

# Rational™ ClearCase® Rational™ ClearCase® LT

## 管理ガイド

バージョン: 2003.06.10 およびそれ以降

G126-5400-00

UNIX/WINDOWS 版



## 法的通知

Copyright © 1992-2003, Rational Software Corporation. All Rights Reserved.

バージョン番号: 2003.06.10 およびそれ以降

本マニュアル (「本著作物」) は、アメリカ合衆国その他の国々の著作権法及び種々の条約により保護されています。Rational Software Corporation の文書による事前の同意を得ることなく本著作物を複製し又は頒布することは、禁じられています。

本著作物はライセンスに基づいて提供されるもので、ライセンス規定に従う場合にのみ、使用または複製できます。ライセンスで明示的に許可されている場合を除き、本マニュアルまたはその複製を第三者に提供することは禁じられています。本著作物の権利または所有権を譲渡することはできません。ライセンス条項の全文については、ライセンス契約書をお読みください。

Rational Software Corporation、Rational、Rational Suite、Rational Suite ContentStudio、Rational Apex、Rational Process Workbench、Rational Rose、Rational Summit、Rational Unified process、Rational Visual Test、AnalystStudio、ClearCase、ClearCase Attache、ClearCase MultiSite、ClearDDTS、ClearGuide、ClearQuest、PerformanceStudio、PureCoverage、Purify、Quantify、Requisite、RequisitePro、RUP、SiteCheck、SiteLoad、SoDa、TestFactory、TestFoundation、TestMate、TestStudio は、Rational Software Corporation の米国およびその他の国における登録商標です。Rational のロゴ、Connexis、ObjecTime、Rational Developer Network、RDN、ScriptAssure、XDE は、Rational Software Corporation の米国およびその他の国における商標です。その他すべての名前は、識別の目的でのみ使用されているものであり、それぞれの会社の商標または登録商標です。

米国特許番号 5,193,180、5,335,344、5,535,329、5,574,898、5,649,200、5,675,802、5,754,760、5,835,701、6,049,666、6,126,329、6,167,534、6,206,584 の請求の範囲内の部分。このほかにも米国特許及び国際特許申請中。

## 米国政府の権利

このソフトウェアおよび文書は、「商業的コンピュータソフトウェア」、「商業的ソフトウェア」または「使用が制限されたコンピュータソフトウェア」として提供され、規約は該当する DFARS 252.227、DFARS 252.211、FAR 2.101、FAR 52.227 ( またそれ以前に定められた条項 ) に規定されています。本ソフトウェア製品およびドキュメントの使用、複製、または開示は、DFARS 227.7202、FAR 52.227-19 の下位条項 (c)、または FAR 52.227-14 ( またはその改訂された規定 ) に定められるように、該当する Rational Software Corporation ライセンス契約書の条項の制約を受けます。

## 免責事項

本書および関連ソフトウェアは、ライセンス契約に基づいて使用することができます。そのような使用許諾契約書に別段の明示的な規定がある場合を除き、また、それぞれの国の法律により禁止または制限されている場合を除き、Rational Software Corporation は、本メディア、ソフトウェア製品、およびその関連文書について、明示的にも暗黙的にも、商品性に関する保証、非権利侵害性に関する保証、特定目的への適合性に関する保証、取り扱い、使用、または取引行為に伴う保証、およびライセンシーによる静穏無事な製品使用に対する妨害がないことの保証について一切の責任を負いません。

### 第三者の通知、コード、使用許諾および確認

Portions Copyright © 1992-1999, Summit Software Company. All rights reserved.

Microsoft、Microsoft のロゴ、Active Accessibility、Active Client、Active Desktop、Active Directory、ActiveMovie、Active Platform、ActiveStore、ActiveSync、ActiveX、Ask Maxwell、Authenticode、AutoSum、BackOffice、BackOffice のロゴ、bCentral、BizTalk、Bookshelf、ClearType、CodeView、DataTips、Developer Studio、Direct3D、DirectAnimation、DirectDraw、DirectInput、DirectX、DirectXJ、DoubleSpace、DriveSpace、FrontPage、Funstone、Genuine Microsoft Products のロゴ、IntelliEye、IntelliEye のロゴ、IntelliMirror、IntelliSense、J/Direct、JScript、LineShare、Liquid Motion、Mapbase、MapManager、MapPoint、MapVision、Microsoft Agent のロゴ、Microsoft eMbedded Visual Tools のロゴ、Microsoft Internet Explorer のロゴ、Microsoft Office Compatible のロゴ、Microsoft Press、Microsoft Press のロゴ、Microsoft QuickBasic、MS-DOS、MSDN、NetMeeting、NetShow、Office のロゴ、Outlook、PhotoDraw、PivotChart、PivotTable、PowerPoint、QuickAssembler、QuickShelf、RelayOne、Rushmore、SharePoint、SourceSafe、TipWizard、V-Chat、VideoFlash、Visual Basic、Visual Basic のロゴ、Visual C++、Visual C#、Visual FoxPro、Visual InterDev、Visual J++、Visual SourceSafe、Visual Studio、Visual Studio のロゴ、Vizact、WebBot、WebPIP、Win32、Win32s、Win64、Windows、Windows CE のロゴ、Windows のロゴ、Windows NT、Windows Start のロゴ、XENIX は、Microsoft Corporation 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。

Sun、Sun Microsystems、Sun のロゴ、Ultra、AnswerBook 2、medialib、OpenBoot、Solaris、Java、Java 3D、ShowMe TV、SunForum、SunVTS、SunFDDI、StarOffice、および SunPCi は、Sun Microsystems の米国および他の国における商標または登録商標です。

Purify は、Sun Microsystems, Inc. の米国特許番号 5,404,499 の下にライセンス供与されています。

Globetrotter ソフトウェア (FLEXIm ライブラリおよびユーティリティ) の本来の用途は、ソフトウェアライセンス管理であり、他の製品またはアプリケーションにこれらのソフトウェアを組み込むことは、ライセンスに含まれません。

BasicScript は、Summit Software Company の登録商標です。

デザイン パターン : Erich Gamma、Richard Helm、Ralph Johnson および John Vlissides による再使用可能なオブジェクト指向のソフトウェアの要素。Copyright © 1995 by Addison-Wesley Publishing Company, Inc. All rights reserved.

Copyright © 1997 OpenLink Software, Inc. All rights reserved.

本ソフトウェアおよびドキュメントは部分的に、カリフォルニア大学理事会により使用許諾されている BSD Networking Software Release 2 に基づいています。当社はその開発におけるカリフォルニア大学バークレー校の Computer Systems Research Group および Electrical Engineering and Computer Sciences Department 並びに「その他の協力者」の役割を認めます。

本製品は、Apache ([http://www.webdav.org/mod\\_dav/](http://www.webdav.org/mod_dav/)) における使用のために Greg Stein により開発されたソフトウェアを含んでいます。

追加の法的通知は、お客様の Rational ソフトウェア インストレーションに含まれています。

# 目次

<b>まえがき</b> .....	<b>xxix</b>
本書について .....	xxix
ClearCase マニュアル ロードマップ .....	xxx
ClearCase LT マニュアル ロードマップ .....	xxxi
ClearCase とほかの Rational 製品との統合 .....	xxxii
表記規則 .....	xxxiv
オンライン マニュアル .....	xxxv
カスタマ サポート .....	xxxvi
 <b>ClearCase ネットワークの計画と管理</b> .....	 <b>1</b>
管理責任の概要 .....	1
ClearCase 製品ファミリ .....	2
ClearCase と ClearCase LT の違い .....	2
ホストとホスト管理 .....	3
ネットワークワイドなりリリース ホスト .....	3
ClearCase LT ホスト .....	3
ClearCase ホスト .....	4
ライセンス方式とライセンス管理 .....	5
データ記憶領域とデータ管理 .....	5
ClearCase レジストリ .....	6
VOB .....	6
ビュー .....	7
ユーザーとグループの管理 .....	8
ClearCase サーバー プロセス .....	8
albd_server .....	9
ポート割り当て .....	10
admin_server .....	10
credmap_server .....	10
view_server .....	11
vob_server .....	11
db_server .....	11
vobrpc_server .....	12
lockmgr .....	12
rwp .....	12

サーバー ログ	13
UNIX でのログ	13
Windows でのログ	14
ClearCase の開始と停止	14
UNIX 環境の場合	14
Windows 環境の場合	15
MVFS の開始と停止	15
ClearCase と NAS (Network-Attached Storage) デバイス	16
NAS デバイスへのネットワーク アクセスの設定	17
DHCP を使用する場合	17
ClearCase ホストに複数のネットワーク インターフェイスがある場合	18
ClearCase と NFS 自動マウント機能	19
自動マウント機能のマップとマウント ポイント	19
非標準マウント ポイントの指定	19
非標準マウント ディレクトリの指定	20

## ライセンス方式とライセンス管理 ..... 21

ClearCase のライセンス方式と Rational Common Licensing	21
ClearCase のライセンス方式	21
ライセンス取得の優先順位	22
ライセンス レポート ユーティリティ	22
ライセンス サーバー ホストの指定	22
ライセンス サーバーの設定	23
ライセンス データベース	23
既存のライセンス サーバー ホストへの新規ライセンスの追加	24
追加ライセンス サーバー ホストの設定	24
別のホストへのライセンスの移動	26
ライセンス サーバー ホスト名の変更	27
ライセンス データベースの形式	27
ライセンス セットを定義する行	27
ユーザーの優先順位を指定する行	28
ライセンス取得の対象外ユーザーを指定する行	28
監視を有効化する行	28
タイムアウトを指定する行	28
ClearCase LT の Rational Common Licensing	29
フローティング ライセンスとノード ロック ライセンス	29
テンポラリー ライセンス、パーマネント ライセンス、期限付き使用承諾ライセンス	30

ClearCase LT と Rational Suite のライセンス .....	30
ライセンス サーバーのインストールと設定 .....	30
Windows クライアント ライセンスのインストールと設定 .....	31
License Key Administrator .....	31
UNIX クライアント ライセンスのインストールと設定 .....	31

## **ClearCase レジストリの管理 ..... 33**

ClearCase レジストリと ClearCase LT レジストリの違い .....	33
レジストリ データ .....	34
オブジェクト .....	34
タグ .....	35
VOB タグと VOB アクセス .....	36
ビュー タグとビュー アクセス .....	37
レジストリ リージョン .....	38
ほかの共有データ .....	41
サーバーの記憶場所 .....	41
サイトのデフォルト値 .....	42
レジストリ サーバー ホスト .....	42
ClearCase レジストリ ホストの定義 .....	43
UNIX コンピュータをレジストリ ホストとして定義するには .....	43
Windows コンピュータをレジストリ ホストとして定義するには .....	43
レジストリ パスワードの設定と使用 .....	44
レジストリ クライアント リスト .....	44
複数の ClearCase レジストリの使用に関するガイドライン .....	44
ClearCase ホストのレジストリとリージョンの割り当て .....	45
レジストリ データの管理 .....	46
オブジェクトとタグの表示 .....	48
VOB またはビューの登録 .....	49
タグの作成、削除、変更 .....	49
サイトのデフォルト値の変更 .....	50
ClearCase レジストリ リージョンの追加 .....	50
新しいレジストリ リージョンを作成するには .....	50
新しいレジストリ リージョンにホストを割り当てるには .....	51
新しいレジストリ リージョンにタグを作成するには .....	51
リージョンが存在する ClearCase レジストリでの VOB とビューの作成 .....	51
レジストリ リージョンの削除 .....	52
レジストリ サーバー ホスト名の変更 .....	52

レジストリ管理のガイドライン .....	52
ClearCase レジストリのバックアップとスイッチオーバー .....	54
バックアップ レジストリ ホストを定義するには .....	54
UNIX コンピュータをバックアップ レジストリ ホストとして定義するには .....	55
Windows コンピュータをバックアップ レジストリ ホストとして定義するには ..	55
バックアップ レジストリ ホストをプロモートしてプライマリ レジストリ ホストにするには .....	56
プライマリ レジストリ ホストを復元するには .....	56
レジストリ バックアップ用に設定されていないホストへのレジストリの移動 .....	57

## **混在環境での ClearCase ネットワーク管理 ..... 59**

共通なユーザー名とグループ名 .....	59
ClearCase によるユーザー名とグループ名の比較 .....	60
認証マッピング .....	60
ClearCase サーバー プロセス ユーザーに対する UNIX 認証 .....	60
異なるプラットフォーム間のファイル システムのアクセス .....	61
VOB とビューへのアクセスに使用されるネットワーク ファイル システム プロトコル .....	62
NAS と異なるプラットフォーム間のファイル アクセス ツール .....	63
ClearCase ファイル サービス .....	64
Windows 上での CCFS の有効化 .....	64
CCFS と Windows のリモート ビュー サーバー ホスト .....	65
大文字/小文字の区別 .....	65
一般的な推奨事項 .....	66
大文字/小文字の区別と MVFS .....	66
大文字と小文字を保存する MVFS の使用が推奨される状況 .....	67
スナップショットと Web ビューでの大文字/小文字の区別 .....	67
ビューのテキスト モードの構成 .....	67
テキスト モード .....	68
ビューのテキスト モードの特定 .....	69
ビューのテキスト モードの選択 .....	69
古い VOB での UNIX/Windows 相互運用テキスト モードのサポートの有効化 .....	70
VOB が UNIX/Windows 相互運用テキスト モードを サポートしているかどうかを特定するには .....	71
MultiSite ユーザー向けの特別な手順 .....	71
サポートされている文字セットの相違 .....	71

<b>VOB とビューのアクセス制御</b>	<b>73</b>
VOB とビューのアクセス コントロールの基本事項	73
ユーザーとグループ	73
プライマリ グループ	74
ユーザーが 33 以上のグループに所属している場合の制限	74
権限を持つユーザーとグループ	74
ClearCase サーバー プロセス ユーザー	75
リモート root ユーザーの制限付き権限	75
ユーザー プロセス	76
ClearCase オブジェクト	76
保護モード	77
VOB データとビュー データへのアクセス	78
VOB と VOB オブジェクトに対するアクセス コントロール	79
VOB に対するアクセス コントロール	80
VOB を作成する権限	80
VOB を削除する権限	80
VOB を読み取る権限	80
VOB に書き込む権限	80
VOB を実行する権限	81
エレメントに対するアクセス コントロール	81
エレメントを作成する権限	82
エレメントを削除する権限	82
エレメントを読み取る権限	82
エレメントに書き込む権限	82
エレメントを実行する権限	83
その他の VOB オブジェクトに対するアクセス コントロール	83
その他の VOB オブジェクトを作成する権限	84
その他の VOB オブジェクトを削除する権限	84
その他の VOB オブジェクトを読み取る権限	84
その他の VOB オブジェクトに書き込む権限	84
VOB オブジェクトのロック	85
タイプ オブジェクトのロック	85
ビューとビュー オブジェクトに対するアクセス コントロール	85
動的ビューに対するアクセス コントロール	86
ビューを作成する権限	87
ビューを削除する権限	87
ビューを読み取る権限	87

ビューを書き込む権限 .....	88
ビューを実行する権限 .....	88
ビュープライベート ファイルに対するアクセス コントロール .....	88
所有者、グループ、保護モードの初期値 (UNIX) .....	89
所有者、グループ、保護モードの初期値 (Windows) .....	89
ビュープライベート ファイルを作成する権限 .....	90
ビュープライベート ファイルを削除する権限 .....	90
ビュープライベート ファイルを読み取る権限 .....	90
ビュープライベート ファイルを書き込む権限 .....	91
ビュープライベート ファイルを実行する権限 .....	91
派生オブジェクトに対するアクセス コントロール .....	91
ClearCase とファイル システム固有の権限 .....	92

## **VOB 管理 ..... 93**

VOB と VOB 管理の概要 .....	93
VOB データとメタデータへのアクセス .....	93
VOB 属性 .....	94
VOB スキーマ バージョン .....	95
VOB スキーマ バージョンを変更するには .....	95
VOB 機能レベル .....	95
VOB 機能レベルを表示するには .....	96
VOB 機能レベルを変更するには .....	96
VOB サーバー ホスト構成のガイドライン .....	97
物理メモリ .....	97
ディスク容量 .....	98
プロセッサの処理能力 .....	98
ネットワークの接続性 .....	98
UNIX カーネル リソースの調整 .....	98
VOB の作成 .....	99
VOB 記憶領域へのアクセス要件 .....	99
VOB のサーバー記憶場所の作成 .....	99
NAS デバイス上でのサーバー記憶場所の作成 .....	100
VOB を VOB サーバー ホスト上に作成するには .....	101
VOB を NAS デバイス上に作成するには .....	102
VOB を リモート Windows ホスト上に作成するには .....	103
パブリック VOB とプライベート VOB .....	103
パブリック VOB を作成するには .....	104

管理 VOB への VOB のリンク .....	104
NAS デバイスに対応する VOB サーバー ホストの置換 .....	105
VOB アクセスに関するトラブルシューティング .....	106
正しくないグローバル パス情報 .....	106
UNIX の自動マウント機能でホスト マップを使用していない場合 .....	107
VOB のグループ所有権情報の調整 .....	107
UNIX での setuid 実行の有効化 .....	108
VOB へのアクセスの無効化 .....	108
VOB をロックするには .....	108
VOB タグを削除するには .....	108
VOB の削除 .....	109
VOB 記憶ディレクトリ .....	109
VOB 記憶プール .....	111
ソース記憶プール .....	111
cleartext 記憶プール .....	112
派生オブジェクト記憶プール .....	112
VOB データベース .....	112
保存されているデータベース サブディレクトリ .....	113
.identity ディレクトリ .....	114
VOB 記憶領域の管理 .....	114
ClearCase スケジューラの使用 .....	114
VOB 記憶領域の監視 .....	115
スクラブ .....	115
VOB からの不要なバージョンの削除 .....	116
lost+found ディレクトリ .....	116
VOB の追加記憶領域の作成 .....	117
追加記憶プールの作成 .....	118
UNIX ホストでのリモート記憶プールの作成 .....	118
エレメントのソース プール割り当ての変更 .....	119
例: ディレクトリ内のすべてのファイルを新しいプールに割り当てる .....	119
例: 既存の記憶プールを別のディスクに移動する .....	120

## **VOB のデータ タイプと管理 VOB 階層 ..... 123**

VOB のデータ タイプ .....	123
VOB オブジェクトとレプリカ オブジェクト .....	123
ファイル システム オブジェクト .....	124
イベント レコード .....	124

共有可能派生オブジェクト .....	124
構成レコード .....	125
タイプオブジェクト .....	125
タイプオブジェクトのインスタンス .....	126
タイプオブジェクトの範囲 .....	127
定義済みタイプオブジェクトとユーザー定義のタイプオブジェクト .....	127
エレメントタイプの変更 .....	127
管理 VOB 階層とグローバルタイプ .....	128
管理 VOB 階層 .....	128
AdminVOB ハイパーリンクの表示 .....	130
管理 VOB 階層の制限 .....	131
管理 VOB が使用できなくなった場合 .....	132
管理 VOB 階層からの VOB の削除 .....	132
AdminVOB ハイパーリンクの削除 .....	132
すべての GlobalDefinition ハイパーリンクの削除 .....	133
管理 VOB の削除 .....	133
グローバルタイプの操作 .....	134
グローバルタイプの作成 .....	134
自動タイプ作成操作 .....	135
グローバルタイプの説明 .....	135
グローバルタイプの表示 .....	136
グローバルタイプの履歴表示 .....	137
グローバルタイプの保護情報の変更 .....	137
グローバルタイプのロックとアンロック .....	138
グローバルタイプのコピー .....	139
グローバルタイプ名の変更 .....	139
タイプの範囲の変更 .....	139
グローバルタイプの削除 .....	141
グローバルタイプのクリーンアップ .....	141
管理 VOB 階層のレプリカの作成 .....	141
グローバルタイプとマスターシップ .....	143
グローバルタイプのマスターシップの変更 .....	145
<b>VOB の移動と VOB データの再配置 .....</b>	<b>147</b>
別の VOB へのエレメントの再配置 .....	147
relocate の役割 .....	147
エレメントの再配置の前に実行する処理 .....	148

ボーダーラインの要素の処理 .....	149
再配置操作で発生するエラーへの対処 .....	149
ロックされたタイプまたはチェックアウト済みのファイルに関連するエラー ...	150
ソース VOB からの要素の削除中に発生するエラー .....	150
要素を再配置した後 .....	151
VOB の移動 .....	151
VOB を移動する際に実行する重要な処理 .....	152
レプリカ VOB に関する特記事項 .....	153
Windows での VOB の移動 .....	153
ドメイン内での VOB の移動 .....	154
別のドメインへの VOB の移動 .....	154
UNIX での VOB の移動 .....	158
VOB にリモート プールがある場合 .....	158
リモート プールの統合 .....	159
ClearCase 以外のアクセス用に VOB がエクスポートされている場合 .....	160
同一アーキテクチャの UNIX ホスト間での VOB の移動 .....	160
異なるアーキテクチャの UNIX ホスト間での VOB の移動 .....	161
異なるタイプのオペレーティング システムへの VOB の移動 .....	162
スキーマ バージョンの互換性 .....	163
Windows から UNIX への VOB の移動 .....	163
UNIX から Windows への VOB の移動 .....	166
NAS デバイスへの VOB の移動 .....	170
リモート プールがない VOB の移動 .....	170
リモート プールがある VOB の移動 .....	171

## **ビュー管理 ..... 173**

ビューとビュー管理の概要 .....	173
動的ビュー .....	174
ビュー ルート .....	175
デフォルトのビュー ルートを変更するには .....	175
ビュー記憶ディレクトリ .....	175
ビュープライベート記憶領域 .....	176
ビュー データベース .....	177
動的ビューによるバージョンの選択 .....	178
動的ビューによる派生オブジェクトの管理 .....	178

マルチバージョン ファイル システム .....	179
サポートされるファイル タイプ .....	180
MVFS と監視付きビルド .....	180
Windows での既知の MVFS 制限事項 .....	180
MVFS での大文字と小文字の区別 .....	181
MVFS 内での実行可能ファイルの実行 .....	181
MVFS のパフォーマンス .....	181
スナップショット ビュー .....	182
スナップショット ビュー ディレクトリ .....	183
ビュー記憶ディレクトリ .....	183
ビュー データベース .....	184
スナップショット ビューによるバージョンの選択 .....	184
ビューの作成 .....	185
リモート ビュー記憶領域 .....	185
NAS デバイスでのビューの作成 .....	185
ビューの移動 .....	186
UNIX 上の動的ビューにリモート プライベート記憶領域を 設定するかどうかの指定 .....	186
同じアーキテクチャのホストまたは NAS デバイスへのビューの移動 .....	187
Windows でスナップショット ビューを移動する場合の特別な手順 .....	189
異なるアーキテクチャの UNIX ホストへのビューの移動 .....	189
NAS デバイスに対応するビュー サーバー ホストの置換 .....	191
ビューのプロパティの表示 .....	192
ビューのサービスの停止 .....	193
ビューのサービスの再開 .....	193
ビューの削除 .....	193
<b>重要な ClearCase データのバックアップ .....</b>	<b>195</b>
VOB とビューのバックアップ ツールに関する要件 .....	195
VOB のバックアップ方法 .....	196
標準バックアップかスナップショット バックアップかの選択 .....	196
vob_snapshot を使用した VOB のバックアップ .....	198
ソース コンテナの削除の遅延 .....	199
標準バックアップ処理を使用した VOB のバックアップ .....	199
VOB 記憶ディレクトリの場所の特定 .....	200
VOB のロックとアンロック .....	200
VOB にリモート記憶プールがある場合 .....	201

すべてをバックアップできない場合にバックアップ対象にするもの	201
派生オブジェクト (DO) プールのバックアップ	202
cleartext 記憶プールのバックアップ	203
管理ディレクトリのバックアップ	203
インクリメンタル バックアップを推奨できない理由	203
ビューのバックアップ	204
ClearCase のレジストリ データのバックアップ	206
関連付けられたデータベースの一括バックアップ	206
データベースの関係の決定	209
<b>重要な ClearCase データの復元</b>	<b>211</b>
復元の準備	211
vob_restore によるバックアップからの VOB の復元	212
vob_restore で実行可能な作業	212
vob_restore の実行	213
vob_restore を使用しないバックアップからの VOB の復元	217
NAS での VOB の復元	219
バックアップからの特定エレメントの復元	219
バックアップからのビューの復元	220
復元後の VOB とビューの同期	222
参照を持つビューを VOB 内で見つけるには	223
動的ビューを同期するには	223
欠落したチェックアウトを調整するには	223
見えなくなっているビュープライベート ファイルを削除するには	224
ウインドウインされた無効な DO の削除	224
動的ビューの派生オブジェクトの状態における整合性の再確立	225
関連付けられたデータベースのグループに属するメンバの復元	225
コンポーネント VOB の復元	226
ClearQuest データベースの復元	226
PVOB の復元	227
管理 VOB 階層のメンバの復元	228
<b>定期的なメンテナンス</b>	<b>229</b>
ClearCase スケジューラ	229
スケジューラ アクセス コントロール リストの管理	230
タスクの作成	231
タスクの編集	232

タスクの削除 .....	233
ジョブの管理 .....	233
ジョブの作成 .....	233
ジョブのスケジュールの指定 .....	234
ジョブ通知の指定 .....	235
ジョブのプロパティの表示 .....	236
ジョブのプロパティの編集 .....	237
ジョブの即座の実行 .....	237
ジョブの削除 .....	237
VOB 記憶領域の増加を制御するスクラブ .....	238
VOB 記憶プールのスクラブ .....	238
VOB データベースのスクラブ .....	239
デフォルトのスクラブ パラメータの調整 .....	239
派生オブジェクトのより頻繁なスクラブ .....	240
派生オブジェクトのスクラブの微調整 .....	240
より控えめなスクラブ .....	241
checkvob の概要 .....	242
レプリカ VOB に関する考慮事項 .....	242
checkvob を使用した内部 VOB 不整合の検出と修正 .....	243
個々のファイル エlement または DO の処理 .....	244
プール モードの処理 .....	245
記憶プールの問題の説明 .....	245
プール名と保護の問題 .....	246
ソース プール、DO プール、cleartext プール: 不良なプール ルート .....	246
ソース プール、DO プール: Windows 上で間違っ保護されたコンテナ .....	246
欠落したデータ コンテナと参照不能データ コンテナ .....	247
ソースプール: 欠落したコンテナ .....	247
ソース プール: 参照不能コンテナ (残骸) .....	248
DO プール: 欠落したコンテナ .....	249
DO プール: 参照不能コンテナ (残骸) .....	250
checkvob -force -fix を使用したデータ消失の最小化 .....	250
checkvob を使用したグローバル タイプの問題の検出と修正 .....	252
checkvob を使用した破損ハイパーリンクの検出と修正 .....	254
UCM 環境での checkvob の使用法 .....	254

動的ビューのビュー記憶領域のメンテナンス .....	254
ビューの内容に関する情報の取得 .....	255
ビュープライベート記憶領域のスクラブ .....	256
ビューの手動クリーンアップ .....	256
<b>データのインポート .....</b>	<b>259</b>
Microsoft SourceSafe からのデータのインポート .....	259
SourceSafe のサンプル構成の概要 .....	259
環境の設定 .....	261
clearexport_ssaf の実行 .....	262
clearimport の実行 .....	263
結果の検証 .....	263
RCS データのインポート .....	265
データ ファイルの作成 .....	265
clearimport の実行 .....	265
PVCS データのインポート .....	266
データ ファイルの作成 .....	266
変換スクリプトの実行 .....	267
<b>トラブルシューティング .....</b>	<b>269</b>
NTFS の記憶ディレクトリ ACL の修復 .....	269
VOB と ビューの記憶ディレクトリの ACL .....	269
VOB または ビューの記憶ディレクトリをコピーするとき NTFS ACL を保持する ..	271
保護問題の原因 .....	272
記憶ディレクトリの不正なコピー .....	272
ファイル システムの NTFS への変換 .....	272
権限の編集 .....	272
保護問題を修復するためのユーティリティ .....	273
fix_prot .....	273
オプション .....	273
例 .....	274
lsacl .....	275
オプション .....	275
保護問題の修復 .....	275
エレメントを再配置した後の既存のビューの問題 .....	276
クリーンアップのガイドライン .....	277
再配置されたエレメントに対するシンボリック リンクの問題 .....	278

ディレクトリ バージョンの手動による更新 .....	279
relocate で作成されたシンボリック リンクの修復 .....	279
再配置されたエレメントを参照するために古いターゲット ディレクトリ バージョンを修正する .....	280
再配置されたエレメントを参照するためにソース ディレクトリの 最新バージョンを修正する .....	280
MVFS Data コンテナの検索 .....	281
シナリオ .....	281
ファイルの ClearCase 状態の判定 .....	281
ファイルのフルパス名の判定 .....	282
VOB の位置 .....	282
ビューの位置 .....	282
個々のファイルの位置 .....	283
チェックアウト バージョンの位置の確認 .....	283
チェックイン バージョンの cleartext コンテナの場所の確認 .....	284
チェックイン バージョンのソース コンテナの場所の確認 .....	284
ビュープライベート ファイルの位置の確認 .....	284
UNIX のリモート記憶ブールの問題 .....	284
UNIX でのリンクとディレクトリ .....	285
UNIX ビューのルート ディレクトリの再帰探索の防止 .....	285
<b>クライアント ホストのパフォーマンスの改善 .....</b>	<b>287</b>
クライアント ホストの構成のガイドライン .....	287
MVFS キャッシュ サイズの調査と調整 .....	288
MVFS キャッシュの統計情報の調査 .....	288
MVFS スケーリング係数の調整 .....	289
個々のキャッシュ パラメータの設定 .....	291
属性キャッシュの失敗の最小化 .....	294
属性キャッシュの失敗の合計 .....	294
閉じてから開くまでの失敗 .....	294
生成の失敗 .....	294
キャッシュ タイムアウトの失敗 .....	294
キャッシュ埋めの失敗 .....	295
イベント タイムの失敗 .....	296
ビュー キャッシュ .....	296
ビュー キャッシュ情報の取得 .....	297
出力の分析 .....	298
ビューの再設定 .....	298

リファレンス情報 .....	299
mvfscache .....	299
構文 .....	299
オプションと引数 .....	299
例 .....	301
mvfsstat .....	301
構文 .....	302
MVFS キャッシュの統計情報 .....	302
オプションと引数 .....	303
<b>VOB ホストのパフォーマンスの改善 .....</b>	<b>305</b>
プロセス オーバーヘッドの最小化 .....	305
ディスク パフォーマンスの最大化 .....	306
Windows でのディスク キャッシュ用のメモリの追加 .....	306
UNIX でのブロック バッファ キャッシュの調整 .....	307
ブロック バッファ キャッシュに関する統計 .....	307
Lock Manager の起動オプションの変更 .....	308
Lock Manager の実装 .....	308
Lock Manager の起動オプションを変更するには .....	308
Windows ホストの場合 .....	308
UNIX ホストの場合 .....	309
lockmgr のリファレンス情報 .....	309
構文 .....	309
オプションと引数 .....	309
サーバーのパフォーマンスと NAS デバイス .....	311
<b>異なるプラットフォームのファイル システムへのアクセスの設定 .....</b>	<b>313</b>
NFS クライアント製品 .....	313
大文字/小文字の自動変換の無効化 .....	314
Shaffer Solutions 社の DiskAccess の場合 .....	314
Hummingbird 社の NFS Maestro の場合 .....	314
NFS クライアントのデフォルト保護モードの設定 .....	315
Shaffer Solutions 社の DiskAccess の場合 .....	315
Hummingbird 社の NFS Maestro の場合 .....	315
正しいログオン名の設定 .....	316
Shaffer Solutions 社の DiskAccess の場合 .....	316
Hummingbird 社の NFS Maestro の場合 .....	317

Hummingbird 社の NFS Maestro: DOS 共有の無効化 .....	317
自動マウントと NFS クライアント ソフトウェア .....	317
DiskAccess: ClearCase サーバー プロセス ユーザーのサポート .....	318
Windows レジストリ ファイルへのユーザー識別情報の保存 .....	319
リモート記憶プールのある UNIX VOB の Windows タグ .....	319
poolmap 構文 .....	321
既存の VOB タグへの記憶プールのマッピング .....	321
SMB サーバー製品 .....	322
Samba のインストールと設定 .....	322
ClearCase サーバー プロセス ユーザーのマッピング .....	322
Samba Web Administration Tool (SWAT) の使用 .....	323
ClearCase 用の Samba 全体設定の設定 .....	323
VOB とビュー記憶領域用の共有の作成 .....	324
Samba サービスの起動 .....	324
Samba をサポートするための ClearCase の設定 .....	325
ClearCase 以外のファイルを使用した Samba の設定のテスト .....	325
ClearCase と共に使用する Samba の設定のテスト .....	326
TAS のインストールと設定 .....	326
AIX 上でのマルチユーザー カーネル ドライバの有効化 .....	326
Syntax Administration Framework へのアクセス .....	327
TAS の初期セットアップの実行 .....	327
TAS の基本設定 .....	328
CIFS レルムの有効化と設定 (TAS 6.x) .....	328
ClearCase サーバー プロセス ユーザーのマッピング (TAS 6.x) .....	328
ボリュームの作成 (TAS 6.x) .....	329
ファイル サービスの設定 (TAS 6.x) .....	330
ClearCase をサポートするための TAS 7.x の設定 .....	331
サービスの開始とサービス接続の承認 .....	333
TAS をサポートするための ClearCase の構成 .....	333
通常のファイルを使用した TAS の設定のテスト .....	333
ClearCase と共に使用する TAS の設定のテスト .....	334
<b>ClearCase と Windows ドメイン .....</b>	<b>335</b>
ClearCase と互換性のあるドメイン構成 .....	335
ClearCase のドメインに対する要件 .....	335
非ドメイン ホスト上の ClearCase .....	336

ドメイン ユーザーとグループ アカウント .....	336
ClearCase プライマリ グループの設定 .....	337
必須のドメイン アカウントの手動定義 .....	338
マルチ ユーザー アカウント ドメインのサポート .....	339
Active Directory ユニバーサル グループの使用 .....	339
Windows NT ドメインでのプロキシ グループとドメイン マッピングの使用法 .....	340
VOB エレメントの権限の設定 .....	341
VOB 記憶領域の ACL の設定 .....	341
Active Directory への変換 .....	342
Active Directory の理解 .....	342
Active Directory が ClearCase に与える影響 .....	342
Active Directory へのアップグレードまたは移行の計画 .....	343
ClearCase ホストの準備 .....	343
ドメイン アップグレード シナリオ .....	344
1 つのドメインのアップグレード .....	344
マスター ドメインとそのリソース ドメインのアップグレード .....	345
複数のマスター ドメインとリソース ドメインのアップグレード .....	345
プロキシ グループの変換 .....	346
ドメイン移行シナリオ .....	347
複数のドメインの移行 .....	348
ユーザーとグループの移行 .....	348
移行中に新しいユーザーを追加しなければならない場合 .....	349
個別のホストの移行 .....	349
クライアントは移行できるが VOB サーバーを移行できない場合 .....	351
vob_sidwalk を使用した VOB ユーザーとグループの変更または更新 .....	352
ドメイン移行後の履歴 SID の再マッピング .....	352
VOB を新しいドメインに移動するときに現在の SID を再マッピングする .....	353
VOB 所有者への所有権の再割り当て .....	353
VOB 記憶ディレクトリ保護の再設定 .....	353
識別情報と権限を維持するレプリカでの -delete_groups の使用 .....	354

## **Rational Web Platform の構成 ..... 355**

RWP のインストール ディレクトリ .....	355
RWP 構成ファイル .....	356
構成ファイルの参照用バージョン .....	356
デフォルトの RWP HTTP ポートを変更するには .....	357
デフォルトの RWP servlet engine ポートを変更するには .....	357

RWP のログを構成するには .....	358
ログのローテーションとクリーンアップ .....	359
RWP が使用するユーザー アカウントを変更するには .....	360
Windows 上で RWP ユーザー アカウントを変更するには .....	360
UNIX 上で RWP ユーザー アカウントを変更するには .....	361
RWP の停止と再起動 .....	361
ほかの Web サーバーから RWP へのアクセスを構成するには .....	362
Apache 用 mod_proxy サポートの構成 .....	362
Internet Information Server (IIS) の URL リダイレクトの構成 .....	363
セキュア ソケットを使用するための RWP の構成 .....	365
RWP への安全なアクセスの構成 .....	366
RWP に対するその他の変更 .....	368
ClearCase Web インターフェイスの構成 .....	368
ClearCase プライマリ グループの指定 .....	369
Web ビュー記憶領域 .....	369
アップロード サイズの制限 .....	370
セッション タイムアウトの指定 .....	370
一時記憶領域用ディレクトリの指定 .....	370
Windows 上にアプレットをダウンロードする権限 .....	370
<b>UNIX 上での非 ClearCase アクセスの構成 .....</b>	<b>371</b>
UNIX ホスト上での非 ClearCase アクセスの使用法 .....	371
使用上の制限事項 .....	371
非 ClearCase アクセスのためのエクスポート ビューの設定 .....	372
複数の VOB のエクスポート .....	373
エクスポート構成 .....	373
マルチホップ構成の回避方法 .....	374
マルチホップ構成での操作 .....	374
特定のホストへのエクスポートを制限する方法 .....	374
非 ClearCase アクセスでの自動マウントの使用法 .....	375
非 ClearCase アクセスでの NFS の問題 .....	376
NFS クライアント キャッシュでの問題 .....	376
NFS ロックでの問題 .....	377

<b>Microsoft Web オーサリング ツールとの統合</b>	<b>379</b>
統合の概要	379
サーバーの設定の概要	380
クライアントの設定の概要	380
サーバーの設定の手順	381
手順 1: IIS のインストール	381
手順 2: FPSE または OSE のインストール	382
手順 3: ClearCase のインストール	383
手順 4: Web オーサリング統合の構成ウィザードの実行	383
クライアントの設定の手順	384
手順 1: クライアント アプリケーションのインストール	384
手順 2: ソース管理への Web の追加	384
FrontPage 98 から	384
FrontPage 2000 またはそれ以降から	385
Visual InterDev 6.0 から	385
手順 3: 新しい Web コンテンツがソース管理に追加されたことの確認	386
手順 4: 権限の設定	386
FrontPage 98	386
FrontPage 2000	386
Visual InterDev 6.0	386
FrontPage 2000 またはそれ以降のためのローカル モードでのクライアントの設定	387
Web Folders への対応	388
Web サーバー上の共有ビューの更新	388
データの移行と変換	388
統合機能のヘルプへのアクセス	389
<b>VOB サイズの見積もり</b>	<b>391</b>
データベース容量の見積もり	391
プール容量の見積もり	392
<b>索引</b>	<b>395</b>



# 表目次

表 1	ClearCase クライアントから VOB データにアクセスするプロトコル	62
表 2	ClearCase クライアントからビュー データにアクセスするプロトコル	63
表 3	view_server から VOB データにアクセスするプロトコル	63
表 4	Windows のファイル名で使用できない文字	71
表 5	ClearCase オブジェクトの保護モード	77
表 6	ClearCase オブジェクトの保護モードを表す数値	78
表 7	部分バックアップでの VOB ディレクトリの重要度	202
表 8	スケジューラ ACL の項目におけるアクセスのタイプ	230
表 9	scrubber と vob_scrubber のデフォルトのアクション	238
表 10	MVFS ファイルの記憶ディレクトリ	283
表 11	メモリ サイズと MVFS スケーリング係数の関係	289
表 12	MVFS スケーリング係数と個々の MVFS キャッシュ サイズの関係	290
表 13	MVFS キャッシュ情報	292
表 14	ClearCase 用の Samba 全体設定の設定	323
表 15	デフォルトの RWP servlet engine ポート	357
表 16	RWP のログ レベル	359
表 17	Microsoft Web オーサリング ツールとの統合を サポートしているプラットフォーム	381
表 18	一般的な VOB データ タイプのサイズ	391
表 19	タイプ マネージャの圧縮率	393



# 図目次

図 1	リージョンが 1 つのみのレジストリ .....	38
図 2	2 つのリージョンがあるレジストリ .....	40
図 3	cleartool コマンドによるレジストリ データへの作用 .....	47
図 4	管理 VOB 階層 .....	129
図 5	管理 VOB 階層のレプリカ作成要件 .....	142
図 6	データベースの関係 .....	207
図 7	SourceSafe のサンプル構成 .....	260
図 8	インポートされたエレメントのバージョン ツリー .....	264
図 9	ccweb と RWP の間の安全な通信 .....	367



# まえがき

## 本書について

---

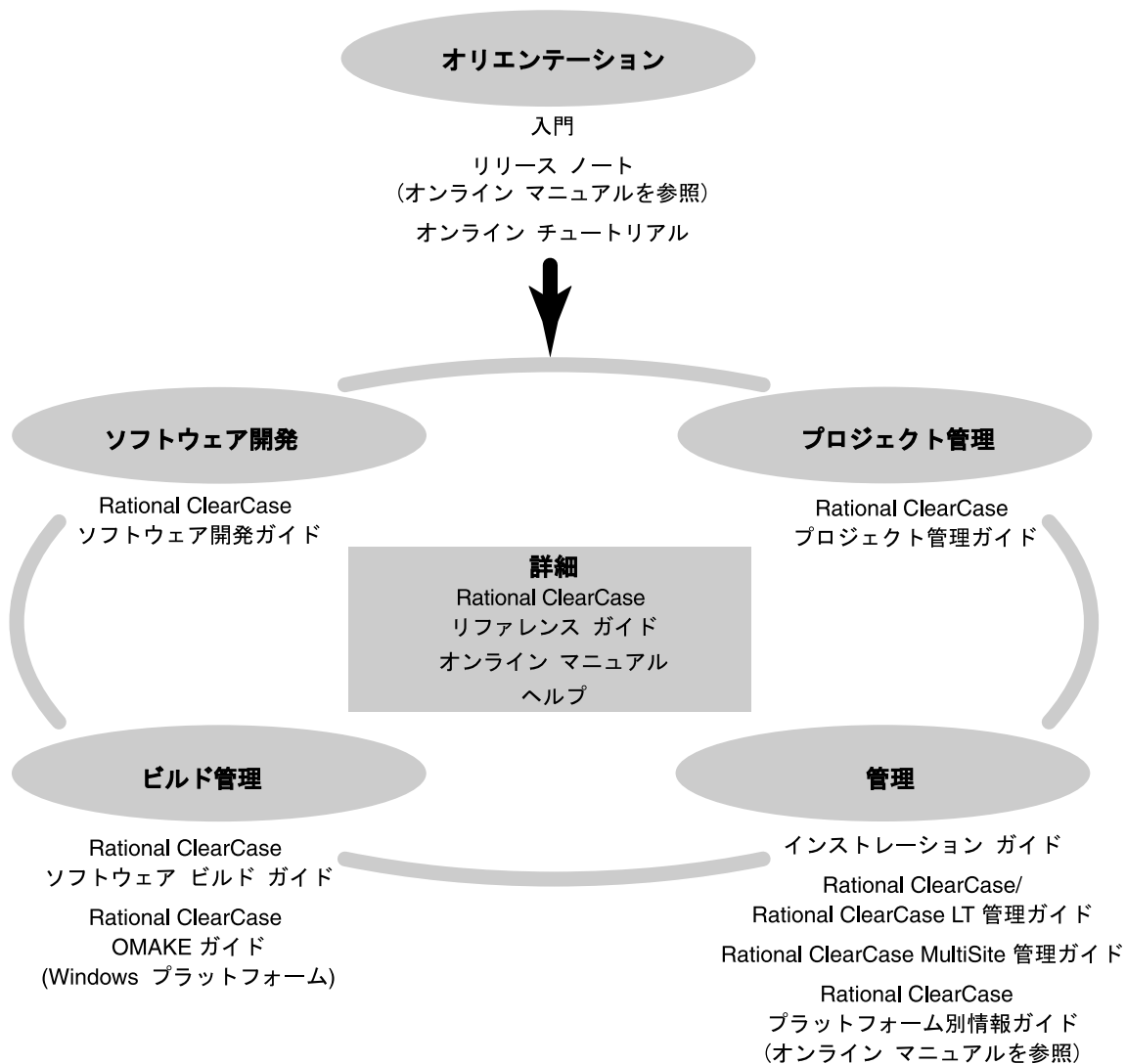
本書は、データ リポジトリの作成と管理、サーバーの管理、ユーザー アカウントとグループ アカウントの管理などを行う Rational™ ClearCase® と Rational ClearCase LT のユーザーや管理者を対象としています。この管理ガイドでは、UNIX または Windows の ClearCase と ClearCase LT に関する以下の事項について詳しく説明します。

- クライアント ホストとサーバー ホストの管理
- ライセンスの管理
- ClearCase レジストリ (共有情報の集中ディレクトリ) の管理
- ユーザー アカウントとグループ アカウントの管理
- データ リポジトリである VOB (バージョン付きオブジェクト ベース) の作成と管理
- ビュー (ユーザー ワークスペース) の作成と管理
- 重要なデータのバックアップと復元
- ほかの構成管理システムからのデータのインポート
- 定期的なメンテナンスとトラブルシューティング
- クライアント ホストとサーバー ホストのパフォーマンス向上のための調整
- サードパーティ製ファイル アクセス製品の設定 (オプション)
- Rational Web プラットフォームと ClearCase Web インターフェイスの設定

本書の Windows に関連する説明は、ClearCase と ClearCase LT をサポートするすべての Microsoft Windows プラットフォームに当てはまります。UNIX に関連する説明は、ClearCase と ClearCase LT をサポートするすべての UNIX プラットフォームと Linux プラットフォームに当てはまります。サポートされている UNIX オペレーティング システムと Windows オペレーティング システムについては、インストール ガイドを参照してください。

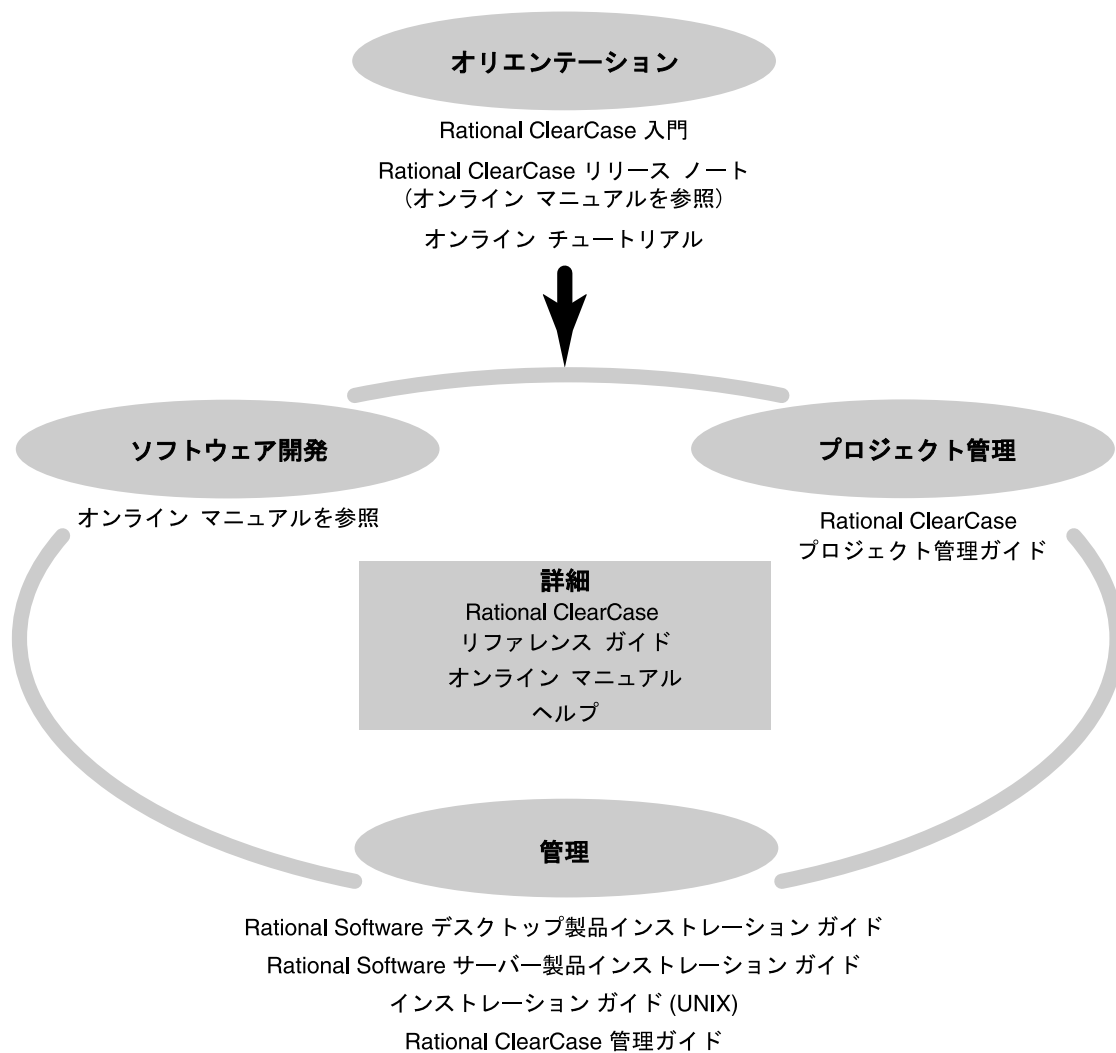
# ClearCase マニュアル ロードマップ

---



# ClearCase LT マニュアル ロードマップ

---



## ClearCase とほかの Rational 製品との統合

統合	説明	説明が記載されている場所
ベース ClearCase - ClearQuest®	変更依頼を ClearCase エレメントのバージョンに関連付けます。	『Rational ClearCase ソフトウェア開発ガイド』 『Rational ClearCase プロジェクト管理ガイド』 『Rational ClearQuest 管理ガイド』
ベース ClearCase - Apex	Apex 開発者がファイルを ClearCase に保存できるようにします。	『Installing Rational Apex (UNIX)』
ベース ClearCase - ClearDDTS	変更依頼を ClearCase エレメントのバージョンに関連付けます。	『ClearCase ClearDDTS Integration』
ベース ClearCase - PurifyPlus	開発者が、PurifyPlus から ClearCase を呼び出せるようにします。	PurifyPlus のヘルプ
ベース ClearCase - RequisitePro	RequisitePro プロジェクトを ClearCase 内にアーカイブします。	『Rational RequisitePro ユーザーズガイド』 RequisitePro のヘルプ
ベース ClearCase - Rose	Rose モデルを ClearCase に保存します。	Rose のヘルプ
ベース ClearCase - Rose RealTime	Rose RealTime モデルを ClearCase に保存します。	『Rose RealTime Toolset Guide』 『Rose RealTime Guide to Team Development』
ベース ClearCase - SoDA	ClearCase から情報を収集し、その情報をさまざまなレポート形式で表示します。	『Using Rational SoDA for Word』 『Using Rational SoDA for Frame』 SoDA のヘルプ
ベース ClearCase - XDE™	XDE モデルを ClearCase に保存します。	XDE のヘルプ
UCM - ClearQuest	UCM アクティビティを ClearQuest レコードにリンクします。	『Rational ClearCase ソフトウェア開発ガイド』 『Rational ClearCase プロジェクト管理ガイド』 『Rational ClearQuest 管理ガイド』
UCM - PurifyPlus	開発者が PurifyPlus から ClearCase を呼び出せるようにします。	PurifyPlus のヘルプ

統合	説明	説明が記載されている場所
UCM - RequisitePro	RequisitePro 管理者が UCM で RequisitePro プロジェクトのベースラインを作成できるようにすると共に、ベースラインから RequisitePro プロジェクトを作成できるようにします。	『Rational RequisitePro ユーザーズガイド』 RequisitePro のヘルプ 『Rational Suite® 統一変更管理 (UCM) ユーザーズガイド』
UCM - Rose	Rose モデルを ClearCase に保存します。	Rose のヘルプ 『Rational Suite 統一変更管理 (UCM) ユーザーズガイド』
UCM - Rose RealTime	アクティビティをリビジョンに関連付けます。	『Rose RealTime Toolset Guide』 『Rose RealTime Guide to Team Development』
UCM - SoDA	ClearCase から情報を収集し、その情報をさまざまなレポート形式で表示します。	『Using Rational SoDA for Word』 『Using Rational SoDA for Frame』 SoDA のヘルプ
UCM - TestManager	テストアセットを ClearCase に保存します。	『Rational TestManager User's Guide』 TestManager のヘルプ 『Rational Suite 統一変更管理 (UCM) ユーザーズガイド』
UCM - XDE	XDE モデルを ClearCase に保存します。	XDE のヘルプ
UCM - XDE Tester	XDE Tester データストアを ClearCase に保存します。	XDE Tester のヘルプ

## 表記規則

---

本書の表記規則は次のとおりです。

- `ccase-home-dir` は、ClearCase 製品ファミリーがインストールされているディレクトリを表します。デフォルトのインストール ディレクトリは、UNIX では `/opt/rational/clearcase`、Windows では `C:\Program Files\Rational\ClearCase` です。
- `cquest-home-dir` は、Rational ClearQuest がインストールされているディレクトリを表します。デフォルトのインストール ディレクトリは、UNIX では `/opt/rational/clearquest`、Windows では `C:\Program Files\Rational\ClearQuest` です。
- **太字**は、コマンド名やブランチ名など、ユーザーが入力可能な名前に使用します。
- **sans serif** フォントは、ファイル名、ディレクトリ名、ファイル拡張子に使用します。
- メニュー名やチェック ボックス名のような、GUI 要素は、`[]` で囲んで表記します。
- **等幅**フォントは、例に使用します。ユーザー入力とプログラム出力を区別する必要がある場合、ユーザー入力には**太字**を使用します。
- 出力されない文字は、`<EOF>`、`<NL>` のように表示します。
- キー名やキーの組み合わせは、大文字で `SHIFT`、`CTRL+G` のように表示します。
- `[]` 大カッコは、書式や構文の記述でオプション項目を囲むために使用します。
- `{ }` 中カッコは、書式や構文の記述で選択項目のリストを囲むために使用します。
- `|` 縦棒は、選択項目のリストを区切るために使用します。
- ... 構文記述内の省略記号は、省略記号に先行する項目や行を 1 回以上繰り返せることを示します。それ以外の省略記号は、情報の省略を示します。

**メモ:** あるコンテキストでは、`"*" や "?"` と同様に、`"..."` をパス名内でワイルドカードとして使用することができます。詳細については、`wildcards_ccase` のリファレンス ページを参照してください。

- コマンド名またはオプション名に省略形がある場合、`"中点" (·)` は最短の有効な省略形を示します。たとえば次のような機能があります。

`lsc:checkout`

## オンライン マニュアル

---

ClearCase 製品ファミリ (CPF) には、次のオンライン マニュアルが含まれています。

**ヘルプ システム:** [ヘルプ] メニュー、[ヘルプ] ボタン、[F1] キーを使用します。一連のオンライン マニュアルの内容を表示するには、次のいずれかを行います。

- UNIX の場合は、「cleartool man <コマンド名>」と入力します。
- Windows の場合は、[スタート]、[プログラム]、[Rational Software]、[Rational ClearCase] をポイントし、[ヘルプ] をクリックします。
- いずれのプラットフォームの場合でも、Rational ClearCase MultiSite の内容を表示するには、「multitool man <コマンド名>」と入力します。
- ダイアログ ボックスに関する情報を表示するには、ダイアログ ボックスの [ヘルプ] ボタンをクリックするか、[F1] を押します。

**リファレンス ページ:** cleartool man コマンドと multitool man コマンドを使用します。詳細については、man のリファレンス ページを参照してください。

**コマンド構文:** -help コマンド オプションまたは cleartool help コマンドを使用します。

**チュートリアル:** 製品の重要な機能の段階的な説明を表示します。チュートリアルを開始するには、次のいずれかを行います。

- UNIX の場合は、「cleartool man tutorial」 と入力します。
- Windows の場合は、[スタート]、[プログラム]、[Rational Software]、[Rational ClearCase] をポイントし、[ClearCase チュートリアル] をクリックします。

**PDF マニュアル:** 次のディレクトリに移動します。

- UNIX の場合は、ccase-home-dir/doc/books
- Windows の場合は、ccase-home-dir¥doc¥books

## カスタマ サポート

---

ソフトウェアやマニュアルの問題については、電話、ファックス、または電子メールで以下の Rational カスタマ サポートまでお問い合わせください。サポートの時間帯、対応言語、その他のサポート情報については、日本ラショナル ソフトウェア社の Web サイトの **サポート情報** (<http://www.rational.co.jp/supports/>) を参照してください。

地域	電話	Fax	電子メール
アジア太平洋 (日本を含む)	+61-2-9419-0111	+61-2-9419-0123	support@apac.rational.com (英語のみ対応) support@japan.rational.com (日本語対応可)

# ClearCase ネットワークの 計画と管理

# 1

Rational ClearCase 製品ファミリーは、地理的に分散したあらゆる規模のチームに対応するソフトウェア構成管理ソリューションを提供します。本書は、ClearCase コミュニティ (ユーザー、コンピュータ、共有データ、それらを接続するネットワーク) も管理する UNIX または Windows の上級管理者を対象としています。本書で説明している手順の多くは、UNIX コンピュータや Windows コンピュータのスクリプトとコマンド行の規則、ファイル システム アクセス コントロール、ネットワーク プロトコル、システム管理ツールの知識があることを前提としています。

## 管理責任の概要

---

この章では、システム管理者の視点から見た ClearCase コミュニティについて説明します。管理責任の概要とともに、本書内の関連する参照先についてもご案内します。

ClearCase 管理者として考慮すべきことは、次の 4 点です。

- ホストとネットワークの管理。サーバー ホストのインストールと設定、効率的なクライアント/サーバー接続の確保、サーバー負荷の管理と分散、最高のパフォーマンスを引き出すためのクライアントとサーバーのチューニングなど。
- データ管理。リポジトリのバックアップとリカバリ、データのインポート/エクスポート、記憶領域の管理、定期メンテナンスなど。
- ユーザー管理。ユーザー アカウントとグループ アカウントの要件、ユーザー アクセス問題の解決、データ セキュリティの確保など。
- ライセンス管理。ライセンス サーバー管理、ライセンス キー管理、ライセンス ユーザーポリシーの確立など。

また、ユーザーが UNIX コンピュータと Windows コンピュータの両方から共通リソース セットにアクセスする ClearCase コミュニティの場合、管理者は異なるオペレーティング システムのユーザーが ClearCase のコントロール下にある成果物へのアクセスを共有する際に発生する問題を考慮する必要があります。

## ClearCase 製品ファミリ

ClearCase 製品ファミリは、次の 3 製品で構成されます。

- **Rational ClearCase:** 企業規模での運用に対応したソフトウェア構成管理ソリューションです。バージョン管理、ワークスペース管理、プロセス構成機能、ビルド管理を提供します
- **Rational ClearCase MultiSite:** Rational ClearCase のオプションアドオン製品です。データのレプリカ作成サービスを提供し、地理的に離れた場所にいるチームによる並行開発を可能にします。
- **Rational ClearCase LT:** 小規模なプロジェクト ワークグループ向けに開発された初級バージョン管理ツールです。

本書は、ClearCase か ClearCase LT、またはその両方を使用するコミュニティの管理者を対象としています。本書で ClearCase と記載されている内容は ClearCase と ClearCase LT の両方に対応しています。ClearCase LT と記載されている内容はその製品のみに対応しています。

**メモ:** 本書では Rational ClearCase MultiSite 使用時の注意事項についても記載していますが、MultiSite の管理方法については MultiSite のガイドを参照してください。

## ClearCase と ClearCase LT の違い

本書では、ClearCase と ClearCase LT の違いについて適宜説明します。ClearCase LT では、ClearCase の機能のサブセットを提供します。

- ClearCase LT コミュニティには、ClearCase LT サーバーという 1 つのサーバー ホストのみが存在します。このホストは、すべての ClearCase サービスを実行し、コミュニティで作成、使用されるすべての ClearCase データ リポジトリを管理します。
- ClearCase LT レジストリでは、レジストリ リージョン、レジストリ バックアップ/切り替えなど、マルチサーバー環境で必要な機能がサポートされません。
- ClearCase LT では、動的ビューや動的ビューを利用するあらゆる機能 (ビルド管理など) がサポートされません。

本書の大部分は、これら 2 つの製品に適用されます。ClearCase LT に該当しない項目については、そのつど明記します。

## ホストとホスト管理

---

ClearCase は、クライアント/サーバー アーキテクチャに基づく分散型アプリケーションです。ClearCase LT は、クライアント/サーバーを厳密に区別し、コミュニティ内のサーバー ホストを 1 つに制限します。ClearCase は複数のサーバーをサポートし、クライアント/サーバー構成により高い柔軟性を与えます。ClearCase と ClearCase LT の多くの操作では、複数のホストのプログラムとデータが使用されます。ここでは、ClearCase コミュニティに存在するホストのタイプと、各ホストの用途について説明します。

### ネットワークワイドなリリース ホスト

ClearCase と ClearCase LT では、ネットワーク内にある 1 台のコンピュータがネットワークワイドなリリース ホストとして機能します。このホストのディレクトリは、ネットワークワイドリリース 領域と呼ばれ、製品リリース全体を配布メディアから抽出した状態、または Rational からダウンロードした状態で格納します。必要に応じてパッチやサービス リリースを適用し、最新の拡張機能や障害修正によってリリース領域を更新できます。更新後に、このリリース領域から ClearCase を再インストールして、これらの更新事項を各ホストに配布できます。いくつかのインストール オプションを使用すると、ローカルにインストールしなくても、UNIX 環境の ClearCase ホストからリリース領域に対するシンボリック リンクを利用して多くの ClearCase 実行ファイルとライブラリにアクセスできるようになります。Windows 環境の ClearCase ホストでは、インストールの完了後にネットワークワイドリリース ホストにアクセスすることはありません。ネットワークワイドリリース ホストでは ClearCase を実行する必要はありません。

### ClearCase LT ホスト

ClearCase LT ネットワークに存在するホストは、次の 2 種類のみです。

- **ClearCase LT クライアント:** 各 ClearCase LT ユーザーが操作するのがクライアント ホストです。ClearCase LT コマンド行プログラムやグラフィカル ユーザー インターフェイス、その他のソフトウェア (開発ツール、Rational Suite、オペレーティング システム ユーティリティなど) を実行します。
- **ClearCase LT サーバー:** すべての ClearCase LT クライアントには ClearCase LT サーバーホストのサービスが必要です。ClearCase LT クライアント ホストは 1 つの ClearCase LT サーバーにしかアクセスできません。ClearCase LT サーバー ホストは、複数のデータ リポジトリをサポートできるので、多くの ClearCase LT クライアントのホストとして機能します。

**メモ:** ClearCase LT サーバー構成のガイドラインについては、97 ページの「VOB サーバー ホスト構成のガイドライン」を参照してください。

## ClearCase ホスト

ClearCase ネットワークのホストの役割は、その時々で変化することもある、1 つのホストが同時に複数の役割を担うこともあります。

- **クライアント ホスト**: 各 ClearCase ユーザーが操作するのがクライアント ホストです。ClearCase コマンド行プログラムやグラフィカル ユーザー インターフェイス、その他のソフトウェア (開発ツール、Rational Suite、オペレーティング システム ユーティリティなど) を実行します。ClearCase クライアントのインストール時に、マルチバージョン ファイル システム (MVFS) も同時にインストールできます。これは、ホストのネイティブ オペレーティング システムを拡張して、動的ビューをサポートするファイル システムを提供する機能です。
- **サーバー ホスト**: ClearCase データ リポジトリを作成できるホストが、サーバー ホストです。サーバー ホストには、クライアント ソフトウェアをほとんど実行しない (そのためクライアント ソフトウェアがインストールされない) 専用サーバーと、クライアントとしても機能する兼用サーバーがあります。
- **ライセンス サーバー ホスト**: 各 ClearCase コミュニティには、ライセンス サーバー ホストが必要です。通常、サイトの準備段階で指定します。ClearCase ライセンス サーバーは、ClearCase コマンドを実行するためのライセンスを提供します。コミュニティ内の各クライアントまたはサーバー ホストは、ClearCase のインストール時にライセンス サーバーに関連付けられます。第 2 章では、ClearCase ライセンス サーバーと ClearCase ライセンス管理について詳しく説明します。
- **レジストリ サーバー ホスト**: 各 ClearCase コミュニティには、レジストリ サーバー ホストが必要です。通常、サイトの準備段階で指定します。コミュニティ メンバーは、ClearCase レジストリを使用してデータ リポジトリなどの共有リソースにアクセスします。コミュニティ内の各クライアントまたはサーバー ホストは、ClearCase のインストール時にレジストリ サーバーに関連付けられます。第 3 章では、ClearCase レジストリとレジストリ サーバー ホストについて詳しく説明します。
- **Rational Web プラットフォーム サーバー ホスト**: ネットワーク内の少なくとも 1 台のホストが、Rational Web プラットフォーム (以下 RWP) サーバー ホストとして機能します。RWP は、ClearCase Web インターフェイス用のアプリケーション サポートを提供します。付録 C では、標準以外のポートの割り当て、プロキシ アクセス、セキュア ソケットなど、サイトで使用する機能をサポートするための RWP の設定方法について説明します。
- **Network-attached storage (NAS) デバイス**: 一部の NAS デバイスは ClearCase との使用を認められています。このデバイスを ClearCase プログラムやデータの保存に使用できます。NAS デバイスでは ClearCase を実行できません。ClearCase ホストからローカル エリア ネットワーク経由でアクセスできる記憶領域を提供します。認定済み NAS デバイスに対するアクセスの設定については、16 ページの「ClearCase と NAS (Network-Attached Storage) デバイス」を参照してください。認定されている NAS デバイスのリストについては、オンラインの『Rational ClearCase プラットフォーム別情報ガイド』を参照してください。

- 非 ClearCase UNIX ホスト: ClearCase を実行できない UNIX ホストでも、標準の UNIX ネットワーク ファイル システムの機能によって、制限付きで ClearCase データにアクセスできます。詳細については、付録 D を参照してください。

## ライセンス方式とライセンス管理

---

ほとんどの ClearCase アプリケーションやコマンド (cleartool コマンドや、ClearCase エクスプローラなどの GUI) は、ライセンスがないと実行できません。ClearCase と ClearCase LT では、ライセンスの仕組みと管理要件が異なります。ClearCase LT は、GLOBETROTTER Software, Inc. の FLEXlm ライセンス管理ツールに基づく Rational Common Licensing を使用します。ClearCase は所有権ライセンス スキーマを使用します。ユーザーは、通常、ClearCase のインストール時にコンピュータ上でライセンスを設定します。管理者は、ライセンス サーバー ホストをセットアップし、ライセンス キーを管理して、ライセンス使用ポリシーを確立する必要があります。

ライセンスとライセンス管理の詳細については、第 2 章を参照してください。

## データ記憶領域とデータ管理

---

ClearCase データは、複数のタイプのリポジトリに保存されます。

- ClearCase レジストリには、クライアントからのリポジトリ アクセスを有効にする情報が保存されます。この情報があれば、詳細なホスト情報がなくてもアクセスできます。
- バージョン付きの成果物はバージョン付きオブジェクト ベース (VOB) に保管されます。これは組み込みデータベース管理システムによって実装されます。
- ビューと呼ばれる成果物ワークスペースはビュー データベースでサポートされます。これも同じ組み込みデータベース管理システムによって実装されます。

リポジトリのメンテナンスは、ClearCase 管理における重要な作業の 1 つです。

**メモ:** ClearCase を Rational ClearQuest と統合することができます。これは独自の変更依頼データベースを作成するための変更依頼管理アプリケーションです。ClearCase コミュニティで ClearQuest も使用する場合は、データ管理計画に ClearCase リポジトリの適用を検討することをお勧めします。

## ClearCase レジストリ

ClearCase レジストリは、VOB とビューの場所の情報や、さまざまなホスト ClearCase 設定のデフォルト値など、ClearCase コミュニティで共有する情報の集中リポジトリになります。各 ClearCase ホストは、1 つのレジストリ サーバー ホストに割り当てられ、そのホストのクライアントになります。ClearCase LT コミュニティでは、レジストリ サーバー ホストは常に ClearCase LT サーバーです。ClearCase コミュニティでは、ClearCase がインストールされているいずれかのコンピュータがレジストリ サーバー ホストになります。

ClearCase コミュニティは、共有のレジストリを一部使用して定義されます。レジストリ サーバー ホストが使用不可能になったり、レジストリ データが破損したりすると、ユーザーは ClearCase で管理されている成果物にアクセスできなくなります。通常、レジストリ データは自動的に作成、変更、削除されますが、場合によっては管理操作が必要になることがあります。

ClearCase レジストリの詳細については、第 3 章を参照してください。

## VOB

ClearCase コミュニティのソフトウェア成果物のパーマネント リポジトリは、1 つまたは一般に複数のバージョン付きオブジェクト ベース (以下 VOB) で構成されます。このリポジトリは任意の VOB サーバー ホスト上に配置できます (ClearCase LT コミュニティでは ClearCase LT サーバーのみが VOB サーバー ホストになります)。各 VOB は VOB 記憶ディレクトリの形で実装され、そのディレクトリに組み込みデータベースと関連するファイルが保存されます。

ネットワークの各 VOB 管理タスクとしては次のものがあります。

- **VOB の作成:** VOB はすべてのユーザーが作成できますが、コミュニティのすべてのメンバーに重要な成果物を保存するための VOB は、通常 ClearCase 管理者が作成し管理します。VOB の作成方法と管理方法については、第 6 章を参照してください。
- **データのインポート:** ClearCase には、ほかの構成管理システムからデータをインポートするためのさまざまなツールがあります。管理者は、このようなデータを VOB にインポートして、コミュニティでできるように編成します。VOB へのデータのインポート方法については、第 13 章を参照してください。
- **バックアップと復元:** VOB は、特別なバックアップ要件と復元要件が定められた重要なデータ リポジトリです。VOB のバックアップと復元の作業をタイミングよく確実に実施することは、ClearCase 管理者の最も重要な責務の 1 つです。バックアップと復元の方法については、第 10 章と第 11 章を参照してください。
- **ネットワーク アクセスの管理:** すべての VOB のネットワーク アクセス情報は、第 3 章で説明する ClearCase レジストリに保存されます。通常のネットワークでは、レジストリのメンテナンスはほとんど必要ありません。VOB を作成するコマンドを実行すると、レジストリも自動的に更新されます。このネットワーク アクセス情報を変更する必要はありません。ただし、VOB を移動した場合、VOB サーバー ホスト名を変更した場合、UNIX コンピュータと Windows コンピュータが混在したネットワークで VOB アクセスを有効にした場合は変更してください (第 4 章を参照)。

- **ユーザー アクセスの管理:** VOB には、それぞれ固有の所有者、プライマリ グループ、オブジェクトの補足グループ リスト、保護モードがあります。これらの情報を総合して、VOB データへのアクセスを制御します。VOB アクセス制御について理解し管理することは、ClearCase 管理者の重要な責務の 1 つです。詳細については、第 5 章を参照してください。
- **定期的メンテナンス:** VOB 管理者は、重要なデータを保持しつつも、ディスク領域を節約するという相反する要求のバランスを絶えず保つ必要があります。ClearCase には、ディスク領域から使用データを収集し、不要データを随時スクラブする (整理する) ツールが用意されています。また、ジョブ スケジューリング サービスや、VOB データの記憶領域を定期的に監視して管理し VOB データの一貫性を保つためのツールもあります。定期メンテナンス タスクについては、第 12 章で詳しく説明します。

## ビュー

ClearCase コミュニティのすべてのユーザーは、1 つまたは複数のビューにアクセスして作業を行います。ビューは、次のような重要な機能を提供します。

- VOB データへのアクセス
- VOB データを変更するためのワークスペース
- 開発プロセスの途中で作成されたデータを保存するための一時記憶領域

標準のビューには、バージョン付きの成果物 (VOB エLEMENT の各バージョン) とバージョンなしの成果物 (VOB に存在しないビュープライベート ファイル) が表示されます。

次の 3 種類のビューがあります。

- **スナップショット ビュー:** 指定したバージョンの VOB ELEMENT のコピーがビュープライベート オブジェクトと共に表示されます。ほかのビューから VOB にチェックインされた新しいバージョンにアクセスするには、スナップショット ビューを手動で更新する必要があります。
- **Web ビュー:** スナップショット ビューとほとんど同じですが、このビューには ClearCase Web インターフェイスからアクセスします。
- **動的ビュー (ClearCase LT ではサポートされません):** VOB 内の各ELEMENT バージョンとビュープライベート オブジェクトに対する透過的なアクセスを提供します。動的ビューでは、ビューの構成仕様で選択したバージョンがチェックインされると同時に、そのバージョンのELEMENT にアクセスできます。

一般に、VOB は長期間存続するリポジトリとして管理者が作成し管理しますが、ビューは存続期間がそれほど長くなく、通常は開発者が個別に作成します。ビューの管理作業は VOB の管理より簡単で、随時行うバックアップや定期メンテナンス、アクセス コントロール問題の対処などがあります。ビューとビューの管理の詳細については、第 9 章と **mkview** リファレンス ページを参照してください。

## ユーザーとグループの管理

---

ClearCase では、ユーザーとグループに基づいて VOB とビューへのアクセスをコントロールします。ClearCase アプリケーションから、VOB またはビューに対して何らかの操作を実行しようとする、アプリケーションの認証情報 (アプリケーションを起動したユーザーの名前とプライマリ グループ) が評価され、その操作を許可するかどうか判断されます。

ユーザーが ClearCase ホストのオペレーティング システムにログオンするという資格は、ユーザーの ClearCase 認証情報で確立されます。ユーザー名、グループ名、各ユーザーのグループメンバシップは、コミュニティ内のすべての ClearCase ホストで整合性を保つ必要があります。この整合性は、通常、Windows ドメインや UNIX Network Information System (NIS) などのネットワークワイド ユーザー/グループ アカウント データベースによって確保されます。ユーザーが UNIX ホストと Windows ホストから VOB とビューの共通セットにアクセスする環境では、両方のプラットフォームで整合性を確保する必要があります (ClearCase ユーザーが使用するユーザー名とグループ名、各ユーザーのプライマリ グループの割り当てが UNIX 環境と Windows 環境で一致する必要があります)。

混在環境でのユーザー名とグループ名については、59 ページの「共通なユーザー名とグループ名」を参照してください。ユーザーの認証情報による VOB とビューへのアクセス コントロールの仕組みについては、第 5 章を参照してください。

**メモ:** 特定のユーザーやグループには、UNIX での root や Windows での管理者グループ メンバーなど、通常はスーパーユーザーに与えられる権限と共に、ClearCase によって VOB データやその他の保護されたオブジェクトに無制限でアクセスできる権限が与えられます。詳細については、74 ページの「権限を持つユーザーとグループ」を参照してください。

## ClearCase サーバー プロセス

---

ClearCase は分散型アプリケーションです。操作の多くは、複数のホスト コンピュータと、各ホストでの複数のサーバー プロセスにより実行されます。ここでは、ClearCase サーバーのプロセスについて説明します。

通常は、これらのサーバーを明示的に設定、開始、停止、管理する必要はありません。ここで説明する情報は、サーバーによって生成されるステータスやエラー メッセージの理解、サーバー ログへのアクセス、ClearCase ネットワークでそれぞれのサーバー プロセスが果たす役割の理解に役立ちます。

ローカル VOB とローカル ビューをサポートするように設定されたホストでは、そのホストで作成された VOB またはビューかどうかに関わらず、`albd_server` と `admin_server` の 2 つのサーバー プロセスが実行されます。その他のサーバー プロセスは、そのホストに存在する VOB やビューを管理するために必要に応じて実行されます。

**メモ:** ClearCase LT クライアントには、ローカル VOB またはローカル ビューのサポート機能を設定できません。ClearCase サーバー プロセスは、ClearCase LT クライアント ホストや、ローカル VOB とローカル ビューのサポート機能が設定されていない ClearCase ホストでは実行されません。

## albd\_server

albd\_server は、VOB とビューをサポートするホストで次のようなさまざまなタスクを処理します。

- 必要に応じてほかの ClearCase サービスを開始または停止します。
- ClearCase クライアントとサーバーの間のネットワーク通信をセットアップします。
- ClearCase スケジューラから実行されるタスクの実行を管理します。
- ClearCase レジストリ サーバー ホストのレジストリ情報に対する要求に応答します。
- ClearCase ライセンス サーバー ホストのライセンス情報に対する要求に応答します (ClearCase LT では albd\_server をこの機能で使用できません)。
- UNIX ホストで、リモート clearmake プロセスからの負荷分散クエリーに応答します (この機能は ClearCase LT ではサポートされません)。

ClearCase を起動すると、albd\_server が最初に起動し、以下のいずれかのユーザー ID で稼働します。

- UNIX の root
- Windows ベースの ClearCase LT の組み込み LocalSystem の識別情報
- Windows ベースの ClearCase で作成された特別な権限を持つユーザー アカウント

ClearCase を停止すると、albd\_server はホスト上のすべての ClearCase サービスを停止して終了します。

**メモ:** ローカル VOB やローカル ビューをサポートしない ClearCase LT クライアントまたは ClearCase クライアントには、albd\_server がインストールされません。停止または開始するための ClearCase サービスはほかにはありません。

クライアント プログラムから ClearCase サーバー ホストのサービス (VOB サーバーまたはビュー サーバーなど) にアクセスする場合は、リモート プロシージャ コール (RPC) を使用してそのホストの albd\_server プロセスに要求を送信します。albd\_server は、要求されたサービスが開始されていない場合はそのサービスを開始し、クライアントに応答を発行してそのサービスのポート番号を通知します。クライアントは、このポート番号を使用して、albd\_server を経由せずにサービスを直接操作します。

albd\_server によって開始された ClearCase サービスは、通常、UNIX のリソース (VOB またはビュー) 所有者の ID、Windows の albd\_server の ID によって実行されます。

## ポート割り当て

albd\_server は、Internet Assigned Numbers Authority で指定されている専用汎用ポート (ポート 371) で RPC をリッスン (受信を監視) します。ClearCase インストール プログラムは、ほかのサービスが UDP 通信または TCP 通信でこのポートを使用していないことを確認します。競合が検出された場合は、競合しているサービスの使用ポートを変更することをお勧めします。競合しているサービスの再設定または削除が不可能な場合は、albd\_server で別の UDP ポートを使用するように設定する必要があります。

- UNIX では、ローカル ホストのサービス データベースか NIS サービス マップ、またはその両方を編集します。
- Windows では、%SystemRoot%\System32\drivers\etc\services ファイル内に適切なエントリを作成します (このファイル内に albd\_server に対応するエントリがない場合は、ポート 371 が使用されます)。

**メモ:** コミュニティ内のすべての albd\_server プロセスは同じポート番号を使用する必要があります。任意の ClearCase ホストの albd\_server ポート割り当てを変更した場合は、すべての ClearCase ホストのポート割り当てを変更する必要があります。

albd\_server は、起動時に albd.conf 設定ファイルを読み込んで、提供するサービスを決定します。このファイルを変更しないでください。

## admin\_server

ClearCase 管理サーバー (admin\_server) は、必要に応じてホストの albd\_server プロセスで起動されます。この使用期間の短いサーバーは、そのほかの管理サポート機能を実行します。

- getlog コマンドでサーバー ログ ファイルを取得し、ClearCase 管理コンソールに表示します。
- ClearCase 管理コンソールの要求に応じて、ローカル ホストの ClearCase プロパティを取得して変更します。
- ホストがバックアップ レジストリ サーバーの場合は、レジストリ ファイルを移動してクライアントを再設定します。

## credmap\_server

認証マッピング サーバーの credmap\_server は、ローカル VOB とローカル ビューをサポートするように設定された任意の ClearCase ホストで実行されます。このサーバーは、ユーザーが UNIX コンピュータと Windows コンピュータの両方から VOB とビューの共通セットにアクセスする環境で認証マッピングを処理します。詳細については、60 ページの「認証マッピング」を参照してください。

## view\_server

**view\_server** は、特定のビューのアクティビティを管理する長時間稼動プロセスです。**view\_server** は、クライアントがビューへのアクセスを要求した場合にホストの **albd\_server** プロセスによって開始されます。**view\_server** は、**cleartool endview -server** コマンド、システムのシャットダウン、**view\_server** プロセスを終了させるオペレーティング システムのシステム コマンドのいずれかで停止されるまでアクティブな状態を維持します。

**view\_server** は、実行されると、ビュー記憶ディレクトリの **.view** ファイルから設定情報を読み込みます。このファイルの値は **mkview** や **chview** などのコマンドによって設定します。このファイルをテキスト エディタで変更しないでください。

## vob\_server

各 VOB に対して、長期間稼動する **vob\_server** プロセスが VOB ホストで実行されます。**vob\_server** は、クライアント プロセスの要求に応じて VOB の記憶プールにあるデータを操作します。

**vob\_server** は、VOB データ コンテナの作成や削除を行う唯一のプロセスです。VOB 所有者や権限を持つユーザーのみが VOB データ コンテナと記憶プールを変更できます。詳細については、109 ページの「VOB 記憶ディレクトリ」を参照してください。

**vob\_server** プロセスは、必要に応じて **albd\_server** によって開始されます。次のイベントのいずれかが発生するまで、アクティブな状態を維持します。

- **rmvob** コマンドによる VOB の削除
- VOB サーバー ホスト上での **ClearCase** の停止
- VOB サーバー ホストのシャットダウンと再起動

**vob\_server** は、実行されると、VOB 記憶ディレクトリの **vob\_server.conf** ファイルから設定情報を読み込みます。このファイルの値は、**vob\_snapshot\_setup** ユーティリティやその他のコマンドで設定します。このファイルを変更しないでください。

## db\_server

ホストの **db\_server** プロセスは、クライアント プログラムの要求に応じて、ホスト上の VOB データベース トランザクションを管理します。クライアント プログラムは VOB データベースに直接アクセスできないため、VOB データやメタデータの作成、読み込み、変更が必要な場合は、データベース トランザクション要求を **db\_server** プロセスに送信しなければなりません。

各 **db\_server** プロセスは、1 度に 1 つのクライアントにしかサービスを提供できませんが、必要な数だけ **VOB** を操作できます。クライアントは、**VOB** ホストの **albd\_server** を利用して **db\_server** との接続を確立します。必要な場合は、**albd\_server** が新しい **db\_server** プロセスを開始して要求を処理します。クライアントを終了したり、アイドル状態 (データベース トランザクションの要求が一定期間途切れた状態) になると、接続が切断されます。このとき、別のクライアントで **db\_server** を利用できるようになります。アイドル期間が終了すると、接続されていない **db\_server** がホストの **albd\_server** によって終了します。

## vobrpc\_server

各 **VOB** サーバー ホストは、内部にある各 **VOB** に対して 1 つまたは複数の **vobrpc\_server** プロセスを実行します。各 **vobrpc\_server** プロセスは、ネットワーク全体の **view\_server** プロセスの要求を処理します。これらの要求では、メタデータ (**VOB** データベース) とファイル システム データ (記憶プール) アクティビティの両方を生成できます。**vobrpc\_server** は **db\_server** と同じ方法 (記憶プールへのアクセス要求を **vob\_server** に送る) で **VOB** データベースにアクセスします。

**vobrpc\_server** プロセスは **albd\_server** によって開始されます。このコマンドは、新しい要求を最も負荷の低いサーバーに送り、システムに負荷がかかっている場合は内部 **vobrpc\_server** プロセスを停止します。

## lockmgr

各 **VOB** サーバー ホストは、**Lock Manager** プロセスの **lockmgr** を 1 つだけ実行します。このコマンドは、対象ホスト上のすべての **VOB** データベースに対するトランザクション要求を調停します。呼び出しプログラムは **lockmgr** をポーリングし、要求されたデータへのアクセスを許可するか制限します。データを利用できる場合は、トランザクションが直ちに処理されます。データの読み込みまたは書き込みを行い、呼び出しプログラムに出力が返されます。データを利用できない場合は (別の呼び出しプログラムによってデータの書き込みアクセスが許可されているため、データがロックされている場合)、呼び出しプログラムは **lockmgr** がデータのアクセスを許可するまで待機します。

ほとんどの **ClearCase** サービスとは異なり、**lockmgr** は **albd\_server** で開始されず、**VOB** ホストが起動したときに開始されます。特定の設定に対して **VOB** サーバーのパフォーマンスを上げる必要がある場合は、**Lock Manager** 起動オプションを変更できます。詳細については、308 ページの「**Lock Manager** の起動オプションの変更」を参照してください。

## rwp

**Rational Web** プラットフォームの **rwp** は、**ClearCase Web** インターフェイス用のアプリケーション サポートを提供します。通常、**RWP** サーバーはインストール時に設定され、特殊な設定 (プロキシによるアクセスをサポートするための設定など) が必要でない限り、管理する必要はありません。**ClearCase** コミュニティには 1 つまたは複数の **rwp** サーバーがあります。詳細については、付録 C を参照してください。

## サーバー ログ

---

各 ClearCase サーバー プロセスは、ホスト上にログを保存します。ローカル ホストで入力したコマンドからエラーが返されると、リモート ホストにログ エントリが生成されるため、UNIX で稼動する ClearCase は、`cleartool getlog` コマンドと ClearCase 管理コンソールのサーバー ログ ノードの 2 つのリモート ログ取得ツールを提供します (Windows では、ClearCase サーバー プロセスがローカル ホストの Windows イベント ログに書き込まれます。このイベント ログは任意の Windows ホストで Windows イベント ビューアを使用して見ることができます)。

### UNIX でのログ

UNIX 環境の ClearCase ホストでは、ログ ファイルが `/var/adm/rational/clearcase/log` ディレクトリに保管されます。ログ ファイルには、さまざまなサーバー プログラムやユーザー プログラムのエラーとステータス情報が記録されます。次のようなログ ファイルがあります。

<code>abe_log</code>	<code>clearmake</code> の並行ビルド時に監視付きビルドエグゼキュータ ( <code>abe</code> ) で使用されます (ClearCase LT ではサポートされません)。
<code>albd_log</code>	<code>albd_server</code> で使用されます。
<code>db_server_log</code>	<code>db_server</code> で使用されます。
<code>error_log</code>	汎用エラー ログ。 <code>cleartool</code> などのユーザー プログラムで使用されます。
<code>event_scrubber_log</code>	<code>event_scrubber</code> プログラムで使用されます。
<code>export_mvfs_log</code>	<code>export_mvfs</code> プログラムで使用されます (ClearCase LT ではサポートされません)。
<code>install_log</code>	<code>install_release</code> (インストール スクリプト) で使用されます
<code>lockmgr_log</code>	<code>lockmgr</code> プログラムで使用されます。
<code>mntpc_server_log</code>	<code>mntpc_server</code> プログラムで使用されます。 <code>cleartool mount</code> で要求される MVFS マウントを実行します (ClearCase LT ではサポートされません)。
<code>msadm_log</code>	MultiSite 管理サーバーで使用されます。マスターシップの要求を処理します (ClearCase LT ではサポートされません)。
<code>promote_log</code>	<code>promote_server</code> プログラムで使用されます (ClearCase LT ではサポートされません)。
<code>scrubber_log</code>	<code>scrubber</code> プログラムで使用されます。
<code>view_log</code>	<code>view_server</code> で使用されます。
<code>vob_log</code>	<code>vob_server</code> で使用されます。
<code>vob_scrubber_log</code>	<code>vob_scrubber</code> プログラムで使用されます。
<code>vobrpc_server_log</code>	<code>vobrpc_server</code> で使用されます。

ログ ファイルは一般的なテキスト ファイルです。標準的なエントリには、イベントの日付と時刻、イベントが発生したソフトウェア モジュール、現在のユーザー、イベント特有のメッセージなどがあります。

エントリが累積されるにしたがって、ログ ファイルも大きくなります。デフォルトの設定では、ClearCase スケジューラによって、ログ ファイルの名前を `logfile_name.old` に変更し、同じ場所に空のテンプレート ファイルを作成するジョブが定期的に行われます。

## Windows でのログ

Windows では、ClearCase プログラムが情報、警告、エラー メッセージを Windows アプリケーション イベント ログに書き込みます。これらのメッセージのソースは ClearCase として Windows イベント ビューアに表示されます。標準的なイベント ログのエントリは、エラーの日付と時刻、エラーが発生したソフトウェア モジュール、現在のユーザー、エラー特有のメッセージなどがあります。

MVFS がインストールされている ClearCase ホストでは、MVFS がエラーとステータス メッセージを `C:\mvfslogs` のファイルに記録します。このパス名は、[コントロール パネル] から ClearCase プログラムを起動して表示される [MVFS] タブで変更できます。デフォルトの設定では、ClearCase スケジューラによって、7 日以上経過している古い MVFS ログ ファイルを削除するジョブが定期的に行われます。スケジュールされているジョブの記述と変更については、[schedule](#) リファレンス ページを参照してください。

## ClearCase の開始と停止

---

ClearCase は、通常、ClearCase ホストの起動やシャットダウンに合わせて開始または停止されます。このとき、プラットフォーム タイプに応じた起動規則またはシャットダウン規則が適用されます。ここでは、ClearCase サービスを手動で停止または開始する手順について説明します。

**メモ:** ClearCase サーバー プロセスは ClearCase LT クライアントでは実行されません。

## UNIX 環境の場合

UNIX のブートストラップ時に、UNIX の `init` プログラムによって ClearCase 起動スクリプト `ccase-home-dir/etc/clearcase` が実行されます。起動スクリプトでは、次の処理が実行されます。

- そのホストの `albd_server` プロセスを開始します。
- ホスト上に VOB がある場合は `lockmgr` プロセスを開始します。
- そのほかの設定タスク (パブリック VOB のマウントなど) を実行します。

root ユーザーの場合は、**clearcase** スクリプトを実行して **ClearCase** をいつでも停止または開始できます。例を次に示します。

```
# ccase-home-dir/etc/clearcase stop
```

**ClearCase** を停止します。

```
# ccase-home-dir/etc/clearcase start
```

**ClearCase** を開始します。

詳細については、**init\_ccase** リファレンス ページを参照してください。

## Windows 環境の場合

明示的に手動で開始するように設定されていない限り、**ClearCase** ホストの起動時に、**lockmgr** サービスと **albd\_server** サービスが開始されます。MVFS がインストールされている場合は MVFS も同時に起動されます。ローカル ホストの管理者またはパワー ユーザー グループのメンバーは、Windows の [コントロール パネル] から **ClearCase** プログラムを起動し、表示される [サービス起動] タブを使用して、ホスト上で **ClearCase** を停止または開始できます。

- 1 [スタート] メニューから [設定] をポイントして [コントロール パネル] をクリックし、**ClearCase** プログラムをダブルクリックして開きます。
- 2 [サービス起動] タブをクリックします。このタブには、**albd\_server** のステータスと、ホストがローカル VOB をサポートしている場合は、**Lock Manager** のステータスが表示されます。MVFS がインストールされているホストでは、このタブに MVFS と MVFS の認証情報キャッシュを管理する **Credentials Manager** のステータスも表示されます。
  - [ClearCase を停止] をクリックすると、**ClearCase** サービスが停止します。
  - [ClearCase を起動] をクリックすると、**ClearCase** サービスが開始します。

Windows の **net start** コマンドや **net stop** コマンドを使用することもできます。

- 「**net stop albd**」と入力すると、**ClearCase** サービスが停止します。
- 「**net start albd**」と入力すると、**ClearCase** サービスが開始されます。

## MVFS の開始と停止

ほかの **ClearCase** サービスとは異なり、MVFS はカーネルで実行されるため、オペレーティング システムで独立して停止または開始できません。UNIX または Windows で稼動する MVFS を停止して再開するには、コンピュータをシャットダウンして再起動する必要があります。

## ClearCase と NAS (Network-Attached Storage) デバイス

---

Network-Attached Storage (以下 NAS) デバイスは、UNIX の Network File System (NFS) や Windows の Common Internet File System (CIFS) などのネットワーク ファイル システム プロトコルを使用して、ローカル エリア ネットワーク 上のほかのホストにデータ記憶リソースを提供します。すべての NAS デバイスを、ClearCase で作成して使用する一般的なファイルの記憶領域として使用できます (ネットワーク リリース領域や UNIX 環境の ClearCase でサポートされているリモート記憶プールなど)。さらに、Rational は一部の NAS デバイスについて、データベースを含むすべての VOB とビューを記憶するための記憶デバイスとして適切に構成され使用される場合のみ使用を認定しています。

**メモ:** ClearCase LT では、VOB またはビューを記憶するための NAS デバイスの使用をサポートしていません。

認定されている NAS デバイス内にビューと (特に) VOB の記憶領域を配置することにより、次のような利点が得られます。

- **拡張性の強化:** VOB またはビューを ClearCase サーバー ホストに保存する際に、標準で割り当てられる 物理的な容量以上の記憶容量を確保できます。
- **柔軟性の向上:** サーバー ホストを VOB またはビューの記憶領域から分離することで、サーバー ホストのアップグレードプロセスが単純になります。
- **管理の単純化:** 多くの NAS デバイスには、VOB とビューのバックアップ処理と復元処理を単純化する、強化されたバックアップ機能と復元機能が備わっています。mkstgloc コマンドを使用して NAS デバイスにサーバーの記憶場所を作成すると、VOB とビューの作成、管理がはるかに容易になります。
- **相互運用の単純化:** すべての認定 NAS デバイスは、UNIX ホストと Windows ホストの両方でネイティブ相互運用のサポートを設定できます。Windows コンピュータと UNIX コンピュータが混在するネットワークに VOB とビューの記憶領域を提供するため NAS デバイスを使用している場合は、異なるプラットフォーム間でのファイル アクセスを提供するソフトウェアを ClearCase クライアント ホストやサーバー ホストに追加インストールする必要はありません。詳細については、61 ページの「異なるプラットフォーム間のファイル システムのアクセス」を参照してください。

**注意:** リモート VOB データベースまたはリモート ビュー データベースをサポートするには、認定 NAS デバイスごとに特別に設定する必要があります。それぞれの認定デバイスの設定要件については、オンラインの『Rational ClearCase プラットフォーム別情報ガイド』を参照してください。このオンライン ガイドの説明に従って認定 NAS デバイスを設定しないと、デバイスに保存された VOB またはビューのデータが失われる可能性があります。また、認定されていない NAS デバイスに VOB またはビューのデータベースを保存した場合も、データが失われることがあります。VOB またはビューの記憶領域として NAS デバイスを使用すると、状況によっては ClearCase のパフォーマンスが低下することがあります。詳細については、311 ページの「サーバーのパフォーマンスと NAS デバイス」を参照してください。

NAS デバイスで VOB やビューを作成、移動、バックアップ、復元する場合は、特別な手順が必要になる場合があります。VOB やビューを管理したり、バックアップまたは復元する際の具体的な手順については、本書内の各章で説明します。

## NAS デバイスへのネットワーク アクセスの設定

VOB またはビューの記憶領域として使用される NAS デバイスは、信頼性の高いネットワーク上に配置する必要があります。可能であれば、この記憶領域を使用する ClearCase VOB/ビューサーバー ホストと同じサブネット上に配置することをお勧めします。ホップを使用する場合は 1 回以内にしてください。NAS デバイスへのネットワーク アクセスは、以下の手順に従って設定します。

- UNIX 環境の VOB またはビューのサーバー ホストでは、NFS ハードマウントを使用して NAS デバイスをマウントする必要があります (ほとんどのシステムのデフォルトです)。ソフトマウントを使用すると、VOB またはビューのサーバー プロセスの開始時にエラーが発生し、ログ ファイルにエラー メッセージが書き込まれます。
- Windows 環境のホストでは、`net use` コマンド (または Windows エクスプローラの [ネットワーク ドライブの割り当て]) を使用して、NAS デバイスに対するアクセスを確立する必要があります。

**メモ:** UNIX 環境の VOB またはビューのサーバー ホストと NAS デバイスの VOB またはビューのデータベースを接続する場合は、NFS プロトコルを使用する必要があります。Windows 環境の VOB またはビューのサーバー ホストと NAS デバイスの VOB またはビューのデータベースを接続する場合は、CIFS プロトコルを使用する必要があります。Rational では、Windows ホストと (ClearCase コミュニティで使用されている) NAS デバイスを接続する NFS ソフトウェアの使用をサポートしていません。

## DHCP を使用する場合

---

Dynamic Host Configuration Protocol (DHCP) はインターネット標準プロトコルです。このプロトコルを適用すると、DHCP サーバーから一時的に割り当てられる動的な IP アドレスをホストで取得できます。DHCP は Windows コンピュータや一部の UNIX コンピュータに幅広く実装されています。ClearCase は、ほとんどの DHCP 設定と互換性がありますが、特殊な要件もいくつかあるので注意してください。

ClearCase ホストは、アクセス先となる別の ClearCase ホスト (通常はサーバー) の IP アドレスをキャッシュします。DHCP によってリモート サーバーに新しい IP アドレスが割り当てられ、キャッシュ アドレスが無効になると、キャッシュ アドレスを使用する ClearCase アプリケーションでエラーが発生します。ClearCase サーバー ホストは、クライアントがキャッシュした IP アドレスを維持しようとします。これは、通常、サーバー ホストがネットワークに接続された状態で稼働しているためで、結果的にエラーが発生しにくくなります。サイトで DHCP を使用している場合は、エラーの発生率をさらに下げるために、以下のガイドラインに従ってください。

- 可能であれば、固定 IP アドレスを ClearCase サーバー ホストに割り当てます。
- DHCP リース時間 (ホストが DHCP 割り当てアドレスを使用できる時間) を、ClearCase ホストがシャットダウンしたりネットワークからログオフするまでの標準時間の最低でも 2 倍に設定します。DHCP リース時間は一般に DHCP サーバーによって確立されます。
- ClearCase ホストの再起動時に、新しい DHCP アドレスを取得するように設定しないでください。

## **ClearCase ホストに複数のネットワーク インターフェイスがある場合**

ClearCase ホスト (クライアントまたはサーバー) にネットワーク インターフェイスが 2 つ以上ある場合は、それぞれのホスト名をリストにまとめたファイルをホスト上で作成する必要があります。

- UNIX では、`/var/adm/rational/clearcase/config/alternate_hostnames` ファイルになります。
- Windows では、`ccase-home-dir\var\config\alternate_hostnames` ファイルになります。

たとえば、NIS ホスト データベースに UNIX ホスト用の次のエントリが記述されているとします。

```
159.0.10.16 widget sun-005 wid
159.0.16.103 widget-gte sun-105
```

この場合は、このホストの `alternate_hostnames` ファイルに次のエントリを追加します。

```
widget
widget-gte
```

各ホスト エントリの最初のホスト名だけがファイルに記述されている必要があります。ファイルにはホスト名を 1 行に 1 つずつ記述します。コメント機能はなく、すべての行が意味を持ちます。ホストに複数のネットワーク インターフェイスが存在しない場合は、ホスト上にこのファイルを作成しないでください。

## ClearCase と NFS 自動マウント機能

---

ここでは、ClearCase で NFS (UNIX) 自動マウント機能を使用する方法について説明します。通常、UNIX 環境のホストでは自動マウント機能を使用して、VOB やビューの記憶ディレクトリなど、リモート ホスト上にあるエクスポートされたファイル システムをマウントします。多くの ClearCase コマンドは、自動マウント機能によってマウントされた (またはマウントされる) VOB またはビューの記憶ディレクトリのグローバルパス名を取得するため、自動マウント機能がリモート記憶ディレクトリにアクセスするためのローカル マウント ポイントをどのように構築しているか理解する必要があります。

**メモ:** 自動マウントの実装はプラットフォームにより異なります。各自の UNIX プラットフォームの自動マウントに関するリファレンス情報を参照してください。

### 自動マウント機能のマップとマウント ポイント

直接または間接に自動マウント マップを使用して、リモート VOB またはリモート ビューの記憶領域にアクセスできます。グローバルパス名を構築するために ClearCase が使用するヒューリスティックは、すべてのホストが共通の自動マウント マップを使用することを前提としています。自動マウント マップを使用しないホストで VOB またはビューを作成する場合は、VOB またはビューに対するネットワーク アクセスを確保するために別の手順が必要になります。詳細については、107 ページの「UNIX の自動マウント機能でホスト マップを使用していない場合」を参照してください。

リモート VOB またはリモート ビューの記憶領域に対するグローバルパスを構築する場合、ClearCase は、標準 NFS マウントポイントである /net、/hosts、/nfs を順にパス名に使用して、リモート ホストにアクセスしようとします。たとえば、mars というホストに VOB 記憶領域へのグローバルパスを構築する場合、ClearCase はまず /net/mars のパス名を設定し、mkvob コマンドまたは VOB 作成ウィザードで指定した VOB 記憶ディレクトリの名前を追加します。このパスで目的の記憶ディレクトリにアクセスできない場合は、別の標準マウント ポイントで試みます。

### 非標準マウント ポイントの指定

環境変数の CCASE\_GPATH\_HINTS を使用すると、ClearCase が標準マウント ポイントの前に使用する 1 つまたは複数のマウント ポイントを指定できます。この環境変数に、ClearCase がリモート UNIX ホスト上の VOB またはビューの記憶領域に対するグローバルパスを構築するときに使用するマウント ポイント リストの値を、コロンで区切って設定します。たとえば、次のように設定します。

```
CCASE_GPATH_HINTS="/vob_servers:/view_servers"
```

ClearCase は、リモート NFS ディレクトリに対するパス名をまず `/vob_servers` マウント ポイントで構築しようとします。このマウント ポイントで有効なグローバルパスを構築できない場合は、`/view_servers` を使用します。`CCASE_GPATH_HINTS` に指定したマウント ポイントで有効なグローバルパスを構築できない場合、ClearCase は標準マウント ポイントを使用します。

**メモ:** `CCASE_GPATH_HINTS` の値を "" (Null) に設定すると、VOB またはビューを作成する `cleartool` コマンドの実行時に、コマンド行で指定した明示的なグローバル記憶パスを強制的に使用するようになります。詳細については、`env_ccase`、`mkvob`、`mkview`、`mkstgloc` のリファレンス ページを参照してください。

## 非標準マウント ディレクトリの指定

ほとんどの UNIX プラットフォームでは、自動マウント機能によってリモート ファイル システムが標準マウント ポイントのいずれかで直接マウントされます。一部の自動マウント機能は、リモート ファイル システムを `/tmp_mnt` でマウントし、`/net` または標準マウント ポイントのいずれかから `/tmp_mnt` に対する UNIX シンボリック リンクを使用してリモート ファイル システムにアクセスします。ClearCase ホストの自動マウント機能によって別のマウント ポイントに対するシンボリック リンクが作成される場合は、`/var/adm/rational/clearcase/config/automount_prefix` ファイルでマウント ポイントの名前を指定する必要があります。たとえば、自動マウント機能が `/tmp_mnt` ではなく `/autom` を使用する場合は、次の行を `automount_prefix` ファイルに追加します。

```
/autom
```

# ライセンス方式と ライセンス管理

# 2

Rational ClearCase と Rational ClearCase LT を実行するには、ライセンスが必要です。この章では、両製品のライセンス方式と、それらのライセンスの管理手順について説明します。

## ClearCase のライセンス方式と Rational Common Licensing

---

ClearCase と ClearCase LT では、採用しているライセンス方式が異なります。ClearCase は独自のフローティング ライセンス方式を採用していますが、ClearCase LT では Rational Common Licensing が採用されています。両方のライセンスは (必要とされる状況、付与方法、有効期間など) 類似しています。ただし、どちらも独立したシステムとして機能しており (つまり、ClearCase のホストが ClearCase LT のライセンスを使用することや、通常の場合で ClearCase LT のホストが ClearCase のライセンスを使用することはできません)、各ライセンス方式の管理に必要な条件も異なります。

ClearCase LT のライセンス方式の詳細については、29 ページの「ClearCase LT の Rational Common Licensing」を参照してください。

## ClearCase のライセンス方式

---

ClearCase では、アクティブなユーザーを基準としたフローティング ライセンス方式を使用しています。ユーザーは、ClearCase を使用するためにライセンスを取得する必要があります。ライセンスを取得すると、同じライセンス サーバーがサービスを提供する任意のホストで、ClearCase を使用する権限が与えられます。実際には、ユーザーが **cleartool** や GUI プログラムなどの ClearCase クライアントユーティリティを実行すると、それらのユーティリティがライセンスの取得を試みます。ライセンスが取得されると、ユーザーはそのライセンスを一定期間保持することができます。この期間は、ClearCase のコマンドを入力した時点で自動的に延長されます。ただし、一定期間 (デフォルトでは 60 分) 内に ClearCase のコマンドを入力しないと、そのライセンスは解放され、別のユーザーが取得できるようになります。

ClearCase ライセンス サーバー ホストとして、LAN 上の 1 台以上のホストを指定できます。各ライセンス サーバー ホストは、ライセンス キーのリストを管理します。詳細については、27 ページの「ライセンス データベースの形式」を参照してください。ユーザーが ClearCase コマンドを実行したり、GUI を使用したりするたびに、ライセンスの照合確認が行われます。ユーザーが既にライセンスを取得している場合は、照合が完了し、コマンドや GUI 操作が通常どおりに実行されます。ユーザーがライセンスを取得していない場合は、次の処理が実行されます。

- 1 ClearCase クライアント ソフトウェアが、ホスト名を基にライセンス サーバー ホストを検索します。
- 2 検出されたライセンス サーバー ホスト上のライセンス サーバー プロセスで、要求されたコマンドや GUI 操作に関して通信が行われ、ライセンスを取得できるユーザーがどうかを確認できます (このライセンス サーバー プロセスは、実際には `albd_server` を実行します。このプロセスは、この処理だけでなくほかのタスクも実行します)。
- 3 ライセンスが取得可能な場合は、ライセンス サーバーが該当ユーザーにライセンスを付与し、コマンドや GUI 操作を実行できるようにします。
- 4 ライセンスを取得できない場合、ライセンス サーバーはその旨を通知するメッセージを表示します。要求されたコマンドや GUI 操作は実行されず、代わりにエラー メッセージが表示されます。

## ライセンス取得の優先順位

ライセンス データベース ファイルでは、各ユーザーにライセンス取得の優先順位を割り当てることができます (この割り当ては必須ではありません)。-user 行に指定されている各ユーザーに優先順位が設定されます。具体的には、1 行目のユーザーに優先順位 1 (最高)、2 行目のユーザーに優先順位 2、... という順番になります。どの -user 行にも指定されていないすべてのユーザーには、共通して最低の優先順位が設定されます。

## ライセンス レポート ユーティリティ

`clearlicense` ユーティリティは、ライセンス データベース ファイルに定義されているライセンスと、現在のユーザー アクティビティに関するレポートを生成します。またこのユーティリティを使用して、特定のユーザーが保持しているライセンスを強制的に解放させ、ほかのユーザーが使用できるようにすることもできます。

## ライセンス サーバー ホストの指定

各 ClearCase ホストを、特定の 1 台のライセンス サーバー ホストからライセンスを取得するように設定する必要があります。通常、この設定はインストール時に行われ、ClearCase 管理者が指定した各サイトのデフォルト値が設定されます。別のライセンス サーバーを使用するように ClearCase ホストの設定を変更するには、次のいずれかの処理を実行します。

- UNIX の場合、ライセンス サーバー ホストの名前は `/var/adm/rational/clearcase/config/license_host` ファイルに記述されています。このファイルを任意のテキスト エディタで編集し、新しいライセンス サーバー ホストの名前を指定します。
- Windows の場合、ライセンス サーバー ホストの名前は、[コントロール パネル] の [ClearCase] プログラムを実行し、[ライセンス機能] タブの [次のホストのライセンス サーバーを使用] に指定します。

ClearCase 環境に UNIX と Windows プラットフォームが混在している場合でも、Windows コンピュータまたは UNIX コンピュータのいずれか 1 台を両方のプラットフォームに対応するライセンス サーバー ホストとして指定できます。

**メモ:** UNIX と Windows の両方に ClearCase ライセンスを提供するサーバーとして Windows コンピュータを使用する場合、この Windows ライセンス サーバー ホストは、ライセンスを提供する UNIX ユーザーすべてのユーザー アカウントが定義されているドメインのメンバである必要があります。詳細については、59 ページの「共通なユーザー名とグループ名」を参照してください。

## ライセンス サーバーの設定

---

ライセンス サーバー ホストの設定は、ClearCase のインストール中に行われます。ライセンス サーバーの初期設定と詳細については、『Rational ClearCase 製品ファミリーのインストール ション ガイド』を参照してください。

## ライセンス データベース

ライセンス データベースには、ライセンス キーやその他のライセンス管理情報が格納されています。

- Windows ホストの場合、ライセンス データベースは Windows のレジストリに保存されています。修正する場合は、[コントロール パネル] の [ClearCase] プログラムを実行して [ライセンス機能] タブを使用してください。
- UNIX ホストの場合、ライセンス データベースはテキスト ファイル `/var/adm/rational/clearcase/license.db` に記述されており、任意のテキスト エディタで編集可能です。

**メモ:** この `license.db` ファイルのすべての行の末尾には、改行文字を挿入する必要があります。

## 既存のライセンス サーバー ホストへの新規ライセンスの追加

組織によっては、ClearCase のライセンスを購入する際、まず最初に一定数を購入し、必要に応じて後から追加のライセンスを購入していく場合があります。このような場合、新しいライセンス セットを新しく購入するたびに、新しいライセンスの承認コード (ライセンス キーと呼ばれます) が 1 つ提供されます。このキーは英数字で構成された文字列で、電子メールまたはファックスで Rational 社より送信されます。この文字列をライセンス データベースに正しく入力する必要があります。大文字と小文字も区別されるため、注意してください。

- 1 **ライセンス サーバー ホストのホスト名を確認します。**ライセンス サーバー ホストの名前がわからない場合は、任意の ClearCase クライアント ホスト上で確認できます。
  - UNIX ホストの場合、ライセンス サーバー ホストの名前は、  
`/var/adm/rational/clearcase/config/license_host` ファイルに記述されています。
  - Windows ホストの場合、ライセンス サーバー ホストの名前は、[コントロール パネル] から [ClearCase] プログラムを起動して確認できます。Windows の [スタート] メニューから、[設定]、[コントロール パネル]、[ClearCase] プログラムの順にクリックします。[ライセンス機能] タブをクリックし、ライセンス サーバー ホスト名を表示します。
- 2 **ライセンス サーバー ホストにログインします。**UNIX ホストの場合、ライセンス データベースを修正できるのは root ユーザーのみです。Windows ホストの場合、ライセンス データベースを変更するには、該当ホストの Windows レジストリを編集する権限を持っている必要があります。
- 3 **ライセンスを追加します。**
  - UNIX ホストの場合、任意のテキスト エディタを使用して、`license_db` ファイルの最後に新しいライセンス キーを追加します。
  - Windows ホストの場合は、Windows の [スタート] メニューから、[設定]、[コントロール パネル]、[ClearCase] の順にクリックし、ClearCase プログラムを起動します。[ライセンス機能] タブをクリックし、[The local system can act as a license server] チェック ボックスをオンにします。[License Keys] ボックスにライセンス キーを入力します。[適用] または [OK] をクリックします。

## 追加ライセンス サーバー ホストの設定

ClearCase の新しいライセンス セットを購入した際、それらのライセンスを既存のライセンス サーバー ホストに追加するのではなく、別のホストに登録できます。複数のライセンス サーバー ホストを使用することで、ライセンスの提供元を分散することができ、万一の障害にも対処できます。

**メモ:** 上記の判断は、ClearCase 製品ファミリーのライセンス フォームを Rational 社に送付する前に行わなければなりません。これは、このフォームに、ライセンス サーバー として使用するホストのマシン ID を記入する必要があるためです。

新しいライセンス サーバー ホストを設定するには

- 1 **新しいライセンス サーバー ホストのマシン ID を確認します。** ClearCase のライセンスは、ホストの IEEE MAC (Media Access Control) アドレスを基に特定のライセンス サーバー ホストに関連付けられます。この MAC アドレスは、ホストのネットワーク インターフェイスに割り当てられている固有の識別子 (ID) です。この マシン ID は、次のように、**clearlicense** コマンドを使用して表示できます。

**clearlicense -hostid**

現在ののマシン識別子 : 00c02e0931cf

Windows ホストの場合、次の方法でも確認できます。Windows の [スタート] メニューから、[設定]、[コントロール パネル]、[ClearCase] の順にクリックし、ClearCase プログラムを起動します。[ライセンス機能] タブをクリックし、ホストのマシン ID を表示します。

ClearCase をまだインストールしていないホストのマシン ID を確認する必要がある場合は、ライセンス フォームに記載されている手順に従ってください。

- 2 **ClearCase 製品ファミリーのライセンス フォームに必要事項を記入します。** このフォームのコピーは、『Rational ClearCase 製品ファミリーのインストレーション ガイド』の巻末に収録されています。記入済みのフォームをファックスで Rational 社まで送信してください。Rational 社から新しいライセンス キーを返信します。
- 3 **新しいライセンス サーバー ホストにログインします。** UNIX ホストの場合、ライセンス データベースを修正できるのは root ユーザーのみです。Windows ホストの場合、ライセンス データベースを変更するには、該当ホストの Windows レジストリを編集する権限を持っていることが必要です。
- 4 **新しいホストのライセンス データベースにライセンスを追加します。**
  - UNIX ホストの場合、任意のテキスト エディタを使用して、**license\_db** ファイルの最後に新しいライセンス キーを追加します。
  - Windows ホストの場合は、Windows の [スタート] メニューから、[設定]、[コントロール パネル]、[ClearCase] の順にクリックし、ClearCase プログラムを起動します。[ライセンス機能] タブをクリックし、[The local system can act as a license server] チェック ボックスをオンにします。[License Keys] ボックスにライセンス キーを入力します。[適用] または [OK] をクリックします。
- 5 **ライセンス サーバー ホストの ClearCase を終了し、再起動します。**

- 6 新しいライセンス サーバー ホストを使用するように、ClearCase クライアント ホストを再設定します。UNIX ホストの場合、`/var/adm/rational/clearcase/config/license_host` ファイルを編集し、新しいライセンス サーバー ホストの名前を設定します。Windows ホストの場合は、Windows の [ スタート ] メニューから、[ 設定 ]、[ コントロール パネル ]、[ ClearCase ] の順にクリックし、ClearCase プログラムを起動します。[ ライセンス機能 ] タブをクリックし、[ Hostname of the license server ] ボックスに新しいライセンス サーバー名を入力します。[ 適用 ] または [ OK ] をクリックします。

ClearCase 管理コンソールを使用して、ネットワーク内の ClearCase ホストが使用するライセンス サーバーを変更することもできます。その手順は次のとおりです。

- 1 ライセンス サーバーを変更するホストのホスト ノードに移動します。このノードは、管理コンソールの [ ホスト ] ノードや [ ClearCase ネットワーク ] ノードのサブノード、または [ ClearCase ホスト管理 ] に表示される最上位ノードのいずれでもかまいません。
- 2 [ 操作 ] メニューの [ プロパティ ] をクリックします。表示されるダイアログ ボックスで、ホストのライセンス サーバーを変更できます。リモート ホストが使用するライセンス サーバーを変更するには、まずそのホストがリモートで管理できるように設定されていなければなりません。また、リモート管理権限を持つユーザーでなければ、サーバーの変更はできません。

**メモ:** 異なるライセンス サーバーを使用する複数の ClearCase ホストにユーザーがアクセスする場合、そのユーザーは各サーバーからライセンスを取得する必要があります。同じライセンス サーバーを使用する複数の ClearCase ホストにアクセスする場合は、ユーザーは 1 つのライセンスを取得するだけです。

## 別のホストへのライセンスの移動

---

ここでは、ライセンス セットを別のホストに移動する手順について説明します。ライセンスの移動の際には、次の点に注意してください。

- ライセンス データベース ファイルに複数のライセンス セットが記述されている (つまり、複数の `license` 行が指定されている) 場合、一部のライセンス セットは移動できるが、ほかのライセンス セットを移動できないことがあります。
- ライセンス サーバー ホストの名前のみを変更する場合は、ここで説明する手順を実行する必要はありません。何らかの理由でライセンス ホストの ID (`clearlicense -hostid` で出力される) が変更された場合にのみ実行する必要があります。このような状況の原因としては、一般的に、ライセンス サーバーのホスト機能を別のコンピュータに移動した場合や、既存のライセンス サーバー ホストのネットワーク インターフェイス ハードウェアを交換した場合が考えられます。

ライセンスをほかのホストに移動するには

- 1 製品ファミリー ライセンス フォームの移動依頼に**必要事項**を記入します。このフォームのコピーは、『**Rational ClearCase 製品ファミリー インストレーション ガイド**』の巻末に収録されています。記入済みのフォームをファックスで **Rational** 社まで送信してください。**Rational** 社から置き換えるライセンス キーを返信します。
- 2 ライセンスを移動します。既にライセンス サーバー ホストとして指定されているホストにライセンスを移動する場合は、24 ページの「既存のライセンス サーバー ホストへの新規ライセンスの追加」の手順に従います。新しくライセンス サーバー ホストとして指定するホストに移動する場合は、24 ページの「追加ライセンス サーバー ホストの設定」の手順に従います。

## ライセンス サーバー ホスト名の変更

---

ライセンス サーバー ホスト名を変更するだけでは、ライセンス承認コードは無効になりません。このコードは、ネットワーク ホスト名ではなくハードウェア レベルのマシン ID に関連付けられているためです。ホスト名の変更後、一部または全部の **ClearCase** ホストのライセンス サーバー ホストの割り当てを切り替えます。詳細については、26 ページの「別のホストへのライセンスの移動」の手順 6 を参照してください。

## ライセンス データベースの形式

---

ライセンス データベースには、いろいろな種類の行が記述されています。これらの行を使用して、複数ライセンスの定義、各ユーザーのライセンス優先順位の指定、ライセンス アクティビティの監視の有効化を設定します。

### ライセンス セットを定義する行

**ClearCase** 製品ファミリーのライセンス キーは 1 行で記述し、一定数のライセンスを定義します。ほとんどのライセンスは特定のライセンス サーバー ホストに関連付けられており、26 ページの「別のホストへのライセンスの移動」に示した処理を実行する場合を除き、ほかのホストに移動することはできません。**vendor** フィールドに **TEMPORARY** と指定されている場合は、該当するライセンスを任意のライセンス サーバー ホストに移動できます。

ライセンス ファイルには、ライセンス セットを定義する **-license** 行を複数記述することができます。すべての行が連結されて 1 つのライセンスを構成します。この行数によって、ライセンス スロットの総数が決まります。

## ユーザーの優先順位を指定する行

ライセンス ファイルには、1 人または複数のユーザーを (名前または数値 ID で) 指定する **-user** 行を複数記述することができます。すべての行が連結され、ライセンス取得の優先順位リストが 1 つ生成されます。このリストの 1 番上のユーザーが最高の優先順位を持ち、それ以降のユーザーにはそれぞれ 1 つずつ低い順位が割り当てられていきます。このリストに記述されていないユーザーも製品を使用することはできますが、ライセンスの取得順位は全員に共通して最低の順位が割り当てられます。

## ライセンス取得の対象外ユーザーを指定する行

ライセンス ファイルには、1 人または複数のユーザーを (名前または数値 ID で) 指定する **-nuser** 行を複数記述することができます。この行に指定されたユーザーはライセンスを取得できないため、製品も使用できません。

**-user** 行と **-nuser** 行の両方に同じユーザーを指定しても問題はありません。同じユーザーが両方の行に指定されている場合は、最初に指定されているエントリの方が採用されます。

## 監視を有効化する行

**-audit** とだけ記述された行は、ライセンス アクティビティの監視を有効化する行です。UNIX の場合、ライセンスの監視メッセージは、`/var/adm/rational/clearcase/log/albd_log` ファイルに書き込まれます。Windows の場合、これらのメッセージは Windows イベント ログに書き込まれます。どちらのプラットフォームでも、次のライセンス イベントがログに記録されます。

- ユーザーが新しいライセンスを取得した
- すべてのライセンスが使用中のため、ユーザーがライセンスの取得を拒否された
- ユーザーが **clearlicense -release** コマンドを使用した (コマンドが正常終了したかエラーが発生したかも記録されます)

## タイムアウトを指定する行

デフォルトの設定では、ユーザーが取得した ClearCase ライセンスは、使用されない状態が 60 分間続くと (つまり、60 分の間、ユーザーが **cleartool** コマンドを実行したり、ClearCase の GUI を使用しなかった場合)、無効になります。このデフォルトの期間は、**-timeout** 行に数値 (分単位) を指定して変更できます。下限値は 30 分です。上限はありません。

## ClearCase LT の Rational Common Licensing

---

Rational ClearCase LT では、Rational Common Licensing が採用されています。このライセンス方式は、GLOBEtrötter Software, Inc. が提供する FLEXlm ライセンス管理ツールに基づいています。ClearCase LT のインストール ショー プロシージャでは、ClearCase LT クライアント ホストに FLEXlm クライアント ソフトウェアをインストールします。

ほとんどのユーザーは、ClearCase LT に付属しているソフトウェアを使用して各自のコンピュータのライセンス機能を設定できます。ライセンス サーバーを設置する必要がある場合は、通常、管理者が Rational 社と GLOBEtrötter 社が提供するソフトウェアを使用し、ホストを Rational ライセンス サーバーとして設定します。その後、このホストがライセンスをほかのコンピュータに提供し、管理します。自社のサイトに Rational ライセンス サーバーがあり、特にこのサーバーがほかの Rational 製品のライセンスを提供するように設定されている場合、そのサーバーを ClearCase LT のライセンス サーバー ホストとしても使用できます。

『Rational Suite ライセンス管理ガイド』に、Windows プラットフォームと UNIX プラットフォームでの Rational Common Licensing のすべての機能について詳しく説明されています。ユーザーや管理者が ClearCase LT のライセンスをインストール、設定、トラブルシューティングするために必要な情報や、Rational ライセンス サーバー ホストの設定手順も記載されています。

### フローティング ライセンスとノード ロック ライセンス

Rational Common Licensing では、フローティング ライセンスとノード ロック ライセンスの 2 種類のライセンスをサポートします。

- フローティング ライセンスは、ライセンス サーバー ホストにインストールされ、指定した数のユーザーに対して、そのサーバーがサービスを提供する任意の ClearCase LT クライアント ホスト上で ClearCase LT を使用する権利を与えます。
- ノード ロック ライセンスは、単一の ClearCase LT クライアント ホストにインストールされ、そのホストにログインする個々のユーザーに対して、そのホスト上で ClearCase LT を使用する権利を与えます。

**メモ:** UNIX プラットフォーム上の ClearCase LT では、フローティング ライセンス方式のみを使用できます。

ユーザーが **cleartool** コマンドを実行したり **ClearCase LT** の GUI を使用したりすると、そのコマンドや GUI 操作を実行するために、ライセンスの取得が試行されます。このライセンスは、有効なノード ロック ライセンスがインストールされているクライアント上では、必ず取得できます。フローティング ライセンスを使用するように設定されているクライアントでライセンス取得を試行すると、そのクライアントからネットワーク経由でライセンス サーバー ホストにアクセス可能で、かつライセンスを取得可能な場合にのみ、ライセンスを取得できます。ユーザーがフローティング ライセンスを取得した後、任意の **ClearCase** コマンドを実行したり、GUI 操作を行うと、ライセンスの有効期限が延長されます。ユーザーが一定の期間（デフォルトでは 30 分）内に **ClearCase** コマンドを実行しないと、ライセンスの有効期限が切れ、別のユーザーがそのライセンスを使用できるようになります。

## テンポラリー ライセンス、パーマネント ライセンス、期限付き使用承諾ライセンス

フローティング ライセンスとノード ロック ライセンスはどちらも、テンポラリー ライセンスまたはパーマネント ライセンスであり、期限付き使用承諾 (TLA) の適用対象となります。テンポラリー ライセンスは、**ClearCase LT** に同梱されており、このライセンスを使用して、パーマネント ライセンス キーの要求処理中でも製品をすぐに使用できます。TLA ライセンス キーは、パーマネント ライセンス キーの一種です。TLA ライセンスの詳細については、Rational の販売担当者にお問い合わせください。

## ClearCase LT と Rational Suite のライセンス

Windows プラットフォームでは、**ClearCase LT** は **Rational Suite** のフローティング ライセンス、または **ClearCase LT** 独自のフローティング ライセンスを使用できます。また、**Rational Suite** のノード ロック ライセンスも使用できます。

## ライセンス サーバーのインストールと設定

---

本製品以外の Rational 製品を使用している場合、既にサイト内に Rational ライセンス サーバーが存在する可能性があります。その場合は、**ClearCase LT** ライセンスをそのホストにインストールすることができます。**ClearCase LT** 専用のライセンス サーバーを設定する必要はありません。

**メモ:** **ClearCase LT** のすべてのクライアントが Windows コンピュータ上で稼動しており、ノード ロック ライセンスを使用している場合、ライセンス サーバーをインストールまたは設定する必要はありません。

Windows コンピュータまたは UNIX コンピュータのどちらかにホストされた Rational ライセンス サーバーは、プラットフォームに関係なく、**ClearCase LT** すべてのクライアントに対してライセンスを提供できます。ライセンス サーバー ホストを設定する必要がある場合は、『**Rational Suite** ライセンス管理ガイド』を参照してください。

## Windows クライアント ライセンスのインストールと設定

---

ライセンス サーバー ホストの設定後、そのホストに ClearCase LT ライセンスをインストールしないと、ユーザーが ClearCase LT を実行することができません。ユーザーや管理者が ClearCase LT クライアント ホストにライセンスをインストールして設定するときに、いくつかのオプションを指定できます。

### License Key Administrator

Windows 版の ClearCase LT には、Rational License Key Administrator プログラムが付属されています。このプログラムは、フローティング方式またはノード ロック方式のライセンスのどちらかを使用する ClearCase LT クライアント ホストの構成を管理し、Rational 社からノード ロック ライセンスを取得するプロシージャを自動化します。詳細については、License Key Administrator のヘルプと『Rational Suite ライセンス管理ガイド』を参照してください。

## UNIX クライアント ライセンスのインストールと設定

---

UNIX プラットフォームでは、ClearCase LT はフローティング ライセンス方式のみをサポートします。フローティング ライセンス方式では、ライセンス サーバー ホストを設定する必要があります。また、管理者がそのホストにフローティング ライセンスをインストールしておかなければ、ユーザーがライセンスを使用することはできません。ClearCase LT のパーマネント ライセンスを取得してインストールする方法については、『Rational Suite ライセンス管理ガイド』を参照してください。UNIX 上で動作する ClearCase LT ホストのライセンス サーバーを指定する方法については、『Rational Suite UNIX Installation Guide』を参照してください。



Rational ClearCase は分散型アプリケーションなので、ユーザーは、特定のホスト名やネットワーク パスなど、ネットワーク リソースへのアクセスに通常必要な情報を知らなくても、共有リソースにアクセスできます。ClearCase レジストリでは、以下の情報についての集中リポジトリを使用できます。

- VOB とビューの場所
- 一般的なクライアント ホスト設定のデフォルト値

各 ClearCase ホストは、1 つのレジストリ ホストに割り当てられます (そのレジストリ ホストのクライアントになります)。ClearCase LT コミュニティでは、レジストリ サーバー ホストは常に ClearCase LT サーバーです。ClearCase コミュニティのレジストリ サーバー ホストは、通常はサイトの準備プロセス中に指定されます。どのコンピュータでも、ClearCase がインストールされていればレジストリ サーバー ホストとして指定できます。クライアントは、リモート プロシージャ コール (RPC) を使用して、レジストリ サーバー と通信します。RPC は、必要に応じてレジストリの情報を要求し、VOB やビューが作成または移動されたときか、コミュニティのデフォルト値が変更されたときに、レジストリ データを変更します。

各 ClearCase コミュニティは、そのレジストリに依存しています。レジストリ サーバー ホストが使用不可能になったり、レジストリ データが破損したりすると、ユーザーは ClearCase で管理されている成果物にアクセスできなくなります。この章では、ClearCase レジストリと一般的なレジストリ管理手順について説明します。

**メモ:** ClearCase レジストリは、Windows レジストリとはまったく関係ありません。

## ClearCase レジストリと ClearCase LT レジストリの違い

---

ClearCase LT は 1 台のサーバーのみをサポートし、動的ビューをサポートしていないため、ClearCase LT レジストリの管理は非常に簡単です。ClearCase LT レジストリは、レジストリ リージョンをサポートしておらず、ClearCase 管理コンソールから直接管理することはできません。また、ClearCase レジストリ バックアップとスイッチオーバー機能も使用できません。ClearCase LT 管理者の中には、レジストリ データ モデルについて理解を深めるためにこの章を参照している方がいるかもしれませんが、ほとんどの内容は ClearCase LT に当てはまりません。

## レジストリ データ

---

レジストリ データ モデルは単純なモデルで、共有リソースの検索と取得に必要な基本情報を提供するように設計されています。レジストリには、次のような種類のデータが保存されています。

- **オブジェクトとタグ:** ほとんどの ClearCase アプリケーションでは、VOB またはビューのオブジェクトとタグの情報を使用して VOB またはビューに接続します。オブジェクトとタグは、VOB やビューが作成されるときにその一部として自動的に作成されます。また、ユーザーが必要に応じて作成、削除、変更することもできます。
- **ほかの共有データ:** レジストリはシステム設定情報のリポジトリでもあり、VOB とビューの推奨記憶場所や、各ホスト構成パラメータのサイト全域用推奨デフォルト値などの情報を提供します。これらのデフォルト値は管理者によって決定され、通常はインストール時に設定されますが、必要に応じて変更できます。

レジストリ データ モデルを理解することは、多くの管理手順において重要です。

### オブジェクト

レジストリには、各 VOB または各ビューのオブジェクト エントリが必ず 1 つ存在している必要があります。オブジェクトは、VOB やビューの作成時に作成され、VOB またはビューのサーバー プロセスで VOB またはビューのデータにアクセスするために必要な情報の常駐レコードを提供します。

VOB オブジェクトには、以下の情報が保存されています。

- VOB のサーバー プロセスが実行されるホストの名前。VOB のサーバー プロセスは、VOB 記憶ディレクトリが配置されているホストで実行されます。ただし、VOB 記憶ディレクトリが、認定されているネットワーク接続記憶 (NAS) デバイスに配置されている場合を除きます。
- VOB のユニバーサル ユニーク ID (以下 UUID)。各 VOB には、レプリカ UUID とファミリー UUID があります。UUID は、VOB の作成時に生成されて、VOB データベースに保存され、register コマンドによって VOB オブジェクト エントリにコピーされます (このコマンドは、VOB を作成する mkvob コマンドや GUI により暗黙的に実行されます)。VOB のレプリカは、Rational ClearCase MultiSite で作成されます。この場合、ファミリー内の各レプリカはそれぞれ固有のレプリカ UUID を持ち、同じファミリー内のすべてのレプリカは同じファミリー UUID を持ちます。VOB のレプリカが作成されない場合、ファミリー UUID は使用されません。
- VOB 記憶ディレクトリへのパス名。VOB ホスト自体で有効な形式で記述されます。このホスト上のローカル パス名は、VOB のサーバー プロセスで使用されます。
- VOB オブジェクト属性。以下の情報が含まれます。
  - UCM VOB (VOB は UCM コンポーネントです)
  - プロジェクト VOB (VOB は UCM PVOB です)

ビュー オブジェクトには、以下の情報が保存されています。

- ビューのサーバー プロセスが実行されるホストの名前。ビューのサーバー プロセスは、ビュー記憶ディレクトリが配置されているホストで実行されます。ただし、ビュー記憶ディレクトリが、認定されている NAS デバイスに配置されている場合を除きます。
- ビューの UUID。VOB の UUID と同じように、ビューの UUID もビューの作成時に生成され、ビュー記憶ディレクトリに保存されビュー オブジェクトにコピーされます。
- ビュー記憶ディレクトリへのパス名。ビューのホスト自体で有効な形式で記述されます。このホスト上のローカル パス名は、ビューのサーバー プロセスで使用されます。
- ビューの所有者であるユーザーの名前。
- ビュー オブジェクト属性。以下の情報が含まれます。
  - スナップショット ビュー
  - UCM ビュー (sumview として表示される場合があります)
  - Web ビュー (常にスナップショット ビュー属性と組み合わせられます)

VOB オブジェクトまたはビュー オブジェクトは、VOB またはビューの作成時にレジストリに追加されます。VOB またはビューの形式が変更されると (**reformatvob** コマンドまたは **reformatview** コマンドを使用)、オブジェクト エントリが更新されます。形式が変更されることによって、VOB のレプリカ UUID (ファミリー UUID は対象外) またはビューの UUID が変更されるためです。

VOB オブジェクトまたはビュー オブジェクトは、手動で更新または削除できます。この場合は、**cleartool register** コマンドと **unregister** コマンドを使用します。ClearCase 管理コンソールを使用して行うこともできます。この処理は、通常、以下のいずれかの場合にのみ必要です。

- VOB またはビューを別のホストに移動する場合
- VOB またはビューのサーバー ホスト名を変更する場合
- VOB またはビューの削除操作が不完全か間違っていたため、レジストリをクリーンアップする場合

## タグ

多くの ClearCase アプリケーションから VOB またはビューにアクセスするには、レジストリに VOB またはビューのオブジェクトだけでなくタグも存在する必要があります。VOB タグとビュー タグは、同じような目的で機能し、VOB とビューへのアクセスを同じような方法で制御します。

## VOB タグと VOB アクセス

VOB データにアクセスするほとんどの操作は、VOB タグを基に VOB を参照します。VOB タグには、以下の情報が保存されています。

- VOB のサーバー プロセスが実行されるホストの名前。VOB のサーバー プロセスは、VOB 記憶ディレクトリが配置されているホストで実行されます。ただし、VOB 記憶ディレクトリが、認定されている NAS デバイスに配置されている場合を除きます。
- VOB のレプリカ UUID。VOB オブジェクトからコピーされます。
- VOB 記憶ディレクトリへのネットワーク パス名。VOB にアクセスするすべてのホストに有効な形式で記述されます。グローバル パスとして参照されるこのパス名は、VOB 記憶ディレクトリにネットワーク経由でアクセスするクライアント プログラムで使用されます。
- デフォルトのマウント オプション。これは、動的ビューと共に使用するために VOB をマウントするときに適用されます。詳細については、[mount リファレンス ページ](#)を参照してください。
- Public 属性 (VOB がパブリック VOB として作成された場合)。この属性の詳細については、[103 ページの「パブリック VOB とプライベート VOB」](#)を参照してください。
- 説明 (必要な場合)。各種コマンドと GUI によって表示されます。

動的ビューで VOB にアクセスするには、事前に VOB をマウントする必要があります。VOB をマウントするすべての操作は、VOB タグに基づいて VOB を参照します。タグのない VOB はマウントできません。

スナップショット ビューや Web ビューで VOB にアクセスする場合は、VOB をマウントする必要はありません。ただし、ビューのロード規則では VOB タグが参照されるので、タグのない VOB からスナップショット ビューや Web ビューにエレメントがロードされることはありません。

すべてのビューで、VOB タグはファイル システム ディレクトリとして表示されます。動的ビューでは、VOB タグはネットワーク ファイルと同様にマウントされます。スナップショット ビューと Web ビューでは、そのビューにロードされた最上位ディレクトリに名前を付けるときに VOB タグが使用されます。VOB タグの命名規則は、VOB にアクセスするプラットフォームによって異なります。

- Windows または UNIX を実行している ClearCase LT クライアントからアクセスする VOB のタグでは、スラッシュまたは円マークで始まる単一のリーフ (たとえば ¥sources) を使用します。
- Windows を実行している ClearCase クライアントからアクセスする VOB のタグでは、円マークで始まる単一のリーフ (たとえば ¥sources) を使用します。

- UNIX を実行している ClearCase クライアントからアクセスする VOB のタグでは、通常、マウント ポイントとリーフ名で構成された名前を使用します。たとえば、/vobs/sources という VOB タグの場合、/vobs がマウント ポイント名、/sources がリーフ名になります。各クライアント ホストでは、VOB をマウントする前に、マウント ポイントをローカルに作成する必要があります。

**メモ:** 単一のリーフのみの UNIX VOB タグを使用することも可能ですが、UNIX コンピュータのルート ディレクトリにマウントできるファイル システムの数に制限があるため、動的ビューを使用する多くのコミュニティでは実用的ではありません。単一のリーフの VOB タグを Windows と UNIX が混在する環境で使用した場合は、先頭の文字 (円マークまたはスラッシュ) のみが異なるタグは同じタグであると認識されます。

## ビュー タグとビュー アクセス

ビューにアクセスするほとんどの操作は、ビュー タグを基にビューを参照します。ビュー タグには、以下の情報が保存されています。

- ビューのサーバー プロセスが実行されるホストの名前。このプロセスは、ビュー記憶ディレクトリが配置されているホストで実行されます。ただし、ビュー記憶ディレクトリが、認定されている NAS デバイスに置かれている場合を除きます。
- ビューの UUID。ビュー オブジェクトからコピーされます。
- ビュー記憶ディレクトリへのネットワーク パス名。ビューを使用するすべてのホストに有効な形式で記述されます。このパス名は、ビュー記憶ディレクトリにネットワーク経由でアクセスするクライアント プログラムで使用されます。
- 説明 (必要な場合)。各種コマンドと GUI によって表示されます。

動的ビューを使用するには、事前に動的ビューを起動する必要があります。動的ビューを起動するすべての操作は、ビュー タグを基にビューを参照します。タグのないビューは起動できません。

スナップショット ビューまたは Web ビューを起動する必要はありません。ただし、ビューの内容を変更するすべての操作 (更新やチェックアウトなど) ではビュー データベースにアクセスする必要があり、ビューにタグがないとアクセスできません。

VOB タグと同様に、ユーザーやアプリケーションは、ビュー タグもファイル システム ディレクトリとして参照します。

- UNIX クライアントでは、各動的ビューは /view ディレクトリのサブディレクトリとして表示されます。
- Windows クライアントでは、各動的ビューが、特別なネットワーク名 (デフォルトでは \\view) で共有として、また、クライアントの MVFS ドライブ (デフォルトではドライブ M) のディレクトリとして表示されます。

- UNIX と Windows のどちらでも、スナップショット ビューはローカル ファイル システムのディレクトリとして表示されます。

ビュー タグの長さや文字セットは、ホスト プラットフォームのファイル命名規則に従っている必要があります。

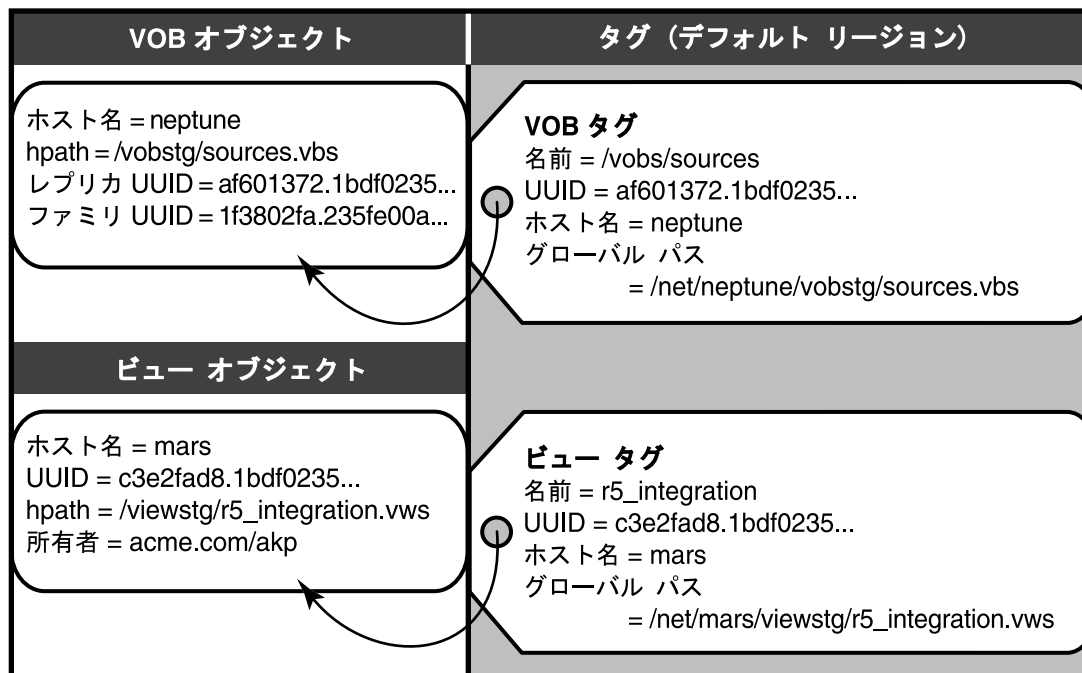
## レジストリ リージョン

レジストリ リージョンとは、レジストリ クライアントのサブセットが共有するタグ ネームスペースのことです。ClearCase LT レジストリでは、1 つのリージョンのみをサポートします。ClearCase レジストリでは、複数のリージョンを作成できます。ClearCase レジストリの VOB タグとビュー タグには、タグが表示されるレジストリ リージョンの名前が含まれます。

各 ClearCase クライアントは、1 つのレジストリ リージョンに属し、そのリージョンにタグが表示される VOB とビューにのみアクセスできます。

ClearCase でも ClearCase LT でも、レジストリには最初、1 つのリージョンだけがあります (図 1 を参照)。すべてのクライアントは、このリージョンに属しています。すべてのタグはこのリージョンで作成され、レジストリ ホストのすべてのクライアントで表示できます。

図 1 リージョンが 1 つのみのレジストリ

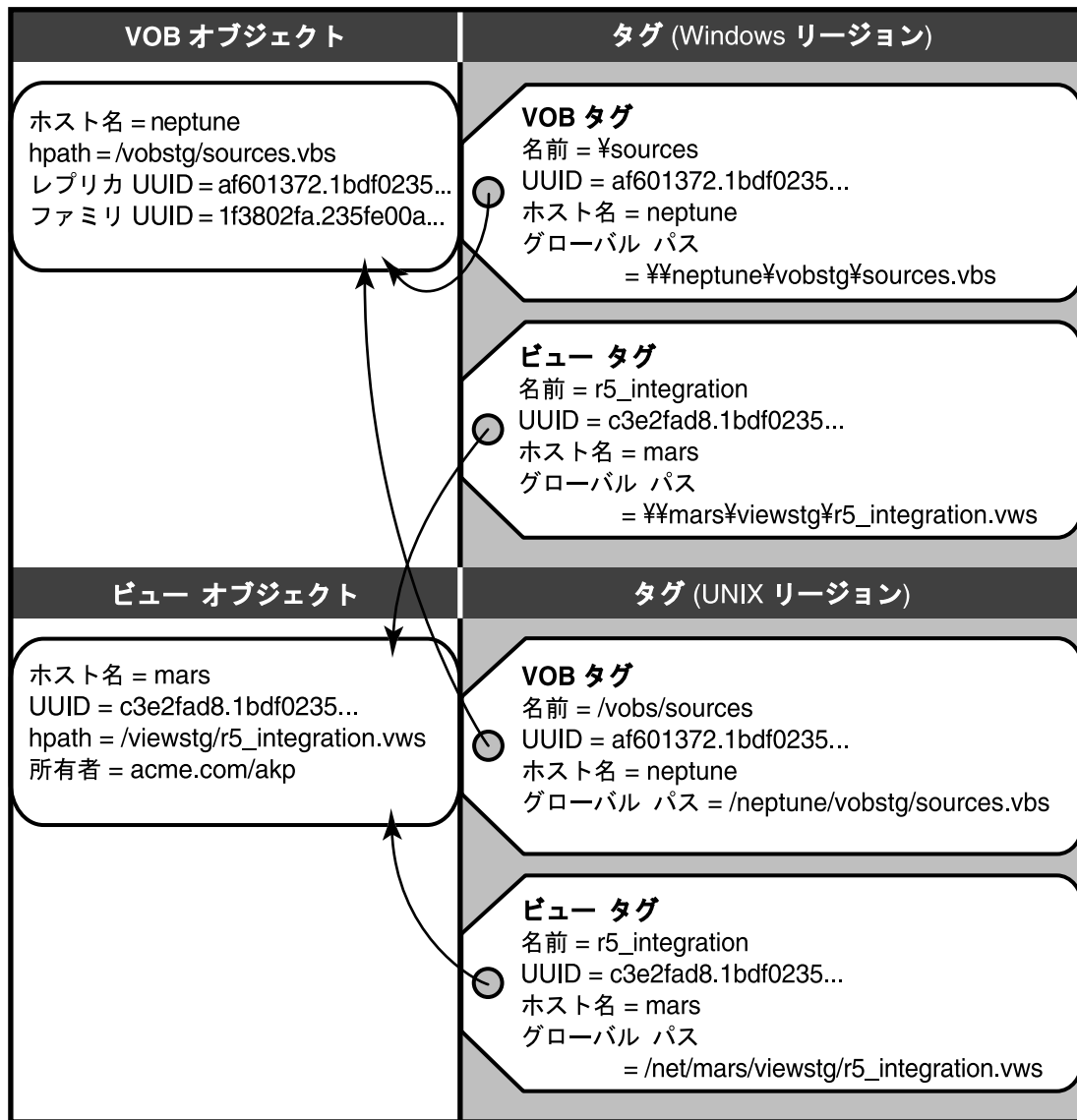


追加のリージョンを作成する最大の理由は、動的ビューを使用する Windows ホストから UNIX ホスト上の VOB やビューにアクセスするときに、Windows オペレーティング システムと UNIX オペレーティング システムで異なるネットワーク ファイル システム命名規則に対応するためです (スナップショット ビューと Web ビューでは、ネットワーク ファイル システムから VOB サーバー ホストにアクセスする必要はありません)。

図 2 に、リージョンが 2 つあるレジストリを示します。1 つは Windows クライアント用で、グローバル パスを UNC 名として表現します。もう 1 つは UNIX クライアント用で、グローバル パスを NFS パス名として表現します。

**メモ:** 複数のリージョンでタグを作成する場合は、汎用命名基準を設けることをお勧めします。たとえば、リーフ名 **sources** を、UNIX では **/vobs/sources** タグ、Windows では **¥sources** タグとして保存すると、この 2 つのタグが同じ VOB を参照することが簡単に分かります。

図 2 2つのリージョンがあるレジストリ



レジストリ内のリージョン数に関係なく、同じレジストリ内の VOB タグまたはビュー タグは、VOB またはビューの同じ基本オブジェクト エントリを参照します。このエントリは、タグの作成前から存在している必要があります。

次の場合も、追加のリージョンを作成する必要があります。

- ネットワーク上に、複数のネットワーク インターフェイスがインストールされている VOB サーバー ホストまたはビュー サーバー ホストがあるため、ホスト名も複数ある場合。
- ユーザーが作成した VOB タグまたはビュー タグにマルチバイト文字が含まれており、レジストリに指定されているクライアントの設定がオペレーティング システムのロケールと異なっている場合。マルチバイト文字はホストのロケール設定 (UNIX の場合は `LOCALE` 環境変数によって、Windows の場合は [コントロール パネル] の [地域のオプション] で設定) に基づいて処理されるので、同じリージョン内のすべてのタグは同じ文字セットを使用し、同じリージョン内のすべてのホストは同じロケール設定を使用する必要があります。
- 特定のクライアントから特定の VOB またはビューにアクセスできないように、VOB またはビューのタグが存在しないリージョンにクライアントを割り当てる場合。

ClearCase ホストは、通常、インストール時にリージョンに割り当てられます。この割り当ては、必要に応じて後で変更できます。手順については、51 ページの「新しいレジストリ リージョンにホストを割り当てるには」を参照してください。ホストのリージョンを表示するには、ClearCase 管理コンソールのホスト ノードを使用するか、[コントロール パネル] の ClearCase プログラムを使用するか、`cleartool hostinfo -long` コマンドを使用します。

**メモ:** VOB のサーバー ホストが割り当てられているリージョンには、VOB のタグが存在している必要があります。VOB のタグは、ほかのリージョンに設定することもできます。

## ほかの共有データ

レジストリは、ほかの共有データの集中リポジトリでもあり、サーバーの記憶場所やサイト全域用のデフォルト値に関する情報が保存されています。

## サーバーの記憶場所

サーバーの記憶場所は、ClearCase サーバー上の共有ディレクトリであり、ユーザーが GUI から VOB またはビューを作成したときの推奨の記憶場所のリストに表示されます。また、`-stgloc` オプションと共に `mkvob` コマンドと `mkview` コマンドに使用することもできます。VOB またはビューの記憶場所の作成は、サーバーのセットアップ時に行うか、`cleartool mkstgloc` コマンドまたは ClearCase 管理コンソールを使用して行います。

サーバーの記憶場所にはグローバル パスを使用するので、1 つのリージョンにのみ有効です。各リージョンには、VOB またはビューの記憶場所をまったく指定しないことも、複数指定することもできます。

サーバーの記憶場所を適切なデバイス (専用の VOB サーバー ホストかビュー サーバー ホスト、または認定されている NAS デバイス) で使用すると、VOB やビューの作成、管理、メンテナンスが簡単になります。

サーバーの記憶場所の詳細については、99 ページの「VOB のサーバー記憶場所の作成」と `mkstgloc` リファレンス ページを参照してください。

## サイトのデフォルト値

ClearCase レジストリには、多数の ClearCase ホスト プロパティのデフォルト値を保存できます。これらの値は、サイトの準備プロセス時に設定されます。いくつかの値は、**cleartool setsite** コマンドまたは ClearCase 管理コンソールを使用して変更できます (50 ページの「サイトのデフォルト値の変更」を参照)。サイトのデフォルト値には、以下のものがあります。

- 新しく作成された動的ビューで使用するキャッシュ サイズ
- 新しく作成されたビューで使用するデフォルトのテキスト モード
- 特定のコマンドのデフォルト動作 (たとえば、**checkout** コマンドで **-reserved** オプションを使用するかどうかなど)

完全なリストについては、**setsite** リファレンス ページを参照してください。

ほとんどのサイトのデフォルト値は、クライアントごとに上書きできます。その場合、コマンド行ツールを使用するか、[コントロール パネル] の ClearCase プログラムなどの GUI を使用します。

**メモ:** サイトのデフォルト値は、クライアントが属しているリージョンにかかわらず、レジストリ ホストのすべてのクライアントに適用されます。

## レジストリ サーバー ホスト

---

どの ClearCase ホストもレジストリ サーバーに指定できます。UNIX ホストと Windows ホストが混在したコミュニティでは、レジストリ サーバーが UNIX コンピュータでも Windows コンピュータでもかまいません。クライアントは RPC を使用してレジストリ サーバーと通信するので、プラットフォームが異なっても問題がないためです。

**メモ:** ClearCase LT コミュニティでは、レジストリ サーバーは常に ClearCase LT サーバーです。

レジストリ要求はコンパクトであり、レジストリ ルックアップが効率よく実行されるので、レジストリ サーバー プロセス (実際は **albd\_server** で処理) で計算リソースが大量に消費されることはありません。レジストリ データベースは、通常はあまり大きくならず、2 ~ 4 MB の記憶容量で数千の VOB とビューのレジストリ データを保存できます。レジストリはほとんどの ClearCase 操作に不可欠なので、レジストリ サーバー ホストはネットワーク上のすべてのホストに対して、安定し、使用可能で、アクセス可能になっている必要があります。バックアップ レジストリ ホストを設定して (54 ページの「ClearCase レジストリのバックアップとスイッチオーバー」を参照)、プライマリ レジストリ ホストに障害が発生した場合にレジストリ サーバーの作業を引き継げるようにすることをお勧めします。ただし、ClearCase LT ではバックアップ レジストリ ホストをサポートしていません。また、レジストリ データのバックアップを、通常のバックアップ作業の一環として実行することをお勧めします。206 ページの「ClearCase のレジストリ データのバックアップ」を参照してください。

## ClearCase レジストリ ホストの定義

**メモ:** ここでの説明は、ClearCase LT には当てはまりません。ClearCase LT コミュニティでは、レジストリ サーバーは常に ClearCase LT サーバーです。

ClearCase コミュニティの定義の一部は共通レジストリの使用法に基づいているので、通常、コミュニティ内では最初にレジストリ サーバーに ClearCase をインストールします。インストール処理の大半は ClearCase インストール プログラムによって自動的に実行されます。レジストリ サーバー ホストに最初にインストールすると、ClearCase コミュニティ内のほかのホストでのインストールと設定が簡単になります。ClearCase を最新版にアップグレードするときは、レジストリ サーバー ホストを最初にアップグレードすることをお勧めします。

ClearCase レジストリ サーバーをローカルの VOB やビューをサポートするように設定する必要はありませんが、多くのコミュニティでは VOB ホストやビュー ホストをレジストリ サーバーに指定しても全体のパフォーマンスが低下することはありません。

既存の ClearCase ホストを再インストールしないでレジストリ サーバーに変換するには、この項で説明する手順のいずれかを使用してください。

**メモ:** 新しいレジストリ ホストを定義したら、そのレジストリ用のパスワードを設定する必要があります。パスワードを設定しないと、パブリック VOB タグの作成やサイト全域用のデフォルト値の変更など、権限が必要な操作を実行できません。詳細については、44 ページの「レジストリ パスワードの設定と使用」を参照してください。

### UNIX コンピュータをレジストリ ホストとして定義するには

UNIX コンピュータをレジストリ ホストとして定義するには、以下の手順に従います。

- 1 ホストにログオンします。root としてログオンしてください。これにより、ClearCase の停止や起動の権限、または、ほとんどのホストの /var パーティションにディレクトリとファイルを作成する権限が得られます。
- 2 ClearCase を停止します。
- 3 /var/adm/rational/clearcase/rgy ディレクトリを作成します。
- 4 /var/adm/rational/clearcase/rgy/rgy\_svr.conf ファイルを作成します。このファイルには、文字列 master で構成される 1 行が含まれている必要があります。
- 5 ClearCase を再起動します。

### Windows コンピュータをレジストリ ホストとして定義するには

Windows コンピュータをバックアップ レジストリ ホストとして定義するには、以下の手順に従います。

- 1 管理者グループのメンバーとしてログオンします。
- 2 ClearCase を停止します。

- 3 [コントロール パネル] から ClearCase プログラムを実行します。
- 4 [レジストリ] タブをクリックし、[ローカル ホストをレジストリ サーバーとして使用] をオンにします。
- 5 ClearCase を再起動します。

## レジストリ パスワードの設定と使用

レジストリの変更は ClearCase コミュニティに大きな影響を与えるので、レジストリ操作の一部はパスワードで保護されています。レジストリ パスワードの設定や変更には、`rgy_passwd` コマンドを使用します。デフォルトのレジストリ パスワードはありません。新しく作成したレジストリでは、レジストリ パスワードを設定するまで、保護されている操作を実行しても失敗します。レジストリ パスワードを設定したら、パブリック VOB タグの作成操作やサイト デフォルト値の変更操作の際に入力します。権限を持つ ClearCase ユーザーでも、保護されている操作を実行する際にはこのパスワードを入力する必要があります。

## レジストリ クライアント リスト

各レジストリ サーバーには、過去 30 日間にこのサーバーに接続したクライアントのリストが保存されています。このクライアント リストを表示するには、`cleartool lsclients` コマンドや ClearCase 管理コンソールの GUI などを使用します。また 54 ページの「ClearCase レジストリのバックアップとスイッチオーバー」で説明されているように、`rgy_switchover` コマンドで使用することもあります。このリストに対してクライアントを手動で追加または削除することはできません。

## 複数の ClearCase レジストリの使用に関するガイドライン

---

1 つのレジストリ サーバー ホストが提供できるレジストリは 1 つのみです。レジストリ ホストを 1 つだけサポートする ClearCase LT 以外では、サイトに複数のコミュニティがあり VOB やビューを共有する必要がない (または共有してはならない) 場合、複数のレジストリを設定できます。各レジストリはそれぞれの専用ホストから提供されます。しかし、一般に簡単で安全なのは、レジストリ リージョン メカニズムを使用して、クライアント グループがアクセスできる一連の VOB やビューを制限する方法です。レジストリ サーバーでは、多くのクライアントが接続して複数の VOB やビューの情報にアクセスしている場合でも、計算リソースが大量に消費されたり、パフォーマンス障害が発生したりすることはありません。複数のレジストリの使用では、パフォーマンス上またはスケーラビリティ上の利点はあまり得られず、それよりも管理の手間のほうが上回ることが多くあります。

ClearCase にはレジストリ間のデータ共有を自動化するツールがないので、1 つのサイトに複数のレジストリを実装した場合は、各レジストリを個別に管理する必要があります。たとえば、異なるレジストリ サーバーを使用するホスト上で作成されたすべての VOB とビューのレジストリにオブジェクト エントリとタグ エントリを手動で作成する必要があります。また、レジストリごとにサイト デフォルト値を個別に設定して管理する必要があります。

## ClearCase ホストのレジストリとリージョンの割り当て

---

**メモ:** ここでの説明は、ClearCase LT には当てはまりません。ClearCase LT でサポートするレジストリ リージョンは 1 つだけです。

各 ClearCase ホストには、通常はインストール時に、レジストリ サーバーの名前と、ホストがタグをルックアップするレジストリのリージョンの名前が設定されます。これらの名前を両方とも、またはいずれかを変更するには、以下の手順に従います。

1 新しいレジストリ ホストとリージョンを指定します。

- UNIX ホストの場合は、`/var/adm/rational/clearcase/rgy/rgy_hosts.conf` ファイルを編集して、新しいレジストリ ホスト名と、必要に応じてバックアップ レジストリ ホスト名を指定します。次に、`/var/adm/rational/clearcase/rgy/rgy_region.conf` ファイルを編集して、新しいレジストリのリージョン名を指定します。これらのファイルを変更するには、`root` としてログオンしている必要があります。
- Windows ホストの場合は、[スタート] ボタンをクリックし、[設定] をポイントして、[コントロール パネル] をクリックします。[コントロール パネル] から、ClearCase プログラムを起動します。[レジストリ] タブで、[次のホストのレジストリ サーバーを使用] テキスト ボックスに新しいレジストリ サーバー ホスト名を入力し、[Windows リージョン] テキスト ボックスに適切なリージョン名を入力します。

2 ClearCase を停止します。

3 ClearCase を再起動します。

新しいリージョンの割り当てが有効になっているかどうかを確認するには、`cleartool hostinfo -long` コマンドを実行します。

**メモ:** ホストがリモート管理できるように設定されている場合は、ClearCase 管理コンソールのホスト ノードを使用してホストのレジストリ リージョンを変更することもできます。ただし、ClearCase 管理コンソールからホスト上の ClearCase を停止したり起動したりすることはできません。

## レジストリ データの管理

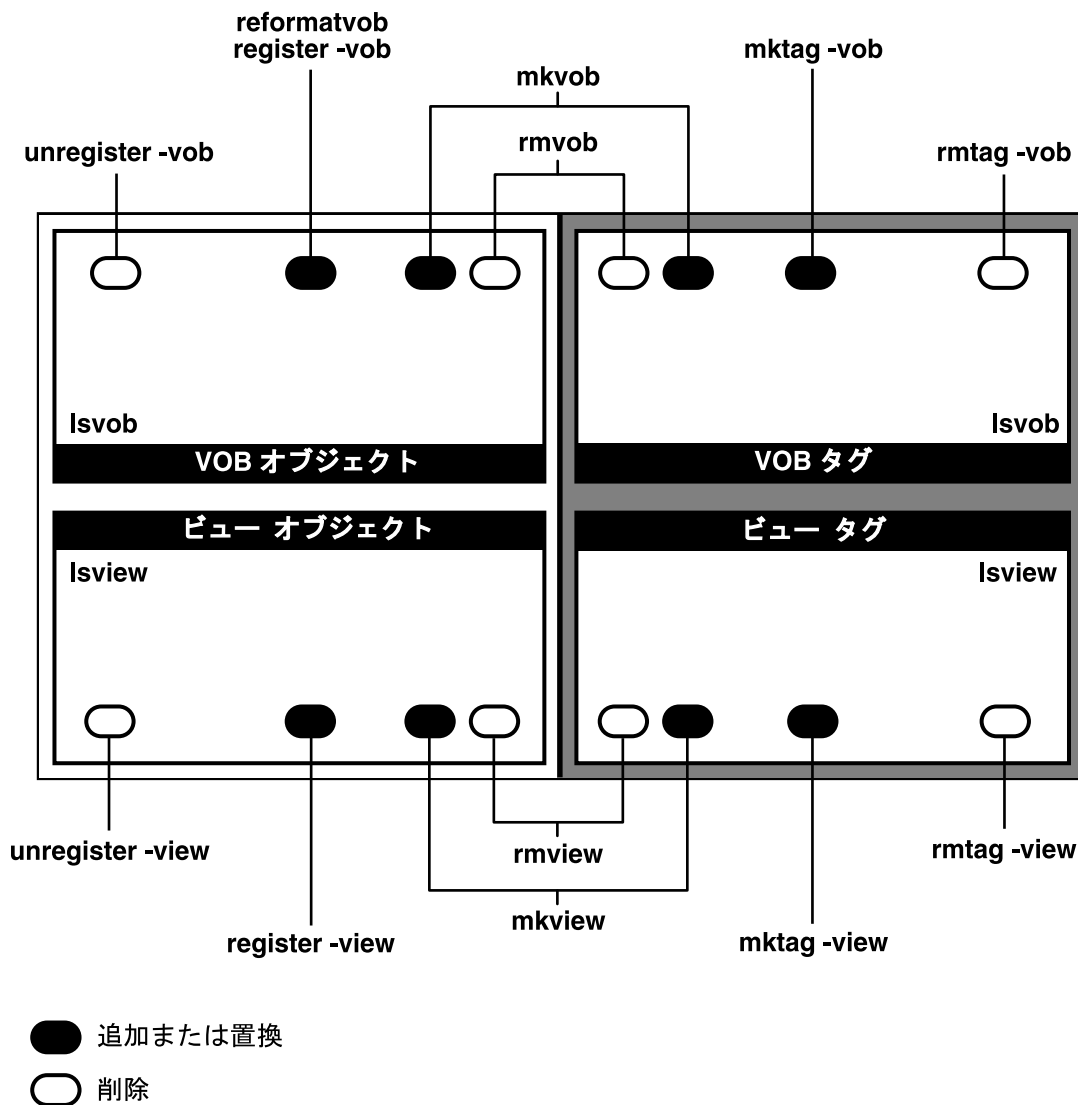
---

ClearCase では、VOB やビューの作成時にオブジェクトとタグが作成されます。ユーザーがビューを削除すると、オブジェクトとタグも削除されます。ほとんどの場合、これ以上の管理操作は必要ありません。VOB やビューのオブジェクトとタグの操作が必要になるのは、以下の場合です。

- タグを別のレジストリ リージョンで使用可能にする場合
- VOB またはビューの名前を変更する場合
- VOB またはビューを別のホストに移動した場合
- VOB またはビューをバックアップ メディアから復元した場合
- 標準的でない手順で VOB またはビューを削除した後にレジストリをクリーン アップする場合

VOB やビューのオブジェクトとタグの検査、作成、修復には、ClearCase 管理コンソールの [ClearCase レジストリ] ノードを使用するか、図 3 の **cleartool** サブコマンドのいずれかを使用します。この図は、このコマンドがレジストリ データにどのように作用するかを示しています。

図 3 cleartool コマンドによるレジストリ データへの作用



## オブジェクトとタグの表示

ClearCase 管理コンソールで [ClearCase レジストリ] ノードの [VOB オブジェクト] サブノードと [ビュー オブジェクト] サブノードを次のように使用して、VOB オブジェクトとビュー オブジェクトを検査します。

- 1 ClearCase 管理コンソールを起動します。
- 2 [ClearCase レジストリ] ノードの [VOB オブジェクト] サブノードまたは [ビュー オブジェクト] サブノードに移動します。
- 3 [詳細] ペインで VOB オブジェクトまたはビュー オブジェクトを選択します。
- 4 [操作] メニューの [プロパティ] をクリックします。

ClearCase 管理コンソールで [ClearCase レジストリ] ノードの [VOB タグ] サブノードと [ビュー タグ] サブノードを次のように使用して、VOB タグとビュー タグを検査します。

- 1 ClearCase 管理コンソールを起動します。
- 2 [ClearCase レジストリ] ノードの [VOB タグ] サブノードまたは [ビュー タグ] サブノードに移動します。
- 3 [詳細] ペインで VOB タグまたはビュー タグを選択します。
- 4 [操作] メニューの [プロパティ] をクリックします。

また、`lsvob` コマンドまたは `lsview` コマンドと `-long` オプションを使用して、VOB やビューに関する以下のレジストリ情報を取得できます。

- タグ
- サーバー ホスト名
- ホスト上のローカル記憶アクセス パス名
- グローバル パス名

例を次に示します。

```
cleartool lsvob -long /vobs/sources
```

```
.  
.
タグ: /vobs/sources
グローバル パス: /net/neptune/vobstg/sources.vbs
サーバー ホスト: neptune
.
.
VOB サーバー アクセス パス: /vobstg/sources.vbs
.
.
```

```
cleartool lsview -long v5_integration
```

```
.  
.   
タグ: v5_integration  
  グローバル パス: /net/mars/viewstg/v5_integration.vws  
  サーバー ホスト: mars  
.   
.   
ビュー サーバー アクセス パス: /viewstg/v5_integration.vws  
.   
.
```

VOB またはビューが **lsvob** や **lsview** の出力に表示されない場合は、VOB またはビューのタグがホストのリージョンに存在していません (**lsview -region** オプションを使用した場合は、指定したリージョンにタグが存在していません)。

## VOB またはビューの登録

VOB オブジェクトまたはビュー オブジェクトの作成や変更には、ClearCase 管理コンソールか **cleartool register** コマンドを使用します。オブジェクトを作成または変更するのは、主に以下の場合です。

- VOB またはビューを新しいホストに移動する場合
- VOB またはビューをホスト上の新しい場所に移動する場合
- VOB またはビューのサーバー ホスト名を変更する場合
- そのほか、VOB またはビューのホスト上のローカル パス名を変更する操作を行う場合

オブジェクトの情報を変更するには、ClearCase 管理コンソールを使用して、オブジェクトを削除し、新しいオブジェクトを作成します。または、**cleartool unregister** コマンドと **register** コマンドを使用します。

## タグの作成、削除、変更

既存の VOB やビューのタグを追加、削除、変更するには、ClearCase 管理コンソールまたは **cleartool mktag** コマンドを使用します。タグを追加または置換するのは、主に以下の場合です。

- ほかのリージョンに追加のタグを作成する場合
- VOB またはビューの名前を変更する場合 (VOB またはビューの名前がタグと同義の場合)
- プライベート VOB をパブリック VOB に変換する場合、またはパブリック VOB をプライベート VOB に変換する場合
- 「VOB またはビューの登録」に記載されているいずれかの場合

タグの情報を変更するには、ClearCase 管理コンソールを使用して、タグを削除し、新しいタグを作成します。または、**cleartool rmtag** コマンドと **mktag** コマンドを使用します。

タグを別のリージョンにコピーまたは移動する手順については、51 ページの「新しいレジストリ リージョンにタグを作成するには」を参照してください。

## サイトのデフォルト値の変更

次のように ClearCase 管理コンソールを使用して、サイト全域のプロパティを表示または編集します。

- 1 ClearCase 管理コンソールを起動します。
- 2 [ClearCase レジストリ] ノードに移動します。
- 3 [操作] メニューの [プロパティ] をクリックして、サイトのデフォルト値を表示または変更します。

また、`cleartool lssite` コマンドを使用して、一連のサイト全体用のプロパティとその値を表示することもできます。サイト全体用のプロパティに新しい値を設定するには `cleartool setsite` コマンドを使用します。使用可能なプロパティとその値の詳細については、`setsite` リファレンスページを参照してください。

**メモ:** サイトのデフォルト値には、サイト準備中にしか設定できないものがあります。このような値は、ClearCase 管理コンソールや `setsite` コマンドを使用して変更することはできません。また、サイトのデフォルト値を変更するには、レジストリ パスワードを入力する必要があります。

## ClearCase レジストリ リージョンの追加

**メモ:** ここでの説明は、ClearCase LT には当てはまりません。ClearCase LT でサポートするレジストリ リージョンは 1 つだけです。

既存の ClearCase レジストリに新しいリージョンを追加するには、以下の手順に従います。

- リージョンを作成します。
- リージョンにホストを割り当てます。
- リージョンにタグを作成します。

### 新しいレジストリ リージョンを作成するには

ClearCase 管理コンソールを使用して、新しいリージョンを作成します。この例でのリージョン名は `dev_win` です。

- 1 ClearCase 管理コンソールを起動します。
- 2 [ClearCase レジストリ] ノードの [リージョン] サブノードに移動します。
- 3 [操作] メニューの [新規作成] をポイントし、[リージョン タグ] をクリックします。このコマンドで、新しいリージョンを作成するダイアログ ボックスが開きます。

または、`mkregion` コマンドを次のように使用して作成することもできます。

```
cleartool mkregion -tag dev_win -tcomment "Windows ClearCase hosts"
```

## 新しいレジストリ リージョンにホストを割り当てるには

ホストのデフォルト レジストリ サーバーとリージョンの割り当ては、通常は ClearCase のインストール時に行います。ホストを別のリージョンに割り当てる手順については、45 ページの「ClearCase ホストのレジストリとリージョンの割り当て」を参照してください。

## 新しいレジストリ リージョンにタグを作成するには

新しく作成したレジストリ リージョンにタグを作成する方法は、次のようにいくつかあります。

- Windows GUI でリージョン シンクロナイザを使用します。この方法では、VOB タグとビュー タグを UNIX リージョンから Windows リージョンに簡単にインポートできます。リージョン シンクロナイザを起動するには、[スタート] ボタンをクリックし、[プログラム] をポイントします。次に、[Rational Software] の [ClearCase] をポイントして、[管理] の [リージョン シンクロナイザ] をクリックします。リージョン シンクロナイザの使用法に関するヘルプを表示するには、リージョン シンクロナイザの起動時に開く [ClearCase リージョン シンクロナイザ] ダイアログボックスで [ヘルプ] をクリックします。UNIX 環境の ClearCase ホストには、リージョン シンクロナイザ アプリケーションが用意されていません。
- ClearCase 管理コンソールを使用します。この方法では、一般的なインターフェイスを使用してリージョンとタグを操作できます。レジストリ ホストが UNIX または Windows のどちらの環境であっても、タグを操作できます。このコンソールでは、タグの作成、削除、別のリージョンへのコピーを行えます。ClearCase 管理コンソールでタグを操作する場合は、タグのグローバル パスを変更して、新しいリージョンのホストに対してそのタグを有効化する必要があります。リージョン シンクロナイザでは、UNIX リージョンから Windows リージョンにタグをインポートするという特定の場、パスが自動的に変更されます。
- cleartool mktag コマンドを使用します。

## リージョンが存在する ClearCase レジストリでの VOB とビューの作成

VOB またはビューを作成する GUI や cleartool コマンドでは、VOB またはビューのオブジェクトとタグも作成できます。タグは、ホストのリージョン (コマンドを実行したホストが割り当てられているリージョン) に作成されます。VOB またはビューをほかのリージョンのホストでできるようにするには、追加のタグを手動で作成する必要があります。これらのタグは VOB またはビューと共に作成されたオブジェクトを参照するので、新しいオブジェクトを作成する必要はありません。追加のタグを作成するには、ClearCase 管理コンソールまたは cleartool mktag コマンドを使用します。例を次に示します。

dev\_unix リージョンの saturn で VOB の /vobs/lib を作成してから、以下のコマンドを実行して、Windows ホストで使われる dev\_win リージョンに VOB タグを作成します。

```
cleartool mktag -vob -tag ¥lib -region dev_win -host saturn ^  
-hpath /vobstg/lib.vbs -gpath ¥¥saturn¥vobstg¥lib.vbs ¥¥saturn¥vobstg¥lib.vbs
```

この例の `mktag` コマンドは、UNIX ホストでも Windows ホストでも実行できます。ただし、グローバルパスの `¥¥saturu¥vobstg¥lib.vbs` を検証するには、このコマンドを Windows ホストで実行する必要があります。

## レジストリ リージョンの削除

レジストリ リージョンを削除するには、まず、51 ページの「新しいレジストリ リージョンにホストを割り当てるには」の説明に従ってリージョン内の各ホストを別のリージョンに再度割り当てます。次に、ClearCase 管理コンソールを使用してリージョンを削除します。

- 1 ClearCase 管理コンソールを起動します。
- 2 [ClearCase レジストリ] ノードで、削除するリージョンのサブノードに移動します。
- 3 [操作] メニューの [すべてのタスク] をポイントし、[リージョン タグの削除] をクリックします。

また、`rmregion` コマンドを以下のように使用して、リージョンを削除することもできます。

```
cleartool rmregion -tag region-name -rmall
```

## レジストリ サーバー ホスト名の変更

ClearCase クライアント ホストでは、レジストリ サーバー ホストの名前と IP アドレスがキャッシュされています。レジストリ サーバー ホスト名を変更した場合は、各クライアントホストのレジストリ サーバー ホストの割り当てを変更し、各ホストの ClearCase を再起動して、キャッシュされているレジストリ サーバー ホスト名を新しい名前に置き換えます。ネットワークのレジストリ サーバー ホスト名を変更するには、以下の手順に従います。

- 1 各クライアントホストの ClearCase を停止します。
- 2 レジストリ サーバー ホストの名前を変更します。ホストのオペレーティングシステムのツールを使用します。
- 3 43 ページの「ClearCase レジストリ ホストの定義」の説明に従って各クライアントホストのレジストリ サーバー ホストの割り当てを変更し、レジストリ サーバーの新しいホスト名を指定します。
- 4 すべてのホストの ClearCase を再起動します。

## レジストリ管理のガイドライン

---

レジストリ管理についてのガイドラインを以下に示します。

- ClearCase 管理コンソールが Windows で実行されていて、Windows 上または UNIX 上のレジストリを管理できる場合は、ClearCase レジストリ データがグラフィカルに表示されます。これにより、レジストリ データの構造と内容を簡単に理解できるようになります。これは、`cleartool` コマンドを使用してこのデータを操作する場合も有効です。

- VOB またはビューにアクセスするには (削除する場合でも)、VOB またはビューのオブジェクトとタグが存在している必要があります。存在を確認するには、ClearCase 管理コンソール、lsvob コマンド、lsview コマンドのいずれかを使用します。タグが存在しない場合は、何も表示されません。オブジェクト エントリが存在しない場合は、-long 出力に記憶領域へのパスが表示されません。

VOB またはビューにアクセスできないときは、タグの有無を確認します。VOB またはビューが lsvob コマンドまたは lsview コマンドの出力に表示されない場合は、タグが存在しません。この場合は、ClearCase 管理コンソールまたは mktag コマンドを使用してタグを作成します。

タグが存在する場合は、lsvob -long コマンドまたは lsview -long コマンドを使用してタグを表示します。出力に表示されたパス名が正しくない場合 (たとえば、ローカルとグローバルのパス名が同じ場合) は、オブジェクトやタグを ClearCase 管理コンソールを使用して修正します (該当するリージョンの [VOB オブジェクト] ノードまたは [ビュー オブジェクト] ノードと、[VOB タグ] ノードまたは [ビュー タグ] ノードを使用します)。

cleartool register コマンドと mktag コマンドを使用することもできます。

- 動的ビューと共に使用するには、VOB やビューの記憶ディレクトリをエクスポート (UNIX の場合) または共有 (Windows の場合) する必要があります。UNIX で VOB タグまたはビュー タグのグローバルパスを使用して動的ビューから VOB またはビューの記憶ディレクトリにアクセスできない場合は、ディレクトリがエクスポートされているかどうかを確認します。ディレクトリにほかのディレクトリへのシンボリック リンクが含まれているときは、そのリンクもエクスポートされているかどうかを確認します。Windows で VOB タグまたはビュー タグのグローバルパスを使用して動的ビューから VOB またはビューの記憶ディレクトリにアクセスできない場合は、ディレクトリが共有されているかどうかを確認します。共有されているときは、VOB にアクセスするすべてのグループに読み込みと書き込みのアクセス権が許可されているかどうかを確認します。
- VOB とビューの削除には、常に ClearCase ユーティリティを使用します。オペレーティングシステムのユーティリティ (rm や rmdir など) または GUI (Windows エクスプローラなど) は使用しないでください。これらを使用しても、レジストリ エントリはクリーンアップされません。
- VOB またはビューの記憶ディレクトリを削除する前に、タグを削除しないでください。rmvob コマンドまたは rmview コマンドを使用すると、タグも削除されます。
- 以下のタスクを実行する場合は、VOB やビューの記憶ディレクトリが配置されているホストでコマンドを実行します。特に、-host -hpath オプションと -gpath オプションを共に使用する場合は、これに従ってください。
  - VOB またはビューの作成
  - VOB タグまたはビュー タグの作成
  - 記憶ディレクトリの登録または登録解除
  - VOB の形式変更

これらのタスクを実行すると、コマンドまたは GUI を使用してホスト上のローカルパス名を検証することができます。リモート ホストからコマンドを実行した場合は、検証できません。

- 定期的に **rgy\_check** ユーティリティを実行して、問題を診断したり、不要になったか取り残されたレジストリ エントリをクリーン アップしたりします。

## ClearCase レジストリのバックアップとスイッチオーバー

---

**注意:** ここでの説明は、ClearCase LT には当てはまりません。

レジストリへのアクセスは、ほとんどすべての ClearCase 操作において重要です。レジストリ データはこのように重要なリソースなので、ClearCase ではどの ClearCase ホストもバックアップ レジストリ ホストとして指定できます。また、定期的に **rgy\_backup** コマンドを使用して、指定したホストにレジストリ データをコピーできます。プライマリ レジストリ ホストで障害が発生した場合は、**rgy\_switchover** プログラムを実行します。これにより、バックアップ レジストリ ホストがプロモートされてプライマリ レジストリ ホストになり、クライアント ホストがそのレジストリ ホストを使用するように再設定されます。

この項では、レジストリ ホストの障害が発生した場合の 2 つの例について説明します。一方では **rgy\_backup** を使用しており、もう一方では使用していなかった (ただし、206 ページの「ClearCase のレジストリ データのバックアップ」の説明に従って、プライマリ レジストリ ホストのファイル システムをバックアップしていた) と仮定します。

ClearCase スケジュール サービスにより、**rgy\_backup** コマンドがすべてのホストに対して定期的に実行されます。バックアップ レジストリ ホストとして設定されていない ClearCase ホストでは、このコマンドが実行されても何も影響はありません。バックアップ レジストリ ホストとして設定されている ClearCase ホストでは、このコマンドによって、プライマリ レジストリ ホストのレジストリ データベースとクライアント リストがバックアップ レジストリ ホストにコピーされます。バックアップ レジストリ ホストがプロモートされてプライマリ レジストリ ホストになった場合は、このコピーが使用されます。詳細については、**rgy\_backup** リファレンス ページと 229 ページの「ClearCase スケジューラ」を参照してください。

**メモ:** **rgy\_backup** コマンドを実行するには、バックアップ レジストリのデータベース ディレクトリに対する書き込み権限が必要です。

### バックアップ レジストリ ホストを定義するには

新しいバックアップ レジストリ ホストを定義するには、**rgy\_switchover** コマンドに **-backup** オプションを指定して実行します。また、次の手順のいずれかで、バックアップ レジストリ ホストを手動で定義することもできます。

**メモ:** バックアップ レジストリ ホストでは、プライマリ レジストリ ホストと同じリリースの ClearCase が実行されている必要があります。

## UNIX コンピュータをバックアップ レジストリ ホストとして定義するには

UNIX コンピュータをバックアップ レジストリ ホストとして定義するには、以下の手順に従います。

- 1 ホストにログオンします。**root** としてログオンしてください。これにより、**ClearCase** の停止や起動の権限、または、ほとんどのホストの **/var** パーティションにディレクトリとファイルを作成する権限が得られます。
- 2 **ClearCase** を停止します。
- 3 **/var/adm/rational/clearcase/rgy/rgy\_hosts.conf** ファイルを編集します。このファイルの 1 行目には、ホストのレジストリ サーバー名が記述されています。2 行目に、ホスト自体のホスト名を追加します。
- 4 **ClearCase** を再起動します。

ホストがバックアップ レジストリ ホストになり、**rgy\_backup** を実行して、**/var/adm/rational/clearcase/rgy/rgy\_hosts.conf** ファイルの 1 行目に記述されているプライマリ レジストリ ホストのレジストリ データのバックアップを開始します。

## Windows コンピュータをバックアップ レジストリ ホストとして定義するには

Windows コンピュータをバックアップ レジストリ ホストとして定義するには、以下の手順に従います。

- 1 ローカル管理者としてログオンします。
- 2 **ClearCase** を停止します。
- 3 [スタート] ボタンをクリックし、[設定] をポイントして、[コントロール パネル] をクリックします。[コントロール パネル] から、**ClearCase** プログラムを起動します。  
[レジストリ] タブをクリックし、[バックアップ レジストリ ホスト] フィールドに現在のホストの名前を入力します。
- 4 **ClearCase** を再起動します。

現在のホストがバックアップ レジストリ ホストになり、[レジストリ] タブの [次のホストのレジストリ サーバーを使用] で指定したプライマリ レジストリ ホストのレジストリ データを **rgy\_backup** を実行してバックアップを開始します。

## バックアップ レジストリ ホストをプロモートしてプライマリ レジストリ ホストにするには

この例では、`rgy_backup` プログラムがバックアップ レジストリ ホスト `rgy2` で正常に実行されていたと仮定します。プライマリ レジストリ ホスト `rgy1` に障害が発生したので、`rgy_switchover` を実行して `rgy2` をプライマリ レジストリ ホストにプロモートします。`rgy1` が使用できるようになったら、`rgy_backup` を実行した後に `rgy_switchover` を実行して、`rgy1` をプライマリ レジストリ ホストに戻します。

バックアップ レジストリ ホストをプロモートするには

- 1 `rgy_switchover` を実行します。`rgy2` をプライマリ レジストリ ホストにして、`rgy3` を新しいバックアップ レジストリ ホストにするため、以下のように指定します。

```
rgy_switchover -backup rgy3 rgy1 rgy2
```

- 2 アクセスできなかったクライアントを処理します。スイッチオーバーを正常に行えなかったクライアント ホストの名前を記録します。これらのホストは手動で再設定する必要がありますので、ホストが使用可能になったら、43 ページの「ClearCase レジストリ ホストの定義」の手順に従って再設定します。

**メモ:** ローカルの VOB やビューをサポートするようにインストールされていない ClearCase クライアント ホストは、`rgy_switchover` を実行しても再設定できないため、アクセスできなかったクライアントのリストに常に表示されます。

- 3 新しいバックアップ レジストリ ホストを設定します。プライマリ レジストリ ホストがダウンし、バックアップ レジストリ ホストがプライマリ レジストリ ホストとして機能するようになったら、別のバックアップ レジストリ ホストを設定して、`rgy_switchover` 機能を引き続き有効にしておくことをお勧めします。手順については、54 ページの「バックアップ レジストリ ホストを定義するには」を参照してください。

## プライマリ レジストリ ホストを復元するには

レジストリの内容は頻繁に変更されるので、バックアップ レジストリ ホストを使用していた後に、障害から復旧したレジストリ ホストをそのまま再起動することはできません。バックアップ レジストリ ホスト (この例では `rgy2`) に切り換えてあった場合に、`rgy1` をプライマリ レジストリ ホストとして復元するには、以下の手順に従います。

- 1 ホスト `rgy1` が再び使用できるようになったら、そのレジストリ サーバーを非アクティブにします。43 ページの「ClearCase レジストリ ホストの定義」の手順に従って `rgy1` を再設定して、`rgy2` をレジストリ ホストとして認識させます。

- 2 次の手順に従って、**rgy1** をバックアップレジストリ ホストにします。その後、プライマリレジストリ ホストにプロモートします。
  - a **rgy1** にログオンします。UNIX では **root** として、Windows ではローカル管理者としてログオンし、ホストのレジストリ サーバーのステータスを変更する権限を持つ必要があります。
  - b UNIX では、**rgy\_svr.conf** を編集して、文字列 **master** を削除します。次に、**rgy\_hosts.conf** を編集して、1 行目を **rgy2** に、2 行目を **rgy1** に変更します。Windows では、[スタート] ボタンをクリックし、[設定] をポイントして [コントロール パネル] をクリックします。[コントロール パネル] から、**ClearCase** プログラムを起動します。**ClearCase** プログラムの [レジストリ] タブをクリックします。[次のホストのレジストリ サーバーを使用] をクリックします。[次のホストのレジストリ サーバーを使用] に「**rgy2**」と入力します。[レジストリ] タブの [バックアップ レジストリ ホスト] ボックスに「**rgy1**」と入力します。[OK] または [適用] をクリックします。
  - c **ClearCase** を停止します。
  - d **ClearCase** を再起動します。
- 3 **rgy1** 上で **rgy\_backup** を手動で実行します。これにより、**rgy1** 上のレジストリ データベースが、**rgy2** 上のレジストリ データベースの内容で更新されます。
- 4 **rgy\_switchover** を実行します。**rgy1** をプライマリ レジストリ ホストにして、**rgy2** をバックアップ レジストリ ホストにするため、以下のように指定します。  

```
rgy_switchover -backup rgy2 rgy2 rgy1
```
- 5 アクセスできなかったクライアントを処理します。手順については、56 ページの「バックアップ レジストリ ホストをプロモートしてプライマリ レジストリ ホストにするには」を参照してください。

## レジストリ バックアップ用に設定されていないホストへのレジストリの移動

プライマリ レジストリ ホストに障害が発生したときに、バックアップ レジストリ ホストを設定していなかった場合は、バックアップしてあったレジストリ データをプライマリ レジストリ ホスト (通常のファイル システム バックアップ ツールを使用) から別のホストに復元し、そのホストをプライマリ レジストリ ホストとして設定します。

この例では、プライマリ ホスト **rgy1** に障害が発生したが、作業を引き継ぐことになるホスト **rgy2** がバックアップ レジストリ ホストとして設定されていないと仮定します。**rgy** ディレクトリのファイル システム バックアップを **rgy1** から **rgy2** の **rgy** ディレクトリに復元し、**rgy1** のすべてのクライアントが **rgy2** をレジストリ ホストとして使用するよう再設定します。以下の手順では、レジストリ データベース ファイルを通常のファイル システム バックアップから復元し、その後、**rgy\_switchover** コマンドを実行します。一部のタスクは自動化できます。

- 1 **rgy1 レジストリ ファイルをバックアップから rgy2 に復元します。** ClearCase rgy ディレクトリと **client\_list.db** ファイルをバックアップから取得し、ファイルを **rgy/backup** にロードします。これが終了すると、すべてのレジストリ データ ファイルが **backup** サブディレクトリに格納されます。レジストリ ホストでは、**client\_list.db** ファイルが **rgy** ディレクトリと同じディレクトリ (**rgy** ディレクトリ自体ではない) に保存されます。このファイルがない場合は、**rgy\_switchover** を実行してクライアントを再設定できません。この場合は、45 ページの「ClearCase ホストのレジストリとリージョンの割り当て」の手順に従ってクライアントを再設定し、**rgy2** をレジストリ ホストとして使用できるようにする必要があります。
- 2 **backup\_list ファイルを作成します。** **backup\_list** という名前のテキスト ファイルを作成して、**backup** ディレクトリに格納します。**backup\_list** の 1 行目には、プライマリ レジストリ サーバー名が記述されている必要があります。この例では、**rgy1** です。**backup\_list** の残りの行には、**rgy** ディレクトリ内にあるすべてのファイルの名前が記述されている必要があります。ファイル名は、1 行に 1 つずつ記述します。指定するのはファイル名だけであり、ファイルへのパスは不要です。例を次に示します。

```
bbase_object
bbase_tag
regions
.
.
.
view_object
view_tag
vob_object
vob_tag
vob_tag.sec
```

- 3 **rgy2 をプライマリ レジストリ ホストにプロモートします。** 56 ページの「バックアップ レジストリ ホストをプロモートしてプライマリ レジストリ ホストにするには」の手順に従ってください。

# 混在環境での ClearCase ネットワーク管理

# 4

UNIX と Windows の両方のコンピュータから共通の VOB とビューにアクセスするユーザーが ClearCase コミュニティに含まれている場合は、次の要件のほとんどまたはすべてを満たす必要があります。

- Windows と UNIX の両方で、共通なユーザー名、グループ名と、共通なプライマリ グループ メンバシップを使用すること。これは、共通の VOB とビューにアクセスするすべてのユーザーとグループに適用されます。
- 異なるプラットフォーム間でファイル システムにアクセスできること。これは、Windows クライアントでホストされた動的ビュー (MVFS) が、UNIX コンピュータでホストされた VOB ビューまたは動的ビューにアクセスする場合に必要です。
- クライアントの構成に関する問題と、ファイルの命名規則に対処すること。たとえば、大文字と小文字の区別、テキスト ファイルの行末コード、ファイル名に使用できる文字など、Windows と UNIX では異なります。

ユーザーが ClearCase コミュニティの管理者であり、そのコミュニティのコンピュータがすべて Windows であるか、すべて UNIX の場合、この章を参照する必要はありません。

## 共通なユーザー名とグループ名

---

ユーザーが ClearCase ホストのオペレーティング システムにログオンするために使用する識別情報は、VOB とビュー内の成果物に対するユーザーのアクセス権を制御する認証を確定します。これらの成果物の多くには 1 つのユーザー名と 1 つ以上のグループ名が保存されていて、その成果物を変更できるユーザーとグループを識別します。このため、UNIX コンピュータと Windows コンピュータから VOB ビューまたは動的ビューにアクセスするユーザーは、UNIX と Windows の両方で同じユーザー名を持つ必要があります。さらに、VOB オブジェクトにアクセスする、ユーザーのプライマリ グループとその他のすべてのグループも、UNIX と Windows の両方で同じ名前を持つ必要があります。VOB とビューでのアクセス権の判定に認証情報が使用される方法の詳細については、第 5 章を参照してください。

**メモ:** ClearCase 認証では、グループ メンバシップは、ディレクトリやファイルなどのファイル システム オブジェクトに対してのみ重要です。ラベルやブランチなどの VOB メタデータに対しては、所有者の名前のみが重要です。グループ メンバシップは考慮されません。

## ClearCase によるユーザー名とグループ名の比較

ClearCase によるユーザー名とグループ名の比較では、Windows では大文字と小文字が区別されず、UNIX では大文字と小文字が区別されます。パスワードは考慮されません。

名前の各タイプ (ユーザーまたはグループ) は、ネームスペースで他方と比較されますが、その比較ではユーザー名とグループ名の関係は考慮されません。

**メモ:** UNIX と Windows では、ユーザー名とグループ名の長さとそれらに使用できる文字に対して異なる制限があります。ClearCase コミュニティでのユーザー名とグループ名は、両方の環境で使用できる名前にする必要があります。

## 認証マッピング

一方の OS タイプ (UNIX または Windows) で実行されているプロセスが、他方の OS タイプの VOB またはビューのオブジェクトへのアクセスを要求した場合、プロセスの認証情報 (ユーザー名とグループ名) は、アクセス先 OS タイプの認証にマッピングされます。

認証情報に含まれるこれらの名前は比較されます。名前が一致すると、名前を組み合わせた `user.group` 認証がアクセス先 OS で評価されて、要求されたアクセスが許可されるかどうか判定されます。ユーザー名とグループ名に一致した名前が見つからなかった場合、オブジェクトが、ユーザー `nobody` またはグループ `nobody` のアクセス権限を持っていない限り、要求されたアクセスはエラーとなります。

**メモ:** 大文字と小文字が混在しているか、すべて大文字のユーザー名またはグループ名が、UNIX における大文字と小文字の区別に関する比較でエラーになった場合は、名前を小文字に変換して、再度比較されます。

認証マッピングは、各 ClearCase サーバー ホスト上で動作する `credmap_server` プロセスを利用して、クライアント プロセスごとに処理されます。

## ClearCase サーバー プロセス ユーザーに対する UNIX 認証

ClearCase サーバー プロセス ユーザー アカウントは、Windows の特別なドメイン アカウント (336 ページの「ドメイン ユーザーとグループ アカウント」を参照) であり、Windows コンピュータ上ではこのアカウントの識別情報で `albd_server` が実行されます。Windows 上の動的ビューが UNIX の VOB にアクセスする混在環境には、UNIX のユーザー アカウントが必要であり、ClearCase サーバー プロセス ユーザーの認証情報はこのアカウントにマッピングする必要があります。UNIX サーバー プロセスは、このアカウントの識別情報で実行されることはありません。このアカウントの識別情報は、ClearCase サーバー プロセス ユーザーが UNIX 上の VOB ビューまたは動的ビューへのアクセスを要求したときの、認証マッピングのみで使用されます。

この目的用の新しいアカウントを作成しても、既存のアカウントを使用してもかまいません。いずれの方法でも、次に示すガイドラインが適用されます。

- アカウントは、Windows からアクセス可能な、UNIX の VOB ビューまたは動的ビューを所有する各グループのメンバーでなければなりません。
- アカウントのプライマリ グループは、Windows ClearCase のユーザーに指定されたプライマリ グループと同じ名前にしてください。UNIX にこのグループが存在していない場合は、作成する必要があります。

**メモ:** Windows の場合、ClearCase サーバー プロセス ユーザーは単一のグループである ClearCase 管理者グループのメンバーです。このグループは特別な ClearCase 権限を持っていますが、オペレーティング システムの特別な権限はありません。いくつかの ClearCase サーバー プロセスが UNIX 上で root として実行される場合でも、UNIX 上で ClearCase サーバー プロセス ユーザー アカウントを root にマッピングすることはお勧めできません。このアカウントに対しては、権限を持つユーザーの状態は必要ありません。このアカウントを root にマッピングすると、セキュリティのリスクが発生する可能性があります。

- このアカウントに適当な名前を付けることができます。ただし、名前を付けるときには、次の条件があります。
  - Windows コンピュータは NFS クライアントを使用して UNIX 上の VOB とビューにアクセスします。その NFS クライアントでは ClearCase サーバー プロセス ユーザー アカウントを異なる名前の UNIX アカウントにマッピングすることはできません。詳細については、313 ページの「NFS クライアント製品」を参照してください。
  - UNIX の VOB/ビュー サーバー ホストが SMB サーバーを実行して Windows コンピュータによる VOB とビューへのアクセスを可能にし、ClearCase サーバー プロセス ユーザー アカウントを既存の UNIX アカウントにマッピングする必要がない場合があります。詳細については、322 ページの「SMB サーバー製品」を参照してください。

上記のいずれの場合も、UNIX アカウントは、Windows 上の ClearCase サーバー プロセス ユーザーと同じユーザー名にする必要があります。

## 異なるプラットフォーム間のファイル システムのアクセス

---

動的ビューは、サーバー ホスト上の VOB データまたはビュー データへのアクセスに、ネイティブのネットワーク ファイル システムを使用することを必要とします。そのため、Windows コンピュータから UNIX のホストされた VOB ビューまたは動的ビューへのアクセスには常に、異なるプラットフォーム間でファイル システムにアクセスする機能が必要です。ClearCase はこの機能を、独自のファイル転送メカニズム、ClearCase File Service (CCFS) として提供します。この機能は、スナップショットと Web ビューで使います。CCFS は、一方の OS タイプ上のスナップショット ビューまたは Web ビューから他方の OS タイプ上の VOB へのアクセスに使用されます。詳細については、64 ページの「ClearCase ファイル サービス」を参照してください。

## VOB とビューへのアクセスに使用されるネットワーク ファイル システム プロトコル

Windows の動的ビューから UNIX 上の VOB ビューまたは動的ビューにアクセスする必要がある場合、動的ビューは次のネットワーク ファイル システム プロトコルのいずれかを使用します。

- **NFS:** NFS (Network File Service) プロトコルは、ほとんどの UNIX コンピュータでネットワーク ファイル システムのアクセスをサポートするネイティブ プロトコルです。Windows 版の NFS クライアント製品は、NFS プロトコルを使用して UNIX ファイル システムにアクセスします。NFS クライアント製品は、UNIX の VOB とビューにアクセスする各 Windows コンピュータにインストールします。
- **SMB:** SMB (Server Message Block、また Common Internet File System、CIFS と呼ばれます) プロトコルは、Windows コンピュータがネットワーク ファイル システムのアクセスに使用するネイティブ プロトコルです。UNIX コンピュータ上で実行される SMB サーバーにより、Windows コンピュータは UNIX の VOB とビューへのアクセスに Windows のネイティブ プロトコルを使用できます。SMB サーバー製品は、Windows クライアントからアクセスする、UNIX の各 VOB サーバーまたはビュー サーバーにインストールします。

**メモ:** Rational は、Windows の NFS サーバー ソリューションと UNIX の SMB クライアントをサポートしていません。そのため、UNIX コンピュータ上の動的ビューは、Windows コンピュータにホストされた VOB ビューまたは共有動的ビューにアクセスできません。

表 1 に、ClearCase クライアントが VOB データへのアクセスに使用できるプロトコルを示します。表 2 には、ClearCase クライアントがビュー記憶領域へのアクセスに使用するプロトコルを示します。これらの表で、各コンピュータ プラットフォームに対してネイティブであるプロトコルは、「ネイティブ」と明示されています。ClearCase File Service (以下 CCFS) は ClearCase に含まれます。ほかのすべてのプロトコルでは、サードパーティ製のソフトウェアによるサポートが必要です。

**表 1      ClearCase クライアントから VOB データにアクセスするプロトコル**

クライアント プラット フォーム	UNIX 上の VOB データへの アクセス	Windows 上の VOB データ へのアクセス
Windows (動的ビュー)	サードパーティ製の NFS または SMB	ネイティブ SMB
Windows (スナップショット ビュー)	CCFS、サードパーティ製の NFS または SMB	ネイティブ SMB
UNIX (動的ビュー)	ネイティブ NFS	サポートせず
UNIX (スナップショット ビュー)	ネイティブ NFS	CCFS

**表 2 ClearCase クライアントからビュー データにアクセスするプロトコル**

クライアント プラット フォーム	UNIX 上のビュー データへの アクセス	Windows 上のビュー データ へのアクセス
Windows (動的ビュー)	サードパーティ製の NFS または SMB	ネイティブ SMB
Windows (スナップショット ビュー)	サードパーティ製の NFS または SMB	ネイティブ SMB
UNIX (動的ビュー)	ネイティブ NFS	サポートせず
UNIX (スナップショット ビュー)	ネイティブ NFS	メモを参照

**メモ:** -ngpath オプションを指定してビューを作成した場合、スナップショット ビューを使用するがローカル `view_server` プロセスを実行できない ClearCase ホストは、ClearCase のネイティブ RPC メカニズムを使用して別のホスト上の `view_server` にアクセスします。これは、ClearCase LT の一般的な設定です。

表 3 に、VOB データにアクセスするために `view_server` プロセスによって使用されるプロトコルを示します。

**表 3 view\_server から VOB データにアクセスするプロトコル**

ビュー サーバーのプラット フォーム	UNIX 上の VOB データへの アクセス	Windows 上の VOB データ へのアクセス
Windows	CCFS、サードパーティ製の NFS または SMB	ネイティブ SMB
UNIX	ネイティブ NFS	CCFS

## NAS と異なるプラットフォーム間のファイル アクセス ツール

Rational では VOB とビュー データの保存のため、NAS (Network-Attached Storage) デバイスをサポートします。これらのデバイスは VOB 記憶領域またはビュー記憶領域をホストするように構成でき、UNIX と Windows コンピュータの両方にネイティブのファイル システム アクセスを提供します。その際、ClearCase クライアント ホストまたはサーバー ホストに別のソフトウェアをインストールすることを必要としません。詳細については、16 ページの「ClearCase と NAS (Network-Attached Storage) デバイス」を参照してください。

これらの NAS デバイスに加え、Rational は ClearCase 用にサードパーティ製の異なるプラットフォーム間のファイル アクセス ツールをサポートしています。これらのツールには次の 2 つのカテゴリがあります。

- UNIX プラットフォーム用の SMB サーバー。Windows クライアントから UNIX ホスト上のファイル システムに、ネイティブ SMB/CIFS によるアクセスを提供します。
- Windows 用の NFS クライアント プログラム。Windows クライアントから NFS プロトコルを使用して UNIX コンピュータのファイル システムにアクセスすることを可能にします。

これらのツールのいずれも、Windows コンピュータ上の動的ビューから UNIX コンピュータにホストされた VOB にアクセスすることを可能にします。また、Windows コンピュータから、UNIX コンピュータにホストされた共有動的ビューを使用できるようにします。

サポートされている、異なるプラットフォーム間のファイル システムのアクセス ツールの詳細については、付録 A を参照してください。

## ClearCase ファイル サービス

ClearCase ファイル サービスは TCP/IP ベースのメカニズムであり、VOB サーバーとスナップショット ビューとの間で、異なるプラットフォーム間のファイル転送を可能にします。このサービスは、Windows コンピュータ上のスナップショット ビューによる UNIX コンピュータ上の VOB データへのアクセスと、UNIX コンピュータ上のスナップショット ビューによる Windows 上の VOB データへのアクセスをサポートします。Web ビューも CCFS を使用して、同じタイプのアクセスを行います。

**メモ:** スナップショット ビューと Web ビューは、同じプラットフォーム タイプの VOB にアクセスします。これらのビューは、常にプラットフォームのネイティブのネットワーク ファイル システムを使用し、CCFS を使用しません。

ClearCase クライアントがスナップショット ビューと Web ビューのみを使用する場合、サードパーティ製の異なるプラットフォーム間のファイル アクセス ソリューションは VOB データへのアクセスを要求されることがありません。

CCFS が有効である場合、スナップショット ビュー クライアントと VOB サーバーの間のファイル転送 (たとえば、チェックアウト、チェックイン、スナップショット ビューの作成や更新) が、標準の TCP/IP 接続を介して行われます。VOB サーバーとスナップショット ビューのビュー サーバーの間のファイル転送も TCP/IP 接続で行われます。

CCFS は、ClearCase を実行している UNIX コンピュータ上では常に有効です。CCFS を Windows 上で有効にするには、[コントロール パネル] で ClearCase プログラムを実行します。

## Windows 上での CCFS の有効化

デフォルトでは、CCFS は、ClearCase を実行している Windows コンピュータ上では無効です。Windows コンピュータ上で CCFS を有効にするには、以下の操作を実行します。

- 1 [スタート] ボタンをクリックし、[設定] をポイントして、[コントロール パネル] をクリックします。[コントロール パネル] から、ClearCase プログラムを起動します。

- 2 [オプション] タブで、[CCFS を使用して UNIX の VOB にアクセスする] チェック ボックスをオンにして CCFS を有効にします。CCFS を無効にするにはこのチェック ボックスをオフにします。
- 3 [OK] をクリックします。
- 4 コンピュータをシャットダウンし、再起動します。

## CCFS と Windows のリモート ビュー サーバー ホスト

スナップショット ビューを使用し、ローカル **view\_server** プロセスを実行できない ClearCase ホストは、UNIX の VOB とのアクセス時に唯一のファイル転送メカニズムとして CCFS を使用します。これらのクライアント上のスナップショット ビューには、別の Windows コンピュータが必要です。その Windows コンピュータで、**view\_server** プロセスを実行し、これらのビューをサポートするビュー記憶ディレクトリ (関連付けられたビュー記憶ディレクトリ) をホストします。

CCFS は、この役割を実行する Windows コンピュータ上で有効であることが必要です。NFS クライアント製品を使用して UNIX の VOB にアクセスするように設定された Windows コンピュータは、ローカル **view\_server** プロセスを実行できない ClearCase クライアントに **view\_server** のサポートを提供できません。

## 大文字/小文字の区別

---

UNIX と Windows は、ファイル名をルックアップするときの大文字と小文字の区別で、異なる規則に従います。UNIX では大文字と小文字は区別され、Windows では区別されません。通常、アプリケーションとユーザーはこれらのオペレーティングシステムの規則に従いますが、これらの規則は、ユーザーが UNIX と Windows コンピュータで、VOB とビューを共有して作業する場合に発生する問題の原因になります。ClearCase での大文字と小文字の区別には、次の 2 つの側面があります。

- ClearCase と、**cleartool** や **clearmake** など VOB のパス名を参照するすべての ClearCase アプリケーションは、UNIX 上でも Windows 上でも大文字と小文字を区別します。
- UNIX 上の MVFS は常に、大文字と小文字を区別します。Windows 上の MVFS は、複数の大文字/小文字の区別オプションと大文字/小文字の保存オプションをサポートするように設定できます。

一般に、Windows と UNIX のいずれを使用する場合も、大文字/小文字の区別の問題をほとんど考慮することなく ClearCase を使用できます。OS の混在環境で予期しない動作が起こる場合は、この項を参照してください。

## 一般的な推奨事項

共通して発生する、大文字/小文字の区別方法の違いに関連する問題を回避するには、次の一般的な推奨事項を実践してください。

- VOB とビューでは、大文字/小文字が異なるだけのファイル名やディレクトリ名は作成しないでください。大文字と小文字を区別しない MVFS では、大文字/小文字が異なるだけの 2 つのファイル名を区別できないため、このようなファイルにアクセスすると、不確定な結果を招きます。
- Windows 上の MVFS を、大文字と小文字を区別しないように設定します (これがデフォルト設定です)。
- NFS クライアントまたは SMB サーバー製品を使用する場合は、大文字/小文字の自動変換機能を無効にします (付録 A を参照してください)。

以下の各項では、UNIX と Windows の両方で起こり得る大文字/小文字の区別に関する問題について解決方法を含めて詳しく説明します。

## 大文字/小文字の区別と MVFS

マルチバージョン ファイル システム (MVFS) は ClearCase の機能で、動的ビューをサポートします。MVFS はファイル システムであり、大文字/小文字の区別は重要です。このため、OS が混在するネットワークでは必要に応じて、大文字/小文字の区別を再設定してください。詳細については、179 ページの「マルチバージョン ファイル システム」を参照してください。

**メモ:** ClearCase LT は MVFS をサポートしていません。

UNIX では、ネイティブ ファイル システムもすべての ClearCase コンポーネント (MVFS、cleartool など) も大文字と小文字を区別します。MVFS は常に大文字と小文字を区別してファイルのルックアップを使用し、大文字/小文字の変換は行いません。

Windows では、ネイティブ ファイル システムで大文字/小文字は区別されませんが、大文字/小文字は保存されます。Windows のネイティブ ファイル システムでは大文字と小文字を区別しないでファイルのルックアップを行うため、MVFS は、Windows 上のデフォルト設定では大文字と小文字を区別せずにルックアップを行います。Windows のネイティブ ファイル システムでは、ファイル作成時に大文字/小文字が変換されずに保存されます。Windows 上では MVFS はファイル名を小文字に変換する設定になっています。

異なる大文字/小文字の区別オプションを使用するように Windows の MVFS を再設定するには、[コントロール パネル] の [ClearCase] プログラムで [MVFS] タブを使用します。

- [大文字と小文字を区別しない] チェック ボックスをオン (デフォルトかつ推奨の設定) にすると、MVFS は大文字と小文字を区別しないでファイルのルックアップを行います。この設定で MVFS がビュープライベート ファイルに書き込みを行う場合、ファイル名の大文字化は [大文字と小文字を保存する] チェック ボックスの設定によって次のようになります。

- [大文字と小文字を保存する] チェック ボックスをオンにした場合、MVFS はビュープライベート ファイル名の`大文字`と`小文字`を保存します。
  - [大文字と小文字を保存する] チェック ボックスをオフにした場合、MVFS はビュープライベート ファイル名を`小文字`に変換します。これはデフォルト設定です。
- [大文字と小文字を区別しない] チェック ボックスをオフにすると、MVFS は`大文字`と`小文字`を区別してファイルのルックアップを行います。この設定では、MVFS はどのファイルを作成するときもファイル名の`大文字`と`小文字`を保存します。

MVFS オプションへの変更は、コンピュータを再起動するまで有効になりません。

**メモ:** MVFS で`大文字`と`小文字`を区別する場合、`大文字`と`小文字`を区別してファイル名を指定しない Windows アプリケーションではエラーが発生する可能性があります。

## 大文字と小文字を保存する MVFS の使用が推奨される状況

デフォルト設定 ([大文字と小文字を区別しない] がオン、[大文字と小文字を保存する] がオフ) の場合、MVFS は、動的ビューで作成されたビュープライベート ファイルの名前を`小文字`に変換します。この設定は、意図的に`大文字`と`小文字`を混在させた名前で作成するツールの操作を妨げることになります。たとえば、Java 開発環境で `util.JAR` という名前のファイルを作成した場合、`大文字`と`小文字`を区別しない MVFS では、そのファイルは `util.jar` として書き込まれます。ファイル `util.JAR` を参照する開発ツールは、このファイルを見つけることができません。この種の問題を修正するには、MVFS を、`大文字`と`小文字`を保存する設定 ([大文字と小文字を区別しない] をオフ、[大文字と小文字を保存する] をオン) にします。

## スナップショットと Web ビューでの大文字/小文字の区別

スナップショットと Web ビューは、ファイルの読み込み/書き込みを行うスナップショットビュー ディレクトリが配置されるホスト上の、ネイティブのファイルシステムを使用します。Windows の場合、これらのファイル システムは`大文字`と`小文字`を区別しないでファイルのルックアップを行います。ファイル名の作成では一般に`大文字`と`小文字`が変換されずに保存されます。UNIX の場合、ファイルのルックアップでは常に`大文字`と`小文字`を区別し、ファイル名の作成では常に`大文字`と`小文字`が保存されます。

## ビューのテキスト モードの構成

---

UNIX と Windows は、テキスト ファイルの行末コードの記述で、異なる規則に従います。一般的に、UNIX のユーティリティとアプリケーションでは行末コードを `<LF>` (改行) 1 文字のみで表します。Windows のユーティリティとアプリケーションでは行末を `<CR><LF>` (復帰、改行) の 2 文字で表します。Windows アプリケーションには、両方の形式のファイルを読み取って表示できるもの、ファイルの書き込みは常に `<CR><LF>` 形式で行うもの、どちらの形式を使用するかを指定できるものがあります。

このように規則が異なる結果、UNIX と Windows プラットフォームの両方で編集可能なテキスト ファイルで、行末コードの問題が発生します。たとえば、ファイルに

```
abc
def
ghi
```

という内容が含まれる場合、これを一般的な Windows のエディタ (メモ帳) で作成し、vi などの一般的な UNIX エディタで読み込むと、次のように表示されます。

```
abc^M
def^M
ghi^M
```

UNIX のテキスト エディタは、<CR> 文字を ^M と表示します。同じファイルを UNIX エディタで作成し、Windows のエディタで読み込むと、次のように表示されます。

```
abc■def■ghi
```

オペレーティング システムの混在環境での並行開発をより適切にサポートするため、ClearCase では、ビューのテキスト モードを設定できます。これにより、テキスト ファイルをアプリケーションで表示するときの行終端記号が制御されます。

## テキスト モード

それぞれのビューには、行終端記号シーケンスの処理方法を指定するテキスト モード設定があります。この設定は、エレメント タイプが **text\_file** または **text\_file** タイプのサブタイプであるファイル エレメントにのみ適用されます。ビューのテキスト モードは、ビューの作成時に指定します。ビューの作成後に、ビューのテキスト モードを変更することはできません。

サイトのデフォルト値は **ClearCase** レジストリに保存され、新しいビューの作成時に使用されるテキスト モードを決定します。このデフォルトは、ビューを作成するコマンド行、GUI で上書きできます。デフォルト設定の詳細については、**setsite** のリファレンス ページを参照してください。

次の 3 つのテキスト モードのいずれかでビューを作成できます。

- **トランスペアレント テキスト モード**。トランスペアレント テキスト モードで作成されたビューの場合、**ClearCase** は行終端記号処理を行いません。サイトのデフォルトがほかに指定されていない場合、ビューは常にトランスペアレント テキスト モードで作成されます。サイトのデフォルトにかかわらず、トランスペアレント テキスト モードでビューを作成するには、ビュー作成ウィザードの [詳細設定] オプションで、[UNIX/Windows 相互運用テキスト モードを使用] チェック ボックスをオフにするか、**mkview** コマンドで **-tmode transparent** オプションを使用します。

自社のサイトのすべての開発者が同じ開発プラットフォーム (Windows または UNIX) を使用する場合、またはいずれの行末コード規則にも互換性のあるツールを使用する場合、すべてのビューはトランスペアレント テキスト モードで作成する必要があります。

- **insert\_cr テキスト モード。**insert\_cr (UNIX/Windows 相互運用) テキスト モードで作成されたビューの場合、ClearCase はすべての <LF> 文字の前に <CR> 文字を挿入します。insert\_cr テキスト モードでビューを作成するには、ビュー作成ウィザードの [詳細設定] オプションで [UNIX/Windows 相互運用テキスト モードを使用] チェック ボックスをオンにするか、mkview コマンドで -tmode insert\_cr オプションを使用します。
- **strip\_cr テキスト モード。**strip\_cr (CR 除去) テキスト モードで作成されたビューの場合、ClearCase はすべての <CR><LF> 文字シーケンスから <CR> 文字を取り除きます。strip\_cr テキスト モードでビューを作成するには、mkview コマンドで -tmode strip\_cr オプションを使用します。ビュー作成ウィザードから、strip\_cr モードでビューを作成することはできません。

insert\_cr テキスト モードまたは strip\_cr テキスト モードのいずれかで作成されたスナップショット ビューでは、ClearCase はビューを更新するたびに <CR> 文字を追加または削除します。動的ビューでは、ClearCase はファイルを開いて読み込むときに <CR> 文字を追加または削除します。スナップショット ビューと動的ビューのいずれでも、ClearCase はチェックイン処理では上記と逆の <CR> 処理 (適宜 <CR> 文字を追加または削除) を行います。

## ビューのテキスト モードの特定

ビューのテキスト モードがわからないときは、次のいずれかの方法で調べることができます。

- Windows エクスプローラで、ビューを表すドライブを右クリックします。次に、[ClearCase] をクリックし、[ビューのプロパティ] をクリックします。ビューのテキスト モードが [アクセス] タブに表示されます。
- cleartool lsview -properties -full コマンドを実行します。

これらの方法により、トランスペアレント テキスト モードであれば unix、strip\_cr テキスト モードであれば strip\_cr、insert\_cr テキスト モードであれば msdos と表示されます。

## ビューのテキスト モードの選択

ClearCase では、ビューのテキスト モードに基づいて VOB へのアクセスを管理するポリシーを強制しません。あるユーザーが「適切でない」テキスト モード構成のビューでファイルを編集した場合、ほかのユーザーがそのファイルを編集する際に問題が発生することがあります。Windows と UNIX の両方の開発プラットフォームが混在するほとんどのサイトでは、主要な開発プラットフォームのユーザーがトランスペアレント テキスト モードでビューを作成できるようにし、strip\_cr テキスト モードまたは insert\_cr テキスト モードの使用は異なる行末コード規則を適用する少数のプラットフォームに限定するようにします。

ほとんどのユーザーが UNIX 上でテキスト ファイルを編集する場合、次のように設定します。

- UNIX クライアントは、トランスペアレント テキスト モードで作成されたビューを使用します。
- Windows クライアントは、insert\_cr テキスト モードで作成されたビューを使用します。

ほとんどのユーザーが Windows 上でテキストファイルを編集する場合、次のように設定します。

- Windows クライアントは、トランスペアレント テキスト モードで作成されたビューを使用します。
- UNIX クライアントは、**strip\_cr** テキスト モードで作成されたビューを使用します。

どのようなポリシーを採用する場合でも、クライアント プラットフォーム、ビューのテキストモード、エレメントの組み合わせの整合性を維持することが重要です。たとえば、UNIX 上のあるユーザーが **strip\_cr** テキスト モードまたは **insert\_cr** テキスト モードでエレメントのバージョンを作成し、UNIX 上の別のユーザーがトランスペアレント テキスト モードのビューで同じエレメントのバージョンを作成すると、これらの 2 つのバージョンの比較やマージ処理は難しくなります。

## 古い VOB での UNIX/Windows 相互運用テキスト モードのサポートの有効化

最新リリースの Rational ClearCase で作成された VOB は、いずれのテキスト モードでのビューとも互換性があります。以前のバージョンの ClearCase で作成された VOB は、VOB で **msdostext\_mode** コマンドを実行するまで、トランスペアレント テキスト モードのビューとのみ互換性があります。

**msdostext\_mode** は VOB 所有者または権限を持つユーザーのみが実行できます。このコマンドの構文は、次のとおりです。

```
ccase-home-dir/etc/utils/msdostext_mode [ -d ] vob-storage-pname
```

オプションを指定しない場合、**msdostext\_mode** は以下のことを行います。

- エレメント タイプが **text\_file** または **text\_file** タイプのサブタイプであるすべてのファイルエレメントの、すべてのバージョンについて、**strip\_cr** テキスト モードまたは **insert\_cr** テキスト モードで作成されたビューでこれらのバージョンにアクセスするための情報を作成し、VOB データベースに格納します。
- **strip\_cr** テキスト モードと **insert\_cr** テキスト モードのサポートを有効にします。これにより、新しく作成されたバージョンについて、この情報が記録されます。

**-d** オプションを指定すると、**msdostext\_mode** は **strip\_cr** テキスト モードと **insert\_cr** テキスト モードのサポートを無効にします。

**msdostext\_mode** コマンドは、ファイルを変換したり、変更したりしません。VOB データベース内のテキスト ファイル バージョンについて記録されている情報にのみ影響を与えます。

## VOB が UNIX/Windows 相互運用テキスト モードをサポートしているかどうかを特定するには

VOB が UNIX/Windows 相互運用テキスト モードをサポートしているかどうかを調べるには、次のコマンドを実行します。

```
cleartool dump vob:vob-tag
```

出力の flags 行に pc\_line\_count という文字列が含まれていれば、この VOB は UNIX/Windows 相互運用テキスト モードをサポートしています。たとえば、¥pc\_src が UNIX/Windows 相互運用テキスト モードをサポートしているかどうかを調べるには、次のコマンドを実行します。

```
cleartool dump vob:¥pc_src
```

このコマンドの出力に次のような内容が含まれていたら、この VOB は UNIX/Windows 相互運用テキスト モードをサポートしています。

```
flags: predefined, pc_line_count, unrestricted
```

## MultiSite ユーザー向けの特別な手順

ClearCase MultiSite は、VOB のテキスト モード サポート特性のレプリカを作成しません。VOB ファミリのレプリカのいずれかが UNIX/Windows 相互運用テキスト モードのサポートを有効にした場合、VOB ファミリのすべてのレプリカはローカル サイトで個別に有効にする必要があります。

- 1 すべての VOB レプリカを同期します。
- 2 -nusers vob\_owner を指定してすべてのレプリカをロックします。ここで、vob\_owner はレプリカの VOB 所有者のユーザー名です。
- 3 vob\_owner として、msdostext\_mode をレプリカごとに実行します。
- 4 すべてのレプリカをアンロックします。

## サポートされている文字セットの相違

Windows のファイル名では、UNIX のファイル名で使えるいくつかの文字を使用できません。このような文字を含むファイル名は Windows 上の MVFS では認識されず、Windows のスナップショット ビューにロードできません。これらの文字を含む VOB のエレメント名は、Windows のビューには表示されません。表 4 に、これらの文字を示します。

表 4 Windows のファイル名で使えない文字

?	*	/	¥		<	>
---	---	---	---	--	---	---



この章では、Rational ClearCase が VOB データとビュー データへのアクセスをどのように制御するかについて説明します。

## VOB とビューのアクセス コントロールの基本事項

---

ClearCase には、VOB データとビュー データの作成、読み込み、書き込み、実行、削除をどのユーザーに許可するかを決定するアクセス コントロール機能が実装されています。アクセス コントロールは、ユーザーとその所属グループ、ClearCase オブジェクトの保護、ユーザー プロセス、またはユーザーに代わって ClearCase データにアクセスするアプリケーション プログラムの認証などの相互関係によって決定されます。

### ユーザーとグループ

ClearCase にはユーザーやグループ用の独自のアカウントはありませんが、ClearCase ホストのオペレーティング システムにログオンするときに使用するユーザー識別情報によって、ClearCase のユーザーの認証情報を確立します。この認証情報は、ClearCase の制御下にあるオブジェクトへのアクセスをユーザーが要求するたびにチェックされるので、オペレーティング システムで定義されたユーザー名、グループ名、各ユーザーのグループ メンバシップは、すべての ClearCase ホスト上で整合性が保たれている必要があります。通常、Windows ドメインまたは UNIX Network Information System (NIS) などのネットワーク アカウント データベースによって、この整合性を維持します。

**メモ:** ユーザーが UNIX ホストと Windows ホストから共通の VOB とビューのセットにアクセスする環境では、この整合性は両方のプラットフォームで維持する必要があります。ユーザー名とグループ名、各ユーザーのグループ メンバシップは Windows 上と UNIX 上で一致している必要があります。詳細については、59 ページの「共通なユーザー名とグループ名」を参照してください。

## プライマリ グループ

ユーザーは 1 つまたは複数のグループに所属することができます。複数のグループに所属する場合、そのうちの 1 つがユーザーのプライマリ グループとなります。あるコミュニティに所属するすべての ClearCase ユーザーが、同じプライマリ グループのメンバーになるように設定することをお勧めします。既存のグループ、または ClearCase コミュニティで使用するために明示的に作成したグループを、プライマリ グループにすることができます。

プライマリ グループの定義は、Windows と UNIX で異なります。

- UNIX では、プライマリ グループは NIS passwd データベース内のユーザー エントリに定義されます。
- Windows では、プライマリ グループはユーザーのドメイン アカウントを作成するときに指定されます。ただし、ユーザーが複数のグループのメンバーである場合、アプリケーションの要求に対してプライマリ グループ名が返されない場合もあります。ClearCase ユーザーが複数のグループのメンバーである場合は、Windows 上で、ユーザー環境変数 CLEARCASE\_PRIMARY\_GROUP を、ClearCase ユーザーのプライマリ グループに指定されているグループの名前に設定する必要があります (337 ページの「ClearCase プライマリ グループの設定」を参照)。

## ユーザーが 33 以上のグループに所属している場合の制限

ユーザーが 33 以上のグループに所属している場合、ClearCase で認識されるのは最初の 32 グループ (UNIX の場合は GID、Windows の場合は SID の数値順に判定) だけです。ユーザーに対してユーザー環境変数 CLEARCASE\_GROUPS が設定されている場合、ClearCase はこの変数の値に指定されているセミコロン区切りのグループ名を最初に参照して、ユーザーが所属するグループのリストを決定します。

## 権限を持つユーザーとグループ

典型的な ClearCase コミュニティは、次の 2 つのユーザー クラスで構成されています。

- 一般ユーザーは、自身が作成した、または自身が所属するグループに割り当てられている ClearCase リソース (VOB、ビューと、これらの関連オブジェクト) を変更または削除する権限を持っています。
- 権限を持つユーザーは、無制限にあらゆる ClearCase リソースを作成、変更、削除する権限を持っています。

ClearCase LT では、権限を必要とする操作は ClearCase LT サーバー ホスト上で直接実行する必要があります。

- UNIX 上の ClearCase ホストまたは ClearCase LT サーバーの場合、権限を持つユーザーは **root** ユーザーです。**root** ユーザーがリモート ホストにログオンしている場合は、一部制限事項が適用されます。この制限事項については、75 ページの「リモート **root** ユーザーの制限付き権限」で説明します。
- Windows 上の ClearCase ホストの場合、権限を持つユーザーは、336 ページの「ドメイン ユーザーとグループ アカウント」で説明している、ClearCase 管理者グループのメンバーです。
- Windows 上の ClearCase LT サーバーの場合、権限を持つユーザーは ClearCase LT サーバー ホストの管理者グループのメンバーです。

## ClearCase サーバー プロセス ユーザー

Windows 上の ClearCase ホストの場合、**albd\_server** プログラムは、ClearCase サイトの準備段階で作成される、ClearCase サーバー プロセス ユーザーと呼ばれる特別なユーザー アカウントの ID で実行されます。このユーザーは ClearCase 管理者グループのメンバーであるため、ステータスは権限を持つユーザーとなります。ClearCase サーバー プロセス ユーザーの詳細については、336 ページの「ドメイン ユーザーとグループ アカウント」を参照してください。ClearCase LT では ClearCase サーバー プロセス ユーザーは存在しません。

## リモート root ユーザーの制限付き権限

UNIX ホスト上での大半の ClearCase 操作では、ローカル ホストにログオンしていないリモート **root** ユーザーを、権限を持つユーザーとして扱います。ただし、いくつかの例外があります。

- **root** としてログオンしているユーザーがリモート UNIX ホスト上のビューにアクセスしようとする、そのユーザーの識別情報は **nobody.nobody** (識別できないユーザー、識別できないグループ) として解釈されます。これは、NFS を実装している UNIX では一般的なもので、NFS と同レベルのセキュリティを提供します。ClearCase では、リモート **root** ユーザーからのアクセス要求を制御するマウント オプションはサポートされていないため、**nobody.nobody** に対する明示的なアクセス権限がビューで指定されていない限り、リモート **root** ユーザーのビュー アクセスは許可されません。
- ビュープライベート ファイルのユーザーまたはグループの所有権を変更する操作 (UNIX の **chown** コマンドなど) をリモート **root** から実行すると、失敗します。

## ユーザー プロセス

プロセスが VOB またはビュー データへのアクセスを要求すると、ClearCase によってプロセスの認証情報がチェックされ、要求されたアクセス形式を承認するかどうかが決まります。この決定を行う上で、次のプロセス認証情報が重要になります。

- **ユーザー**: プロセスを起動したユーザー名です。
- **プライマリ グループ**: プロセスを起動したユーザーが所属するプライマリ グループです。
- **補足グループ リスト**: プロセスを起動したユーザーがメンバーとして所属するその他のグループです。

この章では、プロセスが所属するプライマリ グループとその他のグループをまとめてプロセスのグループと呼びます。

## ClearCase オブジェクト

次の ClearCase オブジェクトがアクセス コントロールの対象になります。

- VOB
- エレメントとバージョン
- タイプとタイプのインスタンス (ラベル、ブランチ、属性など)
- 統一変更管理 (UCM) オブジェクト (プロジェクト、アクティビティ、ストリームなど)
- VOB 記憶プール
- ビュー
- 動的ビュー、ビュープライベート ファイル、ビュープライベート ディレクトリ、派生オブジェクト

各オブジェクトは、アクセス コントロールの重要な要素となる、次のプロパティを 1 つまたは複数持ちます。

- **所有者**: 所有者はユーザーです。初期値は、そのオブジェクトを作成したプロセスのユーザー ID です。一部のオブジェクトでは初期値を変更できます。
- **グループ**: 初期値は、そのオブジェクトを作成したプロセスのプライマリ グループです。一部のオブジェクトでは初期値を変更できます。
- **保護モード**: 一部のオブジェクトには、保護モードも設定できます。保護モードは、各ユーザー カテゴリに対応する次の 3 つの権限セットで構成されます。
  - オブジェクトの所有者
  - オブジェクトのグループに所属するメンバー
  - その他のすべてのユーザー

各権限セットは、そのカテゴリに属するユーザーに対する 3 つのブール値で構成されます。各ブール値は、ユーザーがオブジェクトに次の操作を行う権限を持つかどうかを示します。

- 読み取り権限。オブジェクトのデータを表示する権限です。
- 書き込み権限。オブジェクトのデータを変更する権限です。**VOB** やディレクトリのように内部にほかのオブジェクトを含むコンテナ オブジェクトの場合は、一般にコンテナ オブジェクト内のオブジェクトを作成または削除する権限を意味します。
- 実行権限。ファイル オブジェクトの場合は、そのファイルを実行可能プログラムとして実行する権限を意味します。ディレクトリ オブジェクトの場合は、そのディレクトリを検索する権限を意味します。

## 保護モード

ClearCase オブジェクトの保護モードは、表 5 にまとめられています。通常、オブジェクトの保護モードに関する情報は、保護モードを構成する各ブール値の 1 文字の略号で示します。この表記方法も表 5 に示します。

**表 5 ClearCase オブジェクトの保護モード**

ユーザー カテゴリ	読み取り権限	書き込み権限	実行権限
オブジェクトの所有者	あり (r) または、なし (-)	あり (w) または、なし (-)	あり (x) または、なし (-)
オブジェクトのグループのメンバー	あり (r) または、なし (-)	あり (w) または、なし (-)	あり (x) または、なし (-)
その他	あり (r) または、なし (-)	あり (w) または、なし (-)	あり (x) または、なし (-)

たとえば、あるビューで作業しているときに、**lib** というディレクトリ エレメントの保護モードを調べる場合を考えてみます。**lib** の所有者とグループは読み取り権限、書き込み権限、実行権限を持ちますが、ほかのユーザーは読み取り権限と実行権限しか持っていません。**cleartool describe** コマンドを実行すると、このエレメントの所有者 (**akp**)、グループ (**clearusers**) と共に、保護モードが次のように表示されます。

**cleartool describe lib**

```
...
  エレメント保護 :
    ユーザー : akp          rwx
    グループ : clearusers  rwx
    その他 :                r-x
...
```

1 文字の略号以外にも、ClearCase では、オブジェクト権限をコンパクトに表示するために整数値を使用することがあります。各ユーザー カテゴリ (所有者、グループ、その他) の権限を 0 ～ 7 の数値で表す場合の各数値の意味を、表 6 に示します。

**表 6 ClearCase オブジェクトの保護モードを表す数値**

数値	読み取り権限	書き込み権限	実行権限
0	なし	なし	なし
1	なし	なし	あり
2	なし	あり	なし
3	なし	あり	あり
4	あり	なし	なし
5	あり	なし	あり
6	あり	あり	なし
7	あり	あり	あり

オブジェクトに対する権限を、所有者、グループ、その他の順に 3 桁の数値にして、保護モードを表します。たとえば、750 で表される保護モードの場合、各ユーザー カテゴリの権限は次のようになります。

数値	ユーザー カテゴリ	権限
7	所有者	読み取り、書き込み、実行
5	グループ	読み取り、実行
0	その他	なし

## VOB データとビュー データへのアクセス

プロセスが VOB 内のオブジェクトやビューにアクセスできるかどうかは、次の要素で決まります。

- プロセスのユーザーとグループ
- オブジェクトの所有者とグループ
- 保護モードを持つオブジェクトの場合は、保護モード

プロセスが保護されたオブジェクトにアクセスしようとする、次のアルゴリズムによってアクセスを許可するかどうかを決定します。

- 1 プロセスはオブジェクトの所有者のユーザー ID を持つか。
  - 持つ場合: オブジェクトの保護モードの [所有者] カテゴリに従ってアクセスを許可または拒否します。
  - 持たない場合: 手順 2 に進みます。
- 2 プロセスがオブジェクトのグループのグループ ID を持つか。
  - 持つ場合: オブジェクトの保護モードの [グループ] カテゴリに従ってアクセスを許可または拒否します。
  - 持たない場合: 手順 3 に進みます。
- 3 オブジェクトの保護モードの [その他] カテゴリに従ってアクセスを許可または拒否します。

保護モードを持たないオブジェクトの場合、ClearCase はオブジェクトのタイプに固有の規則に従って、アクセスを許可するかどうかを決定します。詳細については、79 ページの「VOB と VOB オブジェクトに対するアクセス コントロール」と 85 ページの「ビューとビュー オブジェクトに対するアクセス コントロール」を参照してください。

プロセスがあるオブジェクトの 1 つまたは複数のコンテナ オブジェクトにアクセスできる場合にのみ、ClearCase がそのプロセスにオブジェクトへのアクセスを許可する場合があります。たとえば、動的ビューでビュープライベート ファイルを作成するには、ファイルが格納されているディレクトリだけでなく、ビュー自体への書き込み権限が必要です。

## VOB と VOB オブジェクトに対するアクセス コントロール

---

VOB と VOB 内のオブジェクトは、アクセス コントロールの対象になります。VOB オブジェクトには次のものがあります。

- エレメントとバージョン
- タイプとタイプのインスタンス (ラベル、ブランチ、属性など)
- 統一変更管理 (UCM) オブジェクト (プロジェクト、フォルダ、アクティビティ、ストリームなど)
- VOB 記憶プール

## VOB に対するアクセス コントロール

VOB は、アクセス コントロールの重要な要素となる、次のプロパティを持ちます。

- **所有者**: 初期値は、その VOB を作成したプロセスのユーザーです。
- **グループ**: 初期値は、その VOB を作成したプロセスのプライマリ グループです。
- **補足グループ リスト**: Windows 上で作成された VOB には、初期値はありません。  
UNIX 上で作成された場合、VOB 所有者のグループ リストが含まれています。

VOB は保護モードを持ちません。この章では、VOB のプライマリ グループとその他のグループを VOB のグループと呼びます。

`cleartool describe` コマンドを実行すると、VOB の所有者、グループ、補足グループ リストを表示することができます。

VOB の作成後、権限を持つユーザーは `cleartool protectvob` コマンドを実行して、VOB の所有者、グループ、補足グループ リストを変更することができます。

**メモ**: `protectvob` を使用して、ClearCase 管理者グループを VOB の補足グループ リストに追加することはできません。このグループのメンバーは、すべての VOB オブジェクトに対する完全なアクセス権限を持っています。

### VOB を作成する権限

どのユーザーも VOB を作成できます。

### VOB を削除する権限

その VOB の所有者または権限を持つユーザーが VOB を削除できます。

### VOB を読み取る権限

VOB を直接読み取ることはできません。VOB を読み取るとは、VOB 内のオブジェクトを読み取ることを意味します。「エレメントに対するアクセス コントロール」と「その他の VOB オブジェクトに対するアクセス コントロール」を参照してください。

### VOB に書き込む権限

VOB に直接書き込むことはできません。VOB の書き込みには、VOB 内のオブジェクトを作成、削除することも含まれます。「エレメントに対するアクセス コントロール」、「その他の VOB オブジェクトに対するアクセス コントロール」を参照してください。

## VOB を実行する権限

VOB を直接実行することはできません。VOB の実行とは、VOB 内のオブジェクトを実行することを意味します。「エレメントに対するアクセス コントロール」および「その他の VOB オブジェクトに対するアクセス コントロール」を参照してください。

## エレメントに対するアクセス コントロール

エレメントは、アクセス コントロールの重要な要素となる、次のプロパティを持ちます。

- **所有者**: 初期値は、そのエレメントを作成したプロセスのユーザー ID です。
- **グループ**: 初期グループは、UNIX ホストと Windows ホストで異なります。
  - UNIX の場合、そのエレメントを作成したプロセスのプライマリ グループ ID です。
  - Windows の場合、プライマリ グループが VOB のグループ リストに存在している場合は、そのエレメントを作成したプロセスのプライマリ グループ ID です。存在しない場合、そのエレメントを作成したプロセスのグループ リストと VOB のグループ リストに含まれる任意のグループです。

Windows でも UNIX でも、エレメントのグループはいずれかの VOB のグループに属している必要があります。

- **保護モード**: ファイル エレメントの保護モードの初期値は、UNIX ホストと Windows ホストで異なります。
  - UNIX の場合、既存のビュープライベート ファイルから作成された場合は、そのファイルと同じ保護モードになります。ただし、どのユーザー カテゴリも書き込み権限だけはありません。既存のファイルから作成しなかった場合は、すべてのユーザー カテゴリに読み取り権限だけが与えられます。**umask** が 0 の場合、ディレクトリ エレメントは、初期値としてすべてのユーザー カテゴリに対して読み取り、書き込み、実行権限を設定します。0 以外の場合、読み取り、書き込み、実行権限の初期値は、**umask** の値に基づいて決定されます。
  - Windows の場合、ファイル エレメントは、初期値としてすべてのユーザー カテゴリに対して読み取り権限を設定します。**cleartool chmod** コマンドを使用して、エレメントの実行権限を明示的に追加する必要があります。ディレクトリ エレメントは、初期値としてすべてのユーザー カテゴリに対して読み取り、書き込み、実行権限を設定します。

エレメントの所有者、グループ、保護モードは、そのエレメントのすべてのバージョンで同じです。

**cleartool describe** コマンド、Windows の場合は Windows エクスプローラ、ClearCase エクスプローラの [エレメントのプロパティ] ダイアログ ボックスのいずれかを使用すると、エレメントの所有者、グループ、保護モードを表示できます。

エレメントの作成後に、そのエレメントの所有者、VOB の所有者、権限を持つユーザーは、**cleartool protect** コマンドを使用して、そのエレメントの所有者、グループ、保護モードを変更できます。

## エレメントを作成する権限

VOB を作成すると、単一のエレメント (VOB ルート ディレクトリ) が作成されます。このエレメントは、VOB 内のほかのあらゆるエレメントのコンテナです。このエレメントの所有者の初期値は VOB の所有者、このエレメントのグループの初期値は VOB のグループです。

追加でエレメントを作成できるのは、プライマリ グループが VOB のグループのいずれかであるプロセスだけです。エレメントを作成するには、プロセスが新しいエレメントを格納するバージョンのディレクトリ エレメントをチェックアウトする権限も持っている必要があります。詳細については、「エレメントに書き込む権限」を参照してください。

## エレメントを削除する権限

エレメントを削除できるのは、そのエレメントの所有者、VOB の所有者、権限を持つユーザーだけです。`cleartool rmelem` コマンドを使用してエレメントを削除するのと、`cleartool rmname` コマンドを使用してディレクトリのバージョンからエレメント名を削除するのでは異なります。詳細については、「エレメントに書き込む権限」を参照してください。

バージョンを削除できるのは、そのバージョンの作成者、エレメントの所有者、VOB の所有者、権限を持つユーザーです。

## エレメントを読み取る権限

プロセスのユーザーとグループ、エレメントの所有者、グループ、保護モードに基づいたアルゴリズムによって、エレメントの読み取りを許可するかどうかが決定されます。78 ページの「VOB データとビュー データへのアクセス」を参照してください。

## エレメントに書き込む権限

プロセスはエレメントに直接書き込むことはできません。エレメントを変更するには、あるバージョンのエレメントをチェックアウトし、新しいバージョンとしてチェックインします。

プロセスにバージョンのチェックアウトとチェックインを許可するかどうかを決定するときに、エレメントの保護モードは考慮されません。次のいずれかの条件が満たされていれば、プロセスでバージョンをチェックアウトできます。

- プロセスがエレメントの所有者であるユーザー ID を持っている。
- プロセスのグループ ID がエレメントのグループと一致している。
- プロセスが VOB の所有者であるユーザー ID を持っている。
- プロセスが権限を持つユーザーのユーザー ID を持っている。

次のいずれかの条件が満たされていれば、プロセスでバージョンをチェックアウトできます。

- プロセスがそのエレメントをチェックアウトしたユーザーのユーザー ID を持っている。
- プロセスがエレメントの所有者であるユーザー ID を持っている。
- プロセスのグループ ID がエレメントのグループと一致している。
- プロセスが VOB の所有者であるユーザー ID を持っている。
- プロセスが権限を持つユーザーのユーザー ID を持っている。

ディレクトリ エレメントをチェックアウトすると、エレメントの作成や削除を行うことでそのディレクトリを変更できます。**cleartool rmname** コマンドを使用してディレクトリのバージョンからエレメント名を削除するのと、エレメント自体を削除するのでは異なります。詳細については、「エレメントを削除する権限」を参照してください。

## エレメントを実行する権限

プロセスのユーザーとグループ、エレメントの所有者、グループ、保護モードに基づいたアルゴリズムによって、エレメントの実行を許可するかどうかが決まります。78 ページの「VOB データとビュー データへのアクセス」を参照してください。さらに、次の 2 つの特殊なケースでは、エレメントを実行する権限が制限されます。

- Windows の場合、**cleartool chmod** コマンドを実行して追加するまで、ファイル エレメントは実行権限を持ちません。例を次に示します。

```
cleartool chmod +x command.exe
```

実行可能プログラムをソース コントロールに追加しても、このコマンドを実行しない限り、チェックインされていてもこのプログラムを実行できません。

- UNIX の場合、**nosuid** マウント オプションを指定して VOB をマウントすると、チェックイン済みの **setuid** 実行可能ファイルを実行できません。

## その他の VOB オブジェクトに対するアクセス コントロール

VOB には、エレメントとバージョンのほかにもアクセス コントロールの対象となる次のようなオブジェクトがあります。

- メタデータ タイプ (ラベル タイプ、ブランチ タイプ、属性タイプなど)
- 統一変更管理 (UCM) オブジェクト (プロジェクト、アクティビティ、ストリームなど)
- 記憶プール
- 派生オブジェクト

一般にこれらのオブジェクトは、アクセス コントロールの重要な要素となる、次の 2 つのプロパティを持ちます。

- **所有者**: 初期値は、そのオブジェクトを作成したプロセスのユーザーです。
- **グループ**: 初期値は、そのオブジェクトを作成したプロセスのプライマリ グループです。

`cleartool describe` コマンドを実行すると、オブジェクトの所有者とグループを表示できます。オブジェクトの作成後に、オブジェクトの所有者、VOB の所有者、権限を持つユーザーは、`cleartool protect` コマンドを使用して、そのオブジェクトの所有者とグループを変更できます。オブジェクトのグループは、VOB のグループのいずれかである必要があります。

## その他の VOB オブジェクトを作成する権限

タイプまたは UCM オブジェクトは、どのユーザーでも作成できます。記憶プールを作成できるのは、VOB の所有者または権限を持つユーザーだけです。

ラベル、ブランチ、属性などのタイプのインスタンスは一般にエレメントバージョンに関連付けられています。あるタイプのインスタンスを作成するには、次のいずれかの条件が満たされている必要があります。

- プロセスがエレメントの所有者であるユーザー ID を持っている。
- プロセスのグループ ID がエレメントのグループと一致している。
- プロセスが VOB の所有者であるユーザー ID を持っている。
- プロセスが権限を持つユーザーのユーザー ID を持っている。

## その他の VOB オブジェクトを削除する権限

オブジェクトの所有者、VOB の所有者、権限を持つユーザーは、タイプまたは UCM オブジェクトを削除できます。

ラベル、ブランチ、属性などのタイプのインスタンスは一般にエレメントバージョンに関連付けられています。一般に、タイプのインスタンスを作成できる場合は、そのインスタンスを削除することもできます。「その他の VOB オブジェクトを作成する権限」を参照してください。また、ブランチ インスタンスの作成者は、そのインスタンスを削除することもできます。

## その他の VOB オブジェクトを読み取る権限

どのユーザーもタイプ、UCM オブジェクト、記憶プールに関する情報を表示できます。

## その他の VOB オブジェクトに書き込む権限

どのユーザーも UCM オブジェクトを変更できます。オブジェクトの所有者、VOB の所有者、権限を持つユーザーは、タイプを変更できます。

## VOB オブジェクトのロック

VOB オブジェクトのアクセス コントロールの目的は、長期的なアクセス コントロール メカニズムを実現することです。ClearCase では、個々の VOB オブジェクトを明示的にロックすることで、一時的なアクセス コントロールを実現することもできます。lock コマンドを使用すると、さまざまなレベルで変更を禁止または制限できます。最小レベルでは、個々のエレメント単位、またはエレメントの個々のブランチ単位のロックが可能です。最大レベルでは、1 つの VOB 全体をロックして、その VOB への変更を全面的に禁止することができます。

オブジェクトがロックされると、権限を持つユーザー、ロックを作成したユーザーを含むすべてのユーザーがそのオブジェクトを変更できなくなります (ただし、これらのユーザーはオブジェクトのアンロックを実行できます)。lock コマンドは、例外リストを受け付けます。例外リストには、ロックされていてもオブジェクトを修正できるユーザーが指定されます。

## タイプ オブジェクトのロック

タイプ オブジェクトをロックして、そのタイプのインスタンスに対する変更を禁止することができます。たとえば、次のような操作が可能です。

- main というブランチ タイプを特定グループ以外の全ユーザーに対してロックできます。これにより、このグループだけが統合作業を行ったり、すべてのエレメントの main ブランチ上でリリース関連のクリーンアップ作業を行うことができます。ほかのユーザーも、main ブランチ以外のブランチ上で作業を続けることができます。
- ラベル タイプをロックすると、誰もそのラベル タイプのインスタンスを作成または移動できなくなります。特定のリリースで使用されているすべてのバージョンにラベルを付け、そのラベルをロックすると、後でそのリリースを簡単に作成し直すことができます。

## ビューとビュー オブジェクトに対するアクセス コントロール

---

ビューは、ユーザーが VOB データにアクセスするときの媒体として機能します。VOB とその中のオブジェクトと同様に、ビューもアクセス コントロールに関係します。動的ビューでは、エレメントとバージョンの権限は、ビューおよびビュープライベート ファイル、ディレクトリと連携して、VOB データとビュー データへのアクセスを制御します。

たとえば、エレメントのバージョンをチェックアウトしてから、エレメントを変更する必要があります。エレメントは、バージョンをチェックアウトする権限を付与する必要があります。動的ビューでは、バージョンをチェックアウトするとビュープライベート ファイルが作成されます。ビューとビュープライベート ファイルを格納するディレクトリの両方で、ファイルを作成する権限を持っている必要があります。コンテナ ディレクトリは、エレメント バージョンまたはビュープライベート ディレクトリのいずれかです。

一般に、動的ビューで ClearCase データにアクセスするには、以下の一連のテストにパスするためのプロセスが必要になります。

- ビューにアクセスできる。
- コンテナ ディレクトリにアクセスできる。
- エレメントにアクセスできる。

スナップショット ビューでは、そのスナップショット ビュー ディレクトリに対するファイルシステム固有の権限に基づいて、エレメント バージョンのコピーを含む、スナップショット ビュー内のファイルやディレクトリへのアクセス権限が設定されます。スナップショット ビュー内のエレメントを作成、削除、変更するには、各エレメントに対して適切なアクセス権限を持つプロセスが必要です。

**メモ:** UNIX ホストの場合、ClearCase はリモート root ユーザーからのビュー アクセス要求をユーザー nobody.nobody からの要求として処理します。詳細については、75 ページの「リモート root ユーザーの制限付き権限」を参照してください。

## 動的ビューに対するアクセス コントロール

動的ビューには、アクセス コントロールの重要な要素となる以下のプロパティがあります。

- **所有者:** 初期値は、そのビューを作成したプロセスのユーザーです。
- **グループ:** 初期値は、そのビューを作成したプロセスのプライマリ グループです。
- **保護モード:** ビューの保護モードの初期値は、UNIX ホストと Windows ホストで異なります。
  - Windows の場合、ビューは初期値で所有者とグループに対して読み取り、書き込み、実行権限を付与し、その他に対しては読み取りと実行権限を付与します。Windows エクスプローラまたは ClearCase エクスプローラの [エレメントのプロパティ] ダイアログボックスを使用すると、ビューの所有者、グループ、保護モードを表示できます。ビューの作成後に、所有者とグループを変更することはできません。chview コマンドを使用すると、保護モードを「読み取り / 書き込み」または「読み取り専用」に変更できます。
  - UNIX の場合、保護モードの初期値は、そのビューを作成したユーザーの umask によって異なります。umask は UNIX のコマンドで、ユーザーがファイルを作成したときに、一部の権限を許可しないことを指定します。詳細については、UNIX umask のリファレンス ページを参照してください。ユーザーがビューを作成すると、ClearCase は読み取り、書き込み、実行権限をすべてのユーザーに付与してから、ユーザーの umask の指定に従って権限を削除します。たとえば、ユーザーの umask が 002 の場合、ClearCase はほかのユーザーの書き込み権限を削除します。

ビュー記憶ディレクトリを選択して、ビューの所有者、グループ、保護モードを表示するには、次のように -properties と -full オプションを指定して、cleartool lsview コマンドを使用します。

```
cleartool lsview -properties -full v5_integration
* v5_integration          /net/mars/viewstg/v5_integration.vws
.
.
.
所有者 : akp:              rwx (すべて)
グループ : clearusers: rwx (すべて)
その他 ::                  r-x (読み取り)
```

ビュー記憶ディレクトリを表示するには、次のように UNIX `ls` コマンドを使用します。

```
cd /net/mars/viewstg
ls -ld v5_integration.vws
drwxrwxr-x  6 akp  clearusers  512  Nov 10 11:29
v5_integration.vws
```

このコマンドの出力に、ビューの所有者 (akp)、グループ (clearusers)、保護モード (drwxrwxr-x) が表示されます。一部の UNIX システムでは、`ls -g` を使用してグループを表示する必要があります。

## ビューを作成する権限

すべてのユーザーがビューを作成できます。

## ビューを削除する権限

ビューの所有者または権限を持つユーザーがビューを削除できます。

## ビューを読み取る権限

ファイルやディレクトリを読み取るには、動的ビューとビュー内のファイルまたはディレクトリの両方の読み取り権限がプロセスに必要です。ファイル エlement およびディレクトリ Element のバージョンを読み取るには、その Element の読み取り権限がプロセスに必要です。82 ページの「Element を読み取る権限」を参照してください。ビュープライベート ファイルまたはディレクトリを読み取るには、そのビュープライベート ファイルまたはディレクトリの読み取り権限がプロセスに必要です。90 ページの「ビュープライベート ファイルを読み取る権限」を参照してください。

ClearCase では、プロセスのユーザーとグループ、ビューの所有者、グループ、保護モードに基づいたアルゴリズムによって、ビューの読み取り権限を許可するかどうかが決まります。78 ページの「VOB データとビュー データへのアクセス」を参照してください。

## ビューを書き込む権限

構成仕様の設定など、ビュー自体を変更する操作を実行するには、ビューの書き込み権限がプロセスに必要です。

コンテナ ディレクトリ内のファイルやディレクトリを作成または削除するには、動的ビューとビュー内のコンテナ ディレクトリの両方の書き込み権限がプロセスに必要です。コンテナ ディレクトリがエレメント バージョンの場合、そのエレメントの書き込み権限がプロセスに必要です。82 ページの「エレメントに書き込む権限」を参照してください。コンテナ ディレクトリがビュープライベート ディレクトリの場合、そのビュープライベート ディレクトリの書き込み権限がプロセスに必要です。91 ページの「ビュープライベート ファイルを書き込む権限」を参照してください。

ClearCase では、プロセスのユーザーとグループ、ビューの所有者、グループ、保護モードに基づいたアルゴリズムによって、ビューの読み取り権限を許可するかどうかが決まります。78 ページの「VOB データとビュー データへのアクセス」を参照してください。

## ビューを実行する権限

ファイルやディレクトリを実行するには、動的ビューとビュー内のファイルまたはディレクトリの両方の実行権限がプロセスに必要です。ファイル エレメントまたはディレクトリ エレメントのバージョンを実行するには、そのエレメントの実行権限がプロセスに必要です。83 ページの「エレメントを実行する権限」を参照してください。ビュープライベート ファイルまたはディレクトリを実行するには、そのビュープライベート ファイルまたはディレクトリの実行権限がプロセスに必要です。91 ページの「ビュープライベート ファイルを実行する権限」を参照してください。

ClearCase では、プロセスのユーザーとグループ、ビューの所有者、グループ、保護モードに基づいたアルゴリズムによって、ビューの実行権限を許可するかどうかが決まります。78 ページの「VOB データとビュー データへのアクセス」を参照してください。

## ビュープライベート ファイルに対するアクセス コントロール

この項では、動的ビュー内のビュープライベート ファイルに対するアクセス コントロールについて説明します。スナップショット ビューでは、スナップショット ビュー ディレクトリ内のディレクトリやファイルに対するファイル システム固有の権限に基づいて、各ディレクトリとファイルへのアクセス権限が設定されます。

動的ビューでは、ビュープライベート ファイルの所有者、グループ、保護モードの初期値は、UNIX と Windows で異なります。

## 所有者、グループ、保護モードの初期値 (UNIX)

UNIX の場合、ビュープライベート ファイルの所有者、グループ、保護モードの初期値は、次の規則に従って決定されます。

- **所有者**: 初期値は、そのファイルまたはディレクトリを作成したプロセスのユーザーです。
- **グループ**: 初期値は、そのファイルまたはディレクトリを作成したプロセスのプライマリグループです。
- **保護モード**: ビュープライベート ファイルの保護モードの初期値は、そのファイルまたはディレクトリを作成したユーザーの **umask** によって異なります。**umask** は UNIX のコマンドで、ユーザーがファイルを作成したときに、一部の権限を許可しないことを指定します。詳細については、UNIX **umask** のリファレンス ページを参照してください。ユーザーがビュープライベート ファイルまたはディレクトリを作成すると、**ClearCase** はファイルやディレクトリの作成方法に応じた権限を初期値として付与します。次に **ClearCase** は、ユーザーの **umask** によって指定された権限を削除します。たとえば、ユーザーの **umask** が 002 の場合、**ClearCase** はその他のユーザーの書き込み権限を削除します。

**cleartool describe** コマンドまたは UNIX **ls** コマンドを使用すると、ビュープライベート ファイルまたはディレクトリの所有者、グループ、保護モードを表示できます。UNIX **chown** コマンドを使用すると、所有者を変更できます。同様に **chgrp** コマンドを使用するとグループを、**chmod** コマンドを使用すると保護モードを変更できます。

## 所有者、グループ、保護モードの初期値 (Windows)

Windows の場合、ビュープライベート ファイルの所有者、グループ、保護モードの初期値は、次の規則に従って決定されます。

- **所有者**: 初期値は、そのファイルまたはディレクトリを作成したプロセスのユーザーです。
- **グループ**: 初期値は、そのファイルまたはディレクトリを作成したプロセスのグループに基づいて、次のいずれかの方法で割り当てられます。
  - プロセスのプライマリ グループが VOB のグループと同じ場合、そのグループが割り当てられます。
  - それ以外の場合は、プロセスのグループ リストが VOB の補足グループ リストと比較され、両方のリストに含まれている最初のグループが割り当てられます。
- **保護モード**: ビュープライベート ファイルまたはディレクトリは、すべてのユーザーに対して初期値として読み取り、書き込み、実行権限を設定します。

**cleartool describe** コマンド、または **ClearCase** エクスプローラまたは Windows エクスプローラの [ファイルのプロパティ] または [ディレクトリのプロパティ] ダイアログ ボックスを使用すると、ビュープライベート ファイルまたはディレクトリの所有者、グループ、保護モードを表示できます。

ビュープライベート ファイルまたはディレクトリの所有者とグループは変更できません。  
Windows エクスプローラから [プロパティ] ダイアログ ボックスを開いて、[読み取り専用] チェック ボックスをオンにするか、**attrib +R** (モード 777 と同等) と **attrib -R** (モード 555 と同等) コマンドを使用すると、すべてのユーザーに書き込み権限を設定するかどうかを指定できます。これ以外の権限は変更できません。

## ビュープライベート ファイルを作成する権限

コンテナ ディレクトリにファイルまたはディレクトリを作成するには、ビューとビュー内のコンテナ ディレクトリの両方の書き込み権限がプロセスに必要です。ビュー権限については、88 ページの「ビューを書き込む権限」を参照してください。

コンテナ ディレクトリがエレメント バージョンの場合、そのエレメントの書き込み権限がプロセスに必要です。82 ページの「エレメントに書き込む権限」を参照してください。コンテナ ディレクトリがビュープライベート ディレクトリの場合、そのビュープライベート ディレクトリの書き込み権限がプロセスに必要です。詳細については、「ビュープライベート ファイルを書き込む権限」を参照してください。

## ビュープライベート ファイルを削除する権限

コンテナ ディレクトリ内のファイルまたはディレクトリを削除するには、ビューとビュー内のコンテナ ディレクトリの両方の書き込み権限がプロセスに必要です。ビュー権限については、88 ページの「ビューを書き込む権限」を参照してください。

コンテナ ディレクトリがエレメント バージョンの場合、そのエレメントの書き込み権限がプロセスに必要です。82 ページの「エレメントに書き込む権限」を参照してください。コンテナ ディレクトリがビュープライベート ディレクトリの場合、そのビュープライベート ディレクトリの書き込み権限がプロセスに必要です。詳細については、「ビュープライベート ファイルを書き込む権限」を参照してください。

## ビュープライベート ファイルを読み取る権限

ファイルまたはディレクトリを読み取るには、ビューとビュー内のビュープライベート ファイルまたはディレクトリの両方の読み取り権限がプロセスに必要です。ビュー権限については、88 ページの「ビューを書き込む権限」を参照してください。

ClearCase では、プロセスのユーザーとグループ、ビュープライベート ファイルまたはディレクトリの所有者、グループ、保護モードに基づいたアルゴリズムによって、ファイルまたはディレクトリの読み取り権限を許可するかどうかが決まります。78 ページの「VOB データとビュー データへのアクセス」を参照してください。

## ビュープライベート ファイルを書き込む権限

ファイルまたはディレクトリを書き込むには、ビューとビュー内のビュープライベート ファイルまたはディレクトリの両方の書き込み権限がプロセスに必要です。ビュー権限については、88 ページの「ビューを書き込む権限」を参照してください。

ClearCase では、プロセスのユーザーとグループ、ビュープライベート ファイルまたはディレクトリの所有者、グループ、保護モードに基づいたアルゴリズムによって、ファイルまたはディレクトリの書き込み権限を許可するかどうかが決定されます。78 ページの「VOB データとビュー データへのアクセス」を参照してください。

## ビュープライベート ファイルを実行する権限

ファイルまたはディレクトリを実行するには、ビューとビュー内のビュープライベート ファイルまたはディレクトリの両方の実行権限がプロセスに必要です。ビュー権限については、88 ページの「ビューを書き込む権限」を参照してください。

ClearCase では、プロセスのユーザーとグループ、ビュープライベート ファイルまたはディレクトリの所有者、グループ、保護モードに基づいたアルゴリズムによって、ファイルまたはディレクトリの実行権限を許可するかどうか決定されます。78 ページの「VOB データとビュー データへのアクセス」を参照してください。

## 派生オブジェクトに対するアクセス コントロール

派生オブジェクト (DO) は、`clearmake` や `omake` コマンド、または `clearaudit` セッションによって、動的ビューに作成されるファイルです。これらのコマンドを実行しても、スナップショット ビューでは派生オブジェクトは作成されません。

派生オブジェクトは、初期段階ではビュープライベート ファイルと同等の役割を果たします。DO を作成するには、ビュープライベート ファイルを作成するのと同じ権限が必要です。DO の所有者、グループ、保護モードの初期値は、ビュープライベートファイルと同じ方法で決定されます。88 ページの「ビュープライベート ファイルに対するアクセス コントロール」を参照してください。

共有可能派生オブジェクトは、DO をウイंकインすることによって、ほかの動的ビューで使えるようになります。共有可能 DO が初めてウイंकインされると、作成元のビューからプロモートされ、それを含む VOB 内のオブジェクトになります。この VOB オブジェクトが共有 DO です。

共有 DO には、所有者、グループ、保護モードがあります。共有 DO の所有者とグループの初期値は、その DO が VOB にプロモートされた段階での所有者とグループになります。共有可能 DO のグループは、VOB にプロモートされる DO のいずれかの VOB グループである必要があります。

共有 DO の所有者、VOB 所有者、権限を持つユーザーは、**cleartool protect** コマンドを使用して、DO の所有者、グループ、保護モードを変更できます。

共有 DO を動的ビューにウイंकインするプロセスには、その DO の読み取り権限が必要です。**ClearCase** では、プロセスのユーザーとグループ、DO の所有者、グループ、保護モードに基づいたアルゴリズムによって、DO の読み取り権限を許可するかどうかが決まります。78 ページの「VOB データとビュー データへのアクセス」を参照してください。

ウイंकインした DO には、共有 DO と同じ所有者、グループ、保護モードが設定されます。権限の変更や書き込みなど、ウイंकインした DO をビューで変更する場合、**ClearCase** は DO をビュープライベート コピーに変換します。

## ClearCase とファイル システム固有の権限

---

**ClearCase** では、VOB とビューのデータは、各 VOB とビュー ホストの固有のファイル システム内の、通常のディレクトリ、VOB 記憶ディレクトリ、ビュー記憶ディレクトリで管理されます。各ディレクトリへのアクセスはすべて **ClearCase** が管理しているため、ユーザーがこれらのディレクトリを直接読み取り/書き込みすることはできません。ファイル システム保護は、VOB オブジェクトへのアクセス権限を持たないユーザーが、オブジェクトのデータが格納されているファイルにアクセスできないように管理されます。

**警告:** VOB 記憶ディレクトリやビュー記憶ディレクトリに関連する、ファイル システム固有の権限は変更しないでください。

VOB および ビュー記憶領域ディレクトリの権限を誤って変更してしまった場合、VOB とビューへのアクセスは厳しく制限されますが、損傷を修復することはできます。269 ページの「NTFS の記憶ディレクトリ ACL の修復」を参照してください。

スナップショット ビュー ディレクトリは、ファイル システム固有のディレクトリです。このディレクトリには、エレメント バージョンのコピーだけでなく、**ClearCase** で管理されていないその他のファイル システム オブジェクトが格納されます。スナップショット ビューの所有者は、これらのファイルに関して、ファイル システム固有の権限を自分で管理できます。たとえば、ビューの所有者は、**ClearCase** で管理されていないスナップショット ビュー ディレクトリ内のファイルに、グループの書き込み権限を追加したり削除したりできます。

動的ビューと同様に、スナップショット ビューにもビュー記憶ディレクトリがあります。このビュー記憶ディレクトリは、スナップショット ビュー ディレクトリの内部にある場合と外部にある場合があります。**ClearCase** は、動的ビューと同様に、スナップショット ビューのビュー記憶ディレクトリの作成と管理を行います。スナップショット ビューのビュー記憶ディレクトリに関しては、ファイル システム固有の権限を変更しないでください。

この章では、バージョン付きオブジェクト ベース (VOB) の操作の詳細、記憶レイアウト、データ モデルについて説明します。VOB の詳細については、`mkvob` のリファレンス ページを参照してください。

## VOB と VOB 管理の概要

---

ClearCase の開発環境には 1 つまたは複数の VOB が必要です。VOB は ClearCase の制御下にあるソフトウェア成果物の主要なリポジトリであるため、ClearCase 管理者にとって VOB 管理は非常に重要なタスクです。

VOB 管理タスクには次のものがあります。

- VOB ホストの構成
- VOB の作成
- VOB データへのアクセスの有効化
- 別の構成管理システムから VOB へのデータのインポート
- VOB データのバックアップと回復
- VOB 記憶の管理
- VOB の一貫性の監視

## VOB データとメタデータへのアクセス

VOB は、データ (ファイルやディレクトリなどのバージョン付き成果物) とメタデータ (ブランチ、ラベル、イベント レコードなど) のリポジトリです。VOB データは、ビューからアクセスできるようにする必要があります。ほとんどの VOB メタデータは、ClearCase の GUI と `cleartool` コマンドを使用してアクセスできます。

VOB データはユーザーが作成するか、別の構成管理システムからインポートできます。VOB データの作成、インポート、アクセスを行うには、ビューを使用する必要があります。また、VOB と VOB のオブジェクトへのアクセス権限が必要です。VOB とビューのアクセス コントロールの詳細については、第 5 章を参照してください。ビューの詳細については、第 9 章を参照してください。

VOB メタデータは、次のように作成されます。

- **VOB** または **VOB** のオブジェクトを作成するコマンドの実行結果として間接的に生成される。たとえば、**VOB** を作成するコマンドを実行すると、**VOB** 属性と **UUID** も作成されます。**cleartool mkbranch** コマンドを実行すると、必要に応じて、ブランチとブランチ タイプのローカル インスタンスが作成されます。
- 特定タイプのメタデータを作成または処理するコマンドの実行結果として生成される。たとえば、**cleartool mkobjecttype** コマンド (**mkeltype** や **mkbrtype** など) を実行すると、第 7 章で説明されているような特定タイプのオブジェクトが作成されます。

**cleartool describe** コマンド (またはこれに相当する GUI) と、**lstype** や **lsvob** などのコマンドを使用して、**VOB** メタデータを表示したり、編集したりすることができます。

## VOB 属性

すべての **VOB** には、ディスク上に同じディレクトリ構造があり、ディレクトリ構造に類似した管理要件があります。ただし、特定の **VOB** では、**VOB** の作成時に **VOB** オブジェクトに関連付けられた属性によって決定される特別な機能を実行できます。

- **UCM VOB** (**sumvob** と表示される場合もある) 属性を持つ **VOB** には、**UCM** プロジェクトの一部であるコンポーネントが保存されます。
- **プロジェクト VOB** 属性を持つ **VOB** には、プロジェクト、ベースライン、フォルダ、コンポーネントなどの **UCM** 成果物に関するメタデータが保存されます。

これらの両方の属性を持つ **VOB** には、**UCM** プロジェクトのメタデータと **UCM** コンポーネントが保存されます。どちらの属性も持たない **VOB** は、通常 **VOB** です。通常 **VOB** と **UCM** コンポーネント **VOB** は、バージョン付き成果物の永久リポジトリです。これらの **VOB** は、どの **ClearCase** コミュニティにおいても、複数の **VOB** グループから構成されます。

どの **VOB** も管理 **VOB** 階層の一部になり得ます。管理 **VOB** 階層の詳細については、第 7 章を参照してください。

小規模な **ClearCase** コミュニティでは、1 つの **VOB** が管理 **VOB** と **PVOB** であると同時に、通常 **VOB** または **UCM** コンポーネント **VOB** であることがあります。多くの場合、1 つのコミュニティには 1 つまたは複数の管理 **VOB** と多数の通常 **VOB** が必要です。また、**UCM** を使用している場合は、1 つの **PVOB** と多数のコンポーネント **VOB** が必要です。大規模なコミュニティでは、複数の管理 **VOB** または **PVOB** を必要とする場合があります。

## VOB スキーマ バージョン

各 VOB には、VOB データベースの形式を示し、VOB に保存できるデータとメタデータのタイプを特定するデータベース スキーマ バージョンが含まれています。ClearCase では、次の 2 つの VOB スキーマ バージョンをサポートしています。

- スキーマ バージョン 53: ClearCase LT for Windows 以外のすべてのプラットフォームでサポートされます。
- スキーマ バージョン 54: すべての Windows プラットフォームと多くの UNIX プラットフォームでサポートされます。スキーマ バージョン 54 は、Windows セキュリティ識別子をサポートしており、Microsoft Active Directory 環境で ClearCase を使用する場合に必要です。スキーマ バージョン 54 は、2 GB を超える VOB データベース ファイルもサポートします。

ホスト上のすべての VOB は、同じスキーマ バージョンで形式を設定する必要があります。ClearCase サーバー ソフトウェアをホストにインストールするときに、そのホストで作成される VOB で使用するスキーマ バージョンを選択します。

### VOB スキーマ バージョンを変更するには

既存のスキーマ バージョン 53 VOB をスキーマ バージョン 54 にアップグレードするには

- 1 ClearCase を VOB ホストに再インストールし、ホストが VOB スキーマ バージョン 54 をサポートするように指定します (一部の UNIX プラットフォームでは、このスキーマ バージョンをサポートしていない場合があります)。
- 2 再インストールが完了したら、ホスト上のすべての VOB がアクセス可能になるように、`cleartool reformatvob` コマンドを実行して VOB を形式変更します。

スキーマ バージョン 54 をサポートしているホストにスキーマ バージョン 53 VOB を移動すると、形式変更するまで VOB にアクセスできなくなります。

**メモ:** VOB を下位のスキーマ バージョンに形式変更することはできません。

## VOB 機能レベル

機能レベルは、VOB がサポートする ClearCase 機能セットを定義するもので、整数で表されます。ClearCase の新しいリリースで VOB データベースでのサポートを必要とする新機能が導入された場合は、VOB の機能レベルを上げないと、クライアントはその VOB のデータにアクセスするときに新機能を使用できません。機能レベルの主な目的は、ClearCase の別のリリースを実行するサーバー ホストに (ClearCase MultiSite 環境で) VOB のレプリカを作成した場合に、ClearCase リリースのアップグレードの管理に使用することにあります。

ClearCase の各リリースは、機能レベルに関連付けられています。現在のリリースでサポートされている機能レベルの詳細については、Rational ClearCase と ClearCase MultiSite のリリースノートを参照してください。

## VOB 機能レベルを表示するには

レプリカの機能レベルを表示するには、Windows エクスプローラまたは ClearCase 管理コンソールを使用して、VOB のプロパティを表示します。機能レベルは、[カスタム] タブに表示されます。cleartool コマンド行を使用して表示することもできます。

レプリカ ファミリの機能レベルを表示するには、cleartool describe replica:replica-name@vob-tag コマンドを使用します。例を次に示します。

```
cleartool describe replica:tokyo@¥dev
replica "tokyo"
  created 20-Aug-02.13:35:37 by akp (akp.user@boron)
...
  機能レベル : 2
...
```

VOB ファミリの機能レベルを表示するには、cleartool describe vob:vob-tag コマンドを使用します。例を次に示します。

```
cleartool describe vob:/vobs/dev
versioned object base "/vobs/dev"
  created 15-Aug-01.14:19:03 by jjp (jjp.user@mars)
...
  VOB ファミリ機能レベル : 2
...
```

**メモ:** 新規に作成したレプリカの機能レベルを設定するまで、機能レベルの値は**不明**に設定されています。たとえば、describe コマンドを使用して新しいレプリカのプロパティを表示すると、その出力は次のようになります。

```
cleartool describe replica:sanfran_hub@/vobs/dev
...
  機能レベル : 不明
```

## VOB 機能レベルを変更するには

VOB の機能レベルを変更するには、chflevel コマンドを使用します。レプリカのない VOB の機能レベルを上げるには、次の手順に従います。

- 1 VOB 所有者または権限を持つユーザーとして、VOB ホストにログオンします。
- 2 chflevel コマンドに -auto オプションを指定して実行します。出力に、該当のホストで処理される VOB が一覧表示されます。レプリカのない VOB の機能レベルが、ホストにインストールされた ClearCase リリースでサポートされている最高機能レベルでない場合は、ここで各 VOB の機能レベルを上げることができます。

MultiSite を使用すると、VOB ファミリ内の各レプリカに機能レベルが設定されます。同じファミリ内のレプリカに、それぞれ異なる機能レベルを設定できます。VOB ファミリ自体の機能レベルは、その VOB ファミリ メンバの最低レプリカ機能レベル以下に設定する必要があります。VOB ファミリの機能レベルを、ファミリ内のレプリカの最低機能レベルより高く設定することはできません。機能レベルと VOB レプリカの詳細については、『Rational ClearCase MultiSite 管理ガイド』を参照してください。

## VOB サーバー ホスト構成のガイドライン

---

`vob_server` プロセスを実行するホストを、VOB ホストと呼びます。ClearCase LT コミュニティには、1 つの VOB ホストから成る ClearCase LT サーバーがあります。ClearCase コミュニティに複数の VOB ホストを構成することもできます。適切な VOB ホストの選択と構成は、満足のいく ClearCase パフォーマンスを得るためにはきわめて重要です。

VOB ホストでは、ホストがサポートしている各 VOB に対して多数のサーバー プロセスが実行されます (8 ページの「ClearCase サーバー プロセス」を参照)。アクティブな VOB ホストは、数十から数百のプロセスをサポートし、多数のクライアントによるネットワーク アクセスに対応する必要があります。このため、ホストの物理メモリ、記憶領域、プロセッサ、ネットワーク インターフェイス ハードウェアを指定する場合は、この項に記述されているガイドラインに従って行うことをお勧めします。また、パフォーマンスを調整する手順については、第 16 章を参照してください。

**メモ:** Windows VOB ホストでは、オペレーティング システムのワークステーション版ではなく、サーバー版 (Windows 2000 Server など) を実行する必要があります。オペレーティング システムのワークステーション版は、クライアントの同時接続数に制限があり、ほとんどの VOB ホストアプリケーションに適していません。

### 物理メモリ

メイン メモリの推奨最小サイズは、128 MB か、ホストがサポートするすべての VOB データベース サイズの半分のいずれか大きいほうです。この容量に次のサイズを加算します。

- VOB データベースのサイズに関係なく、各 VOB につき 7 MB のメモリ
- VOB ホスト上で実行する `view_server` プロセス用に 750 KB のメモリ

適切な物理メモリ容量は、VOB のパフォーマンスにとって最も重要な要素です。VOB のアクセスを高速にしたり、パフォーマンスを低下させずに同時並行ユーザー数を増やしたりするのに最も簡単で最もコスト効果の高い方法は、VOB ホストのメイン メモリを増設することです。

**メモ:** VOB ホスト上で `view_server` プロセスを実行することはお勧めしません (305 ページの「プロセス オーバーヘッドの最小化」を参照)。

## ディスク容量

VOB データベース (Windows の場合は VOB 記憶ディレクトリ全体) は、1 つのディスク パーティション内に収める必要があります。VOB データベースは、開発が進んでプロジェクトが完成に近づくにつれて、サイズが非常に大きくなる傾向があります。VOB のサイズがどの程度大きくなるかを試算する方法はありませんが、付録 F に VOB サーバーのディスク容量の要件を確立するためのガイドラインについて説明しています。

ディスク サブシステムは、回転速度が速く、シーク時間が短く、平均故障間隔の長い、高性能なものをお勧めします。できれば、ディスクのストライピングとミラーリングを利用できる RAID などのシステムを使用してください。ミラーリングを行うと、使用時のパフォーマンスはわずかに低下しますがデータのバックアップに対して有効です。ストライピングは、全般的なパフォーマンス向上に役立ち、ミラーリングによるパフォーマンス低下を十分に補うことができます。詳細については、306 ページの「ディスク パフォーマンスの最大化」を参照してください。

## プロセッサの処理能力

VOB ホストには十分な CPU 処理能力が必要です。ここで言う十分な処理能力の定義は、ハードウェア アーキテクチャによって異なります。ClearCase や同様の業務用アプリケーションでは、サーバーの CPU 処理能力はクライアントの操作パフォーマンスを左右する重要な要素です。ClearCase のクライアント ツールやビューなどの重要でないプロセスを VOB ホストから切り離して、サーバーの CPU サイクルの大半を有効活用できるようにします。

## ネットワークの接続性

VOB へのアクセスはほとんどの場合 VOB ホストのネットワーク インターフェイスに負荷がかかるため、VOB ホストへの接続には高帯域幅 (100 MB/秒以上) のネットワーク接続を使用する必要があります。VOB ホストへの接続に複数のネットワーク インターフェイスを使用すると、ネットワークへのアクセス可能性は格段に向上しますが、一部のオペレーティング システムでは ClearCase レジストリ 内の複数のリージョンを順番に使用する必要があるため、接続ごとに別のホスト名が必要になります (第 3 章を参照)。

## UNIX カーネル リソースの調整

UNIX を実行中の VOB ホストでは、次のようなカーネル リソースの調整が必要になる場合があります。

- **包括的プロセス テーブル:** オペレーティング システムのプロセス テーブルは、96 以上の同時並行ユーザー プロセスをサポートする必要があります。3 つまたは 4 つ以上の VOB が存在するホストの場合は、プロセス テーブルのサイズを 128 以上にします。
- **包括的ファイル記述テーブル:** オペレーティング システムのファイル記述テーブルのサイズは、700 以上に設定する必要があります。3 つまたは 4 つ以上の VOB をホストに配置する場合は、ファイル記述テーブルのサイズを大きくします。

ClearCase を起動後しばらく実行してから、VOB ホストのカーネル リソースを再調整すると、効果が得られる場合があります。詳細については、第 16 章を参照してください。

## VOB の作成

---

サーバー ホストへの適切なアクセス権限を持つユーザーは、VOB を作成できます。多くのコミュニティでは、管理作業を必要とせずに、臨時の使用目的 (テストや、小人数のみでアクセスする成果物の保存など) で独自の VOB を作成できます。

VOB に重要なコンポーネントやその他の成果物を保存して、コミュニティのほとんどのメンバーが使用する場合は、通常、管理者が作成、管理します。このような VOB の一貫性と可用性はコミュニティの作業にとって重要であるため、さらに詳細な計画と管理サポートが必要になる場合があります。

### VOB 記憶領域へのアクセス要件

VOB 記憶ディレクトリは、ディレクトリに VOB を作成する権限を持つユーザーの書き込みができるようにする必要があります。また、動的ビューからアクセスする VOB の記憶ディレクトリは、ネットワーク経由でアクセスできるようにする必要があります。

- UNIX ホストでは、VOB 記憶場所が作成されているパーティションをエクスポートして、別の UNIX クライアントがマウントできるようにする必要があります。SMB サーバー製品を使用して Windows クライアントから VOB にアクセスできるようにする場合は、SMB サーバーを付録 A で説明されているように構成する必要があります。
- Windows ホストでは、ディレクトリ (フォルダ) を共有できるようにする必要があります。新しく作成した共有には、アクセス制限がほとんどありません。1 つまたは複数の VOB 記憶ディレクトリを持つ共有の ACL を変更する場合は、VOB またはビューにアクセスするすべてのユーザーの完全なアクセス権限を維持する必要があります。また、ClearCase 管理者グループに対しても、完全なアクセス権限を付与する必要があります。

動的ビューからアクセスしない VOB の場合は、エクスポートまたは共有しないパーティション上に VOB を作成できます。

### VOB のサーバー記憶場所の作成

管理者は、サーバー記憶場所を使用して、特定のホストとディスク ボリュームを VOB の優先的な記憶場所に指定できます。サーバー記憶場所は、レジストリ リージョンに固有のものです。これは、グローバル パスを使用してアクセスする必要があります。定義済みの各リージョンには、1 つまたは複数の VOB 記憶場所を指定できます。VOB のサーバー記憶場所を作成した後は、`mkvob -stgloc` コマンドと VOB 作成ウィザードで、この記憶場所を使用できます (ビューのサーバー記憶場所も、同様の手順で作成できます)。

VOB のサーバー記憶場所は、VOB データベースの増加に対応できる十分な容量のディスクパーティション上に作成し、リージョン内のすべてのホストからアクセスできるようにする必要があります。サポートされているネットワーク接続記憶デバイスを使用する VOB ホストは、VOB のサーバー記憶場所として最適です。

VOB のサーバー記憶場所を作成するには、次のいくつかの方法があります。

- Windows の場合は、コンピュータに ClearCase サーバー ソフトウェアをインストールし、そのサーバー上に記憶場所を作成するよう指定すると、インストール後に再起動してログオンしたときに、サーバー記憶構成ウィザードが起動されます。このウィザードを使用して、該当のコンピュータ上に記憶場所を作成できます。
- UNIX または Windows の場合は、`cleartool mkstgloc` コマンドを実行して、サーバー記憶場所を作成できます。
- ClearCase LT の場合は、サーバー セットアップの一環として、サーバー セットアップ ウィザードでサーバー記憶場所を作成します。

ClearCase 管理コンソールを使用して、サーバー記憶場所を追加、変更、削除することもできます。

- 1 ClearCase 管理コンソールを起動します。
- 2 [ClearCase レジストリ] ノードの [記憶場所] サブノードに移動して、次のいずれかの操作を行います。
  - 新しい記憶場所を作成するには、[操作] メニューの [新規作成] をポイントし、[サーバー記憶場所] をクリックします。
  - 既存の記憶場所のプロパティを変更するには、対象の記憶場所を選択し、[操作] メニューの [プロパティ] をクリックします。
  - 既存の記憶場所を削除するには、リストから対象の記憶場所を選択し、[操作] メニューの [サーバー記憶場所の削除] をクリックします。

**メモ:** 動的ビューから VOB のサーバー記憶場所にアクセスする場合は、適切なネットワークアクセスが必要です。99 ページの「VOB 記憶領域へのアクセス要件」を参照してください。

## NAS デバイス上でのサーバー記憶場所の作成

サーバー記憶場所を NAS デバイス上に作成すると、デバイスの記憶領域を簡単に利用できるようになります。次の例では、`cleartool mkstgloc` コマンドを実行して、`ccnasvobstg` という名前の VOB 記憶場所を UNIX リージョンに作成します。

```
cleartool mkstgloc -vob -host vobsvr1 -gpath /net/nasdevice/vobstg/nasvobstg ¥  
-hpath /net/nasdevice/vobstg/nasvobstg ccnasvobstg ¥  
/net/nasdevice/vobstg/nasvobstg
```

ビューのサーバー記憶場所を作成するには、`cleartool mkstgloc -view` コマンドに同様のオプションを指定して実行します。

NAS デバイス上のサーバー記憶場所にある VOB またはビューをサポートする ClearCase プロセスは、`mkstgloc` コマンドの `-host` オプションで指定したホスト上で実行されます。NAS ベースのサーバー記憶場所に作成する VOB またはビューにアクセスするためのサービス要求に対応する構成のホストを選択します。

**メモ:** サーバー記憶構成ウィザードでは、NAS デバイス上に記憶場所を作成することはできません。

## VOB を VOB サーバー ホスト上に作成するには

コマンド行または GUI ツールを使用して VOB を作成できます。GUI ツールでは、完全な手順ガイドと統合されたヘルプを利用できます。コマンド行を使用する場合は、`mkvob` コマンドのリファレンス ページを参照してください。

- 1 VOB 記憶ディレクトリの場所を選択します。既存の VOB サーバー記憶場所を使用するか、新しいサーバー記憶場所を作成できます。また、99 ページの「VOB 記憶領域へのアクセス要件」に説明されている特性を持つ別のディレクトリを VOB 記憶領域に使用することもできます。
- 2 VOB ホストにログオンします。サポートされている NAS デバイス上に VOB 記憶領域を作成する場合以外は、VOB ホストは手順 1 で選択した VOB 記憶ディレクトリがあるホストです。

VOB ホストにログオンするときは、VOB へのアクセスが必要なほかのすべてのユーザーと同じプライマリ グループのメンバーとしてログオンすることをお勧めします。VOB を作成するユーザーの ID は、VOB のアクセス権限を初期化するのに使用されます。VOB へのアクセスが必要なユーザーのプライマリ グループとは別のプライマリ グループに所属するユーザーが VOB を作成する場合は、VOB の補足グループ リストを編集して、該当のユーザーが VOB データにアクセスできるようにする必要があります。詳細については、107 ページの「VOB のグループ所有権情報の調整」を参照してください。

**メモ:** 同じタイプ (Windows または UNIX) のリモート ホストでも VOB を作成できますが、ローカル記憶パスを派生するプロセスを簡素化するために、VOB ホスト上で VOB コマンドを実行することをお勧めします。

- 3 VOB タグを選択します。VOB タグは ClearCase レジストリ リージョン内のすべてのホストが VOB を参照するのに使用する一意の名前です (35 ページの「タグ」を参照)。タグ名には、VOB の内容が用途を表す名前を選択することをお勧めします。
- 4 VOB を作成します。次の例では、`cleartool` コマンド行を使用して VOB を作成する方法を説明します。各種の ClearCase GUI を使用して VOB を作成することもできます。各 GUI では、それぞれの手順ガイドを利用できます。

次のコマンドを Windows ホスト `pluto` 上で実行すると、VOB と VOB タグ `¥lib` が作成されます。このタグの記憶領域は、`¥¥pluto¥vobstg` という名前の共有ディレクトリ内に保存されています。

```
cleartool mkvob -co "library sources" -tag ¥lib ¥¥pluto¥vobstg¥lib.vbs
```

ホスト上のローカル パス: c:¥vobstg¥lib.vbs

グローバル パス: ¥¥pluto¥vobstg¥lib.vbs

VOB の所有権:

所有者 vobadm

グループ dev

コメントは新しい VOB のデータベースに保存され、`cleartool describe` コマンドと各種 GUI によって表示できます。

VOB を作成する際に必要なオプションの詳細については、102 ページの「VOB を NAS デバイス上に作成するには」、103 ページの「VOB を リモート Windows ホスト上に作成するには」、103 ページの「パブリック VOB とプライベート VOB」を参照してください。

- 5 VOB がリージョン内の一般のホストからアクセスできるかどうかを確認します。VOB を初めて作成した場合は、VOB にはルートディレクトリと `lost+found` という名前の特別なディレクトリ (116 ページの「`lost+found` ディレクトリ」を参照) 以外のエレメントは含まれていません。動的ビュー、スナップショットビュー、Web ビューを次のように使用して、VOB アクセスを確認できます。

- 動的ビューを使用する場合は、VOB をマウントし、動的ビューをアクティブにして、ビューから次のようなコマンドを実行します。

```
cleartool ls vob-tag
```

この結果、VOB のルートディレクトリと `lost+found` ディレクトリが表示されます。

- スナップショットビューまたは Web ビューを使用する場合は、VOB をビューのロード規則に追加し、VOB のルートディレクトリをビューにロードできることを確認します。

**メモ:** 動的ビューでは VOB ホストのファイル システムへのネットワーク アクセスが必要なため、スナップショットビューまたは Web ビューからアクセスできる VOB に動的ビューからはアクセスできない場合があります。詳細については、99 ページの「VOB 記憶領域へのアクセス要件」を参照してください。

## VOB を NAS デバイス上に作成するには

認定されている NAS デバイスを VOB 記憶領域として使用する場合は、1 つまたは複数の VOB 記憶場所を NAS デバイス上に作成してから (100 ページの「NAS デバイス上でのサーバー記憶場所の作成」を参照してください)、`mkvob -stgloc` コマンドを実行してこの記憶場所に VOB を作成することをお勧めします。たとえば、`nasvobstg` という名前のサーバー記憶場所を作成した Windows リージョン内で次のコマンドを実行して、¥nasvob というタグの付いた VOB を該当の記憶場所に作成します。

```
cleartool mkvob -tag ¥nasvob -stgloc nasvobstg
```

サーバー記憶場所を使用しない場合は、**mkvob** コマンド行でホスト名、グローバルパス、ホスト上のローカルパス、VOB 記憶パス名を指定する必要があります。たとえば、次のコマンドを実行して、**/vobs/nasvob** というタグの付いた VOB を作成します。VOB は、ClearCase のホスト **vobsvr1** (UNIX コンピュータ) 上で実行中の **vob\_server** プロセスによって処理され、VOB 記憶領域は **vobsvr1** によって **/net/nasdevice** にマウントされた NAS デバイス上に保存されます。

```
cleartool mkvob -tag /vobs/nasvob -host vobsvr1¥  
-gpath /net/nasdevice/vobstg/nasvob.vbs ¥  
-hpath /net/nasdevice/vobstg/nasvob.vbs /net/nasdevice/vobstg/nasvob.vbs
```

## VOB を リモート Windows ホスト上に作成するには

VOB を リモート Windows ホスト上に作成するには、リモート ホストの Windows レジストリに保存されている情報に ClearCase からアクセスできる必要があります。Windows レジストリへのリモート アクセスは、次のようにキーにセキュリティを設定することで制限できます。

**HKEY\_LOCAL\_MACHINE¥SYSTEM¥CurrentControlSet¥Control¥SecurePipeServers¥Winreg**

このキー情報は通常は Windows のワークステーションに表示されないため、個別のキーに設定されたセキュリティによってアクセスが制御されます。

Windows サーバー ホスト上では、デフォルトでキーが設定されていて、ワークステーションでサーバーのレジストリ情報を表示できない場合があります。このような場合に VOB をサーバー上に作成しようとする、と、「無効な引数」エラーが発生します。この問題を解決するには、キーを削除するか、キーに設定されたセキュリティを変更します。

レジストリ キーの設定によって VOB をリモート ホスト上に作成できず、キー設定の変更が必要な場合は、ClearCase ドクターによって警告メッセージが表示されます。

## パブリック VOB とプライベート VOB

VOB のマウント方法を制御するために、各 VOB タグの作成時にパブリックかプライベートかを指定します。VOB 自体は、パブリックとプライベートのどちらでもありません。パブリックとプライベートの区別は、タグ全体に対するものです。VOB タグのパブリック状態またはプライベート状態は、マウントされている VOB にのみアクセスできる動的ビューの場合に重要です。スナップショット ビューと Web ビューの場合は、マウントされているかどうかに関係なく、タグが付いていれば VOB にアクセスできます。

動的ビューをサポートしていない ClearCase LT の場合は、VOB がマウントされることはなく、パブリックとプライベートの区別も意味がありません。

**メモ:** すべてのユーザーが、任意の VOB (パブリックまたはプライベート) をマウントできます。プライベートを指定するには、VOB を **cleartool mount -all** コマンドでマウントするのではなく、名前を指定して明示的にマウントする必要があります。

パブリック VOB のタグの処理は、Windows と UNIX で次のように若干異なります。

- UNIX コンピュータの場合、すべてのパブリック VOB は ClearCase の起動時にグループとしてマウントされます。この動作を無効にするには、VOB の作成時に `noauto` マウント オプションを使用します (`mkvob -options` を参照)。
- Windows コンピュータの場合は、すべてのパブリック VOB は次のコマンドを実行するとグループとしてマウントされます。

```
cleartool mount -all
```

`cleartool mount` コマンドに `-persistent` オプションを指定すると、指定した VOB はログオンのたびにマウントされます。ログオンのたびにすべてのパブリック VOB を強制的にマウントするには、次のコマンドを実行します。

```
cleartool mount -all -persistent
```

また、ClearCase のショートカット メニューから VOB を初めてマウントしたときに [マウント] ダイアログ ボックスの [ログオン時に再接続] チェック ボックスをオンにすることで、次のログオン時から VOB がマウントされるように設定することもできます。

**メモ:** Windows と UNIX のどちらの場合でも、UCM を使用している場合は、ユーザーが動的 UCM ビューを起動すると、プロジェクトのすべての VOB がマウントされます。

## パブリック VOB を作成するには

パブリック VOB を作成するには、`mkvob` コマンド行で `-public` を指定するか、VOB 作成ウィザードで [パブリック VOB にする] チェック ボックスをオンにする必要があります。また、ClearCase のレジストリ パスワードも指定する必要があります。

`-public` オプションを指定しても、ClearCase のレジストリ パスワードが不明な場合 (または ClearCase のレジストリ パスワードが設定されていない場合)、VOB はタグ無しで作成されるため、別の操作でタグを作成する必要があります。ClearCase のレジストリ パスワードの詳細については、44 ページの「レジストリ パスワードの設定と使用」と `rgy_passwd` のリファレンス ページを参照してください。

## 管理 VOB への VOB のリンク

VOB は、ブランチ タイプやラベル タイプなどのタイプ オブジェクトが定義されている管理 VOB にリンクできます。タイプ オブジェクトは、ClearCase データ モデルの基礎となる要素です。タイプ管理を簡素化するために、新しい VOB を保存する場所に管理 VOB 階層を作成することをお勧めします。管理 VOB とグローバル タイプの詳細については、第 7 章を参照してください。

**メモ:** PVOB は、PVOB で定義されているプロジェクトに含まれるすべての UCM コンポーネント VOB に対応する管理 VOB です。各 UCM コンポーネント VOB は、PVOB への AdminVOB ハイパーリンク付きで作成されます。プロジェクトで複数の PVOB を使用している場合、各 PVOB には共有コンポーネントとストリームが定義されている共通 PVOB への AdminVOB ハイパーリンクが含まれている必要があります。

VOB 作成ウィザードを使用すると、新規に作成する VOB ごとに管理 VOB を指定できます。mkvob コマンドを実行して VOB を作成する場合は、次に mkhlink コマンドで VOB の管理 VOB を指定して、VOB から管理 VOB への AdminVOB ハイパーリンクを作成する必要があります。

```
cleartool mkhlink -c "link to admin VOB" AdminVOB vob:¥dev vob:¥admin_dev  
ハイパーリンク "AdminVOB@40@¥dev" を作成しました。
```

## NAS デバイスに対応する VOB サーバー ホストの置換

---

VOB 記憶領域を NAS デバイス上に配置すると、VOB 記憶領域を実際に移動せずに、NAS デバイス上にある VOB の VOB ホストを、同じアーキテクチャを持つ別の VOB ホストに簡単に置き換えることができます。

次の手順では、VOB /vobs/libpub の UNIX VOB ホストを、同じアーキテクチャを持つ別の ClearCase ホスト vobsvr1 と置き換えます。この手順では、VOB をロックし、元のタグとオブジェクトを削除してから、新しいオブジェクトとタグを作成して、代替ホストと既存の記憶領域を指定します。

- 1 VOB をロックします。
- 2 必要に応じて、動的ビュー ホスト上にある VOB をアンマウントします。
- 3 (該当する場合) 現在のホスト上にある VOB スナップショットを無効にします。VOB データベースのスナップショットが有効になっている場合は、次のコマンドを実行して無効にします。

```
vob_snapshot_setup rmparam /vobs/libpub
```

- 4 VOB タグを削除し、VOB の登録を解除します。ClearCase 管理コンソールまたは次のコマンドを使用します。

```
cleartool rmtag -vob -all /vobs/libpub
```

```
cleartool unregister -vob /net/nasdevice/vobstg/libpub.vbs
```

5 現在のホスト上にある VOB のサーバー プロセスを終了します。

- UNIX ホストの場合は、VOB を処理する `vob_server` プロセスと `vobrpc_server` プロセスをプロセス テーブルで検索します。UNIX の `ps` コマンドを実行して、出力から `libpub.vbs` に関連付けられたプロセスを検索します。次に、UNIX の `kill` コマンドを実行して、該当のプロセスを終了します。
- Windows ホストの場合は、ClearCase を停止して再起動します。

6 VOB を登録し、新しい VOB タグを作成します。ClearCase 管理コンソールまたは次のコマンドを使用します。

```
cleartool register -vob -host vobsvr1 -hpath /net/nasdevice/vobstg/libpub.vbs ¥  
-gpath /net/nasdevice/vobstg/libpub.vbs /net/nasdevice/vobstg/libpub.vbs  
  
cleartool mktag -vob -replace -host vobsvr1 -tag /vobs/libpub ¥  
-hpath /net/nasdevice/vobstg/libpub.vbs -gpath ¥  
/net/nasdevice/vobstg/libpub.vbs?/net/nasdevice/vobstg/libpub.vbs
```

7 VOB をアンロックします。

8 必要に応じて、動的ビュー ホスト上に VOB をマウントします。

9 (該当する場合) 新しいホスト上の VOB スナップショットを有効にします。新しい VOB ホスト上の VOB データベース スナップショットを有効にする場合は、`vob_snapshot_setup modparam` に適切なパラメータを指定して実行します。

## VOB アクセスに関するトラブルシューティング

---

この項では、VOB にアクセスできない場合に共通する問題の診断と解決方法について説明します。

### 正しくないグローバル パス情報

VOB を作成する場合、使用するツールによって、指定したローカル パス名とグローバル パス名が正しいかどうかを確認されます。VOB ホスト上でツールを実行することにより、確認処理を確実に実行できます。ツールではサーバー ホストから見た VOB 記憶領域へのグローバル パスとローカル パスの有効性をテストできるため、次の条件をすべて満たしている限り、その他のホストからも VOB にアクセスできると考えられます。

- 同じタイプ (UNIX または Windows) のホストである。
- VOB ホストにネットワーク接続できる。
- VOB 記憶ディレクトリが 99 ページの「VOB 記憶領域へのアクセス要件」に記述されている基準を満たしている。

## UNIX の自動マウント機能でホスト マップを使用していない場合

UNIX ホスト上の自動マウント機能でホスト マップを使用していない場合、**mkvob** コマンドと **mkview** コマンドで VOB 記憶領域またはビュー記憶領域へのグローバルパスを作成させるヒューリスティックは失敗することがあります。このようなホストでは、**mkvob** コマンドと **mkview** コマンドに **-host** オプション、**-hpath**、オプション、**-gpath** オプションを指定して、VOB タグまたはビュー タグに保存されているグローバルパスがレジストリ リージョン内のすべてのホストに対して有効になるようにする必要があります。**ClearCase** と自動マウント機能の詳細については、19 ページの「**ClearCase** と NFS 自動マウント機能」を参照してください。

## VOB のグループ所有権情報の調整

VOB にアクセスできる場合でも、VOB を使用する可能性のあるすべてのユーザーが同じグループに所属していないと、**checkout** コマンドまたは **mkelem** コマンド ([ ソース管理に追加 ] コマンド) の処理中にエラーが発生することがあります (ユーザー ID と VOB のアクセス権限の詳細については、第 5 章を参照してください)。VOB のグループは、初めは VOB を作成したユーザーのプライマリ グループと同じです。VOB に完全にアクセスできるのは、プライマリグループのユーザーのみです。

コミュニティに複数のユーザー グループが存在し、1 つの VOB を複数のグループのメンバーで共有する場合は、次の点に注意する必要があります。

- ユーザーが VOB にエレメントを作成するには、そのプライマリ グループが VOB のグループリストに登録されている必要があります。複数のグループのメンバーがエレメントを作成する必要がある場合は、これらのユーザーのプライマリ グループを VOB のグループリストに追加する必要があります。この場合は、**cleartool protectvob** コマンドを使用します。
- 複数のグループのメンバーがエレメントの読み取りアクセスを必要とする場合は、その他に対するエレメントの読み取りアクセス (ディレクトリの場合は、さらに実行アクセス) を設定する必要があります。また、そのエレメントのパスに含まれるすべてのディレクトリ (VOB ルートディレクトリも含む) にも、その他に対する読み取りアクセスと実行アクセスを設定する必要があります。エレメントへのアクセス権限を変更するには、**cleartool protect** コマンドを使用します。詳細と例については、**protect** のリファレンス ページを参照してください。
- ユーザーがエレメントを変更する (たとえば、新しいバージョンをチェックインする) には、エレメントのグループに所属している必要があります。エレメントのグループには、ユーザーが所属する任意のグループを指定できます。

エレメントを作成するには、そのコンテナディレクトリをチェックアウトできなければなりません。また、次の両方の条件が満たされている必要があります。

- ユーザーのプライマリ グループが VOB のグループリストに登録されていること。
- コンテナディレクトリのグループが、ユーザーの所属するグループのいずれかと一致すること。

## UNIX での setuid 実行の有効化

デフォルトでは、ビュールートと VOB のファイル システムは `setuid` と `setgid` を無効化してマウントされます。これは、推奨の構成です。`setuid` と `setgid` を有効化してこれらのファイル システムをホストにマウントできるようにする必要がある場合 (たとえば、VOB で `setuid` ツールを実行する場合) は、`site_prep` スクリプトを実行してこの構成をすべてのホストのデフォルト設定とするようにリリース領域を構成するか、`ccase-home-dir/etc/utils/change_suid_mounts` を実行してこの構成を特定のホストに適用します。

ホスト上で実行した `change_suid_mounts` をインストール全体に適用するには、スクリプトの実行後にホスト上の `ClearCase` を停止して再起動する必要があります。詳細については、『インストール ショーガイド』に記載されている `site_prep` の説明を参照してください。

## VOB へのアクセスの無効化

---

VOB にアクセスできないようにするには、VOB をロックするか、VOB タグを削除します。

### VOB をロックするには

VOB をロックすると、VOB を変更できなくなります。ロックした VOB は動的ビューからの読み取り専用アクセス用にマウントでき、そのエレメントをスナップショット ビューまたは Web ビューにロードすることはできますが、エレメントのチェックアウトやチェックインとメタデータの変更はできません。VOB をロックすると、保留中のすべてのトランザクションがディスクにフラッシュされ、VOB がバックアップ用に準備されます。

VOB をロックまたはアンロックするには、`ClearCase` 管理コンソールか、`cleartool lock` コマンドまたは `cleartool unlock` コマンドを使用します。VOB 上のロック状態を一覧表示するには、`ClearCase` 管理コンソールまたは `cleartool lslock` コマンドを使用します。

### VOB タグを削除するには

VOB のタグを削除すると、VOB にアクセスできなくなります。VOB タグを削除するには、`ClearCase` 管理コンソールまたは `cleartool rmtag` コマンドを使用します。また、VOB の登録を解除して、新しいタグが作成されないようにすることもできます。VOB の登録を解除するには、`ClearCase` 管理コンソールまたは `cleartool unregister` コマンドを使用します。

## VOB の削除

---

VOB を削除すると、VOB のすべてのデータが破棄されます。VOB に含まれるデータが不要になった場合以外は、VOB を削除しないでください。

**注意:** レプリカ VOB の場合は、特別な注意が必要です。レプリカ VOB を削除する手順については、『Rational ClearCase MultiSite 管理ガイド』を参照してください。

VOB を削除するには

- 1 VOB が使用中でないことを確認します。動的ビューを使用するクライアントの場合は、VOB をアンマウントする必要があります。スナップショット ビューを使用するユーザーの場合は、ロード規則から VOB を削除する必要があります。
- 2 VOB を削除します。rmvob コマンドまたは ClearCase 管理コンソールを使用します。ClearCase 管理コンソールを使用する場合は、VOB の VOB 記憶ノード (VOB 記憶ディレクトリが配置されているホストを表すホスト ノードのサブノード) に移動して、[操作] メニューの [すべてのタスク] をポイントし、[VOB の削除] をクリックします。
- 3 VOB ホスト上で ClearCase を停止して再起動します。これにより、VOB のサーバー プロセスが終了します。

## VOB 記憶ディレクトリ

---

この項では、VOB 記憶ディレクトリの内容について説明します。記憶ディレクトリの内容を理解することは、VOB 記憶領域のメンテナンスを行う上で役立ちます。

**注意:** VOB 記憶ディレクトリとその内容は、ClearCase のコマンドとサービスによって作成、管理されます。VOB 記憶ディレクトリまたは記憶ディレクトリに格納されているファイルやサブディレクトリの内容または保護を変更する場合は、ClearCase 以外のコマンドやユーティリティを使用しないでください。

VOB 記憶ディレクトリは、VOB ホストまたはサポートされているネットワーク接続記憶デバイス上に保存できます。ネイティブのオペレーティング システムのユーティリティを使用して、VOB 記憶ディレクトリとサブディレクトリやファイルを表示すると、ディレクトリ構造も表示されます。サブディレクトリとファイルには次のものがあります。

.pid	VOB に関連付けられた vob_server プロセス ID を含む 1 行から成るテキスト ファイル。
admin	VOB が使用しているディスク容量に関連した管理データが保存されているディレクトリ。管理データを表示するには、cleartool space -vob コマンドまたは ClearCase 管理コンソールを使用します。

<code>vob_oid</code>	ユニバーサルユニーク ID (UUID) で表される VOB のオブジェクト識別子 (OID) を含む 1 行から成るテキスト ファイル。この UUID は、VOB ファミリ (ClearCase MultiSite) 内のすべてのレプリカに共通です。UUID は VOB のデータベースにも保存されています。UUID を変更することはできません。
<code>replica_uuid</code>	VOB の特定レプリカのレプリカ UUID を含む 1 行から成るテキスト ファイル。ClearCase MultiSite で作成された別のレプリカには、異なるレプリカ UUID が割り当てられます。
<code>.identity</code>	UNIX ホスト上のサブディレクトリ。ビューの所有者とグループ メンバシップを確立するファイルが保存されています。79 ページの「VOB と VOB オブジェクトに対するアクセス コントロール」を参照してください。
<code>identity.sd</code>	Windows ホスト上のバイナリ データ ファイル。VOB 記憶ディレクトリのセキュリティ記述子が含まれています。
<code>groups.sd</code>	Windows のファイル。VOB の補足グループのセキュリティ記述子が含まれています。
<code>s</code>	VOB のソース記憶プールが格納されているサブディレクトリ。
<code>d</code>	VOB の派生オブジェクト記憶プールが格納されているサブディレクトリ。
<code>c</code>	VOB の cleartext 記憶プールが格納されているサブディレクトリ。
<code>db</code>	VOB の組み込みデータベースを実装するファイルが保存されているサブディレクトリ。
<code>vob_server.conf</code>	起動時に <code>vob_server</code> によって読み取られるテキスト ファイル。ソース コンテナの削除の遅延に関する設定が保存されています。セミライブ バックアップが有効化されていると、削除の遅延がアクティブになります。詳細については、 <code>vob_snapshot_setup</code> のリファレンス ページを参照してください。
<code>.hostname</code>	VOB のサーバー ホスト名が記録されたテキスト ファイル。
<code>.msadm_acls</code>	MultiSite 管理サーバーの ACL が保存されているファイル。

以下の項では、VOB 記憶ディレクトリのサブディレクトリについて説明します。

## VOB 記憶プール

VOB 記憶ディレクトリのサブディレクトリである **c**、**d**、**s** には、VOB の記憶プールが格納されています。UNIX の場合、プールには別のディスク ボリュームまたは別のホストにあるディレクトリへのシンボリック リンクを指定できます。Windows の場合、プールには VOB 記憶ディレクトリのサブディレクトリを指定する必要があります。これらのディレクトリ内の記憶プールには、データ コンテナ ファイルが保存されています。コンテナ ファイルには、エレメントのバージョン、共有派生オブジェクト、キャッシュされた **cleartext** が保存されています。エレメント タイプによっては、エレメントのバージョンが別のコンテナに保存されたり、バージョン間のデルタ (差分) として 1 つのファイルに収められることがあります。

各 VOB 記憶プール ディレクトリの作成時には、デフォルトの記憶プールである 1 つのサブディレクトリが作成されます。デフォルトのプールには、次の 3 つがあります。

<b>s¥sdft</b>	デフォルトのソース記憶プール。ClearCase の制御下にあるファイルの内容を永久に保存します。
<b>c¥cdfdft</b>	デフォルトの <b>cleartext</b> 記憶プール。現在使用中の <b>cleartext</b> バージョン ( <b>text_file</b> エレメントの構築されたバージョンなど) を一時的に保存します。
<b>d¥ddft</b>	デフォルトの派生オブジェクト記憶プール。プロモートされた派生オブジェクトと共有派生オブジェクトを保存します。

必要に応じて、追加のプールを作成できます。ClearCase 管理コンソールで [VOB] ノードの [プール] サブノードを使用するか、**cleartool mkpool** コマンドまたは **cleartool chpool** コマンドを実行します。各記憶プールには、1 種類のデータ コンテナ (ソース、**cleartext**、DO のいずれか) が格納されます。記憶プールの管理の詳細については、117 ページの「VOB の追加記憶領域の作成」を参照してください。

**メモ:** ClearCase LT では、VOB にデフォルト プールのみが使用されることがあります。

## ソース記憶プール

各ソース記憶プールには、特定のファイル エレメント セットのすべてのソース データ コンテナが保存されています。各ソース データ コンテナには、特定のファイル エレメントの 1 つまたは複数のバージョンの内容が保存されています。たとえば、あるソース データ コンテナには、**text\_file** タイプのエレメントのすべてのバージョンが保存されています。このエレメント タイプのタイプ マネージャ プログラムによって、コンテナ内のデルタから各 **cleartext** バージョンが構築されます。また、新しいバージョンがチェックインされたときに、新しいデルタでデータ コンテナが更新されます。

ソース プールには、GUI と、**checkout** や **checkin** などの **cleartool** コマンドを使用してアクセスできます。また、ファイルやディレクトリがチェックアウトされているかどうかに関係なく、動的ビューでファイルやディレクトリを開くプログラムによってアクセスすることもできます。エレメントの **cleartext** バージョンが有効な場合は、それが使用されます。有効でない場合は、バージョンが作成されて **cleartext** プールに保存されます。

## cleartext 記憶プール

各 **cleartext** 記憶プールには、特定のファイルエレメントセットのすべての **cleartext** データ コンテナが格納されています。各 **cleartext** データ コンテナには、特定のエレメントの 1 つのバージョンが保存されています。**cleartext** 記憶プールは、すべてのバージョンが 1 つのコンテナに保存されている **text\_file** や **compressed\_text\_file** などのエレメント タイプへのアクセスを高速にするためのキャッシュです。

たとえば、**text\_file\_delta** エレメントのバージョンが初めて要求されると、**text\_file\_delta** タイプ マネージャはエレメントのソース データ コンテナから目的のバージョンを構築します。構築されたバージョンは、**cleartext** データ コンテナ (通常のテキスト ファイル) として **cleartext** 記憶プール内にキャッシュされます。次のアクセスからは、**ClearCase** によってまず **cleartext** プールが検索され、要求されたバージョンがプール内に存在しない場合にのみ、新しいバージョンが構築されます。

**cleartext** プールは定期的にスクラブされ、しばらくアクセスされていないバージョンは削除されます。詳細については、238 ページの「**VOB** 記憶プールのスクラブ」を参照してください。

## 派生オブジェクト記憶プール

各派生オブジェクト記憶プールには、派生オブジェクト データ コンテナのコレクションが保存されます。派生オブジェクト データ コンテナには、**clearmake** や **clearaudit** などのビルド ツールによって動的ビューに作成された **DO** のファイル システム データ (通常はバイナリ データ) が保存されます。

**DO** 記憶プールには、**winkin** コマンドによって **VOB** にプロモートされた派生オブジェクトのデータ コンテナのみが保存されます。各ディレクトリ エレメントは、特定の **DO** 記憶プールに割り当てられます。ディレクトリ内に作成された **DO** を初めてウィंकインすると、データ コンテナは対応する **DO** 記憶プールにコピーされます。非共有派生オブジェクトと共有不可派生オブジェクトのデータ コンテナは、ビュープライベート記憶領域に保存されます。

**cleartext** プールの場合と同様に、**DO** 記憶プールも定期的にスクラブされ、関係のない **DO** は削除されます。

**メモ:** **ClearCase LT** は派生オブジェクトをサポートしていないため、**ClearCase LT VOB** の **DO** プールは常に空の状態です。

## VOB データベース

各 **VOB** には、固有のデータベースがあります。各データベースは、組み込みデータベース管理システムによって管理され、**VOB** 記憶ディレクトリの **db** サブディレクトリにファイル セットとして実装されます。**VOB** データベースには、次のデータが保存されています。

- バージョン管理情報: エレメントと、そのブランチ構造とバージョン。
- ファイル システム オブジェクトに関連付けられたメタデータ: バージョン ラベルや属性など。

- イベント レコードと構成レコード: 成果物の変更内容と関連の開発アクティビティを記録します。
- タイプ オブジェクト: バージョン管理構造の実装とメタデータの実装に関係します。
- (PVOB の場合のみ) UCM オブジェクト: フォルダ、プロジェクト、ストリーム、アクティビティ、コンポーネント、ベースラインなど。

ClearCase の制御下にある成果物の永続的な内容は、VOB データベースではなく、ソース プール(.s とそのサブディレクトリ) に保存されます。

db ディレクトリには、次のファイルが保存されます。

vob_db.dbd	データベースにアクセスするために組み込みデータベース ルーチンによって使用されるコンパイル済みデータベース スキーマ。スキーマには、VOB データベースの構造が記述されています。
vob_db_schema_version	コンパイル済みスキーマ ファイルが必要なリビジョン レベルであることを確認するために組み込みデータベース ルーチンによって使用されるスキーマ バージョン ファイル。
vob_db.d0n vob_db.k0n	データベースの内容を保存するファイル。
vista.*	データベース コントロール ファイルとトランザクション ログ。
db_dumper (UNIX)	ccase-home-dir/etc/db_dumper のバックアップ コピー。 ccase-home-dir/etc/dumpers/db_dumper.num (num は VOB のスキーマ バージョン) を起動できない場合に限り、reformatvob によって db_dumper のコピーが起動されます。
db_dumper (Windows)	ccase-home-dir¥bin¥db_dumper.exe のコピー。このファイルは実行可能プログラムであり、reformatvob コマンドのダンプ フェーズにおいて起動されます。各 VOB には db_dumper のコピーがあるため、VOB 自体のダンプをいつでも実行できます (通常は ClearCase の新しいリリースをホストにインストールした後で、VOB をダンプする必要があります。このとき、新しいリリースの ccase-home-dir¥bin¥db_dumper プログラムに古い VOB データベース形式を指定する必要はありません)。
vob_db.str_file	長い文字列を保存するデータベース文字列ファイル。

## 保存されているデータベース サブディレクトリ

reformatvob コマンドで形式変更した VOB のルートディレクトリに、VOB データベースを含む保存済み db ディレクトリが格納されている場合があります。これは、形式変更する前にこのディレクトリが存在していたためです。reformatvob では、名前を変更することによって、古いデータベースを維持します。したがって、VOB 記憶ディレクトリには、db.0318 などの名前が付いた古い (通常は不要な) VOB データベース サブディレクトリが格納されます。reformatvob の処理が中断されると、db.reformat という名前の部分的に形式変更されたデータベースが残ることがあります。

## .identity ディレクトリ

UNIX プラットフォームでは、**.identity** という名前のディレクトリに VOB の所有権とグループメンバシップの情報が記録されます。このディレクトリへのアクセスは、VOB 所有者のみに制限されています。

**.identity** ディレクトリには、次のファイルが保存されます。

uid	このファイルの所有者は VOB 所有者です。
gid	このファイルが所属するグループは、VOB の基本グループです。
group.nn	各追加ファイル (該当する場合) は、グループ メンバシップによって、VOB のグループ リスト上の追加グループを示します。また、ファイル名の数値 ID (group.30 や group.2 など) によってグループを識別します。

cleartool のサブコマンド **describe** と **protectvob** を使用して、**.identity** または **identity.sd** に記録されている VOB 所有権情報の内容を表示したり、必要に応じて変更したりすることができます。

**メモ:** Windows プラットフォームでは、同じような識別情報が VOB ルート ディレクトリにある **identity.sd** ファイルと **groups.sd** ファイルに保存されます。これらのファイルには、VOB の所有者、基本グループ、補足グループを記述する Windows セキュリティ記述子が格納されています。

## VOB 記憶領域の管理

---

VOB 記憶領域は、VOB を使用する開発者の数と、VOB で成果物を作成、変更する頻度按比例して増加します。VOB 記憶領域管理は、重要な管理作業です。VOB 記憶領域管理には、次の 3 つの基本タスクがあります。

- 各 VOB が消費する記憶領域の監視。
- VOB の増加の制御。不要なデータとメタデータを削除することによって制御します。VOB データとメタデータの多くは長期間にわたって保存する必要がありますが、中にはすぐに価値がなくなるものもあり、それらは削除しても問題ありません。
- VOB に新しい記憶プールを追加することによる追加記憶領域の作成、VOB 間でのデータの再配置、別のパーティションやホストへの VOB 全体の移動。

## ClearCase スケジューラの使用

ClearCase スケジューラは、VOB のディスク使用量に関するデータを収集するジョブや、ローカル VOB の超過ディスク使用量を解放するジョブを実行します。ジョブには次のものがあります。

- VOB のディスク使用量に関する毎日のデータ収集
- 共有派生オブジェクトのディスク使用量に関する毎週のデータ収集
- scrubber ユーティリティによる毎日の VOB 記憶プールのスクラブ
- vob\_scrubber ユーティリティによる毎週の VOB データベースのスクラブ
- 必要に応じてカスタマイズできるジョブの毎日および毎週の実行

多くの場合、VOB 記憶領域の増加の監視と管理には、デフォルトのスケジュールで実行されるデフォルトのジョブ セットが適しています。デフォルトのジョブ セットとデフォルトのスケジュールの変更については、第 12 章を参照してください。

## VOB 記憶領域の監視

ClearCase には、VOB のディスク使用量に関する情報を表示するためのコマンド行と GUI ツールがあります。

- ClearCase 管理コンソールでは、ClearCase ホスト ノードの各 [VOB] サブノードに対応する VOB 記憶ディレクトリ ノードに、VOB の現在と過去のディスク使用量が表示されます。VOB 記憶ノードの [派生オブジェクト] サブノードには、VOB 内の共有派生オブジェクトのディスク使用量が表示され、DO への参照が保存されているビューも表示されます。
- cleartool space -vob コマンドを実行すると、VOB の現在と過去のディスク使用量が表示されます。
- cleartool dospace コマンドを実行すると、VOB 内の共有派生オブジェクトのディスク使用量が表示されます。また、DO への参照が保存されているビューも表示されます。

**メモ:** これらのツールには、VOB の容量に関する表示データを生成する 1 つまたは複数のスケジュール済みジョブが関係します。これらのジョブが有効になっていないか実行されていない場合は、ツールを実行しても適切な結果は得られません。スケジュール済みジョブの詳細については、第 12 章を参照してください。

## スクラブ

ClearCase には scrubber と vob\_scrubber という 2 つのユーティリティ プログラムがあります。これらのユーティリティ プログラムを使用して、VOB の記憶プールとデータベースの増加を制御できます。ClearCase スケジューラでは、スクラブの対象を指定するデフォルト設定を使用して、ホスト上のすべての VOB に対し、これらのプログラムを毎日または毎週実行します。デフォルト設定の使用には注意が必要ですが、ほとんどの場合は適度にアクティブな VOB の増加を管理するのに適しています。スクラブのスケジュールおよびパラメータと、各 VOB のニーズに合わせたスクラブの調整方法の詳細については、238 ページの「VOB 記憶領域の増加を制御するスクラブ」を参照してください。

## VOB からの不要なバージョンの削除

スクラブでは、再生成できる成果物のみが削除されます。エレメントやバージョンは削除されません。エレメントとバージョンは履歴データであるため、削除する場合は十分に注意する必要があります。**cleartool rmelem** を実行してエレメント全体を削除するのは特に危険です。

- 現在の作業において不要になったエレメントでも、以前の作業の再構築やメンテナンスにおいて必要になる場合があります。
- **rmelem** を実行すると、エレメントがカタログされていたすべてのディレクトリ バージョンからエレメント名が削除されます。エレメントの履歴が削除されてしまうので、古いディレクトリ バージョンを一覧表示したり比較したりする場合でもそのエレメントは表示されません。**rmname** コマンドを実行してエレメント名を **VOB** から削除した場合は、エレメントの履歴は残りますが、ディレクトリの後続のバージョンからエレメント名が削除されます。
- 操作を誤ると、非常に手間がかかる場合があります。たとえば、バックアップから 1 つのエレメントを回復する手順は複雑です。また、**UCM** コンポーネントではこの処理を使用できません 詳細については、219 ページの「バックアップからの特定エレメントの復元」を参照してください。

ソース プールのディスク容量を解放する必要がある場合は、エレメント全体を削除するよりも、エレメントのバージョンを 1 つずつ削除するほうが賢明です。**rmver** コマンドを使用すれば、今後は必要ないと思われるバージョンを簡単に削除できます。

デフォルトでは、**rmver** は次の基準を満たすバージョンのみを削除します。

- ブランチに関係しないバージョン (ブランチ ポイントになく、ブランチの起点でも終点でもないバージョン)
- メタデータ注釈 (バージョン ラベル、属性、ハイパーリンクなど) を持たないバージョン

**rmver** には、これらの基準を満たさないバージョンを強制的に削除するためのオプションがあります。詳細については、**rmver** のリファレンス ページを参照してください。

また、**cleartool relocate** コマンドを使用して、**VOB** ディレクトリ全体とそのディレクトリに格納されているすべてのエレメント バージョンを **VOB** 間で移動することもできます。詳細については、147 ページの「別の **VOB** へのエレメントの再配置」を参照してください。

## lost+found ディレクトリ

---

各 **VOB** には **lost+found** という名前の特別なディレクトリ エレメントがあります。このディレクトリには、**VOB** のディレクトリ バージョンにカタログされていないために取り残されるエレメントが保存されます。エレメントは、次の場合に取り残されます。

- 新しいエレメントの作成後、エレメントを作成したディレクトリのチェックアウトをキャンセルした場合
- `rmname` コマンドを使用してエレメントへの最後の参照を削除した場合
- `rmver`、`rmbranch`、`rmelem` のいずれかのコマンドを使用してディレクトリ バージョンを削除することによって、エレメントへの最後の参照を削除した場合

`lost+found` に移動したエレメントには、フォーム名が割り当てられます。

`element_leaf_name.UUID`

たとえば、`util.c` という名前のエレメントには、`util.c.41a00000bcaa11caacd0080069021c7` という名前が付けられます。

`lost+found` ディレクトリには、次のような固有の特性があります。

- チェックアウトできない。
- チェックアウトしなくてもディレクトリの内容を変更できる。
- ディレクトリ内にはブランチを作成できない。

ディスク容量を節約するには、`lost+found` ディレクトリを定期的にクリーン アップします。

- `lost+found` ディレクトリ内のエレメントが必要な場合は、`cleartool mv` コマンドを使用してエレメントを別のディレクトリに移動します。
- エレメントが確実に不要になった場合は、`cleartool rmelem` を使用して、`lost+found` ディレクトリからエレメントを削除します。`rmelem` の使用法の詳細については、`rmelem` のリファレンス ページを参照してください。

**メモ：** スナップショット ビューから `lost+found` ディレクトリにアクセスするには、ビューのロード規則にこのディレクトリを含める必要があります。`lost+found` ディレクトリとそのエレメントをビューにロードした後は、必要に応じてエレメントを移動したり削除したりすることができます。

## VOB の追加記憶領域の作成

---

**メモ：** ClearCase LT の VOB ではデフォルトのプールのみが使用されるので、この項で説明する内容は適用されません。

VOB には柔軟性のあるプール構成オプションがあり、VOB 記憶領域管理に利用できます。ここでは、次のトピックについて説明します。

- VOB 記憶ディレクトリでの追加記憶プールの作成
- UNIX シンボリック リンクを使用した、別のディレクトリまたは別のホストでの追加プールの作成
- 特定プールへのエレメントの割り当て

## 追加記憶プールの作成

VOB の作成後は、追加記憶プールを必要な数だけ作成でき、必要に応じてその内容を調整できます。それには、ClearCase 管理コンソールで [VOB] ノードの [プール] サブノードを使用するか、cleartool mkpool コマンドや cleartool chpool コマンドを実行します。UNIX ホストでは、リモート記憶プールも作成できます。

## UNIX ホストでのリモート記憶プールの作成

別の UNIX ホスト上または NAS デバイス上にあるファイル システムへのシンボリック リンクをサポートしている UNIX ホストの場合、VOB の記憶プールを分散して、VOB 記憶領域の容量を増やすことができます。これらのリモート プールは、別のホスト上 (ClearCase がインストールされている必要はない) または VOB ホスト上の別のディスク パーティションに配置できます。

リモート記憶プールを使用するかどうかを判断する場合は、次の要件を検討する必要があります。

- リモート記憶プールを使用する VOB のバックアップと回復は、複雑になる場合があります。201 ページの「VOB にリモート記憶プールがある場合」を参照してください。
- リモート記憶プールを使用する VOB に Windows コンピュータからアクセスするには、VOB タグに詳細情報を含める必要があります。319 ページの「リモート記憶プールのある UNIX VOB の Windows タグ」を参照してください。
- VOB を使用するすべてのホストからネットワーク経由でリモート サイトにアクセスできる必要があります。複数のネットワーク インターフェイス (および複数のホスト名) を使用するリモート ホストでは、正常に処理されないことがあります。
- 認定されている NAS デバイスに VOB 記憶ディレクトリ全体を配置しない場合は、VOB データベース (db サブディレクトリ) を VOB 記憶ディレクトリのサブディレクトリにする必要があります。この場合、別のコンピュータ上のディレクトリへのシンボリック リンクは指定できません。

新しい記憶プールに使用するホストを決定する場合は、記憶プールの種類によって使用パターンが次のように異なることに注意してください。

- **ソース プール:** ソース プールには、ファイル エレメントのチェックイン済みバージョンという最も重要なデータが保存されます。ソース プール間のトラフィックは比較的少ないですが、データの一貫性はきわめて重要です。ソース記憶プールは VOB 記憶ディレクトリ内に配置することをお勧めします。これにより、データの一貫性が最適化されます。1 つのディスク パーティションに VOB のすべての必須データが保存されるためです。また、バックアップと復元の手順も簡単になります。通常はパフォーマンスよりもデータの一貫性を優先します。ソース プールが消失すると、消失したバージョンを開発者が再作成しなければならないからです。リモート ソース プールを使用する場合は、データのバックアップを頻繁かつ確実に行えるような強力なファイル サーバーを選択してください。

- **cleartext プール**: cleartext プール間のトラフィックは (ファイル エLEMENTの多くがデルタまたは圧縮形式で保存されると想定されるため) 最も多いと考えられますが、このプール内のデータはソース プールのデータから再作成できるので維持管理の必要がありません。cleartext プールの理想的なホストは、高速ファイル システムを装備したコンピュータです。
- **派生オブジェクト プール**: 派生オブジェクトが多数のリリースとプラットフォームで維持される場合、派生オブジェクト プールはかなり大きくなる可能性があります。派生オブジェクトは、cleartext と同様に、通常は VOB データベースとソース プールのデータから再作成できますが、古い派生オブジェクトからの再作成は難しい場合があります。ClearCase LT は派生オブジェクトをサポートしていないため、ClearCase LT VOB の DO プールは常に空の状態です。

**メモ**: 派生オブジェクトをチェックインすると、その派生オブジェクトはバージョン付きオブジェクトになり、派生オブジェクト プールに保存されなくなります。

## エレメントのソース プール割り当ての変更

新規に作成されたすべてのエレメントには親ディレクトリのプール割り当てが継承されるため、新しいプールを作成してエレメントを再割り当てするまでは、VOB 内のすべてのエレメントにはデフォルトの記憶プールが使用されます。UNIX ホストでは、リモート プールとプール再割り当てを組み合わせることで、記憶領域の消費を効率よく管理できます。たとえば、VOB の記憶ディレクトリが大規模なバージョン付き DO でいっぱいになりそうな場合は、リモートボリュームに新しい DO プールを作成し、エレメントを新しいプールに再割り当てします。

chpool コマンドを使用して、1 つのエレメントに関連付けられたソース プールか cleartext プール、またはその両方を変更できます。ファイルエレメントのソース プールを変更すると、関連するすべてのデータ コンテナが新しいプールに移動します。ディレクトリ エLEMENTのソース プールを変更すると、ディレクトリ内に作成されたすべての新しいエレメントは新しいソース プールに割り当てられます。既存のエレメントは、現在のプールに残ります。

## 例: ディレクトリ内のすべてのファイルを新しいプールに割り当てる

次に、UNIX 上の VOB に新しいリモート ソース記憶プールを作成し、特定ディレクトリの現在と今後のすべてのエレメントを新しいプールに再割り当てする例を示します。

- 1 **新しい記憶プールを作成します。**リモート プールのグローバルパス名を指定します。パス名には VOB にアクセスするすべてのホストに対して有効な名前を指定する必要があります。

```
cd /vobs/bgr
```

```
cleartool mkpool -source -ln /net/ccsvr02/ccase_pools/bgrsrc2 bgrsrc2
```

"bgrsrc2" に関するコメント:

リモート ソース記憶プール

.

プール "bgrsrc2" を作成しました。

- 2 既存のファイル エlementを新しいプールに再割り当てします。この例では、特定サブディレクトリ内のすべてのファイル エlementを再割り当てします。

```
cd libbgr
```

```
cleartool find . -type f -exec 'cleartool chpool -force bgrsrc2 $CLEARCASE_PN'
```

"/.Makefile" のプールを "bgrsrc2" に変更しました。

"/.getcwd.c" のプールを "bgrsrc2" に変更しました。

```
.  
.
```

"/.strut.c" のプールを "bgrsrc2" に変更しました。

- 3 ディレクトリ エlementを新しいプールに再割り当てします。これにより、ディレクトリ内に新規に作成されたすべてのファイル エlementとディレクトリ エlementが新しいプールを使用するようになります。

```
cleartool chpool bgrsrc2 .
```

"/." のプールを "bgrsrc2" に変更しました。

## 例：既存の記憶プールを別のディスクに移動する

次の手順で、既存の記憶プールを UNIX ホスト上の別のパーティションに移動します。

- 1 記憶プールの場所を特定します。次のように lspool コマンドを使用します。

```
cleartool lspool -long d_aux@/vobs/bgr
```

```
pool "d_aux"
```

```
.  
.
```

プール記憶グローバルパス名

```
"/net/ccsvr01/vobstore/bgr.vbs/d/d_aux"
```

- 2 VOB をロックします。
- 3 記憶プールの内容をコピーします。記憶プールは UNIX の標準ディレクトリです。内容を新しい場所にコピーするには、cp、rcp、tar などのコマンドを使用します。例を次に示します。

```
rlogin ccsvr01
```

```
mkdir -p /vobstore_2/DO_pools
```

```
cp -r /vobstore/bgr.vbs/d/d_aux /vobstore_2/DO_pools
```

- 4 古い記憶プールをシンボリック リンクで置き換えます。古い記憶プールを別の場所に移動し、元の場所にリンクを作成します。

```
cd /vobstore/bgr.vbs/d
```

```
mv d_aux d_aux.MOVED
```

```
ln -s /net/ccsvr01/vobstore_2/DO_pools/d_aux d_aux
```

VOB にアクセスするすべてのホストに対してシンボリック リンクが有効であることを確認します。

- 5 VOB をアンロックします。

- 6 古い記憶プールを削除します。記憶プールが新しい場所で正しく機能することを確認してから、古い記憶プールを削除します。

```
rm -fr /vobstore/bgr.vbs/d/d_aux.MOVED
```

**メモ:** VOB のタグが Windows リージョン内にある場合は、再配置された記憶プールに対応するようにタグを変更する必要があります。319 ページの「リモート記憶プールのある UNIX VOB の Windows タグ」を参照してください。



# VOB のデータ タイプと 管理 VOB 階層

# 7

VOB メタデータには、多くのタイプがあります。特定の VOB に固有のものもありますが、ほとんどのタイプは、関連する成果物 (UCM プロジェクトの各種コンポーネントなど) が保存されている VOB グループ全体で一貫して管理する必要があります。この章では、VOB メタデータについて説明します。また、管理 VOB 階層を使用して、関連する VOB 間でタイプ オブジェクトを簡単に共有する方法についても説明します。

## VOB のデータ タイプ

---

VOB は、データ (ファイルやディレクトリなどのバージョン付き成果物) とメタデータ (プランチ、ラベル、イベント レコードなど) のリポジトリです。メタデータのタイプには VOB オブジェクトとして保存されるものもありますが、それ以外のメタデータはレコードとして保存されるか、オブジェクトに関連付けられた注釈として保存されます。この項では、各種の管理タスクをコンテキスト内に配置するための VOB データ タイプの概要について説明します。

VOB のデータ タイプには、ClearCase コマンドによって自動的に作成されるものがあります。その他のデータ タイプは、ユーザーまたは (通常の場合) 管理者が明示的に作成する必要があります。VOB メタデータにアクセスするには、`cleartool describe` などのコマンドや ClearCase の各種 GUI を使用します。

## VOB オブジェクトとレプリカ オブジェクト

各 VOB データベースには、VOB 自体を表す VOB オブジェクトが含まれています。VOB オブジェクトを使用して、特定の操作を行います。たとえば、次のような操作を行います。

- VOB 全体に影響する操作のイベント レコードの一覧表示 (`lshistory` のリファレンス ページを参照)。このような操作には、タイプ オブジェクトの作成と削除や、エレメントの削除などが含まれます。
- VOB 全体のロック (`lock` のリファレンス ページを参照)。

VOB がレプリカの場合、`mkreplica` コマンドによって作成されたレプリカ オブジェクトも含まれます。

## ファイル システム オブジェクト

VOB データベースでは、次のような VOB 内のファイル システム オブジェクトを使用して、ファイル、ディレクトリ、VOB シンボリック リンクを追跡します。

ファイル エlement	ブランチとバージョンで構成されるバージョン ツリーを持つオブジェクト。ファイル エlementの各バージョンには、ファイル システム データ (バイトで表される一連のデータ) が含まれています。特定のエlementタイプには、バージョンのファイル システム データの性質を制約するものがあります。たとえば、 <b>text_file</b> というタイプのエlementのバージョンには、テキスト行を使用する必要があり、バイナリ データを含めることはできません。
ディレクトリ エlement	ブランチとバージョンで構成されるバージョン ツリーを持つオブジェクト。ディレクトリ エlementの各バージョンは、一連のファイル エlement、ディレクトリ エlement (サブディレクトリ)、VOB シンボリック リンク、ハードリンクをカタログします。
VOB シンボリック リンク	テキスト文字列を含むオブジェクト。UNIX システムの場合、テキスト文字列は標準コマンドによってオペレーティング システムのシンボリック リンクの場合と同じ方法で解釈されます。
VOB ハードリンク	現在のディレクトリのバージョンに既存するエlementの別名。

## イベント レコード

VOB を変更するほとんどの操作 (**checkout** や **checkin** など) によって、VOB データベース内にイベント レコードが作成されます。詳細については、**events\_ccase** のリファレンス ページを参照してください。

## 共有可能派生オブジェクト

VOB のデータベースには、VOB 内の指定されたパス名で作成されたすべての共有可能派生オブジェクト (DO) に関する情報が保存されています。各 DO について、データベースには次のような情報がカタログされています。

- DO の名前と DO に含まれているディレクトリの名前
- DO 固有の識別子
- DO のショッピング情報 (**clearmake** と **omake** で使用)

DO の作成とアクセスは動的ビューによってのみ行われるため、ClearCase LT では DO は作成されません。

# 構成レコード

VOB データベースには、共有可能派生オブジェクトと DO バージョン (チェックイン済みの派生オブジェクト) に関連付けられた構成レコード (CR) が保存されています。各 CR には、通常 1 つのビルドスクリプトの実行結果である 1 つのターゲット再ビルドが記述されています。

DO と同様に、CR の作成とアクセスは動的ビューによってのみ行われるため、ClearCase LT では CR は作成されません。

# タイプオブジェクト

タイプオブジェクトは、VOB データベースに保存された 1 つまたは複数のタイプインスタンスのプロトタイプです。タイプオブジェクトが存在する場合、ユーザーは該当のコマンド (ラベルタイプオブジェクトのインスタンスを作成する `cleartool mklabel` など) を入力することによって、インスタンスを作成できます。

VOB には、次のような各種のタイプオブジェクトを保存できます。

タイプ	ニーモニック	説明
エレメント タイプ	eltype	インスタンスはエレメントです。
ブランチ タイプ	brtype	インスタンスはブランチです。
ハイパーリンク タイプ	hltype	インスタンスは 2 つの関連オブジェクトを接続する VOB ハイパーリンクです。
トリガ タイプ	trtype	インスタンスはトリガです。

VOB には、ほかのタイプオブジェクトのインスタンスの変更にのみ使用される次のようなタイプオブジェクトも保存されます。

タイプ	ニーモニック	説明
ラベル タイプ	lbtype	インスタンスはバージョンオブジェクトに関連付けることができるラベルです。
属性タイプ	attype	インスタンスはタイプオブジェクトのインスタンスに関連付けることができる属性 ( <code>name=value</code> のペア) です。

各タイプオブジェクトに関連付けられたニーモニックは、`describe` などの `cleartool` コマンドのオブジェクトセレクトアのプレフィックスに使用できます。たとえば、`v4_patch` という名前のブランチタイプを記述するには、`brtype` ニーモニックを次のように使用します。

`cleartool describe brtype:v4_patch`

## タイプオブジェクトのインスタンス

タイプオブジェクトを作成したら、タイプのインスタンスを作成できます。タイプのインスタンスを作成すると、タイプオブジェクトへの参照が作成されます。たとえば、**BASELEVEL\_4.2** というラベルを特定のバージョンに関連付けても、**BASELEVEL\_4.2** タイプオブジェクトのコピーは作成されません。その代わりに、エレメントバージョンとラベルタイプオブジェクト間の接続が確立されます。以下に、タイプのインスタンスを作成する **cleartool** コマンドによるタイプオブジェクトの使用方法を示します。

### タイプオブジェクト      タイプオブジェクトとタイプインスタンスの関係

エレメント	<b>VOB</b> 内のファイルエレメントまたはディレクトリエレメントは、 <b>mkelem</b> または <b>mkdir</b> によって、 <b>VOB</b> 内の既存エレメントのインスタンスとして作成されます。
ブランチ	エレメント内の各ブランチは、 <b>mkbranch</b> によって、エレメントの <b>VOB</b> 内の既存ブランチタイプのインスタンスとして作成されます。
ラベル	<b>mklabel</b> コマンドは、オブジェクトに既存ラベルタイプのインスタンスの名前を付けます。
属性	<b>mkattr</b> コマンドは、既存の属性タイプのインスタンスを作成することによって、バージョン、ブランチ、エレメント、 <b>VOB</b> シンボリックリンク、ハイパーリンクに属性の注釈を付けます。属性の各インスタンスは、文字列や整数などの特定の値を持ちます。
ハイパーリンク	<b>mkhlink</b> コマンドは、既存ハイパーリンクタイプのインスタンスであるハイパーリンクオブジェクトを作成します。ハイパーリンクは、同じ <b>VOB</b> 内または異なる <b>VOB</b> 内にある 2 つのオブジェクトを接続します。
トリガ	<b>mktrigger</b> コマンドは、既存トリガタイプのインスタンスであるトリガオブジェクトを作成します。トリガは、1 つまたは複数のエレメントに関連付けることができます。

このスキーマを使用すると、タイプオブジェクトと関連のインスタンスを簡単に管理できます。たとえば、ラベルタイプオブジェクト **BASELEVEL\_4.2** の名前を **BL4.2** に変更すると、関連するすべての既存インスタンスの名前も変更されます。

**メモ:** インスタンスを作成してもタイプオブジェクトのコピーは作成されませんが、新しいオブジェクトが作成される場合があります。たとえば、**mkbranch** コマンドは新しいブランチオブジェクトを作成し、その新しいブランチオブジェクトを既存のブランチタイプオブジェクトに接続するための参照を作成します。

## タイプオブジェクトの範囲

タイプオブジェクトの範囲は、オブジェクトの作成時に定義します。ローカルタイプオブジェクトは、そのオブジェクトが定義されている VOB 内のインスタンスの作成にのみ使用できます。グローバルタイプオブジェクトは、グローバルタイプが定義されている VOB の下にある管理 VOB 階層内のどの VOB 内にインスタンスを作成する場合も使用できます。詳細については、128 ページの「管理 VOB 階層とグローバルタイプ」を参照してください。

**メモ:** トリガタイプオブジェクトは、グローバルタイプとして指定できません。

## 定義済みタイプオブジェクトとユーザー定義のタイプオブジェクト

各 VOB は、定義済みタイプオブジェクトのセットによって作成されます。次の **cleartool** コマンドを使用して、必要に応じて別のタイプオブジェクトを作成することもできます。

- **mkatttype:** 属性タイプオブジェクトを作成または変更します。
- **mkbtype:** ブランチタイプオブジェクトを作成または変更します。
- **mkeltype:** エlementタイプオブジェクトを作成または変更します。
- **mkhltype:** ハイパーリンクタイプオブジェクトを作成または変更します。
- **mklttype:** ラベルタイプオブジェクトを作成または変更します。
- **mktrtype:** トリガタイプオブジェクトを作成または変更します。

各コマンドのリファレンスページには、コマンドに関連付けられたすべての定義済みタイプオブジェクトが一覧表示されます(たとえば、**mkeltype** のリファレンスページには、すべての定義済みElementタイプが一覧表示されます)。ClearCase 管理コンソールまたは **cleartool lstype** コマンドを使用して、特定の VOB で定義されているタイプオブジェクトを一覧表示することもできます。

## Elementタイプの変更

**chtype** を使用して、Elementのタイプを別のタイプに変換できます(たとえば、**file** から **text\_file** へのタイプの変換)。通常は、Elementのバージョンの保存方法を変更するためにElementのタイプを変更します。たとえば、**file** タイプのElementのバージョンが VOB ソースプールの別のデータコンテナに保存されているとします。この場合、Elementのタイプを **text\_file** に変換すると、関連するすべてのバージョンはデルタのセット(バージョン間の差分)として1つのデータコンテナに保存され、その結果、ディスク容量を節約できます。

**メモ:** Elementのすべてのバージョンは、新しいElementタイプの制約に適合している必要があります。たとえば、Elementのタイプを **text\_file** に変換する場合に、バイナリデータを含むバージョンがあると失敗します。ファイルをディレクトリに変換することはできません。また、ディレクトリをファイルに変換することもできません。

## 管理 VOB 階層とグローバル タイプ

---

グローバルに定義されたタイプ オブジェクトを共有できる管理 VOB 階層を作成することで、関連の VOB グループで利用されるタイプ オブジェクトの管理が大幅に簡素化されます。グローバル タイプ オブジェクトを VOB 内に作成すると、AdminVOB ハイパーリンクでリンクされた別の VOB 内にタイプのインスタンスを作成できます。この項では、グローバル タイプと管理 VOB 階層について説明します。

### 管理 VOB 階層

VOB は、次の 2 つの基準を満たしている場合、管理 VOB になります。

- 1 つ以上のグローバル タイプが含まれている。
- 別の VOB からの AdminVOB ハイパーリンクの参照先である。

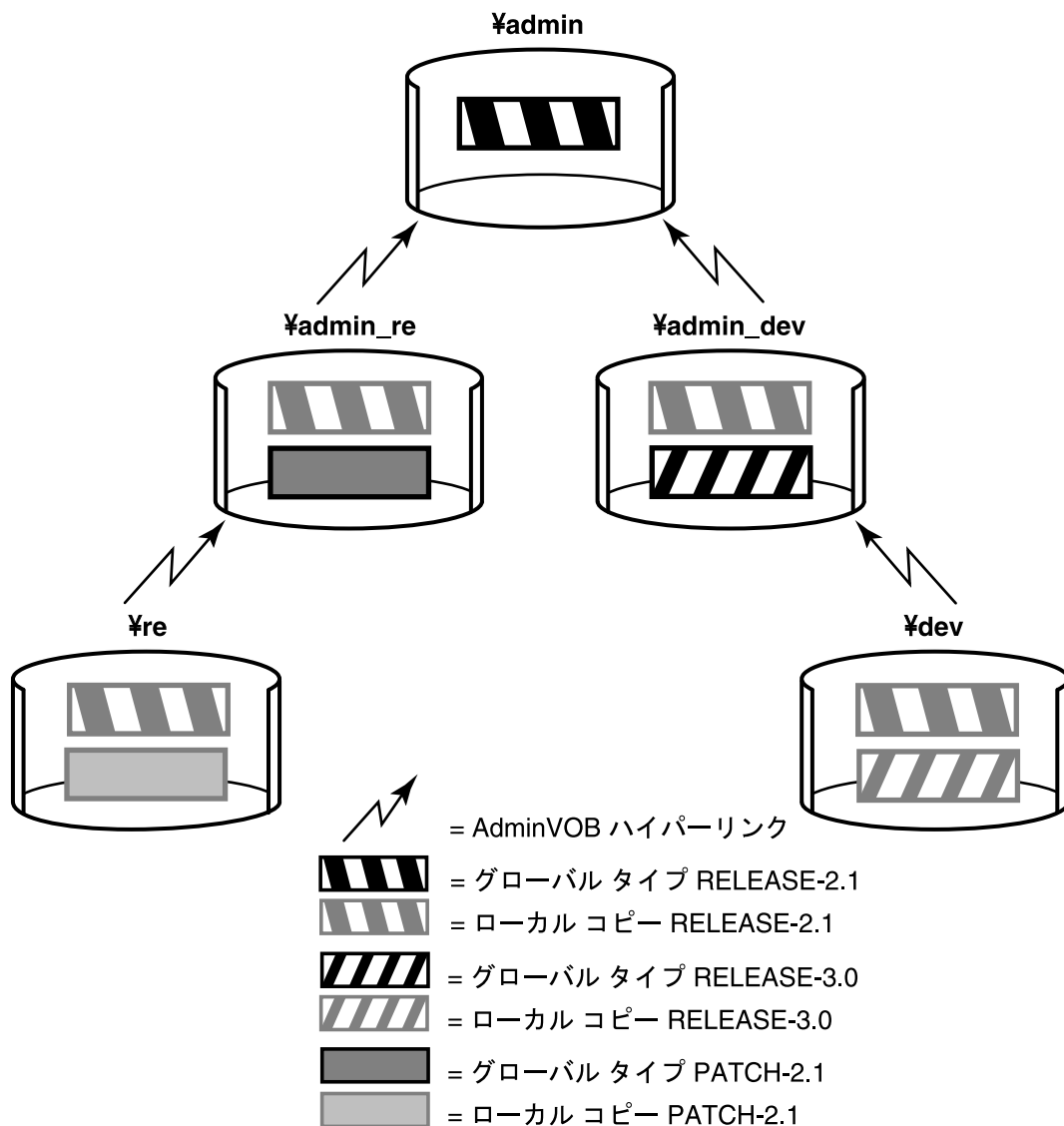
管理 VOB 階層には、1 つ以上の管理 VOB と、AdminVOB ハイパーリンクを持つ 1 つまたは複数の VOB が含まれています。このハイパーリンクは、管理 VOB にリンクしているか、管理 VOB への AdminVOB ハイパーリンクを持つ別の VOB にリンクしています。

管理 VOB 階層内の各 VOB は、1 つの親と 0 または 1 つ以上の子で構成されます。管理 VOB 階層は、これらの制約に違反せず、循環関係を作成しない限り、どんな構造にもなり得ます。図 4 の例では、管理 VOB の ¥admin には階層内のすべての VOB で利用されるグローバル タイプが定義されていますが、下位レベルの管理 VOB の ¥admin\_dev と ¥admin\_re には特定チームのニーズに固有の別のグローバル タイプが定義されています。

すべての UCM コンポーネント VOB は管理 VOB 階層の一部として作成され、プロジェクト、フォルダ、ストリームなどの UCM タイプ オブジェクトが定義されている PVOB への AdminVOB ハイパーリンクが含まれます。複数の PVOB を使用している場合は、各 PVOB には共有 UCM タイプ オブジェクトが定義されている共通 PVOB への AdminVOB ハイパーリンクが含まれている必要があります。

**メモ:** UCM を使用している場合、通常は、この項で説明しているように手動で管理 VOB 階層を作成する必要はありません。

図 4 管理 VOB 階層



VOB を既存の管理 VOB 階層に追加するには、既存の AdminVOB ハイパーリンクを削除し、新しいハイパーリンクを 2 つ追加します。この場合、ローカル コピーとそれに関連付けられたグローバル タイプを結ぶハイパーリンクは壊れないため、既存のタイプ定義には影響ありません

VOB を管理 VOB 階層に追加するには

- 1 新しい VOB を追加する位置にある AdminVOB ハイパーリンクを削除します。たとえば、VOB を ¥admin と ¥admin\_re の間に追加する場合は、**cleartool describe** を使用して AdminVOB ハイパーリンクを表示し、**cleartool rmhlink** を使用して該当のハイパーリンクを削除します。

```
cleartool describe -l vob:¥admin
```

```
...
```

ハイパーリンク :

```
AdminVOB@40@¥admin_re <- vob:¥admin_re
```

```
cleartool rmhlink -c "insert admin VOB" AdminVOB@40@¥admin_re
```

ハイパーリンク "AdminVOB@40@¥admin\_re" を削除しました。

- 2 新しい VOB を階層の上位レベルにリンクします。次のコマンドを使用して、VOB ¥admin\_lb から VOB ¥admin への AdminVOB ハイパーリンクを作成します。

```
cleartool mkhlink -c "link admin_lb to admin" AdminVOB ^
```

```
vob:¥admin_lb vob:¥admin
```

ハイパーリンク "AdminVOB@40@¥admin\_lb" を作成しました。

- 3 手順 1 で階層から削除した VOB を元の階層にリンクし直します。

```
cleartool mkhlink -c "link re to admin_lb" AdminVOB vob:¥re vob:¥admin_lb
```

ハイパーリンク "AdminVOB@40@¥re" を作成しました。

このコマンドにより、VOB ¥re も階層に再接続されます。これは、¥admin\_re へのリンクが変更されていないためです。

## AdminVOB ハイパーリンクの表示

AdminVOB ハイパーリンクを表示するには、ClearCase 管理コンソールまたは **cleartool describe** コマンドを使用します。**describe** コマンドは、VOB から管理 VOB 階層の上位レベルへのすべての AdminVOB ハイパーリンクを表示します。AdminVOB ハイパーリンクを表示する例を次に示します。

- VOB ¥dev の管理 VOB が ¥admin\_dev の場合

```
cleartool describe vob:¥dev
```

```
versioned object base "¥dev"
```

```
...
```

ハイパーリンク :

```
AdminVOB -> vob:¥admin_dev
```

- 管理 VOB `¥admin` に下位レベルの 2 つの VOB からのリンクが含まれている場合

```
cleartool describe vob:¥admin
versioned object base "¥admin"
...
```

ハイパーリンク :

```
AdminVOB <- vob:¥admin_dev
AdminVOB <- vob:¥admin_re
```

- VOB が階層の中央にある場合 (階層内の上位レベルへの `AdminVOB` ハイパーリンクと、下位レベルの VOB からの `AdminVOB` ハイパーリンクがある場合)

```
cleartool describe vob:¥admin_dev
versioned object base "¥admin_dev"
...
```

ハイパーリンク :

```
AdminVOB -> vob:¥admin
AdminVOB <- vob:¥dev
```

ハイパーリンク ID を表示するには、`cleartool describe -long` を使用します。ハイパーリンク ID には、ハイパーリンクが作成されている VOB の VOB タグが含まれています。例を次に示します。

```
cleartool describe -long vob:¥admin_dev
...
ハイパーリンク :
AdminVOB@40@¥admin_dev -> vob:¥admin
AdminVOB@40@¥dev <- vob:¥dev
```

## 管理 VOB 階層の制限

管理 VOB 階層には、次の制限が適用されます。

- VOB には、管理 VOB への `AdminVOB` ハイパーリンクを 1 つだけ含めることができます。`mkhlink` コマンドを使用しても、VOB からの 2 番目の `AdminVOB` へのハイパーリンクを作成することはできません。
- ローカル タイプではグローバル タイプを見えなくする (グローバル タイプと同じタイプと名前を指定する) ことはできません。階層内でグローバル タイプとして定義されているのと同じタイプと名前のローカル タイプを VOB で定義すると、管理 VOB 階層への VOB の追加は失敗します。ローカル タイプを強制的にグローバル タイプのインスタンスにするには、`mkhlink` コマンドに `-acquire` オプションを指定して実行します。詳細については `mkhlink` のリファレンス ページを参照してください。

- 階層内のすべての VOB は、グループとしてバックアップして復元する必要があります。グループ以外で実行した場合、管理 VOB に定義されていないタイプのインスタンスが階層内の一部の VOB で保持されることがあります。これは、タイプが作成される前に管理 VOB をバックアップから復元した場合に発生します。詳細については、225 ページの「関連付けられたデータベースのグループに属するメンバの復元」を参照してください。
- 通常の VOB のタイプ オブジェクト定義の派生元である管理 VOB にアクセスできない場合、そのタイプのインスタンスを通常の VOB 内で作成または操作することはできません。詳細については、「管理 VOB が使用できなくなった場合」を参照してください。

## 管理 VOB が使用できなくなった場合

管理 VOB または PVOB が使用できなくなった場合、階層内の別の VOB で現在使用できないグローバル タイプ定義に基づいてインスタンスを作成しようとすると、次のようなエラーが発生します。

cleartool: エラー : 管理 VOB AdminVOB-tag (VOB-tag) にアクセスできません。

また、使用できない VOB への AdminVOB ハイパーリンクを持つ VOB に対して **cleartool describe** を実行した場合は、ハイパーリンクが表示されないことがあります。

**メモ:** 管理 VOB は、階層のアクティブな部分にするためにマウントする必要はありませんが、階層内に登録し、タグを付ける必要があります。

## 管理 VOB 階層からの VOB の削除

VOB と管理 VOB の関係を解除するには、VOB から管理 VOB への AdminVOB ハイパーリンクとすべての GlobalDefinition ハイパーリンクを削除します。これらを接続しているすべてのハイパーリンクを削除する必要があります。次の項では、**cleartool** コマンドを使用したハイパーリンクの削除方法について説明します。ClearCase 管理コンソールを使用して行うこともできます。

**メモ:** ここで説明する手順は、UCM コンポーネント VOB または PVOB には使用できません。UCM コンポーネント VOB または PVOB を管理 VOB 階層から削除する必要がある場合は、Rational カスタマ サポートまでお問い合わせください。

### AdminVOB ハイパーリンクの削除

AdminVOB ハイパーリンクを削除するには

- 1 AdminVOB ハイパーリンクの名前と ID を特定します。

```
cleartool describe vob:¥dev
versioned object base "¥dev"
...
ハイパーリンク :
AdminVOB@40@¥dev -> vob:¥admin_dev
```

- 2 rmhlink コマンドを使用して、ハイパーリンクを削除します。

```
cleartool rmhlink AdminVOB@40@¥dev
```

ハイパーリンク "AdminVOB@40@¥dev" を削除しました。

## すべての GlobalDefinition ハイパーリンクの削除

VOB 内のローカル コピーを管理 VOB 内のグローバル タイプにリンクするすべての GlobalDefinition ハイパーリンクを削除するには

- 1 グローバルの属性、ブランチ、エレメント、ハイパーリンク、ラベルの各タイプに対するすべてのローカル コピーの名前を特定します。

```
cleartool lstype -local -fmt "%n¥t%[type_scope]p¥n" -kind attype -invob ¥dev
```

```
Tested          ローカル コピー
```

```
Revision        通常
```

```
...
```

```
cleartool lstype -local -fmt "%n¥t%[type_scope]p¥n" -kind brtype -invob ¥dev
```

```
...
```

```
cleartool lstype -local -fmt "%n¥t%[type_scope]p¥n" -kind eltype -invob ¥dev
```

```
...
```

```
cleartool lstype -local -fmt "%n¥t%[type_scope]p¥n" -kind hltype -invob ¥dev
```

```
...
```

```
cleartool lstype -local -fmt "%n¥t%[type_scope]p¥n" -kind lbtype -invob ¥dev
```

```
...
```

- 2 各ローカル コピーについて、ローカル コピーからグローバル タイプへのハイパーリンクの名前と ID を特定します。例を次に示します。

```
cleartool describe -local -long -ahlink GlobalDefinition attype:Tested
```

```
Tested
```

ハイパーリンク :

```
GlobalDefinition@58@¥dev -> attype:Tested@¥admin_dev
```

- 3 rmhlink コマンドを使用して、各ハイパーリンクを削除します。

```
cleartool rmhlink GlobalDefinition@58@¥dev
```

ハイパーリンク "GlobalDefinition@58@¥dev" を削除しました。

## 管理 VOB の削除

rmvob コマンドまたは ClearCase 管理コンソールを使用して管理 VOB を削除すると、関連のハイパーリンクも次のように自動的に削除されます。

- VOB へのすべての AdminVOB ハイパーリンクが削除されます。
- VOB 内のグローバル タイプへのすべての GlobalDefinition ハイパーリンクが削除されます。
- VOB で定義されているグローバル タイプのすべてのローカル コピー (階層内の別の VOB にある) が通常タイプに変換されます。

## グローバル タイプの操作

---

一般に、グローバル タイプまたはグローバル タイプのローカル コピーに対して行われるすべての操作は、グローバル タイプとそのすべてのローカル コピーに適用されます。この項の例では、**cleartool** コマンド行を使用します。**ClearCase** 管理コンソールで [VOB] ノードの [メタデータ] サブノードを使用して、タイプ オブジェクトなどのメタデータを VOB 内で処理することもできます。

### グローバル タイプの作成

**cleartool** のほとんどのタイプ作成コマンド (127 ページの「定義済みタイプ オブジェクトとユーザー定義のタイプ オブジェクト」を参照) では、グローバル タイプ オブジェクトを作成する **-global** オプションをサポートしています。次のコマンドによって、VOB %admin にグローバルなラベル タイプが作成されます。

```
cleartool mklbtype -c "final label for REL6" -global REL6@%admin
ラベル タイプ "REL6" を作成しました。
```

タイプ作成コマンドで **-acquire** オプションを使用しないと、階層内の VOB でタイプ オブジェクトを同じ名前で作成した場合、グローバル タイプを作成できません。新しいグローバル タイプを作成する前に、見えなくなっている可能性のあるローカル タイプの存在を確認し、**-acquire** オプションを使用してこれらのローカル タイプをグローバル タイプのローカル コピーに変換してください。

例を次に示します。

```
cleartool describe -fmt "%n¥t%[type_scope]p¥n" lbtype:V3.2@%dev
V3.2      通常
```

```
cleartool mklbtype -c "Release 3.2" -global -acquire V3.2@%admin
ラベル タイプ "V3.2" を作成しました。
```

```
cleartool describe -local -fmt "%n¥t%[type_scope]p¥n" lbtype:V3.2@%dev
V3.2      ローカル コピー
```

タイプが一致しないと、警告メッセージが表示され、**-acquire** 操作は失敗します。タイプが一致してもロックされている場合は、取得できないことが表示され、別のタイプの処理が進められます。この問題を解決するには、アンロックして、該当のタイプ作成コマンドに **-replace** オプション、**-global** オプション、**-acquire** オプションを指定して再度実行するか、**checkvob -global** コマンドを実行します。

**-replace** オプション、**-global** オプション、**-acquire** オプションを使用して、既存の通常タイプをグローバル タイプに変換することもできます。

## 自動タイプ作成操作

通常、グローバルタイプのローカルインスタンスを作成すると、グローバルタイプのローカルコピーが作成されます。このアクションは、自動タイプ作成操作と呼ばれます。自動タイプ作成操作は、次のような場合に行われます。

- 属性、ブランチ、エレメント、ハイパーリンク、ラベルを作成する操作を実行した場合。グローバルタイプのローカルコピーが作成されます。
- ブランチを作成するチェックアウト操作を実行した場合。自動ブランチ作成規則によってグローバルブランチタイプのローカルコピーが作成されます。
- タイプのローカルコピーに属性やハイパーリンクを関連付ける操作を実行した場合。ローカルコピーが作成されます (ローカルコピーが存在しない場合)。

グローバルタイプにスーパータイプが指定されている場合、自動タイプ作成操作によってスーパータイプのローカルコピーが作成され、関連するタイプ作成トリガが起動されます。また、ローカルコピーの権限と所有権に、グローバルタイプと同じ内容が設定されます。

グローバルラベルタイプのインスタンスを作成する例を次に示します。コマンドの出力には、管理VOBのVOBタグも表示されます。

```
cleartool mklabel -c "Release 6" REL6 ¥dev¥file.c
```

```
Automatically created label type "REL6" from global definition in VOB  
"¥admin".
```

ラベル "REL6" を "¥dev¥file.c" バージョン "/main/rel6\_main/31" 上に作成しました。

**メモ:** トリガタイプを作成して、グローバルタイプを組み込みアクションの引数 (引数の `-mklabel` や `-mkattr` など) として指定しても、ClearCase ではグローバルラベルタイプのローカルコピーは作成されません。これは、組み込みアクションではトリガのカスケードは発生しないためです。したがって、グローバルタイプのローカルコピーが保存されていないVOBにこのようなトリガを作成しようとすると、`mktrtype` コマンドの実行は失敗します。

## グローバルタイプの説明

デフォルトでは、`describe` コマンドを実行すると、指定したオブジェクトセレクタのグローバルタイプの説明が表示されます。このコマンドは、該当のタイプのローカルコピーが保存されていないVOBのコンテキストに入力できます。ローカルコピーの説明を表示するには、`-local` オプションを指定します。

次のコマンドは、グローバルラベルタイプ REL6 の説明を表示します。

```
cleartool describe -long lbtype:REL6@¥dev
```

```
label type "REL6"
```

```
28-Jul-99.14:00:26 は Suzanne Gets (smg.user@neon) によって作成されました  
"final label for REL6"
```

```
所有者: smg
```

```
グループ: user
```

```
範囲: global
制約: one version per element
ハイパーリンク:
GlobalDefinition@47@¥dev <- lbtype:REL6@¥dev
```

次のコマンドは、グローバル ラベル タイプ REL6 のローカル コピーの説明を表示します。

```
cleartool describe -local -long lbtype:REL6@¥dev
label type "REL6"
28-Jul-99.14:23:45 は Suzanne Gets (smg.user@neon) によって作成されました
"Automatically created label type from global definition in VOB
"¥admin".
所有者: smg
グループ: user
範囲: この VOB (グローバル タイプのローカル コピー)
制約: one version per element
ハイパーリンク:
GlobalDefinition@47@¥dev -> lbtype:REL6@¥admin
```

describe コマンドで -local オプションを指定した場合に、ローカル コピーが存在しないと、次のようなエラーが表示されます。

```
cleartool describe -local lbtype:NOLOCAL@¥dev
cleartool: エラー: VOB オブジェクトではありません: "lbtype:NOLOCAL@¥dev"
```

## グローバル タイプの表示

デフォルトでは、lstype コマンドを実行すると、ローカル コピーに関連付けられたグローバル タイプが一覧表示されます。-invob オプションを使用して、グローバル タイプが定義されていない VOB を指定した場合でも、表示されます。出力には、指定した VOB に該当のタイプのローカル コピーが現在保存されていない場合でも、管理 VOB 階層内のこの VOB より上位にあるすべての管理 VOB のグローバル タイプが一覧表示されます。特定の VOB で定義されている (またはコピーが作成されている) タイプのみを表示するには、-local オプションを使用します。

次のコマンドは、VOB ¥dev 内のすべてのラベル タイプを表示します。管理 VOB 階層内にあるすべての管理 VOB のグローバル ラベル タイプも含まれます。

```
cleartool lstype -fmt "%n¥t%[type_scope]p¥n" -kind lbtype -invob ¥dev
BACKSTOP          通常
CHECKEDOUT         通常
LABEL1            グローバル
LATEST            通常
REL6              グローバル
```

次のコマンドは、一般タイプと、グローバルタイプのローカルコピーを表示します (指定した VOB が管理 VOB の場合は、グローバルタイプも表示されます)。

```
cleartool lstype -local -fmt "%n¥t%[type_scope]p¥n" -kind lbtype -invob ¥dev
```

```
BACKSTOP          通常
```

```
CHECKEDOUT        通常
```

```
LATEST            通常
```

```
REL6              ローカル コピー
```

## グローバルタイプの履歴表示

デフォルトでは、`lshistory` コマンドを実行すると、指定したオブジェクトセレクタのグローバルタイプの履歴が一覧表示されます。指定した VOB に該当のタイプのローカルコピーが保存されていない場合でも、表示されます。ローカルコピーの履歴を表示するには、`-local` オプションを使用します。`-all` または `-avobs` を指定すると、暗黙的に `-local` を指定したことになります。

次のコマンドは、グローバルラベルタイプの履歴を表示します。

```
cleartool lshistory -minor lbtype:REL6@¥dev
```

```
28-Jul.14:00 smg          make hyperlink "GlobalDefinition" on label  
type "REL6"
```

```
"Attached hyperlink "GlobalDefinition@47@¥dev".
```

```
Automatically created label type from global definition in VOB  
"¥admin".
```

```
28-Jul.13:57 smg          create label type "REL6"
```

次のコマンドは、グローバルラベルタイプのローカルコピーの履歴を表示します。

```
cleartool lshistory -local -minor lbtype:REL6@¥dev
```

```
28-Jul.14:00 smg          make hyperlink "GlobalDefinition" on label  
type "REL6"
```

```
"Attached hyperlink "GlobalDefinition@47@¥dev".
```

```
Automatically created label type from global definition in VOB  
"¥admin".
```

```
28-Jul.14:00 smg          create label type "REL6"
```

```
"Automatically created label type from global definition in VOB  
"¥admin".
```

## グローバルタイプの保護情報の変更

グローバルタイプまたはそのローカルコピーの保護情報を変更すると、グローバルタイプとそのすべてのローカルコピーの保護情報が変更されます。グローバルタイプの保護情報を変更するには、権限が必要です。VOB にタイプのローカルコピーが保存されていない場合でも、タイプが定義されている階層内の VOB のコンテキストにコマンドを入力できます。

次の例では、**LABEL1** というラベル タイプの所有者を **jtg** に変更します。**describe** コマンドを実行すると、このグローバル タイプのすべてのローカル コピーの保護情報が変更されたことが表示されます。

```
cleartool protect -chown jtg lbtype:LABEL1@%dev
```

"LABEL1" の保護を変更しました。

```
cleartool describe -local lbtype:LABEL1@%re
```

```
label type "LABEL1"
```

```
...
```

```
所有者 : jtg
```

```
グループ : user
```

```
範囲 : この VOB (グローバル タイプのローカル コピー)
```

```
...
```

1 つまたは複数のローカル コピーの保護情報を変更できない場合、操作は失敗してグローバル タイプの保護情報は変更されません。結果として、グローバル タイプとローカル コピーの間に不整合が発生します。その場合は、保護情報を変更できない問題を解決してから、**protect** コマンドを再度実行する必要があります。

## グローバル タイプのロックとアンロック

グローバル タイプまたはそのローカル コピーをロックまたはアンロックすると、すべてのローカル コピーがロックまたはアンロックされます。**describe** コマンドではローカル コピーがロックされていることは表示されませんが、ローカル コピーへのアクセス チェックによってグローバル タイプのロックが確認されます。

たとえば、次のコマンドは **REL6** というグローバル ラベル タイプとそのローカル コピーをロックします。

```
cleartool lock -c "freeze" lbtype:REL6@%dev
```

```
label type "REL6" をロックしました。
```

ロックされているグローバル タイプの新しいインスタンスを作成しようとする、次のように失敗します。

```
cleartool mklabel -c "last version" REL6 %re%tests.txt
```

```
cleartool: エラー: label type "REL6" のロックによって、操作 "make hyperlink" を実行できません。
```

```
cleartool: エラー: ラベル "REL6" を "%re%tests.txt" バージョン "/main/5" 上に作成できません。
```

指定したタイプのローカル コピーが保存されていない **VOB** でタイプをロックしようとする、**ClearCase** によってそのグローバル タイプが管理 **VOB** 階層で検索されます。

デフォルトでは、**lslock** はグローバル タイプのロック状態を表示します。ローカル コピーのロック状態を表示するには、**-local** オプションを使用します。

## グローバル タイプのコピー

`cptype` コマンドでグローバル タイプを別の VOB に同じ名前でコピーするときに、次のいずれかの条件が満たされる場合は、コピーがグローバル タイプとして作成されます。

- このグローバル タイプのコピー元 VOB とコピー先 VOB が、同じ管理 VOB 階層に属していること (この場合、コピーもこの管理 VOB 階層をポイントします)。
- 元のグローバル タイプが、コピー先 VOB (`cptype` でローカル コピーを作成した VOB) の管理 VOB である VOB に保存されていること。

これらの条件が満たされない場合は、通常タイプとして作成されます。

## グローバル タイプ名の変更

グローバル タイプ名を変更すると、そのタイプのすべてのローカル タイプ名も変更されます。また、グローバル タイプのローカル コピーの名前を変更すると、そのローカル コピーだけでなく、ほかのすべてのローカル コピーとグローバル タイプ自体の名前も変わります。指定したタイプのローカル コピーが保存されていない VOB でタイプ名を変更しようとする、`rename` によってそのグローバル タイプが管理 VOB 階層で検索されます。

すべてのローカル コピー名が変更されてから、グローバル タイプ名が変更されます。名前を変更できないローカル コピーがあると、コマンドによる名前の変更が失敗し、グローバル タイプ名は変更されません。結果として、グローバル タイプとローカル コピーの間に不整合が発生します。その場合は、問題を解決してから、`rename` コマンドを再度実行する必要があります。

詳細については、`rename` のリファレンス ページを参照してください。

## タイプの範囲の変更

既存の通常タイプをグローバル タイプに変換するには、`mkobjecttype` コマンドに、`-replace` オプション、`-global` オプション、`-acquire` オプションを指定して実行します。これにより、通常タイプはグローバル タイプに変換され、階層内の別の VOB にある同じ名前の通常タイプもそのタイプのローカル コピーに変換されます。例を次に示します。

- 1 VOB と管理 VOB に、IDENT という同じ名前の通常ラベル タイプがあります。

```
cleartool describe lbtype:IDENT@¥admin
```

```
label type "IDENT"
```

```
02-Aug-99.15:32:52 は Suzanne Gets (smg.user@neon) によって作成されました
```

```
所有者: smg
```

```
グループ: user
```

```
範囲: この VOB (通常タイプ)
```

```
制約: one version per element
```

```
cleartool describe lbtype:IDENT@¥dev
```

```
label type "IDENT"
```

01-Aug-99.15:33:00 は Suzanne Gets (smg.user@neon) によって作成されました  
所有者: smg  
グループ: user  
範囲: この VOB (通常タイプ)  
制約: one version per element

- 2 管理 VOB の IDENT ラベル タイプをグローバル タイプに変換します。

```
cleartool mklbtype -replace -global -acquire IDENT@¥admin
```

ラベル タイプ "IDENT" の定義を置換しました。

- 3 describe コマンドの出力には、管理 VOB 内の IDENT ラベル タイプがグローバルに変換され、その他の IDENT ラベル タイプがグローバル タイプのローカル コピーに変換されたことが示されます。

```
cleartool describe -local lbtype:IDENT@¥admin lbtype:IDENT@¥dev
```

```
label type "IDENT"
```

02-Aug-99.15:32:52 は Suzanne Gets (smg.user@neon) によって作成されました

所有者: smg

グループ: user

範囲: global

制約: one version per element

ハイパーリンク:

```
GlobalDefinition <- lbtype:IDENT@¥dev
```

```
label type "IDENT"
```

02-Aug-99.15:32:52 は Suzanne Gets (smg.user@neon) によって作成されました

所有者: smg

グループ: user

範囲: この VOB (グローバル タイプのローカル コピー)

制約: one version per element

ハイパーリンク:

```
GlobalDefinition <- lbtype:IDENT@¥dev
```

既存のグローバル タイプを通常タイプに変換するには、mkobjecttype コマンドに **-replace -ordinary** オプションを指定します。これにより、指定したグローバル タイプとすべてのローカル コピーが通常タイプに変換されます。このコマンドには、グローバル タイプを指定する必要があります。グローバル タイプのローカル コピーを指定することはできません。たとえば、VOB ¥admin で定義されているグローバル エlement タイプ **doc\_file** を通常タイプに変換するには、次のコマンドを実行します。

```
cleartool mkeltype -replace -ordinary -nc eltype:doc_file@¥admin
```

タイプが既に作成されている場合、または権限を持つユーザーの場合は、**-replace** オプションを指定してタイプの範囲を変更することもできます。

## グローバル タイプの削除

グローバル タイプまたはグローバル タイプのコピーを削除すると、すべてのローカル コピーとグローバル タイプ自体が削除されます。**rmtype** コマンドを実行すると、該当のグローバル タイプのローカル コピーが保存されているすべての VOB が一覧表示され、削除するかどうかを確認するメッセージが表示されます。**rmtype** コマンドでグローバル タイプを削除する場合は、**-rmall** オプションを指定する必要があります。

例を次に示します。

```
cleartool rmtype -nc lotype:LABEL1@¥dev
cleartool: エラー : タイプ "LABEL1" のラベルがあります。
cleartool: エラー : ラベル タイプ "LABEL1" を削除できません。
cleartool rmtype -nc -rmall lotype:LABEL1@¥dev
label type "LABEL1" のインスタンスが ¥re に 1 個あります。
label type "LABEL1" のインスタンスが ¥dev に 1 個あります。
label type "LABEL1" のすべてのインスタンスを削除しますか? [no] yes
label type "LABEL1" を削除しました。
```

**メモ:** 指定したグローバル タイプのローカル コピーを保存していない VOB で **rmtype** コマンドを実行すると、該当のグローバル タイプが管理 VOB 階層で検索されます。すべてのローカル コピーが先に削除されてから、グローバル タイプが削除されます。削除できないローカル コピーがあると、コマンドによる削除が失敗し、グローバル タイプは削除されません。その場合は、ローカル コピーを削除できない問題を解決してから、**rmtype** コマンドを再度実行する必要があります。

タイプの削除の詳細については、**rmtype** のリファレンス ページを参照してください。

## グローバル タイプのクリーン アップ

**checkvob -global** を使用すると、グローバル タイプの不整合の確認と修正ができます。詳細については、254 ページの「**checkvob** を使用した破損ハイパーリンクの検出と修正」を参照してください。

## 管理 VOB 階層のレプリカの作成

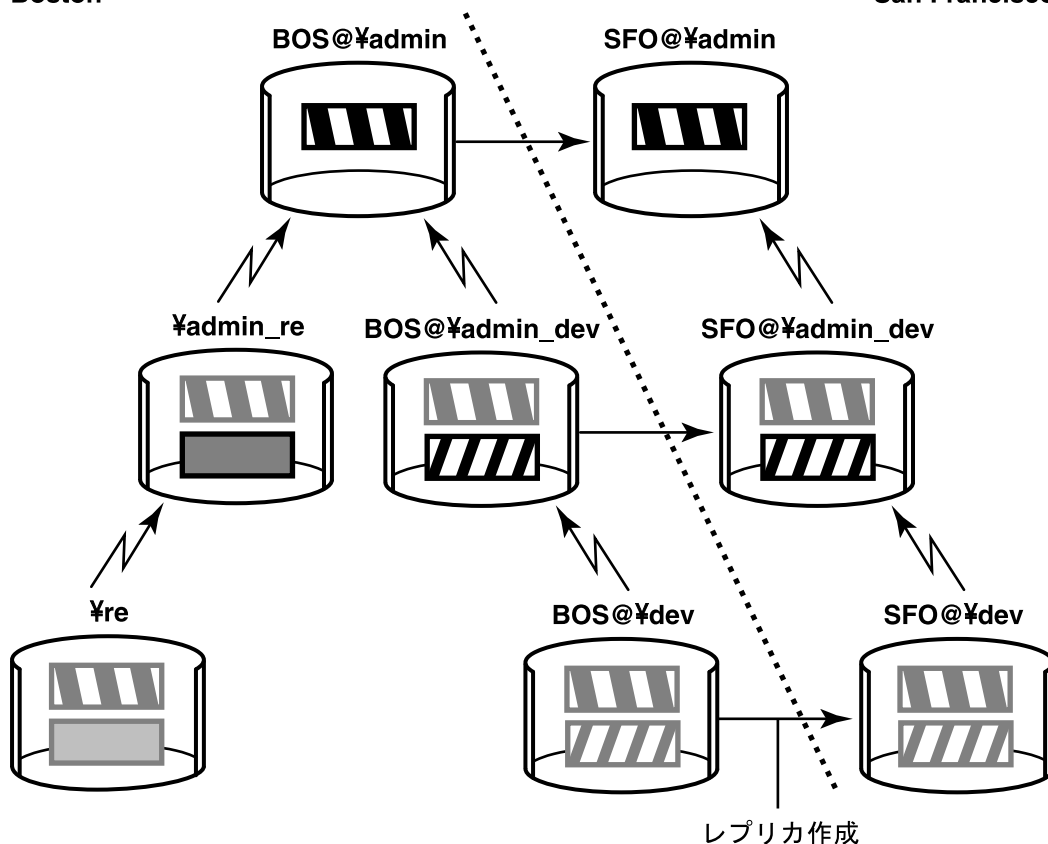
---

Rational ClearCase MultiSite 環境で管理 VOB 階層 (UCM によって作成された階層を含む) を使用する場合、レプリカの作成には特別な注意が必要です。管理 VOB 階層の一部である VOB のレプリカが作成されている場合は、階層内のその他のメンバ (少なくとも階層内の上位のメンバ) のレプリカも作成する必要があります。図 5 の例では、¥re に ¥admin への AdminVOB ハイパーリンクがあり、¥re のレプリカが作成されているため、¥re のレプリカが保存されているすべてのサイトには ¥admin のレプリカを作成する必要があります。

図 5 管理 VOB 階層のレプリカ作成要件

Boston

San Francisco



管理 VOB 階層の一部である VOB のレプリカを作成する場合、`mkreplica -export` コマンドを実行すると、レプリカを作成する VOB より階層内で上位にあるすべての VOB についてもレプリカを作成する必要があることを知らせるメッセージが表示されます。出力には、レプリカの対象となる VOB が一覧表示されます。コマンドはこれらの管理 VOB のレプリカが作成されているかどうかをチェックしないため、既に作成してある場合はこのメッセージを無視してください。

VOB 内のローカルタイプオブジェクトは管理 VOB 階層内のグローバルタイプオブジェクトにリンクされているため、階層内のすべてのメンバを同時に同期することをお勧めします。そうしないと、ユーザーがタイプオブジェクトにアクセスする際に問題が発生する可能性があります。

**メモ:** 図 5 では、VOB ¥admin\_re に ¥admin への AdminVOB ハイパーリンクがありますが、レプリカは作成されていません。このような場合 (管理 VOB のレプリカは作成されているが、階層内で下位にある 1 つまたは複数のメンバのレプリカは作成されていない場合) は、cleartool タイプ作成コマンドを実行すると、次のような形式のメッセージが表示されます。

エラー : レジストリ内でオブジェクト ID "<VOB-oid>" の VOB のレプリカが見つかりません。

エラー : オブジェクト ID を使用してバージョン付きオブジェクト ベースを検索できません : "<VOB-oid>"

メッセージに示されている VOB のレプリカを作成しようとしていなかった場合や、指定したタイプの作成がこれらのエラーによって妨げられない場合は、これらのエラーは重要ではありません。

## グローバル タイプとマスタースhip

グローバル タイプのローカル コピーを作成できるのは、現在のサイトでグローバル タイプがレプリカによってマスタ登録されている場合のみです。グローバル タイプが現在のサイトでマスタ登録されていない場合は、タイプのローカル コピーが既にレプリカに含まれている場合に限り、タイプのインスタンスをローカルのレプリカ内に作成できます。この制限は、タイプのインスタンスを関連付けるオブジェクトが現在のレプリカによってマスタ登録される場合にも適用されます。この制限により、複数サイトで競合するタイプが作成されないようにします。

現在のサイトの VOB にタイプのローカル コピーが保存されていない場合は、次の処理を行う必要があります。

- 1 タイプをマスタ登録するサイトでローカル コピーを作成します。
- 2 マスタ サイトのレプリカから現在のサイトのレプリカに更新パケットをエクスポートします。
- 3 現在のサイトにパケットをインポートします。

たとえば (図 5 を参照)、現在のサイト (San Francisco) にいるエンジニアが VOB ¥dev のバージョンに **RELEASE-2.1** という名前を付けるとします。この場合、ラベル タイプは別のサイトにマスタ登録されているグローバル タイプであり、¥dev にローカル コピーが存在しないため、コマンドの実行は失敗します。

```
cleartool mklabel -nc RELEASE-2.1 ¥dev¥file.txt
```

cleartool: エラー : コピー タイプを使用するには、タイプが元のレプリカ "SFO" にマスタ登録される必要があります。

cleartool: エラー : ラベル "RELEASE-2.1" を "¥dev¥file.txt" バージョン  
"/main/3" 上に作成できません。

タイプのローカル コピーを現在のサイトのレプリカに作成するには

- 1 グローバル タイプが定義されている管理 VOB のタグを特定します。

```
cleartool describe vob:¥dev
versioned object base "¥dev"
...
ハイパーリンク :
AdminVOB -> vob:¥admin
```

- 2 タイプをマスター登録している管理 VOB レプリカを特定します。次のコマンドは、レプリカが作成されている VOB 内のグローバル タイプの説明を表示します。タイプのマスター レプリカはコマンドを入力したレプリカではなく別の VOB ファミリにあるため、出力には、レプリカ名のほかに、マスター レプリカの VOB タグがマスター レプリカ行に表示されます。

```
cleartool describe -long lbtype:RELEASE-2.1@¥admin
label type "RELEASE-2.1"
03-Aug-02.12:29:00 は Pete Sharon (pds.user@argon) によって作成されました
マスター レプリカ: BOS@¥admin
インスタンス マスターシップ: 共有されています
所有者: pds
グループ: user
範囲: global
制約: one version per branch
ハイパーリンク:
GlobalDefinition@43@¥admin <- lbtype:RELEASE-2.1@¥admin
```

- 3 タイプがマスター登録されているサイトで、タイプのローカル コピーを作成します。

```
cleartool cptype -c "forced local copy" lbtype:RELEASE-2.1@¥admin ¥
lbtype:RELEASE-2.1@¥dev
タイプ "RELEASE-2.1" をコピーしました。
```

- 4 タイプがマスター登録されているサイトで、更新パッケージを現在のサイトのレプリカにエクスポートします。

```
multitool syncreplica -export -fship SFO@¥dev
```

- 5 現在のサイトで、更新パッケージをインポートします。

```
multitool syncreplica -import -receive
```

パッケージをインポートしたら、ラベルを作成できます。

```
cleartool mklabel -nc RELEASE-2.1 ¥dev¥file.txt
```

ラベル "RELEASE-2.1" を "¥dev¥file.txt" バージョン "/main/3" 上に作成しました。

## グローバル タイプのマスターシップの変更

グローバル タイプのマスターシップを変更しても、ローカル コピーのマスターシップは変更されません。同様に、ローカル コピーのマスターシップを変更しても、グローバル タイプまたはその他のローカル コピーのマスターシップは変更されません。

たとえば、グローバル ラベル タイプ RELEASE-3.0 が San Francisco レプリカによって VOB ファミリ ¥admin\_dev にマスター登録されており、VOB ¥dev のローカル コピーが San Francisco レプリカによって VOB ファミリ ¥dev にマスター登録されているとします。

```
cleartool describe -fmt "%n¥n  %[master]p¥n  %[type_scope]p¥n"
lotype:RELEASE-3.2@¥admin_dev
RELEASE-3.2
  SFO@¥admin_dev
  グローバル
```

```
cleartool describe -local -fmt "%n¥n  %[master]p¥n  %[type_scope]p¥n" ¥
lotype:RELEASE-3.2@¥dev
RELEASE-3.2
  SFO@¥dev
  ローカル コピー
```

グローバル タイプのマスターシップを BOS レプリカに転送しても、ローカル コピーのマスターシップは変更されません。

```
multitool chmaster BOS@¥admin_dev lotype:RELEASE-3.2@¥admin_dev
label type "RELEASE-3.2" のマスターシップを "BOS@¥admin_dev" に変更しました。

cleartool describe -fmt "%n¥n  %[master]p¥n  %[type_scope]p¥n" ¥
lotype:RELEASE-3.2@¥admin_dev
RELEASE-3.2
  BOS@¥admin_dev
  グローバル
```

```
cleartool describe -local -fmt "%n¥n  %[master]p¥n  %[type_scope]p¥n" ¥
lotype:RELEASE-3.2@¥dev
RELEASE-3.2
  SFO@¥dev
  ローカル コピー
```

指定したタイプのローカル コピーが保存されていない VOB で **chmaster** コマンドを実行した場合、コマンドの実行は失敗し、「タイプが見つかりません」というメッセージが表示されます。例を次に示します。

```
multitool chmaster BOS@¥admin lotype:DOC_SOURCE@¥dev

multitool: エラー : ラベル タイプが見つかりません : "DOC_SOURCE"
```

マスターシップの詳細については、**chmaster** のリファレンス ページと『Rational ClearCase MultiSite 管理ガイド』を参照してください。



# VOB の移動と VOB データの再配置

# 8

Rational ClearCase には、VOB 間でデータやメタデータを移動したり、ディスクやホスト間で VOB 全体を移動したりするためのツールが用意されています。VOB ディレクトリの再編成、データ記憶領域の再割り当て、各サーバーへの負荷分散の調整などを行う場合に、これらのツールを使用します。

## 別の VOB へのエレメントの再配置

---

`cleartool relocate` コマンドは、ディレクトリ エレメントとその中に含まれるエレメントを VOB 間で移動するコマンドです。VOB は同じサーバー ホスト上にも異なるサーバー ホスト上にも配置できますが、同じレジストリ リージョンにタグが付いている必要があります。コンポーネント アーキテクチャや組織構造の変更を反映するために VOB ディレクトリを再編成する場合、VOB からグループ エレメントを移動してサイズを縮小する場合、VOB サーバー間で負荷を適切に分散する場合に、`relocate` コマンドによる操作を行ってください。

**メモ:** UCM コンポーネントの VOB または PVOB では、`relocate` コマンドを使用できません。`relocate` による操作を実行する前に、`relocate` のリファレンス ページを参照してください。

### relocate の役割

`relocate` コマンドは、エレメントを VOB (ソース VOB) から別の VOB (ターゲット VOB) に移動するコマンドです。多くの場合、ターゲット VOB はこの目的のために作成されます。通常、`relocate` コマンドは、次のすべての操作を実行します。

- ある VOB から別の VOB へのエレメントの移動。各エレメントでは、次のデータとメタデータが移動の対象になります。
  - エレメントのデータ コンテナ
  - エレメントのイベント履歴 (特定の重要度の低いイベントは消失)
  - エレメントに関連付けられている双方向のハイパーリンク
  - 関連する VOB のデータベースのレコード
  - バージョン付き派生オブジェクトに関連付けられている構成レコード
- ソース VOB のネームスペースの保持。これは、ターゲット VOB への適切なシンボリックリンクを作成することによって行います。

- 必要に応じて、移動したエレメントに関連付けられているメタデータのタイプ (ラベル、属性、エレメント、トリガ、ハイパーリンクのタイプ) をターゲット VOB にコピー。
- 再配置される各エレメント (ソース VOB 内のエレメントの削除イベントやターゲット VOB 内の再配置イベントなど) の再配置イベント履歴の作成。
- 再配置されるディレクトリ内に含まれる、バージョンなしの DO と DO の構成レコードの削除。
- 活動内容のログの作成。デフォルトでは、**relocate** を実行するディレクトリの **relocate.log.date-time** に保存します。

**relocate** は、以下については実行しません。

- ソース VOB またはターゲット VOB のいずれかが UCM コンポーネントの VOB または PVOB である場合の、エレメントの再配置。
- 再配置されたディレクトリ内に格納されている、ビュープライベート ファイルとバージョンなしの DO の移動。
- 同じ VOB 内での新しい場所へのエレメントの移動。この場合は、**cleartool mv** を使用します。

## エレメントの再配置の前に実行する処理

エレメントの再配置の前に、次の操作を実行します。

- 1 (MultiSite のみ) ほかのレプリカの管理者との調整を行います。再配置するエレメントは、すべてのレプリカ サイトで非アクティブになっている必要があります。**relocate** リファレンス ページのレプリカ VOB に関する項を参照してください。
- 2 (動的ビューのみ) 再配置する予定のディレクトリから、ビュープライベート ファイルを移動します。移動しなかったビュープライベート ファイルは、そのまま取り残されます。取り残されたファイルは、**recoverview** コマンドを実行することによって復元できます。ただし、**relocate** を実行する前に移動する方が簡単です。
- 3 再配置する予定のディレクトリに含まれるすべてのチェックアウトを解決します。再配置する対象のディレクトリにチェックアウト済みのバージョンが検出されると、**relocate** コマンドは終了します。チェックアウト済みのバージョンを検索して解決 (チェックインまたはキャンセル) するには、ClearCase 管理コンソールで [VOB] ノードの [参照されているビュー] サブノードを使用します。**cleartool lsdo** コマンドと **lscheckout** コマンドも使用できます。
- 4 **rmelem** トリガを削除します。**-update** モードで実行していない場合、**relocate** は **rmelem** を使用して、ソース VOB からエレメントを削除します。トリガが **rmelem** に関連付けられている場合、この処理は失敗し、各エレメントを手動で削除する必要があります (150 ページの「ソース VOB からのエレメントの削除中に発生するエラー」を参照)。

- 5 作業ビューを確立します。移動の行われるブランチ (通常は `main`) が構成仕様によって選択されているビューを使用します。

`relocate` を実行するビューは、ソースとターゲット両方の VOB エlement をチェックアウトできる必要があります。このため、構成仕様に `CHECKEDOUT` 規則が設定されている必要があります。`relocate` の実行後、このビュー、または同じ構成仕様が設定されているビューを使用して、クリーンアップを実行する必要があります。詳細については、276 ページの「Element を再配置した後の既存のビューの問題」を参照してください。

- 6 `relocate` をテスト モードで実行します。目的の `relocate` コマンドを実行してその出力をモニターしますが、次のプロンプトに「`no`」で応答することにより、Element の移動をキャンセルすることができます。

これらのオブジェクトを再配置しますか? [no]

## ボーダーラインのElementの処理

Element が複数の VOB ディレクトリに登録されている場合 (通常、`cleartool mv` を実行してElement を移動したため)、そのElement が登録されているディレクトリが、`relocate` を実行したビューによって選択されたElement のセットに表示されていないと、`relocate` はそのElement をボーダーラインのElement として分類します。ボーダーラインのElement を検索するには、`-qall` オプションを指定して `relocate` を実行します。

`relocate -qall` によってボーダーラインのElement が検出されると、次のような出力が生成されます。

Element "`main.c`" は、ディレクトリ "`¥sources¥libs`" の "`main.c`" として選択ツリーの外部に、ディレクトリ "`¥sources¥proto¥libs`" の "`main.c`" として選択ツリーの内部に配置されています。

Element を再配置しますか? [no]

選択セットを展開してすべてのバージョンのElement を再配置するか、プロンプトへのデフォルトの応答 (`no`) を受け入れてElement をそのままにすることをお勧めします。ボーダーラインのElement を再配置した場合、再配置されなかったバージョンのElement を参照するビューは、ボーダーラインのElement を表示できません。この問題は、276 ページの「Element を再配置した後の既存のビューの問題」の説明に従って手動で修正できます。

## 再配置操作で発生するエラーへの対処

再配置操作中にエラーが発生する主な原因を次に示します。

- 再配置対象のディレクトリ内のチェックアウト済みファイル
- ターゲット VOB 内のロックされたタイプ オブジェクト
- ソース VOB 内の `rmelem` 上のトリガ

ほとんどの場合、報告されたエラーを修復し、同じコマンド行で **relocate** を再開できます。操作はエラーが発生した場所から継続されます。ソース **VOB** からエレメントを削除しているときに発生したエラーには、手動で対応する必要があります。

## ロックされたタイプまたはチェックアウト済みのファイルに関連するエラー

ロックされたタイプまたはチェックアウト済みファイルが原因でエラーが発生した後は、**relocate** を再開できます。このようなエラーが発生した場合、次の手順に従います。

- 1 停止して問題点を修復します。たとえば、**relocate** がターゲット **VOB** 内のロックされたタイプをレポートした場合、これをアンロックします。再配置セット内のチェックアウト済みバージョンが報告された場合は、チェックアウトを解決します。
- 2 **relocate** を再実行します。同一のコマンド行で **relocate** を呼び出すと、以下の条件を満たしている限り、中断した時点から処理が再開されます。
  - **relocate** によって自動的に設定されたソース **VOB** エレメントのロックを解除しないこと
  - 移動中のエレメント (新しいチェックイン バージョンや新しいラベルなど) をユーザーが修正しないこと
  - (-qall オプションを指定した場合のみ) オブジェクトを再配置するかどうかを確認するメッセージに対して、すべて同じように応答すること

上記の条件のいずれかが満たされていないと、**relocate** の処理は最初から再開されます。

## ソース **VOB** からのエレメントの削除中に発生するエラー

ターゲット **VOB** 内にエレメントを再作成した後で、**relocate** はソース **VOB** からエレメントを削除します (-update オプションを指定しなかった場合)。この処理でエラーが発生した場合は (通常は **rmelem** 上のトリガが原因)、次の手順に従って手動でエレメントを削除する必要があります。

- 1 ログ ファイルからエレメントの **OID** を見つけます。再配置ログ ファイルを調べて、**relocate** で削除できなかったエレメントのオブジェクト ID (**OID**) が記述された行を見つめます。
- 2 エレメントを削除します。ログ ファイルにレポートされた各エレメントごとに次のコマンドを実行します。

```
cleartool rmelem oid:OID-reported-in-log-file
```

- 3 必要に応じて、**relocate** でエレメントを削除できない原因となっていたトリガを削除します。

このようなエラーをなくすには、エレメントを再配置する前にエレメント上の **rmelem** トリガを削除します。

## エレメントを再配置した後

`relocate` はシンボリック リンクを残して VOB ネームスペースの整合性を維持しますが、既存のビュー、ビルド スクリプト、トリガ、その他の再配置されたエレメントへのアクセスが必要なツールを常にチェックしてください。シンボリック リンクを使用せずに、新しい場所の再配置されたエレメントにアクセスするように、これらのビューとツールを更新します。

**メモ:** バージョンなしの DO は再配置されずに、`rmdo` を実行した場合のように削除されるため、再ビルドする必要があります。エレメントを再配置すると、依存関係としてエレメントを含んでいる DO は再ビルドされます。

エレメントの再配置後に発生する問題のトラブルシューティングについては、276 ページの「エレメントを再配置した後の既存のビューの問題」を参照してください。

## VOB の移動

---

一般的なファイル システム コピー ユーティリティを使用して、VOB を別の場所にコピーすることはできません。VOB データの一貫性と、VOB 記憶ディレクトリに対する権限と所有権を維持するには、特別な手順に従う必要があります。この章の以降の部分では、VOB の移動手順について説明します。次のような場合について説明します。

- VOB を Windows 上の別のディスク パーティション、または同じドメイン内の別の Windows ホストに移動する
- VOB を異なるドメイン内の Windows ホストに移動する
- VOB を UNIX 上の別のディスク パーティション、または同じアーキテクチャ (バイナリデータ形式) の別の UNIX ホストに移動する
- VOB を異なるアーキテクチャの UNIX ホストに移動する
- VOB を Windows ホストから UNIX ホストに移動する
- VOB を UNIX ホストから Windows ホストに移動する

この章では、レプリカ VOB を移動する際の特記事項についても説明します。

**注意:** この章に記載されている手順に従ってレプリカ VOB を移動しないと、レプリカの整合性が失われたり、データが失われたりする可能性があります。

## VOB を移動する際に実行する重要な処理

---

この項に記載されている各処理はそれぞれ異なりますが、VOB の移動時には、常に次の点に注意してください。

- VOB を UNIX と Windows 間、または Windows ドメイン間で移動する場合以外は、ディレクトリをコピーしたときに、VOB 記憶ディレクトリの所有権とアクセス コントロール情報が維持されていることを確認してください。大半のファイル システム コピー ユーティリティ (特に Windows 用のユーティリティ) は、この情報を維持しません。VOB の移動手順のコピー処理中にこの情報が変更された場合、VOB 記憶ディレクトリの保護を修復するまで、VOB は新しい場所で使用できません。ClearCase では、VOB 記憶ディレクトリに対する損傷したファイル システム保護を修復するプログラムが用意されています。詳細については、269 ページの「NTFS の記憶ディレクトリ ACL の修復」を参照してください。
- この項の VOB 移動手順はすべて、元の VOB 記憶ディレクトリを維持します。新しい場所の VOB にアクセスでき、移動後も内容が変更されていないことを確認したら、古い VOB 記憶ディレクトリを削除し、不要な記憶領域を解放します。
- VOB に対してデータベース スナップショットのバックアップが有効になっている場合は、`vob_snapshot_setup` プログラムを使用して無効にしてから移動を開始し、移動が完了したら再度有効にします。
- スキーマ バージョン 53 でフォーマットされた VOB をスキーマ バージョン 54 をサポートするホストに移動する場合、移動が完了してから VOB の形式を変更する必要があります。スキーマ バージョン 54 でフォーマットされた VOB をスキーマ バージョン 53 をサポートするホストに移動することはできません。詳細については、95 ページの「VOB スキーマ バージョン」を参照してください。
- VOB のプールをスクラブして、移動する前にサイズを縮小します。
- 動的ビューで作業しているユーザーに対して、VOB をアンマウントしてから VOB 移動手順を開始するよう指示します。処理が完了したら、VOB を再マウントできます。

**注意:** ClearCase レジストリに VOB のコピーを複数登録しないでください。この章で説明する処理では VOB 記憶ディレクトリをコピーしますが、元のディレクトリを登録解除するまでは、コピーしたディレクトリを登録しません。VOB のコピーが同じレジストリに複数登録されている場合、各コピーの別のリージョンにタグが付いていても、重大な VOB エラーが発生し、データが失われる可能性があります。ClearCase レジストリの詳細については、第 3 章を参照してください。

## レプリカ VOB に関する特記事項

---

VOB レプリカを新しいホストに移動する場合、最初に次の処理を実行します。

- 1 レプリカがそれ自体のレプリカ オブジェクトのマスターであることを確認します。

`portland` という名前のレプリカの場合、`portland` レプリカ オブジェクトのマスターシップを保持するレプリカをチェックします。

```
cleartool describe replica:portland@¥libpub
```

...

マスター レプリカ : west

レプリカ オブジェクトがそれ自体のマスターでない場合、マスターシップを変更します。レプリカ オブジェクトのマスターであるレプリカで、次のコマンドを実行します。

```
multitool chmaster replica:portland@¥libpub replica:portland@¥libpub
```

この処理は必須ではありませんが、すべてのレプリカがそれ自体のレプリカ オブジェクトのマスターになるように設定することをお勧めします。この変更は、レプリカ間の同期が維持されている限り、いつでも実行できます。レプリカを移動する場合、このセルフマスターシップによって、移動が原因で誤送信されたパケットの問題をチームで診断、修復する必要がなくなります。レプリカ オブジェクトで誤ったホストが指定されている場合、パケットはその誤ったホストに送信されます。この場合、誤送信されたパケットを正しいホストに移動して、そのホストにインポートします。

- 2 レプリカのホスト名プロパティを変更します。レプリカの現在の場所から最後のエクスポートを実行した後に、`multitool chreplica` を使用して、レプリカのホスト名を更新します。このコマンドは、レプリカのマスターであるサイトから実行する必要があります。ほとんどの場合、現在のサイトがこれに該当します。

```
multitool chreplica -host target-host replica:portland@¥libpub
```

"portland" のレプリカ情報を更新しました。

**メモ:** 移動対象のレプリカにエクスポートするすべてのレプリカは、移動後のレプリカのサイト (現在のサイトではない場合は、手順 2 で `chreplica` コマンドを実行したサイト) から同期処理を実行して、すべて更新する必要があります。

## Windows での VOB の移動

---

この項では、次の 2 つの一般的な処理について説明します。

- VOB を同じ Windows ドメイン内のホストに移動する。この処理では、VOB を Windows ホスト上の別のパーティションに移動することもできます。
- VOB を別の Windows ドメイン内のホストに移動する。

## ドメイン内での VOB の移動

次の処理は、VOB ¥libpub を VOB サーバー ホスト ¥¥sol 上の記憶ディレクトリ C:¥ClearCaseStorage¥VOBs¥libpub.vbs から、VOB サーバー ホスト ¥¥vobsvr01 上で vobstg として共有されている記憶ディレクトリに移動する手順について説明します。

- 1 VOB 所有者または権限を持つユーザーとして、VOB の現在のサーバーにログオンします。  
この例では、VOB の現在のサーバーは ¥¥sol です。

- 2 VOB をロックします。

- 3 VOB サーバー ホストで ClearCase を停止します。

- 4 すべての所有権情報を維持したまま、VOB 記憶ディレクトリをコピーします。VOB 記憶ディレクトリの ACL に格納されている所有権情報を保持するコピー ユーティリティを使用する必要があります。271 ページの「VOB または ビューの記憶ディレクトリをコピーするとき NTFS ACL を保持する」を参照してください。この場合、ClearCase ユーティリティの ccase-home-dir¥etc¥utils¥ccopy.exe を使用することをお勧めします。

```
C:¥ClearCaseStorage¥VOBs> net use E: ¥¥vobsvr01¥vobstg
```

```
C:¥ClearCaseStorage¥VOBs> ccopy libpub.vbs E:¥libpub.vbs
```

- 5 同じ VOB サーバー ホスト上の新しい場所に VOB 記憶ディレクトリをコピーした場合は、ClearCase を再起動します。

- 6 VOB オブジェクトと VOB タグを、新しい VOB 記憶ディレクトリを参照する新しい VOB オブジェクトと VOB タグに置き換えます。ClearCase 管理コンソールまたは以下のコマンドを使用します。これは、新しい VOB 記憶ディレクトリがサーバー vobsvr01 上にある場合の例です。

```
cleartool register -vob -replace ¥¥vobsvr01¥vobstg¥libpub.vbs
```

```
cleartool mktag -vob -replace -tag ¥libpub ¥¥vobsvr01¥vobstg¥libpub.vbs
```

- 7 VOB をアンロックします。

- 8 すべてのクライアントが新しい場所にある VOB にアクセスできることを確認します。

## 別のドメインへの VOB の移動

Windows の場合、スキーマ バージョン 54 でフォーマットされた VOB は、ユーザー、グループ、リソース (ホスト) を表す Windows セキュリティ識別子 (SID) を保持します。VOB を別のドメインに移動すると、この SID が無効になるので、新しいドメインで有効な SID に変更 (マッピング) する必要があります。ClearCase には、ユーティリティ プログラム vob\_sidwalk が用意されています。このプログラムを使用すると、VOB を異なるドメインに移動した後で、SID を柔軟にマッピングすることができます。この処理を続ける前に、vob\_sidwalk のリファレンス ページを参照することをお勧めします。

次の処理は、VOB ¥libpub を古いドメインである VOB サーバー ホスト ¥¥sol 上の記憶ディレクトリ C:¥ClearCaseStorage¥VOBs¥libpub.vbs から、新しいドメイン内の VOB サーバー ホスト ¥¥vobsvr-new 上で vobstg として共有されている記憶ディレクトリに移動します。この処理を実行するには、古いドメインと新しいドメインの両方に対して ¥libpub の VOB 所有者、または権限を持つユーザーとしてログインする必要があります。

- 1 VOB がスキーマ バージョン 54 でフォーマットされていることを確認します。以前のスキーマ バージョンでは、VOB を別のドメインに移動する操作はサポートされていません。VOB のスキーマ バージョンを確認するには、ClearCase 管理コンソールまたは cleartool describe コマンドを使用します。VOB スキーマ バージョンとその変更方法の詳細については、95 ページの「VOB スキーマ バージョン」を参照してください。
- 2 VOB 所有者または権限を持つユーザーとして VOB サーバー ホストにログオンします。
- 3 VOB をロックします。これにより、手順 4 を終了するまで新しい VOB オブジェクトは作成されません。
- 4 ¥libpub 内のオブジェクトに関連付けられたユーザー名とグループ名を一覧表示する SID ファイルを生成します。vob\_siddump を実行して、カンマ区切り (csv) 形式で SID ファイルを生成します。

```
ccase-home-dir¥etc¥utils¥vob_siddump ¥libpub ^
C:¥ClearCaseStorage¥VOBs¥libpub.vbs¥libpub.csv
```

SID ファイルは、VOB 記憶ディレクトリに作成するようにしてください。これにより、記憶ディレクトリを移動した後でも、新しい VOB ホストで SID ファイルを使用できます (手順 10 で必要になります)。

- 5 VOB サーバー ホストで ClearCase を停止します。
- 6 VOB 記憶ディレクトリを新しい場所にコピーします。

```
C:¥ClearCaseStorage¥VOBs> net use E: ¥¥vobsvr-new¥vobstg
C:¥ClearCaseStorage¥VOBs> xcopy libpub.vbs E:¥libpub.vbs /s
```

**メモ:** 既存の VOB 記憶ディレクトリの ACL は新しいドメインでは無効なため、xcopy など、この手順の ACL を保持しないコピー ユーティリティを使用してください。

- 7 VOB 記憶ディレクトリの保護を修復します。¥libpub の VOB 所有者または権限を持つユーザーとして、新しいドメイン (この例では ¥¥vobsvr-new) 上の VOB サーバー ホストにログオンします。fix\_prot ユーティリティを実行します。この例では、vobadm は新しい VOB 所有者名、ccusers は VOB の新しい基本グループ名、V:¥vobstg¥libpub.vbs は ¥¥vobsvr-new 上の VOB 記憶ディレクトリのホスト上のローカル パス名にそれぞれ該当します。

```
ccase-home-dir¥etc¥utils¥fix_prot -root -r -chown vobadm -chgrp ccusers ^
V:¥vobstg¥libpub.vbs
```

- 8 VOB オブジェクトと VOB タグを、新しい VOB 記憶ディレクトリを参照する新しい VOB オブジェクトと VOB タグに置き換えます。ClearCase 管理コンソールまたは次のコマンドを使用します。

```
cleartool register -vob -replace ¥¥vobsvr-new¥vobstg¥libpub.vbs
```

```
cleartool mktag -vob -replace -tag ¥libpub ¥¥vobsvr-new¥vobstg¥libpub.vbs
```

¥¥vobsvr-new が ¥¥sol と同じレジストリ リージョンにない場合は、cleartool register と cleartool mktag で -replace オプションを使用する必要はありませんが、¥libpub の古い登録内容とタグは移動後に無効になるので削除する必要があります。

- 9 VOB をロックします。これで、VOB が登録され、VOB にタグが付けられましたが、この手順を完了するまでは VOB を使用できません。VOB が使用可能になる前にユーザーがアクセスする可能性がある場合は、ここで VOB をロックしてください。

- 10 マッピング ファイルを作成します。手順 4 で生成した SID ファイル

(¥¥vobsvr-new¥vobstg¥libpub.vbs¥libpub.csv) を開きます。カンマ区切り形式のファイルに対応した表計算プログラムを使用すると、簡単にこのファイルを編集できます。この例では、csv ファイルからある 1 行を抜き出したものを示します。わかりやすくするため、ヘッダー行を併記します。SID 文字列はスペースの関係で省略しています。

Old-name	Type	Old-SID	New-name	Type	New-SID	Count
OLD¥akp	USER	NT:S-1-2-21-532...	IGNORE	USER		137

ファイルの各行で、New-name フィールドの文字列 IGNORE を、新しいドメイン名と Old-name フィールドのユーザー名で構成された文字列で置き換えます。次に、最後の 3 つのフィールド (Type、New-SID、Count) を削除します。この例では、古いドメイン名は OLD で、新しいドメイン名は NEW です。変更後の行は、次のようになります。

Old-name	Type	Old-SID	New-name	Type	New-SID	Count
OLD¥akp	USER	NT:S-1-2-21-532...	NEW¥akp			

この例では古いドメインと新しいドメインのユーザー名は同じですが、古いドメインのユーザー名やグループ名を新しいドメインの異なるユーザー名やグループ名にマッピングする場合にも同じ手順を使用できます。

SID ファイルのすべての行を編集してカンマ区切りファイルとして保存し、このファイルを vob\_sidwalk -map を実行するときにマッピング ファイルとして使用します。マッピング ファイルの各行には、カンマで適切に区切られた 4 つのフィールドが必要です。この処理で作成した例の行は、.csv 形式で次のようになります。

```
OLD¥akp,USER,NT:S-1-2-21-532...,NEW¥akp
```

**メモ:** 文字列 DELETE を New-name フィールドに指定すると、VOB の任意のオブジェクトの所有権を VOB 所有者に再割り当てできます。マッピング ファイルを作成せずに、VOB のすべてのオブジェクトの所有権を VOB 所有者に再割り当てすることもできます。353 ページの「VOB 所有者への所有権の再割り当て」を参照してください。

- 11 マッピング ファイルをテストします。-execute オプションを指定せずに vob\_sidwalk を実行します。マッピング ファイル libpub-map.csv 内のマッピングのリストは SID ファイル (この例では libpub-test.csv) に書き込まれますが、VOB は変更されません。

```
ccase-home-dir¥etc¥utils¥vob_sidwalk -map ^
¥¥vobsvr-new¥vobstg¥libpub.vbs¥libpub-map.csv ¥libpub libpub-test.csv
```

- 12 VOB をアンロックします。この処理を完了する前にユーザーがアクセスする可能性がある場合、自分以外のユーザーを対象に VOB を再度ロックしてください (cleartool lock -nusers you)。この操作を実行するには、VOB への書き込みアクセス権が必要です。

- 13 VOB に格納されているユーザーとグループの識別情報を更新します。マッピング ファイルが正しいことを確認してから、vob\_sidwalk を実行します。この例では、libpub-map.csv は手順 10 で作成したマッピング ファイルです。

```
ccase-home-dir¥etc¥utils¥vob_sidwalk -execute -map ^
¥¥vobsvr-new¥vobstg¥libpub.vbs¥libpub-map.csv ¥libpub libpub-exec.csv
```

vob\_sidwalk は、マッピング ファイルで指定したとおりに所有権を再マッピングして、libpub-exec.csv の変更内容を記録します。

- 14 ファイル システムの ACL を復元します。VOB 所有者または権限を持つユーザーとして ¥¥vobsvr-new にログオンしている間に、-recover\_filesystem オプションを指定して vob\_sidwalk を実行し、正しい ACL を VOB 記憶ディレクトリに適用します。

```
ccase-home-dir¥etc¥utils¥vob_sidwalk -recover_filesystem ¥libpub recov.csv
```

vob\_sidwalk は、この手順でファイル recov.csv に加えられた変更内容を記録します。

- 15 新しいドメインのすべてのクライアントが VOB にアクセスできることを確認します。VOB がまだロックされている場合は、アンロックします。

- 16 新しいドメインのすべての ClearCase ユーザーが VOB のオブジェクトにアクセスできることを確認します。ユーザーは、新しいオブジェクトの作成と、自身が所有するオブジェクトの変更と削除を実行できます。

**メモ:** 新しいドメインと古いドメインでユーザー名が異なる場合、そのユーザーはバージョンやブランチの作成者に関連する権限 (たとえば、自身が作成したバージョンを削除する権限) を失います。この操作は、上位の権限を持つユーザー (VOB 所有者、ClearCase 管理者グループのメンバー) も実行できます。

## UNIX での VOB の移動

---

この項では、UNIX での VOB の一般的な移動処理について説明します。

- VOB を同一アーキテクチャ (バイナリ データ形式) の UNIX VOB サーバー ホストに移動する。この処理を使用すると、VOB を UNIX ホスト上の別のパーティションに移動することもできます。
- VOB を異なるアーキテクチャの UNIX ホストに移動する。

処理をわかりやすくするため、この項の手順では次の例を使用して説明します。

- 移動する VOB 記憶ディレクトリの現在の (移動前の) 場所は、/vobstore/libpub.vbs、ホスト名は sol です。
- VOB タグは /vobs/libpub です。
- VOB 記憶ディレクトリの新しい場所 (移動先) は、/vobstore2/libpub.vbs です。この例では、次のケースについて説明します。
  - 新しい場所も sol 上にある。
  - 新しい場所は別のホスト vobsvr04 上にある。

### VOB にリモート プールがある場合

この項の処理は、リモート プールがある VOB の移動に使用できます。移動を開始する前に VOB にリモート プールがあるかどうかを確認し、移動後にプールにアクセスできるかことを確認してください。

- 1 VOB にリモート記憶プールがあるかどうかを確認します。

```
cleartool lspool -long -invob /proj/libpub | egrep '(^pool|link)'
```

```
pool "cdft"
```

```
pool "ddft"
```

```
pool "sdft"
```

```
pool "s_2"
```

プール記憶リンク ターゲット パス名 "/net/vobsvr04/ccase\_pools/s\_2"

lspool の出力は、VOB にリモート プール、s\_2 があることを示しています。

- 2 ターゲット ホストからリモート プールにアクセスできることを確認します。VOB 記憶ディレクトリを移動しても、そのリモート記憶プールは移動しません。VOB の新しいホストが、VOB の現在のホストで使用されているグローバルパス名と同じものを使用できることを確認する必要があります。このグローバルパス名は、VOB の現在のホストが各リモート記憶プールにアクセスする際に使用するものです。

- VOB を同じホスト上の別の場所に移動する場合、このグローバルパス名はそのまま有効です。
- VOB を別のホストに移動する場合は、移動先のホストにログオンして、そのホストからすべてのリモート記憶プールにアクセスできることを確認します。

## リモート プールの統合

複数のリモート プールがある VOB を UNIX から Windows に移動する場合 (166 ページの「UNIX から Windows への VOB の移動」を参照)、VOB を移動する前にリモート プールを統合する必要があります。この処理を行うと、VOB を NAS デバイスに移動した後も VOB のリモート プールを統合できます (170 ページの「NAS デバイスへの VOB の移動」を参照)。

- 1 VOB サーバー ホストにログオンします。VOB 所有者または権限を持つユーザーとしてログオンします。
- 2 リモート プールを検索します。VOB 記憶ディレクトリに移動して、すべてのリモート プールと各リモート プールを参照するリンクを確認します。この例では、UNIX `find` コマンドによって、リモート プールへの単一のシンボリック リンクが表示されます。

```
cd /vobstg/libpub.vbs
find . -type l -exec ls -l {} \;
lrwxrwxrwx    1 root          12 Dec 30  1999
d/ddft_2 ->/net/vobsvr5/pools/libpub/d/ddft_2
```

- 3 各リモート プールをローカル ディレクトリで置き換えます。各リモート プールについて、リンクをプールのローカル コピーで置き換えます。この操作の間、ファイルとディレクトリの保護と所有権情報を維持する必要があります。この手順で使用する UNIX コマンドは、シンボリック リンク `d/ddft_2` を削除して、リンク先 `/net/vobsvr5/pools/libpub/d/ddft_2` の内容で置きかえます。リンク先に異なるターミナル リーフが設定されている場合、その内容を `ddft_2` というローカル ディレクトリにコピーする必要があります。

```
rm d/ddft_2

cd /net/vobsvr5/pools/libpub/d; tar -cf - ddft_2 | ¥
(cd/vobstg/libpub.vbs; tar -xBpf-)
```

- 4 VOB にリモート プールがないことを確認します。VOB サーバー ホスト上で ClearCase を停止してから再起動し、`cleartool lspool` コマンドを使用して VOB にリモート プールがないことを確認します。

```
cleartool lspool -long -invob /vobs/libpub
```

`lspool` の出力には、リンク先が表示されません。

- 5 **VOB タグを修正します。**統合した VOB を新しいホストに移動しない場合 (VOB 記憶ディレクトリは NAS デバイスに移動するが、VOB サーバー ホストは変更しない場合など)、Windows リージョンにタグが付いているときは、タグを修正して分割プール マップを削除する必要があります。ClearCase 管理コンソールの [レジストリ リージョン] ノードを使用します。VOB タグの [プロパティ] ページには [マウント オプション] タブがあり、ClearCase 管理者グループのメンバーとしてログインしている場合、分割プール マップを編集できます。ClearCase 管理コンソールを使用できない場合、`cleartool rmtag` と `mktag` を使用して VOB タグを削除し、分割プール マップを使用せずに再作成します。
- 6 **統合されたプールにユーザーがアクセスできることを確認します。**VOB のテスト後、古いリモート プール記憶領域を削除できます。

**メモ:** リモート プールを統合してから、VOB のバックアップ手順をチェックして修正します。新しく統合したプールがほかの VOB と共にバックアップされていることと、古いリモートプールがバックアップの対象になっていないことを確認します。プールを統合する前に作成したバックアップを復元する場合、リモート プールが再作成され、復元した VOB は使用できません。

## ClearCase 以外のアクセス用に VOB がエクスポートされている場合

この項の処理を使用すると、ClearCase 以外のアクセス用にエクスポートされた VOB を移動できます。処理を開始する前に、VOB をエクスポートするエクスポート ビューを終了します。移動が完了してから、エクスポート ビューを再実行します。

詳細については、372 ページの「非 ClearCase アクセスのためのエクスポート ビューの設定」を参照してください。

## 同一アーキテクチャの UNIX ホスト間での VOB の移動

次の処理を実行して、VOB を UNIX ホスト上の別のディスク パーティションや、同一アーキテクチャの UNIX ホストに移動できます。

- 1 **VOB サーバー ホストにログオンします。**VOB 所有者または権限を持つユーザーとしてログオンします。
- 2 **VOB をロックします。**
- 3 **VOB サーバー ホストで ClearCase を停止します。**
- 4 **VOB 記憶ディレクトリをコピーします。**VOB を同一ディスク パーティション内で移動する場合と、別のディスク パーティションに移動する場合とでは、実行する処理が異なります。

VOB を別のディスク パーティションに移動する場合、`tar` またはこれに相当するコマンドを使用して、リモート記憶プールを除く VOB 記憶ディレクトリ全体を新しい場所にコピーします。たとえば、VOB を同じホスト上の異なるディスク パーティションに移動するには、次のコマンドを実行します。

```
cd /vobstore
```

```
tar -cf - libpub.vbs | ( cd /vobstore2 ; tar -xBpf - )
```

VOB を異なるホストに移動するには、次のコマンドを実行します。

```
cd /vobstore
```

```
tar -cf - libpub.vbs | rsh vobsvr04 'cd /vobstore2 ; tar -xBpf -'
```

**メモ:** tar コマンドの **-B** オプションは、一部の UNIX プラットフォームではサポートされていません。また、プラットフォームによっては、**rsh** コマンドの名前が異なる (たとえば **remsh**) 場合があります。ご使用のオペレーティング システムのリファレンス ページを参照してください。

VOB 記憶ディレクトリを同一ディスク パーティション内で移動する場合は、UNIX **mv** コマンドを使用します。

- 5 同じ VOB サーバー ホスト上の新しい場所に VOB 記憶ディレクトリをコピーした場合は、**ClearCase** を再起動します。
- 6 VOB オブジェクトと VOB タグを、新しい VOB 記憶ディレクトリを参照する新しい VOB オブジェクトと VOB タグに置き換えます。**ClearCase** 管理コンソールまたは次のコマンドを使用します。これは、新しい VOB 記憶ディレクトリがサーバー **sol** 上にある場合の例です。

```
cleartool register -vob -replace /net/sol/vobstore2/libpub.vbs
```

```
cleartool mktag -vob -replace -tag /vobs/libpub /net/sol/vobstore2/libpub.vbs
```

- 7 VOB をアンロックします。
- 8 すべてのクライアントが新しい場所にある VOB にアクセスできることを確認します。

## 異なるアーキテクチャの UNIX ホスト間での VOB の移動

次の処理を実行して、VOB を異なるアーキテクチャの UNIX ホストに移動します。「同一アーキテクチャの UNIX ホスト間での VOB の移動」の手順とほとんど同じですが、VOB データベースを移動する前にダンプして、ターゲット ホスト上で形式を変更する手順が追加されています。

- 1 VOB サーバー ホストにログオンします。VOB 所有者または権限を持つユーザーとしてログオンします。
- 2 **cleartool reformatvob -dump** コマンドで VOB データベースをダンプします。これにより、VOB もロックされます。

```
cleartool reformatvob -dump /vobstore/libpub.vbs
```

**reformatvob -dump** によって、VOB データベースは無効としてマークされます。

**reformatvob -load** コマンドを実行するまで、VOB データベースは使用できません。

- 3 VOB 記憶ディレクトリをコピーします。まず、移動先の場所が存在し、書き込み可能であることを確認します。次に VOB 記憶ディレクトリを新しいホストにコピーします。

```
cd /vobstore
```

```
tar -cf - libpub.vbs | rsh vobsvr04 'cd /src/vobstore ; tar -xBpf -'
```

**メモ:** tar コマンドの -B オプションは、一部の UNIX プラットフォームではサポートされていません。また、プラットフォームによっては、rsh コマンドの名前が異なる (たとえば remsh) 場合があります。ご使用のオペレーティング システムのリファレンス ページを参照してください。

- 4 以前の VOB のサーバー プロセスを終了します。VOB サーバー ホスト上で ClearCase を停止してから再起動するか、古い VOB にサービスを提供している vob\_server プロセスと vobrpc\_server プロセスのプロセス テーブルを検索します。ps -ax または ps -ef を使用して、VOB 記憶ディレクトリ名 (この例では libpub.vbs) を検索します。次に UNIX kill コマンドを使用して、これらのプロセスを終了します。
- 5 新しい VOB サーバー ホストにログオンします。VOB 所有者または権限を持つユーザーとしてログオンします。
- 6 ダンプ ファイルから VOB データベースを再作成します。

```
cleartool reformatvob -load /src/vobstore/libpub.vbs
```

- 7 VOB オブジェクトと VOB タグを、新しい VOB 記憶ディレクトリを参照する新しい VOB オブジェクトと VOB タグに置き換えます。ClearCase 管理コンソールまたは次のコマンドを使用します。これは、新しい VOB 記憶ディレクトリがサーバー sol 上にある場合の例です。

```
cleartool register -vob -replace /net/sol/vobstore2/libpub.vbs
```

```
cleartool mktag -vob -replace -tag /vobs/libpub /net/sol/vobstore2/libpub.vbs
```

- 8 VOB をアンロックします。
- 9 すべてのクライアントが新しい場所にある VOB にアクセスできることを確認します。

## 異なるタイプのオペレーティング システムへの VOB の移動

---

VOB を Windows ホストと UNIX ホスト間で移動する場合、VOB 記憶ディレクトリと VOB データベースに記録されているユーザー識別情報をすべて変更する必要があります。VOB データベースのバイナリ データ形式も同様に変更する必要があります。この要件を満たすため、ほかの VOB 移動シナリオでは不要だった、いくつかの手順を実行する必要があります。

- 移動前に vob\_sidwalk を実行して、VOB オブジェクトの所有権に関する情報を取得します。
- reformatvob を使用して、VOB データベースを移植可能な形式にダンプします。
- ダンプしたデータベースを含む VOB 記憶ディレクトリを新しいホストにコピーします。
- reformatvob を使用して、VOB データベースを適切なバイナリ形式にロードします。

- 移動後、VOB 記憶ディレクトリ上のファイル システム保護をリセットします。
- VOB のオブジェクト所有者の SID (UNIX の場合は UID と GID) を再マッピングします。

各手順は、VOB の移動に必要なほかの手順とともに、この項に記載されています。

## スキーマ バージョンの互換性

VOB を Windows ホストから UNIX ホストに移動する前に、VOB をスキーマ バージョン 54 でフォーマットし、VOB スキーマ バージョン 54 をサポートするように UNIX ホストを設定する必要があります。一部の UNIX ホストは、この機能に対応していません。詳細については、95 ページの「VOB スキーマ バージョン」を参照してください。

vob\_sidwalk と vob\_siddump は、VOB スキーマ バージョン 53 と互換性がありません。vob\_sidwalk は、ローカル VOB とビューをサポートし、VOB スキーマ バージョン 54 をサポートするように設定されているホストにのみインストールされます。vob\_siddump は、操作対象がローカル VOB に制限されていないので、すべてのホストにインストールされます。

## Windows から UNIX への VOB の移動

処理をわかりやすくするため、この項の手順では次の例を使用して説明します。

- 移動する VOB 記憶ディレクトリの現在の場所 (移動前) は、Windows ホスト vobsvr-nt 上の C:\ClearCaseStorage\libpub.vbs です。この VOB の VOB タグは libpub です。
- VOB 記憶ディレクトリの新しい場所 (移動先) は、UNIX ホスト vobsvr2 上の /vobstg/libpub.vbs です。VOB タグ /vobs/libpub が、この VOB 用に作成されます。

VOB を Windows から UNIX に移動するには

- 1 VOB 所有者または権限を持つユーザーとして、Windows VOB サーバー ホストにログオンします。
- 2 VOB をロックします。これにより、手順 3 を終了するまで新しい VOB オブジェクトは作成されません。
- 3 libpub 内のオブジェクトに関連付けられたユーザー名とグループ名を一覧表示する SID ファイルを生成します。次に示すとおり、vob\_siddump ユーティリティを実行します。

```
ccase-home-dir\etc\utils\libpub -raw_sid libpub ^
C:\ClearCaseStorage\libpub.vbs\libpub.csv
```

SID ファイルは、VOB 記憶ディレクトリに作成するようにしてください。これにより、記憶ディレクトリを移動した後でも、新しい VOB ホストで使用できます。手順 15 で、SID ファイルが必要になります。

- 4 VOB データベースをダンプします。cleartool reformatvob コマンドを使用します。  
cleartool reformatvob -dump C:¥ClearCaseStorage¥libpub.vbs  
  
reformatvob -dump によって、VOB データベースは無効としてマークされます。  
reformatvob -load コマンドを実行するまで、VOB データベースは使用できません。
- 5 VOB 記憶ディレクトリをコピーします。ファイル システム コピー ユーティリティを使用して、VOB 記憶ディレクトリ全体を UNIX ホストにコピーします。この例では、コピー先の UNIX ホスト vobsvr2 上で SMB サーバーが実行され、¥vobstg パーティションが共有されていることを前提としています。  
  
C:¥ClearCaseStorage¥VOBs> net use E: ¥¥vobsvr2¥vobstg  
  
C:¥ClearCaseStorage¥VOBs> xcopy libpub.vbs E:¥libpub.vbs /s  
  
**メモ:** UNIX ホストでは ACL はサポートされていないため、xcopy やこの手順の ACL を保持しないほかのコピー ユーティリティも使用できます。
- 6 Windows 上の VOB のサーバー プロセスを終了します。Windows VOB サーバー ホスト (この例では、¥¥vobsvr-nt) 上で ClearCase を停止し、再起動します。
- 7 UNIX VOB サーバー ホストにログオンします。
- 8 VOB 所有者の識別情報を更新します。次に示すように fix\_prot ユーティリティを使用して、VOB の新しい .identity ディレクトリを作成します。この方法で fix\_prot を実行するには、root ユーザーである必要があります。この例では、VOB の所有者を vobadm に、VOB のプライマリ グループを ccusers に設定しています。  
  
ccase-home-dir/etc/utlis/fix\_prot -root -recurse -chown vobadm -chgrp ccusers ^  
/vobstg/libpub.vbs
- 9 ダンプ ファイルから VOB データベースを再作成します。  
  
# cleartool reformatvob -load /vobstg/libpub.vbs
- 10 VOB オブジェクトと VOB タグを、新しい VOB 記憶ディレクトリを参照する新しい VOB オブジェクトと VOB タグに置き換えます。ClearCase 管理コンソールまたは次のコマンドを使用します。  
  
cleartool register -vob -replace /net/vobsvr2/vobstg/libpub.vbs  
  
cleartool mktag -vob -replace -tag /vobs/libpub /net/vobsvr2/vobstg/libpub.vbs

## 11 マッピング ファイルを作成します。手順 3 で生成した SID ファイル

(/vobstg/libpub.vbs/libpub.csv) を開きます。カンマ区切り形式のファイルに対応した表計算プログラムを使用すると、簡単にこのファイルを編集できます。この例では、csv ファイルからある 1 行を抜き出したものを示します。わかりやすくするため、ヘッダー行を併記します。SID 文字列はスペースの関係で省略しています。

Old-name	Type	Old-SID	New-name	Type	New-SID	Count
OLD¥akp	USER	SID:3.0105037...	IGNORE	USER		137

ファイルの各行で、**New-name** フィールドの文字列 **IGNORE** を、UNIX VOB サーバーホストで有効なユーザー名またはグループ名で置き換えます。次に、最後の 3 つのフィールド (**Type**、**New-SID**、**Count**) を削除します。

Old-name	Type	Old-SID	New-name	Type	New-SID	Count
OLD¥akp	USER	SID:3.0105037...	akp			

この例では UNIX と Windows でユーザー名は同じですが、Windows のユーザー名やグループ名を UNIX の異なるユーザー名やグループ名にマッピングする場合にも同じ手順を使用できます。

SID ファイルのすべての行を編集してから、カンマ区切りファイルとして保存します。**vob\_sidwalk -map** を実行するときには、このファイルをマッピング ファイルとして使用します。マッピング ファイルの各行には、カンマで適切に区切られた 4 つのフィールドが必要です。この手順で作成した例の行は、.csv 形式で次のようになります。

OLD¥akp,USER,SID:3.0105037...,akp

**メモ:** 文字列 **DELETE** を **New-name** フィールドに指定すると、VOB の任意のオブジェクトの所有権を VOB 所有者に再割り当てできます。マッピング ファイルを作成せずに、VOB のすべてのオブジェクトの所有権を VOB 所有者に再割り当てすることもできます。353 ページの「VOB 所有者への所有権の再割り当て」を参照してください。

## 12 マッピング ファイルをテストします。-execute オプションを指定せずに vob\_sidwalk を実行します。ファイル libpub-map.csv 内のマッピングのリストは SID ファイル (この例では libpub-test.csv) に書き込まれますが、VOB は変更されません。

ccase-home-dir/etc/utls/vob\_sidwalk -map /vobstg/libpub.vbs/libpub-map.csv ¥  
/vobs/libpub /libpub-test.csv

## 13 VOB をアンロックします。この処理を完了する前にユーザーがアクセスする可能性がある場合、自分以外のユーザーを対象に VOB を再度ロックしてください (cleartool lock -nusers you)。

- 14 VOB に格納されているユーザーとグループの識別情報を更新します。マッピング ファイルが正しいことを確認してから、`vob_sidwalk` を実行します。この例では、`libpub-map.csv` は 手順 11 で作成したマッピング ファイルです。

```
ccase-home-dir/etc/utls/vob_sidwalk -execute -map ¥  
/vobstg/libpub.vbs/libpub-map.csv /vobs/libpub /libpub-exec.csv
```

`vob_sidwalk` は、マッピング ファイルで指定された変更を適用し、新しい SID ファイル `libpub-exec.csv` の変更内容を記録します。

- 15 VOB のグループ リストとコンテナ保護を更新します。`-recover_filesystem` オプションを指定して `vob_sidwalk` を実行します。

```
ccase-home-dir/etc/utls/vob_sidwalk -recover_filesystem /vobs/libpub recov.csv
```

`vob_sidwalk` は、この手順でファイル `recov.csv` に加えられた変更内容を記録します。

- 16 すべてのクライアントが新しい場所にある VOB にアクセスできることを確認します。VOB がまだロックされている場合は、アンロックします。

- 17 新しいドメインのすべての ClearCase ユーザーが VOB のオブジェクトにアクセスできることを確認します。ユーザーは、新しいオブジェクトの作成と、自身が所有するオブジェクトの変更と削除を実行できます。

## UNIX から Windows への VOB の移動

処理をわかりやすくするため、この項の手順では次の例を使用して説明します。

- 移動する VOB 記憶ディレクトリの現在の場所 (移動前) は、UNIX ホスト `vobsvr2` 上の `/vobstg/libpub.vbs` です。この VOB 用の VOB タグは `/vobs/libpub` です。
- VOB 記憶ディレクトリの新しい場所 (移動先) は、Windows ホスト `vobsvr-nt` 上の `C:¥ClearCaseStorage¥VOBs¥libpub.vbs` です。VOB タグ `¥libpub` が、この VOB 用に作成されます。

VOB を UNIX から Windows に移動するには

**メモ:** VOB にリモート記憶プールがある場合、最初に各プールを VOB ルートディレクトリ下に統合する必要があります。UNIX のシンボリック リンクは Windows ではサポートされていないため、VOB 記憶ディレクトリは Windows ホストの単一のパーティション上に存在する必要があります。具体的な手順については、159 ページの「リモート プールの統合」を参照してください。

- VOB サーバー ホストにログオンします。VOB 所有者または権限を持つユーザーとしてログオンします。
- VOB をロックします。これにより、手順 3 を終了するまで新しい VOB オブジェクトは作成されません。

- 3 /vobs/libpub 内のオブジェクトに関連付けられたユーザーとグループの名前、UID、GIDを一覧表示する SID ファイルを生成します。次の例に示すとおり、vob\_siddump ユーティリティを実行します。

```
ccase-home-dir/etc/utils/vob_siddump /vobs/libpub ^  
/vobstg/libpub.vbs/libpub.csv
```

SID ファイルは、VOB 記憶ディレクトリに作成するようにしてください。これにより、記憶ディレクトリを移動した後も、新しい VOB ホストで使用できます。手順 15 で、このファイルが必要になります。

- 4 VOB データベースをダンプします。cleartool reformatvob コマンドを使用します。

```
cleartool reformatvob -dump /vobstg/libpub.vbs
```

reformatvob -dump によって、VOB データベースは無効としてマークされます。reformatvob -load コマンドを実行するまで、VOB データベースは使用できません。

- 5 VOB 記憶ディレクトリをコピーします。ファイルシステム コピー ユーティリティを使用して、VOB 記憶ディレクトリ全体を Windows ホストにコピーします。この例では、UNIX ホスト vobsvr2 上で SMB サーバーが実行され、¥vobstg パーティションが共有されていることを前提としています。Windows ホストで、次のコマンドを実行します。

```
C:¥ClearCaseStorage¥VOBs> net use E: ¥¥vobsvr2¥vobstg
```

```
C:¥ClearCaseStorage¥VOBs> xcopy E:¥libpub.vbs libpub.vbs /s
```

**メモ:** UNIX ホストでは ACL はサポートされていないため、この手順で xcopy を使用できません。

- 6 UNIX 上の VOB のサーバー プロセスを終了します。VOB サーバー ホスト上で ClearCase を停止してから再起動するか、古い VOB にサービスを提供している vob\_server プロセスと vobrpc\_server プロセスのプロセス テーブルを検索します。ps -ax または ps -ef を使用して、VOB 記憶ディレクトリ名 (この例では libpub.vbs) を検索します。次に UNIX kill コマンドを使用して、これらのプロセスを終了します。

- 7 VOB 記憶ディレクトリの保護を修復します。¥libpub の VOB 所有者または権限を持つユーザーとして Windows VOB サーバー ホストにログインし、fix\_prot ユーティリティを実行します。この例では、vobadm は新しい VOB 所有者名、ccusers は VOB の新しい基本グループ名、C:¥ClearCaseStorage¥VOBs¥libpub.vbs は VOB 記憶ディレクトリのホスト上のローカルパス名にそれぞれ該当します。

```
ccase-home-dir¥etc¥utils¥fix_prot -root -r -chown vobadm -chgrp ccusers ^  
C:¥ClearCaseStorage¥VOBs¥libpub.vbs
```

- 8 ダンプ ファイルから VOB データベースを再作成します。cleartool reformatvob コマンドを使用します。

```
cleartool reformatvob -load C:¥ClearCaseStorage¥VOBs¥libpub.vbs
```

- 9 VOB オブジェクトと VOB タグを、新しい VOB 記憶ディレクトリを参照する新しい VOB オブジェクトと VOB タグに置き換えます。ClearCase 管理コンソールまたは次のコマンドを使用します。この例では、C:\ClearCaseStorage が ¥¥vobsvr-nt¥ClearCaseStorage として共有されていることを前提としています。

```
cleartool register -vob -replace ^
¥¥vobsvr-nt¥ClearCaseStorage¥VOBs¥libpub.vbs
```

```
cleartool mktag -vob -tag ¥libpub ^
¥¥vobsvr-nt¥ClearCaseStorage¥VOBs¥libpub.vbs
```

- 10 VOB をロックします。これで、VOB が登録され、VOB にタグが付けられましたが、この処理を完了するまでは VOB を使用できません。VOB が使用可能になる前にユーザーがアクセスする可能性がある場合は、ここで VOB をロックしてください。

- 11 マッピング ファイルを作成します。この処理の手順 4 で生成した SID ファイル (¥¥vobsvr-nt¥ClearCaseStorage¥VOBs¥libpub.vbs¥libpub.csv) を開きます。カンマ区切り形式のファイルに対応した表計算プログラムを使用すると、簡単にこのファイルを編集できます。この例では、csv ファイルからある 1 行を抜き出したものを示します。わかりやすくするため、ヘッダー行を併記します。

Old-name	Type	Old-SID	New-name	Type	New-SID	Count
akp	USER	UNIX:UID-1247	IGNORE	USER		137

ファイルの各行で、New-name フィールドの文字列 IGNORE を、古い名前をマッピングするユーザーまたはグループのドメインの修飾がある名前で置き換えます。次に、最後の 3 つのフィールド (Type、New-SID、Count) を削除します。

Old-name	Type	Old-SID	New-name	Type	New-SID	Count
akp	USER	UNIX:UID-1247	NEW¥akp			

この例では Windows と UNIX でユーザー名は同じですが、UNIX のユーザー名やグループ名を Windows の異なるユーザー名やグループ名にマッピングする場合にも同じ手順を使用できます。

SID ファイルのすべての行を編集してカンマ区切りファイルとして保存し、このファイルを vob\_sidwalk -map を実行するときにマッピング ファイルとして使用します。マッピング ファイルの各行には、カンマで適切に区切られた 4 つのフィールドが必要です。この手順で作成した例の行は、.csv 形式で次のようになります。

```
akp,USER,UNIX:UID-1247,NEW¥akp
```

**メモ:** 文字列 DELETE を New-name フィールドに指定すると、VOB の任意のオブジェクトの所有権を VOB 所有者に再割り当てできます。マッピング ファイルを作成せずに、VOB のすべてのオブジェクトの所有権を VOB 所有者に再割り当てすることもできます。353 ページの「VOB 所有者への所有権の再割り当て」を参照してください。

- 12 マッピング ファイルをテストします。-execute オプションを指定せずに vob\_sidwalk を実行します。ファイル libpub-map.csv 内のマッピングのリストは SID ファイル (この例では libpub-test.csv) に書き込まれますが、VOB は変更されません。

```
ccase-home-dir¥etc¥utils¥vob_sidwalk -map ^
¥¥vobsvr-nt¥ClearCaseStorage¥VOBs¥libpub.vbs¥libpub-map.csv ^
¥libpub libpub-test.csv
```

- 13 VOB をアンロックします。この処理を完了する前にユーザーがアクセスする可能性がある場合、自分以外のユーザーを対象に VOB を再度ロックしてください。この手順を実行するには、VOB への書き込みアクセス権が必要になります。

- 14 VOB に格納されているユーザーとグループの識別情報を更新します。マッピング ファイルが正しいことを確認してから、-execute オプションを指定して vob\_sidwalk を実行します。この例では、libpub-map.csv は手順 10 で作成したマッピング ファイルです。

```
ccase-home-dir¥etc¥utils¥vob_sidwalk -execute -map ^
¥¥vobsvr-nt¥ClearCaseStorage¥VOBs¥libpub.vbs¥libpub-map.csv ^
¥libpub libpub-exec.csv
```

vob\_sidwalk は、マッピング ファイルで指定されたとおりに所有権を再マッピングし、新しい SID ファイル libpub-exec.csv の変更内容を記録します。

- 15 ファイル システムの ACL を復元します。最後に、VOB 所有者または権限を持つユーザーとして ¥¥vobsvr-nt にログインした状態で、-recover\_filesystem オプションを指定して vob\_sidwalk を実行し、正しい ACL を VOB 記憶ディレクトリに適用します。

```
ccase-home-dir¥etc¥utils¥vob_sidwalk -recover_filesystem ¥libpub recov.csv
```

vob\_sidwalk は、この手順でファイル recov.csv に加えられた変更内容を記録します。

- 16 リージョン内のすべてのクライアントが VOB にアクセスできることを確認します。VOB がまだロックされている場合は、アンロックします。

- 17 Windows のすべての ClearCase ユーザーが VOB のオブジェクトにアクセスできることを確認します。ユーザーは、新しいオブジェクトの作成と、自身が所有するオブジェクトの変更と削除を実行できます。

**メモ:** Windows と UNIX でユーザー名が異なる場合、そのユーザーはバージョンやブランチの作成者に関連する権限 (たとえば、自身が作成したバージョンを削除する権限) を失います。この操作は、上位の権限を持つユーザー (VOB 所有者、ClearCase 管理者グループのメンバー) も実行できます。

## NAS デバイスへの VOB の移動

---

VOB の移動に関する処理はすべて、VOB ホストのアーキテクチャ、すなわちバイナリ データ形式を参照します。VOB データベース形式はアーキテクチャに依存します。VOB を別のアーキテクチャのホストに移動する場合には、データベースの形式を変更する必要があります。

VOB データベースが NAS デバイ스에格納されている場合、NAS デバイスではなく、VOB サーバー ホストのアーキテクチャによってバイナリ データ形式が決定されます。VOB データベースの形式を変更する必要があるのは、サーバー ホストのアーキテクチャを変更する場合だけです。VOB を別のメーカーの NAS デバイスに移動する場合でも、VOB サーバー ホストのアーキテクチャに変更がない限り、VOB データベースの形式を変更する必要はありません。

### リモート プールがない VOB の移動

リモート プールがない VOB を UNIX 上の ClearCase ホストから NAS デバイスに移動するには、158 ページの「UNIX での VOB の移動」の手順に従って操作します。ただし、`-host`、`-hpath`、`-gpath` オプションを指定して、`register` コマンドと `mktag` コマンドを実行してください。この例は、UNIX ホスト `ccvobsvr1` に属する VOB `libpub` を `/net/nasdevice/vobstg` に移動して、登録するように変更されています。

```
cleartool register -vob -replace -host ccvobsvr1¥  
-hpath /net/nasdevice/vobstg/libpub.vbs ¥  
-gpath /net/nasdevice/vobstg/libpub.vbs /net/nasdevice/vobstg/libpub.vbs  
  
cleartool mktag -vob -replace -tag /vobs/libpub -host ccvobsvr1 ¥  
-hpath /net/nasdevice/vobstg/libpub.vbs ¥  
-gpath /net/nasdevice/vobstg/libpub.vbs /net/nasdevice/vobstg/libpub.vbs
```

VOB を Windows VOB サーバー ホストから NAS デバイスに移動する際には、同様の変更が必要になります。153 ページの「Windows での VOB の移動」で説明されている手順に従ってください。ただし、`register` コマンドと `mktag` コマンドを、次に示すとおりに変更してください。

```
cleartool register -vob -replace -host sol -hpath ¥¥sol¥vobstore2¥libpub.vbs ^  
-gpath ¥¥sol¥vobstore2¥libpub.vbs ¥¥sol¥vobstore2¥libpub.vbs  
  
cleartool mktag -vob -replace -tag ¥libpub -host ccvobsvr1 ^  
-hpath ¥¥sol¥vobstore2¥libpub.vbs -gpath ¥¥sol¥vobstore2¥libpub.vbs ^  
¥¥sol¥vobstore2¥libpub.vbs
```

## リモート プールがある VOB の移動

UNIX プラットフォームでホスティングされている VOB には、UNIX のシンボリック リンク 経由でアクセスするリモート UNIX ホスト上に、1 つまたは複数の記憶プールがあります。この 構成の VOB を NAS デバイスに移動する場合、シンボリック リンクをローカルディレクトリで 置き換えると、リモート プールを統合できます。リモート プールが VOB サーバー ホストに アクセスできる限り、統合する必要はありません。ただし、統合によって、VOB のバックアッ プやその他の管理タスクを簡素化できます。159 ページの「リモート プールの統合」を参照し てください。



この章では、ビューにおける操作の詳細と記憶領域の要件について説明します。また、動的ビューをサポートする MVFS (マルチバージョン ファイル システム) についても説明します。ビューの詳細については、**mkview** のリファレンス ページを参照してください。

## ビューとビュー管理の概要

---

ClearCase の開発環境には 1 つまたは複数のビューが必要です。ビューは、ClearCase によって制御されるファイル エlement とディレクトリ Element のバージョンにユーザーがアクセスするためのワークスペースを提供します。ビューには、ClearCase によって制御されないビュープライベート ファイル システム オブジェクト (通常のファイルとディレクトリなど) を含めることもできます。

標準的なビューは、各ユーザーによって、または共通のタスクに取り組む小人数のグループによって作成、使用されます。その他のビュー (UCM インテグレーション ビューなど) は、プロジェクト リーダーまたは管理者によって作成され、複数のユーザーによって共有されます。ビューには、次のような各種の管理権限が関連付けられています。

- ビューの作成とアクセス コントロール
- ビューのバックアップと回復
- ビューに使用される記憶領域の移動、削除、管理

ビューには、次の 3 種類があります。

- 動的ビュー (ClearCase LT ではサポートされていません): MVFS を使用して VOB 内の Element のバージョンとビュープライベート オブジェクトへのトランスペアレント アクセスを提供します。動的ビューは、ClearCase のビルド監視ツールとビルド回避ツールをサポートしており、派生オブジェクト (DO) を含めることもできます。開発者は、特定のブランチで Element の最新バージョンに即時にアクセスする必要がある場合、または監視ビルドとビルド回避を利用する必要がある場合に、動的ビューを使用して作業を行います。動的ビューは、ほかのビューより多くの機能がありますが、ネットワークの使用頻度が高く、多くの管理サポートを必要とします。

- **スナップショット ビュー:** ホストのネイティブ ファイル システムを使用して、指定されたエレメントのバージョンのコピーとビュープライベート オブジェクトを保持します。スナップショット ビューは、ビルド監視ツールとビルド回避ツールをサポートしていません。開発者は、ネットワークに接続されていない場合でも、編集、コンパイル、デバッグの環境を簡素化する場合、エレメントの最新バージョンに即時にアクセスする必要がある場合、監視ビルドとビルド回避が必要ない場合に、スナップショット ビューを使用して作業を行います。スナップショット ビューは、動的ビューより機能の数は少ないですが、ネットワークの使用頻度は低く、必要な管理サポートも少なくてすみます。
- **Web ビュー:** Web ビューはスナップショット ビューに似ていますが、ClearCase Web インターフェイスを使用して作成し、アクセスします。

## 動的ビュー

---

**メモ:** 動的ビューは ClearCase LT ではサポートされていません。

動的ビューは、MVFS ディレクトリであり、VOB エレメントへの動的なアクセスを可能にします。動的ビューには、ソフトウェア開発プロセスで作成されるほとんどすべての成果物が含まれます。成果物が ClearCase によって制御されるかどうかは関係ありません。これらの成果物には次のものがあります。

- 選択されたエレメントのバージョン (実際には VOB 記憶プールに保存されている)
- 変更中のファイル (ビューのプライベート記憶領域に保存されており、チェックアウトされたファイルエレメント)
- 変更中のディレクトリ (VOB データベース内で管理されており、チェックアウトされたディレクトリ エレメント)
- ビューで作業しているユーザーによって作成された共有可能派生オブジェクトと非共有派生オブジェクト (ビューのプライベート記憶領域に保存されている) と、これらの派生オブジェクトを記述した構成レコード
- 別のビューで作成され、この動的ビューにウイंकインされた派生オブジェクト (VOB 記憶プールに保存されている)
- ClearCase に制御されておらず、このビューにのみ表示されるその他のファイル、ディレクトリ、リンクなどのビュープライベート オブジェクト (ビューのプライベート記憶領域に保存されている)

## ビュー ルート

ホスト上の各動的ビューは、ホストのビュー ルート ディレクトリのサブディレクトリとして表示されます。UNIX では、デフォルトのビュー ルート ディレクトリは `/view` です。Windows では、デフォルトのビュー ルート ディレクトリは `¥¥view` で、デフォルトのドライブ M にマッピングされています。

各種ビュー記憶ディレクトリのほかに、ビュー ルートには特殊ファイル `.specdev` が含まれています。このファイルが存在しない場合、または破損している場合、ホスト上の動的ビューにアクセスすると、次のようなエラー メッセージが表示されます。

```
cleartool: エラー : ファイル "viewroot" を開くことができません : ClearCase  
オブジェクトが見つかりません。
```

## デフォルトのビュー ルートを変更するには

既存のディレクトリ名またはネットワーク名との競合を回避するため、デフォルトのビュー ルートを変更する必要がある場合は、次のいずれかの処理を実行します。

- UNIX では、`ccase-home-dir/etc/clearcase` ファイルを編集して、以下の UNIX ビュー ルート定義行を変更します。

```
VIEWPATH="/view"
```

- Windows では、[コントロール パネル] から **ClearCase** プログラムを実行します。  
[MVFS] タブの [ネットワーク上のビューの名前] にビュー ルートを指定します。

## ビュー記憶ディレクトリ

動的ビューは標準ディレクトリとして実装され、そのルートディレクトリはビュー記憶ディレクトリと呼ばれます。ビュー記憶ディレクトリには、以下のファイルとサブディレクトリがあります。これらはすべて、**ClearCase** コマンドによって作成され、変更されます。これ以外の方法では変更できません。

<code>.access_info</code>	<code>view_server</code> によって定期的に更新されるビュー アクセス イベント情報のファイル。
<code>.pid</code>	ビューの <code>view_server</code> プロセス ID を含む 1 行から成るテキスト ファイル。
<code>admin</code>	ビューが使用しているディスク容量に関連した管理データが保存されているディレクトリ。データを表示するには、 <b>space -view</b> コマンドを実行します。
<code>config_spec</code>	ビューの現在の構成仕様が保存されているファイル。内容を表示するには、 <b>catcs</b> コマンドを実行します。
<code>.compiled_spec</code>	<code>config_spec</code> の変更されたバージョン。アカウント情報が含まれています。

<code>.identity</code>	UNIX のサブディレクトリ。ビューの所有者とグループ メンバシップを確立する、ファイル サイズがゼロのファイルが保存されています。
<code>identity.sd</code>	Windows のバイナリ データ ファイル。ビュー記憶ディレクトリのセキュリティ記述子が含まれています。
<code>.s</code>	ビューのプライベート記憶領域を実装するサブディレクトリ。
<code>db</code>	ビューの組み込みデータベースを実装するファイルが保存されているサブディレクトリ。
<code>.view</code>	ビューのユニバーサル ユニーク ID (UUID) とその他の属性のリストが保存されているファイル。

## ビュープライベート記憶領域

ビュー記憶ディレクトリのサブディレクトリ `.s` は、ビューのプライベート記憶領域を実装するサブツリーのルートです。UNIX プラットフォームの場合、`.s` は、ローカル ディレクトリか、別の UNIX コンピュータやネットワーク接続記憶デバイスにあるディレクトリへの UNIX シンボリック リンクです。

プライベート記憶領域には、次のような各種のオブジェクトが保存されます。

- **ビュープライベート オブジェクト:** ビュープライベート オブジェクトは、通常のファイルシステム オブジェクトで、ビューに含まれるファイル、ディレクトリ、UNIX リンクなどです。ビュープライベート オブジェクトは、ビューのプライベート記憶領域内のみ存在します。VOB には保存できません。

**メモ:** UNIX 上の動的ビューの場合、UNIX の特殊ファイル タイプ (ソケット、名前付きパイプ、キャラクタ デバイス ファイル、ブロック デバイス ファイル) は作成できません。

- **チェックアウト済みバージョン:** チェックアウト済みバージョンは、VOB 内のエレメントの編集可能なバージョン コピーです。チェックアウト済みバージョンは、ビュープライベート ファイルとほぼ同じですが、VOB 内にバージョンに対応するオブジェクトが存在します。このオブジェクトは、特別な CHECKEDOUT バージョン ラベルの付いたプレースホルダー バージョンです。
- **非共有派生オブジェクト:** 非共有派生オブジェクト (DO) はデータコンテナで、`clearmake` や `omake` によって、または `clearaudit` で起動するプログラムによって作成されます。DO が共有可能になると、対応する DO が VOB 内に作成されます。

**メモ:** DO が共有可能になっても、DO のコピーはビューのプライベート記憶領域に残ります。DO はディスク容量を大量に消費することがあるので、不要になった DO は削除するようにしてください。詳細については、`winkin` と `view_scrubber` のリファレンス ページを参照してください。

- **共有不可派生オブジェクト**: 共有不可派生オブジェクトはデータコンテナで、**clearmake** や **omake** によって、または **clearaudit** で起動するプログラムによって、動的ビューに作成されます。DO に関する情報は、VOB データベースには保存されません。**winkin** または **view\_scrubber -p** を使用して共有不可 DO を共有可能 DO に変換すると、コマンドの実行によって DO のデータ コンテナが VOB にプロモートされ、ビュー記憶領域からデータ コンテナが削除されます。
- **構成レコード**: **.s** サブディレクトリの **view\_db.crs\_file** ファイルには、ビュー内の派生オブジェクト ビルドの構成レコードが保存されます。
- **取り残されたファイル**: **.s** サブディレクトリの **lost+found** ディレクトリには、取り残されたファイルが保存されます。アクセスに必要な VOB パス名がない場合、ファイルは取り残されます。たとえば、次のような場合です。
  - アンマウントなどによって VOB が一時的に使用不能になった場合
  - 削除などによって VOB が永久に使用不能になった場合
  - VOB ディレクトリが **rmelem** コマンドで削除されて永久に使用不能になった場合

取り残されたファイルの回復については、**recoverview** のリファレンス ページを参照してください。

## ビュー データベース

ビュー データベース サブディレクトリの **db** には、以下のファイルとサブディレクトリがあります。これらはすべて、**ClearCase** コマンドによって作成され、変更されます。これ以外の方法では変更できません。

<b>view_db.dbd</b>	データベースにアクセスするために組み込みデータベースルーチンによって使用されるコンパイル済みデータベーススキーマ。スキーマにはビュー データベースの構造が記述されています。
<b>view_db_schema_version</b>	コンパイル済みスキーマ ファイルが必要なりビジョン レベルであることを確認するために組み込みデータベース ルーチンによって使用されるスキーマ バージョン ファイル。
<b>view_db.d0n</b> <b>view_db.k01</b>	データベースの内容を保存するファイル。
<b>vista.*</b>	データベース コントロール ファイルとトランザクション ログ。

## view\_db.crs\_file

共有不可派生オブジェクトと非共有派生オブジェクトの構成レコードを保存するファイル。このファイルはビュー記憶ディレクトリの **.s** サブディレクトリにあり、リモートでアクセスできます。

構成レコードの圧縮コピーは、ビルドが開始されると、現在のディレクトリにあるビュープライベート ファイル **.cmake.state** にキャッシュされます。これらのコピーにより、次回以降のビルドの際、ビュー内の構成ルックアップが高速化されます。

ビュー データベースには、プライベート記憶領域内のビュープライベート オブジェクト、チェックアウト済みバージョン、共有不可派生オブジェクト、非共有派生オブジェクトなどのオブジェクトが記録されます。

## 動的ビューによるバージョンの選択

動的ビューの要素にアクセスするたびに、ビューの **view\_server** プロセスによってビューの構成仕様が評価され、使用可能にする要素のバージョンが決定されます。ビューサーバーでは、次の手順で要素名がバージョンに解決されます。

- 1 アプリケーション ソフトウェア (エディタやコンパイラなど) によって、パス名が参照されます。MVFS によって、VOB 内のすべてのパス名が適切な **view\_server** プロセスに渡されます。
- 2 **view\_server** によって、構成仕様の最初の規則に一致する要素のバージョンが検索されます。この検索で該当のバージョンが見つからないと、次の規則に進みます。一致するバージョンが見つかるまで、後続の規則に進みます。
- 3 一致するバージョンが見つかり、ビューサーバーによって該当のバージョンが選択され、MVFS によってバージョンのハンドルがアプリケーションに戻されます。

## 動的ビューによる派生オブジェクトの管理

動的ビューでは、ビュープライベート ファイル以外に、ビューのプライベート記憶領域にも、**clearmake** (Windows の場合は **omake**) によってビュー内に作成された派生オブジェクトが保存されます。共有不可派生オブジェクトと非共有派生オブジェクトは、一般にビューのプライベート記憶領域のほとんどのディスク容量を消費します。派生オブジェクトが作成されると、オブジェクトのデータ コンテナ ファイルと構成レコードがビューに保存されます。派生オブジェクトが初めて別のビューにウィンクインされるか VOB にプロモートされると、ビューは次のようにビューと VOB が動作します。

- 1 構成レコードが該当の VOB に移動します。ビルド スクリプトによって派生オブジェクトが複数の VOB に作成された場合は、各 VOB データベースに同じ構成レコードのコピーが作成されます。
- 2 データ コンテナが VOB の DO 記憶プールに ( 移動ではなく ) コピーされます。**clearmake** ビルドまたは **omake** ビルドによってウイंकインした場合は、データ コンテナに現在アクセス中のユーザー プロセスとの衝突を回避するために、元のデータ コンテナはビュー記憶領域に残ります。**winkin** コマンドまたは **view\_scrubber -p** コマンドでウイंकインした場合は、ビュー内のデータ コンテナは VOB 記憶プールにプロモートされてから削除されます。  
  
場合によっては、**view\_scrubber** ユーティリティを使用して重複するコンテナをビューから削除する必要があります。

## マルチバージョン ファイル システム

---

**メモ:** マルチバージョン ファイル システム (MVFS) は ClearCase LT ではサポートされていません。スナップショット ビューと Web ビューでは、MVFS を使用しません。

マルチバージョン ファイル システム (MVFS) は ClearCase の機能で、動的ビューをサポートします。動的ビューで MVFS を使用すると、特定のローカル ファイルとリモート ファイルの組み合わせを、ネイティブ ファイル システムに保存されているかのように表示できます。これらのファイルは、VOB ファイルとビュープライベート ファイルのバージョンです。MVFS を使用するには、ビューをアクティブにして、1 つまたは複数の VOB をマウントする必要があります (VOB を MVFS タイプのファイル システムとしてマウントします)。

MVFS は、ホストのネイティブ オペレーティング システムの拡張としてインストールされます。UNIX では、MVFS を実装するコードはホストのオペレーティング システムにリンクしています。静的にリンクする場合は、オペレーティング システムの新しいバージョンを生成して MVFS を含める必要があります。動的にリンクする場合は、MVFS コードはシステムの起動時にロードされます。MVFS がどのようにリンクされるかは、オペレーティング システムのタイプとバージョンによって異なります。

Windows では、MVFS はファイル システム ドライバです。MVFS は、システムの起動時にサービス コントロール マネージャによってロードされます。MVFS がインストールされている Windows ホストにユーザーがログオンすると、ユーザーの認証情報がキャッシュされます。この情報を使用して、ClearCase の管理下にあるオブジェクトのアクセス権が決定されます。Credentials Manager サービスは、定期的に認証情報キャッシュをチェックして、前回認証情報をチェックした後にログオフしたユーザーの認証情報を削除します。

## サポートされるファイル タイプ

MVFS では次のファイル タイプをサポートしています。

- ファイル
- ディレクトリ
- シンボリック リンク (UNIX の場合のみ)

UNIX の特殊ファイルなどのその他のファイル タイプは、動的ビューでは作成できません。

## MVFS と監視付きビルド

動的ビューでソフトウェアをビルドし、ビルド ツール (コンパイラなど) でパス名が参照されている場合、MVFS はパス名を適切な **view\_server** プロセスに渡します。ビルド スクリプトを実行すると、**clearmake** は MVFS を使用して下位レベルのシステム コールを監視します。このシステム コールは、ファイルまたはディレクトリが作成または削除されるか、読み取りアクセス用に開かれたときに、**ClearCase** データを参照し、各インスタンスを記録します。次のようなオブジェクトを呼び出すコールが監視されます。

- ビルド入力として使用されるエレメントのバージョン
- ビルド入力として使用されるビュープライベート ファイル (チェックアウト済みエレメントバージョンなど)
- ビルド中に VOB ディレクトリ内に作成されたファイル

## Windows での既知の MVFS 制限事項

既に判明している Windows での MVFS の制限事項を次に示します。

- システム ファイルや隠しファイルなどのファイル属性はサポートされません。
- OS/2 拡張属性 (EA) はサポートされません。
- DOS 共有モードはサポートされません。
- アプリケーションから MVFS ボリュームにディスク容量をクエリーすると、常に 512 MB という値が返されます。

これは、次のような理由からです。このアプリケーションは、ビュー記憶ディレクトリまたは VOB 記憶ディレクトリ内の領域を必要とする操作を実行しようとしています。MVFS では、アプリケーションがビュー記憶領域または VOB 記憶領域で使用可能なディスク容量を知る必要があることは分かりません。そのため、MVFS は常に 512 MB という値を返し、アプリケーションは必ずその操作を試行します。つまり、MVFS は成功するはずの操作を失敗させないようにしています。

- Windows での別名 (8.3 形式の短い名前) が VOB ネームスペースでサポートされません。  
別名が特定ディレクトリの別バージョンに既に存在する場合は、ClearCase によって誤ったファイルデータが返される可能性があるため、MVFS は別名を要求元のアプリケーションに返しませんが、
- 短い名前を必要とするアプリケーションを MVFS で実行する場合は、次のいずれかの方法を使用します。
  - 8.3 形式に完全に準拠しており、ビューと VOB のタグとディレクトリ名を含むパス名を使用します。
  - 必要に応じて、8.3 形式の名前を作成する VOB シンボリック リンクを使用します。

## MVFS での大文字と小文字の区別

Windows 上の MVFS では、名前のルックアップで大文字と小文字を区別するかどうかについて設定できます。デフォルトの設定では、MVFS で大文字と小文字を区別せずに名前をルックアップします。これは、Windows のアプリケーションに共通の動作です。Windows と UNIX が混在するネットワークで動的ビューを使用するときは、場合によっては Windows 上の MVFS における大文字と小文字の区別の設定を変更する必要があります。詳細については、65 ページの「大文字/小文字の区別」を参照してください。

## MVFS 内での実行可能ファイルの実行

実行可能ファイルをソース コントロールに追加し、動的ビューで実行するには、実行可能ファイルに対して実行権限を明示的に付与する必要があります。この場合は、`cleartool protect` コマンドを次のように使用します。

```
cleartool protect -chmod +x executable_filename
```

## MVFS のパフォーマンス

MVFS には、パフォーマンス向上のために使用するキャッシュがいくつかあります。ほとんどの場合、デフォルトのキャッシュ サイズによって、MVFS パフォーマンスとメモリ要件のバランスが最適化されています。特殊なニーズに合うように、これらのキャッシュを変更して MVFS を調整することもできます。詳細については、288 ページの「MVFS キャッシュ サイズの調査と調整」を参照してください。

## スナップショット ビュー

---

スナップショット ビューでは、ホストのネイティブ ファイル システムを使用して、VOB からロードされたファイルエレメントとディレクトリ エレメントのバージョンを保存します。スナップショット ビューを作成したら、ファイルエレメントとディレクトリ エレメントのバージョンをスナップショット ビュー ディレクトリにコピーまたはロードする必要があります。スナップショット ビュー ディレクトリは、通常はローカル ホスト上にありますが、ネットワーク上でビュー サーバーからアクセスできるホスト上に保存することもできます。この場合、そのホストに ClearCase がインストールされている必要はありません。

スナップショット ビュー ディレクトリには、バージョンのコピー以外に、ビュープライベート オブジェクトも保存できます。また、ビューの管理用に ClearCase によって作成される各種ファイルも保存されます。

スナップショット ビュー記憶ディレクトリには、次の 2 つのサブディレクトリがあります。

- **ビュー データベース:** db サブディレクトリに、組み込みデータベース管理システムによって管理されるバイナリ ファイルが保存されます。このデータベースでは、ビュー内にロードされた VOB オブジェクトとチェックアウトされたバージョンを追跡します。
- **管理ディレクトリ:** admin ディレクトリに、ビューで使用するディスク領域上のデータが保存されます。ClearCase スケジューラによって、このデータを生成するジョブが定期的に行われます。

スナップショット ビューに派生オブジェクトを含めることはできません。スナップショット ビューで **clearmake** と **omake** は実行できますが、ビルド監査機能とビルド回避機能は使用できません。

スナップショット ビューには、関連付けられた **view\_server** プロセスが必要です。このプロセスは、ローカル ホストまたは別のホストで実行できます。

**メモ:** ClearCase LT クライアントでは **view\_server** プロセスを実行できないため、ClearCase LT スナップショット ビューの **view\_server** プロセスは ClearCase LT サーバー上で実行されます。

スナップショット ビューの管理においては、スナップショット ビューのビュー記憶ディレクトリがスナップショット ビュー ディレクトリのサブディレクトリであるかどうかを知っておく必要があります。たとえば、スナップショット ビューのバックアップまたは移動には、スナップショット ビュー ディレクトリとビュー記憶ディレクトリの両方を含める必要があります。詳細については、186 ページの「ビューの移動」を参照してください。

## スナップショット ビュー ディレクトリ

スナップショット ビュー ディレクトリでは、ホストのネイティブ ファイル システムが使用されます (MVFS を使用する動的ビューとは異なります)。スナップショット ビュー ディレクトリのルート ディレクトリには、エレメントのコピー以外に、以下のファイルとサブディレクトリもあります。これらは、**ClearCase** コマンドによって作成され、変更されます。これ以外の方法では変更できません。

<b>view.dat</b>	ビューの一部である現在のディレクトリを識別するのに使用する読み取り専用のテキスト ファイル (UNIX の場合は <b>.view.dat</b> )。
<b>view.stg</b> または生成されたディレクトリ名	ビューの管理に使用するディレクトリ (UNIX の場合は <b>.view.stg</b> )。配列型のビュー記憶ディレクトリの場合、このディレクトリのデフォルト名は <b>view.stg</b> です。その他のビュー記憶ディレクトリの場合は、このディレクトリには生成された名前が付けられます。

## ビュー記憶ディレクトリ

ビュー記憶ディレクトリには、次のファイルとサブディレクトリがあります。これらは、**ClearCase** コマンドによって作成され、変更されます。これ以外の方法では変更できません。

<b>.access_info</b>	ビュー サーバーによって定期的に更新されるビュー アクセス イベント情報のファイル。
<b>.pid</b>	ビューの <b>view_server</b> プロセス ID を含む 1 行から成るテキスト ファイル。
<b>admin</b>	ビューが使用しているディスク容量に関連した管理データが保存されているディレクトリ。データを表示するには、 <b>cleartool space -view</b> コマンドを実行します。
<b>config_spec</b>	ビューの現在の構成仕様が保存されているファイル。内容を表示するには、 <b>catcs</b> コマンドを実行します。
<b>.compiled_spec</b>	<b>config_spec</b> の変更されたバージョン。アカウント情報が含まれています。
<b>.identity</b>	UNIX のサブディレクトリ。ビューの所有者とグループ メンバシップを確立する、ファイル サイズがゼロのファイルが保存されています。
<b>identity.sd</b>	Windows のセキュリティ記述子。FAT ファイル システムまたは FAT32 ファイル システムに保存されているビューに対して作成されます。NTFS ファイル システムに保存されているビューには、ファイル システム ACL のセキュリティ記述子が含まれるため、このファイルは必要ありません。
<b>.s</b>	ビューのプライベート記憶領域を実装するサブディレクトリ。
<b>db</b>	ビューの組み込みデータベースを実装するファイルが保存されているサブディレクトリ。
<b>.view</b>	ビューのユニバーサル ユニーク ID (UUID) とその他の属性のリストが保存されているファイル。
<b>view_db.state</b>	ビューの現在の状態を記録するファイル。

## ビュー データベース

ビュー データベース サブディレクトリの **db** には、以下のファイルとサブディレクトリがあります。これらはすべて、**ClearCase** コマンドによって作成され、変更されます。これ以外の方法では変更できません。

<b>view_db.dbd</b>	データベースにアクセスするために組み込みデータベースルーチンによって使用されるコンパイル済みデータベーススキーマ。スキーマにはビュー データベースの構造が記述されています。
<b>view_db_schema_version</b>	コンパイル済みスキーマ ファイルが必要なリビジョン レベルであることを確認するために組み込みデータベース ルーチンによって使用されるスキーマ バージョン ファイル。
<b>view_db.d0n</b> <b>view_db.k01</b>	データベースの内容を保存するファイル。
<b>vista.*</b>	データベース コントロール ファイルとトランザクション ログ。

スナップショット ビューでは、ビュー内にロードされた **VOB** オブジェクトとチェックアウトされたバージョンをビュー データベースで追跡します。

## スナップショット ビューによるバージョンの選択

スナップショット ビューの構成仕様とロード規則によって、**cleartool update** コマンドでスナップショット ビュー ディレクトリにロードするエレメントのバージョンが決定されます。ビューを更新するたびに、**view\_server** によってビューの構成仕様が評価されて、ロード規則に定義されている各エレメントのバージョンが選択されます。このとき、エレメントと選択されたバージョンが一致しないと、スナップショット ビューにロードされたエレメントの新しいバージョンがコピーされます。次の方法で、エレメント名がバージョンに解決されます。

- 1 ビュー サーバーによって、構成仕様の最初の規則に一致するエレメントのバージョンが検索されます。この検索で該当のバージョンが見つからないと、次の規則に進みます。一致するバージョンが見つかるまで、後続の規則に進みます。
- 2 一致するバージョンが見つかり、ビュー サーバーによって該当のバージョンが選択され、ビューのデータベースが更新され、バージョンのハンドルが **CCFS** サーバーに渡されます。**CCFS** サーバーでは、ハンドルがスナップショット ビュー ディレクトリにコピーされます。

## ビューの作成

---

ビューは一般に、ユーザーによって作成されます。ビューの作成プロセスは簡単なので、UNIX と Windows では GUI によって自動化でき、管理処理は通常必要ありません。ただし、共有ビューを作成したり、ビューの作成を管理するポリシーを確立したりする場合は、管理者による操作が必要になることがあります。ビューの作成の詳細については、**mkview** のリファレンスページを参照してください。

### リモート ビュー記憶領域

VOB 記憶領域と同様に、ビュー記憶領域は次のいずれかの方法でリモート ホスト (ビューの関連付けられた **view\_server** プロセスを実行していないホスト) 上に保存できます。

- 配列型のビュー記憶領域を伴わない (ビュー記憶ディレクトリがビュー サーバー ホスト上にあり、スナップショット ビュー ディレクトリが別のホスト上にある) スナップショットビューを作成できます。
- ビュー記憶領域の一部または全部を、認定されているネットワーク接続記憶デバイスに保存できます。
- UNIX ホストの場合、ビューのプライベート記憶領域 (データベースではない) をリモートに配置でき、UNIX シンボリック リンクを使用してアクセスできます。これは VOB 記憶プール機能 (118 ページの「UNIX ホストでのリモート記憶プールの作成」を参照) に似ていますが、ビューに関する機能についてはそれよりも単純です。

**メモ:** Windows では、ビュープライベート記憶領域がシンボリック リンク (**mkview -ln**) された UNIX ビューを使用できません。

### NAS デバイスでのビューの作成

ビューを NAS デバイス上に作成する場合は、最初にサーバーの記憶場所を作成することをお勧めします (100 ページの「NAS デバイス上でのサーバー記憶場所の作成」を参照)。次のコマンドは、**ccnasviewstg** という名前のサーバーの記憶場所に動的ビューを作成します。

```
cleartool mkview -tag viewtag -stgloc ccnasviewstg
```

**mkview** コマンドに **-host** オプション、**-hpath** オプション、**-gpath** オプションを指定して動的ビューを作成することもできます。動的ビューを NAS デバイス上に作成する例を以下に示します。ビューは、ClearCase ホスト **ccviewsvr-ux** (UNIX コンピュータ) 上で実行中の **view\_server** プロセスによって処理され、ビュー記憶領域は **ccviewsvr-ux** によって **/net/nasdevice** にマウントされた NAS デバイス上に保存されます。

```
cleartool mkview -tag nasview -host ccviewsvr-ux ¥  
-gpath /net/nasdevice/viewstg/nasview.vws -hpath ¥  
/net/nasdevice/viewstg/nasview.vws ¥  
/net/nasdevice/viewstg/nasview.vws
```

## ビューの移動

---

この項では、ビューを同じホストの別の場所または別のホストに移動する処理について説明します。移動する前にビューをバックアップする必要はありません。ビューの移動では、移動が完了し、移動先のビューにアクセスできるまでは元のビュー記憶ディレクトリを削除しないためです。

**注意:** ビュー記憶ディレクトリを移動する場合は、ファイルとディレクトリの所有権とアクセスコントロールの情報を変更しないコピーまたはバックアップ ソフトウェアを使用してください。これは、NTFS ファイルシステムを使用する Windows ホストでは特に重要です。詳細については、271 ページの「VOB または ビューの記憶ディレクトリをコピーするとき NTFS ACL を保持する」を参照してください。また、スナップショット ビューを移動する場合は、スナップショット ビュー ディレクトリのコピーに使用するユーティリティではロードされたファイルとディレクトリの変更時刻と所有権を保持する必要があります。これらが保持されないと、移動の完了後、該当のファイルとディレクトリはハイジャックされたとして報告されます。

### UNIX 上の動的ビューにリモート プライベート記憶領域を設定するかどうかの指定

この項で説明する処理では、UNIX ホスト上のビューに関連付けられている可能性のあるリモート プールを移動しません。UNIX ホスト上の動的ビューを移動する前に、シンボリック リンクされたプライベート記憶をビューに含めるかどうかを決定します (スナップショット ビューは、シンボリック リンクされたビュープライベート記憶領域をサポートしていません)。ビューの `.s` ディレクトリを表示するには、UNIX の `ls` コマンドを使用します。例を次に示します。

```
ls -ld /net/mars/viewstg/v5_integration.vws/.s
...    .s -> /public/view_aux/v5_integration
```

シンボリック リンクによって、該当のプライベート記憶領域がリモートにあることが示されます。移動したビューに同じ記憶領域を使用する場合は、新しいビュー ホストからアクセスできるようにする必要があります。

**メモ:** ビューを移動しても、ビュー記憶ディレクトリにある `.view` ファイルは変更されません。このファイル内の情報は、常にビューの最初の保存場所を表します。

## 同じアーキテクチャのホストまたは NAS デバイスへのビューの移動

ビューを、同じホスト上の別のパーティション、同じアーキテクチャを持つ別のホスト、NAS デバイスのいずれかに移動するには、次の手順に従います。

- 1 ビュー ホストにログオンします。ビューの所有者または権限を持つユーザーとしてログオンします。

- 2 ビューを非アクティブにします。移動中にビューを非アクティブにしておくには、`cleartool chview view_tag` コマンドに `-readonly` オプションを指定して実行します。これにより、ユーザーがビュー データベースを更新できなくなります。

```
cleartool chview -readonly r5_integration
```

プロパティ : 読み取り専用

- 3 ビューのサーバー プロセスを停止します。`cleartool endview view_tag` コマンドに `-server` オプションを指定して実行します。

```
cleartool endview -server r5_integration
```

- 4 ビューが再びアクティブにならないようにします。ビューのタグを削除してから、ビューの登録を解除します。ClearCase 管理コンソールまたは次のコマンドを使用します。

```
cleartool rmtag -view -all v5_integration
```

```
cleartool unregister -view /net/mars/viewstg/v5_integration.vws
```

**メモ:** ビュー記憶ディレクトリを移動しないスナップショット ビューの場合は、この手順と手順 6 は必要ありません。

- 5 ビューをコピーします。動的ビューを移動する場合は、ビュー記憶ディレクトリ全体を新しい場所にコピーします (ただし、シンボリック リンクされたプライベート記憶領域は含めません)。動的ビューでは、UNIX と Windows の手順は似ています。

- UNIX の場合、`tar` などのユーティリティを使用して、ビュー記憶ディレクトリをコピーします。次の例では、ビュー記憶ディレクトリ `v5_integration.vws` を `venus` という名前のホストにコピーします。

```
cd /viewstg
```

```
tar -cf - v5_integration.vws | rsh venus 'cd /viewstg ; tar -xBpf -'
```

**メモ:** `root` としてログオンしていない場合、ビューに不可欠なファイルとディレクトリを保護する `tar` コマンドの `-p` オプションは無視されることがあります。ネットワーク経由のコピーをサポートする `-B` オプションは、一部の UNIX プラットフォームではサポートされない場合があります。ご使用のオペレーティング システムのリファレンス ページを参照してください。

- Windows の場合、ファイルとディレクトリの所有権とアクセス コントロールの情報を保持するコピー ユーティリティ (271 ページの「VOB または ビューの記憶ディレクトリをコピーするとき NTFS ACL を保持する」を参照) を、ビューの移動先のホスト上で実行する必要があります。次の例では、ClearCase の `ccopy` ユーティリティを使用して、ビュー記憶ディレクトリ `v5_bugfix.vws` を `mars` という名前のホストからコピーします。

```
C:¥ClearCaseStorage¥views> net use E: ¥¥mars¥vws
```

```
C:¥ClearCaseStorage¥views> ccopy E:¥v5_bugfix.vws v5_bugfix.vws
```

- UNIX または WIndows でスナップショット ビューを移動する場合は、スナップショット ビュー ディレクトリかビュー記憶ディレクトリ、またはその両方を移動できます。Windows でスナップショット ビューを移動する場合は、189 ページの「Windows でスナップショット ビューを移動する場合の特別な手順」を参照してください。

- ビューを新しい場所に登録し、新しいビュー タグを作成します。ClearCase 管理コンソールまたは次のコマンドを使用します。

```
cleartool register -view /net/venus/viewstg/v5_integration.vws
```

```
cleartool mktag -view -tag v5_integration ¥  
/net/venus/viewstg/v5_integration.vws
```

**メモ:** ビューを NAS デバイスに移動する場合は、`-host` オプション、`-hpath` オプション、`-gpath` オプションを、`cleartool register` コマンドと `mktag` コマンドに指定して実行する必要があります。次の例では、NAS デバイスは ClearCase ホスト `venus` 上の `/net/nasdev` にマウントされていると仮定しています。

```
cleartool register -view -host venus -hpath ¥  
/net/nasdev/viewstg/v5_integration.vws ¥  
-gpath /net/nasdev/viewstg/v5_integration.vws ¥  
/net/nasdev/viewstg/v5_integration.vws
```

```
cleartool mktag -view -view -tag v5_integration -host ccviewsvr1 ¥  
-hpath /net/nasdev/viewstg/v5_integration -gpath ¥  
/net/nasdev/viewstg/v5_integration /net/nasdev/viewstg/v5_integration
```

- ビューを再びアクティブにし、テストします。ビューの所有者としてログオンしたまま、ビューにアクセスして、次のことを確認します。

- バージョンをチェックアウトでき、動的ビューにビュープライベート ファイルを作成できる。
- すべてのビュープライベート オブジェクトが移動されている。
- ハイジャックされているファイルがない (スナップショット ビューの場合のみ)。

- 8 ビューへの VOB 参照を更新します。ビューが保存されていた元の場所は、ビューがアクセスするすべての VOB に記録されています。この情報を更新するには、ビューに移動し、参照が保存されている各 VOB から該当の要素をチェックアウトします (チェックアウトは必要に応じて直ちにキャンセルできます)。
- 9 元のビュー記憶ディレクトリを削除します。移動したビューがアクセス可能で正常に動作することを確認したら、オペレーティング システムのユーティリティを使用して、元のビュー記憶ディレクトリを削除します。
- 10 必要に応じて、追加のタグを作成します。ビューにほかのネットワーク リージョンのタグが設定されている場合は、新しい場所を参照するタグに置き換えます。

## Windows でスナップショット ビューを移動する場合の特別な手順

Windows でスナップショット ビューを移動する場合は、ビュー記憶ディレクトリの ACL と、ロードされた要素の変更時刻を、スナップショット ビュー ディレクトリに保存する必要があります。

スナップショット ビュー ディレクトリは Windows の `xcopy` ユーティリティを使用してコピーできますが、ビュー記憶ディレクトリは ClearCase の `ccopy` ユーティリティ (271 ページの「VOB または ビューの記憶ディレクトリをコピーするとき NTFS ACL を保持する」を参照) を使用してコピーする必要があります。次の例では、ビュー記憶ディレクトリ `view.stg` はスナップショット ビュー ディレクトリ `v5_integration.vws` のサブディレクトリです。

```
C:\¥ClearCaseStorage¥views> net use E: ¥¥boron¥viewstg
C:\¥ClearCaseStorage¥views> xcopy /e/v/i/r/h/k v5_integration.vws E:¥v5_integration.vws
C:\¥ClearCaseStorage¥views> rmdir E:¥v5_integration.vws¥view.stg
C:\¥ClearCaseStorage¥views> ccopy v5_integration.vws¥view.stg E:¥v5_integration.vws
```

`xcopy /k` を実行すると、ロードされた要素の変更時刻が保存されますが、コピーしたスナップショット ビュー記憶ディレクトリに親ディレクトリの ACL が継承されるため、ビューへのアクセス時に問題が発生する場合があります。`ccopy` を実行すると、ClearCase に必要な NTFS ACL が保存されますが、ファイルの変更時刻は保存されません。

## 異なるアーキテクチャの UNIX ホストへのビューの移動

この項では、異なるバイナリ形式の UNIX ホスト間でビューを移動する処理について説明します。

異なるアーキテクチャの UNIX ホストにビューを移動する場合は、ビューを同じアーキテクチャの別のホストに移動するためのすべての手順と、ビュー データベース ファイルを新しいホストがサポートしているバイナリ形式に変換するための追加の実行する必要があります。

ビューを移動するには

- 1 ビュー ホストにログオンします。ビューの所有者または権限を持つユーザーとしてログオンします。
- 2 ビューを非アクティブにします。移動中にビューを非アクティブにしておくには、`cleartool chview view_tag` コマンドに `-readonly` オプションを指定して実行します。これにより、ユーザーがビュー データベースを更新できなくなります。

```
cleartool chview -readonly r5_integration
```

プロパティ : 読み取り専用

- 3 ビューのサーバー プロセスを停止します。`cleartool endview view_tag` コマンドに `-server` オプションを指定して実行します。

```
cleartool endview -server r5_integration
```

- 4 ビューが再びアクティブにならないようにします。ビューのタグを削除してから、ビューの登録を解除します。ClearCase 管理コンソールまたは次のコマンドを使用します。

```
cleartool rmtag -view -all v5_integration
```

```
cleartool unregister -view /net/mars/viewstg/v5_integration.vws
```

**メモ:** ビュー記憶ディレクトリを移動しないスナップショット ビューの場合は、この手順と手順 7 は必要ありません。

- 5 ビューのデータベースをダンプします。

```
cleartool reformatview -dump /net/mars/viewstg/v5_integration.vws
```

このコマンドによって、ビュー記憶ディレクトリに `view_db.dump_file` ファイルと `view_db.state` ファイルが作成されます。また、ビュー データベース サブディレクトリ名が `db.dumped` に変更されます。

- 6 ビューをコピーします。動的ビューを移動する場合は、ビュー記憶ディレクトリ全体を新しい場所にコピーします (ただし、シンボリック リンクされたプライベート記憶領域は含めません)。スナップショット ビューを移動する場合は、スナップショット ビュー ディレクトリかビュー記憶ディレクトリ、またはその両方を移動できます。次の例では、UNIX の `tar` ユーティリティを使用して、ビュー記憶ディレクトリを `venus` という名前のホストに移動します。

```
cd /viewstg
```

```
tar -cf - v5_integration.vws | rsh venus 'cd /viewstg ; tar -xBpf -'
```

**メモ:** `root` としてログオンしていない場合は、ビューに不可欠なファイルとディレクトリを保護する `tar` コマンドの `-p` オプションは無視されることがあります。ネットワーク経由のコピーをサポートする `-B` オプションは、一部の UNIX プラットフォームではサポートされない場合があります。ご使用のオペレーティング システムのリファレンス ページを参照してください。

- 7 ビューを新しい場所に登録し、新しいビュー タグを作成します。ClearCase 管理コンソールまたは次のコマンドを使用します。

```
cleartool register -view /net/venus/viewstg/v5_integration.vws
```

```
cleartool mktag -view -tag v5_integration ¥  
/net/venus/viewstg/v5_integration.vws
```

- 8 ダンプ ファイルからビュー データベースを再作成します。

```
cleartool reformatview -load -tag sv5_integration
```

- 9 ビューを再びアクティブにし、テストします。ビューの所有者としてログオンしたまま、ビューにアクセスして、次のことを確認します。

- バージョンをチェックアウトでき、動的ビューにビュープライベート ファイルを作成できる。
- すべてのビュープライベート オブジェクトが移動されている。
- ハイジャックされているファイルがない (スナップショット ビューの場合のみ)。

- 10 ビューへの VOB 参照を更新します。ビューが保存されていた元の場所は、ビューがアクセスするすべての VOB に記録されています。この情報を更新するには、ビューに移動し、参照が保存されている各 VOB から該当のエレメントをチェックアウトします (チェックアウトは必要に応じて直ちにキャンセルできます)。

- 11 元のビュー記憶ディレクトリを削除します。移動したビューがアクセス可能で正常に動作することを確認したら、`rmdir` や Windows エクスプローラなどのファイル システム ユーティリティを使用して、元のビュー記憶ディレクトリを削除します。新しいビュー ホスト上で、バックアップ ビュー データベース (通常は `db.dumped` という名前のデータベース) を削除します。

- 12 必要に応じて、追加のタグを作成します。ビューにほかのネットワーク リージョンのタグが設定されている場合は、新しい場所を参照するタグに置き換えます。

## NAS デバイスに対応するビュー サーバー ホストの置換

---

ビュー記憶領域を NAS デバイス上に配置すると、同じアーキテクチャを持つ別の ClearCase ホストにおいて、ビュー記憶領域を実際に移動せずに、ビューへのアクセスを管理する `view_server` プロセスを簡単に実行できます。

以下の手順では、UNIX ビュー サーバー ホストを動的ビュー `V4.1_Int` に対応するよう置換します。これにより、ビューは停止し、古いビュー タグとオブジェクトが削除され、代替ホストと既存の記憶領域を指定する新しいオブジェクトとタグが作成されます。

- 1 ビューのサーバー ホストにログオンします。ビューの所有者としてログオンします。
- 2 ビューを非アクティブにします。cleartool endview コマンドを使用して、ビューを停止し、ビューの view\_server プロセスを終了します。

```
cleartool endview -server V4.1_Int
```

- 3 既存のビュー タグを削除します。ClearCase 管理コンソールまたは次のコマンドを使用します。

```
cleartool rmtag -view V4.1_Int
```

- 4 新しいビュー サーバー ホストと既存の記憶領域を指定する新しいビュー オブジェクトとタグを作成します。ClearCase 管理コンソールまたは次のコマンドを使用します。

```
cleartool register -view -replace -host ccviewsvr1 ¥  
-hpath /net/nasdevice/viewstg/v4.1_int.vws ¥  
-gpath /net/nasdevice/viewstg/v4.1_int.vws /net/nasdevice/viewstg/v4.1_int.vws
```

```
cleartool mktag -view -tag V4.1_Int -host ccviewsvr1 ¥  
-gpath /net/nasdevice/viewstg/v4.1_int.vws ¥  
-hpath /net/nasdevice/viewstg/v4.1_int.vws /net/nasdevice/viewstg/v4.1_int.vws
```

- 5 ビューを再びアクティブにします。代替ビュー サーバー ホスト上のビューをアクティブにします。

```
cleartool startview V4.1_Int
```

## ビューのプロパティの表示

---

ビューのプロパティを表示するには、ClearCase 管理コンソールまたは **cleartool lsview** コマンドを使用します。プロパティには、ビューの作成日、最終変更日、所有者に関する情報が含まれています。ClearCase 管理コンソールを使用するか、**-long** オプションと **-full** オプションを指定して **lsview** コマンドを実行すると、次のような詳細情報を表示できます。

- ビューのテキスト モード
- ビューの保護
- ビューの所有権と権限との整合性
- 構成仕様の最終更新日と更新者
- ビューが動的ビューであるかスナップショット ビューであるか
- ビューが読み取り専用であるか書き込み可能であるか
- 動的ビューに対して、共有可能派生オブジェクトと共有不可派生オブジェクトのどちらを作成するか
- ビュープライベート データへの最終アクセス日とアクセス者
- ビュープライベート派生オブジェクトの最終更新日と更新者
- 動的ビューに対する派生オブジェクトの最終作成日と作成者、最終プロモート日とプロモート実行者、最終ウィンクイン日とウィンクイン実行者

これらの情報を利用することで、ビューが使用中かどうかや、必要な成果物がビューに含まれているかどうかを判断できます。

**メモ:** ClearCase 管理コンソールまたは `cleartool lsview` コマンドでビューのプロパティを変更すると、変更内容は直ちに反映されませんが、ほとんどの場合は 30 秒以内に更新されます。

## ビューのサービスの停止

---

ビューのサービスを一時的に停止するには、ビューの `view_server` プロセスを停止し、ビューのタグを削除します。ビューの `view_server` プロセスを停止するには、`cleartool endview` コマンドを使用します。

```
cleartool endview -server alh_main
```

ビューのサーバー プロセスを停止したら、ClearCase 管理コンソールまたは `cleartool rmtag` コマンドを使用して、ビュー タグを削除します。

## ビューのサービスの再開

ビューのサービスを再開するには、ClearCase 管理コンソールまたは `cleartool mktag` コマンドを使用して、ビューのタグを作成します。クライアントがビュー タグを参照すると、ビューのサーバー プロセスが起動します。

## ビューの削除

---

ClearCase レジストリのエン트리と VOB が保持するビューへのすべての参照を含め、ビューを完全に削除するには、ClearCase 管理コンソールを使用します。

- 1 該当のビューのビュー記憶ノードに移動します。これは、ビュー記憶ディレクトリが存在するホストのホスト ノードのサブノードです。
- 2 [操作] メニューの [すべてのタスク] をポイントし、[ビューの削除] をクリックします。

`cleartool rmview` コマンドを使用して削除することもできます。

単純に記憶ディレクトリを削除することによってビューを削除する場合は、次の手順に従って、VOB が保持しているビューへの参照を削除します。

- 1 `cleartool describe` コマンドを使用して、VOB が参照するビューのリストにあるビューの UUID をメモします。

```
cleartool describe -long vob:vob_tag
```

ビュー タグが残っている場合は、`lsview -long` を使用してビューの UUID を検索できます。

- 2 ビュー タグが残っている場合は、レジストリから削除します。**ClearCase** 管理コンソールで、該当のタグ リージョンの [ビュー タグ] ノードを使用して、ビュー タグを削除します。次のコマンドを使用して削除することもできます。

```
cleartool rmtag -view view_tag
```

- 3 ビューの登録を解除します。**ClearCase** 管理コンソールで、[ビュー オブジェクト] ノードを使用して、ビュー オブジェクトを削除します。以下のコマンドを実行して削除することもできます。手順 1 でメモしたビューの **UUID** を使用します。

```
cleartool unregister -view -uuid uuid
```

- 4 このビューを保持する各 **VOB** から、このビューへの参照を削除します。**ClearCase** 管理コンソールで、**VOB** 記憶ノードの [参照されているビュー] サブノードを使用して、ビューのレコードを **VOB** から削除します。以下のコマンドを実行して削除することもできます。手順 1 でメモしたビューの **UUID** を使用します。

```
cleartool rmview -all -uuid uuid
```

すべての **VOB** からビューへの参照を削除した後、残っているビュー記憶ディレクトリがあれば削除します。

# 重要な ClearCase データの バックアップ

# 10

貴重なデータのバックアップを頻繁に確実に実施することは、どの ClearCase 管理者にも共通する重要な作業です。すべてのデータベースと同様、VOB とビューにはバックアップと復元について特殊な要件があります。復元手順については、第 11 章で説明します。この章では、VOB、ビュー、ClearCase レジストリのバックアップ処理について説明します。

複数のリポジトリに格納されているデータは関連付けられていることが多いため、この章ではデータ間の関係を認識して、その関係を維持するバックアップルーチンの実装についても説明します。特に、UCM を使用している場合はこの点に留意してください。

## VOB とビューのバックアップ ツールに関する要件

---

VOB とビューのデータは、通常のファイルに保存され、一般的なディレクトリに格納されています。そのため、ファイルシステム バックアップに適したあらゆるツールを使用してデータをバックアップできます。ただし、必要なすべてのデータを取り込むツールであり、この章で説明する手順に従って処理する必要があります。VOB またはビュー用にバックアップ ツールを選択するときは、次の要件と推奨事項を考慮してください。

- **NTFS ACL を保持するバックアップ ツールであること。**Windows のプラットフォームでは、VOB またはビューの記憶領域が NTFS ファイル システムに基づいて配置されているため、VOB またはビューの記憶ディレクトリにある NTFS ACL はとても重要です。VOB とビューの記憶領域が FAT ファイル システムに基づいている場合 (このような環境は推奨しません) 以外は、NTFS ACL のバックアップと復元が可能な Windows バックアップ システム ツールを使用してください。
- **書き込みのため開かれているファイルであってもバックアップ可能なツールであること。**Windows 環境と UNIX 環境では、VOB がバックアップのためロックされている場合でも、VOB サーバー プロセスが VOB データベース ファイルを書き込み用に開いた状態に保ちます。したがって、VOB のバックアップ処理中に VOB をロックして VOB ホスト上の ClearCase を停止する場合を除き、書き込みのために開かれているファイルもバックアップ可能なツールを使用する必要があります。copy や xcopy などのファイル システム コピーユーティリティと同様に一般的な Windows バックアップ ツールの多くはこの機能を備えていないため、このようなファイルのバックアップをスキップしてしまい、バックアップが無意味になります。

- **ファイル アクセス時間を維持するバックアップ ツールであること。**UNIX プラットフォームの一部では、バックアップ時に一般的に使用されている **tar** ユーティリティによってファイル アクセス時間がリセットされてしまいます。その結果、**DO** 記憶プールと **cleartext** 記憶プールのスクラブ パターン (238 ページの「**VOB** 記憶プールのスクラブ」を参照) が破壊され、これらのプールがいつまでも削除されないことがあります。**VOB** バックアップには、UNIX **cpio** ユーティリティを使用することをお勧めします。

**メモ:** この章で説明するバックアップ処理の対象は、**NAS (Network Attached Storage)** デバイス上にデータベースが格納されている **VOB** またはビューです。**ClearCase** と共に使用可能な **NAS** デバイスに **VOB** 記憶ディレクトリがある場合、そのデバイス専用のバックアップ ユーティリティを使用できるため、**VOB** ロック時間を大幅に短縮できます。これに該当する **NAS** デバイスとそのバックアップ ユーティリティについては、オンラインの『**Rational ClearCase** プラットフォーム別情報ガイド』を参照してください。

## VOB のバックアップ方法

---

**VOB** のバックアップ方法を決めるときは、次の点を考慮してください。

- 必要なすべてのデータを取り込むことができるバックアップ ツールでなければなりません。
- バックアップ中、**VOB** はロックされる必要があります。ロックされた **VOB** によって、多くの開発作業が遅れるか中断します。そのため、バックアップ処理を業務時間外にスケジュールできない場合は、**VOB** のロック時間の短縮に重点を置き、最小限にとどめる必要があります。
- **VOB** が、ハイパーリンクで関連付けられた **VOB** のグループの一部であることがあるため、グループ内のすべての **VOB** をまとめてバックアップし、復元する必要があります。
- **UCM** と **Rational ClearQuest** を使用している場合、**PVOB** と **ClearQuest** データベースが相互に参照している可能性があります。このような参照関係が、**PVOB** または **ClearQuest** データベースが復元されたときに無効になることがあります。

**メモ:** **VOB** に適用するバックアップ ツールを選択する場合は、109 ページの「**VOB** 記憶ディレクトリ」で説明している、**VOB** 記憶ディレクトリのファイル システム レイアウトについて理解していると役に立ちます。

## 標準バックアップかスナップショット バックアップかの選択

バックアップは、次のいずれかまたは両方の方法を基盤とします。

- **標準バックアップ。****VOB** サーバー ホストのファイル システムをバックアップするときに、通常使用しているツールを採用します。
- **スナップショット バックアップ。****VOB** ロック時間を短縮するために **vob\_snapshot** ユーティリティを使用します。

標準 VOB バックアップ処理では、VOB をロックしてから VOB 記憶ディレクトリ全体をバックアップし、その後 VOB をアンロックする必要があります。バックアップ実行中は VOB がロックされている必要があります。書き込み用に開かれているファイルを取り込めないバックアップユーティリティを使用する場合は、ClearCase を停止する必要もあります。

VOB のロック状態をユーザーが容認できる場合は、標準バックアップ処理を使用して、一般的な VOB または UCM コンポーネットの VOB をバックアップすることをお勧めします。VOB をロックすると、チェックイン、チェックアウトなどの VOB データやメタデータに影響を及ぼす操作が実行できなくなります。これらの操作には、UCM デリバー操作やマージを必要とする UCM リベース操作があります。VOB がロックされていても、チェックアウトまたは (スナップショットか Web ビュー内で) ハイジャックされたバージョンを操作できます。clearmake と omake は、ロックされた VOB に対処できるように設計されています。これらのユーティリティは、ロックされた VOB 内でオブジェクトを開けない場合に「スリープ」状態になり、後でバックアップを再試行するように設定できます (clearmake リファレンス ページを参照)。標準バックアップ処理を使用した場合の利点は次のとおりです。

- 書き込み用に開かれたファイルをバックアップできないバックアップ ツールを使用できます。ただし、バックアップを実行する前に VOB ホスト上の ClearCase を停止する必要があります。ClearCase の停止と起動にかかる時間は、VOB のバックアップ所要時間に大きな影響を与えることはありません。ClearCase が停止している間は、ホスト上の VOB またはビューにアクセスできません。
- バックアップ先が別の物理デバイスでも、特別なディスク領域は必要ありません。
- VOB の復元処理が簡略化されます。VOB データベースと記憶プールが 1 つのユニットとしてバックアップされるため、復元時にデータを消失することはほとんどありません。

スナップショット バックアップは、VOB データベースのみをバックアップするため、VOB がロックされている時間を大幅に短縮できます。容量の大きい VOB には特に効果的です。

vob\_snapshot ユーティリティは、スナップショット バックアップの最も重要な部分の自動化に役立ち、ClearCase のスケジュールされたタスクとして簡単に実行できます。vob\_snapshot を使用する場合は、次の点に注意してください。

- 書き込み用に開かれているファイルもバックアップできるバックアップ ツールを使用する必要があります。VOB がロックされないので、vob\_snapshot の実行完了後に VOB 記憶ディレクトリの残りの部分をバックアップする必要があります。バックアップ中に VOB が使用されている場合、VOB 記憶ディレクトリ内の 1 つまたは複数のファイルが書き込み用に開かれている可能性があります。このようなファイルをバックアップしないと、VOB を復元したときにそのデータが消失してしまいます。
- より多くのディスク容量が必要になります。VOB データベースのコピーをディスクに書き込む必要があります。書き込み先は、原則としてローカル ホスト上になります。さらに、VOB のソース記憶プールの各データ コンテナは、新しいバージョン データを持つコンテナで置き換えられた後も 30 分間ディスク上に残されます。これにより、VOB の復元時に checkvob が VOB データベースと記憶プールを再同期させるとき、そのソース コンテナを再構築できる確率が上がります。

- VOB の復元処理がより複雑になります。VOB 記憶プールと VOB データベースが別々にバックアップされるため、VOB を復元したときに同期させる必要があります。特に、データベースのスナップショット作成と記憶プールのバックアップの間に DO やバージョン データの追加または削除が行われると、データベースと記憶プールに不整合が発生するので、**checkvob** ユーティリティを使用して不整合を解消する必要があります。
- VOB の復元時に一部のデータが失われることがあります。復元された記憶プールが復元された VOB データベースより古い場合は、記憶プールにないデータは当然失われます。復元された記憶プールの方が復元された VOB データベースより新しい場合は、記憶プールにあるバージョン データのうちスナップショットより新しいものは復元されたデータベースには登録されていません。250 ページの「**checkvob -force -fix** を使用したデータ消失の最小化」では、このようなリスクをさらに詳しく説明しています。

**メモ:** コンポーネントがまったく格納されていない PVOB のプールにはデータがないので、データ消失のリスクを負うことなく **vob\_snapshot** を使用してバックアップできます。PVOB のバックアップには **vob\_snapshot** を使用することをお勧めします。

## **vob\_snapshot を使用した VOB のバックアップ**

---

データベース スナップショットを有効にするには、**vob\_snapshot\_setup** を実行します。このコマンドを実行すると、ClearCase スケジューラが VOB データベースに対して定期的に (デフォルトでは毎日) **vob\_snapshot** を実行するようになります。操作については、**vob\_snapshot\_setup** と **vob\_snapshot** のリファレンス ページを参照してください。VOB に対してスナップショットバックアップを有効にした場合は、必ず次の追加処理を実行して、バックアップが完了したことを確認し、データ消失を最小限に食い止める必要があります。

- 各 **vob\_snapshot** が完了したら、すぐに、すべてのリモート記憶プール (201 ページの「VOB にリモート記憶プールがある場合」を参照) を含む VOB 記憶ディレクトリ全体をバックアップします。このバックアップ中は、VOB をロックする必要はありません。**vob\_snapshot** の完了後 30 分以内にこのバックアップを完了すると、復元時にデータが消失する可能性が低くなります。また、バックアップの対象から VOB データベース (db ディレクトリ) 自体を除外すると、バックアップ時間を短縮できます (VOB データベースは **vob\_snapshot** の実行時にバックアップされています)。
- スナップショット自体をバックアップします。この処理は、スナップショットが、VOB 自体と同じ物理デバイス内の一角にコピーされている場合は特に重要です。そのデバイス自体が壊れた場合に、スナップショットがほかのメディアにバックアップされているか、別のデバイスにコピーされていないと、バックアップが無意味になってしまうためです。

## ソース コンテナの削除の遅延

`vob_snapshot_setup` プログラムによって削除の遅延が有効化されます。これにより、`vob_snapshot` の完了後 30 分以内に実行するプール バックアップが効率的に実施され、データベース スナップショットが参照しているすべてのソース コンテナがこのバックアップに含まれます。コンテナが新しいバージョンのデータで置き換えられると (チェックイン中など)、新しいコンテナが作成され、古いコンテナは削除されます。このとき削除の遅延が有効化されていると、古いコンテナが削除保留リストに追加されて 30 分後に削除されます。

削除の遅延を有効化すると、アクティブな VOB で必要なディスク容量が増加します。UNIX プラットフォームでは、`vob_server` プロセスで `kill -HUP` コマンドを実行することにより、このプロセスのログ ファイルに削除の遅延統計を送ることができます。

`checkvob` によってソース プールが調査されると、削除の遅延リストに掲載されているすべてのコンテナが報告されます。

削除の遅延リストは、VOB 記憶ディレクトリに保存されているファイル `delete_list.db` に 5 分ごとに書き込まれます。

## 標準バックアップ処理を使用した VOB のバックアップ

---

`vob_snapshot` ユーティリティを使用しない場合は、この項で説明する処理に従って VOB バックアップの整合性を確認する必要があります。どの VOB バックアップ処理でも、次の処理を必ず行ってください。

- 1 すべてのユーザーに対して VOB をロックします。
- 2 すべてのリモート プールを含む VOB 記憶ディレクトリ全体をバックアップします (必要に応じて、VOB 記憶ディレクトリの特定のサブディレクトリをバックアップ対象から除外できます。詳細については、201 ページの「すべてをバックアップできない場合にバックアップ対象にするもの」を参照)。
- 3 VOB をアンロックします。

## VOB 記憶ディレクトリの場所の特定

VOB 記憶ディレクトリの場所を特定するには、ClearCase 管理コンソールまたは `cleartool` `lsvob` コマンドを使用します。

```
cleartool lsvob -long /vobs/sources
```

```
タグ: /vobs/sources
```

```
グローバル パス: /net/neptune/vobstg/sources.vbs
```

```
.  
. .  
.
```

```
ホスト上の VOB: neptune
```

```
VOB サーバー アクセス パス: /vobstg/sources.vbs
```

バックアッププログラムをローカルで実行する場合は、VOB サーバー アクセス パスを使用して VOB にアクセスします。バックアッププログラムをネットワーク上で実行する場合は、グローバル パスを使用します。バックアッププログラムのパス名を指定します。

## VOB のロックとアンロック

VOB バックアップを実行できるのは、バックアップ開始前にすべてのユーザーに対して VOB がロックされた場合のみです。

**注意:** どのバックアップ ツールまたは方法を選択した場合でも、すべてのユーザーに対して VOB をロックする必要があります。`-nusers` オプションを使用して VOB をロックすると、データベース チェックポイントが実行されません。このチェックポイントがないと、バックアップは無効です。

VOB をロックまたはアンロックするには、VOB の所有者または権限を持つユーザーである必要があります。VOB のロック処理をサポートする GUI がいくつかあり、`cleartool lock` コマンドと `unlock` コマンドと同様に機能します。

- UNIX または Windows のどのホストでも、ClearCase 管理コンソールから次のように操作すると VOB をロックまたはアンロックできます。
  - a VOB の VOB 記憶ノードに移動します。これは、VOB 記憶ディレクトリが存在するホストのホスト ノードのサブノードです。
  - b [操作] メニューの [プロパティ] をクリックします。[VOB のプロパティ] ダイアログボックスで [ロック] タブをクリックします。
- Windows ホストでは、Windows エクスプローラで VOB をロックまたはアンロックできます。[ClearCase] を右クリックし、[プロパティ] をクリックします。[VOB のプロパティ] ダイアログボックスで [ロック] タブをクリックします。
- コマンド行からは、`cleartool lock` コマンドと `unlock` コマンドを実行します。

## VOB にリモート記憶プールがある場合

UNIX ホストの VOB にリモート記憶プールがある場合は、採用したバックアップ方法 (標準または `vob_snapshot`) に関係なく、そのプールもバックアップする必要があります。

すべてのビュー内で `lspool` コマンドを次のように使用して、リモート記憶プールを特定します。

```
cleartool lspool -invob vob:/vobs/flex
13-Jan.16:58   vobadm      pool "cdft"
    "Predefined pool used to store cleartext versions."
26-Jan.22:02   vobadm      pool "cltxt01"
    "remote cleartext storage pool for 'flex' VOB"
13-Jan.16:58   vobadm      pool "ddft"
    "Predefined pool used to store derived objects."
13-Jan.16:58   vobadm      pool "sdft"
    "Predefined pool used to store versions."
```

この例では、リモート プールは `cltxt01` だけです。どの記憶プールがリモートなのかを判別できない場合は、次のように `lspool -long` コマンドを入力して、すべてのプールのプール記憶グローバルパス名を一覧表示します。表示されたパス名を確認して、リモート ロケーションを特定します。

```
cleartool lspool -long -invob vob:/vobs/flex
pool "cltxt01"
.
.
プール記憶グローバル パス名
"/vobstore/flex.vbs/c/cltxt01"
.
.
```

**メモ:** `tar` や `cpio` などの UNIX バックアップ ユーティリティの多くには、シンボリック リンクを追跡するオプションが含まれています。バックアップ時にリモート ホストがアクセス可能な場合は、これらのユーティリティを使用してシンボリック リンクとその内容の両方をコピーできます。この機能は、リモート プールを持つ VOB をバックアップする場合に便利です。

## すべてをバックアップできない場合にバックアップ対象にするもの

ファイル指向のバックアップ プログラムを使用する場合、時間を節約するために VOB 記憶ディレクトリ内の特定のサブディレクトリを除外することがあります。表 7 に、各ディレクトリの相対的な重要度を判定するガイドラインを示します。

**表 7 部分バックアップでの VOB ディレクトリの重要度**

VOB ディレクトリ	バックアップの重要度
最上位 VOB 記憶ディレクトリ	必須
VOB データベース サブディレクトリ	必須
ソース記憶プール	必須
派生オブジェクト記憶プール	必須ではないが重要 (ClearCase LT では不要)
cleartext 記憶プール	任意
管理ディレクトリ	任意

## 派生オブジェクト (DO) プールのバックアップ

**メモ:** ClearCase LT では派生オブジェクトをサポートしていません。ClearCase LT VOB の DO プールは常に空なのでバックアップする必要はありません。

MVFS や clearmake、omake、clearaudit などのプログラムを使用する場合以外は、派生オブジェクトの記憶プールが存在しません。派生オブジェクトの記憶プールをバックアップする必要はありません。ソースから DO を再ビルドできるよう定義されているためです。このようなプールのバックアップの重要性は、次のように、その状況によって変化します。

- プロジェクトの初期の段階では、ソース ベースが急速に変化するので、多くの派生オブジェクトのライフ サイクルは非常に短くなります。この段階では、バックアップ処理の対象から DO 記憶プールを除外しても問題ありません。
- プロジェクトが比較的安定してくると、DO 記憶プールには、頻繁に再利用されるオブジェクトが多く含まれるようになります。この段階で DO 記憶プールを消失すると、次回完全なシステムをビルドするのに非常に時間がかかることがあります。それ以降のビルドでは新しく作成された DO を再利用できるため影響はありません。
- 成熟したプロジェクトでは DO を再ビルドするのが難しいため、DO 記憶プールをバックアップ対象に入れます (その時点で使用できないコンパイラが必要な場合、または削除したビューが必要な場合があるためです)。このような DO は重要なので、定期的にバックアップするか、エレメントとしてバージョン付けする必要があります。

DO プールをバックアップしない場合でも、少なくともプールのルート ディレクトリ (d\ddft など) と pool\_id ファイルをバックアップすることをお勧めします。これにより、checkvob を実行したときの、復元後のプールルート チェック障害を防ぎます。

**注意:** 共有派生オブジェクトは、VOB データベースのオブジェクトと DO 記憶プールのデータ コンテナという 2 つの部分で構成されています。バックアップの失敗などが原因で DO データ コンテナを消失すると、VOB のデータベースが DO 記憶プールと同期できなくなります。再同期するには、空のすべての DO を VOB データベースから削除する必要があります。次に、cleartool rmdo か checkvob、またはその両方を使用します。

## cleartext 記憶プールのバックアップ

cleartext 記憶プールのバックアップは重要ではありません。タイプ マネージャが必要に応じて cleartext データ コンテナを再作成するためです。

cleartext プールをバックアップしない場合でも、少なくともプールのルートディレクトリ (c¥cdf など) と pool\_id ファイルをバックアップすることをお勧めします。これにより、checkvob を実行したときの、復元後のプール ルート チェック障害を防ぎます。

## 管理ディレクトリのバックアップ

admin ディレクトリには、VOB とその派生オブジェクトが使用するディスク容量に関するデータが入っています。ClearCase スケジューラは、ディスク使用量に関するデータを収集して、その結果をこのディレクトリに保存するジョブを定期的に行います。デフォルトでは、過去 30 日間のデータが保存されます。この履歴データは再作成できないので、このデータが必要な場合は admin ディレクトリもバックアップしてください。

## インクリメンタル バックアップを推奨できない理由

エレメントの新しいバージョンがチェックインされると、ClearCase では既存のデータ コンテナが更新されるのではなく、新しいコンテナがソース記憶プール内に別のパス名で作成されます (その後、古いコンテナは削除されます)。ほとんどのインクリメンタルバックアップ スキーマでは、基本的に、前回のバックアップ以降に更新されたファイルのみをバックアップするため、このメカニズムでは意味がありません。VOB をインクリメンタルバックアップすると、すべてのコンテナが取り込まれ、復元時には故意に削除したコンテナも復元されます。つまり、VOB 記憶ディレクトリには削除したはずの不要なデータが格納されることになります。

たとえば、あるエレメントの新しいバージョンが毎日作成されているとします。毎日実行されるインクリメンタル バックアップにより、そのエレメントのその日のソース コンテナがバックアップされます。週の最終日にこのバックアップを復元すると、最新の状態の VOB データベースに対応するコンテナが 1 つだけであっても、すべてのコンテナが記憶プールに復元されます。続いて checkvob を実行すると、参照不能データ コンテナが多く報告されるので、そのコンテナを削除する必要があります。

このような理由から、VOB 記憶ディレクトリのインクリメンタル バックアップをお勧めすることはできません。インクリメンタル バックアップの実行を避けられない場合は、次回フルバックアップを実行するまでの間、インクリメンタルバックアップの実行回数をできるだけ少なくしてください。

## ビューのバックアップ

---

ビューの内容は、VOB と異なり、通常は簡単に再構築できます。チェックアウト済みのバージョンとほかのビュープライベート ファイル以外は、どのビューも再作成すると復元できます。しかし、ビューのバックアップを定期的に行うことが重要であることに変わりはありません。ユーザーに自分の仕事を定期的に見直す習慣がない場合など、特に重要です。ビューのバックアップは VOB のバックアップと似ていますが、通常は VOB よりも簡単です。

- ビューは複数の記憶プールを持ちません。動的ビューは、プライベート記憶領域を 1 つだけ持ちます。それは **.s** サブディレクトリです。ビューの記憶領域が UNIX ホスト上にある場合、リモート ホストへのシンボリック リンクとして実装されている可能性がある場合、NAS デバイス上にある場合を除き、このサブディレクトリはビュー サーバーが実行されているホストに対してローカルである必要があります。スナップショット ビューまたは Web ビュー内にロードされたエレメント バージョンは、1 つのディレクトリとそのサブディレクトリ (この中に **.s** ディレクトリはありません) に保管されます。
- ビュー記憶ディレクトリまたはスナップショット ビュー ディレクトリに対して部分バックアップを実行することはお勧めできません。これらのディレクトリに格納されているデータはすべて重要です。特に、修正したチェックアウト ファイルとほかのビュープライベート ファイルは VOB に記録されないため重要です。

**注意:** ビュー記憶ディレクトリのバックアップに使用するユーティリティは、書き込み用に開かれているファイルもバックアップできる必要があります。ビュー データベース ファイルは、一般的に、ビューが無効化されているか読み取り専用であっても、ClearCase が稼働している間は書き込み用に開かれています。このようなファイルをスキップするバックアップはあまり価値がありません。したがって、書き込みアクセス用に開かれたファイルもバックアップ ソフトウェアの処理対象になることが確実でない場合 (Windows ユーティリティの多くでは対象外になります) は、ビュー ホスト上の ClearCase を停止してからバックアップを実行する必要があります。

スナップショット ビューまたは Web ビューのビュー記憶ディレクトリの場所が、ダウンロードされたファイルが格納されているディレクトリとは異なり、別のホストなどにある場合があります (Web ビューは、ClearCase LT のスナップショット ビューと同様に、通常このような構成になっています)。このような場合は、両方のディレクトリをバックアップする必要があります。

ビューをバックアップするには

- 1 **ビュー記憶ディレクトリの場所を特定します。** ClearCase 管理コンソールを使用するか、lsview を実行して、ビュー記憶領域情報を表示します。バックアップ プログラムをネットワーク上で実行する場合は、グローバル パスが必要です。バックアップ プログラムをローカルで実行する場合は、ビュー サーバー アクセス パスが必要です。

```
cleartool lsview -long r5_integration
```

```
タグ: r5_integration
```

```
グローバル パス: /net/mars/viewstg/r5_integration.vws
```

```
...
```

```
...
```

```
ホスト上のビュー: mars
```

```
ビュー サーバー アクセス パス: /viewstg/r5_integration.vws
```

```
...
```

- 2 バックアップの整合性を保証します。バックアップが終了するまでビューを無効化するため、**chview** コマンドを使用して、ユーザーがビュー データベースを更新できないようにします。

```
cleartool chview -readonly r5_integration
```

プロパティ: 読み取り専用

このコマンドでは、ビューの構成仕様に対する変更を防止することはできません。バックアップ中に構成仕様が変更されないようにするには、バックアップの実行前にビュー記憶ディレクトリの名前を変更し、**cleartool endview -server view-tag** を使用してビュー サーバーを停止します。

- 3 ビューが UNIX ホスト上にある場合は、そのビューにリモート記憶領域があるかどうかを確認します。それには、次のように UNIX の **ls** コマンドを使用します。

```
cd /viewstg/r5_integration.vws
```

```
ls -ld .s
```

```
... .s -> /net/ccsvr04/viewstore/r5_integration.stg
```

シンボリック リンクはリモート プライベート記憶領域を示します。

- 4 バックアップ コマンドを入力します。動的ビューの場合、またはビュー記憶ディレクトリがスナップショット ビュー ディレクトリのサブディレクトリにあるスナップショット ビューの場合は、ビュー記憶ディレクトリ全体をバックアップします。手順 1 の **lsview** によって一覧表示されるローカル パスまたはネットワーク パスを使用します。

バックアップ中のスナップショット ビューまたは **Web** ビューのビュー記憶ディレクトリがビュー ディレクトリのサブディレクトリではない場合は、ビュー ディレクトリとビュー記憶ディレクトリの両方をバックアップします。

**メモ:** スナップショット ビューをバックアップまたは復元するときは、ビュー内にあるすべてのファイルとディレクトリの更新日時と所有者情報を維持するユーティリティを使用してください。維持できない場合は、ロードしたファイルがハイジャックされてしまいます。

- 5 手順 2 でビューを読み取り専用に設定した場合は、それを書き込み可能にします。

```
cleartool chview -readwrite r5_integration
```

プロパティ: 読み取り書き込み

- 6 手順 2 でビュー記憶ディレクトリの名前を変更した場合は、元の名前に戻します。

## ClearCase のレジストリ データのバックアップ

---

ClearCase のレジストリ データは、レジストリ サーバー ホストの **rgy** ディレクトリに格納されているファイルのグループに保存されています。レジストリ データのファイルは、バックアップ可能であり、定期的にバックアップする必要のある通常のファイルです。

- UNIX ホストのレジストリ ディレクトリは **/var/adm/rational/clearcase/rgy** です。
- Windows ホストのレジストリ ディレクトリは **ccase-home-dir\var\rgy** です。

Windows でも UNIX でも、レジストリ ディレクトリに格納されているすべてのファイルをバックアップする必要があります。まとめて復元する必要があります。**rgy** ディレクトリをバックアップするだけでなく、レジストリ クライアント リストもバックアップすることをお勧めします。このリストは **rgy** ディレクトリではなく、次のディレクトリにあります。

- UNIX ホストでは **/var/adm/rational/clearcase/client\_list.db** に格納されています。
- Windows ホストでは **ccase-home-dir\var\client\_list.db** に格納されています。

レジストリ データのバックアップには、書き込み用に開かれているファイルをバックアップできるユーティリティを使用することをお勧めします。

## 関連付けられたデータベースの一括バックアップ

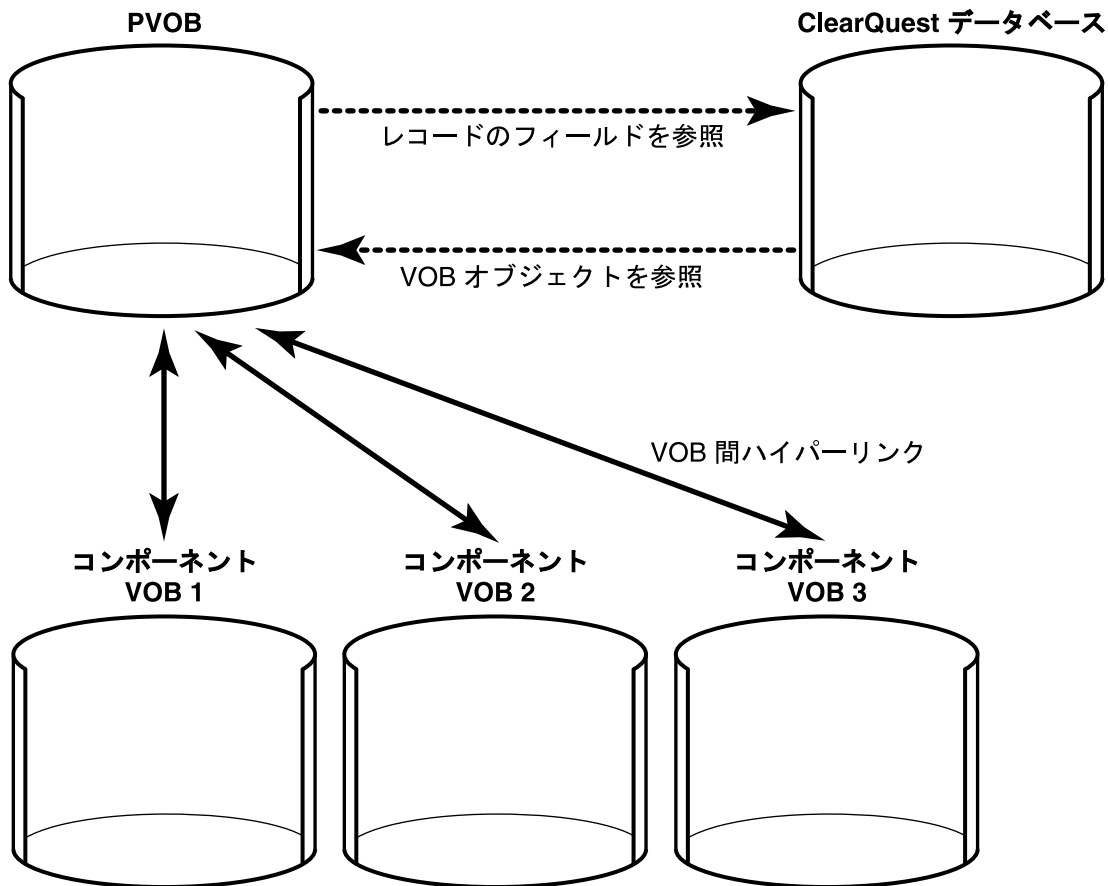
---

複数のレポジトリに格納されてるデータは、相互に関連付けられてるものが多く、特に UCM を使用している場合はその点に留意する必要があります。一般的な関係を次に示します。

- UCM PVOB とコンポーネント VOB は参照関係にあり、ハイパーリンクで関連付けられています。
- 管理 VOB の階層 (第 7 章を参照) はハイパーリンクで関連付けられています。
- UCM と Rational ClearQuest を使用する環境では、PVOB には ClearQuest データベース内のレコードに対する参照が、ClearQuest データベースには PVOB 内のオブジェクトに対する参照が格納されています。

図 6 に、シンプルな構成でのデータの関係を示します。

図 6 データベースの関係



UCM の操作であるプロジェクトへの参加、アクティビティの作成、ストリームのデリバリーまたはリベースなどによって、複数の関連付けられたデータベースに格納されたデータや、場合によっては関係自体が変更されることがあります。UCM を使用していない場合でも、管理 VOB の階層内での変更処理 (たとえば、グローバル タイプやローカル コピーの作成など) は、1 回の操作で複数の VOB に影響を与えます。関連付けられたデータベースの最初のメンバグループがバックアップされてから最後のメンバがバックアップされるまでの間隔が長いほど、バックアップを復元したときのメンバ間のずれが大きくなってしまいます。

図 6 に示したような構成での理想的なバックアップ方法は、次のような手順で構成されます。

- 1 すべての VOB と ClearQuest データベースをロックします。
- 2 すべての VOB と ClearQuest データベースをバックアップします。
- 3 すべての VOB と ClearQuest データベースをアンロックします。

最もシンプルな構成 (小さな VOB 数個と小規模の ClearQuest データベース 1 つ) でも、この処理に時間がかかることがあります。また、関連付けられたすべてのデータベースを 1 つのイベントでバックアップできても、まとめて復元できるとは限りません。関係グループのデータベースのうち 1 つか 2 つが、ほかのデータベースがそのまま残っている環境内に復元される可能性が高く、結果的に、復元されたものと既存のもの間にずれが生じてしまうことがあります。このようなずれの多くは、225 ページの「関連付けられたデータベースのグループに属するメンバの復元」で説明するツールと手順を使用して調整できます。

より実用的なバックアップ方法は、次のような手順で構成されます。

- 1 コンポーネント VOB をバックアップします。可能な場合は、**vob\_snapshot** ツールを使用して各 VOB のバックアップにかかる時間を短縮します。
- 2 ClearQuest データベースをバックアップします。
- 3 **vob\_snapshot** を使用して PVOB をバックアップします。コンポーネントが格納されていない PVOB のプールにはデータもないので、**vob\_snapshot** を使用してバックアップしてもデータ消失のリスクを負うことはありません。構成内に共通管理 VOB を共有する複数の PVOB がある場合は、すべての PVOB と管理 VOB を 1 つのユニットとしてバックアップしてください (すべての PVOB とその管理 VOB をロックし、バックアップしてからアンロックします)。

この方法では、復元時に、PVOB が ClearQuest データベースやコンポーネント VOB よりも新しくなる可能性が大きくなり、結果的にデータベース間で生じるずれの調整が簡単になります。

どのバックアップ方法の場合も、次の点に留意してください。

- データベースに関連付けられたグループのすべてのメンバに対して、メンバ間のバックアップ間隔をできるだけ短くするようにします。
- 頻繁にグループ全体をバックアップします。復元時には、前日にバックアップしたデータを使用した方がそれ以前のデータを使用するよりもずれを最小限にとどめることができます。

## データベースの関係の決定

UCM を使用していない環境では、主に管理 VOB の階層構造によってデータベースの関係が決定されます。ClearCase 管理コンソールまたは **cleartool describe** を使用して、VOB に対する **AdminVOB** ハイパーリンクを表示します (128 ページの「管理 VOB 階層とグローバル タイプ」に説明するように、VOB には、管理 VOB の階層内の上位を指す **AdminVOB** ハイパーリンクが確実に 1 つ含まれています。ほかに、階層内の別の VOB に対する **AdminVOB** ハイパーリンクが含まれていることもあります)。

UCM を使用する環境では、プロジェクトの組織によってデータベースの関係が決定されます。プロジェクト内の変更可能なすべてのコンポーネント、その PVOB、コンポーネントで使われる ClearQuest データベースは関連付けられているので、まとめてバックアップする必要があります (変更不可能なコンポーネントもバックアップする必要がありますが、PVOB によってこれらのコンポーネントに格納されている参照は長期間有効な場合が多いので、ほかのコンポーネントと同じ間隔でバックアップする必要はありません)。

プロジェクト エクスプローラまたは **cleartool lsproject** を使用して、コンポーネントをプロジェクトに表示できます。



# 重要な ClearCase データの 復元

# 11

ClearCase データに関して適切なファイルのバックアップ対策を実行していれば、必要に応じてデータを復元するのは簡単なことです。どのデータベースにも当てはまることですが、ClearCase リポジトリには 特有のバックアップ要件と復元要件があります。バックアップ要件については、第 10 章を参照してください。この章では、VOB、ビュー、ClearCase レジストリの復元の処理について説明します。

複数のリポジトリ内のデータは、UCM を使用している場合は特に、互いに関連していることが多いので、この章でも、関連するリポジトリの既存のグループ メンバと復元したグループ メンバの間の不整合の検出と修正の処理について説明します。

## 復元の準備

---

VOB、ビュー、ClearQuest データベース (UCM 環境で Rational ClearQuest を使用している場合) を復元すると、過去の成果物が別の環境に取り込まれますが、その過去の成果物の履歴が重大な影響を及ぼすことがよくあります。このようなデータベースを復元する場合は、コミュニティ内のほかのデータベースでの復元による影響を考慮する必要があります。

- **vob\_snapshot** バックアップから通常 VOB または UCM コンポーネント VOB を復元した場合は必ず、VOB のデータベースとそのプールの間に不整合が存在しないかどうかを確認し、不整合があったら修正します。この手順の詳細については、212 ページの「**vob\_restore** によるバックアップからの VOB の復元」と 217 ページの「**vob\_restore** を使用しないバックアップからの VOB の復元」を参照してください。

**rmelem** コマンドで VOB から誤って削除してしまったエレメントを復元するには、219 ページの「バックアップからの特定エレメントの復元」の手順に従います。

**メモ:** VOB 記憶ディレクトリのファイルシステムのレイアウトを理解していると、VOB の復元に役立ちます。詳細については、109 ページの「VOB 記憶ディレクトリ」を参照してください。

- 通常 VOB または UCM コンポーネント VOB を復元した後で、VOB とその VOB を参照するすべてのビューを再同期する必要があります。手順については、222 ページの「復元後の VOB とビューの同期」を参照してください。
- ビューを復元した後で、ビューとそのビューの参照先の VOB を再同期する必要があります。手順については、222 ページの「復元後の VOB とビューの同期」を参照してください。

- UCM と Rational ClearQuest を使用していて、VOB データベース、PVOB データベース、ClearQuest データベースを復元した場合は、これらのデータベース間の不整合を修正する必要があります。これらのデータベースの関係と、データベース間の不整合の検出と修正については、225 ページの「関連付けられたデータベースのグループに属するメンバの復元」を参照してください。

## vob\_restore によるバックアップからの VOB の復元

---

vob\_restore ユーティリティを使用すると、バックアップからの VOB の復元に必要な多くの定石的な手順を自動化することができます。このユーティリティは、vob\_snapshot バックアップツールを併用している場合、特に便利です。標準的なバックアップにも便利です (196 ページの「VOB のバックアップ方法」を参照)。

vob\_restore では、次のタスクを処理します。

- VOB の非アクティブ化とそのサーバー プロセスの停止
- 現在登録されている VOB の記憶ディレクトリまたはローカル ホストで指定した別のディレクトリへのバックアップ データのコピー
- vob\_snapshot で作成したデータベース スナップショットと、復元したデータとのマージ (必要な場合)。この処理では、データベース スナップショットの一貫性も検証され、復元した VOB に対してデータベース スナップショットが正しいものであることが確認されます。
- VOB が現在登録されている場所以外に復元される場合、レジストリに新しい VOB オブジェクトとタグを作成
- VOB の再アクティブ化
- checkvob の実行による VOB データベースとそのプール間の整合性チェック。vob\_snapshot を使用している場合は、この処理を必ず実行してください。それ以外の場合にも、実行することをお勧めします。

**メモ:** vob\_restore を使用して、NAS デバイスに保存された VOB を復元することはできません。詳細については、219 ページの「NAS での VOB の復元」を参照してください。

### vob\_restore で実行可能な作業

vob\_restore を実行すると、次の 3 つの情報を確認するためのプロンプトが表示されます。

- VOB が復元されるターゲット ディレクトリのパス名
- ターゲット ディレクトリにコピーされるバックアップ データが保存されているソース ディレクトリのパス名
- vob\_snapshot を使用してバックアップする VOB データベースが置かれているディレクトリのパス名 (必要な場合)

この情報と、**vob\_restore** コマンド行で指定した VOB タグによって、**vob\_restore** で次のどの処理が使用されるかが決まります。

- インプレース復元: バックアップを VOB の現在の記憶ディレクトリに復元します。これには 2 つの方法があります。
  - 使用するディスク容量を最小に抑える方法では、ソース ディレクトリとターゲット ディレクトリを、現在登録されている VOB の記憶ディレクトリと同じにします。アクティブである可能性がある登録済み VOB の記憶ディレクトリにはソース データをコピーできないため、**vob\_restore** は、VOB の登録を解除して、そのサーバー プロセスを停止し、ユーザーにソース データをターゲット ディレクトリにコピーするように指示します。スナップショットのバックアップのパスが指定されていれば、**vob\_restore** はデータベース スナップショットが有効であることを確認してから、ターゲットディレクトリにコピーされたデータをスナップショットとマージします。
  - VOB ホストまたはネットワークに十分なディスク容量がある場合は、一種のインプレース復元を行うのもよいでしょう。これは、ターゲット ディレクトリとソース ディレクトリを別にして、復元対象のバックアップ データを保存する一時的なステージング領域をソースのパス名で指定するというものです。この場合、**vob\_restore** は VOB の復元プロセス全体を管理します。
- 復元と移動: バックアップをローカル ホストの別の場所に復元します。この方法では、ソース ディレクトリとターゲット ディレクトリのパス名は通常は同じで、復元完了後の VOB 記憶ディレクトリのパス名になります。

どちらの方法でも、次の規則が適用されます。

- 復元対象の VOB は登録済みになっており、そのタグがホストの現在のリージョンに存在する必要があります。
- ターゲット ディレクトリは、**vob\_restore** を実行しているホストのローカルに配置されている必要があります。リモート ホストや NAS デバイスにはターゲット ディレクトリを作成できません。
- ターゲット ディレクトリにリモート プールがある場合は、それを手動で復元する必要があります。

## vob\_restore の実行

この項では、インプレース復元で VOB を復元する際に必要な処理をまとめて説明します。その 2 つの方法について説明します。

**メモ:** 多くの **vob\_restore** プロンプトで、作業を停止することができます。このようなプロンプトで作業を停止すると、**vob\_restore** によって再起動ファイルに現在の状態が保存され、標準出力でそのファイル名が表示されます。このファイル名を **vob\_restore** への **restart\_path** 引数として使用し、停止した作業を再開できます。

- 1 VOB ホストにログオンします。権限を持つユーザーとしてログインします。
- 2 ターゲット ディレクトリに十分なディスク容量があることを確認します。通常、インプレース復元では、現在登録されている VOB の記憶ディレクトリに必要なディスク容量があれば十分です。別の場所に復元する場合は、次の方法で VOB に必要な容量を確認できます。
  - VOB の admin ディレクトリが破損していない場合は、ClearCase 管理コンソールの VOB 記憶ノードを調べるか、`cleartool space` コマンドを実行します。
  - VOB の admin ディレクトリが破損しているか、現在のデータ以外のデータがある場合は、`dir`、`ls`、`du` などの通常のファイル システム ユーティリティを使用して、VOB 記憶ディレクトリが使用するディスク容量を概算できます。
- 3 すべてのビュー コンテキストを終了します。ビューで `vob_restore` を実行することはできません。
- 4 `vob_restore` を実行します。完了していない `vob_restore` を再開した場合以外は、このコマンドでは次の VOB タグを指定する単一の引数が使用されます。  
`vob_restore vob-tag`  
完了していない `vob_restore` を再開した場合は、`-restart restart_path` オプションも指定する必要があります。簡単な情報が表示され、続行するかどうかを確認するメッセージが表示されます。
- 5 ターゲット パスを入力します。これは、VOB の復元先のパスです。UNIX コンピュータでは、ホスト上のローカルパス (たとえば、`/vobstg/libpub.vbs`) を指定します。Windows コンピュータでは、UNC パス (たとえば、`¥¥ccsvr01¥vobstg¥libpub.vbs`) を指定します。
- 6 ソース パスを入力します。これは、`vob_restore` がバックアップデータを検索するパスです。インプレース復元の一時的なステージング領域を使用する方法では、そのパス名を入力します。それ以外の方法では、現在登録されている VOB の記憶ディレクトリのパス名 (手順 5 で入力したものと同一パス名) を入力しますが、`vob_restore` から指示されるまで、バックアップデータをそのパスにコピーしないでください (手順 16 を参照)。
- 7 `vob_restore` がデータベース スナップショットを検索するパスを入力します。  
`vob_snapshot` を使用して VOB のバックアップを復元している場合は、ここでスナップショットのパス名を指定します。そうでない場合は、[RETURN] を押して、次の手順に進みます。
- 8 バックアップが現在アクセス可能かどうかを指定します。バックアップデータが手順 6 で入力したパスでアクセス可能であれば、「yes」と入力します。そうでなければ、「no」と入力します。
- 9 要約情報を確認して、続行するかどうかを決定します。情報が正確で、復元の続行が可能であれば、「yes」と入力します。そうでなければ、「no」と入力します。

- 10 VOB データベースを確認します。**バックアップと、必要に応じてデータベース スナップショットがアクセス可能であり、次のいずれかの条件を満たす場合、データベースの一貫性を検証するように **vob\_restore** から指示されます。
- データベース スナップショットを復元している (手順 7 でパスを入力している)。
  - 標準のバックアップを使用しており (手順 7 でパスを入力していない)、バックアップ中にデータベースがロックされていないことが **vob\_restore** によって検出された。
- データベースでこのテストが失敗すると、**vob\_restore** は続行できません (手順 8 で「no」と入力すると、手順 16 でバックアップをロードするまで、データベースの確認は延期されます)。
- 11 リモート プールを復元します。**UNIX ホスト上にある VOB には、リモート記憶プールが存在する可能性があります。ここでリモート記憶プールを復元するように **vob\_restore** から指示されます。リモート記憶プールをバックアップからターゲットの場所にコピーします。リモート プールの復元が終了したか、リモート プールがない場合は、[RETURN] を押して続行します。
- 12 バックアップ内で、データベースがデータベース スナップショットに置き換わったことを確認します。****vob\_snapshot** でバックアップした VOB を復元している場合は、スナップショット内のデータベースでバックアップ内のデータベースを上書きするかどうかを確認するメッセージが **vob\_restore** によって表示されます。上書きする場合は、「yes」と入力します。上書きしない場合は「quit」と入力し、手順 7 で「no」と指定して、この手順を再実行します。
- 13 VOB タグを削除し、VOB の登録を解除します。****vob\_restore** で VOB タグを削除して VOB の登録を解除しないと、作業を続行できません。ほかのレジストリ リージョンに存在する可能性のあるすべてのタグを削除し、ほかのすべてのレジストリの VOB の登録を解除するように、**vob\_restore** から指示されます。この作業を完了したか、ほかのタグやレジストリがない場合は、[RETURN] を押して続行します。
- 14 ClearCase を安全にシャット ダウンできることを確認します。****vob\_restore** で VOB のサーバー プロセスを停止しないと、復元を続行できません。
- 15 既存の VOB 記憶ディレクトリを削除します。**ClearCase のシャット ダウン後、既存の VOB 記憶ディレクトリを削除するように **vob\_restore** から指示されます。削除したら、[RETURN] を押します。
- 16 ターゲット ディレクトリにバックアップ データをコピーします。**手順 8 で「no」と入力すると、ここでバックアップをロードするように **vob\_restore** から指示されます。バックアップをロードしたら、[RETURN] を押します。

**メモ:** UNIX では、各 VOB 記憶ディレクトリには **.identity** という名前のサブディレクトリが含まれています。このサブディレクトリには、特別な権限を持つファイル (**setuid** ビットが設定されている **uid** ファイル、**setgid** ビットが設定されている **gid** ファイル) が格納されています。バックアップデータをターゲットディレクトリにコピーするには、これらの権限が必要です。

- **tar** コマンドを使用する場合は、VOB の復元の際に **-p** オプションを指定します。  
**tar** コマンドは **root** 権限で実行します。そうしないと、**-p** フラグが無視されます。
- **cpio** コマンドを使用する場合は、特にオプションは必要ありません。

- 17 checkvob を実行します。** ターゲットディレクトリにバックアップデータと、必要に応じてデータベーススナップショットがコピーされると、**checkvob** を実行するように **vob\_restore** から指示されます。このユーティリティを実行すると、VOB データベースと記憶プールの間の不整合が検出され、必要に応じて修正されます。確認するプールの種類と、**checkvob** が何か問題を検出したときに問題を修正するかどうかを指定できます。データベーススナップショットで復元された VOB では、このような問題が発生する可能性が高くなります。データベースとプールのバックアップが同時には行われないからです。第 12 章と **checkvob** のリファレンス ページでは、**checkvob** について詳しく説明しています。ぜひこの説明をお読みください。特に、**-fix** モードで **checkvob** を実行する場合は、その前に参照することをお勧めします。

**注意:** **-fix** モードで **checkvob** を実行するには、**vob\_restore** で VOB をアンロックしてから、**-nusers username** を指定して再度ロックします。**username** には、**vob\_restore** を実行するユーザーの名前を入力します。VOB のレプリカが作成されている場合、このユーザーは **checkvob -fix** を使用したときにのみ、VOB を変更できるようになります。**checkvob -fix** を使用しないと、レプリカエラーが発生し、レプリカに相違ができてしまいます。

- 18 VOB のレプリカが作成されたら、MultiSite の **restorerereplica** コマンドを実行します。**  
このコマンドは、VOB がロックされている状態で実行してください。
- 19 VOB がロックされている状態で、整合性チェックを行います。** たとえば、デフォルト構成仕様のビュー内で、復元した VOB にアクセスし、可能性のあるすべてのエレメントとバージョンを確認します。
- 20 復元した VOB をアンロックします。** **vob\_restore** は、常に VOB をロックした状態のまま終了します。

これで、復元した VOB が使用できる状態になりました。ただし、場合によっては次の 2 つの処理を実行する必要があります。

- 必要に応じて、VOB とビューを再同期します。223 ページの「動的ビューを同期するには」を参照してください。
- VOB を別の場所に復元し、VOB タグがほかのリージョンにある場合は、49 ページの「タグの作成、削除、変更」の説明に従って、そのタグを置換します。

## vob\_restore を使用しないバックアップからの VOB の復元

---

多くの場合、VOB を復元するには **vob\_restore** ユーティリティを実行してその指示に従うのが最も簡単です。**vob\_restore** を使用できない場合は、次の手順を実行してください。

- 1 VOB ホストにログオンします。権限を持つユーザーとしてログインします。
- 2 ターゲット ディレクトリに十分なディスク容量があることを確認します。通常、インブレース復元では、現在登録されている VOB の記憶ディレクトリに必要なディスク容量があれば十分です。別の場所に復元する場合は、次の方法で VOB に必要な容量を確認できます。
  - VOB の **admin** ディレクトリが破損していない場合は、ClearCase 管理コンソールの VOB 記憶ノードを使用するか、**cleartool space** コマンドを実行します。
  - VOB の **admin** ディレクトリが破損しているか、現在のデータ以外のデータがある場合は、**dir**、**ls**、**du** などの通常のファイル システム ユーティリティを使用して、VOB 記憶ディレクトリが使用するディスク容量を概算できます。
- 3 ClearCase を停止します。これで、実行中の VOB のサーバー プロセスは完全に停止しました。
- 4 既存の VOB 記憶ディレクトリを削除します。**rmdir** や **rm** などのファイル システム ユーティリティか、Windows エクスプローラなどの GUI を使用します。
- 5 ClearCase を再起動します。ClearCase を起動すると、ホスト上のほかの VOB とビューが使用できるようになります。
- 6 バックアップをロードします。必要に応じて、VOB 記憶ディレクトリを再作成します。次に、バックアップから VOB 記憶ディレクトリの内容を復元します。必ずコピー ユーティリティを使用して、ファイル保護とディレクトリ保護を行ってください。

UNIX では、各 VOB 記憶ディレクトリには **.identity** という名前のサブディレクトリが含まれています。このサブディレクトリには、特別な権限を持つファイル (**setuid** ビットが設定されている **uid** ファイル、**setgid** ビットが設定されている **gid** ファイル) が格納されています。バックアップ データをターゲット ディレクトリにコピーするには、これらの権限が必要です。

- **tar** コマンドを使用する場合は、VOB の復元の際に **-p** オプションを指定します。**tar** コマンドは **root** 権限で実行します。そうしないと、**-p** フラグが無視されます。
- **cpio** コマンドを使用する場合は、特にオプションは必要ありません。

Windows では、271 ページの「VOB または ビューの記憶ディレクトリをコピーするとき NTFS ACL を保持する」の手順に従います。

**メモ:** Windows でも UNIX でも、VOB のバックアップ時に VOB がロックされていた場合 (通常の場合) は、復元時にもロックされます。バックアップ時に VOB がロックされていないと、そのバックアップは役に立たない可能性があります。

- 7 必要に応じて記憶ディレクトリ保護を修正します。Windows のバックアップ ツールで ACL を正しくバックアップおよび復元できない場合、ここで修正が必要になります。詳細については、269 ページの「NTFS の記憶ディレクトリ ACL の修復」を参照してください。
- 8 リモート プールを復元します。VOB が UNIX ホスト上にあり、そのリモート記憶プールが存在する場合は、ここで復元します。バックアップからターゲットの場所にコピーするだけで復元できます。リモート プールは、手順 10 で **checkvob** を実行した場所に存在する必要があります。
- 9 VOB を別の場所に復元した場合は、VOB を再登録します。VOB を再登録するには、ClearCase 管理コンソールで ClearCase レジストリ ノードを使用するか、**cleartool** コマンドを実行します。たとえば、VOB を新しい場所 `/vobst_aux/flex.vbs` に復元した場合は、次のコマンドを実行します。

```
# cleartool unregister -vob /vobstore/flex.vbs

# cleartool register -vob /vobst_aux/flex.vbs

# cleartool mktag -vob -replace -tag /vobs/flex /vobst_aux/flex.vbs
```
- 10 **checkvob** を実行します。このユーティリティは、VOB データベースと記憶プールの間の不整合を検出し、必要に応じてそれを修正します。データベース スナップショットで復元された VOB では、このような問題が発生する可能性が高くなります。データベースとプールのバックアップが同時には行われないからです。第 12 章と **checkvob** のリファレンス ページでは、**checkvob** について詳しく説明しています。この情報を読んでから **checkvob** を実行することを強くお勧めします。

**注意:** **checkvob** を **-fix** モードで実行する場合は、まず VOB をアンロックし、次に **-nusers** にユーザー自身を指定して再度ロックしてください。VOB のレプリカが作成されている場合、**checkvob -fix** による変更以外は一切 VOB を変更しないでください。また、**checkvob** を終了したら、すぐに VOB をすべてのユーザーが使用できるように再度ロックしてください。**checkvob -fix** を使用しないと、レプリカ エラーが発生し、レプリカに相違ができてしまいます。
- 11 VOB のレプリカが作成されたら、MultiSite の **restorereplica** コマンドを実行します。このコマンドは、VOB がロックされている状態で実行してください。
- 12 VOB がロックされている状態で、整合性チェックを行います。たとえば、デフォルト構成仕様のビュー内で、復元した VOB にアクセスし、可能性のあるすべてのエレメントとバージョンを確認します。
- 13 復元した VOB をアンロックします。

これで、復元した VOB が使用できる状態になりました。ただし、場合によっては次の 2 つの処理を実行する必要があります。

- 必要に応じて、VOB とビューを再同期します。223 ページの「動的ビューを同期するには」を参照してください。
- VOB を別の場所に復元し、VOB タグがほかのリージョンにある場合は、49 ページの「タグの作成、削除、変更」の説明に従って、そのタグを置換します。

## NAS での VOB の復元

---

VOB データベースが NAS デバイスに格納されている場合、`vob_restore` を使用して VOB を復元することはできません。VOB を NAS デバイスに復元する場合は、217 ページの「`vob_restore` を使用しないバックアップからの VOB の復元」で説明されている手順に従ってください。復元した VOB を登録してそのタグを作成するには、`cleartool register` コマンドと `mktag` コマンドに、`-host` オプション、`-hpath` オプション、`-gpath` オプションを指定する必要があります。たとえば、VOB が NAS デバイス上の `/vobst_aux` ディレクトリに復元されている場合は、次の `register` コマンドと `mktag` コマンドを使用します。

```
cleartool register -vob -host ccvobsvr1 -hpath /net/nasdevice/vobst_aux/flex.vbs ¥  
-gpath /net/nasdevice/vobst_aux/flex.vbs /net/nasdevice/vobst_aux/flex.vbs  
  
cleartool mktag -vob -replace -host ccvobsvr1 -tag /vobs/flex ¥  
-hpath /net/nasdevice/vobst_aux/flex.vbs ¥  
-gpath /net/nasdevice/vobst_aux/flex.vbs /net/nasdevice/vobst_aux/flex.vbs
```

## バックアップからの特定エレメントの復元

---

`rmelem` で誤ってエレメントを削除してしまった場合は、VOB のバックアップからそのエレメントを復元できます。この項では、その処理について説明します。

**メモ:** この処理で、エレメントを UCM コンポーネントに復元することはできません。

ディレクトリ エレメントを削除した場合でも、その中にカタログされているファイル エレメントまで削除されることはありません。ファイル エレメントとディレクトリ エレメントは独立して存在しています。通常、ディレクトリ エレメントを削除すると、その中のファイルは VOB の `lost+found` ディレクトリに移動します。誤ってディレクトリ エレメントを削除した場合は、最初にこのディレクトリを確認してください。

単一のエレメントを UCM VOB 以外のバックアップから復元するには

- 1 エレメントが削除された VOB を非アクティブにします。VOB がマウントされている場合はアンマウントしてから、タグを削除し、登録を解除します。この手順は必ず実行してください。手順 2 では、バックアップからこの VOB の一時的なコピーを復元します。一時的なコピーと元の VOB を同じレジストリに登録することはできません。

- 2 VOB の最新のバックアップを一時的に復元します。212 ページの「vob\_restore によるバックアップからの VOB の復元」または 217 ページの「vob\_restore を使用しないバックアップからの VOB の復元」の手順に従います。元の VOB が使用している記憶ディレクトリにバックアップを復元しないでください。復元した VOB は削除されたエレメントの取得にのみ使用します。これは一時的な復元なので、復元場所は VOB サーバー ホスト上のどの場所でもかまいません。
- 3 復元した VOB のタグを登録して作成します (手順 2 で説明されている vob\_restore を使用しなかった場合)。
- 4 一時的な VOB を新規作成し、復元する必要があるエレメントを保存します。元の VOB と復元したバックアップは同時にアクティブにできないため、この一時的な VOB が、復元中のエレメントをコピーできるステージング領域となります。
- 5 削除されたエレメントを、復元した VOB から取得します。cleartool relocate を使用して、エレメントをバックアップ VOB から一時的な VOB に再配置します。詳細については、relocate のリファレンス ページと 147 ページの「別の VOB へのエレメントの再配置」を参照してください。
- 6 必要なエレメントが一時的な VOB に再配置されたことを確認してから、復元した VOB を削除します。rmvob または ClearCase 管理コンソールを使用します。
- 7 元の VOB のタグを再登録して再作成します。
- 8 動的ビューを使用している場合は、VOB をマウントします。
- 9 エレメントを一時的な VOB から元の VOB に再配置します。cleartool relocate を使用します。
- 10 エレメントが元の VOB でアクセス可能であることを確認します。
- 11 一時的な VOB を削除します。rmvob または ClearCase 管理コンソールを使用します。

**メモ:** この手順でエレメントを復元すると、手順 5 においてビューで選択したディレクトリのバージョンに復元されます。ディレクトリの以前のバージョンに復元されるものではありません。

## バックアップからのビューの復元

---

以下の手順でバックアップからビューを復元します。この手順は、ClearCase ホストに保存されているビューと、認定されている NAS デバイスに保存されているビューに適用されます。

- 1 ビュー ホストにログオンします。ビュー記憶ディレクトリ (スナップショット ビューの場合はスナップショット ビュー ディレクトリ) があるホストにログオンします。

- 2 **ディスク容量を確認します。**ビューのディスク パーティションにバックアップ コピーを保存するための十分な容量があることを確認します。十分な容量がない場合は、ビュー記憶ディレクトリ (スナップショット ビューの場合はスナップショット ビュー ディレクトリ) を削除するか、ほかの方法で容量を確保します。
- 3 **ビュー サーバーを停止します。**`endview -server view-tag` を使用して、`view-tag` に指定したビューの `view_server` プロセスを停止します。
- 4 **元のビューが削除されていない場合は削除します。**`ClearCase` 管理コンソールまたは `rmview` コマンドを使用します。現在登録されている記憶ディレクトリにビューを復元し、その現在のタグを使用する計画の場合は、`rm` や `rmdir` などのオペレーティング システムのユーティリティを使用して、ビュー記憶ディレクトリ全体を削除することができます。  
**メモ:** スナップショット ビューを手動で削除し、そのビュー記憶ディレクトリがスナップショット ビュー ディレクトリのサブディレクトリでない場合 (`ClearCase LT` での通常の構成) は、スナップショット ビュー ディレクトリとビュー記憶ディレクトリの両方を削除してください。
- 5 **復元したビューの所有権が正しいことを確認します。**プラットフォームやバックアップ ツールによっては、所有者やグループの正しい識別情報でビューが復元されたことを、この時点で確認できるようにする必要があります。
  - UNIX で `tar` コマンドを使用する場合は、VOB の復元の際に `-p` オプションを指定します。`tar` コマンドは `root` 権限で実行します。そうしないと、`-p` フラグが無視されます。`cpio` コマンドを使用する場合は、特にオプションは必要ありません。
  - Windows でバックアップ ツールまたはコピー ユーティリティが NTFS ACL に対応していない場合は、復元後にその修正が必要になることがあります。詳細については、269 ページの「NTFS の記憶ディレクトリ ACL の修復」を参照してください。**メモ:** スナップショット ビューを復元する場合、バックアップの復元に使用するユーティリティは、ビュー内のすべてのファイルとディレクトリのオリジナルの変更時刻と所有権を維持管理している必要があります。そうでないと、ビューがアクティブになったときに、ロードされたすべてのファイルがハイジャックされたと報告されます。
- 6 **バックアップを復元します。**スナップショット ビューを復元する場合は、スナップショット ビュー ディレクトリとビュー記憶ディレクトリ (スナップショット ビューディレクトリのサブディレクトリではない) の両方を復元する必要があります。ビューの復元場所は、同じホストでも別のホストでも、以前の場所でも新しい場所でもかまいません。ビュー記憶ディレクトリを新しい場所に復元する場合は、ビューをその新しい場所に再登録する必要があります (手順 7)。
- 7 **必要に応じてビューのタグを登録して作成します。**現在登録されている場所にビューを復元した場合は、既存のビュー オブジェクトとビュー タグは有効なままです。それ以外の場合は、`ClearCase` 管理コンソールまたは `cleartool register` コマンドと `mktag` コマンドを使用して、ビューを登録し、そのタグを作成します。

**メモ:** NAS デバイスにビューを復元する場合は、**register** コマンドと **mktag** コマンドに、**-host** オプション、**-hpath** オプション、**-gpath** オプションを指定する必要があります。その例については、219 ページの「NAS での VOB の復元」を参照してください。

- 8 **復元後の不整合をすべてクリーン アップします。**ビューのバックアップ後に VOB で行った変更は、復元されたビュー データベースに反映されません。発生する可能性のある問題とその解決方法については、「復元後の VOB とビューの同期」を参照してください。

これで、ビューが使用できる状態になりました。

## 復元後の VOB とビューの同期

---

VOB とビューは互いに永続的参照を維持します。このような参照は、VOB 自体やビュー自体と共にバックアップされて復元されます。VOB をバックアップから復元すると、その VOB に対する参照を保持しているビュー (たとえば、VOB からチェックアウトまたはロードされたバージョン) と同期できなくなることがあります。同様に、ビューをバックアップから復元すると、そのビューの参照先である VOB と同期できなくなることがあります。この不整合が、さまざまな問題の原因となる可能性があります。問題は主にビューで発生しますが、VOB でも発生します。

- **ビューにおける問題:** ビューを復元するか、ビューによって参照されている VOB を復元すると、ビューで次のような問題が発生する可能性があります。
  - カタログされた VOB ディレクトリ エLEMENT が VOB に存在しなくなるので、ビュープライベート ファイルとビュープライベート ディレクトリが取り残される。
  - VOB にはビューがELEMENTをチェックアウトしたと記録されていても、ビューにそのELEMENTのビュープライベート ファイルが存在しないと、ELEMENTがチェックアウトされたが削除されている状態になる。
  - ビューで有効だったELEMENTが VOB で認識できないと、チェックアウト時にELEMENTがビュープライベート ファイルによって見えなくなる。
  - 存在しない非共有 DO に関する情報を VOB が記録すると、DO プロモーションが失敗する可能性がある。
  - VOB が共有 DO を保持していないと、ウィンクインした DO にアクセスできなくなる。この問題は、VOB が DO プールを失い、共有 DO が **checkvob** 修正プロセスで削除されると発生します。
- **VOB における問題:** 動的ビューが復元されても、ビューによって参照されている VOB 内の DO の参照カウントが古いと、DO はスクラップされません (復元されたビューは VOB に記録されている DO に対する参照を保持していません)。

この項では、復元後の不整合を修正する方法と、この問題の発見と回避の方法について説明します。

## 参照を持つビューを VOB 内で見つけるには

VOB が復元されると、VOB と VOB が参照する各ビューとの間で再同期が行われます。このビューのリストを収集するには、ClearCase 管理コンソールで VOB 記憶ノードの [参照されているビュー] サブノードに移動するか、次のコマンドを実行します。

`cleartool describe -long vob:restored-vob`

**メモ:** [参照されているビュー] サブノードと `cleartool describe` コマンドでは、VOB との唯一の連携が VOB からの DO のウィンクインであったビューは表示されません。そのようなビューも、見つけて再同期する必要があります。

## 動的ビューを同期するには

動的ビューまたは動的ビューの参照先 VOB を復元したら、`cleartool recoverview -synchronize` を実行して、取り残されたビュープライベート ファイルを復元します。ビューの復元後は、ビューの参照先のすべての VOB とビューを同期します。VOB の復元後は、復元した VOB に対する `recoverview` 同期を制限することができます。詳細については、`recoverview` のリファレンス ページを参照してください。

## 欠落したチェックアウトを調整するには

スナップショット ビューまたは動的ビューで欠落したチェックアウトを見つけて調整するには

- 動的ビューでは、ClearCase 管理コンソールのビュー記憶ノードの [プライベート ファイル] サブノードでチェックアウト済みファイルのリストを調べるか、`cleartool lsprivate -co` を実行します。
- スナップショット ビューでは、`cleartool ls -recurse -view_only` を実行します。
- 「チェックアウトされたが削除されている」と表示されたファイルを探します。
  - 復元された VOB のファイルの場合、バックアップ完了後にユーザーによってエレメントがチェックアウトされ、チェックアウトが解決されているはずですが。チェックアウトが動的ビューに対して行われ、チェックインで解決された場合、チェックイン バージョンで行われた変更は、それ以降の VOB のバックアップがなければ失われています。チェックアウトがスナップショット ビューまたは Web ビューに対して行われ、チェックインで解決された場合、変更はスナップショット ビュー ディレクトリに残ります。チェックアウトがキャンセルされた場合、チェックアウト ファイルに対する変更は、`.keep` サフィックスを持つビュープライベート ファイル内のビューにあります。
  - 復元されたビュー内にチェックアウトがある場合、チェックアウト以降のビュープライベート変更はすべて失われています。
  - いずれの場合も、ユーザーは `uncheckout` でチェックアウトをキャンセルするか、バージョン データを作成し直してチェックインすることによって、チェックアウトを解決する必要があります。

## 見えなくなっているビュープライベート ファイルを削除するには

スナップショット ビューまたは動的ビューで見えなくなっているビュープライベート ファイルを見つけるには

- 動的ビューでは、ClearCase 管理コンソールのビュー記憶ノードの [プライベート ファイル] サブノードでチェックアウト済みファイルのリストを調べるか、`cleartool lsprivate -other -long` を実行します。
- スナップショット ビューでは、`cleartool ls -recurse -view_only` を実行します。
- 「見えなくなっています」と表示されたエレメントを探します。
  - 復元された VOB のエレメントの場合、エレメントは VOB の最後のバックアップ後にユーザーによってチェックアウトされているはずです。
  - エレメントのチェックアウト先が復元されたビューの場合、ユーザーによってエレメントがチェックアウトされてからチェックアウトがキャンセルされたか、現存するバージョンデータが、復元されたビューにチェックインされているはずです。そのバージョンがチェックインされていた場合、見えなくなっているビュープライベート ファイルは削除され、ユーザーは新しいバージョンをチェックアウトして作業を続行している可能性があります。そのバージョンがチェックインされていなかった場合、見えなくなっているビュープライベート ファイルには最新の変更が残っている可能性があります。この場合、ファイル名を変更すれば、エレメントの新しいバージョンをチェックアウトしてこれらの変更を保持することができます。

## ウインクインされた無効な DO の削除

ClearCase 管理コンソールのビュー記憶ノードの [プライベート ファイル] サブノードで派生オブジェクト (DO) のリストを表示するか、`cleartool lsprivate -do` を実行します。各ファイルを開いてみます。ウインクインされた DO のうち、無効になったものが検出され、削除されます。

ビューのすべての `.cmake.state` ファイルを削除します。このビューで作成された DO の状態について VOB がクエリーされます。ビューで再ビルドを行うと、DO として認識されなくなったビュープライベート ファイルが検出されて再ビルドされます。

動的ビューの DO プロモーション問題を解決した後に、別の問題が発生する場合があります (たとえば、ビューの DO コンテナが存在しないなど)。この問題の再発を防止する方法については、「動的ビューの派生オブジェクトの状態における整合性の再確立」を参照してください。

## 動的ビューの派生オブジェクトの状態における整合性の再確立

VOB をバックアップから復元すると、その VOB データベースが特定の派生オブジェクト (DO) に対して古くなっている場合があります。古いデータベースには、その後のビルドで作成された DO の構成レコードは保存されません。結果として、階層ビルドでエラーが発生します。このビルドでは、上位レベルのターゲットを構築するために後で作成された DO を再使用するためです。通常は、clearmake によって、「内部エラー」というメッセージが表示され、error\_log ファイルに次の形式のエントリが書き込まれます。

派生オブジェクト (OID) "0b5759d0.fb1811cc.a0af.08:00:69:02:2e:aa" の構成レコードがありません

ビューと VOB の整合性を再確立するには

- 1 不整合の原因となっている DO を特定します。cleartool ls コマンドを実行します。そのような DO に「[構成レコードがありません]」という注釈が付けられます。

```
cleartool ls
bldr_comm.ugh@@09-Dec.18:26.287028
bldr_cr.msg.o [構成レコードがありません]
bldr_cr.o [構成レコードがありません]
bldr_cr_cache.msg.c@@24-May.20:51.42929
.
.
.
```

- 2 構成レコードがない DO を削除します。rm や del などのオペレーティング システムのコマンドを使用します。

```
rm bldr_cr.msg.o bldr_cr.o
```

## 関連付けられたデータベースのグループに属するメンバの復元

206 ページの「関連付けられたデータベースの一括バックアップ」で説明されているように、VOB 間には VOB のバックアップ時と特に復元時に考慮すべき関係が存在します。UCM 環境ではこのような関係はきわめて複雑です。多数の VOB が各種の VOB 間ハイパーリンクで接続されており、そこには 1 つまたは複数の PVOB 内のオブジェクトに対する参照を保持する ClearQuest データベースが存在する可能性もあるからです。1 つまたは複数のデータベースがバックアップから復元されると、多くの場合、データベース間に不整合が発生します。

この項では、関連付けられたデータベースのグループのメンバがバックアップから復元される際に不整合が発生させるような状況について説明します。このような状況のうち、次の 3 つは UCM 環境のみに当てはまります。

- コンポーネント VOB がバックアップから復元された。
- ClearQuest データベースがバックアップから復元された。
- PVOB がバックアップから復元された。

この項では、管理 VOB 階層はあるが UCM は使用しない環境での全般的な考慮事項についても説明します。

**メモ:** この項で説明する手順の多くで、`cleartool checkvob -ucm` コマンドを使用します。詳細については、`checkvob` のリファレンス ページを参照してください。

## コンポーネント VOB の復元

バックアップからプロジェクトのコンポーネント VOB の 1 つを復元する場合、次のような不整合が発生することがあります。

- PVOB で定義した変更セットとベースラインが、コンポーネント VOB のバックアップ後に作成されたためにコンポーネントに存在しないバージョンと場合によってはエレメントを参照する。
- VOB に複数のコンポーネントがあり、最新のバックアップ後に作成されたコンポーネントが PVOB で参照されているが、それらのコンポーネントがコンポーネント VOB に存在していない。

このような不整合を分析し、必要に応じて修正するには、`cleartool checkvob -ucm` を実行します。

- 1 この章で説明されているいずれかの手順で、コンポーネント VOB を復元します。
- 2 作業ビューを確立します。この手順は実行することをお勧めしますが、必須ではありません。ビューで `checkvob -ucm` を実行する場合は、コマンド行で PVOB を指定する必要はありません。`checkvob` によって、問題のあるエレメントの名前と、ビューで選択されたバージョンを持つエレメントの名前が報告されます (問題はあってもビューで選択されていないバージョンについては、ID 番号が報告されます)。
- 3 ビューで PVOB のルート ディレクトリに移動します。
- 4 `checkvob -ucm` を実行して、PVOB と復元したコンポーネント間の不整合を検出します。次のコマンドを実行すると、`lib` という名前のコンポーネントがチェックされます。

```
cleartool checkvob -ucm -component lib
```

**メモ:** コンポーネント VOB に複数のコンポーネントが含まれている場合は、`checkvob -ucm -component` を実行すると、VOB 内のすべてのコンポーネントがチェックされます。

## ClearQuest データベースの復元

バックアップからプロジェクトの ClearQuest データベースを復元する場合、次のような不整合が発生することがあります。

- PVOB で定義したアクティビティが、ClearQuest データベースに存在しないフィールドかレコード、またはその両方を参照している。

- **ClearQuest** データベースのバックアップ後に、アクティビティとストリームの名前かタイトル、またはその両方が変更されたために、**PVOB** と **ClearQuest** データベースに相違が発生する。

このような不整合を分析し、必要に応じて修正するには、**cleartool checkvob -ucm** に **-crm\_dbname** オプションを指定して実行します。

- 1 **ClearQuest** データベースを復元します。データベース製造元が指定する手順に従います。復元したデータベースが動作することと、ネットワーク経由でアクセスできることを確認します。
- 2 **checkvob -ucm** を実行して、**PVOB** と復元した **ClearQuest** データベースの間の不整合を検出します。次のコマンド行は、**CQ-UCM-DB** という名前の **ClearQuest** データベース内にある **Release-6** プロジェクトに対するすべてのエントリをチェックします。

```
cleartool checkvob -ucm -crm_dbname CQ-UCM-DB -project Release-6
```

## PVOB の復元

バックアップから **PVOB** を復元する場合、次のような不整合が発生することがあります。

- コンポーネント **VOB** で使用されているバージョン、ベースライン、ラベル タイプが **PVOB** に存在しない。
- **ClearQuest** データベースのアクティビティが、**PVOB** に存在しないオブジェクトを参照している。

このような不整合を分析し、必要に応じて修正するには、まず、**cleartool checkvob -ucm** を実行して、各プロジェクトのコンポーネント **VOB** が保持している **PVOB** への参照をチェックします。次に、**checkvob -ucm** コマンドに **-crmonly** オプションを指定して実行し、**PVOB** が保持している **ClearQuest** データベースへの参照をチェックします。

- 1 この章で説明されているいずれかの手順で、**PVOB** を復元します。
- 2 **checkvob -ucm** を実行して、各コンポーネント **VOB** と復元した **PVOB** の間の不整合を検出します。次のコマンド行は、**¥lib** というタグを持つコンポーネント **VOB** とその **PVOB** 間の整合性をチェックします。

```
cleartool checkvob -ucm vob:¥lib
```

**checkvob** は、コンポーネント **VOB** に保存されている情報を使用して **PVOB** を検索します。プロジェクトのすべてのコンポーネント **VOB** に対して、このようなコマンドを実行する必要があります。

- 3 **checkvob -ucm -crmonly** を実行して、復元した **PVOB** と **ClearQuest** データベースの間の不整合を検出します。次のコマンドは、**¥projects** という **VOB** タグを持つ **PVOB** が保持するすべての **ClearQuest** データベース参照をチェックします。

```
cleartool checkvob -ucm -crmonly vob:¥projects
```

## 管理 VOB 階層のメンバの復元

**メモ:** UCM コンポーネントまたは PVOB の復元中にこの項の処理を実行する場合は、まず `checkvob -ucm` を実行してください。先にこの項の処理を実行すると、破損した VOB 間ハイパーリンクのうち `checkvob -ucm` で修復できるものも削除されてしまいます。

管理 VOB 階層のメンバを常にグループとしてバックアップすることをお勧めします。ただし、通常は一度に復元するのはグループの 2 ～ 3 のメンバ (たとえば、ディスク容量の足りないサーバーに保存されているメンバなど) です。管理 VOB 階層の一部のメンバを復元する場合 (すべてのメンバを復元するわけではない場合) は、グローバル タイプ情報に不一致が発生する可能性があります。たとえば、新しいグローバル タイプを作成する前に管理 VOB をバックアップした場合、バックアップの復元後に、使用できないグローバル タイプに関連する問題が下位階層の VOB に発生することがあります。このような問題は、次のいずれかの方法で対処します。

- バックアップから管理 VOB を復元した場合は、その VOB と階層のその他のすべてのメンバ間に壊れたハイパーリンクがないかどうかを確認し、あれば修復します。
- バックアップから管理 VOB 階層のメンバを復元した場合は、VOB からその管理 VOB へのハイパーリンクで壊れたものがないかどうかを確認し、あればクリーンアップします。

壊れたハイパーリンクを削除するには、`checkvob -hlink` を使用します。詳細については、254 ページの「`checkvob` を使用した破損ハイパーリンクの検出と修正」と `checkvob` のリファレンス ページを参照してください。

記憶領域の管理とリポジトリの整合性の維持に対して定期的に注意を向けることは、重要な管理機能です。この章では、VOB 記憶領域とビュー記憶領域の管理タスク、**checkvob** ユーティリティ、**ClearCase** スケジューラについて説明します。**ClearCase** スケジューラは、定期的なメンテナンス タスクの自動化に使用できます。

## ClearCase スケジューラ

---

**ClearCase** スケジューラを使用すると、ローカル VOB とローカル ビューをサポートするように設定されたホストの定期的なメンテナンスを自動化することができます。スケジューラの動作は、ジョブとタスクで定義されます。

- ジョブは、タスクに実行スケジュールを定義し、最後に実行されたタスクに関連付けられた終了状態を格納するデータ構造です。各ジョブは単一のタスクを実行します。
- タスクは、実行可能なプログラムまたはスクリプトです。タスクは複数のジョブによって実行してもかまいません。タスクが実行する操作は、そのタスクが実行されるスケジュールに合わせてカスタマイズできます。なぜなら、ジョブは、ジョブ固有の引数をタスクに渡すことができるからです。

**ClearCase** が稼働しているオペレーティング システムに関連付けられている汎用のスケジュール実行ユーティリティ (**cron**、**at**、...) とは異なり、**ClearCase** スケジューラは **ClearCase** 管理インフラストラクチャと統合され、次の機能をサポートします。

- **cleartool** または **ClearCase** 管理コンソールでの集中管理
- ユーザー識別情報とグループ識別情報に基づくアクセス制御
- 時刻、周期、別のタスクの実行に基づく柔軟なスケジュール
- すべてのホスト上で実行するデフォルトのタスクとジョブ
- カスタム タスクとカスタム ジョブ

スケジューラは **albd\_server** によって管理されます。**schedule** プログラムは、実行する作業をスケジューラに渡すジョブ/タスク レジストリへのインターフェイスです。デフォルトのタスクとジョブは、多くのホスト システムに適しており、変更せずに済む場合があります。ジョブまたはタスクを変更するには、次のうちの必要な処理を実行します。

- 必要に応じて、スケジューラ アクセス コントロール リスト (ACL) を変更します。
- デフォルトのローカルの日次 (または週次) タスクにジョブを追加します。
- 実行する新しいタスクとジョブを定義します。

## スケジューラ アクセス コントロール リストの管理

**schedule** は、単一のアクセス コントロール リスト (ACL) を使用して、スケジューラのタスク / ジョブ レジストリ、スケジューラ ACL 自体にアクセスを許されるユーザーを判定するために使用しています。

ACL は項目のリストで構成されています。各項目では、識別情報にアクセスのタイプを割り当てます。識別情報は 4 種類あります。具体的には、**Everyone**、**Domain**、**Group**、**User** です。**Domain** は、**Windows** を実行するホストの場合は **Windows** ドメインであり、**UNIX** を実行するホストの場合は **NIS** ドメインです。各グループ、ユーザーはドメイン名によって修飾されます。**Windows** のドメインでは、グループはグローバル グループでなければならず、ユーザーはドメイン アカウントでなければなりません。

**メモ:** **NIS** ドメインの一部ではない **UNIX** ホストは、ACL 項目のドメイン名の代わりに文字列 **<不明>** を使用できます。

アクセスのタイプは 3 種類あります。各識別情報には、それらのいずれか 1 つを割り当てます。表 8 に、アクセスのタイプと、そのアクセスのタイプがスケジュールへのアクセス、ACL 自体へのアクセスに持つ意味を示します。

**表 8      スケジューラ ACL の項目におけるアクセスのタイプ**

アクセスのタイプ	スケジュールへのアクセス	ACL へのアクセス
読み取り	読み取り専用	読み取り専用
変更	読み取りと書き込み、ジョブの起動	読み取り専用
フル	読み取りと書き込み、ジョブの起動	読み取りと書き込み

識別情報ごとにアクセスのタイプが 1 つのみ付与されますが、それぞれのアクセスのタイプは **Everyone** から **Domain**、**Group**、**User** へと順に継承されます。つまり、各ユーザーには最も制限の緩いアクセスのタイプが適用されます。たとえば、あるユーザーの ACL 項目にアクセスのタイプ**読み取り**が指定されていても、このユーザーが属しているグループの ACL 項目にアクセスのタイプ**変更**が指定されている場合、ユーザーはアクセスのタイプ**変更**を持つことになります。

デフォルトの設定では、**Everyone** には**読み取り**アクセス権が割り当てられています。ローカルでログオンした場合、権限を持つユーザーには常にアクセスのタイプ **フル**が付与されます。リモート ホスト上では、権限を持つユーザーを含むすべてのユーザーに対するアクセス権は、スケジューラ ACL によって決定されます。したがって、デフォルトの ACL を変更するには、スケジューラが稼働しているホストにログオンすると共に、権限を持つユーザーである必要があります。

スケジューラ ACL を表示または編集するには、ClearCase 管理コンソールを実行し、次の手順に従います。

- 1 スケジューラ ACL を表示または編集するホストの [スケジュール済みジョブ] ノードに移動します。
- 2 [操作] メニューの [すべてのタスク] をクリックし、[権限の編集] をクリックします。  
表示されるダイアログ ボックスで、スケジューラの ACL を表示または編集します。

または、次のコマンドを実行してスケジューラの ACL を表示します。

```
cleartool schedule -get -acl
```

次のコマンドを実行してスケジューラの ACL を編集します。

```
cleartool schedule -edit -acl
```

このコマンドを実行すると、テキスト エディタが起動し、ACL 定義構文 (schedule のリファレンス ページを参照) を使用した現在の ACL の内容を格納しているファイルが開きます。

スケジューラの ACL 定義構文の ACL 項目を含むテキスト ファイルを作成し、次のコマンドを使用して、ACL 全体をそのファイル (次の例では `acldef.txt` という名前のファイル) 内の ACL で置き換えることもできます。

```
cleartool schedule -set -acl acldef.txt
```

## タスクの作成

タスクには次の 2 つのコンポーネントがあります。

- タスクが実行されるときに実行されるジョブ (実行可能なプログラムやスクリプト)
- スケジューラのタスク レジストリ内にある、タスクの定義

一連の標準タスクと、それらのタスクを実行するスケジュールは、ローカル VOB とローカルビューをサポートするすべてのホストにインストールされます。標準タスクの作成、変更、削除を行うことはできません。ただし、新しいタスクを定義することは可能です。また、あらかじめ定義されている 2 つの事前定義されたタスクをカスタマイズすることもできます。デフォルトのスケジュールでは、それらの一方は、毎日実行され、他方は毎週実行されます。それらのタスクに独自の処理を付け加えたり、スケジュールを変更することができます。

タスク レジストリ内のすべてのタスク定義を表示するには、次のコマンドを使用します。

```
cleartool schedule -get -tasks
```

新しいタスクを作成するには

- 1 スケジューラの実行環境 (schedule のリファレンス ページを参照) で動作できる実行可能プログラムを作成します。作成したプログラムは、スケジューラの `tasks` ディレクトリに配置することをお勧めします。

- 2 そのプログラムを毎日実行する場合は、既存のタスク `ccase_local_day` から起動します。  
そのプログラムを毎週実行する場合は、既存のタスク `ccase_local_wk` から起動します。  
どちらのタスクもスケジューラの `tasks` ディレクトリにあり、スクリプトのサフィックスは、UNIX では `.sh`、Windows では `.bat` です。カスタマイズ可能なそれらのタスクのどちらかにプログラムを追加すれば、それ以上の作業は必要はありません。
- 3 そのプログラムを新しいタスクとして実行する場合は、そのタスクをスケジューラの `tasks` ディレクトリの `task_registry` ファイルに追加します。このファイルにタスクを追加するには、テキスト エディタを使用します。  
  
タスクはジョブ定義構文に従って定義します。詳細については、`schedule` のリファレンス ページを参照してください。タスク定義には、次の必須コンポーネントを記述します。
  - 固有の数値 ID (ジョブからタスクへの参照に使用)
  - 固有のタスク名
  - タスクに関連付けた実行可能プログラムへのパス名

**注意:** 新しいタスク定義はタスク レジストリ ファイルの末尾に追加します。タスク レジストリ ファイルに定義された標準タスクの変更や削除は行わないでください。
- 4 タスク レジストリにタスクを追加したら、そのタスクを実行する新しいジョブを作成する必要があります。233 ページの「ジョブの作成」を参照してください。`ccase_local_day` などの、既存のスケジュール済みジョブにプログラムを追加した場合、新しいジョブを作成する必要はありません。

## タスクの編集

スケジューラのタスク レジストリで既存のタスク定義の編集が必要になる場合もあります。たとえば、タスクの実行可能プログラムを別のディレクトリに移動した場合、タスク レジストリ内のタスク定義のパス名を変更する必要があります。

**注意:** 編集は、ユーザーが追加したタスクのみにしてください。タスク レジストリに定義された標準タスクの変更や削除は行わないでください。スケジューラは、タスクを使用するスケジュール済みジョブを実行するときに、タスク定義の数値 ID を使用してそのタスクを参照します。したがって、タスクの数値 ID を変更すると、スケジュール済みジョブでそのタスクを参照しているすべての箇所を変更する必要があります。233 ページの「ジョブの作成」を参照してください。

タスク定義を変更するには、テキスト エディタを使用して、スケジューラの `tasks` ディレクトリの `task_registry` ファイルを編集します。タスクを編集するときは、タスク定義構文に従います。詳細については、`schedule` のリファレンス ページを参照してください。

## タスクの削除

タスク定義を削除する前に、すべてのスケジュール済みジョブで、そのタスクを参照しているすべての箇所を削除する必要があります。233 ページの「ジョブの作成」を参照してください。タスク定義を削除するには、テキストエディタを使用して、スケジューラの **tasks** ディレクトリの **task\_registry** ファイルを編集します。

**注意:** 削除は、ユーザーが追加したタスクのみにしてください。タスク レジストリに定義された標準タスクの変更や削除は行わないでください。

## ジョブの管理

スケジューラによって実行されるジョブの作成、削除、編集を行うことができます。また、スケジュール済みジョブを直ちに実行することもできます。Windows または UNIX ホスト上でスケジュール済みジョブを管理するには、ClearCase 管理コンソールを使用します。[ホスト] ノードと各ホストのスナップインには、ホスト上のスケジュール済みジョブを管理できる [スケジュール済みジョブ] ノードがあります。スケジューラ ACL が書き込みアクセス (230 ページの「スケジューラ アクセス コントロール リストの管理」を参照) をサポートしている場合は、ジョブ スケジューラが起動されているホスト上で **cleartool schedule** コマンドを使用することもできます。

スケジュール済みジョブを作成、編集、削除、実行するには、スケジューラ ACL 内で**変更** または**フル**のアクセス権を持つ必要があります。スケジュール済みジョブを表示するには、スケジューラ ACL 内で**読み取り**のアクセス権を持つ必要があります。

## ジョブの作成

ジョブを作成するときは、次の情報を指定します。

- ジョブの名前と、必要に応じて説明。
- ジョブで実行するタスク。タスクはタスク レジストリ内に定義されている必要があります。詳細については、231 ページの「タスクの作成」を参照してください。
- タスクに関連付けられた実行可能ファイルに、実行時に渡す引数。VOB かビューを処理の対象とする標準タスクに関しては、その対象として特定の VOB かビュー、またはローカルホスト上のすべての VOB かビューを指定することができます。
- ジョブを実行するスケジュール。
- 電子メールによる通知を実行するジョブ関連のイベントと通知の受信者。235 ページの「ジョブ通知の指定」を参照してください。
- 実行後にジョブを削除する必要があるかどうか。

新しいジョブを作成するには、ClearCase 管理コンソールを使用し、次の手順に従います。

- 1 新しいジョブを実行するホストの [スケジュール済みジョブ] ノードに移動します。
- 2 [操作] メニューの [新規作成] をクリックし、[ジョブ] をクリックします。新しいジョブの定義に必要な情報を指定するダイアログ ボックスが開きます。

次のコマンドを使用して削除することもできます。

`cleartool schedule -edit -schedule`

このコマンドを実行すると、テキスト エディタが起動し、現在スケジュールされているすべてのジョブの定義を格納しているファイルが開きます。新しいジョブを作成するには、ジョブ定義構文に従って、定義を追加します。詳細については、**schedule** のリファレンス ページを参照してください。読み取り専用のジョブ プロパティ (たとえば、**LastCompletionInfo**) を指定することはできません。ジョブは、**schedule** のリファレンス ページで説明されている環境で動作します。

## ジョブのスケジュールの指定

ジョブに設定できるスケジュールには次の 2 種類があります。

- **順次スケジュール**: あるジョブが終了すると、直ちに次のジョブが実行されます。
- **周期スケジュール**: 指定日の指定時刻にジョブが実行されます。

順次スケジュールのジョブを指定するには、現在のジョブの実行が完了したら、次にどのスケジュール済みジョブを実行するかを指定します。周期スケジュールのジョブを指定するには、ジョブを実行する間隔を指定します。周期スケジュールには、次の 4 とおりの間隔があります。

- **1 回限り**: ジョブを 1 回だけ実行する日付と時刻を指定します。開始日と終了日に同じ値を指定すると、ジョブは 1 回だけ実行されます。ClearCase 管理コンソールで、[スケジュール済みジョブ] ノードを使用してスケジュール済みジョブを 1 回だけ直ちに実行することもできます。
- **毎日または n 日ごとに実行**: ジョブを実行する頻度と開始時刻を指定します。
- **毎週または n 週ごとに実行**: ジョブを実行する頻度と曜日、さらに開始時刻を指定します。
- **毎月または n か月ごとに実行**: ジョブを実行する頻度と日付、さらに開始時刻を指定します。

日次、週次、月次のスケジュールに関して、ジョブの開始日と終了日を指定できると共に、1 日のうちに繰り返し実行させる間隔を指定することもできます。

ジョブのスケジュールを指定するには、ClearCase 管理コンソールを使用し、次の手順に従います。

- ジョブのスケジュールを指定するホストの [スケジュール済みジョブ] ノードに移動します。
  - 新しいジョブのスケジュールを指定するには、[操作] メニューの [新規作成] をクリックし、[ジョブ] をクリックします。新しいジョブの定義に必要な情報を指定するダイアログ ボックスが開きます。タスクを指定したら、[スケジュール] タブをクリックしてスケジュールを指定します。
  - 既存のジョブのスケジュールを指定するには、ジョブを選択し、[操作] メニューの [プロパティ] をクリックします。ダイアログ ボックスで、[スケジュール] タブをクリックしてスケジュールを指定します。

または、次のコマンドを実行します。

```
cleartool schedule -edit -schedule
```

このコマンドを実行すると、テキスト エディタが起動し、現在スケジュールされているすべてのジョブの定義を格納しているファイルが開きます。新しいジョブまたは既存のジョブのスケジュールを指定するには、ジョブ定義構文に従って、ジョブの **Schedule** プロパティを編集します。ジョブ定義構文の詳細については、**schedule** のリファレンス ページを参照してください。

## ジョブ通知の指定

指定した受信者に、スケジューラから電子メールによる通知を送信するようにできます。また、ジョブの開始やジョブの異常終了などの、通知を実行するイベントを指定することもできます。

**メモ:** ジョブを通知するには、スケジューラと SMTP メール サーバーとのやり取りが必要です。Windows の場合、スケジューラを実行する各ホストの [コントロール パネル] で、[ClearCase] プログラムの [オプション] タブをクリックし、このサーバーの名前を指定します。UNIX の場合、スケジューラは /bin/mail プログラムを使用して通知を送信します。

ジョブの通知情報を指定するには、ClearCase 管理コンソールを使用し、次の手順に従います。

- 1 ジョブの通知情報を指定するホストの [スケジュール済みジョブ] ノードに移動します。
- 2 新しいジョブの通知情報を指定するには、[操作] メニューの [新規作成] をクリックし、[ジョブ] をクリックします。ダイアログ ボックスに、新しいジョブの定義に必要な情報を指定します。タスクを指定したら、[設定] タブをクリックして、通知するイベントと受信者を指定します。
- 3 既存のジョブの通知情報を指定するには、詳細ペインでジョブを選択し、[操作] メニューの [プロパティ] をクリックします。表示されるダイアログ ボックスで [設定] タブをクリックして、通知するイベントと受信者を指定します。

または、次のコマンドを実行します。

```
cleartool schedule -edit -schedule
```

このコマンドを実行すると、テキスト エディタが起動し、現在スケジュールされているすべてのジョブの定義を格納しているファイルが開きます。新しいジョブまたは既存のジョブの通知情報を指定するには、ジョブ定義構文に従って、ジョブの **NotifyInfo** プロパティを編集します。詳細については、**schedule** のリファレンス ページを参照してください。

## ジョブのプロパティの表示

スケジュール済みジョブのプロパティを表示するには、**ClearCase** 管理コンソールを使用し、次の手順に従います。

- 1 ジョブのプロパティを表示するホストの [スケジュール済みジョブ] ノードに移動します。
- 2 詳細ペインでジョブを選択し、[操作] メニューの [プロパティ] をクリックします。  
ダイアログ ボックスにジョブのプロパティが表示されます。
- 3 ジョブを前回実行した時刻や状態などのメッセージと情報を表示するには、詳細ペインでジョブを選択し、[操作] メニューの [完了情報の詳細を表示] をクリックします。

または、次のコマンドを実行します。

```
cleartool schedule -get -schedule
```

特定のジョブの定義を表示するには、次のコマンドを実行します。

```
cleartool schedule -get -job job-id-or-name
```

ジョブの名前は、大文字と小文字が区別され、名前に空白が含まれる場合は引用符で囲む必要があります。例を次に示します。

```
cleartool schedule -get -job "Weekly Log Scrubbing"
```

```
Job.Begin
    Job.Id: 7
    Job.Name: "Weekly Log Scrubbing"
    Job.Description.Begin:
.
.
.
Job.End
```

ジョブを前回実行した時刻や状態などのメッセージと情報を表示するには、次のコマンドを実行します。

```
cleartool schedule -status job-id-or-name
```

ジョブのプロパティがジョブ定義構文に従って表示されます。ジョブ定義構文の詳細については、**schedule** のリファレンス ページを参照してください。

## ジョブのプロパティの編集

既存のジョブを編集するには、ClearCase 管理コンソールを使用し、次の手順に従います。

- 1 ジョブを編集するホストの [スケジュール済みジョブ] ノードに移動します。
- 2 詳細ペインでジョブを選択し、[操作] メニューの [プロパティ] をクリックします。表示されるダイアログ ボックスで、ジョブのプロパティを編集することができます。ただし、読み取り専用のジョブ プロパティを編集することはできません。

または、次のコマンドを実行します。

```
cleartool schedule -edit -schedule
```

このコマンドを実行すると、テキスト エディタが起動し、現在スケジュールされているすべてのジョブの定義を格納しているファイルが開きます。ジョブ定義構文に従って、ジョブのプロパティを編集します。ジョブ定義構文の詳細については、**schedule** のリファレンス ページを参照してください。読み取り専用のジョブ プロパティ (たとえば、**LastCompletionInfo**) を編集することはできません。

スケジューラのジョブ定義構文に従ったジョブ定義のテキスト ファイルがある場合、以下のコマンドを実行して、スケジュール全体をそのファイル内のジョブ定義で置き換えることができます。pname はジョブ定義のファイルを表します。

```
cleartool schedule -set -schedule pname
```

## ジョブの即座の実行

スケジュール済みジョブを直ちに実行するには、ClearCase 管理コンソールを使用し、次の手順に従います。

- 1 ジョブを実行するホストの [スケジュール済みジョブ] ノードに移動します。
- 2 詳細ペインでジョブを選択し、[操作] メニューの [今すぐ実行] をクリックします。

または、次のコマンドを実行します。

```
cleartool schedule -run job-id-or-name
```

ジョブはスケジューラの実行環境で動作します。詳細については、**schedule** のリファレンス ページを参照してください。

## ジョブの削除

スケジュール済みジョブを削除するには、ClearCase 管理コンソールを使用し、次の手順に従います。

- 1 ジョブを削除するホストの [スケジュール済みジョブ] ノードに移動します。
- 2 詳細ペインでジョブを選択し、[操作] メニューの [ジョブの削除] をクリックします。

または、次のコマンドを実行します。

```
cleartool schedule -delete job-id-or-name
```

## VOB 記憶領域の増加を制御するスクラブ

VOB 記憶領域は、すべての記憶領域と同様、無制限の増加を防止するように管理される必要があります。**scrubber** と **vob\_scrubber** の 2 つのユーティリティは、VOB 記憶領域の増加の管理を支援します。表 9 で、このユーティリティの機能について説明します。

表 9 **scrubber** と **vob\_scrubber** のデフォルトのアクション

プログラム	削除する対象 (デフォルト)	その作成元
scrubber	最後に参照されてから 96 時間より長く経過している <b>cleartext</b> コンテナ	VOB 内の成果物のバージョンにアクセスするすべてのコマンド
	最後に参照されてから 96 時間より長く経過している派生オブジェクト	<b>clearmake</b> 、 <b>omake</b> 、 <b>clearaudit</b> (動的ビューのみ)
vob_scrubber	特定のイベント レコード (詳細については、 <b>vob_scrubber</b> のリファレンス ページを参照)	エレメント、バージョン、その他のメタデータを作成、変更するすべてのコマンド
	特定の <b>oplog</b> 項目 (詳細については、 <b>vob_scrubber</b> のリファレンス ページを参照)	レプリカ VOB におけるデータまたはメタデータのすべての変更

どちらのユーティリティも、デフォルト スケジュールに従い、デフォルト パラメータを使用して、1 つ以上の VOB をホストするすべてのコンピュータ上で実行されます。デフォルトのスクラブ パラメータとスケジュールが適切でない場合は、この項で説明するように、スクラブ パラメータとスケジュールの再構成が必要です。

### VOB 記憶プールのスクラブ

**scrubber** の動作を規制する、基本的な規則があります。

- **cleartext** プールは、記憶プールのサイズを制御するためにスクラブされます。**cleartext** プールは、本質的にはキャッシュです。**cleartext** プールの、必要のないデータ コンテナのスクラブによるパフォーマンスへの影響は、少ないか、まったくありません。
- 派生オブジェクト プールは、すべてのビューで使用されなくなった **DO** データ コンテナを削除するためにスクラブされます。スクラブにより、VOB データベースの対応する派生オブジェクトも削除されます。
- ソース プールがスクラブされることはありません。

デフォルト設定では、**scrubber** は **Daily VOB Pool Scrubbing** タスクの一部分として夜間の実行されます。このタスクは、手動で実行することもできます。

派生オブジェクト プールと **cleartext** 記憶プールにはそれぞれ専用のスクラブ パラメータがあり、**scrubber** がプールを処理する方法はこれらのパラメータによって制御されます。**VOB** 記憶プールがスクラブされる方法を変更するには、**ClearCase** 管理コンソールまたは **mkpool-update** コマンドを使用して、個々のプールに対するスクラブ パラメータを変更します。詳細については、239 ページの「デフォルトのスクラブ パラメータの調整」を参照してください。

## VOB データベースのスクラブ

バージョン付き成果物へのほとんどすべての変更は、**VOB** データベースにイベント レコードとして記録されます。エレメントやバージョンの作成など、一部のイベント レコードにはパーマネント値が設定されます。ほかのイベント レコードは、組織にとって有用ではなかったり、時間の経過と共に値が失われたりします (たとえば、必要でなかったり、古くなったりしたバージョンラベルの削除には関心がない場合があります)。

**VOB** ホストごとに **scrubber** 設定ファイル (**ccase-home-dir¥config¥vob¥vob\_scrubber\_params**) があります。この設定ファイルを使用して、そのホストにあるすべての **VOB** に対する **vob\_scrubber** 操作を制御します。スクラブ スケジュールに対してより細かい制御が必要な場合は、各 **VOB** に対して **scrubber** パラメータ ファイルを作成できます。このファイルも **vob\_scrubber\_params** という名前が付けられ、**VOB** 記憶ディレクトリに保存されます。詳細については、**vob\_scrubber** のリファレンス ページを参照してください。

**メモ:** **vob\_scrubber** またはほかの方法 (**rmver**、**rmelem**、**relocate** など) を使用してイベント レコードとその他のメタデータを **VOB** データベースから削除すると、データベース内の空き容量は増えますが、**VOB** データベースが使用するディスクの使用量は減りません。**VOB** データベースを定期的にスクラブすると、データベース サイズの増加スピードは抑えられ、拡大を制御するための介入も行う必要がなくなります。ただし、**VOB** データベースのサイズを強制的に減らすことが必要になった場合は、データベースをスクラブし、**reformatvob** コマンドを実行してください。

## デフォルトのスクラブ パラメータの調整

一般に、よくある次の 2 つの問題に対応して **VOB** ホストのデフォルトのスクラブ パラメータを調整します。

- **十分な空き容量がない。****VOB** 記憶プールがあるディスク パーティションが頻繁にディスク フルの状態になっている可能性があります。より積極的なスクラブ計画が必要です。
- **十分な時間がない。****scrubber** ユーティリティの実行終了までには多くの時間を要するため、夜間ビルドなどの、ほかの夜間アクティビティに影響を与えている可能性があります。より控えめなスクラブ計画が必要です。

試験実行が必要な場合があります。たとえば、スクラブを調整して実行回数を減らすと、以前に発生しなかったディスクの空き容量の問題が発生する可能性があります。**VOB** ホスト上でスクラブの調整を行う前に、**scrubber\_log** ファイルを分析してください。このファイルの詳細については、**scrubber** のリファレンス ページを参照してください。

**メモ:** VOB 記憶ディレクトリのタイムスタンプを変更するバックアップツールを使用すると、VOB 記憶ディレクトリ内の DO プールと `cleartext` プールがスクラブされないことがあります。デフォルトでは、`scrubber` コマンドは過去 96 時間に参照されなかったオブジェクトをスクラブします。そのようなバックアップツールが毎晩実行されると、プール内のオブジェクトのアクセス時刻は毎晩リセットされ、オブジェクトはスクラブされることがありません。

以降の項で、スクラブ パラメータの調整について、簡単な例を示します。

## 派生オブジェクトのより頻繁なスクラブ

デフォルトでは、`scrubber` を使用すると、参照不能な派生オブジェクトのデータ コンテナを、4 日間 (96 時間)、記憶プールに保存できます。DO によって VOB のディスク パーティションがディスク フル状態にされる場合、この猶予期間を短くすることができます。次のコマンドを実行すると、不要なデータ コンテナの VOB のデフォルト DO 記憶プール (`ddft`) が 24 時間ごとに空にされます。

```
cleartool mkpool -update -age 24 ddft@vob-tag
プール "ddft" を更新しました。
```

## 派生オブジェクトのスクラブの微調整

猶予期間の調整が、ディスク パーティションのディスク フルを防止するほど十分ではないとします。`scrubber` ユーティリティの実行回数を増やし、夜間だけでなく勤務時間にも実行することにした。スクラブの対象を、特定のディレクトリに作成した DO に絞り込みます。これは、勤務時間中にユーザーに与える影響を最小限にするためです。この例では、DO プールを移動し、スクラブ回数を増やすための、UNIX のコマンド行の構文を示します。

- 1 ディレクトリの現在の DO 記憶プール割り当てを特定します。この記憶プールをクリーンアップする必要があります。

```
cd /vobs/sources
```

```
cleartool describe -long reorg@@@
directory element "reorg@@@":
.
.
... derived pool: ddft
```

このディレクトリはデフォルトの DO プールを使用します。

- 2 このディレクトリを別個の記憶プールに割り当てます。この割り当てにより、スクラブの微調整が可能になります。これは、スクラブはプールごとに起動できるためです。

```
cd /vobs/sources
```

```
cleartool mkpool -derived new_do_pool
Comments for "new_do_pool":
pool for DOs created in /vobs/sources/reorg
.
Created pool "new_do_pool".
cleartool chpool new_do_pool reorg
Changed pool for "reorg" to "new_do_pool".
```

- 3 VOB 記憶ディレクトリの場所を特定します。lsvob コマンドを次のように使用します。

```
cleartool lsvob /vobs/sources
* /vobs/sources /net/mars/vobstg/sources.vbs
```

- 4 新しい記憶プールのスクラブを、十分かつ頻繁に実行します。これを行う方法は多数あります。スケジューラに、新しいプールに対して scrubber ユーティリティを呼び出す新しいタスクを作成します。例を次に示します。

```
scrubber -e -p new_do_pool /net/mars/vobstg/sources.vbs
```

このスクリプトは、派生オブジェクトの記憶プール new\_do\_pool に対して scrubber を呼び出します。-e オプションは、参照カウントがゼロの DO を削除します。

次に、作成したタスクをスケジューラのタスク データベースに登録し、新しいスケジュール済みジョブを作成して、1 日に数回、タスクを実行します。タスクとジョブの詳細については、229 ページの「ClearCase スケジューラ」を参照してください。

- 5 以前の DO 記憶プールをクリーンアップします。手順 2 の chpool コマンドでは、既存の DO データ コンテナは移動されません。新しい DO のデータ コンテナが保存されている場所にも影響します。したがって、以前の記憶プールをクリーンアップする必要があります。次のコマンドを実行します。

```
ccase-home-dir/etc/scrubber -e -p ddft /net/mars/vobstg/sources.vbs
```

## より控えめなスクラブ

スクラブ処理の時間が長すぎる場合（たとえば、勤務時間内に収まらないなど）、デフォルトの Daily VOB Pool Scrubbing ジョブの開始時間を早くすることができます。または、Daily VOB Pool Scrubbing ジョブを無効にし、スクラバの呼び出し方法を変更するジョブを定義します。これにより、スクラバの実行時間を短くします。

次に、DO プールのみを処理するようにスクラブを変更する方法を示します。cleartext プールはそのままになります。

- 1 次のように、実行可能プログラムからスクラバを呼び出すタスクを定義します。

```
ccase_home_dir/etc/scrubber -f -a -k do
```

- 2 スケジューラのタスク データベースにタスクに登録します。

- 3 スケジューラに、タスクを毎日実行する新しいジョブを定義します。適切な開始時間を選択します。
- 4 スケジューラで、デフォルトの Daily VOB Pool Scrubbing ジョブを無効にします。ジョブを無効にするには、ジョブの終了日を過去の時間に設定するか、ジョブを削除します。
- 5 順次スケジュールになっているほかのすべてのジョブを確認します。デフォルトの Daily VOB Pool Scrubbing ジョブの後に続いて実行されるように定義されたすべてのジョブのスケジュールを、新しく作成したジョブの後に続いて実行されるように変更します。デフォルトの Daily VOB Snapshots ジョブは Daily VOB Pool Scrubbing の後に続いて実行されるため、このジョブも同様に変更してください。

## checkvob の概要

---

VOB は、複雑な内部関係と外部関係を持ち、システムやネットワークのエラーによって、または VOB がバックアップから復元されたときに断絶が発生することがあります。checkvob は、これらの関係に発生するさまざまな問題を検出（そして多くの場合は修復も）できる cleartool サブコマンドです。たとえば、次のような問題が検出されます。

- VOB のデータベースとそのプールとの間の不整合
- 記憶プールに対する不適切な保護
- AdminVOB ハイパーリンクの不適切な設定
- グローバル タイプを上書きするローカル タイプ
- 壊れた VOB 間ハイパーリンク (UCM コンポーネントと PVOB とのリンクを含む)

次のような場合に、checkvob を使用できます。

- バックアップ メディアからの VOB の復元に vob\_restore を使用しない場合 (vob\_restore スクリプトは標準操作の一部として checkvob を実行します)
- UCM プロジェクトによって使用される ClearQuest データベースを復元する場合
- VOB を移動する場合
- 定期的な VOB メンテナンス ルーチンの一部として使用する場合

## レプリカ VOB に関する考慮事項

**注意:** ここでの説明は、ClearCase LT には当てはまりません。

checkvob はレプリカ単位の操作です。このコマンドを実行して、ローカル プールまたはローカル データベースの整合性を保ちます。checkvob を実行しても、更新に関する oplog 項目は作成されません。ただ実際には、この項目は必要なため、VOB レプリカを復元する手順では、restore replica を実行する前に checkvob が実行できる状態になっていることが必要です。checkvob -fix を実行してからレプリカの同期をとるには、restore replica コマンドを実行します。

データ消失 (欠落したコンテナ) の問題を解決するとき、**checkvob** は、ほかのレプリカに対しては、欠落したコンテナやバージョン データの検索を行いません。同様に、**rmver -data** を使用して、欠落したバージョン データを反映するように現在のレプリカのデータベースが更新された後で、**MultiSite** を使用して、別のレプリカからのバージョン データをこのデータベースに入力することはできません。この方法 (新しいブランチとバージョンを作成し、ラベルを移動するなど) を選択した場合でも、構成レコードに基づくバージョン選択は影響を受けません。つまり、以前のバージョン (バージョン データを持たなくなったもの) が選択されます。

現在のレプリカにおけるデータ消失 (欠落したコンテナ) は、同期エクスポートや同期インポートに影響を与えません。現在のレプリカでのデータ消失は、**mkreplica** を使用してのみ配信できます。この場合、新しいレプリカは、「データ消失」状態を継承します。たとえば、データ消失が、消失したバージョンを作成したレプリカで発生した場合、同期エクスポートのシナリオには次の 2 つがあります。

- 後続のエクスポート パケットに、データなしでの「バージョンの作成」操作が含まれる (**rmver -data** の次に **checkin** を実行した場合と同じ)。

システムは、データなしでの同期インポート「バージョンの作成」イベントに対するイベントレコードの最後にコメントを追加します。追加するテキストには、データなしでチェックインが発生したことが述べられ、データなしの「バージョンの作成」が発生したレプリカ (OID) の ID が書き込まれます。

- フルデータでの「バージョンの作成」操作が、既にシブリング レプリカに配信されている。シブリング レプリカがデータを維持します。

## checkvob を使用した内部 VOB 不整合の検出と修正

---

109 ページの「VOB 記憶ディレクトリ」で説明したように、VOB 記憶ディレクトリには VOB データベースと複数の記憶プールが保存されています。VOB データを変更するすべての操作は、データベースと少なくとも 1 つのプールに対して変更を行います。VOB データベースの内容とそのプールの内容との間で不整合が発生した場合、VOB データの整合性も失われている可能性があります。

そのような不整合が発生する原因としては、主に次の 2 つがあります。

- VOB が、データベースのバックアップ処理とプールのバックアップ処理が同時に行われなかったバックアップ データから復元されている。**vob\_snapshot** ユーティリティで作成されたバックアップがこれに該当します。最も一般的なバックアップ処理では、プールのバックアップがデータベースのバックアップの後で行われます。そのため、バックアップ データにデータベースが参照しないデータが含まれることがあります。

- システムまたはネットワークのエラーによって、ClearCase 操作を正常に終了できなくなっている。これにより、参照不能データ コンテナがプール内に残ったままになっていることがあります。

この項では、**checkvob** の処理方法について説明します。**checkvob** のリファレンス ページも参照してください。

## 個々のファイル エlementまたは DO の処理

個々のElementを処理する場合 (-pool オプションなしの場合)、次の手順で、ファイル Elementに対して **checkvob** を実行します。

- 1 Elementに、**-data** (欠落したコンテナ) と **-protection** の問題がないかどうかを確認します。
- 2 **-fix** モードで実行された場合、保護の問題と欠落したコンテナの問題の修復のため、次の処理を行います。

**メモ:** VOB、ソース プール、Elementのいずれかがロックされている場合、修復処理は中止されます。

- a Elementをロックします。
- b 欠落した保護の問題を修復します。
- c プールを走査して欠落したか間違って配置されたコンテナを探し、必要に応じてコンテナを再構築することによって、欠落したデータ コンテナの問題を修復します。
- d 再構築されたコンテナを参照するように VOB データベースを更新します。
- e 欠落したバージョンデータについては、**rmver -data** と同等のコマンドによって、損失したバージョンに対する参照を解除するように VOB データベースを更新します。
- f マイナー イベントをElementのイベント履歴に適用します。
- g このElementの代替の (参照不能の) コンテナをプールの **lost+found** ディレクトリに移動します。
- h Elementをアンロックします。

**checkvob** コマンドは、DO の処理と同様の方法を実行します。

- 1 **-data** と **-protection** の問題がないかどうかを確認します。
- 2 **-fix** モードで実行された場合、保護の問題と欠落したコンテナの問題の修復のため、次の処理を行います。
  - a 欠落した保護の問題を修復します。
  - b 欠落したコンテナごとに、**rmdo** と同等のコマンドを使用して DO を削除します。

## プール モードの処理

-pool オプションを指定して実行した場合、**checkvob** は VOB のソース プール、DO プール、cleartext プールの一部または全体を検査します。

プールの種類 (ソース、DO、cleartext) ごとに、次の処理を実行します。

- 1 プール ルートを確認します。プールの名前、場所、プール識別情報を確認します。各プールには有効な **pool\_id** ファイルが必要です。この段階で検出された問題は、**checkvob** では修正できません。**Rational** カスタマ サポートにお問い合わせください。
- 2 ソース プールの場合、インストールされたタイプ マネージャが **get\_cont\_info** メソッドをサポートしていることを確認します。このメソッドは、**checkvob** がエレメントを処理するときに必要です。

ソース プールと DO プールの場合のみ、次の処理を実行します。

- 3 VOB データベースを走査して、欠落した (参照されているが、見つからない) または間違って保護されているコンテナを探します。
- 4 問題のある VOB オブジェクトのリストを生成します。
- 5 -fix が指定されている場合、244 ページの「個々のファイル エレメントまたは DO の処理」の説明のとおり、リスト上の VOB オブジェクトごとに処理を行います。
- 6 参照不能データ コンテナがないことを走査します。
- 7 -fix が指定されている場合、参照不能各コンテナをプールの **lost+found** ディレクトリに移動します。

## 記憶プールの問題の説明

以下の項で、記憶プールの問題が発生する過程と **checkvob** がそれらを修復する過程について説明します。

問題の説明についてのメモ

- **予期しないイベント**。ここで説明する問題の多くは、各種イベントの副作用として発生する以外、通常はほとんど発生しません。そうしたイベントには、ネットワーク エラー、システム クラッシュ、クリーンアップ操作の失敗または中断、オペレーティング システムのバグ、ディスク エラー、ネットワークの再構成などがあります。ほかの種類のイベントには、コンテナに対する偶然または故意の削除、移動、名前変更、保護操作があります。あらゆる種類の予期しないイベントがあります。
- **重要な問題とさほど重要でない問題**。**checkvob** からのサマリ出力は、検出した多くの重要な問題とさほど重要でない問題の数を記録しています。不良プール ルートや欠落したデータ コンテナは重要な問題です。そのほかはそれほど重要な問題ではありません。

- **基準時間。**プールまたは VOB データベースの基準時間は VOB のアクティビティが最後に記録されたポイントです。これは現在の時刻、操作中の VOB がロックされた時刻、スナップショット操作またはバックアップ操作がプールやデータベースを取り込んだ時刻のいずれにもなることがあります。**checkvob** からの想定される出力と修復処理は、比較対象の VOB データベースや記憶プールの相対基準時間によって大きく異なります。一般に、次の 3 つのケースがあります。
  - データベースがプールよりも新しい。
  - プールがデータベースよりも新しい。
  - データベースと記憶プールが同期している。両方ともカレント状態であるか、VOB 記憶ディレクトリの完全なバックアップから 1 つのユニットとして取り出されている。

## プール名と保護の問題

**checkvob** は、プールに関する次の問題を検出します。

### ソース プール、DO プール、cleartext プール: 不良なプール ルート

- **説明:** 間違った名前を付けられた、または正しく特定できないプール。
- **原因:** データベースの基準時間と記憶プールの基準時間の間に **pool** の **mkpool**、**rmppool**、**rename** が実行された。
- **修復処理:** 名前と保存場所が VOB データベースに記録されている既存のプールから **pool\_id** ファイルが欠落している場合、そのファイルを作成します。**checkvob** は、プール名や保存場所に関連したほかの問題を修復することはできません。欠落した **pool\_id** ファイル以外の問題がある現在の種類のプールが見つかった場合、次のプールの種類に進みます。

### ソース プール、DO プール: Windows 上で間違って保護されたコンテナ

- **説明:** プールに、正しくない FAT RO 属性または NTFS ACL があります。
- **原因:** ユーザーが保護情報を保持せずに、プールやコンテナをコピーまたは復元した。
- **修復処理:** 必要に応じて、コンテナの RO 属性と ACL をリセットします。プールの内容に対する VOB 所有者の権限が不足している場合、**checkvob** はコンテナの ACL をレポートしますが、修復はできません。

**checkvob** が保護の問題を修復できない場合は、権限を持つユーザーとしてログオンし、次の処理を実行する必要があります。

- 1 Windows のエクスプローラで、[ファイル] メニューの [プロパティ] をクリックします。  
[セキュリティ] タブで、VOB 記憶ディレクトリにあるすべてのファイルとディレクトリの所有権を取得します。

**メモ:** 影響を受けたファイルがわずかし確認できなかった場合、これらのファイルの所有権のみを取得し、時間のかかる手順 6 の **checkvob** 操作を省略します。

- 2 VOB 記憶ディレクトリにあるすべてのファイルとディレクトリに対し、[セキュリティ] タブを使用して、すべての権限を ClearCase 管理者グループと VOB 所有者に付与します。
- 3 ログアウトします。
- 4 VOB 所有者としてログオンします。
- 5 VOB 記憶ディレクトリにあるすべてのファイルとディレクトリの所有権を取得します。
- 6 **checkvob -force -fix -protection** を実行して記憶プールの保護を修復します。

## 欠落したデータ コンテナと参照不能データ コンテナ

**checkvob** は、VOB 記憶プールの内容が VOB データベースの情報と合致するかどうかをチェックします。このチェックでは、データ コンテナをこの情報との関係に基づいて分類するため、**checkvob** が必要です。これらは次の 2 種類に分類されます。

- **欠落したデータ コンテナ。** コンテナが、VOB データベースに記録された正確な名前と場所の下に表示されない場合、このコンテナは欠落していると考えられます。この定義は、欠落したバージョンデータ (ソース プールの場合) または欠落した派生オブジェクト (DO の場合) があることを意味します。修復モード処理の中で、**checkvob** は欠落したコンテナを修復するためすべての記憶プールを走査して代替のコンテナを探し、これにより欠落したコンテナの再構築を試みます。
- **残骸。** パス名が VOB データベースに記録されていない参照不能なコンテナは残骸と考えられます。**-fix -debris** 処理の中で、**checkvob** はソース プール内にあるすべての残骸を検出し、参照不能コンテナのプロパティのチェックを実施してから、これを該当プールの **lost+found** ディレクトリに移動します。

VOB データベースと記憶プールの基準時間が異なるとき、多くの場合、**checkvob** は欠落したコンテナと参照不能コンテナの両方を検出します。たとえば、プール上での名前変更操作の結果、保存場所が変更されたコンテナを考えます。これはデータベースが参照するデータを格納しますが、記憶プール内の保存場所が違っているため、**checkvob** は欠落したコンテナと参照不能コンテナをレポートします。これにより、修復モードで問題を解決できます。

## ソースプール: 欠落したコンテナ

- **説明:** VOB データベースが想定される場所にはないソース プール データ コンテナを参照しています。
- **原因:** 次のうちのいずれかです。最新の基準時間を持つデータベースまたはプールによって異なります。
  - (データベースの方が新しい) データベースが (古い) プールで見つからなかったチェックインを記録する。

- (プールの方が新しい) 新しいチェックイン データの入っている、更新され、名前変更されたコンテナをプールが格納するが、古いデータベースはチェックイン前のコンテナ名 (もう存在しない) を参照する。
- (プールの方が新しい) プールは更新されたコンテナを格納して、**rmver** または **rmbranch** で削除されたバージョンを反映するが、データベースは古いコンテナを参照する。
- (プールまたはデータベースの基準時間が異なる) プールの基準時間とデータベースの基準時間の間にファイル エlement に対して **chpool** 操作を行い、データベースのポインティングが間違っただプールに置かれる。
- 予期しないイベント。
- **修復処理:** 参照不能コンテナがあれば、確認します。正しいデータ (または正しいデータのサブセットかスーパーセット) を持つコンテナが見つかった場合は、新しいコンテナを作成し、参照されるデータ (のサブセット) をコピーします。-**force** が指定されているか **fix?** プロンプトを受け入れる場合は、再構築されたコンテナを参照するよう **VOB** データベースを更新します。回復できないバージョンのリストを表示します。

## ソース プール: 参照不能コンテナ (残骸)

- **説明:** ソース プールには **VOB** データベースによって参照されないデータ コンテナが含まれています。それらのコンテナは一時的に残骸として分類されますが、いくつかのチェックを行ってから該当プールの **lost+found** ディレクトリに移動する必要があります。
- **原因:** 次のうちのいずれかです。最新の基準時間を持つデータベースまたはプールによって異なります。
  - (データベースの方が新しい) データベースはチェックイン後の (欠落している) 新しいコンテナの名前を参照し、プールは参照不能の古いコンテナを格納する。
  - (データベースの方が新しい) データベースは **rmver** または **rmbranch** イベントを記録し、古いプールはブランチまたはバージョンを持つコンテナを格納する。
  - (プールの方が新しい) プールには 1 つ以上のチェックイン操作によって取得したバージョンを持つコンテナがあるが、古いデータベースにはその操作が記録されていない。
  - (プールまたはデータベースの基準時間が異なる) プールの基準時間とデータベースの基準時間の間にファイル エlement に対して **chpool** 操作を行い、データベースのポインティングが間違っただプールと参照不能なコンテナに置かれる。
  - インクリメンタル バックアップからのプールの復元。
  - 予期しないイベント。

- **修復処理:** 通常、**checkvob** は参照不能コンテナをプールの **lost+found** サブディレクトリに移動します。**checkvob** は次のようなカテゴリに適合する参照不能コンテナを移動しません。
  - **必要になる可能性がある。** **checkvob** は欠落したコンテナの問題を検出しても修復を行わず、現在の参照不能コンテナのパス名は、次に実行する **checkvob** で欠落したコンテナの再構築に役立つ可能性があることを示します。**-data** を指定しないで **-fix** と **-debris** を指定すると、**checkvob** は **-data** のチェック処理を実行し、必要になる可能性のある残骸を特定しますが、**-fix** の処理は実行しません。
  - **時間不足。** コンテナの古さが 1 時間未満の場合。**checkvob** は時間不足のコンテナをスキップし、新規に作成されたコンテナを削除せずに VOB データベースを更新してそれを参照するようにします。ソース コンテナは重要なため、一般的には、コンテナの新規作成とデータベース参照の更新の間の時間が 1 秒未満の場合ですが、**checkvob** はコンテナを削除しません。
  - **削除の予定。** VOB はソース コンテナの削除を延期するように設定されており (**vob\_snapshot\_setup** のリファレンス ページを参照)、コンテナには既に削除マークが付けられています。

**メモ:** プールがデータベースより新しく、(古い) データベースには記録されていないより新しいバージョンがプールに含まれている場合、**checkvob** は参照不能コンテナからバージョンを探さず、データベースも更新しません。参照不能コンテナを使用して、プールをデータベースから想定される状態に戻し、(最新バージョンデータの入っている) 参照不能コンテナをプールの **lost+found** に移動します。これらのバージョンは消失したものと考えてください。詳細については、Rational カスタマ サポートにお問い合わせください。

## DO プール: 欠落したコンテナ

- **説明:** VOB データベースは想定される場所がない DO プール データ コンテナを参照します。
- **原因:** 次のうちのいずれかです。最新の基準時間を持つデータベースまたはプールによって異なります。
  - (データベースの方が新しい) データベースは最後にプロモートされた DO を参照し、古いプールにはその DO のコンテナが含まれない。
  - (プールの方が新しい) 新しいプールからスクラブされた DO を古いデータベースが参照する。詳細については、**scrubber** のリファレンス ページを参照してください。
  - 予期しないイベント。
- **修復処理:** **rmdo** と同等のコマンドを使用して、**checkvob** は DO コンテナの参照を削除するようにデータベースを調整します。

## DO プール: 参照不能コンテナ (残骸)

- **説明:** ソース プールには VOB データベースによって参照されないデータ コンテナが含まれています。それらのコンテナは一時的に残骸として分類され、プールの **lost+found** ディレクトリに移動されます。(scrubber は参照不能な DO データ コンテナを削除しません。参照カウントが 0 (ゼロ) の DO データ コンテナのみを削除します。)
- **原因:** 次のうちのいずれかです。最新の基準時間を持つデータベースまたはプールによって異なります。
  - (データベースの方が新しい) プールに、後でスクラブされた DO が含まれる。
  - (プールの方が新しい) DO が DO プールにプロモートされたが、古いデータベースがプロモーションを記録しない。
  - 予期しないイベント。
- **修復処理:** コンテナをプールの **lost+found** ディレクトリに移動します。

## checkvob -force -fix を使用したデータ消失の最小化

-fix 処理の中で、**checkvob** は以下のアルゴリズムを使用して、参照不能コンテナの内容を検査し、欠落したデータを回復します。データベースは、何も変更されません。ただし、**-force** と **-fix** の両方のオプションを指定したり、**fix?** プロンプトを受け入れた場合は除きます。

- 1 プールを走査して代替のコンテナを探します。プールに対する直前のパスで、**checkvob** はすべての参照コンテナを検出しました。今度はそのエレメントの参照不能コンテナを走査して、欠落したコンテナを置き換えるために使用する代替のコンテナを探し、その代替のコンテナを再結合することによって欠落したコンテナを置き換えます。2 種類の再結合操作があります。

**最適化されたリバインド:** 正しいタイプ マネージャによって維持管理される代替のコンテナを検索します。

- a 次の最適一致を検索します。

**同一:** 正しい内容を持つコンテナ (プールとデータベース間の基準時間の間にユーザーが **chpool** を実行した)。

**スーパーセット:** データベースによって想定されるバージョンのスーパーセットを持つコンテナ。プールの方がデータベースより新しい場合にはこれが一般的です。

**サブセット:** データベースによって想定されるバージョンのサブセットを持つコンテナ。データベースの方がプールより新しい場合にはこれが一般的です。

- b 最適一致をコピーし、**prune** します。新しいコンテナを作成し、最良の代替コンテナの内容をコピーします。新しいコンテナから余分なバージョン データを削除します。

**最適化されていないリバインド:** 正しいタイプ マネージャを使用する代替コンテナが見つからない場合、**checkvob** はどのようなソースを使用できる状態でも、一度に 1 つのバージョンのコンテナを構築します。このコンテナには、別のタイプ マネージャと **cleartext** プールのバージョンで維持管理されるコンテナが含まれます。

- 2 コンテナの再構築が不完全な場合は、欠落したバージョンを収集し、そのリストをレポートします。
- 3 **-force** が指定されているか **fix?** プロンプトを受け入れる場合は、VOB データベースを更新します。
  - a データベースを調整して、再構築されたコンテナを参照します。
  - b **rmver -data** と同等のコマンドでデータベースを調整して、消失したバージョン データに対する参照を解除します。
- 4 エレメントのすべての代替コンテナをプールの **lost+found** ディレクトリに移動します。

**メモ:** (参照不能の) 代替コンテナが **lost+found** に移動されると、**checkvob** の次の残骸処理時ではなく、再構築時にディスクがいっぱいになった時点で、**lost+found** からディスク領域を使用できます。再構築は大量のディスク領域を消費します。たとえば、次のようになります。

- **chtype binary\_delta \*.gif** 操作 (エレメント タイプ file) は古いデータベースには記録されません。**checkvob** は新しいプールの参照不能デルタ コンテナを使用して、古いデータベースによって想定される、欠落したコンテナのコピー全体を再構築します。
- **chpool** 操作は、古い記憶プールによって反映されません。**checkvob** は参照不能データ コンテナを検索し、数百、数千のコピーを作成します。

**-force -fix** オプションを使用すると、**checkvob** は不本意なデータ消失が起こらないようにします。

- **¥branch¥0 以外のすべてのバージョンの削除を許可しない:** 強制修復モードでは、バージョン番号が 0 より大きいバージョンが少なくとも 1 つ以上なければ、**checkvob** は VOB データベースを更新してエレメントの欠落したデータ コンテナを反映せず、データはそのまま残ります。すなわち、**-force** を指定しないで **checkvob** を **-fix** モードで実行し、唯一の残りバージョンのバージョン ID が **¥main¥br1¥0** と **¥main¥br1¥br2¥0** になっている再構築されたエレメントを受け入れるかどうかを確認するメッセージが表示された場合は、それに同意する必要があります。
- **データ消失を認めるようプロンプト要求を行う強制修復モードでは、checkvob は次のような確認メッセージを表示します。**

```
Do you want to override the default and allow fixing of
elements involving missing version data? [no]
```

これに「yes」で応答すると、**checkvob** は、データ消失を許容できる (つまり、消失してかまわない) 時間間隔を指定するように要求します。たとえば、24 時間前のバックアップからソース プールを復元すると、過去 24 時間に作成されたバージョン データは消失します。この場合、次のコマンドを実行します。

**checkvob -force -fix -data -pool -source VOB-stg-pname**

表示される Allow missing data created since: プロンプトに応答して、**checkvob** に対し、その時刻以降に VOB データベースに作成され記録されたデータの消失を許可するように指示することができます。

**checkvob** が指定された時刻以前に作成されたデータを持つコンテナを見つけられない場合、この事実は出力ログに記録されますが、**-force** オプションを指定しないで **checkvob** を再度実行してそのようなエレメントを個別に処理するか、データ消失時刻を調整するまで、データ消失は許可されません。

作成時にかかわらずすべての欠落データ コンテナを無条件で許可 (解決) するには、非常に古い日付と時間を使用します。許可されるデータ消失のデフォルトの時間間隔は「昨日の 00:00:00 以降」です。1 週間以上前の日付を指定すると、**checkvob** により確認メッセージが表示されます。

**checkvob** ログ ファイルは **-force -fix** の操作から時間間隔のダイアログを取り込みません。

受け入れ可能な値のリストについては、**lshistory** のリファレンス ページの **date-time** 引数の説明を参照してください。

**メモ:** **checkvob** は壊れたデータ コンテナの検出も修復も行いません。正しい識別情報を正しい場所に持つコンテナは、**checkvob** では、正常と見なされます。

## **checkvob を使用したグローバル タイプの問題の検出と修正**

---

**checkvob** は、VOB での内部不整合の識別と修正だけでなく、管理 VOB 階層に関する問題の識別と修正を行うことができます。たとえば、次のような問題の修正と識別を行うことができます。

- 複数の AdminVOB ハイパーリンクを持つ VOB
- 名前が一致しないローカル コピーを持つグローバル タイプ
- 見えなくなっているタイプ
- グローバル タイプのローカル コピーに対する見えなくなっているロック
- グローバル タイプとそのローカル コピーの間での保護の不一致

階層内のグローバル タイプの問題を検出し、必要に応じて修正するには

- 1 階層内の任意の VOB 上で、**checkvob -global** を実行します。**checkvob** がその VOB で複数の **AdminVOB** ハイパーリンクを検出した場合、処理を停止し、問題の修正を要求するメッセージを表示します。
- 2 問題を修正するには、**cleartool describe -long** を使用して VOB ハイパーリンクを一覧表示し、次に **rmhlink** を使用して、ハイパーリンクを 1 つだけ残して削除します。この例では、**VOB %sources** には 2 つの **AdminVOB** ハイパーリンクがあり、**checkvob** が階層のチェックを続行する前にいずれか 1 つを削除する必要があります。

```
cleartool describe -long vob:%sources
versioned object base "%sources"
...
```

ハイパーリンク :

```
AdminVOB@2@%sources -> vob:%projects
```

```
AdminVOB@3@%sources -> vob:%admin1
```

```
cleartool rmhlink AdminVOB@3@%sources
```

ハイパーリンク "AdminVOB@3@%sources" を削除しました。

- 3 階層が適切に修正されたら、**checkvob -global** を再実行して階層内のグローバル タイプに関する問題がないことを確認します。**checkvob** が検出できる問題のタイプは、次のとおりです。
  - 見えなくなっているローカル タイプ
  - 見えなくなっているローカル ロック
  - 保護の不一致
  - 名前の不一致
- 4 ログ ファイルを確認し、**checkvob -global -fix** を実行して問題を修正します。**-fix** オプションを指定すると、**checkvob** は次の処理を実行します。
  - 各ローカル コピーの名前を、そのグローバル タイプの名前と一致するように変更します。
  - 見えなくなっているローカル コピーや通常タイプを取得しようとします。取得するタイプがロックされている場合、グローバル タイプにロックが設定されていれば、ロックは破棄されます。グローバル タイプにロックが設定されていなければ、最初に取得したタイプのロック情報がグローバル タイプに適用されます。
  - 各ローカル タイプの所有権を、そのグローバル タイプの所有権と一致するように変更します。

**-force** が指定されなかった場合、**checkvob** は、検出した問題の修正を試行する前に確認メッセージを表示します。

## checkvob を使用した破損ハイパーリンクの検出と修正

---

ハイパーリンク モード (-hlinks) では、checkvob は、リンクが壊れているなどの原因で rmhlink によって削除できなかったハイパーリンクをクリーンアップします。checkvob は、ユーザーが指定する各オブジェクトのハイパーリンクを調べて正常かどうかを判定します。一般に、ハイパーリンクは VOB 間ハイパーリンクの片方のオブジェクトが使用できなくなったときに壊れます。checkvob は、検出した機能しない個々のハイパーリンクを削除する前に確認メッセージを表示します。-force を使用すると、これらのメッセージを表示しないようにすることができます。

**メモ:** VOB が指定された場合、checkvob -hlinks は、対象 VOB のすべてのハイパーリンクを調べるわけではありません。その VOB オブジェクト自身に接続されているハイパーリンクのみを調べます。

## UCM 環境での checkvob の使用法

---

UCM 環境 (特に、Rational ClearQuest と統合されている環境) では、プロジェクトの VOB とその ClearQuest データベースとの複雑な関係を作成できます。checkvob には、-ucm オプションで利用できる特別な機能があります。このオプションを使用すると、関連するデータベースセットの 1 つ以上のメンバがバックアップから復元されて発生する問題や、システムまたはネットワークのエラーから影響を受ける問題を検出し、修正することができます。206 ページの「関連付けられたデータベースの一括バックアップ」で説明した方法に従って VOB と ClearQuest データベースのバックアップをスケジュールすると、これらのバックアップによる復元時間の不整合を最小限に抑えることができます。225 ページの「関連付けられたデータベースのグループに属するメンバの復元」で説明した方法に従って checkvob -ucm を使用すると、関連するデータベースセットの 1 つ以上のメンバを復元した後の問題を、検出し修正することができます。また、予期しないシステム イベントやネットワーク イベントが原因で発生した、UCM 環境におけるデータベース関係の不整合も、検出し修正することができます。

**メモ:** checkvob -ucm を実行してから checkvob -hlinks を実行してください。checkvob -hlinks は、checkvob -ucm が修正できるハイパーリンクも削除する可能性があります。

## 動的ビューのビュー記憶領域のメンテナンス

---

動的ビューのビュー記憶領域のメンテナンスには、次の 2 つの基本的なタスクがあります。

- 不要なビューの削除。ビューは有用でなくなっても存在し続けるため、削除できるビューがないかどうか、定期的に検討する必要があります。ビュー内のバージョン付き成果物と DO は、VOB に保存された情報で完全に再構築できます。ただし、ビュープライベート ファイルは再構築できません。無効な動的ビューを使用することは、記憶領域が浪費されるだけでなく、検索対象の DO が残っている場合に、ビルド パフォーマンスに悪影響を与えることになります。

- 不要なビュープライベート オブジェクトの削除。ビューのプライベート記憶領域には不要なファイルが累積される傾向があります。たとえば、一時ファイル、テキスト エディタのバックアップ ファイル、メール メッセージやソース ファイルの抜粋など、さまざまなものが累積されます。

ユーザーに対して、独自ビューの定期的な確認、削除、クリーンアップを奨励してください。

**メモ:** ホストのネイティブ ファイル システムを使用する、DO の作成をサポートしていないスナップショット ビューでは、特別な記憶領域メンテナンス処理は必要ありません。

## ビューの内容に関する情報の取得

ClearCase 管理コンソールに、ビューで使用されているディスク容量に関する情報が表示されます。

- VOB 記憶ノードの [派生オブジェクト] サブノードに、VOB 内の共有派生オブジェクトで使用されているディスク容量が表示されます。これらの DO をどのビューが参照しているかの情報も表示されます。
- ビューのビュー記憶ノード (ビュー記憶ディレクトリが置かれているホストのホスト ノードのサブノード) に、そのビューで使用されている現在のディスク容量と履歴が表示されます。
- 動的ビューのビュー記憶ノードの [プライベート ファイル] サブノードに、ビュープライベート オブジェクトが一覧表示されます。その中に、そのビュー内のファイルと派生オブジェクトが含まれます。

次のいくつかの **cleartool** サブコマンドを使用して、ビュー内で使用されているディスク容量に関する情報を表示することもできます。

- **dospace** コマンドは、VOB 内の共有派生オブジェクトで使用されているディスク容量を表示します。これらの DO をどのビューが参照しているかの情報も表示されます。
- **space -view** コマンドは、ビューに使用されている現在のディスク容量と履歴を表示します。

ClearCase スケジューラによって、ビューに使用されているディスク容量に関するデータを収集するための次のジョブが実行されます。

- ビューによって使用されているディスク容量に関する毎日のデータ収集
- 共有派生オブジェクトのディスク使用量に関する毎週のデータ収集
- ユーザー独自のプログラムを実行するようにカスタマイズできるジョブの毎日および毎週の実行

詳細については、229 ページの「ClearCase スケジューラ」を参照してください。

## ビュープライベート記憶領域のスクラブ

ある DO が最初に **clearmake**、**omake**、**clearaudit** のいずれかによってビルドされると、その DO のデータ コンテナが当該ユーザーのビューのプライベート記憶領域に配置されます。**clearmake** または **omake** ビルドの間に DO が最初にウィंकインされると、そのデータ コンテナが VOB の派生オブジェクト記憶プールにコピーされます (データ コンテナは、移動ではなくコピーされます。データ コンテナを移動すると、その DO にアクセス中のユーザー プロセスが中断されるためです)。その結果、データ コンテナの余分なコピーがビュープライベート記憶領域に残されることになります。**winkin** または **view\_scrubber -p** コマンドを使用して派生オブジェクトをウィंकインした場合、VOB 記憶プールにプロモートされた後で、そのビュー内のデータ コンテナは削除されます。

通常、データ コンテナの余分なコピーに対しては何の処理も必要ありません。それは次の理由によります。

- ビルドのために頻繁に使用されるビューでは、ビルド スクリプトによって古い (重複する可能性のある) DO データ コンテナが新しいコンテナに置き換えられる。
- ビューには DO の余分なコピーが DO ごとに最大 1 つしか存在できない (この状況は VOB と大きく異なります。**scrubber** ユーティリティを一度も実行しない場合、VOB には使用されなくなった 多くの DO が累積されます)。

これらの理由から明らかなように、ディスク領域が極端に不足しているのでなければ、ビュープライベート記憶領域内の余分なデータ コンテナをわざわざクリーンアップする必要はありません。

重複する余分な DO データ コンテナをビューのプライベート記憶領域からどうしても削除しなければならない場合は、**view\_scrubber** ユーティリティを使用します。このユーティリティでは、非共有または共有不可の DO を VOB の記憶領域に移行することもできます。

詳細については、**view\_scrubber** のリファレンス ページを参照してください。

## ビューの手動クリーンアップ

ユーザーは、オペレーティング システムが提供するファイルの削除ツールを使用して、不要なファイルをビューから削除できます。VOB に記述されているパス名でビュープライベートファイルを表示するには、ClearCase 管理コンソールでビュー記憶ノードの [プライベート ファイル] サブノードを使用するか、次の **lsprivate** コマンドを使用します。

```
cleartool lsprivate
¥proj¥lib¥pick.o
¥proj¥lib¥spar.o
¥proj¥lib¥get.c  [チェックアウトされています]
¥proj¥lib¥get.c~
¥proj¥lib¥querytty.c  [チェックアウトされています]
¥proj¥lib¥querytty.c~
¥proj¥lib¥strut.c  [チェックアウトされています]
.
.
.
```

「[チェックアウトされています]」と表示されているファイルは削除しないでください。「not available - deleted perhaps?」と表示されたファイルがある場合は、次の処理を行って、ファイルを復元します。

**メモ:** この手順は動的ビューのみに適用されます。スナップショット ビューの場合は、**cleartool ls -recurse -view\_only** コマンドを使用して、ビュー内のファイルを検査することができます。

- 1 取り残されたファイルのいずれを削除するかを決定します。取り残されたファイルは、現在アクセス不能になっている VOB またはビュー ディレクトリに一度カタログされています。その中には二度とアクセスできないものもあります。**lsprivate** のリファレンス ページを参照して、取り残されたファイルの詳細を確認し、削除するファイルを決定します。通常は、ファイルを個別に選択するのではなく、削除するビュープライベート ファイルが保存されているディレクトリや VOB 全体を選択します。

- 2 取り残されたオブジェクトの UUID を修正します。削除すべきファイルのそれぞれの VOB やディレクトリの UUID を調べます。たとえば、次に示す **lsprivate** からの出力行には、**util.cxx** という名前の、取り残されたファイルに関する情報が示されています。

```
<VOB-beeb313c.0e8e11cd.ad8e.08:00:69:06:af:65>
<DIR-375b5ca0.0e9511cd.ae20.08:00:69:06:af:65>¥util.cxx
```

この例で、取り残されたファイルが所属する VOB の UUID は以下です。

```
beeb313c.0e8e11cd.ad8e.08:00:69:06:af:65
```

取り残されたファイルが作成された VOB ディレクトリの UUID は以下です。

```
375b5ca0.0e9511cd.ae20.08:00:69:06:af:65
```

- 3 取り残されたファイルをビューの **lost+found** ディレクトリに移動します。取り残されたファイルの集合を削除するために、**recoverview** コマンドを使用してそれらのファイルをビューの **lost+found** ディレクトリに移動します。次のコマンドは、**util.cxx** と同じディレクトリに作成された、すべての取り残されたビュープライベート ファイルを移動します。

```
cleartool recoverview -dir 375b5ca0.0e9511cd.ae20.08:00:69:06:af:65 -tag r2integ  
ファイル ccsvr03:¥vws¥integ¥.s¥lost+found¥57FBB6DF.0418.util.hxx を  
移動しました  
ファイル ccsvr03:¥vws¥integ¥.s¥lost+found¥2203B56D.00C2.util.cxx を  
移動しました
```

この例では、**recoverview** は 2 つのファイル **util.hxx** と **util.cxx** を **lost+found** ディレクトリに移動します。

- 4 **lost+found** ディレクトリからファイルを削除します。ここでは、標準のファイル削除コマンドを使用して、取り残されたファイルを削除することができます。

```
cd ¥vws¥integ¥.s¥lost+found  
c:¥vws¥integ¥.s¥lost+found> del 57FBB6DF.0418.util.hxx  
c:¥vws¥integ¥.s¥lost+found> del 2203B56D.00C2.util.cxx
```

Rational ClearCase には、ほかの構成管理システムから、または直接ファイル システム自体からデータをエクスポートするために使用するユーティリティがいくつか用意されています。ここでは、データを ClearCase にインポートする方法の例をいくつか示します。

**メモ:** UCM ビューでは、この章で説明するどの `clearimport` 処理も実行できません。データを UCM プロジェクトにインポートするには、最初にデータを非 UCM ビューにインポートし、『Rational ClearCase プロジェクト管理ガイド』で説明している手順に従ってプロジェクトをセットアップする必要があります。

## Microsoft SourceSafe からのデータのインポート

---

この例では、コマンド行ツールを使用して SourceSafe から新規の VOB にデータをインポートします。まず SourceSafe のサンプル構成について説明し、それをインポートする方法を示します。次に、さまざまな SourceSafe 機能がインポートでどのように処理されるかを検証します。

この例では、インポートしたデータを保持するために、¥payroll という名前の VOB を作成してマウントしたものと仮定します。

### SourceSafe のサンプル構成の概要

サンプルの Payroll 構成には Payroll という名前のアプリケーションをビルドするために使用するソース ファイルがあります。

図 7 SourceSafe のサンプル構成

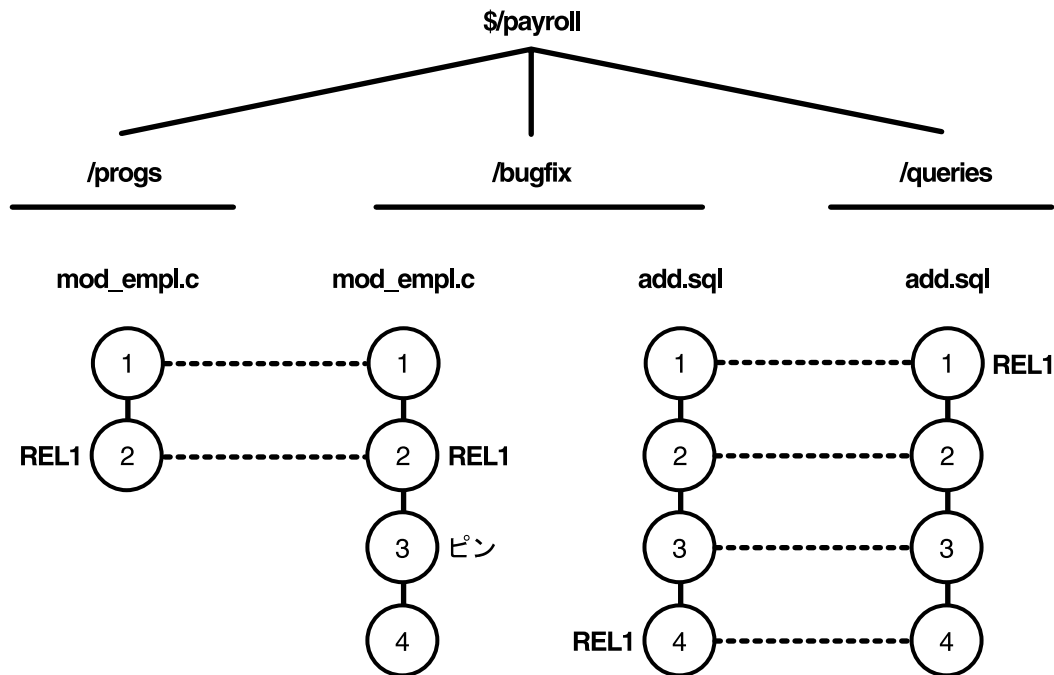


図 7 は、Payroll 構成の SourceSafe での階層構造を示したものです。

- **\$/payroll** プロジェクト。
- **/progs** サブプロジェクト。ソース コードを保存します。
- **/bugfix** サブプロジェクト。開発者は障害の修正をここに保存します。
- **/queries** サブプロジェクト。SQL クエリーを保存します。

Payroll 構成には、次のオブジェクトが含まれています。

- **共有** : SourceSafe では、共有はハード リンクに似ています。この例では、**\$/payroll/bugfix/add.sql** と **\$/payroll/queries/add.sql** の 2 つのファイルを SourceSafe で共有しています。SourceSafe では、一方の共有ファイルを変更すると、もう一方の共有ファイルにも反映されます。

- **ブランチ**: SourceSafe では、ブランチを作成する前に共有を作成する必要があります。  
\$payroll/progs/mod\_empl.c と \$payroll/bugfix/mod\_empl.c の 2 つのファイルはバージョン 1 と 2 で共有されます。バージョン 3 では、\$payroll/bugfix/mod\_empl.c ファイルが独自のブランチを形成します。
- **ラベル**: ラベルは特定のバージョンやプロジェクトを識別するためのものです。この構成では、REL1 ラベルを使用して、Payroll アプリケーションの最初のリリースのビルドに使用するソース ファイルのバージョンを特定します。
- **ピン**: SourceSafe はデフォルトで最新バージョンのファイルを使用します。ピンを使用して、最新バージョン以外のバージョンを使用するように SourceSafe に指示することができます。Payroll 構成では、\$payroll/bugfix/mod\_empl.c のバージョン 3 にピン設定されています。

clearexport\_ssafe は、変換プロセスのエクスポート段階でほとんどのオブジェクトを識別して処理します。

## 環境の設定

変換プロセスを開始する前に、Windows の PATH、TMP 環境変数が正しく設定されていることと、SourceSafe の現在のプロジェクトが指定されていることを確認します。

- PATH 環境変数に SourceSafe の実行可能ファイルの場所が含まれていることを確認します。
- TMP 環境変数には、clearimport がテンポラリ ファイルを作成できる場所を設定します。例を次に示します。

```
set TMP=c:\temp
```

- clearexport\_ssafe ユーティリティは、SourceSafe の現在のプロジェクトにあるファイルとディレクトリをエクスポートします。clearexport\_ssafe を実行する前に、SourceSafe の現在のプロジェクトを調べて、clearexport\_ssafe が目的のファイルやディレクトリをエクスポートするよう設定されていることを確認します。例を次に示します。

```
ss cp
```

カレント プロジェクトは \$/payroll/bugfix です。

```
ss cp ..
```

カレント プロジェクトは \$/payroll です。

```
ss dir
```

```
$/payroll
```

```
$bugfix
```

```
$progs
```

```
$queries
```

```
3 item(s)
```

## clearexport\_ssaf の実行

clearexport\_ssaf ユーティリティは、SourceSafe の現在のプロジェクト内にファイルとサブプロジェクトを読み込んでデータ ファイルを生成します。clearimport はこのデータ ファイルを読み込んで、対応する VOB エlementを作成します。clearexport\_ssaf は、デフォルトではこのデータ ファイルに cvt\_data という名前を付けて現在のディレクトリに保存します。データ ファイルに別の名前と記憶場所を指定するには、-o オプションを使用します。

clearexport\_ssaf は、SourceSafe の現在のプロジェクト内にあるファイルとサブプロジェクトはエクスポートしますが、デフォルトではサブプロジェクト内にあるファイルはエクスポートしません。たとえば、SourceSafe の現在のプロジェクトが \$/payroll の場合、clearexport\_ssaf は、/progs、/bugfix、/queries の 3 つのサブプロジェクトをエクスポートしますが、各サブプロジェクト内のファイルはエクスポートしません。サブプロジェクト内のファイルもエクスポートするには、-r オプションを指定します。

次の例では、SourceSafe の現在のプロジェクトは \$/payroll です。コマンド行で -r オプションが指定されているので、clearexport\_ssaf は /progs、/bugfix、/queries の 3 つのサブプロジェクト内のファイルもすべてエクスポートします。また、-o オプションが指定されているので、clearexport\_ssaf は c:¥datafiles ディレクトリの paycvt という名前のデータ ファイルに出力を保存します。

```
ss cd
```

```
カレント プロジェクトは $/payroll です。
```

```
clearexport_ssaf -r -o c:¥datafiles¥paycvt
```

```
VOB のディレクトリ エlement "." です。
```

```
VOB のディレクトリ エlement "bugfix" です。
```

```
Element "bugfix¥add.sql" を変換しています ...
```

```
Element履歴を抽出しています ...
```

```
....
```

```
完了しました。
```

```
Elementを変換しています ...
```

```
Elementを作成しています ...
```

```
Element "progs¥new_empl.c" が完了しました。
```

```
VOB のディレクトリ エlement "queries" です。
```

```
Element "queries¥add.sql" を変換しています ...
```

```
Element履歴を抽出しています ...
```

```
...
```

```
完了しました。
```

```
Elementを変換しています ...
```

```
Element "queries¥add.sql" が完了しました。
```

```
Element "queries¥update.sql" を変換しています ...
```

```
Element履歴を抽出しています ...
```

```
.
```

```
.
```

```
.
```

## clearimport の実行

任意のビューを使用します。**clearimport** は構成仕様を無視します。動的ビューを使用すると、インポート操作のパフォーマンスが向上します。

作業ディレクトリを、構成をインポートする予定の VOB ディレクトリに設定します。**-d** オプションで VOB ディレクトリを指定すれば、VOB ディレクトリ以外の場所から **clearimport** を実行できます。**clearexport\_ssaf**e で作成したデータ ファイルのパス名を指定する必要があります。

```
clearimport c:¥datafiles¥paycv
```

ラベル タイプを確認しています。

ディレクトリおよびシンボリック リンクを確認しています。

エレメントを確認しています。

エレメント ".¥bugfix/add.sql" を作成しています。

バージョン "1"

バージョン "2"

バージョン "3"

バージョン "4"

エレメント ".¥bugfix/mod\_empl.c" を作成しています。

バージョン "1"

バージョン "2"

バージョン "3"

バージョン "4"

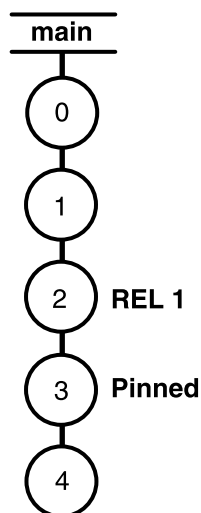
.  
. .  
.

ディレクトリを閉じています。

## 結果の検証

**clearimport** による VOB の配置が完了したら、新しく作成された VOB エレメントのバージョン ツリーを確認し、**clearexport\_ssaf**e と **clearimport** により SourceSafe の構成が意図したとおりに変換されたことを確認します。ClearCase エクスプローラまたは Windows エクスプローラでフォルダを開き、エレメントを選択します。次に、そのバージョン ツリーを表示します。図 8 に、SourceSafe /bugfix プロジェクトの mod\_empl.c エレメントのバージョン ツリーを示します。

図 8 インポートされた要素のバージョン ツリー



- **ブランチ**: SourceSafe の構成では、\$payroll/bugfix/mod\_empl.c ファイルのバージョン 3 に独自のブランチが形成されていました。clearexport\_ssaf は、SourceSafe のブランチから ClearCase ブランチへの変換は行いません。その代わりに、clearexport\_ssaf は独立したエレメントを作成します。この例では、mod\_empl.c エレメントのバージョン 1 と 2 を ¥progs ディレクトリに作成し、mod\_empl.c エレメントのバージョン 1 ～ 4 を ¥bugfix ディレクトリに作成します。
- **バージョン番号**: すべてのエレメントに対し、ClearCase はバージョン ツリーのルートにバージョン 0 を作成します。
- **ラベル**: この変換プロセスで、SourceSafe ラベルが直接 ClearCase のラベルにマッピングされるので、SourceSafe の構成と同様に mod\_empl.c のバージョン 2 に REL1 ラベルが付きます。
- **ピン**: ClearCase には SourceSafe のピンに相当する機能がありません。ピンはラベルと同じ機能を果たす場合があるので、変換プロセスでピンがラベルにマッピングされます。SourceSafe の構成では、mod\_empl.c のバージョン 3 にピン設定されているので、この変換プロセスでは PINNED という名前のラベルが適用されます。
- **共有**: ClearCase には SourceSafe 共有に相当する機能がないため、clearexport\_ssaf は、変換時に共有をハード リンクとして保存しません。代わりに、共有が個別のエレメントになります。

## RCS データのインポート

---

この例では、コマンド行ツールを使用して UNIX RCS ソース ツリーから新規の VOB にデータをインポートします。RCS ツリーのルートは `/usr/libpub` で、同じホストの `/vobs/libpub` で空の VOB がアクティブになっているものとします。

### データ ファイルの作成

- 1 ソース データの場所に移動します。既存の RCS ソース ツリーのルート ディレクトリに移動します。
- 2 エクスポート ユーティリティを実行します。`clearexport_rcs` を使用して、データ ファイルを作成し、RCS ファイルの説明 (`.v` ファイル) を記述します。

```
cd /usr/libpub
```

```
clearexport_rcs
Exporting element "./Makefile,v" ...
エレメント履歴を抽出しています ...
.
完了しました。
Exporting element ...
エレメントを作成しています ...
エレメント "./Makefile" が完了しました。
.
.
エレメント "./lineseq.c" が完了しました。
Creating datafile ./cvt_data ...
```

現在のディレクトリに `cvt_data` という名前のデータ ファイルが作成されます。

### clearimport の実行

- 1 任意のビューを使用します。`clearimport` は構成仕様を無視します。動的ビューを使用すると、インポート操作のパフォーマンスが向上します。
- 2 ターゲット VOB に移動します。`/vobs/libpub` のルート ディレクトリに移動します。
- 3 `clearimport` を実行します。作成したデータ ファイル (`cvt_data`) に対して `clearimport` を実行し、`/vobs/libpub` にデータを配置します。

```
clearimport /usr/libpub/cvt_data
Converting files from /usr/libpub to .
エレメント "././Makefile" (タイプ "text_file") を作成しました。
"././Makefile" の保護を変更しました。

Making version of ././Makefile

バージョン "/main/0" から "././Makefile" をチェックアウトしました。
リストにあるすべてのオブジェクトに関するコメント:
"././Makefile" のバージョン "/main/1" をチェックインしました。
.
.
```

VOB のルートディレクトリに入っているエレメントは、明示的にチェックアウトまたはチェックインする必要はありません。これは自動的に行われます。**clearimport** の実行中にエラーが発生して途中で終了した場合は、問題を解決してからもう一度 **clearimport** を実行します。

## PVCS データのインポート

---

この例では、コマンド行ツールを使用して PVCS から新規の VOB にデータをインポートします。PVCS ツリーは C:¥libpub にあり、同じホストの ¥libpub で空の VOB がアクティブになっているものとします。

### データ ファイルの作成

- 1 ソース データの場所に移動します。既存の PVCS ソース ツリーのディレクトリに移動します。  
c:¥> cd libpub
- 2 エクスポート ユーティリティを実行します。**clearexport\_pvcs** を使用して、データ ファイルを作成し、PVCS ファイルの説明を記述します。

```

c:¥libpub> clearexport_pvcs
Exporting element ".¥makefile" ...
エレメント履歴を抽出しています ...
.
完了しました。
Exporting element ...
エレメントを作成しています ...
エレメント ".¥makefile" が完了しました。
.
.
エレメント ".¥lineseq.c" が完了しました。
Creating datafile .¥cvt_data ...

```

現在のディレクトリに **cvt\_data** という名前のデータ ファイルが作成されます。

## 変換スクリプトの実行

- 1 任意のビューを使用します。**clearimport** は構成仕様を無視します。動的ビューを使用すると、インポート操作のパフォーマンスが向上します。
- 2 ターゲット VOB に移動します。¥libpub VOB のルート ディレクトリに移動します。
- 3 **clearimport** を実行します。作成したデータ ファイル (**cvt\_data**) に対して **clearimport** を実行し、¥libpub にデータを配置します。

```

z:¥vob_libpub> clearimport c:¥libpub¥cvt_data
Converting files from c:¥libpub to .
エレメント ".¥.¥makefile" (タイプ "text_file") を作成しました。
".¥.¥makefile" の保護を変更しました。

```

```
Making version of .¥.¥makefile
```

```

バージョン "¥main¥0" から ".¥.¥makefile" をチェックアウトしました。
リストにあるすべてのオブジェクトに関するコメント:
".¥.¥makefile" のバージョン "¥main¥1" をチェックインしました。
.
.

```

VOB のルート ディレクトリに入っているエレメントは、明示的にチェックアウトまたはチェックインする必要はありません。これは自動的に行われます。**clearimport** の実行中にエラーが発生して途中で終了した場合は、問題を解決してからもう一度 **clearimport** を実行します。

ClearCase のエクスポートとインポートの各ユーティリティの詳細については、**clearexport\_ccase**、**clearexport\_pvcs**、**clearexport\_rcs**、**clearexport\_sccs**、**clearexport\_ssaf**、**clearexport\_cvs**、**clearimport**、**clearfsimport** の各リファレンス ページを参照してください。



この章では、次のいずれかの作業を行う場合の処理について説明します。

- NTFS VOB またはビュー記憶領域の保護問題を修復する
- 参照先のエレメントの再配置後に既存のビューを使用する
- ファイルを保存するために MVFS に作成された物理コンテナを検索する
- /view ディレクトリの再帰探索を防止するために UNIX crontab を編集する

## NTFS の記憶ディレクトリ ACL の修復

---

Windows コンピュータ上で VOB とビューの記憶ディレクトリを保持する場合は、NTFS フォーマットのディスクを使用することをお勧めします。NTFS ファイルシステムのオブジェクトは、所有権情報とアクセス コントロール リスト (以下 ACL) を含むセキュリティ記述子で保護されています。FAT ファイルシステムは ACL をサポートしないので、FAT ファイルシステムのオブジェクトは、**readonly** 属性のみで保護されます。この属性は、NTFS と FAT のどちらでも使用できますが、必須の属性ではないので、簡単に削除できます。

NTFS では、VOB またはビューの記憶ディレクトリの所有権情報 (所有者とプライマリ グループ ID) が、記憶ディレクトリのルートのセキュリティ記述子から決定されます。FAT ファイルでは、この情報が記憶ディレクトリのルートにある **identity.sd** ファイルに保存されます (互換性を維持するために、NTFS ファイルシステムにも **identity.sd** ファイルが作成されます)。FAT と NTFS の **groups.sd** ファイルには、VOB 補足グループ リストが保存されています。

## VOB と ビューの記憶ディレクトリの ACL

Rational ClearCase では、VOB とビューが作成されるときに、VOB 記憶ディレクトリとビュー記憶ディレクトリの ACL を設定します。ACL は ClearCase が利用する特殊なフォームで構成されています。次の例では、プライマリ グループが **user** であるユーザー **NT\_WEST¥ccase\_adm** が作成した、VOB 記憶ディレクトリ **sources.vbs** の正しい ACL を示します。ClearCase 管理者グループ名は **clearcase** です。

**メモ:** 例で説明するように、ClearCase が ClearCase 管理者グループを使用する場合、ClearCase LT では必ず組み込み ID **LocalSystem (NT\_AUTHORITY¥SYSTEM)** を使用します。またこの例では、ClearCase LT サーバー上で、名前 **clearcase** の場所に、名前 **NT\_AUTHORITY¥SYSTEM** が表示されます。ClearCase と ClearCase LT では、最初の ACL エントリ (**NT\_AUTHORITY¥NETWORK**) は、ビュー記憶ディレクトリではなく、VOB 記憶ディレクトリ

にのみに存在します。VOB 記憶ディレクトリが NAS デバイス上にある場合、このエントリは存在しません。

**cacls c:¥vobstore¥sources.vbs**

NT AUTHORITY¥NETWORK: (OI) (CI) (DENY) (special access:) (VOB 記憶領域のみ)

DELETE  
FILE\_WRITE\_DATA  
FILE\_APPEND\_DATA  
FILE\_WRITE\_EA  
FILE\_WRITE\_ATTRIBUTES

NT\_WEST¥user: (CI) R (VOB のプライマリ グループ)

Everyone: (CI) R

NT\_WEST¥ccase\_adm: (CI) (special access:) (VOB 所有者)

STANDARD\_RIGHTS\_ALL  
DELETE  
READ\_CONTROL  
WRITE\_DAC  
WRITE\_OWNER  
SYNCHRONIZE  
STANDARD\_RIGHTS\_REQUIRED  
FILE\_GENERIC\_READ  
FILE\_GENERIC\_WRITE  
FILE\_GENERIC\_EXECUTE  
FILE\_READ\_DATA  
FILE\_WRITE\_DATA  
FILE\_APPEND\_DATA  
FILE\_READ\_EA  
FILE\_WRITE\_EA  
FILE\_EXECUTE  
FILE\_READ\_ATTRIBUTES  
FILE\_WRITE\_ATTRIBUTES

NT\_WEST¥clearcase: (CI) F (ClearCase LT は組み込み ID

である NT AUTHORITY¥SYSTEM を使用)

NT\_WEST¥user: (OI) (IO) (special access:) (VOB のプライマリ グループ)

GENERIC\_READ  
GENERIC\_EXECUTE

Everyone: (OI) (IO) (special access:)

GENERIC\_READ  
GENERIC\_EXECUTE

```
NT_WEST¥ccase_adm:(OI)(IO)(special access:)                (VOB 所有者)
    DELETE
    WRITE_DAC
    WRITE_OWNER
    GENERIC_READ
    GENERIC_WRITE
    GENERIC_EXECUTE

NT_WEST¥clearcase:(OI)(IO)F                                  (ClearCase LT は組み込み ID
である NT AUTHORITY¥SYSTEM を使用)
BUILTIN¥Administrators:(OI)(CI)F
```

## VOB または ビューの記憶ディレクトリをコピーするとき NTFS ACL を保持する

VOB またはビューの記憶ディレクトリを NTFS ボリューム内の別の場所や別の NTFS ボリュームに移動する場合は、常に、記憶ディレクトリの ACL を保持するコピー プログラムを使用する必要があります。

- 通常、VOB またはビューの記憶ディレクトリをコピーする場合は、**ccopy** ユーティリティ (**ccase-home-dir¥etc¥utils¥ccopy**) を使用することをお勧めします。**ccopy** では、必要な ACL 情報をすべてコピーしますが、オブジェクトの完全なセキュリティ記述子はコピーしません。そのため、**ccopy** を効果的に使用すると、このコマンドを実行するユーザーに、コピーしたディレクトリへの完全なアクセス権を与えることができます。VOB またはビューの所有者以外のユーザーが VOB またはビューの記憶ディレクトリをコピーする場合は、**xcopy /o** を使用します。Windows NT ではこのコマンドの代わりに **scopy** を使用します。
- Windows 2000 または Windows XP では、**xcopy** に **/o** オプションを指定して使用できます。  
Windows 2000 と Windows XP では、各ディレクトリが親ディレクトリから ACL を継承できるため、コピーしたディレクトリも親ディレクトリの ACL のエントリを継承できます。これにより、VOB とビューへのアクセスに問題が生じる場合があります。
- Windows NT で **scopy** コマンドを使用する場合は、Windows NT リソース キットから使用し、**/o** オプションを指定します。

**メモ:** **xcopy** または **scopy** を使用する場合は、サブディレクトリのコピーに必要な追加オプションを指定してください。また、ディレクトリを作成する権限とターゲット ホストで ACL を設定する権限を持つユーザーとしてログオンする必要があります。管理者グループとバックアップ オペレータ グループは、通常これらの権限を保有しています。

VOB またはビューの記憶ディレクトリをコピーしたら、273 ページの「保護問題を修復するためのユーティリティ」の説明に従って **fix\_prot -l** を実行し、ターゲット ディレクトリの ACL を確認します。**fix\_prot -l** によってエラーが表示された場合は、275 ページの「保護問題の修復」の手順に従って修正します。

## 保護問題の原因

ここでは、不正な記憶保護の問題を引き起こすおそれのある代表的なシナリオについて説明し、その状況を回避する方法を提示します。

### 記憶ディレクトリの不正なコピー

271 ページの「VOB または ビューの記憶ディレクトリをコピーするとき NTFS ACL を保持する」の説明に従って適切なコピーユーティリティを使用しないと、VOB またはビューの記憶ディレクトリをコピーするときに、コピー先で ACL が正しく保持されません。ディレクトリをコピーすることによって生ずる保護問題の修復については、275 ページの「保護問題の修復」を参照してください。

### ファイル システムの NTFS への変換

FAT または FAT32 としてフォーマットされたディスク パーティションで記憶ディレクトリを作成したあと、そのパーティションを NTFS に変換すると、記憶ディレクトリの保護が正しく行われません。変換後、ClearCase では VOB 所有者を別のユーザーと認識します。identity.sd ファイルが記憶ディレクトリのルートのセキュリティ記述子と合致しなくなるため、VOB とビューのサーバーは起動しません。この問題を回避する方法はありません。記憶ディレクトリ保護の修復については 275 ページの「保護問題の修復」を参照してください。

### 権限の編集

Windows エクスプローラや `cacls` を使用してファイル権限を編集すると、ClearCase の使用時にアクセスや保護の問題が発生します。

**注意:** 変更が VOB やビューの記憶領域に影響する場合は、Windows エクスプローラの [ファイルのアクセス権] ダイアログ ボックス内で [OK] をクリックしないでください。[OK] をクリックすると、ファイル権限を変更していない場合でも、アクセス コントロール エントリの順序が変更されます。

VOB 記憶ディレクトリの保護問題を検出するには、`checkvob` を次のように使用します。

```
cleartool checkvob -protections -pool vob-stg-pname
```

`vob-stg-pname` には、VOB 記憶ディレクトリのパス名を指定します。ファイル権限を編集したことにより発生した保護問題を修復するには、275 ページの「保護問題の修復」を参照してください。詳細については、`checkvob` リファレンス ページを参照してください。

## 保護問題を修復するためのユーティリティ

ClearCase には、VOB とビューの記憶ディレクトリの保護問題を検出して修復する 3 種類のユーティリティ プログラムが用意されています。

- **vob\_sidwalk** は、スキーマ バージョン 54 の VOB の記憶ディレクトリ保護を修復します。また、VOB オブジェクトの所有者を変更することもできます。詳細については、**vob\_sidwalk** リファレンス ページを参照してください。
- **fix\_prot** は、ビューとスキーマ バージョン 53 の VOB の記憶ディレクトリ保護を修復します。
- **lsacl** (Windows のみ) は、ファイル システム オブジェクトの NTFS ACL を表示します。

これら 3 種類のユーティリティは、**ccase-home-dir/etc/utlis** ディレクトリにインストールされます。

### fix\_prot

```
fix_prot [-force] { -root [-r-recurse] [-recover {-chown user | -chgrp group } |  
-replace _server_process_group| [-r-recurse] [-type { d | f }]  
[-chown user] [-chgrp group] [-chmod permissions] } pname ...
```

### オプション

**-force**

確認のメッセージを表示しません。

**-r-recurse**

保護を再帰的に修復します。

**-root**

**pname** が記憶ディレクトリ ルートであることを指定します。

**-type**

修復するファイル システム オブジェクトのタイプを指定します。ディレクトリには **d** を、ファイルには **f** を使用します。**-type** を指定しない場合、**fix\_prot** はディレクトリとファイルの両方で動作します。

**-chown**

新しい所有者を指定します。**-root** の権限が必要です

**-chgrp**

新しいプライマリ グループを指定します。**-root** の権限が必要です

**-chmod**

新しいアクセス権 (**owner**、**group**、**other** (グローバル)) を指定します。記号と数字の両コードが有効です。たとえば、**go-x** (記号) または **0644** (絶対値)などを指定できます。

-recover

(Windows のみ) identity.sd ファイルと groups.sd ファイルの情報に基づいて、VOB またはビューの記憶ディレクトリに正しいファイル システム ACL を復元します。スキーマ バージョン 54 VOB には使用できません。代わりに、vob\_sidwalk -recover\_filesystem がこの機能を実行します。

-replace\_server\_process\_group

(Windows のみ) ClearCase 管理者グループの ACL エントリを置き換えます。ClearCase 管理者グループの名前を変更したり、このグループの別の SID を持つ新しいドメインを表示するためにビューを移動した場合は、このオプションを指定します。

## 例

- VOB を Windows から UNIX に移動した後、UNIX .identity ディレクトリを作成するには、以下のコマンドを実行します。VOB は記憶ディレクトリ /vobstg/winvob.vbs に移動されています。新しい所有者は vobadm、新しいグループは ccusers です。root としてこのコマンドを実行します。

```
ccase-home-dir/etc/utils/fix_prot -root -recurse -chown vobadm -chgrp ccusers ¥  
/vobstg/winvob.vbs
```

- ビュー記憶ディレクトリを NTFS ディレクトリ C:¥ClearCaseStorage¥int.vws にコピーしたとき、ACL を保持するコピーユーティリティを使用しなかったために破損した ACL を修復するには、次のコマンドを実行します。

```
ccase-home-dir¥etc¥utils¥fix_prot -root -recover C:¥ClearCaseStorage¥int.vws
```

- ClearCase によってディレクトリ E:¥vobstg¥sources.vbs と関連付けられている保護モードを表示するには、次のコマンドを実行します。

```
ccase-home-dir¥etc¥utils¥fix_prot E:¥vobstg¥sources.vbs  
drwxr-xr-x MYDOMAIN¥me MYDOMAIN¥mygroup  
E:¥vobstg¥sources.vbs
```

**メモ:** fix\_prot から次のテキストで始まるエラー メッセージが返されることがあります。

```
fix_prot: エラー: directory の保護のスタイルが不明です: The data is invalid.
```

この場合、fix\_prot を返し、-chown、-chgrp、-chmod の各オプションをすべて指定します。また、絶対権限値を数値または ID=rights (o=rwx) の形式で -chmod に指定します。例を次に示します。

```
ccase-home-dir¥etc¥utils¥fix_prot -chown owner -chgrp group -chmod 0666 pname
```

## lsacl

```
lsacl [ -s | -l ] [ -n ] { [ -f path-name ] | [ -r registry-value-name ] }
```

## オプション

**-s | -l**

フォーマットがショートかロングかを指定します。デフォルトでは総称権を表示します。

**-n**

数値セキュリティ ID (SID) をユーザー名に変換しないように指定します。このオプションは、ドメイン コントローラが停止した場合やユーザー アカウントが削除された場合に使用します。

**-f**

ファイルからセキュリティ記述子を読み込みます。すなわち、**identity.sd** と **groups.sd** ファイルの内容を表示できるようになります。

**-r**

(Windows のみ) Windows レジストリ値からセキュリティ記述子を読み込みます。**registry-value-name** を **RootKey¥ValueName** の形式で指定します。

**RootKey** には HKLM、HKCU、HKCR、HKU、HKCC のいずれかを指定します。**RootKey** を省略すると HKLM とみなされます。

**ValueName** には指定した **RootKey** のサブキーのフルパスを指定します。

**メモ:** Windows の **cacls** ユーティリティを使用して NTFS ACL を表示しますが、**cacls** は、**\*.sd** ファイルからセキュリティ記述子を読み込むことができません。

## 保護問題の修復

ここでは、272 ページの「保護問題の原因」で説明した保護問題の修復方法について説明します。

保護問題を修復するには

- 1 管理者またはバックアップ オペレータのグループのメンバーとしてログオンします。
- 2 **groups.sd** ファイルが記憶ディレクトリのルート **stg-pname** にある場合、次のコマンドを実行します。  
**ccase-home-dir¥etc¥utils¥lsacl -f stg-pname¥groups.sd**

補足グループ リストに注意してください。次に出力例を示します。

```

===== stg-pname¥groups.sd
Owner: NT_WEST¥bob (User) (non-defaulted)          (所有者)
Group: NT_WEST¥usersnt (Group) (non-defaulted)      (プライマリ グループ)
ACL (revision 2):
0: allowed
SID: NT_WEST¥user (Group)                          (補足グループ)
rights (00000000)

1: allowed
SID: NT_WEST¥tester (Group)                        (補足グループ)
rights (00000000)

===== stg-pname¥groups.sd
Owner: NT_WEST¥bob (User) (non-defaulted)          (所有者)
Group: NT_WEST¥usersnt (Group) (non-defaulted)      (プライマリ グループ)
ACL (revision 2):
Empty ACL: all access denied                       (補足グループなし)

```

- 3 fix\_prot -root を実行して補足グループ リストを削除します。

```
ccase-home-dir¥etc¥utils¥fix_prot -r -root -chown owner -chgrp group ^
stg-pname
```

ビュー記憶領域を修復する場合は、これで完了です。

- 4 VOB 記憶領域を修復する場合は、VOB 所有者としてログオンし処理を続行します。
- 5 VOB に補足グループ リストがある場合は、次のコマンドを実行します。

```
cleartool protectvob -add_group group-name[...] vob-stg-pname
```

- 6 cleartext コンテナを削除するには、次のコマンドを実行します。

```
scrubber -e -k cltxt vob-stg-pname
```

checkvob では cleartext コンテナを修復できないため、この手順は手順 7 の前に行います。

- 7 記憶プールの保護を修復するには、次のコマンドを実行します。
- ```
cleartool checkvob -force -fix -protections -pool vob-stg-pname
```

## エレメントを再配置した後の既存のビューの問題

---

relocate (147 ページの「別の VOB へのエレメントの再配置」を参照) を実行すると、VOB の履歴のネームスペースを保護するシンボリック リンクが作成されますが、既存のビューやツールからエレメントにアクセスできるようにするため、エレメントを再配置した後でさらに調整を行う必要があります。

## クリーンアップのガイドライン

ファイルやディレクトリのエレメントを再配置した後にクリーンアップするには

- **再配置操作で使ったビューを使用します。**ビルド規則などを修正して構成仕様を調整した場合は、**relocate** を実行したときと同じビュー (または同じ構成仕様のビュー) を使用してください。

ソース VOB では、**relocate** は、再配置されたすべてのエレメントの親ディレクトリをチェックインすることによりその操作を完了します。このディレクトリのバージョンには、再配置されたエレメントに対するシンボリック リンクが含まれません。このため、このバージョンを選択するビューで作業すると、ツールやビルド規則などのプロセスが、再配置されたエレメントにアクセスできない場合に、どのプロセスで失敗するのかが分かります。

- **重要なビューと履歴表示をチェックして、シンボリック リンクが動作するかどうかを確認します。**278 ページの「再配置されたエレメントに対するシンボリック リンクの問題」で説明するように、VOB シンボリック リンク固有の制約により、それを参照するビューのリンクや構成仕様を変更する必要があります。次のような構成仕様の場合、問題が発生することがあります。
    - 再配置されたエレメントに対して明確なパス名規則のある構成仕様 (構成仕様は、ほかの VOB へのシンボリック リンクを辿りません)。
    - 主にラベル ベースの規則を使用している構成仕様。この場合、再配置されたエレメントや、ターゲット VOB 内のこれを含むディレクトリにラベルを追加するように選択できます。
    - **relocate** 操作を実行したブランチ以外のブランチで再配置されたエレメントを検索する構成仕様。**relocate** は、実行したビューで選択されたブランチ上のバージョンに対してのみシンボリック リンクを作成します。
  - **取り残されたビューを修復します。**問題のある構成仕様を検出したら、「再配置されたエレメントを参照するために古いターゲット ディレクトリ バージョンを修正する」で説明する方法のいずれかを使用して、再配置されたエレメントを再び表示できるようにします。
  - **取り残されたビュープライベート ファイルを修復します。**ソース VOB にアクセスするために使用する各ビューで **recoverview -sync** を実行します。取り残されたビュープライベート ファイルと DO が、ビューの **lost+found** ディレクトリに移動されます。
- メモ:** **cleartool lsprivate** を実行すると、取り残されたオブジェクトがオブジェクト ID (OID) 別に表示されます。

## 再配置されたエレメントに対するシンボリック リンクの問題

VOB シンボリック リンクは、VOB オブジェクトにリンクするための強力なメカニズムを提供します。ただし、いくつかの制約があるため、**relocate** では未解決のまま残された VOB シンボリック リンクを介してエレメントにアクセスするビューに影響を与えることがあります。

- 通常、**cleartool** コマンドは VOB シンボリック リンクを移動せず、リンク オブジェクト自体を操作します。たとえば、次のようになります。
  - VOB シンボリック リンクがエレメントを指している場合でも、シンボリック リンクをチェックアウトすることはできません。
  - **describe** コマンドは、リンクが指しているオブジェクトではなく、VOB シンボリック リンク オブジェクトについての情報を表示します。
  - **mklabel -recurse** コマンドはディレクトリ エレメントのサブツリー全体を移動しますが、遭遇する VOB シンボリック リンクをトラバースすることはありません。
- 構成仕様は、ほかの VOB へのシンボリック リンクを辿りません。
- ビルド スクリプトには、シンボリック リンクを見つけるとエラーを起こす操作が含まれているものがあります。
- 再配置されたエレメントを **cleartool mv** を使用して移動すると、ソース VOB からエレメントへのシンボリック リンクがすべて壊れます。
- 壊れたシンボリック リンク (ターゲットが存在しないリンク) の表示は、ほとんどの UNIX コマンドでできません。そのようなシンボリック リンクには、パス名情報が正しくても、現在のビューに選択されていないエレメントを指しているものが含まれます。**cleartool ls** コマンドを使用すると、このようなオブジェクトのリストが表示されます。

これらの制約に対処する方法がいくつかあります。

- ビューの構成仕様を変更して、新しい VOB の実際のパス名でエレメントを参照するようにします。これが最も完璧で予想しやすい方法です。しかし、多数のエレメントが再配置されている場合は実用的でなく、構成仕様に、多数のエレメントを個々に指定する規則が必要になります。
- 再配置されたバージョンにラベルを付加し、ラベル ベースの規則に基づいてエレメントを選択するビューを構成します。
- 再配置されたエレメントに特定のブランチ タイプを適用し、そのブランチを選択するビューを設定します。

## ディレクトリ バージョンの手動による更新

**relocate** 操作の結果を修正できます。それには、特定のディレクトリ バージョンに対する VOB シンボリック リンクを手動で追加して削除します。ただし、このような操作は、**relocate** の当初の結果を変更することになるため、通常は行いません。

**注意:** ここで説明する操作はとても効果的ですが、危険性が高いので注意してください。

**cleartool ln -nco** またはこれと同等のコマンド **rmname -nco** を不用意に使用しないでください。これらのコマンドは、イベント履歴を残さずに VOB を永久に変更します。これらのコマンドを使用するには、VOB 所有者か権限を持つユーザーである必要があります。レプリカ VOB では、これらのコマンドを使用できません。

### **relocate** で作成されたシンボリック リンクの修復

場合によっては、**relocate** で作成された VOB シンボリック リンクを、指定したターゲットを指すように修正する必要があります。特定のディレクトリ バージョンの不正なシンボリック リンクは、**cleartool rmname -nco** を使用して削除し、**cleartool ln -nco** を使用して置換できます。

**relocate** は、ソースとターゲットの VOB ルートの関係についてと、ソースとターゲット ディレクトリの関係について推測します。サイト固有の要素である NIS ホスト マップ、VOB のマウント、UNIX の命名規則などは、ドライブの割り当て、ディスクの共有、Windows の命名規則と同様、間違った推測をすることがあります。

シンボリック リンクのソースとターゲットをより簡単に識別するために、**relocate** は HyperSlink タイプのハイパーリンクを使用して、シンボリック リンク オブジェクトをターゲット エlement オブジェクトに接続します。シンボリック リンク上で **cleartool describe** コマンドを使用すると、このハイパーリンクに関する情報が表示されるので、必要に応じて、この情報を参照して修復します。

次のコマンドでは、**relocate** で作成された相対シンボリック リンクをターゲット VOB への代替絶対パス名で置換します。

```
cd %bigdir
```

```
cleartool rmname -nco %bigdir@@%main%2%dir1
```

```
非チェックアウト ディレクトリ バージョン "%bigdir@@%main%2%" を変更しますか? [no] yes
Link removed: "%bigdir@@%main%2%dir1"
```

```
cleartool ln -nco %new%proj2%dir1 %bigdir@@%main%2%dir1
```

```
非チェックアウト ディレクトリ バージョン "%bigdir@@%main%2%" を変更しますか? [no] yes
リンクが作成されました: "%bigdir@@%main%2%dir1"
```

## 再配置されたエレメントを参照するために古いターゲット ディレクトリ バージョンを修正する

relocate が移動先ディレクトリの新しいバージョンをチェックインする場合、そのバージョンのみが、再配置されたエレメントをカタログします。次のシナリオについて考えてみます。

- ターゲット ディレクトリの以前のバージョン (¥new@@¥main¥2 など) を選択するビューがある。
- このビューでは、古い場所 ¥bigdir¥dir1 ではなく、新しい場所 ¥new¥dir1 で新しく再配置された dir1 エレメントを参照する。

このような場合、移動先ディレクトリの以前のバージョンを手動で更新して、再配置されたエレメントをカタログできます。たとえば、次の手順で、ディレクトリ バージョン ¥new@@¥main¥2 に dir1 を追加します。

- 1 権限を持つユーザーとしてログインします。
- 2 VOB シンボリック リンクを作成します。以下のコマンドを実行するには特別な権限が必要です。ディレクトリ バージョンをチェックアウトせずに、またはイベント履歴を変更せずに、その内容を修正するためです。

```
cd ¥new
```

```
cleartool ln -slink -nco dir1 ¥new@@¥main¥2¥dir1
```

```
非チェックアウト ディレクトリ バージョン "¥new@@¥main¥2" を変更しますか？ [no] yes  
リンクが作成されました： "¥new@@¥main¥2¥dir1"
```

## 再配置されたエレメントを参照するためにソース ディレクトリの最新バージョンを修正する

relocate シナリオの一部では、再配置されたエレメントが、その親ディレクトリの最新バージョンにカタログされません。状況によっては、このようなカタログをシンボリック リンクの形式で手動で追加する場合があります。たとえば、次の一連のコマンドでは、再配置元ディレクトリ バージョン bigdir@@¥main¥3 を更新し、再配置されたディレクトリ dir1 を新しい場所で参照します。

```
cd ¥new
```

```
cleartool ln -slink -nco dir1 ¥bigdir@@¥main¥3¥dir1
```

```
非チェックアウト ディレクトリ バージョン "¥bigdir@@¥main¥3" を変更しますか？ [no] yes  
リンクが作成されました： "¥bigdir@@¥main¥3¥dir1"
```

## MVFS Data コンテナの検索

---

**注意:** ここでの説明は、ClearCase LT には当てはまりません。

ここではオペレーティング システムの標準ユーティリティと `mvfsstorage` コマンドを使用して、MVFS ファイルが物理的に格納されている場所を検索する方法について説明します。この記憶領域の検索は、ファイル アクセスに関する問題を動的ビューを使用して解決する場合に役立ちます。

### シナリオ

ここでは、ビュー `allison_vu` を介して参照できる、VOB ディレクトリ `¥monet¥src` 内の次の 3 つのファイルについて考えます。

- `cmd.c` はエレメント タイプが `text_file` のエレメントであり、現在はチェックアウトされています。
- `monet.icon` はエレメント タイプが `file` のエレメントであり、現在はチェックアウトされていません。
- `ralph_msg` はビュープライベート ファイルであり、電子メールのメッセージをディスクに保存して作成したものです。

### ファイルの ClearCase 状態の判定

`describe` コマンドを使用して、上記 3 つのファイルの状態を調べます。

```
cleartool describe cmd.c monet.icon ralph_msg
```

```
version "cmd.c@@¥main¥CHECKEDOUT" from ¥main¥6 (reserved)
  checked out 03-Feb-99.20:40:30 by (allison.mon@phobos)
  by view: allison_vu ("phobos:d:¥users¥people¥arb¥vw_store¥arb.vws")
  element type: text_file
```

```
version "monet.icon@@¥main¥1"
  created 03-Feb-99.20:17:04 by (allison.mon@phobos)
  element type: file
```

```
View private file "ralph_msg"
```

```
Modified: Wednesday 02/03/99 21:39:49
```

## ファイルのフルパス名の判定

オペレーティング システムの標準コマンド (UNIX の `ls` と `pwd`。 `dir`、 `attrib`、 Windows エクスプローラ) を使用して、ファイルのフルパス名を表示します。

```
ls -l `pwd`/cmd.c
-rw-rw-r--  1 allison  mon    211 Feb  2 12:03 /proj/monet/src/cmd.c
attrib cmd.c
C:\proj\monet\src\cmd.c
```

## VOB の位置

`describe` コマンドを使用して、ネットワーク上の VOB root へのパスと、VOB 記憶領域が保存されているホストへの相対パスを表示します。

```
cleartool describe vob:%monet%src
versioned object base "%monet"
  created 01-Feb-93.17:35:03 by (vobadm.vobadm@sol)
  VOB 記憶ホスト: パス名 "sol:c:%vbstore%monet.vbs"
  VOB 記憶グローバル パス名 "%sol_vbstore%monet.vbs"
  VOB の所有権:
    owner vobadm
    group vobadm
```

また、ClearCase 管理コンソールの [すべての VOB] ノードからこの情報を入手して、ローカル ホストのレジストリ サーバーにあるこのローカル ホストのリージョンに登録されているすべての VOB タグをリストすることもできます。タスクパッドビューと詳細ビューを使用すると、各 VOB の情報と共に VOB 記憶ディレクトリのホストとグローバル パスを表示できます。このリストから ClearCase 管理コンソール にある特定の VOB の VOB 記憶ノードに移動し、そこで VOB 記憶領域を管理できます。

## ビューの位置

`pwv` コマンドと `lsview` コマンドを使用して、ビュー記憶領域の場所を表示します。

```
cleartool pwv
作業ディレクトリ ビュー: allison_vu
現在設定されているビュー: allison_vu
cleartool lsview allison_vu
* allison_vu %phobos_users%people%arb%vw_store%arb.vws
```

**注意:** UNIX コンピュータでは、ビュー記憶領域が存在しているホストを確認するときに、`mount` コマンドを使用する必要があります。

ClearCase 管理コンソールの[すべての VOB] ノードは、ローカル ホストのレジストリ サーバーにあるこのローカル ホストのリージョンに登録されている、すべてのビュー タグをリストします。タスクパッド ビューと詳細ビューを使用すると、各ビューの情報と共にビュー記憶ディレクトリのホストとグローバル パスを表示できます。このリストから ClearCase 管理コンソールにある特定のビューのビュー記憶ノードに移動し、そこで VOB 記憶領域を管理できます。

## 個々のファイルの位置

すべての MVFS ファイルのデータ コンテナは、表 10 に示すように、VOB またはビューの記憶ディレクトリに格納されています。

**表 10 MVFS ファイルの記憶ディレクトリ**

| ファイルの種類             | 記憶ディレクトリ                                               |
|---------------------|--------------------------------------------------------|
| チェックイン バージョン        | VOB ソース記憶プール<br>(VOB 記憶ディレクトリの <b>s</b> サブディレクトリ)      |
| チェックアウト バージョン       | ビュープライベート データ記憶領域<br>(ビュー記憶ディレクトリの <b>.s</b> サブディレクトリ) |
| 非共有または共有不可の派生オブジェクト | ビュープライベート データ記憶領域<br>(ビュー記憶ディレクトリの <b>.s</b> サブディレクトリ) |
| 共有派生オブジェクト          | VOB 派生オブジェクト記憶プール<br>(VOB 記憶ディレクトリの <b>d</b> サブディレクトリ) |
| ビュープライベート ファイル      | ビュープライベート データ記憶領域<br>(ビュー記憶ディレクトリの <b>.s</b> サブディレクトリ) |

次の項では、ccase-home-dir/etc/mvfsstorage ユーティリティを使用して MVFS ファイルの物理的な記憶領域を検索する方法について説明します。

## チェックアウト バージョンの位置の確認

mvfsstorage を使用して、text\_file エレメント cmd.c のチェックアウト バージョンの、ビュープライベート データ記憶領域内の位置を調べます。

mvfsstorage cmd.c

```
¥¥phobos¥vw_store¥arb.vws¥.s¥00050¥8000000B.00B0.cmd.c
```

## チェックイン バージョンの cleartext コンテナの場所の確認

すべてのバージョンを格納するために1つのデータ コンテナを使用するエレメントのチェックイン バージョンについては、**mvfsstorage** を使用して、タイプ マネージャがバージョンを保存する **cleartext** データ コンテナの場所を確認します。

```
mvfsstorage cmd.c@@@¥main¥1
¥¥sol¥vobstore¥monet.vbs¥c¥cdft¥28¥32¥8a1a9a50010e11cca2ca080069021822

mvfsstorage cmd.c@@@¥main¥2
¥¥sol¥vobstore¥monet.vbs¥c¥cdft¥3a¥33¥8e4a9a54010e11cca2ca080069021822
```

## チェックイン バージョンのソース コンテナの場所の確認

各バージョンごとに別のデータ コンテナを使用するエレメントのチェックイン バージョンについては、**mvfsstorage** を使用して、ソース プール内のデータ コンテナの位置を調べます。

```
mvfsstorage monet.icon
¥¥sol¥vobstore¥monet.vbs¥s¥sdft¥26¥4¥474fa2f4021e11cca42f0800690605d8
```

各バージョンを別々のコンテナに格納するエレメント タイプでは、**cleartext** プールを使用しません。その代わりに、プログラムはソース プールのデータ コンテナにアクセスします。

## ビュープライベート ファイルの位置の確認

チェックアウト バージョンと同様に、ビュープライベート ファイルもビューのプライベート データ記憶領域にあります。

```
mvfsstorage ralph_msg
¥¥sol¥view_store¥arb.vws¥.s¥00050¥8000000C.00BD.ralph_msg
```

## UNIX のリモート記憶プールの問題

UNIX コンピュータでは、VOB 記憶プールが VOB 記憶ディレクトリの外にある場合や、ビューのプライベート記憶領域がビュー記憶ディレクトリの外にあることがあります。

**mvfsstorage** の出力から、データ コンテナがデフォルトの VOB 記憶プール以外に存在することが判明した場合は、**lspool** コマンドを使用して記憶プールの場所を調べます。デフォルトの記憶プールとは **sdft** (デフォルトのソース プール)、**cdft** (デフォルトの **cleartext** プール)、**ddft** (デフォルトの派生オブジェクト プール) です。例を次に示します。

```
ccase-home-dir/etc/mvfsstorage hello.h
/vobstore/monet.vbs/c/clrtxt.1/36/f/6b6ed22b08da11cca0ef0800690605d8

cleartool lspool -l clrtxt.1
pool "clrtxt.1"
  08-Feb-93.10:25:46 by (vobadm.vobadm@starfield)
  "nonlocal cleartext storage for monet VOB"
```

```
kind: cleartext pool
pool storage remote host:path
"sol:/netwide/public/ccase_pools/clrtxt.1"
pool storage local pathname "/vobstore/monet.vbs/c/clrtxt.1"
maximum size: 0 reclaim size: 0 age: 96
```

clrtxt.1 はデフォルトの **cleartext** プールではありません。

UNIX の **ls** コマンドを使用して、ビューのプライベート記憶領域 (**.s** サブディレクトリ) がローカルかどうかを判定します。

```
ls -ld ~jones/view_store/temp.vws/.s
lrwxrwxr-x  1 jones   dvt           34 Feb 17 17:06
/usr/people/jones/view_store/temp.vws/.s -> /public/jones_temp
```

## UNIX でのリンクとディレクトリ

UNIX コンピュータでは、**mvfsstorage** は VOB と、ファイル システムのリンクとディレクトリ オブジェクトを次のように扱います。

- リンクに関して、**mvfsstorage** はリンクが指すオブジェクトの記憶ディレクトリを示します。これは、すべてのリンク、つまり、ビュープライベートのハードリンクとシンボリック リンク、VOB のハードリンク、VOB のシンボリック リンクに当てはまります。
- ビュー プライベートディレクトリには、データ コンテナも、ディレクトリ エレメントもあります。どちらの場合も、**mvfsstorage** はディレクトリ パス名を表示します。

## UNIX ビューのルート ディレクトリの再帰探索の防止

---

**注意:** ここでの説明は、ClearCase LT には当てはまりません。

MVFS がインストールされている UNIX ホストでは、**/view** ディレクトリは、MVFS ネームスペースのマウント ポイントとして機能します。この **/view** ディレクトリによって、ルート ディレクトリが再帰的に含まれることになるため、システムのルートからディレクトリ ツリー全体を走査するコマンドは、永久ループします。多くの UNIX システムでは、**cron** スクリプトを毎日実行し、ファイル システムをクリーンアップします。このスクリプトで UNIX の **find** / コマンドを使用している場合は、永久ループの危険性があります。

**メモ:** UNIX ファイル システム全体で機能するコマンド (**/** で始まる) は、このようにしてループする可能性があります。これらのループは、コマンド行、スクリプト、GUI のどれで実行したかにかかわらず、また、**root** ユーザーであっても別のユーザーであっても発生します。

ClearCase インストール スクリプトでは、**root** ユーザーの **crontab** ファイルを分析し、このファイルのエントリを変更して再帰探索を防止します。この変更を完了できない場合は、警告メッセージが表示されます。

インストール後、変更が正しく行われたことを確認して、必要に応じてほかのユーザーの **crontab** エントリを変更し、システム ルートで始まるファイル システム全体で機能するコマンドをすべて削除します。プラットフォーム固有の詳細情報は、オンラインの『**Rational ClearCase** プラットフォーム別情報ガイド』を参照してください。また、再帰検索問題を引き起こすような項目を付加しないように、**crontab** ファイルに警告コメントを追加することをお勧めします。

# クライアント ホストの パフォーマンスの改善

# 15

この章では、クライアント ホスト上で **ClearCase** のパフォーマンスを改善する方法について説明します。クライアント ホストのパフォーマンスに関係する管理責任には次のものがあります。

- クライアント ホストの構成 (メモリ、CPU 処理能力、ネットワーク インターフェイスの特性、ローカル記憶領域) の標準確立
- MVFS キャッシュ サイズとその調整方法の理解
- ビュー キャッシュとその調整方法の理解
- キャッシュ サイズのサイト全域のデフォルト確立

**メモ:** この章で説明するパフォーマンス調整処理のほとんどは、MVFS に対するものです。これらの処理は **ClearCase LT** クライアントや、スナップショットまたは **Web** ビューのみで動作する **ClearCase** クライアントには適用されません。

## クライアント ホストの構成のガイドライン

---

**ClearCase** クライアント ソフトウェアは、類似するワークステーション アプリケーションの要求を満たすホスト メモリ、CPU、記憶領域サブシステムを必要とします。そのようなアプリケーションをサポートするように構成されているコンピュータでは、大部分の **ClearCase** クライアント タスクで良好なパフォーマンスを得られます。

**Rational ClearCase** は分散型アプリケーションなので、クライアント ホストのネットワーク インターフェイスや接続されているネットワークにも良好なパフォーマンスが必要です。ネットワークのパフォーマンスが悪いと、**ClearCase** クライアント ホストの構成が適切でもレスポンスは低下します。

メモリを追加するとクライアント ホストのパフォーマンスが向上する場合があります。たとえば、分散ビルドや共有ビューのホストなどの特別な処理を行うクライアント ホストでは、メモリを追加したり、場合によってはより大容量の高性能ディスク サブシステムを追加する必要があります。**ClearCase** の実行中にほかのワークステーション アプリケーションを実行するクライアントは、両方を同時に使用できるようにリソースを適切に調整する必要があります。

## MVFS キャッシュ サイズの調査と調整

---

動的ビューを使用する ClearCase ホストでは、MVFS キャッシュを調整するとパフォーマンスが改善されることがあります。MVFS のデフォルト キャッシュ サイズは、ホストのメモリに基づいて自動的に調整され、さまざまなクライアントの要求に適したものになりますが、特定の状況で MVFS のデフォルト キャッシュ サイズを変更してパフォーマンスを向上させることもできます。

**メモ:** キャッシュを調整する場合は常に、キャッシュを大きくすることによるメリットと、より多くの物理メモリをキャッシュに割り当てることでアプリケーションのパフォーマンスが低下する可能性との間で、慎重な加減が必要になります。キャッシュの調整により、特定のホストの利用パターンを反映した形で MVFS による空きメモリ使用の再割り当てを促すことができますが、物理メモリが十分でないときの補填はできなくなります。

MVFS キャッシュ サイズと使用状況を調べるには、`cleartool getcache -mvfs` コマンドを使用します。MVFS キャッシュ サイズを変更するには、次の 2 つの方法があります。

- 最も簡単で、推奨される方法は、MVFS スケーリング係数を調整することです。MVFS スケーリング係数は、その他のほとんどの MVFS キャッシュ サイズから導出される整数値です。UNIX や Windows 上で、`cleartool setcache -mvfs -persistent -scalefactor` を使用してスケーリング係数を調整します (スケーリング係数を設定するときは `-persistent` オプションを使用する必要があります)。Windows 上では、コントロール パネルの [ClearCase] も使用できます。[MVFS のパフォーマンス] タブでは、スケーリング係数とほかのいくつかのキャッシュ サイズを設定できます。
- `setcache -mvfs` またはコントロール パネルの [ClearCase] の [MVFS のパフォーマンス] タブを使用することで、個々のキャッシュに選択した値を調整することもできます。この方法では、広範な分析とテストが必要になりますが、より詳細に制御できます。ただし、正しく実行しないと、パフォーマンスが実際に低下することがあります。この方法は、スケーリング係数から導出される値が適していない非日常的な状況がない限りお勧めしません。

**メモ:** `setcache -mvfs` で行った変更は通常、一時的なものであり、MVFS が再起動されるとデフォルト値に戻ります。`setcache` には、スケーリング係数または個々のキャッシュに対して行った変更を再起動後も保持できる `-persistent` オプションがあります。[MVFS のパフォーマンス] タブで行った変更は常に保持されます。

## MVFS キャッシュの統計情報の調査

`cleartool getcache` コマンドを使用して、ホストの MVFS キャッシュの統計情報を表示します。`getcache -mvfs` はファイル/ディレクトリ ハンドル キャッシュ、属性キャッシュに関する情報を表示します。ファイル/ディレクトリ ハンドル キャッシュの統計情報は、`current-number-on-list/list-capacity` と `(percentage-of-list-capacity)` の形式で表示されます。

属性キャッシュの統計情報は、各キャッシュのサイズと、MVFS の前回の再起動からカウントされたキャッシュの失敗の総数を示します。例を次に示します。

#### cleartool getcache -mvfs

```
Mnode: (アクティブ/最大) 1791/8192 (21.863%)
Mnode フリーリスト: 1701/1800 (94.500%)
Cltxt フリーリスト: 737/1800 (40.944%)
```

```
DNC:   ファイル: 848/1600 (53.000%)
        ディレクトリ: 185/400 (46.250%)
        ENOENT: 827/1600 (51.688%)
```

```
RPC ハンドル: 4/10 (40.000%)
```

現在の MVFS キャッシュ有効化の設定:

```
属性キャッシュ: enabled
閉じるから開くまでの作業の再確認: enabled
名前キャッシュ: enabled
Readlink キャッシュ: enabled
Root バージョン キャッシュ: enabled
VOB フリーリストのウォーター マーク低:1620
Cleartext フリーリストのウォーター マーク低:1768
Readdir ブロック キャッシュ サイズ: 4
MVFS スケーリング係数: 1
```

属性キャッシュに集計がありません (調整方法については、ClearCase の管理マニュアルを参照してください):

```
属性キャッシュの失敗の合計: 49609 (100.00%)
ファイル (ビュー プライベート) を
閉じてから開くまでの失敗: 18964 ( 38.23%)
生成 (並行ビルド) の失敗: 1179 ( 2.38%)
キャッシュ タイムアウトの失敗: 234 ( 0.47%)
キャッシュ埋め (新規ファイル) 失敗: 0 ( 0.00%)
イベント タイム (VOB/ビュー mod) 失敗: 29232 ( 58.92%)
```

**メモ:** キャッシュのサイズを実際に変更する前に、mvfsstat (301 ページの「mvfsstat」を参照) と mvfstime の 2 つのコマンドを使用して、キャッシュの有効性を確認することをお勧めします。

## MVFS スケーリング係数の調整

デフォルトの MVFS スケーリング係数は、ホストに取り付けられた使用可能な物理メモリに基づいて、表 11 に示すように自動的に設定されます。

表 11 メモリ サイズと MVFS スケーリング係数の関係

| メモリ サイズ (MB) | 24 より小 | 24 ~ 512 | 512 ~ 2048 | 2048 より大  |
|--------------|--------|----------|------------|-----------|
| スケーリング係数     | 0      | 1        | 2          | size/1024 |

表 11 で、size の値は、オペレーティング システムのカーネルによって要求されるメモリが除外され、size 値が 2048 MB を超える場合は、1024 MB に最も近い倍数まで丸められます。たとえば、ホストに 4096 MB (4 GB) の物理メモリがあり、カーネルに 100 MB が費やされる場合、スケーリング係数は 3 に設定されます。

setcache -mvfs -persistent -scalefactor を使用して、デフォルトのスケーリング係数の設定を上書きできます。Windows 上では、コントロール パネルの [ClearCase] も使用できます。[MVFS のパフォーマンス] タブの [パフォーマンス向上のためにさらに多くのメモリを使用して MVFS を初期化するスケーリング係数] ボックスに値を設定します。[OK] をクリックします。

スケーリング係数を 1 大きくすると、約 500 KB のカーネルのメモリ (うち 200 KB はページング不可、300 KB はページング可) が追加で必要になります。

**メモ:** setcache を使用して現在使用可能なメモリより多くのメモリを要求すると、その要求は失敗し、スペース不足を示すエラーが表示されます。このようなエラーが表示された場合は、より小さなキャッシュ サイズを要求するか、カーネルのメモリを費やしている不要なプロセスを終了させてください。

スケーリング係数の値を変更すると、表 12 に示すように、MVFS キャッシュ サイズはそれに比例して変更されます。

**表 12 MVFS スケーリング係数と個々の MVFS キャッシュ サイズの関係**

| MVFS キャッシュ名                      | スケーリング<br>係数 = 0 | スケーリング<br>係数 = 1 | スケーリング<br>係数 = 2 | スケーリング係数<br>= f (f > 2)               |
|----------------------------------|------------------|------------------|------------------|---------------------------------------|
| DNC ファイルの<br>キャッシュ               | 800 バイト          | 1600 バイト         | 2400 バイト         | 800(f+1) バイト                          |
| DNC ディレクトリの<br>キャッシュ             | 200 バイト          | 400 バイト          | 600 バイト          | 200(f+1) バイト                          |
| DNC ENOENT の<br>キャッシュ            | 800              | 1600             | 2400             | 800(f+1)                              |
| cleartext フリーリスト<br>最大 (UNIX)    | 900              | 1800             | 2700             | 900(f+1)                              |
| cleartext フリーリスト<br>最大 (Windows) | 900              | 1800             | 1800             | 1800                                  |
| RPC ハンドルの<br>キャッシュ               | 5                | 10               | 15               | f ≤ 2 の場合、5(f+1)<br>f > 2 の場合、10(f+1) |

表 12 MVFS スケーリング係数と個々の MVFS キャッシュ サイズの関係

| MVFS キャッシュ名           | スケーリング<br>係数 = 0 | スケーリング<br>係数 = 1 | スケーリング<br>係数 = 2 | スケーリング係数<br>= $f$ ( $f > 2$ )                           |
|-----------------------|------------------|------------------|------------------|---------------------------------------------------------|
| mnode フリーリスト<br>最大    | 901              | 1802             | 2703             | $0.22 f$                                                |
| mnode キャッシュ           | 4096             | 8192             | 12,288           | $f \leq 4$ の場合、 $4096(f+1)$<br>$f > 4$ の場合、 $2048(f+7)$ |
| readdir ブロック<br>キャッシュ | 2                | 4                | 6                | $f < 2$ の場合、 $2(f+1)$<br>$f \geq 2$ の場合、6               |

## 個々のキャッシュ パラメータの設定

スケーリング係数を使用して MVFS キャッシュを調整することをお勧めしますが、個々のキャッシュのサイズを変更することもできます。この方法を選択する前に、いくつかの考慮事項があります。

**メモ:** キャッシュを調整する場合は常に、全体のパフォーマンスに悪い影響を与えることなく、頻繁に参照されるオブジェクトのキャッシュを最適化する慎重な加減が必要になります。MVFS キャッシュの調整を特定のアプリケーションに対して効果的になるように行う場合は、広範な分析、テストと、アプリケーションが MVFS を使用方法に関する詳細な知識が必要になります。

- 個々のキャッシュ パラメータを変更すると、MVFS のパフォーマンスに悪い影響を与える可能性があります。
- スケーリング係数を使用してキャッシュ サイズを変更すると、キャッシュの調整が大幅に簡素化され、多くの状況で効果を発揮します。
- マルチプロセッサ システムでは、キャッシュは、場合によってロックの必要が発生します。キャッシュを大きくすると、ロックのオーバーヘッドも大きくなります。オーバーヘッドはパフォーマンスに影響を与えます。

キャッシュ サイズは段階的に調節し、`cleartool getcache -mvfs` でキャッシュ使用状況を確認してください。ホスト上の MVFS キャッシュ構成を変更する場合は、そのホスト上の各ビューを再設定してそのキャッシュ サイズを大きくすることも検討してください。298 ページの「ビューの再設定」を参照してください。

表 13 に、調整可能なキャッシュ パラメータに関する説明と、設定するサイズの推奨値を示します。また、`getcache` の出力に含まれるフィールドと、`setcache` のさまざまなオプションの関係についても示します。

表 13 MVFS キャッシュ情報(1/2)

| getcache<br>出力の<br>フィールド名             | 説明                                                                                                                                                                                                                 | 調整方法              |                                        |
|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|----------------------------------------|
|                                       |                                                                                                                                                                                                                    | setcache<br>オプション | コントロール<br>パネルの値<br>(Windows)           |
| Mnode<br>フリーリスト                       | <p>mnode は、ファイルやディレクトリを表すために MVFS によって使用されるデータ構造です。スケーリング係数は、MVFS が起動されてから割り当てられた mnode の合計数を決定します。</p> <p>VOB フリー リスト上の mnode は、最近使用されたが、現在は休止状態にあります。多数の mnode をフリー リストに保持することにより、関連ファイルを再度開くのに必要な時間が短縮されます。</p> | -vobfreemax       | VOB フリー リストに保持する mnode 最大数             |
| VOB フリー<br>リストの<br>ウォーター<br>マーク低      | VOB フリー リストに保持する mnode 最少数。VOB フリー リスト上の mnode の数が指定された最大値に達した場合、そのリスト上の mnode の数はこの値まで減らされます。                                                                                                                     | -vobfreemin       | VOB フリー リストに保持する mnode 最少数             |
| Cltxt<br>フリーリスト                       | cleartext フリー リストは、記憶プールのファイルオブジェクトへのポインタのコレクションです。このポインタをキャッシュすることにより、MVFS 常駐ファイルの再オープンを高速化できます。ただし、リソースを余分に消費します (Windows コンピュータでは、記憶領域がリモートコンピュータ上にある場合の、ネットワーク接続上に開かれているファイル記述子などのリソース)。                       | -cvpfreemax       | cleartext フリー<br>リストに保持する<br>mnode 最大数 |
| Cleartext<br>フリーリストの<br>ウォーター<br>マーク低 | cleartext フリー リストに保持する mnode 最少数。cleartext フリー リスト上の mnode の数が指定された最大値に達した場合、そのリスト上の mnode の数はこの値まで減らされます。                                                                                                         | -cvpfreemin       | cleartext フリー<br>リストに保持する<br>mnode 最少数 |

表 13 MVFS キャッシュ情報(2/2)

| getcache<br>出力の<br>フィールド名    | 説明                                                                                                                                                                                                                                                                                                                                      | 調整方法              |                                      |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|--------------------------------------|
|                              |                                                                                                                                                                                                                                                                                                                                         | setcache<br>オプション | コントロール<br>パネルの値<br>(Windows)         |
| DNC ファイル                     | DNC ファイルと DNC ディレクトリは、最近アクセスしたファイルとディレクトリへのポインタを保持します。ENOENT キャッシュは、最近ルックアップに失敗したファイルとディレクトリの名前を保持します。ファイル名またはディレクトリ名が渡されると、MVFS はまずこれらのキャッシュを探します。一致する項目が見つからなかった場合、MVFS はビュー サーバーに対して RPC を実行することが必要になります。各キャッシュ項目の大きさは、約 100 バイトです。ディレクトリ キャッシュは通常、パフォーマンスに対して最大の影響を与えます。                                                            | -regdnc           | ファイルに対するエントリの最大数                     |
| DNC ディレクトリ                   |                                                                                                                                                                                                                                                                                                                                         | -dirdnc           | ディレクトリに対するエントリの最大数                   |
| DNC ENOENT                   |                                                                                                                                                                                                                                                                                                                                         | -noentdnc         | 存在しない名前に対するエントリの最大数                  |
| RPC ハンドル                     | MVFS からビュー サーバーへの進行中の RPC ごとに 1 つの RPC ハンドルが使用されます。多数のプロセスが同時に進行する大きなビルドを実行するときにキャッシュが 100% 使用される場合は、このキャッシュ サイズを増やして RPC の実行時間を短縮する必要があります。いったん RPC ハンドルをこのキャッシュに追加すると、MVFS が再起動されるまで、そのハンドルは使用可能のまま残されます。使用可能な RPC ハンドルがキャッシュになければ、要求に応じて新しいハンドルが作成されます。キャッシュがいっぱいでなければ、RPC が完了したときに新しい RPC ハンドルがキャッシュに返されます。いっぱいになると、RPC 完了時に破棄されます。 | -rpchandles       | RPC ハンドル最大数                          |
| Readdir ブロック<br>キャッシュ<br>サイズ | readdir ブロック キャッシュはディレクトリの内容をキャッシュするために使用されます。非常に大きなディレクトリを定期的に操作する場合は、このキャッシュのサイズを大きくすることができます。                                                                                                                                                                                                                                        | -readdir_blocks   | 各ディレクトリの<br>Readdir キャッシュ<br>ブロック最大数 |

## 属性キャッシュの失敗の最小化

MVFS は、ファイル オブジェクトとディレクトリ オブジェクトの属性をキャッシュして VOB をルックアップする必要性を回避します。属性キャッシュがいつばいでも、ヒット率が高ければ、必ずしも効果がないとはいえません。ヒット率は 85% 以上を維持することをお勧めします。以降の項では、キャッシュの失敗のカテゴリと、失敗回数の低減方法を説明します。

## 属性キャッシュの失敗の合計

失敗の合計とは、すべての失敗項目を加算したものです。この合計は、後に示す各カテゴリの失敗項目の値を減らすことによって低減できます。

## 閉じてから開くまでの失敗

閉じてから開くまでの失敗は、開いているビュープライベート ファイルが別のプロセスによって再度開かれると発生します。MVFS は、ビュープライベート ファイルが再度開かれたとき、`view_server` と照合してチェックし、ファイルがほかの MVFS クライアントによって修正されている場合にはキャッシュされたファイル属性をリフレッシュします。

この機能はホストごとに (ビューごとではありません) 停止することができます。ほかのホストにあるビューで、ホストがビュープライベート ファイルにアクセスしないことがはっきりしている場合は、そのホストに対して、閉じてから開くまでの整合性チェックを無効にしてこのようなキャッシュの失敗を少なくすることができます。

閉じてから開くまでの整合性チェックを無効にするには、`setcache -mvfs -ncto` を使用します。それを再度有効にするには、`setcache -mvfs -cto` を使用します (変更を再起動後も保持するには、`-persistent` オプションを使用します)。

## 生成の失敗

生成の失敗は、並行ビルドを実行する場合に発生します。並行ビルドを行う場合、これらの失敗は回避できません。

## キャッシュ タイムアウトの失敗

キャッシュ タイムアウトの失敗は、ファイルの属性がしばらくチェックされていないときに発生します。MVFS は、定期的にファイルの属性を再検証し、古くなったデータをキャッシュ保存しないようにします。

これらの失敗を完全になくすことは不可能です。しかし属性のタイムアウト時間は調整することができます。キャッシュのタイムアウトは、ファイルまたはディレクトリを最後に修正した時期によって変化します。つまり、修正時期が新しいほどタイムアウト時間は短くなります。このタイムアウト時間の初期値は、属性キャッシュ寿命の最小値と最大値の間になるように制限されています。この値は、VOB をマウントしたときにパラメータ `acregmin/acregmax` (通常のファイルの場合) と `acdirmin/acdirmax` (ディレクトリの場合) によって指定されます。デフォルト値は次のとおりです。

- `acregmin`: 3 秒
- `acregmax`: 60 秒
- `acdirmin`: 30 秒
- `acdirmax`: 60 秒

これらの値を変更するには、次のいずれかの方法で VOB の `mount` オプションを指定します。

- マウント時。例を次に示します。

```
cleartool mount -options acregmin=30 vob-tag
```

この操作を行うには、権限を持つユーザーである必要があります。

- レジストリ内で行う (VOB 作成時に `mkvob` を使用するか、後で `mktag` を使用します)。例を次に示します。

```
cleartool mktag -vob -tag vob-tag -replace -options mount-options ...
```

**メモ:** これらの `mount` オプションで指定されたタイムアウト値は、ビューのメタデータ待ち時間 (VOB メタデータへの変更が、変更が行われた時点で見えている VOB メタデータではなく、動的ビューで見えるようになるまでの時間) に影響します。タイムアウト値を長くすると、待ち時間は長くなりますが、パフォーマンスは向上します。タイムアウト値を短くすると、待ち時間は短くなりますが、キャッシュをより頻繁にリフレッシュすることが必要になるため、ビューのパフォーマンスにも影響します。

## キャッシュ埋めの失敗

キャッシュ埋めの失敗は、ファイルの属性がキャッシュに存在していないときに発生します。この割合が高い場合 (20% を超える場合)、作業セットに対してキャッシュが小さすぎる可能性があります。`setcache -mvfs -vobfreemax` を実行して、フリー リスト上の `mnode` の数を増やすことを検討してください。

## イベント タイムの失敗

イベント タイムの失敗は、キャッシュされている名前からオブジェクトへの変換による結果のオブジェクトの属性を再評価する場合に発生します (たとえば、名前キャッシュに、ファイルオブジェクトに対応するファイル名があっても、属性キャッシュを、タイムアウトの失敗や並行ビルドで発生した生成の失敗のために、リフレッシュする必要がある場合)。

これらの失敗を完全になくすことは不可能ですが、キャッシュ タイムアウトの失敗の場合と同様、最少/最大のキャッシュ寿命を調整することにより低減させることは可能です。

## ビュー キャッシュ

---

パフォーマンスを向上させるために、ビューに対応する **view\_server** プロセスはキャッシュを維持管理します。このキャッシュのデフォルト サイズは通常のビューには十分な大きさですが、キャッシュのサイズをより大きくすることで、**omake** または **clearmake** によって非常に大きいソフトウェア システムがビルドされるビューのパフォーマンスをさらに向上させることができます。ビュー サーバーは次のキャッシュを維持管理します。これらのキャッシュの内容は、主に **VOB** のデータで構成されています。これらのキャッシュを使用すると、クライアントが発行した **RPC** にすばやく応答できます。

- オブジェクト キャッシュ。バージョン、ブランチなどのオブジェクトの取得を容易にします。
- ディレクトリ キャッシュ。ファイル システムのディレクトリ読み取りアクセスを容易にします。
- **stat** キャッシュ。ファイル属性を格納します。
- 名前キャッシュ。名前を格納し、名前ルックアップを高速化します。

**view\_server** プロセスは、開始時にキャッシュ サイズを選択します。具体的には、次の値を順に調べて、最初に取得できた値を使用します。

- ビューが作成される時 (**cleartool mkview -cachesize**)、または **cleartool setcache -view -cachesize** コマンドで修正されるときに設定された、ビューごとの永続値
- **setcache -view -host** で設定された、ホストごとの永続値
- **setcache -view -site** で設定された、サイト全体のデフォルト
- デフォルト値。32 ビット プラットフォームの場合は 512 KB、64 ビット プラットフォームの場合は 1 MB です。

ビュー キャッシュ サイズは、ビューの各キャッシュで割り当てに使用可能なバイト数の合計を表します。

ビューのキャッシュ サイズとサイトのキャッシュ サイズの設定については、**setcache** リファレンス ページを参照してください。キャッシュ情報の表示については、**getcache** リファレンス ページを参照してください。

## ビュー キャッシュ情報の取得

ビュー キャッシュ情報を入手するには、`cleartool getcache -view` を使用します。ビュー サーバーはビュー キャッシュ サイズとそのヒット率に関する情報を出力します。例を次に示します。

`cleartool getcache -view r5_integration`

```
ルックアップ キャッシュ： 29% フル、 1121 項目 ( 56.8K)、 15832 要求、 75% ヒット
Readdir キャッシュ： 4% フル、 24 項目 ( 36.5K)、 4159 要求、 83% ヒット
Fstat キャッシュ： 31% フル、 281 項目 (105.1K)、 55164 要求、 100% ヒット
オブジェクト キャッシュ： 26% フル、 1281 項目 (176.6K)、 40626 要求、 72% ヒット
ビュー キャッシュに使用するメモリの合計： 375.0 キロバイト
```

現在のビュー サーバー キャッシュの制限：

```
ルックアップ キャッシュ： 201312 バイト
Readdir キャッシュ： 838860 バイト
Fstat キャッシュ： 352296 バイト
オブジェクト キャッシュ： 704592 バイト
合計キャッシュ サイズの上限： 2097152 バイト
```

この情報は、次のことに利用できます。

- ビュー キャッシュを増やすべきかどうかを判断する
- ビュー キャッシュの変更後に結果をチェックする
- ほかのビュー サーバー プロセスを分析する

ビュー キャッシュには、次の種類のサブキャッシュがあります。

- **ルックアップ**: ビュー内のオブジェクトへの名前の割り当てを高速化するために使用するデータを格納します。
- **Readdir**: ディレクトリの読み取り操作 (たとえば、`ls` または `dir`) を高速化するために、ビューが使用するデータを格納します。
- **Fstat**: ビューが使用するファイル属性に関するデータを格納します。
- **オブジェクト**: ビューが使用するオブジェクトに関するデータを格納します。

ビュー キャッシュの合計サイズを算出するには、上記の各コンポーネントのサイズを加算します。

**メモ**: UNIX コンピュータ上では、`view_server` プロセスに累積的なキャッシュ パフォーマンスとほかの統計情報をログ ファイル `/var/adm/rational/clearcase/log/view_log` に書き込ませて、すべての統計情報累積カウンタをゼロにリセットすることができます。これを行うには、ビュー サーバーが実行されているホストで次のコマンドを実行します。

```
kill -HUP view_server-process-ID
```

## 出力の分析

`getcache -view` を実行すると、次の情報が得られます。

- キャッシュの種類
- 使用されているビュー キャッシュの割合
- キャッシュの項目数
- 出された要求の回数
- キャッシュのヒット率
- 使用されているビュー キャッシュのメモリ量

この情報を分析する際には、次のことに留意します。

- 使用されているビュー キャッシュの割合をチェックします。この値は特に重要です。
  - ヒット率が **90%** 以下でビュー キャッシュが **100%** であれば、ビュー キャッシュは少なすぎます。ヒット率が **90%** 以上で、ビュー キャッシュが **100%** という組み合わせは、キャッシュ サイズがほぼ適正であることを意味します。
  - 各部分で **50%** 未満であれば、キャッシュは大きすぎる可能性があります。
- このチェックを 2 回以上実行して、それらの値が、新しく起動されたり再起動されたりしたときのビュー サーバーの異常で一時的なものではないことを確認します。
- あるビューのビュー キャッシュ サイズを増やす場合は、「ビューの再設定」で説明する手順に従います。

## ビューの再設定

---

`setcache -view` を使用すると、ビュー サーバーのキャッシュを新しいサイズに再設定することができます。各サブキャッシュに割り当てられるメモリの割合は固定です。個別にキャッシュ サイズを指定する場合は、次のガイドラインに従ってください。

- 32 ビット プラットフォームの場合は **50 KB** 未満の値、64 ビット プラットフォームの場合は **100 KB** 未満の値を指定することはできません。
- このビューに割り当てるサーバー ホスト上の物理メモリ容量よりも大きな値を指定しないでください。
- 一般に、キャッシュ サイズを大きくすると、ビューのパフォーマンスが向上します。ただし、キャッシュ サイズが **4 MB** に近づくにつれて、パフォーマンスの向上は見えにくくなります。
- 定期的にヒット率と利用割合をチェックして変更を検証し、改善効果を確認する必要があります。

ある 1 つのビューに新しいキャッシュ サイズを指定するには、`setcache -view` コマンドを使用します。

## リファレンス情報

---

ここでは、**mvfsstat** と **mvfscache** ユーティリティのリファレンス情報を示します。これらのユーティリティは、主に **Rational** カスタマ サポートが **MVFS** で発生した問題を診断する際に役立てることを目的としたものです。

これらのユーティリティのいずれにも、ロックとマスタースhipの制限は適用されません。ただし、オプションの中には、権限を持つユーザーの状態が必要なものがあります。

### mvfscache

**mvfscache** はホストの **MVFS** キャッシュを操作します。**mvfscache** のオプションのほとんどは、**setcache -mvfs** または **getcache -mvfs** によって提供される同様のオプションと重なるものです。可能な場合は、それらのコマンドを使用することをお勧めします。

#### 構文

- キャッシュの状態を調べます。

```
mvfscache [ cache_name ]
```

- キャッシュ操作を制御します。

```
mvfscache { -e cache_list | -d cache_list | -f cache_list }
```

- 名前キャッシュの内容を表示します。

```
mvfscache -p [ -n name ] [ -v dbid ] [ -i ]
```

#### オプションと引数

**キャッシュの状態の調査**: オプションと引数を指定しないで **mvfscache** を実行すると、**MVFS** の全キャッシュの状態 (有効/無効) が表示されます。オプションを使用しないで引数としてキャッシュ名を指定すると、**mvfscache** は何も表示しませんが、この場合は該当する終了状態が返されます。

0                      指定したキャッシュが有効です。

1                      指定したキャッシュが無効です。

**キャッシュ操作の制御**: 次のいずれか 1 つのオプションを使用して、単一のキャッシュ、一連のキャッシュ、キャッシュ関連の動作を制御します。

## **-e cache-list**

(権限を持つユーザーの状態が必要) 指定したキャッシュとキャッシュ関連の動作 (cto を指定した場合) を有効にします。cache-list では、スペースを使用せず、カンマで区切って指定します。また、次のキーワードをいくつでも指定できます。

|              |                                                                                                                                                                                                                                                          |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>attr</b>  | 属性キャッシュ。最近アクセスしたファイル システム オブジェクトに関する情報をキャッシュします。                                                                                                                                                                                                         |
| <b>name</b>  | 名前キャッシュ。最近アクセスしたファイルとディレクトリに関する名前ルックアップ変換をキャッシュします。                                                                                                                                                                                                      |
| <b>noent</b> | 見つからなかった名前のキャッシュ。最近ルックアップしたが見つからなかったファイルとディレクトリの名前をキャッシュします。                                                                                                                                                                                             |
| <b>slink</b> | シンボリック リンク テキストのキャッシュ。最近アクセスしたシンボリック リンクの内容をキャッシュします。                                                                                                                                                                                                    |
| <b>rvc</b>   | VOB ルート バージョン キャッシュ。動的ビューごとに VOB マウント ポイントのデータをキャッシュします。                                                                                                                                                                                                 |
| <b>cto</b>   | (キャッシュ関連の動作) 閉じてから開くまでの整合性に関する動作。各オペレーティング システムの <b>open</b> 操作で、ビュー サーバーに対して「ファイル情報取得」操作を強制的に実行します。<br><br>mvfsstat または mvfstime を実行して得られた結果が活発な cto アクティビティを示していて、ユーザーがビューを共有していない場合は、この動作を無効にするとパフォーマンスが向上することがあります。ただしこの動作を無効にすると、整合性に支障がきたすことがあります。 |

## **-d cache-list**

(権限を持つユーザーの状態が必要) 指定したキャッシュとキャッシュ関連の動作を無効にします。構文は、-e の場合と同じです。

## **-f cache-list**

指定したキャッシュをフラッシュします。このオプションを使用する場合は、必ず Rational カスタマ サポートの指示に従ってください。cache-list では、次のキーワードをいくつでも指定できます (スペースを使用せず、カンマで区切って指定します)。

|              |                                                                                               |
|--------------|-----------------------------------------------------------------------------------------------|
| <b>mnode</b> | mnode フリーリスト キャッシュ。attr と slink のキャッシュ、開いているフリーリスト ファイル、フリーリストの全 mnode の mnode 記憶領域をフラッシュします。 |
| <b>name</b>  | 名前キャッシュ。見つからなかった名前のエントリもフラッシュします。                                                             |
| <b>rvc</b>   | VOB ルート バージョン キャッシュ。                                                                          |
| <b>lcred</b> | cleartext のルックアップ権限に関するグローバル認証情報キャッシュ。                                                        |

**名前キャッシュの内容表示**：-p を単独で使用するか、-n、-v、-i のうちの 1 つ以上と同時に使用します。名前キャッシュには、最近アクセスしたファイルとディレクトリに関する名前ルックアップ変換が保存されています。名前ルックアップ変換の先頭行の形式は、次のとおりです。

VOB-tag            view:directory-dbid      name ==> view:lookup-dbid

-p

名前キャッシュの内容を出力します。

-n name

name に一致する名前キャッシュのエントリのみを出力します。

-v dbid

directory-dbid (name が見つかったディレクトリのデータベース ID) または lookup-dbid (ルックアップ結果のデータベース ID) に一致する名前キャッシュのエントリのみを出力します。

-i

名前キャッシュの無効なエントリを出力に含めます。該当するエントリは、不正としてマークされ、ルックアップには使用されません。しかしこのエントリは、統計上の目的で保持されています。この情報を利用すると、無効なエントリが新しいデータに置換される頻度がわかります。通常、VOB での変更によって MVFS がキャッシュのリフレッシュを要求された場合に、無効化が実行されます。

## 例

- すべてのキャッシュの状態を調べます。

```
mvfscache
Attr: on
Name: on
Noent: on
Rvc: on
Slink: on
Cto: on
```

- VOB のアンマウントに備えて、使用中のマウント ポイントをクリアします。

```
mvfscache -f mnode
```

## mvfsstat

mvfsstat コマンドは、MVFS の使用状況と動作に関する統計情報を表示します。たとえば、MVFS キャッシュの使用に関する累積的な統計情報、RPC に関する統計情報、cleartext の I/O カウント、vnode の操作カウント、VFS の操作カウントなどの情報が表示されます。このデータは、ファイルシステムのパフォーマンスを評価して、MVFS キャッシュ サイズの調整が必要かどうかを決定する場合に役立ちます。

## 構文

- UNIX

```
mvfsstat [ -icrvVhalzAdF ] [ -o outfile ] [ time ] [ count ]
```

- Windows

```
mvfsstat [ -ilcrvVhalzAdF ] [ -o outfile ] [ time ] [ count ]
```

## MVFS キャッシュの統計情報

-c オプションを指定すると、ホストの MVFS キャッシュの使用が報告されます。このレポートは累積的なもので、MVFS の起動時以降の期間全体をカバーしています。ClearCase クライアント ホスト上の MVFS のパフォーマンスを調整するために役立つ各種の統計情報について、後の項で説明します。

### ディレクトリ名ルックアップ キャッシュ (dnlc)

dnlc セクションは、パス名を ClearCase の識別子にマッピングする名前ルックアップ キャッシュの使用を報告します。キーワードの前に値が表示されている点に留意してください。たとえば、「1301984 dot」は、dot に関して報告された統計情報値が 1301984 であることを示しています。

**キャッシュ ヒット:** hit の行は、キャッシュ内で特定のエン트리 タイプが見つかった回数 (ヒット回数) について報告します。

|       |                                               |
|-------|-----------------------------------------------|
| dot   | 現在の作業ディレクトリがルックアップされた回数 (常にキャッシュ ヒット)         |
| dir   | キャッシュ内でディレクトリ オブジェクト エントリが見つかった回数             |
| reg   | キャッシュ内でファイル オブジェクト エントリが見つかった回数               |
| noent | キャッシュされた File not found (ENOENT) エントリが見つかった回数 |

大きなツリーを移動するようなアクティビティでは、キャッシュのヒット率が低くなります (約 50%)。このようなアクティビティには、多数のファイルを調べて、ビルドが不要だと決定する find コマンドや再帰的な clearmake コマンドなどのアクティビティがあります。

**キャッシュの失敗:** miss 行では、キャッシュの失敗の総数について報告します。events の値は、重大な VOB イベント、エントリのタイムアウト、vnode の再利用のために発生したキャッシュの失敗の数です。キャッシュの失敗は、キャッシュ内にエントリがないために発生することもあります。キャッシュの失敗の総数は、events の値と、キャッシュ内にエントリがないために発生する失敗の数を合計した値です。

**キャッシュの追加:** add 行では、新しいエントリがキャッシュに追加中であったために発生したキャッシュの失敗を報告します。追加は、ディレクトリ エントリ (dir)、ファイル エントリ (reg)、ENOENT エントリ (noent) に分類されます。

**属性キャッシュ** : `attr` セクションでは、**stat** が返すキャッシュの使用 (UNIX の場合)、またはオブジェクト状態問い合わせレコードのキャッシュの使用 (Windows の場合) を報告します。一般に、このキャッシュのヒット率は、ディレクトリ名ルックアップ キャッシュのヒット率と同程度になります。

## オプションと引数

**time**

サンプル間の秒数。各サンプルについて差分を表示します。このオプションを省略すると、すべての情報について絶対値のみが表示されます。

**count**

サンプル数。省略した場合、デフォルト値は無限大です。

**-o outfile**

`outfile` に出力を書き込みます。

**-i**

`cleartext` の I/O カウントと待機時間を表示します。

**-l**

MVFS によって処理された、Windows の I/O 要求パケットの数と待機時間を表示します。

**-c**

MVFS キャッシュの統計情報を表示します (302 ページの「MVFS キャッシュの統計情報」を参照)。

**-r**

MVFS のリモート プロシージャ コール (RPC) の統計情報を表示します。表示される統計情報は、カウントと実待機時間から構成されます。次の 2 つの場合、実待機時間はサンプル期間の 100% よりも大きな値になることがあります。

- ▶ 操作の終了にサンプル期間よりも長い時間がかかった場合。たとえば、30 秒のサンプルで 60 秒の待機時間が記録された場合。
- ▶ 複数のプロセスが同時に待機している場合。

一般に、実時間のパーセンテージは、VOB にアクセスしているプロセスが 1 つしかない場合に限って意味を持ちます。

**-v**

`vnode` 操作のカウントを表示します。

**-V**

`vfs` 操作のカウントを表示します。

-h

RPC ヒストグラムを表示します。**cleartext** フェッチ RPC は、ほかのすべての RPC とは別に計算されます。

-a

監視用の統計情報を表示します。

-l

**-c**、**-r**、**-i**、**-l**、**-v**、**-V** の各オプションによって生成される統計情報を、個々の操作ごとに集計してより詳細に報告します。**-c** の場合は、キャッシュ エントリごとのヒット率も表示します。

-z

(権限を持つユーザーの状態が必要) 動作中のカウンタがすべて **0** にリセットされます。

-A

すべての統計情報を表示します。

-d

**-c** または **-A** と同時に指定すると、問題の診断に使用できるデバッグ情報を表示します。このオプションを使用する場合は、必ず **Rational カスタマ サポート** の指示に従ってください。

-F

**mnode** 操作の統計情報とディレクトリ名ルックアップ キャッシュを表示します。このオプションを使用する場合は、必ず **Rational カスタマ サポート** の指示に従ってください。

# VOB ホストのパフォーマンス の改善

# 16

VOB 内のデータへのアクセスは ClearCase のほとんどすべての操作で発生します。VOB ホストが十分なパフォーマンスを発揮していれば、VOB へのアクセスがボトルネックになることはありません。この章では、VOB ホストのパフォーマンスを改善する方法について説明します。

VOB ホストでは、大量のデータベース アクセスや計算処理が行われています。VOB の使用パターンは、VOB ホストが所定のパフォーマンス レベルで同時にサポートできるユーザー数に大きな影響を与えます。たとえば、同じレベルのパフォーマンスが提供されるのであれば、VOB に書き込む場合と読み込む場合では、読み込む方が同時にサービスできるユーザーの数は多くなります。高パフォーマンスの VOB ホストの持つ特性の詳細については、97 ページの「VOB サーバー ホスト構成のガイドライン」を参照してください。

## プロセス オーバーヘッドの最小化

---

次に示す方法は、VOB ホストのパフォーマンスを改善するために特に効果的です。また、次のようにその導入も組織上の問題を除けば技術的には難しくありません。

- **重要でないプロセスを VOB ホストから排除します。**VOB ホストを、その本来のサービス以外にリソースを多量に消費するアプリケーション（たとえば、データベース管理システムや Web サーバー）、またはシステム レベルのネットワーク サービス（たとえば、UNIX の NIS サーバーや Windows のプライマリ ドメイン コントローラ）のサーバー ホストとして使用しないでください。
- **ClearCase クライアント プロセスを VOB ホストから排除します。**VOB サーバー ホストを ClearCase のアクティブなクライアント（たとえば、開発者のデスクトップシステム）として使用しないでください。
- **view\_server プロセスを VOB ホストから排除します。**VOB サーバー ホストに共有ビューを作成することはできますが、この方法をとると、VOB とビューのパフォーマンスの両方に悪い影響を与えることがあります。可能であれば、それは避けてください。

## ディスク パフォーマンスの最大化

---

VOB ホスト上で最高の I/O パフォーマンスを得るために、次の処理をお勧めします。

- シーク時間と回転速度が高速なディスクを使用する。
- 可能なら、VOB ごとにディスク スピンドルを割り当てる。
- UNIX ホストで VOB の使用頻度が最も高い場合は、VOB あたり 2 台のスピンドルを専用に割り当てる (1 台をデータベース プールとソース プール用にし、もう 1 台を **cleartext** と DO プール用にする)。
- ストライプ技術 (RAID 0) を適用してパフォーマンスを改善する。
- 可用性を向上させる場合は、ミラーリング (RAID 1) を適用する。
- ベンチマークが RAID (0+1) のパフォーマンスに等しい場合を除き、RAID 5 を避ける。
- 動作確認済みの NAS デバイスに VOB 記憶領域を配置する。

ディスクのストライピングやミラーリングの適用法の詳細については、システムに付属のマニュアルを参照してください。

## Windows でのディスク キャッシュ用のメモリの追加

---

Windows には、動的ディスク バッファ キャッシュの機能があります。できるだけ多くのメインメモリが、ユーザー プロセスによって更新されたデータ ファイルのブロックをキャッシュするために使用されます。バッファ キャッシュは定期的にディスクに書き込まれます。ホストにメモリを追加すると、キャッシュ サイズは増えます。

この機能は、ディスクの入出力を著しく高速化します。この機能を十分に利用することは VOB ホストのパフォーマンス向上のために非常に重要です。ディスク バッファ キャッシュが十分でないと、ClearCase のパフォーマンスが著しく低下し、次のような現象が発生します。

- **scrubber** や **vob\_scrubber** の実行に長い時間がかかる。
- **omake** や **clearmake** のビルドに非常に長い時間がかかる。
- クライアントで RPC タイムアウト エラーが発生する。

メイン メモリと VOB サイズの関係の詳細については、97 ページの「VOB サーバー ホスト構成のガイドライン」を参照してください。

## UNIX でのブロック バッファ キャッシュの調整

ClearCase がサポートする UNIX プラットフォームの多くは、動的ブロック バッファ キャッシュを備えています。できるだけ多くのメイン メモリが、ユーザー プロセスによって更新されたファイル ブロックをキャッシュするために使用されます。このキャッシュは定期的にディスクに書き込まれます。ほとんどのタイプの UNIX では、ホストにメモリを追加することで、それに応じてキャッシュ サイズも増えます。アーキテクチャに依存するブロック バッファ キャッシュ操作については、オンラインの『Rational ClearCase プラットフォーム別情報ガイド』を参照してください。

### ブロック バッファ キャッシュに関する統計

UNIX の `sar` ユーティリティを実行すると、ブロック バッファ キャッシュに関するアクティビティのレポートが得られます。たとえば、次のコマンドは 60 秒ごとにサンプルを取って、5 分間のアクティビティを累積して報告します。

```
sar -b 60 5
```

| 12:14:22 | bread/s | lread/s | %rcache | bwrit/s | lwrit/s | %wcache | pread/s | pwrit/s |
|----------|---------|---------|---------|---------|---------|---------|---------|---------|
| 12:15:22 | 0       | 1       | 100     | 1       | 1       | 0       | 0       | 0       |
| 12:16:23 | 1       | 1       | -60     | 2       | 2       | 0       | 0       | 0       |
| 12:17:24 | 0       | 4       | 100     | 4       | 17      | 77      | 0       | 0       |
| 12:18:25 | 0       | 6       | 100     | 3       | 145     | 98      | 0       | 0       |
| 12:19:25 | 17      | 91      | 81      | 28      | 335     | 92      | 0       | 0       |

| 12:19:25 | bread/s | lread/s | %rcache | bwrit/s | lwrit/s | %wcache | pread/s | pwrit/s |
|----------|---------|---------|---------|---------|---------|---------|---------|---------|
| Average  | 4       | 21      | 83      | 8       | 100     | 92      | 0       | 0       |

ブロック バッファ キャッシュのサイズが適切なら、キャッシュからの読み込みは 90 ~ 95% の範囲に、キャッシュへの書き込みは 75% 以上になります。UNIX のタイプによっては、バッファ キャッシュのパフォーマンスを監視するための特別なツールを備えているものがあります。オンラインの『Rational ClearCase プラットフォーム別情報ガイド』を参照してください。

**メモ:** ブロック バッファ キャッシュがディスクに書き込まれる (フラッシュされる) ときに、対話型処理のパフォーマンスが一時的に低下します。UNIX プラットフォームによっては、このイベントに対してユーザーレベルの制御 (たとえば、HP-UX の `syncer` ユーティリティ) を備えているものがあります。これにより、対話型処理のパフォーマンスに対するバッファ キャッシュの書き込みの影響を最小化することがある場合があります。ブロック バッファ キャッシュのサイズが大きいほど、フラッシュされる頻度が少なくなります。

## Lock Manager の起動オプションの変更

---

すべての VOB サーバー ホストは単一の Lock Manager プロセスを実行します。このプロセスは、ホスト上のすべての VOB に対するデータベース アクセスを制御します。Lock Manager はデフォルトの起動オプションで動作します。このオプションは、広範囲の負荷で十分なパフォーマンスを提供するように設定されています。ただし、多数の VOB または多数のユーザーをサポートする VOB サーバー ホストでは、Lock Manager のいくつかの起動オプションを変更することもできます。

### Lock Manager の実装

Lock Manager には 2 つの実装があります。1 つはソケットを使用してデータの読み書きを行うものです。もう 1 つは一部の UNIX プラットフォームで使用可能で、共有メモリを使用するものです。共有メモリにより、各要求に対する待ち時間が短くなります。ある Lock Manager の起動オプションによって許容される値の範囲は、実装によって異なります。システム上にある Lock Manager のタイプは、lockmgr\_log ファイル内の起動メッセージを調べることで知ることができます。従来の Lock Manager は次のような行を出力します。

```
db_VISTA Version 3.20
Database Lock Manager for UNIX BSD
```

共有メモリの Lock Manager は次のような行を出力します。

```
db_VISTA Version 3.20
Database Lock Manager for UNIX System V shared memory
```

### Lock Manager の起動オプションを変更するには

Windows ホストまたは UNIX ホスト上で Lock Manager の起動オプションを変更するには、以下の処理を実行します。

**メモ:** Lock Manager の起動オプションを変更した後は、ホスト上で ClearCase を停止して再起動し、変更を有効にする必要があります。

### Windows ホストの場合

Windows ホストでは、Lock Manager の起動オプションは内部的に定義されます。起動オプションのデフォルトを変更するには、次の Windows レジストリ キーを REG\_SZ 値として作成し、309 ページの「lockmgr のリファレンス情報」の説明に従って適切な num 値を指定します。

```
HKEY_LOCAL_MACHINE¥SOFTWARE¥Atria¥ClearCase¥CurrentVersion¥LockMgr
CmdLine: REG_SZ : -a almd -f num -u num -q num
```

このレジストリ キーは、ClearCase の新しいバージョンがインストールされてもそのまま残ります。

## UNIX ホストの場合

UNIX ホストでは、Lock Manager のデフォルトの起動オプションは起動スクリプト `ccase-home-dir/etc/clearcase` で定義されます。起動オプションのデフォルトを変更するには、`/var/adm/rational/clearcase/config/lockmgr.conf` というファイルを作成し、次の形式の行を記述します。

```
LOCKMGR_OPTS = -f num -u num -q num
```

309 ページの「lockmgr のリファレンス情報」の説明に従って適切な `num` 値を指定します。このファイルの所有者を `root` にし、読み取り専用アクセスに設定することをお勧めします。

このファイルは、ClearCase の新しいバージョンがインストールされてもそのまま残ります。このファイルが存在する場合、ClearCase はファイルに定義されたオプションで Lock Manager を起動します。存在しない場合は、Lock Manager の起動オプションのデフォルト値を使用します。

`/var/adm/rational/clearcase/config/lockmgr.conf` には、次の構文ルールが適用されます。

- コメントは `#` 文字で始まり、行の終わりまで続きます。
- 空白は、LOCKMGR\_OPTS ラベルの前に入れることができ、`=` の前後にも入れることができます。
- 空白は、各オプション (`-f`、`-u`、`-q`) の前では必須です。また、指定する `num` 値とオプションを区切ることができます。
- 指定されていないオプションは、デフォルト値に戻されます。不正なオプションは無視されます。

## lockmgr のリファレンス情報

lockmgr は、VOB サーバー ホスト上にある各 VOB データベースへのすべてのアクセスを調整します。これは、VOB ホスト上で ClearCase が起動されたときに自動的に起動されます。

### 構文

```
lockmgr -a almd -f num -u num -q num
```

### オプションと引数

ソケット名または共有メモリ ファイル名の指定: デフォルトは `almd` です (このデフォルトは変更しないでください)。UNIX システムでは、lockmgr は `/var/adm/rational/clearcase/almd` を介してほかのプロセスと通信します。almd はプラットフォームによって共有メモリ ファイルの場合もあれば、ソケットの場合もあります (308 ページの「Lock Manager の実装」を参照)。

## **-a almd**

lockmgr によって作成されるソケットまたは共有メモリ ファイルのリーフ名として almd を指定します。その他の名前の使用はサポートされていません。

**メモ:** 誤って削除される可能性を低くするため、/var/adm/rational/clearcase/almd は root によって所有されます。共有メモリの Lock Manager をサポートしているプラットフォームでは、このファイルはモード 777 (誰でも書き込み可能) で作成されます。ユーザーが誤ってこのファイルを変更する可能性がある場合は、書き込み権限を root と、ホスト上のすべての VOB の所有者に制限することができます。/var/adm/rational/clearcase/almd への書き込み権限を持たないユーザーによって所有される VOB はアクセス不能になります。

**データベース ファイルの数の指定:** Windows の場合のデフォルトは 128、UNIX の場合のデフォルトは 256 です。

## **-f num**

同時に開くことができるデータベース ファイルの数を指定します。各 VOB データベースは 7 個のデータベース ファイルで構成されています。デフォルトの起動値では、Lock Manager は VOB を UNIX ホストの場合は 36 個まで、Windows ホストの場合は 18 個まで処理できます。ホストに VOB がこれより多く存在する場合、Lock Manager はすべての VOB に対して同時に要求に応えることはできません。その場合、ユーザー応答時間が長くなり、db\_server\_log ファイルまたは vobrpc\_server\_log ファイルに次の形式のメッセージが記録されます。

エラー: ホスト上で開いているデータベースが多すぎます (lockmgr コマンド行の -f 引数を大きくしてください)

このエラーが発生した場合は、VOB のいくつかを別のホストに移動するか、-f オプションの値を 7\*V (V はホスト上の VOB の数を表します) まで増やしてください。

**ユーザー数の指定:** Windows の場合のデフォルトは 128、UNIX の場合のデフォルトは 256 です。

## **-u num**

同時にロックを要求できる db\_server プロセスまたは vobrpc\_server プロセスの数を指定します。それぞれのアクティブなビューは、ビューがアクセスする VOB ごとに 1 つの vobrpc\_server プロセスを必要とします。また、VOB メタデータを変更するさまざまな操作には、Lock Manager を介した、db\_server プロセスによる VOB へのアクセスを伴います。ユーザー応答時間が長くなり、次の形式のメッセージが記録されます。

db\_VISTA database -922: lockmgr is busy

このメッセージが lockmgr\_log ファイルに記録された場合は、-u オプションの値を大きくする必要があります。

共有メモリの **Lock Manager** をサポートしていないホストでは、**-u** オプションを 1018 より大きな値に設定することができません。共有メモリの **Lock Manager** をサポートしているホストでは、**-u** オプションは使用可能な仮想メモリによってのみ制限を受けます。

次の式を使用することで、**-u** の最悪のケースの近似値を計算できます。

$$V \cdot (N/4 + 5)$$

**V** はホスト上の **VOB** の数で、**N** はそれらの **VOB** にアクセスするユーザーの数です。より現実的な値 (**Lock Manager** が **VOB** サーバー ホスト上で仮想メモリを必要以上に消費しない値) にするには、通常の使用で長期間 (1、2 週間) **VOB** サーバー ホスト上で実行される **vob\_server** プロセスと **vobrpc\_server** プロセスの合計数を監視します。その後、ピーク値に、増加 (2、またはそれより少し上) を許容できる係数を掛けます。

**要求キューのサイズの指定**: Windows の場合のデフォルトは 128、UNIX の場合のデフォルトは 1024 です。

**-q num**

ロックがキューに送信されるロック要求の数を指定します。**Lock Manager** は、この値を超えるロック要求をキューに送信する場合、遅延をとります。ユーザー応答時間が長くなり、次の形式のメッセージが記録されます。

```
db_VISTA database -922: lockmgr is busy
```

このメッセージが **db\_server\_log** ファイルまたは **vobrpc\_server\_log** ファイル ( また、多くの場合、同時に **view\_log** ファイル) に記録された場合は、**-q** オプションの値を大きくする必要があります。一般に、この値は **-u** オプションの値の 5 倍を超えないようにしてください。

## サーバーのパフォーマンスと NAS デバイス

---

**VOB** またはビュー サーバーが、**VOB** 記憶領域またはビュー記憶領域に **NAS** デバイスを使用する場合 (16 ページの「**ClearCase** と **NAS (Network-Attached Storage)** デバイス」を参照) は、デバイスのその他の使用を排除するか、最小限にすることをお勧めします。

**NAS** デバイスがほかのアプリケーションによって使用される場合や、**VOB** 記憶領域やビュー記憶領域でないファイルとディレクトリに記憶領域を提供する場合、これらの使用によってデバイスの **I/O** サブシステムに発生する追加の負荷が、**VOB** 記憶領域とビュー記憶領域のディレクトリにアクセスする **ClearCase** 操作に著しい悪影響を与えることがあります。この影響は、**NAS** デバイスに **VOB** 全体 (データベースとプール) が格納される場合には特に深刻になります。

VOB サーバーが NAS デバイスをその他の使用と共有する場合は、サーバー自体に VOB データベースを保持し、NAS デバイスにはプールのみを配置することをお勧めします (この構成は、UNIX を実行するホストに対してのみサポートされます)。

データベースが NAS デバイ스에格納される VOB にアクセスするときにパフォーマンスが著しく低下する場合は、VOB データベースを VOB サーバー ホストに移動することをお勧めします。

**メモ:** CIFS (SMB) プロトコルでは付加的なネットワーク トラフィックが必要になるため、Windows を実行する VOB サーバー ホストは、VOB データベースを NAS デバイスに配置した場合にはパフォーマンスが低下する可能性が非常に高くなります。

# 異なるプラットフォームの ファイル システムへのア クセスの設定

# A

Windows コンピュータが UNIX コンピュータのファイル システムにアクセスできるように、Rational は ClearCase でいくつかのサードパーティ製品の使用を可能にしました。この機能は、Windows 上の動的ビューから、VOB または UNIX 上の動的ビューにアクセスする際に必要です。詳細については、61 ページの「異なるプラットフォーム間のファイル システムのアクセス」を参照してください。次の 2 種類の製品をサポートしています。

- NFS クライアント製品。Windows クライアント上で稼動し、NFS プロトコルを使用して UNIX ファイル システムにアクセスします。
- SMB サーバー製品。UNIX サーバー上で稼動し、SMB (または CIFS) プロトコルを使用して Windows クライアントがサーバーのファイル システムにアクセスできるようにします。

この付録では、現在サポートしているサードパーティ製のファイル システム アクセス製品の設定方法について説明します。サポートしている製品とバージョンの最新情報については、Rational のユーザー専用 Web サイトを参照してください。

**メモ:** この付録の説明は、ClearCase LT には当てはまりません。動的ビューをサポートしていないので、異なるプラットフォームのファイル システムにアクセスする必要がないためです。

## NFS クライアント製品

---

Rational は、Windows コンピュータから UNIX 上の VOB と動的ビューにアクセスできるようになる、次の NFS クライアント製品をサポートしています。

- Shaffer Solutions 社の DiskAccess
- Hummingbird 社の NFS Maestro

NFS クライアント製品を使用する場合は、次の操作を行う必要があります。

- UNIX サーバー上の VOB やビューにアクセスする各 Windows クライアントにこれらの製品をインストールする。
- インストールした製品を ClearCase の特殊な要件を満たすように設定する。

この項では、以降、ClearCase と共に NFS クライアント製品を使用する場合の設定処理について説明します。

**メモ:** NFS クライアント製品でサポートしているのは、1 台の Windows ホストからの UNIX ファイル システムへのアクセスです。つまり、NFS クライアント製品を使用して UNIX 上の VOB にアクセスする Windows コンピュータから、ほかの Windows コンピュータに、このような VOB へのアクセスを提供できません。

## 大文字/小文字の自動変換の無効化

NFS クライアント製品には、デフォルトでファイル名の大文字/小文字を自動変換 (通常は小文字に自動変換) するものがあります。これにより、MVFS と **cleartool** コマンドで問題が発生することがあります (65 ページの「大文字/小文字の区別」を参照)。そのため、この項の説明に従って、大文字/小文字の変換を無効にする必要があります。

**メモ:** 通常は、コマンド行オプションを使用して、特定の NFS マウントに対して大文字/小文字の自動変換を無効化できます。ところが、**ClearCase** ではリモート記憶ディレクトリを自動的にマウントするため、マウントが正しく行われるように、すべてのマウントで大文字/小文字の自動変換が無効になるように NFS を設定する必要があります。

## Shaffer Solutions 社の DiskAccess の場合

大文字/小文字の自動変換を無効にするには

- 1 [スタート] メニューをクリックし、[設定] をポイントして、[コントロール パネル] をクリックします。DiskAccess プログラムを起動します。
- 2 [Filenames] タブで [Preserve Case (no conversion)] をクリックします。

## Hummingbird 社の NFS Maestro の場合

バージョン 8.0 で大文字/小文字の自動変換を無効にするには

- 1 [スタート] メニューをクリックし、[設定] をポイントして、[コントロール パネル] をクリックします。[ネットワーク] をダブルクリックします。
- 2 [サービス] タブで、[Hummingbird NFS Maestro Client] を選択します。
- 3 [プロパティ] をクリックし、クライアント構成ダイアログ ボックスを開きます。
- 4 [File Access] タブの [Filename Case] で [Preserve Case] をオンにします。

8.0 より前のバージョンで自動大文字/小文字変換を無効にするには

- 1 [スタート] メニューをクリックし、[設定] をポイントして、[コントロール パネル] をクリックします。[ネットワーク] をダブルクリックします。
- 2 [サービス] タブで、[NFS Maestro for Windows NT Client] を選択します。
- 3 [プロパティ] をクリックし、クライアント構成ダイアログ ボックスを開きます。
- 4 [Filename Capitalization] の下の [Preserve Case] をクリックします。

## NFS クライアントのデフォルト保護モードの設定

UNIX ホスト上の共有ビューを使用して作業する場合は、グループに書き込みアクセス権が与えられるデフォルトの保護モードで NFS クライアントを設定します。グループに書き込みアクセス権がなければ、ある開発者が作成したビュープライベート ファイルをほかの開発者が変更できません。

### Shaffer Solutions 社の DiskAccess の場合

デフォルト保護モードを設定するには

- 1 [スタート] メニューをクリックし、[設定] をポイントして、[コントロール パネル] をクリックします。DiskAccess プログラムを起動します。
- 2 [File Access] タブで、[User] が RWX、[Group] が RWX、[Other] が RX になっていることを確認します。

### Hummingbird 社の NFS Maestro の場合

バージョン 8.0 でデフォルト保護モードを設定するには

- 1 [スタート] メニューをクリックし、[設定] をポイントして、[コントロール パネル] をクリックします。[ネットワーク] をダブルクリックします。
- 2 [サービス] タブで、[Hummingbird NFS Maestro Client] を選択します。
- 3 [プロパティ] をクリックし、クライアント構成ダイアログ ボックスを開きます。
- 4 [File Access] タブの [Default File Permissions] と [Default Directory Permissions] の両方で、次のように権限を設定します。

| User | Group | Other |
|------|-------|-------|
| RWX  | RWX   | RWX   |
| xxx  | xxx   | x x   |

8.0 より前のバージョンでデフォルトの保護モードを設定するには

- 1 [スタート] メニューをクリックし、[設定] をポイントして、[コントロール パネル] をクリックします。[ネットワーク] をダブルクリックします。
- 2 [サービス] タブで、[NFS Maestro for Windows NT Client] を選択します。
- 3 [プロパティ] をクリックし、クライアント構成ダイアログ ボックスを開きます。
- 4 [Default Protection] の下の保護モードを次のように設定します。

| User | Group | Other |
|------|-------|-------|
| RWX  | RWX   | RWX   |
| xxx  | xxx   | x x   |

## 正しいログオン名の設定

VOB とビューに対するアクセス権の問題が発生しないように、Windows ユーザー名とプライマリ グループ名に一致する UNIX ユーザー名とグループ名を使用してログオンするように NFS クライアントを設定する必要があります。これらの一致する名前が存在することを確認するには、UNIX NFS サーバーの名前を `ccase-home-dir¥etc¥utils¥credmap` に渡します。次の例では、ドメイン グループ `clearusers` のメンバーであるユーザー `NT_WEST¥akp` は、一致するユーザー アカウント (`akp`) とグループ アカウント (`clearusers`) を UNIX ホスト `saturn` 上に持っています。

```
ccase-home-dir¥etc¥utils¥credmap saturn

Identity on local system:
    ユーザー : NT_WEST¥akp
    (NT:S-1-3-21-108034363-97353062-1565875335-1402)
    プライマリ グループ : NT_WEST¥clearusers
    (NT:S-1-3-21-108034363-981762062-1565875335-1044)
    .
    .
    .
Identity on host "saturn":
    User SID: UNIX:UID-2270
    Primary group SID: UNIX:GID-88
    .
    .
    .
```

`credmap` で返されたユーザー ID またはグループ ID が負の値の場合は、ターゲットの UNIX ホストが、現在ログオンしている Windows ユーザーと一致する UNIX ユーザー名またはグループ名を見つけられないことを示しています。詳細については、59 ページの「共通なユーザー名とグループ名」を参照してください。

## Shaffer Solutions 社の DiskAccess の場合

ログオン ユーザー名を設定するには

- 1 [スタート] メニューをクリックし、[設定] をポイントして、[コントロール パネル] をクリックします。DiskAccess プログラムを起動します。
- 2 [Authentication] タブで、[User Name]、[Password]、[PCNFSD Server] に正しい値を入力します。
- 3 [OK] をクリックします。ログオン セッションが検証されます。

## Hummingbird 社の NFS Maestro の場合

ログオン ユーザー名を設定するには、コマンドプロンプトで `nfs register` コマンドを実行します。

```
nfs register username
```

パスワードの入力を要求されます。

## Hummingbird 社の NFS Maestro: DOS 共有の無効化

NFS Maestro の [DOS Sharing] オプションは動的ビューと互換性がありません。NFS Maestro を使用して UNIX 上の動的ビューにアクセスするには、このモードを無効にする必要があります。このオプションをオフにせずに MVFS ファイルを開くと、エラーが発生する場合があります。例を次に示します。

```
ZwOpenFile returned status 0xc0000043
```

このエラーは共有違反を表します。

バージョン 8.0 で DOS 共有を無効にするには

- 1 [スタート] メニューをクリックし、[設定] をポイントして、[コントロール パネル] をクリックします。[ネットワーク] をダブルクリックします。
- 2 [サービス] タブで、[Hummingbird NFS Maestro Client] を選択します。
- 3 [プロパティ] をクリックし、クライアント構成ダイアログ ボックスを開きます。
- 4 [File Access] タブの [File Sharing] で、[None] を選択します。

8.0 より前のバージョンで DOS 共有を無効にするには

- 1 [スタート] メニューをクリックし、[設定] をポイントして、[コントロール パネル] をクリックします。[ネットワーク] をダブルクリックします。
- 2 [サービス] タブで、[NFS Maestro for Windows NT Client] を選択します。
- 3 [プロパティ] をクリックし、クライアント構成ダイアログ ボックスを開きます。
- 4 [Default Links] の下の [DOS-Style Sharing] チェック ボックスをオフにします。

## 自動マウントと NFS クライアント ソフトウェア

UNIX 上の VOB をマウントしたり、動的ビューを起動するときに、ClearCase から UNIX ファイル システム上の VOB またはビュー記憶ディレクトリにアクセスする必要があります。NFS クライアント製品を使用している場合は、[コントロール パネル] の [ClearCase] で [NFS 記憶ディレクトリの自動マウントを有効にする] チェック ボックスをオンにするかどうかにより、ClearCase からこれらの記憶ディレクトリにアクセスする方法が決まります。

ClearCase がサポートする NFS クライアント製品を使用している場合は、自動マウントを無効にすることをお勧めします。これにより、NFS クライアントは UNC 名を使用して UNIX 上の VOB とビュー記憶領域にアクセスできるようになります。ClearCase をインストールする前に DiskAccess をインストールした場合は、ClearCase のインストール処理により自動マウント機能が無効になります。

Hummingbird 社の NFS Maestro を使用する場合は、サポートされている NFS クライアント製品を ClearCase より後でインストールした場合は、次のように [コントロール パネル] の [ClearCase] を使用して自動マウントを無効にします。

- 1 [スタート] メニューをクリックし、[設定] をポイントして、[コントロール パネル] をクリックします。[ClearCase] をダブルクリックします。
- 2 [オプション] タブで [NFS 記憶ディレクトリの自動マウントを有効にする] チェック ボックスをオフにします。
- 3 [OK] をクリックします。

[NFS 記憶ディレクトリの自動マウントを有効にする] チェック ボックスをオンにした状態でも ClearCase を使用できます。その場合、ClearCase は Windows のドライブ名を UNIX の VOB とビュー記憶ディレクトリにマッピングし、そのドライブ名を使用してこれらの記憶ディレクトリにアクセスします。

**メモ:** このドライブ名は ClearCase が内部的に使用するドライブ名で、ユーザーがビュー内で作業するときに割り当てるドライブ名とは異なります。特に、ユーザーが動的ビューを起動するときに割り当てるドライブ名と混同しないようにしてください。

## DiskAccess: ClearCase サーバー プロセス ユーザーのサポート

ClearCase サーバー プロセス ユーザーがログインできるように、DiskAccess を設定する必要があります。コンピュータに DiskAccess をインストールして、DiskAccess ログオン アカウントを設定したら、以下の手順に従って、UNIX 上の ClearCase サーバー プロセス ユーザー アカウント用の追加の DiskAccess ログオン アカウントを設定します。

**メモ:** この手順を実行するには、ClearCase サーバー プロセス ユーザー用の UNIX アカウントが必要です。詳細については、59 ページの「共通なユーザー名とグループ名」を参照してください。

- 1 Windows ホストに ClearCase サーバー プロセス ユーザーとしてログオンします。
- 2 [スタート] メニューをクリックし、[設定] をポイントして、[コントロール パネル] をクリックします。DiskAccess プログラムを起動します。
- 3 [Authentication] タブで、ClearCase サーバー プロセス ユーザーの UNIX ユーザー名とパスワード (手順 1 で使用した名前とパスワード) を指定します。
- 4 [OK] をクリックします。

- 5 プログラムを終了すると、UNIX 上で ClearCase サーバー プロセス ユーザーとして認証されたことを示す確認ダイアログ ボックスが表示されます。エラーが発生した場合、または確認ダイアログ ボックスに「UID -1」と「GID -2」が表示された場合は、手順 3 で指定した名前とパスワードが無効です。

## Windows レジストリ ファイルへのユーザー識別情報の保存

ClearCase サーバー プロセス ユーザー用の DiskAccess ログオン アカウントを作成したら、Windows レジストリ キー ファイルを生成できます。このファイルにより、ほかのユーザーは ClearCase サーバー プロセス ユーザーとしてログオンしなくてもこのサポートを追加できるようになります。

- 1 creds ユーティリティを実行して、ClearCase サーバー プロセス ユーザーのユーザー SID (この例では、clearcase\_albd) を表示します。この SID は手順 3 で必要になります。

```
ccase-home-dir¥etc¥utils¥creds clearcase_albd
Windows NT user info (on local system):
.
.
SID S-1-5-21-103034363-981818062-1465874335-1064
.
```

- 2 Windows レジストリ エディタ内で、次のキーに移動します。

```
HKEY_LOCAL_MACHINE¥SOFTWARE¥Intergraph¥DiskAccess¥CurrentVersion¥Users
```

- 3 creds で返された clearcase\_albd ユーザーの SID と一致するサブキーを選択します。
- 4 レジストリ エディタで [レジストリ] の [レジストリ ファイルの書き出し] をクリックして、キーをレジストリ ファイルに保存します。DiskAccess がインストールされている Windows コンピュータ上でこのファイルを実行すると、Windows レジストリ キーが作成され、UNIX 上の ClearCase サーバー プロセス ユーザーの DiskAccess ログオンが有効になります。

## リモート記憶プールのある UNIX VOB の Windows タグ

**メモ:** この項の説明は、Windows ホストが、NFS クライアント製品を使用して、シンボリックリンクされた記憶プールを持つ VOB にアクセスする場合にのみ当てはまります。Windows クライアントから UNIX 上の VOB にアクセスするために、サポートされている SMB サーバー製品を使用している場合は、次の項に進んでください。

UNIX ホスト上の VOB に 1 つまたは複数のリモート記憶プールが含まれている場合 (118 ページの「UNIX ホストでのリモート記憶プールの作成」を参照) は、VOB の Windows リージョンに、各リモート プールへのネットワーク パスを提供する分割プール マップを含むタグが存在しない限り、NFS クライアント製品を使用する Windows ホストは VOB にアクセスできません。

リモートプールの VOB をチェックするには、UNIX コンピュータ上で次のようなコマンドを入力します。

```
cleartool lspool -long -invob /vobs/libvob
...
プール "libvob"
...
プール記憶リンク ターゲット パス名 "/net/gamma/pools/libvob.1"
  プール記憶グローバル パス名 "/net/io/vb_store/libvob/s/sdft"
...
```

「プール記憶リンク ...」という行が 1 行以上出力された場合は、次のいずれかの手順に従って、NFS クライアント製品から UNIX 上の VOB にアクセスする Windows ホスト用に設定されているリージョンに VOB のタグを作成します。

リージョン シンクロナイザを使用するには

- 1 VOB タグを選択し、[インポート] をクリックします。
- 2 [VOB タグの作成] ダイアログ ボックス内で [マウント オプションの表示] をクリックします。
- 3 [NT 専用のオプション] の下の [プールマップの分割] ボックスで、すべてのリモート記憶プールを 1 行で指定します。たとえば、次の行は 2 つのリモート プールを定義します。

```
s¥sdft¥=¥¥gamma¥pools¥s¥libvob.1 | c¥cdft¥=¥¥gamma¥pools¥c¥libvob.1
```

この例では、VOB 記憶ディレクトリは io 上にありますが、gamma 上のプールへのシンボリック リンクがあります。

**メモ:** パス名は UNC 名で指定し、各パス名の終わりを円マークで表します。個々のプールマッピングは縦棒 (|) で区切ります。

mktag コマンドを使用するには、次の例で示すように -poolmap オプションを指定します。

```
cleartool mktag -vob -tag ¥libvob -region dev_nt -replace -options ^
poolmap=s¥sdft¥=¥¥gamma¥pools¥s¥libvob.1¥|c¥cdft¥=¥¥gamma¥pools¥c¥libvob.1¥^
-host io -hpath /usr1/vb_store/libvob.vbs -gpath ¥¥io¥usr1¥vb_store¥libvob.vbs ^
¥¥io¥usr1¥vb_store¥libvob.vbs
```

この例は、コマンド行を設定するときのいくつかの重要な規則に従っています。

- poolmap 文字列で複数のプールを指定します。各プールを縦棒 (|) で区切ります。縦棒の前にカレット (^) を入力します。
- プールの場所は UNC 名で指定します。
- -host、-hpath、-gpath の 3 つの引数をすべて指定します。
- VOB 記憶ディレクトリへの UNC 名は、-gpath 引数で指定します。

## poolmap 構文

cleartool mktag の poolmap マウント オプションで指定するプール指定構文を次に示します。

pool-spec := symlink-source¥=symlink-target¥

各パラメータの内容は次のとおりです。

- **symlink-source:** リモート プールへのシンボリック リンク。VOB 記憶ディレクトリに対する相対リンクです。
- **symlink-target:** リンクしたプールの UNC 形式のフル パス名。ローカルの Windows コンピュータ上で稼働中の NFS クライアント製品によってマウントされた UNIX ディレクトリ内にプールが存在している必要があります。このパス名は、リージョン内のすべてのコンピュータで有効でなければなりません。

## 既存の VOB タグへの記憶プールのマッピング

リモート プールがある VOB の Windows リージョンに間違っただグを作成し、poolmap を指定しなかった場合、タグを変更するか交換するまで Windows ホストから VOB にアクセスできません。

この問題を解消する最も簡単な方法は、ClearCase 管理コンソールまたは rmtag コマンドを使用してタグを削除し、51 ページの「新しいレジストリ リージョンにタグを作成するには」の説明に従ってタグを再度インポートする方法です。

または、次の手順に従って、ClearCase 管理コンソールを使用して、タグを修正するか、分割プール マップを含むタグに交換することもできます。

- 1 次のどちらかのノードに移動します。
  - VOB 記憶ディレクトリが存在するホストの VOB 記憶ノードの Tags サブノード
  - ClearCase レジストリ ノードにおけるタグのリージョンの VOB Tags サブノード
- 2 詳細ペインで VOB タグを選択します。
- 3 [操作] メニューの [プロパティ] をクリックします。ClearCase 管理者グループのメンバーであれば、表示されたダイアログ ボックスでタグのプロパティを変更できます。
- 4 ダイアログ ボックス内の [マウント オプション] タブをクリックします。[プールマップの分割] ボックスで、321 ページの「poolmap 構文」で説明した構文を使用してプール マッピング文字列を入力します。

または、cleartool rmtag コマンドと mktag コマンドを使用して、古いタグを削除して、同じ名前で、分割プール マップのある新しいタグを作成することもできます。

## SMB サーバー製品

---

Rational では、UNIX ファイル システムを Windows コンピュータからアクセスできるようにする 2 つの SMB サーバー製品をサポートしています。これら 2 つの製品は、[www.samba.gr.jp](http://www.samba.gr.jp) から入手できる Samba と、LSI Logic Storage Systems Inc. の Syntax TotalNET Advanced Server (TAS) です。この項では、これらのインストールと構成方法について説明します。

ClearCase では、Samba と TAS の一部のバージョンをサポートしていません。Rational ClearCase のリリース ノートには、SMB サーバー製品の最新情報が記載されており、サポートしているバージョンとプラットフォームに関する情報も含まれています。

### Samba のインストールと設定

Samba は、[www.samba.gr.jp](http://www.samba.gr.jp) からダウンロードできます。ダウンロードしたら、インストール先のオペレーティング システムに対応したインストール手順に従ってください。Samba は、Windows からアクセスする各 UNIX VOB と各ビュー サーバーにインストールして設定する必要があります。

ClearCase が使用できるように Samba を設定するには

- 1 ClearCase サーバー プロセス ユーザー用の Samba ユーザー名マップを作成します。
- 2 Samba 全体設定を設定します。
- 3 VOB とビュー記憶領域用の共有を作成します。
- 4 Samba サービスを開始します。

### ClearCase サーバー プロセス ユーザーのマッピング

Samba には、ClearCase サーバー プロセス ユーザーの Windows アカウントを適切な UNIX ユーザー アカウントに関連付けるユーザー名マップが必要です (アカウントの詳細については、59 ページの「共通なユーザー名とグループ名」を参照)。この項の例では、Windows 上の ClearCase サーバー プロセス ユーザー アカウントが `clearcase_albd` であると仮定します。

Samba ユーザー名マップを作成するには、Samba がインストールされているホスト上でテキスト エディタを使用し、`username.map` という名前のファイルを作成します。このファイルは、Samba のほかの構成ファイル (`smb.conf` など) がインストールされているディレクトリに作成することをお勧めします。

ファイルには、以下の形式の行を含める必要があります。

```
account = clearcase_albd
```

`account` は、60 ページの「ClearCase サーバー プロセス ユーザーに対する UNIX 認証」で説明している条件を満たす既存の UNIX ユーザー アカウントの名前です。`username.map` ファイルの詳細については、Samba のマニュアルを参照してください。

## Samba Web Administration Tool (SWAT) の使用

この項の例では、Samba と共にダウンロードされる Samba Web Administration Tool (以下 SWAT) を使用して Samba を設定する方法について説明します。ダウンロードに含まれている説明ファイルに、このツールを有効にする方法が記載されています。

SWAT インターフェイスにアクセスするには

- 1 Web ブラウザで、次の形式の URL を入力します。

`http://computer:port#`

`computer` には Samba をインストールした UNIX VOB またはビュー ホストを指定し、`port#` には SWAT ポート番号を指定します (デフォルト値は 901)。

- 2 `root` としてログオンします。SWAT インターフェイスがブラウザに表示されます。

## ClearCase 用の Samba 全体設定の設定

SWAT インターフェイスのホームページ上部にある [GLOBALS] アイコンをクリックし、[Advanced View] をクリックします。表 14 の説明に従って、全体設定のオプションを設定します。

表 14 ClearCase 用の Samba 全体設定の設定 (1/2)

| Base options      |                                                                                                       |
|-------------------|-------------------------------------------------------------------------------------------------------|
| workgroup         | このサーバーにアクセスする ClearCase ホストが属している Windows ドメインの名前。                                                    |
| netbios name      | このコンピュータのホスト名。                                                                                        |
| Security options  |                                                                                                       |
| security          | DOMAIN (推奨) または USER。Samba のセキュリティ オプションの詳細については、 <code>smbpasswd</code> についての Samba のマニュアルを参照してください。 |
| encrypt passwords | Yes                                                                                                   |
| create mask       | 0775                                                                                                  |
| directory mask    | 0775                                                                                                  |
| username map      | <code>username.map</code> ファイルのローカル パス名。                                                              |
| Locking options   |                                                                                                       |
| oplocks           | No                                                                                                    |
| kernel oplocks    | No                                                                                                    |

**表 14 ClearCase 用の Samba 全体設定の設定 (2/2)**

| File-name handling options |     |
|----------------------------|-----|
| case sensitive             | No  |
| preserve case              | Yes |

ClearCase には、このほかに Samba 全体設定に関する特別な要件はありません。各自のサイトに適するように設定できます。

## VOB とビュー記憶領域用の共有の作成

サーバー記憶場所または個別の VOB やビュー記憶ディレクトリを保持する 1 つまたは複数の Samba 共有を作成する必要があります。Samba 共有を作成するには、次の手順に従います。

- 1 SWAT インターフェイスのホームページ上部にある [SHARES] アイコンをクリックします。
- 2 [Create Share] ボタンの右にあるテキスト ボックスに共有の名前を入力します。管理が容易になるように、手順 4 で入力する UNIX ディレクトリ名に類似した共有名を指定することをお勧めします。
- 3 [Create Share] をクリックします。
- 4 [Base Options] の下にある path オプションを編集します。VOB またはビュー記憶領域が存在するディレクトリを設定します。VOB またはビュー記憶領域は、指定したディレクトリに存在しなくてもかまいませんが、そのディレクトリの下位ディレクトリのいずれかに存在している必要があります。
- 5 [Commit Changes] をクリックします。

## Samba サービスの起動

Windows コンピュータが Samba を使用してファイルをアクセスするには、Samba の **smbd** サービスと **nmdb** サービスが事前に実行されている必要があります。起動時に **smbd** サービスと **nmdb** サービスが開始されるように UNIX ホストを設定することをお勧めします。サービスの自動スタートアップを設定するためのプラットフォーム固有の手順については、Samba のマニュアルを参照してください。

SWAT インターフェイスを使用して Samba のサービスを手動で起動することもできます。

- 1 SWAT インターフェイスのホームページ上部にある [STATUS] アイコンをクリックします。
- 2 [Start smbd] をクリックします。ページが更新され、smbd のステータスが `running` になります。
- 3 [Start nmbd] をクリックします。ページが更新され、nmbd のステータスが `running` になります。

## Samba をサポートするための ClearCase の設定

MVFS がインストールされていて、Samba 共有にアクセスする Windows 上のすべての ClearCase クライアントで、次の手順に従って [コントロール パネル] の [ClearCase] で [MVFS のパフォーマンス] の設定を変更します。

- 1 [スタート] メニューをクリックし、[設定] をポイントして、[コントロール パネル] をクリックします。[ClearCase] をダブルクリックします。
- 2 [MVFS のパフォーマンス] タブで、次の設定を行います。
  - [VOB フリー リストに保持する mnode 最大数] と [cleartext フリー リストに対し保持する mnode 最大数] の両方で [上書き] を選択します。
  - Samba 2.2.6 以降の 64 ビット ビルドを使用している場合は、両方の値を 800 に設定します。
  - ほかのバージョンの Samba を使用している場合は、両方の値を 200 に設定します。
- 3 [OK] をクリックして変更内容を適用し、ダイアログ ボックスを閉じます。
- 4 Windows をシャットダウンして再起動します。

## ClearCase 以外のファイルを使用した Samba の設定のテスト

Samba を使用して VOB やビューにアクセスする前に、ClearCase 以外のファイルやディレクトリを使用して Samba のインストールと設定が正しく行われたかどうかをテストすることをお勧めします。その手順を次に示します。

- 1 Samba サーバー上にディレクトリ (たとえば、`/testshare/testdir`) を作成し、その中にテストファイル (たとえば、`/testshare/testdir/testfile`) を作成します。
- 2 共有名として `testshare`、共有のパス名として `/testshare` を使用して Samba 共有を作成します。
- 3 Windows クライアントで、Samba 共有にファイルを作成します。そのファイルの UNIX 側のユーザーとグループに所有権と権限が正しく設定されていることを確認します。
- 4 Samba 共有にすべての Windows クライアントからアクセスできることを確認します。テスト権限やアクセス制限についても確認します。

## ClearCase と共に使用する Samba の設定のテスト

ClearCase と Samba が共に正しく機能していることを確認するには

- 1 UNIX の VOB サーバーまたはビュー サーバーで、この章で説明した手順に従って Samba をインストールして設定します。VOB 記憶領域かビュー記憶領域またはその両方に対応する共有を作成します。
- 2 73 ページの「VOB とビューのアクセス コントロールの基本事項」の説明にしたがって、ClearCase ユーザーとグループに対する割り当てが適切に行われていることを確認します。
- 3 UNIX クライアントからサーバー上の VOB とビューにアクセスできることを確認します。
- 4 Windows で ClearCase クライアントにログオンします。リージョン シンクロナイザを使用して、UNIX サーバー上にある VOB とビューの VOB タグとビュー タグを Windows リージョンにインポートします。
- 5 ClearCase のいくつかの基本操作 (たとえば、mkelem、checkin、checkout) を実行して、これらのビューと VOB を使用できることを確認します。

## TAS のインストールと設定

動的ビューを使用する Windows コンピュータがいくつかの UNIX プラットフォーム上の VOB とビューにアクセスできるように、ClearCase は LSI Logic Storage Systems Inc. の TotalNET Advanced Server (以下 TAS) SMB サーバー製品をサポートしています。

この項では、TAS のインストール方法と、異なるプラットフォーム間でのファイル アクセスをサポートするための TAS と ClearCase の設定方法について説明します。TAS を使用する場合は、Windows クライアントからアクセスする各 UNIX の VOB とビュー サーバーに TAS をインストールして設定する必要があります。

Windows からアクセスする必要がある各 VOB とビュー サーバー上に TAS をインストールする方法については、『TotalNET Advanced Server リリース ノート』の該当するプラットフォームのインストール手順に従ってください。

**メモ:** Rational は、ClearCase で TAS 6.x と 7.x の両方をサポートしています。TAS 7.x には、簡易形式の設定 GUI が組み込まれています。これについては 331 ページの「ClearCase をサポートするための TAS 7.x の設定」で説明します。

## AIX 上でのマルチユーザー カーネル ドライバの有効化

TotalNET Advanced Server を AIX プラットフォームにインストールしている場合は、TAS のインストール後にマルチユーザー カーネル ドライバを有効にする必要があります。これにより、ClearCase を AIX 上の TAS と共に使用する際に必要な、TAS SMB マルチプレクサをサポートするようになります。

マルチユーザー カーネル ドライバを有効にするには、TAS の `smbmxenable` コマンドを使用します。無効にするには、`smbmxdisable` コマンドを使用します。

**メモ:** Framework インターフェイスを使用してマルチユーザー サポートを有効化または無効化することはできません。コマンド行を使用する必要があります。AIX プラットフォームでのマルチユーザー サポートの詳細については、TAS の『Administration Manual』を参照してください。

## Syntax Administration Framework へのアクセス

TAS の構成と管理は、Syntax Administration Framework Web インターフェイスを使用して行います。Framework にアクセスするには、次の手順に従います。

- 1 Web ブラウザで次の形式で URL を入力します。

`http://computer:port#`

各パラメータの内容は次のとおりです。

- `computer` には、TAS をインストールした UNIX VOB またはビュー ホストのホスト名を指定します。
- `port#` には、Framework ポート番号 (デフォルトは 7777) を指定します。

[Syntax Enterprise Services] ページが表示されます。

- 2 [Syntax Administration Framework] をクリックします。Framework ログオン プログラムが表示されます。
- 3 `root` としてログオンします。ブラウザに Framework インターフェイスが表示されます。
- 4 `sphere` フレームで [TAS Configuration and Administration] をクリックします。メニューフレームに TAS の設定と管理のメニューが表示されます。

## TAS の初期セットアップの実行

**メモ:** TAS の既存のインストールをアップグレードする場合は、既存の TAS ボリュームや ClearCase をサポートするファイル サービスを含む以前の設定が維持されるため、この項の以降の部分を読み飛ばしてかまいません。アップグレードが完了したら、ClearCase 記憶領域を含んでいる各 TAS ボリュームで opportunistic ロックが無効になっていることを確認します (ボリューム定義の [Support opportunistic locks] チェック ボックスがオフになっている必要があります)。詳細については、TAS の『Administration Manual』を参照してください。

サーバー上に TAS を初めてインストールした場合は、TAS の『Administration Manual』に従って、その TAS インストール環境で初期セットアップを実行する必要があります。Framework Web インターフェイスのメニュー フレームで [Initial Setup] をクリックします。TAS のマニュアルに従って初期セットアップを実行しますが、以下の各項に示す、TAS を ClearCase と共に使用する環境に固有な変更点に注意してください。

TAS の設定に関する詳細については、TAS の『Administration Manual』を参照してください。

## TAS の基本設定

[General TAS Settings] ペインの [Admin user]、[Admin group] などの項目はデフォルト値をそのまま使用します。

### CIFS レルムの有効化と設定 (TAS 6.x)

[Select Realms to Configure] ペインで、CIFS レルムを有効にし、[Next] をクリックします。  
[CIFS Realm Configuration] ペインが表示されます。

CIFS レルムを次のように設定します。

- **Server Name:** デフォルトで表示されていない場合、サーバー名を入力します。
- **Workgroup:** ClearCase のホストとユーザーが所属する Windows ドメイン名を入力します。
- **Transport:** 各サイトで使用しているプロトコルを選択します。
- **Device for NetBEUI:** デフォルト値をそのまま使用します。
- **WINS Server(s):** CIFS ファイル サービスでプロキシ サーバー認証モードを使用する場合は、認証プロキシサーバーが存在するネットワークの WINS サーバーの IP アドレスを指定する必要があります。

### ClearCase サーバー プロセス ユーザーのマッピング (TAS 6.x)

TAS には、ClearCase サーバー プロセス ユーザーの Windows アカウントを適切な UNIX ユーザー アカウントに関連付けるユーザー名マップが必要です (アカウントの詳細については、59 ページの「共通なユーザー名とグループ名」を参照)。この項では、Windows 上の ClearCase サーバー プロセス ユーザー アカウント名を `clearcase_albd` と仮定します。

TAS ユーザー名マップを作成するには

- 1 メニュー フレームで [TAS System] をクリックします。[TAS System Configuration and Administration] ペインが表示されます。
- 2 [Username Maps] をクリックします。[Username Maps] ペインが表示されます。ClearCase をサポートするために次の変更を行います。
  - テキスト ボックスに適切な UNIX ドメイン ユーザー アカウント名を入力します。  
[Create] をクリックします。このアカウントは、60 ページの「ClearCase サーバー プロセス ユーザーに対する UNIX 認証」で説明している条件を満たしている必要があります。
  - [List of client accounts] に、「clearcase\_albd」と入力します。

フォームの一番下にある [Submit] をクリックし、確認メッセージが表示されたら [OK] をクリックします。

## ボリュームの作成 (TAS 6.x)

VOB やビューの物理的な記憶領域が存在するディレクトリをエクスポートする、TAS ボリュームを作成します。クライアントはこのボリューム名を使用して、VOB やビューの物理的な記憶場所へのパスを示します。

**メモ:** 実際に TAS を使用して VOB やビューにアクセスする前に、通常のファイルを使用して TAS のインストールと設定が正しく行われたかどうかをテストすることをお勧めします。

ClearCase で使用する TAS ボリュームを作成するには

- 1 メニュー フレームで [TAS System] をクリックし、次に [TAS System Configuration and Administration] ペインで [Volumes] をクリックします。
- 2 テキスト ボックスにボリューム名 (たとえば、**ccstore**) を入力します。

ボリューム名は、アクセスするすべてのレルムから使用できる形式にする必要があります。たとえば、12 文字以上のボリューム名を使用できないレルムもあります。

**メモ:** このテキスト ボックスに入力するのは、ボリューム記憶領域のパス名ではなくボリュームのシンボリック名ですが、VOB やビューの記憶場所へのパスに関連した TAS ボリューム名を指定する方がわかりやすくなります。たとえば、UNIX コンピュータで /ccstore と関連する TAS ボリュームの名前を **ccstore** とします。名前に関連性がない場合は、そのボリュームに関連付けられたパス名をボリュームのプロパティで確認できます。

- 3 [Create] をクリックします。[New Volume Definition] ペインが表示されます。ClearCase をサポートするために次の変更を行います。

- **Pathname:** 記憶領域の仮想ルートへのパス名を入力します。VOB サーバーとビューサーバーでは、このパス名が VOB 記憶領域またはビュー記憶領域のルートになります。つまり、VOB 記憶領域やビュー記憶領域はすべてこのパス名の下位に存在しなければなりません (すぐ下のサブディレクトリである必要はありません)。

たとえば、「/ccstore」と入力した場合、このボリュームの VOB とビュー記憶領域の正規の名前は /ccstore/vobstore、/ccstore/home/vobstore、/ccstore/home/project/viewstore となります。

- **Volume umask:** 「002」と入力します。
- **Filename Case:** [preserve] を選択します。
- **Support opportunistic locks:** オフにします。

フォームの一番下の [Submit] をクリックし、確認ペインで [OK] をクリックします。

## ファイル サービスの設定 (TAS 6.x)

ClearCase をサポートするために TAS ファイル サービスを設定するには

- 1 次の手順でファイル サービスにアクセスします。
  - a メニュー フレームで [CIFS (NB) Realm] をクリックします。
  - b [Manage CIFS File Services] をクリックします。ファイル サービスのリストが表示されます。
  - c 自分の TAS サーバーに対応するファイル サービスをクリックしてから、[Administer] をクリックします。ファイル サービス操作のメニューが表示されます。
- 2 [Configuration] をクリックします。ファイル サービスの更新フォームが表示されます。ClearCase をサポートするために次の変更を行います。
  - Volume references: このファイル サービスが参照し、エクスポートする、TAS ボリュームを選択します。
  - Browse master: [off] を選択します。
  - Umask: 「002」と入力します。
  - Freespace report method: [root] を選択します。
  - Windows 95 logon server: オフにします。
  - Windows NT logon server: オフにします。

フォームの一番下の [Submit] をクリックし、確認ペインで [OK] をクリックしてファイル サービスの操作メニューに戻ります。

- 3 [Authentication Options] をクリックします。[Authentication Options] フォームが表示されます。ユーザ認証モードのオプションで、[Local] または [Remote] をクリックします。

**メモ:** ClearCase 記憶領域を含む TAS ボリュームの場合は、共有モード認証を使用できません。

適切な認証モードについては、各サイトのシステム管理者に問い合わせてください。

- 4 [Remote] を選択した場合は、認証モード ペインを次のように設定します。
  - Proxies: [Proxies] をクリックして、このテキスト ボックスにプロキシ サーバーの名前を入力します。名前は、1 行に 1 つずつ入力してください。

**メモ:** 認証プロキシ サーバーが存在するネットワーク上の WINS サーバーの IP アドレスを CIFS レルムで指定する必要があります。詳細については、328 ページの「CIFS レルムの有効化と設定 (TAS 6.x)」を参照してください。
  - Use Username map: このチェック ボックスをオンにして、ファイル サービスが clearcase\_albd のユーザー名マップ (328 ページの「ClearCase サーバー プロセス ユーザーのマッピング (TAS 6.x)」で指定したもの) を参照するようにします。

[Local authentication] を選択した場合は、認証モード ペインを次のように設定します。

- Use Secure Passwords: オンにします。

**メモ** : [Local authentication] を選択した場合は、TAS ファイル サービスにアクセスするすべてのユーザーを、そのファイル サービスをサポートするサーバー上のローカルなパスワード暗号化データベースに登録する必要があります。TAS ファイル サービスをサポートするサーバーが CIFS レルム内に複数存在する場合は、各サーバーでローカルなパスワード暗号化データベースを設定する必要があります。

- Use Username map: このチェック ボックスをオンにして、ファイル サービスが clearcase\_albd のユーザー名マップ (328 ページの「ClearCase サーバー プロセス ユーザーのマッピング (TAS 6.x)」で指定したもの) を参照するようにします。

認証オプションの設定フォームの一番下にある [Submit] をクリックし、確認ペインで [OK] をクリックしてファイル サービス操作メニューに戻ります。

## ClearCase をサポートするための TAS 7.x の設定

TAS 7.x には、ClearCase サービスを設定するための専用の Framework インターフェイスが用意されています。[Configure ClearCase Service] ページには、ClearCase に固有なすべての TAS 設定手順が 1 つの画面に表示され、次の TAS サービス属性の値を設定できます。

- ClearCase が使用する TAS サービス (新しいサービスまたは既存のサービス)
- ClearCase の情報を更新する権限を持つ UNIX ユーザーと Windows ユーザー
- VOB データの記憶領域用のボリューム
- ビュー データの記憶領域用のボリューム (必要な場合)
- サービスの認証方法

[Configure ClearCase Service] 画面からこれらの値を送信すると、指定した内容で新しい TAS サービスが作成されるか、既存のサービスが変更されます。[VOB Storage Area Path] 属性と [View Storage Area Path] 属性に指定したパスに VOB 用 TAS ボリュームとビュー記憶領域用 TAS ボリュームが必要に応じて作成されます。これらのボリュームは、共有として使用できるようになり、パス名の最後の部分が名前に付けられます。たとえば、/usr2/clearcase/vobdata に作成されたボリュームは、vobdata として共有されます。パス名の最後の部分が Windows 共有に適切な名前ではない場合、共有名は VOB (VOB 記憶領域用の場合) または VIEW (ビュー記憶領域用の場合) になります。

[Configure ClearCase Service] ページで指定した属性に加え、TAS は次の属性でサービスを設定します。

- [preserve-whitespace] を on に設定します。
- Browse master、null password ログイン、Windows 95 の機能をオフにします。
- [Freespace report method] を root に設定します。
- Opportunistic locks を無効にします。

- [Filename case] を **preserve** に設定します。
- **umask** 値を **002** に設定します (ファイル サービスのデフォルト値は **007**)。
- **Username map** を有効にします。

**メモ:** ClearCase サービス用に既存のサービスを使用した場合、上記の属性値が、既存のサービス用に設定した値を上書きします。

ClearCase 用の TAS サービスを設定するには、[TAS Sphere] メニューから [CIFS Realm] を選択します。次に、[Configure ClearCase Service] をクリックして、以下の手順を実行します。

## 1 必要に応じて、以下の属性の値を入力または選択します。

- **Service Name:** 新しいサービスの名前を入力するか、再設定する既存のサービスをリストから選択します。
- **Unix ClearCase Admin Account:** Windows の ClearCase サーバー プロセス ユーザーをマッピングする UNIX ユーザー アカウントの名前を入力します。詳細については、60 ページの「ClearCase サーバー プロセス ユーザーに対する UNIX 認証」を参照してください。
- **NT ClearCase Admin Account:** Windows の ClearCase サーバー プロセス ユーザー アカウントの名前を入力します (本書では、このアカウントの名前として **clearcase\_albd** を使用しています)。
- **VOB Storage Area Path:** VOB を作成する (または作成した) UNIX ディレクトリのパスを指定します。この場所を参照するには、[Browse] をクリックします。
- **View Storage Area Path:** ビューを作成する (または作成した) UNIX ディレクトリのパスを指定します。この場所を参照するには、[Browse] をクリックします。

**メモ:** このフィールドの設定は任意です。ただし、このサーバー上で共有動的ビューをホストする場合は、ビュー ボリュームのパスを指定する必要があります。

- **Authentication:** [Local]、[Gateway]、[Proxy] のいずれかのオプションを選択して、サービスへのユーザー認証方法を指定します。[Proxy] を選択した場合は、オプションの隣にあるテキスト フィールドにプロキシ認証サーバーを指定する必要があります。

**メモ:** ClearCase で TAS を使用する場合は、プロキシ認証を選択することをお勧めします。既存のサービスを再設定している場合、そのサービスの認証方法が [Authentication] 属性に表示されます。また、既存のサービスをプロキシ認証用に設定すると、以前に設定したプロキシ サーバーがテキスト フィールドに表示されます。既存のサービスを Active Directory 認証用に設定すると、[Local] オプションが [Configure ClearCase Service] 画面にデフォルトとして表示されます。

## 2 [Submit] をクリックして、ClearCase 用の TAS サービスを再設定します。

## サービスの開始とサービス接続の承認

TAS ファイル サービスを開始してサービス接続を承認するには

- 1 メニュー フレームで [TAS System] をクリックし、[TAS System Administration] をクリックします。
- 2 [TAS System Administration] ペインで [Start Services] をクリックします。  
[Confirmation] ペインで [OK] をクリックします。次に、[OK] をクリックして [TAS System Administration] ペインに戻ります。
- 3 [TAS System Administration] ペインで [Accept Service Connections] をクリックします。  
[Confirmation] ペインで [OK] をクリックします。次に、[OK] をクリックして [TAS System Administration] ペインに戻ります。

これで、ClearCase サポートのための TAS の設定が完了しました。Framework Web インターフェイスを終了します。

## TAS をサポートするための ClearCase の構成

MVFS がインストールされていて、TAS ボリュームにアクセスする Windows 上のすべての ClearCase クライアントで、次の手順に従って [コントロール パネル] の [ClearCase] で [MVFS のパフォーマンス] の設定を変更します。

- 1 [スタート] メニューをクリックし、[設定] をポイントして、[コントロール パネル] をクリックします。[ClearCase] をダブルクリックします。
- 2 [MVFS のパフォーマンス] タブで、次の設定を行います。
  - [VOB フリー リストに保持する mnode 最大数] と [cleartext フリー リストに対し保持する mnode 最大数] の両方で [上書き] を選択します。
  - どちらも値を 800 にします。
- 3 [OK] をクリックして変更内容を適用し、ダイアログ ボックスを閉じます。
- 4 Windows をシャットダウンして再起動します。

## 通常のファイルを使用した TAS の設定のテスト

TAS を使用して VOB やビューにアクセスする前に、ClearCase 以外のファイルやディレクトリを使用して TAS のインストールと設定が正しく行われたかどうかをテストすることをお勧めします。その手順を次に示します。

- 1 TAS サーバー上にディレクトリ構造 (たとえば、`/tasstore/testdir`) を作成し、その中にテストファイル (たとえば、`/tasstore/testdir/testfile`) を作成します。
- 2 この章で説明した方法で TAS をインストールして設定します。そのとき、ボリューム名を `tasstore` とし、それに対するパス名を `/tasstore` とします。
- 3 Windows クライアントで、TAS ボリュームにファイルを作成します。そのファイルの UNIX 側のユーザーとグループが正しく設定されていることを確認します。
- 4 すべての Windows クライアントからその TAS ボリュームにアクセスできることを確認します。TAS が正しく機能していることを確認するまで、テスト権限やアクセス制限を十分にチェックします。

## ClearCase と共に使用する TAS の設定のテスト

ClearCase と TAS が共に正しく機能していることを確認するには

- 1 UNIX の VOB サーバーまたはビュー サーバーで、この章で説明した手順に従って TAS をインストールして設定します。VOB 記憶領域かビュー記憶領域またはその両方を含む TAS ボリュームを作成します。
- 2 UNIX ユーザー名とグループ名が Windows ホストで有効な名前であることを確認します。詳細については、59 ページの「共通なユーザー名とグループ名」を参照してください。
- 3 UNIX クライアントから UNIX サーバー上の VOB とビューにアクセスできることを確認します。
- 4 Windows で ClearCase クライアントにログオンします。リージョン シンクロナイザを使用して、UNIX サーバー上にある VOB とビューの VOB タグとビュー タグを Windows リージョンにインポートします。
- 5 ClearCase のいくつかの基本操作 (たとえば、`mkelem`、`checkin`、`checkout`) を実行して、これらのビューと VOB を使用できることを確認します。

# ClearCase と Windows ドメイン

# B

Rational ClearCase は、オペレーティング システムを利用して、VOB とビュー内のバージョン付き成果物を操作するためのユーザー権限を決定するユーザー ID とグループ メンバシップの詳細を提供します。Windows コンピュータでは、ユーザーとグループの識別情報は、ユーザーが Windows NT または Active Directory ドメインにログオンしたときに作成されます。この章では、ドメイン管理者が理解しておく必要がある、ClearCase が必要とするユーザー、グループ、リソース アカウントと、ClearCase でのドメイン信頼関係の影響について説明します。

この付録は、Windows NT ドメイン、Active Directory ドメイン、またはその両方の使用経験があり、Microsoft のドメイン管理ツールとアカウント メンテナンス ツールを熟知しているドメイン管理者を対象にしています。

## ClearCase と互換性のあるドメイン構成

---

ClearCase は、さまざまなドメイン構成との互換性があります。すべてのユーザー、グループ、コンピュータが 1 つのドメインに含まれている最も単純な構成では、管理の手間が必要最小限で済みます。ただし、次のようなほかの一般的な構成でも正しく機能します。

- ユーザーとグループのアカウントがマスター ドメインに作成され、マスター ドメインを信頼する 1 つまたは複数のリソース ドメインにホスト (コンピュータ) アカウントが作成されているマスターとマルチ マスター Windows NT ドメイン構成。
- 1 つのフォレストの一部である Active Directory ドメイン。
- Windows NT ドメインと Active Directory ドメインの組み合わせをサポートしているドメイン アップグレード環境とドメイン移行環境。

## ClearCase のドメインに対する要件

ClearCase と ClearCase LT には、それらが動作するドメイン環境に対するいくつかの要件があります。

- VOB とビューの共通セットにアクセスするすべてのユーザーは、ドメイン アカウントを持っている必要があり、共通するグループ メンバシップを少なくとも 1 つ持っている必要があります。

- すべてのクライアントとサーバー ホストは、ドメインのメンバでなければなりません。ホストが、ユーザーとグループ アカウントを作成したのと同じドメインのメンバでない場合、それらはそのドメインを信頼するドメインのメンバでなければなりません。
- 複数のドメインのユーザーが VOB とビューの共通セットにアクセスできるようにするには、追加の処理を実行する必要があります。

ClearCase では、`albd_server` プロセスのユーザー識別情報を提供する特別なドメイン アカウントも必要となります (ClearCase LT では必要ありません)。

**メモ:** ClearCase または ClearCase LT コミュニティのメンバが UNIX ホスト上の VOB とビューにアクセスする場合、それらのドメイン アカウントのユーザー名とグループ名は UNIX アカウント データベースのユーザー名とグループ名と同じでなければなりません。詳細については、59 ページの「共通なユーザー名とグループ名」を参照してください。

## 非ドメイン ホスト上の ClearCase

ドメインに含まれていない Windows コンピュータは ClearCase ホストになることができますが、その機能は非常に制限されます。

- これらは、ほかの ClearCase ホストが使用する VOB やビューをホストできません。
- これらは、ほかの Windows コンピュータ上で稼動している VOB やビューにアクセスできません。

非ドメイン システムは、すべての VOB とビューの記憶領域が UNIX ホストに存在するネットワーク上で ClearCase クライアントとして機能できます。

また、非ドメイン システムは、すべての VOB とビューがローカルに使用されるスタンドアロン システムとして機能することもできます。ClearCase ソフトウェアの評価時は、通常、この状態になります。

## ドメイン ユーザーとグループ アカウント

---

ClearCase コミュニティは、VOB とビューの共通セットを使用して定期的に ClearCase 操作を実行するすべてのユーザーが属するグループを定義する必要があります。本書では、このグループのことを ClearCase ユーザー グループと呼びます。このグループには、既存のドメイン グローバル グループまたはこの目的用に特別に作成したグループを使用できます。本書の例では、ClearCase ユーザー グループ名は `clearusers` になっています。

ClearCase ユーザー グループは、次の要件を満たしている必要があります。

- ドメイン グローバル グループまたは Active Directory ユニバーサル グループでなければなりません。グループにほかのグループを追加する必要がある場合を除いて、ドメイン グローバル グループを使用することをお勧めします。
- グループのメンバが UNIX ホスト上の VOB またはビューにアクセスする必要がある場合は、グループ名と UNIX 上の ClearCase ユーザー グループ名が同じでなければなりません。詳細については、59 ページの「共通なユーザー名とグループ名」を参照してください。

**メモ:** ClearCase コミュニティに、VOB とビューをメンバ間では共有するが、ほかのグループのメンバとは共有しない複数のグループが含まれている場合は、それらの各グループに異なる ClearCase ユーザー グループを指定する必要があります。

ClearCase コミュニティでは、ClearCase ユーザー グループのほかに、次の 2 つの追加ドメイン アカウントも必要になります (ClearCase LT コミュニティでは必要ありません)。

- ClearCase サーバー プロセス ユーザー: `albd_server` プログラムは、この識別情報で実行されます。本書の例では、ClearCase サーバー プロセス ユーザー名は `clearcase_albd` になっています。ClearCase サーバー プロセス ユーザーは、ClearCase 管理者グループのメンバでなければなりません。
- ClearCase 管理者グループ: このグループのメンバは、データの完全な破棄を含む、ClearCase のすべての操作を実行できます。このグループ内のメンバシップは、ClearCase サーバー プロセス ユーザーと数名の ClearCase 管理者に限定する必要があります。

**メモ:** ClearCase LT では、ClearCase サーバー プロセス ユーザーと ClearCase 管理者グループは必要ありません。ClearCase LT は、サーバー プロセスに組み込み `LocalSystem` アカウントを使用し、ClearCase LT サーバーのすべての管理者に ClearCase 管理権限を与えます。

ClearCase サーバー プロセス ユーザーと ClearCase 管理者グループが存在しない場合、通常、これらはサイト準備プロセスで作成されます。また、これらは必要に応じて手動で作成することもできます。338 ページの「必須のドメイン アカウントの手動定義」を参照してください。

## ClearCase プライマリ グループの設定

ユーザーのプライマリ グループは、Windows の各種ドメイン アカウント メンテナンス ツールを使用して指定できますが、アプリケーションがユーザーのプライマリ グループ名を要求したときにそのグループ名が返されないことがあります (ただし、ユーザーが 1 つのドメイン グループのメンバである場合は、常にこのグループ名が返されます)。

複数のドメイン グループのメンバである ClearCase ユーザーは、ユーザー環境変数 `CLEARCASE_PRIMARY_GROUP` を ClearCase ユーザー グループのドメインで修飾された名前に設定する必要があります (たとえば、`NT_WEST¥clearusers`)。これにより、ClearCase がユーザーのプライマリ グループであると見なすグループが明確に定義されます。1 つのドメイン グループのメンバである ClearCase ユーザーは、この処理を実行する必要はありません。

**メモ:** CLEARCASE\_PRIMARY\_GROUP は、ユーザー環境変数でなければなりません。システム環境変数にすることはできません。

CLEARCASE\_PRIMARY\_GROUP の値は、VOB にアクセスするためのユーザーの権限を評価するときにだけ使用されます。その他のセキュリティやアクセス コントロールには影響を与えません。複数のグループのメンバになっているユーザーの CLEARCASE\_PRIMARY\_GROUP が正しく設定されていないと、特に複雑なドメイン構成で、エレメントの作成や VOB へのアクセスで問題が発生することがあります。

**メモ:** ClearCase ユーザー グループと ClearCase 管理者グループの両方のメンバになっているユーザーは、CLEARCASE\_PRIMARY\_GROUP を ClearCase ユーザー グループ名 (ClearCase 管理者グループ名ではなく) に設定する必要があります。

CLEARCASE\_PRIMARY\_GROUP が正しく設定されていることを確認するには

- 1 Windows の [スタート] メニューから、[プログラム]、[Rational Software]、[ClearCase]、[管理] をポイントし、[ClearCase ドクター] をクリックします。
- 2 [分析開始] をクリックします。
- 3 分析が完了したら [トピック] タブをクリックして、User Login Account フォルダを開きます。
- 4 [プライマリ グループ] をダブルクリックして、ユーザーのプライマリ グループが正しいことを確認します。

## 必須のドメイン アカウントの手動定義

**注意:** ここでの説明は、ClearCase LT には当てはまりません。

ClearCase Site Preparation ウィザードは、ClearCase 管理者グループと ClearCase サーバー プロセス ユーザーのドメイン アカウントが存在しない場合、それらを作成しようとします。サイト準備中に指定したアカウント名とグループ名は、ユーザーが各ホストで ClearCase インストール プログラムを実行したときにデフォルト名として表示されます。

ClearCase Site Preparation ウィザードを実行しているユーザーがドメイン管理者権限を持っていない場合、ウィザードはこれらのアカウントを作成できません。この場合は、ドメイン管理者が次の手順に従ってこれらのアカウントを作成する必要があります。

- 1 ドメイン管理者としてログオンします。
- 2 適切なドメイン ユーザーとグループ アカウント管理ツールを使用して、clearcase という名前 (または、コミュニティが選択した別のグループ名) の新しいグローバルグループを作成します。[説明] ボックスに、次のテキストを入力します。

Used exclusively by ClearCase server processes and administrators.

- 3 新しいユーザー `clearcase_albd` (または、コミュニティが選択した別のグループ名) を作成し、手順 2 で定義したグループ名を新しいユーザーのグループ リストに登録します。

- [パスワードを無期限にする] チェック ボックスをオンにします。

- [説明] ボックスに、次のテキストを入力します。

Used exclusively by ClearCase servers

- ローカルの VOB とビューをサポートするように設定されている各 ClearCase ホストで、`clearcase_albd` ユーザーにサービスとしてログオンする権利を与えます。

## マルチ ユーザー アカウント ドメインのサポート

---

既存のドメイン構成が必要とする場合は、複数のドメインにわたる ClearCase からのユーザーとコンピュータに対するアクセスを有効にできます。

### Active Directory ユニバーサル グループの使用

Microsoft Active Directory 環境で動作している ClearCase コミュニティに、同じフォレストに属している複数の Active Directory ドメインのユーザーが含まれている場合は、Active Directory ユニバーサル グループを使用して、異なるドメインにログオンしているユーザーに VOB とビューの共通セットへのアクセス権を与えることができます。

1 つのフォレスト内の複数の Active Directory ドメインのユーザーが ClearCase プライマリ グループとして使用可能な Active Directory ユニバーサル グループを作成するには

- 1 Active Directory 環境がネイティブ モードで動作していることを確認します (ユニバーサル グループは、混合モードで動作している Active Directory ドメインでは作成できません)。
- 2 ClearCase ユーザー グループを Active Directory ユニバーサル グループとして作成します。
- 3 メンバが ClearCase コミュニティの一部である各ドメイン グローバル グループを、ClearCase ユーザー グループのメンバにします。個別のユーザー アカウントをユニバーサル グループに追加することはお勧めできません。その代わりに、各 Active Directory ドメインのユーザーを、そのドメインで定義されているドメイン グローバル グループにグループ化し、それらの各グループをユニバーサル グループのメンバにしてください。
- 4 グループに属している ClearCase ユーザーに、`CLEARCASE_PRIMARY_GROUP` を、(ユニバーサル) ClearCase ユーザー グループのドメインで修飾された名前に設定するように要求します (Active Directory アカウント管理ツールを使用して、ユニバーサル グループをユーザーのプライマリ グループとして指定することはできません)。

**メモ:** マルチマスター Windows NT ドメイン環境を Active Directory にアップグレードしている場合は、346 ページの「プロキシ グループの変換」の処理を実行して、プロキシ グループを Active Directory ユニバーサル グループのメンバに変換します。

## Windows NT ドメインでのプロキシ グループとドメイン マッピングの使用法

ClearCase コミュニティに複数の Windows NT ドメインのユーザーが含まれている場合は、この項の説明に従って ClearCase ドメイン マッピング機能を有効にし、すべてのユーザーが VOB の共通セットにアクセスできるようにする必要があります。この構成はセットアップと管理が複雑なので、組織全体またはセキュリティに関する設定が必要な場合以外には使用しないことをお勧めします。

**メモ:** プロキシ グループ内のユーザーが Windows 上で動的ビューを共有する場合は、ビューでアクセスされるすべてのディレクトリ エレメントのモードが 777 (すべてのユーザーに書き込み権限を与える) になっている必要があります。

ClearCase ユーザーが ATLANTA、BOSTON、CHICAGO という名前のドメインにアカウントを持ち、これらが共有する各 VOB のプライマリ グループが ATLANTA¥clearusers だとします。この環境で ClearCase を使用するには、プロキシ グループを作成して、次の手順でドメイン マッピング機能を有効にします。

- 1 各 ClearCase ホストが、ATLANTA、BOSTON、CHICAGO ドメインを信頼しているリソース ドメインのメンバであることを確認します。
- 2 いずれかのユーザー アカウント ドメインで ClearCase ユーザー グループを作成します。この例では、ドメインは ATLANTA で、グループは ATLANTA¥clearusers です。ドメイン マッピングを使用するユーザーが共有する VOB は、ATLANTA¥clearusers グループが所有している必要があります。
- 3 それらの各ドメイン内の各 ClearCase ホストで albd\_server を設定し、プライマリ ClearCase ドメイン (この場合は、ATLANTA¥clearcase\_albd) の clearcase\_albd ユーザーとしてログオンします。
- 4 ほかの 2 つの各ドメインに 1 つずつ、ドメイン グローバル グループをさらに 2 つ作成します。
  - BOSTON ドメインで、グループ BOSTON¥clearusers\_Boston を作成します。
  - CHICAGO ドメインで、グループ CHICAGO¥clearusers\_Chicago を作成します。

この 2 つのグループを作成するとき、説明として次の文字列を含めます。

ClearCaseGroup(ATLANTA¥clearusers)

この文字列には、大文字/小文字が正しく区別されていてスペースが含まれていないことを確認してください。このテキスト文字列がグループの説明内にあるとき、ClearCase はこのグループを、カッコで区切られた名前のグループに対するプロキシ グループとして認識します (この例では、ATLANTA¥clearusers)。VOB アクセス権を評価するとき、ClearCase はプロキシ グループのメンバを、ClearCaseGroup サブストリングで命名されたグループのメンバであるかのように扱います。この例では、BOSTON¥clearusers\_Boston の説明に ClearCaseGroup(ATLANTA¥clearusers) が含まれている場合、BOSTON¥clearusers\_Boston のメンバは ATLANTA¥clearusers のメンバと同じ VOB アクセス権を取得します。

- 5 ClearCase ユーザーを次のように適切なグループのメンバにします。
  - ドメイン ATLANTA にアカウントを持つユーザーを ATLANTA¥clearusers のメンバにします。
  - ドメイン BOSTON にアカウントを持つユーザーを BOSTON¥clearusers\_Boston のメンバにします。
  - ドメイン CHICAGO にアカウントを持つユーザーを CHICAGO¥clearusers\_Chicago のメンバにします。
- 6 各 ClearCase ホスト上でドメイン マッピングを有効にします。ドメイン マッピングを有効にするには、そのホストで Windows レジストリを編集し、次の変更を行います。
  - a Windows レジストリ エディタを使用して  
HKEY\_LOCAL\_MACHINE¥SOFTWARE¥Atria¥ClearCase¥CurrentVersion に移動します。
  - b [編集] メニューの [値の追加] をクリックします。
  - c [値の追加] ダイアログ ボックスで、[値の名前] に DomainMappingEnabled と入力し、  
[値タイプ] で REG\_DWORD を選択します。
  - d [OK] をクリックして DWORD エディタを開きます。
  - e DWORD エディタで、[データ] ボックスに「1」と入力します。
  - f [OK] をクリックして値を追加します。
- 7 ClearCase の各ユーザーに、ユーザー環境変数 CLEARCASE\_PRIMARY\_GROUP を  
Atlanta¥clearusers の値に設定するように要求します。337 ページの「ClearCase  
プライマリ グループの設定」を参照してください。

## VOB エレメントの権限の設定

プロキシ グループのメンバであるユーザーがアクセスする VOB のすべてのエレメントには、**その他**に対する**読み取り**権限を与える必要があります。新しく作成するエレメントにはデフォルトでこのアクセス権が与えられます。エレメントの保護状況を確認するには、**cleartool describe** を使用します。エレメントの保護状況を変更するには、**cleartool protect** を使用します。また、ClearCase エクスプローラなどの GUI を使用して、エレメントの保護状況を確認、変更できます。

## VOB 記憶領域の ACL の設定

必要に応じて、グローバルに読み取りアクセス可能なエレメントを特定ユーザーだけに制限することができます。この制限を設定するには、VOB 記憶ディレクトリを含む共有のアクセス コントロール リスト (ACL) を設定します。たとえば、グローバル パス  
¥¥myserver¥vobstorage¥src\_vob で登録されている VOB の場合、vobstorage 共有の ACL  
を変更して、ATLANTA¥clearusers、BOSTON¥clearusers\_Boston、  
CHICAGO¥clearusers\_Chicago ドメイン グループと ClearCase 管理者グループのメンバだけがアクセスできるように設定できます。

## Active Directory への変換

---

Windows ネットワークのエンタープライズ規模のアプリケーションと同様、ネットワークを Windows NT ドメインから Active Directory ドメインに変換すると ClearCase が影響を受けます。この付録の以降の項では、この変換が ClearCase と ClearCase LT に与える影響と、変換中や変換後にユーザー、グループ、ホスト、データを管理する方法について説明します。

**メモ:** 既に Active Directory を実行している環境で ClearCase または ClearCase LT を使用する場合は、この情報は無視してください。

### Active Directory の理解

Microsoft では、Windows ネットワークを Windows NT ドメインから Active Directory に変換するためのツールとマニュアルを提供しています。この項では、Microsoft が提供している該当するマニュアルを読み終え、そこで使用されている用語や説明されている手順を熟知していることを前提としています。特に、Microsoft のホワイト ペーパー "Planning Migration from Microsoft Windows NT to Microsoft Windows 2000" (これは、Windows 2000 サポート ツールの一部として配布されています。また、Microsoft の Web サイトにも掲載されています) を読み終えたことを前提としています。このホワイト ペーパーと関連ドキュメントでは、ネイティブ モード、混合モード、ドメインのアップグレード、ドメインの移行、SID 履歴、プリンシパルのクローン化など、いくつかの主要な概念について説明しています。

### Active Directory が ClearCase に与える影響

Active Directory 環境では、ユーザーとグループのいくつかの詳細識別情報は、Windows NT ドメイン環境とは異なる方法で処理されます。Windows NT ドメイン環境がどのように構成されているか、ClearCase ユーザーとグループのアカウントがこのドメイン構造のどこに存在しているか、どのように Windows NT ドメインを Active Directory ドメインに変換しようとしているかなどにより、ClearCase の制御下にある成果物へのユーザー アクセスを維持するための処理を、変換中や変換後に実行しなければならないことがあります。

Active Directory への移行は、ClearCase に次の影響を与えます。

- Active Directory では、ドメイン間の信頼関係は Windows NT ドメインとは異なる方法で作成、管理されます。Active Directory への変換中や変換後、これらの違いは、複数のドメインのユーザーが VOB とビューの共通セットにアクセスするのに使用する ClearCase コミュニティに影響を与えます。
- 変換シナリオによっては、ユーザーとグループの Windows セキュリティ識別子 (SID) が変更されることがあります。ClearCase は SID を VOB データベースに保存するため (オブジェクトの所有者を表すために)、このような場合は VOB を新しい SID で更新する必要があります。

一般的に、最も単純なドメイン構造のサイト (すべての ClearCase ユーザーとホストが 1 つのドメインに含まれているサイト) の場合、変換処理中に問題が発生することはほとんどありません。複雑なドメイン構造のサイト (複数のドメインのユーザーが VOB とビューの共通セットにアクセスするサイト) の場合は、ユーザーとグループの既存のアカウント情報のいくつかを変更することで、Active Directory の優れたドメイン間セキュリティ機能を活用できるようになります。

## Active Directory へのアップグレードまたは移行の計画

Microsoft では、ドメインを Windows NT から Active Directory に変換するためのツールとマニュアルを提供しています。変換は、次のいずれかの方法で実行します。

- アップグレード (インプレース アップグレードとも呼ばれます): この場合、Windows NT ドメイン コントローラは、混合モードまたはネイティブ モードで動作する Active Directory ドメイン コントローラに変換されます。アップグレード後、すべてのユーザー、グループ、リソースの SID は、元の Windows NT ドメインの場合と同じになります。
- 移行: この場合、ユーザー、グループ、リソースのアカウントが、クローン化と呼ばれる処理を使用して Windows NT ドメインから Active Directory ドメインに移行されます。移行完了後、すべてのユーザー、グループ、リソースの SID は新しくなります。ネイティブ モードの Active Directory は、各プリンシパルの現在と以前の SID に関する情報を維持するため (このことを Microsoft では、プリンシパルの SID 履歴と呼んでいます)、必要に応じて両方のタイプのドメインを一緒に使用することができます。

ClearCase 管理者は、この章と Microsoft の該当するドキュメントを読み終え、ClearCase のさまざまな変換または移行方式による影響を理解したら、Windows NT ドメインから Active Directory への変換方式を再確認し、可能な場合は計画時に役立てることをお勧めします。

**注意:** Microsoft では、NTFS ACL に格納されている SID を変換するためのツールを提供しています。これらのツール (またはネイティブ ファイル システムの保護情報を変更するあらゆるツール) は、VOB またはビュー記憶ディレクトリで使用しないでください。VOB またはビュー記憶ディレクトリの SID の変換には、ClearCase ユーティリティだけを使用してください。詳細については、352 ページの「vob\_sidwalk を使用した VOB ユーザーとグループの変更または更新」を参照してください。

## ClearCase ホストの準備

変換を開始する前に、Active Directory 環境で使用するように ClearCase ホストを設定する必要があります。

- Windows ホスト上のすべての VOB は、スキーマ バージョン 54 でなければなりません。スキーマ バージョン 54 は、Active Directory の優れたユーザー / グループ認証処理機能に対応できるように、Windows のユーザーとグループの識別情報を SID 形式で格納します。

- すべてのユーザーに対するユーザー環境変数 `CLEARCASE_PRIMARY_GROUP` を定義する必要があります。この変数の値は、`DOMAIN_NAME\group_name` の形式のドメインで修飾されたグループ名にすることをお勧めします。

**メモ:** ホストを Rational ClearCase バージョン 4.2 からアップグレードする場合は、ホスト上の各ビューの形式を変更する必要があります。ClearCase バージョン 4.2 からそれ以降のバージョンにアップグレードしたホストでは、ビューは初めて起動されたときに自動的に形式を変更されます。また、`reformatview` コマンドを使用してビューを手動で形式を変更することもできます。

Active Directory への変換を実行する前に、すべての ClearCase ホストがこの項の説明に従って設定され、ClearCase がすべてのユーザーとホストに対して正常に機能していることを確認してください。

## ドメイン アップグレード シナリオ

---

この項では、Microsoft のインプレース アップグレード手順を使用して、1 つまたは複数の Windows NT ドメインを Active Directory ドメインにアップグレードするいくつかのシナリオについて説明します。サイトでこの手順を使用しない場合は、347 ページの「ドメイン移行シナリオ」を参照してください。

**メモ:** この項では、ドメインのアップグレードとは、ドメインのプライマリ ドメイン コントローラのアップグレードを意味します。

### 1 つのドメインのアップグレード

この手順は、すべての ClearCase ユーザー、グループ、ホストが 1 つの Windows NT ドメインのメンバである場合に使用します。

- 1 343 ページの「ClearCase ホストの準備」の説明に従って ClearCase ホストを準備します。
- 2 アップグレード中にオンラインにすることができるバックアップ ドメイン コントローラがない場合は、アップグレード処理中に ClearCase 操作が実行されないようにすべての ClearCase サーバー ホストで ClearCase を停止します。
- 3 Microsoft の手順を使用して、Windows NT ドメインを Active Directory ドメインにインプレース アップグレードします。
- 4 プライマリ ドメイン コントローラのアップグレードが完了したら、すべての ClearCase ホストをシャットダウンして再起動します。

## マスター ドメインとそのリソース ドメインのアップグレード

この手順は、すべての ClearCase ユーザーとグループが、ClearCase ホストが属している 1 つまたは複数の Windows NT リソース ドメインによって信頼されている 1 つの Windows NT ドメイン (マスター ドメイン) のメンバである場合に使用します。

マスター ドメインとそのリソース ドメインをアップグレードするには

- 1 344 ページの「1 つのドメインのアップグレード」の説明に従って、マスター ドメインをアップグレードします。
- 2 344 ページの「1 つのドメインのアップグレード」の説明に従って、各リソース ドメインをアップグレードします。アップグレードしたリソース ドメインを、アップグレードしたマスター ドメインの子として構成します。
- 3 リソース ドメインのアップグレードが完了したら、そのリソース ドメイン内のすべての ClearCase ホストをシャットダウンして再起動します。

## 複数のマスター ドメインとリソース ドメインのアップグレード

この手順は、ClearCase ユーザーとグループが複数の Windows NT ドメインのメンバで、これらのドメインのユーザーが VOB とビューの共通セットにアクセスできるようにするのにプロキシグループを使用する場合に使用します。ドメインには、ユーザーやグループのほかにリソースを含めたり、リソース ドメインによって信頼されたマスター ドメインにすることができます。

アップグレードが完了したら、Active Directory ドメインをネイティブ モードに変換することをお勧めします。これにより Active Directory ユニバーサル グループを使用できるようになり、プロキシグループとドメイン マッピングを使用しなくて済むようになります。

複数のマスター ドメインとリソース ドメインをアップグレードするには

- 1 343 ページの「ClearCase ホストの準備」の説明に従って ClearCase ホストを準備します。
- 2 アップグレード中にオンラインにすることができる各マスター ドメインとリソース ドメイン用のバックアップ ドメイン コントローラがない場合は、アップグレード処理中に ClearCase 操作が実行されないようにすべてのサーバー ホストで ClearCase を停止します。
- 3 Microsoft の手順を使用して、最初の Windows NT マスター ドメインを Active Directory ドメインにインプレース アップグレードします。ClearCase ユーザー グループが定義されているドメインをアップグレードしてから、プロキシグループが定義されているドメインをアップグレードします。
- 4 残りのマスター ドメインをアップグレードします。

- 5 マスター ドメインのアップグレードが完了すれば、リソース ドメインのアップグレードを開始できます。アップグレードしたドメインとアップグレードしていないドメイン間の既存の信頼関係を変更しない限り、稼動環境に適した任意の順番とスケジュールでリソース ドメインをアップグレードできます。アップグレードしたすべてのドメインを同じフォレストのメンバにすることをお勧めします。これにより、339 ページの「Active Directory ユニバーサル グループの使用」で説明しているように、Active Directory ユニバーサル グループを使用できるようになります。
- 6 リソース ドメインのアップグレードが完了したら、そのリソース ドメイン内のすべての ClearCase ホストをシャットダウンして再起動します。
- 7 すべてのドメインをアップグレードし、Active Directory ドメインをネイティブ モードに変換したら、346 ページの「プロキシ グループの変換」の手順に従います。これにより、プロキシ グループとドメイン マッピングを使用しなくて済むようになります。

## プロキシ グループの変換

Windows NT ドメイン環境で必要なプロキシ グループを、Active Directory のユニバーサル グループである ClearCase ユーザー グループに置き換えるには、以下の処理を実行します。ユニバーサル グループには、以前プロキシ グループとして使用されていたグループも含まれます。

**メモ:** この手順は、ネイティブ モードで動作している Active Directory ドメインでだけ実行できます。ClearCase ユーザー グループとすべてのプロキシ グループは、同じフォレストに存在している必要があります。

- 1 Active Directory 管理ツールを使用して、プライマリ ClearCase ドメイン内の ClearCase ユーザー グループの名前を変更します。たとえば、340 ページの「Windows NT ドメインでのプロキシ グループとドメイン マッピングの使用法」の手順で作成した ATLANTA¥clearusers グループの名前を ATLANTA¥clearusers\_Atlanta に変更します。
- 2 Active Directory 管理ツールを使用して、新しい (ユニバーサル) ClearCase ユーザー グループを作成します。各ユーザーが CLEARCASE\_PRIMARY\_GROUP 環境変数の値を変更しないで済むように、同じ名前 (この例では、ATLANTA¥clearusers) を使用することをお勧めします。
- 3 Active Directory 管理ツールを使用して、以前のプロキシ グループ (この例では、BOSTON¥clearusers\_Boston と CHICAGO¥clearusers\_Chicago) と手順 1 で名前を変更したグループを新しい (ユニバーサル) ClearCase ユーザー グループとして追加します。

- 4 すべての ClearCase クライアントとサーバー ホスト上のドメイン マッピングを無効にします。ClearCase ホスト上のドメイン マッピングを無効にするには、レジストリ エディタを使用してレジストリ キー  
HKEY\_CURRENT\_USER¥SOFTWARE¥Atria¥ClearCase¥CurrentVersion に移動し、  
DWORD 値 DomainMappingEnabled を削除します (ClearCase ホストによっては、この値が HKEY\_LOCAL\_MACHINE¥SOFTWARE¥Atria¥ClearCase¥CurrentVersion に保存されていることもあります)。
- 5 すべての ClearCase クライアントとサーバー ホストをシャットダウンして再起動します。

## ドメイン移行シナリオ

---

ドメインをアップグレードできないサイトでは、ドメイン移行処理を実行します。この処理では、Microsoft ツールを使用して、Windows NT ドメインの既存のアカウントからクローン化したユーザー、グループ、リソース アカウントをネイティブモードの Active Directory ドメインに移行します。両方のタイプのドメインを一緒に動作させることができ、Windows NT と Active Directory ドメイン間で適切な信頼関係が存在している場合は、両方のタイプのドメイン内のユーザーとグループに同等の ClearCase アクセス権が与えられます。

移行は、必要に応じて長期間に渡って実施することもできます。すべてのユーザーとグループ アカウントを Active Directory ドメインに移行したら、移行処理を完了して、Windows NT ドメインの使用を停止できます。移行後、すべてのユーザー、グループ、リソースの SID は新しくなります。これは、ClearCase に次の影響を与えます。

- ClearCase ホストが Active Directory ドメインのメンバになったら、Active Directory ドメインに存在する ClearCase サーバー プロセス ユーザー アカウントとしてログオンするようにホストの albd\_server を再設定する必要があります。アカウント名は変更されませんが、ユーザー SID とグループ SID は変更されています。
- VOB 記憶ディレクトリに対して、VOB 所有者の新しい SID をディレクトリ ACL に含むように保護を再設定する必要があります。
- VOB データベースは、新しい SID で更新する必要があります。ClearCase ユーティリティ プログラム vob\_sidwalk は、古い SID を新しい SID で置き換えます。詳細については、352 ページの「vob\_sidwalk を使用した VOB ユーザーとグループの変更または更新」と vob\_sidwalk リファレンス ページを参照してください。

これらのタスクを実行するための詳細手順については、この項で説明しています。

## 複数のドメインの移行

すべての移行シナリオは基本的に同じですが、移行するドメインの数とそれらのドメイン間の信頼関係によって複雑さが異なります。この項で説明する手順は、1 つのドメインまたは複数のマスターとリソース ドメインを移行する場合に使用できます。

**メモ:** 移行処理を実行する前に、343 ページの「ClearCase ホストの準備」の説明に従ってすべての ClearCase ホストを準備してください。ホストをすぐに移行しない場合でも、この処理を実行しないと、移行済みのユーザーとグループがホストにアクセスできなくなります。

## ユーザーとグループの移行

最初に、ユーザーとグループを、それらが作成された Windows NT ドメインから新しい Active Directory ドメインに移行することをお勧めします。ClearCase ユーザーとグループを移行するには、次の手順に従います。

- 1 Microsoft が定義した処理手順を使用して、ユーザーとグループを Windows NT ドメインから Active Directory ドメインに移行します。必ず、ClearCase ユーザー グループ (存在する場合)、ClearCase 管理者グループ、clearcase\_albd アカウントを移行対象に含めてください。
- 2 移行シナリオの多くには、Active Directory ドメインにログオンしているユーザーが、Windows NT ドメインにログオンしているユーザーと同じ VOB にアクセスする期間があります。両方のタイプのドメインのユーザーが VOB にアクセスできるようにするには、次のいずれかの処理を実行します。
  - Active Directory ドメインに移行した ClearCase ユーザー グループのドメインで修飾された名前を VOB の補足グループ リストに追加します。たとえば、次で示すように cleartool protectvob コマンドを使用して、Active Directory ドメイン AD-DOMAIN の clearusers グループを、VOB サーバー ホスト上の C:¥vobstg¥srcs.vbs にある記憶領域と共に VOB のグループ リストに追加できます。

```
cleartool protectvob add_group AD-DOMAIN¥clearusers C:¥vobstg¥srcs.vbs
```

- Active Directory ドメインにログオンしているユーザーに、CLEARCASE\_PRIMARY\_GROUP 環境変数を、Windows NT ドメイン内の ClearCase ユーザー グループの SID の文字列表現に設定するように要求します。SID 文字列を検索するには、Windows NT ドメインのメンバ、または Windows NT ドメインを信頼しているドメインのメンバであるコンピュータで creds コマンドを実行します。次に例を示します。

```
ccase-home-dir¥etc¥utils¥creds -g NT-DOMAIN¥clearusers
.
.
.
ClearCase group info:
Name: NT-DOMAIN¥clearusers
```

GID: 0x100423

SID credentials NT:S-1-5-21-103034363-981818062-1465874335-1064

この場合、ユーザーは CLEARCASE\_PRIMARY\_GROUP を

NT:S-1-5-21-103034363-981818062-1465874335-1064 に設定します。

- 3 移行したアカウントには SID 履歴が含まれているため、Active Directory ドメイン内のユーザー アカウントには、Windows NT ドメインの 2 倍の数のグループ メンバシップが含まれます (各ユーザー グループ リストには、両方のドメインのグループが含まれます)。Windows NT ドメイン内の複数のグループのメンバであり、移行後に 32 を超えるグループがグループ リストに含まれているユーザーは、Active Directory ドメイン内の ClearCase ユーザー グループの名前と Windows NT ドメイン内の ClearCase ユーザー グループを表す SID 文字列を含むように CLEARCASE\_GROUPS 環境変数を設定する必要があります。例を次に示します。

CLEARCASE\_GROUPS=AD-DOMAIN¥clearusers;NT:S-1-5-21-103034363-981818062-1465874335-1064

手順 2 に、creds を使用して SID 文字列を取得する方法を示します。CLEARCASE\_GROUPS 環境変数の詳細については、74 ページの「ユーザーが 33 以上のグループに所属している場合の制限」を参照してください。

## 移行中に新しいユーザーを追加しなければならない場合

ドメインの移行中に新しい ClearCase ユーザーを追加しなければならない場合は、次のいずれかの処理を実行します。

- Active Directory ドメインにユーザー アカウントを作成して、CLEARCASE\_PRIMARY\_GROUP 環境変数を、クローン化され、Active Directory ドメインに存在する ClearCase ユーザー グループのドメインの修飾がある名前に設定するようにユーザーに要求します。
- Windows NT ドメインでユーザーを作成し、その後ユーザー アカウントを移行します。

## 個別のホストの移行

すべてのユーザーとグループの移行が完了し、Windows NT ドメインにログオンする必要があるユーザーが存在しなくなったら、次の処理を実行して、ClearCase ホストを Active Directory ドメインに移行します。すべての ClearCase ホストを同時に移行できない場合は、最初に VOB サーバーを移行することをお勧めします (レジストリ サーバーとライセンス サーバーは、それらのデータベースに SID を格納しないため、いつでも移行できます)。

**メモ:** ここで説明する手順では、vob\_sidwalk ユーティリティを使用する必要があります。これらの処理を実行する前に、vob\_sidwalk リファレンス ページを参照して vob\_sidwalk の機能を理解してください。

個別の ClearCase ホストを移行するには

- 1 このホストの ClearCase を終了します。
- 2 Microsoft が定義した処理手順を使用して、ホストを Active Directory ドメインに移行します。
- 3 ClearCase ホスト (ClearCase LT も含む) でだけ次の処理を実行します。

- ホストの移行が完了したら、`albd_server` を再設定して、Active Directory ドメイン内の (移行済みの) ClearCase サーバー プロセス ユーザーとしてログオンします。これは、ホストに ClearCase を再インストールし、新しいアカウントを指定するか、338 ページの「必須のドメイン アカウントの手動定義」の手順に従って手動で実行することができます。
- ホストが、UNIX クライアントもアクセスする VOB サーバーの場合は、[コントロールパネル] の [ClearCase] をクリックし、[オプション] タブで認証マッピング ドメインをリセットします。認証マッピング ドメインは、そこに含まれているユーザーとグループ アカウント名が、このサーバー上の VOB にアクセスする UNIX ユーザーのユーザーとグループ アカウント名と一致するものでなければなりません。

- 4 このホストで ClearCase を再起動します。
- 5 この例で示すように、各 VOB またはビュー記憶ディレクトリで `fix_prot` ユーティリティを実行し、ホスト上の VOB とビュー記憶領域の保護を設定し直します。

```
ccase-home-dir¥etc¥utils¥fix_prot -replace storage-dir-pname
```

`storage-dir-pname` は、VOB またはビュー記憶ディレクトリへのパス名です。

- 6 クローン化されたユーザー アカウントとグループ アカウントを表す新しい SID で VOB データベースを更新します。
  - a VOB 所有者または権限を持つユーザーとして VOB サーバー ホストにログオンします。
  - b 自分以外のすべてのユーザーの VOB をロックします (`-nusers you`)。
  - c 古い SID を新しい SID で置き換えます。`vob_sidwalk` を実行します。この例では、`vob-tag` は更新している VOB の VOB タグで、`SIDfile-path` は `vob_sidwalk` が行った変更を記録するファイルの名前です。

```
ccase-home-dir¥etc¥utils¥vob_sidwalk -s -execute vob-tag SIDfile-path
```

`vob_sidwalk` が SID をマッピングし直す方法の詳細については、352 ページの「`vob_sidwalk` を使用した VOB ユーザーとグループの変更または更新」を参照してください。

- d VOB をアンロックします。

**メモ:** プロキシ グループを使用して、複数のドメインのユーザーが ClearCase 成果物の共通セットにアクセスできるようにしている場合は、プロキシ グループとドメイン マッピングを使用しなくて済むように Active Directory ユニバーサル グループを使用することをお勧めします。346 ページの「プロキシ グループの変換」の手順に従います。

## クライアントは移行できるが VOB サーバーを移行できない場合

VOB サーバーのクライアントは移行できるが VOB サーバーを移行できない場合に、そのサーバー上の VOB にクライアントが引き続きアクセスできるようにするには、`vob_siddump` を使用して新しい SID と古い SID 間のマッピングを作成する必要があります。マッピングを作成したら、`vob_sidwalk` を使用して、新しい SID で VOB データベースを更新します。

- 1 Active Directory ドメインに移行したクライアントにログオンします。
- 2 自分以外のすべてのユーザーの VOB をロックします (`-nusers you`)。
- 3 `vob_siddump` を実行してマップ ファイルを作成します (`vob_sidwalk` はリモート ホストからは実行できないため、`vob_siddump` を使用する必要があります)。この例では、`vob-tag` は、現在でも Windows NT リソース ドメイン内にあるサーバー上の VOB の VOB タグで、`SIDfile-path` は、`vob_siddump` が生成するマップ ファイルへのパス名です (VOB サーバーホストにアクセス可能なドライブ上に `SIDfile-path` を作成できない場合は、手順 4 を実行する前にそれを VOB サーバー ホストにコピーする必要があります)。

`vob_siddump -sidhistory vob-tag SIDfile-path`

- 4 手順 3 で使用したタグのある VOB をホストする VOB サーバーにログオンします。自分以外のユーザーに対して VOB がロックされている状態で、`vob_sidwalk` を実行して VOB に格納されている SID 情報を更新します。

`vob_sidwalk -execute -map mapfile-path vob-tag SIDfile-path`

この例では、`mapfile-path` は上記の手順 3 で生成したマップ ファイルです。また、`SIDfile-path` は `vob_sidwalk` が行った変更を記録するファイルの名前です。詳細については、352 ページの「`vob_sidwalk` を使用した VOB ユーザーとグループの変更または更新」を参照してください。

- 5 VOB をアンロックします。

**メモ:** 手順 3 を開始してから手順 4 を終了するまで VOB をロックしたままにしておかないと、ユーザーは処理の合間に新しいオブジェクトを作成することができます。ユーザーが新しいオブジェクトを作成した場合は、両方の処理を再度実行する必要があります。

## **vob\_sidwalk を使用した VOB ユーザーとグループの変更または更新**

VOB の元のホストが存在するドメインを信頼しないドメイン内のホストに VOB を移動した場合、VOB データベースに格納されているすべての SID は新しいドメイン内のアカウントに変換されないため、すべてが無効になります。この問題は、ドメインの移行時 (ホストが異なるドメインに移動し、VOB がホストに残る場合) に発生します。また、VOB が、あるドメイン内のホストから別のドメイン内のホストに移動した場合にも発生します。

**vob\_sidwalk** コマンドを使用して、VOB 内のオブジェクトの所有権の再割り当て、VOB のグループリスト内のグループを表している SID の更新、VOB 記憶ディレクトリの保護の修正を行うことができます。**vob\_sidwalk** の一般的な使用法は、次のとおりです。

- VOB を、Windows NT ドメインから Active Directory ドメインに移行する
- VOB を、元のドメインを信頼しないドメイン内のホストへ移動する
- VOB を、Windows ホストから UNIX ホストに移動する、またはその逆
- VOB サーバー ホストを、元のドメインを信頼しないドメイン内のホストへ移動する

この項では、**vob\_sidwalk** と **vob\_siddump** を使用する手順のいくつかの例について説明します。

**vob\_sidwalk** を使用する手順のその他の例については、第 8 章を参照してください。

**vob\_sidwalk** リファレンス ページでは、**vob\_sidwalk** と **vob\_siddump** のすべてのオプションの詳細について説明しています。

使用する手順にかかわらず、**vob\_siddump** (または、**-execute** オプションを指定しない **vob\_sidwalk**) を実行して出力をチェックし、VOB 内のどのオブジェクトの所有者が変更されたかを確認することをお勧めします。所有者の変更が正しいことを確認したら、**-execute** オプションを指定して **vob\_sidwalk** を実行し、SID を実際に再マッピングします。SID ファイルの出力は、カンマ区切り値 (.csv) 形式で書き出されます。この出力は、このファイル形式を読み込むことが可能なすべてのテキスト エディタやスプレッドシート プログラムで表示、変更できます。

### **ドメイン移行後の履歴 SID の再マッピング**

ドメイン移行シナリオでは、VOB データベースには、VOB 内にオブジェクトを所有しているセキュリティ プリンシパル (ユーザーとグループ) の SID 履歴を表す追加の SID が含まれています。これらの履歴 SID は、移行前にセキュリティ プリンシパルに関連付けられ、Active Directory ドメイン内のプリンシパルの **sidHistory** 属性に格納されます。

VOB データベースに格納されている履歴 SID を、Active Directory ドメイン内の適切なセキュリティ プリンシパルに変換される新しい SID で置き換えるには、次のようなコマンドを使用します。

```
vob_sidwalk -sidhistory -execute vob-tag SIDfile-path
```

-sidhistory オプションを指定して実行した場合、vob\_sidwalk は次のアルゴリズムを使用して SID 履歴を判断します。

- 1 SID を参照してアカウント名を見つけます。
- 2 手順 1 で見つかったアカウント名を参照して、その SID を見つけます。
- 3 手順 2 で返された SID が、手順 1 で使用されている SID と異なる場合、手順 1 で使用されている SID が履歴 SID と見なされ、手順 2 で返された SID が SIDfile-path の現在の行の新しい SID フィールドに書き込まれます。

## VOB を新しいドメインに移動するときに現在の SID を再マッピングする

VOB を別のドメインに移動する場合、vob\_sidwalk を使用して、VOB 内のオブジェクトを所有しているすべてのセキュリティ プリンシパルの新しい SID を取得して VOB データベース内に格納する必要があります。VOB を別のドメインのホストに移動する場合と、VOB サーバー ホストを別のドメインに移動して VOB をホストに残したままにする場合の手順は基本的に同じです。154 ページの「別のドメインへの VOB の移動」では、VOB の移動処理の一部として SID を再マッピングする方法について説明しています。

## VOB 所有者への所有権の再割り当て

既存のすべての SID を、VOB 所有者とグループの新しい SID にマッピングして、VOB 内のオブジェクトの所有権を再割り当てするには、次のようなコマンドを使用します。

vob\_sidwalk -unknown -execute vob-tag SIDfile-path

-unknown オプションと -execute オプションを指定して実行した場合、vob\_sidwalk は変換不能なユーザー SID を VOB 所有者の SID にマッピングし、変換不能なグループ SID を VOB のグループの SID にマッピングします。

**メモ:** UCM を使用する場合は、-unknown オプションを指定して所有権を再割り当てしないことをお勧めします。開いているアクティビティを VOB 所有者に再割り当てすると、アクティビティの作成者はそれを使用できなくなります (アクティビティの作成者が VOB 所有者の場合は、この限りではありません)。

## VOB 記憶ディレクトリ保護の再設定

VOB 記憶ディレクトリ ACL が移行中に (またはほかのイベントによって) 損傷した場合、次のように vob\_sidwalk を使用して、VOB 記憶ディレクトリの ACL とコンテナ ファイルを修復できます。

vob\_sidwalk -recover\_filesystem vob-tag SIDfile-path

-recover\_filesystem オプションを指定した場合、vob\_sidwalk は VOB の補足グループ リストの SID も修復します。

## 識別情報と権限を維持するレプリカでの `-delete_groups` の使用

識別情報/権限を維持したレプリカ (`mkreplica -preserve` を指定して作成) を使用する Rational ClearCase MultiSite ユーザーは、これらのレプリカのホストを Windows NT ドメインから Active Directory に移行する場合、いくつかの追加処理を実行する必要があります。

`vob_sidwalk` で行った SID の変更は、レプリカ作成処理では反映されないため、レプリカをホストするサーバーを Active Directory に移行するときに、レプリカ ファミリ内のそれぞれの識別情報/権限を維持したレプリカで `vob_sidwalk` を実行する必要があります。そのようなレプリカで `vob_sidwalk` を実行すると、VOB のグループ リスト上の元の SID が維持され、ファミリー内のそのようなレプリカがすべて更新されたかどうかにかかわらず、コンテナの作成を必要とする操作が正常に完了するようになります。レプリカ ファミリのそのようなメンバをすべて更新したら、管理者は `-delete_groups` オプションを指定して `vob_sidwalk` を再度実行し、これらの履歴グループ SID を削除する必要があります。VOB のグループ リストのグループ数の上限は 32 グループであるため、履歴 SID を削除することをお勧めします。使用していない履歴 SID をリスト内に保持しておく、新しいグループを追加したときにリストがオーバーフローすることがあります。

**メモ:** この手順では、すべてのレプリカのすべてのユーザーのユーザーとグループ アカウントを Active Directory に移行し、すべてのユーザーが各自の `CLEARCASE_PRIMARY_GROUP` 環境変数を Active Directory ドメイン内の ClearCase ユーザー グループの名前に設定したと仮定しています。

- 1 ファミリ内のすべての各レプリカの同期を取り、各レプリカにユーザーとグループ SID の同じセットが含まれるようにします。
- 2 349 ページの「個別のホストの移行」の手順に従って、ホストを移行します。ファミリー内のすべての識別情報/権限を維持したレプリカは、同じ `vob_sidwalk` オプションを使用して処理する必要があります。`-map` オプションを使用すると、1 つのマッピング ファイルを生成し、それをファミリー内のすべての識別情報/権限を維持したレプリカで使用して時間を節約することができます。
- 3 上の手順 2 に従って、SID が更新された別のレプリカで再度レプリカの同期をとった後、次のコマンドを実行します。

```
vob_sidwalk -sid_history vob-tag SIDfile-path
```

その後、結果の SID ファイルをチェックし、新しい SID マッピングを実行する必要があるかどうか (新しいユーザーまたはグループの識別情報がレプリカに追加されたことにより) を確認します。新しい SID マッピングが必要な場合は、手順 2 で使用したオプションを再び使用して `vob_sidwalk` を実行します。

- 4 すべての識別情報/権限を維持したレプリカを更新し (手順 2)、生成した SID ファイル (手順 3) が新しい SID マッピングを実行する必要があることを示している場合は、各レプリカで `vob_sidwalk -execute -delete_groups` を実行します。このコマンドは、VOB のグループ リストから履歴グループ SID を削除します。

# Rational Web Platform の 構成



Rational Web Platform (以下 RWP) は、Rational ClearCase など、さまざまな Rational 製品への Web インターフェイスのサーバー側でのサポートを可能にします。RWP はデフォルトの構成でインストールされますが、その構成はほとんどのサイトに適しています。一部のサイトでは、ホストまたはサイト固有のさまざまな要件を満たすために、インストール後に RWP の構成を変更する必要があります。たとえば、次のような操作を行います。

- RWP で別の HTTP ポート番号を使用できるようにする
- RWP のログのデフォルトを変更する
- プロキシとして機能する別の Web サーバーから RWP へのアクセスを設定する
- セキュア ソケットを使用するように RWP を構成する

この付録では、デフォルトの構成に対して一般的な変更のいくつかを行うために、RWP 構成ファイルを編集する方法について説明します。また、ClearCase Web インターフェイスの構成についても説明します。

Rational Web Platform では、Apache HTTP Server 2.x に基づく Web サーバーと、Tomcat サーブレット コンテナ バージョン 4.x に基づくサーブレット エンジンを採用しています。Apache HTTP Server に関する詳細情報は [www.apache.org](http://www.apache.org)、Tomcat サーブレット コンテナに関する詳細情報は [jakarta.apache.org](http://jakarta.apache.org) にアクセスして入手できます。

RWP は、UNIX では RWP servlet engine のインスタンスを常に 1 つ実行します。Windows では、必要に応じて、servlet engine の 2 つ目のインスタンスを作成します。

**メモ:** Rational Web Platform は、Rational 製品に対する Web インターフェイスのみをサポートします。ほかの Web アプリケーションまたはコンテンツに対する使用はサポートされていません。

## RWP のインストール ディレクトリ

---

RWP は、通常、次のディレクトリのいずれかにインストールされます。

- Windows の場合は、C:\Program Files\Rational\common\rwp。
- UNIX の場合は、/opt/rational/common/rwp。RWP が別の場所にインストールされる場合、インストーラはシンボリック リンク /opt/rational/common/rwp を作成します。これは、RWP のインストール ディレクトリを指します。

RWP のデフォルトのインストール ディレクトリは、インストール時に変更できます。それには、インストール プログラムからパスの入力を要求されたときに、異なるパスを入力します。

## RWP 構成ファイル

---

RWP の構成は、複数のファイルに指定されます。以下のファイルは、通常、RWP のインストール ディレクトリの **conf** サブディレクトリにインストールされます。

- **rwpc.conf**: RWP サーバーの構成パラメータを指定します。
- **ssl.conf**: RWP サーバーでセキュア ソケットを使用する場合に、その構成パラメータを指定します。
- **server.xml**: RWP servlet engine の構成パラメータを指定します。
- **server2.xml**: (Windows の場合) RWP ReqWeb servlet engine の構成パラメータを指定します。
- **workers.properties**: RWP と RWP servlet engine 間の接続用構成パラメータを指定します。

Windows 環境では、以下のファイルが、通常、RWP のインストール ディレクトリの **bin** サブディレクトリにインストールされます。

- **jk\_service.properties**: Windows サービスとして実行される RWP servlet engine を制御します。
- **jk\_service2.properties**: Windows サービスとして実行される RWP ReqWeb servlet engine を制御します。

任意のテキスト エディタを使用して、これらのファイルを編集できます。これらのファイルには、すべての構成パラメータに関する説明が記述されています。ここでは、変更が必要な可能性のあるいくつかのパラメータについて説明します。

**メモ:** これらのファイルで構成パラメータを変更した後、変更内容を反映させるには、RWP を停止して再起動する必要があります。361 ページの「RWP の停止と再起動」を参照してください。

### 構成ファイルの参照用バージョン

RWP のインストール ディレクトリには、すべての構成ファイルの参照用バージョンが格納されています。

- UNIX の場合
  - **rwpc.conf.template**
  - **server.xml.template**
- Windows の場合
  - **rwpc.in.conf**
  - **server.in.xml**
  - **server2.in.xml**

インストール プログラムは、これらの参照用バージョンを使用して、構成ファイルがカスタマイズされているかどうかを判断します。これらのファイルを変更しないでください。

## デフォルトの RWP HTTP ポートを変更するには

RWP が HTTP の要求をリッスン (受信を監視) するポートは、`rwp.conf` の `Listen` パラメータで定義されます。例を次に示します。

```
Listen 80
```

これは、RWP がポート 80 (HTTP のデフォルト) でリッスン (受信を監視) することを示しています。ここに、利用可能な任意のポート番号を指定することもできます。例を次に示します。

```
Listen 8000
```

これは、RWP がポート 8000 でリッスン (受信を監視) することを示しています。

**メモ:** RWP HTTP ポート番号を 80 以外に変更する場合、RWP を参照するすべての URL にそのポート番号を含める必要があります。例を次に示します。

```
http://RWP_host.domain:8000/ccweb
```

## デフォルトの RWP servlet engine ポートを変更するには

RWP servlet engine が RWP と通信するポートは、`workers.properties` ファイルだけでなく `server.xml` ファイルと `server2.xml` ファイルでも定義されます。表 15 に、デフォルトのポート番号、ポートの使用法、ポート番号が定義されているファイルを一覧表示します。

表 15 デフォルトの RWP servlet engine ポート

| ポート番号                   | 説明                                                        | 地域                                                         |
|-------------------------|-----------------------------------------------------------|------------------------------------------------------------|
| 8009 (HP-UX<br>では 8010) | RWP と RWP servlet engine の間の通信で<br>使用                     | <code>server.xml</code> 、 <code>workers.properties</code>  |
| 8010                    | RWP と RWP ReqWeb servlet engine の<br>間の通信で使用 (Windows のみ) | <code>server2.xml</code> 、 <code>workers.properties</code> |
| 8005 (HP-UX<br>では 8006) | RWP servlet engine のシャットダウン<br>ポート                        | <code>server.xml</code>                                    |

これらのポートを RWP ホスト上のほかのアプリケーションでも使用する場合は、別のポートを使用するようにそのアプリケーションを再構成することをお勧めします。再構成できない場合は、RWP が使用するポートを変更する必要があります。

次の `server.xml` の例は、ポート 8009 を、RWP と RWP servlet engine の間の内部通信に使用するポートとして定義しています。

```
<Connector className="org.apache ajp.tomcat4.Ajp13Connector"
    port="8009" minProcessors="5" maxProcessors="75"
    acceptCount="10" debug="0"/>
```

いずれかのポートを変更するには、該当する **Connector** エレメントの **port** 属性の値を変更します。たとえば、次の行では **port="8088"** 属性が指定されています。

```
<Connector className="org.apache.jsp.tomcat4.Ajp13Connector"
    port="8088" minProcessors="5" maxProcessors="75"
    acceptCount="10" debug="0"/>
```

これにより、RWP と **servlet engine** の間の内部通信にポート 8088 が使用されます。

**メモ:** **server.xml** 内の **Ajp13Connector** エレメントのポート属性を変更する場合は、**workers.properties** ファイルの次の行のポートも変更する必要があります。

```
worker.ajp13.port=8009
```

**server2.xml** 内の **Ajp13Connector** エレメントのポート属性を変更する場合は、**workers.properties** の **worker.ajp13\_2.port** 行のポートも変更する必要があります。これらのファイルは、RWP が RWP **servlet engine** の 2 つ目のインスタンスをサポートしている **Windows** ホスト上にも存在します。

## RWP のログを構成するには

アクセス、エラー、イベントのログに関する **rwp.conf** 内の多数の構成パラメータは、見出し **Logging-related directives** の下にまとめられています。次の構成を変更できます。

- **ErrorLog** には、エラーが記録されるファイルの名前を指定します。例を次に示します。

```
ErrorLog logs/error.log
```

これは、エラーの記録先を RWP のインストールディレクトリ下のファイル **logs/error.log** に指定します。

**メモ:** RWP のログ ファイルは、359 ページの「ログのローテーションとクリーンアップ」に説明されているように、**rotatelogs** コマンドにパイプされる場合があります。

- **LogLevel** は、ログに記録されるエラーのタイプと重要度を指定します。例を次に示します。

```
LogLevel warn
```

これは、ログに記録するエラーのレベルを警告以下に指定します。表 16 に、さまざまなログレベルを重要度の高いものから順に一覧表示します。**LogLevel** にこれらの値のいずれかを指定すると、その重要度以下のすべての重要度のイベントがログに記録されます。

表 16 RWP のログ レベル

| LogLevel | ログに記録されるメッセージ                              |
|----------|--------------------------------------------|
| emerg    | サーバーが動作しなくなる可能性のあるイベントに関する緊急メッセージ (最高の重要度) |
| alert    | 即座に修正する必要がある状態                             |
| crit     | ハードウェア エラーまたはシステム エラーなどの重要な状態              |
| error    | その他すべてのエラー                                 |
| warn     | 警告メッセージ                                    |
| notice   | 特別な処置が必要な可能性のある状態                          |
| info     | 情報メッセージ (最低の重要度)                           |
| debug    | RWP のデバッグ                                  |

- LogFormat は、イベントをログに記録するときの形式を指定します。定義済みの形式 (common など) の 1 つを選択することも、独自の形式を定義することもできます。ログ ファイルの文字列を構築するための形式トークンと規則の詳細については、[www.apache.org](http://www.apache.org) にある `mod_log_config` のマニュアルを参照してください。
- CustomLog は、RWP アクセス要求が記録されるファイルの名前を指定します。例を次に示します。

```
CustomLog logs/access.log common
```

これは、アクセス要求を、RWP のインストール ディレクトリ下のファイル `logs/access.log` に `common` ログ ファイル形式で記録することを指定します。

## ログのローテーションとクリーンアップ

デフォルトの構成では、ほとんどの RWP ログ ファイルは `rotatelog` プログラムにパイプされます。このプログラムは新しいログ ファイルを定期的に作成します。次の行 (ページ内に収まるように 2 行に分かれています) は、`rotatelog` を使用して、86,400 秒 (24 時間) ごとに `access.log` ファイルの新しいコピーを作成します。

```
CustomLog "|¥"/opt/rational/common/rwp/bin/rotatelog.exe¥"  
¥"/opt/rational/common/rwp/logs/access.log¥" 86400" common
```

ログのローテーション期間は、RWP が起動したときに始まります。

ClearCase Weekly Log Scrubbing ジョブは、30 日を超えたすべての RWP ログ ファイルを削除します。このジョブを修正して (229 ページの「ClearCase スケジューラ」を参照)、ジョブの実行頻度や削除するログ ファイルの経過日数など、操作の詳細を変更できます。

**メモ:** RWP ログのデフォルトの格納場所を変更した場合は、cleanuplogs スクリプトも変更して、新しい格納場所にログを検索できるようにする必要があります。

## RWP が使用するユーザー アカウントを変更するには

---

RWP はインストール時に、事前に定義されているユーザー アカウントの識別情報を適用して実行するように構成されます。このアカウントは、必要に応じて変更できます。変更するには、この項で説明するいずれかの手順に従います。

### Windows 上で RWP ユーザー アカウントを変更するには

Windows コンピュータでは、RWP は起動時に Windows サービス コントロール マネージャによって開始され、事前に定義されている LocalSystem アカウントの識別情報 (NT AUTHORITY¥SYSTEM) に基づいて実行されます。

Windows 上で RWP を実行する識別情報を変更するには

- 1 サービス アプリケーションを実行します ([コントロール パネル] で [サービス] をクリックするか、[コントロール パネル] の [管理ツール] で [サービス] をクリックします)。RWP には次のサービスがあります。
  - Rational Web Platform、HTTP Server
  - Rational Web Platform、servlet engine
  - Rational Web Platform、ReqWeb servlet engine

**注意:** HTTP Server と servlet engine は、ClearCase Web インターフェイスをサポートしている Windows RWP ホスト上で LocalSystem として実行する必要があります。これらのサービスがほかのユーザーとして実行された場合、ClearCase Web インターフェイスは動作しなくなります。ReqWeb servlet engine は、必要に応じて別のアカウントとして実行できます。

- 2 各サービスの [ログオン] プロパティを編集して、ローカル アカウントまたはドメイン アカウントを指定します。
- 3 rwp\_restart を実行し、RWP を停止して再起動します (361 ページの「RWP の停止と再起動」を参照)。

## UNIX 上で RWP ユーザー アカウントを変更するには

UNIX コンピュータでは、RWP は最初に **root** として起動し、必要なポートへのアクセスを取得します。その後、識別情報を、最小の権限を持ったユーザーの情報に変更します。このユーザーの名前とグループは、プラットフォームによって異なります。たとえば、**Solaris** では通常 **nobody.nobody** であり、**HP-UX** では **www.other** です。UNIX コンピュータ上でこのアカウントを変更するには、次の手順に従います。

- 1 **rwpcnf** の **User** 行と **Group** 行を編集します。次のように編集すると、RWP がユーザー **rwpcuser.other** として実行されるように構成されます。

```
User rwpcuser
Group other
```

- 2 RWP の起動スクリプト **rwpc\_startup** (RWP の **bin** ディレクトリにあります) の **su** コマンド行を編集します。指定されているユーザー (**su** コマンドの最初のパラメータ) を手順 1 で使用したアカウントと一致するように変更します。**su** コマンド行ではこれ以外変更しないでください。

```
su rwpcuser -c ...
```

- 3 RWP を停止して再起動します。

## RWP の停止と再起動

---

RWP は通常、起動時に開始されます。RWP を停止または再起動する必要がある場合 (変更された構成ファイルを再読み込みする場合など)、以下のコマンドの 1 つを使用します。これらは、通常 RWP の **bin** ディレクトリにインストールされています。

- **rwpc\_startup** は、RWP が実行されていない場合に RWP を起動します。
- **rwpc\_shutdown** は、RWP と、関連付けられている **servlet engine** プロセスを停止します。
- **rwpc\_restart** は、**rwpc\_shutdown** と **rwpc\_startup** コマンドを順に実行して、RWP を再起動します。

**メモ:** ClearCase を停止または起動するコマンドでは、RWP は停止または起動しません。

## ほかの Web サーバーから RWP へのアクセスを構成するには

---

一部のサイトは、プロキシまたはリダイレクトによってほかの Web サーバーから RWP にアクセスする必要があります。この構成では、ほかの Web サーバーは、同じホスト上で実行されているが別のポートを使用している RWP プロセスか、別のホスト上で実行されている RWP プロセスに特定の URL をリダイレクトします。次の 2 種類の一般的な使用環境で、このタイプの構成を行う必要があります。

- **RWP と別の Web サーバーを同じホスト上で実行する必要がある場合。**ほかの Web サーバーを実行する必要のないホストに RWP をインストールすることをお勧めします。それが不可能な場合は、RWP によって使用されていないポートを使用するよう、ほかの Web サーバーを構成することをお勧めします。これができない場合は、ほかの Web サーバーによって使用されないポートを使用するように RWP を構成し (357 ページの「デフォルトの RWP HTTP ポートを変更するには」を参照)、必要に応じて Rational Web クライアントの URL を RWP にリダイレクトするようにほかの Web サーバーを構成する必要があります。
- **RWP をファイアウォールの内側で実行する必要がある場合。**RWP へのアクセスを制限するために、ファイアウォールのパブリック側で実行されている Web サーバーを、ファイアウォールの反対側で実行されている RWP インスタンスに特定の URL を渡すように構成できます。

この項の手順に従って、以下のいずれかの Web サーバーからの RWP へのアクセスを可能にする、プロキシまたはリダイレクトの構成を有効化します。

- Apache HTTP Server
- Microsoft Internet Information Server (IIS)

**メモ:** Rational Web アプリケーションへのプロキシアクセスまたはリダイレクトアクセスを構成する方法は、そのアプリケーションと、プロキシとして動作する Web サーバーに固有のもので、この項で具体的に上げられている Web サーバーと Rational 製品のみが、プロキシまたはリダイレクトの構成においてサポートされます。

### Apache 用 mod\_proxy サポートの構成

Apache HTTP Server のインスタンスを構成して RWP へのプロキシアクセスをサポートするには、Apache の mod\_proxy モジュールが提供するプロキシサポートを使用して Apache HTTP Server を構成する必要があります。この方法に関する詳細情報は、[www.apache.org](http://www.apache.org) で入手できます。必要となる手順の概要は次のとおりです。

- 1 mod\_proxy モジュールなどの必要なモジュールをロードするように Apache HTTP Server を構成します。通常、Apache の httpd.conf ファイル内の mod\_proxy サポートに関するさまざまな LoadModule 指示文のコメントを外す必要があります。以下に例を示します。

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_connect_module modules/mod_proxy_connect.so
LoadModule proxy_http_module modules/mod_proxy_http.so
```

また、次のように、`httpd.conf` 内の `<IfModule mod_Proxy.c>` ブロックの `ProxyRequests On` 指示文のコメントも外す必要があります。

```
<IfModule mod_Proxy.c>
ProxyRequests On
</IfModule>
```

- 2 `httpd.conf` 内の `<IfModule mod_proxy.c>` ブロックに適切な `ProxyPass` 指示文と `ProxyPassReverse` 指示文を追加します。`ProxyPass` 指示文と `ProxyPassReverse` 指示文はアプリケーションに固有です。

ClearCase Web インターフェイスには、次の `ProxyPass` 指示文と `ProxyPassReverse` 指示文を追加します。

```
ProxyPass /ccweb http://hostname[:port]/ccweb
ProxyPassReverse /ccweb http://hostname[:port]/ccweb
ProxyPass /Java_Plugins http://hostname[:port]/Java_Plugins
ProxyPassReverse /Java_Plugins http://hostname[:port]/Java_Plugins
```

`hostname` には RWP サーバー ホストの名前を、`port` にはオプションのポート番号を指定します。これらは、RWP が HTTP の要求をリッスン (受信を監視) するために使用する、デフォルトのポートを変更した場合に必ず指定してください (357 ページの「デフォルトの RWP HTTP ポートを変更するには」を参照)。たとえば、次の指示文は、RWP\_host というホストのポート 81 でリッスン (受信を監視) する RWP プロセスに対して、ClearCase Web インターフェイスによるアクセスをサポートするプロキシサーバーを構成します。

```
ProxyPass          /ccweb http://RWP_host:81/ccweb
ProxyPassReverse   /ccweb http://RWP_host:81/ccweb
ProxyPass          /Java_Plugins http://RWP_host:81/Java_Plugins
ProxyPassReverse   /Java_Plugins http://RWP_host:81/Java_Plugins
```

**メモ:** この例で指定されている URL は、ホスト名とオプションのポート番号以外、ここで指定されているとおりに `httpd.conf` ファイルに記述する必要があります。

## Internet Information Server (IIS) の URL リダイレクトの構成

ポート 80 で HTTP の要求をリッスン (受信を監視) する Microsoft Internet Information Server (IIS) のインスタンスと RWP をホスト上で共存させる必要がある場合、別のポートで HTTP の要求をリッスン (受信を監視) するように RWP を再構成し (357 ページの「デフォルトの RWP HTTP ポートを変更するには」を参照)、以下の 1 つを実行する必要があります。

- RWP のこのインスタンスからサービスを受ける Rational Web インターフェイスが使用する URL に、ポート指定子 (たとえば `http://hostname:81/ccweb/`) を含めます。
- IIS のリダイレクト機能を使用して、ポート 80 に送られる (そして IIS によって受信される) Rational Web インターフェイスの URL を RWP にリダイレクトします。

リダイレクトを使用するように IIS を構成するには

- 1 IIS 構成ユーティリティ (インターネット サービス マネージャ) を実行します。
- 2 IIS の [既定の Web サイト] フォルダに新しい仮想ディレクトリを作成します。
  - [仮想ディレクトリ エイリアス] で、仮想フォルダを使用する Web クライアントの名前を反映する名前を選択します (たとえば、ccweb)。
  - [Web サイトのコンテンツのディレクトリ パス] で、Web サーバー ホスト上の物理ディレクトリを指定する必要があります。このディレクトリはホスト上に存在する必要がありますが、手順 4 でリダイレクトを構成した後は、Web サイトのコンテンツを保持するために使用することはありません。よって、この目的のための新規ディレクトリを作成し、誤って削除されるのを防ぐために保護を適用することをお勧めします。

**メモ:** このディレクトリを RWP のインストールディレクトリのサブディレクトリとして作成すると、RWP をホスト上に再インストールしたときにこのディレクトリが削除されます。
  - [アクセス許可] で、[読み取り] と [ASP 等のスクリプトを実行する] を指定します。
- 3 手順 2 で作成した仮想ディレクトリを右クリックして、[プロパティ] ダイアログ ボックスを開きます。
- 4 [仮想ディレクトリ] タブの [このリソースへの接続時に使用されるコンテンツの場所] セクションで、[URL へのリダイレクト] を選択します。
- 5 [リダイレクト先] ボックスで、RWP にリダイレクトする Rational Web インターフェイスによって使用される URL を入力します。たとえば、ClearCase Web インターフェイス (ccweb) をリダイレクトして、ポート 81 でリッスン (受信を監視) している RWP のインスタンスを使用するには、次のように入力します。

http://hostname:81/ccweb/

hostname には、RWP と IIS を実行しているホストの名前を指定します。
- 6 [クライアントは以下に送信されます] セクションで、[上で入力した URL] を選択します。
- 7 URL http://hostname/ccweb の参照が、手順 5 で指定した URL の ClearCase Web インターフェイスにリダイレクトされることを確認します。

## セキュア ソケットを使用するための RWP の構成

---

Rational Web クライアントと RWP の間で安全な通信を確立するために、Secure Sockets Layer (以下 SSL) プロトコルのサポートを RWP に構成できます。この構成は、次の手順で実行します。

- 1 RWP 構成ファイルを編集し、SSL サポートを有効にします。
- 2 必要に応じてクライアント URL を変更し、HTTPS プロトコルを指定します。
- 3 RWP を停止して再起動します。

SSL をサポートするための RWP の構成の手順は、`mod_ssl` モジュールを使用する Apache HTTPD の構成の場合と同じです。その手順については、`mod_ssl.org` の Web サイトに完全なマニュアルがあります。ここでは、構成手順の概要を説明します。

SSL を使用するには、最初に認証局 (CA) から証明書を取得します。RWP には `openssl` プログラムが含まれており (RWP の `bin` ディレクトリにインストールされています)、それを使用して、テストのために自己署名証明書を生成し、さらに CA から証明書とキーを取得できます。`openssl` の詳細については、Web サイト [www.infoscience.co.jp/technical/openssl/](http://www.infoscience.co.jp/technical/openssl/) を参照してください。

RWP の `bin` ディレクトリで `openssl` を実行して、自己署名証明書と一致する秘密キーを生成し、次の手順の手順 2 と手順 3 で指定した場所にそれらをインストールできます。

- UNIX の場合は、次の `openssl` コマンド行を使用します。

```
OpenSSL> req -new -x509 -nodes -keyout /opt/rational/common/rwp/conf/server.key ¥  
-out /opt/rational/common/rwp/conf/server.crt
```

- Windows の場合は、次の `openssl` コマンド行を使用します。

```
OpenSSL> req -new -x509 -nodes -keyout ../conf/server.key -out ../conf/server.crt
```

**メモ:** 指定した `-out` ディレクトリ内にファイルを作成する権限を所有していない場合、このコマンドは失敗します。

SSL 接続を使用できるように RWP を構成するには

- 1 `ssl.conf` 構成ファイルが含まれるように RWP を構成します。`rwp.conf` 内の次の指示文のコメントを外します。

```
Include conf/ssl.conf
```

**メモ:** `ssl.conf` ファイルには、`Listen` 指示文が含まれており、RWP が HTTPS の要求をリッスン (受信を監視) するポートが指定されています。デフォルトは、ポート 443 です。このポート番号の変更方法は、デフォルトの HTTP ポートを変更する方法と同じです。RWP に HTTPS の要求のみをリッスン (受信を監視) させるには、`rwp.conf` ファイル内の `Listen` 指示文をコメントにします。

- 2 証明書をインストールします。証明書ファイルのデフォルトの場所は、**ssl.conf** ファイル内の次の指示文で指定されます。

```
SSLCertificateFile rwp-root-dir/conf/server.crt
```

**rwp-root-dir** は、ホスト上の RWP がインストールされているディレクトリを表します。証明書ファイルを別の場所にインストールする場合は、**ssl.conf** 内のこの行がその場所を参照していることを確認してください。

- 3 キーをインストールします。キー ファイルのデフォルトの場所は、**ssl.conf** ファイル内の次の指示文で指定されます。

```
SSLCertificateKeyFile rwp-root-dir/conf/server.key
```

キー ファイルを別の場所にインストールする場合は、この指示文がその場所を参照していることを確認してください。

- 4 RWP を停止して再起動します。

**メモ:** SSL を使用するように Web アプリケーションを構成するには、アプリケーションの URL に **https** プロトコルを指定します。例を次に示します。

```
https://RWP_host.domain/ccweb
```

## RWP への安全なアクセスの構成

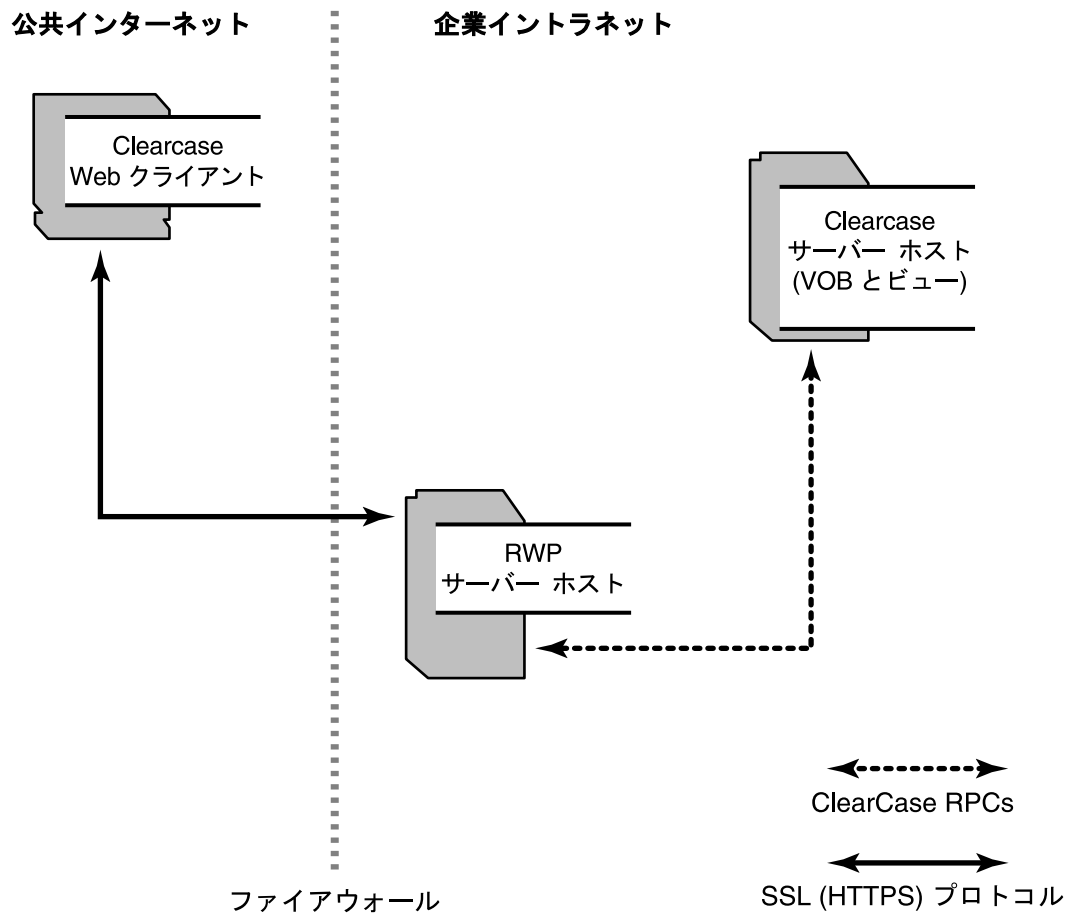
**ccweb** などの Rational Web クライアントとの通信を安全なものにするため、SSL の使用を可能にする RWP の構成方法は多数あります。図 9 は、公共のインターネット上の **ccweb** クライアントがファイアウォールを介して RWP にアクセスすることを可能にする一般的な構成を示しています。クライアントの Web ブラウザと **ccweb** をサポートする RWP ホストとの間の通信は、HTTPS プロトコルを使用し、SSL によって保護されます。企業のイントラネット上の ClearCase サーバーと RWP との間の通信は、通常の ClearCase リモート プロシージャ コール (RPC) を使用しており、安全ではありません。ClearCase RPC は、ファイアウォールを介して通信できません。したがって、**ccweb** が使用する ClearCase サーバーと RWP との間にファイアウォールを配置することはできません。

図 9 で示すような構成では、**ccweb** の URL は次のようになります。

```
https://hostname/ccweb
```

**hostname** は、RWP サーバー ホストの名前です。ClearCase Web クライアントと RWP の間の HTTPS 通信は、HTTPS のデフォルトであるポート 443 を使用します。

図 9 ccweb と RWP の間の安全な通信



## RWP に対するその他の変更

---

RWP 構成ファイルには、この付録に記載されている以外の変更は行わないことをお勧めします。構成オプションの中には、変更すると RWP の処理に悪影響を及ぼすものもあります。この付録で推奨する以外の構成の変更は、実働環境に組み込む前に慎重に評価する必要があります。

## ClearCase Web インターフェイスの構成

---

ほとんどのサイトでは、ClearCase Web インターフェイスは特別な構成を必要としません。ホストに RWP がインストールされていれば、次の URL でこのインターフェイスをサポートできます。

`http://hostname[:port]/ccweb`

hostname は RWP ホストの名前です。port は、RWP が HTTP 用に 80 以外のポートを使用するように構成されている場合の任意のポート番号です (357 ページの「デフォルトの RWP HTTP ポートを変更するには」を参照)。

**メモ:** RWP が Windows コンピュータで実行される場合、ClearCase Web インターフェイスのユーザーに RWP ホストへのローカル ログオンの権限を与える必要があります。Windows は、この権限をデフォルトでは付与していません。

この項の残りの部分では、次の ClearCase Web インターフェイス構成ファイルを編集する方法について説明します。

`ccase-home-dir/config/ccweb/ccweb.conf`

ClearCase Web インターフェイスのデフォルトの構成を変更する必要がある場合に編集します。

**メモ:** このファイルと RWP の ClearCase Web インターフェイス サポート ファイルを混同しないでください。

サポート ファイルは、Windows では `C:\Program Files\Rational\common\rwp\ccweb.conf`、UNIX では `/opt/rational/common/rwp/conf/include/ccweb.conf` としてインストールされます。このファイルは変更しないでください。

## ClearCase プライマリ グループの指定

Windows 上の ClearCase ユーザーが CLEARCASE\_PRIMARY\_GROUP 環境変数を設定しているサイトでは (337 ページの「ClearCase プライマリ グループの設定」を参照)、ccweb.conf ファイルの次の行を変更して、ClearCase Web インターフェイスのユーザーに対してプライマリグループも指定する必要があります。

**-primary\_group group-name**

group-name は、ドメイン グループの名前です。このホスト上の ClearCase Web インターフェイスにアクセスするすべてのユーザーの CLEARCASE\_PRIMARY\_GROUP として使用されます。

**メモ:** 複数のドメインのユーザーが ClearCase Web インターフェイスにアクセスする場合、340 ページの「Windows NT ドメインでのプロキシ グループとドメイン マッピングの使用法」の説明に従って RWP ホスト上のドメインを有効化し、ccweb.conf の -primary\_group に適切な値を指定する必要があります。

## Web ビュー記憶領域

ClearCase Web インターフェイスは、通常、RWP サーバー ホスト上に Web ビュー ディレクトリを作成します。このディレクトリは、チェックアウトされるかエレメントとして作成される準備ができていてファイルの一時記憶領域として使用されます。したがって、Web ビューディレクトリが配置されるディスクのボリュームには、サポート対象の多数の Web ビューを格納するのに十分な空き容量が必要です。各 Web ビューに対して 0.5 ～ 1 MB のディスク容量を割り当てることをお勧めします。

UNIX の場合、これらのディレクトリは通常 /var/adm/rational/clearcase/ccweb の下に作成されます。

Windows の場合、これらのディレクトリは通常

C:\Program\Files\Rational\ClearCase\var\ccweb の下に作成されます。

デフォルトの Web ビュー記憶領域のディレクトリが適切でない場合は、ccweb.conf に次の行を追加すると別の領域を選択できます。

**-view\_storage pathname**

pathname には、Web ビューディレクトリ作成先となるディレクトリを指定します。このディレクトリは、RWP ホスト上にローカルで存在する必要があります。

**メモ:** ClearCase Web インターフェイスは Web ビュー記憶領域用に別のディレクトリを作成する場合がありますが、実行可能ファイルと管理ファイルは引き続き ccase-home-dir/ccweb に保存します。

## アップロードサイズの制限

Web ビュー記憶領域の管理の向上、または DoS 攻撃の可能性を減らすために、RWP ホストにアップロードできるファイルのサイズを制限することができます。それには、**ccweb.conf** ファイルの次の行を変更します。

**-upload\_limit size**

**size** には、サイズ制限のおおよその値を指定します (バイト単位)。アップロードするファイルが大きすぎると、クライアント アップロード出力ウィンドウにエラー メッセージが表示されます。

## セッション タイムアウトの指定

セッション タイムアウトの間隔を設定できます。これにより、ユーザー ログインの有効期間を制御できます。デフォルト値は、14400 秒 (4 時間) です。デフォルト値を変更する場合は、次の行を変更します。

**-session\_timeout seconds**

この行は、**ccweb.conf** にあります。**seconds** には、600 (10 分) ～ 2147483647 (約 68 年) の整数 (秒数) を指定します。600 よりも小さい値は 600 とみなされます。

## 一時記憶領域用ディレクトリの指定

次の形式の行を追加すると、ClearCase Web インターフェイスが一時的にファイルを保存するディレクトリを指定できます。

**-tmpdir directory-name**

**ccweb.conf** にこの行を追加します。**directory-name** には、ClearCase Web インターフェイスがファイルの作成と削除の権限を持つ RWP ホスト上のディレクトリを指定します。**ccweb.conf** 内にこの行を指定しない場合、ClearCase Web インターフェイスは、TMP 環境変数または TEMP 環境変数を適用します (存在する場合)。

## Windows 上にアプレットをダウンロードする権限

Internet Explorer では、通常、アプレットをダウンロードするためにローカル管理者権限が必要です。このアプレットには、ClearCase Web インターフェイスが使用するものも含まれます。これらのアプレットをダウンロードした (Web インターフェイスを初めて使用した) 後は、ClearCase のパッチまたは ClearCase の新しいリリースでアプレットを変更しないかぎり、再度ダウンロードする必要はありません。

# UNIX 上での非 ClearCase アクセスの構成

# D

一部の ClearCase コミュニティでは、Rational ClearCase を実行できないコンピュータに VOB と動的ビューへのアクセスを提供する必要があります。この付録では、このアクセスを提供できるように UNIX 環境の ClearCase ホストを構成する方法について説明します。

**メモ:** 非 ClearCase アクセスは、ClearCase LT を実行している UNIX ホストまたは Windows 環境の ClearCase ホストではサポートされません。

## UNIX ホスト上での非 ClearCase アクセスの使用方法

---

ClearCase がインストールされていない UNIX ホストでは、非 ClearCase アクセスを使用して UNIX VOB サーバーから VOB を読み込むことができます。通常は、以下の方法で行います。

- ClearCase を実行している UNIX ホストからビュー拡張パス名を VOB マウント ポイント (たとえば、/view/exportvu/vobs/vegaproj) にエクスポートする必要があります。ファイル /etc/exports.mvfs を編集して、このパス名を指定します。
- 1 台または複数の非 ClearCase ホストでは、ビュー拡張パス名を使用して VOB にアクセスします。たとえば、ホストのファイル システム テーブルには、次で始まるエントリがあります。

```
mars:/view/exportvu/vobs/vegaproj /usr/vega nfs ...
```

### 使用上の制限事項

非 ClearCase アクセスには、以下の制限事項が適用されます。

- **VOB アクセス。** 非 ClearCase ホスト上のユーザーは、非 ClearCase アクセス用に構成された UNIX VOB サーバー ホスト上の VOB のデータを読み込むことはできますが、変更することはできません。また、特定のビューで選択したエレメント バージョンの使用も制限されます。バージョン拡張パス名またはビュー拡張パス名を使用して、VOB のエレメントのほかのバージョンにアクセスすることはできません。

- **ビルド**。ユーザーは、エクスポート ビューを介してアクセスした VOB を変更できませんが、ビュー プライベート記憶領域に書き込むことはできます。ユーザーは、これらのビュー プライベート ファイルをエディタを使用して変更してビルドできますが、**clearmake** を使用してビルドすることはできません。このような過程でビルドで作成されたファイルは、派生オブジェクトにはなりません。変換しない限り、ビュー プライベート ファイルです。詳細については、『**Rational ClearCase ソフトウェア ビルドガイド**』を参照してください。
- **clearmake** やほかのビルド管理ツールは非 **ClearCase** ホストで実行できないため、構成 ルックアップや DO (派生オブジェクト) の共有は行えません。

## 非 ClearCase アクセスのためのエクスポート ビューの設定

---

ビューと VOB の組み合わせをエクスポートする一般的な手順は次のとおりです。

- 1 MVFS を使用するように構成された **ClearCase** のクライアント ホストが、VOB をアクティブ化 (マウント) します。事前に、**mkvob** または **mktag** に **-ncaexported** オプションを指定して実行し、VOB をエクスポート用にマークしておきます。
- 2 上記の **ClearCase** ホストがエクスポート ビューを有効にします。このビューを通して非 **ClearCase** ホストから VOB にアクセスできます。エクスポート ビューは、次の 2 つの要件を満たす必要があります。
  - **mkview** または **mktag** に **-ncaexported** オプションを指定して実行し、エクスポート用にマークされている。
  - ビューのタグが複数のレジストリ リージョンにある場合は、エクスポートしたタグがビュー サーバー ホストのリージョンに存在する。
- 3 ホストでは **/etc/exports.mvfs** ファイルを使用してビュー拡張パス名を VOB マウント ポイント (たとえば、**/view/exp\_vu/vobs/proj**) にエクスポートします。

**メモ:** この VOB マウント ポイント内のシンボリック リnkはいずれも相対シンボリック リンクでなければなりません。その VOB マウント ポイントを含むシンボリック リンクのすべての絶対ターゲットは、エクスポート機能を有効にするため、相対ターゲットに変更する必要があります。
- 4 ネットワーク内の非 **ClearCase** ホストは、エクスポートされたパス名の NFS マウントを実行します。

オペレーティング システムに固有の手順については、**exports\_ccase** リファレンス ページを参照してください。そこでは、VOB とエクスポート ビューが同一のホストに存在している場合の、最も簡単でお勧めする設定例が説明されています。次の項では、この点について詳細に説明します。また、エクスポート ビューを VOB サーバーホストに配置しない場合の操作に関しても説明します。

**メモ:** エクスポート ビューの構成仕様を変更する場合は、非 ClearCase ホストにアクセスするためにマウントしたビューを保有しているすべてのユーザーが、そのビューを一度アンマウントして再マウントしたか確認する必要があります。再マウントすることにより、更新された構成仕様に指定されている正しいファイルにアクセスできるようになります。

## 複数の VOB のエクスポート

各 VOB とそのエクスポート ビューを同じホストに配置した場合、非 ClearCase ホストで作業する開発者が複数のエクスポート ビューに同時にアクセスする可能性があります。たとえば、独自のエクスポート ビューがある、3 つの異なるホストに存在している 3 つの VOB に非 ClearCase アクセス機能を提供するには、非 ClearCase ホスト上に以下のような NFS マウント テーブル エントリを作成する必要があります。

```
saturn:/view/beta/vobs/proj /vobs/proj nfs rw,hard 0 0
neptune:/view/exp_vu/vobs/proj_aux /vobs/proj_aux nfs rw,hard 0 0
pluto:/view/archive_vu/vstore/tools /vobs/tools nfs rw,hard 0 0
```

これにより、非 ClearCase ホストでは 3 つの VOB を /vobs のサブディレクトリとしてアクセスすることができます。各 VOB とビューの組み合わせに対して、適切なエクスポート ビューで **チェックアウト**や**チェックイン**などの ClearCase 操作を行う必要があります。ビルドやデバッグなどのほかの操作では、ビューを参照する必要はありません。

## エクスポート構成

非 ClearCase ホストからデータにアクセスする場合、次の 3 つのホストが関係します。

- VOB が存在するホスト
- エクスポート ビューが存在するホスト
- 非 ClearCase ホスト

VOB とそのエクスポート ビューが異なるホストに存在するマルチホップ構成では、2 つの ClearCase ホスト間に、NFS 通信よりも低速な MVFS レベルの通信が必要です。また、マルチホップ構成では、2 つのホストがネットワーク関連サービスについて互いに依存し合う、またはそうした依存関係がサードパーティ ホストを介して生じるといったアクセス ループに陥る危険性が生まれます。そのような状況は結果的にタイムアウト (VOB がソフト的にマウントされている場合) か、デッドロック (VOB がハード的にマウントされている場合) の原因となります。

非 ClearCase アクセスを使用する場合は、マルチホップ構成を避けることをお勧めします。

## マルチホップ構成の回避方法

以下のガイドラインに従うと、マルチホップ構成を回避できます。

- エクスポート ビューを VOB と同じホストに配置します。view\_server と vob\_server を同じホスト上で実行する (305 ページの「プロセス オーバーヘッドの最小化」を参照) ことは通常お勧めしませんが、マルチホップ構成を使用するよりも望ましい方法です。
- VOB とビューのどちらにもリモート データ記憶領域が存在しないことを確認します。つまり、VOB にはリモート記憶プールが存在せず、ビューのプライベート記憶領域 (.s ディレクトリ) がホストに対してローカルである必要があります。

## マルチホップ構成での操作

マルチホップ構成を使用する場合は、VOB ホスト ( そのプール ホストが存在する場合はそれも含む) がビュー ホストからネットワーク サービスを要求しないようにしてください。これにより、VOB とプール ホスト上のプロセスで、ビュー ホストでアクセス ループが発生することがなくなります。エクスポート ビュー ホストから、直接またはほかのホストを通じて、ClearCase サービスまたはほかのネットワーク サービスを要求しないでください。

これを実現するための一つの方法として、エクスポート ビュー ホスト上でユーザーがプロセスを直接または間接的に実行することを禁止する方法があります。ただし、次のような場合はこの方法を適用できません。

- ホーム ディレクトリの作成
- ClearCase ホストからのリモート ログオン
- VOB サーバーまたはプール ホストによるビュー サーバー ホストのリモート バックアップ

また、結果的に類似したアクセス パターンになるその他の場合も適用外となります。

## 特定のホストへのエクスポートを制限する方法

マルチホップ構成では、/etc/exports.mvfs の各エントリで -access オプションを指定することをお勧めします。これにより、指定した非 ClearCase ホストかネットグループ、またはその両方に対するエクスポートが制限され、アクセス ループの危険性が大幅に減少します。例を次に示します。

```
/view/exp_vu/usr/src/proj -access=galileo:newton:bohr:pcgroup
```

-access をほかのオプションと組み合わせて使用する場合は、各オプションの先頭にハイフンを1 つ付けて、各オプションをカンマで区切ります。

## 非 ClearCase アクセスでの自動マウントの使用法

---

NFS を使用して ClearCase から ClearCase ビューと VOB の組み合わせをエクスポートすると、承認されている NFS クライアント システムから、ビューで選択した VOB 内のファイルにアクセスできるようになります。NFS クライアントで `mount` コマンドを使用してビューと VOB の組み合わせを直接マウントできる場合は、NFS クライアントの自動マウント機能が使用するマップにその組み合わせを明示的または暗黙的に指定することもできます。明示的なエントリとは、エクスポートされたビューと VOB の組み合わせを直接示します。暗黙的なエントリとは、ワイルドカード構文やほかの高度な自動マウント機能を使用します。

たとえば、`server:/view/viewname/vobs/&` が含まれているマップ ファイルを使用して、`/remote/viewname` に間接的なマップを (一般的な自動マウント ワイルドカード構文を使用して) 構成したとします。NFS クライアント上のプロセスが `/remote/viewname` のサブディレクトリにアクセスすると、自動マウント機能は `server:/view/viewname/vobs` の対応するサブディレクトリから NFS マウントを実行します。

**メモ:** ディレクトリ `/remote/viewname` を表示すると、通常、アクティブなマウントだけが表示され、可能なマウントがすべて表示されることはありません。これは、ホスト マップの `/net` を表示した場合と同様です。

このようなタイプのマップが正常に機能しない場合、明示的な `mount` コマンドを実行してみてください。このコマンドが正常に機能した場合は、クライアントの自動マウント機能に問題があります。自動マウント機能のマップ構文の詳細については、使用しているプラットフォームのマニュアルを参照してください。

**メモ:** `-hosts` マップを使用する自動マウント アクセスは、ルート ディレクトリもエクスポートするホスト上で正常に機能しません。たとえば、NFS クライアント ホストが `/net/cchost/view/viewname/vobs/vobpath` にアクセスするとします。自動マウント機能は、サーバーのルート ディレクトリを `/net/cchost` にマウントし、その後ビューと VOB を `/net/cchost/view/viewname/vobs/vobpath` にマウントしようとします。ところが、NFS エクスポートは `/view` などのローカル ファイル システムのマウントに従わないため、`/net/cchost/view` にはサブディレクトリは存在しません。したがって、ローカルクライアントはビューと VOB の組み合わせをマウントするディレクトリを見つけることができないため、このマウントは失敗します。

ClearCase と自動マウント機能の詳細については、19 ページの「ClearCase と NFS 自動マウント機能」を参照してください。

## 非 ClearCase アクセスでの NFS の問題

---

非 ClearCase ホストによって NFS の実装方法が異なるため、非 ClearCase アクセスを使用したときに発生する問題は多種多様です。

### NFS クライアント キャッシュでの問題

ほとんどの NFS クライアント実装にはキャッシュ機能が備わっており、頻繁に使用するファイルデータやメタデータへのアクセスを高速化します。一般的に、古いクライアントよりも新しいクライアントの方が積極的にキャッシュ機能を使用します。NFS クライアントがキャッシュが有効であると判断していても、ビューまたは VOB 内のデータがキャッシュ データと整合性がないと、クライアントは VOB から正しくないファイルにアクセスすることがあります。

一般的に、別のビューからファイルをチェックインした場合、またはエクスポート ビューの構成仕様を変更した場合に整合性が保てなくなります。結果的に、ビューが新しいバージョンのファイルを選択する場合、NFS クライアントはこの変更に気がつきません。これは、NFS クライアントでは、名前/ファイルのバインドが変更された場合、ファイルが格納されているディレクトリのタイム スタンプが変更されると予測していることが原因です。この場合、エクスポート ビューのディレクトリは変更されていませんが、そのディレクトリにカタログされているファイルのバージョンが変更されます。NFS クライアントでは、容量の制約によりディレクトリが変更されたか、キャッシュからエントリが排除されたと判断しない限り、キャッシュされている名前/ファイルのバインドを再確認しません。

ほとんどの NFS クライアントは、キャッシュの有効時間が短時間 (通常 60 秒未満) であると見なします。キャッシュが短時間で更新されない場合は、次のいずれかの方法でこの制限事項に対処できます。

- 該当するファイルを格納しているディレクトリにダミー ファイルを作成し、それを削除します。これにより、ディレクトリのタイム スタンプが変更され、クライアントのキャッシュが無効になるため、クライアントは新しいファイル バージョンを探すようになります (プラットフォームによっては、この方法は、リモート ホスト上で実行した場合だけ効果があります)。
- クライアントの属性キャッシュを無効にします (通常は、**noac** マウント オプションを指定して無効にします)。ただし、当社のテストでは、これは一部の NFS V2 クライアントでだけ機能し、NFS クライアントとエクスポート ビュー サーバー間のネットワーク トラフィックが増加することが確認されています。クライアントが NFS V3 をデフォルトで使用し、**noac** を使用する場合は、NFS V2 を要求するようにマウント オプションを編集することをお勧めします。
- **root** としてログインし、ファイル システムをアンマウントしてから再マウントします。これにより、ファイル システムの NFS クライアント キャッシュがフラッシュされます (アンマウントが失敗した場合でも、キャッシュはフラッシュされます)。

また、非 ClearCase アクセスの動的特性を制限することをお勧めします。それには、新しいバージョンのファイルを必ず選択する構成仕様の使用を避けます。/main/LATEST 規則ではなく、ラベルベースの規則を使用してください。

## NFS ロックでの問題

非 ClearCase アクセスはファイルに対する NFS ファイル ロックをサポートしていないため、非 ClearCase アクセスを使用してアクセスしたファイルでは、ファイルをロックする必要があるアプリケーションが正常に機能しません。ファイル ロック機能は、一部の UNIX プラットフォーム上のビュープライベート ファイルでは機能しますが、VOB ファイルでは機能しません。ロックを取得できるまでロック要求を再試行し続け、アプリケーションがハングすることがあります。また、そのアプリケーションがロックを取得できず、複数のクライアントが同じファイルを変更した場合には、ファイルが破損することもあります。各自のアプリケーションがファイルをロックする必要がある場合は、スナップショット ビューまたは ClearCase Web インターフェイスを使用して、ClearCase を実行しないホストから VOB データにアクセスしてください。



# Microsoft Web オーサリング ツールとの統合



この章では、ClearCase と Microsoft Web オーサリング ツールとの統合の設定方法について説明します。

## 統合の概要

---

Windows のプラットフォームでは、Rational ClearCase と Rational ClearCase LT は、次の Microsoft 製品と統合できます。

- FrontPage 98、2000 と、それ以降の製品
- Visual InterDev 6.0
- Word、Excel、PowerPoint 用の Office 2000 Web Folders 機能
- Internet Explorer 5.5 と、それ以降の Web Folders

上記の統合を行うと、それらのツールを使用して作成した Web コンテンツと Web アプリケーションに ClearCase の構成管理機能を適用できるようになります。そのためには、サーバーとクライアントの 2 箇所を設定を行う必要があります。

ソース管理下にある Web 用に使用できるサーバー ホストは、Internet Information Server (IIS) が組み込まれている Windows NT コンピュータまたは Windows 2000 コンピュータです。Windows NT Workstation 4.0 上では、IIS は Personal Web Server (PWS) と呼ばれています。Windows 2000 Workstation 上では、Web サーバーは IIS と名付けられています。

クライアントは、サポート対象の Web オーサリング アプリケーションを実行する Windows コンピュータです。クライアントでは、URL を使用してサーバー上の Web を開き、サーバー上に存在する共有のスナップショット ビューを使用して、チェックアウトとチェックインを行います。各 IIS サーバーは、デフォルトの IIS の別名を使用して、単一の Web コンテンツ VOB に関連付けられます。複数の IIS サーバーが同一の Web コンテンツ VOB を共有するように設定できます。

## サーバーの設定の概要

サーバーをインストールする手順の概要を以下に示します。各手順の詳細については、381 ページの「サーバーの設定の手順」を参照してください。

- 1 IIS 4.0 (Windows NT 4.0 Option Pack から) または IIS 5.0 (Windows 2000 から) をインストールします。IIS 5.0 はデフォルトで Windows 2000 Server 上にインストールされます。Windows NT 4.0 上に IIS が既にインストールされている場合、そのバージョンが 4.0 またはそれ以降であることを確認してください。IIS のバージョン番号は、Windows のレジストリキー HKEY\_LOCAL\_MACHINE¥SYSTEM¥CurrentControlSet¥Services¥W3SVC¥Parameters¥MajorVersion に記述されています。
- 2 FrontPage Server Extensions バージョン 4.0.2 (Office 2000 CD3、Office Server Extensions、Windows 2000 のいずれかから) をインストールします。Windows NT 4.0 Server または Windows 2000 Server 上で Option Pack をインストールすると、デフォルトで FPSE がインストールされます。IIS と共に FPSE がインストールされていれば、ここで再インストールする必要はありません。前に Option Pack をインストールしたときにバージョン 3 がインストールされた場合は、ここで Office Server Extensions にアップグレードすることができます。
- 3 ClearCase をインストールします。
- 4 Web オーサリング統合の構成ウィザードを実行して、画面の指示に従います。

## クライアントの設定の概要

クライアントを設定する手順の概要を以下に示します。詳細については、384 ページの「クライアントの設定の手順」を参照してください。

- 1 サポートされているオーサリング ツールをインストールします。
- 2 FrontPage 98 または Visual InterDev 6.0 を使用し、Web を作成してソース管理に追加します。
- 3 新しい Web コンテンツがソース管理に追加されたことを確認します。
- 4 FrontPage 98 または Visual InterDev 6.0 を使用してコンテンツを作成するユーザーのために、作成者や管理者の権限を有効にします。
- 5 必要に応じて、サポートするツールに対するクライアントのローカル モードでのサポートを可能にします。

## サーバーの設定の手順

表 17 に、Web サーバー用にサポートされているサーバー プラットフォームと必要なソフトウェアのバージョンを示します。

**表 17 Microsoft Web オーサリング ツールとの統合をサポートしているプラットフォーム**

Windows のバージョン	Web サーバー ソフトウェア	FrontPage Server Extensions または Office Server Extensions
Windows NT 4.0	IIS 4.0	3.0.2.1105 (FrontPage 98 CD)
Windows NT 4.0	IIS 4.0	3.0.2.1706 (Windows NT 4.0 Option Pack CD)
Windows NT 4.0 または Windows 2000	IIS 4.0 または IIS 5.0	4.0.2.2717 (Office 2000 に同梱)
Windows 2000	IIS 5.0 または IIS 5.1	4.0 またはそれ以降

**メモ:** MS FrontPage Server Extensions 2002 を使用している場合は、Rational カスタマ サポート からファイル iisfix.bat を入手してください。

### 手順 1: IIS のインストール

ホスト上の Web サーバーは IIS のみとし、デフォルトのポート (80) を使用することをお勧めします。IIS のインストールを開始する前に、Web サーバーのソフトウェアをチェックします。

**メモ:** Rational Web Platform (RWP) と IIS を同じホストにインストールする必要がある場合は、一方または両方の Web サーバーを、互いに干渉しないように設定してください。詳細については、付録 C を参照してください。

Windows 2000 サーバーを使用していて、Windows 2000 をインストールするときにすべてデフォルト値を採用した場合、サーバーには IIS 5.0 と FrontPage 2000 Server Extensions がインストールされています。

Windows NT 4.0 サーバーを使用している場合は、IIS 4.0 と、FrontPage Server Extensions (FPSE) か Office Server Extensions (OSE) のどちらかをインストールする必要があります。

Windows NT 4.0 上の Windows NT 4.0 Option Pack から IIS 4.0 をインストールする場合、または Windows 2000 のコントロール パネルから IIS 5.0 をインストールする場合、FrontPage と InterDev を統合するため、次のコンポーネントが必要になります。その他のコンポーネントについては任意です。

- Internet Information Server (このコンポーネントを選択すると、Windows NT Option Pack 共通ファイルなど、IIS が依存しているほかの多くのコンポーネントも選択する必要があります)
- Internet Service Manager
- FrontPage Server Extensions (後で OSE をインストールする予定の場合、この時点で FPSE をインストールしなくてもかまいません)
- Windows Scripting Host

IIS をインストールするときに指定する Web 公開ホーム ディレクトリは、Web オーサリング統合の構成ウィザードが VOB タグの構築とタグの表示に使用します。例を次に示します。

drive-letter:¥view-tag¥VOB-tag

Web 公開ホーム ディレクトリとして C:¥webpub¥web\_content を指定すると、Web オーサリング統合の構成ウィザードは、統合用に次のビューと VOB を作成します。

- ビュー タグ webpub を使用するビュー
- VOB タグ ¥web\_content を使用する VOB

**注意:** IIS のインストールでは、デフォルトのパス名 ¥Inetpub¥wwwroot が Web 公開ホーム ディレクトリに使用されます。このデフォルトのパス名はできるだけ使用しないでください。既存のビューや VOB タグと競合するビューを生成しないパス名を指定することをお勧めします。

IIS をインストールした後で、Internet Service Manager を実行して IIS のオペレータを追加し基本認証を有効にする必要があります。Web コンテンツ VOB が IIS ホストにあれば、基本認証を有効にする必要はありません。

IIS 4.0 をインストールする前に Windows NT 4.0 サーバー上に Office 2000 をインストールした場合は、Windows NT 4.0 Option Pack から FPSE をインストールするように選択していても、FPSE はインストールされません。その場合、Windows NT 4.0 Option Pack の後で OSE をインストールし、バージョン 4.0 Server Extensions にアップグレードすることをお勧めします。

## 手順 2: FPSE または OSE のインストール

手順 1 で FPSE がインストール済みであれば、ここで再インストールする必要はありません。Option Pack のインストール時にバージョン 3 の FPSE がインストールされていれば、OSE にアップグレードすることができます。ただし、必須ではありません。

手順 1 で FPSE をインストールしなかった場合は、FPSE バージョン 3.0 (Windows NT 4.0 Option Pack から) またはバージョン 4.0 (Office 2000 CD3、Office Server Extensions、Windows 2000 のいずれかから) をインストールします。

### 手順 3: ClearCase のインストール

ClearCase または ClearCase LT をインストールします。

- ClearCase のインストールには、ローカル VOB とビュー サポートを含める必要がありますが、MVFS は不要です。
- デフォルトの ClearCase LT サーバー設定は、Web オーサリング統合のホストに適しています。

**メモ:** デフォルトの ClearCase LT サーバーのインストールには、RWP を含める必要があります。ClearCase LT サーバーで RWP と IIS を実行するには、一方または両方の Web サーバーを、互いに干渉しないように設定します。詳細については、付録 C を参照してください。

Web オーサリング統合の構成ウィザードではサーバーの記憶場所を指定する必要があります。通常、ClearCase LT のインストールでは、サーバー構成の一環として、サーバーの記憶場所が作成されます。サーバーの記憶場所の作成については、99 ページの「VOB のサーバー記憶場所の作成」を参照してください。

Web サーバーのホストにサーバーの記憶場所を作成することをお勧めします。この場所に Web コンテンツ VOB を作成すると、パフォーマンスが向上し、基本認証も不要になります。

### 手順 4: Web オーサリング統合の構成ウィザードの実行

Web オーサリング統合の構成ウィザードを実行して、画面の指示に従います。ウィザードの各ページでヘルプが使用できます。ウィザードを実行するときは、必ず [サーバー モード構成] を指定します。Web サーバー上で FrontPage 2000 のクライアントをインストールする予定がある場合にのみ、[ローカル モード構成] をオンにします。

既に Web サーバーに SourceSafe がインストールされていると、ウィザードの [SourceSafe の置き換え] 画面が表示されます。統合を行うには、[はい] をクリックする必要があります。それでも SourceSafe は、SourceSafe COM API を除き、使用できます。

Web オーサリング統合の構成ウィザードによって IIS と Server Extensions のバージョンがチェックされ、その後 Web コンテンツ VOB を作成するよう求められます。デフォルトの VOB 記憶場所をそのまま使用するか、ウィザードを使用して別の場所を選択します。既存の VOB を選択することもできます。

Web オーサリング統合を介した任意の VOB へのアクセスには、次のような制限事項があります。

- Web コンテンツ作成権限を持つすべてのグループを VOB のグループ リストに明示的に追加する必要がある (CLEARCASE\_PRIMARY\_GROUP 環境変数はサーバーに認識されません)。詳細については、protectvob リファレンス ページを参照してください。
- 対話型のトリガを使用できない。
- 統合機能を使用して作成した Web では、VOB シンボリック リンクを使用できない。

Web コンテンツ VOB の指定を終了すると、Web コンテンツ ビューを作成するように要求されます。統合機能で利用できるビューは、このウィザードを使用して作成したものか、FrontPage 2000 で ([ClearCase] の [Create Snapshot View] を使用して) 作成したものだけです。統合機能によって使用されるスナップショット ビューには、標準のスナップショット ビューの場合とは異なる方式の、ハイジャックされたファイルの検出機構が備えられています。その理由は、FrontPage や InterDev ではファイルの書き直しが頻繁に行われることにあります。ほかの方法で作成されたスナップショット ビューを使用しようとすると、エラーになります。統合機能では、UCM ビューは使用できません。

Web オーサリング統合の構成ウィザードでは、ccase-home-dir¥var¥log¥webintegration\_log にログ ファイルを維持管理しています。エラーや予期しない動作が発生した場合は、このログを確認してください。

## クライアントの設定の手順

---

この項では、ClearCase クライアント ホストで Web オーサリング統合機能を設定する方法について説明します。

**メモ:** Web オーサリング統合のユーザーがソース管理が有効な Web を開くたびに、そのユーザーは ClearCase ライセンスを取得します。

### 手順 1: クライアント アプリケーションのインストール

サポートされている次のクライアント アプリケーションのうち必要なものを、クライアント コンピュータ上にインストールします。

### 手順 2: ソース管理への Web の追加

FrontPage 98 または Visual InterDev 6.0 を使用して、新しい Web を IIS サーバー上のソース管理に追加します。

新しい Web をソース管理に追加するときに、統合機能ではその Web がトップレベルの Web と想定されます。たとえば、http://WebServer/WebName です。

### FrontPage 98 から

- 1 新しい Web を作成します。[ファイル]、[新規]、[FrontPage Web] の順にクリックし、FrontPage Web の新規作成ウィザードの画面に従って処理を完了します。
- 2 [ツール] の [Web の設定] をクリックして、新しい Web をソース管理に追加します。  
[FrontPage Web の設定] ダイアログ ボックスの [設定] タブをクリックし、手順 1 の新しい Web の名前を使用して、[ソース管理プロジェクト] ボックスに必要な情報を入力します。

[ソース管理プロジェクト] ボックスの Web 名は \$/ で始まらなければなりません。たとえば、[http://ccWebs/sales\\_collateral](http://ccWebs/sales_collateral) に Web を作成した場合、ソース管理プロジェクト名は `$/sales_collateral` でなければなりません。Web 名が実際の Web ディレクトリと違っていてもかまいません。Web ディレクトリをソース管理プロジェクト名として使用する必要があります。

## FrontPage 2000 またはそれ以降から

- 1 Web オーサリング統合の構成ウィザードを実行します。Web オーサリング統合の構成ウィザードで、[ローカルモード構成] を選択します。
- 2 FrontPage で新しい Web を作成します。[ファイル]、[新規作成]、[Web] の順にクリックし、[新規作成] ダイアログ ボックスで必要な情報を入力します。
- 3 [ClearCase] の [Add Web to source control] をクリックして、新しい Web をソース管理に追加します。

## Visual InterDev 6.0 から

ClearCase の統合機能と共に Visual InterDev 6.0 を使用するために、クライアント コンピュータ上に ClearCase をインストールする必要はありません。クライアントに ClearCase がインストールされていない場合は、Visual InterDev 統合機能のヘルプは使用できません。

- 1 [ファイル] の [新しいプロジェクト] をクリックして、新しい Web プロジェクトを作成します。
- 2 Visual InterDev Project Explorer からプロジェクトを選択し、[プロジェクト]、[ソース管理]、[ソース管理に追加] の順にクリックして、ソリューションを追加するダイアログ ボックスを表示します。
- 3 [選択] をクリックして、[ソース管理を使用する] ダイアログ ボックスを表示します。
- 4 [ソース管理プロジェクト名] ボックスで、デフォルトのテキストからサフィックス `_Web` を削除します。たとえば、Web プロジェクトの名前が `stock_trading` で、デフォルトのテキストが `$/stock_trading_Web` なら、正しいプロジェクト名は `$/stock_trading` になります。
- 5 [OK] をクリックします。これで、プロジェクトが IIS サーバー上のソース管理に追加されました。

### 手順 3: 新しい Web コンテンツがソース管理に追加されたことの確認

新しい Web コンテンツを FrontPage または Visual InterDev のソース管理に追加したら、追加が完了したことを示すソース管理アイコンを探します。FrontPage では、それは緑色の点です。Visual InterDev では、鍵です。

ソース管理アイコンが表示されていないか、エラーまたは予期しない結果に遭遇したら、インストールエラーが発生した可能性があります。ログ ファイル `ccase-home-dir\var\log\ssapi_log` にエラーに関する情報が書き込まれている場合があります。

既存のコンテンツをフラット ファイル内かほかの Web サーバー上に保持している場合、FrontPage と Visual InterDev のインポート機能を使用して、ファイルを新しい Web にインポートし、それをソース管理下に置くこともできます。

### 手順 4: 権限の設定

FPSE は Web のオーサリングと管理に関する作業のアクセス制御を管理します。Web のコンテンツを編集するには、ユーザーまたはグループは Web に対して Author の権限を持つ必要があります。Web のコンテンツにアクセスする必要があるすべてのユーザーとグループを追加します。

#### FrontPage 98

- 1 [ファイル] メニューの [FrontPage Web を開く] をクリックして、新しい Web にアクセスします。
- 2 [ツール] の [権限] をクリックします。[権限] ダイアログ ボックスの [ユーザー] タブまたは [グループ] タブをクリックして、ユーザーまたはグループに権限を追加します。

#### FrontPage 2000

- 1 [ファイル] の [Web を開く] をクリックして、新しい Web にアクセスします。
- 2 [ツール] から、[セキュリティ]、[権限] の順にクリックします。[権限] ダイアログ ボックスの [ユーザー] タブまたは [グループ] タブをクリックして、ユーザーまたはグループに権限を追加します。

#### Visual InterDev 6.0

- 1 [ファイル] の [プロジェクトを開く] をクリックして、新しい Web にアクセスします。
- 2 [プロジェクト] から、[Web プロジェクト]、[Web の権限] の順にクリックします。[権限] ダイアログ ボックスの [ユーザー] タブまたは [グループ] タブをクリックして、ユーザーまたはグループに権限を追加します。

## FrontPage 2000 またはそれ以降のためのローカル モードでのクライアントの設定

---

統合には、サーバー モードとローカル モードという 2 つのモードがあります。

- サーバー モードは、対応しているすべてのオーサリング ツールで使用でき、使用が制限されている一部の ClearCase の機能も使用できます。サーバー モードでは、オーサリング ツールはローカルで稼動し、ClearCase の処理はすべて IIS ホスト上でリモート処理されます。すべてのユーザーは Web サーバー上で単一のスナップショット ビューを共有します。Web を開くには、URL にアクセスします。クライアント システムに ClearCase をインストールする必要はありませんが、インストールすれば、ユーザーはオンライン ヘルプだけではなく、FrontPage 2000 内の [ClearCase] メニューの一部と Visual InterDev 内の ClearCase ツール バーを使用できるようになります。
- ローカル モードでの運用は FrontPage 2000 またはそれ以降でのみ可能です。ただし、FrontPage が稼動しているコンピュータ上に ClearCase をインストールする必要があります。ローカル モードの場合は、FrontPage GUI で使用できる ClearCase メニューなど、その他の機能も使用できます。Web を開くには、ユーザー独自のスナップショット ビュー内にある Web 用のパス名にアクセスします (ディスク ベースの Web)。ローカル モードを使用すると、複数のビューを使用できると共に、ClearCase の GUI すべてにアクセスすることができます。ローカル モードで行われるチェックアウトはデフォルトの設定ではチェックインを予約しません。

FrontPage 2000 用にローカル モードを設定するには

- 1 FrontPage 2000 またはそれ以降をインストールします。
- 2 ClearCase をインストールします。
- 3 Web オーサリング統合の構成ウィザードを実行します。Web オーサリング統合の構成ウィザードで、[ローカル モード構成] を選択します。
- 4 FrontPage で [ClearCase] の [Create Snapshot View] をクリックして、ビュー作成ウィザードを完了します。

FrontPage の共有枠を使用して Web にアクセスするときは、ローカル モードのビュー内で共有枠を適用し直す必要があります。サーバー モードのアクセスだけが使用されている場合以外は、FrontPage のテーマを使用しないことをお勧めします。テーマの結合情報はソース管理には追加されません。テーマの再適用は時間のかかる作業で、同じファイルの多くの余分なバージョンがチェックインされます。

ローカル モードを使用するには、サーバー モードのユーザーが共有ビューを定期的に更新する必要があります。ローカル モードのユーザーがいない場合、または Web コンテンツ VOB へのその他の書き込みアクティビティがない場合、共有ビューの更新は必要ありません。自社の環境内で、ローカル モードとサーバー モードによるアクセスを混在させ適当に組み合わせることができます。

## Web Folders への対応

---

サーバー モードの統合オプションを使用して、ClearCase を Office 2000 や Internet Explorer 5.0 Web Folders と統合するように、IIS Web サーバーを設定することもできます。

Office 2000 のアプリケーションまたは Internet Explorer 5.0 またはそれ以降を使用している場合、Web Folders の機能を使用して IIS Web サーバーにログオンし、Web コンテンツの共有ビューに Web アドレスとしてアクセスすることができます。

ClearCase と Web Folders との統合では、次の 2 つのケースがサポートされます。

- ファイルがソース管理されている場合、ファイルを保存すると、それが Web Folder にコピーされます。そのファイルが ClearCase によってチェックインされます。
- ファイルがソース管理されていない場合、ファイルを保存すると、それが Web Folder にコピーされます。そのファイルが ClearCase によってソースコントロールに追加され、チェックインされます。

## Web サーバー上の共有ビューの更新

---

共有 Web コンテンツ ビューはときどき更新する必要があります。この機能は Web サーバー上でスクリプトによって実行されます (ccase-home-dir\etc\iisfix.bat)。ユーザーは自分のクライアント システムからこの機能にアクセスできます。これには、URL [http://WebServerName/ccase\\_tools](http://WebServerName/ccase_tools) を開いて、[Update Shared View] をクリックします。このスクリプトを IIS サーバー上で定期的に行うことをお勧めします。その際、Windows NT のタスク スケジューリング サービスを利用し、タスクを実行するアカウントに適切なユーザーを指定します。この目的で、Windows NT の at コマンドまたは ClearCase のスケジューラを使用することはできません。

## データの移行と変換

---

既存の Web 上のデータと統合することもできます。場合によっては、既存の VOB からデータを移行したり、ほかのソース管理製品からデータをインポートする必要があります。

- **移行**: 統合機能では、ソース管理下に置かれている Web が、Web コンテンツ VOB ルートのサブディレクトリをルートにしていると想定されます。cleartool relocate コマンドを使用して、Web を既存の VOB から Web コンテンツ VOB に再配置する場合は、Web を VOB ルート サブディレクトリに配置してください。
- **インポート**: Visual SourceSafe 管理下の Web を ClearCase にインポートするには、Web コンテンツ ファイルがある Visual SourceSafe ライブラリで clearexport\_ssafe を実行します。さらに clearimport コマンドを使用して、Web コンテンツ VOB 内にターゲットのディレクトリを指定します。データの ClearCase へのインポートについての詳細は、第 13 章を参照してください。

どちらの方法でも、スクリプト (ccase-home-dir¥etc¥iisfix.bat) を実行して FPSE をインストールし、IIS 別名とソース管理操作を登録し、統合を可能にする必要があります。

**メモ:** FPSE 2002 を使用している場合は、Rational カスタマ サポートから新しい iisfix.bat スクリプトを入手する必要があります。

それが完了すると、FrontPage や Visual InterDev 内で Web を開いたときに、ソース管理アイコンが表示されます。

## 統合機能のヘルプへのアクセス

---

クライアント システム上に ClearCase をインストールすると、統合機能に関するオンライン ヘルプが使用できるようになります。Visual InterDev では [ヘルプ] アイコンを備えた ClearCase のツールバーが、FrontPage 2000 またはそれ以降ではヘルプ コマンドなどの ClearCase メニューが利用可能になります。このヘルプは ClearCase がインストールされていないホストでは使用できません。



VOB に最終的に含まれるエレメント、バージョン、派生オブジェクトの数の見積もりが可能な場合、それらを収容する VOB サイズの概算見積もりを計算できます。VOB 記憶領域には、データベースとプールがあります。Windows の場合、データベースとプールは単一のディスクパーティション上に存在する必要があります。UNIX の場合は、シンボリック リンクを使用して、プールを別のパーティションまたはホスト上に配置できます。詳細については、118 ページの「UNIX ホストでのリモート記憶プールの作成」を参照してください。

## データベース容量の見積もり

VOB データベースの作業セットをメモリ内に保持することにより、VOB メタデータへのアクセス時間が大幅に削減されます。そのため、97 ページの「VOB サーバー ホスト構成のガイドライン」では、ホストする各 VOB データベースの半分を保持するのに十分な物理メモリで VOB サーバーを構成することが推奨されています。

表 18 に、VOB データベースの容量消費に対する 4 つの重要なコントリビュータのサイズ (バイト数) を示します。

**表 18 一般的な VOB データ タイプのサイズ**

データ タイプ	スキーマ バージョン 53	スキーマ バージョン 54
バージョンが 1 つのエレメント	2770	3970
エレメントの各追加バージョン	446	642
非共有 DO (構成レコードを除く)	1356	1924
共有 DO (構成レコードを除く)	60	118

**メモ:** すべての DO に構成レコードが含まれます。構成レコードのサイズは、エレメントの内容と、DO がどのように構築されたか (ビルド スクリプトとコンパイラ オプション) によって決まります。見積もりのためには、構成レコードにより、非共有 DO のサイズに 2 ~ 10 パーセント、共有 DO のサイズに 50 ~ 100 パーセントが追加されると想定できます。

ユーザーがこれらのオブジェクトをいくつ VOB 内に作成するかの見積りは困難です。通常、エレメント数の見積もりは容易です。特に、別のソース コード管理システムの内容を VOB に

入れる場合などです (第 13 章を参照)。バージョン数の見積もりは困難です。一部の要素は、ほかの要素よりも非常に多くのバージョンを累積します。ほかの VOB 内の代表的な要素のバージョン数を、バージョンの作成にどのように影響するかインジケータとして、ソフトウェア開発メソッドで使用できます。作成する派生オブジェクトの数は、ビルドターゲット数と各ターゲット内の要素数の関数です。ClearCase LT を使用する場合、または ClearCase を使用するが動的ビューや clearmake または omake は使用しない場合、ユーザーは派生オブジェクトを作成しません。

VOB に含まれる要素数の見積もりがあれば、表 18 の情報を使用して、次の式により VOB のデータベース サイズを概算で計算することができます。

$$S = N (E + (e * (V - 1)) + D * d)$$

各パラメータの内容は次のとおりです。

- S は、データベース サイズ (バイト数) です。
- N は、VOB 内の要素数です。
- E は、バージョンが 1 つの要素のサイズです。
- e は、要素の各追加バージョンのサイズです。
- V は、要素ごとの平均バージョン数です。
- D は、要素ごとの非共有派生オブジェクト数です。
- d は、非共有派生オブジェクトのサイズです。

次の例では、この式を使用して、要素数 500、要素ごとの平均バージョン数 20、派生オブジェクト数 5 で、スキーマ バージョン 53 形式の VOB のサイズを見積もります。

$$500 * (2770 + (446 * (20 - 1))) + 5 * 1356 = 901,200 \text{ バイト}$$

この見積もりは、プランチタイプとインスタンス、属性タイプとインスタンスなど、ほかの VOB メタデータのサイズには適用されません。これらのデータタイプでは、データベースは一般的に小さなサイズですが、一部の VOB では、特に UCM が使用されている場合は、大きくなる場合があります。

## プール容量の見積もり

---

VOB データベース内で消費する容量以外に、各要素は VOB のソース プール内や、通常その他のプール内にも容量を必要とします。

VOB 内のすべての要素のすべてのバージョンは、ソース プール内のコンテナ ファイルに保存されます。要素タイプが異なれば、タイプ マネージャも異なります。ほとんどのタイプ マネージャは、ソース プール内での各バージョンの容量消費が cleartext 形式よりも少なくなるように、ソース データを圧縮します。表 19 に、一般的に使用される、組み込みのタイプ マネージャの圧縮率を示します。

**表 19 タイプ マネージャの圧縮率**

タイプ マネージャ	タイプの説明	圧縮率
whole_copy	オブジェクトのコピー全体	1:1
text_file_delta	印刷可能テキストのインターリーブされた変分	50:1
z_whole_copy	圧縮された whole_copy	2.5:1
z_text_file_delta	圧縮された text_file_delta	200:1

表 19 の情報を使用して、次の式により VOB ソース プールのサイズの概算見積もりを計算することができます。

$$P = N * V * S / C$$

各パラメータの内容は次のとおりです。

- P は、ソース プールのサイズ (バイト数) です。
- N は、VOB 内のエレメント数です。
- V は、エレメントごとの平均バージョン数です。
- S は、各 cleartext エレメント バージョンの平均ファイル サイズです。
- C は、エレメントのタイプ マネージャの圧縮率です。

次の例では、この式を使用して、タイプ `text_file_delta` のエレメント数 500、エレメントごとの平均バージョン数 20、平均バージョン サイズ 50000 バイトで、VOB のソース プールのサイズを見積もります。

$$500 * 20 * 50000 / 50 = 10,000,000 \text{ バイト}$$

異なるタイプのエレメントを含む VOB の場合、エレメント タイプごとにこの計算を繰り返してください。

エレメントは、cleartext と DO プール内にも容量を必要とします。これらは、増加を制御するために定期的にスクラブされるキャッシュです。cleartext と DO プールの容量の要件はサイト固有の使用パターンによって決定されるので、VOB 内のエレメント数に基づいてサイズを見積るための一般的なルールは提示できません。

VOB 記憶プールの詳細については、111 ページの「VOB 記憶プール」を参照してください。



# 索引

## A

admin\_server プロセス 10

albd\_server プロセス 9, 10

## C

ccase-home-dir ディレクトリ xxxiv

checkvob コマンド

UCM 環境での使用 254

概要 242

グローバル タイプの問題の解決 252

壊れたハイパーリンクの修正 254

動作 244, 245

レプリカでの考慮事項 242

CIFS プロトコル 62

ClearCase File Service (CCFS) 64

ClearCase LT、ClearCase との違い 2

ClearCase オブジェクトのアクセス コントロール

OS の混在した環境 59

VOB オブジェクト 79

VOB のオブジェクトへのアクセス 76

エレメント 81

グループと VOB ユーザー 107

グローバル タイプ、保護の変更 137

仕組み 73

スケジューラのタスクとジョブ 230

動的ビュー 86

派生オブジェクト 91

ビュー オブジェクト 85

ビュープライベート ファイル 88

プール 83

メタデータ 83

ユーザーとグループ 73

ユーザー プロセス認証 76

リモート root ユーザー 75

ロック 85

ClearCase 管理者グループ 61

ClearCase サーバー プロセス

エラー ログ 13

開始と停止 14

概要 8

ClearCase サーバー プロセス ユーザー

Samba マッピング 322

TAS マッピング 328

UNIX の ID 60

概要 60

グループ メンバシップ 75

ClearCase レジストリ

新しいリージョンでのタグの作成 51

オブジェクトとタグの変更 46

概要 33

管理のガイドライン 52

クライアント リスト 44

サイトのデフォルト値 42

サーバー ホスト名の変更 52

データ モデル 34

バックアップ 54, 206

ホストの割り当て 45

リージョン 38, 39

レジストリ パスワード 43, 44

ClearCase レジストリ サーバー ホスト

概要 42

定義 43

バックアップからプライマリへの  
プロモート 54, 56

バックアップの定義 54

バックアップ ホストがない場合の復元 57

プライマリの復元 56

プライマリの変更 54

割り当ての変更 45

ClearCase レジストリ リージョン

削除 52

追加 50

clearexport\_pvcs ユーティリティ 266  
clearexport\_rcs ユーティリティ 265  
clearexport\_ssafes ユーティリティ 262  
clearimport ユーティリティ 263, 265, 267  
clearlicense ユーティリティ 22  
ClearQuest データベース、バックアップからの  
復元 226  
cleartext プール  
概要 112  
使用方法 119  
バックアップ 203  
CPU 処理能力、VOB ホスト 98  
cquest-home-dir ディレクトリ xxxiv  
Credentials Manager サービス 15, 179  
credmap\_server プロセス 10

## D

db\_server プロセス 11  
DCHP のガイドライン 17  
DO (派生オブジェクト)  
アクセス コントロール 83, 91  
共有可能 124  
共有不可 177  
スクラブ 240  
動的ビューでの管理方法 178  
非共有 176  
復元したビュー 225  
DO プール  
Windows でのコンテナ保護 246  
概要 112  
欠落したコンテナ 249  
参照不能データ コンテナ 250  
使用方法 119  
使用方法とバックアップ 202

## F

fix\_prot ユーティリティ 273

## I

insert\_cr テキスト モード 69

## L

lockmgr プロセス  
概要 12  
起動オプションと VOB ホストの  
パフォーマンス 308  
lost+found ディレクトリ 116, 177  
lsacl コマンド 275

## M

mvfsstat コマンド 301  
MVFS キャッシュ  
使用可能な統計情報 302  
スケーリング係数 289  
統計情報 288  
動的ビューのパフォーマンス 288  
MVFS (マルチバージョン ファイル システム)  
Windows での制限事項 180  
大文字と小文字の区別 67  
概要 179  
監視付きビルド 180  
再帰探索、UNIX ビューのルート  
ディレクトリ 285  
設定、大文字と小文字の区別 66  
データ コンテナの検索 281

## N

NAS デバイス  
ClearCase 16  
VOB サーバー ホストの置換 105  
VOB の移動 170  
VOB の作成 102  
VOB バックアップの復元 219  
概要 4

サーバーの記憶場所 100  
ネットワーク アクセスの設定 17  
ビューの作成 185  
NFS クライアント製品 313  
NFS プロトコル 62

## P

PVCS データ、インポート 266

## R

Rational License Key Administrator 31  
Rational Web Platform (RWP)  
    概要 355  
    構成情報 356  
    プロキシ アクセス 362  
    ホスト 4  
RCS データ、インポート 265  
relocate コマンド、動作 147  
rgy\_backup コマンド 54  
rgy\_switchover コマンド 54  
rmelem コマンド 116  
rmname コマンド 116  
root ユーザー、リモートの権限 75  
rwp プロセス 12

## S

Samba、ClearCase サポート用の設定 322  
scrubber ユーティリティ 238, 239  
setuid の実行、VOB での有効化 108  
SMB プロトコル 62  
SourceSafe データ、インポート 259  
strip\_cr テキスト モード 69

## T

TAS (TotalNET Advanced Server)、  
ClearCase 用の設定 326

## U

UCM プロジェクト  
    checkvob の使用 254  
    データのインポート 259  
UNIX/Windows の相互運用  
    ClearCase オブジェクトへのアクセス 60  
    NAS デバイス 63  
    Windows のファイル名に使用できない文字 71  
    概要 59  
    異なるプラットフォーム間でのファイル アクセス  
        ソリューション 313  
    認証 60  
    ビューのテキスト モード 67  
    ユーザー認証 59  
UNIX カーネル、VOB ホスト上のリソース 98

## V

view\_server プロセス  
    概要 11  
VOB  
    PVOB 属性 94  
    アクセスの禁止 108  
    アクセスの問題、解決 106  
    概要 6  
    「管理 VOB 階層」、「コンポーネント VOB」、  
        「レプリカ」も参照  
    機能レベル 95  
    削除 53, 109  
    作成 101, 102, 103  
    スナップショット バックアップ 198  
    テキスト モードのサポート 70

- 登録 49
- バックアップからの復元 211
- バックアップ方法 196
- パブリックとプライベート 103
- ビューとの同期 222
- マウント、ビュー アクセス 36
- ロックとアンロック 108
- オブジェクト、VOB
  - 「エレメント」、「グローバル タイプ」、  
「タイプ オブジェクト」も参照
- vob\_restore ユーティリティ 212, 213
- vobrpc\_server プロセス 12
- vob\_scrubber ユーティリティ 238
- vob\_server プロセス 11
- vob\_snapshot ユーティリティ 198
- VOB タグ
  - 削除 53, 108
  - 内容と構造 36
  - 内容の表示 48
  - 不明 53
  - 変更 49
  - 命名規則 39
- VOB データベース
  - db\_server プロセス 11
  - スキーマ バージョン 95
  - スクラブ 239
  - ディスク容量要件 98
  - 内容と構造 112
  - メモリ要件 97
- VOB の移動
  - NAS デバイス 170
  - UNIX から Windows 166
  - Windows から UNIX 163
  - Windows ドメイン 154
  - 概要 151
  - 異なるオペレーティング システム 162
  - 注意 152
  - リモート記憶プール 158
- VOB のオブジェクト
  - アクセス コントロール 76
  - 概要 123
  - ロック 85

- VOB へのデータのインポート
  - PVCS 266
  - RCS 265
  - SourceSafe 259
  - 概要 259
- VOB ホスト
  - Lock Manager の起動オプション 308
  - UNIX カーネル、リソースの調整 98
  - UNIX、リモート記憶プール 118
  - 概要 4
  - サーバー プロセスの構成 97
  - 多数の VOB 308
  - ディスク サブシステムの推奨事項 306
  - パフォーマンスのヒント 305
  - 複数のネットワーク インターフェイス 18

## W

- Web ビュー
  - 概要 174

## い

- イベント レコード 124

## え

- エラー ログ、サーバー プロセス 13
- エレメント
  - アクセス コントロール 81
  - タイプの変更 127
  - 取り残された 116
  - バックアップからの復元 219
  - 別の VOB への再配置 147
- エレメントの再配置
  - エラー処理 149
  - 概要 147
  - 結果の確認 151
  - シンボリック リンクの修正 278

注意 148  
ビューの問題の解決 276  
ボーダーラインのエLEMENT 149

## お

大文字と小文字を区別するルックアップ  
MVFS の設定 66  
OS の混在した環境での問題 65  
オブジェクト (VOB)  
アクセス コントロール 79  
概要 34, 123

## か

カスタマ サポート xxxvi  
管理 VOB 階層  
VOB の削除 132, 133  
VOB の追加 130  
VOB の復元 228  
VOB のリンク 104  
概要 128  
構造 128  
制限 131  
バックアップ 132, 206  
レプリカ 141  
管理 VOB ハイパーリンク 128, 130, 132

## き

記憶ディレクトリ  
ACL のコピー時の保存 271  
ACL 問題の解決 269  
Samba 共有の作成 324  
VOB、内容と構造 109  
VOB のアクセス要件 99

アクセスの問題、トラブルシューティング 107  
位置の確認 200  
スナップショット ビュー、内容と構造 183  
動的ビュー、内容と構造 175

### 記憶プール

アクセス コントロール 83  
一般的な問題の解決 245  
概要 111  
欠落したデータ コンテナと参照不能  
データ コンテナ 247  
使用方法 118  
スクラブ 238  
追加、作成 118  
名前と保護の問題 246  
不整合の原因 243  
別のディスクへの移動 120  
リモート、UNIX ホスト 118  
リモート、VOB バックアップ 201  
リモート、Windows NFS クライアントによる  
アクセス 319  
リモート、検索 158  
リモート、統合 159

### 記憶領域の管理

VOB での方法 114, 115  
VOB の増加の制御 238  
ビュー用のディスク容量 255  
プールの追加 117

規則、表記 xxxiv

機能レベル 96

機能レベル、VOB 95

## く

クライアント ホスト  
構成のガイドライン 287  
複数のネットワーク インターフェイス 18  
グループの管理 8  
グループ、プライマリ 74  
グループ名、UNIX と Windows 59

## グローバル タイプ

- acquire オプション 134
- checkvob 252
- VOB へのコピー 139
- 削除 141
- 作成 134
- 名前の変更 139
- 不整合状態の修正 141
- プロパティ 135
- 保護の変更 137
- マスターシップ 143, 145
- リスト 136
- 利点 128
- 履歴情報 137
- ローカル タイプの非表示 134
- ローカルの変換 139
- ロックとアンロック 138

## け

- 権限のあるユーザーとグループ 74

## こ

### 構成レコード

- 概要 125
- ビュー記憶領域 177

### コンポーネント VOB

- オブジェクトのアクセス コントロール 83
- 概要 94
- バックアップからの復元 226

## さ

### サイトのデフォルト値

- 変更 50

### サーバーの記憶場所

- 概要 41
- 作成 99, 100

## し

### 自動マウント機能

- ClearCase 19
- 非 ClearCase アクセス 375

## す

### スキーマ バージョン

- VOB 移動時の影響 163
- VOB データベース 95

### スクラブ

- VOB データベース 239
- 概要 115, 238
- 記憶プール 238

### スケジューラ

- 概要 114, 229
- ジョブ通知 235
- ジョブの管理 233
- ジョブの削除 237
- ジョブの作成 233
- ジョブの実行 237
- ジョブのスケジュール 234
- ジョブのプロパティの表示 236
- ジョブのプロパティの編集 237
- スクラブ ジョブの間隔調整 241
- タスクの削除 233
- タスクの作成 231
- タスクの編集 232

### スナップショット ビュー

- 概要 174, 182
- バージョンの選択 184
- ビュー記憶ディレクトリの内容 183
- ビュー ディレクトリ 183
- ビュー データベース 184

## そ

### ソース プール

- Windows でのコンテナ保護 246

エレメントの割り当ての変更 119

概要 111

参照不能データ コンテナ 248

使用方法 118

データ コンテナの欠落 247

## た

タイプ オブジェクト

インスタンス 126

概要 125

「グローバル タイプ」も参照

定義方法 127

範囲 127

ほかのタイプへのインスタンスの変換 127

ロック 85

## て

テキスト モード

VOB でのサポート 70

概要 67

選択基準 69

タイプ 68

レプリカでのサポートの有効化 71

## と

透過テキスト モード 68

動的ビュー

DO の管理 178

アクセス コントロール 86

概要 173, 174

記憶ディレクトリへのアクセス 53

バージョンの選択 178

ビュー データベース 177

ビュープライベート記憶領域 176

復元、クリーン アップ 223, 225

ドキュメント

ヘルプの説明 xxxv

トラブルシューティング

MVFS データ コンテナの検索 281

UNIX ビューのルート ディレクトリ、  
cronjob による再帰探索 285

記憶ディレクトリの保護 269

再配置されたエレメントとビュー 276

取り残されたファイル、原因 116, 177

## ね

ネットワーク ファイル システム プロトコル 62

## は

ハイパーリンク、壊れたハイパーリンクの修正 254

バージョン、削除 116

バージョンの選択

スナップショット ビュー 184

動的ビュー 178

バックアップ

ClearCase レジストリ データ 54, 206

DO ブール 202

vob\_snapshot プロシージャ 198

VOB、関連付けられたデータベース 206

VOB のロックの重要性 200

VOB、方法 196

VOB、リモート記憶ブール 201

概要 195

管理 VOB 132

タイプ、VOB 196

ツールの要件 195

ビュー 204

バックアップからのデータの復元

ClearQuest データベース 226

NAS デバイスの VOB 219

PVOB 227

vob\_restore 212

VOB とビューの同期 222

VOB、標準的な手順 217  
概要 211  
管理 VOB 228  
関連付けられたデータベース 225  
個々のエレメントの復元 219  
コンポーネント VOB 226  
ビュー 220

## ひ

非 ClearCase アクセス、構成 371  
ビュー  
    NAS デバイスでの作成 185  
    VOB との同期 222  
    アクセス コントロール 85  
    一時的な無効化 193  
    移動 186  
    エクスポート、非 ClearCase アクセス 372  
    大文字と小文字を区別するファイル  
        ルックアップ 67  
    概要 7, 173  
    記憶領域のメンテナンス タスク 254  
    起動 37  
    キャッシュ サイズ、再設定 298  
    キャッシュ サイズ、設定方法 296  
    キャッシュの統計情報 297  
    削除 53, 193  
    作成方法 185  
    タイプ 173  
    テキスト モード 67, 69  
    登録 49  
    バックアップ 204  
    バックアップからの復元 220  
    リモート記憶領域 185  
ビュー オブジェクト 35  
ビュー タグ  
    削除 53  
    内容と構造 37  
    内容の表示 48  
    不明 53  
    変更 49

ビュー データベース  
    スナップショット ビューの内容 184  
ビューの移動  
    新しいホストのアーキテクチャ 187, 189  
    概要 186  
ビュープライベート ファイル、  
    アクセス コントロール 88  
ビュー ルート 175  
表記規則 xxxiv  
ビルド監視、MVFS 180

## ふ

ファイル システム オブジェクトのアクセス  
    コントロール  
    概要 92  
ファイル システム オブジェクトへのアクセス  
    コントロール  
    一般的な保護の問題 272  
    保護の問題の解決 275

## へ

ヘルプ、アクセス xxxv

## ほ

ホスト  
    ClearCase 4  
    ClearCase LT 3  
    ClearCase ライセンス サーバー 4  
    ClearCase レジストリ サーバー 4  
    非 ClearCase アクセス用の構成 371  
    複数のネットワーク インターフェイス 18  
    リリース領域 3  
    レジストリの割り当ての変更 45  
ポートの割り当て  
    albd\_server process 10  
    Rational Web Platform 357

## ま

マスターシップとグローバルタイプ 143, 145  
マルチバイト文字、タグ名 41

## め

命名規則  
VOB 39  
ファイル名、大文字の小文字の混在 66  
メタデータ  
アクセス コントロール 83  
概要 93

## ゆ

ユーザーの管理  
UNIX と Windows でのユーザー名 59  
概要 8

## ら

ライセンス (ClearCase)  
概要 21  
除外されたユーザー 28  
有効期限 28  
ユーザー アクティビティの監視 22  
ユーザーの優先順位 22, 28  
ライセンス キー 27  
ライセンス (ClearCase LT)  
License Key Administrator 31  
Rational Suite ライセンス 30  
UNIX クライアント 31  
Windows クライアント 31  
概要 29  
タイプ 29, 30  
有効期限 30

ライセンス サーバー ホスト  
新しいライセンスの追加 24  
指定方法 22  
設定 23  
追加、設定 24  
名前の変更 27  
ライセンスの移動 26  
ライセンス データベース  
内容と形式 27  
場所 23

## り

リリース領域 3

## れ

レジストリ  
概要 6  
レプリカ  
checkvob の考慮事項 242  
移動 153  
管理 VOB 階層 141  
機能レベル 95

## ろ

ロック  
VOB 108, 109  
グローバルタイプ 138

