



共通デスクトップ環境 プログラ マーズ・ガイド (ヘルプ・システム 編)

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303
U.S.A. 650-960-1300

Part Number 806-2975-11
2001 年 5 月

Copyright 2001 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303-4900 U.S.A. All rights reserved.

本製品およびそれに関連する文書は著作権法により保護されており、その使用、複製、頒布および逆コンパイルを制限するライセンスのもとにおいて頒布されます。サン・マイクロシステムズ株式会社の書面による事前の許可なく、本製品および関連する文書のいかなる部分も、いかなる方法によっても複製することが禁じられます。

本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company, Ltd. が独占的にライセンスしている米国ならびに他の国における登録商標です。フォント技術を含む第三者のソフトウェアは、著作権により保護されており、提供者からライセンスを受けているものです。

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

本製品に含まれる HG 明朝 L と HG ゴシック B は、株式会社リコーがリコービイマジクス株式会社からライセンス供与されたタイプフェイスマスクをもとに作成されたものです。平成明朝体 W3 は、株式会社リコーが財団法人日本規格協会 文字フォント開発・普及センターからライセンス供与されたタイプフェイスマスクをもとに作成されたものです。また、HG 明朝 L と HG ゴシック B の補助漢字部分は、平成明朝体 W3 の補助漢字を使用しています。なお、フォントとして無断複製することは禁止されています。

Sun, Sun Microsystems, docs.sun.com, AnswerBook, AnswerBook2 は、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。

サンロゴマークおよび Solaris は、米国 Sun Microsystems 社の登録商標です。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャに基づくものです。

OPENLOOK, OpenBoot, JLE は、サン・マイクロシステムズ株式会社の登録商標です。

Wnn は、京都大学、株式会社アステック、オムロン株式会社で共同開発されたソフトウェアです。

Wnn6 は、オムロン株式会社で開発されたソフトウェアです。(Copyright OMRON Co., Ltd. 1999 All Rights Reserved.)

「ATOK」は、株式会社ジャストシステムの登録商標です。

「ATOK8」は株式会社ジャストシステムの著作物であり、「ATOK8」にかかる著作権その他の権利は、すべて株式会社ジャストシステムに帰属します。

「ATOK Server/ATOK12」は、株式会社ジャストシステムの著作物であり、「ATOK Server/ATOK12」にかかる著作権その他の権利は、株式会社ジャストシステムおよび各権利者に帰属します。

本製品に含まれる郵便番号辞書 (7 桁/5 桁) は郵政省が公開したデータを元に制作された物です (一部データの加工を行なっています)。

本製品に含まれるフェイスマーク辞書は、株式会社ビレッジセンターの許諾のもと、同社が発行する『インターネット・パソコン通信フェイスマークガイド '98』に添付のものを使用しています。© 1997 ビレッジセンター

Unicode は、Unicode, Inc. の商標です。

本書で参照されている製品やサービスに関しては、該当する会社または組織に直接お問い合わせください。

OPEN LOOK および Sun Graphical User Interface は、米国 Sun Microsystems 社が自社のユーザおよびライセンス実施権者向けに開発しました。米国 Sun Microsystems 社は、コンピュータ産業用のビジュアルまたはグラフィカル・ユーザインタフェースの概念の研究開発における米国 Xerox 社の先駆者としての成果を認めるものです。米国 Sun Microsystems 社は米国 Xerox 社から Xerox Graphical User Interface の非独占的ライセンスを取得しており、このライセンスは米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

DtComboBox ウィジェットと DtSpinBox ウィジェットのプログラムおよびドキュメントは、Interleaf, Inc. から提供されたものです。(© 1993 Interleaf, Inc.)

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行われないものとします。

本製品が、外国為替および外国貿易管理法 (外為法) に定められる戦略物資等 (貨物または役務) に該当する場合、本製品を輸出または日本国外へ持ち出す際には、サン・マイクロシステムズ株式会社の事前の書面による承諾を得ることのほか、外為法および関連法規に基づく輸出手続き、また場合によっては、米国商務省または米国所轄官庁の許可を得ることが必要です。

原典: *Common Desktop Environment: Help System Author's and Programmer's Guide*

Part No: 806-2913-10

Revision A



目次

はじめに	15
パートI 概要	
1. ヘルプ・システムの概要	23
ヘルプ・システムの概要	23
開発者のツールキット	24
オンライン・ヘルプの概要	25
ヘルプ情報モデル	25
アプリケーションとの統合	25
ヘルプの種類	26
ユーザがヘルプを獲得する方法	26
ヘルプ・ユーザ・インタフェース	27
ヘルプ・ウィンドウ	28
ハイパーリンク	29
ヘルプ・ナビゲーション	30
ヘルプ・メニュー	31
ヘルプ索引	31
ヘルプからの印刷	33
ヘルプ・トピックの構成	34
ヘルプ・トピック	35

	ヘルプ・ボリューム	35
	ヘルプ・ファミリー	35
	ヘルプ・ブラウザ・ボリューム	35
	設計者が行う作業	37
	オンライン・ヘルプの目的	37
	対象となるユーザ	37
	ヘルプへのアクセス方法	37
	ヘルプの表示方法の評価	38
	アプリケーション・プログラマとの協力	39
	設計者のワークフロー	39
	ヘルプタグによるヘルプ・トピックの記述	39
	形式ではなく構造を考える	41
	実行時のヘルプ・ファイルの作成	42
	ユーザを想定したヘルプのレビュー	42
	プログラマが行う作業	43
	ヘルプへのアクセス方法	43
	ヘルプの設計者との協力	43
	ヘルプのエントリ・ポイントの識別	44
	ヘルプ・ダイアログの作成と管理	44
	パッケージとヘルプの割り当て	44
	パートII 設計者が行う作業	
2.	ヘルプ・ボリュームの構成および記述	49
	ヘルプ・ボリュームのコンポーネント	49
	ホーム・トピック	50
	トピックとサブトピック	50
	エンティティ	50
	メタ情報	51
	用語集	52

一般的なマークアップのガイドライン	52
ソース・ファイル内のマークアップ	52
ヘルプタグ記号の表示	54
ヘルプ・ボリュームの例	54
ヘルプ・ソース・ファイル	55
volume.htg ファイルの作成	55
ファイル・マネージャのヘルプ・ファイル	56
関連項目	57
新規にヘルプ・ボリュームを記述する例	57
ソース・ディレクトリの作成	58
ビルド・ディレクトリの作成	59
マスタ・ヘルプタグ・ファイルの作成	60
helptag.opt ファイルの作成	60
実行時のヘルプ・ファイルの作成	61
ヘルプ・ボリュームの表示	62
トピック階層の作成	62
例	63
▼ ホーム・トピックを作成するには	64
▼ トピックを階層に追加するには	65
メタ情報トピックの作成	66
▼ メタ情報セクションを作成するには	66
非階層トピックの追加	67
▼ 非階層トピックを追加するには	68
トピックへのアクセス	69
ID 名に関する規則	69
▼ ID をトピックに追加するには	70
組み込み ID	70
▼ ID をトピック内の要素に追加するには	71

例 71

エンティティの使い方 72

エンティティの宣言に関する規則 72

▼ テキスト・エンティティを作成するには 72

▼ ファイル・エンティティを作成するには 73

3. ヘルプ・トピックの記述 77

ヘルプ・トピックの作成 77

例 78

トピック内に構造を作成する 78

▼ パラグラフを開始するには 79

▼ リストを作成するには 80

▼ ラベル付きリストを作成するには 81

▼ ラベル付きリストにヘッダを付けるには 82

▼ トピックにサブヘッダを付けるには 83

▼ コンピュータ・リストを表示するには 84

▼ 注、注意、警告を追加するには 86

インライン要素の入力 88

▼ 語句を強調表示するには 88

▼ 本のタイトルを入力するには 89

▼ Bold フォントで強調表示するには 89

▼ コンピュータ・リテラルを表示するには 89

▼ 変数を表示するには 90

ハイパーリンクの作成 91

<xref> 要素の使い方 92

▼ <xref> を使ってリンクを作成するには 92

リンク要素の使い方 93

▼ <link> を使ってリンクを作成するには 93

▼ 定義リンクを作成するには 96

- ▼ マニュアル・ページへのリンクを作成するには 97
- ▼ アプリケーション定義リンクを作成するには 98
- ▼ メタ情報トピックへリンクするには 98
- 実行リンクの制御 99
 - 実行ポリシーのデフォルト動作 100
 - 実行別名 100
 - ハイパーリンクでの実行別名の使用 101
 - DtNexecutionPolicy リソース 102
- グラフィックの表示 103
 - ▼ 図を作成するには 103
 - ▼ インライン・グラフィックを表示するには 105
 - ▼ グラフィックのまわりをテキストで囲むには 106
- 特殊文字の表示 108
 - ▼ 特殊文字を表示するには 108
- コメントと作成者のメモの取り込み 109
 - ▼ コメントを挿入するには 110
 - ▼ 作成者のメモを挿入するには 110
- 索引の作成 111
 - ▼ 索引エントリをマークするには 111
- 用語集の作成 112
 - ▼ 用語集の用語をマークするには 112
 - ▼ 用語集に用語を定義するには 114
- 4. ヘルプ・ボリュームの処理と表示 115**
 - 概要 115
 - ヘルプタグ・ソフトウェア 116
 - ボリュームの表示 116
 - 実行時のヘルプ・ファイルの作成 117
 - ▼ 実行時のヘルプ・ボリュームを作成するには 117

ヘルプタグの出力	118
▼ 手動で <code>dthelptag</code> コマンドを実行するには	119
▼ パーサ・エラーをレビューおよび修正するには	120
ヘルプ・ボリュームの表示	121
▼ ヘルプ・ボリュームを表示するには	121
ブラウザ・ボリュームへのヘルプの追加	123
ブラウザ・ボリューム	124
ヘルプ・ファミリー・ファイル	125
▼ ヘルプ・ファミリーを作成するには	125
▼ ブラウザ・ボリュームを表示するには	127
ヘルプ・トピックの印刷	128
ヘルプのテスト	129
ハイパーリンクの妥当性テスト	129
エントリ・ポイントの確認	130
索引エントリのチェック	130
グラフィックのテスト	130
パーサ・エラーのチェック	131
5. ヘルプタグ・マークアップのリファレンス	133
マークアップ要素の解説	133
<code><!-- ... --></code>	136
<code><abbrev></code>	137
<code><abstract></code>	138
<code><<annotation text>></code>	139
<code><book></code>	140
<code><caution></code>	141
<code><chapter></code>	142
<code><computer></code>	143
<code><copyright></code>	144

<dterm>	145
<emph>	146
<!entity>	147
<esc>	149
<ex>	150
<figure>	152
<glossary>	154
<graphic>	155
<head>	156
<helpvolume>	158
<hometopic>	159
<idx>	160
<image>	161
<item>	163
<keycap>	164
<lablist>	165
<lineno>	168
<link>	169
<list>	172
<location>	174
<memo>	176
<metainfo>	177
<newline>	178
<note>	179
<otherfront>	180
<otherhead>	180
<p>	182
<procedure>	184

<quote> 185
<rsect> 186
<s1>...<s9> 187
<sub> 189
<super> 189
<term> 190
<title> 192
<user> 193
<var> 194
<vex> 195
<warning> 196
<xref> 197

6. 特殊文字エンティティの概要 201

特殊文字 201

7. コマンドの要約 211

ヘルプ・システム・コマンド 211

ヘルプタグ・ファイルの処理 (dthelptag) 212

 コマンド形式 212

 コマンド・オプション 212

 パーサ・オプション 213

ヘルプ・トピックの表示 (dthelpview) 215

 コマンド形式 215

ブラウザ・ヘルプ・ボリュームの作成 (dthelpgen) 216

 コマンド形式 216

 オプション 217

8. ヘルプタグ文書型定義の読み方 219

文書型定義 219

 ヘルプタグ 1.3 DTD 219

	DTD コンポーネント	220
	要素宣言	220
	要素宣言のキーワード	222
	属性リスト宣言	223
	正規マークアップ	224
	正規マークアップの注意点	224
	要素の明確な階層	225
	ファイル・エンティティ宣言	227
	正規マークアップの処理	228
	パートIII プログラマが行う作業	
9.	ヘルプ・ダイアログ・ボックスの作成と管理	231
	ヘルプ・ダイアログ・ボックス	231
	標準 Xt 活用例	232
	一般ヘルプ・ダイアログ	232
	▼ 一般ヘルプ・ダイアログを作成するには	233
	簡易ヘルプ・ダイアログ	234
	▼ 簡易ヘルプ・ダイアログを作成するには	235
	アプリケーション・プログラム・インタフェースの概要	237
10.	ヘルプ要求への応答	239
	ヘルプの要求	239
	コンテキストの検知	240
	エントリ・ポイント	240
	ヘルプ・トピックの表示	240
	関連項目	241
	▼ ヘルプ・トピックを表示するには	242
	▼ テキスト文字列を表示するには	242
	▼ テキスト・ファイルを表示するには	243
	▼ マニュアル・ページを表示するには	244

- ヘルプ・キー ([F1] キー) を使用可能にする 245
- ▼ ヘルプ・コールバックを追加するには 245
 - クライアント・データの重要性 246
- [ヘルプ] メニューの提供 248
 - 関連項目 249
- アイテムヘルプ・モードのサポート 249
- ▼ アイテムヘルプのサポートを追加するには 250
- 11. ヘルプ・ダイアログのイベント処理 253
 - ヘルプ・ダイアログ・イベントのサポート 253
 - ハイパーリンク・イベント 253
 - ダイアログを終了する場合 254
 - 簡易ヘルプ・ボタン 254
 - ハイパーリンク・イベントへの応答 255
 - ▼ ハイパーリンク・コールバックを指定するには 255
 - ヘルプ・ダイアログの終了の検出 256
 - アプリケーション構成ボタンの使い方 257
 - ▼ アプリケーション構成ボタンを使用可能にするには 257
- 12. 「ヘルプの使い方」の作成 259
 - 「ヘルプの使い方」の作成 259
 - アプリケーション・ヘルプの場合 259
 - スタンドアロン・ヘルプの場合 260
 - 「ヘルプの使い方」の検索方法 260
 - アプリケーションの「ヘルプの使い方」へのアクセス 261
 - ▼ helpOnHelpVolume リソースを設定するには 261
 - ▼ [ヘルプの使い方] コマンドを指定するには 262
 - ▼ 「ヘルプの使い方」を表示するには 262
 - 独自の「ヘルプの使い方」ボリュームを記述する 264
 - 必須のエントリ・ポイント 264

▼ Help4Help ソース・ファイルをコピーするには	265
13. インストール・パッケージの準備	267
概要	267
オンライン・ヘルプの配布	268
インストール・パッケージの作成	268
実行時のヘルプ・ファイル	269
グラフィック・ファイル	270
ヘルプ・ファミリ・ファイル	270
アプリケーションとそのヘルプの登録	271
スタンドアロン・ヘルプ	272
アプリケーションの登録時に発生すること	272
ヘルプ・ボリュームの検索方法	272
製品準備チェックリスト	273
設計者の場合	273
製品のインテグレータの場合	274
プログラマの場合	274
パートIV 国際化対応	
14. 母国語のサポート	279
国際化されたオンライン・ヘルプ	279
国際化の要因	280
文字セットおよびマルチバイト文字	280
言語および地域名	282
ロケールおよび文字セット	286
ヘルプタグ・ソフトウェア	287
DtHelp メッセージ・カタログ	288
LANG 環境変数	288
helplang.ent ファイル	289
フォーマット・テーブル	289

はじめに

このマニュアルでは、共通デスクトップ環境アプリケーション・ソフトウェアのオンライン・ヘルプの開発方法について説明します。ヘルプ・トピックの作成方法や、オンライン・ヘルプを OSF/Motif アプリケーションに統合する方法についても説明します。

対象読者

このマニュアルの対象読者は次のとおりです。

- オンライン・ヘルプ情報を設計、作成、検証する設計者
- 完全なヘルプ機能を備えたソフトウェア・アプリケーションを作成することを目的としている開発者

内容の紹介

このマニュアルは4つのパートから構成されています。パート I 「概要」では、アプリケーション・ヘルプの設計段階に必要な設計者と開発者の協力と役割について述べます。パート II 「設計者が行う作業」では、オンライン・ヘルプを構成し記述する設計者に必要な情報を提供します。パート III 「プログラマが行う作業」では、ヘルプ・システムのアプリケーション・プログラマのツールキットを説明します。

パート IV 「国際化対応」では、多言語環境でオンライン・ヘルプを使用する場合に設計者とプログラマの両方に必要な情報を提供します。

このマニュアルは次の章から構成されています。

パート I 「概要」

第 1 章

オンライン・ヘルプの作成にあたり、設計者と開発者の協力的な役割について説明します。

パート II 「設計者が行う作業」

第 2 章

ヘルプ・ボリュームの構成に必要な要素を説明します。

第 3 章

ヘルプ・システム・マークアップ言語を概説し、さまざまなタイプの情報をフォーマットするのに使用する要素の例を示します。グラフィックを入れる方法とハイパーリンクの作成についても説明します。

第 4 章

1 つまたは複数のマークアップ・ファイルを処理して、オンライン表示を行うための実行時のファイルを生成する方法を説明します。

第 5 章

ヘルプタグ・マークアップ言語要素を、各要素の例と共にアルファベット順に説明します。

第 6 章

特殊文字をヘルプ・トピックのテキストに入力するのに使用する文字と、それに対応するエンティティ名のリストを示します。

第 7 章

端末エミュレータ・ウィンドウにコマンドを入力して、ヘルプ・ボリュームを処理および表示する方法を要約します。

第 8 章

ヘルプタグ DTD (文書型定義) の概要と、それを使用して完全に SGML (Standard Generalized Markup Language) に準拠するヘルプ・ファイルを作成する方法を説明します。

パート III 「プログラマが行う作業」

第 9 章

ヘルプ・ダイアログ・ウィジェットの概要と使い方を説明します。

第 10 章

さまざまなヘルプにアクセスするためにエントリ・ポイントを提供する方法を説明します。

第 11 章

コールバック構造を使用してハイパーリンク・イベントを処理する方法を示します。

第 12 章

ユーザにヘルプ・システムの使い方を示すヘルプ・モジュールの作成方法を説明します。

第 13 章

アプリケーションでオンライン・ヘルプを提供するために、インストール・パッケージに含まれているものについて説明します。

パート IV 「国際化対応」

第 14 章

ヘルプ・システムで使用する言語依存ファイルを示します。

「用語集」

このマニュアルに出てくる語句とその定義のリストです。

関連マニュアル

このマニュアルの利用にあたっては、次のマニュアルが参考になります。

- 『Solaris 共通デスクトップ環境 上級ユーザ及びシステム管理者ガイド』
- 『共通デスクトップ環境 プログラマーズ・ガイド(国際化対応編)』
- 『共通デスクトップ環境 スタイル・ガイド』
- 『Solaris 共通デスクトップ環境 ユーザーズ・ガイド』

SGML の技術的な説明については、次の文書を参照してください。

- 『SGML Handbook』 Charles F. Goldfarb、Oxford University Press (ISBN 0-19-853737-9)

Sun のマニュアルの注文方法

専門書を扱うインターネットの書店 Fatbrain.com から、米国 Sun Microsystems™, Inc. (以降、Sun™ とします) のマニュアルをご注文いただけます。

マニュアルのリストと注文方法については、<http://www1.fatbrain.com/documentation/sun> の Sun Documentation Center をご覧ください。

Sun のオンラインマニュアル

<http://docs.sun.com> では、Sun が提供しているオンラインマニュアルを参照することができます。マニュアルのタイトルや特定の主題などをキーワードとして、検索をおこなうこともできます。

表記上の規則

このマニュアルでは、次のような字体や記号を特別な意味を持つものとして使用します。

表 P-1 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 system%
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	system% su password:
<i>AaBbCc123</i>	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm <i>filename</i> と入力します。
『 』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。
[]	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第 5 章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	sun% grep `^#define \ XV_VERSION_STRING`

ただし AnswerBook2™ では、ユーザーが入力する文字と画面上のコンピュータ出力は区別して表示されません。

コード例は次のように表示されます。

■ C シェルプロンプト

```
system% command y|n [filename]
```

■ Bourne シェルおよび Korn シェルのプロンプト

```
system$ command y|n [filename]
```

■ スーパーユーザーのプロンプト

```
system# command y|n [filename]
```

[] は省略可能な項目を示します。上記の例は、*filename* は省略してもよいことを示しています。

| は区切り文字 (セパレータ) です。この文字で分割されている引数のうち 1 つだけを指定します。

キーボードのキー名は英文で、頭文字を大文字で示します (例: Shift キーを押します)。ただし、キーボードによっては Enter キーが Return キーの動作をします。

ダッシュ (-) は 2 つのキーを同時に押すことを示します。たとえば、Ctrl-D は Control キーを押したまま D キーを押すことを意味します。

一般規則

- このマニュアルでは、英語環境での画面イメージを使っています。このため、実際に日本語環境で表示される画面イメージとこのマニュアルで使っている画面イメージが異なる場合があります。本文中で画面イメージを説明する場合には、日本語のメニュー、ボタン名などの項目名と英語の項目名が、適宜併記されています。
- このマニュアルでは、「IA」という用語は、Intel 32 ビットのプロセッサアーキテクチャを意味します。これには、Pentium、Pentium Pro、Pentium II、Pentium II Xeon、Celeron、Pentium III、Pentium III Xeon の各プロセッサ、および AMD、Cyrix が提供する互換マイクロプロセッサチップが含まれます。

パート I 概要

ヘルプ・システムの概要

この章では、ヘルプ・システムの概要を述べ、ユーザ・インタフェースを簡単に説明します。ここではヘルプ情報の構成を示し、ヘルプ・モジュールの作成および処理の方法を概説し、アプリケーション・ヘルプの設計および作成における設計者と開発者の役割を説明します。

- 23ページの「ヘルプ・システムの概要」
- 25ページの「オンライン・ヘルプの概要」
- 25ページの「ヘルプ情報モデル」
- 27ページの「ヘルプ・ユーザ・インタフェース」
- 34ページの「ヘルプ・トピックの構成」
- 37ページの「設計者が行う作業」
- 39ページの「設計者のワークフロー」
- 43ページの「プログラマが行う作業」

ヘルプ・システムの概要

ヘルプ・システムは、アプリケーション・ソフトウェアのオンライン・ヘルプを開発するための一連のツールを提供します。これにより設計者は、グラフィックとテキストを書式化し、ハイパーリンクやアプリケーションとの通信などをオンライン・ヘルプに書き込むことができます。

また、オンライン・ヘルプをアプリケーションに統合するためのプログラムのツールキットも提供します。ヘルプ・システムのアプリケーション・プログラム・インタフェースは、2つの特殊なヘルプ・ダイアログと、オンライン・ヘルプ・モジュールの表示、ナビゲート、検索、および印刷に使用するサポート・ルーチンを提供します。

開発者のツールキット

ヘルプ・システム開発者のツールキットには、オンライン・ヘルプを記述、処理、表示するためのツールと、アプリケーション・プログラム用のライブラリが含まれています。

設計者用ツールキット

- ヘルプタグ・マークアップ言語

オンライン・ヘルプの構成と内容をマークするためにテキスト・ファイルで使用するタグのセット

- ヘルプタグ・ソフトウェア

作成したヘルプタグ・ファイルを変換して実行時のヘルプ・ファイルに書き込むためのソフトウェア・ツールのセット

- ヘルプ表示アプリケーション

オンライン・ヘルプを表示して、ユーザと同じようにそれを読んだり対話したりできるようにするビューア・プログラム

オンライン・ヘルプの作成および処理の詳細は、第2章を参照してください。

アプリケーション開発者用ツール

- DtHelp プログラム・ライブラリ

ヘルプ・ウィンドウをユーザのアプリケーションへ統合するためのAPI (アプリケーション・プログラム・インタフェース)

- プログラム例

ヘルプ・システムを OSF/Motif アプリケーションへ統合する方法を示す単純な例アプリケーション・プログラム用のライブラリについては、第9章から第13章で説明します。

オンライン・ヘルプの概要

コンピュータのハードウェアおよびソフトウェアの機能をすべて覚えることは不可能です。多くのコンピュータ・ユーザは、何度かヘルプを必要とすることがあるはずで

す。オンライン・ヘルプは印刷されたマニュアルと異なり、その場ですぐ参照できる点で優れています。最も重要なのは、ユーザの現在の「コンテキスト」の情報に適応できるようにするという点です。コンテキスト・ヘルプはユーザが行なっている作業に対応するヘルプを提供します。オンライン・ヘルプを開発するときは、ユーザがその場の状況に応じてあらゆるヘルプを必要とすることに注意してください。ユーザの疑問を予測することによって、論理的かつ直観的に応答するアプリケーション・ヘルプを設計できます。

ヘルプ情報モデル

オンライン・ヘルプには次の2つの形式があります。

- アプリケーション・ヘルプ

OSF/Motif アプリケーションに統合されたヘルプの一部として機能します。

- スタンドアロン・ヘルプ

あらゆるアプリケーション・ソフトウェアにも依存せずに、タスク、リファレンス、チュートリアル情報にオンライン・アクセスできるようにします。

アプリケーションのオンライン・ヘルプを開発する場合、情報をアプリケーションでだけアクセスできるように構成できます。あるいはスタンドアロン・ヘルプのように、アプリケーションの外からでも情報をブラウズできるように設定できます。

アプリケーションとの統合

このヘルプ機能により、アプリケーションとそのオンライン・ヘルプを高度に統合できます。ユーザの見地から見れば、ヘルプはアプリケーションの一部ということになります。これによって、ユーザとユーザが参照するヘルプとの間にあるアプリケーションの「距離」が縮められます。

アプリケーションの近くにあるということは、オンライン・ヘルプが使いやすく、要求に迅速に対応できるということです。

ヘルプの種類

オンライン・ヘルプは次の3種類に分類されます。

■ 自動ヘルプ

ヘルプが必要なとき、および表示する内容をアプリケーションが決定します。システム起動ヘルプとも呼ばれます。

■ 半自動ヘルプ

ヘルプが必要かどうかはユーザが判断しますが、表示する内容はシステムが決定します。このヘルプはユーザの操作または [F1] キーを押すなどのヘルプ要求によって起動されます。どの情報を表示するかはユーザの現在のコンテキストによって決められるため、こうしたシステムの応答はコンテキスト・ヘルプと呼ばれます。

■ 手動ヘルプ

[ヘルプ] メニューなどからユーザが特定の情報を要求します。

ユーザがヘルプを獲得する方法

ユーザはいくつかの方法でヘルプを要求できます。多くのアプリケーションが [ヘルプ] メニュー、[ヘルプ] キー、およびダイアログ・ボックスの [ヘルプ] ボタンを提供しています。

[ヘルプ] キー

多くのアプリケーションでユーザがまず使用するのは [ヘルプ] キーを押す方法です。最近では、多くのワークステーションおよびパーソナル・コンピュータにおいて、[F1] ファンクション・キーが事実上の標準ヘルプ・キーとなっています。

『共通デスクトップ環境 スタイル・ガイド』では、ヘルプ・キーとして [F1] キーを使用することと、OSF/Motif アプリケーションでのヘルプ・キーの使用を容易にするような内部処理を提供する OSF/Motif プログラマ・ツールキットを使用することを勧めています。

一部のコンピュータは、キーボード上に [ヘルプ] キーを備えています。

[ヘルプ] メニュー

[ヘルプ] メニューはヘルプ情報にアクセスするための一般的な方法です。OSF/Motif アプリケーションは [ヘルプ] メニューを提供しており、メニュー・バーの右端に表示されます。『共通デスクトップ環境 スタイル・ガイド』では、[ヘルプ] メニューに含めるコマンドについて推薦しています。



図 1-1 アプリケーションの [ヘルプ] メニュー

[ヘルプ] ボタン

ほとんどのダイアログ・ボックスに、ダイアログでヘルプを表示するための [ヘルプ] ボタンがあります。『共通デスクトップ環境 スタイル・ガイド』では、ダイアログ・ボックスで [ヘルプ] ボタンを選択することはそのダイアログ・ボックスを使用中に [ヘルプ] キーを押すのと同じであると述べています。ただし、複雑なダイアログ、すなわちダイアログ・ボックス内で個々のコントロールがヘルプを持つような場合は例外です。

ヘルプ・ユーザ・インタフェース

この節では、ヘルプ・システムが提供するグラフィカル・インタフェースの概要を説明します。ヘルプの機能の詳細は、『Solaris 共通デスクトップ環境 ユーザーズ・ガイド』を参照してください。対応するオンライン・ヘルプを表示するには、デスクトップのフロントパネルの [ヘルプ・ビューア] (127ページの「ブラウザ・ボ

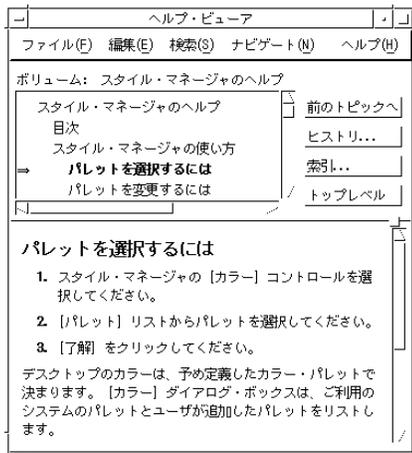
リウムを表示するには」を参照してください)を開きます。それから [デスクトップの紹介] および [ヘルプの使い方] を選択します。

アプリケーションの使用中、[ヘルプ] キーを押すかアプリケーションの [ヘルプ] メニューを選択することにより、ユーザはヘルプを要求できます。また、ヘルプ・システムを統合したアプリケーションをインストールできるので、デスクトップの [ヘルプ・ビューア] からそれぞれのヘルプ・モジュールにアクセスできます。これにより、複数のアプリケーションが提供するヘルプ情報をブラウズする場合でも、各アプリケーションを実行せずに行うことができます。

ヘルプ・ウィンドウ

ユーザがヘルプを要求すると、ヘルプ・システムはヘルプ・ウィンドウを表示します。ヘルプ・ウィンドウには、一般ヘルプと簡易ヘルプの2種類あります。一般ヘルプ・ウィンドウはメニュー・バー、トピック・ツリー、およびトピック表示領域から構成されます。トピック・ツリーはユーザが選択できるヘルプ・トピックの一覧です。ウィンドウの下部 (トピック表示領域) には選択されたトピックが表示されます。

簡易ヘルプ・ウィンドウはむだを省いたヘルプ・ウィンドウです。トピック表示領域と1つ以上のダイアログ・ボタンしか表示されません。このウィンドウは、定義などの短い組み込み情報の表示によく使用されます。



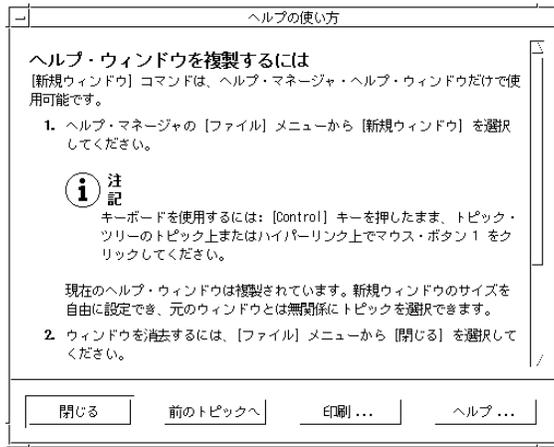


図 1-2 一般ヘルプ・ウィンドウと簡易ヘルプ・ウィンドウ

ハイパーリンク

ヘルプ・トピックには、関連するヘルプ情報へ「ジャンプ」するハイパーリンクがしばしば含まれています。テキストとグラフィックの両方をハイパーリンクとして使用できます。図 1-3 は、ハイパーリンクの識別に使用されるスタイルのフォーマットを示しています。

実線の下線または破線の下線は、ハイパーリンクである話または句を示します。実線の下線、つまり標準ハイパーリンクが最も一般的です。ハイパーリンクを選択すると、関連するトピックが表示されます。ハイパーリンク・トピックを現在のヘルプ・ウィンドウと新規ウィンドウのどちらに表示するかは、設計者が指定します。破線の下線は定義リンクを示します。これを選択すると、関連するトピックが簡易ヘルプ・ウィンドウに表示されます。グレーの隅の開いている枠 (破線または実線の下線) は、グラフィック・ハイパーリンクを示します。



図 1-3 グラフィック・ハイパーリンクとテキスト・ハイパーリンクの形式

ヘルプ・ナビゲーション

図 1-4 のトピック・ツリーは、現在のヘルプ・ボリュームのトピックのリストです。リスト最上部にある最初のトピックはホーム・トピックで、ヘルプ・ボリュームの先頭です。矢印(⇒)は現在のトピックを指し、ヘルプ・ボリュームにおけるユーザの位置を示します。

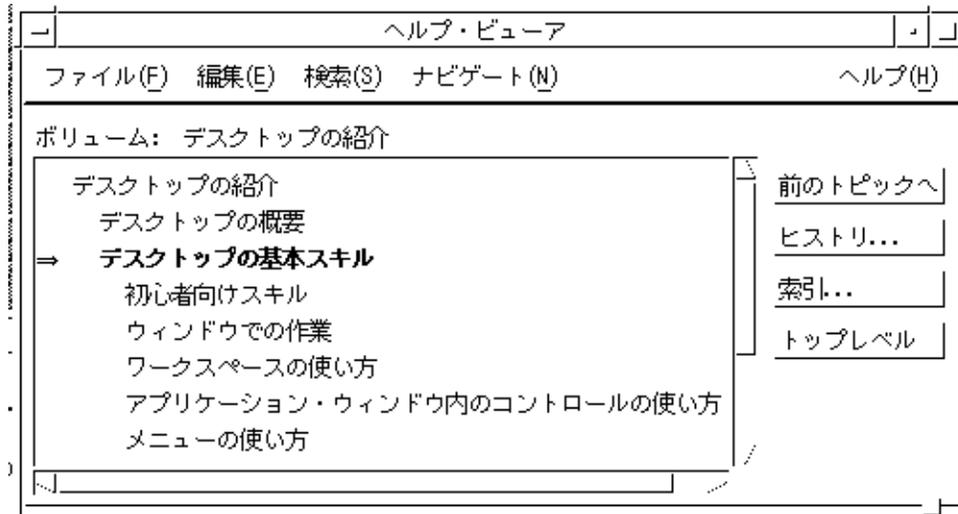


図 1-4 一般ヘルプ・ダイアログ・ボックスのトピック・ツリー

ヘルプ・トピックを表示するには、トピック・ツリーのタイトルか、トピック表示領域のハイパーリンクを選択します。表示をスクロールして任意のトピックを選択することにより、トピックの概要をブラウザできます。ナビゲーション・コマンドにより、前のトピックまたはヘルプ・ボリュームの先頭に戻ることができます。

ヘルプ・ナビゲーション・ボタン

一般ヘルプ・ダイアログには、[前のトピックへ]、[ヒストリ]、[索引]の3つのダイアログ・ボタンがあります。これらの機能は、メニュー項目の選択にも使用できます。

■ [前のトピックへ]

前のトピックに戻ります。先に参照したトピックに戻るには、希望のトピックが表示されるまで[前のトピックへ]ボタンを繰り返し押します。

■ [ヒストリ]

[ヒストリ] ダイアログ・ボックスを表示します。これはヘルプ・ボリュームとすでに参照したトピックをリストします。リストにある任意のトピックに戻るには、そのタイトルを選択します。

■ [索引]

[索引検索] ダイアログ・ボックスを表示します。これは設計者が索引エン트리としてマークした語句をすべてリストします。索引エントリを選択し、表示されたトピックのいずれかを選択すると、一般ヘルプ・ダイアログにそのトピックが表示されます。

デスクトップのフロントパネルから [ヘルプ・ビューア] を使用する場合、一般ヘルプ・ダイアログに [トップレベル] という追加ダイアログ・ボタンがあります。別のヘルプ・ボリュームを参照した後、このボタンを選択してデスクトップ・ブラウザ・ヘルプ・ボリュームのトップレベルに戻ることができます。

ヘルプ・メニュー

一般ヘルプ・ダイアログのメニュー・バーには、[ファイル]、[編集]、[検索]、[ナビゲート]、[ヘルプ] の5つのメニューが表示されます。[検索] メニューと [ナビゲート] メニューには、前記した [索引] ボタンとナビゲーション・ボタンで使用するコマンドがあります。また、[ナビゲート] メニューには、ヘルプ・ボリュームの先頭に戻るための [ホーム・トピック] コマンドがあります。その他のメニューは次の機能を提供します。

■ [ファイル] メニュー

ヘルプ・ウィンドウの複製、ヘルプ・トピックまたは現在のヘルプ・ボリュームの印刷、ヘルプ・ウィンドウのクローズ

■ [編集] メニュー

ヘルプ・ウィンドウから他のアプリケーションへのテキストのコピー

■ [ヘルプ] メニュー

ヘルプ・ダイアログの機能とその使い方に関するヘルプ情報の提供

ヘルプ索引

ヘルプ・ボリュームには、ユーザがサブジェクトのヘルプ・トピックを検索できるように重要な語句の索引が入っています。ユーザは現在のボリューム、選択したボリューム、またはシステムで使用できるすべてのヘルプ・ボリュームの索引をブ

ラウズまたは検索できます。*(アスタリスク) および?(クエスチョン・マーク) などの正規表現もトピックの検索に使用できます。対応するヘルプ・トピックを表示するには、索引エントリを選択します。

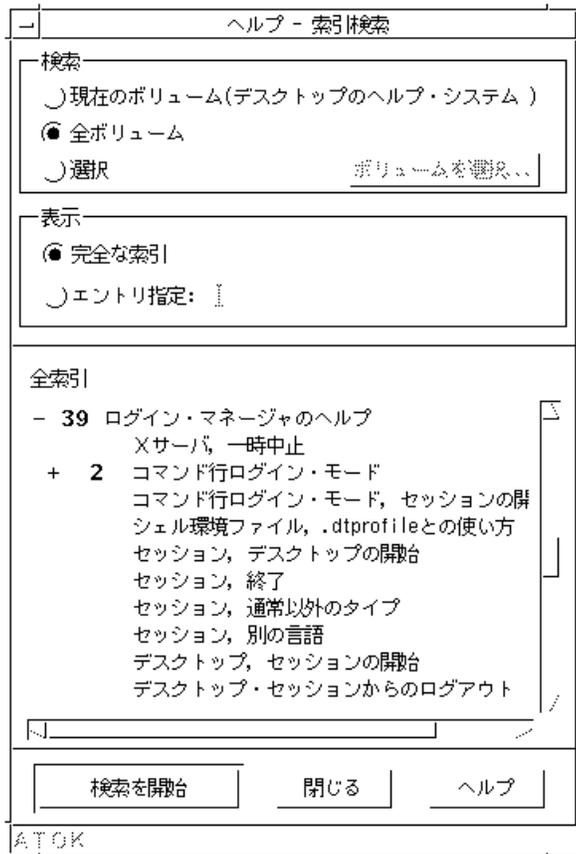


図 1-5 [索引検索] ダイアログ・ボックス

[ヘルプ索引] は一度に表示するには多すぎる場合があるので、表示されていない索引エントリがまだあったり、全部が表示されている場合もあります。+(プラス) または -(マイナス) の符号は表示されていない索引エントリがまだあるのかわかることを示します。マイナス符号はエントリのすべてが表示されていることを示し、プラス符号はまだ表示されていないエントリがあり、索引エントリを表示できることを示します。

図 1-6 の -9 は、9 個の索引エントリが表示されていることを意味しています。+2 の符号は表示されていないエントリがまだあることを示します。表示されていないエ

ントリを選択するとリストが展開され、+ 符号が - 符号に変わります。この図の最後の索引エントリはこのようにして表示されたエントリです。

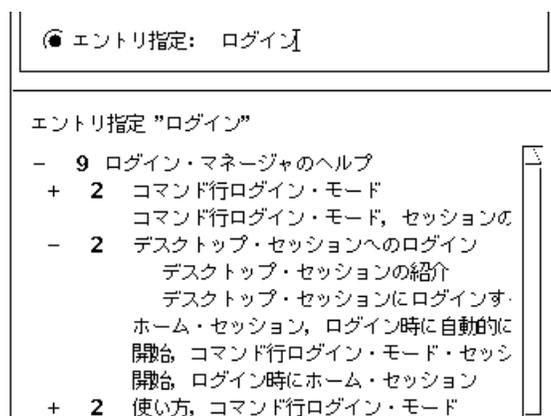


図 1-6 索引エントリ符号

ヘルプからの印刷

ユーザは、個々のヘルプ・トピック、目次、索引、あるいはヘルプ・ボリューム全体を印刷できます。出力はテキストのみ印刷されます。用紙サイズ、部数、出力先プリンタなどのオプションも、[印刷] ダイアログ・ボックスで設定できます。

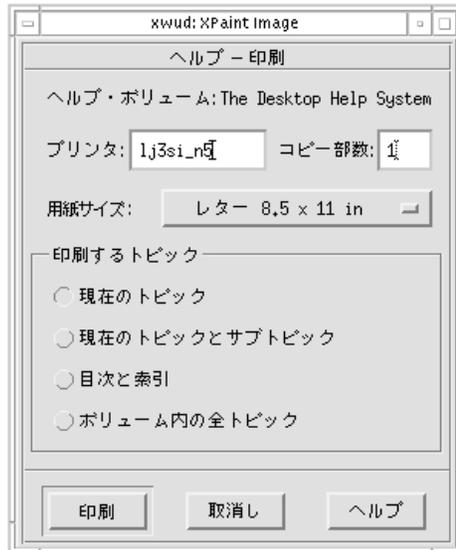


図 1-7 [印刷] ダイアログ・ボックス

ヘルプ・トピックの構成

設計者はヘルプ情報を論理フレームワークで構成します。例外はありますが、トピックの概略または階層を示す形式がほとんどです。図 1-8 に示すトピックの階層は、メイン・レベル、3つのセクション、およびそれに従属するトピックから構成されています。[ヘルプ] は階層から成る情報に最適化されていますが、必要に応じて任意の構成にできます。

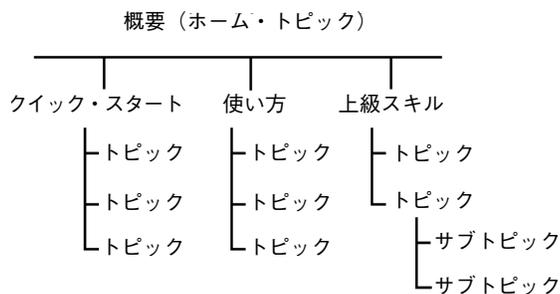


図 1-8 トピックの階層

ヘルプ・トピック

ヘルプ・トピックは固有の ID で識別される情報の単位です。ヘルプ・システムが提供する一連のタグは、ヘルプ・トピックをマークし、構造的なフレームワークを作成するために使用されます。ヘルプ・ビューアはヘルプ・システムの一部で、ヘルプ・トピックに直接アクセスし表示させることができます。

ヘルプ・ボリューム

ヘルプ・ボリュームは、アプリケーションまたは特定のサブジェクトを記述するトピックの集まりです。アプリケーションのヘルプを開発する場合、一般にはアプリケーションごとに1つのヘルプ・ボリュームがあります。ただし、複雑なアプリケーション、または関連するアプリケーションの集まりについては、複数のヘルプ・ボリュームを使用することもあります。

ヘルプ・ファミリ

ソフトウェアは、プロダクト・ファミリとして認識される一連の関連アプリケーションとして使用できます。たとえばオフィスで使用されるアプリケーションにはワードプロセッサ、スプレッドシート、描画プログラムなどがあります。各アプリケーションには独自のヘルプ・ボリュームがあるので、関連するヘルプ・ボリュームをヘルプ・ファミリとしてグループ化するようになっています。ヘルプ・ファミリには単一のヘルプ・ボリュームまたは複数のボリュームが入っています。

ヘルプ・ボリュームをヘルプ・ファミリにまとめるかどうかは任意です。フロントパネルの [ヘルプ・ビューア] などのヘルプ・ブラウザでヘルプをブラウズできるようにする場合にのみ必要です。

ヘルプ・ファミリ・ファイルとその使い方については、125ページの「ヘルプ・ファミリを作成するには」を参照してください。

ヘルプ・ブラウザ・ボリューム

デスクトップは、システムにインストールされたヘルプを表示するブラウザ・ボリュームと呼ばれる特殊なヘルプ・ボリュームを提供します。フロントパネルの [ヘルプ・ビューア] をクリックすると、図 1-9 に示すブラウザ・ボリュームが表示されます。

これは、ヘルプ・ファミリー (下線の付いた名前) とヘルプ・ファミリーのメンバである任意のボリュームを表示します。

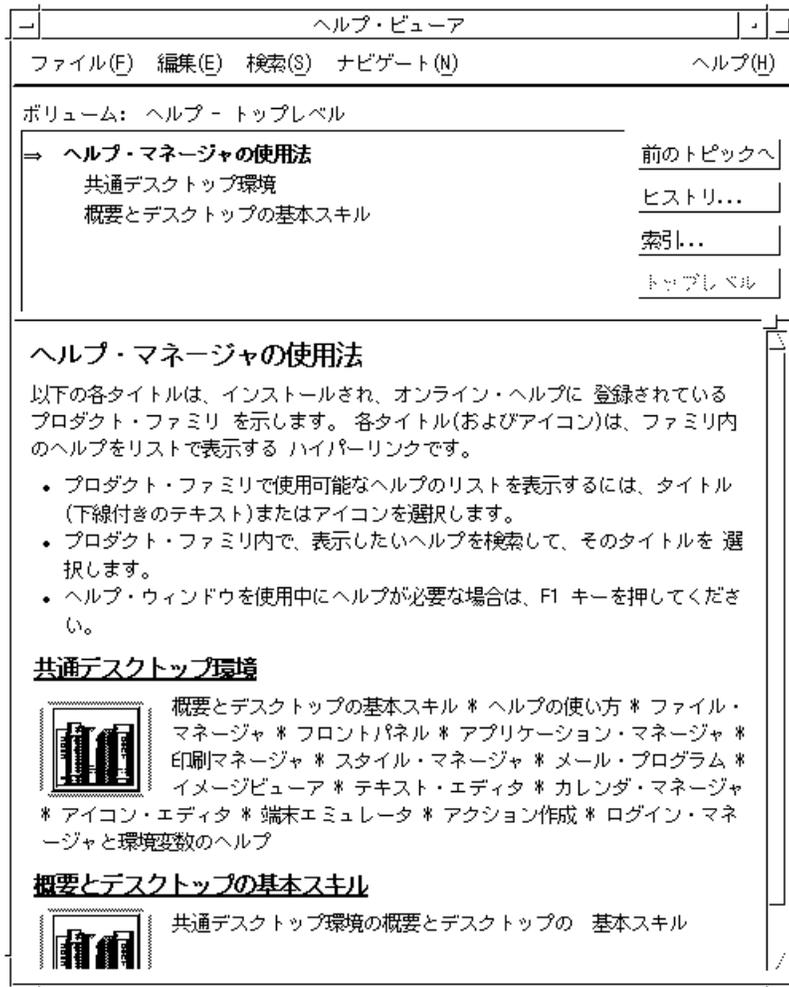


図 1-9 ヘルプ・ブラウザ・ボリューム

ブラウザ・ボリュームにより、アプリケーションを使用せずにアプリケーション固有のヘルプにアクセスできます。またスタンドアロン・ヘルプを書く場合は、ヘルプを使用するための唯一の手段がこのブラウザ・ボリュームを使用する方法です。ヘルプ・ボリュームが1つしかない場合は、[ヘルプ・ビューア]でブラウズできるヘルプ・ファミリーに属していなければなりません。

123ページの「ブラウザ・ボリュームへのヘルプの追加」では、ファミリー・ファイルの作成方法と、ブラウザ・ボリュームからヘルプ・ボリュームにアクセスできるようにするためにユーザーが行うべき作業について説明しています。

設計者が行う作業

オンライン・ヘルプの記述は印刷マニュアルを記述するのとは違うので、対象ユーザー、情報へのアクセスの方法、アプリケーションと情報との合致という点に留意することが重要です。

オンライン・ヘルプの目的

良質のオンライン・ヘルプの設計には、次の2つの重要な目的があります。

- ユーザーができるだけ早く確実に作業をこなせるようにすること。
- ユーザーがいずれヘルプなしでも作業できるように教育すること。

これらに目標を置けば、最適なヘルプは何か、必要な情報は何かを判断するのに役立ちます。

対象となるユーザー

何を書く場合でも、対象となるユーザーとそのユーザーが必要としている情報を理解することが大切です。特にオンライン・ヘルプでは、ユーザーが行おうとしている作業と、起こりうる問題について認識しておくことが重要です。

ヘルプへのアクセス方法

ユーザーを正しく認識すると同時に、ユーザーがどのようにヘルプにアクセスするかを理解することが重要です。

アプリケーションのヘルプ

アプリケーションのヘルプを記述する場合は、ブラウズできるトピックと、コンテキスト・ヘルプとしてアプリケーションから使用できるトピックをそれぞれ決める

ことが必要です。トピックをナビゲートできる場合は、トピック・ツリーまたはハイパーリンクを使用してトピックをブラウズできます。排他的なコンテキスト・ヘルプに指定されているトピックは、アプリケーションの特定のコンテキストからしかトピックを表示できないため、ブラウズできません。

アプリケーションのヘルプボリュームを登録するかどうかにも決める必要があります。登録されたヘルプ・ボリュームは他のアプリケーション (ヘルプ・ビューアなど) によっても表示でき、より広範囲な情報へのアクセスが可能になります。ヘルプ・ボリュームにあるトピックへの別のヘルプ・ボリュームからのハイパーリンクが含まれている場合は、そのヘルプ・ボリュームを登録してください。

アプリケーションのインストールと登録の詳細は、271ページの「アプリケーションとそのヘルプの登録」を参照してください。

スタンドアロン・ヘルプ・ボリューム

スタンドアロン・ヘルプ・ボリューム (アプリケーションに対応付けられていないヘルプ・ボリューム) を記述する場合は、別の方法で実行できます。

まず、記述したすべてのトピックをユーザが使用できるようにパスを設定しなければなりません。つまり、すべてのトピックが、少なくとも 1 つのハイパーリンクによってブラウズできなければなりません。また、ヘルプに対応付けられているアプリケーションがないため、ヘルプ・ボリュームの表示はヘルプ・ビューアに依存しなければなりません。

ヘルプの表示方法の評価

アプリケーションは種類が異なるヘルプを取り入れることができます。この場合、アプリケーションに最適なヘルプの種類がどれかを判断することが重要になります。たとえば同じヘルプ情報でも様々な方法で表示されることがあります。その選択には、主要な機能、チュートリアル、例、作業指示、ショートカット、トラブルシューティング、リファレンス情報、用語集、ハード・コピーまたはその他のオンライン・マニュアルがあります。ヘルプ・ボリュームは異なる表示を組み合わせることもあります。

アプリケーション・プログラマとの協力

アプリケーションのヘルプを記述する場合は、アプリケーション・プログラマと密に連絡をとるようにしてください。ヘルプ・システムをどの程度アプリケーションに統合するかは設計仕様によります。

アプリケーションとそのヘルプの結び付きが弱い場合には、アプリケーションが直接表示できるトピックはほんの一握りです。しかし簡単に実装できます。

反対に、アプリケーションのあらゆる状況に対応する特定のヘルプを提供できるアプリケーションもあります。これには多くの作業を必要としますが、それさえできれば、ユーザにとっては非常に便利です。

ヘルプの統合レベルを決定するのは、あなた自身とそのプロジェクト・チームです。

設計者のワークフロー

ヘルプの設計が終わったら、ヘルプ・ボリュームを作成するためのヘルプ・トピックを作成し、処理します。必要な作業は次のとおりです。

- ヘルプ・トピックの記述
- 実行時のヘルプ・ファイルの作成
- ヘルプ・ボリュームの表示

ヘルプタグによるヘルプ・トピックの記述

オンライン・ヘルプは通常のテキスト・ファイルで記述されます。情報の「要素」をマークアップするには特殊コード、すなわち「タグ」を使用します。タグは、ヘルプタグと呼ばれるマークアップ言語を形成します。

ヘルプタグ・マークアップ言語は、章、セクション、サブセクションなどハイレベルの要素と、段落、リスト、強調語などローレベルの要素とを判別する、要素の階層を定義します。

52ページの「一般的なマークアップのガイドライン」では、マークアップの使い方について簡単に示しています。各要素の詳細は、第5章を参照してください。

簡易マークアップ

タグセットは、実行時のヘルプ・ファイルを作成するため、簡易マークアップ、および正規マークアップの2種類の方法で使用できます。前者は、標準テキスト・エディタを使用して情報を「hand-tag」(タグ付け)する設計者にとって最適です。すなわち、実際のヘルプ・トピック・テキストに加えて、設計者がタグを入力する場合があります。手動によるタグ入力の影響を小さくするため、簡易マークアップではいくつかの省略方法を取り込んでいます。最初に、必要な開始タグと終了タグの数を減らします。また、マークアップやスタイルを変更するときに適宜使用する簡単な文字の組み合わせも提供されています。

正規マークアップ

正規マークアップとは、設計者が使用できる SGML (Standard Generalized Markup Language) によって、完全に SGML に準拠したヘルプ・トピックを作成することです。この場合はすべての要素の最初と最後にタグを付けなければなりません。また、各要素の構造にも明示的にタグを付けなければなりません。したがって、正規マークアップを使用するとタグの数が非常に多くなります。標準エディタで正規マークアップを入力することもできますが、構造化エディタを使用することをお勧めします。

構造化エディタ

構造化エディタと呼ばれる新しいツールが、SGML マークアップを効率よく作成するために使用できるようになりつつあります。典型的には、構造化エディタではコンテキスト・メニューを提供します。つまり、メニューに表示される要素が、ドキュメントのカーソルの位置に基づいて動的に変更されます。

たとえばリストを入力する場合は、メニューにはリスト要素のコンテキストで有効な要素だけが含まれます。この組み込み「知能」によって、設計者は簡単にマークできます。

設計者が、セクション、ヘッダ、リストなどの要素を選択すると、エディタはそれに対応する開始、終了、および任意の中間構造タグを生成します。たとえば設計者が章という要素を選択すると、この要素が必要とする中間タグをエディタが自動的に挿入します。編集者は単に章のタイトルを入力すればよいのです。生成されたタグの表示は任意で、設計者はタグを表示しないようにできます。

注・簡易マークアップ、正規マークアップのいずれの場合も、コンパイルおよび表示の時には同じオンライン情報を生成します。どちらの方法を使用するかは、ヘルプ情報および有効な設計ツールが使えるかどうかにか依存します。

正規マークアップの使い方

正規マークアップを使う場合は、最初にパート II 「設計者が行う作業」を読んでヘルプタグ要素の設定を理解してください。簡易マークアップと正規マークアップは同じタグセットを共有しますが、重要な相違点がいくつかあります。

第 8 章では、文書型定義 (DTD) の主要なコンポーネントを説明し、正規マークアップの方法を示します。完全なヘルプタグ文書型定義は、付録 A 「ヘルプタグ 1.3 DTD」にあります。

注・開発者キットには、ヘルプタグ文書型定義が入っています。ファイルは `/usr/dt/dthelp/dthelptag/dtd` ディレクトリにあり、`helptag.dtd` という名前です。

関連項目

- 第 2 章、第 3 章、第 4 章で、簡易マークアップの使い方を説明します。
- 第 5 章で、各タグをアルファベット順に詳しく説明します。
- 第 8 章で正規マークアップを説明します。
- `dthelptag.dtd(4)` のマニュアル・ページ

形式ではなく構造を考える

他のパブリッシング・システムに精通している場合は、情報を好きな形式にすることに慣れているはずですが、ヘルプタグを用いて設計するときは、形式ではなく構造と内容を考慮することが必要です。

記述するときは、ある種の情報をマークするのにタグを使います。マークする時には情報が何であるかを識別しますが、それをどうフォーマットすべきかは識別しません。

たとえば次のようなマークアップ行を含む本のタイトルを参照するとします。

ここでは、構造と内容が形式とは関係ないので、異なるシステムが同じ情報を違う形式で使用できるようになります。たとえば、ヘルプは本のタイトルをイタリックで表示します。しかしシステムによっては、イタリックは使えず、本のタイトルにはアンダーラインが引かれます。

実行時のヘルプ・ファイルの作成

実行時のヘルプ・ファイルを作成するには、記述したテキスト・ファイルをヘルプタグ・ソフトウェアで「コンパイル」しなければなりません。それはユーザがヘルプを要求したときにアクセスする実行時のヘルプ・ファイルとなります。実行時のファイルは、**Semantic Delivery Language (SDL)** 形式を使用します。この言語は、オンライン情報のデリバリ専用設計された **SGML** 文書型定義に基づいています。

ヘルプ・システムは、デスクトップの動作と、ヘルプ固有のファイルのデータ型を定義します。これによりヘルプ・ソース・ファイルのアイコンを選択し、ファイルを処理するメニュー・コマンドを選択する事により、デスクトップから簡単に実行時のファイルを作成できます。実行時のヘルプ・ファイルの識別には、`.sdl` という拡張子を使用します。処理中にエラーが発生すると、エラー・ファイル (`volume.err`) に通知されます。

実行時のヘルプ・ファイルの作成の詳細は、117ページの「実行時のヘルプ・ファイルの作成」を参照してください。デスクトップの動作とデータ型に関する一般的な情報については、『*Solaris* 共通デスクトップ環境 上級ユーザ及びシステム管理者ガイド』を参照してください。

ユーザを想定したヘルプのレビュー

設計中に、ヘルプを表示させて読者と同じように対話する必要が生じることがあります。その場合デスクトップからヘルプ・ボリュームを表示させるには、実行時のヘルプ・ボリューム (`volume.sdl`) のアイコンをダブルクリックします。あるいは、`dthelpview` コマンドを使用して任意のトピックを表示することもできます。第4章で両方の方法を説明します。

アプリケーション・ヘルプを書いている、ヘルプ・システムがアプリケーションに統合されている場合は、アプリケーションを実行してユーザと同じ方法でヘルプを要求することにより、ヘルプを表示できます。

プログラマが行う作業

プログラマは、ユーザがコンテキスト・ヘルプを要求したときに、その時のアプリケーションの動作に関連するヘルプ情報が表示されるようにアプリケーションにコードを追加します。

注 - `/usr/dt/share/examples/dthelp` ディレクトリには、`dthelpdemo` というプログラム例のソースコードが入っています。OSF/Motif アプリケーションへのヘルプ・ダイアログを追加する方法を示すものです。

ヘルプへのアクセス方法

ユーザに役立つ情報を提供するには、次のことを考慮することが必要です。

- 一般に起こりうる混乱状況は何か。このような状況で有効なヘルプを提供すればユーザは時間を浪費せずに済みます。
- 後でも先でもなく今、ユーザがヘルプを必要としているのはなぜか。実行するステップが複数あってユーザがその最初のステップにない場合は、そのステップに固有のヘルプ情報へ分岐しなければなりません。こうした方が、各ステップごとに同じヘルプ情報を表示するよりずっと役立ちます。ユーザが最初のステップにある場合は、最初のステップの詳細な情報も、全ステップの概要もわかるようにしなければなりません。
- ユーザはコンテキスト・ヘルプを要求しているのか、それとも単にヘルプ情報をブラウズしているだけなのか。コンテキスト・ヘルプを要求している場合は、現在行われている作業に関連した情報を提供しなければなりません。

ヘルプの設計者との協力

オンライン・ヘルプの設計者は各コンテキスト・トピックを表示する方法を知りたがっており、プログラマは各コンテキスト・トピックの内容を知りたがっているため、プログラマは設計者と密に連絡をとることが必要です。双方の協力がないと、情報は不適切で、曖昧で、誤りのあるものとなります。

こうした協力により、プログラマはアプリケーションを理解し、設計者は情報をユーザへどのように伝えたらよいかを理解できます。

ヘルプのエントリ・ポイントの識別

ヘルプのエントリ・ポイントを設定することにより、アプリケーションでオンライン・ヘルプが使えるようになります。エントリ・ポイントはアプリケーションで定義され、特定のヘルプ・トピックに対応付けられます。ユーザが [ヘルプ] キー、[ヘルプ] ボタン、[ヘルプ] メニューのいずれの方法によってヘルプを要求してもエントリ・ポイントが表示されます。たとえば [ヘルプ] ボタンを備えた印刷ダイアログ・ボックスを想定します。設計者はダイアログ・ボックスの内容を説明するヘルプ・トピックを記述し、トピックの ID を提供します。そこでユーザはコールバック・ルーチンを使用し、[ヘルプ] ボタンでヘルプ・トピックの ID を指定できます。

ヘルプ・ダイアログの作成と管理

ヘルプ・システムのアプリケーション・プログラム・インタフェースは、特に OSF/Motif アプリケーションで使用するよう設計されています。特にこのヘルプでは、次の 2 つの新しいウィジェット・クラス (およびこれら进行处理のために便利な機能を加えたもの) を提供することにより、OSF/Motif ウィジェット・セットを拡張しています。

■ 一般ヘルプ・ダイアログ

メニュー・バー、トピック・ツリー、ヘルプ・トピック表示領域を備えたヘルプ・ウィンドウを提供します。

■ 簡易ヘルプ・ダイアログ

トピック表示領域といくつかのダイアログ・ボタンを備えた簡素なヘルプ・ウィンドウを提供します。

アプリケーションでは上記のいずれか、または両方を使用できます。アプリケーションを (ヘルプ・ライブラリといっしょに) コンパイルすると、ヘルプ・ウィンドウがアプリケーションの一部になります。

第 9 章で、一般ヘルプ・ダイアログ・ボックスと簡易ヘルプ・ダイアログ・ボックスを説明します。

パッケージとヘルプの割り当て

プロダクト・パッケージには実行時のヘルプ・ファイル (*volume.sd1*) とグラフィック・ファイルが入っています。また、フロント・パネルの [ヘルプ・ビューア] でボリュームを表示できるようにするヘルプ・ファミリ・ファイルも作成できます。

ヘルプ・ボリュームが実行リンクを使用する場合、適切な実行リンク・リソースをアプリケーションのデフォルト・リソース・ファイルに入れるよう設計者と協力しなければなりません。第 13 章で、アプリケーションにどのヘルプ・ファイルが割り当てられるかを説明します。

パートII 設計者が行う作業

ヘルプ・ボリュームの構成および記述

この章では、ヘルプ・ソース・ファイルの構成とコンポーネントについて説明します。また、ヘルプ・ソース・ファイルを処理してオンライン・ヘルプ・ボリュームを作成する方法についての手順の例も示します。

- 49ページの「ヘルプ・ボリュームのコンポーネント」
- 52ページの「一般的なマークアップのガイドライン」
- 54ページの「ヘルプ・ボリュームの例」
- 55ページの「ヘルプ・ソース・ファイル」
- 56ページの「ファイル・マネージャのヘルプ・ファイル」
- 57ページの「新規にヘルプ・ボリュームを記述する例」
- 62ページの「トピック階層の作成」
- 66ページの「メタ情報トピックの作成」
- 67ページの「非階層トピックの追加」
- 69ページの「トピックへのアクセス」
- 72ページの「エンティティの使い方」

ヘルプ・ボリュームのコンポーネント

ヘルプ・ボリュームには、ホーム・トピック、トピック、サブトピック、エンティティの宣言、メタ情報、および用語集のおもに 6 種類のコンポーネントがあります。

ホーム・トピック

ホーム・トピックは、トピック階層の中でトップレベルにあるトピックです。最初のトピック、つまりヘルプ・ボリュームの先頭になります。その他のトピックはすべてサブトピックになります。トピック階層を複数のレベルの階層で構成することも可能です。しかし、階層内で迷わないようにするために、階層をできる限り浅くしておくべきです。

トピックとサブトピック

トピックとサブトピックは、ホーム・トピックの下に階層を形成します。通常、ホーム・トピックに続く第1レベルのトピックは、<chapter> 要素を使用して章に分割されます。章の内では、トピックはセクションに分割されます。<s1> セクションのサブトピックは <s2> と入力され、<s2> のサブトピックは <s3> のように入力されていきます。

章かセクションのどちらかの要素をホーム・トピックの後に続けることができます。階層を <chapter> と <s1> のどちらで始めてもその違いは表示からはわかりません。図 2-1 では、3つの章がある単純な階層を示しています。各章には第1レベルのセクションがいくつかあります。3番目の章には第2レベルのセクションが2つ追加されています。

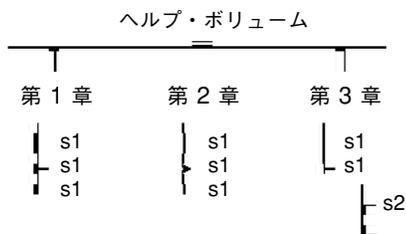


図 2-1 ヘルプ・ボリュームのトピック構成

エンティティ

設計者定義のエンティティには、文字列またはファイル名を指定します。エンティティ宣言は、エンティティ名とエンティティが表す文字列またはファイルを定義します。

エンティティは次の場合に有効です。

- テキストの共通の文字列を参照する

テキストが変更されたり、繰り返し入力したくないような場合に便利です。テキストを挿入したい場所では、エンティティ名を参照してください。

- 外部ファイルを参照する

グラフィック・ファイルにアクセスするにはエンティティが必要です。<figure> 要素と <graphic> 要素にエンティティ名を指定する必須パラメータが入っており、このエンティティ名によりグラフィック・イメージ・ファイルを参照します。

ヘルプ・ボリューム内の他のマークアップの前にすべてのエンティティの宣言を入力しなければなりません。定義済みのエンティティを入れるには、エンティティの参照を使用します。エンティティの参照は、ヘルプ・ボリューム内であればどこでも使用できます。ヘルプタグ・ソフトウェアでヘルプ・ボリュームを処理する場合、各エンティティの参照は、そのエンティティが指すテキストまたはファイルで置換されます。72ページの「エンティティの使い方」では、エンティティの定義方法と使い方について説明しています。

メタ情報

メタ情報はユーザの情報についての情報です。ボリュームのタイトル、著作権に関する記述、および要約などの情報が含まれています。要約は、ボリュームの内容についての簡単な説明です。

ヘルプ・システムは、メタ情報を使用して、ヘルプ・ボリュームのタイトルとその著作権情報を表示します。要約の記述は、フロントパネルのデスクトップ・ヘルプ・ビューアによって表示されます。ヘルプ・ボリュームを表示できる他のアプリケーションもまたこの情報を使用できます。

メタ情報には、通常のトピック階層の一部ではないヘルプ・トピックも含むことができます。メタ情報セクションにある非階層トピックには、リンクでアクセスします。

66ページの「メタ情報トピックの作成」は、メタ情報セクションの作成方法を示しています。

用語集

用語集には、ヘルプ・ボリュームで使用されている用語に対する定義が入っています。用語が <term> 要素を使用して入力される場合、自動的に定義リンクになり、選択されるとその用語に関する用語エントリを表示します。

一般的なマークアップのガイドライン

オンライン・ヘルプは通常のテキスト・ファイルで記述されています。特殊コード、またはタグを使用して、情報内の要素にマークを付けます。タグは、ヘルプタグと呼ばれるマークアップ言語を形成します。標準のテキスト・エディタを使用する場合、ヘルプタグ・マークアップが入力されます。または、エディタがマクロ・パッケージを提供している場合、タグはコマンド・キーを使用すると保存および挿入できます。ヘルプタグ・マークアップは構造化エディタを使用して生成することもできます (40ページの「正規マークアップ」を参照してください)。

ヘルプタグ・マークアップ言語は、章、セクション、サブセクションなどのハイレベル要素と、段落、リスト、および強調された語などのローレベル要素を定義する要素の階層を定義します。

ソース・ファイル内のマークアップ

ほとんどの要素のマークアップは、開始タグと終了タグで構成されます。開始タグは、三角括弧 (< と >) の間に要素名を入れて入力されます。終了タグも同様ですが、要素名の先頭に \ (バックスラッシュ) または ¥ (円記号) が付きます。

```
<element> ... text ... <\element>
```

たとえば、本のタイトルの始めと終わりをマークするには、次のようにマークアップを使用します。

```
<book>Geographical Survey of Northern Wisconsin<\book>
```

<book> は開始タグで、<\book> は終了タグになります。

簡易マークアップ

簡易マークアップは、標準テキスト・エディタを使用して手動でタグ付けする設計者向けに設計されたむだを省いたタグ・セットです。簡易マークアップにはいくつかの省略方法が用意されています。まず、終了タグの使用を最小化します。たとえば、章、節、または段落には終了タグを入力する必要はありません。さらに可能であれば、中間タグは自動的に想定されます。たとえば、章と節の要素により `<head>` タグを省略できます。したがって、ヘッダを入力するだけで済みます。

簡易マークアップは、文体の変更だけでなく多くのインライン要素へのマークアップも簡単にできます。テキストを区切るには、開始と終了のタグを入力するよりも、次のように縦線が使用されます。

```
<element| ... text ... |
```

たとえば、次の例は前に表示した `<book>` 要素の略式です。

```
<book| Geographical Survey of Northern Wisconsin|
```

要素にパラメータがある場合、要素とパラメータは次のように最初の縦線の前に入力します。

```
<element parameters| ... text ... |
```

開始と終了のタグを特殊な 2 文字のショートカットで置き換えるようなさらに略した形式をサポートしている要素もあります。たとえば、`<emph>` (強調表示) 要素の場合、その通常の形式は次のようになります。

```
<emph> ... text ... <\emph>
```

上記の形式は、次のように簡易形式を使用して入力することもできます。

```
!! ... text ... !!
```

第 3 章 では、簡易マークアップを紹介し、最も頻繁に使用される要素の例を示しています。第 5 章では、要素をアルファベット順に並べて、各要素について詳細に説明しています。

正規マークアップ

正規マークアップを使用する場合は、パート II で説明されている情報を理解しておく必要があります。また、正規マークアップの詳細は、第 8 章を参照してください。

ヘルプタグ記号の表示

時には、テキスト文字として < (左三角括弧)、\ (バックスラッシュ)、または & (アンド記号) を使用する必要がある場合もあります。その場合は、各文字の先頭にアンド記号を付けてください (&<、&\、または &)。

ヘルプ・ボリュームの例

次のマークアップは、入力に使用されるヘルプ・ボリュームとタグの重要な要素の例を示しています。この例では簡易マークアップを使用しますが、このマークアップは SGML 構造の中間タグを省略し、必須の終了タグの数を最小にします。インデントは、要素の階層関係を強調表示するために使用されます。したがって、記述するヘルプ・ファイルをインデントする必要はありません。

All entity declarations go here (before any other markup).

```
<helpvolume>
  <metainfo>
    <title> Volume Title
    <copyright>
      Copyright topic goes here ...
    <abstract>
      The abstract describing your help volume goes here.
      There may be other meta information topics.
      .
      .
  <\metainfo>

  <hometopic> Home Topic Title
    Help volume introduction goes here ...
    <s1> Title of First Topic Goes Here
      Body of the first topic goes here ...
    <s1> Title of Second Topic
      Body of the second topic goes here ...
      <s2> Title of Suptopic
        Body of the subtopic goes here ...
      .
      .
  <glossary>
    The body of the glossary, which contains term definitions, goes here ...
<\helpvolume>
```

ヘルプ・ソース・ファイル

オンライン・ヘルプは通常のテキスト・ファイルで記述されます。ヘルプタグ・ソフトウェアでこれらのファイル进行处理またはコンパイルし、ヘルプ・システムによって読み込むことができる実行時のヘルプ・ファイルを作成します。

volume.htg ファイルの作成

ヘルプタグには、*volume.htg* または *volume.ctg* という名前のプライマリ・コントロール・ファイルが必要です。*volume* には選択したボリューム名が入ります。ファイル拡張子によって、コントロール・ファイルが簡易マークアップ (.htg) か正規マークアップ (.ctg) のどちらを参照するかを指定します。

ボリューム名が固有のもので、意味があるものか確認してください。ボリューム名が一般的すぎる場合は、他のユーザが作成した別のボリュームと重複してしまう可能性があります。アプリケーションのヘルプを記述する場合、アプリケーションのクラス名を使用することをお勧めします。たとえば、アイコン・エディタのクラス名は *Dticon* なので、そのヘルプ・ボリューム名は *Dticon.htg* と名付けます。

複数のソース・ファイル

volume.htg ファイルには、エンティティの宣言とヘルプ・ボリュームを作成するファイルへのエンティティの参照が記述されています。ヘルプタグは入力として単一の *volume.htg* ファイルを必要としますが、作業を複数のソース・ファイルに分割できます。追加のファイルは、ファイル・エンティティを使用して *volume.htg* ファイルにまとめられます。ファイル・エンティティは、他のファイルへのポインタのようなものです。そのファイルは実際には、*volume.htg* ファイル内でエンティティの名前が表示されたところであればどこでも挿入されます。参照されるファイルもまた、別のファイルを参照するエンティティを含むことができます。(エンティティはテキスト文字列を参照する場合にも使用できます。)

例

ヘルプ・ボリュームには6つの章があり、各章は独立したファイルであるとし、ファイルには、HomeTopic、Metainfo、TOC、Tasks、Reference、およびGlossaryがあります。ヘルプ・ボリュームの *volume.htg* ファイルには、6つの

ファイルそれぞれに対するファイル・エントリと、ファイルの処理をヘルプタグ・ソフトウェアに指示するエンティティ参照リストが含まれています。

```
<!entity HomeTopic           FILE ``HomeTopic``>
<!entity MetaInformation     FILE ``Metainfo``>
<!entity TableOfContents    FILE ``TOC``>
<!entity Tasks               FILE ``Tasks``>
<!entity Reference           FILE ``Reference``>
<!entity Glossary            FILE ``Glossary``>
```

```
&HomeTopic;
&MetaInformation;
&TableOfContents;
&Tasks;
&Reference;
&Glossary;
```

実行中のヘルプタグの詳細は、117ページの「実行時のヘルプ・ボリュームを作成するには」で説明しています。

ファイル・マネージャのヘルプ・ファイル

ファイル・マネージャは、ヘルプ・ファイルをクエスチョン・マーク付きのファイル・アイコンで表現します。図 2-2には、ソース・ファイルが2つ (.ctg 拡張子と .htg 拡張子) と実行時のファイル (.sdl 拡張子) が1つあります。マークアップ・ファイルをダブルクリックすると、標準エディタが編集用のファイルを開きます。 .sdl ファイルをダブルクリックすると、ヘルプ・ビューアを使用して実行時のファイルが表示されます。



図 2-2 ファイル・マネージャによるヘルプ・ファイル表示

実行時のヘルプ・ボリュームを作成するには、ファイル・マネージャにある `.htg` または `.ctg` ファイルを最初を選択してください。次にファイル・マネージャの [選択] メニューから [コンパイル] を選択してください。

関連項目

- `dthelpaction(4)` のマニュアル・ページ

新規にヘルプ・ボリュームを記述する例

通常は `help` ディレクトリの中にヘルプ・ファイルを入れます。この例では、スタンドアロンのヘルプ・ボリュームの作成および表示方法について示しています。(スタンドアロンのボリュームの場合は、アプリケーションとの対話はありません。)

ヘルプ・ボリュームを作成および処理するには、次の手順に従ってください。

1. ヘルプ・ファイル用のソース・ディレクトリを作成します。
2. ビルド・ディレクトリを作成します。
3. マスタ・ヘルプタグ・ファイルを作成します。
4. `helptag.opt` ファイルを作成します。

5. 実行時のヘルプ・ファイルを作成します。

6. ヘルプ・ボリュームを表示します。

各タスクについては、次からの節で説明します。テキスト・ファイルで使用されているマークアップ言語については、この章で後述します。ヘルプタグのマークアップについては、第3章と第5章でさらに詳しく説明します。

ソース・ディレクトリの作成

1. ヘルプ・ファイルを作成および処理する helpfiles という名前のディレクトリを作成します。

2. 作成したばかりのディレクトリに Commands という名前のテキスト・ファイルを作成します。

この例では、すべての情報は単一のファイルに格納されます。通常、ヘルプを記述しているシステムやアプリケーションについて完全に説明するためには複数のファイルを使用します。

Commands ファイルには、テキストと要素のタグが入っています。〈 と 〉 (三角括弧) の内側にある要素のタグは、情報の構造を示します。

3. Commands ファイルに次のマークアップ・テキストを入力します。

```
<hometopic> Command Summary
                    <idx|commands|

Your &product; is capable of the following operations:
<list bullet>
  * <xref ChannelChange>
  * <xref VolumeUp>
  * <xref VolumeDown>
  * <xref VolumeMute>
<\list>

Choose one of the hyperlinks (underlined phrases)
to find out how to perform that operation.

<s1 id=ChannelChange> Changing the Channel
                    <idx|channel, changing|

Speak the command:
<ex> channel<\ex>
followed by a number from one to ninety nine.
<s1 id=VolumeUp> Turning Up the Volume
                  <idx|volume, changing|

Speak the command:
<ex> volume up<\ex>

For additional volume, speak the command:
```

```
<ex> more<\ex>
```

(See also <xref VolumeDown>)

```
<s1 id=VolumeDown> Turning Down the Volume  
                                <idx|volume, changing|
```

Speak the command:

```
<ex> volume down<\ex>
```

To further reduce the volume, speak the command:

```
<ex> more<\ex>
```

(See also <xref VolumeUp> and <xref VolumeMute>)

```
<s1 id=VolumeMute> Turning Off the Sound  
                                <idx|volume, changing|  
                                <idx|sound, on/off|
```

Speak the command:

```
<ex> sound off<\ex>
```

To restore the sound, speak the command:

```
<ex> sound on<\ex>
```

(See also <xref VolumeDown> and <xref VolumeUp>)

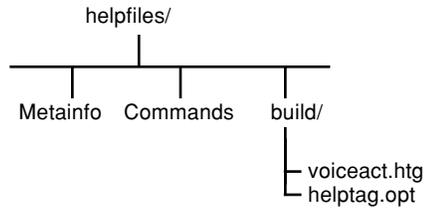
4. 情報にタイトルを付け、著作権情報と、オンライン・ヘルプに関する他の情報を提供するテキスト・ファイルを作成します。

この例では、Commands ファイルと同じディレクトリにある Metainfo という名前のファイルに次のテキストが格納されます。

```
<metainfo>  
  <title> Using the &product;  
  <copyright>  
    &copy; 1995 Voice Activation Company. All rights reserved.  
  <abstract> Help for Using the &product;.  
<\metainfo>
```

ビルド・ディレクトリの作成

helpfiles ディレクトリに build という名前のサブディレクトリを作成します。



マスタ・ヘルプタグ・ファイルの作成

1. build サブディレクトリに、名前が **volume**.htg のような形式になるテキスト・ファイルを作成します。この例では、ファイルは voiceact.htg です。
2. .htg ファイルで、Commands ファイルと Metainfo ファイルの名前とエンティティ名とを関連付けるエンティティを定義します。また、Commands ファイルと Metainfo ファイルで (直接的にまたは間接的に) 使用されているエンティティも定義します。最後に、それぞれのエンティティ名で Commands ファイルと Metainfo ファイルを参照します。

この例では、voiceact.htg ファイルの内容は次のようになります。<!--...--> 内のテキストはコメントですので、無視されます。

```

<!-- Declare an entity for each of the source text files. -->
<!entity MetaInformation      FILE   "Metainfo">
<!entity Commands            FILE   "Commands">

<!-- Define an entity that names the product and includes
      the trademark symbol (&tm;). -->

<!entity product  "VoAc&tm; Voice-Activated Remote Control">

<!-- Include the text files. -->

&MetaInformation;
&Commands;
  
```

helptag.opt ファイルの作成

1. build サブディレクトリに、helptag.opt という名前のファイルを作成し、次のテキストをそのファイルに格納します。この情報はヘルプタグのオプションを選択し、**FILE** エンティティの宣言で定義されたファイルをどこから検索すべきかを示します。

```
onerror=go
memo
search=./
search=../
```

onerror=go オプションは、エラーが発生した場合でもヘルプタグ・ソフトウェアが入力ファイル処理を続けるように指示します。パーサ・オプションの説明については、213ページの「パーサ・オプション」を参照してください。パーサ・オプションの一覧と詳細は、dthelptag(1)のマニュアル・ページを参照してください。

2. 端末エミュレータで次のコマンドを入力し、/usr/dt/bin ディレクトリが検索パスにあるか確認します。

```
echo $PATH
```

そのディレクトリがパスにない場合は、PATH 環境変数に追加します。追加方法がよくわからない場合は、システムのドキュメントを参照するかシステム管理者に問い合わせてください。

実行時のヘルプ・ファイルの作成

1. ファイル・マネージャを開き、build サブディレクトリに移動します。voiceact.htg ファイルのアイコンを選択し、ファイル・マネージャの【選択】メニューから【コンパイル】を選択します。

この手順により、実行時のバージョンのオンライン・ヘルプ・ボリューム(voiceact.sdl)を作成するヘルプタグ・ソフトウェアが実行されます。ステータスとエラー・メッセージが新規ファイルに格納され、volume.err のような形式の名前が付けられます。

2. voiceact.err ファイルを開き、エラーもなくファイルが処理されていることを確認します。エラーが発生している場合、必要に応じてテキスト・ファイルの編集および名前の変更を行なって修正します。

注 - 端末エミュレータで、ヘルプタグを手動で実行することもできます。

そのためには、次のコマンドを実行します。

```
dthelptag -verbose voiceact.htg
```

-verbose オプションは、画面にその経過状況を表示するようにヘルプタグに通知します。

ヘルプ・ボリュームの表示

build サブディレクトリから、voiceact.sdl ファイルのアイコンをダブルクリックします。

この手順により、デスクトップの [ヘルプ・ビューア] を使用してヘルプ・ボリュームが表示されます。情報をスクロールしたり、[ハイパーリンク] を選択して関連情報へジャンプできます。

注・端末エミュレータで、ヘルプ・ビューアを手動で実行することもできます。

そのためには、次のコマンドを実行します。新しいヘルプ・ボリュームが表示されます。

```
dthelpview -h voiceact.sdl
```

関連項目

- 第4章
- 第7章

トピック階層の作成

ヘルプ・ボリューム内のトピック階層は、ホーム・トピックから始まります。各ヘルプ・ボリュームにはホーム・トピックが必ず1つなければなりません。ホーム・トピックの下にある第1レベルのサブトピックは、<chapter> または <s1> と入力されます。

サブトピックのレベルを追加する場合は、<s2> や <s3> などのように入力します。ヘルプタグのマークアップ言語は、<s1> から <s9> までの9つのトピック・レベルをサポートしています。しかし、第3または第4レベルよりも下にある情報は、多くの読者にとって見つけにくいものになってしまいます。

ヘルプ・ボリュームが表示されたら、ヘルプ・ウィンドウはそのトピック・ツリーにトピックのリストを表示します。<chapter> か <s1...s9> のタグが入力され

ているトピックは、トピック・ツリーに自動的に表示されます。トピックをブラウザし表示するには、この方法が簡単です。

トピック内から他の関連情報を表示するには、ハイパーリンクを作成します。そのためには、それぞれのトピックに固有の ID を割り当て、ハイパーリンクがヘルプ情報内のどこでも特定の ID を参照できるようにします。

例

このサンプルに一致する階層を作成したいとします。

```
Tutorial for New Users
  Module 1: Getting Started
  Module 2: Creating Your First Report
  Module 3: Printing the Report
  Module 4: Saving Your Work and Quitting
Task Reference
  Starting and Stopping
    To Start the Program
    To Quit the Program
  Creating Reports
    To Create a Detailed Report
    To Create a Summary Report
Concepts for Advanced Users
  Using Report Hot Links
  Sharing Reports within a Workgroup
Reference
  Command Summary
  Report Attributes Summary
```

その場合、ヘルプ・ボリュームのマークアップの概要は次のようになります。(各トピックの本文や各トピックの ID は省略します。)

```
<hometopic> Welcome to Report Master
<chapter> Tutorial for New Users
  <s1> Module 1: Getting Started
  <s1> Module 2: Creating Your First Report
  <s1> Module 3: Printing the Report
  <s1> Module 4: Saving Your Work and Quitting
<chapter> Task Reference
  <s1> Starting and Stopping
    <s2> To Start the Program
    <s2> To Quit the Program
  <s1> Creating Reports
    <s2> To Create a Detailed report
    <s2> To Create a Summary report
<chapter> Concepts for Advanced Users
  <s1> Using Report Hot Links
  <s1> Sharing Reports within a Workgroup
```

```
<chapter> Reference
  <s1> Command Summary
  <s1> Report Attributes Summary
```

ここでインデントが使用されているのは、ヘルプ・ボリュームの構造をより見やすくするためです。ソース・ファイルでインデントする必要はありません。

関連項目

- 69ページの「トピックへのアクセス」では、トピックへの ID の割り当てについて説明しています。
- 91ページの「ハイパーリンクの作成」では、ハイパーリンクの作成方法について説明しています。

▼ ホーム・トピックを作成するには

- ◆ 次のように `<hometopic>` 要素を使用します。

```
<hometopic> Title
Body of topic.
```

メタ情報セクション (`<metainfo>`) を入れる場合、必ずその後ろにホーム・トピックを続けてください。

例

次に、タイトルと本文 1 文のみで構成されているホーム・トピックの例を示します。

```
<hometopic> Welcome to My Application
```

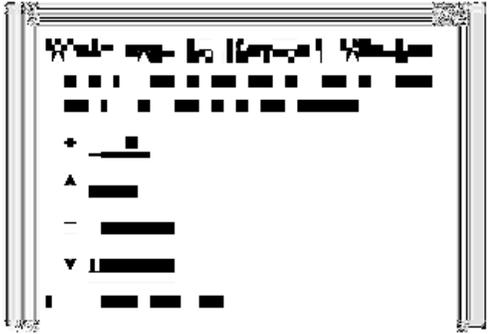
```
  Congratulations, you've entered
  the online help for My Application.
```

次の例は、4 つのサブトピックにハイパーリンクするホームトピックです。

```
<hometopic> Welcome to Report Master
```

```
  Welcome to the online help for Report Master.
  Choose one of the following hyperlinks:
  <list bullet>
  * <xref Tutorial>
  * <xref Tasks>
  * <xref Concepts>
  * <xref Reference>
  <\list>
  If you need help, press F1.
```

上記のマークアップの出力は次のとおりです。



▼ トピックを階層に追加するには

- ◆ 別のトピックを同じレベルに追加するには、同じ要素を繰り返します。
または、サブトピック (階層内でさらに 1 レベル下のトピック) を追加するには、前のトピックよりも 1 レベル下にある要素を使用します。

例

現在のトピックが <s1> である場合、<s2> を使用してサブトピックを入力します。

```
<s1 id=getting-started> Getting Started
```

```
<s2 id=starting-the-program> Starting the Program  
Here's the body of the first subtopic.
```

```
<s2 id=stopping-the-program> Stopping the Program  
Here's the body of the second subtopic.
```

2 番目の <s2> も <s1> のサブトピックです。

注 - 親子兄弟メタファは、階層内のトピック間の関係を記述するのに使用されることもあります。前の例では、<s1> トピックは両方の <s2> (「子」トピック) の「親」です。2 つの <s2> はお互いに「兄弟」になります。3 つのすべてのトピックは、ホーム・トピックの「子孫」です。

メタ情報トピックの作成

メタ情報セクションには、本の中に著作権に関するページがあるのと同様に、ボリューム・タイトル、著作権、商標、およびその他の注意事項などの情報が含まれています。

メタ情報セクションの2番目の使用法は、通常のトピック階層の一部ではないヘルプ・トピックを入力することです。これらの非階層トピックは、簡易ヘルプ・ダイアログ・ボックスにあるトピックをポップアップするカスタム定義のリンクを作成するのに有用です。

▼ メタ情報セクションを作成するには

1. 次のように `<metainfo>` タグを入力してセクションを始め、必須の副要素である `<title>` と `<copyright>` を入力します。

```
<metainfo>

<title> Volume Title Here

<copyright>
Body of copyright topic here..
.
.
```

2. 次のようにオプションの要素のどれかを入力します。

```
<abstract>
Body of the abstract topic here.
Do not use any HelpTag markup within the abstract!
```

3. 終了タグ `<\metainfo>` を入力してセクションを終了します。

```
.
.
.
<\metainfo>
```

注 - メタ情報セクションにある要素の中には、トピックのヘッダの前に `<head>` タグが必要なものもあります。

<abstract> セクションを使用することをお勧めします。ヘルプ・ボリュームにアクセスするアプリケーションはこの情報を使用してボリュームの簡単な説明を行います。その抜粋は単純なテキスト・ウィンドウ (複数のフォントやグラフィック・フォーマットはサポートしていない) に表示される可能性があるため、ヘルプタグのマークアップを取り込まないようにしてください。

例

次の例は、一般的なメタ情報セクションです。

```
<metainfo>
  <title> Report Master, Version 1.0

  <copyright>
    <otherhead> Report Master

    <image>
      Version 1.0
      &copy; Copyright Reports Incorporated 1995
      All rights reserved.
    <\image>

  <abstract>
    This is the online help for the mythical Report Master
    application. This help includes a self-guided tutorial,
    a summary of common tasks, general concepts, and quick
    reference summaries.

<\metainfo>
```

<image> 要素は、設計者が入力した改行を維持するために使用されます。© は著作権のシンボルを挿入します。

関連項目

- 第3章
- 98ページの「メタ情報トピックへリンクするには」

非階層トピックの追加

<chapter> または <s1...s9> の要素タグが入力されているトピックは、自動的にトピック・ツリーに表示されます。タイトルがトピック・ツリーで選択されると、対応するヘルプ・トピックが一般ヘルプ・ダイアログ・ボックスに表示されます。

しかし、すでに作成されているトピック階層からは独立しているトピックを作成および表示したい場合もあります。たとえば、別の簡易ヘルプ・ウィンドウにトピックを表示したい場合などです。

▼ 非階層トピックを追加するには

- ◆ 次のように `<otherfront>` 要素を使用して、メタ情報セクションの最後の直前にトピックを追加します。

```
<otherfront id=id><head> Topic Title
Body of topic.
```

ID パラメータと `<head>` タグは必須です。

希望の数の `<otherfront>` トピックを追加できます。どんな順番でもかまいませんが、`<metainfo> ... <\metainfo>` セクションの中の最後のトピックでなければなりません。

例

この部分的なヘルプ・ボリュームは、一般トピックがどのようにメタ情報セクションに追加されるかを示しています。トピックのタイトルは「Pop-up!」で、ID は `my-popup-topic` です。

```
<metainfo>

<title> My Help
<copyright>
  This is My Help, Version 1.0.  &copy; 1995.
  .
  .
  .

<otherfront id=my-popup-topic> <head> Pop-up!

  This is a pop-up topic, displayed via a definition link
  somewhere in my help volume.
<\metainfo>

<hometopic> Welcome to My Help
  .
  .
  .
```

ヘルプ・ボリューム内の他のトピックのいくつかには、このトピックを表示するための定義リンクがある場合があります。

リンクは次のように定義されます。

Here's a sample of a pop-up `<link my-popup-topic Definition> definition link<\link>`.

「`definition link`」という語はアクティブなハイパーリンクになり、破線の下線が付いて表示されます。そのリンクを選択すると、簡易ヘルプ・ダイアログ・ボックスの「Pop-up!」トピックが表示されます。

関連項目

- 91ページの「ハイパーリンクの作成」
- 180ページの「`<otherfront>`」

トピックへのアクセス

ヘルプタグ言語の多くの要素は ID 属性をサポートしています。ID は、トピックとトピック内の要素を識別するために内部的に使用される独自の名前です。ID は一度しか定義されませんが、複数のハイパーリンクと相互参照は同じ ID を参照できます。ID はユーザには見えません。

アプリケーションのヘルプを記述している場合、ユーザがヘルプを要求したときに表示される特定のトピックを識別するために、アプリケーションが ID を使用することもあります。たとえば、アプリケーションのメニューを説明するいくつかのトピックを記述する場合などです。トピックに割り当てる ID は、アプリケーションの開発者によって使用されます。アプリケーション・コード内に同一の ID を定義することにより、開発者は特定のトピックを統合できます。これによりアプリケーションは、特定のメニューに関してヘルプが要求されたときに正しいトピックへアクセスし、表示できます。

ID 名に関する規則

- ID 文字列は、英字 (A から Z と a から z)、数字 (0 から 9)、マイナス (-) 記号から成り、必ず英字で始まります。
- 設計者定義 ID は `_` (下線文字) を使用できません。ヘルプタグ要素のいくつかに組み込まれている ID のために予約されています。

- 大文字でも小文字でも意味は変わりませんが、読みやすいように使い分けてください。
- ID 文字列の長さは、64 文字以内です。
- 単一のヘルプ・ボリューム内の各 ID は、固有のものでなければなりません。

▼ ID をトピックに追加するには

- ◆ 次のように要素の `id` パラメータを使用します。

`<element id=id> ...`

新しいトピックを開始し、設計者定義 ID をサポートしている要素は次のとおりです。

- `<chapter id=id>`
- `<otherfront id=id>`
- `<rsect id=id>`
- `<s1 id=id>`
- `<s2 id=id> ... <s9 id=id>`

組み込み ID

組み込み ID を持っている要素は数個ありますが、設計者定義 ID をサポートしていません。次の要素のそれぞれも新しいトピックを開始しますが、これらの要素は定義済み ID (括弧内のもの) を持っています。

<code><abstract></code>	(<code>_abstract</code>)
<code><copyright></code>	(<code>_copyright</code>)
<code><glossary></code>	(<code>_glossary</code>)
<code><hometopic></code>	(<code>_hometopic</code>)
<code><title></code>	(<code>_title</code>)

▼ ID をトピック内の要素に追加するには

- ◆ 要素が設計者定義 ID をサポートしている場合、次のように要素の id パラメータを使用します。

```
<element id=id> ...
```

ID 属性をサポートしている (トピック内の) 要素は、次のとおりです。

- <figure id=id>
- <graphic id=id>
- <image id=id>
- <location id=id>
- <p id=id>

または、次のように <location> 要素を使用して、トピック内の任意の場所に ID を設定します。

```
<location id=id> text <\location>
```

text は ID を追加したい語句です。<\location> 終了タグは必須です。位置 ID に対するリンクを起動すると、ヘルプ・ビューアは ID が入っているトピックを表示し、ウィンドウを ID の位置までスクロールします。

例

ID を図に追加する場合、タイトルがなければなりません。図の ID に対して相互参照が行われる場合、タイトルが必要です。その場合、タイトルは図へのハイパーリンクになります。

次は、ID `my-big-picture` の図をマークアップする例です。

```
<figure id=my-big-picture entity=big-picture-TIFF>  
Here's My Figure  
<\figure>
```

次の例は、「easier than ever」という句に ID `easy-spot` が割り当てられているパラグラフです。

```
Getting help is <location id=easy-spot> easier than ever<\location>.
```

エンティティの使い方

エンティティには文字列やファイルの内容を指定できます。エンティティの宣言は、エンティティ名と特定の文字列やファイル名とを関連付けることによりエンティティを定義します。エンティティの参照は、`dthelptag` コマンドでヘルプ・ボリュームを処理すると文字列やファイルの内容に置き換えられます。

エンティティは次の作業を実行できます。

- テキストの共通の文字列を参照する。

テキストを変更したり、繰り返し入力したくないような場合に便利です。テキストを挿入したい場所では、エンティティ名を参照してください。

- 外部ファイルを参照する。

グラフィック・ファイルにアクセスするにはエンティティが必要です。`<figure>` 要素と `<graphic>` 要素にエンティティ名の指定に使用する必須パラメータが入っており、このエンティティ名によりグラフィック・イメージ・ファイルを参照します。

ファイル・エンティティにより、ヘルプタグ・ソースを複数のファイルに分割できます。他のファイルをマスタ・ヘルプタグ・ファイルに取り込んで処理を行うためには、エンティティの参照を使用してください。

エンティティの宣言に関する規則

- エンティティ名は、英字 (A から Z と a から z)、数字 (0 から 9)、マイナス (-) 記号から成り、必ず英字で始まります。
- エンティティ名は大文字でも小文字でも意味は変わりませんが、読みやすいように使い分けてください。
- エンティティ名の長さは、64 文字以内です。
- 各エンティティ名は、単一のボリューム内では固有のものでなければなりません。

▼ テキスト・エンティティを作成するには

1. 次のようにエンティティを宣言します。

```
<!entity Entityname "text">
```

Entityname はエンティティ名で、*text* はエンティティへの各参照に代わる文字列です。エンティティの宣言はすべて、ヘルプ・ボリュームの他のマークアップの前に行く必要があるので注意してください。

2. **text** 文字列を挿入する各位置に、次のようにエンティティの参照を入力します。

```
&Entityname;
```

& (アンド記号) と ; (セミコロン) の文字は、ヘルプタグ・ソフトウェアがエンティティの参照を適切に認識するために必要です。

例

次の行は「Society of Agricultural Engineers」という文字列が入っている Assoc という名前のテキスト・エンティティを宣言しています。

```
<!entity Assoc "Society of Agricultural Engineers">
```

次の文には、エンティティへの参照が記述されています。

```
Welcome to the &Assoc;.
```

ヘルプ・ボリュームがヘルプタグ・ソフトウェアで処理されると、エンティティの参照はエンティティの値に置き換えられます。そのため、文は次のように表示されます。

Welcome to the Society of Agricultural Engineers.

▼ ファイル・エンティティを作成するには

1. 次のようにエンティティを宣言します。

```
<!entity Entityname FILE "filename">
```

Entityname はエンティティ名で、*filename* はファイル名です。キーワード FILE は必須です。

2. 次のようにエンティティを参照します。

- ファイルがテキスト・ファイルの場合、ファイルの内容を挿入する各位置に次のようにエンティティの参照を入力します。

```
&Entityname;
```

&(アンド記号)と;(セミコロン)の文字は、ヘルプタグ・ソフトウェアがエンティティの参照を適切に認識するために必要です。

- ファイルがグラフィック・ファイルの場合、次のマークアップ行のうちの1つにパラメータとしてエンティティ名を入れます。

```
<figure entity=Entityname ... >
```

または

```
<graphic entity=Entityname ... >
```

または

```
<p gentity=Entityname ... >
```

注・ファイル名にはパスを入れないでください。ヘルプタグ・ソフトウェアを実行するときにファイルが現在のディレクトリにない場合は、適切な検索パスを `helptag.opt` ファイルに追加してください (詳細は、117ページの「実行時のヘルプ・ボリュームを作成するには」を参照してください)。

例: テキスト・ファイル・エンティティ

`file1`、`file2`、および `file3` という名前の3つのファイルと `<metainfo> ...</metainfo>` セクションがある4番目のファイルに、ヘルプ・ボリュームのテキストを記述したとします。次のようにして4つのファイルを `volume.htg` ファイルに取り込むことができます。

```
<!entity MetaInformation FILE "metainfo">
<!entity MyFirstFile FILE "file1">
<!entity MySecondFile FILE "file2">
<!entity MyThirdFile FILE "file3">

&MetaInformation;

<hometopic> My Home Title

Welcome to my application's help volume.

&MyFirstFile;
&MySecondFile;
&MyThirdFile;
```

例: グラフィック・ファイル・エンティティ

単一のヘルプ・ボリュームのホーム・トピックに図があり、図のグラフィック・イメージが picture.tif という名前のファイルに格納されているとします。次の例は、そのイメージが図でどのように使用されるかを示しています。

```
<!entity MetaInformation FILE "metainfo">
<!entity MyPicture FILE "picture.tif">

&MetaInformation;

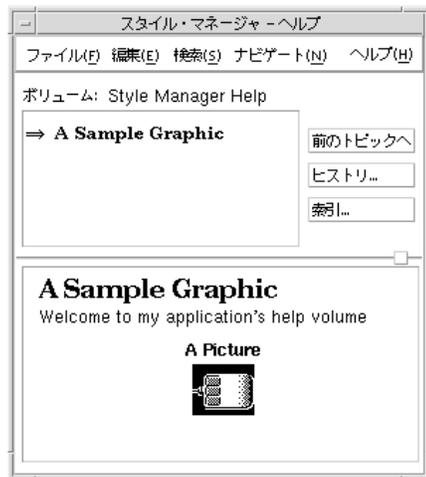
<hometopic> A Sample Graphic

Welcome to my application's help volume

<figure nonumber entity=MyPicture>
A Picture
<\figure>
```

テキスト「A Picture」は、図のタイトルです。

マークアップの出力は次のとおりです。



関連項目

- 103ページの「グラフィックの表示」

ヘルプ・トピックの記述

この章では、テキストをフォーマットするのに使用する要素を説明します。また、グラフィックの取り込み方と、他のヘルプ・トピックへのハイパーリンクの作成方法も説明します。この章で示す例は簡易マークアップを使用しています。

- 77ページの「ヘルプ・トピックの作成」
- 78ページの「トピック内に構造を作成する」
- 88ページの「インライン要素の入力」
- 91ページの「ハイパーリンクの作成」
- 99ページの「実行リンクの制御」
- 103ページの「グラフィックの表示」
- 108ページの「特殊文字の表示」
- 109ページの「コメントと作成者のメモの取り込み」
- 111ページの「索引の作成」
- 112ページの「用語集の作成」

ヘルプ・トピックの作成

ヘルプ・トピックは、固有の ID で識別できる情報の単位です。ヘルプ・トピックは、オンライン・ヘルプを記述しているプロダクトを最適に言い表す論理的な枠組みにグループ化されます。

記述する各トピックには、トピックの開始をマークする要素 (または、タグ) が付いていなければなりません。

```
<element id=id> Help Topic's Title  
The body of the topic
```

element は *chapter*, *s1*, *s2*, ..., *s9* のいずれかです。トピックの本文は、タイトルの後ならどの行からも始められます。

あるトピックのトピック階層内での位置は、そのトピックの開始に使われる *element* と、直前のトピックの開始に使われる *element* によって決まります。たとえば、あるトピックが *<s2>* で始まっていて、*<s1>* で始まるトピックの直後にある場合は、*<s2>* トピックは *<s1>* トピックのサブトピックになります。

アプリケーション (アプリケーション・ヘルプを記述している場合) かハイパーリンクのどちらかからアクセスするトピックには *id* が必要です。

ヘルプ・トピックのタイトルにはどんな文字列でも使用できます。タイトル文字列がソース・ファイルで 1 行を超える場合は、最後の行以外の行は *&* (アンド記号) で終わります。タイトル内の特定の位置で強制的に改行するには、** (バックスラッシュ) 文字を使用してください。

例

次の行は、*<s1>* タグを使用してトピックの開始をマークします。

```
<s1 id=welcome>Welcome to My Application
```

タイトルを強制的に 2 行に表示するには、次のように ** (バックスラッシュ) を使用してください。

```
<s1 id=welcome> Welcome to \ My Application
```

関連項目

- 第 2 章では、トピック階層の作成方法を含めたヘルプ・ボリュームの一般的な構造を説明しています。

トピック内に構造を作成する

ヘルプ・トピックの本文には、情報を構成して提示するために選択する次のような要素があります。

- 段落は、テキストの本文に使用します。
- リストは、箇条書きにして示す情報に使用します。リストの種類には、行頭文字が付いたもの、順番を付けたもの(番号の付いたもの)、何も付かないものがあります。
- サブヘッダは、トピック内を部分に分けるのに使用します。
- グラフィックは、インライン・グラフィックとしてテキスト内に入れたり、独立した図として段落の間に表示できます。
- ハイパーリンクで、関連トピックを参照できます。ハイパーリンクは、その階層内のより深層のサブトピックに続いたり、まったく異なる部分のトピックに分岐したり、他のヘルプ・ボリュームに続いたりします。
- コンピュータ・リテラルは、ファイル名や変数名などの、コンピュータが認識するテキストです。コンピュータ・リテラルは、それだけを分けて表示することも、インライン要素として表示することもできます。
- 注、注意、警告は、重要な情報へ読者の注意を促します。
- 強調表示は、パラグラフのテキスト内の重要な語句を強調表示するのに使用します。

▼ パラグラフを開始するには

- ◆ 前の段落または他の要素の後に空白行を 1 行挿入します。
または、そのパラグラフをインデントする場合には `<p indent>` 要素とパラメータを使用します。
または、ソース・ファイルでの段落の行の切れ目をそのまま維持したい場合は、`<image>` 要素を使用します。

`<p>` に対する終了タグは必須ではありません。しかし、`<image>` 要素には終了タグ `<\image>` は必須です。

例

次の例は、空白行 1 行で区切られた 2 つのパラグラフです。どちらの段落にも特別なパラメータがないので、`<p>` タグを入力する必要はありません(これは 1 行以上の空白行を入力した時点で想定されます)。

The Application Builder provides an interactive, graphical environment that facilitates the development of desktop applications.

The Application Builder is designed to make it easier for developers to construct applications that integrate well into the desktop. It provides two basic services: assembles Motif objects into the desired application user interface, and generates appropriate calls to the routines that support desktop integration services.

段落を左端からインデントしたい場合は、オプションの `indent` パラメータを指定します。

```
<p indent> An indented paragraph can be used to draw the reader's attention to an idea.
```

次の段落は、ヘルプ・ウィンドウでの自動ワード・ラップを無効にし、ソース・ファイルに入力されたとおりの行の切れ目を維持します。`<image>` 要素は住所を入力する際、特に便利です。

```
<image>
Brown and Reed Financial Investors
100 Baltic Place Suite 40 New York, New York
<\image>
```

▼ リストを作成するには

◆ `<list>` 要素は次のように使用します。

```
<list type spacing>
* item
* item
.
.
.
* item
<\list>
```

`type` はリストの種類で、`bullet` (デフォルト)、`order`、`plain` から選択できます。`spacing` は `loose` (デフォルト) か `tight` です。リスト内の各 `item` は * (アスタリスク) でマークします。

例

次の例は、簡単なリストです。*type* が指定されていないので、デフォルトの行頭に黒い丸を付けたリストになります。*spacing* も指定されていないので、デフォルトで各項目の間に空白行が1行ずつ入ります。

```
<list>
* Creating a Mail Message
* Sending a Message
* Reading Your Mail
<\list>
```

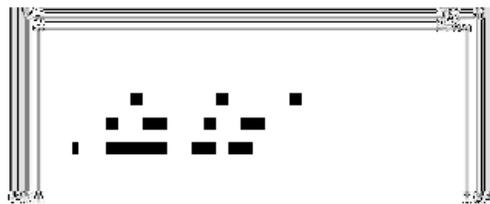
上記のマークアップのオンライン書式は次のとおりです。



同じリストに番号を付けて行間を詰めるには、次のようにします。

```
<list order tight>
* Creating a Mail Message
* Sending a Message
* Reading Your Mail
<\list>
```

出力は次のとおりです。



▼ ラベル付きリストを作成するには

`lablist` は、オプションでカラム・ヘッダが付く2段組みのリストです。

- ◆ ヘッダなしでラベル付きリストを作成するには、次のように `<lablist>` 要素を使用します。

```
<lablist spacing>
  \ label 1\ item 1 text
  \ label 2\ item 2 text
  .
  .
  \ label N\ item N text<\lablist>
```

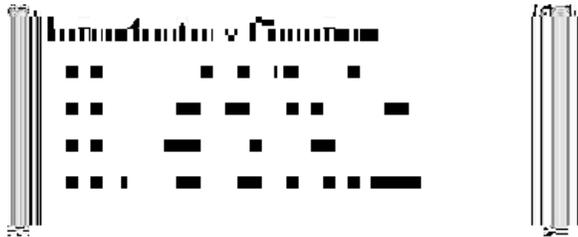
`spacing` は `loose` (デフォルト) か `tight` です。

例

次の例は、ラベル付きの章の説明のリストです。オプションのラベル・ヘッダは付いていません。

```
<lablist tight>
\Chapter 1\ An Overview of the System
\Chapter 2\ Installing the Operating System
\Chapter 3\ Configuring the Desktop
\Appendix A\ System Commands Quick Reference
<\lablist>
```

出力は次のとおりです。



▼ ラベル付きリストにヘッダを付けるには

- ◆ 次のように、`<lablist>` 要素と `<labheads>` 要素を使用します。

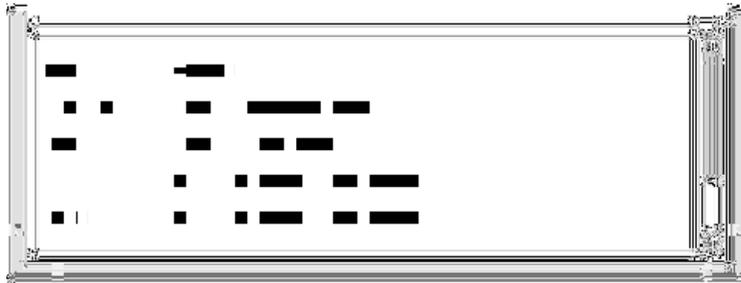
```
<lablist spacing>
  <labheads> \ heading for labels \ heading for items
  \ label 1\ item 1 text
```

```
\ label 2\ item 2 text
.
.
.
\ label N\ item N text
<\lablist>
```

例

```
<lablist>
<labheads>\Key \Action
\Previous\ Scroll to previous page
\Next\ Scroll to next page
\First\ Go to first page in document
\Last\ Go to last page in document
<\lablist>
```

上記のマークアップの出力は次のとおりです。



関連項目

- 172ページの「<list>」では、<list> 要素の使い方をまとめています。
- 165ページの「<lablist>」では、<lablist> 要素の使い方をまとめています。

▼ トピックにサブヘッダを付けるには

- ◆ 中ヘッダ (トピック・タイトルより少し小さい) を付けるには、次のマークアップを使用します。

```
<otherhead> Heading
```

小ヘッダを付けるには、次のマークアップを使用します。

<procedure> *Heading*

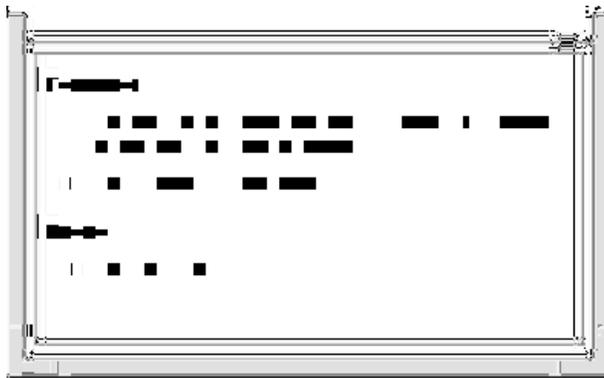
サブヘッダはトピックの中に構造を作りますが、トピック・ツリーのトピックのリストには表示されません。

例

<procedure> 要素を使用して各リストの前に小ヘッダを付けます。

```
<procedure>Keyboard
<list order>
* Use the Tab and direction keys to move the highlight to the icon
  you want to select.
* Press Return or Spacebar.
<\list>
<procedure>Mouse
<list bullet>
* Click the icon.
<\list>
```

上記のマークアップの出力は次のとおりです。



▼ コンピュータ・リストを表示するには

ヘルプタグ・マークアップとして解釈される特殊文字シーケンスを含まないコンピュータ・リストを表示するには、次のように <ex> (例) 要素を使用します。

```
<ex size>
Computer text here.
<\ex>
```

ヘルプタグが用いる特殊文字シーケンスを含むコンピュータ・リストを表示するには、次のように <vex> (逐語例) 要素を使用します。

```
<vex size>  
Computer text here.  
<\vex>
```

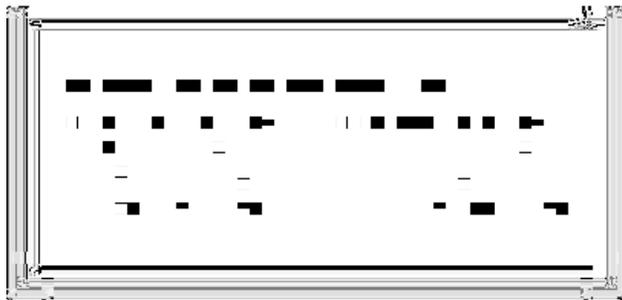
例文の表示に用いるフォントのサイズを決定するオプションの *size* 属性は、*smaller* または *smallest* に指定できます。

例

次の例は、端末エミュレータにディレクトリ・リストを表示するのに <ex> 要素を使用しています。

```
In this tutorial, you will edit these graphics files:  
<ex>  
H_ActionIcons.xwd      H_HelpWindows.xwd  
H_AppHelp.xwd          H_Hyperlinks.xwd  
H_Canonical.xwd        H_Icons.xwd  
H_FrontPanel.xwd       H_InlineGraphic.xwd  
<\ex>
```

上記のマークアップの出力は次のとおりです。



改行は、ソース・ファイルに入力したとおりに表示されます。例文の幅がヘルプ・ウィンドウに対して広すぎる場合は、水平方向のスクロール・バーが表示されるので、ユーザは例文のテキストをスクロールできます。

関連項目

- 89ページの「コンピュータ・リテラルを表示するには」

- 150ページの「<ex>」
- 195ページの「<vex>」

▼ 注、注意、警告を追加するには

- ◆ <note>、<caution>、<warning> の各要素は、次のように指定します。

```
<note>
Body of note here.
<\note>

<caution>
Body of caution here.
<\caution>

<warning>
Body of warning here.
<\warning>
```

注、注意、警告の各要素にアイコンを関連付けるには、そのアイコンを格納するグラフィック・ファイルが識別するファイル・エンティティを定義します。定義済みのエンティティ名を使用してください。

- <!ENTITY NoteElementDefaultIconFile FILE "filename ">
- <!ENTITY CautionElementDefaultIconFile FILE "filename">
- <!ENTITY WarningElementDefaultIconFile FILE "filename">

注、注意、警告にアイコンを付けたくない場合は、該当するエンティティを宣言しないでください。(すべてのエンティティ宣言は、ヘルプ・ボリュームの最初にあるその他のマークアップのどれよりも前に来なければならないことを忘れないでください。)そのようなエンティティの参照を指定する場合は、グラフィック・ファイルがヘルプタグ検索パス (helptag.opt) 上にあることを確認してください。

ヘルプ・システムが注、注意、警告の要素に使用するデフォルト・アイコン名は、次のエンティティで指定されます。

- <!ENTITY NoteElementDefaultIconFile FILE "noteicon.pm">
- <!ENTITY CautionElementDefaultIconFile FILE "cautionicon.pm">
- <!ENTITY WarningElementDefaultIconFile FILE "warningicon.pm">

これらのデフォルト・アイコンは /usr/dt/dthelp/dthelptag/icons ディレクトリにあります。

注、注意、警告の独自のアイコン・イメージを作成する場合は、イメージが割り当てられた領域に入るように小さくしておいてください。また、グラフィック・イメージは、helptag.opt ファイルに指定されたヘルプタグ検索パス上に存在しなければなりません。

例

次の注、警告、注意のマークアップの出力は図 3-1 のとおりです。

```
<note>  
Before installing your application, complete the options checklist  
to determine the amount of disk space required.  
<\note>  
  
<warning>  
This product is highly acidic and can cause skin irritation. Wearing  
protective gloves is mandatory when applying this product.  
<\warning>  
  
<caution>  
Do not place your fingers near the parrot cage!  
<\caution>
```



図 3-1 注、警告、注意のヘルプ・アイコン

関連項目

- 117ページの「実行時のヘルプ・ボリュームを作成するには」では、`helptag.opt` ファイルの使い方を説明しています。
- 72ページの「エンティティの使い方」

インライン要素の入力

インライン要素は、テキストの段落の中で語句をマークするのに使用します。これらの要素は、特定の項目を書式化するのに使用するフォントに影響します。

▼ 語句を強調表示するには

- ◆ `<emph>` (強調表示) 要素を次のように使用します。

```
<emph> text <\emph>
```

簡易形式では次のようになります。

```
!! text !!
```

イタリックで強調表示されたテキストが表示されます。日本語では、イタリック書体のフォントが提供されていない場合、それに代わる書体、たとえばボールドで表示されます。

例

重要な語を強調表示するには、次のようにします。

```
A thousand times <emph>no<\emph>
```

簡易形式では、

```
A thousand times !!no!!
```

どちらの場合も、「no」がイタリックで表示されます。

▼ 本のタイトルを入力するには

- ◆ `<book>` 要素を次のように使用します。

```
<book> title <\book>
```

簡易形式では、

```
book| title |
```

本のタイトルがイタリックで表示されます。日本語では、イタリック書体のフォントが提供されていない場合、それに代わる書体、たとえばボールドで表示されます。

例

このマニュアルのタイトルを入力するには、次のようにします。

```
<book|The Help System Author's and Programmer's Guide|
```

▼ Bold フォントで強調表示するには

- ◆ `<term>` 要素を次のように使用します。

```
<term nogloss> bold text <\term>
```

簡易形式では、

```
<term nogloss |bold text |
```

`<term>` 要素は用語集のエントリを作成するのに使用されます。しかし、`nogloss` パラメータを追加すれば、テキストは Bold フォントで表示されますが用語集には追加されません。

▼ コンピュータ・リテラルを表示するには

- ◆ `<computer>` 要素を次のように使用します。

```
<computer> text <\computer>
```

簡易形式では、

```
" text "
```

例

コンピュータ・テキストはファイル名を識別するのに便利です。次の例では、`helptag.opt` というファイル名が簡易マークアップでタグ付けされています。ファイル名がコンピュータ・テキストで表示されます。

```
Add the search path to your ``helptag.opt`` file.
```

上記のマークアップの出力は、次のとおりです。

```
Add the search path to your helptag.opt file.
```

▼ 変数を表示するには

- ◆ `<var>` (変数) 要素を次のように使用します。

```
<var> text <\var>
```

略式では、

```
<var |text |
```

簡易形式では、

```
%% text %%
```

変数がイタリックで表示されます。

例

次のコマンド行形式には、ユーザがファイル名を指定することを示す変数があります。

```
dtpad %%filename%%
```

出力は次のとおりです。

```
dtpad filename
```

変数は、コンピュータ・テキストまたはコンピュータの例表示に現れます。次の例は *volume* をファイル名の変数部分として指定します。

```
The HelpTag software takes your ``%%volume%%.htg`` file as input.
```

出力は次のとおりです。

```
The HelpTag software takes your volume.htg file as input.
```

上記のどちらの例も、%% の対の部分は正規マークアップ (<var>...<\var>) でも略式マークアップ (<var| ...|) でも入力できます。

ハイパーリンクの作成

ハイパーリンクは、ヘルプ・ボリュームの特定のトピックや位置を参照します。このためには、[参照](#)

したい要素に固有の ID が指定されていなければなりません。これらのヘルプタグ要素は、<chapter>、<s1...s9>、<location>、<p>、<image>、<figure>、<graphic> などの割り当て済み ID のこともあります。

ヘルプは、5 種類のハイパーリンクをサポートしています。

- ハイパーテキスト・リンクは、他のヘルプ・トピックへ「ジャンプ」します。デフォルトでは、新しいトピックは同じウィンドウに表示されますが、新規ウィンドウに表示するよう要求することもできます。
- 定義リンクは、簡単なポップアップのヘルプ・ウィンドウにトピックを表示します。ほとんどの場合、定義リンクは文中の新しい語句の定義を見るために使用されます。
- マニュアル・ページへのリンクは、システムにインストールされているマニュアル・ページを表示します。
- 実行リンクは、シェル・コマンドまたはプログラムを実行します。これにより、ユーザがハイパーリンクを起動すると発生する動作は、非常に広範囲なものになります。

- アプリケーション定義リンクは、アプリケーションが解釈するカスタム・リンクを作成します。これにより、ヘルプ・システムとアプリケーション間の通信機能が提供されます。

ある要素へのハイパーリンクを作成するには、ハイパーリンク・コマンドにその要素の ID を指定します。ヘルプタグは、ID へのハイパーリンク作成に使用できる `<xref>` と `<link>` の 2 つの要素が提供しています。また、`<p>`、`<image>`、`<figure>` の各要素は、グラフィック・イメージを使ったハイパーリンクの作成に使用できます。

<xref> 要素の使い方

章や節などのタイトルの要素にリンクする場合、最も簡単な方法は `<xref>` 要素を使うことです。`<xref>` を使ってリンクを作成するには、リンク先のトピックの ID を指定します。そのトピックのタイトルが `<xref>` 要素に代入され、アクティブなハイパーリンクになります。

`<xref>` で作成されるハイパーテキスト・リンクは、新しいトピックを同じウィンドウに表示します。他のリンク型を使って異なる動作にする場合は、`<link>` 要素を使用しなければなりません。

また、組み込み ID を持つトピック (`<hometopic>`、`<glossary>` など) へジャンプするために `<link>` を使用できません。それらの要素へのハイパーリンクを作成するには、`<xref>` 要素を使用しなければなりません。

▼ <xref> を使ってリンクを作成するには

- ◆ `<xref>` 要素を次のように使用します。

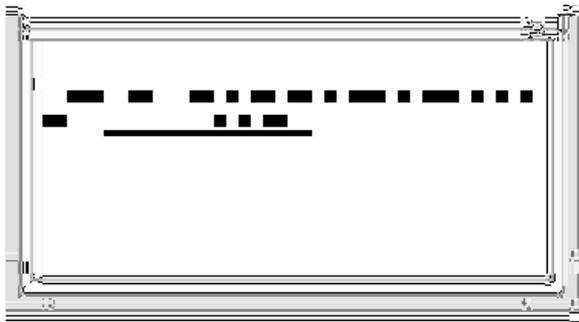
```
<xref id>
```

id は、リンクを作成したい章またはセクションの ID です。ID の大文字と小文字の区別は関係ありません。

次の例は、セクションのタイトルへのリンクを作成します。

```
<sl id=colorpalettes>Desktop Color Palettes  
.  
.  
.  
To learn how to change the colors used on your desktop,  
refer to <xref colorpalettes>.
```

<xref> 要素がセクションのタイトルに置換されます。このタイトルはハイパーリンクであり、下線によって示されています。この文はヘルプ・ボリュームでは、次のように表示されます。



リンク要素の使い方

標準ハイパーテキスト・リンクを作成するには、<xref> と <link> のどちらの要素も使用できます。しかし、91ページの「ハイパーリンクの作成」にあるその他のリンク型を使用するには、<link> 要素を使用しなければなりません。

▼ <link> を使ってリンクを作成するには

- ◆ 同じボリューム内のトピックヘジャンプするには、<link> 要素を次のように使用します。

```
<link id>text<\link>
```

id は、ヘルプ・ボリュームのどこかで宣言されている ID です。*text* は、下線によってアクティブなハイパーリンクだと示しているヘルプ・テキストの部分です。

例

次の例は、前の例の <xref> 要素の代わりに <link> 要素を使用しています。

```
<sl id=colorpalettes>Desktop Color Palettes
.
.
.
To learn how to change the colors used on your desktop,
refer to <link colorpalettes>Desktop Color Palettes<\link>.
```

定義済み ID へのリンクを作成するには

- ◆ 定義済みの ID を持つトピック (同じボリューム内) へジャンプするには、<link> 要素を次のように使用します。

```
<link hyperlink="id">text<\link>
```

定義済みの ID はすべて _ (下線) 文字で始まります。したがって、hyperlink="id" の形式を使用する必要があります。

例

次のリンクは、現在のボリュームのホーム・トピックへジャンプします。

```
Return to <link hyperlink="_hometopic">Introduction<\link>.
```

他のボリュームにあるトピックへのリンクを作成するには

- ◆ 他のヘルプ・ボリュームにあるトピックへジャンプするには、次のようにします。

```
<link hyperlink="volume id" JumpNewView>text<\link>
```

他のボリューム (ジャンプ先) が登録されている場合は、*volume* パラメータはボリューム・ファイルのベース名だけです。登録されていない場合は、そのボリュームへの完全パス名を指定しなければなりません。

他のボリュームへのリンクには、JumpNewView パラメータを使うようにします。このパラメータを使うと、ユーザは他のボリュームへジャンプしたことがわかります。前の画面表示が残るので、どこからジャンプしてきたかがわかります。

例

次のリンクは、GeoMap というヘルプ・ボリュームのホーム・トピックへジャンプします。

```
To view a map of the United States, see <link hyperlink="GeoMap_hometopic"> Geography Maps <\link>.
```

次の例は同じリンクですが、新規ウィンドウにトピックを表示します。

```
To view a map of the United States, see <link hyperlink=''GeoMap_hometopic'' type=JumpNewView> Geography Maps <\link>.
```

次のリンクは、Intromgr というヘルプ・ボリュームの DesktopKeyboardNav というトピックへジャンプします。

```
For more information, see <link hyperlink=''Intromgr_DesktopKeyboardNav''>Keyboard Shortcuts for the Desktop<\link>.
```

目的としているヘルプ・ボリュームがデスクトップに登録されていない場合は、必ずそのボリュームへの絶対パス名を指定するか、helptag.opt ファイルに適切な検索パスを指定してください。

関連項目

- 271ページの「アプリケーションとそのヘルプの登録」
- 152ページの「<figure>」
- 161ページの「<image>」
- 169ページの「<link>」
- 182ページの「<p>」
- 197ページの「<xref>」

▼ 定義リンクを作成するには

- ◆ 用語集にある用語へリンクする場合は、<term> 要素を次のように使用します。

```
<term>text<\term>
```

簡易形式では、

```
++text++
```

<term> 要素を使う場合は、対応する定義が用語集に含まれていることを確認する必要があります。

同じヘルプ・ボリューム内のトピックへリンクする場合は、<link> 要素を次のように使用します。

```
<link id Definition>text<\link>
```

id はトピック ID (またはトピック内の要素の ID) で、*text* はアクティブなハイパーリンクにしたいヘルプ・テキストの部分です。Definition というキーワードは、そのリンクが簡易ヘルプ・ダイアログ・ボックスをポップ・アップで表示するよう指定します。

または、他のヘルプ・ボリューム内のトピックへリンクする場合は、<link> 要素を次のように使用します。

```
<link hyperlink="volume id" Definition>text<\link>
```

他のボリューム (リンク先) が登録されている場合は、*volume* パラメータはボリューム・ファイルのベース名だけです。登録されていない場合は、必ずそのボリュームへの絶対パス名を指定しなければなりません。

例

次のリンクは、著作権のトピックをメタ情報で表示する定義リンクを作成します。

```
<link hyperlink="_copyright" type=Definition>Version Information<\link>
```

「Version Information」の句がハイパーリンク・テキスト (破線の下線) になります。

関連項目

- 112ページの「用語集の作成」
- 190ページの「<term>」
- 169ページの「<link>」

▼ マニュアル・ページへのリンクを作成するには

- ◆ <link> 要素を次のように使用します。

```
<link manpage Man>text<\link>
```

特定のセクションからマニュアル・ページを要求するには、`hyperlink` パラメータを次のように使用します。

```
<link hyperlink="section manpage" Man>text<\link>
```

マニュアル・ページへのリンクの場合、`hyperlink` パラメータは、端末エミュレータ・ウィンドウで `man` コマンドを実行するときに入力するのと同じ文字列です。

注・アプリケーション用のヘルプを記述していて、マニュアル・ページへのリンクを指定する場合、そのアプリケーションはマニュアル・ページに対する特別なサポートを指定していなければなりません。詳細は、244ページの「マニュアル・ページを表示するには」を参照してください。(デスクトップのヘルプ・ビューアでは、マニュアル・ページへのリンクがサポートされています。)

例

次の例は、`grep` コマンドのマニュアル・ページを表示するリンクです。

```
Refer to the <link grep Man> grep(1)<\link> command.
```

「Man」は<link>要素へのキーワードです。ですから、`man` コマンドのマニュアル・ページを表示するリンクを作成する場合は、必ず `hyperlink` パラメータを使用してください。

```
Refer to the <link hyperlink="man" Man>man(1)<\link> command.
```

特定のセクションにマニュアル・ページを表示するには、マニュアル・ページ名の前にセクション番号を付けます。次のリンクは、セクション 2 から「mkdir」マニュアル・ページを表示します (セクション 1 の同じ名前のマニュアル・ページとは異なります)。

Refer to the `<link hyperlink= "2 mkdir" Man>mkdir(2)<\link>` command.

関連項目

- 169ページの「<link>」

▼ アプリケーション定義リンクを作成するには

- ◆ <link> 要素を次のように AppDefined パラメータと共に使用します。

```
<link hyperlink="data" AppDefined>text<\link>
```

data は、このリンクが起動されるときにアプリケーションに渡されるテキスト文字列で、*text* はハイパーリンクです。

例

3 種類のレポートを出力するアプリケーションのためのヘルプを記述していると想定します。次のように 3 つのハイパーリンクを作成します。

```
Choose a report type:
<list plain tight>
* <link hyperlink="Report-Daily" AppDefined>Daily Report<\link>
* <link hyperlink="Report-Month-To-Date" AppDefined>MTD Report<\link>
* <link hyperlink="Report-Year-To-Date" AppDefined>YTD Report<\link>
<\list>
```

アプリケーションが、これらの特別なリンクを処理し、ハイパーリンク文字列を解釈するよう設定されている場合は、アプリケーションはユーザが選択するハイパーリンクに基づいて適切なレポートを生成します。

完全な例については、`/usr/dt/share/examples/dthelp` ディレクトリにあるアプリケーション例のコードを参照してください。

▼ メタ情報トピックへリンクするには

- ◆ <link> 要素を次のように使用します。

```
<link hyperlink="_id">text<\link>
```

`id` は、リンクしたい先の要素に関連付けられた定義済み ID です。`text` は、アクティブなハイパーリンクにしたい語句です。

メタ情報のセクション内のトピックのほとんどは定義済み ID を持っているので、設計者定義 ID は受け付けません。定義済み ID は、要素名の前に下線文字が付いた形式です。たとえば、`<copyright>` トピックの ID は `_copyright` です (大文字と小文字の区別は関係ありません)。

メタ情報の定義済み ID は次のとおりです。

<code><abstract></code>	<code>(_abstract)</code>
<code><copyright></code>	<code>(_copyright)</code>
<code><title></code>	<code>(_title)</code>

`<otherfront>` 要素で入力されたトピックは、トピック階層にある通常のどのトピックとも同じようにリンクできます。

関連項目

- 70ページの「組み込み ID」に、ヘルプ・システムの定義済み ID をリストしています。

実行リンクの制御

ほとんどのハイパーリンクは関連するヘルプ・トピックを表示しますが、ハイパーリンクはシェル・コマンドおよびスクリプトを実行することもできます。このリンクを実行リンクと呼びます。実行リンクはユーザのシステムと対話するので、ヘルプ・システムは実行リンクの処理方法をコントロールするための実行ポリシーを提供します。

ヘルプ・システムは、実行リンクの動作を定義するためにリソースを使用します。DtNexecutionPolicy リソースは、ヘルプ・システムによる実行リンクの処理方法を変更するために、アプリケーションのアプリケーション・デフォルト・ファイルに設定されます。そのほかに、ヘルプ・システムは実行別名と呼ばれるリソースを使用します。実行別名は、実行リンクが実行するコマンド文字列またはスクリプトに名前 (またはラベル) を割り当てるリソースです。

実行ポリシーのデフォルト動作

実行リンクが選択され、そのリンクが実行別名を持つ場合、ヘルプ・システムは別名の値を検索してコマンドを実行します。実行別名が定義されていない場合、ヘルプ・システムは、実行されることになるコマンドを示す確認ダイアログ・ボックスを表示して、そのコマンドを実行するか操作を取り消すかをユーザに尋ねます。

実行別名

ヘルプ・ボリュームで実行リンクを使用するときには、実行別名を作成する必要があります。これはつまり、アプリケーションのアプリケーション・デフォルト・ファイルに、実際に実行されるコマンドを表す別名(名前)を定義することです。この方法の1つの利点は、実際のコマンドをヘルプ・ボリュームのソース・ファイルから切り離すことです。これにより、ヘルプ・ボリュームのハイパーリンクを変更せずに、アプリケーション・デフォルト・ファイルのコマンドを編集できるようになります。各ハイパーリンクは1つの別名を参照します。別名は、内容が編集されても名前が変わることはありません。たとえば、スクリプトを使用するチュートリアル・ヘルプ・ボリュームは、アプリケーション・デフォルト・ファイルでシェル・スクリプト・コマンドを変更することで、特定のシェル環境を簡単にカスタマイズできます。

実行別名を作成するには

実行別名をアプリケーションのアプリケーション・デフォルト・ファイルに作成するには、次のリソース指定構文を使用します。

```
application_name.executionAlias.alias_name: command
```

<i>application_name</i>	ヘルプ・ボリュームを所有するアプリケーション名またはクラス名
<i>executionAlias</i>	リソースが別名であると識別するキーワード
<i>alias_name</i>	コマンドに割り当てられた名前
<i>command</i>	このリンクで実行されるシェル・コマンドまたはスクリプト

command 文字列の長さに制限はありません。複数の行に渡るコマンドを入力するには、(最終行を除く)各行の終わりに \ (バックスラッシュ) を付けてください。

例

次のリソース・エントリは、端末エミュレータを起動する StartDtterm という実行別名を作成します。& (アンド記号) は、コマンドをバックグラウンドで実行します。

```
Dtterm.executionAlias.StartDtterm: dtterm &
```

次のエントリは、NightAlert というアプリケーションで xclock アプリケーションを実行する xclockAlias という別名を作成します。

```
NightAlert.executionAlias.xclockAlias: xclock &
```

ハイパーリンクでの実行別名の使用

実行別名は、<link> 要素を使って参照したり、<p> や <figure> など、ハイパーリンク・パラメータを持つ要素と共に使用できます。

実行別名を使用して実行リンクを作成するには

- ◆ <link>要素を次のように使用します。

```
<link ``DtHelpExecAlias alias_name [default_command]`` Execute >text<\link>
```

DtHelpExecAlias	リンクが実行別名であると識別するキーワード
<i>alias_name</i>	実行別名のリソース指定に別名として定義された名前
<i>default_command</i> オプション	オプションで、これを指定すると、実行別名がアプリケーションのアプリケーション・デフォルト・ファイルからまだ読み込まれていない場合に、このコマンドが実行されます。たとえば、ヘルプ・ビューアなどのインフォメーション・ビューアでヘルプ・ボリュームを表示しているときには、アプリケーションのリソースは読み込まれていません。
<i>text</i>	ハイパーリンク・テキスト (下線が付けられる) として指定したいヘルプ・テキストの部分です。

注 - 実行しているコマンドがすぐに終了しない場合は、そのコマンドに & (アンド記号) を追加してバックグラウンドで実行してください。そうしないと、ヘルプ・ウィンドウはコマンドが終了するまで動作しません。

例

次のハイパーリンクは、xclockAlias という実行別名を参照します。別名のソース定義は、100ページの「実行別名」に示します。

このリンクは、xclock プログラムをバックグラウンドで起動します。「Start the Clock」の句がハイパーリンクになります。そのハイパーリンクをクリックすると、別のウィンドウでxclock プログラムが実行されます。プログラムを終了するには、ウィンドウを閉じてください。

```
<link ``DtHelpExecAlias xclockAlias`` Execute>Start the Clock<\link>
```

次の例は、同じハイパーリンクにオプションのデフォルト・コマンドを指定しています。

```
<link ``DtHelpExecAlias xclockAlias xclock &`` Execute>Start the Clock<\link>
```

DtNexecutionPolicy リソース

DtNexecutionPolicy リソースは、システム管理者またはユーザが、指定したアプリケーションに対して適切なセキュリティのレベルを選択できるようにします。

設定できるリソースの値は次のとおりです。

help_execute_query_all	すべての実行リンクを照会します。
help_execute_query_unaliased	実行別名が定義されていないリンク・コマンドだけを照会します。
help_execute_none	どんな実行リンクも実行しません。
help_execute_all	すべての実行リンクを実行します。

デフォルト値は help_execute_query_unaliased です。実行別名として定義された実行リンクは自動的に実行されますが、その他の実行リンクに関しては、ヘルプ・システムが確認ダイアログ・ボックスを表示します。

アプリケーション開発者が `DtNexexecutionPolicy` を設定することは避けた方がよいでしょう。システム管理者やユーザがこの値を変更できなくなってしまうからです。

関連項目

- 169ページの「<link>」
- 152ページの「<figure>」
- 182ページの「<p>」
- `DtHelpDialog(3)`
- `DtHelpQuickDialog(3)`

グラフィックの表示

ヘルプでは 4 種類のグラフィック形式をサポートしています。

- *TIFF (Tagged Image File Format)*
多くの標準の描画したりスキャンしたりするアプリケーションによって作成されるカラー、グレースケール、白黒イメージ (*filename.tif*)
- X ウィンドウ・ダンプ
`xwd` ユーティリティで作成される X Window System からの画面ダンプ (*filename.xwd*)
- X ピクスマップ
カラーのアイコン・イメージ (*filename.pm*)
- X ビットマップ
2 色のアイコン・イメージ (*filename.bm*)
各グラフィックは別のファイルに保存されます。ファイル形式は、上記のファイル名拡張子を使って決定されます。

▼ 図を作成するには

1. 図中に取り込むイメージ・ファイルを識別するために、ファイル・エンティティを宣言します。

```
<!entity graphic-entity FILE "filename.ext">
```

エンティティ宣言はすべて、ヘルプ・ボリュームの最初にある、その他のマークアップよりも前にある必要があることを忘れないようにして下さい。

2. <figure> 要素を次のように使用します。

```
<figure entity=graphic-entity>  
caption string  
<\figure>
```

graphic-entity は、表示したいグラフィック・ファイルのエンティティ名です。*caption string* はオプションの文字列です。タイトルのテキストはグラフィックの上に表示されます。

デフォルトでは図に番号が付きます。その番号は、タイトルの前につきます。番号を付けない図を作成するには、*nonumber* パラメータを指定してください (下記の「例」を参照してください)。

図をハイパーリンクにしたい場合は、*ghyperlink* (グラフィック・ハイパーリンク) と *glinktype* (グラフィック・リンク型) パラメータを次のように使用します。

```
<figure entity=graphic-entity ghyperlink="id" glinktype=type>  
caption string  
<\figure>
```

ghyperlink パラメータと *glinktype* パラメータは、<link> 要素の *hyperlink* パラメータと *type* パラメータと同じように動作します。

例

以下の例では、ヘルプ・ボリュームの最初で次の2つのファイル・エンティティを宣言したと想定します。

```
<!entity FirstPicture FILE "first.tif">  
<!entity SecondPicture FILE "second.pm">
```

- 次の図は *first.tif* ファイルのグラフィックを表示し、番号 (デフォルト) とタイトルを表示します。

```
<figure entity=FirstPicture>  
Here's the First Picture  
<\figure>
```

- 次の例は、番号とタイトルを付けずに second.pm ファイルを表示します。

```
<figure nonumber entity=SecondPicture>  
<\figure>
```

図に ID を付けるには、タイトルがなければなりません。タイトルは、<xref> が図の ID を使用する場合に必要です。その場合、タイトルは <xref> の位置に挿入され、その図のハイパーリンクになります。

- 次の図は、xclock プログラムを実行する実行ハイパーリンクです。

```
<figure entity=SecondPicture ghyperlink="xclock &" glinktype=execute>  
Choose This Figure to Start the Clock  
<\figure>
```

関連項目

- 152ページの「<figure>」
- 169ページの「<link>」

▼ インライン・グラフィックを表示するには

1. 図中に取り込むイメージ・ファイルを識別するために、ファイル・エンティティを宣言します。

```
<!entity graphic-entity FILE "filename.ext">
```

エンティティ宣言はすべて、ヘルプ・ボリュームの最初にあるその他のマークアップよりも前にある必要があります。

2. <graphic> 要素を次のように使用します。

```
... text <graphic entity=graphic-entity> text ...
```

graphic-entity は、表示したいグラフィック・ファイルのエンティティ名です。

グラフィックをハイパーリンクとして使用するには、<link> 要素の中に入れます。

```
<link parameters><graphic entity=graphic-entity><\link>
```

注 - <graphic> 要素は、大きめのイメージの使用も可能ですが、基本的に小さいグラフィック用です。グラフィックの縦の長さが入るように、空白スペースが追加されます。

例

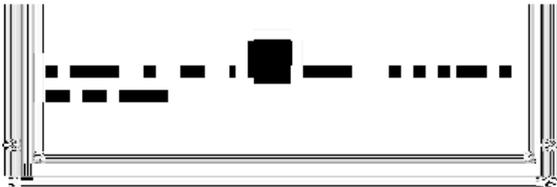
次の例は、文の途中で小さな X ビットマップ・イメージを使用します。まずボリュームの最初で、ビットマップ・ファイルをファイル・エンティティとして宣言しなければなりません。

```
<!entity StopWatch FILE "stopwatch.bm">
```

ヘルプ・テキストの中に、イメージは `<graphic>` 要素を使って挿入されます。

```
Whenever you see the <graphic entity=StopWatch> symbol, stop and answer the quiz questions.
```

上記のマークアップの出力は次のとおりです。



▼ グラフィックのまわりをテキストで囲むには

1. パラグラフ中に入れるイメージ・ファイルを識別するために、ファイル・エンティティを宣言します。

```
<!entity graphic-entity FILE "filename.ext">
```

2. `<p>` (段落) 要素を `gentity` パラメータと共に次のように使用します。

```
<p gentity=graphic-entity> Paragraph text here ...
```

`graphic-entity` は、挿入したいグラフィック・ファイルを参照するエンティティ名です。

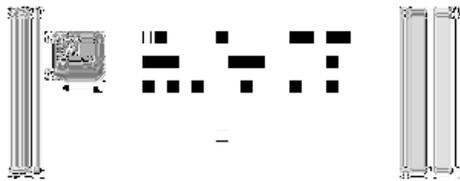
例

sample.pm という名前のアイコンを表示して、そのアイコンのまわりをテキストで囲みたい場合は、まずファイル・エンティティを宣言します。

```
<!entity HelpKeyIcon FILE "helpkey.xwd">
```

次に段落を入力します。

```
<p gentity=HelpKeyIcon gposition=left> The F1 key is a Help key. When  
you press F1, the application you are using displays the help topic  
most closely related to your current activity.
```



グラフィックを右揃えにするには、次のように `gposition` パラメータを追加します。

```
<p gentity=HelpKeyIcon gposition=right>Many desktop components  
support multicolor icons, in addition to two-color images.
```

次の例は、アイコンのまわりを囲む段落のマークアップです。このアイコンは、新規ウィンドウに `icon-editor` という ID のトピックを表示するハイパーリンクです。

<p entity=my-icon hyperlink="icon-editor" glinktype=JumpNewView>
Many desktop components support multicolor icons, in addition to the two-color images.

関連項目

- 182ページの「<p>」

特殊文字の表示

ヘルプタグでは、多くの特殊文字と記号を使用できます。適切なエンティティの参照を入力することで、特定の文字を表示できます。

特殊文字エンティティの一部は `helpchar.ent` ファイルで宣言されています。`helpchar.ent` ファイルは `/usr/dt/dthelp/dthelptag` ディレクトリにあります。これらの文字にアクセスするには、特定のエンティティ宣言を自分のボリュームにコピーするか、`helpchar.ent` ファイル全体を取り込んでください。不要なエンティティ宣言は無視されます。

使用できる文字のリストについては、第 6 章を参照してください。

▼ 特殊文字を表示するには

1. 表示したい文字のエンティティ名を定義するには、第 6 章を参照してください。また、組み込み特殊文字かどうかに注意してください。
2. 組み込み特殊文字でない場合は、次の 2 行をその他のエンティティ宣言に追加します (***entity-name*** は表示したい文字のエンティティ名です)。

```
<!entity entity-name FILE "helpchar.ent"> &entity-name;  
&entity-name;
```

また、次の行を `helptag.opt` ファイルに追加します。

```
search=/usr/dt/dthelp/dthelptag
```

ヘルプタグに組み込まれている文字の場合は、手順 2 を省略できます。

3. 特殊文字を表示したい場所には、エンティティ参照を入力します。

```
&entity-name;
```

例

著作権の記号 (©) のエンティティは組み込み特殊文字なので、次のエンティティを使用するだけで表示できます。

```
&copy;
```

大文字のギリシャ文字シグマ (Σ) を表示するには、まず次のように (ヘルプ・ボリュームの最初に他のエンティティ宣言と共に) `helpchar.ent` ファイルを取り込まなければなりません。

```
<!entity SpecialCharacterEntities FILE "helpchar.ent">
&SpecialCharacterEntities;
```

次に、シグマ文字を表示する場所に次のエンティティ参照を入れます。

```
&Usigma;
```

他のエンティティと同じく、特殊文字のエンティティ名において大文字と小文字の区別は関係ありません。

関連項目

- 第6章

コメントと作成者のメモの取り込み

ソース・ファイルの中に、ヘルプ・テキストの一部にはならないコメントを入れておくと、とても便利です。コメント要素でマークされたテキストは、ヘルプタグ・ソフトウェアには常に無視されます。コメントは、自分自身や他の設計者への注意書きを付けたり、あるマークアップをファイルから取り出さずに除外するのに使用できます。

標準のコメントに加えて、作成者のメモを入力するためにヘルプタグは `<memo>` 要素を提供しています。メモは、レビューの間はヘルプ・トピックに表示されますが、最終的なヘルプ・ファイルの作成時には表示されません。通例、設計者はレビュー担当者に対して質問や注意書きを記述するのに `<memo>` 要素を使用します。

▼ コメントを挿入するには

- ◆ コメントの開始マーク (`<!--`) と終了マーク (`-->`) を次のように使用します。

```
<!-- text here is completely ignored -->
```

ヘルプタグ・ソフトウェアは、`<!--` と `-->` の間のすべてのマークアップを無視します。コメントは、別のコメントの中に入れることはできません。

例

次の例には2つのコメントがあります。1つは段落の前の行で、もう1つはパラグラフ中の語です。

```
<!-- Here is my rough draft of the introduction: -->
```

```
Welcome to my application. This software  
is <!-- perhaps --> the fastest and most  
efficient software you'll ever own.
```

▼ 作成者のメモを挿入するには

- ◆ `<memo>` 要素を次のように使用します。

```
<memo>  
text <\memo>
```

デフォルトでは、`<memo>` 要素内のテキストはヘルプタグ・ソフトウェアには無視されます(コメントと同じ)。しかし、`helptag.opt` ファイルに `memo` オプションを追加すると(または `dthelptag` コマンドで `memo` オプションを指定すると)、ヘルプ・ボリューム内のすべてのメモがボールド・フォント表示されます。

例

アプリケーションについて記述をしているとき、プロジェクト・チームに対して質問があるとします。`<memo>` 要素を次のように使用してテキスト内に質問を入れることができます。

```
<memo>Team: Will the product also  
support 32-bit characters?<\memo>
```

次のコマンドでヘルプ・ボリュームを処理すると (または `helptag.opt` ファイルに `memo` を指定すると)、メモはヘルプ・テキストにボールド・フォントで表示されます。

```
dthelptag volume memo
```

メモ・オプションを使用しない場合 (または `nomemo` オプションを使用する場合)、メモ内のテキストは無視され、ヘルプ・テキストには表示されません。

索引の作成

ヘルプ・ボリュームの索引は本の索引と似ています。設計者にとって、トピックの索引エントリを作成することは大切です。それによって、ユーザがキーワードや概念で検索できます。詳細な索引が作成されていると、ユーザはすばやく正確にトピックを見つけられます。

▼ 索引エントリをマークするには

- ◆ 索引に入れたいトピックの中で、`<idx>` 要素を次のように使用します。

```
<idx>keyword<\idx>
```

略式では、

```
<idx|keyword|
```

エントリをソートするには、`<sort>` 副要素を次のように使用します。

```
<idx>keyword<sort>sortkey<\idx>
```

`keyword` は索引に表示したいテキストで、`sortkey` はソート時に使用されるテキストです。

`<idx>` 要素はトピック内のどこでも使用できます。`keyword` とオプションの `sortkey` のどちらもトピックには表示されません。

例

次の例は、2つのキーワード索引エントリを持つトピックの最初の部分です。

```
<sl id=getting-started>Getting Started with Helpview
<idx>starting Helpview<\idx>
<idx> Helpview, starting<\idx>

Welcome ...
.
.
.
```

次の例は、キーワード索引の「プラス」という語が入るべき位置に + (プラス文字) を挿入します。

```
<idx>+<sort>plus<\idx>
```

用語集の作成

本用語集のように、重要な用語を定義する用語集をヘルプ・ボリュームに入れることができます。用語集は `<glossary>` 要素でマークされ、ヘルプ・ボリュームの最後のトピックになります。

ヘルプ・ボリューム全体を通じて、`<term>` 要素と共に入力したキーワードとなる語句は、自動的に用語集にあるその用語の定義への定義ハイパーリンクになります。

▼ 用語集の用語をマークするには

- ◆ `<term>` 要素を次のように使用します。

```
<term>word or phrase<\term>
```

略式では、

```
<term|word or phrase|
```

簡易形式では、

```
++word or phrase++
```

ヘルプ・テキスト内の用語のスペルが用語集の定義のスペルと完全には同じではない場合は、その用語の「`glossary form`」(用語集形式)を次のように指定できます。

```
<term "glossary form">word or phrase<\term>
```

glossary form は、その用語が用語集に表示されるとおりの形式です。その用語が(コンテキストにより) ヘルプ・トピックでは複数形でなければならず、用語集では単数形でなければならない場合に便利です。

用語はボールド・フォントで表示され、自動的に定義ハイパーリンクになります。用語が選択されると、用語集の定義が簡易ヘルプ・ダイアログに表示されます。

注 - 故意に用語集に定義しない用語をマークする場合は、<term> 要素に `nogloss` 属性を追加してください。これにより、その用語は用語用のボールド・フォントで表示されますが、用語集へのリンクは作成されません。

例

用語集に「widget」の定義がある場合は、次のように入力できます。

```
A ++widget++ is the fundamental building block of OSF/Motif user interfaces.
```

用語集のエントリは「widget」だが、文中では複数形を使用する必要がある場合、次のように用語を入力できます。

```
<term "widget">Widgets<\term> are the fundamental building blocks  
of OSF/Motif user interfaces.
```

同じ用語を入力したいが、用語集に入れたくないか、またはハイパーリンクにしたくない場合は、次のように `nogloss` パラメータを使用します。

```
<term nogloss> Widgets<\term>are the fundamental building blocks  
of OSF/Motif user interfaces.
```

略式では次のとおりです。

<term nogloss|Widgets| are the fundamental building blocks of OSF/Motif user interfaces.

▼ 用語集に用語を定義するには

- ◆ <dterm> 要素を用語集に次のように入力します。

```
<glossary>
.
.
.
<dterm>word or phrase
Definition of the term
.
.
.
```

<dterm> の語句は用語集の中で必ずソートしてください。

例

次の例は、用語集の SGML という用語の定義の部分です。

```
<glossary>
.
.
.
<dterm>SGML
Standard Generalized Markup Language. An
international standard [ISO 8859: 1986] that
establishes a method for information interchange.
SGML describes constructs for marking the
structure of information separate from its
intended presentation or format.
```

関連事項

- 145ページの「<dterm>」
- 154ページの「<glossary>」
- 190ページの「<term>」

ヘルプ・ボリュームの処理と表示

この章では、ヘルプ・システムをオンライン形式で表示するために作成するマークアップしたヘルプ・ファイルの処理方法を説明します。また、デスクトップのフロントパネルのヘルプビューアからヘルプ・ボリュームにアクセスする方法も説明します。

- 115ページの「概要」
- 117ページの「実行時のヘルプ・ファイルの作成」
- 121ページの「ヘルプ・ボリュームの表示」
- 123ページの「ブラウザ・ボリュームへのヘルプの追加」
- 128ページの「ヘルプ・トピックの印刷」
- 129ページの「ヘルプのテスト」

概要

ヘルプ・ボリュームを表示するには、ヘルプタグ・ソフトウェアでファイルを処理し、実行時のヘルプ・ファイルを作成しなければなりません。実行時のファイルは、セマンティック・デリバリ言語 (Semantic Delivery Language) と呼ばれるオンライン・プレゼンテーション形式を使用します。実行時のヘルプ・ファイルは、ファイル拡張子 `.sd1` で識別できます。



ヘルプ・システムは、ヘルプ固有ファイルのデスクトップでのアクションとデータ型を定義します。これにより、デスクトップから実行時のヘルプ・ファイルを簡単に処理および表示できます。

ヘルプタグ・ソフトウェア

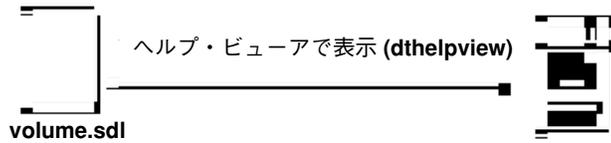
ヘルプタグ・ソフトウェアは、ファイル・マネージャでヘルプのソース・ファイルをダブルクリックすると自動的に起動されます。または、手動で `dthelptag` コマンドを端末ウィンドウで実行すると起動されます。

ヘルプタグには 2 つの重要なタスクがあります。

1. ヘルプタグのパーサは、マークアップされたファイルを、ヘルプ・システムが理解できる内部形式 (セマンティック・デリバリ言語) に変換します。マークアップにエラーがある場合は、**`volume.err`** というファイルにエラーが報告されます。
2. パーサのエラーがない場合はマスタのヘルプ・ボリューム・ファイル (**`volume.sdl`**) が作成されます。

ボリュームの表示

ヘルプタグでのソース・ファイルの処理が終わると、いつでもヘルプ・ボリュームを表示できます。表示するには、ファイル・マネージャで `volume.sdl` のファイル・アイコンをダブルクリックするか、端末ウィンドウで `dthelpview` コマンドを使用します。



アプリケーション用のヘルプを記述していて、そのアプリケーションが使用できる状態であれば、アプリケーションを実行してヘルプを呼び出せばヘルプを表示できます。

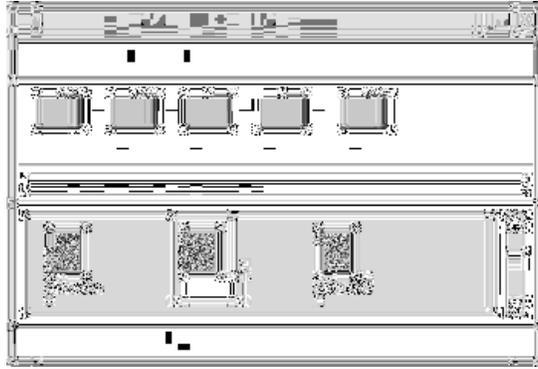
実行時のヘルプ・ファイルの作成

ヘルプタグを実行すると、*volume.htg* ファイル、*volume.ctg* ファイル、エンティティを使って指定された追加のソース・ファイルを読み取ります。また、使用するグラフィック・ファイルが存在するかどうかも確認します。

検索パスに `/usr/dt/bin/dthelptag` コマンドがあることを確認してください (確認の仕方がわからない場合は、システム管理者に問い合わせてください)。

▼ 実行時のヘルプ・ボリュームを作成するには

1. ファイル・マネージャを開き、ディレクトリを *volume.htg* ファイルのあるディレクトリに変更します。



2. ファイル・アイコンを選択します。
3. ファイル・マネージャの [選択] メニューで [コンパイル] を選択します。
volume.htg ファイルが処理され、*volume.sdl* ファイルと *volume.err* ファイルが作成されます。

ヘルプタグの出力

ヘルプタグの出力先は実行時のヘルプ・ボリュームで、名前は *volume.sdl* です。処理中に何らかのエラーが発生すると、エラー・ファイル (*volume.err*) に報告されます。エラーがない場合は、*volume.err* ファイルには著作権情報といくつかのステータス行が入ります。

helptag.opt ファイルに `onerror=go` オプションを設定すると、パーサはエラーを検出した後でも (可能であれば) 処理を続行します。`onerror=go` オプションを設定しないと、パーサは最初のエラーが検出されたところで停止します。*volume.sdl* ファイルは、ソース・ファイルにエラーがなくなるまでは作成されません。

ヘルプ・トピックを表示するために、*volume.sdl* ファイルとグラフィック・ファイルがヘルプ・システムに読み取られます。実行時のヘルプ・ファイル

は、*volume.htg* ファイルと同じベース名を持ちます。たとえば、*volume.htg* ファイル名が *Librarian.htg* の場合、ヘルプ・ボリューム名は *Librarian.sdl* です。



注意 - ヘルプタグ実行後は、実行時のヘルプ・ファイルまたはグラフィック・ファイル名を変更しないでください。*volume.sd1* ファイルに格納される情報は元の名前に基づいています。*volume.htg* ファイル名を変更する場合は、ヘルプタグを再実行してください。

▼ 手動で `dthelptag` コマンドを実行するには

- ◆ `dthelptag` コマンドを次のように実行します。

```
dthelptag command-options volume parser-options
```

command-options は *volume* 名の前に入力するオプションで、*parser-options* は *volume* 名の後に入力するオプションです。使用できるオプションは、212ページの「ヘルプタグ・ファイルの処理 (`dthelptag`)」にリストされています。

例: コマンド

次のコマンドは、MyVolume という名前のヘルプ・ボリュームを処理します。

```
dthelptag MyVolume
```

`-verbose` オプションを指定すると、処理の進み具合が画面に表示されます。

```
dthelptag -verbose MyVolume
```

検索パスを追加すると、ヘルプタグは `graphics` という名前の (現在のディレクトリの) サブディレクトリに格納されているファイルを検索できます。

```
dthelptag -verbose MyVolume search=graphics
```

例: `helptag.opt` ファイル

次の例は、オプションが1つずつ別の行にあるサンプル `helptag.opt` ファイルです。これは、ボリュームのドラフト版を作成するのに適した方法です。

```
memo
onerror=go
search=graphics/
search=entityFiles/
```

ヘルプ・ボリュームの最終版を作成する前に、`memo` と `onerror=go` の行を削除してください。

関連項目

- 第 13 章は、アプリケーション・インストール・パッケージにどのヘルプ・ファイルが含まれているかを説明しています。

▼ パーサ・エラーをレビューおよび修正するには

- ◆ ヘルプタグ実行後、*volume.err* ファイルの中身を見てください (*volume* は *volume.htg* ファイルのベース名です)。

volume.err ファイルにリストされている各エラーは、アスタリスクの文字列 (*****) で始まっています。たとえば、actions ファイルの 54 行目で次のエラーが検出されました。

```
*****
Line 54 of actions,
Missing end tag for LIST:
...the execution host becomes the current working directory.

<s2 id=EverythingYouNeedToKnow> E...
Current element is LIST begun on Line 28 of actions.
```

エラーの状況がわかるように、ファイルの内容が数行表示されています。また、その要素は同じファイルの 28 行目から始まる LIST だというヒントがあります。<s2> はリスト内では許可されないの、設計者が <\list> 終了タグを入力し忘れたようです。

1 つの単純なエラーがいくつかのエラー・メッセージを生成することがあります。これは、最初のエラーによってパーサが意図していたコンテキストを見失い、その後のマークアップを正しく解釈できなくなったためです。

一般的なエラー

処理エラーのほとんどは、以下の一般的なミスによるものです。

- 終了タグの付け忘れ
- 間違ったエンティティ名の使用
- 無効な要素 ID の参照

要素に終了タグを付け忘れるのはよくあるミスです。リスト、図、注、注意、警告などの要素を作成するときには、必ず終了タグを付けてください。特に、リスト中の図のように、ある要素を他の要素の中にネストした場合は、マークアップを丁寧にチェックしてください。

間違ったエンティティ名を使用するエラーもよく起こります。ほとんどの場合は単なるつづりの間違いです。その他は、エンティティ名が変更されたのに元の名前への相互参照が見落とされた場合です。エンティティ名を変更するときには、ソース・ファイルでそのエンティティ名のすべてのインスタンスを忘れずに検索してください。

同じように、要素に関連付けられた ID を変更すると、そのトピックへの相互参照やリンクに影響します。

ヘルプ・ボリュームの表示

ヘルプ・ビューアはどのヘルプ・ボリュームでも表示できます。ヘルプ・ビューアは、アプリケーション定義リンクを除くすべての型のハイパーリンクをサポートしています (アプリケーション定義リンクはどう解釈すべきかヘルプ・ビューアにはわからないからです)。

アプリケーション用ヘルプを記述していて、そのアプリケーションが使用できる状態であれば、アプリケーションを実行し、ユーザがするようにヘルプを呼び出せば、ヘルプを表示できます。

▼ ヘルプ・ボリュームを表示するには

1. ファイル・マネージャを開き、ディレクトリを *volume.sd1* ファイルのあるディレクトリに移動します。
2. アイコンをダブルクリックします。

デフォルトでは、ヘルプ・ビューアを使ってファイルが表示されます。

手動で `dthelpview` コマンドを実行するには

- ◆ 表示したいボリュームの *volume.sd1* ファイルが現在のディレクトリにあるか、または登録されている場合は、次のコマンドを実行します。

```
dthelpview -helpVolume volume.sd1
```

または、*volume.sd1* が他のディレクトリにある (そして登録されていない) 場合は、次のコマンドを実行します。

```
dthelpview -helpVolume /full-path/volume.sdl
```

上記のどちらのコマンドでも、`-helpVolume` パラメータは `-h` に省略できます。

例

ヘルプ・ボリュームを編集する時には、まず、ヘルプタグ・ソフトウェアで処理します。

```
dthelptag MyVolume
```

エラーがなかった場合は、次のコマンドで表示できます。

```
dthelpview -h MyVolume.sdl
```

関連項目

- 271ページの「アプリケーションとそのヘルプの登録」

例: 個人用ヘルプ・ディレクトリ

プロジェクト期間中に、自分が開発しているヘルプ・ボリュームにアクセスしたいけれども、システムのすべてのユーザに公開したくない場合があります。たとえば、作業ディレクトリが `/projects/help` で、ヘルプ・ボリュームが `Myvolume` という名前だとします。

まず、ボリュームを登録できるホーム・ディレクトリに個人用ヘルプ・ディレクトリを作成します。

```
mkdir -p $HOME/.dt/help/C
```

次に、(ヘルプタグ・ソフトウェアによって作成される) `Myvolume.sdl` ファイルへのシンボリック・リンクを作成します。

```
ln -s /projects/help/Myvolume.sdl $HOME/.dt/help/C/Myvolume.sdl
```

これで、(現在のディレクトリとは無関係に) ボリュームを次のコマンドで表示できます。これは、ホーム・ディレクトリ内の `.dt/help/C` ディレクトリは、ヘルプ・システムがヘルプ・ボリュームを検索するときに最初に見るからです。

```
dthelpview -helpVolume Myvolume
```

ブラウザ・ボリュームへのヘルプの追加

デスクトップは、ブラウザ・ボリュームと呼ばれる特別なヘルプ・ボリュームを提供しています。ブラウザ・ボリュームにはシステムで使用可能なヘルプ・ボリュームがリストされます。ブラウザ・ボリュームは、フロントパネルで [ヘルプ・ビューア] コントロールをクリックすると表示されます。

ブラウザ・ボリュームから各種のヘルプ・ボリュームを直接表示できます。これにより、アプリケーションを起動しなくてもアプリケーション固有のヘルプにアクセスできます。また、スタンドアロン・ヘルプを記述している場合は、ユーザーがそのヘルプを得られる唯一の方法です。

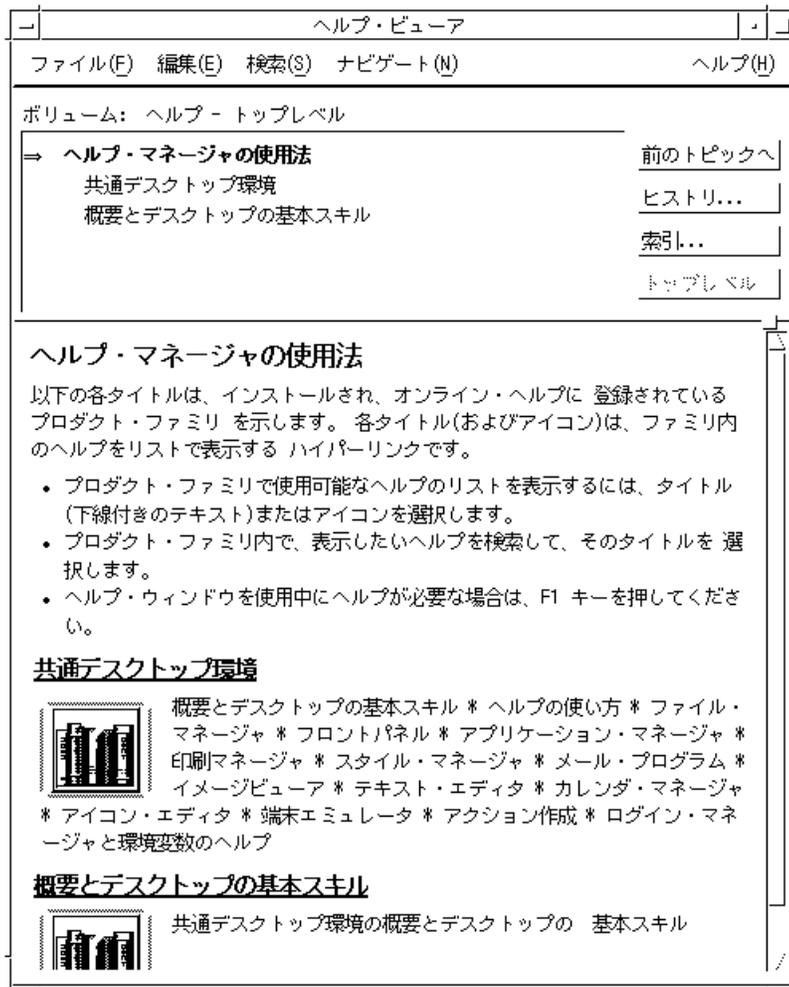


図 4-1 ヘルプ・ファミリを表示するブラウザ・ヘルプ・ボリューム

ヘルプ・ボリュームをブラウザ・ボリュームで使用できるようにするには、ヘルプ・ファミリ・ファイルを作成します。アプリケーションがデスクトップに登録されてファミリ・ファイルができると、ヘルプ・ボリュームがブラウザ・ボリュームの中に取り込まれます。

ブラウザ・ボリューム

デスクトップ・ユーティリティはブラウザ・ボリュームを作成および更新します。ユーザが初めてフロントパネルの [ヘルプ・ビューア] をクリックすると、こ

のユーティリティが自動的に実行されます。このユーティリティは、ヘルプ検索パス・ディレクトリにあるヘルプ・ボリュームとヘルプ・ファミリー・ファイルを識別します。このユーティリティは、`browser.hv` というファイルをユーザの `HomeDirectory/.dt/help/$DTUSERSESSION` ディレクトリに作成します。一度作成された後は、ボリュームは変更があった場合にのみ更新されます。

ブラウザ・ボリュームを手動で更新するには、216ページの「ブラウザ・ヘルプ・ボリュームの作成 (dthelpgen)」を参照してください。

ブラウザ・ボリュームにリストされているヘルプ・ボリュームはどれでも、ボリューム・タイトルを選択すれば表示できます。異なるボリュームから表示やナビゲートができるので、ブラウザ・ヘルプ・ウィンドウには [トップレベル] というボタンがあります。1つまたは複数のボリュームを表示した後に、このボタンを使ってブラウザ・リストに戻ることができます。

ヘルプ・ファミリー・ファイル

デスクトップ・ユーティリティは、どのヘルプ・ボリュームがブラウザ・ボリュームに集められているか識別するために、ヘルプ・ファミリー・ファイルを調べます。図 4-1 は、ブラウザ・ボリュームにリストされている「Common Desktop Environment (共通デスクトップ環境)」と「Overview and Basic Desktop Skills (概要とデスクトップの基本スキル)」という 2つのヘルプ・ファミリーを示します。各ファミリー・ファイルは、1つまたは複数の関連するヘルプ・ボリュームから成ります。たとえば、「Common Desktop Environment (共通デスクトップ環境)」ファミリーには、デスクトップを説明する異なったボリュームが含まれています。

アプリケーションとそのヘルプ・ファイルをデスクトップにインストールする方法の詳細は、『Solaris 共通デスクトップ環境 上級ユーザ及びシステム管理者ガイド』を参照してください。

▼ ヘルプ・ファミリーを作成するには

1. プロダクトに独自のファイル名を付けます。ヘルプ・ファミリーのファイルだと識別できるように拡張子 `.hf` を使用します。

```
family.hf
```

2. 次の行をファイルに入力します。

```
*.charSet:      character-set
*.title:        family title
```

```
*.bitmap:      icon file
*.abstract:    family abstract
*.volumes:    volume volume volume ...
```

character-set は、*family title* 文字列と *family abstract* 文字列が使用する文字セットです。サポートされている文字セットのリストについては、290ページの「フォント・スキーマの理解」を参照してください。*family title* と *family abstract* にはヘルプタグ・マークアップを指定できません。このファイルはヘルプタグ・ソフトウェアでは処理されません。

icon file はオプションです。指定する場合、ファイルの位置を指定するのに使用するパスは絶対パス名でなければなりません。アイコンを付けない場合は、ファミリー・ファイルに **.bitmap* リソースを指定しないでください。

volume 名のリストは、そのファミリーにどのボリュームが属すのかを識別します。ボリュームは、この行に表示される順番にリストされます。1つのボリュームが複数のファミリーにリストされることもあります。

いずれかの値が1行を超える場合、最後の行以外の各行を \ で終わらせます。

ファイル内で ! (感嘆符) で始まる行はコメント行であり、無視されます。

3. 最終的なプロダクトを準備するとき、*family.hf* ファイルと残りのヘルプ・ファイルをインストールしてください。デスクトップ統合スクリプト (*dtappintegrate*) は、実行されるとファミリー・ファイルへのシンボリック・リンクを作成します。

dtappintegrate スクリプトの実行方法は、『*Solaris* 共通デスクトップ環境 上級ユーザ及びシステム管理者ガイド』に説明されています。

例

次の例は、デスクトップのオンライン・ヘルプ用のファミリー・ファイルです。ファイルの最初にあるコメントで、ファミリーとリリース・バージョンが識別できます。

```
#####
!#                                     #
!#           Desktop  Help Family     #
!#                                     #
!#           Version 1.0              #
!#                                     #
#####
*.charSet:      ISO-8859-1
*.title:        Desktop Version 1.0
*.bitmap:       /usr/dt/appconfig/help/C/cdelogo.pm
*.abstract:     Overview and Basic Desktop Skills \
                * File Manager and the Desktop \
```

```
* Front Panel \  
* Application Manager \  
* Style Manager \  
* Text Editor \  
* Mailer
```

```
*.volumes: Intromgr.sdl Filemgr.sdl FPanel.sdl  
Appmanager.sdl Stylemgr.sdl  
Textedit.sdl Mailer.sdl
```

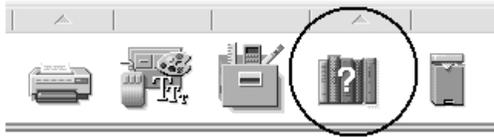
実際にデスクトップ・ソフトウェアに指定されているヘルプ・ファミリーは、この図のとおりではありません。

関連項目

- サポートされている文字セット名のリストは、280ページの「文字セットおよびマルチバイト文字」を参照してください。

▼ ブラウザ・ボリュームを表示するには

1. デスクトップのフロントパネルから [ヘルプ・ビューア] コントロールを選択します。



2. ヘルプ・ウィンドウをスクロールして、システムで使用可能なヘルプ・ファミリーを表示します。
3. ヘルプ・ファミリー・タイトルを選択してボリュームを表示します。

注 - ヘルプ・システムに関するヘルプ情報を参照するには、[デスクトップの紹介] を選択してから [ヘルプの使い方] を選択してください。

手動でブラウザ・ボリュームを表示するには

- ◆ 次のように `dthelpview` コマンドを実行します。

```
dthelpview -helpvolume browser
```

関連項目

- 215ページの「ヘルプ・トピックの表示 (`dthelpview`)」には `dthelpview` コマンド行がリストされています。
- `dthelpgen(1)` のマニュアル・ページ

ヘルプ・トピックの印刷

ヘルプ・ボリュームを表示した後、ヘルプ・トピックを印刷できます。図 4-2 のダイアログ・ボックスで、個々のトピック、目次および索引情報、ヘルプ・ボリューム全体を印刷できます。印刷出力ではグラフィックは省かれます。

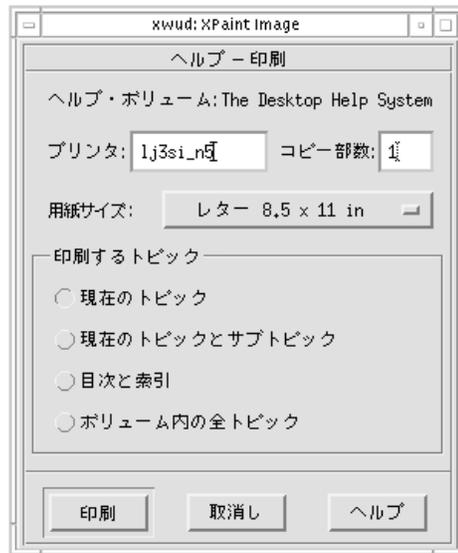


図 4-2 ヘルプの [印刷] ダイアログ・ボックス

ヘルプのテスト

ヘルプ・ボリュームのテストは、他のソフトウェア・プロダクトのテストと同じように重要です。テストを計画するのに役立つヒントをいくつか紹介します。

ハイパーリンクの妥当性テスト

- ヘルプ・ボリュームを表示して、すべてのハイパーリンクを試します。下線(実線または破線)の付いたテキストは、すべてハイパーリンクです。また、ハイパーリンクになっているすべてのグラフィックもテストします。グラフィック・ハイパーリンクは、ハイパーリンクの印として、イメージの周りに(破線または実線の)隅が開いている枠を使用しています。
- アプリケーション固有のヘルプを記述していて JumpNewView、Man、AppDefined リンクのいずれかを指定した場合は、これらのリンクをアプリケーションからテストしなければなりません。これらのリンク

を `dthelpview` を使ってテストしても、アプリケーション内から正しく動作する保証にはなりません。

エントリ・ポイントの確認

特定のヘルプ・トピックへのアクセスに ID を使用するアプリケーション固有のヘルプを記述している場合、ID がヘルプ・ボリューム内で正しく確立されているかを調べる方法は 2 つあります。

- アプリケーションを実行して、ユーザがするようにヘルプを要求してエントリ・ポイントを 1 つずつ確認します。これにより、アプリケーションが正しい ID を使用しているかも検査できます。
- アプリケーションが (開発中で) まだ使用できない場合は、各 ID に対して `dthelpview` を実行することで各 ID をテストできます。

```
dthelpview -helpVolume volume.sdl -locationId id
```

`id` は、テストしたい位置 ID です。`dthelpview` が正しいトピックを表示していれば、その ID は大丈夫です。

索引エントリのチェック

ユーザは、ヘルプ・トピックを見つけるためにヘルプ・ボリュームの索引を検索したりブラウズしたりします。あいまいな用語や重複したエントリがないように、索引エントリを丁寧に検査してください。また、各索引エントリを選択して、表示されるトピックが最も適切な情報かをチェックしてください。

グラフィックのテスト

- グラフィックが、カラー、グレースケール、白黒ディスプレイで表示可能なことを検査するために、物理的にアプリケーションをさまざまなディスプレイで表示します。
- デスクトップが使用するカラーの数を変更して、他のディスプレイをシミュレートすることもできます。そのためには、スタイル・マネージャを開き、[色の数] を選択し、異なるカラー・オプションを選択します。

パーサ・エラーのチェック

ヘルプ・ボリュームの開発中には、helptag.opt ファイルに onerror=go オプションを設定していると便利です。指定してある場合は、最終的にはこのオプションを削除してからソース・ファイルを処理し、エラーが1つも発生しないことを確認してください。

関連項目

- 216ページの「ブラウザ・ヘルプ・ボリュームの作成 (dthelpgen)」

ヘルプタグ・マークアップのリファレンス

この章では、ヘルプタグのマークアップ要素 (およびそれに指定するタグ) のすべてをアルファベット順に説明します。

マークアップ要素の解説

タグ名は使い方に応じて付けますが、それを決めるときに役立つよう、要素は次のように使い方でグループ化されています (1 つ以上のグループに表示される要素もあります)。

メタ情報 (ボリュームに関する情報)

- <metainfo>
- <title>
- <copyright>
- <abstract>
- <otherfront> (非階層トピック)

ヘルプ・ボリュームの構造

- <!entity>
- <helpvolume>
- <hometopic>
- <chapter>
- <s1>...<s9> (ヘッダ)
- <rsect>(参照セクション)
- <otherhead>
- <procedure>
- <p> (段落)

インライン要素

<book>
<computer> (簡易: ``text``)
<emph> (強調表示) (簡易: !!text!!)
<ex> (例) and <vex> (逐語例)
<image>
<keycap> (簡易: [[text]])
<lineno> (行番号)
<newline>
<p> (段落)
<quote> (引用)
<sub> (サブスクリプト) (簡易: text)
<super> (スーパースクリプト) (簡易: ^text^)
<term> (簡易: ++text++)
<user> (ユーザ入力)
<var> (変数) (簡易: %%text%%)
&...; (<!entity> 参照)

重要情報

<note>
<caution>
<warning>
<emph> (強調表示) (簡易: !!text!!)

リスト

<list>
<lablist> (ラベル付きリスト)
<item> (簡易: *)

グラフィック

<figure>
<graphic>

用語集と索引

<glossary>
<dterm> (用語定義)
<term> (簡易: ++text++)
<idx> (索引)

相互参照とハイパーリンク

<xref> (相互参照)
<link>
<location>
<term>

隠しテキスト

<!-- ... --> (コメント)
<memo>

タイトルとヘッダ

<abbrev>
<head>
<otherhead>
<procedure>

<title> (ヘルプ・ボリュームのタイトル)

ヘルプタグ・マークアップの無視

<vex> (逐語例)

<!-- ... -->

コメントヘルプタグ・ソフトウェアが無視するテキストです。コメントはネストされません。

形式

```
<!-- comment text here -->
```

コメント・テキストには2つの破線(--)以外の任意のテキストを指定できます。

例

次のマークアップは、コメントと図の両方を非表示にします。

```
<!-- Let's leave out this figure for now:
<figure entity=ProcessFlowChart>
Before and After Processing
<\figure>
-->
```

関連項目

- 176ページの「<memo>」

<abbrev>

省略タイトル

タイトルが長いトピックの代替ヘッダで、通常は短くした形式のタイトルです。省略タイトルを指定すると、[索引] ダイアログ・ボックスと [ヒストリ] ダイアログ・ボックスで完全なタイトルの代わりに使用されます。

ヘッダにグラフィック要素が入っている場合は、ヘッダのテキストだけがいった `<abbrev>` を指定してください。グラフィック・イメージはトピック・ツリーに表示されますが、[索引] ダイアログ・ボックスと [ヒストリ] ダイアログ・ボックスはグラフィック要素を表示できません。

`<abbrev>` にはマークアップ機能がありません。

形式

```
<topic-element> title  
<abbrev> short title
```

`topic-element` は `<hometopic>`、`<chapter>`、`<s1>`、または新しいトピックとなる任意の要素です。

`<abbrev>` タグは、ヘッダの直後の行に表示してください。

終了タグは必須ではありません。

例

簡単な例を示します。

```
<chapter> Ways of Treating Headings that are Too Long  
<abbrev> Long Headings
```

トピックのタイトルをヘルプ・トピック表示領域には表示したくないが、トピック・ツリーにはタイトルを表示したいとします。この場合、次のようにマークしてください。

```
<chapter> &empty;  
<abbrev> chapter title
```

関連項目

- 142ページの「`<chapter>`」

<abstract>

概要

ヘルプ・ボリュームの簡単な説明です。

形式

```
<metainfo>
  .
  .
  .
  <abstract>
  abstract text here ...
  <\abstract>
  .
  .
  .
<\metainfo>
```

概要はマークアップを認識しないアプリケーションが読んだり表示したりするため、概要のテキストにはヘルプタグ・マークアップを入れません。

<abstract> 要素は自動的に ID 文字列 `_abstract` に割り当てられます。設計者が定義した ID は割り当てられません。`_abstractID` は <link> 要素と共に使用できますが、<xref> 要素とは使用できません。

概要のテキストにはオプションの <head> があります。

例

次のマークアップは、ヘルプ・ボリュームの内容を簡単に記述しています。

```
<abstract>
Online help for the Application Manager Version 1.0.
<\abstract>
```

注

要素を <metainfo> 要素内でリンクさせるときは、必ず `type=Definition` リンクにしてください。

次の例は概要をリンクさせる方法を示します。

```
<link hyperlink= "_abstract" type=Definition>
Choose this link for an abstract.<\link>
```

関連項目

- 177ページの「<metainfo>」
- 156ページの「<head>」

<<annotation text>>

注釈

例 (<ex> タグ) の中で説明補足やコメントを提供します。

形式

```
<ex [side | stack]>  
text of the example ...<<annotation text >>  
<\ex>
```

side デフォルトです。例の 1 行目のテキストの右側に注釈を入れます。

stack テキストの下に注釈を入れます。

注釈のテキストは、<< *this is the annotation text* >> のように二重三角括弧で囲ってください。<ex> タグでのみ注釈が使用できます。<ex> タグの **side** パラメータおよび **stack** パラメータは、テキストに関連する注釈の位置を指定できます。

注釈に空白行を入れるには、スペースの後に空の注釈、つまり空白 <<>> を入れてください。

例

次のマークアップは注釈のデフォルト **side** の配置方法を使用します。

```
<ex>  
Login: <<Enter your name>>  
<\ex>
```

出力は次のとおりです。

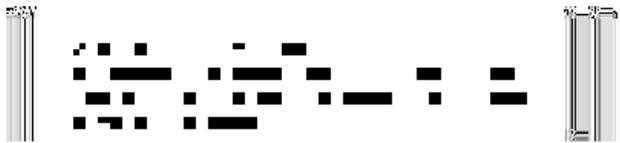
Login: *Enter your name*

次のマークアップは **stack** の配置方法を使用して長い注釈を入れます。

```
<ex stack>
Quarterly Sales Reports

<<Q1: January, February, March Q2: April, May, June Q3: July, August,
September Q4: October, November, December>>
<\ex>
```

出力は次のとおりです。



<book>

本のタイトル

本のタイトルを指定します。

形式

```
<book>book title<\book>
```

または

```
<book|book title|
```

ヘルプタグ・フォーマットではイタリックで本のタイトルを示します。日本語では、イタリック書体のフォントが提供されていない場合、それに代わる書体、たとえばボールドで表示されます。

例

次の 2 種類があります。

```
Refer to <book>The Elements of Style<\book>
for further details.
```

または

Refer to `<book|The Elements of Style|`
for further details.

出力は次のとおりです。

Refer to *The Elements of Style* for further details.

<caution>

注意の通知

ユーザにデータの損失か破損の可能性あることを警告します。

形式

```
<caution>  
text of caution<\caution>
```

デフォルト・ヘッダは「**Caution**」(日本語ロケールでは「**注意**」)です。別のヘッダを指定するには、次のように `<head>` タグを使用します。

```
<caution><head>alternate heading  
text of caution  
<\caution>
```

`<\caution>` 終了タグが必須です。

アイコンを、注意と共に表示されるように指定するには、ファイル・エンティティを次のようにヘルプ・ボリュームの最上部に定義します。

```
<!entity CautionElementDefaultIconFile FILE "filename">
```

filename はアイコン・グラフィック名です。cautions.pm という例の注意アイコンは /usr/dt/dthelp/dthelptag/icons ディレクトリにあります。

例

次の例は注意メッセージです。

```
<caution>  
There is no Undo for this selection. Before performing this task,
```

save any changes to your document.
`<\caution>`

上記のマークアップの出力は次のとおりです。



関連項目

- ヘッダの変更の例を示す 164 ページの「179ページの「`<note>`」
- 196ページの「`<warning>`」
- 152ページの「`<figure>`」
- 156ページの「`<head>`」

`<chapter>`

章

新しいタイトルの新しいトピックの最初を示します。

形式

```
<chapter id=id>title
topic text ...
```

終了タグは必須ではありません。

トピックのタイトルが長い場合、代わりに `<abbrev>` で省略タイトルを指定できます。短いタイトルは [索引] ダイアログ・ボックスと [履歴] ダイアログ・ボックスで使用されます。タイトルにグラフィック要素があれば、テキストだけのタイトルで `<abbrev>` を作成してください。

例

次の 2 つは新しいトピックの開始を示すマークアップ例です。

```
<chapter>A Manual of Style  
<chapter id=DesktopTools>Desktop Tools
```

関連項目

- 137ページの「`<abbrev>`」
- 169ページの「`<link>`」
- 186ページの「`<rsect>`」
- 187ページの「`<s1>...<s9>`」
- 197ページの「`<xref>`」

<computer>

コンピュータ・リテラル

コンピュータの入出力を示すテキストを表示します。

形式

```
<computer>text<\computer>
```

または

```
``text``
```

簡易形式では 2 つの左アポストロフィ (‘) と 2 つの右アポストロフィ (’) を使用します。

例

- マークアップは次のとおりです。

```
<computer>Enter the correct numerical value.<\computer>
```

出力は次のとおりです。

```
Enter the correct numerical value.
```

- 次のマークアップは簡易形式を使用しています。

```
Everything in ``computer`` comes out looking ``like this.``
```

出力は次のとおりです。

```
Everything in computer comes out looking like this.
```

- 変数はコンピュータ・テキストで使用できます。例を示します。

```
``void DisplayTopic (%%topic%%);``
```

出力は次のとおりです。

```
void DisplayTopic (topic);
```

関連項目

- 150ページの「<ex>」
- 193ページの「<user>」
- 194ページの「<var>」

<copyright>

著作権の告知

著作権を告知します。

形式

```
<metainfo>  
  <title>Title (always before copyright)  
  <copyright>  
  &copy; Copyright notice here ...
```

この要素は <metainfo> セクションでのオプションです。使用する場合は、<title> 要素の後に入力してください。

終了タグは必須ではありません。

あらかじめ定義されたエンティティ © は、著作権記号 (©) を生成します。

例

次のマークアップはヘルプ・ボリウムにタイトルを付けて、著作権情報を提供します。

```
<metainfo>
<title>XYZ World Almanac
<copyright>
&copy; Copyright 1995 XYZ Company. All rights reserved.
```

出力は次のとおりです。

© Copyright 1995 XYZ Company. All rights reserved.

関連項目

- 177ページの「<metainfo>」
- 192ページの「<title>」

<dterm>

用語の定義

用語集にある用語および用語の定義を識別します。

形式

```
<glossary>
  <dterm>first term
  definition of first term
  .
  .
  <dterm>Nth term
  definition of Nth term
```

この要素は <glossary> セクションで使用されます。

<dterm> タグの後に用語名が続く形で、1行で表示されます。用語の定義は <dterm> タグに続く行から始まります。

終了タグは必須ではありません。

例

次のマークアップは、用語集で最初の2語を定義します。

```
<glossary>
```

```
<dterm>algorithm  
A mathematical rule or procedure for solving a problem.
```

```
<dterm>click  
To press and release a mouse button.
```

関連項目

- 154ページの「<glossary>」
- 190ページの「<term>」

<emph>

テキストの強調表示

テキストを、注意を促すフォントにします。

形式

```
<emph>text<\emph>
```

または

```
!!text!!
```

<emph> 要素の簡易形式は、テキストの前後に二重の感嘆符 (!!)を付けます。

開始タグ <emph> を使用した場合は、終了タグ <\emph> が必須です。

例

次の2つのマークアップのうちいずれかを使用します。

```
A thousand times <emph>no<\emph>.  
A thousand times !!no!!.
```

出力は次のとおりです。

```
A thousand times no.
```

関連項目

- 140ページの「<book>」
- 194ページの「<var>」

<!entity>

エンティティの宣言

エンティティ名を文字列または外部ファイルに指定します。

形式

```
<!entity entityname "string">
```

または

```
<!entity entityname FILE "filename">
```

エンティティ名は64字以下の英字、数字、ハイフンで指定できます。大文字小文字の区別はエンティティ名については重要ではありませんが、設計者が読みやすいうように使い分けることが多いです。先頭の文字は英字で始めなければなりません。<!entity> 宣言の中の< (左三角括弧)、!(感嘆符)、entityの間には、スペースを入れないでください。

エンティティの宣言は、常に他のマークアップまたはヘルプ・ボリュームのテキストより前に入れなければなりません。

定義したエンティティを表示させるときは、次の形式を使用するエンティティの参照を挿入します。

`&entityname;`

エンティティの参照は & (アンド記号) の後にエンティティ名 (エンティティの宣言で定義) が続き、最後に ; (セミコロン) が付きます。

エンティティの目的

エンティティは、次の4つの理由により定義します。

- エンティティ名で指定されたテキストは一度しか表れないため、テキストを変更するときは1箇所変更するだけで済みます。ヘルプタグがファイルを再処理するとき、エンティティの参照はすべて自動的に変更されます。
- テキストが何度も表示される場合でも、略式のエンティティの参照を入力すればよいので、長く複雑なテキスト文字列を繰り返し入力するという無駄な労力 (あるいは入力ミス) を省くことができます。完全なテキスト文字列を入力するのは一度だけで済みます。
- `<figure>` 要素および `<graphic>` 要素にファイル名は適用できません。図を含むファイル名は、エンティティの宣言で指定してください。
- ヘルプ・テキストを複数のファイルに入れると便利ですが、ヘルプタグは1つのソース・ファイルしか受け入れません。実際のヘルプ・テキストを含むファイルを参照するエンティティの宣言とエンティティの参照を含むファイルを1つ作成することによってこの矛盾を是正できます。

例

- `volume.htg` ソース・ファイルには次のようなエンティティの宣言およびエンティティの参照が含まれているので、実際のテキストが指定したファイルに挿入されます。

```
<!entity topic1 FILE "topic1">
<!entity topic2 FILE "topic2">
<!entity topic3 FILE "topic3">

&topic1;
&topic2;
&topic3;
```

- 次のエンティティの宣言にある「Architectural Analysis of Aircraft Precision Components」という語は、マークアップ・ファイルのエンティティの参照では `&apc;` と表示されます。

```
<!entity apc "Architectural Analysis of Aircraft Precision Components">
```

- 次のような図に対するエンティティの宣言は、ソース・ファイルの先頭に置かれます。

```
<!entity CloseUpFig FILE "figname.tif">
```

図は次のマークアップの場所に挿入されます。

```
<figure entity=CloseUpFig>  
Close Up View  
<\figure>
```

関連項目

- 72ページの「エンティティの使い方」
- 152ページの「<figure>」
- 197ページの「<xref>」
- 第6章

<esc>

エスケープ

テキストがヘルプタグで解釈されずに、実行時のヘルプ・ファイルへ直接渡されます。たとえばカスタマイズされたアプリケーションでは、設計者がヘルプ・ソース・ファイルにSDL (セマンティック・デリバリ言語) マークアップを埋め込むことができます。<esc> 要素があると、ヘルプタグ・パーサはSDL マークアップを読みません。ヘルプ・ボリュームがヘルプ・ビューアで表示されると、設計者が入れたSDL マークアップが処理されます。

個々のヘルプタグ記号やマークアップ例をエスケープするときは <esc> タグを使用しないでください。< (左三角括弧) や \ (バックスラッシュ)、& (アンド記号) などのヘルプタグ記号を表示するには、各記号の前にアンパサンドを付けます。<vex> 要素を入れると、ヘルプ・ボリュームにヘルプタグ・マークアップ例を入れることができます。

形式

```
<esc>text<\esc>
```

または

<esc|text|

注・正規の形式を使用する場合、テキストには <\x という 3 文字の文字列 (記号、バックスラッシュ、文字の順) は入りません。略式を使用する場合はテキストには | (縦線) は入りません。

前者の形式を使用するときは <\esc> 終了タグが必須です。

関連項目

- 54ページの「ヘルプタグ記号の表示」
- 195ページの「<vex>」

<ex>

コンピュータの例

スペースまたは行の切れ目を変更せずにテキストを表示します。

形式

<ex [nonumber | number] [smaller | smallest] [side | stack]>
example text here ...<\ex>

nonumber	(デフォルト) 各行頭に入る行番号を省略します。
number	各行頭に行番号を付けます。
smaller	より小さいフォントを使用して例を表示します。
smallest	最も小さいフォントを使用して例を表示します。長い行のものを狭い幅の中に表示します。
side	例に注釈を入れるときだけ適用します。テキストに関連する注釈の位置を指定します。デフォルト位置は side で、テキストの 1 行目の右側に注釈が表示されます。
stack	テキストの下に注釈が表示されます。

例は、固定幅の文字で左揃えで表示されます。

number 属性を入れると、行番号が付けられます。特定の行を参照するときに便利です。

次に示す例は、他のコンテキストでは特殊な意味を持つ文字ですが、<ex> では通常のテキストとして扱われます。

```
!! 二重感嘆符
-- 二重マイナス記号
++ 二重プラス記号
" 二重引用符
```

<\ex> 終了タグが必須です。

例

マークアップは次のとおりです。

```
<ex>
Examples are printed in computer
font. Line breaks are preserved.
<\ex>
```

出力は次のとおりです。



関連項目

- 143ページの「<computer>」
- 193ページの「<user>」
- 195ページの「<vex>」

<figure>



グラフィック・イメージを挿入します。

形式

```
<figure entity=entity [id=id [nonumber | number=n]  
[left | center | right] [cappos=[capleft | capcenter | capright]]  
[ghyperlink=id [glinktype=type] [gdescription=text ]]] >  
caption string<\figure>
```

entity=<i>name</i>	挿入するグラフィック・イメージを含むファイルを識別するファイル・エンティティを指定します。
id=<i>name</i>	オプションで、図の相互参照に使用する ID 名を定義します。
nonumber	オプションで、「Figure」(日本語ロケールでは「図」) という語と図番号は自動的に生成されません。
number=<i>n</i>	オプションで、自動的に生成された図番号を無効にし、指定された番号を使用します。
left、center、right	現在のページ幅の範囲内で、イメージの水平位置を揃えるよう指定します。
cappos=<i>position</i>	capleft、capcenter、capright の値で、タイトルの水平位置を揃えるよう指定します。タイトルはオプションです。
ghyperlink="<i>id</i>"	オプションで、ハイパーリンクにしたい図のグラフィカルな部分を指定します。<link> 要素の <i>hyperlink</i> 属性と同じように使用します。指定した <i>id</i> によって位置を参照します。
glinktype=<i>type</i>	オプションで、ハイパーリンクの種類を指定します。デフォルトは Jump です。他には JumpNewView、Definition、Man、Execute、

AppDefined があります。このパラメータを使用するときは `ghyperlink` パラメータと `id` 値が必要です。`<link>` 要素の `type` 属性と同じように使用します。

`gdescription="text"`

オプションで、ハイパーリンクの記述を提供します。このパラメータを使用するときは `ghyperlink` パラメータと `id` 値が必要です。

`<\figure>` 終了タグが必須です。

外部グラフィック・ファイルをヘルプ・トピックに統合するには、そのグラフィックのファイル名をエンティティ名に対応付けるようエンティティを宣言してください (`<!entity entityname FILE "filename ">`)。

例

- 次のマークアップは、指定したタイトルの付いたグラフィックを挿入し、自動的に図番号を付けます。

```
<!entity MapFigure FILE "worldmap.xwd">
.
.
.
<figure entity=MapFigure>
Caption for Figure
<\figure>
```

- 次のマークアップは、タイトルなしで図番号の付いた図を挿入します。

```
<!entity StateMap FILE "oregon.xwd">
.
.
.
<figure entity=StateMap>
<\figure>
.
.
.
```

- 次のマークアップは、指定した図番号とタイトルの付いた図を挿入します。タイトルは \ (バックスラッシュ) のところで 2 行に分かれます。

```
<figure number=99 entity=SchemDiag>
Schematic that Illustrates\the Overall System Design
<\figure>
```

関連項目

- 147ページの「<!entity>」
- 155ページの「<graphic>」
- 169ページの「<link>」
- 197ページの「<xref>」
- 100ページの「実行別名」では、実行リンクの使い方を説明しています。

<glossary>

用語集

要素でマークされたすべての用語の定義を含む用語集セクションを開始します。

形式

```
<glossary>
<dterm>first term
  definition of first term can continue over multiple lines or paragraphs

  <dterm>second term
  definition of second term ...
  .
  .
  .
```

「Glossary」(日本語ロケールでは「用語集」)は、用語集セクションのヘッダとして自動的に使用されます。

<dterm> 要素は各用語とその定義を識別します。

nogloss パラメータなしで <term> でマークした用語はすべて用語集に入れなければなりません。その語が用語集にない場合、入っていない用語はヘルプタグを実行するときに作成される *volume.err* ファイルにリストされます。

終了タグは必須ではありません。

例

次の例は、2つの定義を持つ簡単な用語集です。

```
<glossary>
  <dterm>oxymoron
  A combination of contradictory words.
<dterm>veritable
  Being in fact the thing named. Authentic.
```

関連項目

- 190ページの「<term>」
- 145ページの「<dterm>」

<graphic>

インライン・グラフィック

テキストの行にグラフィカル要素を挿入します。

形式

```
<graphic entity=name [id=id]>
```

name エンティティの宣言で定義されているエンティティ名。この宣言により、挿入するグラフィックを含むファイル名にエンティティ名が対応付けられます。

id=*name* オプションで指定します。図の相互参照に使用する ID 名を定義します。

<graphic> 要素は <figure> に似ていますが、<figure> が段落間に図を挿入するのに比べて、<graphic> ではテキスト間に小さな図を埋め込む点が異なります。

例

- 次のマークアップは、最初にグラフィック・ファイル (mini.pm) の内容に対応するエンティティ (mini-icon) を定義します。次に、グラフィックを入れる位置をテキスト行で指定します。

```
<!entity mini-icon FILE "mini.pm">
  .
  .
```

The `<graphic entity=mini-icon>` icon is used to represent very small images.

- 次のマークアップは、最初に `mini-icon-topic` という ID を持つトピックを定義します。ハイパーリンクとしてのインライン・グラフィックの使い方を次に示します。

```
<sl id=mini-icon-topic>When you click on the inline graphic, it
will bring you to this topic.
.
.
.
The <link mini-icon-topic> <graphic entity=mini-icon> <\link>
icon is to represent very small things.
```

関連項目

- 147ページの「`<!entity>`」
- 152ページの「`<figure>`」
- 169ページの「`<link>`」
- 182ページの「`<p>`」

`<head>`

ヘッダの挿入

通常はタイトルを付けない要素

(`<abstract>`、`<paragraph>`、`<list>`、`<otherfront>` など) や、デフォルトのタイトルが付いている要素 (`<note>`、`<caution>`、`<warning>` など) にタイトルを付けます。

形式

```
<element><head>title text
```

ヘッダは、`<head>` タグとタグの後の空白以外の文字とで始まります。`<head>` タグはヘッダを追加する行かそれ以下の行に表示します。

<head> 要素はタイトルと見なされる要素と共に使用できますが、この場合は <head> を付ける必要はありません。

ヘッダ領域より長いヘッダは、自動的に次の行に表示されます。強制的に行の切れ目を入れるには、該当する場所に \ (バックスラッシュ) を入れてください。

& (アンド記号) で終了させない限り、ヘッダはソース・ファイルの行末で終了します。ヘッダがソース・ファイルの中で複数行にわたる場合は、最終行以外の各行の最後にアンド記号を入れてください。

終了タグは必須ではありません。

例

- 次のマークアップは、タイトルをリストに追加し、 \ (バックスラッシュ) の表示された場所で新しい行が開始するよう指定します。

```
<list><head>Printing Options\for the QRZ Hardware
```

出力は次のとおりです。



- 次のマークアップは、デフォルトの「Note」(日本語ロケールでは「注」)のヘッダを無効にし、指定されたヘッダに置き換えます。

```
<note><head>Tips and Shortcuts  
Keyboard menu accelerators provide quick access to menu commands.  
<\note>
```

出力は次のとおりです。



関連項目

- 138ページの「<abstract>」
- 141ページの「<caution>」
- 161ページの「<image>」
- 165ページの「<lablist>」
- 174ページの「<location>」
- 179ページの「<note>」
- 180ページの「<otherfront>」
- 182ページの「<p>」
- 196ページの「<warning>」

<helpvolume>

アプリケーション・ヘルプ・ボリューム

「ルート」構造要素で、全ヘルプ・ボリュームのマークアップすべてが含まれています。

形式

```
all entity declarations
.
.
.
<helpvolume>
.
.
.
all of your help is included here, either
literally or using file entity references
.
.
.
<\helpvolume>
```

このタグを入力しない場合は、ヘルプタグ・ソフトウェアが自動的にタグがあるものと見なして処理します。

エンティティの宣言はすべて、<helpvolume> 開始タグの前になければなりません。

関連項目

- 138ページの「<abstract>」
- 54ページの「ヘルプ・ボリュームの例」
- 147ページの「<!entity>」
- 159ページの「<hometopic>」
- 177ページの「<metainfo>」

<hometopic>

「ホーム」ヘルプ・トピックまたは最上位のヘルプ・トピック
最上位ヘルプ・トピックの始まりを示します。

形式

```
<hometopic>heading  
topic text begins here ...
```

ヘルプ・ボリュームのホーム・トピックは1つだけです。これは、メタ情報 (<metainfo>) の後で、最初の <chapter> または <s1> の前に付きます。

<hometopic> 要素は設計者が定義した ID をサポートしません。ヘルプタグ・ソフトウェアが定義済み ID の `_hometopic` を割り当てます。

ハイパーリンクをホーム・トピックに作成する形式は次のとおりです。

```
<link hyperlink= "_hometopic">...<\link>.
```

例

```
<hometopic>Welcome to Online Help  
This is the home topic for the online help ...
```

```
<chapter>First Subtopic  
This is the first subtopic ...
```

```
<chapter>Second Subtopic
```

This is the second subtopic ...

⋮

関連項目

- 54ページの「ヘルプ・ボリュームの例」
- 169ページの「<link>」
- 64ページの「ホーム・トピックを作成するには」
- 177ページの「<metainfo>」

<idx>

索引エントリ

ヘルプ・ボリューム索引に表示するエントリを定義します。

形式

`<idx>text<\idx>`

または

`<idx|text|`

または

`<idx>text<sort>sort key<\idx>`

text

キーワード索引に表示されるテキスト文字列です。

sort key

索引をソートするときに使用する任意のテキスト文字列です。キーワード索引の *text* が表示される場所に影響を与えます。*sort key* 文字列はキーワード索引には表示されません。

一般的なヘルプ・ダイアログ・ボックスの [索引] ボタンを選択すると、[索引検索] ダイアログ・ボックスが表示されます。索引エントリをヘルプ・トピックに追加すると、ユーザが索引の語句を検索して目的のヘルプを見つけるのに役立ちます。

開始タグと終了タグ `<idx>`、`</idx>`、または略式マークアップ `<idx|...|>` が使用できます。

`<sort>` 要素は索引エントリをソートする順序を変更します。特に、`<idx>` 要素で使用すると、*sort key* 文字列で示された場所にキーワードが表示されます。終了タグは必須ではありません。

例

- 次のマークアップは、一部の索引エントリの定義を略式で記述しています。索引エントリはソース・テキストを読みやすくするためにインデントしています。

```
A portable personal computer has a full-sized keyboard, built-in
disk drives and a detachable LCD screen.
<idx|keyboard|
<idx|disk drive|
<idx|screen, LCD|
<idx|personal computer, portable|
<idx|portable, personal computer|
```

- 次の例では索引に「+」が表示されますが、エントリのアルファベット順リストで「plus」が表示される場所に表示されます。

```
<idx>+<sort>plus<\idx>
```

<image>

イメージ表示

ソース・テキストと同じように改行が入ったテキストを表示します。

形式

```
<image [indent] [id=id] [gentity=graphic-ent [gposition=pos] [ghyperlink=gid
[glinktype=type]]]>text <\image>
```

`indent`

オプションで、パラグラフを現在の左マージンからスペース 6 つ分だけインデントするよう指定します。

id=id	オプションで、この位置を相互参照するのに使用する ID 名を定義します。
gentity=graphic-ent	オプションで、テキストに囲まれるグラフィックのエンティティ名です。gposition、ghyperlink、glinktype パラメータのいずれかを使用する場合は、gentity パラメータと <i>graphic-ent</i> 値は必須です。
gposition=pos	オプションで、left または right は、任意のグラフィックをそれぞれ左揃えまたは右揃えにします。
ghyperlink=gid	オプションで、グラフィックをハイパーリンクとして指定し、ハイパーリンクの宛先を指定します。glinktype パラメータを使用する場合は、ghyperlink パラメータと <i>gid</i> 値は必須です。<link> 要素の <i>hyperlink</i> 属性と同じように使用します。(イメージ・テキストの位置を参照するときは <i>gid</i> 値ではなく <i>id</i> 値を使用します。)
glinktype=type	オプションで、ハイパーリンクの種類を指定します。デフォルトは Jump です。他には、JumpNewView、Definition、Man、Execute、AppDefined があります。<link> 要素の <i>type</i> 属性と同じように使用します。
text	グラフィックのまわりを囲むパラグラフのテキストです。

<image> タグと <\image> タグとの間のテキストは、実際のテキストと同じスペース、同じインデント、同じ改行で表示されます。均等割付、ワード・ラップ、空白行の削除は行われません。ただしプロポーショナル・フォントを使用すると、画面上ではきちんと並んでいたテキストのカラムがヘルプ情報ではそのとおりにならない場合もあります。表示されたテキストの幅が大きすぎて表示領域に収まらないときは自動的に水平スクロール・バーが表示されます。

すべてのインライン・テキスト要素と特殊文字が認識されます。

任意の <head> を <image> で使用できます。<xref> を使用する要素を相互参照する場合は <head> タグが必須です。

`indent` パラメータを使用すると、表示されたテキストが左マージンからインデントされます。

開始タグと終了タグ (`<image>` と `<\image>`)、または簡易形式マークアップ (`<image|...|`) を使用できます。

関連項目

- 150ページの「`<ex>`」
- 195ページの「`<vex>`」
- 182ページの「`<p>`」
- 100ページの「実行別名」では、実行リンクの使い方を説明しています。

`<item>`

リスト項目

リストの項目を示します。

形式

```
<list [id=id]>
  * List item
  * List item
<\list>
```

または

```
<list order>
  <item id=name1>List item
  <item id=name2>List item
  <item id=name3>List item
  .
  .
  .
<\list>
```

通常 * (アスタリスク) を付けた簡易形式で使用されます。

正規の形式では、リストの項目を相互参照できます。これができるのは順序が付いた (番号が付けられた) リストの項目についてのみです。自動的に割り当てられた項目番号が、相互参照テキスト (ヘルプタグが `<xref>` 要素と置き換える) で使用され

ます。番号とは違い、黒い丸の行頭文字は相互参照テキストで置き換えても意味がありません。

関連項目

- 172ページの「<list>」
- 156ページの「<head>」
- 197ページの「<xref>」

<keycap>

キーボード・キー

キーボードのキーを示します。

形式

`<keycap>keycap characters<\keycap>`

または

`[[keycap characters]]`

略式では、[[(二重左角括弧) と]](二重右角括弧) をキーの文字の前後に付けます。

矢印など、特殊記号文字に対するエンティティの参照を使用できます。行をまたがってキーの文字を使用することはできません。

例

マークアップは次のとおりです。

Press [[Control]] + [[Home]] to go to the beginning of your document.

出力は次のとおりです。



関連項目

- 172ページの「<list>」
- 156ページの「<head>」
- 197ページの「<xref>」

<lablist>

ラベル付きリスト

左カラムにラベルが表示され、右カラムに(ラベルが指定する)項目が表示される形式のラベル付きリストを表示します。

形式

```
<lablist [loose | tight][wrap | nowrap]>  
[ <labheads> \Heading 1 \ Heading 2 ]  
 \label\ text for the first item  
 \label\ text for the second item  
 .  
 .  
 .  
<\lablist>
```

loose	デフォルトで、リストの項目間で行送りをします。
tight	リストの項目間で行送りをしません。
wrap	デフォルトで、長いラベルを複数行に改行させます。
nowrap	ラベルを複数行に分割されないようにします。

バックスラッシュ (\) はラベルの最初と最後を示します。先頭と末尾のスペースは無視されます。長いラベルは `nowrap` を使用しない限り複数行に分割されます。あらかじめ定義された文字エンティティ (`&sigspace;`) は、ラベルに非分割スペースを挿入するのに使用できます。

ラベル付けされた項目のテキストは 2 番目のバックスラッシュの後で、その行かそれ以下の行に入ります。項目の最後は次のいずれかで示されます。

- 空白行
- 他のラベル付き項目の最初
- `<\lablist>` 終了タグ

ラベル付き項目が 1 つ以上のパラグラフから構成されている場合、パラグラフ間に空白行を入れてください。ラベル付きリストの最後は、必須の `<\lablist>` 終了タグで示されます。

カラムごとに 1 つ付いている任意のカラム・ヘッダは、`<labheads>` タグの直後に (同じ行に) 入ります。カラム・ヘッダは \ (バックスラッシュ) で分割されます。終了タグ `<\labheads>` は必須ではありません。ただし、`<lablist>` の終了タグは必須です。

例

マークアップは次のとおりです。

```
<lablist tight>
<labheads> \ Unit \ Meaning
  \in\ inches
  \mm\ millimeters
  \cm\ centimeters
<\lablist>
```

出力は次のとおりです。

Unit	Meaning
in	inches
mm	millimeters
cm	centimeters

次のマークアップは、長いラベルを複数行に分割します。

関連項目

- 156ページの「<head>」
- 172ページの「<list>」

<lineno>

行番号

例の中の指定された行へ相互参照します。

形式

```
<ex number>
  example text <lineno id=name>
  .
  .
  .
<\ex>
```

この要素は、番号が付けられた例でのみ使用されます。参照したい行の最後に <lineno> タグを入力します。id パラメータは、相互参照の作成に使用できる ID を行番号に割り当てます。

例

このマークアップは番号が付けられた例で、3 行目への相互参照が記述されています。

```
<ex number>
  Enter Daily Account Total
  Run Invoice Summary Report
  Go to Monthly Ledger <lineno id=ledger>
  Run Daily Update
<\ex>
.
.
.
  To run closing reports, return to <xref ledger> and run the Past Due Accounts Report.
```

<xref ledger> 相互参照は ID が入っている行番号に置き換えられます。次の行が出力されます。

To run closing reports, return to 3 and run the Past Due Accounts Report.

<lineno> の終了タグは必須ではありません。

関連項目

- 150ページの「<ex>」

<link>

ハイパーリンク

テキストまたはインライン <graphic> をハイパーリンクとして使用されるように指定します。

形式

```
<link hyperlink [type] ["description"]>text</link>
```

または:

```
<link hyperlink= "hyperlink" [type=type] [description= "description"]>
```

hyperlink 属性は必須で、宛先やリンクの動作を識別する値です。標準の「jump」リンクの場合、*hyperlink* はジャンプしたい要素の ID です。

type パラメータは次の値を取ることができます。

Jump	デフォルトで、ID <i>hyperlink</i> があるトピックへジャンプします。
JumpNewView	ID <i>hyperlink</i> があるトピックへジャンプしますが、アプリケーションが新規ウィンドウでトピックを表示するように要求します。
Definition	一時ポップアップ・ウィンドウに、ID <i>hyperlink</i> があるトピックを表示します。
Execute	<i>hyperlink</i> 文字列をコマンドとして実行します。
Man	man コマンドへのパラメータとして <i>hyperlink</i> 文字列を使用してマニュアル・ページを表示します。

開始タグと終了タグの間の *text* は、リンクを起動するためにユーザが選択する「ホット・スポット」になります。ハイパーリンクとして使用される語句には、表示のとき下線が引かれます。大文字と小文字の区別は、*hyperlink* や *type* の値では意味はありません。

コマンドを実行するハイパーリンクは、実行リンクと呼ばれます。実行されるコマンドは `<link>` コマンドに取り込んだり、実行別名として定義したりでき、一種のリソースです。実行リンクの使用については、99ページの「実行リンクの制御」を参照してください。

注

- *hyperlink* に対する値として *type* キーワード (上記のリスト参照) を使用しないでください。使用しなければならない場合は、形式の2行目にあるようにパラメータを明示的に指定してください。
- `<xref>` 要素が使用されているところではハイパーリンクは自動的に作成されるので、`<link>` 要素は `<xref>` 要素を使用する相互参照では必須ではありません。

例

- 次のマークアップは、ID `Intro` が付いたトピックへの単一ハイパーリンクを定義します。ID の大文字と小文字の区別には意味がありません。

```
<sl id=Intro>Introducing the Desktop
.
.
.
Refer to the <link intro>Introduction<\link>.
```

- 次のマークアップは、前の例にあるのと同じハイパーリンク・ジャンプを定義しますが、相互参照 (`<xref...>`) は自動的にハイパーリンクになるので `<link>` 要素は使用されません。この場合、`Intro` トピックのタイトルはヘルプタグによって自動的に指定されます。

```
Refer to <xref intro>.
```

このマークアップの出力は、次のとおりです。

```
Refer to Introducing the Desktop.
```

- 次のマークアップは、インライン・グラフィックが選択されたときに起動されるハイパーリンクを定義します。新規ウィンドウが開かれ、「clockfeatures」トピックを表示します。

```
Whenever you see the <link clockfeatures JumpNewView>  
<graphic entity=StopWatchIcon><\link> symbol, stop and answer the quiz questions.
```

出力は次のとおりです。



- 次のマークアップは、grep コマンドに関するマニュアル・ページを表示するリンクを作成します。

```
For more details, refer to the <link grep Man>grep man  
page<\link>.
```

- 次のマークアップは、startDtterm という名前の実行別名を使用して実行リンクを作成します。別名とそれが実行するコマンドは、アプリケーションのデフォルトのリソース・ファイルで定義されます。

```
To open a terminal window, click <link hyperlink='DtHelpExecAlias  
startDtterm' Execute>Start Terminal Emulator.<\link>
```

実行別名とその定義方法の詳細は、100ページの「実行別名」を参照してください。

関連項目

- 152ページの「<figure>」
- 159ページの「<hometopic>」

- 160ページの「<idx>」
- 161ページの「<image>」
- 174ページの「<location>」
- 197ページの「<xref>」
- 99ページの「実行リンクの制御」

<list>

リスト

オプションとして行頭文字が付けられたり、自動的に数字や文字が生成される項目から構成されているリストを開始します。

形式

```
<list [bullet | order | plain] [loose | tight] [continue]
[lalpha | ualpha | lroman | uroman | arabic] >
```

```
  * first item
  * second item
  .
  .
  .
```

```
<\list>
```

bullet	デフォルトで、各項目の前に黒い丸の行頭文字を表示します。
order	各項目の前に番号を表示します。番号は自動的に生成され、番号は1から始まります。デフォルトはアラビア数字です。順番が付けられるリストは、アルファベットやローマ数字を使用することもできます。
plain	各項目の前に行頭文字、数字、文字を入れません。
continue	項目の番号付けを引き続き前のリストから続けます。
loose	デフォルトで、項目間に縦の空間を作ります。

<code>tight</code>	項目間に余分な縦の空間を作しません。
<code>lalpha</code>	小文字のアルファベット
<code>ualpha</code>	大文字のアルファベット
<code>lroman</code>	小文字のローマ数字
<code>uroman</code>	大文字のローマ数字
<code>arabic</code>	<code>order</code> リスト・タイプの場合のデフォルト

各項目は復帰改行され、アスタリスク (*) または `<item>` タグから始まります。アスタリスクは、`<item>` タグの簡易形式です。アスタリスクの両側にスペースとタブを入力できます。項目は複数行にわたって続く場合もあります。項目は複数のパラグラフから成ることもあり、その場合空白行によりパラグラフが区切られます。リストのネスト化が許可されているため、リストをリスト内に表示することもできます。

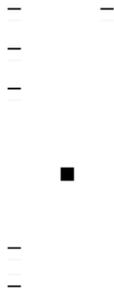
`<\list>` 終了タグは必須です。

例

次は、マークアップの例です。

```
<list>
* chocolate
* raspberry
* vanilla
<\list>
<list plain tight>
* Word Processing
* Graphics
* Printing
<\list>
<list order lalpha>
* Word Processing
* Graphics
* Printing
<\list>
```

出力は次のとおりです。



関連項目

- 163ページの「<item>」
- 165ページの「<lablist>」
- 156ページの「<head>」

<location>

位置

<location> 要素の位置を参照する際に ID を定義します。<location> 要素により、トピックの一部を <link> または <xref> の要素を使用してハイパーリンクの宛先として使用できます。

形式

```
<location id=id>text</location>
```

または

```
<location id=id|text|
```

id

現在の位置の識別子で、ハイパーリンクの宛先として使用できます。

text

ID を割り当てたいテキストのブロックです。

<location> 要素は、組み込み ID か設計者定義 id パラメータを持っている要素 (<hometopic> や <figure> など) がすでにある位置では必須ではありません。

<xref> 要素で作成された相互参照は、<xref> 要素を <location> の開始タグと終了タグの間のテキストに置き換えます。

例

次のマークアップ名は位置に名前を付け、その他の場所ではその位置へのハイパーリンクを作成します。

```
<s1 id=ConfigTopic> Configuration
...
<location id=ConfigTopicBody>some text<\location>
...
<s1 id=UseTopic> Usage
...
See <link ConfigTopicBody>Configuration<\link>
for additional information.
```

<location> 要素の ID へリンクすると、<location> タグが入力されているポイントへ、ヘルプ・ウィンドウが自動的にスクロールするという利点があります。対照的に、トピックの ID (この場合は「ConfigTopic」) へリンクすると、必ずトピックの先頭に行きます。

<location> 要素は、プレースホルダとして定義済みエンティティ (∅) を使用してファイル内の位置を参照することもできます。

ファイル内のキーの位置にこのマークアップを追加すると、その特定位置へのリンクを作成できます。

```
paragraph text
.
.
.
<location id=pointA>&empty;<\location>
.
.
.
```

関連項目

- 169ページの「<link>」
- 197ページの「<xref>」

<memo>

メモ

記述者のコメントや質問を識別しますが、最終的なヘルプ・ボリュームには表示されません。

形式

```
<memo>  
memo text<\memo>
```

または

```
<memo|memo text|
```

helptag.opt ファイルに memo を指定すると、メモ・テキストはヘルプ・ボリュームのドラフトに出力されます。指定しなければ、特にヘルプ・ボリュームの最終版を作成したときは、メモ・テキストは出力されません。メモ・テキストは、表示のときは別のフォントで出力されます。メモ・テキスト内ではマークアップを使用しないでください。

例

次はメモの例です。

```
<memo>  
Patti: We need a drawing to illustrate this.  
<\memo>
```

次のマークアップは、<memo> 要素の略式を使用しています。

```
<memo|Mike: Please explain how the following  
command is supposed to work|
```

関連項目

- 110ページの「作成者のメモを挿入するには」
- 119ページの「例: helptag.opt ファイル」のサンプルの helptag.opt ファイル

<metainfo>

メタ情報

メタ情報セクションを開始しますが、ヘルプ・ボリュームにある情報に関する情報も入っています。メタ情報にはボリュームのタイトルと著作権に関する記述があります。

形式

```
<helpvolume>
  <metainfo>
    <title>volume title
    <copyright>    &copy; Copyright XYZ Company 1995...
    <abstract>     brief description of help volume
    .
    .
    .
  <\metainfo>
<hometopic> ...
.
.
.
```

メタ情報セクションはオプションですが、通常はヘルプ・ボリュームの中に記述されています。オプションではあっても、タイトル、著作権、および概要のサブセクションは、ヘルプ・ボリュームに関する有益な情報ですのでなるべく使用してください。

これらのサブセクションのどれかを入れる場合、メタ情報セクションは必須です。

<otherfront> 要素は、あらかじめ定義されているタイトル、著作権、および概要のサブセクション以外のサブセクションを定義するのに使用されます。

<\metainfo> 終了タグは必須です。

例

```
<metainfo>
<title>Inventory Tracking Software

<copyright>
&copy; Copyright 1995 XYZ Company.
All rights reserved.

<abstract>
Explains how to use the Inventory Tracking Software
```

<\metainfo>

関連項目

- 192ページの「<title>」
- 144ページの「<copyright>」
- 138ページの「<abstract>」
- 180ページの「<otherfront>」

<newline>

復帰改行

段落や注釈内で復帰改行します。

形式

text<newline>text on next line

<newline> 要素に続くテキストが復帰改行してから始まります。

例

次のマークアップにより、パス名は必ず復帰改行してから始まります。

```
Put your files for the manual in the special directory  
<newline>/projects/userguide/draftdoc.
```

関連項目

- 195ページの「<vex>」
- 150ページの「<ex>」
- 161ページの「<image>」

<note>

注

重要なポイントとなるテキストに注目させるような特殊形式を作成します。

形式

```
<note>  
  text of note  
<\note>
```

注のデフォルトのヘッダは、「Note」(日本語ロケールでは「注」)です。別のヘッダを指定するには、<head> 要素を使用してください。

注にアイコンを表示したい場合は、<!entity ...> 宣言で NoteElementDefaultIconFile を定義してください。

icon.pm という名前のデフォルトの注アイコンは、/usr/dt/dthelp/dthelptag/icons ディレクトリにあります。

<\note> 終了タグは必須です。

例

- 次のマークアップはデフォルトのヘッダを使用しています。

```
<note>  
  Warranty information is in your installation manual.  
<\note>
```

- 次のマークアップは別のヘッダを指定しています。

```
<note><head>Read This First  
  Warranty information is in your installation manual.  
<\note>
```

関連項目

- 141ページの「<caution>」
- 196ページの「<warning>」
- 156ページの「<head>」

<otherfront>

他のメタ情報 (前付)

タイトル、著作権、および概要などの定義済みカテゴリのうちの一つで一致しないメタ情報 (前付) のために使用されます。<otherfront> 要素は非階層トピックを作成するためにも使用できます。非階層トピックはトピック・ツリーには表示されないため、トピックが表示されるようにするにはハイパーリンクを追加しなければなりません。<link> や<xref> の要素を使用して、<otherfront> 要素へのハイパーリンクを作成できます。

形式

```
<metainfo>
  .
  .
  .
<otherfront [id=id]><head>title of section
  text
```

ヘッダが必要な場合は、<head> 要素を使用してください。

<otherfront> は、<metainfo> の他のサブセクションすべての後に続きます。

関連項目

- 177ページの「<metainfo>」
- 156ページの「<head>」

<otherhead>

他のヘッダ

トピック内にサブヘッダを作成します。

形式

```
<otherhead>heading
```

ヘッダは、トピックのテキスト内のどこにでも入れることができます。<otherhead> 要素は、トピック・ツリーに表示されたヘルプ・トピックのリストには表示されません。

<\otherhead> 終了タグは必須ではありません。

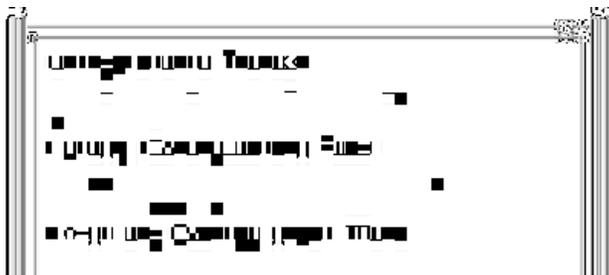
例

次の例は、<otherhead> 要素が <s1> トピック内の 2 つのサブセクションを識別する場合の例です。

```
<s1>Integration Tasks
There are two main tasks required to integrate your application.

<otherhead> Editing Configuration Files
Configuration files identify the colors, icons, and actions used by an application.
  <otherhead> Archiving Configuration Files
  .
  .
  .
```

このマークアップの出力は次のとおりです。



関連項目

- 156ページの「<head>」
- 184ページの「<procedure>」
- 186ページの「<rsect>」
- 187ページの「<s1>...<s9>」

<p>

新しい段落

インデントしたり、グラフィックを囲む段落を開始したりします。

形式

```
<p [indent] [gentity=graphic-ent [gposition=pos]
  [ghyperlink=gid [glinktype=type]]] [id=id] >text...
```

<code>indent</code>	オプションで、段落を現在の左マージンからスペース 6 つ分だけインデントするよう指定します。
<code>gentity=graphic-ent</code>	オプションで、段落がそのまわりを囲むグラフィックのエンティティ名です。gposition、ghyperlink、glinktype パラメータのいずれかを使用するときは、gentity パラメータと <i>graphic-ent</i> 値は必須です。
<code>gposition=pos</code>	オプションで、left または right は、オプションのグラフィックをそれぞれ左揃えまたは右揃えにします。
<code>ghyperlink=gid</code>	オプションで、グラフィックをハイパーリンクとして指定し、ハイパーリンクの宛先を指定します。glinktype パラメータを使用する場合は、ghyperlink パラメータと gid 値は必須です。<link> 要素の <i>hyperlink</i> 属性と同じように使用します。(このパラグラフの位置を参照するときは、gid 値ではなく id 値を使用します。)
<code>glinktype=type</code>	オプションで、ハイパーリンクの種類を指定します。デフォルトは Jump です。他には、JumpNewView、Definition、Man、Execute、AppDefined があります。<link> 要素の <i>type</i> 属性と同じように使用します。
<code>id=id</code>	オプションで、この位置を相互参照するのに使用する ID 名を定義します。

text

グラフィックのまわりを囲む段落のテキストです。

段落をインデントしたり、段落でグラフィックのまわりを囲んだり、または追い込みヘッダ・スタイルの段落を使用したりする必要がある場合は `<p>` 要素を使用してください。

オプションの `<head>` は `<p>` と共に使用できます。`<xref>` を使用して要素への相互参照を作成する場合は、`<head>` タグが必須です。`<head>` タグと `<\head>` タグを使用してヘッダのテキストを区切ってください。

`<\p>` 終了タグは必須ではありません。

例

- 次の例では 2 つの段落がありますが、2 番目の段落がインデントされます。

```
Some people do not like to read instruction manuals.  
<p indent>This is not always a good idea.
```

出力は次のとおりです。

Some people do not like to read instruction manuals. This is not always a good idea.

- 次のマークアップは、追い込みヘッダで段落を作成します。

```
<p><head>Examples and Illustrations <\head>  
Examples, perhaps the most common pattern of organization, are  
appropriate whenever the reader might be tempted to ask <quote>  
For example?<\quote>
```

出力は次のとおりです。



関連項目

- 156ページの「<head>」
- 184ページの「<procedure>」
- 187ページの「<s1>...<s9>」
- 106ページの「グラフィックのまわりをテキストで囲むには」
- 100ページの「実行別名」では、実行リンクの使用について説明しています。

<procedure>

プロシージャ

トピック内でセクションを開始します。

形式

```
<procedure>heading  
procedure text...
```

プロシージャはトピックのテキスト内のどこにでも表示できます。トピック・ツリーに表示されるトピックのリストには含まれません。

終了タグは必須ではありません。

例

マークアップは次のとおりです。

```
<procedure> Entering Special Characters  
To enter Greek or mathematical characters in your document, use the Symbols font.
```

出力は次のとおりです。



関連項目

- 156ページの「<head>」
- 180ページの「<otherhead>」
- 187ページの「<s1>...<s9>」

<quote>

引用符

テキストの前後を引用符を使用して二重引用符でくくります。

形式

`<quote>text<\quote>`

または

`"text"`

テキストを区切るには、開始と終了のタグ (`<quote>...<\quote>`) か一組の二重引用符 ("`...`") を使用します。

例

マークアップは次のとおりです。

```
... referred to in this manual as "the Standard" ...
```

出力は次のとおりです。

```
...referred to in this manual as "the Standard"...
```

関連項目

- 140ページの「<book>」
- 143ページの「<computer>」
- 194ページの「<var>」

<rsect>

参照セクション

参照セクション内のエントリを識別します。

形式

```
<rsect [id=id]>reference section heading  
.  
.  
.  
<rsub>reference subsection heading
```

<rsect> 要素は、参照セクションを識別するために使用できます。一連の同じようなセクションにある参照文献を識別するのに便利です。たとえば、各参照セクションでは1つのソフトウェア・コマンドについて説明しておくことができます。

<rsect> は、次の部分から構成されています。

- 必須のヘッダ
- オプションの導入部分のテキスト
- オプションの参照サブセクション (<rsub>)

各 <rsect> セクションには、複数の <rsub> セクションがあります。各 <rsub> 要素にはヘッダが必要です。参照サブセクションへの相互参照は行えません。

トピック・ツリーには、<rsect> ヘッダは入りますが、<rsub> ヘッダは入りません。

(<rsect> や <rsub> の) 終了タグは必須ではありません。

例

次のマークアップは、この要素の使用方法を示しています。

```
<rsect>purge  
.  
.  
.  
<rsub>Syntax  
purge filename  
  
<rsub>Example  
purge file01  
<rsub>Related Commands
```

delete

関連項目

- 142ページの「<chapter>」
- 187ページの「<s1>...<s9>」

<s1>...<s9>

サブセクション (<s1>, <s2>, ... , <s9>)

階層内でトピックを開始します。

形式

```
<sn [id=name]>heading  
topic text...
```

n は、レベル番号 (1、2、...、9 のいずれか) です。

<chapter> トピックは <s1> サブトピックを持つことができ、<s1> トピックは <s2> サブトピックを持つことができ、というように続いていきます。レベルはスキップできません。

セクションのヘッダは、<sn > タグと同じ行か次の行に置くことができます。したがって、ヘッダは必須です。セクション内のテキストはオプションです。

終了タグは通常省略されますが、いくつかの例では終了タグが必要です。たとえば、同じレベルにある <rsect> 要素がセクションの後に続く場合、セクションの終了タグは必須です。終了タグがないと、<rsect> 要素は直前のセクションのサブセクションと見なされます。

例

- 次の例では、トピック内に3つのレベルの階層があります。

```
<chapter>Running the Processor  
  topic text...  
    <s1>Getting Started
```

To run the program, type in the usercode and your password.

<s1>Customizing

You may now set up this conversion program to change your computer from beige to red.

<s2>Configuration

Use either the disk drive or the tape drive to archive your files.

<s3>Disk Drive Advantages

See data sheet for specifications.

<s3>Tape Drive Advantages

See data sheet for specifications.

<s2>Support

If you really need help, call technical support.

- 次のマークアップでは、<rsect> セクションが階層内で同じレベルになるようにセクション終了タグ (<\s1>) が使用されています。

```
<s1>first-level heading
text
<s1>first-level heading
text<\s1>
<rsect>first-level heading
text
```

反対に、終了タグを使用しないと、<rsect> セクションは2番目の <s1> セクションのサブトピックになります。

```
<s1>first-level heading
text
<s1>first-level heading
text
<rsect>second-level heading
text
```

関連項目

- 142ページの「<chapter>」
- 156ページの「<head>」
- 186ページの「<rsect>」

サブスクリプト

サブスクリプト文字を作成します。

形式

`<sub>character to subscript<\sub>`

または

`__text__`

簡易形式では、サブスクリプトにする文字の前後に2つの `__` (下線) 文字を使用します。

例

マークアップは次のとおりです。

```
<p>The chemical element H<sub>2<\sub>O contains  
two hydrogen molecules.
```

出力は次のとおりです。

The chemical element H₂O contains two hydrogen molecules.

関連項目

- 189ページの「<sup>」

スーパースクリプト

スーパースクリプト文字を作成します。

形式

`<super>character to superscript<\super>`

または

`^^text^^`

速記形式では、スーパースクリプトにする文字の前後に 2 つの ^^ (脱字記号) 文字を使用します。

例

マークアップは次のとおりです。

`<p>The answer to the problem is 2<super>8<\super>.`

出力は次のとおりです。

The answer to the problem is 2⁸.

関連項目

- 189ページの「`<sub>`」

<term>

用語集の用語

新たに紹介する用語を特殊フォントで記述し、用語集にある定義へのハイパーリンクを作成します。

形式

`<term baseform [gloss | nogloss]>text<\term>`

または

`<term baseform [gloss | nogloss]|text|`

または

++text++

baseform	用語の形式がテキストで使用されているものと同じでない場合に、用語集に表示される用語の形式。たとえば、テキストでは複数形式が使用されているのに、用語集では単数形式が使用されているような場合です。用語にスペースや特殊文字が入っている場合、引用符の中に <i>baseform</i> 文字列を入れてください。
gloss	デフォルトで、用語が用語集に入っていることを確認するようにヘルプタグに要求します。
nogloss	用語集には用語を入れませんが、用語はボールド・フォントで表示されます。

簡易形式では、用語集の用語の前後に 2 つの ++ (プラス記号) を使用します。

注・ヘルプ・ボリュームに用語集が入っていない場合は、nogloss パラメータを使用してください。

nogloss パラメータが使用されず、用語集に対応する定義がない場合にヘルプタグがヘルプ・ボリュームを処理すると、警告メッセージが出力されます。

<term> 要素で用語にタグを付けると、用語集へのハイパーリンクが自動的に作成されます。用語集がない場合、リンクは動作しません。

正規マークアップが使用されている場合、<\term> 終了タグは必須です。

例

次のマークアップは、その用語が用語集にある用語であることを示すため特殊フォント (破線の下線が付いたボールド・フォント) で「**structural element**」を記述し、用語集へのハイパーリンクを作成します。用語集のエントリにはスペースが入っているので、テキストには引用符が付けられます。テキスト内では複数形式で表示されます。ヘルプタグは単数形式で用語集内をチェックし、見つからない場合はエラーを出力します。

```
SGML views a document as a hierarchy  
of <term "structural element"|structural elements|.
```

関連項目

- 154ページの「<glossary>」
- 145ページの「<dterm>」

<title>

ヘルプ・ボリューム・タイトル

ヘルプ・ボリュームのタイトルを指定します。

形式

```
<metainfo>  
<title>help volume title
```

<title> 要素は、<metainfo> (メタ情報) セクション内のオプションの要素です。しかし、タイトルにはヘルプ・ダイアログ・ボックスに表示されるボリューム名が表示されるので、この要素を使用するようにしてください。

<title> は、<metainfo> タグのすぐ後に続きます。ボリュームのタイトルは、タイトルをフォーマットできない可能性がある他のアプリケーション (インフォメーション・ビューアなど) によって表示される場合もあるので、タイトル内ではプレーン・テキストだけを使用するようにしてください。

<\title> 終了タグは必須ではありません。

例

次の例は、サンプルのボリューム・タイトルです。

```
<metainfo>  
<title>The Super Hyperlink User's Guide
```

関連項目

- 177ページの「<metainfo>」

<user>

ユーザの応答

コンピュータのプロンプトに対するユーザの応答を表示します。

形式

```
<user>response<\user>
```

または

```
<user|response|
```

この要素は、コンピュータ・ダイアログでユーザ入力とコンピュータ出力とを区別するために使用されます。通常は <ex> 要素内で使用されますが、この場合 <user> 開始タグと <\user> 終了タグの間のスペースと改行は維持されます。

段落内で使用される場合、<user> テキストがソース・ファイル内で行の途中で切れないようにしなければなりません。

正規マークアップが使用されている場合、<user> 終了タグは必須です。

例

次のマークアップは、2種類のフォントを生成します。1つはコンピュータの表示内容を示し、もう1つはユーザの入力を示します。

```
<ex>  
Do you wish to continue? (Yes or No) <user>Yes<\user>  
<\ex>
```

出力は次のとおりです。

```
Do you wish to continue? (Yes or No) Yes
```

関連項目

- 143ページの「<computer>」
- 150ページの「<ex>」
- 195ページの「<vex>」

<var>

変数

コマンド内のユーザ指定変数を表示します。

形式

```
<var>  
text  
<\var>
```

または

```
%%text%%
```

正規マークアップが使用されている場合、<\var> 終了タグは必須です。

簡易形式では、テキストの前後に 2 つの %% (パーセント記号) を使用します。

例

マークアップは次のとおりです。

```
INPUT <var>filename<\var>
```

または

```
INPUT %%filename%%
```

出力は次のとおりです。

```
INPUT filename
```

関連項目

- 150ページの「<ex>」
- 143ページの「<computer>」

<vex>

逐語例

その中ではヘルプタグ要素が要素として解釈されない逐語例を表示します。

形式

```
<vex [number | nonumber][smaller | smallest]> text<\vex>
```

nonumber	デフォルトで、行番号を省略します。
number	各行の先頭に行番号を付けます。
smaller または smallest	より小さいフォントを使用して例を表示します。長い行をより狭い幅の中に表示します。

逐語例の中では、ヘルプタグは認識されませんが、<\ だけは終了タグと見なされません。

マークアップとして別に解釈される可能性のあるマークアップ・タグやその他の文字を表示する必要がある場合はこの要素を使用します。改行やスペースは、ソース・ファイル内で表示されたのと同じ状態のままです。

smaller フォントと smallest フォントを使用すると、幅の広い例をマージン内に収めることができます。

例

マークアップは次のとおりです。

```
<vex>  
<!ELEMENT copyright - O (text)  
-memo | location | idx >  
<\vex>
```

出力は次のとおりです。



関連項目

- 150ページの「<ex>」
- 161ページの「<image>」

<warning>

警告

ユーザにとって危険性の高い状況に対して注意するようにユーザに警告します。

形式

```
<warning>  
text  
<\warning>
```

警告メッセージのテキストはボールドで出力されます。

警告のデフォルトのヘッダは、「Warning」(日本語ロケールでは「警告」)です。別のヘッダを指定するには、<head> 要素を使用してください。

警告と共にグラフィックを表示するには、<!entity> 宣言で WarningElementDefaultIconFile を定義してください。warnicon.pm という名前のデフォルトの警告アイコンは、/usr/dt/dthelp/dthelptag/icons ディレクトリにあります。

<\warning> 終了タグは必須です。

例

- 次のマークアップは、警告メッセージを作成します。

```
<warning>
Failure to follow these guidelines could result
in serious consequences.
<\warning>
```

- 次のマークアップは、警告メッセージの別のヘッダを指定します。

```
<warning><head>Danger!
Do not open the high-voltage compartment.
<\warning>
```

関連項目

- 179ページの「<note>」
- 141ページの「<caution>」
- 156ページの「<head>」

<xref>

相互参照

ヘルプ・ボリューム内で別の位置を指定し、その位置へのハイパーリンクを作成するテキストを挿入します。

形式

```
<xref id>
```

id は相互参照されているトピックまたは位置の識別子です。

相互参照は、章またはセクションのタイトル、ヘッダ、図のタイトル、リスト項目、または行番号に変換されます。相互参照テキストは、ハイパーリンクになり、ユーザが選択すると相互参照された位置へジャンプします。

相互参照を作成するには、参照予定の要素に *id* を定義しなければなりません。宛先要素の *id* を <xref> タグにある *id* パラメータとして使用します。これにより、<xref> 要素から宛先要素へのハイパーリンクが作成されます。*id* はそのまま正確にスペルアウトされなければなりません。しかし、大文字と小文字の区別はありません。

id パラメータは、次の要素につけることができます。

```
<chapter>
<s1>, <s2>, ...<s9>
<otherfront>
<p>
<image>
<item>
<figure>
<location>
<rsect>
```

下線が入っている id (「_abstract」や「_hometopic」など) への相互参照はできません。<link> 要素を使用してください。

例

ユーザが詳細な章を参照できるようにするには、章の id を使用して参照を作成してください。たとえば、id が *windowmgr* である「Window Management」というタイトルの章を参照するには、「Refer to <xref windowmgr> for details.」と記述します。

オンライン・ヘルプ・ボリュームでは、結果は「Refer to Window Management for details.」となります。id は章のタイトル「Window Management」に置き換えられ、ハイパーリンクになります。

次のマークアップは「analyzer」という名前の id をセクションの要素に代入します。

```
<s1 id=analyzer>Logic Analyzers
```

次の例は、このトピックへの相互参照が記述されているマークアップです。

```
The DX16500A logic analysis system, described in
<xref analyzer>, can be configured to a user's needs.
```

処理後、<xref> 要素は次のように「Logic Analyzers」に置き換えられます。



テキスト「Logic Analyzers」はハイパーリンクで、ユーザが選択すると相互参照ヘルプ・トピックへジャンプします。

関連項目

- 142ページの「<chapter>」
- 147ページの「<!entity>」
- 152ページの「<figure>」
- 155ページの「<graphic>」
- 161ページの「<image>」
- 169ページの「<link>」
- 174ページの「<location>」
- 180ページの「<otherfront>」
- 182ページの「<p>」
- 186ページの「<rsect>」
- 187ページの「<s1>...<s9>」

特殊文字エンティティの概要

この章では、ヘルプ・トピックの記述に使用する特殊文字のリストを示します。

特殊文字

この特殊文字は、関連するエンティティ名を、特殊文字を表示する場所へ指定することにより、テキストに入力できます。

記述に * (アスタリスク) のマークが付いているエンティティを使用するには、`helpchar.ent` ファイルを使用してください (詳細は、108ページの「特殊文字の表示」を参照してください)。

表 6-1 印刷上の記号

記号	エンティティ名	説明
©	<code>&copy;</code>	著作権記号
®	<code>&reg;</code>	登録商標
™	<code>&tm;</code>	商標
—	<code>&endash;</code>	エヌ・ダッシュ (短いダッシュ)
—	<code>&emdash;</code>	エム・ダッシュ (長いダッシュ)

表 6-1 印刷上の記号 続く

記号	エンティティ名	説明
•	•	* 行頭文字
↵	&cr;	復帰改行
...	&ellipsis;	省略記号 (水平型)
⋯	&peellipsis;	省略記号 (文の終わり)
⋮	&vellipsis;	垂直型省略記号
'	"e;	一重引用符
"	&dquote;	二重引用符
␣	&vblank;	垂直空白
()	∅	空 (テキストなし)
()	&sigspace;	有意のスペース
-	&sigdash;	行の切れ目ではないハイフン
§	&S;	* セクション
¶	&P;	* 段落

記述に * (アスタリスク) のマークが付いているエンティティを使用するには、`helpchar.ent` ファイルを使用してください (詳細は、108ページの「特殊文字の表示」を参照してください)。

表 6-2 ギリシャ文字

記号	エンティティ名	説明
小文字		
N/A	<code>&alpha;</code>	* アルファ
N/A	<code>&beta;</code>	* ベータ
N/A	<code>&chi;</code>	* カイ
N/A	<code>&delta;</code>	* デルタ
N/A	<code>&varepsilon;</code>	* エプシロン
N/A	<code>&phi;</code>	* ファイ
N/A	<code>&varphi;</code>	* オープン・ファイ
N/A	<code>&gamma;</code>	* ガンマ
N/A	<code>&eta;</code>	* イータ
N/A	<code>&iota;</code>	* イオタ
N/A	<code>&kappa;</code>	* カッパ
N/A	<code>&lambda;</code>	* ラムダ
N/A	<code>&mu;</code>	* ミュー
N/A	<code>&nu;</code>	* ニュー
N/A	<code>&pi;</code>	* パイ
N/A	<code>&varpi;</code>	* パイの代替文字 (またはオメガ)
N/A	<code>&theta;</code>	* シータ
N/A	<code>&vartheta;</code>	* オープン・シータ

表 6-2 ギリシャ文字 続く

記号	エンティティ名	説明
N/A	ρ	* ロー
N/A	N/A	* シグマ
N/A	&tsigma;	* シグマ 1
N/A	τ	* タウ
N/A	ε	* イプシロン
N/A	ω	* オメガ
N/A	ξ	* クシー
N/A	ψ	* プシー
N/A	ζ	* ゼータ
大文字		
N/A	&Udelta;	* デルタ
N/A	&Uphi;	* ファイ
N/A	&Ugamma;	* ガンマ
N/A	&Ulambda;	* ラムダ
N/A	&Upi;	* パイ
N/A	&Utheta;	* シータ
N/A	&Usigma;	* シグマ
N/A	Υ	* イプシロン
N/A	&Uomega;	* オメガ

表 6-2 ギリシャ文字 続く

記号	エンティティ名	説明
N/A	&Uxi;	* クシー
N/A	ϒ	* プシー

記述に * (アスタリスク) のマークが付いているエンティティを使用するには、`helpchar.ent` ファイルを使用してください (詳細は、108ページの「特殊文字の表示」を参照してください)。

表 6-3 数学記号

記号	エンティティ名	説明
基本数学記号		
-	−	減算
±	±	プラス・マイナス
÷	÷	除算
×	×	乗算
≤	≤	以下
≥	≥	以上
≠	&neq;	不等号
拡張数学記号		
2	&squared;	* 二乗
3	&cubed;	* 三乗
1/4	&one-fourth;	* 四分の一

表 6-3 数学記号 続く

記号	エンティティ名	説明
$1/2$	<code>&one-half;</code>	* 二分の一
$3/4$	<code>&three-fourths;</code>	* 四分の三
∞	<code>&infty;</code>	* 無限大
N/A	<code>&equiv;</code>	* 恒等的に等しい
\neq	<code>&not-eq;</code>	* 不等号
\approx	<code>&approx;</code>	* 近似
\ddot{y}	<code>&neg;</code>	* 否定
«	<code>&cap;</code>	* キャップ (共通部分)
»	<code>&cup;</code>	* カップ (和集合)
\vee	<code>&vee;</code>	* ヴィー (論理和)
\wedge	<code>&wedge;</code>	* ウェッジ (論理積)
N/A	<code>&in;</code>	* 属する
N/A	<code>&subset;</code>	* 真部分集合
N/A	<code>&subseteq;</code>	* 部分集合
N/A	<code>&supset;</code>	* 真超集合
N/A	<code>&supseteq;</code>	* 超集
\forall	<code>&forall;</code>	* 全称記号
\exists	<code>&exists;</code>	* 存在記号
N/A	<code>&not-in;</code>	非要素

表 6-3 数学記号 続く

記号	エンティティ名	説明
N/A	<code>&function;</code>	* 関数記号 (フロリン記号)
N/A	<code>&angle;</code>	* 角度
\cong	<code>&cong;</code>	* 合同
N/A	<code>&propto;</code>	* 比例
N/A	<code>&perp;</code>	* 垂直
·	<code>&cdot;</code>	* 中心点
N/A	<code>&oplus;</code>	* 直和
N/A	<code>&otimes;</code>	* 直積
\emptyset	<code>&oslash;</code>	* 空集合
N/A	<code>&partial;</code>	* 微分デルタ
N/A	<code>&sum;</code>	* 総和 (ギリシャ文字の大文字のシグマ)
×	<code>&prod;</code>	* 積 (ギリシャ文字の大文字のパイ)

記述に * (アスタリスク) のマークが付いているエンティティを使用するには、`helpchar.ent` ファイルを使用してください (詳細は、108ページの「特殊文字の表示」を参照してください)。

表 6-4 矢印記号

記号	エンティティ名	説明
←	<code>&leftarrow;</code>	* 左矢印
N/A	<code>&rightarrow;</code>	* 右矢印

表 6-4 矢印記号 続く

記号	エンティティ名	説明
N/A	↑	* 上矢印
N/A	↓	* 下矢印
N/A	↔	* 両方向矢印
←	&bigleftarrow;	* 左大矢印
⇒	&bigrightarrow;	* 右大矢印
N/A	&biguparrow;	* 上大矢印
N/A	&bigdownarrow;	* 下大矢印
N/A	&bigleftrightarrow;	* 両大矢印

記述に * (アスタリスク) のマークが付いているエンティティを使用するには、`helpchar.ent` ファイルを使用してください (詳細は、108ページの「特殊文字の表示」を参照してください)。

表 6-5 その他の記号

記号	エンティティ名	説明
現在の日付と時間		
//	&date;	今日の日付 (ヘルプタグ実行時)
09:50	&time;	現在の時間 (ヘルプタグ実行時)
通貨記号		
¢	¢s;	セント
£	&sterling;	ポンド

表 6-5 その他の記号 続く

記号	エンティティ名	説明
¥	¥	円
単位		
°	°	度
N/A	&minutes;	分、プライム、フィート
"	&seconds;	秒、ダブル・プライム、インチ
AM	&a.m.;	午前
PM	&p.m.;	午後
トランプ組札		
◇	♦	* ダイヤ
N/A	♥	* ハート
N/A	♠	* スペード
N/A	♣	* クローバー
その他の記号		
◇	⋄	* ダイヤ
¿	&invert-question;	逆クエスチョン・マーク
¡	&invert-exclamation;	逆エクスクラメーション・マーク
¤	¤cy;	通貨
N/A	∴	ゆえに
«	&openanglequote;	左三角引用符

表 6-5 その他の記号 続く

記号	エンティティ名	説明
»	&closeanglequote;	右三角引用符
N/A	ℵ	* ヘブライ語のアレフ
∇	∇	* ナブラ (ギリシャ文字の逆大文字デルタ)
N/A	&surd;	平方根、正方向列
N/A	℘	* ワイエルシュトラス記号
N/A	ℜ	* ドイツ文字の R
N/A	&im;	* ドイツ文字の I

記述に* (アスタリスク) のマークが付いているエンティティを使用するには、`helpchar.ent` ファイルを使用してください (詳細は、108ページの「特殊文字の表示」を参照してください)。

コマンドの要約

この章では、ヘルプ・コマンドが端末ウィンドウで手動で実行されるときに使用可能なコマンド行オプションを要約します。

- 211ページの「ヘルプ・システム・コマンド」
- 212ページの「ヘルプタグ・ファイルの処理 (dthelptag)」
- 215ページの「ヘルプ・トピックの表示 (dthelpview)」
- 216ページの「ブラウザ・ヘルプ・ボリュームの作成 (dthelpgen)」

ヘルプ・システム・コマンド

ヘルプ・システムが提供するデスクトップのアクションとデータ型により、ヘルプ・ファイルのアイコンをクリックするかメニュー項目を選択すると、実行時のヘルプ・ファイルをコンパイルしたり表示したりできます。しかし、特定のコマンド・オプションを選択する場合、端末エミュレータでコマンドを入力するか、新しいアクションを作成しなければなりません。

ヘルプのアクションとデータ型は、`/usr/dt/appconfig/types/lang` ディレクトリにある 2 つのファイル `dthelp.dt` と `dthelptag.dt` に定義されています。

ここで要約されているコマンドは、次のとおりです。

<code>dthelptag</code>	ヘルプタグのソース・ファイルを実行ファイルにコンパイルします。
------------------------	---------------------------------

dthelpview	ヘルプ・ボリューム、ヘルプ・トピック、テキスト・ファイルまたはマニュアル・ページを表示します。
dthelpgen	各ファミリー・ファイルのエントリが入っている新しいヘルプ・ボリューム <code>browser.hv</code> にヘルプ・ファミリー・ファイルを格納します。

ヘルプタグ・ファイルの処理 (dthelptag)

ヘルプタグ・ソフトウェアは `dthelptag` コマンドで起動され、ヘルプタグのソース・ファイルを実行時のヘルプ・ファイルにコンパイルします。`volume.htg` ファイルがあるディレクトリで `dthelptag` を実行してください。

コマンド形式

```
dthelptag [command-options] volume [parser-options]
```

command-options は、*volume* 名の前に入力されるオプションで、*parser-options* は *volume* 名の後に入力されるオプションです。

コマンド・オプション

<code>-clean</code>	指定の <i>volume</i> のためにヘルプタグを以前に実行したときに作成されたすべてのファイルを削除します。
<code>-shortnames</code>	作成されたすべてのファイル名を、ベース名は 8 文字まで、拡張子は 3 文字までにそれぞれ制限します。これにより、実行時のヘルプ・ファイルを長い名前がサポートされていないシステムに移植できます。
<code>-verbose</code>	<code>dthelptag</code> コマンドの経過状況を表示し、発生したパーサ・エラーをすべて表示します。パー

サ・エラーは *volume.err* という名前のファイルに保存されます。

-formal

正規パーサを使用して、SGML 準拠マークアップでタグが付けられたヘルプ・ファイルを解釈します。このオプションを指定しない場合、*dthelptag* は入力ファイルに簡易マークアップが記述されているものと判断します。

2 種類のマークアップ (簡易マークアップと正規マークアップ) があるため、ファイル拡張子を使用してその種類を識別するようにしてください。簡易マークアップには *.htg* を、正規マークアップには *.ctg* を使用してください。

パーサ・オプション

パーサ・オプションは *volume* 名の後に入力され、*parser* に直接渡されます。この *parser* は、マークアップ・ファイルを実行時のファイルに変換するヘルプタグ・ソフトウェアの一部です。

これらのオプションは次の方法で使用できます。

- *volume* 名の後にコマンド行で入力する
- 現在のディレクトリにある *helptag.opt* という名前のファイルにリストする
- 現在のディレクトリにある *volume.opt* という名前のファイルにリストする
- *DTTAGOPT* 環境変数を使用して設定する

コマンド行に入力されたオプションは、別の方法を使用して設定されたオプションを無効にします。

onerror

パーサ・エラーが発生した場合に、*dthelptag* コマンドを続行すべきかどうかを指定します。デフォルトは *onerror=stop* で、これによりパーサ・エラーが1つでも発生するとコマンドは停止します。*onerror=go* を指定すると、処理は続行されますが、作成された実行時のヘルプ・ファイルが正しく動作しない可能性があります。

charset

テキストファイルを記述するのに使用された文字セットを指定します。ヘルプ・トピックが適切なフォントで表示されるようにするには文字

セット名が正しくなければなりません。デフォルトは `charset=ISO-8859-1` です。

`LanguageElementDefaultCharset` という名前のエンティティを宣言してヘルプ・ボリューム内の文字セットを指定することもできます。

`/usr/dt/dthelp/dthelptag/helplang.ent` ファイルには、このエンティティの宣言が記述されています。サポートされている文字セットのリストについては、第 14 章を参照してください。

<code>search</code>	参照されたファイル・エンティティを探すために検索されるディレクトリのリストに別のディレクトリを追加します。複数のディレクトリを指定するには、複数の <code>search=directory</code> オプションを使用してください。検索オプションが使用されていない場合は、現在のディレクトリだけが検索されます。
<code>clearsearch</code>	検索ディレクトリのリストを無視します。このオプションは、 <code>helptag.opt</code> ファイルに指定された検索オプションを無効にする場合にコマンド行で有効です。
<code>memo</code>	設計者のメモ (<code><memo></code> 要素を使用して入力される) を取り込みます。デフォルトは <code>nomemo</code> で、これはヘルプタグがメモを無視するようにします。
<code>nomemo</code>	ヘルプタグが設計者のメモ (<code><memo></code> 要素で入力される) を無視するようにします。これがデフォルトです。

関連項目

- 117ページの「実行時のヘルプ・ファイルの作成」
- 268ページの「インストール・パッケージの作成」
- 121ページの「ヘルプ・ボリュームの表示」

ヘルプ・トピックの表示 (dthelpview)

dthelpview コマンドは、ヘルプ・ボリューム、個々のヘルプ・トピック、テキスト・ファイル、またはマニュアル・ページを表示するために使用できます。

コマンド形式

ヘルプ・ビューアを起動するには、次のいくつかの方法があります。

- dthelpview -helpVolume *volume* [-locationId *id*]
- dthelpview -man
- dthelpview -manPage *man*
- dthelpview -file *filename*

-helpVolume *volume* 表示したい *volume.sd1* ファイル名を指定します。ボリュームが現在のディレクトリになく、ボリュームが登録されていない場合はパス名が必要です。

-locationId *id* ID を指定します。dthelpview は、*id* が入っているトピックを表示します。ID を指定しない場合、ヘルプ・ビューアはデフォルトとして `_hometopic` を使用します。

-man 表示するマニュアル・ページを要求するダイアログを表示してから、要求されたマニュアル・ページを表示します。

-manPage *man* 特定のマニュアル・ページが表示されるように指定します。

-file *filename* 特定のテキスト・ファイルが表示されるように指定します。

デフォルトの *volume* と *id* は `dthelpview` の `app-defaults` ファイルである `/usr/dt/app-defaults/lang/Dthelpview` に設定できます。*lang* にはロケール名が入ります。

関連項目

- 271ページの「アプリケーションとそのヘルプの登録」
- 121ページの「ヘルプ・ボリュームの表示」
- `dthelpview(1)` のマニュアル・ページ

ブラウザ・ヘルプ・ボリュームの作成 (`dthelpgen`)

`dthelpgen` ユーティリティは、フロントパネルの [ヘルプ・ビューア] を使用してシステムに登録されたヘルプ・ボリュームを表示できるようにする特殊なヘルプ・ボリュームを作成します。フロントパネルの [ヘルプ・ビューア] コントロールを最初にクリックすると、`dthelpgen` は自動的に実行されます。ヘルプ検索パス・ディレクトリ (ローカルな、またはネットワーク化されたディレクトリ) を検索してヘルプ・ファミリ・ファイルを位置付け、次にユーザの `HomeDirectory/.dt/help/$DTUSERSESSION` ディレクトリにブラウザ・ボリューム (`browser.hv`) を作成します。いったん作成されると、ボリュームは次のアクションのどれかに応じて更新されます。

- ファミリ・ファイルまたはヘルプ・ボリュームを追加、削除、または変更する
- `LANG` 環境変数を変更する
- `ReloadApps` アクションを起動する
- `dthelpgen` を端末エミュレータで手動で実行する

ブラウザ・ボリュームは、フロントパネルの [ヘルプ・ビューア] コントロールをクリックすると表示されます。または、`dthelpview` を手動で実行し、次のコマンド行で示されているようにブラウザ・ボリューム名を指定できます。

```
dthelpview -h browser.hv
```

コマンド形式

```
dthelpgen -dir [options]
```

`-dir` ブラウザ・ボリュームと中間ファイルを格納するディレクトリを指定します。これは、必須のパラメータです。

オプション

`-generate` システム上のファミリー・ファイルとヘルプ・ボリュームが変更されていない場合でも、新しいブラウザ・ヘルプ・ボリュームが作成されるように指定します。

`-file basename` `dthelpgen` が作成したヘルプ・ボリュームと中間ファイルの名前を指定します。デフォルトは、`browser.hv` です。

`-lang` どの言語ディレクトリのヘルプ・ファミリーとヘルプ・ボリュームを検索するかを指定します。`-lang` オプションを設定すると、LANG 環境変数の現在値よりも優先されます。

注・ブラウザ・ボリュームがヘルプ・ウィンドウに表示されている間に `dthelpgen` を実行する場合、ウィンドウを閉じてからブラウザ・ボリュームをもう一度開いてください。

関連項目

- 271ページの「アプリケーションとそのヘルプの登録」
- `dthelpgen(1)` のマニュアル・ページ

ヘルプタグ文書型定義の読み方

この章では、ヘルプタグ 1.3 文書型定義 (DTD) の読み方と、完全に準拠した Standard Generalized Markup Language (SGML) ヘルプ・ファイルを作成するための DTD の使用方法を説明します。

- 219ページの「文書型定義」
- 220ページの「DTD コンポーネント」
- 224ページの「正規マークアップ」
- 228ページの「正規マークアップの処理」

文書型定義

文書型定義 (DTD) は、構造化された (または階層のある) ドキュメントを作成するために要素のセットを定義します。DTD は各要素の形式を指定し、ドキュメント内の要素の使用方法を決定します。

ヘルプタグ 1.3 DTD

ヘルプタグ 1.3 DTD タグ・セットとその関連規則は、正規マークアップとして参照されます。DTD は Standard Generalized Markup Language (SGML) ISO 指定 8879: 1986 に準拠しています。つまり、SGML 準拠のヘルプ・ファイルを作成するのに正規マークアップを使用できます。

付録 A には完全な DTD 仕様が載っています。DTD は開発者用ツールキットでも使用できます。DTD は /usr/dt/dthelp/dthelptag/dtd ディレクトリにあり、名前は helptag.dtd です。

関連項目

- dthelptag.dtd(4) のマニュアル・ページ

DTD コンポーネント

DTD は、これまでの章で説明したヘルプタグの各要素を定義します。この節では、いくつかの重要な用語を紹介し、要素記述の構文の読み方を説明します。DTD の各セクションを完全に説明するものではありません。

要素宣言

DTD は各要素を要素宣言で定義します。宣言には、要素を説明する詳細な記述、それに必要なコンポーネント、指定できる要素またはできない要素を使用します。要素には属性宣言で定義された特性もあります。特性の詳細は、223ページの「属性リスト宣言」のセクションで説明します。

要素宣言の形式は次のとおりです。

```
<ELEMENT element_type minimization (content model) >
```

element_type

要素名を指定します。これはタグ名としても使用されます。たとえば、要素型 head のタグは <head> です。

minimization

開始タグまたは終了タグが必要かどうかを示す 2 文字のエントリです。1 番目の文字は開始タグを表します。2 番目の文字は終了タグを表します。その 2 文字の間に 1 つ空白が入ります。文字 o はそのタグがオプションであることを意味します。- (マイナス記号) はタグが必要であることを意味します。たとえば、-- というエントリは、その要素には開始タグと終了タグの両方が必要であるという意味です。ヘルプタグ 1.3 用 DTD

ではすべての要素に開始タグと終了タグが必要です。

content model

要素型に含められる必要な要素およびオプションの要素のリストを指定します。要素のシーケンスと、適用可能であれば、要素を指定できる回数を定義します。

コンテンツ・モデルは次の記述を使用します。

	「or」という意味です。
+	要素を最低1回は指定しなければなりません。繰り返し可能です。
*	要素を0回以上指定しなければなりません。
?	要素を0回または1回指定しなければなりません。
,	カンマはシーケンスを説明します。つまり、要素型の後には、カンマの後に指定された要素が続かなければなりません。
+ (element_type(s))	+ (プラス記号) は、リストされた1つまたは複数の要素が、この要素内か、それを含む要素どの中でも使用できることを示します。これを取り込みと呼びます。1つ以上の要素を囲むのに括弧を使用します。
- (element_type(s))	- (マイナス記号) は、リストされた1つまたは複数の要素が、この要素内か、それを含む要素のいずれかの中では使用できないことを示します。これを排他と呼びます。1つ以上の要素を囲むには括弧を使用します。

例

各例には、提供されている要素宣言に対する語の説明があります。必要な開始タグと終了タグが使用されます。

- 章は、テキストが後に続く <chaphead> を必要とします。1つの章には0個以上の s1 要素と、それに続く0個以上の rsect 要素を指定できます。

```
<!ELEMENT chapter - - (chaphead, text, (s1*, rsect*)) >
```

- chaphead は、オプションの abbrev が後に続くヘッダを必要とします。chaphead には memo、location、idx の要素は指定できません。

```
<!ELEMENT chaphead - - (head, abbrev?)
- (memo | location | idx) >
```

- 段落要素は、開始タグ (-) と終了タグ (-) を必要とします。段落要素には、partext 要素が後に続くオプションのヘッダ (?) を指定できます。newline 要素は、段落要素に含まれる p 要素かその他の要素の中で使用できます。

```
<!ELEMENT p - - (head?, partext) +(newline) >
```

- note にはテキストが含まれます。オプションのヘッダを指定できます。note には、note、caution、warning の要素は指定できません。

```
<!ELEMENT note - - (head?, text)
- (note | caution | warning) >
```

- list にはオプションのヘッダを指定できます。list には1つ以上の項目 (繰り返し可能) が必要です。

```
<!ELEMENT list - - (head?, item+) >
```

- book 要素宣言は、別の book 要素を含むことができないように指定するため排他処理を使用します。

```
<!ELEMENT book - - (partext) - (book) >
```

要素宣言のキーワード

要素の中には、要素宣言の中に、要素のデータ内容を説明するキーワードを含むものがあります。DTD には、EMPTY、CDATA、#PCDATA の3つのキーワードがあります。

EMPTY 要素が、オンライン情報で表示されるデータ内容を持たないことを指定します。newline 要素と xref 要素がこの例です。

CDATA 「文字データ」を表します。つまり、その要素のデータ内容はマークアップとして認識されません。

#PCDATA

「構文解析された文字データ」を表します。つまり、そのデータ内容にはヘルプ・システムのパーサが解釈するテキストとマークアップ文字の両方が含まれます。

属性リスト宣言

属性リストは、要素をより詳しく説明する追加の属性を宣言します。属性リスト宣言の形式は次のとおりです。

```
<!ATTLIST element_type attribute_values default_value>
```

たとえば `list` 要素には、`type`、`ordertype`、`spacing`、`continue` の 4 つの属性があります。それぞれの型の値が宣言されます。最後のカラムはデフォルト値を示しています。`continue` 属性には値が 1 つしか存在しないので、デフォルト値は省きます。

```
<!ATTLIST list type      ( order
                          bullet
                          plain
                          check )      bullet
      ordertype ( ualpha
                  lalpha
                  arabic
                  uroman
                  lroman )      arabic
      spacing   ( tight
                  loose )      tight
      continue  (continue)      #IMPLIED >
```

上記のマークアップは、リスト項目の間に空白行が入り、番号 (大文字のアルファベット) の付いたリストを作成します。

```
<list order ualpha loose>
  <item>
    <text>
      <p>
        <partext>Introducing the Front Panel</partext>
      </p>
    </text>
  </item>
```

正規マークアップ

正規マークアップの作成には、構造化エディタを使用するのが最良の方法です。基本的な要素のセットを覚えれば、設計者は作業を始められます。作業の開始は、メニューから要素を選択することです。それに応じて、構造化エディタが各要素に必要なすべてのタグを生成します。さらにアプリケーションが、作成中の構造フレームワークが文書型定義に準拠するかを確認します。

簡易マークアップ、正規マークアップ、構造化エディタについては、39ページの「ヘルプタグによるヘルプ・トピックの記述」を参照してください。

正規マークアップの注意点

簡易マークアップと正規マークアップは共通の要素のセット (章、セクション、ヘッダ、リスト、段落など) を共有しています。しかし、正規マークアップは次のような重要な点で簡易マークアップとは異なります。

- どの要素にも開始タグと終了タグが必要です。
- 要素の各サブコンポーネントのタグを入力しなければなりません。
- 正規マークアップでは / (スラッシュ) が終了タグ区切り文字です。終了タグは `</tagname>` の形式です。
- エンティティ宣言には、簡易宣言で使用する FILE パラメータの代わりに、SYSTEM パラメータを使用します。

明示的な開始タグおよび終了タグ

各要素、そのコンポーネント部分、各要素内の要素は明示的にタグ付けされなければなりません。たとえば、次の例は章のヘッダの正規マークアップです。この例やその他のマークアップ例では、読みやすいようにタグをインデントしてあります。実際のマークアップではインデントは必要ありません。

```
<chaphead>
  <head>
    <parttext>Front Panel Help</parttext>
  </head>
</chaphead>
```

追加のタグ <head> と <parttext> に注意してください。これらは <chaphead> 要素のサブコンポーネントです。これらの要素の 1 つ 1 つに対して、明示的な開始タグおよび終了タグが必要です。

要素の明確な階層

各要素宣言は、要素をどこでどのように使用するかを管理する規則のセットの元になります。要素の中には別の要素が指定でき、その中にまた別の要素を指定できるので、ドキュメントは要素の階層になります。トップレベルでは <helpvolume> がその他のすべての要素のコンテナになっています。

ヘルプ・トピックを作成するのにどのマークアップが必要かを判別するために、規則に慣れる必要があります。たとえば、章を作成したいとします。まず、次にリストされている chapter の宣言を見てください。chaphead が必要であると指定されています。次に、chaphead の規則を見てください。今度は head が必要であると指定されています。したがって、head の宣言を見てください。そしてネストにされた最後の要素 (この場合は parttext) まで見てください。一般的に使用する要素に慣れるまで、マークアップを正しく入力するにはこの方法が役立ちます。

```
<!ELEMENT chapter - - (chaphead, text?, (sl*, rsect*)) >
<!ELEMENT chaphead - - (head, abbrev?)
    - (memo | location | idx | footnote) >
<!ELEMENT head - - (parttext)
    - (memo | location | idx) >
<!ELEMENT parttext - - ((#PCDATA . . . ))>
```

構造化エディタを使用すると、設計者に必要な DTD の知識を最小限にできます。エディタ・アプリケーションは DTD を「読み」、各要素に必要なタグを作成します。作成されるタグの多くは中間構造タグです。

例

次の正規マークアップ例は、デスクトップのテキスト・エディタのヘルプ・ボリュームからの抜粋です。関連するオンライン情報を表示するには、[フロントパネル] で [ヘルプ・ビューア] を選択します。[デスクトップの紹介] を選択し、リストされたボリュームの中から [テキスト・エディタのヘルプ] を選択します。テキスト・エディタ・ボリュームで [テキスト・エディタの使い方] を選択し、次に [既存のドキュメントを開くには] を選択します。

テキストと、対応するタグが読みやすいように、この例ではインデントを使用しています。

```

<s2 id=' 'TOOPENANEXISTINGDOCUMENT' ' >
<chaphead><head>
<partext>To Open an Existing Document</partext>
  </head></chaphead>
<text>
<p>
<partext>You can use Text Editor or File Manager to open an existing
document .
  </partext></p>
<idx><indexprimary>
<partext>document</partext></indexprimary>
  <indexsub>
<partext>opening</partext></indexsub></idx>
<idx><indexprimary>
<partext>opening</partext></indexprimary>
  <indexsub>
<partext>existing document</partext></indexsub></idx>
<procedure>
<chaphead><head>
<partext>From Text Editor</partext>
  </head></chaphead>
<text>
<list type=' 'ORDER' ' >
<item><text><p>
<partext>Choose Open from the File menu.</partext></p>
<p>
<partext>The Open a File dialog box lists files and folders on your
system.You can browse the documents listed, or change to a new folder
to locate other files on your system.</partext>
  </p></text></item>
<item><text><p>
<partext>Select the document you want to open in the Files list or
type the file name in the Open a File field.</partext></p>
<p>
<partext><emph><partext>Or,</partext></emph> if the document is not
in the current folder, first change to the folder that contains your
document. Then choose a name in the Folders list or type the path
name of the folder you wish to change to in the Enter path or folder
name field.</partext></p></text></item>
<item><text><p>
<partext>Press Return or click OK.
  </partext></p></text></item></list>
<figure tonumber=' 'NONUMBER' ' entity=' 'TEXTEDITOROPENFILE' ' >
</figure></text></procedure>

<procedure><chaphead><head>
<partext>From File Manager</partext>
  </head></chaphead>
<idx><indexprimary>
<partext>opening</partext></indexprimary>
  <indexsub>
<partext>document from File Manager</partext></indexsub></idx>
<idx><indexprimary>
<partext>document</partext></indexprimary>
  <indexsub>
<partext>opening from File Manager</partext></indexsub></idx>
<idx><indexprimary>

```

(続く)

```

<parttext>File Manager</parttext></indexprimary>
  <indexsub>
<parttext>opening document</parttext></indexsub></idx>
<text>
<list type='BULLET'>
<item><text><p>
<parttext>Display the document's file icon in a File Manager
window.</parttext>
  </p></text></item>
<item><text><p>
<parttext>Do <emph><parttext>one</parttext></emph> of the
following:</parttext></p>
<list type='BULLET'>
<item><text><p>
<parttext>Double-click the document's file icon.</parttext>
  </p></text></item>
<item><text><p>
<parttext>Select the document, then choose Open from the Selected
menu.</parttext>
  </p></text></item>
<item><text><p>
<parttext>Drag the document to Text Editor's control in the Front
Panel.</parttext>
  </p></text></item></list></text>
</item></list><text> </procedure>
<procedure><chaphead><head>
<parttext>See Also</parttext>
  </head></chaphead>
<text>
<list type='BULLET' spacing='TIGHT'>
<item><text><p>
<parttext><xref id='ENTERINGANDEDITINGTEXT'></parttext>
  </p></text></item>
<item><text><p>
<parttext><xref id='TOSAVEADOCUMENTTOTHECURRENTFILE'></parttext>
  </p></text></item>
<item><text><p>
<parttext><xref id='TABLEOFCONTENTS'></parttext>
  </p></text></item></list></text>
</procedure></text></s2>

```

ファイル・エンティティ宣言

正規マークアップでファイル・エンティティを宣言するには、次の形式を使用します。

```
<!entity entityname SYSTEM ``filename''>
```

entityname はエンティティ名で、*filename* はファイル名です。キーワード `SYSTEM` は必須です。

注 - 以前に簡易マークアップで作成されたエンティティ宣言を使用する場合は、FILE パラメータを SYSTEM に置き換えなければなりません。

例

次の例は、3つのテキストから成りグラフィック・イメージを1つ含むヘルプ・ボリュームのエンティティ宣言です。

```
<!entity MetaInformation SYSTEM ``metainfo>``  
<!entity BasicTasks SYSTEM ``basics``>  
  <!entity AdvancedFeatures SYSTEM ``advanced``>  
  <!entity process_diagram SYSTEM ``process.tif``>
```

エンティティは、簡易マークアップの場合とまったく同様に正規マークアップでも参照されます。

正規マークアップの処理

dthelptag を使用して正規マークアップを処理する場合、コマンド行オプションの `-formal` を使用しなければなりません。たとえば、`Icons.ctg` という名前の正規マークアップ・ファイルを詳細モードで処理するには、次のコマンドを入力します。

```
dthelptag -verbose -formal Icons.ctg
```

注 - コマンド・オプションは、入力ファイルのマークアップの型を指定します。dthelptag を実行することによって作成される実行時のファイルは常に `volume.sdl` です。オンラインでは、簡易マークアップと正規マークアップのどちらを使用しても同じです。

パート III プログラマが行う作業

ヘルプ・ダイアログ・ボックスの作成と管理

この章では、ヘルプ・ダイアログ・ウィジェットとその作成方法を説明します。

- 231ページの「ヘルプ・ダイアログ・ボックス」
- 232ページの「一般ヘルプ・ダイアログ」
- 234ページの「簡易ヘルプ・ダイアログ」
- 237ページの「アプリケーション・プログラム・インタフェースの概要」

ヘルプ・ダイアログ・ボックス

ヘルプ・システムは、アプリケーション・プログラマに対して、ヘルプ・ダイアログ・ボックスをすべての OSF/Motif アプリケーションに追加するプログラミング・ライブラリを提供します。

- 一般ヘルプ・ダイアログには、メニュー・バー、トピック・ツリー、ヘルプ・トピック表示領域があります。トピック・ツリーには、ヘルプ・ボリュームのトピックとサブトピックが表示されます。ユーザは、表示するトピックを選択したり、使用可能なトピックをブラウズしたり、トピックがヘルプ・ボリュームのどこにあるかを見つけるために、トピック・ツリーを使用します。
- 簡易ヘルプ・ダイアログには、トピック表示領域と、下部に1つ以上のダイアログ・ボタンがあります。

標準 Xt 活用例

プログラミングの点から見ると、ユーザは、アプリケーション内の他の OSF/Motif ウィジェットと対話するのと同じようにヘルプ・ダイアログと対話します。2 種類のヘルプ・ダイアログ・ボックスは、2 つの新しいウィジェット・クラス (DtHelpDialog と DtHelpQuickDialog) として定義されます。

ボリューム名、トピック ID などのほとんどのヘルプ・ウィンドウの属性は、ウィジェット・リソースとして処理されます。たとえば、新しいトピックを表示するには、DtNhelpVolume、DtNlocationId、DtNhelpType の各リソースを設定するために XtSetValues () 呼び出しを実行するだけです。詳細は、240ページの「ヘルプ・トピックの表示」を参照してください。

注 - ヘルプ・システムをアプリケーションに統合するには、C プログラミング言語、OSF/Motif プログラマ・ツールキット、Xt イントリンシクス・ツールキットに関する実用的な知識が必要です。

一般ヘルプ・ダイアログ

一般ヘルプ・ダイアログには、トピック・ツリーとトピック表示領域の 2 つの表示領域があります。トピック・ツリーは、スクロールできるヘルプ・トピックのリストを提供しています。常にホーム・トピックのタイトルが 1 番目の項目になります。ユーザがタイトルを選択すると、そのタイトルに矢印 (⇒) が付き、そのヘルプ情報がトピック表示領域に表示されます。図 9-1 に、一般ヘルプ・ウィンドウのトピック・ツリーとトピック表示領域を示します。現在のトピック「To select a palette」が表示されます。

一般ヘルプ・ダイアログには、[前のトピックへ]、[履歴]、[索引] の 3 つのダイアログ・ボタンがあります。これらのコマンドはヘルプ・メニューでも使用できます。ヘルプ・ダイアログとグラフィカル・ユーザ・インタフェースの概要については、27ページの「ヘルプ・ユーザ・インタフェース」を参照してください。

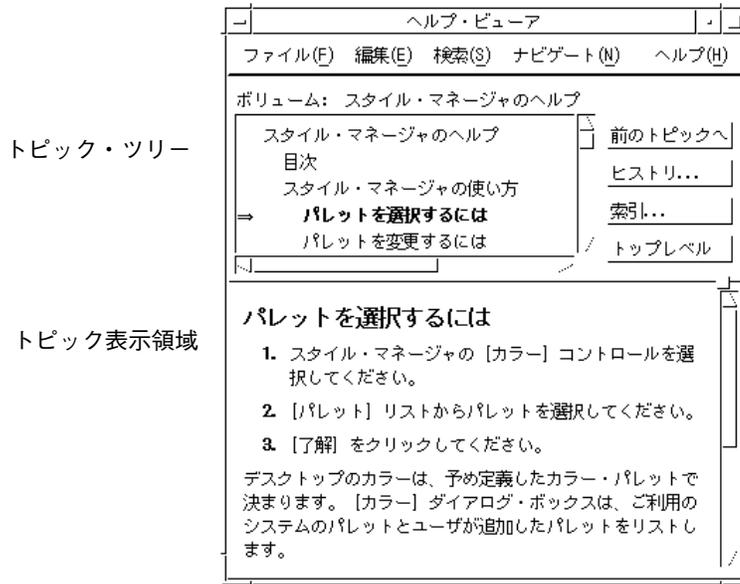


図 9-1 一般ヘルプ・ダイアログ

▼ 一般ヘルプ・ダイアログを作成するには

1. 適切なヘッダ・ファイルを指定します。

```
#include <Help.h>
#include <HelpDialog.h>
```

2. 一般ヘルプ・ダイアログ・ウィジェットのインスタンスを作成します。

DtCreateHelpDialog() 簡易関数

または XtCreateManagedWidget() 関数を使用します。

3. ダイアログ内で発生するハイパーリンク・イベントを処理するためのコールバックを追加します (詳細は、255ページの「ハイパーリンク・イベントへの応答」を参照してください)。
4. Close コマンドを処理するためのクローズ・コールバックを追加します。

例

次のコード・セグメントは、簡易関数を使用して一般ヘルプ・ダイアログを (*parent* の子として) 作成します。ダイアログは管理されていません。ヘルプが要求された時点で何かのアプリケーションで管理されます。

```
Widget mainHelpDialog, moreButton, helpButton;
ac = 0;
XtSetArg (al[ac], XmNtitle, "My Application - Help"); ac++;
XtSetArg (al[ac], DtNhelpVolume, "My Help Volume"); ac++;
XtSetArg (al[ac], DtNlocationId, "Getting Started"); ac++;
XtSetArg (al[ac], DtNhelpType, "DtHELP_TYPE_TOPIC"); ac++;

mainHelpDialog =
    DtCreateHelpDialog (parent, "mainHelpDialog", al, ac);
```

次の2つの呼び出しは、ダイアログにハイパーリンクとクローズ・コールバックを追加します。通常、関数 `HyperlinkCB()` と `CLoseHelpCB` はアプリケーションのどこかで宣言されます。

```
XtAddCallback (mainHelpDialog, DtNhyperLinkCallback,
               HyperlinkCB, (XtPointer)NULL);
XtAddCallback (mainHelpDialog, DtNcloseCallback,
               CloseHelpCB, (XtPointer)NULL);
```

関連項目

- 第12章
- 257ページの「アプリケーション構成ボタンを使用可能にするには」
- `DtCreateHelpDialog(3)` のマニュアル・ページ
- `DtHelpDialog(3)` のマニュアル・ページ

簡易ヘルプ・ダイアログ

簡易ヘルプ・ダイアログ・ボックスは、オンライン・ヘルプの第一目標であるユーザができるだけ早く、またうまく作業に戻れることが達成されるように設計されています。このシンプルなユーザ・インタフェースにより、ユーザは情報に集中できます。ダイアログの情報は、ユーザが読んだ後に消してしまっても作業が続けられるように、便利なものでなければなりません。

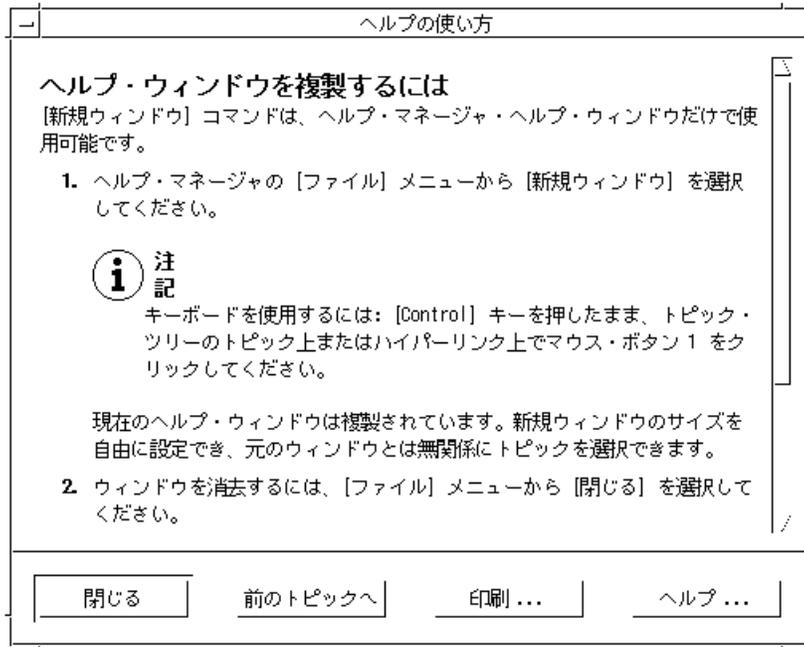


図 9-2 標準ボタンが4つある簡易ヘルプ・ダイアログ

簡易ヘルプ・ダイアログには5つのボタンがあり、そのうち4つが管理されています。残りのダイアログ・ボタンは構成可能なので、このボタンは希望に合わせて使用できます。しかし、その目的としては次の2つの方法のどちらかでヘルプへのパスを提供することです。

- ユーザがより詳しい情報を探せるようにします。この場合、デフォルト・ボタン・ラベル (More) が適切です。これは「段階的な表示」と呼ばれます。
- アプリケーションのヘルプ・ボリュームを通常どおりブラウズするために一般ヘルプ・ダイアログを開けるようにします。この場合、「Browse...」が最も適切なボタン・ラベルになります。

開発者用ツールキットには簡易関数 `DtHelpQuickDialogGetChild()` が含まれています。この関数は、どの簡易ヘルプ・ダイアログ・ボタンにも対応するウィジェット ID を決定します。

▼ 簡易ヘルプ・ダイアログを作成するには

1. 適切なヘッダ・ファイルを指定します。

```
#include <Help.h>
#include <HelpQuickD.h>
```

2. 簡易ヘルプ・ダイアログ・ウィジェットのインスタンスを作成します。

`DtCreateHelpQuickDialog()` 簡易関数

または `XtCreateManagedWidget()` 関数を使用します。

3. ダイアログ内で発生するハイパーリンク・イベントを処理するためのコールバックを追加します (詳細は、255ページの「ハイパーリンク・イベントへの応答」を参照してください)。

4. [了解] ボタンを処理するためのクローズ・コールバックを追加します。

5. 使用するダイアログ・ボタンを構成します。

- アプリケーション定義のボタンを使用するには、ボタンを管理してアクティブ・コールバックを追加します。
- 印刷を許可しないようにするには、[印刷] ボタンを管理しないでください。
- ユーザが「ヘルプの使い方」を使用できるように、[ヘルプ] ボタンを管理してダイアログへのヘルプ・コールバックを追加します。

例

次のコード・セグメントは、簡易関数を使用して簡易ヘルプ・ダイアログを (*parent* の子として) 作成します。ダイアログは管理されていません。通常、ヘルプが要求された時点でアプリケーションのどこかで管理されます。この例では、アプリケーション定義のボタンが使用可能になっており、ヘルプをさらに要求するために使用します。

```
Widget quickHelpDialog, moreButton, helpButton;
```

```
ac = 0;
XtSetArg (al[ac], XmNtitle, "My Application - Help"); ac++;
XtSetArg (al[ac], DtNhelpVolume, "My Help Volume"); ac++;
XtSetArg (al[ac], DtNlocationId, "Getting Started"); ac++;
XtSetArg (al[ac], DtNhelpType, "DtHELP_TYPE_TOPIC"); ac++;
```

```
quickHelpDialog =
DtCreateHelpQuickDialog (parent, "quickHelpDialog", al, ac);
```

次の2つの呼び出しは、ダイアログにハイパーリンクとクローズ・コールバックを追加します。通常、関数 `HyperlinkCB()` と `CloseHelpCB()` はアプリケーションのどこかで宣言されます。

```
XtAddCallback (quickHelpDialog, DtNhyperLinkCallback,
HyperlinkCB, (XtPointer) NULL);
```

```
XtAddCallback (quickHelpDialog, DtNcloseCallback,  
              CloseHelpCB, (XtPointer)NULL);
```

ここでアプリケーション定義のボタンが管理され、アプリケーションの MoreHelpCB() 関数を起動する起動コールバックが割り当てられます。

```
moreButton = DtHelpQuickDialogGetChild (quickHelpDialog,  
                                         DT_HELP_QUICK_MORE_BUTTON);  
XtManageChild (moreButton);  
XtAddCallback (moreButton, XmNactivateCallback,  
              MoreHelpCB, (XtPointer)NULL);
```

「ヘルプの使い方」を提供するために、ダイアログの [ヘルプ] ボタンが管理され、ダイアログにヘルプ・コールバックが追加されます。

```
helpButton = DtHelpQuickDialogGetChild (quickHelpDialog,  
                                         DT_HELP_QUICK_HELP_BUTTON);  
XtManageChild (helpButton);  
XtAddCallback (quickHelpDialog, DtNhelpCallback,  
              HelpRequestCB, USING_HELP);
```

他の OSF/Motif ダイアログと同様に、簡易ヘルプ・ダイアログにヘルプ・コールバックを追加すると、[F1] キーと [ヘルプ] ボタンのどちらからでも簡易ヘルプ・ダイアログが使用できます。

関連項目

- 257ページの「アプリケーション構成ボタンを使用可能にするには」
- 第 12 章
- DtCreateHelpQuickDialog(3) のマニュアル・ページ
- DtHelpQuickDialog(3) のマニュアル・ページ
- DtHelpQuickDialogGetChild(3) のマニュアル・ページ

アプリケーション・プログラム・インタフェースの概要

ヘルプ・システムに関連するマニュアル・ページは次のとおりです。

- ダイアログの作成と操作のための関数

```
DtHelp(5)  
DtHelpDialog(5)
```

```
DtHelpQuickD(5)
DtCreateHelpDialog()
DtCreateHelpQuickDialog()
DtHelpQuickDialogGetChild()
```

■ 項目ヘルプ・モードの実現のための関数

```
DtHelpReturnSelectedWidgetId()
```

■ ヘルプ・ライブラリにメッセージ・カタログを指定するための関数

```
DtHelpSetCatalogName()
```

■ ヘルプ・ボリュームの作成および表示のためのアプリケーションとアクション

```
dthelptag(1)
dthelpview(1)
dthelpgen(1)
dthelpaction(5)
dtmanaction(5)
```

■ 文書型定義

```
dthelptagdtd(4)
dtsdldtd(4)
```

ヘルプ要求への応答

この章では、ヘルプ・ダイアログ・ウィジェットのリソースの設定によってさまざまなヘルプ情報を表示する方法を説明します。

- 239ページの「ヘルプの要求」
- 240ページの「ヘルプ・トピックの表示」
- 242ページの「ヘルプ・トピックを表示するには」
- 242ページの「テキスト文字列を表示するには」
- 243ページの「テキスト・ファイルを表示するには」
- 244ページの「マニュアル・ページを表示するには」
- 245ページの「ヘルプ・キー ([F1] キー) を使用可能にする」
- 248ページの「[ヘルプ] メニューの提供」
- 249ページの「アイテムヘルプ・モードのサポート」

ヘルプの要求

アプリケーションの使用中にユーザがヘルプを要求したときは、どのヘルプ・トピックを表示すべきか、アプリケーション側で決定します。

コンテキストの検知

「バージョン」のヘルプ (通常は著作権のトピックが表示されます) など、特定の情報を明示的に要求するヘルプ要求もあります。ただしそれ以外のヘルプ要求では、コンテキストを要求します。つまり、アプリケーションにおけるユーザの現在のコンテキストに基づいて適切なヘルプ・トピックを選ぶ処理が必要です。

たとえばアプリケーションは、適切なヘルプ・トピックを判断するため、モードや設定のステータスをテストすることがあります。あるいは、入力フィールドの値をテストし、その値が有効でなければ詳細ヘルプを、有効であれば一般ヘルプを表示することもあります。

エントリ・ポイント

エントリ・ポイントとはヘルプ・ボリューム内の特定の場所のことで、通常はトピックの先頭です。アプリケーション内でヘルプを要求すると、エントリ・ポイントに直接アクセスできます。

設計者の見地から言えば、エントリ・ポイントはヘルプ・ボリュームの適切な場所に ID を指定することにより設定されます。プログラマの見地から言えば、ユーザがヘルプを要求できるようにし、特定のヘルプが要求されたときに適切な ID を使用することでエントリ・ポイントが作成されます。

ユーザがヘルプを要求する方法には次の 4 つがあります。

- ヘルプ・キー (ほとんどのキーボードでは [F1] キー) を押す
- ダイアログ・ボックスの [ヘルプ] ボタンをクリックする
- アプリケーションの [ヘルプ] メニューからコマンドを選択する
- [アイテムヘルプ] を選択する

ヘルプ・トピックの表示

ヘルプが要求されると、アプリケーションが表示するヘルプ・トピックを決定します。次にヘルプ・ダイアログを (必要であれば) 作成し、管理して、ヘルプ・トピックを表示するのに適切なりソースを設定します。

ほとんどの要求については、アプリケーションのヘルプ・ボリュームの一部であるヘルプ・トピックが表示されます。しかしヘルプ・システムのヘルプ・ダイアログ

は、マニュアル・ページ、テキスト・ファイル、単一テキスト文字列を表示することもできます。

ヘルプ・システムのヘルプ・ダイアログは Xt イントリンシクスと OSF/Motif プログラミングに明示的に基づいているので、他のウィジェットと同じように、リソースを設定することによりヘルプ・ダイアログ内で値を変更できます。

DtNhelpType リソースは、表示すべき情報の種類を決定します。次のいずれかの値が設定されます。

■ DtHELP_TYPE_TOPIC

ヘルプ・ボリュームの一部である通常のヘルプ・トピックを表示します。ボリュームは DtNhelpVolume リソースを設定して指定します。トピックは DtNlocationId リソースを設定して指定します。

■ DtHELP_TYPE_STRING

アプリケーションが提供する文字列を表示します。自動的な行の折り返しは使用できないので、改行は文字列で指定してください。文字列は DtNstringData リソースを設定して指定します。

■ DtHELP_TYPE_DYNAMIC_STRING

アプリケーションが提供する文字列を、行の折り返しでテキストをフォーマットした状態で表示します。改行はパラグラフの切れ目を示します。文字列は DtNstringData リソースを設定して指定します。

■ DtHELP_TYPE_FILE

テキスト・ファイルを表示します。表示するファイル名は、DtNhelpFile リソースを設定して指定します。

■ DtHELP_TYPE_MAN_PAGE

ヘルプ・ダイアログのマニュアル参照ページ(マニュアル・ページ)を表示します。表示するマニュアル・ページは、DtNmanPage リソースを設定して指定します。

上記の値は Help.h ファイルで定義されています。

関連項目

- 第9章

▼ ヘルプ・トピックを表示するには

1. ヘルプ・ダイアログを作成します。

2. 次のようなヘルプ・ダイアログのリソースを設定します。

DtNhelpType DtHELP_TYPE_TOPIC を設定します。

DtNhelpVolume アプリケーションのボリューム名を設定します。

DtNlocationId 表示するトピックの ID を設定します。

サイズやタイトルなど、その他の値も設定できます。

3. XtManageChild() を使用してダイアログを管理します。

例

次のプログラム・セグメントは、ID である getting-started が付いたトピックを、ボリューム MyVolume に表示します。

```
ac = 0;
```

```
XtSetArg (al[ac], DtNhelpType, DtHELP_TYPE_TOPIC); ac++;
XtSetArg (al[ac], DtNhelpVolume, "MyVolume"); ac++;
XtSetArg (al[ac], DtNlocationId, "getting-started"); ac++;
XtSetArg (al[ac], DtNcolumns, 40); ac++;
XtSetArg (al[ac], DtNrows, 12); ac++;
XtSetValues (helpDialog, al, ac);
XtManageChild (helpDialog);
```

ヘルプ・ボリューム MyVolume が登録されていない場合、DtNhelpVolume の値を指定するためには、MyVolume.sdl ファイルへの絶対パスが必要です。

▼ テキスト文字列を表示するには

1. 簡易ヘルプ・ダイアログを作成します。

一般ヘルプ・ダイアログも使用できますが、このダイアログの機能は文字列データにほとんど適用されないため、使用しないようにしてください。

2. 次のようなヘルプ・ダイアログのリソースを設定します。

DtNhelpType DtHELP_TYPE_DYNAMIC_STRING (行の折り返しを使用する場合) または DtHELP_TYPE_STRING (改行を有効にしたい場合) を設定します。

DtNstringData 表示する文字列を設定します。文字列のコピーは内部的に保持されるので、保存する必要はありません。

サイズやタイトルなど、その他の値も設定できます。

3. XtManageChild() を使用してダイアログを管理します。

例

次のプログラム・セグメントは、変数 descriptionString に格納された文字列を表示します。

```
ac = 0;
XtSetArg (al[ac], DtNhelpType, DtHELP_TYPE_DYNAMIC_STRING); ac++;
XtSetArg (al[ac], DtNstringData, (char *)descriptionString); ac++;
XtSetValues (quickHelpDialog, al, ac);
XtManageChild (quickHelpDialog);
```

ヘルプ・ダイアログはデータのコピーを自身で作成するので、文字列が不要になった場合はメモリを解放できます。

```
XtFree (descriptionString);
```

▼ テキスト・ファイルを表示するには

1. 簡易ヘルプ・ダイアログを作成するか、ダイアログ・キャッシュの1つを取り出します。

一般ヘルプ・ダイアログも使用できますが、このダイアログの機能のほとんどが標準ヘルプ・トピックについてのみ有効なので、使用しないようにしてください。

2. 次のようなヘルプ・ダイアログのリソースを設定します。

DtNhelpType DtHELP_TYPE_FILE を設定します。

DtNhelpFile 表示するファイル名を設定します。アプリケーションの現在のディレクトリにファイルがない場合は、ファイルへのパスを指定してください。

サイズやタイトルなど、その他の値も設定できます。特に幅はテキスト・ファイルの標準である 80 に設定します。

3. `XtManageChild()` を使用してダイアログを管理します。

例

次のプログラム・セグメントは、ファイル `/tmp/printer.list` を表示します。ダイアログのサイズもテキスト・ファイルにより適応するように設定します。

```
ac = 0;
XtSetArg (al[ac], DtNhelpType, DtHELP_TYPE_FILE);      ac++;
XtSetArg (al[ac], DtNhelpFile, "/tmp/printer.list");    ac++;
XtSetArg (al[ac], DtNcolumns, 80);                      ac++;
XtSetArg (al[ac], DtNrows, 20);                         ac++;
XtSetValues (quickHelpDialog, al, ac);
XtManageChild (quickHelpDialog);
```

▼ マニュアル・ページを表示するには

1. 簡易ヘルプ・ダイアログを作成します。

一般ヘルプ・ダイアログも使用できますが、このダイアログの機能のほとんどが標準ヘルプ・トピックについてのみ有効なので、使用しないようにしてください。

2. 次のようなヘルプ・ダイアログのリソースを設定します。

`DtNhelpType` `DtHELP_TYPE_MAN_PAGE` を設定します。

`DtNmanPage` マニュアル・ページ名を設定します。このリソースの値は、システムの `man` コマンドへ直接渡されます。したがって特定のセクションを指定するときは、通常 `man` コマンドを入力するときのように、セクション番号の前にマニュアル・ページ名を付けてください。

サイズやタイトルなど、その他の値も設定できます。

3. `XtManageChild()` を使用してダイアログを管理します。

例

次のプログラム・セグメントは、`grep` コマンドのマニュアル・ページを表示します。ダイアログのサイズもマニュアル・ページに適応するように設定します。

clientData

HelpRequestCB() 関数に渡すデータ。通常は表示するトピックを示します。

ユーザがヘルプ・キーを押すと、ウィジェットのヘルプ・コールバックが現在のキーボード・フォーカスと共に呼び出されます。ウィジェットにヘルプ・コールバックがない場合、ヘルプ・コールバックを持つ最も近い先祖のコールバックが呼び出されます。

ヘルプ・コールバックが見つからない場合は何も処理されません。この場合、ヘルプ・コールバックをアプリケーションの各シェルに追加するようにしてください。こうすると、ユーザのヘルプ要求が失われることはありません。

ヘルプ・コールバックをダイアログ・シェルに追加すると、自動的にダイアログの[ヘルプ] ボタンでヘルプ・コールバックを呼び出せるようになります。

クライアント・データの重要性

追加したそれぞれのヘルプ・コールバックで *clientData* に固有の値を指定すると、各ヘルプ・コールバックを処理するために別々の関数を書かずに済みます。アプリケーションにはすべてのヘルプ要求を処理する 1 つのコールバック・プロシージャを持たせることができます (245 ページの「ヘルプ・コールバックを追加するには」を参照してください)。コールバック・プロシージャでは、ユーザが要求したヘルプを識別するのに *clientData* を使用します。つまり、ヘルプ・コールバックを追加するたびに、*clientData* に固有の値を設定してください。

例

次の例は、ID をエントリ・ポイントに指定する方法の 1 つを示しています。HelpEntryIds.h ファイルは、各ヘルプ・コールバックの *clientData* に固有の整数を定義するのに使用します。各ウィジェットに対する 2 つの ID 文字列も定義されます。1 つは通常の [F1] ヘルプ用、もう 1 つはアイテムヘルプ・モード (ユーザがウィジェットから記述を取得する場所) 用です。

たとえば、アプリケーションのユーザ・インタフェースが、Name、Address、Telephone Number という 3 つの入力フィールドを持つメイン・ウィンドウだとします。次の例は HelpEntryIds.h ファイルの内容です。

```
#define HELP_volumeName          "MyVolume"  
#define HELP_MainWindow         100  
#define HELP_MainWindow_ID      "basic-tasks"  
#define HELP_MainWindow_ITEM_ID "main-window-desc"
```

```

#define HELP_NameField          101
#define HELP_NameField_ID      "specifying-a-name"
#define HELP_NameField_ITEM_ID "name-field-desc"
#define HELP_AddressField      102
#define HELP_AddressField_ID   "specifying-an-address"
#define HELP_AddressField_ITEM_ID "address-field-desc"
#define HELP_PhoneField        103
#define HELP_PhoneField_ID     "specifying-a-phone-no"
#define HELP_PhoneField_ITEM_ID "phone-field-desc"

```

最初にウィジェットを作成したアプリケーションの一部で、次のようにヘルプ・コールバックが各ウィジェットに追加されます。

```

XtAddCallback (mainWindow, DtNhelpCallback,
               HelpRequestCB, HELP_MainWindow);
XtAddCallback (nameField, DtNhelpCallback,
               HelpRequestCB, HELP_NameField);
XtAddCallback (addressField, DtNhelpCallback,
               HelpRequestCB, HELP_AddressField);
XtAddCallback (phoneField, DtNhelpCallback,
               HelpRequestCB, HELP_PhoneField);

```

HelpRequestCB() 関数では、ヘルプ要求を (switch() 文によって) デイスパッチするために *clientData* パラメータを使用します。ヘルプ・コールバックが [F1] キーによって呼び出された (フラグは「false」) のか、アイテムヘルプ・モードでユーザがウィジェットを選択した (フラグは「true」) のかを調べるため、グローバル・フラグ *itemHelp* の値が調べられます。

```

XtCallbackProc HelpRequestCB (
    Widget      w,
    XtPointer   clientData,
    XtPointer   callData )
{
    char        *topicToDisplay;
    Boolean     useQuickHelpDialog;
    /* Determine the topic ID for the given 'clientData.' */
    switch ((int)clientData)
    {
        case HELP_MainWindow:
            useQuickHelpDialog = False;
            if (itemHelpFlag)
                topicToDisplay = HELP_MainWindow_ITEM_ID;
            else
                topicToDisplay = HELP_MainWindow_ID;
            break;
        case HELP_NameField:
            useQuickHelpDialog = True;
            if (itemHelpFlag)
                topicToDisplay = HELP_NameField_ITEM_ID;
            else
                topicToDisplay = HELP_NameField_ID;
            break;
        case HELP_AddressField:
            useQuickHelpDialog = True;
            if (itemHelpFlag)
                topicToDisplay = HELP_AddressField_ITEM_ID;
            else
                topicToDisplay = HELP_AddressField_ID;
    }
}

```

```

        break;          case HELP_PhoneField:
        useQuickHelpDialog = True;
        if (itemHelpFlag)
            topicToDisplay = HELP_PhoneField_ITEM_ID;
        else
            topicToDisplay = HELP_PhoneField_ID;
        break;          default:
        /* An unknown clientData was received. */
        /* Put your error handling code here. */
        return;
        break;
    }
    /* Display the topic. */
    ac = 0;
    XtSetArg (al[ac], DtNhelpType,    DtHELP_TYPE_TOPIC); ac++;
    XtSetArg (al[ac], DtNhelpVolume,  HELP_volumeName);   ac++;
    XtSetArg (al[ac], DtNhelpType,    topicToDisplay);     ac++;
    if (useQuickHelpDialog)
    {
        XtSetValues (mainQuickHelpDialog, al, ac);
        XtManageChild (mainQuickHelpDialog);
    }
    else
    {
        XtSetValues (mainHelpDialog, al, ac);
        XtManageChild (mainHelpDialog);
    }
    /* Clear the ' item help' flag. */
    itemHelpFlag = False;
}

```

上記の関数は、アプリケーションがすべてのヘルプ要求に対して2つのヘルプ・ダイアログ (mainHelpDialog と mainQuickHelpDialog) を使用し、これらのダイアログがすでに作成されていると想定しています。また、(Xt 引き数リストで使用される) al と ac が別の場所で宣言されたものと見なしています。

[ヘルプ] メニューの提供

『共通デスクトップ環境 スタイル・ガイド』では、各メニュー・バーに [ヘルプ] メニューを含むことを推奨しています。[ヘルプ] メニューには、ユーザがアプリケーションでさまざまなオンライン・ヘルプにアクセスできるように多数のコマンドが入っています。

最も重要なコマンドは次のとおりです。

■ Introduction (概要)

アプリケーションのヘルプのホーム・トピックを表示し、ユーザがハイパーリンクを使用して必要な任意の情報へナビゲートできるようにします。

- Using Help (ヘルプの使い方)

ヘルプの使い方を表示します。ユーザにヘルプ・システムの使い方を示す情報です。

- Version (バージョン)

アプリケーションのバージョン情報と著作権情報を表示します。(<copyright>要素で作成される) 著作権のトピックには、識別子 `_copyright` が付いています。

その他、特殊キーボードの使い方、アプリケーションの使い方、リファレンス、チュートリアルに関するヘルプを表示するコマンドがあります。『共通デスクトップ環境 スタイル・ガイド』のガイドラインや推奨事項の範囲で、使用する [ヘルプ] メニューが自分のアプリケーションに適応するように設計してください。

関連項目

- 64ページの「ホーム・トピックを作成するには」では、設計者がヘルプ・ポリュームのホーム・トピックを作成する方法を説明しています。
- 66ページの「メタ情報セクションを作成するには」では、設計者が著作権トピックを作成する方法を説明しています。
- 第 12 章では、ヘルプの使い方の検索方法と、アプリケーションへの追加方法を説明しています。

アイテムヘルプ・モードのサポート

一部のアプリケーションは、[ヘルプ] メニューに [アイテムヘルプ] コマンドまたは [ヘルプ・モード] コマンドを備えています。このコマンドはマウス・ポインタを一時的に ? (クエスチョン・マーク) として再定義し、ユーザに画面の項目を選択するようプロンプト要求します。項目を選択すると、その項目の説明をアプリケーションが表示します。

便利な関数に `DtHelpReturnSelectedWidgetId()` がありますが、これはポインタをクエスチョン・マークに変更してユーザがウィジェットを選択するのを待ちます。選択したウィジェットの ID が返されます。この関数は `XmTrackingLocate()` に似ていますが、ユーザがエスケープ・キーを押してオペレーションを取り消すと `NULL` を返す点が異なります。

選択した項目のヘルプを表示するには、単に返されたウィジェットのヘルプ・コールバックを呼び出すだけです。これはウィジェットの使用中にユーザが [F1] キーを押すのと同じことです。

アイテムヘルプと F1 ヘルプとを区別させる場合は、ウィジェットのヘルプ・コールバックを呼び出す前にフラグを設定します。こうすると、アイテムヘルプの結果として呼び出すコールバックを判別するのにそのフラグを使用し、それによって応答するようになります。

▼ アイテムヘルプのサポートを追加するには

1. `DtHelpReturnSelectedWidgetId()` 関数を使用して関数を記述します。この関数では、選択したウィジェットのヘルプ・コールバックを呼び出します。次のステップではこの関数を `ProcessOnItemHelp()` と呼びますが、他の名前を付けることもできます。
2. [ヘルプ] メニューに [アイテムヘルプ] というラベルの付いたコマンドを追加します。 `ProcessOnItemHelp()` 関数を呼び出すアクティブ・コールバックを追加します。
3. アイテム・ヘルプを使用するアプリケーションの各ウィジェットに、ヘルプ・コールバックを追加します。

選択したウィジェットにヘルプ・コールバックがない場合、アプリケーションはその親ウィジェットの検索を試みます。同様に、親ウィジェットにない場合はその上の階層を検索し、ヘルプ・コールバックが見つかるまで試行を繰り返します。

例

次のプロシージャは `ProcessOnItemHelp()` 関数の例ですが、これは [ヘルプ] メニューから [アイテムヘルプ] を選択すると呼び出されます。

```
void ProcessOnItemHelp(
    Widget widget)
{
    /* Declare a variable for the selected widget. */
    Widget selWidget=NULL;
    int status=DtHELP_SELECT_ERROR;
    /* Get an application shell widget from our widget hierarchy to
     * pass into DtHelpReturnSelectedWidgetId().
     */
    while (!XtIsSubclass(widget, applicationShellWidgetClass))
        widget = XtParent(widget);
```

```

status = DtHelpReturnSelectedWidgetId(widget, NULL, &selWidget);
switch ((int)status)
{
    case DtHELP_SELECT_ERROR:
        printf(`Selection Error, cannot continue\n`);
        break;
    case DtHELP_SELECT_VALID:
        /* We have a valid widget selection, now let's look for a registered help
           * callback to invoke.
        */
        while (selWidget != NULL)
        {
            if ((XtHasCallbacks(selWidget, XmNhelpCallback)
                == XtCallbackHasSome))
            {
                /* Found a help callback, so just call it */
                XtCallCallbacks((Widget)selWidget,
                    XmNhelpCallback, NULL);
                break;
            }
            else
                /* No help callback on current widget, so try the widget's parent */
                selWidget = XtParent(selWidget);
        }
        break;
    case DtHELP_SELECT_ABORT:
        printf(`Selection Aborted by user.\n`);
        break;
    case DtHELP_SELECT_INVALID:
        printf(`You must select a component within your app.\n`);
        break;
}
}

```


ヘルプ・ダイアログのイベント処理

この章では、アプリケーションが処理するヘルプ・ダイアログ・イベントについて説明します。

- 253ページの「ヘルプ・ダイアログ・イベントのサポート」
- 255ページの「ハイパーリンク・イベントへの応答」
- 256ページの「ヘルプ・ダイアログの終了の検出」
- 257ページの「アプリケーション構成ボタンの使い方」

ヘルプ・ダイアログ・イベントのサポート

アプリケーション内の他のウィジェットと同様に、ヘルプ・ウィンドウにはアプリケーションがサポートしなければならない動作がいくつかあります。

ハイパーリンク・イベント

標準ハイパーリンク・イベントのほとんどは、ヘルプ・システムによって内部的に処理されます。しかし、アプリケーションが処理するハイパーリンクは、次の4種類があります。

- 新規ウィンドウ表示ハイパーリンク

アプリケーションは新規のヘルプ・ダイアログを作成し、設計者の要求に応じて新規ヘルプ・ウィンドウにトピックを表示します。

■ マニュアル・ページ・リンク

アプリケーションは、新規の簡易ヘルプ・ダイアログを作成し(またはキャッシュから取り出し)、マニュアル・ページを表示します。通常、マニュアル・ページ・ウィンドウのサイズは、他のヘルプ・ウィンドウとは異なります。

■ アプリケーション定義リンク

アプリケーションは、これらのリンクに関連付けられているデータを解釈します。アプリケーション定義リンクは、アプリケーションの開発者とヘルプの設計者が、このリンクの作成のために共同で作業をした場合にだけ存在します。

■ テキスト・ファイル・リンク

アプリケーションは簡易ヘルプ・ダイアログを作成し(またはキャッシュから取り出し)、テキスト・ファイルを表示します。

ダイアログを終了する場合

ユーザがヘルプ・ダイアログを閉じる場合、そのキャッシュ内にダイアログを格納するかまたは破棄するかを認識する必要があります。一般ヘルプ・ダイアログでは、ヘルプ・クローズ・コールバックをサポートしています。簡易ダイアログの終了を検出するには、[閉じる] ボタンにコールバックを追加してください。

簡易ヘルプ・ボタン

簡易ヘルプ・ダイアログにあるいくつかのボタンの動作は、アプリケーションによって処理されなければなりません。これらのボタンは、必要に応じて管理することも管理しないことも可能です。起動コールバックを使用して、他のプッシュボタンのように動作を追加できます。

関連項目

- 91ページの「ハイパーリンクの作成」では、ヘルプ・システムがサポートしているリンクの種類を記述し、リンクの作成方法について説明しています。

ハイパーリンク・イベントへの応答

アプリケーションは、表示されるヘルプ・ボリューム内で使用されるハイパーリンクの種類だけは少なくともサポートする必要があります。一般的には、すべてのリンクの種類をサポートするようにしてください。

ハイパーリンクが選択されたときにアプリケーションに通知する場合、ハイパーリンク・コールバックをヘルプ・ダイアログに追加します。適切にハイパーリンクを処理するコールバック関数を指定しなければなりません。

▼ ハイパーリンク・コールバックを指定するには

1. 次のようにしてハイパーリンク・コールバックを各ヘルプ・ダイアログに追加します。

```
XtAddCallback (helpDialog, DtNhyperlLinkCallback,  
              HyperlinkCB, (XtPointer)NULL);
```

helpDialog はヘルプ・ダイアログのウィジェット ID で、*HyperlinkCB* はハイパーリンクを処理するコールバック関数名です。

2. ダイアログ内で発生する可能性があるハイパーリンク・イベントを処理するための *HyperlinkCB* 関数を指定します。

ハイパーリンク・コールバック内で、次のコールバック構造体 (<Dt/Help.h> で宣言される) にアクセスしなければなりません。

```
typedef struct  
{  
    int      reason;  
    XEvent  *event;  
    char    *locationId;  
    char    *helpVolume;  
    char    *specification;  
    int     hyperType;  
    int     windowHint;  
} DtHelpDialogCallbackStruct;
```

hyperType 要素は、実行されたリンクの種類を示します。その指定可能な値は、*DtHELP_LINK_TOPIC*、*DtHELP_LINK_MAN_PAGE*、*DtHELP_LINK_APP_DEFINE*、*DtHELP_LINK_TEXT_FILE* です。別の種類に対してどの構造体要素が有効かについては、*DtHelpDialog* (3) のマニュアル・ページを参照してください。

windowHint 要素は、ウィンドウの種類を示します。その指定可能な値は、DtHELP_CURRENT_WINDOW、DtHELP_POPUP_WINDOW、DtHELP_NEW_WINDOW です。

例

次の関数 HyperlinkCB() は、ハイパーリンク・コールバックを処理するために必要な一般構造体を示しています。

```
XtCallbackProc
HyperlinkCB (widget, clientData, callData)
    Widget      widget;
    XtPointer   clientData;
    XtPointer   callData;
    {
        DtHelpDialogCallbackStruct *hyperData =
            (DtHelpDialogCallbackStruct *) callData;
        switch ((int)hyperData-> hyperType)
        {
            case DtHELP_LINK_TOPIC:
                /* Handles "jump new view" hyperlinks. */
                break;
            case DtHELP_LINK_MAN_PAGE:
                /* Handles "man page" hyperlinks. */
                break;
            case DtHELP_LINK_APP_DEFINE:
                /* Handles "application-defined" hyperlinks. */
                break;
            case DtHELP_LINK_TEXT_FILE:
                /* Handles "text file" hyperlinks. */
                break;
            default:
                break;
        }
    }
```

ヘルプ・ダイアログの終了の検出

一般ヘルプ・ダイアログのクローズを検出するには、次のコールバックをダイアログに追加します。

```
XtAddCallback (helpDialog, DtNcloseCallback,
               HelpCloseCB, (XtPointer) NULL);
```

helpDialog はヘルプ・ダイアログのウィジェット ID で、*HelpCloseCB* はダイアログのクローズを処理するために指定したコールバック・プロシージャ名です。

簡易ヘルプ・ダイアログのクローズを検出するには、次のコールバックをダイアログの [了解] ボタンに追加します。

```
XtAddCallback (DtHelpQuickDialogGetChild (helpDialog,  
DtHELP_QUICK_OK_BUTTON), XmNactivateCallback, HelpCloseCB, (XtPointer) NULL);
```

helpDialog はヘルプ・ダイアログのウィジェット ID で、*HelpCloseCB* はダイアログのクローズを処理するために指定したコールバック・プロシージャ名です。

アプリケーション構成ボタンの使い方

簡易ヘルプ・ダイアログのアプリケーション構成ボタンにより、カスタム動作を簡易ヘルプ・ダイアログに追加します。このボタンは希望に合わせて使用できますが、その目的としては次の2つの方法のどちらかでヘルプへのパスを提供することです。

- 段階を追って情報を検索できます。これは、しばしば段階的な表示と呼ばれます。この場合、デフォルト・ボタン・ラベル (More) が適切です。
- アプリケーションのヘルプ・ボリュームを通常どおりブラウズするために一般ヘルプ・ダイアログを開くことができます。この場合、Browse... が最も適切なボタン・ラベルになります。

▼ アプリケーション構成ボタンを使用可能にするには

1. ボタンの ID を獲得します。
2. 起動コールバックをボタンに追加します。
3. ボタンを管理します。

例

次のコード・セグメントはボタンの ID を獲得し、コールバックを割り当て、ボタンを管理します。quickHelpDialog が作成されたばかりであると想定しています。

```
Widget moreButton;  
moreButton = DtHelpQuickDialogGetChild (quickHelpDialog,  
DtHELP_QUICK_MORE_BUTTON);  
XtAddCallback (moreButton, XmNactivateCallback,  
MoreHelpCB, NULL);  
XtManageChild (moreButton);
```

関連項目

- 235ページの「簡易ヘルプ・ダイアログを作成するには」
- DtHelpDialog(3) のマニュアル・ページ
- DtHelpQuickDialog(3) のマニュアル・ページ

「ヘルプの使い方」の作成

この章では、ヘルプ・システムの機能とその使用方法を記述するヘルプ・ボリュームをアプリケーションに組み込む方法について説明します。このヘルプ・ボリュームは、ヘルプ・ダイアログ・ボックスの使い方に関するヘルプを提供します。

- 259ページの「「ヘルプの使い方」の作成」
- 261ページの「アプリケーションの「ヘルプの使い方」へのアクセス」
- 264ページの「独自の「ヘルプの使い方」ボリュームを記述する」

「ヘルプの使い方」の作成

「ヘルプの使い方」はヘルプ・システムの使用方法を通知します。特に、ハイパーリンクの使用、トピックのナビゲート、索引の使用、およびヘルプ・トピックの印刷などの作業について説明しています。通常は、「ヘルプの使い方」は **Help4Help** という名前の個別のヘルプ・ボリュームとして提供されます。

Help4Help ボリュームとそのソース・ファイルは、開発者用ツールキットに組み込まれます。デフォルト・ボリュームを使用するかアプリケーションの設計に合わせて変更できます。

アプリケーション・ヘルプの場合

アプリケーション特定のヘルプを記述する場合、アプリケーションのヘルプ・ダイアログに [ヘルプの使い方] を追加する方法が次の 2 つあります。

- デスクトップ上の「ヘルプの使い方」ボリュームを使用する。

たとえば、デスクトップを実行しているワークステーションには、標準の Help4Help ボリュームがインストールされます。

- 独自の「ヘルプの使い方」ボリュームを作成する。

Help4Help ボリュームのヘルプタグ・ソース・ファイルは、C ロケールの場合、/usr/dt/dthelp/help4help/C ディレクトリに作成されます。control サブディレクトリには、ヘルプタグ処理ファイルがあります。このディレクトリ内で HelpTag を実行して実行時のヘルプ・ファイルを作成してください。「ヘルプの使い方」ボリュームで使用されているグラフィック・ファイルは、control/graphics サブディレクトリに格納されます。

スタンドアロン・ヘルプの場合

スタンドアロン・ヘルプを記述する場合、すでにインストール済みで使用準備ができてヘルプビューア・プログラムを使用します。この場合、ヘルプビューアはデフォルトとして標準の Help4Help ボリュームにアクセスするので、「ヘルプの使い方」について心配する必要はありません。

「ヘルプの使い方」の検索方法

ヘルプ・システム (ヘルプビューアを含む) を使用する各アプリケーションは、「ヘルプの使い方」トピックのためにアクセスされるヘルプ・ボリュームを識別する helpOnHelpVolume リソースを持っています。ヘルプビューアの場合、このリソースは次のように設定されます。

```
DtHelpview*helpOnHelpVolume: Help4Help
```

独自の「ヘルプの使い方」ボリュームを作成する場合、システムにインストールされる可能性のある別の「ヘルプの使い方」ボリュームと競合しないように固有の名前を付けるようにしてください。

アプリケーションの「ヘルプの使い方」へのアクセス

アプリケーションは「ヘルプの使い方」をサポートするために次の作業を実行します。

- `helpOnHelpVolume` リソースを設定し、アクセスしたいヘルプ・ボリュームを識別する。
- [ヘルプの使い方] コマンドをアプリケーションの [ヘルプ] メニューに追加する。

▼ `helpOnHelpVolume` リソースを設定するには

- ◆ アプリケーションのデフォルトのリソース・ファイルに次のような 1 行を追加します。

```
App-class*helpOnHelpVolume: volume
```

App-class はアプリケーションのクラス名、*volume* はアクセスしたい「ヘルプの使い方」ボリューム名です。

またアプリケーション内で、作成した各ヘルプ・ダイアログ `helpOnHelpVolume` リソースを設定します。

例

- `dthelpview` のデフォルトのリソース・ファイル (`DtHelpview`) にある次の行は、「ヘルプの使い方」ボリュームを指定します。

```
DtHelpview*helpOnHelpVolume: Help4Help
```

- ヘルプ・ダイアログの作成中に「ヘルプの使い方」ボリュームを作成するには、次のように作成関数に渡された引き数のリストに追加します。

```
ac = 0;  
XtSetArg (al[ac], XmNtitle, "My Application - Help"); ac++;  
XtSetArg (al[ac], DtNhelpOnHelpVolume, "Help4Help"); ac++;  
helpDialog = DtCreateHelpDialog (parent, "helpDialog", al, ac);
```

▼ [ヘルプの使い方] コマンドを指定するには

1. [ヘルプ] メニューに [ヘルプの使い方] というラベルのボタンを追加します。また、必要な起動コールバックを起動して、HelpRequestCB() 関数を呼び出します。
2. HelpRequestCB() 関数にサポートを追加し、「ヘルプの使い方」を表示します。特に次の作業を実行します。
 - 簡易ヘルプ・ダイアログを作成します。
 - ダイアログのタイトルを「ヘルプの使い方」に設定します。
 - 「ヘルプの使い方」ボリュームのホーム・トピックを表示します。
 - 簡易ヘルプ・ダイアログを管理します。

例

次の行は、HelpRequestCB() 関数を呼び出す [ヘルプの使い方] というラベルのメニュー・ボタンを作成します。

```
/* Create the ' Using Help ...' button. */

labelStr = XmStringCreateLtoR ("Using Help ...", XmSTRING_DEFAULT_CHARSET);
ac = 0;
XtSetArg (al[ac], XmNlabelString, labelStr);      ac++;
button = XmCreatePushButtonGadget (parent, "usingHelpButton", al, ac);
    XtManageChild (button);
    XmStringFree (labelStr);
    /* Add a callback to the button. */
XtAddCallback (button, XmNactivateCallback, HelpRequestCB,
    USING_HELP);
```

USING_HELP は、ユーザがメニュー・ボタンを選択したときに HelpRequestCB() 関数に渡されるクライアント・データです。通常、アプリケーションのどこか (Help.h ファイルなど) で固有の整数として定義されています。

```
#define USING_HELP 47
```

HelpRequestCB() 関数の USING_HELP の処理方法を確認するには、次の「ヘルプの使い方」を表示するには」にある例を参照してください。

▼ 「ヘルプの使い方」を表示するには

1. 簡易ヘルプ・ダイアログを作成します (またはキャッシュから取り出します)。

2. ダイアログに「ヘルプの使い方」ボリュームのホーム・トピックを表示します。

「ヘルプの使い方」は一般ヘルプ・ウィンドウに表示できます。しかし、簡易ヘルプ・ダイアログのユーザ・インタフェースはもっと単純なので、簡易ヘルプ・ダイアログを使用してください。このダイアログは、通常「ヘルプの使い方」を必要とする新しいユーザでも簡単に使用できます。

例

次のプログラム・セグメントは、HelpRequestCB() 関数の一部です。通常、ユーザは、アプリケーションの [ヘルプ] メニューから [ヘルプの使い方] を選択するか簡易ヘルプ・ダイアログの [ヘルプ] ボタンを選択するため、USING_HELP 定数はその関数に渡されます。

次の例は、アプリケーションが [ヘルプの使い方] ダイアログを 1 つしか作成せず、そのウィジェット ID を onHelpDialog と呼ばれる変数に保持するものと仮定しています。

```
case USING_HELP:
    if (onHelpDialog == (Widget)NULL)
    {
        /* Get a quick help dialog for use as the ' help on help' dialog. */
        onHelpDialog = FetchHelpDialog (True);

        if (onHelpDialog == (Widget)NULL)
            /* We didn't get a dialog! Add your error handling code here. */
    }

    /* Set the proper volume and ID to display the home topic of
       the help on help volume. Also, set the dialog's title. */
    ac = 0;  XtSetArg (al[ac], XmNtitle, "Help On Help");      ac++;
    XtSetArg (al[ac], XmNhelpType, DT_HELP_TYPE_TOPIC); ac++;
    XtSetArg (al[ac], XmNhelpVolume, "Help4Help");          ac++;
    XtSetArg (al[ac], XmNlocationId, "_hometopic");          ac++;
    XtSetValues (onHelpDialog, al, ac);

    /* If the ' help on help' dialog is already managed, it might
       be in another workspace, so unmanage it. */
    if (XtIsManaged (onHelpDialog))
        XtUnmanageChild (onHelpDialog);

    /* Manage the ' help on help' dialog. */
    XtManageChild (onHelpDialog);

    break;
```

HelpRequestCB() 関数の残りが構成される方法を確認するには、245ページの「ヘルプ・コールバックを追加するには」にある例を参照してください。

関連項目

- 235ページの「簡易ヘルプ・ダイアログを作成するには」
- 242ページの「ヘルプ・トピックを表示するには」

独自の「ヘルプの使い方」ボリュームを記述する

独自の「ヘルプの使い方」ボリュームを作成する必要がある場合、既存の Help4Help ボリュームで開始し、必要な変更を行なってください。Help4Help ボリュームを記述するのに使用されるソース・ファイルはすべて、`/usr/dt/dthelp/help4help/lang` ディレクトリに用意されています。

インストールの重複を避けるには、「ヘルプの使い方」ボリュームに Help4Help 以外の名前を付けます。自分のプロダクトに固有の名前を付けてください。たとえば、アプリケーションのヘルプ・ボリュームが Netwapp の場合は、「ヘルプの使い方」ボリュームは NetwappH4H という名前にします。

必須のエントリ・ポイント

ヘルプ・ダイアログのコンテキスト・ヘルプを正しく動作させるには、「ヘルプの使い方」ボリュームに次のエントリ・ポイント (ID) を作成しなければなりません。(これらはすでに Help4Help ソース・ファイルにあります。)

ID	トピック記述
<code>_hometopic</code>	ヘルプ・システムの使い方に関するイントロダクションを表示します。このトピックは一般ヘルプ・ダイアログの [ヘルプ] メニューから [ヘルプの使い方] を選択するか、簡易ヘルプ・ダイアログの [F1] を押すと表示されます。(ID <code>_hometopic</code> は、 <code><hometopic></code> 要素によって自動的に作成されます。)
<code>_copyright</code>	「ヘルプの使い方」ボリュームに関する商標とバージョン情報を表示します。このトピックは一般ヘルプ・ダイアログの [ヘルプ] メニューから [バージョン] を選択すると表示されます。(ID <code>_copyright</code> は、 <code><copyright></code> 要素によって自動的に作成されます。)

ID	トピック記述
history	[ヒストリ] ダイアログの使用方を記述しているトピックを表示します。このトピックは、[ヘルプ] を選択するか [ヒストリ] ダイアログの [F1] を押すと表示されます。
printing	[印刷] ダイアログの使用方を記述しているトピックを表示します。このトピックは、[ヘルプ] を選択するか [印刷] ダイアログの [F1] を押すと表示されます。
index-search	[索引検索] ダイアログの使用方を記述しているトピックを表示します。このトピックは、[ヘルプ] を選択するか [索引検索] ダイアログの [F1] を押すと表示されます。
volume-select	[検索ボリューム選択] ダイアログの使用方を記述しているトピックを表示します。このトピックは、[ヘルプ] を選択するか [検索ボリューム選択] ダイアログの [F1] を押すと表示されます。

▼ Help4Help ソース・ファイルをコピーするには

1. 次のようにコマンドを使用して、`/usr/dt/dthelp/help4help/lang` ディレクトリ全体を新しい作業ディレクトリ (*new-dir*) にコピーします。*lang* にはロケール名が入ります。

```
cp -r /usr/dt/dthelp/help4help/lang new-dir
```

これは *new-dir* を作成し、すべてのファイルとディレクトリをその中にコピーします。

2. ファイル (読み取り専用としてコピーされる) の編集をできるようにするには、次のようにコマンドを使用してアクセス権を変更します。

```
chmod -R u+w new-dir
```

Help4Help ボリュームは、これらのヘルプタグ・ソース・ファイルを使用します。

- MetaInfo
- Toc
- Tasks
- HomeTopic

- Concepts
- Reference
- Glossary

また、control ディレクトリも含まれており、ここでヘルプタグを実行して実行時のヘルプ・ファイルを作成します。グラフィック・ファイルは、control/graphics サブディレクトリに保存されます。

ヘルプタグを実行する前に Help4help.htg のファイル名が変更されていることを確認してください。「ヘルプの使い方」ボリューム名は、他の「ヘルプの使い方」ボリュームと重複しないように固有の名前でなければなりません。

例

次のコマンドは「ヘルプの使い方」ボリュームのコピーを作成し、そのファイルを書き込み可能にします。(通常は projects サブディレクトリはすでに存在します。)

```
cp -r /usr/dt/dthelp/help4help/C /users/dex/projects/NewHelp4Help
chmod -R u+w /users/dex/projects/NewHelp4Help
```

実行時のヘルプ・ファイルの新しいバージョンを作成するには、最初にディレクトリ /usr/dt/bin が検索パス内にあることを確認します。次に、新規ディレクトリに移動して Help4Help.htg のファイル名を変更し、ヘルプタグを実行します。

```
cd /users/dex/projects/NewHelp4Help
mv Help4Help.htg NewH4H.htg
dthelptag NewH4H
```

ヘルプタグ・ソフトウェアを実行すると、次のコマンドを使用して新しい「ヘルプの使い方」ボリュームを表示できます。

```
dthelpview -helpVolume NewH4H
```

インストール・パッケージの準備

この章では、アプリケーションのインストール・パッケージにあるヘルプ・ファイルについて説明します。また、アプリケーションがデスクトップに登録されるときにヘルプ・ファイルが処理される方法についても説明しています。

- 267ページの「概要」
- 268ページの「オンライン・ヘルプの配布」
- 268ページの「インストール・パッケージの作成」
- 271ページの「アプリケーションとそのヘルプの登録」
- 273ページの「製品準備チェックリスト」

概要

最後の製品を準備する段階になったら、すべてのヘルプ・ファイルが適切に作成され、インストールされていることを確認しなければなりません。製品パッケージには、実行時のヘルプ・ファイル (*volume.sd1*) とそのグラフィック・ファイルの両方が入っています。さらに、フロントパネルのヘルプ・ビューアを使用してボリュームの表示を可能にするヘルプ・ファミリ・ファイルを作成できます。

オンライン・ヘルプの配布

オンライン・ヘルプは、アプリケーションに完全に統合されるか、スタンドアロン・ヘルプ・ボリュームとして提供されます。完全に統合されたヘルプにより、[ヘルプ]メニューか[ヘルプ]キーを使用してアプリケーションからヘルプ情報に直接アクセスできます。スタンドアロン・ボリュームは一方で、デスクトップのヘルプ・ビューアを使用してしか表示できません。

システム管理者は、アプリケーションが統合されたヘルプを提供しない場合や、カスタマイズされた環境で補足的なヘルプ・ボリュームが作成されない場合に、スタンドアロン・ヘルプ・ボリュームをデスクトップに追加するように選択できます。デスクトップにスタンドアロン・ボリュームをインストールする手順については、272ページの「スタンドアロン・ヘルプ」を参照してください。

インストール・パッケージの作成

インストール・パッケージには次のヘルプ・ファイルが入っています。

- 実行時のヘルプ・ファイル
- グラフィック・ファイル
- ヘルプ・ファミリ・ファイル(オプション)
- アプリケーション・デフォルト・ファイル(オプション)

オンライン・ヘルプで使用されている実行時のヘルプ・ファイルとグラフィックは、インストール・パッケージに入っています。ヘルプ・ファミリ・ファイルは、統合されたアプリケーション・ヘルプのオプションです。しかし、アプリケーション・ヘルプをデスクトップのヘルプ・ビューアを使用してブラウザ可能にしたい場合は、ファミリ・ファイルを作成しなければなりません。スタンドアロン・ヘルプ・ボリュームを配布中の場合は、ヘルプ・ファミリ・ファイルを作成しなければなりません。125ページの「ヘルプ・ファミリを作成するには」を参照してください。

アプリケーションのヘルプ・ボリュームに実行リンクがある場合、設計者はアプリケーションのデフォルトのリソース・ファイルに実行別名を定義してください。このためには、実行別名でリンクを自動的に実行するヘルプ・システムのデフォルトの実行ポリシーを利用します。しかし、ヘルプ・ボリュームが別の情報ビューア

を使用して独立したボリュームとして表示される場合、ヘルプ・システムは実行リンクが選択された時点で確認ダイアログボックスを表示します。

図 13-1 は、アプリケーションとそのヘルプ・ファイルの典型的なインストール・パッケージを示しています。ヘルプ・ファイルは、デフォルトの言語ディレクトリ (C がデフォルト) がある別の `help` サブディレクトリにグループ化されます。実行時のヘルプ・ファイル、ファミリー・ファイル、およびグラフィック・ファイルは、このディレクトリにあります。

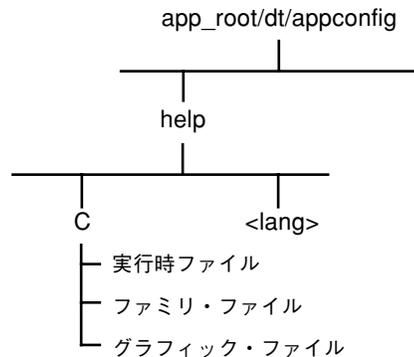


図 13-1 アプリケーション・インストール・パッケージ

アプリケーションが複数の言語でオンライン・ヘルプを提供する場合、各言語を格納するための *language* サブディレクトリを作成してください (*language* はユーザの `LANG` 環境変数に一致します)。たとえば、英語ユーザ・インタフェースとドイツ語ユーザ・インタフェースの両方を提供するアプリケーションは、英語には `C`、ドイツ語には `german` というように 2 つのサブディレクトリにそれぞれ対応するオンライン・ヘルプを保存します。

実行時のヘルプ・ファイル

ヘルプタグは単一の実行時のヘルプ・ファイル `volume.sdl` を作成します。ベース名 `volume` は `volume.htg` ファイルのベース名と同じです。ヘルプ・ビューアはマスタ・ヘルプ・ファイルに保存されている情報を使用し、関連付けられたグラフィック・ファイルにもアクセスします。

ヘルプタグ・ソフトウェアによって作成された `volume.htg` ファイルや追加のファイルを出荷する必要はありません。

グラフィック・ファイル

ヘルプ・ボリュームがグラフィックを使用している場合、イメージ・ファイルは便利のように通常別のディレクトリに保存されます。しかし、*volume.htg* ファイルと同じ場所に保存するように選択することも可能です。

実行時のヘルプ・ファイルには実際のグラフィック・イメージは入っていません。そのかわり、各グラフィック・ファイルの位置への「参照」が入っています。ヘルプタグを実行すると、*dthelptag* コンパイラはグラフィック・ファイルの相対パス名をヘルプ・ボリュームに組み込みます。

ヘルプ・ファイルがインストールされると、グラフィック・ファイルは実行時のファイルが作成されたときと同じ相対位置に位置付けられます。そうでない場合、ヘルプ・ボリュームはグラフィック・ファイルを位置付けることはできません。たとえば、グラフィック・ファイルが *volume.htg* ファイルより 1 つ下のレベルにある *graphics* という名前のサブディレクトリにある場合、インストール・パッケージはその相対位置を確保しておかなければなりません。グラフィック・ファイルは *volume.sdl* ファイルより 1 つ下のレベルにある *graphics* という名前のサブディレクトリに位置付けられなければなりません。

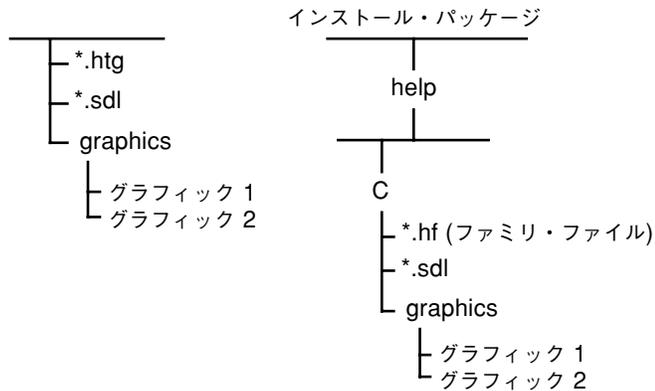


図 13-2 作成ディレクトリとインストール・パッケージの関係

ヘルプ・ファミリ・ファイル

ヘルプ・ファミリ・ファイル (*volume.hf*) をオプションとして提供できます。ファミリ・ファイルには、ヘルプ・ボリュームについての簡単な記述と著作権情報があり

ます。このファイルは、1つ以上の関連のあるボリュームを1つの製品カテゴリにグループ化するのにも使用されます。

デスクトップ・ブラウザ・ボリュームからヘルプ・ボリュームへのアクセスを可能にしたい場合、インストール・パッケージにファミリー・ファイルを作成しなければなりません。ファミリー・ファイルを作成する方法については、125ページの「ヘルプ・ファミリーを作成するには」を参照してください。

アプリケーションとそのヘルプの登録

デスクトップの統合ユーティリティ `dtappintegrate` は、インストールされたアプリケーション・ファイルと特定のデスクトップ・ディレクトリの間にはシンボリック・リンクを作成することにより、アプリケーションとそのヘルプ・ファイルに登録します。アプリケーションの登録により、ヘルプ・システムが使用するディレクトリ検索パスにヘルプ・ファイルを確実に位置付けることができます。

登録により、ヘルプ・システムの重要な2つの機能が使用可能になります。

■ 他のボリュームへのハイパーリンク

1つのヘルプ・ボリュームにあるハイパーリンクは、ボリューム名とボリュームにあるIDだけを使用して別のヘルプ・ボリュームを参照できます。転送先ボリュームが登録されている場合、リンクはボリュームをファイル・システムのどこに保存するかを指定する必要はありません。

■ ヘルプ・ファミリーのブラウズ

「ヘルプ・ファミリー」も登録する場合、ヘルプ・ボリュームはヘルプ・ビューアを使用してブラウズ可能になります。

オンライン・ヘルプを登録すると、提供したヘルプにアクセスしやすくなります。設計者とプログラマにとっては、ボリュームの実際の位置を指定しなくても、ボリュームへの参照にはボリューム名だけを使用すればいいのでより簡単です。

1つ以上のヘルプ・ボリュームがあるヘルプ・ファミリーを登録する場合、フロントパネルの[ヘルプ・ビューア]からの一般的なブラウズにヘルプを使用できるようにします。これにより、アプリケーションを使用しないでアプリケーション固有のヘルプにアクセスできます。また、スタンドアロン・ヘルプを記述している場合は、この方法がヘルプにたどりつく唯一の方法です。

スタンドアロン・ヘルプ

アプリケーションのスタンドアロン・ヘルプ・ボリュームやカスタマイズされた環境はヘルプ・システムの開発者用キットを使用して作成できます。ヘルプ・ボリュームをデスクトップ・ブラウザ・ボリュームからアクセス可能にするには、システム管理者は実行時のヘルプ・ファイル、関連付けられたグラフィック、およびファミリ・ファイルを `/etc/dt/appconfig/help/language` ディレクトリにインストールしてください。

実行時のヘルプ・ファイルとそのグラフィック・ファイルは、ヘルプ・ボリュームが作成されたときと同じ相対位置にインストールされなければなりません。グラフィック・ファイルのインストールをレビューする方法については、270ページの「グラフィック・ファイル」を参照してください。

アプリケーションの登録時に発生すること

アプリケーションの登録により、`app_root/dt/appconfig/help/language` にある実行時のヘルプ・ファイルおよびファミリと、`/etc/dt/appconfig/help/language` ディレクトリの間にはシンボリック・リンクが作成されます。

アプリケーションの登録手順の詳細は、『Solaris 共通デスクトップ環境 上級ユーザ及びシステム管理者ガイド』を参照してください。

ヘルプ・ボリュームの検索方法

ヘルプ・システムはデスクトップ検索パスを使用してヘルプ・ボリュームを位置付けます。ヘルプがアプリケーション内で要求されたり、ヘルプ・ボリュームがコマンド行に指定されたりする場合、ヘルプ・ボリュームは検索パス・ディレクトリのセットをチェックすることにより検索されます。いくつかの環境変数を変更して、ヘルプ・ボリュームのディレクトリ検索パスをコントロールできます。検索パスの指定の詳細は、『Solaris 共通デスクトップ環境 上級ユーザ及びシステム管理者ガイド』を参照してください。

製品準備チェックリスト

次のチェックリストは、製品が適切に準備されていることを確認するのに役立ちます。もちろん、ユーザが行うように製品をテストするのが一番いい方法です。

設計者の場合

1. 最後のバージョンの実行時のヘルプ・ファイルが作成された。

次の例は、実行時のファイルの作成の際に推奨するコマンドです。

```
dthelptag -clean volumedthelptag volume nomemo onerror=stop
```

-clean オプションは、上記の dthelptag コマンドからファイルを削除します。nomemo オプションは、記述者のメモが表示されないようにします。onerror=stop オプションは、パーサ・エラーが発生した場合に処理を停止します。パーサ・エラーが発生したヘルプ・ボリュームを配布してはなりません。

2. すべてのハイパーリンクをテストした。

各ハイパーリンクは適切なトピックを表示し、正しいアクションを実行します。

3. 実行別名が実行リンクに定義された。

実行別名は、アプリケーションのアプリケーション・デフォルト・ファイルにあるリソースとして定義されます。実行別名は実行されるシェル・コマンドと名前とを関連付けます。ヘルプ・ボリュームで実行リンクを使用した場合は、アプリケーション開発者がこれらのリソースをアプリケーション・デフォルト・ファイルに追加するように調整してください。詳細は、100ページの「実行別名」を参照してください。

4. すべてのグラフィックが受け入れ可能である。

グラフィックは、カラー、グレースケール、およびモノクロ・ディスプレイでテストされました。

製品のインテグレータの場合

1. 実行時のファイルがインストールされた。
2. すべてのグラフィックが適切な位置にインストールされた。
各グラフィック・ファイルは、ヘルプタグ・ソフトウェアが実行されたときに .htg ファイルとは相対的な位置にあった .sdl ファイルに対しても、同じ相対的な位置にインストールされなければなりません。
3. ヘルプ・ボリュームが登録された。
dtappintegrate スクリプトが、インストール・ディレクトリから登録ディレクトリにシンボリック・リンクを作成するために実行されました。
4. 製品のファミリ・ファイルがインストールおよび登録された。
ファミリ・ファイルは他のヘルプ・ファイルと共にインストールされます。dtappintegrate を実行すると、ファミリ・ファイルのシンボリック・リンクを作成します。ヘルプ・ボリュームのためにファミリ・ファイルを登録するのは任意です。しかし、ファミリ・ファイルを登録しないと、ヘルプ・ボリュームはフロントパネルの [ヘルプ・ビューア] からはアクセスできません。

プログラマの場合

1. アプリケーションは、次の必須リソースに対して正しい値を設定している。

```
App-class*helpVolume:      volume
App-class*helpOnHelpVolume: help-on-help-volume
```

helpVolume リソースは、アプリケーションのためのヘルプ・ボリュームを識別します。helpOnHelpVolume は、ヘルプ・システムの使用に関するヘルプがあるヘルプ・ボリュームを識別します。

2. 実行別名は、アプリケーションのデフォルトのリソース・ファイルにある。
設計者は、アプリケーションのリソースとして実行別名を定義します。実行別名は、実行するシェル・コマンドと名前とを関連付けます。実行リンクがヘルプ・ボリュームで使用されていた場合は、追加の必要があるリソースを識別していることを設計者が確認するようにしてください。詳細は、100ページの「実行別名」を参照してください。

3. アプリケーションが次のオプションのリソースに対して希望する値を設定している。

```
App-class*DtHelpDialogWidget*onHelpDialog*rows:      rows
App-class*DtHelpDialogWidget*onHelpDialog*columns:    columns
App-class*DtHelpDialogWidget*definitionBox*rows:      rows
App-class*DtHelpDialogWidget*definitionBox*columns:    columns
```

onHelpDialog リソースは、「ヘルプの使い方」を表示するのに使用される簡易ヘルプ・ダイアログのサイズをコントロールします。definitionBox リソースは、定義リンクに使用される簡易ヘルプ・ダイアログのサイズをコントロールします。

4. アプリケーションは、デフォルトのフォント・リソースを使用するか、アプリケーションのデフォルトのリソース・ファイルにフォント・リソースを定義している。

ほとんどの場合、アプリケーションはデフォルトのフォント・リソースを使用できます。しかし、カスタム・フォントが使用される場合、デフォルトのリソース・ファイルに定義されていなければなりません。サンプルのフォント・スキーマが /usr/dt/dthelp/fontschemes ディレクトリに用意されています。フォント・スキーマの詳細は、第 14 章を参照してください。

パート **IV** 国際化対応

母国語のサポート

この章では、ヘルプ・ボリュームを複数の言語で提供する場合に変更が必要なヘルプ・システムが使用するファイルを説明します。

- 279ページの「国際化されたオンライン・ヘルプ」
- 280ページの「国際化の要因」
- 290ページの「フォント・スキーマの理解」
- 294ページの「フォーマット・テーブルの作成」
- 296ページの「ローカライズされたヘルプ・ボリュームの表示」
- 296ページの「オンライン・ヘルプの国際化対応の確認事項」

国際化されたオンライン・ヘルプ

外国のユーザ向けの製品の場合、オンライン・ヘルプをユーザの母国語で提供することが大切です。ヘルプ・システムは、オンライン・ヘルプの記述と表示を実質的にどの言語でもサポートします。

実行時のヘルプ・ファイルを作成するためにヘルプ・ボリュームを処理する場合、ファイルの記述にどの言語および文字セットを使用したかをヘルプタグ・ソフトウェアに必ず指示しなければなりません。言語および文字セットの情報は、ヘルプ・ボリュームを表示するのに適切なフォントを決定するためにも使用されます。

国際化の要因

次節で説明するいくつかの要因は、オンライン・ヘルプをユーザの母国語で提供するのに役立ちます。

文字セットおよびマルチバイト文字

文字セットは、文字を認識するためのコンピュータの内部文字コードのマッピング方法を決定します。ほとんどの言語は、1つの文字セット全体を表すにはシングル・バイト文字で十分です。しかし、何千もの文字を使用する言語もあります。そのような言語は、1つ1つの文字を一意に表すのに2、3、4バイトを必要とします。

ヘルプ・システムがサポートする文字セットを表14-1にリストします。しかし、すべてのプラットフォームに存在するわけではない文字セットもあります。

表 14-1 共通デスクトップ環境文字セット

言語	文字セット名	説明
西ヨーロッパおよび南北アメリカ	ISO-8859-1	ISO ラテン 1
	HP-ROMAN8	HP ローマン
	IBM-850	PC 多国語
中央ヨーロッパ	ISO-8859-2	ISO ラテン 2
キリル文字	ISO-8859-5	ISO ラテン / キリル文字
アラビア語	ISO-8859-6	ISO ラテン / アラビア語
	HP-ARABIC8	HP アラビア語 8
	IBM-1046	PC アラビア語

表 14-1 共通デスクトップ環境文字セット 続く

言語	文字セット名	説明
ヘブライ語	ISO-8859-8	ISO ラテン / ヘブライ語
	HP-HEBREW8	HP ヘブライ語 8
	IBM-856	PC ヘブライ語
ギリシャ語	ISO-8859-7	ISO ラテン / ギリシャ語
	HP GREEK8	HP ギリシャ語 8
トルコ語	ISO-8859-9	ISO ラテン 5
	HP-TURKISH8	HP トルコ語 8
日本語	EUC-JP	日本語 EUC (JIS X-0201, JIS X-0208, JIS X-0212)
	HP-SJIS	HP 日本語 シフト JIS
	HP-KANA8	HP 日本語 カタカナ 8 (JIS X-0201 1976)
	IBM-932	PC 日本語 シフト JIS
韓国語	EUC-KR	韓国語 EUC
中国語	EUC-CN	簡体字 EUC (中華人民共和国) (GB2312)
	EUC-TW	繁体字 EUC (台湾) (CNS 11643.*)
	HP-BIG5	HP 繁体字 Big5
	HP-CCDC	HP 繁体字 CCDC
	HP-15CN	HP 繁体字 EUC

表 14-1 共通デスクトップ環境文字セット 続く

言語	文字セット名	説明
タイ語	TIS-620	タイ語

ヘルプタグ・ファイルを記述する際、どんなヘルプ・テキストにもマルチバイト文字を使用できます。しかし、ヘルプタグのマークアップ自体(タグ名、エンティティ名、ID など)は、8ビット文字で入力しなければなりません。

言語および地域名

言語を選択する場合は、文字セットと言語および地域名の両方を選択します。言語および地域名は、指定された国または地域に対して、通貨や日付の書き方などの違いを適用するために使用されます。

ヘルプ・システムがサポートする言語および地域名を表 14-2 にリストします。言語を選択する前に、ご使用のプラットフォームでサポートされる言語および文字セットを識別するために、システムに付属のマニュアルを参照してください。

表 14-2 ヘルプ・システムの言語および地域名

言語	言語/地域	言語	地域
標準	C	C	
	POSIX	C	
西ヨーロッパ/南北アメリカ			
	da_DK	デンマーク語	デンマーク
	de_AT	ドイツ語	オーストリア
	de_CH	ドイツ語	スイス
	de_DE	ドイツ語	ドイツ

表 14-2 ヘルプ・システムの言語および地域名 続く

言語	言語／地域	言語	地域
	en_AU	英語	オーストラリア
	en_CA	英語	カナダ
	en_DK	英語	デンマーク
	en_GB	英語	イギリス
	en_IE	英語	アイルランド
	en_MY	英語	マレーシア
	en_NZ	英語	ニュージーランド
	en_US	英語	アメリカ
	es_AR	スペイン語	アルゼンチン
	es_BO	スペイン語	ボリビア
	es_CL	スペイン語	チリ
	es_CO	スペイン語	コロンビア
	es_CR	スペイン語	コスタリカ
	es_EC	スペイン語	エクアドル
	es_ES	スペイン語	スペイン
	es_GT	スペイン語	グアテマラ
	es_MX	スペイン語	メキシコ
	es_PE	スペイン語	ペルー
	es_UR	スペイン語	ウルグアイ
	es_VE	スペイン語	ベネズエラ
	et_EE	エストニア語	エストニア

表 14-2 ヘルプ・システムの言語および地域名 続く

言語	言語／地域	言語	地域
	fi_FI	フィンランド語	フィンランド
	fo_FO	フェロー語	フェロー諸島
	fr_BE	フランス語	ベルギー
	fr_CA	フランス語	カナダ
	fr_CH	フランス語	スイス
	fr_FR	フランス語	フランス
	is_IS	アイスランド語	アイスランド
	it_CH	イタリア語	スイス
	it_IT	イタリア語	イタリア
	kl_GL	グリーンランド語	グリーンランド
	lt_LT	リトアニア語	リトアニア
	lv_LV	ラトビア語	ラトビア
	nl_BE	オランダ語	ベルギー
	nl_NL	オランダ語	オランダ
	no_NO	ノルウェー語	ノルウェー
	pt_BR	ポルトガル語	ブラジル
	pt_PT	ポルトガル語	ポルトガル
	sv_FI	スウェーデン語	フィンランド
	sv_SE	スウェーデン語	スウェーデン
中央ヨーロッパ			
	cs_CS	チェコ語	

表 14-2 ヘルプ・システムの言語および地域名 続く

言語	言語/地域	言語	地域
	hr_HR	クロアチア語	クロアチア
	hu_HU	ハンガリー語	ハンガリー
	pl_PL	ポーランド語	ポーランド
	ro_RO	ルーマニア語	ルーマニア
	sh_YU	セルボクロアチア語	ユーゴスラビア
	si_CS	スロベニア語	
	si_SI	スロベニア語	
	sk_SK	スロバキア語	
キリル文字			
	bg_BG	ブルガリア語	ブルガリア
	mk_MK	マケドニア語	
	ru_RU	ロシア語	
	ru_SU	ロシア語	
	sp_YU	セルビア語	ユーゴスラビア
アラビア語 ¹			
	ar_SA	アラビア語	
	ar_AA	アラビア語	
	ar_DZ	アラビア語	
ヘブライ語			

表 14-2 ヘルプ・システムの言語および地域名 続く

言語	言語／地域	言語	地域
	iw_IL	ヘブライ語	イスラエル
ギリシャ語			
	el_GR	ギリシャ語	ギリシャ
トルコ語			
	tr_TR	トルコ語	トルコ
アジア			
	ja_JP	日本語	日本
	ko_KR	韓国語	韓国
	zh_CN	中国語	中華人民共和国
	zh_TW	中国語	台湾
タイ語			
	th_TH	タイ語	タイ

1. 世界中でアラビア語を話す地域に対する ISO 地域名は存在しません。ベンダは、共通デスクトップ環境で使用するために採用した独自の地域名を供給してきました。

ロケールおよび文字セット

ヘルプ・ボリュームのデフォルトの言語および文字セットは、エンティティとして `helplang.ent` ファイルに定義できます。完全なロケール名を指定するには、次の形式を使用して、言語および地域名と文字セット名を組み合わせてください。

`language-and-territory-name.character-set-name`

`helplang.ent` ファイルについては、289ページの「`helplang.ent` ファイル」を参照してください。

例

- 次のエンティティ宣言は、C 標準言語の完全なロケール名と ISO-8859-1 文字セットを指定します。

```
<!ENTITY LanguageElementDefaultLocale SDATA ``C.ISO-8859-1``>
```

- 上記の情報を、次のように2つのエンティティ宣言を使用して入力することもできます。

```
<!ENTITY LanguageElementDefaultLocale SDATA ``C``>  
<!ENTITY LanguageElementDefaultCharset SDATA ``ISO-8859-1``>
```

- 同じ文字セットを使用してドイツ語を指定するには、次のように宣言を行います。

```
<!ENTITY LanguageElementDefaultLocale SDATA ``de_DE.ISO-8859-1``>
```

- または、EUC-JP 文字セットを使用して日本語を指定するには、次のように宣言を行います。

```
<!ENTITY LanguageElementDefaultLocale SDATA ``ja_JP.EUC-JP``>
```

ロケールが `helplang.ent` ファイルに指定されていない場合は、LANG 環境変数から値が引き出されます。

ヘルプタグ・ソフトウェア

実行時のヘルプ・ファイルを作成するためにヘルプ・ボリュームを処理する場合、ファイルの記述にどの言語および文字セットを使用したかをヘルプタグ・ソフトウェアに必ず指示しなければなりません。言語および文字セットの情報は、ヘルプ・トピックを表示するのに適切なフォントを決定するために使用されます。言語および文字セットを指定しない場合、ヘルプタグはデフォルト (英語および ISO-8859-1) を選択します。

言語および文字セットは `helplang.ent` ファイルに定義できます (詳細は、289 ページの「`helplang.ent` ファイル」を参照してください)。また、文字セットは端末エミュレータで `dthelptag` を実行するときにオプションとしてコマンド行に指定できます。

注 - ヘルプタグ・ファイルを記述する際、どんなヘルプ・テキストにもマルチバイト文字を使用できます。しかし、ヘルプタグ・マークアップ自体(タグ名、エンティティ名、ID など)は、8 ビット文字で入力しなければなりません。

DtHelp メッセージ・カタログ

ヘルプ・ダイアログのメニュー、ボタン、ラベルもユーザの母国語で表示されるべきです。これを実現するために、ヘルプ・ダイアログはそれらの文字列を DtHelp.cat という名前のメッセージ・カタログから読み取ります。

メッセージ・カタログのソース・ファイル DtHelp.msg には、メニュー、ボタン、メッセージの文字列が指定されています。必要な言語が提供されていない場合は、サンプルのメッセージ・カタログ (/usr/dt/dthelp/nls/C/DtHelp.msg) を翻訳し、次に gencat コマンドで実行時のメッセージ・カタログ・ファイルを作成しなければなりません。手順については、296ページの「メッセージ・カタログを作成するには」を参照してください。

新規のメッセージ・カタログをインストールする正しいディレクトリを決定するには、システムに付属のマニュアルを参照してください。

LANG 環境変数

ユーザの LANG 環境変数は、次の2つの理由により重要です。

- LANG の値は正しいヘルプ・ボリュームの位置を見つけるのに使用されます。
- ヘルプ・トピックが表示される時、正しいフォントおよび書式化の規則はユーザの LANG 変数に基づいて選択されます。これは、ヨーロッパやアメリカの言語よりも複雑な行の折り返し規則を持つアジアの言語において特に重要です。

関連項目

- 『共通デスクトップ環境 プログラマーズ・ガイド(国際化対応編)』
- ご使用のオペレーティング・システム用の NLS 関連のマニュアルまたはプログラマ用ツールキット

helplang.ent ファイル

helplang.ent ファイルは、ヘルプ・ボリュームのデフォルトのロケールと文字セットを決定するために、ヘルプタグ・ソフトウェアが使用するテキスト・エンティティを定義します。ヘルプ・ボリュームの言語および文字セットの指定方法については、286ページの「ロケールおよび文字セット」を参照してください。

helplang.ent ファイルは、注、注意、警告などのデフォルト文字列のテキスト・エンティティも定義します。ヘルプタグ・ソフトウェアに組み込まれている英語の文字列を無効にしたい場合は、ファイルをコピーして文字列をローカライズしてください。ファイルはディレクトリ /usr/dt/dthelp/dthelptag にあります。

helplang.ent ファイルからの抜粋を次に示します。

```
<!ENTITY LanguageElementDefaultLocale      SDATA ``C.ISO-8859-1''>
<!ENTITY NoteElementDefaultHeadingString    SDATA ``NOTE''>
<!ENTITY CautionElementDefaultHeadingString SDATA ``CAUTION''>
<!ENTITY WarningElementDefaultHeadingString SDATA ``WARNING''>
<!ENTITY ChapterElementDefaultHeadingString SDATA ``Chapter''>
<!ENTITY FigureElementDefaultHeadingString  SDATA ``Figure''>
<!ENTITY GlossaryElementDefaultHeadingString SDATA ``Glossary''>
.
.
```

フォーマット・テーブル

日本語や中国語などのマルチバイト言語には、フォーマット・テーブルが必要です。フォーマット・テーブルは、行頭にできない文字と行末にできない文字のリストを提供します。ヘルプ・ファイルが処理されるときに、フォーマット・テーブルは行が確実に正しく折り返すようにします。294ページの「フォーマット・テーブルの作成」で、新規テーブルの作成方法と、ヘルプ開発者用キットで提供されるサンプル・テーブルの編集方法を説明しています。

フォント・スキーマ

ヘルプタグ・ソフトウェアの主な機能の1つは、マークアップされたファイルを、ヘルプ・システムが解釈できる実行時の形式に変換することです。テキストは、型ファミリー、サイズ、傾斜、線の太さなどの特定の属性を指定してフォーマットされています。フォント・スキーマは、別名のように、ヘルプ・システムがヘッダ、プロシージャ、リストなどのヘルプタグ要素にフォントを割り当てるために使う単なる名前です。フォント・スキーマにより、ヘルプ・システムが使用する一かたまりのテキスト属性を特定のフォントにマップできます。

標準の共通デスクトップ環境フォントを使用するアプリケーションは、追加のフォント・リソースを定義する必要はありません。それ以外のフォント・セットにアプリケーションが依存する場合は、そのアプリケーション用にフォント・スキーマを作成して追加しなければなりません。

関連項目

- [DtStdInterfaceFontNames \(5\) のマニュアル・ページ](#)
- [DtStdAppFontNames \(5\) のマニュアル・ページ](#)

フォント・スキーマの理解

ヘルプタグ・マークアップ言語を使用してヘルプ・ボリュームを記述する場合は、テキストのフォントおよびサイズは指定しません。ヘルプタグ・ソフトウェアを起動すると、入力した要素が、テキスト属性を含む実行時のヘルプ・ファイルにフォーマットされます。

フォント・スキーマは、テキスト属性を実際のフォント指定にマップします。たとえば、ヘルプ・トピックがボールドの **sans serif** (サンセリフ) フォントを使用してフォーマットされている場合、フォント・スキーマは、どの共通デスクトップ環境標準フォントまたは X フォントが実際にテキスト表示に使用されているかを識別します。

フォント・スキーマを使用する主な目的は、フォント・サイズの選択肢を提供することです。ヘルプタグ・ソフトウェアはほとんどのトピックの本文を 10 ポイントにフォーマットします。しかし、実際に表示されるフォントは使用されているフォント・スキーマが決定するため、10 ポイントのテキストはすべて 14 ポイント・フォントを使用するよう指定されます。

フォント・リソース

各フォント・スキーマは、実際は X リソースのセットです。X リソースは、ヘルプを表示しているアプリケーションによって読み取られます。フォント・スキーマを変更したい場合は、フォント・リソースをアプリケーションのデフォルトのリソース・ファイルに設定できます。

フォント・スキーマ内の各リソースの一般的な形式は次のとおりです。

**pitch.size.slant.weight.style.lang.char-set: font specification*

<i>pitch</i>	文字間の水平方向の間隔を指定します。このフィールドは、 p (プロポーショナル) か m (モノスペース) のいずれかです。
<i>size</i>	希望するフォントの高さを指定します。ヘルプタグでフォーマットされたヘルプ・ファイルは、この値は 6、8、10、12、14 のいずれかです。
<i>slant</i>	希望するフォントの傾斜を指定します。このフィールドは通常、垂直な文字なら roman 、傾斜した文字なら italic のいずれかになります。
<i>weight</i>	希望するフォントの線の太さを指定します。このフィールドは通常、 medium か bold のいずれかです。
<i>style</i>	希望するフォントの一般的なスタイルを指定します。ヘルプタグでフォーマットされたヘルプ・ファイルのこの値は serif か sans_serif のいずれかです。
<i>lang</i>	この言語を使用してコンパイルされたボリュームが、これらのフォントを使用するように指定します。通常、指定された char_set を使用するすべてのボリュームがこれらのフォントを使用するように、* (アスタリスク) をエントリに使用します。
<i>char-set</i>	ヘルプ・テキストの記述に使用する文字セットを指定します。この値は、ヘルプタグを実行したときに指定した文字セットと一致しなければなりません。デフォルトは ISO-8859-1 です。一部の特殊文字は symbol 文字セットを使用して表示されます。

* (アスタリスク) を使用すると、特定の属性のすべての値を持つフォントを指定できます。たとえば、記号セット (特殊文字と特殊記号) はサイズと文字セットのみに基づいて一意のフォントを識別します。

フォント・リソースはフォント・スキーマ内で次のように表示されます。

```
*.6.*.*.*.DT-SYMBOL-1:
  -adobe-symbol-medium-r-normal-*-*60*-*p-*adobe-fontspecific
*.8.*.*.*.DT-SYMBOL-1:
  -adobe-symbol-medium-r-normal-*-*80*-*p-*adobe-fontspecific
*.10.*.*.*.DT-SYMBOL-1:
  -adobe-symbol-medium-r-normal-*-*100*-*p-*adobe-fontspecific
*.12.*.*.*.DT-SYMBOL-1:
  -adobe-symbol-medium-r-normal-*-*120*-*p-*adobe-fontspecific
*.14.*.*.*.DT-SYMBOL-1:
  -adobe-symbol-medium-r-normal-*-*140*-*p-*adobe-fontspecific
```

[*char-set*] フィールドは * (アスタリスク) を使用できない唯一のフィールドです。

日本語や韓国語などのマルチバイト言語を表示するには、フォント・セットを使用してフォント・リソースを指定しなければなりません。フォント・セットは、実際にはフォントのグループです。フォント・セットのリソース・エントリは、複数のフォント名が、(カンマ) で区切られ、指定が:(コロン) で終了する点を除き、単一のフォントと同様です。次の例は、日本語フォント・セット用の完全に指定したフォント・リソースです。

```
bridge-gothic-medium-r-normal--18-180-75-75-c-80-jisx0201.1976-0,
bridge-gothic-medium-r-normal--18-180-75-75-c-160-jisx0208.1983-0,
bridge-gothic-medium-r-normal--18-180-75-75-c-160-jisx0212.1990-0:
```

マルチバイト言語のフォントを指定するには、最小限の XLFD フォント指定を提供し、システムがフォント・セットを生成するために文字セットの値を指定できるようにすることも可能です。

```
*.12.roman.medium.*.ja_JP.EUC-JP: -*-*-*-*-*120*-*-*-*-*:
```

フォント・セットを指定するときには、必ず指定を:(コロン) で終了してください。これにより、ヘルプ・システムは情報を表示するためのフォント・セットをロードします。フォント・セットはマルチバイト言語の表示に使用されます。シングルバイト情報を持つボリュームについては、標準フォント指定を使用してください。

フォント・スキーマの例

/usr/dt/dthelp/fontschemes ディレクトリには次の 4 つのフォント・スキーマが格納されています。

fontDef.fns	ヘルプ・システムが使用するデフォルト・フォント
fontLarge.fns	大きなフォントの例

fontMulti.fns マルチバイト・フォントの例

fontX11.fns X11 フォントの例

▼ フォント・スキーマを選択するには

- ◆ オンライン・ヘルプを表示するアプリケーション用のデフォルトのリソース・ファイルを編集します。(もしあれば) 現在のフォント・リソースを新規スキーマに置換します。

自分だけのためにこの変更を行う場合は、デフォルトのリソース・ファイルを自分のホーム・ディレクトリにコピーしてから編集を行なってください。

例

DtStopWatch という名前の個人用アプリケーションの大きいサイズのフォントを(ヘルプ・ダイアログで) 使用するには、次の手順を実行します。

ホーム・ディレクトリへ移動します。

```
cd
```

次に、アプリケーションのデフォルトのリソース・ファイル DtStopWatch をコピーし、書き込み可能にします。

```
cp /usr/dt/app-defaults/C/DtStopWatch .  
chmod u+w DtStopWatch
```

最大のスキーマ (fontLarge.fns) を追加するよう DtStopWatch ファイルを編集します。ファイルの最後に次のファイルの内容を挿入します。

```
/usr/dt/dthelp/fontschemas/fontLarge.fns
```

新しい DtStopWatch ファイルを保存します。

DtStopWatch アプリケーションを起動し、[ヘルプ] を選択し、ヘルプ・トピックが新しいフォント・スキーマを使用して表示されるかを確認します。

フォーマット・テーブルの作成

日本語や中国語などのマルチバイト言語にはフォーマット・テーブルが必要です。フォーマット・テーブルには3つのメッセージ・セットが含まれています。1番目のセットは行頭にできない文字から成ります。2番目のセットは行末にできない文字をリストしています。3番目のセットは、シングルバイト文字とマルチバイト文字の間に発生する改行文字の処理方法を示しています。

フォーマット・テーブルはASCIIファイルで、ファイル名は必ず拡張子 `.msg` が付きます。図 14-1 は中国語 (簡体字) 用のフォーマット・テーブルからの抜粋です。

タを使用してください。マルチバイト言語でヘルプ情報を作成したい場合は、フォーマット・テーブルを作成する必要があります。

▼ メッセージ・カタログを作成するには

DtHelp.msg ファイルを翻訳したり、新しいフォーマット・テーブルを作成したり、テーブルの例 (fmt_tbl.msg) を変更したりした場合は、ヘルプ・システムが使用するメッセージ・カタログを必ず更新してください。

- ◆ カatalog・ファイルを生成するには、次のコマンド構文を使用します。

```
gencat file.cat file.msg
```

標準デスクトップ・アプリケーションのメッセージ・カタログは /usr/dt/lib/nls/msg/lang ディレクトリに位置します。メッセージ・カタログをインストールするには、ご使用のオペレーティング・システムのマニュアルを参照してください。

関連項目

- gencat (1) のマニュアル・ページ

ローカライズされたヘルプ・ボリュームの表示

現在のシステムとは異なるロケールのために作成されたヘルプ・ボリュームを表示するには、そのヘルプ・ボリュームに合うように LANG 環境変数を設定しなければなりません。LANG 環境変数の値はプラットフォーム固有です。この変数がよくわからない場合は、環境を設定するための正しい値およびプロシージャをシステム管理者に確認してください。

オンライン・ヘルプの国際化対応の確認事項

オンライン・ヘルプを外国のユーザに提供する場合に確認すべき事項を次のチェックリストにまとめます。

- ヘルプ・トピックは外国のユーザを想定して書かれているか

- /usr/dt/dthelp/dthelptag/helplang.ent ファイルをコピーして、その中の文字列エンティティをローカライズしたかこのファイルのエンティティを使用して、ヘルプタグ・ソフトウェアに組み込まれているデフォルトの文字列を無効にできます。
- ヘルプタグ・ソフトウェアは正しい文字セットと言語のオプションで実行されたか別の文字セットで記述する場合は、DtHelp.msg メッセージ・カタログ・ファイルを翻訳して、新しい文字セットをサポートするフォント・スキーマを提供しなければなりません。
- ヘルプ・テキストにマルチバイト文字を使用しているも、ヘルプタグ・マークアップ内ではすべてのタグ名、エンティティ名、ID が 8 ビットの文字セットで入力されているか
- ユーザの LANG 環境変数が正しい言語に設定されているとき、ヘルプ・ファイルを適切に見つけて表示できるようにインストールしてあるか
- ヘルプ・システムをアプリケーションに統合した場合、XtSetLanguageProc() 関数を使ってロケールを適切に設定したか

関連項目

- 272ページの「ヘルプ・ボリュームの検索方法」
- XtSetLanguageProc(3) のマニュアル・ページ
- gencat(1) のマニュアル・ページ
- ご使用のコンピュータオペレーティング・システム用の NLS 関連のマニュアルまたはプログラム用ツールキット

ヘルプタグ 1.3 DTD

ヘルプタグ文書型定義 (DTD) は、各ヘルプタグ要素と、要素を使用するための構文を定義します。DTD の詳細は、第 8 章を参照してください。

ヘルプタグ 1.3 DTD は開発者用ツールキットでも使用できます。DTD は DTD ディレクトリにあり、名前は DTD です。

```
HelpTag 1.3 DTD
<!SGML ``ISO 8879:1986''
-- SGML Declaration--
CHARSET
BASESET ``ISO 646-1983//CHARSET International Reference Version
        (IRV)//ESC 2/5 4/0''
DESCSET   0      9      UNUSED
           9      2      9
           11     2      UNUSED
           13     1      13
           14     18     UNUSED
           32     95     32
           127    1      UNUSED
BASESET ``ISO Registration Number 100//CHARSET ECMA-94
        Right Part of Latin Alphabet Nr. 1//ESC 2/13 4/1''
DESCSET   128    32     UNUSED
           160    5      32
           165    1      UNUSED
           166    88     38
           254    1      127
           255    1      UNUSED
CAPACITY SGMLREF
TOTALCAP 350000
ENTCAP   100000
ENTCHCAP 50000
ELEMCAP  50000
GRPCAP   210000
EXGRPCAP 50000
EXNMCAP  50000
ATTCAP   50000
ATTCHCAP 50000
AVGRPCAP 50000
```

```

NOTCAP      50000
NOTHCAP    50000
IDCAP      50000
IDREFCAP   50000
MAPCAP     210000
LKSETCAP   50000
LKNMCAP    50000
SCOPE DOCUMENT
SYNTAX -- The Core Reference Syntax except with ATT CNT,LITLEN,
          NAMELEN,GRPCNT, and GRPGTCNT changed --
SHUNCHAR CONTROLS  0  1  2  3  4  5  6  7  8  9
                   10 11 12 13 14 15 16 17 18 19
                   20 21 22 23 24 25 26 27 28 29
                   30 31 127 255

BASESET `ISO 646-1983//CHARSET International Reference Version
          (IRV)//ESC 2/5 4/0`
DESCSET  0      128      0
FUNCTION  RE      13
          RS      10
          SPACE   32
          TAB     SEPCHAR  9

NAMING
LCNMSTRT `````
UCNMSTRT `````
LCNMCHAR ```-.'`
UCNMCHAR ```-.'`
NAMECASE
GENERAL YES
ENTITY YES

DELIM
GENERAL SGMLREF
SHORTREF SGMLREF -- Removed short references --
NAMES SGMLREF
QUANTITY SGMLREF
ATT CNT  140
LITLEN   4096
NAMELEN  64
GRPCNT   100
GRPGTCNT 253
TAGLVL   48

FEATURES
MINIMIZE
DATATAG NO
OMITTAG NO
RANK NO
SHORTTAG YES

LINK
SIMPLE NO
IMPLICIT NO
EXPLICIT NO

OTHER
CONCUR NO
SUBDOC NO
FORMAL NO
APPINFO NONE

>
<!DOCTYPE helpvolume [
<!ELEMENT helpvolume - - (metainfo?,
                           hometopic?,

```

```

        (chapter* | (s1*, rsect*)),
        message?,
        glossary?)
+ (memo | idx) >
<!ELEMENT metainfo - - (idsection, abstract?, otherfront*)
- (footnote) >
<!ELEMENT idsection - - (title, copyright?) >
<!ELEMENT title - - (parttext)
- (memo | location | idx) >
<!ELEMENT parttext - - ((#PCDATA | acro | emph | computer |
user | term | var | circle |
quote | keycap | graphic | super |
sub | book | xref | footnote |
esc | link | location | newline )*) >

<!ELEMENT acro - - ((#PCDATA | esc | super | sub)*) >
<!ELEMENT emph - - (parttext) - (emph) >
<!ELEMENT computer - - ((#PCDATA | quote | var | user | esc)*) >
<!ELEMENT user - - ((#PCDATA | var | esc)*) >
<!ELEMENT term - - (parttext)
- (emph | computer | term | var |
quote | user | book | footnote) >
<!ATTLIST term base CDATA #IMPLIED
gloss (gloss | nogloss) gloss >
<!ELEMENT var - - ((#PCDATA | esc)*) >
<!ELEMENT circle - - CDATA >
<!ELEMENT quote - - (parttext) - (quote) >
<!ELEMENT keycap - - ((#PCDATA | super | sub | esc)+) >
<!ELEMENT graphic - O EMPTY >
<!ATTLIST graphic id ID #IMPLIED
entity ENTITY #REQUIRED >
<!ELEMENT super - - (#PCDATA) >
<!ELEMENT sub - - (#PCDATA) >
<!ELEMENT book - - (parttext) - (book) >
<!ELEMENT xref - O EMPTY >
<!ATTLIST xref id IDREF #REQUIRED >
<!ELEMENT footnote - - (p+) - (footnote) >
<!ELEMENT esc - - CDATA >
<!ELEMENT link - - (parttext) - (link | xref) >
<!ATTLIST link hyperlink CDATA #REQUIRED
type (jump |
jumpnewview |
definition |
execute |
appdefined |
man )
description CDATA #IMPLIED >
<!ELEMENT location - - (parttext) - (location) >
<!ATTLIST location id ID #REQUIRED >
<!ELEMENT copyright - - (text)
- (memo | location | idx) >
<!ELEMENT text - - ((p | note | caution | warning |
lablist | list | ex | vex |
esc | otherhead | procedure | syntax |
figure | image )*) >
<!ELEMENT p - - (head?, parttext)
+ (newline) >
<!ATTLIST (p | image) indent (indent) #IMPLIED
id ID #IMPLIED
gentity ENTITY #IMPLIED

```

```

gposition      (left | right)      left
ghyperlink    CDATA                #IMPLIED
glinktype      (jump                |
                jumpnewview         |
                definition           |
                execute              |
                appdefined           |
                man                   )      jump
gdescription   CDATA                #IMPLIED >
<!ELEMENT head      - - (parttext)
- - (memo | location | idx) >
<!ELEMENT newline   - O EMPTY >
<!ELEMENT (note     |
           caution  |
           warning ) - - (head?, text)
- - (note | caution | warning | footnote) >
<!ELEMENT lablist   - - (head?, labheads?, lablistitem+) >
<!ATTLIST lablist   spacing      (loose | tight)      loose
longlabel      (wrap | nowrap)    wrap >
<!ELEMENT labheads - - (labh, labhtext)
- (memo | location | idx) >
<!ELEMENT labh      - - (parttext) >
<!ELEMENT labhtext  - - (parttext) >
<!ELEMENT lablistitem - - (label, text) >
<!ELEMENT label     - - (parttext) >
<!ELEMENT list      - - (head?, item+) >
<!ATTLIST list      type          (order |
                                   bullet |
                                   plain |
                                   check )      bullet
ordertype         (ualpha |
                   lalpha |
                   arabic |
                   uroman |
                   lroman )      arabic
spacing           (tight |
                   loose )      tight
continue          (continue)     #IMPLIED >
<!ELEMENT item      - - (text) >
<!ATTLIST item      id            ID            #IMPLIED >
<!ELEMENT ex        - - (head?, (exampleseg, annotation?)+)
- (ex |
   vex |
   note |
   caution |
   warning |
   syntax |
   footnote) >
<!ATTLIST ex        notes         (side | stack)      side
lines              (number |
                   nonnumber )      nonnumber
textsize           (normal |
                   smaller |
                   smallest )      normal >
<!ELEMENT exampleseg - - (parttext) +(lineno) >
<!ELEMENT annotation - - (parttext) +(newline) >
<!ELEMENT lineno    - O EMPTY >
<!ATTLIST lineno    id            ID            #IMPLIED >
<!ELEMENT vex       - - CDATA >

```

```

<!ATTLIST vex          lines      (number |
                                nonnumber) nonnumber
                                textsize (normal |
                                smaller |
                                smallest) normal >

<!ELEMENT otherhead   - - (head, text?) >
<!ELEMENT procedure   - - (chaphead, text?)
                        - (procedure) >
<!ELEMENT chaphead    - - (head, abbrev?)
                        - (memo | location | idx | footnote) >
<!ELEMENT abbrev      - - (parttext) - (footnote) >
<!ELEMENT syntax      - - (head?, synel) >
<!ELEMENT synel       - - ((#PCDATA | esc | var |
                        optblock | reqblock )+) >

<!ELEMENT (optblock |
            reqblock) - - (synel+) >
<!ELEMENT figure      - - (caption?)
                        - (figure | graphic) >
<!ATTLIST figure      number      NUMBER          #IMPLIED
                        tonumber   (number |
                        nonnumber) number

                        id          ID              #IMPLIED
                        entity      ENTITY         #REQUIRED
                        figpos      (left |
                        center |
                        right)      #IMPLIED
                        cappos      (capleft |
                        capcenter |
                        capright)   #IMPLIED
                        ghyperlink  CDATA          #IMPLIED
                        glinktype   (jump |
                        jumpnewview |
                        definition |
                        execute |
                        appdefined |
                        man)         jump
                        gdescription CDATA          #IMPLIED >
<!ELEMENT caption     - - (parttext, abbrev?)
                        - (memo | location | idx) >
<!ELEMENT image       - - (head?, parttext) - (footnote) >
<!ELEMENT abstract    - - (head?, text?, frontsub*) >
<!ELEMENT frontsub    - - (head?, text) >
<!ELEMENT otherfront  - - (head?, text?, frontsub*) >
<!ATTLIST otherfront  id          ID              #IMPLIED >
<!ELEMENT hometopic  - - (chaphead, text?) >
<!ELEMENT chapter     - - (chaphead, text?, (s1*, rsect*)) >
<!ATTLIST (chapter |
            s1 |
            s2 |
            s3 |
            s4 |
            s5 |
            s6 |
            s7 |
            s8 |
            s9)        id          ID              #IMPLIED >
<!ELEMENT s1         - - (chaphead, text?, s2*, rsect*) >
<!ELEMENT s2         - - (chaphead, text?, s3*, rsect*) >
<!ELEMENT s3         - - (chaphead, text?, s4*, rsect*) >
<!ELEMENT s4         - - (chaphead, text?, s5*, rsect*) >

```

```

<!ELEMENT s5          - - (chaphead, text?, s6*, rsect*) >
<!ELEMENT s6          - - (chaphead, text?, s7*, rsect*) >
<!ELEMENT s7          - - (chaphead, text?, s8*, rsect*) >
<!ELEMENT s8          - - (chaphead, text?, s9*, rsect*) >
<!ELEMENT s9          - - (chaphead, text?) >
<!ELEMENT rsect       - - (chaphead, text?, rsub*) >
<!ATTLIST rsect       id          ID          #IMPLIED >
<!ELEMENT rsub        - - (chaphead, text?) >
<!ELEMENT message     - - (chaphead?, text?, (msg+ | msgsub+)) >
<!ELEMENT msg         - - (msgnum?, msgtext, explain?) +(newline) >
<!ELEMENT msgnum      - - ((#PCDATA | esc)+) >
<!ELEMENT msgtext     - - (partext) >
<!ELEMENT explain     - - (text) >
<!ELEMENT msgsub      - - (chaphead, text?, msg+) >
<!ELEMENT glossary    - - (text?, glossent+) >
<!ELEMENT glossent    - - (dterm, definition) >
<!ELEMENT dterm       - - (partext) -(term) >
<!ELEMENT definition  - - (text) >
<!ELEMENT idx         - - (indexprimary, indexsub?)
                        -(term | footnote | location | idx) >
<!ELEMENT indexprimary - - (partext, sort?) >
<!ELEMENT indexsub    - - (partext, sort?) >
<!ELEMENT sort        - - ((#PCDATA | esc)+) >
<!ELEMENT memo        - - CDATA >
<!ENTITY MINUS        SDATA  ``-''>
<!ENTITY PM           SDATA  `[plusmn]''> <!-- ISOnum -->
<!ENTITY DIV          SDATA  `[divide]''> <!-- ISOnum -->
<!ENTITY TIMES        SDATA  `[times ]''> <!-- ISOnum -->
<!ENTITY LEQ          SDATA  `[le ]''> <!-- ISotech -->
<!ENTITY GEQ          SDATA  `[ge ]''> <!-- ISotech -->
<!ENTITY NEQ          SDATA  `[ne ]''> <!-- ISotech -->
<!ENTITY COPY         SDATA  `[copy ]''> <!-- ISOnum -->
<!ENTITY REG          SDATA  `[reg ]''> <!-- ISOnum -->
<!ENTITY TM           SDATA  `[trade ]''> <!-- ISOnum -->
<!ENTITY ELLIPSIS     SDATA  `[hellip]''> <!-- ISOpub -->
<!ENTITY VELLIPSIS    SDATA  `[vellip]''> <!-- ISOpub -->
<!ENTITY PELLIPSIS    SDATA  `''...''>
<!-- ellipsis followed by a period -->
<!ENTITY A.M.         SDATA  ``a.m.'''>
<!ENTITY P.M.         SDATA  ``p.m.'''>
<!ENTITY MINUTES      SDATA  `[prime ]''> <!-- ISotech -->
<!ENTITY SECONDS      SDATA  `[Prime ]''> <!-- ISotech -->
<!ENTITY DEG          SDATA  `[deg ]''> <!-- ISOnum -->
<!ENTITY SQUOTE       SDATA  `''''>
<!ENTITY DQUOTE       SDATA  `''''>
<!ENTITY ENDASH       SDATA  ``-''>
<!ENTITY EMDASH       SDATA  `[mdash ]''> <!-- ISOpub -->
<!ENTITY VBLANK       SDATA  `'_''>
<!ENTITY CENTS        SDATA  `[cent ]''> <!-- ISOnum -->
<!ENTITY STERLING     SDATA  `[pound ]''> <!-- ISOnum -->
<!ENTITY SPACE        SDATA  `'' ''>
<!ENTITY SIGSPACE     SDATA  `''& ''>
<!ENTITY SIGDASH      SDATA  `''&-''>
<!ENTITY MICRO        SDATA  `[micro ]''> <!-- ISOnum -->
<!ENTITY OHM           SDATA  `[ohm ]''> <!-- ISOnum -->
<!ENTITY UP           SDATA  `[uarr ]''> <!-- ISOnum -->
<!ENTITY DOWN         SDATA  `[darr ]''> <!-- ISOnum -->
<!ENTITY LEFT         SDATA  `[larr ]''> <!-- ISOnum -->
<!ENTITY RIGHT        SDATA  `[rarr ]''> <!-- ISOnum -->
<!ENTITY HOME         SDATA  `''home key''>

```

```
<!ENTITY BACK      SDATA ``<--''>
<!ENTITY HALFSIZE SDATA `` ``>

<!ENTITY % user-defined-entities SYSTEM ``helptag.ent''>
%user-defined-entities;
] >
```


用語集

Standard Generalized Markup Language (SGML)

情報交換の方法を確立することを目的とする国際規格 (ISO 8879: 1986) です。SGML は、情報の構造の指定をマークアップにより行い、内容とは別に表示や書式を規定します。ヘルプタグ・マークアップ言語は、この SGML 規格に準拠しています。

Tagged Image File Format (TIFF)

標準グラフィック・ファイル書式です。ヘルプ・システムのダイアログ・ボックスは TIFF 5.0 イメージをサポートしています。TIFF イメージはファイル名拡張子 `.tif` で識別できます。

X ウィンドウ・ダンプ

X Window System ディスプレイから取り込んだイメージです。ウィンドウ・イメージの取り込みには `xwd` ユーティリティを使用します。X Window System のダンプ・イメージ・ファイルはファイル名拡張子 `.xwd` で識別できます。

X ピクスマップ

マルチカラー・イメージです。ピクスマップ・イメージ・ファイルはファイル名拡張子 `.pm` で識別できます。

X ビットマップ

フォアグラウンド・カラーとバックグラウンド・カラーから成る 2 色イメージです。ビットマップ・イメージ・ファイルはファイル名拡張子 `.bm` で識別できます。

アプリケーション・ヘルプ

特定のアプリケーション (ソフトウェア) 用のオンライン・ヘルプです。

アプリケーション定義リンク

あるアプリケーションの動作を呼び出すために特別に設計されたハイパーリンクです。動作を呼び出すには、ヘルプはアプリケー

ションで作成したダイアログに表示されなければなりません。(アプリケーション定義のハイパーリンクは、ヘルプ・ビューアでは無視されます。)

一般ヘルプ・ダイアログ・ボックス	ヘルプ情報が表示されるウィンドウです。ここにはメニュー・バー、トピック・ツリー(トピックのリストを表示します)、およびヘルプ・トピック表示領域があります。「簡易ヘルプ・ダイアログ・ボックス」を参照してください。
インライン・グラフィック	1行のテキスト内に表示される小さなグラフィック(イラスト)です。
ウィジェット	グラフィカル・ユーザ・インタフェースの基本的な構成ブロックです。OSF/Motif ウィジェット・セットは、アプリケーション・ユーザ・インタフェースの構築に適したすべての種類のウィジェットを提供します。
エンティティ	名前の付いているテキスト文字列またはファイルです。ほとんどのエンティティは(<!entity> 要素を使用して)設計者が名付けますが、あらかじめ定義されているものもあります。「エンティティの宣言」と「エンティティの参照」を参照してください。
エンティティの参照	前に &(アンド記号)、後ろに ;(セミコロン) が付いているエンティティ名を使用して、エンティティ名を記述した場所にそのエンティティを挿入するよう、ヘルプタグに指示することです。「エンティティ」と「エンティティの宣言」を参照してください。
エンティティの宣言	エンティティ名とボリュームを確立するようにマークすることです。「エンティティ」と「エンティティの参照」を参照してください。
エントリ・ポイント	ヘルプを要求すると直接表示されるヘルプ・ボリュームにあるポイントです。つまり、ユーザが「入力」するトピックの場所またはオンライン・ヘルプを読み始める場所です。ID を持つあらゆるトピック、またはトピック内の場所がエントリ・ポイントになり得ます。
簡易ヘルプ・ダイアログ・ボックス	ヘルプ・トピック表示領域と1つ以上のプッシュ・ボタンがある、簡素化したヘルプ・ダイアログ・ボックスです。その他の機能を提供する「一般ヘルプ・ダイアログ・ボックス」を参照してください。

簡易マークアップ	<p>要素をマークする方法を省略した形式です。簡易マークアップでは、開始タグと終了タグを特別な 2 文字の文字列で置き換えます。たとえば、<computer> という要素の簡易マークアップは、次のように開始の単一引用符を 2 つ付け、終わりに閉じる単一引用符を 2 つ付けます。</p> <pre>''text''</pre>
強調表示	<p>テキスト上で注意を促す要素です (通常はイタリック文字です)。日本語では、イタリック書体のフォントが提供されていない場合、それに代わる書体、たとえばボールドで表示されます。</p>
クローズ・コールバック	<p>ヘルプ・ダイアログ・ボックスを閉じたときに呼び出すアプリケーション機能です。</p>
警告	<p>発生する可能性がある損害や回復不可能なデータ損失をユーザに警告する情報です。「注意」と「注」を参照してください。</p>
検索ボリューム選択ダイアログ・ボックス	<p>ユーザのシステムで使用可能なヘルプ・ボリュームをリストしたダイアログ・ボックスです。ユーザが索引検索ダイアログ・ボックスで [選択] を選択すると、ユーザが選択できるヘルプ・ボリュームがこのダイアログ・ボックスにリストされます。ボリューム名は 1 つ以上選択でき、対応する索引情報が索引検索ダイアログ・ボックスに報告されます。</p>
コンテキスト・ヘルプ	<p>ユーザがアプリケーションで行う作業に関するオンライン情報です。ヘルプ・トピックの選択はユーザのコンテキストに基づいて行われるため、[F1] キーを押すと「コンテキスト・ヘルプ」が表示される場合もあります。</p>
索引	<p>ヘルプ・ボリューム中に現れる重要な語句のリストです。索引には語句がアルファベット順に並んでいて、知りたい項目についてのヘルプを検索できます。ユーザが (一般ヘルプ・ダイアログ・ボックスの) [索引] ボタンを選択すると、ヘルプ・システムは索引を表示します。「索引検索ダイアログ・ボックス」を参照してください。</p>
索引検索ダイアログ・ボックス	<p>ヘルプ・ボリュームの索引エントリのリストを表示するダイアログ・ボックスです。現在のボリューム、選択したボリューム、すべてのヘルプ・ボリュームの索引を表示できます。ユーザが索引で特</p>

定の語句を検索すると、その検索文字列を含む項目がリストされます。

実行時のヘルプ・ファイル	<code>dthelptag</code> コマンドで生成されるファイルです。これらのファイルはヘルプ・システムを使用するユーザに配布されます。
実行別名	実行リンクが実行するコマンド文字列またはスクリプトへ名前を割り当てるリソースです。
実行ポリシー	実行リンクの動作を制御するよう設定できるリソースを提供するヘルプ・システムです。システム管理者またはユーザが、指定されたアプリケーションに対して適切なセキュリティ・レベルを確立できるようにします。
実行リンク	シェル・コマンドまたはシェル・スクリプトを実行するハイパーリンクです。
自動ヘルプ	特定の条件またはエラーが生じると、システムがヘルプを表示します。「システム起動ヘルプ」と呼ばれることもあります。たとえば、エラー・ダイアログは「自動ヘルプ」の形式です。「半自動ヘルプ」と「手動ヘルプ」を参照してください。
手動ヘルプ	どんなヘルプが必要で、どうすればそれが得られるかを、ユーザがわかっているなければ使用できないオンライン・ヘルプです。たとえば [ヘルプ] メニューのコマンドの大半は、いつ何を表示するかをユーザが選択するので、「手動」ヘルプです。「自動ヘルプ」と「半自動ヘルプ」を参照してください。
新規ウィンドウ表示ハイパーリンク	選択されると新しいダイアログ・ボックスに情報を表示するハイパーリンクです。新規ウィンドウ表示リンクは、他のボリュームへのリンクを目的としています。ユーザは、新規ウィンドウが表示されることによって「新しいコンテキスト」だとわかります。
スタンドアロン・ヘルプ	アプリケーション・ソフトウェアとは無関係に使用されるヘルプ情報です。たとえば、コンピュータ・プログラミングの基本を説明するオンライン・ヘルプは、特定のアプリケーションに関連付けられてはいません。スタンドアロン・ヘルプ・ボリュームは、 <code>dthelpview</code> コマンドで表示できます。
図	ヘルプ情報で表示されるグラフィックまたはイラストです。

正規マークアップ	ヘルプタグ 1.3 文書型定義 (DTD) で指定されたタグ・セットおよびその使用規則です。DTD で決められた規則に従うことにより、設計者は Standard Generalized Markup Language (SGML) 準拠ヘルプ・ソース・ファイルを生成できます。
ダイアログ・キャッシュ	作成されても使用されていないヘルプ・ダイアログのリストです。アプリケーションは、新しいヘルプ・ダイアログを必要とする場合、まずダイアログ・キャッシュに未使用のダイアログがないかどうか検索します。見つければそれを使用し、見つからない場合は新しいダイアログを作成します。
タグ	要素の始めと終わりをマークするテキスト文字列です。開始タグは、< (左三角括弧)、特殊文字列 (文字のみ)、オプションのパラメータと値、> (右三角括弧) から成ります。 終了タグは、< (左三角括弧)、\ (バックスラッシュ)、開始タグと同じ特殊文字列、> (右三角括弧) から成ります。正規マークアップでは、終了タグ構文に / (スラッシュ) を使用します。
注	ユーザに注意してほしい重要な情報であることを示します。情報の内容が特に重大な場合は、代わりに注意や警告が使われます。「注意」と「警告」を参照してください。
注意	データが失われる恐れがあることをユーザに知らせる警告です。「注」と「警告」を参照してください。
登録	ヘルプ・ボリュームが、ブラウザや他のボリュームへのリンクのためにアクセス可能であると宣言するプロセスです。
トピック	特定のサブジェクトに関する情報です。通常、これは約 1 画面分の情報です。オンライン・ヘルプのトピックは、ハイパーリンクによって互いにリンクされています。
トピック階層	ホーム・トピックが (ハイパーリンクによって) だんだん詳しいトピックに分岐していく、ヘルプ・ボリュームの分岐構造です。「ホーム・トピック」を参照してください。
トピック・ツリー	一般ヘルプ・ダイアログ・ボックスで、ヘルプ情報を表示するために選択できるトピックのリストです。

パーサ	ヘルプタグ・ソフトウェアの一部で、(設計者が作成した) ソース・ファイルを読み取り、ヘルプ・システム・ダイアログが読み取れる実行時のヘルプ・ファイルに変換します。設計者がマークアップを不正確に(または不完全に)使用した場合、パーサは障害を検出して「パーサ・エラー」の発生を通知します。
ハイパーリンク	テキスト(語句)のセグメント、あるいはそれに対応する動作を示すグラフィック・イメージです。最も典型的なハイパーリンクは、関連するトピックにつながる「ジャンプ」リンクです。ユーザがジャンプ・リンクを選択すると、関連するトピックが表示されます。また、他の動作、つまりシステム・コマンドの実行や特定のアプリケーションの呼び出しなどを起動することもできます。
ハイパーリンク・コールバック	ユーザがハイパーリンクを選択したときに呼び出されるアプリケーション機能です。ヘルプ・ダイアログで自動的に処理されないタイプのハイパーリンクを処理します。
半自動ヘルプ	ユーザがヘルプを要求すると、システムが現在の状況に応じてどのヘルプ情報を表示するかを決定する形式のオンライン・ヘルプです。「コンテキスト・ヘルプ」([F1] キーを押す)は、半自動ヘルプの一例です。「自動ヘルプ」と「手動ヘルプ」を参照してください。
ヒストリ・ダイアログ・ボックス	ユーザが使用したトピックを順にリスト表示するダイアログ・ボックスです。ヒストリの順序は、ユーザが最初に使用したトピックに戻りやすくするために、逆の順序で表示することもできます。
ブラウザ・ボリューム	デスクトップは、システムにインストールされたヘルプをリストするスペシャル・ブラウザ・ボリュームを表示することにより、ヘルプ・ビュープログラムを「ヘルプ・ブラウザ」として使用します。dthelpgen と呼ばれるユーティリティは、このボリュームをユーザのホーム・ディレクトリに作成します。
文書型定義 DTD (Document Type Definition)	構造化(または階層)情報を作成するのに使用する一連の要素の記述です。DTD は各要素の構文と、ドキュメントでその要素がどのようにどこで使用されるかを指定します。
ヘルプ・キー	通常は [F1] ファンクション・キーですが、現在のコンテキストのヘルプを要求するためのキーです。[F1] キーに代わる [Help] キー

が用意されているキーボードもあります。OSF/Motif アプリケーションでは、ウィジェットにヘルプ・コールバックを追加するとヘルプ・キーが使用可能になります。

ヘルプ・コールバック	ユーザが [F1] キーを押したときに呼び出されるアプリケーション機能です。
ヘルプの使い方	ヘルプ・ダイアログ・ボックスの使用方法を説明したヘルプ情報です。ヘルプ・ウィンドウを使用しているときは [F1] キー、または一般のヘルプ・ダイアログ・ボックスの [ヘルプの使い方] を選択することにより、この情報を取得できます。
ヘルプ・ファミリー	参照するアプリケーションが関連している一連のヘルプ・ボリュームです。
ヘルプ・ボリューム	サブジェクトに関する完全情報です。また、マークアップ・テキストを含む一連のソース・ファイルまたはヘルプタグを使用することにより生成される実行時のヘルプ・ファイルのいずれかを指すこともあります。
ホーム・トピック	ヘルプ・ボリュームの階層の最高位にあるトピックです。ユーザがヘルプ・ボリュームをブラウズしたいことを示すときに表示されるトピックです。ヘルプタグはホーム・トピックの組み込み ID (<code>_hometopic</code>) を提供します。
他のボリュームへのハイパーリンク	他のヘルプ・ボリュームのトピックにジャンプするハイパーリンクです。これは <code><link></code> 要素で入力できます。ここで <code>hyperlink</code> パラメータはボリューム名およびボリューム ID (スペースで区切ります) を指定します。例を以下に示します。 <code><link hyperlink="volume">text<\link></code>
マニュアル・ページへのリンク	起動されると「マニュアル・ページ」を表示するハイパーリンクです。マニュアル・ページとは、システム・コマンドについてのオンラインの短い説明です。マニュアル・ページ上の情報は、ヘルプタグ・システムでは提供されません。
要素	本のタイトル、段落、リスト、またはトピックなど、情報の論理部分です。一部の要素はコンテキストが想定しますが、通常、要素のエクステントはタグによってマークされます。

略式マークアップ

要素をマークする方法を省略した形式です。略式マークアップでは、終了タグと、開始タグの最後の文字を一本の縦線でマークします。たとえば、<book> という要素の略式マークアップは次のとおりです。

```
<book|text|
```

例表示

コンピュータのフォントで表示されるテキストの本文で、行分割が意図的に行われています。一般には、コンピュータ・ファイルの一部です。<ex> または <vex> 要素を使用して入力されます。

索引

記号

- % (パーセント記号)
 - 簡易マークアップ 90
- & (アンド記号)
 - エンティティの参照 72, 147
 - テキスト文字としての使用 54
-) 72, 147
- ;(セミコロン) 72, 147
- \ (バックスラッシュ)
 - テキスト文字としての使用 54
 - 終了タグ形式 52
 - 複数行ヘッダでの使用 157

数字

- 2つの左アポストロフィ (") と 2つの右アポストロフィ (") 143

A

- <abbrev> 137
- <abstract> 139
- _abstract ID 70
- <annotation> 139
- APIに関する関数のリスト 237
- AppDefined パラメータ 98

B

- <book> 140
- bullet パラメータ 172

C

- <caution> 141
- CDE ヘルプ・システムについて 23
- <chapter> 50, 142
- clean パラメータ 212
- CloseHelpCB() 236
- <computer> 143
- continue パラメータ 172
- © 177
- _copyright ID 70, 265

D

- DtCreateHelpQuickDialog() 235
- DtHelp.cat 288
- DtHelpDialogCallbackStruct 255
- DtHelpExecAlias 101
- dthelpgen
 - コマンドライン・オプション 216
- DtHelpReturnSelectedWidgetId() 249, 250
- dthelptag コマンド
 - コマンド・オプション 212
 - コマンド行からの実行 119

dthelpview コマンド
 コマンド行からの実行 121
 コマンド形式 215
dthelpview のアプリケーションのデフォルト
 のリソース・ファイル 261
DtHELP_TYPE_DYNAMIC_STRING 243
DtHELP_TYPE_FILE 243
DtHELP_TYPE_MAN_PAGE 244
DtHELP_TYPE_STRING 243
DtNhelpType 242
DtNhelpVolume 242
DtNlocationId 242
DtNmanPage 244
DTTAGOPT 環境変数 213

E

element tag 72

F

[F1] (ヘルプ・キー) 245
FILE パラメータ 73, 147
formal パラメータ 213

G

gencat コマンド 288
gentity パラメータ 106, 182
ghyperlink パラメータ 103, 182
glinktype パラメータ 103, 182
_glossary ID 70
gloss パラメータ 190
gposition パラメータ 182

H

Help4Help
 Help4Help ボリュームの場所 264
 helpOnHelpVolume リソース 260
 アプリケーションからのアクセス 261
 ソース・ファイルのコピー 265
 必須のエントリ・ポイント 264
Help4Help ソース・ファイルのコピー 265
helpchar.ent ファイル 201
helplang.ent ファイル 289
helpOnHelpVolume リソースの設定 260, 261
HelpRequestCB() 262, 262

helptag.dtd ファイル 220
helptag.opt ファイル
 memo オプション 110
 検索オプション 108
 例 119
<helpvolume>
 要素タグ 158
<hometopic>
 要素タグ 159
_hometopic ID 70
.htg ファイル 55
HyperlinkCB() 235
hyperlink パラメータ 97, 98

I

ID
 <xref> での使用 197
 図 103
 定義済みの名前 70
 トピックの要素への追加 71
 トピックへの追加 70
 命名規則 69
 リストの項目 163
ID パラメータ 152
 <chapter> 142
 <image> 161
 <item> 163
 <location> 174
 <otherfront> 180
 <p> 182
 <rsect> 186
 <s1>...<s9> 187
 要素 70
indent パラメータ
 イメージ要素 161
 パラグラフ要素 182

L

LANG 環境変数 288
<location> 71
loose パラメータ 80, 165, 172

M

Man パラメータ 97

MoreHelpCB() 235
Motif 25

N

nogloss パラメータ 112, 190
nonnumber パラメータ 103, 152
number パラメータ 150, 152

O

order パラメータ 80, 172
OSF/Motif 25

P

plain パラメータ 80, 172
ProcessOnItemHelp() 250

S

<s1>...<s9> 50, 65, 78, 187
SDL (Semantic Delivery Language) 42, 115
Semantic Delivery Language (SDL) 42
SGML (Standard Generalized Markup Language) 219
SGML 準拠 219
shortnames パラメータ 212
smaller パラメータ 150, 195
smallest パラメータ 150, 195

T

Tagged Image File Format (TIFF) 103
TIFF (Tagged Image File Format) 103
tight パラメータ 80, 165, 172
_title ID 70

V

verbose パラメータ 212
volume
 abstract 139

X

<xref>
 相互参照の作成 92
XtAddCallback() 245, 255
xwd グラフィック形式 103

X ウィンドウ・ダンプ 103

あ

アイテムヘルプ
 サポートを追加する 250
 呼び出し 250
アイテムヘルプ・モード 249
アイテムヘルプ・モードのサポート 249
新しい段落 182
宛先
 相互参照 174
 ハイパーリンク 174
アプリケーション
 登録 268
アプリケーション・エントリ・ポイント
 確認 129
アプリケーション・エントリ・ポイントの確
 認 129
アプリケーション構成ボタン 257
アプリケーション構成ボタンの使い方 257
アプリケーション定義
 ハイパーリンク 91
 リンクの作成 98
アプリケーションのヘルプ 37
アプリケーション・パッケージ 268
アプリケーション・プログラマ
 協力 39
 責任 43
アプリケーション・プログラマとの協力 39
アプリケーション・プログラム・インタ
 フェースの関数のリス
 ト 237
アンド記号 (&)
 エンティティの参照 147
 テキスト文字としての使用 54

い

イタリック 89, 90
イタリックフォント 194
一般的なマークアップのガイドライン 52
一般ヘルプ・ダイアログ・ボックス
 機能 231, 232
 作成 233
 ダイアログ・ボタン 232

イベント
アプリケーションがサポートしなければ
ならない処理 253
ハイパーリンク 255

印刷

[印刷] ダイアログ・ボックスのヘルプ
ID 265
ダイアログ・ボックス 34
ヘルプ情報 34
ヘルプ・トピック 128

印刷上の記号 201

インストール・パッケージ 268

引用符

エンティティ名用の二重引用符 202
開始と終了のタグ 185
簡易マークアップでの使用 90, 143
一組の二重引用符 185

インライン・グラフィックの表示 105

インライン要素

入力 88

インライン要素の入力 88

う

ウィジェット・クラス 232

ウィジェット・リソース 232

ウィンドウ・ダンプ 103

え

エスケープ・テキスト 149

エディタ

構造化エディタ 40

エラー

修正 120

エラーの修正 120

エンティティ 202

LanguageElementDefaultCharset 287,
289

LanguageElementDefaultLocale 287

エンティティの使い方 72

宣言 50, 147

テキスト・エンティティの作成 72

特殊文字 201

ファイル・エンティティの作成 73

複数のソース・ファイル 55

例 148

エンティティの参照 108

エンティティ・パラメータ 103, 152, 155

エンティティ名

エヌ・ダッシュ 201

エム・ダッシュ 201

エントリ・ポイントの確認 129

お

応答

ハイパーリンク・イベント 255

ヘルプの要求 239

オプション

メモ 110

オンライン・プレゼンテーション形式 115

オンライン・ヘルプ

目的 37

オンライン・ヘルプの目的 37

か

改行

タイトル内での改行 78

復帰改行 178

ヘッダ 157

改行に変更なし 150, 161, 195

改行の保存 161

外国語でのヘルプ・ボリュームの作成 279

外国ユーザ向けの記述 296

開始タグ 52

解釈されないテキスト 149

階層

作成 62

トピックの追加 65

非階層トピックの追加 68

要素の構造 78

外部ファイルの参照 72

概要

ヘルプ・グラフィカル・ユーザ・インタ
フェース 27

ヘルプ・ボリューム 54

確認

アプリケーション・エントリ・ポイン
ト 129

簡易ヘルプ・ダイアログ・ボックス

作成 235

ボタン 234

簡易マークアップ 40, 53

簡易マークアップで使用する % (パーセント記号) 90, 194
簡易マークアップで使用する感嘆符 (!) 88, 146
簡易マークアップで使用する縦線 88
簡易マークアップで使用する + (プラス) 96, 112, 190

環境変数

LANG 288
システム・ヘルプ検索パス 272
パーサ・オプションの DTTAGOPT 213
ユーザ・ヘルプ検索パス 272

関数

DtHelpReturnSelectedWidgetId() 249, 250
HelpRequestCB() 262, 262
ProcessOnItemHelp() 250
XtAddCallback() 245, 255

簡単な説明

ヘルプ・ボリューム 139

感嘆符 146

関連するボリューム

ボリュームのファミリーとしてグループ化 35

き

キーワード索引をソートする順序 160

危険性の警告 196

記号

+ (プラス記号) 190
記号のリスト 201
挿入 108

記述

外国のユーザ 296
トピック 78
「ヘルプの使い方」ボリューム 264

行

1 行以上に渡る定義の長さ 146
新しい行の開始 157
改行 157
行の続行 157

強調表示

イタリック・フォントの使用 146
注の使用 179
ボールドの使用 89

行頭の空白 172

行頭文字

エンティティ名 202
空白 172
点 172
リストでの使用 172
行頭文字が付けられたリスト 172
ギリシャ文字 204

く

組み込み ID 70

クラス

ダイアログ・ウィジェット 232

グラフィック

インラインの表示 105

形式 103

挿入 152

テキスト内の要素 155

ハイパーリンクとして扱われるグラフィック 169

表示 103

まわりをテキストで囲むには 106

グラフィックのまわりをテキストで囲む 106

繰り返し出てくる同じテキスト 72

け

警告

危険性 196

警告文の追加 86

言語

外国語 279

フォーマット・テーブル 289

マルチバイト 280

現在の時間 201

現在の日付と時間 201

検索メニュー 111

こ

構造化エディタ 40

構造体

DtHelpDialogCallbackStruct 255

コールバック

クローズ・コールバックの追加 236

クローズ・コールバックの例 256

ハイパーリンクの指定 255

ヘルプ・コールバックの追加 245

コマンド

- dthelpgen 216
- dthelpview 215
- gencat 288
- ヘルプ・コマンドの要約 211
- コマンド変数の表示 194
- コメントとして使用する <!--...--> 110
- コメントの挿入 109, 110
- コンピュータ
 - 「入力 / 出力」の表示 150
- コンピュータのプロンプトに対するユーザの応答 193
- コンピュータ・リテラルの表示 89

さ

索引

- エントリのマーク 111
- [検索ボリューム選択] ダイアログ・ボックスのヘルプ ID 265
- 索引検索 ID の取得 265
- [索引検索] ダイアログ・ボックス 32
- [索引検索] ダイアログ・ボックスのヘルプ ID 265
- 作成 111
- ソートする順序 160

作成

- アプリケーション定義リンク 98
- 一般ヘルプ・ダイアログ・ボックス 233
- 簡易ヘルプ・ダイアログ・ボックス 235
- 言語フォーマット・テーブル 294
- 索引 111
- 実行時のヘルプ・ファイル 42, 117
- 図 103
- 定義リンク 96
- テキスト・エンティティ 72
- トピック階層 62
- トピック内の構造の作成 78
- ハイパーリンク 91
- ファイル・エンティティ 73
- ヘルプ・ダイアログ・ボックス 235
- ヘルプ・ファミリー 125
- ホーム・トピック 64
- マニュアル・ページへのリンク 97
- メタ情報トピック 66
- メッセージ・カタログ 296
- 用語集 112
- 作成者のメモの挿入 109, 110
- サブトピック 50

- 作成 65
- サブヘッダ
 - トピック内のサブヘッダ 180
 - トピックにサブヘッダを付ける 83
- さまざまな表示のテスト 129
- <(三角括弧)
 - テキスト文字としての使用 54
- 参照
 - エンティティ 147
 - エンティティの参照 108
 - サブセクション 186
 - セクション 186
- 参照セクションのサブセクション 186

し

時間

- 現在 201
- 実行時のファイル
 - 作成 42, 117
- 実行時のヘルプ・ファイル 42, 117
- 実行ハイパーリンク 91
- 実行別名
 - 作成 100
 - ハイパーリンクでの使用 101
- 実行リンク
 - 制御のポリシー 99
 - デフォルト動作 99
- 実行リンクで使用する別名 100
- ジャンプ型のハイパーリンク 91
- 終了タグ 52
- 順番が付けられたリスト 172

章

- トピック階層 50

商標

- エンティティ名 201

情報

- メタ 51

す

図

- ID 103
- エンティティ 147
- 作成 103
- 挿入 152
- タイトル 103

- 番号 103
- 垂直スペース 205
- 水平スペース 202
- 数学記号 205
- スタンドアロンのヘルプ・ボリュームの例 57
- 作成 57
- スタンドアロン・ヘルプ 25, 38
- 図の番号 103
- スペース
 - 垂直空白 205
 - 水平空白 202
 - 有意スペース 202
- スペースの保存 161
- スペースをそのままの状態 で 150, 195
- せ
- 正規マークアップ 40
 - SGML 準拠 219
 - エンティティ宣言 227
 - 処理 228
 - 注意点 224
 - 文書型定義 219
- 製品の準備チェックリスト 273
- 製品のファミリー
 - 検索 272
 - 作成 125
- 責任
 - 設計者 37
 - プログラム 43
- セクション
 - 参照 186
 - 参照セクションのサブセクション 186
 - トピック 187
 - トピック内のヘッダ 187
 - プロシージャの記述 184
 - ヘッダ 187
- 設計者
 - アプリケーション・プログラマとの協力 39
 - 責任 37
 - ワークフロー 39
- 設計者が行う作業 37
- セマンティック・デリバリ言語 (Semantic Delivery Language) 115
- セミコロン 72, 147
- 先頭項目 177

- 前付
 - 定義済みカテゴリのうちの一つで一致しない 180

そ

相互参照

- ID の値 197
- xref 要素の使用 197
- 位置 ID の使用 174
- リストの項目 163
- リンク要素の使用 169
- 挿入 72
 - コメント 110
 - 作成者のメモ 110
 - 他のファイルの内容 72
 - 特殊文字 108
- ソース・ファイル
 - 複数のソース・ファイル 55
- ソートする順序 160
- 続行
 - 行の続行 157
- 空 (テキストなし) 202

た

ダイアログ

- 終了の検出 256
- 処理イベント 253
- ダイアログ・ボックス
 - 一般ヘルプ 231, 232
 - 一般ヘルプの作成 233
 - 簡易ヘルプ 231
 - 簡易ヘルプの作成 235
 - 作成および管理 231

タイトル

- Note 156
- 新しく開始する 142
- 改行 78
- 省略 137
- 図 103
- セクション 156
- タイトル内での改行 78
- トピック 78, 142, 187
- ヘルプ・ボリューム 192
- 本 140
- 本のタイトルを入力する 89

- リスト 156
- タイトル内での行の開始 78
- タグ 52
- ダッシュ文字
 - エヌ・ダッシュ 201
 - エム・ダッシュ 201
- ダンプ
 - X ウィンドウ 103
- 段落
 - インデント 182
 - 開始 79
- 段落の開始 79, 182

ち

- チェックリスト
 - 国際化対応ヘルプ 296
 - 製品のインテグレータの場合 274
 - 製品の準備 273
 - 設計者の場合 273
 - プログラマの場合 274
- 逐語例 195
- 注意文の追加 86
- 注の追加 86
- 著作権
 - メタ情報での使用 177
 - エンティティ名 201
- 著作権
 - 告知 144
 - 注意事項 66
 - 定義済みのヘルプ ID 264

て

- 定義
 - 用語 154
 - 用語集に入れる 114
- 定義済み ID
 - _abstract 70
 - _copyright 70
 - _glossary 70
 - _hometopic 70
 - _title 70
- 定義済み位置 ID 70
- 定義済みエンティティ 201
- 定義ハイパーリンク 91
- 定義リンク
 - 作成 96

- 用語 52
- テキスト
 - エンティティの作成 72
 - 解釈されないもの 149
 - 行末をそのままの状態に 161
 - グラフィックのまわりを囲む 106
 - ファイルの表示 243
 - 文字列の表示 242
- テキスト文字列 72
 - 表示 242
- テキストを入れる二重引用符 185
- テスト
 - さまざまな表示のテスト 129
 - ヘルプ 129
- 点の行頭文字 172

と

- 登録
 - ヘルプ・ファミリ 268
 - ヘルプ・ボリューム 268
- ドキュメントのタイトル 192
- 特殊文字エンティティの参照 108
- 特殊文字の挿入 108, 201
- 特殊文字の表示 108
- 特殊文字のリスト 201
- 整えられていないテキスト 161
- トピック 35
 - ID をトピック内の要素に追加する 70
 - <hometopic> 159
 - アクセス 69
 - 新しいトピックの開始 142
 - 階層の作成 62
 - 階層への追加 65
 - 記述 78
 - サブトピック 50
 - サブヘッダを付ける 83
 - タイトル 142, 187
 - トピック内に構造を作成する 78
 - トピック内の構造 78
 - トピック内のサブヘッダ 180
 - トピックの集まりであるボリューム 35
 - 非階層トピックの追加 68
 - 表示 242
 - ホーム・トピック 50
 - ホーム・トピックの作成 64
 - メタ情報トピックへのリンク 98

- 要素の構造 78
- 要素へのIDの追加 71
- トピック・ツリー
 - 一般ヘルプ・ダイアログのトピック・ツリー 30
 - トピックの選択 30
- ドラフトのコメントや質問 176
- トランプ組札 209

な

- 長いタイトルの省略 137
- 何も付いていないリスト 172
- ナビゲーション・ボタン 30

は

- パーサ 116
- パーサ・エラー 120
- ハイパーリンク 91
 - 宛先 174
 - アプリケーション定義リンク 91
 - イベントへの応答 255
 - コールバック 235
 - コールバックの指定 255
 - 作成 91
 - 実行リンク 91
 - ジャンプの型 91
 - 妥当性テスト 129
 - 定義リンク 91
 - 表示形式 29
 - マニュアル・ページへのリンク 91
 - リンク要素の属性 169
- ハイパーリンクの妥当テスト 129

配布

- 製品準備 273

バックスラッシュ (\)

- 終了タグでの使用 52
- テキスト文字としての使用 54
- 複数行のヘッダを作成するためのバックスラッシュ 157

パラメータ

- AppDefined 98
- FILE 73, 147
- Man 97
- alias_name 101
- bullet 80, 172
- <chapter> の ID 142

- clean 212
- continue 172
- default_command 101
- entity 103, 152, 155
 - <figure> の ID 152
- formal 213
- gentity 106, 182
- ghyperlink パラメータ 103, 182
- glinktype パラメータ 103, 182
- gloss パラメータ 190
- gposition パラメータ 182
- hyperlink 97, 98
 - <image> の ID 161
- indent 79, 161, 182
- <item> の ID 163
- <location> の ID 174
- loose 80, 165, 172
- nogloss 112, 190
- nonumber 103, 152
- number 150, 152
- order 80, 172
- <otherfront>のID 180
- plain 80, 172
- <p> の ID 182
- <rsect> の ID 186
- <s1>...<s9> の ID 187
- search 108
- shortnames 212
- smaller 150, 195
- smallest 150, 195
- tight 80, 165, 172
- verbose 212

ひ

- 非階層トピックの追加 68
- [ヒストリ] ダイアログ・ボックスのヒストリ・ヘルプ ID 265
- ピクスマップ 103
- 必須のエントリ・ポイント 264
- ビットマップ 103
- 表示
 - インライン・グラフィック 105
 - グラフィック 103
 - コンピュータの例 150
 - コンピュータ・リテラル 89
 - さまざまな表示のテスト 129

- テキスト・ファイル 243
 - テキスト文字列 242
 - ヘルプ・トピック 242
 - 「ヘルプの使い方」 262
 - ヘルプ・ボリューム 115
 - 変数 90
 - マニュアル・ページ 244
- ふ
- ファイル
 - .htg 55
 - DtHelp.cat 288
 - DtHelp.msg 288
 - helpchar.ent 108
 - helptag.dtd 220
 - helptag.opt 110, 119
 - 外部ファイルを参照 72
 - テキストの表示 243
 - ファイル・エンティティの作成 73
 - ファミリー 35
 - 検索 272
 - ファミリーとしてグループ化された関連するボリューム 35
 - ファミリー・ファイル
 - 作成 125
 - フォーマット・テーブル 289
 - フォント
 - イタリック 89, 90, 194
 - コンピュータ・リテラル 89
 - 実際の表示を決めるスキーマ 290
 - 表示用スキーマ 293
 - ボールド・フォントへの変更 89
 - フォント・スキーマ
 - 実際の表示の決定 290
 - フォント・スキーマの表示 293
 - フォント・リソースの指定 290
 - 複数のソース・ファイル 55
 - 複数の入力ファイル 55
 - 復帰改行 178
 - 強制的な開始 178
 - ブラウザ・ヘルプ・ボリューム
 - 作成 124
 - 表示 127
 - ヘルプの追加 123
 - 文書型定義 (DTD) 219
- へ
- ヘッダ
 - 新しい行の開始 157
 - セクション 187
 - ヘルプ
 - アクセス方法 37
 - データ型 42
 - 動作 42
 - トピックの構成 34
 - ヘルプが統合されているアプリケーションの例 43
 - ヘルプ・キー 245
 - ヘルプ・キー ([F1] キー) を使用可能にする 245
 - ヘルプ・システムの概要 23
 - ヘルプ情報モデル 25
 - ヘルプ・ダイアログ
 - 終了の検出 256
 - 処理イベント 253
 - ヘルプ・ダイアログ・ボックス 231
 - ヘルプ・ダイアログ・ボックスの管理 231
 - ヘルプタグ・ソフトウェア 116
 - ヘルプタグ・マークアップのリファレンス 133
 - ヘルプ・トピック 35
 - ID の追加 69
 - ID の割り当て 69
 - ヘルプの獲得 26
 - 「ヘルプの使い方」
 - アクセス 237
 - アプリケーション・リソース 261
 - 簡易ヘルプ・ダイアログ・ボックス 235, 237
 - 表示 262
 - ボリュームを記述する 264
 - [ヘルプの使い方] コマンド 262
 - [ヘルプの使い方] コマンドの指定 262
 - ヘルプのデータ型 42
 - ヘルプの動作 42
 - ヘルプ・ファイル
 - 検索 272
 - ヘルプ・ファイルの検索 272
 - ヘルプ・ファミリー 35
 - 作成 125
 - ヘルプ・ブラウザ 35
 - ヘルプ・ブラウザ・ボリューム 35

- ヘルプ・ボタンの [ナビゲーション] 30
- ヘルプ・ボリューム 35
 - dthelpview コマンド 121
 - 概要 54
 - コンポーネント 49
 - 作成 57
 - 表示 121
 - マークアップの例 54
- ヘルプ・ボリュームの簡単な説明 139
- ヘルプ・ボリュームのコンポーネント 49
- ヘルプ・ボリュームの処理 115
- ヘルプ・ボリュームの表示 121
- ヘルプ・ボリューム・マークアップの例 54
- [ヘルプ] メニューの提供 248
- ヘルプ要求への応答 239
- ヘルプを獲得する方法 26
- ヘルプを見るユーザの視点 42
- 変数
 - コマンドでのマーク 194
 - 表示 90

ほ

- ホーム・トピック 50
 - 作成 64
 - 定義済みのヘルプ ID 264
 - トピック・ツリー 30
 - メニュー・コマンド 31
- ボールド 89
- 母国語のサポート
 - オンラインヘルプの作成 279
 - 記述者と翻訳者のためのチェックリスト 296
- ボリューム 35
 - <helpvolume> 158
 - 概要 54
 - 簡単な説明 139
 - 検索 272
 - コンポーネント 49
 - スタンドアロン・ヘルプ 38
 - スタンドアロン・ヘルプ・ボリュームの例 57
 - タイトル 192
 - デスクトップ・ブラウザ・ボリューム 35
 - 登録 267
 - トピックの集まり 35
 - 表示 115, 121
 - ボリュームのファミリ 35

- 本のタイトルの指定 89, 140

ま

- マークアップ
 - ガイドライン 52
 - 簡易 40, 53
 - 正規 (SGML) 40
 - ヘルプ・ボリュームの例 54
- マークアップ・ガイドライン 52
- マークアップ言語
 - SGML 準拠正規マークアップ 40
 - SGML 準拠の正規マークアップ 219
 - 簡易マークアップ 39
- マークアップのリファレンス 133
- マニュアルのタイトル 192
- マニュアル・ページ
 - ハイパーリンク 91
 - 表示 244
 - リンクの作成 97
- マルチバイト言語のサポート 280

め

- メタ情報 51, 177
 - トピックの作成 66
 - トピックへのリンク 98
 - ボリュームのタイトル 192
- メッセージ・カタログ
 - DtHelp.msg ファイル 288
 - 作成 296
 - 翻訳 288
- メッセージ・カタログのソース・ファイル
 - DtHelp.msg 288
- メニュー
 - 検索 111
 - ナビゲート 31
 - ファイル 31
 - ヘルプ 31
 - [ヘルプ] メニューの提供 248
 - 編集 31
- メモ
 - 作成者のメモの挿入 110
- メモ・オプション 110

も

- 文字セット 280

サポートされているセットの名前 280
モデル
ヘルプ情報 25

や

矢印 207

ゆ

ユーザ

外国のユーザのための記述 296

情報の把握 37

ユーザを想定したヘルプのレビュー 42

よ

用語

新たに紹介する用語 190

定義 154

用語集での定義 114

用語集のマーク 112

用語集

新たに紹介する用語 190

作成 112

ヘルプ・ボリュームの内容 52

マークする用語 112

用語の定義 114

要素タグ 154

要素

ID をトピックに追加する 71

インライン要素の入力 88

開始と終了のタグ 52

階層構造 78

パラメータ 53

要素タグ

<chapter> 62

<!entity> 72

<abstract> 66

<book> 89

<chapter> 50, 78

<computer> 89

<copyright> 66, 144

<dterm> 145

<emph> 146

<!entity> 72, 147

<esc> 149

<ex> 84, 150

<figure> 73, 103, 152

<glossary> 112, 114, 154

<graphic> 105, 155

<head> 156

<hometopic> 64

<idx> 111, 160

<image> 79, 161

<item> 163

<keycap> 164

<labheads> 165

<lablist> 81, 165

<link> 96 - 98, 169

<list> 80, 172

<location> 71, 174

<memo> 110, 176

<metainfo> 66, 177

<newline> 178

<note> 179

<otherfront> 180

<otherhead> 83, 180

<p> 79, 106, 182

<procedure> 83, 184

<quote> 185

<rsect> 186

<rsub> 186

<s1...s9> 50, 62, 65, 187

<sort> 111, 160

<sub> 189

<super> 189

<term> 52, 96, 112, 114, 190

<title> 66, 192

<otherfront> 68

<user> 193

<var> 90, 194

<vex> 84, 195

<warning> 196

<xref> 197

トピックの追加 78

要素のタグ 133

要素のパラメータ 53

要約

CDE ヘルプタグ・マークアップ 133

ヘルプ・システム・コマンド 211

ら

ラベル付きリスト 165

ラベル付きリストのカラム・ヘッダ 165

り
リスト
 <labheads> 82, 165
 <lablist> 81, 165
 行頭文字 172
 行頭文字が付けられた 172
 項目 163
 項目の相互参照 163
 作成 80
 順番が付けられたリスト 172
 何も付いていない 172
 ラベル付き 165
 ラベル付きヘッダ 165
リスト内で使用される * (アスタリスク) 80
リストの項目 163
リソース
 DtNexecutionPolicy 102
 DtNhelpType 242
 DtNhelpVolume 242
 DtNlocationId 242
 DtNmanPage 244
 helpOnHelpVolume 261
 フォントの設定 290
 ヘルプ・ダイアログ・ボックス 232
リテラル
 コンピュータ・リテラルの表示 89
リンク

アプリケーション定義リンク 98
実行別名リンク 100
定義リンク 96
ハイパーテキスト 169
マニュアル・ページへのリンク 97
メタ情報 98

れ
例 197
 <user> の使用 193
 グラフィック・ファイル・エンティティ
 の宣言 74
 スタンドアロンのヘルプ・ボリューム 57
 逐語例の表示 195
 テキスト・ファイル・エンティティの宣
 言 74
 マークアップの使用 195
レビュー
 エラー 120
 ユーザを想定したヘルプ 42

ろ
ローカライゼーション 279
ロケール
 エンティティの作成 286
 ヘルプ・ボリュームの指定 286