



Solaris 共通デスクトップ環境 Motif への移行

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 816-4041-11
2002 年 12 月

Copyright 2002 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 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。

Federal Acquisitions: Commercial Software-Government Users Subject to Standard License Terms and Conditions.

本製品に含まれる HG 明朝 L、HG-MincyoL-Sun、HG ゴシック B、および HG-GothicB-Sun は、株式会社リコーがリコービ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タをもとに作成されたものです。HG 平成明朝体 W3@X12 は、株式会社リコーが財団法人日本規格協会からライセンス供与されたタイプフェイスマスタをもとに作成されたものです。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、OpenWindows、ToolTalk、DeskSet、Rolodex 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、オムロンソフトウェア株式会社で共同開発されたソフトウェアです。© Copyright OMRON Co., Ltd. 1995-2000. All Rights Reserved. © Copyright OMRON SOFTWARE Co., Ltd. 1995-2002 All Rights Reserved.

「ATOK」は、株式会社ジャストシステムの登録商標です。

「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) は郵政事業庁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ぬものとします。

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: Solaris Common Desktop Environment: Motif Transition Guide

Part No: 816-0278-10

Revision A



020926@4660



目次

はじめに	7
1 Motif と CDE への移行	13
Motif ツールキットのまとめ	13
Motif への移行	14
既存の Motif アプリケーションを OpenWindows デスクトップ上で実行する	14
OpenWindows デスクトップ用の Solaris Motif アプリケーションを開発する	14
Solaris CDE への移行	15
既存のアプリケーションを Solaris CDE デスクトップ上で実行する	15
OpenWindows アプリケーションを Solaris CDE デスクトップに移植する	15
Motif アプリケーションを Solaris CDE デスクトップに移植する	16
Solaris CDE デスクトップ用アプリケーションの開発とデバッグ	16
2 Motif 環境	17
Solaris ソフトウェア内の Motif パッケージ	17
Motif プログラムをコンパイルしてリンクする	18
共用ライブラリのポリシー	18
3 Solaris の Motif ツールキット	19
Solaris ソフトウェアの Motif ツールキット	19
Solaris 7 Motif	19
IXI Motif 1.2.2 ツールキット	20
CDE アプリケーションの開発に使用できるウィジェット	21

	Solaris Motif のコントロールウィジェット	22
	CDE 端末ウィジェット	24
	テキストエディタウィジェット	25
4	開発環境の移行に関する問題	27
	Solaris CDE に固有の機能	27
	OpenWindows 開発環境と Solaris CDE 開発環境の比較	28
	GUI アプリケーションビルダ	28
	ドラッグ&ドロップ	28
	ToolTalk メッセージ	29
	入力	29
	ヘルプ	30
	国際化	31
	フォント	31
	Motif のアプリケーションと配色	32
5	ツールキットの移行に関する問題	33
	OPEN LOOK と Solaris Motif のツールキットとの比較	33
	OPEN LOOK ユーザインタフェースに固有の機能	34
	Solaris Motif ユーザインタフェースに固有の機能	34
	ツールキット間での主な実装の違い	34
	XView ライブラリと Solaris Motif ライブラリの違い	36
	用語	36
	OPEN LOOK と Motif のツールキットアーキテクチャ	37
	プログラミングモデル	38
	XView と Solaris Motif の違い	39
	X リソース	40
	外部ファイル	40
	OLIT ライブラリと Solaris Motif ライブラリの違い	41
	OLIT ライブラリだけにあるルーチン	41
	Solaris Motif ライブラリだけにあるルーチン	41
	ウィジェット	42
6	移植に関する問題とその対策	47
	移行の要素	47
	移植の必要があるかどうか	48
	基本的な統合	48

移植を決定した場合	49
移植の利点	49
Solaris CDE 環境に統合する	49
小さい事柄から始める	50
変換と配置	50
Motif GUI ビルダを使用する	51
アーキテクチャの影響	51
GUI と内部	51
静的配置と動的配置	52
ターゲット環境を学習する	54
GUI 開発ツール	55
移行ツールの使用	55
Motif に移行するためのツール	55
Solaris CDE デスクトップに移行するためのツール	56
既存のコードの使用	56
GUI ビルダコード	57
デモコード	57
ヒント	57
フローティングメニュー	57
カラーマップの動作	58
まとめ: 注意事項	59
7 移植例: OPEN LOOK から Solaris Motif へ	61
OpenWindows 3.4 スナップショットアプリケーション	61
変換	62
整列	63
CDE スタイルガイドラインを検討する	63
他の設計上の注意事項	64
A GUI 環境の変更	67
B 国際化と CDE	75
正しい CDE NLS 環境を保証する	75
メッセージカタログ関数	76
ロケール通知	76
文字列と XmString	77
文字列から XmString への変換	77

XmString から文字列への変換	77
app-defaults ファイルを組み込む	77
Motif リソースの現地仕様化	78
メッセージカタログの配信	78
CDE と gencat	78
.msg ファイル	78
.cat ファイル	79
フォント	79
国際化されたシェルスクリプト	79

C 関連文書 81

CDE マニュアル	81
開発環境	81
実行環境	82
ToolTalk マニュアル	82
Motif 2.1 マニュアル	82
グラフィカルユーザインタフェース (GUI)	83
Motif プログラミング	84
OPEN LOOK プログラミング	84
Xt/XLib プログラミング	85

索引	87
----	----

はじめに

このマニュアルの内容は次のとおりです。

- Sun Motif の開発者に関連する問題
- OpenWindows™ 3.6 と Solaris™ 共通デスクトップ環境 (Solaris Common Desktop Environment、以降 Solaris CDE とします) のデスクトップ上で、既存の OPEN LOOK アプリケーションと Motif アプリケーションを実行する方法
- OPEN LOOK アプリケーションと Motif アプリケーションを Solaris CDE 環境に移植する方法

このマニュアルは、OPEN LOOK または Motif のプログラミングの知識がある方を対象としています。このマニュアルを Motif マニュアルと OPEN LOOK マニュアルとともに利用すると、アプリケーションを最新の Sun デスクトップ上で実行できます。

注 - このマニュアルでは、Solaris Motif という用語を、Solaris CDE 別パッケージのソフトウェアまたは Solaris ソフトウェアのどちらかに含まれる Motif ツールキットの意味で使用します。

対象読者

このマニュアルが対象とするのは、次のような方です。

- OpenWindows 3.6 または Solaris CDE デスクトップ用の Solaris Motif アプリケーションの開発に関心をお持ちの Motif のプログラマ
- OPEN LOOK または Motif のプログラマと、既存のアプリケーションをそのまま、またはコードを少し変更して OpenWindows 3.6 または Solaris CDE のデスクトップ上で実行したい方

- OPEN LOOK または Motif のアプリケーションを Solaris CDE デスクトップに移植したい方

このマニュアルは、UNIX プラットフォーム上で OPEN LOOK (XView™ または OLIT) あるいは Motif のアプリケーションを開発する方を対象としています。OPEN LOOK 開発者であれば、Motif にも精通しているものと想定しています。

お読みになる前に

アプリケーションを Solaris CDE デスクトップに移植したいが CDE をよく知らないという方は、まず次のマニュアルを参照してください。

- 『共通デスクトップ環境 プログラマ概要』
CDE 開発環境のコンポーネントの概要を説明します。また、実行 (エンドユーザ) 環境および開発環境を含む CDE システム全体のアーキテクチャの概要も説明します。
- 『Solaris 共通デスクトップ環境 ユーザーズ・ガイド』
Solaris CDE の実行環境について詳細に説明します。

すべての Solaris CDE マニュアルのリストについては、付録 C を参照してください。

内容の紹介

このマニュアルは、次の各章と付録で構成されています。

第 1 章

アプリケーションで実行したいタスクの種類に応じて、このマニュアル内で参照すべき箇所について説明します。

第 2 章

OpenWindows または CDE 環境向けに Solaris Motif アプリケーションを作成する開発者のための情報を取り上げます。

第 3 章

Solaris Motif 7 ツールキットについて説明し、Solaris 2.3 (IXI) Motif ツールキットの規格以外の部分を明確にします。

第 4 章

OpenWindows と CDE の開発環境を比較します。

第 5 章

OPEN LOOK グラフィカルユーザインタフェース (GUI) から Solaris Motif にアプリケーションを移行する方法について説明します。

第 6 章

OPEN LOOK アプリケーションを CDE に移植する場合の注意事項を説明します。

第 7 章

単純な移植例を取り上げます。

付録 A

OPEN LOOK と CDE のグラフィカルユーザインタフェース (GUI) 環境の変更点について説明します。

付録 B

アプリケーションを CDE デスクトップ用に国際化するために、OpenWindows 環境とは異なる作業を実行しなければならない事項について説明します。

付録 C

OPEN LOOK、Motif、CDE の各アプリケーション開発に関連する情報を掲載したマニュアルと記事のリストを示します。

関連マニュアル

OPEN LOOK と Motif の開発者に関連する Solaris CDE マニュアルと関連資料については、付録 C を参照してください。

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:
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 <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`

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。

第 1 章

Motif と CDE への移行

この章では、Solaris CDE の開発者が使用できる各種の Motif と、関連して実行する各作業がこのマニュアルのどこに掲載されているかについて説明します。

- 13 ページの「Motif ツールキットのまとめ」
- 14 ページの「Motif への移行」
- 15 ページの「Solaris CDE への移行」

Motif ツールキットのまとめ

この節では、Solaris 環境でサポートされる各 Motif ツールキットの違いについてまとめます。Solaris 2.4、2.5、2.5.1、または 2.6 操作環境上でコンパイルした Solaris Motif アプリケーションは、Solaris 7 共通デスクトップ環境上で動作します。

- Solaris 7 Motif: バグを修正し、機能拡張した Open Group の Motif 2.1 をベースにしています。Solaris 7 ソフトウェアには Motif ツールキットが含まれます。
- Solaris 2.6 Motif: バグを修正し、機能拡張した OSF/Motif 1.2.5 をベースにしています。Solaris 2.6 ソフトウェアには Motif ツールキットが含まれます。
- Solaris 2.5.1 Motif: バグを修正し、機能拡張した OSF/Motif 1.2.3 をベースにしています。Solaris 2.5.1 ソフトウェアには Motif ツールキットが含まれます。
- Solaris 2.5 Motif: バグを修正し、機能拡張した OSF/Motif 1.2.3 をベースにしています。Solaris 2.5 ソフトウェアには Motif ツールキットが含まれます。
- Solaris 2.4 Motif: バグを修正し、機能拡張した OSF/Motif 1.2.3 をベースにしています。システムにインストールされる CDE のバージョンによって Motif のリリースが変わります。Solaris 2.4 ソフトウェアには必ずしも Motif ツールキットが含まれるとはかぎりません。
- Solaris 2.3 で使用できる Motif: OSF/Motif 1.2.2 をベースにしたバージョンの IXI Motif です。この Motif ライブラリを使用してコンパイルしたアプリケーションは、それより後の Solaris バージョンでは実行できない場合があります。ただし、Solaris 2.4、2.5、または 2.5.1 上で現在実行している Solaris 2.3 Motif アプリケー

ションは、Solaris 2.6 上でも同様に実行できます。Solaris 2.3 ソフトウェアには必ずしも Motif ツールキットが含まれるとはかぎりません。

注 - Solaris CDE 環境では、libDtWidget ライブラリを通して新しいウィジェット (メニューボタンウィジェットや、端末エミュレータウィジェットとライブラリなど) にアクセスできます。詳細は、第 3 章を参照してください。

Motif への移行

Motif に関連して、主に次のプログラミング作業または移植作業が必要になります。

- 既存の Motif アプリケーションを OpenWindows 3.6 デスクトップ上で実行する
- OpenWindows 3.6 デスクトップ用の Solaris Motif アプリケーションを開発する

既存の Motif アプリケーションを OpenWindows デスクトップ上で実行する

標準の Motif 1.2 または Solaris Motif アプリケーションは、OpenWindows 3.6 デスクトップ上で動作します。

OpenWindows デスクトップ用の Solaris Motif アプリケーションを開発する

Motif の知識があり、OpenWindows 3.6 デスクトップ用の Solaris Motif アプリケーションを開発したい方は、第 2 章を参照してください。

また、第 3 章では、Solaris 環境でサポートしている各 Motif ツールキットを比較しているので参照してください。

OPEN LOOK 開発者が OpenWindows 3.6 デスクトップ用の Solaris Motif アプリケーションを開発したい場合は、この節で示した章と付録を必ず参照してください。その他に、次の箇所を参照してください。

- 第 5 章
OPEN LOOK と Motif のグラフィカルユーザインタフェース (GUI) とウィジェットを比較しています。
- 第 6 章
特に GUI の移行に関連する部分を参照してください。

- 82 ページの「Motif 2.1 マニュアル」と 84 ページの「Motif プログラミング」
Motif プログラミングを理解するためのマニュアルのリストが掲載されています。

Solaris CDE への移行

OPEN LOOK または Motif アプリケーションは、Solaris CDE 環境に移植しなくても Solaris CDE デスクトップ上で実行できます。移植する場合は次の各章を参照してください。入手できる CDE マニュアルについては、81 ページの「CDE マニュアル」を参照してください。

Solaris CDE に関連して、主に次のプログラミングまたは移植作業が必要です。

- 既存の OLIT、XView、Motif の各アプリケーションを Solaris CDE デスクトップ上で実行する
- OpenWindows アプリケーションを Solaris CDE デスクトップに移植する
- Motif アプリケーションを Solaris CDE デスクトップに移植する
- Solaris CDE デスクトップ用の Solaris Motif アプリケーションを開発する

既存のアプリケーションを Solaris CDE デスクトップ上で実行する

既存の OLIT、XView、Motif の各アプリケーションは、Solaris CDE デスクトップ上で動作します。

アプリケーションを Solaris CDE デスクトップと統合されているように見せたいがアプリケーションコードを変更したくない場合は、基本的な統合を実行できます。この第 1 レベルの Solaris CDE 統合については、48 ページの「基本的な統合」で説明します。

OpenWindows アプリケーションを Solaris CDE デスクトップに移植する

既存の OpenWindows は、Solaris CDE デスクトップ上でそのまま実行できます。アプリケーションを Solaris CDE 環境に移行して標準サービスを広範囲に使用する場合は、このマニュアルを読んで、OPEN LOOK と Motif のツールキットの違いや、OpenWindows と Solaris CDE のデスクトップの違いを理解してください。

Motif アプリケーションを Solaris CDE デスクトップに移植する

アプリケーションが Motif 2.1 のスタイルガイドに準拠している場合は、CDE スタイルガイドにも準拠しています。Solaris Motif は Motif 2.1 ツールキットに基づいており、CDE スタイルガイドは Motif 2.1 スタイルガイドに基づいています。ただし、アプリケーションを Solaris Motif に移植するには、GUI を変更しなければならないことがあります。

Motif アプリケーションを Solaris CDE デスクトップに移植する場合は、次の章を参照してください。

- 第 3 章。Solaris Motif の拡張部分と、使用可能な CDE ウィジェットに関する節を参照してください。
- 第 4 章。OpenWindows と Solaris CDE の開発環境を比較します。

CDE と Motif 1.2 のスタイルガイドラインの類似点と相違点については、『共通デスクトップ環境 スタイル・ガイド』のチェックリストを参照してください。

Solaris CDE デスクトップ用アプリケーションの開発とデバッグ

Solaris CDE 共有ライブラリは、最新の Solaris ローダ技術により構築され、インタフェースと性能が最適化されています。この技術は、SPARCworks™ バージョン 3.0.1 より前にリリースされたデバッガとは競合します。したがって、CDE でアプリケーションの開発とデバッグを行う場合、SPARCworks バージョン 3.0.1 以降を使用してください。

Solaris Motif アプリケーションを開発する

Motif の知識があり、Solaris CDE デスクトップ用の Solaris Motif アプリケーションを開発したい方は、CDE のマニュアルを参照してください。

OPEN LOOK の開発者で、Solaris CDE デスクトップ用の Solaris Motif アプリケーションを開発したい方は、このマニュアルと CDE のマニュアルを参照してください。

どちらの方も、CDE マニュアルと Motif プログラミングの関連マニュアルのリストについては、付録 C を参照してください。

Solaris CDE の命名規則について

Solaris CDE では、デスクトップクライアント名やデスクトップライブラリ名などに、接頭辞 DT と SDT を大文字と小文字を組み合わせて使用しています。したがって、Solaris CDE デスクトップアプリケーションを作成する場合、これらの接頭辞は使用しないでください。

第 2 章

Motif 環境

この章には、OpenWindows または Solaris CDE 環境用に Solaris Motif アプリケーションを作成する開発者のための情報が掲載されています。

- 17 ページの「Solaris ソフトウェア内の Motif パッケージ」
- 18 ページの「Motif プログラムをコンパイルしてリンクする」
- 18 ページの「共用ライブラリのポリシー」

Solaris ソフトウェア内の Motif パッケージ

Solaris 7 ソフトウェアには、次の Motif 実行時サポートが含まれています。

- 共用ライブラリ
- ヘッダファイル
- キーの割り当て
- uil コンパイラ
- マニュアルページ
- デモとサンプルソース

アプリケーションの開発に必要な Motif ヘッダファイルは、`/usr/dt/include` にあります。Motif ライブラリは、`/usr/dt/lib` にあります。

Motif プログラムをコンパイルしてリンクする

Motif プログラムをコンパイルする場合は、コンパイラで Motif と X Window System のヘッダファイルを検索できるように、次のコンパイラ構文を追加します。

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

次のコンパイラ構文を使用して、リンカーに正しい共用ライブラリを指示します。

```
-R/usr/dt/lib -R/usr/openwin/lib -L/usr/dt/lib -L/usr/openwin/lib
```

次の例は、Motif アプリケーションをコンパイルしてリンクする行を示します。

```
cc -o myprog -I/usr/dt/include -I/usr/openwin/include myprog.c \  
-R/usr/dt/lib -R/usr/openwin/lib -L/usr/dt/lib -lXm -L/usr/openwin/lib \  
-lXt -lX11
```

共用ライブラリのポリシー

新リリースに旧リリースとバイナリ互換のない箇所がある場合は、各共用 Motif ライブラリのメジャーバージョン番号を増やします。各ライブラリのすべての旧バージョンが (Motif が配布するか、他の経路を通じて) 販売されています。したがって、Motif の新リリースをインストールした後も、特定のリリースにリンクされたアプリケーションを引き続き実行できます。

第 3 章

Solaris の Motif ツールキット

この章では、Solaris 7 Motif と IXI Motif について説明し、Solaris CDE アプリケーション開発で利用できるウィジェットについて紹介します。

19 ページの「Solaris ソフトウェアの Motif ツールキット」

21 ページの「CDE アプリケーションの開発に利用できるウィジェット」

Solaris ソフトウェアの Motif ツールキット

この節では Solaris 7 Motif と IXI Motif ツールキットについて説明します。Solaris アプリケーション開発で利用できる Motif ツールキット間の違いについては、13 ページの「Motif ツールキットのまとめ」を参照してください。

Solaris 7 Motif

Solaris 7 Motif は、バグを修正し、機能拡張した OSF/Motif 2.1 をベースにしています。Solaris 7 ソフトウェアには Motif ツールキットが含まれます。

既存の OSF/Motif 2.1 の機能に対する拡張機能

特定の OPEN LOOK ユーザインタフェースと Microsoft Windows の機能をエミュレートするために、Solaris Motif ライブラリには OSF/Motif 2.1 に使いやすい小規模な拡張機能が入っています。使いやすさに関する拡張機能は次のとおりです。

- オプションとして、3 ボタンマウスのマウスボタン 2 を使用して現在の選択を拡張する機能。これは、OPEN LOOK のアジャスト機能と同じです。

- マウスボタン3を使用して「CascadeButton」メニューを起動する機能 (OPEN LOOK との互換性を保つため)
- OPEN LOOK または Microsoft Windows のアプリケーションと整合するように、キーの割り当てをマップし直す機能

Solaris CDE の Motif ライブラリ

Motif ライブラリ (*libXm*)

Solaris CDE には、Motif 1.2.5 のすべてのヘッダファイルが含まれています。Solaris Motif ライブラリは、Motif 1.2.5 ライブラリのバグを修正し機能を拡張したものです。

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指定言語です。Motif UIL ライブラリの CDE バージョンは、実質的には Motif 1.2.5 バージョンから変更されていません。

ヘッダファイル `UilDef.h` (`/usr/dt/include/uil` ディレクトリにあります) を取り込むと、UIL にアクセスできます。

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) には、UIL コンパイラで作成されたユーザインタフェース定義 (UID) ファイル内の定義に基づいてウィジェットを作成する役割があります。MRM は UIL コンパイラの出力を解釈し、ウィジェット作成機能のための引き数リストを生成します。libMrm を使用すると、Motif リソースマネージャにアクセスできます。CDE バージョンは、実質的には Motif バージョンから変更されていません。

ヘッダファイル `Mrm/MrmPublic.h` を取り込むと、アプリケーション内で libMrm にアクセスできます。

関連マニュアル

UIL、UIL コンパイラ、UID、Mrm の詳細は、『OSF/Motif プログラマーズ・リファレンス リリース 1.2』(トッパン発行)を参照してください。

IXI Motif 1.2.2 ツールキット

Solaris 2.3 ソフトウェア開発で使用できる IXI Motif 1.2.2 ツールキットには、標準 OSF/Motif 1.2.2 とは互換性のない機能が含まれています。これらの機能は OSF/Motif 1.2 仕様の一部ではないので、Solaris 2.4 以降のバージョンの Solaris Motif のツールキットには含まれていません。

XmList 簡易関数

IXI Motif には、次の標準ではない関数が含まれています。これらの関数を使用していた場合は、アプリケーションコードから削除します。

- `XmListRecolorItem()`
- `XmListRecolorPos()`
- `XmListSetClientDataPos()`
- `XmListSetClientDatasPos()`

XmForm ウィジェット

IXI Motif と OSF/Motif 1.2.2 では、API は同じですが `XmForm` ウィジェットの実装は異なります。したがって、IXI Motif とリンクされているアプリケーションは、OSF/Motif とリンクし直したときに `XmForm` ウィジェット内で動作や画像に些細な違いが生じることがあります。

Solaris Motif ツールキットは、OSF/Motif 1.2.3 の `XmForm` ウィジェットの実装を使用します。これは IXI Motif の `XmForm` ウィジェットではなく、OSF/Motif 1.2.2 とバイナリ互換です。

Complex Text Layout (CTL) Support

Solaris 7 ソフトウェアは、Motif 2.1 で導入された 5 つの新しい CTL ウィジェットをサポートしています。Solaris 7 のオペレーティング環境で開発された単一のバイナリによって、ヘブライ語、アラビア語、タイ語に標準対応しています。

以下の Motif ウィジェットが新しくサポートされています。

- `XmNotebook` は、全機能ウィジェットで、ノートブックや「タブ」ウィジェットに類似した機能を持っています。
- `XmContainer` は、全機能 GUI アイコンである「ツリー」表示ウィジェットです。
- `XmSpinBox` は、数値テキストフィールドの値の増減を行うユーザコントロールです。
- `XmScale` ウィジェットは新しいバーティカルディスプレイに対応するように変更されています。

CDE アプリケーションの開発に使用できるウィジェット

この節では、Solaris Motif の拡張機能として Solaris CDE アプリケーションの開発に使用できるウィジェットについて説明します。

Solaris Motif のコントロールウィジェット

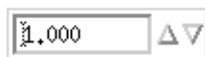
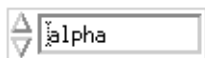
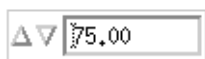
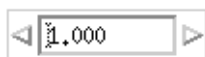
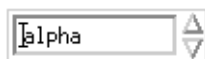
Solaris Motif のコントロールウィジェットは、OPEN LOOK アプリケーションを Solaris CDE デスクトップに簡単に移植できるように、Solaris Motif 内でも同等の機能を提供しています。これらのウィジェットは、Solaris Motif の一部とは見なされず Solaris Motif の拡張機能と見なされます。libDtWidget ライブラリには、すべての CDE アプリケーションで共通の機能を提供するために使用されるウィジェットと関数が入っています。次のウィジェットが提供されます。

表 3-1 CDE のコントロールウィジェット

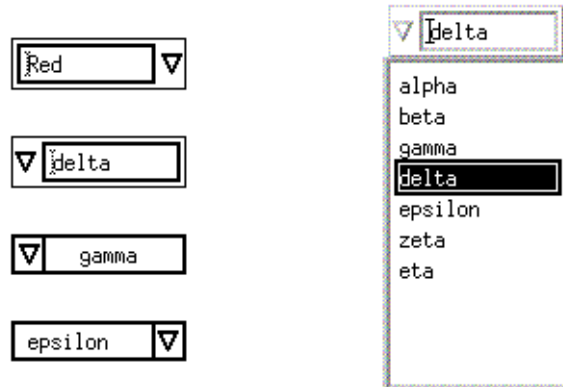
ウィジェット名	説明
DtSpinBox	数値を増減させたり、テキスト文字列のリストをブラウザして選択するための追加のコントロールを持つ TextField ウィジェット。読み専用になります。DtSpinBox の機能は、OPEN LOOK の数値テキストフィールドに似ています。
DtComboBox	TextField ウィジェットと、TextField ウィジェットに有効な選択肢を提供するポップアップリストウィジェットの組み合わせ。読み専用になります。
DtMenuButton	メニューバーの外側またはメニュー区画で XmCascadeButton ウィジェットのメニュー階層化機能を提供するコマンドウィジェット。DtMenuButton ウィジェットの機能は、OPEN LOOK のメニューボタンと同じです。

次に、各タイプのウィジェットの例を示します。

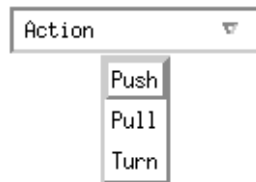
■ テキストフィールドと矢印ボタンのウィジェット (DtSpinBox)



■ テキストフィールドとリストボックスのウィジェット (DtComboBox)



- メニューボタンのウィジェット (DtMenuButton)



`browseSelectCallback` または `defaultActionCallback` がコンボ・ボックスのリストウィジェットのために起動される場合、`DtComboBox` カスタムウィジェットはリストウィジェットではなく、コンボ・ボックスウィジェットの選択コールバックを呼び出します。

このアクションは、リストウィジェットの `browseSelectCallback` または `defaultActionCallback` が起動されたときに呼び出されるリストウィジェットの選択コールバックに依存するアプリケーションでは、エラーが生じることがあります。

注 - Solaris CDE ソフトウェアは、すべての Motif 1.2 ウィジェットをサポートしています。

Motif 2.1 との互換性

DtSpinBox ウィジェットと DtComboBox ウィジェットの API は、Motif 2.1 リリースの XmSpinBox ウィジェットと XmComboBox ウィジェットに似ています。API は、アプリケーションをこれらのウィジェットの Motif 2.1 バージョンに簡単に切り替えられるように設計されています。切り替える場合は、クラス、タイプ、作成ルーチンの Dt という名前を Xm に変更します。

この情報は、アプリケーションを Motif 2.1 に移植する場合のために掲載していますが、移植することをお勧めしているわけではありません。

注 – Solaris CDE ソフトウェアでは、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 互換やバイナリ互換は保証されません。

ライブラリとヘッダファイル

libDtWidget ライブラリを使用すると、DtSpinBox、DtComboBox、および DtMenuButton ウィジェットにアクセスできます。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

Solaris Motif コントロールウィジェットのデモプログラムは、`/usr/dt/examples/dtwidget` にあります。デモ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

Solaris 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。

CDE 端末ウィジェット

DtTerm ウィジェットは、Solaris CDE 開発環境の一部です。このウィジェットには、ANSI X3.64-1979 スタイルの端末 (特に DEC VT220 のような端末が拡張されたもの) をエミュレートするために必要な機能があります。端末ウィジェットライブラリである libDtTerm には、GUI に端末ウィンドウを追加するための DtTerm ウィジェットが入っています。アプリケーションに端末を取り込む場合は、Solaris 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の表示拡張機能を追加します。

Solaris CDE 端末アプリケーションは実行環境の一部であり、端末として動作するウィンドウです。このウィンドウを使用すると、デスクトップ内から従来の端末ベースのアプリケーションにアクセスできます。DtTerm ウィジェットは、デスクトップ実行時端末の基礎です。

libDtTerm ライブラリには、DtTerm ウィジェットを作成、アクセス、およびサポートするための便利な機能一式が入っています。

ライブラリとヘッダファイル

libDtTerm ライブラリには、端末を設計したり端末ウィンドウを GUI に追加したりするための Motif ベースのウィジェット一式が入っています。

ヘッダファイル Dt/Term.h を取り込むと、アプリケーション内で libDtTerm の API にアクセスできます。

デモプログラム

DtTerm のデモプログラムは、`/usr/dt/examples/dtterm` にあります。デモ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

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。

デスクトップ端末アプリケーションの詳細は、端末のヘルプボリューム、関連するマニュアルページ、または『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。

テキストエディタウィジェット

CDE 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構成されています。

- テキストエディタアプリケーション
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、または ToolTalk™ インタフェースを通じて編集サービスを提供します。
- エディタウィジェット DtEditor
次の編集サービスに使用するプログラムインタフェースを提供します。
 - カット&ペースト
 - 検索と置換
 - 単純な書式化

- スペルチェック (8 ビットのロケール用)
- 直前の編集を元に戻す機能
- ASCII テキスト、複数バイトテキスト、データバッファの入出力をサポートする拡張 I/O 処理機能
- ファイルを直接読み書きする操作をサポートする機能

Motif テキストウィジェットはプログラムインタフェースも提供しますが、システム全体で統一されたエディタを保証するアプリケーションでは、DtEditor ウィジェットを使用する必要があります。CDE のテキストエディタとメールプログラムのアプリケーションでは、エディタウィジェットを使用します。次の場合は、このウィジェットを使用してください。

- スペルチェック、元に戻す、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
- ウィジェットがテキストを制御しているが、プログラムはテキストを編集する必要がない場合

ライブラリとヘッダファイル

DtEditor ウィジェットは、libDtWidget ライブラリに入っています。ヘッダファイルは Dt/Editor.h です。

デモプログラム

DtEditor ウィジェットの例を含むデモプログラム (editor.c) は、`/usr/dt/examples/dtwidget` ディレクトリにあります。デモ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

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。

第 4 章

開発環境の移行に関する問題

OPEN LOOK または Motif アプリケーションを Solaris CDE デスクトップに移行する場合は、グラフィカルユーザインタフェース (GUI) の移行だけでなく開発環境の移行も検討する必要があります。OpenWindows と Solaris CDE の開発環境は、さまざまな違いがありますが似ている部分もあります。この章では、この 2 つの開発環境を比較します。

27 ページの「Solaris CDE に固有の機能」

28 ページの「OpenWindows 開発環境と Solaris CDE 開発環境の比較」

Solaris CDE のデスクトップは、OpenWindows 環境の一部である X サーバに基づいています。たとえば Display PostScript™ (DPS)、XIL™、XGL™ ライブラリには、Solaris CDE からアクセスできます。

Solaris CDE に固有の機能

この節では、OpenWindows 環境に類似するものがない Solaris CDE 開発環境だけに組み込まれている機能について説明します。

- デスクトップ Korn シェル (dtksh)。シェルスクリプトを通じてグラフィックなユーザ対話操作を実行できます。デスクトップ Korn シェルの詳細は、『共通デスクトップ環境 *Dtksh* ユーザーズ・ガイド』を参照してください。
- ワークスペースマネージャ。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。ほとんどのデスクトップアプリケーションは、ワークスペースマネージャを知らなくても実行できます。ワークスペースマネージャの詳細は、『共通デスクトップ環境 プログラマ概要』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。

OpenWindows 開発環境と Solaris CDE 開発環境の比較

この節では、OpenWindows と Solaris CDE の開発環境に共通するが実装が異なる機能を比較します。

GUI アプリケーションビルダ

OpenWindows 開発環境と Solaris CDE 開発環境には、GUI アプリケーションビルダが組み込まれています。

- OpenWindows Developer's Guide (Devguide) は、OPEN LOOK アプリケーション用の GUI を生成できるようにするツールです。Devguide には、XView と OLIT のソースコードを生成したコードジェネレータが組み込まれています。Devguide Motif 変換ユーティリティ製品は、最初に Solaris 2.3 リリースとともに出荷されました。この製品は、Devguide GIL ファイルから Motif コードを生成します。
- CDE アプリケーションビルダ (以降、AppBuilder とします) は、Solaris Motif コードを生成します。AppBuilder を使用すると、ドラッグ&ドロップ、ToolTalk メッセージ、セッション、ヘルプ、国際化などのデスクトップサービスをアプリケーションに統合できます。Devguide を使用してアプリケーションを作成した場合は、AppBuilder の GIL から BIL へのコンバータを使用して BIL ファイルを作成できます。BIL ファイルは、AppBuilder で使用する書式です。

Devguide を使用してアプリケーションを構築したことがあれば、すぐに CDE アプリケーションビルダに慣れることができるでしょう。パレットは視覚的に似ています。また、開発とテスト機能やプロジェクトの構築機能などの Devguide の大部分の機能を利用できます。

AppBuilder の詳細は、『共通デスクトップ環境 プログラマ概要』と『共通デスクトップ環境 アプリケーション・ビルダ・ユーザズ・ガイド』を参照してください。

ドラッグ&ドロップ

OpenWindows のドラッグ&ドロップと CDE のドラッグ&ドロップは、基礎となっている基本的機能は似ています。どちらのバージョンのドラッグ&ドロップにも、ユーザから見ると同じ結果をもたらす汎用 API が含まれています。

ただし CDE のドラッグ&ドロップには、次の 2 つの目的に役立つ「簡易」API も含まれています。

- Motif 1.2.3 の API よりも使用しやすい

- データ転送方式を定義して実装する

OpenWindows 環境には、ドラッグ&ドロップポリシーに関する規約がありますが、ポリシーの実装は開発者にまかされています。OPEN LOOK アプリケーション内でドラッグ&ドロップを使用する場合は、作成したデータ転送コードの大部分を CDE の簡易 API を使用して圧縮できます。

CDE のドラッグ&ドロップの詳細は、『共通デスクトップ環境 プログラマ概要』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。

ToolTalk メッセージ

OpenWindows ToolTalk メッセージサービスは、CDE ToolTalk メッセージサービスとの互換性があります。また、CDE ToolTalk メッセージサービスには、OpenWindows ToolTalk の機能に加えて次の機能があります。

- 「メッセージセット」と呼ばれる 2 つの標準 ToolTalk プロトコル
- tttrace
tttrace ユーティリティは、ToolTalk API 呼び出しと ToolTalk 内部メッセージ処理を監視できるようにするデバッグツールです。
- ttsnoop
ttsnoop ユーティリティは、使用中のアプリケーションと他のアプリケーション間で送信されるメッセージを監視できるようにするデバッグツールです。

ToolTalk の CDE 拡張機能の詳細は、『共通デスクトップ環境 プログラマ概要』、『共通デスクトップ環境 ToolTalk メッセージの概要』、および tttrace(1) と ttsnoop(1) のマニュアルページを参照してください。

入力

OpenWindows の「分類機構」はファイルの特性、つまり属性を識別します。分類機構により、特定のファイルタイプの印刷方法、アイコン、「オープン」コマンドなどの属性を指定します。分類機構は、次の 2 つの部分で構成されています。

- ファイルタイプ名と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

Solaris CDE の「データ入力」と「アクション」は、分類機構に似ています。データ入力メカニズムは、2 つのテーブル (DATA_ATTRIBUTES と DATA_CRITERIA) で構成されています。この 2 つのテーブルで、特定のファイルタイプのアイコン、アクション、コマンドなどの属性を指定します。DATA_CRITERIA は、分類機構のファイル名空間に相当します。DATA_ATTRIBUTES は、分類機構のタイプ名空間に相当し

ます。DATA_ATTRIBUTES テーブル内のアクションフィールドは、分類機構の以前の印刷方式と開く方式に相当します。Solaris CDE 開発環境では、このフィールドは ACTION という別のテーブルの参照として機能し、従来の分類機構方式から大幅に拡張されています。

データ入力とアクションの詳細は、『共通デスクトップ環境 プログラマ概要』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。

ヘルプ

CDE のヘルプは、次の 3 つの部分が OpenWindows のヘルプと異なります。

- ユーザモデル
- プログラミング作業
- 豊富なヘルプシステム

CDE ヘルプシステムの詳細は、『共通デスクトップ環境 プログラマ概要』と『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)』を参照してください。

ユーザモデル

OpenWindows のユーザモデルはきわめて単純です。アプリケーション内でヘルプを表示したい部分にポインタを合わせて Help キーを押します。

CDE のユーザモデルは似ていますが、多少の違いがあります。アプリケーションで入力フォーカスが配置されている部分のヘルプが表示されます。つまり、ある部分のヘルプを表示するには、アプリケーション内でその部分を選択してから Help キーを押さなければなりません。

CDE のヘルプでは、入力フォーカスを配置できないオブジェクトについては「アイテムヘルプ」を表示します。ユーザがこれらの項目のヘルプにアクセスできるように、アプリケーションの「ヘルプ」メニューに「アイテムヘルプ」選択を組み込まなければなりません。

▼ プログラミング作業

OLIT と XView のヘルプの作成作業はよく似ています。どちらの場合も、アプリケーションにはウィジェットまたはオブジェクトに該当するテキストがそれぞれ表示されます。ユーザが Help キーを押すと、OpenWindows のウィンドウマネージャはどちらのアプリケーションで処理すべきか判断し、そのアプリケーションにイベントを振り分けます。

OLIT には特殊な Help ウィジェットがあり、XView には Help フレームがあり、Help ダイアログ表示領域に正しい情報を取り込んでそのダイアログを画面に表示します。OLIT に `OLRegisterHelp()` への関数呼び出しを組み込むか、XView 内でオブジェクト属性を設定してウィジェット (オブジェクト) に関するヘルプを使用可能にする必要があります。

アプリケーションに CDE ヘルプ機能を持たせるには、必ず次の操作を実行してください。

1. すべての関連ウィジェット上でヘルプコールバックを設定します。
これらのコールバックが、関連付けられたウィジェットのヘルプ情報を提供できるようにしなければなりません。
2. ヘルプダイアログを作成して管理します。
CDE には、ヘルプダイアログボックスと簡易ヘルプダイアログを作成する `DtHelpDialog` と `DtHelpQuickDialog` ウィジェットがあります。CDE には、ヘルプを表示する 1 つの共用ヘルプダイアログはありません。
3. アイテムヘルプを実装します。
この操作のためのソースコードについては、『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)』を参照してください。

豊富なヘルプシステム

OpenWindows のヘルプにはヘルプダイアログが 1 つあり、テキストしか配置できません。CDE のヘルプははるかに豊富です。CDE には、簡易ヘルプとフルヘルプがあります。また、ヘルプダイアログにはテキストだけでなく、カラーグラフィックスやハイパーリンクなど多彩な要素を取り込むことができます。ヘルプを印刷したり、ナビゲーション機能を使用してさまざまなウィジェットに添付されたヘルプを重ね合わせたり移動したりできます。

国際化

CDE アプリケーションだけでなく、OpenWindows アプリケーションも国際化できます。OpenWindows アプリケーションを国際化する手順のうち、CDE 環境内と OpenWindows 環境内とは異なる部分については、付録 B で説明します。

CDE アプリケーションの国際化の詳細は、『共通デスクトップ環境 プログラマ概要』と『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)』を参照してください。

フォント

OpenWindows 環境では、デフォルトのフォントタイプは Lucida です。これは CDE デスクトップでも同じなので、CDE ユーザにとっては OpenWindows デスクトップとのフォントスタイルの違いはわかりません。

OpenWindows 環境内のフォントには所有権を伴うフォント名が付いていますが、CDE では公開されています。すべての CDE フォント名は -dt- で始まり、すべての CDE プラットフォームで共通です。CDE と OpenWindows のフォントは、X11 XLFD フォント仕様に準拠しています。

OpenWindows 環境でも CDE 環境でも、アプリケーションがシステムフォントのデフォルト値を無効にすることはありません。このためユーザは、各自のデスクトップをカスタマイズできます。OpenWindows 環境では、「ワークスペースプロパティ」デスクトップアプリケーションからフォントのタイプとスタイルを選択できます。CDE では、スタイルマネージャからフォントサイズを選択できます。

OpenWindows 環境では、アプリケーションを実行するロケール用のフォントを指定する必要があります。CDE インタフェースフォントを使用すると、アプリケーションはすべてのロケールやデスクトップで同じように動作します。アプリケーションフォントを使用する場合は、従来どおりアジアロケール用のフォントを指定しなければなりません。

CDE フォントの別名 (エイリアス) は、`/usr/dt/config/xfonts/<locale>` にあります。この場合 `<locale>` は、該当するロケールのディレクトリです。スタイルマネージャが使用するデフォルトのフォントリソースは、`/usr/dt/config/<locale>/sys.fonts` に入っています。

CDE フォントの詳細は、『共通デスクトップ環境 プログラマ概要』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。

Motif のアプリケーションと配色

CDE 環境では Motif 用の色のしきい値が変更されます。したがって、Motif のアプリケーションで、Solaris CDE での配色が OpenWindows 環境とは異なる場合があります。

第 5 章

ツールキットの移行に関する問題

この章では、ツールキットの用語、共通のウィジェット、CDE 専用ウィジェット、高度なウィジェットについて説明します。また、各種ライブラリについても説明します。

- 33 ページの「OPEN LOOK と Solaris Motif のツールキットとの比較」
- 36 ページの「XView ライブラリと Solaris Motif ライブラリの違い」
- 41 ページの「OLIT ライブラリと Solaris Motif ライブラリの違い」

OPEN LOOK と Solaris Motif のツールキットとの比較

OPEN LOOK アプリケーションを Solaris Motif アプリケーションと比べると、対照的な視覚的要素がいくつかあることが一目でわかります。たとえば、OPEN LOOK のボタンは丸型ですが Solaris Motif のボタンは四角型です。三次元的に見せるためにボタンや他のオブジェクトにつけるシェーディングも異なります。このような体裁上の要素はプログラムの動作には影響せず、移植するときには無視されることが多いですが、CDE の見た目からかけ離れるとアプリケーションは CDE スタイルガイドに準拠しなくなります。

変換作業中にいくつかの違いが重要になることがあります。この節では、これらの機能や他の GUI 要素のうち最も重要なものについて説明します。

- Solaris Motif にはない OPEN LOOK ユーザインタフェースの機能
- OPEN LOOK ユーザインタフェースで表示されない Solaris Motif の機能
- どちらの仕様にもあるが実装が異なる機能や他の要素

スタイルガイドのリストと、OPEN LOOK および Solaris Motif の GUI を説明している他の参考資料については、「はじめに」を参照してください。

OPEN LOOK から CDE への GUI 環境の変更点の詳細は、付録 A を参照してください。

OPEN LOOK ユーザインタフェースに固有の機能

次の機能は OPEN LOOK ユーザインタフェースに組み込まれており、XView や OLIT (またはその両方) に実装されていますが、Solaris Motif にはありません。

- 分割ウィンドウコントロール
- スクロールバーのアンカー
- メニューのデフォルト
 - 自動デフォルト。デフォルトを選択するショートカット方式
 - 「リング」で表示されるデフォルトのメニュー項目
- フォントチューザウィジェット
- 通知でのポインタ移動
- アイコン化されたウィンドウへのドロップ

Solaris Motif ユーザインタフェースに固有の機能

次の機能は Solaris Motif には組み込まれていますが、OPEN LOOK ユーザインタフェースには組み込まれていません。

- フロントパネル
- ハイパーテキスト機能を使用してヘルプボリュームを生成し登録するためのツール
- 『共通デスクトップ環境 スタイル・ガイド』で指定され、アプリケーションで使用可能な追加のユーザインタフェースオブジェクト(「ウィジェット」)

ツールキット間での主な実装の違い

OPEN LOOK ユーザインタフェースと Solaris Motif は多くの機能がほぼ同じですが、実装には重要な違いがあります。最も重要な違いは次のとおりです。

- ティアオフメニュー
- 入力フォーカスインジケータ
- ウィジェットクラス (スライダとゲージに対してスケールなど)
- 二次 (secondary) テキストセレクション

その他にも、次の重要な違いがあります。

- キーボードの割り当て
- 各ウィンドウに関連付けられたウィンドウマネージャコントロール

- 国際化
- マウスボタンの動作

この節の残りの部分では、これらの機能について説明します。

ティアオフメニュー

Solaris Motif では、OPEN LOOK ユーザインタフェースの「ピンでとめられたメニュー」の代わりにティアオフメニューを使用します。メニュー最上部の破線を選択すると、そのメニューをそのままの状態を表示しておくことができます。

二次テキストセレクション

OPEN LOOK の二次テキストセレクションは、Motif のクイック転送機構とほぼ同じです。

ウィンドウコントロール

もっとも重要な実装上の違いの1つは、ウィンドウコントロールに関するものです。Motif ユーザがタイトルバーの左上隅の「ウィンドウ」メニューボタンを押すか、OPEN LOOK ユーザがウィンドウのバックグラウンドのどこか(ヘッダなど)で「メニュー」ボタンを押すとメニューが表示されます。この2つの GUI に基づくオプションによってキーの違いが生じます。

Motif の「ウィンドウ」メニューでは、「復元」、「移動」、「サイズ」、「アイコン化」、「最大表示」、「奥へ」、「閉じる」を選択できます。OPEN LOOK の「ヘルプウィンドウ」メニューでは、「クローズ」、「フルサイズ」、「プロパティ」、「後ろへ」、「再表示」、「終了」を選択できます。この2つのリストは根本的には同じですが、その効果には大きな違いがあります。

OPEN LOOK ユーザインタフェースでは、「クローズ」オプションを選択するとウィンドウが最小化(アイコン化)され、「終了」を選択するとアプリケーションが終了します。

Motif では、「アイコン化」オプションを選択するとウィンドウがアイコン化され、「閉じる」オプションを選択するとアプリケーションが終了します。OPEN LOOK ユーザインタフェースに慣れたユーザの多くは、ウィンドウを閉じてアイコン化するときに Motif プログラム自体を終了してしまうことがあるので注意してください。

マウスボタンの動作

マウスボタンの構造はどちらの仕様でもよく似ていますが、混乱を招く可能性のある重要な違いがあります。

表 5-1 は、OPEN LOOK の3つのマウスボタンのデフォルトの左から右へのマッピングを示します。

表 5-1 OPEN LOOK のマウスボタン

ボタン	機能
セレクト	オブジェクトを指定するか、オブジェクトとコントロールを操作してドラッグする
アジャスト	選択したオブジェクトの数を増減させる
メニュー	ポインタ位置または指定したオブジェクトに関連付けられたメニューを表示する

表 5-2 は Motif の 3 つのマウスボタンの割り当てを示します。OPEN LOOKと同様に、デフォルトは左マウスボタンから始まります。

表 5-2 Motif のマウスボタン

ボタン	機能
BSelect	位置カーソルを選択、起動、設定、またはドラッグする
BTransfer	要素を移動してコピーし、転送内容をドラッグ&ドロップする。OPEN LOOK の「アジャスト」ボタンとしてカスタマイズできる。
BMenu	メニューを表示する

XView ライブラリと Solaris Motif ライブラリの違い

この節では、XView と Solaris Motif のライブラリの違いについて説明します。

用語

XView ツールキットと Xt (OLIT と Motif) のツールキットは、次の用語を使用します。

XView	Xt (OLIT と Motif)
パッケージ	ウィジェット
属性	リソース

XView	Xt (OLIT と Motif)
フレーム	シェル

XView は XLib を直接ベースとしています。Motif はイントリンシクス (Xt) ツールキットを、イントリンシクスツールキットは XLib をそれぞれベースとしています。

XView	OLIT	Motif
XLib	Xt	Xt
	XLib	XLib

このように根本的な違いがあるため、環境を初期化し、グラフィカルオブジェクトを作成、変更、削除する基本的なライブラリ関数には、次の例のような違いがあります。

XView	Motif/Xt
xv_init()	XtAppInitialize()
xv_create()	XtCreateWidget()
xv_set()	XtSetValues()
xv_destroy()	XtDestroyWidget()

たとえば、イベント処理コールバックを取り扱う関数や国際化機能はさらに複雑です。これらの機能には、単純な 1 対 1 の対応関係は存在しません。

OPEN LOOK と Motif のツールキットアーキテクチャ

OLIT と Motif のアーキテクチャはよく似ていますが、XView と Motif には違いがあります。XView から Motif に移行するときには、このようなツールキットの違いに注意する必要があります。

XView のツールキットでは、「パッケージ」と呼ばれるユーザインタフェースオブジェクトと、インタフェースをまとめて保持するルーチンやプロセス (作成ルーチン、イベント処理など) が実装されます。これに対して Motif と OLIT には、基本的にはユーザインタフェースオブジェクトである「ウィジェット」しか実装されず、ルーチンとプロセスはイントリンシクスライブラリに残されます。たとえば xv_init() は、XView ライブラリに入っている関数です。それに対応するものとして OLIT や Motif のプログラミングに使用する関数である XtAppInitialize() がありますが、これは Xt イントリンシクスライブラリに入っています。

これら3つのツールキットは、2つの異なる GUI を示します。XView と OLIT のツールキットの外観は似ていますが、Solaris Motif の外観は大きく異なります。ライブラリ内の関数呼び出し間には、ほぼ1対1の対応関係がありますが、並列プログラムの「動作」は異なります。つまり、Solaris Motif ライブラリを使用するように OLIT (または XView) プログラムを変換した後も、ある程度は OPEN LOOK の外観が残ります。このような状態のプログラムは、どちらのスタイルにも完全に準拠していません。

プログラミングモデル

API は異なりますが、XView と Motif はユーザインタフェースのプログラミングに関して、次のように同じオブジェクト指向方法論に基づきます。

- ツールキットを初期化する
- ユーザインタフェースオブジェクトをインスタンス化する
- ユーザインタフェースオブジェクトのコールバックを登録する
- イベントループに入り、ユーザがユーザインタフェースオブジェクトに関するイベントを生成するまで待つ

すべての関数呼び出しを一方の API から他方へ変換しなければならない場合でも、XView から Motif に移植されるプログラムの構造全体はそのまま保持できます。

共通タイプのユーザインタフェースオブジェクト

XView と Motif は、共通タイプのユーザインタフェースオブジェクトをサポートするユーザインタフェースツールキットです。ただし XView では、1つの XView オブジェクトと同等の効果を生むために、これらのオブジェクトの多くは複数の Motif ウィジェットを必要とする上位に実装されます。

表 5-3 は、XView の共通オブジェクトの基本マッピングと、対応する Solaris Motif ウィジェットを示します。

表 5-3 共通オブジェクトの基本マッピング

XView のパッケージ	対応する Solaris Motif ウィジェット
ベース・フレーム	XmTopLevelShell + XmMainWindow
コマンド・フレーム	XmDialogShell + XmBulletinBoard
確認	XmDialogShell + XmMessageBox (メッセージダイアログ)
キャンバス	XmScrolledWindow + XmDrawingArea
パネル	XmBulletinBoard または XmForm

表 5-3 共通オブジェクトの基本マッピング (続き)

XView のパッケージ	対応する Solaris Motif ウィジェット
パネル・ボタン	XmPushButton
メニュー・ボタン	DtMenuButton (OpenWindows 環境ではアクセスできない)
アプリビエイト・メニュー ・ボタン	XmRowColumn (オプションメニュー)
チェックボックス	XmRowColumn + XmToggleButton (チェックボックス)
排他的選択肢	XmRowColumn + XmToggleButton (ラジオボックス)
スクローリング・リスト	XmScrolledWindow + XmList
メッセージ	XmLabel
スライダ	XmScale
テキストフィールド	XmTextField
数値テキストフィールド	DtSpinBox (OpenWindows 環境ではアクセスできない)
テキスト・サブウィンドウ	XmScrolledWindow + XmText (スクロールテキスト)
TTY サブウィンドウ	XmScrolledWindow + XmTermPane、 DtTerm (OpenWindows 環境ではアクセスできない)
スクロールバー	XmScrollBar
ポップアップ・メニュー	XmMenuShell + XmRowColumn (ポップアップメニュー)
プルダウン・メニュー	XmMenuShell + XmRowColumn (プルダウンメニュー)
プルライト・メニュー	XmMenuShell + XmRowColumn (プルダウンメニュー)
ファイルチューザ	XmDialogShell + XmFileSelectionBox

XView と Solaris Motif の違い

XView では、Motif (Xt) では提供されない多数の X11 関数を抽象化して提供していません。XView と同等の機能を Motif に持たせるには、Xlib の関数を直接使用して、これらの関数をコーディングし直さなければなりません。

Motif には、次の XView パッケージに対応するものはありません。

- アイコン
- サーバ・イメージ
- カラーマップ・セグメント (CMS)
- カーソル
- フルスクリーン
- サーバ
- フォント

Solaris Motif には、ウィンドウを別の表示に分割する OPEN LOOK のユーザインタフェースと同等の機能がありません。この機能を実装するには、Motif のマネージャウィジェットの 1 つをサブクラスに入れなければなりません。

XView の機能には Motif/Xt の API を使用してコーディングできるものもありますが、これらの API はまったく異なるので、全面的に設計変更してコーディングし直す必要があります。これらの機能は次のとおりです。

XView の API	Motif/Xt
選択サービス	Xt 選択 API
ドロップターゲットパッケージ	Motif ドラッグ&ドロップ API ¹
通知用インタフェース	Xt イベント管理 API

1. CDE に移植する場合は、CDE のドラッグ&ドロップ API を使用してください。この API は、Motif 1.2 のドラッグ&ドロップ API の拡張機能で簡単に使用できます。

X リソース

XView アプリケーションと Motif アプリケーションでは、その動作と外観を制御する X リソースが異なります。Motif/Xt のオブジェクトと違って、XView のオブジェクトにはインスタンス名が接続されているとは限りません。XView のリソースはインスタンス名から独立しています。たとえば `Window.Foreground.Color` は、関連するすべての XView オブジェクトに影響を与えます。

Motif/Xt のリソースには、`mainframe.control_panel.button1.foreground` などのクラス名またはインスタンス名が入っています。Xt リソースが複数のオブジェクトに影響を与えるようにするには、ワイルドカードとクラス名を使用します。

外部ファイル

XView は、メッセージファイルや、X リソースが入っているアプリケーション固有のデフォルトファイルなどの複数のファイルを起動時に読み込みます。これらのファイルのなかには、内容と位置が Motif と異なるものがあります (たとえば、Motif/Xt は `$OPENWINHOME` の下のメッセージドメインファイル (.mo ファイル) を読み込みません)。国際化メッセージスキーマは、Motif とはまったく異なります。

OLIT ライブラリと Solaris Motif ライブラリの違い

Solaris Motif と OLIT では、ユーザインタフェース内でオブジェクトを操作するときに役立つ多数のルーチンをサポートしています。

OLIT ライブラリだけにあるルーチン

次の機能は、OLIT ライブラリに固有のものです。

- エラー処理ルーチン
OLIT にはアプリケーションでエラー処理をカスタマイズできるように、特定の範囲のルーチンがあります。
- 動的リソース
OLIT ではユーザがアプリケーションを起動した後で、特定のリソース (色やフォントなど) の値を動的に変更できるようサポートしています。

Solaris Motif ライブラリだけにあるルーチン

次の機能は、Solaris Motif ライブラリに固有のものです。

- UIL サポート
Solaris Motif では、ユーザインタフェースのレイアウト用の UIL 定義を使用するためのエンジンをサポートしています。このように分離されているため、プログラムの実行可能ルーチンをコンパイルし直さなくてもユーザインタフェースを変更できます。OLIT の Devguide 解決法 (golit) は、ユーザインタフェースを GIL ファイル形式で定義できるようにして同様の機能を提供します。ユーザインタフェースを変更する場合は、アプリケーションをコンパイルし直さなければなりません。
- クリップボードルーチン
Solaris Motif には、クリップボードとその選択を管理するライブラリがあります。
- ウィジェット作成ルーチン
Solaris Motif には、特定のタイプのウィジェットまたはウィジェットグループを作成する完全なルーチンセットがあります。
- 複合文字列のサポート
Solaris Motif では、ほとんどのテキストに複合文字列を使用する必要があります。これらの特殊な文字列形式をサポートするために、Solaris Motif には複合文字列を作成して処理するための多数のルーチンが組み込まれています。

ウィジェット

OLIT と Solaris Motif のツールキットは、類似する機能を持つ多数の共通ウィジェットとガジェットをサポートし、それぞれのツールキットではより固有の多数のウィジェットをサポートします。あるウィジェットが一方のツールキットには実装されているが他方には実装されていない場合は、実装されていないツールキット内で複数のウィジェットを使用すると同等のオブジェクトを構築できる場合が多いです。

表 5-4 は共通ウィジェット名と、各ツールキット内のそのウィジェットに対応する実際のクラス名を示します

表 5-4 共通ウィジェットのマッピング

OLIT のクラス名	Solaris Motif のクラス名
BulletinBoard	XmBulletinBoard
DrawArea	XmDrawingArea
ExclusiveChoice + RectButtons	XmRowColumn + XmToggleButton (ラジオボックス)
Form	XmForm
Manager	XmManager
MenuButton	DtMenuButton (OpenWindows 環境ではアクセスできない)
NoticeShell	XmDialogShell + XmMessageBox
AbbrevMenuButton	XmRowColumn (オプションメニュー)
PopupWindowShell	XmDialogShell
NonExclusiveRectButton	XmRowColumn + XmToggleButton (チェックボックス)
PopupMenuShell	XmMenuShell
Primitive	XmPrimitive
OblongButton	XmPushButton
ControlArea	XmRowColumn
Scrollbar	XmScrollBar
ScrollingList	XmList + XmScrolledWindow
ScrolledWindow	XmScrolledWindow
Slider	XmScale
StaticText	XmLabel

表 5-4 共通ウィジェットのマッピング (続き)

OLIT のクラス名	Solaris Motif のクラス名
TextEdit	XmText
TextField	XmTextField
RectButton	XmToggleButton

Solaris Motif に固有のウィジェット

この節では、Solaris Motif に固有のウィジェットについて簡単に説明します。

注 – これらのウィジェットのうち、次のウィジェット以外は、OpenWindows 環境での Motif 開発にも使用できます。

DtTerm、DtEditor、DtComboBox、DtSpinBox、DtMenuButton

- DtTerm

このウィジェットは、ANSI X3.64-1979 スタイルの端末エミュレータ (特に DEC VT220 のような拡張機能付き端末) のエミュレートに必要な機能を提供します。
- DtEditor

このウィジェットは、カット&ペーストなどの編集サービス用のプログラムインタフェースを提供します。
- DtComboBox

このウィジェットはテキストフィールドと、そのテキストフィールドに有効な選択肢のリストを提供するリストウィジェットとを組み合わせたものです。このリストから項目を選択すると、そのリスト項目がテキストフィールドに自動的に表示されます。
- DtSpinBox

このウィジェットは、任意の TextField を増減させる便利なユーザインタフェースコントロールです。
- DtMenuButton

このウィジェットは、XmCascadeButton ウィジェットのメニュー階層化機能を補足するコマンドウィジェットです (OLIT には独自のメニューボタンウィジェットがあり、その機能は DtMenuButton と同等です)。
- XmArrowButton

このボタンは、矢印ラベルを表示するプリミティブプッシュボタンウィジェットです。
- XmCommand

これはコマンドボックスを構築し、ユーザが選択したコマンドとコマンド履歴を管理するマネージャウィジェットです。
- XmDrawnButton

このボタンは、ラベルをプログラムで描画できるプリミティブプッシュボタンです。

- XmFrame
このマネージャウィジェットは、1つの子に親を設定して、その子を枠で囲むために使用します。
- XmLabelGadget
このガジェットは、読み専用テキストのための低オーバーヘッドオブジェクトです。
- XmMainWindow
このマネージャウィジェットは、メニューバー、コマンド領域、作業領域をサポートします。
- XmPanedWindow
このマネージャウィジェットは、ウィンドウ内にサイズ変更可能な区画を実装します。
- XmSelectionBox
このウィジェットボックスを使用すると、通常のダイアログボックス内のリストから項目を1つ選択できます。
- FileSelectionBox
このウィジェットは、標準的なファイルの選択方法 (通常はアプリケーションの読み書きのため) を提供します。

libDtWidget ライブラリには DtComboBox、DtSpinBox、DtMenuButton、DtEditor ウィジェットが入っていて、次のライブラリに直接依存します。

- Motif スーパークラスのサポートに使用する Xm ライブラリ
- ウィジェットの作成と操作に使用する Xt ライブラリ
- ベースの X Window System に使用する X ライブラリ

OLIT に固有のウィジェット

この節では、OLIT に固有のウィジェットについて簡単に説明します。

- DropTarget
このプリミティブウィジェットによって、ドラッグ&ドロップ操作のソース側と出力側が実装されます。
- FlatWidget
これらの特殊ウィジェットは、1つのウィジェットのコンテキスト内で任意の数のサブオブジェクトを管理します。多数のサブ項目が入ったメニューや選択オブジェクトを実装すると、メモリを大幅に節約できます。
- FooterPanel
このマネージャウィジェットによって、フローティングフッタ領域を持つウィンドウが自動的にサポートされます。

- RubberTile
このマネージャウィジェットを使用すると、その子に相対サイズの制約を設けることができます。
- Stub
このプリミティブウィジェットを使用すると、サブクラスに入れなくても動作をカスタマイズできます。

第 6 章

移植に関する問題とその対策

この章では、Solaris CDE デスクトップ上での OPEN LOOK アプリケーションの Solaris Motif への移植について説明します。この章に掲載されている情報のほとんどは、OPEN LOOK から OpenWindows デスクトップ上で動作する Motif への移植作業にも当てはまります。

- 47 ページの「移行の要素」
- 48 ページの「移植の必要があるかどうか」
- 49 ページの「移植を決定した場合」
- 50 ページの「小さい事柄から始める」
- 51 ページの「アーキテクチャの影響」
- 55 ページの「移行ツールの使用」
- 56 ページの「既存のコードの使用」
- 57 ページの「ヒント」
- 59 ページの「まとめ: 注意事項」

移行の要素

OPEN LOOK ユーザインタフェースから Motif に移行する作業は複雑です。通常は、ウィジェットを 1 対 1 でスワップするわけにはいきません。コードを 1 行ずつ変換するような単純な作業であるとは期待しないでください。移行作業はアプリケーションによってさまざまで、アーキテクチャに大きな影響を与えるものからウィジェットが少々異なる程度のものまであります。

Solaris CDE デスクトップに移植することは、OPEN LOOK ユーザインタフェースを Motif に移行するだけでなく、アプリケーションに使用可能な Solaris CDE 開発環境の基盤をもつことを意味します。これらの機能の概要については、第 4 章を参照してください。開発環境のコンポーネントとマニュアルの詳細は、『共通デスクトップ環境 プログラマ概要』を参照してください。

移植の必要があるかどうか

まず、アプリケーションを本当に移植する必要があるかどうかを判断する必要があります。15 ページの「既存のアプリケーションを Solaris CDE デスクトップ上で実行する」で説明したように、OPEN LOOK と Motif のアプリケーションは Solaris CDE デスクトップ上で「そのままの状態」で動作します。したがって、Solaris CDE デスクトップ上で動作させるために、既存のアプリケーションを Motif や CDE に移植する必要はありません。

このため、いつ、どんなときにアプリケーションを移植するかを柔軟に判断できます。たとえば、アプリケーションを Solaris CDE デスクトップに移植するのを、手持ちの製品の主要リリースまで待つと決めることもできます。

基本的な統合

基本的なアプリケーションの統合は、アプリケーションを Solaris CDE デスクトップに統合する場合に実行することを強くお勧めする作業です。これらの作業では、アプリケーションのソースコードを変更する必要はありません (ある種の印刷機能を統合してアプリケーション内で印刷可能にするには、コードを少し変更する必要がありますが、基本的な統合の場合はこの作業は省略可能です)。

基本的な統合の場合は、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) を広範囲に使用していません。したがって、ドラッグ&ドロップ機能などのように、デスクトップとその他の対話を行うことはできません。

アプリケーションを Solaris CDE デスクトップに統合するためにコードを変更しなくても、次のさまざまな作業を実行できます。

- アプリケーションのアクションを定義する
- ヘルプボリュームを作成し、ヘルプマネージャのトップレベルから使用可能にする
- アプリケーションをフロントパネルやアプリケーションマネージャに統合する
- 印刷可能にする

Solaris CDE デスクトップは、アプリケーションと他のデスクトップアプリケーションとの相互運用性を提供します。新しいサービスを使用する場合は、コードを変更すれば追加できます (49 ページの「推奨する統合」と 50 ページの「オプションの統合」を参照してください)。

アプリケーション内で印刷可能にする方法と、基本的な統合の手順のリストについては、『Solaris 共通デスクトップ環境 プログラマーズ・ガイド』を参照してください。基本的な統合の手順を実装する方法については、『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。

移植を決定した場合

移植作業を実行してくれるツールはなく、常に従うべき絶対確実なアルゴリズムもありません。移植作業では、Motif と CDE について学習し CDE スタイルガイドラインを理解することが重要です。

移植の利点

CDE デスクトップに移植すると、次の利点があります。

- CDE が提供する新しい機能を活用できる
- 多様なプラットフォームに簡単に移植できるようになる
- アプリケーションが CDE スタイルガイドラインに従って動作することを保証し、より「使いやすく」することができる
- Solaris CDE 環境と OpenWindows 環境が異なる領域で、他の Solaris CDE アプリケーションとの相互運用性を提供する
 - ウィンドウマネージャとの対話
 - ドラッグ&ドロップ
 - セッション管理

Solaris CDE 環境に統合する

推奨する統合とオプションの統合では、コードを変更して、これらのカテゴリに含まれる機能を実装する必要があります。多数の CDE 機能を採用するほど、アプリケーションはデスクトップとさらに統合されます。

推奨する統合

Solaris CDE 開発環境には、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と適正に統合できるように、次のコンポーネントとガイドラインが含まれています。

- ヘルプシステム
- ToolTalk メッセージシステム
- セッションマネージャ
- ドラッグ&ドロップ
- 国際化
- 標準フォント名
- エラーメッセージのガイドライン
- ユーザによるカスタマイズの問題

オプションの統合

次の Solaris CDE コンポーネントを使用すると、デスクトップから提供されるサービスを強化して特殊なタスクを実行できます。

- Solaris Motif のコントロールウィジェット
- データ型
- アクションの起動
- ワークスペースマネージャ
- 端末ウィジェット
- テキストエディタウィジェット
- カレンダー API
- デスクトップ Korn シェル

小さい事柄から始める

アプリケーションを Solaris CDE デスクトップに移植する計画を立てている場合、または移植の練習をする場合は、小規模な例から始めて段階的に作業を進めてください。

変換と配置

単純な GUI を持つ小型アプリケーションの場合は、2 段階のプロセスを使用して移植できます。

- OPEN LOOK ユーザインタフェースから Motif に、GUI をオブジェクト単位で変換します (次節 51 ページの「Motif GUI ビルダを使用する」を参照してください)。
- 変換後の GUI を CDE スタイルガイドラインに準拠するように配置します。この機会を利用して、GUI の使いやすさと顧客特有の問題点を検討します。オブジェクトごとの変換がスタイルガイドに準拠している場合でも、インタフェースを変更できます。

単純な OPEN LOOK アプリケーションと、このプロセスの具体例については、第 7 章を参照してください。

注 - この方法は、大型アプリケーションを移植する場合にはお勧めできません。

Motif GUI ビルダを使用する

アプリケーションビルダ (以降、AppBuilder とします) または Visual WorkShop™ のような Motif GUI ビルダを使用すると、アプリケーション用に新しい Motif GUI を構築できます。50 ページの「変換と配置」で説明した 2 段階のプロセスを使用してポートを移植する場合は、一度試してみてから変更を加えるという柔軟な対応が必要です。特定の GUI 内で手動でコーディングする場合は、変更するのが難しく時間がかかります。AppBuilder を使用すると、オブジェクトをドラッグ&ドロップしてプロトタイプ GUI を簡単に作成できるので柔軟に対応できます。通常は、GUI ビルダを使用して新しい Motif インタフェースをレイアウトする方が、GUI を手作業で移植するよりも時間がかかりません。

実際には、使用する移植プロセスの種類やアプリケーションの大きさに関わらず、Motif GUI ビルダの使用を検討する必要があります。ビルダは GUI とアプリケーションのフレームワークコードをすべて生成するので、アプリケーションコードに専念できます。

アーキテクチャの影響

コードをすぐに Solaris Motif に変換しないでください。まずアプリケーションのアーキテクチャを調べてから、アプリケーションを Motif と CDE に移植してください。

アプリケーションのアーキテクチャが大きいほど、アプリケーションを正しく構成し直す作業は重要になります。このような場合に「変換と配置」戦略を使用すると、移植プロセスがさらに複雑になります。

GUI と内部

重要な機能が周辺の GUI から分離されているプログラムの場合は、OPEN LOOK ユーザインタフェースと Motif との違いが及ぼす影響を無視してもかまいません。しかし、コードがユーザのアクションにリンクされている場合や、OPEN LOOK の特定の機能に依存している場合は、それと同等の Solaris Motif 機能を作成するのが難しい場合があります。

コードモジュールを介して線を描画して、ユーザインタフェースを構成する部分をアプリケーションの残りの部分を構成する部分から完全に分離できる場合は、ユーザインタフェースモジュールを Motif 用に開発されたものと同等のモジュールに置き換えるプロセスに移行作業を集中できます。多くのアプリケーション開発者は、この種の明確な分離が必要なソフトウェア開発方式に従っており、ユーザインタフェースとアプリケーション内部の間でプログラムの境界を正式に指定している場合もあります。

また、ソフトウェアがよりモノリシックで、ユーザインタフェースを提供する機能にアプリケーション固有の機能が埋め込まれている場合は、この2つのタイプの機能を分離するのに膨大な時間が必要になるため移行作業は複雑になります。極端な場合には、スタイルガイドに違反するか、プログラムの一部を設計変更するかを選択しなければなりません。

Solaris CDE ソフトウェアの持つすべての機能を活用するための所要時間は、アプリケーションの配置に大きく左右されます。適切に設計されたアプリケーションほど簡単に移植でき、保守しやすく読みやすいように適切に分割できます。

静的配置と動的配置

前述のように、アプリケーションを Motif に移植する作業は、オブジェクトごとにスワップするわけではありません。このようなスワップは、アプリケーションユーザインタフェースの静的面に集中しています。特に、複雑なアプリケーションには、アプリケーションの基盤を管理する多数のオブジェクトが入っており、基盤を動的な状況で機能させます (アプリケーションの動的面には、ウィンドウのサイズ変更、ローカライゼーション、フォントの変更などが含まれますが、他の要素も含まれます)。OPEN LOOK と Motif のツールキットでは、アプリケーション内の動的ジオメトリを処理するこれらの「マネージャウィジェット」が異なります。マネージャウィジェットをオブジェクトごとにスワップしようとする、アプリケーションは希望どおりの動的面を示しません。通常、アプリケーションの動的面の処理に適した設計を導入すると、アーキテクチャはより複雑になります。

XView では、多様な動的配置は使用できません。主として、オブジェクトは位置 (x,y) で固定されているため、アプリケーションフォントの変更や、アプリケーションの言語対応化は難しい場合があります。Motif と OLIT には、多様なジオメトリマネージャウィジェットがありますが、両者は大きく異なります。

表 6-1 OLIT と Motif のジオメトリマネージャウィジェット

OLIT	Motif	コメント
BulletinBoard	XmBulletinBoard	基本的には同等。子を静的な x,y ピクセルベースで配置できます。
Caption	(なし)	OLIT の Caption ウィジェットを使用すると、コントロールウィジェットの4面のいずれかにラベルを自動的に配置できます。 この機能を Motif に導入するには、ラベルとコントロールという2つの子が入った別の XmRowColumn ウィジェットを作成します。

表 6-1 OLIT と Motif のジオメトリマネージャウィジェット (続き)

OLIT	Motif	コメント
ControlArea	XmRowColumn	<p>どちらのウィジェットを使用しても、行と列で子を配置できます。OLIT の ControlArea ウィジェットは、コロンを使用した Caption ウィジェットの子の垂直整列をサポートしています。これは、Motif の XmRowColumn ウィジェットにはない機能です。¹</p> <p>Motif の XmRowColumn ウィジェットは、特定のサイズポリシー (通常は、特定の行または列内の子を同じサイズにする) を子に適用しますが、OLIT の ControlArea ウィジェットにはこの機能はありません。</p>
FooterPanel	(なし)	<p>OLIT の FooterPanel ウィジェットを使用すると、最下部にフローティングフッタの子が付いたウィンドウを配置できます。</p> <p>フッタとして使用できるウィジェット (XmForm または XmRowColumn ウィジェットなど) として「メッセージ領域の子」を設定すると、Motif の XmMainWindow ウィジェットをこの配置用に構成できます。</p>
Form	XmForm	<p>どちらのウィジェットを使用しても、その子を相互に相対的に接続し Form 自体に相対的に接続できますが、これらの「アタッチメント」の表示はそれぞれ異なります。OLIT の Form ウィジェットは x 軸と y 軸のアタッチメントリソースを提供しますが、Motif の XmForm ウィジェットは 4 面 (上下左右) すべてに別のアタッチメントを提供します。両方のアタッチメントパラダイムがよくわかっている場合は、OLIT の Form ウィジェットアタッチメントを同等の XmForm ウィジェットアタッチメントに変換できます。Motif の XmForm ウィジェットは、「Position」という特殊なアタッチメントタイプも提供します。Form ウィジェットのサイズが変化するたびに変化する Form ウィジェット内の動的位置に子を接続できます。これにより、Form ウィジェットの占有部分に常に一定のパーセントを占めるように子を構成できます。</p>

表 6-1 OLIT と Motif のジオメトリマネージャウィジェット (続き)

OLIT	Motif	コメント
RubberTile	(なし)	OLIT の RubberTile ウィジェットは、RubberTile ウィジェットの高さ (垂直方向の場合) または幅 (水平方向の場合) に一定のパーセントを占めるように子を構成できます。 Motif では、XmForm ウィジェット内で Position ベースのアタッチメントリソースを使用すると、同様の機能を得られません。
ScrolledWindow	XmScrolledWindow	どちらのウィジェットを使用しても、子ウィジェットをスクロール可能な表示ポートに含むことができます。
(なし)	XmMainWindow	Motif の XmMainWindow ウィジェットは、子をウィンドウの特定の領域に配置するマネージャを提供します。これらの領域には、メニューバー領域、コマンド領域、作業領域、メッセージ領域が含まれます。 OLIT には、同等のウィジェットはありません。
(なし)	XmPanedWindow	Motif の XmPanedWindow ウィジェットを使用すると、子を垂直方向の区画内に配置できます。各区画には、垂直方向にサイズ変更できるサッシが付いています。 OLIT には、同等のウィジェットはありません。

1. CDE の AppBuilder は、コロンの垂直に整列させるなど、ウィジェットを共通の位置に自動的に配置できる「グループ」というジオメトリマネージャ抽象化を提供します。この機能を実装するために、AppBuilder はすべてのコードを生成します。

ターゲット環境を学習する

CDE デスクトップは OpenWindows デスクトップとはまったく異なりますが、同じタイプのツールが多数入っています。最も明らかな違いは、CDE デスクトップは Motif デスクトップであることです。エンドユーザ環境と開発環境のアーキテクチャ構造の詳細は、『共通デスクトップ環境 プログラマ概要』を参照してください。また、次のマニュアルも参照してください。

- Motif マニュアルのリストについては、82 ページの「Motif 2.1 マニュアル」と 84 ページの「Motif プログラミング」を参照してください。

- CDE の見た目と使い心地の詳細は、『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。アプリケーションをスタイルガイドに準拠させるには、このスタイルガイドの巻末に掲載されているチェックリストに合格しなければなりません。
- 提供されている CDE マニュアルのリストについては、81 ページの「CDE マニュアル」を参照してください。また、デスクトップアプリケーションごとに、オンラインヘルプボリュームが用意されています。

GUI 開発ツール

Motif GUI ビルダを使用してアプリケーションの GUI を作成した場合は、Solaris CDE デスクトップに簡単に移行できるはずですが、ほとんどの場合、ビルダを使用するとユーザインタフェースの関数とアプリケーションの内部関数がある程度分離することになります。その長所については前述のとおりです。

通常、ビルダは汎用内部記憶形式を使用するか、相互交換ファイルを生成する機能があり、各ファイルは変換プロセスの一部を自動化するために後処理できます。ビルダのベンダーに、現在どんな移行ツールを販売しているか問い合わせてください。

その他にあまり確立されていないものの移行作業を容易にするようなツールには、機能要件に関する文書や高水準の設計を生成した開発設定が含まれます。これらは、CDE と Solaris Motif にマップするとソースコードよりも影響を受けやすい OPEN LOOK ユーザインタフェースに固有の用語を使わずに、アプリケーションを説明している場合があります。

移行ツールの使用

アプリケーションを Solaris CDE デスクトップに移行するために必要なすべての作業を単独で実行できるようなツールは販売されていません。しかし、アプリケーションによっては、いくつかの問題を解決するツールが見つかることがあります。これらの特殊ツールにより、移行作業が簡単にできます。

利用できるツールを調べる場合は、自分のアプリケーションの移行に適しているかどうかと、自分が何をしようとしているかを確認してください。

Motif に移行するためのツール

Motif に移行するためのツールが他社からも多数販売されています。たとえば、Integrated Computer Solutions (ICS) は、2 種類のコンバータを販売しています。1 つのコンバータは XView ソースコードを GIL ファイルに変換し、もう 1 つのコンバータは GIL ファイルを Motif ソースコードに変換します。

Devguide は、ユーザの移行作業を手助けする手段として Motif 変換ユーティリティを提供します。Devguide のフロントエンドは、従来どおり OPEN LOOK ユーザインタフェースで、同じ OPEN LOOK パレットが付いていますが、Motif 変換ユーティリティを使用すると、GIL ファイルを UIL ファイルや Motif コードと C コードに変換できます。Solaris 2.4 ソフトウェア開発キットの Motif 変換ユーティリティ (guil と gmf) を使用して、GIL ファイルを UIL と Motif の C コードに変換します。UIL ファイルを Motif GUI ビルダにインポートすると、Motif ベースの GUI を生成できます。

移行ツールを使用して GUI を移行する場合には、CDE スタイルガイドに準拠するように GUI に手を加える必要が出てくることがあるので注意してください。GUI ビルダツールを使用する場合は、短時間で GUI の配置を変更できます。

Solaris CDE デスクトップに移行するためのツール

Devguide を使用してアプリケーションを作成した場合は、AppBuilder の GIL から BIL へのコンバータを使用して BIL ファイルを作成できます。BIL ファイルは、AppBuilder で使用する形式です。BIL ファイル形式は GIL ファイル形式に似ていますが、BIL ファイルには CDE 固有の情報が入っており、Solaris Motif GUI を作成します。GIL から BIL へのコンバータは、既存の内容に基づいて変換を推論しますが、希望どおりの結果が得られない場合があります。いい結果を得るには、コンバータが生成した BIL ファイルを取り出して AppBuilder に読み込み、必要に応じてユーザインタフェースを変更します。

AppBuilder は、OpenWindows 環境で使用される Devguide によく似た Solaris CDE 環境のツールです。この方法で CDE に移行すると、変換時の大部分の問題は解決できます。AppBuilder によってファイルが生成されるので、これらのファイルはユーザが自分で作成しなければならない C コードよりも簡単に処理できます。AppBuilder はアプリケーションのスタックファイルを保持するので、アプリケーション固有のコードは変更されません。また、AppBuilder は Devguide と同じファイル命名規則を使用します。

Devguide 以外で OPEN LOOK アプリケーションを CDE に変換する他社のツールは、現在販売されていません。55 ページの「Motif に移行するためのツール」で説明した Motif の変換ツールなどを試してください。

既存のコードの使用

この節では、Solaris CDE アプリケーションを開発するために使用または学習するコードが、Solaris CDE ソフトウェアからどのように提供されるかについて説明します。

GUI ビルダコード

AppBuilder または SunSoft Visual Workshop のような Motif GUI ビルダを使用すると、単純な Solaris CDE アプリケーションを構築できます。次に、そのアプリケーションからコードを生成し、コードの内容を調べてください。この方法により、Solaris Motif と CDE の機能をどのように使用するかを学習できます。

AppBuilder で使用できる機能が多くなるほど、生成されたコードに多数の Solaris CDE 機能を組み込むことができます。たとえば AppBuilder を使用して、オブジェクト属性を変更したり、接続を作成したり、アプリケーションフレームワークエディタを使用したりします。次に、コードを生成して検査します。このプロセスを順番に繰り返して、コードのバリエーションを生成し、そのコードを調べて学習します。

デモコード

Solaris CDE の開発環境には、アプリケーションの移植作業を大幅に軽減できるデモソースコードが含まれています。

`/usr/dt/examples` には、開発環境の各コンポーネントごとのデモディレクトリがあります。デモディレクトリには、そのコンポーネントの API を使用するプログラム例が入っています。デモコードを読んで、コンポーネントの動作をアプリケーションに組み込む方法を学習してください。デモコードからアプリケーションにコードをコピー&ペーストできます。

`/usr/dt/examples/template` ディレクトリには、基本的な統合機能と推奨する統合機能を構成する Solaris CDE コンポーネントのほとんどを統合するデモプログラムが入っています。このテンプレートデモは、Solaris CDE デスクトップと密接に統合された単純なアプリケーションの作成方法を具体的に示しています。

ヒント

次に、CDE デスクトップ用のアプリケーションを作成するためのヒントについて説明します。

フローティングメニュー

アプリケーションが CDE デスクトップ上で正常に機能することを保証するには、一部のプルダウンメニューにもフローティングメニューに組み込んだ機能を持たせる必要があります。こうすると、アプリケーションは 2 ボタンマウスデバイスと 3 ボタンマウスデバイスのどちらでも機能するようになります。

カラーマップの動作

CDE 上では、カラーマップのインストールが OpenWindows 環境とは異なる方法で処理されます。この違いは、デフォルト以外のカラーマップを使用するサブウィンドウを指定するアプリケーションで識別できます。このサブウィンドウは、WM_COLORMAP_WINDOWS 属性で指定されます。

OpenWindows 環境では、アプリケーションはこの属性内でサブウィンドウを指定するだけですみます。ユーザが画面内でポインタを移動すると、OpenWindows のウィンドウマネージャ (o1wm) は、ポインタが置かれているウィンドウ用に適切なカラーマップをインストールします。

CDE のウィンドウマネージャである dtwm は、この動作を行いません。o1wm のポインタベースのカラーマップインストール機能に依存するアプリケーションを CDE 上で実行する場合は、正しい色が表示されないことがあります。この問題を回避するために、次のいくつかの対処方法があります。

- WM_COLORMAP_WINDOWS 属性をまったく使用せずに、適切なカラーマップを持つトップレベルウィンドウ (シェルウィジェットのウィンドウ) のカラーマップ属性を更新します。ウィンドウに入力フォーカスが与えられているときは、いつでもこのカラーマップがインストールされます。
- WM_COLORMAP_WINDOWS 属性はウィンドウの並び順のリストであり、通常はこのリスト内の最初のウィンドウにのみカラーマップがインストールされ、正しい色で表示されます。ほとんどの場合、他のサブウィンドウは正しい色で表示されません。特定のサブウィンドウを正しい色で表示することが重要な場合は、リストの最初に配置する必要があります。
- トップレベルウィンドウの ID がリストに表示されない場合は、最初に表示されるものと見なされます。サブウィンドウを正しい色で表示するには、リスト内でそのサブウィンドウの ID の後にトップレベルウィンドウの ID を配置する必要があります。
- アプリケーションは、いつでも WM_COLORMAP_WINDOWS 属性を更新できます。別のサブウィンドウにそのカラーマップがインストールされるなど、アプリケーションの状態が変化する場合は、新しいサブウィンドウが最初に表示されるようにアプリケーションは属性を更新する必要があります。

また、dtwm のキーの割り当てを変更して、キーボードのキーに `f.next_cmap()` 関数と `f.prev_cmap()` 関数を設定できます。該当するキーを押すたびに、これらの関数は WM_COLORMAP_WINDOWS リスト内を前後に移動して別のカラーマップをインストールします。

まとめ: 注意事項

次に、アプリケーションを Solaris CDE デスクトップに移植する場合の注意事項を示します。

- OPEN LOOK から Motif に移行する作業は、複雑で広範囲にまたがる問題を伴います。
通常は、オブジェクトごとのスワップにはなりません。実際に移植する必要がありますか。
- どんなスケジュールで作業しますか。
いつ出荷しなければなりませんか。どんなリソースを使用できますか。移植結果を次の主要リリースに結び付けることができますか。
- どのような方法でアプリケーションを構築しましたか。
AppBuilder を使用しましたか。アプリケーションの生成の際に他のツールを使用しましたか。使用した場合は、CDE のサポートをベンダに問い合わせてください。
- アプリケーションの GUI は内部から分離されていますか。
C++ オブジェクトを使用して GUI をカプセル化しましたか。その場合は、アプリケーションを簡単に移植できるようにします。
- XView と OLIT のどちらから作業を始めますか。
通常、OLIT のアプリケーションは Motif のように Xt イントリンシクスに依存しているので、OLIT アプリケーションの方が簡単に移植できます。
- カスタマには何か問題がありますか。
他のシステムとの相互運用性を考慮する必要がありますか。カスタマには、移行と研修に際して問題がありますか。現在、混在型のデスクトップは受け入れられますか。

第 7 章

移植例: OPEN LOOK から Solaris Motif へ

この章では、単純なグラフィカルユーザインタフェース (GUI) を持つ OPEN LOOK アプリケーションを Solaris Motif に移植する場合の例について説明します。この例は、2 段階の移植プロセスを示しています。まず、OPEN LOOK ユーザインタフェースオブジェクトを 1 つずつ Motif オブジェクトに変換してから、インタフェースを整列し、CDE スタイルガイドに準拠していることを確認します。

- 61 ページの「OpenWindows 3.4 スナップショットアプリケーション」
- 62 ページの「変換」
- 63 ページの「整列」

OPEN LOOK ユーザインタフェースから Solaris Motif への移植作業の目標は、GUI を変換すると同時に整列することです。CDE スタイルガイドに慣れてくれば、この作業は簡単に実行できるようになります。また、Motif に関する知識が増えれば、OPEN LOOK オブジェクトと Motif ウィジェットの相関関係も簡単に理解できるようになります。

アプリケーションを移植する場合は、そのユーザインタフェースを調べる機会を設けてください。簡素化できるかどうかや、よりユーザに親しみやすいものにならないかを考えてください。OPEN LOOK ユーザインタフェースから Motif に移行すると、アプリケーションの見た目は変化します。これは、過去に行なった設計上の決定を見直すよい機会になります。

OpenWindows 3.4 スナップショットアプリケーション

図 7-1 は、OpenWindows 3.4 スナップショットアプリケーションを示しています。このアプリケーションは、Motif ウィンドウマネージャを使用して Solaris CDE デスクトップ上で実行されるので、Motif のタイトルバーが付いています。

スナップショットの GUI はきわめて単純で、次の項目が含まれます。

- 「読み込み (Load)...」、「保存 (Save)...」、「スナップ (Snap)」、および「表示 (View)...」ボタン
- 「印刷 (Print)」メニューボタン
- 「スナップ領域 (Snap Type)」と「遅延秒数 (Snap Delay)」の排他的設定
- 「ピープ音 (Beep)」と「ウィンドウを隠す (Hide)」のチェックボックス設定
- 「ドロップ (Drop)」ターゲット

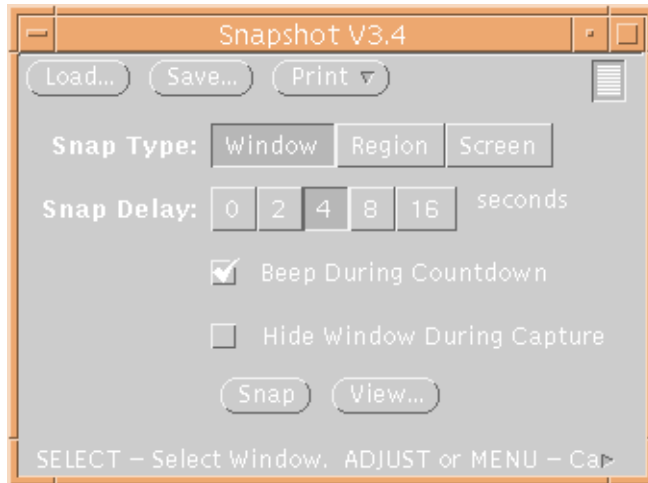


図 7-1 OpenWindows 3.4 スナップショットアプリケーションの GUI

変換

AppBuilder を使用すると、スナップショットの GUI がオブジェクトごとに図 7-2 の GUI に変換されます。

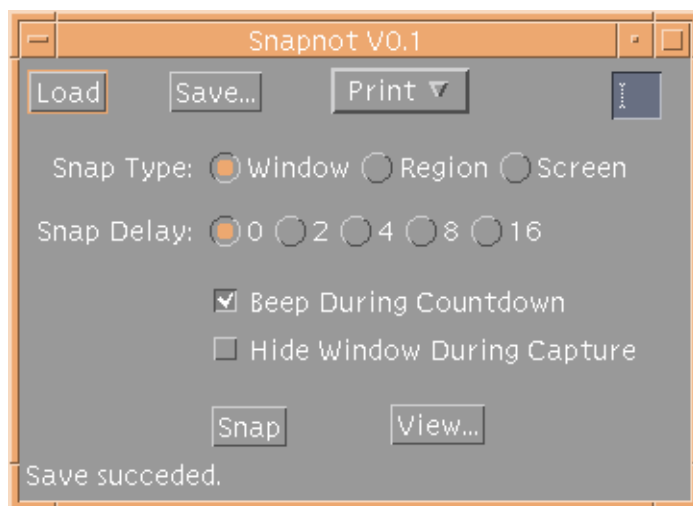


図 7-2 スナップショットの GUI を Motif に移植したもの (直接変換)

インタフェースの各部分と配置全体が、OPEN LOOK バージョンによく似ていることがわかります。ボタンの形が丸型ではなく四角型になったことを除くと、ボタンは同じです。チェックボックスの見た目は同じです。OPEN LOOK の排他的設定は、Motif のラジオボタンに変更されています (同様に相互に排他的な選択肢を提供します)。ドロップターゲットは、ラベルなしの小さいテキストフィールドに置き換えられています。

整列

このアプリケーションの移植作業をユーザが終了することもできます。機能は失われず、GUI は現在 Motif に表示されています。ただし、次の 2 つの点を確認する必要があります。

- 移植後のアプリケーションはスタイルガイドに準拠しているか
- GUI の設計を変更して、ユーザがより親しみやすいアプリケーションにできるか

CDE スタイルガイドラインを検討する

『共通デスクトップ環境 スタイル・ガイド』では、アプリケーションに複数のメニューバーを配置することを強く推奨しています。また、「ファイル (File)」、「オプション (Options)」、「ヘルプ (Help)」のメニューが、標準では左から右に配置するとしています (他のメニューも標準と見なされますが、スナップショットアプリケーションには適用されません)。

整列後のバージョンでは、「ファイル (File)」メニューの下に「読み込み (Load)」、「保存 (Save)」、および「印刷 (Print)」ボタンが配置されています。これらは、典型的なファイル操作です。「秒読み中に警告音を出す (Beep During Countdown)」と「スナップ中にウィンドウを隠す (Hide Window During Capture)」はユーザが使用できるオプションなので、「オプション (Options)」メニューの下にあります。これらのチェックボックスのテキストは、画面上で大量のスペースを占めるので、この設計上の決定によってインタフェースがすっきりします。スタイルガイドラインに準拠するために、「ヘルプ (Help)」メニューが追加されています。

『共通デスクトップ環境 スタイル・ガイド』では、アプリケーションの最下部にステータスが表示されるときにはいつも、そこにフッタメッセージを使用することを求めています。これは、スナップショットの GUI に追加されています。

Motif では、ドロップ領域のオブジェクトはサポートされていません。CDE ではドラッグソースオブジェクトがサポートされています。オブジェクトはラベル付きのカメラアイコンとして最終の GUI に表示されます。CDE スタイルガイドラインは、ドロップサポートが必要な場合、アプリケーションのメインウィンドウ内のどこにでもオブジェクトをドロップ可能にするコードを用意しなければならないことを示しています。

最後に、最下部の「スナップ (Snap)」ボタンが強調表示されるときは、ユーザがそのボタンではなく Return キーを押した場合のデフォルト値 (「スナップ (Snap)」と「表示 (View)」の間) であることを示します。

他の設計上の注意事項

スナップショットの GUI で、ラジオボタンが画面上で大量のスペースを占めます。CDE は、必要な画面スペースが少なく、同じ機能を持つ DtComboBox というウィジェットを提供します。「スナップ領域 (Snap Type)」と「遅延秒数 (Snap Delay)」のラジオボタンは、DtComboBox ウィジェットに置き換えられます。最も一般的に使用される値が、DtComboBox のデフォルトとして表示されます。

OpenWindows デスクトップ用のアプリケーションを開発する場合は、DtComboBox ウィジェットを使用できません。このウィジェットは、OpenWindows 開発環境の一部ではなく、CDE 開発環境の一部です。コンボ・ボックスにアクセスできなくても、人間工学、カスタマのニーズ、整合性などの問題に応じて、ラジオボックスを保持することもできます。

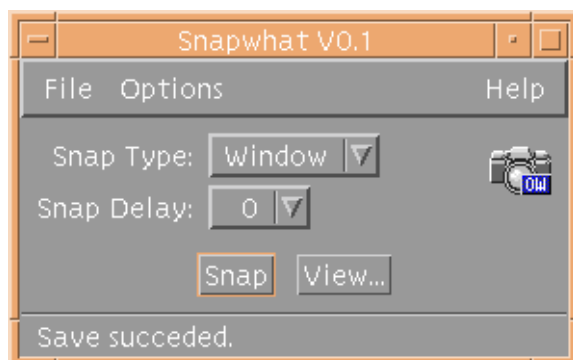


図 7-3 スナップショットの GUI を Motif に移植したもの (最終移植)

付録 A

GUI 環境の変更

この付録では、OPEN LOOK スタイルと CDE スタイルとの間で変更された、すべてのグラフィカルユーザインタフェース (GUI) 環境について説明します。ここで取り上げていない内容は、変更されていません。

この付録の情報を利用して、CDE アプリケーションに正しい GUI 環境を提供してください。

表 A-1 GUI 環境の変更点

概要	内容	OPEN LOOK	CDE
デスクトップ	複数のデスクトップ	1つのメインワークスペース、または仮想、複数の画面のサポート (OLVWM)	フロントパネルから複数のワークスペースにアクセスできる。ワークスペースの追加、削除、ナビゲートができる
	アクションのフィードバック	アプリケーションの起動時にフィードバックの提供を示すインジケータがない	フロントパネル上にインジケータライトがあり、アプリケーションの初期起動時や他のアクションの発生時に点滅する
	アプリケーションのアクセス	アプリケーション (とユーティリティ) は、「ワークスペース」メニューの「プログラム」からアクセス可能。「プログラム」メニューは GUI によりカスタマイズ可能	アプリケーションは、フロントパネル、アプリケーションマネージャ、個人アプリケーションサブパネル、「ワークスペース」メニューの「プログラム」サブメニューからアクセス可能。「プログラム」メニューをカスタマイズする GUI はない
	デスクトップへのドラッグ&ドロップ	ファイルマネージャからアイコンをデスクトップヘドロップすると、そのアイコンに関連付けられたアプリケーションが起動する	ファイルマネージャからアイコンをデスクトップヘドロップしても、アプリケーションは起動しない。そのアイコンへの参照をデスクトップ上に作成する (元のアイコンはファイルマネージャ上に残る)

表 A-1 GUI 環境の変更点 (続き)

概要	内容	OPEN LOOK	CDE
	高速ヘルプ情報	「ヘルプ...」を使用する。詳細情報は「ヘルプハンドブック」にリンクしている	「ヘルプの使い方」を使用してヘルプハンドブックの項目に移動する
	アプリケーションヘルプ	アプリケーションで Help キーを押す	ほとんどのメニュー・バー (一番右) とダイアログボックスから表示可能。アプリケーションで Help キーを押しても表示可能
*****	*****	*****	*****
設定	プロパティを意味する複数の用語	「プロパティ」という用語 (「設定」の場合もある) を使用。通常は「編集」メニューにある	「オプション」と「属性」という用語を使用 オプションとはアプリケーション固有の特性を意味し、通常は「ファイル」または「オプション」メニューにある 属性とは、オブジェクト固有の特性を設定する場合に使用し、通常は「編集」メニューにある
	グローバルなデスクトップオプション	「ワークスペース」メニューから設定する	「スタイル・マネージャ」アプリケーションから設定する
*****	*****	*****	*****
ウィンドウ	ウィンドウのタイトル	通常は次のような書式になります。 ファイルマネージャ V3.4: /home/username/Project_Folder	通常の手書式: ファイルマネージャ - Project_Folder バージョン番号はタイトルに含まれない (アプリケーションの「xx について」ボックスに含まれる場合がある) タイトルの書式は『共通デスクトップ環境 スタイル・ガイド』を参照
	ウィンドウの移動	ウィンドウ枠の任意の場所 (隅を除く) をクリックし、ドラッグしてウィンドウを移動する	ウィンドウのタイトル領域をクリックし、ドラッグしてウィンドウを移動する
	ウィンドウの拡大	枠をダブルクリックすると上下に拡大する。ウィンドウの位置はそのまま	「最大表示」ボタンを押すとウィンドウ全体に拡大する。デフォルトは全画面に拡大する。上下方向だけを変更するには、 Dtwm*maximumClientSize:vertical リソースを使用する

表 A-1 GUI 環境の変更点 (続き)

概要	内容	OPEN LOOK	CDE
	ウィンドウのサイズ変更	ウィンドウの隅をクリックし、ドラッグしてサイズを変更する	ウィンドウ枠の任意の部分 (隅も含む) をクリックし、ドラッグしてウィンドウのサイズを変更する。変更する大きさに外枠が表示され、そのサイズに変更される
	ウィンドウを手前に表示	タイトルバーまたはウィンドウ枠をクリックする	ウィンドウの任意の場所をクリックする
	ウィンドウの削除または消去	プッシュピンをクリックして消去する	「ウィンドウ」メニューから「閉じる」を選択するか、Alt + F4 キーを押す。CDE で実行される OPEN LOOK アプリケーションにはプッシュピンがないため、この方法で消去する
	ウィンドウのアイコン化	「ウィンドウ」メニュー (ウィンドウの左上隅にあるボックス内の矢印) から「クローズ」を選択する。 「クローズ」は「ウィンドウ」メニューのデフォルトなので、「ウィンドウ」メニューのアイコン上で1回クリックするとウィンドウを閉じる (「プロパティ」のデフォルトアクションが「左マウスボタン: デフォルト項目の選択」に設定されている場合)	「ウィンドウ」メニューから「アイコン化」を選択するか、ウィンドウ枠の左上隅の「アイコン化」ボタンをクリックする
	アプリケーションの終了	「ウィンドウ」メニューで「終了」を選択する	「ウィンドウ」メニューまたは「ファイル」メニューで「閉じる」または「終了」コマンドを選択する。または、「ウィンドウ」メニューをダブルクリックしてアプリケーションを終了する
*****	*****	*****	*****
メニュー	メニュー配置の種類	メニューボタンを使用する	メニューバーを使用する
	メニュー配置 (標準)	「ファイル」 - 「表示」 - 「編集」 - 「プロパティ」	「ファイル」 - 「編集」 - 「表示」 - 「オプション」 - 「ヘルプ」
	デスクトップ上にメニューを残す	メニューのプッシュピンをクリックする	ティアオフメニューがある

表 A-1 GUI 環境の変更点 (続き)

概要	内容	OPEN LOOK	CDE
	メニューからのデフォルトアクション	「ワークスペース」メニューの「プロパティ」で「左マウスボタン: デフォルト項目の選択」を設定できる。 このモードでは、メニューボタンを1回クリックするとデフォルトのメニュー項目が選択される。デフォルト項目はメニューの中で囲まれている	デフォルトのメニュー項目はない
	クイックメニューアクセス	簡易ポップアップメニューが使用できるコマンドを表示する。マウスボタン3(メニュー)をクリックして表示する	左記に同じ
	キーボードからのメニューのアクセス	メニューにアクセスするキーボード二モニックはない	キーボードを使用してメニューから選択できる。メニューの二モニックは、メニュー名にある下線文字(メニュー名の最初の文字ではない場合もある)で表示される。メニューは Alt キーと下線文字を同時に押すと表示される。特定のメニューオプションにアクセスするには、二モニックのキーを押すか、下向きの矢印キーを押して項目へ移動し、Return キーまたはスペースバーを押す
*****	*****	*****	*****
ダイアログボックス	ファイル選択(ファイル名を入力するか、スクロールリストから選択する)	通常、ユーザがファイル名の入力や選択ができるコマンド・ウィンドウが提供される(「オープン」、「取り込み」、「保存」など)	標準の CDE ファイル選択ダイアログボックスが提供される。アプリケーション固有のファイルナビゲーションや選択オプション用に変更可能。例とガイドラインについては、『共通デスクトップ環境 スタイル・ガイド』を参照
	印刷ダイアログボックス	アプリケーションによって異なる	アプリケーション固有の印刷オプション用に変更できる共通の印刷ダイアログボックスが提供される。例とガイドラインについては、『共通デスクトップ環境 スタイル・ガイド』を参照

表 A-1 GUI 環境の変更点 (続き)

概要	内容	OPEN LOOK	CDE
	ダイアログボックスの削除または消去	プロパティウィンドウは、プッシュピンとボタンを両方使用するダイアログボックスの1つとして機能する。「適用」ボタンは変更を反映し、ピンで留められていないプロパティウィンドウを消去する。ピンで留められているプロパティウィンドウは消去しない。ピンで留められているウィンドウを消去するには、プッシュピンをクリックする	ボタンアクションは、アクション実行後にダイアログボックスを残すかどうかを決定する。「取消し」は、まだ適用されていないアクションを実行せずにダイアログボックスを消去する。「閉じる」はダイアログボックスを消去する。「了解」は変更を反映し、ダイアログボックスを消去する。「適用」は変更を反映し、ダイアログボックスを消去しない。OPEN LOOK アプリケーションが CDE で実行されている場合は、プッシュピンは使用されない。詳細は、『共通デスクトップ環境 スタイル・ガイド』を参照
	*****	*****	*****
コントロール	相互に排他的な選択コントロール	「排他的設定」コントロールを使用して相互に排他的な選択を提供する。角の丸い四角形表示になる	ラジオボタンを使用する。ガイドラインは『共通デスクトップ環境 スタイル・ガイド』を参照。丸いボタン表示になる
	関連する項目と値を選択するコントロール	メニュー全体ではなく、現在の選択だけを表示する簡易メニューボタンを使用する。ボタンメニューから値を選択するか、テキストフィールドに値を入力して、Return キーを押して確定する	コンボ・ボックス、コマンドボックス、またはスピン・ボックスを使用して省略メニュー・ボタンと同じ機能を実行する。コンボ・ボックスはスクロール可能なリスト・ボックスを使用し、コマンドボックスはテキスト入力用プロンプトとリストコンポーネントを使用し、スピン・ボックスは項目を連続して表示する。説明とガイドラインについては、『共通デスクトップ環境 スタイル・ガイド』を参照
	ツールバー	ツールバーは使用しない	ツールバーを使用する
	ボタンの形	丸いボタン	四角いボタン
	*****	*****	*****
マウス	デフォルトのマウスボタン機能	マウスボタン 1 はセレクト マウスボタン 2 はアジャスト マウスボタン 3 はメニュー	マウスボタン 1 はセレクト マウスボタン 2 は転送(スタイルマネージャによりアジャストに変更可能) マウスボタン 3 はメニュー

表 A-1 GUI 環境の変更点 (続き)

概要	内容	OPEN LOOK	CDE
	範囲選択	3 ボタンマウスの場合、マウスボタン 2 (アジャストボタンとも呼ぶ) を使用して範囲を選択する。 選択するテキストの最初でマウスボタン 1 をクリックし、選択範囲の最後までポインタを移動し、マウスボタン 2 をクリックする。テキストが強調表示される	Shift キーを押しながらマウスボタン 1 を押す。または、スタイルマネージャによりマウスボタン 2 の動作を転送 (デフォルト設定) からアジャストに変更する
	メニューの表示	メニューボタン上でマウスボタン 3 をクリックして、メニューを表示する	メニューバーエントリをマウスボタン 1 または 3 でクリックして、メニューを表示する
	アイコンのオープン: ショートカット	アイコンをダブルクリックして開く	アイコンをダブルクリックして開く。フロントパネルのコントロールを 1 回またはダブルクリックし、スタイルマネージャのコントロールを 1 回クリックして、関連するアプリケーションを起動する
	クイックカット	テキストを挿入点に移動するには、Cut キーを押しながらテキストを選択し (1 本の線が選択したテキストに引かれる)、Cut キーを離す。キーを離すと、テキストが挿入点に移動する	テキストを挿入点に移動するには、Alt キー、Shift キー、マウスボタン 2 (転送) を同時に押し、テキストを選択する (1 本の線が選択したテキストに引かれる)。キーを離すと、テキストは挿入点に移動する
	クイックペースト	テキストを挿入点にコピーするには、Paste キーを押しながらテキストを選択し (1 本の下線が選択したテキストに引かれる)、Paste キーを離す	テキストを挿入点にコピーするには、Alt キーとマウスボタン 2 (転送) を同時に押し、テキストを選択する (1 本の線が選択したテキストに引かれる)。キーを離すと、テキストは挿入点にコピーされる (Alt キー、Control キー、マウスボタン 2 を同時に押ししても同じ)
*****	*****	*****	*****
ドラッグ&ドロップ	ドラッグ&ドロップ (直接操作する場合)	ドラッグ&ドロップし、視覚的に確認する	基本的なドラッグ&ドロップは OPEN LOOK と同じだが、仕様が異なる。ドラッグ&ドロップ操作、性能のガイドライン、フィードバック、CDE 固有のグラフィックスについては、『共通デスクトップ環境 スタイル・ガイド』を参照

表 A-1 GUI 環境の変更点 (続き)

概要	内容	OPEN LOOK	CDE
	アイコンへのドロップ	デスクトップ上のアイコンへドロップできる。ドロップにより適切なアクションが実行される	サポートされていない。 フロントパネルのコントロールにドロップできる (たとえば、プリンタ、カレンダーマネージャ、ゴミ箱、テキストエディタ、メールプログラムなど)
	メールアプリケーションからファイルマネージャへ複数のメッセージをドラッグ&ドロップする	ファイルマネージャで、メッセージごとに1つのメールファイルのアイコンを作成する	選択したすべてのメッセージを含む1つのメールボックスを作成する
	メールメッセージを他のメールボックスウィンドウにドラッグする	利用できない	選択したメッセージを目的のメールボックスへ移動する
	選択したアイコンをテキストウィンドウへドロップする	テキストがドロップポイントに表示される	テキストが挿入点に表示される (「ファイル」メニューから「取込み」を選択するのと同じ)
	ドロップゾーン (ターゲット)	ドロップ・ゾーン (ターゲット) は、アプリケーションのコントロール領域の右上に通常表示される小さい四角形	特別なドロップゾーングラフィックスは使用されない。編集可能なコントロールはドロップゾーンとして機能できる。ユーザはドロップゾーンの有効性についてフィードバックを得る
	ドラッグソース	ドラッグ&ドロップソースインジケータグラフィックスを使用する	ソースインジケータの複数のバージョンが使用され、選択の種類 (1つか複数) と選択したオブジェクトの種類を示す。ドラッグアイコンは動的に作成され、ソースインジケータ情報を含む
*****	*****	*****	*****

表 A-1 GUI 環境の変更点 (続き)

概要	内容	OPEN LOOK	CDE
キーボード	キーボードアクセラレータとバインド	<p>Meta キーをアクセラレータキーとともに使用する。例: Meta + z キーは「最後の編集を取り消す」、Meta + c キーは「コピー」、Meta + v キーは「ペースト」、Meta + x キーは「カット」。</p> <p>キーバインディングのデフォルトセットを使用し、キーバインディングのカスタマイズを可能にする (たとえば、EMACS など)</p>	<p>Control キーをアクセラレータキーとともに使用する。例: Control + z キーは「元に戻す」、Control + c キーは「コピー」、Control + v キーは「ペースト」、Control + x キーは「カット」。</p> <p>デフォルトのキーバインディングは、OPEN LOOK のキーバインディングと異なる場合がある (キーのバインディングについては、『共通デスクトップ環境 スタイル・ガイド』の付録 A 「キーボードの機能」を参照)。カスタマイズも可能 (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)</p>
	Sun の特殊キー	OPEN LOOK でサポートされる	Solaris CDE でサポートされる
	文字の削除	Backspace または Delete キーを使用できる	Backspace キーはカーソルの左の文字を削除するDelete キーはカーソルの右の文字を削除する

国際化と CDE

付録 B は、国際化された Solaris CDE アプリケーションを作成するソフトウェア開発者向けのガイドです。この付録では、OpenWindows と CDE の国際化の違いを示します。Solaris CDE アプリケーションを国際化するために実行しなければならない手順のうち、この付録で説明していないものは OpenWindows のガイドラインから変更されていません。CDE アプリケーションの国際化の詳細は、『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)』を参照してください。

- 75 ページの「正しい CDE NLS 環境を保証する」
- 76 ページの「メッセージカタログ関数」
- 76 ページの「ロケール通知」
- 77 ページの「文字列と XmString」
- 77 ページの「app-defaults ファイルを組み込む」
- 78 ページの「Motif リソースの現地仕様化」
- 78 ページの「メッセージカタログの配信」
- 79 ページの「フォント」
- 79 ページの「国際化されたシェルスクリプト」

Solaris CDE は Solaris 7 がサポートするすべてのロケールをサポートします。

この付録は、Xt と Motif のプログラミングに精通している方を対象としています。

正しい CDE NLS 環境を保証する

必ず環境変数 NLSPATH が正しく設定されているかどうかを確認してください。これにより、アプリケーションがメッセージカタログを見つけることが保証されます。

アプリケーションが DtInitialize() を呼び出す場合、NLSPATH はメッセージカタログを検索するため CDE のデフォルト位置に設定されます。デフォルト位置は、`/<CDE_INSTALLATION>/lib/nls/msg/<locale>/<application>.cat` です。たとえば、`/usr/dt/lib/nls/msg/ja/dtcm.cat` となります。

DtInitialize() はその他の変数を設定し、その他のタスクを実行します。この関数が必要かどうかは、対応するマニュアルページを参照してください。
DtInitialize() は libDtSvc にあります。アプリケーションがまだこのライブラリにリンクされていない場合、NLSPATH を設定する目的だけで DtInitialize() を呼び出す必要はありません。

アプリケーションが DtInitialize() を呼び出さない場合、アプリケーションがメッセージカタログを検索できるように NLSPATH を設定しなければなりません。

デフォルト以外の場所にアプリケーションがメッセージファイルをインストールするか、その存在を想定している場合、ファイルの場所を NLSPATH に追加してください。

注 - NLSPATH を完全にリセットした場合、他の CDE アプリケーションはメッセージカタログを検索できないことがあります。メッセージファイルの場所がデフォルトと異なる場合は、NLSPATH に追加するのがより安全な方法です。

メッセージカタログ関数

他の XPG4 UNIX プラットフォームに移植できるようにするには、Solaris CDE アプリケーションは (gettext() などの代わりに)、catopen()、catclose()、catgets() ファミリの XPG4 メッセージ関数を使用しなければなりません。これらの関数の使用方法については、マニュアルページを参照してください。

また、アプリケーションを作成しようとするシステムがヘッダファイルで NL_CAT_LOCALE を定義していない場合は、アプリケーションで定義しなければなりません。これにより、アプリケーションを UNIX XPG4 準拠の他のプラットフォームに移植できることが保証されます。

ロケール通知

どのロケールで実行中かアプリケーションで認識されるようにするには、次の関数を使用します。

- XtSetLanguageProc() - アプリケーションが Motif/Xt GUI ベースの場合
- setlocale() - アプリケーションが GUI ベースでない場合

文字列と XmString

この節では、文字列と XmString の間で変換する方法について説明します。

文字列から XmString への変換

Motif では、明示的な呼び出しを行なって文字列をその内部表現である XmString に変換しなければなりません。アプリケーションを国際化する場合は、`XmStringCreateLocalized()` を使用して、この変換処理を実行します。

Solaris Motif では Motif 1.2.3 と違って、`XmStringCreateLocalized()` で `\n` が行セパレータとして認識されるので、1つの文字列に複数行を使用できます。

XmString から文字列への変換

XmString から文字列に変換するには、複合文字列を移動してセグメントごとに検索しなければなりません。この変換処理を実行するコード例については、『*Motif* プログラミング・マニュアル』(ソフトバンク発行)のボリューム 6A の第 19.3.3 節を参照してください。

XmString から文字列へ変換する場合は、`XmStringGetLtoR()` を使用しないでください。この関数は、テキストは左から右に向かうと見なします。

app-defaults ファイルを組み込む

アプリケーションがリソースを使用する場合は、必ず `app-defaults` ファイルを組み込んでください。`app-defaults` ファイルの設定方法の詳細は、『*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』(ソフトバンク発行)のボリューム 4 の第 2.3.3 節を参照してください。

Motif リソースの現地仕様化

特定のロケール向けにカスタマイズしなければならない Motif リソースを `app-defaults` ファイルに入れて、そのリソースを現地仕様化します。次のリソースが含まれますが、これ以外にもあります。

- `XmNmnemonic`
- `XmNfontList`
- `XmNcolumns`
- `XmNinputMethod`
- `XmNpreeditType`

メッセージカタログの配信

どの SVR4 UNIX プラットフォームにも、`gencat` ユーティリティが含まれています。変換後のメッセージカタログファイルで `gencat` を実行して、バイナリ形式のメッセージカタログを生成します。生成後のファイルは、フォーマットされたメッセージデータベースとなります。

CDE と `gencat`

各プラットフォームには、`gencat` が固有に実装されています。各メッセージカタログのフォーマットが原因となって異なる `gencat` ユーティリティが破損しないかどうかを、次のメッセージカタログ規則に従って確認してください。

1. スペースが後ろに付いている場合は、`$quote` 宣言を使用します (`gencat (1)` のマニュアルページを参照してください)。
2. メッセージ ID とメッセージ文字列の間には複数のスペースを入れないでください。
3. `$` とコメントの間にはスペースを 1 個挿入します。
4. メッセージ ID を昇順でソートする必要があります。

.msg ファイル

.msg ファイルは、アプリケーションコードに表示される変換可能テキストが入ったメッセージカタログです。これは、翻訳者に渡す翻訳用のファイルです。その形式については、`gencat (1)` のマニュアルページを参照してください。`gencat` を使用して、変換後のメッセージカタログをメッセージデータベースにします。

.cat ファイル

.cat ファイルは、.msg ファイル上で `gencat` を実行した結果生成されるメッセージカタログです。これはバイナリファイルです。

フォント

OpenWindows 環境では、アプリケーションのロケールに応じてフォントの別名が異なります。Solaris CDE では、フォント別名はロケールに依存しません。CDE フォント名の詳細は、『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)』または『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。

国際化されたシェルスクリプト

`dtdspmsg` ユーティリティはシェルスクリプトの国際化に役立ちます。`dtdspmsg` コマンドは、メッセージカタログから、選択したメッセージを表示します。詳細は、`dtdspmsg (1x)` のマニュアルページを参照してください。

付録 C

関連文書

この付録では、OPEN LOOK、Motif、Solaris CDE のアプリケーション開発に関連する情報が掲載されているマニュアルと記事を紹介します。マニュアルは、SunExpress または最寄りの書店でお求めになります。

- 81 ページの「CDE マニュアル」
- 82 ページの「ToolTalk マニュアル」
- 82 ページの「Motif 2.1 マニュアル」
- 83 ページの「グラフィカルユーザインタフェース (GUI)」
- 84 ページの「Motif プログラミング」
- 84 ページの「OPEN LOOK プログラミング」
- 85 ページの「Xt/XLib プログラミング」

CDE マニュアル

Solaris CDE には、このマニュアル以外に次のマニュアルがあります。

開発環境

- 『共通デスクトップ環境 プログラマ概要』
- 『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』
- 『Common Desktop Environment: Product Glossary』

すべての開発環境のコンポーネントに関して、オンラインマニュアルページを入手できます。

実行環境

- 『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への移行』

実行環境のコンポーネントのほとんどについて、オンラインヘルプボリュームを入手できます。

ToolTalk マニュアル

- 『ToolTalk サービスリファレンスマニュアル』, 株式会社アスキー発行, ISBN 4-7561-0290-5
- 『ToolTalk and Open Protocols: Inter-Application Communication』, Astrid Julienne および Brian Holtz 著, SunSoft Press および PTR Prentice Hall 社発行, Englewood Cliffs, NJ 07632, ISBN 013-031055-7

Solaris 8 Software Developer Collection - Japanese AnswerBook™ セットには、次のマニュアルが含まれています。

- 『ToolTalk ユーザーズガイド』

Motif 2.1 マニュアル

この節で紹介するマニュアルは、Motif 2.1 インタフェースの解説書です。これらのマニュアルは、最寄りの書店でお求めになれます。

- 『OSF Application Environment Specification (AES) User Environment Volume, Revision C』, PTR Prentice Hall 発行, 1993.

Motif アプリケーション環境仕様を説明しています。これらの仕様に関する情報は、『Motif 2.1 Programmer's Reference Manual』にも掲載されています。

- 『CDE 2.1/Motif 2.1 User's Guide』 The Open Group (www.opengroup.org)

Motif ユーザ環境および共通デスクトップ環境 (CDE) の基本的な機能について説明しています。これらの機能や Motif および CDE ベースのアプリケーションとの対話の方法、また Motif 環境や CDE のカスタマイズの方法、さらに Information

Manager を使用してオンラインマニュアルを読み、検索する方法などについて説明しています。

- 『*Motif 2.1 - Programmer's Reference Manual*』 The Open Group (www.opengroup.org)

UNIX のマニュアルページ形式で、Motif コマンドと機能に関するリファレンス情報を提供しています。このリファレンス情報には、ツールキット (ライブラリ関数、ウィジェットのマニュアル、リソース)、ウィンドウマネージャ、ユーザインタフェース言語コマンド、ライブラリ関数があります。

- 『*Motif 2.1 - Programmer's Guide*』 The Open Group (www.opengroup.org)

これら 3 冊は、すべての Motif プログラム、Xt ウィジェットクラス、Xm ウィジェットクラス、翻訳、Xm データタイプおよび関数、Mrm 関数、Uil 関数、ファイルフォーマットについて、詳細なリファレンスを提供しています。

- 『*CDE 2.1/Motif 2.1 - Style Guide*』 The Open Group (www.opengroup.org)

新製品の設計と実装を行う開発者に、共通デスクトップ環境 (CDE) ユーザインタフェースに適応する、実行仕様のフレームワークを提供します。この実行動作は、現在のさまざまな動作モデルから共通の要素を抽出することで確立されます。このマニュアルには、共通デスクトップ環境ドキュメントセットで使用する用語を網羅した用語集があります。

グラフィカルユーザインタフェース (GUI)

『*OPEN LOOK to Motif GUI Transition Guide*』 801-6567-10, SunSoft 発行, 1993.

『*OPEN LOOK Graphical User Interface: Programmer's Guide*』, UNIX System Laboratories 発行, 1992.

『*OPEN LOOK Graphical User Interface: User's Guide*』, UNIX System Laboratories 発行, 1992.

『*OPEN LOOK スタイルガイド Graphical User Interface Application Style Guidelines*』, 日本サン・マイクロシステムズ株式会社, アジソン・ウェスレイ発行, 1990.

『*OPEN LOOK ファンクショナルスペック Graphical User Interface Functional Specification*』, 日本サン・マイクロシステムズ株式会社, アジソン・ウェスレイ発行, 1990.

『*OPEN LOOK Graphical User Interface: Programmer's Reference Manual*』, Prentice Hall 発行, 1992.

『*OPEN LOOK Intrinsics Toolkit Widget Set Programmer's Guide*』, AT&T, 1990.

『*OPEN LOOK Intrinsics Toolkit Widget Set Reference Manual*』, AT&T, 1990.

『OSF/Motif プログラマーズ・ガイド リリース 1.2』, Open Software Foundation, トッパン発行, 1993.

『OSF/Motif プログラマーズ・リファレンス リリース 1.2』, Open Software Foundation, トッパン発行, 1993.

『OSF/Motif スタイル・ガイド リリース 1.2』, Open Software Foundation, トッパン発行, 1993.

『OLIT & Motif: A Technical Comparison』, Amy Moore および M. Goyal 著, SunSoft 発行, 1992.

Motif プログラミング

『Motif Programming in the X Window System Environment』, William A. Parrette 著, McGraw-Hill 発行, 1993.

『Motif Programming: The Essentials—and More』, Marshall Brain 著, Digital Press 発行, 1992.

『Motif プログラミング・マニュアル』, Dan Heller 著, ソフトバンク会社, 1992.

『Motif Reference Manual』, Paula Ferguson 著, O'Reilly & Associates 発行, 1992.

『The X Window System Programming and Applications with Xt, OSF/Motif Edition』, Douglas Young 著, Prentice Hall 発行, 1990.

OPEN LOOK プログラミング

『An OPEN LOOK at Unix: A Developer's Guide to X』, John David Miller, M&T Books, 1990.

『The X Window System Programming and Applications with Xt, OPEN LOOK Edition』, Douglas Young and John A. Pew, Prentice Hall, 1992.

『XView プログラミング・マニュアル』, Dan Heller 著, ソフトバンク発行, Inc., 1991.

『XView Reference Manual』, Thomas Van Raalte (ed.), O'Reilly & Associates, Inc., 1991.

Xt/XLib プログラミング

『*Programmer's Supplement for Release 5 of the X Window System, Version 11*』, David Flanagan, O'Reilly & Associates, Inc., 1991.

『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』, Adrian Nye および Tim O'Reilly 著, ソフトバンク発行, 1990.

『*X Window System Toolkit*』, Paul J. Asente および Ralph R. Swick 著, Digital Press 発行, 1990.

『*X Window System, X Version 11 Release 5*』, Third Edition, Digital Press 発行, 1992.

『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』, ソフトバンク発行, 1991.

『*Xlib: C Language X Interface*』, James Gettys, Robert W. Scheifler, および Ron Newman 著, Silicon Press 発行, 1989.

『*Xlib* プログラミング・マニュアル』, Adrian Nye 著, ソフトバンク発行, 1990.

『*Xlib* リファレンス・マニュアル』, Adrian Nye 編, ソフトバンク発行, 1990.

索引

数字・記号

2次テキストセレクション, 35

A

app-defaults, 77

B

BIL ファイル, 56

C

CascadeButton, 20

catclose(), 76

catgets(), 76

.catファイル, 79

CDE

OpenWindows と共通の機能, 28, 32

固有の機能, 27

CDE アプリケーション、国際化, 75, 79

CDE ウィジェット

DtComboBox, 22

DtMenuButton, 23

DtSpinBox, 22

Motif 2.1 との互換性, 24

デモプログラム, 24

ライブラリとヘッダファイル, 24

CDE の例、ソースコードの位置, 57

CDE マニュアル, 81

ComboBox, 64

D

DATA_ATTRIBUTES テーブル, 30

DATA_CRITERIA テーブル, 30

DEC VT220, 24, 43

Devguide, 28, 56

DPS, 27

DropTarget, 44

DtComboBox, 22, 24, 43

DtInitialize(), 75

dtksh, 27

DtMenuButton, 22, 23, 24, 43

DtSpinBox, 22, 24, 43

DtTerm, 43

F

FileSelectionBox, 44

FlatWidget, 44

FooterPanel, 44

G

gencat, 78

gettext(), 76

GIL から BIL へのコンバータ, 28, 56

GIL ファイル, 56

gmf, 56

GUI

- アプリケーションビルダ, 28
- 開発ツール, 55
- マニュアル, 83

guil, 56

I

IXI 1.2.2 Motif, 20

L

- libDtTerm, 25
- libDtWidget, 22, 24, 44
- libMrm, 20
- libUil, 20
- libXm, 20

M

- Microsoft Windows, 19
- Motif, 64
 - GUIビルダ, 51, 55, 57
 - IXI 1.2.2, 20
 - Solaris, 19, 21
 - Solaris ツールキットのまとめ, 13
 - UIL ライブラリ, 20
 - アプリケーションと配色, 32
 - ウィジェットと XView オブジェクト, 38
 - ウィンドウマネージャ (mwm), 61
 - 既存の機能の拡張, 19
 - 共用ライブラリのポリシー, 18
 - プログラミングマニュアル, 84
 - ~への移行, 55
 - 変換ユーティリティ, 56
 - ボタン, 36
 - リソースの現地仕様化, 78
- Motif 1.2, 23
- Motif 2.1, 24
- Motif 2.1 マニュアル, 82
- Motif アプリケーション開発
 - OpenWindows 環境, 14
 - Solaris CDE 環境, 16
- Motif アプリケーションのリンク, 18
- Motif コード、GIL ファイルから生成, 28

.msgファイル, 78

N

- NL_CAT_LOCALE, 76
- NLSPATH, 75
- NLS 環境, 75

O

- OLIT, 15, 28, 34, 37, 59
- OLIT ライブラリ
 - Motif に固有のウィジェット, 43
 - Solaris Motif だけにあるルーチン, 41
 - ウィジェット, 42
 - ~だけにあるルーチン, 41
 - ~と Solaris Motif ライブラリの違い, 41, 44
- OPEN LOOK, 22, 34

P

PostScript, 27

Q

\$quote 宣言, 78

R

RubberTile ウィジェット, 45

S

- Session Manager, 28
- setlocale(), 76
- 「Snap (スナップ)」ボタン, 64
- Solaris
 - Motif 実行時と開発者サポート, 17
 - Motif ツールキットのまとめ, 13
- Solaris Motif
 - Motif 2.1 に追加された機能, 19
 - Solaris と CDE 環境, 14

Solaris Motif (続き)
実行時と開発者サポート, 17
ライブラリ<\${startrange}\>, 20
ライブラリ, 20
Solaris Motif への移行, 47
Stub ウィジェット, 45

T

ToolTalk メッセージサービス, 28
メッセージセット, 29
ttsnoop, 29
tttrace, 29

U

UIL, 20, 41, 56
UilDef.hヘッダファイル, 20
UIL コンパイラ, 17, 20

X

XGL, 27
XIL, 27
Xlib, 37
XmArrowButton ウィジェット, 43
XmComboBox ウィジェット, 24
XmCommand ウィジェット, 43
XmDrawnButton ウィジェット, 43
XmForm ウィジェット, 21
XmFrame ウィジェット, 44
XmLabelGadget ウィジェット, 44
XmList 簡易関数, 21
XmMainWindow ウィジェット, 44
XmPanedWindow ウィジェット, 44
XmSelectionBox ウィジェット, 44
XmSpinBox ウィジェット, 24
XmString, 77
Xm ライブラリ, 44
XtSetLanguageProc(), 76
Xt イントリンシクス, 59
Xt ライブラリ, 44
XView, 15, 28, 34, 37, 38, 59
パッケージ, 39

XView ライブラリ
Motif との違い, , 38
Motif に似たオブジェクト, 39
外部ファイル, 40
用語, 36
X サーバ, 27
X リソース, 40

あ

アーキテクチャ、OPEN LOOK と Motif の, 37
アクション、CDE, 30
アプリケーションビルダ (AppBuilder), 28, 51,
56, 57, 62

い

移植
2 段階のプロセス, 61
Solaris Motif への, 47, 59
アーキテクチャの影響, 51
基本的な統合, 48
小型アプリケーション, 50
の要素, 47
まとめ, 59
利点, 49
例, 61
移植作業
CDE, 15
Motif, 14

う

ウィジェット
CDE コントロール, 22, 24
DtTerm ウィジェット, 24
Motif 1.2, 23
Motif 2.1, 24
XmForm ウィジェット, 21
XmList ウィジェット, 21
ウィンドウコントロール, 35

お

オブジェクト、XView と Motif に共通のタイプ, 38
オプションの統合, 50

か

開発ツール、GUI, 55

き

既存のアプリケーション
CDE 環境での実行, 15
Solaris 環境での実行, 14
基本的な統合, 48
共通デスクトップ環境 Motif、Solaris Motif を参照, 8

こ

コールバック, 37, 38
国際化, 28, 31
CDE アプリケーション, 75, 79
コントロール、ウィンドウ, 35
コンパイル、Motif アプリケーション, 18

さ

作業
CDE への移植, 15
Motif への移植, 14

し

実行時端末、dtterm, 25

す

推奨する統合, 49

そ

ソースコードの位置、CDE の例, 57

た

端末
DEC VT220, 24, 43
DtTerm ウィジェット, 24
端末ウィジェット
デモプログラム, 25
ライブラリとヘッダファイル, 25

つ

ツールキットでの実装の違い, 34

て

ティアオフメニュー, 35
データ入力、CDE, 30
テキストエディタ
DtEditor ウィジェット, 25
ウィジェットを使う場合, 26
デモプログラム, 26
ライブラリとヘッダファイル, 26
テキストエディタ:, 43
テキストセレクション、二次, 35
テキストフィールドと矢印ボタンの
ウィジェット (DtSpinBox), 22
テキストフィールドとリストボックスの
ウィジェット (DtComboBox), 22
デスクトップ Korn シェル (dtksh), 27
デモプログラム, 24, 25

と

統合
CDE 環境に, 49
オプションの, 50
基本的な, 48
推奨する, 49
統合のレベル
オプション, 50
基本的な, 48

統合のレベル (続き)
 推奨する, 49
 ドラッグ&ドロップ, 28

は
 配色, 32

ふ
 フォント, 31, 79
 プログラミングモデル、XView と Motif, 38
 分類機構, 29

へ
 ヘルプシステム, 28, 30, 31

ま
 マウスボタンの動作, 35
 マニュアル
 CDE, 81
 GUI, 83
 Motif 2.1, 82
 Motif プログラミング, 84
 OPEN LOOK プログラミング, 84
 ToolTalk, 82
 Xt/XLib プログラミング, 85
 マネージャウィジェット, 52
 マネージャ、ワークスペース, 27

め
 メッセージカタログの配信, 78
 メッセージセット, 29
 メニュー、ティアオフ, 35
 メニューボタンのウィジェット
 (DtMenuButton), 23

も
 モデル、XView と Motif のプログラミング, 38

ら
 ラジオ・ボタン, 64

わ
 ワークスペースマネージャ, 27

