



Sun WorkShop の新機能

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

Part No. 806-4832-01
2000 年 6 月 Revision A

本製品およびそれに関連する文書は、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。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。

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。Netscape™、Netscape Navigator™、および Netscape Communications Corporation のロゴは、次の著作権で保護されています。

© 1995 Netscape Communications Corporation.

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

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

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

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。

Sun f90 / f95 は、米国 Silicon Graphics, Inc. の Cray CF90™ に基づいています。

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

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、明示的であるか黙示的であるかを問わず、あらゆる説明および保証は、法的に無効である限り、拒否されるものとします。

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 : *What's New in Sun WorkShop 6*
Part No: 806-3565-10
Revision A

© 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環として、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

はじめに xi

新機能のご紹介 1

Sun WorkShop 6 リリースへのアップグレードをお勧めする理由 1

Sun WorkShop 6 C コンパイラ 1

Sun WorkShop 6 C++ コンパイラ 4

部分的特殊化 6

明示的関数テンプレート引数 7

関数テンプレートの型名でないパラメータ 7

メンバーテンプレート 8

定義分離テンプレート編成に対する制約の解除 8

静的変数破棄の順序付け 9

副集合の初期設定 9

自分専用の C++ 標準ライブラリの使い方 10

キャッシュバージョン指定 11

Sun WorkShop 6 Fortran コンパイラ 11

Sun WorkShop 6 Fortran 77 コンパイラ 12

Sun WorkShop 6 Fortran 95 コンパイラ 13

Fortran コンパイラの新機能	16
Sun Workshop 6 Fortran 95 区間演算	17
区間演算とは	18
区間演算が重要である理由	18
詳細情報の入手先	18
Sun Workshop 6 dbx	19
Sun WorkShop 6	22
テキスト編集	24
プログラムのデバッグ	25
プロジェクトの取り扱い	29
Sun WorkShop 6 TeamWare	31
メニュー再編成の設定	32
プットバックの妥当性検査	33
SCCS 管理フラグ	34
ワークスペースの履歴表示	34
Sun WorkShop 6 Visual	35
Swing サポート	35
Windows 拡張サポート	36
Sun WorkShop 6 Performance Library	37
Fortran 95 言語機能サポート	38
Sun Performance Library のライセンス供与に関する変更。	38
Sun WorkShop 6 Analyzer	39
関数リスト	40
「呼び出し元 - 呼び出し先」ウィンドウ	40
注釈付きソースコードの生成	41
注釈付き逆アセンブリ	41
測定値	41
その他の変更事項	42

Sun WorkShop 6 インストール 43

Sun WorkShop 6 マニュアル、マニュアルページ、README、および HTML オンラインヘルプ 43

表目次

表 1	Sun WorkShop 6 C コンパイラ	2
表 2	Sun WorkShop 6 C++ コンパイラ	4
表 3	Sun WorkShop 6 FORTRAN 77 コンパイラ	12
表 4	Sun WorkShop 6 Fortran 95 コンパイラ	13
表 5	Sun WorkShop 6 dbx	19
表 6	Sun WorkShop 6	22
表 7	Sun WorkShop 6 TeamWare	31
表 8	Sun WorkShop 6 TeamWare メニューの変更	32
表 9	Sun WorkShop 6 Visual	35
表 10	X イベントを Windows MFC にマップする	36
表 11	Sun WorkShop 6 Performance Library	37
表 12	Sun WorkShop 6 Analyzer	39
表 13	Sun WorkShop 6 インストール	43

はじめに

このマニュアルでは、Sun WorkShop 6 コンパイラとツールの新機能について説明します。

マルチプラットフォーム対応

この Sun WorkShop リリースは、Solaris 2.6、7、および 8 のオペレーティング環境 (SPARC™ プラットフォームおよび Intel プラットフォーム) をサポートしています。

注 - IA および x86 とは、Pentium、Pentium Pro、Pentium II プロセッサおよび、これらと互換性のある AMD および Cyrix 製のマイクロプロセッサチップを含む、Intel 8086 マイクロプロセッサチップ群を意味しています。このマニュアルでは、これらすべてのプラットフォームアーキテクチャを総称して x86 と呼んでいます。製品名では、Intel プラットフォームと記述しています。

Sun WorkShop 開発ツールへのアクセス方法

Sun WorkShop 製品コンポーネントとマニュアルページは標準ディレクトリ `/usr/bin` および `/usr/share/man` にはインストールされません。そのため `PATH` および `MANPATH` 環境変数を変更して Sun WorkShop コンパイラとツールにアクセスできるようにする必要があります。

`PATH` 環境変数を設定する必要があるかどうか判断するには以下を実行します。

1. 次のように入力して、`PATH` 変数の現在値を表示します。

```
% echo $PATH
```

2. 出力内容から `/opt/SUNWspro/bin` を含むパスの文字列を検索します。

パスがある場合は、`PATH` 変数は Sun WorkShop 開発ツールにアクセスできるように設定されています。パスがない場合は、この節の指示に従って、`PATH` 環境変数を設定してください。

`MANPATH` 環境変数を設定する必要があるかどうか判断するには以下を実行します。

1. 次のように入力して、`workshop` マニュアルページを表示します。

```
% man workshop
```

2. 出力された場合、内容を確認します。

`workshop`(1) マニュアルページが見つからないか、表示されたマニュアルページがインストールされたソフトウェアの現バージョンのものと異なる場合は、この節の指示に従って `MANPATH` 環境変数を設定してください。

注 – この節に記載されている情報は Sun WorkShop 6 製品が `/opt` ディレクトリにインストールされていることを想定しています。Sun WorkShop ソフトウェアが `/opt` ディレクトリにインストールされていない場合は、システム管理者に連絡してください。

`PATH` 変数および `MANPATH` 変数は、C シェルを使用している場合はホームディレクトリの下 `.cshrc` ファイルに設定する必要があります。Bourne シェルか Korn シェルを使用している場合は、ホームディレクトリの下 `.profile` ファイルに設定する必要があります。

- Sun WorkShop コマンドを使用するには、`PATH` 変数に以下を追加してください。

```
/opt/SUNWspro/bin
```

- `man` コマンドで、Sun WorkShop マニュアルページにアクセスするには、`MANPATH` 変数に以下を追加してください。

```
/opt/SUNWspro/man
```

`PATH` 変数についての詳細は、`cs(1)`、`sh(1)` および `ksh(1)` のマニュアルページを参照してください。`MANPATH` 変数についての詳細は、`man(1)` のマニュアルページを参照してください。このリリースにアクセスするために `PATH` および `MANPATH` 変数を設定する方法の詳細は、『Sun WorkShop インストールガイド』を参照するか、システム管理者にお問い合わせください。

書体と記号について

このマニュアルで使用している書体と記号について説明します。

表 P-1 このマニュアルで使用している書体と記号

書体または記号	意味	例
<code>AaBbCc123</code>	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、コーディング例。	<code>.login</code> ファイルを編集します。 <code>ls -a</code> を使用してすべてのファイルを表示します。 <code>machine_name% You have mail.</code>
<code>AaBbCc123</code>	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表わします。	<pre>machine_name% su Password:</pre>
<code>AaBbCc123</code> または <code>ゴシック</code>	コマンド行の可変部分。実際の名前または実際の値と置き換えてください。	<code>rm filename</code> と入力します。 <code>rm</code> ファイル名 と入力します。
『 』	参照する書名を示します。	『SPARCstorage Array ユーザーマニュアル』

表 P-1 このマニュアルで使用している書体と記号 (続き)

書体または記号	意味	例
「 」	参照する章、節、または、強調する語を示します。	第 6 章「データの管理」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、バックスラッシュは、継続を示します。	<pre>machinename% grep `^#define \ XV_VERSION_STRING`</pre>
▶	階層メニューのサブメニューを選択することを示します。	作成: 「返信」▶「送信者へ」

シェルプロンプトについて

シェルプロンプトの例を以下に示します。

表 P-2 シェルプロンプト

シェル	プロンプト
UNIX の C シェル	machine_name%
UNIX の Bourne シェルと Korn シェル	machine_name\$
スーパーユーザー (シェルの種類を問わない)	#

関連マニュアル

次の方法で、関連マニュアルにアクセスすることができます。

- インターネットの docs.sun.com の Web サイトからアクセスできます。特定の本のタイトルで検索するか、主題、マニュアルコレクションまたは製品別にブラウズすることができます。

<http://docs.sun.com>

- ローカルシステムまたはローカルネットワークにインストールされた Sun WorkShop 製品からアクセスできます。Sun WorkShop 6 HTML 文書 (マニュアル、オンラインヘルプ、マニュアルページ、各コンポーネントの README ファイル、リリースノート) が、インストールした Sun WorkShop 6 製品から参照可能です。HTML 文書にアクセスするには、次のいずれかを実行します。
 - Sun WorkShop または Sun WorkShop™ TeamWare ウィンドウで、「ヘルプ」
 - ▶ 「オンラインマニュアルについて」を選択します。
 - Netscape™ Communicator 4.0 またはその互換バージョンのブラウザで、以下のファイルを開きます。

</opt/SUNWspro/docs/ja/index.html>

参照できる Sun WorkShop 6 HTML 文書の一覧がブラウザに表示されます。一覧にあるマニュアルを開くには、マニュアルのタイトルをクリックしてください。

表 P-3 は、Sun WorkShop 6 関連マニュアルをマニュアルコレクション別に一覧にしたものです。

表 P-3 マニュアルコレクション別 Sun WorkShop 6 関連マニュアル

マニュアルコレクション	マニュアルタイトル	内容の説明
Forte Developer 6 / Sun WorkShop 6 リリース マニュアル	Sun WorkShop 6 マニュアルの概要	Sun WorkShop 6 で使用可能なマニュアルとそのアクセス方法について説明しています。
	Sun WorkShop の新機能	Sun WorkShop 6 の現在のリリースと以前のリリースでの新機能についての情報を記載しています。
	Sun WorkShop 6 リリースノート	インストールの詳細と Sun WorkShop 6 最終リリースの直前に判明した情報を記載しています。このマニュアルはコンポーネントごとの README ファイルにある情報を補足するものです。

表 P-3 マニュアルコレクション別 Sun WorkShop 6 関連マニュアル

マニュアルコレクション	マニュアルタイトル	内容の説明
Forte Developer 6 / Sun WorkShop 6	プログラムのパフォーマンス解析	新しい標本コレクタと標本アナライザの使い方について説明しています (上級者向けのプロファイリング事例と説明付き)。コマンド行解析ツール <code>er_print</code> 、ループツール、ループレポートユーティリティおよび UNIX プロファイルツール <code>prof</code> 、 <code>gprof</code> 、 <code>tcov</code> についての情報も含んでいます。
	dbx コマンドによるデバッグ	dbx コマンドを使ってプログラムをデバッグする方法について説明しています。参考情報として、同じデバッグ処理を Sun WorkShop デバッグウィンドウを使って実行する方法も記載しています。
	Sun WorkShop の概要	Sun WorkShop 統合プログラミング環境の基本的なプログラム開発機能について説明しています。
Forte C 6 / Sun WorkShop 6 Compilers C	C ユーザーズガイド	C コンパイラオプション、サン固有の機能 (プラグマ、 <code>lint</code> ツール、並列化、64 ビットオペレーティングシステムへの移行および ANSI/ISO 準拠 C) について説明しています。
Forte C++ 6 / Sun WorkShop 6 Compilers C++	C++ ライブラリ・リファレンス	C++ ライブラリについて説明しています。C++ 標準ライブラリ、Tools.h++ クラスライブラリ、Sun WorkShop Memory Monitor、 <code>Iostream</code> および複素数の情報も含まれます。
	C++ 移行ガイド	コードを本バージョンの Sun WorkShop C++ コンパイラに移行する方法について説明しています。

表 P-3 マニュアルコレクション別 Sun WorkShop 6 関連マニュアル

マニュアルコレクション	マニュアルタイトル	内容の説明
	C++ プログラミングガイド	新しい機能を使ってより効率的なプログラムを記述する方法について説明しています。テンプレート、例外処理、実行時の型識別、キャスト演算、パフォーマンス、およびマルチスレッド対応のプログラムに関する情報も記載されています。
	C++ ユーザーズガイド	コマンド行オプションとコンパイラの使い方についての情報を記載しています。
	Sun WorkShop Memory Monitor ユーザーズガイド	C および C++ のメモリー管理で生じた問題を Sun WorkShop Memory Monitor で解決する方法について説明しています。このマニュアルはインストールした製品 (/opt/SUNWspro/docs/ja/index.html) からのみ参照可能で、 docs.sun.com Web サイトで参照することはできません。
Forte for High Performance Computing 6 / Sun WorkShop 6 Compilers Fortran 77/95	Fortran ライブラリ・リファレンス	Fortran コンパイラによって提供されるライブラリルーチンの詳細について説明しています。
	Fortran プログラミングガイド	入出力、ライブラリ、プログラム分析、デバッグおよびパフォーマンスに関連する内容を記述しています。
	Fortran ユーザーズガイド	コマンド行オプションとコンパイラの使い方についての情報を記載しています。
	FORTRAN 77 言語リファレンス	Fortran 77 言語の包括的な参照情報を記載しています。
	Fortran 95 区間演算プログラミングリファレンス	Fortran 95 コンパイラによってサポートされる組み込み INTERVAL データについて説明しています。
Forte TeamWare 6 / Sun WorkShop TeamWare 6	Sun WorkShop TeamWare ユーザーズガイド	Sun WorkShop TeamWare コード管理ツールの使用方法について説明しています。

表 P-3 マニュアルコレクション別 Sun WorkShop 6 関連マニュアル

マニュアルコレクション	マニュアルタイトル	内容の説明
Forte Developer 6/ Sun WorkShop Visual 6	Sun WorkShop Visual ユーザーズガイド	C++ と Java™ の GUI (グラフィカルユーザーインターフェース) を Sun WorkShop Visual を使用して作成する方法について説明しています。このマニュアルには、旧リリース (Sun WorkShop Visual 5.0) から変更のない機能が記載されています。
	Sun WorkShop Visual の新機能	Sun WorkShop Visual 6.0 で追加または変更された機能について説明しています。
Forte / Sun Performance Library 6	Sun Performance Library Reference (英語のみ)	コンピュータによる線形代数および高速フーリエ変換を実行するサブルーチンと関数の最適化ライブラリについて説明しています。
	Sun Performance Library User's Guide (英語のみ)	線形代数で発生した問題の解決に使用されるサブルーチンと関数のコレクションである Sun Performance Library のサン固有の機能の使用方法について説明しています。
数値計算ガイド	数値計算ガイド	浮動小数点演算における数値の精度に関する問題について説明しています。
標準ライブラリ 2	Standard C++ Library Class Reference (英語のみ)	標準 C++ の詳細について説明しています。
	標準 C++ ライブラリ・ユーザーズガイド	標準 C++ ライブラリの使用方法について説明しています。
Tools.h++ 7	Tools.h++ 7.0 ユーザーズガイド	Tools.h++ クラスライブラリの詳細について説明しています。
	Tools.h++ 7.0 クラスライブラリ・リファレンスマニュアル	C++ クラスを使用して、プログラム効率を向上させる方法について説明しています。

表 P-4 は、docs.sun.com の Web サイトからアクセスできる Solaris 関連マニュアルの一覧です。

表 P-4 Solaris 関連マニュアル

マニュアルコレクション	マニュアルタイトル	内容の説明
Solaris ソフトウェア開発	リンカーとライブラリ	Solaris リンクエディタと実行時リンカーの操作およびそれらが操作するオブジェクトについて説明しています。
	プログラミングユーティリティ	Solaris オペレーティング環境で使用可能な特殊組み込みプログラミングツールに関する開発者向けの情報を記載しています。

新機能のご紹介

このマニュアルでは、Sun WorkShop™ 6 のコンパイラとツールの新しい機能について説明します。これらの新機能については、表にまとめられています。一部の機能については、各表の次の節で詳しく説明しています。

Sun WorkShop 6 リリースへのアップグレードをお勧めする理由

以下の機能が、このリリースの特長です。

- ANSI/ISO C++ への準拠の強化
- OpenMP 並列化指令が追加された Fortran 95 コンパイラ
- UltraSPARC™ III のサポート
- より使いやすくなったプログラミング環境
- 新しいパフォーマンス解析ツール
- Fortran 95 区間演算
- インストールの改良
- HTML 形式による マニュアル、マニュアルページ、README、オンラインヘルプ

Sun WorkShop 6 C コンパイラ

表 1 は、Sun WorkShop 6 C コンパイラの新機能です。これらの機能により、C コンパイラと `lint` ソースコードチェッカーがより優れたものとなっています。

表 1 Sun WorkShop 6 C コンパイラ

機能	説明
<code>__func__</code>	新リリースの C コンパイラは、あらゆる関数定義に指定される静的な一定の文字配列 <code>__func__</code> を事前定義します。この配列は、関名によって初期化され、静的関数のスコープ配列を使用できる場所であればどこにでも使用できます。たとえば、包含する関数の名前を出力する場合に使えます。
可変引数マクロ	C プリプロセッサは、 <code>#define</code> マクロについて可変数の引数を受け付けます。マクロ定義の識別子リストの一部として省略符号 (...) が含まれている場合、マクロ起動時に使用される引数は、マクロ定義内のパラメータより多くなります。
<code>SUNW_MP_THR_IDLE</code>	新リリースではスレッドがタスクを終了した後や「休眠状態」時にもシステム資源を使い続けるかどうかを制御する、 <code>SUNW_MP_THR_IDLE</code> という環境変数を使用できます。
大規模配列	新リリースの C コンパイラは、大規模な配列オブジェクトをサポートします。『C ユーザーズガイド』の付録 A を参照してください。
<code>-errchk=locfmtchk</code>	新リリースの <code>lint</code> は、新しいフラグ <code>-errchk=locfmtchk</code> を受け付けます。このフラグは、 <code>lint</code> の 1 回目のパスにおいて <code>printf</code> に似た書式の文字列があるかどうかを検査します。
<code>lint</code> が受け付ける新しい指令 (<code>PRINTF LIKE (n)</code>)	新リリースの <code>lint</code> は、ポインタによる <code>printf()</code> ファミリーへの呼び出しすべてを識別する新しい指令を受け付けます。こういった、ポインタによる呼び出しの引数整合性を <code>lint</code> によってチェックできます。
<code>-errwarn=t</code>	新リリースの C コンパイラと <code>lint</code> ソースコードチェッカーは、新しいオプション <code>-errwarn=t</code> をサポートします。このオプションが指定されている場合、指定した警告のどれかが発行されるとコンパイラはエラーステータスを出力して終了します。

表 1 Sun WorkShop 6 C コンパイラ (続き)

機能	説明
<code>-errchk</code>	新リリースの <code>lint</code> の <code>-errchk</code> には、新しい値 <code>signext</code> が用意されています。この値は、 <code>-errchk=longptr64,signext</code> のように <code>longptr64</code> オプションと組み合わせて使えます。このオプションは、64 ビットの開発環境への移行を行いやすくするため、符号拡張について警告します。
<code>-xchar_byte_order</code>	<code>-xchar_byte_order</code> オプションは、複数文字の文字定数から成る文字を、指定されたバイト順序で配置することによって整定数を生成します。
<code>-xinline</code>	新リリースの <code>-xinline</code> オプションには、 <code>%auto function_name</code> と <code>no% function_name</code> の 2 つの値が追加されました。 <code>%auto</code> は <code>-xO4</code> の最適化レベルで適用され、それによって <code>-xinline</code> にリストされている関数以外の関数も、自動的にインライン化します。 <code>%no function-name</code> は、関数 <code>function-name</code> をインライン化しないことをコンパイラに指示します。
<code>-xmemalign</code>	新リリースの C コンパイラには、 <code>-xmemalign</code> というオプションが新たに用意されています。このオプションはメモリアクセスが境界整列していない可能性に備えて生成されるコードを制御し、アクセスが境界整列していないときにプログラムの動作を制御します。
<code>-xprefetch</code>	新リリースでは、 <code>sun_prefetch.h</code> ヘッダーファイルと <code>-xprefetch</code> オプションを使うことによって明示的な先読み命令を指定できます。
<code>-xvector</code>	

Sun WorkShop 6 C++ コンパイラ

表 2 では、Sun WorkShop™ 6 C++ コンパイラのリリースで利用できる新機能を一覧表示します。一部の製品の機能については、表の次の節でさらに詳しく説明しています。

表 2 Sun WorkShop 6 C++ コンパイラ

機能	説明
部分的特殊化	テンプレートは部分的に特殊化することができます。つまり、一部のテンプレートパラメータのみを指定したり、または複数のパラメータを特定の型のカテゴリに制限することができます。
明示的関数テンプレート引数	関数の引数からテンプレートの引数を推察できない場合、新リリースでは <code>f<template args>(function args)</code> の構文を使って明示的に指定できます。
関数テンプレートの型名でないパラメータ	新リリースでは、以下のような、関数テンプレートの型名でないパラメータをサポートします。 <pre>template<int I> void foo(int a[I]) { ... } template<int I> void foo(mytype<I> m) { ... }</pre>
メンバーテンプレート	新リリースの標準モードでは、クラスとクラステンプレートにメンバーとしてテンプレートを持たせることができます。
定義分離テンプレート編成に対する制約の解除	新リリースのコンパイラでは、 <code>-instances !=extern</code> (つまり、 <code>-instances=explicit</code> 、 <code>-instances=global</code> 、 <code>-instances=semiexplicit</code> 、または <code>-instances=static</code>) における“定義分離テンプレート編成”に対する制約が解除されています。 <code>-instances</code> の設定値にかかわらず、定義検索時に個別のソースファイルを含めるのがコンパイラのデフォルトの動作となっています。

表 2 Sun WorkShop 6 C++ コンパイラ (続き)

機能	説明
先読み命令	<p>ヘッダーファイル <code>sun_prefetch.h</code> とともに新しいオプション <code>-xprefetch</code> を使用すると、UltraSPARC II など先読みをサポートするアーキテクチャ (<code>-xarch=v8plus, v9plus, v9plusa, v9, v9a</code>) では先読み命令を指定できます。</p>
外部インライン関数	<p>このバージョンのコンパイラでは、外部インライン関数を使用できます。インライン関数に局所的な静的データが入っている場合は、すべてのコンパイル単位でその静的データのコピーが 1 つだけ使用されます。ただし、異なる翻訳単位で取得したインライン関数のアドレスは等しいとはみなしません。</p>
静的変数破棄の順序付け	<p>規格により、静的記憶期間のあるオブジェクトを破棄する順序がさらに詳しく定義されました。静的オブジェクトは、作成した順序とは逆の順序で破棄する必要があります。以前の言語の定義では、いくつかの局面での規定が放置されていました。</p>
副集合の初期設定	<p>(中括弧初期設定を使用できる型に関し) クラスオブジェクトの中括弧初期設定を使用している場合、C++ 標準では、それ自体が集合クラスであるメンバーは、自分自身の型の値で初期設定できるようになりました。</p>
自分専用の C++ 標準ライブラリの使い方	<p><code>-library=no*Cstd</code> オプションを指定することで、コンパイラ付属のバージョンの代わりに、自分専用の C++ 標準ライブラリのバージョンを使用できます。</p>
キャッシュバージョン指定	<p>C++ コンパイラには、キャッシュバージョンの違いを検出し、適切なエラーメッセージを出力する機能があります。</p>

表 2 Sun WorkShop 6 C++ コンパイラ (続き)

機能	説明
ビットフィールドサイズに関する制限の削除	ビットフィールドのサイズを 32 以下にするという制限は削除されました。ビットフィールドのサイズは任意になりました。
関数のポインタと <code>void*</code> の相互間の変換に関する警告	以前のコンパイラは関数のポインタと <code>void*</code> で相互間の変換に関する警告を必ず出しましたが、現在では <code>+w2</code> オプションを使用する場合に限り警告が出されず。
新規オプションおよび変更されたオプション	<p>以下のリストでは、新規オプションと変更されたオプションを示します。</p> <ul style="list-style-type: none"> • 新規オプション <ul style="list-style-type: none"> <code>-xcrossfile</code> <code>-Bsymbolic</code> <code>-features=[no%]strictdestrorder</code> <code>-template=extdef</code> • 変更されたオプション <ul style="list-style-type: none"> <code>-fast</code> <code>-library=[no%]Catd,+p</code> <code>-ptr, -xprefetch</code> <p>詳細は、『C++ ユーザーズガイド』を参照してください。HTML で作成されたこのマニュアルを利用するには、ブラウザで file:/opt/SUNWspro/docs/ja/index.html にアクセスしてください。</p>

部分的特殊化

テンプレートは完全に特殊化できます。つまり、特定のテンプレート引数に対する実装が定義されました。以下のコード例を参照してください。

```
template<class T, class U> class A { ... }; // 初期テンプレート
template<> class A<int, double> { ... }; // 特殊化
```

テンプレートは部分的に特殊化することもできます。つまり、一部のテンプレートパラメータのみを指定したり、複数のパラメータを特定の型のカテゴリに制限することができます。その結果行われた部分的特殊化は、それ自体もテンプレートとして使用できます。以下は、上記の例の初期テンプレートを使用した例です。

- 初期テンプレートパラメータが `int` 型である場合の特殊テンプレートの定義。以下のコード例を参照してください。

```
template<class U> class A<int> { ... };
```

- 初期テンプレートパラメータが任意のポインタ型である場合の特殊なテンプレートの定義。以下のコード例を参照してください。

```
template<class T, class U> class A<T*> { ... };
```

- 初期テンプレートパラメータが任意の型のポインタのポインタで、2番目のテンプレートパラメータが `char` 型である場合の特殊なテンプレートの定義。以下のコード例を参照してください。

```
template<class T> class A<T**, char> { ... };
```

明示的関数テンプレート引数

関数の引数からテンプレートの引数を推察できない場合、`f<template args>(function args)` の構文を使って明示的に指定できます。

以下は、その一例です。

```
template<class Mytype> Mytype* construct(float, float);  
...  
int* x = construct<int>(a, b);
```

関数テンプレートの型名でないパラメータ

新リリースでは、次のような関数テンプレートの型名でないパラメータをサポートします。

```
template<int I> void foo( int a[I] ) { ... }  
template<int I> void foo( mytype<I> m ) { ... }
```

新リリースでは、次の例にあるように、型名でないテンプレートパラメータが入った式を関数パラメータリストに使用することはできません。

```
// 以下はサポートされません
template<int I> void foo( mytype<2*I> ) { ... }
template<int I, int J> void foo( int a[I+J] ) { ... }
```

メンバーテンプレート

標準モードでのクラスとクラステンプレートには、次のコード例のように、メンバーとしてテンプレートを持たせることができます。

```
template <class T1>
class OuterClass {
public:
    // クラスメンバーテンプレート
    template <class T2>
        class MemberClass
        {
            T2 MCmember;
            T1 OCmember;
        };
    template<class T3> operator T3() { ... }
    ...
};
```

注 - メンバーテンプレートは、互換モード (`-compat [=4]`) ではサポートされていません。

定義分離テンプレート編成に対する制約の解除

新リリースのコンパイラでは、`-instances != extern` (`-instances=explicit`、`-instances=global`、`-instances=semiexplicit`、または `-instances=static`) における「定義分離テンプレート編成」に対する制約が解除されています。`-instances` の設定値にかかわらず、定義検索時に個別のソースファイルを含めるのがコンパイラのデフォルトの動作となっています。

この制約を再び有効にするには、`-template=no%extdef` オプションを使用してください。ただし、`-template=no%extdef` オプションが指定されている場合、コンパイラは `-instances=extern` が指定されても個別のソースファイルを検索しません。

静的変数破棄の順序付け

規格により、静的記憶期間のあるオブジェクトを破棄する順序がさらに詳しく定義されました。静的オブジェクトは、生成した順序とは逆の順序で破棄する必要があります。以前の言語の定義では、いくつかの局面での規定が放置されていました。

厳しくなったこの順序付けは標準モードの場合に限り実装されます。互換モード (`-compat [=4]`) では、破棄の順序は以前と同様に実装されます。

プログラムが破棄について特定の順序に依存しており、しかも旧コンパイラで動作していた場合、標準モードでは、この規格で要求される順序によってプログラムが破壊されてしまう可能性があります。この場合、`-features=no%strictdestrorder` コマンドオプションを使用すると、破棄の厳格な順序付けが行われなくなります。

副集合の初期設定

(中括弧初期設定を使用できる型に関し) クラスオブジェクトの中括弧初期設定を使用している場合、C++ 標準では、それ自体が集合クラスであるメンバーは、自分自身の型の値で初期設定できるようになりました。以下のコード例を参照してください。

```
struct S { // 集合の型
    int i, j;
};
struct T { // 集合の型
    S s; // 集合のメンバー
    int k;
};
T t1 = { {1, 2}, 3 }; // 従来の初期化
S s1 = { 1, 2 };
T t2 = { s1, 3 }; // 副集合の初期設定
```

自分専用の C++ 標準ライブラリの使い方

コンパイラ付属のライブラリの代わりに、自分専用の C++ 標準ライブラリを使用する場合は、`-library=no%Cstd` オプションを指定します。このオプションは、以下のヘッダーを検索しないようにします。

```
<algorithm> <bitset> <complex> <deque> <fstream> <functional>
<iomanip> <ios> <iosfwd> <iostream> <istream> <iterator> <limits>
<list> <locale> <map> <memory> <numeric> <ostream> <queue> <set>
<sstream> <stack> <stdexcept> <streambuf> <string> <strstream>
<utility> <valarray> <vector>
```

`-library=no%Cstd` を指定すると、前述のヘッダーを実装する `libCstd` ライブラリがプログラムに自動的にリンクされなくなります。上記のヘッダーで宣言されている機能を使用するには、`-I` オプションを使用して代わりに使用するヘッダーのあるディレクトリを指定し、プログラムを、代替ヘッダーの実装を含むライブラリまたは一連のオブジェクトファイルとリンクする必要があります。

上に記載したヘッダーの一部だけを入れ替えたり、`libCstd` を、別のライブラリの実装の全部または一部とリンクすることは安全ではありません。たとえば、文字列クラスだけを置き換えて `libCstd` を使用することはできません。コンパイラ付属のライブラリを使用するか、または上記に記載したすべての機能を置き換えてください。

残りのヘッダー (`<exception>`、`<new>`、`<typeinfo>`、および C から継承したすべてのヘッダー) は、コンパイラ自体または Solaris にとって必要不可欠で、`-library=no%Cstd` オプションでその使用の有無を制御することはできません。また、ライブラリ `libCrun` のリンクも `-library=no%Cstd` オプションでは制御できません。

`libCrun` の機能を置き換えることはできません。標準ライブラリを置き換える場合は、コードをコンパイラ付属の `<exception>`、`<new>`、および `<typeinfo>` と共にコンパイルする必要があります。標準モード (デフォルトモード) では、C++ プログラムは必ず `libCrun` にリンクしてください。

注 - このオプションは、自分の責任において使用してください。自分専用のバージョンの C++ 標準ライブラリを使用しても、最適な結果が得られないことがあります。

キャッシュバージョン指定

C++ コンパイラはキャッシュバージョンの違いを検出し、適切なエラーメッセージを出す機能があります。コンパイラでは、それぞれのテンプレートキャッシュディレクトリにテンプレートのキャッシュバージョンを固有に識別するバージョンを示す文字列を付けます。コンパイラの後続のリリースもキャッシュのバージョンを示す文字列を使用しますが、これらのバージョンは現行のバージョンと違うことがあります。

このコンパイラと後続のコンパイラはキャッシュディレクトリ内からバージョンストリングを検出して、適宜エラーを出します。たとえば、後続のコンパイラは異なるテンプレートキャッシュバージョンを使用し、このコンパイラのリリースによって生成されるキャッシュディレクトリを処理しようとする以下のエラーを出力する場合があります。

```
SunWS_cache: Error: Database version mismatch
/SunWS_cache/CC_version
```

同様に、このリリースのコンパイラでは、将来リリースされるコンパイラによって生成されたキャッシュディレクトリを検出するとエラーを出します。

Sun WorkShop C++ コンパイラ 5.0 によって生成されるテンプレートキャッシュディレクトリにはバージョン指定が行われていません。ただし、Sun WorkShop 6 C++ コンパイラでは、エラーや警告が出されることなくキャッシュディレクトリが処理されます。これらのキャッシュディレクトリは、Sun WorkShop 6 C++ コンパイラによって使用されるキャッシュディレクトリ形式に変換されます。

Sun WorkShop 6 C++ コンパイラまたはそれ以降のリリースによって生成されるテンプレートキャッシュディレクトリは、Sun WorkShop C++ コンパイラ 5.0 では使用できません。Sun WorkShop C++ コンパイラ 5.0 は、形式の違いを認識できないので、エラーを出します。

Sun WorkShop 6 Fortran コンパイラ

Sun WorkShop 6 には、Sun WorkShop™ 6 Fortran 77 と Sun WorkShop™ 6 Fortran 95 の各コンパイラが含まれます。

Sun WorkShop 6 Fortran 77 コンパイラ

表 3 は、Sun WorkShop 6 Fortran 77 コンパイラの新機能です。一部の機能について詳細は、16 ページの「Fortran コンパイラの新機能」を参照してください。

表 3 Sun WorkShop 6 FORTRAN 77 コンパイラ

機能	説明
<code>FORM="BINARY"</code> 入出力拡張	この新しいオプションを <code>OPEN(..)</code> 文の中で指定すると、レコードマークがない順番探査ファイル (書式なし) としてファイルが取り扱われます。この結果、連続したバイトストリームとしてデータが読み書きされ、他のベンダーシステムとの互換性が確保されます。この機能は、Fortran 95 と Fortran 77 の両方のコンパイラに実装されています。
最適化コードのデバッグ	新リリースでは <code>-g</code> を使用したコンパイルに対する制約が緩和され、デバッグ用 (<code>-g</code>) のフラグとともに <code>-O4</code> と <code>-O5</code> でのコンパイル、またはすべての並列化フラグ (<code>-parallel</code> 、 <code>-explicitpar</code> 、 <code>-autopar</code>) によるコンパイルが可能になりました。
新しいコマンド行フラグ	新リリースの <code>f77</code> には、次のコマンド行フラグが追加されています (<code>f77(1)</code> マニュアルページ参照)。 <ul style="list-style-type: none">■ <code>-aligncommon</code> - 指定のバイト境界に共通ブロック要素を整列します。■ <code>-r8const</code> - 単精度データ定数を <code>REAL*8</code> に拡張します。■ <code>-xmalign</code> - データ要素の一般的整列を指定します。

表 3 Sun WorkShop 6 FORTRAN 77 コンパイラ

機能	説明
拡張コマンド行フラグ	次の <code>f77</code> コマンド行フラグが拡張されました (<code>f77(1)</code> マニュアルページ参照): <ul style="list-style-type: none"> ■ <code>-fast --05</code>、<code>-fsimple=2</code>、<code>-xvector=yes</code>、<code>-pad=common</code> を設定します。 ■ <code>-xprefetch</code> - 明示的なプラグマ <code>prefetch</code> 命令を使用可能にし、UltraSPARC プラットフォームで先読み命令を生成します。 ■ <code>-xtypemap</code> - データ型のサイズを指定することができます。
Cray スタイルの指令 ¹	Cray スタイルの並列化指令に AUTOSCOPE を追加。
ハイパーリンクされた診断	「Sun WorkShop 構築」ウィンドウの <code>f77</code> のエラーメッセージには、メッセージを説明するヘルプページへのハイパーリンクが追加されました。

1) ライセンス供与Fortran 77 コンパイラの並列化機能には、Sun WorkShop HPC ライセンスが必要です。

Sun WorkShop 6 Fortran 95 コンパイラ

表 4 は、Sun WorkShop 6 Fortran 95 コンパイラの新機能です。一部の機能について詳細は、16 ページの「Fortran コンパイラの新機能」を参照してください。

表 4 Sun WorkShop 6 Fortran 95 コンパイラ

機能	説明
標準への準拠	Fortran 95 コンパイラは、Fortran 95 の規格に完全に準拠していません。
新しいコマンド	Fortran 95 コンパイラは、 <code>f90</code> と <code>f95</code> のどちらのコマンドによっても起動できます。 <code>f95</code> コマンドは新しいコマンドです。 <code>f90</code> は <code>f95</code> と同等のコマンドです。
ファイル拡張子	コンパイラは、拡張子 <code>.f90</code> と <code>.F90</code> のほか、 <code>.f95</code> と <code>.F95</code> のソースファイルも受け付けます。

表 4 Sun WorkShop 6 Fortran 95 コンパイラ (続き)

機能	説明
FORM="BINARY" 入出力拡張機能	この新オプションを OPEN(..) 文の中で指定すると、レコードマークなしの順番探査ファイル (書式なし) としてファイルが取り扱われます。この結果、連続したバイトストリームとしてデータが読み書きされ、他のベンダーシステムとの互換性が確保されます。この機能は、Fortran 95 と Fortran 77 の両方のコンパイラで実装されています。『FORTRAN 77 言語リファレンス』を参照してください。
最適化コードの デバッグ	-g を使用したコンパイルに対する制約が緩和され、デバッグ用 (-g) のフラグとともに -O4 と -O5 でのコンパイルや、すべての並列化フラグ (-parallel、-explicitpar、-autopar) によるコンパイルが可能になりました。
F77 フラグ	f77 コンパイラフラグのほとんどは、f95 で実装されています。詳細については、f95(1) マニュアルページを参照してください。具体的なフラグは、次のとおりです。
	-erroff 指定したエラーメッセージを非表示にします。
	-errtags タグ付きのエラーメッセージを表示します。
	-ext_names 下線付きまたは下線なしの外部名を作成します。
	-fpp ソースコードプリプロセッサを指定します。
	-loopinfo どのループが並列化されているかを示します。
	-sbfast ブラウザテーブル情報を作成します。
	-silent コンパイラメッセージを抑制します。
	-U 小文字と大文字の区別を許可します。
	-u IMPLICIT NONE を意味します。
	-xcrossfile ファイル間の最適化を使用可能にします。
	-xF アナライザのための関数レベルの順序変更を可能にします。
	-xinline 関数をインラインでコンパイルします。
	-xtypemap デフォルトのデータサイズを指定します。
新しいフラグ	次の新しいフラグが f95 で実装されています。
	-aligncommon 指定のバイト境界に共通ブロック要素を整列します。
	-mp=openmp OpenMP 指令を受け付けます。

表 4 Sun WorkShop 6 Fortran 95 コンパイラ (続き)

機能	説明
	<p><code>-r8const</code> 単精度定数を REAL*8 に高めます。</p> <p><code>-xia</code> 区間演算拡張機能の処理を使用可能にします。(推奨)</p> <p><code>-xinterval</code> 区間演算拡張機能の処理を使用可能にします。</p> <p><code>-xmalign</code> データ要素の適切な境界への整列を指定します。</p> <p><code>-xrecursive</code> RECURSIVE 属性なしの再帰的呼び出しを許可します。</p>
拡張フラグ	<p>以下の <code>f95</code> コマンド行フラグが拡張されました (<code>f95(1)</code> マニュアルページ参照)。</p> <p><code>-fast</code> <code>-05 -fsimple=2 -xvector=yes -pad=common</code> を設定します。</p> <p><code>-xprefetch</code> 明示的なプラグマ <code>prefetch</code> 指令を使用可能にし、UltraSPARC プラットフォームで先読み命令を生成します。</p> <p><code>-xtypemap</code> 可能なデータ型指定の拡張セットを含んでいます。</p>
OpenMP	<p>このリリースの Fortran 95 は、ソースコード指令のセット、実行時ライブラリルーチン、環境変数を含む、明示的並列化のための OpenMP インタフェースを実装しています。『Fortran ユーザーズガイド』を参照してください。</p>
Cray スタイルの指令	<p>Cray スタイルの並列化ディレクティブに追加された <code>AUTOSCOPE</code>。</p>
区間演算	<p>このリリースの Fortran 95 では、組み込みの <code>INTERVAL</code> データ型のサポートが実装されています。</p>
ライセンス	<p>Fortran 95 コンパイラの並列化機能には、Sun WorkShop HPC ライセンスが必要です。</p>
ハイパーリンクされた診断	<p>Sun WorkShop 「構築」ウィンドウでは、<code>f95</code> のエラーメッセージをオンラインヘルプへのリンクとして解釈されるようになりました。</p>

Fortran コンパイラの新機能

以下の節では、Fortran コンパイラのいくつかの新機能についてさらに詳しく説明します。

入出力操作に対する FORM="BINARY" の効果

- **WRITE** 文：データはバイナリとしてファイルに書き込まれ、出力リストに指定されているバイト数が転送されます。
- **READ** 文：データは入力リスト上の変数に読み込まれ、同リストで要求されているバイト数が転送されます。ファイルにはレコードマークがないので、「記録終了」のエラー検出は行われません。検出対象のエラーは、「ファイル終了」やシステム異常エラーだけです。
- **INQUIRE** 文：FORM="BINARY" で開いたファイルに対する **INQUIRE** は、以下を返します。

```
FORM="BINARY"  
ACCESS="SEQUENTIAL"  
SEQUENTIAL="YES"  
DIRECT="NO"  
FORMATTED="NO"  
UNFORMATTED="YES"  
RECL= と NEXTREC= は未定義。
```

- **BACKSPACE** 文：使用は許可されていないので、エラーが出力されます。
- **ENDFILE** 文：通常どおり、現在位置でファイルを切り捨てます。
- **REWIND** 文：通常どおり、データの先頭にファイルを再配置します。

OpenMP

このリリースの Fortran 95 は、ソースコード指令のセット、実行時ライブラリルーチン、環境変数などをはじめとする、明示的並列化のための OpenMP インタフェースを実装しています。OpenMP の暫定ドキュメントが README として入っています。OpenMP の仕様については、<http://www.openmp.org/> を参照してください。

Fortran コンパイラで受け付ける OpenMP を含むすべての指令については、『Fortran ユーザーズガイド』の付録 E にまとめられています。Fortran コンパイラの並列化機能について詳しくは、『Fortran プログラミングガイド』を参照してください。

注 – Fortran のコンパイラの並列化機能には、Sun WorkShop HPC ライセンスが必要です。

区間演算拡張機能

このリリースの Fortran 95 には、区間演算拡張機能が実装されています。「Sun WorkShop 6 Fortran 95 区間演算」を参照してください。

ハイパーリンクされたコンパイラの診断

Sun WorkShop を使用してアプリケーションの構築やコンパイルを行う場合、「構築」ウィンドウ内の `f77` と `f95` の診断メッセージにヘルプページへのハイパーリンクが追加されました。このエラーメッセージをクリックすると、ヘルプブラウザが立ち上がり、特定のエラー診断に関する詳しい情報が表示されます。

Fortran プラットフォーム

この Sun WorkShop 6 リリースの Fortran コンパイラでサポートしている Solaris SPARC™ プラットフォーム版オペレーティング環境のバージョンは、2.6、7、および 8 に限定されます。Solaris Intel IA-32 プラットフォームについては、サンは Fortran のコンパイラと Sun Performance Library の開発を中止しました。このリリースでは、サンは Solaris Intel IA-32 プラットフォーム用の Forte™ for High Performance Computing または Forte™ Fortran デスクトップ版 (かつての Sun Performance WorkShop Fortran) を提供していません。Solaris Intel IA-32 プラットフォームのソフトウェア開発支援ツールの製品ラインについては、Portland Group (<http://www.pgroup.com>) に問い合わせてください。

Sun Workshop 6 Fortran 95 区間演算

組み込み `INTERVAL` データ型のサポートは、Sun WorkShop 6 Fortran 95 コンパイラの新しい機能です。

新しい 2 つのコンパイラフラグ `-xia` と `-xinterval` により、コンパイラは区間特有の言語を認識し、区間命令を実現するためのコードを生成します。

区間演算とは

区間演算を使用すると、各区間に含まれる数の集合について算術演算の値が求められます。区間とは、その区間の下位の境界から上位の境界までの間にあるすべての実数の集合を意味します。区間演算結果はどれもすべての可能な結果の集合が必ず含まれるひとつの新しい区間です。

Sun WorkShop 6 Fortran 95 を使用すれば、区間プログラムを作成して算術演算の値で厳密な境界を計算するのは容易です。

- 変数を `INTERVAL` 型として宣言します。
- `INTERVAL` 組み込み関数や演算子、関係演算子、および書式編集記述子を使用して通常の Fortran コードを作成します。
- `-xia` コマンド行オプションを使用してコードをコンパイルします。

最適な結果を得るためには、狭幅の区間結果を計算する既存の区間アルゴリズムを使用します。そして、この狭幅の区間結果を計算するアルゴリズムを考案することが、区間分析の主な目的です。

区間演算が重要である理由

区間演算は次の理由により重要です。

- 区間演算を使用すると、入力データエラー、マシンの丸め、それらの相互作用など、あらゆるソースからのエラーに確実な範囲を設けてコンピュータで計算を実行できます。
- 区間アルゴリズムにより、非線形的な問題の解決、たとえば、方程式の非線形システムや非線形プログラミングを行なうことが可能になります。

区間がより一般的になるにつれて、区間ソルバーのライブラリが線形および非線形を問わずあらゆる問題に対するシャープな区間解決法の計算に利用できるようになり、さらにエラーのあらゆる原因を考慮できるようになります。これらのライブラリを用いることによって、科学者やエンジニア、商用アプリケーションの開発者は、それまでは手の届かなかった問題を解決するプログラムを開発できるようになります。

詳細情報の入手先

『Fortran 95 区間演算プログラミングリファレンス』または区間演算の README に記載されているオンラインリソースの一覧を参照してください。

Sun Workshop 6 dbx

表 5 は、Sun WorkShop 6 dbx の新機能です。

表 5 Sun WorkShop 6 dbx

機能	説明
<code>firedhandlers</code> ksh 変数	読み取り専用 ksh 変数 <code>firedhandlers</code> が追加されています。この変数を <code>delete</code> コマンドと <code>handler</code> コマンドと併用すれば、 <code>clear</code> コマンドの代わりとして使用できます。詳細は、Sun WorkShop オンラインヘルプの「dbx コマンドの使い方」の「変数」、「 <code>delete</code> コマンド」、「 <code>handler</code> コマンド」を参照してください。
ブレークポイントの部分的な消去	新リリースの <code>clear</code> コマンドによって、クラス内、メソッド内、関数内のブレークポイントを部分的に消去することが容易になります。詳細は、Sun WorkShop オンラインヘルプの「dbx コマンドの使い方」の「 <code>clear</code> コマンド」を参照してください。
トレース出力	トレースの出力をファイルにリダイレクトできます。詳細は、Sun WorkShop オンラインヘルプの「dbx コマンドの使い方」の「 <code>trace</code> コマンド」を参照してください。

表 5 Sun WorkShop 6 dbx (続き)

機能	説明
新しい dbx 環境変数	<ul style="list-style-type: none"> <li data-bbox="606 244 1229 505">■ 新しい dbx 環境変数 <code>stack_find_source</code> は、プログラム停止時にデバッグ可能ソースコードが含まれているフレームまでコールスタックを dbx が自動的に上方移動させるかどうかを制御します。詳細は、Sun WorkShop オンラインヘルプの「dbx コマンドの使い方」の「<code>stack_find_source</code> 環境変数」を参照してください。 <li data-bbox="606 517 1229 734">■ 新しい dbx 環境変数 <code>proc_exclusive_attach</code> は、別のデバッガやデバッグツールの制御下にあるプロセスに dbx を接続できるかどうかを制御します。詳細は、Sun WorkShop オンラインヘルプの「dbx コマンドの使い方」の「<code>proc_exclusive_attach</code> 環境変数」を参照してください。 <li data-bbox="606 746 1229 963">■ 新しい dbx 環境変数 <code>step_granularity</code> は、<code>step</code> コマンドと <code>next</code> コマンドが文、行、ソースコードのどれを処理対象とするかを制御します。詳細は、Sun WorkShop オンラインヘルプの「dbx コマンドの使い方」の「<code>step_granularity</code> 環境変数」を参照してください。 <li data-bbox="606 975 1229 1192">■ 新しい dbx 環境変数 <code>mt_scalable</code> は、資源の使用量を削減することによって、多くの LWP (軽量プロセス) を使うマルチスレッドアプリケーションのデバッグを行いやすくします。詳細は、Sun WorkShop オンラインヘルプの「dbx コマンドの使い方」の「<code>mt_scalable</code> 環境変数」を参照してください。 <li data-bbox="606 1204 1229 1420">■ 新しい dbx 環境変数 <code>rtc_error_stack</code> は、スタックトレースが RTC 内部構造に対応するフレームを表示するかどうかを決定します。詳細は、Sun WorkShop オンラインヘルプの「dbx コマンドの使い方」の「<code>rtc_error_stack</code> 環境変数」を参照してください。

表 5 Sun WorkShop 6 dbx (続き)

機能	説明
	<ul style="list-style-type: none"> ■ 新しい dbx 環境変数 <code>rtc_inherit</code> は、デバッグプログラムから実行されている子プロセスで実行時検査を行うかどうかを決定します。詳細は、Sun WorkShop オンラインヘルプの「dbx コマンドの使い方」の「<code>rtc_inherit</code> 環境変数」を参照してください。 ■ 新しい dbx 環境変数 <code>rtc_use_traps</code> は、実行時チェックでの 8 メガバイトコード制限の回避を有効にします。詳細は、Sun WorkShop オンラインヘルプの「dbx コマンドの使い方」の「<code>rtc_use_traps</code> 環境変数」を参照してください。
C++ メンバー関数におけるブレークポイント	C++ インラインメンバー関数のブレークポイントは、正しく動作します。ハンドラ条件 (<code>-if</code>) も正しく評価されます。
LWP 関連コマンド	このリリースでは、マルチスレッドアプリケーションのデバッグ時だけではなく、いつでも LWP 関連のコマンドを利用できます。詳細は、Sun WorkShop オンラインヘルプの「dbx コマンドの使用」の「 <code>lwp</code> コマンドと <code>lwps</code> コマンド」を参照してください。
実行プロセスの割り込み	Control-C を押してもハングしたプロセスが停止しない場合は、3 回連続して Control-C を押すと、ハングしたプロセスを強制的に停止できます。Sun WorkShop オンラインヘルプの「dbx コマンドの使い方」の節の「実行プロセスの割り込み」を参照してください。
全行番号情報	<p><code>-g</code> オプションと <code>-o</code> オプションによるコンパイル時に、全行番号情報が記録されるようになりました。これで、<code>step</code> コマンドと <code>next</code> コマンドを使用して、最適化コードを順次実行していくことができます。ただし、現行の行は、コードスケジューリングの都合上、順方向と逆方向にジャンプします。dbx 内から出力される変数値は、最適化コードでは引き続き信頼に欠けます。</p>

Sun WorkShop 6

表 6 は、Sun WorkShop 6 の新機能です。一部の製品の機能については、表の次の節でさらに詳しく説明しています。

表 6 Sun WorkShop 6

機能	説明
テキスト編集	
式のパレーン評価	式のパレーン評価は、テキストエディタで現在マウスでポイントしている数式の現在値をその場で表示します。数式のタイプを調べたり、ポインタを間接参照したりすることもできます。
NEdit と Vim テキストエディタ	今回のリリースでは NEdit と Vim が統合型テキストエディタとして加わりました。(英語環境でのみ使用できます。日本語環境では使用できません。)
プログラムのデバッグ	
「ボタン編集」 ウィンドウ	「ボタン編集」ウィンドウにより、次の処理が行えます。エディタウィンドウと「デバッグ」ウィンドウのツールバーのカスタマイズ。「デバッグ用カスタムボタン」ウィンドウのボタンの追加、削除、または編集。
デバッグウィンドウ	「データ履歴」区画が、「データ履歴」タブ、「プログラム入出力」タブ、それにオプションの「データ表示」タブと置き換えられました。 「セッション」区画と「スレッド」区画を切り替えるためのラジオボタンが、「セッション」タブ、「スレッド」タブ、および「ブレークポイント」タブと置き換えられました。
「デバッグオプション」 ダイアログボックス	オプションのカテゴリを切り替える「カテゴリ」リストボックスは、ダイアログボックスの右側のタブに置き換えられています。

表 6 Sun WorkShop 6 (続き)

機能	説明
新しいデバッグオプション	<p data-bbox="648 244 1293 305">「デバッグ」ウィンドウの「プログラム入出力」タブにプログラムへの入出力指示に関する選択肢が追加されました。</p> <p data-bbox="648 336 1293 470">デバッグ可能なソースコードではないプログラム内の関数で実行が停止した場合に、コールスタックを最初のデバッグ可能ソースコードまでポップするよう選択できるようになりました。</p> <p data-bbox="648 501 1293 595">実行単位を「行」に設定するよう選択できるようになったため、含まれている文の数に関係なく、次の 1 コマンドを特定の 1 行に対して実行できます。</p> <p data-bbox="648 626 1293 760">デフォルトの設定では、ツールバーの下に「状態」、「停止位置」、「評価コンテキスト」、セッション数などの情報は表示されなくなりましたが、従来どおり表示されるよう選択することもできます。</p> <p data-bbox="648 791 1293 885">プログラムの <code>main()</code> モジュールがデバッグ情報付きでコンパイルされない場合でも、デバッガが警告を出させないよう選択できるようになりました。</p> <p data-bbox="648 916 1293 1010">「データ表示」ウィンドウを「デバッグ」ウィンドウのタブとして、または別のウィンドウとして表示するよう選択できるようになりました。</p> <p data-bbox="648 1041 1293 1175"><code>-xs</code> フラグ付きでコンパイルされたモジュールをデバッグセッションの開始時に読み込むのではなく、それらのモジュールのデバッグ情報が必要となった時点で読み込むよう選択できるようになりました。</p> <p data-bbox="648 1206 1293 1302">多数 (300 以上) の LWP (軽量プロセス) を使用するプロセスをデバッグする必要がある場合、デバッグ時にリソースの使用量が少なくなるよう設定できます。</p> <p data-bbox="648 1333 1293 1394">デバッガに <code>dbx</code> がデバッグ対象プログラムを排他的に制御しているかどうかをチェックさせないよう選択できます。</p> <p data-bbox="648 1425 1293 1486">デバッガが、<code>dbx</code> の機能にとって欠かせない一定の共有ライブラリを除外できるようにすることが可能になりました。</p>

表 6 Sun WorkShop 6 (続き)

機能	説明
削除されたデバッグオプション	「マルチスレッド使用のプログラムのウォッチポイントを許可」のオプションが「デバッグオプション」ダイアログから削除されました。
プロジェクトの取り扱い	
プロジェクトとワークセット	この Sun WorkShop のリリースでは、プロジェクトを使って開発プロジェクトに関連付けられたファイル、プログラム、およびターゲットを追跡し、メイクファイルを作成しなくてもプログラムを構築できます。
新しいマニュアルページ	
makeprd(1)	Sun WorkShop プロジェクトファイルビルダ
nedit(1)	Motif UI スタイルのテキストエディタ (英語のみ)
vim(1)	vi の改良版といえるプログラマ用テキストエディタ (英語のみ)
xemacs(1)	次世代 Emacs

テキスト編集

以下の 2 つの節は、新しく追加された Sun WorkShop 6 テキスト編集機能を説明しています。

式のバルーン評価

式のバルーン評価は、テキストエディタにおいて現時点でマウスでポイントしている数式の現在値を即時に表示します。数式のタイプを調べたり、ポインタを間接参照したりすることもできます。詳細については、オンラインヘルプの「テキスト編集」の「式のバルーン評価」の使用を参照してください。

NEdit と Vim テキストエディタ

テキストエディタは、構築、デバッグ、ブラウザを含む Sun WorkShop 統合開発支援ツールセットの中心に位置します。Sun WorkShop プログラミング環境では、数式の評価、ブレークポイントの設定、関数の検討をテキストエディタから行うことができます。

X/Motif システムへのグラフィカルユーザーインターフェースを備えたプレーンテキストエディタである NEdit、および UNIX システムでの改良版 vi 標準テキストエディタである Vim は、このリリースでの新しい統合テキストエディタです。以下は、Sun WorkShop 6 とともに提供される統合エディタのリストです。

- NEdit (version 5.1)
- XEmacs (version 20.4 またはこれと互換のあるバージョン)
- GNU Emacs (version 19.28 またはこれと互換のあるバージョン)
- Vi
- Vim (version 5.3、グラフィカルユーザーインターフェースオプションあり)

各エディタの選び方の詳細については、以下を参照してください。

- エディタのメニューバーのヘルプメニューで利用できるオンラインマニュアル
- オンラインヘルプの「テキスト編集」の「テキストエディタオプション」ダイアログボックス

プログラムのデバッグ

以下の節は、新しく追加されたデバッグ機能について説明しています。

「ボタン編集」ウィンドウ

「ボタン編集」ウィンドウにより、エディタウィンドウと「デバッグ」ウィンドウのツールバーをカスタマイズしたり、「デバッグ用カスタムボタン」ウィンドウのボタンを追加、削除、または編集したりできるようになりました。

詳細については、「ツールバーオプション」を参照してください。

「デバッグ」ウィンドウ

以下の節では、「デバッグ」ウィンドウに組み込まれたいくつかの変更について説明しています。

セッションステータスとコンテキスト情報

デフォルトの設定では、「状態」、「停止位置」、「評価コンテキスト」、「セッション数」などの情報は、これまでのようにはツールバーの下に表示されません。この情報を表示させたい場合は、「デバッグオプション」ダイアログの「ウィンドウレ

「アウト」タブの中の「画面の上部にコンテキスト / ステータスの 3 行を表示」を選択します。詳細については、Sun WorkShop オンラインヘルプの「デバッグウィンドウの使い方」の「セッションステータス情報とコンテキスト情報の表示」を参照してください。

「データ」区画

「データ」区画は、「データ履歴」と「プログラム入出力」のタブ、およびオプションの「データ表示」タブと置き換えられました。

- 「データ履歴」タブには、「データ」区画が表示されます。詳細については、Sun WorkShop オンラインヘルプの「デバッグウィンドウの使い方」の「データ履歴」タブを参照してください。
- 「プログラム入出力」タブは、別の「プログラム入出力」ウィンドウではなく、「デバッグ」ウィンドウ内にプログラムの入出力に関する情報を表示します。デフォルトではこのタブにプログラムの入出力情報が表示されるようになりましたが、「デバッグオプション」ダイアログボックスの「デバッグ出力」タブの「プログラム出力」区画を使用して、プログラムの入出力を行なう場所を選択できます。詳細については、Sun WorkShop オンラインヘルプの「デバッグウィンドウの使い方」の「データ履歴」タブを参照してください。
- 「データ表示」タブは、「デバッグオプション」ダイアログボックスの「データ表示ウィンドウ」タブの「データ表示ウィンドウ 以下として表示」区画で「デバッグウィンドウのタブ」を選択した場合に (別の「データ表示」ウィンドウの代わりに) 含まれます。詳細については、Sun WorkShop オンラインヘルプの「デバッグウィンドウの使い方」の「データ履歴」タブを参照してください。

セッション区画とスレッド区画

「セッション」区画と「スレッド」区画を切り替えるためのラジオボタンが、「セッション」、「スレッド」、および「ブレークポイント」のタブと置き換えられました。

- 「セッション」タブは「セッション」区画を表示します。このタブの上で右マウスボタンを押すと、ポップアップメニューが表示されます。「セッションタブ」を参照してください。詳細については、Sun WorkShop オンラインヘルプの「デバッグウィンドウの使い方」の「データ履歴」タブを参照してください。
- 「スレッド」タブは「スレッド」区画を表示します。「スレッドタブ」を参照してください。詳細については、Sun WorkShop オンラインヘルプの「デバッグウィンドウの使い方」の「データ履歴」タブを参照してください。

- 「ブレークポイント」タブは、プログラムに割り当てられているブレークポイントやトレースポイントをスクロール可能な一覧として表示します。このタブの上で右マウスボタンを押すと、ポップアップメニューが表示されるので、各ブレークポイントごとに瞬時に使用可能、使用禁止、削除、あるいはソースを表示することができます。このポップアップメニューには「追加」項目が含まれており、この項目を選択すると、ブレークポイントの追加、使用可能、使用禁止、変更、削除などの操作を行える別の「ブレークポイント」ウィンドウが表示されます。詳細については、Sun WorkShop オンラインヘルプの「デバッグウィンドウの使い方」の「データ履歴」タブを参照してください。

「デバッグオプション」ダイアログボックス

「デバッグオプション」ダイアログボックスには、以下の新機能が追加されました。

「カテゴリ」タブ

「デバッグオプション」ダイアログボックスの「カテゴリ」リストボックスは、オプションのカテゴリを切り替えるためのウィンドウの右側に表示されるタブと置き換えられました。

新しい「プログラム出力」オプション

プログラムの入出力を、「デバッグ」ウィンドウの「プログラム入出力」タブに変更する選択肢が追加されました。以前と同様、別の「プログラム入出力」ウィンドウ、「dbx コマンド」ウィンドウ、あるいはカスタム `pty` にプログラムの入出力先を変更することもできます。「プログラムの入出力先の変更」を参照してください。

新しい「コールスタック」オプション

デバッグセッション中、プログラムが `-g` フラグ付きでコンパイルされなかった場合、デバッグ可能なソースコードでないプログラム内の関数で実行が停止する場合があります。このバージョンでは、コールスタックをこのケースで最初のデバッグ可能なソースコードにポップするよう選択できるようになりました。詳細については、Sun WorkShop オンラインヘルプの「デバッグウィンドウの使い方」の「実行停止時にスタックをさかのぼる」を参照してください。

新しい「ステップ実行」オプション

デフォルトの設定では、デバッグのステップ実行単位は「文」に設定されます。したがって、1つのソースコード行に複数の文が含まれている場合、その行をステップ実行するのにその文の数だけ `next` コマンドが必要です。新バージョンでは、ステップ実行単位に「行」を設定できるようになったため、行に含まれている文の数に関係なく、`next` コマンド1つで行全体をステップ実行できます。詳細については、Sun WorkShop オンラインヘルプの「ステップ実行単位の設定」を参照してください。

新しい「ウィンドウレイアウト」オプション

デフォルトの設定では、「状態」、「停止位置」、「評価コンテキスト」、および「セッション数」の情報は、ツールバーの下に表示されなくなりました。これらの情報を表示させたい場合には、「デバッグオプション」ダイアログボックスの「ウィンドウレイアウト」タブで「画面の上部にコンテキスト / ステータスの3行を表示」を選択します。詳細については、Sun WorkShop オンラインヘルプの「セッションステータス情報とコンテキスト情報の表示」を参照してください。

新しい「ウィンドウ動作」オプション

デフォルトの設定では、デバッガはプログラムの `main()` モジュールにデバッグ情報が付随していない場合に警告を出します。新バージョンでは、この警告を表示させないようにすることができます。詳細については、Sun WorkShop オンラインヘルプの「デバッグ情報を使用しないで `main()` モジュールをコンパイルした時に警告を表示させる」を参照してください。

新しい「データ表示」オプション

「データ表示」ウィンドウを「デバッグ」ウィンドウのタブとして表示させたり、別のウィンドウとして表示させるできるようになりました。詳細については、Sun WorkShop オンラインヘルプの「データ表示」ウィンドウの表示方法を参照してください。

新しい「デバッグパフォーマンス」オプション

`-xs` フラグ付きでコンパイルされたモジュールを、デバッグセッションの開始時に読み込むのではなく、それらのモジュールのデバッグ情報が必要となった時点で読み込むよう選択できるようになりました。このオプションの追加により、モジュールが `-xs` フラグ付きでコンパイルされたときのデバッグ開始時間を短縮することができます。

す。デフォルトでは、このデバッグオプションはオンです。詳細については、Sun WorkShop オンラインヘルプの「[-xs](#) を使ってコンパイルしたモジュールの読み込みを遅延する」を参照してください。

新しい「フォークおよびスレッド」オプション

多数 (最大 300) の LWP (軽量プロセス) を伴うプロセスをデバッグする場合、新しくデバッグ時のリソース使用量を削減するよう設定できるようになりました。ただし、それによってデバッグのパフォーマンスは低下することがあります。詳細については、Sun WorkShop オンラインヘルプの「[大量の LWP のデバッグ](#)」を参照してください。

新しい拡張オプション

デフォルトの設定では、デバッガは、[dbx](#) がデバッグ対象プログラムを排他的に制御するかどうかを検査します。このため、[dbx](#) は他のツールがすでにそのプログラムに接続されていると、そのプログラムに接続できません。このバージョンでは、この動作を解除することができるようになっています。詳細については、Sun WorkShop オンラインヘルプの「[プログラムの排他的制御が dbx にあることを確認する](#)」を参照してください。

デフォルトでは、デバッガは [dbx](#) の機能にとって重要な一部の共有ライブラリの除外を許可していません。新バージョンでは、これらのライブラリを除外できるようになりましたが、その場合には中核のファイルしかデバッグできません。詳細については、Sun WorkShop オンラインヘルプの「[重要な dbx ライブラリの取り込みが必要](#)」を参照してください。

プロジェクトの取り扱い

この Sun WorkShop のリリースでは、プロジェクトを使って開発プロジェクトに関連付けられたファイル、プログラム、およびターゲットを追跡し、メイクファイルを作成しなくてもプログラムを構築することができます。プロジェクトは、実行プログラム、静的ライブラリまたは静的アーカイブ、共用ライブラリ、Fortran アプリケーション、複雑なアプリケーション、またはユーザーメイクファイルアプリケーションの構築に使用されるファイルおよびコンパイラ、デバッガ、構築関連のオプションが含まれるリストです。

以前のバージョンの Sun WorkShop では、プロジェクトの代わりにワークセットが使用されていました。プロジェクトは依然としてワークセットの性質を持っていますが (プロジェクトに関連するファイルやディレクトリを簡単にアクセスできるようにするメニューピクリストは引き続き存在)、プロジェクトはユーザーがどのようなソースファイルをどのように作成したいか、ユーザーのプログラムをよく把握しています。Sun WorkShop のワークセットが手元にある場合は、それらを読み込み時に自動的に Sun WorkShop 6 のプロジェクトに変換できます。詳細については、Sun WorkShop オンラインヘルプの「ワークセットをプロジェクトに変換する」を参照してください。

Sun WorkShop を起動すると、「Sun WorkShop によろこそ」ダイアログボックスが開き、すぐに Sun WorkShop プロジェクトとプロジェクトウィザードにアクセスできます。Sun WorkShop には、WorkShop のメインウィンドウの「プロジェクトメニュー」から利用できるプロジェクト機能があり、以下を実行できます。

- プロジェクトウィザードと自分専用のメイクファイルまたは Sun WorkShop が作成してくれるメイクファイルによる新しいプログラムの作成や単純なプログラムの構築 (オンラインヘルプの「プロジェクトの取り扱い」の「新規プロジェクトの作成」を参照)。
- 既存のプロジェクトの設定値の変更。設定値には、プロジェクトのコンパイル方法やソースブラウズ情報を生成させるかどうかの選択があります (オンラインヘルプの「プロジェクトの取り扱い」の「WorkShop のターゲット」を参照)。

このほか、プロジェクトを読み込まない状態で Sun WorkShop 6 プログラミング環境を使用することも可能です。ピクリストは、開発プロジェクトに関連するファイル、プログラム、ディレクトリ、ターゲットを記録します (詳細については、「WorkShop のターゲット」を参照)。各ファイルをアクセスし、ターゲットを構築し、実行形式ファイルをデバッグするといった操作を、WorkShop のメインウィンドウから行なうことができます。構築ターゲット情報は常に一定ではないため編集できません。この情報は、構築ターゲットへアクセス、追加、削除するたびに変更されません。

Sun WorkShop 6 TeamWare

表 7 は、Sun WorkShop 6 TeamWare の新機能です。

表 7 Sun WorkShop 6 TeamWare

機能	説明
自動フリーズポイント機能	自動フリーズポイント機能は、特定のトランザクションの前または後にフリーズポイントファイルを自動的に作成します。フリーズポイントファイルの作成する時間をブリングオーバー、プットバック、取消し時、または衝突解決の前または後にするかを選択できます。
「バージョン管理」の「カスタマイズメニュー」	「カスタマイズメニュー」機能は、「バージョン管理」ウィンドウに「カスタマイズ」という新しいメニューを追加し、ユーザー固有のコマンドにアクセスできるようにします。カスタマイズメニューの作成については、オンラインヘルプの「管理ファイル」の節の「カスタマイズメニューの作成」を参照してください。
デルタコメント	この新オプションは、デルタ番号、所有者、コメントなどのデルタコメントをトランザクション出力と電子メール通知に追加します。「ワークスペース管理」で、「ワークスペース」/「作成ブリングオーバー」/「更新ブリングオーバー」/「プットバック」を選択して「デルタコメント」ボックスを選択するか、または <code>-d</code> オプションを付けて <code>bringover</code> 、 <code>putback</code> コマンドを使用します。
ファイルマージの「相違ナビゲータ」	マージされていない 2 個のファイルの間に、「相違ナビゲータ」が表示されます。「相違ナビゲータ」の左右どちらかにあるスライドボックスをクリックすると 2 個のファイルの内どちらかがスクロールされ、最上部または最下部にある矢印をクリックすると両方のファイルにおいて同じ距離だけ移動します。
メニュー再編成の設定	Sun Workshop TeamWare 6 では、ワークスペース属性のユーザーインタフェースが変更されています。

表 7 Sun WorkShop 6 TeamWare (続き)

機能	説明
ブットバックの妥当性検査	ブットバックの妥当性検査をオンにすると、許可されたワークスペースだけがブットバックを行えるようになります。ブットバックを行えるユーザーを管理し、特定のパスワードを要求するように設定することができます。
SCCS 管理フラグ	ファイルに対して SCCS 管理フラグを設定できます。
ワークスペース完全性検査	<code>workspace</code> コマンドで使用できる新オプション。 <code>check [-W] [-s] wsname ...</code> は、ファイル、アクセスモード、親子関係、履歴ファイルの状態をチェックします。コマンドは、ワークスペースに問題がない場合は 0 を、エラーがある場合は 1 の終了コードで終了します。
ワークスペース履歴表示	このリリースの Sun WorkShop TeamWare には、ワークスペース履歴ファイルに入っている情報を簡単に表示できる機能が含まれています。
ワークスペースのラベル	この機能により、チームメンバーがすぐに理解できるような、意味のある名前をワークスペースに指定できます。「ワークスペース」で「プロパティ」を選択して、「説明」タブを選びます。
新しいマニュアルページ	description(4)

メニュー再編成の設定

Sun Workshop TeamWare 6 では、次のような変更がメニューに加えられています。

表 8 Sun WorkShop 6 TeamWare メニューの変更

TeamWare 2.2	TeamWare 6
ワークスペース管理:	
ファイル ▶ 親ワークスペースの読み込み	ワークスペース ▶親ワークスペースの読み込み
ファイル ▶ 子ワークスペースの読み込み	ワークスペース▶子ワークスペースの読み込み

表 8 Sun WorkShop 6 TeamWare メニューの変更

TeamWare 2.2	TeamWare 6
ファイル ▶ 空の子ワークスペース作成	ワークスペース▶子ワークスペースの作成
編集 ▶削除	ワークスペース▶削除
編集 ▶名前変更	ワークスペース▶名前変更
編集 ▶親	ワークスペース▶親の変更
編集 ▶更新 ▶名前テーブル	ワークスペース ▶名前テーブル更新
トランザクション▶ プリングオーバー ▶作成	アクション ▶作成プリングオーバー
トランザクション▶ プリングオーバー ▶更新	アクション ▶更新プリングオーバー
オプション ▶ワークスペース	ワークスペース▶属性
オプション▶ワークスペース ▶ ロック編集	ワークスペース▶ロック編集
新規	表示 ▶再描画
新規	ワークスペース▶属性 ▶フリーズポイント
新規	ワークスペース▶属性 ▶ プットバックの妥当性検査
新規	オプション ▶ワークスペース管理 ▶子ワークス ペース読み込み: 選択する/ すべて
バージョン管理	
新規	ファイル ▶ファイル情報
新規	コマンド ▶アンチェックアウト

プットバックの妥当性検査

プットバック妥当性検査をオンにすると許可されたワークスペースだけがプットバックを行えるようになります。プットバックを行う場合、「パスワード」(統合要求 ID : Integration Request Identifier) の入力を指示するプロンプトが表示されます。この機能は統合要求 ID を記録するだけで、チェックは行いません。統合要求 ID のチェックを

行うには、そのための妥当性検査プログラムを自分自身で作成する必要があります。詳細については、Sun WorkShop オンラインヘルプの「ワークスペースの管理」の「ブットバックの妥当性検査」を参照してください。

SCCS 管理フラグ

SCCS 管理フラグを設定するには、「バージョン管理」で「ファイル」▶「ファイル情報」を選択します。たとえばブットバック中に「バージョン管理」に MR (修正要求文字列) の入力要求を出させるようにするには、妥当性検査プログラムの名前を「妥当性検査プログラム」テキストフィールドに指定してください。

ワークスペースの履歴表示

このリリースの Sun WorkShop TeamWare には、ワークスペース履歴ファイルに入っている情報を簡単に表示できる機能が含まれています（「ワークスペース」▶「履歴の表示」を選択）。ワークスペース履歴ビューワを利用すれば、ワークスペースのトランザクション履歴、トランザクション詳細、コメント、およびコマンドログを表示できます。エントリのソートやフィルタリングができるほか、コメントやコマンドログの検索も行えます。

Sun WorkShop 6 Visual

表 9 では、Sun WorkShop 6 Visual で利用できる新機能を一覧表示します。一部の製品の機能については、表の次の節でさらに詳しく説明しています。

表 9 Sun WorkShop 6 Visual

機能	説明
Swing サポート	新リリースの Visual 6 は、Java 1.0、1.1 に加え Java Swing コードも生成できます。Visual 6 は Motif ウィジェットの Swing コンポーネントを生成するばかりではなく、マップ可能資源の種類も増えています。画像、シェルアイコン、シェルのサイズ変更・削除応答、リスト内容、行/列エントリ位置合わせといった拡張機能を含むトグルの Swing コードを生成できるようになっています。
拡張 Windows サポート	Windows MFC への X イベントのマッピングサポートがさらに強化されています。
Sun WorkShop プロジェクトとの統合	Sun WorkShop プロジェクトウィザードと併用することによって、グラフィカルユーザーインターフェースを持つプロジェクトを作成できます。

Swing サポート

新リリースの Visual 6 は、Java 1.0、1.1 に加え Java Swing コードも生成できます。Visual 6 は Motif ウィジェットの Swing コンポーネントを生成するばかりではなく、マップ可能リソースの種類も増えています。画像、シェルアイコン、シェルサイズ変更、削除応答、リスト内容、行/列エントリ位置合わせといった拡張機能が入ったトグルの Swing コードを生成できるようになっています。

標準クラスに等価性が欠落している Java に Motif コンポーネントをマップする MWT クラスライブラリを Swing に移植することによって、見た目と使い心地の整合性が改善されています。一部の Motif コンポーネントは該当する Swing コンポーネントに直接マップされるようになっているので、MWT にそれほど依存しなくとも Motif 互換のインターフェースを用意できるようになっています。

Java Layout エミュレーションウィジェットに適宜手が加えられた結果、Java Layout 特性における動作の整合性が改善されています。

他社製 (Motif 以外) コンポーネントのクロスプラットフォームコードをサポートする動きの一環として、新リリースの Visual 6 ではあらゆる統合コンポーネントのデフォルトの基本クラス指定行なうことができるようになっていきます。このクラス指定は、一般的言語ベースで行うことも、個別のバリエーションについて行うこともできます。たとえば、Java、Java 1.0、1.1 のデフォルトクラスや Swing 固有のクラスを指定できます。コンポーネント MFC クラスを指定することもできます。ターゲット言語において正しい種類のオブジェクトを作成するには他社コンポーネントを 1 つ 1 つ手作業で設定しなければならなかった従来の問題が解決されています。他社固有のリソースはマップされません。特定のコンポーネントを複数のネイティブオブジェクトにマップしなければならない状況や、組み込み要素を考慮すれば複合コンポーネントをマップできる状況については、対応していません。

Windows 拡張サポート

Visual 6 には、X イベントを Windows MFC にマップするための次のサポートが追加されています。

表 10 X イベントを Windows MFC にマップする

MouseMotion (マウスの動き)	ボタンの押し下げの有無にかかわらず、あらゆるマウス移動について汎用ハンドラを生成します
ButtonPress	Left、Center、Right の 3 種類の押し下げハンドラすべてを生成します
ButtonRelease	Left、Center、Right の 3 種類の解放ハンドラすべてを生成します
EnterWindow	MouseActivate
ExposureMask	EraseBkgnd
KeyPressMask	WM_KEYDOWN
KeyRelease	WM_KEYUP
KeymapstateMask	WM_SYSKEYUP/WM_SYSKEYDOWN
LeaveWindowMask	WM_KILL_FOCUS
ResizeRedirect	WM_SIZE
PropertyChangeMask	ON_WM_PAINT
VisibilityChangeMask	WM_SHOWWINDOW

Sun WorkShop 6 Performance Library

Sun Performance Library™ は、線形代数およびその他数に集中した問題を解決するための最適化された高速数値演算のサブルーチンです。Sun Performance Library は、<http://www.netlib.org> にある Netlib で利用できるパブリックドメインアプリケーションのコレクションに基づいています。これらのルーチンは機能強化され、バンドルされて Sun Performance Library となりました。

表 11 は、Sun WorkShop 6 Performance Library の新機能です。一部の製品の機能については、表の次の節でさらに詳しく説明しています。

表 11 Sun WorkShop 6 Performance Library

機能	説明
LAPACK 3.0 のサポート	LAPACK 3.0 サブルーチンが追加されました。Sun Performance Library の以前のバージョンは、LAPACK 2.0 に基づいていました。Sun Performance Library の現行バージョンでは、引き続き LAPACK 2.0 と LAPACK 1.X との互換性を保っています。
スパースソルバーパッケージ	スパースソルバーパッケージは、直接的な方法およびユーザー指定の順序付けを含むフィルリダクション順序付けのアルゴリズムを使用して、スパース行列 (対称的、構造上対称的、および非対称的な係数行列) を解決するルーチンを提供します。
UltraSPARC-III のサポート	UltraSPARC-III がサポートされるようになりました。UltraSPARC-III に固有のコードを使用するには、32 ビットコードの場合は <code>-xarch=v8plusb</code> 、64 ビットコードの場合は <code>-xarch=v9b</code> のオプションを付けてコンパイルします。
Fortran 95 のサポート	Fortran 95 言語の機能をサポートするようになりました。プログラムに F95 の文 <code>USE SUNPERF</code> を含めることによって、Performance Library のモジュールと定義を使用できます。
Sun Performance Library のライセンス供与に関する変更	Sun Performance Library は、ライセンスを必要としなくなりました。しかし、引き続き <code>-xlic_lib=sunperf</code> を使用して、アプリケーションを正しいサポートライブラリと確実にリンクし、正しいバージョンの Sun Performance Library を選択してください。

Fortran 95 言語機能サポート

F95 の文 `USE SUNPERG` をプログラムに含めることにより、Performance Library のモジュールと定義を利用することができます。これにより、以下の機能が得られます。

- 型への非依存性。FORTRAN 77 のルーチンでは、名前の一部として型を指定する必要がありました。Fortran 95では、特定のデータ型のルーチンは、そのルーチンに引き渡した引数のデータ型によって決めることができます。
- コンパイル時チェック。FORTRAN 77 では、一般的には、どのようなパラメータを特定のルーチンに引き渡すべきかをコンパイラが知ることはできません。Fortran 95 では、`USE SUNPERF` 文を使用することにより、コンパイラは各 Sun Performance Library ルーチンに引き渡される各パラメータの数、型、サイズ、および形状がどのようなものであるべきかを知ることができます。コンパイラは、呼び出しの内容をその期待値と照合し、コンパイル中にエラーを見つけ出します。
- 省略可能なパラメータ。FORTRAN 77 では、すべてのパラメータをすべてのルーチンについて順に指定する必要があります。Fortran 95 では、一部のパラメータが省略可能となっています。Sun Performance Library では、すべての増分パラメータ (`INCX`、`INCY` など)、ワークスペース、先頭次元 (`LDA`、`LDB` など)、長さ/サイズパラメータが省略可能です。

これらの機能の使い方や例については、『Sun Performance Library ユーザーズガイド』を参照してください。

Sun Performance Library のライセンス供与に関する変更。

Sun Performance Library は、もはやライセンスを必要としません。しかし、`-lsunperf` ではなく、引き続き `-xlic_lib=sunperf` を使用してリンクする必要があります。`-xlic_lib=sunperf` を使用して、次のことを確実に行ってください。

- アプリケーションを正しいサポートライブラリにリンクします。このリリースでは、Sun Performance Library は Fortran 77 ではなく Fortran 95 でコンパイルされています。`-xlic_lib` を使うと、Fortran 77 の実行時ライブラリではなく Fortran 95 の実行時ライブラリにリンクします。
- 正しいバージョンの Sun Performance Library を使用します。`-subparallel` を使って構築したプログラムと `-subparallel` を使わないで構築したプログラム、および異なる `-xarch` の値では、サポートする Sun Performance Library のパー

ジョンが異なります。-xlic_lib=sunperf を指定すると、ドライバは、使用中のコマンド行オプションに最適な Sun Performance Library のバージョンを使用します。

Sun WorkShop 6 Analyzer

Sun WorkShop 6 Analyzer は、Sun WorkShop 5.0 に装備された Analyzer を全面的に書き換えたものです。

表 12 は、Sun WorkShop 6 Analyzer の新機能です。一部の製品の機能については、表の次の節でさらに詳しく説明しています。

表 12 Sun WorkShop 6 Analyzer

機能	説明
関数分析	
主ディスプレイ (関数リスト)	新リリースにおける主ディスプレイは「関数リスト」であり、Analyzer 起動時にデフォルトで表示されます。
複数の測定値	新リリースの「関数リスト」では同時に複数の種類の測定値が表示されるので、一度に表示させたい測定値 1 つを選択する必要はありません。「関数リスト」は測定値を絶対値または百分率で表示します。
「概要メトリック」ウィンドウ	「表示」メニューから利用できる新しい「概要メトリック」ウィンドウでは、選択されている関数について記録されている測定値を、値と百分率の両方で表示します。概要メトリック」ウィンドウの内容は、関数リスト表示に表示される内容と無関係です。
「呼び出し元 - 呼び出し先」ウィンドウ	「関数リスト」から新機能である「呼び出し元 - 呼び出し先」ウィンドウにアクセスできます。このウィンドウには、関数の呼び出し側からその関数の被呼び出し側に測定値がどのように帰属されるのかが示されます。
注釈付きソースコードの生成	選択した関数について注釈付きソースコードを生成し、その結果を編集ウィンドウに表示できるようになりました。
注釈付き逆アセンブルの生成	選択した関数について注釈付き逆アセンブリを生成し、その結果を編集ウィンドウに表示できるようになりました。

表 12 Sun WorkShop 6 Analyzer (続き)

機能	説明
サンプル、スレッド、LWP によるデータのフィルタリング	新リリースでは、サンプル、スレッド、または LWP あるいはこれらを組み合わせたものによるデータのフィルタリングを「フィルターを選択」ダイアログボックスで行えます。この結果、選択されたサブセットから取り出されたデータだけを示すように、あらゆるディスプレイとウィンドウが更新されます。
スレッド同期待機測定値	新リリースでは、指定スレッシュホールドを超える同期イベント数とこれらのイベントからの総遅延との 2 種類のスレッド同期待機測定値を利用できます。
複数の実験のロード	新リリースでは、複数の実験を一度に Analyzer にロードできます。これらの実験の測定値が「関数リスト」ディスプレイに表示されます。

関数リスト

新リリースでは、「関数リスト」が主ディスプレイであり、Analyzer 起動時にデフォルトで表示されます。

関数リストでは同時に複数の種類の測定値が表示されるので、ディスプレイの型だけを変更するように「データ」リストボックスが変更されました。「ディスプレイリスト」オプションメニューは削除されました。

このリリースでは、「関数リスト」に記載されている測定値を秒/カウントによる絶対値またはプログラム全体の値に対する百分率、またはその両方の形で確認できます。「関数リスト」ディスプレイから「メトリックの選択」ダイアログボックスを呼び出して、以下が実行できます。

- 「関数リスト」に表示された測定値の選択
- 測定値をカウント、百分率のどちらで表示するか、あるいは両方で表示するか
- どの測定値で「関数リスト」をソートするか
- リストの並べ替え

「呼び出し元 - 呼び出し先」ウィンドウ

「関数リスト」から新機能である「呼び出し元 - 呼び出し先」ウィンドウにアクセスできます。このウィンドウには、関数の呼び出し元からその関数の呼び出し先に測定値がどのように帰属されるのかが示されます。「呼び出し元 - 呼び出し先」ウィンドウ

ウでは、選択されている関数がディスプレイ中央に表示され、この関数の呼び出し側は上のパネルに、この関数の呼び出し元は下のパネルに示されます。選択された関数について、この関数内部における使用率が、帰属測定値によって示されます。上の呼び出し元については、選択されている関数内とこの関数が呼び出す全関数内における使用率が、コールスタックを呼び出し元までたどって示されます。下の呼び出し元については、選択されている関数からの呼び出しに帰属する呼び出し元測定値の割合が示されます。

呼び出し元パネルと呼び出し先パネルのどちらかで関数をクリックすれば、「呼び出し元 - 呼び出し先」ウィンドウでプログラムの構造内をナビゲートできます。ディスプレイは、新たに選択された関数を基準としてセンタリングし直します。

注釈付きソースコードの生成

新リリースでは、選択した関数について注釈付きソースコードを生成し、その結果を編集ウィンドウに表示できるようになりました。ソースコードには、「関数リスト」と同じ測定値セットを使った、行単位の測定値の注釈が付きます。コンパイラの並列化のコメントと Fortran 95 の copyin/copyout のコメントも、ソースとインタリーブされた状態で入ります。ソースコードディスプレイでは、`-g` によるコンパイルを必要とし、最適化コードに対して有効となります (`-g` では最適化と並列化を使用不可にすることはできなくなりました)。

注釈付き逆アセンブリ

選択した関数について注釈付き逆アセンブリを生成し、その結果を編集ウィンドウに表示できるようになりました。逆アセンブルには、「関数リスト」と同じセットを使った、命令単位の測定値の注釈が付きます。また逆アセンブルには、コンパイラのコメントとインタリーブされたソースも入ります。

測定値

Sun WorkShop 6 Analyzerでは、以下のとおり、新しい測定値が加わり、既存の測定値が変更されました。

- 実行プロファイルデータは、新リリースでは時間ベースのプロファイリングと呼ばれます。実行プロファイルデータには常に呼び出された関数の回数が含まれます。
- ヒストグラムデータは、新リリースでは排他的測定値と呼ばれます。
- 累積データ測定値は、新リリースでは包含的測定値と呼ばれます。

- クロックベースのプロファイルデータは以下の測定値を生成します。
 - LWP 総時間
 - ユーザー CPU 時間
 - システム CPU 時間
 - システム待ち時間
 - テキストページフォルト時間
 - データページフォルト時間
 - ウォークロック時間
- 新しい種類のデータであるスレッド同期待機遅延追跡が装備され、以下の測定値を生成します。
 - 指定しきい値を超える同期イベント数
 - これらのイベントからの総遅延

これらの測定値はデフォルトでは集められません。

その他の変更事項

Sun WorkShop 6 Analyzer では、その他にも以下の変更を加えています。

- 「概要」ディスプレイでサンプルをクリックしてサブセットを選択する方法は、このリリースでは使用できなくなっています。「標本の選択」テキストボックスと矢印ボタンは新リリースの Analyzer には備わっていないので、「表示」メニューに選択コマンドが表示されることはありません。サンプル、スレッド、LWP のフィルタ選択は、すべて「フィルタの選択」ダイアログボックスで行なってください。
- Analyzer の中から実験記録ファイルを削除できなくなっています。「実験」▶「削除」コマンドはなくなり、代わりに Analyzer から実験を解除できる「実験」▶「解除」コマンドが用意されています。`er_rm` コマンドによって削除しなにかぎり、実験記録ファイルはディスク上に存在したままとなります。
- `er_export` ASCII データのフォーマットが完全に変更されました。`(er_export` はデバッグをする場合に限り使用されます)。

Sun WorkShop 6 インストール

表 13 は、Sun WorkShop 6 の新しいインストール機能です。

表 13 Sun WorkShop 6 インストール

機能	説明
Web Start	Web Start は、新しい GUI インストールソフトウェアです。
FLEXlm 7.0	FLEXlm 7.0 は、Sun WorkShop 6 とともに提供されるライセンスマネージャソフトウェアです。

Sun WorkShop 6 マニュアル、マニュアルページ、README、および HTML オンラインヘルプ

Sun Workshop および Sun Workshop TeamWare のマニュアル、マニュアルページと README は、テキスト形式のファイルとともに HTML 形式のファイルも用意されています。オンラインヘルプはこのバージョンから HTML 形式に変わっています。

この HTML ドキュメントには、

<インストールディレクトリ>

</SUNWspro/ws6/lib/locale/ja/html/index.html>

を出発点としてアクセスできます。デフォルトの <インストールディレクトリ> は </opt> です。

注 – この節の情報は、使用中の Sun WorkShop 6 の製品が </opt> ディレクトリにインストールされているとみなします。Sun WorkShop 6 ソフトウェアが </opt> にインストールされていない場合は、システム管理者にお問い合わせください。

HTML 形式で利用できる Sun WorkShop のマニュアルを表示するには、Netscape™ Communicator 4.0 または同互換の Netscape バージョンが必要です。Netscape™ Communicator は Solaris™ 7 オペレーティング環境 と Solaris™ 8 オペレーティング環境に含まれています。

Solaris 2.6 オペレーティング環境を使用していて、Netscape Communicator 4.0 または同互換バージョンがお手元がない場合は、(Solaris 2.5.1 オペレーティング環境用の) Netscape Communicator 4.7 をダウンロードできます。Solaris 2.5.1 オペレーティング環境用の Netscape Communicator 4.7 は、Solaris 2.6 オペレーティング環境と互換性があります。

Netscape Communicator 4.7 をダウンロードするには、以下の Netscape Communications Corporation の Web サイトを参照してください。

<http://www.netscape.com/download/index.html>

Sun WorkShop のオンラインヘルプ (HTML 形式) では、JavaScript が使用可能になっていなければなりません。JavaScript は Netscape のデフォルトの設定では、使用可能になっています。