のサイズ = (行数-1)×行送り+和文級数

3.4.3 小組

小組は、必要な数だけ作ることができます。小組を使わない場合は、ファイルから削除しても 構いません。

```
小組
  トリガ
                   c1
  書体
                   Ryumin-Light # 和文書体
  变形
                   正体 # 指定項目:
                   正体/かな/
             #
                   長体1/長体2/長体3/長体4/
             #
                   平体1/平体2/平体3/平体4/
             #
                   斜体1/斜体2/斜体3
             #
  級数
                         # 単位:Q
                   13Q
                         # 単位:歯/pt/mm
  字送り
                   13歯
  強調書体
                                   # 和文強調書体
                   GothicBBB-Medium
  強調変形
                   正体
                         # 指定項目:
             #
                   正体/かな/
                   長体1/長体2/長体3/長体4/
             #
                   平体1/平体2/平体3/平体4/
             #
                   斜体1/斜体2/斜体3
  強調級数
                         # 単位:Q
  強調字送り
                         # 単位:歯/pt/mm
                   13歯
  欧文書体
                   PCTimes-Roman # 欧文書体
  欧文級数
                        # 単位:Q
                   13Q
  欧文ベースライン
                   0字
                        # 単位:字/歯/pt/mm
```

```
# 値を大きくするほど欧文が下がります
  文字色シアン
                      # CMYKでシアン成分を0.0~1.0で指定
                 0.0
                      # CMYKでマゼンタ成分を0.0~1.0で指定
  文字色マゼンタ
                 0.0
  文字色イエロー
                 0.0
                      # CMYKでイエロー成分を0.0~1.0で指定
  文字色ブラック
                 1.0
                      # CMYKでブラック成分を0.0~1.0で指定
  欧文強調書体
                           # 欧文強調書体
                 PCHelvetica
                    # 単位:Q/pt
  欧文強調級数
                 13Q
  欧文強調ベースライン
                 0字
                      # 単位:字/歯/pt/mm
           # 値を大きくするほど欧文が下がります
  強調文字色シアン
                0.0 # CMYKでシアン成分を0.0~1.0で指定
  強調文字色マゼンタ
                     # CMYKでマゼンタ成分を0.0~1.0で指定
                 0.0
  強調文字色イエロー
                 0.0
                     # CMYKでイエロー成分を0.0~1.0で指定
  強調文字色ブラック
                 1.0
                     # CMYKでブラック成分を0.0~1.0で指定
  行送り
                 22歯 # 単位:歯/pt/mm
  字下げ
                 3字
                     # 単位:字/歯/pt/mm
  字上げ
                 0字
                     # 単位:字/歯/pt/mm
  半改行自動挿入
                 有り # 指定項目:有り/無し
                 0.5行 # 単位:行/歯/pt/mm
  改行挿入
                 ₩1 #指定項目:囲み罫トリガ名/無し
  囲み罫
小組終り
```

トリガ

小組のトリガを定義します。この例のように、"c1"と指定すると、開始のトリガが"//c1{"に、終了のトリガが"//c1}"ということになります。

囲み罫

小組を罫線で囲んで出力する場合に指定します。" 囲み罫 " で作られるトリガ名を指定しておくと、小組を使用した場合に自動的に指定された囲み罫も出力されます。

3.4.4 表組

表組は、必要な数だけ作ることができます。表組を使わない場合は、ファイルから削除しても 構いません。

```
表組
   トリガ
                    table
  書体
                    Ryumin-Light
                                # 和文書体
   变形
                    正体
                         # 指定項目:
                    正体/かな/
              #
                    長体1/長体2/長体3/長体4/
                    平体1/平体2/平体3/平体4/
              #
                    斜体1/斜体2/斜体3
                          # 単位:Q/pt
  級数
                    120
  字送り
                    12歯
                          # 単位:歯/pt/mm
  強調書体
                    GothicBBB-Medium
                                      # 和文強調書体
```

```
強調変形
                  正体
                       # 指定項目:
                 正体/かな/
            #
                  長体1/長体2/長体3/長体4/
            #
                  平体1/平体2/平体3/平体4/
            #
            #
                  斜体1/斜体2/斜体3
  強調級数
                       # 単位:Q/pt
                 120
                       # 単位:歯/pt/mm
  強調字送り
                 12歯
  欧文書体
                 PCTimes-Roman # 欧文書体
  欧文級数
                 120
                       # 単位:Q/pt
                       # 単位:字/歯/pt/mm
  欧文ベースライン
                 0字
            # 値を大きくするほど欧文が下がります
  文字色シアン
                 0.0
                      # CMYKでシアン成分を0.0~1.0で指定
  文字色マゼンタ
                       # CMYKでマゼンタ成分を0.0~1.0で指定
                 0.0
  文字色イエロー
                      # CMYKでイエロー成分を0.0~1.0で指定
                 0.0
  文字色ブラック
                 1.0
                      # CMYKでブラック成分を0.0~1.0で指定
  欧文強調書体
                 PCHelvetica
                            # 欧文强調書体
  欧文強調級数
                      # 単位:Q/pt
                 120
  欧文強調ベースライン 0字
                      # 単位:字/歯/pt/mm
            # 値を大きくするほど欧文が下がります
  強調文字色シアン
                      # CMYKでシアン成分を0.0~1.0で指定
                 0.0
                       # CMYKでマゼンタ成分を0.0~1.0で指定
  強調文字色マゼンタ
                 0.0
  強調文字色イエロー
                      # CMYKでイエロー成分を0.0~1.0で指定
                 0.0
  強調文字色ブラック
                      # CMYKでブラック成分を0.0~1.0で指定
                 1.0
                 20歯
  行送り
                     # 単位:歯/pt/mm
  字下げ
                      # 単位:字/歯/pt/mm
                 1字
  字上げ
                 1字
                      # 単位:字/歯/pt/mm
  半改行自動挿入
                 有り
                      # 指定項目:有り/無し
  改行挿入
                 0.5行 # 単位:行/歯/pt/mm
  タブ
                      # タブをいくつのスペースに展開するか
                 2
  項目内行送り
                 16歯 #単位:歯 項目の中の行送り
                 12歯 #単位:歯 項目と項目の横の空き
  項目間隔
  用み罫
                 無し
                      # 指定項目:囲み罫トリガ名/無し
表組終り
```

トリガ

表組のトリガを定義します。この例のように"table"と指定すると、開始のトリガが"/table{ "に、終了のトリガが"/table}"ということになります。

タブ

表中で使用する"タブストップ"の値を入力します。ワープロで設定した値と同じにしましょう。

項目内行送り

表中の項目内で行が複数になった場合に使用する行送りの値を入力します。単位は歯数 (歯) のみです。

項目間隔

表の項目と項目の間に対して、最小のアキを設定します。単位は歯数(歯)のみです。

3.4.5 脚注

通常、脚注は1つだけ指定します。脚注を使わない場合は、ファイルから削除しても構いません。

```
脚注
  トリガ
                  ky
                  Ryumin-Light # 和文書体
  書体
                  正体 # 指定項目:
  变形
                  正体/かな/
                  長体1/長体2/長体3/長体4/
             #
                  平体1/平体2/平体3/平体4/
             #
                  斜体1/斜体2/斜体3
             #
  級数
                  11Q
                       # 単位:Q/pt
  字送り
                  11歯
                       # 単位:歯/pt/mm
  強調書体
                  GothicBBB-Medium
                               # 和文強調書体
  強調变形
                  正体
                      # 指定項目:
                  正体/かな/
             #
                  長体1/長体2/長体3/長体4/
             #
                  平体1/平体2/平体3/平体4/
             #
                  斜体1/斜体2/斜体3
  強調級数
                  11Q
                      # 単位:Q/pt
  強調字送り
                  11歯
                      # 単位:歯/pt/mm
  欧文書体
                  PCTimes-Roman # 欧文書体
                      # 単位:Q/pt
  欧文級数
                  110
                      # 単位:字/歯/pt/mm
  欧文ベースライン
                 0字
            #値を大きくするほど欧文が下がります
                      # CMYKでシアン成分を0.0~1.0で指定
                  0.0
  文字色シアン
                       # CMYKでマゼンタ成分を0.0~1.0で指定
  文字色マゼンタ
                  0.0
  文字色イエロー
                  0.0
                       # CMYKでイエロー成分を0.0~1.0で指定
                      # CMYKでブラック成分を0.0~1.0で指定
  文字色ブラック
                  1.0
  欧文強調書体
                             # 欧文強調書体
                  PCHelvetica
  欧文強調級数 11Q
欧文強調ベースライン 0字
                     # 単位:Q/pt
                       # 単位:字/歯/pt/mm
            #値を大きくするほど欧文が下がります
  強調文字色シアン
                0.0 # CMYKでシアン成分を0.0~1.0で指定
  強調文字色マゼンタ
                  0.0
                      # CMYKでマゼンタ成分を0.0~1.0で指定
  強調文字色イエロー
                  0.0
                      # CMYKでイエロー成分を0.0~1.0で指定
  強調文字色ブラック
                  1.0
                      # CMYKでブラック成分を0.0~1.0で指定
                  16歯 # 単位:歯/pt/mm
  行送り
  字下げ
                  0字
                      # 単位:字/歯/pt/mm
                       # 単位:字/歯/pt/mm
  字上げ
                  0字
  問答
                  3字
                       # 単位:字/歯/pt/mm
  上付き文字
                  "*%章内番号%"
                  "注%章内番号%
  脚注文字列
  シンボル
                  arabic # arabic:数字
             # roman :小文字ローマ数字
             # Roman :大文字ローマ数字
             # alph :小文字英字
                  :大文字英字
             # Alph
                  0.1mm # 単位:mm/歯/pt
  罫線太さ
                  24歯 # 単位:歯/pt/mm
  罫線本文空き
                      # 単位:歯/pt/mm
  罫線脚注空き
                  9歯
  脚注脚注空き
                  7歯
                      # 単位:歯/pt/mm
脚注終り
```

トリガ

脚注のトリガを定義します。この例のように"ky"と指定すると、脚注部分の開始トリガが"//ky{"に、終了トリガが"//ky}"になります。また、定義(脚注の内容を記述する)と参照のトリガは、"//ky番号"になります。

上付き文字

本文中で、脚注を参照する時に使用する上付き文字の出力方法を指定します。"%章内番号%(章ごとに番号付け)%節内番号%(節ごとに番号付け)"という2つの変数が使用できます。これらの変数や固定文字列を記述する時は、必ず、半角のダブルクォーテーションで囲んでください。もし、上付き文字を出力したくない場合は、""と入力してください。

脚注文字列

脚注内容の先頭に付く脚注番号の出力方法を指定します。この文字列の指定の方法は、"上付き文字"と同様です。

シンボル

脚注の番号付けに使う文字種を指定します。"arabic (数字)、roman (小文字ローマ数字)、Roman (大文字ローマ数字)、alph (小文字英字)、Alph (大文字英字)"のいずれかを指定してください。

罫線太さ

ページ末に出力される脚注罫線の太さを指定します。単位はミリ (mm) / 歯数 (歯) / ポイント (pt) が使用できます。

罫線本文空き

本文と脚注罫線との最小のアキを指定します。単位は歯数 (歯) / ポイント (pt) / ミリ (mm) が使用できます。

罫線脚注空き

罫線と脚注のアキを指定します。単位は歯数 (歯) / ポイント (pt) / ミリ (mm) が使用できます。

脚注脚注空き

複数の脚注が並んだ時のアキを指定します。単位は歯数 (歯) / ポイント (pt) / ミリ (mm) が使用できます。

3.4.6 **箇条書**

箇条書は、必要な数だけ作ることができます。箇条書を使わない場合は、ファイルから削除しても構いません。

```
箇条書
  トリガ
                  k1
  書体
                  Ryumin-Light # 和文書体
                  正体
  変形
                      # 指定項目:
                  正体/かな/
                  長体1/長体2/長体3/長体4/
                  平体1/平体2/平体3/平体4/
             #
                  斜体1/斜体2/斜体3
                       # 単位:Q/pt
  級数
                  1.30
                       # 単位:歯/pt/mm
  字送り
                  13歯
                                 # 和文強調書体
  強調書体
                  GothicBBB-Medium
  強調変形
                  正体
                       # 指定項目:
                  正体/かな/
             #
                  長体1/長体2/長体3/長体4/
                  平体1/平体2/平体3/平体4/
             #
                  斜体1/斜体2/斜体3
             #
                       # 単位:Q/pt
  強調級数
                  130
  強調字送り
                       # 単位:歯/pt/mm
                  13歯
  欧文書体
                  PCTimes-Roman
                            # 欧文書体
                      # 単位:Q/pt
  欧文級数
                  130
                       # 単位:字/歯/pt/mm
  欧文ベースライン
                  0字
             # 値を大きくするほど欧文が下がります
  文字色シアン
                      # CMYKでシアン成分を0.0~1.0で指定
                  0.0
  文字色マゼンタ
                  0.0
                       # CMYKでマゼンタ成分を0.0~1.0で指定
  文字色イエロー
                  0.0
                      # CMYKでイエロー成分を0.0~1.0で指定
  文字色ブラック
                       # CMYKでブラック成分を0.0~1.0で指定
                  1.0
  欧文強調書体
                  PCHelvetica
                             # 欧文強調書体
                      # 単位:Q/pt
  欧文強調級数
                  13Q
  欧文強調ベースライン
                  0字
                       # 単位:字/歯/pt/mm
            # 値を大きくするほど欧文が下がります
  強調文字色シアン
                0.0 # CMYKでシアン成分を0.0~1.0で指定
  強調文字色マゼンタ
                  0.0
                      # CMYKでマゼンタ成分を0.0~1.0で指定
  強調文字色イエロー
                      # CMYKでイエロー成分を0.0~1.0で指定
                  0.0
                      # CMYKでブラック成分を0.0~1.0で指定
  強調文字色ブラック
                  1.0
                      # 単位:歯/pt/mm
                  22字
  行送り
                      # 単位:字/歯/pt/mm
  字下げ
                  1字
  字上げ
                  0字
                      # 単位:字/歯/pt/mm
                      # 単位:字/歯/pt/mm
  問答
                  1字
  項目空き
                  2歯
                      # 単位:歯/pt/mm
  半改行自動挿入
                  有り
                      # 指定項目:有り/無し
                 0.5行 # 単位:行/歯/pt/mm
  改行挿入
  囲み罫
                  無し
                      # 指定項目:囲み罫トリガ名/無し
箇条書終り
```

トリガ

箇条書のトリガを定義します。この例のように"k1"と指定すると、開始のトリガが"//k1{"に、終了のトリガが"//k1}"になります。

項目空き

箇条書の項目間のアキを指定します。この値は、箇条書の行送りに足し込まれます。単位は 歯数(歯)/ポイント(pt)/ミリ(mm)が使用できます。

3.4.7 見出し

見出しは、必要な数だけ作ることができます。見出しを使わない場合は、ファイルから削除しても構いません。

見出しは文字だけで構成する方法と、EPS画像を使用する方法が選択できます。

また、見出しを目次に反映させるかどうかの選択と、目次に反映させる場合の目次の体裁の設定もここで行います。

```
見出し
  トリガ
             i
  上位見出し
             無し
                       # 指定項目:見出しトリガ名/無し
  見出し番号指定文字列
                  chapter
  書体
            ShinGo-Bold
                       # 和文書体
  変形
             正体
                       # 指定項目:
                  正体/かな/
                  長体1/長体2/長体3/長体4/
             #
                  平体1/平体2/平体3/平体4/
                  斜体1/斜体2/斜体3
            #
                       # 単位:Q/pt
  級数
            44Q
  字送り
                       # 単位:歯
            42歯
  欧文書体
                      # 欧文書体
            PCSINGB-SE1
  欧文級数
                       # 単位:Q/pt
  欧文ベースライン
                  0字
                       # 単位:字/歯/pt/mm
            # 値を大きくするほど欧文が下がります
  文字色シアン
                       # CMYKでシアン成分を0.0~1.0で指定
            0.0
  文字色マゼンタ 0.0
                       # CMYKでマゼンタ成分を0.0~1.0で指定
  文字色イエロー 0.0
                       # CMYKでイエロー成分を0.0~1.0で指定
                       # CMYKでブラック成分を0.0~1.0で指定
  文字色ブラック 1.0
                       # 単位:歯
  行送り
            54歯
  前空き
                       # 単位:行/歯/pt/mm
            0行
                       # 単位:行/歯/pt/mm
  後空き
            1行
                       #後空きはEPS画像が"無し"の場合のみ有効
  行取り
            13行
                       # 単位:行/歯/pt/mm
                       # 単位:字/歯/pt/mm
  字上げ
            0字
  字下げ
            0字
                       # 単位:字/歯/pt/mm
  左右揃え
            左
                       # 指定項目:左/中/右
            右ページ
                       # 指定項目:
  体裁
                  改ページ
            #
             #
                  左ページ
                  右ページ
             #
                  扉右ページ
             #
             #
                  扉左ページ
                  扉白
             #
                  扉白右ページ
             #
                  扉白左ページ
             #
             #
                  無し
                       #前の体裁で,"無し"以外を指定した
  頁体裁番号
            1
             #場合に有効 .
  継続頁体裁番号 2
                       # この見出し以降の基本頁体裁を切り替える
                       # arabic:数字(default)
  シンボル
            arabic
```

```
# roman :小文字ローマ数字
          # Roman :大文字ローマ数字
          # alph :小文字英字
          # Alph
               :大文字英字
問答
          4字
                    # 単位:字/歯/pt/mm
付き方
          中
               # 上:行取りの上端と文字列の上端が一致する
       #中:行取りの中心と文字列の中心が一致する
       # 下:行取りの下端と文字列の下端が一致する
見出し文字列
         "%i番号%章 "
                    #後に続く本文がこの行数より少なければ
最小継続本文行数
                    # 改ページする
ページ番号参照リンク
               有り
       # 指定項目:有り/無し
       # //pagerefによるページ番号からの
       # PDFリンクを有効にするかどうか
見出し番号参照リンク
               無し
       # 指定項目:有り/無し
       # //numrefによる見出し番号からの
       # PDFリンクを有効にするかどうか
文字列参照リンク
               無し
       # 指定項目:有り/無し
       # //strrefによる見出し文字列からの
       # PDFリンクを有効にするかどうか
目次出力
          有 ()
               # 指定項目:有り/無し
               # 見出しを目次に反映させるかどうか
               # 以下は"有り"の場合のみ有効
目次書体
          ShinGo-Medium # 和文書体
目次变形
          正体
               # 指定項目:
               # 正体/かな/
               # 長体1/長体2/長体3/長体4/
               # 平体1/平体2/平体3/平体4/
               # 斜体1/斜体2/斜体3
目次級数
               # 単位:Q/pt
          20.0Q
          20.0歯 # 単位:歯/pt/mm
目次字送り
目次欧文書体
          PCSINGM-SE1
                    # 欧文書体
              # 単位:Q/pt
目次欧文級数
          20.0Q
目次欧文ベースライン
               0.03字 # 単位:字/歯/pt/mm
               # 値を大きくするほど欧文が下がります
目次文字色シアン
               0.00
                  # CMYKでシアン成分を0.0~1.0で指定
                   # CMYKでマゼンタ成分を0.0~1.0で指定
目次文字色マゼンタ
               0.00
目次文字色イエロー
               0.00
                   # CMYKでイエロー成分を0.0~1.0で指定
目次文字色ブラック
                   # CMYKでブラック成分を0.0~1.0で指定
               1.00
目次ページ書体 PCHelvetica
                   # 欧文書体
目次ページ級数 13.0Q
               # 単位:Q/pt
目次ページベースライン 0.03字 # 単位:字/歯/pt/mm
               # 値を大きくするほど欧文が下がります
目次ページ文字色シアン 0.00 # CMYKでシアン成分を0.0~1.0で指定
目次ページ文字色マゼンタ
                      # CMYKでマゼンタ成分を0.0~1.0で指定
                0.00
                      # CMYKでイエロー成分を0.0~1.0で指定
目次ページ文字色イエロー
                 0.00
                      # CMYKでブラック成分を0.0~1.0で指定
目次ページ文字色ブラック
                1.00
          20.0歯 # 単位:歯/pt/mm
目次行送り
目次問答
         4.0字
              # 単位:字/歯/pt/mm
目次前空き
              # 単位:行/歯/pt/mm
         1.0行
              # 単位:行/歯/pt/mm
目次後空き
         0.0行
目次字下げ
         0.0字
              # 単位:字/歯/pt/mm
```

```
目次字上げ
         0.0字 # 単位:字/歯/pt/mm
目次行揃え
         左
               # 指定項目:左/中/右
目次リーダー罫 2
               # リーダー罫の線種
               # 0 --- ", "
               # 1 --- " 、 "
               # 2 --- 3点リーダー( センター )
               #3---3点リーダー(下)
               #4--- 罫線(センター)
               #5--- 罫線(下)
               #6--- 固定スペース
                   # 単位:mm/歯/pt
目次リーダー太さ
             0.1mm
               # リーダーに罫線を選んだ場合の罫線の太さ
目次リーダースペース
                     # 単位:mm/歯/pt
               # リーダーに固定スペースを選んだ場合の罫線の太さ
目次囲み罫
          無し
               # 指定項目:有り/無し
               #以下の目次枠指定は"有り"の場合のみ有効
              # 単位:行/mm/歯/pt
目次枠内上空き 0.0行
目次枠内下空き 0.0行
              # 単位:行/mm/歯/pt
目次枠内左空き 0.0字 # 単位:行/mm/歯/pt
目次枠内右空き 0.0字
              # 単位:行/mm/歯/pt
目次枠上太さ 0.0mm # 単位:mm/歯/pt
         0.0mm # 単位:mm/歯/pt
目次枠下太さ
        0.0mm # 単位:mm/歯/pt
目次枠左太さ
目次枠右太さ 0.0mm # 単位:mm/歯/pt
目次枠左上半径 0.0mm # 単位:mm/歯/pt
目次枠右上半径 0.0mm # 単位:mm/歯/pt
目次枠左下半径 0.0mm # 単位:mm/歯/pt
目次枠右下半径 0.0mm # 単位:mm/歯/pt
目次枠色シアン 0.0 # CMYKでシアン成分を0.0~1.0で指定
目次枠色マゼンタ 0.0 # CMYKでマゼンタ成分を0.0~1.0で指定
目次枠色イエロー
           0.0 # CMYKでイエロー成分を0.0~1.0で指定
目次枠色ブラック 0.0 # CMYKでブラック成分を0.0~1.0で指定
目次背景色シアン
           0.0 # CMYKでシアン成分を0.0~1.0で指定
目次背景色マゼンタ 0.0 # CMYKでマゼンタ成分を0.0~1.0で指定
目次背景色イエロー 0.0 # CMYKでイエロー成分を0.0~1.0で指定
目次背景色ブラック 0.0 # CMYKでブラック成分を0.0~1.0で指定
EPS画像 無し
               # 指定項目:有り/無し
               # EPSファイルを使用した見出しにするかどうか
               # 以下は"有り"の場合のみ有効
EPSファイル名
EPS番号書体
          ShinGo-Bold
                    # 和文書体
EPS番号变形
          正体
              # 指定項目:
               # 正体/かな/
               # 長体1/長体2/長体3/長体4/
               # 平体1/平体2/平体3/平体4/
               # 斜体1/斜体2/斜体3
EPS番号級数
          44.00
              # 単位:Q/pt
          42歯
               # 単位:歯/pt/mm
EPS番号字送り
EPS番号欧文書体 PCSINGB-SE1
                  # 欧文書体
EPS番号欧文級数 44.0Q # 単位:Q/pt
EPS番号欧文ベースライン 0.03字 # 単位:字/歯/pt/mm
               # 値を大きくするほど欧文が下がります
EPS番号文字色シアン
                    # CMYKでシアン成分を0.0~1.0で指定
               0.0
EPS番号文字色マゼンタ
                    # CMYKでマゼンタ成分を0.0~1.0で指定
               0.0
EPS番号文字色イエロー 0.0
                   # CMYKでイエロー成分を0.0~1.0で指定
EPS番号文字色ブラック
               1.0
                    # CMYKでブラック成分を0.0~1.0で指定
```

EPS番号タグ名 "@n@" EPS番号タグ横基準位置 左 # 指定項目:左/中/右

EPS番号タグ縦基準位置 上 # 指定項目:上/中/下/ベースライン

EPS番号横基準位置 左 # 指定項目:左/中/右

EPS番号縦基準位置 上 # 指定項目:上/中/下/ベースライン

 EPS番号幅
 4.0字
 # 単位:mm/歯/pt

 EPS番号揃え
 左
 # 指定項目:左/中/右

 EPS番号組方向
 横
 # 指定項目:横/縦

EPS文字列タグ名 "@str@"

EPS文字列タグ横基準位置 左 # 指定項目:左/中/右

EPS文字列タグ縦基準位置 上 # 指定項目:上/中/下/ベースライン

EPS文字列横基準位置 左 # 指定項目:左/中/右

EPS文字列縦基準位置 上 # 指定項目:上/中/下/ベースライン

EPS文字列幅10.0字# 単位:字/mm/歯/ptEPS文字列許容圧縮率80# 単位:パーセントEPS文字列揃え左# 指定項目:左/中/右EPS文字列折り返し有り# 指定項目:有り/無し

EPS文字列組方向 横 # 指定項目:横/縦

見出し終り

トリガ

見出しのトリガを定義します。

上位見出し

この見出しの上位となる見出しトリガを指定します。このトリガが最上位の場合は "無し" を指定します。

見出し番号指定文字列

原稿の先頭で見出し番号を設定するための文字列を指定します。

前空き

見出しの前空きを指定します。単位は基本組の行数 (行) / 歯数 (歯) / ポイント (pt) / ミリ (mm) が使用できます。

後空き

見出しの後空きを指定します。EPS 画像を使用した場合は、この設定は無視されます。単位は基本組の行数(行)/歯数(歯)/ポイント(pt)/ミリ(mm)が使用できます。

行取り

見出しの行取りを指定します。EPS 画像を使用した場合は、画像の上端から次の本文までの幅になります。単位は基本組の行数(行)/歯数(歯)/ポイント(pt)/ミリ(mm)が使用できます。"前空き"と"後空き"の指定とうまく組み合せましょう。

左右揃え

見出し全体の左右揃えを指定します。"左、中、右"のいずれかを指定してください。

体裁

見出しの前後で、どのような改ページを行うのかを指定します。" 改ページ、左ページ、右

ページ、扉、扉右ページ、扉左ページ、扉白、扉白右ページ、扉白左ページ、無し"のいずれかを指定してください。"扉白"とは、裏白の扉を示します。

頁体裁番号

" 頁体裁 " で指定した頁体裁番号を指定します。" 体裁 " の指定で、" 無し " 以外を指定した場合に、この項目が有効になります。この番号によって、見出しのあるページの柱やノンブルのスタイルを変えることができます。

継続頁体裁番号

" 頁体裁 " で指定した頁体裁番号を指定します。この番号によって、見出しのあるページ以降のページの柱やノンブルのスタイルを変えることができます。 頁体裁番号が設定されている場合、見出しのあるページではそちらが優先されます。

シンボル

見出しの番号付けに使う文字種を指定します。"arabic (数字)、roman (小文字ローマ数字)、Roman (大文字ローマ数字)、alph (小文字英字)、Alph (大文字英字)"のいずれかを指定してください。

付き方

行取り指定した見出しに対して、行送り方向の揃えを指定します。"上、中、下"のいずれかを指定してください。

見出し番号文字列

見出し番号の出力方法を指定します。"%i 番号%.%ii 番号% "のように " %< トリガ > 番号% " という変数と通常の文字を組み合わせて入力します。必ず、半角のダブルクォーテーションで囲んでください。

最小継続本文行数

見出しの後の本文が見出しと同じページに指定した行数分収まらなければ、見出しごと改ページします。体裁が "無し"の場合のみ有効です。

ページ番号参照リンク

//pageref による見出しページ番号参照を PDF でのリンクとして有効にするかどうかを指定します。"有り、無し"のいずれかを指定してください。"有り"を指定すると参照されたページ番号がリンクとして有効になります。

見出し番号参照リンク

//numref による見出し番号参照を PDF でのリンクとして有効にするかどうかを指定します。 "有り、無し"のいずれかを指定してください。"有り"を指定すると参照された見出し番号がリンクとして有効になります。リンクとして有効になった文字列は途中で改行できないので注意が必要です。

文字列参照リンク

//strref による見出し文字列参照を PDF でのリンクとして有効にするかどうかを指定します。 "有り、無し"のいずれかを指定してください。"有り"を指定すると参照された文字列がリ

ンクとして有効になります。リンクとして有効になった文字列は途中で改行できないので注 意が必要です。

目次出力

見出しを目次に出力するかどうかを指定します。

目次書体

目次の見出し文字列部の書体です。

目次ページ書体

目次のページ番号部の書体です。

目次リーダー罫

目次の見出し文字列部とページ番号部の間に入れるリーダーの形態を番号で指定します。番号は以下のうちから選択します。

- 0 "."
- 1 ", "
- 2 3点リーダー(センター)
- 3 3点リーダー(下)
- 4 罫線(センター)
- 5 罫線(下)
- 6 固定スペース

目次リーダー太さ

目次リーダー罫に罫線(4または5)を指定した場合の罫線の太さを指定します。単位はミリ(mm)/歯数(歯)/ポイント(pt)が使用できます。

目次リーダースペース

目次リーダー罫に固定スペース(6)を指定した場合の罫線の太さを指定します。単位はミリ(mm)/歯数(歯)/ポイント(pt)が使用できます。

目次囲み罫

目次項目を罫線で囲むかどうかを設定します。"有り"で囲み罫を使用します。

目次枠内空き

囲み罫と目次文字列の間の空きを設定します。

目次枠太さ

囲み罫の罫線の太さを設定します。

目次枠半径

囲み罫の角を丸める場合の半径を設定します。丸めない場合は0を設定します。

目次枠色

囲み罫の色を CMYK 形式で設定します。

目次枠色

囲み罫の中の背景色を CMYK 形式で設定します。

EPS 画像

EPS 画像を指定した見出しにするかどうかを指定します。"有り"で EPS 画像を使用します。

EPS ファイル名

見出しに EPS 画像を使用する場合の EPS ファイル名を指定します。 EPS 画像中に見出し番号 / 文字列のタグ文字列が入っていれば、それぞれ見出し番号と見出し文字列に置き換えて出力します。また、ファイル名中に"% トリガ 番号%"が記述してあれば、その部分を見出し番号毎に置き換えて EPS ファイル名として使用します。たとえば"chapter%i番号%.eps"のように記述してあれば、1章では"chapter1.eps"、2章では"chapter2.eps"が見出しとして使用されます。

EPS 番号書体

EPS 画像を使用した場合、見出し番号と見出し文字列の書体を分けて指定します。見出し番号の書体は"EPS 番号書体"、見出し文字列の書体は"見出し書体"で指定します。

EPS 番号タグ名

見出し番号は EPS 画像中の文字列と置き換えられます。置き換える EPS 画像中の文字列をここで指定します。

EPS 番号タグ縦/横基準位置

EPS 画像中の見出し番号タグのどこを置き換えの基準点にするかを指定します。縦は"上、中、下、ベースライン"から、横は"左、中、右"から選択します。

EPS 番号縦/横基準位置

置き換える見出し番号文字列のどこを置き換えの基準点にするかを指定します。選択肢は EPS 番号タグ基準位置と同様です。

EPS 番号幅

置き換える見出し番号文字列の幅を指定します。

EPS 番号揃え

EPS 番号幅に対して、実際に出力される見出し番号の左右揃えを"左、中、右"から選択して指定します。

EPS 番号組方向

見出し番号の組方向を"縦"または"横"で指定します。

EPS 文字列タグ名

見出し文字列は EPS 画像中の文字列と置き換えられます。置き換える EPS 画像中の文字列をここで指定します。

EPS 文字列タグ縦/横基準位置

EPS 画像中の見出し文字列タグのどこを置き換えの基準点にするかを指定します。縦は"上、中、下、ベースライン"から、横は"左、中、右"から選択します。

EPS 文字列縦/横基準位置

置き換える見出し文字列のどこを置き換えの基準点にするかを指定します。選択肢は EPS 文字列タグ基準位置と同様です。

EPS 文字列幅

置き換える見出し文字列の幅を指定します。

EPS 文字列許容圧縮率

見出し文字列幅が EPS 文字列幅を超える場合、どのくらいまで長体を掛けてよいかをパーセントで指定します。

EPS 文字列揃え

EPS 文字列幅に対して、実際に出力される見出し文字列の左右揃えを"左、中、右"から選択して指定します。

EPS 文字列折り返し

見出し文字列幅が EPS 文字列幅を超え、さらに許容圧縮率でも収まらない場合に折り返すかどうかを指定します。"有り"で折り返します。"無し"で折り返しません。

EPS 文字列組方向

見出し文字列の組方向を"縦"または"横"で指定します。

3.4.8 **図表のキャプション**

図表のキャプションには、"図、表、リスト、イラスト、写真、画面、式"の7種類が用意されています。これらのキャプションを使わない場合は、ファイルから削除しても構いません。

```
义
  トリガ
             f
  書体
                              # 和文書体
             GothicBBB-Medium
                        # 指定項目:
             正体
  変形
             #
                   正体/かな/
             #
                   長体1/長体2/長体3/長体4/
                   平体1/平体2/平体3/平体4/
                   斜体1/斜体2/斜体3
             #
  級数
                        # 単位:Q
             12Q
  字送り
             12歯
                        # 単位:歯/pt/mm
  欧文書体
                        # 欧文書体
             PCHelvetica
                        # 単位:Q
  欧文級数
             12Q
  欧文ベースライン
                   0字
                        # 単位:字/歯/pt/mm
             # 値を大きくするほど欧文が下がります
  文字色シアン
                        # CMYKでシアン成分を0.0~1.0で指定
             0.0
  文字色マゼンタ
             0.0
                        # CMYKでマゼンタ成分を0.0~1.0で指定
  文字色イエロー 0.0
                        # CMYKでイエロー成分を0.0~1.0で指定
  文字色ブラック 1.0
                        # CMYKでブラック成分を0.0~1.0で指定
  行送り
             16歯
                        # 単位:歯/pt/mm
                        # 単位:字/歯/pt/mm
  字下げ
             0字
```

```
字上げ
            0字
                      # 単位:字/歯/pt/mm
  問答
            1字
                      # 単位:字/歯/pt/mm
  追加問答
                      # 単位:字/歯/pt/mm
                 2字
  強調の有無
            先頭
                      # 指定項目:
            #
                 先頭
                 全部
            #
                 無し
  文字列
            "図%章%-%章内番号%"
  枝付き文字列
            "図%章%-%章内番号%(%枝番号%)"
  前後空き
                 0.5行 # 単位:行/歯/pt/mm
  図キャプション空き
                    # 単位:歯/pt/mm
                 20歯
  ページブレーク 無し
                      # 無し:ページブレークしない(フロートする)
            # 有り:ページブレークする(フロートしない)
  図表参照リンク
           無し
       # 指定項目:有り/無し
       # //f番号による図表参照からの
       # PDFリンクを有効にするかどうか
  ページ番号参照リンク
                 有り
       # 指定項目:有り/無し
       # //pagerefによるページ番号からの
       # PDFリンクを有効にするかどうか
  図表番号参照リンク 無し
       # 指定項目:有り/無し
       # //numrefによる図表番号からの
       # PDFリンクを有効にするかどうか
  キャプション参照リンク 無し
       # 指定項目:有り/無し
       # //strrefによるキャプション文字列からの
       # PDFリンクを有効にするかどうか
図終り
```

トリガ

図表トリガを定義します。

強調の有無

参照部の強調の仕方を指定します。"先頭、全部、無し"のいずれかを指定してください。1つの図表に対して、参照部が複数ある時に、一番先頭に出てきた番号のみを強調するのが"先頭"という指定です。これに対して、参照部をすべて強調するのが"全部"、強調しないのが"無し"という指定です。

文字列

キャプションの先頭に付く図表番号の出力方法を指定します。"%節内番号%、%章内番号%、%節%、%章%"の変数が使えます。"%章%"と"%節%"は、それぞれ章番号、節番号を表し、"%章内番号%(章ごとに番号付け)"と"%節内番号%(節ごとに番号付け)"は図表の番号を表します。これらの変数と固定文字列を記述する時には、必ず、半角のダブルクォーテーションで囲ってください。もし、出力したくない場合は、""と入力してください。

枝付き文字列

技番号付きキャプションの先頭に付く図表番号の出力方法を指定します。"文字列"で説明し

た変数の他に、"%枝番号%"という変数も使えます。この変数は、原稿中に"//f100-a"などと入力した場合の"a"を表します。

前後空き

図表類と本文とのアキを指定します。単位は基本組の行数 (行) / 歯数 (歯) / ポイント (pt) / ミリ (mm) が使用できます。

図キャプション空き

図表の実体とキャプションとのアキを指定します。単位は歯数 (歯) / ポイント (pt) / ミリ (mm) が使用できます。

ページブレーク

図表トリガで表される図版が、ページで分かれるかどうかを指定します。" 有り、無し " のいずれかを指定してください。" 有り " を指定するとフローティングしない図表となり、" 無し " にするとフローティングする図表となります。

図表参照リンク

//f 番号による図表番号参照を PDF でのリンクとして有効にするかどうかを指定します。" 有 り、無し"のいずれかを指定してください。" 有り"を指定すると参照された図表番号がリンクとして有効になります。リンクとして有効になった文字列は途中で改行できないので注意が必要です。

ページ番号参照リンク

//pageref による図表ページ番号参照を PDF でのリンクとして有効にするかどうかを指定します。" 有り、無し " のいずれかを指定してください。" 有り " を指定すると参照されたページ番号がリンクとして有効になります。

図表番号参照リンク

//numref による図表番号参照を PDF でのリンクとして有効にするかどうかを指定します。" 有 り、無し"のいずれかを指定してください。" 有り"を指定すると参照された図表番号がリンクとして有効になります。リンクとして有効になった文字列は途中で改行できないので注意が必要です。

キャプション参照リンク

//strref による図表キャプション参照を PDF でのリンクとして有効にするかどうかを指定します。"有り、無し"のいずれかを指定してください。"有り"を指定すると参照されたキャプション文字列がリンクとして有効になります。リンクとして有効になった文字列は途中で改行できないので注意が必要です。

3.4.9 プログラム

プログラムリストの体裁を入力します。複数個作れます。プログラムリストを使わない場合は、 削除しても構いません。

```
プログラム
  トリガ
            list
  級数
                      # 単位:Q/pt
            120
  書体
            GothicBBB-Medium
                            # 和文書体
  和文变形
               正体
                      # 指定項目:
                 正体/かな/
            #
                  長体1/長体2/長体3/長体4/
            #
                  平体1/平体2/平体3/平体4/
            #
            #
                  斜体1/斜体2/斜体3
  和文級数
                       # 単位:Q/pt
            11Q
                      # 単位:歯/pt/mm
  字送り
            11歯
  文字色シアン
            0.0
                      # CMYKでシアン成分を0.0~1.0で指定
  文字色マゼンタ 0.0
                      # CMYKでマゼンタ成分を0.0~1.0で指定
  文字色イエロー 0.0
                      # CMYKでイエロー成分を0.0~1.0で指定
  文字色ブラック 1.0
                      # CMYKでブラック成分を0.0~1.0で指定
                      # 単位:歯/pt/mm
  行送り
            14歯
                      # 単位:字/歯/pt/mm
  字下げ
            1字
  字上げ
            0字
                      # 単位:字/歯/pt/mm
  インデント
            2
                      #単位:字 プログラムのインデント
  行番号有無
            無し
                      # 指定項目:有り/無し
                      # 欧文書体
  行番号書体
           Helvetica
                      # 単位:Q/pt
  行番号級数
            13Q
            無し
                      # 指定項目:有り/無し
  円マーク
            # バックスラッシュを円マークにす
            # るかどうか
  半改行自動挿入 有り
                       # 指定項目:有り/無し
  改行挿入
            0.5行
                       # 単位:行/歯/pt/mm
  コメント書体
                 GothicBBB-Medium
                                # コメント和文書体
  コメント変形
                      # 指定項目:
                 正体
                 正体/かな/
            #
            #
                  長体1/長体2/長体3/長体4/
                 平体1/平体2/平体3/平体4/
            #
                  斜体1/斜体2/斜体3
  コメント級数
                      # 単位: Q/pt
                 110
                            # 単位:歯/pt/mm
  コメント字送り
                 11歯
  コメント欧文書体
                 PCHelvetica
                            # コメント欧文書体
  コメント欧文級数
                 11Q # 単位:Q/pt
  コメントベースライン
                 0字
                       # 単位:字/歯/pt/mm
            # 値を大きくするほど欧文が下がります
  コメント文字色シアン
                 0.0
                      # CMYKでシアン成分を0.0~1.0で指定
  コメント文字色マゼンタ 0.0
                      # CMYKでマゼンタ成分を0.0~1.0で指定
  コメント文字色イエロー 0.0
                      # CMYKでイエロー成分を0.0~1.0で指定
                      # CMYKでブラック成分を0.0~1.0で指定
  コメント文字色ブラック 1.0
                 11歯 # 単位:歯/pt/mm
  コメント字送り
                      # 指定項目・囲み罫トリガ名/無し
  用み罫
            無し
プログラム終り
```

プログラムのトリガを定義します。この例のように "list" と指定すると、開始のトリガが "//list{"に、終了のトリガが"//list}"になります。

インデント

プログラムリストのインデント量を指定します。タブストップの値を入力してください。 行番号有無

プログラムリストの先頭に行番号を付加するか、しないかを指定できます。"有り、無し"のいずれかを指定してください。

円マーク

プログラムリスト中の"∖"(バックスラッシュ)を"¥"に変えることができます。"有り" を指定すると変換し、"無し"を指定すると変換しません。

コメント

プログラムリスト内のコメント (" //cmt{ ") に対して、書体や級数などを指定できます。

3.4.10 **本文タブ**

本文中(表組とプログラム以外)で使用されるタブに対して、タブストップの値を設定します。

本文タブ スペース 2 # 半角字 タブをいくつのスペースにかするか 本文タブ終り

スペース

1つのタブを、いくつのスペースに展開するのかを指定します。単位は半角の文字です。

3.4.11**キートップ**

キートップは、必要な個数だけ作ることができます。キートップを使わない場合は、ファイル から削除しても構いません。

キートップ
トリガ key1
書体 Ryumin-Light # 和文書体
変形 正体 # 指定項目:
正体/かな/
長体1/長体2/長体3/長体4/
平体1/平体2/平体3/平体4/

```
斜体1/斜体2/斜体3
  級数
                        # 単位:Q/pt
             110
  欧文書体
                        # 欧文書体
             PCTimes-Roman
  欧文級数
                        # 単位: Q/pt
  欧文ベースライン
                  0字
                        # 単位:字/歯/pt/mm
             # 値を大きくするほど欧文が下がります
                        # CMYKでシアン成分を0.0~1.0で指定
  文字色シアン
             0.0
                        # CMYKでマゼンタ成分を0.0~1.0で指定
  文字色マゼンタ 0.0
  文字色イエロー 0.0
                        # CMYKでイエロー成分を0.0~1.0で指定
  文字色ブラック 1.0
                       # CMYKでブラック成分を0.0~1.0で指定
                       # 単位:歯/pt/mm
  字送り
             11歯
  行送り
             13歯
                       # 単位:歯/pt/mm
  高さ
             14歯
                        # 単位:歯/pt/mm
  幅
                        # 単位:歯/pt/mm
             14歯
  角丸
             無し
                        # 有り:角丸 無し:直角
             # 縦組では、角丸キートップは使えません
キートップ終り
```

トリガ

キートップのトリガを定義します。このように" key1 "と指定すると、開始のトリガが" //key1{ "に、終了のトリガが" //key1}"になります。

言さ

キートップの高さを指定します。単位は歯数 (歯) / ポイント (pt) / ミリ (mm) が使用できます。

幅

キートップの幅を指定します。単位は歯数 (歯) / ポイント (pt) / ミリ (mm) が使用できます。 角丸

キートップのカドを角丸にするか、直角にするのかを指定します。"無し"にすると直角に、"有り"にすると角丸になります。なお、縦組では角丸のキートップは使えません。

行送り

文字列が2段になるキートップを作る場合に使用する行送り(上段と下段とのアキ)を指定します。単位は歯数(歯)/ポイント(pt)/ミリ(mm)が使用できます。

3.4.12 罫線

本文中で使用する横に伸びる罫線を作成します。罫線は、必要な数だけ作成できます。罫線を 使わない場合は、ファイルから削除しても構いません。 罫線 トリガ kei # 単位:字/歯/pt/mm 長さ 43字 太さ # 単位:mm/歯/pt 0.5mm 罫線色シアン # CMYKでシアン成分を0.0~1.0で指定 0.0 罫線色マゼンタ 0.0 # CMYKでマゼンタ成分を0.0~1.0で指定 罫線色イエロー 0.0 # CMYKでイエロー成分を0.0~1.0で指定 # CMYKでブラック成分を0.0~1.0で指定 罫線色ブラック 1.0 前空き # 単位:mm/歯/pt 9mm 後空き # 単位:mm/歯/pt 2.5mm # 単位:字/歯/pt/mm 字下げ 0字 罫線終り

トリガ

罫線のトリガを定義します。このトリガの名前には数字を指定することができません。トリガの名前には、アルファベットを使用してください。

長さ

罫線の長さを指定します。単位は基本組の字送り(字)/歯数(歯)/ポイント(pt)/ミリ(mm)が使用できます。

太さ

罫線の太さを指定します。単位はミリ (mm) / 歯数 (歯) / ポイント (pt) が使用できます。

罫線色

罫線の色を CMYK 形式の数値で指定します。数値は 0.0~1.0 の範囲です。

前空き / 後空き

罫線の上下のアキを指定します。単位はミリ (mm) / 歯数 (歯) / ポイント (pt) が使用できます。

字下げ

罫線の字下げを指定します。単位は基本組の字送り (字) / 歯数 (歯) / ポイント (pt) / ミリ (mm) が使用できます。

3.4.13 囲み罫

文章を罫線で囲む場合の指定です。ここで作られた囲み罫は小組/箇条書/表組/プログラム リストに使用できます。

囲み罫 トリガ w1

```
上罪太さ
           1.0mm # 単位:mm/歯/pt
           1.0mm # 単位:mm/歯/pt
  ち太澤不
  左罫太さ
            1.0mm # 単位:mm/歯/pt
  右罫太さ
            1.0mm
                 # 単位:mm/歯/pt
  左上角半径
            3.0mm
                 # 単位:mm/歯/pt
  右上角半径
            3.0mm
                 # 単位:mm/歯/pt
            3.0mm # 単位:mm/歯/pt
  右下角半径
  左下角半径
           3.0mm # 単位:mm/歯/pt
  外前空き
           1.0行 # 単位:行/歯/pt/mm
            1.0行 # 単位:行/歯/pt/mm
  外後空き
           0.0字 # 単位:字/歯/pt/mm
  外字上げ
  外字下げ
           0.0字 # 単位:字/歯/pt/mm
  内前空き
           1.0行 # 単位:行/歯/pt/mm
            1.0行 # 単位:行/歯/pt/mm
  内後空き
  内字上げ
            0.0字 # 単位:字/歯/pt/mm
  内字下げ
            0.0字 # 単位:字/歯/pt/mm
  罫線色シアン 0.00 # CMYKでシアン成分を0.0~1.0で指定
  罫線色マゼンタ 0.00 # CMYKでマゼンタ成分を0.0~1.0で指定
  罫線色イエロー 0.00 # CMYKでイエロー成分を0.0~1.0で指定
  罫線色ブラック 1.00
                 # CMYKでブラック成分を0.0~1.0で指定
                 # CMYKでシアン成分を0.0~1.0で指定
  背景色シアン
            0.00
  背景色マゼンタ 0.00
                 # CMYKでマゼンタ成分を0.0~1.0で指定
  背景色イエロー 0.00 # CMYKでイエロー成分を0.0~1.0で指定
  背景色ブラック 0.10 # CMYKでブラック成分を0.0~1.0で指定
  ページ分割
            有り
                 # 指定項目:有り/無し
                 # 以下は"有り"の場合のみ有効
  #ページ分割時指定
  下端罪太さ
          0.0mm # 単位:mm/歯/pt
  下端右下角半径 0.0mm # 単位:mm/歯/pt
  下端左下角半径 0.0mm # 単位:mm/歯/pt
  下端内後空き 0.5行 # 単位:行/歯/pt/mm
  上端罫太さ
           0.0mm # 単位:mm/歯/pt
  上端左上角半径 0.0mm # 単位:mm/歯/pt
  上端右上角半径 0.0mm # 単位:mm/歯/pt
  上端内前空き 0.5行 # 単位:行/歯/pt/mm
囲み罫終り
```

トリガ

囲み罫のトリガを定義します。このように"w1"と指定すると、開始のトリガが"//w1{"に、終了のトリガが"//w1}"になります。

上下左右罫太さ

囲み罫の太さを指定します。単位はポイント (pt) / ミリ (mm) / 歯数 (歯) が使用できます。 角半径

各角を丸める場合の半径を指定します。直角に刷る場合は 0pt を指定します。単位はポイント (pt) / ミリ (mm) / 歯数 (歯) が使用できます。

外前後空き

囲み罫の外側の前後の空きを指定します。単位は行 / ポイント (pt) / ミリ (mm) / 歯数 (歯) が使用できます。

外字上げ / 字下げ

囲み罫の外側の字上げ/字下げを指定します。単位は字/ポイント (pt) /ミリ (mm) /歯数 (歯) が使用できます。

内前後空き

囲み罫の内側の前後の空きを指定します。単位は行 / ポイント (pt) / ミリ (mm) / 歯数 (歯) が使用できます。

内字上げ/字下げ

囲み罫の内側の字上げ/字下げを指定します。単位は字/ポイント (pt) /ミリ (mm) /歯数 (歯) が使用できます。

罫線色

罫線の色を CMYK 形式の数値で指定します。数値は 0.0~1.0 の範囲です。

背景色

罫線で囲まれた内側の地の色を CMYK 形式の数値で指定します。数値は $0.0 \sim 1.0$ の範囲です。 ページ分割

ページ内に収まらない場合に途中で分割するかどうかを指定します。"有り"で分割します。 "無し"で分割しません。

下端罫太さ/角半径/後空き

囲み罫で囲まれた文章がページで分割された場合の、下端の体裁を指定します。

上端罫太さ/角半径/前空き

囲み罫で囲まれた文章がページで分割された場合の、上端の体裁を指定します。

"ページ分割"を"有り"にした場合、フロートの表やリストで使用することはできません。ページ分割を"無し"に設定した囲み罫を作成して使用してください。

3.4.14ルビ

漢字に読み仮名(ルビ)を振る場合の指定です。

```
ルビ
  トリガ rb1
  書体
        Ryumin-Light
                  # 和文書体
        正体
             # 指定項目:
  変形
        # 正体/かな/
        # 長体1/長体2/長体3/長体4/
        # 平体1/平体2/平体3/平体4/
        # 斜体1/斜体2/斜体3
  欧文書体
            PCTimes-Roman # 欧文書体
  欧文ベースライン
                  0.03字 # 単位:字/歯/pt/mm
        # 値を大きくするほど欧文が下がります
```

本文ルビ空き 0.07字 # 単位:字/歯/pt/mm ルビ級数比率 0.5 # 本文に対するルビ文字の大きさを0.0~1.0で指定 揃え 中 # 指定項目:左/中/右 食い込み 0.25 # ルビがはみ出した場合の食い込み量 ルビ終り

トリガ

ルビのトリガを定義します。このように "rb1" と指定すると、開始のトリガが "//rb1{"に、終了のトリガが "//rb1}"になります。

本文ルビ空き

本文とルビ文字の間の空きを指定します。単位は字 / ポイント (pt) / ミリ (mm) / 歯数 (歯) が使用できます。

ルビ級数比率

本文文字の級数に対するルビ文字の大きさを比率で指定します。0~1の数値で指定します。 揃え

ルビを振る本文文字列の幅に対してルビをそろえる位置を指定します。"左"で本文文字列の 先頭に、"中"で中心に、"右"で末尾にルビ文字列が揃います。

食い込み

ルビを振る本文文字列よりルビ文字列が長い場合に、前後の本文にどこまで被さるかを指定 します。単位は本文文字幅です。

3.4.15 傍点

文章中の文字列に振る傍点の指定です。

```
傍点
  トリガ bou1
             11 🔨 11
  傍点文字
  書体
                 # 和文書体
       Ryumin-Light
  変形
       正体
            # 指定項目:
       # 正体/かな/
       # 長体1/長体2/長体3/長体4/
       # 平体1/平体2/平体3/平体4/
       # 斜体1/斜体2/斜体3
  欧文書体
           PCTimes-Roman # 欧文書体
                  0.03字 # 単位:字/歯/pt/mm
  欧文ベースライン
       # 値を大きくするほど欧文が下がります
  傍点級数比率 0.7
                 # 本文に対する傍点文字の大きさを0.0~1.0で指定
  本文傍点空き 0.0字 # 単位:字/歯/pt/mm
```

傍点位置補正 0.0字 # 傍点位置の字送り方向補正 # 単位:字/歯/pt/mm

傍点終り

トリガ

傍点のトリガを定義します。このように "bou1" と指定すると、開始のトリガが "//bou1{"に、終了のトリガが "//bou1}"になります。

傍点文字

傍点として本文に振る文字を指定します。

傍点級数比率

本文文字の級数に対する傍点文字の大きさを比率で指定します。0~1の数値で指定します。

本文傍点空き

本文と傍点文字の間の空きを指定します。単位は字 / ポイント (pt) / ミリ (mm) / 歯数 (歯) が使用できます。

傍点位置補正

本文文字に対する、傍点の字送り方向の位置を指定します。単位は字/ポイント (pt) / ミリ (mm) / 歯数 (歯) が使用できます。

3.4.16 **ハイパーリンク**

文書ファイルを PDF に変換したときに文中の URL をハイパーリンクとして有効にする場合の 指定です。通常は URL は他の文中文字と同様に黒で表示 / 印刷されますが、PDF ではリンク文字 を区別できるように文字色を変えたり枠線を付けられます。

```
ハイパーリンク # PDFでのみ有効
  トリガ
            url
  文字色赤
                       # RGBで赤成分を0.0~1.0で指定
             0.0
  文字色緑
                       # RGBで緑成分を0.0~1.0で指定
             0.0
  文字色青
            1.0
                       # RGBで青成分を0.0~1.0で指定
  枠線色赤
                       # RGBで赤成分を0.0~1.0で指定
             0.0
                       # RGBで緑成分を0.0~1.0で指定
  枠線色緑
             0.0
  枠線色青
             1.0
                       # RGBで青成分を0.0~1.0で指定
  枠線太さ
             1.0pt
                       # 単位:pt/mm/H
  枠線スタイル
             破線
                       # 指定項目:破線/実線
  枠破線長さ
            3.0pt
                       #単位:pt/mm/H 破線指定時のみ有効
  枠破線間隔
                       # 単位:pt/mm/H 破線指定時のみ有効
             3.0pt
ハイパーリンク終り
```

トリガ

ハイパーリンクのトリガを定義します。このように "url "と指定すると、開始のトリガが "//url { "に、終了のトリガが "//url } "になります。

文字色

PDF に変換したときの URL 文字色の赤 / 緑 / 青成分を 0~1 の範囲で指定します。全て 0 なら黒、全て 1 なら白になります。

枠線色

PDF に変換したときに URL 文字列を囲む枠線の色をの赤 / 緑 / 青成分を 0~1 の範囲で指定します。

枠線太さ

枠線の太さを指定します。枠線を表示しない場合は Opt を指定します。単位はポイント (pt) / ミリ (mm) / 歯数 (歯) が使用できます。

枠線スタイル

枠線のスタイルを指定します。実線または破線が指定できます。

枠破線長さ

枠線スタイルに破線を指定したときの実線部分の長さを指定します。単位はポイント (pt) / ミリ (mm) / 歯数 (歯) が使用できます。

枠破線間隔

枠線スタイルに破線を指定したときの実線の間隔を指定します。単位はポイント (pt) / ミリ (mm) / 歯数 (歯) が使用できます。

3.4.17 文字変換

原稿中に入力された文字をどのような文字に変換し、出力するのかを指定します。EWBでは、通常、入力された文字はそのまま出力されますが、ここの"文字変換"の項目に、特定の文字を指定すると、他の文字に変換して出力することができます。たとえば、NEC 外字などの EWB では扱えない文字を、写植の該当する文字に割り当てることで出力できるようになります。なお、半角カナ文字は扱えません。

文字変換 | \noexpand{|} | \noexpand{{|} | \noexpand{{|} | \maru{1} | \maru{2} | \maru{3} | \maru{4}

```
(5)
            \maru{5}
   6
            \maru{6}
   7
            \max{7}
    (8)
            \maru{8}
    (9)
            \maru{9}
   (10)
            \maru{10}
            "\leavevmode\kern .25zw\hbox{I}\kern .25zw"
    Ι
    II
            "\leavevmode\kern .25zw\hbox{I\kern -.5pt I}\kern .25zw"
    III
           "\leavevmode\kern .25zw\hbox{I\kern -.5pt I\kern -.5pt I}\kern .25zw"
           \verb|"leavevmode| kern .25zw| hbox{I|kern -.5pt V} kern .25zw| |
    TV
           \verb|"leavevmode\kern .25zw\hbox{V}\kern .25zw"|
    V
    VI
           "\leavevmode\kern .25zw\hbox{V\kern -.5pt I}\kern .25zw"
           "\leavevmode\kern .25zw\hbox{V\kern -.5pt I\kern -.5pt I}\kern .25zw"
    VIII
           "\leavevmode\kern .25zw\hbox{V\kern -.5pt I\kern -.5pt I\kern -.5pt I}
\kern .25zw"
    ΙX
            "\leavevmode\kern .25zw\hbox{I\kern -.5pt X}\kern .25zw"
            "\leavevmode\kern .25zw\hbox\{X\}\kern .25zw"
    X
文字変換終り
```

ここでは、1 行に対して 2 つの項目を記述します。最初の項目は"入力文字"を、2 番目の項目には"プリンタ出力文字"を指定します。上記の例では、NEC 外字の丸付き数字を、プリンタで出力できるようにしています。

入力文字列

原稿中の変換したい文字(変換対象文字)を指定します。半角カナ文字は指定できません。 出力文字列

プリンタ出力時に、変換され出力したい文字を指定します。

3.5 索引用体裁入力ファイルの詳細

ここでは、索引用体裁入力ファイルの内容を、指定項目別に細かく見てみましょう。

3.5.1 その他

索引用体裁入力ファイルも本文用と同様に、"その他"という項目ではじまります。さまざまな 初期設定値を入力します。

```
その他
書名 "psidxsted sample file"
段数 2
ページスタイル 2 # デフォルトの頁体裁番号
```

```
スタイルファイル ./index.cls #
インデックススタイル ./indexsty # makeindex用のスタイル
その他終り
```

書名

作成する本の書名などを入力してください。もし、書名にスペースがはいるようであれば、 半角ダブルコーテーションで文字列を囲ってください。

段数

索引の段数を指定します。

ページスタイル

通常のページに使うページスタイルの"頁体裁番号"を指定します。

スタイルファイル

psidxsted によって生成されるスタイルファイルの名前を設定します。

インデックススタイル

psidxsted によって生成されるインデックススタイルの名前を設定します。

ベーススタイル

基本となる EWB のベーススタイルファイルを設定します。通常は、"ewbbase3_3.sty"にしておきます。

3.5.2 基本組

基本的な組の指定をします。ここで入力する値によって版面のサイズが決ります。

```
基本組
  書体
                         # 和文書体
              Ryumin-Light
                   # 指定項目:
  变形
              正体
                 正体/かな/
           #
                 長体1/長体2/長体3/長体4/
           #
                 平体1/平体2/平体3/平体4/
                 斜体1/斜体2/斜体3
                    # 単位:Q/pt
  級数
              12Q
  字送り
              12歯
                    # 単位:歯/pt/mm
  欧文書体
              PCTimes-Roman
                         # 欧文書体
                   # 単位:Q/pt
  欧文級数
              12Q
              0.03字 # 単位:字/歯/pt/mm
  ベースライン
              # 大きな値にするほど欧文が下がります。
              46字
                   # 単位:字
  字詰め
                   # 単位:歯/pt/mm
  行送り
              20歯
  字下げ
                   # 単位:字/歯/pt/mm
              0字
```

```
字上げ 0字 # 単位:字/歯/pt/mm
行数 41行 # 1ページの行数
基本組終り
```

版面の大きさ

ここで入力した項目の値によって、版面のサイズが決ります。EWB では、版面の大きさの計算は、次のような式で行われます。

版面の横のサイズ = (字下げ+字詰め+字上げ-1)×字送り+和文級数 版面の縦のサイズ = (行数-1)×行送り+和文級数

3.5.3 索引見出し

索引の大見出しの体裁を指定します。

```
索引見出し
  書体
             ShinGo-Ultra
                        # 和文書体
  变形
             正体
                        # 指定項目:
             #
                   正体/かな/
             #
                  長体1/長体2/長体3/長体4/
                  平体1/平体2/平体3/平体4/
             #
                   斜体1/斜体2/斜体3
             #
  級数
             50Q
                        # 単位:Q/pt
                        # 単位:歯/pt/mm
  字送り
             48歯
  欧文書体
             PCSINGU-SE1
                        # 欧文書体
                        # 単位:Q/pt
  欧文級数
             500
                        # 単位:字/歯/pt/mm
  ベースライン
            0.03字
             #大きな値にするほど欧文が下がります。
  文字色シアン
            0.0
                       # CMYKでシアン成分を0.0~1.0で指定
  文字色マゼンタ 0.0
                       # CMYKでマゼンタ成分を0.0~1.0で指定
  文字色イエロー 0.0
                       # CMYKでイエロー成分を0.0~1.0で指定
  文字色ブラック 1.0
                       # CMYKでブラック成分を0.0~1.0で指定
  行送り
             60歯
                       # 単位:歯/pt/mm
  字下げ
             4字
                       # 単位:字/歯/pt/mm
  字上げ
             0字
                       # 単位:字/歯/pt/mm
  前空き
                       # 単位:行/歯/pt/mm
             0行
  後空き
             1行
                       # 単位:行/歯/pt/mm
  行取り
             13行
                       # 単位:行/歯/pt/mm
                        # 指定項目:左/中/右
  行揃え
             中
             右ページ
  体裁
                        # 指定項目:
             #
                   扉
                   改ページ
             #
                  左ページ
             #
                  右ページ
             #
             #
                   無し
                        #前の体裁で、"無し"以外を指定した
  頁体裁番号
             1
```

場合に有効 .

付き方 中 # 上:行取りの上端と文字列の上端が一致する

中:行取りの中心と文字列の中心が一致する 下:行取りの下端と文字列の下端が一致する

見出し文字列 "索 引"

索引見出し終り

前空き

見出しの前空きを指定します。単位は基本組の行送り(行)/歯数(歯)/ポイント(pt)/ミリ(mm)が使用できます。

後空き

見出しの後空きを指定します。単位は基本組の行送り(行)/歯数(歯)/ポイント(pt)/ミリ(mm)が使用できます。

行取り

見出しの行取りを指定します。単位は基本組の行送り(行)/歯数(歯)/ポイント(pt)/ミリ(mm)が使用できます。前空きと後空きの指定とうまく組み合せましょう。

行揃え

行方向の揃えを指定します。"左、中、右"のいずれかを指定してください。

体裁

見出しの前後で、どのような改ページを行うのかを指定します。"扉、扉白、改ページ、左ページ、右ページ、無し"のいずれかを指定してください。"扉白"とは、裏白の扉を示します。

頁体裁番号

" 頁体裁 " で指定した頁体裁番号を指定します。" 体裁 " の指定で、" 無し " 以外を指定した場合に、この項目が有効になります。この番号によって、見出しのあるページの柱やノンブルのスタイルを変えることができます。

付き方

行取り指定した見出しに対して、行送り方向の揃えを指定します。"上、中、下"のいずれかを指定してください。

見出し文字列

見出しの文字列を指定します。文字列を半角のダブルクォーテーションで囲んでください。 本文用の見出しとは異なり、ここの文字列には、変数は使用できません。

3.5.4 文字見出し

索引の項目見出しの体裁を指定します

```
文字見出し
  書体
            FutoMinA101-Bold
                           # 和文書体
                      # 単位:Q/pt
  級数
  字送り
            12歯
                      # 単位:歯/pt/mm
                                # 欧文書体
  欧文書体
           PCCenturyOldStyle-Regular
            120
                      # 単位:Q/pt
  欧文級数
            0.03字
  ベースライン
                      # 単位:字/歯/pt/mm
            # 大きな値にするほど欧文が下がります。
  文字色シアン
            0.0
                      # CMYKでシアン成分を0.0~1.0で指定
  文字色マゼンタ 0.0
                      # CMYKでマゼンタ成分を0.0~1.0で指定
  文字色イエロー 0.0
                      # CMYKでイエロー成分を0.0~1.0で指定
                      # CMYKでブラック成分を0.0~1.0で指定
  文字色ブラック 1.0
  字詰め
           19字
                     # 単位:字/歯/pt/mm
  行送り
                     # 単位:行/歯/pt/mm
           16行
                     # 単位:行/歯/pt/mm
  前空き
           1行
  後空き
                     # 単位:行/歯/pt/mm
           0行
                     # 単位:行/歯/pt/mm
  行取り
           1行
  字上げ
           0字
                     # 単位:字/歯/pt/mm
  字下げ
            0字
                      # 単位:字/歯/pt/mm
                      # 指定項目:左/中/右
  行揃え
            左
                      # 上:行取りの上端と文字列の上端が一致する
  付き方
            下
                      #中:行取りの中心と文字列の中心が一致する
                      # 下:行取りの下端と文字列の下端が一致する
  最小継続索引行数
                      #後に続く索引がこの行数より少なければ
                      # 改ページする
                       # 記号・数字の見出し文字列
  記号数字見出し "記号・数字"
            大文字
                      # 指定項目:大文字/小文字
  欧文頭文字
  和文頭文字
            カタカナ
                     # 指定項目:カタカナ/ひらがな
文字見出し終り
```

前空き

見出しの前空きを指定します。単位は基本組の行送り(行)/歯数(歯)/ポイント(pt)/ミリ(mm)が使用できます。

後空き

見出しの後空きを指定します。単位は基本組の行送り(行)/歯数(歯)/ポイント(pt)/ミリ(mm)が使用できます。

行取り

見出しの行取りを指定します。単位は基本組の行送り (行) / 歯数 (歯) / ポイント (pt) / ミリ (mm) が使用できます。

行揃え

付き方

行方向の揃えを指定します。" 左、中、右 " のいずれかを指定してください。

行取り指定した見出しに対して、行送り方向の揃えを指定します。"上、中、下"のいずれかを指定してください。

最小継続索引行数

見出しの後の索引が見出しと同じページに指定した行数分収まらなければ、見出しごと改ページします。

記号数字見出し

記号・数字部分の見出しの文字列を指定します。

欧文頭文字

欧文索引部分の見出しを大文字にするか小文字にするかを指定します。

和文頭文字

和文索引部分の見出しをカタカナにするかひらがなにするかを指定します。

3.5.5 カラム間

索引を多段組にした場合の各段間のアキと、段間に引く罫線の幅を設定します。

カラム間

カラム間空き 22歯 #単位:歯/pt/mm

罫線有り無し 有り # 段間の罫線 指定項目:有り/無し

罫線太さ 0.2mm # 単位:mm/歯/pt

カラム間終り

カラム間空き

段間のアキを設定します。単位は歯数 (歯) / ポイント (pt) / ミリ (mm) が使用できます。

罫線有り無し

段間に罫線(段間罫)を引くかどうかを設定します。"有り、無し"のいずれかを指定してください。この罫線は、段間の中心に引かれます。

罫線太さ

罫線の太さを設定します。単位はミリ (mm) / 歯数 (歯) / ポイント (pt) が使用できます。

3.5.6 ページ番号

索引項目のページ番号の体裁を指定します。

ページ番号

区切り1 "--" # ページとページの区切り

例 : 12--14

区切り2 ", " # ページとページの区切り

例: 12, 24, 35

 イタリック書体
 Times-Italic
 # 欧文書体

 イタリック級数
 11Q
 # 単位:Q/pt

イタリックベースライン 0.03字 #単位:字/歯/pt/mm #値を大きくするほど欧文が下がります

ボールド書体 Times-Bold # 欧文書体 ボールド級数 11Q # 単位:Q/pt

ボールドベースライン 0.03字 #単位:字/歯/pt/mm #値を大きくするほど欧文が下がります

以下はPDFでのみ有効

文字色赤0.0# RGBで赤成分を0.0~1.0で指定文字色緑0.0# RGBで緑成分を0.0~1.0で指定文字色青1.0# RGBで青成分を0.0~1.0で指定

ページ番号終り

区切り1

EWB では、索引項目が"12,13,14 ページ"のように継続する場合には、"12-14"のように出力します。このページの継続を表す記号を指定します。

区切り2

EWB では、索引項目が "12, 24, 35 ページ " のように継続しない場合には、"12,24,35 " のように出力します。このページの区切りを表す記号を指定します。

イタリック / ボールド

索引入力の際、ページ番号の書体を"イタリック"あるいは"ボールド"に切り換えられます。この書体や級数は、ここで指定します。

文字色

PDF に変換したとき、索引のページ番号はハイパーリンクが有効になります。このときのページ番号文字色の赤 / 緑 / 青成分を $0 \sim 1$ の範囲で指定します。全て 0 なら黒、全て 1 なら白になります。

3.5.7 レベル

索引の中には、「"OS"という項目の中に、"UNIX"という項目があり、その中の"Linux"という項目は、123ページにある」というように、索引項目に対してレベルを付ける場合があります。 EWB では、"レベル 1"から"レベル 3"までの階層を設定することができ、それぞれのレベルに対して体裁が決められます。

```
レベル1
  字下げ
                 0字
                       # 単位:字/歯/pt/mm
  折れたときの頭字下げ
                       # 単位:字/歯/pt/mm
                 1字
  リーダー罫
                       # リーダー罫の線種
                 2
            # 0 ---
            #2--- 3点リーダー(センター)
            #3--- 3点リーダー(下)
            #4--- 罫線
                      # 単位:mm/歯/pt
  太さ
                 0.1mm
            # 上の項目で罫線を選んだときの罫線の太さ
レベル1終り
```

字下げ

索引項目の字下げを入力します。単位は基本組の和文級数 (字) / 歯数 (歯) / ポイント (pt) / ミリ (mm) が使用できます。

折れたときの頭字下げ

項目が複数行になってしまった時に、2行目以降の字下げ量を指定します。単位は基本組の和文級数(字)/歯数(歯)/ポイント(pt)/ミリ(mm)が使用できます。

リーダー罫

索引項目とページ番号の間に引くリーダー種を指定します。"0(ピリオド) 1(句点) 2(中付き3点リーダー) 3(下付き3点リーダー) 4(罫線)"のいすれかを入力します。

太さ

リーダー罫で 4 (罫線) を選んだ場合に、罫線の太さを入力します。単位はミリ (mm) / 歯数 (歯) / ポイント (pt) が使用できます。

3.5.8 psstedとpsidxstedのエラー

pssted と psidxsted は入力ミスなどでエラーが起きると、画面にエラーメッセージを出力して終了します。ここでは、エラーメッセージとその場合の対処法を説明します。

10 行:"の対応が間違っています

見出しの文字列などで、半角のクォーテーションを入れ忘れている(文字列が閉じていない) 場合に起きるエラーです。入力を確認してください。

12 行:1 行が長すぎます

MS-DOS 上で体裁入力ファイルを作成した場合に、ファイルの最後に"~Z"がはいってしまうことがあります。このようなエラーメッセージが出たら、UNIX 上のエディタで体裁入力ファイルを開き、"~Z"を削除してください。

23 行:シンボルの指定が間違っています

見出しなどのシンボルの指定が間違っています。入力を確認してください。

34 行:右/中/左を指定してください

位置指定の文字列が間違っています。入力を確認してください。

45 行:左上/左下/右上/右下を指定してください

位置指定の文字列が間違っています。入力を確認してください。

56 行:上/中/下を指定してください

位置指定の文字列が間違っています。入力を確認してください。

67 行:全部/先頭/無しを指定してください

図表の"強調の有無"の入力が間違っています。入力を確認してください。

78 行:組方向の指定の誤り

組方向の指定が間違っています。入力を確認してください。

Ryumin-Regular:そのようなフォントはありません

和文書体の指定が間違っています。書体一覧を確認して、もう一度入力してください。

Times-roman:そのような欧文フォントはありません

欧文書体の指定が間違っています。書体一覧を確認して、もう一度入力してください。 ノンブルで指定できる変数は、%頁%だけです.

ノンブル文字列の変数が間違っています。入力を確認してください。

柱で指定できる変数は,%< トリガ > 番号%,%< トリガ > 文字列%だけです

柱文字列の変数が間違っています。入力を確認してください。また、使用されている見出し トリガ名を確認してください。

頁体裁:番号2の見開き右の指定がありません

頁体裁の指定で、"見開き右"の指定がありません。入力を確認してください。

3.6 guisted による体裁編集

X Window System が使用できる環境であれば、体裁入力に guisted というプログラムが使用できます(図 3-3)。各項目の対応が GUI で確認できるため、テキストエディタで体裁入力ファイルを扱うよりも楽に作業ができます。

あらかじめサンプル原稿を用意しておき、実際の組版結果を確認しながら作業すると効率良く 体裁の編集ができます。

quisted では共通設定ファイル (common.dat) 本文体裁入力ファイル (pssted.dat) 索引体裁入

カファイル (psixsted.dat) の3つの設定ファイルを扱います。

各ファイルの内容は次の通りです。

共通設定ファイル 本文と索引で共通する項目(書名/判型/PDF関連設定)と、それぞれの体裁ファイル名 本文体裁入力ファイル 本文の体裁に関する設定(pssted 用) 索引体裁入力ファイル 索引の体裁に関する設定(psidxsted 用)

このプログラムは

guisted

で起動します。

起動すると、体裁リストウィンドウが開きます。リストには0番-共通設定、1番-本文体裁、2番-索引体裁が登録されています。体裁の編集はこのリストから編集画面を呼び出して行います。



図 3-3 guisted 起動画面

3.6.1 guisted **の終了と体裁の保存**

guisted を終了するには、"ファイル"メニューから"guisted の終了"を選択します。その時点で保存されていない設定があれば、その設定を保存するかどうかを聞いてきます。

体裁設定を保存した場合、その体裁からスタイルファイルを作成するかどうかを聞いてきます。また、guisted を終了しないで保存する場合は、体裁リストから体裁を選択した状態で"ファイル"メニューの"保存"を選択します。すべての体裁を保存する場合は"全保存"を選択します。保存時にすでに同じファイル名の体裁ファイルがある場合は上書きするかどうかの確認がなされます。"上書き保存"、"上書き全保存"の場合は確認はありません。

共通設定を保存した場合、その時点の体裁のリストも保存されます。次に共通設定ファイルを 開くと、リストに従って各体裁入力ファイルも読み込まれます。

3.6.2 共通設定の編集

書名や判型といった本文と索引部で共通する項目は共通設定ウィンドウを開いて編集します(図3-4)。設定ウィンドウはリストの0番をダブルクリックするか、選択して右の"編集"ボタンを押すと開きます。

書名等で日本語を使用する場合、ctrl-¥キーで漢字入力が可能になります。漢字入力の終了は使用している漢字変換プログラムの終了方法と同じです。



図 3-4 共通設定ウィンドウ

3.6.3 本文体裁の編集

本文の体裁を編集するには、体裁リストの1番をダブルクリックするか、選択してからリストの右の "編集"ボタンを押します。

すでに体裁入力ファイルがあり、それを編集する場合は"ファイル"メニューボタンの"体裁ファイルを開く"からファイルを選択します。

体裁の各要素は左上の体裁要素メニューボタンから要素を選択して編集画面を呼び出します。

基本設定の編集

体裁ファイルから生成されるスタイルファイル類のファイル名と基本組の編集を行います(図3-5)。書体、字詰め、行送り、行数、基本ページスタイルといった体裁の基本となる項目を設定します。

画面の右でノドや天の空き量を指定します。小口、地空きは判型から自動的に算出されます。

書体ボタンで書体設定ウィンドウが開きます。各書体設定ウィンドウでは書体の色を指定できますが、基本書体には色の指定はできません(黒になります)。

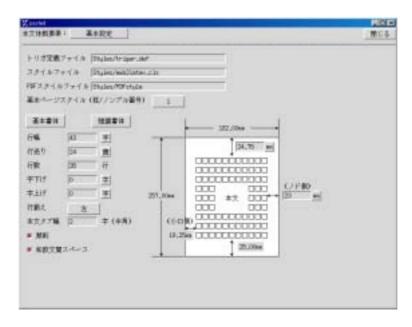


図 3-5 基本組体裁編集画面

柱/ノンブル体裁の編集

柱およびノンブルの編集を行います(図3-6)。上に柱/ノンブルのリストが表示され、右の追加/削除ボタンで増減できます。

柱/ノンブルはそれぞれ左右ページ毎に編集します。左のボタンで編集画面を呼び出し、書体と出力する文字列、原点とそこからの位置、文字列の揃えを指定します。

柱/ノンブルには罫線や EPS ファイルを飾りとして使用できます。右下のリストに要素追加ボタンで修飾要素を追加します。追加すると修飾要素の編集ウィンドウが開きます(図 3-7)。

修飾要素は罫線または EPS ファイルです。罫線の場合は幅/高さと色を、EPS の場合はファイル名と画像サイズを指定します。



図 3-6 柱/ノンブル体裁編集画面

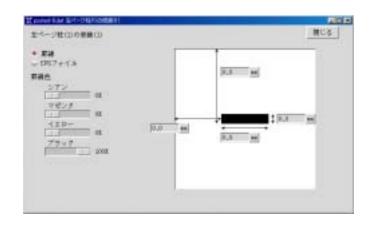


図 3-7 柱/ノンブル修飾編集ウィンドウ

見出し体裁の編集

見出しの体裁指定は文字の位置を前後左右の空きや揃えを指定する方法と、EPS 画像を使用して埋め込まれた文字列と見出し番号/文字列を置きかえる方法があります。

左の "EPS 画像を使用"をチェックしない場合は画面右で前後空き/行取り/字上げ/字下げ/揃えを指定して、出力する文字の位置を指定します(図3-8)。

" EPS 画像を使用"をチェックした場合、EPS ファイルを指定して画像中のタグ文字列を置き換える形で番号や文字列が出力されます(図3-9)。この場合、タグ文字列の位置に対して、実際に出力される文字列の位置を上下左右の揃えで指定します。また、見出し番号部と文字列部の書体

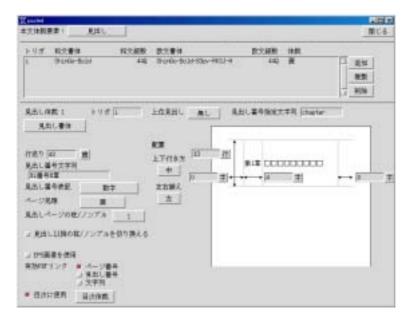


図 3-8 見出し体裁編集画面(文字のみ)

を別個に指定し、それぞれの文字の向き(横/縦)を指定します。

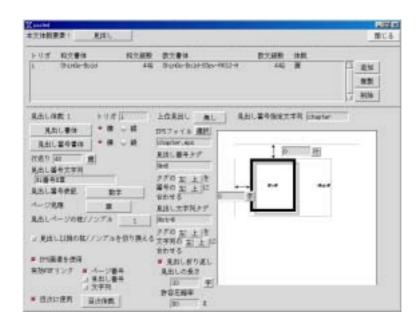


図 3-9 見出し体裁編集画面(EPS画像を使用)

左下の"目次に使用"がチェックしてあれば、その見出しが目次に出力されます。目次の体裁は"目次体裁"ボタンで編集ウィンドウを開いて設定します(図3-10)。

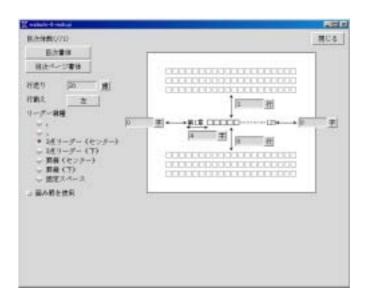


図 3-10 目次体裁編集ウィンドウ

左下の"囲み罫を使用"をチェックすると、囲み罫の設定が可能になります(図 3-11)。

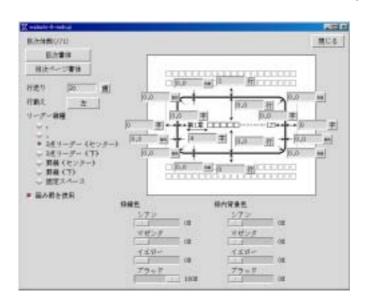


図 3-11 囲み罫付きの目次体裁

小組体裁の編集

"書体を定義して使用"をチェックしていない場合、書体は小組の外側のものがそのまま使用されます(図3-12)。また、"字上げ/字下げの基準"を"周囲環境"にした場合、字下げ/字上げ量は小組の外側の字下げ量に加算されて出力されます。

神蛇養羅士 MC4 你文書件 和文學數 建文書徒 EXH Saule-Light 1% Spain-Light-Spa-Mill-4 itti 被制 NIN. **上分本**[4] 小硬体机 1 # 単体を定義して在月 1個高水電体 149321 - 8 的概念 71 000000 ## 000000 ■ 前張の空きを替える 行響之 1 製品製 金加 0000 1330 **≠上げ/≠下げの基準** 35 -CCCC 44 OUTS+ * NR COOL COLOR 000000000000000 -0.5 IT 00000 ## 00000

囲み罫を使用する場合は、先に必要な囲み罫の体裁を作成してから一覧から選択します。

図 3-12 小組体裁編集画面

箇条書体裁の編集

箇条書体裁の設定は小組と同様です。図 3-13の右の図で箇条書特有の設定を行います。

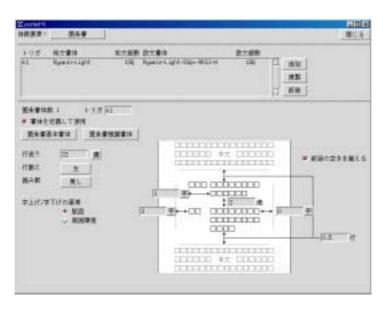


図 3-13 箇条書体裁編集画面

表組体裁の編集

表組体裁の設定は小組と同様です。図 3-14の右の図で表組特有の設定を行います。

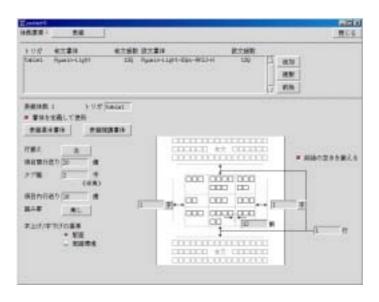


図 3-14 表組体裁編集画面

プログラム体裁の編集

プログラム体裁編集画面では図 3-15のように下段で行番号やインデントの設定を行います。 プログラムリスト中の行は折り返さないため、字上げは指定できません。

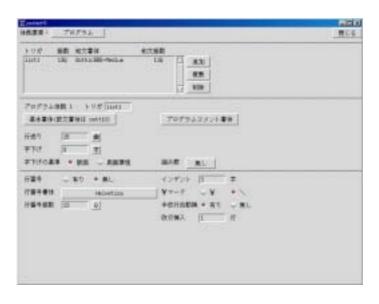


図 3-15 プログラム体裁編集画面

図表キャプション体裁の編集

キャプション書体、番号文字列、文中の参照の強調、ページ処理を指定します。また、文中の参照で PDF リンクとして有効にするものを指定します。図 3-16の右の図で前後左右の空き、図とキャプションの空き、問答を指定します。

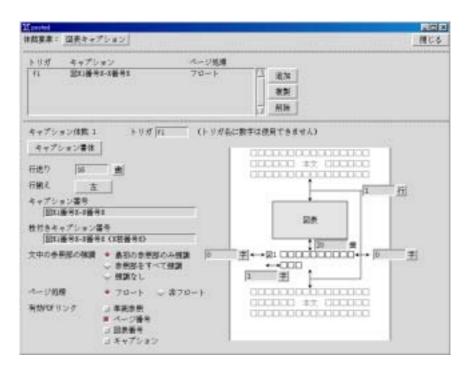


図 3-16 図表キャプション体裁編集画面

脚注体裁の編集

脚注マーク、脚注文字列を指定します。図 3-17の右の図で本文との区切りの罫線や字下げ/字上げ、問答、各上下空きを指定します。

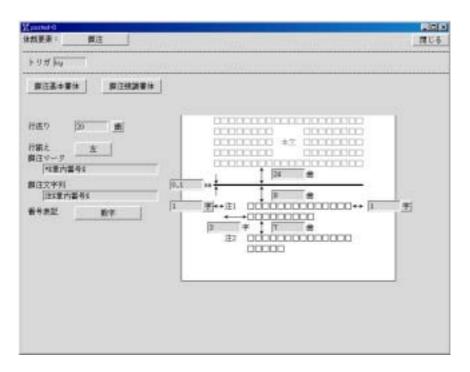


図 3-17 脚注体裁編集画面

キートップ体裁の編集

書体、囲み罫の縦横サイズを指定します(図 3-18)。"角丸"をチェックするとキートップの四隅が丸くなります。

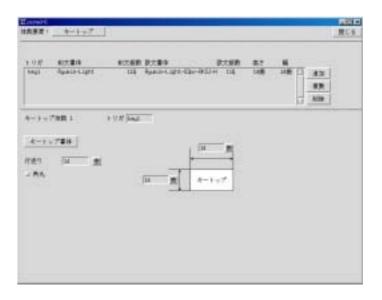


図 3-18 キートップ体裁編集画面

罫線体裁の編集

行間の罫線を定義します(図3-19)。長さ、太さ、前後の空き、字下げ、色を指定します。

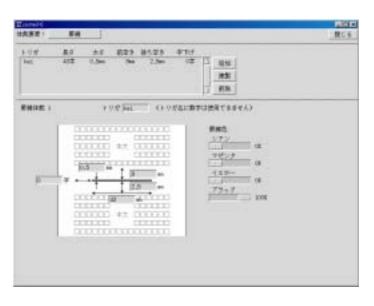


図 3-19 罫線体裁編集画面

囲み罫体裁の編集

囲み罫の編集画面では、まずページ分割させるかどうかをチェックボタンで設定します(図 3-20)。 分割させる場合、罫線の体裁を"全体設定"、"分割時の上端"、分割時の下端"に分けて設定します。各ボタンで設定画面を切り換えます。

"全体"で前後左右の空きや罫線の太さを、"上端"ではページ分割が起こった次のページの開始 位置の上空きや罫線の太さを、"下端"ではページ分割直前の下空きや罫線の太さを設定します。 "上端""下端"で設定されない部分は"全体"のものが適用されます。また、罫線色/罫線内背景 色は"全体""上端""下端"で共通です。

分割させない場合は上端、下端の設定はありません。

ここで作成した囲み罫は小組、箇条書、表組、プログラムリストに使用できます。

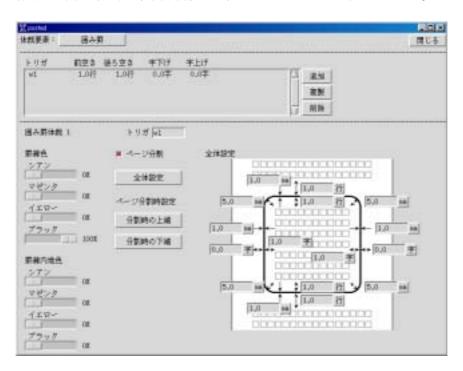


図 3-20 囲み罫体裁編集画面

ルビ体裁の編集

書体、級数比率、ルビ文字の揃え、食いこみ量、本文との空きを指定します(図 3-21)。 ルビ書体の大きさは本文との比率で指定するため、書体設定ウィンドウでの級数指定はできません。また、文字色は指定できません(本文と同色)。

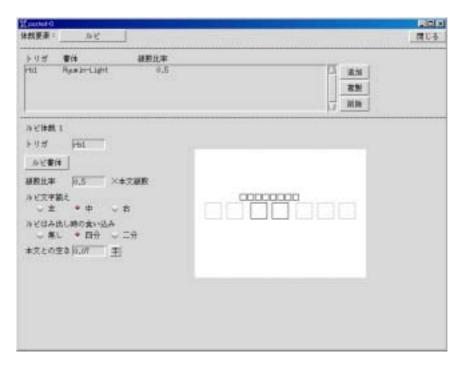


図 3-21 ルビ体裁編集画面

傍点体裁の編集

書体、級数比率、本文との空き、字送り方向の位置の補正量を指定します(図 3-22)。 傍点の大きさは本文との比率で指定するため、書体設定ウィンドウでの級数指定はできません。 また、文字色は指定できません(本文と同色)。

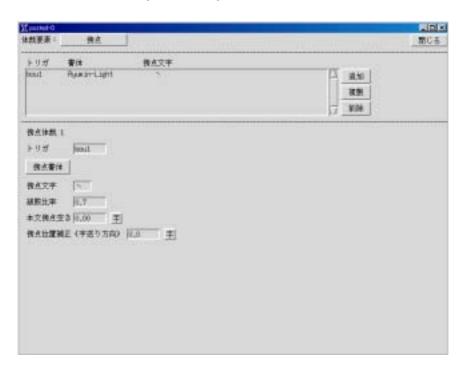


図 3-22 傍点体裁編集画面

3.6.4 索引体裁の編集

索引の体裁を編集するには、体裁リストの2番をダブルクリックするか、選択してからリストの右の "編集"ボタンを押します。

すでに索引体裁入力ファイルがあり、それを編集する場合は"ファイル"メニューボタンの"索引体裁ファイルを開く"からファイルを選択します。

本文体裁と同様に、索引体裁の各要素は左上の体裁要素メニューボタンから要素を選択して編集画面を呼び出します。

基本設定の編集

索引体裁ファイルから生成されるスタイルファイル類のファイル名と索引の基本組の編集を行います(図3-23)、本文体裁と同様ですが、1段組/2段組が選択できます。

本文と同様、基本書体色は黒のみです。

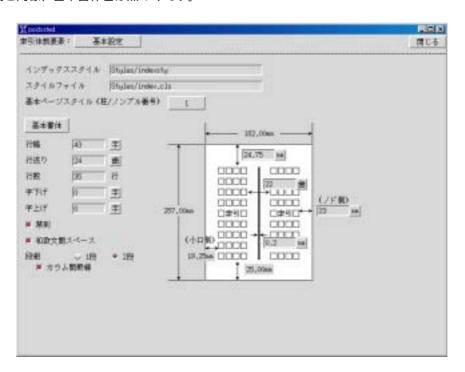


図 3-23 基本組体裁編集画面

柱/ノンブル体裁の編集

柱およびノンブルは本文と同様です。ただし、柱/ノンブル文字列には見出し文字列等は使用できません。

見出し体裁の編集

見出しには本文同様、EPS画像が使用できます(図3-24)。見出し文字列は固定文字列のみです。

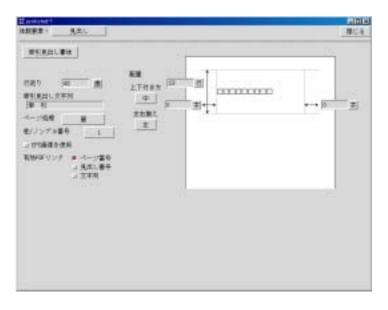


図 3-24 見出し体裁編集画面

文字見出し体裁の編集

索引の各頭文字見出しの体裁です(図3-25)。出力は文字のみです。

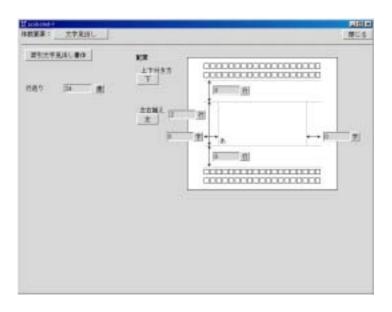


図 3-25 文字見出し体裁編集画面

索引レベル体裁の編集

索引は3レベルまでの階層記述ができます(図3-26)。各階層の出力形式をここで指定します。



図 3-26 索引レベル体裁編集画面

ページ番号体裁の編集

複数ページ記述時の区切り文字とイタリック/ボールド書体を指定します(図3-27)。

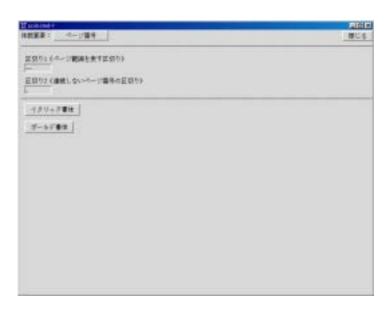


図 3-27 ページ番号体裁編集画面



EWB の操作

設定から簡単な出力まで

EWB は、豊富な機能を実現するために、多くのプログラムから構成されるシステムです。出力のような簡単な処理を行う場合でも、たくさんのプログラムが内部的に起動されます。1つの作業をするのに、複雑な操作を行うのでは、コンピュータを使ったメリットが得られません。そこで、この EWB システムを快適に使えるようにと、"EWB-Shelf"という編集者と EWB との間を取り持つインターフェイスプログラムを用意しました。EWB-Shelf を使うことで、誰でも簡単に、そして効率的に EWB が使えるようになるのです。

この章では、EWB-Shelf の基本的な操作にあたる環境設定やプリンタへの出力などを中心に解説していきます。

4.1 EWB-Shelf とは

EWB-Shelf は、EWBシステムの殻の部分に相当するもので、別名"原稿セット管理システム"と呼ばれています。編集者は、EWB-Shelf に対して簡単な命令を発行するだけで、EWB の機能をすべて利用できます。プリンタへのプリントアウト、X端末を使ったプレビュー、入稿用ファイルの作成などいったことが、コマンド1つで行えます(図4-1)。さらに EWB-Shelf には、台割りの管理などの編集作業を支援する役割りも備わっています。

4.1.1 EWB-Shelf **とディレクトリの深い関係**

EWB-Shelf では、作成する本ごとに作業用ディレクトリを用意して、ファイルを管理します。このディレクトリが、EWB-Shelf にとっての"本"に相当するわけです。すなわち、「ディレクトリの中にあるファイルが"本"を構成する要素である」という考え方で作業を行っていきます。

EWB-Shelf を使うためには、章単位に分けられた原稿ファイルを、作成した作業ディレクトリ

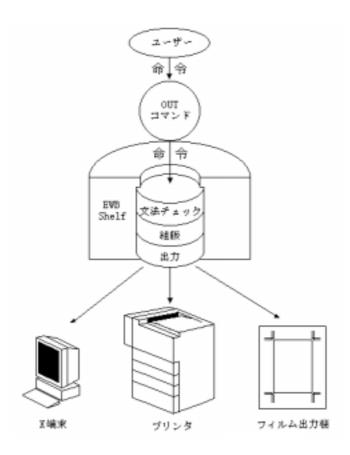


図 4-1 EWB-Shelf の処理の流れ

にコピーし、EWB-Shelf に登録します。この登録された原稿ファイルの集まりのことを原稿セットと呼び、"本の骨組み"に相当します(図 4-2)。また、図や表が EWB で扱える形式のファイルで作成されている場合には、本文原稿と同様に、これらのファイルを作業ディレクトリにコピーします。この時、図表類のファイルを1つのサブディレクトリにまとめておくと、管理がしやすくなります。

4.1.2 EWB-Shelf **の機能**

原稿ファイルを EWB-Shelf に登録すると、EWB-Shelf を通して EWB の持つさまざまな機能を 簡単に利用できます。

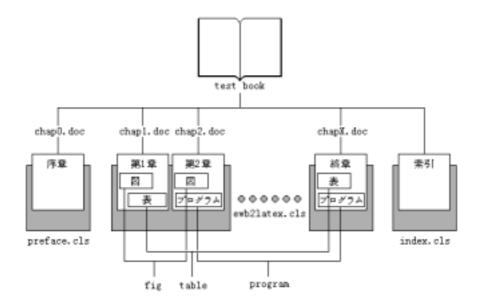


図 4-2 EWB-Shelf における本とファイルの関係

4.1.3 出力

プリンタ出力や X端末を使ったプレビュー、入稿用ファイル作成などの作業を、コマンド 1 つで行えます。

4.1.4 台割りの管理

原稿セットに登録された原稿には、順番に特定の番号が付けられます。EWB-Shelf は、この番号を使って各原稿の先頭ページ番号(ノンブル)を自動的に設定します。たとえば、1章の最終ページが37ページであった場合には、それに続く2章の先頭ページが自動的に計算されて、38ページになります。もちろん、章単位に先頭ページを自由に設定することもできます。

4.1.5 スタイルファイルの設定

前付け用や本文用などといった具合いに、異なったスタイルファイルを作成した場合には、それぞれの原稿に対して、スタイルファイルを設定することができます。

4.1.6 **エラーリカバリー**

プリンタ出力などを行う場合に、トリガの入力ミスが原因でエラーが発生することがあります。この時に、EWB-Shelf は自動的にエディタを起動し、その原稿ファイルを読み込んで、エラーの発生した行にカーソルを移動させます。ここで、入力ミスを修正して、エディタを終了させると、EWB-Shelf は再び同じコマンドを自動的に実行します。

4.1.7 ファイル相互依存関係の把握

EWBでは、ある原稿ファイルを出力するためには、"トリガ定義ファイル"や"スタイルファイル"などのさまざまなファイルを利用します。EWB-Shelf は、原稿ファイルとこれに伴う必要なファイルの関係を常に把握しています。たとえば、原稿をプリンタ出力する場面を考えてみてください。もし、あるスタイルファイルを更新した場合に、それに関係するファイルを1つ1つ確認して出力するのは、大変な労力です。EWB-Shelfでは、1つのスタイルファイルが更新された時に、どの原稿ファイルが出力に影響するかを調べ、影響のある原稿に対して組版処理を行ってから出力してくれます。もちろん、影響のない原稿に対しては、あらためて組版処理は行わず出力します。

4.2 EWB-Shelf のインストール

はじめて EWB-Shelf を使用する場合には、インストールという操作をしなければなりません。 EWB マシンにログインして、"/usr/local/ewb/3.3/ewbshelf/shelf-install" と入力してください。ただし、一度、インストールを行ってしまえば、2回目以降は必要ありません。

% /usr/local/ewb/3.3/ewbshelf/shelf-install EWB-Shelf3.3原稿セット管理システムを組み込みます。

bash用の設定を行います。

ホームディレクトリの .bash_profile の最後に 2 行付け加えます。 よろしいですか? (y/[n]) y ログインシェルが bash であれば次回ログインからEWBが使用できます。

ホームディレクトリの .bash_logout の最後に 1 行付け加えます。 よろしいですか? (y/[n]) y

csh,tcsh用の設定を行います。

ホームディレクトリの .cshrc の最後に2行付け加えます。

```
よろしいですか? (y/[n]) y もとの .cshrc は .cshrc という名前に変更しました。 ログインシェルが csh または tcsh であれば次回ログインからEWBが使用できます。 %
```

これでインストール作業は終りです。念のために、正常にインストールされたかどうかを確認してみましょう。いったん、EWBマシンからログアウトして、もう一度ログインをしてみてください。ログイン時に、次のようなメッセージが表示されたら、正常にインストールされています。

"EWB-Shelfバージョン3.3"が使えます。ヘルプはbook -hです。

もし、このようなメッセージが表示されなかった場合には、もう一度インストール作業を行ってください。

なお、コマンドシェルとして bash を使用している場合は、

shelf: ホームディレクトリにEWB用シェルスクリプト book22383 を作成しました

のようなメッセージが出て、ホームディレクトリに book(番号) というファイルが作成されます。 このファイルは shelf による各種設定を反映させるためのものですので、EWB を使用中は削除しないでください(ログアウト時に自動的に消去されます)。

4.3 EWB-Shelf の操作

一度、インストール作業を行うと、ログインしただけで EWB-Shelf が使えるようになります。 ここでは、実例に従って、原稿登録から出力までの操作を説明します。

4.3.1 原稿の管理

EWBで本を作る時には、まず、作業ディレクトリを作成します。作成するディレクトリ名は、できるだけ書名に関連したわかりやすい名前がよいでしょう。

```
% mkdir testbook
% cd testbook
%
```

このディレクトリに、必要な"原稿ファイル"や"スタイルファイル"、そして"トリガ定義ファイル"をコピーします。正しくコピーされていれば、ディレクトリの内容は、次のようになっているはずです。

```
% ls
            chap4.doc
chap1.doc
                           chap8.doc
                                        test.doc
            chap5.doc
                           chap9.doc
chap10.doc
                                         test15.doc
             chap6.doc
                           diff.sty
chap2.doc
                                         test2.doc
chap3.doc
                           ewb2latex.cls triger.def
             chap7.doc
```

ここで、"book"と入力してみましょう。

```
% book
shelf: 原稿セットが定義されていません
shelf: 環境変数 PATH を変更しました。戻すには 'book -d' を実行してください。
```

このようなメッセージが表示されるはずです。このメッセージが表示されない場合は、EWB-Shelf が正しくインストールされていません。もう一度、インストール操作を行ってください。

なお、EWB-Shelf は最初に book コマンドを使用したときに環境変数 PATH (実行ファイルの検索パス)を変更します。PATH を元に戻すにはログアウトするか "book -d"を入力します。

```
% book −d
shelf: 環境変数 PATH を元に戻しました。
%
```

4.3.2 原稿の登録

まずは、EWB-Shelf の原稿セットに、これから使用する原稿ファイルを登録します。原稿ファイルの登録には、"add、insert、addmulti"という命令を使います。

では、"test.doc"を登録することにしましょう。

```
% book add test.doc
shelf: ワークディレクトリ'Work'が存在しません
shelf: ワークディレクトリを作成しますか? y
shelf: ターゲットディレクトリ'Targets'が存在しません
shelf: ターゲットディレクトリを作成しますか? y
# Body Document file Page Style file Options
1 test test.doc (*1) - tex2
```

EWB-Shelf は、EWB システムが使用する作業用ディレクトリ Work ("ワークディレクトリ"と呼ぶ)と、生成したファイルを保存するためのディレクトリ Targets ("ターゲットディレクトリ"と呼ぶ)を作ります。最初に原稿の登録を行うと、この2つのディレクトリを作成するかどうかを尋ねてきますので、"y"を入力してください。

それでは、続いて"test2.doc"を登録してみましょう。

```
% book add test2.doc

# Body Document file Page Style file Options

1 test test.doc (*1) - tex2

2 test2 test2.doc (?0) - tex2

%
```

このような表示が現われます。ここで、この表示の意味を説明しましょう。

#(原稿番号)登録された原稿には、"原稿番号"という数字が割り振られます。原稿番号は、 EWB-Shelfに対する命令の中で使用されます。

Body (本体名) 原稿ファイル名から拡張子を取ったものを、"本体名"と呼びます。本体名は、 EWB-Shelf に対する命令の中で使われます。

Document file (原稿ファイル名)登録した原稿のファイル名です。

Page (開始ページ) その原稿ファイルの先頭ページ番号 (ノンブル) を表しています。原稿を登録した時点では、未定になっています。

Style file (スタイルファイル)使用するスタイルファイルの名前です。原稿を登録した時点では、"ewb2latex.sty"を使用することになっています。

Options (オプション)原稿に対するその他(オプション)の設定が表示されます。

"add 命令"は、原稿を原稿セットに追加する命令です。この命令では、すでに登録されている原稿の次に付加されます。一方、"insert 命令"は、原稿を原稿セットの好きな位置に挿入する命令です。

たとえば、原稿 "test15.doc"を2番目に挿入してみます。

```
% book 2 insert test15.doc
 # Body Document file
                                    Page Style file
                                                        Options
             test.doc
                                    (*1) -
 1 test
                                                        tex2
                                    (?0) -
 2 test15
             test15.doc
                                                        tex2
 3 test2
              test2.doc
                                    (?0) -
                                                        tex2
%
```

このように"insert 命令"では、挿入したい原稿番号を指定します。

それでは、次に複数の原稿ファイルを、一度に登録する方法を説明します。まとめて原稿ファイルを登録するには、"addmulti"という命令を使います。

たとえば、"chap1.doc, chap2.doc, ..., chap10.doc" という複数の原稿をまとめて登録してみます。

```
% book addmulti chap*.doc

# Body Document file Page Style file Options
1 test test.doc (*1) - tex2
2 test15 test15.doc (?0) - tex2
```

```
3 test2
            test2.doc
                                      (?0) -
                                                            tex2
 4 chap1
                                      (?0) -
            chap1.doc
                                                            tex2
 5 chap10 chap10.doc
                                      (?0) -
                                                            tex2
            chap2.doc
chap3.doc
chap4.doc
chap5.doc
                                      (?0) -
 6 chap2
                                                            tex2
                                      (?0) -
7 chap3
                                                            tex2
8 chap4
                                      (?0) -
                                                            tex2
                                      (?0) -
9 chap5
                                                            tex2
             chap6.doc
10 chap6
                                      (?0) -
                                                            tex2
             chap7.doc
11 chap7
                                      (?0) -
                                                            tex2
            chap8.doc
                                      (?0) -
12 chap8
                                                            tex2
13 chap9
             chap9.doc
                                      (?0) -
                                                            tex2
```

4.3.3 原稿の登録解除

原稿ファイルを原稿セットから解除するには、"delete"という命令を使います。"test.doc, test15.doc, test2.doc"の登録を解除します。この命令は、原稿ファイルを削除するわけでないので、原稿ファイルを失うことはありません。

# Body	Document file	Page Style file	Options
1 test15	test15.doc	(*1) -	tex2
2 test2	test2.doc	(?0) -	tex2
3 chap1	chap1.doc	(?0) -	tex2
4 chap10	chap10.doc	(?0) -	tex2
	ì略		
book delete	e 1		
# Body	Document file	Page Style file	Options
1 test2	test2.doc	(*1) -	tex2
2 chap1	chap1.doc	(?0) -	tex2
3 chap10	chap10.doc	(?0) -	tex2
4 chap2	chap2.doc	(?0) -	tex2
雀	ì略		
book delete	e 1		
# Body	Document file	Page Style file	Options
1 chap1	chap1.doc	(*1) -	tex2
2 chap10	chap10.doc	(?0) -	tex2
3 chap2	chap2.doc	(?0) -	tex2
4 chap3	chap3.doc	(?0) -	tex2

なお、この命令は、原稿ファイル名ではなく原稿番号を指定します。

4.3.4 原稿の順序変更

すでに登録された原稿の順番を変えるには、"move"と"exchange"という命令を使います。 "move"命令は、2つの原稿番号を指定し、最初に指定した原稿番号の原稿ファイルを、2番目に 指定した原稿番号の位置に移動します。

```
% book
 # Body
            Document file
                                  Page Style file
                                                     Options
 1 chap1
            chap1.doc
                                  (*1) -
                                                     tex2
 2 chap10 chap10.doc
                                  (?0) -
                                                     tex2
            chap2.doc
                                 (?0) -
                                                     tex2
 3 chap2
 .....省略
 8 chap7 chap7.doc
                                 (?0) -
                                                     tex2
                                 (?0) -
 9 chap8
            chap8.doc
                                                     tex2
10 chap9
            chap9.doc
                                 (?0) -
                                                     tex2
% book move 2 10
 # Body Document file
                               Page Style file
                                                    Options
 1 chap1
            chap1.doc
                                 (*1) -
                                                     tex2
         chap2.doc
chap3.doc
                                 (?0) -
                                                     tex2
 2 chap2
                                 (?0) -
 3 chap3
                                                     t.ex2
 .....省略
                                 (?0) -
 7 chap7
           chap7.doc
                                                     tex2
 8 chap8
             chap8.doc
                                  (?0) -
                                                     tex2
 9 chap9
             chap9.doc
                                  (?0) -
                                                     tex2
                                  (?0) -
10 chap10
             chap10.doc
                                                     tex2
```

また、"exchange" 命令は2つの原稿の位置を入れ換えます。"move" 命令と同じように2つの原稿番号を指定してください。

4.3.5 原稿出力

ここでは、プリンタ出力とX端末プレビューの方法について説明します。

4.3.6 プリンタ出力

原稿登録が終了したら、さっそく、プリンタに "chap1.doc"を出力してみましょう。プリンタ出力には、"out"または "OUT"という命令を使います。"out"と"OUT"の違いは、EPSファイルが含まれる場合に "out"では EPSファイルをその大きさの枠だけで出力し、"OUT"では図としてそのまま出力します。通常は"OUT"でよいでしょう。"chap1.OUT"と入力します。

```
% chap1.0UT
debug 'ewb2latex -y -f ../triger.def -s ../ewb2latex.cls -P 'cat chap1.fpg' -
T idx.tmp -o chap1.tex ../chap1.doc '
pl-new chap1.tex && pl-new chap1.tex && :
This is pTeX, Version p2.1.9, based on TeX, Version 3.14159 (EUC) (Web2C 7.3.1)
(chap1.tex
pLaTeX2e <1996/03/05>+2 (based on LaTeX2e <1996/06/01> patch level 0)
(../ewb2latex.cls
Document Class: ewb2latex 1996/10/31 19:59 Document style for 'pssted sample fil
e'
*** make dvi for printer ***
```

```
(/usr/local/ewb/3.3/ewbmacros/lib/ewbbase3_3.sty
(/usr/local/ewb/3.3/ewbmacros/lib/cm/OT1cmr.fd)
(/usr/local/ewb/3.3/ewbmacros/lib/cm/OT1cmtt.fd)
(/usr/local/ewb/3.3/ewbmacros/lib/cm/OMLcmm.fd)
  .....省略
(see the transcript file for additional information)
Output written on chap1.dvi (9 pages, 32252 bytes).
Transcript written on chap1.log.
mv chap1.dvi chap1.Dvi
/usr/local/ewb/3.3/bin/indexassign -i idx.tmp -o tmp.idx chap1.idx && mv tmp.idx
chap1.idx
/usr/local/ewb/3.3/bin/dvips -NO -mode supre -D2400 -M -O-10mm,0mm -s -o ../Targ
ets/chap1.PS ../Targets/chap1.Dvi
This is dvipsk 5.86 p1.5c Copyright 1996-2000 ASCII Corp.(www-ptex@ascii.co.jp)
based on dvipsk 5.86 Copyright 1999 Radical Eye Software (www.radicaleye.com)
' TeX output 1996.12.16:1657' -> ../Targets/chap1.PS
dvips: no match for special paper size found; using default
<texc.pro><cmtt9.pfb><cmsy10.pfb><cmex10.pfb><cmmi10.pfb><cmr10.pfb>
<cmtt10.pfb><texps.pro><special.pro><color.pro>. [1] [2] [3] [4] [5] [6] [7]
[8] [9]
pspr ../Targets/chap1.PS
```

" chap1.OUT "という命令で、"トリガの文法解析、組版、プリンタへの出力"という処理が、一度に行われます。この1つの操作で、プリンタから出力されるはずです。複数のプリンタが繋がっていて、プリンタを選んで出力したい場合は、" out "命令に続いてプリンタ名を入力します。" pslp2 "という名前のプリンタに出力してみましょう。

```
% chap1.0UT pslp2
pspr -s-10 -P"pslp2" ../Targets/chap1.PS
```

今度は、"pspr"というプリンタ出力の処理だけが実行されています。この理由は、EWB-Shelf が常に原稿ファイルの更新が行われたかどうかを監視しているからです。"chap1.doc"という原稿ファイルがすでに組版までを終え、それ以後原稿が更新されていないため、同じ"out"命令を実行しても、プリントアウトしか行われないのです。また、一度プリンタ設定をすると、ログアウトまで有効となります。

さらに、出力するページ範囲の指定もできます。" out " に続いて、ページ範囲を入力してください。

```
% chap1.OUT 2 4-6 8
dviselect -i ../Targets/chap1.Dvi 2 4-6 8 > tmp.dvi; dvips -NO -mode supre -D2
400 -o tmp.ps tmp.dvi; pspr -Ppslp2 tmp.ps
[2] [4] [5] [6] [8]
Wrote 5 pages, 18384 bytes
This is dvipsk 5.86 p1.5c Copyright 1996-2000 ASCII Corp.(www-ptex@ascii.co.jp)
based on dvipsk 5.86 Copyright 1999 Radical Eye Software (www.radicaleye.com)
```

```
'TeX output 2000.12.19:1212' -> ../Targets/tmp.ps
dvips: no match for special paper size found; using default
<texc.pro><cmtt9.pfb><cmsy10.pfb><cmr10.pfb><cmtt10.pfb><cmmi10.pfb>
<texps.pro><special.pro><color.pro>. [2] [4] [5] [6] [8]
```

この例では、"chap1.doc の 2, 4~6, 8 ページ "を出力しています。このように、継続するページは "-"(半角ハイフン)を使い、個別のページは "半角スペース"で区切って指定します。

最後に、用紙サイズの指定方法を説明します。"out"命令でプリントアウトすると、通常は"A4 用紙"を使用して出力されますが,この用紙サイズは"form"という命令で変更することができます。ここで、"B4 用紙"を使用して出力する指定をしてみましょう。

```
% book form B4
 :用紙[B4]を選択しました。
          Document file
 # Body
                                  Page Style file
                                                     Options
            chap1.doc
                                 (*1) -
 1 chap1
                                                     tex2
 3 chap2 chap2.doc
                                  (?0) -
                                                     t.ex2
            chap3.doc
                                 (?0) -
 3 chap3
                                                     tex2
 .....省略
 9 chap9
            chap9.doc
                                 (?0) -
                                                     tex2
10 chap10
            chap10.doc
                                  (?0) -
                                                     tex2
```

この例のように、"form"命令は用紙サイズを引数で指定します。用紙サイズには B5、A4、B4、A3 が指定できますが、プリンタがその用紙サイズに対応している必要があります。

また、2ページずつ見開きで印刷することもできます。"form"命令の用紙指定で、用紙サイズの後ろに"W"を付加して指定すると見開き印刷を指定したことになります。

```
% book form A3W
 :用紙[A3](見開き)を選択しました。
 # Body
           Document file
                                  Page Style file
                                                      Options
 1 chap1
             chap1.doc
                                  (*1) -
                                                      tex2
 3 chap2
             chap2.doc
                                  (?0) -
                                                      tex2
                                  (?0) -
 3 chap3
             chap3.doc
                                                      tex2
 .....省略
                                   (?0) -
 9 chap9
             chap9.doc
                                                      tex2
                                   (?0) -
10 chap10
             chap10.doc
                                                      tex2
```

このとき、用紙は横長で使用されます。実際のページの2ページ分に対して用紙サイズが小さ いと、はみ出してしまいますので注意してください。

4.3.7 X 端末でのプレビュー

もし、X 端末を使える状況にあれば、X 端末上でページイメージを確認できます。このプレビューには、" pre "または" PRE "という命令を使用します。プリンタ出力の時と同じように、" pre "は EPS ファイルを枠で表し、" PRE "は EPS ファイルをそのまま表示します。 EPS の表示には時

間が掛かるので、レイアウトの確認だけなら "pre" でいいでしょう。。本体名に続いて ".pre" または ".PRE" と指定します。 "chap1.pre" と入力してみます。

```
% chap1.pre
debug 'ewb2latex -y -e -f ../triger.def -s ../ewb2latex.cls -P 'cat chap1.fpg
'-T idx.tmp -o chap1.tex ../chap1.doc '
pl-new chap1.tex && pl-new chap1.tex &&
This is pTeX, Version p2.1.9, based on TeX, Version 3.14159 (EUC) (Web2C 7.3.1)
(chap1.tex
pLaTeX2e <1996/03/05>+2 (based on LaTeX2e <1996/06/01> patch level 0)
(../ewb2latex.cls
Document Class: ewb2latex 1996/10/31 19:59 Document style for 'pssted sample fil
*** make dvi for previewer ***
(/usr/local/ewb/3.3/ewbmacros/lib/ewbbase3_3.sty
(/usr/local/ewb/3.3/ewbmacros/lib/cm/OT1cmr.fd)
(/usr/local/ewb/3.3/ewbmacros/lib/cm/OT1cmtt.fd)
  .....省略
(see the transcript file for additional information)
Output written on chap1.dvi (9 pages, 32244 bytes).
Transcript written on chap1.log.
/usr/local/ewb/3.3/bin/indexassign -i idx.tmp -o tmp.idx chap1.idx && mv tmp.idx
 chap1.idx
/usr/local/ewb/3.3/bin/dvips -NO -mode supre -D2400 -M -O-10mm,0mm -s -o ../Targ
ets/chap.ps ../Targets/chap1.dvi
dvips -NO -mode supre -D2400 -o ../Targets/chap1.ps tmp.dvi
This is dvipsk 5.86 p1.5c Copyright 1996-2000 ASCII Corp.(www-ptex@ascii.co.jp)
based on dvipsk 5.86 Copyright 1999 Radical Eye Software (www.radicaleye.com)
' TeX output 2000.12.19:1212' -> ../Targets/chap1.ps
dvips: no match for special paper size found; using default
<texc.pro><cmtt9.pfb><cmsy10.pfb><cmex10.pfb><cmmi10.pfb><cmr10.pfb>
<cmtt10.pfb><texps.pro><special.pro><color.pro>. [1] [2] [3] [4] [5] [6] [7]
[8] [9]
ghostview -a4 ../Targets/chap1.ps
```

プレビューは " q " キーで終了します。

プレビューを開始するページ番号の指定ができます。" pre " に続いて、ページ番号を入力してください。

```
% chap1.pre 3
ghostview -a4 -page 3 ../Targets/chap1.ps
```

4.3.8 開始ページ番号の設定

EWB-Shelf は、原稿番号に従って自動的にノンブルを振ります。ここで、原稿セットの状態を見るために、"book"と入力してみます。

```
% book
 # Body
             Document file
                                  Page Style file
                                                      Options
 1 chap1
            chap1.doc
                                   (1) -
                                                      tex2
            chap2.doc
                                 (*10) -
 2 chap2
                                                      tex2
           chap3.doc
                                  (?0) -
 3 chap3
                                                      tex2
 4 chap4
           chap4.doc
                                  (?0) -
                                                      tex2
                                  (?0) -
 5 chap5
            chap5.doc
                                                      tex2
 6 chap6
            chap6.doc
                                  (?0) -
                                                      tex2
 7 chap7
            chap7.doc
                                  (?0) -
                                                      tex2
 8 chap8
            chap8.doc
                                  (?0) -
                                                      tex2
 9 chap9
            chap9.doc
                                  (?0) -
                                                      tex2
                                  (?0) -
             chap10.doc
                                                      tex2
10 chap10
```

" Page " という項目を見てください。" chap1.doc"のところが"(1)"になっています。これは、" chap1.doc"の組版がすでに終り、その先頭ページが1ページであるということを示しています。次に" chap2.doc"を見てみると、"(*10)"となっています。これは、" chap1.doc (chap2.doc の前)"を組版した結果、" chap1.doc"が9ページで終り、これに続く" chap2.doc"は10ページからはじまるということを意味します。" chap3.doc"以降は、"(?0)"となっていますが、これはまだページ番号が未定であることを表しています。このように、EWB-Shelf は1つ前の原稿の組版結果から先頭ページを自動的に決定します。

また、明示的に、章の先頭ページを指定する場合は、"page"という命令を使います。

```
% book 1 page 21
 # Body
          Document file
                                 Page Style file
                                                      Options
 1 chap1
             chap1.doc
                                  *21 -
                                                      tex2
         chap2.doc
chap3.doc
 2 chap2
                                  (*10) -
                                                      tex2
 3 chap3
                                  (?0) -
                                                      tex2
 .....省略
                                  (?0) -
 7 chap7 chap7.doc
                                                      t.ex2
 8 chap8
            chap8.doc
                                  (?0) -
                                                      tex2
 9 chap9
                                  (?0) -
            chap9.doc
                                                      tex2
                                  (?0) -
10 chap10
            chap10.doc
                                                      tex2
%
```

これで、" chap1.doc " の先頭ページが 21 ページになります。明示的に指定した場合は、" Page " 項目の括弧がなくなります。もう一度、" chap1 " を出力してみましょう。

```
% chap1.0UT
debug 'ewb2latex -y -f ../triger.def -s ../ewb2latex.cls -P 'cat chap1.fpg' -
T idx.tmp -o chap1.tex ../chap1.doc '
pl-new chap1.tex && pl-new chap1.tex && :
This is pTeX, Version p2.1.9, based on TeX, Version 3.14159 (EUC) (Web2C 7.3.1)
(chap1.tex
pLaTeX2e <1996/03/05>+2 (based on LaTeX2e <1996/06/01> patch level 0)
(../ewb2latex.cls
```

```
Document Class: ewb2latex 1996/10/31 19:59 Document style for 'pssted sample fil
*** make dvi for printer ***
(/usr/local/ewb/3.3/ewbmacros/lib/ewbbase3_3.sty
(/usr/local/ewb/3.3/ewbmacros/lib/cm/OT1cmr.fd)
(/usr/local/ewb/3.3/ewbmacros/lib/cm/OT1cmtt.fd)
(/usr/local/ewb/3.3/ewbmacros/lib/cm/OMLcmm.fd)
  .....省略
(see the transcript file for additional information)
Output written on chap1.dvi (9 pages, 32252 bytes).
Transcript written on chap1.log.
mv chap1.dvi chap1.Dvi
/usr/local/ewb/3.3/bin/indexassign -i idx.tmp -o tmp.idx chap1.idx && mv tmp.idx
/usr/local/ewb/3.3/bin/dvips -NO -mode supre -D2400 -M -O-10mm,0mm -s -o ../Targ
ets/chap1.PS ../Targets/chap1.Dvi
This is dvipsk 5.86 p1.5c Copyright 1996-2000 ASCII Corp.(www-ptex@ascii.co.jp)
based on dvipsk 5.86 Copyright 1999 Radical Eye Software (www.radicaleye.com)
' TeX output 2000.12.19:1212' -> ../Targets/chap1.PS
dvips: no match for special paper size found; using default
<texc.pro><cmtt9.pfb><cmsy10.pfb><cmex10.pfb><cmmi10.pfb><cmr10.pfb>
<cmtt10.pfb><texps.pro><special.pro><color.pro>. [21] [22] [23] [24] [25]
[26] [27] [28] [29]
pspr -Ppslp2 ../Targets/chap1.PS
% book
            Document file
                                  Page Style file
 # Body
                                                        Options
                                   21 -
            chap1.doc
 1 chap1
                                                        tex2
                                  (*30) -
                                                        tex2
 2 chap2
            chap2.doc
                                  (?0) -
 3 chap3
            chap3.doc
                                                        tex2
                                   (?0) -
 4 chap4
            chap4.doc
                                                        tex2
 5 chap5
            chap5.doc
                                   (?0) -
                                                       tex2
            chap6.doc
                                   (?0) -
 6 chap6
                                   (?0) -
 7 chap7
            chap7.doc
                                                       tex2
 8 chap8
            chap8.doc
                                   (?0) -
                                                        tex2
 9 chap9
            chap9.doc
                                   (?0) -
                                                        tex2
10 chap10
                                   (?0) -
             chap10.doc
                                                        tex2
```

" chap1 " の先頭ページが 21 ページに、" chap2 " の先頭ページが 30 ページになっていることがわかります。

ただし、この命令でページを変更すると入稿のための面付け時に折りとページの関係がおかし くなることがあります。この命令は途中からノンブルの体裁が変わるような本の場合にのみ使用 してください。

4.3.9 スタイルファイルの変更

以前にも述べたように、EWB-Shelf を使うと、複数のスタイルファイルを別々に指定することができます。原稿を組版する時には、通常、本文用のスタイルファイル "ewb2latex.cls"が使われますが、原稿によっては別のスタイルファイルを使って組版しなければなりません。このような

場合には、"style"という命令を使って、スタイルファイルを切り替えます。それでは、"chap10"のスタイルファイルを "appendix.cls"に変更してみましょう。この指定には原稿番号を使います。

```
% book 10 style appendix.cls
 # Body
            Document file
                                  Page Style file
                                                      Options
 1 chap1
            chap1.doc
                                   21 -
                                                      tex2
            chap2.doc
 2 chap2
                                 (*31) -
                                                      tex2
            chap3.doc
                                  (?0) -
 3 chap3
                                                      tex2
            chap4.doc
 4 chap4
                                  (?0) -
                                                      tex2
            chap5.doc
 5 chap5
                                  (?0) -
                                                      t.ex2
            chap6.doc
                                  (?0) -
 6 chap6
                                                      tex2
           chap7.doc
 7 chap7
                                  (?0) -
                                                     tex2
           chap8.doc
                                  (?0) -
 8 chap8
                                                     t.ex2
                                  (?0) -
 9 chap9
           chap9.doc
                                                     tex2
10 chap10
            chap10.doc
                                  (?0) appendix.cls
                                                     tex2
```

4.4 エラーリカバリー

これまでは、原稿入力から出力までの処理がすべてうまくいっているということを前提にして、お話を進めてきました。しかし、大抵の場合は、入力ミスなどでエラーが発生するものです。 ここでは、EWB-Shelf がエラーに対してどのような対処をするのかを見ていきましょう。

4.4.1 トリガの入力ミス

たとえば、"chap1.doc"の中で、"//g{"と入れるべきところを"//G{"と入力していたとします。

.....省略

//g{ダイレクトリンク//g}と//g{モデムリンク//g}の場合には、自分のシステムのシリアル回線(RS-232)を通信媒体として使うことになる。ダイレクトリンクではシステムとシステムとを直接ケーブルで接続し、モデムリンクではケーブルをモデムに接続する。本章では、RS-232のケーブル接続、モデム、シリアル回線の通信パラメータを制御するUNIXのシステムファイルについて解説していく。

//G{ローカルエリアネットワーク(LAN)//g}には数多くの種類があり、それぞれ設定条件が異なる。ローカルエリアネットワークの物理的な設定方法は、本書では扱わない(既存のLANを利用するときのUUCPの設定方法については、次章で述べる)。したがって、詳細についてはシステムに添付されているマニュアルを参照してほしい。

//ii シリアルケーブルの作製

.....省略

この状態で出力してみましょう。

% chap1.out
debug 'ewb2latex -y -f ../triger.def -s ../ewb2latex.cls -P 'cat chap1.fpg' T idx.tmp -o chap1.tex ../chap1.doc '
../chap1.doc(12行):未定義トリガがあります
../chap1.doc(14行):ブロック//g(12行)を閉じていません
../chap1.doc:エラー発生:pass1

このようなエラーが表示がされた後に、エディタ"jvim"(日本語対応 vi)が立ち上がります。

本章では、RS-232のケーブル接続、モデム、シリアル回線の通信パラメータを制御するUNIXのシステムファイルについて解説していく。

//G{ローカルエリアネットワーク(LAN)//g}には数多くの種類があり、それぞれ設定条件が異なる。ローカルエリアネットワークの物理的な設定方法は、本書では扱わない(既存のLANを利用するときのUUCPの設定方法については、次章で述べる)。したがって、詳細についてはシステムに添付されているマニュアルを参照してほしい。

//ii シリアルケーブルの作製

Electronic Industries Association (EIA; 米国電子工業会)によるRS-232C規格(ふつうは単にRS-232と呼ぶ)では、コンピュータや端末とモデムを接続するシリアルケーブルについて規定している。この規格を拡大解釈することによって(規格違反とまではいかないにせよ)、RS-232ケーブルはモデムのみならず、端末、プリンタ、他のコンピュータ上のポートなどあらゆる種類のシリアルデバイスの接続に利用できるようになっている。"/tmp/chap1.doc25558" 497 lines, 50351 characters

そして、エラーのある行の先頭にカーソルが移動します。もし、あなたが "vi"を操作できるのであれば、ここで入力ミスを訂正して、保存と終了を行ってください。すると、EWB-Shelf は再び出力処理を実行しはじめます。このように、EWB-Shelf には入力ミスによるエラーに対して、処理を中断せずにエラーを修正する機能が備わっています。

もし、"vi"が苦手であれば、エラーのある位置をメモして、":q"と入力しリターンキーを押してください。EWB-Shelf は実行を終了します。

デバッグされなかったようなので終了します shelf: 終了ステータス1.終了します %

ここで、好みのエディタを使って入力ミスを修正して、もう一度 " chap1.OUT " と入力してください。



EWB **の操作**

レイアウトから索引作成まで

前章では、原稿ファイルの登録やプリンタ出力、X端末を使ったプレビューなどの操作を説明しました。この章では、さらに一歩進んで、図表類の取り込みから索引作成までの操作を見ていきます。

5.1 図表類の管理

"//f"や"//t"といった図表トリガは、図表のキャプションを定義し、本文中から参照するために使われるものです。図表トリガに、実際どんな図表が貼り付けられるのかは、トリガではわかりません。このような物理的な情報を EWB では、図表の物理属性と呼んでいます。ここでは、この物理属性をどのように扱っていくのかという説明をしていきます。

5.1.1 EWB **で扱える図表**

図表を EWB で扱うためには、ファイルになっている必要があります。この中で、EWB が扱える図表ファイルは、表 5-1のようになります。

これらの図表ファイルの一部は"ファイル拡張子"によって区別されます。

種別	拡張子
EWB コーディングされた表組	.tbl
EWB コーディングされたリスト	.lst
T _E X コーディングされた数式など	.tex
EPS 形式の画像	.eps

表 5-1 EWB で扱える図表ファイル

5.1.2 物理属性ファイルと物理属性エディタ

図表の物理属性は、"物理属性ファイル"という原稿ファイルとは別のファイルに記憶します。このファイルは、図表種別の他に、キャプションの位置やページのどの位置に図表を出力するかなどといった図表に関する様々な情報がはいっています。EWB は、原稿ファイルとこの物理属性ファイルの情報に基づいて、図表のはいったページを出力をします(図5-1)。また、この物理属性ファイルは、EWB-Shelf から起動される"物理属性エディタ"という会話式のツールを使って入力や修正を行います。

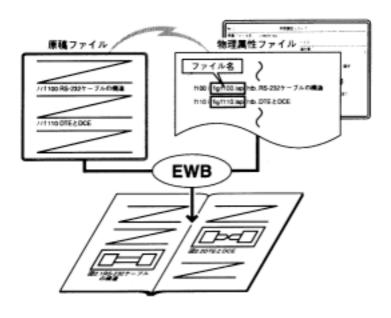


図 5-1 図表トリガと物理属性ファイル

5.1.3 EWB-Shelf で図表を取り込む操作

簡単なサンプル原稿を使って、図表を取り込んだ出力をしてみましょう。サンプル原稿は次のような例です。

chapter 2 //i 接続の手順 //ii データの伝送 25ピンのうち、データ転送に使用するのは、つぎの2つだけである。 2 送信データ; Transmit Data 3 受信データ; Receive Data //c1} これらは、コンピュータとモデムとで異なった用途に使用されている。RS-232規格では 、DTE (Data Terminal Equipment;データ端末装置)とDCE (Data Communicati ons Equipment;データ回線終端装置)の2種類の装置について規定している。コンピュー タの大半は(すべてではないが)DTEであり、モデムは必ずDCEである。 DTEでは、2番ピンでデータを送信し、3番ピンで受信する。DCEでは、この逆になる。 端末やコンピュータをモデムやプリンタに接続する場合(DTE DCE)には、//f0201 のように同じ番号のピン同士を//g{ストレート//g}に接続する。 //f0201 DTEとDCEの相互接続(ストレート) コンピュータとコンピュータとを接続するには(DTE DTE)、2番と3番の線を//g{ク $\Box X//g$ }させなければならない(//f0202)。 //f0202 DTE同士の相互接続(クロス) これは、 $//g{ヌルモデム//g}ケーブルあるいは//g{モデム//g}ケーブルと呼ぶ$ ことが多い。 1番ピンはフレーム(保安用)接地である。コンピュータとモデムを接続する場合はホ スト側に、また、コンピュータ同士を直に接続する場合はいずれか一方の側にこのピンを 接続し、ケーブルのもう一方の側は接続しないままとするべきである。RS-232の正しい実 装は、ポートの1番ピンをシステムのフレーム内部でアースに接続するというものである 。実際RS-232C規格では、両側とも接続し、モデム内部でブレーク(断線)することにな っている。RS-232D規格では、上記のように、DTE側だけで接続する。ほとんどすべてのメ カーが信号用接地とフレーム用接地をモデム内部で相互に接続させているが、これは誤 りである。

この原稿を4章で説明した通りのやり方で、出力します。

% book
shelf: 原稿セットが定義されていません
% book add sample.doc
shelf: ワークディレクトリ'Work'が存在しません
shelf: ワークディレクトリを作成しますか? y
shelf: ターゲットディレクトリでTargets'が存在しません
shelf: ターゲットディレクトリを作成しますか? y
Body Document file Page Style file Options
1 sample sample.doc (*1) - tex2

```
% sample.OUT
debug 'ewb2latex -y -f ../triger.def -s ../ewb2latex.cls -P 'cat sample.fpg'
-T idx.tmp -o sample.tex ../sample.doc '
pl-new sample.tex && pl-new sample.tex && :
This is pTeX, Version p2.1.9, based on TeX, Version 3.14159 (EUC) (Web2C 7.3.1)
(sample.tex
pLaTeX2e <1996/03/05>+2 (based on LaTeX2e <1996/06/01> patch level 0)
(../ewb2latex.cls
Document Class: ewb2latex 1996/10/31 19:59 Document style for 'pssted sample fil
*** make dvi for printer ***
(/usr/local/ewb/3.3/ewbmacros/lib/ewbbase3_3.sty
(/usr/local/ewb/3.3/ewbmacros/lib/cm/OT1cmr.fd)
(/usr/local/ewb/3.3/ewbmacros/lib/cm/OT1cmtt.fd)
(/usr/local/ewb/3.3/ewbmacros/lib/cm/OMLcmm.fd)
  .....省略
(see the transcript file for additional information)
Output written on sample.dvi (2 pages, 2708 bytes).
Transcript written on sample.log.
mv sample.dvi sample.Dvi
/usr/local/ewb/3.3/bin/indexassign -i idx.tmp -o tmp.idx sample.idx && mv tmp.id
x sample.idx
/usr/local/ewb/3.3/bin/dvips -NO -mode supre -D2400 -M -O-10mm,0mm -s -o ../Targ
ets/sample.PS ../Targets/sample.Dvi
This is dvipsk 5.86 p1.5c Copyright 1996-2000 ASCII Corp.(www-ptex@ascii.co.jp)
based on dvipsk 5.86 Copyright 1999 Radical Eye Software (www.radicaleye.com)
' TeX output 2000.12.19:1212' -> ../Targets/sample.PS
dvips: no match for special paper size found; using default
<texc.pro><cmtt9.pfb><cmsy10.pfb><cmex10.pfb><cmmi10.pfb><cmr10.pfb>
<texps.pro><special.pro><color.pro>. [1] [2]
pspr -Ppslp2 ../Targets/sample.PS
```

この例の出力は、図 5-2のようになります。この原稿中にある"図 2.1"と"図 2.2"に対して、Macintosh の Illustrator で作成した図 5-3のような図を入れてみることにしましょう。これらの図には、それぞれ"f2-1.eps"と"f2-2.eps"という名前を付けておきます。



図 5-2 キャプションのみの出力

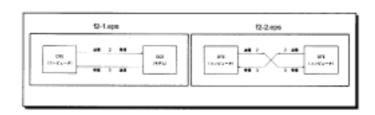


図 5-3 Illustrator で作成した図

5.1.4 図表の登録

まず、最初に行うことは、原稿ファイルと同じように図表ファイルを作業ディレクトリにコピー することです。この時、図表ファイルはサブディレクトリを作って、原稿ファイルと分けて管理 した方が効率的です。

```
% ls
Targets
              diff.sty
                             sample.doc
Work
               ewb2latex.cls triger.def
% mkdir fig
% ls
               diff.sty
                                             triger.def
Targets
                              fig
               ewb2latex.cls sample.doc
% cp ~/lapic/f2-1.lap ~/lapic/f2-2.lap fig
% ls fig
f2-1.lap
               f2-2.lap
%
```

次に、EWB-Shelf に対して「これから図表を取り込みます」という宣言をします。この宣言は次のように行います。

```
% book
 # Body
              Document file
                                     Page Style file
                                                               Options
 1 sample
             sample.doc
                                     (*1) -
                                                               tex2
% book 1 usephy
 # Body
             Document file
                                     Page Style file
                                                               Options
  1 sample
              sample.doc
                                     (*1) -
                                                               usephy tex2
```

この宣言には、原稿番号を指定します。オプションのところに"usephy"という文字列が付け加わってます。これ以降、本体名が"sample"という原稿に対して、図表を取り込んで出力することになります。

5.1.5 **物理属性エディタ** "past" **の操作**

先ほどの"usephy"という命令は、単純に「図表を使います」という宣言をするだけです。実際にどのような図表を扱うのかは、物理属性ファイルに入力していきます。物理属性ファイルへの入力をはじめるには、本体名に続いて".phy"と入力します。

では、"sample.phy"と入力してみましょう。

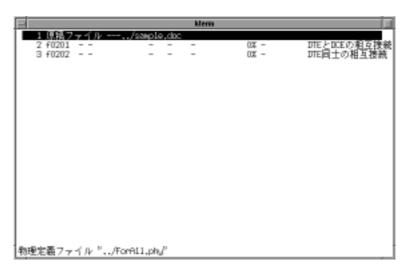


図 5-4 物理属性エディタのリスト画面

画面が一瞬にして切り替わり、図 5-4のようになります。この画面のことを"リスト画面"といいます。このリスト画面には、"sample.doc"という原稿から抽出した図表トリガがそれぞれ表示されています。表示されたリストの中で 1 行が反転していますが、これがリスト画面におけるカーソルとなります。このカーソルは、上下の矢印キーで移動ができます。"f0201"の位置までカーソルを移動し、スペースキーを押すとカーソルの左端に"*"が表示されます。この"*"の

ことをマークと呼び、マークの付いた項目が着目している項目になります(図 5-5)。項目にマークを付けたら、リターンキーを押してください。これで、入力作業のスタートとなります。

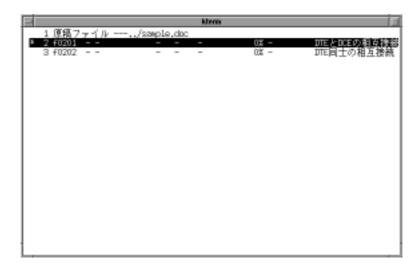


図 5-5 "*"マークの付いたリスト画面



図 5-6 物理属性エディタのメニュー画面

画面が再び切り替わって、図 5-6のようになります。この画面のことを"メニュー画面"と呼びます。メニュー画面は、1 つのトリガに対して 1 つの画面が割り当てられています。画面の上の方には、"原稿ファイル名、トリガ、キャプション"が表示されています。そして、その下に並んでいるのが、入力フィールドです。

まずは、これらの入力フィールドの意味について説明しましょう。

図表の実体

これから扱う図表が "EWB 形式のファイルであるのか、EPS ファイルなのか、それともアタリ罫を引くのか "ということを指定します。

EWB 形式のファイルの場合は"ファイル"を選択し、"ファイル名"を入力します。EPS ファイルの場合には"EPS ファイル"を選択し、"ファイル名"を入力します。このとき縦横のサイズも指定した場合はその大きさで出力します。縦または横のサイズが0の場合は0でない方が有効となり、縦横の比率はそのままで出力します。両方とも0の場合はサイズはそのままで出力します。拡大率を指定するとその拡大率で出力します。縦横のサイズと拡大率では拡大率が優先します。

アタリ罫を引く時には"箱を書く"を選択し、矩形の"縦横サイズ"を指定します。

図表の位置

図表をページのどこに位置させるかということを指定します。"h(ここ) t(ページの上) b (ページの下) p(別ページ) これらの組み合せ"のいずれかを選択します。画面には"固定"という項目がありますが、この項目は将来拡張のために用意したもので、現在は選択できないようになっています。なお、体裁入力でページブレーク"有り"を選択した(フロートしない図表)場合は、ここでどんな項目を選択しても影響がありません。

左あき

図表の左のアキを指定します。"左あき"と"揃え"の関係は、図5-7のようになります。

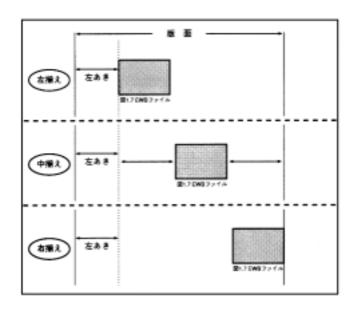


図 5-7 "左あき"と"揃え"の関係

図表の揃え

図表の揃えを指定します。"左、中、右"の中から選択します。

キャプションとのあき

図表とキャプションのアキを指定します。通常はトリガの体裁入力を行う時に、一括でこの 値を設定できますが、ここで行うアキの指定は、個別に行うものです。

キャプションの位置

図表に対して、キャプションをどこに置くのかということを指定します。図表の実体に"ファイル"を選択した場合、キャプション位置は上下の左/中/右のみです。左右には指定できません。

入力フィールドには、文字列を入力するフィールドといくつかの候補から選択するフィールドの2種類があります。入力の方法は、まず上下の矢印キーを使い、入力したいフィールドにカーソルを合わせます。このフィールドが候補から選択するフィールドであれば、左右の矢印キーを使い必要な項目を選択します。また、文字列を入力するフィールドであれば、文字入力をはじめる前にスペースキーを1回押します。文字入力が終ったらリターンキーか下矢印キーを押し、フィールドの内容を確定します。

では、実際に入力してみましょう。ファイル名には、"fig/f2-1.eps"のように、ディレクトリ名も指定します。さらに、図の位置に"htb"を、図の揃えに"中"を、キャプションの位置に"下中"を指定します(図 5-8)。



図 5-8 メニュー画面の入力

入力が終了したら、エスケープキーを 2 回押しリスト画面に戻ります。先ほど付けた "*"のマークが "+"になっています (図 5-9)。この "+"マークは、行の内容に変更があったことを示す印です。この状態で終了キー "q"を押そうとすると、"このまま終了すると変更箇所が失われます (y/n)"と表示されます。この時、"y"を押さなければ、終了しません。

それでは、引続き次の行にカーソルを移動し、"f0202"も同じように入力してみましょう。入力が

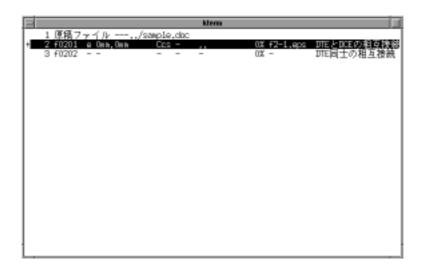


図 5-9 "+"マークが付いたリスト

終了したらリスト画面に戻り、保存キー"s"を押してください。ここで、画面の下に"../ForAll.phy"と表示されます。これが保存するファイル名です。ここでは、そのままリターンキーを押してください(図 5-10)。

リターンキーを押すと、"+"のマークが消えます。この状態で"q"を押すと終了します。



図 5-10 保存中のリスト画面

5.1.6 図表を取り込んだ出力

物理属性の入力が終ったら、さっそく出力してみましょう。今まで通りに、" sample.out " と入力すると図のはいった出力が得られます(図 5-11)。

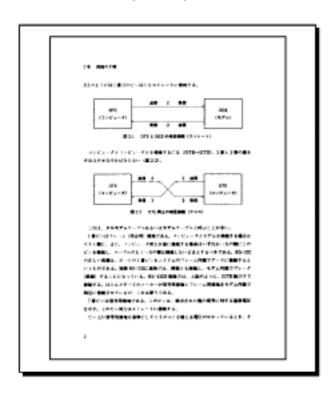


図 5-11 図を取り込んだ出力

5.1.7 アタリ罫の出力

編集中にまだ図版ができていない場合があります。そのような場合、図の大きさのアタリ罫だけでも出力しておくとレイアウトがわかりやすくなります。

例として、横 30mm、縦 30mm のアタリ罫を出力してみましょう。" sample.phy "と入力し、" 60202" を選択して、図 5-12のように入力します。 さらに、キャプションの位置を "右下"にしてみます。 それでは、この設定で、もう一度出力してみましょう(図 5-13)。



図 5-12 アタリ罫の指定



図 5-13 アタリ罫の出力

5.2 レイアウト修正

図表の読み込み出力を行って、このままで版下が得られれば大変に楽なのですが、あともう少

しだけ、作業が残っています。第 1 部の基礎編で説明したように、組版に関連した作業は、すべて EWB システムが処理してくれます。しかし、コンピュータが美しいと考えて組んだページイメージが、必ずしも読みやすく、きれいに見えるとは限りません。このため、出力を見て満足がいかない部分に関しては、修正を加える必要があります。このことを"レイアウト修正"と呼んでいます。

ここでは、"図表の移動"と"組版トリガ"を使った位置調整の説明をします。これが済めば、すぐに入稿ですのでがんばりましょう。なお、この作業は、必ず原稿が完全にフィックスしてから行ってください。

5.2.1 図表の移動

図表には、"フロートしないもの"と"フロートするもの"があります。この指定は体裁入力を行う時に、トリガ別に決められるものです。フロートしない図表は、原稿中で指定された位置に出力され、そのページに収まらない場合は、本文と同様に適切な位置で改ページされます。一方、フロートする図表は、改ページされずに EWB が考えた最適な位置に出力されます。しかし、コンピュータは原稿の内容まで考えて処理を行っているわけではないので、常に最適な位置に図表を配置するとは限りません。そこで、満足のいかない図表に関しては、改めて位置指定や調整をしなければなりません。

このような図表を移動させるには、次の4つの方法があげられます。これらの方法は1でだめなら2、2でもだめなら3といったように順番に試してください。

- 1. "phy "による図表位置の変更
- 2. 図表トリガの位置を変える
- 3. 本文中のアキを調整する
- 4. 図表の大きさを変える

いずれの方法を選択するにしても、前ページのレイアウト変更が後のページに影響するので、 図表の移動は先頭の図表から順番に行ってください。

では、これらの方法について詳しく見ていくことにしましょう。

5.2.2 "phy"**による図表位置の変更**

以前に説明したように、物理属性ファイルで指定できる図表の位置指定は、次の4種類です。

指定 意味

here = = =

top ページの先頭

bottom ページの終り

page 1ページに図表類だけ

これらをそれぞれ単独で指定して、図表が指定した位置にはいらなかった場合には、置く場所がないため、その図表は最終ページに移動してしまいます。さらにその影響で、それに続くすべての図表が最終ページに集められてしまう仕掛けになっています。このため、最初に行う物理属性ファイルの位置指定は、必ず"組み合せ"を選択してください。これが一番効率の上がる方法です。まずは、この指定で出力してみて、満足のいかない図表に対して、先頭の図表から順番に、再度、位置指定を行ってください。

5.2.3 図表トリガの位置を変える

たとえば、どうしてもそのページに入れたい図表が、次のページに出力されてしまう場合などに有効な方法です。原稿中の図表トリガの入力位置を、前ページに出力された文書の中に移動し、図表の位置指定を"top"あるいは"bottom"にします。ただし、見出しを越えて移動してしまうと、図表のナンバリングに影響してしまう場合があるので注意が必要です。

5.2.4 本文中のアキを調整する

ほんの数 mm ぐらい本文中のスペースが足りなくて、図表がはいらない場合などに、見出しや 箇条書などの前後のアキを調整して、図表を入れてしまう方法です。この調整には、縦方向のスペースを調整する"組版トリガ"を使って行います。

5.2.5 図表の大きさを変える

本文中にアキを調整する場所がない場合などに、図表自体の大きさを調整してしまいます。ア タリ罫を引いている場合には、アタリ罫の縦の長さを少し小さくし、ファイルを読み込んでいる 場合には、図表自体の前後のアキを調整します。しかし、この方法はあまりお勧めできません。

5.2.6 組版トリガによるレイアウト修正

EWB が自動的に組んだページレイアウトを調整するための "組版トリガ "について説明していきます。

5.2.7 組版トリガの種類

組版トリガは、編集トリガと異なり"@@"がトリガの開始文字列になっています。もし、原稿中で"@@"を使いたい場合は、"@@@@"と入力してください。また、ほとんどの組版トリガは、トリガ本体の後に半角パーレン"()"を入力します。この半角パーレンの中には、いくつかの引数がはいります。

強制改行 "@@break()", "@@hbreak()"

両者ともに、行を強制的に分割するトリガです。行中の改行したい位置に入力してください。 "@@break()"は折った行を左右均等に割り、"@@hbreak()"は折った行を左詰めにします。

//i 折りたい見出し@@hbreak()文字列

改行の抑制 "@@mbox()"~"@@end"

"@@mbox()"と"@@end"で囲まれた文字列の途中での改行を抑制します。

@@mbox()Windows Programing@@end

強制改ページ "@@clearpage(n)"

強制的に改ページするトリガです。改ページする回数を引数で渡します。"@@clearpage(1)"と入力すると1回改ページを行い、"@@clearpage(2)"と入力すると1回改ページをし、さらに白ページを出力します。もし、"@@clearpage()"のように引数を省略すると、1回改ページをします。

@@clearpage(2) 改ページして白ページを出力

白ページ作成 "@@siropage(n)"

柱もノンブルも無い白ページを作成するトリガです。白ページのページ数を引数で渡します。 "@@siropage(1)"と入力すると1ページ作成され、"@@siropage(2)"と入力すると2ページ の白ページを作ります。もし、"@@siropage()"のように引数を省略すると、1ページの白を 作ります。 @@siropage(2) 2ページの白ページを出力

スペースの調整 "@@vspace(x)"、"@@hspace(x)"

縦と横のスペースを調整するためのトリガです。" @@vspace() " は縦方向のスペースに、" @@vspace() " は横方向のスペース調整に使います。両者ともに引数として調整量を入力します。引数の単位としては、" mm、H (歯) W (字) I (行) " が使えます。引数には小数も使え、正の値を指定すると広がり、負の値を指定すると詰まります。" @@vspace(-0.5I) " のように入力します。

@@vspace(11) 縦方向のスペース調整

@@hspace(0.1mm) 横方向のスペース調整

右揃え "@@right()"~"@@end"

"@@right()"と"@@end"で囲まれた部分を右揃えにします。

@@right()......開始を表すトリガ

@@end終了を表すトリガ

中揃え "@@center()"~"@@end"

"@@center()"と"@@end"で囲まれた部分を中揃えにします。

@@center()......開始を表すトリガ

@@end終了を表すトリガ

左揃え "@@left()"~"@@end"

"@@left()"と"@@end"で囲まれた部分を左揃えにします。

@@left()開始を表すトリガ

@@end終了を表すトリガ

字下げ "@@jisage(n)"~"@@end"

"@@jisage()"と"@@end"で囲まれた部分を字下げします。字下げしたい文字数を、引数として指定します。文字数の単位は基本組の字送りです。

@@jisage(3).....開始を表すトリガ

@@end終了を表すトリガ

表の項目前後のアキ調整 "@@lineheight(x,y)"

表組トリガによって作った表の項目前後のアキを広げるために使います。前と後をどれだけ 広げるかを半角カンマ","で区切られた2つの引数で指定します。単位は、"mm、H(歯) I (行)"が使えます。

//table{ 1 行目の前後を 5H づつ空ける @@lineheight(5H,5H)

//table}

5.2.8 組版トリガを使ってみる

組版トリガを使った組版調整の具体的な例をいくつか見ていきましょう。

飛び出した行の修正

図 5-14を見てください。"bootwait"という単語が版面から飛び出ています。この単語を"boot"と"wait"の間で分割してみます。分割した行が少し広がって、左右均等になっているのがわかります(図 5-15)。

```
つけるときに 抽削 プロセスを使用するもので、 連業は しゃつもとはか
         ぎらない)シリアの個種のデバイス名に対応する。 ボート 取得、)
         ステムによって美なる。詳細については、システムのマニュアルを参
        無してはしい。
         [8-6,a-c] で構成される文字符であり、プロセスのサンレベル (van
        head this interestant made virtain, to
         スントリはアクティブであるとみなりたる。マックしない場合、プロ
         それが実行中であれば当日はこのプロセスを終了を収る。レベルを指
        望しない場合、デフェルトとして大学的 "NESSE" か展示される。of
        のような大学院の場合、そのデバイスと対してはプロセスが実施され
        $100
タイプ
        init がプルモスを扱う方式。off. one. wait, response, book, booksait,
power, powerwait, initidelault ベーチルナル文字符である。これな
        OH This HORBER FUNIX Seaton Administrator's Goldey LA
        あいはこのは現場するシステムのマニュアのとは異様をあている。
プロモス
        エントリがアクティブ映像になったことをも後が開業したときに呼び
        出されるプログラム、シリアル部構の場合、ログインプロンプトを掲
        GFA getty ($500 magetty) FOFTATAA.
```

図 5-14 飛び出した行

•

"が採用される。offのような文字列の場合、そのデバイスに対してはプロセスが起動されない。

タイプ//|initがプロセスを扱う方法。off、once、wait、respawn、boot、boot@break() wait、power、powerwait、initdefaultのいずれかの文字列である。これらのオプションの詳細は『UNIX System Administrator's Guide』(あるいはこれに相当するシステムの

•

つけるときに治療プロセスが開展するもので、過度は りょつもとはか ぎらない シリアル製器のデバイス長に対応する。ボート IDは、リ **ステムによって異なる。根据については、システムのマニュアの有機** 囲してはしい。 DOM: N [かまかっ] で機構される文字列である。アルセスのランシベル (type level である。hot の内容レベルがこのシベルにマッチすると、その エントリはアクティブであるとみなされる。マッチもない場合、アル **ものが実現をであれば (6) はこのプロセスを終了させる。レイスを施** 発しない場合、デフォルトとして文字所でM23490Tが開展される。eff のような文字所の場合、そのアイイスに対してはプロセスが起動され init がデーセスを扱う方法、off. once. wait. respaces, back. Back wait, power, powerwait, inidefault rhi-fits inst 9971A-6. たらのサプションの**West 「UNIX System Administrator's Guidey** (あるいはこれに相称するシステムのマニュアの) に詳値されている。 エントリテアクティア製造になったことを当然が認識したときに呼び おされるプログラム、シラアル回義の場合、ログインフロンアトを発 fill 6 getty (BASS tagetty) 74794786.

図 5-15 "@@break"の利用

見出しの改行

次に、見出しを分割してみましょう。図 5-16を見てください。この行を"の"の後で改行してみます。今度は、"@@hbreak()"を使ってみます。分割された行が左詰めになったことがわかります(図 5-17)。

155 Minory ファイルのトリング

中様のC News マリースでは、自 News 211 のように、この機能が影響的に指し ある。したタップ、このステリデトは不要になるほ子である。

9.5 history ファイルと log ファイルのトリミング

「LEDIE アイレクトリはある3つのファイル Minory、log. emingel. 電テニュー スの機能を設けてもののものである。これルのファイルは急速に大きくなるのの。 トリングラムを使かある。提供で加てはなるない。 Minory ファイルには、システムが変化した変化率が影響がある。ニュース レステムは、このファイルをもとに、どの影響がですに受けるされている。またどの影響を無明的ますマストの特別する。 (上級アイトが優別人を持ちたけ、これからくに 影響になる)。記事が開発するると、expine が影響的に始めなりファイルを指揮する が、このとと呼ばなると、expine が影響的に始めなりファイルを指揮する が、このととはなくがにファリンプを表でいるそのトラッチですったがまかだろ り、自 News 211 の Minory ファイルの特別。のぞのようなものである。 「MINOR」ので、例び7980年の中の表現になった。progresser/MINOR それぞれの情は、まずノッセーには、つぎに支援した日命(UNIX の citize 1)

図 5-16 なりゆきで折れた見出し

.

今後のC Newsリリースでは、B News 2.11のように、この機構が自動的に行われる。したがって、このスクリプトは不要になるはずである。

//ii historyファイルとlogファイルの@@hbreak()トリミング

LIBDIRディレクトリにある3つのファイルhistory、 \log 、errlogは、電子ニュースの動作を記録するためのものである。これらのファイルは急速に大きくなるため、トリミングする必要がある。//g{絶対に忘れてはならない。//g}

.

図 5-17 "@@hbreak"の利用

5.3 目次の生成

目次は見出しを抽出して作ります。あらかじめスタイルファイルで目次として使用されるよう に定義された見出しを、コマンドで一括して抽出して目次ファイルを作成します。

目次の作成には gettoc コマンドを使用します。

% PS % gettoc

ここで大事なのは、全原稿ファイルが組版済みである、ということです。そのため、gettoc を実行する前に全部の文書を PS コマンドでコンパイルしておきます。そして使用されている見出しのうち目次として使用するものだけが抜き出され、contents.ewb というファイルが作られます(リスト 5-1)。

contents.ewb は通常の EWB 原稿ファイルと同じ扱いですので、テキストエディタで編集することも可能です。ただし、もう一度 gettoc を実行すると contents.ewb は変更されるので注意してください (それまで使用していた contents.ewb は contents.ewb ~として保存されます)。

リスト 5-1 contents.ewb

こうしてできた目次ファイルを book コマンドで EWB 文書として登録します。

```
% book 1 insert contents.ewb
```

これで本の先頭に目次が作られることになります。

5.4 索引作成

ここでは、索引作成の作業について触れてみます。索引は項目が出てくるページを拾わなくて はならないため、レイアウト修正が終り、各ページの内容がすべてフィックスしていないと作成 できません。ただし、索引に入れる項目の入力や読みなどは、レイアウトとは無関係ですから、 本文原稿の編集が終った時点から行うと大変効率的です。

5.4.1 索引作成の手順

図 5-18は、EWB での索引作成の手順を示したものです。索引の入力には、"索引トリガ入力"と"索引項目入力"という2つの作業があります。原稿の中に入れる"索引トリガ"は単なる印の意味しか持っておらず、実際に、索引ページに出力される項目は、"索引ファイル"と呼ばれるものに入力します。これらの2つの情報が本文の組版時に結合され、索引出力用の中間ファイルが出来上がります。

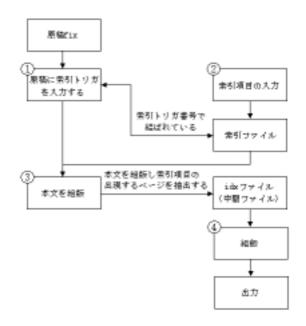


図 5-18 索引作成の手順

索引トリガと索引ファイルの関係は、図表トリガと物理属性ファイルとの関係に似ています。 索引トリガは、原稿のどの位置に索引項目があるのかを示す印で、その実体である索引として出力 する文字列やソーティングに必要な読みは、原稿とは別の索引ファイルに記述する仕掛けになっ ています(図 5-19)。このように索引入力を2つに分けることで、原稿から余計な情報をはずす ことで可読性が上がり、段落や節全体などを要約した索引を入れることができるようになります。

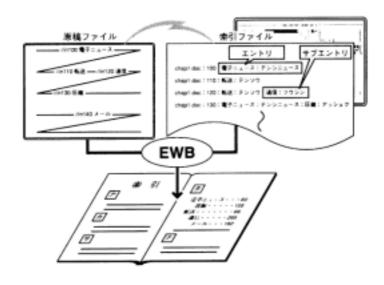


図 5-19 索引トリガと索引ファイルの関係

5.4.2 索引トリガの入力

索引トリガは、本文の出力には影響しないため、原稿中のどんな場所にでも入力することができます。原稿中に"//in 番号"と入力し、この番号が重複しなければ、昇順になっている必要はありません。もし、番号のすぐ後に数字がある場合には、トリガとの区別ができなくなってしまうため、数字の前に半角スペースを1つだけ入れてください。また、1つの文字列を複数の索引として扱いたい場合には、索引トリガを必要な数だけ入力します。次に示す原稿は、"LIBDIR ディレクトリ"に対して3つの索引トリガを入力した例です。

//in10//in20//in30LIBDIRディレクトリにある3つのファイルhistory、 \log 、errlogは、電子ニュースの動作を記録するためのものである。これらのファイルは急速に大きくなるため、トリミングする必要がある。//g{絶対に忘れてはならない。//g}

なお、物理属性ファイルで読み込みを指定した図表ファイルの中には、索引トリガを記述する ことができません。

5.4.3 索引の宣言

索引項目を入力する前に、EWB-Shelfに対して「これから索引を扱います」という宣言をします。図表取り込みの宣言と同じ扱いです。

```
% book
 # Body
             Document file
                                    Page Style file
                                                             Options
 1 sample2 sample2.doc
                                    (*1) -
                                                             tex2
% book 1 useindex
 # Body
            Document file
                                    Page Style file
                                                              Options
 1 sample2 sample2.doc
                                    (*1) -
                                                              useindex tex2
%
```

この宣言には、原稿番号を指定します。オプションのところに"useindex"という文字列が付け加わっています。

さらに、索引出力の宣言をします。通常の原稿と同様に"book add index"などの命令で登録します。

```
% book add index
# Body Document file Page Style file Options
1 sample2 sample2.doc (*1) - useindex tex2
2 index --- (?0) index.cls tex2
```

このとき、原稿の名前は必ず"index"です。索引原稿ファイルの実体は必要ありません。

5.4.4 索引項目の入力

索引トリガの入力と宣言が終ったら、索引項目として出力する文字列の入力を行います。本体名に続いて、".index "と入力します。ここでは、"sample2.index "と入力してみましょう。



図 5-20 索引入力エディタの入力画面

画面が、一瞬にして切り替わり、"索引入力エディタ"が起動します(図 5-20 》。このエディタは、1つの索引トリガに対して、1つの画面が割り当てられます。入力画面の下半分に、注目する索引トリガの行が表示され、着目している索引トリガが反転しています。一方、画面の上半分が入力する部分であり、7つの入力フィールドがあります。最初の状態では、"エントリ"と書かれているフィールドが反転しています。この反転しているフィールドを着目フィールドと呼び、このフィールドの移動には上下左右の矢印キーを使います。

では、それぞれのフィールドの意味を説明しましょう。

エントリ

索引の"1レベル"に出力される文字列を入力します。原稿の入力と同じように半角カナ文字は使えません。

サブエントリ

索引の"2レベル"に出力される文字列を入力します。原稿の入力と同じように半角カナ文字は使えません。省略可能です。

サブサブエントリ

索引の"3 レベル"に出力される文字列を入力します。原稿の入力と同じように半角カナ文字は使えません。省略可能です。

読み

各エントリの読みを入力します。EWB では、この読みの順序で並べ替えを行い、出力します。"全角ひらがな"あるいは"全角カナ"のどちらかに統一して入力してください。エントリに漢字が含まれていない場合は、省略可能です。

コマンド

項目のページ番号を修飾する命令を指定します。"it (イタリック書体) bd (ボールド書体)"のどちらかを指定します。

着目フィールドの入力をはじめる時に、特殊な操作は必要ありません。そのまま文字を入力すると、"文字入力モード"に切り替わり、カーソルが現れます(図 5-21)。このモードでは、左右の矢印キーでカーソル移動ができ、バックスペースキーで文字の削除ができます。文字の入力が終ったら、リターンキーを1回押すと内容が確定し、着目フィールドが次に移動します。既に、フィールド内に文字がある場合には、スペースキーを1回押してください。カーソルが先頭文字に移動し、再び、文字入力モードになります。



図 5-21 文字入力モード

必要なフィールドへの入力が終了したら、"ctrl-N"を押すと、次の索引トリガの入力画面になります。この時に、"ctrl-P"を押すと、1つ前の画面に戻ることができます。また、索引入力工ディタには、文字列検索の機能が用意されています。入力画面で、"ctrl-R"を押すと、検索モードに切り替わります。画面右下の"文字列"というフィールドに、検索したい索引トリガ番号や文字列を入力し、リターンキーを押してください(図 5-22)。検索された索引トリガの入力画面に切り替わります。

すべての入力が終ったら、"ctrl-W"を入力してください。画面下に"../ForAll.index"という表示が現われます。これが保存されるファイル名ですので、そのままリターンキーを押してください。そして、この状態で、"ctrl-E"を入力すると終了します。



図 5-22 検索モード

5.4.5 索引の出力

索引トリガの入力と索引入力エディタによる入力が終了したら、さっそく出力してみましょう。 索引の出力は "index.OUT"と入力します。

```
% book
  # Body
                                     Page Style file
                                                                Options
              Document file
                                     (*1) -
  1 sample2
            sample2.doc
                                                                useindex tex2
                                     (?0) index.cls
  2 index
                                                                tex2
% index.out
debug 'ewb2latex -y -f ../triger.def -s ../ewb2latex.cls -P 'cat sample.fpg'
-T idx.tmp -o sample2.tex ../sample2.doc '
pl-new sample2.tex && pl-new sample2.tex && :
This is pTeX, Version p2.1.9, based on TeX, Version 3.14159 (EUC) (Web2C 7.3.1)
(sample2.tex
pLaTeX2e <1996/03/05>+2 (based on LaTeX2e <1996/06/01> patch level 0)
(../ewb2latex.cls
Document Class: ewb2latex 1996/10/31 19:59 Document style for 'pssted sample fil
*** make dvi for printer ***
(/usr/local/ewb/3.3/ewbmacros/lib/ewbbase3_3.sty
(/usr/local/ewb/3.3/ewbmacros/lib/cm/OT1cmr.fd)
(/usr/local/ewb/3.3/ewbmacros/lib/cm/OT1cmtt.fd)
(/usr/local/ewb/3.3/ewbmacros/lib/cm/OMLcmm.fd)
  .....省略
(see the transcript file for additional information)
Output written on sample2.dvi (1 pages, 2708 bytes).
Transcript written on sample2.log.
mv sample2.dvi sample2.Dvi
indexassign -i idx.tmp -o tmp.idx sample2.idx && mv tmp.idx sample2.idx
mendex -t index.ilg -o tmp.tex -s ../indexsty ../Targets/test.idx ../Targets/ind
```

```
ex.idx
This is mendex version2.3f [10-Dec-96].
Scanning input file ../Targets/sample2.idx....done.(3 entries accepted.)
Scanning input file ../Targets/index.idx....done.(no entry accepted.)
3 entries accepted.
Sorting index....done(4 comparisons).
Sorting pages....done(no comparison).
Making index file....done.
no warning, written in index.ilg.
Output written in tmp.tex.
cp index.fpg tmp.fpg
echo '\n' >> tmp.fpg
ter{page}{\1}\//' < tmp.fpg > tmp.sed
sed -e 's/\begin{document}\\/\\begin{document}\\/' < tmp.sed > tmp2.sed
sed -f tmp2.sed < tmp.tex > tmp2.tex
sed -e 's/\documentclass/\documentclass[printer]/' < tmp2.tex > index.tex
pl-new index.tex
This is pTeX, Version p2.1.9, based on TeX, Version 3.14159 (EUC) (Web2C 7.3.1)
  .....省略
(see the transcript file for additional information)
Output written on index.dvi (2 pages, 2972 bytes).
Transcript written on index.log.
mv index.dvi index.Dvi
/usr/local/ewb/3.3/bin/dvips -NO -mode supre -D2400 -M -O-10mm,0mm -s -o ../Targ
ets/index.PS ../Targets/index.Dvi
This is dvipsk 5.86 p1.5c Copyright 1996-2000 ASCII Corp.(www-ptex@ascii.co.jp)
based on dvipsk 5.86 Copyright 1999 Radical Eye Software (www.radicaleye.com)
' TeX output 2000.12.19:1212' -> ../Targets/index.PS
dvips: no match for special paper size found; using default
<texc.pro><cmtt9.pfb><cmr10.pfb><cmsy10.pfb><cmtt10.pfb><texps.pro>
<special.pro><color.pro>. [29] [30]
psp -Ppslp2 ../Targets/index.PS
```

5.5 **エラー**

EWB-Shelf の使用時に出力されるエラーメッセージとその対処方法を説明します。発生するエラーの種類としては、次の3つがあげられます。

- ・文字入力のエラー
- ・トリガ入力のエラー
- ・その他のエラー

5.5.1 文字入力のエラー

:sample.doc(11 行): T e X では扱えない文字です。終了します

EWB で、扱えない文字(半角カナ文字など)が原稿中に含まれています。原稿の中から、その文字を取り除いてください。

:sample.doc(12 行): T e X では扱えない文字です。げた文字(■)に置き換えます

EWB で、扱えない文字(NEC 外字など)が原稿中に含まれています。ゲタ文字に置き換えて出力されます。

:sample.doc(13 行):写植機で出力できないかもしれません

JIS 第2水準の文字で、写植機から出力できない可能性がある文字が表示されます。

sample.doc(123 行):pass1:1 行が長すぎます

原稿ファイルの最後の行に " ^Z " (0x1a) が入っています。取り除いてください。

5.5.2 **トリガのエラー**

sample.doc(213 行):未定義トリガがあります

未定義トリガがあります。トリガの入力を確認してください。

sample.doc(213 行):ブロック//c1(200 行) を閉じていません

グループトリガを閉じていません。または、グループトリガの範囲に、見出しトリガがあるかもしれません。確認してください。

sample.doc(200 行):200 行の//c1{ を閉じていません

グループトリガを閉じていません。トリガを入力してください。

sample.doc(200 行):sample.doc(180) 行の//c1{ が対応してません

グループトリガの開始と終了の対応が間違っています。確認してください。

sample.doc(200 行):トリガ//c1 は本文の途中に入れられません

小組や箇条書などの開始トリガが行の途中にあります。開始トリガは、行頭に入力してください。

sample.doc(200 行):トリガ c1 の後に中括弧がありません

小組や箇条書などのトリガの直後に中括弧がありません。中括弧を入力してください。

sample.doc(200 行):グルーピングを閉じた直後にトリガがあります

小組や箇条書などの終了トリガの後に、文字あるいはトリガがあります。終了トリガの後に は改行を入れてください。

sample.doc(200 行):トリガ//c1 は//list のグループの中に入れられません

許されていないトリガのネストがあります。トリガがネストしないように修正してください。 sample.doc(200 行)f0201:ラベルが二重定義されています

原稿中の図表トリガの番号が重なっています。

sample.doc(200 行) ラベル (//f0201) が見つかりません

定義されていない番号の図表を参照しています。

sample.doc(200 行):キャプション (//f0201) の中ではタブは使えません

図表キャプションの中にタブがはいっています。タブを取り除いてください。

sample.doc(200 行)://cmt{~//} は一行に書いて下さい

プログラムトリガの中のコメントトリガが複数行になっています。コメントは1行で記述してください。

sample.doc(200 行):組版トリガ@@vspace の引数が間違っています 組版トリガの引数が間違っています。単位を確認してください。

5.5.3 **その他のエラー**

組版時に次のようなメッセージが表示されて、画面が止まることがあります。このような時は、 "x"を入力し、リターンキーを押して実行を停止させてください。

```
% chap02.pre
debug 'ewb2latex -y -f ../triger.def -s ../ewb2latex.cls -P 'cat sample.fpg'
-T idx.tmp -o chap02.tex ../chap02.doc '
pl-new chap02.tex && pl-new chap02.tex &&
This is pTeX, Version p2.1.9, based on TeX, Version 3.14159 (EUC) (Web2C 7.3.1)
(chap02.tex
pLaTeX2e <1996/03/05>+2 (based on LaTeX2e <1996/06/01> patch level 0)
(../ewb2latex.cls
Document Class: ewb2latex 1996/10/31 19:59 Document style for 'pssted sample fil
e,
*** make dvi for printer ***
(/usr/local/ewb/3.3/ewbmacros/lib/ewbbase3_3.sty
(/usr/local/ewb/3.3/ewbmacros/lib/cm/OT1cmr.fd)
  .....省略
[1] [2]
! Undefined control sequence.
1.51 \kexxx
Output written on chap02.dvi (2 pages, 1288 bytes).
Transcript written on chap02.log.
shelf: 終了ステータス1. 終了します
```

このようなエラーが起きる原因は原稿入力のミスではなく、スタイルファイルやトリガ定義ファイルなどにあることが多いので、担当の制作者や EWB の管理者に相談してみましょう。



仕上がりファイル の作成

レイアウトが無事に終了し、索引作成も行ったら、いよいよ仕上がりファイルの作成です。仕上がりファイルは PostScript ファイルですが、目的に応じて大きく2つの作成方法があります。

- ・印刷し、製本することを目的とした面付け PostScript ファイル
- ・PDF 形式へ変換して WEB 等での閲覧を目的としたリンク情報付き PostScript ファイル

この章では、これらの仕上がりファイルを作成する方法について説明します。

6.1 最終チェック

どちらの方法でも、仕上がりファイルを作る前の最終チェックが必要です。まずはチェックの 方法、確認しておくべき点について説明しておきます。

6.1.1 全出力

最終チェックは、最新の出力を使わなければ何の意味もありません。このため、入稿前には、 必ずプリンタによる全出力を行います。まずは、この全出力のやり方について説明します。

全出力のために、まず今までの出力時に作成された作業用ファイルを、"cln"という命令を使って、すべて消去します。この後に、原稿セットに登録されたすべての原稿を出力する"OUT"という命令を入力します。

% book

```
# Body
              Document file
                                    Page Style file
                                                               Options
  1 chap1
             chap1.doc
                                     21 -
                                                               useindex tex2
  2 chap2
              chap2.doc
                                     (25) -
                                                              useindex tex2
                                     (29) -
  3 chap3
              chap3.doc
                                                              useindex tex2
 4 chap4
              chap4.doc
                                     (33) -
                                                              useindex tex2
                                                              useindex tex2
                                     (37) -
 5 chap5
              chap5.doc
                                     (41) -
                                                              useindex tex2
  6 chap6
              chap6.doc
                                     (45) -
                                                               useindex tex2
 7 chap7
              chap7.doc
 8 chap8
              chap8.doc
                                     (49) -
                                                               useindex tex2
 9 chap9
              chap9.doc
                                     (53) -
                                                               useindex tex2
 10 chap10
              chap10.doc
                                     (57) -
                                                               useindex tex2
11 index
                                     (60) index.cls
                                                               tex2
% cln
rm -f chap1.aux chap1.idx chap1.log chap1.tex chap1.fpg chap1.lpg chap1.LOG
/Targets/chap1.Dvi ../Targets/chap1.PS
rm -f chap2.aux chap2.idx chap2.log chap2.tex chap2.fpg chap2.lpg chap2.LOG
/Targets/chap2.Dvi ../Targets/chap2.PS
rm -f chap3.aux chap3.idx chap3.log chap3.tex chap3.fpg chap3.lpg chap3.LOG
/Targets/chap3.Dvi ../Targets/chap3.PS
rm -f chap4.aux chap4.idx chap4.log chap4.tex chap4.fpg chap4.lpg chap4.LOG
/Targets/chap4.Dvi ../Targets/chap4.PS
rm -f chap5.aux chap5.idx chap5.log chap5.tex chap5.fpg chap5.log chap5.LOG
/Targets/chap5.Dvi ../Targets/chap5.PS
rm -f chap6.aux chap6.idx chap6.log chap6.tex chap6.fpg chap6.lpg chap6.LOG
/Targets/chap6.Dvi ../Targets/chap6.PS
rm -f chap7.aux chap7.idx chap7.log chap7.tex chap7.fpg chap7.lpg chap7.LOG
/Targets/chap7.Dvi ../Targets/chap7.PS
rm -f chap8.aux chap8.idx chap8.log chap8.tex chap8.fpg chap8.lpg chap8.LOG
/Targets/chap8.Dvi ../Targets/chap8.PS
rm -f chap9.aux chap9.idx chap9.log chap9.tex chap9.fpg chap9.lpg chap9.LOG
/Targets/chap9.Dvi ../Targets/chap9.PS
rm -f chap10.aux chap10.idx chap10.log chap10.tex chap10.fpg chap10.lpg chap10.L
   ../Targets/chap10.Dvi ../Targets/chap10.PS
rm -f index.aux index.idx index.log index.tex index.fpg index.lpg index.LOG
rm -f ALL.DVI ../Targets/ALL.*
% OUT
debug 'ewb2latex -y -f ../triger.def -s ../ewb2latex.cls -P 'cat chap1.fpg' -
T idx.tmp -o chap1.tex ../chap1.doc '
pl-new chap1.tex && pl-new chap1.tex && :
This is pTeX, Version p2.1.9, based on TeX, Version 3.14159 (EUC) (Web2C 7.3.1)
(chap1.tex
pLaTeX2e <1996/03/05>+2 (based on LaTeX2e <1996/06/01> patch level 0)
(../ewb2latex.cls
Document Class: ewb2latex 1996/10/31 19:59 Document style for 'pssted sample fil
e,
*** make dvi for printer ***
(/usr/local/ewb/3.3/ewbmacros/lib/ewbbase3_3.sty
(/usr/local/ewb/3.3/ewbmacros/lib/cm/OT1cmr.fd)
 .....省略
psp -s-10 -Ppslp2 ../Targets/chap1.PS ../Targets/chap2.PS ../Targets/chap3.PS
.../Targets/chap4.PS .../Targets/chap5.PS .../Targets/chap6.PS .../Targets/chap7.PS
../Targets/chap8.PS ../Targets/chap9.PS ../Targets/chap10.PS
%
```

"OUT"という命令による出力は、長時間に渡って、プリンタを占有します。このため、プリンタの利用者が多い時間帯に出力する時には、今まで通り章単位に、少しずつ間をおきながら出力してください。

6.1.2 チェック項目

プリンタによる全出力が得られたら、チェックをはじめてください。ここでは、いくつかの チェックポイントをあげてみます。もし、少しでも疑問に思うような点があれば、担当の制作者 や EWB の管理者に質問しましょう。

6.1.3 ノンブル

各章のノンブルが台割りと合っているかどうかを確認してください。EWB-Shelf のページ番号 設定などをミスしているかもしれません。

6.1.4 ペースライン補正

和欧文間のベースラインがずれていないかどうかを確認してください。この補正値は、本文や 見出し、箇条書などといった各要素ごとに決めるものです。要素によっては、見落としがあるか もしれませんので注意が必要です。

6.1.5 版面からの飛び出し

英単語などが、版面から飛び出していないかどうかを確認してください。レイアウト修正の時 に見落としているかもしれません。

6.1.6 索引の読み合わせ

索引の読み合わせも行ってください。索引トリガの入力位置によっては、索引ページに出力されたページ番号がずれている可能性があります。もし、ずれていた場合は、索引トリガの位置を修正して、もう一度出力してください。

6.1.7 変な出力はありませんか?

EWB のプリンタ出力は、「書体と解像度の違いを除いて、フィルム出力と全く同じ」です。もし、罫線や文字の出力位置がずれていれば、フィルムで出力しても同じようにずれてしまいます。おかしな箇所があったら、担当の制作者や EWB の管理者に質問してみましょう。

以上、ページ出力での組版チェックが終ったら、いよいよ仕上がりファイル作成の作業に入ります。

6.2 **面付け** PostScript ファイルの作成

まずは、印刷/製本を前提とした面付けファイルの作成方法を説明します。

6.2.1 面付けファイル作成の前に

印刷を前提とした作業の場合、EWB はフィルム出力を最終出力としています。そのため、出力センターに渡す入稿用ファイルはこちらで面付けしたものになります。入稿用ファイルを作成するために、まず面付けの設定をします。面付けの設定は"面付け情報ファイル"に記述します。ファイル名は MENTUKEinfo です。標準的に EWB がインストールされていれば、雛型が"/usr/local/ewb/3.3/MENTUKEinfo"に用意されています。これを作業ディレクトリにコピーして書き換えてしてください。内容は次のようになっています。

```
# psmentuke用設定ファイル
```

指定項目:縦/横 折り単位 16 # 指定項目:8/16 面付け単位 8 # 指定項目:4/8 無線綴じ 無し # 指定項目:有り/無し タイトル "sample book(書名例)" #書名 末尾折り補正 有り # 指定項目:有り/無し # 指定項目:折り/面 ファイル分割 折り

ページモード 絶対ページ # 指定項目:絶対ページ/ノンブル

面付けサイズ トンボサイズ # 指定項目: A4/B4/A3/B3/A2/B2/A1/B1/ # トンボサイズ/指定無し

組方向

本文の組方向を指定します。"縦"、"横"が指定できます。

折り単位

1折りが何ページになるかを指定します。"8"、"16"が指定できます。

面付け単位

1 面が何ページ分になるかを指定します。" 4"、" 8"が指定できます。折り単位に" 16"を指定した場合、面付け単位に" 8"を指定すると表/裏、" 4"を指定すると表左/表右/裏左/に分けて面付けを行います。折り単位に" 8"を指定した場合、面付け単位は" 4"しか指定できません。

無線綴じ

無線綴じのための断ちしろをとるかどうかを指定します。無線綴じの場合は"有り"、そうでなければ"無し"を指定します。

タイトル

書名を指定します。書名にスペースが入る場合は"で囲んで指定します。

末尾折り補正

最後の折りのページ数が半端になった場合、最後を折り単位で指定したページ数にするかど うかを調整します。

たとえば折り単位を"16"に指定したときに12ページ分半端がでた場合、この12ページは8ページの折りと4ページの折りに分けられます。このとき、末尾折り補正に"無し"を指定すると最後の折りが4ページ、その前の折りが8ページで組まれます。"有り"を指定した場合は、最後に16ページの折りを作り、その前は4ページ、さらにその前に8ページの折りを作ります。つまり、最後の折りは強制的に折り単位で指定したページ数となります。

ファイル分割

面付けしたファイルをどのように分割するかを指定します。"折り"を指定するとファイルは1折りごとに分割されます。"面"を指定するとファイルは面ごとに分割されます。このとき、折り単位が"16"、面付け単位が"4"である場合は、1つの折りが"表左面、表右面、裏左面、裏右面"の4つのファイルとして作成されることになります。それ以外の場合は"表面、裏面"の2つが作成されます。

ページモード

面付けのページの取り方を指定します。"絶対ページ"を指定するとノンブルに関係なく、ページの順番のみに依存した並べ方で組みます。"ノンブル"を指定するとノンブルに合った位置にページを並べます。目次と本文が別ノンブルである場合などは"絶対ページ"を、ページが途中で切れてその部分は白ページにしたい場合などは"ノンブル"を指定してください。

面付けサイズ

面付けしたものを出力するフィルムのサイズを指定します。トンボサイズ(面付けのトンボ に合わせる)、A4~B1、指定無しが選択できます。出力センターによっては指定無しだと出力がおかしくなる(欠けて出力される)場合がありますので、できるだけ"トンボサイズ"を 指定してください。

6.2.2 面付けファイルの作成

面付けの準備ができたら入稿用ファイルを作成します。入稿するために必要なファイルは、 "ALL.PS ファイル"と呼ばれるファイルです。"ALL.PS ファイル"は、面付けされた PostScript ファイルです。このファイルは、"ALL.PS"という命令で作成します。では、入力してみます。

```
% book
 # Body
              Document file
                                   Page Style file
                                                             Options
 1 chap1
             chap1.doc
                                    21 -
                                                             useindex tex2
                                    (25) -
 2 chap2
            chap2.doc
                                                             useindex tex2
                                    (29) -
 3 chap3
            chap3.doc
                                                            useindex tex2
                                   (33) -
 4 chap4
            chap4.doc
                                                            useindex tex2
                                   (37) -
 5 chap5
            chap5.doc
                                                           useindex tex2
 6 chap6
            chap6.doc
                                   (41) -
                                                           useindex tex2
 7 chap7
            chap7.doc
                                   (45) -
                                                           useindex tex2
 8 chap8
             chap8.doc
                                   (49) -
                                                           useindex tex2
 9 chap9
             chap9.doc
                                    (53) -
                                                            useindex tex2
10 chap10
                                    (57) -
              chap10.doc
                                                            useindex tex2
11 index
                                    (60) index.cls
                                                             tex2
% ALL.PS
debug 'ewb2latex -y -f ../triger.def -s ../ewb2latex.cls -P 'cat chap1.fpg' -
T idx.tmp -o chap1.tex ../chap1.doc '
pl-new chap1.tex && pl-new chap1.tex &&
This is pTeX, Version p2.1.9, based on TeX, Version 3.14159 (EUC) (Web2C 7.3.1)
(chap1.tex
pLaTeX2e <1996/03/05>+2 (based on LaTeX2e <1996/06/01> patch level 0)
(../ewb2latex.cls
Document Class: ewb2latex 1996/10/31 19:59 Document style for 'pssted sample fil
*** make dvi for printer ***
(/usr/local/ewb/3.3/ewbmacros/lib/ewbbase3_3.sty
(/usr/local/ewb/3.3/ewbmacros/lib/cm/OT1cmr.fd)
(/usr/local/ewb/3.3/ewbmacros/lib/cm/OT1cmtt.fd)
(/usr/local/ewb/3.3/ewbmacros/lib/cm/OMLcmm.fd)
  .....省略
Wrote 13 pages, 48936 bytes
This is dvipsk 5.86 p1.5c Copyright 1996-2000 ASCII Corp. (www-ptex@ascii.co.jp)
based on dvipsk 5.86 Copyright 1999 Radical Eye Software (www.radicaleye.com)
' TeX output 1996.12.16:1829' -> psmentuketmp.ps
dvips: no match for special paper size found; using default
<texc.pro><cmtt10.pfb><cmmi10.pfb><cmr10.pfb><cmsy10.pfb><texps.pro>
<special.pro><color.pro>. [49] [50] [51] [52] [53] [54] [55] [56] [57] [58]
[59] [60] [61]
% ls Targets
ALL.DVI
          ALL.PS.003 chap2.DVI
                                         chap6.DVI
                                                         index.DVI
ALL.PS
          ALL.PS.004 chap3.DVI
                                         chap7.DVI
ALL.PS.001 chap1.DVI chap4.DVI
                                         chap8.DVI
ALL.PS.002 chap10.DVI chap5.DVI
                                         chap9.DVI
%
```

" Targets " というディレクトリに、" ALL.PS. (番号)" というファイルが出来上がります。これが、入稿用のファイル" ALL.PS ファイル"です。MENTUKEinfo ファイルで"ファイル分割"に "折り"を指定した場合は、拡張子の番号は面付け状態での"折り番号"を表します。"面"を指定した場合、16ページ単位の折りで面付け単位が4ページの場合は1折りから"表左面、表右面、裏左面、裏右面"の順で、それ以外の場合は"表面、裏面"の順で番号が付けられています。 さらに、このファイルをプリンタへ出力して面付けの状態をチェックします。"ALL.out"とい

う命令でプリンタ出力します。

```
% ALL.out
mamehon ../Targets/ALL.PS.[0-9]*
Processing ../Targets/ALL.PS.001 ... done.
Processing ../Targets/ALL.PS.002 ... done.
Processing ../Targets/ALL.PS.003 ... done.
Processing ../Targets/ALL.PS.004 ... done.
%
```

面付けされたものをそのままの大きさで出力すると A3 の用紙でも入り切れないので、この出力は縦横がそれぞれ半分の大きさに縮小したものになっています。プリンタから出力されたら面付けの状態をチェックします。ここでのチェックポイントは次のようなものです。

折りとページの関係

台割りと見比べながら、折り番号とページの関係が正しいかチェックします。白ページが入るべきところで入っていなかったりするとノンブルが途中で不連続になり、面付けがおかしくなります。

トンボ、背丁の位置

ページの位置に対してトンボ、背丁の位置がおかしくなっていないかチェックします。

版面からの飛び出し

全出力でもチェックしましたが、ここでもチェックします。全出力では版面からはみ出すと 見えなくなるだけなので見落としがちですが、面付けしてみると隣のページにかぶっていた、 ということがあります。

おかしい部分を修正したら面付けし直して再出力して確認します。特定の折りのみ出力するには "ALL.out 番号" というコマンドを使います。

```
% ALL.out 3
mamehon 3 ../Targets/ALL.PS
Processing ../Targets/ALL.PS.003 ... done.
%
```

この番号は "ALL.PS. 番号 "の番号と同じです。MENTUKEinfo ファイルの "ファイル分割"の 指定によって変わってきますので注意してください。

6.2.3 入稿

これで入稿ファイルはすべてできたわけですが、ファイルを出力センターに渡す前に、使用しているフォントを出力センターが持っているかを確認しておきます。使用フォントの一覧は "fontscan"というコマンドで行います。

% fontscan Targets/ALL.PS.*
Helvetica
Helvetica-Bold
Helvetica-Condensed-Light
Helvetica-Light
Ryumin-regular-H
ShinGo-Bold-H
ShinGo-Light-H
ShinGo-Medium-H
Univers-Condensed
Univers-CondensedBold

表示されたフォント一覧を出力センター側に確認してもらい、持っていないフォントがあれば 原稿ファイルと一緒にフォントファイルを送ってください。

確認ができたら、入稿用ファイルを ftp 等でローカルマシンにダウンロードし、すべて MO にコピーします。

こうしてできた MO を出力センターに送れば入構作業は終了です。

6.3 入稿後の作業

入稿が終了して、青焼きのチェックを行っていると、時々、ミスを発見してしまいます。ここでは、ページの出し直し作業と将来の増刷りや再版に備えた原稿のバックアップについて解説していきます。

6.3.1 ページの出し直し

入稿後の青焼きをチェックしていて、もしミスを見つけてしまった場合には、修正を行わなければなりません。EWBでは、ページ単位の出力しかできませんので、1文字単位の修正であればバラ打ちを行い、切り貼りした方が効率アップにつながります。しかし、この場合に原稿ファイルとの整合性がとれなくなってしまいますので、注意が必要です。また、段落単位あるいは節単位で修正がはいってしまった場合には、それ以降のページのレイアウトに影響が出てしまうため、EWBで再度、出力しなければなりません。

6.3.2 修正と確認

まずは、これまで説明してきた方法で原稿を修正し、プリンタで確認します。文字が正しく修正されているかどうかの確認をしてください。修正により次のページ以降も変わってしまうことがあるので注意してください。

6.3.3 **修正出力用ファイルの作り方**

出力しなおすページが複数ページに渡る場合、折りごとまたは面ごとに面付けファイルを作成して出力した方がよいでしょう。たとえば、20~25ページを出し直す場合、2折り目を出し直せば済みます。この場合は、"ALL.PS"コマンドですべての折りを作り直して"ALL.PS.002"を出し直すことになります。1つの面で済む場合は MENTUKEinfo の"ファイル分割"に"面"を指定して同じく"ALL.PS"コマンドで面付けファイルを作り直します。

修正が 1 ページで済んで、フィルムの切り貼りで対処することになると、そのページだけの PostScript ファイルを作る必要があります。そのための手順を説明します。

1 ページだけのファイルを作成するには、"PS"コマンドを使用します。たとえば"chap3.doc"の 32 ページ目だけのファイルを作りたい場合、"chap3.PS 32"と入力します。

```
% cln
% PS
% chap3.cln
% chap3.PS 32
% 1s Targets
chap3.Dvi chap3.PS.cut
%
```

できた " chap3.PS.cut " が指定のページを抜き出した PostScript ファイルです。このコマンドで連続したページも抜き出せます。20 ページから 25 ページの場合には、" - "を使って、" 20-25 " のように指定します。

こうしてできたファイルをフロッピーディスクまたは MO で入稿すれば完了です。

6.4 PDF 用 PostScript ファイルの作成

最近、文書を PDF 形式にして WEB サイトで閲覧できるようにするという方法が広く利用されています。 EWB の出力は PostScript なのでアドビ社の Acrobat Distiller を使用して PDF の文書へ変換することが可能です。

6.4.1 しおり情報ファイルの作成

PDF ファイルには、任意のページに「しおり」が付けられます。EWB でしおりを付けるには、「しおり情報ファイル」が必要です。しおり情報ファイルは mksiori コマンドで作成します。

% mksiori

これで文中の見出しのしおり情報が siori.dat というファイルに作成されます。

6.4.2 PDF **情報ファイルの作成**

PDF ファイル中には著作者、タイトルといった情報が埋め込まれます。そのための設定ファイルとして PDFinfo というファイルです。

PDFinfo ファイルは以下のような中身のテキストファイルで、雛型として/usr/local/ewb/3.3/PDFinfo が用意されています。

```
<general>
author = "EWBコンソーシアム" # 著作者名
title = "EWBハンドブック" # タイトル
subject = "バージョン3.3" # サブタイトル
pagemode = 2 # PDFを開いたときの閲覧方法
</general>
```

このうち、pagemode は Acrobat Reader 等で PDF ファイルを開いた時の最初の表示方法で、しおりやサムネールを文書本体と一緒に表示するかどうかの設定です。

pagemode の番号は以下のようになっています。

- 1 しおり/サムネールを表示しない
- 2 しおりを表示する
- 3 サムネールを表示する
- 4 文書本体をフルスクリーンモードで表示する

これらの情報を PDFinfo というファイルに記述しておきます。

6.4.3 PostScript ファイルの作成

以上のファイルが用意できたら、PS ファイルを作成します。作成コマンドは ALL.PDF です。

```
% ALL.PDF
debug 'ewb2latex -y -f ../triger.def -s ../ewb2latex.cls -P 'cat chap1.fpg' -
T idx.tmp -o chap1.tex ../chap1.doc '
pl-new chap1.tex && pl-new chap1.tex && :
This is pTeX, Version p2.1.9, based on TeX, Version 3.14159 (EUC) (Web2C 7.3.1)
(chap1.tex
pLaTeX2e <1996/03/05>+2 (based on LaTeX2e <1996/06/01> patch level 0)
```

```
(../ewb2latex.cls
Document Class: ewb2latex 1996/10/31 19:59 Document style for 'pssted sample fil
*** make dvi for printer ***
(/usr/local/ewb/3.3/ewbmacros/lib/ewbbase3_3.sty
(/usr/local/ewb/3.3/ewbmacros/lib/cm/OT1cmr.fd)
(/usr/local/ewb/3.3/ewbmacros/lib/cm/OT1cmtt.fd)
(/usr/local/ewb/3.3/ewbmacros/lib/cm/OMLcmm.fd)
  .....省略
/usr/local/ewb/3.3/bin/dviconcat -o ALL.DVi ../Targets/pre.DVi ../Targets/conten
ts.DVi ../Targets/1-1.DVi ../Targets/index.DVi ../Targets/test.DVi
Wrote 20 pages, 52920 bytes
/usr/local/ewb/3.3/bin/dvips -NO -mode supre -D2400 -M -z -o tmp.ps ALL.DVi
This is dvipsk 5.86 p1.5c Copyright 1996-2000 ASCII Corp.(www-ptex@ascii.co.jp)
based on dvipsk 5.86 Copyright 1999 Radical Eye Software (www.radicaleye.com)
' TeX output 2000.12.19:1240' -> tmp.ps
/usr/local/ewb/3.3/bin/dvips: no match for special paper size found; using defau
<texc.pro><texps.pro><special.pro><color.pro><hps.pro>. <cmsy10.pfb>
<cmtt10.pfb><Cmtt10-Bold.pfb><cmtt9.pfb>[1<FIG/ewbhandbook.eps>] [2] [3] [4]
[5] [6] [7] [8] [9] [10<FIG/f010101.eps>] [11] [12] [13<FIG/f010102.eps>]
[14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28]
[29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [41] [42] [43]
[44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55] [56] [57] [58]
[59] [60] [61] <head.tmp><body.tmp>
/usr/local/ewb/3.3/bin/settopdf -n -i ../PDFinfo -t ../siori.dat -s ../PDFstyle
-o PDF.PS tmp.ps
mv PDF.PS ../Targets
Targets ディレクトリに PDF.PS を作成しました
% ls Targets
ALL.DVi
             chap10.DVI
                              chap4.DVI
                                             chap6.DVi
                                                               chap9.DVI
PDF.PS
              chap2.DVi
                             chap4.DVi
                                             chap7.DVI
                                                              chap9.DVi
chap1.DVI
             chap2.DVI
                             chap5.DVI
                                             chap7.DVi
                                                              index.DVI
                             chap5.DVi
                                             chap8.DVI
                                                              index.DVi
chap1.DVi
             chap3.DVi
chap10.DVI
             chap3.DVI
                             chap6.DVI
                                             chap8.DVi
```

これですべての EWB ファイルをコンパイルし、リンク情報の入った PS ファイルが Targets ディレクトリに PDF.PS という名前で作成されます。

なお、直接 PDF 形式のファイルが作られるわけではないのでご注意ください。 PDF 形式への変換には Adobe Acrobat 等の変換プログラムが必要です。

6.5 **原稿のバックアップ**

EWB は PostScript ファイルを生成するため膨大なディスクスペースを消費します。そのため、すべての作業が終了したら EWB マシンのディスクスペース節約のために作業に使用したファイルを消去します。その前に将来の増刷りや再版時に備えて原稿のバックアップを行ってください。この作業を怠って、原稿を失ってしまっては、電子化の苦労が無駄になってしまいます。自分のファイルは、自分で管理するようにしましょう。

6.5.1 バックアップするファイル

バックアップしなければならないファイルは、次にあげるものです。

- ・トリガ入りの原稿ファイル
- ・読み込んでいる図表ファイル
- ・物理属性ファイル
- ・索引ファイル
- ・スタイルファイル
- ・トリガ定義ファイル
- ・面付け情報ファイル
- ・PDF スタイルファイル
- ・PDF 情報ファイル

その他のファイルは、EWB が中間的に生成するファイルなので、バックアップの必要はありません。バックアップの前に不必要なファイルを、"cln"命令で消しましょう。

これらのファイルを MO 等に保存してバックアップ終了です。



GUI シェルによる EWB **の操作**

EWB は UNIX コマンドをキーボードで入力して使用することが前提で作られていましたが、近年のユーザーインターフェイスのグラフィック化に伴い、新規ユーザーに対してはコマンド入力自体が敷居となってしまいました。

この敷居を低くする意味で EWB を GUI で利用ができるように作成されたインターフェイスが "guishell"です。

7.1 GUI **シェルの準備**

7.1.1 GUI シェルの動作に必要なもの

guishell 自体は EWB サーバ (UNIX) の tcl/tk 上で動作します。X Window を使用するため、ユーザーの端末には X 端末の機能が必要です。

7.1.2 GUI シェルの起動

GUI シェルの起動はコマンドラインから行います。以下のコマンドで起動します。

guishell

起動すると GUI シェルのウインドウが X 端末に表示されます(図 7-1)。

起動後、作業ディレクトリを設定します。GUI シェルのウインドウの上部にある"設定"メニューから"作業ディレクトリ選択"を選びます。するとディレクトリ選択画面が表示されるのでマウスでディレクトリを選択し、目的のディレクトリにたどりついたら"OK"ボタンでその

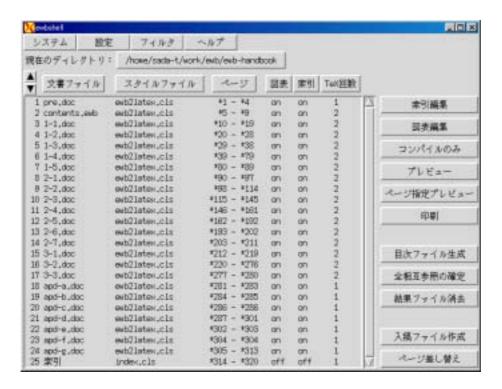


図 7-1 GUI シェル

ディレクトリへ移動します。この作業は"現在のディレクトリ:"の横のボタンでも可能です。 さらに"設定"メニューの"プレビューモード"からプレビュー時の EPS 画像の表示モードを 選択します。"EPS 表示"は文書に貼り付けられた EPS 画像をプレビュー時にそのまま表示します。"EPS 非表示"にすると、EPS 画像の部分がアタリ罫線のみで表示され、プレビューの表示速度が速くなります。

7.2 GUI シェルによる作業

7.2.1 文書の編集

EWBの文書ファイルは通常のエディタで編集できます。また、GUIシェルの中から利用できる EWB エディタを利用して文書を編集することもできます。

7.2.2 文書の登録

EWB で組版を行うには、一般のエディタで入力された文書ファイルを EWB に登録する必要があります。

組版(コンパイル)は登録された文書ファイル毎に行われます。全部をひとつの文書ファイルにまとめてしまうと修正を確認するためのコンパイルに時間が掛かってしまいます。そのため、文書ファイルは章ごとに分けた方が効率が良いでしょう。

文書ファイルのファイル名には漢字は使用できません。また、拡張子は ".ewb "".txt "".doc "のいずれかにして、作業ディレクトリに置いてください。

EWB の組版は画面上のリスト順に行われ、その順にノンブルが自動的に振られます。 文書の登録/削除は"文書ファイル"ボタンから行います。

・文書追加

文書を末尾に登録します。

・文書削除

選択した文書をリストから削除します。文書ファイルがこの操作により消えるわけではありません。

・文書挿入

選択した文書の前に文書を登録します。

・文書移動

文書の登録位置を変更します。登録位置は左側の番号で指定します。移動は文書を選択して """"ボタンを押すことでも行うことができます。

・索引登録

索引を末尾に登録します。索引はファイルの実体を持たず、他の文書中に索引として登録された単語のリストが自動的に索引になります。そのため、索引は位置だけを指定することになります。"索引登録"で索引を使用することを EWB に知らせ、"文書移動"や""""を使って位置を指定します。

7.2.3 スタイルファイルの指定

EWB は文書ファイルから目的の体裁を得るためにスタイルファイルというものを使用します。 スタイルファイルにはページサイズや書体、見出しや小組の体裁、ノンブルや柱の体裁が記述されています。このスタイルファイルは pssted コマンドで作成されます。また、guisted でグラフィカルにパラメータを指定して作成することもできます。作成されたスタイルファイルは作業ディレクトリに置き、文書ファイル毎に対応を指定します。

スタイルファイルを指定していない文書には ewb2latex.cls というスタイルファイルが使用されます。普通はこの状態で組版を行えばよいのですが、目次などのようにある個所だけ体裁を変えたい場合にはそのためのスタイルファイルを用意してその文書ファイルへの対応を指定します。スタイルファイルの指定は割り当てたい文書を選択して"スタイルファイル"ボタンから行います。

索引には通常は index.cls というスタイルファイルを使用します。索引用のスタイルファイルは psidxsted または guisted で作成します。

7.2.4 TeX **回数の指定**

1回の組版につき、何回 TFX を走らせるかというのが "TeX 回数 "の指定です。

TeX は組版をページ順に行いながら、同時に原稿中の各種情報(見出し、索引、図表の位置)を抽出します。TeX を 1 回走らせただけでは、相互参照の情報がまだ抽出されていないため、文中の図表参照トリガ (//f101 のようなトリガ) は「図 ??」となってしまいます。抽出された情報を相互参照に反映させるには、もう一度 TeX を走らせて組版する必要があります。その結果、図表参照トリガは「図 1-1」となって目的の組版結果が選られるわけです。

文中に相互参照がなければ TFX は 1 回実行するだけで充分です。

7.2.5 **文書のコンパイル**

文書ファイルを元に T_EX を実行して組版を行うことを"コンパイル"と呼びます。 EWB は文書ファイルをコンパイルし、その結果を PostScript ファイルに変換することで、プリンタや画面で組版結果を確認できるようにします。

登録された文書の中から目的の文書を選択し、"コンパイル""プレビュー""ページ指定プレビュー""印刷"のいずれかのボタンを押すことでコンパイルが行われます。

・コンパイルのみ

文書をコンパイルするだけです。これにより文書中の各種情報 (索引、見出しなど)が抽出されます。直接結果が見れるものではありません。

・プレビュー

文書をコンパイルし、結果を画面上に表示します。ページの移動はスペースキーと'B' キーで行います。

・ページ指定プレビュー

文書をコンパイルし、指定されたページをを画面上に表示します。コンパイル前にページ番号を問い合わせてきますので見たいページを指定します。

印刷

文書をコンパイルし、結果をプリンタで印刷します。印刷時に使用するプリンタと用紙サイ

ズ、出力するページ番号を指定します。

なお、これらのボタンを押してもコンパイルが行われず、すぐに結果が返ってくることがあります。これは前回のコンパイル以降に目的の文書が変更されておらず、再コンパイルの必要がないと判断されたためです。

強制的にコンパイルを行うためには "結果ファイル消去"ボタンで目的の文書の組版結果を消去してください。

7.2.6 **フィルタ**

選択してある文書について、"フィルタ"メニューから以下のようなフィルタが使用できます。

- ・カタカナ語の一覧 カタカナで始まり、カタカナ、中黒 (・)、音引き (ー) が続く全角文字のみの文字列の一覧を
- 表示する。
 ・英単語の一覧

英字またはアンダースコア (_) で始まり、英字、数字、アンダースコア、ピリオド (.)、マイナス (-) が続く 2 文字以上の半角文字のみの文字列の一覧を表示する。 EWB トリガは無視。

7.2.7 ページの指定

EWB では自動的にノンブルを振っていきます。しかし、目次と本文で別ノンブルになるような本の場合、ノンブルの表記が変わる個所で開始ページを指定する必要があります。

開始ページの指定は"ページ"ボタンで行います。 ノンブルの表記が変わる章を選択し、"ページ"ボタンを押してページ番号を指定します。 自動ノンブルに戻す場合は"なりゆき"を指定します。

なお、このページ指定は別ノンブル以外の場合はできるだけ使用しないようにしてください。 ページとページの間に空きができてしまうと、面付けしたときにページの割付が狂ってしまうお それがあります。

空きページを作る場合は文書ファイル中に「@@clearpage()」または「@@siropage()」を入れることで調整してください。

7.2.8 目次の作成

EWB では見出しを自動的に抽出し、目次を作成することができます(目次に対応したスタイルファイルが必要です)。

手順は以下のようになります。

- 1. すべての文書を作成、登録しコンパイルする すべての見出しを抽出します。
- 2. "目次ファイル作成"ボタンを押すと contents.ewb というファイルが生成される この contents.ewb には目次に必要な見出しが含まれています。必要であれば contents.ewb を 編集します。
- 3. contents.ewb を "文書ファイル"ボタンで目次を入れたい場所に登録する

のノンブルを正しくするためにすべての文書を一度コンパイルします。

- 4. すべての文書をコンパイルしなおす contents.ewb を登録するとそれ以降の文書のノンブルがずれてきます。まず、目次以降の文書
- 5. contents.ewb の結果ファイルを消去し、contents.ewb をコンパイルしなおすすべての文書をコンパイルしなおすと本文のノンブルは正しくなりますが、目次以降のページは contents.ewb がコンパイルされたあとに組みなおされるので目次中のノンブルには古いノンブルが入ってしまいます。正しいノンブルを目次に反映させるために contents.ewb の結果ファイルを消去し、もう一度 contents.ewb をコンパイルしなおします。

7.2.9 図表の登録

GUIシェルには物理属性エディタに相当する図表管理ツールが用意されています。

文書を選んで"図表編集"ボタンを押すとその文書中の図表トリガについての編集画面が表れます(図7-2)。

編集画面中のボタンの機能は以下のとおりです。

・ファイル名

トリガがある位置に実際に出力するファイルを指定します。図表ファイルは、

EPS 図版(*.eps) プログラムリスト(*.lst *.ewb) 表(*.tbl *.ewb)

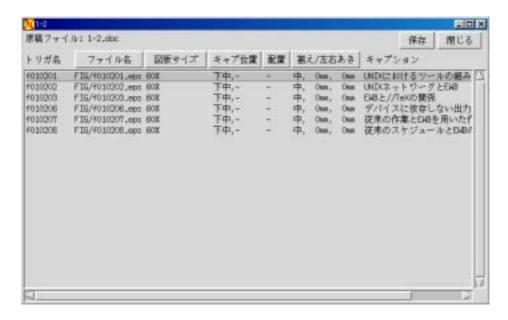


図 7-2 図表の編集

が選択できます。また、図版がまだできていない場合に、代わりにおおよその大きさのアタリ罫を描いておくこともできます。

画像プレビューのチェックボックスを ON にすると、選択中の表 / リストファイルの内容、画像のプレビューが見られます(プレビュー画像を作成するため、最初は表示に時間が掛かります)。

・図版サイズ

図版の大きさを指定します。

大きさは縦横寸法を指定する方法と拡大/縮小率を指定する方法があります。EPS 図版で、縦横寸法のどちらかしか指定しなければもう片方は縦横の比率を変えない大きさになります。 この指定は EPS 画像またはアタリ罫でのみ有効です。

・キャプ位置

図表に対するキャプションの位置と、図表とキャプションとの空きを指定します。リスト/表では左右にキャプションを付けることはできません。

・配置

ページ中に図表をどう配置するかを指定します。指定できるのは

- h (トリガの位置に配置する)
- t (トリガの次のページの先頭に配置する)
- b (トリガのあるページ以降の末尾に配置する)
- p (トリガの次のページ以降に図表だけのページを作って配置する)

の4通りです。これらを優先順位に沿って組み合わせて指定します。通常は http の組み合わせでよいでしょう。

・揃え/左右アキ

本文の幅に対し、図表の左右揃えと左右空きを指定します。揃えは左/センター/右が指定できます。

これらを指定したら、コンパイル前に"保存"ボタンで設定を保存してください。保存しなければ設定は反映されません。

7.2.10 索引の登録

GUI シェルでは専用の EWB エディタを使用して索引の登録ができるようになっています(図7-3)。

索引を登録したい文書を選択し、"文書編集"ボタンを押すと EWB エディタが起動します。

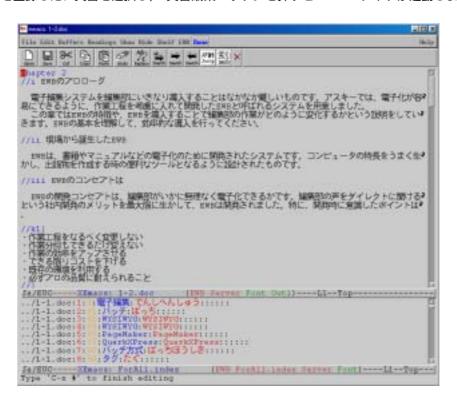


図 7-3 EWB エディタ

索引を登録するには、まず登録したい文中の単語をマウスで選択します(図 7-4)。次に、右上の"索引"ボタンを押します(図 7-5)。

すると、一番下の行に選択された単語が"項目1"として表示されます(図7-6)。これが実際

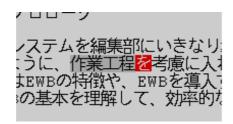


図 7-4 単語の選択



図 7-5 索引ボタン

の索引ページに記載されることになります。変更が必要な場合はこの行の単語を編集します。

```
../1-1.doc:7:90:バッチ
../1-1.doc:8:90:タグ:
Ja/EUC----XEmacs: F
項目1: 作業工程
```

図 7-6 索引項目

索引に記載する単語が決まったら,リターンキーを押します。一番下の行に単語の読みが表示されます(図7-7)。この読みは、索引を読み順に並べ替えるためのものです。読みが正しくない場合は読みの行を修正します。

```
../1-1.doc: 0: 00: QuarkArres
../1-1.doc: 7: 90: バッチ方式:に
../1-1.doc: 8: 90: タグ: たぐ:::
Ja/EUC----XEmacs: ForAll.
読み1: さぎょうこうてい
```

図 7-7 索引の読み

読みが正しく記述できたら、またリターンキーを押します。すると一番下の行で"項目 2"の入力を要求してきます。これは"項目 1"の下にぶらさがる索引になります。必要なければリターンキーのみを入力します。

項目2が入力された場合、さらにその読み、そして項目3の入力が要求されます。必要なけれ

ばやはりリターンキーのみを入力します。

最後に"ページ番号文字種"を聞いてきます(図7-8)。索引でのページ番号を斜体にしたい場合は"it"、太字にしたい場合は"bf"を入力します。通常の文字でよければリターンキーのみを入力します。

```
../1-1.doc:6:85:QuarkXPress:Quar
../1-1.doc:7:90:バッチ方式:ばっちほう
../1-1.doc:8:90:タグ:たぐ:::::
Ja/EUC----XEmacs: ForAll.index
ページ番号文字種(なし/it/bf): ■
```

図 7-8 ページ番号文字種

これで索引が登録できたことになります。文中のマークされた場所のまえに「//in 番号」が挿入されます(図7-9)。なお、索引の登録をキャンセルしたい場合は"索引"ボタンの右の"×"ボタンを押します。

こうして入力された索引は GUI シェル画面で "索引"をコンパイルすることで自動的に集められ、索引ページとして組版されます。

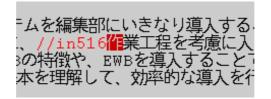


図 7-9 索引の登録完了

索引の登録が終わったら編集した文書を保存し、EWB エディタを終了します。文書の保存は左上の "Save "ボタンで行います(図7-10)。



図 7-10 Save ボタ ン

保存ができたら右上のメニューの右の方にある"Done"で EWB エディタを終了します。

7.2.11 仕上がリファイルの作成

EWB3.2 では最終的なファイルとして PostScript ファイルを作成しますが、その用途によって 3 種類の PostScript ファイルから選択して作成します。

- ・フィルム入稿用に面付けしたもの
- ・PDF 用にリンク情報を付加したもの
- ・書籍サンプル PDF 用に、一部のページを切り出したもの

これらのファイルの作成は、"仕上がりファイル作成"ボタンから行います。

フィルム入稿用ファイルの作成

EWB で作成したファイルをフィルム出力をする場合、出力センターに渡す入稿用ファイルはこちらで面付けしたものになります。一般の DTP ソフトでは面付け作業は出力センターで行いますが、EWB では自分で面付けを行うことになります。

入稿用ファイルを作成するために、面付けの設定をします。設定するパラメータは次のようになっています。書名に日本語を使用する場合、ctrl-¥キーで漢字入力が可能になります。漢字入力の終了は使用している漢字変換プログラムの終了方法と同じです。

- ・タイトル 書名を指定します。
- ・組方向 本文の組方向を指定します。
- ・折り単位 1 折りが何ページになるかを指定します。
- ・面付け単位

1 面が何ページ分になるかを指定します。" 4 "、" 8 " が指定できます。折り単位に " 16 " を指定した場合、面付け単位に " 8 " を指定すると表/裏、" 4 " を指定すると表左/表右/裏左/に分けて面付けを行います。折り単位に " 8 " を指定した場合、面付け単位は " 4 " しか指定できません。

・無線綴じ

無線綴じのための断ちしろをとるかどうかを指定します。無線綴じの場合は"有り"、そうでなければ"無し"を指定します。

・末尾折り補正 最後の折りのページ数が半端になった場合、最後を折り単位で指定したページ数にするかど

うかを調整します。

たとえば折り単位を"16"に指定したときに12ページ分半端がでた場合、この12ページは8ページの折りと4ページの折りに分けられます。このとき、末尾折り補正をチェックしなければ最後の折りが4ページ、その前の折りが8ページで組まれます。末尾折り補正を指定した場合は、最後に16ページの折りを作り、その前は4ページ、さらにその前に8ページの折りを作ります。つまり、最後の折りは強制的に折り単位で指定したページ数となります。

・ファイル分割

面付けしたファイルをどのように分割するかを指定します。"折り"を指定するとファイルは1折りごとに分割されます。"面"を指定するとファイルは面ごとに分割されます。このとき、折り単位が"16"、面付け単位が"4"である場合は、1つの折りが"表左面、表右面、裏左面、裏右面"の4つのファイルとして作成されることになります。それ以外の場合は"表面、裏面"の2つが作成されます。

・ページモード

面付けのページの取り方を指定します。"絶対ページ"を指定するとノンブルに関係なく、ページの順番のみに依存した並べ方で組みます。"ノンブル"を指定するとノンブルに合った位置にページを並べます。目次と本文が別ノンブルである場合などは"絶対ページ"を、ページが途中で切れてその部分は白ページにしたい場合などは"ノンブル"を指定してください。

・面付けサイズ

面付けしたものを出力するフィルムのサイズを指定します。トンボサイズ(面付けのトンボ に合わせる)、A4~B1、指定無しが選択できます。出力センターによっては指定無しだと出力がおかしくなる(欠けて出力される)場合がありますので、できるだけトンボサイズを指 定してください。

これらの設定をしたら"面付け実行"ボタンで面付けします。このとき、ハードディスクのスペースを空けるためにこれまでの組版結果のファイルを一旦消去することになります。

面付け前には自動的にすべての文書のコンパイルが行われます。ページの整合性をとるため、 このコンパイルは3回行われますので多少時間が掛かります。

コンパイルが終わると、使用されているフォントの一覧が表示されます。出力センターによってはこのフォント一覧を知らせておく必要があります。また、独自に作ったフォントを使用している場合はそのフォントファイルを出力センターに渡す必要があります。

面付けが終ったら"豆本印刷"を行って、台割どおりに面付けが行われているかどうかを確認 します。できれば豆本を実際に折ってみて確認することをお勧めします。

正しく面付けされていることが確認できたら、面付けされたファイルを入稿します。入稿するファイルは Targets ディレクトリの中の ALL.PS. 番号 というファイルすべてです。

ファイル名の番号はファイル分割を "折り"にした場合は 1 折り毎に、"面"にした場合は面の順で振られています。"面"で分割した場合、順番は"表""裏"または"左表""左裏""右表""右

裏"になります。

入稿後に一部分のみ修正して差し替える場合は、文書ファイルを修正した後にもう一度面付けをやりなおします。この場合はファイル分割は"面"にして、該当するファイルのみを再入稿するとよいでしょう。

PDF 用リンク情報付きファイルの作成

EWB で作成するファイルは PostScript 形式なので、変換プログラムを用いて PDF に変換することが可能です。 PDF は文書内外へのハイパーリンク情報を持たせることができます。 EWB では目次、索引、本文中の図表番号、本文中の URL にリンク情報を持たせられるようになっています。 PDF のための情報として以下の項目を指定します。

・著作者

著作者名を指定します。

・タイトル

書名を指定します。

・サブタイトル

書名に副題がある場合に指定します。

・しおり/サムネールの表示

PDF ファイルを開いたときの表示状態を指定します。しおり / サムネールのどちらかの表示の指定、またはフルスクリーンモードでの表示が指定できます。

これで OK を押すとリンク情報の入った PS ファイルが Targets ディレクトリに PDF.PS という名前で作成されます。

なお、直接 PDF 形式のファイルが作られるわけではないのでご注意ください。 PDF 形式への変換には Adobe Acrobat 等の変換プログラムが必要です。

ページ指定 PDF 用ファイルの作成

ホームページ等で書籍のサンプルとして一部のページだけを PDF に変換するための PostScript ファイルを作成します。この場合、リンク情報は付加されません。

指定項目はリンク情報を付ける場合と同様ですが、さらに取り出すページ番号を指定します。 たとえば 1,2,5~10 ページの、次のように入力します。

1 2 5-10

番号はスペースで区切り、範囲で指定する場合は開始番号と終了番号を-でつないで指定します。 これで OK を押すとページを切り出した PS ファイルが Targets ディレクトリに pdf.ps というファイルが作成されます。

7.2.12 入稿後のページ差し替え

フィルム入稿後にミスが見つかり、ページ単位で差し替えを行うときは "ページ差し替え"ボタンで PS ファイルを作成します。

まず原稿を修正し、すべてのファイルを組み直し、目的のページが正しく組まれていることを 印刷で確認しておきます。

確認できたら差し替える原稿を選択し、"ページ差し替え"ボタンを押します。

確認のメッセージの後、差し替えるページ番号を聞いてくるのでページ番号を入力します。 たとえば 10,15,16,17,20 ページを差し替える場合、次のように入力します。

10 15-17 20

番号はスペースで区切り、範囲で指定する場合は開始番号と終了番号を-でつないで指定します。 これで OK を押すとページを切り出した PS ファイルが Targets ディレクトリに < 本体名 >.PS.cut というファイル名で作成されます。

第3部応用編



図表ファイルの作成

EWBのトリガでは、簡単な表やリストの要素を作成する機能しかありません。この機能だけでは図版を含んだページアップをするのに少々力不足です。そこで EWB では、表プロセッサなどの図表作成ツールを使って電子化された図表ファイルを取り込んで処理する仕掛けを用意しています。 さらに、EWB は組版エンジンとして TEX を利用しているため、TEX ファイルの図表類であれば原稿中に取り込むことができます。

ここでは、EWB で扱えるこれらの図表要素をうまく出力するためのノウハウを説明していきます。

1.1 EWB **で扱える図表**

EWB で扱える図表の種類を、各要素ごとに見ていきましょう。

1.1.1 図の扱い

図には、フローチャートのようなものから、イラストレータやデザイナーが作成する凝った絵までいろいろな種類があります。EWB ではこれらの図を EPS 形式のファイルにすることですべて扱うことができます。

1.1.2 表の扱い

EWB には、"//table"を使って表を作成する機能があります。この表組トリガはタブを使って表を表現するため、格子状の表しか作成できません。複雑な表を作成するのであれば、Illustrator等の画像作成ツールで作成し、EPS 画像として扱ったほうが効率が良いでしょう。

1.1.3 **リストの扱い**

プログラムリストは、EWB がもっとも得意とする要素です。コンスタントピッチの書体を使い、タブやスペースによる桁揃えを認識します。さらに、書体や字送りなどの体裁を変更したコメントを埋め込むこともできます。

1.1.4 数式の扱い

EWBシステムの組版エンジンである T_EX は、もともと数式をきれいに作成するために開発されたものです。ところが、T_EX を利用しているにもかかわらず、EWB には数式を記述する方法が、現時点では備わっていません。このため、数式を組み込むためには、数式を直接 T_EX 形式でコーディングして対処することになります。

1.2 図表ファイルと本文とのリンク

次に図表類と本文の関係を見てみましょう。図表類は大きく分けると、

- ・原稿中に埋め込まれた図表
- ・原稿とは別になっている図表ファイル

の2つになります。

本文原稿中に埋め込まれた表やリストなどは、図表というよりはむしろ本文と同様の扱いを受けます。そのため、これらの図表類はフロートすることはなく、本文原稿中で入力された位置に必ず出力されます。また、原稿中に埋め込まれた図表には、番号やキャプションを付けることはできません。もし、キャプションの付いた図表として扱いたい場合には、図表部分を本文から切り出して別の外部ファイルにする必要があります。

1.2.1 外部ファイルにした図表

EWB で扱える図表ファイルの種類は、次のようになっています。

拡張子	説明
.tbl	EWB コーディングした表
.lst	EWB コーディングしたリスト
.tex	T _E X コーディングした数式など
.eps	EPS 形式の画像

EWB は、拡張子によってファイルの種別を判断します。これらの拡張子以外を指定した場合は、すべて EWB コーディングしたファイルであると判断します。

また、これらの図表ファイルを本文中に読み込むには、物理属性ファイルを介して読み込む方法とインクルードトリガ"//inc"を使う2つの方法があります。物理属性ファイルを介して取り込む場合は、図表トリガの後に入力した文字列がキャプションとして出力され、インクルードトリガを使って本文に取り込む場合には、キャプションが付きません。EPSファイルだけは//incで読みこむことはできません。

1.2.2 EPS **ファイルの扱い**

EWB では EPS ファイルを取り込むことができます。これにより、イラスト的な図や複雑な線画も扱うことができるわけです。ただし、扱える EPS ファイルには以下のような制限があります。

- ・ASCII 形式 (テキスト形式) であること
- ・改行コードが UNIX 形式であること
- ・ヘッダ部に漢字が使用されている場合、漢字コードが EUC であること

Macintosh で作成した EPS ファイルの場合、UNIX に転送したときに改行コードが Macintosh 形式のままである場合があります。これらのコードチェックをするには epscheck というコマンドを使用します。

```
% epscheck EPSファイル名
```

ファイルが適切でない場合、次のようなエラーメッセージを出力します。

```
% epscheck test.eps
test.eps : warning:バイナリ形式になっています
test.eps : warning:改行コードがUNIX用ではありません
test.eps : warning:漢字コードが違います
%
```

改行コードが UNIX 形式になっていない場合は以下のコマンドで改行コードを変換してください。

```
% tounix ファイル名
```

ファイルがバイナリ形式であった場合は UNIX 上で ASCII 形式に直すことはできません。 Mac inctosh などで ASCII 形式で保存しなおしてください。

1.2.3 インクルードトリガで取り込む

作成した図版を本文に取り込むには、編集トリガの"//inc"を使います。本来、別ファイルになった図表を EWB の本文に中に取り込むには、ファイルの種類ごとに適切な処理を必要とします。"//inc"は、その処理を自動的に図表ファイルに対して行ってくれるため、何もする必要がありません。ただし、EPS ファイルだけは"//inc"では扱えません。

1.2.4 インクルードトリガの入力方法

図表を取り込むには、インクルードトリガの後に"ファイル名"を入力します。

//inc [図表ファイル名].[拡張子]

たとえば、"list1.lst"というリストファイルを本文に取り込むには、次のような指定を行います。

//inc list1.lst

なお、" //inc " は段落と段落の間にしか指定できません。そのため、たとえば数式を段落の途中に貼り込むのに、

...... 因数分解の//inc equ1.texという公式は 、......

のような使い方はできません。もし、このような表現をしたい場合には、組み込みトリガ" //LaTeX "を使用して、

..... 因数分解の//LaTeX{ \$(\alpha + \beta)^2 = \alpha^2 + 2\alpha \beta + \beta^2\$ //LaTeX}という公式は 、......

のように直接書き込みます。すると

因数分解の $(\alpha+\beta)^2=\alpha^2+2\alpha\beta+\beta^2$ という公式は 、

という出力が得られます。

EPS ファイルは段落の間でも途中でも"//inc"トリガでは扱うことはできません。EPS ファイルを直接取り込むには、"//ZUHAN"トリガまたは"//ICON"トリガを使用して、

```
//ICON{Omm,Omm,50,icon.eps//}
.....文中の場合

//ZUHAN{100mm,Omm,0,c,zuhan.eps//}
......行間の場合
```

のように書きます。//ZUHAN, //ICON トリガについては Appendix B 標準編集トリガー覧表 (337ページ) を参照してください。



標準機能の拡張

EWB の標準機能は、本を制作していく上で、必要最低限の要素を表現するものばかりです。実際に EWB で本を作っていくと、標準機能だけでは、どうしても表現しきれない部分が出てきてしまいます。このような時のために、EWB には拡張を行う機能が備わっています。

ここでは、EWB に少し手を加えることによって、EWB の表現力を広げる方法について解説していきましょう。

2.1 EWB **の表現力を広げる**

書店に並べられている書籍などをよく眺めてみると、特殊な文字や記号が数多く使われ、網かけや特殊な罫線などをうまく利用しています。もちろん、これらは、出版業界ではあたりまえのように使われているものばかりです。

しかし、EWB は UNIX コンピュータ上で動作するソフトウェアであり、利用しているキーボードやプリンタは、通常のコンピュータ規格に沿った標準的なものです。そのため、一般の書籍に用いられている特殊な文字や記号などは、キーボードを使って表現できません。

そこで、EWB では、本を制作する上で必要な罫線や特殊記号などが取り扱える拡張機能を備えています。EWB が実際に組版を行う場合に、いくつかのファイルを参照するようになっています。参照されるファイルには、組版時に必要な規則や特殊記号が記述されており、このファイルを修正することで拡張できるわけです。

2.2 約物修正の一括処理

入力された文字の中には、基本の組版処理では不格好になってしまうことがあります。たとえば、行頭に位置する開き括弧は、そのまま EWB で組版処理させると、インデントと約物のアキが重なって、大きな空白ができてしまいます。これを避けるには、行頭の括弧をファイルの中から探し出し、特殊なトリガを入れていかなければなりません。このような作業を効率的に行うために、EWB のスクリプト処理と呼ばれる機能が用意されていて、EWB で組版を行う前に、"sed"というツールを用いて処理をさせます。約物のアキ調整を行うには、"/usr/local/ewb/3.3"にある"yakumono.sed"というスクリプトファイルを利用します。このファイルを、作業ディレクトリにコピーすると、EWB-Shelf が自動的にスクリプト処理を行うような仕掛けになっています。

2.3 トリガの表現力を広げる

これまでに見てきた各種トリガはどれも、すでに EWB に備わっている機能でした。しかし、備わっている機能だけでは、EWB で表現できるレイアウトに制限が出てきてしまい、実際に出版されているような多種多様なものには、対応することができません。本によっては、見出しに罫飾りをしたいとか、小組を罫線で囲みたいなどといったようなトリガ単位で EWB の表現力を広げなければならない場合があります。 EWB では、このような拡張を行うには、組版エンジンである LATEX 2_ε に手を加えて、必要な機能を実現します。

2.3.1 組版に必要なファイル

まずは、EWB が組版を行う時に参照するファイルを明らかにしておきましょう。

2.3.2 参照ファイルの内容

EWB が実際に組版を行う場合には、いくつかのファイルを参照します。そのうち 2 つは、利用できるトリガにはどういったものがあるのかということを記述した"トリガ定義ファイル"です。あとは、組版を担当する LATEX 2_ε が参照するマクロがたくさん収められている"スタイルファイル"と呼ばれているものです。これらのファイルは作業ディレクトリか、その下の"Styles"ディレクトリに置いておく必要があります。

トリガ定義ファイル"triger.def"

LATEX 2_ε のマクロと原稿中に指定した編集トリガとの対応を設定しているのが、"トリガ定義ファイル"です。このファイルには、マクロとトリガが"一対一"で対応しています。そして、利用できるトリガを実際に記述したファイルが"triger.def"と呼ばれ、原稿ファイルを LATEX 2_ε 形式のファイルに変換する際に参照します。なお、トリガ定義ファイルに記述してあるトリガは編集トリガだけであり、組版トリガの定義はありません。

このファイルは pssted により生成されるため、トリガ定義をエディタで追加した場合、新たに体裁ファイルから pssted でスタイルファイルを作成すると修正部分が失われてしまいますので注意してください。

特殊トリガ定義ファイル" triger.diff"

"triger.def"の内容に追加する形で使用されるトリガ定義ファイルです。"triger.def"と同様に、マクロと編集トリガの対応を記述します。トリガ定義を追加する場合に使用します。EWBの標準機能だけを利用する時には必要ありません。

スタイルファイル "ewb2latex.cls"

それぞれの本の体裁に合わせた T_EX マクロが収められているファイルです。pssted で入力した 体裁情報が記述されています。このファイルを直接修正した場合、新たに体裁ファイルから pssted でスタイルファイルを作成すると修正部分が失われてしまいますので注意してください。

ベーススタイルファイル "ewbbase3_3.sty"

EWB の基本的なマクロが収められているファイルです。このファイルは組版の中枢部分に相当し、EWB の管理者がマクロの不具合修正や仕様変更の際に触れるものです。このファイルは、"/usr/local/ewb/3.3/ewbmacros/lib"に置いてあり、カレントディレクトリにこのベーススタイルが存在しなくても、スタイルファイルが自動的に共有ディレクトリから探して読み込むようになっています。

特殊スタイルファイル "diff.sty"

"ewb2latex.cls"の内容に追加する形で使用されるスタイルファイルです。体裁マクロを追加する場合に使用します。EWBの標準機能だけを利用する時には必要ありません。

索引スタイルファイル "index.cls"

索引の体裁に合わせた TEX マクロが収められているファイルです。psidxsted で入力した体裁情報が記述されています。このファイルを直接修正した場合、新たに体裁ファイルから psidxsted でスタイルファイルを作成すると修正部分が失われてしまいますので注意してください。

特殊索引スタイルファイル "indiff.sty"

"index.cls"の内容に追加する形で使用される索引スタイルファイルです。索引体裁マクロを追加する場合に使用します。EWBの標準機能だけを利用する時には必要ありません。

索引変換情報ファイル"indexsty"

抽出された索引を並び替えて T_EX 形式に変換する際に、どのような T_EX ファイルへ変換するかという情報を収めたファイルです。psidxsted が生成します。

2.3.3 参照ファイルの相互関係

囲み罫や特殊記号などといった特別の機能を持ったマクロが収められているファイルです。トリガ拡張を行う場合に作成するファイルで、EWBの標準機能だけを利用する時には必要ありません。

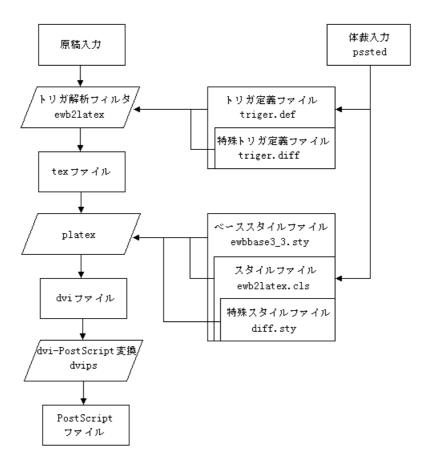


図 2-1 トリガ定義ファイルとスタイルファイルの関係

EWB が組版時に参照するファイルは、図 2-1のような関係になっています。EWB で組版を行うと、まず最初に"triger.def"ファイルを参照して、トリガを含んだ原稿ファイルを LaTeX 2ε の形式に変換します。このとき、"triger.diff"があればそれも参照します。次に、platex が起動され、"ewb2latex.cls"が自動的に読み込まれます。さらに、"ewbbase3_3.sty"と"diff.sty"が順次、読み込まれるようになっています。なお、拡張専用の"diff.sty"というファイルが存在しない場合には、読み込みは行いません。

2.3.4 トリガ拡張の作業手順

EWB のトリガ拡張を行う場合には、拡張の内容によって作業手順が多少異なってきます。

トリガの表現力を拡張する場合には、「トリガを追加するだけで作業が済むのか」、「マクロまで作成しなければならないのか」などといったことを十分に考察してから拡張を行なわなければなりません。トリガの単純な追加で済むような内容であれば、sted で作成されるトリガ定義ファイル"triger.def"を修正するだけです。また、特殊な機能を実現するためにマクロを作成しなければならない場合には、次の2つのステップを踏まなければなりません。

- ・" diff.sty"をエディタで作成し、機能を実現するために必要なマクロを記述する
- ・作成したマクロとトリガを対応させるために、pssted で自動的に生成された "triger.def"を修正する

トリガを拡張する場合には、他の"ewbbase3_3.sty"や"ewb2latex.cls"といったスタイルファイルを修正する必要はありません。

2.4 トリガ定義の拡張

ここでは、実際に"triger.def"にトリガを定義する方法を解説しましょう。

2.4.1 トリガ定義ファイルの文法

"triger.def"の基本的な書式は、"トリガの種類のキーワード"、"トリガ名"、"トリガに対応したマクロ"、"その他の情報"の各項目をタブやスペースで区切り、1つのトリガに対して1行ずつ記述します。各項目が空白を含む時は、ダブルクォーテーションで囲うようにしてください。

トリガ定義ファイルでは、編集トリガの種類を識別するためのキーワードには、次のようなものがあります。

文字置換 defstr

単純トリガ defsimple 強調トリガ defgothic グループトリガ defgroup リストトリガ deflist 表組トリガ defTable 見出しトリガ defcaption 図表トリガ deffigure 箇条書トリガ defitem 脚注トリガ deffootnote 非展開トリガ defnoexpand ラベルトリガ deflabel 参照トリガ defref URL トリガ defurl その他 defescape deftab defchapter

2.4.2 文字置換

" defstr"は、1つの文字列を別の1つの文字列に置換します。

defstr 置換する文字列 置換先文字列

2.4.3 単純トリガ

単純トリガは単純に文字を置き換えるトリガです。これを定義するには、"defsimple"を使います。

defsimple トリガ名 置換文字列

置換文字列

ここに入力する文字列は、マクロ名でもかまいません。ただし、バックスラッシュの記号は エスケープされてしまうため、2 つ続けて " \\ " と記述してください。

2.4.4 強調トリガ

書体の切り換えを行う強調トリガ"//g{~//g}"は、"defgothic"を使って定義します。

defgothic トリガ名 始まりのマクロ 終りのマクロ

引数を取るマクロを作成した場合は、ここに定義します。たとえば、引数の文字列に対して罫囲いする"\boxit"というマクロを作成し、"//kakomi{文字列//kakomi}"のようなトリガ入力を行いたい場合には、

defgothic kakomi "\\boxit{" "}"

と記述します。

ルビ、傍点もこのトリガにより定義されています。

2.4.5 **グループトリガ**

小組トリガ "//c1"は "defgroup"を使って定義します。

defgroup トリガ名 マクロ名 改行挿入の有無(Y/N)[脚注の扱い(Y/N)]

マクロ名

環境型に定義したマクロの定義名を記述します。

改行挿入の有無

改行挿入する "Y"か、しない "N"かを指定します。

脚注の扱い

このグループトリガで囲まれた範囲に書かれた脚注トリガの処理方法を指定します。通常の方法で作成した小組の中では脚注トリガが自由に使えますが、拡張した罫巻した小組では、脚注が出力できない場合があります。このような場合に、この項目を"Y"にすると出力できるようになります。この"脚注の扱い"は省略可能です。

実際に入力すると、次のようになります。

defgroup c1 kogumi YN

2.4.6 プログラムトリガ

プログラムトリガ " //list " は " deflist " を使って定義します。

deflist トリガ名 マクロ名 タブストップ 字詰め オプション オプション:=行番号自動発生の有無(Y/N) バックスラッシュを¥マークにする(Y/N) 改行挿入の有無(Y/N)

マクロ名

環境型に定義したマクロの定義名を記述します。

タブストップ

このトリガでタブを何個のスペースに展開するかを指定します。

字詰め

EWB の旧バージョンで使われたもので現在は使用しません。

オプション

行番号自動発生の有無、バックスラッシュを円マークにするかどうか、そして改行挿入の有無を"Y"か"N"を指定します。

実際に入力すると、次のようになります。

deflist list programi 2 74.00 NNY

2.4.7 表組トリガ

表組のトリガ "//table "は "defTable"を使って定義します。

defTable トリガ名 マクロ名 タブストップ 項目間隔 項目内行送り 改行挿入の有無(Y/N)

項目間隔

1つの行の項目間のアキを歯単位で指定します。

項目内行送り

1つの項目が複数行になった場合の行送りです。歯を単位にして指定します。

実際に入力すると、次のようになります。

defTable table hyoui 2 10.00 20.00 N

2.4.8 見出しトリガ

見出しトリガ " //i " は、" defcaption " を使って定義します。

defcaption トリガ名 マクロ名

実際に入力すると次のようになります。

defcaption i chapter defcaption ii section

2.4.9 図表トリガ

図表トリガ "//f"は "defffigure"を使って定義します。

deffigure トリガ名 マクロ名 図とキャプションの空き

図とキャプションの空き

歯単位で指定します。ここで指定した値が通常のアキになります。もし、個々に変更する場合は、物理属性ファイルで設定します。

deffigure f figure 20.00

2.4.10 箇条書トリガ

箇条書のトリガは " //k1 " は " defitem " を使って定義します。

defitem トリガ名 マクロ名 改行挿入の有無(Y/N)

たとえば、箇条書環境 "kajyoix"を作成し、それを"/k9{~/k9}"で利用したければ、

defitem k9 kajyoix Y

と記述します。

2.4.11 脚注トリガ

脚注トリガ "//ky"は "deffootnote"を使って定義します。

deffootnote トリガ名 マクロ名

実際に入力すると、次のようになります。

deffootnote ky \\footnote

2.4.12 非展開トリガ

TEX にファイル名や数値を渡す場合に使用します。 "defnoexpand" を使って定義します。

defnoexpand トリガ名 始まりのマクロ 終りのマクロ

ここでは EPS ファイルを読み込む "//ZUHAN"の例を示します。

defnoexpand ZUHAN "\\ZUHAN " "\\end{}"

2.4.13 ラベルトリガ

見出しの参照元を指定するラベルトリガ "//label "は "deflabel "を使って定義します。

deflabel トリガ名 始まりのマクロ 終りのマクロ

実際に入力すると、次のようになります。

deflabel label "\\label[" "]"

2.4.14参照トリガ

見出しを参照するトリガ " //numref (番号の参照) "、 " //strref (文字列の参照) "、 " //pageref (ページ番号の参照) " は " defref " を使って定義します。

defref トリガ名 始まりのマクロ 終りのマクロ

実際に入力すると、次のようになります。

```
defref numref "\\numref[" "]"
defref strref "\\strref[" "]"
defref pageref "\\pageref[" "]"
```

2.4.15 URL **トリガ**

URL を記述するためのトリガ " //url { ~ //url } " は " defurl " を使って定義します。

defurl トリガ名 始まりのマクロ 第1引数と第2引数の区切り 終りのマクロ

実際に入力すると、次のようになります。

```
defurl url "\\ewbhyperurl{" "}{" "}"
```

2.4.16 その他

その他に、" defescape "、" deftab "、" defchapter " といった指定項目があります。

defescape

" defescape " は、トリガ開始文字列 " // " を定義します。他の文字列に置き換えることができますが、完全にサポートされていないので " // " で使用するようにしてください。

deftab

" deftab "は、表組とリスト以外の本文中に現れたタブ文字をいくつの空白に置き換えるかを指定します。

defchapter

" defchapter "は、見出し番号設定のための文字列を定義します。

defchapter chapter

このように記述した場合、原稿ファイルの先頭に "chapter 5 "と入力すると、その原稿の章番号は"5章"からとなります。

2.5 マクロの拡張

ここでは、トリガの定義だけでは実現できないような機能を追加する場合に行うマクロ拡張に触れます。マクロを定義する際の基本的な書き方や EWB の組版の核となる "ewbbase3_3.sty"の解析、さらに実例を使った解説を行います。

2.5.1 マクロ定義ファイルの文法

T_EX では、組版の命令や組版時に使う変数を意味する文字列の先頭には、バックスラッシュ記号 "\"を付けて、他の文書と区別できるようになっています。 たとえば、"\inhibitglue "や"\parindent"、"\advance"といった文字列が T_EX の命令です。

一般的に、TEX が識別できる命令の文字列には英文字しか使えないため、バックスラッシュからはじまり、数字や記号類、空白などが書かれている位置までの文字列が命令として見なされ、解釈 / 処理されます。

マクロ定義の仕方

マクロを定義する場合には "\def"という命令を使用し、次の形式で記述します。

```
\def\マクロ名{
.......
}
```

これで、"マクロ名"という名前のマクロが定義できます。"マクロ名"の後ろに書かれている 開き括弧と閉じ括弧の中には、そのマクロをどのように動作させるかを記述します。単純に文字 列を出力したいのであれば、その文字列をそのまま記述するだけで定義ができ、罫線を引いたり したい場合には、その動作を行う命令を書けばマクロとして使用できます。

たとえば、「東京都渋谷区」という文字列を、マクロを使って入力してみましょう。"\address"という名前のマクロを次のように定義します。

```
\def\address{
東京都渋谷区
}
```

TFX の原稿中で、

私の会社は、\address にあります。

と入力すると、

私の会社は、東京都渋谷区にあります。

という出力が得られます。

引数と区切り文字

マクロには引数を持たせることができ、マクロを利用する時にパラメータとして受け渡すことができます。引数は、マクロを使って定義する際に、マクロ名と開き括弧の間に"#数字"を指定することで表します。

```
\def\マクロ名#1#2...#9{
.......
}
```

マクロ名の後ろに記述した"#1,#2…#9"が、マクロへの引数の1番目、2番目~9番目に対応し、最大で10個の引数を取ることができます。たとえば、先ほどのマクロを「東京都渋谷区」の後ろに、引数を取るようなマクロに変更してみます。

```
\def\address#1{
東京都渋谷区#1
}
```

そして、次のような入力を行うと

\address{ 代々木 } は 東京都渋谷区代々木 \address{ 初台 } は 東京都渋谷区初台

と展開されます。

また、引数の間に文字や記号を加えてマクロを定義をすると、その文字や記号類は入力するパラメータを区切る文字として動作します。たとえば、次のようなマクロを定義することもできます。

```
\def\position(#1,#2){
中点のX座標は#1で、Y座標は#2となる
```

}

このように定義した場合、このマクロは必ず

...したがって、\position(250,310)。そのため、...

というように、カンマで区切って引数を渡さなければ、正しく動作しませんので注意が必要です。 これを正しく入力すれば、次のように出力されます。

...したがって、中点の X 座標は 250 で、Y 座標は 310 となる。そのため、...

一般的なマクロを定義する際には、このようなカンマ区切りではあまり指定しませんが、ちょっとした工夫で、マクロへの引数を 10 個以上渡せるようにできるため、便利な機能といえます。標準のルビトリガも引数を 2 個持つため、この方法で定義されています。

2.5.2 ewbbase3_3.sty ファイルの詳細

"ewb2latex.cls"の中では、書体や見出し、小組、箇条書などといった各要素の体裁指定を行いますが、この中で使用しているマクロは、"ewbbase3_3.sty"ファイルの中で定義されています。この"ewbbase3_3.sty"には EWB が使用する基本的な体裁のマクロがすべて定義されているため、特別に"ベーススタイルファイル"と呼んで、"ewb2latex.cls"や"diff.sty"といったマクロファイルと区別しています。

EWB の表現力を拡張するためには、このベーススタイルファイル "ewbbase3_3.sty"の中にどのようなマクロが定義されており、どのように利用したらよいのかということを理解しておかなければなりません。

そこで、このベーススタイルファイルの詳細を眺めてみましょう。ベースファイル全体は、大きく分けて、次のように構成されています。

- ・変数
- ・書体と組版の設定
- ・小組
- ・箇条書
- ・見出し
- ・版面と柱 / ノンブル
- ・脚注
- ・図表の参照とフロート / 非フロート
- ・キートップ

- ・囲み罫
- ・ルビ
- ・傍点

变数

ベーススタイルファイルでは、まず EWB 専用のマクロを作成する場合に使用する単位を定義しています。定義されている変数は次の2つです。

- ・行送りを表す単位
- ・字送りを表す単位

この行送りと字送りの単位は、さまざまな環境で利用できるように、個別に定義されています。

行送り

行送りは、"\vokuri"として参照できます。これは、"vertical (垂直の) okuri"の意味を表しています。この変数には、使用した時点の環境の行送りが設定されています。たとえば、このマクロを小組環境の中で使えば小組の行送りが、箇条書環境の中では箇条書の行送りが設定されます。このため、"2行アキ"という指定を行う場合には、統一して"2\vokuri"という表現が可能になります。

また、本文の行送りは、先頭文字を大文字にした"\Vokuri"を用いて参照することができ、この変数を使えばどの環境で利用しても、必ず本文の行送りとなります。

字送り

字送りは、"\hokuri (horizontal okuri の意味を表す)"を用いて参照できます。行送りと同様に、それぞれの環境で参照でき、基本となる本文の字送りは、先頭文字を大文字にした"\Hokuri"という変数で参照できるようになっています。

また、字詰めの指定を行う場合には、T_EX の全角文字の大きさを表す"zw"を使わずに、この "\hokuri"で表してください。"zw"では、"字送り1歯ツメ"などといった指定を表すことができないためです。

組版

ベースファイル中の組版という名称の部分には、使う書体や字送り、行送りを設定するためのマクロや設定したそれらの要素を利用するためのマクロが定義されています。このマクロは、和文/欧文/数式フォントなどの関連した書体に名前を付けてひとまとめにして定義するものです。 EWB では、各フォントを組みにしたものを"コンポジットフォント (composit font)"と呼んで、

一括で管理できるようにしています。

コンポジットフォントを取り扱う命令としては、次のようなマクロが用意されています。

- ・フォントの定義マクロ
- ・フォントのコピーマクロ
- ・書体切り換えマクロ
- ・強調書体の定義マクロ

フォントの定義マクロ

コンポジットフォントを定義するには、次のマクロを使います。

\DeclareCompositFont{ 名前 }[文字色]{ 和文級数 }[字送り]{ 和文フォント } [欧文級数]{ 欧文フォント }[オフセット]

文字色

文字の色を CMYK 形式で指定します。シアン、マゼンタ、イエロー、ブラックの順で 0.0 ~ 1.0 の値を " , " で区切って記述します。

和文級数/欧文級数

利用する書体の和文や欧文の級数を、Q(級数)を単位として指定します。

字送り

指定した文字の字送りを、H(歯)を単位として指定します。この字送りは省略することもできます。省略した場合には、和文級数"の値となります。

オフセット

和欧文間のベースライン位置を調整します。省略した場合には、" Opt " となります。また、書体によって、このオフセットの値は違います。

和文フォント/欧文フォント

利用する書体名を記述します。ただし、書体名は LATEX 2_ε が理解できる NFSS2 形式で指定しなければなりません。NFSS2 形式とは、LATEX 2_ε で採用された書体表現の形式で、エンコード/ファミリ/シリーズ/シェイプで表します。たとえば"新ゴ B"を指定する場合には、

......{13Q}[13H]{新ゴB}[13Q......

ではなく、

.....{13Q}[13H]{JY1/sgo/b/n}[13Q......

と記述しなければなりません。LATEX 2_{ε} では、"新ゴB"を"JY1/sgo/b/n"という名前で指定します。EWB で使用できる書体名は「Appendix D 書体一覧表」に記載してありますので、参

照してください。

名前

環境名を指定します。環境名とは、どんな種類の要素であるかということを示した名前のことです。一般に、部は"part"、章見出しは"chapter"などといった環境名を付けています。 pssted では、"見出し番号指定文字列"で指定されたものが使われ、指定されなければ"sectioni" sectionii"といった名前が使われます。

たとえば、次のような書体を新たに定義する場合には、

環境名 chapter 和文書体 見出しミン 欧文書体 Times-Roman 和文級数 18級 欧文級数 18級 字送り 17 歯(1歯ツメ) 文字色 シアン 0.0 マゼンタ 0.0 イエロー 0.0 ブラック 1.0

" \DeclareCompositFont " マクロを使って、次のように書きます。

これで、和文や欧文の書体セットを、"chapter"という環境名で参照できるようになります。 小組や箇条書の環境名は、自由な名前を付けることができます。ただし、名前の文字列に数字 や記号類は含ませることはできません。また、すでに使われている環境名を使って新たに書体を 定義することはできないので注意が必要です。

Column:変形書体について

EWB の和文書体には、かなつめ、長体、平体、斜体といった変形書体が用意されています。これらの変形は"\DeclareCompositFont"マクロ中の書体名で指定します。

変形を使わない場合、書体名は "JY1/rmn/l/n" のように、最後が "n" になっています。この部分を以下のように記述することで変形の書体として指定できます。

- ba 長体 1 (字送り方向に 10%縮小)
- ca 長体 2 (字送り方向に 20%縮小)
- da 長体3(字送り方向に30%縮小)
- ea 長体 4 (字送り方向に 40%縮小)
- ga 平体 1 (行送り方向に 10%縮小)
- ha 平体 2 (行送り方向に 20%縮小)
- ia 平体 3 (行送り方向に 30%縮小)
- ja 平体 4 (行送り方向に 40%縮小)
- ab 斜体1(字送り方向に10度傾斜)
- ac 斜体 2 (字送り方向に 20 度傾斜)
- ad 斜体3(字送り方向に30度傾斜)
- ae 斜体-1 (字送り方向に-10 度傾斜)
- af 斜体-2(字送り方向に-20度傾斜)
- ag 斜体-3(字送り方向に-30度傾斜)
- c かなつめ(かなの文字間を詰める。文字自体は変形しない)

新ゴBの平体1を指定する場合、書体名は"JY1/sgo/b/ga"のように記述します。

長体を使用する場合、字送りを指定しないと文字級数と同じ字送りになり、文字間が不自然に空いてしまいます。長体を掛けた分だけ歯数を詰めて字送りを指定しておいてください。

フォントのコピーマクロ

新しく定義するのではなく、すでに定義した書体を別の名前で利用するためのマクロです。新しくコンポジットフォントを定義しないで、既にある環境名を別の環境名で参照できるように、環境を一括でコピーすることができるようになっています。コピーを行うには、次のマクロを使います。

\copycomposit{ コピー先の名前 }{ コピー元の名前 }

たとえば、書体の定義として"chapter"という環境名がすでに定義され、その環境を"section" としても利用できるようにするには、

\copycomposit{section}{chapter}

と記述します。

書体切り換えマクロ

" \DeclareCompositFont " や " \copycomposit " で定義した書体を実際に使う場合には、

\selectcomposit{ 名前 }

というマクロを使います。

たとえば、先ほど定義した "chapter"の書体を使う場合には、

...ということで、新しく{\selectcomposit{chapter}定義した書体が}使われます。

というようになります。"名前"には、書体定義の際に付けた環境名を記述します。

強調書体の定義マクロ

TeX で書体を変更する場合には、"\rm "、"\tt "、"\sl "、"\it " などのコントロールシーケンスを使います。 EWB のコンポジットフォントも同様に、コンポジットフォントの一部の書体を変更することができます。 たとえば、"\tt " を指定した場合に、その切り換えた時点の環境に設定されているコンポジットフォントの書体を使うようになっています。

切り換える書体を定義するには、コンポジットフォントの定義と同じマクロを使います。書体切り換えの命令を表す文字列は、環境名のプレフィックスと同時に定義しなければなりません。たとえば、"\tt"という書体切り換えを行いたいのであれば、まず特定の箇所の書体を変えるための命令を、次のマクロを使って定義します。

\newemph{ 名前 }

"\tt"で切り換えたけば、

\newemph{tt}

とします。次に、その "\tt"用のコンポジットフォントを

と定義します。すなわち、書体切り替えは

強調名 @ コンポジット名

という環境名にして書体を定義しておくのです。

字詰め / 行送り

EWB では、字詰めや行送りも書体と同様に、ひとまとめにして"スリップ(slip)"と呼ばれる名前を付けて管理するようになっています。スリップには、

- ・スリップの定義マクロ
- ・スリップのコピーマクロ
- ・スリップ切り換えマクロ

というマクロが定義されています。

スリップの定義マクロ

新しくスリップを定義するには、次のマクロを使います。

```
\newslip{ 名前 }{ 行送り }% 
{ 段落インデント }{ 左マージン }{ 右マージン }
```

ここで設定する "段落インデント"や"左/右マージン"などは、図 2-2に示されている箇所の値を意味しています。

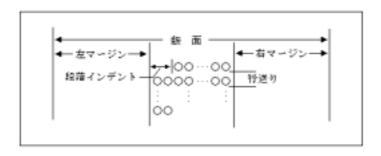


図 2-2 スリップのパラメータ

左マージン / 右マージン

" 左マージン" および" 右マージン"は、それぞれ字下げ/字上げのことです。

たとえば、

環境名	honmo
行送り	24 歯
段落インデント	13 歯
左マージン	13 歯
右マージン	0 歯

というスリップを定義すると、

 $\newslip{honmon}{24H}{13H}{13H}{0H}$

となります。

また、左 / 右マージンにはマイナスの値も指定することができます。もし、右マージンにマイナスの値を設定すると、版面から文字が飛び出すことになります。

行送り/段落インデント/左右マージンの各パラメータの後に"[R]"を付加すると、直前のスリップのパラメータを基準にして相対的に各パラメータが設定されるようになります。 たとえば、

というスリップは、直前の環境よりも13歯だけ左マージンを増やすということになります。

スリップのコピーマクロ

新しく定義するのではなく、既に定義したスリップを別の名前にコピーするためのマクロです。

```
\copyslip{ コピー先の名前 }{ コピー元の名前 }
```

記述の方法は、コンポジットフォントのコピーマクロと同じです。

スリップの切り換えマクロ

定義した送りを実際に使う場合には、

```
\selectslip{ 名前 }
```

というマクロを使います。

たとえば、"honmon"という名前で定義したスリップで"\vbox"を作成する場合には、

```
\vbox{\selectslip{honmon}
.....
}
```

とします。これで、行送りや左右マージン、段落インデントが設定通りに変更されます。なお、このスリップは、"\vbox "の先頭に指定するようにしてください。"\hbox "の中や "\vbox "の途中では、変更されません。

小組

EWB のマクロでは、同じ 名前 を持つコンポジットフォントとスリップを呼び出し、その前後に アキ を入れることで、小組を実現しています。そのため、コンポジットフォントとスリップを別々に定義し、"\mkkogumi"マクロを使って前後のアキを指定し、小組を作成するようになっています。

```
\mkkogumi{ 名前 }{ アキ }
```

このマクロを使って環境名を定義すると、LATEX 2_{ε} の "\begin{ 環境名 } ~ \end{ 環境名 } " 環境と同じ形式で利用できます。

```
\begin{ 名前 }
    :
\end{ 名前 }
```

このように利用すると、"\begin~\end"で囲った部分と本文との間に、指定した アキ がはいります。

このマクロでは、指定した環境の前後にアキを入れるだけですが、この環境の文字組みは、囲った環境名として定義された書体が使われる規則になっています。つまり、

```
\mkkogumi{kogumi}{...
```

と定義した場合に、

```
\begin{kogumi}
   :
\end{kogumi}
```

と記述すると、囲った部分は"kogumi"という環境名のコンポジットフォントとスリップが利用されるようになっています。もし、コンポジットフォントの定義用マクロなどを定義しておかないと、

```
! No such composit 'kogumi'.
```

といったエラーメッセージが表示されてしまいます。

新しく小組を定義する場合には、通常は "\mkkogumi "、"\DeclareCompositFont "、"\newslip "の3 つを定義します。たとえば、

環境名 kogumi 和文書体 見出しミン 欧文書体 Times-Roman

```
和文級数
         18級
欧文級数
         18級
字送り
         17歯(1歯ツメ)
行送り
         24 歯
字上げ
         1 文字
字下げ
         1 文字
上下アキ
         24 歯
文字色
 シアン
         0.0
 マゼンタ
         0.0
 イエロー
         0.0
 ブラック
         1.0
```

という小組を定義するには、次のように記述します。

```
\mkkogumi{kogumi}{24H}
```

\newslip{kogumi}{24H}{17H}{17H}{17H}

書体やスリップを定義しないで使用することもできます。

```
\mkkogumi*{kogumi}{......
```

のように"\mkkogumi "の後に" * "を付加すると、書体やスリップの定義が無い場合は"\begin{kogumi}"の直前の書体やスリップを使用します。

箇条書

箇条書用マクロ

箇条書は、小組と同様に、同じ名前を持つコンポジットフォントとスリップを呼び出し、その 前後に アキ を入れることで実現します。そして、箇条書の場合にはさらに、

- ・"\item"で、行頭を問答量だけ飛び出す事ができる
- ・"\item[見出し]"で、箇条書内の見出しを作ることができる
- ・見出し部分の書体を統一的に変更することができる

といった機能が付加されています(図2-3)。

この付加機能は、次のマクロで定義できるようベースファイルで設定されています。

\mkkajyou{ 名前 }{ 問答 }[項目間アキ]{ アキ }

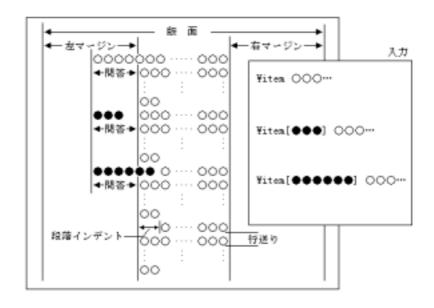


図 2-3 箇条書のパラメータ

アキ

箇条書環境の前後にどれくらいのアキを入れるかを指定します。

問答

"\item"命令を使った際の問答量を指定します。

項目間アキ

カレントのスリップに指定されている行送りに、さらにどれくらいのアキを追加するのかを 指定します。"\item"命令を使わない場合には、項目間のアキは影響しません。

箇条書環境は、ちょうど小組環境と同じ形式で、次のように記述できます。

\begin{箇条書の環境名}

\end{箇条書の環境名}

使われる書体や行送りも、やはり小組同様、指定された環境名に定義されているものが用いられます。ただし、"\item[]"マクロを使って囲った部分については、"item@ 名前 "という名称で定義されたコンポジットフォントが使われます。たとえば、

環境名 kajyoi 和文書体 見出しミン 欧文書体 Times-Roman 和文級数 18 級

18級

欧文級数

```
字送り
          17 歯 (1 歯ツメ)
行送り
          24 歯
字上げ
          1 文字
字下げ
          1 文字
問答量
          3 文字
項目間アキ
          2 歯
上下アキ
          24 歯
文字色
 シアン
          0.0
 マゼンタ
          0.0
 イエロー
          0.0
 ブラック
          1.0
```

という箇条書を定義するには、次のように記述します。

```
\mkkajyou{kajyoi}{51H}[2H]{24H}
\DeclareCompositFont{kajyo}[0.0,0.0,0.0,1.0]{18Q}[17H]{JY1/mmc/m/n}[18Q]{0T1/ptm/m/n}[0.03zh]
\DeclareCompositFont{item@kajyo}[0.0,0.0,0.0,1.0]{18Q}[17H]{JY1/mmc/m/n}[18Q]{0T1/ptm/m/n}[0.03zh]
\newslip{kajyo}{24H}{17H}{17H}{17H}
```

\mkkakyou も \mkkogumi と同様に、

```
\mkkajyou*{kajyoi}{......
```

のように記述すると、書体やスリップが定義されていない場合には直前の書体やスリップが使用 されます。

問答

問答量を部分的に変更するマクロも定義されています。しかし、このマクロは、小組や箇条書マクロの代りとして使われることはありません。

このマクロは、次の形式で用います。

```
\begin{mondo}{ 問答 }
:
\end{mondo}
```

問答のマクロは、図 2-4に示されているような利用形態となります。

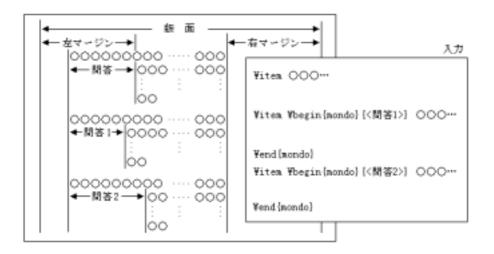


図 2-4 問答のパラメータ

見出し

見出しは、"\chapter"や"\section"などといったマクロを使って T_EX の原稿ファイルに記述します。さらに、このマクロの定義を行うために、

- ・見出し文字列の定義マクロ
- ・見出し行取りや揃えの定義マクロ

が用意してあります。

見出し文字列の定義マクロ

見出しの文字列を定義するには、次のマクロを使います。

\newsection{ 名前 }[表記]{ 文字列 }[上位構造]

表記

番号の表記を指定します。ここに指定できる変数は、次に示すように LATEX 2_ε でページ番号の表記に用いるものと同じです。

	小文字	大文字
算用数字	\arabic	
ローマ数字	\rdown roman	\Roman
アルファベット	\alph	\Alph

たとえば、見出しの番号をローマ数字にして印字したければ、

 $\verb|\newsection{section}| [\norman] {...}$

と指定します。何も指定しなかった場合には、算用数字 "\arabic "となります。

文字列

ナンバリングの方法を指定します。固定文字列や変数を使って出力方法を表します。また、 この文字列は見出しの内容自身を示すものではありません。

この "\newsection" の 文字列 には、次のマクロを使ってナンバリング方法を記述します。

\the 名前

下位構造やキャプション文字列の中で、カウンタの値を 表記 で表したものに置換されます。 \THE 名前

同じく、カウンタの値を 文字列 で表したものに置換されます。

カウンタの値というのは、対象としている見出しに付けられる番号で、章番号や節番号のことを表します。また、 上位構造 という引数には、カウンタの値をリセットするタイミングの見出しレベルを指定します。この指定の見出しレベルに達すると、新たに 1 から番号付けをはじめます。ここで、実際に小見出しのレベルを定義をしてみましょう。たとえば、節見出しでカウンタをリセットし、カウンタの前にはダイヤ記号を付けた見出しを定義するには、次のように記述します。

\newsection{subsection}[\roman]{ \thesubsection}[section]

見出し行取りと起こしの定義

本文に対して見出しの行取りなどを行うには、次のマクロを使います。

\headsection{ 名前 }[起こし]{ 前空き }% [問答]{ 本体高さ }[付き方]{ 後空き }[続ける本文行数]

起こし

見出しのページ体裁を指定します。次に示すパラメータを組み合わせて指定することができます。

(起こし) := {t,p,P}[l,r]

t: 改ページ起こし tl: 左ページ起こし tr: 右ページ起こし

p: 扉 pl: 扉左起こし pr: 扉右起こし

P: 扉(裏白) PI: 扉(裏白) 左起こし Pr: 扉(裏白) 右起こし

(省略):成行き

"t"を指定すると、新たに改ページを行って見出しを出力し、ノンブルや柱も印字されます。 "p"を指定すると、改ページを行って見出しを出力し、そのページにはノンブルや柱は印字 されません。"P"を指定すると改ページして見出しのみを出力し、さらに次のページを空け てから本文が始まります。"I"は左ページになるように、"r"は右ページに見出しが出力され るように改ページ処理されます。また、 起こし に何も指定しないとなりゆきになります。 たとえば、左ページ起こしの設定を行いたい場合には、

...}[t1]{...

と指定します。

前空き / 後空き

見出し本体の前後に何行のアキを入れるかを指定します。 前空き や 後空き は、本体高さ に追加されるアキです。 前空き は、ページの切れ目では省略されてしまいます。さらに、 起こし に何も指定しなかった場合にも、 前空き は改ページされた位置で省かれます。もし、改ページ位置に 前空き を残したい場合には、 本体高さ に 前空き のアキを追加した値に設定するようにしてください。

本体高さ

見出しとして何行取りにするのかを指定します。ただし、行取りは前空きや後空きとはまったく別のもので、指定された行数を必ず確保します。また、行取りとして確保した領域内にどのように見出しを置くかは、次の引数である 付き方 で指定します。

付き方

次のいずれかを指定することができます。

- t 本体 の上端と文字列の上端が一致する
- c 本体 の中心と文字列の中心が一致する
- b 本体 の下端と文字列の下端が一致する

問答

見出しが複数行になった場合の問答量を指定します。たとえば、次のような場合には、見出 しの文字列の長さを考慮して、折れる箇所の一番広い幅を指定するようにしてください。

1.2.1 見出し文字列 1.2.105 見出し文字列

続ける本文行数

改ページを起こさずに見出しの後に続けるべき本文行数を指定します。たとえば"2"を指定した場合、見出しの後の本文が見出しと同じページに2行分収まらなければ、見出しごと改ページします。

なお、見出しの書体は小組などと同じように、同じ環境名で定義されたコンポジットフォントが用いられます。たとえば、

環境名	subsection
和文書体	見出しミン
欧文書体	Times-Roman
和文級数	18 級
欧文級数	18 級
字送り	17 歯(1 歯ツメ)
行送り	17 歯(1 歯ツメ)
文字色	
シアン	0.0
マゼンタ	0.0
イエロー	0.0
ブラック	1.0
行取り	2 行
付き方	下付き
左右揃え	中心に揃える
前アキ	なし
後アキ	なし
表記	算用数字
文字列	節番号. 小見出し番号
問答	なし
続ける本文行数	2 行

という見出しの定義は、

という記述になります。

見出しには、EPS画像を使用することもできます。この場合、文字だけの見出しと比べて指定方法が大きく異なります。

EPS 画像を使用する場合、用意する EPS 画像にちょっとした細工を施しておく必要があります。

見出し番号や見出し文字列が入る箇所に"タグ"と呼ばれる文字列を書いておき、出力時に"タグ"文字列を実際の見出し番号や見出し文字列に自動的に置き換えて出力する、という方法を取ります。このとき、画像中のタグ文字列の書体は実際の出力には関係しません。

EPS 画像を使用する見出しには以下のマクロを使用します。

\setheadpsfrag, \setheadstrbox の 2 つのマクロは見出しの番号部分と文字列部分をそれぞれ指定します。

EPS 拡大率/幅/高さ

EPS 画像の大きさです。拡大率と幅/高さでは拡大率が優先します。拡大率が0の場合、幅/高さで指定された大きさで出力されます。幅または高さが0の場合、0でない方に合わせて縦横比を保ったまま大きさが変化します。いずれも0の場合は、原寸大で出力されます。拡大/縮小した場合、出力される見出し番号や見出し文字列も拡大/縮小されるので注意してください。

EPS ファイル名

出力する EPS ファイル名を指定します。

タグ名

EPS 画像中の見出し番号や見出し文字列が出力される箇所に書いてある文字列を指定します。 これらのタグ文字列には日本語は使用できません。

番号 / 文字列基準位置

実際に出力される見出し番号や文字列の位置揃えの基準点を指定します。上下の基準点と左右の基準点の2文字で書き表します。

上下位置:

- t 上端に揃える
- c 中心に揃える
- b 下端に揃える
- B 文字列のベースラインに揃える

左右位置:

- I 左端に揃える
- c 中心に揃える

r 右端に揃える

タグ基準位置

タグ文字列のどこを基準点にするかを指定します。指定は番号 / 文字列基準位置と同様に 2 文字で指定します。

番号 / 文字列拡大率

実際に出力される見出し番号や文字列の拡大率を指定します。

番号 / 文字列角度

実際に出力される見出し番号や文字列を回転させる角度を指定します。

番号 / 文字列幅

見出し番号や文字列の折り返し幅を指定します。

縦横

見出し番号や文字列の組方向を指定します。" y " で横組、" t " で縦組になります。

折り返し

見出し番号や文字列が指定された幅を超えた場合、折り返すかどうかを指定します。"t"で折り返し有り、"f"で折り返し無しです。

長体幅

番号/文字列幅を超えた場合に、この幅までなら折り返さないで長体を掛けて圧縮し、指定幅に収める、という限度の幅を指定します。つまり、指定幅を最大圧縮率で割った値になります。

段落形態

折り返した場合の段落の形を TeX マクロの \parshape の引数の形で指定します。通常は指定する必要はありません。

実際に使用する場合、書体の指定も番号部と文字列部の2つの書体とスリップを指定する必要があります。EPS 画像が拡大/縮小された場合、書体級数や行送りは一緒に拡大/縮小されます。

EPS 画像を利用した見出しの指定は、

環境名	chapter
番号和文書体	見出しミン
番号欧文書体	Times-Roman
番号和文級数	20 級
番号欧文級数	20 級
番号字送り	20 歯
文字列和文書体	見出しミン
文字列欧文書体	Times-Roman

文字列和文級数 18 級 文字列欧文級数 18 級

文字列字送り 17歯(1歯ツメ)

文字列行送り 22 歯

文字色

 シアン
 0.0

 マゼンタ
 0.0

 イエロー
 0.0

 ブラック
 1.0

起こし 左ページ起こし

行取り5 行付き方下付き前アキなし後アキ1 行表記算用数字

番号文字列 第 見出し番号 章

番号基準位置左、ベースライン揃え番号タグ基準位置左、ベースライン揃え

番号拡大率 100% 番号角度 0度 番号幅 5字 番号組方向 横組 番号の折り返し 無し 文字列タグ名 @str@

文字列基準位置左、ベースライン揃え文字列タグ基準位置左、ベースライン揃え

文字列拡大率100%文字列角度0度文字列幅20字文字列組方向横組文字列の折り返し有り

文字列の最大長体率 80%まで許容する

という見出しの場合、

\newsection{chapter}[\arabic]{第\thechapter 章}

```
\headsection{chapter}[t1]{0\Vokuri}{5\Vokuri}[b]{1\Vokuri}[2]
\makeheadeps{chapter}[0.8](0.00mm,0.00mm){chapter.eps}
\setheadpsfrag{chapter}{num}{@n@}[B1][B1][1.0][0]
\setheadstrbox{chapter}{sum}{5\hokuri}[y][f]
\setheadpsfrag{chapter}{str}{@str@}[B1][B1][1.0][0]
\setheadstrbox{chapter}{str}{20\hokuri}[y][t][25\hokuri]
\DeclareCompositFont{num@chapter}[0.0,0.0,0.0,1.0]{25Q}{JY1/mmc/m/n}[25Q]{0T1/ptm/m/n}[0.03zh]
\DeclareCompositFont{str@chapter}[0.0,0.0,0.0,1.0]{22.5Q}[21.25H]{JY1/mmc/m/n}[22.5Q]
{0T1/ptm/m/n}[0.03zh]
\newslip{num@chapter}[1]{27.5H}{\z@}{\z@}{\z@}
\newslip{str@chapter}[1]{27.5H}{\z@}{\z@}
```

となります。

EPS 画像を使用する場合には、上記のように、同時に num@ 環境名 、str@ 環境名 の 2 種類の書体とスリップを定義しておく必要があります。また、この例では画像を 80%に縮小しているため、書体級数や字送り、行送りをその分拡大して指定しておく必要があります。

行取りボックス

見出しを定義する際に絶対に必要となるマクロではありませんが、行取りを行いやすくするためのツール的なマクロを定義してあります。

```
\sec@box{ 行数 }{ 本体 }
```

行数

確保したい行数を、"3\Vokuri"というように指定します。

本体

自由な命令や文字列が書けます。なお、本体 は "\vbox "で囲まれて処理されます。

目次

目次は見出しと関連して定義されます。目次項目には囲み罫を使用できます。指定された見出 しから抽出された目次情報を組版するための定義を、以下のマクロを使って行います。

```
\newtoc{ 名前 }{ リーダ体裁 }{ 左空き , 上空き , 右空き , 下空き }% [ 背景色 ]{ 左上角半径 , 右上角半径 , 右下角半径 , 左下角半径 }% [ 罫線色 ]{ 左罫太さ , 上罫太さ , 石罫太さ , 下罫太さ }% { 前空き }{ 後空き }
```

名前

目次に出力する見出しの環境名です。

リーダ体裁

目次中の見出し文字列とページ番号の間に挿入するリーダの体裁を指定します。

上下左右空き

囲み罫の内側の空きを指定します。

背景色

囲み罫の内側の地の色を CMYK 形式で指定します。

角半径

囲み罫の角を丸める場合の半径を指定します。0を指定すると直角になります。

罫線色

囲み罫の罫線色を CMYK 形式で指定します。

ち太律

上下左右の罫線の太さを指定します。

前後空き

囲み罫の外側の空きを指定します。

また、目次に使用する見出しについて、目次情報を抽出しておくために次の宣言をしておきます。

\ 見出し名 labeltrue

また、目次書体については見出し番号と文字列部分のコンポジットを toc@ 見出し名、ページ番号のコンポジットを page@ 見出し名 で定義しておきます。行送り、字上げ/字下げ、問答は toc@ 見出し名 というスリップで定義します。

たとえば、

見出し名	chapter
見出し文字列和文書体	見出しミン
見出し文字列欧文書体	Times-Roman
見出し文字列和文級数	18 級
見出し文字列欧文級数	18 級
見出し文字列字送り	18 歯
ページ書体	Times-Roman
ページ級数	15 級
文字色	
シアン	0.0
マゼンタ	0.0
イエロー	0.0
ブラック	1.0
行送り	22 歯
問答	4字

字上げ	0字
字下げ	0字
前空き	1 行
後空き	0 行
リーダ	3 点リーダ
囲み罫	
角半径	すべて0(直角)
ち太霍上	0.5mm
ち太霍不	0.5mm
左右罫太さ	0mm
上空き	1mm
下空き	1mm
左右空き	0mm
罫線色	
シアン	0.0
マゼンタ	0.0
イエロー	0.0
ブラック	0.5
背景色	
シアン	0.0
マゼンタ	0.0
イエロー	0.0
ブラック	0.0

のような目次項目の指定は、

```
\chapterlabeltrue
\newtoc{chapter}{\hskipOpt\hbox{}\nobreak\cleaders\hbox{$\m@th \mkern1.5mu\cdot \mkern1.5mu\ship5mm plus1fill}{Omm,1mm,0mm,1mm}[0,0,0,0]{Omm,0mm,0mm,0mm}[0,0,0,0.5]{Omm,0.5mm,0.5mm}{1\Vokuri}{0\Vokuri}
\DeclareCompositFont{toc@chapter}[0.00,0.00,0.00,1.00]{18Q}[18H]{JY1/mmc/m/n}
\DeclareCompositFont{page@chapter}[0.00,0.00,0.00,1.00]{18Q}[18H]{JY1/mmc/m/n}
\OHORDON_0TI/ptm/m/n}
\newslip{toc@chapter}{22H}{4\hokuri}{0\Hokuri}{0\Hokuri}
```

のようになります。

これらの指定により、以下のコマンドが生成されます。

```
    \TOC
    見出し名 {
    見出し番号 }

    \TOCSTR
    見出し名 {
    見出し文字列 }

    \TOCPAGE
    見出し名 {
    見出しラベル }
```

見出しラベルは \ 見出し名 labeltrue を指定したことで自動生成されるラベル名で、\TOCPAGE 見出し名 で使用することでページ番号に変換されます。このラベルと見出しの対応は Work/*.rog というファイルに記述されています。

実際の記述は以下のようになります。

\TOCchapter{第1章}\TOCSTRchapter{電子編集とは}\TOCPAGEchaper{1-10002}

体裁

ベースファイルには、体裁指定を行う命令として、

- ・版面設定
- ・柱/ノンブル設定

のマクロが定義されています。用紙サイズや基本的な版面の設定と、柱やノンブルを出力する位 置を自由に決められるようになっています。

版面

ページの版面と用紙サイズ、さらにノド側マージンと天側マージンを定義するには、次のマクロを使います。

```
\defpage{ 字詰め }{ 行数 }% { ページ幅 }{ ページ高さ }{ ノド }{ トップ } \def\EWB@begindvi{\special{papersize= ページ幅+50mm , ページ高さ+50mm }}
```

字詰め / 行数

版面の文字数と行数を指定します。ただし、行数には実際の版面の行数よりも 1 行少ない行を指定してください。

ページ幅 / ページ高さ

用紙の縦横サイズを指定します。指定する単位は、"mm"あるいは"H"です。

ノド / トップ

用紙サイズから版面までの天側とノド側からの距離を指定します。

\def\EWB@begindvi{...} は PostScropt に変換したときのページ位置調整のための宣言です。 たとえば、用紙サイズが縦 257mm ×横 182mm で、

字詰め 43 字 行数 35 行 トップ 24.75mm ノド 23mm

という体裁を設定する場合には、

\defpage{43\Hokuri}{34\Vokuri}{182mm}{257mm}{23mm}{24.75mm}\def\EWB@begindvi{\special{papersize=232mm,307mm}}

というマクロを記述すれば設定できます。

柱/ノンブル

柱およびノンブルには、出力する位置と体裁の2つのマクロが用意されています。スリップやコンポジットフォントと同様に、柱とノンブルも名前を付けて(この場合には、環境名ではなく、スタイル名と呼んでいる) ひとまとめにして管理するようになっています。

柱およびノンブルを出力する位置は、次のマクロを用いて定義します。

```
\newstyle{ スタイル名 }(x,y)[ L ハシラ POS ](x,y)[ L ノンブル POS ]% (x,y)[ R ハシラ POS ](x,y)[ R ノンブル POS ]
```

" (x,y) " と " [~ POS] " は、それぞれ組みになっています。" (x,y) " には、その後ろに指定した " ~ POS " を出力する位置を数値で指定します。

L ハシラ POS、L ノンブル POS、R ハシラ POS、R ノンブル POS 付き方のいずれかの項目を指定します。

付き方には次のようなものがあります。

 I
 左ツキ
 c
 中ツキ

 r
 右ツキ
 n
 ハシラなし

左/右/中央という位置は、実際に柱やノンブルとして出力する文字列を四角いボックスと考えて、そのボックスのどの部分を、指定した"(x,y)"の位置に合わせるかということです。たとえば、左ツキの"I"を記述した場合には、ボックスの左下隅が指定した"(x,y)"の位置にくるように出力されます。

位置指定の他にも、柱の文字列定義をしなければなりません。柱の文字列を定義するには、次のマクロを使います。

\headingstyle{ スタイル名 }[左右]{ 構造名 }[ハシラ文字列]

左右

"I"か"r"を指定します。左ページ/右ページの意味です。

構诰名

見出しの環境名を指定します。指定した環境名の文字列が柱に出力されます。

たとえば、次のように定義します。

```
\headingstyle{ewbstyle}[1]{chapter}
\headingstyle{ewbstyle}[r]{section}
```

この場合には、左ページには章見出しの文字列が、右ページには節見出しの文字列が出力されることになります。

さらに、ノンブルに出力する数字や文字列を定義するには、次のマクロを使います。

\nombrestyle{ スタイル名 }[表記]{ 左頁文字列 }{ 右頁文字列 }

表記

見出し定義の時に利用した同じ命令を記述し、ノンブルの数字表記を指定します。 左頁文字列 / 右頁文字列

ノンブルとしてどのような数字や文字列を出力するのか指定します。

たとえば、左右ページに"-198-"という形式で出力するには、

```
\nombrestyle{ewbstyle}[\arabic]{--\thepage--}{--\thepage--}
```

と記述します。このマクロの中で使われている "\thepage " は、そのページ番号を表す LATEX 2_{ε} の 変数です。

なお、柱とノンブルの書体は、次のコンポジットフォントがそれぞれ使われるようになっています。そのため、"\DeclareCompositFont"もしくは"\copycomposit"マクロを用いて、事前に定義して置かなければなりません。

	左ページ用		右ページ用	
ノンブル書体	Lnombre@	スタイル名	Rnombre@	スタイル名
柱書体	Lrunning@	スタイル名	Rrunning@	スタイル名

また、これらの柱およびノンブルには罫線または EPS 画像による飾りを付けることが可能です。 罫線で修飾する場合は次のマクロを使用します。

```
\addruletostyle{ スタイル名 }{ 左右 }(x,y)[ 揃え ][ 罫線色 ]%
{ 長さ }{ 高さ }{ 深さ }
```

(x,y)

原点からの位置を指定します。

左右

左右のどちらのページに対する修飾かを指定します。"L"で左、"R"で右ページです。

揃え

指定した位置に対する左右揃えを指定します。"I"で左、"c"で中央、"r"で右揃えになります。

長さ/高さ/深さ

罫線のサイズを指定します。

EPS 画像で修飾する場合は次のマクロを使用します。

```
\addepstostyle{ スタイル名 }{ 左右 }(x,y)[ 揃え ][ 拡大率 ]% ( 幅 , 高さ ){ EPSファイル名 }
```

拡大率/幅/高さ

EPS 画像の大きさです。拡大率と幅/高さでは拡大率が優先します。拡大率が0の場合、幅/高さで指定された大きさで出力されます。幅または高さが0の場合、0でない方に合わせて縦横比を保ったまま大きさが変化します。いずれも0の場合は、原寸大で出力されます。

EPS ファイル名

出力する EPS ファイル名を指定します。

\addruletostyle や\addepstostyle は1つのページスタイルに対して複数指定することができます。このため、罫線を組み合わせて囲み罫のようにして使用することも可能です。EPS 画像を飾りに使用した場合、PS 化された組版ファイルにはページ毎に EPS ファイルが組み込まれることになるため、ファイルサイズが非常に大きくなってしまうおそれがありますので注意してください。

このように設定した柱およびノンブルのスタイルは、次のマクロを用いて、 T_{EX} に対して明示的に利用することを伝えておかなければなりません。そのためのマクロは、 LAT_{EX} 2_{ε} に用意されているものと同じで、

```
\pagestyle{ スタイル名 }
```

を使います。そのため、たとえば"ewbstyle"というスタイル名で柱とノンブルを定義した場合には、次のように指定します。

\pagestyle{ewbstyle}

また、ある見出しのページにのみ使用する場合は、

\setstyle*{ 見出し名 }{ スタイル名 }

を宣言しておきます。

同じ見出しに \setstyle と \setstyle* で異なるスタイルを指定した場合、見出しの有るページでは \setstyle* で指定したスタイルが、見出し以降のページでは \setstyle で指定したスタイルが使用されます。

脚注

脚注の定義を行うために、ベースファイルには、

- ・脚注のアキ指定
- ・脚注の体裁指定

のマクロが用意されています。

脚注のアキ指定

脚注と脚注や本文と脚注のアキなど、脚注に関連したアキは、次のマクロを用いて指定します。

```
\formatfootnote{ 本文-脚注アキ } 脚注罫線 ]{ 脚注-脚注アキ }
```

本文-脚注アキ

本文と脚注のアキを指定します。

脚注罫線

本文と脚注の間に引かれる罫線を指定します。TEX の罫線命令を指定することができます。脚注-脚注アキ

脚注が複数続く場合の脚注間のアキを指定します。

脚注の体裁指定

脚注の体裁を指定するには、次のマクロを使います。

表記

見出しと同様に、カウンタ番号の表現形式を指定します。

本文マーク/脚注マーク

カウンタ番号の他に、アスタリスクなどの記号を付ける場合には、マークを指定します。 問答 脚注が長くて複数行になる場合の問答量を指定します。

上位カウンタ

見出しのどのレベルで脚注番号のカウンタをリセットするかを指定します。

たとえば、本文と脚注のアキを"24H"にし、版面一杯の罫線を引いて、脚注間のアキを"13H"にしたければ、

\formatfootnote{24H}[\vrule height.1mm depth0pt width\textwidth]{13H}

と定義します。また、問答が3文字で、"注1"のような記号を脚注に付け、本文にはアスタリスクと脚注番号を出力したければ、

と定義するだけです。もちろん、脚注も名前のついた環境となるため、脚注に使われる書体はコンポジットフォントとして、定義しておかなければなりません。

参照

TeX では図表類を取り扱うマクロとして、次のマクロがベースファイルに定義されています。

- ・参照項目のアキ/キャプション設定
- ・フロート設定
- ・非フロート設定
- ・レイアウト設定

次に、それぞれのマクロを見てみましょう。

参照項目のアキ/キャプション設定

参照する項目に対して、統一したアキを設定することができ、参照番号とキャプションを出力 するためのマクロが備わっています。

参照される図表類の前後のアキは、次のマクロを用いて定義します。

```
\formatfloat{ 名前 }{ t-t(b-b)アキ }{ t-(-b)アキ }{ hアキ }% { -pアキ }{ p-pアキ }{ p-アキ }
```

スリップやコンポジットフォント同様に、アキの設定も、やはり名前を付けて管理するようになっています。

"t、b、p、h"をプレフィックスに付けたアキを設定するようになっており、これは、図表類を物理属性エディタでフロート指定した場合の"TOP、BOTTOM、PAGE、HERE"に相当します。

h アキ には、"here"でフロートした際に挿入される前後のアキを指定し、 t-t(b-b) アキ には"top"もしくは"bottom"にフロートする図表の前後に挿入されるアキを指定します。また、 t-(-b) アキ には"top"もしくは"bottom"にフロートした図表と本文との間に挿入されるアキを指定します。このアキの単位は、"mm"あるいは"H"を用います。

キャプションは、LATEX 2_ε に備わっているキャプション命令を再定義して、 ラベル の定義ができるようになっています。キャプションは、次のマクロを使って出力します。

```
\caption{ キャプション }[ ラベル - 枝番 ]
```

また、指定した文字列による参照を行う場合には、"\ref"マクロを使います。このマクロには 2 通りの記述形式があります。

\ref{ ラベル }

もしくは、

\ref[ラベル - 枝番]

です。たとえば、

\caption{.....}[chap01.title]

というようなキャプションを用いて、そのキャプションの付いた図表を参照する場合には、

\ref[chap01.title]

と入力すれば、原稿中のどこに書かれていても、正しく参照番号が出力されます。

また、参照時の書体変更には、次のような "fref@~ "と "ref@~ "で定義されたコンポジットフォントが使われます。

はじめて参照する場合 「fref@ カレントコンポジット 」 2度目以降の参照 「ref@ カレントコンポジット 」

フロート設定

フロートする図表類の要素を定義するには、次のマクロを使います。

\newfloat{ 名前 }[表記]{カウンタ名}{ 文字列 }% { 枝文字列 }[上位カウンタ]

文字列 / 枝文字列

枝番号が付かない場合と枝番号が付く場合のフロート図表を表す文字列を指定します。どちらにも、フロート番号 "\the 名前 "や"\thechapter"などといったカウンタが使えます。 枝文字列 の指定の中では、"\thesubnumber"が枝番号に置換されるようになっています。

カウンタ名

カウンタとして利用する変数名を指定します。

上位カウンタ

カウンタをリセットするタイミングを見出しのレベルで指定します。

たとえば、図専用のフロートを "figure "という名前にして、図番号を "図 1.3 "というような出力にするには、

```
\newfloat{figure}{figure}{\Q\thechapter.\thefigure}%
   {\Q\thechapter.\thefigure(\thesubnumber)}[chapter]
```

と定義すればよいのです。

非フロート設定

プログラムのリストなど、フロートせずに参照されるものを定義するには、次のマクロを使います。

```
\newreferred{ 名前 }[ 表記 ]{カウンタ名}% { 文字列 }{ 枝文字列 }[ 上位カウンタ ]
```

使い方はフロートと同じで、"\caption"もそのまま利用できます。しかし、"\newreferred"は、

- ・必ず here にはいる。オプション指定は無視される 途中でページブレークが可能である
- ・キャプションと本体の揃えは、レイアウト設定のマクロで行える " {ud} [tcb] " しか許されない
- ・キャプションと本体の順番は、原稿中に書かれている順番になる
- ・アタリ罫を引けない

といった違いがあります。

たとえば、参照時に出力する文字列を"リスト1.19"というようにする場合には、

```
\newreferred{source}{source}{\UA}\thechapter.\thesource}\{\UA}\thechapter.\thesource(\thesubnumber)\}[chapter]
```

とマクロ定義します。

レイアウト設定

参照される図表類の本体とキャプションをレイアウトするには、次のマクロを使います。フロートしない場合には反映されないパラメータがありますので注意が必要です。

縦横の大きさが事前にわかっている一定のボックス型をした図表類に対しては、次のマクロを 使ってレイアウトを行うと便利です。

\layoutfloat*(w,h){ キャプション幅 }{ 揃え }{ アキ }

また、フロートせずにページで自由に切れる図表類に対しては、次のマクロを使います。

\layoutfloat*[本体幅]{ キャプション幅 }{ 揃え }{ アキ }

アスタリスク"*"

マクロ名のあとにアスタリスクを付けると、アタリ罫を引きます。

アキ

本体とキャプションとのアキを指定します。

(w,h)

図表類の横"w"と縦"h"のサイズを指定します。単位は"mm"あるいは"H"で指定します。 本体幅

図表類の横の大きさを指定します。

揃え

" {u,d,l,r}[t,c,b] "の中から、それぞれの揃えを選択して指定します。

- u 上キャプション t 左揃え(u、d)または上揃え(l、r)
- d 下キャプション c 中心揃え
- l 左キャプション b 右揃え(u、d)または下揃え(l、r)
- r 右キャプション

なお、アタリ罫の太さも変更できるようになっており、

\lorulewidth= アタリ罫の太さ

という変数に値を設定すれば、任意の太さに変えられます。

キートップの定義

キートップを定義するには、次のマクロを使います。

\newkeytop*{ 名前 }{ 横サイズ }{ 縦サイズ }{ 行送り }

アスタリスク"*"

オプションで指定すると角丸型のキートップを定義することになります。指定しなければ、 直角型のキートップを定義します。ただし、縦組では、角丸のキートップは使えません。

横サイズ

キートップの横サイズを指定します。

縦サイズ

キートップの縦サイズを指定します。

行送り

キートップの文字列が2行になる場合の行送りを指定します。

キートップを使う場合には、

\ 名前 { キーの文字列 }

を用います。 キーの文字列 には、 名前 として定義されたコンポジットフォントが使われます。

\newkeytop*{keyii}{24H}{12H}{12H}
\DeclareCompositFont{keyii}{...

罫線

罫線を定義するには、次のマクロを使います。

\newkei{ 名前 }[罫線色]{ 長さ }{ 太さ }{ 前空き }{ 後空き }

長さ

罫線の長さ(横サイズ)を指定します。

太さ

罫線の太さ(縦サイズ)を指定します。

前/後空き

罫線の前後の文章との空きを指定します。

この罫線はそれだけで 1 行分として扱われます。改行無しで前後に文章を続けることはできません。

囲み罫

囲み罫は、ページで分割するものと分割しないものが作れるようになっています。分割する囲み罫はフロートする図表として使用できません。

ページ分割しない囲み罫を定義するには、次のマクロを使います。

```
\newkakomikei{ 名前 }%
{ 罫外字下げ , 罫外前空き , 罫外字上げ , 罫外後空き }%
{ 罫内左空き , 罫内上空き , 罫内右空き , 罫内下空き }%
[ 罫内背景色 ]%
{ 左上角半径 , 右上角半径 , 右下角半径 , 左下角半径 }%
[ 罫線色 ]%
{ 左罫太さ , 上罫太さ , 石罫太さ , 下罫太さ }
```

罫外字上げ/字下げ

版面と左右罫線との空きを指定します。

罫外前後空き

本文と上下罫線との空きを指定します。

罫内上下左右空き

罫線と罫内文章との空きを指定します。

角半径

四隅の角を丸める場合の半径を指定します。0であれば直角になります。

罫内背景色

罫線内の背景色を CMYK 形式で指定します。

罫線色

罫線色を CMYK 形式で指定します。

ち太律

上下左右の罫線の太さを指定します。

罫内行幅

罫線内の文章の行幅を指定します。

ページ分割する囲み罫の場合は次のようなマクロで指定します。

```
\DeclareWaku{ 名前 }%
{ 罫外前空き }{ 罫外後空き }{ 罫外字下げ }{ 罫外字上げ }%
{ 罫内上空き }{ 罫内下空き }{ 罫内左空き }{ 罫内右空き }
\DeclareWakuFrame{wakui}{all}{ 枠定義 }
\DeclareWakuFrame{wakui}{body}{ 枠定義 }
\DeclareWakuFrame{wakui}{tail}{ 枠定義 }
\DeclareWakuFrame{wakui}{tail}{ 枠定義 }
```

罫外前後空き / 字上げ / 字下げ

罫線の外側の空きを指定します。空きの大きさは前空きが \innertopskip、後空きが \innerbottomskip、字下げが \innerleftskip, 字上げが \innerrightskip という変数に保存されます。

枠定義

実際に出力される囲み罫の体裁をページ分割の形態毎に指定します。

ページ分割の形態として

```
all ページ分割が起こらない場合
```

 head
 ページ分割した場合の最初のページの囲み罫

 body
 ページ分割した場合の中間のページの囲み罫

 tail
 ページ分割した場合の最後のページの囲み罫

の4通りがあります。

それぞれの形態毎に、

```
\begin{ascpsbox}%
{ 罫内左空き , 罫内上空き , 罫内右空き , 罫内下空き }%
[ 罫内背景色 ]%
{ 左上角半径 , 右上角半径 , 右下角半径 , 左下角半径 }%
[ 罫線色 ]%
{ 左罫太さ , 上罫太さ , 右罫太さ , 下罫太さ }%
\unvbox\Wakubox%
\end{ascpsbox}
```

のように、囲み罫の形を定義します。

罫内空きは \DeclareWakubox でも定義しますが、こちらで定義したものが優先されます。

たとえば、

環境名	wakui
ち太程上	1.0mm
ち太程不	1.0mm
左罫太さ	1.0mm
右罫太さ	1.0mm
左上角半径	3.0mm
右上角半径	3.0mm
右下角半径	3.0mm
左下角半径	3.0mm
外前空き	1.0 行
外後空き	1.0 行
外字上げ	0.0字
外字下げ	0.0字
内前空き	1.0 行

内後空き	1.0 行
内字上げ	0.0字
内字下げ	0.0字
下端罫太さ	0.0mm
下端右下角半径	0.0mm
下端左下角半径	0.0mm
下端内後空き	0.5 行
上端罫太さ	0.0mm
上端左上角半径	0.0mm
上端右上角半径	0.0mm
上端内前空き	0.5 行
罫線色	
シアン	0.00
マゼンタ	0.00
イエロー	0.00
ブラック	1.00
背景色	
シアン	0.00
マゼンタ	0.00
イエロー	0.00
ブラック	0.10

のような囲み罫を作る場合は、

```
\DeclareWaku{wakui}%
  {1.00\Vokuri}{1.00\Vokuri}{0.00\Hokuri}{0.00\Hokuri}%
  {1.00\Vokuri}{1.00\Vokuri}{0.00\Hokuri}{0.00\Hokuri}%
\DeclareWakuFrame{wakui}{all}{%
  \begin{ascpsbox}%
    {0.00\Hokuri,1.00\Vokuri,0.00\Hokuri,1.00\Vokuri}%
    [0.00,0.00,0.00,0.10]%
   {3.00mm,3.00mm,3.00mm,3.00mm}%
    [0.00,0.00,0.00,1.00]%
    {1.00mm, 1.00mm, 1.00mm, 1.00mm}%
   \unvbox\Wakubox%
 \end{ascpsbox}}
\DeclareWakuFrame{wakui}{head}{%
 \begin{ascpsbox}%
    {\tt \{\nnerleftskip,\nnerropskip,\nnerrightskip,0.50\Vokuri\}\%}
    [0.00,0.00,0.00,0.10]%
    {3.00mm,3.00mm,0.00mm,0.00mm}%
    [0.00,0.00,0.00,1.00]%
    {1.00mm, 1.00mm, 1.00mm, 0.00mm}%
    \unvbox\Wakubox%
 \end{ascpsbox}}
\DeclareWakuFrame{wakui}{body}{%
 \begin{ascpsbox}%
```

```
{\innerleftskip,0.50\Vokuri,\innerrightskip,0.50\Vokuri}%
   [0.00.0.00.0.00.0.10]%
   {0.00mm,0.00mm,0.00mm,0.00mm}%
   [0.00,0.00,0.00,1.00]%
   {1.00mm,0.00mm,1.00mm,0.00mm}%
   \unvbox\Wakubox%
 \end{ascpsbox}}
\DeclareWakuFrame{wakui}{tail}{%
 \begin{ascpsbox}%
   {\innerleftskip,0.50\Vokuri,\innerrightskip,\innerbottomskip}%
   [0.00,0.00,0.00,0.10]%
   {0.00mm,0.00mm,3.00mm,3.00mm}%
   [0.00,0.00,0.00,1.00]%
   {1.00mm,0.00mm,1.00mm,1.00mm}%
   \unvbox\Wakubox%
 \end{ascpsbox}}
```

のように指定します。

また、ここで作った囲み罫を使って、小組等に自動的に囲み罫を発生させることができます。 \mkkogumi や \mkkajyou で作られた環境に対して、

```
\def\waku@ 環境名 {\ 囲み罫名 }
\def\endwaku@ 環境名 {\end 囲み罫名 }
```

のように宣言しておくと、その環境は自動的に囲み罫で囲まれます。

kogumi という名前の小組を使用した場合に wakui という囲み罫で囲む場合は

```
\def\waku@kogumi{\wakui}
\def\endwaku@kogumi{\endwakui}
```

のように宣言します。

この場合は、kogumi という名前の小組を使用した場合にその小組は wakui という名前の囲み罫で囲まれます。

ルビ

ルビを定義するには、次のマクロを使います。

```
\newruby{ 名前 }[ 揃え ]{ 欧文書体名 }{ 横和文書体名 }{ 縦和文書体名 }% { ルビ級数比率 }[ 食い込み ]{ 本文-ルビ空き }
```

揃え

ルビ文字列をルビを付ける本文文字列のどこに揃えるかを指定します。

- f ルビ文字列の先頭と本文文字列の先頭を揃える
- c ルビ文字列の中心と本文文字列の中心を揃える
- b ルビ文字列の末尾と本文文字列の末尾を揃える

書体名

欧文書体名、縦 / 横和文書体名を TFM 名で指定します。書体の TFM 名は Appendix D の書体 一覧表を参照してください。

ルビ級数比率

ルビとして出力する文字の大きさを、ルビを付ける本文文字の大きさを 1.0 として比率で指定します。

食い込み

ルビ文字列がルビを付ける本文文字列より幅が大きい場合、前後の本文文字列にどこまで食い込ませるかを本文文字数で指定します。0 であれば食い込ませず、はみ出したルビの分だけ前後の本文が離れることになります。

本文-ルビ空き

ルビ文字列と本文文字列をどれだけ離すかを指定します。

たとえば、

ルビ名 rubyi 揃え 中

欧文書体名Times-Roman横和文書体名リュウミン L縦和文書体名リュウミン L

ルビ級数比率0.5食い込み0.5 文字本文-ルビ空き0.1 文字

のようなルビであれば、

\newruby{rubyi}[c]{rptmr}{Ryumin-Light-H}{Ryumin-Light-V}{0.5}{0.5}{0.1zh}

と指定します。

傍点

傍点を定義するには、次のマクロを使います。

\newbouten{ 名前 }{ 傍点文字 }{ 書体名 }% [傍点級数比率][本文-傍点空き][傍点位置補正]

傍点文字

傍点として出力する文字を指定します。

書体名

傍点書体名を TFM 名で指定します。書体の TFM 名は Appendix D の書体一覧表を参照してください。

傍点級数比率

傍点として出力する文字の大きさを、傍点を付ける本文文字の大きさを 1.0 として比率で指定します。

本文-傍点空き

傍点と本文をどれだけ離すかを指定します。

傍点位置補正

傍点と傍点を付ける本文文字は中心揃えになっています。傍点位置を左右にずらす場合、このパラメータを指定します。

たとえば、

傍点名 bouteni 傍点文字 \

傍点書体名 リュウミン L

傍点級数比率0.7本文-傍点空き0.1 文字傍点位置補正0.0 文字

のような傍点は、

\newbouten{bouteni}{ \ }{Ryumin-Light-H}[0.7][0.1zh][0.0zw]

と指定します。

また、このマクロを利用してアンダーラインを発生させることもできます。

\newbouten{uline}{\vrule width1.2zw height0.1mm depth0.0mm}{Ryumin-Light-H}[1.0]
[-1.1zh][0.0zw]

のように傍点文字に \vrule を、本文-傍点空きにマイナスを指定すると、\uline{......} はアンダーラインとして機能することになります。

2.5.3 マクロ拡張の実例

ここでは、いくつかのサンプルを示しながら、EWB の機能拡張を解説していきます。

T_FX による組版処理の基本的な考え方

サンプルを示す前に、まず T_EX の組版がどのように行われているのか、またどういった要素を基にして処理しているのかを簡単に説明しておきましょう。

TEX の組版処理は、"ボックス"、"グルー"、"罫線"の3つを基本にして行われています。ボックスの大きさや要素はさまざまで、最小の単位が文字というボックスです。文字のボックスを横に並べたものが行のボックスで、行のボックスを縦方向に重ねたものがパラグラフのボックスとなります。そして、パラグラフとパラグラフを積み重ね、柱やノンブルのボックスを加えたものがページのボックスとなります。

TEX はボックスの形状(高さ、横幅、深さ)だけを基に、美しく見えるように並べたり重ねたりして組版しており、実際の文字や単語の意味は知りません。つまり、"a"や"g"という文字ボックスは、ボックスの形状が違うだけで、その中身の文字の形はまったく意味を持たないのです。組版時に参照するフォントにも、そのような情報しか記述されていません。

また、作成するボックスの種類によって、縦か横のどちらか一方向にしか並べられないようになっています。行のボックスを作成している際には横方向にしか並べられず、パラグラフのボックスを作成している場合には、縦方向にしか重ねられません。そのため、任意のボックスを作成する際には、どちらの方向のボックスを作成しているのかをきちんと認識していなければなりません。

さて、このボックス同士を並べたり重ねたりする際に、その間にグルーと呼ばれる伸び縮み自由な糊を付け加えることができます。このグルーは縦/横方向に伸び縮みし、マイナス方向に縮むこともできます。このグルーを付け加えることによって、文字間や単語間のアキ調整を行い、また行ボックス間のアキ、すなわち行間のアキ調整を行っているのです。文字間のグルーは、参照するフォントによって適切なグルーが半自動的に挿入されます。また、行間やパラグラフの間は、それぞれ専用のパラメータを調整することで自由に変えられます。もちろん、意図的に縦横方向の空白を挿入することもできます。

ボックスとグルーの関係は、ちょうど左官屋さんがセメントを使ってレンガのブロックを重ね合わせてブロック塀を組み上げて行くのと似ています。セメントをグルー、レンガをボックスと考えればわかりやすいでしょう(図2-5)。

もう1つの要素である罫線は、ボックスの特殊なものに相当します。罫線にも高さ、幅、深さがあり、自由な長さ、太さの罫線を任意の場所に作成することができます。ただし、ボックス同様に、縦方向か横方向のどちらか一方向の罫線しか作成できないようになっています。

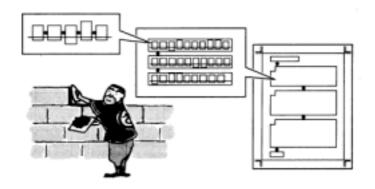


図 2-5 ボックスとグルー

この3つの要素が T_EX の組版処理の基本となります。そして、T_EX では"ボックス"、"グルー"、 "罫線"は、それぞれ次の命令を使って表現します。

	横方向	縦方向
ボックスを作成する命令	\hbox	\vbox
罫線を作成する命令	\hrule	\vrule
グルーを挿入する命令	\hskip、\hss	\vskip、\vss

それでは、実際にマクロを作成して EWB を機能拡張するサンプルを説明してみましょう。ここにあげるサンプルはあくまでも例題ですので、その実現方法は他にいくつもあります。なお、ここでは、

- ・マクロの作成方法
- ・トリガの定義方法

の2段階に分けて解説します。作成したマクロ用のトリガは、体裁入力エディタで自動的に作成される"triger.diff"に追加するようにしてください。また、作成したマクロは必ず"diff.sty"という名前のファイルを作成して追加してください。

サンプル 1 T型の罫線

まず最初に、特殊な罫線を本文中に引くマクロを考えてみましょう。ここでは、次のような罫線を自由な位置に引くものとします。

……たとえば、パラグラフとパラグラフの切れ目に、こんな風に罫線を引いて、

文章の切れ日、話しの切れ日をはっきりと分けることもできます。……

罫線の太さはいずれも"1H"、左右の縦罫は上下がそれぞれ"20H"とします。

説明図



マクロ解説

罫線は "\hrule " もしくは "\vrule " 命令を使って引きます。この "\[hv]rule " 命令には引数があり、罫線の幅、高さ、深さを指定します。形式としては、次のように記述します。

\[hv]rule width 横幅 height 高さ depth 深さ

値は単位を付けて指定します。単位としては、"mm(ミリ)"や"cm(センチ)"、"pt(ポイント)"などが書け、さらに"H(歯)"や"Q(級数)"も指定できます。また、版面の縦横サイズは、それぞれ縦を"\textheight"、横を"\textwidth"という変数で表すことができます。たとえば、版面の横幅一杯に罫線を引く場合には、この変数を使って

\hrule width\textwidth ...

という指定ができます。"\hrule"、"\vrule"はともに罫線を引くマクロですが、それぞれ横方向の 罫線と縦方向の罫線を表します。どちらの方向の罫線を作成したいかによって選択します。

この罫線を組み合わせれば、作成したNT型の罫線が作成できます。たとえば、左右に引かれている縦の罫線は、幅"1H"、高さ"20H"、深さ"20H"であるため、

\vrule width1H height20H depth20H

という指定になります。これを左右に書いて、その罫線の間に版面一杯の罫線を引けば、望みの 罫線が出来上がるわけです。

\vrule width1Hheight20H depth20H 縦の罫\vrule width\textwidth height1H depth0pt 横の罫\vrule width1Hheight20H depth20H 縦の罫

ただし、このやり方では不備な点が 1 つあります。このマクロを使うと、TEX が " Overfull " という警告メッセージを表示してしまいます。この理由は、" \textwidth " 一杯に引いている罫線の横に、さらに幅 " 1H " ずつの罫線を左右に引いているため、" 2H " だけ版面を飛び出してしまっているからです。そのため、実際には版面一杯の罫線ではなく、" \textwidth - 2H " というサイズの罫線を引かなければなりません。

版面の横幅は字詰めや字送りによってわかるため、たとえば 13Q、25 字詰め、べた送りならば 13Q × 25W - 2H = 323H として、横幅から "2H"差し引いた値を計算して罫線を引けばいいのです。しかし、このやり方では、版面の大きさが変更になったりすると、この計算を再び、やり直さなければなりません。そこで、版面のサイズを変数に代入しておき、計算自体を TEX に行わせるようにすれば、融通性の効くマクロになります。寸法用の変数を定義して、その変数に横幅を代入し、はみでている "2H"を引くようにするのです。このようにすれば、横幅が変っても、横幅の変数の値が自動的に変るため、マクロを変更する必要がなくなるわけです。

では次に、寸法用変数の定義や計算の仕方について簡単に説明します。まず、寸法用の変数を 定義するには、"\newdimen"命令を使います。次のように記述します。

\newdimen\变数名

これで、"\変数名"という寸法が定義できます。定義した変数に値を代入するには、

\变数名=寸法

とするだけです。たとえば、"\tmpdimen "という変数を定義し、"2H "という値を代入するには、

\newdimen\tmpdimen 変数を定義し、 \tmpdimen=2H 値を代入する

とします。また定数だけでなく寸法用の変数も右辺に指定できるため、たとえば版面の縦サイズ を代入するには、

\tmpdimen=\textheight

という指定もできます。もちろん、定義した変数と組み込みの寸法用変数には何の違いもありません。寸法を記述できる場所に対して自由に使えます。

次に、足し算と引き算ですが、残念ながら TeX には足し算の命令しかありません。そのため、引き算は足し合わせる値をマイナスにすることで実現します。 足し算は、

\advance\変数 by 值

という命令を使って計算させます。計算結果は、"\変数"に代入されます。すなわち、C のようなプログラミング言語で"i=i+5;"と記述するのと同じことになります。たとえば、"\a"という変数に"10mm"を代入し、それに"2H"加えるには、次のように行います。

```
\newdimen\a ............. 変数を定義して、
\a=10mm ................値を設定して、
\advance\a by 2H ....... 次に計算を行う
```

これで、"\a"には"10mm + 2H"、すなわち"10.5mm"が代入されます。

さてこれで、変数の定義方法、値の計算方法がわかりました。最終的なマクロ定義は、次のようになります。"diff.sty"という名前のファイルを作成し、次のマクロを記述しておいてください。

トリガ定義

マクロの作成はこれで終りですが、次に EWB で利用できるようにトリガを定義しなければなりません。トリガの追加 / 修正は、"triger.diff"ファイルに対して行います。

"triger.diff"ファイルの最後に、次の1行を追加してください。

```
defsimple trule "\\Tbackrule"
```

"//trule"という名前の単純トリガを追加したことになります。この単純トリガは"\Tbackrule"に 展開されます。

試しに、次のような EWB のファイルを作成して実際に処理してみましょう。

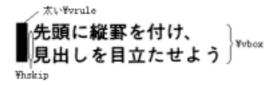
//trule

サンプル 2 縦罫付き見出し

サンプル1に示した罫線を応用して、文字列と罫線を組み合わせた見出しを作成してみましょう。

■ 先頭に縦罫を付け、 ■ 見出しを目立たせよう

説明図



マクロ解説

ボックスを作るには " \hbox " と " \vbox " という命令があります。" \hbox " は横方向に組み合わせられるボックス、" \vbox " は縦方向に組み合わせられるボックスを表しています。" \[hv]box " には、それぞれどのくらいの横幅や高さのボックスを作るのかということを、オプションで指定することができます。

```
\hbox to 寸法{...} \vbox to 寸法{...}
```

また、縦横の寸法を指定するもう1つの方法として、"\hsize"と"\vsize"を使うやり方があります。ボックスの中、つまり"{}"で囲まれた中に"\hsize"や"\vsize"を指定すると、そのボックスの横と高さを任意に指定できます。

たとえば、

```
\vbox{\hsize=横の寸法\vsize=縦の寸法
.......
}
```

とすると、指定した縦横サイズの "\vbox"が作れます。

さて、このボックス命令と罫線を組み合わせれば、サンプルのような見出しを作ることができます。たとえば、次のように書くと罫線と横幅 10cm の見出し文字列用のボックスを横に並べることができます。

```
\hbox{%
\vrule width10H height10H depth0pt
\vbox{\hsize=10cm
.......
```

}}

ただし、この指定では罫線の高さに定数を使っているため、一定の大きさに固定されてしまっています。そこで、一度テンポラリのボックスに文字列を挿入してボックスを組み立て、その後で、作成したボックスの高さを調べることによって、罫線の高さを決めるようにします。ボックスを表す変数は、

\newbox\ボックス

という命令で作成できます。作成したボックスには、次の命令を用いて内容を設定できます。

 $\star{inv} \Delta = [hb] box{...}$

代入したボックスの内容を実際に組版させるには、次の命令を使います。

\box\ボックス または \copy\ボックス

このように入力すれば、ボックスの内容が取り出せます。

また、" \wd\ ボックス "、" \ht\ ボックス "、" \dp\ ボックス " という命令で、それぞれボックス の " 幅 "、" 高さ "、" 深さ " を調べることができます。

\wd\ボックス 横幅を表す \ht\ボックス 高さを表す \dp\ボックス 深さを表す

これらの命令を組み合わせて、最終的なマクロ定義を行うと次のようになります。

これでマクロとしては正しく作成できたのですが、このマクロを見出しトリガとして応用する 場合には、さらにテクニックを使わなければなりません。

見出しは、LATeX2gでは "\chapter"、"\section"などといったマクロを用いて指定します。この

マクロは内部で、

 $\mbox{\colored}$ \circ make ~ head #1#2

という形式のマクロを呼び出すようになっています。"~"には、対応するレベルの見出しを表すマクロ名、たとえば"chapter"や"section"などが当てはまります。

仮に"\chapter"マクロを使った場合には、内部では"\@makechapterhead"マクロが呼び出され、実際の見出しが処理されます。このマクロを定義し直せば、新たにトリガを登録しなくとも、これまでの"//i~//iiiii"で望みの見出しを出力することができます。

なお、この"\@make~head"マクロは、引数を2つ取るようになっています。最初の引数は、 "第3章"のように、見出しに統一して出力される文字列や見出しの番号です。2番目の引数には、 章ごとに変る見出しの内容です。

このサンプルの見出しでは、章番号や章タイトルも一緒にして扱うため、次のように用いて、 内部のマクロを定義し直すだけで動作させることができます。

```
\def\@makechapterhead#1#2{%
\rulemidashi{#1#2}
}
```

マクロを見出しとして利用する場合には、トリガを新しく登録する必要はありません。

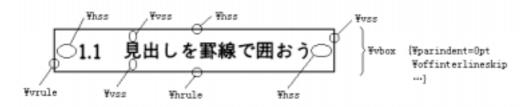
サンプル 3 囲み罫

次に、囲み罫を考えてみましょう。

囲み罫は \newkakomikei マクロで作ることができますが、ここでは単純な罫線とボックスの組み合わせで見出しを囲む方法を説明します。

1.1 見出しを罫線で囲おう

説明図



マクロ解説

罫線を四角いボックスのように囲って出力するには、"\vbox"の中に、そのボックスの天地左右 一杯に罫線を4本引けば実現できます。その際、利用する"\vbox"は次のように書きはじめます。

```
\vbox{\parindent=0pt\offinterlineskip
......
}
```

"\parindent"は段落インデントの大きさを表す変数です。"\offinterlineskip"はマクロで、行間のアキをまったくなくす働きをします。そのため、この"\vbox"の中に書いたものは、インデントを行わず、また上下に隙間を空けずに並ぶことになります。

次に、この中に罫線を四隅に寄せて引くのですが、ボックスの端まで寄せるにはグルーやアキを挿入しなければなりません(グルーとスペースは厳密には違う)。

文字間や行間に任意のアキを入れるには、次の命令を使います。

\hskip 値 横方向のアキ \vskip 値 縦方向のアキ

このマクロを使って、アキの大きさを計算して指定すると、

という記述になります。これで横幅一杯にそれぞれの罫を寄せることができます。もちろん、これでもかまわないのですが、グルーには "\hss"と "\vss"という便利なものがあり、面倒な計算を行わずに、与えられた範囲一杯に伸び縮みさせることができます。

```
\hss .... 横方向へ無限に伸び縮みするグルー \vss .... 縦方向へ無限に伸び縮みするグルー
```

この命令を使うと、先のマクロは次のように記述できます。

```
\hbox to\textwidth{%
\vrule width1H depth0pt height60H
\hss
\vrule width1H depth0pt height60H}
```

この"\hbox"は縦の罫線を書いて、その間に無限に伸びる糊を置き、さらにその右側に最初と同じ縦罫線を引くという命令です。その結果、間の糊が無限に伸びるため、指定してある限界の

"\textwidth"まで左右の罫線が押されることになり、版面の左右ぴったりに縦方向の罫線が引かれることになります。

これを応用して、四角いボックス型に罫線を引くには、

```
\vbox{\parindent=0pt\offinterlineskip
\hrule width\textwidth heigh1H depth0pt
\hbox to\textwidth{%
 \vrule width1H height60H depth0pt
 \hss
 \vrule width1H height60H depth0pt}
 \hrule width1H height60H depth0pt}
```

とすればよいことになります。

さて、この箱の中に文字列を書いて、指定したサイズの天地左右中央に位置させるには、次のようなマクロを使います。

```
......
\vrule ......
\hss\vbox to60H{\vss\hbox{#1}\vss}\hss
\vrule ......
.....
```

左右中央に位置させるには対象文字列の左右に"\hss"を挿入し、天地中央に位置させるには上下に"\vss"を挿入すればいいのです。ただし、ここで1つポイントがあります。引数"#1"を余分な"\hbox"で囲っている点です。この"\hbox"を指定しないと、"\vbox"だけでは、デフォルトの横幅が版面一杯になってしまうため、左右の中央に位置させられなくなってしまうのです。この点を注意してマクロを記述すると、次のようになります。

```
\def\kmakimidashi#1{%
\vbox{\parindent=0pt\offinterlineskip
\hrule width\textwidth height1H depth0pt
\hbox to\textwidth{%
\vrule width60H height1H depth0pt
\hss\vbox to60H{\vss\hbox{#1}\vss}\hss
\vrule width60H height1H depth0pt
\hrule width60H height1H depth0pt
\hrule width\textwidth height1H depth0pt}
```

さて、これで囲み罫は作成できたのですが、もう少し手を加えてもっと融通の効くマクロに直 してみましょう。このように作成したマクロでは、囲む罫線の大きさが固定されてしまいます。 縦横のサイズを限定するのではなく、中に記述する文字列の大きさによって可変となるようにし てみましょう。

これは、中身を一度テンポラリのボックスに代入し、その縦横を計ってから罫線を引けば実現できます。そのため、まずボックスに中身を代入するマクロを次のように記述します。

```
\newbox\tmpbox

\def\kmakimidashi#1{%
 \setbox\tmpbox=\vbox{\selectcomposit{...}\selectslip{...}
 \leftskip=1\hokuri \rightskip=1\hokuri
 \vskip1\vokuri #1 \vskip1\vokuri}
 ......
}
```

ここで、"\leftskip "と"\rightskip "という命令を使っています。これは、中にはいる文字列に対してぴったりの罫線を引くと、その文字列に罫線がくっついてしまうためで、"\leftskip "と"\rightskip"で左右を1文字ずつ空けているのです。同じように、上下も"\vskip"命令で1行ずつ空けてあります。もちろん、"\leftskip"と"\rightskip"命令を使わなくても、スリップを定義して利用すれば、同じ効果を得ることができます。

次に、ボックスの高さと幅を調べて、その大きさの罫線を天地左右に引くマクロを作成します。 これは、次のようになります。罫線の引き方は、前に説明したマクロと同じ方法です。単純に、 罫線の寸法を変数に換えているだけです。

```
\newdimen\tmpwd \tmpwd=\wd\tmpbox
\newdimen\tmpht \tmpht=\ht\tmpbox

\vbox{\parindent=0pt\offinterlineskip
\hrule width\tmpwd height1H depth0pt
\hbox to\tmpwd{%

\vrule width1H height\tmpht depth0pt
\hss

\vbox to\tmpht{\vss\hbox{\box\tmpbox}\vss}
\hss

\vrule width1H height\tmpht depth0pt}

\hrule width1H height\tmpht depth0pt}
```

このマクロを、先ほどのボックスを作成するマクロと一緒に定義すれば、罫囲みを行うことができます。

```
\newbox\tmpbox
\newdimen\tmpwd \newdimen\tmpht

\DeclareCompositFont{kmaki}[0.0,0.0,0.0,1.0]{18Q}[17H]{JY1/mmc/m/n}[18Q]{0T1/ptm/
m/n}[0.03zh]
\newslip{kmaki}{24H}{0H}{0H}{0H}

\def\kmakimidashi#1{%
\setbox\tmpbox=\vbox{\selectcomposit{kmaki}\selectslip{kmaki}}
\leftskip=1\hokuri \rightskip=1\hokuri
\vskip1\vokuri #1 \vskip1\vokuri}
%
```

```
\tmpwd=\wd\tmpbox \tmpht=\ht\tmpbox
%
\vbox{\parindent=0pt\offinterlineskip
\hrule width\tmpwd height1H depth0pt
\hbox to\tmpwd{%
\vrule width1H height\tmpht depth0pt
\hss\vbox to\tmpht{\vss\hbox{\box\tmpbox}\vss}\hss
\vrule width1H height\tmpht depth0pt}
\hrule width1H height\tmpht depth0pt}
\hrule width\tmpwd height1H depth0pt}}
```

トリガ定義

このマクロをトリガとして利用する場合には、いくつかの方法が考えられます。1つには、見出しに応用する方法、もう1つは、グループトリガとして利用する方法です。前者のように見出しとして登録する場合には、少しマクロを追加するだけで、トリガのファイルを修正する必要はありません。また、後者のようにグループトリガとして利用できるようにするには、次のようなトリガを追加しなければなりません。

```
defgothic kmidashi "\\kmakimidashi{" "}"
```

このように追加すれば、

//kmidashi{囲む文字列//}

と EWB ファイルを作って簡単に利用できます。

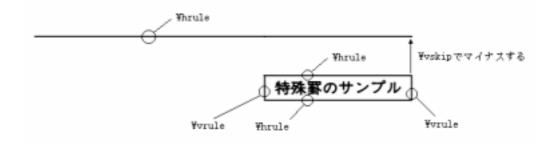
サンプル 4 特殊罫

罫線と文字列とを組み合せて、次のような文字列が付いた罫をサンプルとして作成してみま しょう。

特殊罫のサンプル

罫で囲った文字列の両側は、それぞれ罫から"5H"ずつ空いているものとします。

説明図



マクロ解説

特殊罫のサンプルは、これまでに説明した何パターンかの罫線の作成方法を組み合せれば、簡単に実現できます。

ここでは、まず一時的なボックス変数に文字列を代入し、文字列を罫線で囲ったボックスを作成しておきます。次に、上側の長い罫線を引いて、その右端に作成しておいた罫囲みのボックスを出力するようにします。

罫囲みのボックスは、これまでのサンプルで作成した命令を使って、次のように作成できます。

```
\newbox\tmpbox
\newdimen\tmpwd
\newdimen\tmpht
\DeclareCompositFont{flagfont}[0.0,0.0,0.0,1.0]{18Q}[17H]{JY1/mmc/m/n}[18Q]{0T1/
ptm/m/n [0.03zh]
\def\flagrule#1{%
\setbox\tmpbox=\hbox{\selectcomposit{flagfont}\hskip5H#1\hskip5H}
\tmpwd=\wd\tmpbox \tmpht=\ht\tmpbox
\setbox\tmpbox=\vbox{\parindent=0pt\offinterlineskip
\hrule width\tmpwd height1H depth0pt
\hbox to\tmpwd{%
\vrule width1H height\tmpht depth0pt
\hss\vbox to\tmpht{\vss\hbox{\box\tmpbox}\vss}\hss
\vrule width1H height\tmpht depth0pt}
\hrule width\tmpwd height1H depth0pt}
}
```

さて、作成したボックスと横に長い罫線とを組み合せて出力する部分は、次のようになります。

```
......
\vbox{\parindent=0pt\offinterlineskip
\hrule width\textwidth height1H depth0pt
\vskip-1H
\hbox to\textwidth{\hss\box\tmpbox}}}
```

まず横罫を引き、次に罫線の太さだけ上側に移動して、それから罫囲みしたボックスを版面の右

詰めにして出力しています。この罫線とボックスとがページ替えで分かれてしまわないように、" \vbox " の中に代入し、行間のスペースを Opt にしています。

最終的な特殊罫全体のマクロは、次のようになります。

```
\newbox\tmpbox
\newdimen\tmpwd
\newdimen\tmpht
ptm/m/n}[0.03zh]
\def\flagrule#1{%
\setbox\tmpbox=\hbox{\selectcomposit{flagfont}\hskip5H#1\hskip5H}
\tmpwd=\wd\tmpbox \tmpht=\ht\tmpbox
\setbox\tmpbox=\vbox{\parindent=Opt\offinterlineskip
\hrule width\tmpwd height1H depth0pt
\hbox to\tmpwd{%
\vrule width1H height\tmpht depth0pt
\box{\box{\box{\box{\tmpbox}\tms}\hss}} \label{tmpbox} $$ \box{\box{\tmpbox}\tms} $$ \end{tmpbox} $$ \box{\tmpbox} $$ \box{
\vrule width1H height\tmpht depth0pt}
\hrule width\tmpwd height1H depth0pt}
\vbox{\parindent=0pt\offinterlineskip
\hrule width\textwidth height1H depth0pt
                                        ...... 罫線の太さだけマイナス
\vskip-1H
\hbox to\textwidth{\hss\box\tmpbox}}}
```

トリガ定義

次に、この特殊罫のマクロをトリガに追加してみましょう。" triger.diff " ファイルに次のようなトリガ定義を追加します。

```
defgothic fgrule "\\flagrule{" "}"
```

これで、"//fgrule"というトリガが使えるようになります。原稿中に、

```
//fgrule{サンプル//}
```

と記述すれば、その場所に特殊罫を引くことができます。

2.6 **書体の追加**

EWB の標準状態では和文 21 書体、欧文約 200 書体が使用できますが、ここではさらに市販の 書体や自分で作成した書体を追加する方法を説明します。

2.6.1 **和文書体の追加**

和文書体の追加には ifontadd というコマンドを使用します。

和文フォントを EWB で扱えるようにするために TFM と VF いうフォントデータを作成し、フォントとして EWB システムに登録します。

このコマンドでフォントの実体がインストールされるわけではありません。実際に出力されるフォントは出力デバイスに依存します。

コマンドの実行前に、EWB および T_EX のライブラリにデータを書き込むことができるように、su コマンドで root アカウントになっておいてください。

jfontadd PSフォント名 VF用TFM名 ファミリ名 ウエイト

- ・PS フォント名は Ryumin-Light のようにフォント固有部分のみ、-RKSJ-H 等はつけない
- ・VF 用 TFM 名は ryumin-I のように区別できる適当な名前
- ・ファミリ名は rmn のように 3~4 文字程度
- ・ウエイトは l.r.m.b.h.u のような太さを表す文字

例として、「教科書 ICA-M」を追加する場合を示します。

jfontadd KyokaICA-Medium kyoka-m kyo m

これで EWB システムで「教科書 ICA-M」が使用できるようになります。

実際に追加したフォントを使用する場合は pssted 等を使ってスタイルファイル中でフォントを 指定しておく必要があります。

ghostscript の設定

jfontadd は出力デバイス (ghostscript およびプリンタ) の調整は行いません。そのため、追加したフォントを使用したファイルを ghostview でプレビューした場合、ghostscript が追加したフォントを和文フォントとして認識せず、文字化けしてしまいます。これを防ぐため、次のような設定が必要です。

/usr/local/share/ghostscript/ 以下のディレクトリに pcf.ps または vflib.ps というファイルがありま

す。これらのファイルを書き換えます。

vflib.ps のファイル中に

のような部分があるので、/GothicMB101-Ult...の直後に、次のような1行を追加します。

/<PSフォント名> <数値> (<minまたはgoth>) VFlibkanji

< 数値 > は直前の行の数字に 200 を加えた値になります。たとえば次のようになります。

```
/KyokaICA-Medium 4324000 (min) VFlibkanji
```

pcf.ps についても同様です。

```
:
/FutoGoB101-Bold 4320000 (/usr/X11R6/lib/X11/fonts/misc/k14.pcf) pcfkanji
/FutoMinA101-Bold 4320200 (/usr/X11R6/lib/X11/fonts/misc/jiskan24.pcf) pcfkanji
~中略~
/GothicMB101-Bold 4323600 (/usr/X11R6/lib/X11/fonts/misc/k14.pcf) pcfkanji
/GothicMB101-Ult 4323800 (/usr/X11R6/lib/X11/fonts/misc/k14.pcf) pcfkanji
:
```

vflib.ps の場合と同様に、/GothicMB101-Ult...の直後に、次のような1行を追加します。

/<PSフォント名> <数値> (<pcfファイル>) pcfkanji

<数値>は直前の行の数字に 200 を加えた値になります。たとえば次のようになります。

```
/KyokaICA-Medium 4324000 (/usr/X11R6/lib/X11/fonts/misc/jiskan24.pcf) pcfkanji
```

プリンタ代用フォントの設定

使用したい和文フォントがプリンタにインストールされていない場合、代用フォントの設定が必要になります。

/usr/local/ewb/3.3 にある psfontalias というファイルで代用フォントを定義しておきます。 psfontalias の中身は

```
Ryumin-regular-H Ryumin-Light-H
Ryumin-regular-V Ryumin-Light-V
Ryumin-Medium-H Ryumin-Light-H
Ryumin-Medium-V Ryumin-Light-V
:
```

のようになっており、各行は

<置換されるフォント名> <置換するフォント名>

といった意味です。

「教科書 ICA-M」が EWB で使用できるようになったがプリンタにはインストールされていない、 という場合は

```
KyoukaICA-Medium-H Ryumin-Light-H
KyoukaICA-Medium-V Ryumin-Light-V
KyoukaICA-Medium-RKSJ-H Ryumin-Light-RKSJ-H
KyoukaICA-Medium-RKSJ-V Ryumin-Light-RKSJ-V
KyoukaICA-Medium-83pv-RKSJ-H Ryumin-Light-83pv-RKSJ-H
```

という行を追加しておけば文字化けすることなく代用フォントのリュウミン-L で印刷されます。 代用フォントはプリンタにインストールされている和文フォントであればなんでも構いません。

複数のプリンタを使用している場合、この置換リストファイルは出力するプリンタ名で切り換えることができます。psfontalias と同じディレクトリに

<プリンタ名>fontalias

という名前(プリンタ名 Ip であれば Ipfontalias)で置換リストファイルを作り、EWB での印刷時にプリンタを指定すれば印刷時に参照するファイルが切り替わります。

2.6.2 **欧文書体の追加**

欧文書体の追加には efontadd というコマンドを使用します。

欧文フォントを EWB で扱えるようにするために TFM というフォントデータを作成し、フォントとして EWB システムに登録します。

このコマンドでフォントの実体がインストールされるわけではありません。実際に出力されるフォントは出力デバイスに依存します。

TFM 作成するためのデータとして AFM という文字幅および高さを記述したファイルが必要です。 AFM は PS フォントファイルを元に Macintosh 上で Fontographer というソフトを使って作成できます。

コマンドの実行前に、EWB および TFX のライブラリにデータを書き込むことができるように、

su コマンドで root アカウントになっておいてください。

efontadd PSフォント名 AFM名 ファミリ名 ウエイト シェイプ

- ・AFM ファイルは必ず拡張子が.afm で、AFM 名には拡張子を付けずに記述
- ・ファミリ名は tmr のように 3~4 文字程度
- ・ウエイトは I,r,m,b のように太さを表す文字
- ・シェイプは it,sl,sc のように変形を表す文字。変形なしは n を記述

例として、timesitalic.afm を元に「Times-Italic」を追加する場合を示します。

efontadd Times-Italic timesitalic tms m it

これで EWB システムで「Times-Italic」が使用できるようになります (NFSS 名では OT1/tms/m/itになります)。

実際に追加したフォントを使用する場合は pssted 等を使ってスタイルファイル中でフォントを 指定しておく必要があります。



EWB でのトラブル

これまで本書では、主に、EWB システムの概念やオペレーションを中心に話を進めてきました。そのため、正しく動作することを前提として説明を行ってきました。しかし、実際に EWB システムを利用していると、ちょっとした入力ミスや思い違いで、いろいろなエラーが発生してしまいます。また、EWB システムを利用している際に発生するトラブルの原因は、オペレーションのミスだけでなく、組版に使っている TeX 本体のトラブル、さらに出力するプリンタや表示用端末のハードウェアトラブルなどがあります。

そこで本章では、EWB システムを利用した際のトラブルについて説明します。

3.1 出力時のトラブル

3.1.1 プリンタ出力時のトラブル

組版が終ったファイルをプリンタから出力しようとして、"ファイル名.OUT"と入力しても、プリンタから出力されない場合があります。この多くは、使用している EPS ファイルが正しくないこと原因であることがほとんどです。使用している EPS ファイルが EWB で扱える状態になっているか

% epscheck EPSファイル名

で確認してください。

3.1.2 **プレビュー時のトラブル**

プリンタの場合と同様の原因で、EWB-Shelf の"ファイル名.PRE"コマンドを用いても表示されないことがあります。そのほかに、EPS 画像の中の和文フォントが化ける場合があります。これはプレビューに使用している ghostscript というプログラムが図中の和文を正しく表示していないのが原因です。EPS 画像の中の和文についてはプリンタ出力で確認するようにしてください。

3.2 組版時のトラブル

EWB-Shelf で原稿を組版している際には、入力 / 操作ミスによるトラブル以外にも、組版を行っている TFX がエラーを表示してしまうこともあります。

3.2.1 組版に関する警告メッセージ

EWB-Shelf で原稿ファイルを処理している最中に、"Overfull \nbox"や"Underfull \vbox"というメッセージが表示されることがありますが、これは単にTEX がきれいに組版を行えなかったことを伝えているだけです。この警告メッセージが表示されても、作業は中断されません。組版した結果に問題がなければ気にする必要はありません。

3.2.2 書体に関する警告/エラーメッセージ

マクロを追加したり修正した際に、一部の書体定義を忘れてしまい、そのまま気が付かずに組版処理を行ってしまった時に、TEX がメッセージを表示します。

たとえば、小組や箇条書の和欧文フォントの定義を忘れると、

```
Type H <return> for immediate help.
! No such composit 'gt@honmon'.
.........?x
```

といったメッセージが表示されます。

このような "?" が表示されて EWB システムが停止してしまった場合には、" x " を入力して、 処理を一時中断してください。

トリガの入力ミスはないか、指定通りのトリガを正しい位置に記述しているかどうかをまず調べて、再度実行してみてください。それでも直らない場合には、もう一度、マクロファイルを調べてみましょう。

3.2.3 入力ファイルの指定ミス

体裁入力エディタを使って標準の"ewb2latex.cls"以外の名前でスタイルファイルを作成し、EWB-Shelf の book コマンドで原稿ファイルを登録する場合に、利用するスタイルファイルを正しく指定しなければなりません。この指定を行わないと、次のようなメッセージが表示されてしまいます。

```
! I can't find file 'ewb2latex.cls'.
```

1.1 \documentclass{ewb2latex}

Please type another input file name:

同じようなメッセージは、自分が特別に作成したマクロの中でファイルの読み込みを行っている場合にも当てはまります。単純な指定ミスや、何かの理由でファイルがなくなってしまった場合にこのメッセージが表示された時には、自分の端末に設定されているインターラプトキー(ctrl-C や ctrl- DEL に割り当てられている)を押し、一旦処理を強制終了してください。強制終了後、原稿ファイルをよく調べ、正しいファイル名に修正しましょう。

3.3 EPS ファイルの作成ミス

EPS ファイルの改行コードが UNIX 形式になっていない場合、TEX が EPS ファイルを読み込ん だ時点で次のようなメッセージを表示して処理を停止します。

```
! Unable to read an entire line---bufsize=4096. Please ask a wizard to enlarge me.
```

この場合は、読み込まれている EPS ファイルを

% epscheck EPSファイル名

で改行コードが正しいかどうかをチェックしてください。正しくなければ次のコマンドで改行コードを UNIX 形式に変換します。

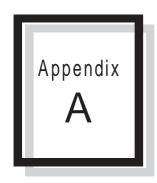
% tounix EPSファイル名

3.4 **その他のトラブル**

EWB-Shelf の操作中に、EWB-Shelf 自身がエラーメッセージを表示することがあります。このエラーメッセージの多くは、単純な入力ミスであったり、原稿ファイルや作業ディレクトリに対して不注意な操作を行ったために表示されます。次に、気を付けなければならない操作をいくつかあげておきます。

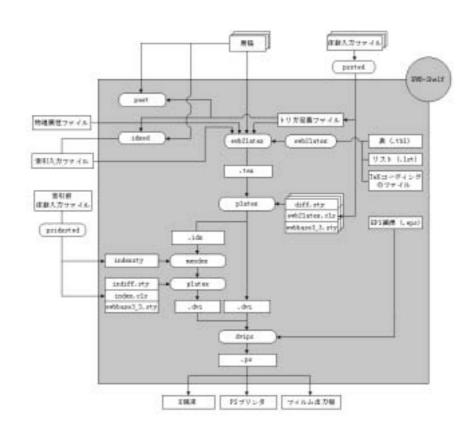
- ・EWB-Shelf が作成した"Work"と"Targets"という2つのディレクトリは、勝手に消さない
- ・原稿ファイルやディレクトリのパーミッションを読み込み不可に変更しない
- ・原稿ファイル名やディレクトリ名を勝手に変更しない。変更した場合には、book コマンドで EWB-Shelf に対して変更したことを伝える

Appendix



EWB のツール群

EWB は、さまざまなツールが有機的にリンクされ、1 つのシステムを構成しています。備わっているツールの種類はいくつもあるため複雑に見えます。そこで、それぞれのツールがどのような関係にあるのかということを示しておきましょう。



Appendix **B**

標準編集トリガ 一覧表

EWB で利用できる標準的な編集トリガー覧表を次に掲載します。

種別	トリガ	補足説明
見出し	//i 文字列	体裁入力ファイル上で個数の増減ができます。
小組	//c1{~//c1} または、//c1{~//}	体裁入力ファイル上で個数の増減ができます。
表組	//table1 { ~ //table1 } または、//table1 { ~ //} または、 //table1[オプション] { ~ //table1 } または、 //table1[オプション] { ~ // }	オプション:= 1 行目のオプション] 2 行目のオプション][n 行目以降のオプション] n 行目のオプション:= 1 カラム目の体裁 2 カラム目の体裁m カラム目の体裁 m カラム目の体裁:= {l,c,r}[(カラム幅)] !:=左揃え c:=中揃え c:=中揃え r:=右揃え カラム幅:= "5w"のように、幅をワード数で指定します。
		ワード数の単位は表組の文字サイズです。 体裁入力ファイル上で個数の増減ができます。
箇条書	//k1{~//k1} または、//k1{~//}	区切りのトリガ " // " が使えます。 体裁入力ファイル上で個数の増減ができます。
脚注	定義部 //ky{ ~ //ky} または、//ky{ ~ //} 参照部 //ky 数字 脚注文字列 //ky 数字 文字列	
強調	//g{~//g} または、//g{~//}	

図表	参照 //f 数字 [-半角英数字] 定義 //f 数字 [-半角英数字] 文字列	EWBでは、キャプションに"図1.1(a)"のような枝番号を付けることができます。 枝番号は"//f1000-a"のように半角ハイフンの後に指定します。 枝番号には半角英数文字が使えます。 体裁入力ファイル上で個数の増減ができます。
ラベル	//label{~//label} または、//label{~//}	
参照	見出し、図表の番号 //numref{~//numref} または、//numref{~//numref} または、//numref{~//numref{~//numref} または、//strref{~//} 見出し文字列、図表キャプション //strref{~//strref} または、//strref{~//} 見出し、図表のページ番号	/}
	//pageref{~//numref} または、//pageref{~/	//}
プログラム	//list1{~//list1} または、//list1{~//}	体裁入力ファイル上で個数の増減ができます。
コメント	//cmt{ ~ //cmt}	コメントトリガはプログラムトリガの中だけで使えるトリガです。 トリガの表記は強調トリガとほぼ同じですが、 開始を表すトリガと終了を表すトリガの間に改 行を入れることができません。
キートップ	//key1{~//key1} または、//key1{~//}	2 段のキートップを作るときは、" ROLL// UP " のように入力します。 体裁入力ファイル上で個数の増減ができます。
罫線	//kei	このトリガの名前に数字は使えません。 体裁入力ファイル上で個数の増減ができます。
囲み罫	//w1{~//w1} または、//w1{~//}	体裁入力ファイル上で個数の増減ができます。
ルビ	//rb1{ 漢字, 読み//rb1} または、 //rb1{ 漢字, 読み//}	体裁入力ファイル上で個数の増減ができます。
傍点	//bou1{~//bou1} または、//bou1{~//}	体裁入力ファイル上で個数の増減ができます。
区切り	//	箇条書やキートップの中で使います。
索引	//in 数字	索引の実体は索引入力エディタまたは EWB エディタで入力します。
ファイル 読み込み	//inc ファイル名	拡張子によって読み込むファイルの種類を判別 します。
LaTeX	$\label{eq:latex} $$ \slash\hspace{-0.5em} $\slash\hspace{-0.5em} \slash\hspace{-0.5em} \slash-0$	このトリガで囲まれた部分のトリガパーシング をしません。
連数字	//suji{~//suji} または、//suji{~//}	縦書き時に半角英数字の向きを縦に合わせま す。
扉画像 読み込み	//TOBIRA{ 横位置, 縦位置, ファイル名//TO BIRA} または、 //TOBIRA{ 横位置, 縦位置, ファイル名//}	EPS 画像ファイルを扉画像として使用します。 横位置、縦位置は用紙の左上からの移動量です。
行間画像 読み込み	//ZUHAN{ 幅, 高さ, 拡大率, 左右揃え, ファイル名//ZUHAN} または、 //ZUHAN{ 幅, 高さ, 拡大率, 左右揃え, ファイル名//}	EPS 画像を読み込みます。行と行の間で使用します。拡大率はパーセントで指定、左右揃えはIで左、cで中、rで右揃えになります。

文中画像 読み込み	//ICON{幅,高さ,拡大率,ファイル名 //ICON}または、 //ICON{幅,高さ,拡大率,ファイル名//}	文中に EPS 画像を読み込みます。文中で使用 します。拡大率はパーセント指定です。
EWB バナー	//ABOUNTEWB	EWB のバナーを出力します。

次に上げるのは、"//"が付かない特殊なトリガです。原稿ファイルの先頭行に入力します。

章番号指定 chapter 番号 章番号を設定します。

Appendix C

組版トリガ一覧表

EWB で利用できる組版トリガー覧表を次に掲載します。

種別	トリガ	補足説明
強制改行	@@break() @@hbreak()	折った行を左右に広げる。 折った行を左揃えにする。
改行の抑制	@@mbox() ~ @@end	囲まれた部分の改行を禁止する。
強制改ページ	@@clearpage(n)	n 回改ページを行う。
白ページ作成	@@siropage(n)	n ページの白ページを作る。 n を省略すると、1 ページだけ作成。
スペースの調整	@ @ vspace(x)	縦方向のスペース調整をする。 単位は mm / H(歯)/ L(行) @@vspace(5I) や@@vspace(3mm) のように指定する。
	@ @hspace(x)	横方向のスペース調整をする。 単位は mm / H(歯)/ W(字) @@hspace(5mm) や@@hspace(3H) のように指定する。
 揃え	@@left() ~ @@end	囲まれた部分を左揃えにする。
	@@center() ~ @@end	囲まれた部分を中揃えにする。
	@@right() ~ @@end	囲まれた部分を右揃えにする。
字下げ	@ @ jisage(n) ~ @ @ end	囲まれた部分を n 文字分だけ字下げする。 n には 1~9 までを指定する。 字下げの単位は基本組の字送り。
表組の項目の前 後のアキの調整	@@lineheight(x,y)	表組のトリガで作った表の項目の前後のアキを広げるために使う。 x に前アキ、y に後アキの調整量を指定する。 単位は mm / H (歯) / L (行)



書体一覧表

EWB で取り扱える和欧文書体の一覧表を次に掲載しておきます。

D.1 T_EX フォント

 T_{EX} フォントは Type1 形式で PS ファイル中に埋め込まれるのでプリンタ側にフォントをインストールしておく必要はありません。

TFM 名 NFSS2 形式名
cmr10 OT1/cmr/m/n ABCDEF abcdef 1234567890
cmti10 OT1/cmr/m/it ABCDEF abcdef 1234567890
cmsl10 OT1/cmr/m/sl ABCDEF abcdef 1234567890
CMCSC10 OT1/cmr/m/SC ABCDEF ABCDEF 1234567890
ABCDEF abcdef 1234567890
Cmb10 OT1/cmr/b/n ABCDEF abcdef 1234567890

cmbx10

OT1/cmr/bx/n

ABCDEF abcdef 1234567890

cmtt10

OT1/cmtt/m/n

ABCDEF abcdef 1234567890

cmitt10

OT1/cmtt/m/it

ABCDEF abcdef 1234567890

cmsltt10

OT1/cmtt/m/sI

ABCDEF abcdef 1234567890

cmtcsc10

OT1/cmtt/m/sc

ABCDEF ABCDEF 1234567890

cmssI10

OT1/cmss/m/n

ABCDEF abcdef 1234567890

cmssi10

OT1/cmss/m/sl

ABCDEF abcdef 1234567890

cmssdc10

OT1/cmss/sbc/n

ABCDEF abcdef 1234567890

cmssbx10

OT1/cmss/bx/n

ABCDEF abcdef 1234567890

cmdunh10

OT1/cmdh/m/n

ABCDEF abcdef 1234567890

cmfib8

OT1/cmfib/m/n

ABCDEF abcdef 1234567890

cmff10

OT1/cmfr/m/n

ABCDEF abodef 1234567890

cmfi10

OT1/cmfr/m/it

ABCDEF abcdef 1234567890

cmmi10

OML/cmm/m/it

ABCDEFabcdef1234567890

cmmib10

OML/cmm/b/it

D.2 PS **和文フォント**

縦書き書体の場合、TFM 名の末尾が "-V"、NFSS 形式名の先頭が "JT1"になります。 * が付いているフォントはダミーの書体見本です。

pssted での名前	TFM 名	NFSS2 形式名
Ryumin-Light あいうえま	Ryumin-Light-H らアイウエオ漢字	JY1/rmn/l/n
Ryumin-regular あいうえま	Ryumin-regular-H らアイウエオ漢字	JY1/rmn/r/n
Ryumin-Medium あいうえま	Ryumin-Medium-H らアイウエオ漢与	JY1/rmn/m/n
Ryumin-Bold あいうえま	Ryumin-Bold-H らアイウエオ漢字	JY1/rmn/b/n
Ryumin-heavy あいうえま	Ryumin-heavy-H らアイウエオ漢	JY1/rmn/bx/n
Ryumin-Ultra あいうえま	Ryumin-Ultra-H 3アイウエオ漢気	JY1/rmn/u/n
GothicBBB-Medium あいうえま	GothicBBB-Medium-H アイウエオ漢字	JY1/gbbb/m/n
ShinGo-Light あいうえま	ShinGo-Light-H アイウエオ漢写	JY1/sgo/l/n
ShinGo-regular あいうえま	ShinGo-regular-H アイウエオ漢写	JY1/sgo/r/n
ShinGo-Medium あいうえま	ShinGo-Medium-H らアイウエオ漢写	JY1/sgo/m/n
ShinGo-Bold あいうえま	ShinGo-Bold-H 3アイウエオ漢写	JY1/sgo/b/n
ShinGo-Ultra あいうえま	ShinGo-Ultra-H アイウエオ漢字	JY1/sgo/u/n

*GothicMB101-Bold

GothicMB101-Bold-H

JY1/gmb/b/n

あいうえおアイウエオ漢字

*GothicMB101-UIt

GothicMB101-Ult-H

JY1/gmb/u/n

あいうえおアイウエオ漢字

FutoMinA101-Bold

FutoMinA101-Bold-H

JY1/fmin/b/n

あいうえおアイウエオ漢字

FutoGoB101-Bold

FutoGoB101-Bold-H

JY1/fgo/b/n

あいうえおアイウエオ漢字

MidashiMin-MA31

MidashiMin-MA31-H

JY1/mmc/m/n

あいうえおアイウエオ漢字

MidashiGo-MB31

MidashiGo-MB31-H

JY1/mgo/m/n

あいうえおアイウエオ漢字

Jun101-Light

Jun101-Light-H

JY1/jun/l/n

あいうえおアイウエオ漢字

Jun34-Medium

Jun34-Medium-H

JY1/jun/m/n

あいうえおアイウエオ漢字

Jun501-Bold

Jun501-Bold-H

JY1/jun/b/n

あいうえおアイウエオ漢字

D.3 PostScript 従属欧文フォント(プロポーショナルピッチ)

和文フォントに対するプロポーショナルピッチの従属欧文フォントです。

pssted での名前(和文書体名)	TFM 名	NFSS2 形式名
PCTimes-Roman (Ryumin-Light) ABCDEF abcdef 1234567	rpptmr 1890	JE1/rmn/l/n
PCRR-SE1 (Ryumin-regular) ABCDEF abcdef 12345	rpprsr 67890	JE1/rmn/r/n
PCRM-SE1 (Ryumin-Medium) ABCDEF abcdef 12345	rpprsm 67890	JE1/rmn/m/n
PCRB-SE1 (Ryumin-Bold) ABCDEF abcdef 12345	rpprsb 67890	JE1/rmn/b/n
PCRH-SE1 (Ryumin-heavy) ABCDEF abcdef 12345	rpprsh 67890	JE1/rmn/bx/n
PCRU-SE1 (Ryumin-Ultra) ABCDEF abcdef 12345	rpprsu 67890	JE1/rmn/u/n
PCHelvetica (GothicBBB-Medium) ABCDEF abcdef 12345	rpphvr 67890	JE1/gbbb/m/n
PCSINGL-SE1 (ShinGo-Light) ABCDEF abcdef 123	rppss1 34567	JE1/sgo/l/n 7890
PCSINGR-SE1 (ShinGo-regular) ABCDEF abcdef 123	rppssr 34567	JE1/sgo/r/n 7890
PCSINGM-SE1 (ShinGo-Medium) ABCDEF abcdef 123	rppssm 3456 7	JE1/sgo/m/n 7890
PCSINGB-SE1 (ShinGo-Bold) ABCDEF abcdef 12	rppssb 2345	JE1/sgo/b/n 67890
PCSINGU-SE1 (ShinGo-Ultra) ABCDEF abcdef 1	rppssu 2345	JE1/sgo/u/n 67890

PCMB101-SE1 (GothicMB101-Bold)

rppmbsb

JE1/gmb/b/n

ABCDEF abcdef 1234567890

PCMB101-SE1 (GothicMB101-Ult)

rppmbsu JE1/gmb/u/n

ABCDEF abcdef 1234567890

PCCenturyOldStyle-Regular (FutoMinA101-Bold)

rppcor

JE1/fmin/b/n

ABCDEF abcdef 1234567890

PCHelveticaNeue-Medium (FutoGoB101-Bold) rpphnm

JE1/fgo/b/n

ABCDEF abcdef 1234567890

PCCentury-Bold (MidashiMin-MA31)

rppcnb

JE1/mmc/m/n

ABCDEF abcdef 1234567890

PCHelvetica-Bold (MidashiGo-MB31)

rpphvb

JE1/mgo/m/n

ABCDEF abcdef 1234567890

PCJun101-Light (Jun101-Light)

rppjl

JE1/jun/l/n

ABCDEF abcdef 1234567890

PCJ34-SE1 (Jun34-Light)

rppjm

JE1/jun/m/n

ABCDEF abcdef 1234567890

PCJ501-SE1 (Jun501-Bold)

rppjb

JE1/jun/b/n

ABCDEF abcdef 1234567890

D.4 PostScript 従属欧文フォント(コンスタントピッチ)

和文フォントに対するコンスタントピッチの従属欧文フォントです。

pssted での名前(和文書体名)	TFM 名	NFSS2 形式名
Ryumin-Light.Roman (Ryumin-Light) ABCDEF abcdef 1234567890	rprmlr	JE1/rmn/l/f
 GothicBBB-Medium.Roman (GothicBBB-Medium) ABCDEF abcdef 1234567890	rpgbmr	JE1/gbbb/m/f

ShinGo-Light.Roman (ShinGo-Light) ABCDEF abcdef 1234567890	rpsglr	JE1/sgo/I/f
ShinGo-regular.Roman (ShinGo-regular) ABCDEF abcdef 1234567890	rpsgrr	JE1/sgo/sm/f
ShinGo-Medium.Roman (ShinGo-Medium) ABCDEF abcdef 1234567890	rpsgmr	JE1/sgo/m/f
ShinGo-Bold.Roman (ShinGo-Bold) ABCDEF abcdef 1234567890	rpsgbr	JE1/sgo/b/f
ShinGo-Ultra.Roman (ShinGo-Ultra) ABCDEF abcdef 1234567890	rpsglr	JE1/sgo/u/f
FutoMinA101-Bold.Roman (FutoMinA101-Bold) ABCDEF abcdef 1234567890	rpfmbr	JE1/fmin/b/f
FutoGoB101-Bold.Roman (FutoGoB101-Bold) ABCDEF abcdef 1234567890	rpfgbr	JE1/fgo/b/f
MidashiMin-MA31.Roman (MidashiMin-MA31) ABCDEF abcdef 1234567890	rpmm31r	JE1/mmc/m/f
MidashiGo-MB31.Roman (MidashiGo-MB31) ABCDEF abcdef 1234567890	rpmg31r	JE1/mgo/m/f
Jun101-Light.Roman (Jun101-Light) ABCDEF abcdef 1234567890	rpjlr	JE1/jun/I/f
Jun34-Medium.Roman (Jun34-Medium) ABCDEF abcdef 1234567890	rpjmr	JE1/jun/m/f
Jun501-Bold.Roman (Jun501-Bold) ABCOEF abcdef 1234567890	rpjbr	JE1/jun/b/f

D.5 PostScript 欧文フォント

pssted 書体名 TFM 名 NFSS2 形式名 AvantGarde-ExtraLight OT1/pag/I/n rpagi DEF abcdef 1234567890 AvantGarde-ExtraLightObl OT1/pag/I/sI rpagio ABCDEE abcdef 1234567890 AvantGarde-Book OT1/pag/sm/n ABCDEF abcdef 1234567890 AvantGarde-BookOblique rpagko OT1/pag/sm/sl ABCDEF abcdef 1234567890 AvantGarde-Medium OT1/pag/m/n ABCDEF abcdef 1234567890 AvantGarde-MediumObl OT1/pag/m/sl rpagmo ABCDEF abcdef 1234567890 AvantGarde-Demi OT1/pag/sb/n ABCDEF abcdef 1234567890 AvantGarde-DemiOblique OT1/pag/sb/sl rpagdo ABCDEF abcdef 1234567890 AvantGarde-Bold OT1/pag/b/n ABCDEF abcdef 1234567890 AvantGarde-BoldObl OT1/pag/b/sI rpagbo ABCDEF abcdef 1234567890 Bodoni-Book rpbdk OT1/pbd/sm/n ABCDEF abcdef 1234567890 Bodoni-BookItalic rpbdki OT1/pbd/sm/it ABCDEF abcdef 1234567890 Bodoni rpbdr OT1/pbd/m/n ABCDEF abcdef 1234567890

Bodoni-Bold

rpbdb

OT1/pbd/b/n

ABCDEF abcdef 1234567890

Bodoni-BoldCondensed

rpbdbrc

OT1/pbd/b/c

ABCDEF abcdef 1234567890

Bodoni-BoldItalic

rpbdbi

OT1/pbd/b/it

ABCDEF abcdef 1234567890

Bodoni-Poster

rnhdn

OT1/pbd/u/n

ABCDEF abcdef 1234567890

Bodoni-PosterCompressed

rpbdprp

OT1/pbd/u/c

ABCDEF abcdef 1234567890

Bodoni-PosterItalic

rpbdpi

OT1/pbd/u/it

ABCDEF abcdef 1234567890

Bookman-Light

rpbkl

OT1/pbk/l/n

ABCDEF abcdef 1234567890

Bookman-LightItalic

rpbkli

OT1/pbk/l/it

ABCDEF abcdef 1234567890

Bookman-Demi

rpbkd

OT1/pbk/sb/n

ABCDEF abcdef 1234567890

Bookman-Demiltalic

rpbkdi

OT1/pbk/sb/it

ABCDEF abcdef 1234567890

BrushScript

rpbsr

OT1/pbs/m/n

ABCDE7 abcdef 1234567890

Courier

rncrr

OT1/pcr/m/n

ABCDEF abcdef 1234567890

Courier-Oblique

rnerro

OT1/pcr/m/sl

ABCDEF abcdef 1234567890

Courier-Bold

rpcrb

OT1/pcr/b/n

ABCDEF abcdef 1234567890

Courier-BoldOblique

rpcrbo

OT1/pcr/b/sl

Futura-Light rpful OT1/pfu/l/n ABCDEF abcdef 1234567890 Futura-LightOblique rpfulo OT1/pfu/I/sI ABCDEF abcdef 1234567890 Futura-CondensedLight rpfulrc OT1/pfu/l/c ABCDEF abcdef 1234567890 Futura-CondensedLightOblique rpfuloc OT1/pfu/l/slc ABCDFF abcdef 1234567890 Futura-Book rpfuk OT1/pfu/sm/n ABCDEF abcdef 1234567890 Futura-BookOblique rpfuko OT1/pfu/sm/sI ABCDEF abcdef 1234567890 OT1/pfu/m/n Futura rpfur ABCDEF abcdef 1234567890 Futura-Oblique rpfuro OT1/pfu/m/sl ABCDEF abcdef 1234567890 Futura-Condensed rpfurrc OT1/pfu/m/c ABCDEF abcdef 1234567890 Futura-CondensedOblique rpfuroc OT1/pfu/m/slc ABCDEF abcdef 1234567890 Futura-Bold rpfub OT1/pfu/b/n ABCDEF abcdef 1234567890 Futura-BoldOblique rpfubo OT1/pfu/b/sl ABCDEF abcdef 1234567890 Futura-CondensedBold rpfubrc OT1/pfu/b/c ABCDEF abcdef 1234567890 Futura-CondensedBoldOblique rpfuboc OT1/pfu/b/slc

ABCDEF abcdef 1234567890

Futura-Heavy rpfuh OT1/pfu/sb/n

Futura-HeavyOblique

rpfuho

OT1/pfu/sb/sI

ABCDEF abcdef 1234567890

Futura-ExtraBold

rpfux

OT1/pfu/ex/n

ABCDEF abcdef 1234567890

Futura-ExtraBoldOblique

rpfuxo

OT1/pfu/ex/sl

ABCDEF abcdef 1234567890

Futura-CondensedExtraBold

rpfuxrc

OT1/pfu/ex/c

ABCDEF abcdef 1234567890

Futura-CondExtraBoldObl

rpfuxoc

OT1/pfu/ex/slc

ABCDEF abcdef 1234567890

Garamond-Light

rpgml

OT1/pgm/l/n

ABCDEF abcdef 1234567890

Garamond-LightItalic

rpgmli

OT1/pgm/l/it

ABCDEF abcdef 1234567890

GaramondThree

rpg3r

OT1/pgm/m/n

ABCDEF abcdef 1234567890

GaramondThree-Italic

rpg3ri

OT1/pgm/m/it

ABCDEF abcdef 1234567890

GaramondThree-Bold

rpg3b

OT1/pgm/sb/n

ABCDEF abcdef 1234567890

GaramondThree-BoldItalic

rpg3bi

OT1/pgm/sb/it

ABCDEF abcdef 1234567890

Garamond-Bold

rpgmb

OT1/pgm/b/n

ABCDEF abcdef 1234567890

Garamond-BoldItalic

rpgmbi

OT1/pgm/b/it

ABCDEF abcdef 1234567890

Helvetica-Light

rphvl

OT1/phv/l/n

ABCDEF abcdef 1234567890

Helvetica-LightOblique

rphylo

OT1/phv/l/sl

Helvetica-Condensed-Light

rphvlrc

OT1/phv/l/c

ABCDEF abcdef 1234567890

Helvetica-Condensed-LightObl

rphvloc

OT1/phv/l/slc

ABCDEF abcdef 1234567890

Helvetica

rphvr

OT1/phv/m/n

ABCDEF abcdef 1234567890

Helvetica-Oblique

rnhyro

OT1/phv/m/sl

ABCDEF abcdef 1234567890

Helvetica-Condensed

rphyrrc

OT1/phv/m/c

ABCDEF abcdef 1234567890

Helvetica-Condensed-Oblique

rphvroc

OT1/phv/m/slc

ABCDEF abcdef 1234567890

Helvetica-Narrow

rphvrrn

OT1/phv/m/d

ABCDEF abcdef 1234567890

Helvetica-Narrow-Oblique

rphvron

OT1/phv/m/sld

ABCDEF abcdef 1234567890

Helvetica-Compressed

rphvbrp

OT1/phv/m/cx

ABCDEF abcdef 1234567890

Helvetica-Bold

rphyb

OT1/phv/b/n

ABCDEF abcdef 1234567890

Helvetica-BoldOblique

rphybo

OT1/phv/b/sI

ABCDEF abcdef 1234567890

Helvetica-Condensed-Bold

rphybrc

OT1/phv/b/c

ABCDEF abcdef 1234567890

Helvetica-Condensed-BoldObl

rphyboc

OT1/phv/b/slc

ABCDEF abcdef 1234567890

Helvetica-Narrow-Bold

rphvbrn

OT1/phv/b/d

ABCDEF abcdef 1234567890

Helvetica-Narrow-BoldOblique

rphybon

OT1/phv/b/sld

Helvetica-Black

rphvrc

OT1/phv/bx/n

ABCDEF abcdef 1234567890

Helvetica-BlackOblique

rphyco

OT1/phv/bx/sI

ABCDEF abcdef 1234567890

Helvetica-Condensed-Black

rphycrc

OT1/phv/bx/c

ABCDEF abcdef 1234567890

Helvetica-Condensed-BlackObl

rphycoc

OT1/phv/bx/slc

ABCDEF abcdef 1234567890

ItcKabel-Book

rpikk

OT1/pik/sm/n

ABCDEF abcdef 1234567890

ItcKabel-Medium

rpikm

OT1/pik/m/n

ABCDEF abcdef 1234567890

ItcKabel-Demi

rpikd

OT1/pik/sb/n

ABCDEF abcdef 1234567890

ItcKabel-Bold

rpikb

OT1/pik/b/n

ABCDEF abcdef 1234567890

ItcKabel-Ultra

rpiku

OT1/pik/u/n

ABCDEF abcdef 1234567890

NewCenturvSchlbk-Roman

rpncr

OT1/pnc/m/n

ABCDEF abcdef 1234567890

NewCenturySchlbk-Italic

rpncri

OT1/pnc/m/it

ABCDEF abcdef 1234567890

NewCenturySchlbk-Bold

roncb

OT1/pnc/b/n

ABCDEF abcdef 1234567890

NewCenturySchlbk-BoldItalic

rpncbi

OT1/pnc/b/it

ABCDEF abcdef 1234567890

Optima

rpopr

OT1/pop/m/n

ABCDEF abcdef 1234567890

Optima-Oblique

rpopro

OT1/pop/m/sl

Optima-Bold rpopb OT1/pop/b/n

ABCDEF abcdef 1234567890

Optima-BoldOblique rpopbo OT1/pop/b/sl

ABCDEF abcdef 1234567890

Palatino-Roman rpplr OT1/ppl/m/n

ABCDEF abcdef 1234567890

Palatino-RomanExtended rppIrre OT1/ppI/m/x

ABCDEF abcdef 1234567890

Palatino-Italic rpplri OT1/ppl/m/it

ABCDEF abcdef 1234567890

Palatino-Oblique rppIro OT1/ppI/m/sI

ABCDEF abcdef 1234567890

Palatino-Bold rpplb OT1/ppl/b/n

ABCDEF abcdef 1234567890

Palatino-BoldItalic rpplbi OT1/ppl/b/it

ABCDEF abcdef 1234567890

RmAlpha rprmalp OT1/prm/m/n

ABCDEF abcdef 1234567890

Times-Roman rptmr OT1/ptm/m/n

ABCDEF abcdef 1234567890

Times-RomanOblique rptmro OT1/ptm/m/sl

ABCDEF abcdef 1234567890

Times-RomanCondensed rptmrrn OT1/ptm/m/c

ABCDEF abcdef 1234567890

Times-RomanExtended rptmrre OT1/ptm/m/x

ABCDEF abcdef 1234567890

Times-Italic rptmri OT1/ptm/m/it

ABCDEF abcdef 1234567890

Times-Bold rptmb OT1/ptm/b/n

Times-BoldItalic

rptmbi

OT1/ptm/b/it

ABCDEF abcdef 1234567890

Times-BoldOblique

rptmbo

OT1/ptm/b/sI

ABCDEF abcdef 1234567890

Univers-Light

rpunl

OT1/pun/l/n

ABCDEF abcdef 1234567890

Univers-LightOblique

rpunlo

OT1/pun/I/sI

ABCDEF abcdef 1234567890

Univers-CondensedLight

rpunirc

OT1/pun/I/c

ABCDEF abcdef 1234567890

Univers-CondensedLightOblique

rpunloc

OT1/pun/I/sIc

ABCDEF abcdef 1234567890

Univers

rpunr

OT1/pun/m/n

ABCDEF abcdef 1234567890

Univers-Oblique

rpunro

OT1/pun/m/sI

ABCDEF abcdef 1234567890

Univers-Condensed

rpunrrc

OT1/pun/m/c

ABCDEF abcdef 1234567890

Univers-CondensedOblique

rpunroc

OT1/pun/m/slc

ABCDEF abcdef 1234567890

Univers-Black

rpunc

OT1/pun/sm/n

ABCDEF abcdef 1234567890

Univers-BlackOblique

rpunco

OT1/pun/sm/sI

ABCDEF abcdef 1234567890

Univers-Bold

rnunh

OT1/pun/b/n

ABCDEF abcdef 1234567890

Univers-BoldOblique

rpunbo

OT1/pun/b/sl

ABCDEF abcdef 1234567890

Univers-CondensedBold

rpunbrc

OT1/pun/b/c

Univers-CondensedBoldOblique rpunboc OT1/pun/b/slc

ABCDEF abcdef 1234567890

ZapfChancery-MediumItalic

rpzcmi

OT1/pzc/m/n

ABCDEF abcdef 1234567890

Symbol

rpsyr

U/psy/m/n

ΑΒΧΔΕΦ αβχδεφ 1234567890

PifontUt1 rppfut1 U/psy/m/ut1

mmcm m kmmmcm² mmcm m kmmm²cm² I II III IV V VI

PifontWk1 rppfwk1 U/psy/m/wk1

(日)(月)(火)(水)(木)(金)(日)(月)(火)(水)(木)(金)(株)(有)(資)(名)(相)(社)

Key-IBM rp

rpkibm

U/psy/m/ibm

ABCDEF

1 2 3 4 5 6

Key-MAC

rpkmac

U/psy/m/mac

A B C D E F

1 2 3 4 5

ZapfDingbats

rpzdr

U/psy/m/zap



WoodtypeOrnaments-Two

rpwotr

U/psy/m/wod



約物/記号類一覧表

約物/記号は全角で入力するか半角で入力するかで、その出力が大きく違ってきます。次に約物/記号一覧表を示します。

約物 / 記号		全角横組	全角縦組	半角横組	半角縦組
句点	`			×	×
読点	0	o	٥	×	×
カンマ	,			Ħ	D-
ピリオド				ė	B
波型	~	~	Ţ.	~	_₹
3点リーダ			:	×	×
2点リーダ				×	×
パーレン	(<u>(</u>))
カギカッコ	г	Γ		×	×
ギュメ				×	×
2連ギュメ	«	«		×	×
キッコー	([Ĺ	П
波カッコ	{	-	<u>~</u>	{	
コロン	:		:	Ġ	₽
セミコロン	;	;	;	Ġ	•
シングルクォーテーション	6			Ì	□
ダブルクォーテーション	"	ď		<u>"</u>	□

 		 	 		
円マーク	¥	¥	¥	×	×
スラッシュ	/	/	/	Ø	Ø
バックスラッシュ	\	Ν	X		
感嘆符	!	!	!	<u>[</u>]	.
疑問符	?	?	?	?	[2]
アンダーバー	_			1	
プラス	+	+	+	±	H
マイナス	-	-	-	Е	
アスタリスク	*	*	*	*	*
除算記号	÷	÷	÷	×	×
等号	=	=	II		Ш
正負符号	±	±	±	×	×
左不等号	<	<	<		
右不等号	>	>	>	\triangleright	\square
単価記号	@	@	@	@	@
アンパサンド	&	&	&	&	&
パーセント記号	%	%	%	%	%
番号記号	#	#	#	#	#
ドル記号	\$	\$	\$	\$	S
白丸				×	×
黒丸				×	×
白四角				×	×
黒四角				×	×
上向き白三角				×	×
上向き黒三角				×	×
下向き白三角				×	×
下向き黒三角				×	×
白ひし形				×	×
黒ひし形				×	×
白星型				×	×
黒星型				×	×
<u> </u>		<u> </u>	<u> </u>		



T_EX / L^AT_EX 書籍一覧表

EWB で機能を拡張する場合、日本語 T_EX の知識が必要となります。以下に、 T_EX の参考となる書籍をあげておきます。

- ・改訂新版 T_EX ブック 株式会社アスキー Donald E. Knuth 著 斎藤 信男 監修 鷺谷 好輝 訳 1992 年 T_EX の開発者であるドナルド・E・クヌース教授が著した、T_EX システムに関する詳しいバイブル的な書籍です。マクロを作成する人は一読することをお勧めします。
- ・日本語 $\mbox{Late}X\ 2_{\varepsilon}$ ブック 株式会社アスキー 中野 賢 著 1996 年 $\mbox{TeX}\ \mbox{Late}X\ 2_{\varepsilon}$ の日本語化を行った本人による $\mbox{Late}X\ 2_{\varepsilon}$ の解説書です。PostScript 用のコマンド についても書かれており、マクロ作成時になにかと便利です。
- ・The L^AT_EX コンパニオン 株式会社アスキー Michel Goossens, Frank Mittelbach, Alexander Samarin 共著 アスキー書籍編集部 監訳 1998 年

LATEX 2。で利用できる各種マクロパッケージについての解説書です。

・The L^AT_EX グラフィックスコンパニオン 株式会社アスキー Michel Goossens, Sebastian Rahtz, Frank Mittelbach 共著 鷺谷 好輝 訳 2000 年

LATeX 2gで利用できる各種マクロパッケージについての解説書です。

・日本語 $ext{Late}$ $ext{Late$

UNIX 系システム用の日本語 LATEX 2。および関連プログラムのインストールキットです。

・T_EX by Topic 株式会社アスキー Victor Eijkhout 著 富樫 秀昭 訳 1999 年
T_EX の動作原理についての解説書です。T_EX の上級者向けに、各コントロールシーケンスについて詳細に書かれています。

Appendix

コマンド リファレンス

G.1 EWB 内部コマンド一覧

G.1.1 FWB-Shelf **原稿セット管理プログラム**

ヘルプを表示する book -h

book -help もうすこし詳しいヘルプ 原稿セットの状態を表示する book

book add ファイル名

book addmulti ファイル名 ファイル名 ファ 複数の原稿を一度に追加する

イル名...

book 原稿番号 insert ファイル名

book move 原稿番号 1 原稿番号 2

book exchange 原稿番号 1 原稿番号 2

book delete 原稿番号

book 原稿番号 page ページ番号

book 原稿番号 page -

book 原稿番号 style スタイルファイル名

book 原稿番号 style -

book 原稿番号 {phy | usephy}

book 原稿番号 nophy

book 原稿番号 {index | useindex}

新しい原稿を原稿セットの一番最後に追加する

新しい原稿を指定した原稿番号に挿入する

原稿番号1の原稿を原稿番号2の原稿の位置に移動する

原稿番号1と原稿番号2の原稿の位置を交換する

原稿番号の原稿の登録を削除する

原稿の開始ページ番号を指定する

原稿が継続ページであることを指定する

スタイルファイルを指定する

スタイルファイルの設定を元 (ewb2latex.cls) に戻す

物理属性ファイルを使うことを宣言する

物理属性ファイルを使わないことを宣言する

索引ファイルを使うことを宣言する

book 原稿番号 noindex 索引ファイルを使わないことを宣言する

book 原稿番号 tex 数 TFX を行う回数を指定する

book form 用紙サイズ プリンタの出力用紙サイズを指定する

用紙サイズには B5、A4、B4、A3 が指定可能

用紙サイズの後ろに W を付加することで見開き印刷とな

1回設定すると、ログアウトするまで設定が保存される

環境変数 PATH を EWB 操作前の状態に戻す book -d

本体名.dvi 本体名の原稿から EPS 画像を含まない dvi ファイルを作

成する

本体名.Dvi 本体名の原稿から EPS 画像を含む dvi ファイルを作成す

本体名.DVI 本体名の原稿から面付け用の dvi ファイルを作成する 本体名.PS

本体名の原稿から EPS 画像を含む PostScript ファイルを作

成する

本体名.out [出力ページ] [プリンタ名] 本体名の原稿をプリンタから出力する。EPS 画像は出力し ない。出力ページの指定には、継続ページを表す"-"(半

角ハイフン) と、複数ページを表す""(半角スペース)が

使える

本体名の原稿をプリンタから出力する。EPS 画像も出力 本体名.OUT [出力ページ] [プリンタ名]

する。出力ページの指定には、継続ページを表す"-"(半 角ハイフン) と、複数ページを表す""(半角スペース)が

使える

本体名.pre [出力ページ] [端末名] 本体名の原稿をプレビューする。EPS 画像は表示しない 本体名の原稿をプレビューする。EPS 画像も表示する 本体名.PRE [出力ページ] [端末名]

本体名の原稿に対して物理属性の入力を行う

本体名.phy 本体名.index 本体名の原稿に対して索引項目の入力を行う 本体名.cln 本体名の原稿の作業ファイルを削除する

ALL.PS すべての原稿を面付けした PostScript ファイルを作成する ALL.PDF PDF 用のリンク情報付き PostScript ファイルを作成する ページを選択してリンク無し PDF 用の PostScript ファイ ALL.pdf

ルを作成する

面付けした PostScript ファイルを縮小してプリンタ出力す ALL.out [折り番号]

ALL.cln 面付けした PostScript ファイルを削除する

dvi 原稿セットに登録されたすべての原稿の dvi ファイルを作

成する。EPS 画像は含まない

Dvi 原稿セットに登録されたすべての原稿の dvi ファイルを作

成する。EPS 画像を含む

DVI 原稿セットに登録されたすべての原稿の面付け用 dvi ファ

イルを作成する

PS 原稿セットに登録されたすべての原稿の PostScript ファイ

ルを作成する。EPS 画像を含む

out [プリンタ名] 原稿セットに登録されたすべての原稿をプリンタから出

力する。EPS 画像は出力しない

OUT [プリンタ名] 原稿セットに登録されたすべての原稿をプリンタから出

力する。EPS 画像も出力する

pre [端末名] 原稿セットに登録されたすべての原稿をプレビューする。

EPS 画像は表示しない

PRE [端末名] 原稿セットに登録されたすべての原稿をプレビューする。

EPS 画像も表示する

phyphy の宣言をしたすべての原稿の物理属性の入力を行うindexindex の宣言をしたすべての原稿の索引項目の入力を行うcln原稿セットに登録されたすべての原稿の作業用ファイル

を削除する

G.1.2 ewb2latex トリガ解析フィルタ

ewb2latex [オプション] EWB文書ファイル

起動時オプション

-h ヘルプ表示

-n ファイルがplain text

-N 面付け用LstLxファイルを出力

-e dvipsでEPSファイルを読み込まない

-f <ファイル名> トリガ定義ファイルの指定(指定しなければ ./triger.def) -s <ファイル名> スタイルファイルの指定(指定しなければ ./ewb2latex.sty)

-1 <ファイル名> ログファイルの指定(指定しなければ標準エラー出力)

-o <ファイル名> 出力ファイルの指定 (指定しなければ標準出力)

-p <ファイル名> 物理属性ファイルの指定

-I <ファイル名> インデックスファイルの指定

-T <ファイル名> インデックステンポラリファイルの指定

-S <ファイル名> 見出しテンポラリファイルの指定

-P <番号> 先頭ページの設定 -y EWB-shelfから使用する

G.1.3 past **物理属性エディタ**

past [オプション] EWB文書ファイル

起動時オプション

- -h 簡単なヘルプ
- -n 物理属性ファイルがすでに存在しても読み込まない
- -p 物理属性ファイル 物理属性ファイルを指定する
- -o 物理属性出力ファイル 出力するファイルを指定する
- -t トリガファイル トリガファイルを指定する

[&]quot;-p"で物理属性ファイルを指定しない時には、一番目の原稿ファイルの名前のサフィックスを

phy に変えたものを物理属性ファイルの名前として使います。

一度、物理属性ファイルを作っておけば、次からは物理属性ファイルだけ指定すれば原稿のファイルは勝手に読んできます。

キー操作

リスト画面

上矢印、k 上方向へのカーソル移動 下矢印、i 下方向へのカーソル移動

スペーキー マークの反転

マークが付いている場合はマークを消し、マークが付いていない場合はマー

クを付ける

原稿ファイル名の行で、スペースを押すとすべての行のマークを反転できる

a すべての行にマークを付ける u すべての行のマークを消す

r 一度、編集した行 (+が付いている行) にマークを付ける

RETURN マークした行の編集開始

e カーソルのある行にマークを付けて編集を開始

 s
 物理属性ファイルへの保存

 a
 物理属性エディタの終了

Y コピー

カーソルのある行の内容を一時的に記憶する。ペーストと組み合せて使う

P ペースト

マークを付けた行にコピーで記憶しておいた内容をコピーする。この時、特

定の項目のみを選んでコピーできる

メニュー画面

上矢印、k上方向へのカーソル移動下矢印、j下方向へのカーソル移動

リターンキー下方向へのカーソル移動、項目の実行右矢印、I、ctrl-F選択するフィールドでの項目の選択 (右)右矢印、h、ctrl-B選択するフィールドでの項目の選択 (左)

スペースキー
文字列を入力するフィールドの場合、入力開始

選択するフィールドの場合は項目の選択

ctrl-N 次のページへ移動

N RETURN

ctrl-P 前のページへ移動

P RETURN

ESC リスト画面へ戻る

文字列を入力するフィールド

左右矢印カーソル移動BS(バックスペース)文字の消去ctrl-G文字の消去リターンキー文字列の確定

上下矢印 文字列の確定とフィールドの移動

ESC 入力の取消し

G.1.4 idxed 索引入力エディタ

idxed [オプション] EWB文書ファイル

起動時オプション

-h ヘルプ表示

-t <ファイル名> トリガ定義ファイル名(指定しなければ triger.def)

-i <ファイル名> インデックスファイル名

キー操作

索引入力画面

上下矢印 着目フィールドの動移

スペースキー 着目モードに修正モードでの入力開始

 ctrl-E
 終了

 ctrl-W
 保存

 ctrl-N
 次の画面

 ctrl-P
 前の画面

ctrl-R 検索モードに入る。検索したい文字列を入力して RETURN キーで検索開始

その他の文字 着目フィールドへの入力開始

文字列を入力するフィールド

左右矢印カーソル移動BS(バックスペース)文字の消去ctrl-G文字の消去リターンキー文字列の確定ESC入力の取消し

G.1.5 dvips - DVI ファイルを PostScript ファイルへ変換

dvips [オプション] DVIファイル

主な起動時オプション

-fフィルタ動作モードにする-o <ファイル名>
-t <用紙形式>
-D <数値>出力ファイルの指定
出力用紙サイズの指定
解像度の指定(dpi)

-E EPSファイルとして出力する

G.1.6 ghostview PostScript ファイルを表示

ghostview [オプション] PSファイル

主な起動時オプション

-page <数值>

-dpi <数値>解像度を指定する-portrait縦長画面で表示-landscape横長画面で表示

-{letter|tabloid|a3|a4|a5|b4|b5}

用紙サイズを指定する 開始ページを指定する

-geometry <ジオメトリ> 表示ウィンドウの座標、大きさを指定する

-display <端末名> 出力する端末名を指定する

キー操作

q 終了

スペースキー 次のページへ移動 b 前のページへ移動

k 上 j 下 h 左

キーボード以外のデバイスとしてマウスが使用でき、ボタンは以下のような割り当てになって います。

右 / 左ボタン 拡大

中ボタン (または左右 ページ番号をクリックでそのページへジャンプ ボタンの同時押し)

なお、設定によっては ghostview を起動しても実際には "gv "というプログラムが起動する場合があります。gv でも ghostview と同様に PostScript ファイルの閲覧ができますが、キーボード / マウスの操作が若干異なります。

q 終了

f次のページへ移動b前のページへ移動矢印上下左右ページのスクロール

右 / 左ボタン ページのスクロール

ページ番号をクリックでそのページへジャンプ

中ボタン (または左右 範囲を指定して拡大 ボタンの同時押し)

G.1.7 pspr PS フォントを変換して印刷

pspr [オプション] PSファイル

主なオプション:

-P <プリンタ名>

-f <ファイル名> フォント変換リストファイル

-s <数値> 左右マージン(mm)

-b5 または -B5 B5用紙に印刷

-a4 または -A4 A4用紙に印刷

-b4 または -B4 B4用紙に印刷

-a3 または -A3 A3用紙に印刷

用紙サイズの後ろに"w"または"w"を付加すると2ページ見開きで印刷-t 縦書きモード(見開き印刷時に奇数/偶数ページを左右逆にする)

フォント変換リストファイルの内容に従って PS ファイル内のフォントを置換し、プリンタへ出力します。

フォント変換リストファイルは/usr/local/ewb/3.3/psfontalias で、内容は

置換元PSフォント名 置換先PSフォント名

のリストになっています(区切りはタブまたはスペース)。

たとえばプリンタにリュウミン-R がインストールされていない場合、フォント変換リストファイルに

Ryumin-regular-H Ryumin-Light-H Ryumin-regular-V Ryumin-Light-V

Ryumin-regular-RKSJ-H Ryumin-Light-RKSJ-H Ryumin-regular-RKSJ-V Ryumin-Light-RKSJ-V

Ryumin-regular-83pv-RKSJ-H Ryumin-Light-83pv-RKSJ-H

と記述しておけばリュウミン-R をリュウミン-L に変換して印刷します。

G.1.8 psmentuke - 面付けプログラム

psmentuke [オプション] 入力ファイル名

起動時オプション

ヘルプ表示 -o <ファイル名> 出力ファイルの指定 (指定しなければ標準出力) 面付け情報ファイルの指定 -f <ファイル名> 折り単位を16ページ単位、面付け単位を8ページにする (指定しなければ16ページ折り、4ページ面) 折り単位を8ページ単位、面付け単位を4ページにする -4 折りごとに分割してファイルに出力する ("-o"と共に使用) -s 面ごとに分割してファイルに出力する ("-o"、"-s"と共に使用) -S -X <数値> ページの幅を明示的に指定する(mm) -Y <数値> ページの高さを明示的に指定する(mm) -x <数値> 横方向のオフセットを指定する(mm) 縦方向のオフセットを指定する(mm) -y <数值> 90°回転して出力する -r 左ページから面付け開始(縦書き) -L -M無線綴じ用として面付けする DVIファイルを元に面付けする -d (指定しなければPSファイルを元にする) -с 日付をチェックして出力する 出力する折り番号を指定する <番号> -T 書名を指定する

面付け情報ファイルには以下のパラメータがあります。

パラメータ	意味	指定項目
組方向	本文の組方向	縦/横
折り単位	1 折りあたりのページ数	8/16
面付け単位	1 面あたりのページ数	4/8
無線綴じ	無線綴じの断ちしろの有無	有り/無し
タイトル	本のタイトル	
末尾折り補正	最後の折りを中途半端な折 りにしないための補正	有り/無し
ファイル分割	面付けファイルの分割単位	折り/面
面付けサイズ	面付けフィルムのサイズ	指定無 U/A4/B4/A3/B3/A2/B2/A1/B1/トンボサイズ

G.1.9 mamehon - **面付け** PS ファイルを縮小し **て印刷**

mamehon [オプション] 面付けPSファイル名

起動時オプション

- -P <プリンタ名>
- -a4 または -A4 A4用紙に印刷
- -b4 または -B4 B4用紙に印刷
- -a3 または -A3 A3用紙に印刷(デフォルト)

psmentuke で作成された面付け済み PS ファイルをプリンタに出力します。

面付けすると通常のプリンタでは版面が用紙からはみ出してしまうので、用紙サイズに合わせて縮小されます。

G.1.10 gettoc **目次を生成**

gettoc

EWB 文書から目次を生成するためのツールです。トリガ定義ファイルと EWB 文書として登録されているファイルから抽出したデータで contents.ewb というファイルを生成します。

トリガ定義ファイルとスタイルファイルには、目次に加える見出しレベルについてトリガおよびマクロの記述が必要です。

また、あらかじめ EWB 文書ファイルを全て組版し、見出しを抽出しておかなければなりません。 以上の準備ができたら作業ディレクトリで gettoc を実行し、作成された contents.ewb を book コマンドで EWB 文書として登録します。

G.1.11 mksiori - PDF **用のしおり情報ファイル を作成**

mksiori [オプション]

起動時オプション --h ヘルプ表示

EWB の作業ディレクトリで mksiori を実行すると "siori.dat" という PDF 用のしおり情報ファイルがそのディレクトリに作られます。

siori.dat にはすべての見出しレベルがしおりとして書き出され、ALL.PDF 実行時に参照されます。

G.2 補助コマンド一覧

G.2.1 pssted 本文用スタイルファイル作成

pssted [-h] [-s] [-t] [-j] 本文用体裁ファイル

起動時オプション

- -h ヘルプ表示
- -s スタイルファイルを生成しない
- -t トリガ定義ファイルを生成しない
- -p PDFスタイルファイルを生成しない

本文用のスタイルファイル、トリガ定義ファイル、PDF 用スタイルファイルを生成します。

作成されるスタイルファイル名は ewb2latex.cls、トリガ定義ファイル名は triger.def、PDF 用スタイルファイル名は PDFstyle です。

G.2.2 psidxsted 索引用スタイルファイル作成

psidxsted [オプション] 索引用体裁ファイル

起動時オプション

- -h ヘルプ表示
- -s スタイルファイルを生成しない
- -i インデックススタイルファイルを生成しない

索引用のスタイルファイル類を生成します。

作成されるスタイルファイル名は index.cls および indexsty です。

G.2.3 tounix, tomac, todos テキストコード 変換プログラム

tounix ファイル名 ファイルの漢字コード、改行コードをUNIX用に変換します。

tomac ファイル名 ファイルの漢字コード、改行コードをMacintosh用に変換します。

todos ファイル名

ファイルの漢字コード、改行コードをDOSおよびWindows用に変換します。

ファイル名は複数書くことができます。

これらのコマンドは変換したファイルを元のファイルに上書きするので注意してください。

G.2.4 epscheck EPS ファイルチェックプロ グラム

epscheck EPSファイル名

EPS ファイルの形式がテキスト形式、改行コードが UNIX 形式になっているかをチェックします。また、ヘッダ部分に漢字が入っていれば漢字コードが EUC になっているかチェックします。ファイル名は複数書くことができます。

G.2.5 fontscan PS ファイル中のフォント名 一覧を表示

fontscan PSファイル名

EWB で作成された PS ファイルの中で使用されているフォント一覧を表示します。 ファイル名は複数書くことができます。

G.2.6 jfontadd EWB で使用する和文フォント を追加

jfontadd PSフォント名 VF用TFM名 ファミリ名 ウエイト

和文フォントを EWB で扱えるようにするために TFM と VF を作成し、フォントとして EWB システムに登録します。

このスクリプトでフォントの実体がインストールされるわけではありません。実際に出力されるフォントは出力デバイスに依存します。

- ・PS フォント名は Ryumin-Light のようにフォント固有部分のみ、-RKSJ-H 等はつけない
- ・VF 用 TFM 名は ryumin-I のように区別できる適当な名前
- ・ファミリ名は rmn のように 3~4 文字程度
- ・ウエイトは I,r,m,b,h,u のような太さを表す文字

G.2.7 efontadd EWB で使用する欧文フォントを追加

efontadd PSフォント名 AFM名 ファミリ名 ウエイト シェイプ

EWB で使用する欧文フォントを追加するためのスクリプトです。

欧文フォントを EWB で扱えるようにするために TFM を作成し、フォントとして EWB システムに登録します。

このスクリプトでフォントの実体がインストールされるわけではありません。実際に出力されるフォントは出力デバイスに依存します。

TFM 作成するためのデータとして AFM という文字幅および高さを記述したファイルが必要です。 AFM は PS フォントファイルを元に Macintosh 上で Fontographer というソフトを使って作成できます。

- ・AFM ファイルは必ず拡張子が.afm で、AFM 名には拡張子を付けずに記述
- ・ファミリ名は tmr のように 3~4 文字程度
- ・ウエイトは I,r,m,b のように太さを表す文字
- ・シェイプは it.sl.sc のように変形を表す文字。変形なしは n を記述

G.2.8 eword 英単語およびカタカナ語の抽出

文章中の英単語またはカタカナ語を抽出します。

抽出した単語一覧を uniq, sort フィルタに掛けることで文章中の用語統一を図ることができます。 //で始まる EWB トリガは無視します。

eword [オプション] 入力ファイル名

起動時オプション

-e 英単語のみ抽出 -k カタカナ語のみ抽出 -o <ファイル名> 出力ファイルを指定

- ・英単語は英字またはアンダースコア (_) で始まり、英字、数字、アンダースコア、ピリオド (.)、マイナス (-) が続く 2 文字以上の半角文字のみの文字列。
- ・カタカナ語はカタカナで始まり、カタカナ、中黒 (・)、音引き (ー) が続く全角文字のみの文

字列。

・EWBトリガは無視する。



A	defgothic 264,265
//ABOUTEWB · · · · · · 339	defgroup · · · · · · 264,265
add · · · · · · 184,185	defitem · · · · · · 264,267
\addepstostyle · · · · · 297	deflabel · · · · · · 264,268
addmulti	deflist · · · · · · 264,266
\addruletostyle · · · · · · 296	defnoexpand 264,268
\advance 313	\defpage294
ALL.out · · · · · 230	defref264,269
ALL.PS 229	defsimple · · · · · · 264
ASCII 形式·······255	defstr263,264
	deftab 264,269
В	defTable · · · · · · 264,266
book · · · · · 184	defurl264,269
bottom83	delete · · · · · · 186
\box 316	diff.sty 261
@@break · · · · · · 209,341	\dp · · · · · 316
	DTP19
C	dvips 368
\caption 300	
@ @ center · · · · · · 210,341	E
chapter 96,339	efontadd · · · · · 326,374
@@clearpage · · · · · 209,341	emacs · · · · · · 110
cln · · · · · · · 225	EP19
//cmt · · · · · · 338	epscheck · · · · · · · 255,331,373
contents.ewb · · · · · 214	EPS 画像 · · · · · · · · 139
\copy 316	EPS ファイル・・・・・・52,255
$\label{local_copy} $$ \copycomposit \cdots 276 $$$	EUC 95
\copyslip 279	\EWB@begindvi294
CTS19	EWB-Shelf · · · · · · 93,179,363
	ewb2latex·····365
D	ewb2latex.cls······115,261
\DeclareCompositFont · · · · · · · · · · · · · · · 274	ewbbase3_3.sty · · · · · · · · · · 126,153,261,270,272
\DeclareWaku · · · · · 304	EWB バナー・・・・・・339
\DeclareWakuFrame · · · · · · · 304	EWB マシン・・・・・・94
\def · · · · · · 270	eword · · · · · · 374
defcaption · · · · · 264,267	exchange · · · · · 186,187
defchapter · · · · · 264,270	
defescape264,269	F
deffigure264,267	fontscan 231,373
deffootnote264,268	form 189

\formatfloat299	L
\formatfootnote · · · · · · 298	//label · · · · · · 338
FreeBSD94	//LaTeX · · · · · · 338
	LaTeX トリガ · · · · · · · 108,256
G	\layoutfloat · · · · · · 302
gettoc · · · · · 214,371	@@left210,341
ghostview368	\leftskip · · · · · 320
guishell237	@@lineheight · · · · · · 211,341
guisted · · · · · · 114,160	Linux94
GUI シェル······237	
	M
Н	Macintosh
\hbox · · · · · 311,315	\@make ~ head · · · · · · · 317
@ @ hbreak · · · · · · · · 209,341	\makeheadeps · · · · · 288
\headingstyle · · · · · 295	mamehon · · · · · · 371
\headsection 285	@@mbox·····209,341
here · · · · · 83	MENTUKEinfo · · · · · · 228
\Hokuri 273	\mkkajyou · · · · · 281
\hokuri · · · · · · 273	\mkkogumi · · · · · · 280
\hrule 311,312	mksiori · · · · · · 233,371
\hsize 315	move
\hskip 311,318	
@ @ hspace · · · · · · · · · · · · · · · · · · ·	N
\hss 311,318	NEC 外字 · · · · · · · 99
\ht · · · · · 316	\newbouten 308
	\newdimen 313
l	\newemph 277
//ICON · · · · · · 339	\newfloat · · · · · 300
idxed367	\newfootnote · · · · · 298
//inc · · · · · · 256	\newkakomikei · · · · · · 304
index.cls261	\newkei · · · · · · 303
indexsty	\newkeytop302
indiff.sty262	\newreferred · · · · · · 301
insert	\newruby 307
	\newsection284
J	\newslip278
jfontadd · · · · · 324,373	\newstyle295
@@jisage · · · · · 211,341	\newtoc · · · · · 291
	NFSS2 形式 · · · · · · · · · 274
	\nombrestyle 296

//numref 338	S
	\sec@box · · · · · 291
O	sed · · · · · · 260
\offinterlineskip · · · · · 318	\selectcomposit · · · · · · 277
OUT · · · · · 187,225	\selectslip 279
out · · · · · · 187	\setbox · · · · · 316
Overfull	\setheadpsfrag · · · · · · 288
	\setheadstrbox · · · · · 288
P	\setstyle297
page · · · · · · 83,191	shelf-install······182
PageMaker	siori.dat······234
//pageref338	@@siropage · · · · · 209,341
\pagestyle297	slip278
\parindent 318	//strref · · · · · · 338
past · · · · · 200,365	style193
PDFinfo234	//suji · · · · · · · 338
PDF 情報ファイル······234	
PostScript93	Т
PRE189	Targets 184
pre · · · · · 189	\textheight · · · · · · 312
psfontalias · · · · · 325	\textwidth 312
psidxsted 115,116,372	//TOBIRA · · · · · · · 338
psidxsted.dat · · · · · · 114	todos · · · · · · 372
psmentuke · · · · · · 370	tomac · · · · · · 372
pspr · · · · · 188,369	top · · · · · · 83
pssted · · · · · · 115,116,372	tounix · · · · · 95,110,331,372
pssted.dat · · · · · · 114	triger.def · · · · · · · 261
PS ファイル・・・・・・93	triger.diff · · · · · · 261
PS プリンタ・・・・・・93	
	U
Q	Underfull 330
QuarkXPress17	UNIX22,94
	URL·····77,150,269
R	URL トリガ · · · · · · · · · · 101,107,264,269
\ref · · · · · · 300	usephy 200
@@right · · · · · 210,341	
\rightskip320	V
	\vbox · · · · · 311,315
	vi · · · · · · · 110
	\Vokuri · · · · · · 273

\vokuri273	I
\vrule · · · · · · 311,312	英数字 · · · · · · · · 96
\vsize 315	枝番号 · · · · · · · 64,65,141
\vskip · · · · · · 311,318	エラー・・・・・・・・・・・・159,193,221
@ @ vspace · · · · · · 210,341	エンコード・・・・・・・274
\vss311,318	エントリ・・・・・・・218
	円マーク・・・・・・144
W	
\wd316	オ
Word · · · · · 110	追い込み・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・45
Work184	追い出し・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・45
WYSIWYG17	欧文送り42,96
	欧文級数 · · · · · · · 119
X	欧文書体 · · · · · · · · 119
X 端末······93,94,189	欧文書体の追加・・・・・・・・326
	オートナンバリング・・・・・・・24
Υ	オートレイアウト・・・・・・・・・・・24,81
yakumono.sed·····260	オプション・・・・・・185
	オプション指定 ・・・・・・57
Z	表組トリガ・・・・・・264,266
//ZUHAN·····338	折り単位・・・・・・228
zw · · · · · 273	
_	カ
ア	改行 · · · · · · 98
アキの調整・・・・・・86	改行自動挿入 · · · · · · · 103
アタリ罫・・・・・・205	改行挿入 · · · · · · · 55,120,265
後空き·····51	改行の抑制・・・・・・209,341
アンダーライン309	外部ファイル 254
	改ページ・・・・・・59
1	囲み罫・・・・・・・・54,55,72,128,146,304,338
イタリック・・・・・・158	箇条書 · · · · · · · · · 62,131,281,337
一太郎	箇条書トリガ・・・・・・・・・・・・・・・・・・101,105,264,267
インクルードトリガ・・・・・・・108,256	頭文字 157
インデックススタイル・・・・・・・・115,153	角半径 · · · · · · · 73,147
インデント・・・・・・・69,144	角丸 · · · · · · · · 145
יָּ	カラム・・・・・- 75
	カラム間・・・・・・・157
上付き文字・・・・・・131	

+	原稿の登録・・・・・・・184
キートップ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	原稿の登録解除186
記号97,359	原稿番号185
記号数字 · · · · · · · · 157	
基本組 · · · · · · · 126	コ
脚注 · · · · · · · · 59,130,298,337	項目空き・・・・・・・・・・63,132
脚注トリガ・・・・・・・・・・・・・・・・・101,106,264,268	項目間隔 · · · · · · · 59,129
脚注番号 · · · · · · · 59	項目内行送り・・・・・・58,129
キャプション・・・・・・・・・・・・・・・・・64,140	小口側のマージン46
級数 · · · · · · 41,119	小組・・・・・・・54,127,280,337
行送り41,120,273,278	コマンド・・・・・・・219
行間画像 · · · · · · · 338	コメント・・・・・・・・・・・・・・・・・・69,144,338
強制改行 · · · · · · · 86,209,341	コメントトリガ・・・・・・・・・・・・107,108
強制改ページ・・・・・・86,209,341	コンスタントピッチ・・・・・・67
強調47,66,337	コントロールコード・・・・・・・・99
強調書体 · · · · · · · 119	コンポジットフォント ・・・・・・・・・・・・・・・・・・・・・・・・ 273,274
強調書体の定義277	
強調トリガ・・・・・・・・・・・・・・・・・・101,104,264,265	Ħ
共通設定ファイル・・・・・・160	最小継続本文行数 · · · · · · · 137
行取り51	索引77,93,244,338
行番号 · · · · · · 69,144	索引エディタ244
禁則・・・・・・・・・・・・44,120	索引作成 · · · · · · · 215
禁則解除 · · · · · · · 121	索引スタイルファイル261
	索引体裁入力ファイル・・・・・ 77,92,113,152,160
ク	索引トリガ・・・・・・108
食い込み・・・・・・ 75,149	索引入力エディタ · · · · · · · · · · · · 77,79,93,218
区切り338	索引ファイル・・・・・・・93,215
区切りトリガ・・・・・・・・・・・・・・・・・・63	索引変換情報ファイル · · · · · · · · 262
組み込みトリガ102,107	索引見出し・・・・・・・・・・・・・・・・・・・・・・・・154
組版トリガ・・・・・・・・・・・・・・・・・・・37,85,100,209,341	参照299,338
組方向 · · · · · · 125,228	参照トリガ・・・・・・・・101,106,264,269
グルー・・・・・・310	参照部・・・・・・59,60,64,65
グループトリガ ・・・・・・・・・・・・・・101,102,264,265	
	シ
ケ	仕上がりサイズ・・・・・・・121
罫線 · · · · · · · · · 70,131,145,157,303,310,311,338	字上げ·······41,120
罫線色 · · · · · · 73,148	シェイプ・・・・・・・274
罫線本文空き61	字送り41,119,273
原稿入力 · · · · · · 91	しおり233
原稿の順序変更 · · · · · · · 186	しおり情報ファイル・・・・・・・233
	字下げ・・・・・・・・・・・・・・・41,120,211,341

字詰め41,119,127,154,278	タ
修飾要素 · · · · · · 124	ターゲットディレクトリ · · · · · · · · · 184
上位見出し・・・・・・136	タグ・・・・・・17,139,288
章定義文字·····126	縦組96
章番号指定·····339	タブ・・・・・・57,58,99,129,144
書体 · · · · · · · · 41,118	単純トリガ・・・・・・・・・・・・・・・・・・・・・・・101,102,264
書体一覧表 · · · · · · · 343	段数·····79,153
書体切り換え277	端末 · · · · · · 94
書体の追加・・・・・・・324	段落形態289
書名125,153	
シリーズ・・・・・・・274	チ
白ページ作成209,341	着目フィールド・・・・・・・・・・218
シンボル・・・・・・・・・・・・・・・・・48,53,131	
	テ
ス	定義部・・・・・・59,60,64,65
図253	体裁 · · · · · · 136,294
数式 · · · · · · 254	体裁入力 … 92,113
スクリプト処理・・・・・・260	体裁入力ファイル39,92,113,161
スタイルエディタ・・・・・・115	デザインカンプ・・・・・・・・・・92,113
スタイルファイル ・・・・・・・・・・ 32,92,115,126,153,261	デスクトップパブリッシング・・・・・・19
スタイルファイルの変更192	天側マージン 294
図表64,140,195,338	電算写植システム19
図表トリガ・・・・・・・101,104,264,267	電子組版システム · · · · · · 19
図表の登録・・・・・・242	電子編集13
スペース・・・・・・98	
スペースの調整 · · · · · · · · · · · · 210,341	F
スリップ・・・・・・278	等幅 · · · · · · 67
スリップのコピー・・・・・・279	特殊索引スタイルファイル・・・・・・262
スリップの定義278	特殊スタイルファイル261
_	特殊トリガ定義ファイル261
tz	トップマージン · · · · · · · 46,121
全角	扉画像 · · · · · · · 338
全角送り······96	トラブル・・・・・・・329
全角スペース98	トリガ29,100,260
N.	トリガ開始文字列 · · · · · · · · · 30,100,126
y 	トリガ定義ファイル・・・・・・・32,115,126,261
揃え 341	トリガ定義文字126
	トリガの強さ109
	トリガのネスト・・・・・・110

ナ	フ
中揃え・・・・・・・210	ファイル分割 229
	ファイル読み込み・・・・・・・・338
=	ファミリ・・・・・・・274
入稿94,231,247	フィルタ・・・・・・・241
入力フィールド・・・・・・・201	フォントのコピー・・・・・・・・・276
	フォントの定義 · · · · · · · · · · · · · · · · · · ·
ネ	物理属性195
ネットワーク ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・22	物理属性エディタ · · · · · · · · · 66,93,196,200,242
	物理属性ファイル・・・・・・・・・・・196
J	プリンタ・・・・・・・94
のど側のマージン・・・・・・121	プリンタ出力 ・・・・・・・・・・・・・・・・・・・・・・・ 187
ノド側マージン294	プリンタ代用フォント · · · · · · · 325
ノンブル・・・・・・・・・・・・・・・・・・48,295	プレビュー・・・・・・189
	フロート・・・・・・・・・・・・・・・・83,207,300
八	プログラム・・・・・・・・・・・67,143,338
背景色 · · · · · · 54,73,148	リスト・・・・・・・67
バイナリ形式255	プログラムトリガ・・・・・・・・266
ハイパーリンク ・・・・・・・・・・・・・・・・・・・・・・・・ 77,150	プロポーショナルピッチ · · · · · · · 42,96
ハイフネーション・・・・・・・・24,45,86,98	文書の登録・・・・・・・239
柱48,295	文中画像 · · · · · · · 339
柱 / ノンブル修飾48,124,296	
バックアップ · · · · · · · 235	^
バッチ17	ページサイズ・・・・・・・・121
バッチ方式・・・・・・・17,22	ページスタイル · · · · · · · · · 48,125,153
半角 · · · · · · 42	ページ体裁51
半角カナ 99	頁体裁番号122,124,125
半角スペース98	ページの指定 241
判型 · · · · · · 46	ページ番号・・・・・・157
版面 · · · · · · 127,154,294	ページ番号の設定・・・・・・190
	ページブレーク142
۲	ページ分割・・・・・・73,148
左揃え210	ページモード・・・・・・・・・229
非展開トリガ・・・・・・・・・・・・・・・・101,106,264,268	ベーススタイルファイル・・・・・ 126,153,261,272
非フロート・・・・・・301	ベースライン・・・・・・・・42,119
表253	变形·····41,118,275
表組 · · · · · · · · 56,337	編集トリガ・・・・・・・37,100,337
	变数273

ホ	文字コード・・・・・・・95
傍点 · · · · · · · · 75,149,308,338	文字色42,119
傍点文字 · · · · · · · · 76,150	文字置換263,264
ボールド・・・・・・・・・158	文字变換 · · · · · · · 151
ボックス・・・・・・・310	文字見出し・・・・・・・・・・・・・・・・・・・・・・・156
本体名 · · · · · · 185	問答52,63,105,120,283
本文体裁入力ファイル113,160	
	7
マ	約物97,260,359
マーク201	
マークアップ17	∃
マージン・・・・・・・・46	用紙サイズ・・・・・・189,294
前空き・・・・・・・・51,66	_
マクロ・・・・・・・270	ラ
マクロ拡張・・・・・・・310	ラベル・・・・・・・338
末尾折り補正 229	ラベルトリガ・・・・・・・・・・・・101,106,264,268
丸付き数字100	
_	IJ
Ξ.	リーダ罫54
右揃え210	リスト・・・・・・254
見出し・・・・・・50,133,154,156,284,337	リストトリガ・・・・・・・264
見出し禁則・・・・・・・51	
見出しトリガ・・・・・・・101,102,264,267	JV
見出し番号指定文字列 · · · · · · · 136	ルビ・・・・・73,148,307,338
Д	L
無線綴じ・・・・・・228	・ レイアウト修正······206
XXIII ADIII III	レベル・・・・・・・・・・・・・・・・・・79,158
*	連数字338
 面付け94,228	
面付けサイズ・・・・・・・・・・229	
面付け情報ファイル・・・・・・228	ローマ数字・・・・・・100
面付け単位・・・・・・228	
	ワ
ŧ	ワークディレクトリ・・・・・・184
目次・・・・・・・・・54,138,214,291	和欧文間スペース・・・・・・・・120,121
目次囲み罫・・・・・・・・・・・・・・・・・・・・・・・138	和欧文間四分アキ・・・・・・・・・・・44
目次の作成・・・・・・・242	和文書体の追加・・・・・・・324
— ·- I	