共通デスクトップ環境 プログラマ概要

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

Part No: 805-5813-10
1998 年 11 月

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。日本サン・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ョ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ースマスタ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ースマスタ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、SunSoft、SunDocs、SunExpress、OpenWindows、NFS、PostScript、ToolTalk 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。

サンのロゴマークおよび 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、日本サン・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. 1998 All Rights Reserved.)

ATOK は、株式会社ジャストシステムの登録商標です。

ATOK7 は株式会社ジャストシステムの著作物であり、ATOK7 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。

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。

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。(Copyright (c) 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、日本サン・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: Programmer's Overview*

Part No: 805-3912-10

Revision A

© 1998 by Sun Microsystems, Inc.

Adobe PostScript

目次

はじめに **ix**

パート**I**　共通デスクトップ環境アーキテクチャの概要

1.　アーキテクチャの概要　**3**

概念　3

データ対話型 GUI　5

マルチユーザでの共用　6

デスクトップ管理　7

　セッション管理　7

　アプリケーション管理　9

　オブジェクト管理　10

　ウィンドウ管理　11

　スタイル管理　11

Motif GUI エンジン　12

　共通デスクトップ環境 Motif ツールキット　13

　共通デスクトップ環境 Motif ウィジェット　13

　GUI シェル　14

　GUI 構築　15

統合テクノロジ　16

　プロセスの実行　16

アプリケーション・メッセージ機構 16
データ型の作成 17
メソッドの実行 18
パートII 開発環境の概要
2. 開発環境の要件 **23**
共通デスクトップ環境の特性 23
根底となる基礎事項 25
既存アプリケーションの実行 26
ライブラリおよびヘッダ・ファイル 27
デモ・プログラム 27
マニュアル・ページ 27
開発環境ディレクトリ構造 28
64 ビットライブラリ 31
3. アプリケーションの開発 **33**
デスクトップ統合のレベル 33
デスクトップ命名規則 34
公開しているインタフェースと公開していないインタフェース 36
共通デスクトップ環境データベースのガイドライン 36
アプリケーションの初期化と libDtSvc 36
アプリケーション・ビルダ 37
デモ・プログラム 38
4. 移植性と保守性 **39**
移植性の問題 39
標準 39
Makefile 40
ファイル命名規則 41
ディスプレイ・サポート 41
共通デスクトップ環境の Motif ウィジェット・バイナリの互換性のガイドライン 42

5. 基本的なアプリケーションの統合方法 **43**
基本的な統合方法の特徴 43
基本的な統合方法情報の構成 44
基本的な統合方法の作業 45
6. 推奨する統合方法 **47**
ヘルプ・システム 48
ライブラリおよびヘッダ・ファイル 49
デモ・プログラム 49
ToolTalk メッセージ・サービス 49
デスクトップ・メッセージ・セット 50
ライブラリおよびヘッダ・ファイル 52
デモ・プログラム 52
セッション・マネージャ 53
ライブラリおよびヘッダ・ファイル 54
デモ・プログラム 54
ドラッグ&ドロップ 54
ライブラリおよびヘッダ・ファイル 56
デモ・プログラム 56
国際化対応 56
デモ・プログラム 57
標準フォント名 58
アプリケーション・フォント 58
インタフェース・フォント 60
デモ・プログラム 60
アプリケーションのエラー・メッセージの表示 61
エラー・メッセージの表示方法 61
エラー・ダイアログに表示する情報 62
メッセージ・ダイアログをオンライン・ヘルプへリンクする方法 62

ユーザ・カスタマイズの問題　62
使用する色の数　62
使用するフォント　63
アクセスのしやすさ　64
マウスのダブルクリック間隔　65
デモ・プログラム　65
7. オプションの統合方法　**67**
共通デスクトップ環境 Motif コントロール・ウィジェット　68
Motif 2.1 との互換性　70
ライブラリおよびヘッダ・ファイル　70
デモ・プログラム　71
データ型の作成　71
データベースの読み込み　72
データベースの照会　72
ライブラリおよびヘッダ・ファイル　73
デモ・プログラム　73
アクションの実行　73
ライブラリおよびヘッダ・ファイル　74
デモ・プログラム　74
ワークスペース・マネージャ　75
ライブラリおよびヘッダ・ファイル　76
デモ・プログラム　76
端末エミュレータ・ウィジェット　76
ライブラリおよびヘッダ・ファイル　77
デモ・プログラム　77
テキスト・エディタ・ウィジェット　77
ライブラリおよびヘッダ・ファイル　78
デモ・プログラム　78

カレンダ　79

ライブラリおよびヘッダ・ファイル　80

デモ・プログラム　80

デスクトップ Korn シェル (dtksh)　80

デモ・プログラム　81

A. 共通デスクトップ環境 **Motif　83**

Motif に追加された機能　84

拡張された機能　84

Motif ライブラリ　85

Motif ライブラリ (libXm)　85

Motif UIL ライブラリ (libUil)　85

Motif リソース・マネージャ・ライブラリ (libMrm)　85

デモ・プログラム　86

B. コンポーネントおよびガイドラインのリファレンス　**87**

索引　**91**

はじめに

『共通デスクトップ環境プログラマ概要』は、共通デスクトップ環境の開発環境と開発者のドキュメンテーション・セットについて、高度な概要を提供するものです。アプリケーションの設計と開発を始める前に、まずこのマニュアルをお読みください。

注 - このマニュアルでは、「共通デスクトップ環境」という用語と「デスクトップ」という用語を同義で使用します。

「はじめに」以外の部分では、共通デスクトップ環境の開発または実行環境用マニュアルを指すときの「共通デスクトップ環境」を省いています。たとえば、『共通デスクトップ環境プログラマ概要』は『プログラマ概要』と表現します。

対象読者

対象読者は次のとおりです。

- 新しく共通デスクトップ環境のアプリケーションを開発する、あるいは既存の OSF/Motif または OPEN LOOK アプリケーションをデスクトップへ統合するアプリケーション開発者
- 共通デスクトップ環境のデスクトップで実行するアプリケーションを含むプロジェクトの設計に関心のあるマネージャ、設計者、およびプロジェクトのリーダー

このマニュアルでは OSF/Motif は Motif と表記します。

このマニュアルの構成

このマニュアルは 2 つのパートから構成されています。パート I は、共通デスクトップ環境のアーキテクチャの概要で、実行時および開発時の環境の高度な情報を含んでいます。パート II ではアプリケーションの開発にあたって事前に知っておくと役立つ情報と、開発環境のコンポーネントを説明します。

次にこのマニュアルの章と付録について簡単に説明します。

第 1 章

共通デスクトップ環境アーキテクチャの概要を示します。

第 2 章

アプリケーションを開発する前に知っておくべき情報について説明します。

第 3 章

命名規則とそのガイドラインなど、共通デスクトップ環境アプリケーションの開発に固有の情報について説明します。

第 4 章

移植性があり保守性のあるアプリケーションを作成するときの問題点を説明します。

第 5 章

アプリケーションの起動方法を統合する (つまりデスクトップのアイコンをダブルクリックして起動する) 方法を説明します。

第 6 章

作成するアプリケーションを、他の共通デスクトップ環境のデスクトップ・アプリケーションと同じようなルック&フィールを持つものにし、相互に操作できるようにするための、すべてのコンポーネントの概要とガイドラインについて説明します。

第 7 章

追加した機能の必要性に応じてアプリケーションに組み込むコンポーネントの概要を示します。

付録 A

Motif 1.2.3 と共通デスクトップ環境 Motif との相違点を説明します。

付録 B

すべての開発環境コンポーネントとガイドラインを、対応するライブラリ、ヘッダ、ファイル、ドキュメントと共にアルファベット順に示します。

関連文書

Motif については次のマニュアルを参照してください。

- 『*OSF/Motif Programmer's Guide, Release 1.2*』by Open Software Foundation, 11 Cambridge Center, Cambridge, MA 02142, published by PTR Prentice Hall, Englewood Cliffs, NJ 07632

  (翻訳 : 株式会社 日立製作所ソフトウェア開発部訳、『*OSF/Motif* プログラマーズ・ガイド リリース *1.2*』、株式会社トッパン、1993)

- 『*OSF/Motif Programmer's Reference, Release 1.2*』 by Open Software Foundation, 11 Cambridge Center, Cambridge, MA 02142, published by PTR Prentice Hall, Englewood Cliffs, NJ 07632

  (翻訳 : 株式会社 日立製作所ソフトウェア開発部訳、『*OSF/Motif* プログラマーズ・リファレンス リリース *1.2*』、株式会社トッパン、1993)

- 『*OSF/Motif Reference Guide*』 by Douglas A. Young, published by PTR Prentice Hall, Englewood Cliffs, NJ 07632
- 『*OSF/Motif 1.2 Style Guide*』 by Open Software Foundation, 11 Cambridge Center, Cambridge, MA 02142, published by PTR Prentice Hall, Englewood Cliffs, NJ 07632

  (翻訳 : 株式会社 日立製作所ソフトウェア開発部訳、『*OSF/Motif* スタイル・ガイド リリース *1.2*』、株式会社トッパン、1993)

注 - 『共通デスクトップ環境 スタイル・ガイド』は、共通デスクトップ環境の『*CDE 2.1/Motif 1.2 Style Guide and Glossary*』の拡張版です。

- 『*OSF Application Environment Specification (AES) User Environment Volume, Revision C*』 by Open Software Foundation, 11 Cambridge Center, Cambridge, MA 02142, published by PTR Prentice Hall, Englewood Cliffs, NJ 07632

POSIX の詳細は、IEEE Std 1003.1-1990 標準を参照してください。この標準は次の宛先までご注文ください。

IEEE Service Center

445 Hoes Lane

P.O. Box 1331

Piscataway, NJ 08855

Xlib の詳細は次のマニュアルを参照してください。

- 『*Xlib Programming Manual for Version 11 (Volume 1)*』 by Adrian Nye, published by O'Reilly & Associates, Sebastopol, CA, 95472

  (翻訳 : 坂下 秀、荒井 美千子、西垣内 昌喜、藤井 裕史　監訳、『*Xlib* プログラマーズ・マニュアル』、ソフトバンク株式会社、1993)

- 『*Xlib Reference Manual for Version 11 (Volume 2)*』 published by O'Reilly & Associates, Sebastopol, CA, 95472

  (翻訳 : 坂下 秀　監訳、『*Xlib* リファレンス・マニュアル』、ソフトバンク株式会社、1993)

Xt の詳細は次のマニュアルを参照してください。

- 『*X Toolkit Intrinsics Programming Manual, (Volume 4)*』 by Adrian Nye and Tim O'Reilly, published by O'Reilly and Associates, Sebastopol, CA 95472.

  (翻訳 : 今泉 貴史　監訳、『*X* ツールキット・イントリンシクス・プログラミング・マニュアル』、ソフトバンク株式会社、1992)

- 『*X Toolkit Intrinsics Reference Manual, (Volume 5)*』 edited by Tim O'Reilly, published by O'Reilly and Associates, Sebastopol, CA 95472.

  (翻訳 : 今泉 貴史　監訳、『*X* ツールキット・イントリンシクス・リファレンス・マニュアル』、ソフトバンク株式会社、1994)

ToolTalk™ の詳細は次のマニュアルを参照してください。

- 『*The ToolTalk Service: An Inter-Operability Solution*』published by SunSoft Press and PTR Prentice Hall, Englewood Cliffs, NJ 07632, ISBN 0-13-088717-X

  (翻訳 : サンソフト監修、データリンク訳、『*The ToolTalk* サービス リファレンスマニュアル』、アスキー出版局)

- 『*ToolTalk and Open Protocols: Inter-Application Communication*』by Astrid Julienne and Brian Holtz, published by SunSoft Press and PTR Prentice Hall, Englewood Cliffs, NJ 07632, ISBN 013-031055-7

開発環境用のマニュアルは、このマニュアル以外に次のマニュアルがあります。

- 『共通デスクトップ環境 スタイル・ガイド』

- 『共通デスクトップ環境　アプリケーション・ビルダ・ユーザーズ・ガイド』
- 『*Solaris* 共通デスクトップ環境　プログラマーズ・ガイド』
- 『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)』
- 『共通デスクトップ環境　*ToolTalk* メッセージの概要』
- 『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　*Dtksh* ユーザーズ・ガイド』
- 『*Solaris* 共通デスクトップ環境　*Motif* への移行』
- 『*Common Desktop Environment: Product Glossary*』
- 『オンライン・マニュアル・ページ』

上記のマニュアルの詳細は、次節の xiiiページの「開発環境用マニュアル」を参照してください。

実行環境用のマニュアルは次のとおりです。

- 『*Solaris* 共通デスクトップ環境ユーザーズ・ガイド』
- 『*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』
- 『*Solaris* 共通デスクトップ環境の紹介』
- 『*Solaris* 共通デスクトップ環境への移行』
- 『日本語 *Solaris* 共通デスクトップ環境のインストール』
- 『オンライン・ヘルプ・ボリューム』

注 - 『*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』には、アプリケーションをデスクトップに統合するときに役立つ情報が説明されています。

開発環境用マニュアル

この節では、このマニュアル以外の各開発者用マニュアルの概要を説明します。

共通デスクトップ環境　スタイル・ガイド

『共通デスクトップ環境 スタイル・ガイド』は、アプリケーション設計スタイルのガイドラインと、共通デスクトップ環境のアプリケーション・レベルの認定の要件を説明します。これは Motif バージョン 1.2 の要件に共通デスクトップ環境固有の要件を追加したものです。

チェックリストはモデル・キーボードの形式を使用して、キーについて説明しています。チェックリストは英語ロケールで左から右へ書かれる言語を対象としたアプリケーションを設計することを前提としています。キーボード入力を示す箇所では、Motif モデル・キーボードの文字でキーが示されています。マウス・ボタンは、マウスのボタンの数に依存しない動作を示すために、仮想ボタン名称を使用して説明されています。

このマニュアルでは、アプリケーション設計者が一貫性のあるアプリケーションを開発し、アプリケーション内の動作に一貫性を持たせるために役立つ情報を提供しています。

共通デスクトップ環境　アプリケーション・ビルダ・ユーザーズ・ガイド

共通デスクトップ環境アプリケーション・ビルダ (App Builder とも呼ばれます) は、共通デスクトップ環境アプリケーションを開発するための対話型ツールです。このツールはアプリケーションのグラフィカル・ユーザ・インタフェース (GUI) の構築と、デスクトップの多くの便利なデスクトップ・サービス (ヘルプ、ToolTalk、ドラッグ&ドロップなど) の組み込みとを容易にする機能を提供します。『共通デスクトップ環境アプリケーション・ビルダ・ユーザーズ・ガイド』では、パレットから「オブジェクト」をドラッグ&ドロップしてインタフェースを作成する方法を説明します。また、インタフェース内のオブジェクト間の接続方法、アプリケーション・フレームワーク・エディタを使用してデスクトップ・サービスとの統合を簡単にする方法、C コードの生成方法、App Builder 出力にアプリケーション・コードを追加して最終的なアプリケーションを作成する方法についても説明しています。

Solaris 共通デスクトップ環境プログラマーズ・ガイド

このマニュアルは、3 つのパートから構成されています。各パートに、共通デスクトップ環境の各要素の詳しい説明、概念図、各要素の具体的な使い方の説明、コーディング例があります。

パート I 「基本的な統合方法」では、アプリケーション・レベルと印刷レベルの登録方法を説明します。

パート II 「推奨する統合方法」では、既存のアプリケーションを共通デスクトップ環境に統合する方法を説明します。

パート III 「オプションの統合方法」では、新しいアプリケーションをセッション・マネージャおよびドラッグ＆ドロップと統合する方法を説明します。また、ロケールがログイン・マネージャ、ウィンドウ・マネージャ、および端末エミュレータに与える影響についても説明します。

これらの中でコンポーネントのアプリケーション・プログラム・インタフェース (API) の概要を、関連するマニュアル・ページと相互参照しながら説明しています。詳細はマニュアル・ページで説明しています。

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)

『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)』は、アプリケーション・ソフトウェアのためのオンライン・ヘルプの開発方法について説明しています。ヘルプ・トピックの作成方法と、オンライン・ヘルプを Motif アプリケーションに統合する方法も説明しています。

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・ヘルプ情報の設計、作成、および表示する設計者
- 完全に統合されたヘルプ機能を提供するアプリケーション・ソフトウェアを作成する開発者

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 ではアプリケーションのヘルプを設計するために設計者と開発者とが協力して行う役割について説明しています。パート II はオンライン・ヘルプを構成および記述する設計者に必要な情報を説明しています。パート III 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ています。パート IV は、国際化対応環境向けのオンライン・ヘルプの作成について、設計者とプログラマに必要な情報を説明しています。

共通デスクトップ環境 ToolTalk メッセージの概要

『共通デスクトップ環境 *ToolTalk* メッセージの概要』では、メディア交換およびデスクトップ・サービスのメッセージ・セット規則に準拠したアプリケーションを作成するための便利なルーチンとして提供される ToolTalk のコンポーネント、コマンド、およびエラー・メッセージについて説明しています。このマニュアルは ToolTalk サービスを使用して他のアプリケーションと相互運用するアプリケーションを作成または保守する開発者のためのマニュアルです。

『*ToolTalk* メッセージの概要』では、一般的な ToolTalk の機能については説明していません。ToolTalk サービスの詳しい説明は、『*ToolTalk* リファレンスマニュアル』を参照してください。ToolTalk をより簡単に使用するには、『*ToolTalk and Open Protocols: Inter-Application Communication*』を参照してください。いずれも xi ページの「関連文書」にリストされています。

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プログラマーズ・ガイド (国際化対応編)』は、アプリケーションを簡単にローカライズして、さまざまな言語と文化的規則を一貫したユーザ・インタフェースでサポートできるように、アプリケーションの国際化対応について説明しています。

特に、次の情報を提供しています。

- 開発者に対し、世界中で使用できるようなアプリケーションを書くためのガイドラインとヒントを提供しています。
- デスクトップのさまざまな階層にまたがる国際化トピックの全体像を提供しています。
- 参考資料および詳細な記述のあるマニュアルを示しています。標準の規格文書を参照する場合もあります。

このマニュアルは、既存の参考資料または概念的なドキュメントの説明をそのまま掲載するのではなく、特定の国際化対応トピックに関するガイドラインと規則を説明するものです。国際化トピックに焦点を置くもので、オープン・ソフトウェア環境の特定のコンポーネントや階層について説明したものではありません。

共通デスクトップ環境 Dtksh ユーザーズ・ガイド

『共通デスクトップ環境 *Dtksh* ユーザーズ・ガイド』では、デスクトップ Korn シェル (dtksh) スクリプトで Motif アプリケーションを作成する方法を説明しています。開発者が作業を始めるにあたって必要な基本的な情報に加え、徐々に複雑になるスクリプトの例を示します。

このマニュアルは特定の作業に適応したシェル形式のスクリプト環境を探している開発者を対象としています。Korn シェル・プログラミング、Motif、Xt イントリンシクスの知識と、Xlib についてのある程度の知識があることを前提としています。

Solaris 共通デスクトップ環境 Motif への移行

このマニュアルは、アプリケーション開発のために Solaris の Motif を使用する方法と、OPEN LOOK または Motif アプリケーションを Solaris CDE へ移植する方法について説明します。

Common Desktop Environment: Product Glossary

『*Common Desktop Environment: Product Glossary*』は、共通デスクトップ環境で使用する用語の包括的なリストです。この用語集はデスクトップのすべてのユーザにとって、ソースおよび参照の基本となります。この用語集の読者はエンド・ユーザ、開発者、移植者など多岐にわたるため、読者や、用語の由来や、グラフィカル・ユーザ・インタフェースでその用語を使用する共通デスクトップ環境コンポーネントについての情報も、用語定義の書式に含まれています。

マニュアルの注文方法

SunDocs™ プログラムでは、米国 Sun Microsystems™, Inc. (以降、Sun™ とします) の 250 冊以上のマニュアルを扱っています。このプログラムを利用して、マニュアルのセットまたは個々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、米国 SunExpress™, Inc. のインターネットホームページ `http://www.sun.com/sunexpress` にあるカタログセクションを参照してください。

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、およびディレクトリ名を示します。または、画面上のコンピュータ出力を示します。	`.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`	ユーザが入力する文字を、画面上のコンピュータ出力とは区別して示します。	`system%` **`su`** `password:`
AaBbCc123	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` *filename* と入力します。
『 』	参照する書名を示します。	『ユーザーズ・ガイド』を参照してください。
「 」	参照する章や節を示します。また、ボタンやメニューなど、強調する単語を囲む場合にも使用します。	第 1 章「基本スキル」を参照してください。
[ ]	アイコン、ボタン、メニューなどのラベル名を使用します。	[了解] ボタン

注 - \ (バックスラッシュ) は、デバイスによって ¥ (円記号) で表示されるものがあります。

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`** [*filename*]

■ Bourne シェルおよび Korn シェルのプロンプト

```
system$ command [filename]
```

■ スーパーユーザのプロンプト

```
system# command [filename]
```

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

ただし AnswerBook2™ では、ユーザ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。

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。

パート I 共通デスクトップ環境アーキテクチャの概要

第 **1** 章

アーキテクチャの概要

この章では、共通デスクトップ環境のハイレベルなアーキテクチャを説明します。デスクトップ実行環境の詳細は、実行時のマニュアル・セットとオンライン・ヘルプ・ボリュームを参照してください。デスクトップ開発環境のコンポーネントの詳細は、第 6 章、第 7 章、付録 A、開発環境用マニュアル・セット、およびオンライン・マニュアル・ページを参照してください。

3ページの「概念」

5ページの「データ対話型 GUI」

6ページの「マルチユーザでの共用」

7ページの「デスクトップ管理」

12ページの「Motif GUI エンジン」

16ページの「統合テクノロジ」

概念

共通デスクトップ環境アーキテクチャにはプロセスの相互関係が多数あります。X クライアント、ウィンドウ・マネージャ、X Window System サーバの 3 つの関係は、比較すると簡単です。共通デスクトップ環境が適用される範囲は広いですが、システムの階層は Motif、Xt、Xlib ほどに厳格なものではありません。この章ではテクノロジをグループ化し、それぞれのデスクトップ・コンポーネントを図解します。共通デスクトップ環境は次のように分類されます。

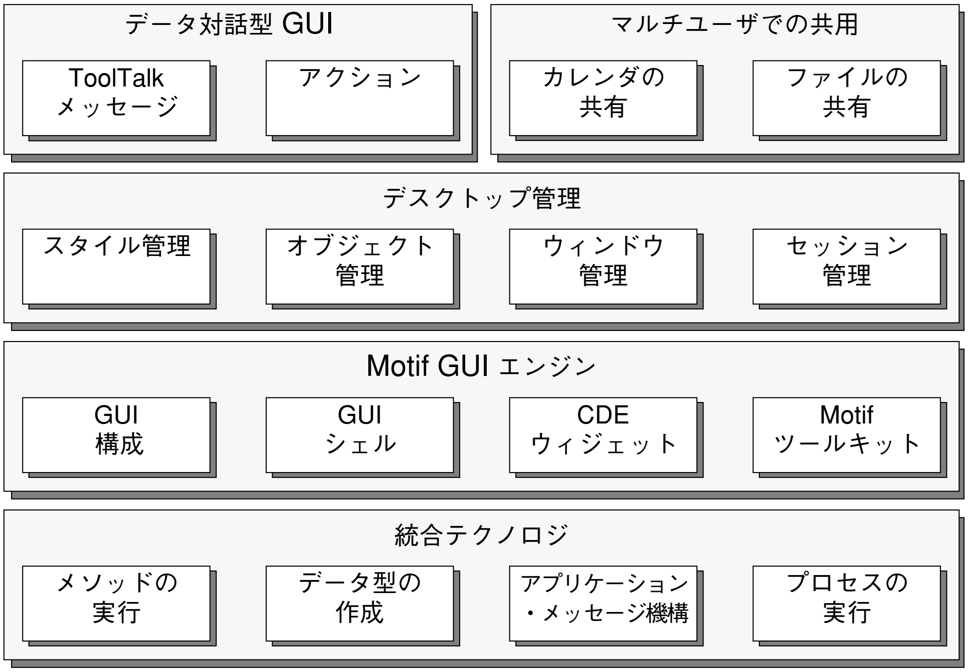

図 1–1　共通デスクトップ環境の概念

■ データ対話型 GUI (グラフィカル・ユーザ・インタフェース)

ユーザ対話に使用でき、他のアプリケーションからも呼び出せるアプリケーション・レベルのコンポーネントです。ウィジェットより単位が大きいプログラミング・コンポーネントと考えてください。

■ マルチユーザでの共用

特にカレンダ管理、ネットワーク・リソースの命名、ネットワーク・ファイル共有の領域において、ネットワーク上での、ユーザ間共用を可能にする API (アプリケーション・プログラム・インタフェース) を定義し使用します。

■ デスクトップ管理

デスクトップのエンティティ間の視覚的な関連性を調整するコンポーネントを提供します。ウィンドウ・マネージャ、ワークスペース・マネージャ、セッション・マネージャ、アプリケーション・マネージャ、ファイル・マネージャ、スタイル・マネージャ、およびフロントパネルがこれに含まれます。

■ Motif GUI エンジン

ユーザが使用できるコントロールを実現するコンポーネントで、共通デスクトップ環境の Motif ツールキット、追加ウィジェット、GUI シェル (デスクトップ Korn シェル)、GUI 構築ツール (アプリケーション・ビルダ) などです。

- 統合テクノロジ

  GUI を生成しませんが、デスクトップの他のコンポーネントでインフラストラクチャとして使用されるテクノロジです。プロセス実行の制御、アプリケーション・メッセージ機構 (機能およびプロトコル)、データ型の作成、メソッドの実行などが含まれます。

データ対話型 GUI

共通デスクトップ環境は登録サービスである ToolTalk メッセージ・サービスを提供します。これによりアプリケーションは使用可能なサービス・プロバイダを見つけることができます。ToolTalk はローレベルのメッセージ機構インフラストラクチャを提供します。アクション・システムと呼ばれるガイド機能は、アプリケーションに対する従来の UNIX コマンド行インタフェースと共通デスクトップ環境で推奨する ToolTalk インタフェースの両方の上に一貫した抽象的な階層を提供します。アクションは意味を持つエンティティとして、ソフトウェアの上位レベルによってエンド・ユーザに示されます。アクションおよび ToolTalk の詳細は、いずれも 16ページの「統合テクノロジ」で説明します。

デスクトップには、アクションまたは ToolTalk API が使用できるコンポーネントがあります。ディレクトリの表示、印刷ジョブの発行、[ごみ箱] の内容表示、テキストの編集、ヘルプ情報の表示、カレンダ・アポイントの作成、およびメール・メッセージの作成などを行う GUI があります。

アクションや ToolTalk メッセージのサポートをアプリケーションに組み込んで、アプリケーション固有のサービスをデスクトップおよび他のアプリケーションで使用できるようにすることもできます。特に、専用フォーマット・データおよび標準フォーマット・データの両方で、作成、表示、編集、印刷のサービスを行います。このように、拡張可能な一連のデータを受け入れるようにコード化されたアプリケーションは、システムにメディア・ハンドラを追加すればするほどその機能が自動的に向上します。このようなアプリケーションには、共通デスクトップ環境のファイル・マネージャ、フロントパネル、およびメール・プログラム接続 GUI などがあります。

「メディア」は、情報をユーザに伝えるものを指す一般用語として使用されています。デスクトップはアポイント、メール・メッセージ、メール・フォルダ、テキスト、アイコン、およびヘルプ・データ用のメディア・ハンドラを提供します。ベンダは、PostScript™、あらゆる種類のイメージ・ファイル・フォーマット、オーディオ・データなどその他のメディア・ハンドラを追加することによりデスクトップを拡張できます。

マルチユーザでの共用

ToolTalk およびアクションの機能はアプリケーション間の連携を強化しますが、デスクトップもユーザ間での共用テクノロジを定義します。つまり、共有のユーザ・データへの分散アクセスです。デスクトップはいくつかの基本共有機能を定義し、既存の機能の一番上に構築しています。

既存機能上での構築の例としては、カレンダ管理のリモート・プロシージャ・コール (RPC) クライアント／サービスがあります。デスクトップは、ユーザがアポイント情報を共有できるようにクライアント側ライブラリと API、RPC プロトコル、およびデーモン／サービスを提供します (API は X.400 アプリケーション・プログラミング・インタフェース・アソシエーション (XAPIA) によって標準化され、UNIX、PC、パームトップ・カレンダで相互に使用できる標準になります)。RPC プロトコルにより、ユーザは他のユーザのカレンダをブラウズし、直接編集できます。アクセスはユーザ固有のアクセス・コントロール機能で制御されます。カレンダはホストに結びつけられ、カレンダのデータはホスト固有のデーモンが管理します。デスクトップは `user@host` フォーマットによりカレンダを管理します。

共通デスクトップ環境は従来の分散ファイル・システムを使用して、ネットワークで共有できるファイルを管理します。分散ファイル・システムに依存しないインタフェースを提供するため、デスクトップは ホスト関連ファイル名をローカルに表現できるファイル名に変換する API を提供します。デスクトップは NFS™ システムに基づいていますが、他の分散ファイル・システムへ移植できます。デスクトップ・ファイル名マッピング API を使用すると、遮断されたファイル名オブジェクトを構築したり、ネットワークのデスクトップ・クライアント間で渡したり、ホスト固有の方法で解釈できます。また、プログラミング・タスクとエンド・ユーザ・メタファを単純にするため、共通デスクトップ環境のアプリケーションはローカル・ファイル・パスとしてリモート・ファイルのリファレンスを提供できます。

マルチユーザの共用アプリケーションを構築する基礎となるものの 1 つに、ファイルの共有機能があります。ネットワーク・ファイルの命名規則は、ファイル範囲指

定と呼ばれる ToolTalk ファイル共有機能と結合して、ファイル共有によってマルチユーザでの共用を可能にします。ファイル範囲指定は、単純で排他的なアクセス・コントロールのためのメカニズムをもっています。協調するクライアントは、ファイル範囲指定アクセスを使用してファイルへのアクセスを調整できます。たとえばファイルに排他的アクセスをしているアプリケーションは、他のアプリケーションがファイルへの排他的アクセスを望んでいるときに、ユーザがアクセスを終了したかどうかを確認します。

デスクトップ管理

共通デスクトップ環境に対応する物理メタフォは、たくさんの机 (ワークスペース) に囲まれて、椅子に座っているユーザの一人のようなものです。ユーザが椅子を (フロントパネルの押しボタンをクリックすることによって) 回転させると、他の机にアクセスできます。各机では次のものを使用できます。

- フォルダ (ディレクトリ) およびレポート (ファイル) が入っている引き出しの集まり (ファイル・マネージャ表示)
- デスクトップ (ウィンドウ) で使用する書類の集まり。(アイコンとして) わきに置いてあるものもありますが、すぐに手の届く範囲にあります。
- 時計、日付、新規メールの表示、ごみ箱の内容表示を連続して表示する (フロントパネル・アイコンによる)。
- アポイント・ブック (カレンダ)、メモ帳 (テキスト・エディタ)、端末 (エミュレータ)、メール・ボックス (メール・プログラム)、プリンタ (印刷マネージャ)、オフィス照明コントロール (スタイル・マネージャ)、電子エージェント・リスト (アプリケーション・マネージャおよびフロントパネル個人用ツール・ボックス)、ガイドブック (ヘルプ) への直接アクセス

ユーザはオブジェクトをドラッグ&ドロップして位置を変更しコピーします。サービスにオブジェクトをドロップすると、アポイント・スケジュール、編集、メール作成、印刷などの補助を受けられます。

セッション管理

デスクトップの状態は保存しておくことができます。後で、あるいは別のコンピュータで、デスクトップの状態を再現できます。セッションは、ユーザのデスクトップの状態の、ある瞬間におけるスナップショットです。共通デスクトップ環境

は次のような 2 つのセッションをサポートし、いずれかをユーザが選択できるようにしています。

- ホーム・セッション

  常に起動時と同じ方法で再組み立てされる、デスクトップの状態のスナップショット

- 現在のセッション

  ログアウト時に保存されるデスクトップの状態

共通デスクトップ環境セッション・マネージャはこれらのアクティビティを調整しますが、アプリケーションの状態の保存についてはアプリケーションが責任を持ちます。

デスクトップはセッション管理の X11R5 クライアント間通信規約のスタイルを使用します。これには、トップレベル・ウィンドウの属性を設定するための規約のほとんどが含まれています。デスクトップはこれを拡張し、アプリケーションが状態を格納できる特定のファイルを割り当てる機能を提供します。アプリケーションを再起動するときにコマンド行フラグがこのファイルを指します。複数のトップレベル・ウィンドウを管理するアプリケーションは、それぞれの状態を保存しなければなりません。

セッションは特定のユーザに関連付けられます。共通デスクトップ環境では、ログイン・マネージャがユーザの最初のログインに責任を持ちます。ログイン・マネージャは、UNIX ログイン・プログラムに代わる GUI です。通常は入力されたパスワードがユーザの登録したパスワードかどうかチェックします。ただし、ベンダは各プラットフォームに最適な認証スキーマを提供できます。

ログイン・マネージャはネットワークを認識します。ユーザは、ホスト A がサーバである X ディスプレイに向かい、ホスト B からのセッションを実行することにより、ユーザのディスクトップにログインできます。このセッションにより、ユーザの通常のファイルやホスト B のサービスにアクセスできます。これはデスクトップの X11 ディスプレイ・マネージャ (XDM) としてログイン・マネージャを動作させることにより可能です。XDM コントロール・プロトコル (XDMCP) は、ネットワーク上で X11 ウィンドウ・サーバと XDM との間で使用されます。ログイン・マネージャはログイン・ウィンドウまたはホスト選択ウィンドウを、XDM サービスを要求する任意の X11 サーバに表示します。これにより、XDMCP を認識する X 端末で共通デスクトップ環境が使用できます。

X サーバへ接続するため、デスクトップは X マジック・クッキー・スキーマを使用してアクセスをコントロールします。あるホスト・マシン上でユーザが、セッション所有者のホーム・ディレクトリ内の特定のファイルを読むことができる場合、X サー

バへアクセスが許可されます。このユーザごとの利用権の他にホストごとの利用権があります。X マジック・クッキー・スキーマを使用して X サーバへ接続できない X11R4 以前のクライアントをサポートするインストールを行う場合に便利です。

X リソース・ファイルは共通デスクトップ環境のセッションのコンテキストで次のように処理されます。まず、一連の共通デスクトップ環境のデフォルト・リソースはホストのリソース・ファイルとマージされ、次にユーザの `$HOME/.Xdefaults` ファイルが続き、次にスタイル・マネージャでのユーザ対話により変更されるリソースのセッション固有ファイルが続きます。その結果はルート・ウィンドウの `RESOURCE_MANAGER` 属性に格納されます。精密なカスタマイズを可能にするために、リソース・ファイル上に C プリプロセッサを実行します。

アプリケーション管理

エンド・ユーザがネットワーク環境を十分に活用できない理由の 1 つに、リモート・アプリケーションへアクセスしづらいという点があります。共通デスクトップ環境は以下の内容に関する規約を提供します。

- ユーザがリモートで実行できるアプリケーションのインストール
- 使用可能なアプリケーションのユーザのナビゲーション
- リモート・アプリケーションの実行

ユーザは使用可能なアプリケーションの集まりをアプリケーション・マネージャと呼ばれる GUI ツールでブラウズできます。アプリケーションは簡単にアクセスできるようデスクトップにドラッグできます。リモート・アプリケーションでも、実行アプリケーションのネットワークの位置を隠したまま、単にダブルクリックするだけで起動できます。ユーザはローカル・アプリケーションとリモート・アプリケーションとの違いに気付きません。

このようなネットワークの透過性は、アプリケーション・サーバとして指定したネットワーク・ホスト上にアプリケーションをインストールすると実現されます。デスクトップに関するインストールの一部では、アプリケーションのインストール階層の規定の場所に特定のファイルを置く必要があります。アプリケーション・サーバはそれが提供するアプリケーションのリストを管理します。ネットワークの各ホストは、ユーザがデスクトップにログインしたときに照会するネットワーク上のアプリケーション・サーバのリストを管理します。このプロセスはアプリケーションの収集といいます。これにより、動的に生成されたアクションのファイル階層がフォルダに配置されます (アクションとは、アプリケーションの開始など、エンド・ユーザが起動できるオペレーションのことです)。

共通デスクトップ環境のアプリケーション・マネージャはエンド・ユーザに対してファイル・システムの特殊な表示を提供します。アプリケーションはグループ化され、グループは (ディレクトリ階層のように) ネストできます。アプリケーションのインストール・スクリプトは、アプリケーションをグループと関連づけます。この関連は、アプリケーション・サーバの構成の一部としてシステム管理者が変更できます。アプリケーション・マネージャが表示するアクションのセットと配列は、一般に複数のユーザ間で共有されているシステム・リソースです。ユーザは表示を変更できません。

ユーザはアプリケーション・マネージャからデスクトップ、ファイル・マネージャ、フロントパネルなどへアイコンをドラッグできます。収集されたアプリケーションが有効な間、関連したアクションは有効です。アクションは抽象化された形式および間接形式で示されるため、アプリケーションの実際の位置はいつでも変更できます。この変更はエンド・ユーザに対して透過的です (詳細は 18ページの「メソッドの実行」で説明します)。ユーザはアクション・アイコンをダブルクリックして、アクションを起動できます。

オブジェクト管理

共通デスクトップ環境は、完全なオブジェクト指向のインフラストラクチャに依存せずに、オブジェクト指向のシステム属性を取り込んでいます。デスクトップはユーザが選択および移動でき、これを任意の場所でドロップするとセマンティックを作成するグラフィック・オンスクリーン・イメージを提供します。これらはオブジェクトとしてユーザに表示されます。ファイル・マネージャは、ファイル・システム内でファイルおよびディレクトリ・オブジェクトをグラフィカルにブラウズおよび変更する方法を提供し、オブジェクトの考え方を実現しています。また、アクションを呼び出すための GUI を提供します。ユーザがファイルを選択すると、選択したファイルのタイプに定義されたアクションがユーザに示されます。

デスクトップ・ベースのアプリケーションが管理するオブジェクトは、ファイルベースである必要はありません。メモリ・バッファ内でデスクトップ・オブジェクトを示すこともできます。共通デスクトップ環境のメール・プログラムは、スクロール可能なパネルのアイコンでメッセージへのアタッチメントを表示することにより、MIME (Multipurpose Internet Mail Extensions) メッセージを処理します。これらはドラッグ&ドロップなどを行なっている間、ファイルベースのオブジェクトのように動作します。ユーザはファイル・マネージャとメール・プログラムとの間でドラッグできます。ドラッグ&ドロップを使用するアプリケーションは、ファイルベースのオブジェクトとバッファベースのオブジェクトの両方をサポート

することにより、この重要なユーザ・モデルを管理します。デスクトップ・ドラッグ&ドロップ API およびプロトコルがこれを容易にします。

ウィンドウ管理

ウィンドウ・マネージャは、基本的には、フロントパネル GUI とワークスペースを提供するよう拡張した Motif 1.2 ウィンドウ・マネージャです。

フロントパネルは、多くのウィンドウ・マネージャがサポートするルート・ウィンドウ・メニューのグラフィック・バージョンと見なすことができます。また、ユーザが共通オブジェクトを使用できるように変更したオブジェクト・マネージャとも見なされます。フロントパネルは動的システム情報を示し、ユーザがアクションおよびシステム機能を起動できるようにします。ユーザはアプリケーション・マネージャおよびファイル・マネージャからサブパネルへアクション・アイコンをドラッグ&ドロップすることにより、フロントパネルを動的にカスタマイズします。アプリケーションは、フロントパネルを拡張する特殊構成ファイルを備え、ドロップの動作やドロップ領域のアニメーションによるフィードバックなどを定義します。これらの構成ファイルはカスタマイズの設定に依存してオプションでインストールできます。図 1–2 に一般的なデスクトップ・フロントパネルを表示します。

図 1–2　一般的なフロントパネル

ワークスペースとは、仮想デスクトップと見なされるウィンドウ・マネージャがサポートする概念です。アプリケーション・ウィンドウは、1 つ以上、またはすべての使用可能なワークスペースに存在します。通常は、ユーザのカスタマイズの一部としてアプリケーション・ウィンドウがどのワークスペースに存在するかをユーザが決定します。セッションの再起動時にどのワークスペースにアプリケーションが表示されるかを明示的に指定する以外には、ワークスペース API を使用しないでください。一般にアプリケーションを複数のワークスペースに置くとユーザの優先権が無効になるので、置かないでください。

スタイル管理

スタイル・マネージャにより、ユーザは GUI を使用してデスクトップをカスタマイズできます。ユーザはほとんどの共通カスタマイズ・オプションについて X リソー

スなどの高度な概念を気にしないですみます。スタイル・マネージャは、背景の設定、キーボード設定、マウス設定、スクリーンセーバ・オプション、ウィンドウ管理、セッション管理などデスクトップ全体の属性を制御できるようにします。これらの属性は、アプリケーションに直接影響を与えないか、X サーバまたはウィンドウ・マネージャによって間接的に影響を与えます。

アプリケーション開発者は、フォントの選択、カラーの選択、入力デバイス・マッピングによってより直接的に影響を受けます。Motif ツールキットおよび共通デスクトップ環境は、これらの設定のほとんどをウィジェットに対して透過的に処理します。ただし、アプリケーションがユーザのフォントとカラーの設定に対応する場合、アプリケーションはデスクトップに対して、より統合されているように見えます。直接マウスで対話するアプリケーションは、たとえばマウス・ボタンのダブルクリックに同じ最小間隔の値 (`multiClickTime` リソース) を使用するなど他のアプリケーションと一貫した動作をすれば、デスクトップにより統合されているように感じられます。

プラットフォーム・ベンダのディスプレイと使用可能なフォント・セットとの相違点をなくすため、共通デスクトップ環境は実際のフォント名の二次的な名前であるフォント別名を定義します。デスクトップが使用するように別名を使用してください。

スタイル・マネージャはユーザにカラー選択オプションを提供して、デスクトップのカラー・スキーマを調整できるようにします。このカラー情報は共通デスクトップ環境だけで使用されるものです。ウィジェットをサブクラス化するアプリケーションは、継承されたバックグラウンド・ピクセル値を参照することによって、カラー・スキーマの一部に間接的にアクセスできます。`XmGetColors()` を呼び出すと、3D シャドウ・カラーが生成されます。

共通デスクトップ環境は、アイコンで使用するような固定のカラーに対してカラーの使用法を指定しません。ただしこの場合、カラーの共有を増進するため、アプリケーションは共通デスクトップ環境アイコン・エディタが提供するカラーを使用してください。

Motif GUI エンジン

Motif ツールキットはデスクトップの GUI エンジンとして考えてください。この節では共通デスクトップ環境 Motif、共通デスクトップ環境ウィジェット、および Motif プログラミングの代替モードについて説明します。

共通デスクトップ環境 **Motif** ツールキット

共通デスクトップ環境 Motif ツールキットは、Motif 1.2.3 のバグを修正し、機能を拡張し、いくつかの新機能を加えたものです。新機能を使用できるよう、明示的にリソースを設定してください。追加された機能には、ファイル選択ボックス GUI の変更、既存のリソースの別のデフォルト設定 (あらかじめデフォルトのボーダ幅を目立たせてあります)、カラー管理機能の向上、エラー・メッセージの国際化対応、使いやすさの向上 (OPEN LOOK ユーザの共通デスクトップ環境への移行を容易にする効果があります) などがあります。

共通デスクトップ環境 Motif と Motif 2.0 も互換性が高くなっています。共通デスクトップ環境 Motif にある機能のほとんどが Motif 2.0 に導入されています。したがって、開発者はアプリケーションを共通デスクトップ環境 Motif でコンパイルし、Motif 2.0 に再リンクして、アプリケーションを正常に実行できます。ウィジェットサイズの変更からプログラムを保護するように設計されている Motif 1.2 サブクラス化ガイドラインに従っていないウィジェットのサブクラス化は、うまく動作しない可能性が高くなります。

ドラッグ&ドロップ簡易階層は、Motif 1.2 ドラッグ&ドロップ API の最上位に追加されます。また、共通デスクトップ環境は Motif 1.2 登録済みドラッグ・フィードバック・プロトコルを使用します。ドロップ・サイト・ドラッグ・マネージャ・プロセスは、デスクトップの可視ドロップ領域を追跡しています。ドラッグ・ソース・クライアント・プロセスはこのデータを使用してドラッグ・フィードバック対話を管理します。ドロップ領域の制限付きドラッグ時の有効性は、ドロップ時まですべて継承され、ドロップが失敗した場合は、もとにスナップ・バックするアニメーションが行われます。

共通デスクトップ環境 Motif には、Motif 1.2 スタイル・ガイドを大幅に拡張した GUI スタイル・ガイドと確認チェックリストが入っています。追加事項は入力モデル、ウィンドウ管理、GUI 設計原理に関係するものです。

共通デスクトップ環境 **Motif** ウィジェット

共通デスクトップ環境 Motif は、Motif 1.2.3 では使用できない 2 種類のウィジェットを提供します。

- MenuButton - raw column ウィジットに入れる必要のないメニューのウィジェット

  これは Microsoft Windows や OPEN LOOK 環境からアプリケーションをポートするのに役立ちます。

- 端末エミュレータ・ウィジェット

  コマンド行ユーザ・インタフェースを GUI に最適な方法で組み合わせるよう設計されたアプリケーションに便利です。

- エディタ・ウィジェット

  Motif Text ウィジェットよりも高機能を備えたプレーン・テキスト・エディタを埋め込むのに使用できます。

- ヘルプ・ウィジェット

  アプリケーション・ヘルプ・ボリュームとのナビゲーションと対話を処理します。

ヘルプは、Standard Generalized Markup Language (SGML) 形式のファイルをヘルプタグでコンパイルした、セマンティック記述言語 (SDL) ファイル形式のアプリケーションで提供されます。ヘルプ・システムの特徴は、テキストとグラフィックス、ハイパーリンク、テキストの動的フォーマット変更、構造化ナビゲーション機能を統合したことです。

GUI シェル

共通デスクトップ環境には、デスクトップ Korn シェルがあります。これは Motif ツールキットの C プログラミングの代わりのインタプリンタ・スクリプト言語です。デスクトップ Korn シェルには、選択された頻繁に使用される共通デスクトップ環境 API、Xt API、Xlib API があります。環境の機能を十分活用するには、コンパイル言語でアクセスしてください。しかし、ドラッグ&ドロップ、セッション管理、ToolTalk メッセージなどのデスクトップ統合アクティビティを利用するデスクトップ Korn シェルスクリプトを書くことができます。

シェル・プログラミングに習熟している場合は、次のような理由により、簡単なプログラミング・タスクにデスクトップ Korn シェルを使用した方がいいこともあります。

- シェル・コマンドと GUI コントロールを簡単に組み合わせることができるので、システム管理型アプリケーションによく適応する。
- シェル環境は文字ベースの対話を普通に処理するので、文字ベースのアプリケーションに GUI 制御プログラムを加えやすい。
- 命令セットに依存しないプログラムを、いろいろなホストへ配信するのに適した方法である。たとえば、共通デスクトップ環境のメール・プログラムを使用し

て、受信者がダブルクリックするだけで実行するスクリプトをメールに添付する。

GUI 構築

共通デスクトップ環境アプリケーションを生成するのに最も簡単で、たぶん最も早い方法は、Motif ツールキット・プログラミングをなるべく使わないことです。共通デスクトップ環境のアプリケーション・ビルダ (AppBuilder) を使用し、アプリケーションの GUI コントロール部分を構築してください。AppBuilder はデフォルト・ウィジェット動作をアクセスしやすくすることを目的としています。これは、多くのウィジェットで使用可能な難解なリソースをほとんど隠してしまうことにより実現します。また、ドラッグ&ドロップ、セッション管理、ToolTalk メッセージなどのデスクトップ統合インフラストラクチャをアプリケーションに組み込みやすくします。

ヒント - アプリケーションビルダはよい教材になります。Sun Microsystems Advanced Developer Tools から、全機能 Motif GUI ビルダである XDesigner を購入することができます。

AppBuilder は、ビルダ・インタフェース言語 (BIL) ファイルに、ユーザ・インタフェースの状態を保管します。コード・ジェネレータは BIL ファイルを取り込んで、Motif ツールキット・コードを作成します。AppBuilder はユーザ・インタフェース言語 (UIL) ファイルを生成することもできます。

アプリケーションのユーザ・インタフェースを変更すると、AppBuilder はカスタム・コードを生成したコードにマージします。生成されたコードは、アプリケーションの GUI 状態を保持するために AppBuilder を使用していない場合でも、良いソースコード例になります。

さらに、プログラマ以外の人も AppBuilder を使用してアプリケーション GUI プロトタイプを作成できます。プロトタイプは、開発の製造過程でプログラマに渡すことができます。

統合テクノロジ

今まで、共通デスクトップ環境のテクノロジは、GUI を画面に表示させることに直接関係していると述べてきました。この節で説明する統合テクノロジは、GUI プロバイダではなく基本インフラストラクチャです。

プロセスの実行

ネットワーク使用環境を提供するため、リモート・ホストで実行しているアプリケーションの起動、管理、結果収集を行うサブ・プロセス・コントロール (SPC) 機能を、共通デスクトップ環境は提供しています。リモート・ホストは、ソケットベースのコントロール機能のリモート側として供給される SPC デーモンをインストールします。このコントロール機能はリモート・プロセスが親プロセスに対するローカルな子であるような錯覚を起こさせます。親プロセスを管理するユーザの認証は、ユーザのホーム・ディレクトリに `setuid` ファイルを書く親プロセスの権限と、結果を読む子プロセスの権限に基づいています。

SPC API および関連するコントロール・プログラムは、共通デスクトップ環境専用です。アクションは、リモートで実行するアプリケーションに対する公開している API を表します。

アプリケーション・メッセージ機構

ToolTalk メッセージ・サービスは共通デスクトップ環境用のアプリケーション・メッセージ機能です。アプリケーション・メッセージ機構は、シングル・ユーザのために動作している複数のアプリケーション間のコントロールおよび協調を指示します。ToolTalk セッション・デーモンは、コントロール範囲が通常は X サーバのコントロール範囲と対応しているローカル・メッセージ・ルーティング・プロセスです。つまり、セッション内のクライアントが要求を出すと、ToolTalk セッション・マネージャが要求を処理できるクライアントをセッション内から見つけるか起動し、ToolTalk セッション・デーモンが、要求を完了するまで追跡します。

デスクトップはメッセージ・セットという 2 つの標準 ToolTalk プロトコルを提供します。メッセージ・セットには、プロセスの送信側と処理側とで交換できる多くのメッセージが入っています。これらのメッセージは、関連する要求や通知を記述しているものによってグループ化されています。送信側および受信側は、同じプロセ

スまたは異なるホスト上に存在できます。メッセージ・セットには、ロー・レベル・メッセージングの詳細に関係なく、プロトコルのセマンティクスを集められる関連のユーティリティ機能があります。一部のメッセージ・セット機能には、デフォルト動作をほとんど何もせずに区別できるようにするものもあります。

デスクトップ・メッセージ・セット

このセットは 3 つの領域に分類されます。 1 番目はウィンドウ動作に関するメッセージです。2 番目はファイル・アクセスと短期ファイルのライフ・サイクル・コントロールに関連するメッセージです。3 番目は拡張言語を備えているが、ライブラリのサポートを一般的に保証できないアプリケーションに特有のメッセージです。

メディア・メッセージ・セット

このセットは、アプリケーションを任意のメディア用コンテナ、またはそのようなコンテナから駆動されるメディア・プレイヤ／エディタにします。メディア・メッセージ・インタフェースは、コンテナ・アプリケーション (メール・プログラムやファイル・マネージャなど) が、メディア型のフォーマットを認識していなくても、任意のメディア型のファイルまたはバッファの作成、表示、編集、印刷を可能にします。ToolTalk は、指定されたメディアの型とオペレーションに適するツールに対するコンテナの要求のルートを決定します。すでに実行されているインスタンスがその要求を処理するのに最適である場合は、そのインスタンスが選ばれます。

データ型の作成

共通デスクトップ環境は、デスクトップのオブジェクトに対する一定のユーザ・インタフェースを提供します。このため、デスクトップにはデータ型の作成と呼ばれる機能があり、一連の規準によってオブジェクトの型を決定します。規準には、名前のパターンや内容のパターンなど、ファイルベースおよびバッファベースのオブジェクトが共有する潜在的な属性が含まれています。他の規準はファイルに関するもので、パス名のパターンとファイル・アクセス権を含んでいます。各デスクトップ型に関連するのは、アイコン名、名前のテンプレート、ユーザに提示するアクションのリスト、他の型スペース (MIME 型など) に相当する型の名前、この型のテキスト記述などの、拡張可能な一連の属性です。アクションおよびデータ型データベースはデータ規準とデータ属性を格納します。

一連のデスクトップ型定義は、共通デスクトップ環境が定義し、プラットフォーム・ベンダが供給します。アプリケーションのインストール時に、専用データ

型およびパブリックなデータ型の両方について、アプリケーションのデータベースを増やしてください。

共通デスクトップ環境ライブラリ API によって、アクションおよびデータ型から情報が抽出されます。データ型作成 API は、オブジェクトのデスクトップ型を決定するためのデータベース型規準に、オブジェクトの属性を一致させます。そのアルゴリズムは、重複を解決するため一連の優先規則を使用します。

共通デスクトップ環境型スペースは、X/Open 共通デスクトップ環境標準で定義され、アイコン表示やアクション指定などのデスクトップ指向アクティビティをサポートするために存在しています。MIME 型スペースは Internet Engineering Task Force で定義され、メール・メッセージ部分の交換を行うために存在しています。ToolTalk メディア型スペースは、データをハンドラに一致させるために存在し、X コンソーシアムで定義される X 選択ターゲット型のサブセットです。従って型の定義を完全に行うには、共通デスクトップ環境型、X 選択ターゲット型、MIME 型を定義してください。プライベートな共通デスクトップ環境型については、組織名に型名を追加してください。こうすると、型の割り当ての集中管理なしにネーム・スペースが区切られます。共通デスクトップ環境ではデスクトップを表す Dt という接頭辞を付ける必要があります。

メソッドの実行

共通デスクトップ環境の型は、デスクトップ・オブジェクトのクラスと見なすことができます。この類似性により、アクションをクラスのインスタンスで使用できる方法と見なすことができます。したがって、型属性リストのアクション属性は、型に使用できるオペレーションを記述しています。アクションおよびデータ型のデータベースにある 1 つのアクションは複数の部分から構成されていますが、その多くはオプションです。次のような部分があります。

- オペレーション起動方法の記述。ToolTalk による方法、SPC 機能へ渡す実行文字列による方法、端末エミュレータから行う方法などがあります。
- アクションに関連する引き数の型についての記述。受け入れられるデスクトップ・オブジェクト (ファイルおよびバッファ) の型は、アクションおよびデータ型のデータベースで定義されます。アクションはデータ型によってさまざまです。たとえば [開く] アクションはテキスト・ファイルが引き数の場合はテキスト・エディタを、グラフィック・ファイルが引き数の場合はグラフィック・エディタを起動します。
- アクションに関連する引き数があれば、引き数の数の記述

- オペレーションを実行する場所のオプション指定。ローカル・マシン、特定のリモート・マシン、実行可能ファイルのあるマシンなどです。さらに、これらの場所はリストされるので、ホストが使用できない場合、リストにある次のホストで実行されます。これにより、リモート・ホストが使用不可能になった場合でも、アプリケーションを起動できる可能性を高められるような冗長性が提供されます。したがって、組み込み ToolTalk 機能または SPC 機能を直接使用することによって、アクションはネットワーク分散ガイダンスを提供できます。

- アクションの GUI と対話するときにユーザが見るオプションのラベル、ヘルプ文字列、アイコン。アプリケーションがアクションをユーザに示すときのヒントとして役立ちます。フロントパネル構成ファイルが代替アイコンを提供する場合は、フロントパネルがこのアイコンを無視し、これらのヒントを無視します。

ユーザが使用できるアクションの集まりは、システムが型データベース情報を集めるのと同時に組み立てられます。実際、関連するアクションと型情報は、通常は同じファイルに同時に存在します。デスクトップ定義ファイル、システム管理者定義 (ホスト固有) ファイル、ユーザ定義ファイルはこの順序で単一の (アクションおよびデータ型) データベースに組み込まれ、後の定義が優先されます。この検索パス優先順位と検索の順序は、ヘルプ・ボリュームおよびアイコン・ファイル検索用などデスクトップで一般的に使用されます。

アクションおよびデータ型のデータベースとファイル・マネージャは、アクション・ファイルを使用して、表示、起動、移動、コピーなどを行えるファイル・システム・オブジェクトとしてアクションをインスタンス化します。データベースにはアクションの実装に対するリファレンスが含まれます (「`net_app_svr` 上の `/usr/bin/app` を実行する」など)。ただし、ユーザが直接処理できるオブジェクトとしてアクションを表示することが必要です。これはオブジェクト名を使用して実行されます。オブジェクト名は、アクションを検索する任意のオブジェクト・マネージャに対するアクションとして識別されます。したがって、`Dtstyle` という実行可能ファイルと Dtstyle というアクションがある場合、ファイル・マネージャはそのファイルを内容に関わらず Dtstyle アクションが参照するものとして解釈します。また、ファイル・マネージャはアクションのラベルを、ユーザがファイルを見るための名前として使用します。アクション・ラベルはローカライズできますが、アクション名はローカライズできないプログラムのエンティティです。

アクションおよびデータ型のデータベースへのポインタとしてファイルを使用する場合の利点は、ユーザが何もしなくても基本の実装を展開できることです。しかし、あるユーザのアクションおよびデータ型のデータベースが、他のユーザのアクションとデータ型に一致しないこともあります。したがって、ユーザはアクションのリファレンスを、たとえばメール・メッセージのアタッチメントなどとして交換

したり、そのアクションに対して比較できる定義を持つユーザが他にいるかどうかを予期することができません。この問題を解決するには、デスクトップ Korn シェル・スクリプトを交換するのが最善の方法です。

アクションは継承したコマンド行アプリケーションと ToolTalk アプリケーションを、デスクトップ・オブジェクト上の多様な分散オペレーションとしてデスクトップに統合するので便利です。

パートII　開発環境の概要

第 **2** 章

開発環境の要件

この章では、共通デスクトップ環境のアプリケーション・プログラム・インタフェース (API) を使用する前に知っておくべき一般的な情報について説明します。

アプリケーションをデスクトップに統合する前に、デスクトップの動作について基本的な理解をしておくことが必要です。共通デスクトップ環境をプラットフォームにインストールし、その機能を把握してください。デスクトップの概要については、『*Solaris* 共通デスクトップ環境 ユーザーズ・ガイド』または『デスクトップの紹介』のオンライン・ヘルプ・ボリュームを参照してください。

23ページの「共通デスクトップ環境の特性」

25ページの「根底となる基礎事項」

26ページの「既存アプリケーションの実行」

27ページの「ライブラリおよびヘッダ・ファイル」

27ページの「デモ・プログラム」

27ページの「マニュアル・ページ」

28ページの「開発環境ディレクトリ構造」

共通デスクトップ環境の特性

共通デスクトップ環境には、UNIX ユーザに対して、生産性の高い快適なデスクトップ環境を提供します。アプリケーションを開発するときは、共通デスクトップ環境がユーザに提供する環境を常に念頭に置いてください。次のような特性を備え

たアプリケーションを開発すれば、高機能で、一貫性のある、予期可能な共通デスクトップ環境の一部にできます。

- UNIX の複雑さを隠します。

  共通デスクトップ環境は主な顧客としてエンド・ユーザをターゲットにしているので、できるだけ UNIX を隠したアプリケーションを提供することは、優れた製品の重要な要因です。

- 共通のルック&フィールを提供します。

  共通デスクトップ環境の優れたアプリケーションは、デスクトップ上の他のアプリケーションと同じようなルック&フィールを実現します。スタイルその他のガイドライン (共通デスクトップ環境標準フォント名など) に従うと、アプリケーションに共通デスクトップ環境 Motif のルック&フィールが備わります。

- アプリケーションを使いやすくします。

  読みやすいデフォルト・フォント・サイズと、マウス指向アクション用のキーボード・アクセラレータを提供します。デスクトップ・オンライン・ヘルプ・コンポーネントを使用して、完全なヘルプシステムをアプリケーションに統合してください。基本的なコンピュータ対話形式は、プラットフォーム全体で可能な限り一貫させなければなりません。

- デスクトップ統合サービスを利用します。

  共通デスクトップ環境には、アプリケーションをデスクトップにうまく統合できるようにするための、一連のデスクトップ統合サービスがあります。アプリケーションがローカルなコンピュータで実行されているのかネットワークの他の場所で実行されているのか、(もしあれば) 実行されているアプリケーションを書くのにどのツールキットが使用されているのかを、ユーザは知る必要がないので便利です。アプリケーションをデスクトップから起動し、他の共通デスクトップ環境のアプリケーションと通信できるようにする機能をアプリケーションに与えてください。オンライン・ヘルプシステムを使用して、ユーザに簡易情報を提供してください。ドラッグ&ドロップを使用して、ユーザがシステムの使用方法を予期できるようにしてください。

- 個別の、および文化的な相違点を設計します。

  共通デスクトップ環境の規則とポリシーに従うと、アプリケーションは次の項目に関して、スムーズで一貫性があり、適切なカスタマイズを自然に提供できます。

  - フォント
  - カラー

- キーボードとマウスの割り当て
- ロケール固有の構成ファイル

ロケール固有の構成ファイルの詳細は、『*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』の第 18 章『ローカライズされたデスクトップ・セッションの構成』を参照してください。

根底となる基礎事項

デスクトップ API を使用するアプリケーションをコンパイルするには、次のものが必要です。

- 共通デスクトップ環境のヘッダ・ファイルおよびライブラリ
- X11R5 以降のヘッダ・ファイルおよびライブラリ
- ANSI C コンパイラ、または C++ アプリケーションを開発する場合は、C++ コンパイラのバージョン 2.0 以降のもの

コンパイルしたバイナリ・ファイルは、一致するライブラリをインストールした環境で実行してください。実行時のライブラリは `/usr/dt/lib` にあり、表 2–2 にリストされているディレクトリを格納しています。

共通デスクトップ環境ベースのアプリケーションを正しく実行するには、次の実行時のディレクトリのファイルをインストールした環境が必要です。

- `/etc/dt/*`
- `/var/dt/*`
- `/usr/dt` の下のサブディレクトリ
  - `app-defaults`
  - `appconfig`
  - `bin`
  - `dthelp`
  - `lib`

アプリケーションは、X/Open XCDE 標準で明示的に言及されている実行時のファイルだけに依存しなければなりません。`/usr/dt/appconfig/icons` など標準で言及していないファイルに依存してはなりません。このような依存性をアプ

リケーションに構築した場合、標準にないファイルが失われたり変更されるような環境で実行できるようにアプリケーションを強化しなければなりません。

共通デスクトップ環境アプリケーションを実行するのに最低限必要な実行環境のリストについては、`dtfilsys(5)` のマニュアル・ページを参照してください。

実行環境には、Motif 1.2.3 のバグを修正して機能強化した共通デスクトップ環境 Motif が含まれます (機能強化の詳細は、付録 Aを参照してください) 。共通デスクトップ環境の実行環境の詳細は、実行環境用のマニュアル・セットを参照してください。

既存アプリケーションの実行

どの共通デスクトップ環境ライブラリともリンクされていない、既存の X Window System ベースのアプリケーションは、デスクトップ・ウィンドウ・マネージャ (`dtwm`) のもとで、Motif ウィンドウ・マネージャで動作するのと同様な方法で動作します。共通デスクトップ環境との相互運用性のレベルのガイドラインを説明します。

表 2–1 既存 X ベースのアプリケーションと共通デスクトップ環境との相互運用性

既存 X ベースアプリケーションの特性	デスクトップとの相互運用ステータス
Motif 1.2 (以降) のドラッグ&ドロップ (登録済みプロトコルを使用)	有
Motif 1.2 (以降) のドラッグ&ドロップ (動的プロトコルを使用)	無[1]
カット&ペースト (すべてのアプリケーション)	有
OPEN LOOK ドラッグ&ドロップ	有 (複数の項目のドラッグ&ドロップ以外)[2]
ToolTalk メディア交換プロトコルおよびデスクトップ・プロトコル	有

1. これは Motif 1.2 ドラッグ&ドロップ動的プロトコル相互運用における問題点で、共通デスクトップ環境に原因があるわけではありません。

2. これはドロップ・サイト・データベース・マネージャ (dsdm) のプロトコル解釈機能によって行われます。

既存の Motif アプリケーションを、共通デスクトップ環境 Motif 共有ライブラリで再コンパイルおよび再リンクする場合は、アプリケーションは Motif 1.2 に対応していなければなりません。

ライブラリおよびヘッダ・ファイル

ベンダ固有の場所にある X11R5 ヘッダ・ファイルおよびライブラリを考慮して共通デスクトップ環境アプリケーションをコンパイルしてください。表 2–2 は、/usr/dt のサブディレクトリにあるすべての開発環境ライブラリおよびヘッダ・ファイルの場所をリストしています。

デモ・プログラム

/usr/dt/examples サブディレクトリには、テンプレート・アプリケーションとしての開発環境コンポーネント・デモのソースコードが入っています。表 2–2 ですべてのデモのサブディレクトリをリストしています。

それぞれのデモのサブディレクトリには、1 つ以上のデモ・プログラムが、プログラムの makefile と共に入っています。デモを説明する README ファイルも含まれています。

ソースが template サブディレクトリにあるデモは、単純な描画プログラムです。これは、デスクトップに統合された共通デスクトップ環境アプリケーションの基本構造を図示したものです。国際化されており、別々のサブディレクトリにはローカライズされたコンポーネントがすべて入っています。アプリケーションでは、この描画プログラムをテンプレートとして使用できます。

マニュアル・ページ

共通デスクトップ環境のマニュアル・ページは /usr/dt/man にあります。man コマンドまたはデスクトップのマニュアル・ページ・ビューアを使用してマニュア

ル・ページを表示させるには、環境変数 MANPATH に /usr/dt/man を追加してください。たとえば ~/.dtprofile ファイルでは、次のように設定します。

```
MANPATH=$MANPATH:/usr/dt/man
```

/usr/dt/man サブディレクトリと内容の一覧のリストは、表 2–2 を参照してください。man コマンドの詳細は、man(1) のマニュアル・ページを参照してください。

開発環境ディレクトリ構造

表 2–2 は開発環境ディレクトリ構造のトップレベル・ディレクトリと、そのサブディレクトリをリストしています (トップレベル・ディレクトリ名にはすべて前に /usr/dt が付いています)。

表 2–2 /usr/dt の開発環境ディレクトリ

ディレクトリ	サブディレクトリ	内容
examples		開発環境コンポーネント・デモ・プログラムのソースコードを含むサブディレクトリ
		README ファイル
	dtaction	アクション起動 API デモ
	dtbuilder	アプリケーション・ビルダの例
	dtcalendar	カレンダ API デモ
	dtdnd	ドラッグ&ドロップ API デモ
	dtdts	データ型作成 API デモ
	dthelp	ヘルプ API デモ
	dtksh	dtksh スクリプトの例
	dtsession	セッション・マネージャ API デモ

表 2–2　/usr/dt の開発環境ディレクトリ　続く

ディレクトリ	サブディレクトリ	内容
	dtterm	端末エミュレータ・ウィジェット API デモ
	dtwidget	共通デスクトップ環境 Motif ウィジェット・デモ
	dtwsm	ワークスペース・マネージャ API デモ
	template	テンプレート共通デスクトップ環境アプリケーション
	motif	Motif 1.2 API デモ
	motif/clipboard	XmClipboard API デモ
	motif/dogs	ウィジェット・バイナリ互換機能デモ
	motif/draganddrop	Motif 1.2 ドラッグ&ドロップ API デモ
	motif/periodic	Motif ウィジェット・デモ
	tt	ToolTalk メッセージ・サービス・デモ
include		開発環境ライブラリのヘッダ・ファイル
	csa	カレンダのヘッダ・ファイル
	Dt	DtSvc、DtWidget、DtHelp、DtTerm のヘッダ・ファイル
	Mrm	Motif 1.2 リソース・マネージャのヘッダ・ファイル
	Tt	ToolTalk メッセージ・サービスのヘッダ・ファイル
	Xm	Motif 1.2 ツールキットのヘッダ・ファイル

表 2–2 /usr/dt の開発環境ディレクトリ 続く

ディレクトリ	サブディレクトリ	内容
	uil	Motif 1.2 呼び出し可能 UIL コンパイラのヘッダ・ファイル
lib		libcsa (カレンダ)、 libDtHelp (ヘルプ・システム)、 libDtSvc (アクション、データ型作成、ドラッグ&ドロップ、セッション・マネージャ、ワークスペース・マネージャ)、 libDtTerm (端末エミュレータ)、 libDtWidget (共通デスクトップ環境 Motif コントロールおよびエディタ・ウィジェット)、 libMrm (Motif 1.2 リソース・マネージャ)、 libUil (Motif 1.2 呼び出し可能 UIL コンパイラ)、 libXm (Motif 1.2 ツールキット)、 libtt (ToolTalk) sparcv9 (lib サブディレクトリ中の 64 ビットバージョン全ライブラリを格納するサブディレクトリ)
man		開発環境のマニュアル・ページ
	man1	クライアントおよびユーティリティ関連のマニュアル・ページ
	man3	API 関連のマニュアル・ページ
	man4	データ・フォーマット
	man5	ヘッダ・ファイルおよびアクション関連のマニュアル・ページ

64 ビットライブラリ

lib サブディレクトリの下にリストされたライブラリの 64 ビットバージョンが、sparcv9 サブディレクトリにあります（表 2–2 参照)。これらのライブラリで、CDE 用の 64 ビットアプリケーションを開発することができます。

第 **3** 章

アプリケーションの開発

この章では、命名規則およびその他のガイドラインなど、共通デスクトップ環境アプリケーションの開発に固有の情報を示します。アプリケーションへ組み込んでデスクトップとの統合を促進するデスクトップ機能を決定するためのガイドラインである、統合のレベルを概説します。また、共通デスクトップ環境アプリケーションの開発を容易にするツールであるアプリケーション・ビルダの概要も説明します。

33ページの「デスクトップ統合のレベル」

34ページの「デスクトップ命名規則」

36ページの「公開しているインタフェースと公開していないインタフェース」

36ページの「共通デスクトップ環境データベースのガイドライン」

36ページの「アプリケーションの初期化と libDtSvc」

37ページの「アプリケーション・ビルダ」

デスクトップ統合のレベル

ユーザは共通デスクトップ環境のシェル・コマンド行から、任意の X11 ベースのアプリケーションを実行できます。ただしアプリケーションをデスクトップへ統合する場合は、従わなければならないガイドラインがあります。アプリケーションの設計または既存アプリケーションの移植において最大限の適応性を与えるために、共通デスクトップ環境は 3 つの統合レベルを定義しています。

- 基本的な統合方法

アプリケーションをデスクトップから起動できます。基本的な統合を行うときはアプリケーションのコードを変更する必要はありません。詳細は、第 5 章を参照してください。

- 推奨する統合方法

  アプリケーションとデスクトップとの一貫性のレベルを向上できます。詳細は、第 6 章を参照してください。

- オプションの統合方法

  特殊な作業を実行するためにデスクトップが提供するサービスが利用できます。詳細は、第 7 章を参照してください。

統合の全レベルの詳細は、『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』を参照してください。基本的な統合については『*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』でも説明しています。

デスクトップ命名規則

共通デスクトップ環境は、X および Motif で使用されているのと同様の命名規則を使用します。デスクトップ・クライアント、デスクトップ・ライブラリ、その他のデスクトップ・コンポーネントは、外部から見ることができる名前に共通の `dt`、`Dt`、`DT` などの接頭辞を共有します。個人用のデスクトップ構造、関数、定義 (開発者が使用するのではなく、共通デスクトップ環境コードに入っている) には、`_dt`、`_Dt`、`_DT` という接頭辞が付いています。表 3–1 では、デスクトップの命名規則をリストします。

表 3–1 デスクトップ命名規則

名前	接頭辞	例
デスクトップ・クライアントおよびユーティリティ	`dt`	`dthelpview`
リソース名およびクラス	`Dt`	`DtNhelpType, DtCHelpType`
ライブラリ名	`Dt`	`libDtHelp`
ヘッダ・ファイル	`Dt`	`#include <Dt/Help.h>`

表 3–1 デスクトップ命名規則 続く

名前	接頭辞	例
公開している関数名	Dt	DtCreateHelpDialog
公開しているデータ構造名	Dt	DtHelpDialogCallbackStruct
定数名	Dt	DtHELP_NEW_WINDOW
環境変数	DT	DTHELPSEARCHPATH
公開していないデスクトップ・シンボル (構造、関数、定義)	_dt, _Dt, _DT	_DtHelpFunction, _DtHELP_DEFINE

表 3–2 では、上記の命名規則の例外をリストします。

表 3–2 デスクトップ命名規則の例外

名前	接頭辞	例
共通デスクトップ環境 Motif	Xm	XmCreateLabel
dtksh 簡易関数	Dtksh	DtkshAddButtons
ToolTalk メッセージ・サービス	tt（関数用）	tt_open
	Tt（typedef用）	Tt_message
	TT（定数用）	TT_NOTICE
X11R5	X, Xt	XOpenDisplay, XtCreateWidget

注意 - アプリケーションのコードで新しいシンボルを定義するときは、接頭辞 dt、Dt、DT、_dt、_Dt、_DT、Xm、tt、Tt、TT、X、Xt を使用しないでください。使用した場合は、共通デスクトップ環境、ToolTalk、X11R5、Motif コードにすでに定義されているものか、定義される予定のものと重複する可能性があります。

公開しているインタフェースと公開していないインタフェース

マニュアル・ページまたは共通デスクトップ環境マニュアル・セットで、共通デスクトップ環境インタフェースについて言及されている場合、特に断り書きがなければそのインタフェースは公開されていると見なすことができます。インタフェースに対応するヘッダ・ファイルがあるという事実だけでは、そのインタフェースが公開されているものであるとは言いきれません。マニュアルまたはその他のドキュメントに記載されていないインタフェースは共通デスクトップ環境に対して内部専用であり、予告なしに変更されることがあります。

共通デスクトップ環境データベースのガイドライン

`/usr/dt/man/man4` ディレクトリにあるマニュアル・ページに、アクションやデータ型などに使用するデスクトップ・データベースの形式があります。

データベースの詳細は、『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』を参照してください。

アプリケーションの初期化と libDtSvc

アプリケーションが任意の `libDtSvc` API (アクション、データ型作成、ドラッグ＆ドロップ、セッション・マネージャ、ワークスペース・マネージャのための) を使用している場合、`DtInitialize()` または `DtAppInitialize()` を呼び出して、まず `libDtSvc` ライブラリを初期化しなければなりません。詳細は、`DtInitialize(3)` または `DtAppInitialize(3)` のマニュアル・ページを参照してください。

アプリケーション・ビルダ

アプリケーション・ビルダ (AppBuilder) は、デスクトップのアプリケーション・プログラム・インタフェース (API) を呼び出すためにコードを書かなくても、共通デスクトップ環境アプリケーションのグラフィカル・ユーザ・インタフェース (GUI) を簡単に作成できるようにするツールです。このツールは Motif ツールキットを単純なオブジェクト・パレットとオブジェクト属性シートに要約します。AppBuilder を使用して、単純な GUI ベース・プログラムから複雑で統合されたシステムまであらゆる範囲のアプリケーションを構築できます。ユーザ・インタフェース言語 (UIL) ファイルのインポートおよびエクスポートをサポートし、他の Motif ベース・ツールおよびプロダクトの中にアプリケーションを移植できるようにします。

AppBuilder は、次のような場合に使用するのに適しています。

- Motif プログラマのエキスパートではない
- 共通デスクトップ環境 Motif ウィジェットに精通していない
- (ドラッグ&ドロップ、ToolTalk メッセージ、セッション、ヘルプ、国際化対応などの) デスクトップ・サービスに精通していない
- アプリケーション・ユーザ・インタフェースを早く構築し、簡単に変更できるようにしたい
- 1 つのアプリケーションを他の人と共同で構築している

実際に上記のいずれかに該当しない場合でも、アプリケーションを開発する上で AppBuilder の使用が適切であり役に立つと思われることはよくあります。

AppBuilder を使用すると、次のようなことが実行できます。

- 共通デスクトップ環境 Motif ツールキットのオブジェクトの部品を集めて構成することで、ユーザ・インタフェースをアプリケーション用にレイアウトする
- オブジェクト間の接続を定義し、アプリケーション GUI 動作を作成して、接続をテストできるテスト・モードを使用する
- デスクトップ・サービス機能の一部をアプリケーションへ追加する
- AppBuilder を使用してあらかじめ作成したアプリケーションを編集する
- 自動的に生成されたコードを手動で生成したコードにマージする
- C 言語ソースコードと関連するプロジェクト・ファイル (メッセージ・カタログなど) をアプリケーション用に生成する

AppBuilder からアプリケーションをコンパイルおよび起動できます。AppBuilder を終了して再起動しなくても、共通環境から構築、実行、デバッグのサイクルを実行できます。

ヒント - アプリケーションビルダはよい教材になります。Sun Microsystems Advanced Developer Tools から、全機能 Motif GUI ビルダである XDesigner を購入することができます。

デモ・プログラム

AppBuilder のプログラム例は `/usr/dt/examples/dtbuilder` にあります。このプログラムの詳細は、`README` ファイルを参照してください。

関連マニュアル

アプリケーション・ビルダの詳細は、適切なマニュアル・ページ、AppBuilder のヘルプ・ボリューム、および『アプリケーション・ビルダ・ユーザーズ・ガイド』を参照してください。

第 **4** 章

移植性と保守性

この章では、移植性の高いアプリケーションを書くために、およびアプリケーションが共通デスクトップ環境の今後のリリースと互換性があることを保証するために使用できる情報を示します。

39ページの「移植性の問題」

42ページの「共通デスクトップ環境の Motif ウィジェット・バイナリの互換性のガイドライン」

移植性の問題

この節では、共通デスクトップ環境をサポートする、異なるプラットフォーム間でのアプリケーションの移植性に関する問題を説明します。

標準

アプリケーションを共通デスクトップ環境に準拠させるには、Motif 1.2、ANSI-C、X11R5 標準に従わなければなりません。C++ でアプリケーションを開発する場合は、C++ バージョン 2.0 以降を使用してください。共通デスクトップ環境のアプリケーションを作成するとき、POSIX など他の標準に固執することはありません。デスクトップのアプリケーション・プログラム・インタフェース (API) を使用するアプリケーションは、他の共通デスクトップ環境プラットフォームへ移植できます。しかし、POSIX を使用するとソフトウェアの移植性は拡張されます。

ここで言う POSIX 標準は、IEEE Std 1003.1-1990 、『*IEEE Standard for Information Technology－Portable Operating System Interface (POSIX)－Part 1: System Application Program Interface (API) [C Language], ISBN 1-55937-061-0*』を指します。

また Motif 1.2 標準は、IEEE Std 1295、『*Standard for Information Technology－X Window System Graphical User Interface－Modular Toolkit Environment*』を指します。

上記 2 点の注文方法については、xiページの「関連文書」を参照してください。

Makefile

X11R5 など、共通デスクトップ環境が依存しているライブラリは、プラットフォームが異なれば別の場所にインストールされていることがよくあります。これを解決するには、プラットフォーム固有のリファレンスを取り込むか、各プラットフォーム別の makefile を書いてください。

また、`make` プログラムの機能はプラットフォームによって異なります。アプリケーションに対して makefile を 1 つだけしか書かないのであれば、プログラムの移動先であるプラットフォームが使用している共通の `make` 機能を使用してください。プラットフォーム固有の `make` 機能は使用しないでください。

デスクトップと統合するのに、共通デスクトップ環境では定義された定数 (`–D` パラメータなど) を追加する必要はありません。POSIX などの標準に従う場合は、標準固有のフラグを追加してコンパイルする必要があります。特殊なコンパイラ要件があるかどうかについては、標準のドキュメントを参照してください。

`/usr/dt/examples` の各サブディレクトリには、異なるプラットフォーム用の makefile の例が入っています。これらの makefile はシステムの相違点を考慮しています。特に、一般的な makefile の例については、`/usr/dt/examples/dtdts` ディレクトリを参照してください。

コンパイル・オプション

アプリケーションがデスクトップの `include` ファイルを検索できるようにするには、次の行を各 makefile のコンパイル行に追加してください。

```
-I/usr/dt/include
```

リンク・オプション

アプリケーションがデスクトップ・ライブラリを参照できるようにするには、次の行を各 makefile のリンク行に追加してください。

```
-L/usr/dt/lib -l<libname1> -l<libname2>...
```

libname1、*libname2* はアプリケーションが参照するライブラリ名です。デスクトップ・ライブラリ名は必要なだけ指定できます。次に例を示します。

```
-L/usr/dt/lib -lDtSvc -ltt -lXm
```

このように指定すると、アプリケーションはデスクトップ・サービス、ToolTalk メッセージ・システム、Motif 1.2 ライブラリを参照します。

ファイル命名規則

アプリケーション・ファイル名とアプリケーションが生成するファイル名は、14 文字以内にしてください。そうすれば、長いファイル名をサポートしていないプラットフォームに移植できるようになります。この制限を持つプラットフォームがいくつかあります。

エンド・ユーザが生成するファイル名にはこの制限は当てはまりません。

ディスプレイ・サポート

アプリケーションは次のようなディスプレイ・オプションと構成をサポートする必要があります。

- モノクロ・ディスプレイ (白黒アイコンを含む)
- カラー・ディスプレイ (カラー・アイコンを含む)
- VGA (640 × 480 - アプリケーション・ウィンドウまたはフォント・サイズが完全に画面に適合するようにサイズを調整する) のような小さい画面

カラー・アイコンを作成するためにアイコン・エディタを使用すると、より容易にアプリケーションが他のデスクトップ・アプリケーションとカラーを共有できます。これにより、疑似カラー・ディスプレイで実行するときにカラー・セルの浪費を防ぎます。

共通デスクトップ環境の **Motif** ウィジェット・バイナリの互換性のガイドライン

すでにサブクラス化したウィジェットのデータ構造体サイズに依存する標準 Xt API を使用して実装するウィジェット・サブクラスは、Motif または共通デスクトップ環境の新バージョンとの互換性がない可能性があります。Motif の新バージョンのスーパークラスに新規フィールドが追加されている可能性があるからです。たとえば Motif 2.0 の `XmManager` および `XmPrimitive` クラスに新規フィールドが追加されています。

サブクラスは、ウィジェット・インスタンスの開始アドレスに関連して指定されたインスタンス・フィールドに対するコンパイルされたリファレンスを持つので、非互換が起こります。スーパークラス・インスタンス構造体を拡張したウィジェットを持つ新しい Motif ライブラリをインストールした場合に、コンパイルされたリファレンスは間違ったメモリの場所を指します。

このような問題を避けるため、Motif にはリソースを定義する機能と、インスタンスおよびウィジェットの全構造体の代わりにウィジェットの部分構造体の先頭 (アドレス) に関連する制約構造体にあるすべてのフィールドを参照できるようにするウィジェット・フィールドへアクセスする機能があります (ウィジェットの全構造体は、スーパークラス部分構造体を含みます) 。ウィジェット・クラスを最初に初期化したとき、この機能はこれらの関連するリファレンスを実行時に解決します。そのため、現在リンクされている Motif ライブラリから読み取る、ウィジェットのスーパークラス・インスタンス構造体のサイズを計算に入れます。

注 - サブクラス化を実装する場合、アプリケーションと共通デスクトップ環境の今後のリリースとのバイナリ互換可能なものにするには、必ず Motif リファレンスの解決機能を使用してください。

この Motif 機能に関する詳細は、Motif 1.2 の `XmResolvePartOffsets(3x)` および `XmResolveAllPartOffsets(3x)` のマニュアル・ページと、『*OSF/Motif* プログラマーズ・リファレンス』を参照してください。ソースコード例は `/usr/dt/examples/motif/dogs` にあります。

第 5 章

基本的なアプリケーションの統合方法

基本的なアプリケーションの統合作業は、実行することを強く推奨します。これらの作業は、アプリケーションのソースコードの変更を要求しません。

43ページの「基本的な統合方法の特徴」

44ページの「基本的な統合方法情報の構成」

45ページの「基本的な統合方法の作業」

基本的な統合方法では、デスクトップ アプリケーション・プログラム・インタフェース (API)を拡張して使用することはありません。したがって、ドラッグ&ドロップ、セッション管理、ToolTalk メッセージ、アクションおよびデータ型のデータベースへのプログラムによるアクセスなど、他のデスクトップとの対話は提供しません。

基本的な統合方法の特徴

基本的なアプリケーションの統合方法には、エンド・ユーザ向けの次のような特徴があります。

- デスクトップ上にアプリケーションをグラフィカルに位置づけ、起動する方法

  アプリケーションはデスクトップの登録パッケージを提供し、インストレーション・スクリプトは自動的にアプリケーションを登録します。

  登録すると、アプリケーション・マネージャのトップレベルにアプリケーション・グループが作成されます。アプリケーション・グループにはアイコンがあり、ユーザがダブルクリックするとアプリケーションが起動されます。

- アプリケーションのデータ・ファイルを認識し、処理する能力

  アプリケーションはデータ・ファイルのデータ型を提供します。

  ユーザがデータ型を識別しやすいように、データ型の作成はファイル・マネージャ上で特有のアイコンを使用するようにデータ・ファイルを構成します。データ・ファイルには重要なデスクトップの動作も含まれます。次に例を 2 つ示します。

  - データ・ファイルをダブルクリックして、アプリケーションを起動できます。
  - デスクトップのプリンタ・ドロップ領域の上にデータ・ファイルをドロップすると、適切な印刷コマンドでファイルを印刷します。

- スタイル・マネージャを使用した、フォントとカラーの容易な選択

  アプリケーションは、インタフェース・フォントと、バックグラウンド、フォアグラウンド、シャドウのカラーを動的に変更します。

  アプリケーション固有のリソースが存在しなければ、デスクトップが一般的なインタフェース・フォントとカラー・リソースを定義します。

基本的な統合方法では、以下の機能をシステム管理者に提供します。

- 簡単なインストールと登録

  インストールすると、アプリケーションが自動的に登録されます。システム管理者が行わなければならない作業は少ないか、まったくありません。

- 簡単な運用中の管理

  デスクトップの構成ファイルはすべて 1 つの場所に集められます。さらに、たとえばシステム管理者がアプリケーションを更新または他のアプリケーション・サーバへ移動させたい場合など、簡単に登録解除できます。

基本的な統合方法情報の構成

基本的な統合方法で実行される作業のほとんどは、既存のアプリケーションをデスクトップへ統合するシステム管理者が実行するものです。したがって、基本的な統合方法のマニュアルはほとんど『*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』にあります。

『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』には、基本的な統合方法を説明している章があります。そこでは、『*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』に含まれている情報を指すものもあります。また、アプリケーション・プログラマに固有の追加情報もあります。

基本的な統合方法の作業

基本的な統合方法には次のような一般的な作業があります。

- フォントおよびカラーを設定する任意のアプリケーション・リソースの変更。これによりユーザがアプリケーションのフォントとカラーをスタイル・マネージャを使用して変更できます。

  『*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』にあるフォントおよびカラー・リソース変更に関する節を参照してください。

- アプリケーションのデータ・ファイルの印刷

  詳細は、『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』の基本的な統合方法に関する章を参照してください。

  印刷の型を統合するのに、アプリケーション・コードの変更が必要なものがあります。これらはオプションで、基本的な統合方法の作業に密接に関連しているため、『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』の基本的な統合方法に関する章で説明しています。

- アプリケーションの登録パッケージの作成

  次のマニュアルを参照してください。

  - 『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』の基本的な統合方法の章
  - 『*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』

- 登録パッケージ・ファイルをインストールし、登録プロシージャを実行するためのアプリケーションのインストール・スクリプトの変更

  次のマニュアルを参照してください。

  - 『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』の基本的な統合方法の章
  - 『*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』の `dtappintegrate` を使用したアプリケーションの登録に関する節

第 6 章

推奨する統合方法

共通デスクトップ環境には、コンポーネントおよび使用に関するガイドラインがあり、これに従うとデスクトップ上でアプリケーションを他のアプリケーションより高度なレベルで統合できます。この章では、アプリケーションとデスクトップとの一貫性のレベルを高めるために使用することを推奨するコンポーネントとガイドラインの概要を説明します。

- 48ページの「ヘルプ・システム」
- 49ページの「ToolTalk メッセージ・サービス」
- 53ページの「セッション・マネージャ」
- 54ページの「ドラッグ&ドロップ」
- 56ページの「国際化対応」
- 58ページの「標準フォント名」
- 61ページの「アプリケーションのエラー・メッセージの表示」
- 62ページの「ユーザ・カスタマイズの問題」

注 - コンポーネントの統合とこの節で説明するガイドラインの他に、第 5 章で概説する基本的な統合方法も参照してください。

推奨する統合方法の詳細は、『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』を参照してください。

ヘルプ・システム

共通デスクトップ環境のヘルプ・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・ヘルプを開発および表示するための完全なシステムです。これにより、設計者は豊富なグラフィックとテキスト・フォーマット、ハイパーリンクを備えたオンライン・ヘルプを書くことができ、アプリケーションからヘルプ・システムへアクセスできます。ヘルプ・システムはヘルプ機能をアプリケーションへ統合するためのプログラマのツールキットを提供します。

オンライン・ヘルプの作成とアプリケーションへの統合は、共同作業で行われます。ユーザのヘルプ要求にアプリケーションがどのように応答するかは、開発者が設計し実装します。設計者は、実際に表示されるヘルプ情報を構成し作成します。

ヘルプ・システムには次のものが含まれます。

設計者用

- 共通デスクトップ環境のヘルプタグ・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のテキスト・ファイルに使用するタグ・セット

- 共通デスクトップ環境のヘルプタグ・ソフトウェア

  ヘルプタグ・ファイルを実行時のヘルプ・ファイルに変換するためのソフトウェア・ツール・セット

- 共通デスクトップ環境のヘルプビュー・アプリケーション

  オンライン・ヘルプを表示するためのビューア・プログラム

設計者はヘルプ・タグ・セットを使用し、Structured Graphic Markup Language (SGML) タグ規則に従ってヘルプ・トピックを作成します。SGML マークアップは一次データ・フォーマットです。コンパイルされた実行時のフォーマットは SGML に準拠します。

ヘルプ・システムは、UNIX マニュアル・ページ、テキスト・ファイル、テキスト文字列などの SGML でないフォーマットもサポートします。

プログラマ用

- `DtHelp` プログラミング・ライブラリ

ヘルプ・ウィンドウを作成してアプリケーションと統合するための アプリケーション・プログラム・インタフェース (API)

- `DtHelp` ウィジェット

  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 (いずれもヘルプ・ライブラリの一部) を作成するための `DtHelpDialog` および `DtHelpQuickDialog` ウィジェット

ライブラリおよびヘッダ・ファイル

ヘルプ・ライブラリ `libDtHelp` は、Motif に基づくヘルプ・ダイアログの作成と管理をサポートします。`libDtHelp` ヘッダ・ファイルは次のとおりです。

- `Dt/Help.h`
- `Dt/HelpDialog.h`
- `Dt/HelpQuickD.h`

デモ・プログラム

`/usr/dt/examples/dthelp` にヘルプ・システム・デモがあります。デモの詳細は、 `README` ファイルを参照してください。

関連マニュアル

ヘルプ・システムの詳細は、関連するマニュアル・ページと『プログラマーズ・ガイド (ヘルプ・システム編)』を参照してください。

ToolTalk メッセージ・サービス

共通デスクトップ環境は、メッセージ・セットと呼ばれる 2 つの標準 ToolTalk プロトコルを定義します。メッセージ・セットは、送信側プロセスと処理側プロセスとで交換できるメッセージの集まりです。これらのメッセージは、関連する要求および通知を記述するものごとにグループ化されています。送信者および受信者は同じプロセスにあっても異なるホストにあってもかまいません。メッセージ・セットには、ローレベルの詳細に気をとられることなくプロトコルのセマンティクスだけに

集中できるようにする関連ユーティリティ機能があります。機能の中には、簡単にデフォルト動作に従えるようにするものもあります。

デスクトップ・メッセージ・セットには次の 3 つの領域があります。

- ウィンドウ動作
- ファイル・アクセスおよびファイルの短期ライフサイクル制御
- アプリケーション拡張言語

ウィンドウ動作の詳細は、50ページの「デスクトップの処理」および 51ページの「デスクトップの送信」の項を参照してください。ファイル・アクセスおよびファイルの短期ライフサイクル制御の詳細は、52ページの「デスクトップ・ファイル」の項を参照してください。Do_Command 要求の実装はアプリケーションの拡張言語に固有のもので、ToolTalk メッセージ・サービスではサポートしません。

メディア・メッセージ・セットにより、アプリケーションが、任意のメディアのコンテナまたはそのようなコンテナから起動できるメディア・プレイヤおよびエディタとなることができます。メディア・メッセージ・セットによって、コンテナ・アプリケーションは、該当するメディアの型のフォーマットを知らなくても任意のメディアのドキュメントを作成、表示、編集、印刷できます。ToolTalk メッセージ・サービスは、コンテナの要求を、指定されたメディアの型とオペレーション用のツールへ渡します。すでに実行中のツールのインスタンスがその要求を処理するのに最適であれば、そのインスタンスが要求されます。51ページの「メディアの送信」および51ページの「メディアの処理」を参照してください。

デスクトップ・メッセージ・セット

ToolTalk メッセージ・サービスは次のようなメッセージ・セットをサポートします。

デスクトップの処理

デスクトップ要求の処理は、メッセージング統合の中で最も基本的なものです。ToolTalk メッセージを送信するアプリケーションは、tt_message_send() または DtActionInvoke() のいずれを呼び出す場合も、デスクトップ要求を処理します。これによって、アプリケーションの現在のディレクトリ、アイコンの状態、$DISPLAY などを他のアプリケーションが設定または照会できます。詳細は、ttdt_open()、ttdt_session_join()、ttdt_session_quit()、ttdt_close() のマニュアル・ページを参照してください。

デスクトップの送信

アプリケーションを `ttsession` で起動し、ToolTalk 要求を処理するようにすると、このアプリケーションは要求送信者ではなく `ttsession` の子になります。アプリケーションは通常、送信者と同じ X の表示セッションで起動されますが、同じ X11 画面または同じ現在のディレクトリ・コンテキストにある必要はありません。アプリケーションがサーバ・プロセスとして実装された場合、すでに特定の画面または特定のディレクトリ・コンテキストに表示されています。

デスクトップ要求を使用すると、アプリケーションの操作は、デスクトップ以外でコマンド行の起動で継承される送信者の属性から継承できます。このようにデスクトップ・メッセージ・セットを使用し、ハンドラのロケール、現在の作業ディレクトリ、`$DISPLAY` をリセットしてください。これにより、入念にコード化された受信アプリケーションが送信者として同じ X11 画面に表示されます。要求ハンドラは要求送信者の現在のディレクトリとウィンドウのジオメトリを見つけることもできます。ウィンドウのジオメトリを知っていると、要求ハンドラのウィンドウが要求送信者のウィンドウをできる限り隠さないようにできます。詳細は、`ttdt_sender_imprint_on()` のマニュアル・ページを参照してください。

メディアの処理

ToolTalk メッセージ・サービスは、エディタが処理するメディアの型に対する標準メディア要求を、処理しやすくします。詳細は次のマニュアル・ページを参照してください。

`ttmedia_ptype_declare()`、`ttdt_message_accept()`、
`ttmedia_load_reply()`、`ttmedia_Deposit()`

メディアの送信

ToolTalk メッセージ・サービスは、コンテナのメディア要求送信と、ハンドラが返す一連のドキュメントの管理を容易にします。コンテナがメディア・ハンドラで 実行中の ToolTalk ダイアログを処理していない場合は、ToolTalk API を直接使用するのではなく、アクション API を使用してください。相当するアクション ([開く] と [印刷]) は、ToolTalk および ToolTalk 以外が検知するメディア・ハンドラと同等のハンドラをサポートする上位の概念を示します。詳細は、`ttmedia_load()` および `ttdt_subcontract_manage()` のマニュアル・ページを参照してください。ほとんどの場合、コンテナ・アプリケーションは `ttmedia_load()` ではなく `DtActionInvoke()` を使用してオブジェクトのオペレーションを実行するので注

意してください。アクションによって ToolTalk アプリケーションを起動する方法の詳細は、『*ToolTalk* メッセージの概要』を参照してください。

デスクトップ・ファイル

ToolTalk メッセージ・サービスは、ファイルに関するデスクトップのメッセージを送受信しやすくします。これらのメッセージにより、アプリケーションがファイルへのアクセスを調整できるようになります。詳細は次のマニュアル・ページを参照してください。

`ttdt_file_join()`、`ttdt_file_quit()`、`ttdt_file_event()`、`ttdt_Get_Modified()`、`ttdt_Save()`、`ttdt_Revert()`

ToolTalk メッセージ・サービスをすでに使用しているアプリケーションの例は、共通デスクトップ環境のアイコン・エディタ、メール・プログラム、テキスト・エディタ、カレンダなどです。共通デスクトップ環境の他の部分では、メッセージを送信するアクションを定義することにより、ToolTalk メッセージ・サービスを間接的に使用しています。

ライブラリおよびヘッダ・ファイル

ToolTalk メッセージ・ライブラリは `libtt` と呼ばれます。`libtt` ヘッダ・ファイルは次のとおりです。

- `Tt/tt_c.h`
- `Tt/tttk.h`

デモ・プログラム

ToolTalk メッセージ・サービスのデモは `/usr/dt/examples/tt` にあります。デモの詳細は `README` ファイルを参照してください。

関連マニュアル

ToolTalk メッセージ・サービスの詳細は、関連するマニュアル・ページと『*ToolTalk* メッセージの概要』を参照してください。

セッション・マネージャ

セッション・マネージャは ICCCM.1.1 WM_COMMAND および WM_SAVE_YOURSELF プロトコルをサポートし、次のことを許可します。

- アプリケーションがログアウト時の状態情報を保存する
- セッション・マネージャがログイン時にアプリケーションを再起動する

セッション・マネージャは API も提供し、アプリケーションがログアウト時およびログイン時の状態を保存および格納するのを補助します。

セッション・マネージャはログイン時のアプリケーション再起動に責任を持ちます。これを行うには、再起動に必要なコマンドおよびコマンド行オプションをアプリケーションがセッション・マネージャに通知しなければなりません。Xlib の XSetCommand() を使用して、アプリケーションのトップ・レベル・ウィンドウに WM_COMMAND 属性を設定してください。

セッション・マネージャがログアウト時などにセッションを保存する際に、アプリケーションは似たような状態での再開のために一部の状態情報を保存する必要があります。セッション・マネージャは、オプションでセッションが保存されていることをアプリケーションに通知します。このような通知が必要であることをアプリケーションはセッション・マネージャに知らせなければなりません。これは WM_SAVE_YOURSELF プロトコルをトップ・レベル・ウィンドウの WM_PROTOCOLS 属性に登録し、コールバック・プロシージャを設定して通知を処理します。これには XmAddWMProtocols() および XmAddWMProtocolsCallback() 関数を使用します。WM_SAVE_YOURSELF コールバックを処理しているときに何らかの方法でアプリケーションがユーザと対話すべきではありません (たとえば [別名保存] ダイアログ・ボックスは表示するべきではありません) 。このコールバックは WM_COMMAND 属性をトッププレベル・ウィンドウに設定して、セッション・マネージャに状態の保存が終了していることを通知しなければなりません。

アプリケーションが状態情報を保存できるようにするには、DtSessionSavePath() 関数を使用して、情報を保存するファイルの絶対パス名を獲得してください。セッションの復元時は、DtSessionRestorePath() 関数を使用して、アプリケーションが状態を復元するのに使用する状態ファイルの絶対パス名を獲得してください。

共通デスクトップ環境のワークスペース・マネージャは、アプリケーションのメイン・トップレベル・ウィンドウ (WM_COMMAND を含む) 属性を正しいワークスペース、ジオメトリ、アイコン状態に復元します。アプリケーションに複数のトップ・

レベル・ウィンドウがある場合、他の上位ウィンドウの状態の復元はアプリケーションが担当します。その他の情報については、75ページの「ワークスペース・マネージャ」を参照してください。

ライブラリおよびヘッダ・ファイル

デスクトップ・ライブラリ `libDtSvc` は、セッション・マネージャも含めて多数のデスクトップ API へアクセスできるようにします。`Dt/Dt.h` および `Dt/Session.h` ヘッダ・ファイルを取り込んで、セッション・マネージャ API にアクセスしてください。

注 - アプリケーションが任意のセッション・マネージャ API を使用している場合、まず `DtInitialize()` または `DtAppInitialize()` を呼び出して `libDtSvc` ライブラリを初期化しなければなりません。詳細は、`DtInitialize(3)` または `DtAppInitialize(3)` のマニュアル・ページを参照してください。

デモ・プログラム

セッション・マネージャのデモは `/usr/dt/examples/dtsession` にあります。詳細は `README` ファイルを参照してください。

関連マニュアル

セッション・マネージャの詳細は、関連するマニュアル・ページと『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』を参照してください。

ドラッグ&ドロップ

共通デスクトップ環境は、 Motif 1.2 ドラッグ&ドロップ API の一番上のレイア上にドラッグ&ドロップ API を提供し、デスクトップにおいて便利で一貫した相互運用可能なドラッグ&ドロップ API をサポートします。共通デスクトップ環境ドラッグ&ドロップ API により、開発者はドラッグ&ドロップを容易に実現できます。ドラッグ&ドロップがあれば、ユーザはディスプレイ上でグラブしたり、ドラッグし

たり、他のオブジェクトにドロップしてオブジェクトの場所の変更やデータ転送を行い、直接画面上でオブジェクトを処理できます。

Motif 1.2 ドラッグ&ドロップはローレベルの機能を提供し、共通デスクトップ環境ドラッグ&ドロップはこれらの機能のポリシーを取り込んでいます。

共通デスクトップ環境ドラッグ&ドロップは、API と、Motif ドラッグ&ドロップへのインタフェースを単純化するプロトコルから成ります。これはバッファ転送プロトコルやドラッグ・カーソルの形状などのポリシーを実現します。共通デスクトップ環境ドラッグ&ドロップ API を組み込みポリシーと共に使用して、一貫した相互運用性を保証してください。共通デスクトップ環境ドラッグ&ドロップ・ポリシーは、テキスト転送およびファイル名転送用の標準 Motif 1.2 ドラッグ&ドロップ・プロトコルと互換性があります。

共通デスクトップ環境ドラッグ&ドロップは、データを転送するのに X 選択機能を使用します。適切なターゲットはすでに存在し、X コンソーシアムに登録されています。2 つのデスクトップ・アプリケーションは、テキスト、ファイル名、データの転送プロトコルによってデータを転送します。

既存のドラッグ&ドロップのための Motif 1.2 API は柔軟性があり、したがって熟練していない開発者には使用するのが幾分難しいところがあります。共通デスクトップ環境ドラッグ&ドロップ API は、API を単純で簡単に使用できるよういくつかの便利な機能を提供しています。

- ドラッグ・アイコンの構成と形状を管理します。

  共通デスクトップ環境ドラッグ&ドロップには、Motif 1.2 のドラッグ・アイコンを作成するデフォルトのソース、状態、操作アイコンのグラフィックがあります。

- バッファ転送プロトコルを定義します。

  Motif 1.2 ドラッグ&ドロップはファイル名とテキスト文字列だけのプロトコルを定義します。

- ドロップにおけるアニメーションを可能にします。

  ドロップを完了したときに呼び出されるアニメーション手続きをドロップ領域が定義できます。

- `TEXT` および `FILE_NAME` 転送のターゲットを列挙します。
- 重複して登録できます。

  テキスト・ウィジェットをテキスト以外のデータ用にドロップ領域として登録しながら、テキストのドロップを受け入れる機能も残しておくことができます。

- 優先順位の付いたドロップ・フォーマットを提供します。

  ドロップ領域でプロトコルを指定した順に優先順位が付けられます。

ライブラリおよびヘッダ・ファイル

デスクトップ・サービス・ライブラリ `libDtSvc` は、ドラッグ&ドロップも含めたあらゆるデスクトップ API へのアクセスを提供します。ドラッグ&ドロップ API をアクセスするには、`Dt/Dt.h` および `Dt/Dnd.h` ヘッダ・ファイルを取り込んでください。

注 - アプリケーションが任意のドラッグ&ドロップ API を使用している場合、まず `DtInitialize()` または `DtAppInitialize()` を呼び出して `libDtSvc` ライブラリを初期化してください。詳細は、`DtInitialize(3)` または `DtAppInitialize(3)` のマニュアル・ページを参照してください。

デモ・プログラム

ドラッグ&ドロップのデモは `/usr/dt/examples/dtdnd` にあります。詳細は `README` ファイルを参照してください。

関連マニュアル

共通デスクトップ環境ドラッグ&ドロップの詳細は、関連するマニュアル・ページと『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』を参照してください。

国際化対応

共通デスクトップ環境は 1 バイトおよびマルチバイトのロケールをサポートするよう国際化対応しています。開発者は、任意の共通デスクトップ環境プラットフォームで実行するため簡単にローカライズできる、国際化対応アプリケーションを作成できます。

共通デスクトップ環境アプリケーション (ソースおよびバイナリの両方) は、いろいろな言語および地域にローカライズでき、複数のベンダおよびハードウェア・プラットフォームで使用できます。

- ラテン・アメリカ
- 西ヨーロッパ
- 日本
- 韓国
- 中国 (繁体字と簡体字)
- タイ語
- ヘブライ語
- アラビア語

共通デスクトップ環境は次のような標準で国際化対応機能を使用します。

- IEEE 1003.2-1992 (POSIX 2 Annex B)
- X Window System、Version 11 Release 5 (ロケールおよび国際化対応テキスト機能)
- Motif 1.2 (Motif クライアントの国際化対応およびローカライズ)
- Motif 2.1 CTL (Complex Text Layout) サポート (アラビア語、ヘブライ語、タイ語環境に対応)

アプリケーションを国際化対応させる場合、マルチバイト文字の入出力をサポートしていることを確認してください。また、メッセージ・カタログを使用していてコードを完全にローカライズできることも確認してください。

デモ・プログラム

`/usr/dt/examples/template` にある描画プログラムのデモは国際化されています。詳細は `README` ファイルを参照してください。

関連マニュアル

共通デスクトップ環境国際化対応の詳細は、開発環境コンポーネントのマニュアル・ページと『プログラマーズ・ガイド (国際化対応編)』を参照してください。

標準フォント名

共通デスクトップ環境で定義された標準フォント名は、すべての共通デスクトップ環境準拠システムで使用できることが保証されています。これらは実際のフォントを示すものではありません。各システムのベンダが最も適切に使用できるフォントにマップするための別名です。アプリケーションでこのフォント名のみを使用していれば、任意の共通デスクトップ環境準拠システムで最も近いフォントを使用できます。これには、最も一般的な設計および形式で使用できる X Window System のフォント名を含みます。

標準フォント名は、別の共通デスクトップ環境プラットフォームでは、通常 X のフォント別名機能によって別のフォントにマップされます。これによって異なるプラットフォーム上のさまざまなフォントから選択しなければならないという問題から解放されます。また、特定のベンダの共通デスクトップ環境の実装でデフォルトのフォント・セットを使用できるようになります。

共通デスクトップ環境は 2 種類の標準フォントを定義します。アプリケーション・フォントとインタフェース・フォントです。アプリケーション・フォントをアプリケーションからの出力に使用してください。Motif ウィジェットおよびデスクトップではインタフェース・フォントを使用します。このデフォルト・フォントは変更しないでください。

アプリケーション・フォント

すべての共通デスクトップ環境プラットフォームで、最小 6 種類のフォントサイズが使用できます。各フォントは、標準フォント名 8、10、12、14、16、18、24 に関連しています。共通デスクトップ環境用 XLFD フォントの記述は次のようになります。

```
-dt-application-*
```

上記のようなパターンを使用すれば有効です。

テキストの表示に使用されるフォントのデザインのバリエーションで最も一般的なのは、serif と sans serif 、およびプロポーショナルと固定幅文字の選択です。これら 2 つのデザインのバリエーションを組み合わせると、次の 4 つのデザインがあります。

- serif でプロポーショナル

- sans serif でプロポーショナル
- serif で固定幅
- sans serif で固定幅

上記 4 つのデザインの一般例は次のとおりです (順序は上記のとおりです)。

- Times Roman (タイムズ・ローマン)
- Helvetica (ヘルベチカ)
- Courier (クーリエ)
- Lucida Typewriter (ルシダ・タイプライタ)

これらのデザインには、(太さと傾斜の組合せによる) 次のような 4 つのスタイルがあります。

- Plain (プレーン)
- Bold (ボールド)
- Italic (イタリック)
- Bold-Italic (ボールド・イタリック)

上記 4 つのスタイルにはそれぞれ 4 つのデザインのバリエーションがあるので、合計 16 種類のフォントが生成されます。この 16 種類のフォントは一般のデスクトップ・コンピューティングで最も一般的に使用されるものです。たとえば Times Roman (タイムズ・ローマン)、Helvetica (ヘルベチカ)、Courier (クーリエ) は 4 つのスタイルにありますが、シンボル・フォントと共に Adobe 13、つまりすべての PostScript プリンタに組み込まれるフォントの最小セットを構成します。

アプリケーションは、正式なフォント・ファミリまたはフォント名を必要としませんが、たとえば固定幅フォント、sans serif フォント、serif フォントなどの使用は必要です。特定の共通デスクトップ環境プラットフォームに存在する正式フォント名を知る必要はありません。共通デスクトップ環境標準フォントは、ベンダのプラットフォームで最適な特定のデザインの選択をデフォルトにしています。

アプリケーションが必要とするフォントのリソース値として、アプリケーションの `app-defaults` ファイルでは概要アプリケーション・フォントの XLFD フォント名を指定して下さい。このフォント名を使用しない場合は、各共通デスクトップ環境プラットフォーム上の各アプリケーションに対して、別の `app-defaults` ファイルを提供する必要があります。

インタフェース・フォント

インタフェース・フォントは、特定のプラットフォームでデスクトップの外観を定義するのに最適化されたフォントの小さなセットです。これらのフォントはウィンドウ・タイトル、ボタン、メニュー、テキスト・フィールドなどに表示されている様に、少量の情報を明確にすばやく伝達します。

標準デスクトップおよび Motif ツールキット・ウィジェットはインタフェース・フォントを使用します。これらのフォントは、アプリケーション・ウィンドウ内で直接使用しないでください。

標準インタフェース・フォント名は標準アプリケーション・フォント名とは異なります。これはアプリケーション・フォント名のように、別の共通デスクトップ環境プラットフォーム上では、別のフォントにマップされます。インタフェース・フォントには次の 3 つのスタイルがあります。

- System

  読み専用テキスト (メニュー、ボタン、ラベルなど、限られた量のテキストに使用)

- User

  エンド・ユーザが入力したテキスト、または `XmText` 型および `DtTerm` 型のウィジェットから構築されるオブジェクトに示されるテキスト

- User bold

  ユーザ・フォントと同じで、ボールド

上記のスタイルサイズには 7 種類のサイズがあります。スタイル・マネージャを使用して、ユーザがデスクトップで使用するインタフェース・フォントのサイズを選択することもできます。

デモ・プログラム

`/usr/dt/examples/template` にある描画プログラムのデモは、独自のインタフェース・フォントを指定しません。共通デスクトップ環境 Motif インタフェース・フォントの表示例を示します。ただし、このデモでは、アプリケーション・フォントは利用していません。

関連マニュアル

標準フォントの詳細は、関連するマニュアル・ページ、特に `DtStdAppFontNames(5)` および `DtStdInterfaceFontNames(5)` の XLFD フォント名のリストに関する記述と、『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』を参照してください。

アプリケーションのエラー・メッセージの表示

共通デスクトップ環境のアプリケーションは、共通モデルに従ってエラー・メッセージと警告を表示します。アプリケーションを実行しているユーザは、メッセージがメッセージ・フッタ、エラー・ダイアログ・ボックス、警告ダイアログ・ボックスのいずれかに表示され、詳細な記述は適切にオンライン・ヘルプに示されると思っています。

この節では、アプリケーションのエラー・メッセージの表示規則を概説します。メッセージ・テキストの処理方法ですので、エラー表示のガイドラインには正確に従ってください。たとえばフロントパネルからアプリケーションを起動すると、ユーザは標準エラーまたは標準出力に送信されるメッセージを見ることができません。共通デスクトップ環境では、そのようなメッセージは多くのユーザが定期的に調べないようなログ・ファイル (`$HOME/.dt/*log`) に出力されます。

エラー・メッセージの表示方法

ユーザに警告、メッセージ、エラー状態を通知する場所を決定するときは、次の規則に従ってください。

- 情報を示すメッセージの場合は、アプリケーションのメッセージ・フッタにテキストを表示する (例:「`MyDoc` ファイルをコピーしました。」)
- エラーまたは重大な警告についてのメッセージの場合は (ユーザにとって重要な操作が失敗した場合のトラブルなど)、エラー・ダイアログ・ボックスまたは警告ダイアログ・ボックスに表示する

エラー・ダイアログに表示する情報

エラー・ダイアログまたは警告ダイアログは、ユーザに次のような情報を示す必要があります。

- 何が起こったか (ユーザの見地から)
- なぜ起こったかをわかりやすく
- 問題の解決方法

メッセージ・ダイアログをオンライン・ヘルプへリンクする方法

追加のバックグラウンド情報が必要な場合、またはエラーを完全に説明するのに4、5 行以上のダイアログが必要な場合は、ユーザを適切なオンライン・ヘルプのセクションに導くボタンを追加してください。

関連マニュアル

アプリケーションにおけるエラー・メッセージの表示、およびメッセージ・ダイアログのオンライン・ヘルプへのリンクの詳細は、『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』を参照してください。

ユーザ・カスタマイズの問題

この節では、アプリケーションのユーザ・インタフェースを設計するときのガイドラインを示します。

使用する色の数

アプリケーションのユーザ・インタフェースを設計するときは、共通デスクトップ環境が Motif およびデスクトップ・ウィジェットに提供するデフォルトのカラー・スキーマを無効にするようなカラーを設定しないでください。アプリケーション定義のカラーについては、次のカラーを使用して他のデスクトップ・アプリケーションとの共用を促進してください。

- 黒

- 白
- 赤
- 緑
- 青
- 黄
- シアン
- マゼンタ
- グレー (濃淡は #e1、#c8、#af、#96、#7d、#64、#4b、#32 の 8 段階)

普通はカラーを指定する必要がなく、デスクトップのスタイル・マネージャでエンド・ユーザが選択したカラーを使用します。

使用するフォント

Motif ウィジェットでは、共通デスクトップ環境が提供するフォントを使用し、アプリケーションのウィンドウが他のデスクトップ・クライアントのウィンドウと同じようになるように、またユーザがスタイル・マネージャを使用してこれらのフォントのサイズを変更できるようにしてください。提供されたフォントを、Motif `fontList` リソース仕様を変更して無効にする場合は、ユーザがアプリケーションでフォントをカスタマイズできるようにさせたければ、その機能を追加しなければなりません。

共通デスクトップ環境標準アプリケーション・フォント名にあるフォントを使用して、(Motif がウィジェット用に使用しているのとは別の) アプリケーションで使用する、`app-defaults` ファイルのリソースを指定してください。これは、アプリケーションがすべての共通デスクトップ環境プラットフォームで適切なフォントを見つけ、プラットフォーム上への移植性が高まることを保証します。詳細は、58ページの「標準フォント名」を参照してください。

注 - スタイル・マネージャは、Motif バージョン 1.2 以降を使用して書かれたアプリケーションのフォントしか管理しません。Motif 1.1 (以前) のアプリケーションにはフォントは正しく提供されません。これらのアプリケーションには `app-defaults` ファイルで独自のフォントを指定してください。

アクセスのしやすさ

この節では、ソフトウェア・アプリケーションを障害者が使用できるようにするためのガイドラインを示します。

身体的な障害への対応

通常はメニューやドラッグ&ドロップなどで操作するアプリケーションの全機能を、キーボードで操作できるようにし、身体的に障害のある人々が容易にアプリケーションを使用できるようにしてください。

視覚障害への対応

視覚障害を持つ人がアプリケーションによりアクセスしやすくするために、次のガイドラインに従ってください。

- アプリケーションのカラーをハードコードしないでください。
- 線、ボーダ、シャドウの厚みなどグラフィック属性をハードコードしないでください。これらの属性はフォント・サイズによって大きさが調整されます。
- フォント・サイズおよびスタイルをハードコードしないでください。
- すべてのウィジェットに記述名を付けてください。特に、パレット項目やアイコンなど画面のラベルに表示されないウィジェットについては、アプリケーション・コードで記述名を入れてください。これによって画面読み込みソフトウェアが記述情報を目の不自由なユーザにも提供できます。

聴覚障害への対応

聴覚障害を持つ人がアプリケーションによりアクセスしやすくするために、次のガイドラインに従ってください。

- すべてのエンド・ユーザは音による通知が聞こえると想定しないでください。
- 適宜、エンド・ユーザが情報を得る方法を耳による合図と目による合図と選択できるようにしてください。
- 情報の取得方法を、聴覚によるものだけに依存しないようにしてください。
- 聴覚による情報取得を行う場合の周波数とボリュームを、エンド・ユーザが調整できるようにしてください。

言語、知覚、その他の障害への対応

視覚、聴覚、身体的な障害に関するガイドラインは、言語、知覚、その他に障害のあるエンド・ユーザにとって役立つものです。可能な限りティアオフ・メニューやユーザ構成のメニューなどの重要なアプリケーションの機能を取り込んでください。

マウスのダブルクリック間隔

エンド・ユーザにとってのアプリケーション間の一貫性を保つために、アプリケーションまたは app-defaults ファイルにダブルクリックの間隔をハードコードしないで下さい。ユーザがスタイル・マネージャでダブルクリック時間を変更すると、アプリケーションは他のデスクトップ・アプリケーションと同様にその変更に従います。

デモ・プログラム

/usr/dt/examples/template にある描画プログラムのデモは、共通デスクトップ環境のデフォルトのカラーとフォントを使用します。このため、ユーザはスタイル・マネージャを使用し、このプログラムのカラーとフォントをカスタマイズできます。詳細は README ファイルを参照してください。

関連マニュアル

カスタマイズの問題の詳細は、『共通デスクトップ環境 スタイル・ガイド』を参照してください。

第 **7** 章

オプションの統合方法

この章で説明する共通デスクトップ環境のコンポーネントは、特殊なタスクを実行するためにデスクトップが提供するサービスを使用できるようにします。

68ページの「共通デスクトップ環境 Motif コントロール・ウィジェット」

71ページの「データ型の作成」

73ページの「アクションの実行」

75ページの「ワークスペース・マネージャ」

76ページの「端末エミュレータ・ウィジェット」

77ページの「テキスト・エディタ・ウィジェット」

79ページの「カレンダ」

80ページの「デスクトップ Korn シェル (dtksh)」

注 - この節で説明するコンポーネントの統合方法以外に、第 5 章で概説する基本的な統合方法に関する指示にも従ってください。第 6 章で説明するコンポーネントは、アプリケーションを高度にデスクトップと統合できるようにするのに重要です。

オプションの統合方法の詳細は、『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』を参照してください。

共通デスクトップ環境 **Motif** コントロール・ウィジェット

共通デスクトップ環境 Motif コントロール・ウィジェットは、共通デスクトップ環境 Motif に同等の機能を提供することにより、OPEN LOOK および Microsoft Windows のアプリケーションを、共通デスクトップ環境へ移植しやすくするよう設計されています。共通デスクトップ環境 Motif ウィジェット・ライブラリ `libDtWidget` には、すべての共通デスクトップ環境アプリケーションに共通する機能を提供するのに使用するウィジェットと関数が含まれています。ウィジェットが提供するものは次のとおりです。

- テキスト・フィールドおよび矢印ボタン・ウィジェット (`DtSpinBox`)

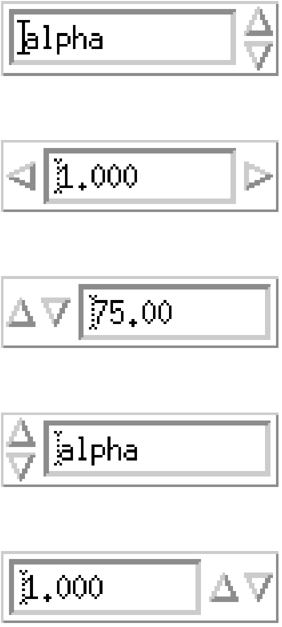

図 7–1　テキスト・フィールドおよび矢印ボタン・ウィジェット (`DtSpinBox`) の例

- テキスト・フィールドおよびリスト・ボックス・ウィジェット (`DtComboBox`)

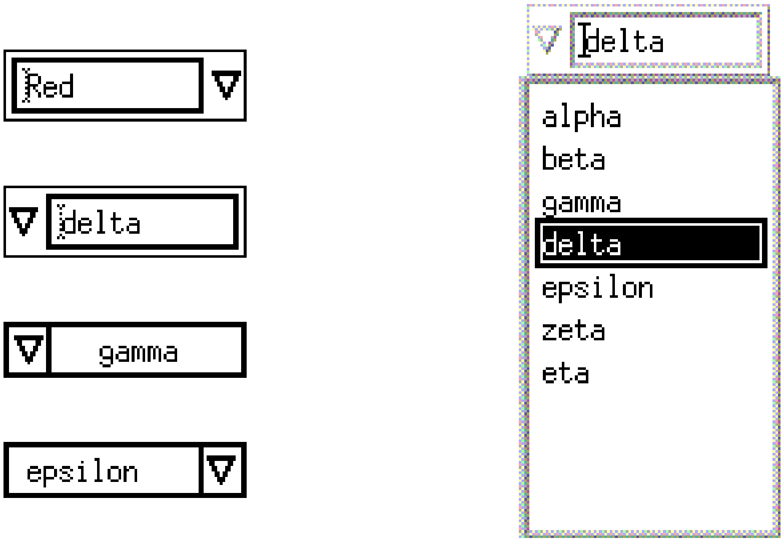

図 7–2　テキスト・フィールドおよびリスト・ボックス・ウィジェット (DtComboBox) の例

- メニュー・ボタン・ウィジェット (DtMenuButton)

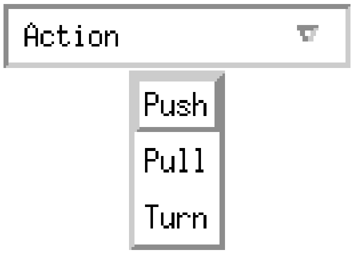

図 7–3　メニュー・ボタン・ウィジェット (DtMenuButton) の例

共通デスクトップ環境の Motif ウィジェット・ライブラリ libDtWidget は共通デスクトップ環境のアプリケーションと OPEN LOOK アプリケーションとの間の操作性をサポートします。つまり、カット、コピー、ペースト、マウス・ボタン機能など、OPEN LOOK が行うのと同じ方法で動作するアプリケーションを作成できます。

注 - 共通デスクトップ環境は Motif 1.2 ウィジェットをすべてサポートします。共通デスクトップ環境 Motif の詳細は、付録 A を参照してください。

表 7–1 共通デスクトップ環境コントロール・ウィジェット

ウィジェット名	説明
DtSpinBox	数値を増減するか、ブラウズしてテキスト文字列のリストから選択するためのコントロールを持つ TextField ウィジェット
DtComboBox	TextField と、TextField に有効な選択項目のリストを提供するポップアップ・リスト・ウィジェットとの組合せ
DtMenuButton	メニュー・バー、またはメニュー区画 (プルダウン・メニュー、ポップアップ・メニュー、オプション・メニュー) 以外の XmCascadeButton ウィジェットの機能を階層化するメニュー機能を提供するコマンド・ウィジェット

Motif 2.1 との互換性

DtSpinBox および DtComboBox ウィジェットの API は、 Motif 2.1 リリースの XmSpinBox および XmComboBox ウィジェットと同等です。API は、アプリケーションが簡単に Motif 2.1 バージョンのウィジェットに交換できるように設計されています。交換するにはクラス、型、作成ルーチンの名前の Dt を Xm に変更してください。たとえばコードにある DtSpinBox という名前はすべて XmSpinBox に変更してください。この情報は、アプリケーションを Motif 2.1 にポートすることを選択した場合に提供されますが、そうすることを推奨しているわけではありません。

注 - 共通デスクトップ環境は、そのウィジェットと Motif 2.1 のウィジェット間での厳密な API またはバイナリ互換を保証するものではありません。

ライブラリおよびヘッダ・ファイル

ライブラリ libDtWidget は、DtSpinBox、DtComboBox、DtMenuButton ウィジェットへのアクセスを提供します。これらのウィジェットの libDtWidget ヘッダ・ファイルは次のとおりです。

- Dt/SpinBox.h
- Dt/ComboBox.h

- `Dt/MenuButton.h`

デモ・プログラム

共通デスクトップ環境 Motif コントロール・ウィジェットのデモは、`/usr/dt/examples/dtwidget` にあります。デモの詳細は、`README` ファイルを参照してください。

関連マニュアル

共通デスクトップ環境 Motif コントロール・ウィジェットの詳細は、関連するマニュアル・ページと『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』を参照してください。

データ型の作成

アプリケーションの基本的な統合方法を実行するときは、データ型を定義できます。この節では、データ型作成の別の面、つまりアクションおよびデータ型のデータベースから情報を抽出する方法に焦点を当てて説明します。

データ型を作成すると、従来の UNIX ファイル・システムが提供するファイルとデータの属性が拡張されます。この拡張には型や属性管理も含まれます。アプリケーションが外部ソースからデータを受信して動作するには、データ型作成 API を使用します。たとえばアプリケーションでアイコンを表示する、またはシステム全体に定義されているアクションを実行する必要があるかもしれません。

データ型作成は次の 2 つの部分から成ります。

- データ基準およびデータ属性を格納するデータベース
- データベースを照会するルーチンの集まり

データ型作成システムは、ファイルまたは基準のセットに基づいたバイト・ベクタの型を決定します。基準には名前、アクセス権、シンボリック・リンク値、内容が含まれます。型に関連付けられた属性は、説明、グラフィカルにそれを示すアイコン、それに適用されるアクションを含んだ、ユーザが見ることができるインタフェースを記述しています。また、属性はデータが準拠する互換フォーマットに名前を付けるためにも存在します。

共通デスクトップ環境のファイル・マネージャおよび共通デスクトップ環境のメールのアタッチメント・ウィンドウは、データ型作成によって、アイコンとファイルに関連するアクションを決定します。たとえば C ファイル *filename*.c について、ファイル・マネージャは .c という拡張子を使用してファイルのデータ型 C_SOURCE を判別します。それから C_SOURCE を使用して *filename*.c に使用できるアイコンのアイコン・ファイル名を決定します。

データベースの読み込み

この節では、アクションおよびデータ型データベースを読み込む API について説明します。この形式と、開発者が定義する任意の共通デスクトップ環境データベースについては、`/usr/dt/man/man4` ディレクトリにあるマニュアル・ページで説明しています。

外部データベースを読み込む API は、次の 2 つの関数から成ります。

- `DtDbLoad()`
- `DtDbReloadNotify()`

`DtDbLoad()` はアクションおよびデータ型データベースを読み込みます。データベースを検索する一連のディレクトリを決定し、見つけた .dt ファイルをデータベースに読み込みます。アプリケーションは、アクションおよびデータ型データベースを照会する任意のルーチンを起動する前に、`DtDbLoad()` を起動する必要があります。

`DtDbReLoadNotify()` を使用して、アクションおよびデータ型データベースの再読み込みイベントの通知を要求してください。アプリケーションがデータベース再読み込みメッセージを受け取ることを登録します。

アプリケーションは、データベースが変更されたら、アプリケーションを再起動しなくても更新があったことがエンド・ユーザにわかるように、必ずデータベースを再読み込みしなければなりません。

データベースの照会

データ・オブジェクトの属性を調べるには、まずオブジェクトの型を決定し、その型に適切な属性値を求めてください。データベース照会関数により、データ型および属性の検索、メモリの解放、データベースの読み込みと解放などのオペレーションを実行できます。これらの関数については、『*Solaris* 共通デスクトップ環境 プ

ログラマーズ・ガイド』および man3 ディレクトリにあるマニュアル・ページで説明しています。

ライブラリおよびヘッダ・ファイル

デスクトップ・サービス・ライブラリ libDtSvc は、データ型作成を含む多くのデスクトップ API へのアクセスを提供します。データ型作成 API にアクセスするには、Dt/Dt.h および Dt/Dts.h ヘッダ・ファイルを取り込んでください。

注 - アプリケーションが任意のデータ型作成 API を使用する場合、またはアクションおよびデータ型データベースを読み込む場合は、DtInitialize() または DtAppInitialize() のいずれかを呼び出して libDtSvc ライブラリを初期化してください。詳細は、DtInitialize(3) または DtAppInitialize(3) のマニュアル・ページを参照してください。

デモ・プログラム

データ型作成のデモは、/usr/dt/examples/dtdts にあります。デモの詳細は、README ファイルを参照してください。

関連マニュアル

データ型作成の詳細は、関連するマニュアル・ページと『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』を参照してください。

アクションの実行

アクション実行 API により、アプリケーションが、ファイルまたはバッファの引き数でデスクトップのアクションを実行できます。アクション引き数の Class、Type、Mode、Number に基づいて、ファイルまたはバッファに適切なアクションを選択します。たとえば [開く] アクションは GIF ファイルに対してはイメージ・ビューアを、複雑なドキュメントに対してはワープロを、通常の ASCII ファイルに対しては単純なテキスト・エディタを起動します。アプリケーションは、アクションの選択や起動の詳細に関与する必要がありません。

アクション実行 API を使用すると、アプリケーションはデスクトップの他の部分と同じ機能を使用します。ユーザにとっては、デスクトップの動作が一定していることになります。

libDtSvc ライブラリのアクション関数は、デスクトップのアクション ([開く] や [印刷] など) をファイルおよびバッファに対して起動する方法を提供します。これには、アクションの動作を変更できるパラメータが含まれます。アクション起動 API は次の関数から構成されます。

DtActionInvoke()	引き数に指定したアクションを起動します。引き数はファイルかバッファです。
DtActionLabel()	アクションに関連した、ローカライズ可能なラベル文字列へのアクセスを提供します。
DtActionDescription()	アプリケーションが起動したアクションに関連する説明情報を含む文字列を返します。
DtActionExists()	指定した名前が既存のアクションに対応するかどうかをチェックする Boolean 関数です。
DtActionIcon()	指定したアクションに関連するアイコン名を返します。

ライブラリおよびヘッダ・ファイル

デスクトップ・サービス・ライブラリ libDtSvc は、アクション など多くのデスクトップ API へのアクセスを提供します。アクション API へアクセスするには、Dt/Dt.h および Dt/Action.h ヘッダ・ファイルを取り込んでください。

注 - アプリケーションが任意のアクション実行 API を使用する場合は、DtInitialize() または DtAppInitialize() のいずれかを呼び出して libDtSvc ライブラリを初期化してください。詳細は、DtInitialize(3) または DtAppInitialize(3) のマニュアル・ページを参照してください。

デモ・プログラム

アクション起動のデモは、/usr/dt/examples/dtaction にあります。デモの詳細は、README ファイルを参照してください。

関連マニュアル

アクションの詳細は、関連するマニュアル・ページと『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』を参照してください。

ワークスペース・マネージャ

共通デスクトップ環境のワークスペース・マネージャは、複数のワークスペースをサポートします。各ワークスペースは仮想画面です。ウィンドウは 1 つのワークスペース、すべてのワークスペース、個々のワークスペースの任意の組合せのいずれにも配置できます。ワークスペースに対しては、ユーザによる動的な追加、削除、名前の変更ができます。

ワークスペース・マネージャ API は、ウィンドウが存在するワークスペースを知るのに必要な、またはウィンドウのワークスペースへの配置方法をコントロールするのに必要なアプリケーションの関数を提供します。さらに、現在のワークスペースがどれであるかなど、ワークスペース全体の状態の変更内容をアプリケーションが監視できるようにします。

共通デスクトップ環境では、デスクトップで実行するアプリケーションはワークスペース・マネージャ API の使用が必ずしも必要なわけではありません。ワークスペース・マネージャを知らなくても、ほとんどのデスクトップ・アプリケーションが予想したとおりに実行されます。特に、アプリケーションが 1 つのメイン・トップレベル・ウィンドウだけを持ち、ICCCM 1.1 および Motif 1.2 に従う場合は、デスクトップで実行するアプリケーションの共通デスクトップ環境ワークスペース・マネージャと統合する必要はありません。

ただし、複数のトップ・レベル・ウィンドウを持つ複雑なアプリケーションは、アプリケーションの状態を正しく保存および復元するために、セッション・マネージャ API と共にワークスペース・マネージャ API を使用する必要があります。ワークスペース・マネージャ API により、アプリケーションはそれぞれのウィンドウが入っているワークスペースを見つけることができます。また、セッションが再開したときに正しいワークスペースにウィンドウを表示することもできます。

ワークスペース・マネージャ API は次の作業を行います。

- ワークスペースの情報の獲得
- 現在のワークスペースの獲得と設定
- ワークスペースの状態の変更内容のクライアントへの通知

- クライアント向けワークスペース関数の追加と削除
- クライアントが配置しているワークスペースの獲得と設定
- 背景ウィンドウの識別

すべてのワークスペース・マネージャ API 関数は接頭辞 `DtWsm` を共用します。

ライブラリおよびヘッダ・ファイル

デスクトップ・サービス・ライブラリ `libDtSvc` は、ワークスペース・マネージャなど多くのデスクトップ API へのアクセスを提供します。ワークスペース・マネージャ API へアクセスするには、`Dt/Dt.h` および `Dt/Wsm.h` ヘッダ・ファイルを取り込んでください。

注 - アプリケーションが任意のワークスペース・マネージャ API を使用する場合は、`DtInitialize()` または `DtAppInitialize()` のいずれかを呼び出して `libDtSvc` ライブラリを初期化してください。詳細は、`DtInitialize(3)` または `DtAppInitialize(3)` のマニュアル・ページを参照してください。

デモ・プログラム

ワークスペース・マネージャのデモは、`/usr/dt/examples/dtwsm` にあります。デモの詳細は、`README` ファイルを参照してください。

関連マニュアル

ワークスペース・マネージャの詳細は、関連するマニュアル・ページと『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』を参照してください。

端末エミュレータ・ウィジェット

`DtTerm` ウィジェットは、ANSI X3.64-1979 型端末エミュレータ (特に拡張された DEC VT220 仕様の端末) をエミュレートするのに必要な機能を提供します。端末エミュレータ・ウィジェット・ライブラリ `libDtTerm` は、端末エミュレータ・ウィンドウを GUI に追加するのに使用する `DtTerm` ウィジェットを提供します。端末エ

ミュレータをアプリケーションに取り込む場合は、共通デスクトップ環境 Motif ウィジェットを使用して、ポップアップ・メニューやスクロール・バーなど表示の拡張機能を追加してください。

共通デスクトップ環境の端末エミュレータは、実行環境の一部ですが、端末として動作し、デスクトップから従来の端末ベースのアプリケーションへのアクセスを可能にするウィンドウです。DtTerm ウィジェットはデスクトップの実行時の端末エミュレータ dtterm の基礎となるものです。

libDtTerm ライブラリには、DtTerm ウィジェットを作成、アクセス、サポートするのに便利な一連の関数が入っています。

ライブラリおよびヘッダ・ファイル

libDtTerm ライブラリは、端末エミュレータの設計や、端末エミュレータ・ウィンドウの GUI への追加を行うために、Motif に基づいた一連のウィジェットを提供します。libDtTerm の API へアクセスするには、Dt/Term.h ヘッダ・ファイルをアプリケーションに取り込んでください。

デモ・プログラム

端末エミュレータのデモは、/usr/dt/examples/dtterm にあります。デモの詳細は、README ファイルを参照してください。

関連マニュアル

DtTerm ウィジェットの詳細は、関連するマニュアル・ページを参照してください。

デスクトップ端末エミュレータの詳細は、端末エミュレータのヘルプ・ボリューム、関連するマニュアル・ページ、および『*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。

テキスト・エディタ・ウィジェット

共通デスクトップ環境のテキスト編集システムは、次の 2 つのコンポーネントから成ります。

- グラフィカル・インタフェース、アクション・インタフェース、ToolTalk インタフェースによって編集サービスを行えるようにする、テキスト・エディタ・アプリケーション dtpad
- 次の編集サービスに関してプログラム・インタフェースを提供するエディタ・ウィジェット DtEditor
    - カット&ペースト
    - 検索と置換
    - 簡単なフォーマット
    - スペルチェック (8 ビットのロケール用)
    - 前の編集を元に戻す
    - ASCII テキスト、マルチバイト・テキスト、データのバッファの入出力をサポートする拡張 I/O 処理機能
    - ファイルの読み取りと書き込みのサポート

Motif テキスト・ウィジェットもプログラム・インタフェースを提供しますが、システム全体で統一されたエディタを要求するアプリケーションは、DtEditor ウィジェットを使用してください。共通デスクトップ環境のテキスト・エディタおよびメール・プログラムは、エディタ・ウィジェットを使用します。このウィジェットは次のような場合に使用してください。

- スペルチェック、元へ戻す、検索／変更などの、DtEditor ウィジェットが提供する機能が必要な場合
- ユーザがファイルのデータを読み書きできるようにする場合
- プログラムはテキストの編集をする必要がなくても、ウィジェットがテキストを制御している場合

ライブラリおよびヘッダ・ファイル

DtEditor ウィジェットは libDtWidget ライブラリにあります。ヘッダ・ファイルは Dt/Editor.h です。

デモ・プログラム

デモは、/usr/dt/examples/dtwidget ディレクトリにあり、 DtEditor ウィジェット (editor.c) の例が入っています。デモの詳細は、README ファイルを参照してください。

関連マニュアル

テキスト・エディタ・ウィジェットの詳細は、関連するマニュアル・ページと『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』を参照してください。

カレンダ

共通デスクトップ環境のカレンダは、ネットワーク環境でユーザが時間とリソースをスケジュール調整できるようにする、インフラストラクチャと API から成ります。カレンダ GUI は共通デスクトップ環境の実行環境の一部です。

カレンダは次の 3 つから構成されています。

- カレンダ・データベースを管理するデーモン
- 一連のハイレベル関数を定義し、カレンダを使用するアプリケーションが、デーモンがサポートする機能にアクセスできるようにする、カレンダおよびスケジュール調整 API
- カレンダおよびスケジュール調整 API のライブラリ実装

また、GUI と TTY の両方の対話に対するユーザ・インタフェースを提供します。システムは、カレンダ項目の入力、削除、変更だけでなくブラウズおよび検索機能もサポートします。これらの機能すべてに、ネットワークによってアクセスできます。

開発環境は、クライアントがカレンダ・データへアクセスするためのライブラリを提供します。ユーザが独自のカレンダ項目属性を定義できるように拡張できます。ライブラリは、データベースの更新を通知するためのクライアント・コールバック機能を提供します。

カレンダ・デーモンは、カレンダおよびスケジュール調整 API コールのライブラリの背後にあるサービスを実装しています。カレンダ項目の削除、挿入、変更をサポートしています。また、カレンダ通知方法を管理し、カレンダ・データベースの作成および削除をサポートします。さらに、カレンダ・データの検索機能も提供します。

カレンダ項目データは、ドラッグ&ドロップおよび ToolTalk メッセージ・インタフェースによって、デスクトップと統合します。

カレンダおよびスケジュール調整 API は、アプリケーション・プログラミング・インタフェース・アソシエーション XAPIA (X.400) カレンダおよびスケジュール調整 API 1.0 を実現したものです。カレンダおよびスケジュール API を使用してアプ

リケーションにカレンダを統合するか、独自のカレンダ・アプリケーションを開発してください。

ライブラリおよびヘッダ・ファイル

カレンダ・ライブラリ libcsa は、ネットワーク環境においてカレンダ・データにアクセスし、管理するプログラム方法を提供します。

libcsa の API にアクセスするには、csa/csa.h ヘッダ・ファイルをアプリケーションに取り込んでください。

デモ・プログラム

カレンダのデモは /usr/dt/examples/dtcalendar にあります。デモの詳細は、README ファイルを参照してください。

関連マニュアル

カレンダの詳細は、関連するマニュアル・ページ、カレンダのヘルプ・ボリューム、『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』を参照してください。

デスクトップ **Korn** シェル **(dtksh)**

デスクトップ Korn シェル (dtksh) は、シェル・スクリプトによってグラフィック・ユーザ対話を行う方法を提供します。ユーザ・インタフェース機能は、共通デスクトップ環境 Motif ウィジェット・セット、Xt イントリンシクス、X11 ライブラリに基づいています。

dtksh は多くの X、Xt、Motif、および共通デスクトップ環境の機能にアクセスできるよう拡張した ksh-93 バージョンです。ksh-93 は、コマンド・シェルおよびプログラミング言語 ksh である Korn シェルのバージョンです。dtksh は、次のものをサポートするよう ksh を拡張したものです。

- シェル・スクリプトから共通デスクトップ環境 Motif ウィジェット・セットへのアクセス

- 完全にローカライズされたシェル・スクリプト。dtksh スクリプトは catopen および catgets コマンドを使用できます。
- 共通デスクトップ環境のアプリケーションのヘルプ・システムへのアクセス
- セッション管理保存状態指示への応答
- ほとんどの共通デスクトップ環境のデスクトップ・サービス・メッセージ・セットへのアクセス
- 多くの共通デスクトップ環境のデータ型作成 API 関数へのアクセス
- ほとんどの共通デスクトップ環境のアクション API 関数へのアクセス

デモ・プログラム

dtksh のデモは /usr/dt/examples/dtksh にあります。デモの詳細は、README ファイルを参照してください。

関連マニュアル

dtksh の詳細は、関連するマニュアル・ページと『*Dtksh* ユーザーズ・ガイド』を参照してください。

付録 A

共通デスクトップ環境 Motif

共通デスクトップ環境 Motif ツールキットは、Motif 1.2.3 ウィジェット・ライブラリの既存の機能を拡張してバグを修正し、いくつかの新機能を追加したものです。Motif 1.2.3 は Motif 1.2 のパッチです。Motif 1.2 と Motif 1.2.3 の API のルック&フィールは同じです。

また、共通デスクトップ環境 Motif ツールキットは、Motif 1.2.3 にはないグラフィカル・ユーザ・インタフェース・オブジェクトのコントロール・ウィジェットを提供します。これらのウィジェットの詳細は、68ページの「共通デスクトップ環境 Motif コントロール・ウィジェット」と『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』を参照してください。

共通デスクトップ環境 Motif は Motif 1.2.3 リリースに機能を追加していますが、以前のバージョンとのバイナリ互換性は残っています。Motif 1.2 アプリケーションとのソースおよびバイナリ互換があります。既存の Motif 1.2 アプリケーションは、共通デスクトップ環境 Motif でコンパイルしてください。既存の Motif 1.2 バイナリは、共通デスクトップ環境 Motif を使用するため変更しなくても動作します。

84ページの「Motif に追加された機能」

84ページの「拡張された機能」

85ページの「Motif ライブラリ」

86ページの「デモ・プログラム」

Motif に追加された機能

共通デスクトップ環境では、次の機能が Motif 1.2.3 に追加され、デスクトップ・アプリケーションをサポートしています。

- ツールキット・エラー・メッセージの完全なローカライズ
- `XmGetPixmap()` および `XmGetPixmapByDepth()` は、アイコン検索パスとして環境変数 `XMICONSEARCHPATH` または `XMICONBMSEARCHPATH` を使用します。いずれの変数も設定されない場合は、`XBLANGPATH` を使用します。これは Motif 1.2 の動作です。詳細は、共通デスクトップ環境 Motif のマニュアル・ページを参照してください。

拡張された機能

共通デスクトップ環境の Xm ライブラリでは、Motif の使いやすさを拡張し、OPEN LOOK や Microsoft Windows との相互運用性を高めています。拡張機能は次のとおりです。

- オプションで、3 つボタン・マウスのボタン 2 を、現在の選択項目の拡張に使用できるようにします。これは OPEN LOOK のアジャスト機能と同じです。
- `PushButton` ウィジェットおよびガジェット、`ArrowButton` ウィジェットおよびガジェット、`DrawnButton` ウィジェットのグループ間を移動するのにタブを使用できるようにします。
- マウス・ボタン 3 で CascadeButton メニューを起動できるようにします。
- `XmFileSelectionBox` ウィジェットの 3 つの新しいリソース (`pathMode`、`fileFilterStyle`、`dirTextLabelString`) を提供し、ルック&フィールを改善します。
- 複数の仮想キー割り当てによって、Microsoft Windows および OPEN LOOK と相互運用できるようにします。

上記の拡張機能はいずれもウィジェット・リソース (`XmFileSelectionBox` 用) か、アプリケーション全体のリソース (その他全般用) によって制御されます。リソースをデフォルト値にすると Motif 1.2 と同じ動作および API が提供されます。拡張機能およびリソースの詳細は、`XmDisplay(3x)` と `XmFileSelectionBox(3x)` のマニュアル・ページを参照してください。

Motif ライブラリ

共通デスクトップ環境 Motif と X11R5 ライブラリを使用して、X Window System 用の共通デスクトップ環境 Motif 準拠アプリケーションを開発してください。共通デスクトップ環境 Motif ライブラリは、Motif 1.2.3 ライブラリのバグを修正して拡張したものです。

Motif ライブラリ (libXm)

共通デスクトップ環境はすべての Motif 1.2.3 ヘッダ・ファイルを提供します。

Motif UIL ライブラリ (libUil)

Motif ユーザ・インタフェース言語 (UIL) は、Motif アプリケーションのユーザ・インタフェースの初期状態を記述するための特殊言語です。共通デスクトップ環境バージョンは基本的に、Motif バージョンと変わりません。

UIL にアクセスするには、`UilDef.h` ヘッダ・ファイル (`uil` ディレクトリにあります) を取り込んでください。

Motif リソース・マネージャ・ライブラリ (libMrm)

Motif リソース・マネージャ (MRM) は、UIL コンパイラが作成した ユーザ・インタフェース定義 (UID) ファイルに含まれる定義に基づくウィジェットの作成を担当します。MRM は UIL コンパイラの出力を解釈し、ウィジェット作成関数に適切な引き数リストを生成します。`libMrm` を使用して Motif リソース・マネージャにアクセスしてください。共通デスクトップ環境バージョンは基本的に、Motif バージョンと変わりません。

`libMrm` にアクセスするには、`Mrm/MrmPublic.h` ヘッダ・ファイルをアプリケーションに取り込んでください。

デモ・プログラム

Motif 1.2 のデモは `/usr/dt/examples/motif` にあります。これらのデモは、共通デスクトップ環境 Motif の拡張機能やウィジェットを取り込んでいません。

関連マニュアル

Motif の詳細は、xiページの「関連文書」にリストされている Motif のマニュアルを調べてください。Motif 1.2.3 の拡張機能に関するさらに詳細な情報が必要な場合は、関連するマニュアル・ページを参照してください。

付録 B

コンポーネントおよびガイドラインのリファレンス

この付録では、CDE 開発環境の全コンポーネントとガイドラインを、関連するライブラリおよびヘッダ・ファイルと共にアルファベット順に並べ、関連するマニュアルのリファレンスを示します。リストしたマニュアル以外に、すべてのコンポーネントのマニュアル・ページが /usr/dt/man ディレクトリにあります。ヘッダ・ファイルは、特に断りがない限り、すべて /usr/dt/include の Dt サブディレクトリにあります。

表 B–1 コンポーネントおよび関連するマニュアル

コンポーネント	ライブラリ	ヘッダ・ファイル	マニュアル
アクションおよびアクション実行	libDtSvc	Action.h	『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』、『*Solaris* 共通デスクトップ環境 ユーザーズ・ガイド』、マニュアル・ページ
アプリケーション・ビルダ			『アプリケーション・ビルダ・ユーザーズ・ガイド』、アプリケーション・ビルダのヘルプ

表 B–1 コンポーネントおよび関連するマニュアル 続く

コンポーネント	ライブラリ	ヘッダ・ファイル	マニュアル
カレンダ	libcsa	csa/csa.h	『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』、カレンダのヘルプ
コントロール・ウィジェット	libDtWidget	ComboBox.h, SpinBox.h, MenuButton.h	『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』、マニュアル・ページ
データ型作成	libDtSvc	Dts.h	『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』、『*Solaris* 共通デスクトップ環境 ユーザーズ・ガイド』、マニュアル・ページ
ドラッグ&ドロップ	libDtSvc	Dnd.h	『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』、マニュアル・ページ
デスクトップ Korn シェル			『*Dtksh* ユーザーズ・ガイド』、マニュアル・ページ
ヘルプ・システム	libDtHelp	Help.h, HelpDialog.h, HelpQuickD.h	『プログラマーズ・ガイド (ヘルプ・システム編)』、マニュアル・ページ
共通デスクトップ環境 Motif	libMrm	Mrm/MrmPublic.h	『*OSF/Motif* プログラマーズ・ガイド リリース 1.2』、『*OSF/Motif 1.2 Reference Guide*』、マニュアル・ページ

表 B–1 コンポーネントおよび関連するマニュアル　続く

コンポーネント	ライブラリ	ヘッダ・ファイル	マニュアル
	libUil	uil/UilDef.h	『*OSF/Motif* プログラマーズ・ガイド リリース 1.2』、マニュアル・ページ
	libXm	Xm/*.h	『*OSF/Motif* プログラマーズ・ガイド リリース 1.2』、『*OSF/Motif 1.2 Reference Guide*』、マニュアル・ページ
セッション・マネージャ	libDtSvc	Session.h	『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』、マニュアル・ページ
端末エミュレータ・ウィジェット	libDtTerm	Term.h	マニュアル・ページ
テキスト・エディタ・ウィジェット	libDtWidget	Editor.h	『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』、 マニュアル・ページ
ToolTalk メッセージ・サービス	libtt	Tt/tt_c.h, Tt/tttk.h	『*ToolTalk* メッセージの概要』、マニュアル・ページ
ワークスペース・マネージャ	libDtSvc	Wsm.h	『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』、マニュアル・ページ

表 B–2 ガイドラインおよび関連するマニュアル

ガイドライン	マニュアル
国際化対応	『プログラマーズ・ガイド (国際化対応編)』
標準フォント名	『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』
エラー・メッセージの表示	『*Solaris* 共通デスクトップ環境 プログラマーズ・ガイド』
ユーザのカスタマイズ (使う色、アクセス機能、マウスのダブルクリック間隔)	『共通デスクトップ環境 スタイル・ガイド』

索引

数字

1 バイトのロケール, 56
64 ビットトライブラリ, 31

A

ANSI C, 25, 39
app-defaults ファイル, 59, 63, 65
AppBuilder (アプリケーション・ビルダ), 37

C

C++, 25

D

DEC VT220, 77
DtAppInitialize, 36, 54, 56, 76 - 74
DtComboBox, 69
DtDbLoad(), 72
DtDbReLoadNotify(), 72
DtInitialize, 36, 54, 56, 76 - 74
dtksh, 80
DtMenuButton, 69
dtpad, 78
DtSessionRestorePath(), 53
DtSessionSavePath(), 53
DtSpinBox, 68

G

GUI エンジン
  アプリケーション・ビルダ, 15
  デスクトップ Korn シェル, 15
GUI エンジン
  共通デスクトップ環境 Motif ツールキット, 13

K

Korn シェル, 80
ksh-93, 80

M

makefiles, 40, 41
Microsoft Windows, 68, 84
MIME (Multipurpose Internet Mail Extensions), 11
Motif, ix, 3, 35, 37
  拡張された機能, 84
  関連文書, xi
  共通デスクトップ環境, 84, 86
  追加された機能, 84
  デモ・プログラム, 86
Motif 1.2, 27, 39, 57, 69, 75
  共通デスクトップ環境 Motif, 84
Motif 2.0, 13
Motif 2.1, 70
Motif 2.1 CTL サポート, 57

O

OPEN LOOK, ix, 13, 68

P

POSIX, xii, 40, 57

S

SGML (Standard Generalized Markup Language), 14, 48

T

ToolTalk マニュアル, xii
ToolTalk メッセージ・サービス, 5, 16, 17, 35, 37
  デスクトップ・メッセージ・セット, 50
  デモ・プログラム, 52
  メッセージ・セット, 50
  メディア・メッセージ・セット, 50
  ライブラリおよびヘッダ・ファイル, 52

U

UNIX, 5, 8, 24, 48, 71

V

VGA, 41

W

WM_COMMAND, 53
WM_SAVE_YOURSELF, 53

X

X.400 アプリケーション・プログラミング・インタフェース・アソシエーション (XAPIA), 6
X11R5, 25, 35, 39, 57
X11 ディスプレイ・マネージャ (XDM), 8
XLFD フォント名, 61
Xlib マニュアル, xii
XmAddWMProtocols(), 53
XmAddWMProtocolsCallback(), 53
XmGetPixmap(), 84
XmGetPixmapByDepth(), 84
Xt マニュアル, xii
X ベースのアプリケーション, 26, 33
X マジック・クッキー・スキーマ, 9

あ

アイコン・エディタ, 41
アクション, 5, 9, 18, 20
アクションの起動
  デモ・プログラム, 74
  ライブラリおよびヘッダ・ファイル, 74
アクションの実行
  API, 74
アクセスのしやすさ
  言語、知覚、その他の障害, 65
  視覚障害, 64
  身体的な障害, 64
  聴覚障害, 64
アプリケーション
  64 ビットの開発, 31
アプリケーション開発
  アプリケーション・ビルダ, 38
  デスクトップ命名規則, 35
アプリケーション管理, 9
アプリケーション・サーバ, 9
アプリケーションの開発
  アプリケーション・ビルダ, 37
  ガイドライン, 24, 25
  公開しているインタフェースと公開していないインタフェース, 36
  初期化と libDtSvc, 36
  データベースのガイドライン, 36
  デスクトップ命名規則, 34
  統合のレベル, 33
  必要なソフトウェア, 25
アプリケーションの開発に使用するコンパイラ, 25
アプリケーションの開発に必要なソフトウェア, 25
アプリケーション・ビルダ, 15
  AppBuilder, 37
  使用する場合, 37
  デモ・プログラム, 38
アプリケーション・フォント, 58, 59
アプリケーション・プログラミング・インタフェース・アソシエーションXAPIA (X.400), 80
アプリケーション・マネージャ, 7, 10, 11
アプリケーション・メッセージ機構, 16

い
移植性の問題, 39, 41
印刷マネージャ, 7
インタフェース
公開していない, 36
公開している, 36
インタフェース・フォント, 58, 60

う
ウィジェット
DtTerm, 77
Motif 1.2, 69
Motif 2.1, 70
共通デスクトップ環境, 68, 71
互換性のガイドライン, 42
互換のためのガイドライン, 42
ヘルプ・システム, 49
ウィンドウ管理, 11
ウィンドウ・マネージャ, 11

え
エラー・メッセージ
エラー・ダイアログに表示する情報, 62
表示, 61, 62
表示方法, 61
メッセージ・ダイアログをオンライン・ヘルプへリンクする方法, 62

お
オブジェクト管理, 10
オプションの統合, 81
オプションの統合方法, 34, 68

か
ガイドライン
アクセスのしやすさ, 64, 65
アプリケーションの開発, 24, 25
エラー・メッセージの表示, 61, 62
共通デスクトップ環境 Motif ウィジェット・バイナリの互換性, 42
共通デスクトップ環境の Motif ウィジェット・バイナリの互換性, 42
国際化対応, 56, 57
使用する色の数, 62
使用するフォント, 58, 61
データベースの形式, 36
マウスのダブルクリック間隔, 65
開発環境ヘッダ・ファイル, 27
開発環境ライブラリ, 27, 30
カスタマイズ
デスクトップ, 24
ユーザが行うカスタマイズ, 62
ユーザ問題, 65
カレンダ, 7
XAPIA, 80
デモ・プログラム, 80
ライブラリおよびヘッダ・ファイル, 80

き
疑似カラー・ディスプレイ, 41
規則
デスクトップの命名, 35
デスクトップ命名, 34
既存アプリケーションの実行, 26
既存のアプリケーションと共通デスクトップ開発環境, 26
基本的な統合方法, 34
共通デスクトップ環境 Motif, 13
Motif に追加された機能, 84
拡張された機能, 84
ライブラリ, 85
共通デスクトップ環境 Motif ウィジェット, 13
共通デスクトップ環境 Motif ツールキット, 13
共通デスクトップ環境アーキテクチャ, 3
共通デスクトップ環境ウィジェット, 13
DtComboBox, 69
DtMenuButton, 69
DtSpinBox, 68
デモ・プログラム, 71
ライブラリおよびヘッダ・ファイル, 70

く
グラフィカル・ユーザ・インタフェース・エンジン, 5

け
言語障害とユーザのカスタマイズ, 65

こ
公開していないインタフェース, 36
公開しているインタフェース, 36
互換性
共通デスクトップ環境ウィジェットと Motif 2.1 ウィジェットとの互換性, 70
共通デスクトップ環境ウィジェットのガイドライン, 42
共通デスクトップ環境ウィジェットのためのガイドライン, 42
国際化対応, 37
1 バイト, 56
デモ・プログラム, 57
標準, 57
マルチバイトのロケール, 56
ロケール, 56
コンパイル
アプリケーション, 25, 38
共通デスクトップ環境による Motif 1.2 アプリケーションのコンパイル, 84

さ
サブ・プロセス・コントロール (SPC), 16

し
視覚障害とユーザのカスタマイズ, 64
実行環境, 26
カレンダ GUI, 79
端末エミュレータ, 77
マニュアル・セット, xiii
実行時
ディレクトリとアプリケーションの実行, 25
障害とユーザのカスタマイズ
言語、知覚、その他, 65
視覚, 64
身体的, 64
聴覚, 64
使用する色の数とユーザのカスタマイズ, 62
使用するフォント, 63
身体的な障害とユーザのカスタマイズ, 64

す
推奨する統合方法, 34, 48, 65
スタイル管理, 12
スタイル・マネージャ, 7, 12, 60, 63, 65

せ
セッション管理, 7
セッション・マネージャ, 7, 9, 37, 75
DtSessionRestorePath(), 53
DtSessionSavePath(), 53
WM_COMMAND, 53
WM_SAVE_YOURSELF, 53
XmAddWMProtocols(), 53
XmAddWMProtocolsCallback(), 53
デモ・プログラム, 54
ライブラリおよびヘッダ・ファイル, 54
セマンティック記述言語 (SDL), 14

た
端末エミュレータ
DEC VT220 仕様, 77
DtTerm ウィジェット, 77
端末エミュレータ・ウィジェット
デモ・プログラム, 77
ライブラリおよびヘッダ・ファイル, 77

ち
聴覚障害とユーザのカスタマイズ, 64

つ
使いやすさ, 24

て
ディスプレイ・サポート, 41
ディレクトリ構造, 28, 30
データ型の作成, 17, 18
2 つの部分, 71
データベース照会関数, 72
データベースの読み込み, 72
デモ・プログラム, 73
ライブラリおよびヘッダ・ファイル, 73
データ対話型 GUI, 4 - 6

データベース, 19
DtDbLoad(), 72
DtDbReloadNotify(), 72
形式, 36
照会関数, 72
よみこみ, 72
データベースの形式, 36
テキスト・エディタ, 7
DtEditor ウィジェット, 78
dtpad, 78
ウィジェットの使用時, 78
ウィジェットを使用する場合, 78
デモ・プログラム, 78
ライブラリおよびヘッダ・ファイル, 78
テキスト・フィールドおよび矢印ボタン・ウィジェット (DtSpinBox), 68
テキスト・フィールドおよびリスト・ボックス・ウィジェット (DtComboBox), 69
デスクトップ
libDtSvc とアプリケーションの起動, 36
カスタマイズ, 24
共通デスクトップ環境, ix
統合サービス, 24
統合のレベル, 33
命名規則, 34, 35
ルック&フィールの目的, 25
デスクトップ Korn シェル, 14
デスクトップ Korn シェル (dtksh), 14, 15, 80
ksh の拡張, 80
デモ・プログラム, 81
デスクトップ管理, 4
アプリケーション管理, 10
ウィンドウ管理, 11
スタイル管理, 12
セッション管理, 9
デスクトップ統合サービス, 24
デスクトップのルック&フィール, 24
デスクトップ・メッセージ・セット, 50
デモ・プログラム, 27, 38, 49, 52, 54, 56, 57, 74 - 73, 80, 81

と

統合
オプション, 81
オプションの統合方法, 68
推奨, 65
推奨する統合方法, 48
デスクトップ統合サービス, 24
レベル, 33
統合テクノロジ, 5
アプリケーション・メッセージ機構, 17
統合のレベル
オプション, 81
オプションの統合方法, 34, 68
基本的な統合方法, 34
推奨, 65
推奨する統合方法, 34, 48
ドラッグ＆ドロップ, 37
Motif 1.2 ドラッグ＆ドロップ, 55
デモ・プログラム, 56
ライブラリおよびヘッダ・ファイル, 56

ひ

必要なソフトウェア
アプリケーションの開発, 25
表記上の規則, xviii
標準, 39, 40
国際化対応, 57
標準フォント名
XLFD フォント名, 59
app-defaults ファイル, 59
インタフェース・フォント, 60
デモ・プログラム, 60
ビルダ・インタフェース言語 (BIL), 15

ふ

ファイル範囲指定, 7
ファイル・マネージャ, 5, 7, 10, 11, 17
ファイル命名規則, 41
フォント
XLFD, 59, 61
アプリケーション, 58, 59
インタフェース, 58, 60
標準フォント名, 58, 61
ユーザ・カスタマイズの問題, 63
インタフェース
スタイル・マネージャ, 60
プログラム
デモ, 27
例, 27

プログラム例, 27
プロセスの実行, 16
プロトコル
WM_COMMAND, 53
WM_SAVE_YOURSELF, 53
フロントパネル, 5, 7, 11

へ

ヘッダ・ファイル, 29
ヘルプ・システム, 7, 37
SGML, 48
UNIX マニュアル・ページ, 48
設計者用, 48
デモ・プログラム, 49
プログラマ用, 48
ヘルプタグ, 48
ライブラリおよびヘッダ・ファイル, 49
ヘルプタグ, 14, 48

ま

マウスのダブルクリック間隔, 65
マニュアル・セット
開発環境, xii
実行環境, xiii
マニュアル・ページ, 30
マルチバイトのロケール, 56
マルチユーザでの共用, 4, 6, 7

め

命名規則
注意事項, 35
デスクトップ, 34, 35
ファイル, 41
メール・プログラム, 5, 7, 11, 15, 17
メソッドの実行, 18
メッセージ・セット, 50
メディア・メッセージ・セット, 50
メニュー・ボタン・ウィジェット
(DtMenuButton), 69

ゆ

ユーザ・インタフェース言語 (UIL), 15
ユーザ・カスタマイズの問題, 62
アクセスのしやすさ, 64, 65
使用する色の数, 62
使用するフォント, 63
デモ・プログラム, 65
マウスのダブルクリック間隔, 65

ら

ライブラリ
64 ビット, 31

り

リモート・プロシージャ・コール (RPC), 6

る

ルック&フィール, 24

ろ

ログイン・マネージャ, 8
ロケール
1 バイト, 56
アプリケーションをローカライズできる, 56
マルチバイト, 56

わ

ワークスペース・マネージャ, 54
デモ・プログラム, 76
ライブラリおよびヘッダ・ファイル, 76