



Forte 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. 816-2853-01
2001 年 10 月 Revision A

Copyright © 2001 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303, U.S.A. All rights reserved.

Sun Microsystems, Inc. は、この製品に組み込まれている技術に関連する知的所有権を持っています。具体的には、これらの知的所有権には <http://www.sun.com/patents> に示されている 1 つまたは複数の米国の特許、および米国および他の各国における 1 つまたは複数のその他の特許または特許申請が含まれますが、これらに限定されません。

本製品またはドキュメントは著作権法で保護され、ライセンス規定に従って配布され、本製品またはドキュメントの使用、コピー、配布、逆コンパイルには制限があります。本製品または関連ドキュメントのいかなる部分も、その形態および方法を問わず、Sun およびそのライセンサーの事前の書面による許可なく複製することを禁じます。

フォントテクノロジーを含むサードパーティ製のソフトウェアの著作権およびライセンスは、Sun のサプライヤが保有しています。PointBase ソフトウェアは社内開発用にのみ利用できます。商用の利用には、PointBase から別途ライセンスが必要です。

Sun、Sun Microsystems、Sun のロゴ、Java、Forte、NetBeans、Solaris、iPlanet、StarOffice、StarPortal、Jini、および Jiro は、米国および他の各国における Sun Microsystems, Inc. の商標または登録商標です。

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。

連邦政府による取得: 市販ソフトウェア -- 米国政府機関による使用は、標準のライセンス条項に従うものとします。

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行われぬものとします。

原典 : *Forte TeamWare User's Guide*
 Part No: 816-1404-10
 Revision A

© 2001 by Sun Microsystems, Inc.



目次

はじめに	xv
1. Forte TeamWare の概要	1
Forte TeamWare の機能と特長	1
Forte 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
Forte TeamWare のモデル	4
コピー / 変更 / マージモデル	4
チームプロジェクトモデル	6
Forte TeamWare での作業	9
Forte TeamWare の使用例	10
既存のチームへの新規参加	10
Forte TeamWare 環境の設定	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	16

空のワークスペースの作成	17
既存のファイルからのワークスペースの作成	17
子ワークスペースの作成	19
ワークスペース間の変更内容の伝達	22
子ワークスペースの更新 (更新ブリングオーバー)	22
親ワークスペースへの変更内容のプットバック	25
ワークスペースに対する変更の取り消し	27
ワークスペースの名前の変更と移動	28
ワークスペースの削除と設定の解除	29
ワークスペースの削除	29
ワークスペース設定の解除	29
ワークスペースの履歴表示	30
ワークスペースの履歴表示のカスタマイズ	31
トランザクション情報の追加と削除	33
トランザクション情報のソート	33
フィルタオプションの追加	33
表示するトランザクション情報の選別	34
トランザクションの検索	34
3. 高度なワークスペース管理	35
ブリングオーバーとプットバックのオプション	35
ブリングオーバー/プットバックトランザクションごとのオプションの設定	36
ツール属性の設定	38
独自のブリングオーバーおよびプットバック用ファイルリストの作成	39
デフォルトのファイルリストの保存	39
独自のファイルリストの作成	39
他のユーザーへのトランザクションの実行通知	41
ワークスペースに分かりやすい名前を付ける	42

ワークスペースの親の変更	43
ワークスペースの親を変更する理由	44
ワークスペースの親を変更する方法	45
親の変更例	46
ツール属性によるワークスペース管理のカスタマイズ	49
環境変数の設定	52
ワークスペースの自動読み込み	52
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	53
検索パスの設定	53
RCS プロジェクトの変換	54
4. ワークスペースへのアクセスの制御	57
アクセス権の設定	57
全体レベルのアクセス権の設定	58
グループまたはユーザーレベルのアクセス権の設定	60
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	61
プットバックの妥当性検査を有効にする	62
プットバックの妥当性検査プログラムの起動	63
ワークスペースのロックの解除	67
5. ファイルの管理	69
バージョン管理ツールの起動	69
ワークスペースへのファイルの追加	72
ファイルのチェックアウト	73
ファイルの編集	75
デフォルトのエディタの変更	76
ファイルのチェックイン	76
ファイルに加えた変更の取り消し	78

Forte TeamWare の内部からのファイルの削除	112
カスタマイズメニューの作成	112
「読み込み」メニューへのパスの追加	114
バージョン管理の属性の設定	115
SCCS ファイルの属性の設定	117
8. フリーズポイントの使用	119
フリーズポイントツールの概要	119
フリーズポイントの仕組み	120
フリーズポイントにおける作成	120
フリーズポイントにおける抽出	121
ソースワークスペース	121
抽出先ディレクトリ	121
フリーズポイントの起動	122
フリーズポイントファイルの作成	125
フリーズポイントファイルの更新	126
ファイルの抽出	127
フリーズポイントの自動作成	131
フリーズポイントファイルの形式	133
9. dmake ユーティリティの使用	135
dmake の基礎	135
構成ファイル	136
dmake ホスト	136
構築サーバー	139
dmake ユーティリティについて	140
dmake ユーティリティがメイクファイルに与える影響	140
メイクファイルテンプレートの使用	140

ターゲットの並列構築	141
メイクファイルに関する制限事項	141
ライブラリの同時更新	143
複数のターゲット	144
並列処理に対する制限	144
分散メイクの入れ子呼び出し	145
dmake ユーティリティの使用法	146

10. Forte TeamWare のショートカット 149

コマンド行からの TeamWare の使用法	149
ワークスペース管理関係のコマンド	150
バージョン管理関係のコマンド	151
ファイルマージ関係のコマンド	151
フリーズポイント関連のコマンド	153
GUI のショートカット	153
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	155
バージョン管理ツールでのダブルクリック時の動作	156
ファイル履歴でのダブルクリック時の動作	156

11. Forte for Java IDE での Forte TeamWare の使用 157

Forte TeamWare と Forte for Java の関係	157
Forte for Java における Forte TeamWare の機能	158
Forte TeamWare ファイルシステムのマウント	158
Forte for Java の TeamWare ツールバーの表示方法	159
Forte for Java での Forte TeamWare の実行	160
Forte TeamWare コマンドの使用法と機能	161
Forte TeamWare ツールと機能	161
GUI と Forte TeamWare コマンド	167

12. Forte TeamWare のアーキテクチャー 175

ワークスペースのメタデータディレクトリ 175

ワークスペース管理のデフォルト値ファイル 178

access_control ファイル 179

ワークスペース管理によるファイルのマージ 180

衝突していないファイルのマージ 181

衝突しているファイルのマージ 182

ファイルマージによるデルタの追跡 183

SCCS マージ可能 ID について 193

SMID が必要な理由 193

SMID と SID 間の変換 193

A. エラーメッセージと警告メッセージ 197

エラーメッセージ 197

警告メッセージ 219

B. 問題の対処方法 227

ブリングオーバーとプットバック 228

環境変数 229

Java 230

Microsoft Windows と UNIX オペレーティングシステム間の互換性 231

Microsoft Windows のファイル共有 232

プロセス監視 233

使用すべきでない SCCS コマンド 234

SCCS 履歴ファイル 235

SCCS のエラー 236

テキスト形式 237

ワークスペース 239

バージョンの確認方法 240

C. Microsoft Windows 環境における Forte TeamWare の使用 241

Forte TeamWare ツールの起動 242

環境変数の設定 243

コマンド行インタフェースの使用法 244

ドライブ名の選択とパス名の構文 244

改行コードの CR と LF の変換について 245

テキストファイルへの改行文字の追加 246

クロスプラットフォーム対応 246

 SCCS 履歴ファイルの共有 246

 ワークスペースの共有 247

 論理パスとワークスペースメタファイルの問題 247

Solaris ユーザーとのファイルの共有 247

Samba ファイルサーバーの構成 248

Microsoft Visual Studio のプラグインとしての
Forte TeamWare の使用 249

 Microsoft Visual Studio プロジェクトを Forte TeamWare の SCC 管理下に置
 く 249

 Microsoft Visual Studio からの Forte TeamWare コマンドの使用法 249

用語集 253

索引 261

図目次

図 1-1	一般的な Forte TeamWare 環境の設定	7
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	15
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	20
図 2-3	「ファイル追加」ダイアログボックス	21
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	23
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	25
図 2-6	「履歴表示」ウィンドウ	32
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	37
図 3-2	親子関係のない 2 つのワークスペース	47
図 3-3	作成したワークスペースのクローン	47
図 3-4	クローンの親を Release2.0 に変更	48
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	48
図 3-6	修正内容が取り込まれた Release2.0 (patch1.0_clone は削除)	49
図 3-7	「ツール属性」の「ワークスペース管理」タブ	50
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	59
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	63
図 5-1	「バージョン管理」ウィンドウ	71
図 6-1	ワークスペース内の衝突例	83
図 6-2	「ファイルマージ」ウィンドウ	85
図 6-3	「ファイルマージ」ウィンドウの「ファイルを開く」ダイアログボックス	87
図 6-4	「ツール属性」ダイアログボックスの「衝突解決」タブ	93

- 図 6-5 自動マージ後の file_1 と file_2 のマージ結果 97
- 図 6-6 自動マージ後の子区画 (file_1) 98
- 図 6-7 自動マージ後の親区画 (file_2) 98
- 図 7-1 「ファイル履歴」ウィンドウの各部 103
- 図 7-2 ファイル「C」の名前を「D」に変更 108
- 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 110
- 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 111
- 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 115
- 図 8-1 フリーズポイント用デルタの保存 120
- 図 8-2 「フリーズポイント」ウィンドウの「作成」タブ 123
- 図 8-3 フリーズポイント処理の進行状況 128
- 図 8-4 「フリーズポイント」ウィンドウの「抽出」タブ 129
- 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 132
- 図 11-1 Forte TeamWare 統合モジュールが組み込まれた Forte for Java 160
- 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 182

表目次

表 1-1	Forte 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	16
表 2-2	ワークスペースの履歴表示の項目	31
表 3-1	ブリングオーバーとプットバックのオプション	37
表 3-2	ブリングオーバー/プットバックのツール属性	38
表 3-3	workspace descr コマンドのオプション	43
表 3-4	ワークスペース管理ツールの属性	51
表 4-1	プットバックの妥当性検査のモード	62
表 5-1	「バージョン管理」ウィンドウのメニュー	71
表 6-1	ファイルマージの「ファイルを開く」ダイアログボックスのフィールド	88
表 6-2	「トランザクションのチェックイン」ダイアログボックス	89
表 6-3	自動マージ規則のまとめ	91
表 6-4	「ツール属性」ダイアログボックスの「衝突解決」タブ	93
表 6-5	「ファイルマージ」ウィンドウの表示オプション	94
表 7-1	「ファイル履歴」ウィンドウの各部	104
表 7-2	ファイル履歴表示で使用されるシンボル	104
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の「一般」タブ	116
表 7-4	SCCS ファイルの属性	117
表 8-1	フリーズポイントの「作成」タブの項目	123
表 8-2	フリーズポイントの「抽出」タブの項目	130
表 10-1	「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	151

表 10-2	マウスおよびキーボードショートカット	154
表 11-1	Forte TeamWare ツールと機能、アクセス方法	161
表 11-2	Forte for Java の GUI と割り当てられている Forte TeamWare コマンド	167
表 11-3	選択したファイルのマウント状態によって使用されるファイルエディタ	173
表 12-1	Codemgr_wsdata メタデータディレクトリの内容	176
表 12-2	デフォルトのアクセス権設定	179
表 12-3	ワークスペースに対するアクセス制御値	180

はじめに

このマニュアルでは、Forte™ for Java™, release 3.0, TeamWare モジュール (以降、Forte TeamWare または TeamWare と表記) コード管理ツールの使用方法について説明します。このツールは、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が利用することもできます。

このマニュアルは、Solaris™ オペレーティング環境および UNIX コマンドに関する知識があ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。

マルチプラットフォーム対応

Forte TeamWare の本リリースは、次のプラットフォームおよびオペレーティングシステムに対応しています。

- Solaris 8 (SPARC プラットフォーム版)
- Solaris 7 (SPARC プラットフォーム版)
- Microsoft Windows 2000
- Microsoft Windows NT 4.0
- Red Hat Linux 6.2

Microsoft Windows で使用する場合の注意事項

Microsoft Windows 環境で Forte TeamWare を使用する際の注意事項については、付録 C 「Microsoft Windows 環境における Forte TeamWare の使用」を参照してください。Microsoft Windows プラットフォームに Forte TeamWare をインストールすると、自動的に「スタート」メニューの「TeamWare」サブディレクトリにショートカットが作成されます。ただし、このショートカットを使用して「スタート」メニューから Forte TeamWare を起動した場合、Forte TeamWare はスタンドアロンアプリケーションとして動作します。したがって、Forte for Java 環境には統合されません。詳細は、13 ページの「ワークスペース管理ツールの起動」を参照してください。

統合モジュールとしての Forte TeamWare の使用

第 11 章「Forte for Java IDE での Forte TeamWare の使用」では、Forte for Java 統合開発環境 (IDE) 内の 1 つのモジュールとして Forte TeamWare を使用方法について説明しています。Forte for Java の GUI から、Forte TeamWare のコマンドを直接使用する方法についてもまとめてあります。

マニュアルページのアクセス方法

Forte TeamWare では、man コマンドを使用してマニュアルページ (一般的に UNIX システムで提供されるテキストベースのオンラインマニュアル) にアクセスすることはできません。マニュアルページは、「ヘルプ」メニューからアクセス可能な Forte TeamWare のオンラインヘルプシステムでアクセスできます。

マニュアルページにアクセスするには、Forte TeamWare のオンラインヘルプシステムの目次から「コマンドリファレンス」を選択してください。

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「Forte TeamWare の概要」では、Forte 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の制御」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プ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「フリーズポイントの使用」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 10 章「Forte TeamWare のショートカット」では、コマンド行から Forte 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 11 章「Forte for Java IDE での Forte TeamWare の使用」では、Forte for Java IDE への統合モジュールとしての Forte TeamWare を取り上げ、その環境内から Forte TeamWare コマンドを使用する方法を説明します。

第 12 章「Forte 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。

付録 B 「問題の対処方法」では、Forte TeamWare の一般的な問題と、問題の対処方法を示しています。

付録 C 「Microsoft Windows 環境における Forte TeamWare の使用」では、Microsoft Windows 上で Forte TeamWare を使用するときの注意事項を説明します。

「用語集」は、このマニュアルで使用されている用語の説明です。

表記上の規則

このマニュアルでは、次のような字体や記号を特別な意味を持つものとして使用します。

表 P-1 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 system%
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	system% su Password:
<i>AaBbCc123</i>	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには rm <i>filename</i> と入力します。

表 P-1 表記上の規則 (続き)

字体または記号	意味	例
『 』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。
[]	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第 5 章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	sun% grep `^#define \ XV_VERSION_STRING'

コード例は次のように表示されます。

■ C シェルプロンプト

```
system% command y|n [filename]
```

■ Bourne シェルおよび Korn シェルのプロンプト

```
system$ command y|n [filename]
```

■ スーパーユーザーのプロンプト

```
system# command y|n [filename]
```

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

| は区切り文字 (セパレータ) です。この文字で分割されている引数のうち 1 つだけを指定します。

キーボードのキー名は英文で、頭文字を大文字で示します (例: Shift キーを押します)。ただし、キーボードによっては Enter キーが Return キーの動作をします。

ダッシュ (-) は 2 つのキーを同時に押すことを示します。たとえば Ctrl-D は Control キーを押したまま D キーを押すことを意味します。

関連マニュアル

以下の方法で、Forte TeamWare の関連マニュアルにアクセスできます。

- 『Forte TeamWare リリースノート』から
『Forte TeamWare リリースノート』は HTML 形式のファイルです。このマニュアルを表示するには、Netscape™ Communicator 4.0 またはこれと互換性のあるブラウザで以下のファイルを開きます。

`Eff-user-dir/docs/com/sun/teamware/relnotes/ja/relnotes.html`

`Eff-user_dir` は、Forte TeamWare をインストールしたディレクトリです。

- Forte TeamWare のオンラインヘルプから
Forte TeamWare の任意のウィンドウで「ヘルプ」▶「目次」を選択します。
- Forte for Java のオンラインヘルプから
Forte for Java のメインウィンドウで「ヘルプ」▶「内容」を選択します。
- インターネットの Web サイトから
`http://docs.sun.com` を参照してください。
本のタイトル、主題、マニュアルコレクション、製品別に検索できます。

第1章

Forte TeamWare の概要

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

- 1 ページの「Forte TeamWare の機能と特長」
- 2 ページの「Forte TeamWare の基本概念」
- 4 ページの「Forte TeamWare のモデル」
- 9 ページの「Forte TeamWare での作業」
- 10 ページの「Forte TeamWare の使用例」

Forte TeamWare の機能と使用方法についての詳細は、この後の章で説明します。

Forte TeamWare の機能と特長

Forte 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ツールです。ソースコード管理を簡素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。具体的には、Forte TeamWare では次のことが行えます。

- 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。
- 同一のファイルの編集を禁止することによる、同一のワークステーション内の衝突防止。
- 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の作成。特定のバージョンをアーカイブ保管して、後で使用できます。
- 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。
- チームのメンバーへの、ファイルに加えられた個々の処理の自動通知

Forte 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) というプログラムを土台にしています。Forte 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、Forte TeamWare に移行することによって管理できます。

Forte TeamWare の基本概念

Forte TeamWare 開発モデルは、「コピー / 変更 / マージ」の概念に基づいています。また同じく基本的な概念として、「親」と「子」ワークスペースの概念と両者の関係もあります。Forte 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機能します。このため、SCCS も理解する必要があります。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。この問題に対する 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。

Forte 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て、そのワークスペース内でファイルを変更し、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 (フォルダともいう) とファイル単位で作業を行います。Forte TeamWare を活用するには、1つのディレクトリの下にすべての作業用ディレクトリをまとめ、Forte TeamWare 使用して、そのディレクトリ階層全体を1つのワークスペースにします。Forte 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**: TeamWare ユーザー各自が所有するワークスペース間に「インテリジェント」な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**: ファイルの履歴と各ファイルに対するデルタ (変更) を管理します。
- **ファイルマージ**: 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**: ワークスペースの「スナップショット」を作成します。

Forte TeamWare は、上位ディレクトリを1つのインテリジェントな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を作成します。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て作業します。Forte 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。チームの各メンバーは1つ以上の子ワークスペースを持ち、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むことができます。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)

Forte 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を「デルタ」と呼びます。Forte 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、コピーされた場合、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。Forte TeamWare による SCCS 履歴ファイルの処理とマージについては、180 ページの「ワークスペース管理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。

Forte TeamWare のモデル

ソフトウェア開発プロジェクトにおける Forte TeamWare の利用形態は次の 2 つのモデルによって表されます。

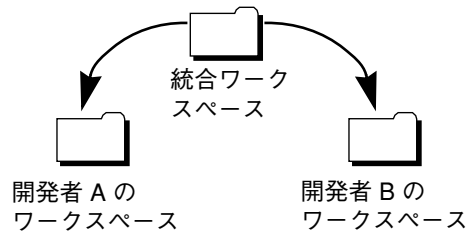
- コピー / 変更 / マージモデル : Forte TeamWare の基本モデル
- チームプロジェクトモデル : 環境の設定時や日々の作業における、Forte TeamWare ユーザーモデル

コピー / 変更 / マージモデル

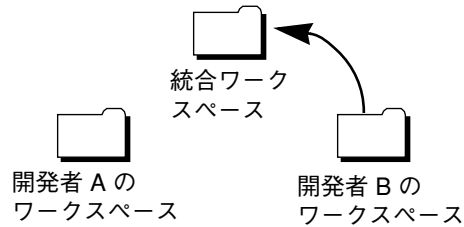
Forte TeamWare のワークスペース管理ツールは、「コピー / 変更 / マージ」という並行開発モデルに基づいています。開発プロジェクトに Forte TeamWare ツールを利用した場合、日々の作業はプロジェクトファイルに対する次のような基本作業の繰り返しになります。

1. **ブリングオーバー** : 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。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 組の変更を選別しながらマージできます。

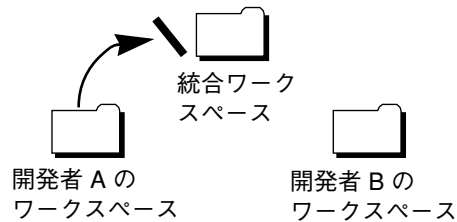
次のコピー / 変更 / マージの例は、2 人のユーザーがあるプロジェクトの同一または関連する部分を同時に操作するという、一般的なソフトウェア開発例を示しています。



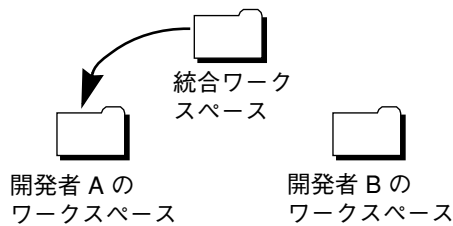
開発者 A と B が、プロジェクト統合ワークスペースから自分のワークスペースに同じファイルをコピーします。



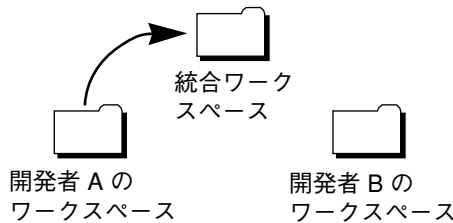
開発者 B がファイルを変更し、変更したファイルを統合ワークスペースに戻します。



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します。しかし、開発者 B の変更が上書きされることになるため、Forte TeamWare のワークスペース管理ツールによって開発者 A の行為が阻止されます。



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

個別ユーザーレベルのコピー / 変更 / マージモデルは、チームレベルの大規模な Forte TeamWare 開発モデルにも当てはまります。一般的な大規模プロジェクトでは、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を組み立て、作業工程を計画する必要があります。以下は、そうした構成と作業工程の 1 例です。最初に Forte TeamWare 環境の設定、続いて日々の作業へと進みます。

Forte 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う必要があります。

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、次のような段階または部門別にプロジェクト内容を収納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を置くワークスペース
 - 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
 - チームワークスペース : エンジニアリングチームごとのワークスペース
 - ベータ / 最終 : プロジェクトの段階、リリース、プラットフォーム、ロケールなどに基づくビルド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、所属するチームのワークスペース (部門 1、部門 2、部門 3) を基準に自分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、親ワークスペースから自分のワークスペースに必要なディレクトリやファイルのコピーをブリングオーバーします。

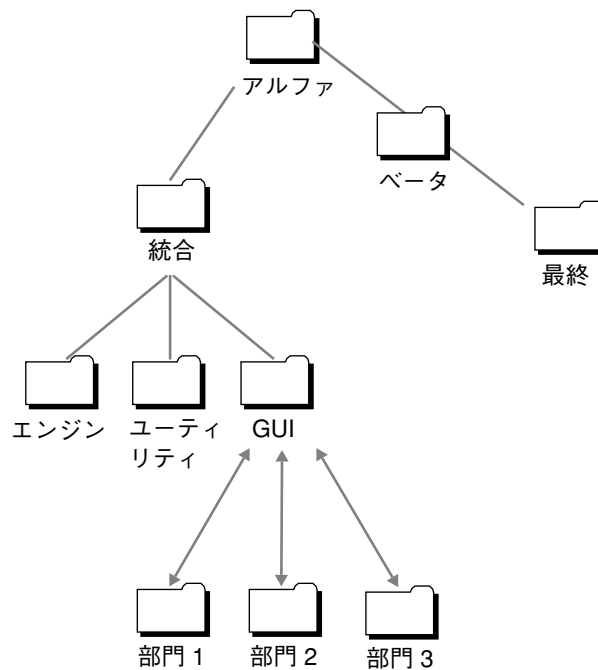


図 1-1 一般的な Forte TeamWare 環境の設定

これで、並行開発のためのチームの環境が整ったこととなります。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。

Forte TeamWare 環境における日々の作業

開発チームの各メンバーはファイルやディレクトリに対して作業を行います。アプリケーションの開発工程は、次のような作業の定期的な繰り返しになります。

1. チームメンバーが、チームの親ワークスペースから自分の子ワークスペースにディレクトリをブリングオーバーします。
 - 各メンバーは、ブ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。
 - メンバーが自分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。
 - Forte TeamWare は、通知リストに登録されているメンバーに変更がプットバックされたことを自動的に通知します。
2. チームのリーダーはプ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」ワークスペースにプットバックします。
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」ワークスペースにプットバックします。
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、アーカイブ保管します。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時点の全内容が含まれています。

Forte TeamWare での作業

次の表は、各 Forte TeamWare ツールの用途を簡単にまとめています。

表 1-1 Forte TeamWare ツールの用途

作業	Forte 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
親ワークスペースからの子ワークスペースの作成	ワークスペース管理
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
自分のワークスペース内のファイルのチェックアウト	バージョン管理
ファイルの変更と保存	バージョン管理
自分のワークスペースへのファイルのチェックイン	バージョン管理
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
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
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 のマージ	ファイルマージ
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
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

Forte TeamWare の使用例

Forte 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。

既存のチームへの新規参加

ここでは、Forte TeamWare ワークスペース管理ツールを使ってソフトウェアの並行開発を行なっていて、すでにワークスペースの構成を終えているチームに新規参加するときの作業手順例を紹介します。

既存のチームに新規参加するときの手順例

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作業工程、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ことになります。
4. Forte TeamWare 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。

変更内容は、チームに自動的に通知されます。

9. 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。

Forte TeamWare 環境の設定

ここでは、プロジェクト用の Forte TeamWare ワークスペースを初めて作成するときの作業手順例を紹介します。

Forte TeamWare 環境を設定する作業手順例

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プラットフォームとターゲットプラットフォーム
 - 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. チーム内で次の事項に関する取り決めをします。
 - 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リリースポイント作成
5. アプリケーションの開発も行っている場合は、10 ページの「既存のチームへの新規参加」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 Forte 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

Forte TeamWare の機能を利用するには、ファイルとディレクトリを Forte 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、およびそれらディレクトリに含まれるファイル全体を指します。Forte 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は、Forte TeamWare の入り口となるものです。Forte TeamWare は、起動方法に応じて、スタンドアロンのアプリケーションまたは Forte for Java IDE に統合された 1 機能として動作します。

Forte 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)。

▼ Forte for Java IDE から Forte TeamWare を起動する

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

- Forte for Java の TeamWare ツールバーにある「TeamWare ワークスペース管理」ボタン  をクリックする。

注 – デフォルトでは、TeamWare の簡略ツールバーが表示されます。TeamWare ツールバー全体を表示する方法については、159 ページの「Forte for Java の TeamWare ツールバーの表示方法」を参照してください。

- Forte for Java のメインウィンドウから、次の操作をする。
 - i. Forte for Java の「エクスプローラ」ウィンドウで Forte TeamWare のディレクトリを選択し、メインウィンドウの「バージョン管理」メニューをクリックします。
 - ii. 「TeamWare」▶「ワークスペース管理」を選択します。
- Forte for Java の「エクスプローラ」ウィンドウから、次の操作をする。
 - i. Forte TeamWare のディレクトリをマウスの右ボタンでクリックします。

注 – Forte for Java の「エクスプローラ」ウィンドウに Forte TeamWare のディレクトリを表示するには、まずそのディレクトリをマウントする必要があります。詳細は、158 ページの「Forte TeamWare ファイルシステムのマウント」を参照してください。

- ii. 「TeamWare」▶「ワークスペース管理」を選択します。

Forte TeamWare ツールの起動と Forte for Java IDE 内からのコマンドの使用方法についての詳細は、第 11 章を参照してください。

▼ スタンドアロンアプリケーションとして Forte TeamWare を起動する

Solaris または Linux 上でワークスペース管理ツールを起動する場合は、コマンド行で次のコマンドを入力します。

```
% twconfig &
```

Microsoft Windows 上でワークスペース管理ツールを起動する場合は、「スタート」メニューから次の順にメニュー項目を選択します。

「プログラム」 ▶ 「Forte for Java」 ▶ 「TeamWare」 ▶ 「CodeManager Tool (Configuring)」

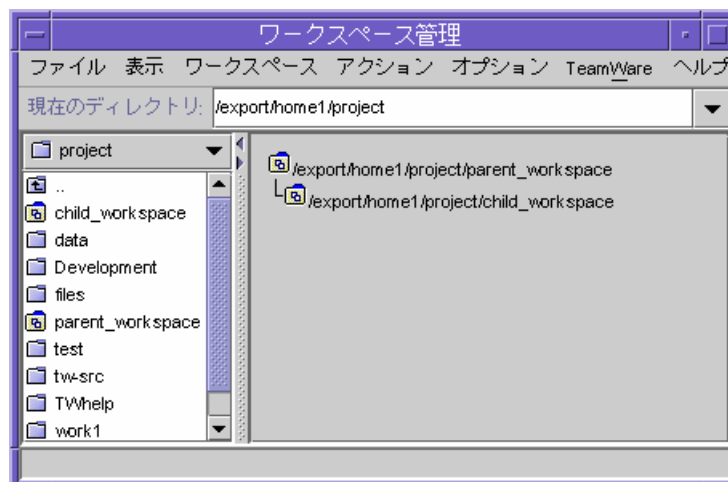


図 2-1 親ワークスペースが読み込まれた状態の「ワークスペース管理」ウィンドウ

「ワークスペース管理」ウィンドウには、表 2-1 に示すメニューがあります。

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	ワークスペースを読み込み、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	ワークスペースおよびワークスペース管理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、ファイルマージ、フリーズポイントなどの他の Forte TeamWare ツールを起動するためのコマンドがあります。
ヘルプ	オンラインヘルプの起動、マニュアル一覧の表示、Forte 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。

これらのメニューとその機能についての詳細は、Forte 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のカスタマイズ方法については、49 ページの「ツール属性によるワークスペース管理のカスタマイズ」を参照してください。

親ワークスペースの作成

Forte 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 Forte TeamWare に読み込むこともできます。どちらにしても Forte TeamWare を使用するには、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、「ブリングオーバー」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。開発者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方法は次の 2 つあります。

- 空のワークスペースを作成し、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にワークスペース管理ツールを使用するには、それらファイルの格納場所の情報 (パス名) が必要になります。

空のワークスペースの作成

空の新規ワークスペースを作成する場合は、次のようにします。

1. ワークスペース管理を起動します。
詳細は、13 ページの「ワークスペース管理ツールの起動」を参照してください。
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。
指定されたディレクトリが存在しない場合は、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。72 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

ワークスペースに入れるファイルを含むディレクトリがすでにある場合は、次の手順で新規ワークスペースを作成します。

1. ワークスペース管理を起動します。
詳細は、13 ページの「ワークスペース管理ツールの起動」を参照してください。

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. Forte 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。

Forte TeamWare は、SCCS でバージョン管理されているファイルだけ認識します。ファイルが SCCS のバージョン管理下でない場合は、72 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後、開発チームのメンバーは親ワークスペースのファイルのコピーを持つ自分専用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のトランザクションはこの親子関係を軸に回転します。すなわち、親ワークスペースからファイルをブリングオーバーし、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プットバックします。

子ワークスペースを作成する場合は、次のようにします。

1. ワークスペース管理を起動します。
詳細は、13 ページの「ワークスペース管理ツールの起動」を参照してください。
2. 取得する必要があるファイルが含まれている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。
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」 ▶ 「作成ブリングオーバー」を選択します。

「トランザクション」ダイアログボックスの「作成ブリングオーバー」タブが開きます (図 2-2 を参照)。

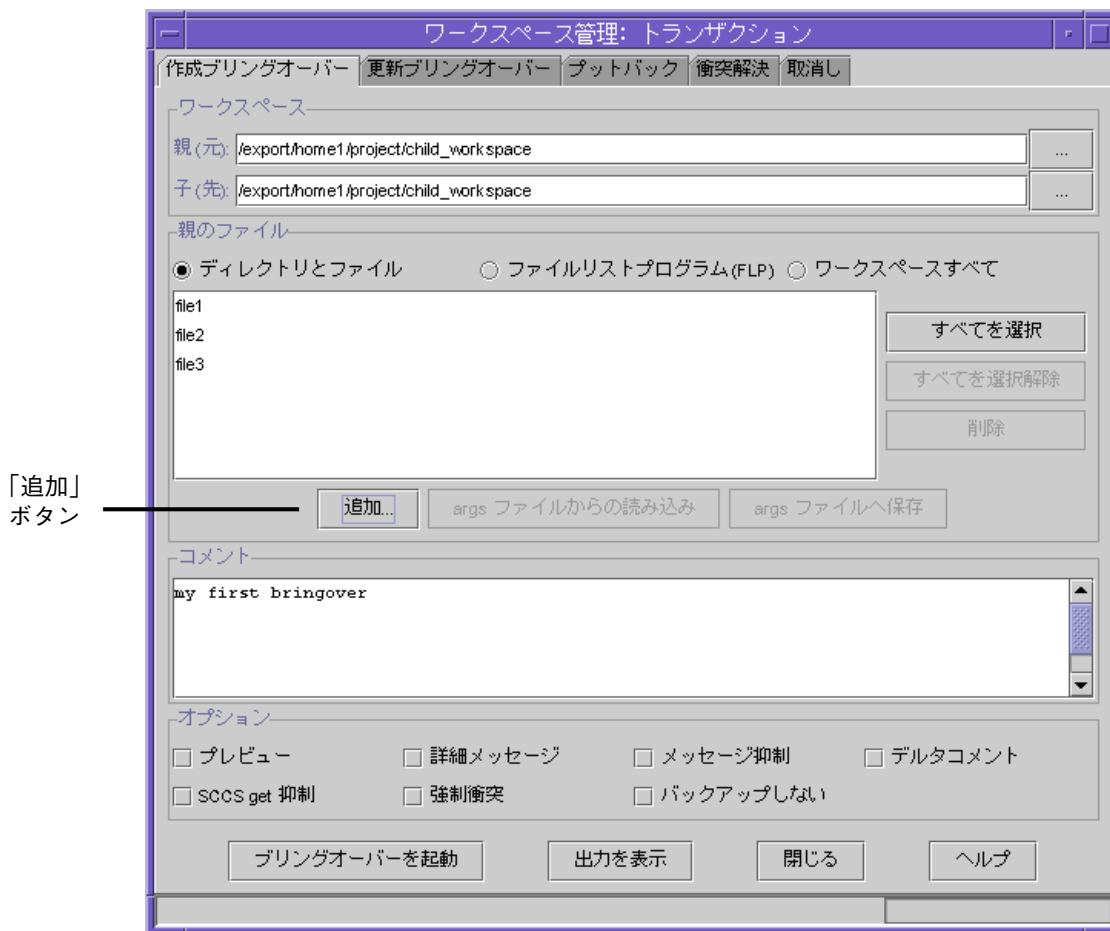


図 2-2 「トランザクション」ダイアログボックスの「作成ブリングオーバー」タブ

5. 次のいずれかの操作を行い、ブ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「./」を受け入れ、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で目的のファイルを探します (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に移動することはできません。

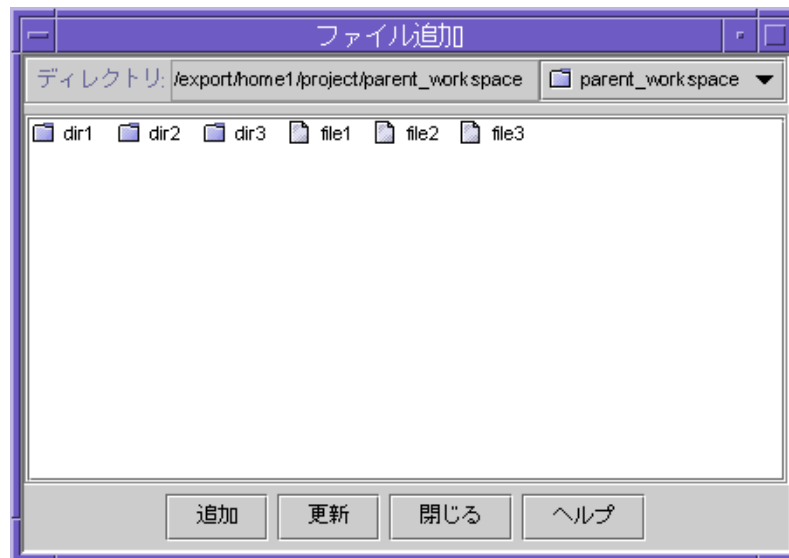


図 2-3 「ファイル追加」 ダイアログボックス

- 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. 「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「ブリングオーバーを起動」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、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、開発者はこのワークスペースから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します。ファイルをチェックアウトする方法については、76 ページの「ファイルのチェックイン」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、22 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、親から子にファイルを「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「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場合は、次のようにします。

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、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」 ▶ 「更新ブリングオーバー」を選択します。

「トランザクション」ダイアログボックスの「更新ブリングオーバー」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。

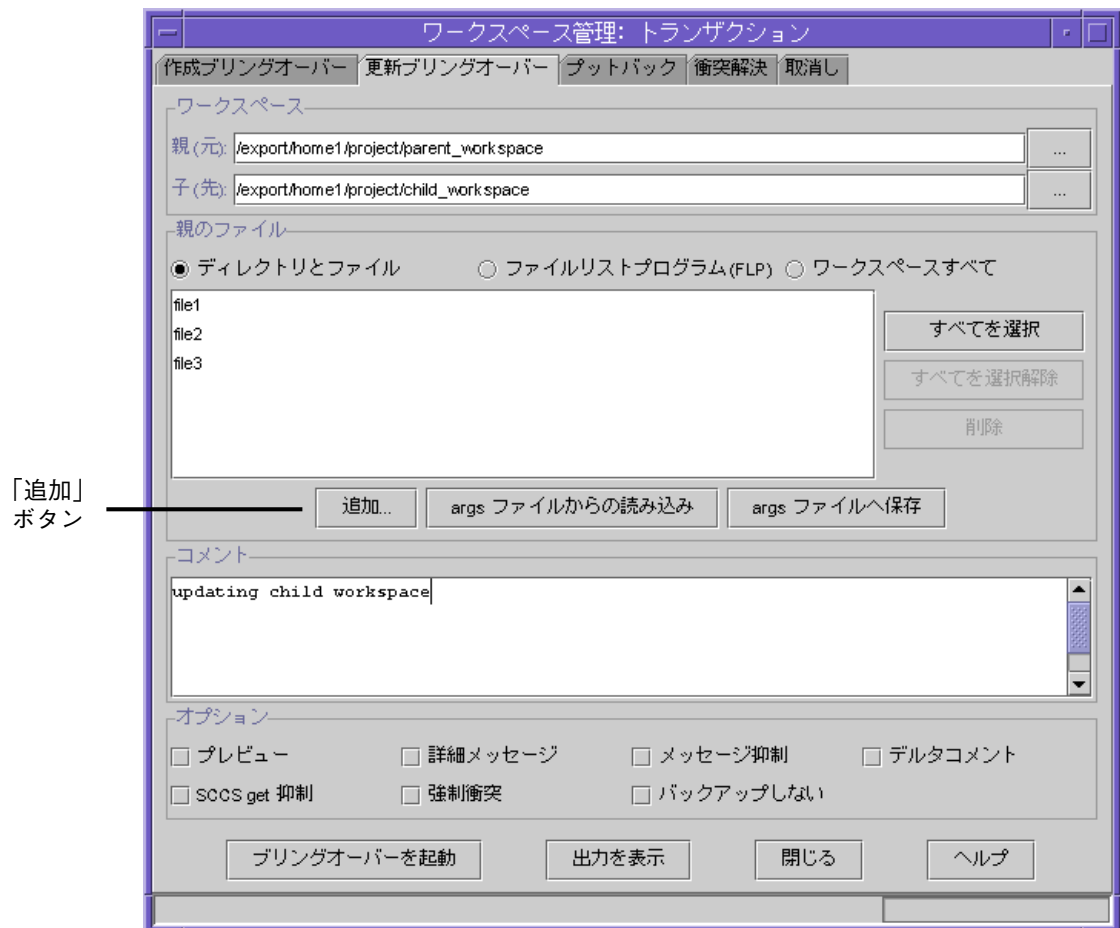


図 2-4 「トランザクション」ダイアログボックスの「更新ブリングオーバー」タブ

4. 次のいずれかの操作を行い、ブ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「./」を受け入れ、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で目的のファイルを探します (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に移動することはできません。

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、複数のファイルとディレクトリをまとめて選択できます。

c. 「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」タブにある「ブリングオーバーを起動」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場合は、次のようにします。

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、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」▶「プットバック」を選択します。

「トランザクション」ダイアログボックスの「プットバック」タブが表示されます。

「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。

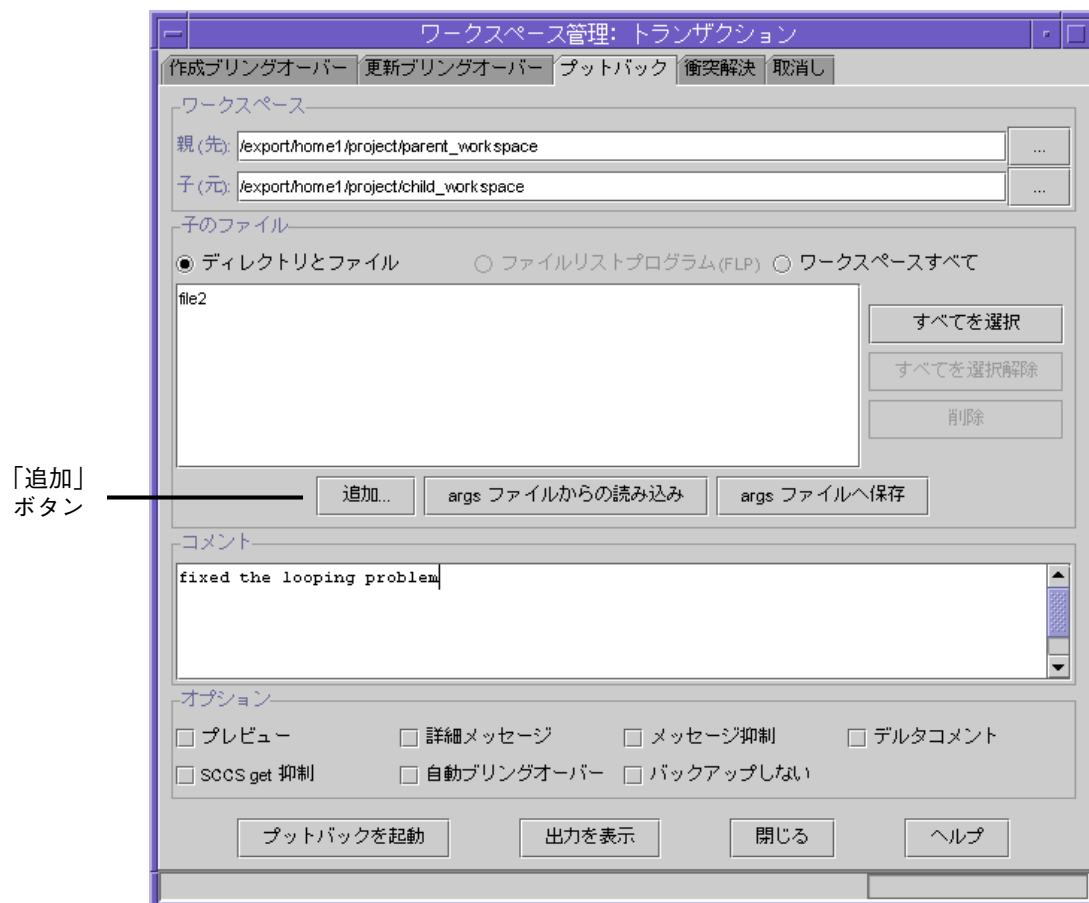


図 2-5 「トランザクション」ダイアログボックスの「プットバック」タブ

4. 次のいずれかの操作を行い、プットバック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「./」を受け入れ、すべてのファイルをプットバックする。
 - 「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プットバックする。
 - 「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で目的のファイルを探します (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に移動することはできません。
 - 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. 「追加」ボタンをクリックし、「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」タブにある「プットバックを起動」ボタンをクリックします。「トランザクション出力」ウィンドウにプットバック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となります。

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。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、「取り消し」トランザクションで取り消すことができます。詳細は、この後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「取り消し」タブを使用し、ワークスペースに対する直前の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だけです。

プットバックまたはブ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、ブ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ファイルが新たに作成されている場合)、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場合は、次のようにします。

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。

「取り消し」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という管理作業のすべてを行います。ワークスペース管理を使用してワークスペースの名前変更や移動、削除が行われることによって、ワークスペース管理は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を行うには、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には、「ワークスペース管理」メニューにあるコマンドを使用します。

ワークスペースを名前変更または移動する場合は、次のようにします。

1. 「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」 ▶ 「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表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、この節で紹介する手順を使用してください。

ワークスペースの削除

注 – この手順は、システムからファイルおよびディレクトリを完全に削除します。

ワークスペースを削除する場合は、次のようにします。

1. 「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。
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. 「ソースファイルと Codemgr_wsdata ディレクトリの削除」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。対応するディレクトリ、その中のファイル、サブディレクトリが完全に削除されます。

ワークスペース設定の解除

Forte TeamWare ワークスペースを通常のディレクトリに戻す場合は、次のようにします。

1. 「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。

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内にだけ記録されます。
- 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。
- 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 10 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場合は、次のようにします。

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クリックして選択状態にします。
2. 「ワークスペース」▶「履歴表示」を選択します。
「履歴表示」ウィンドウが開きます (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

デフォルトでは、「履歴表示」ウィンドウには、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

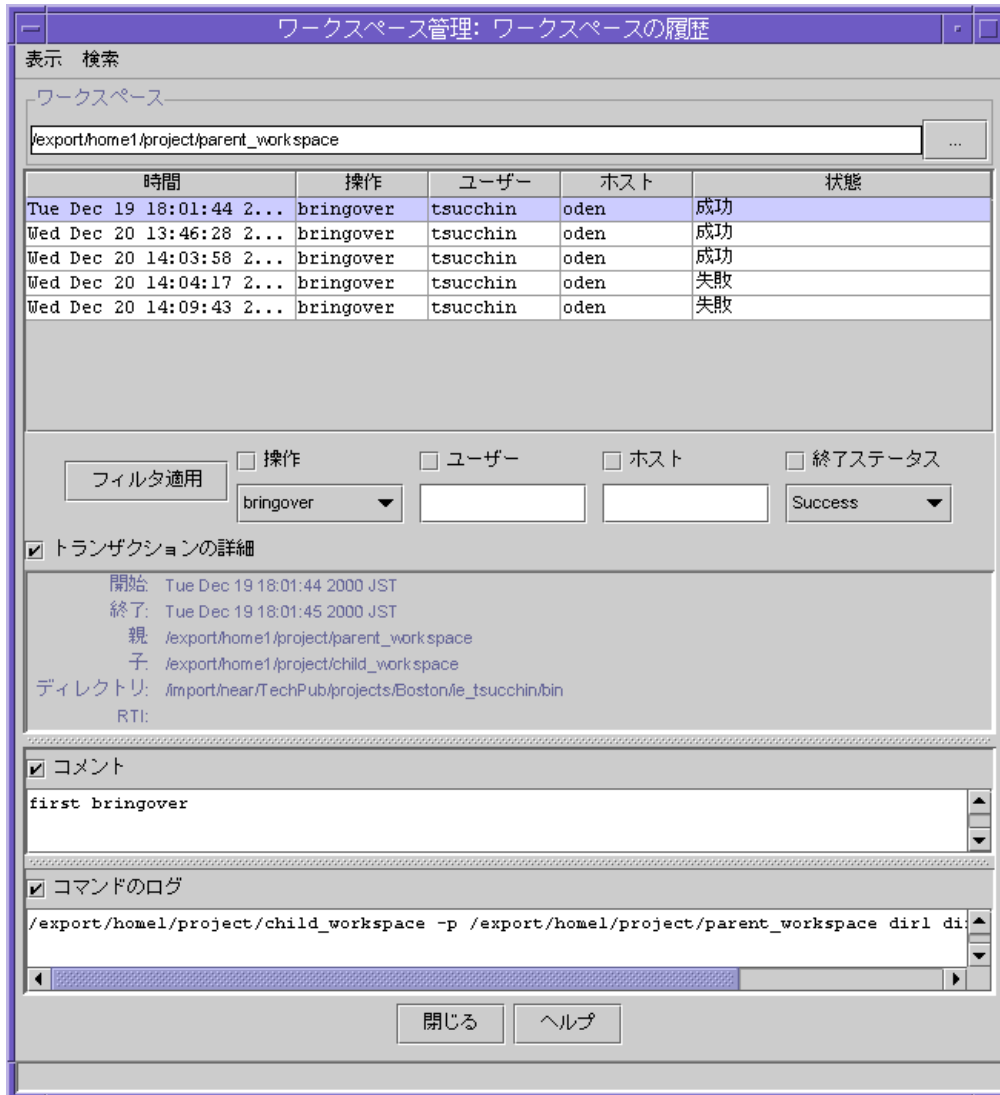


図 2-6 「履歴表示」ウィンドウ

トランザクション情報の追加と削除

「履歴表示」ウィンドウのトランザクション情報を追加または削除する場合は、次のようにします。

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「表示」から、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。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「履歴表示」ウィンドウに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。デフォルトでは、トランザクション情報およびフィルタオプションとして「TW リリース」以外の情報が表示されます。

「履歴表示」ウィンドウにフィルタオプションを追加する場合は、次のようにします。

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「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情報の選別

「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別する場合は、次のようにします。

1. 「履歴表示」ウィンドウから使用するフィルタ (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クリックして選択します。
複数のフィルタを選択できます。
2. 選択したフィルタ条件に従って、必要な情報を入力または選択します。
3. 「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。

コマンドログを検索する場合は、次のようにします。

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. 「コメント検索」ダイアログボックスに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ったと、「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場合は、次のようにします。

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. 「コマンドのログの検索」ダイアログボックスに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ったと、「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。

- 35 ページの「ブリングオーバーとプットバックのオプション」
- 39 ページの「独自のブリングオーバーおよびプットバック用ファイルリストの作成」
- 41 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」
- 42 ページの「ワークスペースに分かりやすい名前を付ける」
- 43 ページの「ワークスペースの親の変更」
- 49 ページの「ツール属性によるワークスペース管理のカスタマイズ」
- 52 ページの「環境変数の設定」
- 54 ページの「RCS プロジェクトの変換」

ブリングオーバーと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。

- トランザクションの実行のたびに設定可能な「トランザクション」ダイアログボックスのオプション
- あらゆるブリングオーバー/プットバックトランザクションに適用されるツール属性

ブリングオーバー/プットバックトランザクションごとのオプションの設定

ブリングオーバー / プットバックトランザクションごとのオプションを設定する場合は、次のようにします。

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)。

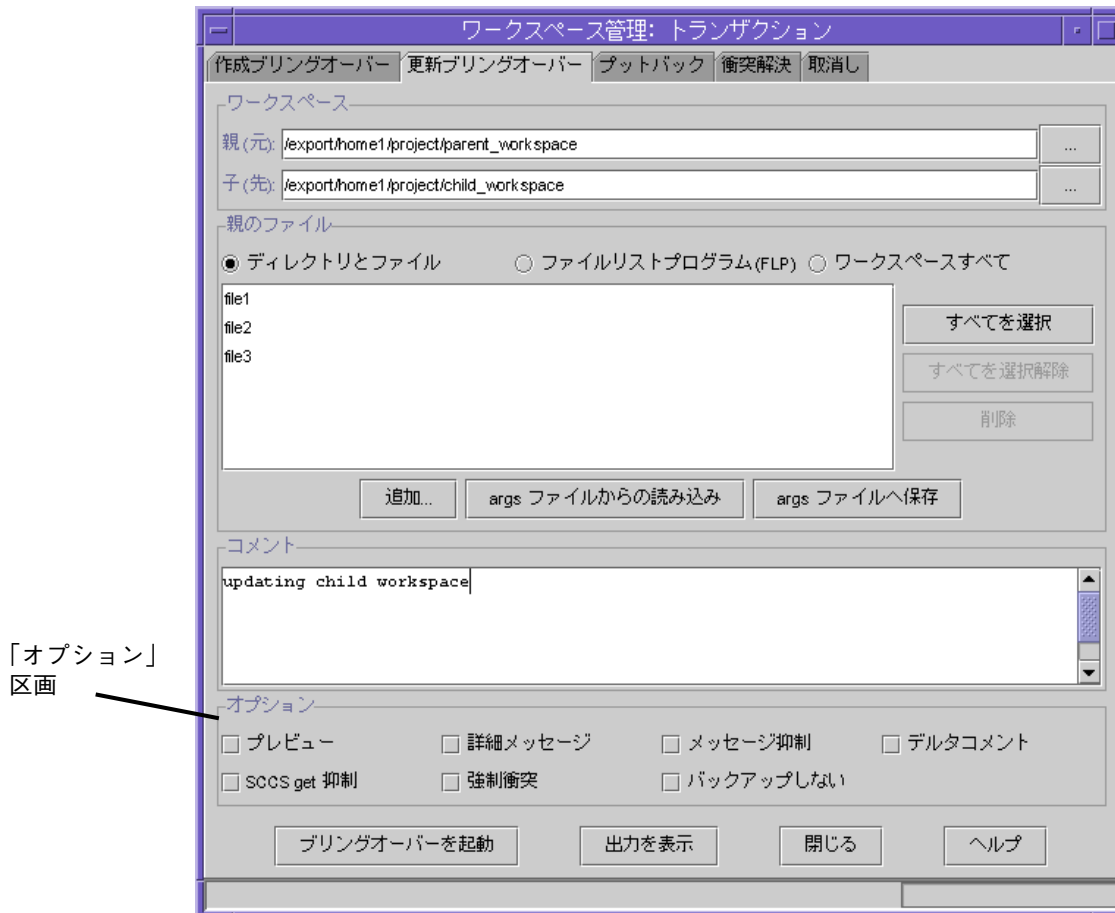


図 3-1 「トランザクション」ダイアログボックスの「オプション」区画

「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」、「プットバック」タブのオプション区画には、表 3-1 に示すチェックボックスがあります。

表 3-1 プリングオーバーとプットバックのオプション

チェックボックス	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、実際のファイル転送を伴わないトランザクション処理が行われます。「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予測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 get 抑制	プリングオーバーの実行時に SCCS の get コマンドが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 g-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 get 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	転送先のワークスペースにある Codemgr_wsdata/backup/files ディレクトリに既存のファイルを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。

ツール属性の設定

ブリングオーバーとプットバックに対するオプションを設定するには、「ツール属性」ダイアログの「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「ブリングオーバー/プットバック」タブを開くには、次のようにします。

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。

「ブリングオーバー/プットバック」のツール属性には、表 3-2 に示すチェックボックスがあります。

表 3-2 ブリングオーバー/プットバックのツール属性

チェックボックス	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 Codemgr_wsdata/args 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。デフォルトは有効です。
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。デフォルトは有効です。
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。デフォルトは無効です。
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。デフォルトは有効です。

独自のブリングオーバーおよびプットバック用 ファイルリストの作成

ワークスペース管理は、ブリングオーバーまたはプットバック対象の1つのワークスペースに対して1つのファイルリストを保有します。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および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またはプットバックトランザクションでは、デフォルトでこのファイル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、プログラムを実行して、ブリングオーバーおよびプットバック用の独自のファイルリストを作成することもできます。

デフォルトでは、Forte TeamWare は、ファイルリストプログラム (FLP) を使用し、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対象とな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、「ファイルリスト」区画で指定されたファイルの他に、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。

ブリングオーバーおよび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場合は、次のようにします。

1. ブリングオーバーおよび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。

例:

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

2. 「ブリングオーバー/プットバック」タブにある「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」をクリックします。
4. 「FLPS 追加」ダイアログボックスで適切な 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使用する FLP プログラムを設定したことになります。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。この変数の設定は、ワークスペース管理ツールのデフォルトの FLP (def.dir.flp) に優先します。

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

コマンド行からプリングオーバーまたはプ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 10 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、他のユーザーに電子メールを自動的に送信す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、ファイル名、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場合は、次のようにします。

1. ワークスペース管理ツールを起動します。
詳細は、13 ページの「ワークスペース管理ツールの起動」を参照してください。
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。
 - 「取り消し」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/re17/ver2` などの長いパス名になります。Forte TeamWare では、`current development area` や `Bob's workspace` 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」ウィンドウに表示され(「表示」▶「説明」で指定した名前)を選択)、電子メールによる通知で使用されます。

個人的な目的でワークスペースに詳細な説明を付けることもできます。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場合は、次のようにします。

1. ワークスペース管理ツールを起動します。
詳細は、13 ページの「ワークスペース管理ツールの起動」を参照してください。
2. 「ファイル」▶「ワークスペース作成」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」▶「履歴表示」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」タブをクリックします。
7. 「名前」フィールド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。このとき、情報が保存されることはありません。

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
-n	名前だけ表示します。
-d	説明だけ表示します。
-a	名前と説明の両方を表示します (デフォルト)。
wsname	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 10 章を参照してください。

ワークスペースの親の変更

3 ページの「親ワークスペースと子ワークスペース」で説明しているように、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には、この親子関係を変更する手段が用意されています。この節では、次のことを説明します。

- ワークスペースの親を変更する理由
- ワークスペースの親を変更する方法
- 親の変更例

ワークスペースの親を変更する理由

ワークスペースの親は、永久に変更することも、一時的に変更することもできます。ワークスペースの親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。たとえば、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は、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。46 ページの「親の変更例」は、このような親の変更例です。
- 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を回復する場合。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を変更する方法

ワークスペースの親を永久に変更する方法は「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 & ドロップする方法があります。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のみ有効になります。

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を変更する場合は、次のようにします。

1. ワークスペース管理を起動します。
詳細は、13 ページの「ワークスペース管理ツールの起動」を参照してください。
2. 「ファイル」 ▶ 「ワークスペース作成」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」 ▶ 「履歴表示」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、その子ワークスペースは親を失うことになります。「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要求されます。新しい親子関係は画面に自動的に反映されます。

また、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。

一時的な親の変更

「更新ブリングオーバー」タブ(「アクション」▶「更新ブリングオーバー」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、その更新ブリングオーバーの間だけワークスペースの親を変更できます。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、そのプットバックトランザクションの間だけワークスペースの親を変更することもできます。どちらの場合も、新しい親子関係が有効なのはそのトランザクションの間だけです。

親の変更例

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、通常は製品の次のリリースにも適用する必要があります。Forte 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)。

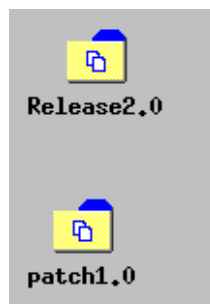


図 3-2 親子関係のない2つのワークスペース

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)。

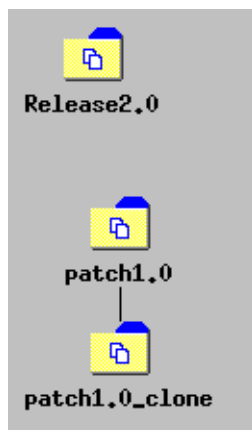


図 3-3 作成したワークスペースのクローン

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)。

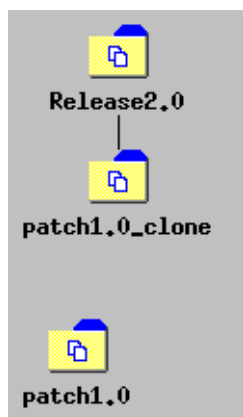


図 3-4 クローンの親を Release2.0 に変更

4. 新しい親の Release2.0 から更新ブリングオーバーを実行し、patch1.0_clone を更新します (図 3-5 を参照)。

これで、patch1.0_clone 内のパッチに加えられた修正と、Release2.0 にあるファイルがマージされます。

5. patch1.0_clone から Release2.0 に変更をプットバックします (図 3-5 を参照)。

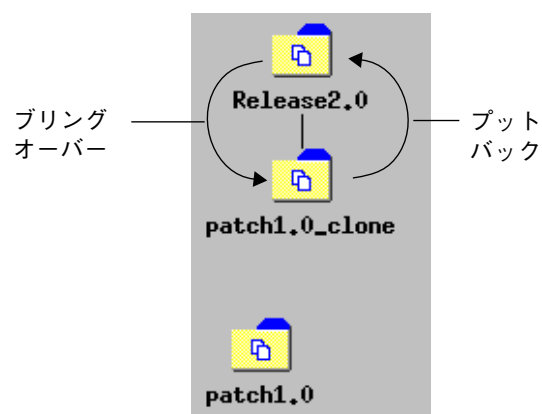


図 3-5 ファイルをブリングオーバー、マージして、新しいリリースに適用

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できています。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)。

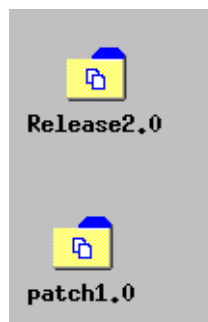


図 3-6 修正内容が取り込まれた Release2.0 (patch1.0_clone は削除)

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、ワークスペース管理の次の動作をカスタマイズできます。

- 「ワークスペース管理」ウィンドウの機能
- ブリングオーバーおよび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」、「ブリングオーバー/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、92 ページの「ファイルマージのオプション」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のメインウィンドウの動作を変更できます (図 3-7 を参照)。表 3-4 は、「ワークスペース管理」タブの属性をまとめています。

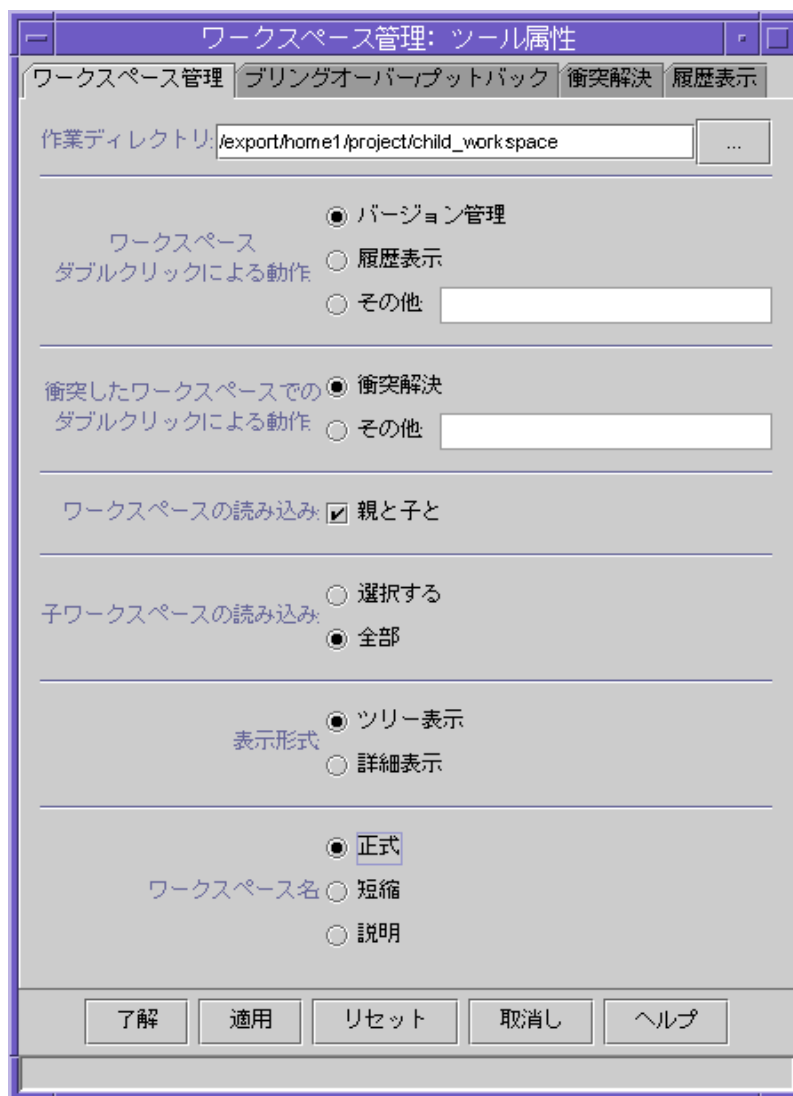


図 3-7 「ツール属性」の「ワークスペース管理」タブ

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の処理の基準となるディレクトリを指定します。
ワークスペース 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。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 (twversion) 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。作業ディレクトリと検索パスに基づいてコマンドを実行するために必要なパス名を入力します。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」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。読み込む子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。
表示形式	ワークスペース階層を縦方向に表示する場合は「ツリー表示」、ワークスペース階層を横方向に表示する場合は「詳細表示」を選択します。デフォルトは「ツリー表示」です。この属性は、ワークスペース管理のメインウィンドウの「表示」メニューにある上位 2 項目に対応しています。
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 (42 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)。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、39 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。以下は、CODEMGR_WSPATH 環境変数の設定例です。

```
% 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。

例:

```
% 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。この変数を設定すると、コマンド行コマンド (**bringover**、**putback**、**freezept 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、指定したワークスペースがデフォルトで自動的に使用されます。以下は、デフォルトのワークスペースの設定例です。

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

この変数の設定は、Forte TeamWare ツールを起動したときのワークスペースの読み込みにも関係します。

コマンド行コマンドについては、第 10 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 Forte TeamWare ツールの格納場所を指定できます。Forte TeamWare は、TeamWare ツールを探すとき、この変数に指定された場所を検索します。以下は、CODEMGR_PATH_ONLY 環境変数の設定例です。

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

CODEMGR_PATH_ONLY 変数が設定されていない場合、Forte 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 プロジェクトの変換

`rcs2ws` は、RCS (Revision Control System) の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、その階層をたどって、すべての RCS ファイルを SCCS ファイルに変換します。

`rcs2ws` プログラムは、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。親ディレクトリの階層構造は、`rcs2ws` の影響を受けません。`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、`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。また、`rcs2ws` は、`PATH` 変数の設定に基づいてこれらのコマンドを検索します。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、`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、`-w` オプションは省略できます。

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

```
% 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の場合は、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には、次のコマンドを使用します。

```
% 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の制御

Forte 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。

- 57 ページの「アクセス権の設定」
- 61 ページの「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」
- 67 ページの「ワークスペースのロックの解除」

アクセス権の設定

Forte 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、58 ページの「全体レベルのアクセス権の設定」、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、60 ページの「グループまたはユーザーレベルのアクセス権の設定」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は、次に挙げる Forte 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すると、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。

全体レベルのアクセス権の設定

全体にわたるアクセス権を設定する場合は、次のようにします。

1. ワークスペース管理を起動します。
詳細は、13 ページの「ワークスペース管理ツールの起動」を参照してください。
2. 「ファイル」▶「ワークスペースの読み込み」を選択します。
3. 「ワークスペースの読み込み」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。

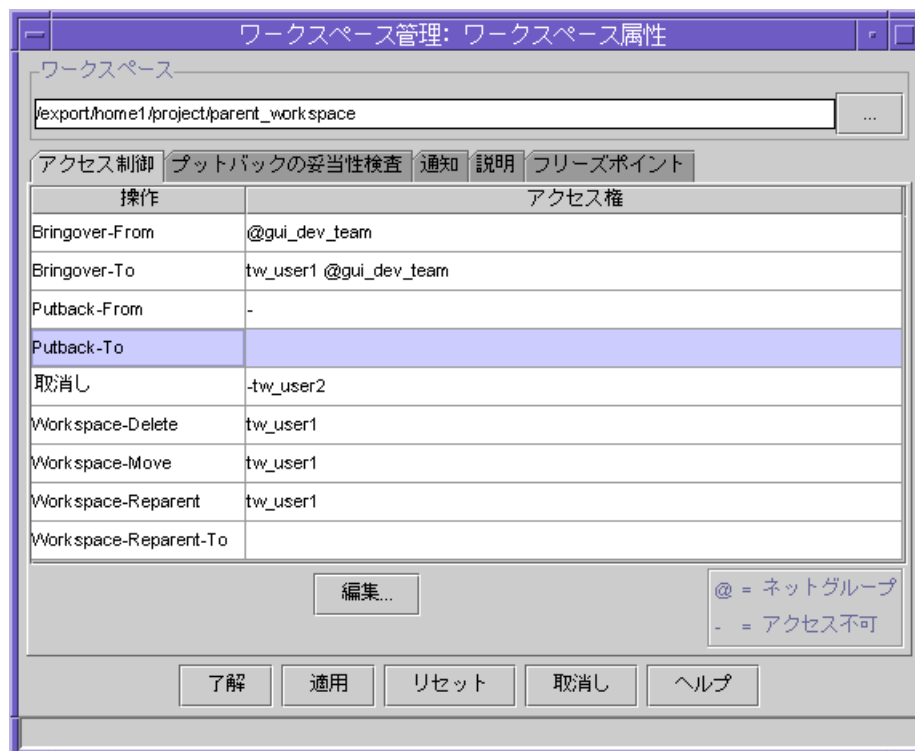


図 4-1 「ワークスペース属性」ダイアログボックスの「アクセス制御」タブ

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. 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。

9. 「了解」をクリックします。

グループまたはユーザーレベルのアクセス権の設定

1. ワークスペース管理を起動します。
詳細は、13 ページの「ワークスペース管理ツールの起動」を参照してください。
2. 「ファイル」▶「ワークスペースの読み込み」を選択します。
3. 「ワークスペースの読み込み」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」をクリックして、「アクセス制御属性」ダイアログボックスを表示します。
8. 「指定」ラジオ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て、個人ユーザーまたはグループのどちらの指定を行うのかを指示します。
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. 「追加」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。追加したユーザーのリスト内の順序を変更するには、「上へ移動」および「下へ移動」ボタンを使用します。
ユーザーの順序の重要性については、61 ページの注を参照してください。
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権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、個人レベルではユーザーに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レベルで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、ユーザーにアクセス権が付与されます。グループが先に登録されている場合、アクセス権は付与されません。

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、ワークスペースに対するプットバックを制限することができます。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。63 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、ワークスペース管理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。

プットバックの妥当性検査には、表 4-1 に示す 3 つのモードがあります。

表 4-1 プットバックの妥当性検査のモード

モード	ユーザー	ワークスペース管理
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プットバックを許可します。
妥当性検査をする	プットバックの妥当性検査を有効にします。	プ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、単にパスワードが記録されます。
妥当性を検査してパスワードを検査	プットバックの妥当性検査プログラムを作成し、プ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。	プ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。統合要求 ID は妥当性検査プログラムに渡されます。妥当性検査プログラムは、プ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。117 ページの「SCCS ファイルの属性の設定」を参照してください。

プットバックの妥当性検査を有効にする

プットバックの妥当性検査を有効にする場合は、次のようにします。

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」▶「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の「プ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)。
5. 「プ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。

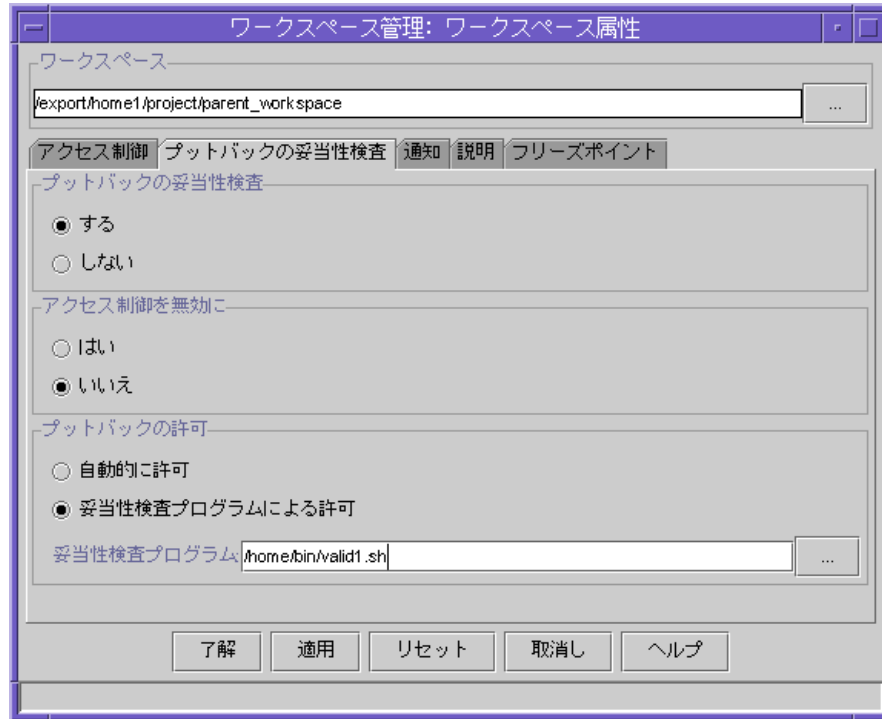


図 4-2 「ワークスペース属性」ダイアログボックスの「プットバックの妥当性検査」タブ

プ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プットバックのたびに、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、プットバックは取り消されます。

プットバックの妥当性検査プログラムは、Forte 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。

プットバックの妥当性検査プログラムを起動する場合は、次のようにします。

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. 「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「プットバックの妥当性検査」タブをクリックします。
5. 「プ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. 「プットバックの許可」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

ここでは、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の検査に使用できる妥当性検査プログラムを紹介します。妥当性検査プログラムは、プットバックによって渡される引数を受け取り、プ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。

プットバックを行おうとすると、統合要求 ID の入力が必要で、続いて、次の引数を使って妥当性検査プログラムが起動されます。

- プ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に、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例

```
## Start of sample validation program
#!/bin/sh
##
## Here is the list of users who can perform
## putback to the workspace
##
valid_users="nikm azv builder vvg aar"
```

コード例 4-1 プットバックの妥当性検査プログラム例 (続き)

```
##
## 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=祖at $1é
##
## Validate user
##
isValid="false"
for u in $valid_users
do
    if [ "$User" = "$u" ] ; then
        isValid="true"
        break
    fi
done
if [ "$isValid" = "false" ] ; then
    # invalid user
    echo ""
    echo "*** Validation failed:User $User is not allowed \
        to putback to $Parent"
    echo ""
    exit 1
fi
##
## Validate Integration Request Id (IRI)
##
```

コード例 4-1 プットバックの妥当性検査プログラム例 (続き)

```
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 $uuë)
    if [ "$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、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、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:  
Command: bringover (pid 20291), user: jack, machine: holiday,  
time: 12/02/91 16:25:23  
(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り、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、手動でロックを解除してください。

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い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場合は、次のようにします。

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、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「ロックの編集」ダイアログボックスを閉じます。

第5章

ファイルの管理


Forte TeamWare ワークスペースを作成すると、プロジェクトファイルに対する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。

- 69 ページの「バージョン管理ツールの起動」
- 72 ページの「ワークスペースへのファイルの追加」
- 73 ページの「ファイルのチェックアウト」
- 75 ページの「ファイルの編集」
- 76 ページの「デフォルトのエディタの変更」
- 76 ページの「ファイルのチェックイン」
- 78 ページの「ファイルに加えた変更の取り消し」
- 79 ページの「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、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、すべての変更を履歴ファイルに記録します。

バージョン管理ツールを起動するには、次のいずれかの操作を行います。

- Forte for Java の TeamWare ツールバーにある「TeamWare バージョン管理」ボタン  をクリックする。

注 – デフォルトでは、TeamWare の簡略ツールバーが表示されます。TeamWare ツールバー全体を表示する方法については、159 ページの「Forte for Java の TeamWare ツールバーの表示方法」を参照してください。

- Forte for Java の「エクスプローラ」ウィンドウから、次の操作をする。

注 – Forte for Java の「エクスプローラ」ウィンドウに Forte TeamWare のディレクトリを表示するには、まずそのディレクトリをマウントする必要があります。詳細は、158 ページの「Forte TeamWare ファイルシステムのマウント」を参照してください。

- i. Forte TeamWare のディレクトリをマウスの右ボタンでクリックします。
- ii. 「TeamWare」▶「バージョン管理」を選択します。

Forte TeamWare ツールの起動と Forte for Java IDE 内からのコマンドの使用方法についての詳細は、第 11 章を参照してください。

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

バージョン管理ツールを起動すると、「バージョン管理」ウィンドウが表示されます。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)。

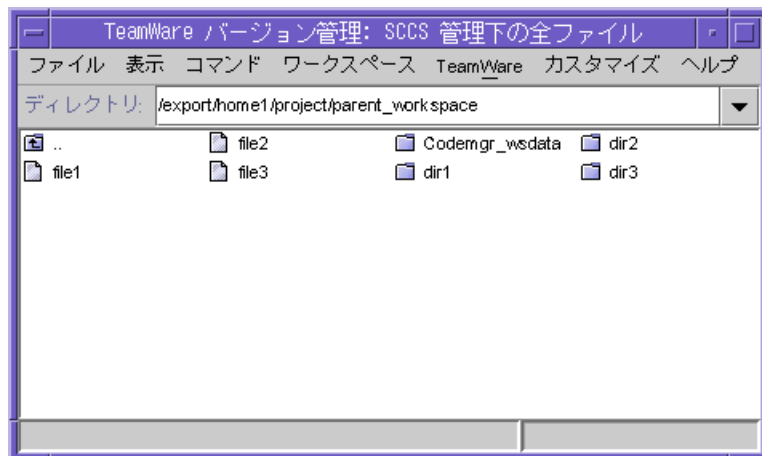


図 5-1 「バージョン管理」 ウィンドウ

「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。72 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。

「バージョン管理」ウィンドウには、表 5-1 に示すメニューがあります。

表 5-1 「バージョン管理」ウィンドウのメニュー

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。
ワークスペース	SCCS 管理下にあるファイルを検索、削除、移動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。

ディレクトリは、そこに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含む階層構造の入れ物であり、その中にさらに SCCS ファイルやディレクトリが含まれていることがあります。ファイルは、SCCS 管理下にあるファイルのみ表示されます。ディレクトリ内の SCCS 管理下にはないファイルを表示するには、「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

Forte TeamWare のワークスペース管理ツールでファイルを扱うには、そのファイルを SCCS でバージョン管理する必要があります。

既存の Forte TeamWare ワークスペースにファイルを追加する場合は、次のいずれかの操作を行います。

- Forte for Java の「エクスプローラ」ウィンドウから、次の操作をする。
 - i. Forte TeamWare ディレクトリに追加するファイルを選択します (ディレクトリはマウントされている必要があります)。
ファイルは [チェックインが必要] とマークされています。
 - ii. マウスの右ボタンをクリックし、「チェックイン」を選択します。
 - iii. 「バージョン管理: チェックインダイアログ」にコメントを入力します。
 - iv. 「チェックイン」をクリックします。
ファイルが「チェックイン」とマークされます。
- Forte for Java のメインウィンドウの「バージョン管理」メニューから、次の操作をする。
 - i. Forte for Java の「エクスプローラ」ウィンドウで、チェックインするファイルを選択します。
ファイルは [チェックインが必要] とマークされています。
 - ii. 「バージョン管理」▶「チェックイン」を選択します。
 - iii. 「バージョン管理: チェックインダイアログ」にコメントを入力します。
 - iv. 「チェックイン」をクリックします。
ファイルが「チェックイン」とマークされます。

- 「ワークスペース管理」ウィンドウから (13 ページの「ワークスペース管理ツールの起動」を参照)、次の操作をする。
 - i. 「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。
ディレクトリを TeamWare ワークスペースにしていない場合は、16 ページの「親ワークスペースの作成」を参照してください。
 - ii. 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。
 - iii. 「TeamWare」▶「バージョン管理」を選択します。

注 – 「バージョン管理」ウィンドウは、Forte for Java の TeamWare ツールバーから「TeamWare バージョン管理」ボタン  をクリックすることで表示することもできます。


- iv. 「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。
- v. 「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することもできます。
- vi. 「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 人が子ワークスペースを 1 つ作成し、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場合は、次のいずれかの操作を行います。

- Forte for Java の「エクスプローラ」ウィンドウから、次の操作をする。
 - i. チェックアウトするファイルを選択します。
ファイルは「チェックイン」とマークされています。
 - ii. マウスの右ボタンをクリックし、「チェックアウト」または「チェックアウトして編集」を選択します。
- Forte for Java のメインウィンドウの「バージョン管理」メニューから、次の操作をする。
 - i. Forte for Java の「エクスプローラ」ウィンドウで、チェックアウトするファイルを選択します。
ファイルは「チェックイン」とマークされています。
 - ii. 「バージョン管理」▶「チェックアウト」(または「チェックアウトして編集」)を選択します。
- 「ワークスペース管理」ウィンドウから (13 ページの「ワークスペース管理ツールの起動」を参照)、次の操作をする。
 - i. 「TeamWare」▶「バージョン管理」を選択します。

注 - 「バージョン管理」ウィンドウは、Forte for Java の TeamWare ツールバーから「TeamWare バージョン管理」ボタン  をクリックすることで表示することもできます。

- ii. 「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 キーを押しながらかlick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。
- iii. 「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、76 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場合は、次のいずれかの操作を行います。

- Forte for Java の「エクスプローラ」ウィンドウから、次の操作をする。
 - i. 編集するファイル (チェックアウト済み) を選択します。

注 – ファイル形式を認識できない場合は、172 ページの「Forte for Java IDE エディタと Forte TeamWare ツール」を参照してください。

- ii. マウスの右ボタンをクリックし、「開く」を選択します。
- 「ワークスペース管理」ウィンドウから、次の操作をする。
 - i. 「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。
 - ii. 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。
 - iii. 「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。

注 – 「バージョン管理」ウィンドウは、Forte for Java の TeamWare ツールバーから「TeamWare バージョン管理」ボタン  をクリックすることで表示することもできます。

- iv. 「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。
- v. 「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き、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、76 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。「バージョン管理」ウィンドウを Forte for Java IDE から開いた場合は、IDE ファイルエディタが開きます。使用されるエディタの種類は、ファイル形式によって決まります。
- vi. 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。
76 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

Forte TeamWare の各ウィンドウで使用するデフォルトのエディタを変更する場合は、次のようにします。

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。
3. 「オプション」ダイアログボックスの「エディタの選択」フィールドに使用するエディタ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場合は、次のいずれかの操作を行います。

- Forte for Java の「エクスプローラ」ウィンドウから、次の操作をする。
 - i. チェックインするファイルを選択します。
ファイルは「チェックアウト」とマークされています。
 - ii. マウスの右ボタンをクリックし、「チェックイン」を選択します。
 - iii. 「バージョン管理: チェックインダイアログ」にコメントを入力します。
 - iv. 「チェックイン」をクリックします。
ファイルが「チェックイン」とマークされます。変更内容を親ワークスペースに統合する方法については、79 ページの「プットバックによる変更の統合」を参照してください。
- Forte for Java のメインウィンドウの「バージョン管理」メニューから、次の操作をする。
 - i. Forte for Java の「エクスプローラ」ウィンドウで、チェックインするファイルを選択します。
ファイルは「チェックアウト」とマークされています。

- ii. 「バージョン管理」 ▶ 「チェックイン」 を選択します。
- iii. 「バージョン管理: チェックインダイアログ」 にコメントを入力します。
- iv. 「チェックイン」 をクリックします。

ファイルが「チェックイン」とマークされます。変更内容を親ワークスペースに統合する方法については、79 ページの「プットバックによる変更の統合」を参照してください。
- 「ワークスペース管理」 ウィンドウから (13 ページの「ワークスペース管理ツールの起動」を参照)、次の操作をする。
 - i. 「TeamWare」 ▶ 「バージョン管理」 を選択します。

注 – 「バージョン管理」 ウィンドウは、Forte for Java の TeamWare ツールバーから「TeamWare バージョン管理」 ボタン  をクリックすることで表示することもできます。

- ii. 「バージョン管理」 ウィンドウで、ファイルが格納されているディレクトリに移動します。
- iii. 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。
- iv. 「コマンド」 ▶ 「チェックイン」 を選択します。
- v. 「バージョン管理: チェックインダイアログ」 にコメントを入力します。
- vi. 「了解」 をクリックします。


コメントが保存され、ファイルがチェックインされます。変更内容を親ワークスペースに統合する方法については、79 ページの「プットバックによる変更の統合」を参照してください。

ファイルに加えた変更の取り消し

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加えた変更を取り消す場合は、次のいずれかの操作を行います。

- Forte for Java の「エクスプローラ」ウィンドウから、次の操作をする。
 - i. チェックアウトされているファイルを選択します。
選択したファイルは「チェックアウト」とマークされます。
 - ii. マウスの右ボタンをクリックし、「アンチェックアウト」を選択します。
- Forte for Java のメインウィンドウの「バージョン管理」メニューから、次の操作をする。
 - i. Forte for Java の「エクスプローラ」ウィンドウで、チェックアウトされているファイルを選択します。
選択したファイルは「チェックアウト」とマークされます。
 - ii. 「バージョン管理」▶「アンチェックアウト」を選択します。
- 「ワークスペース管理」ウィンドウから、次の操作をする。
 - i. 「TeamWare」▶「バージョン管理」を選択します。

注 – 「バージョン管理」ウィンドウは、Forte for Java の TeamWare ツールバーから「TeamWare バージョン管理」ボタン  をクリックすることで表示することもできます。

- ii. 「バージョン管理」ウィンドウで、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。

iii. ファイルをクリックして選択状態にします。

Shift キーを押しながらかlick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。

iv. 「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、以前の状態に戻されます。

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、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場合は、次のようにします。

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」フィールド内をクリックし、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. 「プットバックを起動」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)。

5. 「閉じる」をクリックし、「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、25 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。親ワークスペースに変更されているファイルがある場合、それと同じファイルを子ワークスペースで変更し、プットバックすることはできません。プットバックするには、親ワークスペース内と子ワークスペース内のファイル間の相違を解決する必要があります。ファイルマージは、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。

- 82 ページの「ファイルマージツールの起動」
- 84 ページの「ワークスペース内の衝突の解決」
- 83 ページの「「ファイルマージ」ウィンドウの表示」
- 89 ページの「相違の解決」
- 91 ページの「自動マージ」
- 92 ページの「変更の取り消し」
- 92 ページの「ファイルマージのオプション」
- 95 ページの「ファイルのマージ例」

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、次のいずれかの操作を行います。

- Forte for Java の TeamWare ツールバーにある「TeamWare ファイルマージ」ボタン  をクリックする。

注 - デフォルトでは、TeamWare の簡略ツールバーが表示されます。TeamWare ツールバー全体を表示する方法については、159 ページの「Forte for Java の TeamWare ツールバーの表示方法」を参照してください。

- Forte for Java の「エクスプローラ」ウィンドウから、次の操作をする。
 - i. Forte TeamWare のディレクトリをマウスの右ボタンでクリックする。
 - ii. 「TeamWare」▶「ファイルマージ」を選択する。

Forte TeamWare ツールの起動と Forte for Java IDE 内からのコマンドの使用方法についての詳細は、第 11 章を参照してください。

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

ファイルマージツールは「衝突解決」タブからも起動できます。詳細は、83 ページの「「衝突解決」タブからのファイルマージの起動」を参照してください。

ワークスペース内の衝突

「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。

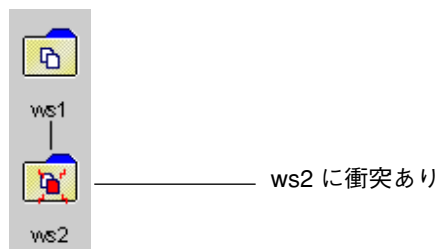


図 6-1 ワークスペース内の衝突例

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したら、最新のデルタを親からブリングオーバーしたバージョンとマージします。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、ブ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示されます。

「衝突解決」タブからのファイルマージの起動

「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、更新ブ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している場合は、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、マージした最終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各デルタは、その共通の祖先デルタとの比較で表示されます。

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

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。

ワークスペース内の衝突の解決

ワークスペース内の衝突を解決する場合は、次のようにします。

1. 「ワークスペース管理」ウィンドウで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。

ファイルマージが起動し、「ファイルリスト」区画一覧のファイルの処理が開始されます。一覧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。

ファイルマージの使用方法については、89ページの「相違の解決」を参照してください。

4. 衝突の解決を終えたら、ファイルを保存します。

ファイルマージを使用して、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。

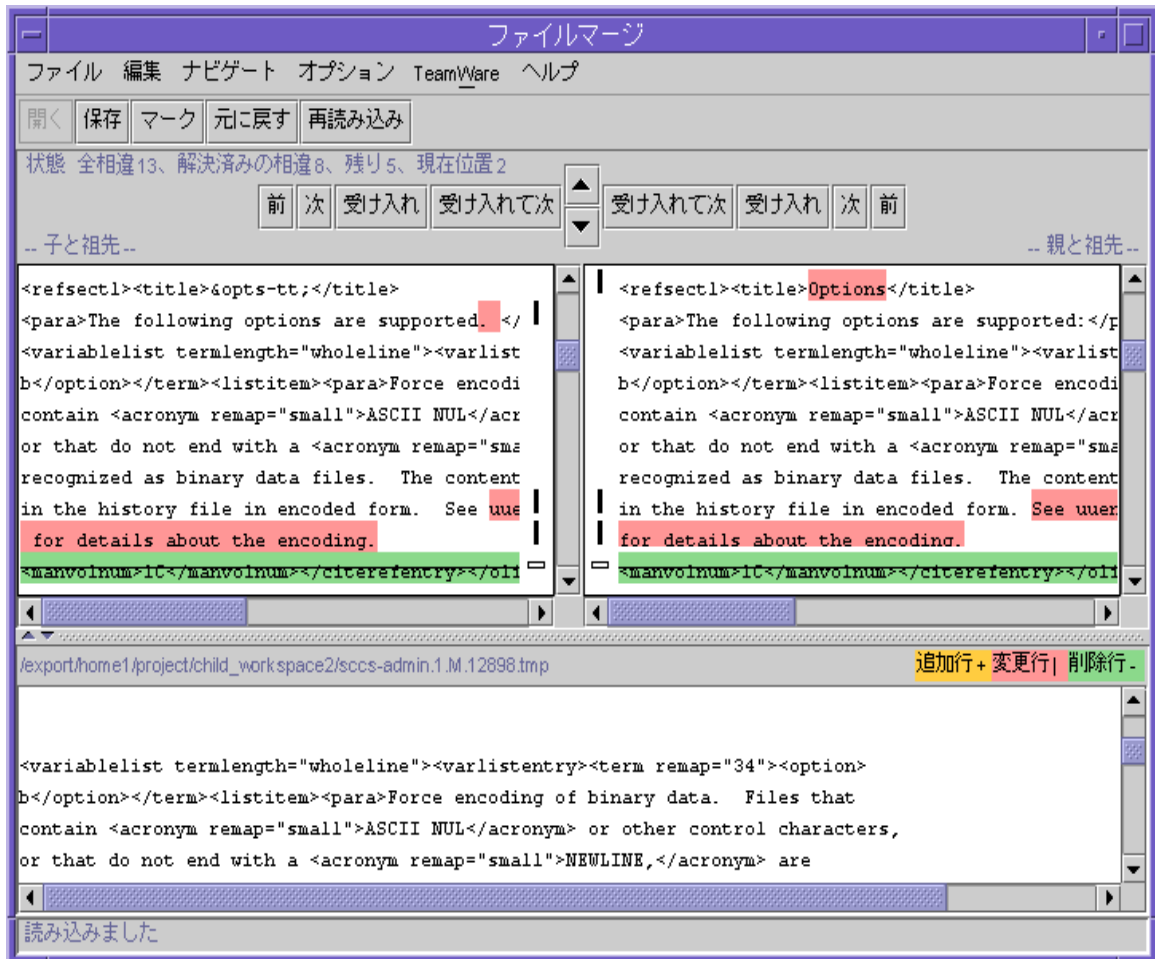


図 6-2 「ファイルマージ」 ウィンドウ

「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、白抜きのグリフを表示されます。

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つのファイルを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。各グリフと色は以下のことを意味します。

- 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。

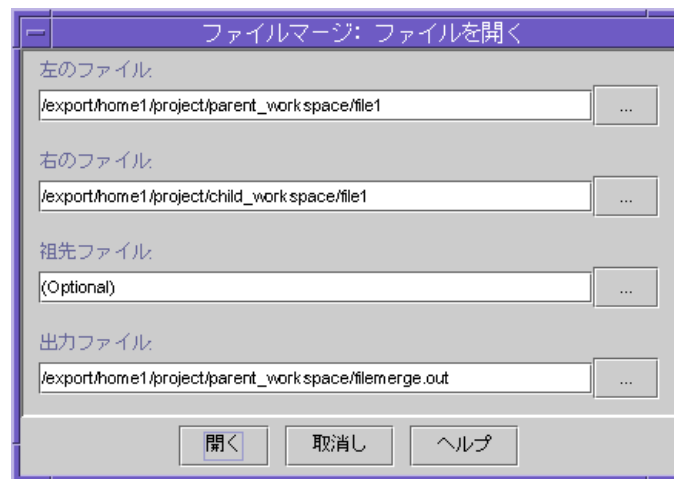


図 6-3 「ファイルマージ」ウィンドウの「ファイルを開く」ダイアログボックス

「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 ファイルマージの「ファイルを開く」ダイアログボックスのフィールド

フィールド	説明
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強調表示します。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、filemerge.out 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が書き込まれます。

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の部分には、それぞれのファイル名 (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。

トランザクションのチェックイン

チェックアウト中のファイルがあって、衝突が存在するワークスペースからプットバックを行うと、「トランザクションのチェックイン」ダイアログボックスが表示されます。このダイアログボックスには、以下に示すオプションがあります。

表 6-2 「トランザクションのチェックイン」ダイアログボックス

オプション	説明
チェックイン	ファイルの子ワークスペースに戻します。
アンチェックアウト	ファイルの子ワークスペース内のチェックアウト前の状態に戻します。
スキップ	次のチェックアウト済みファイルに進みます (ある場合)。

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も、マージ版を手作業で編集することもできます。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変更が完了したことを示すと、現在の相違は「解決された」こととなります。相違が解決されると、相違箇所のグリフは白抜きの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の上部の「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には、「オプション」▶「自動マージ」を選択します。このように設定すると、ファイルマージツールを起動したときに、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、91 ページの「自動マージ」を参照してください。

ファイル間の相違を解決する場合は、次のようにします。

1. 次のいずれかを参照して処理する相違箇所を決定します。

- 「ファイルマージ」ウィンドウの左上の「状態」行
- 強調表示されているグリフ

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に移動します。

別の相違箇所に移動するには、次のいずれかの操作をします。

- 「次」ボタンの1つをクリックする。
- 「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

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、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、白抜きのグリフが表示されます。

表 6-3 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は、それらの行のどちらもマージ版に取り込まれません。

表 6-3 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	自動マージなし

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のすべてが下の区画に表示されます。

ファイルマージのオプション

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。表 6-4 は、「ワークスペース管理」タブの属性をまとめています。

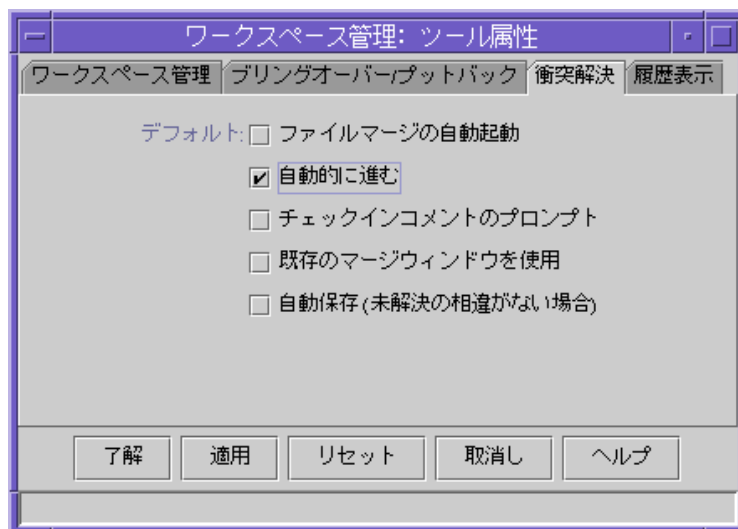


図 6-4 「ツール属性」ダイアログボックスの「衝突解決」タブ

表 6-4 「ツール属性」ダイアログボックスの「衝突解決」タブ

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、以降の衝突解決にも実行中のファイルマージを再利用します。
自動保存 (未解決の相違がない場合)	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式で、選択するたびに有効と無効が切り替わります。有効にすると、オプションの左側のボックスにチェックマークが付きます。

表 6-5 に、「ファイルマージ」ウィンドウの表示オプションを示します。

表 6-5 「ファイルマージ」ウィンドウの表示オプション

属性	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。
行番号表示	左右の区画内に行番号を表示します。
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3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 6-5 「ファイルマージ」ウィンドウの表示オプション (続き)

属性	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。子孫ファイルの `file_1` および `file_2` は、`matriarch` ファイルを編集することによって作成されています。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 &&&
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
```

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)。

状態 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

```

/export/home1/project/child_workspace/file_1
This line is deleted in file_1.
This line is in all three files.
This line is deleted in file_2.
This line is in all three files.
$$$ Added to file_1 $$$
This line is in all three files.
This line is modified in file_1 from matriarch.
This line is in all three files.
## Changed in file_1 and file_2 ##
This line is in all three files.
This line is deleted in file_2.
This line is in all three files.

This line is in all three files.

```

図 6-6 自動マージ後の子区画 (file_1)

```

/export/home1/project/child_workspace/file_2
This line is deleted in file_1.
This line is in all three files.
- This line is deleted in file_2.
This line is in all three files.

This line is in all three files.
| This line is altered in file_2 from matriarch.
This line is in all three files.
| ## Changed in file_1 and file_2 ##
This line is in all three files.
| ### Changed in file_2 ###
This line is in all three files.
+ ### Added to file_2 ###
This line is in all three files.

```

図 6-7 自動マージ後の親区画 (file_2)

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。デフォルトは有効です。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。

相違箇所を確認する

未解決の相違 (上記の例では file_1 と file_2 の行番号 6 の行) は強調表示されます。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「マージの結果」(filemerge.out) 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、一方のファイルにだけ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」ボタンをクリックする。
- 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」ボタンをクリックする。
- 手作業でマージ版を編集する。

マージ版を編集する

マージ版のファイルを編集する場合は、次のようにします。

1. マージ版のある下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になります。

「ファイルマージ」ウィンドウの左上の「状態」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。file_1 に 1 行 (3 番目の相違) が追加されたとき同様、この行もマージ版に取り込まれています。

第7章

高度なファイル管理

この章では、以下をはじめとする高度なファイル管理機能の使用方法を説明します。

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

子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、ブ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場合は、次のようにします。

1. 「ワークスペース管理」ウィンドウで「アクション」▶「更新ブリングオーバー」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「更新ブリングオーバー」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. 「ブリングオーバー」をクリックします。

更新ブリングオーバートランザクションの実行方法についての詳細は、22 ページの「子ワークスペースの更新 (更新ブリングオーバー)」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、Forte 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」▶「差分をテキストウィンドウで」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箇所がテキスト形式で表示されます。
- デルタを 2 つ選択して、「表示」▶「差分をコンテキスト付きで」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箇所がテキスト形式で表示されます。

ファイルの履歴を表示する場合は、次のようにします。

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には、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。

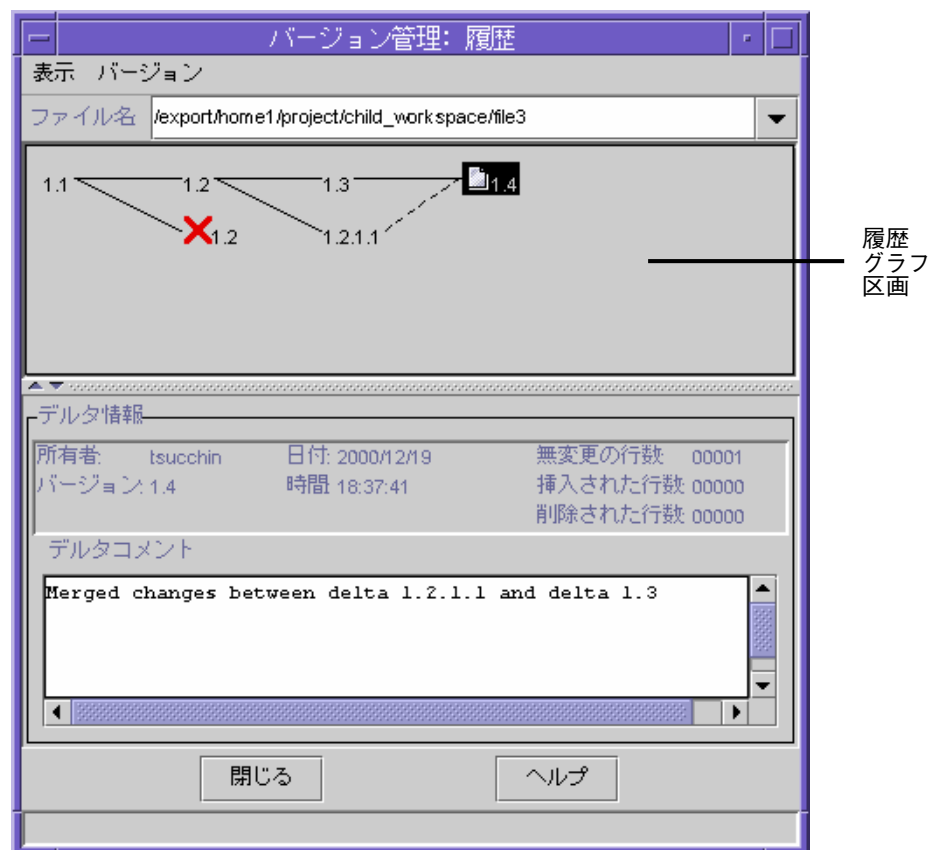


図 7-1 「ファイル履歴」ウィンドウの各部

「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 に示すメニューや区画などから構成されています。





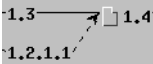

表 7-1 「ファイル履歴」ウィンドウの各部

項目	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ための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。
「デルタコメント」区画	デルタのコメントを表示します。

ファイル履歴表示で使用されるシンボル

「ファイル履歴」ウィンドウでは、表 7-2 に示す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> で設定します。117 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。

Forté 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 1.1 というデルタが割り当てられ、以降のデルタには、1.2、1.3 というように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 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 つ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 の形式で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 つ目の分岐の 2 つ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、そのデルタとトランクの祖先デルタ (1.3) 間のすべてのデルタを知ることはできません。

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。バージョン管理は、集められたデルタの 1 つのまとまりからファイルの 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場合は、次のようにします。

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」 ▶ 「アンチェックアウト」を選択して、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」を選択して、履歴グラフにファイルを表示します。
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」 ▶ 「差分をテキストウィンドウで」
 - 「表示」 ▶ 「差分をコンテキスト付きで」
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 - ファイルの名前変更や移動を行うには、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、ファイルの適切な関係や正確なファイル履歴が維持されます。

ファイル名の変更とファイルの移動

名前を変更したファイルや、移動したファイルがブリングオーバーまたはプットバックされた場合、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。そして、ファイル内容の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伝達します。

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1つずつ処理します。ディレクトリ名が変更されるか、ディレクトリが移動されると、その中のすべてのファイルが名前変更または移動されたかのように、ファイルを1つずつ評価します。

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。このファイルが子ワークスペースにブリングオーバーされた場合、ワークスペース管理は、その変更が次のどちらに該当す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。

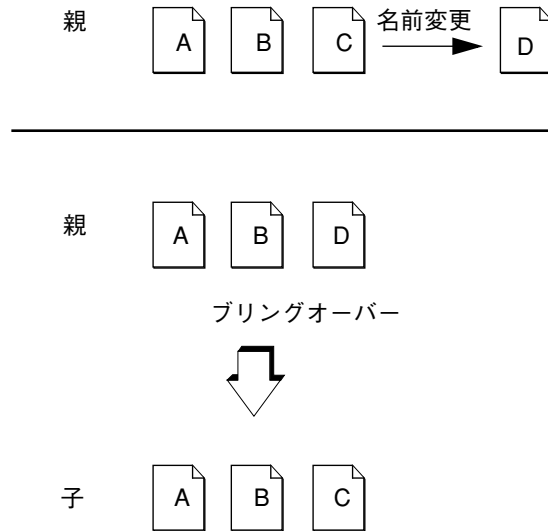


図 7-2 ファイル「C」の名前を「D」に変更

この判断は、プットバックトランザクションでも同様に行われます。

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履歴ファイルにその変更情報を記録します。下記は、ブリングオーバーまたはプットバックの出力の表示で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プ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、ブ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 :old_filename 新 :new_filename

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

名前の衝突

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」が「E」に名前変更されています。

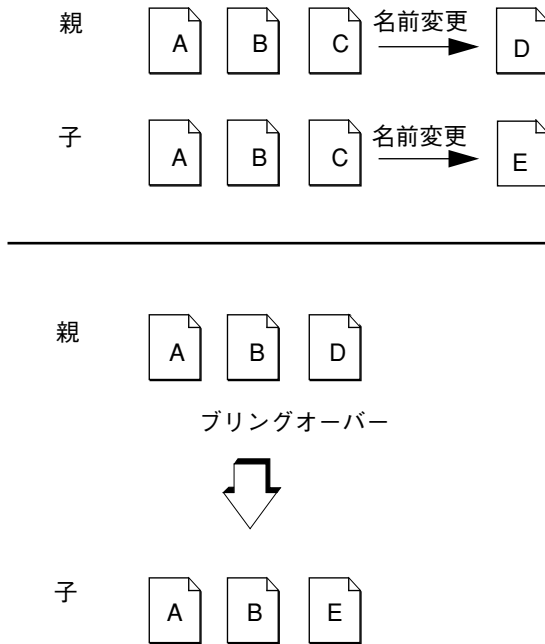


図 7-3 親と子ワークスペースにおけるファイル「C」の名前変更の衝突例

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、名前変更の衝突を通知します。

```

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

Forte TeamWare の外部からのファイルの削除

ワークスペースからファイルを削除する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。Forte 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、その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。

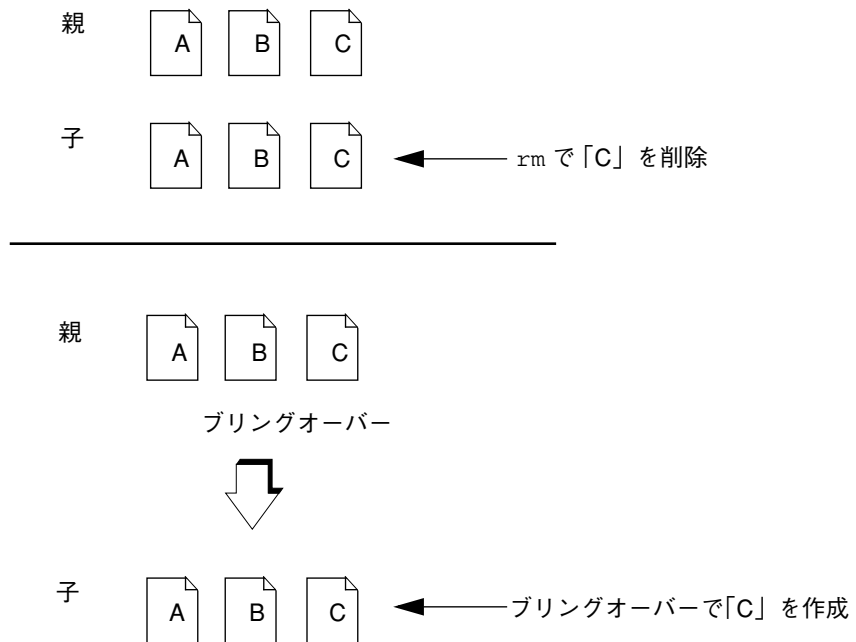


図 7-4 `rm` コマンドで子から「C」を削除し、ブリングオーバーで再作成

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 Forte TeamWare のコマンドを使用してください。

Forte TeamWare の内部からのファイルの削除

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。

Forte TeamWare ファイルを削除する場合は、次のようにします。

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達され、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心配はありません (111 ページの「Forte TeamWare の外部からのファイルの削除」を参照)。

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場合は、次のようにします。

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
```

Forte 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。

例：

```
/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. 「了解」をクリックします。

「バージョン管理」ウィンドウに、「カスタマイズ」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。そのような場合、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のは煩わしいものになります。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場合は、次のようにします。

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. 「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。

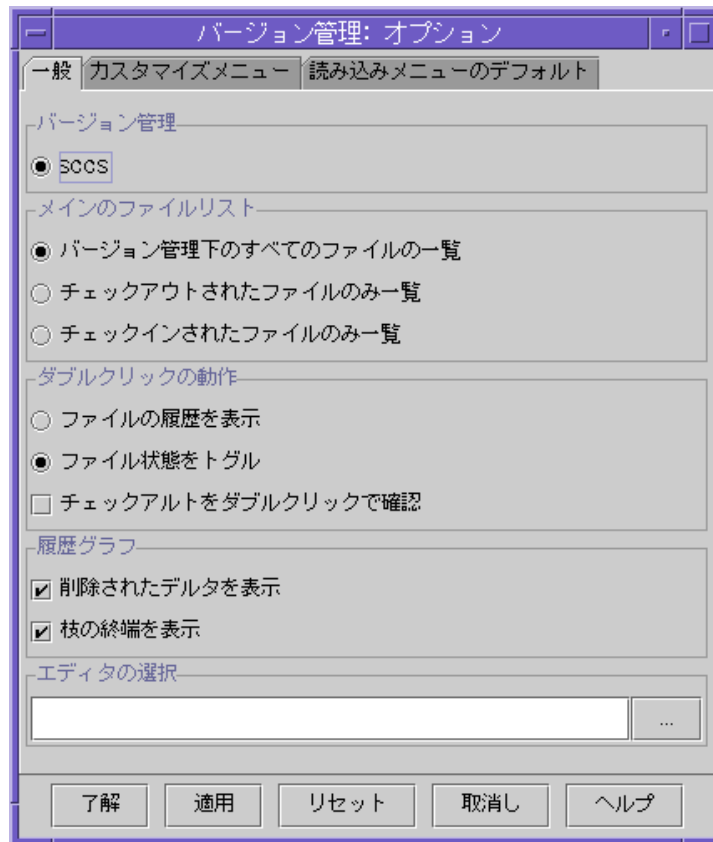


図 7-5 バージョン管理ツールの「オプション」ダイアログボックス

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の「一般」タブ

項目	説明
バージョン管理	現在は SCCS だけです。
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にチェックマークが表示されたままになります。
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。
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に自動的に起動するエディタを指定できます。使用するエディタのパスを入力します。

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 ファイルの属性については、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場合は、次のようにします。

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、「SCSI 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の属性を設定します。
「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
エンコーディングの使用 (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。
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。
「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」または「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。

表 7-4 SCCS ファイルの属性 (続き)

フラグ	説明
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、105 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。
'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のキーワード値を設定します。
モジュール名 (m 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 s を除いた SCCS ファイル名です。
モジュールの型 (t 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 (v 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時、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 番号が妥当性検査プログラムに渡されます。この検査にパスすると、チェックインが許可されます。

第8章

フリーズポイントの使用

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取り出すことができます。

この章は、次の節で構成されています。

- 119 ページの「フリーズポイントツールの概要」
- 120 ページの「フリーズポイントの仕組み」
- 122 ページの「フリーズポイントの起動」
- 125 ページの「フリーズポイントファイルの作成」
- 126 ページの「フリーズポイントファイルの更新」
- 127 ページの「ファイルの抽出」
- 131 ページの「フリーズポイントの自動作成」
- 133 ページの「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、作業の主だった時点で「フリーズポイント」を作成しておくことで役に立つことがあります。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、主な開発時点でのプロジェクトの状態の復元に使用できます。

プロジェクトの状態を保存する1つの方法として、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ファイル全体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が必要になります。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残すことができます。

この章では、グラフィカルユーザーインターフェース (GUI) 形式のフリーズポイントツールについて説明します。フリーズポイントツールは、コマンド行からも利用できます (第 10 章を参照)。

フリーズポイントの仕組み

フリーズポイントによって、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、フリーズポイントはファイルとディレクトリだけ、すなわち、ファイル履歴のない各ファイルのデルタを取り出します。ただし、フリーズポイントには、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。

フリーズポイントにおける作成

フリーズポイントファイルの作成では、「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。こうして、各デルタごとのファイルと一意の数値識別子の一覧で構成されるフリーズポイントファイルを作成します (133 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デルタのことも、異なるデルタのこともあります。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、フリーズポイントはその 1.2.1.1 を保存します。

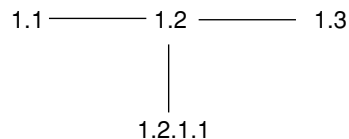


図 8-1 フリーズポイント用デルタの保存

フリーズポイントにおける抽出

フリーズポイントファイルを作成すると、そのファイルを使用してファイルを抽出できます。このために、フリーズポイントファイル名、デルタの抽出元のディレクトリ階層のパス名 (元の抽出元の階層と異なる場合)、抽出先の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するには、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。履歴ファイルそのものは、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かぎり再作成されません。デルタは抽出元のワークスペースにあ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となる、SCCS 履歴ファイルが含まれているディレクトリ階層です。通常、このワークスペースは、階層を再作成するときの抽出元となるディレクトリ階層でもあります。抽出を実行するときに、これとは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記録されているファイルの抽出先となる最上位のディレクトリです。「フリーズポイント」ウィンドウの「抽出先のディレクトリ」フィールドには、この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起動

フリーズポイントを起動するには、次のいずれかの操作を行います。

- Forte for Java の TeamWare ツールバーにある「TeamWare フリーズポイント」ボタン  をクリックする。

注 – デフォルトでは、TeamWare の簡略ツールバーが表示されます。TeamWare ツールバー全体を表示する方法については、159 ページの「Forte for Java の TeamWare ツールバーの表示方法」を参照してください。

- Forte for Java の「エクスプローラ」ウィンドウから、次の操作をする。
 - i. Forte TeamWare のディレクトリをマウスの右ボタンでクリックする。
 - ii. 「TeamWare」▶「フリーズポイント」を選択する。

Forte TeamWare ツールの起動と Forte for Java IDE 内からのコマンドの使用方法についての詳細は、第 11 章を参照してください。

- 「ワークスペース管理」「バージョン管理」「ファイルマージ」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 を参照)。

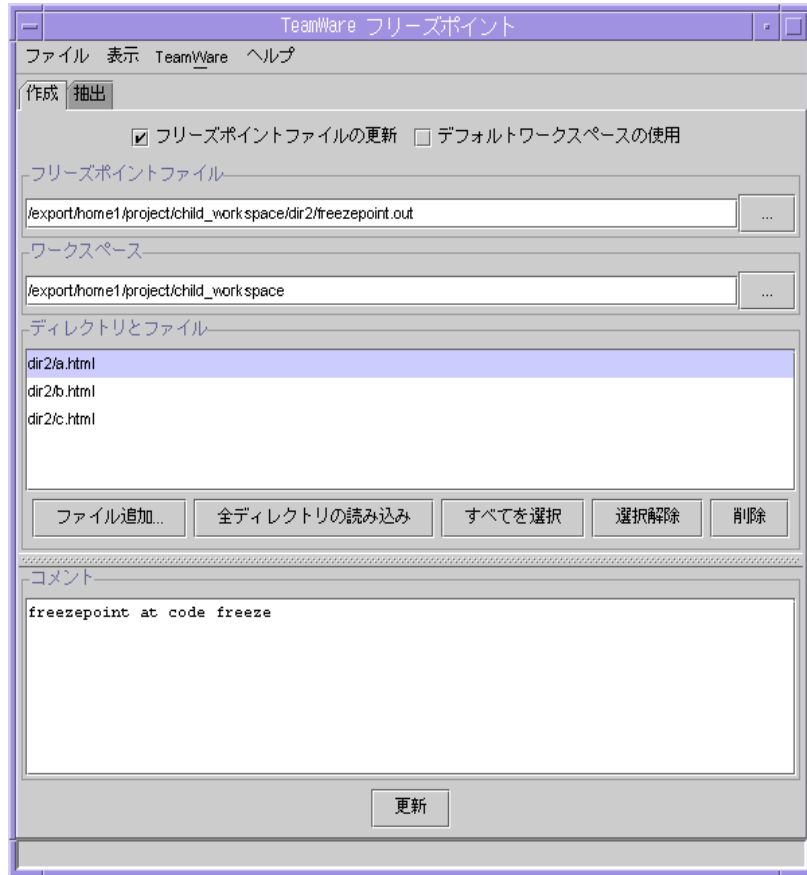


図 8-2 「フリーズポイント」ウィンドウの「作成」タブ

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	フリーズポイントを終了するためのコマンドがあります。
「表示」メニュー	「出力を表示」コマンドがあります。状態メッセージおよび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 ツールを起動するためのコマンドがあります。

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
「ヘルプ」メニュー	ヘルプを表示するための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	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	フリーズポイントファイルに記録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記録を残すファイルおよびディレクトリの一覧です。
「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。
「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を選択解除します。
「削除」ボタン	「ディレクトリとファイル」区画からファイルを削除します。
「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ます。作成に関するメッセージの入った「フリーズポイント出力」ダイアログボックスが表示されます。

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場合は、次のようにします。

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」を選択して、「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の `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` というファイル名が表示されます。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。このデフォルト値の代わりに別のパス名またはファイル名を入力することができます。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を基準にした相対パスとみなされます。

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. フリーズポイント対象に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をフリーズポイント対象にするを意味します。
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場合は、次のようにします。

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」を選択して、「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の `freezepoint.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」区画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。
8. 既存のフリーズポイントファイルを更新する場合は、「フリーズポイントファイルの更新」チェックボックスを選択します。

「デフォルトワークスペースの使用」チェックボックスが選択可能になります。

既存のフリーズポイントファイルに記録されているワークスペースをフリーズポイント対象にするには、「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録されているソース階層を抽出する場合は、次のようにします。

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」を選択して、「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を基準にした相対パスとみなされます。
4. 「すべて抽出」または「部分抽出」ラジオボタンをクリックします。
「すべて抽出」では、フリーズポイントに記録されているファイル群のすべてが抽出されます。「部分抽出」では、指定されたファイルのみ抽出されます。ワークスペースを作成するかどうかに関係なく、すべて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クリックします。

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記録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。

ソースワークスペースとして、フリーズポイントファイルに記録されているのとは異なるワークスペースを指定するには、「ユーザー指定」ボタンを選択して、「ワークスペース」フィールドにその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、抽出した新しい階層を書き込む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を基準にした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または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、確認する必要があります。

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ファイルの一覧が表示されます。一覧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記録されているすべての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-file` 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」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 - ファイルの名前変更や削除などが原因で抽出するファイルが見つからない場合、抽出は中止され、そのファイル名が表示されます。この場合は、フリーズポイントファイルを編集して、見つからなかったファイルの情報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決定する方法については、freezeptfile のマニュアルページを参照してください。

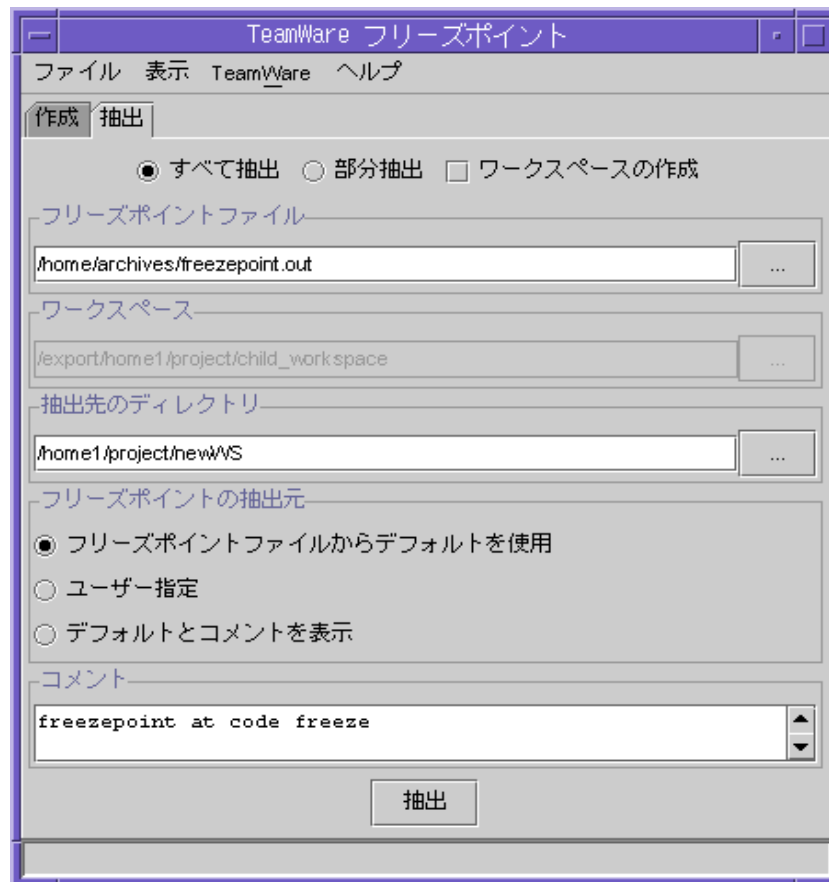


図 8-4 「フリーズポイント」ウィンドウの「抽出」タブ

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	フリーズポイントを終了するためのコマンドがあります。	
「表示」メニュー	「出力を表示」コマンドがあります。	
「TeamWare」メニュー	他の TeamWare ツールを起動するためのコマンドがあります。	
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に記録されている全ファイルを抽出します。	
「部分抽出」ラジオボタン	フリーズポイントファイルに記録されているファイル群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 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	新しい階層を書き込む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抽出元」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に記録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

ブ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場合は、次のようにします。

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。
「ワークスペースの属性」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. フリーズポイントを作成する時期に対応する「はい」ラジオボタンを選択します。複数のボタンを選択することができます。
5. 「了解」をクリックします。

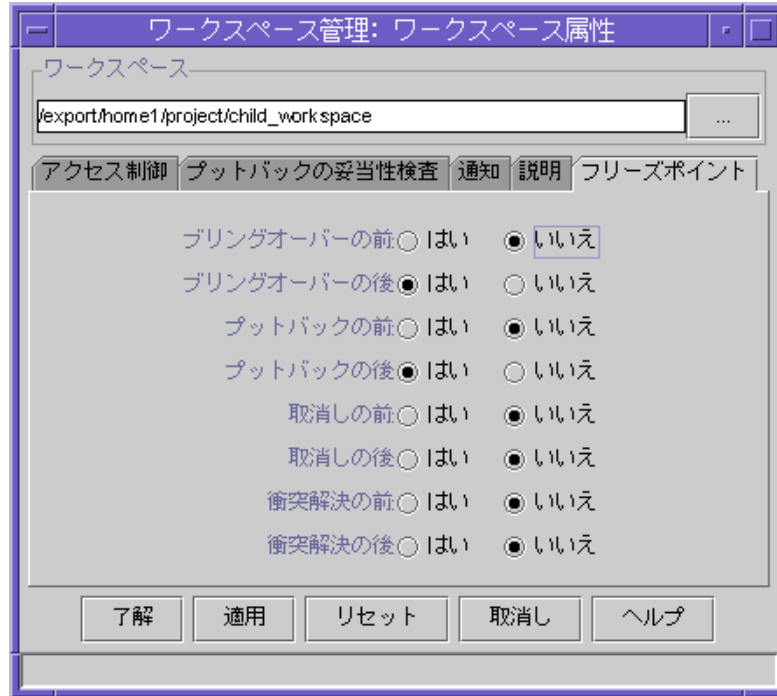


図 8-5 「ワークスペースの属性」ダイアログボックスの「フリーズポイント」タブ

設定に従って、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には、HHMMSS.fp.z (HH は時、MM は分、SS は秒) の形式の名前が付けられます。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す。

フリーズポイントファイルは、フリーズポイント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。フリーズポイントツールは、階層を復元するとき、フリーズポイント内のエントリを、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であ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) ではなく、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の更新プリングオーバーランザクションで SID が変更されたファイルをフリーズポイントで正しく処理できます。詳細は、193 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
- 各ファイルに対応する 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一部内容例を示します。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痴 are the SCCS Mergeable Id of the root delta of the
# containing delta tree.
# Last four hex痴 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

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。

注 – 本リリースでは、dmake ユーティリティは、Solaris プラットフォームでのみ利用できます。

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、136 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、139 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 という名のメー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)/.dmake.rc

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、.dmake.rc 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 9-1 dmake.rc 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` 命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` ユーティリティは、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、144 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方法

`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 stimpy { 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。
 - 同じ名前が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10章

Forte TeamWare のショートカット

Forte 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、次の節で構成されています。

- 149 ページの「コマンド行からの TeamWare の使用方法」
- 153 ページの「GUI のショートカット」

コマンド行からの TeamWare の使用方法

Forte TeamWare のツールは、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 Forte TeamWare のコマンドを入力して利用することもできます。Forte TeamWare のコマンドは、GUI またはコマンド行のどちらからも利用でき、両者の間に機能上の違いはありません。ある機能では GUI、別の機能ではコマンド行を利用して、それらの機能を同時に実行するといったこともできます。スクリプトやプログラムでコマンドを使用して、ファイル管理の自動化をさらに推進することもできます。

Microsoft Windows 環境でコマンド行から Forte TeamWare を使用方法については、244 ページの「コマンド行インターフェースの使用法」を参照してください。

ワークスペース管理関係のコマンド

ワークスペース管理関係のコマンド一覧を表示するには、`codemgr` と入力します。

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

実際には、この一覧には、コマンドオプションを角括弧で囲った、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]
[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
```

Forte TeamWare のツールには、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情報を表示するには、Forte TeamWare のオンラインヘルプから利用可能な、そのコマンドのマニュアルページを参照してください。

注 – Forte TeamWare では、`man` コマンドを使用してマニュアルページにアクセスすることはできません。Forte TeamWare のマニュアルページは、オンラインヘルプシステムから利用できます。Forte TeamWare のマニュアルページにアクセスするには、Forte TeamWare オンラインヘルプシステムの目次から「コマンドリファレンス」を選択してください。xvi ページの「マニュアルページのアクセス方法」を参照してください。

表 10-1 は、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ています。

表 10-1 「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>
元に戻す	<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。52 ページの「環境変数の設定」を参照してください。

バージョン管理関係のコマンド

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 &
```

ファイルマージ関係のコマンド

入力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で `twmerge` に続けてアンパサンド (&) を入力します (ファイルマージ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、152 ページの「起動時に 2 つのファイルを読み込む場合」を参照してください。

以下は `twmerge` コマンドのコマンド行構文で、角括弧に囲まれているのがコマンドオプションです。

```
twmerge [-b] [-r] [-tabsize 値] [-diffopt [bwi]]
[-a 祖先] [-f1 名前 1] [-f2 名前 2]
[-l 左のファイル] [ 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場合

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 つのファイルをマージするには、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場合

2 つのファイルを読み込むのと同時に、それらファイルを `ancestor_file` という名前の共通の祖先ファイルと比較するには、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。

フリーズポイント関連のコマンド

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。フリーズポイント関係のコマンド一覧を表示するには、`freezept` と入力します。.

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

GUI のショートカット

Forte 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。これらショートカットは、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、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。ボタンを離し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 10-2 は、Forte 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 10-2 マウスおよびキーボードショートカット

操作	働き	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」タブが表示されます。	
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」タブが表示されます。	
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」タブが表示されます。	
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が表示されます。	
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	親が変更されます。	43 ページの「ワークスペースの親の変更」
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	ワークスペースが親から切り離されます。	43 ページの「ワークスペースの親の変更」
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できます。	
「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」メニューを合わせたメニューが表示されます。	
「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	ツールが起動されます。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です。	155 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」

表 10-2 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
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	ツールが起動されます。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」ウィンドウです。	155 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」
「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	ファイルがチェックアウトされるか、ファイルの履歴が表示されます (どちらにするかはユーザーが設定可能)。	155 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」
「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デルタが表示されます。	155 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」
「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合わせたメニューが表示されます。	
「バージョン管理」の「履歴」ウィンドウ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」を合わせたメニューが表示されます。	102 ページの「ファイルの履歴の表示」
「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」メニューが表示されます。	89 ページの「相違の解決」

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、Forte TeamWare のバージョン管理ツールが自動的に起動され、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、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るため、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での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、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。115 ページの「バージョン管理の属性の設定」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「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のウィンドウにそのデルタが表示されます。

第11章

Forte for Java IDE での Forte TeamWare の使用

本リリースの Forte TeamWare は、Forte for Java 統合開発環境 (IDE) に組み込むことができる統合モジュールです。この章では、Forte TeamWare 統合モジュールと Forte for Java IDE との関係、および Forte for Java で Forte TeamWare コマンドを使用する方法を説明します。

この章は、次の節で構成されています。

- 157 ページの「Forte TeamWare と Forte for Java の関係」
 - 161 ページの「Forte TeamWare コマンドの使用方法和機能」
-

Forte TeamWare と Forte for Java の関係

Forte TeamWare は、Forte for Java 環境に組み込むことができる統合モジュールです。したがって、Forte TeamWare を Forte for Java 内で動作させることができます。Forte for Java の GUI により、Forte TeamWare ツールおよびコマンドを直接使用することができます。

Forte TeamWare は、Forte for Java IDE を必要としないスタンドアロンアプリケーションとして使用することもできます。

Forte for Java における Forte TeamWare の機能

Forte for Java IDE 内では、Forte TeamWare の機能は 2 つの異なるレベルで実装されています。

- Forte for Java の GUI から、バージョン管理ツールおよびワークスペース管理ツールのさまざまなコマンドを直接使用できます。
- その他のコマンドと機能はすべて、Forte TeamWare の GUI ツールから使用できます。これらのツールは、Forte for Java の GUI から簡単に起動できます。

Forte TeamWare コマンドは、Forte for Java IDE の次の場所から使用できます。

- コンテキストメニュー（マウスの右ボタンをクリックすることによって表示）
- TeamWare ツールバー
- メインウィンドウの「バージョン管理」メニュー

Forte TeamWare ファイルシステムのマウント

Forte for Java の「エクスプローラ」ウィンドウから Forte TeamWare ワークスペースにアクセスするには、まず、次のいずれかの操作によって、そのワークスペースをマウントしておく必要があります。

- Forte for Java のメインウィンドウから、次の操作をする。
 - i. 「ファイル」▶「ファイルシステムをマウント」を選択します。
 - ii. ダイアログで「追加 (他のシステムタイプ)」を選択します。
 - iii. 「種類」として「TeamWare Version Control System」を選択します。
 - i. 「ブラウズ」をクリックして既存のワークスペースへのパスを指定します。
- Forte for Java のメインウィンドウから、次の操作をする。
 - i. 「バージョン管理」▶「TeamWare Version Control System をマウント」を選択して既存の Forte TeamWare ワークスペースを読み込みか、「バージョン管理」▶「TeamWare 新ワークスペース」を選択して新規の Forte TeamWare ワークスペースを作成します。

- ii. 「既存の TeamWare ファイルシステムをマウント」ダイアログでワークスペースへのパスを入力して「マウント」をクリックするか、「TeamWare 新ワークスペース」ダイアログで新規のワークスペースへのパスを入力して「作成」をクリックします。
- Forte for Java の「エクスプローラ」ウィンドウから、次の操作をする。
 - i. 「ファイルシステム」上でマウスの右ボタンをクリックします。
 - ii. 「TeamWare Version Control System をマウント」を選択します。
 - ワークスペース管理ツールから、次の操作をする。
 - i. 「ファイル」▶「ワークスペースの読み込み」を選択して既存の Forte TeamWare ワークスペースを読み込みか、「ファイル」▶「ワークスペース作成」を選択して新規の Forte TeamWare ワークスペースを作成します。
 - ii. 「ワークスペース管理」ウィンドウで、ワークスペースを選択して「ワークスペース」▶「エクスプローラにマウント」を選択するか、ワークスペースをマウスの右ボタンをクリックして「エクスプローラにマウント」を選択します。

Forte for Java の TeamWare ツールバーの表示方法

Forte TeamWare モジュールを Forte for Java IDE とともにインストールしている場合、最初に Forte for Java IDE を起動したときには、デフォルトでは、Forte TeamWare 簡略ツールバーが表示されます。このツールバーには、「TeamWare ワークスペース管理」アイコンと「TeamWare 新ワークスペース」アイコンだけが表示されます。Forte TeamWare のツールバー全体を表示するには以下のようにします。

1. Forte for Java のツールバーの任意の場所でマウスの右ボタンをクリックします (ボタンの上ではクリックしないでください)。
2. 表示されたコンテキストメニューで「TeamWare (Full)」を選択します。

注 – コンテキストメニューで「TeamWare」を選択解除すると、簡略ツールバーが表示されなくなります。

Forte for Java での Forte TeamWare の実行

図 11-1 に、Forte TeamWare 統合モジュールが組み込まれた Forte for Java の画面を示します。Forte for Java GUI のウィンドウに、Forte TeamWare コマンドと機能へのアクセス手段が追加されています。図の例では、Forte TeamWare ツールバー全体が表示されています。

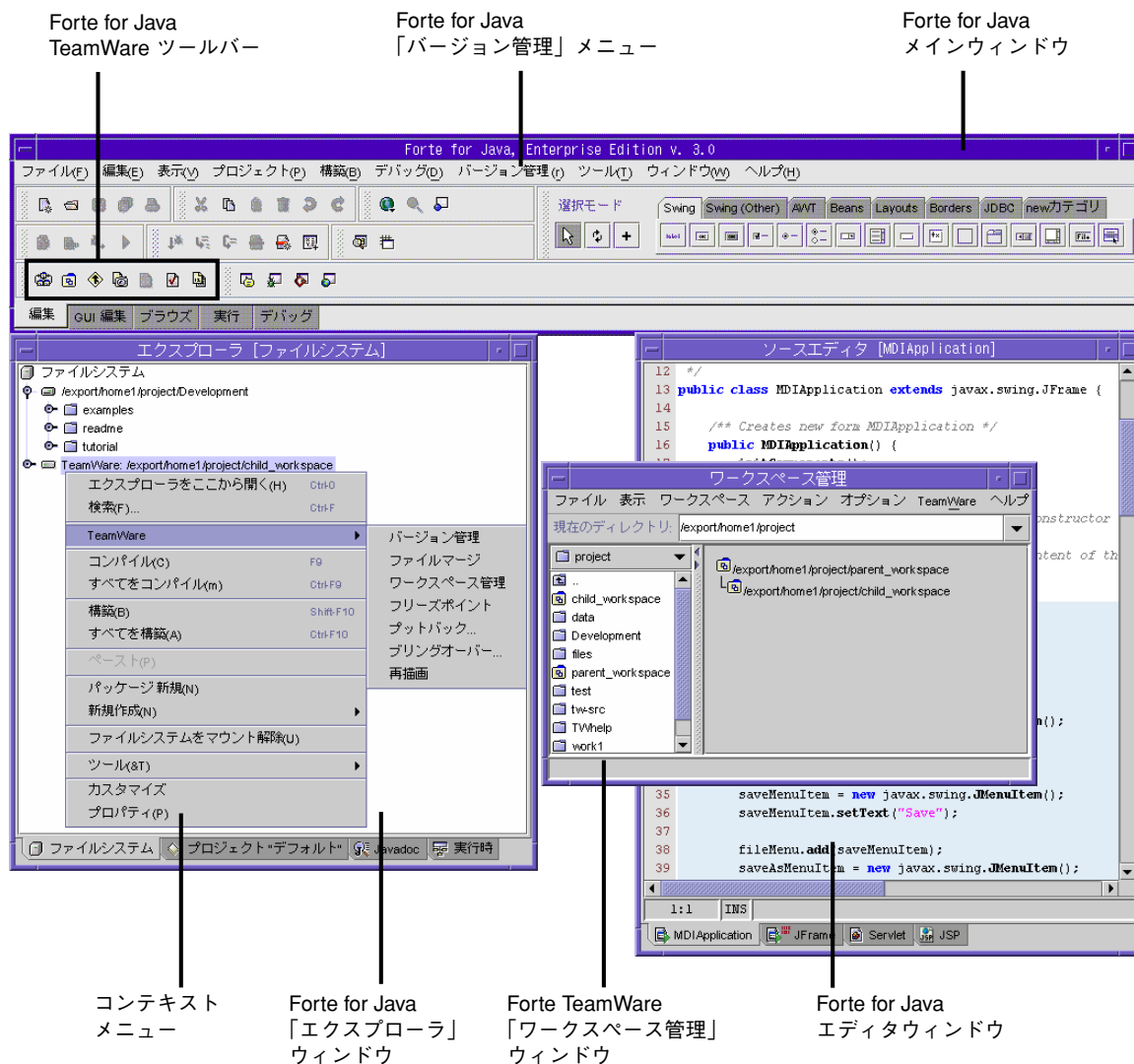


図 11-1 Forte TeamWare 統合モジュールが組み込まれた Forte for Java

Forte TeamWare コマンドの使用方法和機能

この節では、Forte for Java GUI から使用できる Forte TeamWare コマンドとその機能について説明します。詳細は、このマニュアルの対応する章と Forte TeamWare のオンラインヘルプを参照してください。

注 – ここで示す Forte TeamWare のコマンドや機能を使用するには、Forte TeamWare ツールバー全体を使用できることが前提になります。デフォルトでは、簡略ツールバーが表示されます。詳細は、159 ページの「Forte for Java の TeamWare ツールバーの表示方法」を参照してください。


Forte TeamWare ツールと機能

表 11-1 に、Forte for Java で使用可能な Forte TeamWare ツールとその主な機能、およびアクセス方法を示します。

表 11-1 Forte TeamWare ツールと機能、アクセス方法

ワークスペース管理 (ワークスペース管理ツールについての詳細は、第 2 章を参照)


- Forte TeamWare のワークスペース管理ツールを開く

Forte for Java の TeamWare ツールバー	「TeamWare ワークスペース管理」ボタンをクリック	
---------------------------------	------------------------------	---

Forte for Java の エクスプローラ	1. Forte TeamWare にアクセス可能なディレクトリをマウスの右ボタンでクリック 2. 「TeamWare」▶「ワークスペース管理」を選択	
--------------------------	--	--

- Forte TeamWare ワークスペースの新規作成

Forte for Java の メニューバー	「バージョン管理」▶「TeamWare 新 ワークスペース」を選択	
-------------------------	-----------------------------------	--

Forte for Java の TeamWare ツールバー	「TeamWare 新 ワークスペース」ボタンをクリック	
---------------------------------	------------------------------	---

ワークスペース管理ツール	「ファイル」▶「ワークスペース作成」を選択	
--------------	-----------------------	--

表 11-1 Forte TeamWare ツールと機能、アクセス方法 (続き)

<ul style="list-style-type: none"> 既存の Forte TeamWare ワークスペースの読み込み 		
Forte for Java の TeamWare ツールバー	<ol style="list-style-type: none"> 「TeamWare ワークスペース管理」ボタンをクリック 「ワークスペース管理」ウィンドウで「ファイル」▶「ワークスペースの読み込み」を選択 	
Forte for Java の エクスプローラ	<ol style="list-style-type: none"> 「ファイルシステム」をマウスの右ボタンでクリック 「TeamWare Version Control System をマウント」を選択 「既存の TeamWare ファイルシステムをマウント」ダイアログでワークスペースのフルパス名を入力 	
ワークスペース管理ツール	「ファイル」▶「ワークスペースの読み込み」を選択	
<ul style="list-style-type: none"> プットバック 		
Forte for Java の エクスプローラ	<ol style="list-style-type: none"> Forte TeamWare のファイルシステムまたはディレクトリをマウスの右ボタンでクリック 「TeamWare」▶「プットバック」を選択 	
Forte for Java の メニューバー	Forte for Java の「エクスプローラ」ウィンドウで Forte TeamWare のファイルシステムまたはディレクトリを選択し、 「バージョン管理」▶「TeamWare」▶ 「プットバック」を選択	
ワークスペース管理ツール	「アクション」▶「プットバック」を選択	


表 11-1 Forte TeamWare ツールと機能、アクセス方法 (続き)

<ul style="list-style-type: none"> • プリンティングオーバー 	
Forte for Java の エクスプローラ	<ol style="list-style-type: none"> 1. Forte TeamWare のディレクトリをマウスの右ボタンでクリック 2. 「TeamWare」▶「プリンティングオーバー」を選択
Forte for Java の メニューバー	Forte for Java の「エクスプローラ」ウィンドウで Forte TeamWare のファイルシステムまたはディレクトリを選択し、 「バージョン管理」▶「TeamWare」▶「プリンティングオーバー」を選択
ワークスペース管理ツール	「アクション」▶「作成プリンティングオーバー」を選択するか、 「アクション」▶「更新プリンティングオーバー」を選択
<p>ファイルマージ (ファイルマージツールについての詳細は、第 6 章を参照)</p> <ul style="list-style-type: none"> • Forte TeamWare のファイルマージツールを開く 	
Forte for Java の TeamWare ツールバー	「TeamWare ファイルマージ」ボタンをクリック 
Forte for Java の エクスプローラ	Forte TeamWare ファイルまたはディレクトリをマウスの右ボタンでクリック 「TeamWare」▶「ファイルマージ」を選択
Forte for Java の メニューバー	Forte for Java の「エクスプローラ」ウィンドウで Forte TeamWare のファイル、ファイルシステムまたはディレクトリを選択し、「バージョン管理」▶「TeamWare」▶「ファイルマージ」を選択

表 11-1 Forte TeamWare ツールと機能、アクセス方法 (続き)

フリーズポイント (フリーズポイントツールについての詳細は、第 8 章を参照)

- Forte TeamWare のフリーズポイントツールを開く

Forte for Java の TeamWare ツールバー	「TeamWare フリーズポイント」ボタンをクリック	
Forte for Java の エクスプローラ	Forte TeamWare ファイルまたはディレクトリをマウスの右ボタンでクリック 「TeamWare」▶「フリーズポイント」を選択	
Forte for Java の メニューバー	Forte for Java の「エクスプローラ」ウィンドウで Forte TeamWare のファイルシステムまたはディレクトリを選択し、 「バージョン管理」▶「TeamWare」▶「フリーズポイント」を選択	

バージョン管理 (バージョン管理ツールについての詳細は、第 5 章を参照)

- Forte TeamWare のバージョン管理ツールを開く

Forte for Java の TeamWare ツールバー	「TeamWare バージョン管理」ボタンをクリック	
Forte for Java の エクスプローラ	1. Forte TeamWare のディレクトリをマウスの右ボタンでクリック 2. 「TeamWare」▶「バージョン管理」を選択	
Forte for Java の メニューバー	Forte for Java の「エクスプローラ」ウィンドウで Forte TeamWare のファイル、ファイルシステム、ディレクトリのいずれかを選択し、 「バージョン管理」▶「TeamWare」▶「バージョン管理」を選択	

表 11-1 Forte TeamWare ツールと機能、アクセス方法 (続き)

<ul style="list-style-type: none"> • チェックアウト/チェックアウトして編集 	
Forte for Java の エクスプローラ	<ol style="list-style-type: none"> 1. チェックインされているファイルをマウスの右ボタンでクリック 2. 「チェックアウト」または「チェックアウトして編集」を選択
Forte for Java の TeamWare ツールバー	「チェックアウト」ボタンをクリック 
Forte for Java の メニューバー	マウント済の Forte TeamWare ファイルシステムに含まれているチェックインされているファイルを選択し、「バージョン管理」▶「チェックアウト」、または「バージョン管理」▶「チェックアウトして編集」を選択
バージョン管理ツール	「コマンド」▶「チェックアウト」を選択するか、 「コマンド」▶「チェックアウトして編集」を選択
<ul style="list-style-type: none"> • チェックイン 	
Forte for Java の エクスプローラ	<ol style="list-style-type: none"> 1. チェックアウトされているファイルをマウスの右ボタンでクリック 2. 「チェックイン」を選択
Forte for Java の TeamWare ツールバー	「チェックイン」ボタンをクリック 
Forte for Java の メニューバー	マウント済の Forte TeamWare ファイルシステムに含まれている、チェックアウトされているファイルか新規のファイルを選択し、「バージョン管理」▶「チェックイン」を選択
バージョン管理ツール	「コマンド」▶「チェックイン」を選択

表 11-1 Forte TeamWare ツールと機能、アクセス方法 (続き)

• アンチェックアウト	
Forte for Java の エクスプローラ	1. チェックアウトされているファイルをマ ウスの右ボタンでクリック 2. 「アンチェックアウト」を選択
Forte for Java の メニューバー	マウント済の Forte TeamWare ファイルシ ステムに含まれているチェックアウトされ ているファイルを選択し、「バージョン管 理」▶「アンチェックアウト」を選択
バージョン管理ツール	「コマンド」▶「アンチェックアウト」を 選択
• ファイル履歴	
Forte for Java の エクスプローラ	1. Forte TeamWare のファイルをマウスの 右ボタンでクリック 2. 「TeamWare」▶「ファイル管理」を選 択
バージョン管理ツール	「ファイル」▶「ファイル履歴」を選択

GUI と Forte TeamWare コマンド

表 11-2 に、Forte for Java の GUI と Forte for Java IDE から使用可能な Forte TeamWare コマンドを示します。

表 11-2 Forte for Java の GUI と割り当てられている Forte TeamWare コマンド

GUI 要素	説明
TeamWare ツールバー	
 チェックアウト (チェックインされた Forte TeamWare ファイルがエクスプローラで選択されている場合にのみ選択可能)	Forte TeamWare ファイルをチェックアウト
 チェックイン (チェックアウトされた Forte TeamWare ファイルがエクスプローラで選択されている場合にのみ選択可能)	Forte TeamWare ファイルをチェックイン
 TeamWare 新ワークスペース	Forte TeamWare ワークスペースを新規作成
 TeamWare ワークスペース管理	Forte TeamWare のワークスペース管理ツールを開く
 TeamWare バージョン管理	Forte TeamWare のバージョン管理ツールを開く
 TeamWare ファイルマージ	Forte TeamWare のファイルマージツールを開く
 TeamWare フリーズポイント	Forte TeamWare のフリーズポイントツールを開く
メニューバー	
「バージョン管理」メニュー	
TeamWare 新ワークスペース	Forte TeamWare ワークスペースを新規作成
TeamWare Version Control System をマウント	既存の Forte TeamWare ファイルシステムをマウントし、「エクスプローラ」ウィンドウからアクセスできるようにする

表 11-2 Forte for Java の GUI と割り当てられている Forte TeamWare コマンド (続き)

GUI 要素	説明
「エクスプローラ」ウィンドウ	
コンテキストメニュー (Forte TeamWare のファイルを選択している場合に使用可能)	
注: これらのコンテキストメニューの項目は、Forte for Java のメニューバーにある「バージョン管理」メニューから利用することもできます。	
「TeamWare」 ▶ 「バージョン管理」	Forte TeamWare のバージョン管理ツールを開く
「TeamWare」 ▶ 「ファイルマージ」	Forte TeamWare のファイルマージツールを開く
「TeamWare」 ▶ 「ファイル履歴」	Forte TeamWare のバージョン管理ツールの「履歴」ダイアログボックスを開く バージョン管理ツールの「ファイル」 ▶ 「ファイル履歴」と同じ
「TeamWare」 ▶ 「再描画」	Forte TeamWare のディレクトリ一覧を更新
「TeamWare」 ▶ 「移動」	バージョン管理ツールの「移動の確認」ダイアログを開く ファイルのコピーと履歴ファイルを指定した場所に移動する
<ul style="list-style-type: none"> 注: ファイルの移動は、Forte TeamWare のソース管理下にあるファイルだけが対象となります。また、同一のワークスペース内への移動のみが可能です。ファイルに設定されていたアクセス権は、移動後も保持されます。新しいパッケージディレクトリの場所を反映するように、Java ファイル内部のパッケージ文が自動的に更新されることはありません。この操作を、ファイルの名前変更を行う目的で使用することはできません。 	

表 11-2 Forte for Java の GUI と割り当てられている Forte TeamWare コマンド (続き)

GUI 要素	説明
<p>「TeamWare」 ▶ 「削除」</p>	<p>選択したファイルの現在の状態が「チェックイン」の場合、そのファイルを Forte TeamWare の deleted_files ディレクトリに移動する</p> <ul style="list-style-type: none"> • ファイルの現在の状態が「チェックアウト」の場合、「チェックインまたはアンチェックアウトの確認」ダイアログが表示されます。「チェックイン」「アンチェックアウト」「取消し」のいずれかを選択する必要があります。 • 注: この操作は、Forte TeamWare のソース管理下にあるファイルだけが対象となります。
<p>「TeamWare」 ▶ 「デフォルトバージョンの取得」</p>	<ul style="list-style-type: none"> • ファイルの状態が「チェックイン」で読み取り専用の場合にこの操作を行うと、ファイルの最新のデフォルトバージョンが取得され、「エクスプローラ」ウィンドウに読み込まれている現在のコピーが置き換えられます。 • ファイルの状態が「チェックアウト」で書き込み可能の場合にこの操作を行うと、ファイルの最新のデフォルトバージョンは取得されません。
<p>チェックイン (チェックアウトした Forte TeamWare ファイルを選択しているか、ファイルが新規で「チェックインが必要」とマークされている場合にのみ選択可能)</p>	<p>Forte TeamWare のファイルをチェックイン</p>
<p>チェックアウト (チェックインされている Forte TeamWare ファイルを選択している場合にのみ選択可能)</p>	<p>Forte TeamWare のファイルをチェックアウト</p>

表 11-2 Forte for Java の GUI と割り当てられている Forte TeamWare コマンド (続き)

GUI 要素	説明
チェックアウトして編集 (チェックインされている Forte TeamWare ファイルを選択している場合にのみ選択可能)	Forte TeamWare のファイルをチェックアウトして Forte for Java IDE エディタで開く (注: 使用されるエディタは、ファイルの種類によって決まります。詳細は、172 ページの「Forte for Java IDE エディタと Forte TeamWare ツール」を参照してください)。
アンチェックアウト	Forte TeamWare のファイルに対する変更を取り消す
カット / ペースト	<ul style="list-style-type: none"> • ペースト先が、同一の Forte TeamWare ワークスペース内の場合、ファイルの現在のコピーと履歴が指定したディレクトリに移動されます。新しいパッケージディレクトリの場所を反映するように、Java ファイル内部のパッケージ文が自動的に更新されます。 • ペースト先が、現在の Forte TeamWare ワークスペース内ではない場合、選択したファイルの現在のコピーが指定した場所に移動され、対応する履歴は現在のワークスペースの deleted_files ディレクトリに移動されます。新しいパッケージディレクトリの場所を反映するように、Java ファイル内部のパッケージ文が自動的に更新されます。 <p>注: この操作の代わりに、「TeamWare」▶「移動」を行うことを強く推奨します。</p>
コピー / ペースト	ファイルの現在のコピーだけを指定したディレクトリにコピーし、履歴はコピーしない

表 11-2 Forte for Java の GUI と割り当てられている Forte TeamWare コマンド (続き)

GUI 要素	説明
削除	Forte TeamWare の管理下のファイルを deleted_files ディレクトリに移動する Forte TeamWare の管理下ではないファイルの場合、そのファイルはファイルシステムから削除される
名前を変更	ファイルの現在のコピーの名前と SCCS ディレクトリにあるその履歴ファイルの名前を変更する
コンテキストメニュー (Forte TeamWare の管理下にあるファイルシステムやディレクトリを選択している場合に使用可能)	
注: これらのコンテキストメニューの項目は、Forte for Java のメニューバーにある「バージョン管理」メニューから利用することもできます。	
「TeamWare」▶「バージョン管理」	Forte TeamWare のバージョン管理ツールを開く
「TeamWare」▶「ファイルマージ」	Forte TeamWare のファイルマージツールを開く
「TeamWare」▶「ワークスペース管理」	Forte TeamWare のワークスペース管理ツールを開く
「TeamWare」▶「フリーズポイント」	Forte TeamWare のフリーズポイントツールを開く
「TeamWare」▶「プットバック」	選択されているディレクトリをプットバック
「TeamWare」▶「ブリングオーバー」	選択されているディレクトリをブリングオーバー
「TeamWare」▶「更新」	選択したディレクトリ内の各ファイルに対して「デフォルトバージョンの取得」を行う
「TeamWare」▶「更新 (再帰的に)」	選択したディレクトリ内の各ファイルに対して再帰的に「デフォルトバージョンの取得」を行う

表 11-2 Forte for Java の GUI と割り当てられている Forte TeamWare コマンド (続き)

GUI 要素	説明
「TeamWare」 ▶ 「移動」	バージョン管理ツールの「移動の確認」ダイアログを開き、ファイルのコピーと履歴ファイルを指定した場所に移動する <ul style="list-style-type: none"> 注: ファイルの移動は、Forte TeamWare のソース管理下にあるファイルだけが対象となります。また、同一のワークスペース内への移動のみが可能です。ファイルに設定されていたアクセス権は、移動後も保持されます。新しいパッケージディレクトリの場所を反映するように、Java ファイル内部のパッケージ文が自動的に更新されることはありません。この操作を、ファイルの名前変更を行う目的で使用することはできません。
「TeamWare」 ▶ 「削除」	<ul style="list-style-type: none"> ディレクトリを <code>deleted_files</code> ディレクトリに移動する <p>注: この操作は、Forte TeamWare のソース管理下にあるファイルだけが対象となります。</p>
「TeamWare」 ▶ 「再描画」	Forte TeamWare のディレクトリ一覧を更新

Forte for Java IDE エディタと Forte TeamWare ツール

Forte TeamWare で以下の操作を行う場合は、ソースファイルや一時ファイルを扱うためにファイルエディタを使用します。

■ Forte TeamWare の「バージョン管理」ウィンドウから

- 「ファイル」 ▶ 「新規」
- 「ファイル」 ▶ 「表示」
- 「ファイル」 ▶ 「編集」 (チェックアウトされているファイルに対して)
- 「コマンド」 ▶ 「チェックアウトして編集」

上記の操作を選択すると、Forte for Java の IDE ソースエディタが開きます。使用されるソースエディタの種類は、ファイルの種類によって決まります。

次の表に、Forte TeamWare バージョン管理ツールが行うファイル操作によって使用されるファイルエディタを示します。

表 11-3 選択したファイルのマウント状態によって使用されるファイルエディタ

選択したファイルのマウント状態	Forte TeamWare バージョン管理ツールが行うファイル操作
Forte for Java エクスプローラでマウントされている	ファイルの種類に対応した Forte for Java エディタウィンドウが開く
IDE エクスプローラではマウントされていないが Forte TeamWare ワークスペースの一部	<ul style="list-style-type: none"> • ファイルの種類が .java ではない場合、Forte for Java エディタウィンドウが開く。使用されるソースエディタの種類は、ファイルの種類によって決まる。 • ファイルの種類が .java の場合、ファイルが属する Java パッケージを決定するための試みが行われる。ユーザーは、候補として表示されたパッケージを確認してそれに決定するか、パッケージを自ら選択する必要がある。選択したマウントポイントは、Forte for Java エクスプローラウィンドウに Forte TeamWare ワークスペースとしてマウントされる。Forte for Java エディタウィンドウが開く。
IDE エクスプローラでマウントされておらず、Forte TeamWare ワークスペースの一部でもないが、バージョン管理下にある	Forte TeamWare バージョン管理ツールの「表示」▶「オプション」ダイアログで設定されているデフォルトのエディタが開く

- Forte TeamWare の「履歴」ウィンドウから
 - 「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」
 - 「表示」▶「ファイルの内容を表示」

上記の操作を選択すると、一時ファイルが作成され、そのファイルが Forte TeamWare Viewer [一時ファイル名] ウィンドウに表示されます。一時ファイルは、Forte for Java エクスプローラウィンドウにファイルシステムとしてマウントされません。

第12章

Forte TeamWare のアーキテクチャー

この章では、Forte TeamWare がワークスペースとファイルの追跡に使用する、その基礎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のワークスペース管理による SCCS 履歴ファイルの操作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。

- 175 ページの「ワークスペースのメタデータディレクトリ」
- 178 ページの「ワークスペース管理のデフォルト値ファイル」
- 180 ページの「ワークスペース管理によるファイルのマージ」
- 193 ページの「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 という名前のディレクトリを含むディレクトリ階層です。ワークスペース管理は、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には、Codemgr_wsdata ディレクトリ内の情報を管理するためのツールが用意されています。推奨できる方法ではありませんが、ワークスペース管理関係のファイルは手作業で編集することもできます。ただし、その場合は、編集する各ファイルの形式を壊さないよう最新の注意を払ってください。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	特定のワークスペースに対するユーザーのワークスペース管理トランザクションおよびコマンドの実行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 179 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 引数のリストが含まれています。ワークスペース管理のプリングオーバーおよびプ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。当初、このファイルには、ワークスペースの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プットバックトランザクションで引数を指定すると、args ファイル内の既存の引数よりも新しい引数の適用範囲の方が広いかが判断され、適用範囲が広い場合は、古い引数が新しい引数に置き換えられます。39 ページの「独自のプリングオーバーおよびプ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がプリングオーバーやプ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が保存されています。27 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、その親の 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。
description	ユーザーがワークスペースに付けた名前と説明が含まれるファイルです。42 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。

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
freezepoints/	自動フリーズポイントを設定したときに作成されるファイルです。131 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。30 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、プットバック、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は、ワークスペース内での処理を開始する前に、このファイルを参照します。67 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 進ワードで表される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ます。ワークスペース管理は、プリングオーバーおよびプ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。このファイルが使用できない場合、ワークスペース管理は次の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で自動的にファイルを作成します。107 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
notification	ワークスペース管理は、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。41 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。
parent	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。このファイルは、作成プリングオーバートランザクション、またはワークスペースが reparent コマンドで作成されたために、親がない場合に reparent コマンドによって作成されます。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特定します。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、その子の parent ファイルが更新されます。
putback.cmt	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。ワークスペース管理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の動作の設定を変更して、「適用」ボタンをクリックすると、新しい設定がそのユーザーのホームディレクトリの実行時構成ファイルに保存されます。ワークスペース管理は、起動されたときにその実行時構成ファイルを参照し、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まれるワークスペースを制御できます (52 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 (39 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するユーザーのトランザクション実行制御の概念を説明しました。「アクセス制御」タブ (「ワークスペース」▶「属性」で表示) 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

操作	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 は、ワークスペースに対するアクセスの制御に指定できる値の形式例とその意味をまとめています。

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 engineering 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 engineering 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 special および engineering 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、user2 は実行できません (user2 が special ネットグループに属してない場合)。~ は否定を意味します。
user1 user2	ユーザー user1 と user2 は、このトランザクションを実行できます。
-	誰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)	誰でも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によるファイルのマージ

この節を理解するには、分岐の概念などの SCCS に関する知識が前提になります。分岐については、105 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プットバックトランザクションにおける転送元 (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は、そのファイルの SCCS 履歴ファイル (s ドットファイル) を操作する必要があります。

ブ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)を、転送先のSCCS履歴ファイルにマージする必要があります。転送元の情報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プ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はデルタ履歴の分岐場所を特定し、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ます。

図 12-1 に、プ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。

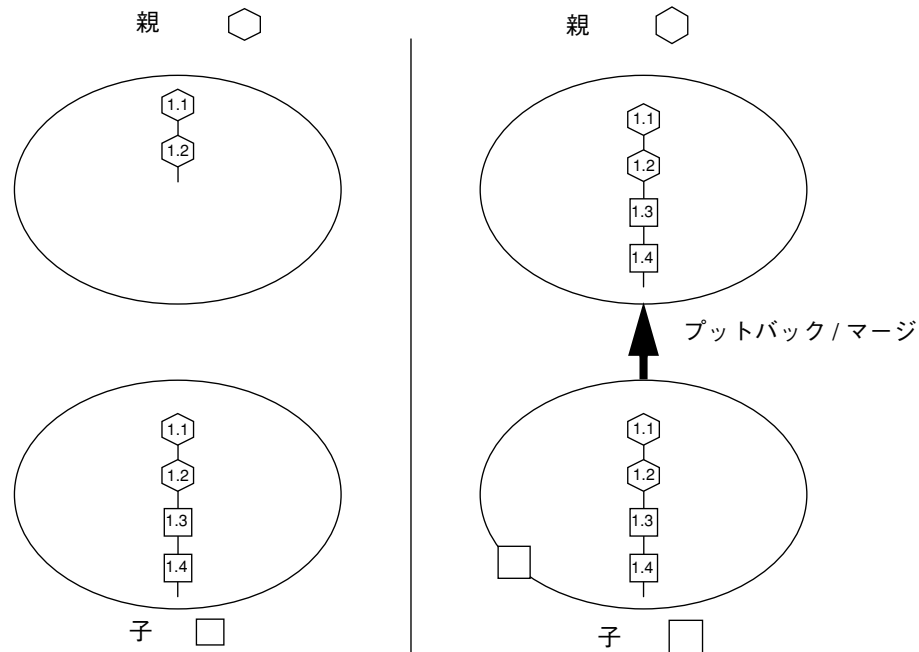


図 12-1 転送先ワークスペース内の変更されていないファイルの更新

衝突しているファイルのマージ

親と子ワークスペース間のファイルのやりとりでは、最後に更新されてから、同じファイルの、どちらのワークスペース内のバージョンも変更されていることがあります。これは、ファイルの親バージョンと子バージョンが衝突している状態です。

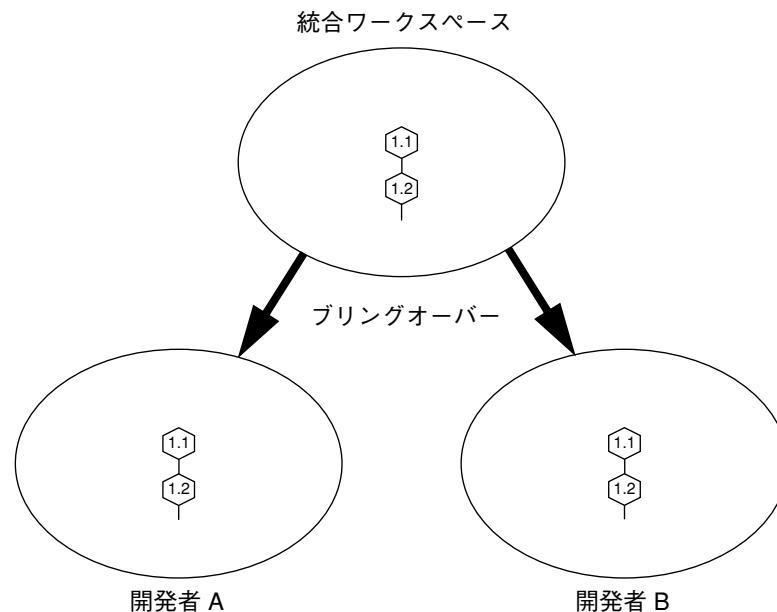
ファイルが衝突している場合、ワークスペース管理はファイルに加えられた変更の矛盾をユーザーが効率的に解決できるように支援し、そのデルタ、管理情報、コメント履歴を保持します。ワークスペース管理は、親の SCCS デルタを子の履歴ファイルにマージします。次に、衝突解決トランザクションによって、子の側でこの衝突を解決します。衝突解決についての詳細は、89 ページの「相違の解決」を参照してください。

ファイルマージによるデルタの追跡

この節では、1つの統合ワークスペースと2つの子ワークスペースが関係するファイルマージ例を紹介します。この2つのワークスペースは、それぞれ異なる開発者によって所有されています。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。ここでは、衝突が発生してから、その衝突が解決されるまでのSCCS履歴ファイルの変化を図で表しています。図については、次の点に注意してください。

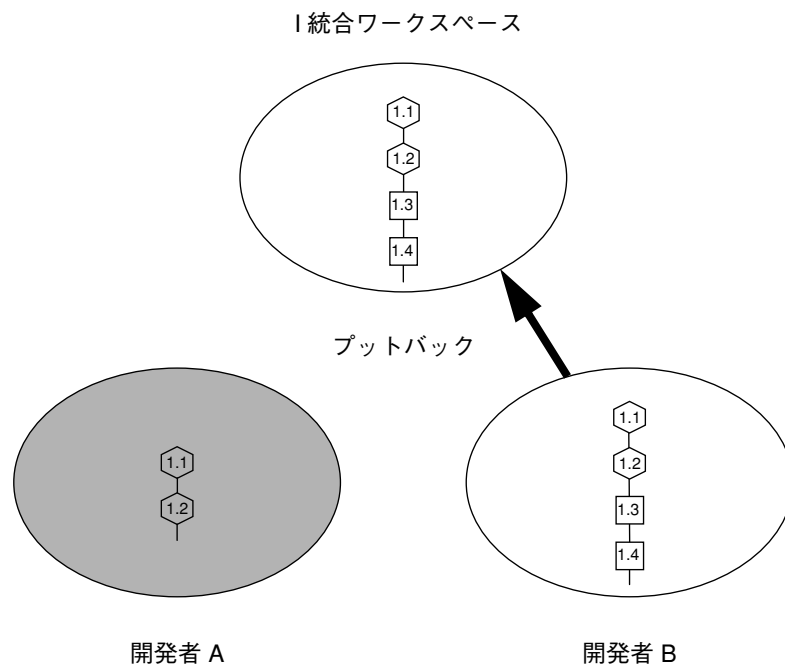
- デフォルトのデルタ (SCCS デルタツリーへの次のデルタの追加先) は、接続されていない下向きの線で表しています。
- バージョン管理では、ここで示しているのと同様の形式でSCCSデルタツリーをグラフィカルに表示できます。

開発者はそれぞれ、プリングオーバートランザ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新規のファイルであるため、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。



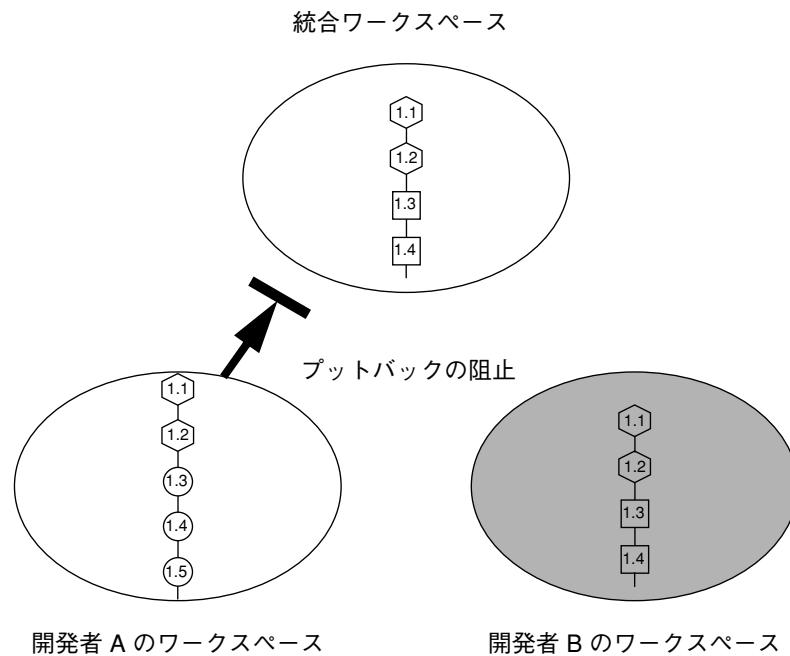
開発者 B は、このファイルに変更を加えて、2つの新しいデルタ 1.3 と 1.4 を作成した後、ファイルを統合ワークスペースに戻します (プットバックトランザクション)。このときワークスペース管理は、2つの新しいデルタを親の SCCS デルタツリーに追加します。

アクセスリストなどの管理情報を保持するために、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。



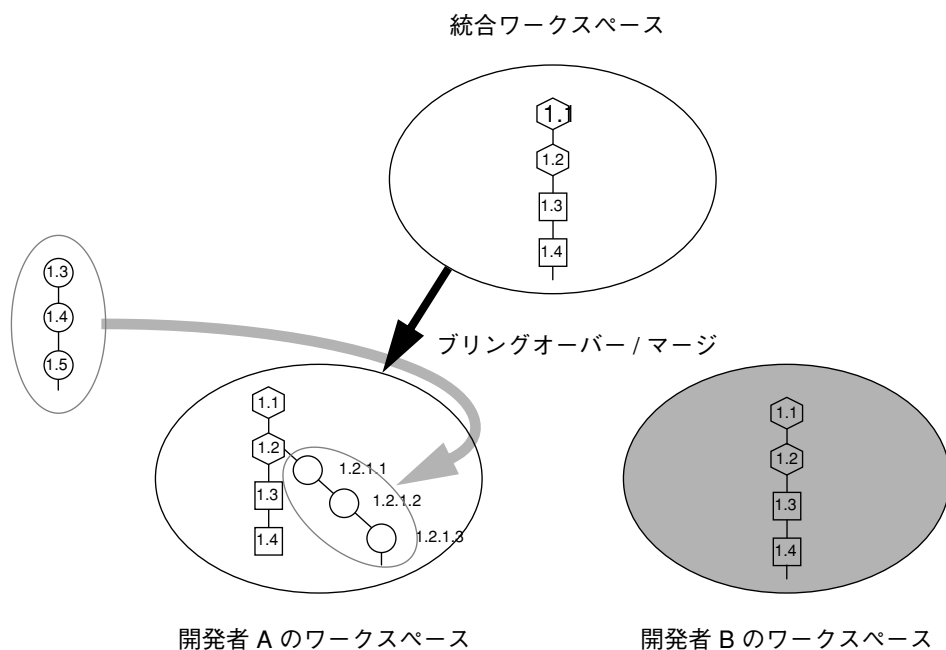
一方、開発者 A も、コピーしてきた同じファイルに独自の変更を加えて (3 つの新しいデルタ 1.3、1.4、1.5 を作成)、ファイルを統合ワークスペースに戻そうとします。

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阻止します。阻止しなかった場合、開発者 B がプットバックした変更は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。



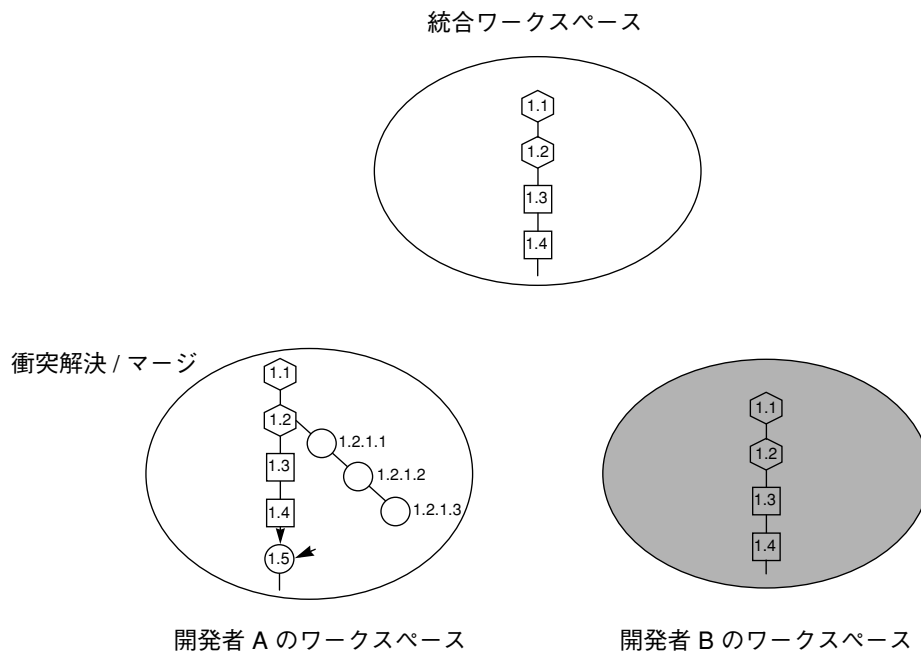
開発者 A は、開発者 B による変更を含むファイルを統合ワークスペースから自分のワークスペースにコピーします。これで、開発者 B によって作成されたデルタが開発者 A の SCCS 履歴ファイルに追加されます。

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、それぞれに新しい 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。このため、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。

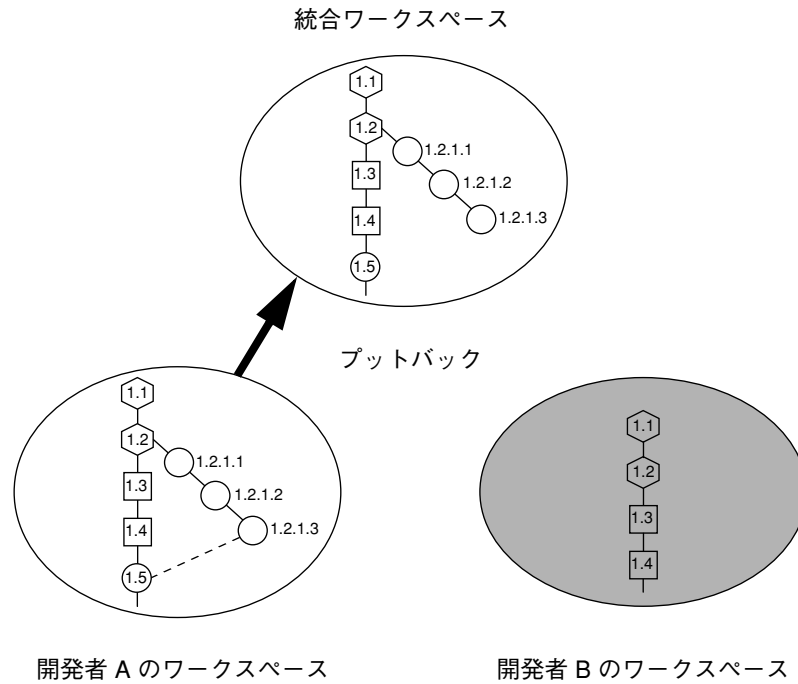


開発者 A は、自分のワークスペースで衝突解決トランザクションを使用して衝突を解決します (衝突解決についての詳細は、89 ページの「相違の解決」を参照)。開発者 A は衝突解決トランザクションを使用して、SID 1.2.1.3 と 1.4 で表わされるそれぞれのファイルバージョンをどのようにマージするかを決定します。開発者 A が変更を適用すると、衝突解決トランザクションは新たにマージされた内容を新しいデルタ (1.5) に書き込みます。

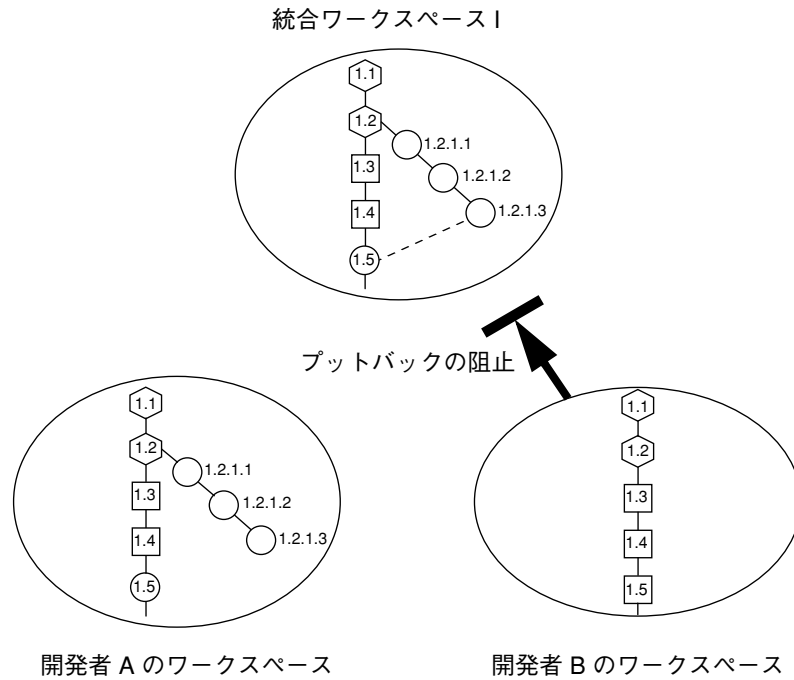
- 丸で囲まれた新しいデルタ 1.5 は、開発者 A によって作成されたことを示します。
- この新たに作成されたデルタは、開発者 A が次に行う変更のデフォルトの追加先になります。



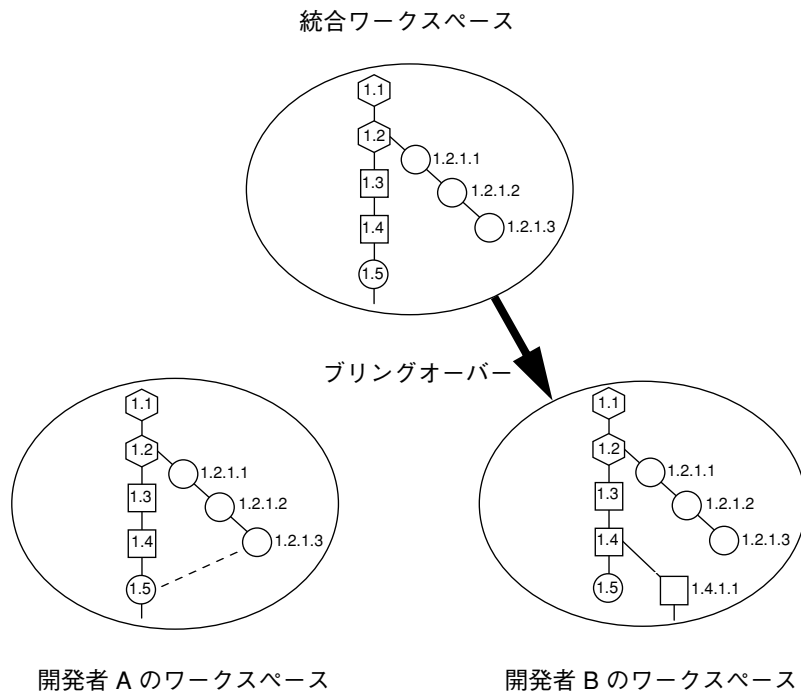
以上でファイルの衝突が解決されたため、開発者 A は自分の最新のファイルを統合ワークスペースにプットバックできます。このとき、分岐と新たに作成されたデルタは、統合ワークスペースの SCCS 履歴ファイルに追加されます。



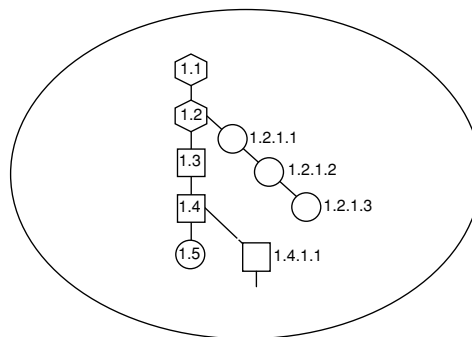
開発者 B は、自分のワークスペースでファイルに別の変更を行います。この結果、デルタ 1.5 が作成されます。続いて、この新しい変更を統合ワークスペースにプットバックしようとしても、開発者 A がその前に統合ワークスペースに戻した新しいデルタ 1.5 と衝突するため、このトランザクションは阻止されます。



開発者 B は、開発者 A が統合ワークスペースに戻し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。



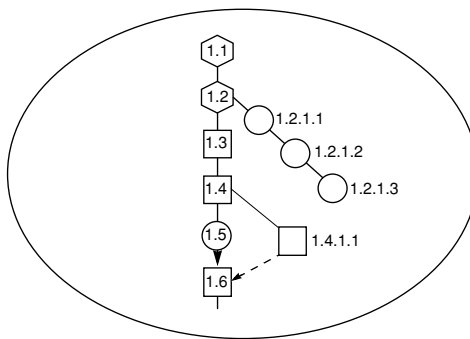
前述の例と同様に、ワークスペース管理は、開発者 B が作成したデルタを、基幹デルタツリーの最新の共通デルタに分岐として追加し、それに応じて番号を付けなおします。1.5 が 1.4.1.1 になります。1.4.1.1 は、デフォルトのデルタのままです。衝突が解決される以前に開発者 B のワークスペースで新たに作成されたデルタはすべて、この分岐に追加されます。



開発者 B のワークスペース

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 が所有します。
- 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 のワークスペース

上記のファイルマージ例は、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ブ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。変更をプ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、更新ブ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) を始点とする SCCS 分岐に移動されます。

ワークスペース管理は、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 間の変換

SID から SMID への変換および SMID から SID への変換には、それぞれ `freezept sid` コマンド、`freezept smid` コマンドを使用します。これらのコマンドをスクリプトまたはプログラムで使用して、デルタを追跡することもできます。

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] file
```

- `-r` オプションには、*file* 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、*file* 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 "d6a5c61f 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 "b1e24be3 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] file
```

- `-m` オプションには、*file* 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、*file* 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 には空白文字が含まれるため、SMID 全体を引用符で囲む必要があります。

例

```
example% freezept sid -m "64fdd0df de9d7dd de75812 23da96aa"  
module.c  
SMID "64fdd0df de9d7dd de75812 23da96aa" = SID 1.36
```

```
example% freezept sid -a bringover.1  
SMID "b05b0a2f 1db5246e 1a466014 707e38f5" = SID 1.1  
SMID "d6a5c61f 5634f0ef 9847a080 d0d7b212" = SID 1.2  
SMID "e31acdd5 6c1232e2 9e81c287 ledb2f41" = SID 1.2  
SMID "c34c91b4 a818622a 2457356a 489b2728" = SID 1.3  
SMID "98c0fd8d 889563fb cf722c2b 6afc9636" = SID 1.4  
SMID "b1e24be3 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 b95b9973 93958a07 b960074c" = 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、Forte 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>に長すぎる行があるか、期待していないファイルの終りがあります</p> <p>意味:<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>
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(プ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> <p>対処法:workspace parent 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>

表 A-1 エラーメッセージ (続き)

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「..」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「..」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「..」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 /etc/passwd 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 (Codemgr_wsdata 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、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、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プ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>

表 A-1 エラーメッセージ (続き)

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> <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 <ロック名> in workspace <ワークスペース名></p> <p>意味: ワークスペースロックファイル (Codemgr_wsdata/locks) 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、locks(4) のマニュアルページまたは 67 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 (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>

表 A-1 エラーメッセージ (続き)

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> <p>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。更新ブリングオーバーランザクション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。</p> <p>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。</p>
2022	<p>無効なサブコマンド - <コマンド名></p> <p>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が、存在しないサブコマンドが指定されました。</p> <p>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。</p>
2023	<p>統合リクエストは失敗しました</p> <p>意味: 親ワークスペースでプットバックの妥当性検査プログラムが起動されましたが、プログラムからゼロ以外のステータスが返されました。</p> <p>対処法: 正しい変更要求 ID を指定し、プットバック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)

2024	<p>ファイル <ファイル名> にデルタはありません</p> <p>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。</p> <p>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。</p>
2025	<p><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</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 PATH 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 (Codemgr_wsdata 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 175 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。</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>

表 A-1 エラーメッセージ (続き)

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 resolve 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てしました。</p> <p>対処法: edit m 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「^<<<<<<<<」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: resolve 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: list 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>
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ったため、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: resolve コマンドは、起動時に ~/.codemgr_resrc 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 ~/.codemgr_resrc 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、resolve(1) 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 CODEMGR_WSPATH で指定してください</p> <p>意味: このメッセージは、workspace list 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 CODEMGR_WSPATH 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、CODEMGR_WSPATH 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>

表 A-1 エラーメッセージ (続き)

2034	<p>内部エラー: access control operation operation_name does not have a built-in default</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 (プリングオーバー、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 (プリングオーバー、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 (Codemgr_wsdata/access_control) が存在しませんでした。</p> <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>
2036	<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>
2037	<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>

表 A-1 エラーメッセージ (続き)

2039	<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>
2040	<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> <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 twmerge コマンドを使用して、ファイルをマージしてください。twmerge コマンドについての詳細は、twmerge(1) 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: resolve 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、resolve 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 admin -h 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>

表 A-1 エラーメッセージ (続き)

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 fprintf 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>
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 (Codemgr_wsdata/*) 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 (Codemgr_wsdata/*) 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>

表 A-1 エラーメッセージ (続き)

2048	<p>子ワークスペースを作成ブリングオーバーするには、ファイルかディレクトリ、または <code>-f</code> 引数のいずれかを指定しなければなりません</p> <p>意味: ブ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/mstab</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> <p>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。</p>
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	<p>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>
	<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 Codemgr_wsdata/notification 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、notification(4) のマニュアルページまたは 41 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	<p>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>
	<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 Codemgr_wsdata/notification 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、notification(4) のマニュアルページまたは 41 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	<p>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>
	<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	<p>形式の不正な行があります。<ファイル名>:<行番号></p>
	<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: workspace updatenames 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	<p>大きさ 0 の SCCS ファイルがあります <ファイル名></p> <p>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。</p> <p>対処法: 該当する SCCS 履歴ファイルを削除してください。</p>
2058	<p>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</p> <p>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。</p> <p>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。</p>
2059	<p>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</p> <p>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。</p> <p>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。</p>
2060	<p>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</p> <p>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。</p> <p>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。</p>
2061	<p>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</p> <p>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。</p> <p>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。</p>
2062	<p>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</p> <p>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。</p> <p>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。</p>
2063	<p>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</p> <p>意味: 既存のワークスペースの下の階層に新しいワークスペースを作成しようとしていました。</p> <p>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。</p>

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。 「workspace delete <ワークスペース名>」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: workspace delete 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>/Codemgr_wsdata/access_control 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、access_control 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 access_control(4) 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 Codemgr_wsdata/notification 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：Codemgr_wsdata/notification 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、notification(4) のマニュアルページまたは 41 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 Codemgr_wsdata/notification 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、プリンティングオーバー) が指定されていないことを示しています。</p> <p>対処法：Codemgr_wsdata/notification 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>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っ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	未使用
2070	<p>説明ファイルが壊れています：名前がありません</p> <p>説明ファイルが壊れています：説明のキーワードがありません</p> <p>意味：ワークスペースの説明メタデータファイル (Codemgr_wsdata/description) が壊れています。</p> <p>対処法：ワークスペースの description のメタデータファイルを再作成してください。</p>

表 A-1 エラーメッセージ (続き)

2071	<p>-m または -c オプションを指定してください</p> <p>-m または -c オプション 1 個のみが使用可能です</p> <p>意味: <code>workspace find</code> コマンドは、-m または -c オプションのいずれかのみを受け付けます。これらのオプションを同時に指定することはできず、また、どちらか一方を必ず指定する必要があります。</p> <p>対処法: 引数を正しく指定していることを確認し、<code>workspace find</code> コマンドを再実行してください。</p>
2072	未使用
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> <p>対処法: ご購入先にお問い合わせください。</p>
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 777 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>

表 A-1 エラーメッセージ (続き)

	<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、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> を消去しようとしてしました。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な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 777 です。</p>
2081-2087	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: 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、<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> <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移動しません。</p> <p>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。</p>

表 A-1 エラーメッセージ (続き)

2090	<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> の名前テーブルへ書き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>
2091	<p>内部エラー: <code>hash table missing entry</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>
2092	<p>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。</p> <p>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。</p> <p><code>path_name/workspace updatenames <ワークスペース名></code></p> <p>意味: ワークスペース内で SCCS 履歴ファイルが <code>cp</code> 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。<code>workspace updatenames</code> 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できるようになります。</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>
2093	<p>内部エラー : <code>SmIDs not equivalent</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>

表 A-1 エラーメッセージ (続き)

2094	<p>内部エラー : SmID not found</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>
2095	<p>SmID をファイル <ファイル名> 内で一意にできません</p> <p>意味 : ワークスペース管理コマンドにより、ルートのデルタが同じファイルが検出され、それらファイルの識別に失敗しました。</p> <p>対処法 : <ファイル名>を含むワークスペースで 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。</p>
2096	未使用
2097	<p>ワークスペース <ワークスペース名> を削除できません。</p> <p>意味 : workspace move コマンドが、コピー後の元のワークスペースの削除に失敗しました。</p> <p>対処法 : アクセス権を確認し、元のワークスペースを手動で削除してください。</p>
2098	<p>ワークスペース <ワークスペース名> を削除できません。アクセス権をチェックしてください。</p> <p>意味 : workspace delete コマンドが指定したワークスペースの削除に失敗しました。</p> <p>対処法 : アクセス権を確認し、コマンドを再実行してください。</p>
2099-2499	未使用
2500 - 2536	<p>内部エラー。</p> <p>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示し、ユーザー側で対処することはできません。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。</p>
2537	<p>ファイル <親ファイル> は、ファイル <子ファイル> と互換性がありません (text/binary)</p> <p>意味 : プリンクオーバーまたはプットバックで、<親ファイル> と <子ファイル> のタイプ (テキスト/バイナリ) に互換性がないことが検出されました。</p> <p>対処法 : ファイルを修正するか、プリンク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>
2538	<p>Can't set autofreezepoint properties.</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>

表 A-1 エラーメッセージ (続き)

2539-2600	<ワークスペース名> へのプットバックしか許可されていません
	<p>意味: プットバックの妥当性検査が有効になっているワークスペース <ワークスペース名> にファイルをプリングオーバーしようとしています。プットバックの妥当性検査が有効になっているワークスペースには、プットバックしか行えません。</p> <p>対処法: <ワークスペース名> に対するプットバックの妥当性検査を無効にし、プリングオーバーを再実行してください。</p>
2610	<ワークスペース名> へのプットバックしか許可されていません
	<p>意味: プットバックの妥当性検査が有効になっているワークスペース <ワークスペース名> に、ファイルをプリングオーバーしようとした。プットバックの妥当性検査が有効になっているワークスペースには、プットバックしか行えません。</p> <p>対処法: <ワークスペース名> に対するプットバックの妥当性検査を無効にし、プリングオーバーを再実行してください。</p>
5005	作成先ディレクトリ <ディレクトリ名> は空ではありません
	<p>意味: <code>freezept extract</code> に指定されたディレクトリ <ディレクトリ名> が空ではありません。</p> <p>対処法: 抽出先ディレクトリとして空のディレクトリを指定し、<code>freezept extract</code> コマンドを再実行してください。</p>
5009	不正な日付の書式:
	<p><日付></p> <p>正しい書式:</p> <p>YYYY/MM/DD または YYYY/MM/DD/hh/mm/ss</p>
	<p>意味: <code>freezept</code> コマンドに指定された日付の形式が不正です。</p> <p>対処法: YYYY/MM/DD または YYYY/MM/DD/hh/mm/ss の正しい形式で日付を指定し、<code>freezept</code> コマンドを再実行してください。</p>
5010	フリーズポイント <フリーズポイントファイル名> で指定されている <ファイル名> は、ワークスペース <ワークスペース名> に存在しません
	<p>意味: このメッセージは、<code>freezept extract</code> コマンドが、<フリーズポイントファイル名> に、ワークスペース <ワークスペース名> 内のファイル <ファイル名> に存在しないデルタに対する参照を検出したときに返されます。このエラーの原因として可能性が高いのは、フリーズポイントファイルが <code>freezept extract</code> コマンドに指定されたワークスペースを基準に、<code>freezept update</code> 処理されたワークスペースから作成されていることです。</p> <p>対処法: 適切なワークスペースを指定し、<code>freezept extract</code> 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)

5011	<p>このディレクトリには、smid が重複している SCCS ファイルがあります。</p> <p>意味: freezept create コマンドで指定したディレクトリに、ルートデルタが同じファイルが含まれています。</p> <p>対処法: ディレクトリから新しい Forte TeamWare ワークスペースを作成し、freezept create コマンドを再実行してください。</p>
5103	<p>フリーズポイントファイル <フリーズポイントファイル名> は破壊されています。</p> <p>意味: 指定したフリーズポイントファイルが破壊されています。</p> <p>対処法: ファイルを修正するか、新しいバージョンのファイルをコピーし、freezept コマンドを再実行してください。</p>
5107	<p>フリーズポイントファイル <フリーズポイントファイル名> には、デフォルトワークスペースはありませんので、-w オプションでワークスペースを必ず指定してください</p> <p>意味: 指定されたフリーズポイントファイルには、freezept update または freezept extract コマンドにデフォルトで使用するワークスペースが含まれていません。</p> <p>対処法: 適切なワークスペース名を指定し、freezept コマンドを再実行してください。</p>
5108	<p>抽出中のエラー数は <数></p> <p>意味: freezept extract コマンドによって、遭遇したエラーの数が作業終了後に表示されます。</p> <p>対処法: なし。</p>
5109	<p>フリーズポイントファイル <フリーズポイントファイル名> にはデフォルトのワークスペースがありません</p> <p>意味: 指定されたフリーズポイントファイルには、freezept update または freezept extract コマンドにデフォルトで使用するワークスペースが含まれていません。</p> <p>対処法: 適切なワークスペース名を指定し、freezept コマンドを再実行してください。</p>
6000	<p>-l オプションを使用する場合、左右両ファイルともディレクトリでなければなりません。</p> <p>意味: filemerge コマンドに -l オプションを使用する場合、コマンド行に指定する左右のファイルはディレクトリ名でなければなりません。</p> <p>対処法: 左右両ファイルに適切なディレクトリ名を指定し、filemerge コマンドを再実行してください。</p>
6003	<p>ファイルマージ: 出力行の相違の解析ができません: <行></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>

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

2020	<p>-p オプションで定義されている親： <ワークスペース名> は、「本来の親」： <親ワークスペース名> と一致していません</p> <p>意味：bringover または putback の -p オプションに指定されたワークスペース<ワークスペース名>は、bringover または putback トランザクションに指定された子の親ワークスペースではありません。この子ワークスペースの実際の親は <親ワークスペース名> です。</p>
2023	<p>コマンド実行中にそのコマンドのロックファイルを削除すると、親または子ワークスペースの SCCS ファイルを壊すことがあります。</p> <p>意味：メッセージの説明のとおりです。workspace locks コマンドから返されるメッセージです。</p>
2601	<p>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</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> を消去しようとしていました。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。</p> <p>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。</p>
2602	<p>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</p> <p>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。</p> <p>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。</p>

表 A-2 警告メッセージ (続き)

2603	<p>ファイル名の長さが 0 です - 無視しました</p> <p>意味: コマンド行の引数として (または /Codemgr_wsdata/args 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、args 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 /Codemgr_wsdata/args 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、args 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。</p> <p><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 workspace delete または workspace move 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 children 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 children 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

表 A-2 警告メッセージ (続き)

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 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するか、CLI の 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、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	<p>ワークスペース <ワークスペース名> には Codemgr_wsdata/args ファイルがありません ワークスペース <ワークスペース名> に空の Codemgr_wsdata/args ファイルがあります</p> <p>意味: プリングオーバーやプットバックに指定したワークスペースには、args メタデータファイルが存在しないか、args メタデータファイルが空です。この場合、トランザクションで使用されるデフォルトのファイルやディレクトリを特定できません。</p> <p>対処法: トランザクションに使用する適切なファイル/ディレクトリをコマンド行に指定し、プリングオーバー/プットバックコマンドを再実行してください。</p>
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: Codemgr_wsdata/nametable 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、nametable 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>

表 A-2 警告メッセージ (続き)

	<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プ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> <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>

表 A-2 警告メッセージ (続き)

	<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 Codemgr_wsdata/locks 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 workspace locks 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。</p> <p>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。ブリングオーバート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> <p>対処法: なし。</p>
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 Codemgr_wsdata/conflicts 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 conflicts 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、ブ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、ブ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 resolve コマンドを使用してその衝突を解決してから、ブ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、ブリングオーバーまたはプ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>

表 A-2 警告メッセージ (続き)

	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。
2628	<p>チェックアウトコマンドが PATH <コマンドパス> にないため、ファイルは更新されません。</p> <p>意味: SCCS get 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-ファイルをブリングオーバーまたはプ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 get コマンドのパスを含めるようにしてください。</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 Codemgr_wsdata 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 workspace delete 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> をブリングオーバーできませんでした。</p> <p>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-file が存在する)、書き込み可能でないため、ブリングオーバートランザクションでブ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トランザ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: Forte 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>

表 A-2 警告メッセージ (続き)

2699	<p>ファイル <ファイル名> は、<ソースワークスペース名> で <ファイル名_1> に名前変更され、<ファイル名> と <ファイル名_1> が <作成先ワークスペース名> に存在するので無視されました</p> <p>意味: 名前変更が巡回して行われています。たとえば、<ファイル名> と <ファイル名_1> というファイルが、親および子ワークスペース両方にあるとします。このとき、親ワークスペースで、<ファイル名> が <ファイル名_1> に、<ファイル名_1> が <ファイル名> に名前変更されたとします。</p> <pre data-bbox="475 674 1122 758">% workspace filemv <ファイル名> <一時ファイル> % workspace filemv <ファイル名_1> <ファイル名> % workspace filemv <一時ファイル> <ファイル名_1></pre> <p>bringover では、このような名前変更 (一時的なファイル名を使用する方法) を子ワークスペース内では実行できないため、警告メッセージが表示されます。</p>
------	--

付録 B

問題の対処方法

この付録では、Forte TeamWare の一般的な問題と対処方法を説明します。

- 228 ページの「プリングオーバーとプットバック」
- 229 ページの「環境変数」
- 230 ページの「Java」
- 231 ページの「Microsoft Windows と UNIX オペレーティングシステム間の互換性」
- 232 ページの「Microsoft Windows のファイル共有」
- 233 ページの「プロセス監視」
- 234 ページの「使用すべきでない SCCS コマンド」
- 235 ページの「SCCS 履歴ファイル」
- 236 ページの「SCCS のエラー」
- 237 ページの「テキスト形式」
- 240 ページの「バージョンの確認方法」
- 239 ページの「ワークスペース」

ブリングオーバーとプットバック

表 B-1 に、ブリングオーバーとプットバックに関連する一般的な問題を示します。

表 B-1 ブリングオーバーとプットバックに関連する問題

問題	対処法
プットバック中にエラーが返される	<ol style="list-style-type: none">1. プットバック処理をプレビューします。これにより、実際にファイルを更新せずに、いくつかのファイルが更新されることになるかを調べます。Forte TeamWare の GUI で「プットバック」ダイアログの「プレビュー」ボックスを選択するか、コマンド行で次のコマンドを使用します。 \$ cd <子ディレクトリ名> \$ putback -n (<子ディレクトリ名> は、子ワークスペースの最上位のディレクトリ名に置き換えてください。)2. プットバック結果を書き込むためのディスク領域が十分にあるかどうかを確認します。親ワークスペースがあるディスクの空き領域を確認してください (次の対処方法を参照)。また、/tmp ディレクトリと使用可能なスワップ空間も確認してください。3. 問題が解決されない場合は、部分的にプットバックを行って問題がどのファイルにあるかを特定します。プットバックは1つのディレクトリだけのこともあれば、1つのファイルだけのこともあります。
プットバック中に次のエラーメッセージが返される。 「プットバック: ファイル <ファイル名> の XXX 行目の書式が正しくありません... (警告 2615)」 (XXX は行番号)	親ワークスペースに書き込むために必要なディスク領域が不足している可能性があります。次のように、親ワークスペースがあるディレクトリで <code>df -k</code> コマンドを実行し、使用可能なディスク領域を確認してください。 \$ df -k <親ワークスペースのディレクトリ> 注: 使用可能なディスク領域の大きさは、K バイト単位で表示されます。
空白文字を含むファイル名に関する問題	コマンド行インタフェースで指定するファイル名に空白文字が含まれている場合は、必ず二重引用符 (") で囲ってください。Forte TeamWare GUI では、名前に空白文字を含むファイルが関係するトランザクションを正しく処理できます。

環境変数

表 B-2 に、環境変数に関連する問題を示します。

表 B-2 環境変数に関連する問題

問題	対処法
Samba のファイル共有機能を使用して正しく設定しているにも関わらず、Microsoft Windows プラットフォームでのプットバックでエラーが発生する	243 ページの「環境変数の設定」に示す環境変数を調べ、設定が適切でない場合は訂正します。具体的には、SENDMAILHOST 環境変数に有効な Sendmail ホストが設定されていないと、問題が発生することがあります。
ブリングオーバー、プットバック、ワークスペース管理、SCCS のいずれかの操作で、次のエラーメッセージまたは症状のいずれかが発生する。 <ul style="list-style-type: none">• /util/find: not found PC/bin/def.dir.flp:/SunWorkshop/ TeamWare/bin• cd : lstat /X failed• エラーは発生せずにプットバックが終了するが、実際には更新内容がプットバックされていない。• sccs edit <ファイル名> コマンドを実行しても、何の処理も行われずにプロンプトが返される。	PATH 環境変数を調べ、/bin ディレクトリ内の他のアプリケーションよりも前に TeamWare/bin が指定されていることを確認してください。

Java

表 B-3 に、Java に関連する問題を示します。

表 B-3 Java に関連する問題

問題	対処法
Forte TeamWare の GUI ツールを起動すると次のエラーになる。 NoClassDefFoundError	<インストール先のルート>/bin ディレクトリにある以下の起動スクリプトを使用し、Forte TeamWare GUI ツールを起動してください (<インストール先のルート> は Forte TeamWare をインストールしたルートディレクトリ名に置き換えてください)。 <ul style="list-style-type: none">• codemgrtool• vertool• filemerge• freezepttool Microsoft Windows では、「スタート」メニューにこれら起動スクリプトのショートカットがあります。これらのスクリプトには、必須 CLASSPATH と Java インタプリタ呼び出しが含まれています。この方法を使用しても NoClassDefFoundError が返される場合は、Forte TeamWare の再インストールが必要になる場合があります。再インストールすることによって、スクリプトが正しい CLASSPATH 情報で更新されます。

Microsoft Windows と UNIX オペレーティングシステム間の互換性

サポートされている UNIX オペレーティングシステム (Solaris および Linux) と Microsoft Windows との間に互換性がないことが原因で、問題が発生する場合があります。表 B-4 に、それらの問題を示します。

表 B-4 Microsoft Windows と UNIX 間の互換性に関連する問題

問題	対処法
行の終わり (EOL) 文字の問題 UNIX 環境では改行 (LF) 文字で EOL を示すのに対し、Microsoft Windows 環境では復帰改行文字 (CRLF) 文字で EOL を示します。たとえば、Microsoft Windows にあるワークスペースに対し、UNIX 環境で作成されたファイルの更新プリングオーバーを行うと、問題が発生することがあります。	デフォルトでは、Forte TeamWare は EOL 文字を変換しません (ほとんどの場合、このことは問題になりません)。EOL に関する問題が発生する場合は、次のように環境変数を設定することによって強制的に EOL 文字を変換できます。 <code>SCCS_GET_TRANS_LF=Yes</code> 注：この問題は、Microsoft Windows 環境にだけ該当します。
空白文字を含むワークスペース名に関する問題	通常、UNIX 環境では、空白文字を含む名前は有効ではありません。空白文字を含むワークスペース名で問題が発生した場合は、ワークスペースの親を変更することによって解決できます。ワークスペース管理ツールで「ワークスペース」▶「親の変更」を選択します。
空白文字を含むファイル名に関する問題	Forte TeamWare GUI では、名前に空白文字を含むファイルが関係するトランザクションを正しく処理できます。コマンド行インタフェースで指定するファイル名に空白文字が含まれている場合は、必ず二重引用符 (") で囲んでください。

Microsoft Windows のファイル共有

Microsoft Windows のファイル共有サービスとの互換性、またはその不適切な設定が原因で問題が発生することがあります。表 B-5 に、それらの問題を示します。

表 B-5 Microsoft Windows のファイル共有に関連する問題

問題	対処法
Microsoft Windows のファイル共有サービスとの互換性、またはその不適切な設定が原因で発生する問題には、以下のものがあります。	Samba は、Forte TeamWare がサポートする唯一のファイル共有サービスです。Forte TeamWare における Samba の設定については、248 ページの「Samba ファイルサーバーの構成」を参照してください。
<ul style="list-style-type: none">• Codemgr_wsdata ファイルに対する不適切なアクセス権の設定によるエラー。• プットバックのプレビューコマンド (putback -n) では機能するが、実際のプットバックではハングアップする。• 複数のユーザーが同時に同じワークスペースにプットバックしようとするすると、Forte TeamWare が動かなくなる。• Microsoft Windows で画面に何も表示されず、制御を回復するために再起動が必要になる。• Solaris で、SCCS マスターファイルを所有しているユーザーの場合はプットバックに成功するが、SCCS マスターファイルを所有していないユーザーの場合はプットバックに失敗する。	

表 B-5 Microsoft Windows のファイル共有に関連する問題 (続き)

問題	対処法
Samba のファイル共有機能を使用して正しく設定しているにもかかわらず、プットバックでエラーが発生する	243 ページの「環境変数の設定」に示す環境変数を調べ、設定が適切でない場合は訂正します。具体的には、SENDMAILHOST 環境変数に有効な Sendmail ホストが設定されていないと、問題が発生することがあります。
Microsoft Windows NT Service Pack 5 をインストールすると、Samba で共有しているファイルシステムをマウントできなくなる	Microsoft Windows NT Service Pack 5 では、いくつかのバージョンの Samba で問題が発生することがあります。最新バージョンの Samba を使用するか、Microsoft Windows NT Service Pack 4 に戻してください。

プロセス監視

表 B-6 に、プロセス監視に関連する問題を示します。

表 B-6 プロセス監視に関連する問題

問題	対処法
現在実行中の Forte TeamWare プロセスの監視方法	次の対処方法は、Solaris プラットフォームだけで使用できます。 コマンド行で次のコマンドを入力します。 <pre>truss -o /tmp/truss_output -f \ -vall -p <TeamWare の PID></pre> <TeamWare の PID> は、現在実行中の Forte TeamWare プロセスのプロセス ID に置き換えてください。

使用すべきでない SCCS コマンド

次の SCCS コマンドは使用しないでください。これらのコマンドを使用すると、SCCS 履歴ファイルが変更されるため、Forte TeamWare のツールがファイルの履歴を正しく追跡できなくなります。

- cdc
- comb
- fix
- rmDEL

また、最終的にマージする予定の複数のワークスペース内にあるファイルに対して、これらのコマンドを使用すると、悪影響が生じることがあります。

表 B-7 に、これらのコマンドが原因で発生する可能性のある問題を示します。

表 B-7 使用すべきでない SCCS コマンド

SCCS コマンド	関連する問題
cdc	不要な分岐が発生して履歴が混乱します。
comb	SCCS 履歴ファイルを完全に再作成します。
fix	rmDEL のフロントエンドとして使用するコマンドです。
rmDEL	rmDEL コマンドは、分岐上の最新の (葉状の) デルタを削除する場合にのみ使用します。別のワークスペースにも存在するデルタを削除すると、別のユーザーが他のワークスペース内にデルタを追加することが可能になります。自分のワークスペースで削除したデルタは、ファイルがマージされた時には葉状のデルタではなくなります。

SCCS 履歴ファイル

表 B-8 に、SCCS 履歴ファイルに関連する一般的な問題を示します。

表 B-8 SCCS 履歴ファイルに関連する問題

問題	対処法
SCCS 履歴ファイルの 妥当性検査の方法	特定のファイルの診断テストには、 <code>sccs-val</code> コマンドを使用してください。 <code>sccs-val</code> コマンドは各ファイルについて次の問題を検査します。 <ul style="list-style-type: none">履歴ファイルが破壊されていないかどうか履歴ファイルの読み取りに問題がないかどうか
破壊された SCCS 履歴 ファイルの復元方法	SCCS 履歴ファイルにはチェックサムが含まれています。SCCS 履歴ファイルが破壊された場合は、チェックサムを修正する必要があります。この修正は、次のコマンドで行います。 <pre>sccs admin -z program.c</pre> <p>注：SCCS 履歴ファイルが破壊されると、チェックサムが不正になること以上に重大な問題になる可能性があります。SCCS 履歴ファイルを修正する前には、必ず最新の履歴ファイルのバックアップを作成してください。</p>

SCCS のエラー

表 B-9 に、SCCS のエラーに関連する問題を示します。

表 B-9 SCCS エラーに関連する問題

問題	対処法
<p>Y2K (西暦 2000年) 問題と古いバージョンの SCCS が原因で、次のいずれかのエラーメッセージが返されることがあります。</p> <ul style="list-style-type: none">• エラー [SCCS/s.foo.java] X 行で書式エラー (co4)• プリンクオーバー: get -s -Gfoo SCCS/s.foo コマンドは、ステータス 1 で終了しました (エラー 2012) <p>(foo は実際のファイル名、X は行番号)</p>	<p>次の対処方法は Solaris プラットフォームだけで使用できます。Microsoft Windows および Linux プラットフォームで判明している Y2K 関連の問題はありません。</p> <p>次のコマンドを実行し、使用している SCCS のバージョンが Y2K 問題に対応しているかどうかを調べます。</p> <pre>\$ which get</pre> <p>次のように表示されます。</p> <pre>/usr/ccs/bin/get</pre> <p>この内容に従い、次のように入力します。</p> <pre>\$ /usr/ccs/bin/what /usr/ccs/bin/get</pre> <p>次のようないずれかの値が表示されます。</p> <ul style="list-style-type: none">• SunOS 5.7 October 1998• SunOS 5.8 June 1999 <p>Solaris 2.6 (SunOS 5.6) より以前のバージョンは Y2K 問題に対応していません。使用している SCCS のバージョンが Y2K 問題に対応していない場合は、SunSolve Online から適切なパッチを入手してください。</p>
<p>sccs diffvs SCCS コマンドを実行すると、次のエラーになる。</p> <pre>diff:SCCS/xxx0.0: ファイルもディレクトリもありません。</pre>	<p>Solaris 2.6 に付属している SCCS のバージョンには、diffvs サブコマンドが SCCS ディレクトリをパラメータとして認識しないという既知のバグがあります。この問題を解決するには、Solaris 2.6 用の適切なパッチを入手してください。</p> <p>注: この問題は、Microsoft Windows および Linux プラットフォームには該当しません。</p>

テキスト形式

表 B-10 に、テキストファイルの形式に関連する問題を示します。

表 B-10 テキスト形式に関連する問題

問題	対処法
<p>Forte TeamWare がテキストファイルを誤ってバイナリファイルと認識し、次のようなエラーになることがあります。</p> <p>ブリングオーバー: ファイル "//c/builds/<子ワークスペース名>/<ファイル名>" に長すぎる行があるか、期待していないファイルの終りがあります(エラー 2000)</p> <p>ソースファイルに空白行や余分な空白文字が追加される</p>	<p>Forte TeamWare では、テキストファイルは改行で終了することを前提にしているため、改行で終了していない場合は、バイナリファイルであると認識します。この問題を回避するには、必要に応じてソースファイルに新たな行を追加し、すべてのテキストファイルを改行で終了するようにします。</p> <p>この問題を回避するには、Microsoft Windows 環境における CRLF 変換を無効に設定します (SCCS_GET_TRANS_LF=NO)。この設定によって、余分な CR 文字がクリアファイルに現れなくなります。Microsoft Visual Studio C++ では、これまでどおりファイルを処理できますが、メモ帳では、ファイルのアクセスに問題が発生する可能性があります。</p>
<p>Microsoft Windows と UNIX 環境間で共有しているテキストファイルを編集すると、ファイルに余分な行終了文字が挿入されることがあります。次のような場合に、この問題が発生する可能性があります。</p>	

表 B-10 テキスト形式に関連する問題 (続き)

問題	対処法
<p>1. Microsoft Windows 環境でファイルを作成または編集します。この場合、ファイルは、最後に復帰改行 (CRLF) が付加されて保存されます。</p>	<p>通常、Solaris と Microsoft Windows 環境の間で「クリアファイル」(sccs get コマンドで作成したファイル) を共有すると、問題が発生することがあります。このため、</p>
<p>2. そのファイルを Solaris 環境にチェックインします。Solaris の SCCS は CRLF を LF に変換しないため、履歴ファイルに CR がコピーされます (CRLF 変換を行う Solaris SCCS パッチが提供される予定です)。</p>	<p>Solaris/Samba ワークスペースは Microsoft Windows でのアクセス専用にすることを推奨します。Solaris ツールで処理するワークスペースに統合する必要がある場合は、Solaris 環境で Forte TeamWare を使用し、</p>
<p>3. Microsoft Windows 環境において、デフォルトの SCCS 操作 (SCCS_GET_TRANS_LF=YES) で同じファイルをチェックアウトします。これで、CRLF 文字が CRCRLF に変換されます。実際にはこの変換は、Microsoft Win32 の fwrite システムコールのテキストモード操作で行われます。</p>	<p>Solaris/Samba ワークスペースからプットバックを行ってください。こうすることによって、親ワークスペース内のクリアファイルが Solaris 用の正しい形式になり、子ワークスペース内のクリアファイルが Microsoft Windows 用の正しい形式になります。</p>
<p>4. Microsoft Visual Studio C++ でファイルを編集すると、余分な CR が CRLF に変換され、この結果、余分な行が作成されることとなります。メモ帳、ワードパッド、VI などのその他のエディタでは、この処理は行われません。</p>	

ワークスペース

表 B-11 に、ワークスペースに関連する一般的な問題を示します。

表 B-11 ワークスペースに関連する問題

問題	対処法
ワークスペースの破壊に関する問題	<ol style="list-style-type: none">1. <code>workspace check</code> コマンドはワークスペースを検査し、矛盾に関する情報を報告します。ファイル、アクセスモード、親子関係、履歴ファイルの状態を調査します。このコマンドを実行するには、次のようにします。 <code>workspace check <ワークスペース名></code> (<code><ワークスペース名></code> は、実際に検査するワークスペースの名前に置き換えてください。) このコマンドは次のいずれかの値で終了します。 0 = ワークスペースは正常です。 1 = エラー2. ワークスペースの自動フリーズポイント機能は、さまざまなトランザクションの前後にワークスペースのスナップショットをとります。この機能を利用したときに多少のオーバーヘッドが伴うのは、データの保全が最重要であるマスター統合ワークスペース向けの機能であるためです。自動フリーズポイント機能についての詳細は、131 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。
以前のバージョンの Forte TeamWare で作成したワークスペースの互換性の問題	以前のバージョンの Forte TeamWare で作成したワークスペースについて互換性の問題はありません。

バージョンの確認方法

表 B-12 に、使用している Forte TeamWare のバージョンの確認方法を示します。

表 B-12 バージョンの確認方法

問題	対処法
使用している Forte TeamWare のバージョンの確認方法	Forte TeamWare の GUI で「ヘルプ」▶「TeamWare について」を選択するか、コマンド行で次のコマンドを使用します。 <code>teamware -V</code>

付録 C

Microsoft Windows 環境における Forte TeamWare の使用

この付録では、Microsoft Windows 環境で Forte TeamWare を使用するときの注意事項について説明します。この付録では、以下の項目について説明します。

- 242 ページの「Forte TeamWare ツールの起動」
- 243 ページの「環境変数の設定」
- 244 ページの「コマンド行インタフェースの使用法」
- 244 ページの「ドライブ名の選択とパス名の構文」
- 245 ページの「改行コードの CR と LF の変換について」
- 246 ページの「テキストファイルへの改行文字の追加」
- 246 ページの「クロスプラットフォーム対応」
- 247 ページの「Solaris ユーザーとのファイルの共有」
- 248 ページの「Samba ファイルサーバーの構成」
- 249 ページの「Microsoft Visual Studio のプラグインとしての Forte TeamWare の使用」

Forte TeamWare ツールの起動

Forte for Java 環境で Forte TeamWare の GUI ツールを起動するには、161 ページの「Forte TeamWare ツールと機能、アクセス方法」で説明している方法を使用します。たとえば、Forte for Java の TeamWare ツールバーで「TeamWare ワークスペース管理」ボタンをクリックします。

Microsoft Windows では、「スタート」メニューから Forte TeamWare の GUI ツールを起動することもできます。ただし、この場合、ツールはスタンドアロンモードで起動されるため、ツールは Forte for Java 環境に統合されません。スタンドアロンモードで Forte TeamWare ツールを起動する場合は、「スタート」メニューから「プログラム」▶「Forte for Java」▶「TeamWare」を選択し、表 C-1 に示すショートカットを選択します。

表 C-1 Microsoft Windows の「スタート」メニューにある Forte TeamWare のショートカット

ショートカット	bin ディレクトリ内の対応するファイル
CodeManager Tool (ワークスペース管理)	Codemgrtool.BAT
Command-Line Interface (コマンド行インタフェース)	CommandShell.BAT
FileMerge Tool (ファイルマージ)	FileMerge.BAT
Freezepting Tool (フリーズポイント)	Freezepttool.BAT
Versioning Tool (バージョン管理)	Vertool.BAT
ショートカット	説明
Quick Tour	対話形式の Forte TeamWare Quick Tour
Release Notes	重要な最新情報
User's Guide	このマニュアル

注意 – Forte TeamWare インストールディレクトリ内のバッチファイルは、製品をインストールしなおすたびに上書きされます。バッチファイルをカスタマイズした場合は、ホームディレクトリなどの、安全な場所にそれらのファイルをコピーしておいてください。

環境変数の設定

ショートカットや対応するバッチファイル (表 C-1 を参照) を使用せずに Forte TeamWare ツールを起動する場合は、次の環境変数を設定する必要があります。Microsoft Windows プラットフォーム上で環境変数を設定するには、以下のようになります。

1. 「スタート」 ▶ 「設定」 ▶ 「コントロールパネル」 を選択します。
2. 「システム」 をダブルクリックし、「システムのプロパティ」 コントロールパネルを起動します。
3. 「環境変数」 タブをクリックします。
4. 表 C-2 で示す環境変数を設定します。

表 C-2 Microsoft Windows における Forte TeamWare 用の環境変数の設定

環境変数	説明
PATH	Forte TeamWare がインストールされている bin ディレクトリへのフルパス名を指定します。実行ファイルのインストールディレクトリについての詳細は、リリースノートを参照してください。
SENDMAILHOST	sendmail デーモンを実行しているマシンのホスト名を指定します。この環境変数の設定は、プリングオーバーやプットバックなどのトランザクション、および使用状況の追跡を送信する通知機能のために必要です。
CLASSPATH	バッチファイルスクリプト (「プログラム」メニューのショートカット) を使用して Forte TeamWare を起動する場合、CLASSPATH を設定する必要はありません。それ以外の場合は、必ず、CLASSPATH に teamware.jar ファイルへのパスを設定します。teamware.jar ファイルのインストールディレクトリについての詳細は、リリースノートを参照してください。

コマンド行インタフェースの使用方法

Forte TeamWare の機能は、GUI ツールおよびコマンド行のどちらからでも利用できます。コマンド行インタフェースを使用する場合は、MKS、Cygnus、UWIN、Microsoft Windows のコマンドプロンプトなどのコマンドシェル環境を利用することができます。Microsoft Windows のコマンドプロンプトを利用するには、「スタート」メニューから、順に次のメニュー項目を選択します。

「プログラム」▶「Forte for Java」▶「TeamWare」▶「Command-Line Interface」

ドライブ名の選択とパス名の構文

Microsoft Windows プラットフォーム上のファイル名およびパス名の構文規則は、サポートされている UNIX 環境 (Solaris および Linux) での構文規則とは異なります。また、Microsoft Windows プラットフォームでは、標準の DOS 構文のほかに、MKS、Cygnus、UWIN などの、一般的に使用されている UNIX エミュレーションツールでサポートされる他の構文規則もサポートしています。

Forte TeamWare では、多様なパス名の構文規則をサポートしているため、Microsoft Windows 上で任意の構文を選択できます。使用する構文は、環境変数 CODEMGR_PATHNAME_STYLE を設定することで選択します。表 C-3 に、サポートされる構文規則と対応する CODEMGR_PATHNAME_STYLE の設定を示します。

表 C-3 Microsoft Windows 用にサポートされているパス名の構文規則

構文	例	CODEMGR_PATHNAME_STYLE の設定
Microsoft Windows (DOS シェル)	c:\dir\file	DOS
代替 1: MKS	c:/dir/file	ALT1
代替 2: Canonical UNIX	//c/dir/file	ALT2

c はドライブ名、dir はディレクトリ名、file はファイル名を示します。

注 – Microsoft Windows では、/dir1/dir2 のようにドライブ名がないと指定が不完全になるため、必ずドライブ名を指定してください。

改行コードの CR と LF の変換について

UNIX と Microsoft Windows プラットフォームでは、テキストファイルの行の終わり (EOL) 記号の規則が異なります。UNIX 環境では改行 (LF) 文字で EOL を示すのに対し、Microsoft Windows 環境では復帰改行文字 (CRLF) 文字で EOL を示します。互換性を維持するために、SCCS では SCCS 履歴ファイル内のユニバーサル EOL 記号として単一の LF を使用しています。Microsoft Windows のテキストファイルを SCCS でチェックインすると、すべての CRLF 行区切り記号は自動的に LF 文字に変換されます。この処理をユーザーが無効にすることはできません。

`sccs get` や `sccs edit` などの SCCS 処理によって「クリアファイル」と呼ばれるファイルの作業用バージョンが作成された場合、デフォルトでは、SCCS は単一の LF を行区切り記号として残します。ほとんどの Microsoft Windows アプリケーションは、単一の LF を行区切り記号として正しく処理することができます。CRLF 規則を適用する必要がある場合は、環境変数 `SCCS_GET_TRANS_LF` に `Yes` を設定することによって、クリアファイル内の LF 区切り記号を CRLF 区切り記号に変換するように SCCS を設定できます。

例：

```
set SCCS_GET_TRANS_LF=Yes
export SCCS_GET_TRANS_LF
```

CRLF 変換を行うかどうかは、Forte TeamWare トランザクションの動作にも影響します。Forte TeamWare によってプリングオーバーまたはプットバック処理が行われた場合、デフォルトでは、SCCS は更新された履歴ファイルごとにクリアファイルを 1 つ抽出します。ワークスペース内のクリアファイルの形式は、`SCCS_GET_TRANS_LF` 変数の設定によって決まります。

Microsoft Windows ほとんどのエディタでは、単一の LF 文字が問題になることはなく、UNIX プラットフォームからの表示に適しているという付加的な利点があります。ただし、プラットフォーム間で Forte TeamWare ワークスペースを共有する場合は、デフォルトの SCCS (非変換) モードを使用するか、`SCCS_GET_TRANS_LF` に `No` を設定してください。

テキストファイルへの改行文字の追加

必ずしもすべてのアプリケーションが、テキストファイルの最終行に改行文字を付加するわけではありません。テキストファイルの最終行に改行文字があると、Forte TeamWare がそのテキストファイルをバイナリファイルと解釈することがあります。これを防ぐために、Forte TeamWare では、以前にテキストファイルと認識したファイルに改行文字を付加します (ファイルの最後に改行文字がない場合)。

この機能は、次のように環境変数を設定することによって無効にできます。

```
SCCS_NO_NL_TERMINATE=YES
```

クロスプラットフォーム対応

Forte TeamWare では、SCCS 履歴ファイルとワークスペースメタデータ構造にプラットフォームに依存しない共通の形式を使用しています。適切なファイル共有サービスを使用することによって、Forte TeamWare が動作する、異なるプラットフォーム間でワークスペースを共有することができます。

SCCS 履歴ファイルの共有

SCCS (ソースコード管理システム) 履歴ファイルの形式はプラットフォーム間で互換性があるため、次のことを行うことができます。

- プラットフォーム間のコピーまたは伝達
- プラットフォームを選ばない SCCS および Forte TeamWare ユーティリティーによる処理

UNIX と Microsoft Windows 環境とで、テキスト編集ソフトウェアによる行の終わり (EOL) 区切り記号の表現方法が異なるため、両方の環境で共有されているテキストファイルを編集すると、予期しない結果になることがあります。この EOL の違いによる問題を解消するために、SCCS には、プレーンテキストファイルに対する EOL 変換機能があります。詳細は、245 ページの「改行コードの CR と LF の変換について」を参照してください。

ワークスペースの共有

Forte TeamWare ワークスペースは、プラットフォーム間で互換性があります。たとえば、Microsoft Windows 環境の Forte TeamWare のユーザーは、Solaris 環境の Forte TeamWare で作成されたワークスペースに対し、プットバックやブリングオーバーを行うことができます。これらの作業を行うには、Microsoft Windows システムから Solaris ファイルシステムにアクセスするためのファイル共有サービスが必要です。このサービスを提供する製品として、Samba だけが現在サポートされています。Samba は、GNU パブリックライセンスに基づいて無償で配布されている Microsoft Windows/UNIX ネットワーキング/ファイル共有ソフトウェアです。Samba についての詳細は、Samba の Web サイト (<http://us1.samba.org>) を参照してください。

論理パスとワークスペースメタファイルの問題

Forte TeamWare ワークスペースは、他のワークスペースと共有している親子関係の記録を保持します。この情報は、すべての親および子ワークスペースの位置情報であり、ワークスペース内のメタデータファイルに書き込まれます。異なるプラットフォームにおいて、Forte TeamWare を実行する複数のユーザー間でワークスペースを共有している場合、そのワークスペースへの論理パスは、ユーザーのシステムに基づくワークスペースのマウント方法によって異なることがあります。親および子ワークスペースのメタファイルでエントリが認識されないことが原因で問題が発生した場合は、関係するワークスペースの親子関係を変更することで問題を解決できます。ワークスペース管理ツールで「ワークスペース」▶「親の変更」を選択するか、次のコマンドを使用します。

```
workspace parent -p <新しい親>
```

Solaris ユーザーとのファイルの共有

Forte TeamWare は、PC-NFS™ ソフトウェアを使用してマウントされたファイルシステム上に存在する統合ワークスペースを正しく処理できません。SCCS では、書き込み権を持つユーザーは、読み取り専用ファイルを削除できると見なされます。PC-NFS では、これが不可能になり、このため SCCS および Forte TeamWare が誤作動します。この問題を避けるには、代替のファイル共有サービスを使用する必要があります。

たとえば、Samba ファイルサーバーを正しく設定することによって、Microsoft Windows から UNIX ファイルシステムにアクセスできるようになります。

Samba ファイルサーバーの構成

Samba ファイルサーバーは、GNU General Public License に基づいて無償で提供されているソフトウェアです。このソフトウェアの詳細とダウンロード情報については、Samba の Web サイト (<http://us1.samba.org>) を参照してください。

Microsoft Windows 上で Samba が Forte TeamWare と連携して正しく機能するようにするには、Samba 設定ファイルの `smb.conf` をエディタで開き、表 C-4 に示すようにパラメータを設定します。

表 C-4 Samba 設定ファイルのパラメータ設定

パラメータグループ	パラメータ名	設定
[global]	<code>preserve case</code>	<code>yes</code>
	<code>short preserve case</code>	<code>yes</code>
	<code>case sensitive</code>	<code>yes</code>
	<code>delete readonly</code>	<code>yes</code>
	<code>create mask</code>	<code>777</code>
	<code>directory mask</code>	<code>777</code>
	<code>encrypt passwords</code>	<code>yes</code>
[<i>export-directory-name</i>]	<code>comment</code>	<code>export-directory-name</code>
	<code>path</code>	<code>/export</code>
	<code>writeable</code>	<code>yes</code>
	<code>public</code>	<code>no</code>
	<code>create mask</code>	<code>777</code>

Windows NT Service Pack 3 または互換オペレーティングシステムでは、パスワードの暗号化が有効になります。パスワードの暗号化に対する Samba のサポートは、バージョン 1.9.18 または互換バージョンで提供されています。詳細は、Samba の Web サイトの FAQ (<http://us1.samba.org/samba/docs/FAQ>) を参照してください。

Microsoft Visual Studio のプラグインとしての Forte TeamWare の使用

Microsoft Windows 環境に Forte TeamWare をインストールすると、Forte TeamWare は自動的に、Microsoft Visual Studio のデフォルトの SCC (ソースコード管理) システムとして機能します。Visual Studio プロジェクトを Forte TeamWare SCC の下に置くことができます。また、「プロジェクト」▶「ソース管理」メニューから基本的な Forte TeamWare コマンドにアクセスできます。

Microsoft Visual Studio プロジェクトを Forte TeamWare の SCC 管理下に置く

Microsoft Visual Studio プロジェクトとその構成ファイルのすべてを Forte TeamWare SCC 管理下に置く場合は、以下のようにします。

1. Microsoft Visual Studio の「ワークスペース」ウィンドウでプロジェクトを選択します。
2. 「プロジェクト」▶「ソース管理」メニューから「ソース管理へ追加」を選択します。

Microsoft Visual Studio からの Forte TeamWare コマンドの使用方法

Microsoft Visual Studio のプロジェクトを Forte TeamWare の SCC 管理下に置くと、「プロジェクト」▶「ソース管理」メニューのコマンドを使用して Forte TeamWare の基本機能を利用できるようになります。

「ソース管理」メニューの項目は、このマニュアルの他の場所で説明している Forte TeamWare のコマンドに対応しています。表 C-5 は、「ソース管理」メニューの項目と Forte TeamWare コマンドの対応をまとめたものです。このマニュアルでの各コマンドの説明箇所も示しています。

表 C-5 「Source Control」メニューの項目と Forte TeamWare コマンド

「ソース管理」 メニューの項目	Forte TeamWare コマンド	記載箇所
最新バージョンを取得	ブリングオーバー	22 ページの「ワークスペース間の変更内容の伝達」
チェックアウト	チェックアウト	73 ページの「ファイルのチェックアウト」
チェックイン	チェックイン	76 ページの「ファイルのチェックイン」
チェックアウトの取り消し	アンチェックアウト	78 ページの「ファイルに加えた変更の取り消し」
ソース管理へ追加	新たにチェックイン	72 ページの「ワークスペースへのファイルの追加」
	ワークスペース作成	19 ページの「子ワークスペースの作成」
ソース管理から削除	ワークスペースの削除	29 ページの「ワークスペースの削除と設定の解除」
履歴の表示	ファイル履歴	102 ページの「ファイルの履歴の表示」

表 C-5 「Source Control」メニューの項目と Forte TeamWare コマンド (続き)

「ソース管理」 メニューの項目	Forte TeamWare コマンド	記載箇所
相違点の表示	最新の差分	「ファイルマージ」ウィンドウを開き、現在のファイルの相違を表示します。詳細は、86 ページの「「ファイルマージ」ウィンドウのグリフについて」を参照してください。
Forte-SCCS からの共有	Microsoft Visual Studio ワークスペースには含まれていない、Forte TeamWare の SCC 管理下にあるファイルに、Visual Studio SCC 環境内からアクセスできます。 注 : Forte TeamWare には、このコマンドに相当するコマンドはありません。	
最新表示	「表示」 ▶ 「再描画」 または「更新」	ファイルの現在の状態が正確に GUI に反映されるようにします。詳細は、オンラインヘルプを参照してください。

用語集

Codemgr_wsdata ディレクトリ

Forte 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は、ワークスペースに関する情報をこの Codemgr_wsdata ディレクトリに格納します。ディレクトリが通常のディレクトリではなく、Forte TeamWare ワークスペースであるかどうかは、唯一、このディレクトリが存在するかどうかによって決まります。ワークスペース管理コマンドは、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。175 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 プログラム

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」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。プリングオーバーおよびプットバックトランザクションでワークスペース管理が使用するファイルリストを生成するプログラムまたはスクリプトです。「def.dir.flp プログラム」も参照してください。

g-file (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 の最初の部分はリリース番号、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と、レベル番号が「1」大きくなります。

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、SCCS ディレクトリは、`g-file` と同じディレクトリに置かれます。「[g-file \(SCCS 用語\)](#)」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に戻すこと。

親の変更

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 トランザクション

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、更新ブリングオーバートランザクションは、親から子にファイルを「取り込み」ます。「作成ブリングオーバー トランザクション」「ワークスペース」「プットバック」も参照してください。

コピー/変更/マージ モデル

並行して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。複数の開発者は、このモデルに基づいて、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(プットバック)。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。

作成

ワークスペース管理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プ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 トランザクション

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すべての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。したがって、作成ブリングオーバートランザクションは、親から子にファイルを「取り込み」ます。「更新ブリングオーバー トランザクション」「ワークスペース」「プットバック」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です。「マージ」「衝突」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。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」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。

取り消し

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。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」も参照してください。

バージョン管理ツール

Forte TeamWare ファイルを管理するための ファイアウォール ツール。第 5 章も参照してください。

ファイルマージツール

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 **Forte TeamWare** ツール。第 6 章を参照してください。

プットバック

子ワークスペース上に存在するファイルで親ワークスペースを更新するとき使用するトランザクション。ワークスペース管理のあらゆる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、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「戻し」ます。「更新ブリングオーバー トランザクション」「作成ブリングオーバー トランザクション」「ワークスペース」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。

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 **Forte TeamWare** ツール。第 8 章を参照してください。

分岐 (SCCS 用語)

SCCS 履歴ファイル内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。

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.file (エスドットファイル) とも呼ばれます。

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。

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ブリングオーバーおよびプ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コマンドは、ワークスペース内での処理を開始する前に、このファイルを参照します。

ワークスペース

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、それらディレクトリに含まれるすべてのファイルのことです。Forte TeamWare が操作できるように、ワークスペース内のファイルは Forte TeamWare と SCCS の管理下に置かれています。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を持ち、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ツールには、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。「親ワークスペース」「子ワークスペース」も参照してください。

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
ツール

Forte 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

-, 87
+, 87
|, 87

数字

2000 年問題 (Y2K) 対応, 236

A

access_control ファイル, 176, 179
args ファイル, 176

B

backup サブディレクトリ, 176

C

children ファイル, 176
CLASSPATH 環境変数, 243
CodeManager ツール
 codemgr コマンド, 150
 「ワークスペース管理」を参照
CODEMGR_DIR_FLP, 41
CODEMGR_PATH, 52

CODEMGR_PATHNAME_STYLE, 244
CODEMGR_PATH_ONLY, 53
.codemgr_resrc ファイル, 178
.codemgrtoolrc ファイル, 178
CODEMGR_WS, 53, 125
Codemgr_wsdata サブディレクトリ
 access_control ファイル, 176, 179
 args ファイル, 176
 backup サブディレクトリ, 176
 children ファイル, 176
 conflicts ファイル, 176
 description ファイル, 176
 freezepoints サブディレクトリ, 177
 history ファイル, 177
 locks ファイル, 177
 nametable ファイル, 177
 notification ファイル, 178
 parent ファイル, 178
 putback.cmt ファイル, 118, 178
 関連する問題, 232
codemgr コマンド, 150
conflicts ファイル, 176
Cygnus ツール, 244

D

def.dir.flp ファイル, 40
description ファイル, 176

diffs コマンド, 102

dmake

-j オプション, 147

-m オプション, 147

入れ子呼び出し, 145

基本概念, 135

コマンド, 145

ホスト, 146, 147

メイクファイルに与える影響, 140

理解, 140

dmake.conf ファイル, 136, 139

.dmakerc ファイル, 136

E

EOL 記号, 231, 245

F

Forte for Java

TeamWare ツールバー, 167

エクスプローラウィンドウ, 168

バージョン管理関係のコマンド, 164

ファイルマージ関係のコマンド, 163

フリーズポイント関係のコマンド, 164

メニューバー, 167

ワークスペース管理関係のコマンド, 161

FreezePoints サブディレクトリ, 177

freezept コマンド, 153

H

history

ファイル, 177

L

locks ファイル, 177

M

Microsoft Windows

NT SP5, 233

ファイルの共有, 247

MKS ツール, 244

MR (変更要求), 118

N

nametable ファイル, 177

NoClassDefFoundError, 230

.NO_PARALLEL

ターゲット, 145

notification ファイル, 178

P

parent ファイル, 178

PATH 環境変数, 229, 243

PCNFS, 247

putback.cmt ファイル, 118, 178

R

RCS (Revision Control System), 54

rcs2ws、使用方法, 54

RCS プロジェクトの TeamWare 形式への変換, 54

RTI (Request To Integrate), 61

S

Samba, 248

SCCS

diffs サブコマンド、問題, 236

s ドットファイル, 180

Y2K 問題対応, 236

エラー, 236

使用すべきでないコマンド, 234

定義, 4

デルタ ID (SID), 105

- ファイルの属性, 117
- マスターファイル, 232
- 履歴ファイル, 180, 235
- 履歴ファイル、共有, 246
- 履歴ファイルとファイルマージ, 175
- 履歴ファイルと分岐, 186
- 履歴ファイルとマージ, 187
- 履歴ファイルの妥当性検査, 235
- 履歴ファイルの復元, 235
- SCCS get 抑制オプション, 37
- sccs-admin コマンド, 117
- SCCS_GET_TRANS_LF, 231, 237, 245
- SCCS_NO_NL_TERMINATE, 246
- SCCS デルタ ID (SID)
 - SCCS マージ可能 ID (SMID) への変換, 194
 - SCCS マージ可能 ID (SMID), 133
- SCCS マージ可能 ID (SMID)
 - SCCS デルタ ID (SID), 133
 - SCCS デルタ ID (SID) への変換, 193, 194
 - 必要な理由, 193
- SENDMAILHOST, 229, 233, 243
- Shift + クリック, 153
- SID
 - SMID への変換, 194
 - 定義, 105
- s ドットファイル, 180

T

TeamWare

- ツール, 9
- ワークフロー, 8

TeamWare の設定, 6

- 「TeamWare」メニュー
 - バージョン管理, 71
 - フリーズポイント, 123
 - ワークスペース管理, 16

- twconfig コマンド, 15

- twmerge コマンド, 152

- twversion コマンド, 151

U

- UNIX 構文, 244

W

- .WAIT ターゲット, 142

- workspace descr コマンド, 43

X

- X で消されたデルタ, 104

Z

- .z 拡張子, 132

あ

- アーカイブライブラリ, 143

- 赤色のチェックマーク, 104

- 「アクション」メニュー (ワークスペース管理), 16

- アクセス権, 58

- アクセスの制御、ワークスペース値, 180

- 設定, 57

- デフォルトのアクセス権, 179

- アンチェックアウト, 78, 89

い

- 依存関係リスト、順序, 142

- 「一般」タブ, 116

移動

- ファイル, 107

- ワークスペース, 28

え

- エディタ
 - デフォルト, 116
 - 変更, 76
- エラーメッセージ
 - フリーズポイント, 123
 - ワークスペース管理, 197

お

- オプション
 - 相違解決 (ファイルマージ), 92
 - ファイルマージ, 92
 - ブリングオーバー/プットバック, 35
 - ワークスペース管理, 49
- 「オプション」メニュー (ワークスペース管理), 16
- 親ワークスペース
 - 作成, 16
 - 定義, 3

か

- 改行コード, 245
- 改行の変換, 237, 245
- 改行文字, 237, 246
- 解除
 - ワークスペースのロック, 67
- 環境変数
 - CLASSPATH, 243
 - CODEMGR_DIR_FLP, 41
 - CODEMGR_PATH, 52
 - CODEMGR_PATHNAME_STYLE, 244
 - CODEMGR_PATH_ONLY, 53
 - CODEMGR_WS, 53, 125
 - PATH, 229, 243
 - SCCS_GET_TRANS_LF, 231, 237, 245
 - SENDMAILHOST, 229, 233, 243
 - パス名構文の設定, 244
- 監視
 - プロセス, 233

き

- キーボードショートカット, 154
- 既存のファイル、ワークスペースの作成, 17
- 起動
 - コマンド行からのバージョン管理, 151
 - コマンド行からのファイルマージ, 151
 - バージョン管理, 69, 155
 - フリーズポイント, 122
 - ワークスペース管理, 13
- 強制衝突オプション, 37

く

- 空白
 - 空白行と余分な空白, 237
 - ファイル名内の, 228
 - ワークスペース名内の, 231
- グリフ、意味 (ファイルマージ), 86, 98
- グループレベルのアクセス権, 60

け

- 警告メッセージ、ワークスペース管理, 219
- 現在の相違 (ファイルマージ), 90

こ

- 更新
 - フリーズポイントファイル, 126
 - ブリングオーバーおよびプットバック用ファイルリスト, 39
- 更新ブリングオーバー
 - タブ, 23
 - トランザクション, 22, 23
 - ファイルリスト, 39
 - ワークスペースの更新, 101
- 構築サーバー, 146
- 構築サーバーの構成ファイル, 136
- コピー / 変更 / マージ, 2
- コマンド, 149 ~ 156

- freezept, 153
 - twconfig, 15
 - twmerge, 152
 - twversion, 151
 - 使用すべきでない SCCS コマンド, 234
 - マニュアルページの表示, 150
 - コマンド行インタフェース, 149~156
 - Microsoft Windows におけるアクセス, 244
 - ファイルマージ, 151
 - フリーズポイント, 153
 - ワークスペース管理, 150
 - 「コマンド」メニュー (バージョン管理), 71
 - コメント
 - チェックイン, 77
 - チェックインのデフォルト, 93
 - デルタ, 37
 - デルタのマージ, 106
 - 電子メールのトランザクションコメント, 41
 - プットバック, 26, 79
 - ブリングオーバー / プットバックの警告, 38
 - 子ワークスペース
 - 親の変更メニュー項目による親の変更, 45
 - 親の変更例, 46
 - 親を変更する理由, 44
 - 作成, 19
 - 定義, 3
 - ドラッグ & ドロップによる親の変更, 46
- さ
- サーバーの設定と Samba, 248
 - 最後の改行文字, 246
 - 削除
 - 移動, 28
 - ファイル, 107
 - ワークスペース, 29
 - 作成
 - 親ワークスペース, 16
 - カスタマイズメニュー (バージョン管理), 112
 - 空のワークスペース, 17
 - 子ワークスペース, 19
 - ファイル階層からのワークスペースの作成, 17
 - フリーズポイントファイル, 120, 125
 - 作成ブリングオーバー
 - タブ, 20
 - トランザクション, 20
- し
- 実行時構成ファイル, 136
 - 実線, 104
 - 自動作成
 - フリーズポイントファイル, 131
 - 自動的な作成
 - ブリングオーバーおよびプットバック用ファイルリスト, 39
 - 自動マージ, 91
 - 詳細メッセージオプション, 37
 - 衝突
 - 定義, 83
 - ファイルマージ, 84
 - 衝突解決トランザクション, 83, 155
 - SCCS 履歴ファイルのマージ, 187
 - 子ワークスペース内の衝突, 182
 - 衝突状態のファイルのマージ, 84
 - 使用にあたって, 6~11
 - ショートカット
 - キーボード, 154
 - マウス, 154
 - 白抜きのグリフ, 87
 - 白抜きのフォント, 87
 - シンボル
 - ファイルの履歴, 104
 - ファイルマージのグリフ, 86
- せ
- 正符号 (+), 87
 - 設定、Samba ファイルサーバー, 248
 - 全体レベルのアクセス権, 58

そ

- 相違 (ファイルマージ)
 - 解決, 89
 - 解決済み, 89
 - 現在の, 90
 - 次の, 90
 - 前の, 90
- 相違の解決, 89
- 相互互換性, 246
- ソースコード管理システム (SCCS)
- ソースワークスペース, 121
- 祖先ファイル (ファイルマージ), 88
 - 起動時に読み込む, 152

た

- ターゲット
 - 複数, 144
 - 並列構築, 141, 142
- 縦棒 (|), 87
- 妥当性検査, 61
- ダブルクリック, 51, 116, 154

ち

- 抽出
 - 抽出元の階層, 121, 127
 - フリーズポイントファイル, 127
 - フリーズポイントファイルからのデルタ, 121, 127
- 抽出元の階層
 - 再作成, 127
 - 抽出, 127

つ

- 追加 (マウント)
 - 「読み込み」メニューへのパス (バージョン管理), 114
 - ワークスペースに対するファイル, 72

- 通知、電子メール, 41
 - プットバックのコメントの入力, 79
- ツール、一覧, 9
 - 「ツール属性」ダイアログボックス, 49
 - 「衝突解決」タブ, 92
 - 「ワークスペース管理」タブ, 49
- 次の相違 (ファイルマージ), 90

て

- ディスク領域
 - プットバック用の空きの確認, 228
- テキストファイル, 246
- デフォルトのエディタ, 76, 116
- デフォルトのファイルリスト, 39
- デルタ
 - コメントオプション, 37
 - 定義, 4, 105
 - フリーズポイントファイル内の, 133
 - 履歴表示, 102
- デルタのマージ, 106
- 電子メールによる通知, 41

と

- 統合要求 ID, 64
- ドライブ名, 244
- ドラッグ & ドロップ, 154
 - ワークスペースのアイコン, 46
 - ワークスペースの親の変更, 45
- トランザクション
 - 更新ブリングオーバー, 22, 23
 - 作成ブリングオーバー, 20
 - 衝突解決, 83, 155, 182
 - 制御, 57
 - 取り消し (ファイルマージ), 92
 - 取り消し (ワークスペース管理), 27
 - プットバック, 25
 - 他のユーザーへの通知, 41
- 「トランザクション出力」ウィンドウ, 22

取り消しトランザクション, 27

な

名前

履歴, 109

ワークスペース, 42

名前の変更

ファイル, 107

ワークスペース, 28

名前変更の衝突, 109, 110

に

二重下線, 104

ね

ネットグループ, 60

は

バージョン、Forte TeamWare の妥当性検査, 240

バージョン管理

Forte for Java からのコマンドへのアクセス方法, 164

カスタマイズメニュー, 112

コマンド行からの起動, 151

ダブルクリックで自動的に起動, 155

バージョンの確認方法, 240

バージョンの定義, 106

パス名構文, 244

波線, 104

バックアップしないオプション, 37

バッチファイル, 242

ひ

表示

diffs コマンド, 102

SCCS 管理下でないファイル, 72

デルタ間の相違, 115

ファイルの履歴, 102

ワークスペースの履歴, 30

「表示」メニュー (バージョン管理), 71

「表示」メニュー (ワークスペース管理), 16

「表示」メニュー (フリーズポイント), 123

ふ

ファイル

access_control, 176, 179

args, 176

children, 176

.codemgr_resrc, 178

.codemgrtoolrc, 178

conflicts, 176

description, 176

dmake.conf, 136, 139

.dmaker, 136

history ファイル, 177

locks, 177

nametable, 177

notification, 178

parent, 178

putback.cmt, 118, 178

SCCS の属性, 117

.z 拡張子, 132

アンチェックアウト, 78

移動, 107

起動時に 2 つ読み込む場合 (ファイルマージ), 152

起動時に 3 つ読み込む場合 (ファイルマージ), 152

構築サーバーの構成, 136

再読み込み (ファイルマージ), 92

削除, 107, 111

実行時構成, 136

衝突していない (ファイルマージ), 181

衝突している (ファイルマージ), 182

祖先, 88

チェックアウト, 73

独自のプリンクオーバーとプットバック用リ

- スト, 39
- 名前の変更, 107
- 名前の履歴, 109
- バージョン, 106
- ファイルのチェックイン, 76, 78
- フリーズポイント, 120
 - 更新, 126
 - 作成, 120, 125
 - 自動作成, 131
 - 抽出, 127
 - 定義, 133
- 読み込み (バージョン管理), 114
- 履歴, 102
- ファイル共有, 247
- ファイルサーバーの設定, 248
- ファイルの共有, 247
- ファイルの最後の改行文字, 237
- ファイルのチェックアウト, 73
- ファイルの並行修正, 143
- ファイルの履歴
 - シンボル, 104
 - 表示, 102
- ファイル変更の統合, 79
- ファイルマージ
 - Forte for Java からのコマンドへのアクセス方法, 163
 - SCCS 履歴ファイル, 182
 - オプション, 92
 - コマンド行インタフェース, 151
 - 自動, 91
 - 衝突していないファイル, 181
 - 衝突しているファイル, 182
 - ツール, 242
 - 変更の取り消し, 92
- 「ファイルマージ」ウィンドウの色, 86
- ファイルマージ例, 95 ~ 100
- ファイル名構文, 244
 - 「ファイル」メニュー
 - バージョン管理, 71
 - フリーズポイント, 123
 - ワークスペース管理, 16
- ファイルリストプログラム (FLP), 40
 - 例, 40
 - 「ファイルを開く」ダイアログボックス (ファイルマージ), 87
- 複数のターゲット, 144
- 複数ファイルの一括選択, 153
- プットバック
 - オプション, 35
 - コメント, 26
 - タブ, 25
 - トランザクション, 25
 - ファイルリスト, 39
 - 複数の同時ユーザーの問題, 232
 - プレビュー, 37, 228
- プットバックの妥当性検査, 61
 - サンプルプログラム, 64
 - 有効にする, 63
- 負符号 (-), 87
- フリーズポイント
 - Forte for Java からのコマンドへのアクセス方法, 164
 - 起動, 122
 - コマンド行インタフェース, 153
 - フリーズポイントにおける作成, 120
 - フリーズポイントにおける抽出, 121
- フリーズポイント、定義, 119
- 「フリーズポイント」ウィンドウ
 - 作成モード, 123
 - 抽出モード, 129
- フリーズポイントツール, 242
- フリーズポイントファイル, 120
 - 更新, 126
 - 作成, 120, 125
 - 自動フリーズポイント, 131
 - 抽出, 127
 - 定義, 133
 - 内容, 133
- ブリングオーバーとプットバックのオプション
 - SCCS get 抑制, 37
 - 強制衝突, 37
 - 詳細メッセージ, 37
 - 設定, 35

- デルタコメント, 37
- バックアップしない, 37
- プレビュー, 37, 38
- メッセージ抑制, 37, 38
- プレビューオプション, 37
- プロジェクト、RCS から TeamWare への変換, 54
- プロセス監視, 233
- 分岐, 44, 105
 - SCCS 履歴ファイル, 186

へ

- 並列処理, 144
- 並列処理に対する制限, 144
- 「ヘルプ」メニュー
 - バージョン管理, 71
 - フリーズポイント, 124
 - ワークスペース管理, 16
- 変更
 - デフォルトのエディタ, 76
 - ワークスペースの履歴表示, 31
- 変更の伝達, 22
- 変更の取り消し
 - ファイル, 78
 - ファイルマージ, 92
 - ワークスペース, 27
- 変更要求 (MR), 118

ま

- マージ版 (ファイルマージ), 99
- マージ版の編集 (ファイルマージ), 99
- マウスショートカット, 51, 154
- マウスの右ボタン, 154, 155
- マウント、「追加」を参照
- 前の相違 (ファイルマージ), 90
- マクロ
 - 動的, 143
- マニュアルページ, xvi

め

- メークファイル
 - dmake ユーティリティが与える影響, 140
 - 制限事項, 141
 - テンプレート、使用, 140
 - ファイルの衝突, 144
- メッセージ抑制, 37
- メニューのカスタマイズ (バージョン管理), 112

や

- 矢印, 104

ゆ

- ユーザーレベルのアクセス権, 60

よ

- 読み込み (「追加」を参照)
 - 「ファイルマージ」ウィンドウからのファイル, 87
 - ファイルマージ起動時に 2 つまたは 3 つ読み込む場合, 152
- 「読み込み」メニュー (バージョン管理), 114

ら

- ライブラリの更新、同時, 143

り

- 履歴
 - SCCS ファイルの共有, 246
 - ファイル, 180
 - ワークスペース, 30

れ

- 例

検索パスの設定, 53
コピー / 変更 / マージ, 5
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, 53
チームプロジェクト, 6
名前変更の衝突, 109
ファイルマージ, 95 ~ 100
ファイル名の変更とファイルの移動, 107
プットバックの妥当性検査プログラム, 64
ワークスペースの親の変更, 46
ワークスペースの自動読み込み, 52

ろ

ロック
ワークスペースのロックの解除, 67
ロックの編集, 67
論理パスとワークスペースメタファイル、問
題, 247

わ

ワークスペース
アクセス, 57
アクセスの制御, 179, 180
移動, 28
イベント通知, 41
親, 3, 16
親の変更, 43, 46, 247
親を変更する理由, 44
空の, 17
子, 3, 19
互換性, 239
削除, 29
衝突, 82, 83
設定解除, 29
ソース, 121
定義, 13
名前, 42
名前の変更, 28
破壊, 239

ファイル階層からの作成, 17
ファイルの追加, 72
プットバックの妥当性検査, 61
変更の取り消し, 27
メタデータディレクトリ
(Codemgr_wsdata), 175
メタファイルと論理パスの問題, 247
履歴の検索, 34
履歴の表示, 30
ロック, 67
ロックの解除, 67
ワークスペース管理、定義, 175
分かりやすい名前, 42
ワークスペース、共有, 247
ワークスペース管理
Forte for Java からのコマンドへのアクセス方
法, 161
エラーメッセージ, 197
カスタマイズ, 49
警告メッセージ, 219
コマンド行インタフェース, 150
属性, 49
ワークスペース管理ツールの起動, 13
「ワークスペース管理」ウィンドウ, 15
ワークスペース管理のカスタマイズ, 49
ワークスペース管理のワークスペース、定
義, 175
ワークスペース設定の解除, 29
ワークスペースの親の変更, 43
一時的な, 46
例, 46
論理パスとワークスペースメタファイルの問
題, 247
ワークスペースの共有, 247
ワークスペースの履歴表示
起動, 30
情報の選別, 33
ワークスペースの履歴表示情報の選別, 33
ワークスペースの履歴表示情報のソート, 33
ワークスペースへのアクセス権, 57
ワークスペースへのアクセスの制御, 57

ワークスペース名の表示, 43
「ワークスペース」メニュー
バージョン管理, 71
ワークスペース管理, 16
ワークスペース履歴の検索, 34
分かりやすいワークスペース名, 42

