

HCP Viewer(Ver 1.2.0) User's Manual

杉浦 学

gackt@crew.sfc.keio.ac.jp

慶應義塾大学 大岩研究室 (CreW Project)

2004.04.14

目次

1	準備編	2
1.1	実行環境	2
1.1.1	ハードウェアと OS の要件	2
1.1.2	JDK の準備	2
1.2	HCP Viewer のインストール	3
1.2.1	Windows の場合	3
1.2.2	その他の OS の場合	4
2	実践編	5
2.1	ソースの作成	5
2.1.1	インデント	5
2.1.2	コメント	6
2.1.3	コマンド	7
2.2	アプリケーションの概観と機能の概要	14
2.3	チャートの閲覧	15
2.3.1	ソースの読み込み	15
2.3.2	再読み込み	15
2.3.3	表示するモジュールの切り替え	15
2.3.4	デバッグコンソールの利用	15
2.4	チャートの印刷	16
2.5	清書したチャートの保存	16
3	応用編	17
3.1	コマンドラインツール	17
3.1.1	パスの設定	17
3.1.2	ツールの解説	17
3.2	コマンドのカスタマイズ	21
A	コマンド速見表	22

Over View

この文書は HCP Viewer(Ver 1.2.0) の使用説明書です.

アプリケーションの概要

HCP Viewer は、インデントしたテキストをコンパイルし、HCP チャートとして清書した結果を閲覧するためのアプリケーションです. 清書したチャートは、SVG, PNG, EPS 等のイメージファイルとして保存することができます (図 1).

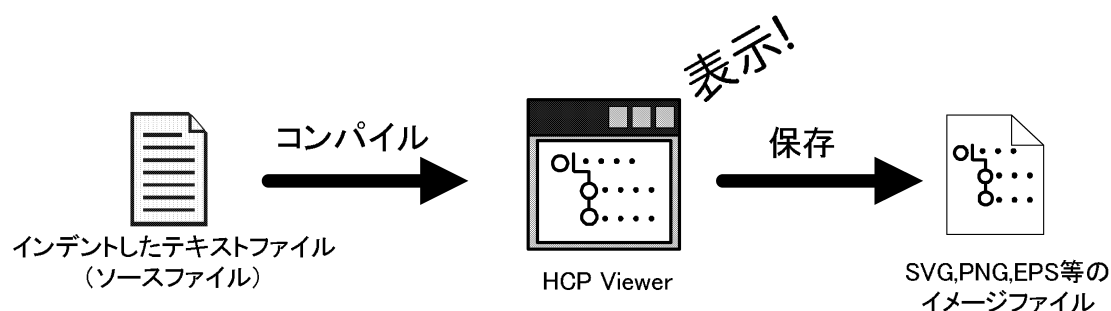


図 1: HCP Viewer を利用した HCP チャートの作成

インデントしたテキストをコンパイルするという方式を採用することで、GUI によるペイントソフトでありがちなマウスによる編集の手間を省き、スピーディーに HCP チャートを作成できるのが、HCP Viewer の最大の特徴です.

また、ソースに“ \forall ”で始まるコマンドを記述するだけで「振り分け処理」や「繰り返し処理」等の記号の描画を行うことができます. モジュール分割をしたチャートの作成にも対応しており、1つのソースに複数のモジュールを記述することができます.

現行の Ver1.2.0 ではデータの記述はサポートしていませんが、将来的にはデータの記述もサポートする予定です.

マニュアルの構成

このマニュアルは 3 章で構成されていて、各章の内容は以下のようになっています.

一般的な使用方法を想定していれば、2 章までの内容を読めば十分です. 3 章は HCP Viewer を使用して、(教材等の) 本格的なドキュメント作成を行う場合に有益な情報を掲載しました.

付録にはコマンドの一覧表をまとめたものを掲載しました.

- 第 1 章「準備編」 … HCP Viewer を使用するために必要な環境について
- 第 2 章「実践編」 … HCP Viewer を使用したチャート作成の基本について
- 第 3 章「応用編」 … HCP Viewer の発展的な機能とカスタマイズについて
- 付録 A … コマンドの速見表

1 準備編

1.1 実行環境

1.1.1 ハードウェアと OS の要件

HCP Viewer は、以下の要件を満たすハードウェアで実行することを推奨します。

- CPU … 1GHz 以上を推奨
- メモリ … 256MB 以上を推奨
- HD … 8MB 以上の空き容量がインストール時に必要

OS は以下に示す、JDK 1.4 以上が導入可能なものが必要です。

- OS … JDK 1.4 以上が動作する OS
 - Solaris (SPARC and x86)
 - Sun Java Desktop System
 - Windows 2000
 - Windows XP
 - Windows Server
 - Red Hat Enterprise Linux
 - Mac OS

1.1.2 JDK の準備

HCP Viewer を実行するためには、JDK 1.4 以上がインストールされていることが必要です。JDK のインストールが済んでいない場合は、ダウンロードとインストールを済ませておいてください。

- ダウンロード … <http://java.sun.com/j2ee/1.4/download.html>(Sun Microsystems)
- インストール方法 … <http://nautilus.crew.sfc.keio.ac.jp/lecture/2004objprog/jdk.install.html>

1.2 HCP Viewer のインストール

1.1節の環境が整ったら、<http://www.crew.sfc.keio.ac.jp/projects/hcpviewer/>から、各自の OS に適合した Installer をダウンロードし、インストールを行ってください。

1.2.1節にて、Windows ユーザを対象に、インストール手順の詳細を解説します。

1.2.1 Windows の場合

(1) Installer のダウンロード

<http://www.crew.sfc.keio.ac.jp/projects/hcpviewer/install.htm> から Windows 用の Installer をダウンロードします。

上記のページにある、「Start Installer for Windows...」のボタンをクリックするか、ページ中央にある、Available Installers から、Windows 用の Installer をダウンロードして任意の場所に保存してください。

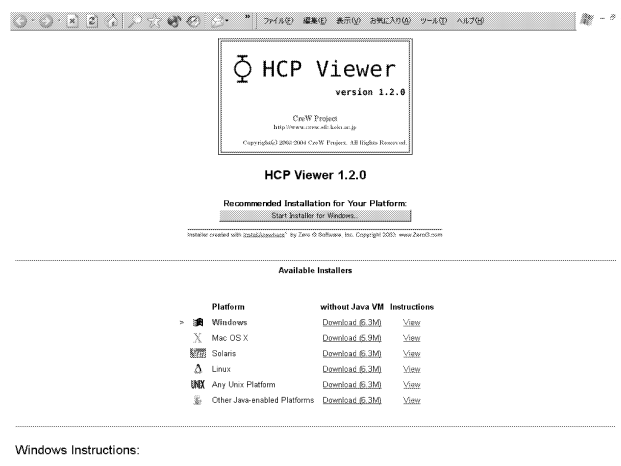


図 2: HCP Viewer の Web ページ

(2) インストール

ダウンロードした「HCP Viewer 1.2.0.exe」をダブルクリックして Installer を起動します。「Start Installer for Windows...」のボタンをクリックした場合は、自動的に Installer が起動します。

画面に表示されるウィザードに従って、インストールを行ってください。通常はデフォルトの設定を使用してください。その場合、全てのウィザードで「NEXT」ボタンをクリックするだけでインストールが完了します。

アンインストールは、「スタート」→「コントロールパネル」→「プログラムの追加と削除」から行ってください。アンインストールウィザードにより、アンインストールを行うことができます。

(3) パスの設定

第3章「応用編」に書かれているコマンドを実行するためには、パスの設定が必要になります。コマンドラインからのツール実行を行わない場合、この設定は必要ありません。

「スタート」→「コントロールパネル」→「システム（詳細設定タブをクリック）」→「環境変数」ボタンをクリックして、パスの設定を行ってください。

必要な設定は以下の2項目です。

1. HCP_VIEWER_HOME を新規作成する

- (a) 「変数名：HCP_VIEWER_HOME」
- (b) 「変数値：HCP Viewer をインストールしたディレクトリのパス（デフォルトは、"C:\Program Files\HCP Viewer 1.2.0"）」

2. PATH に%HCP_VIEWER_HOME%\bin を追加する

設定が終了したら、コマンドプロンプトを起動し、hcp というコマンドを入力します。HCP Viewer が起動すれば、設定が正常に終了したことになります。

(4) インストールの確認とアプリケーションの起動

インストールが終了すると、スタートメニューに HCP Viewer の起動アイコンが追加されます（「スタート」→「すべてのプログラム」→「HCP Viewer 1.2.0」→「HCP Viewer 1.2.0」）。

アイコンと共に表示されている、「HCP Viewer 1.2.0」をクリックすると、アプリケーションが起動します。

このマニュアルへのリンクも「マニュアル」として作成されていることを確認してください。

1.2.2 その他の OS の場合

お使いの OS に適合した Installer をダウンロードし、適宜インストールを行ってください。説明は割愛します。

2 実践編

2.1 ソースの作成

チャートのソースは、普段使い慣れているテキストエディタを使用するなどして作成してください。

保存するソースファイルの拡張子に制限はありませんが、「.hcp」などの分かりやすい名前にして保存¹することをお勧めします。

2.1.1 インデント

項目の前にタブやスペースを挿入してソースをインデントすることで、チャートの階層構造を表現することができます。インデントしたソース（リスト 1）と、その表示結果（図 3）を示します。

リスト 1: 段下げを行ったソース

```
処理
  前処理
  本処理
  後処理
```

```
タイトル：
モジュール：
作成者：
日付：
バージョン：
```

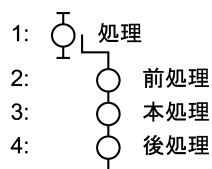


図 3: 段下げを行ったソースの表示結果

¹これは、第 3 章で紹介する、一括変換コマンドを利用する際に必要になる事柄です

2.1.2 コメント

コメントアウトしたいソースの一行目に「#」を挿入すると、その行全体がコメントとして扱われます。

リスト 2: コメントアウトを行ったソース

処理
前処理
本処理
後処理

タイトル：
モジュール：
作成者：
日付：
バージョン：

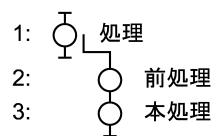


図 4: コメントアウトを行ったソースの表示結果

2.1.3 コマンド

「○」で表現される項目以外を表現したり、モジュール分割を行う場合はソースにコマンドを記述します。

コマンドは半角の“¥”の後にコマンド名を記述し、**半角のスペースを空けて**、内容を記述するという書式(リスト3)です。ただし、内容の記述が不要なコマンドも存在します。

リスト 3: コマンドの書式

¥コマンド名	内容
--------	----

コマンドは以下の3種類に大別することができます。次からそれぞれのコマンドの詳細を解説します。

- ヘッダーコマンド … チャートのヘッダーを記述するためのコマンド
- モジュールコマンド … チャートをモジュール分割するためのコマンド
- 通常コマンド … チャート中の記号を表現するためのコマンド

(1) ヘッダーコマンド

ソースにチャートのタイトル, 作成した人などのヘッダー情報を付けることができます.

チャートのソース本体の前に, 表 1に示すヘッダーコマンドを記述してください. 記述する順番は問いません. コマンドを記述しない場合, 内容が空のヘッダーが付与されます.

表 1: ヘッダーコマンド一覧

コマンド	内容
¥title	タイトル
¥author	作成者
¥date	作成日
¥version	バージョン情報

リスト 4: ヘッダーコマンドを記述したソース

```
¥title はじめての HCP チャート
¥author 杉浦 学
¥date 2004.04.12
¥version 1.0
```

```
処理
  前処理
  本処理
  後処理
```

```
タイトル:   はじめてのHCPチャート
モジュール:
作成者:     杉浦 学
日付:      2004.04.12
バージョン: 1.0
```

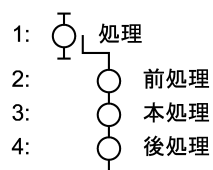


図 5: ヘッダーコマンドを記述したソースの表示結果

(2) モジュールコマンド

モジュールの開始位置に「¥module モジュール名」というコマンドを書くことで、1つのファイルで複数のモジュールを表現し、各モジュールをタブで分割して閲覧することができます。

このモジュール名は自動的にヘッダーにも記述されます。

なお、他のモジュールを呼び出す二重丸の項目を記述する場合、「¥mod 呼び出しモジュール名」で記述します。

表 2: モジュールコマンド一覧

コマンド	内容
¥module	モジュール分割位置の指定
¥mod	モジュール呼び出し項目

モジュールコマンドを使用したソースをリスト 5に、その表示結果を図 6と図 7に示します。

リスト 5: モジュールコマンドを記述したソース

¥title 自動販売機アプリケーション
¥author 杉浦 学
¥date 2004.04.12
¥version 1.0

¥module main
商品（缶ジュース）の販売と管理を行う
自動販売機アプリケーションの開始を知らせる
¥mod 商品を販売する
自動販売機アプリケーションの終了を知らせる

¥module sell
商品を販売する
購入したい商品の入力を受け取る
選択された商品を排出する

タイトル： 自動販売機アプリケーション
モジュール： main
作成者： 杉浦 学
日付： 2004.04.12
バージョン： 1.0





- 1:  商品（缶ジュース）の販売と管理を行う
- 2:  自動販売機アプリケーションの開始を知らせる
- 3:  商品を販売する
- 4:  自動販売機アプリケーションの終了を知らせる

図 6: モジュールコマンドを記述したソースを表示結果 (main モジュール)

タイトル： 自動販売機アプリケーション
モジュール： sell
作成者： 杉浦 学
日付： 2004.04.12
バージョン： 1.0






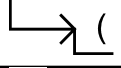



- 1:  商品を販売する
- 2:  購入したい商品の入力を受け取る
- 3:  選択された商品を排出する

図 7: モジュールコマンドを記述したソースを表示結果 (sell モジュール)

(3) 通常コマンド

チャート中の記号を表現するためのコマンドの一覧と、良く使用される組み合わせ例を示します。

表 3: 通常コマンド一覧

コマンド	内容	表示形態
¥repeat	繰り返し処理	
¥fork	振り分け処理	
¥branch	分岐条件	 ()
¥return	脱出	
¥ec	エラーチェック処理	
¥ex	エラー出口	

※「¥return 数字」で脱出する階層数を指定することもできます。

(i) 振り分け処理と条件の記述例

振り分け処理と条件を記述したソース例とその表示結果を示します。

リスト 6: 振り分け処理と条件を記述したソース

```
¥title 振り分け処理の記述例
¥author 杉浦 学
¥date 2004.04.12
¥version 1.0

¥module main
処理
  ¥fork 振り分け処理
    ¥branch 条件その 1
      条件その 1 が真だったときの処理
    ¥branch 条件その 2
      条件その 3 が真だったときの処理
    ¥branch 条件その 3
      条件その 2 が真だったときの処理
```

タイトル : 振り分け処理の記述例
モジュール : main
作成者 : 杉浦 学
日付 : 2004.04.12
バージョン : 1.0

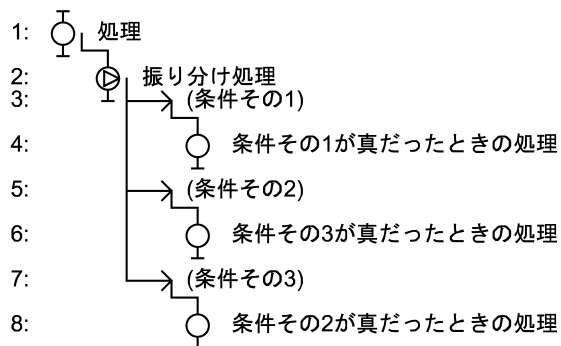


図 8: 振り分け処理と条件を記述したソースの表示結果

(ii) 繰り返し処理と脱出の記述例

繰り返し処理と脱出を記述したソース例とその表示結果を示します。

リスト 7: 繰り返し処理と脱出を記述したソース

```
¥title 繰り返し処理と脱出
¥author 杉浦 学
¥date 2004.04.12
¥version 1.0

¥module main
処理
  ¥repeat 繰り返し処理
    ¥fork 終了条件の確認
      ¥branch 終了条件
        ¥return
      処理
```

タイトル： 繰り返し処理と脱出
モジュール： main
作成者： 杉浦 学
日付： 2004.04.12
バージョン： 1.0

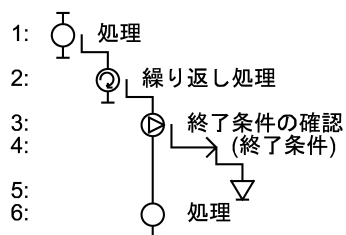


図 9: 繰り返し処理と脱出を記述したソースの表示結果

2.2 アプリケーションの概観と機能の概要

図 10 にアプリケーションの概観と、それぞれがどのような機能を果たすかを示します。



図 10: アプリケーションの概観

- メニュー
 - File … ソースファイルの (再) 読み込みと保存, 印刷, アプリケーションの終了を行います
 - Option … デバックコンソールの表示・非表示を切り替えます
 - Help … HCP Viewer のクレジットを表示します
- 表示モジュール切り替えタブ … クリックすると, 表示するモジュールを切り替えることができます
- チャート表示部分 … 清書されたチャートが表示される部分です
- デバックコンソール … ソースファイルのコンパイルエラーを表示します

2.3 チャートの閲覧

作成したチャートのソースを清書した結果を閲覧するための方法を解説します。

2.3.1 ソースの読み込み

作成したチャートのソースを読み込むには、「File」→「Open」から表示したいソースファイルを選択するか、起動しているアプリケーションにソースファイルをドラッグアンドドロップします。

2.3.2 再読み込み

チャートのソースを編集した場合、ソースファイルの再読み込みを行うことができます。

「File」→「Reload」を選択するか、ショートカットキーである F5 をタイプしてください。

2.3.3 表示するモジュールの切り替え

複数のモジュールからなるソースを読み込んだ場合、メニューバーの下方のタブをクリックすることで表示するモジュールを切り替えることができます。

2.3.4 デバッグコンソールの利用

表示するソースを読み込んだ際のコンパイルエラーを表示することができます。

「Option」→「Debug Console」にチェックを入れると、アプリケーションのウィンドウの下方に Debug Console を表示（デフォルトは OFF）することができます。チェックをはずすと非表示になります。

コンソールに表示されている内容をクリアしたい場合は、右にある消しゴムのボタンを押してください。

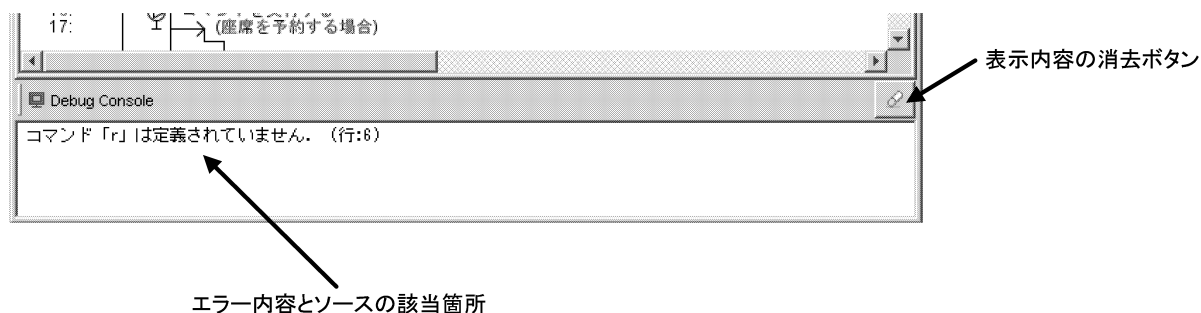


図 11: デバッグコンソール

2.4 チャートの印刷

清書されたチャートの表示結果を印刷したい場合は、「File」→「Print」を選択してください。印刷用のプリンタを選択するウィンドウが表示されます。

なお、印刷されるチャートは、タブで選択されて表示されているモジュールのみになります。

2.5 清書したチャートの保存

チャートの表示結果を SVG 形式のイメージとして保存することができます。

「File」→「Print」→「Export」→「SVG」を選択すると、保存するファイル名を入力するためのウィンドウが表示されます（デフォルトは「ファイル名_モジュール名.svg」になります）。

なお、保存されるチャートは、タブで選択されて表示されているモジュールのみになります。

3 応用編

3.1 コマンドラインツール

3.1.1 パスの設定

コマンドラインから各種のコマンドを実行するためには、パスの設定が必要になります。1.2.1節(3)を参照し、パスの設定を行ってください。

3.1.2 ツールの解説

(1) ファイル形式の変換

(i) SVG への変換

HCP チャートのソースを SVG 形式のイメージファイルに変換することができます。

```
| tosvg [変換したい hcp チャートのソースファイル名] |
```

図 12: SVG への変換

大量かつ複数のファイルを一括で変換したい場合は以下のコマンドを利用してください。

```
| tosvgall [変更履歴ログを取る場合は clog] |
```

図 13: SVG への一括変換

tosvgall は、実行されたディレクトリ以下の全ての.hcp という拡張子のファイルに対して tosvg を実行します。

clog という引数を付けて実行すると、実行ディレクトリに変換履歴のログを作成し、変換時にチェックを行います。同じディレクトリから clog を引数として与えて実行した場合、ソースファイルの変更があった場合のみに変換を行うことができます。

(ii) PNG への変換

SVG 形式に変換した HCP チャートを PNG 形式のイメージファイルに変換することができます。

```
| topng [変換したい svg のソースファイル名] |
```

図 14: PNG への変換コマンド

大量かつ複数のファイルを一括で変換したい場合は以下のコマンドを利用してください。

```
| topngall [変更履歴ログを取る場合は clog] |
```

図 15: PNG への一括変換コマンド

`topngall` は、実行されたディレクトリ以下の全ての `.svg` という拡張子のファイルに対して `topng` を実行します。

`clog` という引数を付けて実行すると、実行ディレクトリに変換履歴のログを作成し、変換時にチェックを行います。同じディレクトリから `clog` を引数として与えて実行した場合は、SVG の変更があった場合のみに変換を行うことができます。

(iii) EPS への変換

SVG 形式に変換した HCP チャートを EPS 形式のイメージファイルに変換することができます。

```
| toeps [変換したい svg のソースファイル名] |
```

図 16: EPS への変換コマンド

大量かつ複数のファイルを一括で変換したい場合は以下のコマンドを利用してください。

```
| toepsall [変更履歴ログを取る場合は clog] |
```

図 17: EPS への一括変換コマンド

このコマンドは、実行されたディレクトリ以下の全ての.svg という拡張子のファイルに対して toeps を実行します。

clog という引数を付けて実行すると、実行ディレクトリに変換履歴のログを作成し、変換時にチェックを行います。同じディレクトリから clog を引数として与えて実行した場合は、SVG の変更があった場合のみに変換を行うことができます。

(2) アプリケーションの起動

```
hcp [表示する HCP チャートのソースファイル名]
```

図 18: アプリケーションの起動コマンド

アプリケーションの起動を、コマンドラインから行うことが可能です。デバッグコンソールに表示される内容と同一のコンパイルエラーを、起動したコンソールに表示することができます。

(3) チャートの印刷

```
hprint [印刷する HCP チャートのソースファイル名] [プリンターの登録名]
```

図 19: チャートの印刷コマンド

HCP チャートの印刷を、コマンドラインから行うことが可能です。

3.2 コマンドのカスタマイズ

定義されているコマンドのリテラルを、自分の好みに変更することができます。

HCP Viewer がインストールされているディレクトリにある「command.properties」を編集してください。

リスト 8: コマンドの定義ファイル (command.properties)

```
### HCPViewer コマンド定義ファイル ###  
  
moduleCommandName=module  
defaultCommandName=node  
  
module=hcpviewer.compiler.analyzer.ModuleLineAnalyzer  
  
title=hcpviewer.compiler.analyzer.TitleLineAnalyzer  
author=hcpviewer.compiler.analyzer.AuthorLineAnalyzer  
date=hcpviewer.compiler.analyzer.DateLineAnalyzer  
version=hcpviewer.compiler.analyzer.VersionLineAnalyzer  
  
node=hcpviewer.compiler.analyzer.NodeLineAnalyzer  
repeat=hcpviewer.compiler.analyzer.RepeatLineAnalyzer  
fork=hcpviewer.compiler.analyzer.ForkLineAnalyzer  
branch=hcpviewer.compiler.analyzer.BranchLineAnalyzer  
ec=hcpviewer.compiler.analyzer.ErrorCheckLineAnalyzer  
mod=hcpviewer.compiler.analyzer.CallModuleLineAnalyzer  
ex=hcpviewer.compiler.analyzer.ExceptionalExitLineAnalyzer  
return=hcpviewer.compiler.analyzer.ReturnLineAnalyzer
```

例えば、`repeat` を `r` と書いてもコンパイルできるようにするには、

```
r=hcpviewer.compiler.analyzer.RepeatLineAnalyzer
```

という行を追加で記述し、アプリケーションを再起動してください。

A コマンド速見表



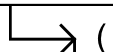


表 4: ヘッダーコマンド一覧

コマンド	内容
¥title	タイトル
¥author	作成者
¥date	作成日
¥version	バージョン情報

表 5: モジュールコマンド一覧

コマンド	内容
¥module	モジュール分割位置の指定
¥mod	モジュール呼び出し項目

表 6: 通常コマンド一覧

コマンド	内容	表示形態
¥repeat	繰り返し処理	
¥fork	振り分け処理	
¥branch	分岐条件	 ()
¥return	脱出	
¥ec	エラーチェック処理	
¥ex	エラー出口	