

Rational™ ClearQuest®

管理ガイド

バージョン: 2003.06.10 およびそれ以降

G126-5406-00

UNIX/WINDOWS 版

法的通知

Copyright ©1997-2003, Rational Software Corporation. All Rights Reserved.

バージョン番号: 2003.06.10

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

ライセンシーは、本ソフトウェアおよびドキュメントが、DFARS 252.7202、DFARS 252.211、FAR 2.101、または FAR 52.227 (またはその改訂された規定) のもとに、商業的コンピュータ ソフトウェア、商業的資料、または制限されたコンピュータ ソフトウェアとして提供されることに同意するものとします。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エLEMENT。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/) 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

まえがき	xxv
本書について	xxv
ClearQuest マニュアル ロードマップ	xxvi
ClearQuest とほかの Rational 製品との統合	xxvii
表記規則	xxviii
カスタマ サポート	xxix
1. ClearQuest 管理について	1
変更管理の概念的な理解	1
変更管理が必要な理由	1
変更依頼のライフサイクル	2
変更管理プロセスモデル	2
ClearQuest: 変更管理に対する障害/変更追跡システム	4
ClearQuest スキーマ、スキーマ リポジトリ、データベース、接続の定義	5
スキーマ	5
スキーマ リポジトリ	5
データベース	6
接続	7
製造元データベース管理システム	7
ClearQuest のアーキテクチャとコンポーネントの理解	8
Windows インストール用 ClearQuest のアーキテクチャとコンポーネント	8
カスタマイズ: フックと ClearQuest API	10
UNIX インストール用 ClearQuest のアーキテクチャ	11
ClearQuest 管理者の役割の理解	13
システムの設計と実装	14
データベース管理システムの設定	15
ClearQuest のインストール	15
ClearQuest ユーザー グループへの加入	15
スキーマ リポジトリと接続の作成と管理	15
コード ページ タスクの実行	16
ユーザー データベースの作成	16

スキーマ (プロセス モデル) の作成	16
ユーザー データベースの管理	18
Rational E-Mail Reader を使用した電子メール通知の有効化	18
Web アクセスの有効化	19
データのインポートとエクスポート	19
クエリー、グラフ、レポートの作成	19
ユーザー アカウントの管理	19
レコードに対するセキュリティ制御の設定	20
管理機能の委託	20
ClearQuest MultiSite のセットアップ	21
ClearQuest 入門	21
管理アカウントのパスワードの設定	21
ツールバー、パレット、メニュー、ショートカット	22
ツールバーの用途	22
パレットの用途	22
メニューの用途	22
ショートカット メニューの用途	22
キーボード ショートカットとマウス操作	23
2. ClearQuest とコード ページ	27
コード ページのタスクの構成	28
ClearQuest のデータ コード ページの概要	28
ClearQuest のデータ コード ページと新規スキーマ リポジトリ	28
ClearQuest のデータ コード ページと既存のスキーマ リポジトリ	29
ClearQuest データ コード ページの選択	29
ClearQuest のデータ コード ページを選択するためのガイドライン	29
Rational によってサポートされている Windows のコード ページ	31
ClearQuest のデータ コード ページを選択した結果	31
ClearQuest のデータ コード ページ値の変更	32
製造元データベースの文字セットの概要	33
例	33
製造元データベースの文字セットの設定	34
ClearQuest のデータ コード ページ値の設定	35
新規スキーマ リポジトリ	35
既存のスキーマ リポジトリとユーザー データベース	37

以前のバージョンの ClearQuest との相互運用	39
CharacterSetValidation パッケージの概要	39
CharacterSetValidation パッケージの適用	40
ClearQuest のデータ コード ページ値のユーザー データベースへのコピー	41
MultiSite と ClearQuest のデータ コード ページ値	41
MultiSite による同期の失敗の例	42
ClearQuest 統合と ClearQuest のデータ コード ページ値	42
例	43

3. データベースの管理 45

データベースの操作プロセスの理解	46
データベースをバックアップする手順の確立	49
空の製造元データベースとデータベースのエイリアス (別名) の作成	50
製造元データベースの文字セットの設定	50
新規スキーマ リポジトリとサンプル データベースの作成	51
ユーザー データベースの作成	54
接続の管理	58
新規接続の作成	59
接続の変更	59
接続の名前の変更	59
接続の編集	60
接続の削除	61
既存の接続の複製	61
UNIX クライアントとの接続の維持	61
cqreg のオプション	62
接続プロファイルの使用法	63
接続プロファイルのエクスポート	63
接続プロファイルのインポート	64
スキーマ リポジトリの移動	65
スキーマ リポジトリ移動の準備	65
スキーマ リポジトリの移動	66
製造元データベースのツールを使用したスキーマ リポジトリの移動	67
既存の接続の更新	67
接続を使用しないスキーマ リポジトリの更新	68

データベースを ClearQuest の新しい機能レベルにアップグレードする	68
SQL Anywhere 5.0 から SQL Anywhere 8.0 へのアップグレード.....	69
既存の接続のアップグレード.....	69
接続を使用しないスキーマ リポジトリとユーザー データベースのアップグレード.....	70
ユーザー データベースの移動と削除.....	71
ユーザー データベース移動の準備.....	71
ユーザー データベースの移動	71
ユーザー データベースのプロパティの更新.....	72
ユーザー データベースの削除	74
削除されたデータベースの復元	75
データベース プロパティの表示.....	75
製造元データベースのパラメータ	76
Microsoft Access	77
SQL Anywhere	77
SQL Server	78
Oracle	79
DB2	80

4. ClearQuest スキーマの操作 81

スキーマの変更手順	82
テスト データベースの作成.....	82
スキーマのチェックアウト	84
スキーマを読み取り専用で開く	85
Blank スキーマからの新規スキーマの作成	86
スクリプト言語の選択.....	87
スキーマの変更の確認.....	88
スキーマ用テスト データベースの設定.....	89
テスト データベースでのスキーマのテスト.....	90
スキーマのチェックイン.....	91
スキーマのチェックアウトの取り消し.....	91
ユーザー データベースのアップグレード	92
進行中の作業の保存	93
スキーマまたはスキーマ バージョンの削除	94
別のスキーマへのデータベースの変更	94

5. スキーマのカスタマイズ	97
スキーマのカスタマイズの概要	97
レコードタイプの操作	98
状態のあるレコードタイプと状態なしレコードタイプ	98
レコードタイプの数	98
新規レコードタイプの追加	99
状態なしレコードタイプの一意のキーの選択	100
スキーマのデフォルトのレコードタイプの選択	101
レコードタイプのデフォルトへの設定	101
レコードタイプファミリーの操作	101
レコードタイプファミリーの追加	102
レコードタイプファミリーへのメンバの追加	102
共通フィールドの作成	103
レコードタイプファミリーからのメンバの削除	103
レコードタイプまたはファミリーの名前の変更	103
レコードタイプまたはファミリーの削除	104
フィールドの操作	104
新規フィールドの追加	106
フィールドのデータ型を選択する上での注意事項	107
フィールドへのヘルプテキストの追加	109
フィールド動作の定義	111
デフォルトのフィールド動作の定義	112
フィールドの修正	112
フィールド名の変更	113
フィールドの削除	113
フィールドを使用したレコードのリンク	114
レコードのリンクによる親/子階層の作成	116
フックの追加によるフィールドのカスタマイズ	118
状態モデルの概要	119
状態遷移マトリックスを使用した状態モデルのカスタマイズ	121
状態遷移マトリックスの表示	121
新規状態の追加	122
状態タイプのマッピング	122
状態の名前の変更	123
状態の削除	124

アクションとアクション タイプの操作	125
アクションとアクション タイプの理解	125
サポートしているアクション タイプ	126
新規アクションの追加	128
状態遷移の作成	128
アクションの修正	129
フックの追加によるアクションのカスタマイズ	129
デフォルト アクションの使用法	130
アクションの削除	131
状態モデルを作成する上での考慮事項	132
ブラックホール	132
状態の数	132
Duplicate	132
事前定義のスキーマ	133
並行開発	133
6. パッケージの適用	135
パッケージと統合の概要	135
パッケージと事前定義されたスキーマ	136
パッケージを適用する際の注意事項	137
使用可能なパッケージ	138
例: Notes パッケージ、Attachments パッケージ、Email パッケージ	139
スキーマに適用したパッケージの確認	140
パッケージの適用方法	141
レコード タイプの有効化	144
状態タイプのマッピング	144
デフォルト アクションの作成	145
パッケージのアップグレード方法	146
7. フォームの操作	147
フォーム操作の概要	147
フォームの作成と変更	148
新しいフォームの作成	148
フォーム名の変更	149
フォームのサイズの変更	149
フォームのフォントの変更	150

フォームの削除	150
複数のプラットフォームで使用するフォームの作成	150
インポートしたデータ用のフォームの作成と変更	151
フォーム コントロールの操作	151
フォームへのコントロールの追加	153
フォームを開く	154
コントロール パレットによるコントロールの追加	154
[フィールド リスト] によるコントロールの追加	155
[フォーム コントロール] メニューによるコントロールの追加	156
フォームでのコントロールの選択	157
コントロールのプロパティの編集	157
フォーム上のコントロールの削除	158
コントロールのサイズと位置の変更	159
コントロールの移動	160
コントロールの配置	160
コントロールのサイズの変更	160
コントロールのタブ順序の変更	161
フォーム上のタブの操作	162
フォームへのタブの追加	162
タブ名の変更	162
タブへのアクセスの制限	163
タブの順序の変更	163
タブの削除	164
タブのコピー	164
レコード フォームの再使用	165
フォームのエクスポート	165
フォームのインポート	165

8. ユーザーの管理 167

ユーザー管理の概要	167
データベース登録の表示	168
ClearQuest ユーザー権限	169
ユーザーの操作	171
新規ユーザーの追加	171
ユーザー アクセス権限の割り当て	173
データベースへのユーザーとグループの登録	174

データベースからのユーザーやグループの登録取り消し	175
ユーザー データベースへのスキーマの変更の適用	176
ユーザーの編集	176
ユーザー権限の変更	177
ClearQuest クライアントでのユーザー プロファイルの編集	178
ユーザーやグループの無効化	178
ユーザー グループの操作	180
新しいユーザー グループの作成	180
サブグループの作成	181
グループへのユーザーの追加	182
グループからのユーザーまたはサブグループの削除	182
データベースへのユーザー グループの登録	182
アクションへのユーザーのアクセス制限	182
ユーザーやユーザー グループのエクスポートとインポート	184
MultiSite 環境でのユーザーの管理	184
マスタースhipが ClearQuest クライアントのユーザーに与える影響	185
マスタースhipがユーザー管理に与える影響	185
ユーザーがマスター登録されている場所の確認	186
ユーザーのマスタースhipの変更	186
マスタースhipの変更	186
マスタースhip変更後のレプリカの同期	186
マスタースhipの変更によるユーザー データベースのアップグレード	186

9. Rational ClearQuest でのセキュリティの使用法 187

ClearQuest でのレコードの非表示化	188
制御するレコード タイプの決定	188
セキュリティ コンテキストに基づいたユーザー グループの作成	188
セキュリティ コンテキストとして使用するレコード タイプの決定	189
セキュリティ コンテキスト フィールドの作成	189
セキュリティ コンテキスト レコードの登録	190
ユーザー アクセスを許可するためのレコードの編集	190
セキュリティ システムの設計	191
セキュリティの例	192
スキーマのチェックアウト	193
セキュリティ コンテキスト フィールドの作成	193
フォームへのセキュリティ コンテキスト フィールドの追加	195

スキーマ変更の適用	195
ユーザー グループの作成	195
セキュリティ コンテキスト レコードの登録	196
各セキュリティ コンテキスト レコードへのグループの関連付け	196
レコードの編集によるグループへの権限の付与	197
その他の ClearQuest セキュリティ 機能の使用法	197
フィールドへのアクセス制限	198
アクションへのアクセス制限	198
ダイアログ タブへのアクセスの制限	198
パスワード保護機能の追加	198

10. ワークフローのカスタマイズにおけるフックの使用法..... 199

フックの理解	200
フックについての重要な注意事項	200
スクリプトの作成	201
スクリプトを使用するための動作状況	202
フィールド フックの操作	203
フィールド フックの追加	204
フィールド フックの編集	205
フィールド フックの削除	205
フィールド間の依存関係の作成	206
ClearQuest Web での依存フィールドの有効化	207
フィールドの選択リストの作成	207
動的選択リストの作成	208
動的リストの編集	210
選択リストの動作の定義	211
フィールドのデフォルト値の指定	212
ユーザーが入力したフィールド値の確認	212
アクション フックの操作	214
アクション フックの追加	216
アクション フックの編集	217
アクション フックの削除	217
フィールド フックとアクション フックの実行順序	218
アクション開始時	218
フィールド値の設定時	219
レコードの確認時	219

レコードのコミット時	219
レコード スクリプトの操作	220
レコード スクリプトの理解	220
ClearQuest Web でのレコード スクリプトの使用法	221
フォーム コントロール イベント	222
レコード タイプへのレコード スクリプトの追加	223
レコード スクリプトの編集	224
レコード スクリプトの削除	224
グローバル スクリプトの操作	225
グローバル スクリプトの理解	225
グローバル スクリプトの作成	226
グローバル スクリプトの編集	226
グローバル スクリプトの削除	227
外部アプリケーションの作成	227
ClearQuest API の使用法	228
セッションの操作	228
クエリーの操作	228
レコードの操作	229
一般的な API 呼び出し	229
フック スクリプト内のテキストの検索	231

11. ClearQuest Web の管理 233

ClearQuest Web に関する考慮事項	233
ClearQuest データ コード ページと ClearQuest Web	234
ClearQuest Web のカスタマイズ	234
ClearQuest Web 設定	235
フィールド スクリプトの実行	236
[電子メールによる通知を有効にする] の設定	237
[表示スキーム] の設定	238
事前定義された表示スキームの操作	239
事前定義された表示スキームの使用法	239
表示スキームの編集	240
ほかの表示スキームの適用	240
表示スキームの削除	240
表示スキームの名前変更	240
新しい表示スキームの作成	240

ClearQuest Web へのアクセス制限	241
ClearQuest Web でのフックの使用法	241
ClearQuest Web に対する依存フィールドの有効化	242
ClearQuest Web でのメッセージの表示	243
フックを使用した Web セッションの検出	243
VBScript での Web セッションの検出	243
Perl での Web セッションの検出	243

12. ClearQuest 電子メールの管理..... 245

ClearQuest 電子メール機能の概要	245
自動電子メール通知の有効化	246
電子メール ルールの作成	246
電子メール ルールのルール コントロールの指定	247
電子メールのアクション コントロールの指定	249
電子メール ルールの表示フィールドの指定	250
電子メール ルールの電子メール アドレスの指定	251
電子メール ルールの同報アドレスの指定	252
電子メール ルールのサンプル	253
既存の電子メール ルールの検索	253
既存の電子メール ルールの修正	254
電子メール アドレスの管理	254
非表示のユーザー	254
ClearQuest クライアントでの電子メール通知の有効化	254
ClearQuest Web での電子メール通知の有効化	255
往復電子メールの使用	256
アクション フックを使用した電子メールの送信	257
Rational E-Mail Reader を使用した電子メール登録の有効化	258
Rational E-Mail Reader のメールボックスの作成	258
Rational E-mail Reader の設定	259
Rational E-Mail Reader の追加オプション設定	263
ClearQuest MailServer Service のプロパティ設定	264
登録のための電子メールの書式設定	265
電子メールの書式設定のガイドライン	266
電子メールの書式例	267
新規レコードを登録する場合の書式の例	267
デフォルトのアクションを使用してレコードを変更する場合の書式例	268

13. データのインポートとエクスポート	269
データのインポートとエクスポートの概要	269
インポート プロセスの開始	269
インポート プロセスのテスト	270
インポート スキーマの作成	270
元のレコード ID の指定	272
インポートしたデータ用のデータベースの作成	272
ClearQuest インポート ファイルの作成	272
レコード タイプ用のインポート ファイルの形式設定	273
インポート ファイル形式に関する重要な考慮事項	273
ClearQuest でサポートされているデータ型	274
履歴用のインポート ファイルの形式設定	275
添付ファイル用のインポート ファイルの形式設定	276
ClearQuest からのデータのエクスポート	277
ClearQuest へのデータのインポート	280
レコードのインポート順序の決定	281
ClearQuest インポート ツールの使用法	281
重複レコードのインポート	284
重複レコード専用のインポート	285
インポート エラーの回復	285
既存のレコードの更新	286
A. ClearQuest のスキーマとパッケージ	287
ClearQuest の事前定義されたスキーマ	288
ClearQuest のパッケージ	289
Defect レコード タイプの状態モデル	297
EnhancementRequest レコード タイプの状態モデル	299
パッケージの状態タイプ モデル	300
Resolution パッケージの状態タイプ モデル	300
UnifiedChangeManagement パッケージの状態タイプ モデル	302
B. ClearQuest 統合の追加	305
ClearQuest 統合の概要	305
自主統合	306
依存先統合	306

ClearQuest 統合とコード ページ	307
例	307
統合用のレコード タイプの有効化	308
スキーマのパッケージの参照	309
統合のテスト	309
自主統合の追加	310
依存先統合の追加	311
Rational Administrator 統合の追加	312
Repository パッケージの追加	312
スキーマの変更の保存	312
Rational Administrator の設定	313
Rational ClearQuest Project Tracker 統合の追加	313
AMBaseActivity パッケージの追加	313
AMWorkActivitySchedule パッケージの追加	314
スキーマの変更の保存	314
Rational ClearQuest Project Tracker の設定	314
Rational TeamTest 統合の追加	314
Repository パッケージの追加	315
TeamTest パッケージの追加	315
スキーマの変更の保存	315
Rational TeamTest の設定	315
Rational UCM 統合の追加	315
AMStateTypes パッケージの追加	317
UCM のデフォルト アクションの設定	318
UCMPolicyScripts パッケージの追加	319
UnifiedChangeManagement パッケージの追加	320
BaseCMAActivity パッケージ (オプション) の追加	320
スキーマの変更の保存	320
Rational UCM の設定	320
Microsoft Visual SourceSafe 統合の追加	321
Visual SourceSafe パッケージの追加	321
スキーマの変更の保存	321
ClearQuest クライアントでのクエリーの作成	322
各クライアント コンピュータの設定	322
パッケージのアップグレードの適用	323
パッケージのアップグレード ステータスの表示	324

C. フックの例	325
フィールド フックの例	326
フィールド選択リスト フックの例	326
依存先リストを作成するフック	327
ユーザー情報を表示するフィールド選択リスト フック	328
フィールド デフォルト値フックの例	330
例 1	330
例 2	331
フィールド権限フックの例	332
フィールド検証フックの例	333
フィールド値変更フックの例	334
アクション フックの例	335
アクション初期化フックの例	335
フィールド値のアクション初期化フック	336
親レコードの値を設定するアクション フック	337
アクション アクセス制御フックの例	340
アクション コミット フックの例	342
アクション通知フックの例	344
アクション検証フックの例	347
レコード スクリプトの例	350
グローバル スクリプトの例	351
ClearQuest フック スクリプトでの CAL メソッドの使用例	352
cqperl を使用した詳細レポートと自動化	355
D. フォーム コントロール	359
ActiveX コントロール	360
添付ファイル コントロール	361
チェック ボックス コントロール	362
コンボ ボックス コントロール	363
ドロップダウン リスト ボックス コントロール	364
ドロップダウン コンボ ボックス コントロール	366
重複元コントロール	367
重複先コントロール	368
グループ ボックス コントロール	369
履歴コントロール	369
リスト ボックス コントロール	370

一覧コントロール	371
オプション ボタン コントロール	372
親子コントロール	373
図コントロール	375
プッシュ ボタン コントロール	375
静的テキスト ボックス コントロール	377
テキスト ボックス コントロール	377

E. コマンド行ユーティリティ..... 381

コマンド行ユーティリティの dbset パラメータ	381
installutil の使用法	381
データベース コピーの作成	383
installutil convertschemarepo	384
installutil convertuserdb	385
データベースのロック解除	386
例	386
Oracle	386
SQL Server	387
cqload によるスキーマのコピー	388
cqload を使用するための準備	388
スキーマ リポジトリへのスキーマのコピー	388
既存のスキーマへの部分的なスキーマのコピー	388
exportschema	389
構文	389
例	389
importschem	390
構文	390
例	390
exportintegration	391
構文	391
例	391
importintegration	392
構文	392
例	393

importutil による動的リストのインポートとエクスポート	393
動的リストのエクスポート	393
構文	394
例	394
動的リストのインポート	394
構文	394
例	395
bkt_tool によるユーザー データベース間のワークスペース項目の移動	395
bkt_tool コマンド行オプション	396
エクスポート	396
構文	396
例	397
インポート	397
構文	397
例	397
更新	397
構文	397
例	397
UNIX での cqtool を使用した夜間レポートの実行	398

F. データベース/トラブルシューティング ガイド 401

UNIX 環境での Oracle との接続に関する問題	401
データベースの接続オプション	401
Unknown host エラー	402
Program unavailable エラー	403
Oracle not available エラー	404
診断処理	405
ホストへの ping	405
ClearQuest データベース設定の確認	406
OpenLink Request Broker が開始しているかどうかの確認	406
Oracle の接続性の確認	407
その他の Oracle 関連の問題	408
Contains 演算子を使用した検索で常に大文字と小文字が区別される	408
dbwin32 による電子メール通知に関する問題のデバッグ	408

索引..... 409

図目次

図 1	変更管理プロセス モデル	3
図 2	スキーマ リポジトリと複数のユーザー データベース	7
図 3	REFERENCE_LIST フィールドを使用する共通データの共有	114
図 4	親/子階層の例	116
図 5	Enhancement レコード タイプの状態モデル	120
図 6	セキュリティ システムのエLEMENT	191
図 7	Defect レコード タイプの状態モデル	298
図 8	Enhancement レコード タイプの状態モデル	299
図 9	Resolution パッケージの状態タイプ モデル	301
図 10	Enhancement レコード タイプの状態タイプ モデル	303

表目次

表 1	変更依頼のライフサイクルにおける役割	2
表 2	Windows インストール用 ClearQuest のアーキテクチャ.....	8
表 3	UNIX インストール用 ClearQuest のアーキテクチャ.....	11
表 4	ClearQuest 管理者のタスク	13
表 5	事前定義のスキーマ	17
表 6	キーボード ショートカット	23
表 7	マウス操作	25
表 8	管理者のコード ページ関連タスク	28
表 9	ClearQuest のデータ コード ページ値を選択するためのガイドライン	30
表 10	Rational によってサポートされている Windows のコード ページ.....	31
表 11	データベース製造元の文字セットと Windows コード ページ.....	34
表 12	ClearQuest データベース アクティビティ	46
表 13	Oracle の文字セットと Windows のコード ページ.....	51
表 14	ClearQuest パッケージの概要	138
表 15	製造元の種類別のサンプル データ	382

まえがき

このたびは、Rational ClearQuest をお買い上げいただき、まことにありがとうございます。
ClearQuest は、カスタマイズ可能な障害/変更追跡システムです。Rational ClearQuest を使用すると、変更依頼の登録、割り当て、追跡、テスト、リリースに関わるプロセスを構成し、自動化できるため、開発チームにとって便利なシステムです。追跡の対象になる変更依頼には、障害の修正、既存の機能の拡張、新しい機能の追加、マニュアルやパッケージの変更などがあります。

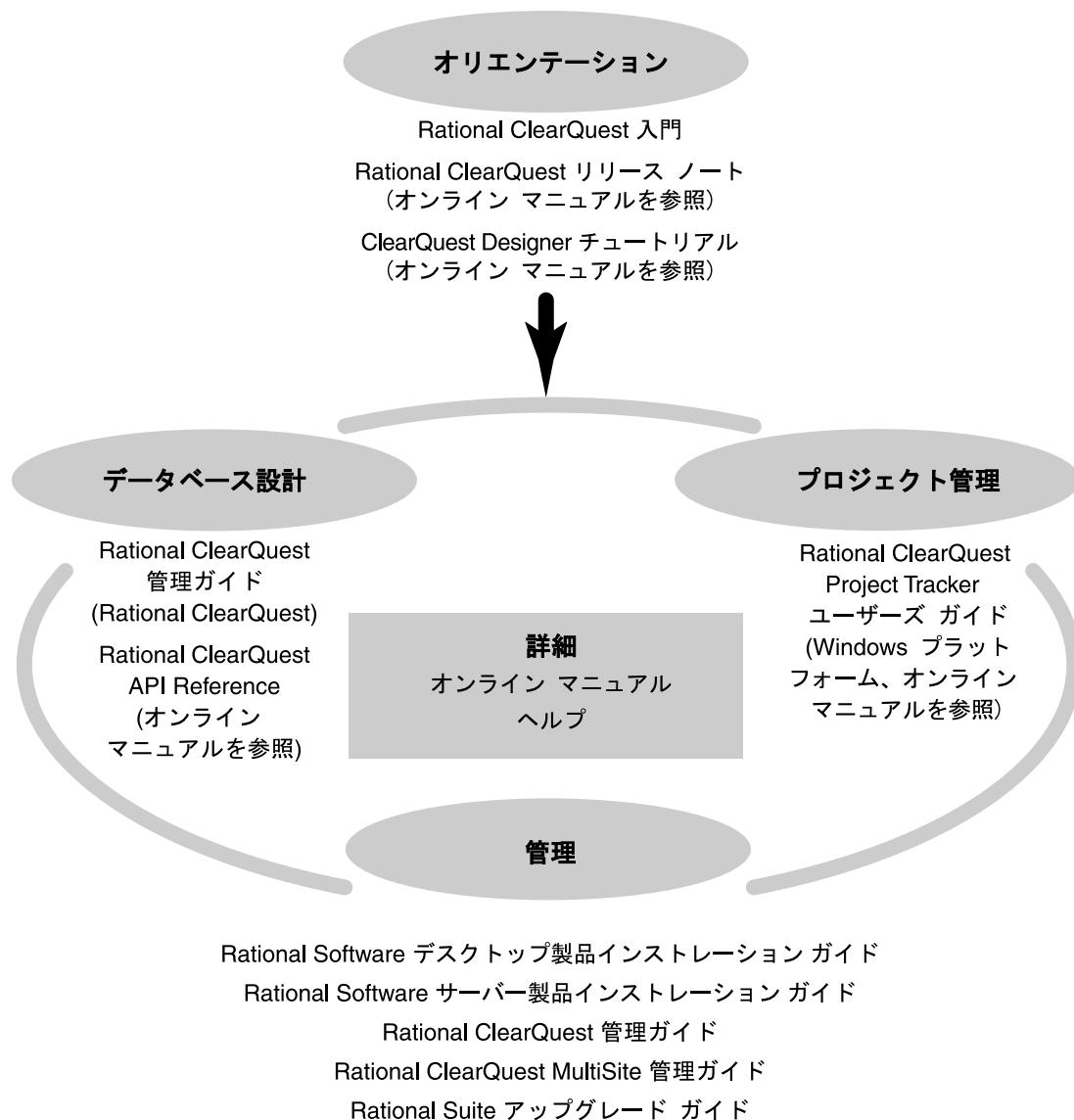
本書について

本書は ClearQuest 管理者を対象としています。ClearQuest 管理者とは、ClearQuest 環境の設定、変更依頼を管理するためのプロセス モデルの作成、ユーザーの追加と管理、ClearQuest 環境の基本的で継続的な管理に対して、責任を持つユーザーのことです。

本書では、変更管理の概念、ClearQuest 管理者の役割、ClearQuest データベースの管理、ClearQuest スキーマの作成と変更、セキュリティにおけるユーザーの管理、ClearQuest ツールとユーティリティの利用方法、ClearQuest とほかの Rational ソフトウェア アプリケーションとの統合の導入について説明します。

本書では、ClearQuest クライアントを使用した経験があり、リレーショナル データベースの管理に精通し、VBScript または Perl でスクリプトを記述できることを前提としています。

ClearQuest マニュアル ロードマップ



ClearQuest 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
ClearQuest - ベース ClearCase®	変更依頼を ClearCase エLEMENT 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』
ClearQuest - PurifyPlus	開発者が、PurifyPlus から ClearQuest を呼び出せるように します。	PurifyPlus のヘルプ 『Rational ClearQuest 管理ガイド』
ClearQuest - RequisitePro	開発者が、RequisitePro から ClearQuest を呼び出して、要求を ClearQuest 変更依頼に関連付ける ことができますようにします。	『Rational Suite® 管理ガイド』 『Rational ClearQuest 管理ガイド』
ClearQuest - SoDA	ClearQuest から情報を収集して、 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
ClearQuest - Test Manager	開発者が、TestManager から ClearQuest を呼び出せるように します。	『Using Rational TestManager』 『Rational ClearQuest 管理ガイド』
ClearQuest - Robot	開発者が、Robot から ClearQuest を 呼び出せるようにします。	『Rational Robot User's Guide』 Rational Robot のヘルプ
ClearQuest - UCM	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』

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:heckout`

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)

ClearQuest 管理について

1

Rational ClearQuest は、カスタマイズ可能な障害/変更追跡システムです。この製品を使用すると、開発チームで作業を行う場合に、より高品質の製品を無駄な労力を抑えて迅速に開発するための変更管理システムを実装できます。

ClearQuest を導入する前に、変更管理の概念と ClearQuest 管理者の役割を理解しておく役に立ちます。この章の内容は、次のとおりです。

- 1 ページの「変更管理の概念的な理解」
- 5 ページの「ClearQuest スキーマ、スキーマ リポジトリ、データベース、接続の定義」
- 8 ページの「ClearQuest のアーキテクチャとコンポーネントの理解」
- 21 ページの「ClearQuest 入門」

変更管理の概念を理解しているユーザーは、「変更管理の概念的な理解」の項を読み飛ばしてかまいません。

変更管理の概念的な理解

変更管理が必要な理由

変更管理システムを使用すると、変更依頼の登録、アサイン、追跡、テスト、リリースに関わるプロセスを構成し、自動化できます。追跡の対象になる変更には、バグや障害の修正、既存の機能の拡張、新しい機能の追加、マニュアルやパッケージの変更などがあります。

変更管理システムを使用すると、開発チームにとって次のような利点があります。

- 作業の重複を防ぐことができます。
- 個々の変更依頼の進捗状況を追跡できます。
- 変更依頼に優先順位を付け、最も重要なものから対応できます。
- 開発チーム メンバー間での作業負荷のバランスをとることができます。
- チームワークを向上させ、チーム メンバー間の違いを吸収できます。
- プロジェクトの進捗状況とステータスを全体的に監視できます。
- 製品開発プロセスの理解を深め、プロセスを改良することができます。

変更依頼のライフサイクル

変更管理システムを運用する上で、変更依頼のライフサイクルという概念を理解していることは重要です。変更依頼は、開発チームのメンバーまたはそれ以外のメンバー（顧客、プロダクトマネージャー、カスタマ サポート担当など）によって登録されます。登録された変更依頼は、開発チームのメンバーにより検討され、その変更の実装作業を果たすことのできる担当者にアサインされます。作業の結果は検討、テストできる別の担当者に渡され、その後プロジェクトに統合されます。変更は完了をもって、つまりテストと実装の終了をもってクローズされたと見なされます。

表 1 に、一般的な製品開発部署でのライフサイクルの運用例を示します。

表 1 変更依頼のライフサイクルにおける役割

チーム メンバー	変更依頼のライフサイクルにおける役割
ユーザー	変更依頼の登録
変更管理マネージャー	変更依頼の確認と受け入れ
プロジェクト マネージャー	作業の割り当てとスケジュール作成
エンジニア	変更依頼を解決するための作業の実施
テスト担当者	テスト ビルドでの解決方法の確認

当然ながら、変更依頼のライフサイクルは、表 1 に示したものよりも複雑になる場合があります。より多くの（または少ない）関係者が存在する場合や、変更依頼の拒否、改訂、リアサイン、延期、重複、その他の変更、経路変更などが行われる場合があります。

変更管理プロセスモデル

変更管理プロセス モデル（状態遷移モデルとも呼ばれます）は、1 つの変更依頼のライフサイクル内で発生する可能性があるステップを体系的に表現したものです。このモデルは、変更依頼が通過する状態、状態が変化するときに行われるアクション、アクションをいつどのように実行できるかを定義するルールを使用して、ライフサイクルを定義します。

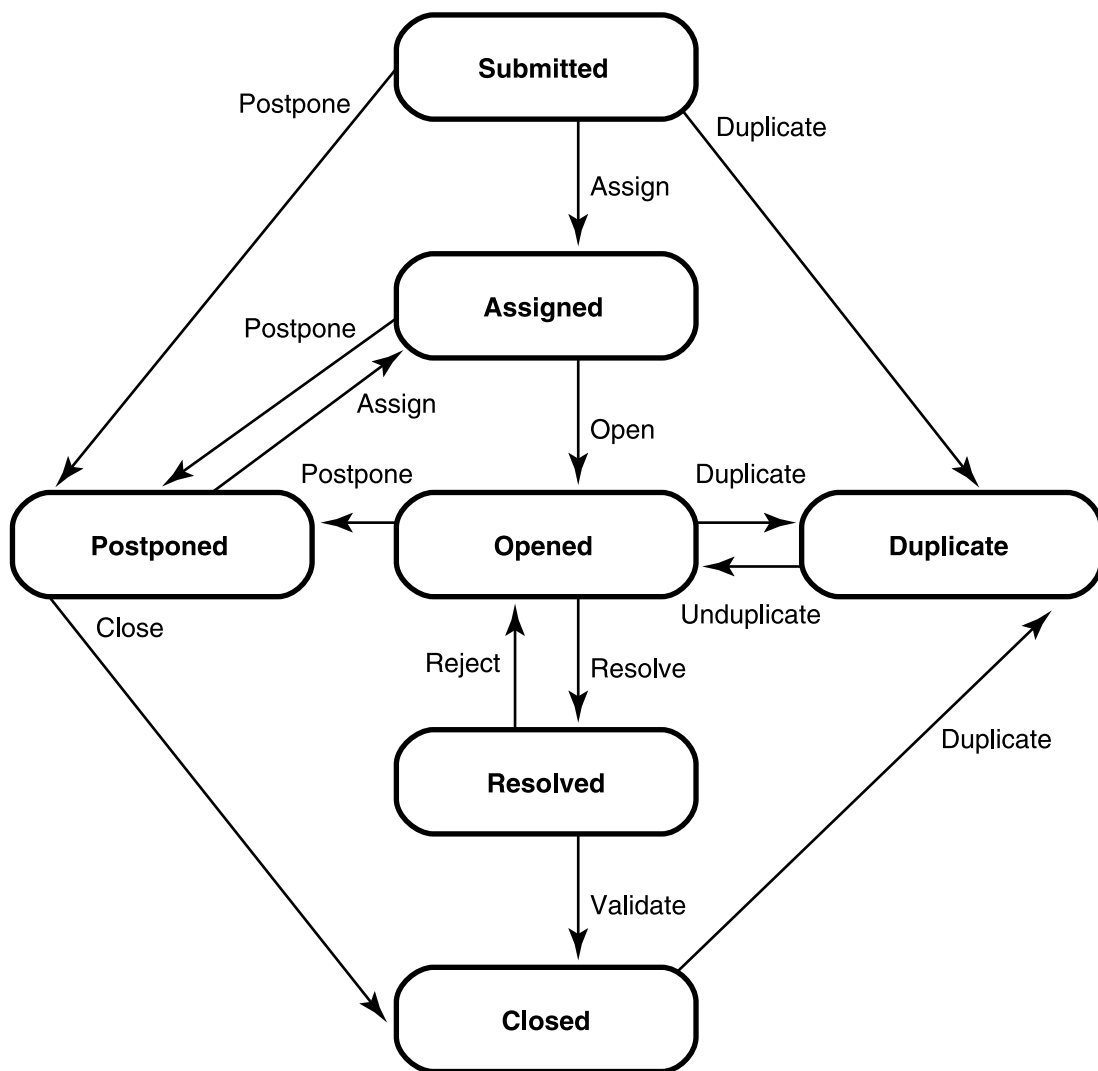
変更依頼の状態は、現在のステータスです。一般的な状態には、Submitted、Assigned、Opened、Postponed、Duplicated、Resolved、Closed があります。

アクションは、変更依頼を 1 つの状態から別の状態に移す（状態遷移）アクティビティです。一般的なアクションには、Assign、Reject、Open、Postpone、Duplicate、Validate、Resolve、Close があります。

ルールは、アクションをいつどのように実行できるかを定義します。たとえば、**Validate** アクションを使用して変更依頼を **Resolved** 状態から **Closed** 状態に変更できるユーザーを、品質管理担当のエンジニアに限定します。

図 1 に示すように、プロセス モデルは、図で示すことができます。この図では、状態は角の丸い矩形で表され、アクションは矢印で表されています。

図 1 変更管理プロセス モデル



プロセス モデルは、変更依頼のタイプによって異なる場合があります (たとえば、障害修正プロセスの場合と、新規機能追加プロセスの場合では異なります)。このようなプロセス モデルはカスタマイズして、担当する開発チームやプロジェクトに合ったニーズや慣行に対応する必要があります。

ClearQuest: 変更管理に対する障害/変更追跡システム

ClearQuest は、一貫した管理プロセス モデルの実装と管理を行うことができる、自動化された障害/変更追跡システムです。

さまざまなタイプの変更依頼に対し、次のことを行うことができます。

- 状態とアクションを使用してプロセス モデルを定義できます。
- アクションをいつどのように実行できるかを制御するカスタマイズしたルールをプログラミングします。
- システムの関係者のために、変更依頼へのアクセス、メンバー相互の連絡、適切なアクションの実行ができるフォームを作成できます。
- レポートとクエリーを作成し、マネージャーが、それらを使用して、個々の変更依頼、変更依頼のクラス、プロジェクトに関連するすべての変更依頼を参照できるようにします。

ClearQuest では、組織で確立されている方針や手順に従って個々の変更依頼が変更依頼ライフサイクルを通過することを保証しつつ、開発チームのメンバーによる作業を簡略化しています。

また、ClearQuest では作業の割り当て、作業負荷の管理、プロジェクトの進捗状況の追跡、チームワークの向上、プロセスや製品の継続的な改良に役立つ各種管理ツールを用意しています。

さらに、ClearQuest は Rational Suite の統合部分として実装できます。変更依頼に関する情報をほかの Rational 製品に転送して、開発やテストに使用することができます。各種ツールから返される情報を ClearQuest で参照することにより、変更依頼のステータスの追跡を最も完全に近い形で実現できます。

ClearQuest スキーマ、スキーマ リポジトリ、データベース、 接続の定義

スキーマ

ClearQuest のスキーマは、1 つのタイプの変更依頼についてプロセス モデルを完全に記述したもの (そのプロセス モデルのメタデータ) です。この中には、モデルの状態とアクションの記述、個々の変更依頼について保存できるデータの構造、ビジネス ルールの実装に使用できるフック コードまたはスクリプト、変更依頼に関する情報の表示や入力に使用できるフォームやレポートなどが含まれます。

このスキーマは、ClearQuest ユーザー データベースの原型または青写真です。実際のユーザー データを保持するデータベースを作成すると、そのデータベースはスキーマに定義されている青写真に従います。ただし、スキーマ自体はデータベースではありません。実際の変更依頼に関するユーザー データは一切保持せず、ユーザーがユーザー データベースのデータに追加または変更を実行してもスキーマは変わりません。

管理者には、Rational ClearQuest Designer を使用して、スキーマを作成または修正する権限があります。一般的には、スキーマを作成または修正する権限は開発チームのほかのメンバーには与えられません。ただし、クエリーやレポートの修正など、特定の変更を実行する権限を与えることができます。

スキーマ リポジトリ

ClearQuest では、スキーマ リポジトリという特別な種類のデータベースにスキーマを格納します。スキーマ リポジトリは、マスター リポジトリまたはマスター データベースとも呼ばれます。

スキーマ リポジトリには、複数のスキーマを格納できます。たとえば、障害に関する変更依頼用の 1 つのスキーマと、機能拡張に関する変更依頼用の別のスキーマを格納できます。

スキーマ リポジトリには、同じスキーマの複数のバージョンを格納することもできます。たとえば、アクションを変更したり新規レポートを追加することでスキーマの変更を行うたびに、新しいバージョンが作成されます。

データベース

ClearQuest でのデータベースは、1 つのプロセス モデルで使用する変更依頼データの集まりです。

各データベースは特定のスキーマの特定のバージョンに関連付けられます (図 2 を参照)。スキーマは、データベース内のデータの格納方法と変更方法を定義します (つまり、データベースはスキーマのインスタンスです)。ユーザーが変更依頼に関する情報を追加、変更すると、データベースのデータが変更されます。

各変更依頼のレコードはデータベースに含まれます。変更依頼がライフサイクルを通過すると、このレコードに格納されているデータがそれに従って変更されます。

複数のデータベースを作成して 1 つのスキーマに関連付けることができます。たとえば、3 つのプロジェクトで同じ障害修正プロセス モデルを使用する場合は、プロジェクトごとに 1 つのデータベースを作成し、3 つのすべてのプロジェクトを同じスキーマに関連付けることができます。

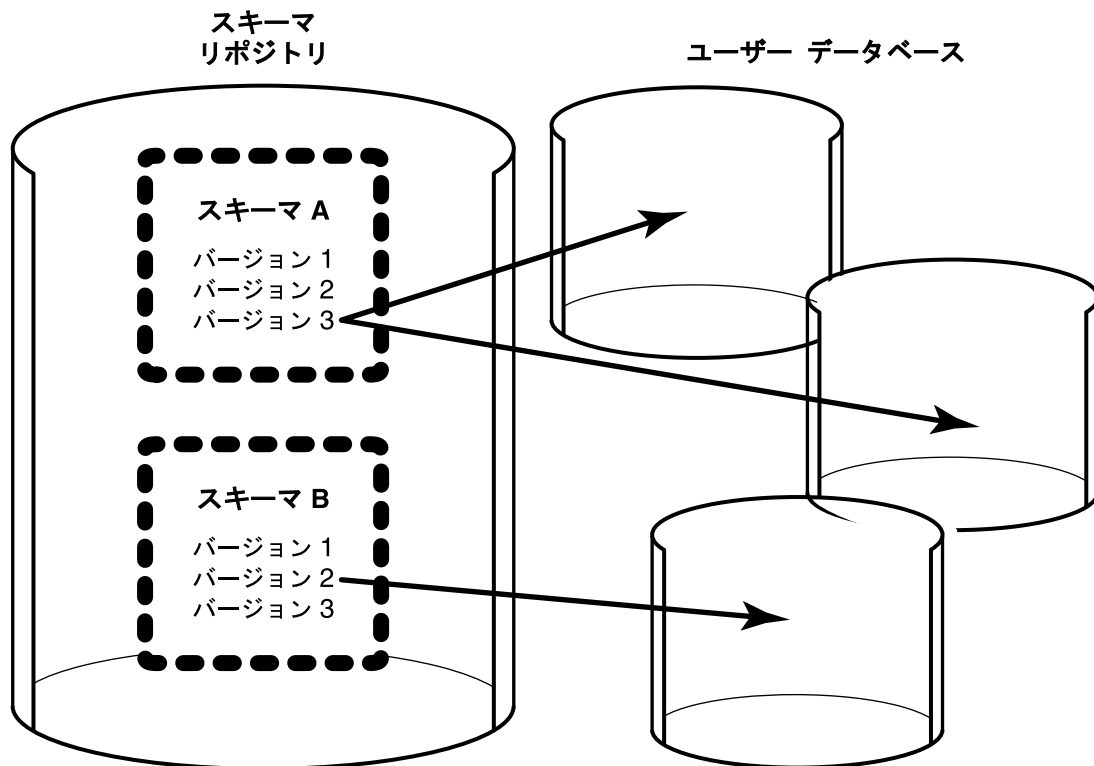
ClearQuest では、3 つのタイプのデータベースが用意されています。

- ユーザー データベースは、実動データベースとも呼ばれ、開発チームのメンバーが入力した情報を管理します。このデータベースには、実際の変更依頼の処理中に作成される実動データが含まれています。
- サンプル データベースは、ユーザー情報のサンプル障害レコード 40 個、サンプル クエリー、レポートを提供する作業用ユーザー データベースです。サンプル ユーザー データベースは、ClearQuest の動作の理解やユーザーのトレーニングに役立てることができます。サンプル ユーザー データベースは、ClearQuest に付属している事前定義されたスキーマのうち、Blank スキーマと Common スキーマを除くすべての事前定義されたスキーマで使用できます。サンプル ユーザー データベースは、スキーマ リポジトリを作成するときに ClearQuest メンテナンス ツールを使用して作成できます。事前定義されたスキーマの詳細については、288 ページの「ClearQuest の事前定義されたスキーマ」を参照してください。
- テスト データベースは、Rational ClearQuest Designer を使用して新規スキーマや既存のスキーマの新しいバージョンを作成するときに作成できる一時的なデータベースです。テスト データベースを作成したら、Rational ClearQuest クライアントを使用してテスト データベースでデータの追加や表示を実行して、予期したとおりに新規スキーマが実行されることを確認します。テスト データベースを使用することにより、ユーザー データベースにある実動データに損傷を与える危険を冒すことなくテストできます。

接続

接続とは、データベース セットのことです。図 2 に示すように、接続は 1 つのスキーマ リポジトリと関連付けられたすべてのユーザー データベースで構成されます。

図 2 スキーマ リポジトリと複数のユーザー データベース



通常、1 つの接続には 1 つのプロジェクトのすべてのスキーマとデータベースが含まれます。ただし、1 つの接続で複数のプロジェクトをサポートしたり、1 つのプロジェクトに複数の接続を設定することもできます。

製造元データベース管理システム

ClearQuest で使用するスキーマ、スキーマ リポジトリ、データベース、接続は、1 つまたは複数のデータベース管理システムで保持と管理を実行します。

ClearQuest には Sybase SQL Anywhere が同梱されており、一般的に、小規模または中規模のインストールで使用します。このほか、ClearQuest では Microsoft Access をサポートしており、一般的に、小規模のインストールで使用します。また、Microsoft SQL Server、Oracle DBMS、IBM DB2 もサポートしており、これらは大規模または分散環境での使用に適しています。

ClearQuest のアーキテクチャとコンポーネントの理解

Windows インストール用 ClearQuest のアーキテクチャとコンポーネント

ClearQuest のアーキテクチャとコンポーネントは、Microsoft Windows インストールと UNIX インストールで一部異なります。ここでは、まず Windows インストールについて説明し、次に UNIX インストールでの違いについて説明します。Windows インストールの一部の機能は両方のインストールで必要になるので、UNIX インストールを使用する場合は両方の項の説明をお読みください。

Windows 用 ClearQuest のアーキテクチャ レイアウトを表 2 に示します。

表 2 Windows インストール用 ClearQuest のアーキテクチャ

層	コンポーネント
層 1	<ul style="list-style-type: none">▪ ClearQuest クライアント▪ ClearQuest Web▪ ClearQuest Designer▪ ClearQuest メンテナンス ツール▪ ClearQuest インポート ツール▪ ClearQuest エクスポート ツール▪ ClearQuest ユーザー管理▪ ClearQuest API▪ ClearQuest MultiSite▪ コマンド ユーティリティ▪ Rational E-Mail Reader
層 2	<ul style="list-style-type: none">▪ ClearQuest Core
層 3	<ul style="list-style-type: none">▪ Microsoft Access▪ SQL Anywhere▪ Microsoft SQL Server▪ Oracle RDBMS▪ IBM DB2▪ ODBC ドライバ

表 2 の下の層から説明します。

- 層 3 はデータベース層で、この層にはスキーマとユーザー データベースに含まれる情報を物理的に管理するデータベース管理システム (1 つまたは複数) があります。Windows 環境では、データベース管理システムとして Microsoft Access (一般的に 5 ユーザー以下の、小規模インストールに使用)、SQL Anywhere (小規模または中規模のインストールに使用)、Microsoft SQL Server、Oracle RDBMS、IBM DB2 (大規模インストールに使用) を使用できます。
- 層 2 は ClearQuest Core で、アーキテクチャのアプリケーション ロジックやビジネス ルールの保存と実行が発生する層です。

ClearQuest Core では、データベース層 (層 3) との通信に OpenLink ドライバなどの Open Database Connectivity (以下 ODBC) ドライバを使用します。詳細については、『Rational Software サーバー製品インストールガイド』を参照してください。

ClearQuest Core には、ClearQuest API (アプリケーション プログラミング インターフェイス) を介した基本機能の容量に制限があります。これらの制限は通常、フックのスク립トに対して使用されます。

- ClearQuest アーキテクチャの層 1 には、データの入力や表示、スキーマやデータベースの作成と管理、ほかの環境からのデータのインポートとエクスポート、複数の保存場所を対象としたデータの複製などを実行するための、さまざまなモジュールやツールが含まれています。また、層 1 のアプリケーションとツールを使用した通信に対する ClearQuest API (アプリケーション プログラミング インターフェイス) も提供します。

ClearQuest がシステムに既にインストールされている場合は、[スタート] メニューから [プログラム]、[Rational Software] の順にポイントし、[Rational ClearQuest] をクリックすると、層 1 を構成するエレメントを見ることができます。

層 1 の一部である ClearQuest クライアントは単に Rational ClearQuest と呼ばれ、管理者や開発チームのすべてのメンバーが使用します。これは Windows ではネイティブ モジュールで、クライアント/サーバー モードで動作し、ユーザー、マネージャー、管理者による変更依頼の入力と変更、クエリーとレポートの実行と表示、メンバー相互の連絡に使用します。

ClearQuest Designer は、クライアント/サーバー管理アプリケーションで、スキーマの作成と変更、ユーザー データベースの作成とアップグレードに使用するツールです。ClearQuest Designer を使用すると、スキーマのデータ レイアウトの定義と、変更依頼の各タイプ用プロセスモデルの構築を実行できます。

層 1 には、管理機能の実行に使用するツールや、開発チームのユーザーに対して追加の機能を提供するように設定できるツールも含まれます。用意されているツールは次のとおりです。

- ClearQuest メンテナンス ツール。管理者が使用する管理ツールで、スキーマ リポジトリと接続を作成します。また、サンプル データベースを作成したり、スキーマ リポジトリとデータベースを ClearQuest の新しいリリース レベルにアップグレードします。

- **ClearQuest エクスポート ツール。**管理者と ClearQuest ユーザーが使用するユーティリティで、ClearQuest データベースのデータを外部ファイルにエクスポートします。
- **ClearQuest インポート ツール。**管理者と ClearQuest ユーザーが使用するユーティリティで、外部ファイルのデータを ClearQuest データベースにインポートします。
- **ClearQuest Web。**Web サーバーに設定する ClearQuest クライアントで、インターネットブラウザを使用するユーザーは ClearQuest クライアントで利用できる機能と同じもの多くを実行できます。ClearQuest Web では、Active Server Pages (ASP) を使用します。ClearQuest Web は Microsoft Internet Information Server (IIS) を実行する Windows ベースの Web サーバー上で実行する必要があります。また、ユーザーのコンピュータまたは UNIX ワークステーションには、Java Virtual Machine (JVM) が存在する必要があります。
- **ClearQuest 電子メール。**特定のイベントの発生を通知する機能です。たとえば、障害が修正されるたびにテスト チームに警告を電子メールによる通知で送信するルールを作成できます。ClearQuest 電子メールには Rational E-Mail Reader というアプリケーションがあり、ClearQuest クライアントを使用せず、電子メールによる通知で変更依頼の登録や変更ができます。
- **ClearQuest ユーザー管理。**ユーザーとユーザー グループの設定と管理に使用するツールです。このツールを使用すると、ユーザーとユーザー グループの作成、ユーザー ID とパスワードの設定、ユーザーのユーザー グループへの割り当て、ユーザーとユーザー グループのスキーマ間でのインポートとエクスポート、ユーザー グループのデータベースへの登録を実行できます。ユーザー管理ツールは、ClearQuest Designer でも使用できます。

カスタマイズ: フックと ClearQuest API

ユーザーや管理者と ClearQuest Core との通信のほとんどは、層 1 のモジュールで利用できる標準機能で処理できます。ただし、必要に応じてビジネス ルールをカスタマイズし、高度なデータ処理を実行することもできます。ClearQuest にはフック機能があり、アプリケーションプログラミング インターフェイス (API) 経由で ClearQuest Core にアクセスすることで、このような処理を実現することができます。

フックとは、指定の条件下で実行されるスクリプト (トリガ) のことで、特定のタスクを実行します。たとえば、フックは次の用途に使用できます。

- フィールドで指定されるデータを制御します。これには、デフォルト値のフィールドへの指定、有効な値を備えた選択リストの作成、エントリが適切なデータ型であることの確認、エントリが指定範囲内にあることの確認などの方法があります。
- アクセス管理を定義して、誰がどのようなタイミングでアクションを実行できるかを制限します。
- 特定のアクションが実行されると、電子メールによる通知の送信など、別のイベントが実行されます。

ClearQuest の Windows 環境では VBScript または Perl で記述されたフック、UNIX 環境では Perl で記述されたフックのみをそれぞれサポートしています。Windows で作業する場合、管理者は最初から VBScript または Perl のどちらかを排他的に使用するかを選択する必要があります。ClearQuest では、スクリプト言語が混在したフックは実行できません。

フックは、ユーザーがレコードに対してアクションを実行するときに ClearQuest の指示の下で実行されます。フックは ClearQuest API を介して ClearQuest Core にアクセスします。ClearQuest API はオブジェクト、メソッド、関数のセットが用意されている強力な API で、フック コードから呼び出すことができます。たとえば、フックを使用すると、ユーザー ID、時刻、別のデータベース レコードのステータスの呼び出しなどが必要な、ほかの要因に基づくフィールドで使用できる値のリストを設定できます。

外部アプリケーションも ClearQuest API を呼び出すことができます。ただし、OLE オートメーションまたは embedded Perl がサポートされていることが必要です。たとえば、外部アプリケーションを使用すると、クエリーの作成と実行を行い、グラフとレポートを実行し、クライアントワークスペースの内容を調べることができます。

フック、外部アプリケーション、ClearQuest API の詳細については、Rational ClearQuest を参照してください。また、www.rational.com で Rational Developer Network を参照してください。ただし、英語のみのご利用となります。

UNIX インストール用 ClearQuest のアーキテクチャ

表 3 に示す UNIX インストール用 ClearQuest のアーキテクチャは、Windows インストールのアーキテクチャと 3 層構造は同じですが、コンポーネントの数と機能が異なります。

また Windows では、ClearQuest メンテナンス ツールと ClearQuest Designer を使用してスキーマとデータベースを設定、カスタマイズ、管理するには、ClearQuest をインストールする必要があります。

UNIX 版 ClearQuest のアーキテクチャ レイアウトを表 3 に示します。

表 3 UNIX インストール用 ClearQuest のアーキテクチャ

層	コンポーネント
層 1	<ul style="list-style-type: none">▪ ClearQuest クライアント▪ ClearQuest Web▪ ClearQuest API▪ ClearQuest MultiSite▪ コマンド ユーティリティ▪ Windows インストールにおける ClearQuest 管理アプリケーション
層 2	<ul style="list-style-type: none">▪ ClearQuest Core

表 3 UNIX インストール用 ClearQuest のアーキテクチャ

層	コンポーネント
層 3	<ul style="list-style-type: none"> ▪ Microsoft Access ▪ SQL Anywhere ▪ Microsoft SQL Server ▪ Oracle RDBMS ▪ IBM DB2 ▪ ODBC ドライバ

表 3 の下の層から説明します。

- 層 3 はデータベース層で、この層にはスキーマとユーザー データベースに含まれる情報を物理的に管理するデータベース管理システム (1 つまたは複数) があります。UNIX 環境では、データベース管理システムとして UNIX サーバー上の Oracle RDBMS、Windows サーバー上の Microsoft SQL Server または IBM DB2 を使用できます。
- 層 2 は ClearQuest Core で、アーキテクチャのアプリケーション ロジックやビジネス ルールの保存と実行が発生する層です。

ClearQuest Core では、データベース層 (層 3) との通信に OpenLink ドライバ、OpenLink Request Broker などの Open Database Connectivity (以下 ODBC) ドライバを使用します。詳細については、『Rational ClearCase 製品ファミリー インストレーション ガイド』を参照してください。

- ClearQuest アーキテクチャの層 1 には、ユーザー、管理者による変更依頼の入力と変更、レポートの作成、相互の連絡、複数の保存場所を対象としたデータの複製に使用するツールが含まれています。また、層 1 アプリケーションとツールを使用した通信に対する ClearQuest API も提供します。

層 1 の一部である ClearQuest クライアントは単に Rational ClearQuest と呼ばれ、管理者や開発チームのすべてのメンバーが使用します。これは UNIX システムではネイティブ モジュールで、ユーザー、マネージャー、管理者による変更依頼の入力と変更、クエリーとレポートの実行と表示、メンバー相互の連絡に使用します。

UNIX ワークステーションのブラウザは ClearQuest Web で使用できます。ただし、このブラウザを使用するには、ワークステーションに Java Virtual Machine (JVM) が存在し、ClearQuest Web と Microsoft IIS Web サーバーが使用する環境の Windows サーバーで設定されていることが必要です。詳細については、『Rational ClearQuest インストレーション ガイド』と『Rational Software サーバー製品インストール ガイド』を参照してください。

また、UNIX インストールの層 1 には、UNIX サーバー上の配布ポイントを作成する機能があります。ユーザーはここから ClearQuest Client UNIX 版をダウンロードし、インストールすることができます。

ClearQuest UNIX インストールには、ClearQuest Designer、ClearQuest メンテナンス ツール、ClearQuest インポート ツール、ClearQuest エクスポート ツール、ClearQuest ユーザー管理 ツール、ClearQuest Web、Rational E-mail Reader は含まれていません。

Windows では、UNIX 環境でスキーマとデータベースを設定、カスタマイズ、管理するためには、ClearQuest Designer、ClearQuest メンテナンス ツール、ClearQuest ユーザー管理ツール、Rational E-mail Reader を含む ClearQuest インストールが必要です。

同様に、Windows で外部ファイルからデータをインポートしたり、外部ファイルにデータをエクスポートするには、ClearQuest インポート ツールと ClearQuest エクスポート ツールを含む ClearQuest インストールが必要です。

ClearQuest 管理者の役割の理解

ClearQuest 管理者が担当する作業には、ClearQuest のインストール、リリース領域の作成とサービス リリースのインストール、スキーマの定義と管理 (つまり、プロセス モデルの設計と変更)、ユーザー データベースの作成と管理、Web と電子メールアクセスの有効化、共用のクエリー、チャート、レポートの作成と維持、ユーザー アカウントの管理などがあります。

これらの機能については、この項で説明します。また、それぞれの役割に関する詳細情報を説明するこのマニュアルの該当項やほかの ClearQuest マニュアルへの参照先も示します。

表 4 に、管理者が行うタスクのリストを示します。

表 4 ClearQuest 管理者のタスク

管理者のタスク	アプリケーション
変更管理プロセスの設計	
RDBMS のインストールと設定	製造元データベース
ClearQuest のインストール	Rational ClearQuest セットアップ ウィザード
スキーマ リポジトリと接続の作成	ClearQuest メンテナンス ツール
ユーザー データベースの作成	ClearQuest Designer
プロセスのワークフローの文書化	
ワークフローに既に適合している事前定義されたスキーマの選択 事前定義されたスキーマをワークフローに適合するように変更 ワークフローに適合する新規スキーマの作成	ClearQuest Designer

表 4 ClearQuest 管理者のタスク

管理者のタスク	アプリケーション
適切なコード ページ設定の決定と、データの一貫性の確認のための管理タスクの実行	製造元データベースのツール、ClearQuest installutil コマンド ユーティリティ
ユーザー データベースの新規スキーマへのアップグレード	ClearQuest Designer
Web アクセスと電子メール通知の有効化 (オプション)	ClearQuest Designer、Rational E-Mail Reader
共用のクエリー、グラフ、レポートの作成	ClearQuest、Rational SoDA か Seagate Crystal Reports またはその両方
セキュリティ制御の設定	ClearQuest Designer
管理機能の委託	
ユーザー アカウントの管理	ClearQuest ユーザー管理
スキーマのカスタマイズや管理などの、継続的な保守	ClearQuest Designer
ClearQuest MultiSite のインストールと設定	Rational ClearQuest セットアップ ウィザード、ClearQuest MultiSite 管理ツール、Rational Shipping Server

これらのタスクの詳細については、以降の項を参照してください。

システムの設計と実装

変更依頼管理システムの設計と実装に関するすべての手順については、このマニュアルには記載しません。ただし、包括的な実装計画を作成する時間は設ける必要があります。既存のプロセスを分析し、今後のあるべき姿に対するコンセンサスを形成するには、(これらのプロセスが文書化されていることはまれであるため) ある程度の労力が必要です。

変更依頼管理プロセスを設計し、実装するには、次のことを実行する必要があります。

- プロセスの適用範囲の明確化
- さまざまなユーザー タイプの役割の規定
- 変更管理プロセス モデルの構築
- ユーザーと上司からのモデル承認の取得
- ユーザーのトレーニングとシステムの導入
- 変更管理プロセスの運用

これらの作業の多くにチームも共同責任を負いますが、ClearQuest 管理者はこれらに見落としがないことを確認する必要があります。

データベース管理システムの設定

ClearQuest ソフトウェアをインストールする前に、ClearQuest のスキーマ、スキーマ リポジトリ、データベースの保存に使用するサードパーティ製データベース管理システムをインストールし、設定する必要があります。

サードパーティ製データベースの設定に関する推奨事項については、Windows の場合は『Rational Software サーバー製品インストールガイド』を参照してください。

ClearQuest のインストール

ライセンス サーバーを設定し、ClearQuest 用のライセンス キーを取得します。次に、ClearQuest ソフトウェアを、サポートするユーザーの要件に合わせてクライアントとサーバーのシステムにインストールします。

Windows インストールについては、Windows 版の『Rational Software デスクトップ製品インストールガイド』と『Rational Software サーバー製品インストールガイド』を参照してください。後者のマニュアルでは、Rational ClearQuest のプラットフォームとソフトウェアの要件についても説明しています。

ClearQuest ユーザー グループへの加入

また、ClearQuest をインストールした後で、ClearQuest ユーザー グループに参加できます。ClearQuest ユーザー グループは、質問の送信、ほかの ClearQuest ユーザーとの情報の共有に役立つ電子メール フォーラムです。グループに加入するには、Rational Web サイト <http://www.rational.com/support/usergroups/> にアクセスしてください。ただし、英語のみのご利用となります。

お客様の電子メール アドレスがほかのユーザーに開示されることはありません。

スキーマ リポジトリと接続の作成と管理

ClearQuest メンテナンス ツールを使用して、スキーマ リポジトリ (マスター リポジトリまたはマスター データベースと呼ばれることもある) の作成と管理を行います。

新規のスキーマ リポジトリを作成する場合は、新規の接続 (データベース セット) も作成して、そのスキーマ リポジトリに含まれるスキーマに関連付けられているすべてのデータベースを追跡管理します。接続の管理にも ClearQuest メンテナンス ツールを使用します。

詳細については、45 ページの「データベースの管理」を参照してください。

コード ページ タスクの実行

ClearQuest を使用する場合、ClearQuest データベースにアクセスするクライアントごとに、クライアント独自のオペレーティング システム コード ページがあります。コード ページでは、使用している環境で有効な文字セットを指定します。つまり、特定のクライアントに対する適切な文字操作を、コード ページで制御します。

複数言語を使用する環境では、ClearQuest クライアントは、異なるオペレーティング システムのコード ページを使用する、異なる言語ベースのオペレーティング システムで実行されることがあります。このような環境では、すべてのオペレーティング システム コード ページに共通する文字のみを ClearQuest で入力する必要があります。ClearQuest には、ClearQuest 管理者が、複数言語環境を保護し、ユーザーが入力した文字がクライアントで適切に表示されない問題を防止するための特定の機能があります。

この機能には、スキーマ リポジトリとユーザー データベースに対する ClearQuest データ コード ページ値と、以前のバージョンの ClearQuest を実行しているクライアントでのインストールに対する CharacterSetValidation パッケージがあります。ClearQuest 管理者は、これらの機能を利用するために ClearQuest を導入するときに特別なコード ページ タスクを実行する必要があります。

これらのタスクの詳細については、「第 2 章 ClearQuest とコード ページ」を参照してください。

ユーザー データベースの作成

ユーザー データベースの作成と、事前定義されたスキーマへの関連付けには ClearQuest Designer を使用します。

詳細については、「第 3 章 データベースの管理」を参照してください。

スキーマ (プロセス モデル) の作成

まず、プロセス モデル、つまりスキーマに取り込むモデルを文書化します。次のような内容を文書化します。

- ユーザーのタイプとその役割の記述
- プロセスのモデル化に使用する状態とアクションの図
- アクションを制御するルールの記述
- データの表示と登録に使用するフォームやレポートの概略、またはフォームやレポートに含めるフィールドのリスト

こうした要件が明確になったら、ClearQuest Designer を使用して、新規スキーマを作成したり、既存のスキーマを変更して、目的のプロセス モデルを構築します。

この作業を支援するため、ClearQuest には事前定義されたスキーマ (すぐに利用できるスキーマ) が 8 つ用意されています (表 5)。

表 5 事前定義のスキーマ

スキーマ	説明
AnalystStudio	ClearQuest を Rational AnalystStudio との組み合わせで 使用できるフィールドとルールを備えたスキーマです。 Rational RequisitePro と連携するようにカスタマイズ されています。
Blank	システム フィールドのみで構成されています。スキーマ 構築用の最低限の構成ベースです。
Common	最もよく使用されるスキーマ要素を備えたベースレベルの 構成です。
DefectTracking	障害追跡プロセス用の機能を備えたスキーマです。
DevelopmentStudio	ClearQuest を Rational DevelopmentStudio との組み合わ せで使用できるフィールドとルールを備えたスキーマです。 Rational の Purify、Quantify、PureCoverage を操作する フィールドとルールが含まれます。
Enterprise	ClearQuest を Rational Suite Enterprise との組み合わ せで使用できるフィールドとルールを備えたスキーマです。 すべての Rational 製品に使用可能なフィールドとフック が含まれています。
TestStudio	ClearQuest を Rational TestStudio との組み合わせで使 用できるフィールドとルールを備えたスキーマです。また、 Rational の TeamTest、RequisitePro、Purify、Quantify、 PureCoverage を操作するフィールドとルールも含まれ ます。
UnifiedChangeManagement	ClearQuest を Rational UCM との組み合わせで使用でき るフィールドとルールを備えたスキーマです。

目的のプロセス フローをモデル化するスキーマを作成するには、ClearQuest Designer を使用
して次の処理を実行します。

- Blank スキーマから新規スキーマを作成し、変更依頼のレコードにフィールドを追加し
ます。次に、状態遷移マトリックスを作成して変更依頼の状態を移動させるアクション
を定義し、データの表示と変更に使用するフォーム、クエリー、レポートを作成します。
- 既存のスキーマ (前に作成したスキーマ、ClearQuest 付属の Common スキーマまたは
DefectTracking スキーマ) から新規スキーマを作成します。また、必要に応じて、作成した
スキーマを変更します。
- 既存のスキーマを変更します。

- 既存のスキーマにパッケージを追加します。ClearQuest パッケージはスキーマ コンポーネント (レコードタイプ、フィールド、タブ、フォーム、スクリプト、クエリー) のセットで、複雑な機能を既存のスキーマに一回で追加できます。パッケージにより追加される機能には、電子メール通知の送信、変更依頼履歴の記録、レコードへの添付ファイルの追加などがあります。使用可能なパッケージには、ClearQuest Designer の [パッケージ] メニューからアクセスできます。
- フックを追加して、カスタム処理を実施します。
- ほかの Rational ソフトウェア製品と統合します。Rational ClearCase、統一変更管理、ClearQuest Project Tracker、Rational Suite、Microsoft Visual Source Safe などのサードパーティの統合製品など、ほかの Rational 製品と ClearQuest を統合するための専用パッケージが用意されています。

詳細については、「第 4 章 ClearQuest スキーマの操作」、「第 5 章 スキーマのカスタマイズ」、「第 6 章 パッケージの適用」、「第 7 章 フォームの操作」、「第 10 章 ワークフローのカスタマイズにおけるフックの使用法」、「付録 A ClearQuest のスキーマとパッケージ」を参照してください。

ユーザー データベースの管理

スキーマを作成または変更する場合は ClearQuest Designer を使用し、ユーザー データベースを作成または更新して最新バージョンのスキーマを使用します。

詳細については、「第 8 章 ユーザーの管理」を参照してください。

Rational E-Mail Reader を使用した電子メール通知の有効化

Rational E-Mail Reader を設定して、ClearQuest クライアントの代わりに電子メールを使用して変更依頼の登録と変更を実行できるようにすることも可能です。

この機能を使用するためには、専用の電子メールアドレスを少なくとも 1 つ設定する必要があります。変更依頼の登録または変更を実行するユーザーは、この電子メールアカウントに電子メール メッセージを送信します。この場合の電子メール メッセージは、データベース サーバーで Windows サービスとして実行される Rational E-Mail Reader が認識できるよう、所定のフォーマットに従う必要があります。この所定のフォーマットに従った電子メール メッセージのテンプレートを作成して配布し、Rational E-Mail Reader が解析して処理できる電子メール メッセージの作成を支援してください。

また、オプションで、特定のイベントが発生したときに ClearQuest ユーザーに電子メールで通知するルールを作成することもできます。

この機能を実現するためには、電子メールで情報を受信するすべてのスキーマに E-mail パッケージを適用してください。このパッケージにより、状態なしレコードタイプ Email_Rule が作成されます。次に、変更依頼の状態遷移などの特定のイベントが発生したときに電子メールによる通知を実行するルールセットを作成します。

このためには、ClearQuest の構成を指定し、電子メール サーバーにアカウントを作成する必要があります。

詳細については、「第 12 章 ClearQuest 電子メールの管理」を参照してください。

Web アクセスの有効化

ClearQuest Web を設定して、ユーザーがインターネット ブラウザを使用して、ClearQuest クライアントで使用できる機能の多くを実行できるようにすることも可能です。この機能を使用するためには、ClearQuest Web と Microsoft IIS Web サーバー ソフトウェアを Web サーバーに設定します。詳細については、「第 11 章 ClearQuest Web の管理」と、Windows 版の『Rational Software サーバー製品インストールガイド』を参照してください。

データのインポートとエクスポート

オプションで、別のデータ ソースからのデータのインポートに ClearQuest インポート ツールを、別のデータ ソースへのデータのエクスポートに ClearQuest エクスポート ツールを、それぞれ使用できます。

詳細については、「第 13 章 データのインポートとエクスポート」を参照してください。

クエリー、グラフ、レポートの作成

オプションで、ClearQuest Designer を使用してクエリー、チャート、レポートの作成または変更を実行できます。クエリー、チャート、レポートを [共用クエリー] フォルダに保存して、すべての ClearQuest ユーザーからアクセス可能にすることができます。また、これらの作業をプロジェクト マネージャーなどの別のユーザーに委任することもできます。

メモ: レポートを作成するには、Rational SoDA ソフトウェアや Seagate Crystal Reports などのレポート ツールのライセンスを取得する必要があります。

詳細については、ClearQuest クライアントのヘルプと「第 8 章 ユーザーの管理」を参照してください。

ユーザー アカウントの管理

ユーザー アカウントの作成、データベースへのユーザー登録、ユーザー グループの作成、グループのデータベースへの登録など、ユーザー アカウントの管理には ClearQuest Designer を使用します。

詳細については、「第 8 章 ユーザーの管理」を参照してください。

レコードに対するセキュリティ制御の設定

ユーザー アカウント管理に関連する最も強力な機能の 1 つに、セキュリティ制御をレコードに対して設定できることが挙げられます。これにより、どのユーザー グループがどの情報を閲覧できるかを厳密に管理できます。たとえば、顧客には変更依頼の登録と表示を許可しながら、ほかの顧客や開発チームのメンバーが登録した、変更依頼に関して慎重な扱いを要する情報を表示不可にできます。

このようなセキュリティを実現するため、レコードの表示を特定のユーザー グループに制限できます。また、レコードのフォーム タブに表示されるフォルドへのアクセスを特定のユーザー グループに制限することもできます。

セキュリティ制御をレコードに設定するには、レコード タイプでセキュリティ コンテキスト フィールドを作成し、このフィールドに特定の値が設定されたレコードを参照する許可をユーザー グループに割り当てます。

詳細については、「第 9 章 Rational ClearQuest でのセキュリティの使用法」を参照してください。

管理機能の委託

ClearQuest 管理者は、自身にスーパー ユーザー権限を割り当て、ClearQuest ツールから使用できるすべての管理機能を実行できるようにします。

ただし、作業負荷を軽減し、開発チームのボトルネックになることを避けるために、ClearQuest Designer を使用して管理機能を別のユーザーに委託できます。別のスーパー ユーザーを設定したり、権限がより少ない別のタイプのユーザーや管理者を作成することができます。

- スーパー ユーザーは、ClearQuest のすべての権限が与えられています。スキーマとユーザー データベースの作成と削除を実行できる唯一のユーザー タイプです。
- アクティブ ユーザーは、開発チームの典型的なメンバーで、ClearQuest へのログオン、自身のパスワード、名前、電子メール アドレス、電話番号の変更、スキーマとデータベースの情報の表示 (変更は不可) を実行できます。
- すべてのユーザー/グループを表示権限を持つユーザーは別のユーザーやグループの情報を表示できます (変更は不可)。
- スキーマ デザイナーは、スキーマの変更とデータベースのアップグレードを実行できます (データベースの作成、削除と、別のユーザーの情報の編集は不可)。
- ユーザー管理者は、ユーザーとグループに関する情報の追加と編集、ユーザー権限の付与と取り消しを実行できます。

- 共用フォルダ管理者は、クエリーの作成と更新などにより、[共用クエリー] フォルダの内容を管理できます。
- 動的選択リスト管理者は、フォームに表示される動的選択リストの内容に関するすべての権限を持ちます。
- SQL エディタは、SQL クエリーが生成する SQL コードを編集できます。
- セキュリティ管理者は、どのユーザー グループがどのレコード タイプにアクセスできるかを決定できます。セキュリティ管理者は、SQL エディタと同じ権限も持ちます。

詳細については、「第 8 章 ユーザーの管理」を参照してください。

ClearQuest MultiSite のセットアップ

オプションで、ClearQuest MultiSite のインストールと設定を行うことができます。ClearQuest MultiSite は複数の保存場所を対象としてデータを自動的に複製します。そのため、異なる地域にいる開発者でも、同一のプロジェクトで作業することができます。

詳細については、『Rational ClearQuest MultiSite 管理ガイド』を参照してください。

ClearQuest 入門

Designer、メンテナンス ツール、ユーザー管理ツール、インポート ツール、エクスポート ツールなどの ClearQuest の管理アプリケーションを起動するには、[スタート] メニューから [プログラム]、[Rational Software] の順にポイントし、[Rational ClearQuest] をクリックします。

ClearQuest のインストールが終了したら、ログオンします。ログオンするには、ユーザー名に「admin」と入力し、パスワードを空白のままにして [OK] をクリックします。

管理アカウントのパスワードの設定

初めて ClearQuest をインストールする場合は、スーパー ユーザー権限を持つ、デフォルトの admin アカウントを使用できます。このアカウントを使用して、ClearQuest Designer、メンテナンス ツール、インポート ツール、エクスポート ツールを含む、すべての ClearQuest 管理ツールにアクセスできます。このデフォルト admin アカウントを使用すると、パスワードなしでログインできます。

セキュリティ上の理由から、ユーザー管理ツールを使用して、このデフォルト admin アカウントにパスワードを設定することをお勧めします。

詳細については、176 ページの「ユーザーの編集」を参照してください。

ツールバー、パレット、メニュー、ショートカット

ClearQuest Designer 以外のほとんどの ClearQuest 管理ツールでは、1 つのウィンドウに、主要なツール アイコンとコントロールが表示されます。

ClearQuest Designer には、ツールバー、パレット、動的メニュー、キーボード ショートカットを使用してアクセスできる、さまざまな機能が用意されています。ただし、これらのアイテムは常に表示されているわけではありません。管理アクティビティを開始する前に、ClearQuest Designer での操作やアクションの実行に使用する、これらの機能について知っておく必要があります。

ツールバーの用途

ClearQuest Designer には 2 つのツールバーがあります。

- デザイナ ツールバーには、一般的に使用されるメニュー オプションの一部のアイコン (スキーマのオープン、スキーマのチェックインとチェックアウト、データベースの更新など) が設定されています。
- フォーム レイアウト ツールバーには、フォーム上のコントロールを整列するアイコンが設定されています。

パレットの用途

ClearQuest Designer には、コントロールの作成とそのフォームへの追加に使用するパレットが 2 つあります。

- コントロール パレット: 特定のコントロールをフォームに追加します。
- フィールド リスト パレット: 特定のフィールドを (デフォルトのコントロールと共に) フォームに追加します。

メニューの用途

ClearQuest Designer の機能とコマンドの大部分は、プログラムのプルダウン メニューから利用できます。メニュー オプションは、使用する ClearQuest 機能によって変わります。たとえば、フォームを表示すると、[編集] メニューと [表示] メニューが変更され、そのフォームに適用される新しいメニューになります。

ショートカット メニューの用途

クリックされた項目のショートカット メニューを表示するには、グリッド内のセル、フォーム コントロール、[確認の出力] ウィンドウ内のエラー メッセージ、ワークスペース内の項目のいずれかを右クリックします。一部のメニュー オプションと、操作時間を短縮するための機能は、ショートカット メニューだけにあります。たとえば、デフォルトのレコード タイプを設定する機能は、レコード タイプのショートカット メニューだけにしかありません。

キーボード ショートカットとマウス操作

ClearQuest Designer には次のキーボードショートカット (表 6) とマウス操作 (表 7) が用意されています。

表 6 キーボード ショートカット

キー	結果
【ファイル】メニュー	
[CTRL] + [O]	スキーマを開くウィザードを表示します。
[CTRL] + [K]	[チェックアウト] ダイアログ ボックスを表示します。
[CTRL] + [I]	スキーマを確認し、[チェックイン] ダイアログ ボックスを表示します。
[CTRL] + [U]	スキーマのチェックアウトを取り消し、そのスキーマを ClearQuest Designer から削除します。
[CTRL] + [D]	スキーマの変更を確認します。
[CTRL] + [P]	[印刷] ダイアログ ボックスを表示します。
【編集】メニュー	
[CTRL] + [X]	選択された項目を削除します。
[CTRL] + [C]	選択された項目を作業中のウィンドウからクリップボードにコピーします。
[CTRL] + [V]	最後にクリップボードにコピーされた項目を作業中のウィンドウに貼り付けます。
[CTRL] + [R]	選択された項目のプロパティ ダイアログ ボックスを表示します。
[DEL]	選択された項目を削除します。
[CTRL] + [Z]	前回のアクションを取り消します。
[CTRL] + [SHIFT] + [→]	選択されたコントロールを横方向に等間隔に並べます。
[CTRL] + [SHIFT] + [↓]	選択されたコントロールを縦方向に等間隔に並べます。
[F9]	選択されたコントロールを縦方向の中央揃えにします。
[SHIFT] + [F9]	選択されたコントロールを横方向の中央揃えにします。

表 6 キーボード ショートカット

キー	結果
[フォーム レイアウト] メニュー	
[F7]	内容に基づいてコントロールのサイズを設定します。
[Ctrl] + [←]	選択されたコントロールの左端を揃えます。
[Ctrl] + [→]	選択されたコントロールの右端を揃えます。
[Ctrl] + [↑]	選択されたコントロールの上端を揃えます。
[Ctrl] + [↓]	選択されたコントロールの下端を揃えます。
[Ctrl] + [E]	選択されたコントロールの高さと幅を同じサイズに設定します。
キーボード アクセラレータ	
[F1]	選択された項目または作業中のウィンドウのヘルプを表示します。
[→]	選択された項目を展開します。
[←]	選択された項目を折りたたみます。
キーパッドの [*]	選択された項目のブランチをすべて展開します。
キーパッドの [+]	選択された項目を展開します。
キーパッドの [-]	選択された項目を折りたたみます。
[↓]	グリッド内の次の行、またはワークスペース内の次の項目に移動します。
[↑]	グリッド内の前の行、またはワークスペース内の前の項目に移動します。
[←]	グリッド内で 1 つ左のセルに移動します。
[→]	グリッド内で 1 つ右のセルに移動します。
[Ctrl] + クリック	項目 (グリッド内のセルなど) を不連続に選択します。

表 7 マウス操作

マウス操作	
ダブルクリック	グリッドまたはフォーム コントロール内のセルをダブルクリックすると、プロパティ ダイアログ ボックスが表示される場合があります。
ドラッグ アンド ドロップ	[フィールドリスト] ダイアログ ボックスのフィールドをフォーム コントロールにドラッグ アンド ドロップします。
右クリック	グリッド内のセル、フォーム コントロール、[確認の出力] ウィンドウ内のエラー メッセージ、ワークスペース内の項目のいずれかを右クリックすると、クリックされた項目のショートカット メニューが表示されます。

Rational ClearQuest データベースにアクセスする各クライアントには、その環境に有効な文字セットを指定する、オペレーティング システムのコード ページが設定されます。つまり、コード ページによって、特定のクライアント上で正しく表示される文字が決定されます。

複数の言語が存在する ClearQuest 環境では、クライアントが、さまざまな言語に基づいたオペレーティング システムを、オペレーティング システムのコード ページが互いに異なる設定で実行することがあります。このような環境では、オペレーティング システムのすべてのコード ページで共通な文字のみを ClearQuest に入力できます。

重要なメモ: Windows では、ClearQuest はさまざまなコード ページと言語の実行をサポートしており、その動作はテストにより保証されています (31 ページの表 10 を参照)。UNIX では、ClearQuest は ASCII (印字可能な英語文字) のみをサポートしています。UNIX クライアントを実行する場合は、ClearQuest の展開について適切な計画を立てる必要があります。

v2002.05.01 パッチ 02 より前のバージョンの ClearQuest では、異なるコード ページの文字をデータベースに入力すると、表示の問題、データの破壊、MultiSite の同期の失敗などいくつかの問題が発生していました。ClearQuest のこのバージョン (v2003.06.10) には、データベースが適切に処理できるデータのみを入力を許可するいくつかの新機能があります。この章の以下の項では、これらの新機能と、その使用手順について説明します。

メモ: 複数のコード ページによるデータのレプリカを作成することは、MultiSite の同期が失敗する原因になります。したがって、すべての ClearQuest MultiSite カスタマが展開に備えて、有効で、テストされ、サポートされているコード ページを選択することが必須です (31 ページの表 10 を参照)。

コード ページのタスクの構成

表 8 に、管理者が ClearQuest 環境で単一のコード ページの使用を課すために実行するタスクを示します。

表 8 管理者のコード ページ関連タスク

管理者のタスク	詳細情報
スキーマ リポジトリに対して適切なデータ コード ページ値を決定します。	29 ページの「ClearQuest データ コード ページの選択」を参照してください。
すべての製造元データベースの文字セットを設定します。	33 ページの「製造元データベースの文字セットの概要」を参照してください。
各スキーマ リポジトリのデータ コード ページ値を設定します。	35 ページの「ClearQuest のデータ コード ページ値の設定」を参照してください。
ClearQuest 環境に v2003.06.10 より前のバージョンの ClearQuest が含まれる場合は、CharacterSetValidation パッケージを適用し、データ コード ページ値をすべてのユーザー データベースにコピーします。	39 ページの「以前のバージョンの ClearQuest との相互運用」を参照してください。

ClearQuest のデータ コード ページの概要

ClearQuest には、各スキーマ リポジトリに対して指定される、「ClearQuest のデータ コード ページ」という設定があります。スキーマ リポジトリに関連付けられたすべてのユーザー データベースは、ClearQuest の同じデータ コード ページ値を使用します。この値は、データベース セットに単一のコード ページの使用を課し、選択したコード ページ以外の文字がデータベースに入力されないようにします。

ClearQuest のデータ コード ページと新規スキーマ リポジトリ

ClearQuest の新規スキーマ リポジトリを作成した後は、ClearQuest データ コード ページ値を適切な値に設定する必要があります (29 ページの「ClearQuest データ コード ページの選択」を参照)。データ コード ページ値を設定していない場合、デフォルト値である ASCII が使用され、データベースは ASCII (印字可能な英語文字) によるデータ入力に制限されます。

注意: UNIX では、ClearQuest は ASCII のみをサポートしています。UNIX クライアントを使用するカスタマは、ClearQuest のデータ コード ページを ASCII (推奨) に設定するか、UNIX システムのユーザーには Web クライアントでのみデータを入力させるかのいずれかを選択してください。

ただし、新規スキーマ リポジトリとサンプル データベースを作成する際に名前またはプロパティに ASCII 以外の文字を使用すると、ClearQuest によって、ClearQuest のデータ コード ページ値はメンテナンス ツールを実行しているシステムのオペレーティング システムのコード ページに自動的に設定されます。

メモ: ClearQuest のデータ コード ページが自動的に設定されるのは、スキーマ リポジトリを作成したときだけです。

ClearQuest のデータ コード ページ値を設定または変更する方法については、35 ページの「ClearQuest のデータ コード ページ値の設定」を参照してください。

ClearQuest のデータ コード ページと既存のスキーマ リポジトリ

以前のバージョンの ClearQuest で作成された ClearQuest データベースがある場合、それらのデータベースにはさまざまなコード ページによるデータが含まれていることがあります。ClearQuest のデータ コード ページを設定しても、データ ベース内のデータは選択したコード ページの文字に変換されません。データベースに、新規のコード ページの文字にマッピングされない文字があった場合、データが破壊されます。

重要なメモ: 既存のスキーマ リポジトリに対して ClearQuest のデータ コード ページを設定する前に、製造元データベース内のデータをコード ページの正しい文字に変換する必要があります。手順については、製造元データベースのマニュアルを参照してください。

既存のスキーマ リポジトリに対して ClearQuest のデータ コード ページを設定する方法については、37 ページの「既存のスキーマ リポジトリとユーザー データベース」を参照してください。

ClearQuest データ コード ページの選択

スキーマ リポジトリに対して選択する、ClearQuest のデータ コード ページ値は、クライアントとサーバーのプラットフォーム、実行する ClearQuest のバージョン、製造元データベースがサポートしている文字セットなど、さまざまな要因に依存します。データベースセットに適した、ClearQuest のデータ コード ページを決定するには、次の表 9 のガイドラインとフローチャートを使用してください。

ClearQuest のデータ コード ページを選択するためのガイドライン

- データの一貫性を保証するには、Rational によってサポートされているコード ページを選択する必要があります。これらのコード ページは ClearQuest での使用について完全にテストされ、承認されています。Rational がサポートしているコード ページのリストについては、31 ページの表 10 を参照してください。
- ClearQuest に、Rational によってテスト、サポートされていないデータ コード ページを使用することもできます。値をサポートされていないコード ページに設定することで、少なくとも ClearQuest MultiSite の動作は可能であり、コード ページ数の増加を防止できます（ただし、データ破壊は防止できません）。

- ClearQuest に選択するデータ コード ページ値は、すべての ClearQuest データベースに使用する製造元データベースによってサポートされている必要があります。すべてのユーザー データベースの製造元データベース文字セットは、ClearQuest のデータ コード ページと一致するように設定してください。製造元データベースの文字セットの詳細については、33 ページの「製造元データベースの文字セットの概要」を参照してください。
- すべての Windows クライアントは、実行するオペレーティング システムのコード ページが同じである必要があり、かつ、そのコード ページは ClearQuest のデータ コード ページと一致する必要があります。混合プラットフォーム環境 (Windows クライアントと UNIX クライアントの両方) の場合、またはクライアントによってオペレーティング システムのコード ページが異なる場合は、ClearQuest のデータ コード ページを 20217 (ASCII) に設定してください。このコード ページは、すべてのコード ページで共通の文字セットです。また、ClearQuest のデータ コード ページを ASCII 以外の値に設定し、UNIX システムを使用するすべてのユーザーに Web クライアントのみで ClearQuest を操作させるという使用モデルもあります。

表 9 ClearQuest のデータ コード ページ値を選択するためのガイドライン

質問	あり	なし
UNIX 上で動作する ClearQuest クライアント ¹ がありますか。	ClearQuest のデータ コード ページを 20127 (ASCII) に設定します。	次の質問に進みます。
Windows クライアントとサーバー ² でオペレーティング システムに複数のコード ページを使用しますか。	ClearQuest のデータ コード ページを 20127 (ASCII) に設定します。	次の質問に進みます。
v2002.05.01 パッチ 02 よりも前のバージョンの ClearQuest を実行する Windows クライアントまたはサーバーがありますか。	ClearQuest のデータ コード ページを 20127 (ASCII) に設定します。	ClearQuest のデータ コード ページを、表 10 の「Rational」によってサポートされている、Windows のコード ページ」のいずれかに設定します。選択する ClearQuest のデータ コード ページ値は、製造元データベースによってサポートされている必要があります。
¹ 「クライアント」という用語は、ネイティブな ClearQuest クライアント、ClearQuest Designer、E-Mail Reader のことを指します。 ² 「サーバー」という用語は、ClearQuest Web サーバー、Rational Shipping Server のことを指します。		

メモ: ClearQuest のデータ コード ページを NOCHECKING に設定することによって、スキーマ リポジトリに単一のコード ページの使用を課さないようにできます。このオプションを使用すると、ClearQuest では、入力するデータをデータベースに格納できるかどうかや、入力するデータが破壊されることなく ClearQuest クライアントに表示されるかどうかを確認されません。Rational では、ClearQuest のデータ コード ページをこの値に設定することをお勧めしていません。NOCHECKING 値が指定された場合、ClearQuest MultiSite は機能しません。

Rational によってサポートされている Windows のコード ページ

表 10 に、Rational によってサポートされている Windows のコード ページを示します。

メモ: UNIX クライアントを使用する場合、20127 (ASCII) のみが ClearQuest でサポートされているデータ コード ページです。

表 10 Rational によってサポートされている Windows のコード ページ

言語	Windows のコード ページ
英語	20127 (ASCII)
イタリア語、英語、オランダ語、スウェーデン語、スペイン語、デンマーク語、ドイツ語、ノルウェー語、フィンランド語、ブラジル ポルトガル語、フランス語	1252 (Latin-1)
中国語 - 簡体字	936 (GBK)
日本語	932 (Shift-JIS)

ClearQuest のデータ コード ページを選択した結果

データ コード ページ値を選択したら、このデータ コード ページ値がデータベースに与える影響について知る必要があります。コード ページ値が設定されると、ユーザーは特定のデータベースを読み取り専用モードでのみ開くことができる場合もあり、データベースを開くことがまったくできなくなる場合もあります。例を次に示します。

- ClearQuest のデータ コード ページを ASCII 以外の値に設定した場合、ユーザーは、コード ページが同じであるオペレーティング システムが動作している Windows クライアントからのみ、そのデータベースのデータを修正できます。コード ページが一致しない場合、データベースは読み取り専用モードで開かれます。
- ClearQuest のデータ コード ページを ASCII 以外の値に設定した場合、UNIX クライアントは、データベースに対して読み取り専用でアクセスできます (UNIX ユーザーは Web クライアントのみを使用することもできます。これにより、ASCII 以外のデータ コード ページ値が選択された場合のデータ破壊を防止できます)。

- ClearQuest のデータ コード ページを ASCII 以外の値に設定しても、無効な文字が、ClearQuest によって検出されことなくデータベースに入力される可能性があります。たとえば、ClearQuest は、電子メールまたは Web ページから切り取られ、データベースレコードに貼り付けられるテキストを確認できません。電子メールまたは Web ページのテキストに、ClearQuest のデータ コード ページにない文字が含まれている場合、文字は表示時に破壊され、無効な文字 (たとえば、疑問符 (?) の文字) として表示されることがあります。
- ClearQuest v2003.06.10 はすべてのデータ入力を確認します。これには、Web クライアントまたは Rational E-Mail Reader によって入力されたデータも含まれます。統合機能 (ClearQuest と Microsoft Project 2000 の統合など) によって入力されたデータも確認されます。ClearQuest のデータ コード ページのデフォルト設定は 20127 (ASCII) です。
- ClearQuest のデータ コード ページ値が Web サーバーのオペレーティングシステムのコード ページと一致しない場合、ユーザーは ClearQuest Web クライアントでデータベースを開くことができません。
- ClearQuest は添付ファイル内に入力されたデータ (たとえば、ClearQuest レコードに添付された Word ドキュメント内のデータなど) を確認しません。選択した ClearQuest のデータ コード ページにないデータをレコードに含める必要がある場合は、添付ファイル内にそれを入力できます。ただし、添付ファイルに格納されているデータに対してクエリーを実行することはできません。また、レポート、グラフはこのデータを利用しません。

ClearQuest のデータ コード ページ値の変更

スキーマリポジトリに対する ClearQuest のデータ コード ページはいつでも、データを破壊することなく 20127 (ASCII) から ASCII 以外のコード ページに変更できます。

ただし、ClearQuest のデータ コード ページを最初から ASCII 以外の値に設定していた場合、それをほかのコード ページに変更すると、データが破壊されることがあります。ClearQuest のデータ コード ページ値を ASCII 以外の値から変更する場合は、その前に、既存のすべてのデータを新規のコード ページによってサポートされる文字に変換してください。手順については、製造元データベースのマニュアルを参照してください。

また、ClearQuest のデータ コード ページ値を NOCHECKING からそれ以外の値に変更すると、データ破壊が発生することがあります。ClearQuest のデータ コード ページ値を NOCHECKING から変更する場合は、その前に、既存のすべてのデータを新規のコード ページによってサポートされる文字に変換してください。手順については、製造元データベースのマニュアルを参照してください。

ClearQuest MultiSite 環境では、データ コード ページを ASCII 以外の値から変更すると、oplog にも不正なデータが入力されることがあります。したがって、Rational では、すべてのレプリカを削除し、マスター側のサイトでデータベースを消去し、oplog ログをスクラブしてから、レプリカを再作成することをお勧めします。レプリカの削除と再作成については、『Rational ClearQuest MultiSite 管理ガイド』を参照してください。

製造元データベースの文字セットの概要

ClearQuest 環境での各製造元データベースには、スキーマ リポジトリに選択した、ClearQuest のデータ コード ページによるデータをサポートする文字セットを使用する必要があります。製造元データベースの文字セットが ClearQuest のデータ コード ページをサポートしていない場合、データベースは ClearQuest によって入力される情報を正しく格納できません。ClearQuest は、製造元データベースの文字セットがスキーマ リポジトリに対する ClearQuest のデータ コード ページと互換性があるかどうかを確認することで、この問題を発生しにくくしています。この確認は、以下の状況のいずれかが発生した場合に行われます。

- ClearQuest のデータ コード ページを手動で設定したとき (35 ページの「ClearQuest のデータ コード ページ値の設定」を参照)
- スキーマ リポジトリを作成する際に、フィールドのいずれかで ASCII 以外の値が検出され、ClearQuest によってデータ コード ページが自動的に設定されたとき

次の例では、製造元データベースの文字セットが ClearQuest のデータ コード ページ設定と互換性があるかどうかを ClearQuest が確認してデータ破壊を防止する様子を示します。

例

ClearQuest 管理者がメンテナンス ツールを使用して、スキーマ リポジトリとサンプル ユーザー データベースを作成します。管理者が 1252 (Latin-1) コード ページにある ASCII 以外の文字を説明フィールドに入力します。ClearQuest はこれらの ASCII 以外の文字を検出し、SQL Server データベースの文字セットがそれらの文字を扱うことができるかどうかを確認します。メンテナンス ツールでは、以下のように確認状況が表示されます。

マスター データベース、サンプル データベースのいずれか、または両方のプロパティに ASCII 以外の文字セットを入力しました。

マスター データベースのデータ コード ページ サポートを確認しています ; 1252 (ANSI - Latin I) 文字セット。

サンプル データベースのデータ コード ページ サポートを確認しています ; 1252 (ANSI - Latin I) 文字セット。

このスキーマ リポジトリとサンプル ユーザー データベースは Latin-1 コード ページの文字を含んでいたため、ClearQuest は ClearQuest のデータ コード ページを次のように 1252 (Latin-1) に設定します。

マスター データベースをプラットフォームのコード ページに設定しています ; 1252 (ANSI - Latin I) 文字セット。

データベース MASTR がコード ページ 1252 (ANSI - Latin I) をサポートするかを確認しています...

すべてのデータベースの確認が成功しました。

マスター データベースのコード ページを 1252 (ANSI - Latin I) 文字セットに設定しました。

指定先の位置を確認しています...

ソースにログオンしています...

指定先にログオンしています...

スキーマを初期化しています...

製造元データベースの文字セットの設定

各製造元データベースには、Rational によってサポートされる、ClearQuest のデータ コード ページ値と等価な文字セットが 1 つ以上あります (31 ページの表 10 を参照)。34 ページの表 11 に、Oracle、SQL Server、SQL Anywhere、DB2 の各データベースで等価な文字セットのリストを示します。文字セットの完全なリストと、特定の製造元データベースの文字セットを設定する手順については、167217701 を参照してください。このソリューションは、Rational Support Web サイト (<http://www.rational.com/support>) からアクセスできます。ただし、英語のみのご利用となります。

メモ: 既存の ClearQuest データベースがある場合は、選択した ClearQuest のデータ コード ページによるデータをサポートするように、データベースの文字セットを変更する必要がある場合があります。既存のデータベースの文字セットを変更するには、通常、古いデータを新しいデータベースに移動することが必要になります。既存のデータが ClearQuest のいずれのデータ コード ページでもサポートされていない場合は、まず、データをサポートされているコード ページのいずれかの値に変換する必要があります (31 ページの表 10 を参照)。データベース製造元は、データを分析し、変換するツールを提供しています。詳細については、Rational Solution 167217701 を参照してください。ただし、英語のみのご利用となります。

表 11 データベース製造元の文字セットと Windows コード ページ

ClearQuest の データ コード ページ	Oracle	SQL Server	SQL Anywhere	DB2
20127	US7ASCII	SQL_Latin1_ General_CP437	437	ASCCSID
1252	WE8MSWIN1252	SQL_Latin1_ General_CP1	英語: 437 ヨーロッパ: 850	SCCSID または 1252
936	ZHS16GBK	Chinese_PRC	EUC_CHINA	1386
932	JA16SJIS	Japanese	SJIS	943
Access に関するメモ: Access データベースを使用する場合、製造元データベースの文字セットを設定する必要はありません。				

メモ: ClearQuest のデータ コード ページが NOCHECKING に設定されている場合、コード ページの確認は行われません。NOCHECKING 値が使用されている場合、ClearQuest MultiSite は機能しません。

ClearQuest のデータ コード ページ値の設定

以下の項では、新規スキーマ リポジトリ、以前のバージョンの ClearQuest で作成されたスキーマ リポジトリ、MultiSite 環境の既存のスキーマ リポジトリに対して ClearQuest のデータ コード ページ値を設定する方法について説明します。ClearQuest 環境に以前のバージョンの ClearQuest を実行するクライアントがある場合、詳細については 39 ページの「以前のバージョンの ClearQuest との相互運用」を参照してください。

新規スキーマ リポジトリ

この手順では、スキーマ リポジトリにユーザー データベースがないことと、ClearQuest Designer からいずれのユーザーもスキーマ リポジトリにアクセスしていないことを前提にしています。

- 1 **メモ:** この例のコマンドは 2003.06.10 のスキーマ リポジトリを使用します。スキーマ リポジトリに対して ClearQuest のデータ コード ページ値を選択します。29 ページの「ClearQuest データ コード ページの選択」を参照してください。
- 2 製造元データベースでサポートされている文字セットを確認します。33 ページの「製造元データベースの文字セットの概要」を参照してください。
- 3 コマンドプロンプトを開き、ClearQuest をインストールしたディレクトリ (たとえば C:\Program Files\Rational\ClearQuest) に移動します。
- 4 installutil lscodepage コマンドを実行して、オペレーティング システムのコード ページを特定します。

```
installutil lscodepage -dbset 2003.06.10 admin_user admin_password
```

```
Code page of 2003.06.10: 20127 (US-ASCII) (default)
```

```
Code page of client: 1252 (ANSI - Latin I)
```

メモ: この例では、オペレーティング システムのコード ページは 1252 (Latin-1) です。2003.06.10 のスキーマ リポジトリに対する ClearQuest のデータ コード ページ値は、現在、20127 (ASCII) というデフォルト値です。

- 5 以下のいずれかの **installutil** サブコマンドを実行して、**ClearQuest** のデータ コード ページを適切な値に設定します。

- Windows では、**Iscodepage** によって一覧表示されたオペレーティング システムのコード ページ値が **Rational** によってサポートされている場合は、**installutil setdbcodepagetopplatformcodepage** を実行して **ClearQuest** のデータ コード ページをこの値に設定します。**Rational** によってサポートされているコード ページのリストについては、31 ページの表 10 を参照してください。

```
installutil setdbcodepagetopplatformcodepage -dbset 2003.06.10 admin_user  
admin_password
```

- ClearQuest** のデータ コード ページを **20127 (ASCII)** に設定するには、**installutil setdbcodepagetoascii** コマンドを実行します。

```
installutil setdbcodepagetoascii -dbset 2003.06.10 admin_user admin_password
```

- Iscodepage** で一覧表示されたオペレーティング システムのコード ページ値がサポートされていないコード ページであり、データベース内のデータの破壊を防止する必要がない場合は、**-force** オプションを使用して **ClearQuest** のデータ コード ページをこの値に設定できます。ただし、31 ページの表 10 のリストにあるコード ページ以外のコード ページの使用は **Rational** ではサポートしません。

```
installutil setdbcodepagetopplatformcodepage -dbset 2003.06.10 -force admin_user  
admin_password
```

- ClearQuest** のデータ コード ページを **NOCHECKING** に設定することによって、単一のコード ページの使用を課さないようにできます。

警告 : **ClearQuest** クライアント間でオペレーティング システムのコード ページが異なり、ASCII 以外のデータが入力された場合、**ClearQuest** のデータ コード ページを **NOCHECKING** に設定すると、データ破壊されることがあります。このオプションは新規スキーマ リポジトリに対してはお勧めしません。また、このオプションを選択した場合、**ClearQuest MultiSite** は機能しません。

```
installutil setcodepagetonochecking -dbset 2003.06.10 admin_user admin_password
```

- 6 スキーマ リポジトリに対する **ClearQuest** のデータ コード ページが設定され、作成するユーザー データベースは、このコード ページ設定を継承します。

既存のスキーマ リポジトリとユーザー データベース

メモ: この例のコマンドは 2002.05.00 のデータベース セットを使用します。

既存のスキーマ リポジトリに対して ClearQuest のデータ コード ページを設定するには

- 1 スキーマ リポジトリに対して ClearQuest のデータ コード ページ値を選択します。
29 ページの「ClearQuest データ コード ページの選択」を参照してください。
- 2 製造元データベースでサポートされている文字セットを確認します。33 ページの「製造元データベースの文字セットの概要」を参照してください。

メモ: 既存の製造元データベースの文字セットが手順 1 で選択した ClearQuest のデータ コード ページと一致しない場合、製造元データベースの文字セットを変更し、サポートされていないデータをクリーンアップすることが必要になることがあります。特定の製造元データベースに対して文字セットを設定する手順については、製造元データベースのマニュアルを参照してください。また、Rational Solution 167217701 には、製造元データベース固有のマニュアルの場所を特定する方法に関する情報があります。このソリューションは Rational Support Web サイト (<http://www.rational.com/support>) からアクセスできます。ただし、英語のみのご利用となります。

- 3 いずれの ClearQuest クライアントもスキーマ リポジトリやユーザー データベースに接続していないことを確認してください。接続している場合、ClearQuest のデータ コード ページ値を設定した後も、接続しているクライアントが新規のコード ページにない文字を入力する可能性があります。
- 4 スキーマ リポジトリとすべてのユーザー データベースをバックアップします。
- 5 `installutil lscodpage` コマンドを実行して、オペレーティング システムのコード ページを特定します。

```
installutil lscodpage -dbset 2002.05.00 admin_user admin_password
```

```
Code page of 2002.05.00: 20127 (US-ASCII) (default)
```

```
Code page of client: 1252 (ANSI - Latin I)
```

メモ: この例では、オペレーティング システムのコード ページは 1252 (Latin-1) です。2002.05.00 のスキーマ リポジトリに対する ClearQuest のデータ コード ページ値は、現在、20127 (ASCII) というデフォルト値です。

- 6 以下のいずれかの `installutil` サブコマンドを実行して、**ClearQuest** のデータ コード ページを適切な値に設定します。

- Windows では、`lscodepage` によって一覧表示されたオペレーティング システムのコード ページ値が **Rational** によってサポートされている場合は、`installutil setdbcodepagetoplatformcodepage` を実行して **ClearQuest** のデータ コード ページをこの値に設定します。**Rational** によってサポートされているコード ページのリストについては、31 ページの表 10 を参照してください。

```
installutil setdbcodepagetoplatformcodepage -dbset 2002.05.00 admin_user  
admin_password
```

- ClearQuest** のデータ コード ページを 20127 (ASCII) に設定するには、`installutil setdbcodepagetoascii` コマンドを実行します。

```
installutil setdbcodepagetoascii -dbset 2002.05.00 admin_user admin_password
```

メモ : **ClearQuest** のデータ コード ページ値を別のコード ページから 20127 (ASCII) に変更した場合、ASCII 文字で表現できないすべてのデータが破壊されることがあります。

- `lscodepage` で一覧表示されたオペレーティング システムのコード ページ値がサポートされていないコード ページであり、データベース内のデータの破壊を防止する必要がない場合は、`-force` オプションを使用して **ClearQuest** のデータ コード ページをこの値に設定できます。ただし、このオプションの使用は **Rational** ではサポートしません。

```
installutil setdbcodepagetoplatformcodepage -dbset 2002.05.00 -force admin_user  
admin_password
```

- ClearQuest** のデータ コード ページを **NOCHECKING** に設定することによって、単一のコード ページの使用を課さないようにできます。

警告 : **ClearQuest** クライアント間でオペレーティング システムのコード ページが異なり、ASCII 以外のデータが入力された場合、**ClearQuest** のデータ コード ページを **NOCHECKING** に設定すると、データ破壊されることがあります。このオプションは新規スキーマ リポジトリに対してはお勧めしません。また、このオプションを選択した場合、**ClearQuest MultiSite** は機能しません。

```
installutil setcodepagetonochecking -dbset 2002.05.00 admin_user admin_password
```

- 7 スキーマ リポジトリに対する **ClearQuest** のデータ コード ページが設定され、作成するユーザー データベースは、このコード ページ設定を継承します。

以前のバージョンの ClearQuest との相互運用

Rational では、すべての ClearQuest クライアントを ClearQuest v2003.06.10 にアップグレードすることをお勧めします (MultiSite 固有の要件については、41 ページの「MultiSite と ClearQuest のデータ コード ページ値」を参照)。以前のバージョンの ClearQuest には、ClearQuest 環境で単一のコード ページの使用を課す機能がありません。また、ClearQuest v2002.05.01 パッチ 02 より前のバージョンの ClearQuest を実行する場合、ASCII データ入力 のみに制限されます (表 930 ページを参照)。

ClearQuest v2003.06.10 には、以前のクライアントを使用している場合に ClearQuest レコードでのデータ破壊を防止するために役立つ **CharacterSetValidation** パッケージが含まれています。ただし、スキーマ、クエリー名、レコード名、レポート書式、ユーザー プロファイル情報など、ほかのすべての ClearQuest オブジェクトでは、データ破壊が発生する可能性があります。ユーザーには、ClearQuest のデータ コード ページで有効な文字のみを使用してこれらのオブジェクトを作成するよう指示します。

ClearQuest 管理者は、CharacterSetValidation パッケージをすべてのスキーマに適用し、ClearQuest のデータ コード ページ値を、関連付けられたすべてのユーザー データベースにコピーしてください。

CharacterSetValidation パッケージの概要

CharacterSetValidation パッケージは、ClearQuest の以前のバージョンを実行するクライアントから、データベースのデータ コード ページ値以外のコード ページによるデータをユーザー レコードに入力できないようにします。CharacterSetValidation パッケージをスキーマに適用していない場合、ユーザーがクライアントから入力したサポートされていないデータが、あるクライアント上で修正されると、データが破壊される可能性があります。

CharacterSetValidation パッケージは、ユーザーが修正可能なすべてのレコード タイプに次のような 2 つの Perl フックを追加します。

- ベース アクセス制御フック「**CheckCodePageMismatch**」は、ClearQuest のデータ コード ページとクライアントのオペレーティング システムのコード ページに不一致があった場合にレコードへの読み取り専用アクセスを許可します。
- ベース確認フック「**CharacterSetValidation**」は、レコードの修正後に ClearQuest のデータ コード ページにない文字があれば、それらの文字を拒否します。

このパッケージは新しい状態なしレコード タイプ `_ratl_data_code_page` を追加します。このレコード タイプは以前のバージョン ClearQuest を実行するクライアントをサポートします。

CharacterSetValidation パッケージの適用

ClearQuest v2003.06.10 には、CharacterSetValidation パッケージをインストールするためのスクリプトが含まれ、これにより、スキーマ内のすべてのレコードに同時にこのパッケージが適用されます。また、パッケージ ウィザードを使用してこのパッケージを適用することもできますが、パッケージ ウィザードはパッケージを 1 つのレコードに適用し、すべてのレコードが有効化されるまで再実行されるため、より長い時間を必要とします。このパッケージは、スキーマに新規のレコードタイプを追加するたびに再適用する必要があります。

メモ: すべてのクライアントが ClearQuest v2003.06.10 以降を実行する場合、CharacterSetValidation パッケージを適用する必要はありません。

各スキーマに対し、次の手順に従って CharacterSetValidation パッケージを適用します。

- 1 ClearQuest Designer から、スキーマにチェックインし、各ユーザー データベースによってスキーマの最新のバージョンが使用されていることを確認します。または、手順 4 でユーザー データベースをアップグレードする際、パッケージの適用に関係しないスキーマ変更を含めることができます。

- 2 スクリプト `cquest-home-dir¥apply_character_set_validation_package.bat` を実行します。

```
apply_character_set_validation_package admin-username admin-password myDB
myDBSet schema
```

各パラメータの内容は次のとおりです。

`myDB` 修正中のスキーマを使用しているユーザー データベースのいずれか 1 つ

`myDBSet` そのスキーマのデータベース セット名

`schema` CharacterSetValidation パッケージを適用中のスキーマ

- 3 `apply_character_set_validation_package` スクリプトは、新しいバージョンのスキーマへのチェックインを試行し、確認エラーが発生すると失敗します。スキーマへのチェックイン時に確認エラーが発生した場合は、カスタマ サポートにお問い合わせください。
- 4 ClearQuest Designer にログインし、ユーザー データベースをアップグレードします。
- 5 ClearQuest のデータ コード ページ値をすべてのユーザー データベースにコピーします。詳細については、「ClearQuest のデータ コード ページ値のユーザー データベースへのコピー」を参照してください。

ClearQuest のデータ コード ページ値のユーザー データベースへのコピー

スキーマに CharacterSetValidation パッケージを適用した後、installutil setuserdbcodepage コマンドを実行して、ClearQuest のデータ コード ページ値をユーザー データベースにコピーしてください。この値をコピーすることで、以前のバージョンの ClearQuest を実行するクライアントがデータ コード ページ情報にアクセスできるようになり、選択したコード ページにない文字がデータベースに入力されなくなります。

- ClearQuest のデータ コード ページを特定のユーザー データベースにコピーするには、次のようにデータベース名を指定します。

```
installutil setuserdbcodepage -dbset myDBSet admin_user admin_password user_db_name
```

- ClearQuest のデータ コード ページをすべてのユーザー データベースにコピーするには、次のようにデータベース名を指定しません。

```
installutil setuserdbcodepage -dbset myDBSet admin_user admin_password
```

ClearQuest のデータ コード ページを変更した場合、setuserdbcodepage コマンドをもう一度実行して、この値をユーザー データベースにコピーし直す必要があります。

MultiSite と ClearQuest のデータ コード ページ値

すべての MultiSite 同期サーバーは同じコード ページであるオペレーティング システムのコンピュータで実行される必要があります、ClearQuest v2003.06.10 以降を使用する必要があります。一時的な制限ですが、すべての MultiSite 同期サーバーは Windows コンピュータ上で実行される必要があります。

メモ: クライアントをアップグレードする前に MultiSite 同期サーバーを v2003.06.10 にアップグレードする必要があります。

MultiSite 環境でデータ コード ページ値を設定すると、パケットのエクスポート中にデータベースのデータ コード ページ値とクライアントのオペレーティング システムのコード ページが比較されます。コード ページが同じである場合または ClearQuest のデータ コード ページ値が 20127 (ASCII) の場合のみ、エクスポート操作が正常終了します。次に、エクスポートされているパケットのヘッダーに ClearQuest のデータ コード ページ値が書き込まれます。インポートが正常終了するのは、ヘッダーにある ClearQuest のデータ コード ページ値がインポート元プロセスのオペレーティング システムのコード ページと同じであるか、そのサブセットであるかのいずれかの場合のみです。

UNIX 上では、ClearQuest は ASCII (印字可能な英語文字) のみをサポートしていることに注意してください。UNIX クライアントを実行する場合は、ClearQuest のデータ コード ページを 20127 (ASCII) に設定するか、Web クライアントのみからデータを入力する必要があります。

MultiSite による同期の失敗の例

ClearQuest MultiSite では、レプリカにアクセスするすべてのクライアントが同じコード ページを使用していない場合、無効なデータが oplog に入力されることがあります。これが発生した場合、同期は失敗し、次のようなエラーが表示されます。

Multiutil: XML データをローカルのコード ページに変換できません。

不正な文字が次の場所で見つかりました：

行 30

列 0

バイト 920

oplog シリアル番号 185458

エクスポートされたパケットは、データベース 'SAMPL'、複製 'SITEA' (UUID =1802CAEC-44D1-11D5-AF98-00B0D0682333) の oplog シリアル番号 185458 でした。

Multiutil: syncreplica -export コマンドに失敗しました。

Multiutil: エラー：作業ディレクトリ 'c:\temp\export' を削除できません (The directory is not empty.)

無効なデータがレコードから削除され、oplog が修復されるまで、レプリカを同期することはできません。

ClearQuest 統合と ClearQuest のデータ コード ページ値

ClearQuest 統合を使用する際に注意することがいくつかあります。ClearQuest データベース内のすべてのデータが同じコード ページによるものでなければならないためです。

- 統合を正常に設定するには、統合される製品が、データベース セットの ClearQuest のデータ コード ページと、オペレーティング システムのコード ページが同じであるコンピュータ上で実行される必要があります。そうでない場合は、ClearQuest のデータ コード ページを 20217 (ASCII) に設定してください。
- 統合機能によって ASCII 以外のデータを ClearQuest データベースに入力するには、統合する製品が、データベース セットの ClearQuest のデータ コード ページと、オペレーティング システムのコード ページが同じであるコンピュータ上で実行される必要があります。ClearQuest は、異なるコード ページを使用する統合製品による ASCII 以外のデータをブロックします。
- ClearQuest のデータ コード ページ値が 20217 (ASCII) の場合、統合機能は ASCII 以外のコード ページによるすべての文字を拒否することに注意してください。ただし、ClearQuest に統合する製品では、データ入力は ASCII 文字に制限されていません。統合された製品からは ASCII 以外の文字に関する警告が表示されませんが、ASCII データ コード ページ値で設定された ClearQuest データベース セットは、統合された製品から入力される ASCII 以外のすべての文字を拒否します。

例

以下の例では、オペレーティング システムのコード ページと ClearQuest のデータ コード ページの組み合わせが異なる状況での、ClearQuest 統合の動作を示します。

- この例では、コード ページの不一致があった場合に Microsoft Project 2000 との ClearQuest 統合を設定できなくなる様子を示します。

ClearQuest の管理者はスキーマ リポジトリとユーザー データベースを作成し、ClearQuest のデータ コード ページを 1252 (Latin-1) に設定します。管理者は、コード ページが日本語であるオペレーティング システムのシステム上で実行される Microsoft Project 2000 の新規データベースにリンクを試みます。このリンクは失敗し、管理者に次のエラー メッセージが表示されます。

ClearQuest データベース コード ページはクライアント マシンのコード ページと互換性がありません。このため、Project Tracker は処理を継続できません。

- この例では、統合自体が正常に完了しても、コード ページの不一致によって Project Tracker が機能しなくなる様子を示します。

ClearQuest の管理者はスキーマ リポジトリとユーザー データベースを作成し、ClearQuest のデータ コード ページを 20127 (ASCII) に設定します。ユーザーは日本語の Windows オペレーティング システム上に ClearQuest をインストールします。ユーザーは ClearQuest で障害を作成し、修正します。ユーザーは Project Tracker のセットアップを正常に終了します。ユーザーは Microsoft Project 2000 で日本語の文字を含んだ計画を作成します。

Microsoft Project 2000 から、ユーザーは Project Tracker の機能である [ClearQuest に新規タスクをエクスポート] を選択します。ClearQuest によって、タスクに ASCII 以外の文字が含まれていることが確認され、エクスポートの実行が中断されます。ユーザーは Microsoft Project 2000 タスクから ASCII 以外の文字を削除し、エクスポートをもう一度実行する必要があります。

Rational ClearQuest の管理者が担当する業務の大半は、データベース関連エンティティの管理に関連しています。このため、ClearQuest メンテナンス ツールと ClearQuest Designer を使用して ClearQuest データベースを管理する方法について十分に理解する必要があります。

ClearQuest メンテナンス ツールでは、スキーマ リポジトリと接続の作成/管理、サンプル データベースの作成、スキーマ リポジトリとユーザー データベースの ClearQuest の新しい機能レベルへのアップグレードが可能です。

ClearQuest Designer では、ユーザー データベースの作成/管理、ユーザー データベースのスキーマの新しいバージョンへのアップグレードが可能です。

この章の内容は、次のとおりです。

- 46 ページの「データベースの操作プロセスの理解」
- 49 ページの「データベースをバックアップする手順の確立」
- 51 ページの「新規スキーマ リポジトリとサンプル データベースの作成」
- 58 ページの「接続の管理」
- 65 ページの「スキーマ リポジトリの移動」
- 68 ページの「データベースを ClearQuest の新しい機能レベルにアップグレードする」
- 71 ページの「ユーザー データベースの移動と削除」
- 76 ページの「製造元データベースのパラメータ」

データベースの操作プロセスの理解

ClearQuest 管理者は、さまざまなデータベース (主にスキーマ リポジトリとユーザー データベース) を作成し、管理します。スキーマ、スキーマ リポジトリ、ユーザー データベース、データベース接続の概要については、「第 1 章 ClearQuest 管理について」を参照してください。

これらのデータベースを管理するには、ClearQuest メンテナンス ツールと ClearQuest Designer を使用します。また、ClearQuest で使用するデータベース管理システムの製造元のツールも使用します。

表 12 に、プロジェクトの工程を通じて実行され得る一般的な作業手順を示します。

表 12 ClearQuest データベース アクティビティ

作業	ツール/ インターフェイス	コメント
データベース管理システムをインストールします。	データベース製造元のツール	ランタイム バージョンの Microsoft Access は、ClearQuest のインストールの際にインストールされます。SQL Anywhere は ClearQuest インストール ディスクにあります。SQL Anywhere をインストールするように選択しないとインストールされません。ほかのデータベースは、そのデータベースの製造元からライセンスを取得し、提供されるメディアからインストールしてください。
データベースとスキーマ リポジトリをバックアップする手順を設定します。	データベース製造元のツール	バックアップを定期的に行います。スキーマ リポジトリとユーザー データベースのバックアップは同時に行ってください。 49 ページの「データベースをバックアップする手順の確立」とデータベース製造元のマニュアルを参照してください。
空の製造元データベースとデータベースのエイリアス (別名) を作成します。	データベース製造元のツール	Microsoft Access、SQL Anywhere の場合は、データベースを ClearQuest メンテナンス ツールから作成できます。 Oracle と DB2 では、データベースのエイリアス (別名) も作成する必要があります。

表 12 ClearQuest データベース アクティビティ

作業	ツール/ インターフェイス	コメント
製造元データベースの文字セット (コード ページ) を設定します。	データベース 製造元のツール	製造元データベースの文字セットは、スキーマ リポジトリの ClearQuest データ コード ページ 値と一致している必要があります。適切な データ コード ページ値の選択については、29 ページの「 ClearQuest データ コード ページ の選択」を参照してください。製造元データベース コード ページの設定方法については、50 ページの「製造元データベースの文字セットの設定」を参照してください。
新規スキーマ リポジトリと新規接続を作成します。	ClearQuest メンテナンス ツール	先に作成しておいた空の製造元データベースのいずれかに構築します。51 ページの「新規スキーマ リポジトリとサンプル データベースの作成」を参照してください。
スキーマ リポジトリの ClearQuest データ コード ページ値を設定します。	installutil コマンド 行ユーティリティ	ClearQuest データ コード ページ値は、データベース セットでのデータの破損を防止します。適切なデータ コード ページ値の選択については、29 ページの「 ClearQuest データ コード ページの選択」を参照してください。データ コード ページ値の設定方法については、35 ページの「 ClearQuest のデータ コード ページ値の設定」を参照してください。
バージョン 2003.06.10 より前の ClearQuest を実行しているクライアントからスキーマ リポジトリにアクセスする場合は、 CharacterSetValidation パッケージをスキーマ リポジトリに適用します。	ClearQuest Designer	CharacterSetValidation パッケージは、以前のバージョンの ClearQuest を実行しているクライアントから有効なデータのみが入力されていることを確認します。詳細については、146 ページの「パッケージのアップグレード方法」を参照してください。
サンプル データを使用して、サンプル ユーザー データベースを作成します。	ClearQuest メンテナンス ツール	サンプル データベースを使用して、スキーマ とプロセス モデルを確認します。51 ページの「新規スキーマ リポジトリとサンプル データベースの作成」を参照してください。
新しいデータベースの作成とスキーマへの関連付けを行います。	ClearQuest Designer	ユーザー データベースはスキーマの青写真を使用します。「第 4 章 ClearQuest スキーマの操作」を参照してください。

表 12 ClearQuest データベース アクティビティ

作業	ツール/ インターフェイス	コメント
CharacterSetValidation パッケージをスキーマ リポジトリに適用する場合は、各ユーザー データベース (テストと実動用) に ClearQuest データ コード ページ値を設定する必要があります。	installutil コマンド 行ユーティリティ	ClearQuest データ コード ページ値は、CharacterSetValidation パッケージで使用され、以前のバージョンの ClearQuest を実行しているクライアントから有効なデータのみが入力されていることを確認します。詳細については、35 ページの「ClearQuest のデータ コード ページ値の設定」を参照してください。
接続の管理と接続プロファイルの作成を行います。	ClearQuest メンテナンス ツール	接続は、ユーザーがスキーマ リポジトリと関連付けられたユーザー データベースにアクセスできるようにするために使用されます。「第 3 章 接続の管理」を参照してください。
新規スキーマの作成または既存のスキーマの変更を行います。	ClearQuest Designer	スキーマのバージョン番号は自動的に増加します。スキーマを変更しても、ユーザー データベースは、アップグレードされるまで変更されません。81 ページの「ClearQuest スキーマの操作」を参照してください。
テスト データベースを作成します。	ClearQuest Designer	新規スキーマまたは変更したスキーマをテストするために使用します。81 ページの「ClearQuest スキーマの操作」を参照してください。
更新したスキーマにチェックインし、ユーザー データベースをアップグレードします。	ClearQuest Designer	スキーマに加えた変更が、アップグレードされたデータベースに反映されます。「第 4 章 ClearQuest スキーマの操作」を参照してください。
ユーザーに対する、ユーザー データベース内データの追加と変更の許可を行います。	ClearQuest クライアント、 ClearQuest Web	ClearQuest ツールのみを使用して、ClearQuest のデータとスキーマを変更します。データベース製造元のツールを使用すると、データベースの一貫性が失われることがあります。 また、Rational E-Mail Reader と ClearQuest MailServer Service が実装されている場合、ユーザーは電子メールで変更依頼の登録と変更を行うことができます。

表 12 ClearQuest データベース アクティビティ

作業	ツール/ インターフェイス	コメント
ユーザー データベースの削除/削除取り消し、ほかの位置へのユーザー データベースの移動、ユーザー データベースを管理している製造元データベース管理システムの変更を行います。	ClearQuest Designer	ClearQuest Designer では、ユーザー データベースが削除される際に、スキーマ リポジトリとデータベースの間のリンクが削除されますが、データベースが物理的に削除されるわけではありません。物理データベースを削除するには、データベース製造元のツールを使用します。71 ページの「ユーザー データベースの移動と削除」を参照してください。
ほかの位置へのスキーマ リポジトリと接続の移動、スキーマ リポジトリを管理している製造元データベース管理システムの変更を行います。	ClearQuest メンテナンス ツール	65 ページの「スキーマ リポジトリの移動」を参照してください。
スキーマ リポジトリとユーザー データベースを Rational ClearQuest の新しい機能レベルにアップグレードします。	ClearQuest メンテナンス ツール	68 ページの「データベースを ClearQuest の新しい機能レベルにアップグレードする」を参照してください。操作する前に、Windows の場合は『Rational Suite アップグレード ガイド』を確認してください。

データベースをバックアップする手順の確立

データベースを作成する場合は、その種類にかかわらず、データベースを定期的にバックアップするための手順とスケジュールを設定することが重要です。バックアップの実行に使用するデータベース ツールは、必要に応じて選択できます。ほとんどのハイエンドデータベース製品には、定期的なバックアップの実行に使用できるツールが組み込まれています。

データベースのバックアップを実行するときには必ず、ClearQuest のすべてのデータベース (スキーマ リポジトリと関連付けられたすべてのユーザー データベースを含む) を同時にバックアップする必要があります。これにより、データとカスタマイズ情報の両方を保持でき、バックアップからの復元が必要になった場合にスキーマ リポジトリとユーザー データベースの同期がとれます。

また、ClearQuest データまたはテーブルを直接変更するときには ClearQuest ツールのみを使用してください。

データベースのバックアップで問題が発生した場合は、Rational 技術サポートにお問い合わせください。

空の製造元データベースとデータベースのエイリアス (別名) の作成

ClearQuest のスキーマ リポジトリとユーザー データベースはいずれも、データベース管理システムによって物理的に管理される種類のデータベースです。ClearQuest では、いくつかのデータベース管理システムの中から選択できます。選択できるデータベース管理システムはプラットフォームにもよりますが、Sybase SQL Anywhere、Microsoft Access、Microsoft SQL Server、Oracle DBMS、IBM DB2 などがあります。

スキーマ リポジトリとユーザー データベースを作成する前に、まず空のデータベースを作成する必要があります。

Microsoft SQL Server、Oracle DBMS、IBM DB2 を使用する場合は、その製造元から提供されるツールを使用して空のデータベースを作成する必要があります。

IBM DB2 と Oracle DBMS では、データベースのエイリアス (別名) を作成する必要があります。DB2 では、サーバー上と Windows と UNIX 両方のクライアント上に別名を作成する必要があります。Oracle では、サーバー上と Windows クライアント上にエイリアスを作成する必要があります。Oracle では Open Link がデータベース接続に使用されているので、UNIX クライアント上にエイリアスを作成する必要はありません。ただし、Oracle サーバー上に Open Link を設定する必要があります。DB2 と Oracle のエイリアス (別名) を作成する方法については、『Rational Software サーバー製品インストールガイド』、『Rational Software デスクトップ製品インストールガイド (Windows 版)』を参照してください。

SQL Anywhere、Microsoft Access の場合は、スキーマ リポジトリまたはユーザー データベースを作成するときに、空のデータベースを作成します。前もってこの手順を実行する必要はありません。

製造元データベースの文字セットの設定

ClearQuest では、1 つのデータベース セットに含まれるすべての製造元データベース、スキーマ リポジトリ、ユーザー データベースで単一のコード ページまたは文字セットを使用することが必要です。単一のコード ページを要求することによって、各データベースにはデータベース セットで有効なデータのみが格納されるように設定しています。

新規の製造元データベースを作成するときは、文字セットをスキーマ リポジトリの ClearQuest データ コード ページ値に対応する値に設定する必要があります。ClearQuest データ コード ページ値の詳細については、28 ページの「ClearQuest のデータ コード ページの概要」を参照してください。

各製造元データベースには、Rational がサポートする ClearQuest データ コード ページ値に相当する文字セットがいくつかあります。表 13 に、Oracle データベースの文字セットと、それに対応するデータ コード ページ値のリストを示します。適切な文字セットを選択する際に、このリストは参考になります。たとえば、スキーマ リポジトリに ClearQuest データ コード ページ値 932 が設定されていて、Oracle データベースを使用している場合、製造元データベースの文字セットは JA16SJIS に設定します。

製造元データベースの文字セットの設定方法については、製造元データベースのマニュアルを参照してください。

メモ: 既存のデータベースの文字セット値を変更すると、一部のデータが新しい文字セットに変換されない可能性があります。データベースの文字セットを変更する前に、製造元データベースのマニュアルを参照してください。

表 13 Oracle の文字セットと Windows のコード ページ

ClearQuest のデータ コード ページ値	Oracle の 文字セット	言語
1252	WE8MSWIN1252	デンマーク語、オランダ語、フィンランド語、フランス語、ドイツ語、英語、ブラジル語、ポルトガル語、イタリア語、ノルウェー語、スペイン語、スウェーデン語
20127	US7ASCII	英語 /ASCII
932	JA16SJIS	日本語
936	ZHS16GBK	中国語 (簡体字)

新規スキーマ リポジトリとサンプル データベースの作成

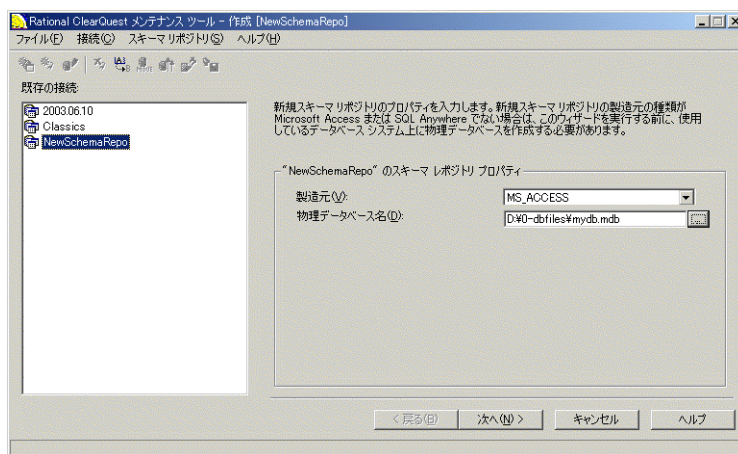
ClearQuest メンテナンス ツールを使用して、新規スキーマ リポジトリを作成します。スキーマ リポジトリは、スキーマのグループ (スキーマのバージョンのすべて) の格納と管理に使用される特殊な種類のデータベースです。スキーマ リポジトリは、マスター リポジトリまたはマスター データベースと呼ばれることもあります。

新規スキーマ リポジトリを作成すると、ClearQuest メンテナンス ツールによって接続が自動的に作成されます。この接続はデータベース セットとも呼ばれます。接続は、1 つのスキーマ リポジトリと、そのスキーマ リポジトリでスキーマに関連付けられているすべてのユーザー データベースで構成されます。

スキーマ リポジトリのデータベース管理システムとして Microsoft SQL Server、Oracle DBMS、IBM DB2 のいずれかを使用するには、その製造元から提供されるツールを使用して空のデータベースを作成しておく必要があります。また、DB2、Oracle のいずれかを使用するには、スキーマ リポジトリを作成する前にデータベース サーバー コンピュータのエイリアス (別名) を設定する必要があります。複数のスキーマ リポジトリを同じエイリアスに指定できます。スキーマ リポジトリごとに新規エイリアスを作成する必要はありません。

スキーマ リポジトリを作成するには

- 1 ClearQuest メンテナンス ツールで、[スキーマ リポジトリ] メニューの [作成] をクリックします。



- 2 [既存の接続] 領域で、選択された項目のスキーマ リポジトリ接続に付ける名前を入力し、[Enter] を押します。

デフォルトでは、最初の接続の名前は ClearQuest のバージョンである 2003.06.10 です。接続の名前は後で変更できます。

- 3 [<接続名> のスキーマ レポジトリ プロパティ] 領域の [製造元] でデータベースを選択し、必要なデータベース プロパティを入力します。必要なプロパティは、選択するデータベース製造元によって異なります。

製造元データベースとして Microsoft SQL Server、Oracle DBMS、IBM DB2 のいずれかを選択した場合は、データベース製造元のツールを使用して作成した空のデータベースのいずれかの物理データベース名を入力します。Oracle または DB2 を選択した場合は、データベース サーバー コンピュータのエイリアス (別名) を指定する必要があります。

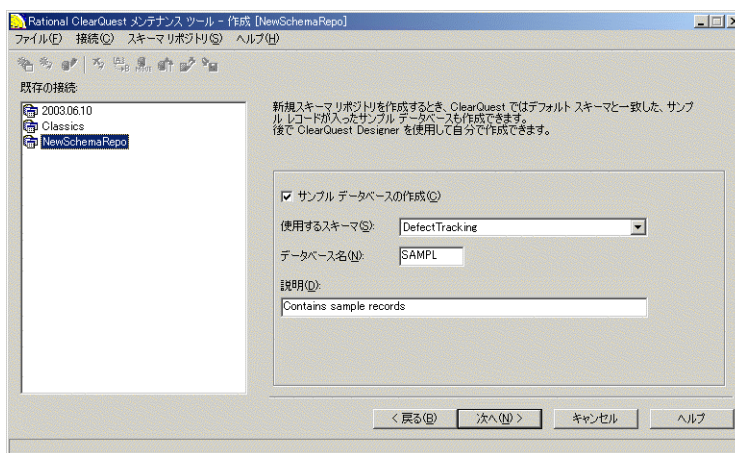
SQL Anywhere または Access を選択した場合は、物理データベース名を入力します。ClearQuest メンテナンス ツールによって、入力した名前でも新規データベースが作成されます。SQL Anywhere の場合、[ホスト名] は必須です。詳細については、76 ページの「製造元データベースのパラメータ」を参照してください。

メモ: ClearQuest バージョン 2003.06.10 では、SQL Anywhere 8.0 でのみ新規スキーマリポジトリを作成できます。

- 4 [次へ] をクリックします。
- 5 新規スキーマ リポジトリを作成するときに、サンプル ユーザー データベースを作成することもできます。サンプル ユーザー データベースは、ユーザー情報のサンプル障害レコード 40 個、サンプルクエリー、レポートを提供する作業用ユーザー データベースです。サンプル ユーザー データベースを使用して、ClearQuest の機能の理解を深め、ユーザーの教育に役立てることをお勧めします。

サンプル データベースを作成するには

- a [サンプル データベースの作成] チェック ボックスをオンにします。
- b [使用するスキーマ] で、サンプル データベースに使用するスキーマを選択します。
- c [データベース名] に、5 文字以下の英数字によるデータベース名を入力します。
デフォルトの名前は **SAMPL** です。
- d 必要に応じて、[説明] に、サンプル データベースの説明を入力します。



- 6 [次へ] をクリックします。
- 7 サンプル ユーザー データベース プロパティの領域で、[製造元] で、サンプル ユーザー データベースのデータベースの製造元を選択し、[物理データベース名] に、5 文字以下の英数字による物理データベース名を入力します。
- 8 [完了] をクリックします。

新しいスキーマ リポジトリのプロパティとサンプル ユーザー データベースに ASCII コード ページ以外の文字が含まれている場合、ClearQuest はスキーマ リポジトリの ClearQuest データ コード ページ値を自動的に設定します。この値には、メンテナンス ツールを実行しているコンピュータのオペレーティング システム コード ページが設定されます。それ以外の場合、ClearQuest データ コード ページ値は設定されないままです。この値を明示的に設定しない場合、データベースは ASCII データだけに制限され、スキーマ リポジトリを開くたびに、ClearQuest Designer に警告メッセージが表示されます。

ClearQuest データ コード ページ値の詳細については、28 ページの「ClearQuest のデータ コード ページの概要」を参照してください。データ コード ページ値の設定と変更方法については、35 ページの「ClearQuest のデータ コード ページ値の設定」を参照してください。

ユーザー データベースの作成

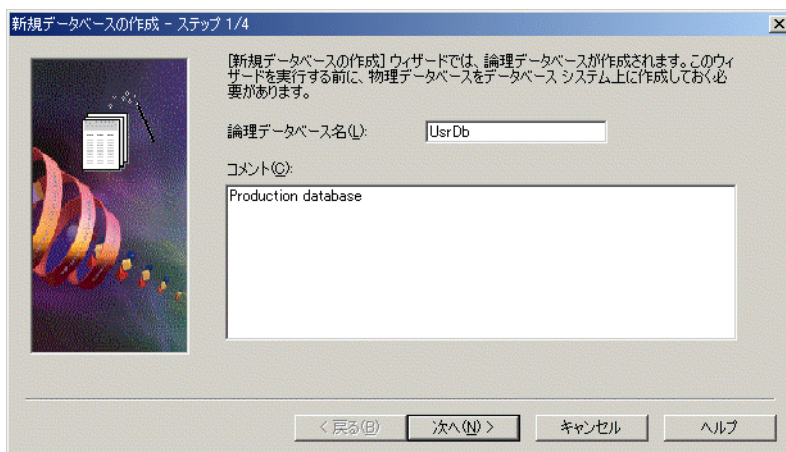
スキーマ リポジトリを作成した後は、ClearQuest Designer を使用して新規ユーザー データベースを作成できます。ユーザー データベースは、開発チームのメンバーが入力した変更依頼に関する情報を管理します。ユーザー データベースは常にスキーマ (そのデータベースのデータ、プロセス モデル、レポート、クエリを記述します) と関連付けられます。

スキーマ リポジトリを作成するときと同様、ユーザー データベースのデータベース管理システムとして Microsoft SQL Server、Oracle RDBMS、IBM DB2 のいずれかを使用する場合は、最初にその製造元から提供されるツールを使用して空のデータベースを作成する必要があります。

新規ユーザー データベースを作成するには

- 1 Windows の [スタート] メニューから、[プログラム]、[Rational Software]、[Rational ClearQuest] の順にポイントし、[ClearQuest Designer] をクリックして、ClearQuest Designer を起動します。
- 2 [スキーマを開く] ダイアログ ボックスが表示されたら、[キャンセル] をクリックします。
- 3 [データベース] メニューの [新規データベースの作成] をクリックします。

- 4 新規データベース ウィザードのステップ 1 の画面で、[論理データベース名] に、ユーザーデータベースに付ける論理データベース名を入力します。この名前は、英数字で 5 文字以下にする必要があります。

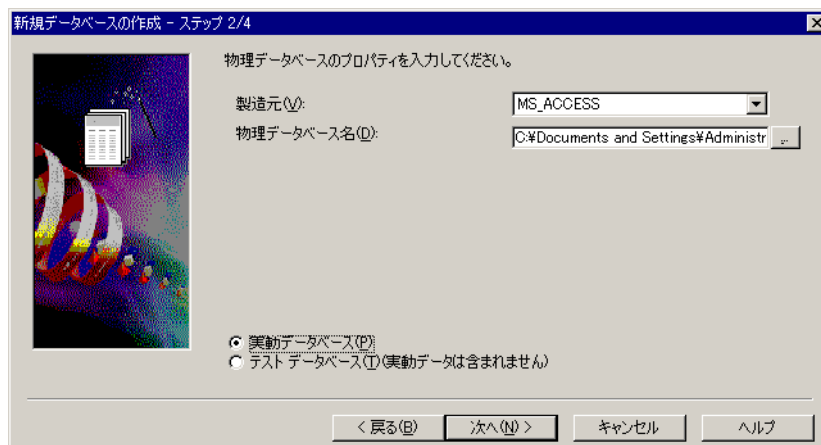


- 5 必要に応じて [コメント] にコメントを入力し、[次へ] をクリックします。
- 6 新規データベース ウィザードのステップ 2 の画面の [製造元] で、ユーザー データベースに使用するデータベースの製造元を選択し、必要なデータベース プロパティを入力します。必要なプロパティは、選択するデータベース製造元によって異なります。

製造元データベースとして Microsoft SQL Server、Oracle RDBMS、IBM DB2 のいずれかを選択した場合は、データベース製造元のツールを使用して作成した空のデータベースのいずれかの物理データベース名を入力します。

SQL Anywhere または Access を選択した場合は、物理データベース名を入力します。ClearQuest Designer によって、入力した名前で新規データベースが作成されます。SQL Anywhere の場合、[ホスト名] は必須です。詳細については、76 ページの「製造元データベースのパラメータ」を参照してください。

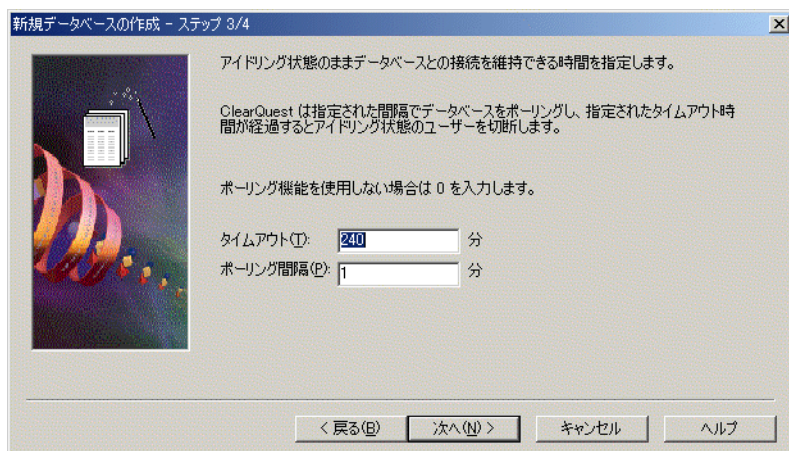
メモ: ClearQuest バージョン 2003.06.10 では、SQL Anywhere 8.0 でのみ新規スキーマリポジトリを作成できます。



- 7 [実動データベース] または [テスト データベース] のどちらかのユーザー データベースを選択し、[次へ] をクリックします。
- 8 ステップ 2 の画面の下部に、ユーザー データベースを実動データベースとテスト データベースのいずれにするかを指定するオプションがあります。デフォルトでは、[実動データベース] がオンになります。[次へ] をクリックします。

テスト データベースは、新規スキーマまたは既存のスキーマの新規バージョンを開発するときに作成する一時的なユーザー データベースです。テスト データベースを使用すると、新規スキーマが予想どおりに動作することを確認できます。これにより、ユーザー データベースにある実動データに損傷を与えないで済みます。テスト データベースの詳細については、「第 4 章 ClearQuest スキーマの操作」を参照してください。

- 9 新規データベース ウィザードのステップ 3 の画面で、[タイムアウト]、[ポーリング間隔] にそれぞれタイムアウト、ポーリング間隔を分単位で入力します。アイドル状態の接続に対してタイムアウトにならないようにする場合は、いずれもゼロ (0) に設定します。[次へ] をクリックします。



- 10 新規データベース ウィザードのステップ 4 の画面で、新規ユーザー データベースに関連付けるスキーマを、事前定義されたスキーマのリストから選択します。選択できるスキーマには、次のものがあります。
- Blank: スキーマ構築用の最低限の構成ベースです。
 - Common: 最もよく使用されるスキーマ要素を備えたベースレベルの構成です。
 - Defect Tracking: 障害追跡プロセス用の機能を備えたスキーマです。
 - AnalystStudio: ClearQuest を Rational AnalystStudio との組み合わせで使用できるフィールドとルールを備えたスキーマです。
 - DevelopmentStudio: ClearQuest を Rational DevelopmentStudio との組み合わせで使用できるフィールドとルールを備えたスキーマです。
 - TestStudio: ClearQuest を Rational TestStudio との組み合わせで使用できるフィールドとルールを備えたスキーマです。
 - UnifiedChangeManagement: ClearQuest を Rational UCM との組み合わせで使用できるフィールドとルールを備えたスキーマです。
 - Enterprise: ClearQuest を Rational Suite Enterprise との組み合わせで使用できるフィールドとルールを備えたスキーマです。
- 11 [完了] をクリックします。

メモ: 新規データベースに v2003.06.10 より前のバージョンの ClearQuest を実行するクライアントがアクセスする場合、CharacterSetValidation パッケージを適用し、ユーザー データベースに ClearQuest データ コード ページ値を設定してください。

詳細については、146 ページの「パッケージのアップグレード方法」と 35 ページの「ClearQuest のデータ コード ページ値の設定」を参照してください。

接続の管理

スキーマ リポジトリと、そのスキーマ リポジトリに関連付けられるすべてのユーザー データベースを作成した後、ClearQuest メンテナンス ツールを使用してその接続を管理します。

接続はデータベース セットとも呼ばれ、1 つのスキーマ リポジトリと、それに関連付けられるすべてのユーザー データベースで構成されます。

接続を使用すると、チームのメンバーから見える環境の管理もスムーズになります。たとえば、チームのメンバーが ClearQuest クライアントにログオンすると、使用するデータベースを含んだ接続を選択するように要求され、次にデータベース自体を選択するように要求されます (これは、チームが、複数の接続でデータベースを使用することが許可されている場合を仮定しています。チームのメンバーが 1 つの接続でのみ作業する場合は、ログイン時に接続の選択を要求するダイアログ ボックスは表示されません)。

ここでは、次の方法について説明します。

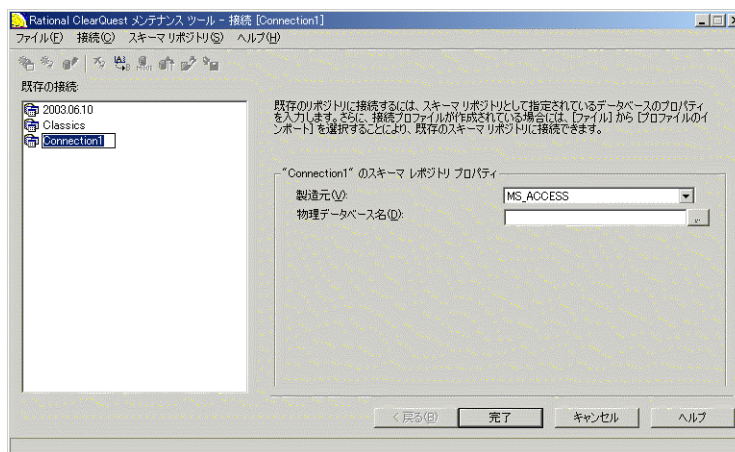
- 新規接続を作成する方法
- 接続の名前を変更する方法
- 接続を編集して基盤となるデータベース管理システムを変更する方法、データベースのプロパティを変更する方法、スキーマ リポジトリの場所を変更する方法
- 接続を削除する方法
- 既存の接続に類似した新規接続を作成するために接続の複製を作成する方法

新規接続の作成

メモ: スキーマ リポジトリの ClearQuest データ コード ページ値が、メンテナンス ツールを実行するコンピュータのオペレーティング システムのコード ページと一致しない場合は、接続を作成できません。ClearQuest データ コード ページ値の詳細については、28 ページの「ClearQuest のデータ コード ページの概要」を参照してください。

新規接続を作成するには

- 1 ClearQuest メンテナンス ツールで、[接続] メニューの [新規作成] をクリックします。



- 2 [既存の接続] 領域で、反転表示された項目の接続に付ける名前を入力します。
- 3 [<接続名> のスキーマ レポジトリ プロパティ] 領域の [製造元] で、スキーマ リポジトリに使用する物理データベースの製造元を選択し、必要なプロパティを入力します。必要なプロパティは、選択するデータベース製造元によって異なります。
- 4 [完了] をクリックします。

接続の変更

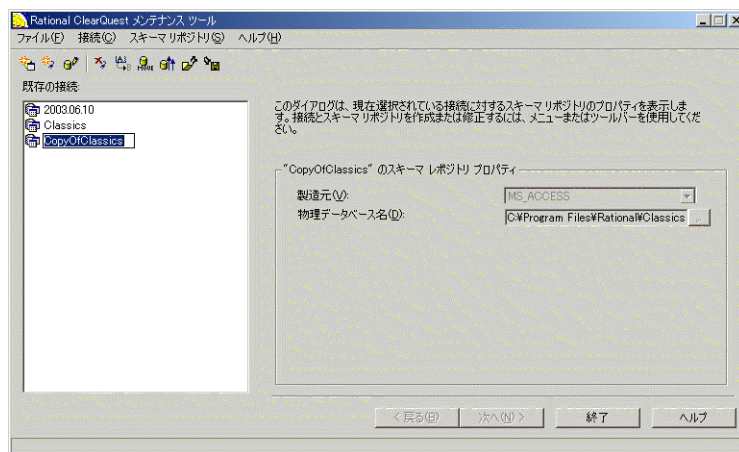
ClearQuest メンテナンス ツールを使用することで、いつでも既存の接続を変更できます。接続の名前の変更、データベース プロパティの編集、接続の複製の作成を行うことができます。

接続の名前の変更

接続の名前を変更することによって、その接続に関連付けられたスキーマまたはユーザー データベース情報が変更されるわけではありません。接続はコンピュータ単位で維持管理されます。スキーマ リポジトリ接続は、コンピュータごとに異なる名前を付けることができます。

ClearQuest メンテナンス ツールを使用して接続の名前を変更するには

- 1 [既存の接続] 領域で、名前を変更する接続を選択します。次に、[接続] メニューの [名前の変更] をクリックします。



- 2 接続の新しい名前を入力し、[Enter] を押します。

接続の編集

接続を編集することで、スキーマ リポジトリを管理するために使用されるデータベース管理システムを変更したり、スキーマ リポジトリのデータベース プロパティを変更できます。

スキーマ リポジトリを新しい位置に移動する場合は、接続を編集して位置の変更を反映させる必要があります。

ClearQuest メンテナンス ツールを使用して接続を編集するには

- 1 [既存の接続] 領域で、編集する接続を選択します。次に、[接続] メニューの [編集] をクリックします。
- 2 [<接続名> のスキーマ レポジトリ プロパティ] 領域の [製造元] で、スキーマ リポジトリに使用する物理データベースの製造元を選択し、必要なプロパティを入力します。必要なプロパティは、選択するデータベース製造元によって異なります。
- 3 [完了] をクリックします。

接続の削除

必要がなくなった接続は削除できます。接続はコンピュータ単位で維持され、接続で指定された ClearQuest のユーザーまたはデータベースには影響を与えません。

ClearQuest メンテナンス ツールを使用して接続を削除するには

- 1 [既存の接続] 領域で、削除する接続を選択します。次に、[接続] メニューの [削除] をクリックします。
- 2 削除するかどうかを確認するメッセージが表示されたら、[はい] をクリックします。

メモ: 削除した接続を復元するには、[接続] メニューの [新規] をクリックするか、既存の接続プロファイルをインポートします。

詳細については、63 ページの「接続プロファイルの用法」を参照してください。

既存の接続の複製

既存の接続に対して複製を作成することができます。たとえば、新規スキーマ リポジトリでもプロパティが既存の接続に類似しているものに接続する必要が発生した場合は、まず既存の接続の複製を作成し、複製した接続のプロパティを編集して、新規スキーマ リポジトリに合わせることができます。

ClearQuest メンテナンス ツールを使用して既存の接続の複製を作成するには

- 1 [既存の接続] 領域で、複製を作成する接続を選択します。次に、[接続] メニューの [複製] をクリックします。
- 2 複製の接続に付ける新しい名前を入力し、[Enter] を押します。

メモ: 複製した接続のプロパティを編集するには、[接続] メニューの [編集] をクリックします。

UNIX クライアントとの接続の維持

UNIX 上の ClearQuest クライアントの接続を作成または更新するには、対象のクライアントから `cqreg add_dbset` コマンドを実行します。

- 以前のリリースで設定したスキーマ リポジトリに接続するには、保存した接続プロファイルをデータベース ディレクトリ `$CQ_HOME/./ClearQuest_Databases/2003.06.10` に移動します。
- 新規の接続を確立するには、[Register Database] ダイアログ ボックスを使用できます。このダイアログ ボックスは ClearQuest を初めて実行したときに表示されます。このダイアログ ボックスは一度しか表示されません。それ以降は、`cqreg add_dbset` コマンドを使用して接続を更新する必要があります。
- 既存の接続を更新するか、新規の接続を確立するには、`cqreg add_dbset` コマンドを実行します。

cqreg のオプション

```
cqreg add_dbset -v-endor db_vendor -s-erver db_server -d-atabase database -u-ser user  
-p-assword password -dbset dbset_name [-co connect_options]
```

オプション	説明
-v-endor	データベースの製造元。
-s-erver	Oracle と SQL Server の場合、データベース サーバー ホストの名前。 DB2 の場合、データベース ノード。
-d-atabase	SQL Server と DB2 の場合、スキーマ リポジトリ データベースの名前。 Oracle の場合、SID。
-u-ser	データベースへのアクセスに使用するユーザー名。
-p-assword	データベースへのアクセスに使用するパスワード。
-dbset	接続の名前。デフォルトの接続名は、ClearQuest のバージョンです (たとえば、2003.06.10)。
-co	(Oracle のみ) 接続オプション。 「HOST=hostname;SID=sid_name;SERVER_VER=8.1」のような値を 指定する場合に使用します。

さまざまな製造元データベースへの接続を追加する例を以下に示します。

Oracle の場合：

```
cqreq add_dbset -v ORACLE -s cqtest1 -d CQ1 -u cq_unix_m -p  
cq_unix_m -dbset CQ_UNIX_TEST -co "SERVER_VER=8.1"
```

SQL Server の場合：

```
cqreg add_dbset -v SQLSERVER -s cqtest2 -d musitea_m -u mila -p mila  
-dbset CQ_DEV
```

DB2 の場合：

```
cqreg add_dbset -v DB2 -s db2tcpnode -d musitea_m -u mila -p mila -dbset  
CQ_DEV
```

接続プロファイルの使用法

接続プロファイルは、スキーマ リポジトリへのアクセス方法に関する情報を含んだテキストファイルです。

接続プロファイルが存在しない場合、ユーザーはログオンするたびに接続とその位置に関する情報を入力する必要があります。接続プロファイルが存在する場合は、ユーザーは必要なすべての情報がある 1 つのファイルをインポートでき、その後はスキーマ リポジトリと関連付けられたデータベースにアクセスできます。これにより、エンド ユーザーが新規スキーマ リポジトリを展開するのにかかる時間が節約できます。

接続プロファイルを使用すると、別の位置に移動されたスキーマ リポジトリと関連付けられたデータベースに再接続することもできます。

ClearQuest の管理者と ClearQuest クライアントのユーザーは、接続プロファイルのインポートとエクスポートを行うことができます。

接続プロファイルの使用には、次の操作があります。

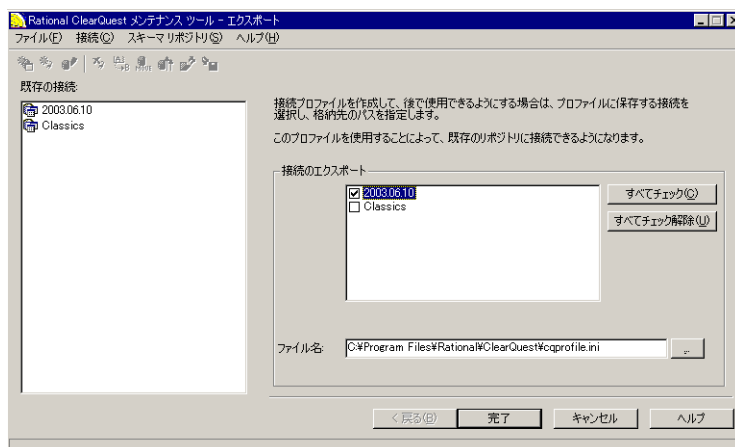
- 63 ページの「接続プロファイルのエクスポート」
- 64 ページの「接続プロファイルのインポート」

接続プロファイルのエクスポート

接続プロファイルを作成するには、1 つの接続または複数の接続のスキーマ リポジトリ プロパティを含む情報をテキスト ファイルにエクスポートします。このファイルは後で、ユーザーが接続にアクセスするためにインポートできます。

接続プロファイルのエクスポートするには

- 1 ClearQuest メンテナンス ツールで、[ファイル] メニューの [プロファイルのエクスポート] をクリックします。



- 2 [接続のエクスポート] 領域で、接続プロファイルでエクスポートする接続のチェックボックスをオンにします。

自分の ClearQuest システムで使用可能なすべての接続が一覧表示されます。任意の個数を選択し、1 つの接続プロファイルにすべての情報をエクスポート用に格納できます。

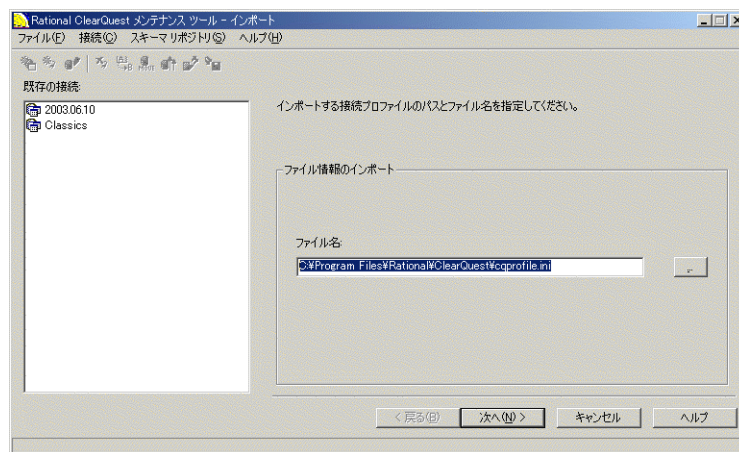
- 3 デフォルトでは、接続プロファイルは **cqprofile.ini** というファイルの中にある ClearQuest のインストールディレクトリに保存されます。必要に応じて、新しい名前と位置を入力できます。
- 4 [完了] をクリックします。

接続プロファイルのインポート

メモ: スキーマリポジトリの ClearQuest データコードページ値が、メンテナンスツールを実行するコンピュータのオペレーティングシステムのコードページと一致しない場合は、接続プロファイルをインポートできません。

接続プロファイルをインポートして使用するには

- 1 ClearQuest メンテナンス ツールで、[ファイル] メニューの [プロファイルのインポート] をクリックします。



- 2 [ファイル名] で、インポートするプロファイルのパスとファイル名を入力します。
複数の接続によってプロファイルが構成される場合は、各接続の名前が一覧表示され、それに対応するスキーマリポジトリプロパティが表示されます。
- 3 接続の名前を変更する場合は、接続を選択し、[名前の変更] をクリックします。反転表示された領域に新しい名前を入力します。
- 4 プロファイルからインポートする接続を選択し、[完了] をクリックします。

スキーマ リポジトリの移動

スキーマ リポジトリを新しい位置または異なる製造元のデータベース管理システムに移動すると、ClearQuest により、スキーマ リポジトリが新しい位置にコピーされ、古いスキーマ リポジトリがロックされ、新しい位置にコピーしたリポジトリが、作業対象のスキーマ リポジトリとしてマークされます。

メモ: ClearQuest データベースの移動と操作には、ClearQuest ツールのみを使用します。

ClearQuest は、スキーマ リポジトリの位置とユーザー データベースの位置をスキーマ リポジトリに格納します。製造元データベースのツールを使用してスキーマ リポジトリの物理的な位置を移動すると、ClearQuest からスキーマ リポジトリへの接続が失われます。このような場合は、接続のスキーマ リポジトリ プロパティを更新し、編集する必要があります。

スキーマ リポジトリ移動の準備

スキーマ リポジトリを新しい位置または別のデータベース管理システムに移動するには

- 1 データベース製造元のツールを使用して、新しい位置に空の新規データベースを作成します (ClearQuest メンテナンス ツールの内部から Microsoft Access データベースと Anywhere データベースを作成できます)。製造元データベースの設定方法については、『Rational Software サーバー製品インストールガイド (Windows 版)』を参照してください。
- 2 ClearQuest からログオフするようにすべてのユーザーに通知します。ClearQuest は、この処理中にユーザーが新しくデータベースにアクセスすることを禁止します。しかし、準備手順を開始した時点でログインしているユーザーを検出してログオフさせることはできません。

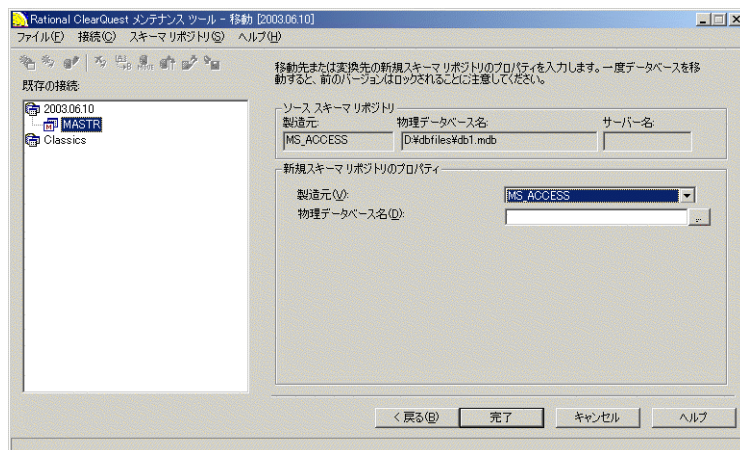
メモ: ハイエンド データベース製品の中には、データベースからユーザーを強制的にログオフさせるツールを備えている製品もあります。データベースの製造元がこのようなツールを提供しているかどうかを確認してください。

- 3 ClearQuest のすべてのデータベースをバックアップします (これは、スキーマ リポジトリのみを移動するか、1 つのユーザー データベースのみを移動するかは問いません)。これは、ネットワークの障害など、移動中に発生し得る予測できない状況のためにデータが消失することを防止します。

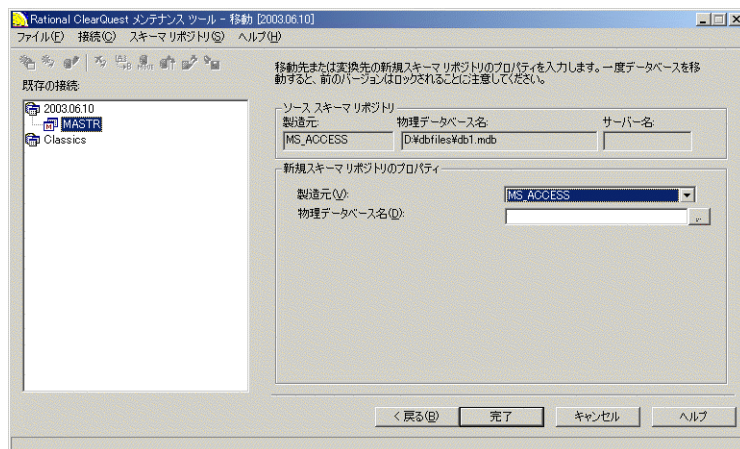
スキーマ リポジトリの移動

ClearQuest メンテナンスツールを使用してスキーマ リポジトリを移動するには

- 1 [既存の接続] 領域で、移動するスキーマ リポジトリを選択します。次に、[スキーマ リポジトリ] メニューの [移動] をクリックします。



- 2 [ログオン名] と [パスワード] に入力し、[次へ] をクリックします。移動処理中はデータベースがロックされることに注意してください。
- 3 [新規スキーマ リポジトリのプロパティ] に、移動するプロパティを入力します。[製造元] でデータベース製造元を選択し、必要な製造元データベース プロパティを入力します。必要なプロパティは、選択するデータベース製造元によって異なります。詳細については、76 ページの「製造元データベースのパラメータ」を参照してください。



- 4 [完了]をクリックします。
- 5 更新したスキーマ リポジトリ プロパティを含む情報をエクスポートすることによって、新規の接続プロファイルを作成します。ClearQuest メンテナンス ツールを使用してプロファイル情報をエクスポートするには、[ファイル]メニューの[プロファイルのエクスポート]をクリックします。
- 6 新規スキーマ リポジトリに接続するために、すべての ClearQuest ユーザーに ClearQuest メンテナンス ツールで新規の接続プロファイルをインポートさせます。ユーザーが接続するために必要な情報を提供してください。ClearQuest メンテナンス ツールを使用してプロファイルをインポートするには、[ファイル]メニューの[プロファイルのインポート]をクリックします。

詳細については、59 ページの「新規接続の作成」を参照してください。

製造元データベースのツールを使用したスキーマ リポジトリの移動

ClearQuest メンテナンス ツールの代わりに製造元データベースのツールを使用してスキーマ リポジトリを移動することもできます。ただし、ClearQuest メンテナンス ツールを使用して、スキーマ リポジトリの新規の物理的な位置に関する情報で接続を更新する必要があります。このタスクを実行するには、スーパー ユーザー権限が必要です。

既存の接続の更新

既存の接続があるスキーマ リポジトリを、製造元データベースのツールを使用して移動する場合は、ClearQuest メンテナンス ツールを使用して次の操作を行って接続を更新します。

- 1 [既存の接続]領域で、移動または更新する接続を選択します。次に、[スキーマ リポジトリ]メニューの[更新]をポイントし、[選択した接続]をクリックします。
- 2 [<接続名>のスキーマ レポジトリ プロパティ]領域の[製造元]で、スキーマ リポジトリに使用する物理データベースの製造元を選択し、必要なプロパティ(スキーマ リポジトリの新しい位置も含む)を入力します。必要なプロパティは、選択するデータベース製造元によって異なります。詳細については、76 ページの「製造元データベースのパラメータ」を参照してください。
- 3 [次へ]をクリックします。更新が完了したら、[完了]をクリックします。
- 4 更新したスキーマ リポジトリ プロパティを含む情報をエクスポートすることで、新規の接続プロファイルを作成します。ClearQuest メンテナンス ツールを使用してプロファイル情報をエクスポートするには、[ファイル]メニューの[プロファイルのエクスポート]をクリックします。

- 5 新規スキーマ リポジトリに接続するために、すべての ClearQuest ユーザーに ClearQuest メンテナンス ツールで新規の接続プロファイルをインポートさせます。ユーザーが接続するために必要な情報を提供してください。ClearQuest メンテナンス ツールを使用してプロファイル情報をインポートするには、[ファイル] メニューの [プロファイルのインポート] をクリックします。

詳細については、59 ページの「新規接続の作成」を参照してください。

接続を使用しないスキーマ リポジトリの更新

ClearQuest システムに、2001.04.00 より前のバージョンの ClearQuest で作成された既存のスキーマ リポジトリがある場合は、接続でスキーマ リポジトリにアクセスすることができません。ただし、ほかのデータベース製造元のツールを使用してスキーマ リポジトリを移動する必要がある場合は、ClearQuest メンテナンス ツールを使用することにより、スキーマ リポジトリを接続に関連付けなくともプロパティを更新することができます。

スキーマ リポジトリの物理的な位置に関する情報を更新するには

- 1 [スキーマ リポジトリ] メニューの [更新] をクリックし、[その他] をクリックします。
- 2 [<接続名> のスキーマ レポジトリ プロパティ] 領域の [製造元] で、スキーマ リポジトリに使用する物理データベースの製造元を選択し、必要なプロパティ (スキーマ リポジトリの新しい位置も含む) を入力します。必要なプロパティは、選択するデータベース製造元によって異なります。詳細については、76 ページの「製造元データベースのパラメータ」を参照してください。
- 3 [次へ] をクリックします。更新が完了したら、[完了] をクリックします。
- 4 ClearQuest の適用可能なバージョンでユーザーがスキーマ リポジトリに接続するために必要な情報を提供してください。

データベースを ClearQuest の新しい機能レベルにアップグレードする

ClearQuest メンテナンス ツールを使用して、スキーマ リポジトリとユーザー データベースを ClearQuest の新しい機能レベルにアップグレードすることができます。機能レベルは、スキーマ リポジトリとユーザー データベースがサポートする機能を定義します。

ただし、ClearQuest メンテナンス ツールを使用してスキーマ リポジトリまたはユーザー データベースを ClearQuest の新しい機能レベルにアップグレードする前に、Windows の場合は、『Upgrade Guide for Rational Suite for Windows』を参照してください。ここでは、このマニュアルにある詳細なガイドラインを既に読んだことを前提に説明します。

SQL Anywhere 5.0 から SQL Anywhere 8.0 へのアップグレード

SQL Anywhere 5.5 を使用して作成されたデータベースは、SQL Anywhere 8.0 を使用する ClearQuest の新しいバージョンへのアップグレード直後にアップグレードする必要はありません。ClearQuest では、SQL Anywhere 5.5 と SQL Anywhere 8.0 で作成されたデータベースが共存できます。ただし、すべての新規データベースは SQL Anywhere 8.0 で作成し、徐々に古いデータベースを 5.5 から 8.0 に移行する必要があります。

既存の接続のアップグレード

既存の接続に対してスキーマ リポジトリと関連付けられたデータベースをアップグレードするには、ClearQuest メンテナンス ツールを使用して次の操作を行います。

- 1 [既存の接続] 領域で、アップグレードする接続を選択します。次に、[スキーマ リポジトリ] メニューの [アップグレード] をクリックし、[選択した接続] をクリックします。
- 2 アップグレードするスキーマ リポジトリの [<接続名>のスキーマ レポジトリ プロパティ] を確認します。
- 3 [次へ] をクリックします。
- 4 スキーマ リポジトリに関連付けられているユーザー データベースをアップグレードするかどうかを確認する画面が表示されたら、[はい] または [いいえ] をクリックします。アップグレードプロセス中はユーザー データベースがロックされることに注意してください。
- 5 [新規スキーマ リポジトリのプロパティ] 領域で、アップグレードしたスキーマ リポジトリを格納する新規の製造元データベースのプロパティを入力し、[次へ] をクリックします。
- 6 ユーザー データベースをアップグレードするように選択した場合は、[新規ユーザー データベース] 領域で、アップグレードしたユーザー データベースを格納する新規の製造元データベースのプロパティを入力します。
- 7 複数のユーザー データベースをアップグレードする場合は、[次へ] をクリックし、手順 6 を繰り返します。そうでない場合は、[完了] をクリックします。
- 8 更新したスキーマ リポジトリ プロパティを含む情報をエクスポートすることによって、新規の接続プロファイルを作成します。ClearQuest メンテナンス ツールを使用してプロファイル情報をエクスポートするには、[ファイル] メニューの [プロファイルのエクスポート] をクリックします。
- 9 新規スキーマ リポジトリに接続するために、すべての ClearQuest ユーザーに ClearQuest メンテナンス ツールで新規の接続プロファイルをインポートさせます。ユーザーが接続するために必要な情報を提供してください。ClearQuest メンテナンス ツールを使用してプロファイル情報をインポートするには、[ファイル] メニューの [プロファイルのインポート] をクリックします。

詳細については、59 ページの「新規接続の作成」を参照してください。

接続を使用しないスキーマ リポジトリとユーザー データベースのアップグレード

ClearQuest システムに、2001.04.00 より前のバージョンの ClearQuest で作成された既存のスキーマ リポジトリがある場合は、接続でスキーマ リポジトリにアクセスすることができません。ただし、ClearQuest メンテナンス ツールを使用することにより、スキーマ リポジトリを接続に関連付けなくてもプロパティと関連付けられたユーザー データベースをアップグレードすることができます。

- 1 ClearQuest メンテナンス ツールで、[スキーマ リポジトリ] メニューの [アップグレード] をポイントし、[その他] をクリックします。
- 2 [<接続名> のスキーマ レポジトリ プロパティ] 領域で、アップグレードするスキーマ リポジトリのプロパティを入力します。
- 3 [次へ] をクリックします。
- 4 スキーマ リポジトリに関連付けられているユーザー データベースをアップグレードするかどうかを確認する画面が表示されたら、[はい] または [いいえ] をクリックします。アップグレードプロセス中はユーザー データベースがロックされることに注意してください。
- 5 [ClearQuest ログイン] 領域の [ログオン名] と [パスワード] に入力し、[次へ] をクリックします。
- 6 ユーザー データベースをアップグレードするように選択した場合は、アップグレードするユーザー データベースを選択し、[次へ] をクリックします。
- 7 [新規スキーマ リポジトリのプロパティ] 領域で、アップグレードしたスキーマ リポジトリを格納する新規の製造元データベースのプロパティを入力し、[次へ] をクリックします。
- 8 ユーザー データベースをアップグレードするように選択した場合は、[新規ユーザー データベース] 領域で、アップグレードしたユーザー データベースを格納する新規の製造元データベースのプロパティを入力します。
- 9 複数のユーザー データベースをアップグレードする場合は、[次へ] をクリックし、手順 8 を繰り返します。そうでない場合は、[完了] をクリックします。
- 10 更新したスキーマ リポジトリ プロパティを含む情報をエクスポートすることによって、新規の接続プロファイルを作成します。ClearQuest メンテナンス ツールを使用してプロファイル情報をエクスポートするには、[ファイル] メニューの [プロファイルのエクスポート] をクリックします。
- 11 新規スキーマ リポジトリに接続するために、すべての ClearQuest ユーザーに ClearQuest メンテナンス ツールで新規の接続プロファイルをインポートさせます。ユーザーが接続するために必要な情報を提供してください。ClearQuest メンテナンス ツールを使用してプロファイル情報をインポートするには、[ファイル] メニューの [プロファイルのインポート] をクリックします。

詳細については、59 ページの「新規接続の作成」を参照してください。

ユーザー データベースの移動と削除

ClearQuest Designer を使用して、ClearQuest ユーザー データベースを新しい位置または別のデータベース管理システムに移動できます。

ユーザー データベース移動の準備

ユーザー データベースを新しい位置または異なる製造元のデータベース管理システムに移動すると、ClearQuest により、ユーザー データベースが新しい位置にコピーされ、古いユーザー データベースがロックされ、新しい位置にコピーしたユーザー データベースが、作業対象のユーザー データベースとしてマークされます。

ユーザー データベースを新しい位置または別の製造元のデータベース管理システムに移動するには

- 1 データベース製造元のツールを使用して、新しい位置に空の新規データベースを作成します (ClearQuest Designer の内部から Microsoft Access データベースと SQL Anywhere データベースを作成できます)。新規データベースの作成方法については、『Rational Software サーバー製品インストレーション ガイド』を参照してください。
- 2 ClearQuest からログオフするようにすべてのユーザーに通知します。ClearQuest は、この処理中にユーザーが新しくデータベースにアクセスすることを禁止します。しかし、準備手順を開始した時点でログインしているユーザーを検出してログオフさせることはできません。

メモ: ハイエンド データベース製品の中には、データベースからユーザーを強制的にログオフさせるツールを備えている製品もあります。データベースの製造元がこのようなツールを提供しているかどうかを確認してください。

- 3 すべてのデータベースをバックアップします (これは、スキーマ リポジトリのみを移動するか、1 つのユーザー データベースのみを移動するかは問いません)。

メモ: ClearQuest のすべてのデータベース (スキーマ リポジトリと関連付けられたすべてのユーザー データベース) をバックアップする必要があります。これは、ネットワークの障害など、移動中に発生する可能性がある、予想できない状況が原因でデータが消失することを防止します。

ユーザー データベースの移動

ClearQuest Designer を使用して、ユーザー データベースを新しい位置または別の製造元のデータベース管理システムに移動できます。

ここでは、ClearQuest Designer を使用してユーザー データベース SAMPLE を別の製造元のデータベースに移動する例について説明します。ユーザー データベースを新しい位置に移動する場合には、同じ手順を使用できます。

メモ: 既にデータベースの移動を完了していて、データベース プロパティだけを変更する必要がある場合には、72 ページの「ユーザー データベースのプロパティの更新」を参照してください。

ユーザー データベースを移動するには

- 1 71 ページの「ユーザー データベース移動の準備」の手順に従います。
- 2 Windows の [スタート] メニューから、[プログラム]、[Rational Software]、[Rational ClearQuest] の順にポイントし、[ClearQuest Designer] をクリックして、ClearQuest Designer を起動します。
- 3 [スキーマを開く] ダイアログ ボックスが表示されたら、[キャンセル] をクリックします。
- 4 [データベース] メニューの [ユーザー データベースの移動] をクリックします。
- 5 [ユーザー データベースの移動] ダイアログ ボックスの [論理データベース名] で論理データベース名を選択し、[プロパティ] をクリックします。

メモ: [OK] をクリックすると、[コメント] テキスト ボックスでの変更だけが更新され、データベースは移動されません。

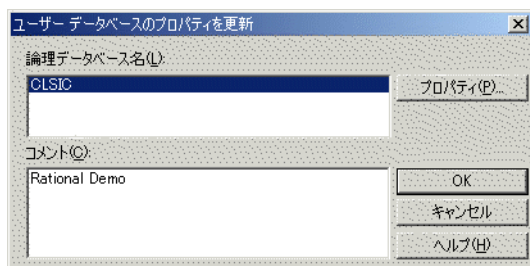
- 6 [ユーザー データベースの移動] ダイアログ ボックスで、[製造元] で別のデータベース製造元を選択し、新規データベース プロパティ (実動データベースかテスト データベースかなど) を入力します。
- 7 指定が完了したら、[移動] をクリックします。[データベースを移動しています] ダイアログ ボックスに、データベースの移動のステータスが表示されます。

ステータス メッセージを出力ファイルに保存する場合は、[作業状態に名前を付けて保存] をクリックし、出力ファイルのファイル名とパスを指定します。[保存] をクリックし、手順 8 に進みます。
- 8 [OK] をクリックして、[データベースを移動しています] ダイアログ ボックスを終了します。次に、データベース更新メッセージの [OK] をクリックします。
- 9 [OK] をクリックして、[ユーザー データベースの移動] ダイアログ ボックスを終了します。

ユーザー データベースのプロパティの更新

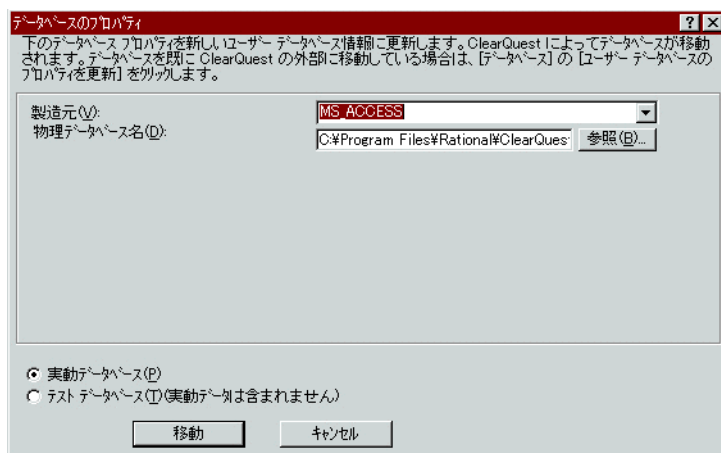
ここでは、ユーザー データベースを ClearQuest の外部に移動した場合に、[ユーザー データベースのプロパティを更新] を使用して、選択したデータベースのプロパティを変更する方法について説明します。データベースを ClearQuest の外部に移動していない場合は、71 ページの「ユーザー データベース移動の準備」で説明する手順に従って、データベースを移動する前にプロパティを変更します。

- 1 Windows の [スタート] メニューから、[プログラム]、[Rational Software]、[Rational ClearQuest] の順にポイントし、[ClearQuest Designer] をクリックして、ClearQuest Designer を起動します。
- 2 [スキーマを開く] ダイアログ ボックスが表示されたら、[キャンセル] をクリックします。
- 3 [データベース] メニューの [ユーザー データベースのプロパティを更新] をクリックします。
- 4 [ユーザー データベースのプロパティを更新] ダイアログ ボックスの [論理データベース名] で、論理データベース名を選択します。



[コメント] ボックスのみを変更する場合は、コメントを入力し、[OK] をクリックします。

- 5 データベース プロパティを変更するには、[プロパティ] をクリックします。
- 6 [データベースのプロパティ] ダイアログ ボックスで、必要に応じてプロパティを編集します。詳細については、76 ページの「製造元データベースのパラメータ」を参照してください。



- 7 操作が終了したら、[更新] をクリックして変更を保存します。

ユーザー データベースの削除

ClearQuest からユーザー データベースを削除すると、スキーマ リポジトリとユーザー データベースの間のリンクが削除されます。物理データベース自体は削除されず、データベースに関連付けられたスキーマも削除されません。

後日、データベースとスキーマ リポジトリの間のリンクを復元すると、削除したデータベースを復元できます。

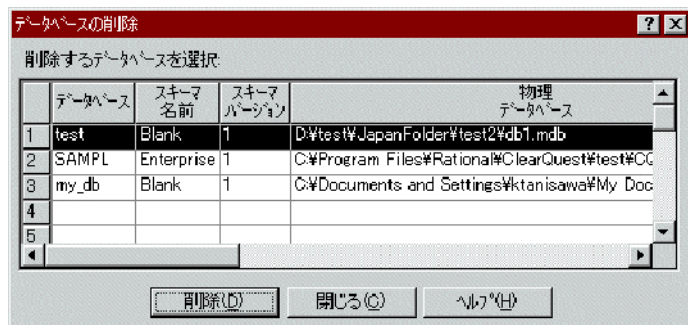
詳細については、75 ページの「削除されたデータベースの復元」を参照してください。

メモ: データベースに関連付けられたスキーマ バージョンを削除するか、ClearQuest を新しいバージョンにアップグレードすると、削除したデータベースへのリンクは完全に失われます。

ClearQuest では、スキーマ リポジトリは削除できません。

ClearQuest からユーザー データベースを削除するには

- 1 Windows の [スタート] メニューから、[プログラム]、[Rational Software]、[Rational ClearQuest] の順にポイントし、[ClearQuest Designer] をクリックして、ClearQuest Designer を起動します。
- 2 [スキーマを開く] ダイアログ ボックスが表示されたら、[キャンセル] をクリックします。
- 3 [データベース] メニューの [データベースの削除] をクリックします。



- 4 [データベースの削除] ダイアログ ボックスで、削除するデータベースを選択し、[削除] をクリックします。

ClearQuest からユーザー データベースを削除した後は (データベースとスキーマ リポジトリの間のリンクを削除した後は)、ユーザー データベースを物理的に削除できます。ただし、これを行うと、削除されたデータベースは復元できなくなります。Oracle データベースまたは SQL Server データベースを物理的に削除するには、データベース製造元のツールを使用します。Microsoft Access データベースと SQL Anywhere データベースの場合は、データベース ファイルを手動で削除できます。

削除されたデータベースの復元

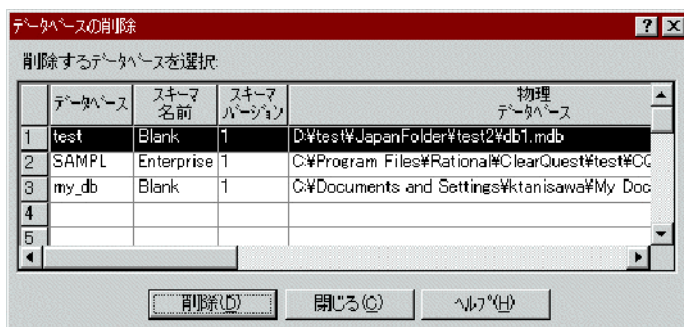
メモ: 物理データベースがない場合は、削除されたデータベースを復元できません。データベース製造元のツールを使用して物理データベースを削除した場合には、削除されたデータベースを復元することはできません。

削除されたユーザー データベースを復元するときは、ユーザー データベースとスキーマ リポジトリの間のリンクを復元します。ユーザー データベースが以前にリンクされていたスキーマ リポジトリと同じスキーマ バージョンへのリンクを復元する必要があります。物理データベースは、リンクが削除されたときと同じ場所にある必要があります。

ユーザー データベースからスキーマ リポジトリへのリンクが完全に失われるのを避けるため、削除されたデータベースの復元は、データベース削除後のできる限り早い時点で実行してください。データベースに関連付けられたスキーマ バージョンを削除した場合、または ClearQuest を新しいバージョンにアップグレードした場合には、リンクを復元することはできません。

削除されたユーザー データベースを復元するには

- 1 Windows の [スタート] メニューから、[プログラム]、[Rational Software]、[Rational ClearQuest] の順にポイントし、[ClearQuest Designer] をクリックして、ClearQuest Designer を起動します。
- 2 [スキーマを開く] ダイアログ ボックスが表示されたら、[キャンセル] をクリックします。
- 3 [データベース] メニューの [データベース削除の取り消し] をクリックし、[データベース削除の取り消し] ダイアログ ボックスを表示します。



- 4 リストから削除されたデータベースを選択し、[削除取り消し] をクリックします。

データベース プロパティの表示

ユーザー データベースのプロパティを表示するには

- 1 Windows の [スタート] メニューから、[プログラム]、[Rational Software]、[Rational ClearQuest] の順にポイントし、[ClearQuest Designer] をクリックして、ClearQuest Designer を起動します。
- 2 [スキーマを開く] ダイアログ ボックスが表示されたら、[キャンセル] をクリックします。

- 3 [データベース] メニューの [データベースのプロパティの表示] をクリックします。
- 4 [データベースのプロパティ] ダイアログ ボックスで、目的のデータベースを選択し、[プロパティ] をクリックします。

[データベースのプロパティ] ダイアログ ボックスは読み取り専用です。データベース プロパティを移動または編集するには、71 ページの「ユーザー データベースの移動と削除」または 72 ページの「ユーザー データベースのプロパティの更新」を参照してください。
- 5 [OK] をクリックし、[データベースのプロパティ] ダイアログ ボックスを終了します。

製造元データベースのパラメータ

ClearQuest のスキーマ リポジトリまたはユーザー データベースを作成するとき、ClearQuest で使用するために作成した製造元データベースのパラメータを入力するよう要求されます。

データベース パラメータを指定する際に、製造元データベースによって大文字/小文字の区別は異なるという点に注意してください。特に、データベースでの大文字/小文字の区別の設定は、次の 2 つのシナリオで一貫している必要があります。

- ある製造元のデータベースから別の製造元のデータベースに移行する場合。
- ClearQuest MultiSite を使用し、異なる製造元のデータベース製品で作成した複数のデータベースを同期させる場合。

データベース 製造元	デフォルト設定	可能な設定	コメント
Microsoft Access	大文字/小文字を区別しない	大文字/小文字を区別しない	Microsoft Access データベースでは大文字/小文字は区別されません。
SQL Anywhere	大文字/小文字を区別しない	大文字/小文字を区別しない、大文字/小文字を区別する	SQL Anywhere で作成されたデータベースでは大文字/小文字は区別されません。ただし、作成後に大文字/小文字を区別するように設定を変更できます。
DB2	大文字/小文字を区別する	大文字/小文字を区別する	データの大文字/小文字のみ区別され、データベース オブジェクト名の大文字/小文字は区別されません。

データベース 製造元	デフォルト設定	可能な設定	コメント
SQL Server	大文字/小文字を 区別しない、 大文字/小文字を 区別する	大文字/小文字を 区別しない、大文字/ 小文字を区別する	大文字/小文字の区別は、インス トール時またはデータベースの 作成時にオプションで指定でき ます。 メモ: データだけでなくデータ ベース オブジェクト名の大文字/ 小文字も区別されます。
Oracle	大文字/小文字を 区別する	大文字/小文字を 区別する	データの ¹ 大文字/小文字のみ区別 され、データベース オブジェクト 名の大文字/小文字は区別されま せん。

Microsoft Access

メモ: ClearQuest Web と ClearQuest MultiSite では、Microsoft Access データベースはサポート
されていません。

- [物理データベース名] ボックスで、スキーマ リポジトリの名前を入力します。また、
UNC スタイルのパスを使用して、スキーマ リポジトリ ファイルへの完全なパスを入力し
ます。例を次に示します。

¥¥DevServer¥ProjectShare¥CQ_DBS¥schema_repo.mdb

データベースを格納するディレクトリを参照して指定することもできます。UNC スタイル
のパス名になるように、Windows の [ネットワーク コンピュータ] でパス名を確認してく
ださい。

SQL Anywhere

- [物理データベース名] ボックスで、スキーマ リポジトリの名前を入力します。また、
UNC スタイルのパスを使用して、スキーマ リポジトリ ファイルへの完全なパスを入力し
ます。例を次に示します。

¥¥DevServer¥ProjectShare¥CQ_DBS¥schema_repo.mdb

UNC スタイルのパス名になるように、Windows の [ネットワーク コンピュータ] で、
データベースを格納しているディレクトリを参照して指定することもできます。

- [データベース サーバー名] ボックスにデータベース サーバーの名前を入力します。
- SQL Anywhere サーバーとの通信に使用するプロトコルを選択します。

- [ホスト名] ボックスにホスト名を入力します。ホスト名は必須です。ネットワーク共有や1つまたは複数の通信プロトコルを使用して、データベース ホスト コンピュータをクライアント コンピュータで認識できるようにする必要があります。TCP/IP の場合、IP アドレスを入力する場合もあります。
- [接続オプション] ボックスに、使用している SQL Anywhere のバージョンを入力します。SQL Anywhere 5.0 の場合は、「Server_Ver=5.0」と入力します。SQL Anywhere 8.0 の場合は、「Server_Ver=8.0」と入力します。このフィールドが空白のままの場合、ClearQuest メンテナンス ツールのデフォルトでは SQL Anywhere 5.0 に設定されます。

メモ: ClearQuest バージョン 2003.06.10 では、SQL Anywhere バージョン 8.0 でのみ新規スキーマ リポジトリを作成できます。

SQL Server

- [物理データベース名] ボックスで、SQL Server スキーマ リポジトリのデータベースの名前を入力します。
- [データベース サーバー名] ボックスにデータベース サーバーの名前を入力します。
メモ: 各 ClearQuest ネイティブ クライアントが、このエントリに表示されているとおりのサーバー ホスト名にアクセスできるようにする必要があります。
- [管理者名] ボックスに、データベース所有者 (DBO または管理者) のユーザー名を入力します。
- [管理者パスワード] ボックスに、データベース所有者 (DBO または管理者) のパスワードを入力します。
- [読み取り/書き込みユーザー名] ボックスに、一般ユーザーまたは読み取り/書き込み権限を持つユーザーの名前を入力します。
- [読み取り/書き込みユーザー パスワード] ボックスに、一般ユーザーまたは読み取り/書き込み権限を持つユーザーのパスワードを入力します。
- [読み取り専用ユーザー名] ボックスに、スキーマ リポジトリ用または読み取り専用のユーザー名を入力します。
- [読み取り専用ユーザー パスワード] ボックスに、スキーマ リポジトリ用または読み取り専用のユーザー パスワードを入力します。

Oracle

- [SQL*Net エイリアス] ボックスに TNS 名 (Oracle インスタンスを指す SQL*Net エイリアス) を入力します。
- [ユーザー名] ボックスに、スキーマ リポジトリ用に作成した Oracle ユーザー名を入力します。
- [パスワード] ボックスに、ユーザー名のパスワードを入力します。
- [接続オプション] ボックスに次のオプションをセミコロンで区切って入力します。
例を次に示します。

Host=name; SID=name; Server_ver=8.1; Client_ver=8.1;
Lob_type=LONG or CLOB

オプション	定義
HOST	データベース サーバー ホストの名前です。ClearQuest 環境に Unix と Windows の両方のクライアントが含まれている場合、この名前は必須です。
SID	データベース インスタンスの名前です。ClearQuest 環境に UNIX と Windows の両方のクライアントが含まれている場合、この名前は必須です。 UNIX では、SID の大文字と小文字は区別されます。
SERVER_VER	使用している Oracle サーバーのバージョンです。Oracle 8i の場合は 8.1 と指定します。
CLIENT_VER	使用している Oracle クライアント ソフトウェアのバージョンです。Oracle 8i の場合は 8.1 と指定します。 ClearQuest クライアント ユーザーが使用する Oracle クライアントのバージョンと、管理者がここに入力するバージョンが異なる場合は、ユーザーが手動で設定をオーバーライドする必要があります。 デフォルトの設定値は 8.1 です。
LOB_TYPE	使用するデータ型を設定します。有効なデータ型は、LONG または CLOB です。 既存のデータベースがデフォルトで LONG の場合は、CLOB に変更しないでください。LONG という LOB_TYPE プロパティで既に実装されたデータベースは CLOB と互換性がありません。 ただし、CLOB データ タイプの使用には制限があります。制限事項とガイドラインの詳細については、『Rational ClearQuest 管理ガイド』を参照してください。

メモ : すべての ClearQuest ユーザーが同じエイリアスまたは Net Service Name (TNS 名)、データベース インスタンス (システム ID または SID)、TCP/IP プロトコル、ホスト名 (Oracle サーバー ホスト) を使用する必要があります。

DB2

- [データベースの別名] ボックスに、スキーマ リポジトリ用のデータベースの別名 (DB2 データベースを指す別名) を入力します。
- [ユーザー名] ボックスに、スキーマ リポジトリ用に作成した DB2 ユーザー名を入力します。
- [パスワード] ボックスに、ユーザー名のパスワードを入力します。

メモ: DB2 では、各 ClearQuest データベースが一意の DB2 ログイン名を使用することを前提に、ClearQuest から単一で物理的なデータベース サーバーを指定できます。各 DB2 クライアントに対して別名を作成する必要はありません。

ClearQuest スキーマの 操作

4

Rational ClearQuest のスキーマは、1 つのタイプの変更依頼についてプロセス モデルを完全に記述したもの (そのプロセス モデルのメタデータ) です。スキーマには、モデルの状態とアクション、データ構造、フック コード、フォーム、レポート、クエリーなどが記述されています。

あるタイプの変更依頼の処理方法を修正するには、スキーマをチェックアウト、テスト、チェックインする方法と、データベースをアップグレードして修正を反映する方法を理解する必要があります。

この章の内容は、次のとおりです。

- 82 ページの「スキーマの変更手順」
- 82 ページの「テスト データベースの作成」
- 84 ページの「スキーマのチェックアウト」
- 86 ページの「Blank スキーマからの新規スキーマの作成」
- 87 ページの「スクリプト言語の選択」
- 88 ページの「スキーマの変更の確認」
- 89 ページの「スキーマ用テスト データベースの設定」
- 91 ページの「スキーマのチェックイン」
- 92 ページの「ユーザー データベースのアップグレード」
- 93 ページの「進行中の作業の保存」
- 94 ページの「スキーマまたはスキーマ バージョンの削除」
- 94 ページの「別のスキーマへのデータベースの変更」

ClearQuest Designer を使用してスキーマの各要素 (プロセス モデル、データ構造、フック コードなど) を修正する方法と、パッケージ (スキーマに複雑な機能を追加するスキーマ コンポーネント バンドル) を適用してスキーマを変更する方法については、後に示します。

メモ: スキーマを操作するには、スキーマ デザイナ権限またはスーパー ユーザー権限が必要です。ユーザー権限の設定方法については、「第 8 章 ユーザーの管理」を参照してください。

ClearQuest には、一般的なワークフロー モデルを提供するスキーマがいくつか事前定義されています。事前定義のスキーマのリストは、287 ページの「ClearQuest のスキーマとパッケージ」を参照してください。

スキーマの変更手順

スキーマを変更して修正を有効にしたり、新規スキーマを作成するには、ClearQuest Designer を使用して次の手順に従います。

- 1 テスト データベースを作成し、カスタマイズするスキーマと関連付けます。
- 2 カスタマイズするスキーマをスキーマ リポジトリからチェックアウトするか、新規スキーマを作成します。
- 3 フック コードに使用するスクリプト言語を選択します。
- 4 ClearQuest Designer を使用してスキーマの要素を修正するか、パッケージを適用して、スキーマを変更します。
- 5 ClearQuest Designer で、[確認] コマンドを使用して、カスタマイズしたスキーマを確認します。
- 6 テスト データベースをこのスキーマ用に設定します (この手順の初めに作成したテスト データベースを、チェックアウトしたスキーマのテストで使用するよう指定します)。
- 7 テスト データベースを使用して、カスタマイズしたスキーマをテストします。
- 8 進行中の作業を保存します。
- 9 カスタマイズしたスキーマを、スキーマにチェックインします。
- 10 変更したスキーマを使用してユーザー データベースをアップグレードします。

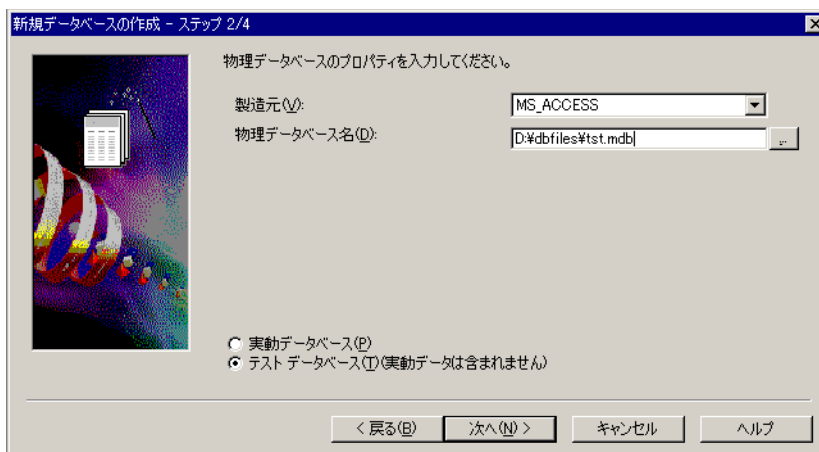
テスト データベースの作成

テスト データベースは、一時的なデータベースです。新規スキーマまたは既存のスキーマの新しいバージョンを作るときに作成します。テスト データベースを使用すると、新規スキーマが予想どおりに動作することを確認できます。これにより、ユーザー データベースにある実動データに損傷を与えないで済みます。

テスト データベースの作成手順は、ユーザー データベースの作成手順と同じです。

テスト データベースを作成するには

- 1 ClearQuest Designer を起動していない場合は、ClearQuest Designer を起動します。[スキーマを開く] ダイアログ ボックスが表示されたら、[キャンセル] をクリックします。既に ClearQuest Designer を起動している場合は、手順 2 に進みます。
- 2 ClearQuest Designer で、[データベース] メニューの [新規データベースの作成] をクリックします。
- 3 新規データベース ウィザードの指示に従って、データベースを作成します。データベース名は、半角で 1 ～ 5 文字の長さにし、テスト データベースであることがわかるような名前にします。
- 4 ステップ 2/4 で、画面の下部にある [テスト データベース] をオンにします。



データベース プロパティの設定の詳細については、76 ページの「製造元データベースのパラメータ」を参照してください。

- 5 ステップ 4/4 で、修正するスキーマとスキーマ バージョンを選択し、作成しているテスト データベースと関連付けます。

メモ: 2003.06.10 より前のバージョンの ClearQuest を実行するクライアントがアクセスするテスト データベースには、CharacterSetValidation パッケージを適用し、テスト データベースに ClearQuest データ コード ページ値を設定してください。詳細については、146 ページの「パッケージのアップグレード方法」と 35 ページの「ClearQuest のデータ コード ページ値の設定」を参照してください。



これで、スキーマをチェックアウトして修正する準備が整いました。以後、スキーマを修正し終わってテストする段階になったら、このテスト データベースを設定し、テストに使用します。詳細については、89 ページの「スキーマ用テスト データベースの設定」を参照してください。

スキーマのチェックアウト

スキーマをカスタマイズするには、スキーマ リポジトリからスキーマをチェックアウトする必要があります。スキーマをチェックアウトすると、ClearQuest によって編集用にスキーマの新しいバージョンが作成されます。

メモ: スキーマ リポジトリの ClearQuest データ コード ページ値が、ClearQuest Designer を実行しているコンピュータのオペレーティング システムのコード ページに一致しない場合は、スキーマをチェックアウトできません。データ コード ページ値の詳細については、28 ページの「ClearQuest のデータ コード ページの概要」を参照してください。

ClearQuest Designer で編集するスキーマをチェックアウトするには

- 1 [スキーマを開く] ダイアログ ボックスを表示します。[スキーマを開く] ダイアログ ボックスは、[このウィザードを起動時に表示] オプションをオンにしていると、ログオン時に表示されます。また、このダイアログ ボックスはいつでも、[ファイル] メニューの [スキーマを開く] をクリックして表示できます。
- 2 [スキーマを開く] ダイアログ ボックスで、[スキーマをチェックアウトして編集または前に保存したスキーマを引き続き編集する] がオンになっていることを確認します。
- 3 スキーマを選択し、[次へ] をクリックします。



- 4 コメントを [コメント] ボックスに入力し (オプション)、[完了] をクリックします。ClearQuest Designer にワークスペースが開き、スキーマとその構造が表示されます。

メモ: スキーマ リポジトリの ClearQuest データ コード ページ値が設定されていない場合は、スキーマには ASCII データのみを入力できます。

スキーマを読み取り専用で開く

スキーマの内容を確認するのみで変更しない場合は、スキーマをチェックアウトする必要はありません。この場合は、スキーマを読み取り専用で開きます。

ClearQuest Designer でスキーマを読み取り専用で開くには

- 1 [スキーマを開く] ダイアログ ボックスを表示します。[スキーマを開く] ダイアログ ボックスは、[このウィザードを起動時に表示] をオンにしてあると、ログオン時に表示されます。また、このダイアログ ボックスはいつでも、[ファイル] メニューの [スキーマを開く] をクリックして表示できます。
- 2 [スキーマを開く] ダイアログ ボックスで、[既存のスキーマ バージョンを表示する] をクリックします。
- 3 表示するスキーマ バージョンを選択し、[次へ] をクリックします。

選択したスキーマが読み取り専用で開きます。カスタマイズすることはできません。

スキーマを表示したまま別のスキーマを開くと、現在表示しているスキーマが ClearQuest Designer によって閉じられます。

現在表示されているスキーマを閉じるには

- [ファイル] メニューの [作業を閉じる] をクリックします。

Blank スキーマからの新規スキーマの作成

事前定義されたスキーマと大幅に異なる新規スキーマを作成する場合は、Blank スキーマをチェックアウトし、編集します。

Blank スキーマにはシステム フィールドのみがあります。システム フィールドは、ClearQuest にとって必要なものであり、編集することはできません。Blank スキーマを使用する場合は、状態モデルのフィールド、コンポーネント、その他の新規スキーマの要素を追加して、まったく新しいスキーマを作成します。

Blank スキーマから新規スキーマを作成するには

- 1 ClearQuest Designer で、[ファイル] メニューの [新規スキーマの作成] をクリックします。
- 2 [新規スキーマの作成] ダイアログ ボックスで、Blank スキーマを選択し、[次へ] をクリックします。



- 3 [スキーマ名] を入力します。次に、[コメント] を入力し (オプション)、[完了] をクリックします。
- 4 ClearQuest Designer で、新規スキーマのバージョン 1 が作成され、データベースを新規スキーマと関連付けるかどうかをたずねるメッセージが表示されます。[はい] をクリックして新規データベース ウィザードを開き、データベースを関連付けます。この時点でデータベースを関連付けない場合は、[いいえ] をクリックします。
- 5 ClearQuest Designer から、その新規スキーマを編集するためにチェックアウトするかどうかをたずねられます。[はい] をクリックしてスキーマをチェックアウトして、[コメント] を入力し (オプション)、[OK] をクリックします。

スクリプト言語の選択

フックとは、指定の条件下で実行されるスクリプト (トリガ) のことで、特定のタスクを実行します。

フックは、Windows 用には BASIC (VBScript)、Windows、UNIX の両方には Perl で作成できます。UNIX 用のスクリプトを作成する場合は、Perl を使用します。

デフォルトでは、ClearQuest でのスクリプトの実行には、Windows の場合は BASIC、UNIX の場合は Perl が使用されます。

スキーマをチェックアウトした後に、Windows で ClearQuest がスクリプトの実行に使用するスクリプト言語を選択できます。

ClearQuest Designer でスクリプト言語を選択するには

- 1 ワークスペースで、[スキーマのプロパティ] をダブルクリックします。
- 2 [スキーマのプロパティ] ダイアログ ボックスで、[BASIC] または [PERL] を選択します。



ワークフローのカスタマイズの詳細については、「第 10 章 ワークフローのカスタマイズにおけるフックの使用法」を参照してください。スクリプト言語については、次のサイトを参照してください。

- <http://msdn.microsoft.com/scripting>
- <http://www.perl.com> (ただし、英語のみのご利用となります)

スキーマの変更の確認

スキーマを修正したら、そのスキーマをスキーマ リポジトリにチェックインする前に確認する必要があります。

実際には、スキーマをチェックインしようとする、ClearQuest によって自動的に確認され、確認エラーがある場合はチェックインできません。

確認中は ClearQuest によって多数のテストがスキーマに対して実行されます。たとえば、次のことが行われているかどうかを確認されます。

- 各フィールドと各アクションに一意の名前が入力されていること。
- 各フィールドに、データ型と、レコード タイプの各状態の動作が割り当てられていること。
- REFERENCE データ型または REFERENCE_LIST データ型である各フィールドに対し、reference_to プロパティにレコード タイプが入力されていること。
- すべての状態遷移アクションに、遷移元の状態と遷移先の状態が指定されていること。
- すべての状態なしレコード タイプに、一意のキーが定義されていること。
- SQL 予約語が、適切に使用されていること。

スキーマにパッケージが適用されている場合は、ClearQuest ではそのパッケージに適用される一意の確認ルールでのテストも実行されます。

メモ: フック コードが正しく作成されていないと、実行時にエラーが発生することがあります。フックをテストするには、スクリプト エディタで [フック] メニューの [コンパイル] をクリックし、フック コードを必要に応じてデバッグします。また、確認ではフック コードのエラーは検出されない、確認してもスキーマが想定どおりに機能しない場合があります。スキーマをチェックインする前に、テスト データベースを使用してスキーマをテストしてください。また、スキーマの変更を適用する前に、ユーザー データベースをバックアップしてください。

詳細については、90 ページの「テスト データベースでのスキーマのテスト」を参照してください。

スキーマを確認するには

- 1 ClearQuest Designer で、[ファイル] メニューの [確認] をクリックします。

ClearQuest によって、ClearQuest Designer ウィンドウの下部にある [確認] ペインに確認結果が表示されます。



- 2 確認結果を検証して、必要であればスキーマを修正してエラーを解決します。

スキーマの変更の確認をすべて済ませることができない場合は、スキーマの変更を保存し、後で編集を続行できます。スキーマのチェックアウトを取り消して、スキーマを前のバージョンに戻すこともできます。詳細については、91 ページの「スキーマのチェックアウトの取り消し」を参照してください。

スキーマ用テスト データベースの設定

スキーマをチェックアウトして編集する前に、テスト データベースを作成し、スキーマと関連付けておく必要があります。詳細については、82 ページの「テスト データベースの作成」を参照してください。

スキーマをチェックアウト、修正、検証したら、テスト データベースをこのスキーマ用に設定します (つまり、前で作成したテスト データベースを、チェックアウトしたスキーマのテストで使用するよう指定します)。

テスト データベースを使用して、スキーマに加えた修正をテストできます。テスト データベースでのスキーマのテストについては、90 ページの「テスト データベースでのスキーマのテスト」を参照してください。

テスト データベースをスキーマ用に設定するには

- 1 カスタマイズするスキーマをチェックアウトしていない場合は、ClearQuest Designer で、対象のスキーマをチェックアウトします。
- 2 [データベース] メニューの [テスト データベースの設定] をクリックして、このスキーマに使用するために作成したテスト データベースを選択します。

テスト データベースでのスキーマのテスト

スキーマのカスタマイズをテストする前に、まずテスト データベースを作成し、そのスキーマ用に設定する必要があります。詳細については、82 ページの「テスト データベースの作成」と 89 ページの「スキーマ用テスト データベースの設定」を参照してください。

ClearQuest Designer を使用してスキーマをテスト データベースでテストするには

- 1 ClearQuest Designer でスキーマを修正した後、[ファイル] メニューの [作業内容のテスト] をクリックします。

ClearQuest によって、スキーマの変更の保存と確認が行われます。エラーが発生した場合は、ClearQuest Designer ウィンドウの下部にある [確認] ペインにエラーが表示されます。詳細については、88 ページの「スキーマの変更の確認」を参照してください。

ClearQuest によって、テスト データベースが自動的に新規のスキーマ バージョンのデータベースに更新されます。

- 2 Windows では、ClearQuest によって自動的に ClearQuest クライアントが起動され、予期したとおりにスキーマが動作するかどうかをテストできます。

メモ: UNIX クライアントまたは Web クライアントからスキーマをテストするには、手動でテスト データベースにログインする必要があります。UNIX でスキーマをテストするときは、特にフォームのレイアウトとスクリプトの実行に注意してください。

- 3 確認エラーを修正します。確認エラーがある場合は、スキーマをチェックインできません。

スキーマの編集セッションで [ファイル] メニューの [作業内容のテスト] をクリックした後は、スキーマのチェックアウトを取り消すことはできません。

テスト データベースは、スキーマの最新の変更で更新しておく必要があります。作業内容をテストしないでスキーマをチェックインしようとする、作業内容をテストするよう求めるメッセージが ClearQuest によって表示されます。

設計、開発、修正の作業量が多い場合は、作業内容を頻繁に保存してください。スキーマをチェックインしないで変更を保存するには、[ファイル] メニューの [作業内容の保存] をクリックします。この [作業内容の保存] では、変更が確認されません。また、ここで保存したバージョンのスキーマでテスト データベースをアップグレードすることはできません。

スキーマのチェックイン

スキーマの編集とテストを終了したら、スキーマをスキーマ リポジトリにチェックインする必要があります。スキーマをチェックインすると、ClearQuest によってスキーマの新しいバージョンがスキーマ リポジトリに保存されます。

チェックインした新規のスキーマ バージョンは、既存のユーザー データベースまたは新規データベースに適用できます。ClearQuest では各スキーマ バージョンの履歴が保存されるので、以前のバージョンを参照して新規データベースに使用することができます。

スキーマをチェックインしても、既存のユーザー データベースには影響しません。変更を有効にするには、スキーマの新しいバージョンをユーザー データベースに適用する必要があります。92 ページの「ユーザー データベースのアップグレード」を参照してください。

スキーマをチェックインするには

- 1 ClearQuest Designer で、[ファイル] メニューの [チェックイン] をクリックします。

ClearQuest によってスキーマが確認されます。エラーが発生した場合は、ClearQuest Designer ウィンドウの下部にある [確認] ペインにエラーが表示されます。

- 2 確認結果を検証します。

確認エラーがあったら、必要に応じてスキーマを修正してエラーを解決してください。

確認エラーがある場合は、スキーマをチェックインできません。

スキーマの確認で問題がなければ、ClearQuest Designer に [チェックイン] ダイアログ ボックスが表示され、元のチェックアウト コメントが表示されます。必要に応じてコメントを編集し、[OK] をクリックします。

スキーマのチェックアウトの取り消し

スキーマの変更を保存した後も、スキーマを以前のバージョンに戻すことができます。スキーマのチェックアウトを取り消すことで、チェックアウト後に加えられたすべての変更が削除され (複数のセッションにわたって変更され、保存されていても)、スキーマは 1 つ前のチェックイン バージョンに戻ります。

スキーマのチェックアウトを取り消すには

- ClearQuest Designer で、[ファイル] メニューの [チェックアウトの取り消し] をクリックします。

メモ: [ファイル] メニューの [作業内容のテスト] をクリックして、作業内容をテストすると、ClearQuest によってテスト データベースが自動的に新規のスキーマ バージョンに更新されます。スキーマのチェックアウトを取り消す場合は、その前にテスト データベースを削除する必要があります。テスト データベースを削除すると、ClearQuest でスキーマのチェックアウトを取り消すことができるようになります。その後、新規のテスト データベースを作成します。

ユーザー データベースのアップグレード

スキーマをチェックインした後、以前のバージョンのスキーマを使用したユーザー データベースをアップグレードすることができます。直ちにアップグレードしなくてもかまいませんが、アップグレードしないと、スキーマの変更がユーザーに対して有効になりません。

変更をユーザー データベースに適用する前に、ユーザーがデータベースにログインしていないことを確認してください。ユーザーがユーザー データベースに接続していると、アップグレードできません。**ClearQuest** では、データベースの更新中はユーザーが新規にログインすることはできませんが、既にログインしているユーザーの接続は切断されません。

スキーマの変更をユーザー データベースに適用する際は、次の制約に注意してください。

- ユーザー データベースを特定のスキーマ バージョンと関連付けた後は、そのデータベースに適用できるのは同じスキーマの、より新しいバージョンのみです。以前のバージョンまたは別のスキーマを適用することはできません。

警告: ユーザー データベースをアップグレードしたら、元に戻すことはできません。スキーマの変更をユーザー データベースに適用する前に、ユーザー データベースをバックアップしてください。

ユーザー データベースにスキーマの変更を適用するには

- 1 **ClearQuest Designer** で、[データベース] メニューの [データベースのアップグレード] をクリックします。



更新の前にスキーマ リポジトリとユーザー データベースをバックアップするよう求めるメッセージが **ClearQuest** によって表示されます。まだデータベースのバックアップを実施していなければ [いいえ]、既にバックアップ済みであれば [はい] をクリックします。

- 2 [データベース] リストでデータベースを選択し、[次へ] をクリックします。
 - 3 [バージョン] リストでスキーマのバージョンを選択し、[完了] をクリックします。
- データベースの詳細については、「第3章 データベースの管理」を参照してください。

進行中の作業の保存

変更をコミットしたりスキーマ バージョン番号を上げることなく、同じスキーマ バージョンを複数の編集セッションで編集できます。スキーマをチェックアウトして編集し、進行中の作業を保存して、ログオフします。次にログインしたときに、同じスキーマ バージョンの作業を続行できます。この機能は、その時点までの作業を保存する必要があっても、変更をデータベースにコミットする準備が整っていない場合に便利です。

スキーマの編集セッション中に進行中の作業を保存するには

- 1 ClearQuest Designer で、[ファイル] メニューの [作業内容の保存] をクリックします。
- 2 [ファイル] メニューの [作業を閉じる] または [終了] をクリックして、編集セッションを終了します。

スキーマを再度開き、編集を続行するには

- 1 [スキーマを開く] ダイアログ ボックスを表示します。[スキーマを開く] ダイアログ ボックスは、ClearQuest Designer にログインすると自動的に表示されます。既に ClearQuest Designer を開いている場合は、[ファイル] メニューの [スキーマを開く] をクリックして表示します。
- 2 [スキーマを開く] ダイアログ ボックスで、[スキーマをチェックアウトして編集または前に保存したスキーマを引き続き編集する] をクリックし、以前に作業していたスキーマ バージョンを選択します。

ClearQuest Designer で、編集するスキーマ バージョンのスキーマが開きます。

メモ: スキーマをスキーマ リポジトリにチェックインする前に、スキーマを確認し、作業内容をテストする必要があります。詳細については、88 ページの「スキーマの変更の確認」、90 ページの「テスト データベースでのスキーマのテスト」、91 ページの「スキーマのチェックイン」を参照してください。

スキーマまたはスキーマ バージョンの削除

警告: スキーマの削除は最終作業になります。削除したスキーマまたはスキーマ バージョンは復元できません。

スキーマの 1 つのバージョン、またはすべてのバージョンを削除できます。スキーマまたはスキーマ バージョンを削除する前に、次の点を確認してください。

- 現在、そのスキーマがユーザー データベースと関連付けられていないこと。関連付けられている場合は、まずユーザー データベースを削除する必要があります。
- 現在、そのスキーマがスキーマ リポジトリからチェックアウトされていないこと。チェックアウトされている場合は、まずそのスキーマをチェックインする必要があります。

スキーマまたはスキーマ バージョンをスキーマ リポジトリから削除するには

- 1 ClearQuest Designer で、以下の操作を行います。
 - スキーマのすべてのバージョンを削除するには、[ファイル] メニューの [スキーマの削除] をクリックします。
 - スキーマの 1 つのバージョンを削除するには、[ファイル] メニューの [スキーマ バージョンの削除] をクリックします。
- 2 削除するスキーマまたはスキーマ バージョンをクリックし、[削除] をクリックします。削除するかどうかを確認するメッセージが表示されたら、[はい] をクリックします。

別のスキーマへのデータベースの変更

ユーザー データベースをスキーマと関連付けた後は、そのデータベースのアップグレードに使用できるのは同じスキーマの、より新しいバージョンのみです。まったく別のスキーマをデータベースに適用することはできません。たとえば、ユーザー データベースで DefectTracking スキーマのバージョン 1 を使用している場合には、このデータベースを DefectTracking スキーマのバージョン 2 にアップグレードできます。また、このデータベースをアップグレードして Enterprise スキーマを使用することもできません。

ただし、既存のデータベースのデータを、新規スキーマと関連付けられている新規データベースに移動することはできます。

既存のユーザー データベースから新規スキーマと関連付けられた新規データベースにデータを移動するには、まず新規データベースを作成し、それを新規スキーマと関連付ける必要があります。その後、ClearQuest ツールを使用して、元のデータベースからデータをエクスポートし、新規データベースにそのデータをインポートします。

ユーザー データベースを別のスキーマに変更するには

- 1 最初に、「第 3 章 データベースの管理」を読んでください。
- 2 新規ユーザー データベースを作成し、そのデータベースを新規スキーマと関連付けます。

Designer で、[データベース] メニューの [新規データベースの作成] をクリックして、新規ユーザー データベースを作成します。ClearQuest 新規データベースをスキーマと関連付けるよう求めるメッセージが ClearQuest Designer に表示されます。新規データベースを新規スキーマと関連付けます。

- 3 ClearQuest エクスポート ツールを使用して、現在のユーザー データベースからデータをエクスポートします。詳細については、「第 13 章 データのインポートとエクスポート」を参照してください。
- 4 ClearQuest インポート ツールを使用して、新規データベースにデータをインポートします。詳細については、「第 13 章 データのインポートとエクスポート」を参照してください。

cqlload コマンド行ユーティリティを、ClearQuest のインポート ツール、エクスポート ツールと共に使用すると、スキーマの全体または一部を別のスキーマ リポジトリにコピーできます。詳細については、「第 13 章 データのインポートとエクスポート」と「付録 E コマンド行ユーティリティ」を参照してください。

変更依頼の処理方法を修正するには、**Rational ClearQuest Designer** を使用してスキーマのレコードとフィールドをカスタマイズする必要があります。

この章の内容は、次のとおりです。

- 97 ページの「スキーマのカスタマイズの概要」
- 98 ページの「レコード タイプの操作」
- 104 ページの「フィールドの操作」
- 119 ページの「状態モデルの概要」
- 121 ページの「状態遷移マトリックスを使用した状態モデルのカスタマイズ」
- 125 ページの「アクションとアクション タイプの操作」

スキーマのカスタマイズの概要

スキーマを変更する場合、「第 4 章 ClearQuest スキーマの操作」で説明している標準的な手順に従います。第 3 章では、スキーマのチェックアウト、**ClearQuest Designer** の使用によるスキーマの変更、パッケージの適用によるスキーマの変更、スキーマの確認、スキーマのテスト、スキーマのチェックイン、ユーザー データベースのアップグレードについて説明しています。

ClearQuest Designer を使用したスキーマの変更とは、次の処理を指します。

- 1 必要なレコード タイプを定義します。複数の変更管理プロセスがあり、それぞれが従うプロセス モデル (障害の処理や新規機能の依頼の管理など) が異なる場合、レコード タイプを複数作成する必要があります。
- 2 各レコード タイプのデータを格納するフィールドとそのフィールドの動作を定義します。
- 3 各レコード タイプの状態モデルに属する状態を定義します。
- 4 各状態モデルのアクションを定義します。
- 5 新規レコードの追加と既存のレコードの操作に使用するフォームを定義します。
- 6 フック (スクリプト) を追加して、フィールドとアクションをカスタマイズします。

この章では、最初の 2 つの手順、つまり、レコード タイプを定義する方法と、フィールドとその動作を定義する方法について説明します。また、フックを追加してフィールドをカスタマイズする方法についても説明します。そのほかの手順については、以降の項で説明します。

レコードタイプの操作

ClearQuest でレコード タイプとは、特定のタイプの変更依頼に使用するレイアウトまたはフォーマットのことで、リレーショナルデータベースのテーブルと大筋では似ています。各レコード タイプは、1 つのタイプの変更依頼を収集できるデータを定義します。個々の変更依頼に関する情報をレコード、変更依頼に関するデータの個々の部分をフィールドと呼びます。

各レコード タイプは専用の状態モデル、フォーム、フックに関連付けられており、そのタイプの変更依頼で使用するデータの収集と表示を制御します。

状態のあるレコード タイプと状態なしレコード タイプ

ClearQuest では、状態のあるレコード タイプと状態なしレコード タイプの両方をサポートしています。

状態のあるレコード タイプは、ClearQuest のユーザーが実行するアクションの結果として、一連のステータスまたは状態 (Submitted、Assigned、Resolved など) の中で状態が移行するレコード タイプです。

状態なしレコード タイプは、データを保持しますが、状態を変化させないレコード タイプです。状態なしレコード タイプには、ユーザー、プロジェクト、顧客などがあります。状態なしレコード タイプに実行できるアクションは、Submit、Modify、Delete、Import です。

状態のあるレコードは1つ以上の状態なしレコードを参照できます。たとえば、ClearQuest のユーザーが障害 (状態のあるレコード タイプ) をプロジェクト (状態なしレコード タイプ) にアサインしたとします。

ClearQuest では、History、Attachments、Groups、Users の4つの状態なしシステム レコード タイプが維持されます。システム レコード タイプは、削除できません。

レコード タイプの数

スキーマには複数のレコード タイプを格納することができます。たとえば、1 つのスキーマに Software Enhancements 専用のレコード タイプと Hardware Enhancements 専用のレコード タイプを設定することができます。また、Issues、Problem Reports、Change Requests、Defects、Enhancement Requests 用に異なるレコード タイプを設定することもできます。

一般的には、変更依頼ごとに異なるプロセス モデルが含まれている場合、または変更依頼ごとに異なるデータの追跡が必要な場合、別個のレコード タイプを作成してください。たとえば、組織にソフトウェア拡張とハードウェア拡張とで異なるプロセス モデルがある場合、2 種類の別個のレコード タイプを作成してください。一方、ソフトウェア拡張とハードウェア拡張とで使用するプロセス モデルが同じ場合、拡張の種類を指定するフィールドがある Enhancements レコード タイプを1つのみ作成してください。

どのようなレコード タイプを作成するかは慎重に検討してください。レコード タイプを多くすると、プロセス モデルのより多くのバリエーションを取り込むことができます。ただし、レコードタイプを作成しすぎると管理が複雑になり、多数の変更依頼を受け容れるクエリーやレポートの作成が難しくなります。また、次のケースを前もって検討してください。現在は同じプロセス モデルを保持している 2 つのタイプの変更依頼が将来的に分かれることが予想される場合、最初から 2 つのレコードタイプを作成する方が、後で分割するより簡単です。

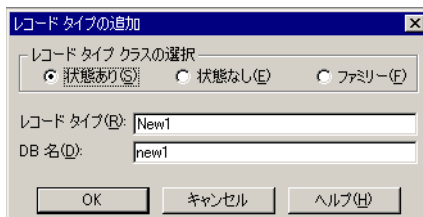
リレーショナルデータベースの設計で問題になるような点についても検討してください (または、こうした問題に詳しいデータベース管理者の支援を受けてください)。たとえば、**Defects** レコードタイプとして、登録者、登録者の電子メールアドレス、登録者の電話番号を作成するのではなく、**Defects** レコードタイプとしては登録者のみを作成し、**Submitters** レコードタイプを追加で作成するというようにします。これにより、障害を登録するたびに電子メールアドレスと電話番号を入力する必要がなくなります。その場合には、**REFERENCE** フィールドを使用してレコードタイプ **Defect** と **Submitter** との間にリンクを作成すると、登録者の電子メールアドレスと電話番号をフォームとレポートに表示できます。116 ページの「レコードのリンクによる親/子階層の作成」を参照してください。

新規レコード タイプの追加

新規レコード タイプをスキーマに追加できます。新規レコード タイプを作成する際、状態ありにするか状態なしにするかを決定します。作成後は変更できません。

レコードタイプを追加するには

- 1 ClearQuest Designer で、[編集] メニューの [レコード タイプ/ファミリーの追加] をクリックします。
- 2 [レコードタイプの追加] ダイアログ ボックスで、レコードタイプのクラス ([状態あり] または [状態なし]) を選択します。
- 3 [レコードタイプ] ボックスに、レコードタイプの名前を入力します。



メモ: ClearQuest ではテーブルの DB 名を使用します。デフォルトでは、レコードタイプ名と同じです。

状態なしレコードタイプを追加する場合は、1 つ以上のフィールドを一意のキーとして選択する必要があります。詳細については、100 ページの「状態なしレコードタイプの一意のキーの選択」を参照してください。

4 [OK] をクリックします。

各スキーマにはデフォルトのレコードタイプが必要です。ClearQuest ではこれを使用してレコードを登録するためのショートカットを作成します。作成したスキーマにデフォルトのレコードタイプがまだ指定されていない場合、指定してください。詳細については、101 ページの「スキーマのデフォルトのレコードタイプの選択」を参照してください。

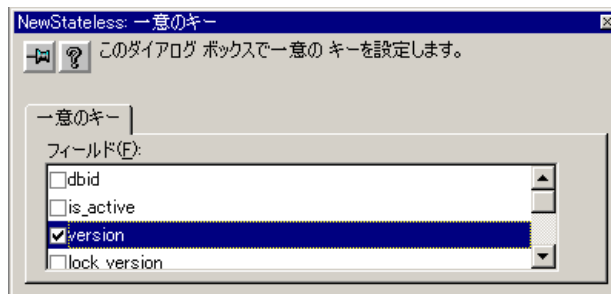
状態なしレコードタイプの一意のキーの選択

新規の状態なしレコードタイプをスキーマに追加する場合は、1 つ以上のフィールドを一意のキーに選択する必要があります。ClearQuest では、一意のキーを使用して個々の変更依頼を追跡します。

メモ: 一意のキーを明示的に参照するフック コードがスキーマに含まれていて、その一意のキーを変更する場合は、新しい一意のキーのフィールド名を参照するようにコードを修正してください。

状態なしレコードタイプに一意のキーを選択するには

- 1 ClearQuest Designer のワークスペースで、[レコードタイプ - 状態なし] フォルダを展開し、[NewStateless] フォルダを展開します。[一意のキー] をダブルクリックします。
- 2 [NewStateless: 一意のキー] ダイアログボックスで、一意のキーとして使用する 1 つ以上のフィールド名をオンにします。



- 3 [NewStateless: 一意のキー] ダイアログボックスを終了します。

スキーマのデフォルトのレコード タイプの選択

各スキーマは、デフォルトのレコード タイプを持つ必要があります。デフォルトのレコード タイプは、状態があるものでも、ないものでもかまいません。ClearQuest は、デフォルトのレコード タイプを使用して ClearQuest のクライアントに、そのタイプのレコードの登録に使用できるショートカット ボタンを作成します。また、別のレコード タイプが明示的に指定されていない場合、ClearQuest はデフォルトのレコード タイプを使用します。

レコード タイプのデフォルトへの設定

ClearQuest Designer のワークスペースで、[レコード タイプ] フォルダまたは [レコード タイプ - 状態なし] フォルダをクリックします。次に、デフォルトに設定するレコード タイプを右クリックし、ショートカット メニューの [デフォルトのレコード タイプ] をクリックします。

レコード タイプ ファミリーの操作

共通フィールドを持つ、状態のあるレコード タイプのいくつかをグループ化して、レコード タイプ ファミリーを作成できます。これにより、ClearQuest のユーザーは一度に複数のレコード タイプのクエリーを実行できます。

レコード タイプ ファミリーを作成する際は、次の点に注意してください。

- ファミリーに含まれるレコード タイプは、1 つ以上の共通フィールド (各レコード タイプで同じフィールド) を持つ必要があります。レコード タイプ ファミリーのクエリーに使用できるフィールドは、共通フィールドのみです。
- ClearQuest のユーザーが [クエリー] メニューの [新規クエリーの作成] をクリックすると、レコード タイプとレコード タイプ ファミリーが同じウィンドウに表示されます。このため、レコード タイプ ファミリーと個別のレコード タイプを区別できる命名規則を使用してください。

レコード タイプ ファミリーを作成するには、次の作業を実行します。

- 102 ページの「レコード タイプ ファミリーの追加」
- 103 ページの「共通フィールドの作成」

また、レコード タイプ ファミリーからのメンバの削除、レコード タイプ ファミリーの名前の変更または削除も実行できます。

レコード タイプ ファミリーの追加

スキーマに新規のレコード タイプ ファミリーを作成するには

- 1 [編集] メニューの [レコード タイプ/ファミリーの追加] をクリックします。
- 2 [レコード タイプの追加] ダイアログ ボックスで、レコード タイプのクラスとして [ファミリー] を選択します。
- 3 [レコード タイプ] ボックスに、レコード タイプ ファミリーの名前を入力し、[OK] をクリックします。ClearQuest Designer により、新規ファミリーが [レコード タイプ ファミリー] フォルダに追加されます。

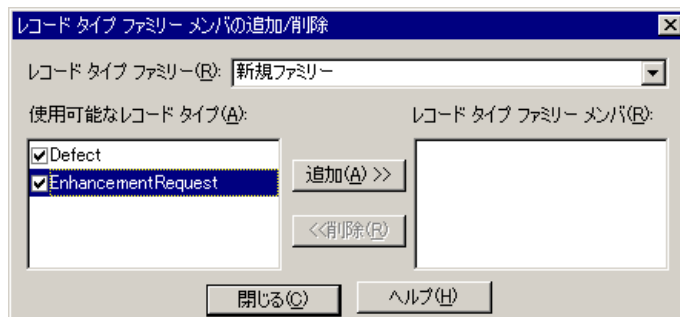


メモ: ClearQuest ではテーブルの DB 名を使用します。デフォルトでは、レコード タイプ ファミリー名と同じです。

レコード タイプ ファミリーへのメンバの追加

目的のレコード タイプを新規のレコード タイプ ファミリーに追加するには

- 1 [編集] メニューの [レコード タイプ ファミリー メンバ] をクリックします。



- 2 [レコード タイプ ファミリー メンバの追加/削除] ダイアログ ボックスで、作成したレコード タイプ ファミリーを選択します。
- 3 [使用可能なレコード タイプ] リストでファミリーに追加するレコード タイプを選択し、[追加] をクリックします。次に [閉じる] をクリックします。

レコードタイプが、ClearQuest Designer によってレコードタイプファミリーの [メンバ] フォルダに追加されます。

メモ: レコードタイプをすばやくファミリーに追加するには、[レコードタイプ] フォルダとファミリーフォルダを展開します。次に、レコードタイプをファミリーの [メンバ] フォルダにドラッグします。

共通フィールドの作成

レコードタイプファミリーに含まれるすべてのメンバレコードタイプには、1 つ以上の共通フィールドが必要です。レコードタイプメンバに共通フィールドがない場合、共通フィールドを追加してください。

共通フィールドは、同じ名前を持ち、同じデータ型である必要があります。たとえば、Defect レコードタイプと Enhancement レコードタイプをグループ化する場合、各レコードタイプの Description フィールドを共通フィールドとして使用できます。ただし、両レコードタイプでフィールドが同じデータ型でなければなりません (一般的に Short String)。レコードタイプへのフィールドの追加方法とフィールドのデータ型の定義方法の詳細については、104 ページの「フィールドの操作」を参照してください。

レコードタイプファミリーからのメンバの削除

レコードタイプファミリーからレコードタイプを削除するには

- 1 [編集] メニューの [レコードタイプファミリーメンバ] をクリックします。
- 2 [レコードタイプファミリーメンバの追加/削除] ダイアログボックスで、目的のレコードタイプファミリーを選択します。
- 3 [使用可能なレコードタイプ] リストで、削除するレコードタイプを選択し、[削除] をクリックします。次に [閉じる] を選択します。

レコードタイプまたはファミリーの名前の変更

レコードタイプ名を変更する場合は、レコードタイプを参照しているフックのレコードタイプ名も変更する必要があります。フックを編集しないと、想定どおりにスクリプトが動作しません。

メモ: 状態なしのシステムレコードタイプ Ratl_replicas、History、Attachments、Group、Users は名前を変更できません。

レコードタイプまたはファミリー名を変更するには

- 1 [編集] メニューの [レコードタイプ/ファミリーの名前の変更] をクリックします。
- 2 [状態あり]、[状態なし]、[ファミリー] など名前を変更するレコードタイプのクラスを選択します。

- 3 [現在の名前] リストからレコードタイプを選択します。
- 4 [新規の名前] ボックスに新しい名前を入力します。スキーマ内では、各レコードタイプおよびレコードタイプファミリー名は一意である必要があります。

メモ: ワークスペースのレコードタイプまたはファミリーを右クリックし、ショートカットメニューの[名前の変更]をクリックすることもできます。

レコードタイプまたはファミリーの削除

次のレコードタイプは削除できません。

- 状態なしシステム レコードタイプ: Ratl_replicas、History、Attachments、Groups、Users。
- 読み取り専用パッケージで追加されたレコードタイプ。詳細については、「第6章 パッケージの適用」を参照してください。
- デフォルトのレコードタイプ。現在のデフォルトレコードタイプを削除するには、まず別のレコードタイプをデフォルトに割り当てる必要があります。

フックでレコードタイプの名前を明示的に参照している場合は、スクリプトコードを修正してその名前の参照を削除する必要があります。

レコードタイプまたはファミリーを削除するには

- 1 [編集] メニューの[レコードタイプ/ファミリーの削除]をクリックします。
- 2 削除するレコードタイプのクラス([状態あり]、[状態なし]、[ファミリー]のいずれか)を選択します。
- 3 [レコードタイプ] リストでレコードタイプまたはファミリーを選択します。

メモ: ワークスペースのレコードタイプまたはファミリーを右クリックし、ショートカットメニューの[削除]をクリックして削除することもできます。

フィールドの操作

フィールドを使用して、ユーザーがデータベースに追加するデータの型を制御します。フィールドで次の操作を実行できます。

- フィールドの追加または削除。
- フィールドの動作の定義。
- フィールドへのヘルプテキストの追加。
- フィールドを使用したレコードのリンク。
- フックの追加によるフィールドのカスタマイズ。

各レコードタイプは、そのレコードタイプと関連付けられたフィールドを示すフィールドグリッドを持っています。

フィールドグリッドを表示するには

- 1 ワークスペースでレコードタイプを展開し、[フィールド]をダブルクリックします。
- 2 行に各フィールドが表示され、その行の列に各フィールドのプロパティが表示されます。

フィールドグリッドを使用して、レコードタイプに新規フィールドを追加したり、既存フィールドのプロパティを修正できます。



ClearQuest の各レコードタイプには、システムフィールドが含まれています。システムフィールドは、そのタイプのすべてのレコードに必要です。システムフィールドは、フィールドグリッドではグレーで表示されます。

フィールドの追加と変更については、次の点に留意してください。

- ユーザーに対し、新規フィールドを使用可能にしたり、削除したフィールドを表示しないようにするには、そのフィールドを参照するフォームも修正する必要があります。
- フィールドの追加または変更の後、スキーマをスキーマリポジトリにチェックインし、スキーマのそのバージョンをユーザーデータベースに適用する必要があります。
- スキーマへのチェックイン後は、そのフィールドの型、サイズ、DB名は変更できません。ただし、フィールドの参照名は変更できます。

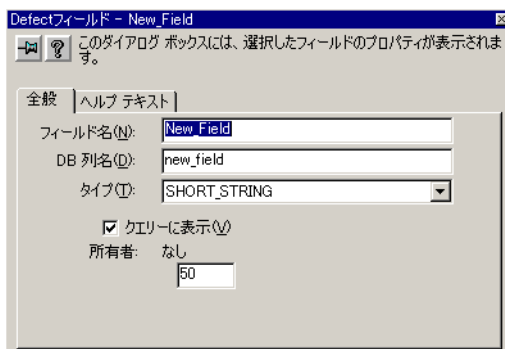
新規フィールドの追加

新規フィールドをレコードタイプに追加するには

- 1 フィールドをフィールドグリッドに追加します。
- 2 その新規フィールドにフォームコントロールを関連付け、そのフィールドを ClearQuest クライアントのユーザーに対して使用可能にします。詳細については、153 ページの「フォームへのコントロールの追加」を参照してください。
- 3 スキーマをテストし、スキーマリポジトリにチェックインして、そのスキーマをユーザーデータベースに適用します。詳細については、「第4章 ClearQuest スキーマの操作」を参照してください。

新規フィールドをフィールドグリッドに追加するには

- 1 ClearQuest Designer のワークスペースで、[レコードタイプ] フォルダを展開し、レコードタイプを展開します。次に、[フィールド] をダブルクリックしてフィールドグリッドを表示します。
- 2 [編集] メニューの [フィールドの追加] をクリックします。



- 3 [<レコードタイプ> フィールド] ダイアログ ボックスで、[フィールド名] にフィールド名を入力します。

[DB 列名] は、ClearQuest がテーブルの列に使用する名前です。デフォルトでは、フィールド名と同じです。

警告: フィールドに付ける名前に、データベースの製造元により予約されているキーワードを使用しないでください。予約されているキーワードのリストについては、各製造元のマニュアルを参照してください。

4 [タイプ] ボックスからフィールドのデータ型を選択します。選択するデータ型により、このフィールドに入力できる値の種類が決定されます。詳細については、107 ページの「フィールドのデータ型を選択する上での注意事項」を参照してください。

- 型が **SHORT_STRING** であるフィールドの場合は、[最大長] に値を入力します。値は正の数で、254 文字以下にします。スキーマへのチェックイン後は、最大長を変更できません。

ClearQuest は、最大長のフィールドを収容できる量の、データベースの記憶領域を確保します。無駄な記憶領域を確保しないようにするには、最大長を可能な限り短くしてください。フィールドの最大長を超えたデータは切り捨てられます。

- 型が **REFERENCE** または **REFERENCE_LIST** であるフィールドの場合は、フィールドの参照先にするレコード タイプを選択します。このとき、レコード タイプは、状態のあるものでも状態のないものでもかまいません。オプションの逆参照フィールドを使用して、参照されたレコードからこのフィールドのレコードへのリンクを作成できます。詳細については、114 ページの「フィールドを使用したレコードのリンク」を参照してください。

5 [クエリーに表示] チェック ボックスをオンにすると、このフィールドは **ClearQuest** クライアントから実行されるクエリーに含まれます。このフィールドがクエリーに含まれないようにするには、[クエリーに表示] チェック ボックスをオフにします。

メモ: フィールドをフィールド グリッドに追加したら、その新規フィールドをフォーム コントロールに関連付けて、**ClearQuest** のユーザーに対して使用可能にしてください。詳細については、153 ページの「フォームへのコントロールの追加」を参照してください。

フィールドのデータ型を選択する上での注意事項

フィールドを追加する際は、データ型を選択する必要があります (手順 4 の「新規フィールドの追加」を参照してください)。選択するデータ型により、このフィールドに入力できる値の種類が決定されます。

ClearQuest では、フィールドで、次に示すデータ型をサポートしています。

データ	説明/コメント
ATTACHMENT_LIST	レコードに関連するファイルをレコードに格納できます。
DATE_TIME	SQL の日付および時間。
INT	SQL の整数。
MULTILINE_STRING	サイズ制限のない可変長文字列。

データ	説明/コメント
REFERENCE	レコード タイプに含まれる一意のキーへの参照。 REFERENCE 型のフィールドには、参照先として、状態のあるレコード タイプまたは状態なしレコード タイプを選択する必要があります。オプションの逆参照フィールドを入力して参照されたレコードからこのフィールドのレコードにリンクし、参照されたレコード タイプがセキュリティの制御下にあることを指定できます。
REFERENCE_LIST	レコード タイプに含まれる一意のキーへの参照 (複数)。参照リスト フィールドにより、1 つのフィールドで複数のレコードを参照できます。参照リスト フィールドは、親子コントロールと使用して、関連レコードをリンクできます。 REFERENCE_LIST 型のフィールドには、参照先として、状態のあるレコード タイプまたは状態なしレコード タイプを選択する必要があります。オプションの逆参照フィールドを使用して、参照されたレコードからこのフィールドのレコードへのリンクを作成できます。
SHORT_STRING	254 文字を最大長とする可変長の文字列。フィールドを定義する際に、プロパティ ダイアログ ボックスで長さを設定します。 [最大長] フィールドに、1 ～ 254 の値を入力します。 ユーザーによって SHORT_STRING 型のフィールドに値が入力されると、ClearQuest によって自動的に前後の空白が削除されます。
DBID	システム フィールドとして予約されています。
ID	システム フィールドとして予約されています。
JOURNAL	システム フィールドとして予約されています。
STATE	システム フィールドとして予約されています。

フィールドの型を選択する際、次のことに留意してください。

選択するタイプの検討とテストに十分時間をかけてください。後で変更するのが難しい場合があります。スキーマへのチェックイン後は、フィールドの型や DB 列名を変更できません。SHORT_STRING 型のフィールドの場合、[最大長] プロパティも変更できません。これらのプロパティを変更するには、そのフィールドを削除し、同様なプロパティの新規フィールドを作成します。

データ型が SHORT_STRING であるフィールドの場合、考えられる文字列長より大きな値を [最大長] に指定してください。[最大長] を大きくすると、ClearQuest が確保するディスク領域も大きくなりますが、入力内容が ClearQuest によって切り捨てられる可能性は低くなります。

説明、コメント、既知の回避方法など、一般的な自由形式情報には、単一の文字列フィールド (MULTILINE_STRING と SHORT_STRING) が適しています。ただし、フィールドの情報をクエリーでソートまたはフィルタリングする場合は、複数の小さなフィールドに分けることをお勧めします。たとえば、コンピュータ環境のフィールドを 1 つ定義するのではなく、ハードウェアの製造元、オペレーティング システム、OS のバージョンにそれぞれ個別のフィールドを割り当ててください。

電話番号には数字のみ受け付けるなど、内容にルールを設定する場合も、複数の小さなフィールドを使用してください。

SHORT_STRING フィールドは、選択肢のリスト (choicelist) から 1 つのみを選択するフィールドに適しています。choicelist は、スキーマに (定数リストとして) ハード コーディングするか、クライアント インターフェイスで (動的リストとして) 管理できます。後者の方法では、選択肢の変更の際に、スキーマのチェックアウト、チェックイン、ユーザー データベースのアップグレードのいずれも必要ないという利点があります。一覧表示される選択肢のみに制限するには、choicelist のプロパティ シートで [リストに制限] チェック ボックスをオンにします。これにより、無効なエントリの入力を禁止できます。

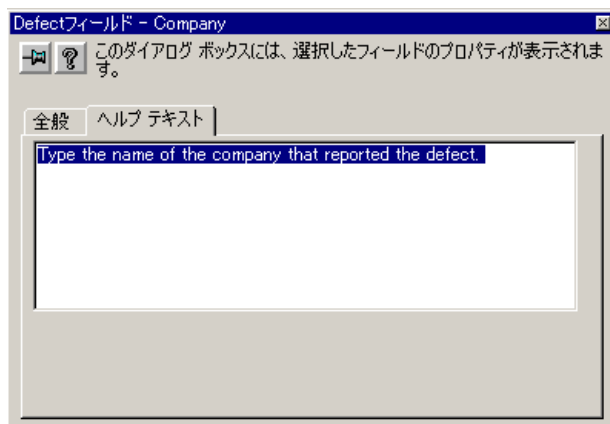
選択肢をあらかじめ用意しておき、ユーザーが複数選択できるようにするには、フィールドを別個に用意するか (さらに、対応するチェック ボックス コントロールも用意するか)、MULTILINE_STRING フィールドとリスト ボックス コントロールを使用します。

フィールドへのヘルプ テキストの追加

フィールドレベル ヘルプを ClearQuest クライアントのユーザーに提供するには、フィールドにヘルプ テキストを追加します。ヘルプ テキストには、フィールドの説明やフィールドの使用法についての具体的な説明を記述します。ClearQuest クライアントのユーザーがヘルプ テキストを表示するには、レコード フォームのフィールドを右クリックし、ショートカット メニューの [ヘルプ] をクリックします。

ヘルプ テキストをフィールドに追加するには

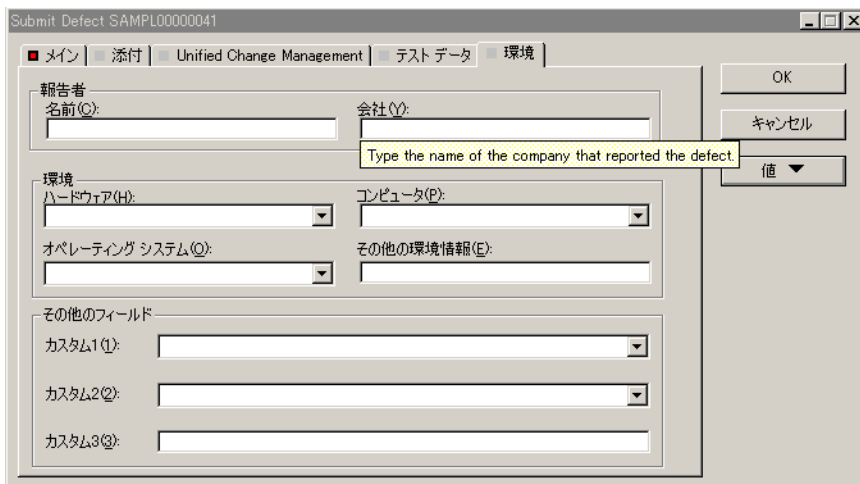
- 1 [フィールドのプロパティ] ダイアログ ボックスで、[ヘルプ テキスト] タブを選択します。
- 2 ヘルプ テキストを入力します。テキストは半角で最大 254 文字で、[ヘルプ テキスト] タブのスペースのおよそ 3 分の 2 に相当します。



- ヘルプ テキストを ClearQuest クライアントのユーザーが使用できるようにするには、スキーマへのチェックインとユーザー データベースのアップグレードを実行します。

ヘルプ テキストを ClearQuest クライアントで表示するには

ClearQuest クライアントでフィールドを右クリックし、ショートカット メニューの [ヘルプ] をクリックします。



メモ: ヘルプ テキストを ClearQuest Web で表示するには、カーソルをフィールド名の上に置き、疑問符型のカーソルが表示されたらクリックします。

フィールド動作の定義

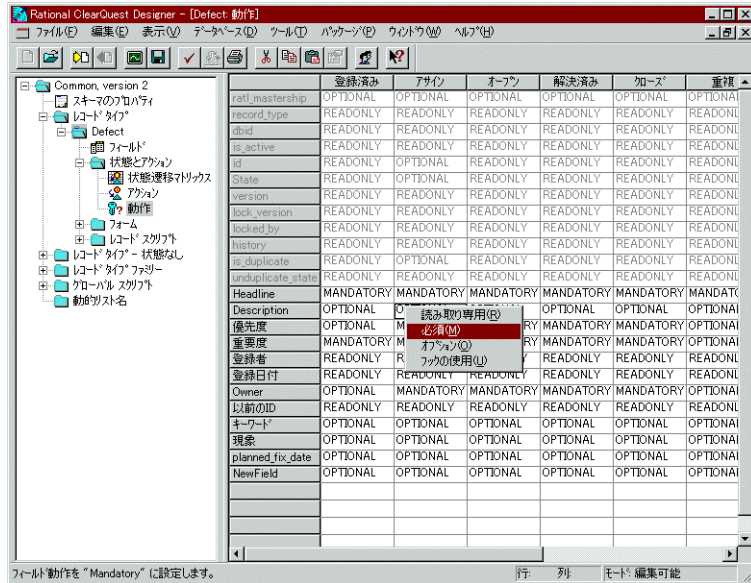
各フィールドは、1 つ以上の関連付けられた動作を持っています。状態のあるレコードタイプのフィールドには、状態ごとに異なる動作を定義できます。たとえば、あるフィールドを、**Opened** 状態ではオプション、**Resolved** 状態では必須にすることができます。状態なしレコードタイプのフィールドの場合、必要とするのは 1 つの動作のみです。

ClearQuest では、次のフィールド動作をサポートしています。

動作	説明
必須	ユーザーは、レコードへの変更を適用する前に、このフィールドに値を入力する必要があります。これを実行しないと、実行時確認エラーが発生します。必須フィールドは、レコードフォームに赤色で表示されます。
オプション	ユーザーは、このフィールドにデータを入力できますが、必ずしもそうする必要はありません。 メモ: [オプション] は、新規フィールドのデフォルト設定です。
読み取り専用	ユーザーは、フィールドの内容を表示できますが、変更できません。 メモ: フックを使用すると、[読み取り専用] フィールドを変更できます。
フックの使用	フィールドの権限フックを使用して、ユーザー アクセスのレベルを決定します。

フィールドの動作を定義するには

- 1 ClearQuest のワークスペースで、[レコードタイプ] フォルダを展開し、目的のレコードタイプを展開します。
- 2 [状態とアクション] を展開し、[動作] をダブルクリックします。



- 3 動作グリッドで、定義するフィールドの列を右クリックし、フィールドに対して目的の動作を選択します。

デフォルトのフィールド動作の定義

フィールドのデフォルト動作は、特定の状態における動作を明示的に設定していない限り、すべての状態に適用されます。また、レコードタイプに新しい状態を追加した場合も自動的に適用されます。

フィールドのデフォルトの動作を定義するには、動作グリッドで、目的のフィールドの [デフォルト動作] 列 (右側の最後の列) をクリックし、デフォルトとして使用する動作を選択します。

メモ: フックを使用しても、フィールドの動作を設定できます。フックはスーパー ユーザー権限で稼働するため、フィールド動作が読み取り専用の場合も含め、いずれのフィールドも変更できます。

フィールドの修正

フィールドのいくつかのプロパティは、そのフィールドのプロパティ ダイアログ ボックスを使用して修正できます。それら以外のプロパティは、フィールド グリッドから直接修正する必要があります。

メモ: スキーマへのチェックイン後は、フィールドのデータ型や DB 列名を変更できません。SHORT_STRING 型のフィールドの場合、[最大長] プロパティも変更できません。これらのプロパティを変更するには、そのフィールドを削除し、同様なプロパティの新規フィールドを作成します。

フィールドを変更するには

- 1 ClearQuest Designer のワークスペースで、[レコードタイプ] フォルダを展開し、目的のレコードタイプを展開します。次に、[フィールド] をダブルクリックし、フィールドグリッドを表示します。
- 2 変更するフィールドを右クリックし、ショートカットメニューの [フィールドのプロパティ] をクリックします (フィールドを選択し、[編集] メニューの [フィールドのプロパティ] をクリックする方法もあります)。

フィールド名の変更

フィールドの名前は変更できますが、スクリプトでその名前を明示的に参照している場合は、新しい名前を使用するようにスクリプトを修正する必要があります。

フィールド名を変更するには

- 1 フィールドを右クリックし、ショートカットメニューの [フィールド名の変更] をクリックします。
- 2 新しい名前を [フィールド名] に入力します。

フィールドに付ける名前に、データベースの製造元により予約されているキーワードを使用しないでください。予約されているキーワードのリストについては、各製造元のマニュアルを参照してください。

フィールドの削除

フィールドの削除には次の制限があります。

- システム フィールドの削除、変更、名前の変更はできません。システム フィールドは、フィールドグリッドではグレーで表示されます。
- フィールドを削除しても、ClearQuest ではフィールドに生成した DB 列名 (デフォルトではフィールド名と同じ) が保持されます。後で同じ名前を再利用して新規フィールドを作成しても、ClearQuest では一意の DB 列名が生成されます。
- スクリプトコードにおいてフィールドの名前を明示的に参照していて、そのフィールドを削除した場合は、そのフィールドへの参照もスクリプトコードから削除する必要があります。
- ClearQuest クライアントでは、削除されたフィールドを使用するクエリーはすべて無効になります。

フィールドを削除するには

- 1 フィールド グリッドを表示します。
- 2 フィールドを右クリックし、ショートカット メニューの [フィールドの削除] をクリックします。
- 3 レコード フォームからフィールドを削除します。113 ページの「フィールドの削除」を参照してください。

フィールドを使用したレコードのリンク

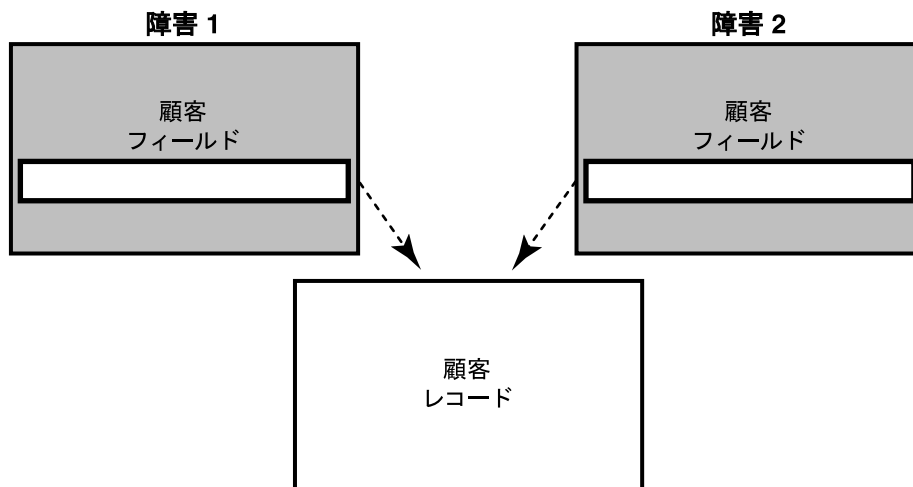
フィールドを使用して、同じタイプまたは異なるタイプのレコードをリンクできます。レコードをリンクすることで、次のことが可能です。

- 共通データの共有。
- 親子階層の作成。
- レコードのリンクによる共通データの共有。

共通データを共有するためにレコードをリンクするには、REFERENCE 型または REFERENCE_LIST 型のフィールドを使用できます。2 つのレコードをリンクする場合、REFERENCE フィールドを使用します。1 つのレコードで複数のレコードをリンクする場合、REFERENCE_LIST フィールドを使用します。

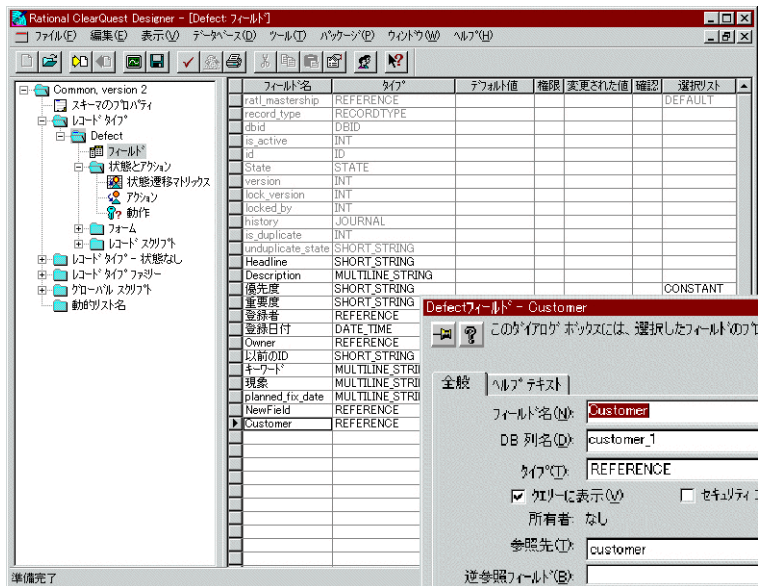
たとえば、同じ顧客データを複数のレコードに対して入力することが必要な場合がこれに当たります (図 3 を参照)。

図 3 REFERENCE_LIST フィールドを使用する共通データの共有

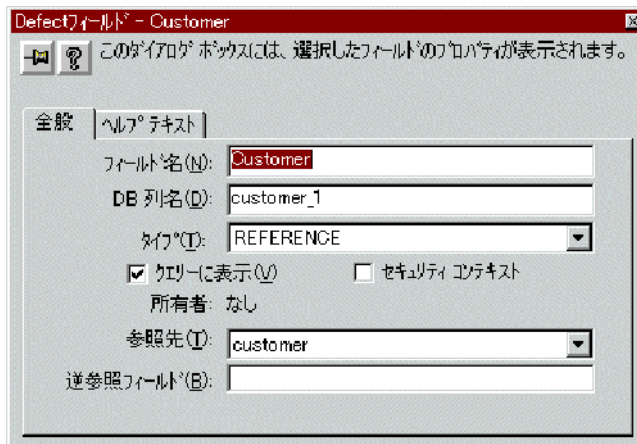


レコードをリンクして共通データを共有するには

- 1 レコードタイプのフィールドグリッドを表示し、新規フィールドを作成します。型として REFERENCE または REFERENCE_LIST を選択します。



- 2 作成した新規フィールドを右クリックし、ショートカット メニューの [フィールドのプロパティ] をクリックします。



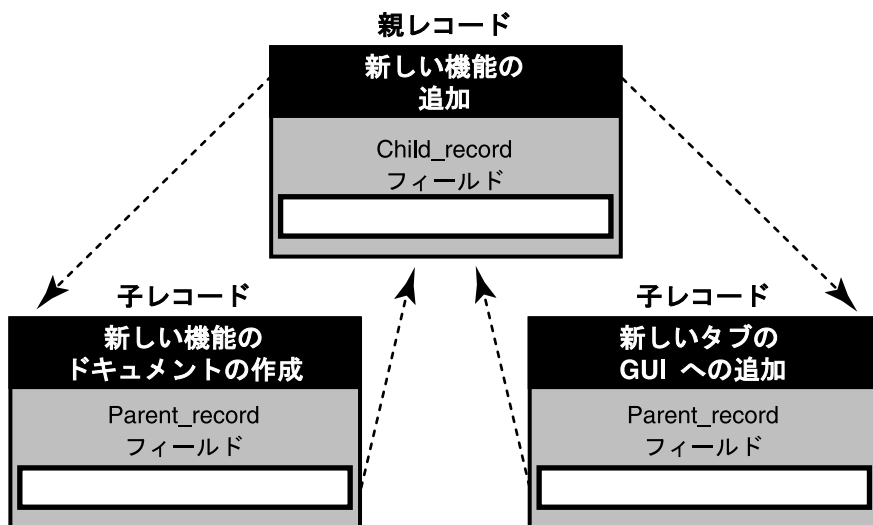
- 3 [フィールドのプロパティ] ダイアログ ボックスで、参照するレコードタイプを [参照先] リストから選択します。

フィールドをユーザーに対して使用可能にするには、そのフィールドをレコード フォームに追加する必要があります。REFERENCE_LIST フィールド タイプで親子コントロールを使用します。詳細については、153 ページの「フォームへのコントロールの追加」を参照してください。

レコードのリンクによる親/子階層の作成

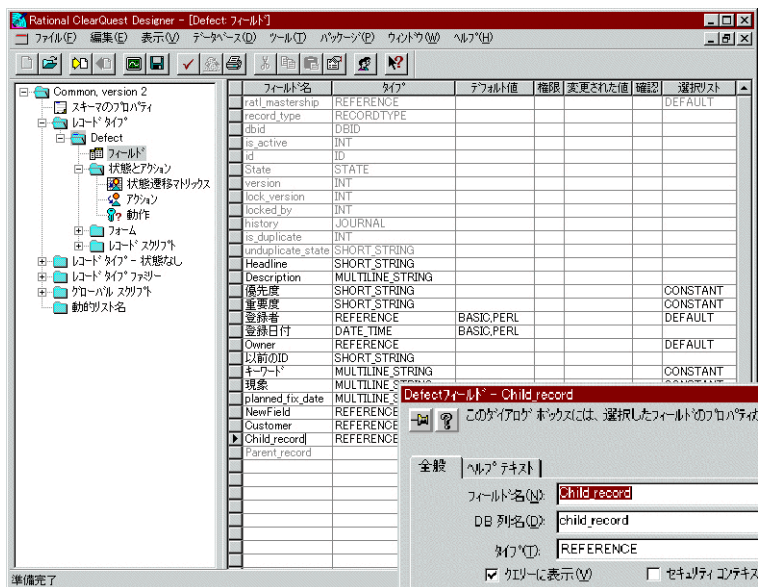
REFERENCE 型または REFERENCE_LIST 型のフィールドを使用し、同じタイプのレコードをリンクして親/子階層を作成できます。たとえば、新規機能の追加を依頼する親レコードを、関連タスク（新規機能の記述や、新規タブのインターフェイスへの追加など）を記述する 1 つ以上の子レコードに関連付けることができます（図 4 を参照）。

図 4 親 / 子階層の例



レコードをリンクして親/子階層を作成するには

- 1 親レコードに、REFERENCE_LIST 型または REFERENCE 型のフィールドを追加します。



- 2 フィールドを右クリックし、ショートカットメニューの [フィールドのプロパティ] をクリックします。



- 3 [逆参照フィールド] にフィールド名を入力します。[逆参照フィールド] は、子レコードの視点でリンクを表示することにより、親/子関係を簡単にたどることができる読み取り専用フィールドです。このフィールドは、子レコードのフィールドリストに自動的に追加されます。
- 4 その新規フィールドをレコードフォームに追加します。REFERENCE_LIST フィールドタイプで親子コントロールを使用します。153 ページの「フォームへのコントロールの追加」を参照してください。

詳細については、129 ページの「フックの追加によるアクションのカスタマイズ」を参照してください。

メモ： `Reference_List` はパフォーマンスに影響を及ぼすので、使いすぎないようにしてください。フォームが表示されるたび、またはフォームの内容がデータベースから更新されるたびに（クエリーの結果グリッドで別の行を選択したときなどに）、そのレコードの内容全体がデータベースから要求されることになります。次に、各 `reference_list` フィールドについて、フォームに表示されている参照レコードのすべてのフィールドを取得するクエリーが実行されます。フォーム コントロールでネスト レベルの深い属性を表示している場合は、すべてのデータが取得されるまでこの処理が繰り返されます。

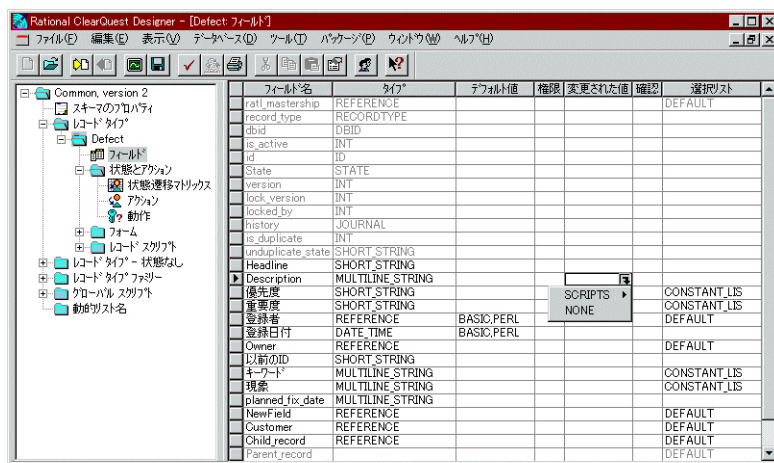
フックの追加によるフィールドのカスタマイズ

フックにより、フィールドの動作をカスタマイズできます。たとえば、新規レコードが登録されたときに常にフィールドのデフォルト値が割り当てられるようにスキーマをカスタマイズできます。詳細については、212 ページの「ユーザーが入力したフィールド値の確認」を参照してください。

ClearQuest では、次のようなフィールド フックが用意されています。

- デフォルト値
- 権限
- 変更された値
- 確認
- 選択リスト

フィールド フックを定義するには、フィールド グリッドを使用します。



ClearQuest のフックは、ClearQuest API を使用するスクリプトを組み込むことによりカスタマイズできます。スクリプト フックの編集を終了した後、[フック] メニューの [コンパイル] を選択してコードの構文をチェックします。

詳細については、203 ページの「フィールド フックの操作」を参照してください。

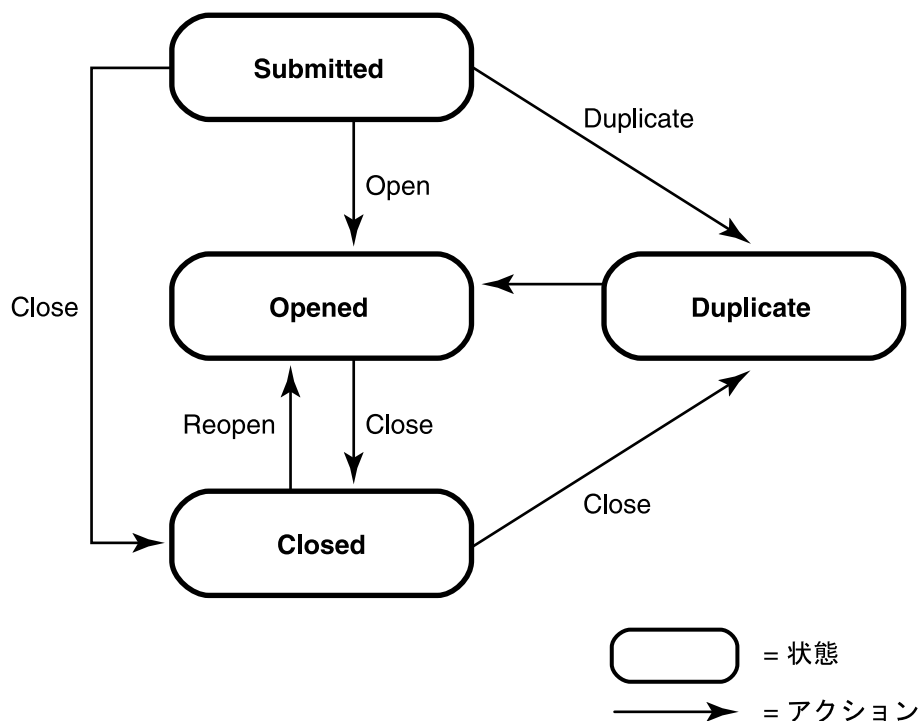
状態モデルの概要

ClearQuest では、状態モデルとは 1 つのタイプの変更依頼に使用するプロセス モデルを表したものです。状態は、Submitted、Assigned、Opened、Closed など、個々の変更依頼の状態を表します。アクションとは、ClearQuest のユーザーが変更依頼の状態を移行させるアクティビティのことです。典型的なアクションには、Submit、Assign、Modify、Close などがあります。

ある状態から別の状態に変更依頼が移行することを状態遷移と呼びます状態遷移は、遷移元の状態 (現在の状態)、遷移先の状態、レコードを遷移元の状態から遷移先の状態に移行させるアクションから構成されます。典型的なアクションには、Submit、Assign、Modify、Close などがあります。

状態モデルは、変更依頼のタイプに該当するすべての状態と状態遷移を表します。たとえば、次の状態モデルは、実行されたアクションの結果として、EnhancementRequest レコードタイプ (いくつかの事前定義されたスキーマに含まれます) がどのように別の状態に移行するかを示しています (図 5 を参照)。

図 5 Enhancement レコード タイプの状態モデル



状態モデルの設計は、特定のレコードタイプ（つまり、変更依頼の具体的な種類）が取り得るすべての状態のリストを作成し、それらの状態を記述するところから始めるのが最善です。たとえば、EnhancementRequest レコードタイプの状態は次の表のとおりです。

状態	説明
Submitted	レコード登録時の初期状態
Opened	レコードで作業中
Closed	レコードの修正確認済み
Duplicate	ほかのレコードと重複

ClearQuest では、状態モデルとアクションを変更する方法を 2 つ用意しています。1 つは状態遷移マトリックス、もう 1 つはアクショングリッドです。

状態遷移マトリックスでは、状態モデルがスプレッドシートのようなグリッドとして表されます。状態は上と横に一覧表示され、状態間の遷移を引き起こすアクションがセルに示されます。



新規状態の追加

新しい状態を追加するには、まず状態遷移マトリックスに状態を追加し、次の手順で状態遷移を作成します。

- 1 ClearQuest Designer のワークスペースで、[レコードタイプ] フォルダを展開し、目的のレコードタイプを選択します。次に、[状態とアクション] フォルダを展開し、[状態遷移マトリックス] をダブルクリックします。
- 2 [編集] メニューの [状態の追加] をクリックします。



- 3 [状態の追加] ダイアログ ボックスに、新しい状態の名前を入力し、[OK] をクリックします。

ClearQuest Designer により、新しい状態が、遷移元と遷移先として状態遷移マトリックスの行と列の見出しに自動的に追加されます。

次に、状態モデルで新しい状態の使用方法を定義する状態遷移を作成します。128 ページの「状態遷移の作成」を参照してください。

状態をスキーマに追加したら、何らかのアクションで、その状態を少なくとも 1 つの別の状態に結び付けてください。アクションによって遷移できない状態を定義すると確認エラーが発生します。

状態タイプを使用するパッケージがスキーマに含まれている場合は、新しい状態を追加する際に、その新しい状態をスキーマの状態タイプにマッピングする必要があります。122 ページの「状態タイプのマッピング」を参照してください。

状態タイプのマッピング

スキーマ パッケージによっては、変更依頼が特定の状態に入ると実行されるフック (スクリプト) がスキーマに追加されることがあります。このような種類のパッケージの例としては、UnifiedChangeManagement (UCM) パッケージと Resolution パッケージがあります。

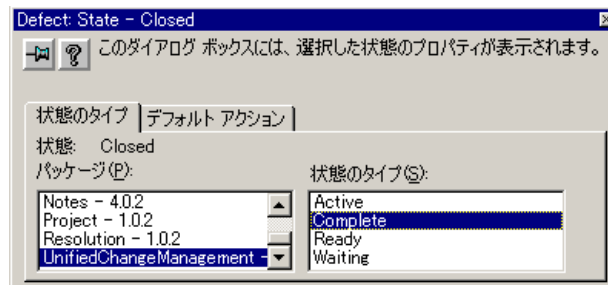
フックが適切なタイミングで実行されるようにするには、目的のレコードタイプの各状態をパッケージの状態タイプにマッピングする必要があります。パッケージの各状態タイプにレコードタイプの状態が少なくとも 1 つマッピングされている必要があります (これにより、パッケージのフックが実行される状態が少なくとも 1 つ確保されます)。複数の状態を、パッケージの状態タイプ 1 つにマッピングできます。

状態タイプを使用するスキーマに新しい状態を追加する場合は、その状態を該当するパッケージの状態タイプにマッピングする必要があります。

状態タイプをマッピングするには

- 1 ClearQuest Designer で、レコード タイプの状態遷移マトリックスを表示します。
- 2 状態を右クリックし、[プロパティ] を選択して、状態のプロパティ ダイアログ ボックスを表示します。
- 3 [状態のタイプ] タブをクリックし、状態タイプのマッピングが必要なパッケージを選択します。次に、状態タイプを選択します。
- 4 ダイアログ ボックスを閉じて状態のマッピングを保存します。
- 5 レコード タイプに存在する各々の新しい状態について、この作業を繰り返します。

たとえば、既存のスキーマの Closed 状態を UnifiedChangeManagement パッケージの Complete 状態にマッピングします。次に、Complete 状態タイプを選択し、Closed 状態にマッピングします。



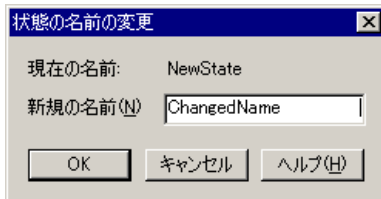
メモ: UCM スキーマまたはパッケージを使用する場合、状態にデフォルトのアクションを割り当てる必要があります。詳細については、318 ページの「UCM のデフォルト アクションの設定」を参照してください。

状態の名前の変更

状態の名前はいつでも変更できます。ClearQuest では、状態の名前を変更すると、その状態を参照するすべてのアクションの状態名情報が自動的に更新されます。

状態の名前を変更するには

- 1 ClearQuest Designer のワークスペースで、[レコード タイプ] フォルダを展開し、目的のレコード タイプを選択します。次に、[状態とアクション] フォルダを展開し、[状態遷移マトリックス] をダブルクリックします。
- 2 状態遷移マトリックスで、名前を変更する状態をクリックし、[編集] メニューの [状態の名前の変更] をクリックします。または、右クリックし、ショートカット メニューの [状態の名前の変更] をクリックします。



- 3 [状態の名前の変更] ダイアログ ボックスで、[新規の名前] に新しい名前を入力し、[OK] をクリックします。

メモ: フックが状態の名前を明示的に参照している場合は、フックのコード上で手作業で状態の名前を変更する必要があります。

状態の削除

使用する予定のない状態はすべて削除します。1 つ以上のアクションによって遷移できない状態を定義すると確認エラーが発生します。

状態を削除する前に、次の点に留意してください。

- 状態を削除する場合は、その状態を参照しているすべてのアクションを編集する必要があります。**ClearQuest** では、アクションの遷移元の状態または遷移先の状態が再割り当てされません。
- スクリプトで状態を明示的に参照している場合は、スクリプトを修正して状態に対するすべての参照を削除してください。
- 現在状態を使用しているデータベースのアップグレードを計画している場合は、その状態を削除してはなりません。状態を使用するレコードがある場合は、**ClearQuest** によって、データベースのアップグレードが禁止されます。

状態を削除するには

- 1 **ClearQuest Designer** のワークスペースで、[レコードタイプ] フォルダを展開し、目的のレコードタイプを選択します。次に、[状態とアクション] フォルダを展開し、[状態遷移マトリックス] をダブルクリックします。
- 2 状態遷移マトリックスで、削除する状態をクリックし、[編集] メニューの [状態の削除] をクリックします。または、右クリックし、ショートカットメニューの [状態の削除] をクリックします。

アクションとアクション タイプの操作

アクションとアクション タイプの理解

アクションは、ClearQuest クライアントのユーザーによって行われる、新規レコードのデータベースへの登録、レコードのある状態から別の状態への移行、レコードの修正または削除の手段です。

ClearQuest クライアントのユーザーがアクションを選択するには、フォームの [アクション] をクリックするか、ツール バーの [アクション] メニューをクリックします。どちらの場合も、有効なアクションのリストが表示されます。デフォルト アクションは太字で強調表示されます。デフォルト アクションを定義するには、状態のプロパティ ダイアログ ボックスの [デフォルト アクション] タブを使用します。130 ページの「デフォルト アクションの使用法」を参照してください。フックからデフォルト アクションを呼び出すこともできます。

アクションで、次のことを行うことができます。

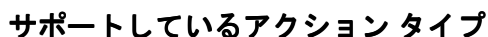
- 新規レコード (変更依頼) を作成してデータベースに追加する。
- レコード内の情報を修正する (各フィールドに関連付けられた動作は、レコードの特定のフィールドのアクセスを制限することもできます)。
- レコードをある状態から別の状態に移動する。
- あるレコードを別のレコードの重複としてマークする。
- フックを実行する。アクションフックでは、アクセス制御、初期化、確認、通知を処理できます。129 ページの「フックの追加によるアクションのカスタマイズ」を参照してください。
- レコードをデータベースから削除する。

どのユーザーがどのアクションにアクセスできるか、いつアクションを実行できるかをカスタマイズできます。

ClearQuest には、Submit アクションや Change State アクションなど、よく使用されるアクションのためにいくつかの事前定義されたアクション タイプが用意されています。126 ページの「サポートしているアクション タイプ」を参照してください。

ClearQuest Designer では、各レコードタイプに、そのタイプのレコードが使用できるアクションを定義するアクション グリッドがあります。アクション グリッドを使用して、アクションの追加、変更、削除が可能で、状態遷移も作成できます。

- 1 **ClearQuest Designer** のワークスペースで、[**レコードタイプ**] フォルダまたは [**レコードタイプ - 状態なし**] フォルダを展開し、目的のレコードタイプを展開します。
- 2 状態のあるレコードタイプの場合、[**状態とアクション**] を展開します。[**アクション**] をダブルクリックします。



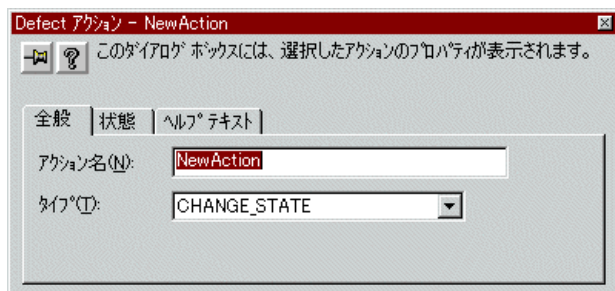
アクションタイプ	説明/コメント
Base	<p>Base アクションは、ほかのアクションの副作用として二次的に実行されるアクションです。Base アクションによってアクションフックを書くことができるのは 1 回のみですが、複数のアクションでそのアクションフックを使用できます。アクションが起動するたびに、Base アクションはフックの基準を満たしているかどうかをチェックし、基準を満たしている場合に自身のプロセスを実行します。</p> <p>たとえば、[通知] アクションフックを Base アクションに追加して、Close アクション (レコードを Closed 状態に遷移させる Change_state アクションタイプ) が起動した際に、Base アクションによって自動的に電子メールの通知を送信することができます。</p> <p>ClearQuest クライアントのアクション リストに、Base アクションは表示されません。</p>

アクションタイプ	説明/コメント
Change_state	<p>Change_state アクションは、状態のあるレコードタイプでのみ使用可能です。レコードを遷移元の状態から遷移先の状態に移行させます。多くの遷移元の状態を参照できますが、参照できる遷移先の状態は1つのみです。</p> <p>Change_state アクションは、現在のレコードが遷移元の状態の1つであれば、ClearQuest クライアントのアクションリストに表示されます。</p>
Delete	<p>Delete タイプでは、ユーザーはデータベースからレコードを削除できます。Delete アクションは、ClearQuest クライアントのアクションリストに表示されます。</p>
Duplicate	<p>状態のあるレコードタイプでのみ使用可能です。Duplicate タイプのアクションはレコードを、類似情報を持つほかのレコードにリンクします。</p> <p>Duplicate アクションは、現在のレコードが遷移元の状態の1つであれば、ClearQuest クライアントのアクションリストに表示されます。</p>
Import	<p>レコードを別のソースから ClearQuest にインポートします。</p> <p>インポートの際に、ClearQuest はインポートされたレコードの内容を確認します。ただし、ClearQuest は、インポートの際にフィールドレベルの確認は行いません。さらに、状態のあるレコードのセットがインポートされると、ClearQuest は、その状態に有効に遷移したかどうかを確認することなく、インポートされたレコードをデータファイルで指定した状態に割り当てます。</p> <p>Import タイプのアクションは、ClearQuest クライアントのアクションリストに表示されません。</p>
Modify	<p>レコードの状態を移行させずにレコードのフィールドの値を修正します。Modify アクションは、ClearQuest クライアントのアクションリストに表示されます。</p>
Record_script_alias	<p>アクションをレコードスクリプトと関連付けます。</p> <p>Record_script_alias アクションは、ClearQuest クライアントのアクションリストに表示されます。</p>
Submit	<p>新規レコードを ClearQuest ユーザー データベースに入力します。</p> <p>状態のあるレコードには、遷移先の状態を割り当てますが、遷移元は必要ありません。各レコードタイプは、タイプが Submit のアクションを1つのみ持つことができます。</p>
Unduplicate	<p>状態のあるレコードタイプでのみ使用可能です。重複するレコード間のリンクを削除します。</p>

新規アクションの追加

新規アクションを追加するには

- 1 ClearQuest Designer で、アクション グリッドを表示します。
- 2 [編集] メニューの [アクションの追加] をクリックします。
- 3 [アクションのプロパティ] ダイアログ ボックスの [全般] タブで、[アクション名] を入力します。

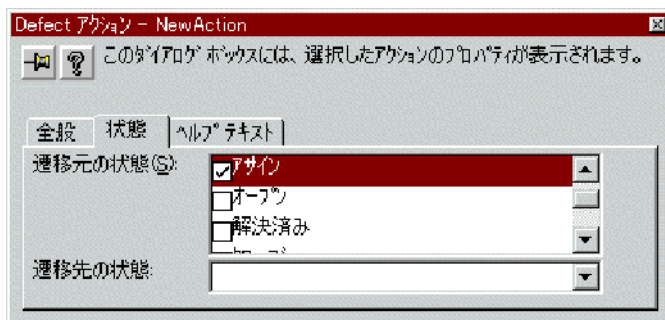


- 4 アクション タイプを選択します。このアクションを使用して状態遷移を実行する予定がある場合は、タイプとして **CHANGE_STATE** を選択します。

状態遷移の作成

ClearQuest の管理者は、ユーザーがレコードの状態を移行させるためのルールを定義します。状態遷移は、アクションによって実行されます。状態遷移を作成するには、**CHANGE_STATE** タイプのアクションを定義し、次にそのアクションの遷移元の状態と遷移先の状態を選択します。

- 1 ClearQuest Designer のワークスペースで、アクション グリッドを開きます。
[レコード タイプ] フォルダを展開し、目的のレコード タイプを選択します。次に、[状態とアクション] フォルダを展開し、[アクション] をダブルクリックします。
- 2 新規アクションを作成し、タイプとして **CHANGE_STATE** を選択します。
または、**CHANGE_STATE** タイプの既存のアクションを右クリックし、[アクションのプロパティ] を選択します。
- 3 [アクションのプロパティ] ダイアログ ボックスの [状態] タブで、アクションに対して遷移元の状態 (1 つ以上) と遷移先の状態を選択します。



CHANGE_STATE の各アクションは、少なくとも各々 1 つの遷移元の状態と遷移先の状態を持つ必要があります。1 つも定義されていない場合は、確認の際に ClearQuest からエラーが報告されます。

- 4 状態遷移を作成したら、状態遷移マトリックスを表示して ClearQuest によってその遷移が状態に適用されたことを確認します。121 ページの「状態遷移マトリックスの表示」を参照してください。

アクションの修正

アクションを修正するには

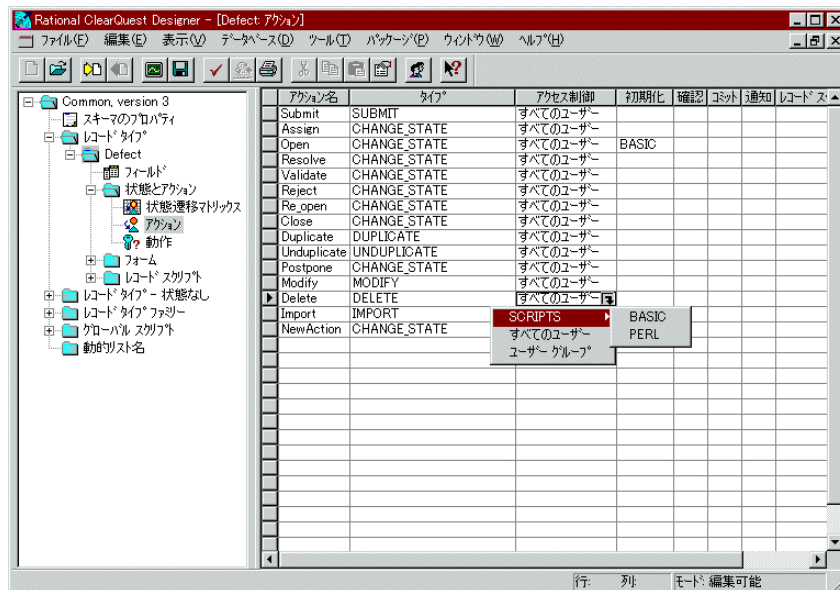
- 1 ClearQuest Designer のワークスペースで、[レコードタイプ] フォルダまたは [レコードタイプ - 状態なし] フォルダを展開します。
- 2 目的のレコードタイプを展開します。
- 3 状態のあるレコードタイプの場合は、[状態とアクション] フォルダを展開します。
[アクション] をダブルクリックします。

フックの追加によるアクションのカスタマイズ

レコードライフでの主要ポイントでタスクを実装するアクションフックを追加できます。たとえば、ClearQuest は、デフォルトで、すべてのユーザーにすべてのアクションへのアクセスを許可します。アクセス制御フックを使用して、アクションへのアクセスを制限できます。

ClearQuest には、アクセス制御、初期化、確認、コミット、通知というアクションフックがあります。

アクションフックを定義するには、アクション グリッドを使用します。



ClearQuest API を使用するスクリプトを組み込むことにより、ClearQuest のアクションフックをカスタマイズできます。スクリプト フックの編集を終了したら、[フック] メニューの [コンパイル] を選択してコードの構文をチェックします。

アクション フックに関する詳しい説明とフィールドフックとの関連については、「第 10 章 ワークフローのカスタマイズにおけるフックの使用法」を参照してください。アクセス制御アクション フックの作成方法については、340 ページの「アクション アクセス制御フックの例」を参照してください。また、87 ページの「スクリプト言語の選択」も参照してください。

デフォルト アクションの使用法

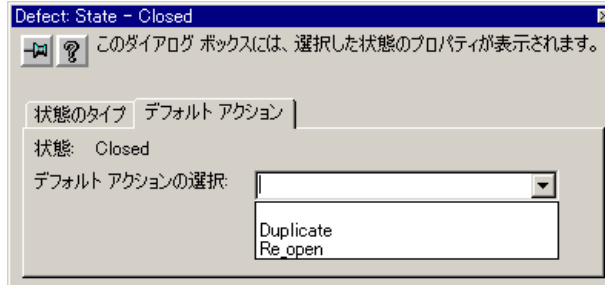
状態のデフォルト アクションを定義できます。状態のデフォルト アクションは、ClearQuest クライアントの [アクション] メニューに太字で表示されます。デフォルト アクションには、次の特徴があります。

- ユーザーに対する状態モデルのガイドとして役に立ちます。
- UCM スキーマやパッケージなど、特定のスキーマとパッケージに必要です。UCM スキーマやパッケージを使用する場合は、状態のデフォルト アクションが、状態タイプ モデルで有効なパスを提供する必要があります。詳細については、318 ページの「UCM のデフォルト アクションの設定」を参照してください。
- フック コードから呼び出すことができます。

メモ: 状態のデフォルト アクションを定義するには、先に状態遷移を作成する必要があります。

状態のデフォルト アクションを定義するには

- 1 ClearQuest Designer のワークスペースで、レコード タイプの状態遷移マトリックスを表示します。
- 2 状態遷移マトリックスで、状態を右クリックし、ショートカット メニューの [プロパティ] をクリックします。
- 3 状態のプロパティ ダイアログ ボックスの [デフォルト アクション] タブで、その状態のデフォルト アクションを選択します。



アクションの削除

アクションの削除の際に、スキーマの変更が必要な場合があります。たとえば、CHANGE_STATE アクションを削除する場合は、アクションの削除に対応するよう状態遷移マトリックスを変更する必要があります。

アクションを削除するには

- 1 ClearQuest Designer のワークスペースで、アクション グリッドを開きます。
[レコードタイプ] フォルダを展開し、目的のレコードタイプを選択します。次に、[状態とアクション] フォルダを選択し、[アクション] をダブルクリックします。
[レコードタイプ - 状態なし] フォルダを展開し、目的のレコードタイプを選択します。次に、[アクション] をダブルクリックします。
- 2 削除するアクションの行を選択します。
- 3 [編集] メニューの [アクションの削除] をクリックします。または、右クリックし、ショートカット メニューの [アクションの削除] をクリックします。

メモ: 削除したアクションをスクリプトで明示的に参照している場合、スクリプトを修正して、削除したアクションへのすべての参照も削除する必要があります。

状態モデルを作成する上での考慮事項

状態モデルの設計と管理には十分な配慮が必要です。

ブラックホール

モデルにブラックホールがないことを確認してください。ブラックホールとは、変更依頼が入り込み、そのまま何もされずに見落とされる可能性がある場所のことです。たとえば、Postponed という状態が存在するのに、この状態にあるレコードの処理が誰にもアサインされていないと、通知も修正もされない変更依頼が蓄積します。

こうしたブラックホールの原因は、往々にして些細なものです。たとえば、Open 状態の変更依頼の処理に 3 人のエンジニアがアサインされており、コンポーネント フィールドの値はそれぞれ赤、緑、青であるとしめます。この場合、コンポーネント フィールドが黄である変更依頼が登録されると、その変更依頼は解決されずに残ります。

状態の数

状態の数を少なくして開発アクティビティを多くするか、状態の数を多くして開発アクティビティの数を減らすかには、検討すべきトレードオフがあります。たとえば、変更依頼のライフサイクルのいくつかのポイントで何らかの確認アクティビティが発生する場合、確認アクティビティを各状態の一部として含めるのではなく、確認という独立した状態として作成することを検討してください。これにより、確認処理が適切に行われたかどうかを知ることが簡単になります。一方、状態の数を多くしすぎると、モデルが扱いにくくなり、保守が難しくなります。

Duplicate

Duplicate アクションはとても便利です。データベース内で重複しているレコードの 1 つをアクティブ レコードとして認識し、それ以外のすべてのレコードを複製としてマークして、修正されるのを防ぎます。これにより、共通の問題に対するすべての操作がデータベース内の 1 つのレコードを通じて調整されます。この機能は組み込みフィールドを使用します。また、この機能をスキーマに追加するには、フォームに重複先コントロールと重複元コントロールを追加し、Duplicate アクションと Unduplicate アクションを作成します。さらに、この機能が持つその他の特長としては、Unduplicate によって、レコードが複製として指定された時点の状態に戻されることが挙げられます。

事前定義のスキーマ

ClearQuest には、さまざまな事前定義されたスキーマが用意されており、それぞれが具体的な機能や設定をカバーしています。カスタム スキーマを作成する場合、これらのスキーマをたたき台に使用できますが、その際、事前定義されたスキーマの機能と目的の要件との違いを十分に比較検討してください。

事前定義されたスキーマを流用する際に考慮すべき重要な点を次に示します。

- スキーマを最初から設計する場合は、**Blank** スキーマをベースとして使用してください。
- **Common** スキーマでは、**Defect** レコード タイプを 1 つと、有用な (場合によっては不要な) 数多くの機能を定義しています。
- **DefectTracking** スキーマでは、データベース内でのレコードの相互参照機能と ClearQuest クライアントのナビゲーション機能のデモを示します。このスキーマは、**Common** スキーマがベースで、**Defect** レコード タイプからは **Reference_List** フィールドを持つ、**Customer** という状態なしレコード タイプが追加されています。
- 事前定義された **Studio** スキーマである **Analyst**、**Development**、**Test** スキーマは、**Common** スキーマがベースで、対応する **Rational Suite** 製品との統合に利用できます。
- **UnifiedChangeManagement** スキーマも **Common** スキーマがベースで、**UCM ClearCase** 統合用に拡張されています。
- **EnterpriseStudio** スキーマは、**Common** スキーマがベースで、**UCM** との統合のみでなく、すべての **Rational Suite** 統合に利用できます。

並行開発

よく使用される成果物を共有する複数の製品 (または製品バージョン) を並行開発する場合、それを支援するスキーマの設計はとても困難です。

このようなケースでは、「共有する成果物に原因があり、その修正に複数の製品がかかわるような障害をレポートする場合に、システムでこの情報をどのようにして収集し、処理するか」、「(スケジュールが異なる可能性がある) 複数のビルドを作成する必要がある場合に、障害の現在の状態をシステムでどのように追跡するか」というような疑問が浮かびます。

このような問題に対し、単一レコードでさまざまな作業を追跡するのでは無理があります。

1 つの方法として、影響を受ける各製品に複数のレコードを登録するという方法があります。この場合は、各アクティビティの状態を個別に追跡できます。入力された最初のレコードに **Save Default Values** メカニズムを使用し、それ以降のレコードに **Load** を使用すると、各コピーに対してユーザーがデータ エントリの複製を作成せずに済みます。ただし、この機能は **Web** クライアントでは使用できません。

この方法では、各レコードがほかのレコードから隔離されていて、別の関連する問題に対して何らかの作業が行われていることがわからないことから、無駄な労力を費やす可能性があることが欠点です。

より好ましい方法は、階層構造を使用することです。この場合は、親で問題の特徴付け、子でそれぞれの製品、種別、バージョンで共通の問題を追跡します。レコードタイプは、親と子で異なっていてもかまいません。

スキーマ間で同じレコードタイプを使用して、すべての情報を子に格納するようにできます(これにより、親子間の移動を省きます)。また、形式的に子であるレコードタイプを通知目的で使用して、影響を受けるほかの製品、種別、バージョンに同じ問題を通知し、その状態を追跡することもできます。

Rational ClearQuest パッケージを使用すると、効率的な方法で、新機能をスキーマに追加したり、スキーマをほかのソフトウェア パッケージと統合することができます。

この章の内容は、次のとおりです。

- 135 ページの「パッケージと統合の概要」
- 137 ページの「パッケージを適用する際の注意事項」
- 138 ページの「使用可能なパッケージ」
- 139 ページの「例: Notes パッケージ、Attachments パッケージ、Email パッケージ」
- 140 ページの「スキーマに適用したパッケージの確認」
- 141 ページの「パッケージの適用方法」
- 146 ページの「パッケージのアップグレード方法」

パッケージと統合の概要

Rational ClearQuest パッケージは、スキーマ コンポーネントのセットです。既存のスキーマに追加して、複雑な機能を提供できます。提供した新機能は、スキーマに永久的に組み込まれます。

パッケージを使用して、多くの ClearQuest ユーザーにとって有用な機能を追加できます。たとえば、障害レコードにメモのリストを付加する機能や、特定のイベントが発生したときに自動的に電子メール通知を送信する機能などを追加できます。

また、パッケージを使用して、ClearQuest のデータをほかの Rational ソフトウェア製品とサードパーティのソフトウェア パッケージ (Microsoft Visual Source Safe (VSS) や Microsoft Project など) と交換することもできます。このようなパッケージを、統合パッケージまたは統合と呼びます。

パッケージを使用して追加できるスキーマ コンポーネントには、次のものがあります。

- レコード タイプ
- フォーム機能 (タブや新規フォームを含む)
- フィールド定義と動作
- フック コードとグローバル スクリプト
- 動的リスト名
- スキーマ確認テスト
- クエリー、グラフ、レポート

1 つのスキーマに複数のパッケージを適用できます。

通常は、1 回の操作で 1 つのパッケージを自動的に適用できます。また、パッケージのインストール時にパッケージのカスタマイズまたは設定 (フィールド名の変更など) ができる場合があります。パッケージの設定が必要な場合もあります。たとえば、統合パッケージによっては、既存のスキーマの状態タイプをパッケージ内の状態タイプにマッピングする必要があります。カスタマイズの詳細については、『Rational Suite アップグレード ガイド』の 287 ページの「ClearQuest のスキーマとパッケージ」を参照してください。

パッケージを初めてインストールする場合は、ClearQuest Designer のパッケージ ウィザードを使用すると、新しいパッケージの選択とインストールが簡単にできます。詳細については、141 ページの「パッケージの適用方法」を参照してください。コマンド行でスクリプトを実行して CharacterSetValidation パッケージを適用することもできます。詳細については、39 ページの「CharacterSetValidation パッケージの概要」を参照してください。

パッケージに含まれている機能をユーザーに対して有効にするには、スキーマに対するその他の変更と同じように、パッケージをスキーマに適用し、スキーマの新しいバージョンを使用するようにユーザー データベースをアップグレードする必要があります。

インストールしたパッケージの開発または管理について心配する必要はありません。Rational では、新しいバージョンの提供と、既存パッケージの変更管理のサポートを行っています。ClearQuest の新しいバージョンにアップグレードするときは、[インストール済みパッケージのアップグレード] 機能を使用して、対象のスキーマ用にインストールされているすべてのパッケージを自動的にアップグレードします。詳細については、146 ページの「パッケージのアップグレード方法」を参照してください。

メモ: パッケージ ウィザードと [インストール済みパッケージのアップグレード] 機能は、Rational が提供したパッケージにしか使用できません。

パッケージと事前定義されたスキーマ

事前定義されたスキーマとは、Common スキーマと、パッケージのセットから作成されたスキーマのことです。

たとえば、DefectTracking スキーマは、以下のパッケージの適用後に Common スキーマから派生します。

- Notes
- Resolution
- Attachments
- History
- Email
- Customer

パッケージを適用する際の注意事項

パッケージをスキーマに追加した後に、スキーマのそのバージョンからパッケージを削除することはできません。パッケージを適用する前に、内容を調べて、要求に合ったパッケージかどうかを確認してください。

また、パッケージを適用してスキーマを修正したときは、修正したスキーマに対して実動ユーザー データベースをアップグレードする前に、テスト データベースを使用して、そのスキーマが想定どおりに動作するかどうかを確認してください。

修正したスキーマが想定どおりに動作しない場合は、パッケージを適用したスキーマ バージョンを削除できます。ただし、実動ユーザー データベースをこのバージョンにアップグレードした後は削除できません。

パッケージのコンポーネントには、パッケージの適用後は修正できないものや、特別な状況で慎重に修正する場合以外は修正するべきでないものがあります。これは、ClearQuest または ClearQuest との統合が、特定の設定またはコンポーネントの特定の名前があることを予期する場合があるためです。たとえば、見出しというフィールド名を変更すると失敗する統合パッケージがあります。また、変更アクションを名前変更または削除すると失敗する統合パッケージもあります。

この問題を回避するために、パッケージを使用してインストールするコンポーネントの一部は読み取り専用になっています。こうしたコンポーネントは、名前を変更したり機能を拡張したりして修正することはできません。その他のコンポーネントは修正可能ですが、必要不可欠でないかぎり修正しないでください。このタイプの依存の詳細については、289 ページの「ClearQuest のパッケージ」を参照してください。

使用可能なパッケージ

ClearQuest で現在使用可能なパッケージの概要については、287 ページの「ClearQuest のスキーマとパッケージ」を参照してください。この付録には、各パッケージの名前、説明、追加または修正される機能の概要、修正されるフィールドのリストがあります。

表 14 に、ClearQuest で現在使用可能なパッケージをカテゴリ別に示します。

表 14 ClearQuest パッケージの概要

パッケージタイプ	パッケージ名
全般	<ul style="list-style-type: none">▪ Attachments▪ Email▪ History▪ Notes▪ Resolution
障害の追跡	<ul style="list-style-type: none">▪ Project▪ Customer
データ/コード ページ確認	CharacterSetValidation
UCM 統合	<ul style="list-style-type: none">▪ AMStatesTypes▪ BaseCMActivity▪ UCMPolicyScripts▪ UnifiedChangeManagement
Rational Suite 統合	<ul style="list-style-type: none">▪ Repository (Rational リポジトリ)▪ PQC (Rational Development Studio と TestStudio の Purify、Visual Quantify、PureCoverage)▪ TeamTest▪ RequisitePro▪ RequisiteProSupplement▪ EnhancementRequest▪ ContentStudio
構成管理 (CM) 統合	<ul style="list-style-type: none">▪ ClearCase▪ VisualSourceSafe
Rational ClearCase Project Tracker	<ul style="list-style-type: none">▪ AMBaseActivity▪ AMWorkActivitySchedule

Repository パッケージでは、Rational Suite に含まれる Rational リポジトリの統合された機能をサポートします。PQC パッケージでは、Rational Development Studio と TestStudio に含まれる Purify、Visual Quantify、PureCoverage の統合された機能をサポートします。TeamTest パッケージでは、Rational TeamTest に含まれる統合された機能をサポートします。RequisitePro パッケージと RequisiteProSupplement パッケージでは、Rational Suite 製品に含まれる Rational RequisitePro の統合された機能をサポートします。Enhancement Request パッケージでは、以前の障害レコード タイプに加えて新しい、状態のあるレコード タイプを導入して、障害と変更の追跡プロセス用のアプリケーションを幅広く提供します。ContentStudio パッケージでは、Rational ContentStudio との統合をサポートします。

例：Notes パッケージ、Attachments パッケージ、Email パッケージ

最もよく使用されるパッケージが、Notes、Attachments、Email です。

Notes パッケージを適用すると、レコード タイプのメモをログに記録できるようになります。メモのログは一般に、ClearQuest ユーザーからの変更依頼に関する進行中のコメントを取得するために使用されます。このコメントには、変更依頼の分析または解明、推奨の解決法、変更依頼の優先度または重要度に関するコメント、その他の有用な観測結果などが記述されます。

Notes パッケージをスキーマに適用するときは、1 つまたは複数のレコード タイプ (Defects と EnhancementRequests など) を選択します。Notes パッケージによって、[メモ] タブが追加されます。このタブには、選択した各レコード タイプ用にコントロールが 2 つずつあります。1 つは新しいメモを入力するためのコントロールで、もう 1 つはログに取得したすべてのメモを表示するためのコントロールです。ClearQuest ユーザーが 2 つめのコントロールをクリックすると、ClearQuest から既存のメモが表示されます。メモはメモ ヘッダーで区切られています。メモ ヘッダーには、そのメモの状態、ユーザー名、日付、時刻が含まれます。

Attachments パッケージを適用すると、レコードに関連する添付ファイルをユーザーが追加または削除できるようになります。Attachments パッケージをスキーマに適用するときは、1 つまたは複数のレコード タイプを選択します。このパッケージによって、選択した各レコード タイプに [添付] タブが追加されます。

Email パッケージを適用すると、特定のアクションが実行されたときに電子メール メッセージによる通知をチーム メンバーに自動送信できるようになります。このパッケージでは、EmailRules 状態なしレコード タイプが追加されます。このレコード タイプでは、クライアントでインスタンスが生成された場合に、ユーザーが電子メール通知の生成条件、受信者、内容を定義することができます。複数の EmailRules レコードを作成すると、電子メール通知を自動生成する条件を複数定義できます。電子メール通知は、異なるユーザーまたはユーザー グループに送信できます。

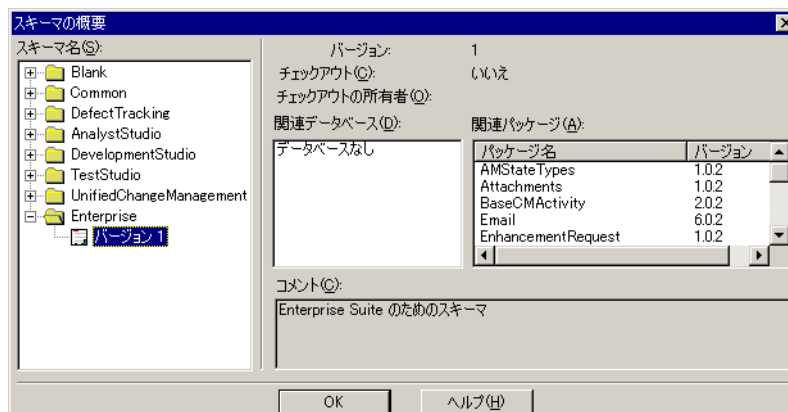
この例では、パッケージを適用して設定するときには判断が必要な理由についても説明します。たとえば、Attachments パッケージを ClearQuest Windows Web クライアントで展開する場合、サイズの大きい添付ファイルをアップロードまたはダウンロードすると、ボトルネックが発生し、ほかの ClearQuest Web ユーザーへの応答時間が長くなります。また、EmailRules はレコードがデータベースに書き込まれるたびに評価されるので、EmailRules が多すぎると、ユーザーが [OK] または [適用] をクリックしたときの応答時間に影響します。

こうした問題を減らすには、カスタム フックを作成します。たとえば、添付ファイルの最大サイズを制限するカスタム フックや、カスタム電子メール通知フックを作成します。このような解決法の詳細については、Rational Web サイトの Rational Developer Network を参照してください。ただし、英語のみのご利用となります。

スキーマに適用したパッケージの確認

新しいパッケージまたはパッケージ バージョンを適用する前に、既存のスキーマに既に適用されているパッケージ バージョンを確認できます。

- 1 ClearQuest で、[表示] メニューの [スキーマの概要] をクリックします。ClearQuest に、[スキーマの概要] ダイアログ ボックスが表示されます。



- 2 スキーマ (DefectTracking など) を選択し、フォルダを展開して、使用可能なバージョンを表示します。[スキーマの概要] ダイアログ ボックスに [関連パッケージ] ペインが開き、そのスキーマに適用されているパッケージと関連バージョンが一覧表示されます。

パッケージの適用方法

パッケージをスキーマに適用するには、パッケージを選択して [終了] をクリックするだけの簡単な方法と、次の 4 つの手順を実行する方法があります。

- スキーマを選択します。
- 適用するパッケージを選択します。
- パッケージを使用して変更するスキーマのレコードタイプを選択します。
- スキーマの既存の状態タイプをパッケージの状態にマッピングします。

最後の 2 つの手順は、パッケージ ウィザードでパッケージを適用する場合に実行できます。また、後で ClearQuest Designer のメイン ウィンドウから実行することもできます。

処理手順の順序は、パッケージ ウィザードを起動してから変更するスキーマを選択するか、スキーマをチェックアウトしてからパッケージ ウィザードを起動するかによって異なります。

以降で説明する処理の順序は、先にパッケージ ウィザードを起動した場合を前提にしています。また、すべての処理を実行する必要があるかどうか、選択したパッケージとスキーマによって異なります。

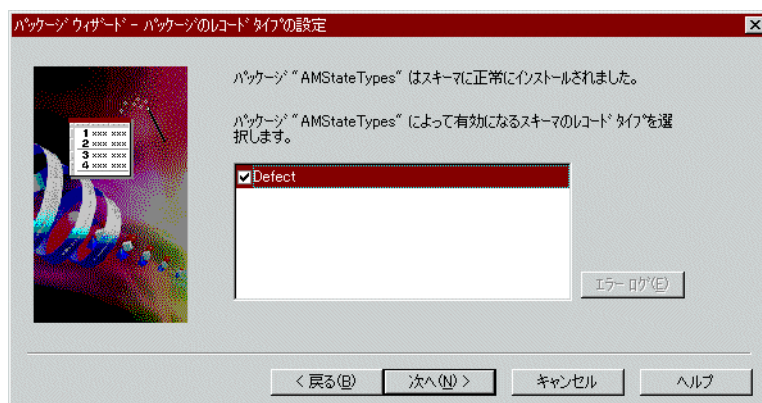
メモ: コマンド行でスクリプトを実行して CharacterSetValidation パッケージを適用することはできません。詳細については、39 ページの「CharacterSetValidation パッケージの概要」を参照してください。

パッケージ ウィザードを起動してパッケージを適用するには、ClearQuest Designer で次の手順に従います。

- 1 [パッケージ] メニューの [パッケージ ウィザード] をクリックして、パッケージ ウィザードを起動します。最新のパッケージ バージョンが表示されます。



- 2 目的のパッケージ バージョンが表示されていない場合は、[その他のパッケージ] をクリックします。このリストからバージョンを選択し、[OK] をクリックすると、元の [パッケージ ウィザード - パッケージのインストール/選択] ダイアログ ボックスにそのパッケージが表示されます。
- 3 [パッケージ ウィザード - パッケージのインストール/選択] ダイアログ ボックスで、適用するパッケージを選択し、[OK] をクリックします。
- 4 ウィザードの 2 番目の画面で、パッケージを使用して変更するスキーマを選択します。
選択したパッケージがスキーマに適用可能かどうかを検証されます。パッケージをスキーマに適用できない場合は、エラー メッセージが表示されます。たとえば、特定のフィールドを修正するパッケージの場合、そのフィールドがスキーマに存在していないと、**ClearQuest** ではパッケージのいずれのコンポーネントもそのスキーマに適用することはできません。
次に、選択したパッケージをスキーマに適用した場合に、スキーマ コンポーネントに影響しないかどうか判断されます。影響がないと判断された場合は、[終了] をクリックして処理を完了できます。影響があると判断された場合は、[次へ] をクリックして、次の画面に進んでください。
- 5 パッケージの実装を正常に行うためにスキーマのレコード タイプの変更が必要だと判断された場合は、[パッケージ ウィザード - パッケージのレコード タイプの設定] ダイアログ ボックスが表示されます。

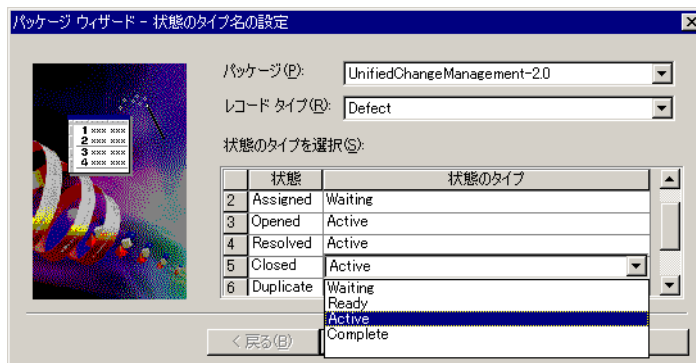


このダイアログ ボックスには、パッケージで (フィールド、アクション、スクリプトの追加などによって) 有効化または修正される可能性のあるスキーマのレコード タイプのリストが表示されます。既存のレコード タイプを有効にするためのカスタマイズ情報がパッケージの中にある場合、このリストは空です。

パッケージを使用して有効にするレコード タイプをオンにします。[終了] または [次へ] をクリックします。

メモ: この手順は、必要に応じてスキップできます。スキップするには、すべての項目をオフにし、[終了]または[次へ]をクリックします。後でパッケージウィザードでレコードタイプを有効にできます。そのときは、[パッケージ]メニューの[パッケージのレコードタイプの設定]をクリックします。詳細については、144 ページの「レコードタイプの有効化」を参照してください。

- 6 状態タイプを使用するパッケージを追加する場合は、[パッケージ ウィザード - 状態のタイプ名の設定] ダイアログ ボックスが表示されます。



ウィザードのこの画面では、有効化するために選択した各レコードタイプに状態タイプを割り当てます。

パッケージの各状態に対して、[状態のタイプ] フィールドをクリックし、既存のスキーマからその状態にマッピングする適切な状態タイプを選択します。既存の各状態を、パッケージの状態タイプにマッピングする必要があります。複数の状態を、1つの状態タイプにマッピングできます。各状態タイプには少なくとも1つの状態が必要です。終了したら、[終了]をクリックします。

メモ: この手順は、必要に応じてスキップできます。スキップするには、[終了]をクリックします。後で状態タイプを設定できます。そのときは、[パッケージ]メニューの[状態のタイプ名の設定]をクリックします。詳細については、144 ページの「状態タイプのマッピング」を参照してください。

先にパッケージウィザードを起動するのではなく、先にスキーマをチェックアウトしてパッケージを適用するには、既に説明した手順に従います。ただし、最初の[ファイル]メニューの[スキーマを開く]をクリックし、パッケージに適用するスキーマとバージョンを選択する手順は除きます。手順1から手順3を実行して、パッケージウィザードを起動し、適用するパッケージを選択します。必要に応じて、手順5と手順6を実行して、レコードタイプを有効化し、状態に状態タイプを割り当てます。次に、変更したスキーマの確認とチェックインを行います。

必要に応じて、手順5と手順6(レコードタイプの有効化と状態タイプのマッピング)をスキップし、これらの処理を後でClearQuest Designerで行います。

レコード タイプの有効化

パッケージを設定してスキーマに既に存在するレコード タイプを有効にする場合、ユーザーは既存のレコード タイプの中で変更するタイプと変更しないタイプを指定できます。たとえば、新しいフィールドを既存のレコード タイプに追加するようにパッケージを設定した場合、それらの新しいフィールドを一部のレコード タイプに追加しないように指定できます。ほかに適用するまたは適用しない変更のタイプには、アクションとスクリプトの追加が含まれます。

レコード タイプの変更を含むパッケージの詳細については、289 ページの「ClearQuest のパッケージ」を参照してください。

ClearQuest Designer でレコード タイプを有効にするには

- 1 [パッケージ] メニューの [パッケージのレコード タイプの設定] をクリックします。
[パッケージ ウィザード - パッケージのレコード タイプの設定] ダイアログ ボックスが表示されます。
- 2 パッケージを使用して有効にするレコード タイプをオンにし、[終了] をクリックします。

状態タイプのマッピング

パッケージには、UnifiedChangeManagement パッケージや Resolution パッケージのように、状態タイプを持つ状態モデルが含まれているものがあります。こうしたパッケージには、これらの状態のいずれかを変更依頼によって入力したときに実行されるフックがあります。

この種のパッケージを既存のスキーマに適用したときに、パッケージに含まれている状態に遷移する方法が既存のスキーマの状態モデルに備わっていないと、問題が発生します。たとえば、変更依頼で Ready 状態を入力したときにのみ実行される機能がパッケージに含まれていても、既存のスキーマに Ready 状態がない場合は、新機能が実行されることはなく、ユーザーはこの新機能を使用することができません。

この問題を回避するために、適用するパッケージに状態タイプが含まれている場合は、既存のスキーマに含まれている状態をパッケージの状態タイプにマッピングしてください。

パッケージの各状態タイプには、既存のスキーマの状態を最低でも 1 つはマッピングする必要があります。こうすることにより、パッケージを適用して作成される新規スキーマで、パッケージに含まれるすべての状態タイプに遷移できるようになります。

たとえば、変更依頼で Ready 状態を入力したときに実行される機能がパッケージに含まれており、既存のスキーマの Assigned 状態をパッケージの Ready 状態タイプにマッピングした場合は、変更依頼で Assigned 状態を入力するとこの新機能が実行されます。

ClearQuest Designer を使用して状態タイプをマッピングするには

- 1 状態タイプを使用するパッケージをスキーマに適用するには、[パッケージ] メニューの [状態のタイプ名の設定] をクリックします。[パッケージ ウィザード - 状態のタイプ名の設定] ダイアログ ボックスが表示されます。レコードタイプを選択します。
- 2 パッケージの各状態に対して、[状態のタイプ] フィールドをクリックし、既存のスキーマからその状態にマッピングする適切な状態タイプを選択します。
- 3 [レコードタイプ] フィールドに複数のレコードタイプが表示される場合は、追加分のレコードタイプをそれぞれ選択します。次に、既存のスキーマの状態タイプを、そのレコードタイプに関連付けられているパッケージの状態にマッピングします。
- 4 [終了] をクリックします。
- 5 パッケージ ウィザードの最後のステップが完了したら、[ファイル] メニューの [確認] をクリックします。変更したスキーマが確認されます。

デフォルト アクションの作成

状態のデフォルト アクションを定義できます。デフォルト アクションを設定すると、遷移元の状態から遷移先の状態に移行する方法が常に確保されます。またデフォルト アクションは、ユーザーに対する状態モデルのガイドとして役に立ちます。状態のデフォルト アクションは、ClearQuest クライアントの [アクション] メニューに太字で表示されます。

UCM のスキーマやパッケージなどの特定のスキーマとパッケージは、デフォルト アクションに必要です。UCM のスキーマやパッケージを使用する場合は、状態のデフォルト アクションによって、状態タイプ モデルにおける有効なパスを提供する必要があります。

メモ: 状態のデフォルト アクションを定義するには、先に状態遷移を作成する必要があります。128 ページの「状態遷移の作成」を参照してください。

状態に対してデフォルト アクションを割り当てるには

- 1 ClearQuest Designer で、[レコードタイプ] を展開し、パッケージに対して有効になっているレコードタイプを展開します。
- 2 そのレコードタイプの [状態遷移マトリックス] をダブルクリックします。
- 3 [状態遷移マトリックス] の最初の列で、状態を右クリックします。[プロパティ] をクリックして、その状態の [プロパティ] ダイアログ ボックスを表示します。
- 4 [プロパティ] ダイアログ ボックスの [デフォルト アクション] タブで、状態のデフォルト アクションを選択します。状態遷移マトリックスで状態遷移に作成したアクションが [デフォルト アクション] タブに一覧表示されます。

パッケージのアップグレード方法

Rational では、定期的にパッケージの新しいバージョンを提供しています。新しいバージョンには、ClearQuest のユーザーから要望のあった新機能が含まれています。

既にパッケージをスキーマに適用した場合は、スキーマをそのパッケージの新しいバージョンにアップグレードすることにより、この新機能を使用できます。

スキーマをパッケージの新しいバージョンにアップグレードするには

- 1 ClearQuest で、[パッケージ] メニューの [インストール済みパッケージのアップグレード] をクリックします。ClearQuest Designer に、[インストール済みパッケージのアップグレード - アップグレードするスキーマの選択] ダイアログ ボックスが表示されます。
- 2 アップグレードするスキーマを選択します。
スキーマがチェックアウトされ、アップグレードが必要なパッケージが分析されて、これらのパッケージに対する前提条件があるかどうかが判定されます。
- 3 [ステータス] 画面で、アップグレードが必要なパッケージのリストを確認します。
- 4 ClearQuest で、パッケージがスキーマに正しい順序で適用され、ステータスが表示されます。終了したら、ステータス メッセージを読んで、パッケージのアップグレードが正常に完了したことを確認します。
- 5 [ファイル] メニューの [作業内容のテスト] をクリックして、変更を確認します。
- 6 テストが終了したら、[ファイル] メニューの [チェックイン] をクリックして、スキーマをチェックインします。
- 7 新しいスキーマ バージョンをチェックインしたら、そのスキーマの以前のバージョンを使用するユーザー データベースをアップグレードします。

この手順は、スキーマ リポジトリにあるほかのスキーマにも繰り返すことができます。

以上の手順の完了後、新しいパッケージのバージョンが適用されていることを確認できます。140 ページの「スキーマに適用したパッケージの確認」を参照してください。

Rational ClearQuest Designer では、グラフィカルな機能を使用して、レコード フォームの作成や変更を行うことができます。

この章の内容は、次のとおりです。

- 147 ページの「フォーム操作の概要」
- 148 ページの「フォームの作成と変更」
- 151 ページの「フォーム コントロールの操作」
- 162 ページの「フォーム上のタブの操作」
- 165 ページの「レコード フォームの再使用」

各コントロール タイプのプロパティの詳細については、「付録 D フォーム コントロール」を参照してください。

フォーム操作の概要

各レコード タイプは、レコード フォームと登録フォームという 2 つのフォームに関連付けることができます。レコード フォームは各レコード タイプに対して作成する必要がありますが、登録タイプは必要に応じて作成します。登録フォームを使用すると、そのレコード タイプのレコードを登録するときに別のフォームを使用することができます。Rational ClearQuest では、新しいレコードを登録するときは、最初に登録フォームが表示されます。その後、ユーザーがそのレコードを表示して操作するときは、ClearQuest でレコード フォームが使用されます。そのレコード タイプ用の登録フォームが ClearQuest にない場合は、レコードの登録と操作の両方にレコード フォームが使用されます。

フォーム ウィンドウでは、次の操作を行うことができます。

- フォームの作成と変更。フォームの追加や削除を行ったり、フォームの名前を変更したり、フォームで使用するフォントを変更できます。
- フォーム コントロールの操作。フォームの作成後に、フォームに表示するフィールドのコントロールを追加することができます。
- フォーム上のタブの操作。デフォルトでは、各レコード フォームにはタブが 1 つ作成されます。フォームに別のタブを追加したり、タブへのアクセスを制限したり、フォーム上でのタブの表示順序を変更できます。

詳細情報

- 作成したレコード フォームは、複数のスキーマで使用することができます。165 ページの「レコード フォームの再使用」を参照してください。
- 複数のプラットフォームで表示するフォームを作成する場合は、150 ページの「複数のプラットフォームで使用するフォームの作成」を参照してください。
- ClearQuest API を使用すると、フォームではなく、プログラムによってレコードの登録、表示、編集ができます。詳細については、Rational ClearQuest のマニュアルを参照してください。

フォームの作成と変更

スキーマ内の各レコード タイプに対して、レコード フォームを 1 つずつ定義する必要があります。各レコード タイプに対して、登録フォームを定義することもできます。デフォルトでは、レコード フォームと登録フォームにはタブが 1 つずつ作成されます。別のタブを追加することもできます。

新しいフォームの作成

フォームを作成すると、ClearQuest によって、フォームに [適用] ボタン、[元に戻す] ボタン、[アクション] ボタンの 3 つのコントロールが自動的に作成されます。これらのボタンを使用して、アクションの開始と管理を行います。これらのボタンは削除できませんが、ボタンのプロパティは変更できます。たとえば、[適用] ボタンのラベルを [OK] に変更したり、[元に戻す] ボタンのラベルを [キャンセル] に変更できます。

フォームを作成するには

- 1 ClearQuest Designer のワークスペースで、[レコードタイプ] または [レコードタイプ - 状態なし] を展開し、対象のレコードタイプを表示します。
- 2 [フォーム] フォルダを右クリックし、ショートカットメニューの [追加] をクリックします。
右側のペインに新しい空のフォームが表示され、ワークスペースでフォーム名が反転表示されて変更可能な状態になります。
- 3 ワークスペースで、新しいフォームの名前を入力します。
名前の長さは半角で 25 文字までです。使用できる文字は、英大文字、英小文字、数字、アンダースコアのみです。
- 4 新しいフォーム名を右クリックし、[レコード フォーム] または [登録フォーム] をクリックします。

- 5 フォームを作成したら、フォームに表示する各フィールドのコントロールを追加する必要があります。詳細については、151 ページの「フォーム コントロールの操作」を参照してください。また、タブを追加してコントロールをグループ化することもできます。162 ページの「フォーム上のタブの操作」を参照してください。

メモ: フォントと言語によっては、Windows、UNIX、Web の各 ClearQuest クライアントでフォームの表示が異なる場合があります。使用する特定のクライアントでフォームの表示テストを行い、必要に応じて修正してください。

フォーム名の変更

フォーム名を変更するには

- ClearQuest Designer のワークスペースでフォーム名を右クリックし、ショートカットメニューの [名前の変更] をクリックします。

フォームのサイズの変更

フォームのサイズを変更して、コントロールに使用するスペースを最適化することができます。フォームのプロパティを変更すると、フォーム上のタブとデフォルトのボタンのサイズおよび位置も変化します。

フォームのコントロールが多すぎてメイン タブに表示できない場合は、別のタブを追加することができます。162 ページの「フォーム上のタブの操作」を参照してください。

フォームのサイズを変更するには

- 1 ClearQuest Designer のワークスペースで、[レコードタイプ] または [レコードタイプ - 状態なし] フォルダを展開し、サイズを変更するフォームを表示します。
- 2 フォームをダブルクリックします。
- 3 フォーム ウィンドウの角をクリックし、目的の形とサイズになるまでドラッグします。

フォームの幅と高さを特定の値に設定するには、フォームのプロパティ シートを編集します。

- 1 フォームのタブを右クリックし、ショートカット メニューの [フォームのプロパティ] をクリックします。
- 2 [フォーム プロパティ シート] で、幅と高さの値を入力し、[OK] をクリックします。

フォームのフォントの変更

スキーマ内にあるすべてのフォームのフォントとフォントのプロパティを変更することができます。フォントの変更は、スキーマ内のすべてのフォームのタブに表示されるすべてのテキストに反映されます。コントロールのラベルやタブ名のフォントにも適用されます。

スキーマ内のすべてのフォームのフォントとフォントのプロパティを変更するには

- 1 ClearQuest Designer のワークスペースでフォームを右クリックし、ショートカットメニューの [フォント] をクリックします。
- 2 [フォント] ダイアログ ボックスで、使用するフォントとフォントのプロパティを選択します。
- 3 フォームを一度閉じて再度開くと、新しいフォントで表示されます。

フォームの削除

フォームが不要になった場合は、削除することができます。ただし、レコードタイプにフォームが 1 つしかない場合は、そのフォームがレコードフォームとして必要です。レコードタイプにフォームが 1 つしかない場合は、そのフォームを削除することはできません。

フォームを削除するには

- 1 ClearQuest Designer のワークスペースで、[レコードタイプ] または [レコードタイプ 状態なし] フォルダを展開し、対象のレコードタイプを展開して、[フォーム] フォルダを展開します。
- 2 フォームを右クリックし、ショートカットメニューの [削除] をクリックします。

複数のプラットフォームで使用するフォームの作成

ClearQuest Designer で作成したフォームは、Windows クライアントと ClearQuest Web クライアントとは表示方法が異なります。たとえば、Windows ではタブをクリックして表示しますが、ClearQuest Web ではリンクをクリックするかスクロールしてタブを表示します。しかし、どちらのクライアントを使用しても利用できる情報は同じです。

使用するすべてのプラットフォームでフォームの表示テストを行ってください。

詳細については、「第 11 章 ClearQuest Web の管理」を参照してください。

インポートしたデータ用のフォームの作成と変更

インポートしたレコードに合わせてスキーマを更新する場合は、適切なフィールドを追加または削除することで、既存のフォームを変更できます。新しいスキーマを作成する場合は、インポートしたレコードからのフィールドを表示した新しいフォームを作成できます。

インポートしたレコードを表示するフォームを作成または変更するときは、インポートしたレコードの古い ID を表示するフィールドを 1 つ追加することをお勧めします。古い ID 用のフィールドは新しいレコードに必須ではありませんが、古いデータベースからの情報を使用するインポートレコードを識別するのに便利です。

フォーム コントロールの操作

コントロールは、フォーム上にフィールドを表示するために使用します。ClearQuest Designer では、テキスト ボックス、リスト ボックス、チェックボックス、オプション ボタンなどのコントロールを使用できます。たとえば、文字列を含むフィールドをテキスト ボックス コントロールに関連付けることができます。ただし、すべてのコントロールをすべてのフィールドタイプに使用できるわけではありません。たとえば、一覧コントロールや親/子コントロールは、参照リスト フィールドでしか使用できません。

一部のコントロールでは、フィールドの内容を表示するほかに、特別なタスクを実行することもできます。プッシュ ボタンやリスト ボックスなどのコントロールは、レコード スクリプトに関連付けることができます。たとえば、TestStudio スキーマでは、プッシュ ボタンを Build_Properties レコード スクリプトに関連付けます。ユーザーはこのレコード スクリプトを使用して、選択したビルドのプロパティを表示できます。

ClearQuest Designer には、登録済みの ActiveX コントロールをフォームに組み込むための ActiveX コントロールも用意されています。たとえば、ActiveX コントロールを使用して、外部のデータベースと対話できます。このコントロールを使用するには、ActiveX の機能と、コントロールを登録する方法に熟知している必要があります。

ClearQuest では、次のフォーム コントロールがサポートされています。

フォーム コントロール	説明
ActiveX	登録されている ActiveX コントロールをフォームに組み込みます。初期化レコード スクリプトとアクション レコード スクリプトを記述する必要があります。
添付ファイル	添付ファイルのリストを表示し、ユーザーが添付ファイルを追加、削除、表示するための一連のコントロールを提供します。
チェック ボックス	ブール値または 2 つの値しか持たないフィールドに使用できる、2 値のコントロールです。2 つの値を指定するには、フォーム上のコントロールを右クリックして [プロパティ] を選択します。

フォーム コントロール	説明
コンボ ボックス	編集可能なテキスト フィールドをリスト ボックスと組み合わせます。
ドロップ ダウン リスト ボックス	特定のフィールドで使用可能な値のリストを表示します。
ドロップ ダウン コンボ ボックス	編集可能なテキスト フィールドをドロップダウン リスト ボックスと組み合わせます。
重複元	このレコードの重複元レコードの ID を表示します。
重複先	このレコードの重複レコードがある場合、その ID を表示します。
グループ ボックス	複数のコントロールを視覚的にグループ化します。
履歴	レコードに対して実行されたアクションの履歴情報を表示します。
リスト ボックス	特定のフィールドで使用可能な値のリストを表示します。リスト ボックスには、選択リストから 1 つまたは複数の項目を選択するための追加コントロールが含まれます。
一覧	REFERENCE_LIST タイプのフィールドに関連付けられたレコードを表示します。関連付けられている参照リストを複数列形式で表示します。
オプション ボタン	グループ内で同時に指定できない一連の選択肢を示すために使用します。1 つのグループで 1 つのオプションしか選択できないようにします。
親/子	フォームを、関連付けられたレコードにリンクするように設定できます。REFERENCE_LIST フィールド タイプに使用します。親/子コントロールは、1 つの一覧コントロールと 3 つのプッシュ ボタンで構成されます。一覧コントロールとプッシュ ボタンは、一意の一覧 ID で自動的に関連付けられます。一覧 ID を変更する場合は、プッシュ ボタンも更新する必要があります。
図	フォーム上に静的なイメージを表示します。
プッシュ ボタン	レコードに関連する特定のタスクを開始します。プッシュ ボタンは、レコードフックまたは一覧に関連付けることができます。
静的テキスト	編集不可能なテキスト文字列を表示します。
テキスト ボックス	フィールドの値を編集可能なテキスト文字列として表示します。

各コントロール タイプのプロパティの詳細については、「付録 D フォーム コントロール」を参照してください。

フォームへのコントロールの追加

フォームにフィールドを追加する前に、フィールド グリッドにフィールドを追加する必要があります。詳細については、106 ページの「新規フィールドの追加」を参照してください。

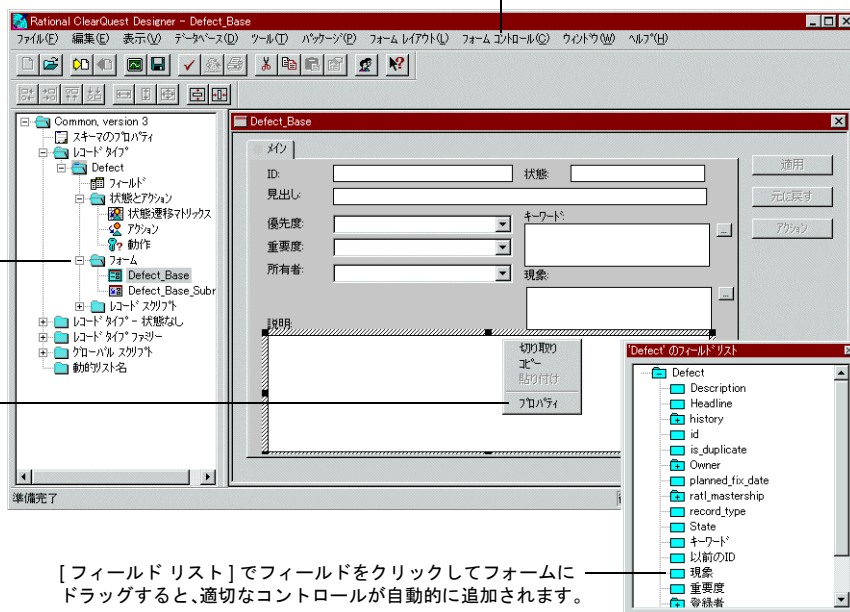
フォームへのコントロールの追加は、コントロールパレット、[フィールド リスト]、[フォーム コントロール] メニューを使用して行うことができます。

- コントロール パレットでは、追加するコントロールのタイプに関するヒントを視覚的に得られます。
- [フィールド リスト] でフィールドを選択してフォームにドラッグすると、適切なコントロールが自動的に追加されます。
- [フォーム コントロール] メニューを使用すると、コントロールパレットを使用した場合と同じコントロールを作成できます。さらに、このメニューには、ActiveX コントロールを作成するためのコマンドも含まれています。

フォームにコントロールを追加するには、[フォーム コントロール] メニューまたはコントロールパレットを使用します。

フォームツールバーを使用して
フォーム上にコントロールを
配置します。

コントロールを右クリックし、
[プロパティ] をクリックして、
コントロールのプロパティを
編集します。



メモ: 1 つのフォームに 1 つのフィールドを何度でも追加できますが、フィールドのすべてのインスタンスは同じ値である必要があります。

フォームを開く

フォームを開いて編集するには

- 1 ClearQuest Designer のワークスペースで、[レコードタイプ] または [レコードタイプ - 状態なし] フォルダを展開し、対象のフォームを表示します。
- 2 フォームをダブルクリックします。

フォームが開き、コントロールパレットと [フィールドリスト] が表示されます。表示されない場合は、[表示] メニューの [コントロールパレット] か、[表示] メニューの [フィールドリスト] をクリックします。

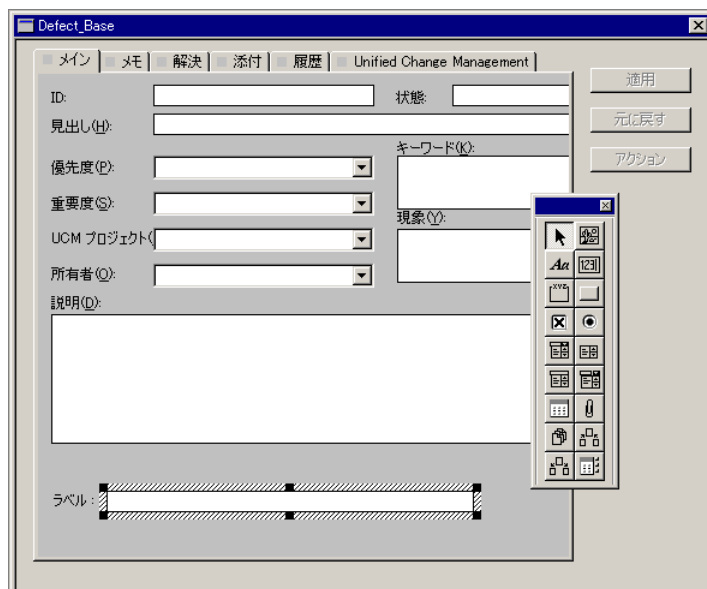
コントロールパレットによるコントロールの追加

フォームを開いたときにコントロールパレットが表示されない場合は、[表示] メニューの [コントロールパレット] をクリックします。

ClearQuest Designer でコントロールパレットを使用してコントロールを追加するには

- 1 コントロールパレットで、追加するコントロールをクリックします。
- 2 フォーム上でコントロールを配置する場所をクリックし、コントロールが目的のサイズになるまでドラッグします。

コントロールをフォーム上に配置すると、ClearQuest Designer のカーソルが選択用の矢印に変わり、コントロールの位置やサイズを変更することができます。



- 3 コントロールをフォーム上に配置したら、既存のフィールドを新しいコントロールに関連付ける必要があります。コントロールをダブルクリックしてプロパティシートを開き、コントロールに関連付けるフィールドを選択します。

コントロールの表示オプションを設定することもできます。詳細については、157 ページの「コントロールのプロパティの編集」を参照してください。

[フィールド リスト] によるコントロールの追加

[フィールド リスト] を使用してコントロールをフォームに追加する場合は、リストからフィールドを選択してフォームにドラッグするだけで、そのフィールドタイプに適したコントロールが ClearQuest によって自動的に追加されるという利点があります。コントロールのプロパティを編集して、コントロールにフィールドを関連付ける手間が省けます。

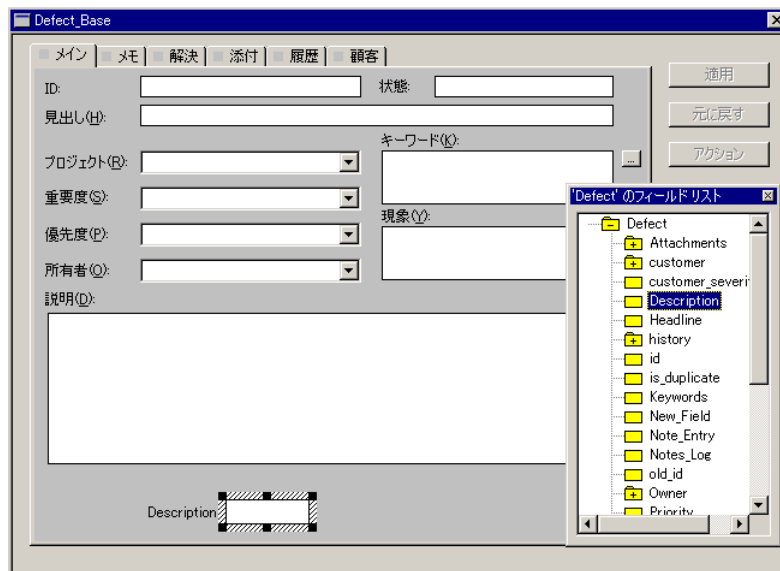
次の表に、ClearQuest Designer で各フィールドタイプに対して自動的に作成されるデフォルトのコントロールを示します。

フィールド タイプ	デフォルトのコントロール
添付リスト	添付ファイル
定数リスト 選択リスト	ドロップ ダウン リスト ボックス
日付と時刻	テキスト ボックス
整数	テキスト ボックス
複数行文字列	テキスト ボックス
参照リスト	親/子
参照リスト 定数	一覧
参照 選択リスト = デフォルト	ドロップ ダウン リスト ボックス
短い文字列 定数リスト	ドロップ ダウン リスト ボックス
短い文字列	テキスト ボックス

[フィールドリスト]を使用してコントロールを追加するには

- 1 フォームを開いたときにフィールドリストが表示されない場合は、[表示]メニューの[フィールドリスト]をクリックします。
- 2 [フィールドリスト]からフィールドをクリックし、タブにドラッグします。

コントロールをフォーム上に配置すると、ClearQuest Designer のカーソルが選択用の矢印に変わり、コントロールの位置やサイズを変更することができます。



- 3 フィールドをフォームに追加したら、コントロールプロパティを編集してコントロールの表示オプションを設定できます。コントロールをダブルクリックして、プロパティシートを開きます。詳細については、157 ページの「コントロールのプロパティの編集」を参照してください。

[フォーム コントロール] メニューによるコントロールの追加

ClearQuest Designer の [フォーム コントロール] メニューを使用してコントロールを追加するには

- 1 [フォーム コントロール] メニューから、フォームに追加するコントロールのタイプを選択します。
- 2 コントロールを追加する場所をクリックし、コントロールが目的のサイズになるまでドラッグします。

コントロールをフォーム上に配置すると、ClearQuest Designer のカーソルが選択用の矢印に変わり、コントロールの位置やサイズを変更することができます。

- 3 コントロールをフォーム上に配置したら、既存のフィールドを新しいコントロールに関連付ける必要があります。コントロールをダブルクリックしてプロパティシートを開き、コントロールに関連付けるフィールドを選択します。

コントロールの表示オプションを設定することもできます。詳細については、157 ページの「コントロールのプロパティの編集」を参照してください。

フォームでのコントロールの選択

1 つまたは複数のコントロールを選択してコントロールの移動やサイズ変更を行うには、選択ツールを使用します。

フォームで 1 つまたは複数のコントロールを選択するには

- 1 コントロールパレットで選択ツールが選択されていることを確認します。
- 2 コントロールをクリックして選択します。ほかのコントロールも選択するには、[Shift] を押しながらクリックします。

コントロールのプロパティの編集

メモ: コントロールパレットや [フォーム コントロール] メニューを使用してフォームに新しいコントロールを追加した場合は、コントロールのプロパティを編集して、コントロールと既存のフィールドに関連付ける必要があります。

フォームにコントロールを追加すると、コントロールのプロパティの編集が可能になります。コントロールに関連付けるフィールドを選択したり、コントロールの表示属性を指定できます。コントロールによっては、レコードフックを割り当てることで、そのコントロールを使用して特定のタスクを実行することもできます。

ClearQuest Web で依存フィールドを有効にする場合は、依存フィールドを指定する必要があります。依存フィールドは、コントロールの [プロパティ] シートの [Web 依存先フィールド] タブで指定します。

フィールド コントロールのプロパティを編集するには

- 1 コントロールを右クリックし、ショートカットメニューの [プロパティ] をクリックします。



- 2 コントロールの [プロパティ] シートで、コントロールのプロパティを編集します。フィールド名とそのラベル、データ形式、コンテキスト メニュー フック、フィールドでの横方向または縦方向のスクロールの使用などを変更することができます。選択できるオプションは、フィールドとコントロールのタイプによって異なります。

各コントロール タイプのプロパティの詳細については、「付録 D フォーム コントロール」を参照してください。

フォーム上のコントロールの削除

コントロールが不要になった場合は、フォームから削除することができます。

メモ: コントロールを削除すると、コントロールとそのラベルがフォームから削除されますが、関連付けられているフィールドはスキーマから削除されません。フィールドをスキーマから完全に削除するには、フィールドをフィールドグリッドから削除する必要があります。詳細については、113 ページの「フィールドの削除」を参照してください。

フォームからコントロールを削除するには

- 1 ClearQuest Designer のワークスペースで、[レコードタイプ] または [レコードタイプ - 状態なし] を展開し、対象のフォームを表示します。
- 2 フォームをダブルクリックします。
- 3 削除するコントロール (複数可) を選択します。
- 4 [編集] メニューの [削除] をクリックするか、[Delete] を押します。

コントロールのサイズと位置の変更

コントロールのサイズと位置を変更するには、フォーム上でコントロールをクリックして新しい場所にドラッグするか、フォーム ツールバーのツールを使用して 1 つまたは複数のコントロールの位置を調整します。

[フォーム レイアウト] メニューには、フォーム ツールバーにあるツールに加えて、その他のツールも表示されます。次の表に、各ツールとその機能を示します。

メモ: 配置ツールとサイズを揃えるツールでは、最初に選択したコントロールを基準として、ほかのコントロールが変更されます。

レイアウト ツール	説明
コントロールの配置 - 左端	各コントロールの左端を、最初に選択したコントロールの左端に揃えます。
コントロールの配置 - 右端	各コントロールの右端を、最初に選択したコントロールの右端に揃えます。
コントロールの配置 - 上端	各コントロールの上端を、最初に選択したコントロールの上端に揃えます。
コントロールの配置 - 下端	各コントロールの下端を、最初に選択したコントロールの下端に揃えます。
コントロールの配置 - 上下中央揃え	各コントロールの縦中央を、最初に選択したコントロールの縦中央に揃えます。
コントロールの配置 - 左右中央揃え	各コントロールの横中央を、最初に選択したコントロールの横中央に揃えます。
均等割り付け - 左右均等	選択したコントロールが等間隔になるように横に配置します。左右両端のコントロールは移動しません。
均等割り付け - 上下均等	選択したコントロールが等間隔になるように縦に配置します。上下両端のコントロールは移動しません。
ダイアログ内での整列 - 縦方向	各コントロールの縦中央を、タブの縦中央に揃えます。
ダイアログ内での整列 - 横方向	各コントロールの横中央を、タブの横中央に揃えます。
同じサイズに設定 - 幅	各コントロールの幅を、最初に選択したコントロールの幅に揃えます。

レイアウト ツール	説明
同じサイズに設定 - 高さ	各コントロールの高さを、最初に選択したコントロールの高さに揃えます。
同じサイズに設定 - 両方	各コントロールの幅と高さを、最初に選択したコントロールの幅と高さに揃えます。
サイズを内容に合わせる	選択した各コントロールのサイズを、内容全体を表示できるように調整します。これは、文字の読みやすさを保ったまま静的テキストコントロールのサイズを最小化する場合に便利です。

コントロールの移動

選択した 1 つまたは複数のコントロールを移動するには、次のいずれかの操作を行います。

- [Shift] を押しながら矢印キーを押します。
- コントロール (複数可) を新しい位置にドラッグします。

コントロールの配置

1 つのコントロールを基準にして、ほかのコントロールを配置することができます。

コントロールを配置するには

- 1 コントロールを選択します。

最初に選択したこのコントロールを基準に、次に選択するほかのすべてのコントロールが配置されます。

- 2 [Shift] を押しながらほかのコントロールをクリックして選択し、[フォーム レイアウト] メニューの [配置] のいずれかのコマンドをクリックするか、ツールバーの配置ボタンをクリックします。

コントロールのサイズの変更

コントロールのサイズを個別に変更することができます。また、複数のコントロールを同じ高さや幅に揃えることもできます。

各コントロールのサイズを変更するには

- コントロールの角をドラッグするか、[フォーム レイアウト] メニューの [サイズを内容に合わせる] をクリックします。

複数のコントロールを同じサイズに揃えるには

- 1 コントロールを選択します。

最初に選択したこのコントロールを基準に、次に選択するほかのすべてのコントロールが同じサイズに変更されます。

- 2 [Shift] を押しながらほかのコントロールをクリックして選択し、[フォーム レイアウト] メニューの [同じサイズに設定] のいずれかのコマンドをクリックするか、ツールバーの サイズ ボタンをクリックします。

コントロールのタブ順序の変更

コントロールのタブ順序によって、[Tab] を押したときに選択されるコントロールが決定します。[Tab] を押すと、タブ順序で次に指定されているコントロールが選択されます。

デフォルトでは、コントロールのタブ順序はフォームに追加された順序になっています。ユーザーがコントロールを使用する順序を想定し、その順序に応じてタブの順序を変更することができます。

コントロールのタブ順序を変更するには

- 1 ClearQuest Designer のワークスペースで、[レコードタイプ] または [レコードタイプ - 状態なし] を展開し、対象のフォームを表示します。
- 2 フォームをダブルクリックします。
- 3 タブ順序を設定するコントロールを含むダイアログ タブを選択します。
- 4 [フォーム レイアウト] メニューの [タブ順序を設定] をクリックします。

ClearQuest がタブ順序モードになります。このモードになると、各コントロール上に、タブ順序を示す数字が表示されます。

- 5 タブ順序の最初に指定するコントロールをクリックします。
- 6 残りのコントロールを、選択させる順序でクリックします。

各コントロールをクリックすると、表示されている番号が、新しいタブ順序の番号に変わります。

各コントロールを 1 回ずつクリックすると、ClearQuest のタブ順序モードが終了します。また、ダイアログ タブ上の何も表示されていない箇所をクリックして、タブ順序モードを終了することもできます。

フォーム上のタブの操作

タブを使用すると、コントロールをいくつかのグループに分けることができます。フォームのコントロールが多すぎてメイン タブに表示できない場合は、タブを追加してフォーム上のタブをグループ化することができます。タブは必要な数だけ追加できます。

フォームへのタブの追加

フォームにタブを追加するには

- 1 ClearQuest Designer のワークスペースで、[レコード タイプ] または [レコード タイプ - 状態なし] を展開し、対象のフォームを表示します。
- 2 フォームをダブルクリックします。
- 3 [編集] メニューの [タブの追加] をクリックします。

タブを追加した後は、タブ名の変更 (およびタブへのアクセス キーの指定)、タブへのユーザーアクセスの制限、タブ位置の変更、使用していないタブの削除、タブのコピーを行うことができます。

タブ名の変更

タブの名前を変更できます。また、ユーザーがキーボードの該当するキーを押すとタブが表示されるように、アクセス キーを追加することもできます。アクセス キーは、タブ名に下線付きで表示されます。

タブ名を変更するには

- 1 タブを右クリックしてショートカットメニューの [タブのプロパティ] をクリックするか、タブ上の何も表示されていない部分をダブルクリックします。
- 2 [タブのプロパティ] ダイアログ ボックスで、[タブのキャプション] テキスト ボックスに新しい名前を入力します。タブのアクセス キーを指定するには、アンパサンド (&) を使用します。英語名の場合は、アクセス キーを表す文字の直前に「&」を挿入します。日本語名の場合は、タブ名の後ろにアンパサンドとアクセス キーをカッコで囲んで「(&A)」のように入力します。

アクセス キーを設定しておく、アクセス キーを押すことでタブを表示できます。たとえば、[Attachments] または [添付] というタブのアクセス キーとして T を指定するには、「A&ttachments」または「添付 (&T)」のように入力します。

タブへのアクセスの制限

デフォルトでは、タブはすべてのユーザーに表示されますが、選択したグループのユーザーのみに表示されるようにタブへのアクセスを制限することができます。

ClearQuest Designer のタブへのアクセスを制限するには

- 1 フォーム上のタブを右クリックしてショートカットメニューの [タブのプロパティ] をクリックするか、タブ上の何も表示されていない部分をダブルクリックします。
- 2 [タブのプロパティ] ダイアログ ボックスで、次のいずれかの処理を行います。
 - すべてのユーザーにタブが表示されるようにするには、[すべてのユーザー] を選択します。
 - 選択したグループのユーザーのみにタブが表示されるようにするには、[すべてのユーザー] チェックボックスをオフにし、[使用可能なグループ] リストでグループを選択して [追加] をクリックします。
 - グループのタブへのアクセス権限を削除するには、[選択されたグループ] リストでグループを選択して [削除] をクリックします。

タブの順序の変更

デフォルトでは、ClearQuest のフォーム上のタブは作成された順序で表示されます。タブの順序を変更するには、タブに新しいインデックス番号を割り当てます。インデックス番号はフォーム上のタブの順序を表します。1 番目のタブのインデックス番号は 0、2 番目のタブのインデックス番号は 1 で、以下同様に続きます。

タブの順序を変更するには、各タブのインデックス番号を個別に変更します。その後、フォームを一度閉じて再度開くと、変更が反映されます。

- 1 フォームを開き、順序を変更するタブをクリックします。
- 2 そのタブを右クリックし、ショートカットメニューの [タブのプロパティ] をクリックします。
- 3 [タブのプロパティ] ダイアログ ボックスで、インデックス番号を選択します。

タブをフォーム上で 1 番目に表示する場合は 0、2 番目に表示する場合は 1、以下同様に選択します。
- 4 フォームを一度閉じて再度開くと、変更が反映されます。

タブの削除

タブをフォームから削除できます。タブを削除すると、そのタブ上にあるすべてのコントロールも削除されます。

警告: タブを削除したら元に戻すことはできません。

タブを削除するには

- 1 ClearQuest Designer のワークスペースで、[レコードタイプ] または [レコードタイプ - 状態なし] をクリックし、対象のフォームを表示します。
- 2 フォームをダブルクリックします。
- 3 タブをクリックし、[編集] メニューの [タブの削除] をクリックします。

タブのコピー

タブをそのすべてのコントロールと共にコピーし、同じフォームの別のタブやほかのフォームに貼り付けることができます。タブをフォームに貼り付けた後は、通常の手順で変更できます。

ClearQuest Designer のタブをコピーするには

- 1 フォームを開き、コピーするタブを表示します。
- 2 [編集] メニューの [ダイアログ全体をクリップボードにコピー] をクリックします。

タブを貼り付けるには

- 1 コントロールを貼り付けるタブをクリックします。
- 2 [編集] メニューの [貼り付け] をクリックするか、タブを右クリックしてショートカットメニューの [貼り付け] をクリックします。
- 3 タブを必要に応じて編集します。

レコード フォームの再使用

1 つのフォームを複数のスキーマの共用フォームにする場合は、フォームを 1 つ作成し、このフォームをローカル システムにエクスポートしてから、ほかのスキーマにインポートします。インポートしたフォームは、そのまま使用したり、変更できます。

メモ: フォームは、フォームをエクスポートしたレコード タイプと同じ名前のレコード タイプにインポートする必要があります。

フォームのエクスポート

フォームをエクスポートすると、すべてのタブ、タブ上のすべてのコントロール、すべてのフォーム プロパティ、すべてのタブ プロパティ、すべてのフォント プロパティが保存されます。また、1 つのタブのみを別のフォームで使用することもできます。詳細については、164 ページの「タブのコピー」を参照してください。

フォームをエクスポートする前に、フォームを閉じます。フォームを閉じると、すべての変更が保存されます。

フォームをエクスポートするには

- 1 **ClearQuest Designer** のワークスペースで、[レコード タイプ] または [レコード タイプ - 状態なし] をクリックし、エクスポートするフォームを表示します。
- 2 エクスポートするフォームが開いている場合は、閉じるボックスをクリックしてフォームを保存します。
- 3 ワークスペースでフォーム名を右クリックし、ショートカット メニューの [フォームのエクスポート] をクリックします。
- 4 [フォームのエクスポート] ダイアログ ボックスで、エクスポートするフォームのファイル名を入力し、ローカル システムのエクスポート先を選択して、[保存] をクリックします。

フォーム ファイルは、拡張子 **FRM** で保存されます。

フォームのインポート

フォームは、フォームをエクスポートしたレコード タイプと同じ名前のレコード タイプにインポートする必要があります。エクスポートしたフォームを、同じスキーマ内の別のレコード タイプにインポートすることはできません。

フォームをインポートすると、**ClearQuest** によって、フォームに関連付けられているフィールドが、選択したレコード タイプのフィールドにマッピングされます。**ClearQuest** では、フィールドはフィールド名に基づいてマッピングされます。両方のフィールドのタイプが一致しているかどうかは確認されません。フォーム上のフィールドが、選択したレコード タイプ内のフィールドと対応していない場合は、該当するコントロールを有効なフィールドに再度割り当てるか、そのコントロールをフォームから削除する必要があります。

ClearQuest Designer のフォームをインポートするには

- 1 インポート データを受け取るスキーマをチェックアウトします。
- 2 ワークスペースで [レコード タイプ] または [レコード タイプ - 状態なし] を展開します。
- 3 対象のレコード タイプの [フォーム] フォルダを右クリックし、ショートカット メニューの [フォームのインポート] をクリックします。
- 4 [フォームのインポート] ダイアログ ボックスで、フォームを含むファイルを選択し、[保存] をクリックします。
- 5 フォームをインポートしたら、必要に応じてフィールドをコントロールに再度割り当てます。

ユーザーとユーザー グループの設定と管理には、Rational ClearQuest が提供するユーザー管理ツールを使用します。このツールは、ユーザーとユーザー グループの作成、ユーザー ID とパスワードの設定、ユーザー グループへのユーザーの割り当て、スキーマ間でのユーザーとユーザー グループのインポートとエクスポート、データベースへの複数ユーザーの登録、データベースへのユーザー グループの登録、ログイン名やフル ネームによる特定のユーザーの検索に対応しています。

ユーザーとユーザー グループに関する情報は、ClearQuest のユーザー データベースとスキーマリポジトリに格納されます。

この章の内容は、次のとおりです。

- 167 ページの「ユーザー管理の概要」
- 171 ページの「ユーザーの操作」
- 182 ページの「アクションへのユーザーのアクセス制限」
- 184 ページの「ユーザーやユーザー グループのエクスポートとインポート」
- 184 ページの「MultiSite 環境でのユーザーの管理」

メモ: ユーザーとユーザー グループを管理するには、ユーザー管理者権限が必要です。

ClearQuest Designer では、はじめてログインするときにも使用できるデフォルトのユーザー名 (admin) が用意されています。ClearQuest を初めてインストールしたときは、パスワードを指定しないで、この admin アカウントでログインできます。admin ユーザー アカウントは、さまざまな管理機能を実行する際に必要なスーパー ユーザー権限を備えています。詳細については、169 ページの「ClearQuest ユーザー権限」を参照してください。

ユーザー管理の概要

ClearQuest ユーザーを管理するには、[ユーザー管理] ダイアログ ボックスを使用します。このディスカッションでは、ClearQuest Designer の [ツール] メニューの [ユーザー管理] からユーザー管理ツールにアクセスします。

これ以外の方法でも、ユーザー管理ツールにアクセスできます。ユーザー管理ツールにアクセスするには、Windows の [スタート] メニューから、[プログラム]、[Rational Software] の順にポイントし、[Rational ClearQuest] をクリックします。

[ユーザー管理] ダイアログ ボックスは、次の目的に使用します。

- 新しいユーザーとユーザー グループを作成し、ユーザーをユーザー グループに追加します。
- ClearQuest と ClearQuest Designer で実行できるタスクを定義する権限をユーザーやユーザー グループに割り当てます。

169 ページの「ClearQuest ユーザー権限」を参照してください。アクションにアクセス制御フックを追加することによって、ClearQuest の特定のアクションへのユーザーやユーザー グループのアクセスを制限できます。182 ページの「アクションへのユーザーのアクセス制限」を参照してください。

- 特定データベースへのアクセスを許可することによって、ユーザーやユーザー グループがアクセスできるデータを制御します。174 ページの「データベースへのユーザーとグループの登録」を参照してください。
- ユーザー データをエクスポートして、別のスキーマ リポジトリにインポートします。184 ページの「ユーザーやユーザー グループのエクスポートとインポート」を参照してください。

メモ: ユーザーまたはユーザー グループを追加または変更する場合は、新しい情報でユーザー データベースを更新する必要があります。174 ページの「データベースへのユーザーとグループの登録」を参照してください。

データベース登録の表示

特定のデータベースへのアクセスを許可されているユーザーとユーザー グループを確認できます。

各データベースに登録されているユーザーまたはユーザー グループを表示するには

- 1 ClearQuest Designer で、[ツール] メニューの [ユーザー管理] をクリックします。
- 2 [ユーザー管理] ダイアログ ボックスで、[DB アクション] をクリックし、[登録状況の表示] をクリックします。
- 3 [登録したユーザーおよびグループ] ダイアログ ボックスで、データベースを選択します。これにより、対象のデータベースへのアクセスを許可されているユーザーとユーザー グループを確認できます。
- 4 [OK] をクリックして、[登録したユーザーおよびグループ] ダイアログ ボックスを終了します。

ClearQuest ユーザー権限

ClearQuest では、次のユーザー権限とユーザー グループ権限がサポートされています。

権限	実行可能な内容
アクティブ ユーザー	<p>すべての ClearQuest 新規ユーザーに対するデフォルトの権限です。</p> <p>ClearQuest の基本機能へのアクセス : ClearQuest のユーザー データベースにログインし、新規レコードの登録、既存レコードの変更、個人用クエリー、グラフ、レポートの作成、変更、保存を行うことができます。</p> <p>ClearQuest Designer にログインしてスキーマとユーザー管理情報を参照できます。スキーマの編集またはユーザー情報の変更はできません。</p>
すべてのユーザー / グループを表示	<p>すべての ClearQuest 新規ユーザーに対するデフォルトの権限です。</p> <p>他のすべてのユーザーとグループに関する情報を参照できます。</p> <p>184 ページの「MultiSite 環境でのユーザーの管理」を参照してください。</p>
ユーザー管理者	<p>アクティブ ユーザーのタスクの実行に加え、以下のタスクを実行できます。</p> <ul style="list-style-type: none">▪ ClearQuest Designer を使用して、ユーザーとユーザー グループを作成し、ユーザー アクセス権限の割り当て、変更を行うことができます。▪ スーパー ユーザーのパスワードを含むユーザー パスワードを作成、変更します。
動的リスト管理者	<p>アクティブ ユーザーのタスクの実行に加え、以下のタスクを実行できます。</p> <ul style="list-style-type: none">▪ ClearQuest クライアントで動的リストを編集できます。
共用フォルダ管理者	<p>アクティブ ユーザーのタスクの実行に加え、以下のタスクを実行できます。</p> <ul style="list-style-type: none">▪ ClearQuest クライアントで共用クエリー、グラフ、レポートの作成、変更、保存、削除を行うことができます。
SQL エディタ	<p>アクティブ ユーザーのタスクの実行に加え、以下のタスクを実行できます。</p> <ul style="list-style-type: none">▪ ClearQuest クライアントでクエリーの SQL を編集できます。また、フックで API コールの BuildSQL を編集できます。

権限	実行可能な内容
セキュリティ管理者	<p>アクティブ ユーザーのタスクの実行に加え、以下のタスクを実行できます。</p> <ul style="list-style-type: none"> ▪ ClearQuest クライアントでクエリーの SQL を編集できます。 ▪ セキュリティ コンテキスト レコードのコンテキスト グループ リスト フィールドを編集できます。 ▪ すべてのレコードを参照できます。
スキーマ デザイナ	<p>アクティブ ユーザーのタスクの実行に加え、以下のタスクを実行できます。</p> <ul style="list-style-type: none"> ▪ ClearQuest Designer を使用してスキーマの作成と変更を行うことができます。レコード タイプの追加、フィールドの定義と変更、状態とアクションの作成と変更、スキーマへのフックの追加、既存のデータベースの更新を行うことができます。 ▪ ClearQuest クライアントで共用クエリー、グラフ、レポートの作成、変更、保存を行うことができます。 <p>メモ: ユーザー管理者のタスクを行うことはできません。</p>
スーパー ユーザー	<p>アクティブ ユーザー、スキーマ デザイナ、ユーザー管理者、セキュリティ管理者、共用フォルダ管理者、動的リスト管理者、SQL エディタのすべてのタスクに加え、以下のタスクを実行できます。</p> <ul style="list-style-type: none"> ▪ ClearQuest Designer を使用してデータベースとスキーマの作成と削除を行うことができます。 ▪ ClearQuest Web の設定を編集できます。 <p>ClearQuest Designer に付属している admin ユーザー アカウントには、スーパー ユーザー権限があります。</p>

許可された権限は、拒否された権限より優先されます。たとえば、特定のデータベースへのアクセスが許可されているグループにユーザーが属している場合、ユーザー個人にアクセスが許可されていない場合でも、そのデータベースにアクセスできます。

ユーザーの操作

ユーザーやユーザー グループの作成と変更を行うには、ユーザー管理者またはスーパーユーザーの権限が必要です。

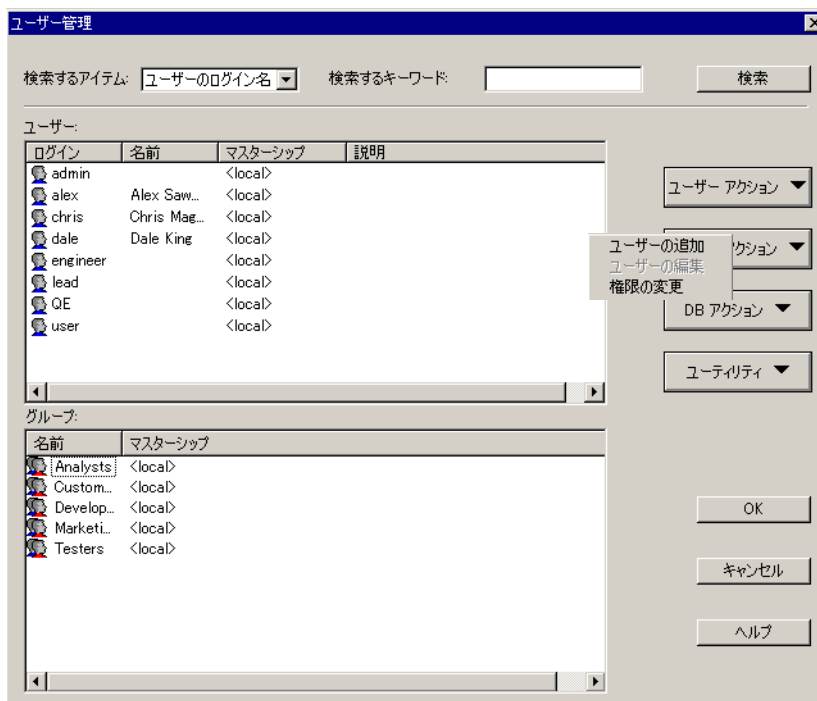
新規ユーザーの追加

新規ユーザーを追加するには

- 1 ClearQuest Designer で、[ツール] メニューの [ユーザー管理] をクリックします。
- 2 [ユーザー管理] ダイアログ ボックスで、[ユーザー アクション] をクリックし、[ユーザーの追加] をクリックします。
- 3 [ユーザーの追加] ダイアログ ボックスで、ユーザー情報を入力します。[ログイン]、[パスワード]、[名前]、[電子メール]、[電話番号]、[説明] の各フィールドに、ユーザーのログイン、パスワード、名前、電子メールアドレス、電話番号、説明を入力します。なお、[説明] フィールドは、必須ではありません。

警告: ユーザーのログインに、! "" # \$ % & ' () * + , . / : ; < = > ? @ [¥] ^ ` { | } ~ の各文字は使用しないでください。ClearQuest クライアントや ClearQuest Web にログインするとき問題が発生することがあります。

ユーザーの名前にはハイフンを入力しないでください。



あいまいなユーザー名を使用するには、ユーザー ログイン名の一部に主なサイトの名前を使用します。ログイン時にサイト拡張子を付けないであいまいなユーザー名を使用すると、「無効なログイン」エラーが発生します。

- 4 ClearQuest MultiSite を使用している場合、マスターシップを選択します。詳細については、184 ページの「MultiSite 環境でのユーザーの管理」を参照してください。
- 5 新規ユーザーが所属するグループを選択します。詳細については、180 ページの「ユーザー グループの操作」を参照してください。
- 6 新規ユーザーの権限を選択します。詳細については、173 ページの「ユーザー アクセス権限の割り当て」を参照してください。
- 7 新規ユーザーがアクセスできるデータベースを選択します。システム内のすべてのユーザーデータベースへのアクセスを許可するには、[既存のおよび将来使用するすべてのデータベース] をオンにします。個別のデータベースのアクセスを指定するには、[データベースの選択] をオンにします。
- 8 [ユーザーの追加] をクリックし、[OK] をクリックします。
- 9 [ユーザー管理] ダイアログ ボックスで [DB アクション] をクリックし、[アップグレード] をクリックします。次に [アップグレード] ダイアログ ボックスで [OK] をクリックし、選択したユーザー データベースにこの情報を追加します。詳細については、176 ページの「ユーザー データベースへのスキーマの変更の適用」を参照してください。

ユーザー アクセス権限の割り当て

作成した新規ユーザーにユーザー権限を割り当てできます。また、既存のユーザーの権限を追加、変更できます。

権限を別のユーザーに許可するには、その権限を保持している必要があります。デフォルトでは、すべてのユーザーがアクティブユーザーの権限を持ちます。アクセス権限については、169 ページの「ClearQuest ユーザー権限」を参照してください。

新規ユーザーの作成時に、ユーザー アクセス権限を割り当てるには

- 1 ClearQuest Designer で、[ツール] メニューの [ユーザー管理] をクリックします。
- 2 [ユーザー管理] ダイアログ ボックスで、[ユーザー アクション] をクリックし、[ユーザーの追加] をクリックします。
- 3 [ユーザーの追加] ダイアログ ボックスで、[権限] ペインのユーザーに対する権限をオンにします。
- 4 [OK] をクリックします。
- 5 [ユーザー管理] ダイアログ ボックスで [DB アクション] をクリックし、[アップグレード] をクリックします。次に [アップグレード] ダイアログ ボックスで [OK] をクリックし、ユーザー データベースにこの情報を追加します。詳細については、176 ページの「ユーザー データベースへのスキーマの変更の適用」を参照してください。

既存のユーザーのユーザー アクセス権限を割り当てまたは変更するには

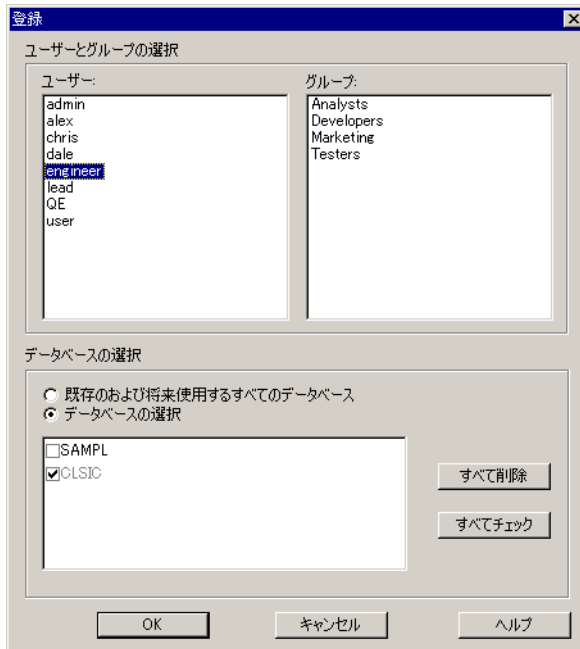
- 1 ClearQuest Designer で、[ツール] メニューの [ユーザー管理] をクリックします。
- 2 [ユーザー管理] ダイアログ ボックスで、[ユーザー アクション] をクリックし、[権限の変更] をクリックします。
- 3 ユーザーを選択し、対象のユーザーの権限をオンまたはオフにします。
- 4 [OK] をクリックします。
- 5 [ユーザー管理] ダイアログ ボックスで [DB アクション] をクリックし、[アップグレード] をクリックします。次に [アップグレード] ダイアログ ボックスで [OK] をクリックし、ユーザー データベースにこの情報を追加します。詳細については、176 ページの「ユーザー データベースへのスキーマの変更の適用」を参照してください。

データベースへのユーザーとグループの登録

データベースにユーザーまたはユーザー グループを登録することで、そのユーザーまたはグループのメンバーにデータベースへのアクセスを許可します。

選択したデータベースにユーザーやユーザー グループを登録するには

- 1 ClearQuest Designer で、[ツール] メニューの [ユーザー管理] をクリックします。
- 2 [ユーザー管理] ダイアログ ボックスで [DB アクション] をクリックし、[登録] をクリックします。
- 3 [ユーザーとグループの選択] ペインで、ユーザーまたはユーザー グループを選択します。
[データベースの選択] ペインで、ユーザーにアクセスを許可するデータベースを選択します。システム内のすべてのユーザー データベースへのアクセスを許可するには、[既存のおよび将来使用するすべてのデータベース] をオンにします。個別のデータベースのアクセスを指定するには、[データベースの選択] をオンにします。



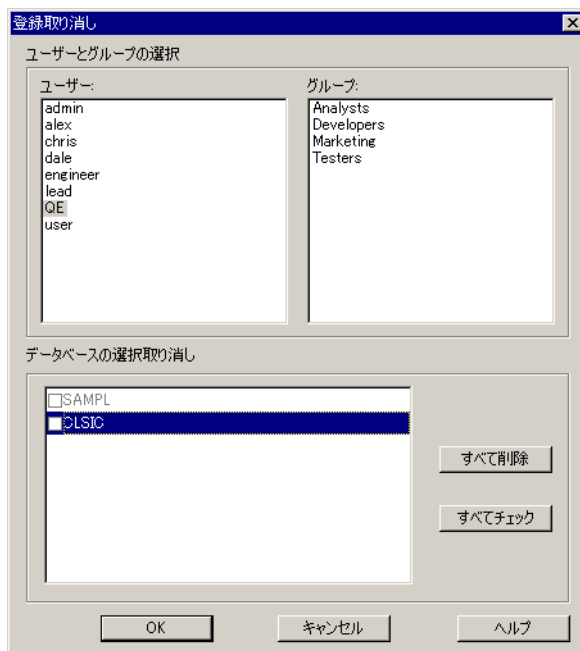
- 4 [OK] をクリックします。
- 5 [ユーザー管理] ダイアログ ボックスで [DB アクション] をクリックし、[アップグレード] をクリックします。次に [アップグレード] ダイアログ ボックスで [OK] をクリックし、ユーザー データベースにこの情報を追加します。詳細については、176 ページの「ユーザー データベースへのスキーマの変更の適用」を参照してください。

データベースからのユーザーやグループの登録取り消し

データベースからユーザーまたはユーザー グループの登録を取り消すことで、そのユーザーまたはグループのデータベースへのアクセスを拒否します。

選択したデータベースに対してユーザーまたはユーザー グループの登録を取り消すには

- 1 ClearQuest Designer で、[ツール] メニューの [ユーザー管理] をクリックします。
- 2 [ユーザー管理] ダイアログ ボックスで [DB アクション] をクリックし、[登録取り消し] をクリックします。
- 3 [登録取り消し] ダイアログ ボックスで、ユーザーまたはユーザー グループを選択し、アクセスを取り消すデータベースを選択解除します。



- 4 [OK] をクリックします。
- 5 [ユーザー管理] ダイアログ ボックスで [DB アクション] をクリックし、[アップグレード] をクリックします。次に [アップグレード] ダイアログ ボックスで [OK] をクリックし、ユーザー データベースにこの情報を追加します。詳細については、176 ページの「ユーザー データベースへのスキーマの変更の適用」を参照してください。

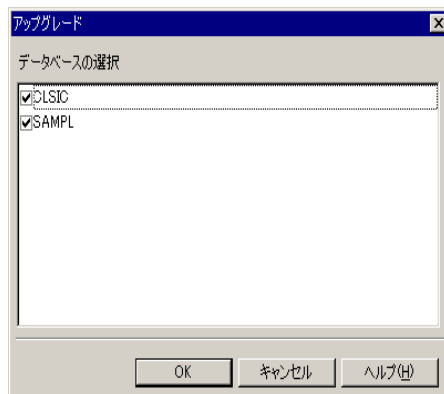
ユーザー データベースへのスキーマの変更の適用

ユーザーまたはユーザー グループを追加または変更する場合は、スキーマの変更をユーザー データベースに適用する必要があります。

警告: ユーザー データベースに変更を適用する前に、データベースにログインしているユーザーがいないことを確認してください。ユーザーがユーザー データベースに接続していると、アップグレードできません。更新の最中は、ClearQuest によって新しいユーザーのデータベースへのログインが防止されますが、既にログインしているユーザーの切断は実行されません。

ユーザー データベースにスキーマの変更を適用するには

- 1 ClearQuest Designer で、[ツール] メニューの [ユーザー管理] をクリックします。
- 2 [ユーザー管理] ダイアログ ボックスで [DB アクション] をクリックし、[アップグレード] をクリックします。
- 3 [アップグレード] ダイアログ ボックスで、変更を適用するデータベースを選択し、[OK] をクリックします。



- 4 [OK] をクリックし、[ユーザー管理] ダイアログ ボックスを終了します。

ユーザーの編集

ユーザー プロファイルを編集するには

- 1 ユーザー管理者権限またはスーパー ユーザー権限で ClearQuest Designer にログインします。次に、[ツール] メニューの [ユーザー管理] をクリックします。
- 2 [ユーザー管理] ダイアログ ボックスで、[ユーザー アクション] をクリックし、[編集] をクリックします。

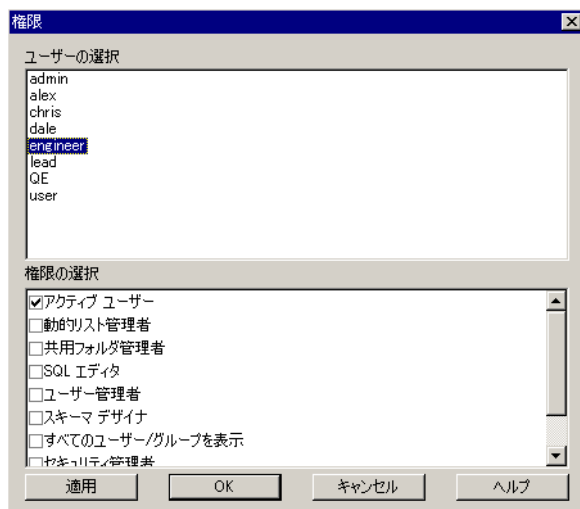
- 必要に応じてユーザー プロファイルを編集します。詳細については、171 ページの「新規ユーザーの追加」を参照してください。

メモ: ClearQuest のユーザーは ClearQuest クライアントから自分自身のユーザー情報の一部を編集できます。178 ページの「ClearQuest クライアントでのユーザー プロファイルの編集」を参照してください。

ユーザー権限の変更

ユーザー権限を変更するには

- [ユーザー管理] ダイアログ ボックスで、ユーザーを選択します。次に、[ユーザー アクション] をクリックし、[権限の変更] をクリックします。
- [権限] ダイアログ ボックスで、ユーザーを選択し、対象の権限を選択するか選択解除します。



- [適用] をクリックし、[OK] をクリックします。
- [ユーザー管理] ダイアログ ボックスで [DB アクション] をクリックし、[アップグレード] をクリックします。次に [アップグレード] ダイアログ ボックスで [OK] をクリックし、ユーザー データベースにこの情報を追加します。詳細については、176 ページの「ユーザー データベースへのスキーマの変更の適用」を参照してください。

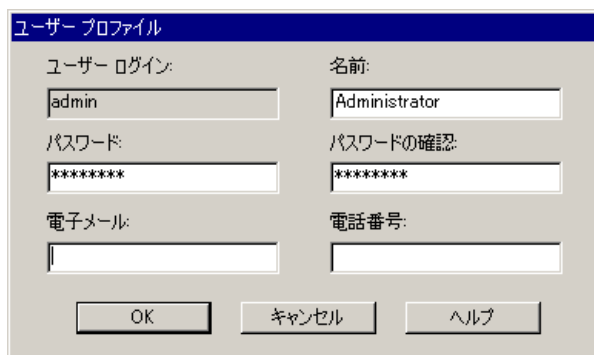
ClearQuest クライアントでのユーザー プロファイルの編集

アクティブ ユーザーは、名前、電子メールアドレス、パスワード、電話番号など、自分自身のユーザー情報の一部を ClearQuest クライアントで変更できます。ただし、アクティブ ユーザーは、自分自身のユーザー ログインまたは権限を変更できません。

メモ: ユーザー プロファイルに加えた変更は、ユーザーが登録されているすべてのデータベースに適用されます。

ClearQuest クライアントからユーザー プロファイルを変更するには

- 1 ClearQuest クライアントで、[表示] メニューの [ユーザー プロファイルの変更] をクリックします。
- 2 必要なフィールドを変更します。



- 3 [OK] をクリックします。

ユーザーやグループの無効化

ユーザーやユーザー グループを無効にできます。無効とは、ユーザーの ClearQuest へのログインを禁止することで、ユーザーは、割り当てられたレコードを持てなくなります。無効なユーザーやグループは、ClearQuest クライアントのユーザーの選択リストに表示されません。

メモ: 履歴情報とデータの一貫性確保の目的で、ClearQuest Designer は無効なユーザーの名前を、登録していたデータベースに保持します。

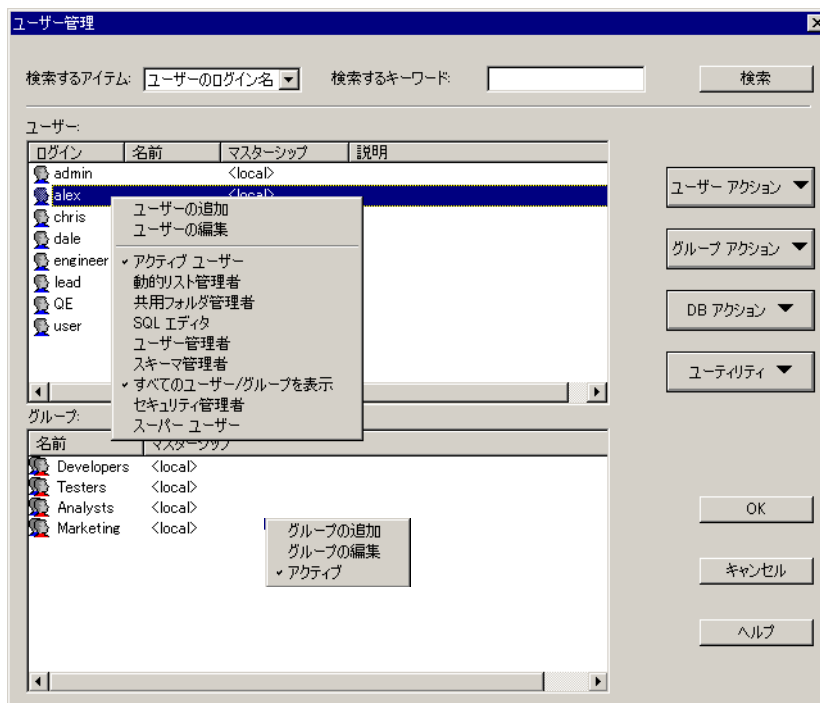
ユーザーまたはユーザー グループを無効にするには

- 1 ClearQuest Designer で、[ツール] メニューの [ユーザー管理] をクリックします。
- 2 [ユーザー管理] ダイアログ ボックスで、次の処理を行います。

- ユーザーを無効にするには、ユーザー名をマウスの右ボタンでクリックし、ショートカットメニューを使用してすべてのユーザー権限 ([アクティブ ユーザー] も含む) をオフにします。
- グループを無効にするには、グループ名をマウスの右ボタンでクリックし、ショートカットメニューの [アクティブ] をオフにします。

3 [DB アクション] をクリックし、[アップグレード] をクリックします。次に [アップグレード] ダイアログ ボックスで [OK] をクリックし、ユーザー データベースにこの情報を追加します。詳細については、176 ページの「ユーザー データベースへのスキーマの変更の適用」を参照してください。

ユーザーまたはユーザー グループを再度有効にするには、ユーザーに対応する [アクティブ ユーザー] をオンにして、[DB アクション] をクリックし、[アップグレード] をクリックします。次に [アップグレード] ダイアログ ボックスで [OK] をクリックします。



ユーザー グループの操作

個別のユーザーをグループに統合すると、管理業務が簡単になります。たとえば、次のようなことができます。

- ソフトウェア エンジニアまたは管理職などのグループを作成し、異なるデータベースに登録できます。
- 新しい障害がユーザーから登録されたら、必ず品質保証グループに通知するなど、1 つのグループに基づいて電子メールのルールを設定できます。
- 管理職グループのメンバーのみが **Resolved** とマークされた障害を再度開くことができるようにするなど、グループごとの条件を付けてフック スクリプトまたは外部アプリケーションを作成できます。

新しいユーザー グループの作成

ユーザー グループを作成するには、ユーザー管理者またはスーパー ユーザーの権限が必要です。ユーザー グループを作成するには、次の手順で操作します。

- 1 ClearQuest Designer で、[ツール] メニューの [ユーザー管理] をクリックします。
- 2 [ユーザー管理] ダイアログ ボックスで、[グループ アクション] をクリックし、[グループの追加] をクリックします。
- 3 [グループ] ダイアログ ボックスで、[名前] フィールドに新しいグループの名前を入力します。
- 4 ClearQuest MultiSite を使用している場合、[マスターシップ] からマスターシップを選択します。

サブグループの作成

グループにサブグループを追加するには

- [グループの追加] ダイアログ ボックスで、[グループ] リストから既存のグループを選択し、[追加] をクリックします。

グループへのユーザーの追加

グループにユーザーを追加するには

- [グループの追加] ダイアログ ボックスで、グループに追加するユーザーを選択し、[追加] をクリックします。

メモ: 代わりに、[ユーザー管理] ダイアログ ボックスで、[ユーザー] リストからユーザーを選択し、[グループ] リストのグループにドラッグすることもできます。

グループからのユーザーまたはサブグループの削除

グループからユーザーまたはサブグループを削除するには

- [グループの追加] ダイアログ ボックスで、サブグループまたはユーザーをクリックし、[削除] をクリックします。

データベースへのユーザー グループの登録

グループをデータベースに登録するには

- 1 [グループの追加] ダイアログ ボックスで、ユーザーにアクセスを許可するデータベースを選択します。システム内のすべてのユーザー データベースへのアクセスを許可するには、[既存のおよび将来使用するすべてのデータベース] オプションをオンにします。個別のデータベースのアクセスを指定するには、[データベースの選択] オプションをオンにします。
- 2 [適用] をクリックし、[グループ] ダイアログ ボックスを終了します。
- 3 [ユーザー管理] ダイアログ ボックスで [DB アクション] をクリックし、[アップグレード] をクリックします。次に [アップグレード] ダイアログ ボックスで [OK] をクリックし、ユーザー データベースにこの情報を追加します。詳細については、176 ページの「ユーザー データベースへのスキーマの変更の適用」を参照してください。

174 ページの「データベースへのユーザーとグループの登録」も参照してください。

アクションへのユーザーのアクセス制限

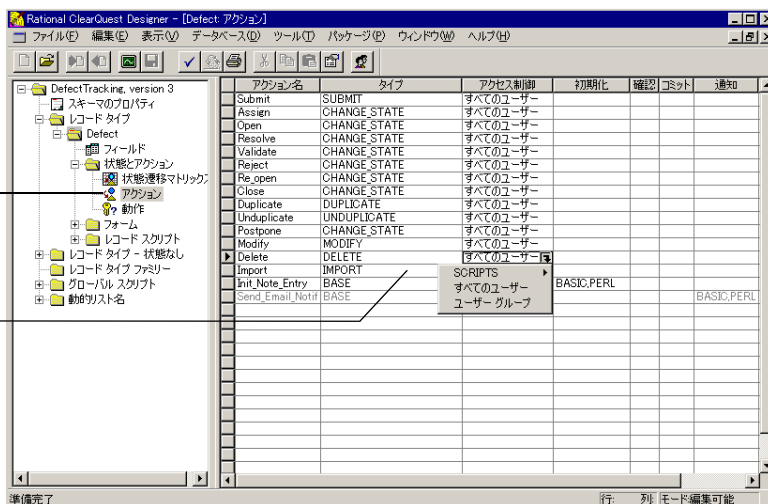
デフォルトでは、ClearQuest クライアントのすべてのユーザーがレコードに対してすべてのアクションを実行できます。ClearQuest クライアントでユーザーが実行できるアクションを制限することで、ユーザーまたはユーザー グループの責任を限定できます。たとえば、Assign アクションを管理職グループに限定し、Verify アクションを品質管理グループに限定できます。アクションへのアクセスを制限するには、アクセス制御フックをアクションに追加します。

アクションへのアクセスを制限するには

- 1 ClearQuest Designer で、[レコードタイプ] から [<record type>] フォルダの [状態とアクション] フォルダをクリックし、[アクション] をダブルクリックしてアクショングリッドを表示します。

ダブルクリックして
アクショングリッドを
開きます。

アクションの
[アクセス制御] 列を
クリックして、アクセス
のタイプを選択します。



- 2 アクション グリッドで、制限するアクションの [アクセス制御] セルをクリックし、下矢印をクリックし、アクションと関連付けるアクセス制御のタイプを選択します。

- [SCRIPTS] を選択して、指定する条件に基づいてこのアクションへのアクセスを制限する独自のアクセス制御フックを作成します。たとえば、スーパー ユーザー権限を持つユーザーに対してアクションへのアクセスを許可するフックを作成できます。詳細については、199 ページの「ワークフローのカスタマイズにおけるフックの使用法」を参照してください。
- [すべてのユーザー] (デフォルト) を選択して、アクションへのアクセスをすべての ClearQuest ユーザーに許可します。
- [ユーザー グループ] を選択し、このアクションにアクセスを許可するグループを選択します。

メモ: プライマリ アクション (Modify、Submit、Delete、Import) を実行するには、現在のユーザーが、すべての基本アクションのみでなく、プライマリ アクションのアクセス制御リストに属している必要があります。

ユーザーやユーザー グループのエクスポートとインポート

スキーマ リポジトリからユーザーやユーザー グループのプロファイルをエクスポートし、別のスキーマ リポジトリにインポートすることができます。

ユーザーやユーザー グループをエクスポートするには

- 1 ClearQuest Designer で、[ツール] メニューの [ユーザー管理] をクリックして、[ユーザー管理] ダイアログ ボックスを表示します。
- 2 [ユーザー管理] ダイアログ ボックスで、[ユーティリティ] をクリックし、[エクスポート] を選択します。

ClearQuest Designer により、各ユーザーやユーザー グループのプロファイルを含むテキスト ファイルが作成されます。このテキスト ファイルは別のスキーマ リポジトリにエクスポートできます。

ユーザーやユーザー グループをインポートするには

- 1 ClearQuest Designer で、[ツール] メニューの [ユーザー管理] をクリックして、[ユーザー管理] ダイアログ ボックスを表示します。
- 2 [ユーザー管理] ダイアログ ボックスで、[ユーティリティ] をクリックし、[インポート] を選択します。

インポートするユーザー情報やグループ情報を含むファイルを選択し、[開く] をクリックします。

メモ: 特定のデータベースに登録されたユーザーまたはユーザー グループのプロファイルをインポートする際に、プロファイルをインポートするスキーマに対象のデータベースが含まれていない場合、インポートは失敗します。この問題を回避するには、[既存のおよび将来使用するすべてのデータベース] オプションをオンにして、すべてのユーザーとグループを登録します。

MultiSite 環境でのユーザーの管理

ClearQuest MultiSite 環境では、変更の追跡とデータ破壊の防止はマスターシップと呼ばれる排他的に変更する権利をもつスキームによって行われます。マスターシップでは、レプリカデータベースのユーザーがデータを変更できる時期を決定します。

マスターシップでは、ユーザー / グループ、その他の ClearQuest オブジェクト (レコード、クエリーなど) はマスタリング レプリカを割り当てられます。ClearQuest のユーザーまたはグループの初期マスタリング レプリカはオブジェクトが作成されるサイトです。

ここでは、次のトピックについて説明します。

- マスターシップが ClearQuest クライアントのユーザーに与える影響
- マスターシップがユーザー管理に与える影響
- ユーザーがマスター登録されている場所の確認
- ユーザーのマスターシップの変更

マスターシップが ClearQuest クライアントのユーザーに与える影響

ClearQuest のユーザー アクセス権限はマスターシップによる影響を受けません。データベース権限はレプリカ間で反映されます。たとえば、ユーザーにローカルのレプリカに対して SQL エディタ権限がある場合、レプリカが同期の対象になるのであれば、そのユーザーはすべてのレプリカに対して SQL エディタ権限があります。

マスターシップは次の点で ClearQuest クライアントのユーザーに影響を与えます。

- ユーザーはログインしているレプリカでマスター登録されているレコード、オブジェクト（共用クエリーなど）のみを変更できます。ローカルのレプリカでマスター登録されていないレコード、オブジェクトを変更することはできません。
- ユーザーはマスター登録されているレプリカでのみ、次のタスクを実行できます。たとえば、ユーザーがレプリカ A でマスター登録されている場合、そのユーザーはレプリカ A にログインしているときのみ次に示すタスクを実行できます。
 - ユーザー プロファイルの変更
 - クエリーを発行中のデフォルトの保存
 - 起動クエリーの割り当て
 - ワークスペース項目を [クエリー] メニューにお気に入りとして追加

マスターシップがユーザー管理に与える影響

マスターシップは次の点でユーザー管理に影響を与えます。

- ユーザーのマスターシップを変更するには、スーパー ユーザー権限が必要です。
- ログインしているレプリカでマスター登録されているユーザー、グループのみを変更できます。別のサイトでマスター登録されているユーザーの権限を変更することはできません。
- ユーザーのマスターシップを変更できます。

ユーザーがマスター登録されている場所を知ることが重要になります。

ユーザーがマスター登録されている場所の確認

ユーザーがあるサイトで作成された場合、そのユーザーはレプリカを使用してすべてのサイトで表示されます。ただし、そのユーザーはマスターシップがあるサイトでのみ変更することができます。

ユーザーまたはグループがマスター登録されている場所を知るには、[ユーザー管理] ダイアログボックスの [マスターシップ] 列を参照します。

ユーザーのマスターシップの変更

メモ: ユーザーのマスターシップを変更するには、スーパー ユーザー権限が必要です。

ユーザーのマスターシップの変更には、次の 3 つのタスクが必要です。

- 1 マスターシップの変更。これはユーザーのマスターシップがあるサイトで行います。
- 2 マスターシップ変更後のレプリカの同期。
- 3 マスターシップの変更によるユーザー データベースのアップグレード。これは新規のマスター サイトとその他すべてのサイトで行います。

マスターシップの変更

ユーザーのマスターシップを変更するには

- 1 [ユーザー管理] ダイアログ ボックスでユーザーを選択し、[ユーザー アクション] をクリックします、次に、[ユーザーの編集] をクリックし、[ユーザー プロパティ] ダイアログボックスを表示します。
- 2 [マスターシップ] リストでレプリカを選択し、[OK] をクリックします。
- 3 ユーザーのマスターシップが変更されたことを新規サイトのユーザー管理者に通知します。新規サイトのユーザー管理者は、sync replica パケットの受信、適用を実行してから、新しいユーザー情報でユーザー データベースを更新します。

マスターシップ変更後のレプリカの同期

レプリカの同期については、『Rational ClearQuest MultiSite 管理ガイド』を参照してください。

マスターシップの変更によるユーザー データベースのアップグレード

ユーザーの新規マスターシップを取得するサイトのユーザー管理者は、新規のユーザー情報でユーザー データベースをアップグレードする必要があります。

- [ユーザー管理] ダイアログ ボックスで、[DB アクション] をクリックし、[アップグレード] をクリックします。詳細については、176 ページの「ユーザー データベースへのスキーマの変更の適用」を参照してください。

Rational ClearQuest

でのセキュリティの 使用法

9

Rational ClearQuest のセキュリティ機能を使用すると、データを保護して、権限を持つユーザーのみがレコードの参照や変更を行えるように設定できます。また、一部のユーザーに対して特定のレコードを非表示にし、ほかのユーザーにはそのレコードの参照や変更を許可することもできます。

このようなセキュリティ機能は、複数のユーザー グループが同じデータベースにアクセスする場合に特に有用です。たとえば、複数のグループが 1 つのデータベースに障害を登録する場合や、複数のプロジェクトが 1 つのデータベースを共有する場合などです。実動データベースをオープンして、さまざまな顧客が障害や拡張依頼をデータベースに登録できるようにし、同時に、ある企業の顧客が、別の企業の顧客または内部ユーザーが登録したレコードを参照できないようにすることもできます。

この章の内容は、次のとおりです。

- 188 ページの「ClearQuest でのレコードの非表示化」
- 192 ページの「セキュリティの例」
- 197 ページの「その他の ClearQuest セキュリティ機能の使用法」

ClearQuest のセキュリティ機能を実装するには、次の操作方法を理解する必要があります。

- ClearQuest スキーマの操作。スキーマ レポジトリからのスキーマのチェックインとチェックアウト、フィールドの追加、ユーザー データベースへのスキーマ変更の適用など。詳細については、「第 4 章 ClearQuest スキーマの操作」と「第 5 章 スキーマのカスタマイズ」を参照してください。
- ユーザーとユーザー グループの管理。詳細については、171 ページの「ユーザーの操作」を参照してください。
- レコード フォームへのフィールドの追加。詳細については、153 ページの「フォームへのコントロールの追加」を参照してください。

ClearQuest でのレコードの非表示化

ClearQuest のセキュリティ機能は、ユーザー グループへの帰属関係に基づいて、データベース内のレコードに対するユーザー アクセスを制限します。レコードの非表示化は、アクセスを制限するレコードのレコード タイプにセキュリティ コンテキスト フィールドを設定すると実行されます。セキュリティ コンテキスト フィールドは、セキュリティ コンテキスト レコードを参照します。このレコードには、制御されたレコードの参照や変更を行うことができるユーザーを特定するデータが含まれます。

たとえば、障害を参照できる顧客を制御するには、**Defects** レコード タイプに **customer_defects** というフィールドを設定し、このフィールドが **Customer** レコード タイプを参照するようにします。その後、顧客レコードごとにユーザー グループを割り当て、その顧客レコードを参照する障害レコードの参照権限をこれらのグループに付与します。セキュリティ コンテキスト レコードのグループ リストに登録されているユーザーのみが、制御されたレコードを参照できます。

ClearQuest 内のレコードを非表示化するには、次の操作が必要です。

- 制御するレコード タイプの決定
- セキュリティ コンテキストに基づいたユーザー グループの作成
- セキュリティ コンテキストとして使用するレコード タイプの決定
- セキュリティ コンテキスト フィールドの作成
- セキュリティ コンテキスト レコードの登録
- ユーザー アクセスを許可するためのレコードの編集

制御するレコード タイプの決定

制御されたレコード タイプとして使用するレコード タイプを決定します。制御されたレコード タイプとは、非表示化またはアクセス制限の対象になるレコード タイプのことです。たとえば、ClearQuest システム内の障害に対するアクセスを制限するには、制御されたレコード タイプとして **Defect** レコードを使用します。

セキュリティ コンテキストに基づいたユーザー グループの作成

使用するセキュリティ コンテキストに従って、新しいユーザー グループを作成するか、既存のユーザー グループを編成します。ユーザー アクセス権限を揃えるユーザー グループを作成します。その後、ユーザーをグループに割り当てます。

たとえば、特定の顧客らをセキュリティ アクセスの基準にするには、アクセス権のタイプ別に顧客グループを作成します。既存のグループを使用することも、新しいグループを作成することもできます。

メモ: 複数のセキュリティ コンテキスト フィールドを追加する場合に必要な手順は、該当するタイプのレコードを参照するため、ユーザー グループのいずれかのメンバーになることのみです。

セキュリティ コンテキストとして使用するレコード タイプの決定

セキュリティ コンテキストとして使用するレコード タイプを決定します。セキュリティ コンテキスト レコード タイプとは、セキュリティ コンテキスト フィールドが参照するレコード タイプのことです。これにはユーザー グループ情報が格納されます。

セキュリティ コンテキスト レコードは、状態があるものでも、ないものでもかまいません。この目的のために特別に作成したレコードでも、**Project** レコード タイプまたは **Customers** レコード タイプなど既存のレコードでもかまいません。

メモ: セキュリティ コンテキスト フィールドを使用して履歴、ユーザー、グループ、添付ファイルなどのシステム レコード タイプを参照することはできません。

たとえば、グループを顧客別に編成した場合、**Customer** レコード タイプをセキュリティ コンテキスト レコードとして使用できます。障害が、特定の顧客レコードと関連付けられるか、特定の顧客レコードを参照している場合、**ClearQuest** はレコードの参照許可をその顧客のセキュリティ コンテキスト グループ リストに登録されているユーザーのみに与えます。

セキュリティ コンテキスト フィールドの作成

セキュリティ コンテキスト フィールドとは、セキュリティ コンテキスト レコード タイプを参照する制御されたレコードの中の **Reference** タイプ フィールドのことです。新規フィールドを作成してセキュリティ コンテキスト フィールドとして使用することも、既存のフィールドを使用することもできます。複数のセキュリティ コンテキスト フィールドを追加できます。

メモ: セキュリティ コンテキスト フィールドは、**Reference** フィールド タイプでなければなりません。

制御されたレコード タイプに **Reference** タイプのセキュリティ コンテキスト フィールドを作成し、このフィールドがセキュリティ コンテキスト レコード タイプを参照するようにします。既存の **Reference** タイプ フィールドを使用することも、新規フィールドを作成することもできます。作成したセキュリティ コンテキスト フィールドを、制御されたレコード タイプのフォームに追加します。

セキュリティ コンテキスト レコード タイプを参照するときに、該当するフィールドの [セキュリティ コンテキスト] チェック ボックスをオンにして、セキュリティを有効にすることもできます。これにより、参照先レコード タイプのフォームに [Ratl_Security] という新しいタブが自動的に作成されます。[Ratl_Security] タブには、コンテキスト グループのリスト コントロールがあります。

たとえば、**Defect** レコード タイプに **customer_defects** という参照タイプ フィールドを作成し、このフィールドが **Customer** レコード タイプを参照するようにします。**ClearQuest** によって、[Ratl_Security] タブが **Customer** レコード タイプのフォームに追加されます。

セキュリティ コンテキスト レコードの登録

セキュリティ コンテキスト ごとにレコードを登録します。セキュリティ コンテキストが **Customer** の場合は、さらに、顧客ごとにレコードを登録する必要があります。

登録したレコードごとに、[Ratl_Security] タブを使用してグループをコンテキスト グループ リストに追加し、顧客レコードにアクセスできるユーザー グループを指定します。

警告: セキュリティ コンテキスト レコードを登録し、グループに関連付けて、制御されたレコードが表示されるようにしてください。セキュリティ コンテキスト フィールドの値が空の **Defect** レコードは表示されません。

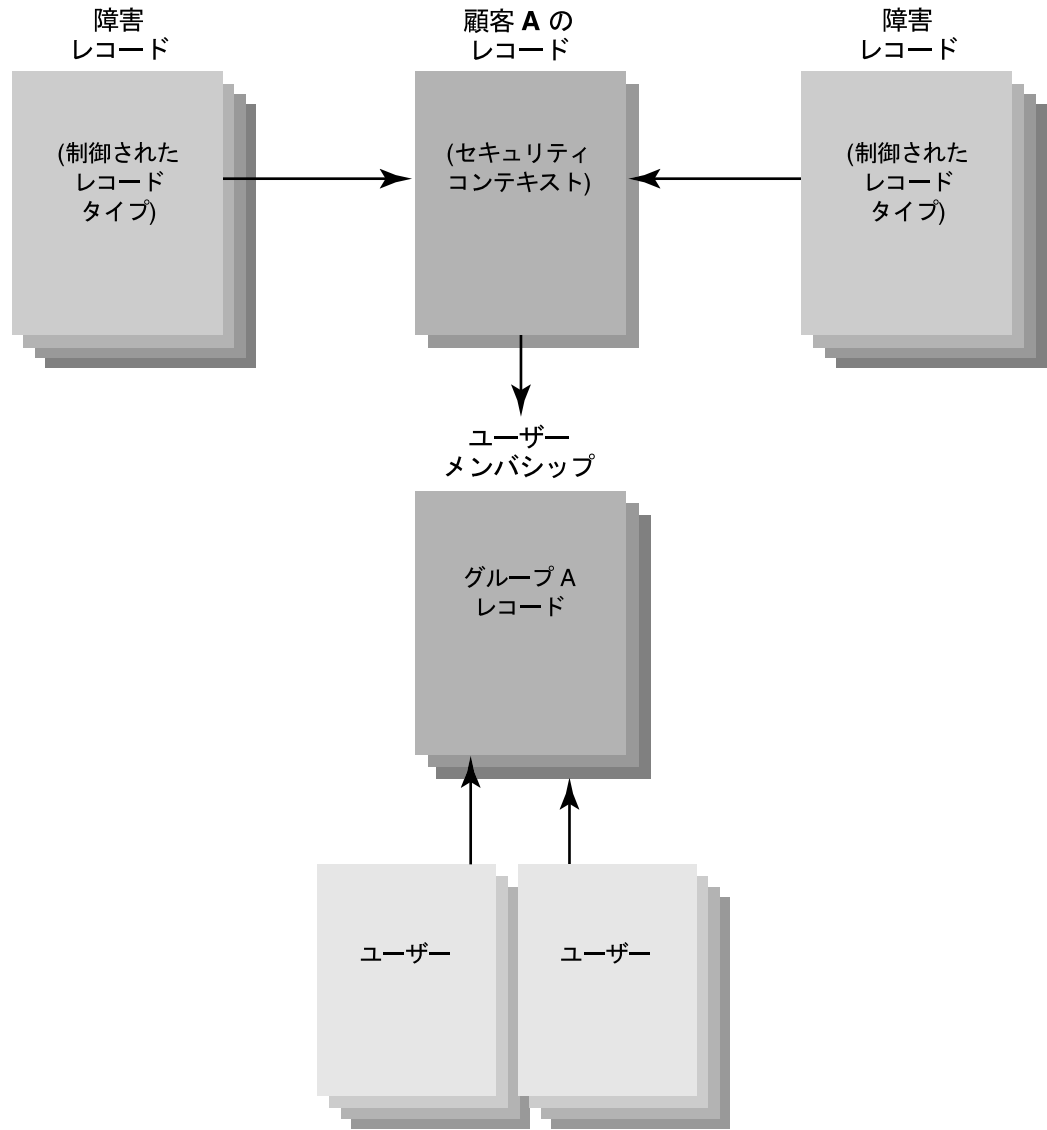
ユーザー アクセスを許可するためのレコードの編集

既存の制御されたレコード (たとえば **Defects**) を編集し、適用可能なセキュリティ コンテキスト レコード (このケースでは、**Customer** レコード) を選択します。

セキュリティ システムの設計

図 6 に、セキュリティ システムの設計で使用するエレメントを示します。ユーザーの帰属関係は、ユーザー グループ別に定義されます。ユーザー グループは、特定の顧客に関連付けられます。各障害レコードは 1 つまたは複数の顧客レコードを参照します。

図 6 セキュリティ システムのエレメント



ClearQuest で図 6 に示す設計を実装すると、エンド ユーザーには次のように適用されます。ユーザー A が ClearQuest にログインします。ユーザー A はグループ A (顧客 A と関連付けられています) のメンバーです。ユーザー A がクエリーを実行するかレコードを要求すると、ClearQuest により、顧客 A のセキュリティ コンテキストに基づいてすべての障害が自動的にフィルタリングされます。顧客 A に属する障害のみが表示されます。

セキュリティの例

この項では、ClearQuest でレコードを非表示化する方法の一例を説明します。この例では、顧客を Logic Equipment、Widgets Inc.、Modern Software の 3 社とします。Defect レコードタイプのレコードを制御し、顧客が実動データベースにアクセスして次の処理を行えるようにしたいと考えています。

- 障害の登録。
- 障害の状態の確認。事前定義されたクエリーを実行するか、新規クエリーを作成します。
- 既存の障害の編集。

Modern Software がデータベースにログインしているときに、Widgets Inc.、Logic Equipment、社内の QE チームによって登録された障害を参照できないようにします。また、Modern Software の顧客が ClearQuest クライアントにクエリーを作成した場合、顧客が参照できるクエリー結果を、Modern Software の顧客が登録した障害に関する情報のみに制限する予定です。

この例では、次の手順について説明します。

- スキーマのチェックアウト
- セキュリティ コンテキスト フィールドの作成
- フォームへのセキュリティ コンテキスト フィールドの追加
- スキーマ変更の適用
- ユーザー グループの作成
- セキュリティ コンテキスト レコードの登録
- 各セキュリティ コンテキスト レコードへのグループの関連付け
- レコードの編集によるグループへの権限の付与

メモ: 次の例で説明しているように、この手順には、さまざまなユーザー アクセス権限が必要です。この例で挙げられているすべての処理を完了するには、スーパー ユーザー権限が必要です。

スキーマのチェックアウト

この例では、事前定義された DefectTracking スキーマに基づくスキーマを使用します。このスキーマには、Defect レコード タイプと Customer レコード タイプがあります。また、スキーマがチェックアウトされていると仮定します。

セキュリティ コンテキスト フィールドの作成

メモ: 新規フィールドを作成したり、フィールドをレコード フォームに追加するには、スキーマ デザイナ権限が必要です。

この例では、障害レコードへのアクセスを制御します。そのため、Defect レコード タイプに、Customer レコードを参照するセキュリティ コンテキスト フィールドを作成します。フィールド グリッドでフィールドを作成し、そのフィールドをレコード フォームに追加して、スキーマの変更を適用します。

セキュリティ コンテキスト フィールドは、Reference フィールド タイプでなければなりません。複数のセキュリティ コンテキスト フィールドを追加できます。複数のセキュリティ コンテキスト フィールドを追加する場合に必要な処理は、該当するタイプのレコードを参照するため、グループのいずれかのメンバーになることのみです。

Defect レコード タイプにセキュリティ コンテキスト フィールドを作成するには

- 1 ClearQuest Designer で、[レコード タイプ] フォルダを展開し、[Defect] フォルダを展開します。次に、[フィールド] をダブルクリックします。
- 2 フィールド グリッドで、customer_defects という名前の新規フィールドを作成し、フィールド タイプとして [Reference] を選択します (自分のセキュリティ システム内で、新規フィールドを作成することも、既存の Reference タイプ フィールドを使用することもできます)
- 3 作成した customer_defects フィールドを右クリックし、[フィールドのプロパティ] をクリックします。

- 4 [フィールドのプロパティ] ダイアログ ボックスで、[参照先] リストから Customer レコードを選択します。

[参照先] リストで Customer レコード タイプをクリックすると、[セキュリティ コンテキスト] チェック ボックスが有効になります。

- 5 [セキュリティ コンテキスト] チェック ボックスをオンにします。

[セキュリティ コンテキスト] チェック ボックスをオンにすると、ClearQuest Designer により、セキュリティ コンテキスト (Customer) レコード タイプの登録フォームとデフォルト フォームに [Ratl_Security] タブが追加されます。ClearQuest クライアントでこのタブを使用して、レコードを参照できるグループを選択します (各自のシステムで [Ratl_Security] タブの名前を変更できます。詳細については、162 ページの「タブ名の変更」を参照してください)。

メモ: レコード タイプには、複数のセキュリティ コンテキスト フィールドを追加できます。たとえば、Customer レコード タイプを参照するセキュリティ コンテキスト フィールドと、Quality_Assurance レコード タイプを参照する別のセキュリティ コンテキスト フィールドを追加できます。Customer レコード タイプに顧客を追加し、Quality_Assurance レコードに Quality Assurance グループのメンバーを追加すると、これらのレコード タイプのグループ リストに属するユーザーは誰でも、セキュリティの制御下にあるレコードにアクセスできます。

ログインしたユーザーに基づいて、フィールドに値を自動的に入力するフックを含めることができます。フックを含めると、有効な値がフィールドに必ず入力されます。特定のアクションをユーザーに実行させないようにすることもできます。たとえば、障害のクローズを内部のユーザーのみに許可し、顧客によるレコードの削除を制限できます。詳細については、197 ページの「その他の ClearQuest セキュリティ機能の使用法」を参照してください。

フォームへのセキュリティ コンテキスト フィールドの追加

フィールド グリッドで `customer_defects` フィールドを作成したら、そのフィールドを Defect レコード フォームに追加する必要があります。

作成した `customer_defects` フィールドを Defect レコード フォームに追加するには

- 1 ClearQuest Designer で、[レコード タイプ] フォルダを展開し、[Defect] フォルダを展開します。次に、[フォーム] フォルダをクリックし、[Defect_Base] をダブルクリックします。
- 2 [フィールド リスト] で、[`customer_defects`] フィールドをクリックし、それをフォーム上にドラッグします。

スキーマ変更の適用

新規フィールドを追加したら、スキーマをスキーマ レポジトリにチェックインし、スキーマ変更をユーザー データベースに適用する必要があります。この手順を実行すると、変更を元に戻すことはできません。詳細については、82 ページの「スキーマの変更手順」を参照してください。

ユーザー グループの作成

メモ: ユーザーやグループを作成するには、ユーザー管理者権限が必要です。

次に、Customer セキュリティ コンテキスト レコードに関連付けるグループを作成し、そのグループにユーザーを追加し、新しいユーザー情報を使用してユーザー データベースを更新します。この例では、Widgets Inc.、Modern Software、Logic Equipment のユーザー グループを作成し、そのグループにユーザーを追加します。

メモ: 各自のセキュリティ システムでは、既存のグループも使用できます。また、内部ユーザーが登録したすべてのレコードを参照できるグループ、すべての企業が登録したすべてのレコードを参照できるグループ、登録元が内部ユーザーまたは顧客であるかどうかに関係なくすべてのレコードを参照できるグループなど、追加のグループも作成できます。

グループ作成の詳細については、180 ページの「新しいユーザー グループの作成」と 182 ページの「グループへのユーザーの追加」を参照してください。

セキュリティ コンテキスト レコードの登録

メモ: セキュリティ コンテキスト レコードを登録するには、セキュリティ管理者権限が必要です。

次に、データベースへのアクセスを許可する各企業 (Widgets Inc.、Modern Software、Logic Equipment) の Customer レコードを登録します。

ClearQuest クライアントで、次の手順に従います。

- 1 [アクション] メニューの [新規作成] をポイントし、[Customer] をクリックして、[Submit Customer] ダイアログ ボックスを開きます。
- 2 Widgets Inc.、Modern Software、Logic Equipment の顧客レコードを登録します。

すべてのレコードを参照できるグループを作成することもできます。その場合は、このグループを各顧客レコードに追加します。

各セキュリティ コンテキスト レコードへのグループの関連付け

次に、特定のグループを各セキュリティ コンテキスト レコードに関連付ける必要があります。たとえば、Widgets Inc.、Logic Equipment、Modern Software の各企業用に登録した顧客レコードに関連付けるユーザー グループを選択します。これらのグループには、レコードを参照する権限と変更する権限を許可するユーザーが含まれます。グループを選択するには、セキュリティ管理者またはスーパー ユーザー権限が必要です。

グループを Customer レコードに関連付けるには

- 1 ClearQuest クライアントで、すべての Customer セキュリティ コンテキスト レコードを一覧表示する「すべての顧客」というクエリーを作成して実行します。
- 2 Widgets Inc. の顧客レコードを開き、[Ratl_Security] タブをクリックします。[アクション] メニューの [Modify] をクリックします。
- 3 [Ratl_Security] タブで、Widgets Inc. グループを選択し、[追加] をクリックします。[適用] をクリックします。
- 4 Modern Software のレコードと Logic Equipment のレコードについても、手順 2 から手順 3 を繰り返します。

レコードの編集によるグループへの権限の付与

次に、顧客がアクセスできる各障害レコードを編集して、顧客を `customer_defects` フィールドに割り当てます。これにより、Logic Equipment、Widgets Inc.、Modern Software の各グループにレコードへのアクセス権が与えられます。この手順は、セキュリティ コンテキスト レコードの値をセキュリティ コンテキスト フィールドに割り当てます。

ClearQuest クライアントで、次の手順に従います。

- 1 制御する障害レコードが格納されているデータベースにログインします。
- 2 「すべての障害」クエリーを実行して、すべての障害レコードを表示します。
- 3 制御するレコードを選択して編集します。たとえば、障害レコード「ログイン画面でのスペリングエラー」を選択します。
- 4 `[customer_defects]` リストで、このレコードへのアクセス権を与える顧客を選択します。たとえば、Widgets Inc. を選択して、Widgets_Inc グループのユーザーにこのレコードへのアクセス権を与えます。
- 5 Logic Equipment グループと Modern Software グループについても、各レコードを編集して、権限を許可します。

これで、ClearQuest でレコードを非表示にする例は完了です。

この例では、Widgets Inc. の顧客は、データベースにログインし、次の作業を実行できるようになります。

- 既存の障害レコードを編集できます。顧客選択リストには、Widgets Inc. のみが表示されます。
- Customer レコードタイプについて、クエリー、グラフ、レポートを実行し、Widgets Inc. のレコードのみを参照できます。

その他の ClearQuest セキュリティ機能の使用法

次の処理を行うと、ClearQuest 内でデータを保護して、権限のあるユーザーのみがレコードの参照や変更を行えるように設定できます。

- フィールドへのアクセス制限
- アクションへのアクセス制限
- ダイアログ タブへのアクセスの制限
- パスワード保護機能の追加

フィールドへのアクセス制限

次の操作で、フィールドへのユーザー アクセスを制限できます。

- フィールド動作を読み取り専用に定義します。ユーザーは、読み取り専用フィールドの内容を参照できますが変更できません。
- フィールドの権限フックを使用して、ユーザー アクセスのレベルを決定します。

アクションへのアクセス制限

指定した条件に基づいてアクセスを制御するアクセス制御フックを追加するか、アクションを実行できるユーザー グループを選択できます。

ダイアログ タブへのアクセスの制限

複数のフィールドをダイアログ タブに配置し、選択したユーザー グループのみがタブにアクセスできるように設定すると、その複数のフィールドに対するアクセスを制限できます。

パスワード保護機能の追加

テキスト ボックス コントロールを使用してフォームにパスワード フィールドを追加できます。このフィールドに入力した文字はエコー出力されません。このフィールドにユーザーが入力した各文字は、アスタリスク (*) に置き換わります。

フォームにパスワード フィールドを追加するには

- 1 テキスト ボックス コントロールを追加します。
- 2 追加したテキスト ボックス コントロールをダブルクリックして、プロパティ シートを表示します。
- 3 [拡張] タブをクリックして、[パスワード/エコーなし スタイル] をクリックします。

ワークフローのカスタマイズ におけるフックの使用法

10

Rational ClearQuest で提供されるさまざまな事前定義フックを使用して、ワークフローを実装できます。また、ClearQuest のアプリケーション プログラミング インターフェイス (以下 API) を使用して、事前定義されたフックのカスタマイズ、独自のフックの作成、ClearQuest データベース上でタスクを実行する外部アプリケーションの作成を行うこともできます。

この章では、フックと ClearQuest API を使用方法について説明します。この章の内容は、次のとおりです。

- 200 ページの「フックの理解」
- 203 ページの「フィールド フックの操作」
- 214 ページの「アクション フックの操作」
- 218 ページの「フィールド フックとアクション フックの実行順序」
- 220 ページの「レコード スクリプトの操作」
- 225 ページの「グローバル スクリプトの操作」
- 227 ページの「外部アプリケーションの作成」
- 228 ページの「ClearQuest API の使用法」
- 229 ページの「一般的な API 呼び出し」
- 231 ページの「フック スクリプト内のテキストの検索」

関連項目

- ClearQuest API の詳細については、Windows の [スタート] ボタンをクリックし、[プログラム]、[Rational Software]、[Rational ClearQuest]、[ClearQuest API リファレンス] の順にクリックしてください。『Rational ClearQuest API Reference』は、すべての ClearQuest API 呼び出しに関するオンライン リファレンス ガイドです。
- Web 環境でのフックの使用法については、「第 11 章 ClearQuest Web の管理」を参照してください。
- ClearQuest のサンプル フック リポジトリは、ユーザー同士が複数のフック スクリプトをやり取りする場所を提供します。リポジトリは <http://clearquest.rational.com/> にあります。データベースにアクセスするには、次のユーザー名とパスワードを入力します。

ユーザー名 : hooks

パスワード : password

サンプル フック データベースへのリンクをクリックします。そこで既存のフック スクリプトのデータベースを参照できます。

フックの理解

フックは、トリガと同様、コードのためのエントリ ポイントです。ClearQuest では、ユーザーによる ClearQuest の操作方法をカスタマイズするため、コードを指定された回数だけ実行します。ClearQuest では、4 種類のフックをサポートしています。

- フィールド フック

フィールド フックを使用すると、実行時に 1 つのフィールド値をチェックし、必要に応じてほかのフィールドを調整できます。たとえば、フィールドの内容を確認したり、フィールドにデフォルト値を割り当てることができます。

- アクション フック

アクション フックを使用すると、レコードのライフサイクルでの主要ポイントでタスクを実行できます。一般に、アクション フックはレコード全体に影響を及ぼすイベントに関連付けられます。これに対して、フィールド フックは特定のフィールドに影響を及ぼすイベントに関連付けられます。たとえば、レコード全体を確認したり、アクション完了時に通知を送信できます。

- レコード スクリプト

レコード スクリプトを使用すると、実行時に特定のタスクを実行できます。レコード スクリプトは 1 つのレコード タイプ専用であり、通常はフォーム コントロールに関連付けられています。

- グローバル スクリプト

グローバル スクリプトを使用すると、スキーマ内のすべてのレコード タイプで共有できるルーチンのライブラリを定義できます。たとえば、電子メールによる通知など、すべてのレコード タイプのすべてのフックから呼び出すことができるサブルーチンを作成できます。

フックについての重要な注意事項

フックを計画するときには、次の点を考慮してください。

- フックは、VBScript (Windows の場合) と Perl (Windows と UNIX の場合) で作成できます。デフォルト設定の場合、Windows 上では ClearQuest は VBScript 形式のフックを実行します。Windows 上で Perl 形式のフックを実行する方法の詳細については、87 ページの「スクリプト言語の選択」を参照してください。
- フックはスーパー ユーザー権限を使用して実行されます。したがって、通常のアクセス制御またはフィールド動作制限の対象にはなりません。つまり、フックを使用すると、通常は読み取り専用であるフィールドの値の設定とリセットを行えます (ただし、履歴フィールドなどの ClearQuest システム フィールドをリセットすることはできません)。必須フィールドはフックを使用しても必須フィールドのままです。

- 外部コードを利用してフック機能を拡張できます。VBScript からは COM オブジェクトにアクセスできます。サードパーティのパッケージを使用すると、Perl からも COM オブジェクトにアクセスできます。さらに、Perl ではさまざまなサードパーティ パッケージを利用できます。たとえば、ダイアログ ボックスを使用してユーザーにフックを対話的に操作させることができます。また、外部ファイルからのデータの読み込みや、外部ファイルへのデータの書き込みも可能です。クライアント コンピュータに適切なサードパーティ オブジェクトをインストールするのは、ClearQuest 管理者の責任です。
- スクリプトに SQL コードを組み込むことは可能ですが、最適な結果を得るために、スクリプト コード内でのクエリーの実行とデータの取得には ClearQuest API を使用することをお勧めします。なぜなら、ClearQuest では、レコードタイプとフィールドの名前を変更できるため、SQL コードを組み込んだ場合には問題が発生することがあるためです。
- ClearQuest Web 上でフックを実行する場合には、そのフックを Web 環境でテストしてください。243 ページの「フックを使用した Web セッションの検出」を参照してください。
- フック コードのテストとデバッグを実行してください。これにより、データベースに不正な値を書き込んだり、無限ループの処理や存在しない入力を待機してプログラムがハングする事態を避けることができます。デバッグ情報 (OutputDebugString メソッドの出力) を参照するには、ClearQuest に添付されている dbwin32.exe を使用します。

メモ: フックの実行中に例外が発生した場合に詳細なデバッグ情報を入手するには、Windows レジストリ キー
 HKEY_CURRENT_USER¥SOFTWARE¥RationalSoftware¥ClearQuest¥2.0¥Common¥
 HookDebug を作成し、DWORD 値「1」を設定します。デフォルトのフック例外デバッグ
 メッセージ (簡潔な情報) に戻すには、この値を「0」に変更します。

スクリプトの作成

フィールドフックとアクションフックには、VBScript または Perl で記述したスクリプトを追加できます。グローバル スクリプトまたはレコード スクリプトを作成する場合は、必ず VBScript または Perl を使用してください。

格納や参照を簡単に行うために、同一スキーマ内に VBScript と Perl の両方のスクリプトを記述することが可能です。ただし、スキーマが実行するのは、指定された言語のスクリプトのみです (87 ページの「スクリプト言語の選択」を参照)。

スクリプトの記述または編集は、スクリプト エディタ内で行います。新規スクリプトを定義すると、ClearQuest Designer によって、そのフックの呼び出し構文がスクリプト エディタ ウィンドウに自動的に追加されます。呼び出し構文は灰色表示され、編集はできません。また、ClearQuest Designer によって、本文テキスト例も追加されます。この本文テキスト例は、必要に応じて編集できます (本文テキストの 1 行めはコメント化されているので、この行を実行するにはコメント マーカーを削除してください)。

メモ:

- ClearQuest Designer には、VBScript 用と Perl 用のスクリプト エディタが別々に用意されています。ClearQuest ウィンドウのタイトル バーには、エディタ タイプが表示されます。適切なエディタを使用していることを確認してから、コードを記述してください。

スクリプトを使用するための動作状況

ClearQuest は、VBScript フックと Perl フックを記述するプロセスを簡略化するために、一貫した動作状況を提供します。ClearQuest は、ClearQuest フックが VBScript スクリプトまたは Perl スクリプトを呼び出す前でも、セッション オブジェクトを作成し、ユーザーがシステムにログインします。グローバル スクリプトを含むすべてのフックが現在のレコードのコンテキストから実行されるため、ClearQuest は、そのレコードに対応するエンティティ オブジェクトを自動的に提供します (グローバル スクリプトでは、呼び出し元のフックに関連付けられたエンティティ オブジェクトを共有します)。

スクリプト内では、前に識別子を指定しなくてもエンティティのメソッドを呼び出すことができます。たとえば、次の方法でエンティティの `GetSession` メソッドを呼び出すことができます。

```
set curSession = GetSession
```

この方法でメソッドを呼び出すと、ClearQuest は、フックに渡される暗黙的なエンティティ オブジェクトのメソッドを呼び出しているものと自動的に見なします。明示的にエンティティ オブジェクトを参照する場合は、オブジェクトの識別子としてレコード タイプ名を使用します。以下の例で示すように、この識別子を使用すると、同時に複数のエンティティ オブジェクトを処理するコードが簡単に識別できます。この例では、エンティティに別のエンティティの重複としてマークします。

```
set curSession = GetSession
idName = GetFieldValue("id").GetValue
set currentObj = curSession.GetEntity("defect", idName)
' ID "SAMPL00000031" のエンティティをこのエンティティの重複としてマークします。'
"duplicate" という名前のアクションを使用します。
set dupEntityObj = curSession.GetEntity("defect", "SAMPL00000031")
curSession.MarkEntityAsDuplicate dupEntityObj, currentObj, "duplicate"
```

スクリプトはフィールドの動作に影響を与えることがあるため、フック コードの設計とテストは慎重に行う必要があります。たとえば、フックがフィールドに何らかの値を含めることを要求した場合、フィールドの動作を `MANDATORY` に設定していなくてもフィールドは必須となります。

フィールド フックの操作

フィールドフックは、レコード内の特定のフィールドに影響を及ぼすイベントのために使用されます。フィールドフックを使用すると、初期値を設定したり、フィールド値が変更されたときにイベントに応答できます。また、管理者が指定したユーザー グループだけがフィールド値を変更できるようにアクセス権を管理したり、ユーザーが指定した値を確認できます。

フィールドフックの適用範囲は、現在のレコード内の現在のフィールドです。ClearQuest が提供するフィールドフックのタイプは次のとおりです。

フィールド フック	説明
選択リスト	リーガル値のセットを返します。このフックは、リスト形式のコントロール (リスト ボックスやコンボ ボックスなど) を使用して表示されるフィールドと共に使用します。 定数または動的リストを使用すると、スクリプトで記述しなくても値を表示できます。207 ページの「フィールドの選択リストの作成」を参照してください。
デフォルト値	フィールドの初期値を設定します。このフックは、Submit アクションの開始時に呼び出されます。 スクリプト サブルーチンを使用してデフォルト値フックを作成できます。デフォルト値として定数値を割り当てることもできます。
権限	フィールドへのユーザーのアクセス権を示す BehaviorType 定数の 1 つを返します。このフックを使用して、ワークフローまたはセキュリティ (あるいはその両方) を設定します。列挙定数については、オンラインの『Rational ClearQuest API Reference』を参照してください。 フィールドに権限フックを追加する場合には、動作グリッドを変更して、フィールドの動作の少なくとも 1 つを USE_HOOK に設定する必要があります。この設定が正しくない場合には、確認エラーが発生します。
確認	フィールドの内容を確認します。このフックは、値が変更されると呼び出され、レコードをデータベースにコミットする前に、フィールドの内容の有効性に関するフィードバックをユーザーに直ちに返します。
変更された値	フィールドでの値の変更に応答します。このフックを使用して、ほかのフィールド (依存先リストなど) の更新が実行されます。 このフックを実行すると、ClearQuest によってフィールドの確認フック (存在する場合) が呼び出され、スクリプトで変更されたすべてのフィールドが確認されます。

関連項目

- 118 ページの「フックの追加によるフィールドのカスタマイズ」
- 204 ページの「フィールドフックの追加」
- 218 ページの「フィールドフックとアクションフックの実行順序」
- 326 ページの「フィールドフックの例」

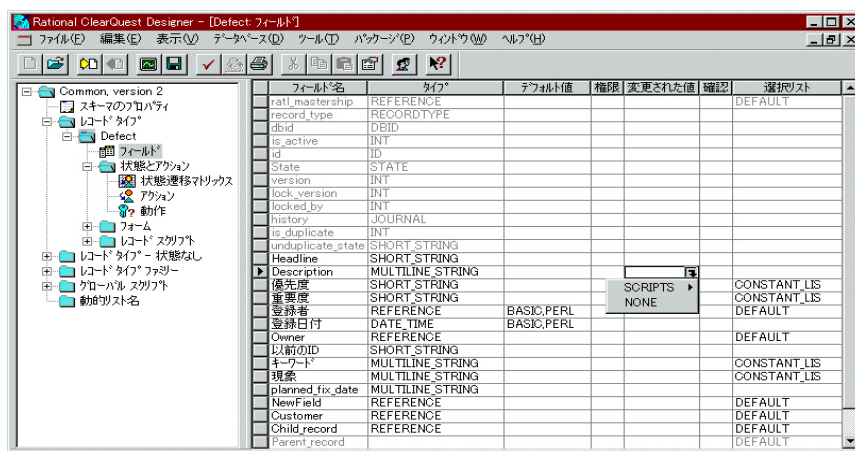
フィールドフックの追加

選択リスト、フィールド確認、デフォルト値、権限などの機能を備えることで、フィールドの動作をカスタマイズする場合は、フィールドフックを追加できます。ClearQuest は、218 ページの「フィールドフックとアクションフックの実行順序」で指定されたルールに基づいてフックを実行します。

フィールドフックを追加するには

- 1 ワークスペースで、[レコードタイプ] フォルダを展開し、目的のレコードタイプを展開します。次に、[フィールド] をダブルクリックしてフィールドグリッドを表示します。

フィールドグリッドには、事前定義されたフィールドフックのタイプごとの列があります。



- 2 目的のフィールドのフック列をクリックします。下矢印をクリックし、[SCRIPTS] メニューの [BASIC] または [PERL] をクリックします。

[インスタント編集モード] が有効な場合は、ClearQuest Designer によって適切なスクリプトエディタが自動的に開きます。[インスタント編集モード] が無効な場合は、セルをダブルクリックしてスクリプトエディタを開きます。

メモ: 選択したフックのタイプによって ClearQuest が開くエディタが異なります。たとえば、[SCRIPTS] メニューの [BASIC] をクリックすると、Basic スクリプトエディタが開きます。ClearQuest 一方、[CONSTANT LIST] を選択すると、[選択リスト] ダイアログボックスが開きます。

- 3 フックを編集します。ClearQuest によって、そのフックの呼び出し構文がスクリプトエディタに自動的に追加されます。呼び出し構文は灰色表示され、編集はできません。
- 4 [フック]メニューの[コンパイル]をクリックして、スクリプトにエラーがないか確認します。ClearQuest では、エラーがあると、ワークスペースの下部の[確認]ペインにエラーが表示されます。
- 5 スクリプトエディタを閉じます。ClearQuest Designer によって変更が保存されます。

メモ: フィールドに権限フックを追加する場合は、動作グリッドでそのフィールドの値を調整してください。権限フックが機能するためには、フィールドの動作を USE_HOOK に設定する必要があります。

フィールド フックの編集

フィールド フックを編集するには

- 1 ワークスペースで、[レコードタイプ]フォルダを展開し、目的のレコードタイプを展開します。次に、[フィールド]をダブルクリックしてフィールドグリッドを表示します。
- 2 編集対象のフックが設定されたフィールドの列をダブルクリックします。フックエディタを開きます(または、列を右クリックし、ショートカットメニューの[編集<フックタイプ>]をクリックすることもできます。たとえば、スクリプトを編集するには、[スクリプトの編集]をクリックします)。
- 3 フックを編集します。
- 4 [フック]メニューの[コンパイル]をクリックして、スクリプトにエラーがないか確認します。ClearQuest では、エラーがあると、ワークスペースの下部の[確認]ペインにエラーが表示されます。
- 5 スクリプトのコンパイルが正常に終了したら、エディタを終了します。ClearQuest Designer によって変更が保存されます。

フィールド フックの削除

フィールド フックを削除するには

- 1 ワークスペースで、[レコードタイプ]フォルダを展開し、目的のレコードタイプを展開します。次に、[フィールド]をダブルクリックしてフィールドグリッドを表示します。
- 2 フィールドグリッドで、削除対象のフックがあるセルをクリックし、リストから[なし]をクリックします(フィールドの型がREFERENCE またはREFERENCE_LIST で、フックが選択リストフックの場合は、[DEFAULT]をクリックします)。
- 3 削除するかどうかを確認するメッセージが表示されたら、[はい]をクリックします。

メモ: フックを削除した後は、現在使用しているバージョンのスキーマにチェックインしても、削除したフックを復元できません。ただし、そのフックが以前のバージョンのスキーマに格納されている場合は、以前のバージョンを開き、そのフックをスキーマにコピーできます。

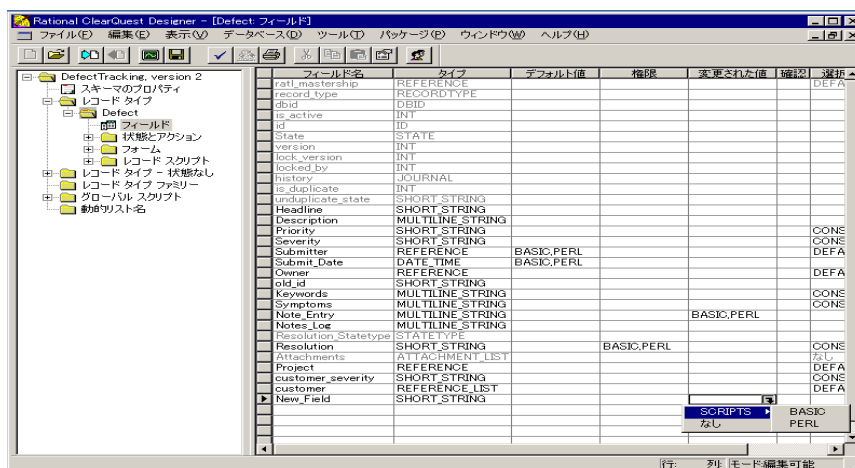
フィールド間の依存関係の作成

親フィールドの値が変更された場合には子フィールド (依存フィールド) の値も変更されるように、2つのフィールドの間で依存関係を作成できます。

メモ: スキーマを作成するときは、フィールド依存関係を慎重に計画してください。依存フィールドにはフックの使用が欠かせません。フックが正しく記述されていないと、実行時エラーが発生することがあります。スクリプト フックをユーザーに提供する前に、十分なテストを行ってください。

2つのフィールドの間で依存関係を作成するには、変更された値フックを親フィールドに追加します。

- 1 フィールド グリッドを表示します。



- 2 親フィールドの [変更された値] セルをクリックします。
- 3 下矢印をクリックし、[SCRIPTS] メニューの [BASIC] または [PERL] をクリックします。

[インスタント編集モード] が有効な場合は、ClearQuest Designer によってスクリプト エディタが自動的に開きます。インスタント編集を有効化または無効化するには、[編集] メニューの [インスタント編集モード] をクリックします。スクリプト エディタが自動的に表示されない場合には、セルをダブルクリックすると開きます。

メモ: ClearQuest Designer では、使用しているエディタのタイプが Designer ウィンドウのタイトル バーに表示されます。正しいエディタを使用していることを確認してから、コードの追加や編集を行ってください。

- 4 親フィールドの値を取得し、その値を使用して子フィールドの値を設定するスクリプトを、スクリプト エディタで入力します。子フィールドの値を基準にして親フィールドの値を設定する例については、337 ページの「親レコードの値を設定するアクション フック」を参照してください。
- 5 スクリプトの編集が終了したら、[フック] メニューの [コンパイル] をクリックして、スクリプト コードの構文をチェックします。

ClearQuest Web での依存フィールドの有効化

依存フィールドを含むフォームを ClearQuest Web に表示する場合、次の手順を実行する必要があります。

- 1 フィールドをレコード フォームに追加するときに、親フィールドとその依存フィールドに次のフォーム コントロールのいずれかを使用します。複数のコントロールを組み合わせで使用することもできます。
 - ドロップダウン リスト ボックス
 - コンボ ボックス
 - ドロップダウン コンボ ボックスコントロールの詳細については、151 ページの「フォーム コントロールの操作」を参照してください。
- 2 コントロールをフォームに追加したら、親フィールドのコントロールを右クリックし、ショートカット メニューの [プロパティ] をクリックします。
- 3 [Web 依存先フィールド] タブで、[使用可] リストから適切な子フィールドを選択し、[選択] リストに追加します。

フィールドの選択リストの作成

選択リストを使用して、フィールドのオプション リストを表示できます。オプションの定数リストを表示したり、スクリプトを使用してリーガル値のリストを生成したり、動的リストを設定できます。

フィールドの選択リストを作成するには

- 1 フィールド グリッドを開きます。
- 2 目的のフィールドの [選択リスト] セルをクリックします。

3 下向きの矢印をクリックし、次のいずれかをクリックします。

- 選択リストのスクリプトを作成するには、[SCRIPTS] をクリックします。ユーザーが別のフォーム コントロールで特定の値を選択したときに値が変更される依存先リストを定義できます。その例については、327 ページの「依存先リストを作成するフック」を参照してください。
- 選択リストに対して値のセットを指定するには、[CONSTANT LIST] をクリックします。
- スキーマ デザイナ権限、動的リスト管理者権限、スーパー ユーザー権限のいずれかを持つユーザーが ClearQuest クライアントを使用して値を変更できる選択リストを作成するには、[DYNAMIC LIST] をクリックします。動的リスト フィールドフックを使用すると、スキーマを変更せずにリスト値を更新できるため、リストを頻繁に変更する場合には便利です。208 ページの「動的選択リストの作成」を参照してください。
- フィールド タイプが REFERENCE と REFERENCE_LIST の場合に限り、[なし] (デフォルト) をクリックします。これらのフィールドタイプのデフォルトの選択リストとして、指定したタイプのアクティブなレコードがすべて表示されます。

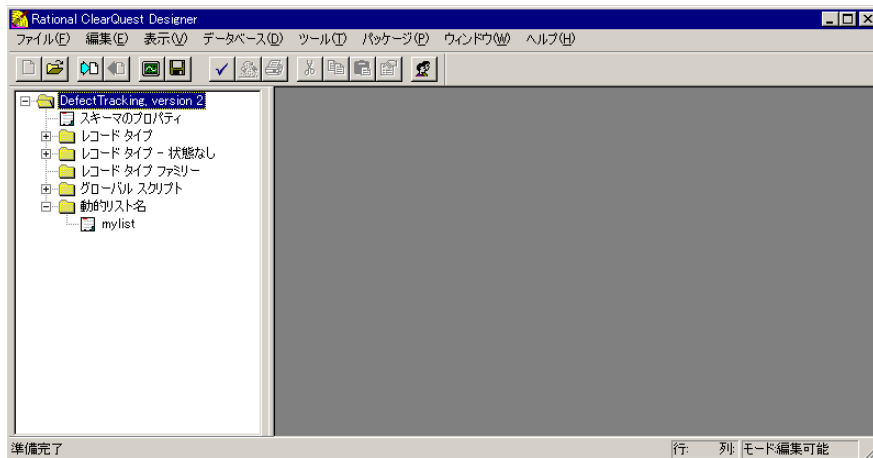
動的選択リストの作成

動的リストは、状態のリストなどのドロップダウン リストにすることができます。スキーマを編集しないでリストに値を追加する場合は、動的リストを使用すると便利です。スキーマ デザイナ権限、動的リスト管理者権限、スーパー ユーザー権限のいずれかを持つユーザーは、どの ClearQuest クライアントからでも動的選択リストに値を追加できます。

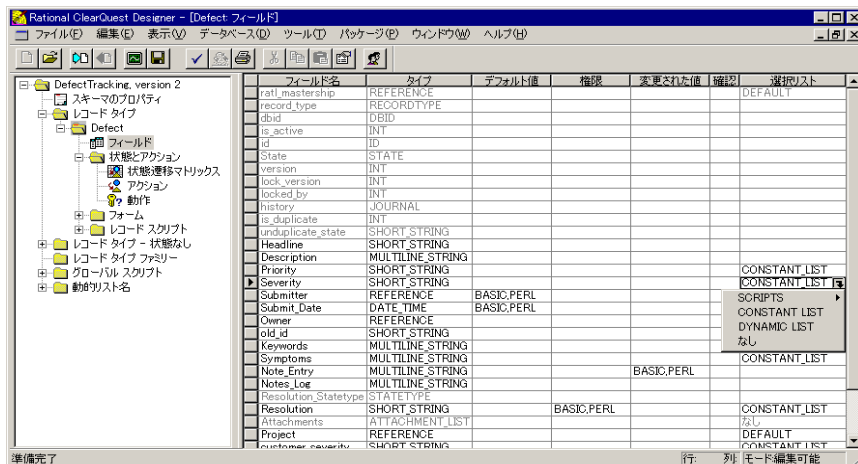
メモ: 1 つの動的リストを複数のフィールドで使用できます。このことは、レコード タイプに関係ありません。このフィールドの内容を更新すると、動的リストに関連付けられているすべてのフィールドが更新されます。

フィールドの動的リストを作成するには、まず、リストを作成し、それを ClearQuest Designer のフィールドに関連付けます。次に、ClearQuest クライアントでリストの値を定義します。手順は以下のとおりです。

- 1 ワークスペースで、[動的リスト名] を右クリックし、ショートカット メニューの [追加] をクリックします。
- 2 動的リストの名前を入力します。



- 3 フィールド グリッドで、目的のフィールドの [選択リスト] セルをクリックします。次に、ドロップダウン矢印アイコンをクリックし、[DYNAMIC LIST] をクリックします。



- 4 [動的リスト名] ダイアログ ボックスで、作成した動的リスト名をクリックします。



- 5 必要に応じて、[選択リストを再計算] または [リストに制限] をオンにします。
211 ページの「選択リストの動作の定義」を参照してください。

- 6 [OK] をクリックします。

変更内容を保存するには、まず、スキーマをテストし、スキーマをスキーマ レポジトリにチェックインします。次に、スキーマ変更をユーザー データベースに適用します。82 ページの「スキーマの変更手順」を参照してください。

- 7 動的リストの値を定義するには、ClearQuest クライアントを使用します。210 ページの「動的リストの編集」を参照してください。

動的リストの編集

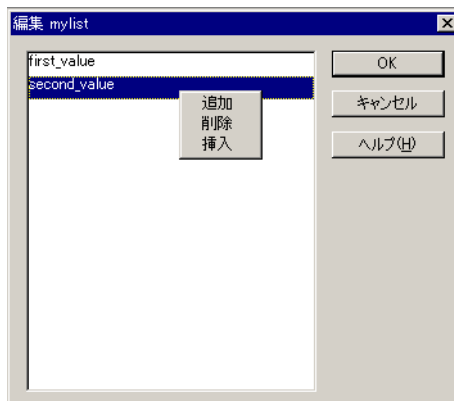
まず、ClearQuest Designer で動的リストを追加します (208 ページの「動的選択リストの作成」を参照)。次に、ClearQuest クライアントで動的リストの値を定義します。

動的リストを編集するには、スキーマ デザイナ権限、動的リスト管理者権限、スーパー ユーザー権限のいずれかが必要です (169 ページの「ClearQuest ユーザー権限」を参照)。

動的リストを編集するには

- 1 ClearQuest クライアントで、[編集] メニューの [名前付きリスト] をクリックし、編集するリストをクリックします。

- 2 [編集<名前付きリスト>] ダイアログ ボックスを使用して、次の手順で値の追加、削除、挿入を行います。



- 値をリストに追加するには、リスト ボックスを右クリックし、[追加] をクリックします。次に、追加する値の名前を入力し、[Enter] を押します。必要に応じて値の追加を続けます。
- 値をリストから削除するには、値を右クリックし、[削除] をクリックします。
- 値をリストに挿入するには、既存の値を右クリックし、[挿入] をクリックします。既存の値の上方に新しい行が挿入されます。挿入する値の名前を入力し、[Enter] を押します。

- 3 [OK] をクリックします。

選択リストの動作の定義

定数値、定数リスト、動的リスト フックを定義するときには、次の動作を選択できます。

- 選択リストにない値をユーザーが入力しないようにするには、[リストに制限] をオンにします。リスト内にない値は、すべて確認エラーになります。[リストに制限] をオフにした場合には、選択リストにない値をユーザーが入力できます。
- リストが別の値の変更に依存する場合に、ClearQuest で実行時にリストを再初期化するには、[選択リストを再計算] をオンにします。スクリプトで記述された選択リストでは、再計算処理によってパフォーマンス オーバーヘッドが発生することがあります。[選択リストを再計算] をオフにした場合には、ClearQuest は各アクションの開始時にだけ値を更新します。

フィールドのデフォルト値の指定

場合により、フィールドのデフォルト値を指定する必要があります。デフォルト値には、定数を指定することができます。また、スクリプトを使用してデフォルト値を計算することもできます。

フィールドのデフォルト値を指定するには

- 1 フィールドグリッドで、変更するフィールドの [デフォルト値] セルをクリックします。
- 2 下矢印をクリックし、使用するフックのタイプをクリックします。

定数値を指定する場合は、[CONSTANT] を選択し、[定数] フィールドに値を入力します。

スクリプトを記述する場合は、[SCRIPTS] メニューの [BASIC] または [PERL] をクリックし、デフォルト値を設定するコードを次の行の後に入力します (Perl の場合は行が # で始まります)。

```
REM Set the initial value of the field here. Example:
```

```
REM SetFieldValue fieldname, 12345
```

フックを削除するには、[なし] をクリックします。

メモ:

- スクリプトについては、そのスクリプトをフィールドに関連付ける前に、スクリプトをコンパイルして構文エラーがないか確認します。
- デフォルト値スクリプトの例については、330 ページの「フィールド デフォルト値フックの例」を参照してください。

ユーザーが入力したフィールド値の確認

確認フックを使用して、フィールドにユーザーが有効な情報を入力しているかどうかを確認できます。フィールドに無効な情報を入力した場合、ClearQuest から、有効な情報を入力するよう求めるメッセージが表示されます。

- 1 ワークスペースで、[レコードタイプ] フォルダを展開し、目的のレコードタイプを展開します。次に、[フィールド] をダブルクリックしてフィールドグリッドを開きます。
- 2 フィールドグリッドで、変更するフィールドの [確認] セルをクリックします。下矢印アイコンをクリックして使用可能なフックを表示し、[SCRIPTS] メニューの [BASIC] または [PERL] をクリックします。

[インスタント編集モード] が有効な場合は、ClearQuest Designer によってスクリプトエディタが自動的に開きます。[インスタント編集モード] が無効な場合は、フィールドの [確認] セルをダブルクリックしてスクリプトエディタを開きます。

メモ: BASIC と Perl には、それぞれ専用のスクリプトエディタがあります。ClearQuest Designer では、Designer ウィンドウのタイトルバーにエディタのタイプが表示されます。適切なエディタを使用していることを確認してから、コードを編集してください。

- 3 入力値を確認するコードを、次に示す行の後に入力します (Perl の場合は、行が # で始まります)。

```
REM Return a non-empty string explaining why the
REM field's current value is not permitted.
REM Or, if it is valid, return an empty string value.
REM Example:
REM Dim value_info
REM Set value_info = GetFieldValue(fieldname)
REM If Len(value_info.GetValue()) < 10 Then
REM resolution_date_Validation = "Must be at least 10 chars long"
REM End If
```

例を次に示します。

フィールド名が「**user_number**」、データ型が INT の場合、次のコードにより、1 ～ 100 の値だけをユーザーが入力できるようにします。

```
REM Return a non-empty string explaining why the field's current
REM value is not permitted
REM Or, if it is valid, return an empty string value.

value = GetFieldValue(fieldname).Get Value()
if Not IsNumeric(value)
    user_number_Validation="フィールドに番号が入力されていません。"
Else If (value < 1) or (value > 100) then
    user_number_Validation="ユーザー番号は、1 ～ 100 の間の値でなければなりません。"
end if
```

- 4 [フック] メニューの [コンパイル] をクリックします。スクリプトがコンパイルされ、構文エラーがないかどうかを確認されます。
- 5 スクリプトエディタを閉じます。

メモ:

- フィールドまたはアクションに新しい BASIC フックまたは PERL フックを定義すると、ClearQuest により、そのフックの呼び出し構文がスクリプトエディタウィンドウに自動的に追加されます。スクリプトエディタウィンドウを開くと、呼び出し構文は灰色で表示され、編集できません。
- フックはフィールドの動作に影響を与えることがあるため、慎重に計画し、テストしてからユーザーに提供してください。たとえば、このユーザー入力フックの例では、動作グリッドの設定に関係なく、ユーザー入力フックによって **user_number field** が必須フィールドとして設定されます。

アクション フックの操作

アクション フックを使用すると、レコード値の変更権限を持つユーザーを管理し、ユーザーの入力値が ClearQuest によってデータベースにコミットされる前に入力内容を確認できます。また、レコード全体の確認とアクション完了時の電子メールによる通知の送信も実行できます。

アクション フックの適用範囲は、現在のレコードです。ClearQuest が提供するアクション フックのタイプは次のとおりです。次の表に、事前定義アクション フックを実行される順序で示します。

アクション フック	用途	実行されるタイミング
アクセス制御	<p>指定したユーザーが、あるレコードに対して指定したアクションを開始できるかどうかを示すブール値を返します。このフックは、ユーザーがアクションを実行する前に呼び出されます。</p> <p>アクセス制御フックは、VBScript 形式または Perl 形式のサブルーチンとして記述できます。</p> <p>メモ: プライマリ アクション (Modify、Submit、Delete、Import) を実行するには、すべてのベースアクションのアクセス制御リストだけでなく、プライマリ アクションのアクセス制御リストにも現在のユーザーが登録されている必要があります。</p> <p>182 ページの「アクションへのユーザーのアクセス制限」を参照してください。</p>	アクションが開始される直前
初期化	<p>初期フィールド値を設定します (または管理者が指定する任意のタスクを実行します)。</p> <p>レコードに対し、複雑な初期設定を行うことが可能です。このフックを使用すると、ClearQuest でアクションが開始される前に、フィールド値を設定できます。このフックは、アクションが初期化された後で、レコードの内容がフォーム内に表示される前に呼び出されます。</p> <p>初期化フックは、スクリプト サブルーチンとして記述する必要があります。</p>	アクションが開始される時点

アクション フック	用途	実行されるタイミング
確認	<p>管理者が指定するフィールド値を確認します。ユーザーが無効なデータを入力した場合は、ClearQuest からユーザーに対して、有効なデータの入力を求めるメッセージが表示されます。</p> <p>このフックを使用すると、個別フィールドの確認フックの内部では確認が困難な条件をチェックできます。たとえば、このフックを使用すると、フィールドのグループ全体の情報を確認できます。このフックは、ClearQuest によって、データベースに変更がコミットされる前に実行されます。</p> <p>確認フックは、スクリプトを使用する必要があります。</p> <p>212 ページの「ユーザーが入力したフィールド値の確認」を参照してください。</p>	ユーザーがアクションをコミットする時点
コミット	<p>複数のレコードに対する 1 つのアクションを、1 つのトランザクションにリンクします (たとえば、元の変更依頼が解決した場合には、その依頼に対するすべての複製も解決します)。</p> <p>外部データ ソースとデータベースの内容が並行関係を維持するように、外部データ ソースのセットを更新します。このフックは、変更がデータベースに追加された後で、その変更がコミットされる前に呼び出されます。</p> <p>コミット フックは、VBScript 形式または Perl 形式のサブルーチンとして記述できます。</p>	ClearQuest によってトランザクションがデータベースにコミットされる直前
通知	<p>アクションが実行されたことをユーザーに通知するコミットの後続アクションを開始します。「第 12 章 ClearQuest 電子メールの管理」を参照してください。</p> <p>通知フックは、スクリプトを使用する必要があります。</p>	ClearQuest によってトランザクションがコミットされた後

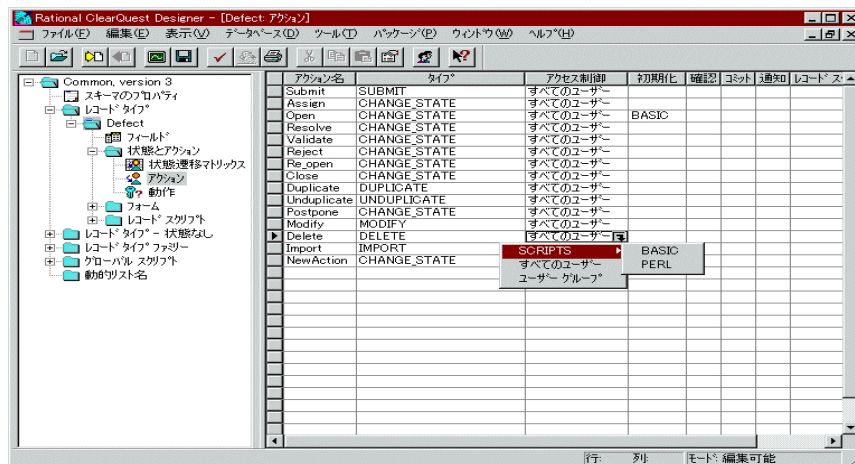
関連項目

- 218 ページの「フィールド フックとアクション フックの実行順序」
- 335 ページの「アクション フックの例」

アクション フックの追加

アクション フックを定義するには、アクション グリッドを使用します。

- 1 ワークスペースで、[レコード タイプ] フォルダを展開し、目的のレコード タイプを展開します。
- 2 [状態とアクション] を展開し、[アクション] をダブルクリックして、アクション グリッドを表示します。



- 3 追加対象のフックに対応する列をクリックし、ドロップダウン リストから [SCRIPTS] メニューの [BASIC] または [PERL] をクリックします。

アクセス制御フックを追加する場合は、[すべてのユーザー] を選択してユーザー全員にアクセス権を付与するか、[ユーザー グループ] を選択して特定のユーザー グループに属するユーザーにアクセス権を付与することもできます。[ユーザー グループ] を選択する場合は、スキーマが使用するすべてのデータベースにグループが登録されていることを確認してください。

[インスタント編集モード] が有効な場合は、ClearQuest Designer によってスクリプト エディタが自動的に開きます。[インスタント編集モード] が無効な場合は、セルをダブルクリックしてスクリプト エディタを開きます。

- 4 フックを編集します。ClearQuest によって、そのフックの呼び出し構文がスクリプト エディタに自動的に追加されます。呼び出し構文は灰色表示され、編集はできません。
- 5 作業が終了したら、スクリプト エディタを閉じます。

アクション フックの編集

アクション フックを編集するには、アクション グリッドを使用します。

- 1 ワークスペースで、[レコード タイプ] フォルダを展開し、目的のレコード タイプを展開します。
- 2 [状態とアクション] を展開し、[アクション] をダブルクリックして、アクション グリッドを表示します。
- 3 編集対象のフックが設定されているセルをダブルクリックします。ClearQuest Designer によってスクリプト エディタが開きます。

メモ: ClearQuest Designer には、VBScript 用と Perl 用のスクリプト エディタが別々に用意されています。ClearQuest ウィンドウのタイトル バーには、エディタ タイプが表示されます。適切なエディタを使用していることを確認してから、コードを記述してください。VBScript と Perl の両方で記述されたスクリプトがあるセルをダブルクリックすると、両方のスクリプト エディタが重なって開きます。

- 4 フックを編集します。ClearQuest によって、そのフックの呼び出し構文がスクリプト エディタに自動的に追加されます。呼び出し構文は灰色表示され、編集はできません。
- 5 作業が終了したら、スクリプト エディタを閉じます。

アクション フックの削除

メモ: フックを削除した後は、フックに加えた変更は、スキーマの以前のバージョンに格納されていない限り復元できません。

アクション フックを削除するには

- 1 ワークスペースで、[レコード タイプ] フォルダを展開し、目的のレコード タイプを展開します。
- 2 [状態とアクション] を展開し、[アクション] をダブルクリックして、アクション グリッドを表示します。
- 3 削除対象のフックが設定されているセルをクリックし、ドロップダウン リストの[なし] をクリックします。アクセス制御フックの場合は、[すべてのユーザー] または[ユーザーグループ] のいずれかをクリックします。
- 4 削除するかどうかを確認するメッセージが表示されたら、[はい] をクリックします。

フィールド フックとアクション フックの実行順序

ClearQuest は、アクション フックとフィールド フックを、事前定義された時点に事前定義された順序で実行します。レコードが開いているときには、4 つのフック実行ポイントがあります。

- アクション開始時
- フィールド値の設定時
- レコードの確認時
- レコードのコミット時

アクション開始時

ユーザーがアクションを開始すると、ClearQuest は次の順序でフックを実行します。

- 1 アクション アクセス制御フックが、ユーザー アクセスをチェックし、ユーザーにアクションを開始する権限がない場合はアクションの処理を停止します。この場合には、後続のフックは実行されません。
- 2 フィールド権限フックが、各フィールドが必須、オプション、読み取り専用のいずれであるかを判定します。
- 3 アクション初期化フックが、ClearQuest によってアクションが開始される前に、フィールド値の設定などの初期化タスクを実行します。
- 4 フィールド デフォルト値フックが、各フィールドの値を設定します (Submit アクションの場合のみ)。
- 5 フィールド確認フックが、各フィールドの内容を確認します。ただし、確認が発生しないインポート アクションは除きます。フィールド確認フックは、値を設定していない必須フィールドなど、存在しない値を確認することはできません。
- 6 フィールド選択リスト フックが、[選択リストを再計算] オプションを使用する各フィールドに対して実行されます。依存フィールドが存在し、そのフィールドの作成時に [選択リストを再計算] オプションをオンにした場合、ClearQuest により、各フィールドに対してフィールド確認フックとフィールド選択リスト フックが順次実行されます。このフック実行は、変更したすべてのフィールドの設定と確認が終了するまで続きます。

フィールド値変更フックが、エンティティ オブジェクトの `SetFieldValue` メソッドを呼び出す場合、`SetFieldValue` 関数が呼び出されるたびにそのフィールドに対する `VALUE_CHANGED` フックが実行されます。

フィールド値の設定時

ユーザーがレコードを編集すると、ClearQuest は次の順序でフックを実行します。

- 1 フィールド値変更フックが、変更される各フィールドに対して実行されます。

[リストに制限] オプションがオンの場合、ユーザーが不正な値を入力すると、ClearQuest によってそのフィールドが無効なフィールドとしてマークされます。ユーザーがリーガル値を入力するまで次のフックは実行されません。

メモ: 別のフィールドの値を変更するためにフィールド値変更フックで SetFieldValue を呼び出す場合は、ClearQuest によってそのフィールドのフィールド値変更フックが即時実行されます。

- 2 フィールド確認フックが、各フィールドに対して実行されます。
- 3 フィールド選択リスト フックが、[選択リストを再計算] オプションが有効な各フィールドに対して実行されます。

[選択リストを再計算] オプションをオンにして依存フィールドを使用する場合には、ClearQuest により、各フィールドに対してフィールド確認フックとフィールド選択リストフックが順次実行されます。このフック実行は、変更されたすべてのフィールドの設定と確認が終了するまで続きます。

メモ: ClearQuest Web では、フィールド フックが実行されるのは、ユーザーが Submit アクションを呼び出したときだけです。ただし、フィールドが、依存フィールドを含むものとしてマークされている場合は除きます。243 ページの「フックを使用した Web セッションの検出」を参照してください。

レコードの確認時

データベースに変更をコミットする前に、ClearQuest では次の順序でフックを実行してレコードを確認します。

- 1 フィールド確認フックが、レコード内の各フィールドに対して実行されます。
- 2 アクション確認フックが、実行されたアクションに対して実行されます。

レコードのコミット時

コミット フックは、データベースで現在のレコードに対する変更が更新された後で、更新トランザクションがデータベースにコミットされる前に実行されます。したがって、コミット フックを使用して現在のレコードを変更することはできません。そのような変更はレコードに適用されません。

コミット フックは、主要なアクションと同じデータベース トランザクションの一部として組み込む (ほかのレコードへの) アクションに対してだけ使用します。たとえば、親レコードの障害が解決したときに重複レコードの障害も解決する場合です。

ClearQuest はレコードの確認後、次の順序でフックを実行してレコードをデータベースにコミットします。

- 1 アクション コミット フックが実行されます。
- 2 ClearQuest がトランザクションをデータベースにコミットします。
- 3 アクション通知フックが実行されます。たとえば、管理者が設定した電子メール ルールに基づいて、ClearQuest からさまざまなユーザーに電子メール メッセージが送信されます。

レコード スクリプトの操作

レコードのコンテキストでカスタム動作を実行するスクリプトを作成できます。レコード スクリプト サブルーチンは、1 つのレコード タイプ専用です。たとえば、レコード スクリプトを使用して、現在のレコードに関するデータを別のシステムに送信できます。

レコード スクリプトを呼び出す方法は 3 つあります。

- レコード スクリプトをフォーム コントロール (ボタンまたは右クリック ショートカットメニュー) と関連付けます。

たとえば、TeamTest スキーマの **Build_Properties** レコード スクリプトを使用すると、ユーザーは自分が選択したビルドのプロパティを表示できます。このスクリプトはボタンと関連付けられていて、このボタンをクリックするとスクリプトが実行されます。

Build_Properties レコード スクリプトは、ClearQuest リポジトリからビルド情報を取得し、ダイアログ ボックスに情報を表示します。

- アクションを使用して、レコード スクリプトを呼び出します。**Record_script_alias** アクションを作成し、このアクションをレコード スクリプトと関連付けます。ユーザーがこのアクションを選択すると、レコード スクリプトが即時に呼び出されます。このようなスクリプトの内部から現在のレコードを更新するには、セッション オブジェクトの **ClearQuest API EditEntity** メソッドを使用してスクリプト内でアクションを開始します。
- エンティティ オブジェクトの **ClearQuest FireNamedHook** メソッドを使用して、別のスクリプトまたはフックの内部からレコード スクリプトを呼び出します。

詳細については、オンラインの『**Rational ClearQuest API Reference**』を参照してください。

レコード スクリプトの理解

レコード スクリプトは、ClearQuest フォーム上のイベントに応じて、または別のフックから呼び出される、フックの一般フォームです。通常、レコード スクリプトを使用するのは、フォーム上の特定のフィールドに関連付けられたプッシュ ボタンまたはコンテキスト メニュー項目でのクリック イベントに応じて実行するアクションを実装する場合です。

レコード スクリプトは、現在選択しているレコードのコンテキストで実行します。

すべてのレコード スクリプトは、同じ構文を使用します。

```
Function RecordName_HookName(param)

    ' param As Variant

    ' RecordName_HookName As Variant

    ...

End Function
```

レコード スクリプトを別のフックから呼び出す場合、フックに渡すパラメータは適切なデータを含む単純な **Variant** 型です。フックが呼び出し側のフックに情報を返す場合、情報を **Variant** 型で返してください。

レコード スクリプトをフォーム コントロールに関連付ける場合、メソッドに渡されるパラメータには、**EventObject** クラスのインスタンスが含まれます。このインスタンスには、フックを呼び出すイベント情報が含まれます (222 ページの「フォーム コントロール イベント」を参照)。**ClearQuest** は、レコード スクリプトがフォーム コントロールに関連付けられている場合、レコード スクリプトからの戻り値を要求しません。レコード フックから値が返された場合、**ClearQuest** クライアントがエラーとして解釈し、その値をメッセージ ボックスに表示することがあります。

レコード スクリプトは、プッシュ ボタン、テキスト フィールド、リストに関連付けることができます。ボタンに関連付けると、ボタンをクリックしたときに、自動的にフックが実行されます。テキスト フィールドやリスト関連コントロールに関連付けると、**ClearQuest** によってフックがコントロールのショートカット メニューに追加されます。それによって、ユーザーはショートカット メニューからフックを選択し、実行することができるようになります。

また、レコード スクリプトを、**RECORD_SCRIPT_ALIAS** タイプのアクションに関連付けることもできます。これによって、**ClearQuest** フォームの [アクション] メニューにカスタム ボタンを設定できます。

関連項目

- 201 ページの「スクリプトの作成」
- 202 ページの「スクリプトを使用するための動作状況」
- 350 ページの「レコード スクリプトの例」

ClearQuest Web でのレコード スクリプトの使用法

- **ClearQuest Web** は、プッシュ ボタンによって起動されるレコード スクリプトの戻り値を調べます。戻り値が文字列である場合は、エラー メッセージと見なされ、フックはエラーになります。
- レコード スクリプト関数の戻り値を明示的に設定しないと、フックが正常に実行されたことを **ClearQuest Web** に示す際、NULL または空の値が返されます。

- **Web** に対してレコードフックを有効にするには、コントロールのプロパティ シートにある [拡張] タブで [WEB で有効] をオンにします。
- **Web** では、レコードフックは ClearQuest Web サーバーで実行されます。したがって、ウィンドウを表示する Windows ルーチンを呼び出さないでください。これは、レコードフックがサーバー上で実行され、Web クライアントではなくサーバー上で表示を行うためです。ClearQuest Web サーバーは、サーバー セッションの起動時に CQ_WEB という Session.NameValue を設定して、スクリプトを Web で使用できるようにします。

フォーム コントロール イベント

レコード スクリプトがフォーム コントロールによって起動される場合、ClearQuest はレコード スクリプトにパラメータとして EventObject オブジェクトを渡します。このオブジェクトには、発生したイベント タイプ情報が含まれます。各種コントロールは、ボタン クリック、項目の選択などを含む、さまざまなイベント タイプを生成できます。EventObject オブジェクト内の情報を使用して、イベントの処理方法を決定してください。

ClearQuest は、フォーム コントロールに対して以下のイベント タイプを生成します。

- ボタン クリック: ユーザーがプッシュ ボタン コントロールをクリックしたことを示します。
- コンテキスト メニュー項目の選択: ユーザーがコンテキスト メニューからフックを呼び出したことを示します。

以下の表に、各コントロールに対してサポートされているイベント タイプ、EventObject によって提供される追加情報を示します。サポートされているイベント タイプ列に示されている定数は、EventType 列挙型の一部です。

コントロール タイプ	サポートされているイベント タイプ	詳細情報
プッシュ ボタン	AD_BUTTON_CLICK	ボタン名
コンボ ボックス	AD_CONTEXMENU_ITEM_SELECTION	Null 文字列
ドロップダウン リスト ボックス	AD_CONTEXMENU_ITEM_SELECTION	Null 文字列
リスト ボックス	AD_CONTEXMENU_ITEM_SELECTION	表示されるフィールド値を選択
リスト表示	AD_CONTEXMENU_ITEM_SELECTION	表示されるフィールド値を選択
テキスト ボックス	AD_CONTEXMENU_ITEM_SELECTION	表示されるフィールド値を選択
ドロップダウン コンボ ボックス	AD_CONTEXMENU_ITEM_SELECTION	Null 文字列

レコード タイプへのレコード スクリプトの追加

レコード タイプの [レコード スクリプト] フォルダまたは動作グリッドを介して、レコード スクリプトをレコード タイプに追加します。アクション グリッドを介してレコード スクリプトを追加する場合は、128 ページの「新規アクションの追加」を参照してください。

[レコード スクリプト] フォルダを介してレコード スクリプトを追加するには

- 1 ワークスペースで、[レコード タイプ] フォルダを展開し、目的のレコード タイプを展開します。
- 2 [レコード スクリプト] フォルダを展開します。
- 3 目的の言語のフォルダを右クリックし、ショートカット メニューの [追加] をクリックします。ClearQuest Designer によって、スクリプトがフォルダに追加され、名前を変更できるように強調表示されます。
- 4 レコード スクリプトに対して新しい名前を入力します。
- 5 新しいレコード スクリプト名をダブルクリックして、スクリプト エディタを開きます。
- 6 レコード スクリプトを編集します。以下のコメント行の場所にコードを入力します。
`REM added your hook code here`
- 7 コードの編集が終了したら、[フック] メニューの [コンパイル] をクリックして構文エラーがないかを確認します。コンパイルすると、ClearQuest によってフック コードが自動的に保存されます。
- 8 スクリプト エディタ ウィンドウを閉じます。

定義するレコード スクリプトごとに、スクリプト エディタはレコード名、フック名、必要な構文を含めたスタブを作成します。各フックは、以下の書式になります。

```
Function RecordName_HookName (param)
    ' param as EventObject
    REM Your code here
End Function
```

レコード スクリプトに渡されるパラメータは、`EventObject` クラスのインスタンスです。このインスタンスには、フックへの呼び出しをトリガしたイベント情報が含まれます。このインスタンスには、イベント タイプ、イベントを開始するコントロール、イベント タイプに応じた追加情報が含まれます。詳細については、222 ページの「フォーム コントロール イベント」を参照してください。

レコード スクリプトの編集

レコード スクリプトを編集するには

- 1 ワークスペースで、[レコード タイプ] フォルダを展開し、目的のレコード タイプを展開します。
- 2 [レコード スクリプト] フォルダ以下を展開して、修正対象のレコード スクリプトを見つけます。
- 3 編集するレコード スクリプトをダブルクリックします (または、フックを右クリックし、ショートカット メニューの [開く] をクリックします)。
- 4 レコード スクリプトを必要に応じて編集します。このフックが新規レコード スクリプトの場合、次に示すコメント行の場所にコードを入力します。

```
REM added your hook code here
```

- 5 コードの編集が終了したら、[フック] メニューの [コンパイル] をクリックして構文エラーがないかを確認します。コンパイルすると、ClearQuest によってフック コードが自動的に保存されます。
- 6 スクリプト エディタ ウィンドウを閉じます。

レコード スクリプトの削除

レコード スクリプトを削除するには

- 1 ワークスペースで、[レコード タイプ] フォルダを展開し、目的のレコード タイプを展開します。
- 2 [レコード スクリプト] フォルダ以下を展開して、削除対象のレコード スクリプトを見つけます。
- 3 削除するレコード スクリプトを右クリックし、ショートカット メニューの [削除] をクリックします。
- 4 スクリプトを削除するかどうかを確認するメッセージが表示されたら、[はい] をクリックします。

メモ: 削除対象のレコード スクリプトがアクションで使用される場合は、警告が表示されます。アクションに関連付けられている場合は、削除処理が失敗します。

グローバル スクリプトの操作

グローバル スクリプトを使用すると、スキーマ内の任意のフックからの呼び出しが可能な共通関数を定義できます。グローバル スクリプトは一種のサブルーチン ライブラリです。ClearQuest がグローバル スクリプトを直接呼び出すことはありません。

グローバル スクリプトは、同じフック コードを呼び出す複数のレコード タイプがある場合に役立ちます。グローバル スクリプトを使用すると、フック コードを集中管理できるため、フック コードをさまざまな位置にコピーする必要がなくなります。たとえば、ClearQuest の Email パッケージにあるグローバル スクリプトを使用すると、1 つのグローバル スクリプトを呼び出すだけで複数の通知送信アクションを実行できます。

グローバル スクリプトの理解

ClearQuest では、グローバル スクリプトを使用できます。グローバル スクリプトは、スキーマ全体にグローバルに使用できる関数です。グローバル スクリプトは、レコード スクリプトとグローバル スクリプトを含む、ほかのあらゆるタイプのフックから呼び出すことができます。特定のレコードに関連付けられるフックとは異なり、グローバル スクリプトには、事前定義された構文はありません。フックを関数にするか、サブルーチンにするかを制御し、さらにフックが受け付けて返すパラメータの数と種類を制御します。

グローバル スクリプトを呼び出す唯一の方法は、別のスクリプトから呼び出すことです。ただし、最終的にグローバル スクリプトへの呼び出しは、フィールド フック、アクション フックまたはレコード スクリプトなどの、レコードベースのフックへの呼び出しにトレース バックする必要があります。グローバル スクリプトへの呼び出しは、常にレコードベースのフックまで追跡されるため、各グローバル スクリプトは、暗黙のエンティティ オブジェクトと共に提供されます。このエンティティ オブジェクトは、呼び出しが発生したレコードベースのフックに渡されます。

メモ:

- VBScript と Perl のいずれを使用してもスクリプトを作成できますが、一方の言語タイプからもう一方の言語タイプを呼び出すことはできません。グローバル スクリプトを記述する場合は、それを呼び出すスクリプトと同じ言語で記述してください。
- グローバル フックでは関数のみを使用してください。なぜなら、グローバル スクリプトのコードは、関数内のコード以外すべて実行されるためです。
- 201 ページの「スクリプトの作成」と 202 ページの「スクリプトを使用するための動作状況」を参照してください。

グローバル スクリプトの作成

グローバル スクリプトを作成するには

- 1 ワークスペースで、[グローバル スクリプト] フォルダを展開します。
- 2 目的の言語のフォルダ ([Basic] または [Perl]) を右クリックし、ショートカット メニューの [追加] をクリックします。
- 3 新規のグローバル スクリプトの名前を入力します。
- 4 新しい名前をダブルクリックして、スクリプト エディタを開きます。
- 5 グローバル スクリプトを必要に応じて編集します。次に示すコメント行の場所にコードを入力します。

```
REM TODO -- put your script code here
```

新規のグローバル スクリプトを作成すると、ClearQuest によってフックのブレースホルダが作成されますが、コードは作成されません。Function または Sub 宣言とパラメータリストを含む関数またはサブルーチンのためのすべてのコードを記述してください。グローバル スクリプトに付ける名前は、スキーマ内で一意である必要があります。

- 6 グローバル スクリプトの編集が終了したら、[フック] メニューの [コンパイル] をクリックして構文エラーがないかを確認します。コンパイルすると、ClearQuest によってコードが自動的に保存されます。
- 7 作業が終了したら、スクリプト エディタ ウィンドウを閉じます。

グローバル スクリプトの編集

グローバル スクリプトを編集するには

- 1 ワークスペースで、[グローバル スクリプト] フォルダを展開し、目的の言語のフォルダ ([Basic] または [Perl]) を展開します。
- 2 編集するスクリプトをダブルクリックします
- 3 グローバル スクリプトを必要に応じて編集します。このスクリプトが新規グローバル スクリプトの場合、次に示すコメント行の場所にコードを入力します。

```
REM TODO -- put your script code here
```

- 4 編集が終了したら、[フック] メニューの [コンパイル] をクリックして構文エラーがないかを確認します。コンパイルすると、ClearQuest によってコードが自動的に保存されます。
- 5 スクリプト エディタ ウィンドウを閉じます。

グローバル スクリプトの削除

グローバル スクリプトを削除するには

- 1 ワークスペースで、[グローバル スクリプト] フォルダを展開し、目的の言語のフォルダ ([Basic] または [Perl]) を展開します。
- 2 削除するスクリプトを右クリックし、ショートカット メニューの [削除] をクリックします。
- 3 スクリプトを削除するかどうかを確認するメッセージが表示されたら、[はい] をクリックします。

メモ: 削除対象のスクリプトがフック コードで使用される場合でも、警告は表示されません。スクリプトを削除しても、フックからの呼び出しを削除しないと、削除したスクリプトを呼び出すアクションまたは変更をユーザーが実行したときに、エラーが発生する原因になります。

外部アプリケーションの作成

ClearQuest データベースに対してタスクを実行する外部アプリケーションを、VBScript または Perl で作成できます。外部アプリケーションを使用すると、クエリーの作成、保存されているクエリーの実行、新規レコードの作成、既存のレコードに対するアクションの実行が可能です。たとえば、外部アプリケーションは次のタスクを実行できます。

- 毎日午前 0 時に、事前定義されたクエリーのセットを実行し、その結果をマネージャーのグループに電子メールで送信します。
- 毎日曜日に、1 か月以上オープンのままになっている障害をすべて検出し、その障害のステータスを高く設定し直します。

外部アプリケーションを開始するには、セッション オブジェクトを作成し、ClearQuest データベースにログインする必要があります。228 ページの「セッションの操作」を参照してください。

メモ: ClearQuest には、Perl のバージョンの 1 つ (CQPerl.exe、CQPerl.dll) が含まれます。Perl を使用して外部アプリケーションを作成する場合には、CQPerlExt パッケージを使用してください。オンラインの『Rational ClearQuest API Reference』の「Using the ClearQuest API」を参照してください。

ClearQuest API の使用法

ClearQuest API を使用すると、ClearQuest の事前定義されたフックのカスタマイズ、独自のフックの作成、ClearQuest データベースに対してタスクを実行する外部アプリケーションの作成を行うことができます。

フックか外部アプリケーション、またはその両方を作成するときに使用できるオブジェクト、メソッド、プロパティ、定数の詳細については、オンラインの『Rational ClearQuest API Reference』を参照してください。

セッションの操作

セッション オブジェクトは、現在のデータベース アクセス セッションを表す、すべての操作の起点です。フックを作成する場合、ClearQuest では、エンティティ オブジェクトの `GetSession` メソッドを使用して、現在のセッション オブジェクトにアクセスできます。フックはレコード (エンティティ) 変更のコンテキストで実行されるため、対応するエンティティ オブジェクトは常に存在し、このオブジェクトから `GetSession` を呼び出せます。

ClearQuest データベースにアクセスする外部アプリケーションを作成する場合、セッション オブジェクトを作成し、データベースにログインする必要があります。エンティティを操作するには、エンティティ オブジェクトを返す API を呼び出す必要があります。

詳細については、オンラインの『Rational ClearQuest API Reference』の「Working with sessions」を参照してください。

クエリーの操作

クエリーを実行すると、管理者が指定した検索基準のセットに基づいて、ClearQuest データベースからデータを取得できます。クエリーを作成するには、次の手順に従います。

- 取得するデータを指定する `QueryDef` オブジェクトを使用して、クエリーを作成します。
- データを収容する `ResultSet` オブジェクトを作成します。
- 結果セット内のデータを取得するクエリーを実行します。
- データにアクセスします。

`QueryDef` や `ResultSet` などのオブジェクトを使用してクエリーを作成する方法の詳細については、オンラインの『Rational ClearQuest API Reference』の「Working with Queries」を参照してください。

レコードの操作

1 人のユーザーが変更依頼を入力すると、そのデータは ClearQuest によってエンティティという論理レコードに保存されます。管理者は、レコードのデータを作成、編集、表示できます。また、レコードのエンティティ タイプに関するデータも表示できます。BuildEntity メソッドを使用すると、新規レコードを作成できます。EditEntity メソッドを使用すると、既存のレコードを編集できます。ClearQuest API には、変更内容を確認し、更新したレコードをデータベースにコミットするメソッドもあります。

詳細については、オンラインの『Rational ClearQuest API Reference』の「Working with Records」を参照してください。

一般的な API 呼び出し

ここでは、フックの作成に使用できる基本的な構成要素について説明します。それぞれの API 呼び出しを最初に VBScript 形式で、次に Perl 形式で示します。構文として <object.><method> の書式を使用します。

メモ: Perl では、現在のエンティティ オブジェクトとセッションオブジェクトを `entity` と `session` (小文字) として事前定義しています。VBScript では、現在のエンティティ オブジェクトが引き継がれるため、メソッドを呼び出すときにこのオブジェクトを明示的に指定する必要はありません。

詳細については、オンラインの『Rational ClearQuest API Reference』を参照してください。

API 呼び出し (VBScript/Perl)	関数
[entity.]GetSession \$entity->GetSession	ほかの多数の API を呼び出すために必要なセッションを取得します。
session.OutputDebugString \$session->OutputDebugString	フック コードまたは外部アプリケーションのデバッグ時に使用できるデバッグ ストリーム情報を出力します。
session.GetEntity \$session->GetEntity	データベースからレコードを取得します。
session.EditEntity \$session->EditEntity	データベースから ClearQuest が取得したレコードを編集します。
[entity.]SetFieldValue \$entity->SetFieldValue	値をフィールドに割り当てます。

API 呼び出し (VBScript/Perl)	関数
[entity.]Validate \$entity->Validate	ClearQuest がデータベースにレコードを保存する前に、レコード内のデータが有効かどうかを確認します。
[entity.]Commit \$entity->Commit	レコードとレコードに対するすべての編集内容をデータベースにコミットします。
[entity.]Revert \$entity->Revert	変更を取り消します。確認エラーが発生したためにコミットが実行されない場合には、このメソッドを使用すると便利です。
[entity.]GetFieldValue \$entity->GetFieldValue	指定したフィールドのフィールド情報オブジェクトを取得します。
FieldInfo.GetValue \$FieldInfo->GetValue	フィールドの値を取得します。
session.BuildQuery \$session->BuildQuery	クエリーを作成します。
QueryDef.BuildField \$QueryDef->BuildField	クエリーの検索結果に特定のフィールドを含めます。
QueryDef.BuildFilterOperator QueryFilterNode.BuildFilterOperator \$QueryDef->BuildFilterOperator \$QueryFilterNode->BuildFilterOperator	「等しい」または「より大きい」など、クエリーのフィルタ演算子を作成します。
QueryFilterNode.BuildFilter \$QueryFilterNode->BuildFilter	複雑なクエリーのサポートを可能にします。
session.BuildResultSet \$session->BuildResultSet	クエリーの実行に必要な ResultSet オブジェクトを作成します。
ResultSet.Execute \$ResultSet->Execute	現在の ResultSet オブジェクトを使用してクエリーを実行します。
ResultSet.MoveNext \$ResultSet->MoveNext	データ セット内の次のレコードにカーソルを移動します。
ResultSet.GetColumnValue \$ResultSet->GetColumnValue	現在の行で、指定した列の値を取得します。
session.GetUserLoginName \$session->GetUserLoginName	ユーザーのログイン ID を取得します。

API 呼び出し (VBScript/Perl)	関数
entity.Revert \$entity->Revert	エンティティ オブジェクトに対して行われたすべての変更を破棄します。 フックの内部から現在のアクションを中断する場合には、 Revert API を使用しないでください。この API は、フックまたはスクリプト内で明示的に開始されたアクションを元に戻す場合にだけ使用できます。現在のアクションを中断する必要がある場合は、スクリプト言語の例外メカニズムを使用して例外を発生させるか、アクション確認フックから「false」を返させます。

フック スクリプト内のテキストの検索

フック内で特定のスクリプト テキストを検索できます。特定のフックを指定しなくても、スキーマ内のすべてのフックが検索対象になります。

メモ: フォームを開いている場合は、そのフォームを閉じてから次の手順で操作してください。

フック スクリプト内のテキストを検索するには

- 1 テキストを検索する対象になるスキーマを開きます。
- 2 [編集] メニューの [フック内の検索] をクリックします。
- 3 [フック内の検索] ダイアログ ボックスで、次の手順で操作します。
 - [検索文字列] に、検索する対象になるテキスト文字列を入力します。
 - [大文字と小文字を区別する] をクリックして、入力したテキスト文字列と完全に一致する文字列を検索します (大文字小文字も区別します)。
 - [検索] をクリックします。

[確認] ペインに検索結果が表示されます。
- 4 [確認] ペインで、表示または編集する個別の項目をダブルクリックします。エディタ ウィンドウが開き、該当するスクリプトとテキストが表示されます。

Rational ClearQuest Web を使用すると、ユーザーは Web ブラウザから ClearQuest データにアクセスできます。この章の内容は、次のとおりです。

- 233 ページの「ClearQuest Web に関する考慮事項」
- 234 ページの「ClearQuest Web のカスタマイズ」
- 241 ページの「ClearQuest Web へのアクセス制限」
- 241 ページの「ClearQuest Web でのフックの使用法」

ClearQuest Web クライアントを設定する手順の詳細については、『Rational Software サーバー製品インストールガイド』を参照してください。

ClearQuest Web に関する考慮事項

ClearQuest 統合は Web クライアントに適用されません。したがって、統合パッケージによって Windows 上または UNIX 上の ClearQuest クライアントに追加されたフィールド、フォームレポート、スクリプト、その他の機能は、Web クライアントから使用することはできません。

次の事項を ClearQuest Web のユーザーに通知する必要があります。

- ユーザーは、グラフをドリル ダウン (クエリー) できません。
- ユーザーは、共用のグラフとレポートのみを使用できます (個人用のグラフとレポートは作成できません)。
- ユーザーは、HTML 形式でのみレポートを作成できます。
- ユーザーは、ClearQuest Web 上での移動に ClearQuest Web のボタンを使用する必要があります。ClearQuest Web 上での移動には、Web ブラウザの [戻る] ボタンと [進む] ボタンを使用しないようにします。

メモ: ClearQuest Designer で作成したフォームは、Windows クライアントと ClearQuest Web クライアントとでは表示方法が異なります。たとえば、Windows ではタブをクリックして表示しますが、ClearQuest Web ではリンクをクリックするかスクロールしてタブを表示します。しかし、どちらのクライアントを使用しても利用できる情報は同じです。フォームを設計するときは、必ず ClearQuest Web でテストしてください。

ClearQuest データ コード ページと ClearQuest Web

Rational ClearQuest の Web クライアント、UNIX クライアント、Windows クライアントでは、データの破壊と拡散に対して異なる保護レベルを提供します。Web クライアントでは Web ブラウザで使用するオペレーティング システム コード ページを判断できないため、ユーザーが Web インターフェイスを使用する場合の無効なコード ページ文字に対して制限された保護が設定されています。ネイティブ クライアントではなく Web クライアントを使用する場合、無効な文字がデータベースに入力されたり、無効な文字によってデータが破壊される可能性がないとは言いきれません。

したがって、ClearQuest Web インターフェイスを使用する場合は、ClearQuest ユーザー データベースの ClearQuest データ コード ページ値でサポートされている文字のみを入力してください。サポートされていない文字を入力すると、文字が拒否されたり化けることがあります。

ClearQuest Web のカスタマイズ

メモ: Web 設定を編集するには、スーパー ユーザー権限またはスキーマ デザイナ権限が必要です。

ClearQuest Web インターフェイスは、カスタマイズが可能です。たとえば、使用するフォントや、インターフェイスのカラー スキームを変更できます。また、パフォーマンスに関する設定も編集できます。

Web 設定には次のものがあります。

- 235 ページの「ClearQuest Web 設定」
- 237 ページの「[電子メールによる通知を有効にする] の設定」
- 238 ページの「[表示スキーム] の設定」

ClearQuest Web 設定を編集するには

- 1 ClearQuest クライアントにスーパー ユーザー権限またはスキーマ デザイナ権限でログオンしたら、[操作] をクリックし、[Web 設定の編集] をクリックします。
- 2 必要に応じて Web 設定を編集します。
- 3 終了したら、[適用] をクリックして変更を適用します。この場合は、[Web 設定の編集] ウィンドウが開いたままになります。または、[OK] をクリックして変更を保存します。この場合は、[Web 設定の編集] ウィンドウが閉じます。

ClearQuest Web 設定

ClearQuest Web 設定では、アクセス権限、ダイナミック HTML のメニューの切り替え、タイムアウト設定など、さまざまな設定を指定します。次の表は、各 Web 設定の説明です。

設定	説明
制限付きアクセス Web サイト	すべてのユーザーのアクセスを制限するには、このチェック ボックスをオンにします。
制限付きアクセス ユーザー名	制限されたアクセス権のみを付与するユーザーを指定します。ユーザー名はカンマで区切ります。スペースを入力した場合は、これらの名前は無視されます。ここで指定されたユーザーは、障害の登録と [制限付きアクセス クエリー名] で指定されたクエリーの実行しか行うことができません。
制限付きアクセス ユーザー グループ	制限されたアクセス権のみを付与するユーザー グループを指定します。ユーザー グループ名はカンマで区切ります。スペースを入力した場合は、これらの名前は無視されます。ここで指定されたユーザーは、障害の登録と [制限付きアクセス クエリー名] で指定されたクエリーの実行しか行うことができません。
制限付きアクセス クエリー名	<p>制限付きアクセス権を持つユーザーが実行できるクエリーを指定します。制限付きアクセス権を持つユーザーが実行できるクエリーは、ここに一覧表示されるクエリーのみです。このクエリーでは Submitter = CurrentUser フィルタを使用して、すべてのユーザーが自分の登録したレコードのみを参照できるようにするのが一般的ですが、クエリーの種類は問いません。</p> <p>ClearQuest は、次のような場合に自動的にアクセス制限されます。</p> <ol style="list-style-type: none"> 1) ClearQuest サイト、ClearQuest ユーザー、ClearQuest グループのいずれかが Web 設定で制限付きに指定されている。または、 2) ClearQuest がライセンスなしで実行されている。
制限付きの場合レコード検索を許可する	制限付きアクセス権を持つユーザーがデータベースを検索できないようにする場合は、FALSE に設定します。
詳細クエリー エディタの使用を許可する	ユーザーが詳細クエリーを作成できないようにする場合は、FALSE に設定します (228 ページの「クエリーの操作」を参照)。
DHTML メニューを使用する	メイン ツールバーのメニュー機能をオフにする場合は、FALSE に設定します。FALSE に設定した場合、メニュー オプションのリストが使用可能になります。Web ブラウザがダイナミック HTML をサポートしていない場合は、この設定を FALSE にする必要があります。
更新時に読み取り専用タブを表示する	修正する各レコードに表示される情報を減らしてスクロールを高速にする場合は、FALSE に設定します。読み取り専用のタブが表示されなくなります。

設定	説明
スプラッシュ イメージ	Web サイトのスプラッシュ イメージを含むファイルの名前を入力します。これは、ログイン後に表示される主な製品ページです。
Oracle 用のクライ アント側カーソル	Oracle データベースを使用していてレポートを生成しない場合は、TRUE に設定します。
CCIntegration セッ ションを呼び出し 間で維持する	ClearQuest 技術サポートからの指示がない限り、この設定は変更しないでください。
デフォルト フィール ド スクリプトの表示 モード	ユーザーがスクリプトを含むフォームを最初に開いたときに、すべてのユーザー向けのデフォルトのスクリプト モードを設定するように変更します。ユーザーは、各自のセッション用にこの設定を変更することができます。この設定の詳細については、236 ページの「フィールド スクリプトの実行」を参照してください。
ログインのロック スキーム	ClearQuest 技術サポートからの指示がない限り、この設定は変更しないでください。
ログイン タイム アウト	このログイン タイムアウト設定では、ログインを中止するまでグローバル ロックを待機する時間をミリ秒単位で指定します。デフォルト値は 15000 (15 秒) です。
デバッグ追跡レベル	Active Server Pages (ASP) 内の問題をデバッグする必要がある場合を除き、0 に設定します。

フィールド スクリプトの実行

ClearQuest では、フォーム上で ActiveX フィールド コントロールを使用して、ほかのデータベースからの情報を表示します。ActiveX コントロールは、Microsoft Script Control と、ClearQuest Server アプリケーションに格納されているユーザー定義の Active Server Pages (ASP) スクリプトを使用して ClearQuest に実装されます。Web 上で ActiveX コントロールを実装するためのスクリプトは、ClearQuest 管理者によって作成されます。

ActiveX コントロールを使用するフォームを開くと、フィールドのスクリプトによって、フィールドに関連付けられているデータに関する情報が生成、表示されます。生成、表示される情報はスクリプトによって異なります。たとえば、障害に関連付けられた添付ファイルがある場合、そのファイルを表示するようにスクリプトを作成することができます。

スクリプトの実行には時間がかかり、また、生成された情報の表示が不要な場合もあるため、TestExpert ではスクリプトを実行するかどうかを選択することができます。

この選択は、外部 ASP スクリプトによって生成されるフォーム上の最初のフィールドの最後で行います。このフィールドで選択したオプションは、フォーム上のすべての同様なフィールドに適用されます。次のような選択肢があります。

- 読み取り専用のフォームと変更可能なフォームで、スクリプトを含むフィールドを実行し、更新された情報を生成する場合は、[すべて]をクリックします。
- スクリプトを実行せず、更新された情報を生成する場合は、[読み取り専用]をクリックします。これはデフォルト設定です。
- 読み取り専用のフォームと変更可能なフォームでいずれもスクリプトを含むフィールドを実行しないようにする場合は、[いいえ]をクリックします。このオプションは、フォームを表示していて最新情報を生成する必要がない場合に便利です。

選択するオプションの横にあるボタンをクリックします。たとえば、[すべて]をクリックすると、最初のフィールドの最後に次のようなメッセージが表示されます。

次のフォーム上のこのフィールドを演算するよう選択しました: すべてのフォーム次のフォーム上のこのフィールドが演算できます: 読み取り専用 または いいえ 上に示すラジオ ボタンを選択することによって実行できます。

メモ: ClearQuest での ActiveX コントロールの使用法の詳細については、Rational 技術サポートまでお問い合わせください。

[電子メールによる通知を有効にする]の設定

次の表は、ClearQuest Web の [電子メールによる通知を有効にする] の各設定の説明です。

設定	説明
アクティブなアイテムの送信	「1」を入力すると電子メールによる通知が有効になります。 「0」を入力すると電子メールによる通知が無効になります。
電子メール送信方法	電子メールの転送タイプに従って SMTP または MAPI を指定します。
MAPI サーバー	MAPI を使用している場合は、メール サーバー名を入力します。
MAPI プロファイル	MAPI を使用している場合は、プロファイル エイリアス名を入力します。
SMTP ホスト	SMTP を使用している場合は、ホスト名を入力します。
実際の名前	Web サーバーのコンピュータ名を入力します。Web サーバー名を入力すると、電子メール メッセージの [差出人] フィールドにサーバー名が表示されます。サーバー名を電子メール メッセージに表示しない場合は、何も入力しないでください。
Reply-to	Web サーバーの一般返信先アドレスを入力します。このアドレスは自動的に電子メール メッセージの [Reply-to] フィールドに表示されます。空白にしておくと、アドレスは表示されません。

メモ: 自動電子メール通知を使用するには、スキーマに Email パッケージをインストールし、ClearQuest クライアントで電子メール メッセージの送信を有効にする必要があります。

[表示スキーム] の設定

[表示スキーム] の設定では、色、枠、フィールド幅、フォントなどの一般的な表示オプションを指定します。これらの設定は個別に変更できます。また、事前定義された表示スキームを選択して適用することもできます (239 ページの「事前定義された表示スキームの操作」を参照)。

次の表は、[表示スキーム] の各設定の説明です。

設定	説明
単一列のレイアウトを使用する	製品の従来のバージョンと同様の方法でレコードを表示する場合は、 TRUE に設定します。
Netscape 7 用の単一列	Netscape 7 を使用していて、製品の従来のバージョンと同様の方法でレコードを表示する場合は、 True に設定します。
表枠の幅	レコードテーブルの幅を入力します。
テキスト フィールドの幅	単一列レイアウトを使用する場合、テキスト フィールドの幅をピクセルで入力します。
テキスト フィールドの高さ	単一列レイアウトを使用する場合、テキスト フィールドの高さをピクセルで入力します。
垂直グリッドの公差	複数列のレイアウトを使用する場合、フォーム内の同じ行にある項目間の許容間隔を入力します。
Null フィールド文字列	NULL のフィールドに使用する文字列を入力します。
メニュー フレームの幅	メイン ページ上のワークスペースの幅を入力します。
表題フレームの高さ	メイン ページの上部に表示される Rational ツールバーの高さを入力します。
表見出しの背景色	設定表の下の色表から色を選択します。
表の背景色	設定表の下の色表から色を選択します。
ページの背景色	設定表の下の色表から色を選択します。
ページのテキスト色	設定表の下の色表から色を選択します。
ページのリンク色	設定表の下の色表から色を選択します。
ページの表示済みリンクの色	設定表の下の色表から色を選択します。
メニュー タイトルのフォント色	設定表の下の色表から色を選択します。

設定	説明
メニュー タイトルの フォント書体	メニューのタイトルに使用できるフォントのリストを入力します。
タイトル テキストの フォント色	設定表の下の色表から色を選択します。
タイトル テキストの フォント書体	フィールドのタイトルに使用できるフォントのリストを入力します。
ダイアログ見出しの 背景色	設定表の下の色表から色を選択します。
ダイアログ見出しの テキスト色	設定表の下の色表から色を選択します。
ダイアログの背景色	設定表の下の色表から色を選択します。
ダイアログの テキスト色	設定表の下の色表から色を選択します。

事前定義された表示スキームの操作

ClearQuest には、事前定義された表示スキーム (名前セット) がいくつかあります。[表示スキーム] の各設定の下部にあるボタンを使用すると、事前定義されたスキームを次の方法で変更できます。

- 240 ページの「表示スキームの編集」
- 240 ページの「ほかの表示スキームの適用」
- 240 ページの「表示スキームの削除」
- 240 ページの「表示スキームの名前変更」
- 240 ページの「新しい表示スキームの作成」

事前定義された表示スキームの使用法

事前定義された表示スキームの使用するには

- 1 リストで表示スキームをクリックし、[ロード] をクリックします。
- 2 [適用] をクリックします。

表示スキームの編集

表示スキームを編集するには

- 1 リストで表示スキームをクリックし、[ロード]をクリックします。
- 2 表示設定を必要に応じて編集します。

メモ: 色に関するオプションについては、表示するカラーパレットのタイプを選択します。
[ClearQuest カラーパレット] ウィンドウが表示されます。カラー番号をコピーし、表示スキームオプション表に貼り付けます。

- 3 [適用]をクリックします。

ほかの表示スキームの適用

ほかの表示スキームを適用するには

- 1 リストで表示スキームをクリックし、[ロード]をクリックします。
- 2 [適用]をクリックします。

表示スキームの削除

表示スキームを削除するには

- リストで表示スキームをクリックし、[削除]をクリックします。

表示スキームの名前変更

表示スキームの名前を変更するには

- 1 リストで表示スキームをクリックします。
- 2 テキストボックスに新しい名前を入力し、[名前の変更]をクリックします。

新しい表示スキームの作成

新しい表示スキームを作成するには

- 1 新規スキームのベースにする既存の表示スキームをリストでクリックします。
- 2 テキストボックスに新しい名前を入力し、[名前を付けて保存]をクリックします。

ClearQuest Web へのアクセス制限

ClearQuest Web へのアクセスを制限するには、ユーザーまたはユーザー グループのアクティビティを次の 2 つに制限します。

- レコードの登録
- 指定した単一のクエリーの実行

この ClearQuest Web の Web エントリ バージョンには、ユーザー権限の層がもう 1 つあります。たとえば、ベータ版の顧客が、パスワードで保護されているエクストラネットを通じてフィードバックできるようにすることが可能です。まず、フィードバックを目的とする顧客で構成される ClearQuest ユーザー グループを作成し (180 ページの「ユーザー グループの操作」を参照)、そのグループに ClearQuest の URL を配布します。このグループは、レコードの登録と、指定のクエリーの使用のみを行うことができます。クエリーの結果は、読み取り専用です。

ClearQuest Web が特定のユーザーまたはユーザー グループに対して Web エントリ クライアントになるように設定するには、ClearQuest Web の [操作] をクリックし、[Web 設定の編集] をクリックします。Web エントリを設定するには、スーパー ユーザー権限が必要です。

詳細については、ClearQuest Web のヘルプで「Web 設定の編集」を参照してください。

ClearQuest Web でのフックの使用法

スキーマに作成したフックは、ClearQuest Web と共に Web サーバーで実行されます。ClearQuest Web でフックを使用する際は、次の点に注意してください。

- ClearQuest Web に対して依存フィールドを有効にする必要があります。242 ページの「ClearQuest Web に対する依存フィールドの有効化」を参照してください。
- メッセージ ボックスは使用できません。
- コンテキストメニュー フックは、サポートされていません。
- フックを使用して、Web セッションを検出できます。243 ページの「フックを使用した Web セッションの検出」を参照してください。

ClearQuest Web に対する依存フィールドの有効化

依存フィールドを含むフォームを ClearQuest Web に表示する場合、次の手順を実行する必要があります。

- 1 フィールドをレコード フォームに追加するときに、親フィールドとその依存フィールドに次のフォーム コントロールのいずれかを使用します。複数のコントロールを組み合わせで使用することもできます。
 - ドロップダウン リスト ボックス
 - コンボ ボックス
 - ドロップダウン コンボ ボックス
- 151 ページの「フォーム コントロールの操作」を参照してください。
- 2 コントロールをフォームに追加したら、親フィールドのコントロールを右クリックし、ショートカット メニューの [プロパティ] をクリックします。
- 3 [Web 依存先フィールド] タブで、[使用可] リストで適切な子フィールドをクリックし、選択した子フィールドを [選択] リストに追加します。
- 4 フォーム上のコントロールを右クリックし、ショートカット メニューの [プロパティ] をクリックします。

コントロールのプロパティ シートにある [Web 依存先フィールド] タブで、各フィールドの値に依存するフィールドを指定します。依存関係にある親フィールドのみを Web に対して有効にする必要があります。



メモ: 依存フィールドに関連付けられた選択リストを更新するには、フィールドに選択リストを作成する際に、[選択リストを再計算] をクリックします。ClearQuest によって、選択リストがユーザーに表示される前に、リスト内容が再計算されます。この処理によって、Web のパフォーマンスが低下する場合があります。

ClearQuest Web でのメッセージの表示

ほかの Windows アプリケーションを呼び出す関数 (メッセージ ボックスなど) を実行すると、Web クライアントの動作が停止することがあります。たとえば、Web サーバーでメッセージ ボックス関数を実行すると、メッセージ ボックスがサーバーの画面に表示されます。ユーザーはサーバー上で [OK] をクリックできないので、クライアントが待機状態のままになります。この場合は、Web サーバーを再起動する必要があります。

レコード スクリプト フックが文字列値を返した場合は、その文字列がユーザーに表示されます。

フックを使用した Web セッションの検出

フックを作成する際に、ClearQuest API を使用して、ユーザーが固有のクライアントではなく、Web ブラウザにログインしているかどうかを検出できます。その結果によって、Web で使用できる機能に合うようにスキーマを調整していない場合には、適切な措置を取ることができます。たとえば、メッセージ ボックスまたは新規ウィンドウを作成する関数の中に Web セッションを検出した場合は、Web 環境用に変更したコードを呼び出すか、関数を終了することができます。

VBScript での Web セッションの検出

```
dim currDBSession          ' 現在の DB セッション

set currDBSession = GetSession

' Test for existence of the web session variable
if currDBSession.HasValue ("_CQ_WEB_SESSION") then
    ' Either exit or do something else
end if
```

Perl での Web セッションの検出

```
my $currDBSession;    # 現在の DB セッション

$currDBSession = $entity->GetSession();

# Test for existence of the web session variable
if ( $currDBSession->HasValue ("_CQ_WEB_SESSION") {
    # Either exit or do something else
}
```


ClearQuest 電子メールの 管理

12

ここでは、Rational ClearQuest の電子メール機能を有効にする手順について説明します。

この章の内容は、次のとおりです。

- 245 ページの「ClearQuest 電子メール機能の概要」
- 246 ページの「自動電子メール通知の有効化」
- 257 ページの「アクション フックを使用した電子メールの送信」
- 258 ページの「Rational E-Mail Reader を使用した電子メール登録の有効化」
- 265 ページの「登録のための電子メールの書式設定」。

ClearQuest 電子メール機能の概要

ClearQuest では、次の電子メール機能をサポートしています。

- **電子メールによる自動通知**: ClearQuest では、定義した条件に一致するレコードが発生したときに、ユーザーまたはユーザー グループに電子メールによる通知を自動的に送信できます。たとえば、障害が修正されたときに、テスト チームに電子メール メッセージを送信するように ClearQuest を設定できます。246 ページの「自動電子メール通知の有効化」を参照してください。
- **往復電子メール**: 往復電子メールを使用すると、電子メール通知と電子メールによる登録を組み合わせることでレコードを更新することができます。たとえば、新しいレコードが登録されたという自動電子メール通知を受け取ったときに、電子メール メッセージで返信して、データベース内のレコードにメモを追加することができます。256 ページの「往復電子メールの使用」を参照してください。
- **電子メール通知アクション フック**: ユーザーがアクションを完了したときに電子メール メッセージを作成して送信するように、フック コードを追加できます。257 ページの「アクション フックを使用した電子メールの送信」を参照してください。
- **電子メールによる障害の登録と変更**: ClearQuest ユーザーは、ClearQuest データベースにログインせずに、電子メールを使用してレコードの登録と変更を行うことができます。258 ページの「Rational E-Mail Reader を使用した電子メール登録の有効化」を参照してください。

自動電子メール通知の有効化

ClearQuest では、定義した条件に一致するレコードが発生したときに、ユーザーまたはユーザー グループに電子メールによる通知を自動的に送信できます。ClearQuest 自動電子メール通知を有効にするには、次の作業が必要です。

- 電子メール ルールの作成: ClearQuest 管理者が ClearQuest クライアントで電子メール ルールを作成します。
- ClearQuest クライアントでの電子メール通知の有効化: ClearQuest クライアント ユーザーが ClearQuest を設定して電子メール ルールを使用します。

電子メール ルールの作成

ClearQuest 管理者は、自動電子メール通知の条件を定義する電子メール ルールを作成します。電子メール ルールは、ClearQuest クライアントから **Email_Rule** レコードを登録することで作成します。電子メール ルールでは、次のような条件が発生したときに電子メール通知を送信することができます。

- レコードのフィールド値が変更された。
- レコード内で特定のアクションが実行された。
- レコードが特定のクエリー基準に一致する。

電子メール ルールを有効にするには、これらの条件の少なくとも 1 つを指定する必要があります。たとえば、電子メール ルールによって、障害が解決したときに品質保証チームに電子メール メッセージを送信するように設定することができます。電子メールのメッセージには、解決した障害に関するあらゆるフィールドを含めることができます。

電子メール ルールは状態なしレコード タイプとして、スキーマに組み込まれています。Common スキーマと Blank スキーマを除くほとんどの ClearQuest 定義済みスキーマは、**Email_Rule** レコード タイプに組み込まれています。対象のスキーマが **Email_Rule** レコード タイプに含まれていない場合は、Email パッケージをインストールすることで組み込むことができます。

スーパー ユーザーまたはスキーマ デザイナーの権限を持つユーザーだけが、電子メール ルールを作成または変更できます。詳細については、169 ページの「ClearQuest ユーザー権限」を参照してください。

メモ: 既存の電子メール ルールを持つスキーマのレコード タイプやアクションの名前を変更した場合は、電子メール ルールを更新して新しい名前を反映する必要があります。

電子メール ルールを作成するには

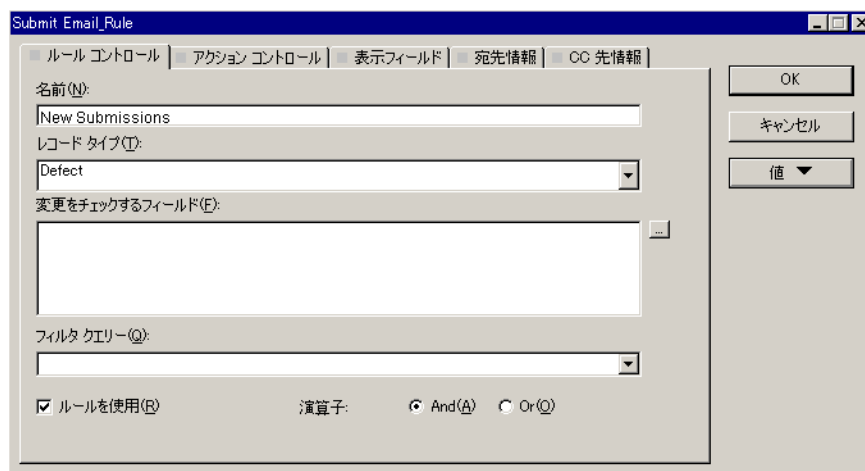
- 1 スキーマ デザイナ権限またはスーパー ユーザー権限を持つユーザーとして、ClearQuest クライアントにログオンします。
- 2 ClearQuest クライアントで、[アクション] メニューの [新規作成] をクリックして、[Email_Rule] レコード タイプを選択します。

[Submit Email_Rule] ダイアログ ボックスが表示されます。このダイアログ ボックスで、電子メール メッセージを送信するタイミング、電子メール メッセージを受け取るユーザーまたはユーザー グループ、電子メール メッセージの内容を決定するパラメータを設定します。

次の項に、電子メール ルールの設定例を示します。この例では、障害が登録されたときや障害が新しい所有者にアサインされたときに、特定のユーザーまたはユーザー グループに電子メール メッセージが送信されます。

電子メール ルールのルール コントロールの指定

[Submit Email_Rule] ダイアログ ボックスの [ルール コントロール] タブで、電子メール ルールの基準となるフィールドやクエリーを指定します。



- 1 [名前] に、「New Submissions」など電子メール ルール名を指定します。
- 2 [レコード タイプ] で、[Defect] など電子メール ルールの基準となるレコード タイプを選択します。
- 3 [変更をチェックするフィールド] で、フィールドを選択します。この場合は、選択したフィールドが変更されると、電子メール メッセージが自動的に送信されます。新しい障害が登録されたときに電子メール メッセージを送信する場合は、フィールドを選択する必要はありません。その他のタイプの電子メール ルールを設定する場合は、Headline、State、Ownerなどのフィールドを選択します。

- 4 [フィルタ クエリー] で、電子メール メッセージを送信するトリガとなる共用クエリーを選択できます。この場合は、クエリー基準に一致するレコードが変更された場合のみ、電子メール メッセージが送信されます。たとえば、ベータ版に関連付けられたすべてのレコードを検索するクエリーを選択すると、そのクエリーを含むレコードが変更されたときに電子メール メッセージが送信されます。

この電子メール ルール例では、[フィルタ クエリー] を選択する必要はありません。

- 5 [ルールを使用] がオンになっていることを確認します。電子メール ルールを一時的に無効にするには、[ルールを使用] をオフにします。
- 6 [演算子] で、[And] または [Or] を選択できます。演算子は、[ルール コントロール] タブにある次のフィールドで機能します。

- 変更をチェックするフィールド
- フィルタ クエリー

演算子は、[アクション コントロール] タブにある次のフィールドでも機能します (249 ページの「電子メールのアクション コントロールの指定」を参照)。

- アクション
- アクション タイプ
- 遷移元の状態
- 遷移先の状態

[And] を選択すると、このフィールドで指定したすべての条件に一致した場合のみ電子メール メッセージが送信されます。

[Or] を選択すると、このフィールドで指定したいずれかの条件に一致した場合に電子メール メッセージが送信されます。

電子メールのアクションコントロールの指定

[Submit Email_Rule] ダイアログ ボックスの [アクション コントロール] タブで、電子メール ルールの基準となるアクション、アクション タイプ、状態を指定します。

メモ: [アクション コントロール] タブで指定した内容には **And/Or** 演算子が適用されます。
248 ページの手順 6 を参照してください。

- 1 [アクション] で、電子メール ルールの基礎になるアクションを選択します。たとえば、[Assigned] を選択すると、アクションが割り当てられたときに電子メール メッセージが送信されます。
- 2 [アクション タイプ] で ClearQuest Designer の [アクション] グリッドに事前定義されている一般的なアクションの 1 つを選択して、電子メール ルールの基準にすることもできます。たとえば、[Submit] を選択すると、レコードが登録されたときに電子メール メッセージが送信されます。
- 3 [遷移元の状態] や [遷移先の状態] を電子メール ルールの基準にすることもできます。この例では、[遷移元の状態] と [遷移先の状態] は空白にします。

電子メール ルールの表示フィールドの指定

[Submit Email_Rule] ダイアログ ボックスの [表示フィールド] タブで、電子メール メッセージに含めるフィールドを指定します。

- 1 [送信者アドレス] で、「name@company.com」など電子メール メッセージの返信先アドレスを指定します。

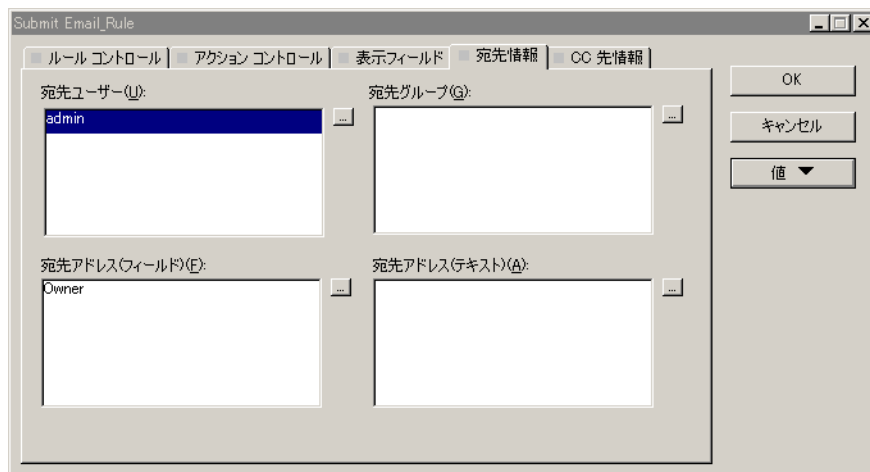
この例では、[送信者アドレス] は「Current User」のままでかまいません。

メモ: MAPI プロトコルを使用するときは、[送信者アドレス] は適用できません。

- 2 [件名フィールド] で、[Headline] など電子メール メッセージの件名に表示するフィールドを選択します。
- 3 [表示するフィールド] で、電子メール メッセージの本文に表示するフィールドを選択します。
 - 電子メール メッセージに以前のフィールド値を含める場合は、[直前の値を表示] をオンにします。
 - 電子メール メッセージにすべてのフィールドを含める場合は、[障害全体を含める] をオンにします。

電子メール ルールの電子メール アドレスの指定

電子メール メッセージを受信するユーザーとユーザー グループを指定するには、[Submit Email_Rule] ダイアログ ボックスの [宛先情報] タブを使用します。



- 1 [宛先ユーザー] で、[admin] など電子メール通知を受け取るユーザーを選択できます。
- 2 [宛先グループ] で、[Triage] など電子メールを受け取るユーザー グループを選択できます。
- 3 [宛先アドレス (フィールド)] で、フィールド値に基づいて電子メール アドレスを導き出すことができます。

たとえば、障害が新しい所有者にアサインされたときに電子メール メッセージを送信するには、[アクション コントロール] タブの [アクション] で **Assign** アクションを選択し、[宛先情報] タブの [宛先アドレス (フィールド)] で [Owner] フィールドを選択します。

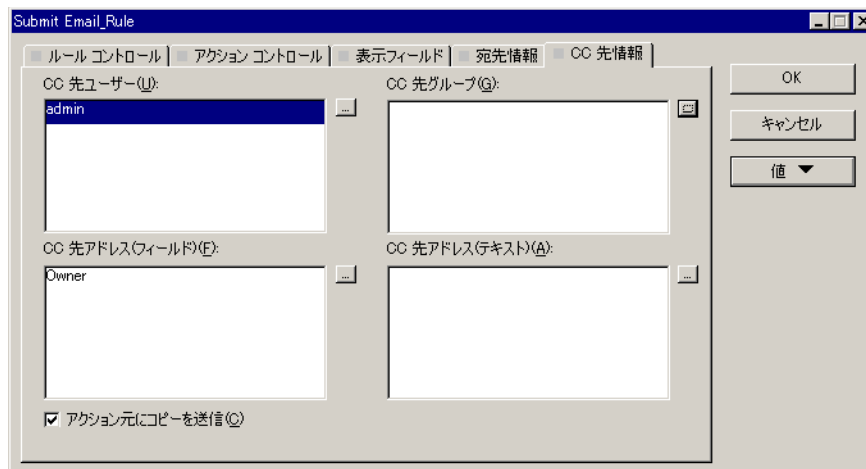
メモ: [宛先アドレス (フィールド)] に特定の電子メールアドレスを入力しない場合は、電子メール メッセージを受信するために、すべてのユーザーの電子メールアドレスがユーザー プロファイルの一部として定義されている必要があります。171 ページの「新規ユーザーの追加」を参照してください。

- 4 [宛先アドレス (テキスト)] で、特定の電子メール アドレスを入力できます。

電子メール ルールの同報アドレスの指定

[Submit Email_Rule] ダイアログ ボックスの [CC 先情報] タブで、電子メール メッセージのコピーを受け取るユーザーまたはユーザー グループを指定します。

メモ: 電子メール メッセージのコピーを受け取るには、すべてのユーザーとグループの各ユーザーの電子メール アドレスがユーザー プロファイル情報の一部として定義されている必要があります。171 ページの「新規ユーザーの追加」を参照してください。



- 1 [CC 先ユーザー] で、[admin] など電子メール通知のコピーを受け取る特定のユーザーを選択します。
- 2 [CC 先グループ] で、[Triage] など電子メール メッセージのコピーを受け取るユーザーグループを選択できます。
- 3 [CC 先アドレス (フィールド)] で、フィールド値に基づいて電子メール アドレスを指定できます。

メモ: [CC 先アドレス (フィールド)] に特定の電子メールアドレスを入力しない場合は、電子メールを受信するために、すべてのユーザーの電子メールアドレスがユーザー プロファイルの一部として定義されている必要があります。171 ページの「新規ユーザーの追加」を参照してください。

- 4 [CC 先アドレス (テキスト)] で、特定の電子メール アドレスを入力できます。
- 5 [アクション元にコピーを送信] を選択すると、電子メール メッセージのコピーが自分宛てに送信されます。
- 6 [OK] をクリックして、電子メール ルールを適用します。

電子メール ルールのサンプル

ClearQuest のユーザー データベースの評価版 (サンプル データベース) をインストールした場合は、いくつかの電子メール ルールのサンプルにアクセスできます。これらのサンプルはワークスペースの [共用クエリー] フォルダに保存されています。以下のものが含まれます。

- 新規登録
- 変更
- 解決
- クロージャ

既存の電子メール ルールの検索

クエリーを作成して、既存の ClearQuest システムにある電子メール ルールの検索と、電子メール ルールのアーカイブリストの作成を行うことができます。

既存の電子メール ルールを検索するクエリーを作成するには、ClearQuest クライアントで以下の操作を行ってください。

- 1 次の手順で [共用クエリー] フォルダ内に [電子メール ルール] という新規フォルダを作成します。[共用クエリー] フォルダを右クリックして表示されるメニューで [新規フォルダ] をクリックし、作成された新規フォルダに「電子メール ルール」という名前を付けます。
- 2 [クエリー] メニューの [新規クエリーの作成] をクリックします。
- 3 レコード タイプとして [Email_Rule] を選択し、[OK] をクリックします。
- 4 クエリーを選択するダイアログ ボックスで、クエリーを選択せずに [次へ] をクリックします。
- 5 [クエリーの表示方法の定義] ダイアログ ボックスで、[Name]、[record_type]、[Is_Active_Rule] などいくつかのフィールドを選択します。クエリーを実行するとこれらのフィールドが表示されるので、既存の電子メールルールと区別する役に立ちます。[次へ] をクリックします。
- 6 [クエリー フィルタとして使用するフィールドの選択] ダイアログ ボックスで、クエリーをフィルタせずに [次へ] をクリックします。
- 7 [クエリー フィルタの定義] ダイアログ ボックスで、[実行] をクリックします。既存の電子メール ルールが一覧表示されます。
- 8 [ファイル] メニューの [保存] をクリックし、新規クエリーに「すべての電子メール ルール」という名前を付けます。[OK] をクリックします。
- 9 新規クエリーの名前が [共用クエリー] フォルダに表示されます。新規クエリーを [電子メール ルール] フォルダにドラッグします。システム内のすべての電子メール ルールを検索するときは、[電子メール ルール] フォルダの [すべての電子メール ルール] クエリーをダブルクリックします。

既存の電子メール ルールの修正

既存の電子メール ルールを修正するには

- 1 既存のすべての電子メール ルールを検索するクエリーをまだ作成していない場合は、このクエリーを作成します。詳細については、253 ページの「既存の電子メール ルールの検索」を参照してください。
- 2 作成した [電子メール ルール] フォルダで、既存の電子メール ルールを一覧表示するクエリーをダブルクリックします。
- 3 既存の電子メール ルールの表で、修正するルールを選択します。
- 4 [アクション] メニューの [Modify] をクリックします。
- 5 [Email_Rule] フォームを使用して、[ルール コントロール]、[アクション コントロール]、[表示フィールド]、[宛先情報]、[CC 先情報] を変更します。
- 6 [OK] をクリックします。

電子メール アドレスの管理

ユーザーの電子メール アドレスを常に最新の状態にしておくことが重要です。たとえば、[宛先情報] や [CC 先情報] に、1 人以上のユーザーの電子メール アドレスが欠落または無効なグループがリストされていると、電子メール ルールを使用して必要な結果を得ることができません。

詳細については、180 ページの「ユーザー グループの操作」を参照してください。

非表示のユーザー

Rational ClearQuest E-Mail Reader は、ClearQuest ユーザー名が電子メール ルールで使用されている、非表示のユーザーには電子メール メッセージを送信しません。

この場合は、ClearQuest ユーザー名の代わりに、name@company.com など電子メール アドレスを明示的に指定します。

ClearQuest クライアントでの電子メール通知の有効化

自動電子メール通知を利用するには、Windows と UNIX の各 ClearQuest クライアントで電子メール通知を有効にする必要があります。電子メール通知を有効する方法については、ClearQuest クライアントのヘルプに記載されている指示を参照するようユーザーに伝えてください。ユーザーに、メール サーバーの SMTP ホスト アドレスや MAPI プロファイル名など、電子メール プロバイダのパラメータを伝える必要があることに注意してください。

メモ: Windows プラットフォームで MAPI 通知を使用するには、Collaboration Data Objects をインストールする必要があります。詳細については、『Rational Software サーバー製品インストールガイド』を参照してください。

ClearQuest Web での電子メール通知の有効化

ClearQuest Web で電子メール通知を有効にするには、スーパー ユーザー権限またはスキーマ デザイナ権限が必要です。

ClearQuest Web のインストール (初期インストールまたはアップグレードインストール) 後に、スキーマに設定されたビジネス ルールに従って電子メールによる通知を行なうように ClearQuest Web クライアントを設定できます。

- 1 スーパー ユーザー権限またはスキーマ デザイナ権限で ClearQuest Web クライアントにログインします。
- 2 [操作] メニューで [Web 設定の編集] をクリックします。
- 3 [電子メールによる通知を有効にする] までスクロールします。

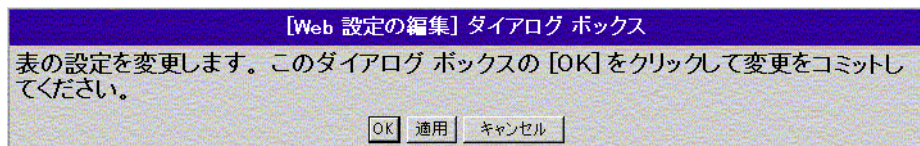
電子メールによる通知を有効にする	
送信を有効にする	<input type="checkbox"/> (0 または 1)
メールトランスポート	<input type="text" value="MAPI"/> (SMTP または MAPI)
MS Exchange Server	<input type="text" value="OE_TEST3"/> (トランスポートが MAPI の場合のみ)
MAPI プロファイルのエイリアス	<input type="text" value="qetest"/> (トランスポートが MAPI の場合のみ)
SMTP ホスト	<input type="text"/> (トランスポートが SMTP の場合のみ)
実名	<input type="text"/> (使用している Web サーバーのコンピュータ名)
返信アドレス	<input type="text"/> (書式: MyGenericReturnAddress@MyDomainName)

- 4 電子メールの設定を入力します。

設定	説明
アクティブなアイテムの送信	「1」を入力すると電子メールによる通知が有効になります。 「0」を入力すると電子メールによる通知が無効になります。
電子メール送信方法	電子メールの転送タイプに従って SMTP または MAPI を指定します。
MAPI サーバー	MAPI を使用している場合は、メール サーバー名を入力します。
MAPI プロファイル	MAPI を使用している場合は、プロファイル エイリアス名を入力します。
SMTP ホスト	SMTP を使用している場合は、ホスト名を入力します。

設定	説明
実際の名前	Web サーバー マシン名を入力します。Web サーバー名を入力すると、電子メール メッセージの [差出人] フィールドにサーバー名が表示されます。サーバー名を電子メール メッセージに表示しない場合は、何も入力しないでください。
Reply-to	Web サーバーの一般返信先アドレスを入力します。このアドレスは自動的に電子メール メッセージの [Reply-to] フィールドに表示されます。空白にしておくと、アドレスは表示されません。

- 5 Web ページの上部にある [OK] をクリックし、変更を保存します。



往復電子メールの使用

チーム内で電子メール通知と電子メール登録を組み合わせることでレコードを更新することができます。このような電子メールを往復電子メールと呼びます。

メモ: 往復電子メールを使用するには、Rational E-mail Reader を設定するときに、[Submit E-mail Rule] ダイアログ ボックスの [表示フィールド] タブにある [送信者アドレス] に同じ電子メール アドレスを指定する必要があります。246 ページの「電子メール ルールの作成」と 259 ページの「Rational E-mail Reader の設定」を参照してください。

往復電子メールの使用例を次に示します。

- 1 ユーザーが、決められた書式を使用して電子メールでレコードを登録します (265 ページの「登録のための電子メールの書式設定」を参照)。ClearQuest によって新しい障害 ID 番号 PROD0000137 が割り当てられます。
- 2 作成した電子メール ルールに従い、リード エンジニアにこの障害を通知する電子メールが自動的に送信されます (246 ページの「電子メール ルールの作成」を参照)。
- 3 「PROD0000137」という件名の電子メール メッセージを受け取ったリード エンジニアは、説明フィールドにコメントを記入して電子メールを返信します。
- 4 E-mail Reader によって電子メール メッセージが処理され、変更されたレコードが ClearQuest データベースに登録されます。

アクション フックを使用した電子メールの送信

アクション通知フックを使用すると、ユーザーが操作を完了したときに ClearQuest で自動的に電子メール メッセージを送信することができます。フック コードでは、ClearQuest API によりメッセージを作成して送信します。

ClearQuest の事前定義スキーマには、すべての有効なレコードに `Send_Email_Notif` ベースアクションが含まれています。このアクションによって `RSEM_ProcessEmailRules` が呼び出され、すべてのアクションに対してアクション タイプが使用されます。

メモ: スキーマのレコード タイプやアクションの名前を変更した場合は、アクション フックを更新して新しい名前を反映する必要があります。

アクション通知フックを作成するには

- 1 ClearQuest Designer に、スキーマ デザイナ権限またはスーパー ユーザー権限でログオンします。
- 2 ClearQuest Designer で、[レコードタイプ] または [レコードタイプ - 状態なし] を展開し、[アクション] フォルダを表示します。
- 3 [アクション] をダブルクリックして [アクション] グリッドを開きます。
- 4 [アクション] グリッドで、通知フックを追加するアクションの [通知] 列をクリックし、矢印アイコンをクリックして、[SCRIPTS] の [BASIC] または [SCRIPTS] の [PERL] をクリックします。

インスタント編集モードが有効になっている場合は、ClearQuest にスクリプト エディタが自動的に表示されます。インスタント編集モードが無効になっている場合は、セルをダブルクリックして、スクリプト エディタを表示します。

メモ: VBScript と Perl には、それぞれ専用のスクリプト エディタがあります。ClearQuest Designer ウィンドウのタイトル バーにエディタの種類が表示されます。正しいエディタを使用していることを確認してから、コードの追加や編集を行ってください。

- 5 フック コードを追加して、[フック] メニューの [コンパイル] をクリックしてコードをコンパイルし、構文エラーがないことを検証します。
- 6 スクリプト エディタ ウィンドウを閉じます。

詳細については、214 ページの「アクション フックの操作」と 344 ページの「アクション通知フックの例」を参照してください。

メモ: スキーマまたは電子メール通知フックを変更した場合は、ClearQuest MailServer Service を停止して再起動する必要があります。

Rational E-Mail Reader を使用した電子メール登録の有効化

Rational E-Mail Reader は、特別な書式の電子メール メッセージを受け取り、それに含まれる情報を ClearQuest が認識可能な書式に変換して、ClearQuest データベースに登録するツールです。これにより、ClearQuest クライアントの代わりに電子メールを使用してレコードの登録と変更を実行できます。

現在のリリースでは、Rational E-Mail Reader は ClearQuest MailServer Service という Windows サービスを使用して、これまでに説明した機能を実行します。ClearQuest MailServer Service の実行の開始、停止、一時停止、再開、スタートアップと復帰オプションの設定、使用の有効化と無効化を行うことができます。

Rational E-Mail Reader は、ClearQuest と RequisitePro の 2 つの Rational 製品で使用されます。設定プロセスの際に、どの製品を実装するか指定する必要があります。

ここでは、次のトピックについて説明します。

- 258 ページの「Rational E-Mail Reader のメールボックスの作成」
- 259 ページの「Rational E-mail Reader の設定」
- 263 ページの「Rational E-Mail Reader の追加オプション設定」
- 265 ページの「登録のための電子メールの書式設定」

Rational E-Mail Reader のメールボックスの作成

Rational E-Mail Reader では、ClearQuest データベースと RequisitePro プロジェクトそれぞれに対して電子メールアカウントが必要です。このアカウントが、変更依頼の登録と変更の送信先となる電子メールのメールボックスになります。ClearQuest MailServer Service はこのメールボックスに送信されたメッセージを受け取り、処理して、ClearQuest データベースまたは RequisitePro プロジェクトに依頼を登録します。

Rational E-Mail Reader を設定する前に、ユーザーと所属組織の電子メール管理者は以下の作業を行う必要があります。

- 電子メール システムが SMTP プロトコルと MAPI プロトコルのどちらを使用しているか確認する。
- ClearQuest データベースと RequisitePro プロジェクトそれぞれに専用の電子メール アカウントを作成する。
- 各電子メール アカウントに、SMTP サーバーと POP3 サーバー (SMTP プロトコルを使用している場合) または MAPI メール サーバー (MAPI プロトコルを使用している場合) のサーバー名または IP アドレスを指定する。

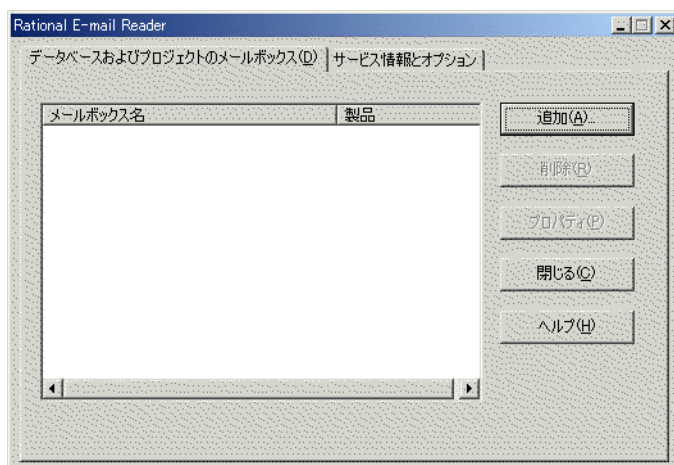
- 各電子メールアカウントに、POP3 サーバーのログイン情報とパスワードを指定する (SMTP プロトコルを使用している場合)。MAPI プロトコルを使用している場合、メールボックスのログインは、メールサーバー用コンピュータと同じオペレーティングシステム認証を使用するように設定する必要があります。
- MAPI を使用している各電子メールアカウントのメールボックスのログインと ClearQuest MailServer Service は、同じオペレーティングシステム認証を使用するように設定する必要があります。

Rational E-mail Reader の設定

Rational E-Mail Reader を設定する手順は、電子メールの送信に SMTP プロトコルと MAPI プロトコルのどちらを使用するかによって若干異なります。SMTP プロトコルを使用している場合は、手順 8 と 9 を省略してください。MAPI プロトコルを使用している場合は、手順 5 ～ 7 を省略してください。

SMTP プロトコル用に Rational E-Mail Reader を設定するには、以下の操作を行ってください。

- 1 [スタート] メニューの [ファイル名を指定して実行] をクリックします。
- 2 [ファイル名を指定して実行] ダイアログボックスで、E-Mail Reader のパスを入力します。たとえば、インストール時にすべての ClearQuest ファイルを保存するデフォルトディレクトリを C:¥Program Files¥Rational¥ClearQuest にした場合は、以下のように入力します。
C:¥Program Files¥Rational¥ClearQuest¥mailreader.exe
[OK] をクリックします。
- 3 [Rational E-Mail Reader] ダイアログボックスが表示されますこのダイアログボックスには以下の 2 つのタブがあります。
 - [データベースおよびプロジェクトのメールボックス] タブ



このタブには、電子メール アカウントが設定されている **ClearQuest** データベースと **RequisitePro** プロジェクトが一覧表示されます。このタブを使用して、新規 **ClearQuest** データベースと **RequisitePro** プロジェクトに、作成した新規電子メール アカウントを追加します。258 ページの「**Rational E-Mail Reader** のメールボックスの作成」を参照してください。

▫ [サービス情報とオプション] タブ

このタブには、**Rational E-Mail Reader** の追加オプションとステータス情報が含まれています。

4 [データベースおよびプロジェクトのメールボックス] タブで、[追加] をクリックして、新規 **ClearQuest** データベースまたは **RequisitePro** プロジェクトに新規電子メール アカウントを追加します。

▫ SMTP プロトコルを使用している場合は、手順 5 ～手順 7 まで実行した後、手順 10 に進んでください。

▫ MAPI プロトコルを使用している場合は、手順 8 に進んでください。

5 [電子メール ハンドラの選択] ダイアログ ボックスで、[電子メール プロトコル] として [SMTP] を選択します。[次へ] をクリックします。



6 [メール サーバーの選択] ダイアログ ボックスで、送信メール用の **SMTP** サーバー名と、受信メール用の **POP3** サーバー名を指定します。2 つのサーバー名は同じです。[次へ] をクリックします。

- 7 [メール サーバー アカウントの設定] ダイアログ ボックスが表示されます。ClearQuest データベースの専用電子メール アカウント情報を入力します。これには、POP3 電子メール アドレス、ログイン、パスワードが含まれます。この電子メール アカウントは Rational E-Mail Reader 専用で作成し、ほかのユーザーが使用しないようにする必要があります。[次へ] をクリックします。
- SMTP プロトコルを使用している場合は、手順 10 に進んで、手順 13 まで実行してください。
 - MAPI プロトコルを使用している場合は、このまま手順 8 ～手順 13 まで実行してください。
- 8 [電子メール ハンドラの選択] ダイアログ ボックスで、[MAPI] を選択します。[次へ] をクリックします。
- 9 [MAPI サーバー アカウントの設定] ダイアログ ボックスで、[電子メール サーバー] と [ログイン] に値を入力します。[次へ] をクリックします。

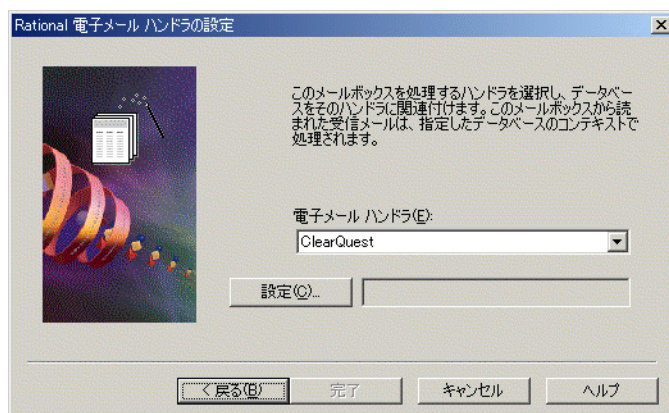


メモ: メールボックスのログインは、サーバー用コンピュータと同じオペレーティングシステム認証を使用するように設定する必要があります。

- 10 [Rational 電子メール ハンドラの設定] ダイアログ ボックスが表示されます。

Rational E-Mail Handler は、ClearQuest と Requisite Pro 両方をサポートします。しかし、電子メール アカウントが 1 つの場合は、Rational E-mail Reader を ClearQuest または RequisitePro どちらかの電子メールだけを受信するように設定できます。[Rational 電子メール ハンドラの設定] ダイアログ ボックスで [ClearQuest] を選択し、[設定] をクリックします。

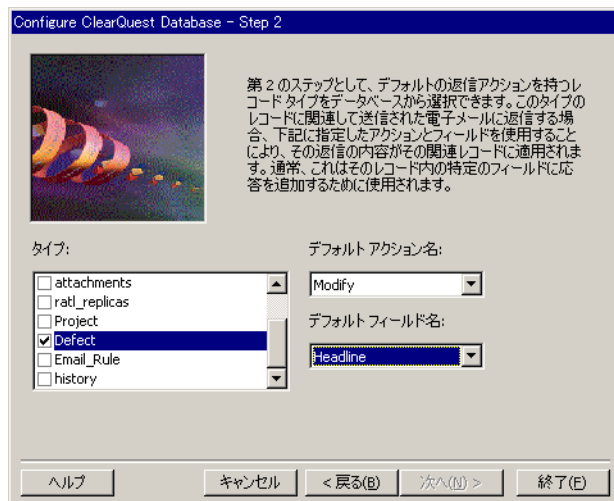
- 11 ClearQuest データベースを構成するためのウィザードで、Rational E-Mail Reader が ClearQuest システムへのログインに使用できる、有効なユーザー名とパスワードを入力します。受信メールのアクティビティがこのユーザー アカウントまで追跡されるようになります。



次に、スキーマ リポジトリとユーザー データベースの名前を選択します。[次へ]をクリックします。

ユーザー アカウントが Rational E-Mail Handler によって ClearQuest データベースで検証されます。ClearQuest データベースを構成するためのウィザードが表示されます。

- 12 ClearQuest データベースを構成するためのウィザードでは、電子メール メッセージの送信と応答のデフォルト アクションを設定します。



Rational E-Mail Reader を使用して送信された電子メール メッセージには、応答で行うアクションとメッセージ中の応答が追加されるレコードのフィールドが指定されていることがあります。しかし、電子メール メッセージでアクションもフィールドも指定されていない場合は、ここで指定するデフォルト設定が自動的に使用されます。

たとえば、Rational E-Mail Reader が SAMPL ユーザー データベースを使用するように設定して、レコードタイプとして [Defect]、デフォルト アクションとして [Modify]、デフォルト フィールドとして [Description] を選択します。すると、SAMPL データベース内の障害レコードに関する電子メールが送信され、その電子メールで応答のアクションもフィールドも指定されていない場合、メッセージが自動的に [Modify] アクションを使用して作成され、[Description] フィールドに置かれます。

メモ: 電子メール メッセージの書式を設定する場合、複数行のテキスト フィールド ([Description] フィールドなど) は、{ } で囲む必要があります。

レコード タイプ、デフォルト アクション、デフォルト フィールド名を選択し、[終了] をクリックします。

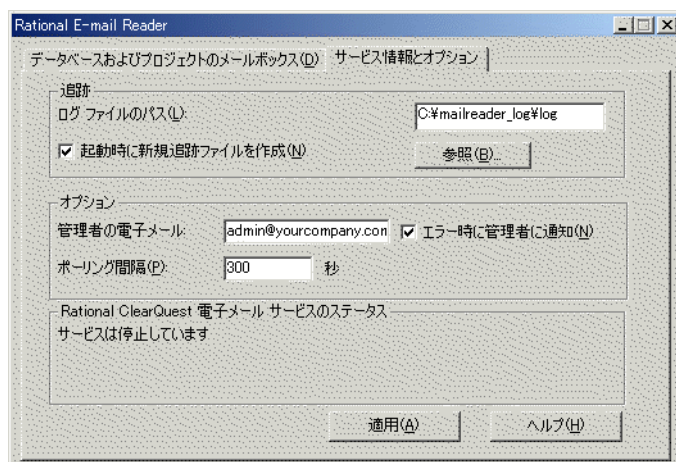
- 13 [Rational 電子メール ハンドラの設定] ダイアログ ボックスの [設定] ボックスに、接続とユーザー データベースの名前が表示されます。[終了] をクリックします。

ClearQuest の電子メール アカウントは、Rational E-Mail Reader の [データベースおよびプロジェクトのメールボックス] タブに一覧表示されます。

Rational E-Mail Reader の追加オプション設定

Rational E-Mail Reader の追加オプションを設定するには、[サービス情報とオプション] タブをクリックします。このタブで以下のことを設定できます。

- 電子メールを個別に追跡するための、ログ ファイルの作成。
- Rational E-Mail Reader のポーリング間隔の設定。ポーリング間隔は、Rational E-Mail Reader が電子メール メッセージをチェックする頻度を決定します。デフォルト値は 300 秒です。
- ClearQuest 管理者の電子メール アドレスを指定し、エラー発生時に Rational E-Mail Reader が管理者に通知するようにする。



メモ: このタブで行った変更は、ClearQuest MailServer Service を停止して再起動した後に有効になります。

これらのオプションを設定するには、次の操作を行います。

- 1 [ログ ファイルのパス] に、ログ ファイルを作成して保存するパスを入力します。
 - 2 サービスが再起動されるごとに現在のログ ファイルを削除して新規ログ ファイルを作成する場合は、[起動時に新規追跡ファイルを作成] をオンにします。
 - 3 エラー発生時に Rational E-Mail Reader からメッセージを受信するには、電子メール アドレスを入力して [エラー時に管理者に通知] をオンにします。また、デフォルト値の 300 秒を変更する場合は、時間を秒単位で入力します。
 - 4 [適用] をクリックします。
- [データベースおよびプロジェクトのメールボックス] タブを選択し、[閉じる] をクリックして終了します。

ClearQuest MailServer Service のプロパティ設定

サービスを開始する前に、[ClearQuest MailServer Service プロパティ] ダイアログ ボックスで追加のログオン情報を入力する必要があります。

ClearQuest MailServer Service のログオン情報を入力するには

- 1 [スタート] メニューから、[設定]、[コントロール パネル]、[管理ツール]、[サービス]、[ClearQuest MailServer Service] の順に選択します。右クリックして表示されるメニューで、[プロパティ] をクリックします。
- 2 [ClearQuest MailServer Service プロパティ] ダイアログ ボックスが表示されます。
[ログオン] タブをクリックします。

- 3 デフォルト ユーザー アカウントは [ローカル システム アカウント] に設定します。
ClearQuest MailServer Service がネットワーク リソースにアクセスする必要がある場合は、そのネットワーク リソースへの権限を持つアカウントをサービスに対して指定する必要があります。
- この場合は、デフォルトを変更して [アカウント] をオンにします。ユーザー アカウントを指定してから [パスワード] と [パスワードの確認入力] に入力します。
- 4 [全般] タブをクリックし、[スタートアップの種類] を確認します。
ほとんどの場合、[スタートアップの種類] は [自動] にします。こうすると、システムが再起動するごとにサービスも自動的に再起動します。
- 5 [OK] をクリックし、[サービス] ウィンドウを閉じます。
- 6 ClearQuest MailServer Service を設定しているコンピュータを再起動します。
再起動すると、システムですべての変更が更新されます。ClearQuest MailServer Service が自動的に再起動します。その後、サービスを使用して電子メールの処理を停止、無効化、再起動できます。

登録のための電子メールの書式設定

ClearQuest ユーザーが電子メールでレコードの登録や変更を行うと、必要なフィールドとリーガル値を含むすべてのスキーマ ルールが適用されます。たとえば、ユーザーが **Submit** アクションを指定したにもかかわらず、**Submit** 状態に必要なフィールドを含めなかった場合、E-mail Reader から送信者に電子メール メッセージがエラー メッセージと共に返信されます。

電子メールで ClearQuest レコードの登録や変更を行うときは、次のような書式を使用します。電子メール メッセージがこの書式に従っていない場合は、電子メールでの登録が中止され、電子メール メッセージが削除されます。

件名: <record type> <action> <record ID>

本文: [<fieldname>:]< legal value>

[<fieldname>:]< legal value>

{[<multiline_fieldname>:] これはデータ型が複数行のテキスト フィールドである場合のフィールド値の例です。中カッコは必須です。
}

メモ: 閉じ中カッコ (}) は、その中カッコが囲んでいる複数行のテキストとは別の行になければなりません。

電子メールの書式設定のガイドライン

電子メールで ClearQuest レコードの登録や変更を行うときは、次のガイドラインに従ってください。

- **<record type>** は必須です。これは件名行の先頭に指定する必要があります。
- 件名行のレコード タイプの後に **<action>** を指定します。**<action>** はレコードで実行されるアクションの名前です。スキーマでデフォルトのアクションを定義してある場合は、**<action>** を省略できます (267 ページの「電子メールの書式例」を参照)。ユーザーは、別のアクションを指定することでデフォルトを上書きできます。

メモ: **<action>** を指定しない場合、スキーマでデフォルトのアクションが設定されていないと、電子メールでの登録は拒否されます。

- 件名行の **<action>** の後に **<record ID>** を指定します。新しいレコードを登録するときは、**<record ID>** は必要ありません。
- 複数行のテキスト フィールドには、中カッコを使用します。閉じ中カッコ (}) は、その中カッコが囲んでいる複数行のテキストとは別の行になければなりません。たとえば、次のように指定します。

```
{ 説明: 説明の内容 ... この後に複数行の説明が続く ...  
}
```

- 変更する各フィールドの有効な値を入力します。有効な値が入力されていないと、Rational E-mail Reader から送信者にエラー メッセージが送信されます。
- フィールドは順不同でかまいません。
- すべての必須フィールドに値を入力します。
- 上書きするフィールドのみを書式に含めます。ほかのユーザーが行った更新を上書きしないようにするには、テキストを追加して入力済みのエントリを保持するフィールドを使用します。
- 障害 ID フィールドのような読み取り専用フィールドには、値を入力しないでください。ClearQuest では、読み取り専用フィールドは変更されません。
- ClearQuest 電子メール登録では、添付ファイルはサポートされていません。
- 件名行の先頭の関連標識 (RE:) は、必須ではありません。
- ClearQuest 電子メール通知メールでは、決められた書式が自動的に使用されます。ユーザーは、このメールをテンプレートとして使用し、電子メールでレコードの登録や変更を行うことができます。
- E-mail Reader では、電子メールでの登録が失敗したときに、ログを保存して電子メールメッセージで通知するように設定することができます。

電子メールの書式例

この例では、デフォルトのアクションは **Modify**、デフォルトのフィールドは **Notes_Log** になっています。件名行ではレコード タイプとレコード ID は必須ですが、デフォルトが設定されているためアクション名はオプションになります。

件名: RE: defect SAMPL00000017

本文: { 本日、この問題のパッチを社内イントラネット上に置きました。
このパッチをダウンロードして、顧客の問題の解決に使用してください。
}

新規レコードを登録する場合の書式の例

この例の電子メール メッセージでは、障害レコードを登録し、指定したフィールドを入力します。

件名: 障害の登録

本文: 見出し: 在庫レポートが正常に機能していません
重要度: 1- 最大
プロジェクト: ClassicProject
優先度: 2- 十分に注意
{ 詳細: 在庫レポートを実行したときに、レポートに 50 を超える品目があるとアプリケーションがクラッシュします。
}

レコードの状態を変更する場合の書式例

件名: defect SAMPL00001234 の終了

本文: {Note_Entry: 説明の手順に従って、エラーが発生しなくなったことを確認しました。
}
テスト方法: 手動

デフォルトのアクションを使用してレコードを変更する場合の書式例

この例では、レコードタイプが **Defect** の場合にデフォルト アクションが **Modify**、デフォルト フィールドが [説明] になるように、**ClearQuest** 管理者がスキーマを設定しています。件名行にアクションが指定されていない場合に、このデフォルトが使用されます。この電子メール メッセージでは、**Modify** アクションを使用して **SAMPL00000017** 障害を更新し、[説明] フィールドに値を追加します。

件名: defect SAMPL00000017

本文: { 許容品目数が 50 個以内の在庫レポートで問題を再現しようとした
 が、問題は発生しません。再試行して、詳細な情報を提供してください。
 }

データのインポートと エクスポート

13

この章では、データを 1 つの RationalClearQuest データベースからエクスポートして、別の ClearQuest データベースにインポートする方法と、データを別の障害追跡システムから ClearQuest へインポートする方法について説明します。

この章の内容は、次のとおりです。

- 269 ページの「データのインポートとエクスポートの概要」
- 270 ページの「インポート スキーマの作成」
- 272 ページの「インポートしたデータ用のデータベースの作成」
- 272 ページの「ClearQuest インポート ファイルの作成」
- 277 ページの「ClearQuest からのデータのエクスポート」
- 280 ページの「ClearQuest へのデータのインポート」

関連項目

- 388 ページの「cqload によるスキーマのコピー」
- 165 ページの「レコード フォームの再使用」

データのインポートとエクスポートの概要

インポート ツールとエクスポート ツールを使用すると、ほかの変更依頼システムから、または ClearQuest データベース間で、レコード、履歴、添付ファイルなどのデータを移動することができます。

インポート プロセスの開始

データのエクスポートとインポートの手順は次のとおりです。

- 新規インポート スキーマを作成するか、インポートするデータをサポートするように既存のスキーマを変更します。270 ページの「インポート スキーマの作成」を参照してください。
- 変更したインポート スキーマを使用して、既存のユーザー データベースをアップグレードするか、新規ユーザー データベースを作成して、インポート スキーマに関連付けます。272 ページの「インポートしたデータ用のデータベースの作成」を参照してください。

- ClearQuest インポートファイル形式を使用するテキスト ファイルに、データをエクスポートします。
 - ClearQuest の外部のシステムからデータをエクスポートする場合は、272 ページの「ClearQuest インポート ファイルの作成」を参照してください。
 - 既存の ClearQuest データベースからデータをエクスポートする場合は、277 ページの「ClearQuest からのデータのエクスポート」を参照してください。ClearQuest エクスポート ツールを使用して、インポート ファイルを作成することができます。
- ClearQuest インポート ツールを使用して、インポート ファイルからユーザー データベースにデータをインポートします。280 ページの「ClearQuest へのデータのインポート」を参照してください。

インポート プロセスのテスト

ClearQuest にすべてのデータをインポートする前に、次の手順に従って、データのサブセットに対してインポート プロセス全体をテストします。

- テスト データベースを作成します。
- 既存のデータのサブセットを ClearQuest インポート ファイルにエクスポートします。
- ClearQuest インポート ツールを実行して、データをインポートします。
- ClearQuest クライアントのデータを操作して、インポート スキーマが予定どおり機能していることを確認します。
- テスト結果に基づき、必要に応じてインポート スキーマを変更し、プロセスを再びテストします。

インポート スキーマの作成

ClearQuest にデータをインポートする前に、インポートが必要なデータをサポートするように、インポート スキーマを作成するか、既存のスキーマを変更します。インポート スキーマを作成する際は、次の点に注意してください。

- データの破損を防止するために、インポート スキーマのスキーマ リポジトリのデータ コード ページ値が、データをエクスポートするデータベース セットのデータ コード ページ値と同じ値になっている必要があります。詳細については、「第 2 章 ClearQuest とコード ページ」を参照してください。
- インポート スキーマには、データをサポートするための適切なレコードタイプ、フィールド、フィールド動作、データ型を設定する必要があります。また、このスキーマでは、インポートしたレコードに対して使用するアクションと状態遷移を設定する必要があります。

- インポート ファイルの状態名は、インポート スキーマに定義した状態名と一致する必要があります。異なる状態名を使用する場合は、新しい名前を使用するようにインポート ファイルを編集してください。たとえば、現在のアプリケーションでは **Submitted** という状態名を使用し、インポート スキーマではその状態を **New** と呼ぶ場合は、インポート ファイルを編集して、**Submitted** を **New** に置き換えます。
- 現在のシステムのすべてのフィールドと状態を、**ClearQuest** のフィールドと状態にマッピングする必要があります。エクスポートするフィールドと **ClearQuest** のフィールドとのマッピングを行わないと、そのフィールドのデータはインポートされません。状態フィールドをマッピングしないと、デフォルトで、すべてのレコードが **Submitted** 状態に設定されるため、状態モデルが無効になります。
- 既存のレコードを更新する場合は、インポート スキーマに **Modify** タイプのアクションを設定する必要があります。
- 重複レコードをインポートする場合は、インポート スキーマに **Duplicate** タイプのアクションを設定する必要があります。さらに、**Duplicate** 状態と **Duplicate** 状態に変更される前の元の状態も設定する必要があります。
- 重複レコードをインポートする場合は、レコード フォームに、重複レコードの表示と変更をサポートするコントロールが含まれている必要があります。親レコードの ID を表示するには、重複元コントロールを使用します。重複レコードの ID を表示するには、重複先コントロールを使用します。たとえば、**ClearQuest** の任意の事前定義スキーマで、**Defect** レコードタイプの [重複先] タブを参照してください。
- インポート スキーマには、ユーザーが、インポート済みのデータの表示と変更に使用できるレコード フォームを設定する必要があります。
- 履歴、添付ファイル、重複レコードをインポートする場合、または既存のレコードをアップグレードする場合は、インポート スキーマには、元の ID 値を格納するフィールドを設定する必要があります。272 ページの「元のレコード ID の指定」を参照してください。
- レコードに参照リストがある場合は、インポート スキーマに、参照先のレコード タイプを設定する必要があります。たとえば、プロジェクト用のフィールドがある障害レコードをインポートする場合は、障害レコードをインポートする前にプロジェクト レコードを作成してから、プロジェクト名を指定してプロジェクト レコードタイプを取り込む必要があります。

メモ: **ClearQuest** では、インポート時にデータ型が確認されます。インポートするレコードに、**ClearQuest** でサポートしていないデータ型が設定されている場合は、そのデータ型を **ClearQuest** でサポートしているほかのデータ型にマッピングできます。274 ページの「**ClearQuest** でサポートされているデータ型」を参照してください。

詳細については、『**Rational ClearQuest 管理ガイド**』の「第 4 章 **ClearQuest** スキーマの操作」と「第 5 章 スキーマのカスタマイズ」を参照してください。

元のレコード ID の指定

インポートした各レコードには新規レコード ID が割り当てられるので、インポート スキーマには元のレコード ID を格納するフィールドが必要です。元のレコード ID は、重複レコード、履歴、添付ファイルのインポート時と、既存のレコードの更新時に、データの一貫性を維持するために使用します。元のレコード ID は、親子のリンクを再作成するときにも使用します。また、元のレコード ID に基づいたレコードの検索にも役立ちます。

メモ: ClearQuest エクスポート ツールを使用して、別の ClearQuest データベースからデータをエクスポートする場合は、インポート ファイルの ID フィールドと元の (旧) レコード ID フィールドをマッピングします。

インポートしたデータ用のデータベースの作成

ClearQuest にデータをインポートする前に、インポートしたデータを保持するユーザー データベースを作成して、そのデータベースをインポート スキーマに関連付ける必要があります。詳細については、54 ページの「ユーザー データベースの作成」を参照してください。

インポート スキーマとして使用するスキーマに既に関連付けてある既存のユーザー データベースを使用できます。その場合は、最新のインポート バージョンのスキーマを使用して、ユーザー データベースをアップグレードしてください。ただし、ユーザー データベースのアップグレードに使用できるのは、同じスキーマの最新のバージョンのみです。異なるスキーマは使用できません。詳細については、97 ページの「スキーマのカスタマイズの概要」を参照してください。

ClearQuest インポート ファイルの作成

既存の ClearQuest データベースから既存のデータをエクスポートする場合は、ClearQuest エクスポート ツールを使用して、インポート ファイルを作成することができます。

別の障害追跡システムから ClearQuest にデータをインポートするには、まず、選択したツールを使用して、現在のシステムから、ClearQuest インポート ファイル形式の区切りテキスト ファイルにデータをエクスポートします。

レコード タイプごとに個別のインポート ファイルを作成する必要があります。レコードタイプに履歴と添付ファイルが含まれている場合は、レコード用、履歴用、添付ファイル用の少なくとも 3 つのインポート ファイルを作成する必要があります。

ここでは、レコードタイプ、履歴、添付ファイル用のインポート ファイルに必要なファイル形式の要件について説明します。

- レコード タイプ用のインポート ファイルの形式設定
- 履歴用のインポート ファイルの形式設定
- 添付ファイル用のインポート ファイルの形式設定

メモ: 前のシステムからデータをエクスポートする方法が、確実であることを確認します。エクスポートを何度かテストして、結果に矛盾がないかどうかをチェックしてください。270 ページの「インポート プロセスのテスト」を参照してください。

レコード タイプ用のインポート ファイルの形式設定

レコード タイプ用のインポート ファイルは、二重引用符で囲まれた文字列のリストで、区切り文字を付け、改行文字で終了します。最も一般的なファイル形式はカンマ区切りですが、コロン、セミコロン、パイプ、タブなども使用できます。

インポート ファイルの最初の行は、エクスポートされるフィールド名のリストです。以降の行は、最初の行にあるフィールド名の順番どおりに、各レコードのフィールド値を示します。インポート時に、各フィールドの値は、ClearQuest スキーマに定義済みの対応するフィールドタイプに変換されます。例を次に示します。

```
"id","state","submitdate","severity","priority","summary","description"
```

```
"1","Submitted","4/11/00 7:00:00","3-Workaround","3","The shortcut  
for ""Printing"" is grayed out","See summary --John"
```

```
"2","Opened","4/14/00 11:32:00","1-Crash","1","Can't log in to the  
system","I typed my login and the hourglass sign appears, but after  
15 minutes, I still can't type my password. There's an infinite loop  
somewhere."
```

State フィールドの値によって、インポート時のレコードの状態が決定します。ClearQuest では、インポートするレコードのフィールドのデータ型を確認しますが、特定の状態にするために必要なアクションは確認しません。

インポート ファイル形式に関する重要な考慮事項

インポート ファイルを形式設定する際は、以下の点に注意してください。

- カンマ区切りのデータ ファイルでは、フィールドを囲む引用符内であれば、フィールド値にカンマを埋め込むことができます。例を次に示します。

"I typed my login and the hourglass sign appears, but after 15 minutes, I still can't type my password. There's an infinite loop somewhere."

- フィールド値に埋め込んだ二重引用符は、さらに二重引用符で囲む必要があります。例を次に示します。

```
"The shortcut for "Printing all" does not work."
```

- フィールドが空の場合は、" "、"<<None>>"、"<<Unassigned>>"を使用します。
- 区切り文字の前後には、スペースを入れないでください。
- 参照フィールドをインポートするには、まず、参照先のレコード タイプの値をインポートします。インポート ファイル内の参照フィールドに対する値は、あらかじめ **ClearQuest** に存在している必要があります。これには、ユーザーとユーザー グループが含まれます。ユーザーとユーザー グループの作成の詳細については、171 ページの「ユーザーの操作」を参照してください。
- インポート ファイルでは、参照リストの項目をカンマで区切る必要があります。例を次に示します。

```
"Ref1, Ref2, Ref3"
```

- データ値内で復帰改行文字を使用できます。例を次に示します。

```
"To reproduce the error, follow these steps:<CR>
1. Launch Explorer.<CR>
2. Choose File > Open."
```

復帰改行文字を使用すると、複数の値を複数行のフィールドにインポートできます。

ClearQuest でサポートされているデータ型

インポート ファイル内のフィールドの値は、**ClearQuest** スキーマで対応するフィールドに定義されているデータ型に従って解釈されます。次の表に、**ClearQuest** でサポートされているデータ型の一覧を示します。

データ型	説明	動作
ATTACHMENT_LIST	タイプが Attachment であるフィールドのリスト	添付ファイルのパス名のリストとして解釈される。
DATE_TIME	SQL 日時	日時に変換される。
INT	SQL 整数	整数に変換される。
MULTILINE_STRING	無制限のサイズの変長文字列	このタイプに対するスキーマでの最大長としてそのまま挿入される。
REFERENCE	状態のあるレコード タイプまたは状態のないレコードタイプの表示名の参照	状態のないレコード タイプのキーとして解釈される。

データ型	説明	動作
REFERENCE_LIST	状態のあるレコードタイプ または状態のないレコード タイプの表示名に対する複数の 参照	参照のリストとして解釈される。
SHORT_STRING	最大長半角 254 文字の 可変長文字列	そのまま挿入される。
STATE	システム フィールド用に 予約済み	ClearQuest では、フィールドの 値は確認するが、該当の状態に するために必要なアクションは トリガしない。

履歴用のインポート ファイルの形式設定

履歴用の各インポート ファイルに含める履歴情報は、1 つのレコードタイプのもののみにする必要があります。複数のレコードタイプをインポートする場合は、履歴情報を含む履歴用のインポート ファイルをレコードタイプごとに作成する必要があります。

履歴用のインポート ファイル形式では、各履歴が 1 行になり、各エントリに、その履歴が属するレコードを識別する元の ID 番号が付くようにする必要があります。

その他のフィールドには、実行したアクションについての情報が入ります。次に、必要なフィールドを示します。

- 元のレコード ID
- タイムスタンプ
- ユーザー名
- 実行されたアクション
- 前の状態
- 新規の状態

メモ: ClearQuest エクスポート ツールを使用して、別の ClearQuest データベースから履歴データをエクスポートすると、エクスポート ツールによって、「display_name」というフィールドに元の (旧) レコード ID が保存されます。display_name フィールドを元の (旧) レコード ID にマッピングしてください。

履歴用のインポート ファイルの例を次に示します。この例では、この履歴に対応するレコードの元の ID を使用しています。

```
"id","timestamp","user_name","action_name","old_state","new_state"
"1","Apr 6 2000 8:31AM","srahman","close","open","closed"
"2","Apr 6 2000 9:40AM","srahman","verify","closed","verified"
```

日付の年は、完全な形で指定する必要があります。以下の日付書式を使用できます。

- "6 April 2003"
- "April 6, 2003 8:30:00"
- "8:30:00 Apr 6 2003"
- "4/6/2003 8:30:00PM"

添付ファイル用のインポート ファイルの形式設定

添付ファイル用の各インポート ファイルに含める添付ファイル情報は、1 つのレコードタイプのもののみにする必要があります。複数のレコードタイプをインポートする場合は、添付ファイルを含む添付ファイル用のインポート ファイルをレコードタイプごとに作成する必要があります。

添付ファイル用のインポート ファイルを形式設定する際は、以下の点に注意してください。

- 各添付ファイルを個別のファイルに含める必要があります。
- 添付ファイル用のインポート ファイル形式では、各エントリに、その添付ファイルが属するレコードを識別する元の ID 番号が付くようにする必要があります。各エントリのその他のフィールドには、レコードの各添付ファイル フィールドに関連付けられた添付ファイルを示します。
- 添付ファイル データ型の単一のフィールドに示す添付ファイルのパス名は、引用符で 1 つにまとめ、区切り文字で区切ります。

メモ: ClearQuest エクスポート ツールを使用して、別の ClearQuest データベースから添付ファイル データをエクスポートすると、エクスポート ツールによって、「display_name」というフィールドに元の (旧) レコード ID が保存されます。display_name フィールドを元の (旧) レコード ID にマッピングしてください。

次の例では、重複元の ID が 101 のレコードに 3 つの添付ファイルに関連付けています。スキーマには、添付ファイル フィールド attfield1 と attfield2 が記述されています。

```
"id","attfield1","attfield2"
```

```
"101","c:¥temp¥101_1.txt,c:¥temp¥101_2.txt","c:¥temp¥101_3.txt"
```

この例で、attfield1 には、2 つの添付ファイルが関連付けられています。これらのファイルのパス名は、カンマで区切られています。ClearQuest では、添付ファイルの参照だけではなく、実際の内容が格納され、ファイルを探して読み取る際にパス名を使用します。

ClearQuest からのデータのエクスポート

メモ: ClearQuest 以外の障害追跡システムからデータをエクスポートするには、データベース製造元のツールを使用する必要があります。272 ページの「ClearQuest インポート ファイルの作成」を参照してください。

既存の ClearQuest データベースからデータをエクスポートするには、ClearQuest エクスポート ツールを使用します。ClearQuest エクスポート ツールは、ClearQuest データベースからのデータを、ClearQuest インポート ツールに対して最適化された ASCII 形式のデータとしてエクスポートします。

メモ: データの破損を防止するために、データのエクスポートとインポートは、同じデータ コード ページ値を使用するデータベース間でのみ行う必要があります。さらに、インポート サーバーとエクスポート サーバーのオペレーティング システム コード ページが同じである必要があります。詳細については、「第 2 章 ClearQuest とコード ページ」を参照してください。

ClearQuest エクスポート ツールを実行するには

- 1 Windows の [スタート] メニューから、[プログラム]、[Rational Software]、[ClearQuest] の順にポイントし、[Rational ClearQuest エクスポート ツール] をクリックします。
- 2 ClearQuest エクスポート ツールで、データのエクスポート元にするデータベースが属しているスキーマ リポジトリをクリックします。デフォルトでは、ClearQuest の現在のバージョン番号に関連付けられたスキーマ リポジトリが選択されています。



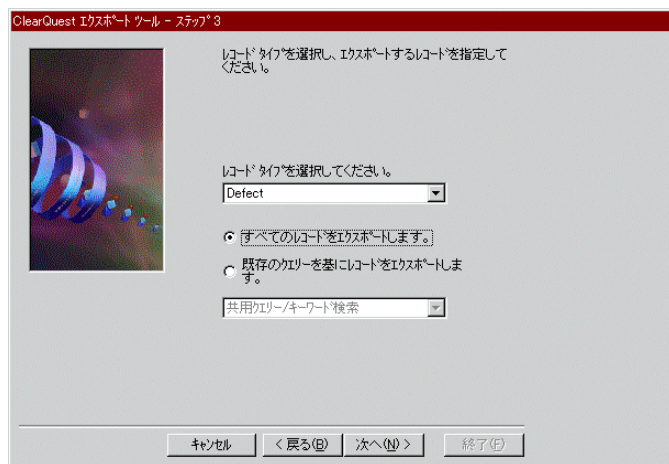
- 3 [次へ] をクリックします。

- 4 ユーザー名とパスワードを入力し、エクスポートするデータベースを選択して、[次へ]をクリックします。



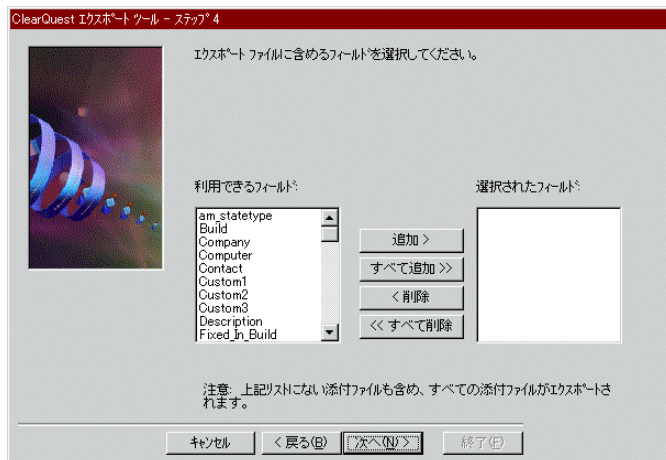
- 5 エクスポートするレコードタイプを選択します。

エクスポート ツールによって、自動的に、選択したレコードタイプの履歴、添付ファイル、重複情報がエクスポートされます。



- 6 [すべてのレコードをエクスポートする] または [既存のクエリーに基づいてレコードをエクスポートする] をクリックして、クエリーを選択し、[次へ]をクリックします。

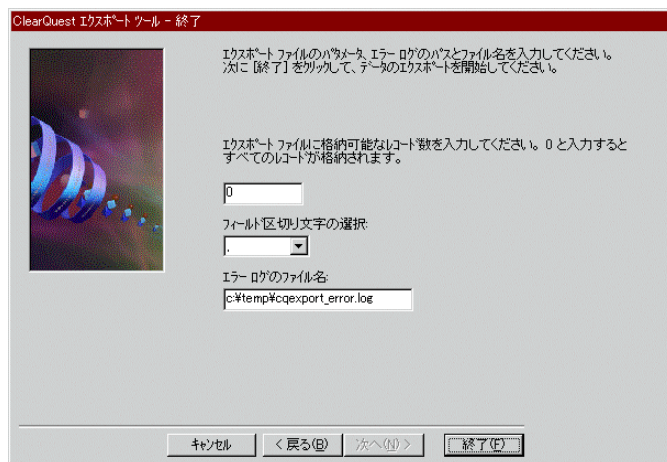
7 エクスポートするフィールドを選択します。



8 エクスポート ファイルの名前を入力します。records、history、attachments、duplicates 用に個別のファイルが作成されます。これらのファイルは、ClearQuest インポート ツールで必要になります。



9 最終エクスポート パラメータを入力します。



- 「0」を指定して、すべてのレコードを 1 つのファイルにエクスポートするか、ファイルごとのレコード数を入力して複数のファイルにエクスポートします。
- エラー ログのパスと名前を入力します。

10 [終了] をクリックすると、エクスポートが開始されます。

エクスポートが完了するとダイアログ ボックスが表示され、エラーが検出された場合にはエラーのリストが示されます。

メモ: エクスポート ツールを使用して ClearQuest からデータをエクスポートすると、records.txt ファイルには ID フィールドが含まれますが、Old_id フィールドの値は空白になります。履歴ファイルをインポートする場合は、古い ID を格納するフィールドを指定する必要があります。インポート処理を行うには、ID フィールドの値を Old_id フィールドに複製する必要があります。これにより、履歴ファイルをインポートできます。

ClearQuest へのデータのインポート

ClearQuest にデータをインポートするには、次の操作を完了している必要があります。

- 270 ページの「インポート スキーマの作成」
- 272 ページの「インポートしたデータ用のデータベースの作成」
- 272 ページの「ClearQuest インポート ファイルの作成」
または 277 ページの「ClearQuest からのデータのエクスポート」

レコードのインポート順序の決定

解決しているレコードも含め、システムにあるすべてのレコードをインポートすることを検討してください。すべてのレコードをインポートすることによって、プロジェクトに関する履歴情報にアクセスし、有効な管理レポートを直ちに生成できます。

ClearQuest インポート ツールを使用すると、レコード タイプの履歴と添付ファイルの情報も含めて、一度に 1 つのレコード タイプをインポートできます。複数のレコード タイプがある場合は、参照可能な順番でインポートする必要があります。たとえば、プロジェクト レコード タイプを参照する障害レコード タイプがある場合は、まず、プロジェクト レコードをインポートしてから、障害レコードをインポートします。

重複レコードがある場合は、それらを個別にインポートする必要があります。284 ページの「重複レコードのインポート」を参照してください。

ClearQuest インポート ツールの使用法

メモ: データの破損を防止するために、データのエクスポートとインポートは、同じデータ コード ページ値を使用するデータベース間でのみ行う必要があります。さらに、インポート サーバーとエクスポート サーバーのオペレーティング システム コード ページが同じである必要があります。詳細については、「第 2 章 ClearQuest とコード ページ」を参照してください。

インポート ファイルから ClearQuest にレコードをインポートするには、ClearQuest インポート ツールを使用します。

- 1 Windows の [スタート] メニューから、[プログラム]、[Rational Software]、[ClearQuest] の順にポイントし、[Rational ClearQuest インポート ツール] をクリックします。
- 2 ユーザー名とパスワードを指定してログインし、インポート先のスキーマ リポジトリを選択して、[OK] をクリックします。
- 3 [ClearQuest インポート ツール - ステップ 1/5] ダイアログ ボックスで、次の処理を行います。
 - a [ステップ 1] で、レコードのインポート先のレコード タイプの名前を選択します。
 - b [ステップ 2] で、インポートするデータ型を選択します。該当するすべてのタイプを選択します。選択したレコード タイプの履歴と添付ファイルの情報を、そのタイプのレコードをインポートするときに同時にインポートできます。
 - c 必要な場合は、[ステップ 3] で、既存のレコードを更新するかどうかを指定します。初めてインポートする場合は、この手順は不要です。

履歴情報、添付ファイル情報、重複情報をインポートする場合は、[はい] をクリックする必要があります。
 - d [次へ] をクリックします。

4 [ClearQuest インポート ツール - ステップ 2/5] ダイアログ ボックスで、次の処理を行います。

- a [レコード データ] に、インポートするデータを含むインポート ファイルの名前と場所を入力します。

破棄データ ログの名前と場所を入力します。このファイルには、新規ファイルを指定する必要があります。既存のファイルは使用できません。

正常に変換されなかったレコードは、指定した破棄データ ファイルに保存されます。このファイルを使用して、問題を修正し、レコードをインポートし直すことができます。285 ページの「インポート エラーの回復」を参照してください。

インポート ファイルに使用するフィールド区切り文字を選択します。

- b 履歴情報をインポートする場合は、[履歴データ] に履歴データを入力します。

インポートするデータを含むインポート ファイルの名前と場所を入力します。

破棄データ ログの名前と場所を入力します。このファイルには、新規ファイルを指定する必要があります。既存のファイルは使用できません。

正常に変換されなかった履歴データは、指定した破棄データ ファイルに保存されます。このファイルを使用して、問題を修正し、レコードをインポートし直すことができます。285 ページの「インポート エラーの回復」を参照してください。

インポート ファイルに使用するフィールド区切り文字を選択します。

- c 添付ファイル情報をインポートする場合は、[添付データ] に添付ファイル データを入力します。

インポートするデータを含むインポート ファイルの名前と場所を入力します。

破棄データ ログの名前と場所を入力します。このファイルには、新規ファイルを指定する必要があります。既存のファイルは使用できません。

正常に変換されなかった添付ファイル データは、指定した破棄データ ファイルに保存されます。このファイルを使用して、問題を修正し、レコードをインポートし直すことができます。285 ページの「インポート エラーの回復」を参照してください。

インポート ファイルに使用するフィールド区切り文字を選択します。

- d [次へ] をクリックします。

5 [ClearQuest インポート ツール - ステップ 3/5] ダイアログ ボックスで、次の処理を行います。

- a インポート状態の値に対して、インポート スキーマに定義した状態フィールドにマッピングするエクスポート フィールドを選択します。指定した状態フィールドによって、インポートしたときのレコードの状態が決定します。このフィールドが空の場合は、デフォルトで Submitted 状態になります。

- b 重複情報をインポートする場合は、インポート ファイルに重複レコードに関する情報を含めるかどうかをマークします。

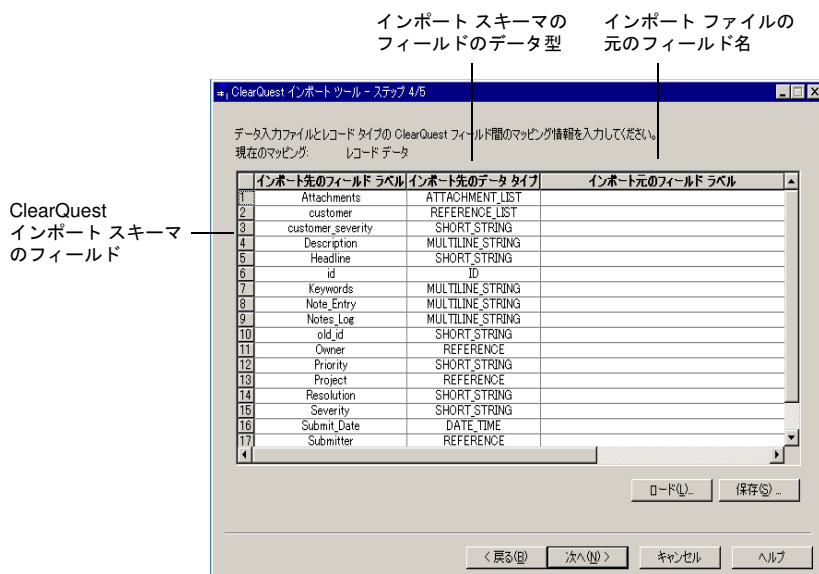
インポート ファイルに重複に関する情報を含める場合は、重複情報が含まれるエクスポート フィールド名を選択します。重複のインポートの詳細については、284 ページの「重複レコードのインポート」を参照してください。

- c レコードの更新、重複、履歴、添付ファイルのいずれかをインポートする場合は、インポート スキーマ内で、エクスポート データの元のレコード ID (または他の一意の識別子) を含むフィールドを選択します。

履歴、添付ファイル、重複レコードをインポートする場合は、ClearQuest で、インポートされるレコードの以前の一意の識別子 (通常は古い ID 番号) を追跡する必要があります。ClearQuest では、添付ファイル、履歴情報、重複レコードが属するレコードを認識する必要があります。

- d [次へ] をクリックします。

- 6 [ClearQuest インポート ツール - ステップ 4/5] ダイアログ ボックスで、インポート ファイルのフィールドと、ClearQuest インポート スキーマに作成したフィールドとの間に 1 対 1 のマッピングを作成します。フィールド名が同じ場合は、マッピングは自動的に行われます。



- インポート ファイルの 1 つのフィールドを ClearQuest スキーマの複数のフィールドにマッピングすることはできません。
- インポート ファイルの元の ID をその ID を格納する宛先フィールドにマッピングする必要があります。

メモ: ClearQuest エクスポート ツールを使用して、インポート ファイルを作成した場合、元の ID は、エクスポート ファイルの「ID 列」に格納されます。この「ID」フィールドは、元の ID フィールドにマッピングする必要があります。

マッピングを保存するには

- a [保存] をクリックします。
- b マッピング プロファイルを保存する場所を参照し、*.txt 拡張子を付けて名前を入力し、[保存] をクリックします。

既存のマッピングを再使用するには

- a [ロード] をクリックします。
- b 既存のマッピング プロファイルを選択し、[OK] をクリックします。

7 [ClearQuest インポート ツール - ステップ 5/5] ダイアログ ボックスで、次の操作を行います。

- a インポートの要約を参照して、設定が正しいことを確認します。
- b エラーがいくつ発生したらインポート プロセスを終了するかを指定します。
- c [インポート] をクリックします。

重複レコードのインポート

すべてのレコードをインポートする場合は、同時に重複レコードをインポートできます。重複レコードをインポートする際は、次の要件を考慮してください。

- インポートする重複には、元の ID のポインタが必要です。
- ClearQuest インポート スキーマのどのフィールドを元のレコード ID にマッピングするかを指定する必要があります。ClearQuest では、重複レコードの親の検索に元のレコードを使用します。子レコードをインポートする前に、親レコードをインポートする必要があります。
- ClearQuest では、重複レコードは、Duplicate 状態とアクションを使用して処理されます。インポート スキーマに、この状態とアクションが組み込まれていることを確認してください。
- さらに、レコードが Duplicate 状態に変更される前の元の状態もインポート スキーマに組み込まれている必要があります。

メモ: ClearQuest エクスポート ツールを使用して、別の ClearQuest データベースからデータをエクスポートする場合は、重複レコード用のインポート ファイルが自動的に作成されます。

重複レコード専用のインポート

重複レコードをインポートする別の方法として、履歴と添付ファイルに対して行ったように、重複レコード用のインポート ファイルを別に作成します。インポート ファイルを別に作成すると、重複以外のすべてのレコード (子レコードを除く) をインポートした後で、重複をインポートできます。

重複レコードを別にインポートする場合は、[ClearQuest インポート ツール - ステップ 1/5] ダイアログ ボックスで、既存のレコードの更新に対して [はい] を選択します。

インポート エラーの回復

インポート中にエラーが発生した場合は、次のファイルが作成されます。

- 破棄データ ログ ファイル: インポートされなかったレコードは、上記の手順 3 で定義した名前と場所のエラー ファイルに保存されます。
- **errlog.txt**: すべての障害に関する詳細情報は、TEMP ディレクトリに保存したエラー ファイルに保存されます。ClearQuest で破棄データ ログに問題が発生した場合は、エラー メッセージが、破棄データ ログと同じディレクトリにあるテキスト ファイルに保存されます。このファイルの名前は次のいずれかになります。

- 「.log」が付いた破棄データ ログ ファイル名
- **errlog.txt**

- TEMP ディレクトリに保存されるファイルには、ClearQuest インポート ツールのステータス ログに出力されたすべての情報のコピーが含まれます。このファイルは、テキスト ファイルであり、次の命名規則に従って名前が付けられます。

`import_status_<year><month><day>_<hour><minute>_<second>.txt`

たとえば、`import_status_20021012_1355_01.txt` は、2002 年 10 月 12 日の午後 1 時 55 分 1 秒に作成されたステータス ファイルを示します。

これらの問題のレコードをインポートし直すには

- 1 **errlog.txt** ファイルをチェックして、発生したエラーのタイプを確認します。
- 2 インポートされなかったレコードがあるエラー ファイルを開きます。
- 3 レコード内のエラーを修正します。エラー ファイルにインポート ファイル形式が使用されていることを確認します。273 ページの「レコード タイプ用のインポート ファイルの形式設定」を参照してください。
- 4 インポート プロセスを再び実行しますが、今回は、インポート ファイルとしてエラー ファイルを指定します。

既存のレコードの更新

ClearQuest インポート ツールを使用して、前にインポートしたレコードを更新できます。たとえば、前のシステムで、ClearQuest に既にインポートしているレコードを変更した場合は、その新規データを使用して、インポート済みのレコードを更新できます。

警告: 前のシステムと ClearQuest の両方でレコードを変更しないようにしてください。

ClearQuest で既に変更済みの既存のレコードをアップグレードすると、変更内容が失われます。

既存のレコードをアップグレードするには、280 ページの「ClearQuest へのデータのインポート」に示した手順に従います。手順 3 で、[既存レコードをアップグレード] を選択します。その他の手順は、すべて同じです。

ClearQuest のスキーマと パッケージ

A

Rational ClearQuest は、事前定義されたスキーマをいくつか備えています。Windows 版の Rational Suite 製品用のスキーマや、必要なレコード タイプのみを持ち、スキーマの作成に使用する Blank スキーマがあります。

ClearQuest の事前定義された各スキーマは、Rational Suite との統合を実現する特定の機能またはサポートを提供するスキーマ パッケージで構成されています。スキーマに 1 つまたは複数のこれらのパッケージを追加することで、既存のスキーマをカスタマイズできます。

この章の内容は、次のとおりです。

- 288 ページの「ClearQuest の事前定義されたスキーマ」
- 289 ページの「ClearQuest のパッケージ」
- 297 ページの「Defect レコード タイプの状態モデル」
- 299 ページの「EnhancementRequest レコード タイプの状態モデル」
- 300 ページの「パッケージの状態タイプ モデル」

ClearQuest を新しいバージョンにアップグレードする際に、新しいバージョンの新機能を利用するため、既存のスキーマへのアップグレードパッケージの適用が必要な場合があります。適用が必要なアップグレード パッケージのリストについては、Rational ClearQuest のリリース ノートを参照してください。

スキーマへのパッケージの追加方法については、「第 6 章 パッケージの適用」を参照してください。スキーマのカスタマイズ方法については、「第 5 章 スキーマのカスタマイズ」を参照してください。

警告：パッケージの追加は、慎重に計画してください。パッケージは、スキーマに追加すると削除することはできません。このような場合は、パッケージが存在するすべてのスキーマ バージョンを削除してください。スキーマ バージョンを削除できるのは、ユーザー データベースに適用していない場合のみです。

ClearQuest の事前定義されたスキーマ

次の表に、ClearQuest の事前定義されたスキーマのリストを示します。

スキーマ	説明	含まれているパッケージ
Blank	レコード タイプのみが含まれます。 Blank を使用して新規スキーマを作成します。	なし
Common	すべての事前定義されたスキーマに共通のフィールドとレコードタイプが含まれます。ClearQuest の各スキーマは、このスキーマと 1 つまたは複数のパッケージで構成されます。	なし
DefectTracking	ソフトウェアの開発環境において、ClearQuest を利用して障害の追跡を行う際に必要なフィールドが含まれます。	Attachments、Customer、Email、History、Notes、Project、Resolution
AnalystStudio	Rational AnalystStudio と共に使用します。Rational Rose と RequisitePro を操作するフィールドとルールが含まれます。	Attachments、Email、EnhancementRequest、History、Notes、Repository、RequisitePro、Resolution、TeamTest
Development Studio	Rational DevelopmentStudio と共に使用します。Rational の Purify、Quantify、PureCoverage を操作するフィールドとルールが含まれます。	Attachments、Email、EnhancementRequest、History、Notes、PQC、Repository、RequisitePro、Resolution、TeamTest
TestStudio	Rational TestStudio と共に使用します。Rational の TeamTest、RequisitePro、Purify、Quantify、PureCoverage を操作するフィールドとルールが含まれます。	Attachments、Email、EnhancementRequest、History、Notes、PQC、Repository、RequisitePro、Resolution、TeamTest
Enterprise	Rational EnterpriseStudio と共に使用します。ほとんどの Rational Suite 製品を操作するフィールドとフックが含まれます。 このスキーマは UCM を使用可能です。	AMStateTypes、Attachments、BaseCMActivity、Email、EnhancementRequest、History、Notes、PQC、Resolution、Repository、RequisitePro、TeamTest、UCMPolicyScripts、UnifiedChangeManagement、TeamTest

スキーマ	説明	含まれているパッケージ
UnifiedChange Management (UCM)	UCM との統合のサポートを提供し、ClearQuest をセットアップして事前定義された ClearCase ポリシーを使用します。	AMStateTypes、Attachment、BaseCMAActivity、Email、History、Notes、Resolution、UCMPolicyScripts、UnifiedChangeManagement

ClearQuest のパッケージ

この項目では、各 ClearQuest パッケージの最新バージョンについて説明します。前のパッケージバージョンは異なるレコードタイプとフィールドを作成することがあります。パッケージの中には、読み取り専用のものがあり、この場合は機能を変更できません。

パッケージ	説明	追加点または変更点	フィールド
AMBaseActivity	Rational ClearQuest ProjectTracker に追加サポートを提供します。	有効なレコードタイプのフォームに [メイン] タブを追加します。	有効なレコードタイプで変更または追加されるフィールド <ul style="list-style-type: none"> Headline Owner Description
AMStateTypes	Rational 統一変更管理とその状態タイプに追加サポートを提供します。 Waiting、Ready、Active、Complete の状態タイプにスキーマの状態をマッピングする必要があります。	レコードタイプは追加しません。	有効なレコードタイプに追加されるフィールド <ul style="list-style-type: none"> am_statetype
AMWorkActivity Schedule	Rational ClearQuest Project Tracker を使用して、Rational ClearQuest と Microsoft Project 2000 を統合するために必要なスケジューリング属性を提供します。 AMWorkActivity Schedule レコードタイプファミリーを使用して、Rational ClearQuest Project Tracker で作成され、更新されるレコードをクエリーできます。	有効なスキーマに AMSchedule レコードタイプファミリーを定義、追加します。このパッケージで有効化されるレコードタイプは、このレコードタイプファミリーに追加されます。 有効なレコードタイプに [スケジュール] タブを追加します。	有効な AMSchedule レコードタイプに追加されるフィールド <ul style="list-style-type: none"> am_planned_start_date am_planned_end_date am_planned_work am_planned_rem_work am_planned_duration am_planned_rem_duration am_actual_start_date am_actual_end_date am_actual_work

パッケージ	説明	追加点または変更点	フィールド
Attachments (読み取り専用)	レコードの関連ファイルを追加/削除できます。	有効なレコードタイプに[添付]タブを追加します。	有効なレコードタイプに追加されるフィールド <ul style="list-style-type: none"> ▪ Attachments
BaseCMActivity	BaseCMActivity レコードタイプをサポートします。軽いアクティビティレコードタイプとして、UCMとEnterpriseスキーマに含まれています。そのままDefectレコードタイプの代わりに使用して、統一変更管理(UCM)に対応したり、新規レコードタイプとして作成できます。 詳細については、『Rational ClearCase プロジェクト管理ガイド』を参照してください。	BaseCMActivity レコードタイプを追加します。	BaseCMActivity レコードタイプに含まれるフィールド <ul style="list-style-type: none"> ▪ Owner ▪ Description ▪ Headline
CharacterSetValidation	ネイティブクライアントからレコードに入力されるすべてのデータに対して、スキーマリポジトリのClearQuestデータコードページと同じコードページによるものであることを確認して、データの破損を防止します。 パッケージウィザードやapply_character_set_validation_packageを使用して、CharacterSetValidationパッケージを適用することもできます。 詳細については、39ページの「CharacterSetValidationパッケージの概要」を参照してください。	すべての非状態なしレコードタイプに対して、次の2つのPerlフックを追加します。 <ul style="list-style-type: none"> ▪ アクセス制御フック CheckCodePageMismatch ▪ ベース検証フック CharacterSetValidation 新しい状態なしレコードタイプ_ratl_data_code_pageを追加します。	

パッケージ	説明	追加点または変更点	フィールド
ClearCase (読み取り専用)	Rational ベース ClearCase との統合の 基礎的サポートを提供 します。このパッケー ジは事前定義された ClearCase ポリシーに 使用する ClearQuest を 設定しません。 ClearCase 管理者が セットアップする必要 があります。	cc_change_set と cc_vob_object レコード タイプを追加します。 有効なレコード タイプ に [ClearCase] タブを追 加します。	cc_change_set レコード タイプ に含まれるフィールド <ul style="list-style-type: none"> objects cc_vob_object レコード タイプ に含まれるフィールド <ul style="list-style-type: none"> name object_oid vob_family_uuid 有効なレコード タイプに追加 されるフィールド <ul style="list-style-type: none"> cc_change_set
ContentStudio	Rational Suite ContentStudio との統合 をサポートします。	ContentChangeRequest レコード タイプを追加 します。	ContentChangeRequest レコー ドタイプに含まれるフィールド <ul style="list-style-type: none"> Description Headline Note_Entry Notes_Log Owner vgnAssignee vgnCMS vgnDueDate vgnManagementID vgnProjectPath vgnTaskName
Customer	顧客データと、障害/ 変更追跡システムとの 統合をサポートします。	状態なしレコード タ イプ Customer を追加 します。 選択したレコード タイプに、顧客情報の 参照フィールドも追加 します。	Customer レコード タイプに 含まれるフィールド <ul style="list-style-type: none"> Attachment CallTrackingID Company Description Email Fax Name Phone 有効なレコード タイプに追加 されるフィールド <ul style="list-style-type: none"> Customer Customer_Severity

パッケージ	説明	追加点または変更点	フィールド
Email (読み取り専用)	レコードが変更された際の電子メールによる自動通知をサポートします。	状態なしレコードタイプ Email_Rule を作成します。 有効なレコードタイプ Send_Email_Notif に基本アクションを追加します。アクションが呼び出されると、常にこの基本アクションが電子メールルールを起動します。	Email_rule レコードタイプに含まれるフィールド <ul style="list-style-type: none"> ▪ Actions ▪ Action_Types ▪ CC_Actioner ▪ CC_Addr_fields ▪ CC_Additional ▪ CC_Groups ▪ CC_Users ▪ Change_Fields ▪ Display_Fields ▪ Entity_Def ▪ From_Addr ▪ Include_Defect ▪ Is_Active_Rule ▪ Filter_Query ▪ Name ▪ Operator_Value ▪ Show_Previous ▪ Source_States ▪ Subject_Fields ▪ Target_States ▪ To_Additional ▪ To_Addr_Fields ▪ To_Groups ▪ To_Users
Enhancement Request	製品拡張依頼の追加のレコードタイプをサポートします。	Enhancement Request レコードタイプを追加します。	EnhancementRequest レコードタイプに含まれるフィールド <ul style="list-style-type: none"> ▪ Customer_Company ▪ Customer_Email ▪ Customer_Name ▪ Customer_Phone ▪ Customer_Priority ▪ Description ▪ Headline ▪ Keywords ▪ Owner ▪ Priority ▪ Product ▪ Product_Area ▪ Request_Type ▪ Submit_Date ▪ Submitter ▪ Target_Release

パッケージ	説明	追加点または変更点	フィールド
History (読み取り専用)	レコードに実行されたすべてのアクションの履歴アカウントを保持できます。	有効なレコードタイプに [履歴] タブを追加します。	追加されるフィールドはありません。
Notes	日付とユーザーによって、レコードに入力されたすべての注記の履歴アカウントを保持できます。	有効なレコードタイプに [メモ] タブを追加します。また、基本アクション Init_Note_Entry を有効なレコードタイプに追加し、Note_Entry の値を削除します。	有効なレコードタイプに追加されるフィールド <ul style="list-style-type: none"> ▪ Note_Entry ▪ Notes_Log
PQC (読み取り専用)	Rational Purify、Quantify、PureCoverage との統合をサポートします。	有効なレコードタイプのフォームに [PQC] タブを追加します。	有効なレコードタイプに追加されるフィールド <ul style="list-style-type: none"> ▪ PQC_DiagnosticTool ▪ PQC_Executable ▪ PQC_TestCommand ▪ PQC_TestTool ▪ PQC_Stack ▪ PQC_StackID
Project	プロジェクトによって、レコードを追跡できます (メモ: UCM パッケージの「Project」コンセプトとは関係ありません。)	状態なしレコードタイプ Project を作成します。	Project レコードタイプに含まれるフィールド <ul style="list-style-type: none"> ▪ Name ▪ Description 有効なレコードタイプに追加されるフィールド <ul style="list-style-type: none"> ▪ Project
Repository (読み取り専用)	Rational RequisitePro、Rational Administrator、Rational TeamTest に必要なサポートを提供します。	状態なしレコードタイプ RAProject を作成します。	RAProject レコードタイプに含まれるフィールド <ul style="list-style-type: none"> ▪ Name ▪ TT_Repo (TeamTest Repository を参照してください) ▪ RA_Project_Path 有効なレコードタイプに追加されるフィールド <ul style="list-style-type: none"> ▪ RAProject

パッケージ	説明	追加点または変更点	フィールド
RequisitePro (読み取り専用)	Rational RequisitePro との統合をサポートし ます。	状態なしレコードタイプ Requirement と RequirementMap を追加します。 有効なレコードタイプに [要求] タブを追加します。 また、有効なレコードタイプに ASCQIBase の基本アクションを追加します。	Requirement レコードタイプに含まれるフィールド <ul style="list-style-type: none"> ▪ Name ▪ Project_Name ▪ Req_GUID ▪ Req_ID ▪ Requirement ▪ Tag RequirementMap レコードタイプに含まれるフィールド <ul style="list-style-type: none"> ▪ CQBackReqListAttName ▪ CQDatabase ▪ CQDatabasePath ▪ CQDialogTitle ▪ CQEntityDefName ▪ CQHelpContextID ▪ CQModifyAction ▪ CQReqListAttName ▪ CQRepoProjectAttName ▪ HelpFileName ▪ RPAttrGUID ▪ RPHelpContextID ▪ RPProjectName ▪ RPProjectPath ▪ RPReqTypeGUID 有効なレコードタイプに追加されるフィールド <ul style="list-style-type: none"> ▪ Requirements_List
Resolution	レコードがどのように解決されたかを追跡できるサポートを追加します。 Not_Resolved、Resolved の状態タイプにスキーマの状態をマッピングする必要があります。	有効なレコードタイプに [解決] タブを追加します。	有効なレコードタイプに追加されるフィールド <ul style="list-style-type: none"> ▪ Resolution ▪ Resolution_Statetype (読み取り専用)

パッケージ	説明	追加点または変更点	フィールド
TeamTest (読み取り専用)	Rational TeamTest との統合をサポートします。	指定するレコードタイプに [テスト データ] と [環境] タブを追加します。 状態なしレコードタイプ TestInput も追加します。	有効なレコードタイプに追加されるフィールド <ul style="list-style-type: none"> ▪ Build ▪ Company ▪ Computer ▪ Contact ▪ Custom1 (変更可能) ▪ Custom2 (変更可能) ▪ Custom3 (変更可能) ▪ Fixed_In_Build ▪ Hardware ▪ Log ▪ Log_Folder ▪ old_internal_id ▪ Operating_System ▪ Other_Environment ▪ Resolution_Description ▪ Requirement ▪ Requirement_ID ▪ Test_Case ▪ Test_Case_UID ▪ Test_Script ▪ Test_Script_ID ▪ Test_Source_UID ▪ Test_Input_List ▪ Verification_Point TestInput レコードタイプに追加されるフィールド <ul style="list-style-type: none"> ▪ Test_Input_Name ▪ Test_Input_ID ▪ Source_UID
UCMPolicy Scripts	3 つのグローバル スクリプトを追加することで、UnifiedChange Management (UCM) パッケージをサポートします。	レコード タイプは追加しません。	

パッケージ	説明	追加点または変更点	フィールド
UnifiedChange Management (UCM) (読み取り専用)	<p>Rational ClearCase バージョン 4.0 以上との統合を実現することで、UCM プロセスをサポートし、ClearCase のプロジェクト VOB と ClearQuest ユーザー データベースをリンクします。UCMPolicyScripts パッケージが必要です。BaseCMActivity パッケージと共に使用できます。</p> <p>Waiting、Active、Ready、Complete の状態タイプにスキーマの状態をマッピングする必要があります。</p>	<p>UCMUtilityActivity レコードタイプを追加します。</p> <p>状態なしレコードタイプ UCM_Project を追加します。</p> <p>また、共有フォルダ内のクライアント ワークスペースに UCM クエリーを追加します。</p> <p>有効なレコードタイプに ucm_base_synchronize アクションを追加します。</p>	<p>UCMUtilityActivity レコードタイプに含まれるフィールド</p> <ul style="list-style-type: none"> ▪ am_statetype ▪ Description ▪ Owner ▪ Headline ▪ ucm_vob_object ▪ ucm_stream_object ▪ ucm_stream ▪ ucm_view ▪ ucm_project <p>UCM_Project レコードタイプに含まれるフィールド</p> <ul style="list-style-type: none"> ▪ name ▪ ucm_vob_object ▪ ucm_chk_before_deliver ▪ ucm_chk_before_work_on ▪ ucm_chk_mstr_before_dlvr ▪ ucm_cq_act_after_deliver <p>有効なレコードタイプに追加されるフィールド</p> <ul style="list-style-type: none"> ▪ am_statetype ▪ ucm_vob_object ▪ ucm_stream_object ▪ ucm_stream ▪ ucm_view ▪ ucm_project

パッケージ	説明	追加点または変更点	フィールド
Visual SourceSafe (読み取り専用)	Microsoft Visual SourceSafe との統合をサポートします。	<p>状態なしレコードタイプ SSOBJECT と SCSnapObject を追加します。</p> <p>有効なレコードタイプのフォームに [SourceSafe] タブを追加します。</p>	<p>有効なレコードタイプに追加されるフィールド</p> <ul style="list-style-type: none"> VSSChangeSet <p>SSOBJECT レコードタイプに追加されるフィールド</p> <ul style="list-style-type: none"> CQDefects VSSCheckOutState VSSFileName VSSSpec VSSUser VSSVersion <p>SCSnapObject レコードタイプに追加されるフィールド</p> <ul style="list-style-type: none"> CreatedBy CreatedOn Label SnapElements

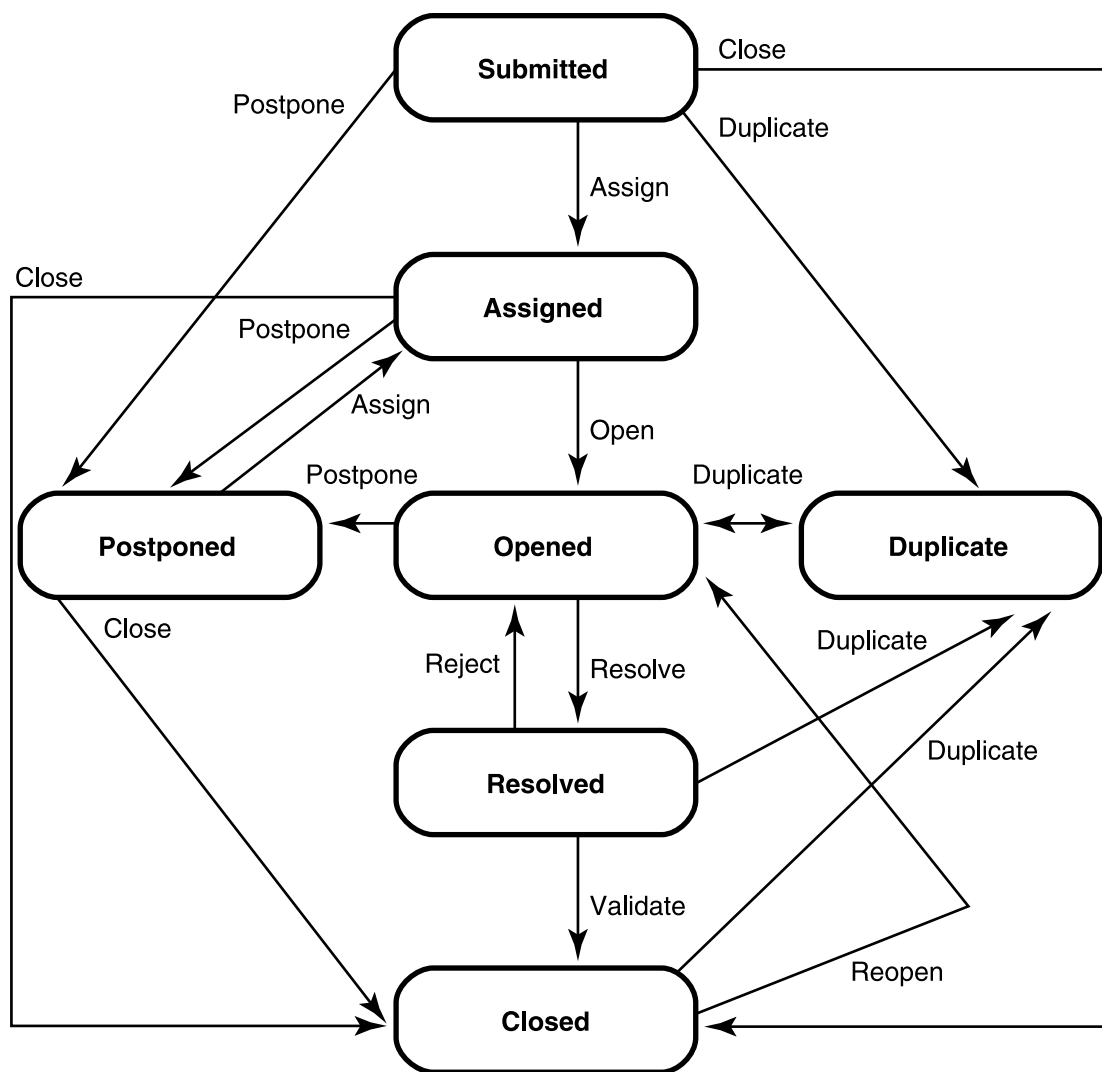
Defect レコード タイプの状態モデル

次の表に、Defect レコード タイプの状態のリストを示します。

状態	説明
Submitted	障害登録時の初期状態
Assigned	チーム メンバーにアサインされた状態
Opened	作業中
Resolved	修正済み
Closed	確認済み
Duplicate	ほかの障害と重複
Postponed	次のリリースまたは反復まで延期

事前定義された各スキーマの Defect レコード タイプの状態モデルは同じです。この情報は、Defect レコード タイプの状態遷移マトリックスでも参照できます。

図 7 Defect レコード タイプの状態モデル



すべての Duplicate アクションは、Unduplicate アクションを使用して元に戻すことができます。

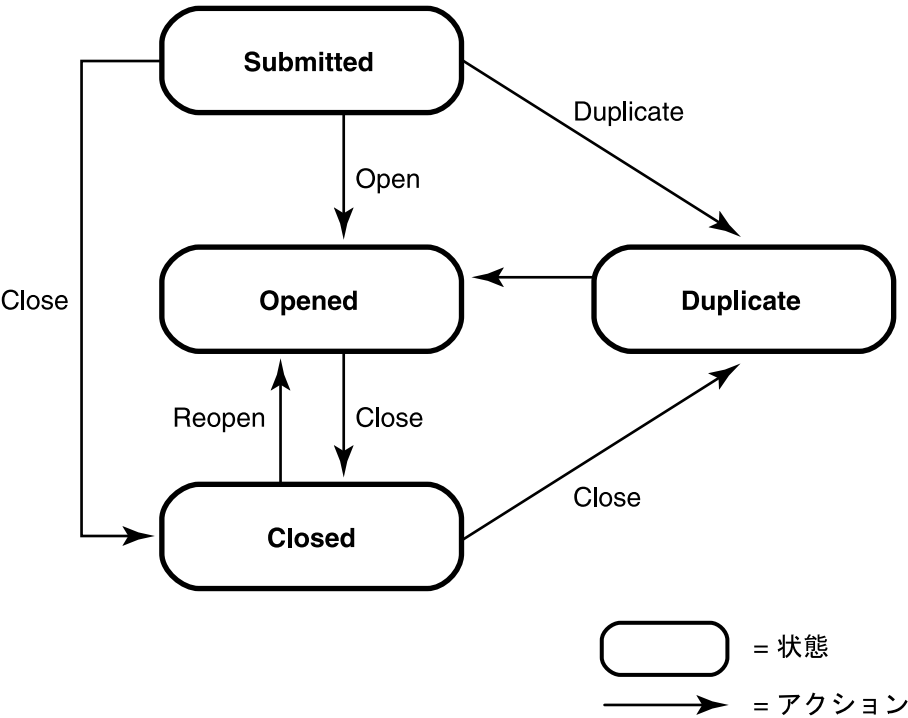
EnhancementRequest レコード タイプの状態モデル

次の表に、EnhancementRequest レコード タイプの状態のリストを示します。

状態	説明
Submitted	拡張依頼登録時の初期状態
Opened	作業中
Closed	確認済み
Duplicate	ほかの拡張依頼と重複

次の状態モデル図は、アクションの結果として EnhancementRequest のレコード タイプがどのように 1 つの状態から別の状態に移行するかを示しています。この情報は、EnhancementRequest レコード タイプの状態遷移マトリックスでも参照できます。

図 8 Enhancement レコード タイプの状態モデル



パッケージの状態タイプ モデル

状態タイプは、状態モデルでの状態の役割を定義するラベルです。ClearQuest のスキーマの中には、状態モデルの各状態を特定の状態タイプに割り当てまたはマッピングする必要のあるパッケージを持つものがあります。たとえば、UnifiedChangeManagement スキーマとパッケージでは、状態タイプを使用して特定の ClearCase アクションを呼び出すことがあります。ClearQuest の Resolution パッケージでは、状態タイプを使用して、レコードが Resolved 状態タイプにマッピングされた状態に移行すると、特定のフックが起動されます。

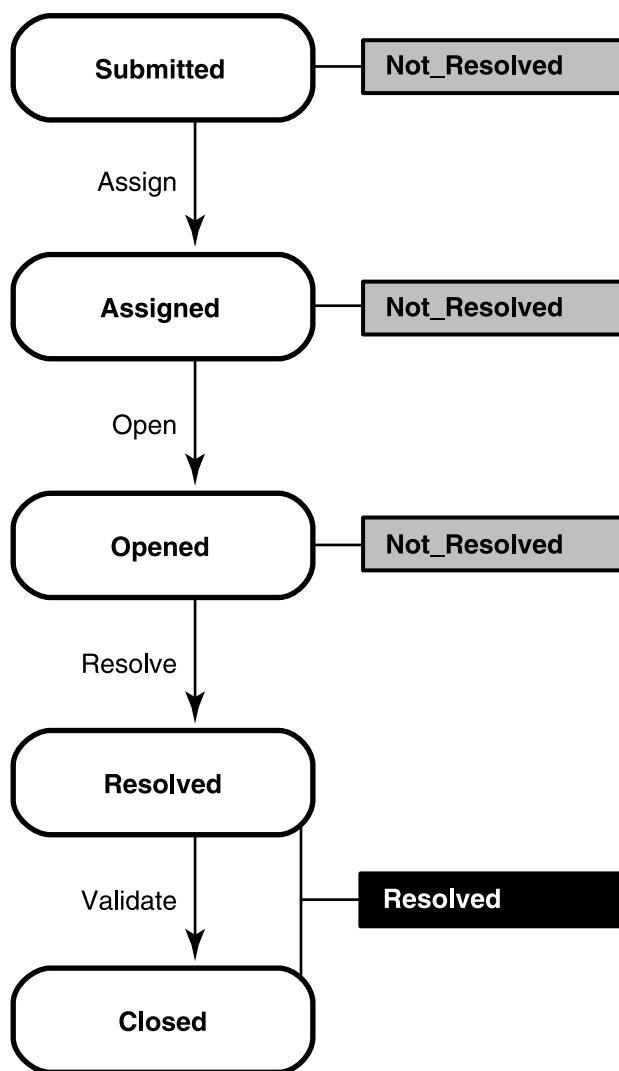
メモ： 状態タイプを使用するスキーマに新しい状態を追加する場合は、新しい状態を状態タイプにマッピングする必要があります。122 ページの「状態タイプのマッピング」を参照してください。

Resolution パッケージの状態タイプ モデル

次の表と図は、Resolution パッケージの状態タイプと有効な状態マッピングの例を示しています。

状態タイプ	説明
Not_Resolved	レコードが解決されていない状態タイプ
Resolved	レコードが解決されている状態タイプ

図 9 Resolution パッケージの状態タイプ モデル



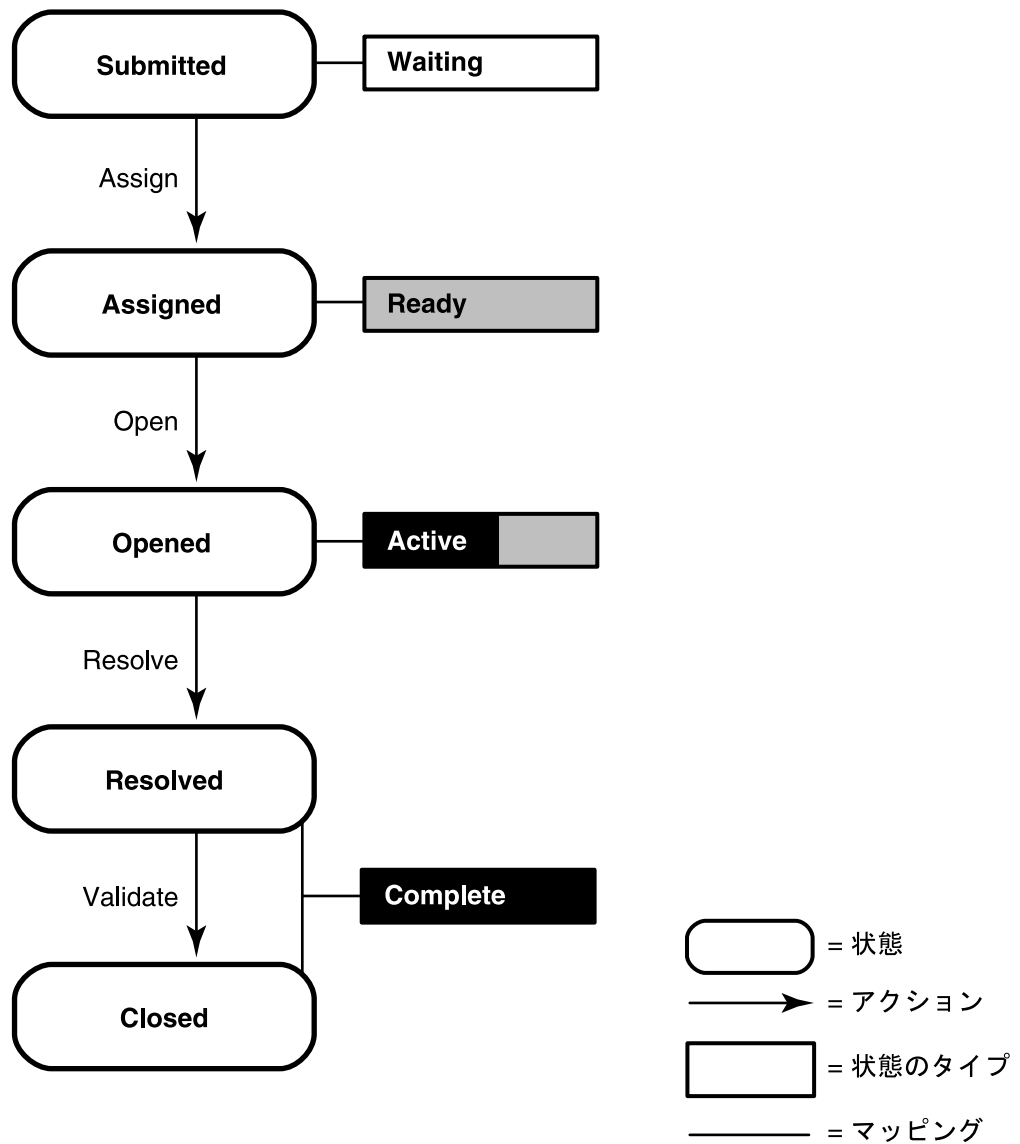
UnifiedChangeManagement パッケージの状態タイプ モデル

次の表と図は、UCM パッケージの状態タイプと有効な状態マッピングを示しています。状態のデフォルト アクションにより、状態タイプ モデル全体で完全な遷移が実行される必要があります。

詳細については、「付録 B ClearQuest 統合の追加」を参照してください。

状態タイプ	説明	UCM スキーマにマッピングされる状態
Waiting	レコードのアサイン/優先順位の割り当て/スケジュールが行われていない状態タイプ	Submitted Postponed
Ready	レコードがアサインされている状態タイプ。アサインされたユーザーの To Do List クエリーにレコードが表示され、作業する準備ができています。ucm_project フィールドを設定する場合もあれば、設定しない場合もあります。	Assigned
Active	ユーザーがレコードの処理を開始しており、レコードは ClearCase プロジェクトに関連付けられ、ClearCase のエレメント情報を含むことができます。	Opened
Complete	レコードは以下のいずれかです。 1) レコードの処理が完了し、レコードは ClearCase プロジェクトに関連付けられ、ClearCase のエレメント情報を含むことができます。または処理されず、放棄されました。 または 2) 延期またはクローズされました。ClearCase プロジェクトと関連付けられたか、関連付けられていないかのどちらかです。	Resolved Closed Duplicate

図 10 Enhancement レコード タイプの状態タイプ モデル



Rational ClearQuest をソフトウェア構成管理ツールと統合して、完全な変更管理システムを構築できます。この付録では、ClearQuest で利用できる統合について説明します。

この章の内容は、次のとおりです。

- 305 ページの「ClearQuest 統合の概要」
- 309 ページの「スキーマのパッケージの参照」
- 309 ページの「統合のテスト」
- 310 ページの「自主統合の追加」
- 311 ページの「依存先統合の追加」
- 323 ページの「パッケージのアップグレードの適用」

メモ: 各項の操作手順の説明は、ClearQuest スキーマの操作方法について理解していることを前提にしています。「第 4 章 ClearQuest スキーマの操作」を参照してください。

ClearQuest 統合の概要

ClearQuest をほかのソフトウェアと統合するには、既存のスキーマに ClearQuest パッケージを追加します。一部の ClearQuest 製品は自主統合、つまり、該当するパッケージの追加のみを必要とする統合です。また、一部の製品は依存先統合、つまり、1 つまたは複数のパッケージを特定の順序で追加する必要がある上に、ClearQuest への追加設定も必要になる統合です。

ClearQuest 統合が Web クライアントに適用されないということに注意してください。つまり、統合パッケージによって Windows または UNIX のどちらかで起動されている ClearQuest クライアントに追加されたフィールド、フォーム、レポート、スクリプト、その他の機能は、Web クライアントを経由して使用することはできないということです。

メモ: この付録での説明では、新規の統合を追加していることと、必要なパッケージがスキーマにまだ存在しないことを前提にしています。統合またはパッケージをアップグレードする必要がある場合には、323 ページの「パッケージのアップグレードの適用」を参照してください。

警告: パッケージの追加は、慎重に計画してください。パッケージは、スキーマに追加すると削除することはできません。このような場合は、パッケージが存在するすべてのスキーマバージョンを削除してください。スキーマバージョンを削除できるのは、ユーザー データベースに適用していない場合のみです。

自主統合

以下の統合は自主統合です。すべての自主統合に対して同じインストール処理を使用できます。310 ページの「自主統合の追加」を参照してください。

- **Rational ClearCase と ClearQuest**
1 つまたは複数の ClearQuest 変更依頼を 1 つまたは複数の ClearCase バージョンに関連付けます。
- **Rational Suite ContentStudio と ClearQuest**
ClearQuest で内容の変更を登録、追跡することができます。
- **Rational PureCoverage と ClearQuest**
コード適用範囲データを ClearQuest データベースに登録し、それを追跡することができます。
- **Rational Purify と ClearQuest**
データを ClearQuest データベースに登録し、それを追跡することができます。
- **Rational Quantify と ClearQuest**
パフォーマンス データを ClearQuest データベースに登録し、それを追跡することができます。
- **電子メール システムと ClearQuest**
電子メールシステムを介して ClearQuest がユーザーと通信することを可能にします。
電子メール システム統合には、Rational E-Mail Reader の設定と Email notification パッケージの追加が含まれます。「第 12 章 ClearQuest 電子メールの管理」を参照してください。

依存先統合

以下の統合は依存先統合です。依存先統合の詳細については、310 ページの「自主統合の追加」を参照してください。

- **Rational Administrator と ClearQuest**
Rational プロジェクトを ClearQuest データベースに関連付けます。312 ページの「Rational Administrator 統合の追加」を参照してください。
- **Rational ClearQuest Project Tracker と ClearQuest**
2 つのシステム間でプロジェクト データを交換することができます。313 ページの「Rational ClearQuest Project Tracker 統合の追加」を参照してください。
- **Rational RequisitePro と ClearQuest**
RequisitePro 要求を ClearQuest レコードと関連付けます。
RequisitePro と ClearQuest の統合の追加方法の詳細については、『Rational Suite 管理ガイド』を参照してください。

- Rational TeamTest と ClearQuest

TeamTest を介して見つけた障害を ClearQuest データベースに登録し、それらを追跡することができます。314 ページの「Rational TeamTest 統合の追加」を参照してください。

- Rational 統一変更管理 (UCM) と ClearQuest

ClearCase UCM プロジェクトと UCM アクティビティを ClearQuest レコードにリンクします。315 ページの「Rational UCM 統合の追加」を参照してください。

- Microsoft Visual SourceSafe と ClearQuest

Visual SourceSafe 情報を ClearQuest レコードに関連付けます。321 ページの「Microsoft Visual SourceSafe 統合の追加」を参照してください。

ClearQuest 統合とコード ページ

以下に、ClearQuest 統合を使用する場合に注意すべき点をいくつか示します。これは、ClearQuest データベースのすべてのデータが同じコード ページのデータでなければならないために必要です。

- ClearQuest 統合を介して ClearQuest データベースにデータを入力するには、統合された製品が、データベース セットの ClearQuest データ コード ページと同じオペレーティング システム コード ページを使用するコンピュータで起動されている必要があります。ClearQuest は、異なるコード ページを使用する統合された製品からのすべてのデータの入力をブロックします。
- ClearQuest データ コード ページ値が ASCII の場合は、ClearQuest をほかの製品と統合できないことがあります。ClearQuest が統合する製品では、データ エントリが ASCII 文字のみに制限されていません。ASCII データ コード ページ値を使用する ClearQuest データベース セットでは、統合された製品からの ASCII 以外のすべての文字が拒否されます。

例

ここでは、コード ページの不一致により ClearQuest と Rational Administrator 間の統合が適切に機能しなくなる例を示します。

Rational ClearQuest が 1252 オペレーティング システム コード ページを使用するコンピュータにインストールされ、データベース セットの ClearQuest データ コード ページ値も 1252 に設定されています。

Rational Administrator クライアントが 932 (日本語) オペレーティング システム コード ページを使用するコンピュータにインストールされています。

ユーザーが Rational Administrator プロジェクトを作成します。ただし、そのユーザーは 932 コード ページを使用するコンピュータで作業しているため、プロジェクト名に日本語文字が含まれています。ユーザーは次に、Rational Administrator プロジェクトを ClearQuest データベースに関連付けようとしますが、エラーが発生します。これは、Rational Administrator プロジェクトの名前に含まれている日本語文字が、ClearQuest データベース セットのデータ コード ページ値である 1252 コード ページと異なることが原因です。

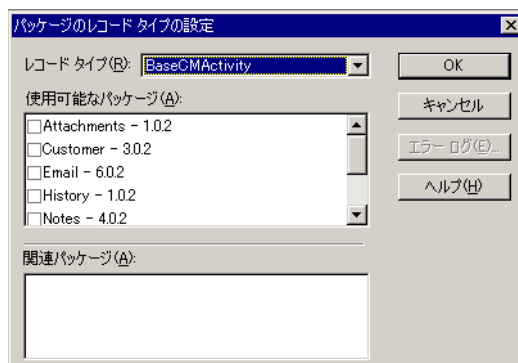
統合用のレコード タイプの有効化

パッケージには、スキーマに存在するレコード タイプを有効にするものがあります。これらのパッケージをインストールする場合は、パッケージで有効にするレコード タイプを選択するオプションが用意されています。パッケージを追加した後で新規のレコード タイプを追加する場合でも、パッケージ機能で新規のレコード タイプを有効にできます。

パッケージとパッケージが変更するレコード タイプのリストについては、289 ページの「ClearQuest のパッケージ」を参照してください。

新規のレコード タイプに対してパッケージ機能を有効にするには

- 1 ClearQuest Designer で、[ファイル] メニューの [スキーマを開く] をクリックし、スキーマを選択します。
- 2 スキーマが開いたら、[パッケージ] メニューの [パッケージのレコード タイプの設定] をクリックします。



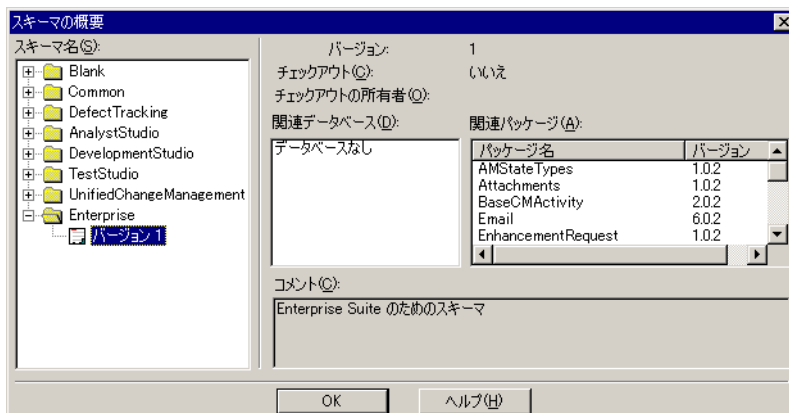
- 3 [パッケージのレコード タイプの設定] ダイアログ ボックスで、[レコード タイプ] リストからレコード タイプを選択します。
- 4 [使用可能なパッケージ] リストで、レコード タイプを有効にするパッケージをオンにします。

スキーマのパッケージの参照

多くのパッケージは、複数の統合によって使用されます。スキーマに統合を追加する前に、既に存在しているパッケージを確認する必要があります。

既にスキーマにあるパッケージのリストを参照するには

- 1 ClearQuest Designer で、[表示] メニューの [スキーマの概要] をクリックします。
- 2 [スキーマの概要] ダイアログ ボックスで、目的のスキーマ フォルダを開き、スキーマの最新バージョンを選択します。



- [関連データベース] リストに、スキーマに関連付けられているデータベースが表示されます。
- [関連パッケージ] リストに、スキーマに現在インストールされているすべてのパッケージバージョンが表示されます。

統合のテスト

スキーマにパッケージを追加し、スキーマをチェックインして、ユーザー データベースに適用すると、変更を元に戻すことはできません。実動データベースに対する修復不能な損傷を防止するには、実動環境を変更する前に、テスト環境で統合パッケージを追加することを強くお勧めします。

テスト統合を実行するときは、両方の実動環境、つまり、ClearQuest の環境と、ClearQuest と統合されているアプリケーションの環境を複製します。たとえば、ClearQuest と ClearCase の統合では、次のものが必要となります。

- テスト用 ClearQuest データベース
- テスト用 ClearQuest スキーマ
- テスト用 ClearCase VOB

詳細については、82 ページの「テスト データベースの作成」を参照してください。

自主統合の追加

自主統合に必要な処理は、次の表に示す適切なパッケージを追加することだけです。

統合	追加するパッケージ	追加のドキュメント
Rational ClearCase (ベース ClearCase)	ClearCase	ClearCase ヘルプ。目次で、[ClearCase ユーザー インターフェイス]、[ほかの製品との統合]、[ClearQuest とベース ClearCase との統合] の順にクリックします。 メモ: この項で説明する ClearCase 統合はベース ClearCase 統合です。事前定義されたポリシーはありません。ポリシーは、ClearCase 管理者が設定します。315 ページの「Rational UCM 統合の追加」で説明する UCM 統合では、事前定義された ClearCase ポリシーが自動的に設定されます。
Rational Purify	PQC	Rational Purify ヘルプ。「ClearQuest」を参照してください。
Rational Quantify	PQC	Rational Quantify ヘルプ。「ClearQuest」を参照してください。
Rational PureCoverage	PQC	Rational PureCoverage ヘルプ。「ClearQuest」を参照してください。
Rational Suite ContentStudio	ContentStudio	『Installing Rational Suite ContentStudio』

パッケージを追加するには、スーパー ユーザー権限またはスキーマ デザイナ権限を所有している必要があります。

パッケージが既にスキーマにインストールされているかどうかの確認については、309 ページの「スキーマのパッケージの参照」を参照してください。

可能であれば、テスト環境ですべての統合を追加してから、実動データベースに統合を追加してください。309 ページの「統合のテスト」を参照してください。

警告: パッケージの追加は、慎重に計画してください。パッケージは、スキーマに追加すると削除することはできません。このような場合は、パッケージが存在するすべてのスキーマバージョンを削除してください。スキーマバージョンを削除できるのは、ユーザー データベースに適用していない場合のみです。

自主統合を追加するには

- 1 ClearQuest Designer で、パッケージを追加するスキーマがチェックインされていることを確認します。スキーマをチェックインするには、[ファイル] メニューの [チェックイン] をクリックします。
- 2 [パッケージ] メニューの [パッケージ ウィザード] をクリックし、目的のパッケージをインストールします。詳細については、135 ページの「パッケージの適用」を参照してください。
- 3 スキーマの変更を確認します。詳細については、88 ページの「スキーマの変更の確認」を参照してください。
- 4 [ファイル] メニューの [チェックイン] をクリックし、スキーマの変更を保存します。詳細については、91 ページの「スキーマのチェックイン」を参照してください。
- 5 [データベース] メニューの [データベースのアップグレード] をクリックし、スキーマの変更をユーザー データベースに適用します。詳細については、92 ページの「ユーザー データベースのアップグレード」を参照してください。
- 6 必要に応じて、統合したアプリケーションを設定します。追加設定については、アプリケーションのマニュアルを参照してください。

依存先統合の追加

依存先統合には、複数のパッケージの追加が必要な場合があります。また、追加設定が必要な場合もあります。統合パッケージは、指定された順序でインストールする必要があります。

パッケージを追加するには、スーパー ユーザー権限またはスキーマ デザイナ権限を所有している必要があります。

可能であれば、統合を実動データベースに追加する前に、統合をテスト データベースに追加し、統合機能の理解を深めます。309 ページの「統合のテスト」を参照してください。

警告: パッケージの追加は、慎重に計画してください。パッケージは、スキーマに追加すると削除することはできません。このような場合は、パッケージが存在するすべてのスキーマバージョンを削除してください。スキーマバージョンを削除できるのは、ユーザー データベースに適用していない場合のみです。

依存先統合には、次のものがあります。

- Rational Administrator 統合の追加。
- Rational ClearQuest Project Tracker 統合の追加。

- Rational TeamTest 統合の追加。
- Rational UCM 統合の追加。
- Microsoft Visual SourceSafe 統合の追加。
- Rational RequisitePro。詳細については、『Rational Suite 管理ガイド』を参照してください。

Rational Administrator 統合の追加

Rational Administrator と統合すると、Rational プロジェクトは ClearQuest データベースに関連付けられます。Rational Administrator 統合には、次の処理が必要です。

- Repository パッケージの追加。
- スキーマの変更の保存。
- Rational Administrator の設定。

メモ: Repository パッケージが既にスキーマにあるかどうかの確認については、309 ページの「スキーマのパッケージの参照」を参照してください。Repository パッケージがスキーマに既にあり、かつ、新規のレコードタイプにパッケージを適用するだけの場合には、308 ページの「統合用のレコードタイプの有効化」を参照してください。

Repository パッケージの追加

- 1 ClearQuest Designer で、パッケージを追加するスキーマがチェックインされていることを確認します。スキーマをチェックインするには、[ファイル] メニューの [チェックイン] をクリックします。
- 2 [パッケージ] メニューの [パッケージ ウィザード] をクリックし、最新の Repository パッケージを追加します。詳細については、135 ページの「パッケージの適用」を参照してください。

スキーマの変更の保存

Repository パッケージを追加した後の手順は、次のとおりです。

- 1 スキーマの変更を確認します。詳細については、88 ページの「スキーマの変更の確認」を参照してください。
- 2 [ファイル] メニューの [チェックイン] をクリックし、スキーマの変更を保存します。詳細については、91 ページの「スキーマのチェックイン」を参照してください。
- 3 [データベース] メニューの [データベースのアップグレード] をクリックし、スキーマの変更をユーザー データベースに適用します。詳細については、92 ページの「ユーザー データベースのアップグレード」を参照してください。

Rational Administrator の設定

必要に応じて、Rational Administrator アプリケーションを設定します。追加設定については、Rational Administrator のヘルプを参照してください。

Rational ClearQuest Project Tracker 統合の追加

Project Tracker と統合すると、Project Tracker と ClearQuest との間でプロジェクト データを交換することができます。Project Tracker 統合には、次の処理が必要です。この処理は、説明する順序で実行してください。

- 1 AMBaseActivity パッケージの追加
- 2 AMWorkActivitySchedule パッケージの追加
- 3 スキーマの変更の保存
- 4 Rational ClearQuest Project Tracker の設定

警告: エラーの発生を防止するには、この順序でパッケージをインストールする必要があります。

メモ: AMBaseActivity パッケージと AMWorkActivitySchedule パッケージが既にスキーマにインストールされているかどうかの確認については、309 ページの「スキーマのパッケージの参照」を参照してください。これらのパッケージがスキーマに既にあり、かつ、新規のレコードタイプにパッケージを適用するだけの場合は、308 ページの「統合用のレコードタイプの有効化」を参照してください。

AMBaseActivity パッケージの追加

- 1 ClearQuest Designer で、パッケージを追加するスキーマがチェックインされていることを確認します。スキーマをチェックインするには、[ファイル] メニューの [チェックイン] をクリックします。
- 2 [パッケージ] メニューの [パッケージ ウィザード] をクリックし、追加する最新の AMBaseActivity パッケージを選択します。詳細については、135 ページの「パッケージの適用」を参照してください。
- 3 [ファイル] メニューの [チェックイン] をクリックし、スキーマをスキーマ リポジトリにチェックインします。

メモ: Project Tracker の設定時に、AMBaseActivity パッケージを BaseCmActivity レコードタイプに適用することはできません。BaseCmActivity は UCM レコードタイプです。AmBaseActivity パッケージは、Defect レコードタイプと Enhancement Request レコードタイプにのみ適用できます。

AMWorkActivitySchedule パッケージの追加

スキーマがチェックアウトされている状態で、[パッケージ] メニューの [パッケージ ウィザード] をクリックし、最新の AMWorkActivitySchedule パッケージを追加します。

スキーマの変更の保存

AMWorkActivitySchedule パッケージを追加した後の手順は、次のとおりです。

- 1 スキーマの変更を確認します。詳細については、88 ページの「スキーマの変更の確認」を参照してください。
- 2 [ファイル] メニューの [チェックイン] をクリックし、スキーマの変更を保存します。詳細については、91 ページの「スキーマのチェックイン」を参照してください。
- 3 [データベース] メニューの [データベースのアップグレード] をクリックし、スキーマの変更をユーザー データベースに適用します。詳細については、92 ページの「ユーザー データベースのアップグレード」を参照してください。

Rational ClearQuest Project Tracker の設定

必要に応じて、Project Tracker を設定します。プロジェクト プランと ClearQuest データベースやほかの統合タスクとのリンクの詳細については、『Rational ClearQuest Project Tracker ユーザーズ ガイド』を参照してください。

Rational TeamTest 統合の追加

Rational TeamTest との統合によって、TeamTest を介して確認した障害を登録したり、変更の追跡を簡素化することができます。Rational TeamTest 統合には、次の処理が必要です。この処理は、説明する順序で実行してください。

- 1 Repository パッケージの追加。
- 2 TeamTest パッケージの追加。
- 3 スキーマの変更の保存。
- 4 Rational TeamTest の設定。

警告: エラーの発生を防止するには、この順序でパッケージをインストールする必要があります。

メモ: Repository パッケージと TeamTest パッケージが既にスキーマにインストールされているかどうかの確認については、309 ページの「スキーマのパッケージの参照」を参照してください。これらのパッケージがスキーマに既にあり、かつ、新しいレコードタイプにパッケージを適用するだけの場合は、308 ページの「統合用のレコードタイプの有効化」を参照してください。

Repository パッケージの追加

- 1 ClearQuest Designer で、パッケージを追加するスキーマがチェックインされていることを確認します。スキーマをチェックインするには、[ファイル] メニューの [チェックイン] をクリックします。
- 2 [パッケージ] メニューの [パッケージ ウィザード] をクリックし、最新の **Repository** パッケージを追加します。詳細については、135 ページの「パッケージの適用」を参照してください。
- 3 [ファイル] メニューの [チェックイン] をクリックし、スキーマをスキーマ リポジトリにチェックインします。

TeamTest パッケージの追加

スキーマがチェックアウトされている状態で、[パッケージ] メニューの [パッケージ ウィザード] をクリックし、最新の **TeamTest** パッケージを追加します。

スキーマの変更の保存

TeamTest パッケージを追加した後の手順は、次のとおりです。

- 1 スキーマの変更を確認します。詳細については、88 ページの「スキーマの変更の確認」を参照してください。
- 2 [ファイル] メニューの [チェックイン] をクリックし、スキーマの変更を保存します。詳細については、91 ページの「スキーマのチェックイン」を参照してください。
- 3 [データベース] メニューの [データベースのアップグレード] をクリックし、スキーマの変更をユーザー データベースに適用します。詳細については、92 ページの「ユーザー データベースのアップグレード」を参照してください。

Rational TeamTest の設定

必要に応じて、TeamTest アプリケーションを設定します。設定の詳細については、Rational TeamTest のヘルプを参照してください。

Rational UCM 統合の追加

UCM (統一変更管理) 統合によって、ClearCase の UCM プロジェクトや UCM アクティビティは ClearQuest レコードにリンクされます。これは UCM-ClearCase 統合とも呼ばれます。

UCM 統合には、次のものがが必要です。

- UCM で使用可能な ClearQuest スキーマ
- ClearQuest を使用可能なプロジェクトを含む ClearCase 4.x

ClearQuest では、UCM をサポートする 2 つのスキーマが事前定義されています。UCM (統一変更管理) スキーマと Enterprise スキーマです。これらのスキーマのいずれかを使用するのが、UCM を実現する最も簡単な方法です。スキーマの使用法については、「第 4 章 ClearQuest スキーマの操作」を参照してください。

また、既存のスキーマに適切なパッケージを追加することで、そのスキーマに UCM サポートを追加することができます。ここでは、パッケージの追加による ClearQuest と UCM の統合について説明します。これらのパッケージは、各手順で説明する順序で追加する必要があります。

メモ: UCM 統合によって、ClearCase を併用することができますが、この統合で ClearCase パッケージを追加してはなりません。ClearCase パッケージはベース ClearCase 統合に限って使用されます。ベース ClearCase 統合では、事前定義された ClearCase ポリシーは設定されません。ベース ClearCase 統合の追加の詳細については、310 ページの「自主統合の追加」を参照してください。

Rational UCM とパッケージの統合には、次の処理が必要です。この処理は、説明する順序で実行してください。

- 1 AMStateTypes パッケージの追加。
- 2 UCM のデフォルト アクションの設定。
- 3 UCMPolicyScripts パッケージの追加。
- 4 UnifiedChangeManagement パッケージの追加。
- 5 BaseCMActivity パッケージ (オプション) の追加。
- 6 スキーマの変更の保存。
- 7 Rational UCM の設定。

警告: エラーの発生を防止するには、この順序でパッケージをインストールする必要があります。

メモ: AMStateType、UCMPolicyScripts、UnifiedChangeManagement、BaseCMActivity の各パッケージが既にスキーマにインストールされているかどうかの確認については、309 ページの「スキーマのパッケージの参照」を参照してください。これらのパッケージがスキーマに既にあり、かつ、新しいレコード タイプにパッケージを適用するだけの場合は、308 ページの「統合用のレコード タイプの有効化」を参照してください。

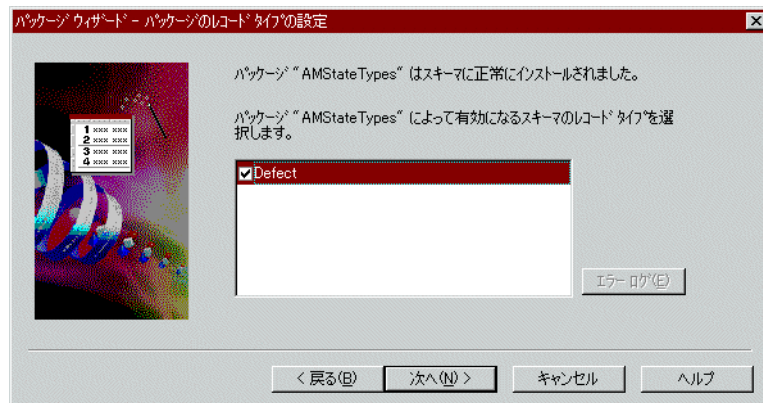
UCM 統合の設定と使用法の詳細については、『Rational ClearCase プロジェクト管理ガイド』を参照してください。

AMStateTypes パッケージの追加

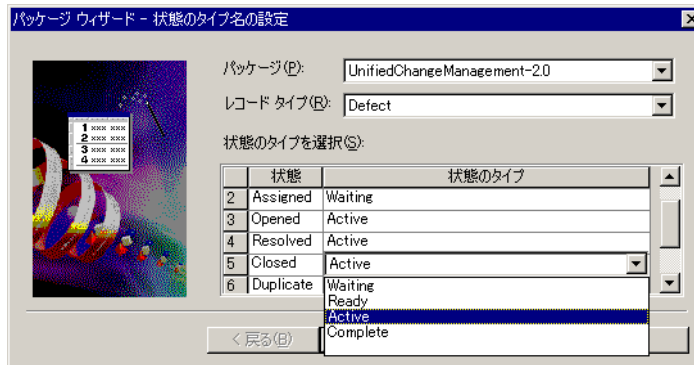
- 1 ClearQuest Designer で、パッケージを追加するスキーマがチェックインされていることを確認します。スキーマをチェックインするには、[ファイル] メニューの [チェックイン] をクリックします。
- 2 [パッケージ] メニューの [パッケージ ウィザード] をクリックし、最新の **AMStateType** パッケージを追加します。詳細については、135 ページの「パッケージの適用」を参照してください。

AMStateType パッケージでは、状態タイプをマッピングし、デフォルト アクションが未定義の場合には、デフォルト アクションを定義する必要があります。

- 3 UCM に対して有効にするレコード タイプを選択し、[次へ] をクリックして [状態のタイプ名の設定] ダイアログ ボックスを表示します。



- 4 [状態のタイプ名の設定] ダイアログ ボックスで、スキーマの状態を UCM の状態タイプにマッピングします。
 - レコード タイプを選択します。
 - レコード タイプの各状態に対して、[状態のタイプ] フィールドをクリックし、該当する UCM の状態タイプを選択します。状態タイプのマッピングの詳細については、122 ページの「状態タイプのマッピング」を参照してください。



- 5 有効にしたレコードタイプごとに、状態タイプのマッピングを繰り返します。[終了]をクリックします。

ClearQuest は自動的にスキーマを確認します。検証ウィンドウに、デフォルトアクションを設定する必要があることが表示されます。

UCM のデフォルトアクションの設定

スキーマの状態遷移マトリックスは、状態タイプ モデルとして、Waiting 状態タイプから Ready、Active、Complete に至る UnifiedChangeManagement パッケージのパスを少なくとも 1 つ備える必要があります。詳細については、121 ページの「状態遷移マトリックスの表示」と 302 ページの「UnifiedChangeManagement パッケージの状態タイプ モデル」を参照してください。

スキーマの各状態 (UCM の Complete 状態にマッピングされた状態を除く) に、UCM の状態タイプ モデルにおいて、その状態から次の状態に移行させるデフォルトアクションを割り当てる必要があります。詳細については、130 ページの「デフォルトアクションの使用法」を参照してください。

デフォルトアクションを割り当てるには

- 1 ClearQuest Designer で、[レコードタイプ] を展開し、UCM で有効にしたレコードタイプを展開します。次に、[状態遷移マトリックス] をダブルクリックします。
- 2 [状態遷移マトリックス] で、状態を右クリックし、[プロパティ] をクリックして、その状態の [プロパティ] ダイアログ ボックスを表示します。

UnifiedChangeManagement パッケージの追加

- 1 スキーマがチェックインされている状態で、[パッケージ] メニューの [パッケージ ウィザード] をクリックし、最新の UnifiedChangeManagement パッケージを追加します。
- 2 [ファイル] メニューの [チェックイン] をクリックし、スキーマをチェックインします。

BaseCMActivity パッケージ (オプション) の追加

BaseCMActivity パッケージは、スキーマに軽いアクティビティ レコード タイプを追加します。そのまま Defect レコード タイプの代わりに使用して、統一変更管理 (UCM) に対応したり、新規レコード タイプとして作成できます。このパッケージはオプションです。アクティビティ 追跡の詳細については、『Rational ClearQuest Project Tracker ユーザーズ ガイド』を参照してください。

- 1 スキーマがチェックインされている状態で、[パッケージ] メニューの [パッケージ ウィザード] をクリックし、最新の BaseCMActivity パッケージを追加します。
- 2 BaseCMActivity パッケージの状態タイプを編集します。詳細については、135 ページの「パッケージの適用」を参照してください。
- 3 [ファイル] メニューの [チェックイン] をクリックし、スキーマをチェックインします。

スキーマの変更の保存

最新の UCM パッケージをインストールした後は、次の処理を実行します。

- 1 スキーマの変更を確認します。詳細については、88 ページの「スキーマの変更の確認」を参照してください。
- 2 [ファイル] メニューの [チェックイン] をクリックし、スキーマの変更を保存します。詳細については、91 ページの「スキーマのチェックイン」を参照してください。
- 3 [データベース] メニューの [データベースのアップグレード] をクリックし、スキーマの変更をユーザー データベースに適用します。92 ページの「ユーザー データベースのアップグレード」を参照してください。

Rational UCM の設定

必要に応じて、UCM アプリケーションを設定します。追加設定の詳細については、『Rational ClearCase プロジェクト管理ガイド』を参照してください。

Microsoft Visual SourceSafe 統合の追加

Visual SourceSafe 統合は、選択したレコードタイプに Visual SourceSafe フィールドを追加し、レコードタイプフォームに Visual SourceSafe タブを追加します。

Microsoft Visual SourceSafe 統合に必要な処理は、次のとおりです。

- Visual SourceSafe パッケージの追加。
- スキーマの変更の保存。
- ClearQuest クライアントでのクエリーの作成。
- 各クライアント コンピュータの設定。

これらの手順を完了した後で統合を使用する方法については、Visual SourceSafe の ClearQuest ツールのヘルプを参照してください。

メモ: Visual SourceSafe パッケージが既にスキーマにインストールされているかどうかの確認については、309 ページの「スキーマのパッケージの参照」を参照してください。このパッケージがスキーマに既にあり、かつ、新しいレコードタイプにパッケージを適用するだけの場合は、308 ページの「統合用のレコードタイプの有効化」の説明を参照してください。

Visual SourceSafe パッケージの追加

メモ: Visual SourceSafe パッケージを適用できるのは、1 つのスキーマにつき 1 つのレコードタイプのみです。

- 1 ClearQuest Designer で、パッケージを追加するスキーマがチェックインされていることを確認します。スキーマをチェックインするには、[ファイル] メニューの [チェックイン] をクリックします。
- 2 [パッケージ] メニューの [パッケージウィザード] をクリックし、最新の Visual SourceSafe パッケージを追加します。135 ページの「パッケージの適用」を参照してください。

スキーマの変更の保存

Visual SourceSafe パッケージをインストールした後の手順は、次のとおりです。

- 1 スキーマの変更を確認します。詳細については、88 ページの「スキーマの変更の確認」を参照してください。
- 2 [ファイル] メニューの [チェックイン] をクリックし、スキーマの変更を保存します。91 ページの「スキーマのチェックイン」を参照してください。
- 3 [データベース] メニューの [データベースのアップグレード] をクリックし、スキーマの変更をユーザー データベースに適用します。92 ページの「ユーザー データベースのアップグレード」を参照してください。

ClearQuest クライアントでのクエリーの作成

Visual SourceSafe プロジェクトに関連付けるレコードを確認するためにユーザーが使用できるクエリーを作成する必要があります。

クエリーを作成するには

- 1 ClearQuest クライアントに共有フォルダ権限のあるユーザーとしてログインします。
- 2 クエリーを定義します。詳細については、ClearQuest クライアントのヘルプを参照してください。

各クライアント コンピュータの設定

メモ: Visual SourceSafe の ClearQuest ツール、cqvss.exe を実行する場合、入力した情報は、次の場所にある .ini ファイルに保存されます。

```
vss¥users¥<user_name directory>¥cqsspref.ini
```

各クライアント コンピュータで Visual SourceSafe 統合を使用するには

- 1 ClearQuest ホーム ディレクトリにある Visual SourceSafe の ClearQuest ツール (cqvss.exe) を起動します。cqvss.exe のコピーをデスクトップにドラッグすると、ショートカットを作成することができます。
- 2 ログインし、使用する Visual SourceSafe データベースを選択します。



- 3 [Visual SourceSafe の ClearQuest ツール] ダイアログ ボックスで、[設定ツール] をクリックします。
- 4 [Visual SourceSafe 統合] ダイアログ ボックスで、322 ページの「ClearQuest クライアントでのクエリーの作成」で作成した ClearQuest クエリーを選択し、[OK] をクリックします。

パッケージのアップグレードの適用

メモ: ClearQuest で RequisitePro 統合をアップグレードする場合は、『Rational Suite アップグレードガイド』の「ClearQuest のアップグレード」を参照してください。

既存のすべてのパッケージをアップグレードするには

- 1 スーパー ユーザー権限またはスキーマ デザイナ権限で ClearQuest Designer にログインします。
- 2 [スキーマを開く] ダイアログ ボックスが表示されたら、[キャンセル] をクリックします。
- 3 ClearQuest Designer で、[パッケージ] メニューの [インストール済みパッケージのアップグレード] をクリックします。
- 4 スキーマ リポジトリのリストからスキーマ名を選択し、[次へ] をクリックします。
- 5 適用する必要があるすべてのパッケージが一覧表示されます。続行するには、[アップグレード] をクリックします。
- 6 アップグレードが完了したら、[終了] をクリックします。

UCM 統合をリリース 2000.02.10 からアップグレードする場合は、手順 7 でアップグレードをテストする前に UCM 状態タイプをマッピングします。手順は次のとおりです。

- a ClearQuest Designer で、[パッケージ] メニューの [状態のタイプ名の設定] をクリックします。
- b パッケージ ウィザードで、[パッケージ] リストから [AMStateTypes] を選択し、[レコードタイプ] リストのデフォルト値をそのまま使用します。
- c 右の列に表示されている ClearQuest の各状態を、[状態のタイプ] のプルダウン メニューまたは左の列から、該当する状態のタイプにマッピングします。
- d UCM 統合に新しいレコード タイプを追加した場合は、[レコードタイプ] のスクロール ダウン メニューに表示されます。次のレコード タイプを選択し、レコードタイプごとにマッピング処理を繰り返します。

メモ: 状態タイプのマッピングの詳細については、122 ページの「状態タイプのマッピング」を参照してください。

- 7 ClearQuest Designer で、[ファイル] メニューの [作業内容のテスト] をクリックし、作業内容を検証します。選択したスキーマに関連付けられたテスト データベースを選択し、[OK] をクリックして処理を続行します。スキーマの変更に対するテストの詳細については、82 ページの「テスト データベースの作成」を参照してください。

- 8 テストが完了したら、エラーが発生していないことを確認し、[次へ]をクリックします。
メモ: アップグレード処理が失敗した場合は、ステータス ウィンドウにエラーが表示されます。
- 9 [ファイル] メニューの [チェックイン] をクリックして変更をチェックインし、[OK] をクリックします。

パッケージのアップグレード ステータスの表示

パッケージをアップグレードするとき、プロセスの完了に合わせてステータスが表示されます。一般的なステータスは、次のように表示されます。

次のパッケージは DefectTracking 用にアップグレードされます:

Email 1.0 はリビジョン 6.0 にアップグレードされます。

Notes 1.0 はリビジョン 3.0 にアップグレードされます。

Customer 1.0 はリビジョン 2.0 にアップグレードされます。

UnifiedChangeManagement 3.0 はリビジョン 4.0 にアップグレードされます。

パッケージ アップグレードに成功しました。ウィザードを終了するには [終了] をクリックしてください。すると、変更が参照できるようにスキーマが開きます。

フックの例で、スキーマにフックを追加する方法に関する一般的なアイデアを示します。読みやすくするため、この例にはエラー チェックが含まれていません。データベースにレコードをコミットする前に、**Validate API** の戻り値を確認してエラーがないことを確認してください。

フックは管理者権限で実行するので、フィールドの動作が読み取り専用でもこれらの例は機能します。

ここでは、以下のフックの例を示します。

フィールド フックの例

- 326 ページの「フィールド選択リスト フックの例」
- 327 ページの「依存先リストを作成するフック」
- 328 ページの「ユーザー情報を表示するフィールド選択リスト フック」
- 330 ページの「フィールド デフォルト値フックの例」
- 332 ページの「フィールド権限フックの例」
- 333 ページの「フィールド検証フックの例」
- 334 ページの「フィールド値変更フックの例」

アクション フックの例

- 335 ページの「アクション初期化フックの例」
- 336 ページの「フィールド値のアクション初期化フック」
- 337 ページの「親レコードの値を設定するアクション フック」
- 340 ページの「アクション アクセス制御フックの例」
- 342 ページの「アクション コミット フックの例」
- 344 ページの「アクション通知フックの例」
- 347 ページの「アクション検証フックの例」
- 350 ページの「レコード スクリプトの例」
- 351 ページの「グローバル スクリプトの例」
- 352 ページの「ClearQuest フック スクリプトでの CAL メソッドの使用例」
- 355 ページの「cqperl を使用した詳細レポートと自動化」

詳細については、『Rational ClearQuest API Reference』を参照してください。

フィールド フックの例

フィールド選択リスト フックの例

選択リスト フックを使用すると、ユーザーに対して選択リストを作成できます。ClearQuest は、選択リスト フックを呼び出すと、選択パラメータ内に **HookChoices** クラスのインスタンスを提供します。フック内のこのオブジェクトを使用して、リストに項目を追加したり、既存のリスト項目をソートしてください。選択リスト内の項目をソートしなければ、項目はリストに追加された順に表示されます。

以下の例は、各種オペレーティング システム名の内容を持つ選択リストの作成を示しています。

VBScript

```
Sub OS_type_ChoiceList(fieldname, choices)
    ' fieldname As String
    ' choices As Object
    ' entityDef = defect
    choices.AddItem("Solaris")
    choices.AddItem("Windows")
    choices.AddItem("HP/UX")
End Sub
```

Perl

```
sub OS_type_ChoiceList {
    my($fieldname) = @_ ;
    my @choices;
    # $fieldname as string scalar
    # @choices as string array
    # entityDef is Defect
    # use array operation to add items. Example:
    # push(@choices, "red", "green", "blue");
    push(@choices, "Solaris", "Windows", "HP/UX");
    return @choices;
}
```

依存先リストを作成するフック

次の例では、クライアント オペレーティング システムで使用する値が、サーバー オペレーティング システムでユーザーが選択する値に依存することを前提にしています。

- 1 server_os フィールドで、列挙リストの値を Windows NT と UNIX に設定して、選択リスト フックを作成します。

VBScript

```
choices.AddItem("NT")
choices.AddItem("Unix")
```

Perl

```
push(@choices, "NT", "Unix");

return @choices; #ClearQuest Designer provides this line of code
```

- 2 ユーザーが新規のメンバーをリストに追加できないようにするには、[リストに制限] オプションをオンにします。
- 3 server_os で新規の値が選択されたときに client_os で以前の値を消去するには、server_os の変更された値フックに次のコードを追加します。

VBScript

```
SetFieldValue "client_os", ""
```

Perl

```
$entity->SetFieldValue("client_os", "");
```

- 4 client_os フィールドで、選択リスト フックを作成します。

VBScript

```
dim server_os_choice
set server_os_choice = GetFieldValue("server_os")
select case server_os_choice.GetValue()
case "NT"
choices.AddItem ("Win95")
choices.AddItem ("NT")
choices.AddItem ("Web")
case "Unix"
choices.AddItem ("Web")
end select
```

Perl

```
$server_os_choice = $entity->GetFieldValue("server_os");  
$svalue = $server_os_choice->GetValue();  
if ($svalue eq "NT") {  
    push(@choices, "Win95", "NT", "Web");  
} elsif ($svalue eq "Unix") {  
    push(@choices, "CQWeb");  
}  
return @choices;  
  
#ClearQuest Designer provides this line of code
```

- 5 server_os フィールドが変更されたときには常に値が再計算されるように、client_os フックのプロパティで [選択リストを再計算] をオンにします。
- 6 リスト ボックス コントロールを使用して、client_os フィールドと server_os フィールドをフォームに追加します。

ユーザー情報を表示するフィールド選択リスト フック

このフックは、このデータベースへのアクセス権限を持つユーザー ログイン名を自動的に抽出し、抽出したユーザー ログイン名をこのフィールドの選択リストにロードします。

VBScript

```
Sub AssignedTo_ChoiceList(fieldname, choices)  
    ' fieldname As String  
    ' choices As Object  
    ' entityDef = Defect  
  
    Dim sessionObj  
    Dim queryObj  
    Dim filterObj  
    Dim resultSetObj  
  
    Set sessionObj = GetSession()  
    ' start building a query of the users  
    Set queryObj = sessionObj.BuildQuery("users")  
  
    ' have the query return the desired field of the user object(s)  
    queryObj.BuildField ("login_name")
```

```

' filter for members of group "MyGroup" (whatever group you want)
Set filterObj = queryObj.BuildFilterOperator(AD_BOOL_OP_AND)
filterObj.BuildFilter "groups", AD_COMP_OP_EQ, "MyGroup"
Set resultSetObj = sessionObj.BuildResultSet(queryObj)

' run it
resultSetObj.Execute

' add each value in the returned column to the choicelist
Do While resultSetObj.MoveNext = AD_SUCCESS
    choices.AddItem resultSetObj.GetColumnValue(1)
Loop
End Sub

```

Perl

```

sub AssignedTo_ChoiceList {
    my($fieldname) = @_;
    my @choices;
    # $fieldname as string scalar
    # @choices as string array
    # record type name is Defect
    # field name is myuser
    # use array operation to add items. Example:
    # push(@choices, "red", "green", "blue");

    my $session = $entity->GetSession();
    # start building a query of the users
    my $queryDefObj = $session->BuildQuery("users");

    # have the query return the desired field of the user object(s)
    $queryDefObj->BuildField("login_name");

    # filter for members of group "MyGroup" (whatever group you want)
    my $filterOp = $queryDefObj->BuildFilterOperator(
        $CQPerlExt::CQ_BOOL_OP_AND);
    my @array = ("MyGroup");
    $filterOp->BuildFilter("groups", $CQPerlExt::CQ_COMP_OP_EQ,
        ¥@array);
    my $resultSetObj = $session->BuildResultSet($queryDefObj);
}

```

```

# run it
$resultSetObj->Execute();

# add each value in the returned column to the choicelist
while ($resultSetObj->MoveNext() == $CQPerlExt::CQ_SUCCESS) {
    push(@choices,$resultSetObj->GetColumnValue(1));
}
return @choices;
}

```

フィールド デフォルト値フックの例

ユーザーが新規レコードを作成すると、ClearQuest はそのレコードを作成し、そのフィールドには適切なデフォルト値を初期設定します。特定のフィールドに関連付けられたデフォルト値フックがある場合、ClearQuest はフックを実行してフィールドの値を設定します。フィールドに関連付けられたフックが存在しない場合、ClearQuest はフィールドの型に適したデフォルト値を割り当てます。たとえば、整数フィールドは 0、文字列フィールドは空文字列、リストフィールドは空のリストに設定されます。ただし、ClearQuest では日付フィールドはデフォルト値に初期化されないため、日付フィールドには常にデフォルト値フックを指定してください。

メモ: フックは管理者権限で実行するので、フィールドの動作が読み取り専用でもこれらの例は機能します。

例 1

この例では、日付フィールドを現在の日付と時間に初期化する方法を示します。

VBScript

```

Sub submit_date_DefaultValue(fieldname)
    ' fieldname As String
    ' entityDef = defect
    call SetFieldValue(fieldname, now)
End Sub

```

Perl

```

# Define a function for the current timestamp
sub GetCurrentDate {
    my ($sec, $min, $hour, $mday, $mon, $year, $wday, $yday,
        $time)=localtime();
    if ($year < 100) {

```

```

        $year = 19 . $year;
    } else {
        $year = 2000 + $year - 100;
    }
    return sprintf("%2.2d/%2.2d/%4s %2.2d:%2.2d:%2.2d", $mon + 1,
$mday,
        $year, $hour, $min, $sec);
}
# Define a routine to call the timestamp function
sub Submit_Date_DefaultValue {
    my($fieldname) = @_;
    # $fieldname as a string scalar and entityDef is Defect
    $entity->SetFieldValue($fieldname, GetCurrentDate());
}

```

例 2

この例では、レコードを登録しているユーザーのログイン名の `submitter` フィールドを初期化します。

VBScript

```

Sub submitter_DefaultValue(fieldname)
    ' fieldname As String
    ' entityDef = swbug
    SetFieldValue fieldname, GetSession().GetUserLoginName()
End Sub

```

Perl

```

sub Submitter_DefaultValue {
    my($fieldname) = @_;
    # $fieldname as a string scalar
    # entityDef is Defect
    my $session;
    my $username;
    $session = $entity->GetSession();
    $username = $session->GetUserLoginName();
    $entity->SetFieldValue($fieldname, $username);
}

```

フィールド権限フックの例

権限フックを使用すると、実行時のフィールドの動作を定義できます。フィールドの動作を定義するには、通常 **ClearQuest Designer** 内の動作グリッドを使用します。動作グリッド内に入力した値は、ユーザー グループのメンバー全員に等しく適用されます。権限フックを使用すると、フィールドの動作を詳細に指定できます。この例では、現在のユーザーが **managers** グループに属していて **engineering** グループには属していない場合、このフックはそのフィールドをオプションにします。ユーザーがどのグループにも属していない場合、フックは失敗します。

メモ: フックは管理者権限で実行するので、フィールドの動作が読み取り専用でもこの例は機能します。

VBScript

```
Function field1_Permission(fieldname, username)
    ' fieldname As String
    ' username As String
    ' field_Permission As Long
    ' entityDef = defect

    ' Assign the default return value
    set curSession = GetSession
    userGroups = curSession.GetUserGroups()
    for each group in userGroups
        if group = "managers" And group <> "engineers" Then
            field1_Permission = AD_OPTIONAL
        End If
    Next
End Function
```

Perl

```
sub field1_Permission {
    my($fieldname, $username) = @_ ;
    my $result;
    # $fieldname as string scalar
    # $username as string scalar
    # $result as long scalar
    # entityDef is Defect

    # Assign the default return value
    $curSession = $entity->GetSession();
    $userGroups = $curSession->GetUserGroups();
```

```

foreach $group (@$userGroups) {
    if ($group eq "managers" && $group ne "engineers") {
        $result = $CQPerlExt::CQ_OPTIONAL;
    }
}
return $result;
}

```

フィールド検証フックの例

検証フックを使用すると、フィールドに適切な値が設定されているかどうかを照合することができます。**ClearQuest** は、あらかじめ定義した時期に検証フックを呼び出し、フィールドの内容が有効であることを確認します。レコードに無効な値が設定されたフィールドがあると、エラーを修正しない限り、**ClearQuest** はそのレコードをデータベースにコミットしません。

アクション検証フックを使用してレコード全体を確認するのとは対照的に、フィールド検証フックを使用して個々のフィールドを確認することの利点は、提供されたフィールド値が無効な場合に直ちにユーザーに通知されることです。

フィールド検証フックは、文字列値を返す関数として記述します。関数の戻り値は、エラーメッセージと見なされます。戻り値が空の文字列の場合、**ClearQuest** はフィールド値を有効であると見なします。

以下は、フィールドに入力された文字数が 10 文字未満になっている場合には、検証フックはエントリを拒否し、ユーザーに 10 文字以上の文字を入力するよう指示する例です。

VBScript

```

Function word_Validation(fieldname)
    ' fieldname As String
    ' word_Validation As String
    ' entityDef = puzzle_words

    Dim val
    val = GetFieldValue(fieldname).GetValue()
    If Len(val) < 10 Then
        word_Validation = "All words must be at least 10 letters long"
    End If
End Function

```

Perl

```
sub word_validation {
    my($fieldname) = @_ ;
    # $fieldname as string scalar
    # $result as string scalar
    # $entityDef = puzzle_words
    my($value);

    $value = $entity->GetFieldValue($fieldname)->GetValue();
    if (length ($value) < 10) {
        $result = "All words must be at least 10 letters long";
    }
    return $result;
}
```

フィールド値変更フックの例

値変更フックを使用すると、フィールド内の値を変更した後に、フィールドを同期したり、ほかのタスクを実行することができます。

以下の例の場合、フックは、現在のフィールドに保存したオペレーティング システム名を確認します。オペ載[ティング システム名の確認が済むと、フックは、オペレーティング システムに従ってバージョン番号を **OS_version** フィールドに割り当てます。現在のフィールドが未設定で、オペレーティング システム名が設定されていないと、このフックは対応するバージョン番号を設定しません。

VBScript

```
Sub OS_type_ValueChanged(fieldname)
    ' fieldname As String
    value = GetFieldValue(fieldname).GetValue()

    If value = "solaris" Then
        SetFieldValue "OS_version", "7.x"
    ElseIf value = "windows" Then
        SetFieldValue "OS_version", "95"
    ElseIf value = "hpux" Then
        SetFieldValue "OS_version", "10.x"
    End If
End Sub
```

Perl

```
sub OS_type_ValueChanged {  
    my($fieldname) = @_;  
    my($value);  
    $value = $entity->GetFieldValue($fieldname)->GetValue();  
  
    if ($value eq "solaris") {  
        $entity->SetFieldValue("OS_version", "7.x");  
    } elsif ($value eq "windows") {  
        $entity->SetFieldValue("OS_version", "95");  
    } elsif ($value eq "hpux") {  
        $entity->SetFieldValue("OS_version", "10.x");  
    }  
}
```

アクション フックの例

アクション初期化フックの例

初期化フックを使用すると、アクションの先頭で複雑な初期設定を行うことができます。このフックを使用して、フィールドをリセットしたり、アクションのタイプに基づいてフィールドに別の値を割り当てることができます。

以下の例は、ユーザーが別のユーザーに障害を再度アサインする際に実行されるフックのデモを示します。このフックは、**Reassign** アクションの先頭で、**action_reason** フィールドの内容を消去します。このフィールドの動作を **Mandatory** に設定した場合、ユーザーは、アクションを再度割り当てる理由を入力する必要があります。

VBScript

```
Sub swbug_Initialization(actionname, actiontype)  
    ' actionname As String  
    ' actiontype As Long  
    ' action = reassign  
    ' Empty the string at the beginning of the action  
    SetFieldValue "action_reason", ""  
End Sub
```

Perl

```
sub swsub_initialization {
    my($actionname, $actiontype) = @_ ;
    # $actionname as string scalar
    # $actiontype as long scalar
    # action is reassign
    # do any setup for the action here
    # Empty the string at the beginning of the action
    $entity->SetFieldValue("action_reason", "");
}
```

メモ: スキーマが各アクションの開始理由を入力することをユーザーに要求する場合は、DEFAULT_VALUE フックを使用して、各アクションの先頭で `action_reason` フィールドを消去することができます。上の例では、フィールドの消去が、**Reassign** アクションに対してのみ必要になります。したがって、アクション初期化フックを使用するのが適しています。

フィールド値のアクション初期化フック

この例では、ユーザーが **Submit** アクションを呼び出すと、アクションフックによって `problem_description` フィールドが初期化され、レコードを登録したユーザーのログイン名がフィールドに設定されます。これにより、元の変更依頼を作成したユーザーを識別できます。

VBScript

```
Dim session
Dim loginName

' Get the session
set session = GetSession

' Get the logon name
loginName = session.GetUserLoginName
SetFieldValue "problem_description", loginName
```

Perl

```
# session is preset for Perl. No GetSession is required.
# Get the logon name
$loginName = $session->GetUserLoginName();
$entity->SetFieldValue("problem_description", $loginName);
```

親レコードの値を設定するアクション フック

次のアクション フックと親/子リンク関係を組み合わせで使用し、1 つの親レコードとその子レコードの間に、状態を基準にした関係を作成します。このフックを使用すると、親レコードを次の状態に自動的に移動できます (すべての子レコードがその状態である場合)。114 ページの「フィールドを使用したレコードのリンク」を参照してください。

メモ: このコードでは、所定のフィールド名とレコードタイプを前提にしています。このコードをカット アンド ペーストして使用する場合には、コードを部分的に変更する必要があります。

VBScript

```
Dim SessionObj
Dim ParentID 'Dimension variables that hold objects
Dim ParentObj
Dim ChildRefList
Dim ChildArray
Dim ChildID
Dim DefectChildEntityObj
Dim StateStatus
Dim SameState
Dim CurrentState
Dim RetValue
Dim ActionJustPerformed

set SessionObj = GetSession
ThisID = GetDisplayName
ActionJustPerformed = GetActionName
SessionObj.OutputDebugString "action is: "& ActionJustPerformed &
vbCrLf
StateStatus = ""
SameState = 0

SessionObj.OutputDebugString "current db id is: " & ThisID & vbCrLf

ParentID = GetFieldValue("parent").GetValue()
SessionObj.OutputDebugString "parent id is: " & ParentID & vbCrLf

if ParentID <> "" then
    set ParentObj=SessionObj.GetEntity("defect", parent_id)
    ChildRefList=ParentObj.GetFieldValue("children").GetValue
    ChildArray=split (ChildRefList, vbCrLf)

    For Each ChildID In ChildArray
```

```

        set DefectChildEntityObj=SessionObj.GetEntity("Defect", ChildID)
        CurrentState=DefectChildEntityObj.GetFieldValue ("State").GetValue

        SessionObj.OutputDebugString "StateStatus is: " & StateStatus &
vbCrLf
        SessionObj.OutputDebugString "CurrentState is: " & CurrentState
&vbCrLf
        SessionObj.OutputDebugString "SameState is: " & SameState &
vbCrLf
        if StateStatus = "" then
            StateStatus = CurrentState
            SameState = 1
            SessionObj.OutputDebugString "coming to statestatus is null" &
vbCrLf

        elseif StateStatus = CurrentState then
            SessionObj.OutputDebugString "coming to same state" & vbCrLf
            SameState = 1

        else
            SessionObj.OutputDebugString "states are different" & vbCrLf
            SameState = 0
        end if
    Next

    if SameState = 1 then
        SessionObj.OutputDebugString "samestate=1, setting parent state"
            & vbCrLf
        SessionObj.EditEntity ParentObj, ActionJustPerformed
        status = ParentObj.Validate

        if (status <> "") then
            SessionObj.OutputDebugString "error when updating parent state: "_
            & status & vbCrLf
            ParentObj.Revert
            exit sub
        end if
        ParentObj.Commit
    end if
end if

```

Perl

```
$SessionObj=$Entity->GetSession();
$ThisID=$Entity->GetDisplayName();
$ActionJustPerformed=$Entity->GetActionName();
$ParentID=$Entity->GetFieldValue("parent1")->GetValue();
$StateStatus="";
$SameState=0;

# ClearQuest has a message monitor to display
# ClearQuest messages and your messages
$SessionObj->OutputDebugString ("perl current db id is: $ThisID¥n");
$SessionObj->OutputDebugString ("perl parent id is: $parent_id¥n");

if ($ParentID ne "") {
    $ParentObj = $SessionObj->GetEntity("defect", $ParentID);
    $ChildRefList=$ParentObj->GetFieldValue("children")->GetValue();
    $SessionObj->OutputDebugString ("children are: $ChildRefList¥n");
    @ChildArray = split (/¥n/, $ChildRefList);

    foreach $ChildID (@ChildArray) {

        $DefectChildEntityObj = $SessionObj->GetEntity("defect",
$ChildID);
        $CurrentState=$DefectChildEntityObj->GetFieldValue("State")->
            GetValue();

        $SessionObj->OutputDebugString("perl StateStatus is:
$StateStatus¥n");
        $SessionObj->OutputDebugString("perl Current Status is:
            $CurrentStatus¥n");
        $SessionObj->OutputDebugString("perl SameState is:
$SameState¥n");

        if ($StateStatus eq "") {
            $StateStatus = $CurrentState;
            $SameState = 1;
            $SessionObj->OutputDebugString("coming to statestatus is
null¥n");
        } elseif ($StateStatus eq $CurrentState) {
            $SessionObj->OutputDebugString ("coming to same state¥n");
        }
    }
}
```

```

        $SameState = 1;

    } else {
        $SessionObj->OutputDebugString("states are different¥n");
        $SameState = 0;
    }    #nested if statements
} #End foreach Loop

if ($SameState == 1) {
    $SessionObj->OutputDebugString("samestate = 1, setting parent
        state ¥n");
    $SessionObj->EditEntity($ParentObj, $ActionJustPerformed);
    $status = $ParentObj->Validate();

    if ($status ne "") {
        $SessionObj->OutputDebugString ("error when updating parent
state:
        $status¥n");
        $ParentObj->Revert();
        return -1;    # Exit
    }
    $ParentObj->Commit();
}    #end if for ($SameState == 1)
}    #end if for ($ParentID ne "")

```

アクション アクセス制御フックの例

アクセス制御フックを使用すると、指定した条件セットに基づいて、特定のアクションへのアクセスを制御できます。**ClearQuest Designer** では、[ユーザー グループ] のフックの種類を選択して、特定のユーザー グループにアクションを制御したり、[すべてのユーザー] を選択して、ユーザー全員にアクションへのアクセス権限を与えることができます。また、[Scripts] オプションを選択して、アクセスを決定する VBScript または Perl フックを記述することもできます。

次の例は、**Pat** という名前の 1 人のユーザーのみにアクセスを制限する方法を示します。

VBScript

```
Function swbug_AccessControl(actionname, actiontype, username)
    ' actionname As String
    ' actiontype As Long
    ' username As String
    ' swbug_AccessControl As Boolean
    ' action = close

    Dim is_ok
    ' Test whether the current user has the privilege to close this bug
    If username = "Pat" Then
        is_ok = TRUE
    Else
        is_ok = FALSE
    End If
    swbug_AccessControl = is_ok

End Function
```

Perl

```
sub swbug_AccessControl {
    my($actionname, $actiontype, $username) = @_;
    my $result;
    # $actionname string scalar, $actiontype as long scalar
    # $username as string scalar, # action is Close
    # return TRUE if the user has permission to perform this action

    if ($username eq "Pat") {
        $result = 1;
    } else {
        $result = 0;
    }
    return $result;
}
```

アクションコミットフックの例

コミットフックを使用すると、レコードをデータベースにコミットする前に、さらにアクションを実行できます。

以下の例では、障害に重複 (dups) があるかどうかを調べます。元の障害に **tested** とマークした場合、フックは、重複に **dupdone** とマークし、それらの重複が修正されているかどうかを確認するために再度評価が必要であることを示します。これらの更新データの 1 つをコミットする際にエラーが発生した場合は、このフックが属する更新を含めて、すべてのデータベース トランザクションがロールバックされます。

VBScript

```
Sub swbug_Commit(actionname, actiontype)
    ' actionname As String
    ' actiontype As Long
    ' action = tested
    Dim dups    ' Array of all direct duplicates of this defect
    Dim dupsvar ' Variant containing link to a duplicate
    Dim dupsobj ' The same link, but as an object rather than a variant
    Dim record  ' The record extracted from the link
    Dim session
    Dim parent_id ' The display name of this defect

    ' Make an API to call to see if this record has duplicates
    If HasDuplicates() Then
        Set session = GetSession
        dups = GetDuplicates
        parent_id = GetDisplayName
        For Each dupvar In dups
            Set dupobj = dupvar
            Set entity = dupobj.GetChildEntity
            session.EditEntity entity, "dupdone"
            entity.SetFieldValue "action_reason", "Original " & parent_id &
                " is tested"
            entity.Validate
            entity.Commit
        Next
    End If
End Sub
```

Perl

```
sub swbug_Commit {
    my($actionname, $actiontype) = @_ ;
    # $actionname As string scalar
    # $actiontype as long scalar
    # action is Submit
    # This hook is fired during the "commit" step of an
    # entity update. It is the appropriate place to put an
    # activity which should be bundled into the same
    # transaction as the commit, such as subactions
    # or updates of external data storage.

    my ($dups, # Array of all direct duplicates of this defect
        $record, # The record extracted from the link
        $parent_id, # The display name of this defect
        $session,
        $locEntity,
        $dupobj
    );
    # Make an API to call to see if this record has duplicates
    if ($entity->HasDuplicates()) {
        $session = $entity->GetSession();
        $dups = $entity->GetDuplicates();
        $parent_id = $entity->GetDisplayName();
        my $count = $dups->Count();
        my $i = 0;
        for (i=0;$i<$count;$i++){
            $dupobj = $dups->Item($i);
            $locEntity = $dupobj->GetChildEntity();
            $session->EditEntity($locEntity, "dupdone");
            $locEntity->SetFieldValue("action_reason", "Original "
                . $parent_id . " is tested");
            $locEntity->Validate();
            $locEntity->Commit();
        }
    }
}
```

アクション通知フックの例

通知フックを使用すると、データベースに変更セットをコミットした後に、さらにアクションを起動できます。1 人のユーザーまたは複数のユーザーに電子メールの通知を送信したり、ほかの関連レコードを修正することができます。(また、電子メール メッセージを送信するための電子メール ルールを作成することもできます。詳細については、246 ページの「電子メール ルールの作成」を参照してください。)

メモ: 電子メール通知フックを変更するたびに、Rational ClearQuest 電子メール サービスを停止し、再開する必要があります。

以下の例では、修正した障害のフィールドごとにダイアログ ボックスが表示されています。また、通知フックを使用して、電子メール メッセージを生成し、適切な配信リストに送信することもできます。

メモ: フックから自分自身で開始されたアクションは、フック スクリプトでセッション変数 CQHookExecute を 1 という値に設定しない限り、通知を送信しません。この変数のデータ型は、VBScript では Long、Perl では long scalar です。

VBScript

```
Sub swbug_Notification(actionname, actiontype)
    ' actionname As String
    ' actiontype As Long
    ' action = modify
    ' Note: don't use MsgBox for web-based databases
    MsgBox "Modify action completed, notification hook started"

    fieldnames = GetFieldNames
    If IsArray(fieldnames) Then
        I = LBound(fieldnames)
        limit = UBound(fieldnames) + 1

        ' Get three kinds of values
        Do While I < limit
            onename = fieldnames(I)
            Set oldinfo = GetFieldOriginalValue(onename)
            Set newinfo = GetFieldValue(onename)
            oldstat = oldinfo.GetValueStatus
            If oldstat = AD_HAS_NO_VALUE Then
                oldempty = True
            End If
            I = I + 1
        Loop
    End If
End Sub
```

```

Else
    oldempty = False
    oldval = oldinfo.GetValue
End If

newstat = newinfo.GetValueStatus
If newstat = AD_HAS_NO_VALUE Then
    newempty = True
Else
    newempty = False
    newval = newinfo.GetValue
End If

' Compare the values
If oldstat = AD_VALUE_UNAVAILABLE Then
    MsgBox "Field " & onename & ": original value unknown"
Else
    If newempty And Not oldempty Then
        MsgBox "Field " & onename & " had its value deleted"
    ElseIf oldempty And Not newempty Then
        MsgBox "Field " & onename & " now= " & newval
    ElseIf oldval <> newval Then
        MsgBox "Field " & onename & " was= " & oldval
        MsgBox "Field " & onename & " now= " & newval
    Else
        MsgBox "Field " & onename & " is unchanged"
    End If
End If

I = I + 1
Loop
End If
MsgBox "Modify action and notification hook completed"
End Sub

```

Perl

```
sub swsub_Notification {
    my($actionname, $actiontype) = @_;
    # $actionname as string scalar
    # $actiontype as long scalar
    # action is Submit
    # Post-commit notifications about actions may be handled here

    my ($fieldnames,
        $session,
        $fieldname,
        $oldinfo,
        $newinfo,
        $newstat,
        $oldstat,
        $oldval,
        $oldempty,
    );

    $session = $entity->GetSession();
    $fieldnames = $entity->GetFieldNames();

    # Get three kinds of values
    foreach $fieldname (@$fieldnames) {
        $oldinfo = $entity->GetFieldOriginalValue($fieldname);
        $newinfo = $entity->GetFieldValue($fieldname);
        $oldstat = $oldinfo->GetValueStatus();
        if ($oldstat eq CQPerlExt::CQ_HAS_NO_VALUE) {
            $oldempty = 1;
        } else {
            $oldempty = 0;
            $oldval = $oldinfo->GetValue();
        }
        $newstat = $newinfo->GetValueStatus();
        if ($newstat eq CQPerlExt::CQ_HAS_NO_VALUE) {
            $newempty = 1;
        } else {
            $newempty = 0;
            $newval = $oldinfo->GetValue();
        }
    }
}
```

```

    }

    # Compare the values
    if ($oldstat eq CQPerlExt::CQ_VALUE_UNAVAILABLE) {
        $session->OutputDebugString("Field " . $fieldname . ":
            original value unknown¥n");
    } else {
        if ($newempty && !$oldempty) {
            $session->OutputDebugString ("Field " & $fieldname .
                " had its value deleted¥n");
        } elsif ($oldempty && !$newempty) {
            $session->OutputDebugString ("Field " . $fieldname . " now
= " . $newval. "¥n");
        } elsif ($oldval != $newval) {
            $session->OutputDebugString ("Field " . $fieldname . "
was = " . $oldval. "¥n");
            $session->OutputDebugString ("Field " . $fieldname . " now
= " . $newval. "¥n");
        } else {
            $session->OutputDebugString ("Field " . $fieldname . " is
unchanged¥n");
        }
    }
}

$session->OutputDebugString ("Modify action & notification hook
completed¥n");
}

```

アクション検証フックの例

アクション検証フックを使用すると、関連フィールド値のセットの検証など、フィールドレベルでの検証が困難な条件を調べることができます。フィールドが修正された直後に、ClearQuestはフィールドレベルの検証フックを実行しますが、ユーザーがレコードの編集を完了し、それをデータベースにコミットできるようにしなければ、ClearQuestはアクション検証フックを実行しません。

以下の例では、ユーザーが、担当するプロジェクトに適したクライアントオペレーティングシステム (OS) を入力したかどうかを確認されます。担当するプロジェクトに対して指定した OS がサポートされていない場合、このメソッドは検証エラーメッセージを生成し、それを返します。

VBScript

```
Function defect_Validation(actionname, actiontype)
    ' actionname As String
    ' actiontype As Long
    ' defect_Validation As String
    ' action = teststate
    set sessionObj = GetSession
    ' Get the client OS platform the user indicated.
    platform = GetFieldValue("client_os").GetValue()
    ' Get the project name the user indicated. This information
    ' is stored on a referenced, stateless record.
    projectName = GetFieldValue("project.name").GetValue()
    ' Check the project name against the OS type. If the given project
    ' is not targeted for that platform, return a validation error.
    If projectName = "Gemini" Then
        If platform <> "NT" Then
            defect_Validation = "That project only supports NT."
        End If
    ElseIf projectName = "Aquarius" Then
        If platform <> "Unix" Then
            defect_Validation = "That project only supports Unix."
        End If
    End If
End Function
```

Perl

```
sub defect_Validation {
    my($actionname, $actiontype) = @_ ;
    my $result;
    # $actionname as string scalar
    # $actiontype as long scalar
    # $result as string scalar
    # action = teststate
    # Returns a non-empty string explaining why the action
    # can not commit with the current values.
    # Or, if it is valid, returns an empty string value.
```

```

my ($session,
    $platform,
    $projectRecordID,
    $projectRecord,
    $projectName,
);

$session = $entity->GetSession();

# Get the client OS indicated by the user.
$platform = $entity->GetFieldValue("client_os")->GetValue();

# Get the project name the user indicated. This information
# is stored on a referenced, stateless record.
$projectName = $entity->GetFieldValue("project.name")->GetValue();

# Check the project name against the OS type. If the
# given project is not targeted for that platform,
# return a validation error.
if ($projectName eq "Gemini") {
    if ($platform != "NT") {
        $result = "That project only supports NT.";
    }
} elseif ($projectName eq "Aquarius") {
    if ($platform != "Unix"){
        $result = "That project only supports Unix.";
    }
}
return $result;
}

```

レコード スクリプトの例

VBScript を使用すると、レコード スクリプト、フィールド フック、アクション フックは、Entity オブジェクトに暗黙的に関連付けられます。つまり、特に別の Entity オブジェクト名を指定しない限り、この Entity クラスのメソッドへのすべての呼び出しは、この暗黙的なオブジェクトを参照します。Perl を使用する場合、定義済みの変数 \$entity でこの関連付けを参照してください。

以下の例では、ボタン クリックとコンテキスト メニュー項目選択の両方に応答可能なレコード スクリプトが示されています。ボタンをクリックすると、このフックは、**component_ref** フィールドにコンポーネントのリード エンジニアの名前を設定します。このフィールドには、障害の作業にアサインされている担当者が表示されます。

メモ: この例で、スキーマにレコード スクリプトを追加する方法に関する一般的なアイデアを示します。読みやすくするため、この例にはエラー チェックが含まれていません。データベースにレコードをコミットする前に、**Validate API** の戻り値を確認してエラーがないことを確認してください。

VBScript

```
Function Request_AssignEngineer(param)
    ' param As Variant
    ' This hook responds to changes in the current component and
    ' assigns the request to the lead engineer for that component.
    Dim eventType, componentObj, leadname
    eventType = param.EventType
    If eventType = AD_BUTTON_CLICK Then
        ' Get the lead person for the given component
        leadName = GetFieldValue("component_lead").GetValue
        If leadName = "" Then
            Request_AssignEngineer = "Couldn't get Component Lead value"
            Exit function
        End if
        ' Put that person's name in the Assigned To: field
        SetFieldValue "component_ref", leadName
        Request_AssignEngineer = SetFieldValue "component_ref", leadName
    ElseIf eventType = AD_CONTEXTMENU_ITEM_SELECTION Then
        SetFieldValue "component_ref", GetSession.GetUserFullname
        Request_AssignEngineer = SetFieldValue "component_ref",
        GetSession.GetUserFullname
    End if
End Function
```

グローバル スクリプトの例

以下の例は、現在のユーザーが指定したグループのメンバーであるかを確認するグローバル スクリプトです。ユーザーが指定したグループに属している場合、フックは **True** を返します。

メモ: この例で、グローバル スクリプトを作成する方法に関する一般的なアイデアを示します。読みやすくするため、この例にはエラー チェックが含まれていません。データベースにレコードをコミットする前に、**Validate API** の戻り値を確認してエラーがないことを確認してください。

VBScript

```
Function IsInGroup(groupname)
    ' groupName As String
    ' IsInGroup As Bool
    Set curSession = GetSession
    groupList = curSession.GetUserGroups
    IsInGroup = False
    For Each group in groupList
        If group = groupname Then
            IsInGroup = True
            Exit For
        End If
    Next
End Function
```

Perl

```
sub IsInGroup {
    my ($groupName) = @_ ;
    my ($curSession,
        $groupList,
        $isInGroup,
        $group,
    );

    $curSession = $entity->GetSession();
    $groupList = $curSession->GetUserGroups();
    $isInGroup = 0;

    foreach $group (@$groupList) {
        if ($group eq $groupName) {
            $isInGroup = 1;
        }
    }
}
```

```

        last;
    }
}
return $isInGroup;
}

```

ClearQuest フック スクリプトでの CAL メソッドの使用例

ここでは、ClearQuest フック スクリプトに ClearCase Automation Library (CAL) メソッドを含めて、UCM と ClearQuest の統合の動作をカスタマイズする方法について説明します。以下の例では、UCM_CQActBeforeChact Visual Basic スクリプトのバージョンを変更します。このスクリプトは、作業変更前に ClearQuest アクションを実行するポリシーを実装します。このポリシーが有効な場合は、開発者がアクティビティの終了処理を GUI から、または **cleartool chactivity -cqact** コマンドを入力して開始すると、統合処理でスクリプトが実行されます。

変更されたスクリプトは CAL メソッドを使用して、開発者が単一ストリームのプロジェクトで作業しているのか、複数ストリームのプロジェクトで作業しているのかを判断します。開発者が単一ストリームのプロジェクトで作業している場合、スクリプトによりアクティビティの終了処理が可能になります。それ以外の場合、スクリプトはエラー メッセージを返し、アクティビティの終了処理をキャンセルします。

VBScript

REM Start of Global Script UCM_CQActBeforeChact

関数 UCM_CQActBeforeChact (entity_type, entity_id, project_info, stream_info)

- ' この関数は、「作業変更前に ClearQuest アクションを実行する」ポリシーを実装する
- ' スクリプトです。このスクリプトは、最初にインストールしたときには、デフォルトのスクリプトを
- ' 呼び出します。このポリシーをカスタマイズしたい場合は、デフォルトのスクリプトを
- ' 呼び出すかわりに、ユーザーのコードを使用するように、このスクリプトを編集する
- ' 必要があります。デフォルトのスクリプト コードは例として使用できます。
- '
- ' 入力：
- ' - entity_type: アクションの実行対象のエンティティのタイプ
- ' - entity_id: アクションの実行対象のエンティティの ID (たとえば
- ' "SAMPLE0000001")
- ' 出力：
- ' - アクションが正常に実行された場合、このスクリプトは空白の
- ' 文字列を返す必要があります。
- ' - アクションの実行が失敗した場合、このスクリプトは、エラー メッセージとして表示される文字列
- ' を返す必要があります。

```

' 単一ストリームのプロジェクトのアクティビティである場合のみ chact が許可されます。
proj_model = "DEFAULT"

' フックのセッションのコンテキストを取得します
Set session = GetSession()

' エンティティを取得します
Set entity = session.GetEntity(entity_def_name, entity_id)

' エンティティの ucm_vob_object フィールド値を取得します。この値は、
' UCM プロジェクトのオブジェクト ID と PVOB の UUID が含まれる文字列です。
ucm_vob_object = entity.GetFieldValue("ucm_vob_object").GetValue()
Dim pvob_uuid

' 文字列からプロジェクトのオブジェクト ID 取り出して、PVOB の
' UUID を返します。
pvob_uuid = Right(ucm_vob_object, 40)

' ClearCase.Application COM オブジェクトを初期化します
' ClearCase アプリケーション オブジェクトを作成します。ClearCase アプリケーション
' オブジェクトが存在している場合は、CAL メソッドを使用できます。このスクリプトの
' 残りのステップでは CAL メソッドを使用します。
On Error Resume Next
Set CC = CreateObject("ClearCase.Application")
If Err.Number <> 0 Then
MsgBox "ClearCase.Application Error 1:" & Err.Description
End if
On Error Resume Next

' PVOB の UUID を使用して、アクティビティの ClearCase PVOB オブジェクトを取得します。
Set PVOB = CC.ProjectVOB(pvob_uuid)
If Err.Number <> 0 Then
MsgBox "ClearCase.Application Error 2:" & Err.Description
End if
On Error Resume Next

' ClearCase アクティビティ オブジェクト (CCActivity) を PVOB と
' エンティティ ID (UCM アクティビティ名と等しい) に基づいて作成します。
Set Act = PVOB.Activity(entity_id)
If Err.Number <> 0 Then
MsgBox "ClearCase.Application Error 3:" & Err.Description
End if
On Error Resume Next

' アクティビティが作成されたストリームを返します。
set stream = Act.Stream
If Err.Number <> 0 Then

```

```

MsgBox "ClearCase.Application Error 4:" & Err.Description
End if
On Error Resume Next
' ストリームを含むプロジェクトを返します。
set project = stream.Project
If Err.Number <> 0 Then
MsgBox "ClearCase.Application Error 5:" & Err.Description
End if
On Error Resume Next
' プロジェクト モデルを返します。
proj_model = project.Model
If Err.Number <> 0 Then
MsgBox "ClearCase.Application Error 6:" & Err.Description
End if
' プロジェクト モデルの値をテストします。
' model = SIMPLE: 単一ストリームのプロジェクト
' この値が SIMPLE の場合は単一ストリームであり、スクリプトが
' 空の文字列を返し、開発者がアクティビティの終了処理を
' 完了することができます。
' model = DEFAULT: 階層プロジェクト
' この値が DEFAULT の場合は複数ストリームであり、スクリプトが
' エラー メッセージを返し、アクティビティの終了処理をキャンセルします。
If proj_model = "SIMPLE" Then
' 単一ストリームのモデルであり、アクティビティの変更が許可されます
UCM_CQActBeforeChact = ""
Else
' 階層プロジェクトであり、処理が失敗します
UCM_CQActBeforeChact = "Must be in a single stream project."
End if
End Function
REM End of Global Script UCM_CQActBeforeChact

```

cqperl を使用した詳細レポートと自動化

ClearQuest UNIX 版では、`cqtool` コマンドでコマンド行とバッチをサポートしているほか、`cqperl` で外部 Perl スクリプトを完全にサポートしています。ClearQuest UNIX 版で Perl スクリプトを使用するには `cqperl` が必要です。これ以外のバージョンの Perl では正常に動作しません。

UNIX クライアントで `cqperl` を使用する場合は、次のことを考慮してください。

各 Perl スクリプトには、`CQPerlExt` Perl モジュールをロードするためのコードを含める必要があります。例を次に示します。

`use CQPerlExt;`

ClearQuest Perl API の各要素の詳細については、『Rational ClearQuest API Reference』を参照してください。

以下の例では、`cqperl` コードがレポートを生成します。このレポートは、398 ページの「UNIX での `cqtool` を使用した夜間レポートの実行」で説明するレポートに類似しています。

```
# nightlysubmits.pl - 現在 Submitted 状態にある
# すべての障害を一覧表示する Perl スクリプト
use CQPerlExt;

# すべての ClearQuest の処理はセッション オブジェクトを使用して実行されます。Cqperl
# CQPerlExt Perl モジュールからアクセス可能な CQSession Build
# メソッドを使用してセッション オブジェクトにアクセスします。
# API リファレンス: セッション オブジェクト->Build
$session = CQSession::Build();

# セッションを取得したら、ログインする必要があります。ログイン
# には、UserLogon メソッドを使用します。その際、ユーザー名、
# パスワード、データベース名を指定します。4 番目の
# パラメータ dbset は通常指定しません。
# API リファレンス: セッション オブジェクト->UserLogon メソッド
$session->UserLogon("admin", "", "SAMPL", "");

# クエリーを生成するには、QueryDef オブジェクトを作成する
# が必要です。これには、BuildQuery というセッション オブ
# ジェクト メソッドを使用します。パラメータは entitydef のみ
# をとります。これはレコード タイプとも呼ばれ、クエリーの対象を
# 指定します。この例では「Defect」を指定します。
# API リファレンス: セッション オブジェクト->BuildQuery メソッド
#
# QueryDef オブジェクト
# EntityDef オブジェクト->Name プロパティ
$querydef = $session->BuildQuery("Defect");
# 次に (ClearQuest UNIX GUI を使用したクエリーの作成と
```

```

# 同様)、クエリー結果セットに含めるフィールドを決定します。
# これには、QueryDef オブジェクトの BuildField メソッドを
# 使用します。ID、見出し、登録者を
# 表示します。
#   API リファレンス: QueryDef オブジェクト->BuildField メソッド
$querydef->BuildField("id");
$querydef->BuildField("headline");
$querydef->BuildField("submitter");
# 次に、このクエリーで使用するフィルタを作成する必要があります。
# フィルタは、FilterOperator オブジェクトのツリーを構築することに
# より作成します。サブツリーの最上位 FilterOperator オブジェクト
# を作成するには、QueryDef オブジェクトの BuildFilterOperator
# メソッドを使用します。BuildFilterOperator メソッドは引数を
# 1 つとります。これはブール型演算子で、各サブツリーの
# 動作を決定します。フィルタが 1 つのみの場合、AND、
# OR のいずれも使用できます。ブール型演算子を適切に
# 指定するためには、適切な BoolOp 定数と Perl
# プレフィックスを選択します。たとえば、使用する定数は
# $CQPerlExt::CQ_BOOL_OP_AND のようになります。
#   API リファレンス: QueryDef オブジェクト->BuildFilterOperator メソッド
#
#           BoolOp 定数
#           Perl の表記規則
$rootfilternode =
    $querydef->BuildFilterOperator($CQPerlExt::CQ_BOOL_OP_AND);
# ルート FilterOperatorNode ができたら、それにフィルタを
# 割り当てます。その場合、state は submitted になります。
# この処理には、QueryFilterNode オブジェクトの BuildFilter
# メソッドを使用します。BuildFilter の 3 番目のパラメータは
# 配列への Perl 参照であることに注意してください。
#   API リファレンス: QueryFilterNode オブジェクト->BuildFilter メソッド
#
#           BoolOp 定数
#           Perl の表記規則
@statetest = "Submitted";
$rootfilternode->BuildFilter("State",
                                $CQPerlExt::CQ_COMP_OP_EQ,
                                \@statetest);
# クエリーが定義できました。それでは実行してみましょう。
# セッション オブジェクトに戻り、BuildResultSet
# メソッドを使用します。パラメータは 1 つのみで、

```

```

# 前に作成した QueryDef オブジェクトです。結果セット
# オブジェクトの準備ができたら、クエリーを実行します。
#   API リファレンス: Session オブジェクト->BuildResultSet メソッド
#                               ResultSet オブジェクト->Execute メソッド
$resultset = $session->BuildResultSet($querydef);
$resultset->Execute();
# 出力に備えて見出しを印刷します。
printf("%13.13s %50.50s %9.9s\n", "id", "headline", "submitter");
printf("%13.13s %50.50s %9.9s\n",
      "-----",
      "-----",
      "-----");
# 結果セットに移動し、出力を印刷します。
# この処理には、結果セット オブジェクトの MoveNext メソッドを
# 使用します。表示する行がある限り $CQPerlExt::CQ_SUCCESS
# が返されます。GetColumnValue を使用して結果セットの
# 行からデータを取得します。
#   API リファレンス: ResultSet オブジェクト->MoveNext メソッド
#                               ResultSet オブジェクト->GetColumnValue メソッド
while ($resultset->MoveNext() == $CQPerlExt::CQ_SUCCESS) {
    printf("%-13.13s %-50.50s %-9.9s\n",
          $resultset->GetColumnValue(1),
          $resultset->GetColumnValue(2),
          $resultset->GetColumnValue(3));
# 完了したので、セッションを解放します。
CQSession::Unbuild($session);

```


ここでは、Rational ClearQuest の次のフォーム コントロールについて説明します。

- 360 ページの「ActiveX コントロール」
- 361 ページの「添付ファイル コントロール」
- 362 ページの「チェック ボックス コントロール」
- 363 ページの「コンボ ボックス コントロール」
- 364 ページの「ドロップダウン リスト ボックス コントロール」
- 366 ページの「ドロップダウン コンボ ボックス コントロール」
- 367 ページの「重複元コントロール」
- 368 ページの「重複先コントロール」
- 369 ページの「グループ ボックス コントロール」
- 369 ページの「履歴コントロール」
- 370 ページの「リスト ボックス コントロール」
- 371 ページの「一覧コントロール」
- 372 ページの「オプション ボタン コントロール」
- 373 ページの「親子コントロール」
- 375 ページの「図コントロール」
- 375 ページの「プッシュ ボタン コントロール」
- 377 ページの「静的テキスト ボックス コントロール」
- 377 ページの「テキスト ボックス コントロール」

レコード フォームへのコントロールの追加方法については、「第 7 章 フォームの操作」を参照してください。

ActiveX コントロール

ClearQuest の ActiveX コントロールを使用して、登録済みの ActiveX コントロールをフォームに組み込みます。たとえば、ActiveX コントロールを使用して、外部のデータベースにアクセスし、更新することができます。ActiveX コントロールを使用する場合には、このコントロールを含むフォームが開いたときに起動する初期化レコードスクリプトと、ユーザーがアクション (Submit など) を開始したとき、またはアクションが完了または再開したときに起動するアクションレコードスクリプトを記述する必要があります。

このコントロールを使用するには、ActiveX の機能と、コントロールを登録する方法に熟知している必要があります。

ActiveX コントロールのプロパティシートには、[全般] と [ActiveX コントロール] の 2 つのタブがあります。

[全般] タブ

次の表に、[全般] タブのプロパティを示します。

プロパティ	説明
フィールド名	コントロールに関連付けるフィールドを選択します。
ラベル	コントロールに対するラベルを入力します。ClearQuest のデフォルトの設定では、ラベルは選択したフィールドの名前に設定されます。
X	コントロールの左端の位置 (ピクセル単位)。
Y	コントロールの上端の位置 (ピクセル単位)。
幅	コントロールの幅 (ピクセル単位)。
高さ	コントロールの高さ (ピクセル単位)。

[ActiveX コントロール] タブ

次の表に、[ActiveX コントロール] タブのプロパティを示します。

プロパティ	説明
登録された名前	使用する ActiveX コントロールの登録名。
初期化レコード スクリプト	このドロップダウン リストでレコード スクリプトを選択します。このレコード スクリプトは、コントロールを初期化する際に使用されます。また、コントロールのフォームが開いたときに起動されます。
アクション レコード スクリプト	このドロップダウン リストでレコード スクリプトを選択します。このレコード スクリプトは ClearQuest の実行中ユーザーが Submit や Modify などのアクションを開始するたびに、または、そのアクションが完了または元に戻されたときに呼び出されます。

添付ファイル コントロール

添付ファイル コントロールには、添付ファイルが一覧表示されます。また、このコントロールには、ユーザーが添付ファイルの追加、削除、表示を行うためのコントロールのセットも含まれます。添付ファイル コントロールは、添付ファイル フィールド、つまり ATTACHMENT_LIST 型のフィールドと関連付けます。

[全般] タブ

次の表に、[全般] タブのプロパティを示します。

プロパティ	説明
フィールド名	コントロールに関連付けるフィールドを選択します。
ラベル	コントロールに対するラベルを入力します。 ClearQuest のデフォルトの設定では、ラベルは選択したフィールドの名前に設定されます。
X	コントロールの左端の位置 (ピクセル単位)。
Y	コントロールの上端の位置 (ピクセル単位)。
幅	コントロールの幅 (ピクセル単位)。
高さ	コントロールの高さ (ピクセル単位)。

チェック ボックス コントロール

チェック ボックス コントロールは 2 つの値を持つコントロールで、ブール値または 2 つの値しか持たないフィールドに対して使用することができます。チェック ボックス コントロールは、オン/オフのいずれかの状態になります。このコントロールを作成する際、オンまたはオフになった場合に、対応するフィールドに関連付ける値を指定します。コントロールのオン用の値とオフ用の値として 2 つの文字列を割り当てることができます。

このコントロールのフィールドに対して、デフォルト値フックを設定することが必要です。コントロールの初期値を明示的に設定しなければ、ClearQuest により、オンが初期値になります。

チェック ボックス コントロールのプロパティ シートには、次の 2 つのタブがあります。

[全般] タブ

次の表に、[全般] タブのプロパティを示します。

プロパティ	説明
フィールド名	コントロールに関連付けるフィールドを選択します。
ラベル	コントロールに対するラベルを入力します。ClearQuest のデフォルトの設定では、ラベルは選択したフィールドの名前に設定されます。
X	コントロールの左端の位置 (ピクセル単位)。
Y	コントロールの上端の位置 (ピクセル単位)。
幅	コントロールの幅 (ピクセル単位)。
高さ	コントロールの高さ (ピクセル単位)。

[拡張] タブ

次の表に、[拡張] タブのプロパティを示します。

プロパティ	説明
チェックする値	このチェック ボックスがオンになっているときに表示するテキストを入力します。
チェックしない値	このチェック ボックスがオフになっているときに表示するテキストを入力します。

コンボ ボックス コントロール

コンボ ボックス コントロールは、編集可能なテキスト ボックスとリスト ボックスが組み合わされたコントロールです。コンボ ボックスは特定のフィールドに関連付けられている選択リストを表示します。ユーザーはコンボ ボックスの編集可能なテキスト ボックスに値を入力することもできます。リストの項目数がリストのサイズを超える場合、ClearQuest により、このコントロールの右側に縦のスクロール バーが自動的に追加されます。

コンボ ボックス コントロールには、複数行フィールド タイプと参照リスト フィールド タイプのみを割り当てることができます。ただし、コンボ ボックスに割り当てられるフィールドは、選択リストに関連付けられているか、コンボ ボックスを使用して選択リストを追加できる必要があります。ほとんどのフィールド タイプの場合、コンボ ボックスでは、選択リストから 1 項目のみを選択できます。ただし、ClearQuest では、複数選択が可能なフィールドもあります。

コンボ ボックス コントロールのプロパティ シートには、[全般]、[拡張]、[コンテキストメニュー フック] の 3 つのタブがあります。

[全般] タブ

次の表に、[全般] タブのプロパティを示します。

プロパティ	説明
フィールド名	コントロールに関連付けるフィールドを選択します。
ラベル	コントロールに対するラベルを入力します。ClearQuest のデフォルトの設定では、ラベルは選択したフィールドの名前に設定されます。
X	コントロールの左端の位置 (ピクセル単位)。
Y	コントロールの上端の位置 (ピクセル単位)。
幅	コントロールの幅 (ピクセル単位)。
高さ	コントロールの高さ (ピクセル単位)。

[拡張] タブ

[拡張] タブには、自動ソート プロパティのチェック ボックスのみがあります。このチェック ボックスをオンにすると、ドロップダウン コンボ ボックスのリスト内の項目がアルファベット順にソートされます。このチェック ボックスをオフにすると、リスト内の項目は追加された順番になります。

[コンテキストメニュー フック] タブ

[コンテキストメニュー フック] タブには、使用できるレコードフックと、このコントロールに現在関連付けられているフックのリストが表示されます。コントロールにフックを関連付けるには、[使用可] ボックスからフックを選択し、[追加] をクリックします。フックの関連付けを解除するには、[選択] ボックスからフックを選択し、[削除] をクリックします。このコントロールに関連付けられたフックは、ClearQuest クライアント アプリケーションのコントロールのショートカット メニューに表示されます。ユーザーがフックを実行するには、コントロールを右クリックし、表示されるショートカット メニューからフック名を選択する必要があります。

[Web 依存先フィールド] タブ

ClearQuest Web クライアントで依存フィールドを有効にするには、[Web 依存先フィールド] タブで、依存フィールドを指定する必要があります。これは、フォームにフィールドを追加する場合に行います。

依存フィールドを指定するには、[使用可] ボックスからフィールドを選択し、[追加] をクリックします。フィールドを削除するには、[選択] ボックスからフィールドを選択し、[削除] をクリックします。

ドロップダウン リスト ボックス コントロール

ドロップダウン リスト ボックス コントロールには、特定のフィールドの可能値が一覧表示されます。このコントロールをクリックすると、そのフィールドに関連付けられている値が一覧表示され、ユーザーは、ここから値を選択できます。フィールドの動作が読み取り専用設定されている場合、現在選択されている値が表示され、ユーザーはこの値を編集できません。ユーザーはドロップダウン リスト ボックスの項目を編集することはできませんが、選択リストフックを使用して、プログラミングによりリストを更新することはできます。

ドロップダウン リスト ボックス コントロールには、複数行フィールド タイプと参照リストフィールド タイプのみを割り当てることができます。ただし、このコントロールを割り当てるフィールドには、選択リストが関連付けられている必要があります。このコントロールのリストからは、一度に 1 項目のみを選択できます。

ドロップダウン リスト ボックス コントロールのプロパティ シートには、[全般]、[拡張]、[コンテキストメニュー フック]、[Web 依存先フィールド] の 4 つのタブがあります。

[全般] タブ

次の表に、[全般] タブのプロパティを示します。

プロパティ	説明
フィールド名	コントロールに関連付けるフィールドを選択します。
ラベル	コントロールに対するラベルを入力します。 ClearQuest のデフォルトの設定では、ラベルは選択したフィールドの名前に設定されます。
X	コントロールの左端の位置 (ピクセル単位)。
Y	コントロールの上端の位置 (ピクセル単位)。
幅	コントロールの幅 (ピクセル単位)。
高さ	コントロールの高さ (ピクセル単位)。

[拡張] タブ

[拡張] タブには、自動ソート プロパティのチェック ボックスのみがあります。このチェック ボックスをオンにすると、ドロップダウン コンボ ボックスのリスト内の項目がアルファベット順にソートされます。このチェック ボックスをオフにすると、リスト内の項目は追加された順番になります。

[コンテキスト メニュー フック] タブ

[コンテキスト メニュー フック] タブには、使用できるレコードフックと、このコントロールに現在関連付けられているフックのリストが表示されます。コントロールにフックを関連付けるには、[使用可] ボックスからフックを選択し、[追加] をクリックします。フックの関連付けを解除するには、[選択] ボックスからフックを選択し、[削除] をクリックします。このコントロールに関連付けられたフックは、**ClearQuest** クライアント アプリケーションのコントロールのショートカット メニューに表示されます。ユーザーがフックを実行するには、コントロールを右クリックし、表示されるショートカット メニューからフック名を選択する必要があります。

[Web 依存先フィールド] タブ

ClearQuest Web クライアントで依存フィールドを有効にするには、[Web 依存先フィールド] タブで、依存フィールドを指定する必要があります。これは、フォームにフィールドを追加する場合に行います。

依存フィールドを指定するには、[使用可] ボックスからフィールドを選択し、[追加] をクリックします。フィールドを削除するには、[選択] ボックスからフィールドを選択し、[削除] をクリックします。

ドロップダウン コンボ ボックス コントロール

ドロップダウン コンボ ボックス コントロールは、編集可能なテキスト ボックスとドロップダウン リスト ボックスが組み合わされたコントロールです。コンボ ボックスには、特定のフィールドの、現在選択されている項目が表示されていますが、ユーザーは、ドロップダウン リストから別の項目を選択したり、テキスト ボックスに値を入力できます。

ドロップダウン コンボ ボックス コントロールには、ほとんどのフィールド タイプを割り当てることができます。ただし、このコントロールに割り当てるフィールドは、関連付けられている選択リストを持っているか、ドロップダウン コンボ ボックスを使用して選択リストを追加できる必要があります。このコントロールのリストからは、一度に 1 項目のみを選択できます。

ドロップダウン コンボ ボックス コントロールのプロパティ シートには、[全般]、[拡張]、[コンテキスト メニュー フック]、[Web 依存先フィールド] の 4 つのタブがあります。

[全般] タブ

次の表に、[全般] タブのプロパティを示します。

プロパティ	説明
フィールド名	コントロールに関連付けるフィールドを選択します。
ラベル	コントロールに対するラベルを入力します。ClearQuest のデフォルトの設定では、ラベルは選択したフィールドの名前に設定されます。
X	コントロールの左端の位置 (ピクセル単位)。
Y	コントロールの上端の位置 (ピクセル単位)。
幅	コントロールの幅 (ピクセル単位)。
高さ	コントロールの高さ (ピクセル単位)。

[拡張] タブ

[拡張] タブには、自動ソート プロパティのチェック ボックスのみがあります。このチェック ボックスをオンにすると、ドロップダウン コンボ ボックスのリスト内の項目がアルファベット順にソートされます。このチェック ボックスをオフにすると、リスト内の項目は追加された順番になります。

[コンテキスト メニュー フック] タブ

[コンテキスト メニュー フック] タブには、使用できるレコードフックと、このコントロールに現在関連付けられているフックのリストが表示されます。コントロールにフックを関連付けるには、[使用可] ボックスからフックを選択し、[追加] をクリックします。フックの関連付けを解除するには、[選択] ボックスからフックを選択し、[削除] をクリックします。このコントロールに関連付けられたフックは、ClearQuest クライアント アプリケーションのコントロールのショートカット メニューに表示されます。ユーザーがフックを実行するには、コントロールを右クリックし、表示されるショートカット メニューからフック名を選択する必要があります。

[Web 依存先フィールド] タブ

ClearQuest Web クライアントで依存フィールドを有効にするには、[Web 依存先フィールド] タブで、依存フィールドを指定する必要があります。これは、フォームにフィールドを追加する場合に行います。

依存フィールドを指定するには、[使用可] ボックスからフィールドを選択し、[追加] をクリックします。フィールドを削除するには、[選択] ボックスからフィールドを選択し、[削除] をクリックします。

重複元コントロール

このコントロールは、重複レコードの重複元であるレコードの ID を表示します。このコントロールは、フィールドとの関連付けができないことから、特殊なタイプのテキスト ボックス コントロールと言えます。ClearQuest により、そのレコードが別のレコードの重複であることが確認されると、このコントロールの内容が更新されます。

メモ: このコントロールを作成する場合には、コントロールを、単一のレコード ID を格納できるだけのサイズに設定してください。レコードがほかのレコードの重複の場合、重複元のレコードは 1 つのみです。

重複元コントロールのプロパティ シートには、[全般] タブがあります。

[全般] タブ

次の表に、[全般] タブのプロパティを示します。

プロパティ	説明
フィールド名	コントロールに関連付けるフィールドを選択します。
ラベル	コントロールに対するラベルを入力します。ClearQuest のデフォルトの設定では、ラベルは選択したフィールドの名前に設定されます。

プロパティ	説明
X	コントロールの左端の位置 (ピクセル単位)。
Y	コントロールの上端の位置 (ピクセル単位)。
幅	コントロールの幅 (ピクセル単位)。
高さ	コントロールの高さ (ピクセル単位)。

重複先コントロール

このコントロールは、このレコードの重複レコードであるレコードがあれば、その ID を表示します。このコントロールは、フィールドと関連付けされないことから、特殊なタイプのリスト ボックス コントロールと言えます。ClearQuest により、このレコードの重複となる 1 つまたは複数のレコードが確認されると、このコントロールの内容が更新されます。

メモ: 1 つのレコードに複数の重複がある場合に備えて、このコントロールのサイズを調整する必要があります。一度に表示できない数の重複が存在する場合、ClearQuest により、自動的にスクロールバーが付き、リスト内をスクロールできます。

重複先コントロールのプロパティ シートには、[全般] タブがあります。

[全般] タブ

次の表に、[全般] タブのプロパティを示します。

プロパティ	説明
フィールド名	コントロールに関連付けるフィールドを選択します。
ラベル	コントロールに対するラベルを入力します。ClearQuest のデフォルトの設定では、ラベルは選択したフィールドの名前に設定されます。
X	コントロールの左端の位置 (ピクセル単位)。
Y	コントロールの上端の位置 (ピクセル単位)。
幅	コントロールの幅 (ピクセル単位)。
高さ	コントロールの高さ (ピクセル単位)。

グループ ボックス コントロール

グループ ボックス コントロールは、1 つまたは複数のコントロールを視覚的にグループ化するためのコントロールです。このコントロールにフィールドを関連付けることはできませんが、ラベルを割り当てることはできます。通常、グループ ボックス コントロールのラベルには、グループ ボックス内のコントロールの目的を指定します。

グループ ボックス コントロールのプロパティ シートには、[全般] タブがあります。

[全般] タブ

次の表に、[全般] タブのプロパティを示します。

プロパティ	説明
フィールド名	コントロールに関連付けるフィールドを選択します。
ラベル	コントロールに対するラベルを入力します。ClearQuest のデフォルトの設定では、ラベルは選択したフィールドの名前に設定されます。
X	コントロールの左端の位置 (ピクセル単位)。
Y	コントロールの上端の位置 (ピクセル単位)。
幅	コントロールの幅 (ピクセル単位)。
高さ	コントロールの高さ (ピクセル単位)。

履歴コントロール

履歴コントロールは、レコードに対して実行されたアクションの履歴を表示します。履歴コントロールそのものは特殊な一覧で、各項目には、そのレコードに対して実行された単一のアクションに関する情報が示されます。ClearQuest ではレコードのアクション履歴は内部的に管理され、フォームの履歴コントロール上に表示されます。履歴コントロールとフィールドを関連付けることはできません。

履歴コントロールのプロパティ シートには、[全般] タブがあります。

[全般] タブ

次の表に、[全般] タブのプロパティを示します。

プロパティ	説明
フィールド名	コントロールに関連付けるフィールドを選択します。
ラベル	コントロールに対するラベルを入力します。ClearQuest のデフォルトの設定では、ラベルは選択したフィールドの名前に設定されます。
X	コントロールの左端の位置 (ピクセル単位)。
Y	コントロールの上端の位置 (ピクセル単位)。
幅	コントロールの幅 (ピクセル単位)。
高さ	コントロールの高さ (ピクセル単位)。

リスト ボックス コントロール

リスト ボックス コントロールには、特定のフィールドの可能値が一覧表示されます。リスト ボックス内の項目数は、選択リスト フックを使用して、プログラミングにより項目を変更しない限り、固定されています。リストの項目数がリストのサイズを超える場合、ClearQuest により、このコントロールの右側に縦のスクロール バーが自動的に追加されます。

リスト ボックス コントロールには、ほとんどのフィールドタイプを割り当てることができます。ただし、このコントロールを割り当てるフィールドには、選択リストが関連付けられている必要があります。ほとんどのフィールドタイプの場合、このコントロールでは、選択リストから 1 項目のみを選択できます。ただし、ClearQuest では、複数選択が可能なフィールドもあります。

リスト ボックス コントロールのプロパティ シートには、[全般]、[拡張]、[コンテキスト メニュー フック] の 3 つのタブがあります。

[全般] タブ

次の表に、[全般] タブのプロパティを示します。

プロパティ	説明
フィールド名	コントロールに関連付けるフィールドを選択します。
ラベル	コントロールに対するラベルを入力します。ClearQuest のデフォルトの設定では、ラベルは選択したフィールドの名前に設定されます。
X	コントロールの左端の位置 (ピクセル単位)。

プロパティ	説明
Y	コントロールの上端の位置 (ピクセル単位)。
幅	コントロールの幅 (ピクセル単位)。
高さ	コントロールの高さ (ピクセル単位)。

[拡張] タブ

[拡張] タブには、[自動ソート] チェック ボックスのみがあります。[自動ソート] をオンにすると、リスト ボックス内の項目がアルファベット順にソートされます。[自動ソート] をオフにすると、リスト内の項目は追加された順番になります。

[コンテキスト メニュー フック] タブ

[コンテキスト メニュー フック] タブには、使用できるレコードフックと、このコントロールに現在関連付けられているフックのリストが表示されます。コントロールにフックを関連付けるには、[使用可] ボックスからフックを選択し、[追加] をクリックします。フックの関連付けを解除するには、[選択] ボックスからフックを選択し、[削除] をクリックします。このコントロールに関連付けられたフックは、ClearQuest クライアント アプリケーションのコントロールのショートカット メニューに表示されます。ユーザーがフックを実行するには、コントロールを右クリックし、表示されるショートカット メニューからフック名を選択する必要があります。

一覧コントロール

一覧コントロールを使用して、REFERENCE_LIST 型のフィールドに関連付けられているレコードを表示することができます。一覧コントロールの主な使用目的は、親子関係での親を表示することです。一覧を使用して関連レコードをリンクする場合は、親子コントロールを使用してください。

- 一覧コントロールに表示する、レコードのフィールドを指定するには、列を一覧に追加する必要があります。これを行うには、ヘッダー バーを右クリックし、ショートカット メニューの [列の追加] をクリックします。
- 各列は、参照されたレコードの特定のフィールドを参照します。列を右クリックし、ショートカット メニューの [プロパティ] をクリックし、列名と参照するフィールドを指定します。

メモ: フォーム上に逆参照フィールドを表示するには、一覧コントロールを使用する必要があります。逆参照フィールドは読み取り専用の REFERENCE_LIST 型フィールドで、親子関係における親レコードの情報を格納します。

一覧コントロールのプロパティ シートには、[全般]、[拡張]、[コンテキスト メニュー フック] の 3 つのタブがあります。

[全般] タブ

次の表に、[全般] タブのプロパティを示します。

プロパティ	説明
フィールド名	コントロールに関連付けるフィールドを選択します。
ラベル	コントロールに対するラベルを入力します。ClearQuest のデフォルトの設定では、ラベルは選択したフィールドの名前に設定されます。
X	コントロールの左端の位置 (ピクセル単位)。
Y	コントロールの上端の位置 (ピクセル単位)。
幅	コントロールの幅 (ピクセル単位)。
高さ	コントロールの高さ (ピクセル単位)。

[拡張] タブ

[拡張] タブには、一覧の ID を入力するためのテキスト ボックスのみがあります。この ID は、そのタブにあるほかの一覧コントロールとの間で一意な値であることが必要です。この ID は、一覧と関連付けるプッシュ ボタンのために必要となります。

[コンテキスト メニュー フック] タブ

[コンテキスト メニュー フック] タブには、使用できるレコードフックと、このコントロールに現在関連付けられているフックのリストが表示されます。コントロールにフックを関連付けるには、[使用可] ボックスからフックを選択し、[追加] をクリックします。フックの関連付けを解除するには、[選択] ボックスからフックを選択し、[削除] をクリックします。このコントロールに関連付けられたフックは、ClearQuest クライアントのコントロールのショートカット メニューに表示されます。ユーザーがフックを実行するには、コントロールを右クリックし、表示されるショートカット メニューからフック名を選択する必要があります。

オプション ボタン コントロール

相互排他的なオプションのセットを作成するには、グループ化したオプション ボタン コントロールを使用します。各オプション ボタンをグループ内に作成し、同じグループ名を割り当てます。ClearQuest では、同じグループ名で作成したオプション ボタンの各グループには、ラベルが 1 つ付きます。

各オプション ボタンはフィールドに関連付けることができ、ボタンが選択されたときにそのフィールドに入力する適切な値を設定できます。グループ内の各オプション ボタンに別々のフィールドを割り当てることもできます。

オプション ボタン コントロールのプロパティ シートには、[全般] と [拡張] の 2 つのタブがあります。

[全般] タブ

次の表に、[全般] タブのプロパティを示します。

プロパティ	説明
フィールド名	コントロールに関連付けるフィールドを選択します。
ラベル	コントロールに対するラベルを入力します。 ClearQuest のデフォルトの設定では、ラベルは選択したフィールドの名前に設定されます。
X	コントロールの左端の位置 (ピクセル単位)。
Y	コントロールの上端の位置 (ピクセル単位)。
幅	コントロールの幅 (ピクセル単位)。
高さ	コントロールの高さ (ピクセル単位)。

次の表に、[拡張] タブのプロパティを示します。

プロパティ	説明
グループ名	このオプション ボタンに関連付けるグループ名。グループ内のオプション ボタンは、一度に 1 つのみを選択できます。グループ内の別のボタンを選択すると、現在選択されているボタンの選択が解除されます。
グループ ラベル	オプション ボタンのグループのラベル。同一のグループ名を持つオプション ボタンの各グループに 1 つのラベルが付きます。
値	このオプション ボタンが選択された場合に返される値。

親子コントロール

親子コントロールは、関連付けられたレコードをユーザーがリンクすることができるようなフォームを設定するときに使用します。親子コントロールは、REFERENCE_LIST 型のフィールドで使用します。

親子コントロールは、1 つの一覧コントロールと 3 つのプッシュ ボタンから構成されています。一覧コントロールとプッシュ ボタンは、一意の一覧 ID で自動的に関連付けられます。一覧 ID を変更する場合は、プッシュ ボタンも更新する必要があります。

親子コントロールに表示する、参照先レコードのフィールドを指定するには、一覧に列を追加します。その方法は次のとおりです。

- ヘッダー バーを右クリックし、ショートカット メニューの [列の追加] をクリックします。
- 列を右クリックし、ショートカット メニューの [プロパティ] をクリックし、列名と参照するフィールドを指定します。各列は、参照されたレコードの特定のフィールドを参照します。

親子コントロールのプロパティ シートには、[全般]、[拡張]、[コンテキスト メニュー フック] の 3 つのタブがあります。

[全般] タブ

次の表に、[全般] タブのプロパティを示します。

プロパティ	説明
フィールド名	コントロールに関連付けるフィールドを選択します。
ラベル	コントロールに対するラベルを入力します。 ClearQuest のデフォルトの設定では、ラベルは選択したフィールドの名前に設定されます。
X	コントロールの左端の位置 (ピクセル単位)。
Y	コントロールの上端の位置 (ピクセル単位)。
幅	コントロールの幅 (ピクセル単位)。
高さ	コントロールの高さ (ピクセル単位)。

[拡張] タブ

[拡張] タブには、親子コントロールの ID を入力するためのテキスト ボックスのみがあります。この ID は、そのタブ上にあるほかの一覧コントロールまたは親子コントロールとの間で一意な値であることが必要です。また、関連付けられているボタンが使用する一覧 ID と同じであることも必要です。一覧 ID を変更する場合は、プッシュ ボタンも更新する必要があります。

[コンテキスト メニュー フック] タブ

[コンテキスト メニュー フック] タブには、使用できるレコードフックと、このコントロールに現在関連付けられているフックのリストが表示されます。コントロールにフックを関連付けるには、[使用可] ボックスからフックを選択し、[追加] をクリックします。フックの関連付けを解除するには、[選択] ボックスからフックを選択し、[削除] をクリックします。このコントロールに関連付けられたフックは、**ClearQuest** クライアントのコントロールのショートカット メニューに表示されます。ユーザーがフックを実行するには、コントロールを右クリックし、表示されるショートカット メニューからフック名を選択する必要があります。

図コントロール

図コントロールを使用すれば、フォーム上に動かない画像を表示できます。図コントロールを使用して、どのような .bmp イメージ ファイルでも表示できます。ビットマップ イメージは、ファイルまたはクリップボードのいずれかから取得します。ClearQuest では、ビットマップ イメージは、図コントロールの幅と高さに合うように、自動的にサイズが調整されます。

次の表に、図コントロールのプロパティを示します。

プロパティ	説明
ファイル	ファイルから図を読み込む場合は、このオプションを選択します。このオプションを選択した場合、[図のファイル名] テキスト ボックスにファイル名を入力する必要があります。
クリップボード	クリップボードから図を読み込むには、このオプションを選択します。
図のファイル名	このテキスト ボックスに、ファイル名を入力します。
X	X 軸上における図の左上隅の位置 (ピクセル単位)。
Y	Y 軸上における図の左上隅の位置 (ピクセル単位)。
幅	図の幅 (ピクセル単位)。
高さ	図の高さ (ピクセル単位)。

プッシュ ボタン コントロール

プッシュ ボタン コントロールを使用して、レコードに関連する特定のタスクを実行できます。プッシュ ボタンは、レコードフックまたは一覧に関連付けることができます。

- レコードフックに関連付けた場合、ボタンをクリックすると、指定したレコードフックが実行されます。スキーマに関連付けるレコードフックの定義とデバッグは、設計者の責任で行ってください。
- 一覧に関連付ける場合、プッシュ ボタンを事前定義された 3 つのタスク (レコードの追加、レコードの削除、レコードの新規作成) のいずれかに割り当てることが必要です。
ClearQuest では、これらのタスクを実行するコードが提供されています。プッシュ ボタンは、対応する一覧 ID と関連付けることも必要です。

メモ: ダイアログ ボックスやほかの Windows アプリケーションを呼び出すスクリプト ボタンは、ClearQuest Web クライアントでは使用しないでください。

プッシュ ボタン コントロールのプロパティ シートには、[全般] と [拡張] の 2 つのタブがあります。

[全般] タブ

次の表に、[全般] タブのプロパティを示します。

プロパティ	説明
フィールド名	コントロールに関連付けるフィールドを選択します。
ラベル	コントロールに対するラベルを入力します。ClearQuest のデフォルトの設定では、ラベルは選択したフィールドの名前に設定されます。
X	コントロールの左端の位置 (ピクセル単位)。
Y	コントロールの上端の位置 (ピクセル単位)。
幅	コントロールの幅 (ピクセル単位)。
高さ	コントロールの高さ (ピクセル単位)。

[拡張] タブ

プッシュ ボタンをレコード フックまたは一覧コントロールに関連付ける場合は、[拡張] タブを使用します。

プロパティ	説明
クリック フック	ボタンがクリックされたときに実行するレコード スクリプトを選択します。
WEB で有効	ボタンを Web で有効にする場合は、[クリック フック] をオンにします。
関連付けられたコンポーネント	ボタンに関連付ける一覧 ID を選択します。
追加	一覧に項目を追加する場合に選択します。
削除	一覧から項目を削除する場合に選択します。
新規	一覧に新規レコードを追加する場合に選択します。
その他	一覧にレコード スクリプトを関連付ける場合を選択します。

静的テキスト ボックス コントロール

静的テキスト ボックスは、編集できないテキストを表示します。静的テキスト ボックス コントロールにフィールドを割り当てることはできますが、一般的には、ユーザーがフィールドの内容を修正できるようにテキスト ボックスを使用する方がよいとされています。ユーザーがフィールドの内容を編集しないようにする場合に、静的テキスト ボックスを使用します。特定のアクションが行われている間のみ、フィールドを編集できないようにする場合は、テキスト ボックス コントロールを使用して、フィールドの動作設定を調整します。

静的テキスト ボックス コントロールのプロパティ シートには、[全般] タブがあります。

[全般] タブ

次の表に、[全般] タブのプロパティを示します。

プロパティ	説明
フィールド名	コントロールに関連付けるフィールドを選択します。
ラベル	コントロールに対するラベルを入力します。ClearQuest のデフォルトの設定では、ラベルは選択したフィールドの名前に設定されます。
X	コントロールの左端の位置 (ピクセル単位)。
Y	コントロールの上端の位置 (ピクセル単位)。
幅	コントロールの幅 (ピクセル単位)。
高さ	コントロールの高さ (ピクセル単位)。

テキスト ボックス コントロール

テキスト ボックス コントロールは、フィールドの値を編集可能なテキスト文字列として表示します。フィールドの動作がオプションまたは必須であった場合、ユーザーは、テキスト ボックス コントロールの内容を編集することで、フィールドを修正できます。変更できないテキスト文字列を表示する場合は、静的テキスト ボックス コントロールを使用します。

テキスト ボックス コントロールを使用する場合は、入力テキストの外観に関係するさまざまなオプションを選択できます。入力文字のエコーを無効にすることができます。これは、パスワードなどのセキュリティ情報を入力する場合に役立ちます。日付/時刻情報の設定は制御できます。複数の行を使用して、ユーザーがコントロール内のテキストをフォーマットできるように設定することができます。また、テキスト ボックスの内容を自動または手動でスクロールするためのスクロール バーを追加することもできます。

テキスト ボックス コントロールのプロパティ シートには、[全般]、[拡張]、[日付/時刻]、[コンテキスト メニュー フック] の 4 つのタブがあります。

[全般] タブ

次の表に、[全般] タブのプロパティを示します。

プロパティ	説明
フィールド名	コントロールに関連付けるフィールドを選択します。
ラベル	コントロールに対するラベルを入力します。 ClearQuest のデフォルトの設定では、ラベルは選択したフィールドの名前に設定されます。
X	コントロールの左端の位置 (ピクセル単位)。
Y	コントロールの上端の位置 (ピクセル単位)。
幅	コントロールの幅 (ピクセル単位)。
高さ	コントロールの高さ (ピクセル単位)。

[拡張] タブ

次の表に、[拡張] タブのプロパティを示します。

プロパティ	説明
パスワード/エコーなし スタイル	テキスト ボックス内の文字をセキュリティ目的でアスタリスクに置き換えます。
複数行	テキスト ボックス内のテキストを複数の行構成にします。このオプションを無効にすると、テキストはすべて 1 行内に置かれます。
自動縦スクロール	このチェック ボックスをオンにすると、ユーザーがテキスト ボックスのサイズ以上に入力しようとしたり、テキスト ボックスの上端より上または下端より下にあるテキストを選択しようとする、自動的に縦方向にスクロールします。
自動横スクロール	このチェック ボックスをオンにすると、ユーザーがテキスト ボックスのサイズ以上に入力しようとしたり、テキスト ボックスの右端より右または左端より左にあるテキストを選択しようとする、自動的に横方向にスクロールします。
縦スクロール バー	テキスト ボックスの右側に縦のスクロール バーを表示します。
横スクロール バー	テキスト ボックスの下に横のスクロール バーを表示します。

[日付/時刻] タブ

次の表に、[日付/時刻] タブのプロパティを示します。

プロパティ	説明
日付	このチェック ボックスをオンにすると、[日付書式] リストが有効になり、ここから日付形式を選択できます。
時刻	このチェック ボックスをオンにすると、[時刻書式] リストが有効になり、ここから適切な時刻書式を選択できます。
日付書式	日付情報を表示するために使用できる書式が一覧表示されます。リストから目的の書式を選択します。
時刻書式	時間情報を表示するために使用できる書式が一覧表示されます。リストから目的の書式を選択します。

メモ: ClearQuest では、[日付] チェック ボックスをオンにすると、テキスト ボックス コントロールの右側にボタンが追加されます。このボタンをクリックすると、フィールドに入力する日付を選択するためのカレンダーが表示されます。

[コンテキスト メニュー フック] タブ

[コンテキスト メニュー フック] タブには、使用できるレコードフックと、このコントロールに現在関連付けられているフックのリストが表示されます。コントロールにフックを関連付けるには、[使用可] ボックスからフックを選択し、[追加] をクリックします。フックの関連付けを解除するには、[選択] ボックスからフックを選択し、[削除] をクリックします。このコントロールに関連付けられたフックは、このコントロールのショートカット メニューに表示されます。ユーザーがフックを実行するには、コントロールを右クリックし、表示されるショートカット メニューからフック名を選択する必要があります。

Rational ClearQuest には、コマンド行で起動できる多数のユーティリティが用意されています。これらのコマンド行ユーティリティは、特定タイプの管理タスクを実行するための高機能なツールです。

この章の内容は、次のとおりです。

- 381 ページの「コマンド行ユーティリティの dbset パラメータ」
- 381 ページの「installutil の使用法」
- 388 ページの「cqload によるスキーマのコピー」
- 393 ページの「importutil による動的リストのインポートとエクスポート」
- 395 ページの「bkt_tool によるユーザー データベース間のワークスペース項目の移動」
- 398 ページの「UNIX での cqtool を使用した夜間レポートの実行」

コマンド行ユーティリティの dbset パラメータ

-dbset パラメータは、必要に応じて多くの ClearQuest コマンド行ユーティリティで使用できます。ClearQuest システムに複数のスキーマ リポジトリがある場合は、-dbset と接続名を指定してコマンドを実行します。-dbset を省略した場合、コマンド行ツールは、デフォルトの dbset 名を使用します。このデフォルトの dbset 名は、製品のバージョン番号 (たとえば、2003.06.10) です。

installutil の使用法

installutil コマンド行ユーティリティには多数のサブコマンドがあります。これらのサブコマンドは、データベースの設定と変更に使えます。ここでは、いくつかの共通 installutil サブコマンドについて説明します。

多くの installutil サブコマンドのオプションと引数は、表 15 に示す特定のデータベース製造元に特化しています。installutil サブコマンドのすべてのパラメータは、表 15 に明記されていない限り必須です。また指定された順序に従って入力する必要があります。NULL 値を入力する必要のあるオプションは "" で表します。入力も同じ値を使用します。

表 15 製造元の種類別のサンプル データ

パラメータ	Access	SQL Anywhere	MS SQL	Oracle	DB2
db_vendor	MS_ACCESS	SQL_ANYWHERE	SQL_SERVER	ORACLE	DB2
server	""	データベース サービス名	データベース サーバー ホ スト名	SQL*Net エイリ アス	""
database	スキーマ リ ポジトリまた はユーザー データベース のフル パス	スキーマ リ ポジトリまた はユーザー データベース のフル パス	スキーマ リ ポジトリまた はユーザー データベース の名前	スキーマ リポジ トリまたはユー ザー データベー スの名前	スキーマ リポジトリ またはユー ザー データ ベースの 別名
dbo_login	admin	admin	データベース 所有者の ユーザー名	データベース 所有者の ユーザー名	データベー ス所有者の ユーザー名
dbo_password	admin	admin	データベース 所有者の パスワード	データベース 所有者の パスワード	データベー ス所有者の パスワード
rw_login	admin	admin	データベース 所有者の ユーザー名	データベース 所有者の ユーザー名	データベー ス所有者の ユーザー名
rw_password	admin	admin	データベース 所有者の パスワード	データベース 所有者の ユーザー名	データベー ス所有者の パスワード
ro_login	admin	admin	データベース 所有者の ユーザー名	データベース 所有者の ユーザー名	データベー ス所有者の ユーザー名
ro_password	admin	admin	データベース 所有者の パスワード	データベース 所有者の ユーザー名	データベー ス所有者の パスワード

表 15 製造元の種類別のサンプル データ

パラメータ	Access	SQL Anywhere	MS SQL	Oracle	DB2
connect_ options	""	SERVER_ VER=server_ version_ number	""	HOST=host; SID=sid; SERVER_ VER=server_ version_number; CLIENT_ VER=client_ version_number; LOB_ TYPE=data_ type	""

データベース コピーの作成

アップグレード プロセスをテストするためにスキーマ リポジトリをコピーし、サンプル データベースを作成するには、**installutil convertschemarepo** を実行します。このサブコマンドを実行すると、次の処理が行われます。

- オリジナル データベースが新しいデータベースとしてコピーされます。
- **ClearQuest** をデータベース コピーに接続します。
- オリジナル データベースをロックします。

メモ: このコマンドを実行する前に、データベース製造元のツールを使用してデータベースをバックアップし、新しいデータベースを作成してください。

データベースのコピーを作成して、アップグレード プロセスをオフラインでテストするには

- 1 新しいテスト データベースを作成します。詳細については、『**Rational Software** サーバー製品インストール ガイド』を参照してください。
- 2 **installutil convertschemarepo** を使用して既存のスキーマ リポジトリをコピーし、そのコピーを **ClearQuest** に接続します。製造元の種類別の入力パラメータのサンプルの詳細については、384 ページの「**installutil convertschemarepo**」を参照してください。
- 3 **installutil convertuserdb** を使用して既存の各ユーザー データベースをコピーし、そのコピーを **ClearQuest** に接続します。製造元の種類別の入力パラメータのサンプルの詳細については、385 ページの「**installutil convertuserdb**」を参照してください。

installutil convertschemarepo

installutil convertschemarepo [-dbset dbset_name] cq_login cq_password to_db_vendor
to_server to_database to_dbo_login to_dbo_password to_rw_login to_rw_password to_ro_login
to_ro_password connect_options [tcpip,ipx,netbios,namedpipes] [host1,host2,...]

パラメータ	説明
dbset_name	既存の接続の名前。
cq_login	コピーする ClearQuest データベースに対するユーザーのログイン名。このユーザーには、スーパー ユーザー権限が必要です。
cq_password	ClearQuest スーパー ユーザーのログイン パスワード。
to_db_vendor	新しくアップグレードする製造元データベース名。製造元固有の値の詳細については、382 ページの表 15 の database エントリを参照してください。
to_server	新しくアップグレードするデータベース サーバー名。
to_database	新しくアップグレードするデータベース名。
to_dbo_login	空のデータベース所有者のログイン名。
to_dbo_password	空のデータベース所有者のログイン パスワード。
to_rw_login	空のデータベースに対する読み取り / 書き込みユーザーのログイン名。
to_rw_password	空のデータベースに対する読み取り / 書き込みユーザーのログイン パスワード。
to_ro_login	空のデータベースに対する読み取り専用ユーザーのログイン名。
to_ro_password	空のデータベースに対する読み取り専用ユーザーのログイン パスワード。
connect_options	製造元固有の値の詳細については、382 ページの表 15 の connect_options エントリを参照してください。
[tcpip,ipx,netbios,namedpipes]	これらのパラメータは、ネットワークを指定するパラメータで、指定は必須ではありません。
[host1,host2,...]	これらのパラメータは、複数のホスト コンピュータを識別するパラメータで、指定は必須ではありません。

installutil convertuserdb

```
installutil convertuserdb [-dbset dbset_name] cq_login cq_password user_dbname  
to_user_db_vendor to_user_server to_user_database to_user_dbo_login to_user_dbo_password  
to_user_rw_login to_user_rw_password connect_options [tcpip,ipx,netbios,namedpipes]  
[host1,host2,...]
```

パラメータ	説明
dbset_name	既存の接続の名前。
cq_login	コピーする ClearQuest データベースに対するユーザーのログイン名。このユーザーには、スーパー ユーザー権限が必要です。
cq_password	ClearQuest スーパー ユーザーのログイン パスワード。
user_dbname	変換するユーザー データベース名。
to_user_db_vendor	新しくアップグレードする製造元データベース名。製造元固有の値の詳細については、382 ページの表 15 の database エントリを参照してください。
to_user_server	新しくアップグレードするユーザー データベース サーバー名。
to_user_database	新しくアップグレードするユーザー データベース名。
to_user_dbo_login	空のユーザー データベースの所有者のログイン名。
to_user_dbo_password	空のユーザー データベースの所有者のログイン パスワード。
to_user_rw_login	空のユーザー データベースに対する読み取り / 書き込みユーザーのログイン名。
to_user_rw_password	空のユーザー データベースに対する読み取り / 書き込みユーザーのログイン パスワード。
connect_options	製造元固有の値の詳細については、382 ページの表 15 の connect_options エントリを参照してください。
[tcpip,ipx,netbios,namedpipes]	これらのパラメータは、ネットワークを指定するパラメータで、指定は必須ではありません。
[host1,host2,...]	これらのパラメータは、複数のホスト コンピュータを識別するパラメータで、指定は必須ではありません。

データベースのロック解除

installutil を使用してデータベース コピーを作成する場合は、その前に **convertschemarepo** サブコマンドを使用して実動データベースをロックします。コピーを使用してテストする間も実動データベースを使用できるようにするには、実動データベースのロックを解除する必要があります。

メモ: アップグレード処理が完了した後は、オリジナル データベースのロックを解除しないでください。オリジナル データベースをロックしたままにしておくと、ユーザーが間違ったデータベースにアクセスしないようにできます。

- スキーマ リポジトリのロックを解除するには、次の **installutil** コマンドを入力し、**database** パラメータでデータベース コピーを参照します。

```
installutil unlockschemarepo [-dbset dbset_name] db_vendor server database dbo_login  
dbo_password connect_options
```

- ユーザー データベースのロックを解除するには、次の **installutil** コマンドを入力し、**database** パラメータで、スキーマ リポジトリではなくオリジナル データベースを参照します。

```
installutil unlockuserdb [-dbset dbset_name] db_vendor server database dbo_login  
dbo_password connect_options
```

製造元固有のオプションと引数の詳細については、382 ページの表 15 を参照してください。

例

ここでは、スキーマ リポジトリとユーザー データベースのコピーを作成してから、オリジナルのロックを解除する **installutil** コマンド行の例を示します。このようなコマンドは、テスト用のスキーマ リポジトリとユーザー データベースのコピーを作成し、オリジナルのスキーマ リポジトリとユーザー データベースをそのまま使用する処理で、一般的に使用されます。この例のデータベース製造元には、**Oracle** と **SQL Server** が含まれます。

Oracle

この例では、次の状態を前提としています。

- DBSet の既存のスキーマ リポジトリにはデフォルト名が付いています (そのため、**-dbset** オプションを指定する必要がありません)。
- 既存の **ClearQuest** ユーザー データベースは **MyUserDB** です。このデータベースに対するデータベース所有者の **Oracle** のユーザー ID は **udbo** で、パスワードは **P** です。
- 既存の **ClearQuest** スキーマ リポジトリは **MySR** です。このスキーマ リポジトリに対するデータベース所有者の **Oracle** のユーザー ID は **srdbo** で、パスワード **P** はです。
- すべてのデータベースとスキーマ リポジトリに対するデフォルトの **ClearQuest** スーパーユーザー ID は **admin** で、パスワードは **""** です。

- Oracle TNS 名 MYTNS がデータベースのコピー先の Oracle サーバーを指します。
- Oracle ユーザー ID srusr と dbusr は、パスワードが両方とも P で、ターゲット Oracle サーバーに登録されています。これらのユーザーには CONNECT 役割と RESOURCE 役割が付与されています。
- Oracle サーバーのホスト名は DBServ で、SID (Oracle システム識別子) は ORC です。
- Oracle 8.1 を Oracle データベース サーバーとクライアント ワークステーションで使用しています。

スキーマ リポジトリをコピーするには

```
installutil convertschemarepo admin "" ORACLE MYTNS srusr srusr P srusr P srusr P¥
"HOST=DBServ;SID=ORC;CLIENT_VER=8.1;SERVER_VER=8.1;LOB_TYPE=LONG"
```

ユーザー データベースをコピーするには

```
installutil convertuserdb admin "" ORACLE MYDB MYTNS dbusr dbusr P dbusr P ¥
"HOST=DBServ;SID=ORC;CLIENT_VER=8.1;SERVER_VER=8.1;LOB_TYPE=LONG"
```

コピー処理の完了後、オリジナルのロックを解除するには

```
installutil unlockschemarepo ORACLE MYTNS MySR srdb P¥
"HOST=DBServ;SID=ORC;CLIENT_VER=8.1;SERVER_VER=8.1;LOB_TYPE=LONG"
```

```
installutil unlockuserdb ORACLE MYTNS MyUserDB udbo P¥
"HOST=DBServ;SID=ORC;CLIENT_VER=8.1;SERVER_VER=8.1;LOB_TYPE=LONG"
```

SQL Server

この例では、次の状態を前提としています。

- DBSet の既存のスキーマ リポジトリにはデフォルト名が付いています (そのため、-dbset オプションを指定する必要がありません)。
- 既存の ClearQuest ユーザー データベースは MYDB です。
- デフォルトの ClearQuest スーパー ユーザー ID は admin で、パスワードは "" です。
- ターゲット SQL Server ホスト名は MySQLServer です。
- この SQL Server データベースはスキーマ リポジトリの新しいコピーを保持します。データベース名は MyNewSchemaRepo です。このデータベースは、SQL Server ユーザー ID dbowner、パスワード P で所有されています。
- この SQL Server データベースはユーザー データベースの新しいコピーを保持します。データベース名は MyNewUserDB です。このデータベースも SQL Server ユーザー ID dbowner で所有されています。

スキーマ リポジトリをコピーするには

```
installutil convertschmarepo admin "" SQL_SERVER ^  
MySQLServer MyNewSchemaRepo dbowner P dbowner P dbowner P ""
```

ユーザー データベースをコピーするには

```
installutil convertuserdb admin "" MYDB SQL_SERVER ^  
MySQLServer MyNewUserDB dbowner P dbowner P ""
```

コピー処理の完了後、オリジナルのロックを解除するには

```
installutil unlockschmarepo admin "" MYDB SQL_SERVER ^  
MySQLServer MyNewSchemaRepo dbowner P ""
```

```
installutil unlockuserdb admin "" MYDB SQL_SERVER ^  
MySQLServer MyNewUserDB dbowner P ""
```

cqload によるスキーマのコピー

cqload コマンド行ユーティリティを使用すると、スキーマ全体をスキーマ リポジトリにコピーしたり、スキーマを部分的に別のスキーマにコピーすることができます。

cqload を使用するための準備

cqload を適用するスキーマが、スキーマ リポジトリにチェックインされていることを確認します。スキーマがチェックアウトされている場合は、チェックインします。

メモ: ClearQuest Designer の動作中に cqload を実行した場合、変更を確認するには、ClearQuest Designer を終了し、再度ログインします。

スキーマ リポジトリへのスキーマのコピー

- 1 コマンド `exportschema` を使用して、コピーするスキーマ (すべてのリビジョン) をエクスポートします。389 ページの「`exportschema`」を参照してください。
- 2 コマンド `importschemata` を使用して、コピーするスキーマをスキーマ リポジトリにインポートします。390 ページの「`importschemata`」を参照してください。

既存のスキーマへの部分的なスキーマのコピー

- 1 コマンド `exportintegration` を使用して、スキーマの一部 (1 つ以上のリビジョン) をエクスポートします。391 ページの「`exportintegration`」を参照してください。
- 2 コマンド `importintegration` を使用して、コピーするスキーマを既存のスキーマにインポートします。392 ページの「`importintegration`」を参照してください。

exportschema

exportschema サブコマンド (cqload コマンド行ユーティリティの一部) を使用して、スキーマ全体をテキスト ファイルにエクスポートします。次に、importschema サブコマンドを使用して、スキーマを別のスキーマ リポジトリにインポートします。

部分的なスキーマをエクスポートするには、exportintegration サブコマンドを使用します。

構文

cqload exportschema [-dbset name] login password schema_name "schema pathname"

パラメータ	説明
-dbset name	スキーマの接続名を指定します。システムに複数のスキーマ リポジトリがある場合に必要です。
login	ユーザーの ClearQuest ログイン名。このユーザーには、スーパーユーザー権限が必要です。
password	ユーザーの ClearQuest パスワード。パスワードが設定されていない場合は、二重引用符のみの空のパラメータを入力します。
schema_name	テキスト ファイルにエクスポートされる、スキーマ リポジトリ内のスキーマ名。たとえば、TeamTest など。
schema_pathname	保存されるスキーマのテキスト ファイル バージョンです。パス名を二重引用符で囲みます。

例

cqload exportschema -dbset 2003.06.10 admin "" DefectTracking "c:¥schema.txt"

この例では、DefectTracking スキーマの内容がファイル c:¥schema.txt にエクスポートされます。

importschema

importschema サブコマンド (cqload コマンド行ユーティリティの一部) を使用して、スキーマ全体をテキスト ファイルからインポートし、スキーマ リポジトリに追加します。このサブコマンドは、スキーマ リポジトリにアクセスできないサイトや別のスキーマ リポジトリを使用できないサイトとスキーマを共有する場合に便利です。

importschema を使用する前に、exportschema コマンドを使用してスキーマをエクスポートしてください。部分的なスキーマをインポートするには、importintegration サブコマンドを使用します。

構文

```
cqload importschema [-dbset name] login password "schema_pathname"
```

パラメータ	説明
-dbset name	スキーマの接続名を指定します。システムに複数のスキーマ リポジトリがある場合に必要です。
login	ユーザーの ClearQuest ログイン名。このユーザーには、スーパー ユーザー権限が必要です。
password	ユーザーの ClearQuest パスワード。パスワードが設定されていない場合は、二重引用符のみの空のパラメータを入力します。
schema_pathname	exportschema サブコマンドで保存されたスキーマのテキスト表現が格納されているファイルのパス名です。パス名を二重引用符で囲みます。

例

```
cqload importschema -dbset 2003.06.10 admin "" "c:¥schema.txt"
```

この例では、テキスト形式でエクスポートされ、c:¥schema.txt に保存されているスキーマが現在のスキーマ リポジトリにインポートされます。

メモ: c:¥schema.txt は、cqload exportschema コマンドを使用して作成されています。その処理中に、エクスポートされたスキーマ名が、このファイルに保存されています。そのため、このスキーマをインポートすると、そのスキーマ名を使用して、cqload importschema でスキーマが作成されます。その名前がスキーマ リポジトリで既に使用されている場合、インポートは失敗します。

exportintegration

cqload exportintegration を使用して、スキーマの特定のバージョンを、既存のスキーマの変更に使用できる形式でエクスポートします。データを別のスキーマにインポートするには、importintegration サブコマンドを使用します。スキーマのエクスポートとインポートには、類似のレコードタイプを使用する必要があります。

構文

cqload exportintegration [-dbset name] login password schema_name begin_rev end_rev
[record_type_to_rename] "schema_pathname"

パラメータ	説明
-dbset name	スキーマの接続名を指定します。システムに複数のスキーマ リポジトリがある場合に必要です。
login	ユーザーの ClearQuest ログイン名。このユーザーには、スーパーユーザー権限が必要です。
password	ユーザーの ClearQuest パスワード。パスワードが設定されていない場合は、二重引用符のみの空のパラメータを入力します。
schema_name	統合に対応するスキーマの名前です。スキーマ名では、大文字と小文字は区別されます。
begin_rev	変更を保存する最初のスキーマのリビジョンです。
end_rev	変更を保存する最後のスキーマのリビジョンです。
record_type_to_rename	cqload importintegration でインポート時に名前を変更するスキーマのレコード タイプの名前 (オプション)。このオプションを省略する場合は、このパラメータの位置に NULL オプション "" を指定してください。
schema_pathname	スキーマのリビジョンをエクスポートした結果が格納されるファイルのパス名です。パス名を二重引用符で囲みます。

例

cqload exportintegration -dbset 2003.06.10 admin "" Enterprise 5 5 "" ^"
c:\tmp\export.txt"

この例では、Enterprise スキーマのバージョン 5 に加えられた変更のみがエクスポートされます。

cqload exportintegration -dbset 2003.06.10 admin " Enterprise 5 8 "" ^"
c:\tmp\export.txt"

この例では、Enterprise スキーマのバージョン 5 ～ 8 に加えられた変更がエクスポートされます。

```
cqload exportintegration -dbset 2003.06.10 admin "" Enterprise 5 5 ^
ChangeRequest "c:¥temp¥scriptchanges.txt"
```

この例では、Enterprise スキーマのバージョン 5 に加えられた変更のみがエクスポートされ、レコード タイプ ChangeRequest の名前がインポート時に変更されることが指定されます。この後の cqload importintegration コマンドの new_record_type_name オプションで、レコードの新しい名前を指定する必要があります。

importintegration

cqload importintegration サブコマンドを使用して、cqload exportintegration でエクスポートされたスキーマの変更内容をインポートします。

メモ： ClearQuest MultiSite を使用している場合、importintegration は作業マスター サイトでのみ実行できます。

構文

```
cqload importintegration [-dbset name] login password schema_name
[ new_record_type_name ] "integration name" integration_version "schema_pathname"
"form_name"
```

パラメータ	説明
-dbset name	スキーマの接続名を指定します。システムに複数のスキーマ リポジトリがある場合に必要です。
login	ユーザーの ClearQuest ログイン名。このユーザーには、スーパーユーザー権限が必要です。
password	ユーザーの ClearQuest パスワード。パスワードが設定されていない場合は、二重引用符のみの空のパラメータを入力します。
schema_name	統合に対応するスキーマの名前です。この名前は、スキーマ リポジトリで定義されているスキーマの名前です (たとえば、TeamTest など)。このスキーマは、cqload の実行後にリビジョン番号が 1 つずつ増えます。
new_record_type_name	オプション。cqload exportintegration コマンドの record_type_to_rename オプションで指定したレコード タイプの名前が、新しいスキーマにインポートされたときに変更されます。このオプションを省略する場合は、このパラメータの位置に NULL オプション "" を指定してください。
integration_name	統合に指定する名前です。この名前は、ロード中の統合に対する英数字のインジケータになる場合があります。

パラメータ	説明
integration_version	ロードするバージョンのことです。これには、数字を指定します。
schema_pathname	exportintegration で作成されたファイルのパス名。パス名を二重引用符で囲みます。
form_name	統合で作成された新しいタブが追加されるフォームです。フォームの更新がない場合は、"" を入力して、更新するフォームがないことを示します。

例

```
cqlload importintegration -dbset 1.1 admin "" Testit "" Email_Integ 1 ^
"c:¥tmp¥export.txt" ""
```

この例では、エクスポートされたスキーマを **Testit** スキーマにインポートしますが、すべてのレコードタイプを元の名前のままにします。

```
cqlload importintegration -dbset 1.1 admin "" Testit Defect Email_Integ 1 ^
"c:¥tmp¥export.txt" ""
```

この例では、エクスポートされたスキーマを **Testit** スキーマにインポートし、直前の **cqlload exportintegration** コマンドの **record_type_to_rename** オプションで指定したレコードタイプの名前を **Defect** に変更します。

importutil による動的リストのインポートとエクスポート

importutil コマンド行ユーティリティを使用すると、テキスト ファイルから動的リストのインポート/エクスポートを実行できます。この処理は、次の **importutil** サブコマンドを適用して行います。

- **importlist**: ユーザー データベースにリストをインポートします。
- **exportlist**: ユーザー データベースからリストをエクスポートします。

エクスポートされた動的リストはテキスト ファイルです。必要に応じて、テキスト エディタを使用して内容を変更できます。

動的リストのエクスポート

exportlist サブコマンドを使用して、動的リストをテキスト ファイルにエクスポートします。

構文

```
importutil exportlist [-dbset name] login password dbname listname "output_file_name"
```

パラメータ	説明
-dbset name	スキーマの接続名を指定します。システムに複数のスキーマ リポジトリがある場合に必要です。
login	ユーザーの ClearQuest ログイン名。このユーザーには、スーパーユーザー権限が必要です。
password	ユーザーの ClearQuest パスワード。パスワードが設定されていない場合は、二重引用符のみの空のパラメータを入力します。
db_name	ユーザー データベースの論理名。
list_name	ユーザー データベース db_name に対する動的リストの名前。
output_file_name	エクスポートするファイルのパスと名前。パス名を二重引用符で囲みます。

例

```
importutil exportlist -dbset 2003.06.10 "admin" "" PUSER domains ^  
"C:¥temp¥dynamlist.txt"
```

この例では、domains 動的リストの内容をデータベース PUSER からエクスポートし、その内容をファイル C:¥temp¥dynamlist.txt に保存します。

動的リストのインポート

importlist サブコマンドを使用して、動的リストをユーザー データベースにインポートします。動的リストをインポートすると、同じ名前を持つ既存の動的リストの内容が上書きされます。

構文

```
importutil importlist [-dbset name] login password dbname listname "output_file_name"
```

パラメータ	説明
-dbset name	スキーマの接続名を指定します。システムに複数のスキーマ リポジトリがある場合に必要です。
login	ユーザーの ClearQuest ログイン名。このユーザーには、スーパーユーザー権限が必要です。

パラメータ	説明
password	ユーザーの ClearQuest パスワード。パスワードが設定されていない場合は、二重引用符のみの空のパラメータを入力します。
db_name	ユーザー データベースの論理名。
list_name	ユーザー データベース db_name に対する動的リストの名前。
input_file_name	インポートするファイルのパスと名前。パス名を二重引用符で囲みます。

例

```
importutil importlist -dbset 2003.06.10 "admin" "" PUSER domains ^
"C:¥temp¥dynalist.txt"
```

この例では、ファイル C:¥temp¥dynalist.txt の内容をインポートし、その内容を使用して domains という名前の動的リストをデータベース PUSER に作成します。

bkt_tool によるユーザー データベース間のワークスペース項目の移動

bkt_tool を使用して、共用フォルダのすべての内容を 1 つのユーザー データベースからほかのデータベースにコピーまたは移動します (共用フォルダは ClearQuest クライアント ワークスペースで使用可能です)。このユーティリティは、事前定義されたスキーマ Blank または Common をベースにして新しいスキーマを開発する場合や、クエリー、グラフ、レポートの操作を、すべての項目を作成しないで行う場合に、便利なユーティリティです。

このユーティリティは、テスト データベースを処理する場合にも便利です。たとえば、実動データベースでスキーマの変更を行う前に、テスト データベースでテスト用のクエリー、グラフ、レポートをテストする場合があります。このような場合は、bkt_tool を使用すると、すべての実動ワークスペース項目をテスト データベースにコピーまたは移動できます。

bkt_tool で使用可能なサブコマンドを実行すると、次の処理が行われます。

- 共用フォルダからすべてのワークスペース内容をファイルにエクスポートします (共用フォルダは ClearQuest クライアントで使用可能です)。
- すべてのワークスペース内容を別のユーザー データベースにインポートします。
- 指定したユーザー データベースに対して、ワークスペース内容を更新します。
- ユーザー データベースのワークスペース内容を削除します。

bkt_tool コマンド行オプション

すべての bkt_tool コマンドには、類似した構文とコマンド行オプションがあります。

```
bkt_tool option -dbset dbset -us "username" -password password -db "dbname" {-directory "pathname" | -storagefile "pathname"}
```

パラメータ	説明
option	-i (インポート)、-u (更新)、-e (エクスポート)、-d (削除)、-t (変換) のいずれか 1 つ。
dbset	スキーマの接続名を指定します。システムに複数のスキーマ リポジトリがある場合に必要です。
username	ユーザーの ClearQuest ログイン名。このユーザーには、スーパーユーザー権限が必要です。
password	ユーザーの ClearQuest パスワード。パスワードが設定されていない場合は、二重引用符のみの空のパラメータを入力します。
dbname	ワークスペース内容をエクスポートするユーザー データベースの論理名。
pathname	エクスポートするファイルのパス名。パス名を二重引用符で囲みます。 -directory を指定する場合に、option が -i または -u のときは、その読み込みを構成するさまざまなファイルが保存されている pathname を指定します。このディレクトリには、CONTENTS.CFG という名前のテキスト ファイルと、各ワークスペース項目のバイナリ データを格納する 0001.DAT から nnnn.DAT の一連のファイルが格納されます。 -storagefile を指定する場合、pathname は IStorage 複合ドキュメントのパス名です。

エクスポート

-e サブコマンドを使用して、共用フォルダのすべてのワークスペース内容を 1 つのユーザー データベースからほかのデータベースにエクスポートします (共用フォルダは ClearQuest クライアントで使用可能です)。

構文

```
bkt_tool -e -dbset dbset -us "username" -password password -db "dbname" {-directory "pathname" | -directory "pathname"}
```

例

```
bkt_tool -e -dbset 1.1 -us "admin" -password "" -db "CLSIC" -directory ^  
"d:¥temp¥bkt_test"
```

この例では、共用フォルダのワークスペース内容が、CLSIC ユーザー データベースからディレクトリ d:¥temp¥bkt_test にエクスポートされます。

インポート

-i サブコマンドを使用して、ワークスペース項目が格納された、エクスポートされたファイルを、ユーザー データベースにインポートします。ユーザー データベースに、クエリー、グラフ、レポートが既にある場合、bkt_tool コマンド行ユーティリティはエラーを表示し、インポート処理を続行することができません。

構文

```
bkt_tool -i -dbset name -us "username" -password password -db "dbname" -directory "pathname"
```

例

```
bkt_tool -i -dbset 1.1 -us "admin" -password "" -db "test1" -directory ^  
"d:¥temp¥bkt_test"
```

この例では、d:¥temp¥bkt_test の内容を test1 ユーザー データベースにインポートします。

更新

-u サブコマンドを使用して、ユーザー データベースに対して、ClearQuest クライアントの既存のワークスペース項目を更新します。ユーザー データベースに、クエリー、グラフ、レポートが既にある場合、bkt_tool コマンド行ユーティリティは既存の項目を最新のデータで上書きします。

構文

```
bkt_tool -u -dbset dbset -us "username" -password password -db "dbname" -directory "pathname"
```

例

```
bkt_tool -u -dbset 1.1 -us "admin" -password "" -db "test1" -directory "d:¥temp¥bkt_test"
```

この例では、ディレクトリ d:¥temp¥bkt_test から test1 ユーザー データベースを更新します。

UNIX での cqtool を使用した夜間レポートの実行

一般的な関心事として、さまざまなレポートを夜間に実行する方法がないかということが挙げられます。これは通常、電子メール通知の量と密接な関係があります。ここでは **cqtool** を使用した例を示します。これは UNIX 上の **ClearQuest** コマンド行インターフェイスであり、**Submitted** 状態の障害を表示する特別なクエリーを動的に作成し、実行します。この例の詳細については、次のコマンドを実行して参照してください。

```
man cqtool or cqtool new_query -man
```

cqtool の使用には、次の 3 つの基本的な要素があります。

- データベースへのログイン
- 実行する一連のコマンドの連結
- 出力の決定

すべての例では、サンプル ユーザー データベースを使用します。

サンプル ユーザー データベースへのログインは通常、データベース名は **SAMPL**、ユーザー ID は **admin**、パスワードは空白で実行します。次のように、コマンド行で **cqtool login** を実行すると、コマンド行シェルが開始され、UNIX 上の **ClearQuest** と対話形式で処理を行うことができます。

```
cqtool login -database SAMPL -user admin -password ""
```

ほかのコマンドをバッチ モードで実行するには、**login** を、実行するコマンドで置き換えます。この例では、**cqtool** を使用して新しいクエリーを作成し、実行します。これは、**new_query** コマンドを使用して実行します。**new_query** コマンドには、フィールド表示とフィルタ操作のいくつかのパラメータがあります。フィールドは **-field fieldid** オプションを使用して表示し、フィルタは **-filterop** 変数の値を使用して実行します。この例では、すべての **Submitted** 状態の障害について、**id**、**headline**、**submitter** の各フィールドを表示します。

```
cqtool new_query -type defect -field id -field headline -field submitter -eq state ^  
submitted -database SAMPL -user admin -password ""
```

使用するデータベースの指定を要求するメッセージが表示されます。

Master: **SAMPL**

データベースを指定すると、クエリー結果が返されます。以下は、サンプル データベースの場合の結果です。

```
idHeadlineSubmitter
```

```
-----
```

```
SAMPL00000011change due amount is supposed to be redengineer  
SAMPL00000012would like logout button to be largereengineer  
SAMPL00000016too many spaces in "change due" fieldlead  
SAMPL00000019sales tax incorrect for NHlead  
SAMPL00000021inventory report is not running correctlylead  
SAMPL00000024overriding price operation allows negative numberQE
```

SAMPL00000027add item button is out of line with the other buttonsQE
SAMPL00000028context sensitive help fails from reorder windowQE
SAMPL00000029formatting does not look right in inventory reportQE
SAMPL00000030add items fails for large quantitiesQE
SAMPL00000032shortcut to logout does not workQE
SAMPL00000033unable to add item already in sale listQE
SAMPL00000034cancel sale leaves ite in purchase listengineer
SAMPL00000036inventory report is displaying an empty columnengineer
SAMPL00000037need report for items ordered on a given dayengineer
SAMPL00000038sales tax amount is offset from labelengineer
SAMPL00000039need automatic logout with QEeoutengineer
SAMPL00000040spelling error in help for override priceengineer
Count: 22

最後に、`-output_file filename` パラメータ を使用して出力先を指定できます。たとえば、これを使用して、出力を管理者にメールで送信できます。以上の処理をすべて実行するには、次のように指定します。

```
cqtool new_query -type defect -field id -field headline -field submitter -eq state ¥  
submitted -database SAMPL -user admin -password "" -output_file ¥  
/tmp/cqoutput  
mail cqadmin@yourco.co.jp < /tmp/cqoutput  
rm /tmp/cqoutput
```


データベース / トラブル シューティング ガイド

F

Rational ClearQuest は、さまざまなオペレーティング システムとデータベースに対応しています。ここでは、データベースとデータベース接続に影響を与える、実行時の問題を診断するさまざまな手法について説明します。

UNIX 環境での Oracle との接続に関する問題

ここでは、共通のエラー メッセージ タイプと、UNIX が稼働する ClearQuest ホストでのコマンドとその結果表示されるエラー メッセージの例を示し、詳細な分析のための推奨事項について説明します。また、UNIX での Oracle との接続で発生する問題に対する、一般に有益な解決法についての情報も記載されています。

データベースの接続オプション

Oracle データベースに接続する場合、ClearQuest ではデータベース プロパティ `connect_options` を使用して、特定の構成でのクライアントの動作を指定します。すべてのクライアントに対するデフォルトの `connect_options` は、スキーマ リポジトリが作成されたときに確立され、スキーマ リポジトリに格納されて、接続時に各クライアントに複製されます。ほかのバージョンの Oracle クライアント ソフトウェアが起動しているクライアントでは、デフォルトの `connect_options` を `installutil registeroracleoptions` コマンドを使用して上書きする必要があります。

ユーザーが特定のクライアントから Oracle データベースに接続できず、Oracle 対応の OpenLink ODBC ドライバのバージョンが Oracle クライアントの不正なバージョンを参照しているというエラー メッセージが表示された場合は、ホストの Oracle クライアント ソフトウェアをデフォルトの `connect_options` で指定されているバージョンにアップグレードするか、クライアントで `installutil registeroracleoptions` を実行してデフォルトの `connect_options` を上書きする必要があります。ただし、このエラー メッセージが表示されるのは、ログインのエラー メッセージを表示するダイアログ ボックスの [詳細] チェック ボックスをオンにした場合です。実行可能ファイル `installutil` は ClearQuest のインストール ディレクトリにあります。

以下の `installutil registeroracleoptions` コマンドは、このクライアント ホストが `CLIENT_VER` の値を 8.1 に設定した `connect_options` 文字列を使用することを指定します。クライアントで使用されるほかの `connect_options` 値は、スキーマ リポジトリに指定された値が使用されます。

メモ: この変更を行うには、コンピュータ管理者としてログインする必要があります。

```
installutil registeroracleoptions "CLIENT_VER=8.1"
```

このコマンドを使用すると、

`HKEY_LOCAL_MACHINE\Software\RationalSoftware\ClearQuest\2003.06.10\Core` にレジストリ キー `OverrideOracleConnectOptions` が作成され、値が `CLIENT_VER=8.1` に設定されます。

メモ: `ClearQuest` を再インストールすると、再インストールの過程でこのレジストリが削除される場合があります。各インストールの後に、クライアントにこれが設定されていることを確認してください。

`installutil` の詳細については、381 ページの「`installutil` の使用法」を参照してください。

Unknown host エラー

Unknown host エラーは、`ClearQuest` で、`OpenLink Request Broker` に接続しようとしたときに、ホストが接続文字列で指定されているにもかかわらず、ホスト名を IP アドレスに解決できないか、IP アドレスが存在しない場合に発生します。

この例では、IP アドレスに解決できないホスト名 (`badhost`) が接続文字列に記述されています。この種類のエラーは、`ClearQuest` スキーマ リポジトリの初期登録中または `ClearQuest` のログオン処理中にも発生する場合があります。

```
pdsql -v ora -s badhost:SID -u system -p manager -co "SERVER_VER=8.1"
```

```
OpenLink: RPC: Unknown host
EXCEPTION: [OpenLink] [ODBC]RPC: Unknown host
State: 08004 Native: 0
Connect String used: SVT=Oracle 8;
DRIVER=/files/a/rational/releases/ClearQuestClient.
2002.05.00/linux/shlib/db_ORACLE8;HOST=badhost; PROTOCOL=TCP/IP;
UID=admin; DATABASE=SID
```

Unknown host エラーを受け取った場合の解決法は、次のとおりです。

- 1 この操作で **ClearQuest** がアクセスしようとしていたホストを特定します。このホストは、入力したコマンドに指定したホストであるとは限りません。**connect_options** の **HOST** パラメータを確認します。この例では **badhost** です。

HOST パラメータに指定するホストは、データベース サーバーのホスト名と一致する必要があります。一致しなかった場合は、406 ページの「**ClearQuest** データベース設定の確認」の手順に従ってホスト名の記述元を特定してから、失敗した操作をやり直します。

- 2 指定されていたホストがデータベース サーバーのホスト名と一致する場合は、405 ページの「ホストへの **ping**」の手順に従って接続性を確認または修復してから、失敗した操作をやり直します。

Program unavailable エラー

Program unavailable エラーは、**ClearQuest** で、**connect_options** に指定されたホストの **OpenLink Request Broker** に接続しようとしたとき、宛先ホストは使用可能だったが、**OpenLink Request Broker** に接続できなかった場合に発生します。

この例では、名前 **goodhost** は IP アドレスに解決できますが、**goodhost** の **OpenLink Request Broker** に接続できません。この種類のエラーは、**ClearQuest** スキーマ リポジトリの初期登録中または **ClearQuest** のログオン処理中にも発生する場合があります。

```
pdsql -v ora -s goodhost:SID -u system -p manager -co "SERVER_VER=8.1"
```

```
OpenLink: RPC: Program unavailable
EXCEPTION: [OpenLink] [ODBC]RPC: Program unavailable
State: 08004 Native: 0
Connect String used: SVT=Oracle 8;
DRIVER=/files/a/rational/releases/ClearQuestClient.
2002.05.00/linux/shlib/db_ORACLE8;HOST=goodhost; PROTOCOL=TCP/IP;
UID=admin; DATABASE=SID
```

Program unavailable エラーを受け取った場合の解決法は、次のとおりです。

- 1 この操作で **ClearQuest** がアクセスしようとしていたホストを特定します。このホストは、入力したコマンドに指定したホストであるとは限りません。**Connect String** の **HOST** パラメータを確認します。この例では **goodhost** です。

HOST パラメータに指定するホスト名は、予期されたデータベース サーバー名と一致する必要があります。一致しなかった場合は、406 ページの「**ClearQuest** データベース設定の確認」の手順に従ってホスト名の記述元を特定してから、失敗した操作をやり直します。

- 2 指定されていたホストがデータベース サーバーの予期されたホスト名と一致する場合は、406 ページの「**OpenLink Request Broker** が開始しているかどうかの確認」の手順に従って接続性を確認または修復してから、失敗した操作をやり直します。

Oracle not available エラー

Oracle not available エラー (Oracle エラー コード ORA-01034) は、ClearQuest で、`connect_options` に指定されたホストの OpenLink Request Broker に接続しようとしたときに発生します。OpenLink Request Broker は使用可能ですが、データベース サーバーの Oracle インスタンスには接続できません。このエラーは通常、Oracle インスタンス ID (SID) が `connect_options` に適切に指定されていないか、Oracle データベースまたはそのリスナ プロセスがデータベース サーバーで開始していない場合に発生します。

この例では、`goodhost` をデータベース サーバー ホストの IP アドレスに解決できますが、`badSID` が有効な Oracle インスタンス ID ではありません。この種類のエラーは、ClearQuest スキーマ リポジトリの初期登録中または ClearQuest のログオン処理中にも発生する場合があります。

```
pdsql -v ora -s goodhost:badSID -u system -p manager -co "SERVER_VER=8.1"
```

```
OpenLink: [Oracle Server]ORA-01034: ORACLE not
available[SQLSTATE:S1000]
EXCEPTION: [OpenLink][ODBC] ORA-01034: ORACLE not available State:
S1000
Native: 0
Connect String used: SVT=Oracle 8;
DRIVER=/files/a/rational/releases/ClearQuestClient.
2002.05.00/linux/shlib/db_ORACLE8;HOST=goodhost; PROTOCOL=TCP/IP;
UID=admin; DATABASE=badSID
```

Oracle not available エラーを受け取った場合の解決法は、次のとおりです。

- 1 この操作で ClearQuest がアクセスしようとしていたホストを特定します。このホストは、入力したコマンドに指定したホストであるとは限りません。`connect_options` の `HOST` パラメータを確認します。この例では `goodhost` です。

`HOST` パラメータに指定するホストは、データベース サーバーのホスト名と一致する必要があります。一致しなかった場合は、406 ページの「ClearQuest データベース設定の確認」の手順に従ってホスト名の記述元を特定してから、失敗した操作をやり直します。

- 2 指定されていたホストがデータベース サーバーの予期されたホスト名と一致する場合は、407 ページの「Oracle の接続性の確認」の手順に従って Oracle インスタンスの `SID` と現在の状態を確認してから、失敗した操作をやり直します。

診断処理

ここでは、UNIX 上の Oracle の接続性に関する問題の解決に必要ないくつかの共通診断処理について説明します。

ホストへの ping

UNIX への ClearQuest の 接続性では、データベースの設定時に指定された特定のホストにアクセスできるかどうか重要です。UNIX の **ping** ユーティリティを使用して、指定されたホストに指定された名前で接続できるかどうかを確認できます。

- 1 特定のリモート ホストへの接続性が必要な UNIX ホストで、次のコマンドを入力します。

```
ping remote_host_name
```

ping が現在のユーザーのパスに存在しない場合は、`/usr/sbin/ping` などを探します。

- 2 クライアント オペレーティング システムによっては、**ping** は正常に処理され、次の結果が返されます。

```
remote_host_name is alive
```

または、ある個数のバイトがホストから返されたことを示すメッセージが表示されます。

- 3 **ping** でホストに接続できなかった場合は、次のような応答が表示されます。

a Unknown Host remote_host_name

この応答は、現在のホストのネームサーバーが、指定された **remote_host_name** を認識していないことを示します。リモート ホストが **Windows** の場合、UNIX クライアントから受信したネーム解決をサポートするよう、サイトのネームサーバーを設定する必要があります。サイトの IT 部署に連絡して、この問題を解決するための支援を受けてください。

b No answer from remote_host_name or Destination Host Unreachable

remote_host_name からの応答がない場合、宛先ホスト名は現在のホストに認識されていますが、ネットワークの設定または目的のホストの可用性が原因でネットワークに接続性の問題があります。サイトの IT 部署に連絡して、この問題を解決するための支援を受けてください。

ClearQuest データベース設定の確認

UNIX の場合、ClearQuest では、管理者が [Register Databases] ダイアログ ボックスで入力した情報を使用して、スキーマ リポジトリへの最初の接続を行います。次に、スキーマ リポジトリのデータベース接続情報とすべてのユーザー データベースがクライアントにダウンロードされ、ClearQuest データベース ディレクトリに保存されます。UNIX クライアントにコピーされる接続情報は、データベースを作成したときに ClearQuest メンテナンス ツールを使用して指定された初期情報、または後で ClearQuest Designer のデータベース プロパティ変更機能を使用して変更された情報です。

これらの設定が正しいことを確認するには

- 1 ClearQuest クライアントで、次のコマンドを入力します。

```
cqreg show
```

これにより、スキーマ リポジトリとユーザー データベース プロパティについて ClearQuest クライアントに存在するすべての情報が表示されます。

- 2 Oracle データベースの場合は、`connect_options` 文字列の `HOST` パラメータがデータベース サーバーのホスト名と一致することを確認します。このホスト名には、ClearQuest クライアントがアクセス可能である必要があります (405 ページの「ホストへの ping」を参照)。Server パラメータは、Windows クライアント用に指定されている SQLNet エイリアス名です。UNIX クライアントでは、これは使用されず、無視できます。
- 3 情報が正しくなかった場合、または古かった場合は、次の処理を実行します。
 - a ClearQuest メンテナンス ツールのスキーマ リポジトリのプロパティと ClearQuest Designer のユーザー データベースのプロパティを確認します。プロパティ項目の不一致を解決します。変更を行った場合は、ユーザー データベースをアップグレードします。
 - b 次のコマンドを ClearQuest コマンド行から実行します (ClearQuest では、スキーマ リポジトリからのデータベース接続情報を自動的に更新しないので、この操作が必要です)。

```
cqreg refresh
```

OpenLink Request Broker が開始しているかどうかの確認

ClearQuest では、Oracle データベースとの通信に OpenLink Request Broker を使用します。OpenLink ドライバが OpenLink Request Broker に接続すると、OpenLink Request Broker はデータベース製造元専用のエージェントを起動して実際の依頼を処理します。この機能を正常に動作させるには、OpenLink Request Broker がデータベース サーバーのホストで開始しており、使用できることが必要です。

OpenLink Request Broker が開始しているかどうかを確認するには

- 1 Oracle Database サーバーから、次のコマンドを入力します。

```
ps -e | grep oplrqb
```

名前に文字列 **oplrqb** を含む、開始中のすべてのプロセスが表示されます。これは、OpenLink Request Broker のプロセス名です。実行されていなかった場合、開始することとシステム起動時の実行を有効にすることが必要です。

- 2 必要に応じて、OpenLink がインストールされているディレクトリから次のコマンドを実行し、OpenLink Request Broker を開始します。

```
oplrqb start/stop/start
```

オプション **S** を選択します。これにより Openlink Request Broker が開始します。OpenLink Request Broker がシステム起動時に自動的に実行されるように設定するには、Windows の場合は『Rational Software サーバー製品インストールガイド』を、UNIX の場合は『Rational ClearCase 製品ファミリー インストールガイド』を参照してください。

Oracle の接続性の確認

ClearQuest では、Oracle データベースとの通信に OpenLink Request Broker を使用します。OpenLink Request Broker は、ほかの Oracle クライアントと同様に動作し、Oracle データベースにアクセスするための接続性と権限を必要とします。

Oracle の接続性を確認するには

- 1 OpenLink Request Broker を実行するユーザーとしてログオンします。
- 2 OpenLink インストール ディレクトリに移動します。
- 3 テキスト エディタを使用して OpenLink ルール ファイル **oplrqb.ini** を開きます。文字列 **generic_ora8** (Oracle 8.x を使用していることが前提です) を検索します。そこで、変数 **ORACLE_HOME** と **ORACLE_SID** を探します。これらの変数の設定が、大文字と小文字の区別も含めて、正しいことを確認します。
- 4 Oracle データベース サーバーで実行されている実際の **SID** を確認するには、次のコマンドを実行します。

```
ps -ef | grep pmon
```

フォーム **pmonnnnn** の文字列を含む行を探します。ここで、**nnnn** は **SID** を表します。

ClearQuest では、データベースを登録する場合、[Register Database] ダイアログ ボックスに指定された情報を使用して初期接続を開始します。これ以降の接続には、スキーマ リポジトリに格納されている情報が使用されます。したがって、接続オプション文字列を確認することが重要です。確認するには、ClearQuest Designer で [Database] メニューの [Database Properties] をクリックします。HOST、SID、SERVER_VER が正しいことを確認します。また、LOB_TYPE が LONG または CLOB に設定されていることを確認します。

その他の Oracle 関連の問題

ここでは、Oracle データベースを使用する場合に、ClearQuest に影響する可能性のあるその他の問題について説明します。

Contains 演算子を使用した検索で常に大文字と小文字が区別される

Oracle データベースを使用する場合、Contains 演算子を使用した検索では常に大文字と小文字が区別されます。

dbwin32 による電子メール通知に関する問題のデバッグ

ClearQuest は、レジストリ設定によって有効にされている場合、必要なデバッグ情報を Windows のデバッグ ログに書き込みます。この情報を表示するには、ClearQuest のインストール ディレクトリにある dbwin32 ツールまたは Windows のデバッグ ログを参照できるほかのツールを使用します。

- 1 次の情報を使用して、ネイティブ クライアントか Web サーバー コンピュータ、またはその両方に使用する .reg ファイルを作成します。

```
Native Client
REGEDIT4
[HKEY_CURRENT_USER¥Software¥Rational Software¥ClearQuest¥Diagnostic]
"Trace"="Email"
"Output"="ODS"
"EMailSendVB"="ODS"
Webserver
REGEDIT4
[HKEY_USERS¥.default¥Software¥Rational Software¥ClearQuest¥Diagnostic]
"Trace"="Email"
"Output"="ODS"
"EMailSendVB"="ODS"
```

- 2 ClearQuest を停止します。
- 3 レジストリに、手順 1 で作成した .reg ファイルをインポートします。
- 4 ClearQuest を起動します。

Windows 2000 で実行している場合、デバッグ出力を表示するには、ローカル管理者のアクセス許可が必要です。

索引

A

admin ユーザー、デフォルト ID 167
API
 共通呼び出し 229
 説明 10, 228

B

bkt_tool ユーティリティ 395

C

ccase-home-dir ディレクトリ xxviii
CharacterSetValidation パッケージ 39
ClearQuest Designer 21, 22, 147
ClearQuest データ コード ページ
 「データ コード ページ」を参照
cqload ユーティリティ 388
cqtool ユーティリティ 398
cquest-home-dir ディレクトリ xxviii

D

DB2 データベース 80

E

exportintegration サブコマンド 391
exportschema サブコマンド 389

I

importintegration サブコマンド 392
importschema サブコマンド 390
installutil ユーティリティ 381

M

Microsoft Access 77
MultiSite、データ コード ページ 41

O

Oracle データベース 79
 UNIX 接続のトラブルシューティング 401
 Web アクセスの設定 236
 検索のトラブルシューティング 408
 サポートされているデータ コード ページ 51
 ロック解除、コマンドの例 386

S

SQL Anywhere 77
SQL Server データベース 78
 コピー、コマンドの例 387

U

UNIX クライアント、データベース接続 61

W

Web アクセス

- ASP スクリプト 236
- インターフェイスのカスタマイズ 234
- 制限 233, 241
- 設定 235, 236
- 説明 233
- 電子メール通知 237, 255, 256
- 表示オプション 238, 239
- 表示オプション、事前定義 239, 240
- フォーム依存フィールド 207
- フックの動作 241, 242, 243

あ

アーキテクチャ

- UNIX 11
- Windows 8

アクションとアクション タイプ

- 削除 131
- サポート対象のリスト 126
- 状態モデルへの追加 128
- 説明 125
- デフォルト 130
- フック 129
- 変更 129

アクション フック

- 削除 217
- サポート対象のリスト 214, 215
- 実行順序 218, 219, 220
- 説明 214
- 追加 216
- 電子メール通知 257
- 変更 217
- 例、初期化 336

アップグレード機能 68

か

カスタマ サポート xxix

管理タスク

- 委任、必要な権限 20
- 説明 13
- ユーザーとグループ 167, 168

管理ツール 21

き

規則、表記 xxviii

機能レベル、向上 68

く

クエリー、作成 228

グローバル スクリプト

- 削除 227
- 作成 226
- 説明 225
- 変更 226

け

権限 20

- サポートされている権限のリスト 169
- フックに必要な権限 200
- 変更 177

こ

コマンド行ユーティリティ 381

し

状態タイプ

- スキーマへのパッケージのマッピング 144
- レコードタイプへのマッピング 122

状態タイプ モデル

- パッケージ 300, 302

状態モデル

- カスタマイズ 121
- 状態の削除 124
- 状態の追加 122
- 状態名の変更 123
- 設計に関する考慮事項 132
- 説明 119
- 遷移マトリックス 121
- レコードタイプ 297, 299

す

スキーマ

- Blank スキーマからの作成 86
- カスタマイズ 82, 97
- コピー 388
- コンポーネントの追加 135
- 削除 94
- 事前定義、作成方法 136
- 事前定義されたスキーマ 287, 288
- 事前定義、変更 133
- 事前定義、リスト 16, 17
- 説明 5, 81
- チェックアウト 84
- チェックイン 91
- テスト 82
- テスト データベース、設定 89
- データベースとの関連付けの変更 94
- パッケージ適用の確認 140
- パッケージの表示 309
- 変更の確認 88
- ユーザー アクションの制限 182
- ユーザー データベースのアップグレード 92
- 要求の定義 16

スキーマ リポジトリ

- 移動 65, 67
- コピー 383
- 作成 51
- 説明 5
- データ コード ページ値 28, 35, 37

せ

製造元データベース

- installutil パラメータ 381
- サポートされている文字セット 33, 34
- 説明 7
- データ コード ページの設定 50
- パラメータの指定 76
- 予約済みキーワード、フィールド 106

セキュリティ

- 制御されたレコードタイプ 188
- 設計の問題 191
- 説明 187
- 追加機能 197
- 例 192

セキュリティ コンテキスト フィールド 188, 189

接続

- UNIX クライアント 61
- 概要 58
- 削除 61
- 説明 7
- データ コード ページ値 59
- 名前の変更 59
- 複製 61
- プロファイル 63, 64
- 編集 60

て

テスト データベース

- 作成 82
- 設定 89

データ コード ページ

- MultiSite 関連の問題 41
- 値の設定 35
- 値の選択のガイドライン 29
- 値の適用 28, 31
- 以前のリリースとの相互運用 39
- サポート対象のリスト 31
- 接続 59
- 設定 50
- 説明 16, 27
- データの破壊 28
- 統合 307
- 統合関連の問題 42
- パッケージの適用 39
- 変更、影響 32

データのインポート

- cqload ユーティリティ 388
- インポート ファイル 272
- インポート ファイル形式 273, 274
- インポート ファイルの区切り文字 273
- エラーの回復 285
- 概要 269
- サポートされているデータ型 274, 275
- 重複レコード 284
- 前提条件 280
- 添付ファイル 281
- 添付ファイル形式 276
- 動的リスト 393
- 破棄データ 282
- フォーム 165
- フォームの変更 151
- 履歴ファイル形式 275, 276
- レコード ID 272
- レコードの更新 286

データのエクスポート

- cqload ユーティリティ 388
- 概要 269, 277
- 動的リスト 393

データベース

- 「スキーマ リポジトリ」、「ユーザー データベース」、「製造元データベース」を参照
- 説明 6
- データ コード ページ値の影響 31

データベース管理

- エイリアスの作成 50
- 概要 45, 46
- 空のデータベースの作成 50
- コード ページの設定 50
- コピーの作成 383
- 接続 58
- データベースの作成 51
- バックアップ 49

電子メール通知

- Rational E-Mail Reader 259, 263
- Web 設定 237
- アクション フック 257
- 演算子 248
- 往復 256
- 概要 245
- 書式例 267
- 説明 18
- 登録の有効化 258
- トラブルシューティング 408
- メールボックス 258
- 有効化 246
- ユーザーの設定 254
- ルール コントロール 247
- ルールのアクション コントロール 249
- ルールのアドレス 251, 252
- ルールの作成 246
- ルールのフィールドの表示 250
- ルールの無効化 248
- レコードの書式 265, 266

と

統合

- Microsoft Visual SourceSafe 307, 321, 322
- Project Tracker 313, 314
- Rational Administrator 312, 313
- Rational TeamTest 314, 315
- UCM 315, 316, 317, 318, 320
- 依存先、追加 311
- 依存先、リスト 306

現在のパッケージの表示 309
コード ページの問題 307
自主、必要なパッケージ 310
自主、リスト 306
説明 305
テスト 309
電子メール システム 306
パッケージのアップグレード 323
トラブルシューティング 401

は

パッケージ

アップグレード方法 146
コンポーネントの変更 137
サポート対象のリスト 289, 290, 291, 292, 293,
294, 295, 296, 297
使用可能なパッケージのリスト 138
状態タイプ モデル 300, 302
スキーマからの削除 137
説明 135
適用の確認 140
適用方法 141
データ コード ページ値 39
レコード タイプの有効化 308

ひ

表記規則 xxviii

ふ

フィールド

親子リスト 117
削除 113
スキーマのチェックイン後の変更 112
説明 104, 105
名前での予約済みキーワード 106, 113
レコード タイプへの追加 106, 107
レコードのリンク 114, 116

フィールド フック

削除 205
作成 204
サポート対象のリスト 203
実行順序 218, 219
説明 203
選択リスト 207, 208, 211
フィールド依存関係 206, 207
変更 205

フォーム

インポート 165
エクスポート 165
概要 147
再使用 165
削除 150
作成 148
名前の変更 149
複数のプラットフォーム 150
変更 149, 150, 151

フォーム上のコントロール

ActiveX 360
一覧 371
イメージの表示 375
オプション ボタン 372
親子リスト 373
グループ ボックス 369
コンボ ボックス 363
重複先 368
重複元 367
スクリプト トリガ 222
チェック ボックス 362
テキスト ボックス 377
添付リスト 361
ドロップダウン コンボ ボックス 366
ドロップダウン リスト 364
ボタン 375
リスト ボックス 370
履歴 369

フォームのコントロール

概要 151
削除 158
タブ 162, 163, 164
追加 153, 154, 155, 156
変更 157

フック

- アクション タイプ 129
- 外部アプリケーション 227
- 計画と開発の問題 200, 201
- スキーマ確認とエラー 88
- スクリプト 201, 202
- スクリプト内のテキストの検索 231
- 説明 10, 200
- 動作状況 228
- 例、UCM 統合用の CAL メソッド 352
- 例、依存先リスト 327
- 例、親レコードの値の設定 337
- 例、選択リスト 328

へ

変更依頼

- ライフサイクル 2
- レコード タイプ 98

変更管理 1, 2

ま

マスターシップ 184

ゆ

[ユーザー管理] ダイアログ ボックス 168

ユーザー データベース

- アクセス権の削除 175
- アクセス権の付与 174
- 移動 71
- 削除 74
- 削除の復元 75
- 作成 54
- サブスクライバの表示 168
- 情報のアップグレード 168, 176
- スキーマのアップグレード 92
- スキーマの変更 94

データ コード ページ値、設定 41

プロパティの更新 72

プロパティの表示 75

ユーザー情報のアップグレード 168

ユーザーの追加 171

ユーザー プロファイルの編集 176

ワークスペース項目の移動 395

ユーザーとグループ

アクションへのアクセス制限 182, 183

概要 167

グループの管理 180, 181, 182

権限の変更 177

サポートされている権限 169, 170

セキュリティ コンテキスト 188

プロファイルのインポートとエクスポート 184

有効化と無効化 178, 179

れ

レコード

API からの操作 229

アクセス制限 188

インポートされたレコードの更新 286

重複レコードのインポート 284

セキュリティ コンテキスト 190

電子メールの書式 265

非表示 188

リンク 114, 116

レコード スクリプト

Web アクセス 221

削除 224

説明 220

フォーム コントロール イベント 222

変更 224

レコード タイプへの追加 223

レコード タイプ

削除 104

状態あり 98

状態タイプへのマッピング 122

状態なし 100

状態モデル 297, 299

スキーマへの追加 99

- スクリプトの追加 223
- 制御、セキュリティ 188
- セキュリティ コンテキスト 189, 190
- 説明 98
- デフォルト 101
- 名前の変更 103
- パッケージの適用 308
- パッケージを適用した場合の既存のレコード
 タイプの処理 144
- ファミリー 101, 102, 103
- レポート
 - cqtool による生成 398
 - Web クライアントの制限 233

ろ

- ログイン、制限される文字 171

