



Sun WorkShop TeamWare ユーザーズガイド

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

Part No. 806-4846-01
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 : *Sun WorkShop TeamWare User's Guide*
Part No: 806-3573-10
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 xvii

1. Sun WorkShop TeamWare の概要 1
 - Sun WorkShop TeamWare の機能と特長 1
 - Sun WorkShop TeamWare の基本概念 2
 - TeamWare を使用する理由 2
 - 親ワークスペースと子ワークスペース 3
 - ソースコード管理システム (SCCS) 4
 - Sun WorkShop TeamWare のモデル 4
 - コピー / 変更 / マージモデル 5
 - チームプロジェクトモデル 7
 - Sun WorkShop TeamWare での作業 9
 - Sun WorkShop TeamWare の使用例 10
 - チームに新しいメンバーを追加する 10
 - Sun WorkShop TeamWare 環境を設定する 11
2. ワークスペースの管理 13
 - ワークスペース管理ツールの起動 13

親ワークスペースの作成	15
子ワークスペースの作成	17
ワークスペース間の変更内容の伝達	20
子ワークスペースの更新 (更新ブリングオーバー)	20
親ワークスペースへの変更内容のプットバック	22
ワークスペースに対する変更の取り消し	25
ワークスペースの名前の変更と移動	25
ワークスペースの削除と設定の解除	26
ワークスペースの削除	26
ワークスペース設定の解除	27
ワークスペースの履歴表示	27
ワークスペースの履歴表示のカスタマイズ	28

3. 高度なワークスペース管理 33

ブリングオーバー / プットバックのオプション	34
個々のブリングオーバー / プットバック時のオプション設定	34
ツール属性オプションの設定	37
独自のブリングオーバー / プットバック用ファイルリストの作成	38
デフォルトのファイルリストの保存	38
独自のファイルリストの作成	38
他のユーザーへのトランザクションの実行通知	40
ワークスペースに分かりやすい名前を付ける	41
ワークスペースの親子関係の変更	42
ワークスペースの親子関係を変更する理由	42
ワークスペースの親子関係を変更する方法	43
ワークスペースの親子関係の変更例	45
ツール属性によるワークスペース管理ツールのカスタマイズ	48
環境変数の設定	51

ワークスペースの自動読み込み	51
コマンド行コマンドに対するデフォルトのワークスペース設定	52
検索パスの設定	52
RCS プロジェクトからの変換	53
4. ワークスペースへのアクセス制御	55
アクセス権の設定	55
全体レベルでのアクセス制御	56
グループまたはユーザーレベルでのアクセス制御	58
プットバックの妥当性検査によるワークスペースの保護	59
プットバックの妥当性検査を有効にする	60
プットバックの妥当性検査プログラムの起動	61
ワークスペースのロックの解除	65
5. ファイルの管理	67
バージョン管理ツールの起動	67
ワークスペースへのファイルの追加	69
ファイルのチェックアウト	70
ファイルの編集	71
デフォルトのエディタの変更	71
ファイルのチェックイン	72
ファイルに対する変更の取り消し	72
プットバックによる変更の統合	73
6. ファイル間の相違の解決	75
ファイルマージツールの起動	75
「TeamWare」メニューからのファイルマージツールの起動	76
「衝突解決」タブからのファイルマージツールの起動	76

「ファイルマージ」ウィンドウの表示	77
ワークスペース内での衝突の解決	77
「ファイルマージ」ウィンドウのグリフについて	79
ファイルマージツールへのファイルの読み込み	81
相違の解決	83
自動マージの使用方法	84
自動マージ規則のまとめ	85
変更の取り消し	85
ファイルマージツールのオプション	85
ファイルのマージ例	88
相違を確認する	92
7. 高度なファイル管理	95
ワークスペース内のファイルの更新	95
ファイルの履歴の表示	96
「ファイル履歴」ウィンドウ	98
ファイルの履歴 (デルタ、分岐、バージョン)	99
ファイル名の変更とファイルの移動、削除	101
ファイル名の変更と移動	101
ファイルの削除	105
Sun WorkShop TeamWare ファイルの削除	106
カスタマイズメニューの作成	106
「読み込み」メニューへのパスの追加	107
バージョン管理ツールの属性の設定	108
SCCS ファイルの属性の設定	110
8. フリーズポイントツールの使用方法	113
フリーズポイントツールの概要	113

フリーズポイントツールの仕組み	114
作成	114
抽出	115
ソースワークスペース	115
抽出先ディレクトリ	115
フリーズポイントツールの起動	115
フリーズポイントファイルの作成	118
フリーズポイントファイルの更新	119
ファイルの抽出	120
フリーズポイントの自動作成	124
フリーズポイントファイルの形式	126
9. Sun WorkShop TeamWare におけるプログラムの構築	129
「構築」ウィンドウ	129
WorkShop ターゲットの構築	131
Sun WorkShop ターゲット	132
ユーザーメイクファイルターゲット	133
プログラムの構築	133
デフォルト値を使用した構築	135
独自の構築値を使用した構築	137
既存の WorkShop ターゲットの編集	138
構築出力の収集	138
構築出力の保存	139
WorkShop ターゲットの削除	139
構築のカスタマイズ	139
構築オプションの指定	140
メイクファイルマクロの使用法	142
環境変数の使用法	144

構築エラーの修正	146
エラーのソースの表示	147
エラーの修正	148
構築の終了	149

10. `dmake` ユーティリティの使用 151

<code>dmake</code> の基礎	151
構成ファイル	152
<code>dmake</code> ホスト	152
構築サーバー	155
<code>dmake</code> ユーティリティについて	156
<code>dmake</code> ユーティリティがメイクファイルに与える影響	156
メイクファイルテンプレートの使用	156
ターゲットの並列構築	157
メイクファイルに関する制限事項	157
ライブラリの同時更新	159
複数のターゲット	160
並列処理に対する制限	160
分散メイクの入れ子呼び出し	161
<code>dmake</code> ユーティリティの使用方法	162

11. Sun WorkShop TeamWare の ショートカット 165

コマンド行からの TeamWare の使用方法	165
ワークスペース管理ツールのコマンド	166
バージョン管理ツールのコマンド	167
ファイルマージツールのコマンド	167
フリーズポイントツールのコマンド	169

GUI のショートカット	169
ワークスペース管理ツールでのダブルクリック時の動作	172
バージョン管理ツールでのダブルクリック時の動作	172
12. Sun WorkShop TeamWare のアーキテクチャー	173
ワークスペースのメタデータディレクトリ	173
ワークスペース管理ツールのデフォルト値ファイル	176
access_control ファイル	177
ワークスペース管理ツールによるファイルのマージ	178
衝突していないファイルのマージ	179
衝突しているファイルのマージ	180
ファイルマージによるデルタの追跡方法	181
SCCS マージ可能 ID について	191
SMID が必要な理由	191
SMID と SID の変換	191
A. エラーメッセージと警告メッセージ	195
エラーメッセージ	195
警告メッセージ	214
用語集	221
索引	229

目次

図 1-1	一般的な TeamWare 環境の設定	8
図 2-1	親ワークスペースが読み込まれた状態の「ワークスペース管理」ウィンドウ	14
図 2-2	「トランザクション」ダイアログボックスの「作成ブリングオーバー」タブ	18
図 2-3	「ファイル追加」ダイアログボックス	19
図 2-4	「トランザクション」ダイアログボックスの「更新ブリングオーバー」タブ	21
図 2-5	「トランザクション」ダイアログボックスの「プットバック」タブ	23
図 2-6	ワークスペースの履歴表示	29
図 3-1	「トランザクション」ダイアログボックスのオプション区画	35
図 3-2	親子関係のない2つのワークスペース	45
図 3-3	patch1.0 ワークスペースのクローンを作成	46
図 3-4	クローンの親を Release2.0 に変更	46
図 3-5	ファイルをブリングオーバー、マージして、新しいリリースに適用	47
図 3-6	修正内容が取り込まれた Release2.0 (patch1.0_clone は削除)	47
図 3-7	「ツール属性」ダイアログの「ワークスペース管理」タブ	49
図 4-1	「ワークスペース属性」ダイアログボックスの「アクセス制御」タブ	57
図 4-2	「ワークスペース属性」ダイアログボックスの「プットバックの妥当性検査」タブ	61
図 5-1	「バージョン管理」ウィンドウ	68
図 6-1	ワークスペースの衝突例	76
図 6-2	「ファイルマージ」ウィンドウ	79
図 6-3	ファイルマージツールにおける「ファイルを開く」ダイアログボックス	81
図 6-4	「ツール属性」ダイアログボックスの「衝突解決」タブ	86

- 図 6-5 自動マージ後の `file_1` と `file_2` のマージ結果 90
- 図 6-6 自動マージ後の子区画の `file_1` の内容 91
- 図 6-7 自動マージ後の親区画の `file_2` の内容 91
- 図 7-1 「ファイル履歴」ウィンドウ 97
- 図 7-2 ファイル "C" の名前を "D" に変更 102
- 図 7-3 親と子ワークスペースにおけるファイル "C" の名前変更の衝突例 104
- 図 7-4 `rm` コマンドで子から "C" を削除し、プリングオーバーで再作成 105
- 図 7-5 バージョン管理ツールの「オプション」ダイアログボックス 109
- 図 8-1 フリーズポイントツールが保存するデルタ 114
- 図 8-2 「フリーズポイント」ウィンドウの「作成」タブ 116
- 図 8-3 フリーズポイント処理の進行状況 121
- 図 8-4 「フリーズポイント」ウィンドウの「抽出」タブ 122
- 図 8-5 「ワークスペースの属性」ダイアログボックスの「フリーズポイント」タブ 125
- 図 9-1 「構築」ウィンドウ 130
- 図 9-2 「ターゲットの新規定義」ダイアログ 134
- 図 9-3 構築出力表示区画に表示された構築エラー 147
- 図 9-4 エラーが存在するソースファイルをテキストエディタのウィンドウに表示 148
- 図 12-1 転送先ワークスペース内の変更されていないファイルの更新 180

表目次

表 1-1	TeamWare ツールの用途	9
表 2-1	「ワークスペース管理」ウィンドウのメニュー	15
表 2-2	ワークスペースの履歴表示の項目	28
表 3-1	プリングオーバー / プットバックのオプション	36
表 3-2	プリングオーバー / プットバックのツール属性	37
表 3-3	<code>workspace descr</code> コマンドのオプション	42
表 3-4	ワークスペース管理ツールの属性	50
表 4-1	プットバックの妥当性検査のモード	60
表 5-1	「バージョン管理」ウィンドウのメニュー	68
表 6-1	ファイルマージツールの「ファイルを開く」ダイアログボックスのフィールド	82
表 6-2	自動マージ規則のまとめ	85
表 6-3	相違解決ツール属性	86
表 6-4	「ファイルマージ」ウィンドウの表示オプション	87
表 7-1	「ファイル履歴」ウィンドウの各部	98
表 7-2	ファイル履歴表示で使用されるシンボル	98
表 7-3	バージョン管理ツールの「オプション」ダイアログボックス	109
表 7-4	SCCS ファイルの属性	111
表 8-1	フリーズポイントの「作成」タブの項目	117
表 8-2	フリーズポイントの「抽出」タブの項目	123
表 9-1	「構築」ウィンドウの項目	131
表 9-2	「ターゲットの新規定義」ダイアログボックスの項目	134

表 9-3	dmake のオプション	141
表 11-1	「ワークスペース管理」のメニュー項目と対応するコマンド	167
表 11-2	マウスおよびキーボードショートカット	170
表 12-1	Codemgr_wsdata メタデータディレクトリの内容	174
表 12-2	デフォルトのアクセス権設定	177
表 12-3	ワークスペースに対するアクセス制御値	178
表 A-1	エラーメッセージ	195
表 A-2	警告メッセージ	214

はじめに

このマニュアルでは、Sun WorkShop™ TeamWare コード管理ツールの使用方法について説明します。主としてソフトウェア開発者を対象としていますが、コード管理に関わりのある作業を行う、統合担当者、システム管理者、リリースエンジニアなども対象としています。

通常、コード開発者は、統合領域またはワークスペースからコードを持ってきて、以下の作業を行います。

- プログラムモジュールに新機能を追加する
- プログラムのテストおよびデバッグを行う
- コードを実装領域または統合ワークスペースに戻す

このマニュアルは、Solaris™ オペレーティング環境および UNIX® コマンドに関するある程度の知識を持っていることを前提としています。ソースコード管理システム (SCCS) の使用経験は問いません。

第 9 章「Sun WorkShop TeamWare におけるプログラムの構築」と第 10 章「dmake ユーティリティの使用」は、標準の make ユーティリティについて書かれたドキュメントの補足となるものです。構築ツールおよび分散メークを使用してプログラムの構築プロセスをより効率的に処理する方法について説明しています。make ユーティリティを使用してプログラム管理する上で、構築処理をより短時間に行いたい場合にお読みください。これらの章は、読者が標準の make ユーティリティに関して十分な知識を持っていることを前提としています。

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 - Intel アーキテクチャとは、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 Intel アーキテクチャと呼んでいます。

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、`csh(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。

内容の紹介

このマニュアルは次の章と付録から構成されています。

第 1 章「Sun WorkShop TeamWare の概要」では、Sun WorkShop TeamWare 製品の概要を説明します。

第 2 章「ワークスペースの管理」では、ワークスペースを作成、管理する手順を具体的に説明します。

第 3 章「高度なワークスペース管理」では、ワークスペース管理ツールをカスタマイズし、ワークスペースに対してさらに高度な作業を行う手順を具体的に説明します。

第 4 章「ワークスペースへのアクセス制御」では、ワークスペースに対するトランザクションの実行権を制御する手順を具体的に説明します。

第 5 章「ファイルの管理」では、ファイルのチェックアウトとブットバックの手順を具体的に説明します。

第 6 章「ファイル間の相違の解決」では、ファイルマージツールでファイル間の相違を解決する手順を具体的に説明します。

第 7 章「高度なファイル管理」では、ファイル履歴の表示やファイルの移動、ファイル名の変更、バージョン管理ツールのカスタマイズ手順を具体的に説明します。

第 8 章「フリーズポイントツールの使用方法」では、フリーズポイントを作成、利用する手順を具体的に説明します。

第 9 章「Sun WorkShop TeamWare におけるプログラムの構築」では、特定のターゲットを構築する手順を具体的に説明するとともに、構築エラーの修正に関するヒントを示します。

第 10 章「dmake ユーティリティの使用」では、複数のホストに構築ジョブを分散して、並列構築を可能にするツール `dmake` について説明します。`dmake` の操作と、構築ジョブを効率的に分散する方法について説明しています。

第 11 章「Sun WorkShop TeamWare のショートカット」では、コマンド行から Sun WorkShop TeamWare コマンドを使用する方法を説明します。また、マウスとキーボードを利用したショートカットも一覧で示しています。

第 12 章「Sun WorkShop TeamWare のアーキテクチャー」では、メタデータファイルと、ファイル転送トランザクション中の、ワークスペース管理ツールによる SCCS 履歴ファイルの処理について説明します。また、SCCS マージ可能 ID (SMID) と SCCS デルタID (SID) についても説明しています。

付録 A「エラーメッセージと警告メッセージ」では、エラーメッセージと警告メッセージを一覧表示し、各メッセージの意味と対策を示しています。

「用語集」は、このマニュアルで使用されている特殊用語を説明した用語集です。

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
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、コーディング例。	<pre>.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。 machine_name% You have mail.</pre>
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表わします。	<pre>machine_name% su Password:</pre>
AaBbCc123 または ゴシック	コマンド行の変数部分。実際の名前または実際の値と置き換えてください。	<pre>rm <i>filename</i> と入力します。 rm ファイル名 と入力します。</pre>
『 』	参照する書名を示します。	『SPARCstorage Array ユーザーマニュアル』
「 」	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

以下の方法で、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。
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 (このマニュアル)	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。

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 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、lint 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、Iostream 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。
	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 の 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。

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
	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。
	FORTTRAN 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。
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。

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
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。

第1章

Sun WorkShop TeamWare の概要

この章では、Sun WorkShop™ TeamWare の概要を説明します。この章は、次の節で構成されています。

- Sun WorkShop TeamWare の機能と特長
- Sun WorkShop TeamWare の基本概念
 - TeamWare を使用する理由
 - 親ワークスペースと子ワークスペース
 - ソースコード管理システム (SCCS)
- Sun WorkShop TeamWare のモデル
 - コピー / 変更 / マージモデル
 - チームプロジェクトモデル
- Sun WorkShop TeamWare での作業
- Sun WorkShop TeamWare の使用例
 - チームに新しいメンバーを追加する
 - Sun WorkShop TeamWare 環境を設定する

以降の章では、各 TeamWare ツールと機能の使用方法について詳細を説明します。

Sun WorkShop TeamWare の機能と特長

Sun WorkShop TeamWare は、ソフトウェアを並行して開発するチーム向けのソースコード管理製品です。ソースコード管理を簡略化することによって時間を節約し、組織の生産性を高めます。Sun WorkShop TeamWare では、次のことを行えます。

- プロジェクトの各段階およびチームのメンバー構成に合わせたワークスペースの管理。作業用ディレクトリおよびサブディレクトリ (フォルダ) がワークスペースになります。
- 同じファイルの異なるバージョンに対する作業。メンバー各自のワークスペースに最新のバージョンを保持できます。
- 1つのワークスペース内で同一ファイルの編集を禁止することによる衝突の防止。
- 複数のワークスペースで編集された、同じファイルの異なるバージョンの検査と選択マージ。
- ワークスペースの特定のバージョンの検出とフリーズポイント作成。後で使用するためのそのバージョンのアーカイブ保管。
- 選択したファイルおよびディレクトリからのリリース用アプリケーションの構築。ローカル (逐次または並列) に構築することも、複数のシステムを使用して分散メークすることもできます。
- チームのメンバーに対する、ファイルに加えられた個々の処理の自動通知

Sun WorkShop TeamWare のバージョン管理機能は、ソースコード管理システム (SCCS) というプログラムに基づいています。Sun WorkShop TeamWare ツールで設定したワークスペースは、SCCS の管理下にあるファイルに対してだけ機能します。RCS ソースコード管理システム下にあるファイルは、Sun WorkShop TeamWare に移行することによって管理できます。

Sun WorkShop TeamWare の基本概念

Sun WorkShop TeamWare を使用するには、TeamWare の基盤となっている開発形態である、「コピー / 変更 / マージ」を理解する必要があります。また、「親」と「子」ワークスペースの概念に基づく、ワークスペース間の関係を理解する必要もあります。TeamWare は、UNIX の SCCS バージョン管理下にあるファイルに対してだけ使用できます。以下では、これらの概念を簡単に説明します。

TeamWare を使用する理由

多くの大規模なソフトウェア開発プロジェクトで最も難しいのは、共通および相互依存のファイルを共有する開発者間の作業の調整です。

開発者がそれぞれに自分のソースコードのコピーを保持している場合、最終的にコード全体をマージするときに、元のソースに加えられた変更を完全に追跡するのは困難です。この1つの解決策は、一度に一人の開発者だけが共通ファイルに逐次アクセスできるようにすることです。しかし、この方法は現実的ではありません。

Sun WorkShop TeamWare では、開発者ごとに独立した専用のワークスペースを作成できるため、調整しながら並行開発を進めることができます。各開発者は、中央のワークスペースからプロジェクトファイルを自分のワークスペースにコピーし、そのワークスペース内でファイルを変更して、変更を中央ワークスペースに戻すことができます。

親ワークスペースと子ワークスペース

チームは、ディレクトリ (またはフォルダ) とファイル単位で作業を行います。Sun WorkShop TeamWare を活用するには、1つの最上位ディレクトリ内にすべての作業用ディレクトリをまとめ、そのディレクトリ階層を1つのワークスペースとします。Sun WorkShop TeamWare には、ワークスペースを管理するための次のツールが用意されています。

- ワークスペース管理ツール：複数の TeamWare ユーザーがそれぞれに所有するワークスペース間でのつながりを形成します。また、ワークスペースの履歴と、実行されたあらゆるトランザクションを管理します。
- バージョン管理ツール：ファイルの履歴と各ファイルに対するデルタを管理します。
- ファイルマージツール：ファイルの変更部分が互いに上書きされることのないように保護します。
- フリーズポイントツール：ワークスペースの「スナップショット」を取得します。
- 構築ツール：ファイルを結合してアプリケーションを作成します。

Sun WorkShop TeamWare は、上位ディレクトリを1つのワークスペースに変換します。ワークスペースから新しいワークスペースを作成すると、元のワークスペースとそのコピーの間に特別な関係が作成されます。元のワークスペースは、新たに作成された「子」ワークスペースの「親」とみなされます。

親ワークスペース内のマスターファイルを壊すことのないよう、チームのメンバーは、自分の子ワークスペース内のファイルのコピーを使用します。Sun WorkShop TeamWare では、子ワークスペースと親ワークスペースとの間でディレクトリを簡単にコピーし合うことができます。

親ワークスペースは、多数の子ワークスペースを持つことができます。チームのメンバーは、自分の子ワークスペースに必要なディレクトリとファイルを取り込みます。子ワークスペースには、親ワークスペース内のすべてのディレクトリとファイルのコピーを取り込むことも、親ワークスペースの一部内容だけ取り込むこともできます。

多数のレベルで構成される複雑なプロジェクトでは、1つのワークスペースがいくつかのワークスペースの親であるとともに、別のワークスペースの子であることがあります。

ソースコード管理システム (SCCS)

Sun WorkShop TeamWare は、ソースコード管理システム (SCCS) の管理下にあるファイルだけ認識します。ファイルがチェックアウトされ、変更されて、チェックインされるたびに、SCCS はその変更を記録します。同じファイルの 2 つのバージョン間の相違点を、「デルタ」と呼びます。Sun WorkShop TeamWare は、SCCS デルタに基づいてファイルを管理します。ファイルが編集、移動、コピーされると、ワークスペース管理ツールはそのファイルの SCCS 履歴ファイルをコピーまたはマージします。Sun WorkShop TeamWare による SCCS 履歴ファイルの処理とマージについては、178 ページの「ワークスペース管理ツールによるファイルの マージ」を参照してください。

SCCS については、『プログラミングユーティリティ』を参照してください。

Sun WorkShop TeamWare のモデル

組織内での Sun WorkShop TeamWare のモデルは、次の 2 つがあります。

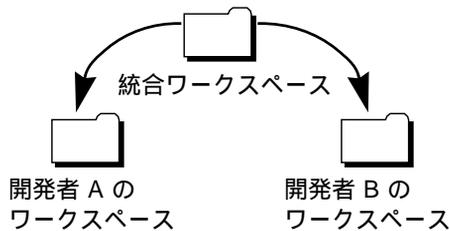
- コピー / 変更 / マージモデル : Sun WorkShop TeamWare の基本概念を表す利用形態です。
- チームプロジェクトモデル : 設定時や日々の作業における、Sun WorkShop TeamWare ユーザーの利用形態です。

コピー / 変更 / マージモデル

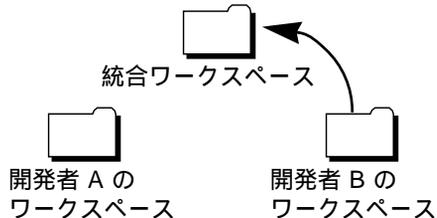
Sun WorkShop TeamWare のワークスペース管理ツールは、「コピー / 変更 / マージ」という並行開発の形態に基づいています。TeamWare を使用した日々の作業は、プロジェクトファイルに対する次のような基本作業の繰り返しになります。

1. プリングオーバー : 親ワークスペースから自分のワークスペースにディレクトリの最新バージョンをコピーします。
2. 変更 : 自分のワークスペース内のディレクトリにあるファイルを変更します。
3. プットバック : 変更したファイルを親ワークスペースに戻します。その間、他のメンバーが同じファイルのコピーで作業をしていて、親ワークスペースにコピーを戻した場合は、その 2 組の変更を選択してマージできます。

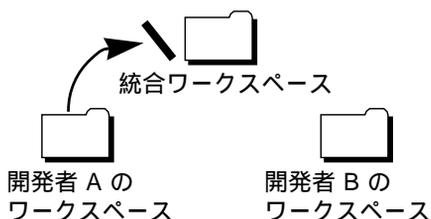
次のコピー / 変更 / マージの例は、2 人のユーザーがある 1 つのプロジェクトの同一または関連する部分を同時に使用するという、一般的なソフトウェア開発例を示しています。



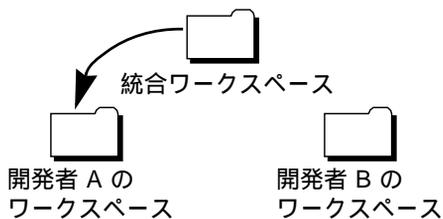
- 開発者 A と B が、プロジェクト統合ワークスペースから各自のワークスペースに同じファイルをコピーします。



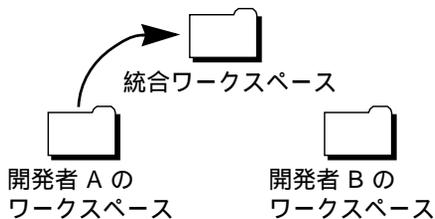
- 開発者 B がファイルを変更し、変更したファイルを統合ワークスペースに戻します。



- 開発者 A も自分のワークスペース内で同じファイルを変更し、統合ワークスペースに戻そうと試みます。しかし、開発者 B の変更が上書きされることになるため、TeamWare のワークスペース管理ツールによって、その試みが阻止されます。



- ワークスペース管理ツールによって、開発者 A に変更の衝突が通知されます。開発者 A は、統合ワークスペースから自分のワークスペースに開発者 B の変更を含むファイルをコピーします。



- ワークスペース管理ツールを使用して衝突を解決し、変更をマージおよびテストして、適切な変更内容のファイルを統合ワークスペースに戻します。

チームプロジェクトモデル

個別ユーザーに対するコピー / 変更 / マージの開発形態は、より大きなチーム向けの開発形態にも適合します。一般的な大規模プロジェクトでは、たいてい、そのプロジェクトの形態、スケジュールに合わせて複雑なプロジェクト構成とプロセスを組み立てる必要があります。以下は、TeamWare 環境の設定時間と日々の作業を進める上での、そうした構成とプロセスの例です。

Sun WorkShop TeamWare 環境の設定

プロジェクトを開始するときは、チーム開発を始める前に次のことを行います。

1. チームのメンバーの 1 人 (チームリーダーやシステム管理者) が、次のようなプロジェクトの進捗段階や各部門を収容する一群の親ワークスペースを作成します。
 - アルファ : チームの最新のファイルが置かれるワークスペース
 - 統合 : チームリーダーがチーム全員の作業内容をまとめるワークスペース
 - チーム : 各エンジニアリングチーム独自のワークスペース
 - ベータ / 最終 : プロジェクトの進捗段階、リリース、プラットフォーム、ロケールなどの基準に基づく、バージョン別のワークスペース
2. ファイルに対する作業を開始する前に、各チームメンバーが、チームのワークスペース (部門 1、部門 2、部門 3) に関係する各自の子ワークスペースを作成します。

3. 各チームメンバーが、必要なディレクトリやファイルのコピーを親ワークスペースから各自のワークスペースにプリングオーバーします。

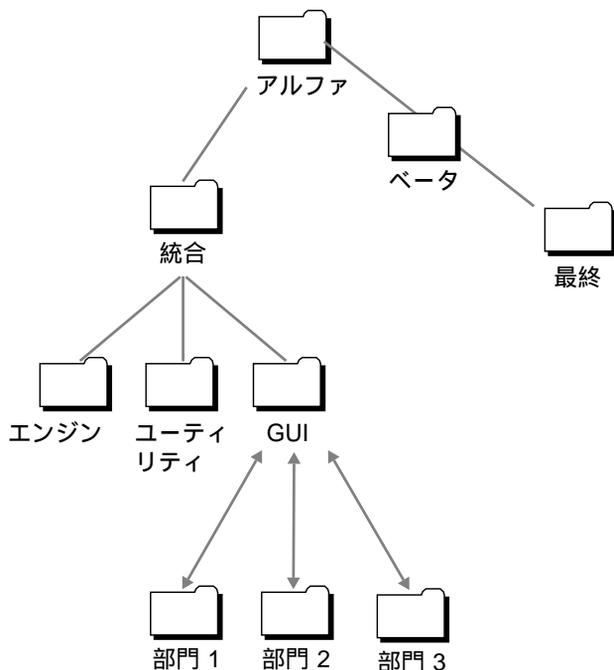


図 1-1 一般的な TeamWare 環境の設定

これで、チームの並行開発のための環境が準備できたことになります。この環境で作業を行うには、以下で説明するようにワークスペースを構成します。

Sun WorkShop TeamWare 環境における日々の作業

開発チームがファイルやディレクトリに対して作業を行い、アプリケーションの開発行程を進めていくにしたがい、次の作業を定期的に繰り返し行うことになります。

1. チームメンバーが、チームの親ワークスペースから各自の子ワークスペースにディレクトリをプリングオーバーします。
 - メンバーは、プリングオーバーを頻繁に行う必要があります。このことを習慣付けることによって、チーム内の新しい変更がメンバーのワークスペース全体に反映されます。メンバー全員が最新の状態を維持するために必要です。
 - メンバーが各自の子ワークスペースで変更を行い、変更したディレクトリをチームの親ワークスペースにプットバックします。

- TeamWare は、変更がブットバックされたことをチームの全員 (または通知リストに登録されているメンバー) に自動的に通知します。
2. チームのリーダーは、ブットバックされた変更を調べ、変更されたディレクトリをチームのワークスペースから「統合」ワークスペースにブットバックします。
 3. 「統合」ワークスペース内のファイルの準備ができたことを通知された構築担当者が、すべてのファイルを統合してアプリケーションを構築し、変更が反映されたディレクトリを「アルファ」ワークスペースにブットバックします。
 4. 「アルファ」ワークスペースの準備ができたことを通知されたソフトウェア検査担当者は、そのワークスペースに移動してアプリケーションをテストします。
 5. 製品のアルファ段階が完了したら、「アルファ」ワークスペースの子として「ベータ」ワークスペースを作成し、「統合」ワークスペースの親を「ベータ」ワークスペースに切り替えます。
 6. 安全のために、チームのメンバーは、ワークスペースの特定の「スナップショット」(フリーズポイント) を定期的に残しておくか、アーカイブします。作成したフリーズポイントからワークスペースを抽出できます。このワークスペースには、フリーズポイントを取ったときのその時点の全内容が含まれています。

Sun WorkShop TeamWare での作業

次の表に、各 TeamWare ツールの用途を簡単にまとめます。

表 1-1 TeamWare ツールの用途

作業	TeamWare ツール
ディレクトリのワークスペースへの変換	ワークスペース管理ツール
親ワークスペースへからの子ワークスペースの作成	ワークスペース管理ツール
親ワークスペースから子ワークスペースへのディレクトリのブリングオーバー	ワークスペース管理ツール
自分のワークスペースのファイルのチェックアウト	バージョン管理ツール
ファイルの変更と保存	バージョン管理ツール
自分のワークスペースのファイルのチェックイン	バージョン管理ツール
ワークスペースにおけるファイルの新規作成	バージョン管理ツール

表 1-1 TeamWare ツールの用途 (続き)

作業	TeamWare ツール
新規および変更したファイルの親ワークスペースへのプットバック	ワークスペース管理ツール
複数メンバーがそれぞれのワークスペースで変更したファイルの矛盾バージョンのマージ	ファイルマージツール
マージしたファイルの親ワークスペースへのプットバック	ワークスペース管理ツール
特定のバージョンのワークスペースの保存	フリーズポイントツール
複数ファイルからのアプリケーションの構築 (検査やリリース用)	構築ツール

Sun WorkShop TeamWare の使用例

Sun WorkShop TeamWare は、プロジェクトやチームにおける各自の役割に応じて、さまざまな使い方ができます。ここでは、2 つの例を取り上げます。

チームに新しいメンバーを追加する

TeamWare を使用してソフトウェアの並行開発を行なっていて、すでに親ワークスペースと子ワークスペースが作成されているチームに新しいメンバーを追加する場合は、そのメンバーは次のことを行う必要があります。

1. チームの親ワークスペースの構成を確認します。
2. 構築に関するチームの方針やプロセス、スケジュールを確認します。
3. 自分の子ワークスペースを作成します。この子ワークスペース内で作業をするようになります。
4. Sun WorkShop TeamWare 内で電子メール通知の設定をします。
5. 親ワークスペースから必要なディレクトリをブリングオーバーします。
6. 各ファイルをチェックアウトし、変更、保存をして、チェックインします。

7. 必要に応じて、自分のワークスペースのファイルの追加または削除を行います。追加したファイルは、SCCS の管理下に置きます。
8. ファイルの変更を終えたら、ディレクトリ全体を親ワークスペースにプットバックします。
9. 行なった変更は、チームに自動的に通知されます。
10. プットバックしようとしたファイルが、すでに別のメンバーによって変更されプットバックされていた場合は、変更内容をマージし、そのマージ結果を反映したファイルを親ワークスペースにプットバックします。

Sun WorkShop TeamWare 環境を設定する

Sun WorkShop TeamWare ワークスペース環境を初めて作成する場合は、次のことを行う必要があります。

1. 次のような基準に基づいて、チームのワークスペースの構成を決定します。
 - プロジェクトの各段階
 - チームのメンバーの所在地、またはメンバーが使用するネットワークやファイルシステム
 - アプリケーション開発に使用される、およびアプリケーション開発のターゲットとなるプラットフォーム
 - プロジェクトのリリース構造
2. プロジェクト用の親ワークスペースを作成します。
3. チームのメンバーが子ワークスペースを作成できるように、作成した親ワークスペースの情報をチームに通知します。
4. 次のような事項に関してチーム内で取り決めます。
 - 定期的なブリングオーバー
 - プットバック
 - 定期的な更新期限 (例: 毎週の構築前まで)
 - 下位の親ワークスペースから上位の親ワークスペースへの定期的なプットバック
 - 定期的な構築
 - 構築またはその他の主要バージョンのフリーズポイント作成

5. チームと共にアプリケーションの開発も行なっている場合は、10 ページの「チームに新しいメンバーを追加する」で説明しているように、自分の子ワークスペースを作成し、定期的に Sun WorkShop TeamWare ツールを使用して、必要な作業を行います。

第2章

ワークスペースの管理

Sun WorkShop TeamWare の機能を利用するには、ファイルとディレクトリを TeamWare ワークスペースに置く必要があります。ワークスペースとは、ワークスペースとして指定されたディレクトリとそのサブディレクトリ、およびそれらのディレクトリに含まれるファイル全体を指します。Sun WorkShop TeamWare のツールを利用することによって、ワークスペース内のファイルとその他のワークスペースとの関係を管理します。ワークスペース管理ツールは、ワークスペースの内容やワークスペース間の関係の表示、コマンドの実行を行うツールです。この章では、ワークスペース管理ツールで次の基本的な作業を行う方法を説明します。

- ワークスペース管理ツールの起動
- 親ワークスペースの作成
- 子ワークスペースの作成
- ワークスペース間の変更内容の伝達
- ワークスペースに対する変更の取り消し
- ワークスペースの名前の変更と移動
- ワークスペースの削除と設定の解除
- ワークスペースの履歴表示

ワークスペース管理ツールの起動

ワークスペース管理ツールを起動するには、コマンド行から次のコマンドを入力します。

```
% twconfig &
```

Sun WorkShop™ を使用している場合は、次のいずれかの方法で TeamWare のワークスペース管理ツールを起動できます。

- Sun WorkShop のメインウィンドウにある「TeamWare」ボタンをクリックする。
- Sun WorkShop のメインウィンドウのツールバーにある「ツール」メニューから「TeamWare」を選択する。

注 - Sun WorkShop 製品のコンポーネントとマニュアルページは、標準のディレクトリの `/usr/bin/` と `/usr/share/man` にはインストールされません。このため、Sun WorkShop TeamWare のツールにアクセスするには、`PATH` および `MANPATH` 環境変数の設定を変更する必要があります。xviii ページの「Sun WorkShop 開発ツールへのアクセス方法」を参照してください。

Sun WorkShop TeamWare のワークスペース管理ツールを起動すると、「ワークスペース管理」ウィンドウが開きます (図 2-1 を参照)。

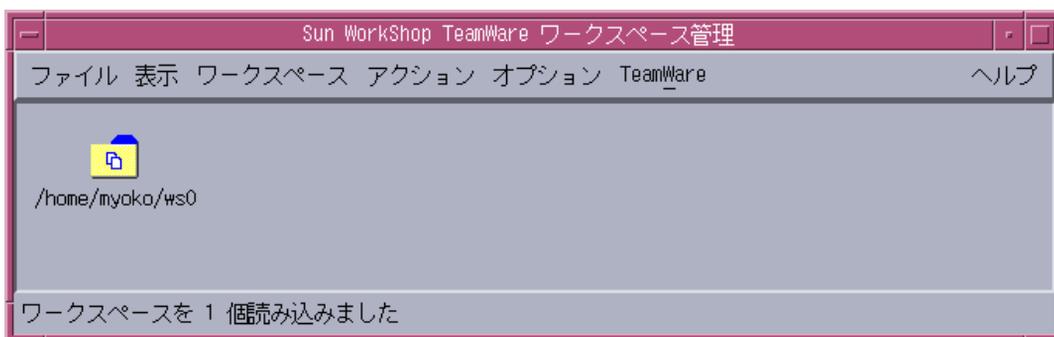


図 2-1 親ワークスペースが読み込まれた状態の「ワークスペース管理」ウィンドウ

「ワークスペース管理」ウィンドウにあるメニューを、表 2-1 に示します。

表 2-1 「ワークスペース管理」ウィンドウのメニュー

メニュー	説明
ファイル	ワークスペースの読み込み、読み込み解除、作成するためのコマンドがあります。
表示	ワークスペース区画内のワークスペースの表示方法を変更するためのコマンドがあります。
ワークスペース	ワークスペースを管理するためのコマンドがあります。
アクション	「トランザクション」ダイアログボックスを開きます。このダイアログボックスには、ファイルの同期を取るためのコマンド (ブリングオーバー、更新、プットバック) があります。
オプション	ワークスペースとワークスペース管理ツールのオプションを設定するためのコマンドがあります。
TeamWare	バージョン管理ツール、ファイルマージツール、フリーズポイントツールなどの他の TeamWare ツールを起動するためのコマンドがあります。
ヘルプ	ヘルプの起動、マニュアル一覧の表示、Sun WorkShop TeamWare 開発チームへのコメントの送信、バージョン番号の表示を行うためのコマンドがあります。

これらのメニューについての詳細は、Sun WorkShop TeamWare のオンラインヘルプを参照してください。ワークスペース管理ツールのカスタマイズ方法については、48 ページの「ツール属性によるワークスペース管理ツールのカスタマイズ」を参照してください。

親ワークスペースの作成

Sun WorkShop TeamWare では、新たにプロジェクトを開始することも、既存のプロジェクトファイルを Sun WorkShop TeamWare に読み込むこともできます。このためには、ワークスペースを作成する必要があります。

プロジェクトを開始するときは、まず、最上位のワークスペース (親ワークスペース) を作成します。その後で、プリングオーバーというファイルの特殊なコピーを行うことによって、親ワークスペースから子ワークスペースを作成します。開発チームのメンバーは、それぞれの子ワークスペース内で変更を行い、他の開発者が加えた変更と統合します。変更を統合する作業を、「プットバック」と呼びます。

ワークスペースを作成すると、TeamWare によって、`Codemgr_wsdata` という名前のサブディレクトリが自動的に作成されます。このサブディレクトリには、作成されたワークスペース内のファイルに関する情報が格納されます。親ワークスペースを作成するには、次の 2 つの方法があります。

- 空のワークスペースを作成して、その中に新しいディレクトリとファイルの階層を作成する。
- 既存のファイル階層からワークスペースを作成する。

注 - 既存のプロジェクトファイルを利用する場合は、ワークスペース管理ツールを使用する前に、それらファイルの格納場所 (パス名) を調べておいてください。

空のワークスペースの作成

空のワークスペースを新たに作成する手順は次のとおりです。

1. ワークスペース管理ツールを起動します。
2. 「ファイル」▶「ワークスペース作成」を選択します。
3. 「ワークスペースディレクトリ」フィールドにワークスペース名を入力します。
入力したディレクトリ名が存在しない場合は、Sun WorkShop TeamWare によって自動的に作成されます。
4. 「了解」をクリックします。

指定したディレクトリにワークスペースが作成され、「ワークスペース管理」ウィンドウにそのアイコンが作成されます。

ファイルを作成するには、ワークスペースにファイルをチェックインする必要があります。69 ページの「ワークスペースへのファイルの追加」を参照してください。

既存のファイルからのワークスペースの作成

ワークスペースに置きたいファイルを含むディレクトリがすでにある場合は、次のようにします。

1. TeamWare のワークスペース管理ツールを起動します。
2. 「ファイル」▶「ワークスペース作成」を選択します。
3. 「ワークスペースディレクトリ」フィールドに、ファイルが含まれているディレクトリのパス名を入力します。
4. 「了解」をクリックします。

指定したディレクトリにワークスペースが作成され、「ワークスペース管理」ウィンドウにそのアイコンが作成されます。

5. Sun WorkShop TeamWare のバージョン管理ツールを使用して、ファイルをチェックインします。

Sun WorkShop TeamWare は、SCCS でバージョン管理されているファイルだけを認識します。ファイルが SCCS でバージョン管理されていない場合は、ファイルをチェックインしてください。詳細は、69 ページの「ワークスペースへのファイルの追加」を参照してください。

子ワークスペースの作成

親ワークスペースを作成したら、開発チームの各メンバーが、親ワークスペースのファイルのコピーを持つ各自の子ワークスペースを作成する必要があります。ワークスペース管理ツールの各トランザクション (親ワークスペースからファイルをプリンクオーバーし、子ワークスペースでファイルを変更し、親ワークスペースにファイルをブットバックする作業) を、この親子関係に基づいて実行します。

子ワークスペースを作成する手順は次のとおりです。

1. ワークスペース管理ツールを起動します。
2. ファイルの取得先のワークスペースが自動的に読み込まれない場合は、「ファイル」▶「ワークスペースの読み込み」を選択します。

3. 「ワークスペース読み込み」ダイアログボックスからワークスペースを選択し、「ワークスペースの読み込み」ボタンをクリックします。
選択した親ワークスペースが読み込まれ、「ワークスペース管理」ウィンドウにそのアイコンが表示されます。
4. 「アクション」▶「作成プリングオーバー」を作成します。
「トランザクション」ダイアログボックスの「作成プリングオーバー」タブが開きま
す (図 2-2 を参照)。

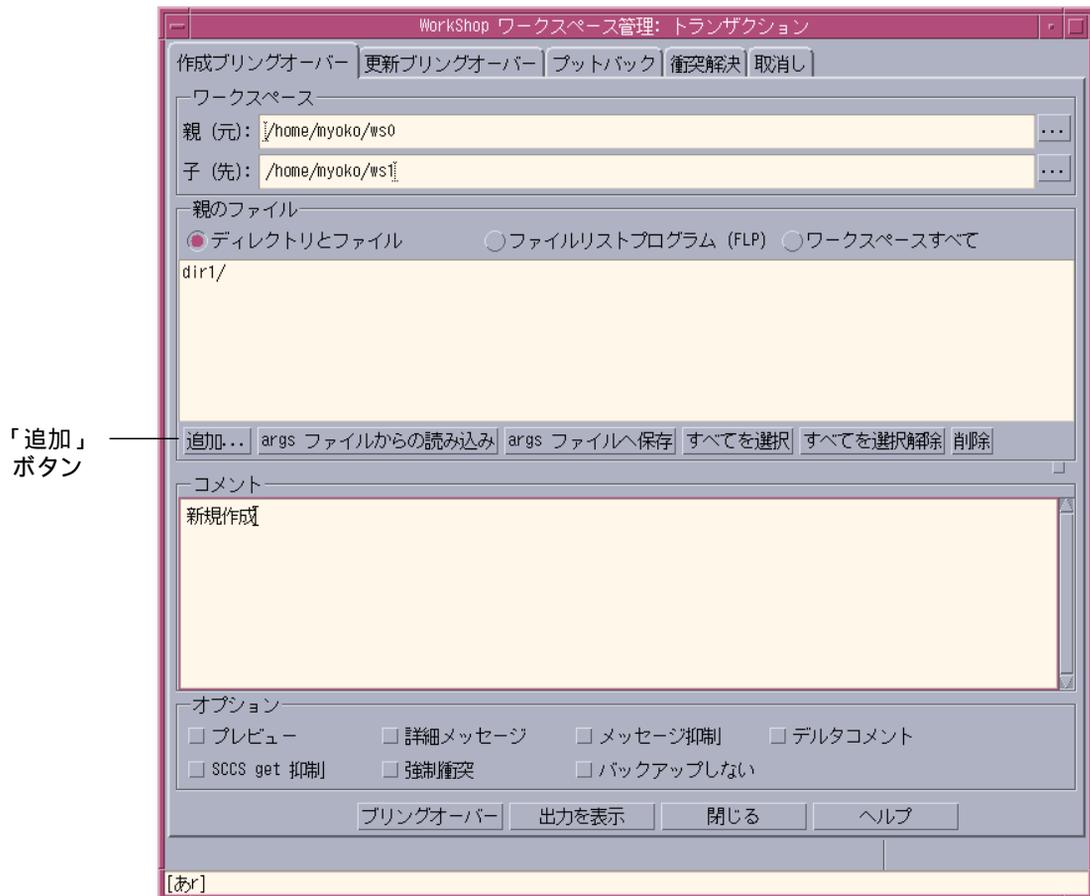


図 2-2 「トランザクション」ダイアログボックスの「作成プリングオーバー」タブ

5. 次のいずれかの操作を行い、プリングオーバーするディレクトリとファイルを選択します。

- デフォルトの「./」を選択し、すべてのファイルをブリングオーバーする。
 - 「ワークスペースすべて」ボタンをクリックし、すべてのファイルをブリングオーバーする。
 - 「追加」ボタンをクリックして「ファイル追加」ダイアログボックスを表示し、特定のファイルを追加または削除する。
- a. 「ファイル追加」ダイアログボックスを使用し、目的のファイルを選択します (図 2-3 を参照)。

このダイアログボックスでは、ディレクトリのアイコンをダブルクリックすることによってファイルシステムの階層を移動できます。特定のディレクトリに直接移動するには、「名前」フィールドにパス名を入力し、「ディレクトリの読み込み」ボタンを選択します。親ワークスペース階層の外へは移動できません。

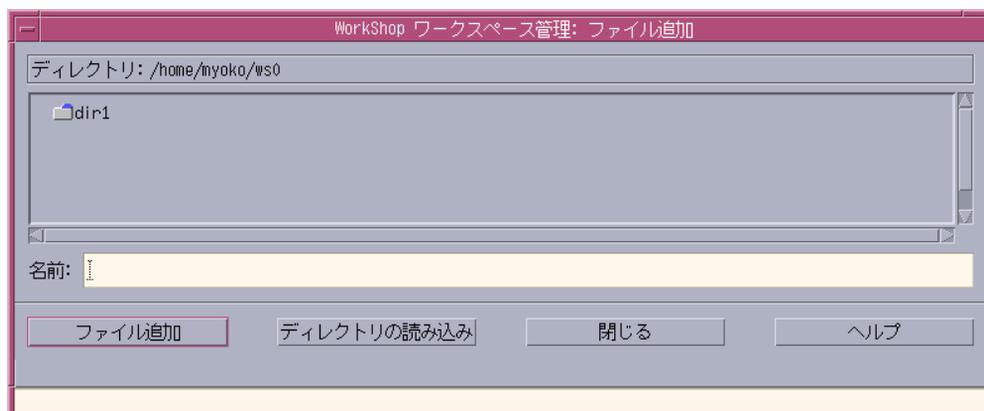


図 2-3 「ファイル追加」ダイアログボックス

- b. ファイルとディレクトリを選択します。
- ファイルまたはディレクトリのアイコンをクリックしてください。Shift キーを押しながらクリックすることによって、複数のファイルとディレクトリをまとめて選択できます。
- c. 「ファイル追加」ボタンをクリックして「作成ブリングオーバー」タブにファイルを追加します。
- d. 「閉じる」をクリックします。

6. 「作成ブリングオーバー」タブの「ブリングオーバー」ボタンをクリックします。
「トランザクション出力」ウィンドウにブリングオーバーの状態が表示されます。ブリングオーバーが完了すると、そのことが示されます。

これで、子ワークスペースが作成されます。以降、ファイルをチェックアウトして変更し、親ワークスペースにプットバックできます。ファイルをチェックアウトする方法については、70 ページの「ファイルのチェックアウト」を参照してください。親ワークスペースにファイルをプットバックする方法については、20 ページの「ワークスペース間の変更内容の伝達」を参照してください。

ワークスペース間の変更内容の伝達

ワークスペース階層を作成した後で大切なことは、それらワークスペースの内容の同期を取ることです。ワークスペース管理ツールのあらゆるファイル転送トランザクションは、子ワークスペースの視点から行われます。すなわち、ブリングオーバートランザクションは、子が親からファイルを「ブリングオーバーする」(取り込む)ことです。逆に、プットバックトランザクションは、子から親にファイルを「プットバックする」(戻す)ことです。

親ワークスペースの変更を子ワークスペースに適用するには、「更新ブリングオーバー」トランザクションを使用します。子ワークスペースの変更を親ワークスペースに適用するには、「プットバック」トランザクションを使用します。親にファイルをプットバックすることによって、チームの他のメンバーがそのファイルを利用できるようになります。

子ワークスペースの更新 (更新ブリングオーバー)

更新ブリングオーバートランザクションを開始する手順は次のとおりです。

1. 「ワークスペース管理」ウィンドウから「ファイル」▶「ワークスペースの読み込み」を選択して、自分のワークスペースを読み込みます。
2. 読み込んだワークスペースをクリックして選択状態にします。

3. 「アクション」▶「更新プリングオーバー」を選択します。

「トランザクション」ダイアログボックスの「更新プリングオーバー」タブが表示されます (図 2-4 を参照)。

「ワークスペース」区画のフィールドに自動的に親と子の名前が表示されます。これらのフィールドには新しいパス名を入力でき、いつでも変更できます。

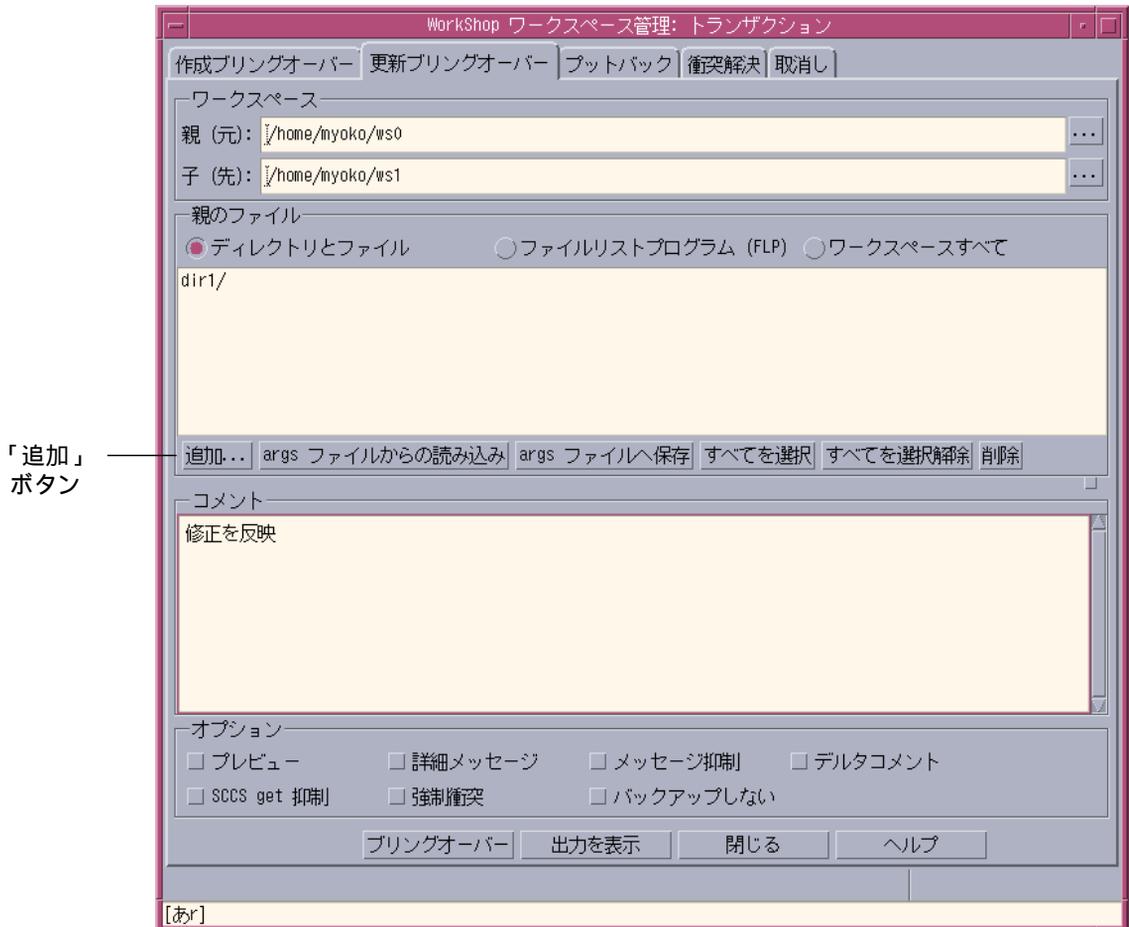


図 2-4 「トランザクション」ダイアログボックスの「更新プリングオーバー」タブ

4. 次のいずれかの操作を行い、プリングオーバーするディレクトリとファイルを選択します。

- デフォルトの「./」を選択し、すべてのファイルをプリングオーバーする。

- 「ワークスペースすべて」ボタンをクリックし、すべてのファイルをブリングオーバーする。
 - 「追加」ボタンをクリックして「ファイル追加」ダイアログボックスを表示し、特定のファイルを追加または削除する。
 - a. 「ファイル追加」ダイアログボックスを使用し、目的のファイルを選択します (図 2-3 を参照)。

このダイアログボックスでは、ディレクトリのアイコンをダブルクリックすることによってファイルシステムの階層を移動できます。特定のディレクトリに直接移動するには、「名前」フィールドにパス名を入力し、「ディレクトリの読み込み」ボタンを選択します。親ワークスペース階層の外へは移動できません。
 - b. ファイルとディレクトリを選択します。

ファイルまたはディレクトリのアイコンをクリックしてください。Shift キーを押しながらクリックすることによって、複数のファイルとディレクトリをまとめて選択できます。
 - c. 「ファイル追加」ボタンをクリックして「更新ブリングオーバー」タブにファイルを追加します。
 - d. 「閉じる」をクリックします。
5. 「更新ブリングオーバー」タブから「ブリングオーバー」ボタンをクリックします。

「トランザクション出力」ウィンドウにブリングオーバーの状態が表示されます。ブリングオーバーが完了すると、そのことが示されます。これで、親ワークスペースにあるファイルの最新版を自分のワークスペースにコピーしたことになります。

注 - 「プレビュー」オプションを選択することによって、ファイルを実際に転送する前にトランザクションの内容を確認できます。

親ワークスペースへの変更内容のプットバック

プットバックトランザクションを開始する手順は次のとおりです。

1. 「ワークスペース管理」ウィンドウから「ファイル」▶「ワークスペースの読み込み」を選択して、自分のワークスペースを読み込みます。
2. 読み込んだワークスペースをクリックして選択状態にします。

3. 「アクション」▶「ブットバック」を選択します。

「トランザクション」ダイアログボックスの「ブットバック」タブが表示されます(図 2-5 を参照)。

「ワークスペース」区画のフィールドに自動的に名前が表示されます。これらのフィールドには新しいパス名を入力でき、いつでも変更できます。

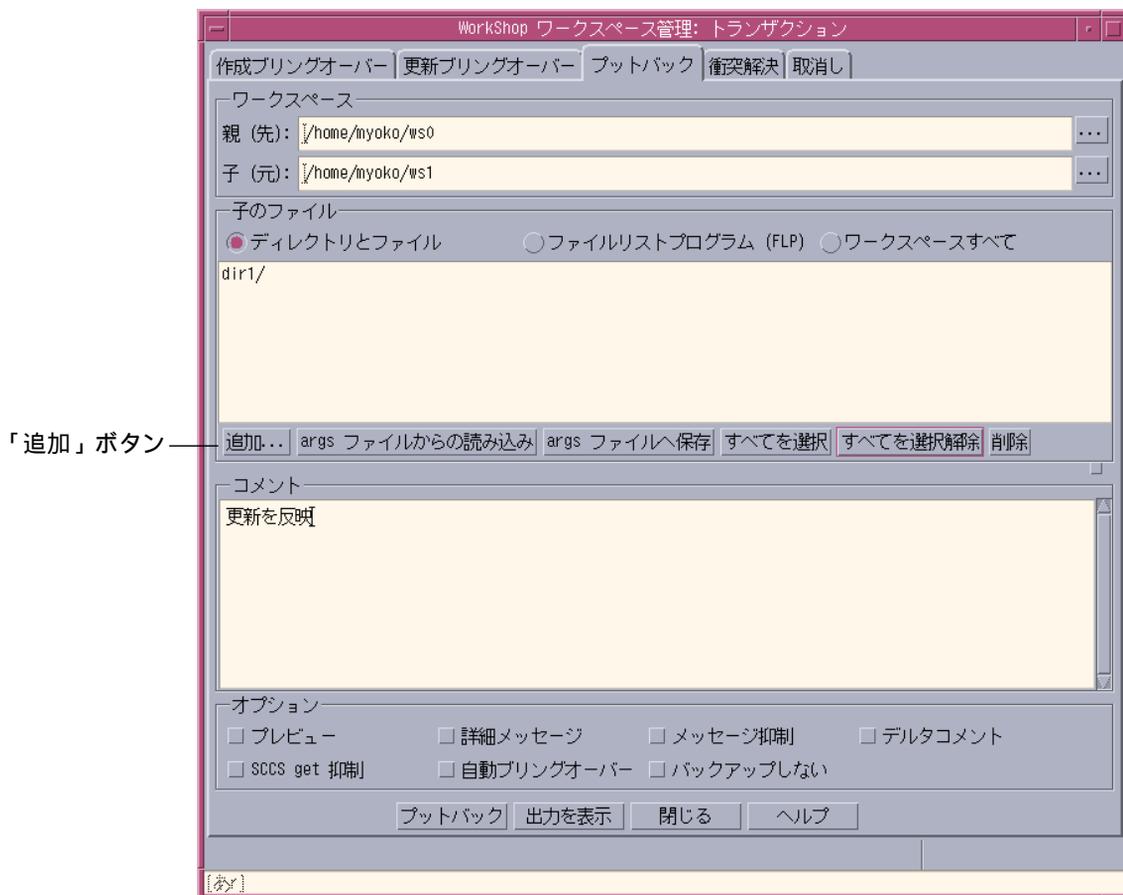


図 2-5 「トランザクション」ダイアログボックスの「ブットバック」タブ

4. 次のいずれかの操作を行い、ブットバックするディレクトリとファイルの選択に進みます。

- デフォルトの「./」を選択し、すべてのファイルをブットバックする。
- 「ワークスペースすべて」ボタンをクリックし、すべてのファイルをブットバックする。

- 「追加」ボタンをクリックして「ファイル追加」ダイアログボックスを表示し、特定のファイルを追加または削除する。
 - a. 「ファイル追加」ダイアログボックスを使用し、目的のファイルを選択します (図 2-3 を参照)。

このダイアログボックスでは、ディレクトリのアイコンをダブルクリックすることによってファイルシステムの階層を移動できます。特定のディレクトリに直接移動するには、「名前」フィールドにパス名を入力し、「ディレクトリの読み込み」ボタンを選択します。親ワークスペース階層の外へは移動できません。
 - b. ファイルとディレクトリを選択します。

ファイルまたはディレクトリのアイコンをクリックしてください。Shift キーを押しながらクリックすることによって、複数のファイルとディレクトリをまとめて選択できます (図 2-3 を参照)。
 - c. 「ファイル追加」ボタンをクリックして、「トランザクション」ダイアログボックスの「プットバック」タブにファイルを追加します (図 2-5 を参照)。
 - d. 「閉じる」をクリックします。
- 5. 現在のプットバックトランザクションに関するコメントを入力します。

入力したコメントは、ワークスペースの「履歴」として保存されます。最大 8K バイトのコメントを入力できます。
- 6. 「プットバック」タブから「プットバック」ボタンをクリックします。

「トランザクション出力」ウィンドウにブリングオーバーの状態が表示されます。ブリングオーバーが完了すると、そのことが示されます。これで、自分の子ワークスペース内のファイルと親ワークスペース内のファイルの内容の同期を取ったことになります。

注 – 子ワークスペースで変更したファイルが、親ワークスペースですでに変更されている場合、そのファイルをプットバックすることはできません。親と子のファイル間の相違を解決する必要があります。詳細は、第 6 章を参照してください。

プットバックトランザクションで行われた処理は、「取消し」トランザクションで取り消すことができます。詳細は、25 ページの「ワークスペースに対する変更の取り消し」を参照してください。

ワークスペースに対する変更の取り消し

「トランザクション」ダイアログボックスの「取消し」タブを使用し、ワークスペースに対する直前のプリングオーバーまたはプットバックトランザクションを取り消すことができます。この取り消しの対象になるのは、そのトランザクションによってファイルが変更された側のワークスペース (転送先のワークスペース) です。別のプリングオーバーまたはプットバックトランザクションによって、そのワークスペースに変更が加えられない限り、トランザクションは何度でも取り消すことができます。ただし、取り消せるのは直前のプリングオーバーまたはプットバックトランザクションだけです。

プットバックまたはプリングオーバートランザクションによってファイルが更新されているか、衝突していることが検出された場合は、取り消しトランザクションはファイルを元の状態に戻します。新規のファイルがあった場合 (つまり、プリングオーバーやプットバックトランザクションによって新たに作成された場合)、そのファイルは削除されます。

取り消しトランザクションを開始する手順は次のとおりです。

1. トランザクションを取り消すワークスペースを選択します。

「取消し」タブを表示する前に、「ワークスペース管理」ウィンドウでワークスペースのアイコンを選択しておく、と、「ワークスペースディレクトリ」フィールドにそのワークスペース名が自動的に表示されます。このフィールドには新しいパス名を入力でき、いつでも変更できます。
2. 「取り消し実行」ボタンをクリックして、トランザクションを開始します。

ワークスペースの名前の変更と移動

ワークスペース管理ツールでは、ワークスペースやファイル、それらの関係の記録を残すために必要な管理作業のすべてを行います。ワークスペース管理ツールを使用して、ワークスペースの名前の変更や、移動、削除をすることによって、ワークスペースの履歴やワークスペース間の関係を管理できます。

注 – ワークスペースの名前変更や移動には、CDE (共通デスクトップ環境) のファイルマネージャや SunOS™ オペレーティングシステムの `mv` コマンドではなく、この節で紹介する手順を使用してください。この手順を使用することによって、ワークスペースの親子関係が保持されます。

ワークスペースの名前変更や移動をするには、「ワークスペース管理」ウィンドウにあるメニューを使用します。

ワークスペースの名前変更または移動をする手順は次のとおりです。

1. 「ファイル」▶「ワークスペースの読み込み」を選択して、ワークスペースを読み込みます。
2. 読み込んだワークスペースをクリックして選択状態にします。
3. 「ワークスペース」▶「名前変更」を選択します。
4. 「名前変更」ダイアログボックスにワークスペースの新しい名前または場所を入力します。
5. 「了解」をクリックします。

「ワークスペース管理」ウィンドウに、新しい名前または場所が示されたワークスペースが表示されます。

ワークスペースの削除と設定の解除

ワークスペースの削除や、ワークスペース設定の解除を行う場合は、オペレーティングシステムのコマンドではなく、この節で説明する手順を使用してください。

ワークスペースの削除

ワークスペースを削除する手順は次のとおりです。

1. 「ファイル」▶「ワークスペースの読み込み」を選択して、ワークスペースを読み込みます。

2. 読み込んだワークスペースをクリックして選択状態にします。

Shift キーを押しながらクリックすることによって、複数のワークスペースをまとめて選択できます。

3. 「ワークスペース」▶「削除」を選択します。

4. 「削除確認」ダイアログボックスで「了解」をクリックします。

削除したワークスペースのアイコンが「ワークスペース管理」ウィンドウから消去されます。

ワークスペース設定の解除

Sun WorkShop TeamWare のワークスペースを通常のディレクトリに戻す手順は次のとおりです。

1. 「ファイル」▶「ワークスペースの読み込み」を選択して、ワークスペースを読み込みます。

2. 読み込んだワークスペースをクリックして選択状態にします。

Shift キーを押しながらクリックすることによって、複数のワークスペースをまとめて選択できます。

3. 「ワークスペース」▶「削除」を選択します。

4. 「Codemgr_wsdata ディレクトリのための削除」ボタンを選択します。

5. 「削除確認」ダイアログボックスで「了解」をクリックします。

削除したワークスペースのアイコンが「ワークスペース管理」ウィンドウから消去されます。ただし、ファイル自体はそのまま残っています。

ワークスペースの履歴表示

ワークスペース管理ツールのトランザクションは、ワークスペース履歴ファイルに記録されます。

- 1つのワークスペースに対するコマンドは、そのワークスペースにだけ記録されません。

- 複数のワークスペースに対するコマンドは、転送元と転送先の両方のワークスペースに記録されます。ただし、コマンドエントリは転送元と転送先の両方に記録されますが、変更されたファイルのリストは、転送先のワークスペースにだけ記録されます。

ワークスペースの履歴ファイルの内容を表示し、ワークスペースに加えられた変更の履歴を調べたり、変更を再構成したりできます。コマンドログのエントリは、コマンド行の入力内容で構成されます。コマンドの意味や構文について不明な点がある場合は、そのコマンドのマニュアルページを参照してください。マニュアルページを表示する方法については、第 11 章を参照してください。

ワークスペースの履歴を表示する手順は次のとおりです。

1. 「ワークスペース管理」ウィンドウでワークスペースをクリックして選択状態にします。
2. 「ワークスペース」▶「履歴の表示」を選択します。
「履歴表示」ウィンドウが表示されます (図 2-6 を参照)。

ワークスペースの履歴表示のカスタマイズ

「履歴表示」ウィンドウに表示される次の項目をカスタマイズできます。

- 表示するトランザクションの情報量
- トランザクションを表示する順序
- 表示する項目

「履歴表示」ウィンドウには、デフォルトでは、トランザクションの状態だけが表示されます。表 2-2 に「履歴表示」ウィンドウに表示できる項目を示します。

表 2-2 ワークスペースの履歴表示の項目

項目	内容
時間	トランザクションの実行日時
操作	トランザクション名
ユーザー	トランザクションを開始したユーザー名
ホスト	トランザクションが開始されたマシン名
TW リリース	トランザクションを実行した TeamWare のバージョン
状態	トランザクションの終了ステータス

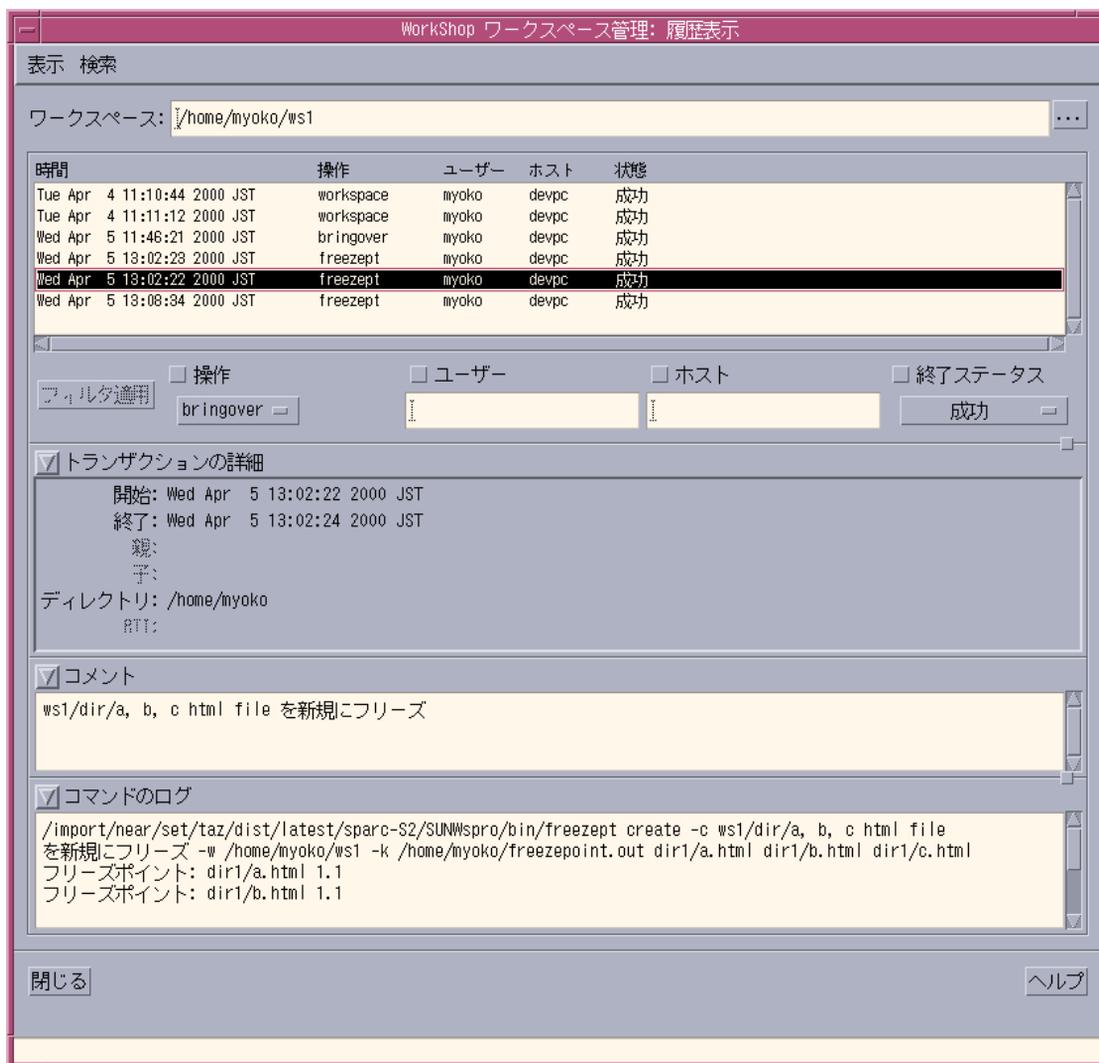


図 2-6 ワークスペースの履歴表示

トランザクション情報の追加と削除

「履歴表示」ウィンドウに表示するトランザクションの情報を追加または削除する手順は次のとおりです。

1. 「履歴表示」ウィンドウから「表示」▶「オプション」を選択します。

2. 「履歴表示オプション」ダイアログボックスの「表示」から、「履歴表示」ウィンドウに追加または削除する項目を選択します。
3. 「了解」をクリックします。
「履歴表示オプション」ダイアログボックスが閉じ、選択した項目が「履歴表示」ウィンドウに追加または削除されます。

トランザクション情報のソート

デフォルトでは、「履歴表示」ウィンドウに表示されるトランザクションは、発生日時を基準にソートされます。最新のトランザクションが履歴の最後に表示されます。

1. 「履歴表示」ウィンドウから「表示」▶「オプション」を選択します。
2. 「履歴表示オプション」ダイアログボックスの「ソート基準」ボタンの1つをクリックします。
3. 「了解」をクリックします。
選択した条件に基づいて、トランザクション情報がソートされます。

フィルタオプションの追加

「履歴表示」ウィンドウに表示するトランザクション情報を、「操作」、「ユーザー」、「ホスト」、「TW リリース」、「終了ステータス」で選別できます。デフォルトでは、トランザクション情報として、すべてのトランザクションが表示され、フィルタオプションは「ユーザー」だけが表示されます。「履歴表示」ウィンドウに、トランザクションの選別に利用するフィルタオプションが表示されていない場合は、そのオプションを追加する必要があります。

「履歴表示」ウィンドウにフィルタオプションを追加する手順は次のとおりです。

1. 「履歴表示」ウィンドウから「表示」▶「オプション」を選択します。
2. 「履歴表示オプション」ダイアログボックスの「フィルタ」から、フィルタとして使用する項目を選択します。
3. 「了解」をクリックします。
「履歴表示」ウィンドウに、選択した項目が表示されます。

表示するトランザクションの選択

「履歴表示」ウィンドウに表示するトランザクション情報を選択する手順は次のとおりです。

1. 「履歴表示」ウィンドウから「表示」▶「オプション」を選択します。
2. 使用するフィルタ (操作、ユーザー、ホスト、TW リリース、終了ステータス) のチェックボックスを選択します。
複数のフィルタを選択できます。
3. 選択したフィルタ条件に従って、必要な項目を入力または選択します。
4. 「フィルタ適用」をクリックします。

トランザクションの検索

コメントまたはコマンドログに含まれている特定の文字列を検索することによって、特定のトランザクションを見つけることができます。

コメントを検索する手順は次のとおりです。

1. 「履歴表示」ウィンドウから「検索」▶「コメント検索」を選択します。
2. 検索する文字列を入力します。
3. 「次を検索」をクリックします。

指定した文字列が見つかると、「履歴表示」ウィンドウの「コメント」区画にその文字列が強調表示されます。

コマンドログを検索する手順は次のとおりです。

1. 「履歴表示」ウィンドウから「検索」▶「コマンドのログの検索」を選択します。
2. 検索する文字列を入力します。
3. 「次を検索」をクリックします。

指定した文字列が見つかると、「履歴表示」ウィンドウの「コマンドのログ」区画にその文字列が強調表示されます。

第3章

高度なワークスペース管理

第2章では、基本的なワークスペース管理作業について説明しました。この章では、高度なワークスペース管理作業について説明します。

- プリングオーバー / プットバックのオプション
- 独自のプリングオーバー / プットバック用ファイルリストの作成
- 他のユーザーへのトランザクションの実行通知
- ワークスペースに分かりやすい名前を付ける
- ワークスペースの親子関係の変更
- ツール属性によるワークスペース管理ツールのカスタマイズ
- 環境変数の設定
- RCS プロジェクトからの変換

ブリングオーバー / プットバックのオプション

ブリングオーバー / プットバックには、次の 2 種類のオプションがあります。

- 個々のブリングオーバー / プットバックの実行時に設定できる「トランザクション」ダイアログボックスのオプション。
- すべてのブリングオーバー / プットバックに適用されるツール属性。

個々のブリングオーバー / プットバック時のオプション設定

個々のブリングオーバー / プットバックにおいてオプションを設定する手順は次のとおりです。

1. 「ワークスペース管理」ウィンドウの「アクション」メニューから、以下のいずれかのトランザクションを選択します。
 - 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
2. オプション区画にあるチェックボックスをクリックします (図 3-1 を参照)。

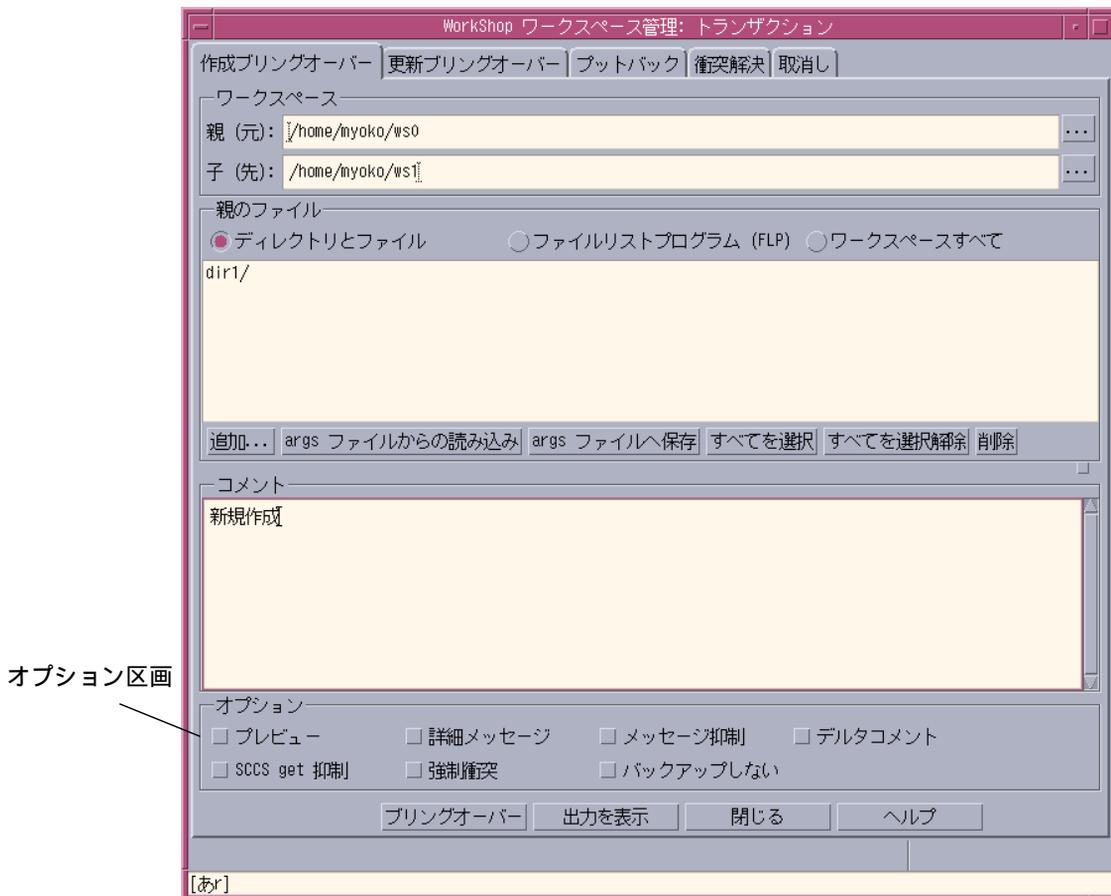


図 3-1 「トランザクション」ダイアログボックスのオプション区画

「トランザクション」ダイアログボックスの「作成プリングオーバー」、「更新プリングオーバー」、「ブットバック」タブのオプション区画にある各オプションの機能を、表 3-1 に示します。

表 3-1 プリングオーバー / ブックアップのオプション

オプション	説明
プレビュー	トランザクションの実行結果をプレビューします。このオプションを選択してトランザクションを開始すると、トランザクションは進行しますが、ファイル転送は行われません。「トランザクション出力」ウィンドウに出力メッセージが表示され、実際にトランザクションを行なった場合の期待する結果を確認できます。
詳細メッセージ	「トランザクション出力」ウィンドウに詳細な情報を表示します。デフォルトの設定では、作成、更新、衝突したファイルに関するメッセージだけが表示されます。「詳細メッセージ」オプションを選択すると、プリングオーバーされなかったファイルを含めて、すべてのファイルに関するメッセージが表示されます。このオプションと「メッセージ抑制」オプションを両方指定した場合は、「メッセージ抑制」オプションが優先されます。
メッセージ抑制	「トランザクション出力」ウィンドウに状態メッセージを出力しません。
デルタコメント	「トランザクション出力」ダイアログボックスに追加情報 (デルタ番号、所有者、コメント) を含めます。
SCCS <code>get</code> 抑制	トランザクションの実行時に SCCS <code>get</code> コマンドが自動的に実行されないようにします。通常は、プリングオーバーを実行した後、SCCS 履歴から <code>g</code> -ファイルが抽出されます。このオプションを選択すると、後でユーザー自身が適切な <code>get</code> を実行する必要がありますが、トランザクションの実行が速くなります。
強制衝突	すべての更新を衝突として処理します。
バックアップしない	既存のファイルを転送先のワークスペースにある <code>Codemgr_wsdata/backup/files</code> ディレクトリにコピーする処理を省略します。このオプションを選択すると、子ワークスペースが使用するディスク領域が減り、トランザクションの実行が速くなりますが、処理を取り消して元に戻すことはできなくなります。

ツール属性オプションの設定

ブリングオーバーやプットバックに対するオプションを設定するには、「ブリングオーバー / プットバック」タブを使用します。「ワークスペース管理」ウィンドウからこのタブを開く手順は次のとおりです。

1. 「オプション」▶「ワークスペース管理」を選択します。
2. 「ブリングオーバー / プットバック」タブを選択します。

表 3-2 に、ブリングオーバーとプットバックの「ツール属性」のチェックボックスを示します。

表 3-2 ブリングオーバー / プットバックのツール属性

属性	説明
トランザクションファイルリスト: 自動読み込み	新しいワークスペースが選択されるたびに <code>Codemgr_wsdata/args</code> ファイルをファイル区画に読み込みます。異なるワークスペースが関係する複数のトランザクションで同じファイルリストを使用する場合は、この属性を選択解除してください。デフォルトはオンです。
トランザクション出力: 自動表示	ブリングオーバー、プットバック、取り消しトランザクションの実行中に「トランザクション出力」ダイアログボックスを表示します。この属性を選択しない場合、「トランザクション出力」ダイアログボックスを表示するには「出力を表示」ボタンを使用する必要があります。デフォルトはオンです。
プットバック: 自動更新ブリングオーバー	プットバックで親ワークスペース内のファイルに変更が加えられていることが検出されたとき、自動的にブリングオーバーを開始します。デフォルトはオフです。
ブリングオーバー / プットバック: コメント拒否の警告	コメントが前回のブリングオーバー / プットバックと同じであることを通知する警告メッセージを表示します。デフォルトはオンです。

独自のブリングオーバー / プットバック用ファイルリストの作成

ワークスペース管理ツールでは、特定のワークスペースにブリングオーバーまたはプットバックするためのファイルのリストを管理します。この節では、ワークスペース管理ツールによるブリングオーバー / プットバック用のデフォルトのファイルリストの作成と、デフォルトリストの変更方法を説明します。ブリングオーバーまたはプットバックするファイルのリストは、ファイルリストプログラム (FLP) を使用して、ユーザー自身が作成することもできます。

デフォルトのファイルリストの保存

ワークスペース管理ツールでは、ブリングオーバーまたはプットバックのときに指定されたファイルとディレクトリのリストを保存します。同じワークスペースにおける次のブリングオーバー / プットバックトランザクション時に、デフォルトで、このリストが読み込まれます。ブリングオーバーまたはプットバックが実行されるたびに、ワークスペース管理ツールは、前回のトランザクションよりファイルリストの項目数が多くなっているかどうかを調べ、項目数が多くなっている場合は、新しいファイルリストに置き換えます。

デフォルトのファイルリストは、「args ファイルからの読み込み」をクリックすることによっていつでも再読み込みできます。この機能は、変更を加えたリストを保存しない場合に有用です。「args ファイルからの読み込み」を使用して、リストをデフォルトの状態に戻すことができます。

デフォルトのリストを変更し、その変更したリストをワークスペース内の [args](#) ファイルのデフォルトにする場合は、「args ファイルへ保存」をクリックします。この機能は、リストから不要なファイルやディレクトリを削除した場合に有用です。ファイルが追加された場合は、ブリングオーバーまたはプットバックトランザクション中に自動的に [args](#) ファイルにそのファイルが追加されます。

独自のファイルリストの作成

ワークスペース管理ツールでは、転送するファイルを個々に指定することも、あるプログラムを実行して、ブリングオーバー / プットバック用に独自のファイルリストを作成することもできます。

デフォルトは、Sun WorkShop TeamWare は、ファイルリストプログラム (FLP) を使用して、ブリングオーバー / プットバックトランザクションの処理対象にするファイルのリストを作成します。FLP はファイルリストを作成し、そのリストをブリングオーバーやプットバックトランザクションに渡します。ワークスペース管理ツールでは、デフォルトの FLP として `def.dir.flp` を使用します。`def.dir.flp` は、ファイルリスト区画に指定されたディレクトリ内の SCCS 管理下にあるファイル名を再帰的に収集し、一覧表示します。ブリングオーバーやプットバックの転送対象には、この FLP (その他のプログラムでもよい) によって収集されたファイルと、ファイルリスト区画で指定したファイルが含まれます。

自分で FLP を作成し、ブリングオーバーやプットバックの対象にするファイルをより細かく管理できます。たとえば、html などの特定の拡張子を持つファイルだけ処理対象にすることができます。トランザクションで独自の FLP を使用する場合は、ファイルリスト区画でそのプログラムを指定します。

ブリングオーバー / プットバックの対象にするファイルのリストを独自に作成する手順は次のとおりです。

1. ブリングオーバー / プットバックをするファイルのリストを作成するファイルリストプログラム (FLP) を作成します。以下に例を示します。

```
# my FLP
cd subdir/webfiles
ls *.html
```

2. ブリングオーバーまたはプットバックのタブから「ファイルリストプログラム (FLP)」ボタンをクリックします。
3. 「追加」ボタンをクリックします。
4. 「FLPS 追加」ダイアログボックスから適切な FLP を選択します。
5. 「ファイル追加」ボタンをクリックします。

これで、現在のトランザクションにだけに使用する FLP プログラムが設定されます。`CODEMGR_DIR_FLP` 環境変数を使用することによって、常に同じ FLP を利用するように設定することもできます。この変数は、ワークスペース管理ツールのデフォルトの FLP (`def.dir.flp`) を書き換えます。

```
% setenv CODEMGR_DIR_FLP /home/workspaces/my.flp
```

コマンド行からブリングオーバーまたはブットバックコマンドを使用する場合は、`-f` オプションを使用して FLP を指定します。コマンド行コマンドの使用方法については、第 11 章を参照してください。

他のユーザーへのトランザクションの実行通知

ワークスペース管理ツールでは、ワークスペースでトランザクションが実行されるたびに、他のユーザーへ電子メールが自動的に送信されるように設定できます。電子メールには、トランザクションの種類とファイル名、トランザクションのコメントが含まれます。

電子メールによる通知の設定をする手順は次のとおりです。

1. ワークスペース管理ツールを起動します。
2. 「ワークスペース」▶「属性」を選択します。
3. 「ワークスペース属性」ダイアログボックスにワークスペース名を入力します。
4. 「通知」タブを選択します。
5. 「通知」タブから「エントリ作成」をクリックします。
6. 「通知入力」ダイアログボックスで次の操作を行います。
 - a. 「メールの宛先」フィールドに電子メールアドレスを入力します。
 - b. 通知するトランザクションを選択します。
 - c. ワークスペース内の特定のファイルについてだけ通知する場合は、「指定」ボタンをクリックします。
 - 「リストに追加」ボタンをクリックしてファイルを選択します。
 - 「ファイル追加」をクリックします。
 - 「取消し」をクリックし、「ファイル追加」ダイアログボックスを閉じます。
 - d. 「通知入力」ダイアログボックスで「了解」をクリックします。
7. 「ワークスペース属性」ダイアログボックスで「了解」をクリックし、新しい設定を保存します。

ワークスペースに分かりやすい名前を付ける

ワークスペース名は、しばしば、`/home/src/rel7/ver2` などの長いパス名になります。Sun WorkShop TeamWare では、"current development area" や "Bob's workspace" などのように、メンバーにとって分かりやすい名前をワークスペースに付けることができます。指定したワークスペース名は、「ワークスペース管理」ウィンドウ（「表示」▶「説明」で指定した名前）を選択します）と電子メールによる通知で使用されます。

各自のワークスペースにも詳細な説明を付けることができます。この説明は、「表示」▶「説明」で指定した名前）を選択すると表示されます。

ワークスペースに分かりやすい名前を付ける手順は次のとおりです。

1. ワークスペース管理ツールを起動します。
2. 「ファイル」▶「ワークスペース読み込み」を選択します。
3. 名前を付けるワークスペースを読み込みます。
4. 読み込んだワークスペースをクリックして選択状態にします。
5. 「ワークスペース」▶「属性」を選択します。
6. 「ワークスペース属性」ダイアログボックスから「説明」タブを選択します。
7. 「名前」テキストボックスに分かりやすい名前を入力します。

入力した名前が、ワークスペースのラベルとして表示されます。

8. 「説明」フィールドにワークスペースに関する情報を入力します。

この情報は、`Codemgr_wsdata/description` ファイルに保存されます。

注 - 「説明」タブから「読み込み」をクリックすると、説明ファイルに含まれている情報が「説明」フィールド表示されます。この際、それまでに「説明」フィールドに入力していた情報は保存されません。

9. 「了解」をクリックします。

指定したワークスペース名を表示するには、次のいずれかの操作を行います。

- 「ワークスペース管理」ウィンドウから「表示」▶「説明」で指定した名前」を選択する。
- `workspace descr` コマンドを使用する。

構文: `workspace descr [-n | -d | -a] wsname ...`

表 3-3 に、`workspace descr` コマンドのオプションを示します。

表 3-3 `workspace descr` コマンドのオプション

オプション	説明
<code>-n</code>	名前だけを表示します。
<code>-d</code>	説明だけを表示します。
<code>-a</code>	名前と説明の両方を表示します (デフォルト)。
<code>wsname</code>	ワークスペース名。

コマンド行コマンドの使用方法については、第 11 章を参照してください。

ワークスペースの親子関係の変更

3 ページの「親ワークスペースと子ワークスペース」で説明したように、ワークスペース同士は、親子関係で結ばれています。ワークスペース管理ツールでは、この親子関係を変更できます。この節では、次のことを説明します。

- ワークスペースの親子関係を変更する理由
- ワークスペースの親子関係を変更する方法
- ワークスペースの親子関係の変更例

ワークスペースの親子関係を変更する理由

ワークスペースの親子関係を永続的または一時的に変更できます。ワークスペースの親子関係を変更する必要があるのは、次のような場合です。

- 最上位のワークスペースを新たに作成する場合。たとえば、開発製品のリリース 1 を完了し、次のリリース 2 に着手する必要がある場合などです。この場合は、次の作業を行います。

1. リリース 2 用の空のワークスペースを新たに作成する (「ファイル」▶「ワークスペース作成」を選択)。
 2. リリース 2 用ワークスペースをリリース 1 用ワークスペースの新しい親にする。
 3. プットバックトランザクションを使用してリリース 2 用ワークスペースにファイルをコピーする。
 4. リリース 1 用ワークスペースの親を元の親に戻す。
- 新しいリリースに機能を移す場合。特定のリリース用に計画していた機能の完成が間に合わなかった場合など、その機能の開発に使用しているワークスペースの親を次のリリースの統合ワークスペースに変更します。
 - 複数のリリースに対して同じバグ修正を適用する場合。バグの修正作業を完了したワークスペースの親を別の階層に移動し、プットバックトランザクションを使用して、関連するワークスペースの新しい親に変更内容を反映します。45 ページの「ワークスペースの親子関係の変更例」では、この例を紹介します。
 - ワークスペース階層構造を変更する場合。次のことを行えます。
 - ワークスペース階層構造に階層を追加する。
 - ワークスペース階層構造から階層を削除する (親子関係の変更時に新しい親を指定しない)。
 - プロジェクト階層内のワークスペース分岐構造を変更する。
 - 親を失ったワークスペースの親子関係を復元する場合。たとえば、親が壊れたり、[Codemgr_wsdata/parent](#) ファイルが削除されたなどの理由でワークスペースの親が失われた場合に、親の変更機能を使用して親子関係を復元できます。

ワークスペースの親子関係を変更する方法

ワークスペースの親子関係を永続的に変更する方法は 2 つあります。「親の変更」コマンドを使用する方法と、ワークスペースのアイコンをドラッグ & ドロップする方法で、機能的にはまったく同じです。プリングオーバーまたはプットバックトランザクション中にワークスペースの親子関係を変更することもできますが、その場合、新しい親子関係はそのトランザクションの間でのみ有効です。

親の変更コマンド

「親の変更」コマンドを使用してワークスペースの親子関係を変更する手順は次のとおりです。

1. ワークスペース管理ツールを起動します。
2. 「ファイル」▶「ワークスペースの読み込み」を選択します。
3. 親を変更するワークスペースを読み込みます。
4. 読み込んだワークスペースをクリックして選択状態にします。
5. 「ワークスペース」▶「親の変更」を選択します。
6. 「親の変更」ダイアログボックスに新しい親の名前を入力します。
7. 「了解」をクリックします。

「親ワークスペースディレクトリ」フィールドに新しい親を指定しなかった場合は、その子ワークスペースは親を失うこととなります。「ワークスペース管理」ウィンドウには、その新しい状態が自動的に反映されます。

ワークスペースアイコンのドラッグ & ドロップ

「ワークスペース管理」ウィンドウでワークスペースのアイコンを選択し、Control キーを押しながら、親のアイコンの上にドラッグ & ドロップすることによって、そのワークスペースの親子関係を変更できます。この場合、変更してもよいかどうかの確認が求められます。新しい親子関係は画面に自動的に反映されます。

ワークスペースのアイコンを選択し、Control キーを押しながらアイコンのない部分にドラッグ & ドロップすることによって、ワークスペースを親から切り離し、独立させることもできます。この場合、ワークスペースには親がなくなり、画面に自動的に反映されます。

親子関係の一時的な変更

「更新プリングオーバー」タブ(「アクション」▶「更新プリングオーバー」で表示)の「親(元)」フィールドに新しい親のパス名を指定することによって、その更新プリングオーバーに対するワークスペースの親子関係を変更できます。また、「プットバック」タブ(「アクション」▶「プットバック」で表示)の「親(先)」フィールドに

新しい親のパス名を指定することによって、そのブットバックトランザクションに対するワークスペースの親子関係を変更することもできます。どちらの場合も、変更した親子関係が有効なのはそのトランザクションに対してだけです。

ワークスペースの親子関係の変更例

製品の1つのバージョンでバグが修正されると、修正したコードを配布するためにパッチが公開されることがよくあります。修正したコードは、通常は製品の次リリースにも適用する必要があります。Sun WorkShop TeamWare を使用して製品を開発すると、親の変更機能を利用することによって、パッチを比較的容易に適用できます。

ここでは、ある製品のリリース 1.0 のバグを修正するためのパッチが作成され、このパッチをリリース 2.0 のコードに適用する例を紹介します。

1. 「ワークスペース管理」ウィンドウに、2つのワークスペース ([Release2.0](#) と [patch1.0](#)) を読み込みます。

この2つのワークスペースには親子関係はありません (図 3-2 を参照)。

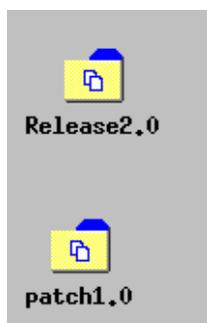


図 3-2 親子関係のない2つのワークスペース

2. 変更を加える前に、作成プリングオーバートランザクションを使用し、[patch1.0](#) ワークスペースの子を作成します (図 3-3 を参照)。

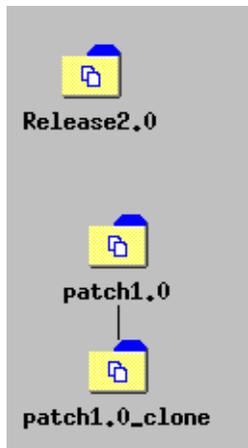


図 3-3 patch1.0 ワークスペースのクローンを作成

3. 「親の変更」コマンドを使用し、`patch1.0_clone` ワークスペースの親を `patch1.0` から `Release 2.0` に変更します (図 3-4 を参照)。

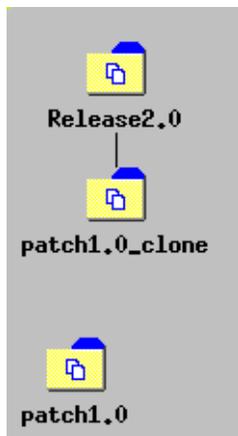


図 3-4 クローンの親を `Release2.0` に変更

4. 新しい親となった `Release2.0` から更新プリングオーバーを実行し、`patch1.0_clone` を更新します。

ここで、`patch1.0_clone` 内のパッチに加えられた修正と、`Release2.0` にあるファイルをマージします (図 3-5 を参照)。

5. `patch1.0_clone` から `Release2.0` に変更をプットバックします。

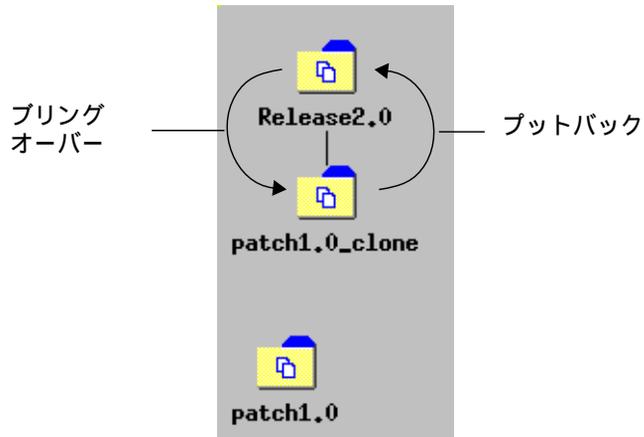


図 3-5 ファイルをブリングオーバー、マージして、新しいリリースに適用

6. これで目的は達成されました。「ワークスペース」▶「削除」を使用し、`patch1.0_clone` を削除できます。

この時点で、`Release2.0` と `patch1.0` という、親子関係を持たない2つのワークスペースがあります。`Release2.0` には、`patch1.0` の修正内容が取り込まれています。`patch1.0` には、トランザクションによる変更は何も加えられていません。これで、このパッチはを `Release2.0` の子で利用できます (図 3-6 を参照)。

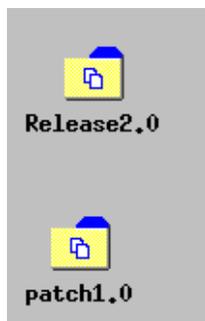


図 3-6 修正内容が取り込まれた `Release2.0` (`patch1.0_clone` は削除)

ツール属性によるワークスペース管理ツールのカスタマイズ

「ツール属性」ダイアログ (図 3-7 を参照) を使用し、次の動作をカスタマイズできます。

- ワークスペース管理ツールのウィンドウ機能
- プリングオーバー / プットバックトランザクション
- 衝突解決トランザクション
- ワークスペースの履歴表示

「ツール属性」ダイアログを開くには、「オプション」▶「ワークスペース管理」を選択します。「ツール属性」ダイアログには、「ワークスペース管理」、「プリングオーバー / プットバック」、「衝突解決」、「履歴表示」を切り替えるためのタブがあります。「衝突解決」タブについては、85 ページの「ファイルマージツールのオプション」を参照してください。

「ワークスペース管理」タブでは、ワークスペース管理ツールのメインウィンドウの動作を変更できます。表 3-4 に、このタブにある属性を示します。

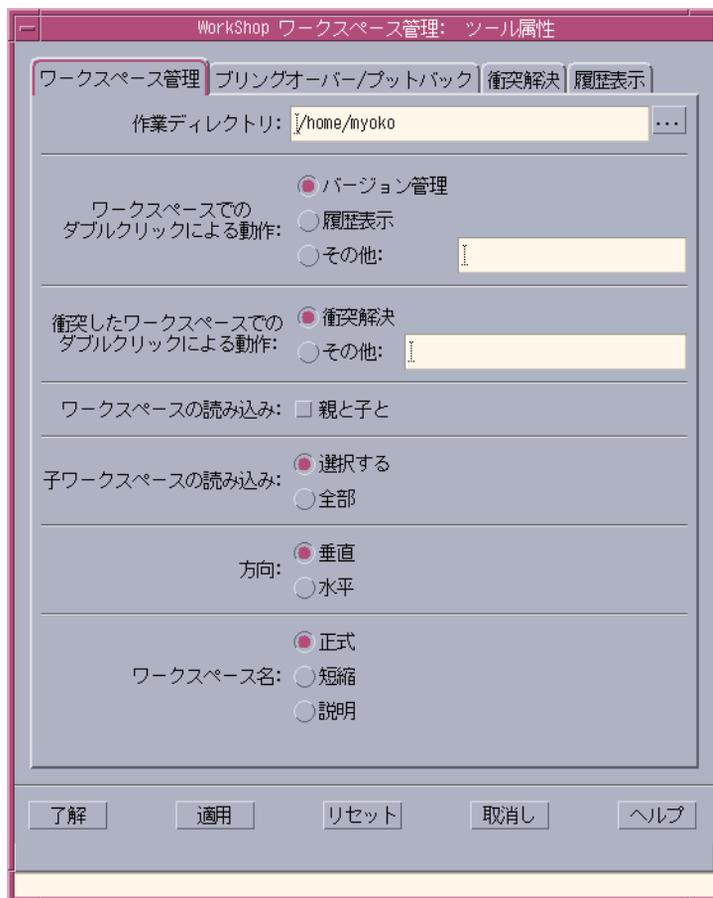


図 3-7 「ツール属性」ダイアログの「ワークスペース管理」タブ

表 3-4 ワークスペース管理ツールの属性

属性	説明
作業ディレクトリ	ワークスペース管理ツールの処理と関係付けられるディレクトリを指定します。
ワークスペースでのダブルクリックによる動作	標準のワークスペースアイコンをダブルクリックしたときに実行するコマンドを指定します。「バージョン管理」「履歴表示」「その他」から選択します。「その他」の場合は、コマンドのパス名を入力します。そのコマンドは、現在の作業ディレクトリと検索パスに基づいて動作します。デフォルトは、バージョン管理ツール (<code>twversion</code>) です。
衝突したワークスペースでのダブルクリックによる動作	衝突があるワークスペースをダブルクリックしたときに実行するコマンドを指定します。現在の作業ディレクトリと検索パスに基づいて動作する、コマンドのパス名を入力する必要があります。デフォルトでは、衝突ファイルの存在するワークスペースに対する衝突解決トランザクションのウィンドウが開きます。
ワークスペースの読み込み	「ワークスペース管理」ウィンドウにワークスペースを読み込むときに、その親と子も自動的に読み込む場合は、このチェックボックスを選択します。デフォルトでは、この属性は選択されていません。
子ワークスペースの読み込み	これらのラジオボタンは、「ワークスペースの読み込み」属性が選択されている場合にだけ有効です。読み込むワークスペースを指定する場合は「選択する」、すべての子を読み込む場合は「全部」を選択します。デフォルトでは、すべての子が読み込まれます。
方向	「ワークスペース管理」ウィンドウに表示されるワークスペースアイコンの階層を横方向に表示する場合は「水平」、縦方向に表示する場合は「垂直」を選択します。デフォルトは「垂直」です。この属性は、ワークスペース管理ツールのメインウィンドウの「表示」▶「方向」に対応しています。

表 3-4 ワークスペース管理ツールの属性 (続き)

属性	説明
ワークスペース名	ワークスペース名の形式を変更します。この属性は、ワークスペース管理ツールのメインウィンドウの「表示」▶「名前」に対応しています。
正式	ワークスペースのフルパス名を表示します (デフォルト)。
短縮	ワークスペースのファイル名だけを表示します。
説明	ワークスペースに付けた名前を表示します。41ページの「ワークスペースに分かりやすい名前を付ける」を参照してください。名前を指定していない場合は、<myworkspace> のようにファイル名が < > で囲まれて表示されます。

環境変数の設定

この節では、ワークスペース管理ツールの以下の動作に関する環境変数の使用例を示します。

- ワークスペースの自動読み込み
- コマンド行コマンドに対するデフォルトのワークスペース設定
- 検索パスの設定

`CODEMGR_FLP_DIR` 変数を使用したデフォルトの FLP の設定方法については、38ページの「独自のファイルリストの作成」を参照してください。

ワークスペースの自動読み込み

`CODEMGR_WSPATH` 変数には、1つのワークスペース、ワークスペースのリスト、またはあるディレクトリ内のすべてのワークスペースを設定できます。`CODEMGR_WSPATH` 環境変数にワークスペースの場所を設定するには、次のように入力します。

```
% setenv CODEMGR_WSPATH /home/ws/myworkspace
```

複数のワークスペースを読み込む場合は、次のように、ワークスペースのリストを引用符で囲みます。

```
% setenv CODEMGR_WSPATH "/home/ws/myworkspace /home/ws/anotherws"
```

特定のディレクトリ内のすべてのワークスペースを読み込むには、次のように、読み込むワークスペースが含まれているディレクトリを設定します。

```
% setenv CODEMGR_WSPATH /home/ws/myworkspaces
```

コマンド行コマンドに対するデフォルトのワークスペース設定

`CODEMGR_WS` 環境変数は、コマンド行コマンドに対するデフォルトのワークスペースを設定します。この変数を設定しておくことで、コマンド行コマンド (`bringover`、`putback`、`freezpt extract` など) で `-w` オプションを使用してワークスペースを指定しなくとも、デフォルトで設定されたワークスペースが自動的に使用されます。デフォルトのワークスペースを設定するには、次のように入力します。

```
% setenv CODEMGR_WS /home/workspaces/myworkspace
```

この変数の設定は、Sun WorkShop TeamWare ツールを起動したときに読み込まれるワークスペースにも影響します。

コマンド行コマンドについては、第 11 章を参照してください。

検索パスの設定

`CODEMGR_PATH_ONLY` 環境変数を使用して、起動する Sun WorkShop TeamWare ツールの場所を指定できます。Sun WorkShop TeamWare は、TeamWare ツールを探るとき、この変数に指定された場所を検索します。`CODEMGR_PATH_ONLY` 環境変数を設定するには、次のように入力します。

```
% setenv CODEMGR_PATH_ONLY /bin/install/Teamware
```

`CODEMGR_PATH_ONLY` 変数が設定されていない場合は、Sun WorkShop TeamWare は、最初に現在のディレクトリ (ツールが動作しているディレクトリ) を検索し、続いて、`PATH` 環境変数に指定されたディレクトリを検索します。

RCS プロジェクトからの変換

`rcs2ws` は、RCS (Revision Control System) ソース階層からワークスペースを作成するプログラムです。RCS で開発されたプロジェクトを変換し、RCS ソース階層をたどって、すべての RCS ファイルを SCCS ファイルに変換します。

`rcs2ws` は、親ディレクトリ下にある RCS ファイルを処理して、SCCS ファイルに変換し、変換した SCCS ファイルをワークスペースに格納します。ワークスペースが存在しない場合は、自動的に作成されます。`rcs2ws` によって、親ディレクトリの階層構造が影響を受けることはありません。`rcs2ws` は、ディレクトリを再帰的に検索します。

ファイルを変換する際に、`rcs2ws` は RCS の `co` コマンドと SCCS の `admin`、`get`、`delta` コマンドを呼び出します。これらのコマンドは、ユーザーの `PATH` 変数を使用して検索されます。SCCS コマンドが見つからない場合は、`/usr/ccs/bin` ディレクトリが検索されます。

注 - `rcs2ws` を使用するには、RCS ユーティリティが必要です。「コマンドが見つかりません。」というエラーメッセージが表示された場合は、RCS が存在するかどうかと、`PATH` 変数に RCS の格納場所が設定されているかどうかを確認してください。

`rcs2ws` は、RCS のキーワードを SCCS のキーワードに変換しません。キーワードは、SCCS デルタ内でテキストとして処理されます。

`rcs2ws` の基本構文は、次のとおりです。

```
rcs2ws -p [RCS_source_dir] -w [teamware_workspace] [files | directory]
```

`-p` オプションは必須で、RCS ソースディレクトリを指定します。相対ファイル名は、`RCS_source_dir` からの相対的なファイル名として解釈されます。

`-w` オプションには、TeamWare ワークスペースを指定します。指定したワークスペースが存在しない場合は作成されます。TeamWare ワークスペースがすでに存在していて、デフォルトのワークスペースになっている場合や、現在のディレクトリが既存の TeamWare ワークスペース内のディレクトリに含まれている場合は、`-w` オプションは省略できます。

たとえば、RCS プロジェクトである `/projects/prodA/release1` を変換して、新しい TeamWare ワークスペース `/tw/workspaces/dev1` を作成する場合は、次のコマンドを入力します。

```
% rcs2ws -p /projects/prodA -w /tw/workspaces/dev1 release1
```

`/tw/workspaces/dev1` ワークスペースがすでに存在していて、デフォルトのワークスペースである場合は、次のコマンドを使用できます。

```
% rcs2ws -p /projects/prodA release1
```

`RCS_source_dir` 下にあるすべての RCS ファイルを変換する場合は、`."` を使用します。たとえば、プロジェクトディレクトリの `/projects/prodA` 内のすべての RCS ファイルを変換する場合は、次のコマンドを入力します。

```
% rcs2ws -p /projects/prodA .
```

`rcs2ws` オプションについての詳細は、`rcs2ws(1)` のマニュアルページを参照してください。

第4章

ワークスペースへのアクセス制御

Sun WorkShop TeamWare では、アクセス権の設定とプットバックの妥当性検査によって、どのユーザーがワークスペースにファイルをプットバックできるのかを制御できます。この章では、以下について説明します。

- アクセス権の設定
- プットバックの妥当性検査によるワークスペースの保護
- ワークスペースのロックの解除

アクセス権の設定

Sun WorkShop TeamWare では、全体、グループ、個人ユーザーのレベルでアクセス権を許可または禁止できます。全体レベルでのアクセス権の制御については、56 ページの「全体レベルでのアクセス制御」、グループまたは個人ユーザーレベルでのアクセス権の制御については、58 ページの「グループまたはユーザーレベルでのアクセス制御」を参照してください。

個人ユーザーまたはグループレベルで、次に挙げる Sun WorkShop TeamWare のトランザクションの実行を許可または禁止できます。

- bringover-from
- bringover-to
- putback-from
- putback-to
- undo
- workspace delete
- workspace move

- workspace reparent
- workspace reparent-to

注 – アクセス制御の設定をすると、設定をしていないときよりも、トランザクションの実行に時間がかかることがあります。多人数のグループにアクセス制御を設定していると、トランザクションを実行する前にグループのメンバーを調べるために数秒かかることがあります。

全体レベルでのアクセス制御

全体にわたるアクセス制御を設定する手順は次のとおりです。

1. ワークスペース管理ツールを起動します。
2. 「ファイル」▶「ワークスペースの読み込み」を選択します。
3. 「ワークスペース読み込み」ダイアログボックスでワークスペースを選択し、「ワークスペースの読み込み」をクリックします。
4. 読み込んだワークスペースをクリックして選択状態にします。
5. 「ワークスペース」▶「属性」を選択します。
6. 「ワークスペース属性」ダイアログボックスの「アクセス制御」タブを選択します。

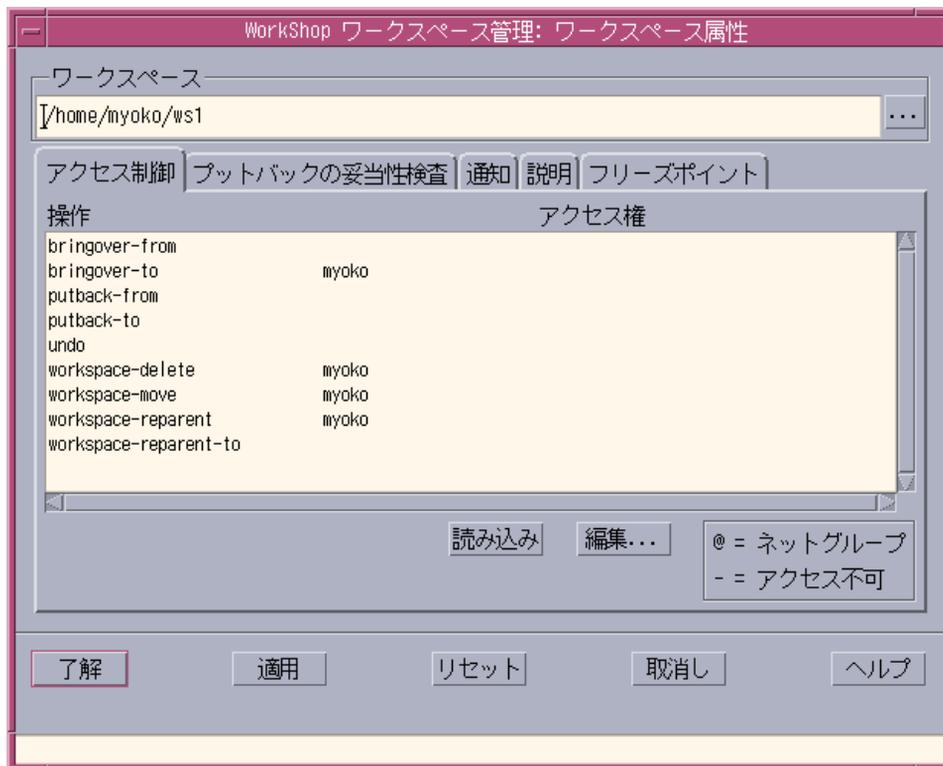


図 4-1 「ワークスペース属性」ダイアログボックスの「アクセス制御」タブ

- 「操作」リストからアクセス権を設定するトランザクションを選択し、「編集」をクリックします。

「アクセス制御属性」ダイアログボックスが表示されます。

- すべてのユーザーに適用するアクセス権を選択します。

- なし: どのユーザーにもアクセス権を付与しません。
- 全部: すべてのユーザーにアクセス権を付与します。

デフォルトでは、あらゆるユーザーにトランザクションの実行アクセス権が付与されます。

- 「了解」をクリックします。

グループまたはユーザーレベルでのアクセス制御

1. ワークスペース管理ツールを起動します。
2. 「ファイル」▶「ワークスペースの読み込み」を選択します。
3. 「ワークスペース読み込み」ダイアログボックスでワークスペースを選択し、「ワークスペースの読み込み」をクリックします。
4. 読み込んだワークスペースをクリックして選択状態にします。
5. 「ワークスペース」▶「属性」を選択します。
6. 「ワークスペース属性」ダイアログボックスの「アクセス制御」タブを選択します。
7. 「操作」リストからアクセス権を設定するトランザクションを選択し、「編集」をクリックします。
「アクセス制御属性」ダイアログボックスが表示されます。
8. 「指定」ボタンをクリックします。
9. 「名前」フィールドにユーザーのログイン名 (またはグループ名) を入力します。
10. 「ユーザー」または「ネットグループ」ボタンをクリックします。入力したログイン名が個人ユーザーのログイン名の場合は「ユーザー」、グループ名の場合は「ネットグループ」をクリックします。
11. 「許可する」または「許可しない」ボタンをクリックします。
12. 「前に挿入」または「後ろに挿入」ボタンをクリックして、リストにユーザーを追加します。リストがない場合は、新たにリストを作成します。
13. 「了解」をクリックします。
「アクセス制御」タブのアクセス権リストにユーザー名 (またはネットグループ名) が表示されます。

注 - リスト内のユーザーの登録位置によって、そのユーザーのアクセス制御が影響を受けることがあります。同じユーザーに対して「許可する」と「許可しない」の両方が指定されている場合は、最初の登録だけが参照されます。このような状況は、個人ユーザーにはアクセスが許可されているにも関わらず、そのユーザーが所属するグループに対してはアクセスが許可されていない場合などに発生する可能性があります。この場合、ユーザーが先に登録されていれば、ユーザーにはアクセスが許可されます。グループが先に登録されている場合は、ユーザーへのアクセスは許可されません。

プットバックの妥当性検査によるワークスペースの保護

特定のワークスペースを厳重に管理するには、プットバックの妥当性検査機能を使用します。プットバックの妥当性検査機能を有効にすることによって、特定のユーザーだけがワークスペースにプットバックできるように設定できます。ユーザーは、プットバックを行う前に、統合要求 ID (Integration Request Identifier) と呼ばれる「パスワード」の入力が求められます。ワークスペース管理ツール自体は、入力された統合要求 ID の妥当性を検査せずに、別のプログラムに渡します。入力された ID の妥当性を検査するには、自分でそのためのプログラムを作成する必要があります。詳細は、61 ページの「プットバックの妥当性検査プログラムの起動」を参照してください。プットバックの妥当性検査が有効になっていても、妥当性検査をするためのプログラムがない場合は、ワークスペース管理ツールは常にプットバックを許可します。入力された統合要求 ID は、ワークスペース履歴ファイルの見出し語 RTI (Request To Integrate) の行に記録されます。

表 4-1 に、ブットバックの妥当性検査の 3 つのモードをまとめます。

表 4-1 ブットバックの妥当性検査のモード

モード	ユーザーの操作	ワークスペース管理ツール
妥当性検査をしない (デフォルト)	なし	ワークスペースに対するすべてのブットバックを許可します。
妥当性検査をする	ブットバックの妥当性検査を有効にします。	ブットバックを許可する前に「パスワード」(統合要求 ID) の入力を要求します。パスワードの妥当性は検査されず、単に記録されるだけです。
妥当性検査をする パスワードを検査	ブットバックの妥当性検査プログラムを作成し、ブットバックの妥当性検査を有効にして作成したプログラムを実行する。	ブットバックを許可する前に「パスワード」(統合要求 ID) の入力を要求し、入力されたパスワードを妥当性検査プログラムに渡します。妥当性検査プログラムは、ブットバックを許可する前にアクセス権を付与する必要があります。

SCCS ではまた、各ファイルに対するチェックインの妥当性検査を要求することもできます。詳細は、110 ページの「SCCS ファイルの属性の設定」を参照してください。

ブットバックの妥当性検査を有効にする

ブットバックの妥当性検査を有効にする手順は次のとおりです。

1. 「ワークスペース管理」ウィンドウから「ファイル」▶「ワークスペースの読み込み」を選択し、ワークスペースを読み込みます。
2. 読み込んだワークスペースをクリックして選択状態にします。
3. 「ワークスペース」▶「ワークスペース属性」を選択します。
4. 「ワークスペース属性」ダイアログボックスから「ブットバックの妥当性検査」タブをクリックします (図 4-2 を参照)。

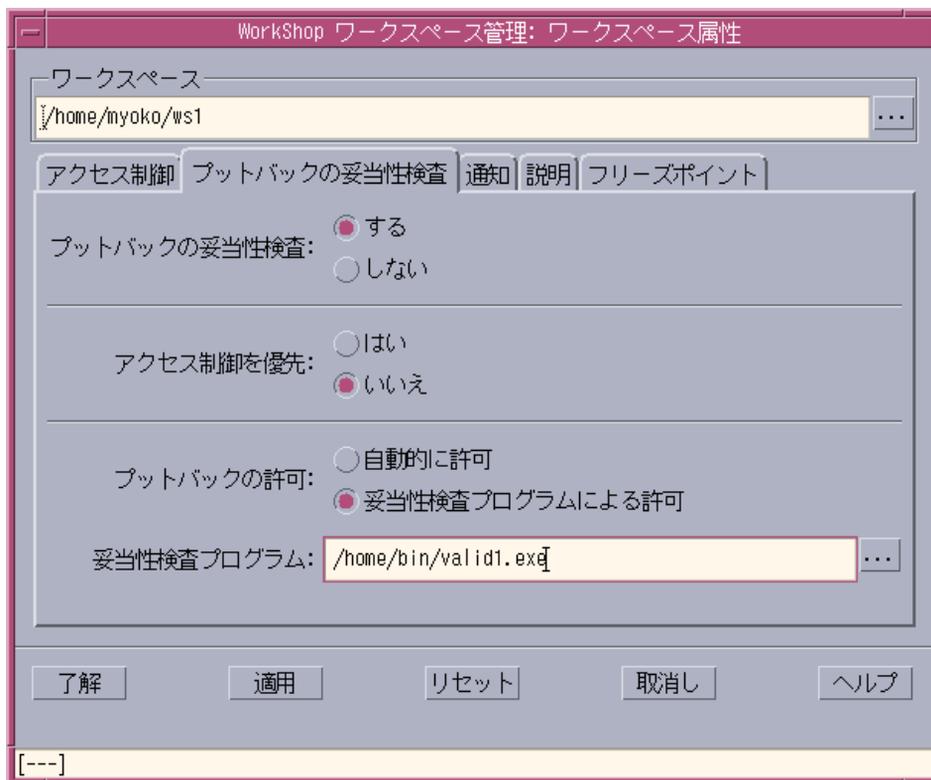


図 4-2 「ワークスペース属性」ダイアログボックスの「ブットバックの妥当性検査」タブ

5. 「ブットバックの妥当性検査:」の「する」をクリックして設定を有効にします。
6. 「了解」をクリックします。

ブットバックの妥当性検査プログラムの起動

ワークスペースに対して妥当性検査プログラムを設定すると、そのワークスペースに対するブットバックランザクションが試みられるたびに、その妥当性検査プログラムが起動されます。プログラムで問題が発生すると、ブットバックはブロック (阻止) されます。

ブットバックの妥当性検査プログラムは、Sun WorkShop TeamWare には付属していません。妥当性検査プログラムは、ユーザー自身が用意する必要があります。コード例 4-1 に妥当性検査プログラムの例を示します。

ブットバックの妥当性検査プログラムを起動する手順は次のとおりです。

1. 「ワークスペース管理」ウィンドウから「ファイル」▶「ワークスペースの読み込み」を選択し、ワークスペースを読み込みます。
2. 読み込んだワークスペースをクリックして選択状態にします。
3. 「ワークスペース」▶「ワークスペース属性」を選択します。
4. 「ワークスペース属性」ダイアログボックスから「ブットバックの妥当性検査」タブをクリックします。
5. 「ブットバックの妥当性検査:」の「する」をクリックして設定を有効にします。
6. 「ブットバックの許可:」の「妥当性検査プログラムによる許可」をクリックします。
7. 「妥当性検査プログラム」フィールドに使用する妥当性検査プログラムのパスを入力します。
8. 「了解」をクリックします。

妥当性検査プログラムのコード例

特定のワークスペースに対するブットバックを許可するかどうかを検査するためのプログラム例を紹介します。妥当性検査プログラムは、ブットバックによって渡される引数を受け取り、ブットバックを拒否するときにゼロ以外の終了ステータスを返す実行可能なファイル (シェルスクリプト、プログラムなど) であれば、どのようなプログラムでもかまいません。

ブットバックを試みると、統合要求 ID の入力が必要とされ、続いて、次の引数を付加して妥当性検査プログラムが起動されます。

- ブットバックを実行しようとするユーザーのユーザー ID (\$1)
- 転送元 ("from") ワークスペースのフルパス名 (\$2)
- 転送先 ("to") ワークスペースのフルパス名 (\$3)
- 統合要求 ID (\$4)
- 変更するファイルのリストが含まれているファイル名 (\$5)

コード例 4-1 のプログラムは、すべての引数を検査し、問題がなかった場合は、終了ステータスとしてゼロを返してプットバックを許可します。

コード例 4-1 プットバックの妥当性検査プログラムの例

```
## Start of sample validation program
#!/bin/sh
##
## Here is the list of users which can run
## putback to the workspace
##
valid_users="nikm azv builder vvg aar"
##
## Here is the list of Integration Request Ids which
## are accepted
##
P1Bugs="1111111 2222222 1234567 bobs_bug"
##
## Here are the directories which cannot change
##
DirsReady="doc subdir/doc"
##
## Save arguments
##
User=$1
    shift
Parent=$1
    shift
Child=$1
    shift
IRI=$1
    shift
Files='cat $1`
##
## Validate user
##
isValid="false"
for u in $valid_users
do
    if [ "$User" = "$u" ] ; then
        isValid="true"
        break
    fi
if [ "$isValid" = "false" ] ; then
    # invalid user
    echo ""
    echo "*** Validation failed: User $User is not allowed \
        to putback to $Parent"
    echo ""
    exit 1
fi
```

コード例 4-1 プットバックの妥当性検査プログラムの例 (続き)

```
##
## Validate Integration Request Id (IRI)
##
isValid="false"
for u in $P1Bugs
do
  if [ "$IRI" = "$u" ] ; then
    isValid="true"
    break
  fi
done
if [ "$isValid" = "false" ] ; then
  # invalid IRI
  echo ""
  echo "*** Validation failed: Integration request $IRI \
    is invalid"
  echo ""
  exit 1
fi
##
## Validate files
##
for u in $Files
do
  for uu in $DirsReady
  do
    x=`echo $u | grep $uu`
    if [ "$x" != "" ] ; then
      isValid="false"
      echo ""
      echo "*** Validation failed: File $u \
        cannot be changed"
      echo ""
      exit 1
    fi
  done
done
##
## Exit 0 - putback is allowed
##
exit 0
## End of sample validation program
```

ワークスペースのロックの解除

整合性を保つために、ワークスペース管理ツールは、トランザクション (ブリングオーバー、取り消し、プットバック) が実行されている間は、その処理対象のワークスペースをロックします。

ロックされているワークスペースにファイルをブリングオーバーしようとする、ロックしているユーザー名と実行しようとしているコマンド、ロックが取得された時刻が示されたエラーメッセージが返されます。

```
bringover: 読み込みロックを取得できません。ワークスペース
"/home/my_home/projects/mpages" に以下のロックが存在しています:
  ユーザー jack は putback コマンドを実行中 (pid 20291)
  マシン holiday で 10 分
  (エラー 2021)
```

トランザクションが終了すると、これらのロックは自動的に解除されます。このため、通常はこうしたエラーメッセージが返されることはありませんが、トランザクションで問題が発生すると、ロックが解除されずに残ることがあるため、ワークスペースにアクセスできないことがあります。他に誰もそのワークスペースにアクセスしていないことが確実な場合は、手動でロック解除してください。

注 - ロックを解除する前に、そのワークスペースを変更してるユーザーがいないことを確認してください。変更中のワークスペースのロックを解除すると、ワークスペースの完全性が失われます。

ワークスペースのロックを解除する手順は次のとおりです。

1. ロックを解除するワークスペース内で、誰もファイルを変更していないことと、トランザクションが実行されていないことを確認します。
2. 「ワークスペース管理」ウィンドウから「ファイル」▶「ワークスペースの読み込み」を選択して、ワークスペースを読み込みます。
3. 読み込んだワークスペースをクリックして選択状態にします。
4. 「ワークスペース」▶「ロックの編集」を選択します。

5. 「ロック編集」ダイアログボックスで解除するロックをクリックして、選択状態にします。
6. 「削除」をクリックします。
7. 「了解」をクリックして「ロック編集」ダイアログボックスを閉じます。

第5章

ファイルの管理

TeamWare ワークスペースを作成すると、プロジェクトファイルを使用して作業を開始できます。複数のユーザーがソースファイルに変更を加える場合は、それらのファイルに対する書き込みアクセスを調整することが重要です。ファイルの更新記録を残すことによって、いつ、どのような理由で変更が加えられたのかを調べることができます。バージョン管理ツールは、ファイルを管理するための TeamWare ツールです。この章では、バージョン管理ツールで次の基本作業を行う方法を説明します。

- バージョン管理ツールの起動
- ワークスペースへのファイルの追加
- ファイルのチェックアウト
- ファイルの編集
- デフォルトのエディタの変更
- ファイルのチェックイン
- ファイルに対する変更の取り消し
- プットバックによる変更の統合

バージョン管理ツールの起動

バージョン管理ツールを使用して、ソースファイルに対する書き込みアクセスの制御や、ファイルに対する変更の監視を行うことができます。バージョン管理ツールでは、一度に1人のユーザーだけにファイルの更新を許可します。すべての変更は履歴ファイルに記録されます。

- バージョン管理ツールを起動するには、「ワークスペース管理」ウィンドウから「TeamWare」▶「バージョン管理」を選択します。

バージョン管理ツールを起動すると、「バージョン管理」ウィンドウが開きます。バージョン管理ツールのメインウィンドウには、現在のディレクトリにあるファイルとサブディレクトリが表示されます (図 5-1 を参照)。

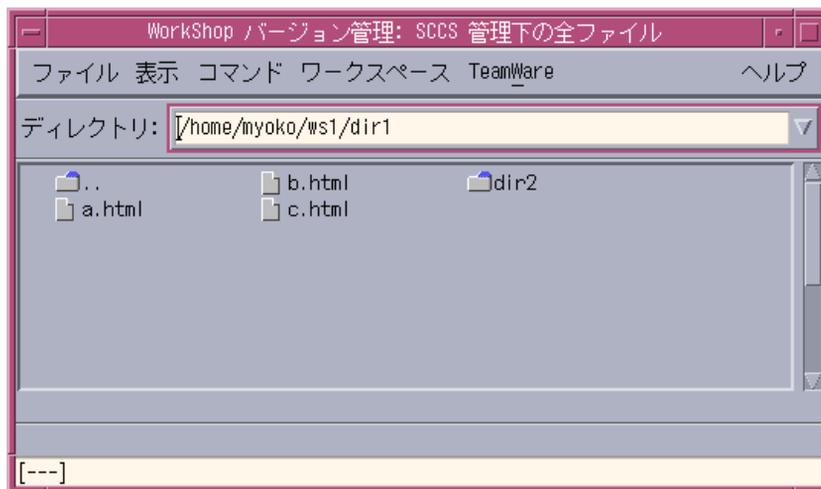


図 5-1 「バージョン管理」ウィンドウ

「バージョン管理」ウィンドウには、プロジェクトのディレクトリとファイルが表示されます。ウィンドウには、すべてのディレクトリが表示されますが、ファイルについては、SCCS 管理下にあるファイルだけが表示されます。詳細は、15 ページの「親ワークスペースの作成」を参照してください。

「バージョン管理」ウィンドウにあるメニューを、表 5-1 に示します。

表 5-1 「バージョン管理」ウィンドウのメニュー

メニュー	説明
ファイル	ファイルを管理するためのコマンドがあります。
表示	ディレクトリ区画に表示するアイコンを制御、更新するためのコマンドがあります。
コマンド	ファイルをチェックイン、チェックアウトするためのコマンドがあります。

表 5-1 「バージョン管理」ウィンドウのメニュー (続き)

メニュー	説明
ワークスペース	SCCS 管理下にあるファイルの検索、削除、移動をするためのコマンドがあります。
TeamWare	他の TeamWare ツールを起動するためのコマンドがあります。
ヘルプ	オンラインヘルプとリリース情報を表示するためのコマンドがあります。

「バージョン管理」ウィンドウでは、ディレクトリについては、含まれているファイルが SCCS 管理下にあるかどうかには関係なく表示されます。ディレクトリはファイルとディレクトリを収納する入れ物なので、その階層の下にさらに SCCS ファイルやディレクトリが含まれている場合があります。ファイルは、SCCS 管理下にある場合にだけ表示されます。ディレクトリに SCCS 管理下にはないファイルを見るには、「新たにチェックイン」ウィンドウを使用します。

ワークスペースへのファイルの追加

ワークスペース管理ツールがファイルを認識するには、そのファイルが SCCS でバージョン管理されている必要があります。

TeamWare ワークスペースにファイルを追加する手順は次のとおりです。

1. ワークスペース管理ツールを起動します。
2. 「ワークスペース管理」ウィンドウから「ファイル」▶「ワークスペースの読み込み」を選択して、ワークスペースを読み込みます。

ディレクトリを TeamWare ワークスペースにしていない場合は、15 ページの「親ワークスペースの作成」を参照してください。

3. 読み込んだワークスペースをクリックして選択状態にします。
4. 「TeamWare」▶「バージョン管理」を選択します。
5. 「バージョン管理」ウィンドウから「コマンド」▶「新たにチェックイン」を選択します。

6. 「新たにチェックイン」ダイアログボックスで、ワークスペースに追加するファイルを選択します。
Shift キーを押しながらクリックすることによって、複数のファイルをまとめて選択できます。
7. 「了解」をクリックします。
「バージョン管理」ウィンドウに、チェックインしたファイルが表示されます。

ファイルのチェックアウト

ファイルを変更するには、ワークスペースからファイルをチェックアウトする必要があります。ファイルをチェックアウトすると、他のユーザーが同じファイルをチェックアウトできなくなります。同じファイルに 2 人のユーザーがアクセスする必要がある場合は、ユーザーの 1 人が、自分の子ワークスペースを作成して、そこにファイルのコピーを作成するようにします。

ファイルをチェックアウトする手順は次のとおりです。

1. 「ワークスペース管理」ウィンドウから「TeamWare」▶「バージョン管理」を選択します。
2. 「バージョン管理」ウィンドウから、使用するファイルが格納されているディレクトリに移動し、変更するファイルのアイコンを選択します。
Shift キーを押しながらクリックすることによって、複数のファイルをまとめて選択することもできます。
3. 「コマンド」▶「チェックアウト」を選択します。
「コマンド」▶「チェックアウトして編集」を選択すると、自動的にエディタが起動されます。
選択したファイルのアイコン上にチェックマークが表示されます。デフォルト以外のエディタを使用する場合は、71 ページの「デフォルトのエディタの変更」を参照してください。

ファイルの編集

ファイルを編集する手順は次のとおりです。

1. 「ワークスペース管理」ウィンドウから「ファイル」▶「ワークスペースの読み込み」を選択して、ワークスペースを読み込みます。
2. 読み込んだワークスペースをクリックして選択状態にします。
3. 「Teamware」▶「バージョン管理」を選択します。
「バージョン管理」ウィンドウが表示されます。
4. 「バージョン管理」ウィンドウで、編集するファイルをクリックして選択状態にします。
5. 「コマンド」▶「チェックアウトして編集」を選択します。
デフォルトのエディタのウィンドウが開いて、選択したファイルが表示されます。デフォルトのエディタを変更する方法については、「デフォルトのエディタの変更」を参照してください。
6. ファイルを編集して、保存したら、ファイルをチェックインします。
72 ページの「ファイルのチェックイン」を参照してください。

デフォルトのエディタの変更

デフォルトのエディタを変更する手順は次のとおりです。

1. 「ワークスペース管理」ウィンドウから「TeamWare」▶「バージョン管理」を選択します。
2. 「バージョン管理」ウィンドウから「表示」▶「オプション」を選択します。
「オプション」ダイアログボックスが表示されます。
3. 「一般」タブにある「エディタの選択」で、デフォルトで使用するエディタを選択します。

4. 「その他」を選択した場合は、表示されたフィールドにエディタへのパスを入力します。

ファイルのチェックイン

ファイルをチェックインする手順は次のとおりです。

1. 「バージョン管理」ウィンドウで、チェックインするファイルが存在するディレクトリに移動します。
2. チェックインするファイルのアイコンをクリックします。
Shift キーを押しながらクリックすることによって、複数のファイルをまとめて選択することもできます。チェックアウトされているファイルのアイコンには、赤いチェックマークが付いています。
3. 「コマンド」▶「チェックイン」を選択します。
4. 「チェックイン」ダイアログボックスにコメントを入力します。
5. 「了解」をクリックします。

Sun WorkShop TeamWare によって、コメントが保存され、ファイルがチェックインされます。加えた変更を親ワークスペースに統合する方法については、73 ページの「ブットバックによる変更の統合」を参照してください。

ファイルに対する変更の取り消し

バージョン管理ツールでは、ファイルをチェックアウト前の状態に戻すこともできます。この機能は、ファイルをチェックアウトして変更したが、その変更したファイルはチェックインしないといった場合に使用します。

チェックアウトしたファイルに行なった変更を取り消す手順は次のとおりです。

1. 「ワークスペース管理」ウィンドウから「TeamWare」▶「バージョン管理」を選択します。

2. 「バージョン管理」ウィンドウから、変更を取り消すファイルが格納されているディレクトリに移動します。
チェックアウトされているファイルのアイコンには、赤いチェックマークが付いています。
3. ファイルをクリックして選択状態にします。
Shift キーを押しながらクリックすることによって、複数のファイルをまとめて選択できます。
4. 「コマンド」▶「アンチェックアウト」を選択します。
ファイルに最後に加えた変更が取り消され、ファイルが以前の状態に戻ります。

プットバックによる変更の統合

ワークスペース内のファイルに変更を加えて、その変更を親ワークスペースに反映するには、親ワークスペースにその変更をプットバックする必要があります。プットバックすることによって、チームの他のメンバーが変更された内容を利用できるようになります。

ファイルを親ワークスペースにプットバックする手順は次のとおりです。

1. 「ワークスペース管理」ウィンドウから「アクション」▶「プットバック」を選択します。
2. 「プットバック」タブでプットバックするファイルを選択します。
3. 「コメント」フィールドにコメントを入力します。
電子メールによる通知の設定によっては、多くの人がこのコメントを見ることになります。
4. 「プットバック」をクリックします。
「トランザクション出力」ウィンドウが表示され、プットバックに関する情報が表示されます。プットバック中に衝突が発生した場合は、自動的に「衝突解決」タブが表示されます。第6章を参照してください。
5. 「閉じる」をクリックして、「トランザクション出力」ウィンドウを閉じます。

注 - プットバックについての詳細は、22 ページの「親ワークスペースへの変更内容のプットバック」を参照してください。

第6章

ファイル間の相違の解決

同じファイルの2つのコピーの両方に変更を加えた場合、それらのファイル(コピー)は衝突状態にあります。自分の子ワークスペースで変更したファイルをプットバックしようとした際に、親ワークスペースのファイルも更新されていた場合には、これらのファイル間の相違を解決するまで、そのファイルのプットバックはできません。ファイルマージツールは、ファイル間の相違を解決するためのツールです。

この章では、次のトピックについて説明します。

- ファイルマージツールの起動
- ワークスペース内での衝突の解決
- 「ファイルマージ」ウィンドウの表示
- 相違の解決
- 自動マージの使用方法
- 変更の取り消し
- ファイルマージツールのオプション
- ファイルのマージ例

ファイルマージツールの起動

「ワークスペース管理」ウィンドウでは、衝突があるワークスペースは、通常とは異なるアイコンで表示されます。ワークスペースアイコン内のファイルの1つが赤く表示され、放射状に出ている赤い線によって衝突が存在することが示されます。図 6-1 の例は、`ws2` ワークスペースに、その親の `ws1` と衝突するファイルが存在することを示しています。

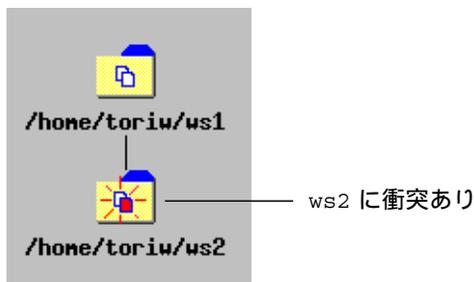


図 6-1 ワークスペースの衝突例

衝突をすぐに解決する必要はありません。そのまま変更を続けて、子ワークスペース内に衝突しているファイルの新しいデルタを作成できます。新しいデルタは全体が1つの分岐になります。最終的に衝突を解決すると、最新のデルタが、親からプリングオーバーしたバージョンとマージされます。親にファイルをプットバックするには、あらかじめ衝突を解決しておく必要があります。ファイルが衝突した状態でプットバックしようとする、プリングオーバーをするように求められ、「トランザクション」ダイアログボックスの「衝突解決」タブに衝突しているファイルが表示されます。

ファイルマージツールは、次の2つの場所から起動できます。

- 「Teamware」メニュー
- 「トランザクション」ダイアログボックスの「衝突解決」タブ

「TeamWare」メニューからのファイルマージツールの起動

ファイルマージツールを起動するには、ワークスペース管理ツール、バージョン管理ツール、フリーズポイントツールのいずれかのメインウィンドウから「TeamWare」▶「ファイルマージ」を選択します。

「衝突解決」タブからのファイルマージツールの起動

「トランザクション」ダイアログボックスの「衝突解決」タブは、ユーザーとファイルマージ機能の仲介役として、更新プリングオーバートランザクション中に検出された衝突しているファイルの一覧を表示します。衝突解決トランザクションでは、衝突しているファイルを選択し、そのファイルが読み込まれた状態でファイルマージツ

ルを起動できます。プットバック中に衝突の解決を行うように設定すると、自動的に「トランザクション」ダイアログボックスの「衝突解決」タブが表示されます。この「衝突解決」タブは、「アクション」▶「衝突解決」を選択して開くこともできます。

「ファイルマージ」ウィンドウの表示

ファイルマージツールを起動すると、「ファイルマージ」ウィンドウが表示されます(図 6-2 を参照)。この「ファイルマージ」ウィンドウには、上の部分に横に並んで 2 つ、下の部分に 1 つの区画があります。上の 2 つの区画は、同じファイルの異なる 2 つのバージョン、下の区画はマージ結果を表示します。上の 2 つの区画は読み取り専用です。下の区画には、ファイルの 2 つのバージョンの一方または両方から選択された行が含まれます。そして、この下の区画で編集し、最終的なマージしたバージョンを作成できます。

上の 2 つの区画に表示されるデルタは、その共通の祖先デルタとの比較した結果です。

- 「子」というラベルの付いた左側の区画は、子のデルタを示します。
- 「親」というラベルの付いた右側の区画は、親のデルタを示します。

親と子に表示された両方のデルタは、共通の祖先であるデルタから派生しています。この区画構成により、親および子デルタと祖先、親と子デルタ同士の 3 方向から比較を行うことができます。

ワークスペース内での衝突の解決

ワークスペース内の衝突を解決する手順は次のとおりです。

1. 「ワークスペース管理」ウィンドウから、衝突が存在するワークスペースのアイコンをダブルクリックします。

「トランザクション」ダイアログボックスの「衝突解決」タブが開き、そのファイルリスト区画に、衝突しているファイル名が表示されます。

2. 「衝突をマージ」をクリックします。

ファイルマージツールが起動し、ファイルリスト区画に示されているファイルの処理が開始されます。リスト内のすべてのファイルについて、SCCS 履歴ファイルから親デルタと子デルタ、共通祖先の情報が抽出され、そのパス名がファイルマージツールに渡されます。「ファイルマージ」ウィンドウが表示され、該当するファイルが読み込まれて、マージが可能になります。

3. ファイルマージツールを使用し、ファイルの親と子のバージョン間の相違を解決します。

ファイルマージツールの使用方法については、83 ページの「相違の解決」を参照してください。

4. 衝突を解決したらファイルを保存します。

ファイルマージツールを使用して、ファイルの親と子のバージョン間の相違を解決すると、子の SCCS 履歴ファイルに新しいデルタが作成されます。新しいデルタには、ファイルマージツールを使用して作成したマージ結果が含まれます。

5. この手順を繰り返してすべての衝突を解決します。

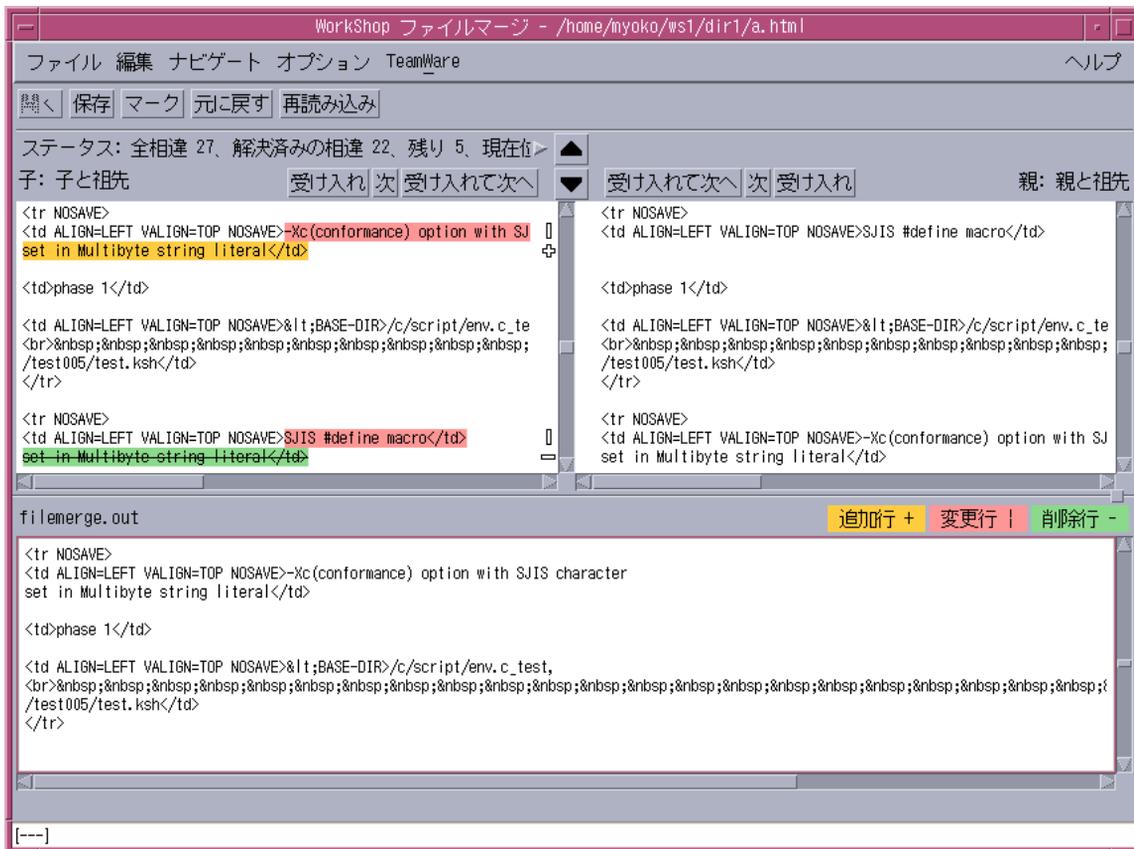


図 6-2 「ファイルマージ」ウィンドウ

「ファイルマージ」ウィンドウのグリフについて

相違が簡単に見分けられるよう、ファイルマージでは、いくつかの色とグリフを使用して、相違する行を強調表示します。黄色は追加されている行、赤色は変更されている行、緑色は削除されている行を示します。

「ファイルマージ」ウィンドウで使用されるグリフの意味は、2つのバージョンの比較か（入力ファイル2つ）、2つのファイルのバージョンに対して共通の祖先を指定したか（入力ファイル3つ）で異なります。図6-2は、2つのファイル間の相違を示すグリフが表示されている「ファイルマージ」ウィンドウを示しています。

入力ファイルが 2 つの場合

ファイルマージに 2 つのファイルだけが読み込まれ、それらのファイルの対応する行に相違がある場合は、それぞれのファイルの該当する行にグリフが表示されます。

- 2 つの行に相違がない場合、グリフは表示されません。
- 2 つの行に相違がある場合は、両方のテキスト区画のその行の先頭に縦棒 (|) が表示され、相違部分が黄色で強調表示されます。
- ある行が一方のファイルにあって、もう一方にない場合は、その行が存在する方のファイルの該当する行の先頭に正符号 (+) が表示され、相違部分が赤色で強調表示されます。
- 相違が解決された行には、白抜きのグリフが表示されます。

入力ファイルが 3 つの場合

マージする 2 つのファイルを読み込むときに、3 つ目のファイルとして、それら 2 つのファイルの祖先ファイルを指定できます。祖先ファイルは、マージする 2 つのファイルより古いバージョンのファイルです。祖先ファイルを指定すると、そのファイルが、マージする 2 つのファイルと比較する基準として使用されます。また、祖先ファイルを指定することによって、自動マージを行うことができます。ファイルマージは、祖先ファイルから派生したファイル、すなわち、子孫ファイル内の、祖先と異なるすべての行にマークを付け、それら 3 つのファイルに基づいてマージファイルを作成します。

マージする 2 つのファイル内の、祖先ファイルと異なる行は、変更バーのマークが付けられ、いくつかの色を使用して強調表示されます。

- 3 つのファイルの間で行に相違が存在しない場合、グリフは表示されません。
- 祖先には存在しない行が、子孫のいずれかまたは両方に存在する場合は、その行が存在するファイルの該当する行の先頭に正符号 (+) が表示され、相違部分が黄色で強調表示されます。
- 祖先に存在する行が、子孫のいずれかまたは両方の対応する行と異なる場合は、その異なる行が存在するファイルの該当する行の先頭に縦棒 (|) が表示され、相違部分が赤色で強調表示されます。
- 祖先に存在する行が、子孫のいずれかまたは両方に存在しない場合は、その行が存在しないファイルの該当する行位置の先頭に負符号 (-) が表示され、相違部分が抹消線付きの緑色で強調表示されます。

- 相違が解決された行には、白抜きのグリフが表示されます。

ファイルマージツールへのファイルの読み込み

衝突解決によってファイルマージツールにファイルが自動的に読み込まれなかった場合は、「ファイル」▶「開く」を選択するか、「開く」ボタンをクリックすることによって、手動でファイルを読み込むことができます。図 6-3 は、「ファイルを開く」ダイアログボックスを示しています。



図 6-3 ファイルマージツールにおける「ファイルを開く」ダイアログボックス

「ファイルを開く」ダイアログボックスには、表 6-1 に示すフィールドがあります。

表 6-1 ファイルマージツールの「ファイルを開く」ダイアログボックスのフィールド

テキストボックス	説明
ディレクトリ	Sun WorkShop またはコマンド行から引数を付けずにファイルマージツールを起動すると、現在の作業用ディレクトリが表示されます。このフィールドは編集可能です。ファイルマージツールは、このウィンドウ内のファイル名を現在の作業用ディレクトリを基準にした相対パス名として解釈します。このため、サブディレクトリ内のファイルは <i>subdir/filename</i> の形式、親ディレクトリ内のファイルは <i>../filename</i> の形式で指定できます。スラッシュ (/) で始まるファイル名は、現在の作業用ディレクトリを基準にした相対パス名ではなく、フルパス名として解釈されます。
左のファイル	左側のテキスト区画 (子区画) に表示するファイルを指定します。
右のファイル	右側のテキスト区画 (親区画) に表示するファイルを指定します。
祖先ファイル	祖先ファイル名を指定します。このテキストボックスにファイル名を入力した場合、ファイルマージツールは、マージされる 2 つのファイルとそのファイルを比較し、2 つのファイル内の、祖先と内容が異なる行を特定します。自動マージは祖先ファイルに基づいて実行されますが、祖先ファイルそのものは「ファイルマージ」ウィンドウに表示されません。祖先ファイル名が省略された場合は、左と右のファイルだけが比較され、その 2 つのファイルからマージ版を作成します。祖先ファイルなしで、自動マージを行うことはできません。
出力ファイル	マージ結果の出力ファイルの名前を指定します。省略した場合は、 <code>filemerge.out</code> という名前で、現在の作業用ディレクトリに出力ファイルを書き込みます。

マージするファイルが読み込まれた「ファイルマージ」ウィンドウの各テキスト区画の上部には、それぞれ適切なファイル名 (左のファイル名、右のファイル名、出力ファイル名) が表示されます。祖先ファイルを指定した場合は、ウィンドウのヘッダー部分にそのファイル名が表示されます。

相違の解決

相違の解決では、元のデルタ (親または子) のいずれか一方の行をそのまま採用すること、マージ版を手作業で編集することもできます。適切な変更が行われたことを、コントロールパネルのボタンをクリックすることによって示すと、現在注目している相違点は「解決された」こととなります。相違点が解決されると、相違箇所のグリフは白抜きグリフに変わります。ファイルマージツールは、自動的に次の相違箇所に進むか (「自動的に進む」属性がオンの場合)、指定された相違箇所に進みます。

下部のテキスト区画では、相違箇所が空白行で示されます。相違を解決するには、次のいずれかの操作を行います。

- 採用するバージョンのテキスト区画上部の「受け入れ」または「受け入れて次へ」をクリックして表示されている行を採用し、マージ版に反映する。
- マージ版を手作業で編集し、「編集」▶「現在のマークを解決済みに変更」を選択することによって相違箇所に解決済みのマークを付ける。

衝突状態にない相違行 (すなわち、一方のファイルでだけ変更されている行) を自動的にマージ版に取り込むように設定する場合は、「オプション」▶「自動マージ」を選択します。そうすると、ファイルマージツールの起動時に、解決されているすべての行が自動的にマージ版に取り込まれます。詳細は、84 ページの「自動マージの使用方法」を参照してください。

ファイル間の相違を解決する手順は次のとおりです。

1. 処理する相違箇所を決めます。

- 「ファイルマージ」ウィンドウの左上のステータス行を参照します。
- 強調表示されているグリフを確認します。

強調表示されている相違箇所を「現在の相違」といいます。同じファイル中の直後の相違箇所は「次の相違」、同じファイル中の直前の相違は「前の相違」です。

2. 適切な行を選択して、受け入れます。

- 子のバージョンの行を採用する場合は、左のテキスト区画の上にある「受け入れ」ボタンをクリックします。

- 親のバージョンの行を採用する場合は、右のテキスト区画の上にある「受け入れ」ボタンをクリックします。

採用した行が、下部の区画 (マージ版) に表示されます。

3. 「次」ボタンをクリックして、ファイルの次の相違箇所へ移動します。

別の相違箇所へ移動するには、次のいずれかの操作をします。

- 「次」ボタンをクリックする。
- 「ナビゲート」メニューを利用する
- 相違ナビゲータのボタン (左右の区画の境界の上にある上向きおよび下向き矢印ボタン) を利用する。

加えた変更を放棄して最初からやり直す場合は「再読み込み」ボタンをクリックします。相違の解決がすべて放棄され、相違を解決する前のファイルが読み込まれます。

4. 相違をすべて解決したら、「保存」をクリックします。

自動マージの使用方法

共通の祖先ファイルを読み込むことによって、自動的に相違を解決できることがあります。自動マージは、次の規則に基づいて相違を解決します。

- どちらの子孫でも行が変更されていない、すなわち、3つのファイル間で行の内容が同じ場合は、その行をマージ版に取り込む。
- いずれか一方の子孫でだけ行が変更されている場合は、変更されている方の行をマージ版に取り込む。変更は、行全体の追加または削除のこともあれば、一部変更のこともあります。
- 行に対する変更内容が両方の子孫でまったく同じ場合は、その行をマージ版に取り込む。
- 両方の子孫で変更内容が異なる、すなわち、3つのファイルの間で行の内容が異なる場合は、マージ版には行は取り込まれません。この場合は、右または左のファイルのいずれかの行を採用するか、マージ版を手作業で編集して相違を解決する必要があります。
- 相違が解決された行には、白抜きのグリフを表示する。

自動マージ規則のまとめ

表 6-2 に、自動マージのアルゴリズムを示します。表で使用されている用語は、それぞれ次のことを意味します。

- 祖先は、祖先ファイル内のテキスト行を意味します。
- 変更 1 は、子孫の一方のファイル内の、変更が加えられた行を意味します。
- 変更 2 は、変更 1 と異なる別の変更を意味します。
- 左右の子孫で行の変更内容が異なる場合、それらの行はいずれもマージ版に取り込まれません。

表 6-2 自動マージ規則のまとめ

左の子孫	右の子孫	自動マージ結果
祖先	祖先	祖先
変更 1	祖先	変更 1
祖先	変更 2	変更 2
変更 1	変更 1	変更 1
変更 1	変更 2	自動マージされない

変更の取り消し

「編集」▶「元に戻す」を使用して変更を取り消すことができます。

「ファイルマージ」ウィンドウの「再読み込み」ボタンを使用して相違を解決するためのすべての編集内容を放棄し、ディスクからファイルを読み込むこともできます。「自動マージ」オプションが選択されている場合は、衝突状態にない相違はすべて下部の区画に表示されます。

ファイルマージツールのオプション

次の 2 つの場所で相違解決とファイルマージツールの動作を変更できます。

- 「ワークスペース管理」ウィンドウの相違解決オプション
- 「ファイルマージ」ウィンドウの表示オプション

相違解決オプション

「ツール属性」ダイアログボックスの「衝突解決」タブ (図 6-4) で、「トランザクション」ウィンドウの「衝突解決」区画の動作を変更できます。「衝突解決」タブには、表 6-3 に示す属性があります。

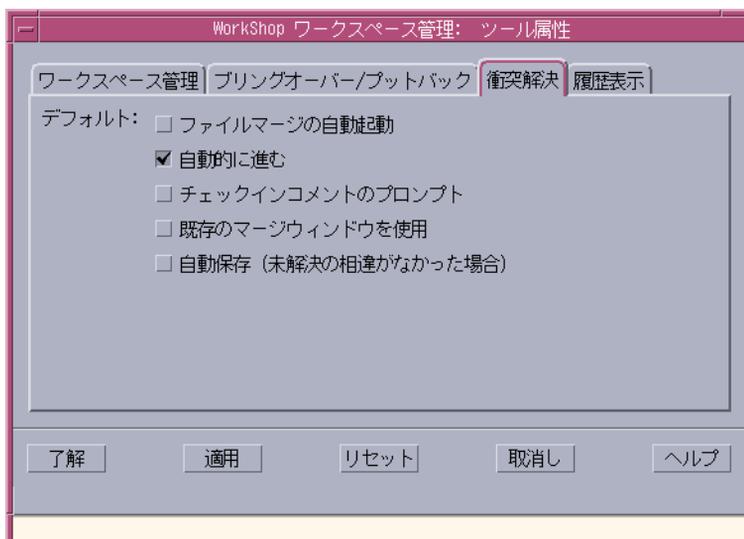


図 6-4 「ツール属性」ダイアログボックスの「衝突解決」タブ

表 6-3 相違解決ツール属性

属性	説明
ファイルマージの自動起動	相違解決のトランザクション区画が選択されたとき自動的にファイルマージツールを起動します。
自動的に進む	現在のファイルの相違が解決された後、リストの次のファイルを自動的にファイルマージツールに読み込みます。
チェックインコメントのプロンプト	ファイルの相違を解決した後のチェックインでは自動的にデフォルトのコメントが追加されます。この属性を有効にするとデフォルトのコメントに追記する追加コメントの入力が求められます。
既存のマージウィンドウを使用	この属性が有効な場合は、以降に続く衝突解決作業に実行中のファイルマージツールを再利用します。
自動保存 (未解決の相違がなかった場合)	この属性が有効でファイル内のすべての変更が自動マージ可能な場合は、ファイルが自動的に保存され、チェックインされます。

「ファイルマージ」ウィンドウの表示オプション

「ファイルマージ」ウィンドウの「オプション」メニューには、次の表に示すオプションが含まれています。このメニューの最初の5つのオプションはトグル方式で、選択するたびに有効 / 無効が切り替わります。有効にすると、オプションの左側に小さなボックスが表示されます。

表 6-4 は、「ファイルマージ」ウィンドウの表示オプションをまとめています。

表 6-4 「ファイルマージ」ウィンドウの表示オプション

オプション	説明
自動マージ	衝突状態にない相違を自動的に解決し、マージ結果区画にマージ版を作成します。デフォルトは有効です。
スクロール — 連動して	テキスト区画のスクロールの同期を取るかどうかを設定します。スクロールの同期を取ると、各区画に対応する行が並んで表示されます。デフォルトは有効です。
行番号表示	左右の区画内に行番号を表示します。
行末表示	左右の区画内の各行の末尾に小さな黒いボックスを表示します。
相違ナビゲータ表示	左右の区画の間に相違ナビゲータを表示します。相違ナビゲータは、2つのファイル間の相違箇所を色付きの線で表示します。相違ナビゲータのいずれかの側のスライドボックスをクリックすると、そのボックスがある方のファイルがスクロールします。上向きまたは下向きの矢印をクリックすると、両方のファイル内を同じ距離、移動します。
タブ表示	タブの設定をカスタマイズできます。次の2つのオプションがあります。 制御文字 (^I) 各タブ位置に^I制御文字を表示します。デフォルトはオンです。 スペース タブの空白文字数を1、2、3、4、6、8、10、12、16のいずれかの文字数に設定できます。デフォルトは8文字です。
相違オプション	相違の動作をカスタマイズできます。次の3つのオプションがあります。

表 6-4 「ファイルマージ」ウィンドウの表示オプション (続き)

オプション	説明
行末のスペースを無視	相違行の検出で行の末尾の空白を無視します。デフォルトは無効です。
すべてのスペースを無視	相違行の検出で行の途中および末尾の空白を無視します。デフォルトは無効です。
大文字小文字を区別しない	相違行の検出で英大文字と小文字を同一視します。デフォルトは無効です。

ファイルのマージ例

ここでは、共通の祖先を持つ 2 つのファイルのマージ例を紹介します。子孫ファイルの名前は `file_1` と `file_2`、祖先ファイルの名前は `matriarch` です。`matriarch` を編集することによって、`file_1` と `file_2` が作成されています。子孫ファイルでは、行の削除や追加、行の一部変更などの、さまざまな変更が行われています。

下記のマージ例の各行を見ると、変更が加えられているかどうか、また変更が加えられている場合は、どのように変更されているかが分かります。コード例 6-1 に示すように、祖先ファイルはわずか 12 行の大きさです。

デフォルトでは、読み込んだファイルに行番号は表示されませんが、分かりやすくするために、この例では行番号を表示しています。

コード例 6-1 祖先ファイル (`matriarch`)

```

1 This line is deleted in file_1
2 This line is in all three files
3 This line is deleted in file_2
4 This line is in all three files
5 This line is in all three files
6 This line is changed in descendants
7 This line is in all three files
8 This line is changed in descendants
9 This line is in all three files
10 This line is changed in file_2
11 This line is in all three files
12 This line is in all three files

```

コード例 6-2 は、`file_1` の内容を示しています。このファイルと祖先ファイル `matriarch` との相違は次のとおりです。

- `matriarch` ファイルの行番号 1 の行が削除されています。
- 行番号 4 の行の後に新しい行が追加されています。
- 行番号 6 の行が変更されています (`file_2` では、これとは異なる別の変更が加えられています)。
- 行番号 8 の行が変更されています (`file_2` でも同じ変更が加えられています)。

コード例 6-2 子孫ファイル (`file_1`)

```
2 This line is in all three files
3 This line is deleted in file_2
4 This line is in all three files
  &&& Added to file_1 &&&
6 This line is modified in file_1 from matriarch
5 This line is in all three files
7 This line is in all three files
8 #&&# Changed in file_1 and file_2 #&&#
9 This line is in all three files
10 This line is changed in file_2
11 This line is in all three files
12 This line is in all three files
```

コード例 6-3 は、`file_2` の内容を示しています。このファイルと祖先ファイル `matriarch` との相違は次のとおりです。

- `matriarch` ファイルの行番号 3 の行が削除されています。
- 行番号 6 の行が変更されています (`file_1` では、これとは異なる別の変更が加えられています)。
- 行番号 8 の行が変更されています (`file_1` でも同じ変更が加えられています)。
- 行番号 10 の行が変更されています (`file_1` では変更されていません)。
- 行番号 11 の行の後に新しい行が追加されています。

コード例 6-3 子孫ファイル ([file_2](#))

```
1 This line is deleted in file_1
2 This line is in all three files
4 This line is in all three files
5 This line is in all three files
6 This line is altered in file_2 from matriarch
7 This line is in all three files
8 ## Changed in file_1 and file_2 ##
9 This line is in all three files
10 ### Changed in file_2 ###
11 This line is in all three files
    ### Added to file_2 ###
12 This line is in all three files
```

これらのファイルを自動マージすると、「ファイルマージ」ウィンドウの左上にマージ結果が表示されます。相違は7つで、そのうちの6つが解決され、1つが未解決になります (図 6-5 を参照)。解決された相違には、白抜きのグリフが表示されます (図 6-6 と 図 6-7 を参照)。

ステータス: 全相違 7、解決済みの相違 6、残り 1、現在位置 4

図 6-5 自動マージ後の [file_1](#) と [file_2](#) のマージ結果

```
子: file_1  受け入れ 次 受け入れて次へ
1 This line is deleted in file_1
2 This line is in all three files
3 This line is deleted in file_2
4 This line is in all three files
  &&& Added to file_1 &&&
5 This line is in all three files
6 This line is modified in file_1 from matriarch
7 This line is in all three files
8 ##&# Changed in file_1 and file_2 ##&#
9 This line is in all three files
10 This line is changed in file_2
11 This line is in all three files

12 This line is in all three files
```

図 6-6 自動マージ後の子区画の file_1 の内容

```
受け入れて次へ 次 受け入れ 親: file_2
1 This line is deleted in file_1
2 This line is in all three files
3 This line is deleted in file_2
4 This line is in all three files

5 This line is in all three files
6 This line is altered in file_2 from matriarch
7 This line is in all three files
8 ##&# Changed in file_1 and file_2 ##&#
9 This line is in all three files
10 ##&# Changed in file_2 ##&#
11 This line is in all three files
12 This line is in all three files
  &&& Added to file_2 &&&
```

図 6-7 自動マージ後の親区画の file_2 の内容

各グリフの意味は次のとおりです。縦棒 (|) は、その行に変更があったことを意味します。正符号 (+) はその行が追加されていること、負符号 (-) はその行が削除されていることを意味します。未解決の行は塗りつぶしのグリフ、解決済みの行は白抜きグリフで示されます。カラーマップが使い切られていない限り、これらのグリフは通常とは異なる色で強調表示されます。デフォルトでは、赤色の変更、緑色が削除、黄色が追加を意味します。

未解決の相違 (上記の例では行番号 6 の行) には、縦棒のマークが付きます。

相違を確認する

ファイルマージツールでは、未解決の相違 (上記の例の `file_1` と `file_2` の行番号 6 の行) が強調表示されます。自動マージで相違を解決すると、その間のファイルの最新状態が「マージ後の結果」区画に表示されます。

次の相違に進むには、その操作をする区画の上の下矢印ボタンをクリックするか、「ナビゲート」▶「次」▶「相違」を選択します。次の相違が現在の相違になります。

下矢印ボタンを使用することによって、相違を順にたどることができます。

自動マージでは、一方のファイルだけで加えられた変更を残します。自動マージによって相違が解決されなかった場合は (相違に関係する行には、通常とは異なる色の塗りつぶされたグリフが表示されます)、手作業でその相違を解決する必要があります。縦棒は、その行が追加や削除をされたものではなく、変更されていることを示します。2 つのファイルの同じ行に変更が加えられていて、その変更内容が異なる場合、自動マージはその行をマージ版に取り込みません。そうした行は、どちらの行を採用するかをユーザー自身で決定する必要があります。

相違を解決する

上記のマージ例の相違は、次のいずれかの方法で解決できます。

- `file_1` の行をマージ版に採用する場合は、左の区画の「受け入れ」または「受け入れて次へ」ボタンをクリックします。
- `file_2` の行をマージ版に採用する場合は、右の区画の「受け入れ」または「受け入れて次へ」ボタンをクリックします。
- 手作業でマージ版を編集します。

出力ファイルを編集する

出力ファイルを編集する手順は次のとおりです。

1. マージ版のある下部のテキスト区画内の、編集する行位置にポインタを置きます。
この例では、次の行を入力すると仮定します。

```
>>> This line edited by hand <<<
```

2. 「編集」▶「現在のマークを解決済に変更」を選択します。

このメニュー項目は、相違行に解決済みのマークを付けます。これで、相違はすべて解決されたこととなります。

「ファイルマージ」ウィンドウの左上のステータス行に、すべての相違が解決されたことが示されます。

3. 自動マージによる相違の解決が適切であるかどうかを確認します。

下矢印ボタンを使用することによって、相違を順にたどることができます。

最後の相違は、`file_2` にだけ追加されている 1 行です。ファイルマージツールは、`file_1` に追加されていた 1 行 (3 番目の相違) について行われたのと同様に、この行をマージ版に取りこんでいます。

第7章

高度なファイル管理

この章では、以下のような高度なファイル管理機能について説明します。

- ワークスペース内のファイルの更新
- ファイルの履歴の表示
- ファイル名の変更とファイルの移動、削除
- カスタマイズメニューの作成
- 「読み込み」メニューへのパスの追加
- バージョン管理ツールの属性の設定
- SCCS ファイルの属性の設定

ワークスペース内のファイルの更新

ワークスペースの作成後は、プリングオーバートランザクションを使用して定期的にワークスペースを更新する必要があります。

ワークスペースを更新する手順は次のとおりです。

1. 「ワークスペース管理」ウィンドウから「アクション」▶「更新プリングオーバー」を選択します。
2. 「トランザクション」ダイアログボックスの「更新プリングオーバー」タブをクリックします。
3. 親および子ワークスペースのディレクトリ名を確認します。
4. 更新するディレクトリとファイルを選択します。
5. 「プリングオーバー」をクリックします。

更新プリングオーバーランザクションについての詳細は、20 ページの「子ワークスペースの更新 (更新プリングオーバー)」を参照してください。

ファイルの履歴の表示

バージョン管理ツールでは、Sun WorkShop TeamWare ワークスペース内のファイルの履歴をグラフィカルに表示できます。ファイルを変更したユーザーや変更した日時を確認したり、変更理由などを記したコメントが追加されている場合に、そのコメントを表示したりできます。

「履歴」ウィンドウの履歴グラフ区画には、選択されたファイルのデルタの分岐がグラフィカルに表示されます (図 7-1 を参照)。この履歴グラフによって、ファイルのデルタ構造とバージョン間の関係を知ることができます。履歴グラフ区画において、デフォルトで表示される破線は、その右側のデルタに左側のデルタの変更が反映されていることを示します。破線をたどることによって、時系列にデルタをたどることができます。

「履歴」ウィンドウでは、次のことを行うことができます。

- 履歴グラフ区画からデルタを選択して、デルタ情報区画にそのデルタに関する情報を表示する。
- 履歴グラフ区画からデルタを選択し、その現在の SCCS 状態に応じてチェックインまたはチェックアウトする。
- 「表示」▶「ファイルの中身を表示」を選択して選択したデルタの内容を表示する。エディタのウィンドウが開いて、選択したデルタの内容が表示されます。
- デルタを 2 つ選択して、「表示」▶「差分をファイルマージで」を選択する。「ファイルマージ」ウィンドウに選択した 2 つのデルタが横に並べて表示され、内容を比較できます。
- デルタを 2 つ選択して、「表示」▶「差分をテキストエディタで」を選択する。エディタのウィンドウが開き、SCCS の `diffs` コマンドによって検出された相違点が表示されます。
- デルタを 2 つ選択して、「表示」▶「コンテキストの相違」を選択する。エディタのウィンドウが開き、SCCS の `diffs -c` コマンドによって検出された相違点が表示されます。

ファイルの履歴を表示する手順は次のとおりです。

1. 「バージョン管理」ウィンドウからファイルをクリックして選択状態にします。

2. 「ファイル」▶「ファイル履歴」を選択します。
「ファイル履歴」ウィンドウが表示されます (図 7-1 を参照)。
3. 同じファイルの別のデルタを表示する場合は、履歴グラフ区画からそのデルタ (1.2 などの番号で表示) をクリックします。

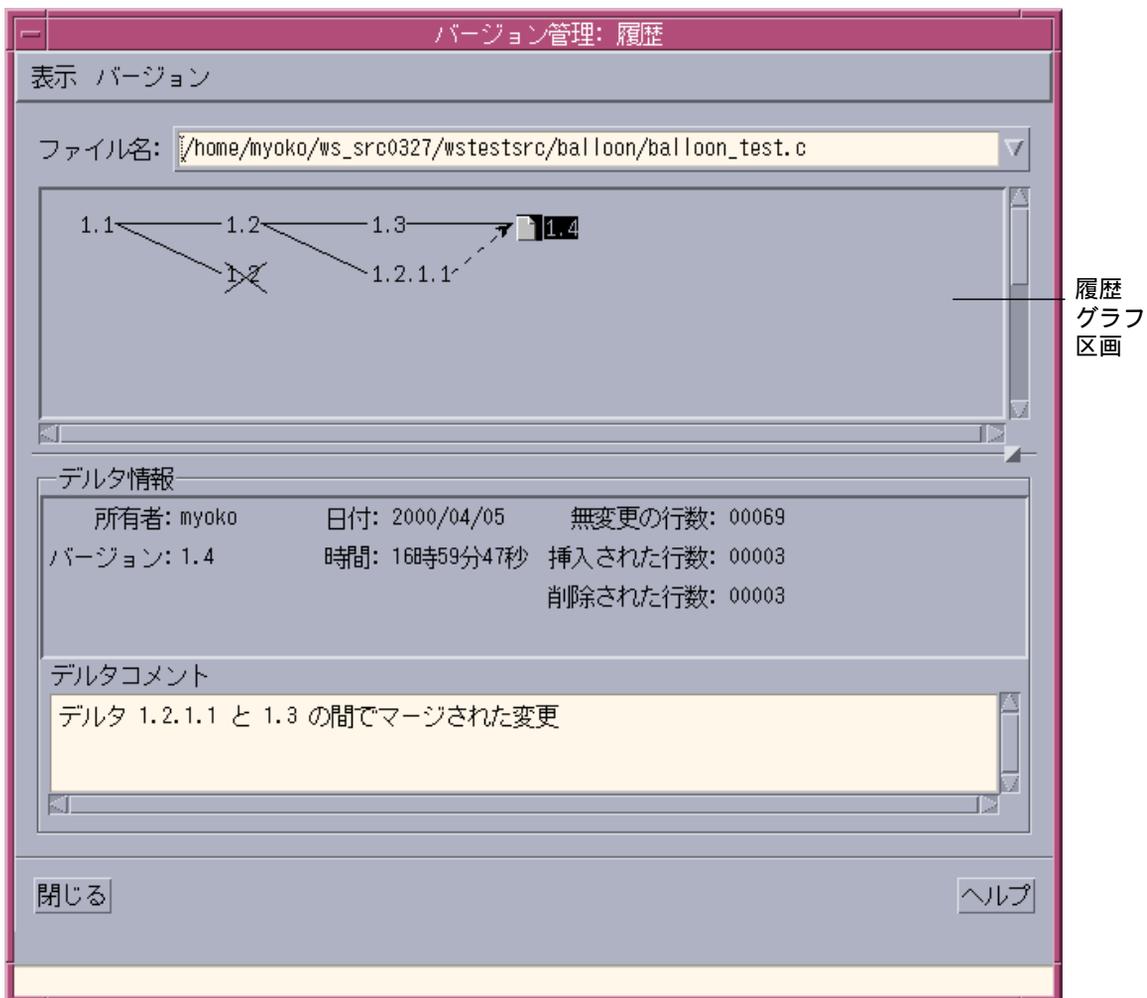


図 7-1 「ファイル履歴」ウィンドウ

「ファイル履歴」ウィンドウ

「ファイル履歴」ウィンドウは、表 7-1 に示す部分で構成されています。

表 7-1 「ファイル履歴」ウィンドウの各部

メニュー	説明
「表示」メニュー	ファイルを管理するためのコマンドがあります。
「バージョン」メニュー	履歴グラフ区画のアイコンを管理するコマンドがあります。
「ファイル名」フィールド	ファイルのパス名を表示します。
履歴グラフ区画	ファイルのデルタのアイコンを表示します。
デルタ情報区画	デルタの履歴情報を表示します。

ファイル履歴表示で使用されるシンボル

「ファイル履歴表示」ウィンドウでは、次のシンボルが使用されます。

表 7-2 ファイル履歴表示で使用されるシンボル

シンボル	説明
	バージョン番号の左横にあるファイルアイコンは、デフォルトのデルタを示します。
	赤色のチェックマークは、そのファイルがチェックアウトされていることを示します。
	×が付いているデルタは、マージされていないことを示します。
	実線は、標準のデルタの流れを示します。
	矢印付きの波線は、分岐がマージ済みであることを示します。
	二重下線は、デフォルトのデルタを示します。デフォルトのデルタは、 <code>sccs-admin</code> コマンドのフラグ <code>d</code> で設定されます。110 ページの「SCCS ファイルの属性の設定」を参照してください。

ファイルの履歴 (デルタ、分岐、バージョン)

バージョン管理ツールは、ファイルが存在している間のさまざまなバージョンの記録を取ります。ファイルがチェックインされるたびに、チェックインされたファイルとその前のバージョンとの間の行ごとの相違を記録します。この行ごとの相違情報全体をデルタと呼びます。

Sun WorkShop TeamWare に初めてファイルをチェックインすると、デフォルトでは、そのファイルには 1.1 というデルタが割り当てられます。これ以降のデルタには、1.2、1.3 という番号が割り当てられます。このようなデルタの縦構造が、SCCS ツリーのトランク (幹) を形成します。しかし、実際のソフトウェアの開発では、トランクからの枝分かれが必要になることがあります。これを、分岐と呼びます。分岐することによって、たとえば、同じファイルに対して、異なる 2 つのバージョンを並行して開発できます。たとえば、ソースコードのバグ修正などを行うときがそうです。分岐には、その分岐が発生した位置に基づいて、1.2.1.1 のような番号が割り当てられます。

SCCS のデルタ ID (SID) は、特定のデルタを表すために使用される番号です。この番号は、ドット (.) で区切られた 2 つの部分で構成されます。デフォルトでは、初期デルタの SID は 1.1 です。SID の前の部分はリリース番号、後の部分はレベル番号と呼ばれます。デルタをチェックインすると、レベル番号が自動的に大きくなります。分岐デルタに対する SID は、リリース番号とレベル番号、分岐番号、シーケンス番号 (リリース . レベル . 分岐 . シーケンス) の 4 つの部分で構成されます。分岐番号は、幹の特定のデルタから分かれた分岐に割り当てられる番号で、最初の分岐が 1、次の分岐が 2 というようになります。シーケンス番号は、特定の分岐上の各デルタに順に割り当てられる番号です。たとえば 1.2.1.1 という SID は、デルタ 1.2 から派生した最初の分岐の最初のデルタを示します。デルタ 1.2 に 2 目目の分岐を追加すると、その分岐デルタに 1.2.2.1 という番号が割り当てられます。

分岐の概念は、ツリー上のあらゆるデルタに適用されます。分岐のコンポーネントは、トランクを基準にした位置とは関係なく、分岐上の作成順に番号が割り当てられます。このため、すべての分岐のデルタはその番号によって特定できます。トランクのデルタは分岐のデルタの番号から特定できますが、トランクのデルタから分岐のデルタまでの完全なパスを特定することはできません。

たとえば、デルタ 1.3 に 1 つの分岐がある場合、その分岐上のすべてのデルタには 1.3.1.n という番号が割り当てられます。この分岐のデルタから別の分岐が発生している場合、その新しい分岐のデルタには 1.3.2.n が割り当てられます。デルタ 1.3.2.2 の番号から推測できることは、そのデルタが、トランクであるデルタ 1.3 から発生した

2 目目の分岐の 2 目目のデルタであるということだけです。デルタ 1.3.2.2 という番号から基、そのデルタとトランクの祖先デルタ (1.3) との間のすべてのデルタを知ることにはできません。

デルタとバージョンという用語はしばしば同じ意味で使用されますが、実際には異なります。バージョン管理ツールは、集められたデルタの 1 つのまとめりからファイルの 1 つのバージョンを作成します。いくつかのデルタを除外したファイルのバージョンを取り出すこともできます。

デルタのマージ

開発を並行して進め、その時々に変更をマージしてトランクに戻すことが合理的である場合があります。分岐上のファイルをマージすると、その分岐は完結することになります。

2 つのデルタをマージする手順は次のとおりです。

1. バージョン管理ツールを起動し、作業用ディレクトリを選択します。
2. 「コマンド」▶「チェックアウト」を選択し、ファイルをチェックアウトします。
3. 「ファイル」▶「ファイル履歴」を選択し、ファイルの履歴を表示します。
4. 「履歴」ウィンドウの履歴グラフ区画からデルタを 2 つを選択します。
5. 2 つのデルタ間の相違を調べるには、次のいずれかを選択します。
 - 「表示」▶「差分をファイルマージで」
 - 「表示」▶「差分をテキストエディタで」
 - 「表示」▶「コンテキストの相違」
6. ファイルを変更します。
7. 必要なコメントを追加します。
8. 「コマンド」▶「チェックイン」を選択して、ファイルをチェックインします。

ファイル名の変更とファイルの移動、削除

ファイル名を変更したり、ファイルを移動または削除した場合、ワークスペース管理ツールは、ブリングオーバーやプットバックトランザクションの処理中に、その変更を記録し、変更されたファイルを管理します。こうした処理はワークスペース管理ツールによって自動的に行われますが、ここでは、ワークスペース管理ツールの仕組みを理解するために、ファイル名の変更やファイルの移動、削除を行なったときに、ワークスペース管理ツールによってどのような処理が行われるのかを説明します。

注 - ワークスペースの名前変更や移動には、CDE (共通デスクトップ環境) のファイルマネージャや SunOS™ オペレーティングシステムの `mv` コマンドではなく、この節で紹介する手順を使用してください。この手順を使用することによって、ワークスペースの親子関係が保持されます。

ファイル名の変更と移動

名前を変更したファイルや、移動したファイルをブリングオーバーまたはプットバックするとき、ワークスペース管理ツールは、そのファイルが新たに作成されたものか、すでに存在していたファイルが名前変更または移動されたものかを判断する必要があります。名前変更または移動されたファイルの場合、ワークスペース管理ツールはそのファイルのファイル名と履歴ファイルの両方を更新します。ファイル内容更新と衝突で使用されたのと同じ規則に基づいて、ワークスペース階層全体にその変更を反映させます。

トランザクション中、ワークスペース管理ツールはファイルを別々に処理します。ディレクトリ名を変更したり、ディレクトリを移動したりした場合は、その中のファイルが別々に名前変更または移動されたかのように、各ファイルは1つずつ評価されます。

ファイル名の変更例

図 7-2 は、親ワークスペース内のファイルの名前を "C" から "D" に変更している例を示しています。このファイルを子ワークスペースにブリングオーバーするとき、ワークスペース管理ツールは、次のどちらの場合であるのかを判断する必要があります。

- "D" は親ワークスペースに新たに作成されたファイルである。

- "D" は子ワークスペース内の既存の "C" と同じファイルで、新しい名前が付けられただけである。

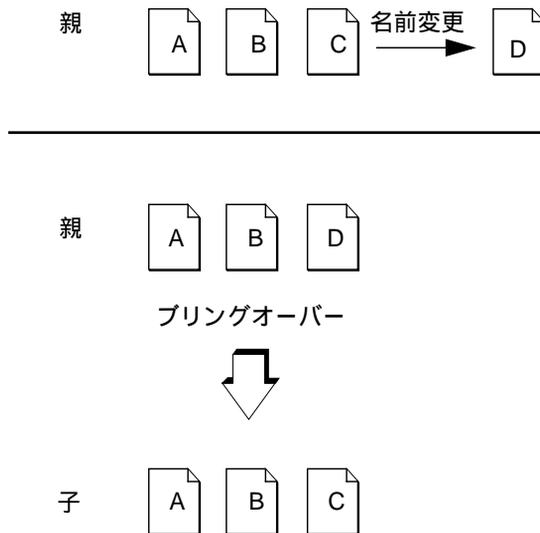


図 7-2 ファイル "C" の名前を "D" に変更

この判断は、ブットバックトランザクションでも同様に行われます。

ワークスペース管理ツールが行う処理は、ファイル "D" が新たに作成されたファイルなのか、名前が変更された既存のファイルなのかによって異なります。ファイル "D" が親ワークスペースの新規ファイルの場合は、子ワークスペースにそのファイルを作成します。ファイル "D" が、名前が変更されただけの既存ファイルの場合は、子ワークスペース内のファイル "C" の名前を "D" に変更します。

ワークスペース管理ツールは、名前が変更された場合でもファイルを特定できるよう、SCCS 履歴ファイルに変更情報を記録します。下記は、ブリングオーバーまたはブットバックの出力を表示するときに表示されるメッセージです。

検査したファイル数:

更新ブリングオーバーまたはブットバックトランザクションを処理するとき、ワークスペース管理ツールは、そのトランザクションに関係するすべてのファイルについて、名前が変更されていないかどうかを調べてから、ファイルの転送を開始します。

名前が変更されているファイルを検出した場合、プリングオーバーでは子ワークスペース、プットバックでは親ワークスペースにその名前変更を反映します。また、「トランザクション出力」ウィンドウには、次のようなメッセージによって名前の変更が通知されます。

名前変更	旧: <i>old_filename</i> 新: <i>new_filename</i>
------	--

名前の履歴

ワークスペース管理ツールは、その SCCS 履歴ファイルにファイル名の履歴情報を記録します。この名前の履歴は、ファイルが作成されてから、そのファイルに付けられた名前のリストであり、ワークスペースを基準にした相対ファイル名の形式になっています。ワークスペースは、この情報を利用して、名前が変更されたファイルか新規のファイルかを判断します。ファイル名が変更された場合、そのファイルを処理対象とする次のプリングオーバーまたはプットバックトランザクションで、そのファイルの名前履歴を更新します。名前履歴が更新されると、「トランザクション出力」ウィンドウにその情報が表示されます。

名前に関する要約:
1 親の名前履歴を更新
1 子の名前履歴を更新

名前変更の衝突

ごくまれに、親ワークスペースと子ワークスペースで同じファイルが同時に名前変更されることがあります (名前変更の衝突)。たとえば、図 7-3 の例では、親ワークスペースのファイル "C" の名前が "D" に変更されたのと同時に、子ワークスペースでファイル "C" のが "F" に名前変更されています。

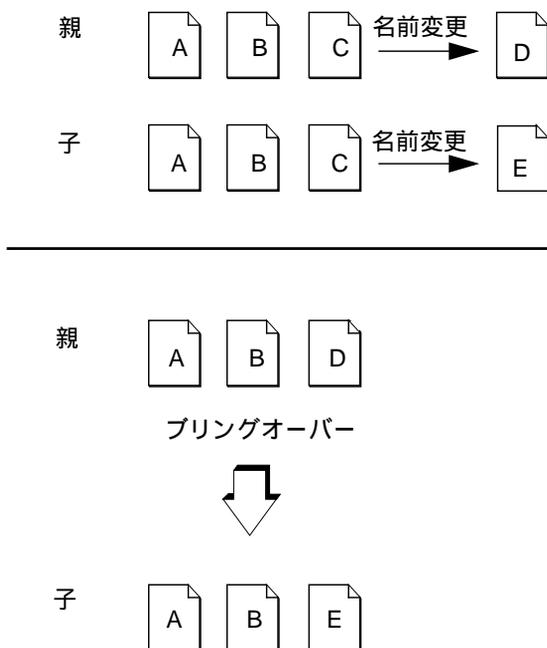


図 7-3 親と子ワークスペースにおけるファイル "C" の名前変更の衝突例

この場合、ワークスペース管理ツールは、親ワークスペースのファイル "D" と子ワークスペースのファイル "E" が同じファイルで、名前が違うだけであることを検出します。名前変更の衝突がある場合、ワークスペース管理ツールは次のことを行います。

- 子ワークスペースのファイル名を使用して衝突を報告します。
- 子ワークスペースのファイル名を親ワークスペースのファイル名に自動的に変更することによって衝突を解決します。プットバックトランザクションであっても、常に親ワークスペースのファイル名が使用されます。

また、ワークスペース管理ツールは、「トランザクション出力」ウィンドウに次のメッセージを表示して、このことを通知します。

名前の衝突です: *name_in_child*
 名前変更 旧: *name_in_child*
 新: *name_in_parent*

ファイルの削除

ワークスペースからファイルが削除されたときにワークスペース管理ツールが行う処理は多少複雑です。Sun WorkShop TeamWare のコマンドを使用しないでワークスペースからファイルが削除された場合、ワークスペース管理ツールは、そのファイルを、ワークスペースの親また子で新たに作成されたファイルとみなします。

たとえば、図 7-4 の例では、Solaris オペレーティング環境の `rm` コマンドを使用して、子ワークスペースからファイル "C" を削除した後で、更新ブリングオーバーアクションでその子ワークスペースを更新しています。

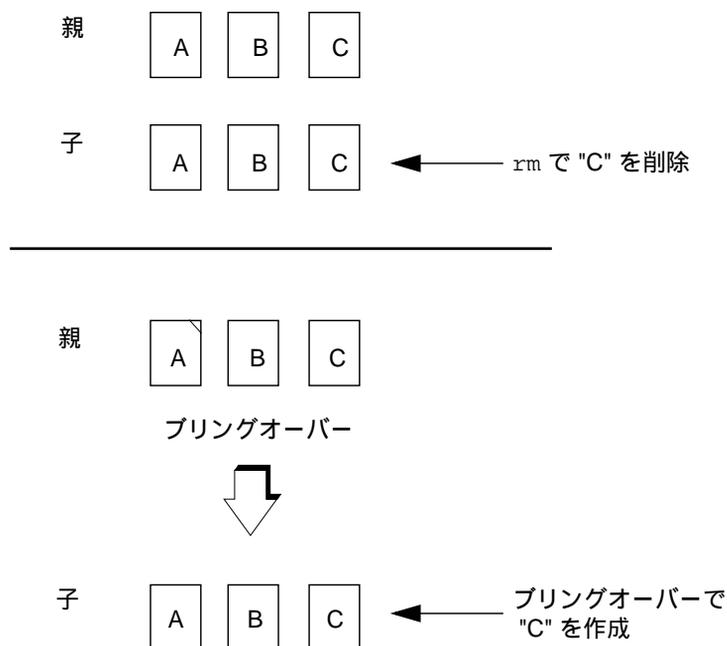


図 7-4 `rm` コマンドで子から "C" を削除し、ブリングオーバーで再作成

ワークスペース管理ツールは 2 つのワークスペースを調べ、ファイル "C" が親ワークスペースには存在していて、子ワークスペースには存在しないと判断します。そして、通常のワークスペース管理ツールの規則に従って、子ワークスペースにファイル "C" を作成します。

ファイルおよびワークスペースを削除する場合は、必ず Sun WorkShop TeamWare のコマンドを使用してください。

Sun WorkShop TeamWare ファイルの削除

ファイルを削除する場合は、オペレーティングシステムのコマンドではなく、以下に説明する手順を使用してください。この手順を使用することによって、ファイルの関係と履歴が保持されます。

Sun WorkShop TeamWare ファイルを削除する手順は次のとおりです。

1. バージョン管理ツールを起動します。
2. ファイルをクリックして選択します。
3. 「ワークスペース」▶「ファイル削除」を選択します。
4. 「了解」をクリックします。

バージョン管理ツールは、実際にはファイルを削除しないで、`deleted_files` ディレクトリにファイルを移動します。この変更は、ワークスペース階層全体に名前変更として伝えられ、すべてのワークスペースから該当するファイルが削除されます。

この手順を使用してファイルを削除した後でファイルが再び表われるという心配はありません。詳細は、105 ページの「ファイルの削除」を参照してください。

カスタマイズメニューの作成

バージョン管理ツールでは、他のプログラムやよく使うコマンドにアクセスするための独自のプルダウンメニュー (カスタマイズメニュー) を作成できます。

カスタマイズメニューを作成するには、次の手順に従います。

1. 「ワークスペース管理」ウィンドウから「TeamWare」▶「バージョン管理」を選択して、「バージョン管理」ウィンドウを開きます。
2. 「バージョン管理」ウィンドウで「表示」▶「オプション」を選択します。
3. 「カスタマイズメニュー」タブをクリックします。
4. 「コマンド」フィールドにコマンドのフルパス名を入力します。

例: `/bin/grep`

Sun WorkShop TeamWare には、コマンドの引数に使用できる、ARG および FILE という 2 つのマクロが用意されています。

```
/bin/grep $ARGS $FILES
```

5. 「メニューラベル」フィールドに、メニューに表示する項目名を入力します。
6. コマンドの出力を表示する場合は、「出力ウィンドウ」のチェックボックスを選択します。
7. ユーザーに入力を求めるメッセージを表示する場合は、「プロンプトウィンドウ」のチェックボックスを選択します。
8. 「プロンプト」フィールドに、プロンプトウィンドウに表示するテキストを入力します。
9. 「追加」をクリックし、定義したメニュー項目をリストに追加します。
10. 「テスト」をクリックし、定義したコマンドをテストします。
11. 問題がなければ、「了解」をクリックします。

「バージョン管理」ウィンドウに、「Customized」という見出しのメニューが表示されます。追加したコマンドがメニューに含まれていることを確認してください。

「読み込み」メニューへのパスの追加

プロジェクトによっては、階層構造がかなり複雑になることがあります。そうした場合、定期的に作業をするファイルにアクセスするために、何回もクリックして、いくつかのディレクトリをたどったり、長いパス名を入力したりする必要があります。ここで説明する、「読み込み」メニューにパスを追加する方法で、この作業を簡単にすることができます。

「読み込み」メニューにパスを追加する手順は次のとおりです。

1. 「ワークスペース管理」ウィンドウから「TeamWare」▶「バージョン管理」を選択し、「バージョン管理」ウィンドウを開きます。
2. 「バージョン管理」ウィンドウから「表示」▶「オプション」を選択し、「読み込みメニューのデフォルト」タブをクリックします。

3. 「パス名」フィールドにディレクトリのフルパス名を入力します。
例: `/set/pubs/Work/Workspaces/TAZ/IntA/sig_team`
4. 「メニューラベル」フィールドに「読み込み」メニューに表示する名前を入力します。
例: `sigteam`
5. 「追加」をクリックします。
「読み込みメニューのデフォルト」タブに指定した名前が表示されます。
6. 「了解」をクリックします。
「ファイル」▶「読み込み」メニューに、指定した名前のメニュー項目が表示され
ます。この項目を選択すると、バージョン管理ツールによって、定義したディレクトリ
の内容が読み込まれます。

バージョン管理ツールの属性の設定

「オプション」ダイアログボックスを使用して、バージョン管理ツールの属性を設定
できます (図 7-5 を参照)。「オプション」ダイアログボックスを開くには、「表示」
▶「オプション」を選択します。

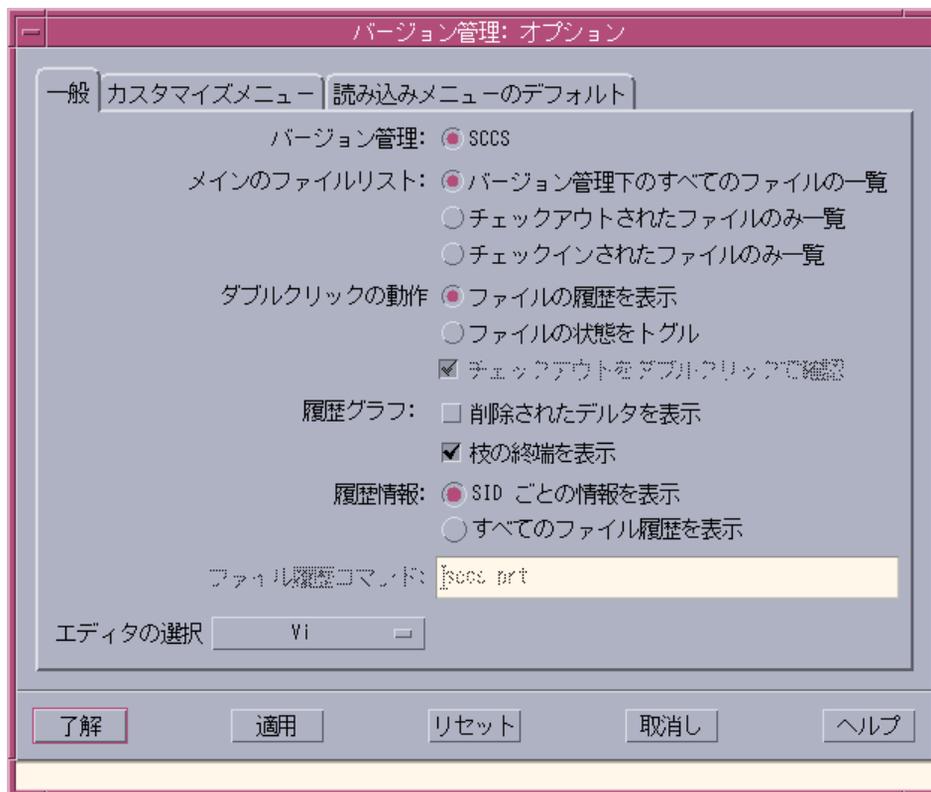


図 7-5 バージョン管理ツールの「オプション」ダイアログボックス

バージョン管理ツールの「オプション」ダイアログボックスの「一般」タブには、表 7-3 に示す項目があります。

表 7-3 バージョン管理ツールの「オプション」ダイアログボックス

項目	説明
バージョン管理	現在は SCCS だけが選択できます。
メインのファイルリスト	「バージョン管理」ウィンドウに表示する SCCS ファイルの種類を指定できます。
ダブルクリックの動作	「ファイルの状態をトグル」を選択した場合は、「チェックアウトをダブルクリックで確認」チェックボックスも選択できます。この場合は、チェックインの際、ダイアログボックスの「了解」をクリックするまで、ファイルのアイコン上にチェックマークが表示されたままになります。

表 7-3 バージョン管理ツールの「オプション」ダイアログボックス

項目	説明
履歴グラフ	履歴グラフ区画に表示する項目を設定できます。
履歴情報	表示する履歴情報の種類を選択できます。「すべてのファイル履歴を表示」を選択した場合は、履歴を収集するコマンドを指定できます。
ファイル履歴コマンド	「すべてのファイル履歴を表示」属性を選択した場合に有効になるフィールドです。履歴ファイルの表示に使用する <code>sccs prt</code> コマンドのオプションを指定できます。デフォルトは、オプションなしの <code>sccs prt</code> コマンドです。
エディタの選択	デルタの内容を表示したり、編集するデルタを開いたりするとき自動的に起動するエディタを指定できます。「その他」を選択した場合は、独立したウィンドウでエディタを起動するコマンドをテキストボックスに入力する必要があります。ファイル名は指定したコマンドに付加されます。

SCCS ファイルの属性の設定

「SCCS ファイルの属性」ダイアログボックスを使用して、SCCS フラグの設定を変更できます。これらのフラグは、`sccs-admin` コマンドに対するオプションになります。SCCS ファイル属性については、Solaris の『プログラミングユーティリティ』または `sccs-admin` のマニュアルページを参照してください。

SCCS ファイルの属性を設定する手順は次のとおりです。

1. 「バージョン管理」ウィンドウからファイルを 1 つ選択します。
2. 「ファイル」▶「ファイル情報」を選択して、「SCCS ファイルの属性」ダイアログボックスを開きます。
3. 「SCCS ファイルの属性」ダイアログボックスでファイル属性を設定します。

「SCCS ファイルの属性」ダイアログボックスには、表 7-4 に示す項目 (フラグ) があります。

表 7-4 SCCS ファイルの属性

フラグ	説明
エンコーディング使用 (e' フラグ)	読み取り専用で、値は「はい」または「いいえ」のいずれかです。
'No id keywords' の扱い (i' フラグ)	ID キーワードがないファイルが検出されたときに「エラー」または「警告」のどちらを発行するかを指定します。
空のリリース (n' フラグ)	リリースが省略されたときに空のリリースの作成を許可するかどうかを指定します。値は「使用する」、「使用しない」のいずれかです。
デルタの分岐 (b' フラグ)	分岐デルタを有効にするかどうかを指定します。値は「有効」または「無効」のいずれかです。
並行更新 (j' フラグ)	並行更新を許可するかどうかを指定します。値は「使用する」または「使用しない」のいずれかです。
リリースの上限 (c' フラグ)	チェックアウト可能な最大リリース数を設定します。値は 1 ~ 9999 の範囲で、デフォルトは 9999 です。
リリースの下限 (f' フラグ)	チェックアウト可能な最小リリース数を設定します。値は 1 ~ 9999 の範囲で、デフォルトは 1 です。
デフォルト SID (d' フラグ)	デフォルトのデルタ番号 (SID) を設定します。たとえば、1.6 というような番号を入力できます。SID については、99 ページの「ファイルの履歴 (デルタ、分岐、バージョン)」を参照してください。
リリースをロック (l' フラグ)	デルタからリリースを保護します。指定したリリースのファイルのチェックアウトや編集ができなくなります。リリース番号を入力します。複数のリリース番号を入力する場合は、2.1, 2.1.3 というようにコンマで区切って入力します。
'Q' キーワードの値 (q' フラグ)	ファイルを読み取り専用で開いたときにキーワード値を設定します。

表 7-4 SCCS ファイルの属性 (続き)

フラグ	説明
モジュール名 (<code>'m'</code> フラグ)	モジュール名キーワード値を設定します。デフォルトは、先頭の <code>s</code> が削除された SCCS ファイル名です。
モジュールの型 (<code>'t'</code> フラグ)	モジュールタイプ値を設定します。
妥当性検査プログラム (<code>'v'</code> フラグ)	新しいデルタに対する変更要求 (MR) 番号用の妥当性検査用プログラムを指定します。ファイルをチェックインしようとする、ファイルに対する MR 番号の入力が求められ、入力された MR 番号が妥当性検査用プログラムに渡されます。妥当性検査用プログラムが成功した場合に、チェックインが許可されます。

第8章

フリーズポイントツールの使用方法

フリーズポイントツールを利用して、ワークスペース内のファイルの「スナップショット」を保存し、後でそのときのファイルを抽出できます。

この章は、次の節で構成されています。

- フリーズポイントツールの概要
- フリーズポイントツールの仕組み
- フリーズポイントツールの起動
- フリーズポイントファイルの作成
- フリーズポイントファイルの更新
- ファイルの抽出
- フリーズポイントの自動作成
- フリーズポイントファイルの形式

フリーズポイントツールの概要

ソフトウェアの開発過程では、作業の主な時点での「フリーズポイント」を作成しておくのが便利です。これらのフリーズポイントはプロジェクトの「スナップショット」の役割を果たし、プロジェクトの主な時点での状態を後で復元できます。

プロジェクトの状態を保存する方法として、標準のバックアップユーティリティを使用してプロジェクト階層のコピーをそのまま作成するという方法があります。この方法は効果的ですが、大量の記憶資源と時間を消費します。フリーズポイントツールを使用することによって、わずかな記憶資源でフリーズポイントを迅速かつ簡単に保存できます。

この章では、グラフィカルユーザーインターフェイス (GUI) 形式のフリーズポイントツールを説明します。フリーズポイントツールは、コマンド行から利用することもできます(第 11 章を参照)。

フリーズポイントツールの仕組み

フリーズポイントツールによって、ワークスペースからフリーズポイントファイルを作成し、後でそのフリーズポイントファイルを使用して、ワークスペースに含まれていたファイルおよびディレクトリ階層を復元できます。デフォルトでは、フリーズポイントツールはファイルとディレクトリだけを抽出します。すなわち、ファイル履歴を無視して各ファイルのデフォルトのデルタだけを取り出します。ただし、フリーズポイントツールには、すべての抽出ファイルの履歴を含むワークスペースを復元するオプションも用意されています。

作成

フリーズポイントファイルを作成するには、「フリーズポイント」ウィンドウの「ディレクトリとファイル」区画にディレクトリとファイルを指定します(図 8-2 を参照)。フリーズポイントツールはディレクトリ階層を再帰的に下降し、各 SCCS 履歴ファイルに含まれる最後にチェックインされたデルタを検出します。その後それらのファイルのリストと、各デルタの一意の数値識別子から構成されるフリーズポイントファイルを作成します(126 ページの「フリーズポイントファイルの形式」を参照)。

フリーズポイントツールは、ファイルの最後にチェックインされたデルタを保存します。このデルタは、デフォルトのデルタと同じである場合も、異なる場合もあります。次の例では、デフォルトのデルタは 1.3 です。最後にチェックインされたデルタが 1.2.1.1 の場合、フリーズポイントツールが保存するのは、その 1.2.1.1 です。

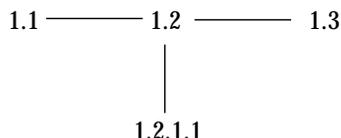


図 8-1 フリーズポイントツールが保存するデルタ

抽出

フリーズポイントファイルを使用して、ファイルを抽出 (回復) できます。フリーズポイントファイルからファイルを抽出するには、フリーズポイントファイルの名前、デルタの抽出元のディレクトリ階層のパス名 (元の階層と異なる場合)、および復元するソース階層を置くディレクトリを指定します。

注 – フリーズポイントファイルは単に相違のリストです。このため、フリーズポイントファイルから抽出を行うには、元のワークスペース (または、その親か子) が存在している必要があります。

抽出では、フリーズポイントファイルに含まれている情報に基づいて、新しいディレクトリ階層が作成されます。新しい階層は、元の履歴ファイルに定義されているファイルで構成されます。抽出時にワークスペースを作成するオプションを選択していない場合は、履歴ファイルそのものは復元されません。デルタは元のソースワークスペースに置かれている履歴ファイルから抽出されます。

ソースワークスペース

ソースワークスペースとは、フリーズポイントファイルの作成元である SCCS 履歴ファイルが含まれるディレクトリ階層のことです。通常、このソースワークスペースから、後でファイルが抽出されて階層が再作成されます。抽出を実行するとき、別のソースディレクトリを指定することもできます。

抽出先ディレクトリ

抽出先ディレクトリとは、フリーズポイントファイルに示されているファイルを抽出するときの最上位のディレクトリです。このディレクトリのパス名は、「フリーズポイント」ウィンドウの「抽出先のディレクトリ」フィールドに指定します。

フリーズポイントツールの起動

フリーズポイントツールを起動するには、次の 2 つの方法があります。

- シェルコマンドプロンプトで後にアンパサンド (&) を付けて `twfreeze` と入力する。

- 「ワークスペース管理」、「ファイルマージ」、「バージョン管理」のいずれかのウィンドウから、「TeamWare」▶「フリーズポイント」を選択する。

「フリーズポイント」ウィンドウの「作成」タブが表示されます (図 8-2 参照)。

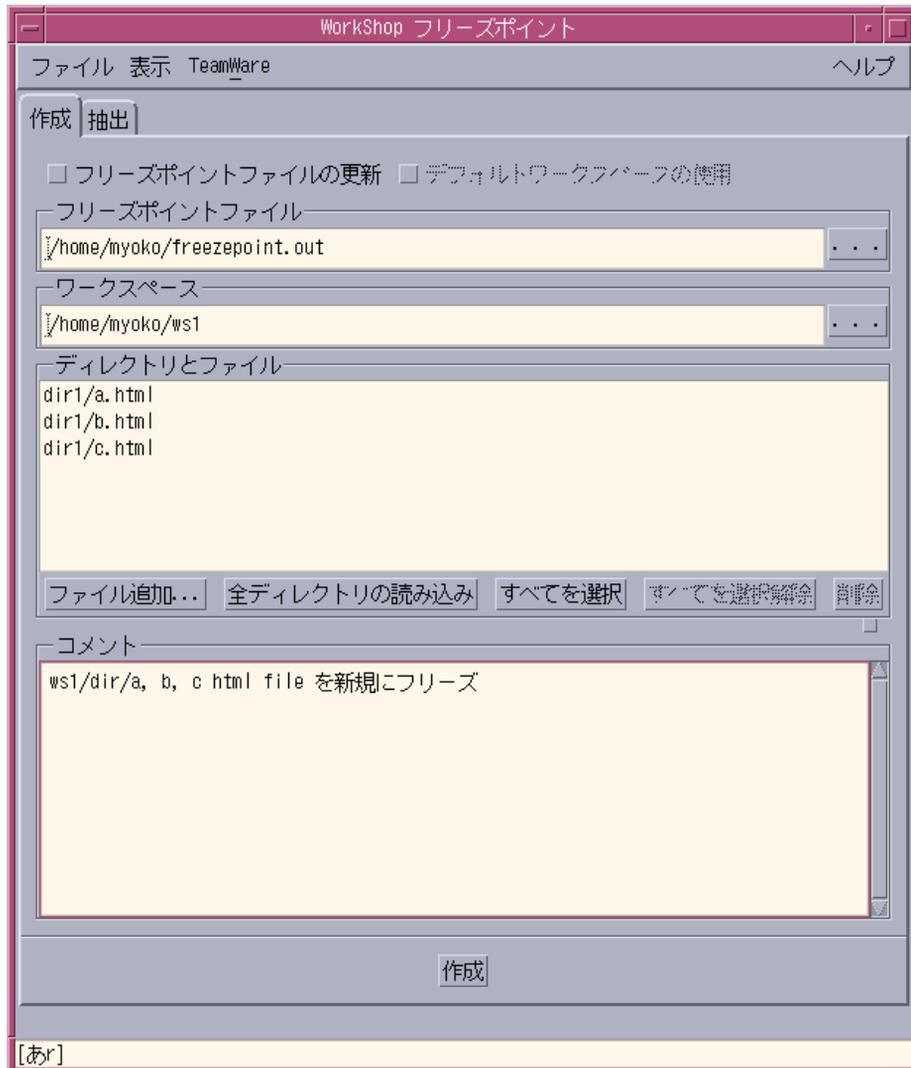


図 8-2 「フリーズポイント」ウィンドウの「作成」タブ

フリーズポイントツールの「作成」タブには、表 8-1 に示す項目があります。

表 8-1 フリーズポイントの「作成」タブの項目

項目	説明
「ファイル」メニュー	フリーズポイントツールを終了するコマンドが表示されます。
「表示」メニュー	「出力を表示」コマンドが表示されます。状態メッセージやエラーメッセージの表示や保存ができます。
「TeamWare」メニュー	他の TeamWare ツールを起動するコマンドが表示されます。
「ヘルプ」メニュー	ヘルプを表示するコマンドがあります。
「フリーズポイントファイルの更新」 チェックボックス	このチェックボックスを選択すると、新しいフリーズポイントを作成せずに既存のフリーズポイントファイルを更新します。
「デフォルトワークスペースの使用」 チェックボックス	このチェックボックスを選択すると、フリーズポイントファイルに指定されているワークスペースが使用されます。
「フリーズポイントファイル」フィールド	フリーズポイントファイルの絶対パス名を指定します。
「ワークスペース」 フィールド	ソースワークスペースを指定します。
「ディレクトリとファイル」区画	フリーズポイントファイルに保存するファイルおよびディレクトリが一覧表示されます。
「ファイル追加」 ボタン	「ファイル追加」ダイアログが開き、「ディレクトリとファイル」区画に追加するファイルを選択できます。
「全ディレクトリの読み込み」 ボタン	ワークスペースのディレクトリ全体を読み込みます。
「すべてを選択」 ボタン	「ディレクトリとファイル」区画に示されているファイルおよびディレクトリをすべて選択します。
「すべてを選択解除」 ボタン	「ディレクトリとファイル」区画に示されているファイルおよびディレクトリすべての選択を解除します。

表 8-1 フリーズポイントの「作成」タブの項目 (続き)

項目	説明
「削除」ボタン	選択した項目をリストから削除します。
「コメント」区画	フリーズポイントファイルに含めるコメントを指定できます。
「作成」ボタン	フリーズポイントファイルを作成して、作成に関するメッセージが含まれた「フリーズポイント出力」ダイアログボックスを表示します。

フリーズポイントファイルの作成

フリーズポイントファイルを作成する手順は次のとおりです。

1. 「ワークスペース管理」ウィンドウから「TeamWare」▶「フリーズポイント」を選択し、「フリーズポイント」ウィンドウを開きます。
「作成」タブが表示されます (図 8-2 を参照)。

2. 「フリーズポイントファイル」フィールドにフリーズポイントファイル名を入力します。デフォルトのファイル名である `freezepoint.out` をそのまま使用することもできます。

フリーズポイントツールを起動すると、「フリーズポイントファイル」フィールドに自動的に `freezepoint.out` が表示されます。このファイル名の前には、フリーズポイントツールを起動したディレクトリのパス名が付けられています。このフィールドを書き換えて任意のフリーズポイントファイルのパス名を入力できます。絶対パス名ではないパス名は、フリーズポイントツールを起動したディレクトリに対する相対パス名とみなされます。

3. 「ワークスペース」フィールドにソースワークスペース名を入力します。

フリーズポイントツールを起動すると、「ワークスペース」フィールドに、`CODEMGR_WS` 環境変数で指定されているワークスペースが自動的に設定されます。この変数の設定がなく、フリーズポイントツールを起動したディレクトリがワークスペース内の階層であった場合は、そのワークスペースのパス名が「ワークスペース」フィールドに設定されます。

4. 「ディレクトリとファイル」区画に、保存するディレクトリとファイルのリストを作成します。
「ファイル追加」ボタンをクリックして、「ファイル追加」ダイアログボックスを開きます。
 - a. 保存するファイルを選択します。
ファイル名をクリックして選択状態にします。Shift キーを押しながらクリックすることによって、複数のファイルをまとめて選択することもできます。
 - b. 「リストにファイルを追加」をクリックして、選択したファイルを「ディレクトリとファイル」区画に追加します。
「全ディレクトリの読み込み」ボタンをクリックすると、「ディレクトリとファイル」区画に「./」と表示されます。これは、ワークスペース階層全体が保存対象になることを示しています。
5. 「コメント」区画にコメントを入力します (省略可)。
このコメントはフリーズポイントファイルに格納され、後で参照できます。
6. 「作成」ボタンをクリックします。
「フリーズポイント」ウィンドウの左下のカウンタに、フリーズポイント処理の進行状況が表示されます。

フリーズポイントファイルの更新

フリーズポイントファイルを更新する手順は次のとおりです。

1. 「ワークスペース管理」ウィンドウから「TeamWare」▶「フリーズポイント」を選択し、「フリーズポイント」ウィンドウを開きます。
2. 「フリーズポイントファイルの更新」のチェックボックスをクリックします。
3. 「フリーズポイントファイル」フィールドにフリーズポイントファイル名を入力します。デフォルトのファイル名である `freezept.out` をそのまま使用することもできます。
4. 「ワークスペース」フィールドの右側にある「…」ボタンをクリックしてワークスペースを選択するか、「ワークスペース」フィールドにワークスペース名を入力します。

5. 「ファイル追加」をクリックして、「ディレクトリとファイル」リストにファイルとディレクトリを追加します。
6. 「コメント」区画にコメントを入力します (省略可)。
7. 「更新」をクリックします。

既存のフリーズポイントファイルに指定されているワークスペースをフリーズポイント処理する場合は、「デフォルトワークスペースの使用」チェックボックスを選択します。

ファイルの抽出

フリーズポイントファイルに記述されているソース階層を新たに抽出する手順は次のとおりです。

1. 「ワークスペース管理」ウィンドウから「TeamWare」▶「フリーズポイント」を選択し、「フリーズポイント」ウィンドウを開きます。
「作成」タブが表示されます。
2. 「抽出」タブをクリックします。
「抽出」タブが表示されます (図 8-4 を参照)。
3. 「フリーズポイントファイル」フィールドに既存のフリーズポイントファイルのパス名を入力します。
絶対パス名でないパス名は、フリーズポイントツールを起動したディレクトリに対する相対パス名とみなされます。
4. 「すべて抽出」または「部分抽出」ラジオボタンを選択します。
「すべて抽出」では、凍結されたファイルセット全体が抽出されます。「部分抽出」では、指定したサブセットが抽出されます。ワークスペースを作成するかどうかに関係なく、全抽出または部分抽出のいずれかを選択する必要があります。
5. 凍結ファイルの SCCS 履歴を含むワークスペースを作成するには、「ワークスペースの作成」チェックボックスを選択します。
6. 「フリーズポイントの抽出元」の 3 つのラジオボタンのいずれかを選択して、ソースワークスペースを指定します。

- 「フリーズポイントファイルからデフォルトを使用」ボタンを選択した場合は、フリーズポイントファイルに指定されているソースワークスペースのパス名が使用されます。
- 「ユーザー指定」ボタンを選択した場合は、ワークスペースのパス名を入力します。
- 「デフォルトとコメントを表示」ボタンを選択した場合は、「ワークスペース」フィールドにソースワークスペースのパス名が表示されます。

フリーズポイントファイルに含まれていないソースワークスペース階層を指定するには、「ユーザー指定」ボタンを選択して、「ワークスペース」フィールドに別のソースワークスペースのパス名を入力します。

7. 「抽出先のディレクトリ」フィールドに、新しい (抽出した) 階層を置くディレクトリのパス名を入力します。

絶対パス名でないパス名は、フリーズポイントツールを起動したディレクトリに対する相対パスとみなされます。

抽出先ディレクトリには、新しいディレクトリと既存のディレクトリのどちらでも指定できます。抽出先ディレクトリに既存のディレクトリを指定した場合は、警告メッセージが表示されるため、操作の確認を行う必要があります。

8. 「抽出」ボタンをクリックして、抽出を開始します。

「部分抽出」オプションを選択した場合は、ダイアログが開き、フリーズポイントファイルに含まれているソースファイルのリストが表示されます。リストから抽出するファイルを選択してください。

「抽出」ボタンをクリックすると、フリーズポイントファイルに示されている各ソースファイルに対して `SCCS get` が実行されます。抽出されるファイルのバージョンは、フリーズポイントファイルの SMID で指定されているバージョンです。抽出された g- ファイルは、抽出先のディレクトリに書き込まれます。「ワークスペースの作成」オプションを選択した場合は、SCCS 履歴も抽出先ディレクトリに書き込まれます。

「フリーズポイント」ウィンドウの左下のカウンタに、抽出処理の進行状況が表示されます。

終了 - 3 個のファイルを処理しました

図 8-3 フリーズポイント処理の進行状況

注 - 抽出の際にファイルの名前変更や削除などが原因でファイルが見つからない場合は、抽出は中止され、エラーとなったエントリの名前が表示されます。この場合、フリーズポイントファイルを編集して、エラーとなったエントリを削除する必要があります。名前が変更されたファイルの新しい名前を確認する方法については、[freezeptfile\(4\)](#)のマニュアルページを参照してください。

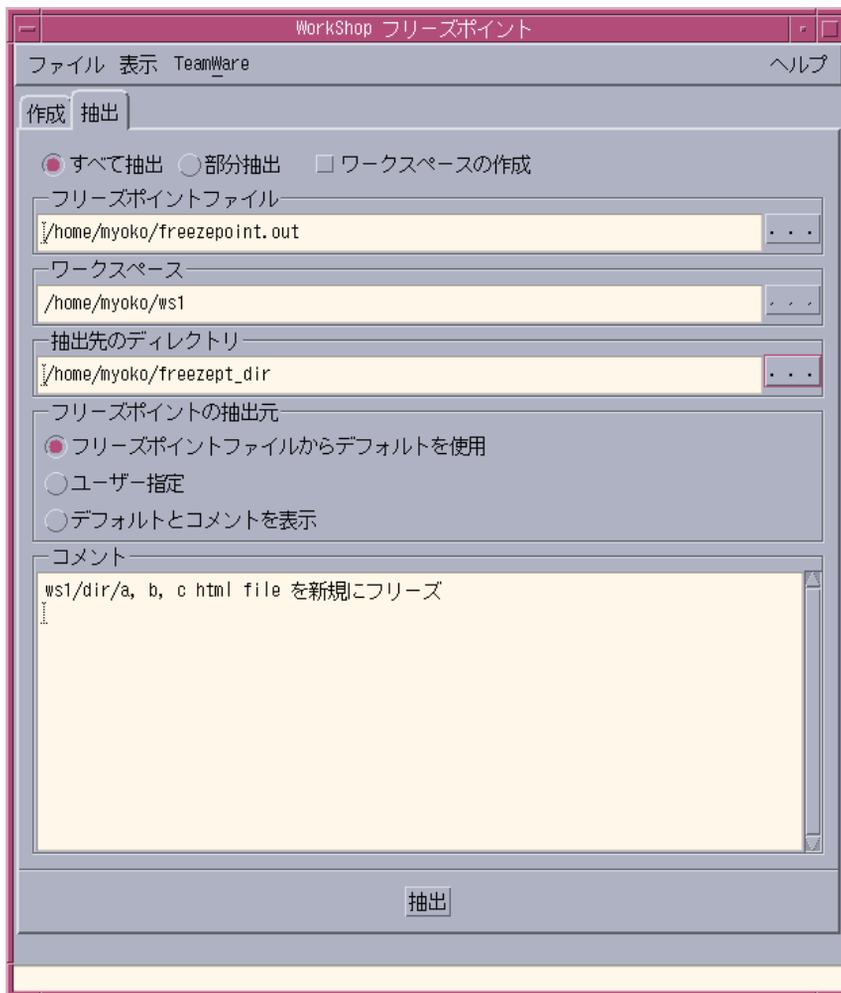


図 8-4 「フリーズポイント」ウィンドウの「抽出」タブ

フリーズポイントの「抽出」タブには、表 8-2 に示す項目があります。

表 8-2 フリーズポイントの「抽出」タブの項目

項目	説明
「ファイル」メニュー	フリーズポイントツールを終了するコマンドが表示されます。
「表示」メニュー	「出力を表示」コマンドが表示されます。
「TeamWare」メニュー	他の TeamWare ツールを起動するコマンドが表示されます。
タブ	「フリーズポイント」ウィンドウの「作成」タブと「抽出」タブを切り替えることができます。
「すべて抽出」 ラジオボタン	フリーズポイントファイルのセット全体をすべて抽出します。
「部分抽出」 ラジオボタン	フリーズポイントファイルのセットの一部を抽出します。
「ワークスペースの作成」 チェックボックス	フリーズポイントファイルから TeamWare のワークスペースを作成します。
「フリーズポイントファイル」 フィールド	フリーズポイントファイルの絶対パス名を指定します。
「ワークスペース」 フィールド	ソースワークスペースを指定します。
「抽出先のディレクトリ」 フィールド	新しい階層を置くディレクトリのパス名を指定します。
「フリーズポイントの出元」 ラジオボタン	「フリーズポイントファイルからデフォルトを使用」 フリーズポイントファイルで指定されているソースワークスペースのパス名が使用されます。 「ユーザー指定」 ワークスペースのパス名を入力します。 「デフォルトとコメントを表示」 ソースワークスペースのパス名が使用されます。

フリーズポイントの自動作成

プリングオーバーやプットバックなどの、特定のトランザクションを実行するたびに、フリーズポイントを作成するように設定できます。

自動フリーズポイントを設定する手順は次のとおりです。

1. 「ワークスペース管理」ウィンドウからワークスペースを選択します。
2. 「ワークスペース」▶「属性」を選択します。
3. 「フリーズポイント」タブを選択します。
「ワークスペース属性」ダイアログボックスの「フリーズポイント」タブが表示されます (図 8-5 を参照)。
4. 「フリーズポイント」タブは、フリーズポイントを作成する時期を示すオプションのリストです。適切なオプションの「はい」ラジオボタンを選択します。
5. 「了解」をクリックします。

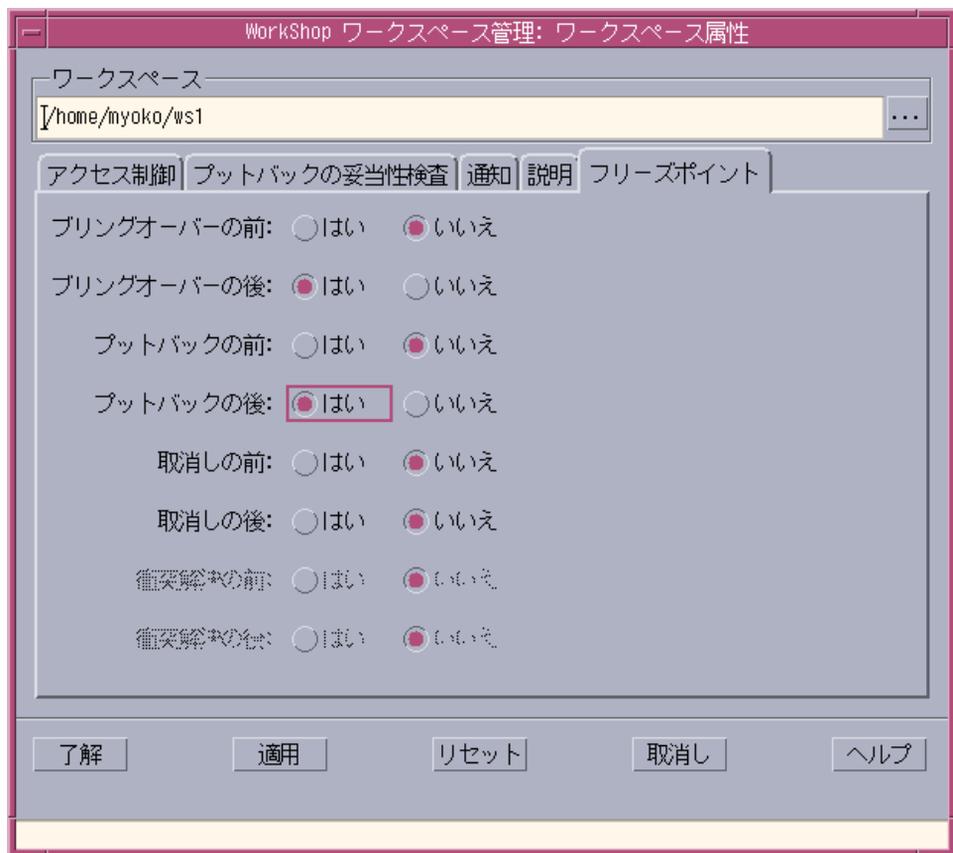


図 8-5 「ワークスペースの属性」ダイアログボックスの「フリーズポイント」タブ

フリーズポイントツールによって、`Codemgr_wsdata/FreezePoints` ディレクトリ下のワークスペースにフリーズポイントファイルが作成されます。自動フリーズポイント機能は、`Codemgr_wsdata/FreezePoints/YYYY/MM/DD` という形式 (YYYY は年、MM は月、DD は日) で、フリーズポイントファイルが作成された日ごとにディレクトリを作成します。フリーズポイントファイルの名前は、`HHMMSS.fp.Z` という形式 (HH は時、MM は分、SS は秒) になります。拡張子の `.Z` は、自動フリーズポイントファイルが `/bin/compress` コマンドで圧縮されていることを示します。

フリーズポイントファイルの形式

フリーズポイントファイルの内容は、通常のテキストエディタを使用して表示できません。

フリーズポイントファイルは、保存対象のワークスペース階層中の SCCS 履歴ファイルに含まれるデフォルトのデルタのリストからなるテキストファイルです。階層を後に復元するときに、フリーズポイントツールはこれらのエントリを、元の履歴ファイルと、フリーズポイントファイルの作成時点でデフォルトだったデルタへのポインタとして使用します。

これらのデルタの識別には、標準の SCCS デルタ ID (SID) の代わりに、新しい識別子、SCCS マージ可能 ID (SMID) が使用されます。SMID を使用すると、ワークスペース管理ツールの更新プリングオーバートランザクションの中で SID が変更されたファイルを、フリーズポイントツールで正しく処理できます。詳細は、191 ページの「SCCS マージ可能 ID について」を参照してください。

フリーズポイントファイルには次の情報が含まれています。

- 作成したデルタリストの元となったワークスペースのパス名
- ファイルの作成日時
- フリーズポイントを作成したユーザーのログイン名
- 各ファイルに対応する SCCS 履歴ファイル中の最新の SCCS デルタを示す一連の 16 進数
- 各ファイルの対応する SCCS 履歴ファイル内のルートのデルタを示す、一連の 16 進数
- ユーザーが入力したコメント (オプション)

以下に、フリーズポイントファイルの例を示します。この例には、3つのエントリがあり、残りの行はコメントになっています。

```
# Format:
# sfilename (previously SID) date time user hex hex hex hex hex hex hex hex
#
# First four hex's are the SCCS Mergeable Id of the root delta of the
# containing delta tree.
# Last four hex's are the SMID of the desired delta.

#sdata=99/03/10 09:20:26
#sdata=SunPro Code Manager data about conflicts, renames, etc...
#sdata=Name history : 1 0 list.c
#leaf=9f7398c4 cc06b ff6ce975 10b7656b
#leaf=f6ea91e2 bbd23cd1 3e052ed1 ca969a9e
./list.c (previously 1.5) 99/06/21 14:04:22 toriw 11db401e cd439eeb ca3782dc
1aa255e9 97701645 bda0137e d24a3d6b 69f31f25
#sdata=99/03/10 09:20:24
#sdata=SunPro Code Manager data about conflicts, renames, etc...
#sdata=Name history : 2 1 testdir/index.html
#sdata=Name history : 1 0 index.html
#leaf=1fffddc9 85c63827 8172c838 52ba549b
./twtest/testdir/index.html (previously 1.6) 99/06/08 16:21:37 toriw 3ed3beea
bb06794d 3f235871 dd89b225 d10d3db7 b8384098 3bb361a0 32e64f5e
#sdata=99/03/10 09:20:24
#sdata=SunPro Code Manager data about conflicts, renames, etc...
#sdata=Name history : 2 1 testdir/routine
#sdata=Name history : 1 0 routine
./twtest/testdir/routine1fs (previously 1.2) 99/03/12 09:11:10 toriw e59da845
d7e3b7e6 f9e7765f 43d41389 89c058c1 f758edad 81fc5a22 576e5015
```


第9章

Sun WorkShop TeamWare におけるプログラムの構築

Sun WorkShop TeamWare では、一度に1つの構築ジョブを実行することも、同時に複数の構築ジョブを実行することもできます。この章では、1つのアプリケーションを短時間で構築する方法、構築のカスタマイズ方法、および「構築」ウィンドウと任意の Sun WorkShop エディタを使用して構築エラーを修正する方法について説明します。この章は、次の節で構成されています。

- 「構築」ウィンドウ
- WorkShop ターゲットの構築
- プログラムの構築
- 構築のカスタマイズ
- 構築エラーの修正

「構築」ウィンドウ

「構築」ウィンドウには、プログラムのコンパイルに関する情報が表示されます。このウィンドウを開くには、「ワークスペース管理」ウィンドウから「TeamWare」▶「構築」を選択します。

注 – Sun WorkShop から利用できるものと同じ構築ツールが起動されます。Sun WorkShop のメインウィンドウから「ウィンドウ」▶「構築ウィンドウの表示」を選択して起動できます。

「構築」ウィンドウでは、次の作業を行うことができます。

- 構築の開始
- 構築の中止
- 構築パラメータの編集
- 別のファイルへの構築出力の保存
- 構築エラーの表示

図 9-1 に「構築」ウィンドウを示します。

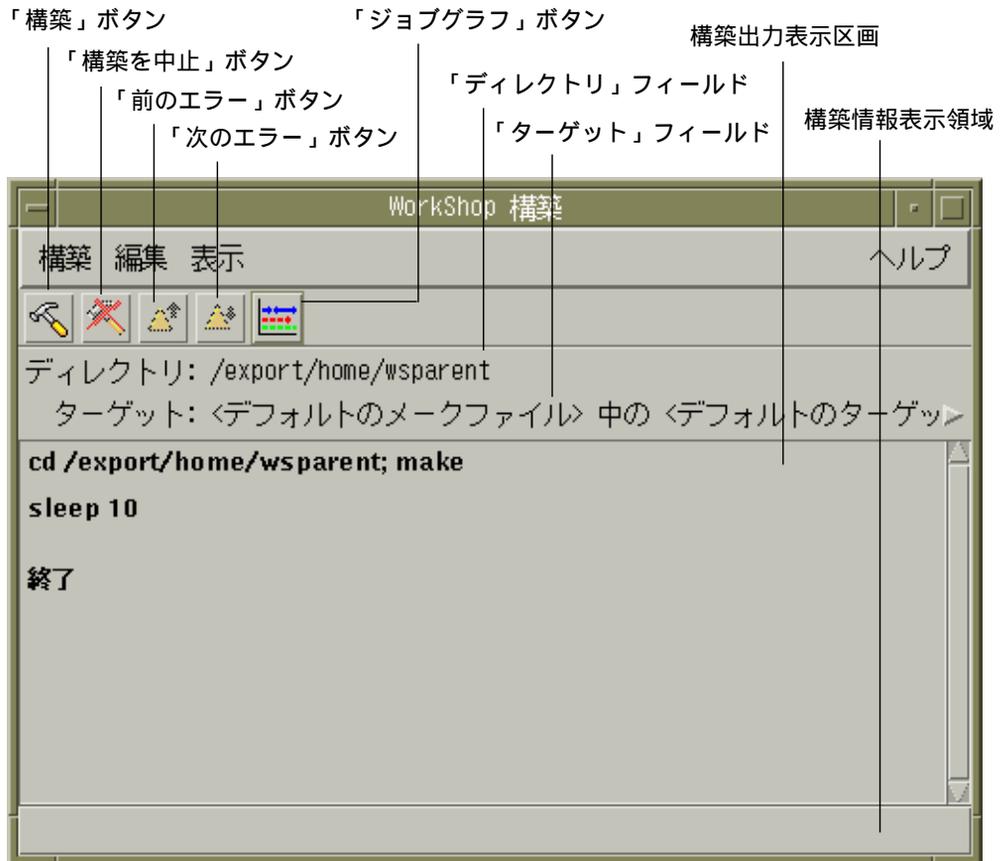


図 9-1 「構築」ウィンドウ

「構築」ウィンドウには、表 9-1 に示す項目があります。

表 9-1 「構築」ウィンドウの項目

項目	説明
「構築」メニュー	ターゲットの指定、構築パラメタの設定、構築の起動と停止、構築出力の保存を行うためのコマンドがあります。
「編集」メニュー	構築出力表示区画に表示する内容を変更するためのコマンドがあります。
「表示」メニュー	「分散メークのジョブグラフ」ウィンドウに構築エラーを表示したり、 <code>dmake</code> による構築の進行状況をグラフィカルに表示したりするためのコマンドがあります。
「構築」ボタン	現在の Sun WorkShop ターゲットの構築を開始します。
「構築を中止」ボタン	現在実行中の構築を中止します。
「前のエラー」ボタン	構築出力表示区画で、前の構築エラーにカーソル移動し、そのエラーの発生場所をテキストエディタに表示します。
「次のエラー」ボタン	構築出力表示区画で、次の構築エラーにカーソル移動し、そのエラーの発生場所をテキストエディタに表示します。
「ジョブグラフ」ボタン	「分散メークのジョブグラフ」ウィンドウを開きます。複数のジョブの構築時間を確認できます。
「ディレクトリ」フィールド	現在の構築ディレクトリのパス名が表示されます。
「ターゲット」フィールド	現在の構築ターゲットの名前が表示されます。
構築出力表示区画	構築の出力が読み取り専用で表示されます。
構築情報表示領域	現在実行中の構築に関する情報が表示されます。

WorkShop ターゲットの構築

Sun WorkShop プログラミング環境における構築に関するターゲットは、Sun WorkShop ターゲットとユーザーメークファイルターゲットの 2 種類があります。

Sun WorkShop ターゲット

Sun WorkShop ターゲットは、構築ディレクトリ、構築コマンド、メイクファイル、構築ターゲットで構成されるオブジェクトです。

- 構築ディレクトリ：構築プロセスが起動されるディレクトリで、メイクファイルのデフォルトのディレクトリでもあります。
- 構築コマンド：`make` ユーティリティを起動するコマンド。`make` ユーティリティはメイクファイルを読み取り、構築ターゲットを構築します。
- メイクファイル：依存関係に基づいて構築ターゲットを最新にする方法を記述した項目を含んでいるファイルです。1つの依存関係が1つの構築ターゲットになるため、依存関係自体が複数の依存関係を持つことがあります。構築ターゲットを再構築するかどうかを決定するときに、ターゲットとファイル間の依存関係と下位の依存関係によって `make` がトレースするツリー構造が形成されます。
- 構築 (`make`) のターゲット：`make` は特定のメイクファイルに含まれている指示 (規則) から構築ターゲットの構築方法を得ます。たとえば、`make` のターゲットは `all` になることもあれば、`clean` になることもあります。通常、メイクファイルは、最も一般的な構築ターゲットがデフォルトターゲット (ターゲットを指定しなかったときに構築されるターゲット) になるように設計されています。

構築された Sun WorkShop ターゲットは、「構築」メニューのピックアップと「構築」▶「ターゲットを編集」コマンドの Sun WorkShop ターゲットのピックアップに追加されます。構築を開始すると、Sun WorkShop ターゲットのリストの先頭にあるターゲットが構築されます。

プロジェクトには、複数のターゲットを含むことができます。実行可能ファイル、静的ライブラリ / アーカイブ、共有ライブラリ、Fortran アプリケーションの場合は、実行可能ファイル / ライブラリが1つのターゲットで、特殊な `clean` ターゲットがもう1つのターゲットということになります (「構築」メニューのピックアップにある)。`clean` ターゲットは、プロジェクトで作成されたすべてのファイル (たとえば `.o` ファイル)、ソースブラウザ用データベース、C++ テンプレートデータベース、実行可能ファイルそのもの、およびその他の構築関係のファイルを削除します。

複雑なプロジェクトの場合は、「構築」メニューのピックアップに複数のターゲットが含まれることがあります。たとえば、プロジェクトで、5つのライブラリと1つの実行可能ファイルを作成し、まとめてリンクできます。この場合、それぞれのライブラリおよび実行可能ファイルは1つの Sun WorkShop ターゲットになり、「構築」メニューのピックアップから選択することで個別に構築できます。

ユーザーメイクファイルターゲット

ユーザーメイクファイルターゲットは、`make` が特定のメイクファイルに含まれる指示 (規則) からその構築方法に関する情報を得るオブジェクトです。通常、メイクファイルは、最も一般的なターゲットがデフォルトターゲット (ターゲットを指定しなかったときに構築されるターゲット) になるように設計されています。

メイクファイルには、依存関係に基づいて構築ターゲットを最新にする方法を記述した項目が含まれています。1 つの依存関係が 1 つの構築ターゲットであるため、依存関係自体が複数の依存関係を持つことがあります。構築ターゲットを再構築するかどうかを決定するときに、ターゲットとファイル間の依存関係と下位の依存関係によって `make` がトレースするツリー構造が形成されます。

ユーザーメイクファイルプロジェクトの場合、「構築」メニューのピックリストに含まれるターゲットは、メイクファイルが構築するメイクファイルターゲットになります。

プログラムの構築

Sun WorkShop TeamWare では、構築コマンド、メイクファイル、構築ターゲットを指定しなくても構築を開始できます。これらのうちの 1 つまたはすべてを指定して構築を開始することもできます。構築オプションの指定、構築モードの指定、メイクファイルマクロの変更、環境変数の編集などの構築のカスタマイズも可能です (139 ページの「構築のカスタマイズ」を参照)。

構築パラメータは、「ターゲットの新規定義」ダイアログおよび「ターゲットの編集」ダイアログを使用して指定します。この 2 つのダイアログの基本的な構成は同じです。新しい WorkShop ターゲットの指定には「ターゲットの新規定義」ダイアログを使用し、既存の WorkShop ターゲットの編集には「ターゲットの編集」ダイアログを使用します。図 9-2 は「ターゲットの新規定義」ダイアログです。

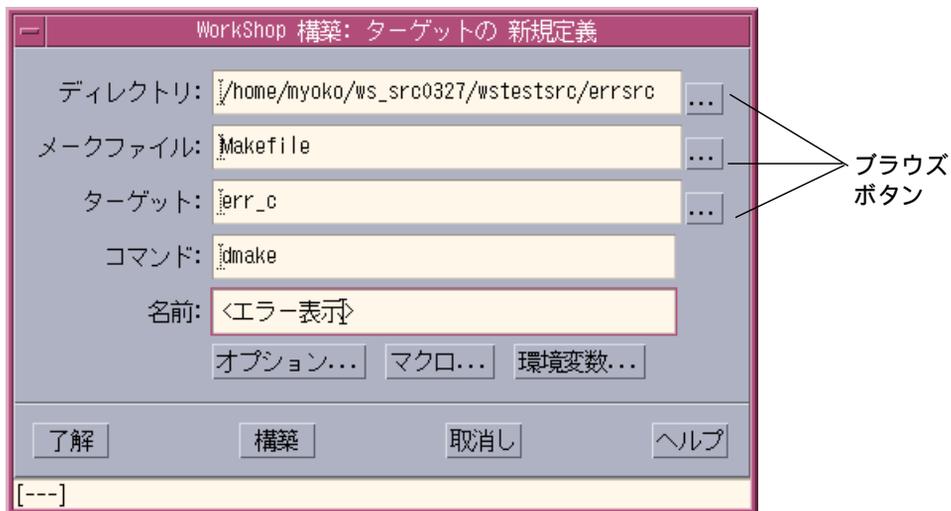


図 9-2 「ターゲットの新規定義」ダイアログ

「ターゲットの新規定義」および「ターゲットの編集」ダイアログボックスには、表 9-2 に示す項目があります。

表 9-2 「ターゲットの新規定義」ダイアログボックスの項目

項目	説明
「ディレクトリ」フィールド	構築ディレクトリのパスを入力します。ブラウザボタンをクリックしてディレクトリを選択することもできます。
「メイクファイル」フィールド	メイクファイルを指定します (デフォルトのファイル名は <code>makefile</code>)。ブラウザボタンをクリックしてメイクファイルを選択することもできます。
「ターゲット」フィールド	ターゲットを指定します。ブラウザボタンをクリックすると「ターゲット選択」ダイアログボックスが開きます。
「コマンド」フィールド	構築コマンドを入力します。デフォルトのコマンドは <code>dmake</code> です。

表 9-2 「ターゲットの新規定義」ダイアログボックスの項目 (続き)

項目	説明
「名前」フィールド	指定したディレクトとメイクファイル、ターゲットの組み合わせに名前を付けることができます。指定した名前は、メニューピックリスト上でターゲットを示すために使用されるだけです。ターゲットに名前を付けなかった場合、メニューピックリストには、「ディレクトリ: <code>makefile</code> 中のターゲット」と表示されません。名前を付けることによって構築ターゲットを区別することができ、同じディレクトリとメイクファイル内に複数の構築ターゲットがある場合に便利です。
「オプション」ボタン	構築パラメータを変更できます。
「マクロ」ボタン	構築に渡すマクロの追加、変更、書き換え、削除を行うことができます。
「環境変数」ボタン	構築に渡す環境変数の追加、変更、削除できます。
「了解」ボタン	構築パラメータを適用してダイアログを終了します。
「構築」ボタン	構築パラメータを適用してターゲットを構築します。
「取消し」ボタン	変更内容を適用せずにダイアログを終了します。
「ヘルプ」ボタン	ダイアログに関するオンラインヘルプが表示されます。

デフォルト値を使用した構築

Sun WorkShop には、デフォルトの構築ターゲットとデフォルトの `make` コマンド (`dmake`) があります。このため、構築コマンドや構築ターゲットを指定しなくても構築を開始できます。ただし、ユーザーメイクファイルプロジェクトを構築する場合や、プロジェクトが読み込まれていない場合は、メイクファイルを指定する必要があります (Sun WorkShop は、`makefile` または `Makefile` という名前のファイルを検索して、`make` がどちらのファイルを使用するか決定できるようにします)。

Sun WorkShop のこのプロジェクト機能によって、「新規プロジェクト作成」ウィザードや「現在のプロジェクトを編集」ウィンドウからメイクファイルを作成するように指示できます。詳細は、オンラインヘルプを参照してください。

Sun WorkShop には、構築コマンドとして `dmake` が用意されています。`dmake` コマンドはメイクファイルを解析して、並行して構築可能なターゲットを判断し、ターゲットの構築ジョブを分散します。`dmake` は、次のいずれかの構築モードで動作します (構築モードの設定については、139 ページの「構築のカスタマイズ」を参照)。

- 逐次モード — `make` 同様、ローカルホスト上で一度に 1 つずつジョブを実行します。
- 並列モード — ローカルホスト上で複数のジョブを並行実行します。
- 分散モード — 複数の構築サーバーに分散して複数のジョブを実行します。分散モードでは、多数のプログラムで構成される大規模プロジェクトの構築処理を複数のサーバーに同時に分散できます。`dmake` はメークファイルを解析して、並行して構築可能なターゲットを判断し、指定された複数の構築サーバーにそれらターゲットの構築ジョブを分散します。分散モードで実行するにあたっては、139 ページの「構築のカスタマイズ」を参照してください。

これらのどのモードで `dmake` を実行して、「ジョブグラフ」ウィンドウを使用して、その処理の進行状況を監視したり、各構築ジョブの状態を表示したりできます。「構築」ウィンドウから「ジョブグラフ」ウィンドウを開くには、「表示」▶「分散メークのジョブグラフ」を選択します。

「ターゲットの新規定義」と「ターゲットの編集」ダイアログボックスには、デフォルトの構築値が表示されます。特定のメークファイルや構築ターゲットが指定されなかった場合、Sun WorkShop は、構築ディレクトリの `makefile` という名前のファイルを探し、そのファイル内の最初の構築ターゲットを使用します。ただし、`makefile` ファイルよりも新しい SCCS 履歴ファイル (`s.makefile`) を検出した場合、Sun WorkShop は最新の `s.makefile` を使用します。メークファイルが存在しない場合は、`Makefile` というファイルを探します。

デフォルトの構築値を使用してプログラムを構築する手順は次のとおりです。

1. 「構築」ウィンドウの「ディレクトリ」フィールドに適切な構築ディレクトリが設定されていることを確認します。

構築ディレクトリは、構築プロセスを実行するディレクトリで、メークファイルのデフォルトのディレクトリでもあります。「ディレクトリ」フィールドに構築ディレクトリが表示されていないか、構築ディレクトリを変更する場合は、「構築」▶「新規ターゲット」を選択して、「ターゲットの新規定義」ダイアログボックスを開き、「ディレクトリ」テキストボックスに構築パスを入力します。

2. 「構築」▶「Start Build」を選択して、構築を開始します。

「構築」ウィンドウの構築出力表示区画に構築出力が表示されます。構築を中止するには、「構築」ウィンドウから「構築を中止」ボタンをクリックするか、「構築」▶「構築を中止」を選択します。

注 – 次に「構築」ウィンドウを開いたときは、最後に構築ジョブを実行したディレクトリが構築ディレクトリとして設定され、「ディレクトリ」フィールドにそのパス名が表示されます。

独自の構築値を使用した構築

一意の名前を持つメイクファイル、独自の構築ターゲット、または独自の構築コマンドがある場合は、「ターゲットの新規定義」ダイアログボックスや「ターゲットの編集」ダイアログボックスでその構築値を指定できます（この指定ができるのは、ユーザーメイクファイルプロジェクトに対してか、プロジェクトが読み込まれていない場合です）。たとえば、独自の構築コマンドを指定することによって、`make` の出力をフィルタにかけ、不要な警告を排除できます。少なくとも構築ディレクトリを指定する必要があります。Sun WorkShop は `make` コマンドを使用し、`make` の検索順序に従ってメイクファイルを検索します。詳細は、[make\(1S\)](#) のマニュアルページと、オンラインヘルプを参照してください。

独自の構築値を指定する手順は次のとおりです。

1. 「ディレクトリ」フィールドに、構築に使用するディレクトリの名前を入力します。
構築ディレクトリが指定されない場合、Sun WorkShop は現在表示されているディレクトリで構築に行おうとします。ディレクトリが表示されていない場合は、エラーを示すポップアップウィンドウが表示されます。
2. 「メイクファイル」フィールドに、使用するメイクファイル名を入力します。
3. 「ターゲット」フィールドに、使用する構築ターゲット名を入力します。
4. 「コマンド」フィールドに、使用する構築コマンド名を入力します。
`make` または `dmake` 以外の構築コマンドを使用する場合は、「コマンド」フィールドにそのコマンドと引数を指定します。`PATH` 環境変数にその構築コマンドへのパスを指定していない場合は、フルパスで指定する必要があります。
構築コマンドは、「環境変数」ダイアログボックスで指定された環境変数に対する `setenv` コマンドを前に追加し、「メイクオプション」と「メイクのマクロ」ダイアログボックスで指定された構築オプションを後ろに付加することによって形成されます。
5. 「構築」ボタンをクリックします。ダイアログボックスで指定された設定に基づいて構築が開始されます。

「構築」ウィンドウの構築出力表示区画に、構築の出力が表示されます。構築を中止するには、「構築」ウィンドウの「構築を中止」ボタンをクリックするか、「構築」▶「構築を中止」を選択します。

既存の WorkShop ターゲットの編集

既存の WorkShop ターゲットを編集する手順は次のとおりです。

1. 「構築」ウィンドウから「構築」▶「ターゲットを編集」を選択します。
2. リストから WorkShop ターゲットを選択します。
「ターゲットの編集」ダイアログボックスが開き、構築ディレクトリとメイクファイル、構築ターゲット、構築コマンドの現在の設定値が表示されます。
3. 必要に応じて、これらの設定を編集します。
詳細は、137 ページの「独自の構築値を使用した構築」を参照してください。
4. 「構築」ボタンをクリックします。新しい設定で WorkShop ターゲットが再構築されます。

構築出力の収集

「構築」ウィンドウの構築出力表示区画の構築出力は、構築ジョブを実行するたびに消去されます。

前の構築出力を残す手順は次のとおりです。

1. 「編集」▶「前の結果を消去しない」を選択します。
構築出力表示区画は消去されず、前の構築の出力に続けて、以降の構築出力が表示されます。
2. 「構築」▶「構築」を選択します。
3. 各構築の出力を確認するには、構築出力表示区画をスクロールします。
各構築ジョブの出力の先頭には、構築パスと構築ターゲット名が表示されます。
構築出力表示区画の内容を消去するには、「編集」▶「結果を消去」を選択します。

構築出力の保存

出力をファイルに保存し、構築出力情報の履歴を作成できます。

構築出力を保存する手順は次のとおりです。

1. 「構築」▶「結果を別名で保存」を選択します。
2. 「構築結果保存」ダイアログを使用して、出力を保存するファイルを選択または作成します。

構築出力のログが、テキストファイルとして保存されます。

WorkShop ターゲットの削除

「構築」メニューの「ターゲットを編集」のリストからターゲットを削除できます。

ターゲットを削除する手順は次のとおりです。

1. 「構築」ウィンドウから「構築」▶「メニューからターゲットを削除」を選択します。
2. 「メニューからターゲットを削除」ダイアログのリストからターゲットを選択します。

Control キーを押しながらクリックすることによって、複数のターゲットをまとめて選択できます。

3. 「了解」ボタンをクリックします。

構築のカスタマイズ

構築オプションの変更、構築モードの指定、メイクファイルマクロの使用、環境変数の使用により、構築をカスタマイズできます。構築をカスタマイズするには、「構築」▶「ターゲットを編集」を選択して、リストからターゲットを選択します。

「ターゲットの編集」ダイアログが開きます。

構築オプションの指定

構築オプションは、「構築オプション」ダイアログボックスで指定できます。「構築オプション」ダイアログボックスを開くには、「ターゲットの編集」ダイアログボックスから「オプション」ボタンをクリックします。必要なオプションを選択したら、「了解」をクリックして、「ターゲットの編集」ダイアログボックスに戻ります。

次のコマンドおよびオプションについての詳細は、[dmake](#) および [make](#) のマニュアルページを参照してください。

カテゴリ: 基本

- コマンドを表示するだけで実行はしません (-n)。
- エラーが発生するか規則が見つからない場合は、依存関係ツリー上の、ターゲットの依存しない分岐で処理を継続します (-k)。

カテゴリ: コマンドの実行と表示

- ターゲットを再構築する理由を表示します (-d)。
新しい依存関係がある場合は、その全情報が表示されます。[MAKEFLAGS](#) 環境変数から構築表示オプションも読み込まれます。
- 依存関係の検査および処理に関する詳細情報を表示します (-dd)。
- 読み込まれたメイクファイルの内容を表示します (-D)。
- メイクファイル、[make.rules](#) ファイル、状態ファイル、隠れている依存関係に関するレポートのすべてを表示します (-DD)。
- エコーなしでコマンドを実行します (-s)。このオプションは、特殊な意味を持つターゲットの [.SILENT](#): と同等です。

カテゴリ: 実行せず表示だけするためのオプション

- マクロ定義とターゲット記述セットの全内容を表示します (-p)。
- 構築しないで、依存関係のレポートを作成します (-P)。
- ターゲットファイルが最新であるかどうかに従って、ゼロまたはゼロ以外の状態コードを返します (-q)。

カテゴリ: その他

- 規則を適用しないでターゲットファイルを更新します (-t)。

- デフォルトの `/usr/share/lib/make/make.rules` メークファイルを無視します (`-r`)。
- メークファイル内の設定より環境変数の設定を優先します (`-e`)。
- コマンドから返されたエラーコードを無視します (`-i`)。このオプションは、特殊な意味を持つターゲットの `.IGNORE:` と同等です。

カテゴリ: 分散メーク

このカテゴリでは、`make` プロセスを実行するモードを指定できます。表 9-3 は、`dmake` オプションと選択したモードで行う必要がある操作をまとめています。

表 9-3 `dmake` のオプション

項目	説明
モード	構築の実行モードを選択します。
逐次	このモードを選択した場合、フィールドへの入力はありません。
並列	「最大ジョブ数」フィールドに、実行する最大構築ジョブ数を指定します。
分散	<ol style="list-style-type: none"> 1. 「最大ジョブ数」フィールドに、実行する最大構築ジョブ数を指定します。 2. 「実行時の構成ファイル」フィールドに、<code>.dmakerc</code> ファイルの名前またはパスを指定します。 3. 「構築サーバーグループ」フィールドに構築サーバーグループを指定します。 4. 「一時出力ディレクトリ」フィールドに、出力ディレクトリのパス名を入力します。
最大ジョブ数	構築サーバーに分散する最大ジョブ数を入力します。省略された場合は、グループのサーバーに指定されたジョブ数の合計が適用されます。
実行時の構成ファイル	実行時構成ファイルを指定します。デフォルトでは、 <code>~/ .dmakerc</code> が使用されます。
構築サーバーグループ	ジョブの分散先のサーバーグループ名を入力します。デフォルトでは、 <code>.dmakerc</code> ファイル内の最初のグループが使用されます。
一時出力ディレクトリ	一時的な出力を書き込むディレクトリ名を指定します。デフォルトでは、 <code>~/ .dmake</code> が使用されます。

初めて分散メークを行う場合は、`dmake` 構築サーバーとして参加するマシンを指定した `.dmakerc` 実行時構成ファイルを作成する必要があります。このファイルには、構築サーバーのグループ (リスト) と各構築サーバーに分散するジョブ数を含めます。`dmake` ユーティリティは、`dmake` ホストからこのファイルを検索することによって、ジョブを分散する場所の情報を取得します。一般に、`.dmakerc` 実行時構成ファイルは、作成した人のホームディレクトリに置きます。実行時構成ファイルを検索できなかった場合、`dmake` はローカルホストに 2 つのジョブを割り当てます。実行時構成ファイルの作成については、`dmake(1)` のマニュアルページを参照してください。

マシンを構築サーバーとして使用するには、そのマシンにジョブを割り当てられるように設定する必要があります。構築サーバーには、`dmake` ホストと同じアーキテクチャで、同じバージョンのオペレーティングシステムが動作しているホストを使用します。デフォルトでは、構築サーバー上の `dmake` 実行可能ファイルへのパスは、`dmake` ホストと同じであるとみなされます。構築サーバー上の `dmake` 実行可能ファイルへのパスが異なる場合は、そのサーバーに対するパス属性を変更する必要があります。

マシンを構築サーバーとして設定するには、そのマシンのファイルシステムに `/etc/opt/SPROdmake/dmake.conf` という構成ファイルを作成する必要があります。このファイルが存在しない場合、`dmake` はそのマシンにジョブを割り当てるのを拒否します。`dmake.conf` ファイルには、そのマシン上で同時に実行可能なジョブ (全ユーザーからのジョブ) の最大数を指定します。`dmake` の詳細については、第 10 章と `dmake(1)` のマニュアルページを参照してください。

メークファイルマクロの使用方法

「メークのマクロ」ダイアログボックスでメークファイルマクロを指定できます。メークファイルマクロを使用することによって、記述ファイル内で使用されているファイルやコマンドオプションを参照できます。「メークのマクロ」ダイアログボックスでは、WorkShop ターゲットの「持続的構築マクロ」リストに対してメークファイルマクロを追加、削除したり、マクロに対する値を変更したりできます。また、メークファイルで現在定義されているマクロをリストに追加し、メークファイルで定義されている値を無視して代わりにこのリストで設定した値を使用することもできます。マクロの定義については、『Sun WorkShop の概要』の付録 B を参照してください。

マクロの追加

「持続的構築マクロ」リストにマクロを追加する手順は次のとおりです。

1. 「ターゲットの編集」ダイアログボックスで「マクロ」をクリックします。
2. 「名前」フィールドにマクロ名を入力します。
3. 「値」フィールドに、追加するマクロに対する値 (または定義) を入力します。
4. 「追加」ボタンをクリックし、新しいマクロをリストに追加します。
5. 必要に応じて、上記の手順を繰り返して他のマクロも追加します。
6. 「了解」ボタンをクリックし、変更内容を適用してダイアログを閉じます。

マクロの削除

「持続的構築マクロ」リストからマクロを削除する手順は次のとおりです。

1. 「持続的構築マクロ」のリストから削除するマクロを選択します。
2. 「削除」ボタンをクリックします。リスト内のすべてのマクロを削除するには「すべてを削除」ボタンをクリックします。
3. 「了解」ボタンをクリックし、変更内容を適用してダイアログを閉じます。

マクロの変更

「持続的構築マクロ」リスト内のマクロの値 (マクロ名によって実際に表される値) を変更する手順は次のとおりです。

1. 「ターゲットの編集」ダイアログボックスで「マクロ」をクリックします。
2. 「メークのマクロ」ダイアログボックスで「詳細」をクリックします。
3. 「メークファイル中のマクロ」リストから変更するマクロを選択します。
4. 「<<追加」をクリックし、選択したマクロを「持続的構築マクロ」リストに追加します。
5. 「値」フィールドに新しい値を入力します。
6. 「変更」ボタンをクリックします。

7. 「了解」ボタンをクリックし、変更内容を適用してダイアログを閉じます。
8. 「ターゲットの編集」ダイアログの「構築」ボタンをクリックすると、新しい値を使用して構築が開始されます。

メイクファイルマクロの表示と置き換え

「持続的構築マクロ」リストにあるマクロ定義が、メイクファイル内の同名のマクロに置き換えられます。

現在のマクロ定義を表示するには、「詳細」ボタンをクリックして「メイクファイル中のマクロ」リストを開きます。このリストには、現在の構築ターゲットに対するメイクファイルで定義されているすべてのマクロが表示されます。「フィルタ」フィールドを使用してリストにフィルタをかけることができます。

メイクファイルマクロの値を置き換える手順は次のとおりです。

1. 「メイクファイル中のマクロ」リストからマクロを選択します。
2. 「<<追加」ボタンをクリックし、選択したマクロを「持続的構築マクロ」リストに追加します。
3. 「値」フィールドに新しい値を入力します。
4. 「変更」ボタンをクリックします。
5. 「了解」ボタンをクリックして、変更内容を適用してダイアログを閉じます。
「持続的構築マクロ」リスト内のマクロ定義が、メイクファイル内の同名のマクロ定義の代わりに使用されるようになります。
6. 「ターゲットの編集」ダイアログの「構築」ボタンをクリックすると、新しい値を使用して構築が開始されます。

環境変数の使用方法

「環境変数」ダイアログボックスで、構築のための環境変数を指定できます。

WorkShop ターゲットの「持続的環境変数」リストに対して環境変数を追加、削除したり、環境変数に対する値を変更したりできます。構築を開始すると、環境変数を設定するための `setenv` コマンドが構築コマンドの前に付加されます。

環境変数の追加

「持続的環境変数」リストに環境変数を追加する手順は次のとおりです。

1. 「ターゲットの編集」ダイアログボックスから「環境変数」をクリックします。
2. 「名前」フィールドに環境変数名を入力します。
3. 「値」フィールドに変数の値を入力します。
4. 「追加」ボタンをクリックし、入力した環境変数を「持続的環境変数」リストに追加します。
5. 必要に応じて、上記の手順 2 ~ 4 を繰り返して他の環境変数も追加します。
6. 「了解」ボタンをクリックしてダイアログを閉じます。

環境変数の削除

「持続的環境変数」リストから環境変数を削除する手順は次のとおりです。

1. 「持続的環境変数」リストから変数を選択します。
2. 「削除」ボタンをクリックします。リスト内のすべての環境変数を削除するには「すべてを削除」ボタンをクリックします。
3. 「了解」ボタンをクリックし、変更内容を適用してダイアログを閉じます。

環境変数の値の変更

「持続的環境変数」リスト内のマクロの値を変更する手順は次のとおりです。

1. 「持続的環境変数」リストから環境変数を選択します。
2. 「値」フィールドに新しい値を入力し、「変更」ボタンをクリックします。
3. 「了解」ボタンをクリックし、変更内容を適用してダイアログを閉じます。
4. 「ターゲットの編集」ダイアログの「構築」ボタンをクリックすると、新しい構築環境で構築が開始されます。

環境変数の表示と置き換え

「持続的環境変数」リストにある環境変数定義が、現在の構築プロセス環境の同名の環境変数の代わりに使用されます。現在の構築プロセス環境変数の定義を表示するには、「詳細」ボタンをクリックして「現在の環境」リストを開きます。このリストには、現在の構築プロセス環境で定義されているすべての環境変数が表示されます。

「フィルタ」フィールドを使用してリストにフィルタをかけることができます。

環境変数の値を置き換える手順は次のとおりです。

1. 「現在の環境」リストから環境変数を選択します。
2. 「<<追加」ボタンをクリックし、選択した環境変数を「持続的環境変数」リストに追加します。
3. 「値」フィールドに新しい値を入力し、「変更」ボタンをクリックします。
4. 「了解」ボタンをクリックし、変更内容を適用してダイアログを閉じます。
5. 「ターゲットの編集」ダイアログの「構築」ボタンをクリックすると、新しい値を使用して構築が開始されます。

構築エラーの修正

構築プロセスにテキストエディタを統合することにより、構築エラーの修正プロセスが容易になります。構築が失敗した場合は、「構築」ウィンドウの構築出力表示区画に構築エラーメッセージが表示されます。構築エラーメッセージには、そのエラーが含まれているファイルへのハイパーテキストリンク（下線付きで強調表示されます）があります（図 9-3 参照）。「構築」ウィンドウ内の下線付きのエラーをクリックするとテキストエディタが開き、そのエラーを含むソースファイルが表示されます。

各エラー行には、そのエラーが存在しているファイル名、エラーが発生した行番号、エラーメッセージが示されます。



図 9-3 構築出力表示区画に表示された構築エラー

C コンパイラによって出力されたエラーメッセージの構築エラーメッセージ部分には、情報アイコン () が表示されます。このアイコンをクリックすると、そのエラーに関する情報を示すポップアップウィンドウが表示されます

注 - 出力をハイパーテキストリンクに変換して生成できるのは、サンのコンパイラだけです。サンのコンパイラを呼び出さない構築コマンドを使用した場合、「構築」ウィンドウに表示された構築エラーからソースファイルへのリンクは作成されません。

エラーのソースの表示

下線が付いているエラーをクリックすると、テキストエディタが起動され、エラーが存在しているソースファイルが表示されます。ソースファイルのエラー行は強調表示され、その行の左側にエラーグリフが表示されます (図 9-4 参照)。

キーボードショートカットの F4 (次のエラー) および Shift+F4 (前のエラー) を使用することによって、テキストエディタのウィンドウにフォーカスを置いたまま構築エラー間を移動できます。

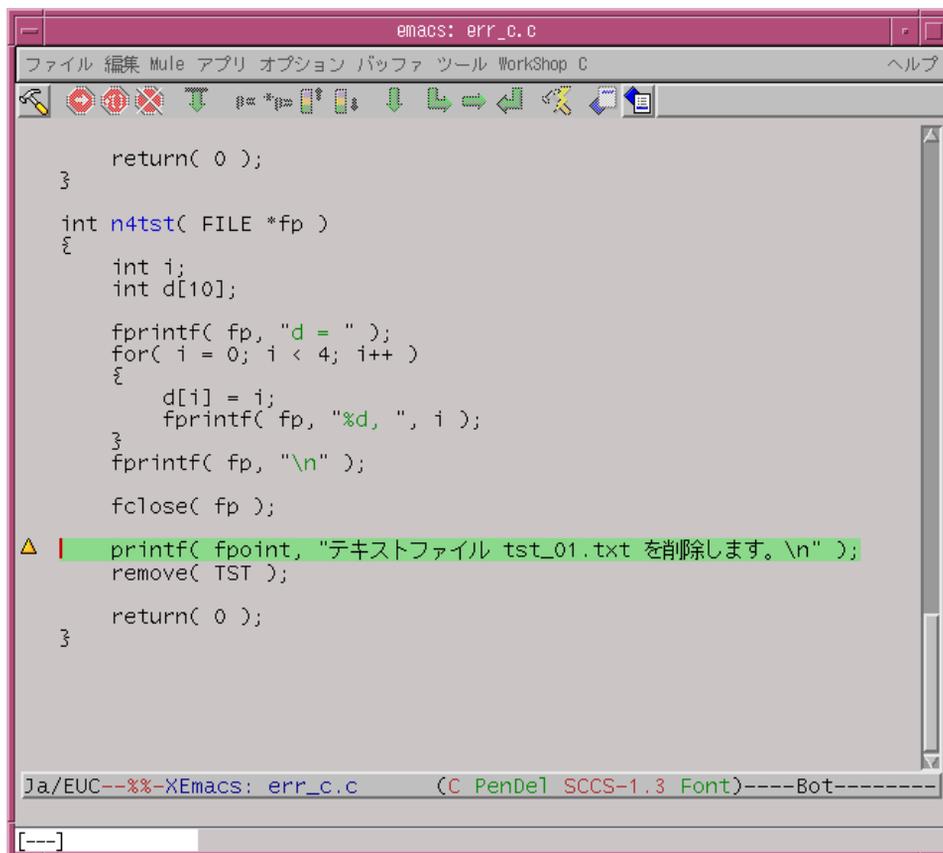


図 9-4 エラーが存在するソースファイルをテキストエディタのウィンドウに表示

エラーの修正

「構築」ウィンドウとエディタを使用して構築エラーを修正する手順は次のとおりです。

1. 「構築」ウィンドウの出力表示区画で強調表示されているエラーをクリックします。
エディタのウィンドウが開き、エラーが存在するソースファイルが表示されます。エラー行は強調表示され、カーソルがその行に置かれています。黄色で強調表示されているのが、現在のエラーです。
2. エラーのあるソースファイルを編集します。

3. 「構築」ウィンドウで「次のエラー」ツールバーボタンをクリック (または、キーボードショートカット F4 を使用) し、テキストエディタの次の構築エラーの場所に移動します。
「次のエラー」ボタンをクリックすると、構築出力の次のエラーが強調表示され、同時にテキストエディタ内の対応するソース行も強調表示されます。
4. 編集したファイルを保存します。
5. エディタウィンドウのツールバーの「構築」ボタンをクリックして、再構築します。
構築処理の進行状況は、構築の出力表示区画で監視できます。

構築の終了

現在の構築プロセスを強制終了して、すべての構築ウィンドウを閉じるには、「構築」ウィンドウから「構築」▶「構築終了」を選択します。

現在の構築処理を強制終了しないで構築ウィンドウを閉じるには、「構築」▶「閉じる」を選択します。

第10章

dmake ユーティリティの使用

この章では、分散メーク (`dmake`) ユーティリティで構築プロセスを複数のホストに分散し、複数のワークステーションやマルチプロセッサ上で並列的にプログラムを構築する方法について説明します。

この章は、次の節で構成されています。

- `dmake` の基礎
 - `dmake` ユーティリティについて
 - `dmake` ユーティリティがメークファイルに与える影響
 - `dmake` ユーティリティの使用方法
-

`dmake` の基礎

分散メーク (`dmake`) を使用して、多数のプログラムから構成される大規模なプロジェクトを構築する処理を、複数のワークステーション上、またはマルチプロセッサシステムの複数の CPU 上に分散できます。`dmake` ユーティリティは、メークファイルを解析して次のことを行います。

- どのターゲットを並行して構築できるか判断します。
- 複数のホストにそれらのターゲットの構築を分散します。

`dmake` ユーティリティは、`make` ユーティリティのスーパーセットです。

`dmake` を理解するためには、次の事柄についての知識が必要です。

- 構成ファイル (実行時構成ファイルおよび構築サーバーの構成ファイル)
- `dmake` ホスト
- 構築サーバー

構成ファイル

`dmake` ユーティリティは、どの構築サーバーにジョブを分散するか、また各サーバーにどれだけのジョブを分散するかを決定する際に、2つのファイルを参照します。

実行時構成ファイル

`dmake` ユーティリティは、`dmake` ホスト上の実行時構成ファイルから、ジョブの分散先に関する情報を得ます。通常、このファイルは `dmake` ホストのホームディレクトリに `.dmakerc` という名前で格納されています。このファイルには、構築サーバーおよび各構築サーバーに分散できるジョブ数が記述されています。詳細は、152 ページの「`dmake` ホスト」を参照してください。

構築サーバーの構成ファイル

それぞれの分散構築に使用する構築サーバーには、必ず `/etc/opt/SPROdmake/dmake.conf` ファイルが存在している必要があります。このファイルは、`dmake` ユーザーが全体として各構築サーバーに割り当てることができる `dmake` ジョブの最大数を指定します。また、すべての `dmake` ジョブの実行に適用される `nice` 優先度も指定できます。

詳細は、155 ページの「構築サーバー」を参照してください。

`dmake` ホスト

`dmake` ホストとは、`dmake` コマンドが最初に起動されるマシンです。`dmake` ユーティリティは、実行時構成ファイルの内容に基づいて、ジョブをどこに分散するかを決定します。一般に、このファイルは、`dmake` ホストのホームディレクトリ内に、`.dmakerc` という名前で置かれている必要があります。`dmake` ユーティリティは、以下の順序で実行時構成ファイルを探します。

1. コマンド行で `-c` オプションを使用して指定されたパス名
2. `DMAKE_RCFILE` という名のメイクファイルマクロを使用して指定されたパス名
3. `DMAKE_RCFILE` 環境変数を使用して指定されたパス名
4. `$(HOME)/.dmakerc`

実行時構成ファイルが見つからない場合、`dmake` ユーティリティは `dmake` ホストに 2 つのジョブを割り当てます。

実行時構成ファイルによって、各構築サーバーのリストと各構築サーバーに分散するジョブの数を指定できます。次に、`.dmakerc` ファイルの例を示します。

```
# My machine. This entry causes dmake to distribute to it.
falcon { jobs = 1 }
hawk
eagle { jobs = 3 }
# Manager's machine. She's usually at meetings
hero { jobs = 4 }
avocet
```

コード例 10-1 `dmakerc` ファイル

- `falcon`、`hawk`、`eagle`、`heron`、`avocet` は、構築サーバーのリストです。
- 構築サーバーごとに、分散するジョブ数を指定できます。デフォルトのジョブ数は 2 です。
- `#` の文字で始まる行は、コメントとして解釈されます。

注 - 上記のコード例の構築サーバーのリストには、`dmake` ホストでもある `falcon` が含まれています。`dmake` ホストも構築サーバーとして指定できます。実行時構成ファイルに `dmake` ホストを指定しなかった場合、`dmake` ホストには `dmake` ジョブは分散されません。

実行時構成ファイルには、構築サーバーのグループを設定することもできます。この方法によって、状況に応じて構築サーバーのグループを簡単に切り替えることができます。たとえば、異なるオペレーティングシステムでの構築処理ごとにユーザーが構築サーバーグループを定義したり、特殊なソフトウェアがインストールされている構築サーバーのグループを定義したりできます。

次に、構築サーバーのグループが定義されている実行時構成ファイルの例を示します。

```
earth { jobs = 2 }
mars  { jobs = 3 }

group lab1 {
    host falcon { jobs = 3 }
    host hawk
    host eagle { jobs = 3 }
}

group lab2 {
    host heron
    host avocet { jobs = 3 }
    host stilt { jobs = 2 }
}

group labs {
    group lab1
    group lab2
}

group sunos5.x {
    group labs
    host jupiter
    host venus { jobs = 2 }
    host pluto { jobs = 3 }
}
```

- グループであることが `group` 命令によって定義され、そのメンバーのリストが中括弧 ({}) で囲まれています。
- グループのメンバーになる構築サーバーは、`host` 指令によって指定されます。
- グループをほかのグループのメンバーにすることもできます。
- 構築サーバーのグループが指定されている実行時構成ファイルに、個別に構築サーバーを指定することもできます。この場合、`dmake` は、個別に指定された構築サーバーを名前が指定されていないグループのメンバーとして扱います。

`dmake` コーティリティは、次の優先順位に従ってジョブを分散します。

1. コマンド行で `-g` オプションの引数として指定されたグループ
2. `DMAKE_GROUP` という名のメークファイルマクロを使用して指定されたグループ
3. `DMAKE_GROUP` 環境変数を使用して指定されたグループ

4. 実行時構成ファイルで最初に指定されたグループ

`dmake` ユーティリティは、構築サーバーごとに異なる実行パスを指定できます。デフォルトでは、`dmake` は、`dmake` ホストと同じ論理パスを使用して、構築サーバー上の `dmake` サポートバイナリファイルを検索します。次のように、`.dmakerc` ファイルで構築サーバー用の別のパスをホスト属性として指定できます。

```
group lab1 {
    host falcon { jobs = 10 , path = "/set/dist/sparc-S2/bin" }
    host hawk   { path = "/opt/SUNWspro/bin" }
}
```

`.dmakerc` ファイルでは、グループ名およびホスト名を二重引用符で囲むことができます。文字列の一部として二重引用符を使用することで、グループおよびホストに付ける名前の制約が緩和されます。数字も英字と同様に使用できるようになります。数字で始まる名前は、次のように、二重引用符で囲む必要があります。

```
group "123_lab" {
    host "456_hawk" { path = "/opt/SUNWspro/bin" }
}
```

構築サーバー

構築サーバーのファイルシステム内に `/etc/opt/SPROdmake/dmake.conf` というファイルがあります。このファイルは、この構築サーバーでユーザーが全体として並行して実行できる `dmake` ジョブの最大数を指定するために使用します。また、すべての `dmake` ジョブの実行に適用される `nice` 優先度も指定できます。次に `/etc/opt/SPROdmake/dmake.conf` ファイルの例を示します。このファイルでは、1つの構築サーバーで実行可能な(すべての `dmake` ユーザーからの) `dmake` ジョブの最大数を、8 に設定しています。

```
max_jobs: 8
nice_prio: 5
```

注 - 構築サーバーに `/etc/opt/SPROdmake/dmake.conf` ファイルが存在しない場合、そのサーバーで、`dmake` ジョブを実行することはできません。

dmake ユーティリティについて

分散メークを実行するには、標準の `make` ユーティリティの代わりに、`dmake` を使用します。`dmake` を使用する前に、Solaris の `make` ユーティリティを理解しておく必要があります。`make` ユーティリティについての詳細は、Solaris の『プログラミング ユーティリティ』を参照してください。`make` ユーティリティを使用している場合、ほとんど変更なしで `dmake` を使用できます。

dmake ユーティリティがメークファイルに与える影響

この節では、`dmake` で解決可能な種類の問題を例に挙げて説明します。手続きが複雑になるにしたがって、それらを実装するメークファイルの内容も複雑になります。適切なメークファイルをもたらす方法を知る必要があります。この節では、一般的なコード開発の例を挙げ、`dmake` を使用して作業を簡略化する方法を説明します。

メークファイルテンプレートの使用

プロジェクトを開始する時点でメークファイルテンプレートを使用することがわかっている場合は、メークファイルテンプレートをカスタマイズして独自のメークファイルを作成すると、次の利点があります。

- 使い慣れたものになる
- わかりやすい
- 統合しやすい
- 保守が簡単
- 再利用が簡単

メークファイルの編集にかかる時間を節約し、プログラムやプロジェクトの開発により多くの時間を費やすことができます。

ターゲットの並列構築

一般的に、大規模なソフトウェア開発プロジェクトは、並列して構築できる独立した多数のモジュールで構成されています。dmake ユーティリティは、ネットワークにまたがった複数のマシン上でのターゲットの並列処理を行うことができます。こうした並列処理により、大規模な開発プロジェクトを構築するのに必要な時間が大幅に省けます。

dmake は、構築するターゲットが与えられると、そのターゲットに関連する依存関係を調べ、古くなっている依存関係を構築します。この依存関係の構築では、依存関係の依存関係の構築を伴うこともあります。ジョブを分散する際、dmake は開始可能なすべてのターゲットを開始します。それらのターゲットが終了すると、dmake はその他のターゲットを起動します。デフォルトでは、入れ子構造になった dmake の呼び出しは同時には実行されません。ただし、この設定は変更可能です (詳細は、160 ページの「並列処理に対する制限」を参照)。

dmake では複数のターゲットが並列に構築されるため、各構築プロセスの出力は同時に生成されます。各コマンドからの出力が混ざらないよう、dmake では各構築の出力を個別に収集します。また dmake は、コマンドを実行する前に、そのコマンドの内容を表示します。実行したコマンドによって出力、警告、またはエラーが生成されると、dmake ユーティリティは該当するコマンドの出力全体を表示します。後から開始したコマンドが先行するコマンドよりも先に終了することもあるので、出力の順序が予想と異なる場合があります。

メイクファイルに関する制限事項

複数のターゲットを同時に構築する場合は、メイクファイルに多少の制限が伴います。たとえば、暗黙の依存関係の順序に依存しているメイクファイルは、並列構築を行なったとき正しく機能しないことがあります。また、同じファイルを修正する複数のターゲットは、そのファイルが2つの異なるターゲットによって同時に変更されると失敗することがあります。この節では、こうした問題の例を取り上げます。

依存関係リスト

ターゲットを並列構築するときは、依存関係リストが正確であることが重要なポイントになります。たとえば、2つの実行可能ファイルが同じオブジェクトファイルを共有しているときに、一方のファイルでしか依存関係が指定されていないと、並列構築がエラーになる可能性があります。次のメイクファイルの例を考えてみます。

```
all: prog1 prog2
prog1: prog1.o aux.o
      $(LINK.c) prog1.o aux.o -o prog1
prog2: prog2.o
      $(LINK.c) prog2.o aux.o -o prog2
```

逐次構築では、ターゲット `aux.o` は `prog1` に依存するファイルとして構築され、`prog2` の構築時に更新されます。これに対し、並列構築では、`prog2` のリンクが、`aux.o` の構築前に開始されることがあり、その場合は、不正になります。`make` の `.KEEP_STATE` 機能で検出できる依存関係も一部ありますが、上記のような関係はその対象にはなりません。

依存関係リストにおける明示的な順序指定

次に、暗黙の順序にもとづく依存関係がある場合は、問題がさらに複雑になるという例を示します。たとえば、あるシステムのヘッダーをすべて作成してから構築を開始する場合は、すべての要素がこの構築に依存していなければなりません。このため、メイクファイルがさらに複雑になり、新しいターゲットをメイクファイルに追加した場合などにエラーが起きる可能性が高くなります。この場合は、メイクファイルの中で特殊な `.WAIT` ターゲットを指定して、暗黙の依存関係があることを明示できます。`dmake` は依存関係リストの中に `.WAIT` ターゲットを検出すると、これまでの依存関係の処理を終了してから、以降の依存関係の処理を実行します。1つの依存関係リストに、複数の `.WAIT` ターゲットを入れることもできます。次に、`.WAIT` を使用して、まずヘッダーを作成してから、以降の処理を続行する例を示します。

```
all: hdrs .WAIT libs functions
```

`.WAIT` ターゲットに対する空の規則をメイクファイルに追加することによって、`make` ユーティリティでも、このメイクファイルを使用できます。

ファイルの同時修正

ターゲットの構築プロセスでは、同時に同じファイルを修正しないように注意する必要があります。この問題は、さまざまな形で起きる可能性があります。一時ファイルを使用する新しい接尾辞の規則を定義する場合、一時ファイルの名前は、ターゲットごとに異ならなければなりません。この場合は、`$@` や `$*` の動的マクロを利用できます。たとえば、ソースファイルをコンパイルする前に `.c` ファイルに修正を加える場合、`.c.o` の規則は次のように定義できます。

```
.c.o:
    awk -f modify.awk $*.c > $*.mod.c
    $(COMPILE.c) $*.mod.c -o $*.o
    $(RM) $*.mod.c
```

ライブラリの同時更新

並列処理でもう 1 つの問題になる可能性があるのは、ライブラリを作成するときのデフォルトの規則です。すなわち、ライブラリという固定ファイルが変更される場合があるということです。本来適用されるべきでない `.c.a` 規則によって、`dmake` がそれぞれのオブジェクトファイルを構築し、このオブジェクトファイルをアーカイブする可能性があります。`dmake` が 2 つのオブジェクトファイルを同時にアーカイブすると、並列的に更新が行われてアーカイブファイルが壊れることがあります。

```
.c.a:
    $(COMPILE.c) -o $% $<
    $(AR) $(ARFLAGS) $@ $%
    $(RM) $%
```

この場合は、各オブジェクトファイルを構築し、その構築が完了した後、すべてのオブジェクトファイルをアーカイブするようにします。正しい接尾辞の規則と対応するライブラリの規則は次のとおりです。

```
.c.a:
    $(COMPILE.c) -o $% $<
    $(COMPILE.c) -o $% $<
lib.a: lib.a($(OBJECTS))
    $(AR) $(ARFLAGS) $ (OBJECTS)
    $(RM) $ (OBJECTS)
```

複数のターゲット

ファイルの同時更新の問題は、複数のターゲットに対して同じ規則を定義した場合にも起きます。たとえば、プログラムと `lex(1)` で使用されるヘッダーの両方を構築する `yacc(1)` というプログラムがあります。1つの規則にもとづいて複数のターゲットファイルを構築する場合は、`+` 表記を使用して各ファイルを1つのグループとして指定する必要があります。特に、並列構築の場合に、このことが重要になります。

```
y.tab.c y.tab.h: parser.y
$(YACC.y) parser.y
```

この規則は、次のように2つの規則が指定された場合と同等になります。

```
y.tab.c: parser.y
$(YACC.y) parser.y
y.tab.h: parser.y
$(YACC.y) parser.y
```

逐次処理の `make` は、最初の規則に基づいて `y.tab.c` を作成してから、`y.tab.h` が最新であり、構築の必要がないと判断します。これに対し、並列構築では、`yacc` が `y.tab.c` の構築を完了する前に `y.tab.h` を調べるため、`y.tab.h` の構築が必要であると判断し、最初の `yacc` と並行して別の `yacc` を起動します。両方の `yacc` が同じファイル (`y.tab.c` と `y.tab.h`) に書き込みを行うため、これらのファイルが壊れ、不正になる可能性があります。正しい規則では、次の例に示すように、`+` 構文を使用して、両方のターゲットを同じ規則に従って同時に作成するように指示します。

```
y.tab.c + y.tab.h: parser.y
$(YACC.y) parser.y
```

並列処理に対する制限

1つのメイクファイルの中でファイルの衝突が避けられない場合があります。その1つの例が、Cプログラムから文字列を抽出して、共有文字列を生成する `xstr(1)` コマンドです。`xstr` コマンドでは、変更されたCプログラムは `x.c` という固定した

ファイルに書き込まれ、抽出した文字列は `strings` という固定ファイルに追加されます。`xstr` は C ファイルごとに実行する必要があるため、次の新しい `.c.o` 規則がよく定義されます。

```
.c.o:
    $(CC) $(CPPFLAGS) -E $*.c | xstr -c -
    $(CC) $(CFLAGS) $(TARGET_ARCH) -c x.c
    mv x.o $*.o
```

この場合、各ターゲットを作成するごとに、`x.c` と `strings` という同じファイルに書き込みを行うので、`dmake` ユーティリティは上の規則にもとづいてターゲットを同時に構築することはできません。また使用するファイルを変更することもできません。この場合は、`.NO_PARALLEL:` という特殊なターゲットを使用して、`dmake` に各ターゲットを同時に構築しないように指示できます。たとえば、`.c.o` 規則を使用して構築するオブジェクトを `OBJECTS` マクロで定義している場合は、次のようにして、各ターゲットを逐次構築するよう `dmake` に指示します。

```
.NO_PARALLEL: $(OBJECTS)
```

大部分のオブジェクトを逐次構築する必要がある場合は、依存関係リストを持たない `.NO_PARALLEL:` ターゲットをメークファイルに含めることによって、全オブジェクトをデフォルトで強制的に逐次処理するようにすると安全で簡単です。並列構築できるターゲットは、次のように `.PARALLEL:` ターゲットに依存するファイルとして記述できます。

```
.NO_PARALLEL:
.PARALLEL: $(LIB_OBJECT)
```

分散メークの入れ子呼び出し

`dmake` は、ほかの `dmake` コマンドを呼び出すターゲットを検出した場合、そのターゲットを並列構築しないで、逐次構築します。これによって、2つの異なる `dmake` が同じディレクトリ中の同じターゲットを構築しようとするという問題を回避できます。この問題は、同じライブラリにアクセスする必要がある2つの異なるプログラムが並列構築された場合に起こります。各 `dmake` 呼び出しでライブラリが最新であるこ

とを確認できる唯一の方法として、`dmake` を再帰的に呼び出してライブラリを構築できます。コマンド行で `$(MAKE)` マクロを指定した場合のみ、`dmake` ユーティリティは入れ子呼び出しを認識できます。

入れ子にしたコマンドが衝突を起こさないと分かっている場合は、`.PARALLEL:` を使用して、強制的に並列構築が行われるように設定できます。

メイクファイルの中に、並行して実行される入れ子のコマンドが多数含まれている場合、負荷均衡アルゴリズムが、強制的にローカルマシンに構築処理を過剰に割り当ててしまうことがあります。これによって、負荷が大きくなったり、スワップ領域が不足するなどの問題を引き起こすことがあります。こうした問題が発生した場合は、入れ子にしたコマンドを逐次実行するようにしてください。

dmake ユーティリティの使用法

`dmake` ホストで `dmake` コマンドを実行すると、構築サーバーにジョブが分散されます。また、`dmake` ホストにジョブを分散することもできます。この場合、`dmake` ホストも構築サーバーとみなされます。`dmake` ユーティリティは、(メイクファイルをもとにして) 並行構築が可能であると判断したターゲットに基づいて、ジョブを分散します。次の条件を満たしているマシンを、構築サーバーとして使用できます。

- `dmake` ホスト (`dmake` を実行するために使用しているマシン) からは、パスワードを入力しなくても、`rsh` を使用して、リモートから構築サーバー上でコマンドを実行できる必要があります。`rsh` コマンドに関する詳細は、`rsh(1)` のマニュアルページを参照してください。`rsh` コマンドの使用例を次に示します。

```
demo% rsh build_server which dmake
/opt/SUNWspro/bin/dmake
```

- `dmake` がインストールされている `bin` ディレクトリに、構築サーバーからアクセスできる必要があります。すべての構築サーバーが共通の `dmake` インストールディレクトリを共有するようにしてください。共有ファイルシステムの作成についての詳細は、`share(1M)` および `mount(1M)` のマニュアルページ、またはオペレーティングシステムの AnswerBook を参照してください。

- デフォルトでは、`dmake` は、`dmake` ホストと構築サーバーの `dmake` 実行可能ファイルのパスが同じであることを前提に処理を行います。この設定を変更する場合は、実行時構成ファイルで、ホストの属性としてパス名を指定します。たとえば、次のようにパスを指定します。

```
group sparc-cluster {
    host wren    { jobs = 10 , path = "/export/SUNWspro/bin" }
    host stimpay { path = "/opt/SUNWspro/bin" }
}
```

- 構築を行うソース階層は、次の条件を満たす必要があります。
 - 構築サーバーからアクセスできる。
 - 同じ名前でもマウントされている。

`dmake` ホストから、使用する構築サーバーと各構築サーバーに割り当てる `dmake` ジョブの数を制御できます。特定の構築サーバーで実行可能な `dmake` ジョブの数は、その構築サーバー上でも制限できます。

`-m` オプションと `parallel` 引数を指定した場合、または `DMAKE_MODE` 変数またはマクロの値として `parallel` を設定している場合、`dmake` は、実行時構成ファイルをスキャンしません。このため、ジョブの数は、`-j` オプション、`DMAKE_MAX_JOBS` 変数またはマクロを使用して指定する必要があります。このどの方法でもジョブ数を指定しなかった場合は、デフォルトのジョブ数の 2 が使用されます。

`dmake` を分散モードで使用している時に、`-j` オプション、`DMAKE_MAX_JOBS` 変数またはマクロを使用してジョブの最大数を変更した場合、指定した値が実行時構成ファイルで設定されている値の代わりにしようされます。指定した値は、すべての構築サーバーへ分散可能なジョブ数の合計値として使用されます。

「構築」ウィンドウから `dmake` を使用する場合は、オンラインヘルプで構築サーバーとジョブ数の指定方法についての説明を参照してください。コマンド行から `dmake` を使用する場合は、`dmake(1)` のマニュアルページ (`dmake.1`) を参照してください。

第11章

Sun WorkShop TeamWare の ショートカット

Sun WorkShop TeamWare には、簡単に利用できるように、いくつかのショートカットが用意されています。この章は、次のトピックで構成されています。

- コマンド行からの TeamWare の使用方法
 - ワークスペース管理ツールのコマンド
 - バージョン管理ツールのコマンド
 - ファイルマージツールのコマンド
 - フリーポイントツールのコマンド
- GUI のショートカット

コマンド行からの TeamWare の使用方法

Sun WorkShop TeamWare ツールは、グラフィカルユーザーインターフェース (GUI) からだけでなく、システムプロンプトで TeamWare のコマンドを入力して利用することもできます。GUI およびコマンド行のどちらからでも Sun WorkShop TeamWare コマンドを利用することができ、両者の間に機能上の違いはありません。同時にある機能では GUI を利用して、別の機能ではコマンド行を利用するといったこともできます。スクリプトやプログラムにこれらのコマンドを含めて、ファイル管理をさらに自動化することもできます。

ワークスペース管理ツールのコマンド

ワークスペース管理ツールのコマンドの一覧を表示するには、`codemgr` と入力します。

```
example% codemgr
codemgr bringover ...
codemgr codemgrtool
codemgr help
codemgr putback ...
codemgr resolve ...
codemgr workspace ....
codemgr ws_undo ....
```

実際には、このリストには、コマンドオプションが角括弧で囲まれた、各コマンドの構文が示されます。たとえば、次は `bringover` コマンドの構文です。

```
% bringover [-w 子ワークスペース] [-p 親ワークスペース] [-c コメント]
[-m コメントファイル] [[-f flp]...] [-n] [-g] [-q] [-v] [-B] [-C]
[-i 取り込みファイル] [-x 除外ファイル] [ファイルおよびディレクトリ]
```

次は、コマンド行から `bringover` コマンドを発行している例です。

```
% bringover -w my_child -p their_parent /usr/ws/project
```

Sun WorkShop TeamWare ツールには、環境変数ファイルや、以前に指定された引数を格納した引数ファイルなどの、長いコマンド行の入力の手間を省く機能がいくつか用意されています。コマンドとそのコマンドで使用できるオプションの詳細リストを表示するには、次に示すように、`man commandname` の形式でコマンドを入力します。

```
% man bringover
```

注 – Sun WorkShop 製品のマニュアルページは、標準のディレクトリの `/usr/share/man` にはインストールされません。このため、Sun WorkShop TeamWare のマニュアルページにアクセスするには、`MANPATH` 環境変数の設定を変更する必要があります。xviii ページの「Sun WorkShop 開発ツールへのアクセス方法」を参照してください。

表 11-1 に、ワークスペース管理ツールでよく使用されるメニュー項目と対応するコマンドをまとめます。

表 11-1 「ワークスペース管理」のメニュー項目と対応するコマンド

GUI メニュー項目	対応するコマンド
ワークスペース作成	<code>workspace create</code>
名前変更	<code>workspace move</code>
親の変更	<code>workspace parent</code>
作成ブリングオーバー	<code>bringover</code>
更新ブリングオーバー	<code>bringover</code>
プットバック	<code>putback</code>
取り消し	<code>ws_undo</code>
衝突解決	<code>resolve</code>

特定のワークスペースをコマンド行コマンドのデフォルトとして設定するには、`CODEMGR_WS` 環境変数を設定します。51 ページの「環境変数の設定」を参照してください。

バージョン管理ツールのコマンド

コマンド行からバージョン管理ツールを起動するには、`twversion` に続けてアンパサンド (&) を入力します。

```
demo% twversion &
```

ファイルマージツールのコマンド

ファイルを読み込まずに、コマンド行からファイルマージツールをバックグラウンドで起動するには、シェルコマンドプロンプトに対し、`twmerge` に続けてアンパサンド (&) を入力します (ファイルマージツールの実行可能ファイルが、指定した検索パスにある場合)。

```
demo% twmerge &
```

この `twmerge` コマンドは、ファイルを読み込まずにバックグラウンドでファイルマージツールを起動します。各区画にファイルを読み込んでファイルマージツールを起動する方法については、168 ページの「起動時に 2 つのファイルを読み込む」を参照してください。

次は `twmerge` コマンドの構文です。コマンドオプションは、角括弧で囲まれています。

```
twmerge [-b] [-r] [-tabsize 値] [-diffopt [bwi]]
[-a 祖先] [-f1 名前] [-f2 名前] [-l リストファイル]
[左のファイル [ 右のファイル [ 出力ファイル ] ] ] [-V]
```

起動時に 2 つのファイルを読み込む

コマンド行からファイルマージツールを起動するときに 2 つのファイルを読み込むには、それらのファイルが存在するディレクトリに移動して、コマンド行にそれらのファイル名を指定します。たとえば、`file_1` と `file_2` という 2 つのファイルのマージするには、次のコマンドを使用します。

```
demo% twmerge file_1 file_2 &
```

1 つ目のファイルが左のテキスト区画、2 つ目のファイルが右のテキスト区画にそれぞれ表示されます。

起動時に 3 つのファイルを読み込む

`ancestor_file` という名前の共通の祖先ファイルと比較して、2 つのファイルのマージするには、それらのファイルが存在するディレクトリに移動して、次のコマンドを使用します。

```
demo% twmerge -a ancestor_file file_1 file_2 &
```

祖先ファイルそのものは表示されませんが、祖先ファイルと 2 つの子孫ファイル間の相違にマークが付けられ、祖先ファイルに基づいて、マージ結果がファイルに出力されます。

フリーズポイントツールのコマンド

コマンド行からフリーズポイントを作成、更新、抽出できます。

フリーズポイントコマンドのリストを表示するには、`freezept` と入力します。

```
example% freezept
freezept compare ...
freezept create ...
freezept diff ...
freezept extract ...
freezept help
freezept sid ...
freezept smid ...
freezept update ....
```

コマンド行からのフリーズポイントの使用方法については、`freezept` のマニュアルページを参照してください。マニュアルページを表示するためのコマンドは次のとおりです。

```
% man freezept
```

GUI のショートカット

Sun WorkShop TeamWare インタフェースには、いくつかのショートカットが組み込まれています。ショートカットは、ワークスペースやディレクトリ、ファイルの選択で利用できます。

- アイコンをクリックして、ワークスペース、ファイル、ディレクトリを選択する。Shift キーを押しながらクリックすることによって、複数のワークスペース、ファイル、ディレクトリをまとめて選択することもできます。
- ダイアログボックス上の項目が何もない部分でマウスの左ボタンを押して、ドラッグし、選択するアイコンを長方形で囲むことによって、ファイルをまとめて選択する。ボタンを離れたときに長方形で囲まれていたすべてのファイルが、選択状態になります。
- ファイルまたはディレクトリのアイコンをクリックして、選択状態にしてから、別のアイコン上にポインタを置いて、マウスの中央ボタンをクリックすることによって、ファイルやディレクトリを追加選択する。

表 11-2 に、Sun WorkShop TeamWare のその他のマウスおよびキーボードショートカットをまとめています。

表 11-2 マウスおよびキーボードショートカット

操作	動き	詳細説明
項目の何もない部分にワークスペースのアイコンをドラッグ & ドロップ	「作成プリングオーバー」タブを表示	
子ワークスペースのアイコン上に親ワークスペースのアイコンをドラッグ & ドロップ	「更新プリングオーバー」タブを表示	
親ワークスペースのアイコン上に子ワークスペースのアイコンをドラッグ & ドロップ	「プットバック」タブを表示	
親子関係のないワークスペースのアイコン上に別のワークスペースのアイコンをドラッグ & ドロップ	プリングオーバーまたはプットバックを行うかどうかを問い合わせるポップアップダイアログボックスを表示	
Control キーを押しながら、ワークスペースのアイコン上に別のワークスペースのアイコンをドラッグ & ドロップ	親を変更	42 ページの「ワークスペースの親子関係の変更」を参照
Control キーを押しながら、項目の何もない部分にワークスペースをドラッグ & ドロップ	ワークスペースを親から切り離す	42 ページの「ワークスペースの親子関係の変更」を参照
ワークスペースのアイコンの名前フィールドをクリック	ワークスペース名を変更	
「ワークスペース管理」でマウスの右ボタンをクリック	「ファイル」、「ワークスペース管理」、「トランザクション」メニューを結合したメニューを表示	

表 11-2 マウスおよびキーボードショートカット (続き)

操作	働き	詳細説明
「ワークスペース管理」の項目のない部分で Props キーを押す	「ツール属性」ダイアログボックスの「ワークスペース管理」タブを表示	48 ページの「ツール属性によるワークスペース管理ツールのカスタマイズ」を参照
「ワークスペース管理」でワークスペースを選択した状態で Props キーを押す	「ワークスペース属性」ダイアログボックスを表示	48 ページの「ツール属性によるワークスペース管理ツールのカスタマイズ」を参照
「ワークスペース管理」でワークスペースのアイコンをダブルクリック	ツールを起動 (起動するツールはユーザーが設定可能で、デフォルトはバージョン管理ツール)	172 ページの「ワークスペース管理ツールでのダブルクリック時の動作」を参照
衝突を含むワークスペースのアイコンをダブルクリック	ツールを起動 (起動するツールはユーザーが設定可能で、デフォルトは「衝突解決」ウィンドウ)	172 ページの「ワークスペース管理ツールでのダブルクリック時の動作」を参照
「バージョン管理」でファイルのアイコンをダブルクリック	ファイルをチェックアウトするか、ファイルの履歴を表示 (どちらにするかはユーザーが設定可能)	172 ページの「ワークスペース管理ツールでのダブルクリック時の動作」を参照
「バージョン管理」のファイル履歴でデルタをダブルクリック	デフォルトのエディタのウィンドウにそのデルタを表示	172 ページの「ワークスペース管理ツールでのダブルクリック時の動作」を参照
「バージョン管理」でマウスの右ボタンをクリック	ファイルの選択とコマンドメニューを結合したメニューを表示	
「バージョン管理」のファイル履歴でマウスの右ボタンをクリック	「バージョン」メニューと「枝のマージ」を結合したメニューを表示	96 ページの「ファイルの履歴の表示」を参照
「ファイルマージ」でマウスの右ボタンをクリック	「ナビゲート」メニューを表示	83 ページの「相違の解決」を参照

ワークスペース管理ツールでのダブルクリック時の動作

ワークスペースのアイコン上にポインタを置いて、マウスをダブルクリックすると、TeamWare のバージョン管理ツールが自動的に起動し、選択したワークスペースが自動的に読み込まれます。未解決の衝突が存在するワークスペースのアイコン上にポインタを置いて、ダブルクリックすると、ワークスペース管理ツールによって自動的に「衝突解決」タブが表示されます。選択状態のワークスペースから衝突しているファイルが自動的に読み込まれ、すぐに処理を行うことができます。ワークスペース管理ツールにでもダブルクリック時の動作は、「ツール属性」ダイアログボックスを使用してカスタマイズできます (図 3-7 を参照)。

バージョン管理ツールでのダブルクリック時の動作

「バージョン管理」ウィンドウでファイルをダブルクリックすると、デフォルトでは、ファイルの 2 つの状態が交互に切り替わります。そのファイルがチェックアウトされていない場合は、チェックアウトされ、すでにチェックアウトされている場合は、チェックインされます。バージョン管理ツールの「一般」タブを使用して、ダブルクリックでファイルの履歴が表示されるように設定することもできます。「一般」タブを表示するには、「表示」▶「オプション」を選択します。108 ページの「バージョン管理ツールの属性の設定」を参照してください。

ファイル履歴でのダブルクリック時の動作

バージョン管理ツールの「ファイル履歴」ウィンドウでデルタをダブルクリックすると、デフォルトのエディタにそのデルタが表示されます。

第12章

Sun WorkShop TeamWare のアーキテクチャー

この章では、Sun WorkShop TeamWare がワークスペースとファイルを追跡するのに使用する、基本となるファイルについて説明します。また、ワークスペース間でファイルがコピーされたり衝突が解決されたりしたときに、ワークスペース管理ツールがどのように SCCS 履歴ファイルを操作するかについても説明します。

この章は、次の節で構成されています。

- ワークスペースのメタデータディレクトリ
- ワークスペース管理ツールのデフォルト値ファイル
- ワークスペース管理ツールによるファイルの マージ
- SCCS マージ可能 ID について

ワークスペースのメタデータディレクトリ

ワークスペース管理ツールのワークスペースは、その最上位ディレクトリに [Codemgr_wsdata](#) というディレクトリを含むディレクトリです。ワークスペース管理ツールは、[Codemgr_wsdata](#) ディレクトリにワークスペースに関するデータ (メタデータ) を記録します。ワークスペース管理ツールのコマンドは、[Codemgr_wsdata](#) ディレクトリの有無に基づいて、ディレクトリがワークスペースであるかどうかを判断します。

ワークスペース管理ツールには、[Codemgr_wsdata](#) ディレクトリ内に格納されている情報を管理するために必要なツールが用意されています。推奨しませんが、特定のファイルを手作業で編集することが必要になる場合があります。ただし、その場合も、編集する個々のファイルの形式を保つように注意する必要があります。表 12-1

に、[Codemgr_wsdata](#) ディレクトリに含まれるファイルとディレクトリの概要を示します。これらのファイルの形式については、各ファイルのマニュアルページを参照してください。

表 12-1 [Codemgr_wsdata](#) メタデータディレクトリの内容

ファイル / ディレクトリ名	説明
access_control	ワークスペースに対してどのユーザーがワークスペース管理ツールのトランザクションとコマンドを実行できるようにするかを制御する情報が含まれています。ワークスペースが作成されると、デフォルトのアクセス制御ファイルが作成されます。第 4 章および 177 ページの「 access_control ファイル」を参照してください。
args	ファイルやディレクトリ、FLP 引数のリストを含んでいます。ワークスペース管理ツールのプリングオーバーおよびブットバックトランザクションコマンドによって管理されます。最初、このファイルには、ワークスペースが作成時に指定された引数が含まれます。以降のプリングオーバーまたはブットバックトランザクションでユーザーが引数を明示的に指定すると、 args ファイル内にすでに存在する引数よりも新しい引数が広範囲かどうか判断され、より広範囲である場合は、古い引数は新しい引数に置き換えられます。38 ページの「独自のプリングオーバー / ブットバック用ファイルリストの作成」を参照してください。
backup/	ワークスペース管理ツールがプリングオーバーやブットバックトランザクションを取り消すときに使用する情報を保存します。25 ページの「ワークスペースに対する変更の取り消し」を参照してください。
children	ワークスペースの子ワークスペースのリストです。作成プリングオーバートランザクションを実行すると、このファイルに子ワークスペース名が書き込まれます。ワークスペース管理ツールは、このファイルを参照して、子ワークスペースのリストを取得します。ワークスペースを削除、移動したり、ワークスペースの親を変更したりすると、親の children ファイルが更新されます。
conflicts	ワークスペース内で、現在、衝突状態にあるファイルのリストです。衝突とその解決方法については、第 6 章を参照してください。

表 12-1 Codemgr_wsdata メタデータディレクトリの内容 (続き)

ファイル / ディレクトリ名	説明
description	ユーザーがワークスペースに付けた名前と詳細な説明が含まれるファイルです。41 ページの「ワークスペースに分かりやすい名前を付ける」を参照してください。
Freezepoints/	自動フリーズポイントを設定したときに作成されます。124 ページの「フリーズポイントの自動作成」を参照してください。
history	ワークスペースに関するトランザクションおよび更新ファイルの履歴です。27 ページの「ワークスペースの履歴表示」を参照してください。
locks	プリングオーバー、ブットバック、取り消しトランザクションの実行中に、ワークスペース管理ツールはワークスペースをロックして、その整合性を保ちます。このロック情報は、各ワークスペースの locks ファイルに記録されます。ワークスペース管理ツールは、ワークスペースに対する処理を開始する前に、このファイルを参照し、ロックの有無を確認します。65 ページの「ワークスペースのロックの解除」を参照してください。
nametable	SCCS ファイル名 (ワークスペースを基準にした相対パス名) と 4 つの 32 ビット 16 進ワードで表現された一意の番号からなるテーブルです。このテーブル内の各エントリは改行文字で区切られています。ワークスペース管理ツールは、プリングオーバーおよびブットバックでこのファイルを使用し、名前が変更されたファイルの処理が高速に行われるようにします。このファイルが見つからない場合、ワークスペース管理ツールは次のブットバックまたはプリングオーバートランザクションで自動的にファイルを作成します。101 ページの「ファイル名の変更とファイルの移動、削除」を参照してください。

表 12-1 Codemgr_wsdata メタデータディレクトリの内容 (続き)

ファイル / ディレクトリ名	説明
<code>notification</code>	ワークスペース管理ツールは、このファイルを参照することによって、ワークスペースに関係するイベントを検出し、そのイベントに対する応答として電子メールを送信します。40 ページの「他のユーザーへのトランザクションの実行通知」を参照してください。
<code>parent</code>	作成プリングオーバートランザクションで作成される、ワークスペースの親ワークスペース名が含まれるファイルです。また、ワークスペースがワークスペース作成コマンドで作成され、親を持っていない場合は、親の変更コマンドを実行したときにも作成されます。ワークスペース管理ツールは、このファイルを参照してワークスペースの親を認識します。ワークスペースを削除や移動、またはワークスペースの親を変更すると、子の <code>parent</code> ファイルが更新されます。
<code>putback.cmt</code>	最後にブロックされたプットバックトランザクションのコメントを一時的に保持するキャッシュです。プットバックトランザクションがブロックされると、コメントが廃棄されます。ワークスペース管理ツールはこのファイルにコメントを書き込むことによって、トランザクションを再実行したときに元のテキストを取り出せるようにします。

ワークスペース管理ツールのデフォルト値ファイル

「ツール属性」ウィンドウを使用してワークスペース管理ツールの動作の設定を変更して、「適用」ボタンをクリックすると、その変更をホームディレクトリの実行時構成ファイルに保存できます。ワークスペース管理ツールは、起動されたときにその実行時構成ファイルを参照し、その中の新しい設定をデフォルト値として使用します。

「ツール属性」ウィンドウの「ワークスペース管理」や「プリングオーバー/プットバック」タブで行なった変更は、`~/ .codemgrtoolrc` ファイルに書き込まれます。このファイルは、XWindows のリソースファイルです。

「ツール属性」ウィンドウの「衝突解決」タブで行なった変更は、
~/.codemgr_resrc という実行時構成ファイルに書き込まれます。

環境変数を使用して、デフォルトで読み込むワークスペースを制御できます (51 ページの「環境変数の設定」を参照)。デフォルトでプリングオーバー / プットバックするファイルのリストを作成することもできます (38 ページの「独自のファイルリストの作成」を参照)。

access_control ファイル

第 4 章では、ワークスペースに対してどのユーザーがどのトランザクションを実行できるかを制御する方法を説明しました。「アクセス制御」タブでアクセス権を設定すると、`access_control` ファイルにその情報が書き込まれます。表 12-2 は、ワークスペースを作成したときの `access_control` ファイルのデフォルトの内容を示しています。

表 12-2 デフォルトのアクセス権設定

トランザクション	デフォルトのアクセス権
bringover-from	全ユーザー
bringover-to	作成者
putback-from	全ユーザー
putback-to	全ユーザー
undo	全ユーザー
workspace-delete	作成者
workspace-move	作成者
workspace-reparent	作成者
workspace-reparent-to	全ユーザー

`access_control` ファイルを手作業で編集し、ワークスペースに対するアクセスを許可するユーザーを制御できます。表 12-3 に、ワークスペースに対するアクセスを制御するために指定できるすべての値と、その意味をまとめます。

表 12-3 ワークスペースに対するアクセス制御値

値	意味
@engineering	ネットグループ <code>engineering</code> のユーザー全員が、このトランザクションを実行できます。
~@engineering	ネットグループ <code>engineering</code> のユーザーは、このトランザクションを実行できません。 "~" は否定を意味します。
@special ~user2 @engineering	ネットグループ <code>special</code> および <code>engineering</code> のユーザーは、このトランザクションを実行できます。ただし、 <code>user2</code> は実行できません (<code>user2</code> が <code>special</code> ネットグループに属していない場合)。 "~" は否定を意味します。
user1 user2	ユーザー <code>user1</code> と <code>user2</code> は、このトランザクションを実行できます。
"~"	どのユーザーも、このトランザクションを実行できません。
作成者	ワークスペースの作成者だけ、このトランザクションを実行できます。実際には、作成者のログイン名が示されます。
(エントリなし)	ユーザー全員がこのトランザクションを実行できます。

ワークスペース管理ツールによるファイルのマージ

この節では、分岐の概念などを含め、読者が SCCS について理解していることを前提に説明をしています。分岐については、99 ページの「ファイルの履歴 (デルタ、分岐、バージョン)」を参照してください。SCCS についての詳細は、Solaris の『プログラミングユーティリティ』を参照してください。

ブリングオーバーやブットバックトランザクションでは、転送元（ソース）ファイルは SCCS デルタから得られるファイルで、SCCS デルタ ID (SID) によって識別されます。これらのトランザクションによってファイルがコピーされた場合、ワークスペース管理ツールプログラムは、そのファイルの SCCS 履歴ファイル (s ドットファイル) を処理する必要があります。

ブリングオーバーやブットバックトランザクションによって転送元ワークスペースから転送先ワークスペースにファイルをコピーした場合、その処理は、表面的にはファイルが 1 つ転送されただけのように見えます。しかし、実際には、そのファイルに関するすべての SCCS 情報（デルタやコメントなど）が、転送先の SCCS 履歴ファイルにマージされます。転送元の情報を転送先の履歴ファイルにマージすることによって、現在のバージョン（デルタ）を作成して、ファイルのデルタとコメント履歴全体を利用できるようになります。ただし、これはファイルが転送先ワークスペースに存在する場合に限ります。存在しない場合は、転送元ワークスペースの履歴ファイルのすべての全内容が転送先ワークスペースにコピーされます。

衝突していないファイルのマージ

転送先ワークスペースのファイルを更新する場合（つまり、ブリングオーバーまたはブットバックトランザクションの転送元のファイルが変更されていて、転送先では変更されていない場合）は、新しいデルタが転送先の履歴ファイルに追加されます。この場合、SCCS 履歴ファイルを転送元から転送先にコピーせずに、マージするのは、転送先の履歴ファイルに格納されているフラグやアクセスリストなどの管理情報が上書きされるのを防ぐためです。

ファイルマージを行う際、ワークスペース管理ツールはデルタ履歴の分岐箇所を判別し、その分岐の発生以後に転送元ワークスペースで作成されたデルタのみを転送先ワークスペースに追加します。履歴の分岐箇所を特定するために、親と子の履歴ファイルのデルタテーブルを比較します。この比較に使用される情報には、該当デルタがいつ誰によって作成されたかといったデータやコメントが含まれています。

図 12-1 に、ブットバックトランザクションで子ワークスペースのデルタ 1.3 と 1.4 を親ワークスペースの SCCS 履歴ファイルに追加する例を示します。

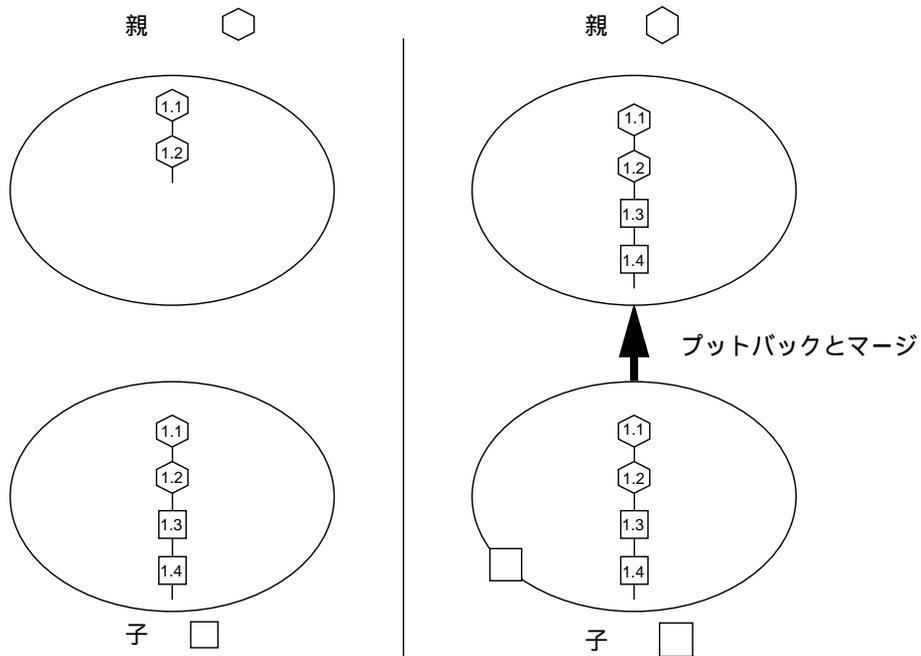


図 12-1 転送先ワークスペース内の変更されていないファイルの更新

衝突しているファイルのマージ

親と子の中でファイルをコピーする場合、それぞれのファイルのバージョンが最後に更新されてから変更されていることがあります。このような場合、親と子のファイルが「衝突」していると言います。

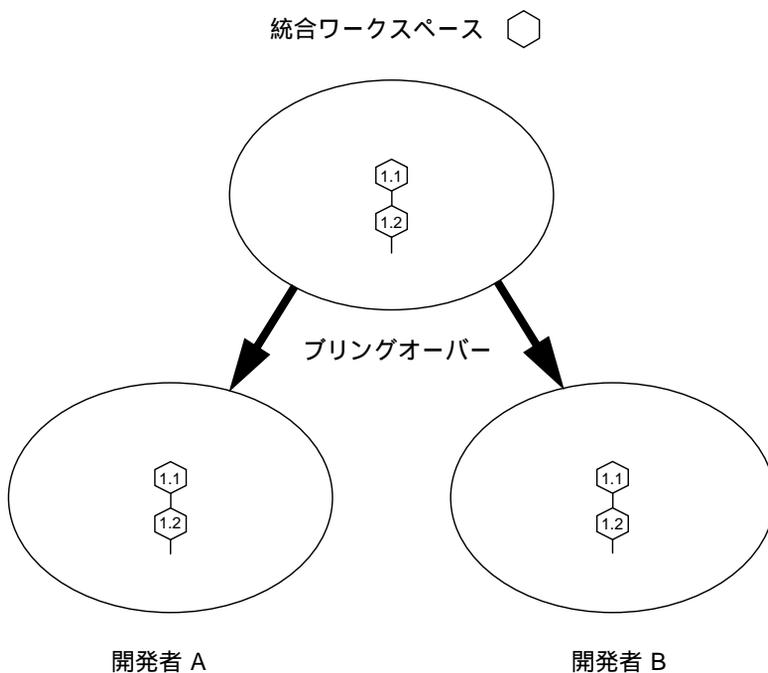
ファイルが衝突していると、ワークスペース管理ツールはそれぞれのファイルに加えられた変更の矛盾をユーザーが効率的に解決できるように支援し、それらのファイルのデルタ、管理情報、コメント履歴を保持しようとします。ワークスペース管理ツールは、親の SCCS デルタを子の履歴ファイルにマージします。次に、衝突解決トランザクションによって、子の側でこの衝突を解決します。衝突解決についての詳細は、77 ページの「ワークスペース内での衝突の解決」を参照してください。

ファイルマージによるデルタの追跡方法

この節では、1つの統合ワークスペースとそれぞれ別の開発者が所有する2つの子ワークスペースを使用した例を紹介します。2人の開発者はそれぞれ、統合ワークスペースから同じファイルをコピーし、このファイルに独自の変更を加えています。この例で、衝突が発生したときに、それらを解決するためにワークスペース管理ツールがSCCS ファイルをどのように処理するかを説明します。説明図については、次の点に留意してください。

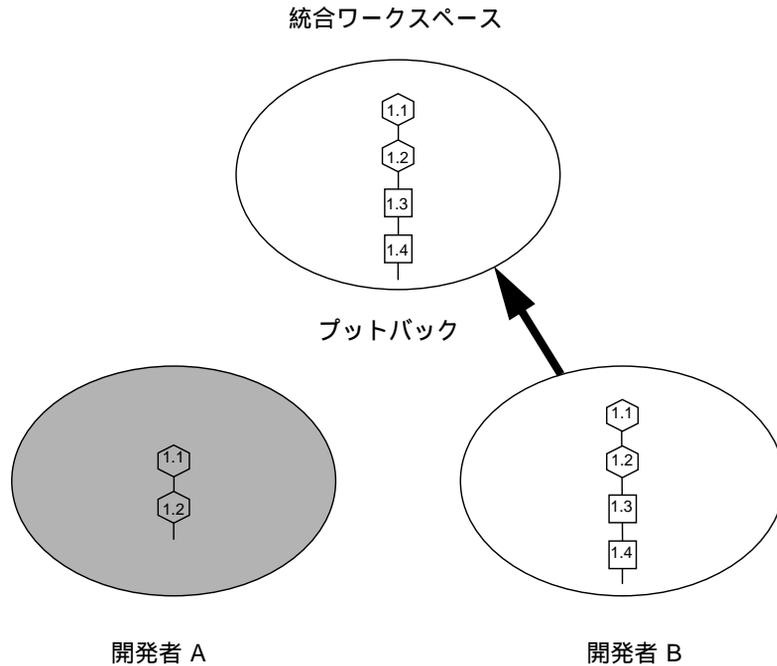
- デフォルトのデルタ (次のデルタが SCCS デルタツリーに追加されるポイント) は、下にシンボルが接続されていない線で表しています。
- バージョン管理ツールを使用することによって、ここで示しているのと同じような形式でSCCS デルタツリーをグラフィカルに表示できます。

開発者はそれぞれ、ブリングオーバーアクションにより統合ワークスペースから同じファイルをコピーします。このファイルは両開発者のワークスペースでは新規のものであるため、SCCS 履歴ファイルが両ワークスペースにコピーされます。統合ワークスペースは、両開発者が使用する親ワークスペースです。



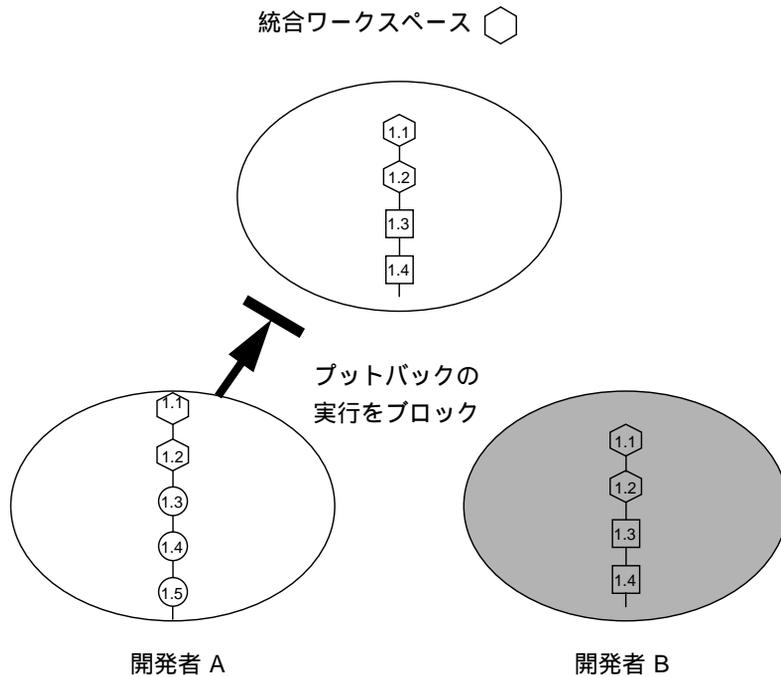
開発者 B は、このファイルに変更を加えて、2つの新しいデルタ 1.3 と 1.4 を作成した後、ファイルを統合ワークスペースに戻します (プットバックトランザクション)。このときワークスペース管理ツールは、2つの新しいデルタを親の SCCS デルタツリーに追加します。

アクセスリストなどの管理情報を保持するため、転送先ワークスペース側の SCCS 履歴ファイルが転送元ワークスペース側のファイルに置き換えられることはありません。新しいデルタが転送先の SCCS 履歴ファイルに追加されます。



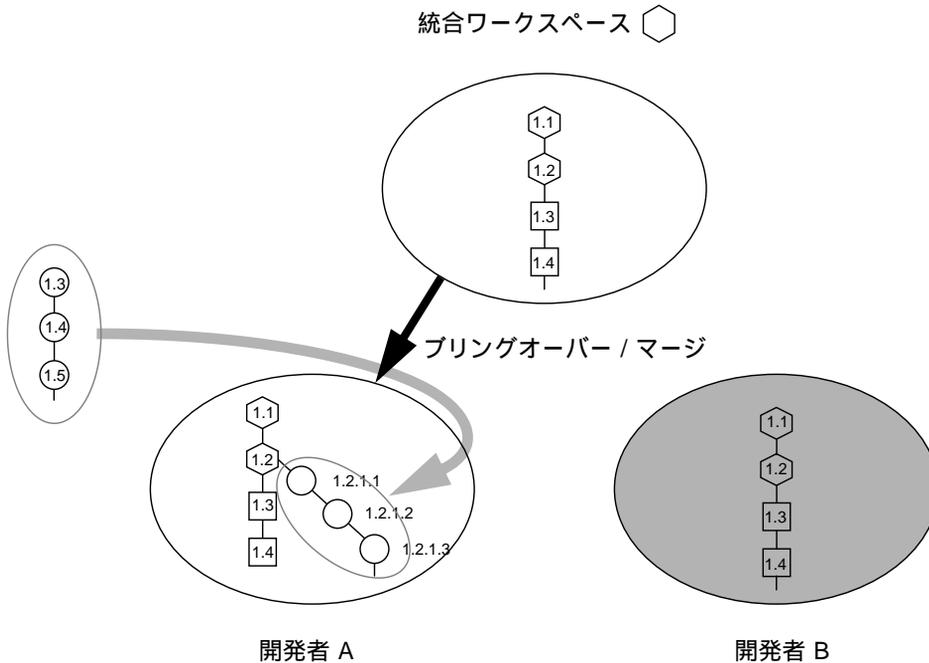
一方、開発者 A も、コピーしてきた同じファイルに独自の変更を加え (3 つの新しい
デルタ 1.3、1.4、1.5 を作成)、ファイルを統合ワークスペースに戻そうとします。

ファイルが衝突しているため、ワークスペース管理ツールは、開発者 A のプット
バックトランザクションの実行要求をブロック (拒否) します。もし、ここでプット
バックトランザクションの実行を許可すると、開発者 B がプットバックした変更が失
われてしまいます。開発者 A はプットバックの前に、開発者 B による変更内容を自分
のファイルに取り込む必要があります。



開発者 A は、この時点で開発者 B による変更を含むファイルを統合ワークスペースから自分のワークスペースにコピーします。このとき、開発者 B によって作成されたデルタが、開発者 A の SCCS 履歴ファイルに追加されます。

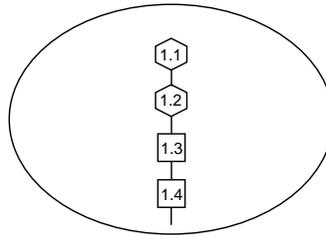
親からコピーされたデルタツリーが子の側で変更されることはありません。子で作成された新しいデルタが SCCS 分岐として、子と親に共通の最後のデルタに追加され、それぞれ新しい SID が割り当てられます。これらのデルタには、デルタの分岐点から SID を生成する SCCS 分岐番号割り当てアルゴリズムによって新しい番号が付けられます。この例では、分岐は SID 1.2 に追加され、最初のデルタの番号は 1.2.1.1 になります。また、子の側で作成された最後のデルタ (1.2.1.3、分岐になる前の 1.5) はその後もデフォルトのデルタとして扱われます。したがって、開発者 A が衝突の解決以前に子ワークスペースで作成した新しいデルタはすべて、子の分岐に追加され、親の基幹ツリーには追加されません。



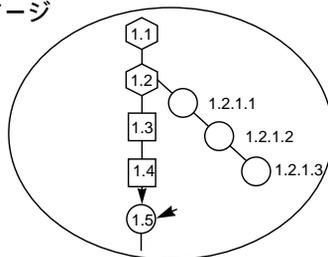
開発者 A は、自分のワークスペースで衝突解決トランザクションを使用して衝突を解決します (衝突解決についての詳細は、77 ページの「ワークスペース内での衝突の解決」を参照)。開発者 A は衝突解決トランザクションを使用して、SID 1.2.1.3 と 1.4 で表わされるそれぞれのファイルバージョンをどのようにマージするかを決定します。開発者 A が変更を適用すると、衝突解決トランザクションは新たにマージされた内容を新しいデルタ (1.5) に書き込みます。

- 新しいデルタ 1.5 をこのように丸で囲んで表現しているのは、これが開発者 A によって作成されたものであることを示すためです。
- 新たに作成されたデルタは、この時点で、開発者 A が次に行う変更内容が追加されるデフォルトの位置になります。

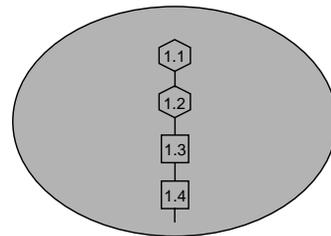
統合ワークスペース 



衝突解決/マージ

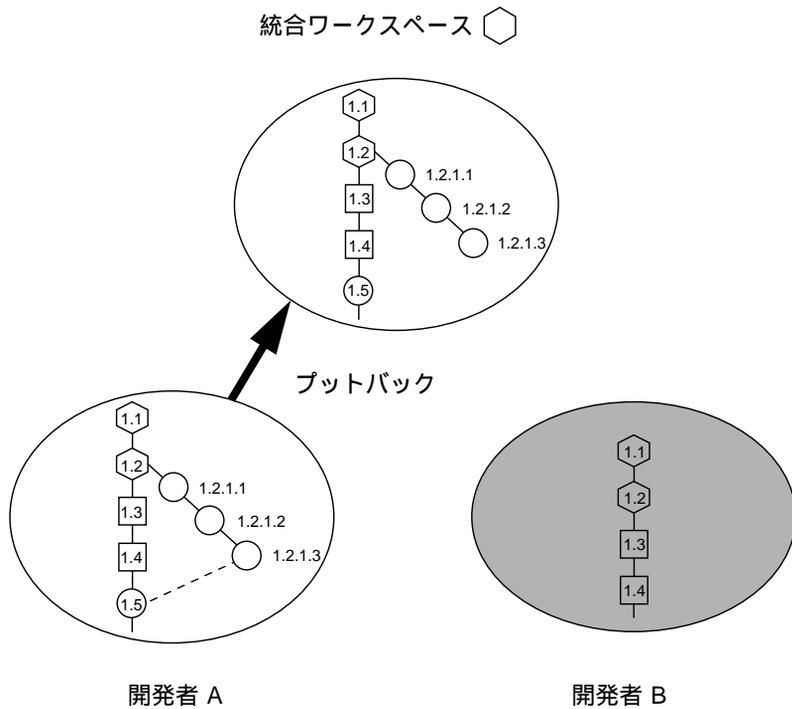


開発者 A

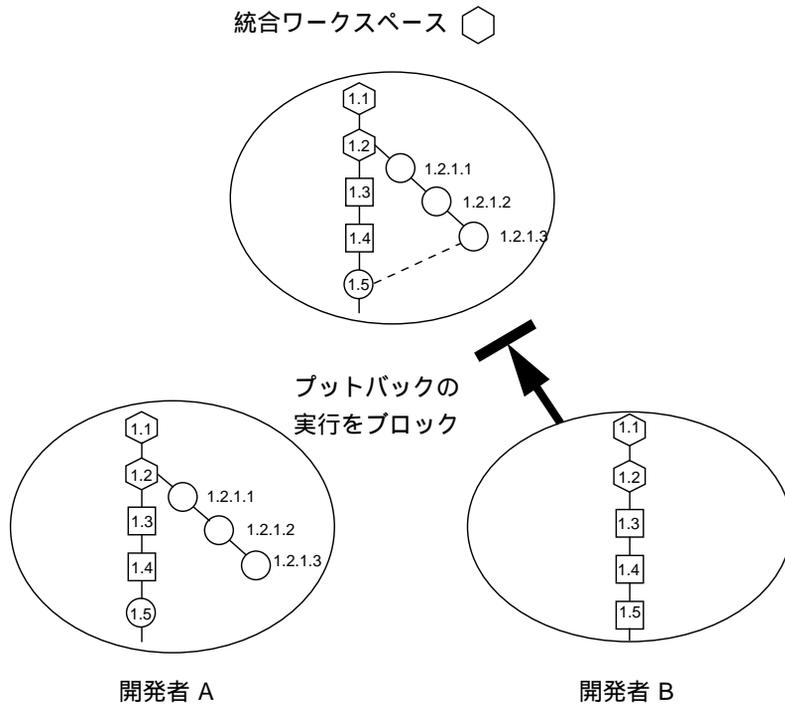


開発者 B

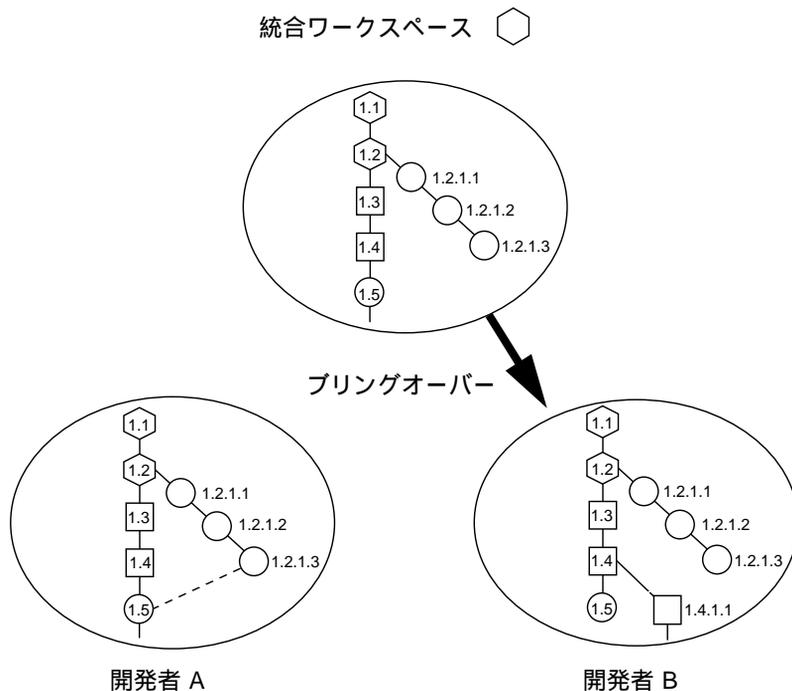
以上でファイルの衝突が解決されたので、開発者 A は自分の最新のファイルを統合ワークスペースにプットバックできます。このとき、分岐と新たに作成されたデルタは、統合ワークスペースの SCCS 履歴ファイルに追加されます。



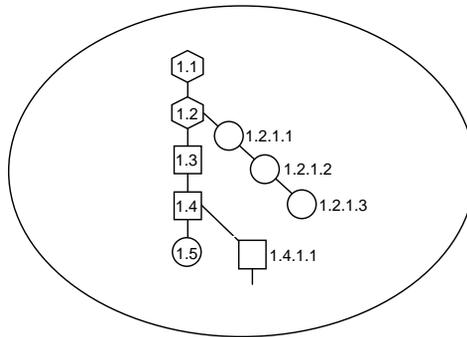
開発者 B は、自分のワークスペースでファイルに別の変更を行います。この結果、デルタ 1.5 が作成されます。続いて、この新しい変更を統合ワークスペースにプットバックしようとするのですが、開発者 A がその前に統合ワークスペースに戻した新しいデルタ 1.5 と内容が異なるため、このトランザクションの実行は許可されません。



開発者 B は、開発者 A が統合ワークスペースに戻した変更されたファイルを自分のワークスペースにブリングオーバーします。ここで、このファイルのデルタは、ワークスペース管理ツールによって開発者 B のワークスペースの SCCS 履歴ファイルに追加され、番号が付けなおされます。



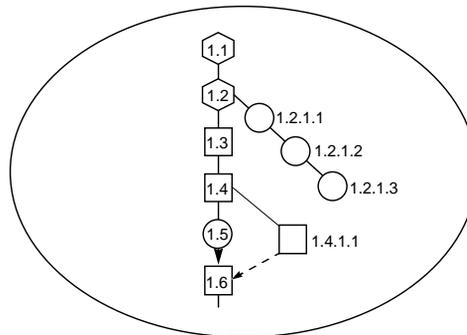
前述の例と同様に、ワークスペース管理ツールは、開発者 B が作成したデルタを、基幹デルタツリーの最新の共通デルタに分岐として追加し、それに応じて番号を付けなおします。1.5 が 1.4.1.1 になります。1.4.1.1 は、デフォルトのデルタのままです。衝突が解決される以前に開発者 B のワークスペースで新たに作成されたデルタはすべて、この分岐に追加されます。



開発者 B

開発者 B は、衝突解決トランザクションを使用して、1.5 と 1.4.1.1 との相異をマージして衝突を解決し、新しいデルタ 1.6 を作成します。

- マージされた新しい変更内容は、新しいデルタとして親デルタ 1.6 に追加されます。
- 新しいデルタは、そのワークスペースのオーナーである開発者 B によって所有されます。
- 新しいデルタはデフォルトのデルタになるため、子ワークスペースで以後行われる変更の結果はすべて、このデルタの下に追加されることになります。



開発者 B

上記のマージ例は、統合ワークスペースと 2 つの子ワークスペースがあって、子ワークスペースをそれぞれ異なる開発者が所有しているときに行われる処理を示しています。2 人の開発者は統合ワークスペースから同じファイルのコピーをプリングオーバーして、それぞれにファイルを変更しました。変更をブットバックしようとする、衝突が発生します。ワークスペース管理ツールは変更が上書きされないようにデルタを管理し、SCCS 履歴ファイルを操作して、ファイルの履歴が正確で完全であるようにします。

SCCS マージ可能 ID について

この節では、SCCS マージ可能 ID (SMID) が必要な理由、SCCS デルタ ID (SID) から SMID への変換方法、および SMID から SID への変換方法について説明します。

SMID を使用すると、SID の変更に関係なく、すべてのデルタを一意に識別できます。SMID は、Xerox Secure Hash Function を使用して生成される番号です。フリーズポイントツールを使用してフリーズポイントファイルを作成すると、SCCS 履歴ファイル内の現在のデルタとルートデルタの両方について SMID が計算されます。この 2 つの値を使用することにより、フリーズポイントツールでは、SID が変更された場合でもファイル内のデルタを識別できます。

SMID が必要な理由

ワークスペース管理ツールは、更新プリングオーバーランザクションの実行中にファイルの衝突 (ファイルが親ワークスペースと子ワークスペースの両方で変更された状態) を検出すると、親ワークスペースの新しいデルタを、子ワークスペースの SCCS 履歴ファイル内にマージします。このマージ処理では、子ワークスペースで作成されたデルタは、親子両方のデルタに共通するデルタ (共通の祖先) からの SCCS 分岐に移動します。

ワークスペース管理ツールは、子デルタを分岐に移動する際にその SID を変更します。フリーズポイントファイルでのデルタの識別に SID を使用している場合は、この移動により、フリーズポイントファイル内の情報が無効になります。このため、衝突する SCCS 履歴ファイルをマージした後は、デルタの識別に SID を使用できません。

SMID と SID の変換

SID から SMID への変換および SMID から SID への変換には、`freezept` コマンドの `sid` サブコマンドおよび `smid` サブコマンドを使用します。これらの ID の変換は、デルタを追跡するための独自のスクリプトやプログラムを作成する場合に使用すると便利です。

SID から SMID への変換

SID から SMID への変換には、`freezept smid` コマンドを使用します。構文は次のとおりです。

```
freezept smid [-w workspace] [-r SID] [-a] filename
```

- `-w` オプションを使用して、ワークスペースを指定できます。
- `-r` オプションには、(`filename` ファイル内の) SID を指定します。この SID から SMID が求められます。
- `-a` オプションは、`filename` ファイル内のすべての SID に対する SMID を求める場合に使用します。

例

```
example% freezept smid -r 1.38 module.c  
SID 1.38 = SMID "f5b67794 705f0768 a89b1f4 588de104"
```

```
example% freezept smid -a bringover.1  
SID 1.1 = SMID "b05b0a2f 1db5246e 1a466014 707e38f5"  
SID 1.2 = SMID "d6s5c61f 5634f0ef 9847a080 d0d7b212"  
SID 1.2 = SMID "e31acdd5 6c1232e2 9e81c287 1edb2f41"  
SID 1.3 = SMID "c34c91b4 a818622a 2457356a 489b2728"  
SID 1.4 = SMID "98c0fd8d 889563fb cf722c2b 6afc9636"  
SID 1.5 = SMID "ble24be3 752fec3e df2d2717 a9b3f1fa"  
SID 1.6 = SMID "2b93d39 1ea2f6ba 9814320c bc609acb"  
SID 1.7 = SMID "1db7d640 42b0f009 35c60d7b b230bd85"  
SID 1.8 = SMID "906dfe9a ca7e2d6c a64da5be 4baef254"
```

SMID から SID への変換

SMID から SID への変換には、`freezept sid` コマンドを使用します。構文は次のとおりです。

```
freezept sid [-w workspace] [-m "SMID"] [-a] filename
```

- `-w` オプションを使用して、ワークスペースを指定できます。

- **-m** オプションには、(*filename* ファイル内の) SMID を指定します。この SMID から、SID が求められます。
- **-a** オプションは、*filename* ファイル内のすべてのデルタの SID を求める場合に使用します。

注 – SMID にはスペース文字が含まれているため、SMID 全体を引用符で囲む必要があります。

例

```
example% freezept smid -m "64fdd0df de9d7dd de75812 23da96aa"  
module.c  
SMID "f5b67794 705f0768 a89b1f4 588de104" = SID 1.36
```

```
example% freezept smid -a bringover.1  
SMID "b05b0a2f 1db5246e 1a466014 707e38f5" = SID 1.1  
SMID "d6s5c61f 5634f0ef 9847a080 d0d7b212" = SID 1.2  
SMID "e31acdd5 6c1232e2 9e81c287 1edb2f41" = SID 1.2  
SMID "c34c91b4 a818622a 2457356a 489b2728" = SID 1.3  
SMID "98c0fd8d 889563fb cf722c2b 6afc9636" = SID 1.4  
SMID "ble24be3 752fec3e df2d2717 a9b3f1fa" = SID 1.5  
SMID "2b93d39 1ea2f6ba 9814320c bc609acb" = SID 1.6  
SMID "1db7d640 42b0f009 35c60d7b b230bd85" = SID 1.7  
SMID "906dfe9a ca7e2d6c a64da5be 4baef254" = SID 1.8  
SMID "77481e8a 61542339 cc28f532 e5fc6389" = SID 1.9  
SMID "cb97c9a6 d0342cf6 19b7b743 2436ca1c" = SID 1.10  
SMID "46de4131 b96b9973 93958a07 b960074c2" = SID 1.11
```


付録 A

エラーメッセージと警告メッセージ

この付録では、Sun WorkShop TeamWare で表示されるエラーメッセージおよび警告メッセージについて説明します。

この付録は、次の節で構成されています。

- エラーメッセージ
- 警告メッセージ

メッセージにはすべて番号が付けられており、この付録ではメッセージを番号順に列挙します。また、メッセージごとにその意味を示し、エラーに関してはその対処法について説明します。

エラーメッセージ

表 A-1 に、エラーメッセージと、その意味および対処方法を示します。

表 A-1 エラーメッセージ

1000 - 1999 システムエラー

1000 から 1999 までの番号のエラーメッセージは、ワークスペース管理ツールのコマンドが出したオペレーティングシステムコールから返されるエラーを表わします。これらのエラーメッセージは、簡単なワークスペース管理ツールのメッセージと、これに付加されるシステムエラーメッセージとその番号で構成されています。システムエラーに関しては、オペレーティングシステムのマニュアルを参照してください。

表 A-1 エラーメッセージ (続き)

2000	<p>ファイル <ファイル名> に長すぎるか行があるか、期待していないファイルの終わりがあります</p> <p>意味: <ファイル名> の読み取り中に、ワークスペース管理ツールのコマンドがバッファに入れるには多すぎる文字数の行を検出しました。各行の最大長は 1024 文字です。</p> <p>対処法: 長い行を短くしてから、コマンドを再実行してください。</p>
2001	<p>-w オプション、CODEMGR_WS 環境変数、または現在の作業ディレクトリでワークスペースを指定してください</p> <p>意味: ワークスペース管理ツールのコマンドは、処理対象のワークスペースを特定できませんでした。ワークスペース管理ツールのコマンドは次の順序でワークスペースのパス名を入手しようとしています。</p> <ul style="list-style-type: none">• コマンドの -w オプションで指定されたパス名• 環境変数 CODEMGR_WS に設定されたパス名• 現在のディレクトリ (そのディレクトリがワークスペース内の階層に存在する場合) <p>対処法: 上記いずれかの方法でワークスペースのパス名を指定してください。</p>
2002	<p>NSE 環境下のワークスペースの親変更には -p オプションは使用できません</p> <p>意味: 親が NSE 環境であるワークスペースについては、CLI (コマンド行インタフェース) の bringover および putback コマンドで -p オプションを使用して親を変更することはできません。</p> <p>対処法: 親が NSE 環境であるワークスペースの親をワークスペース管理ツールのワークスペースに変更するには、workspace reparent コマンドを使用してください。このようなワークスペースの親を別の NSE 環境に変更することはできません。</p>
2003	<p><ディレクトリ名> はワークスペースではありません</p> <p>意味: コマンドで指定された <ディレクトリ名> は、ワークスペース管理ツールのワークスペースではありません。ワークスペース管理ツールのワークスペースであるかどうかは、最上位ディレクトリに Codemgr_wsdata ディレクトリが存在するかどうかで判別されます。</p> <p>対処法: 別のワークスペース名を指定してください。または、workspace create コマンドを使用するか、GUI の「ワークスペース管理」ウィンドウの「ファイル」メニューから「ワークスペース作成」を選択して、ディレクトリをワークスペースに変換してください。</p>
2004	<p>ワークスペース <ワークスペース名> には親ワークスペースはありません</p> <p>意味: ワークスペース管理ツールのコマンド (ブリングオーバーまたはプットバック) は、<ワークスペース名> の親を検出できなかったため、処理を完了することができませんでした。</p>

表 A-1 エラーメッセージ (続き)

	<p>対処法: <code>workspace parent</code> コマンドを使用するか、GUI の「ワークスペース管理」ウィンドウの「編集」メニューから「親の変更」を選択して、親のいないワークスペースに親を指定してください。</p>
2005	<p>親ワークスペース <ワークスペース名> は <マシン名> にマウントされていないため見えません</p> <p>意味: 親ワークスペースが含まれているファイルシステムは現在、<マシン名> にマウントされていません。</p> <p>対処法: 親ワークスペースを含むファイルシステムをマウントしてから、コマンドを再実行してください。</p>
2006	<p>ファイル名 <ファイル名> 中にパス構成要素 <code>..</code> が多過ぎます</p> <p>意味: ワークスペース管理ツールで指定された相対ファイル名は、ワークスペースのルートディレクトリに対する相対パスとして解釈されます。ファイル名中に複数の <code>..</code> を使用している場合は、その中の 1 つが、ワークスペースのルートよりも上の階層のディレクトリを指している可能性があります。</p> <p>対処法: パス名に含まれる <code>..</code> の数を減らしてください。</p>
2007	<p>UID <ユーザー ID> のユーザー名が取得できません</p> <p>意味: NIS マップでも <code>/etc/passwd</code> でも UID が見つかりません。</p> <p>対処法: NIS サーバーおよび NIS マップを調べてください。</p>
2008	<p>ファイル <ファイル名> にバージョン番号がありません</p> <p>意味: ワークスペース管理ツールのコマンドは、メタデータファイル (<code>Codemgr_wsdata</code> ディレクトリにあるファイル) にアクセスする場合に、そのファイルの作成時にファイル内に書き込まれたバージョン番号 (たとえば、VERSION 1) を確認します。メタデータファイル <ファイル名> にこのバージョン番号の文字列が含まれていません。</p> <p>対処法: <ファイル名> が壊れていないかどうかを調べてください。ファイルを編集した際にバージョン番号が削除された可能性があります。バージョン番号がないだけで、ファイルが壊れていない場合は、<code>workspace create</code> コマンドを使用するか、GUI の「ワークスペース管理」ウィンドウの「ファイル」メニューから「ワークスペース作成」を選択して、新しいワークスペースを作成してください。新しいワークスペース内の同じファイルでバージョン番号を調べ、<ファイル名> を編集してその値を追加してください。</p>
2009	<p><コマンド名> コマンドが失敗しました。/bin/sh はシグナル <シグナル> により終了しました</p> <p>意味: ワークスペース管理ツールのコマンドが <コマンド名> の実行を試みたところ、シェルが <シグナル> によって終了したために実行に失敗しました。</p> <p>対処法: ワークスペース管理ツールのコマンドを再実行してください。</p>

表 A-1 エラーメッセージ (続き)

2010	<p><コマンド名> コマンドが失敗しました。シェル (/bin/sh) を実行できませんでした</p> <p>意味: ワークスペース管理ツールのコマンドがシェルを起動できませんでした。これは、スワップ空間またはメモリなどのシステム資源が不足していることを意味します。</p> <p>対処法: システム資源を調べてください。</p>
2011	<p><コマンド名> コマンドはシグナル <シグナル> により終了しました</p> <p>意味: ワークスペース管理ツールが起動したコマンドが <シグナル> を受信しました。</p> <p>対処法: コマンドを再実行してください。エラーが再発する場合は、シグナルについて、Solaris のマニュアルを参照してください。</p>
2012	<p><コマンド名> コマンドは、ステータス <ステータス> で終了しました</p> <p>意味: ワークスペース管理ツールのツールで実行したコマンドは、正常に終了すると、ステータス 0 を返します。実行したコマンドが 0 以外のステータスを返した場合はエラーと見なされます。</p> <p>対処法: <コマンド名> のステータスの意味は、該当するマニュアルを参照してください。</p>
2013	<p>FLP <FLP名> は親ワークスペースにも子ワークスペースにも存在しません</p> <p>意味: プリンクオーバーまたはブットバックトランザクションに指定された FLP (ファイルリストプログラム) <FLP 名> が、親ワークスペースにも子ワークスペースにも存在しませんでした。</p> <p>対処法: 目的とする FLP のパス名を確認してから、トランザクションを再度実行してください。</p>
2014	<p><プログラム名> を実行できませんでした</p> <p>意味: ワークスペース管理ツールのコマンドが別のプログラムを実行できませんでした。</p> <p>対処法: インストールが正しく行われたかどうか確認してください。また、プログラムが検索パス上に存在し、かつそのパーミッションが正しく設定されていることを確認してください。</p>
2015	<p>ワークスペース <ワークスペース名> はすでに存在します</p> <p>意味: すでに存在するワークスペースを作成しようとしてしました。</p> <p>対処法: 別のワークスペース名を使用してコマンドを再実行してください。</p>
2016	<p>ワークスペース <ワークスペース名> は存在しません</p> <p>意味: ワークスペース管理ツールのコマンドの実行時に引数として指定した <ワークスペース名> が見つかりませんでした。</p> <p>対処法: 正しいパス名を指定しているかどうか確認ください。</p>
2017	<p><ファイル名> を開けられないためチェックインのコメントを得られません</p>

表 A-1 エラーメッセージ (続き)

	<p>意味: ワークスペース管理ツールはチェックインコメントを一時ファイルに格納しましたが、そのファイルを開いてコメントを読み取ることができませんでした。</p> <p>対処法: ファイルのパーミッション、およびファイルを開くことができない原因となっているその他のファイルシステム上の問題を調べてください。</p>
2018	<p>同じワークスペースに対しての親の変更はできません</p> <p>意味: (トランザクションの一部として、または明示的な親の変更コマンドによって、) あるワークスペースをそのワークスペースの新しい親にしようとした。</p> <p>対処法: 別の親を指定してコマンドを再実行してください。</p>
2019	<p>内部エラー: unknown locktype lock</p> <p>意味: ワークスペースロックファイル (<code>Codemgr_wsdata/locks</code>) が壊れています。未知のロック値が検出されました。</p> <p>対処法: ロックファイルを編集して、修復してください。詳しくは、<code>locks(4)</code> のマニュアルページまたは 65 ページの「ワークスペースのロックの解除」を参照してください。</p>
2020	<p>ワークスペース名を指定してください</p> <p>意味: ワークスペース管理ツールのコマンドは、処理対象であるワークスペースを特定できませんでした。ワークスペース管理ツールの各コマンドは、次の順序でワークスペースのパス名を入手しようとします。</p> <ul style="list-style-type: none">• コマンドの <code>-w</code> オプションで指定されたパス名• 環境変数 <code>CODEMGR_WS</code> に設定されたパス名• 現在のディレクトリ (このディレクトリがワークスペース内に存在する場合) <p>対処法: 上記いずれかの方法を使用してワークスペースのパス名を指定してください。</p>
2021	<p><ロックの種類> ロックを取得できません。ワークスペース <ワークスペース名> に以下のロックが存在しています: ユーザー <ユーザー名> は <コマンド名> コマンドを実行中 (pid <プロセス番号>) マシン <マシン名> です。</p>

表 A-1 エラーメッセージ (続き)

意味: 整合性を保持するために、ワークスペース管理ツールのワークスペースコマンドは、ワークスペース内のデータを読み書きする間そのワークスペースをロックします。実行しようとしたコマンドは、対象となるワークスペースがすでにロックされていたため、実行できませんでした。更新プリングオーバーラップでワークスペース管理ツールが親ワークスペース内のファイルを読み取ったり検査したりしている間、そのワークスペースには読み取りロックがかかります。また、子ワークスペース内のファイルを操作している間、子ワークスペースには書き込みロックがかかります。ワークスペース内で複数のワークスペース管理ツールのコマンドが同時に別々のファイルを操作する場合には、複数の読み取りロックが発生します。読み取りロックが存在する間、そのワークスペースに別のコマンドで書き込みを行うことはできません。一方、書き込みロックは一度に1つしか有効になりません。書き込みロックがかかっている場合、ワークスペース管理ツールのコマンドでそのワークスペースに書き込みを行うことはできません。ロックの状態は、各ワークスペースに存在する `Codemgr_wsdata/locks` ファイルで制御されます。

対処法: システムが正常に動作している場合、該当ワークスペースをロックしているコマンドがロックを解除するまで待ちます。ワークスペースがロック状態に固定されてしまった場合には (たとえば、あるコマンドでロック状態になっているときにシステムがクラッシュした場合など)、GUI の「ワークスペース管理」ウィンドウの「オプション」メニューから「ワークスペース」を選択し「ワークスペース属性」ダイアログの「カテゴリ」リストボックスから「ロック」を選択するか、`workspace locks` コマンドを使用して、ロックを解除してください。

2022 無効なサブコマンド - <コマンド名>

意味: `resolve`、`workspace`、または `codemgr` コマンドのサブコマンドに関するヘルプ情報を表示しようとしたますが、存在しないサブコマンドが指定されました。

対処法: コマンドを入力した後、`help` サブコマンドを指定すると、各コマンドの有効なサブコマンドの一覧を表示できます。

2023 未使用

2024 ファイル <ファイル名> にデルタはありません

意味: SCCS 履歴ファイル <ファイル名> にデルタがありません。このため、そのファイルを処理できません。

対処法: おそらく、この履歴ファイルは誤って上書きされています。

2025 <コマンド名> コマンドが見つかりません。実行可能形式の <コマンド名> はありません。また、`PATH` <PATH の内容> にもコマンドが見つかりません

意味: ワークスペース管理ツールのコマンドが別のプログラムを実行しようとしたますが、そのプログラムが見つかりませんでした。

表 A-1 エラーメッセージ (続き)

	<p>対処法: プログラムが正しくインストールされていることを確認し、そのプログラムがあるディレクトリを <code>PATH</code> 環境変数に加えてください。</p>
2026	<p>不正な SCCS 制御文字 (<文字>) がファイル <ファイル名> の <行番号> 行目にあります</p> <p>意味: ワークスペース管理ツールのコマンドは <ファイル名> を SCCS 履歴ファイルであると想定しますが、検出した文字からこのファイルが履歴ファイルでないか、壊れていることが判明しました。</p> <p>対処法: SCCS 履歴ファイルのフォーマットについて、Solaris の SCCS 関連マニュアルを参照してください。</p>
2027	<p>ファイル <ファイル名> が壊れています。 <行番号> 行目</p> <p>意味: ワークスペース管理ツールのコマンドは、ワークスペースメタデータファイル (<code>Codemgr_wsddata</code> ディレクトリ内のファイル) を読み取ることができませんでした。 <行番号> に無効な文字があります。</p> <p>対処法: ファイルを確認して必要な修正を行なってください。ワークスペース管理ツールのメタデータファイルはすべて、ASCII 形式のテキストファイルであり、編集が可能です。メタデータファイルのフォーマットについての詳細は、<ファイル名> のマニュアルページ、または第 6 章「ワークスペースの操作と設定」を参照してください。</p>
2028	<p><コマンド名> コマンドは PATH <パス名> では見つかりません</p> <p>意味: ワークスペース管理ツールのコマンドが別のプログラムを実行しようとしたのですが、そのプログラムが見つかりませんでした。</p> <p>対処法: プログラムが正しくインストールされていることを確認し、そのプログラムがあるディレクトリを加えてください。</p>
2029	<p>ファイルに解決されていない衝突があります。 'edit m' を実行し "^<<<<<<<" を検索してください</p> <p>意味: このエラーは <code>resolve</code> コマンドによって出されたものです。ファイルにまだ解決されていない衝突が残っているのに、そのファイルを保存しようとした。</p> <p>対処法: <code>edit m</code> サブコマンド (マージ結果の編集) を実行して衝突を解決してから、ファイルを保存してください。衝突には、文字 "<code>^<<<<<<<</code>" のマークが付けられています。</p>
2030	<p>番号 <ファイル番号> のファイルはありません</p> <p>意味: <code>resolve</code> コマンドは、衝突のあるファイルについて番号を付けてリストを作成します。このメッセージは、選択された <ファイル番号> がリストに存在しないことを表わしています。</p> <p>対処法: <code>list</code> サブコマンドを使用してファイルのリストを作成し、指定するファイルの正しい番号を確認してください。</p>

表 A-1 エラーメッセージ (続き)

2031	<p>ホームディレクトリが見つからないので、ファイル <ファイル名> に書き込みができません</p> <p>意味: ワークスペース管理ツールのコマンドはユーザーのホームディレクトリを見つけることができなかつたため、その中の <ファイル名> も見つけられませんでした。通常、これは NIS マップの問題であることを示しています。</p> <p>対処法: NIS サーバーおよび該当する NIS マップを調べてください。</p>
2032	<p>ファイル <ファイル名> の中の行を構文解析できません: <行></p> <p>意味: <code>resolve</code> コマンドは、起動時に <code>~/.codemgr_resrc</code> ファイルを読み取ってユーザー定義の属性を入手します。このとき、<行> をプログラムが正しく解釈できませんでした。</p> <p>対処法: 有効なエントリだけで構成されるように <code>~/.codemgr_resrc</code> ファイルを修正してください。これらのエントリについての詳細は、<code>resolve(1)</code> のマニュアルページを参照してください。</p>
2033	<p>ディレクトリリストを引数または環境変数 <code>CODEMGR_WSPATH</code> で指定してください</p> <p>意味: このメッセージは、<code>workspace list</code> コマンドの実行時に出されるもので、ディレクトリ (またはディレクトリリスト) が正しく指定されていないことを示します。ディレクトリは、このコマンドの標準引数として指定するか、そのパス名を <code>CODEMGR_WSPATH</code> 環境変数にあらかじめ設定しておくことができます。</p> <p>対処法: ディレクトリの指定を含めてコマンドを再実行するか、<code>CODEMGR_WSPATH</code> 環境変数にディレクトリパスを設定してからコマンドを再実行してください。</p>
2034	<p>内部エラー: <code>access control operation operation_name does not have a built-in default</code></p> <p>意味: ワークスペース管理ツールのコマンドがワークスペース操作 (ブリングオーバー、プットバック、親の変更) に関するアクセス権を検証しようとした。内部的な整合性検査に失敗しました。</p> <p>対処法: ご購入先にお問い合わせください。</p>
2035	<p>アクセス制御ファイルが存在しません</p> <p>意味: ワークスペース管理ツールのコマンドがワークスペース操作 (ブリングオーバー、プットバック、親の変更) に関するアクセス権を検証しようとした。影響を受けるワークスペースにアクセス制御ファイル (<code>Codemgr_wsdata/access_control</code>) が存在しませんでした。</p>

表 A-1 エラーメッセージ (続き)

	<p>対処法: アクセス制御ファイルがワークスペースから削除されていた場合、新しいファイルを別のワークスペースからコピーし、そのファイルの内容を編集してアクセス権を適切に変更します。他のワークスペースがない場合は、CLI の <code>workspace create</code> コマンドを使用して、または GUI の「ワークスペース管理」ウィンドウの「ファイル」メニューから「ワークスペース作成」を選択して新しいワークスペースを作成し、そのワークスペースからファイルをコピーします。詳細は、第 4 章または <code>access_control(4)</code> マニュアルページを参照してください。</p>
<p>2036</p>	<p>共通祖先ファイルを指定できません。共通祖先デルタはありません</p> <p>意味: <code>resolve</code> のサブコマンド (<code>diff</code>、<code>edit</code>、<code>more</code>) の引数として、祖先 (a) が指定されました。2 つの衝突解決の対象ファイルには、共通の祖先が存在しません。これは、同じ名前のファイルが親と子で同時に作成された場合に最もよく起こる現象です。これらのファイルは同じ名前を持っていますが、共通の祖先から派生したものではありません。祖先についておよび衝突解決における祖先の役割については、第 6 章を参照してください。</p> <p>対処法: <code>diff</code>、<code>edit</code>、<code>more</code> サブコマンドの引数に祖先 (a) を指定せずに衝突解決の処理を実行してください。</p>
<p>2037</p>	<p>無効な引数 - <文字列></p> <p>意味: <code>resolve</code> のサブコマンドの 1 つに無効な引数が指定されました。このコマンドでは有効な引数として a (祖先)、c (子)、p (親)、および m (マージ結果) のいずれかを受け付けます。</p> <p>対処法: 有効な引数 (a、c、p、または m) のいずれか 1 つを指定してください。詳細は、<code>resolve(1)</code> のマニュアルページを参照してください。</p>
<p>2038</p>	<p>親ワークスペースは NSE 環境です。'nseputback' コマンドを使用してください</p> <p>意味: プットバックトランザクションで指定された親が NSE 環境であり、ワークスペース管理ツールのワークスペースではありませんでした。</p> <p>対処法: <code>nseputback</code> コマンドを使用して、変更内容を該当ワークスペースからその環境へプットバックしてください。</p>
<p>2039</p>	<p>ファイル <ファイル名> はおそらく s ファイルではありません。<行番号> 行に ^A が必要ですが <文字> (16 進数) が見つかりました。</p> <p>意味: ワークスペース管理ツールのコマンドは <ファイル名> が SCCS 履歴ファイルであることを想定しています。ファイル形式を見ると、履歴ファイルではないか、壊れてます。</p> <p>対処法: SCCS 履歴ファイルのフォーマットについては、Solaris の SCCS 関連マニュアルを参照してください。</p>
<p>2040</p>	<p>ファイル <ファイル名> はマージされませんでした。merge または filemerge サブコマンドを使用してください。</p>

表 A-1 エラーメッセージ (続き)

	<p>意味: まだマージされていないファイルを保存しようとしてしました。</p> <p>対処法: ファイルマージツールまたは <code>twmerge</code> コマンドを使用して、ファイルをマージしてください。 <code>twmerge</code> コマンドについての詳細は、 <code>twmerge(1)</code> のマニュアルページを参照してください。</p>
2041	<p><パス名> はワークスペースでもディレクトリでもありません</p> <p>意味: 作成プリングオーバーランザクションで指定された <パス名> は、ワークスペース管理ツールのワークスペースでもディレクトリでもありません。</p> <p>対処法: 別のワークスペースまたはディレクトリ名を指定してください。</p>
2042	<p>ToolTalk メッセージを作成できません。エラー <ToolTalk エラーコード></p> <p>意味: <code>resolve</code> コマンドは ToolTalk サービスを介してファイルマージプログラムとやりとりします。 ToolTalk は、ウィンドウシステムとともに提供されるアプリケーション間通信サービスです。このエラーの場合は、 <code>resolve</code> コマンドは ToolTalk ルーチン呼び出しして、ファイルマージツールに対する ToolTalk メッセージを作成しようとしてしました。しかし、 ToolTalk ルーチンはこのメッセージを作成できず、 <ToolTalk エラーコード> を返してきました。</p> <p>対処法: このエラーについては、 ToolTalk のマニュアルを参照してください。</p>
2043	<p>sccs ファイル <ファイル名> は壊れています</p> <p>意味: このメッセージは SCCS の <code>admin -h</code> コマンドから返されるメッセージで、新たに算出されたチェックサムが該当ファイルの先頭行に記述されているチェックサムと一致しなかったことを示すものです。</p> <p>対処法: 詳細は、 Solaris の SCCS 関連マニュアルを参照してください。</p>
2044	<p>テンプレート <テンプレートファイル名> から一時ファイル名を作成することができません</p> <p>意味: ワークスペース管理ツールのコマンドが一時ファイルを作成できませんでした。これは、ワークスペース管理ツールの内部エラーです。</p> <p>対処法: このファイルに書き込みを実行できなかったシステムレベルでの原因の有無を調べてください (たとえば、ファイルパーミッション、コマンドのオーナーが不適切である、など)。</p>
2045	<p><ファイル名> の <code>fprintf</code> に失敗しました</p> <p>意味: コマンドが <ファイル名> に書き込みを行うことができませんでした。</p> <p>対処法: ファイルのパーミッション、およびファイルシステムへの書き込みを禁止するその他のファイルシステム上の問題を調べてください。</p>

表 A-1 エラーメッセージ (続き)

2046	<p>このバイナリとファイル <ファイル名> のバージョンが一致しません。このバイナリはバージョン <期待する番号> と互換性がありますが、ファイルはバージョン <現状の番号> です。リリース <リリース番号> より新しいリリースにアップグレードしてください。</p> <p>意味: ワークスペース管理ツールの各メタデータファイル (<code>Codemgr_wsdata/*</code>) には、バージョン番号を示す文字列が含まれています (リリース 1.0 の場合、バージョン番号は VERSION 1)。ワークスペース管理ツールのコマンドはこれらのファイルの読み書きを行う際に、整合性を検査するために、このファイルに期待するバージョン番号が含まれているかどうかを調べます。コマンドは <期待する番号> を想定していましたが、実際には <現状の番号> でした。このメッセージは、古いバイナリが新しいメタデータファイルと共に使用されていて、それがファイルを破壊する原因になる可能性があることを示しています。</p> <p>対処法: 最新バージョンのワークスペース管理ツールのバイナリを使用してください。</p>
2047	<p>ファイル <ファイル名> のバージョン <検出したバージョン番号> から <現在のバージョン番号> への変換方法が不明です</p> <p>意味: バージョン番号が 1 より古いメタデータファイルが検出されました。ワークスペース管理ツールの各メタデータファイル (<code>Codemgr_wsdata/*</code>) には、バージョン番号を示す文字列が含まれています (リリース 1.0 の場合、バージョン番号は VERSION 1)。ワークスペース管理ツールのコマンドはこれらのファイルの読み書きを行う際に、整合性検査として、このファイルに期待するバージョン番号が含まれているかどうかを調べます。ワークスペース管理ツールのバイナリとメタデータファイルの新しいバージョンがリリースされると、これらのファイルのいくつかはその形式が変更されている可能性があります。ワークスペース管理ツールのコマンドには、この形式の変換を行うコードが含まれています。</p> <p>対処法: メタデータファイルのバージョン文字列が編集集中に誤って変更されたと考えられます。ファイルを調べて、その先頭行を正しいバージョン番号に変更してください。</p>
2048	<p>子ワークスペースを作成プリングオーバーするには、ファイルがディレクトリ、または -f 引数のいずれかを指定しなければなりません</p> <p>意味: プリングオーバートランザクションのコマンド行が正しく構成されていませんでした。少なくとも 1 つのファイル、ディレクトリ、または FLP を引数として指定しなければなりません。この引数が省略された場合、ワークスペース管理ツールはワークスペースの <code>Codemgr_wsdata/args</code> ファイルから引数を取り込みます。</p> <p>対処法: コマンドを再入力し、正しい数の引数を指定してください。</p>
2049	<p><ファイル名> がどこからマウントされているかわかりません</p> <p>意味: ワークスペース管理ツールのコマンドは NFS マウントされたディレクトリのパス名を <マシン名 : パス名> 形式に変換してその処理の多くを実行します。このメッセージは、<code>/etc/mtab</code> (Solaris 1.x の場合) または <code>/etc/mnttab</code> (Solaris 2.x の場合) に <ファイル名> を含むマウントエントリが存在しないことを示しています。</p>

表 A-1 エラーメッセージ (続き)

	対処法: <ファイル名> が含まれているファイルシステムをマウントしなおしてください。
2050	<p>絶対パス <ファイル名> を決定できません</p> <p>意味: ワークスペース管理ツールのコマンドがディレクトリを読み取ることができませんでした。このメッセージは、ファイルシステムに問題があることを示しています (たとえば、ディレクトリのパーミッションが不適切である)。</p> <p>対処法: ファイルシステム、<ファイル名> のパス中のディレクトリやファイルのパーミッションを確認してください。</p>
2051	<p>名前を <ファイル名> に変更できません。同じ名前があります</p> <p>意味: ブリリングオーバー、プットバック、または取り消しトランザクションにおいて、ソースワークスペース内のファイルが、宛先ワークスペース内ですでに使用されている名前と同じ名前に変更されたものがありました。</p> <p>対処法: 一方のディレクトリで名前を変更してください。</p>
2052	<p>ファイル <ファイル名> が壊れています。<行番号> 行目の "BEGIN" の後にテキストがあります。通知は送られませんでした</p> <p>意味: ワークスペース管理ツールのコマンドによるワークスペース通知ファイル <code>Codemgr_wsdata/notification</code> の読み取り中にエラーが発生しました。通知対象とするファイル/ディレクトリのリストを区切る BEGIN 文の行中に、BEGIN 文以外のテキストが検出されました。ワークスペース管理ツールのコマンドは通知の設定を正しく解析することができないので、通知ファイルに通知要求を設定している場合でも、その通知は送信されません。</p> <p>対処法: この通知ファイルに BEGIN 文を正しく入力してください。通知ファイルの形式については、<code>notification(4)</code> のマニュアルページまたは 第 4 章を参照してください。</p>

表 A-1 エラーメッセージ (続き)

2053	ファイル <ファイル名> が壊れています。 <行番号> 行目の "END" の後にテキストがあります。通知は送られませんでした
	<p>意味: ワークスペース管理ツールのコマンドによるワークスペース通知ファイル <code>Codemgr_wsdata/notification</code> の読み取り中にエラーが発生しました。通知対象とするファイル/ディレクトリのリストを区切る END 文の行中に、END 文以外のテキストが検出されました。ワークスペース管理ツールのコマンドは通知の設定を正しく解析することができないので、通知ファイルに通知要求を設定している場合でも、その通知は送信されません。</p> <p>対処法: この通知ファイルに END 文を正しく入力してください。通知ファイルの形式についての詳細は、<code>notification</code>(4) のマニュアルページまたは 40 ページの「他のユーザーへのトランザクションの実行通知」を参照してください。</p>
2054	ファイル <ファイル名> が壊れています。 <行番号> 行目に "END" がありません。通知は送られませんでした
	<p>意味: ワークスペース管理ツールのコマンドによるワークスペース通知ファイル <code>Codemgr_wsdata/notification</code> の読み取り中にエラーが発生しました。通知対象とするファイル/ディレクトリのリストを区切る END 文がありません。ワークスペース管理ツールのコマンドは通知の設定を正しく解析することができないので、通知ファイルに通知要求を設定している場合でも、その通知は送信されません。</p> <p>対処法: 通知ファイルに END 文を正しく入力してください。通知ファイルの形式についての詳細は、<code>notification</code>(4) のマニュアルページまたは 40 ページの「他のユーザーへのトランザクションの実行通知」を参照してください。</p>
2055	ファイル <ファイル名> には不完全なデルタテーブルがあります
	<p>意味: SCCS 履歴ファイル <ファイル名> のデルタテーブルが不完全です。ファイルが壊れている可能性があります。</p> <p>対処法: ファイルを修正するか、新しいバージョンをコピーしてください。</p>
2056	形式の不正な行があります。 <ファイル名>: <行番号>
	<p>意味: ワークスペース管理ツールのコマンドが、ブリングオーバーまたはプットバックトランザクションが中止された結果残された一時ログファイルを読み取っているときに、形式が正しくない行を検出しました。このメッセージは、ファイルが壊れていることを示しています。</p> <p>対処法: <code>workspace updatenames</code> コマンドを使用して、名前テーブルを作成しなおしてから、コマンドを再実行してください。</p>

表 A-1 エラーメッセージ (続き)

2057	大きさ 0 の SCCS ファイルがあります <ファイル名> 意味: データが含まれていない SCCS 履歴ファイルが検出されました。 対処法: 該当する SCCS 履歴ファイルを削除してください。
2058	チェックインされるまで子ファイルのバージョンは得られません 意味: 衝突解決トランザクションの実行中に、SCCS にチェックインされていないファイルが検出されました。ファイルをチェックインしなければ衝突の解決はできません。 対処法: ファイルをチェックインし、トランザクションを再開してください。
2059	名前履歴のシリアル番号 <番号> の順序がファイル <ファイル名> で間違っています 意味: SCCS 履歴ファイル <ファイル名> 内の名前変更に関する情報が壊れています。この SCCS ファイル内の名前履歴レコードは番号が降順に並んでいません。 対処法: 名前履歴レコードを並べ直すか、プリングオーバーまたはプットバックトランザクションを使用してファイルの新しいバージョンをコピーしてください。
2060	ファイル <ファイル名> 中のデルタシリアル番号 <番号> の順序が不正です 意味: デルタ番号の順序が、SCCS 履歴ファイル <ファイル名> 内で降順になっていません。これは、ファイルが壊れていることを示しています。 対処法: デルタ番号を並べ直すか、プリングオーバーまたはプットバックトランザクションを使用してファイルの新しいバージョンをコピーしてください。
2061	DISPLAY 環境変数を設定してファイルマージを起動してください 意味: ファイルマージツールは XWindows プログラムです。DISPLAY 環境変数を使用して、ウィンドウの表示場所を決定します。通常、CDE または OpenWindows が実行されているサーバーでは、この変数は自動的に設定されます。その他の XWindow サーバーでは、ファイルマージツールの起動時にこの変数を手動で設定してください。 対処法: サーバーの設定を確認して、DISPLAY 環境変数を設定してください。
2062	ファイル <ファイル名> は書き込み可能なため衝突解決できません 意味: ファイル <ファイル名> は SCCS からチェックアウトされていませんが、ファイルのパーミッションは書き込み可能な状態になっています。この衝突を解決すると、チェックアウトされていないファイルに書き込みが行われます。 対処法: ファイルのパーミッションを変更し (たとえば、ファイルをチェックアウトした後、再度チェックインする)、もう一度衝突解決トランザクションを実行してください。
2063	ワークスペース <ワークスペース名> は作成できませんでした。ワークスペース <ワークスペース名> の中で入れ子になっている可能性があります 意味: 既存のワークスペースの下に新しいワークスペースを作成しようとして失敗しました。 対処法: 新しいワークスペースは、既存のワークスペースの階層の外で作成してください。

表 A-1 エラーメッセージ (続き)

2064	<p>シンボリックリンクのワークスペースを削除することはできません。 <code>"workspace delete <ワークスペース名>"</code> を実行してください</p> <p>意味: ワークスペース管理ツールのコマンドは、シンボリックリンクであるディレクトリやファイルを削除しません。そのファイルの実体を削除しなければなりません。このためのコマンドが用意されています。</p> <p>対処法: <code>workspace delete</code> コマンドを使用して <ワークスペース名> を削除してください。</p>
2065	<p>このエラーメッセージは、以下の各形式で出されます。</p> <p>ユーザー <ユーザー名> には、ワークスペース <ワークスペース名> からブリングオーバーする許可が与えられていません</p> <p>ユーザー <ユーザー名> には、ワークスペース <ワークスペース名> ヘブリングオーバーする許可が与えられていません</p> <p>ユーザー <ユーザー名> には、ワークスペース <ワークスペース名> からプットバックする許可が与えられていません</p> <p>ユーザー <ユーザー名> には、ワークスペース <ワークスペース名> ヘプットバックする許可が与えられていません</p> <p>ユーザー <ユーザー名> には、ワークスペース <ワークスペース名> の取り消しをする許可が与えられていません</p> <p>ユーザー <ユーザー名> には、ワークスペース <ワークスペース名> を削除する許可が与えられていません</p> <p>ユーザー <ユーザー名> には、ワークスペース <ワークスペース名> を移動する許可が与えられていません</p> <p>ユーザー <ユーザー名> には、ワークスペース <ワークスペース名> の親を変更する許可が与えられていません</p> <p>ユーザー <ユーザー名> には、親をワークスペース <ワークスペース名> に変更する許可が与えられていません</p> <p>意味: ユーザー <ユーザー名> が、ワークスペース <ワークスペース名> に影響する処理を実行しようとした。<ワークスペース名> のアクセス権では、<ユーザー名> がその処理を実行することはできません。</p> <p>対処法: <ワークスペース名>/<code>Codemgr_wsdata/access_control</code> は、ワークスペースに対するさまざまな操作のアクセス権を指定するテキストファイルです。該当操作を実行できるようにするには、そのワークスペースの所有者が <ユーザー名> に対するアクセス権の設定を変更する必要があります。アクセス権の設定は、「ワークスペース管理」ウィンドウから「ワークスペース」を選択して「ワークスペース属性」ダイアログの「アクセス制御」タブを選択するか、<code>access_control</code> ファイルの内容を直接編集して変更します。詳細は、第 4 章または <code>access_control(4)</code> のマニュアルページを参照してください。</p>

表 A-1 エラーメッセージ (続き)

2066	<p>ファイル <ファイル名> が壊れています。 <行番号> 行目のパス名に空白が含まれていません。通知は送られませんでした</p> <p>意味: ワークスペース通知ファイル <code>Codemgr_wsdata/notification</code> の読み取り時に出されるエラーで、パス名が記述される行に空白文字があったことを示しています。</p> <p>対処法: <code>Codemgr_wsdata/notification</code> ファイルの内容を編集して、該当行から空白文字を削除してください。このファイルの形式についての詳細は、<code>notification(4)</code> のマニュアルページまたは 40 ページの「他のユーザーへのトランザクションの実行通知」を参照してください。</p>
2067	<p>ファイル <ファイル名> が壊れています。 <行番号> 行目に通知イベントがありません。通知は送られませんでした。</p> <p>意味: ワークスペース通知ファイル <code>Codemgr_wsdata/notification</code> の読み取り時に出されるエラーで、ワークスペース管理ツールのイベント (たとえば、プリングオーバー) が指定されていないことを示しています。</p> <p>対処法: <code>Codemgr_wsdata/notification</code> ファイルの内容を編集して、現在のイベントを追加してください。</p>
2068	<p>ブットバック: ファイル <ファイル名> の SCCS エラーです。ロックファイルを作成できません</p> <p>意味: ほかのユーザーが SCCS ファイルまたは p- ファイルを更新中であるか、SCCS ファイルが存在するディレクトリへの書き込み権がありません。</p> <p>対処法: ほかのユーザーが SCCS ファイルまたは p- ファイルを更新中の場合は、そのユーザーがロックを解除するのを待つてファイルへのアクセスを再度試みてください。ディレクトリへの書き込み権がない場合は、書き込み権を得ない限り、そのディレクトリにはロックファイルを作成できません。</p>
2069 - 2073	未使用
2074	<p>ワークスペース <ワークスペース名> にロックファイルがありません</p> <p>意味: ロックが存在しないワークスペースからロックを解除しようとしてしました。</p>
2075	<p>ロック <ロック名> はワークスペース <ワークスペース名> で見つかりません</p> <p>意味: <code>workspace locks -r</code> コマンドの使用時に指定されたロック番号がロックリストに存在しませんでした。</p> <p>対処法: <code>workspace locks</code> コマンドを使用してワークスペースのロック番号を調べ、有効な番号を入力してください。</p>
2076	<p>内部エラー: Cannot find the directory in which command <code>command_name</code> is located because <code>avo_find_dir_init()</code> has not been called</p> <p>意味: 内部エラー。</p>

表 A-1 エラーメッセージ (続き)

	対処法: ご購入先にお問い合わせください。
2077	<p><番号> は正しい番号ではありません</p> <p>意味: <code>resolve</code> コマンドを使用している際に、リストされている番号以外の番号が参照されました。</p> <p>対処法: 値リストを表示して正しい番号を調べてください。</p>
2078	<p>ワークスペース <ワークスペース名> にアクセスできません</p> <p>意味: <ワークスペース名> のパーミッションの設定が、指定されたワークスペース管理ツールのコマンドによるアクセスを禁止しています。</p> <p>対処法: <ワークスペース名> ディレクトリのパーミッションを確認してください。デフォルトのパーミッションは <code>777</code> です。</p>
2079	<p>ファイル <ファイル名> の名前履歴の構文を解析できません。行に含まれるもの: <テキスト></p> <p>意味: SCCS 履歴ファイル <ファイル名> 内の名前履歴レコードにフォーマットエラーがあります。このメッセージには、問題のテキストが表示されます。</p> <p>対処法: 可能であれば、そのレコードを修正します。あるいは、プリングオーバーまたはプットバックトランザクションを使用してファイルの新しいバージョンをコピーしてください。</p>
2080	<p>バックアップディレクトリ <ディレクトリ名> は削除も名前変更もできませんでした</p> <p>意味: ワークスペース管理ツールが、新しいトランザクションをバックアップできるようにするために、バックアップ領域 <ディレクトリ名> を消去しようとした。ワークスペース管理ツールが、この方法でファイルやディレクトリを削除したり名前の変更を行ったりすることができませんでした。そのディレクトリに対するパーミッションが変更されている可能性があります。</p> <p>対処法: <ディレクトリ名> のディレクトリのパーミッションを確認してください。このディレクトリのデフォルトのパーミッションは <code>777</code> です。</p>
2081	<p>ファイル <ファイル名> の <code>sccs</code> エラーです。ロックファイルを作成できません (cm4)</p> <p>意味: ファイルが SCCS 管理下でないか、または SCCS ファイルが壊れているためにロックファイルを作成することができませんでした。</p> <p>対処法: ご購入先にお問い合わせください。</p>
2082	<p>ワークスペース <ワークスペース名 1> の親 <ワークスペース名 2> はファイルシステムに存在しません</p> <p>意味: 親ワークスペースが使用マシンにマウントされていない、あるいは存在しません。</p> <p>対処法: 実行マシンに親ワークスペースをマウントしてください。</p>

表 A-1 エラーメッセージ (続き)

2083	<p>ワークスペース <ワークスペース名 1> の子 <ワークスペース名 2> はファイルシステムに存在しません</p> <p>意味: 子ワークスペースが使用マシンにマウントされていない、あるいは存在しません。</p> <p>対処法: 実行マシンに子ワークスペースをマウントしてください。</p>
2084	<p><code>codemgrtool: internal error in arg_strlist_from_wsname () : NULL arg_list</code></p> <p>意味: 内部エラー。</p> <p>対処法: ご購入先にお問い合わせください。</p>
2085	<p><code>codemgrtool: internal error in arg_strlist_from_wsname () : NULL undo_list</code></p> <p>意味: 内部エラー。</p> <p>対処法: ご購入先にお問い合わせください。</p>
2086	<p><code>codemgrtool: path_name doesn't start with a /</code></p> <p>意味: 内部エラー。</p> <p>対処法: ご購入先にお問い合わせください。</p>
2087	未使用
2088	<p>以下の 2 つの履歴ファイルが同一の初期デルタをもっているため、ワークスペース <ワークスペース名> の名前テーブルを読み込むことができません。</p> <p><ファイル名> <ファイル名></p> <p>以下のコマンドを実行した後、再度 <コマンド名> コマンドを実行してください。</p> <p><code>path_name/workspace updatenames <ワークスペース名></code></p> <p>意味: <code>cp</code> コマンドによってワークスペース内で SCCS 履歴ファイルがコピーされました。このため、同じルートデルタを含むファイルが 2 つ存在しています。ワークスペース管理ツールは、ルートデルタを使用して 2 つのファイルを区別します。ワークスペース管理ツールが 2 つのファイルを区別できるようにするには、<code>workspace updatenames</code> コマンドを使用します。</p> <p>対処法: <code>workspace updatenames</code> コマンドを実行してから、エラーとなったコマンドを再度実行してください。</p>
2089	<p>ワークスペースを <パス名> へ移動できません。</p> <p><パス名> へのシンボリックリンクです。</p> <p>シンボリックリンクではないワークスペース名を使用してください。</p>

表 A-1 エラーメッセージ (続き)

意味: ワークスペース管理ツールのコマンドは、シンボリックリンクであるディレクトリやファイルを移動しません。

対処法: ワークスペースをシンボリックリンクでない名前に移動してください。

2090

以下の 2 つの履歴ファイルが同一の初期デルタをもっているため、ワークスペース <ワークスペース名> の名前テーブルへ書き込むことができません。

<ファイル名>

<ファイル名>

以下のコマンドを実行した後、再度 <コマンド名> コマンドを実行してください。

`path_name/workspace updatenames <ワークスペース名>`

意味: `cp` コマンドによってワークスペース内で SCCS 履歴ファイルがコピーされました。このため、同じルートデルタを含むファイルが 2 つ存在しています。ワークスペース管理ツールは、ルートデルタを使用して 2 つのファイルを区別します。ワークスペース管理ツールが 2 つのファイルを区別できるようにするには、`workspace updatenames` コマンドを使用します。

対処法: `workspace updatenames` コマンドを実行してから、エラーとなったコマンドを再度実行してください。

2091

内部エラー: `hash table missing entry`

意味: 内部エラー。

対処法: ご購入先にお問い合わせください。

2092

SCCS ファイル (A) を (ファイル B) にコピーしました。元の SCCS ファイル (A) が見つかりません。

次のコマンドを実行後、もう一度 <コマンド名> コマンドを実行してください。

`path_name/workspace updatenames <ワークスペース名>`

意味: ワークスペース内で SCCS 履歴ファイルが `cp` コマンドでコピーされました。この後、オリジナルのファイル (A) が名前を変更されているか、ワークスペースから削除されています。この場合、ワークスペース管理ツールは、これらのファイルの名前が変更されているかどうか (どの名前に) およびワークスペースから削除されているかどうかを判別することができません。`workspace updatenames` コマンドは、可能性のある変更後の名前を対話形式で表示し、ファイルの現在の状態 (新しい名前、存在の有無など) を指示するよう求めてきます。これにより、ワークスペース管理ツールはワークスペース階層全体にわたって変更を正しく伝えるようになります。

対処法: `workspace updatenames` コマンドを実行してから、エラーとなったコマンドを再度実行してください。

2093

内部エラー: `SmIDs not equivalent`

意味: 内部エラー。

表 A-1 エラーメッセージ (続き)

対処法: ご購入先にお問い合わせください。

2094 内部エラー: SmID not found

意味: 内部エラー。

対処法: ご購入先にお問い合わせください。

2095 - 2499 未使用

2500 - 2600 内部エラー。

意味: 2500 から 2600 までのエラー番号はすべてワークスペース管理ツールの内部エラーを表わし、ユーザー側で対処することはできません。したがって、これらのエラーが発生した場合には、ご購入先にお問い合わせください。

警告メッセージ

表 A-2 に、警告メッセージと、その意味および対処法を示します。

表 A-2 警告メッセージ

2601 **バックアップディレクトリ <旧ディレクトリ名> を削除できなかったため <新ディレクトリ名> に名前を変更しました**

意味: ワークスペース管理ツールが、新しいトランザクションのバックアップを行うことができるように、バックアップ領域 <旧ディレクトリ名> を消去しようとした。しかし、このバックアップディレクトリを消去することができなかったため、<新ディレクトリ名> という名前に変更しました。これは、ディレクトリのパーミッションが変更されたために生じたエラーと考えられます。

対処法: <旧ディレクトリ名> のパーミッションを調べてください。このディレクトリのデフォルトのパーミッションは 777 です。また、<新ディレクトリ名> の内容を削除してください。

2602 **ファイル <ファイル名> は sccs 管理下にもワークスペース中にもありません - 無視しました**

意味: ワークスペース管理ツールは、どちらのワークスペースからも SCCS 履歴ファイル <ファイル名> を見つけることができませんでした。

対処法: 入力されたファイル名が間違っていると考えられます。正しいファイル名でコマンドを再度実行してください。

2603 **ファイル名の長さが 0 です - 無視しました**

表 A-2 警告メッセージ (続き)

	<p>意味: コマンド行の引数として (または <code>/Codemgr_wsdata/args</code> ファイルの中で) 指定されたファイル名に文字が存在しません ("").</p> <p>対処法: 正しいファイル名引数を指定して、コマンドを再度実行してください。それでもなお問題が解決されない場合は、<code>args</code> ファイル内にリストされている引数を調べてください。</p>
2604	<p>ファイル名 <ファイル名> 中に空白文字があります - 無視しました</p> <p>意味: コマンド行の引数として (または <code>/Codemgr_wsdata/args</code> ファイルの中で) 指定されたファイル名に空白文字が含まれています。ワークスペース管理ツールは、空白文字を含むファイル名を受け付けません。</p> <p>対処法: 正しいファイル名引数を指定して、コマンドを再度実行してください。それでもなお問題が解決されない場合は、<code>args</code> ファイル内にリストされている引数を調べてください。</p>
2605	未使用
2606	<p>ファイル <ファイル名> はプリングオーバーされませんでした。 <ファイルタイプ 1> がワークスペース <ワークスペース名 1> に存在し、 <ファイルタイプ 2> がワークスペース <ワークスペース名 2> にも存在しています。</p> <p>意味: ファイル名のファイルタイプが、親と子で異なっています (通常のファイル、ディレクトリ、シンボリックリンクなど)。</p> <p>対処法: メッセージに示されているファイルを同じファイルタイプに変更するか、一方の名前を変更してください。</p>
2607	未使用
2608	<p>ワークスペース <子ワークスペース名> は <親ワークスペース名> の子です。親のファイルを更新できませんでした</p> <p>意味: <子ワークスペース> に対する <code>workspace delete</code> または <code>workspace move</code> コマンドの実行時に、該当ワークスペースの親 (<親ワークスペース>) 内の <code>children</code> ファイルに、この親の子として <子ワークスペース> を指定しているエントリが含まれていないことが検出されました。</p> <p>対処法: コマンドはこの不一致を訂正しますが、親の <code>children</code> ファイルが壊れている可能性があります (これは単なる警告で対処法ではありません)。</p>
2609	未使用
2610	<p><パス名> はワークスペースではありません</p> <p>意味: コマンドで指定されたディレクトリはワークスペース管理ツールのワークスペースではありません。ワークスペース管理ツールのワークスペースであるかどうかは、最上位ディレクトリに <code>Codemgr_wsdata</code> ディレクトリが存在するかどうかで判別されます。</p>

表 A-2 警告メッセージ (続き)

	<p>対処法: 別のワークスペース名を指定するか、CLI の <code>workspace create</code> コマンドを使用するか、GUI の「ワークスペース管理」ウィンドウの「ファイル」メニューから「ワークスペース作成」を選択し、ディレクトリをワークスペースに変換してください。</p>
2611	<p><ファイル名> はどちらのワークスペースにも存在しません - 無視しました</p> <p>意味: <ファイル名> が親ワークスペースにも子ワークスペースにも存在しません。</p> <p>対処法: 名前が正しく指定されているかどうかを確認してください。</p>
2612	未使用
2613	<p>ファイル名 <ファイル名> 中にパス構成要素 ".." が多過ぎます - 無視しました</p> <p>意味: このメッセージが出力される原因としては、次のものが考えられます。</p> <ul style="list-style-type: none"> • ワークスペース管理ツールのコマンドがパス名をワークスペース相対パス名に変換できない。 • ワークスペース管理ツールのコマンドがパス名を完全指定のワークスペース名に変換できない。 <p>対処法: パス名の指定時に、パス名の要素として ".." の使用を制限するか、あるいはまったく使用しないでください。</p>
2614	<p>ファイル <ファイル名> の <行番号> 行目が長すぎるか、期待していないファイルの終りがあります</p> <p>意味: <code>Codemgr_wsdata/nametable</code> ファイルの読み取り時に、ワークスペース管理ツールのコマンドがバッファに入れるには多すぎる文字数が検出されました。行の最大長は 1024 文字です。これは、<code>nametable</code> が壊れている可能性があることを示しています。</p> <p>対処法: ワークスペース管理ツールは自動的に名前テーブルを再作成します。これには若干の時間を要します。</p>
2615	<p>ファイル <ファイル名> の <行番号> 行目の書式が正しくありません</p> <p>意味: このメッセージは、ファイル <code>Codemgr_wsdata/nametable</code> が壊れていることを示しています。</p> <p>対処法: ワークスペース管理ツールは自動的に名前テーブルを再作成します。これには若干の時間を要します。</p>
2616	未使用
2617	<p>名前テーブルの編集ログのレコード型 <タイプ番号> が不正です <ファイル名> - 無視します</p> <p>意味: プリンクオーバーやブットバックが途中で異常終了したときに残された一時的なログファイルを、ワークスペース管理ツールのコマンドが読み取ろうとしたときに、形式が正しくないレコードを検出しました。これはファイルが壊れていることを示しています。</p>

表 A-2 警告メッセージ (続き)

	<p>対処法: <code>workspace updatenames</code> コマンドを実行して <code>nametable</code> を作成しなおしてから、コマンドを再度実行してください。</p>
2618	<p><ファイル名> を開けません。 - 通知メールを送信できません</p> <p>意味: ワークスペース管理ツールの通知機能が <ファイル名> を開くことができませんでした。このため、現在の操作に関する通知メールは送出されません。</p> <p>対処法: <ファイル名> のファイルパーミッションを確認してください。</p>
2619	未使用
2620	<p>通知メールを送るプロセスを <code>fork</code> できません</p> <p>意味: システム資源 (メモリー、スワップ領域) の不足により、ワークスペース管理ツールの通知機能は通知メールを送信できませんでした。</p> <p>対処法: システム資源の状態を確認してください。</p>
2621	未使用
2622	<p>ファイル名 <ファイル名> はコメント文字 (<コメント文字>) を含んでいます - 無視しました</p> <p>意味: コマンド (または <code>Codemgr_wsdata/args</code> ファイル) に引数として指定したファイル名に、<コメント文字> が含まれています。ワークスペース管理ツールはこの文字をコメントを表わすための文字として認識します。</p> <p>対処法: <コメント文字> が含まれない名前に変更してください。それでもなお問題が解決されない場合は、<code>args</code> ファイルにリストされている引数を調べてください。</p>
2623	<p>ワークスペース <ワークスペース名> に読み込みロックが残っています ワークスペース <ワークスペース名> に書き込みロックが残っています</p> <p>意味: ワークスペース管理ツールは、<ワークスペース名> にあるロックを削除できませんでした。ディスク領域が足りないか、または <code>Codemgr_wsdata/locks</code> ファイルのパーミッションが変更されています。</p> <p>対処法: ワークスペース管理ツールの「オプション」メニューから「ワークスペース」を選択して「ワークスペース属性」ダイアログの「カテゴリ」リストボックスから「ロック」を選択するか、または <code>workspace locks</code> コマンドを実行して、ロックを削除してください。</p>
2624	<p>ファイル <ファイル名> はワークスペース <ワークスペース名> 中でチェックアウト中です。 チェックアウトされるファイル中の変更はブリングオーバーされません。</p> <p>意味: <ファイル名> が親ワークスペース内でチェックアウトされています。ブリングオーバーランザクションの実行時に g-ファイル内の変更内容はコピーされなかったため、注意してください</p>

表 A-2 警告メッセージ (続き)

	対処法: なし。
2625	<p>ファイル <ファイル名> は衝突していません。SCCS ファイルの同期がとれています。衝突ファイルリストから削除します</p> <p>意味: SCCS 履歴ファイル内の情報は、そのファイルに未解決の衝突が存在しないことを示していますが、ワークスペース内の <code>Codemgr_wsdata/conflicts</code> ファイルには衝突が存在するファイルとしてこのファイルがリストされています。コマンドは <code>conflicts</code> ファイルからこのファイルを削除します。</p> <p>対処法: なし。</p>
2626	<p>ファイル <ファイル名> はワークスペース <ワークスペース名> で未解決のため、プリングオーバーできませんでした。</p> <p>意味: <ファイル名> は、<ワークスペース名> の中で未解決の衝突が残っているため、プリングオーバーされませんでした。</p> <p>対処法: GUI の衝突解決トランザクション、または CLI の <code>resolve</code> コマンドを使用してその衝突を解決してから、プリングオーバートランザクションを再度実行してください。</p>
2627	<p>ディレクトリ <ディレクトリ名> は読み取り専用でマウントされています</p> <p>意味: ワークスペース管理ツールは、プリングオーバーまたはブットバックのトランザクションを行う前に、使用するワークスペースの最上位ディレクトリが書き込み可能であるかどうかをチェックします。ワークスペース内の最上位ディレクトリよりも下のディレクトリが、別の領域から書き込み可能としてマウントされていることもあるため、これはエラーではありません。このメッセージは、ワークスペース最上位ディレクトリのパーミッションが正しく設定されていないことを警告するためのものです。</p> <p>対処法: 書き込みが拒否されてしまう場合は、ディレクトリのパーミッションを変更してください。</p>
2628	<p>チェックアウトコマンドが <code>PATH <コマンドパス></code> がないため、ファイルは更新されません。</p> <p>意味: <code>SCCS get</code> コマンドが検索パス上に存在せず実行できなかったため、g-ファイルをプリングオーバーまたはブットバックのトランザクションで更新できませんでした。</p> <p>対処法: トランザクション実行時に g-ファイルを更新する必要がある場合には、検索パスに <code>get</code> コマンドのパスを含めるようにしてください。</p>

表 A-2 警告メッセージ (続き)

2629	<p>ToolTalk 接続を確立ができなかったため、ファイルマージは起動できません</p> <p>意味: <code>resolve</code> コマンドが ToolTalk メッセージサービスに接続できませんでした。ToolTalk サービスは、衝突解決コマンドがファイルマージツールとの通信を行うときに使用されます。</p> <p>対処法: ToolTalk サービスは通常、CDE または OpenWindowsのインストール時に一緒にインストールされます。CDE または OpenWindowsのマニュアルを参照して、ToolTalk サービスが存在しない、あるいは応答しない理由を調べてください。</p>
2630	<p>このワークスペースは既存のディレクトリに作られます</p> <p>意味: 既存のディレクトリをワークスペース管理ツールのワークスペースに変換しています。既存のディレクトリ階層からワークスペースを作成するということは、最上位ディレクトリに <code>Codemgr_wsdata</code> メタデータディレクトリを作成することです。ディレクトリがワークスペースになると、その内容は <code>workspace delete</code> コマンドを使用して削除できます。</p> <p>対処法: なし。</p>
2631	<p>ファイル <ファイル名> をプリングオーバーできませんでした。 チェックアウトされていますが、ワークスペース <ワークスペース名> の中で書き込み許可がありません。</p> <p>意味: ファイル <ファイル名> が、子ワークスペース <ワークスペース名> 内でチェックアウトされ (p-ファイルが存在する)、書き込み可能でないため、プリングオーバーアクションでプリングオーバーされませんでした。このメッセージは、該当ファイルを処理しない方が安全であることを示しています。</p> <p>対処法: 書き込み権を SCCS の状態に合わせて変更してください。</p>
2632	<p>名前変更でエラーが発生したため、ファイル <ファイル名> の内容変更を行いません</p> <p>意味: <ファイル名> の名前の処理時にエラーが発生しました。このため、コピー元のソースワークスペースのファイルに対する変更内容を宛先ワークスペースにコピーできませんでした。</p> <p>対処法: 名前変更の問題を修正し、ワークスペース管理ツールのアクションを再度実行してください。</p>
2633	<p>パス名 <パス名> はシンボリックリンクを含んでいます - 無視しました</p> <p>意味: トランザクション実行中にシンボリックリンクを検出しました。そのディレクトリは無視されます。トランザクションは継続されます。</p> <p>対処法: Sun WorkShop TeamWare ワークスペース内では、シンボリックリンクを使用しないでください。</p>

用語集

`Codemgr_wsdata` ディレクトリ

TeamWare の各ワークスペースには、そのルートディレクトリに `Codemgr_wsdata` という名前のメタデータディレクトリがあります。ワークスペース管理ツールは、ワークスペースに関するデータをこの `Codemgr_wsdata` ディレクトリに格納します。 `Codemgr_wsdata` ディレクトリが存在するかどうかで、通常のディレクトリか、ワークスペース管理ツールのワークスペースが判断できます。詳細は、173 ページの「ワークスペースのメタデータディレクトリ」を参照してください。

`def.dir.flp`

ワークスペース管理ツールに付属しているデフォルトの FLP (ファイルリストプログラム)。この FLP はディレクトリ階層を再帰的に下り、対応する SCCS 履歴ファイルが存在するすべてのファイルの一覧を示します。「FLP」の項目も参照してください。

FLP

File List Program (ファイルリストプログラム) の略。ファイルリストを生成して標準出力 (`stdout`) に出力するプログラムまたはスクリプトです。ワークスペース管理ツールは、このファイルリストをプリンクオーバーおよびプットバックトランザクションで使用します。

g-ファイル (SCCS 用語)

`sccs get` コマンドによって SCCS 履歴ファイルから取り出されたファイルの作業用コピー。

MR

Modification Request (変更要求) の略。ファイルをチェックインする前に、バージョン管理ツールから入力求められるパスワードです。

RTI

Request To Integrate (統合要求) の略で、ワークスペース履歴ファイルに統合要求 ID を記録するために使用される見出しのことです。

SCCS デルタ ID (SID)

SID は、特定のデルタを表わすときに使用する番号です。ID は、2 つの部分から構成され、その間はドット (.) で区切られています。デフォルトでは、最初のデルタの SID は 1.1 になります。SID の最初の部分はリリース番号を、2 番目の部分はレベル番号を表わします。デルタをチェックインするたびに、レベル番号は自動的に「1」ずつ加算されます。

SCCS ファイル属性

個々のファイルに割り当て可能な属性です。

SCCS マージ可能 ID (SMID)

SMID は、Xerox Secure Hash Function を使用して生成される番号です。SID が変更された場合でも、すべてのデルタを一意に識別できます。

SCCS 履歴ファイル

与えられたファイルのデルタ履歴が格納されているファイル。ファイル名の先頭に s. が付くことから「エスドットファイル」とも呼ばれます。SCCS 履歴ファイルは SCCS ディレクトリに置かれます。SCCS ディレクトリは、g-ファイルと同じディレクトリに置かれます。「g-ファイル (SCCS 用語)」の項目も参照してください。

SID

「SCCS デルタ ID (SID)」を参照してください。

SMID

「SCCS マージ可能 ID (SMID)」を参照してください。

アクセス制御

ワークスペースへのアクセスを制御するために使用するワークスペース管理機能。

アンチェックアウト

直前のチェックアウト前の状態にファイルを戻すことです。

親の変更 (Reparent)

子ワークスペースの親を変更することです。

親ワークスペース

子ワークスペースを持つワークスペース。親ワークスペースは通常、統合領域として使用されます。これに対し、子ワークスペースは、開発、テスト、衝突解決などの作業に使用されます。

更新

1 つまたは複数のファイルが転送元ワークスペースと転送先ワークスペースの両方に存在し、転送元ワークスペース側で変更があった場合に、ブリングオーバーまたはプットバックトランザクションでそれらのファイルを更新することを言います。転送先ワークスペース上の SCCS 履歴ファイルは転送元ワークスペースから得られる新しいデルタで更新されます。

更新ブリングオーバー (Bringover Update)

既存の子ワークスペースの内容を、その親ワークスペース上に存在するファイルで更新するときに使用されるトランザクション。ワークスペース管理ツールのトランザクションはすべて、子ワークスペース側の視点から実行されます。このため、更新ブリングオーバートランザクションは親ワークスペースから子ワークスペースにファイルを「引き渡す (ブリングオーバー)」と言います。「作成ブリングオーバー (Bringover Create)」、「ワークスペース」、および「プットバック (Putback)」の項目も参照してください。

コピー / 変更 / マージ

並列的に共同開発を行うための作業モデルのことで、ワークスペース管理ツールの基礎となります。これによって、複数の開発者はそれぞれ共通の統合領域からソースをコピーし、そのソースに独自の変更を加え、自分が加えた変更をほかの共同開発者が加えた変更内容とマージする、という一連の開発作業を進めます。

子ワークスペース

`Codemgr_wsdata/parents` ファイルに示されている親ワークスペースを持つワークスペース。開発作業は通常、子ワークスペース内で行われ、テスト終了後、親ワークスペースに戻されます(ブットバック)。ワークスペース管理ツールのトランザクションは子ワークスペース側の視点から実行され、すべての衝突は子ワークスペースで解決されます。

ルートディレクトリ

ワークスペース管理ツールのワークスペースの最上位ディレクトリ。このディレクトリのパス名が、ワークスペースを参照するための名前となります。

作成 (Create)

ワークスペース管理ツールのトランザクション出力で使用される用語です。あるファイルが転送元 (ソース) ワークスペース上には存在するが転送先ワークスペース上には存在しない場合、ブリングオーバーまたはブットバックのトランザクション実行時に、そのファイルが転送先ワークスペースにコピーされたとき、そのファイルは「作成」されたと言います。

作成ブリングオーバー (Bringover Create)

ファイルを親ワークスペースから、存在しない子ワークスペースにコピーするときに使用されるトランザクション。このトランザクションを実行すると、指定された子ワークスペースが新たに作成されます。ワークスペース管理ツールのトランザクションはすべて、子ワークスペース側の観点から実行されます。このため、作成ブリングオーバートランザクションは親ワークスペースから子ワークスペースにファイルを「引き渡す (ブリングオーバー)」と言います。「更新ブリングオーバー (Bringover Update)」、「ワークスペース」、および「ブットバック (Putback)」の項目も参照してください。

衝突

あるファイルが子ワークスペースと親ワークスペースの両方で変更されたときに生じる状態。衝突は更新ブリングオーバートランザクションによってその存在が確認され、衝突解決トランザクションによって解決します。

衝突解決

2 つの衝突したデルタから新しいデルタを作成すること。「マージ」および「衝突」の項目も参照してください。

通知

ファイルやディレクトリの変更などのイベントが発生したことをユーザにメールで通知する、ワークスペース管理機能の 1 つです。

デフォルトのパス

SCCS 履歴ファイル内の次のデルタが追加される分岐です。履歴ファイル中、この分岐は実線で示されます。

デルタ

あるファイルの 2 つのバージョンの差分について SCCS 履歴ファイルに収録されたデータの集まり。ファイルをチェックインすると、SCCS は、チェックインされたファイルとそのファイルの前バージョン間の行ごとの違いだけを記録します。この一連の差分をデルタと呼びます。最初にチェックアウトした時点のバージョンのファイルは、一連の累積されたデルタから構成されています。デルタとバージョンという用語は、多くの場合同義語として使用されますが、意味は同じではありません。特定のデルタを除外したバージョンを抽出できます。「バージョン」の項目も参照してください。

統合要求 ID

ワークスペース管理がプットバックを許可する前に、プットバック妥当性検査によって入力及要求されるパスワード。

統合ワークスペース

複数の開発者がそれぞれの作業結果をプットバックする場所となるワークスペース。

取消し (Undo)

直前に実行されたプリングオーバーやプットバックトランザクションを無効にして、ワークスペースを直前の状態に戻すこと。

バージョン

ファイルをチェックインすると、SCCS は、チェックインされたファイルとそのファイルの前バージョン間の行ごとの違いだけを記録します。この一連の相違をデルタと呼びます。最初にチェックアウトした時点のバージョンのファイルは、一連の累積されたデルタから構成されています。デルタおよびバージョンという用語は、多くの場合同義語として使用されますが、意味は同じではありません。特定のデルタを除外したバージョンを抽出できます。「デルタ」の項目も参照してください。

バージョン管理

Sun Workshop TeamWare ファイルを管理するための TeamWare ツール。第 5 章を参照してください。

ファイルマージ

衝突解決トランザクション実行時にデルタをマージするために使用される Sun Workshop TeamWare ツール。第 6 章を参照してください。

プットバック (Putback)

子ワークスペース上に存在するファイルによって親ワークスペースを更新するために使用するトランザクション。ワークスペース管理ツールのトランザクションはすべて、子ワークスペース側の観点から実行されます。このため、プットバックトランザクションは子ワークスペースから親ワークスペースにファイルを「戻す (プットバック)」と言います。「作成ブリングオーバー (Bringover Create)」、「更新ブリングオーバー (Bringover Update)」、「ワークスペース」の項目も参照してください。

プットバックの妥当性検査

特定のワークスペースに対するプットバックを許可するかどうかを制御することを可能にするワークスペース管理機能です。

分岐 (SCCS 用語)

SCCS 履歴ファイルの主デルタツリーから枝分かれしたデルタまたは一連のデルタです。

フリーズポイントツール

重要な箇所ではワークスペース (またはその一部) のスナップショットをとるために使用される Sun Workshop TeamWare ツールです。第 8 章を参照してください。

マージ

内容に相違が存在する 2 つのファイル (デルタ) から、相違を統合した 1 つのファイルを作成すること。通常、この処理はファイルマージプログラムを使用して行います。

履歴ファイル

初めてファイルを SCCS の管理下に置くと、その新しい SCCS ファイル用に履歴ファイルが作成されます。履歴ファイルの初期バージョンには、ソースファイルの全テキストがそのまま使用されます。この後作成されるデルタは、この初期の履歴ファイルと比較されます。履歴ファイルは、その接頭辞 (.s) から s. ファイル (エスドットファイル) とも呼ばれます。

ロック

整合性を保持するため、ワークスペース管理ツールのプリングオーバーおよびブットバックトランザクションは、その処理中にワークスペースをロックします。ロック状態は、各ワークスペース内の [Codemgr_wsdata/lock](#) ファイルに記録されます。ワークスペース管理ツールのコマンドは、ワークスペースに対する実行前にワークスペースのこのファイルを確認します。

ワークスペース

Sun WorkShop の管理対象として指定されたディレクトリであり、サブディレクトリやファイルが含まれています。通常、プロジェクトの各開発者は各自専用のワークスペースで並行してプログラミング作業を行います。ワークスペース管理には、ワークスペースを管理するための各種ユーティリティが用意されています。第 2 章を参照してください。

ワークスペース階層

プログラマやリリースエンジニアがソフトウェア製品の開発、テスト、共有、リリースを行えるようにする親子ワークスペースの階層です。

ワークスペース管理

Sun WorkShop TeamWare ワークスペースの管理に使用するツールです。第 2 章を参照してください。

索引

記号

`-`, 80
`+`, 80
`./`, 19, 21, 23
`|`, 80

A

`access_control` ファイル, 174, 177
`args` ファイル, 174

B

`backup` サブディレクトリ, 174

C

`children` ファイル, 174
`CODEMGR_DIR_FLP`, 39
`CODEMGR_PATH`, 51
`CODEMGR_PATH_ONLY`, 52
`.codemgr_resrc` ファイル, 177
`.codemgrtoolrc` ファイル, 176
`CODEMGR_WS`, 52, 118
`Codemgr_wsdata` サブディレクトリ, 173
 `access_control` ファイル, 174, 177
 `args` ファイル, 174

`backup` サブディレクトリ, 174
`children` ファイル, 174
`conflicts` ファイル, 174
`description` ファイル, 175
`Freezepoints` サブディレクトリ, 175
`history` ファイル, 175
`locks` ファイル, 175
`nametable` ファイル, 175
`notification` ファイル, 176
`parent` ファイル, 176
`putback.cmt` ファイル, 112, 176

`codemgr` コマンド, 166

`conflicts` ファイル, 174

Customized メニュー, 107

D

`def.dir.flp` ファイル, 39

`description` ファイル, 175

`dmake`

`-j` オプション, 163

`-m` オプション, 163

 入れ子呼び出し, 161

 基本概念, 151

 コマンド, 136, 161

 ホスト, 162, 163

 メイクファイルに与える影響, 156

 理解, 156

`dmake.conf` ファイル, 142, 152, 155

`.dmakerc` ファイル, 152

F

`FreezePoints` サブディレクトリ, 175

`freezept` コマンド, 169

H

`history` ファイル, 175

L

`locks` ファイル, 175

M

`make` ユーティリティ, 132

MR (変更要求), 112

N

`nametable` ファイル, 175

`.NO_PARALLEL`

ターゲット, 161

`notification` ファイル, 176

P

`parent` ファイル, 176

`putback.cmt` ファイル, 112, 176

R

`rsc2ws`、使用, 53

RCS プロジェクトから TeamWare への変換, 53

Revision Control System (RCS), 53

RTI (Request To Integrate), 59

S

SCCS

`s` ドットファイル, 179

デルタ ID (SID), 99

ファイルの属性, 110

履歴ファイル, 179

ファイルマージ, 173, 185

分岐, 184

「SCCS get 抑制」オプション, 36

`sccs-admin` コマンド, 110

SCCS デルタ ID (SID), 126, 191

SCCS マージ可能 ID (SMID), 126, 191

SCCS デルタ ID (SID) への変換, 192

SCCS デルタ ID (SID)への変換, 192

必要な理由, 191

Shift キーを押しながらクリック, 169

SID

定義, 99

`s` ドットファイル, 179

T

TeamWare

ツール, 9

ワークフロー, 8

TeamWare における

ファイル名の変更, 101

マージデルタ, 178

TeamWare の設定, 7

「TeamWare」メニュー (バージョン管理), 69

「TeamWare」メニュー (フリーズポイント), 117

「TeamWare」メニュー (ワークスペース管理), 15

`twconfig` コマンド, 13

`twfreeze` コマンド, 115

`twmerge` コマンド, 168

`twversion` コマンド, 167

W

- [.WAIT](#) ターゲット, 158
- WorkShop ターゲット, 138
- [workspace descr](#) コマンド, 42

Z

- [.z](#) 拡張子, 125

あ

- アーカイブライブラリ, 159
- 赤色のチェックマーク, 98
- 「アクション」メニュー (ワークスペース管理), 15
- アクセス権
 - 全体レベルでのアクセス権, 56
- アクセス権、ワークスペース値, 178
 - デフォルトのアクセス権, 177
- アクセスの制御、ワークスペース, 55
- 新たにチェックイン, 69
- アンチェックアウト, 72

い

- 依存関係リスト、順序, 158
- 「一般」タブ, 109
- 移動
 - ファイル, 101
 - ワークスペース, 25

え

- エディタ
 - デフォルト, 110
 - 変更, 71
- エラーメッセージ
 - 構築エラーの表示, 146
 - ワークスペース管理, 195

お

- 置き換え
 - 構築のための環境変数, 146
 - メークファイルマクロ, 144
- オプション
 - 相違解決 (ファイルマージ), 86
 - ファイルマージ, 85
 - プリングオーバー / プットバック, 34
 - ワークスペース管理, 48
 - 「オプション」メニュー (ワークスペース管理), 15
- 親ワークスペース
 - 作成, 15
 - 定義, 3

か

- 開始
 - 構築, 136
- 解除
 - ワークスペースのロック, 65
- カスタマイズ
 - 構築, 139
 - ワークスペース管理, 48
 - ワークスペースの履歴表示, 28

環境変数

- [CODEMGR_DIR_FLP](#), 39
- [CODEMGR_PATH](#), 51
- [CODEMGR_PATH_ONLY](#), 52
- [CODEMGR_WS](#), 118, 52
- 構築に対する
 - 値を変更, 145
 - 削除, 145
 - 追加, 145
 - 構築のための, 144
 - 置き換え, 146
- 「環境変数」ダイアログボックス (構築), 144

き

- キーボードショートカット, 170
- 既存のプロジェクトファイル, 17

起動

- バージョン管理, 67, 172
 - コマンド行から, 167
- ファイルマージ, 76
 - コマンド行から, 167
- フリーズポイント, 115
- ワークスペース管理, 13
- 「強制衝突」オプション, 36

く

グリフ

- 意味 (ファイルマージ), 92
- グリフ、ファイルマージの意味, 79

け

- 警告メッセージ、ワークスペース管理, 214
- 現在の相違 (ファイルマージ), 83

こ

更新

- フリーズポイントファイル, 119
- ブリングオーバー / プットバックファイルリスト, 38
- 更新ブリングオーバー
 - タブ, 21
 - トランザクション, 20, 21
 - ファイルリスト, 38
 - ワークスペースを更新, 95

構築

- 開始, 136
- カスタマイズ, 139
- ディレクトリ, 132
- 構築 **make** ターゲット、定義, 132
- 「構築」ウィンドウ
 - 構築出力表示区画, 138, 146, 147
 - 開く, 129
- 構築エラー
 - 修正, 146, 148

ソースの表示, 146, 147

- 構築エラーの修正, 146, 148
- 「構築オプション」ダイアログボックス, 140
- 構築コマンド
 - 指定, 137
- 構築サーバー, 162
- 構築サーバーの構成ファイル, 152
- 構築出力
 - 収集, 138
 - 保存, 139
- 構築出力表示区画, 138, 146, 147
- 構築ターゲット
 - 指定, 137
- 構築ディレクトリ
 - 指定, 137
- 構築に対する環境変数, 145
- 構築の終了, 149
- 構築の出力
 - 表示, 138
- コピー / 変更 / マージ, 2
- コマンド, 165, 169
 - freezept**, 169
 - twconfig**, 13
 - twfreeze**, 115
 - twmerge**, 168
 - twversion**, 167
- コマンド行, 165, 169
 - ファイルマージ, 167
 - フリーズポイント, 169
 - ワークスペース管理, 166
- 「コマンド」メニュー (バージョン管理), 68
- コメント
 - プットバック, 24
- 子ワークスペース
 - 親子関係の変更
 - 親の変更コマンドを使用, 44
 - ドラッグ & ドロップによる, 44
 - 例, 45
 - 親子関係を変更する理由, 42
 - 作成, 17
 - 定義, 3

さ

削除

- 構築に対する環境変数, 145
- ファイル, 101
- メークファイルマクロ, 143
- ワークスペース, 26

作成

- 親ワークスペース, 15
- カスタマイズ (バージョン管理), 106
- 空のワークスペース, 16
- 既存のファイルからワークスペース, 17
- 子ワークスペース, 17
- フリーズポイントファイル, 114, 118

作成ブリングオーバー

- タブ, 18
- トランザクション, 18

差分, 96

し

実行時構成ファイル, 152

実線, 98

指定, 144

- 構築コマンド, 137
- 構築ターゲット, 137
- 構築ディレクトリ, 137
- 構築のための環境変数, 144

自動作成

- フリーズポイントファイル, 124

自動的に作成

- ブリングオーバー / ブットバックファイルリスト, 38

自動マージ, 84

- 「詳細メッセージ」オプション, 36

衝突

- 定義, 76
- ファイルマージ, 78
- 「衝突解決」トランザクション, 172

衝突解決トランザクション, 76, 180

- ファイルマージ SCCS 履歴ファイル, 185

衝突するファイルをマージ, 78

ショートカット

- キーボード, 170
- マウス, 170

「ジョブグラフ」ウィンドウ, 136

白抜きのグリフ, 81

シンボル

- ファイルマージのグリフ, 79
- ファイル履歴, 98

せ

正符号 (+), 80

そ

相違 (ファイルマージ)

- 解決, 83
- 現在の, 83
- 次の, 83
- 前の, 83

相違を解決, 83

ソース階層

- 再作成, 120
- 抽出, 120

ソースコード管理システム (SCCS), 4

ソースワークスペース, 115

- ソース, 115

祖先ファイル (ファイルマージ), 82

- 起動時に読み込み, 168

た

ターゲット

- Sun WorkShop, 132
- 複雑なプロジェクト, 132
- 複数, 160
- 並列構築, 157, 158

- ユーザーメークファイル, 133

「ターゲットの新規定義」ダイアログボックス, 133, 136

「ターゲットの編集」ダイアログボックス, 133

ダイアログボックス
「メークのマクロ」, 142

縦棒 (|), 80

妥当性検査, 59

ダブルクリック, 50, 109, 171

ち

抽出

ソース階層, 115, 120

フリーズポイントファイル, 120

フリーズポイントファイルからのデルタ, 115,
120

つ

追加

メークファイルマクロ, 143

「読み込み」メニューへの (バージョン管
理), 107

ワークスペースへのファイルの, 69

ツール, 9

「ツール属性」ダイアログボックス, 48

「衝突解決」タブ, 86

「ワークスペース管理」タブ, 48

次の相違 (ファイルマージ), 83

て

ディレクトリ、現在の作業用 (ファイルマ
ージ), 82

デフォルト値を使用した構築, 135

デフォルトのエディタ, 71, 110

デフォルトのファイルリスト, 38

デルタ

定義, 4, 99

フリーズポイントファイルに含まれる, 126

「デルタコメント」オプション, 36

デルタのマージ, 100

電子メールによる通知, 40

と

統合要求 ID, 62

ドラッグ & ドロップ, 170

トランザクション

更新ブリングオーバー, 20, 21

作成ブリングオーバー, 18

衝突解決, 172, 180

制御, 55

取り消し (ファイルマージ), 85

取り消し (ワークスペース管理), 25

プットバック, 23

他のユーザーへの通知, 40

取り消しトランザクション, 25

な

名前

履歴, 103

ワークスペース, 41

名前の変更

ワークスペース, 25

名前変更の衝突, 103, 104

に

二重下線, 98

ね

ネットグループ, 58

は

バージョン管理

カスタマイズメニュー, 106

コマンド行から起動, 167

ダブルクリックで自動的に起動, 172

バージョンの定義, 100

波線, 98

「バックアップしない」オプション, 36

Xで消されたデルタ, 98

ひ

日々の作業, 8

表示

SCCS 管理下でないファイル, 69

構築エラーのソース, 146

差分, 96

デルタ間の相違, 108

ファイルの履歴, 96

ワークスペース履歴, 27

「表示」メニュー (バージョン管理), 68

「表示」メニュー (フリーズポイント), 117

「表示」メニュー (ワークスペース管理), 15

ふ

ファイル

.z 拡張子, 125

access_control, 174, 177

args, 174

children, 174

.codemgr_resrc, 177

.codemgrtoolrc, 176

conflicts, 174

description, 175

dmake.conf, 142, 152, 155

.dmakerc, 152

history, 175

locks, 175

nametable, 175

notification, 176

parent, 176

putback.cmt, 112, 176

SCCS 属性, 110

アンチェックアウト, 72

移動, 101

構築サーバーの構成, 152

再読み込み (ファイルマージ), 85

削除, 101, 105

実行時構成, 142, 152

祖先, 82

独自のプリングオーバー / プットバックリスト, 38

名前の履歴, 103

バージョン, 100

ファイルマージ

衝突してしない, 179

衝突している, 180

ファイル名の変更, 101

フリーズポイント, 114

更新, 119

作成, 114, 118

自動作成, 124

抽出, 120

定義, 126

読み込み

起動時に2つ (ファイルマージ), 168

起動時に3つ (ファイルマージ), 168

読み込み (バージョン管理), 107

履歴, 96

ファイルのチェックアウト, 70

ファイルの並行修正, 159

ファイルの履歴

表示, 96

ファイル変更の統合, 73

ファイルマージ

SCCS 履歴ファイル, 180

オプション, 85

コマンド行, 167

衝突していないファイル, 179

衝突している, 180

変更の取り消し, 85

例, 88, 93

「ファイルマージ」ウィンドウの色, 79

ファイル名の変更

ファイル, 101

「ファイル」メニュー (バージョン管理), 68

「ファイル」メニュー (フリーズポイント), 117

「ファイル」メニュー (ワークスペース管理), 15

ファイルリストプログラム (FLP)

例, 39

独自で作成する, 39

ファイル履歴

シンボル, 98

- 「ファイルを開く」ダイアログボックス (ファイルマージ), 81
- ファイルをまとめて選択, 169
- 複数のターゲット, 160
- プットバック
 - オプション, 34
 - コメント, 24
 - タブ, 23
 - トランザクション, 23
 - ファイルリスト, 38
- プットバックの妥当性検査, 59
 - 起動, 61
 - プログラム例, 62
- 負符号 (-), 80
- フリーズポイント
 - 起動, 115
 - コマンド行, 169
 - 作成, 114
 - 抽出, 115
- 「フリーズポイント」ウィンドウ
 - 作成モード, 116
 - 抽出モード, 122
- フリーズポイント、定義, 113
- フリーズポイントファイル, 114
 - 更新, 119
 - 作成, 114, 118
 - 自動フリーズポイント, 124
 - 情報, 126
 - 抽出, 120
 - 定義, 126
- プリングオーバー / プットバックのオプション
 - SCCS get 抑制, 36
 - 強制衝突, 36
 - 詳細メッセージ, 36, 37
 - 設定, 34
 - デルタコメント, 36, 37
 - バックアップしない, 36
 - プレビュー, 36, 37
 - メッセージ抑制, 36, 37
- 「プレビュー」オプション, 36
- プロジェクト、RCS から TeamWare への変換, 53

- 分岐, 99, 184
- 分散メーク, 142

へ

- 並列処理, 160
- 並列処理に対する制限, 160
 - 「ヘルプ」メニュー (バージョン管理), 69
 - 「ヘルプ」メニュー (フリーズポイント), 117
 - 「ヘルプ」メニュー (ワークスペース管理), 15
- 変更
 - 構築に対する環境変数の値, 145
 - デフォルトのエディタ, 71
 - メークファイルマクロの値, 143
 - ワークスペース, 25
- 変更の伝達, 20
- 変更の取り消し
 - ファイル, 72
 - ファイルマージ, 85
- 変更要求 (MR), 112

ま

- マージ
 - 自動, 84
- マージ版 (ファイルマージ), 93
- マージ版を編集 (ファイルマージ), 93
- マウス
 - ショートカット, 170
 - ダブルクリック, 50
- マウスの右ボタン, 170, 171
- 前の相違 (ファイルマージ), 83
- マクロ
 - 動的, 159
 - メークファイル
 - 追加, 143
 - 変更, 143
- マニュアルページ, 166

め

「メイクのマクロ」ダイアログボックス, 142

メイクファイル

`dmake` ユーティリティが与える影響, 156

制限事項, 157

定義, 132

テンプレート、使用, 156

ファイルの衝突, 160

メイクファイルマクロ

置き換え, 144

削除, 143

追加, 143

定義, 142

変更, 143

「メッセージ抑制」オプション, 36

メニューのカスタマイズ (バージョン管理), 106

や

矢印, 98

ゆ

ユーザーアクセス権

グループアクセス権, 58

ユーザーメイクファイル

ターゲット, 133

プロジェクト, 135, 137

よ

読み込み

「ファイルマージ」ウィンドウからのファイルの, 81

ファイルマージの起動時に2つのファイルを, 168

ファイルマージの起動時に3つのファイルを, 168

「読み込み」メニュー (バージョン管理), 107

ら

ライブラリの更新、同時, 159

り

履歴

ファイル, 179

ワークスペース, 27

れ

例

ファイルマージ, 88, 93

ワークスペースの親子関係の変更, 45

ろ

ロック, 65

わ

ワークスペース

アクセス権, 55

値, 178

デフォルトのアクセス権, 177

移動, 25

イベント通知, 40

親

作成, 15

定義, 3

親子関係の変更, 42

理由, 42

例, 45

空の, 16

既存のファイルから, 17

子

作成, 17

定義, 3

削除, 26

衝突, 76

設定, 55

- 設定の解除, 27
- 定義, 13
- 名前, 41
- 名前の変更, 25
- ファイルの追加, 69
- プットバックの妥当性検査, 59
- 変更, 25
- メタデータディレクトリ
 - (Codemgr_wsdata), 173
- 履歴の検索, 31
- 履歴の表示, 27
- ロック, 65
- ロックの解除, 65
- ワークスペース管理、定義, 173
- 分かりやすい名前, 41
- ワークスペース管理
 - エラーメッセージ, 195
 - カスタマイズ, 48
 - 警告メッセージ, 214
 - コマンド行, 166
 - 属性, 48
- 「ワークスペース管理」ウィンドウ, 14
- ワークスペース管理、定義, 173
- ワークスペース設定の解除, 27
- ワークスペースの親子関係の変更, 42
 - 一時的な, 44
 - 例, 45
- ワークスペースの親子関係を変更する
 - 理由, 42
- ワークスペースの履歴表示
 - 起動, 28
 - フィルタ情報, 30
- ワークスペースの履歴表示情報のフィルタ, 30
- ワークスペースへのアクセスの制御, 55
- ワークスペース名を表示, 41
- 「ワークスペース」メニュー (バージョン管理), 69
- 「ワークスペース」メニュー (ワークスペース管理), 15
- ワークスペース履歴の検索, 31
- ワークフロー, 8