



Sun WorkShop Visual の新機能

Sun WorkShop Visual 6.0

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

Part No. 806-4849-01
2000 年 6 月 Revision A

Copyright 2000 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303 U.S.A. All rights reserved.

Copyright 2000 Imperial Software Technology Limited. 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 Limited 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、著作権法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).

Sun、Sun Microsystems、SunDocs、SunExpress、Java、JavaBeans、Java Workshop 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。

Visaj は、Pacific Imperial, Inc の商標です。X-Designer は、Imperial Software Technology Limited の商標です。

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。

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。

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： *WorkShop Visual User's Guide Enhancements*

© 2000 by Sun Microsystems, Inc.



製品名の変更について

Sun は新しい開発製品戦略の一貫として、Sun の開発ツール群の製品名を Sun WorkShop™ から Forte™ Developer に変更いたしました。製品自体の内容に変更はなく、従来通りの高品質をお届けいたします。

これまでの Sun の主力製品である基本プログラミングツールに、Forte Fusion™ や Forte™ for Java™ といった Forte 開発ツールの得意とする、マルチプラットフォームおよびビジネスアプリケーション実装の機能を盛り込むことで、より広範囲できめ細かな製品ラインが完成されました。

WorkShop 5.0 で使用されていた名称と、Forte Developer 6 で使用される新しい名称の対応については、以下の表をご覧ください。

旧名称	新名称
Sun Visual WorkShop™ C++	Forte™ C++ Enterprise Edition 6
Sun Visual WorkShop™ C++ Personal Edition	Forte™ C++ Personal Edition 6
Sun Performance WorkShop™ Fortran	Forte™ for High Performance Computing 6
Sun Performance WorkShop™ Fortran Personal Edition	Forte™ Fortran Desktop Edition 6
Sun WorkShop Professional™ C	Forte™ C 6
Sun WorkShop™ University Edition	Forte™ Developer University Edition 6

製品名の変更に加えて、次の 2 つの製品について大きな変更があります。

- Forte for High Performance Computing には Sun Performance WorkShop Fortran に含まれていたすべてのツール、および C++ コンパイラが含まれます。したがって、High Performance Computing のユーザーは開発用に 1 つの製品だけを購入すれば済むことになります。
- Forte Fortran Desktop Edition は以前の Sun Performance WorkShop Personal Edition と同じです。ただし、この製品に含まれる Fortran コンパイラでは、自動並列化されたコード、および明示的な指令に基づいた並列コードは生成できません。この機能は Forte for High Performance Computing に含まれる Fortran コンパイラでは使用できます。

Sun の開発製品を引き続きご利用いただきましてありがとうございます。今後もみなさまのご要望にお応えする製品をお届けできるよう努力してまいります。

目次

製品名の変更について	iii
Sun WorkShop Visual 6 の新機能	1
はじめに	1
Java Swing コードの生成	1
Motif ウィジェットから Swing コンポーネントへのマッピング	3
MFC コード生成の機能改善	6
イベントハンドラ	7
プロジェクトファイル	7
MFC バージョン 4 用の機能改善	8
アプリケーションシェル	9
Microsoft Windows でのヘルプ	9
スクロールウィンドウのマッピング	10
ユーザー定義ウィジェット用の Java コード	11
ユーザー定義ウィジェット用の MFC コード	12
その他の機能改善	12
コード生成ダイアログ	12
デフォルトのヘルプブラウザ	13

Sun WorkShop Visual 6 の新機能

はじめに

本書では、Sun WorkShop Visual 6 の新機能および主要な変更点について説明します。Sun WorkShop Visual 5 のユーザーズガイドとあわせてご覧ください。参照先として記述されているページ、節、章は、すべて Sun WorkShop Visual 5 のユーザーズガイドのものであります。

Sun WorkShop Visual 6 での主な変更箇所は以下のとおりです。

1. Java Swing コードの生成
2. MFC コード生成の機能改善
3. ユーザー定義ウィジェット用の Java コード
4. その他の機能改善

Java Swing コードの生成

Sun WorkShop Visual は Swing コンポーネントセットを使用するためのコードも生成できるようになりました。Swing コンポーネントは JFC (Java Foundation Classes) を使用しているため、ウィンドウシステムに依存しないコンポーネントを作ることができます。また、JFC の機能である「プラグイン可能な Look & Feel」も実現できるため、主要な Java プラットフォーム (Microsoft Windows、Solaris、Macintosh) のいずれの Look & Feel にも対応可能な、単一のユーザーインターフェースを作ることができ

ます。この Look & Feel は、実行時にアプリケーションを再起動しなくても切り替えることができます。Swing によって、一貫性のあるインタフェースを持つ、プラットフォームに依存しないアプリケーションを開発できます。

自分のデザインから Swing コードを生成した場合、CLASSPATH に Swing の jar ファイルのパスも含める必要があります。ただし Java 2 を使用している場合は、その中に Swing コンポーネントが含まれているため、この設定は必要ありません。

自分のデザインから Swing コードを生成するには、図 1 に示すように「Java オプション」ダイアログで Java バージョンとして「Swing」を選択します。このダイアログは、生成する言語として Java を選択して「コード生成」ダイアログを起動し、そこで「Java オプション」ボタンを押すと表示されます。Java のバージョンに「Swing」を選択すると、生成されたアプリケーションは Java 1.0、1.2、および Java 2 と互換性のあるすべてのバージョンで実行できます。

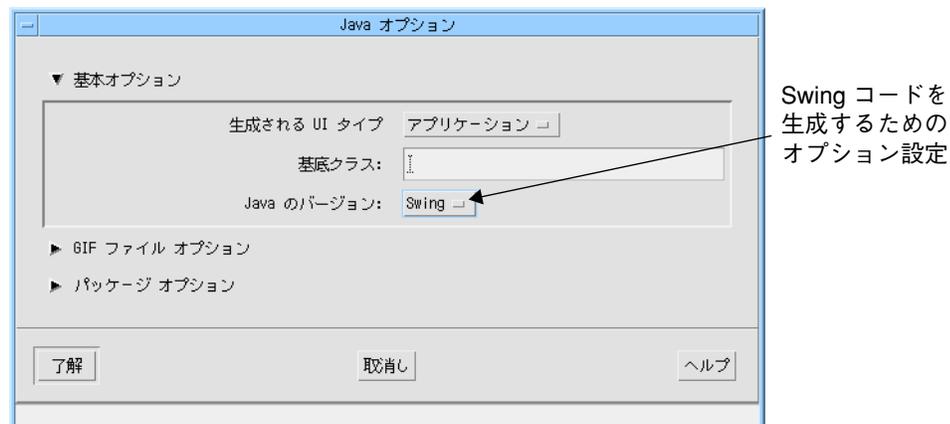


図 1 Swing 生成オプション

リソースパネルには新しい Java アイコンが表示されます。☕ という、コーヒーカップに「S」という文字が付いたアイコンは、そのリソースが Swing でサポートされていることを示しています。図 2 に、この Swing のアイコンを含むリソースパネルを示します。



図 2 Swing 用のアイコンが表示されたりソースパネル

Motif ウィジェットから Swing コンポーネントへのマッピング

Motif ベースのデザインを Swing コンポーネントを使って Java コードとして生成する場合、Sun WorkShop Visual はデザイン中の各 Motif ウィジェットを Swing コンポーネントにマップしようとします。完全に一致するマップ先がない場合は、Sun

WorkShop Visual は自分自身の MWT ライブラリを使用して Swing コンポーネントを「模倣」し、そのウィジェットに対するコードを生成します。次の表に各 Motif ウィジェットのマッピングを示します。

表 1 Swing コード生成のための各 Motif ウィジェットのマッピング

Motif ウィジェット	Swing へのマッピング
シェル	シェルがアプリケーションシェルか最上位シェルに設定されている場合は、JFrame コンポーネントにマップされる。ダイアログシェルに設定されている場合は、JDialog コンポーネントにマップされる。
メッセージボックス	<p>XmNdialogType リソースの値に応じて、2 種類の Java クラスのどちらかにマップできる。XmNdialogType リソースは、メッセージボックスのリソースパネルの「設定」ページで設定可能。</p> <p>ダイアログの種類が「エラー」、「情報」、「質問」、「警告」、「作業中」のいずれかに設定されている場合は、MWT ライブラリの一部である IconMessagePanel クラスにマップされる。</p> <p>ダイアログの種類が「メッセージ」に設定されている場合は、MWT ライブラリの一部である MessagePanel クラスにマップされる。</p>
メインウィンドウ	JPanel コンポーネントにマップされる。
描画領域	JPanel コンポーネントにマップされる。
ダイアログテンプレート	JPanel コンポーネントにマップされる。
メニューバー	JMenuBar コンポーネントにマップされる。
ブリテンボード	JPanel コンポーネントにマップされる。
コマンド	MWT ライブラリの CommandPanel クラスにマップされる。
メニュー	JMenu コンポーネントにマップされる。ポップアップメニューは JPopupMenu コンポーネントにマップされる。
フォーム	JPanel コンポーネントにマップされる。
選択プロンプト	MWT ライブラリの SelectionPanel クラスにマップされる。
ラジオボックス	JPanel コンポーネントにマップされる。

表 1 Swing コード生成のための各 Motif ウィジェットのマッピング

Motif ウィジェット	Swing へのマッピング
区画ウィンドウ	子が 2 つだけの場合は JSplitPane コンポーネントにマップされる。子が 2 つを超える場合は単に Panel コンポーネントにマップされる。
選択ボックス	XmNdialogType リソースが「コマンド」に設定されている場合は MWT ライブラリの CommandPanel クラスに、そうでない場合は MWT ライブラリの SelectionPanel クラスにマップされる。
ローカラム	XmNrowColumnType リソース設定に応じて、次のようにマップされる。 <ol style="list-style-type: none"> 1. 「メニューバー」の場合は JMenuBar コンポーネント 2. 「プルダウンメニュー」の場合は JMenu コンポーネント 3. 「ポップアップメニュー」の場合は JPopupMenu コンポーネント 4. 「オプションメニュー」の場合は JComboBox コンポーネント 5. 上記以外の場合は JPanel コンポーネント
スクロールウィンドウ	「スクロール方針」リソースが「自動」に設定されている場合は、JScrollPane コンポーネントにマップされる。そうでない場合は MWT ライブラリの ScrollablePanel クラスにマップされる。
ファイル選択	JFileChooser コンポーネントにマップされる。
ラベル	JLabel コンポーネントにマップされる。
カスケードボタン	親ウィジェットがメニューバーかプルダウンメニューの場合は、親が適切なコンポーネントにマップされるためカスケードボタン自身はマップされない。そうでない場合は、JLabel コンポーネントにマップされる。
テキストフィールド	JTextField コンポーネントにマップされる。
プッシュボタン	親ウィジェットがプルダウンメニューの場合は、JMenuItem コンポーネントにマップされる。そうでない場合は、JButton コンポーネントにマップされる。
オプションメニュー	オプションメニューの子であるラベルウィジェットにラベル文字列が設定されている場合、JPanel コンポーネントにマップされる。ラベルウィジェットにラベル文字が設定されていない場合は、JComboBox コンポーネントにマップされる。

表 1 Swing コード生成のための各 Motif ウィジェットのマッピング

Motif ウィジェット	Swing へのマッピング
テキスト	JTextArea コンポーネントにマップされる。
トグルボタン	プルダウンメニューの子である場合、JCheckBoxMenuItem コンポーネントにマップされる。ラジオボックスウィジェット、またはラジオボタンが有効に設定されているリソースを持つローカラムウィジェットの子である場合、JRadioButton コンポーネントにマップされる。上記のすべてに当てはまらない場合は JCheckBox コンポーネントにマップされる。
セパレータ	メニューウィジェットの内部で使用されている場合は、JMenuItem コンポーネントにマップされる。そうでない場合は、JSeparator コンポーネントにマップされる。
スクロールテキスト	JTextArea コンポーネントを持つ JScrollPane コンポーネントにマップされる。
描画ボタン	JButton コンポーネントにマップされる。
スケール	JSlider コンポーネントにマップされる。
リスト	JList コンポーネントにマップされる。
矢印ボタン	MWT ライブラリの ArrowButton クラスにマップされる。
スクロールバー	JScrollBar コンポーネントにマップされる。
スクロールリスト	JList コンポーネントを持つ JScrollPane コンポーネントにマップされる。

MFC コード生成の機能改善

Sun WorkShop Visual で作成したデザインから生成される MFC コードの機能は、大幅に改善されました。この内容は、ユーザーインタフェースには影響がありません。以降では、実際の改善内容について詳しく説明します。

イベントハンドラ

ユーザーズガイドの 237 ページの「イベントハンドラ」では、Sun WorkShop Visual でデザイン中のウィジェットにイベントハンドラを追加する方法について説明しています。Motif ウィジェットに使用できる大部分のイベントマスクは、Microsoft Windows コードにマップできます。マッピング内容は次の表のとおりです。

表 2 イベントマスクと、生成される Microsoft Windows コード

Motif イベントマスク	Microsoft Windows コード
MouseMotion	ボタン押下の有無を問わず、すべてのマウス動作に対応可能な汎用ハンドラを生成
ButtonPress	左ボタン押下、中ボタン押下、右ボタン押下の 3 種類のハンドラをすべて生成
ButtonRelease	左ボタン解放、中ボタン解放、右ボタン解放の 3 種類のハンドラをすべて生成
EnterWindow	OnMouseActivate
ExposureMask	OnEraseBkgnd
KeyPressMask	WM_KEYDOWN
KeyRelease	WM_KEYUP
KeymapstateMask	WM_SYSKEYUP または WM_SYSKEYDOWN
LeaveWindowMask	WM_KILL_FOCUS
ResizeRedirect	WM_SIZE
PropertyChangeMask	ON_WM_PAINT
VisibilityChangeMask	WM_SHOWWINDOW

プロジェクトファイル

図 3 に示すように、「コードオプション」ダイアログにトグルが追加されました。このトグルをオンにすると、Microsoft Visual C++ で使用できるプロジェクトファイルが生成されます。これによって、デザイン用に Microsoft Windows コードを生成するたびにプロジェクトファイルを個別に生成する必要がなくなります。このトグルはデフォルトでオンに設定されています。

生成されるファイルは、これまでどおり以下の 2 種類です。

- <generate_filename>.dsw - 主要なプロジェクトファイル。このファイルは、次に挙げるファイル中の情報を使用します (<generate_filename> には、生成するファイルとして指定した名前が入ります)。
- <generate_filename>.dsp - 生成されたファイルに関する情報が含まれています。

上記のファイルを使用するには、Microsoft Visual C++ で主要なプロジェクトファイル (接尾辞 .dsw のファイル) を開きます。生成されたプロジェクトファイルは Visual C++ のバージョン 5 および 6 で使用できます。

プロジェクトファイル生成の有無を決定するためのチェックボックス

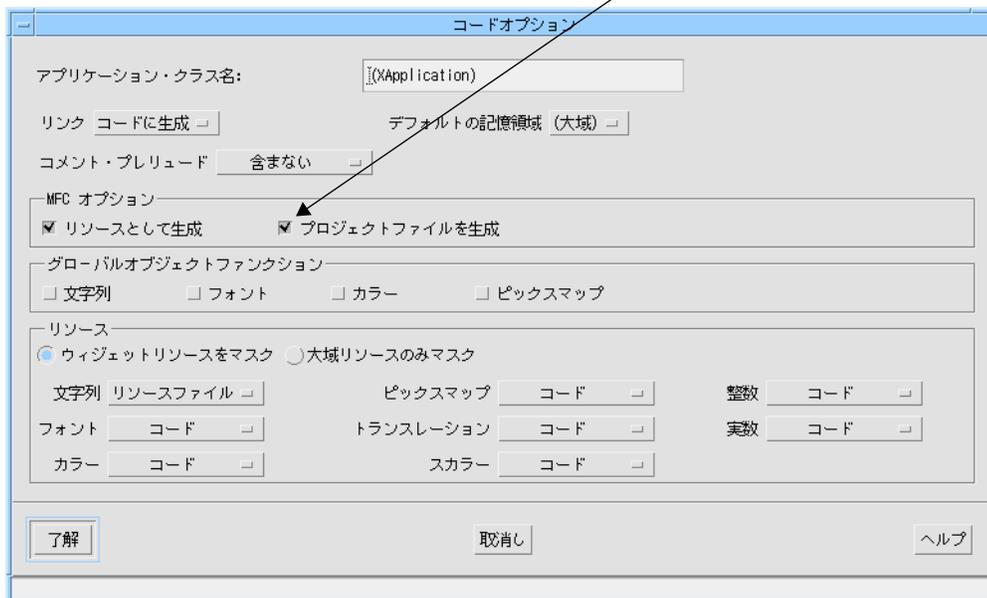


図 3 「コードオプション」ダイアログ

MFC バージョン 4 用の機能改善

以前のバージョンの Sun WorkShop Visual では MFC 4 機能拡張を利用するには、「コード生成」ダイアログから表示される「コードオプション」ダイアログでトグルをオンに設定しなければなりません。Sun WorkShop Visual 6 では、この機能はリソース `mfcFourEnhancements` として組み込まれました。このリソースを `true` に設定すると、MFC バージョン 4 の拡張された作成関数を使用したコードを生成でき、アプリケーションに立体的な Look & Feel を持たせることができます。また、このリ

ソースを `true` にすることで、ラベルおよびアイコンにビットマップがマップされるため、選択不可にする必要のない 3D ボタンおよびイメージをデザイン中で使用できます。この方法で追加されたイメージは、自分自身の自然なサイズを検知できるため、意図したとおりに Motif デザインを表現することができます。このリソースはデフォルトで `true` に設定されています。「コードオプション」ダイアログにあった MFC 4 機能拡張のトグルは、このバージョンから削除されました。

アプリケーションシェル

生成された MFC コードで、アプリケーションシェルは `CDialog` を継承するようになりました。つまり、ダイアログ自体を選択した状態で `Enter` キーを押すとアプリケーションが閉じます。この動作を使用したくない場合は、`OnOK` と `OnCancel` の 2 つのメソッドをそのダイアログのスタブファイルに生成します。これらのメソッドは、スタブファイルに追加したその他のメソッドと同様に、コードを再生成した場合にも保持されます。

Microsoft Windows でのヘルプ

Microsoft Windows コードにマップされる「`Activate Help`」コールバックを指定できるようになりました。

生成されたアプリケーションを Microsoft Windows 上で実行すると、「`Activate Help`」コールバックが、指定された URL と事前に定義されたヘルプのパスを使用して、システムのデフォルトの Web ブラウザを起動します。ヘルプのパスは「モジュール」メニューの「ヘルプのデフォルト」で表示される「ヘルプのデフォルト」ダイアログで指定します。デフォルトのパスはアプリケーションに含まれる HTML のルートになります。

HTML ファイルを複数のプラットフォームで開けるようにするには、プラットフォームに応じて HTML のルートへのパスを変更可能にする必要があります。たとえば、`c:\program files\myapp\myhtml` というパス名は Motif 版ヘルプシステムでは使用できません。このような設定がより簡単にできるように、Sun WorkShop Visual は次の構造をスタブファイルおよび main コードファイルに生成します。

main コードファイルには次のコードが生成されます。

```
#ifndef DUAL_PLATFORM
char * _xd_help_path = "c:\\program files\\myapp\\myhtml\\"
#else
```

```
extern char * _xd_help_path;
#endif
```

スタブファイルには、次のコードが生成されます。

```
#ifndef DUAL_PLATFORM
extern char * _xd_help_path;
#else
char * _xd_help_path = "c:\\program files\\myapp\\myhtml\\"
#endif
```

この構造は変更および拡張できます。変更した内容はコードを再生成した場合にも保持されます。この構造を利用すると、次の例のようにプラットフォームごとにヘルプのパスを指定することができます。

```
#ifndef DUAL_PLATFORM
extern char * _xd_help_path;
#else
#ifdef WIN32
char * _xd_help_path = "c:\\program files\\myapp\\myhtml\\"
#else
char * _xd_help_path = "/opt/myapp/HTML"
#endif
#endif
```

このコードを追加して、アプリケーション用のプロジェクト設定に上記の DUAL_PLATFORM を指定します。絶対パス名を指定したくない場合はこの仕組みを利用して、環境変数や登録されている設定を取り出すこともできます。

スクロールウィンドウのマッピング

「スクロールウィンドウ」ウィジェットの「スクロール方針」リソースが「自動」に設定されている場合は、スクロールウィンドウは生成された MFC コードでは CScrollView にマップされるようになりました。

ユーザー定義ウィジェット用の Java コード

Sun WorkShop Visual は、デザイン中のユーザー定義ウィジェットに対して Java コードを生成します。これには、ユーザー定義ウィジェットから Java コンポーネントへのマッピングを実現するための、特別に構成されたリソースファイルが使用されます。Sun WorkShop Visual に付属しているすべての統合キット (ユーザーズガイドの 720 ページの「事前構成済み統合キットの使用」を参照) には、このリソースファイルが含まれています。

```
$VISUROOT/user_widgets/<USER_WIDGET_SET>/app-defaults/visu
```

上記の VISUROOT は Sun WorkShop Visual のインストールディレクトリ、<USER_WIDGET_SET> は使用するウィジェットセットの名前です (XRT や Athena など)。

独自の統合キットを作成した (または今後作成する) 場合、独自のリソースファイルを作成してもかまいません。Java コード生成に関連するリソースは、次の 4 つです。

```
visu*xw_<WidgetClassName>.javaClassName  
visu*xw_<WidgetClassName>.java10ClassName  
visu*xw_<WidgetClassName>.java11ClassName  
visu*xw_<WidgetClassName>.javaSwingClassName
```

生成されたバージョンの Java に特定の Java リソースが見つからない場合は、Sun WorkShop Visual は「javaClassName」を使用します。上の例にある

「java10ClassName」や「java11ClassName」に対する値は、デフォルトと同じなので指定する必要はありません。

ウィジェットがリソースファイル中にない場合は、Sun WorkShop Visual は Java のパネルクラス用のコンテナウィジェットの代わりに、そのほかのすべての種類のウィジェットに使用できる Java キャンバスクラスを使用します。Java コード生成に関する詳細については、ユーザーズガイドの第 10 章「Java 用のデザイン」を参照してください。

ユーザー定義ウィジェット用の MFC コード

ユーザー定義ウィジェットに対する MFC コードは、前述の Java コード生成とまったく同じ仕組みで生成されます。Sun WorkShop Visual 6 は、他社製のコンポーネントを C++ クラスにするために以下のリソースを使用します。

```
visu*xw_<WidgetClass>.cppClassName  
visu*xw_<WidgetClass>.motifClassName  
visu*xw_<WidgetClass>.motifMfcClassName  
visu*xw_<WidgetClass>.mfcClassName
```

たとえば、次のように設定します。

```
visu*xw_XtXrtLabel.mfcClassName: CStatic
```

ファイル中に特定の様式が見つからない場合は、デフォルトで「cppClassName」が使用されます。

次の設定では、MFC、MotifXP、Motif の情報がない場合に、すべての様式の C++ に対して MyClass というクラスが生成されます。

```
visu*xw_SomeWidgetClass.cppClassName: MyClass
```

次の設定では、純粋な MFC 以外のすべての様式の C++ に対して AnotherClass というクラスが生成されます。

```
visu*xw_SomeWidgetClass.cppClassName: AnotherClass  
visu*xw_SomeWidgetClass.mfcClassName: AnotherMfcClass
```

その他の機能改善

ここでは、Sun WorkShop Visual バージョン 6 で改善されたその他の機能について説明します。

コード生成ダイアログ

コード生成ダイアログには、次の 2 点の変更があります。

- リセットボタンの追加
- 複数言語対応メイクファイルの生成方法の改善

「コード生成」ダイアログには「リセット」ボタンが追加されました。このボタンを押すと、選択された言語のページにあるファイル名がすべて元に戻ります。デフォルトのファイル名であることを示す括弧 <> も表示されます。

「メイクファイル・オプション」ダイアログには、図 4 のように「現在の言語のみ」というラベルの付いたトグルボタンが追加されました。ある言語でコードを生成し、続けて別の言語で別のコードを生成すると、生成されたすべてのメイクファイルには各言語用に生成されたファイルに対する規則がすべて盛り込まれます。しかし「現在の言語のみ」トグルをオンにすると、現在の言語専用のメイクファイルが生成されます。

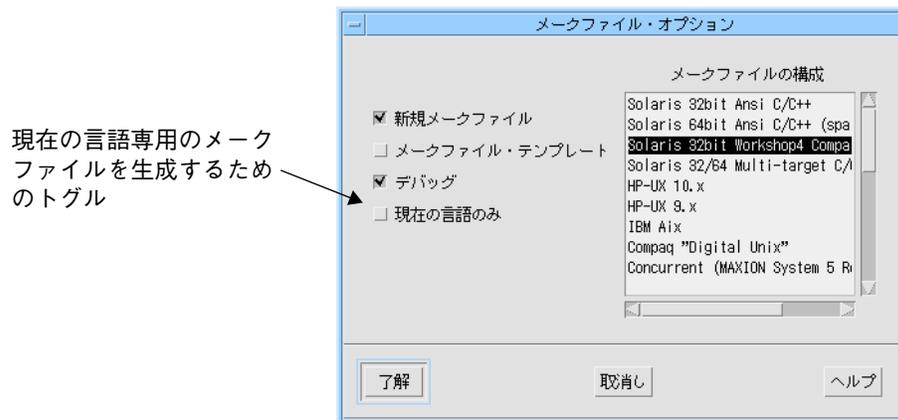


図 4 「メイクファイル・オプション」ダイアログ

デフォルトのヘルプブラウザ

デフォルトのヘルプブラウザは Netscape になりました。以前のデフォルトは Sun WorkShop Visual ヘルプでした。Sun WorkShop Visual ヘルプを使用するには、Sun WorkShop Visual の「ヘルプ」メニューの「ビューア」項目のプルダウンメニューから「Sun WorkShop Visual ヘルプを使用」を選択してください。

索引

A

「Activate Help」コールバック、Microsoft Windows での, 9

B

ButtonPress X イベントからイベントハンドラへのマッピング, 7

ButtonRelease X イベントから Windows イベントへのマッピング, 7

C

CDialog 中の OnCancel メソッド (アプリケーションシェルから), 9

CDialog 中の OnOK メソッド (アプリケーションシェルから), 9

D

DUAL_PLATFORM、Microsoft Windows 用, 9

E

EnterWindow X イベントから Windows イベントへのマッピング, 7

ExposureMask X イベントから Windows イベントへのマッピング, 7

J

Java

ユーザー定義ウィジェットからのマッピング, 11

java10ClassName、リソースファイル中の, 11

java11ClassName、リソースファイル中の, 11

javaSwingClassName、リソースファイル中の, 11

Java バージョンと Swing, 2

K

KeymapstateMask X イベントから Windows ハンドラへのマッピング, 7

KeyPressMask X イベントから Windows ハンドラへのマッピング, 7

KeyRelease X イベントから Windows ハンドラへのマッピング, 7

L

LeaveWindowMask X イベントから Windows ハンドラへのマッピング, 7

M

MFC

- 3D の Look & Feel, 8
- X イベントハンドラからのマッピング, 7
- コード生成の機能改善, 6
- バージョン 4 機能拡張, 8

mfcFourEnhancements, 8

MFC バージョン 4, 8

Microsoft Windows

- コード生成の機能改善, 6

Microsoft Windows イベントからイベントハンドラへのマッピングをまとめた表, 7

Microsoft Windows

- MFC 4 機能拡張, 8
- 複数プラットフォームでヘルプを実行, 9
- ヘルプを指定, 9

Microsoft Windows イベントハンドラ

- X イベントマスクからのマッピング, 7

Microsoft Windows コード用のイベントハンドラ, 7

Microsoft Windows 用のプロジェクトファイル, 7

Motif ウィジェットから Swing へのマッピング, 4

Motif ウィジェットから Swing へのマッピングをまとめた表, 4

MouseMotion X イベントからイベントハンドラへのマッピング, 7

N

Netscape, 13

O

OnEraseBkgnd, 7

OnMouseActivate, 7

ON_WM_PAINT, 7

P

PropertyChangeMask X イベントから Windows ハンドラへのマッピング, 7

R

ResizeRedirect X イベントから Windows ハンドラへのマッピング, 7

S

Swing

CLASSPATH に追加, 2

Motif ウィジェットからのマッピング, 4 説明, 1

Swing アイコンが表示されたリソースパネル, 3

V

VisibilityChangeMask X イベントから Windows ハンドラへのマッピング, 7

Visual C++ バージョン 5 およびバージョン 6, 8

W

WM_KEYDOWN, 7

WM_KEYUP, 7

WM_KILL_FOCUS, 7

WM_SHOWWINDOW, 7

WM_SIZE, 7

WM_SYSKEYDOWN, 7

WM_SYSKEYUP, 7

あ

アプリケーションシエル、Microsoft Windows コードでの, 9

い

イベントハンドラ
MFC へのマッピング, 7

お

オプションメニュー
Swing へのマッピング, 5

か

カスケードボタン
Swing へのマッピング, 5

く

区画ウィンドウ
Swing へのマッピング, 5

こ

「コード生成」ダイアログ
デフォルトの名前に戻す, 13
「コード生成」ダイアログ中の「リセット」ボタ
ン, 13
コーヒーカップのアイコン, 2
コマンド
Swing へのマッピング, 4

し

シェル
Swing へのマッピング, 4
自動スクロール方針、Microsoft Windows 用, 10

す

スクロールテキスト
Swing へのマッピング, 6
スクロールウィンドウ

Swing へのマッピング, 5

スクロールウィンドウ、Microsoft Windows への
マッピング, 10

スクロールバー
Swing へのマッピング, 6

スクロールリスト
Swing へのマッピング, 6

スケール
Swing へのマッピング, 6

せ

セパレータ
Swing へのマッピング, 6

選択プロンプト
Swing へのマッピング, 4

選択ボックス
Swing へのマッピング, 5

た

ダイアログテンプレート
Swing へのマッピング, 4
他社製ウィジェット
~用の Java コード, 11

て

テキスト
Swing へのマッピング, 6
テキストフィールド
Swing へのマッピング, 5
デフォルトのヘルプブラウザ, 13
へのマッピング, 5

と

トグルボタン
Swing へのマッピング, 6

ひ

描画ボタン

Swing へのマッピング, 6

描画領域

Swing へのマッピング, 4

ふ

ファイル選択

Swing へのマッピング, 5

フォーム

Swing へのマッピング, 4

プッシュボタン

Swing へのマッピング, 5

ブリテンボード

Swing へのマッピング, 4

へ

ヘルプ

デフォルトのブラウザ, 13

ヘルプ、Microsoft Windows コード用, 9

め

メインウィンドウ

Swing へのマッピング, 4

メイクファイル

現在の言語のみ生成, 13

「メイクファイル・オプション」ダイアログ中の

「現在の言語のみ」トグル, 13

メッセージボックス

Swing へのマッピング, 4

メニュー

Swing へのマッピング, 4

メニューバー

Swing へのマッピング, 4

や

矢印ボタン

Swing へのマッピング, 6

ゆ

ユーザー定義ウィジェット

~用の Java コード, 11

ユーザー定義ウィジェット用の Java コード, 11

ら

ラジオボックス

Swing へのマッピング, 4

ラベル

Swing へのマッピング, 5

り

リスト

Swing へのマッピング, 6

リソースファイル中の javaClassName, 11

ろ

ローカラム

Swing へのマッピング, 5