



**IBM Rational Build Forge**



# 目次

## 第 1 章 IBM Rational Build Forge につ

### いて . . . . . 1

Rational Build Forge の概要 . . . . .	1
Build Forge 製品エディション . . . . .	2
プロセスの自動化 . . . . .	2
Build Forge の概念および目的 . . . . .	4

## 第 2 章 Build Forge の入門 . . . . . 9

コンソールへのアクセスおよびその使用 . . . . .	9
リストのフィルタリングおよびソート . . . . .	12
Hello World プロジェクトの作成 . . . . .	12
サーバーのセットアップ . . . . .	13
プロジェクトの定義 . . . . .	15
ジョブの実行 . . . . .	16
ジョブ結果の表示 . . . . .	17
プロジェクト・サンプル . . . . .	18

## 第 3 章 IBM Rational Build Forge の 資料に関する特記事項 . . . . . 19

## 第 4 章 要件 . . . . . 23

ハードウェア、ソフトウェア、およびデータベース の要件 . . . . .	23
データベースの要件 . . . . .	23
Installation Manager の要件 . . . . .	23
製品エディションごとのライセンス要件 . . . . .	24
インストール中のライセンス・ファイルの指定 . . . . .	24
Build Forge 用の Rational License Server の構成 ライセンス・キーの取得と、Rational License Server の設定 . . . . .	24
IPv6 サポートのネットワーキング要件 . . . . .	28
各国語サポート要件 . . . . .	30
管理コンソールおよびエージェントの言語設定 . . . . .	30
データベース・ホストでの国際データのサポート . . . . .	31
管理コンソールにおけるユーザー言語設定の変更 . . . . .	31
UNIX/Linux ホストの言語/文字セットの特定 . . . . .	31
Windows ホストの言語コード・ページの特定 . . . . .	32

## 第 5 章 インストールの計画 . . . . . 33

コンポーネント . . . . .	33
デプロイメントの種類 . . . . .	34
インストール方法 . . . . .	35
構成オプション . . . . .	35
他の製品との統合 . . . . .	35

## 第 6 章 インストール前のセットアップ 37

国際データのセットアップ . . . . .	37
データベースのバックアップ (必須) . . . . .	39
データベースのセットアップ . . . . .	39
DB2 のセットアップ . . . . .	39

Microsoft SQL Server のセットアップ . . . . .	43
MySQL のセットアップ . . . . .	49
Oracle のセットアップ . . . . .	53
セキュリティのセットアップ . . . . .	61

## 第 7 章 管理コンソールのインストール 63

ランチパッドを使用した Installation Manager の開始 . . . . .	64
製品 DVD からのランチパッドの開始 . . . . .	64
ダウンロードしたパッケージからのランチパッド の開始 . . . . .	65
Installation Manager でのインストール・ステップ . . . . .	66
インストール後のチェックリスト . . . . .	71
Linux のファイル・ハンドルの数の増加 . . . . .	73

## 第 8 章 代替のインストール方法 . . . . . 75

独自のコンポーネントを使用したインストール . . . . .	75
前提条件 . . . . .	75
国際データのサポート . . . . .	76
データベースのインストールと構成 . . . . .	77
Apache HTTP Server のインストールと構成 . . . . .	78
PHP のインストールと構成 . . . . .	80
Apache Tomcat のインストールと構成 . . . . .	83
Installation Manager の手動インストール . . . . .	86
Installation Manager の開始 . . . . .	86
リポジトリ URL の指定 . . . . .	87
製品コンポーネントのサイレント・インストールの 実行 . . . . .	88
Installation Manager での応答ファイルの作成 . . . . .	88
サイレント・モードでの Installation Manager の インストールおよび実行 . . . . .	89
製品コンポーネントのサイレント・アップグレード の実行 . . . . .	90
Installation Manager での更新用応答ファイルの作 成 . . . . .	90
サイレント・モードでの更新インストールの実行 . . . . .	91
VMware への Build Forge システムのインストール . . . . .	92
System z 上の Linux への管理コンソールのインスト ール . . . . .	92

## 第 9 章 管理コンソールでの追加機能の構 成 . . . . . 99

Build Forge 構成ファイル (buildforge.conf) . . . . .	99
buildforge.conf ファイルの更新 . . . . .	99
Buildforge.conf の参照 . . . . .	100
代替ポートを使用するように管理コンソールを構成 . . . . .	101
冗長化構成 . . . . .	102
冗長化について . . . . .	102
冗長化システムのインストール . . . . .	103
冗長化での処理 . . . . .	104
IPv6 ネットワーク・サポートの有効化 . . . . .	104
httpd.conf の変更 . . . . .	104

セキュリティ・フィーチャー	105
シングル・サインオンの実装	105
SSL および HTTPS の有効化	121
パスワード暗号化の有効化	143

## 第 10 章 エージェントのインストール 157

Windows プラットフォームへのエージェントのインストール	157
ユーザー・モード・エージェント	158
Windows オペレーティング・システムでのサイレント・エージェント・インストールの実行	159
UNIX および Linux システムへのエージェントのインストール	159
System i プラットフォームへのエージェントのインストール	161
System z プラットフォームでのエージェントのインストールおよび実行	163
z/OS でのエージェント・インストールのトラブルシューティング	165
「エージェント更新およびデプロイメント」ユーティリティー	166
サポートされるオペレーティング・システム	167
インストール	167
UNIX または Linux が稼働するターゲット・コンピュータの接続のセットアップ	168
Windows が稼働するターゲット・コンピュータの接続のセットアップ	168
使用法の概要	171
XML ファイルでのターゲット・システムの指定	172
agentupdate ユーティリティーの実行	176
agentupdate ユーティリティーのリファレンス	178
XML ファイル・パラメーター、コマンド行オプション、および環境変数の優先順位	179
ログ・ファイルを使用したユーティリティー・アクションの表示	181
トラブルシューティング	181
エージェントの実行	187
Windows でのエージェントの実行	187
UNIX、Linux、および MacOS でのエージェントの実行	187
System i でのエージェントの実行	188
bfagent の参照	190
エージェントの構成	190
エージェント構成ファイルの配置	190
エージェントのポートの変更	191
異なるシェルの構成	191
ネットワーク・ファイル共有システムでのエージェント・コマンドの実行 (Windows)	191
エージェント間の直接ファイル転送を構成するトリガー変数とエージェントのパフォーマンス	193
bfagent.conf の参照	193
エージェントのトラブルシューティング	207
ホストの名前解決のテスト	208
接続のテスト	208
Windows でのエージェントのトラブルシューティング	210

UNIX、Linux、または MacOS でのエージェントのトラブルシューティング	210
---	-----

## 第 11 章 インストール後のタスク 213

エンジンの始動と停止	213
Windows 上でのエンジンの始動と停止	213
UNIX または Linux システムでのエンジンの開始および停止	213
ユーザーのセットアップ	214
root ユーザー	214
ユーザーの作成および編集	215
インストール済み環境の検査	217
サーバー認証	217
サーバーの構成	218
テスト・プロジェクトの作成	218
プロジェクトの実行	218
ログの場所およびセットアップ	220
共通問題のトラブルシューティング	221

## 第 12 章 以前のバージョンからのアップグレード 223

アップグレードの前提条件	223
バージョン 7.1 コンソールのアップグレード	223
更新インストールの実行	224
バージョン 7.1.3.1 へのアップグレード後、アダプターをコピーする	226
バージョン 7.0.2.x コンソールのアップグレード	227
エージェントのアップグレード	227

## 第 13 章 製品コンポーネントのアンインストール 229

Installation Manager を使用した製品のアンインストール	229
Installation Manager によるインストールが失敗した場合の製品の手動アンインストール	230
アンインストール後の Build Forge ファイルのクリーンアップ	230
Installation Manager を使用した製品の再インストールおよび既存の DB2 Express の使用	231
Windows Build Forge エージェントのアンインストール	232
UNIX または Linux の Build Forge エージェントのアンインストール	232

## 第 14 章 リリース・ノート 235

リリース・ノート - IBM Rational Build Forge バージョン 8.0.0.2	235
---	-----

## 第 15 章 ダッシュボード・タブ 239

## 第 16 章 管理 241

管理について	241
アクセス・グループ	241
アクセスの概説	243
アクセスの例: グループへのジョブ実行権限の付与	244

チームおよびプロジェクトのセキュリティー・プラン	245
アクセス・プロパティの管理	246
ユーザー	246
ユーザー・プロパティ	246
ユーザーの管理	249
ユーザー・タイプを使用したアクセスの制御	250
アカウント・ロックアウト	250
root ユーザー	251
API ユーザー	252
権限	252
権限の演習	253
LDAP の統合	254
LDAP の統合について	254
LDAP ドメインのプロパティ	255
タスク	259
システム構成の設定	262
その他のシステム設定	273
監査ロギング	274
メッセージ	278
翻訳メッセージ	280
ジョブ状況に関する RSS データ・フィードへの加入	280
システム・メッセージに関する RSS データ・フィードのフィルター操作	281
「セキュリティー」パネル	281
「鍵ストア」パネル	284
「SSL」パネル	285
「SSO」パネル	286
ライセンスの管理	287
新規ライセンス・キーの入力	288
トークン・ライセンス・ログ	288
Perl エンジンの管理	288
Perl エンジンの一時停止	288
Perl エンジンの始動と停止	289
データベースの管理	290
データベース・ログ・ファイルの削除	290
エラー・メッセージ	290
アクティブ・ステップが存在しません	290
ライセンス・キーが無効であるか、または Build Forge ライセンス・キーが破損または欠落しています	291
データベース・ライセンスが必要です	291

## 第 17 章 サーバー 293

サーバー・リソースについて	293
「サーバー」パネルについて	294
サーバーの作成	295
サーバーのテスト	296
サーバーの変更	297
サーバーの有効化および無効化	297
サーバーでの同時ジョブの制限	297
ジョブ数のリセット	298
サーバーでのジョブ数のゼロへのリセット	298
すべてのサーバーでのジョブ数のゼロへのリセット	298

サーバー状況に関する RSS データ・フィード	299
サーバー認証	300
サーバー認証について	300
サーバー認証の作成	301
サーバー認証のオーバーライド	301
制限されたサーバー認証の使用の許可	302
セクター	302
セクターについて	303
セクターのセットアップ方法	304
セクター変数のタイプ	305
セクター変数の比較規則	305
セクターを用いた適切なサーバー・リソースの評価	306
コレクター	307
コレクターについて	307
「コレクター」パネルについて	307
コレクターの作成	308
マニフェストおよび動的サーバーの選択	309
マニフェストの表示	310
手動でのマニフェストのリフレッシュ	311
サーバー・マニフェストの更新頻度の設定	311
静的および動的なサーバー選択のセットアップ例	311
プロパティの参照	312
セクターの新規インスタンスを作成するためのスナップショットの使用	319
セクター・スナップショットの概要	319
セクター・スナップショットに関する計画	321
既存のセクターまたはセクター・スナップショットからのセクター・スナップショットの作成	321
デフォルトのセクター・スナップショットの変更	322
セクター・スナップショットのスナップショット名の変更	323
セクター・スナップショット・セット内のスナップショットのアクセスおよび表示	324
セクター・スナップショットの削除	325

## 第 18 章 環境変数の処理 327

環境変数について	327
環境の継承	327
継承の特殊なケース	328
ジョブの開始時に行われるプロジェクト変数の変更	329
変数について	329
ステップにおける変数の解釈	330
未宣言変数の解釈	331
「環境変数」パネルについて	331
「詳細」タブ	332
「スナップショット」タブ	335
環境変数の作成	335
変数の使用	336
変数のプルダウンの作成	336
他の環境変数の組み込み	337
ステップ実行時の変数値の変更	337
Windows ドライブのマッピング	338

変数でのドット・コマンドの使用 . . . . .	340
システム変数のリファレンス . . . . .	341
トリガー変数のリファレンス . . . . .	344
環境変数のスナップショット . . . . .	351
環境変数のスナップショットの概説 . . . . .	351
環境変数のスナップショットの計画 . . . . .	353
環境変数のスナップショットの作成 . . . . .	354
デフォルトの環境変数スナップショットの変更 . . . . .	355
環境変数のスナップショット名の変更 . . . . .	355
スナップショットへのアクセスおよびその表示 . . . . .	356
環境変数のスナップショットの削除 . . . . .	357

## 第 19 章 プロジェクトの処理 . . . . . 359

プロジェクトについて . . . . .	359
プロジェクト・パネルについて . . . . .	359
「プロジェクト」プロパティの変更 . . . . .	363
プロジェクトのコピー . . . . .	365
プロジェクトの削除 . . . . .	366
ステップをサーバーにスティックさせる . . . . .	367
チェーン: 別のプロジェクトまたはライブラリーの 条件付き実行 . . . . .	368
チェーン・プロジェクトでの環境変数の継承 . . . . .	369
待機が使用可能な場合のチェーニングされたプロ ジェクトの取り消し . . . . .	369
タグの定義 . . . . .	369
プロジェクトのタグ形式の編集 . . . . .	370
タグの同期化 . . . . .	370
タグのシステム定義変数 . . . . .	372
タグ変数の作成または編集 . . . . .	373
ライブラリー . . . . .	373
ライブラリーについて . . . . .	374
ライブラリーのコピー . . . . .	374
ログ・フィルター . . . . .	375
ログ・フィルターについて . . . . .	375
ログ・フィルターの作成 . . . . .	376
ステップへのログ・フィルターの割り当て . . . . .	377
フィルター・パターン . . . . .	378
フィルター・アクション . . . . .	379
フィルター通知 . . . . .	382
エラーしきい値 . . . . .	382
エラー・カウントおよび警告カウント . . . . .	382
クラス . . . . .	382
クラスについて . . . . .	382
通知のセットアップ . . . . .	385
通知テンプレートについて . . . . .	386
SMTP サーバーの構成 . . . . .	389
プロジェクトおよびステップの通知プロパティ の設定 . . . . .	389
通知の演習 . . . . .	389
通知テンプレートのカスタマイズ . . . . .	390
インライン化されたプロジェクトの通知 . . . . .	394
スナップショットを使用したプロジェクトの新規イ ンスタンスの作成 . . . . .	395
プロジェクト・スナップショットの概要 . . . . .	395
プロジェクト・スナップショットの計画 . . . . .	397
プロジェクト・スナップショット・オプション . . . . .	397

スナップショット権限のためのアクセス・グルー プの検証および編集 . . . . .	398
既存プロジェクトまたは既存プロジェクト・スナ ップショットからのプロジェクト・スナップシ ョットの作成 . . . . .	399
デフォルトのプロジェクト・スナップショットの 変更 . . . . .	400
プロジェクト・スナップショットのスナップシ ョット名の変更 . . . . .	401
プロジェクト・スナップショット・セット内のス ナップショットのアクセスと表示 . . . . .	402
デフォルトのプロジェクト・スナップショットの ジョブの開始 . . . . .	403
デフォルト以外のプロジェクト・スナップシ ョットのジョブの開始 . . . . .	403
プロジェクト・スナップショットの削除 . . . . .	404

## 第 20 章 ステップの処理 . . . . . 407

ステップについて . . . . .	407
「ステップ」パネルについて . . . . .	407
ステップの追加 . . . . .	411
ステップの編集 . . . . .	412
ステップの無効化 . . . . .	412
追加のステップ操作 . . . . .	413
実行フローの制御 . . . . .	413
ステップの実行方法 . . . . .	414
インライン: プロジェクトまたはライブラリーの ステップの組み込み . . . . .	415
ステップの成功チェーンおよび失敗チェーン . . . . .	416
スレッド: ステップの並列実行 . . . . .	417
複数サーバーへのステップのブロードキャスト . . . . .	418
条件付きステップの実行 . . . . .	420
While ループの実行 . . . . .	420
条件関数 . . . . .	421
ステップからのプロジェクトの起動 . . . . .	423
ログ出力のカスタマイズ . . . . .	424
ステップでのログ出力のラベル付け . . . . .	424
カラー・リンクまたはアクティブ・リンクを使用 したステップ出力の強調表示 . . . . .	425
ジョブ・データの処理 . . . . .	426
プロジェクト・ファイルへのビルド番号の組み込 み . . . . .	426
ジョブ中のビルド・タグの変更 . . . . .	427
ジョブ中の環境変数値の変更 . . . . .	427
レジスターの使用 . . . . .	429
プロジェクト・レジスター . . . . .	430
ステップでのサーバー・リソース間のファイルのコ ピー . . . . .	431
サーバー・リソースでのファイルのコピーの有効 化 . . . . .	431
サーバーからのファイルの取得 . . . . .	432
サーバーでのファイルの配置 . . . . .	432
直接ファイル転送を構成する . . . . .	432
ステップ処理のトラブルシューティング . . . . .	432
ANT ビルド・コマンド以降のステップ・コマン ドがジョブで処理されない . . . . .	433

Windows バッチ・コマンド以降のステップ・コマンドが実行されない	433
ドット・コマンドのリファレンス	434
ドット・コマンドの構文	434
.bom	434
.bomexport	436
.break	437
.bset	437
.buildstatus	438
.date	438
.defect	440
.drill	441
.edit	442
.email	443
.expand	443
.export	444
.get	444
.load	445
.lock	448
.mkdir	449
.monitor	449
.pack	449
.pop	450
.poptag	451
.purge	451
.push	451
.put	452
.rem	453
.report	453
.retag	454
.retry	454
.rget	454
.rmdir	455
.rput	455
.run および .runwait	456
.scan	458
.semget	459
.semput	459
.set	460
.sleep	460
.snapshot	461
.source	462
.stop	463
.strsub	463
.test	464
.tset	465
.unlock	465

## 第 21 章 ジョブの処理 . . . . . 467

ジョブについて	467
ホーム・パネルについて	467
「ジョブ」パネルについて	467
ジョブの実行および結果の表示	470
ジョブの開始	470
ジョブ結果の表示	472
ジョブ状況に関する RSS データ・フィード	473

失敗したジョブの再始動	474
BOM (Bill of Materials) の使用	475
ジョブのスケジューリング	478
スケジュール・パラメーター	480
ジョブのクラスのページのスケジューリング	483
ジョブの管理	483
ジョブのロック	484
ジョブの削除	484
ジョブ用の作業ディレクトリー	485
セマフォ	488

## 第 22 章 レポートの処理 . . . . . 491

レポートについて	491
レポート出力にデータを表示するための前提条件	492
レポート内のデータを表示するためのアクセス権限	492
組み込みレポートの結果を CSV ファイルにエクスポートする	492
パフォーマンスに関する標準レポート	493
プロジェクトのジョブ・パフォーマンス統計の表示	493
プロジェクトのジョブ所要時間の表示	494
プロジェクトごとのステップおよびサーバー・パフォーマンスの表示	494
照会のための定義済みレポート	495
セクター使用率履歴の表示	495
サーバーごとに現行サーバーのマニフェストを表示する	496
ジョブの成功/失敗/警告結果の表示	496
日付範囲内のジョブのサーバー使用率の表示	497
MD5 値を使用して行うジョブ・ファイルの検索	498
クイック・レポートでのレポートの作成	499
クイック・レポートのためのレポート・グループ権限	499
クイック・レポートのためのレポート・タイプ参照	500
クイック・レポートのためのレポート・フォーマットおよび表示の参照	506
サンプル・レポートの参照	511
用意されたレポート・タイプを使用したレポート作成	513
ジョブ BOM へのレポート出力の追加	514
クイック・レポートでのレポートの変更および管理	515

## 第 23 章 ユーティリティーの処理 . . . 519

コマンド行ユーティリティーのアクセスと実行	519
「非表示の割り当て」のマイグレーション	519
bfassignhiddenmigration リファレンス	519
プロジェクトのエクスポート	521
bfexport 参照	522
.export の使用	525
プロジェクトのインポート	525
インポート・ユーティリティーを使用したプロジェクトおよびその他のオブジェクトのインポート	525
bfimport 参照	527



インポート・オブジェクトへのアクセス・グループの割り当て方法 . . . . .	531
アクセス・グループの割り当ておよびセキュリティ . . . . .	531
インポート時のオブジェクトの名前変更および置換 . . . . .	532
レガシー Perl ジョブ・コントローラー・プロジェクトのマイグレーション . . . . .	534
Bfperlprojectmigration ユーティリティーのリファレンス . . . . .	534

## 第 24 章 「UI 構成」タブの Web リソースへのリンク . . . . . 537

<b>第 25 章 Build Catalyst . . . . .</b>	<b>539</b>
サポートされるオペレーティング・システム . . . . .	539
インストールの概要 . . . . .	540
Linux および Solaris オペレーティング・システムでの Build Catalyst のインストール . . . . .	540
Windows での Build Catalyst のインストール . . . . .	541
Build Catalyst の例 . . . . .	542
rafmake ユーティリティーのリファレンス . . . . .	545
Build Catalyst の環境変数 . . . . .	548

<b>第 26 章 Rational Build Forge の統合方式 . . . . .</b>	<b>551</b>
コマンド行インターフェースを使用した統合 . . . . .	551
アダプターを使用した統合 . . . . .	552
アダプター要件 . . . . .	552
アダプター . . . . .	553
アダプター・タスクの概説 . . . . .	563
コア・アダプター・タスク . . . . .	566
ClearQuest ビルド・レコードの更新 . . . . .	573
拡張アダプター・タスク . . . . .	574
アダプター参照 . . . . .	580
アダプター XML 参照 . . . . .	583

## 第 27 章 Rational Build Forge と Rational ClearCase . . . . . 595

## 第 28 章 Rational Build Forge と Rational ClearQuest . . . . . 601

## 第 29 章 Rational Build Forge と CVS . . . . . 607

<b>第 30 章 IDE 統合 . . . . .</b>	<b>609</b>
IDE 統合について . . . . .	609
テスト・プロジェクト用の特殊変数 . . . . .	610
Eclipse および Rational Application Developer のプラグイン . . . . .	611
Eclipse または Rational Application Developer 用プラグインのインストール . . . . .	612
SSL が使用可能な場合の代替インストール . . . . .	613

Eclipse および Rational Application Developer 用プラグインの使用 . . . . .	615
Rational Team Concert 用プラグイン . . . . .	616
Rational Team Concert プラグインの使用 . . . . .	616
Rational Team Concert プラグインのトラブルシューティング . . . . .	617

## 第 31 章 Rational Build Forge と Perforce . . . . . 619

## 第 32 章 Rational Build Forge と Rational Quality Manager . . . . . 621

<b>第 33 章 Rational Build Forge と Rational Team Concert . . . . .</b>	<b>623</b>
Rational Team Concert バージョン 3.x、バージョン 4.x . . . . .	623

## 第 34 章 Rational Build Forge と StarTeam . . . . . 625

## 第 35 章 Rational Build Forge とサブバージョン . . . . . 627

<b>第 36 章 Rational Build Forge と WebSphere 製品 . . . . .</b>	<b>629</b>
Apache Tomcat の代わりに WebSphere Application Server を使用 . . . . .	629
Build Forge アプリケーションを実行するように Websphere Application Server を構成 . . . . .	629
Websphere Application Server で、Java 2 セキュリティーを Build Forge 用に構成 . . . . .	633
Websphere Application Server で、SSL およびパスワード暗号化サポートを Build Forge 用に構成 . . . . .	633
Websphere Application Server を使用して Build Forge アプリケーションを実行するときの始動順序 . . . . .	634
Apache HTTP Server の代わりに IBM HTTP Server を使用 . . . . .	635
プリコンパイルされた PHP ファイルをセットアップする . . . . .	636
IBM HTTP Server 構成ファイルの編集 . . . . .	637
PHP でのプロキシ・サーバーの識別 . . . . .	638
IHS 用 SSL の構成 . . . . .	638
Windows: DLL の置換 . . . . .	640
AIX システム: 追加のセットアップ . . . . .	640

<b>第 37 章 API の処理 . . . . .</b>	<b>641</b>
API プログラム用の Build Forge ユーザーの作成 . . . . .	641
Java クライアント API . . . . .	642
Java クライアント API パッケージの取得 . . . . .	642
Java クライアント API のセットアップ . . . . .	642
Perl クライアント API . . . . .	643
Perl クライアント API パッケージの取得 . . . . .	643



Perl クライアント API のセットアップ . . . . .	644	コレクター (collector). . . . .	658
<b>第 38 章 管理コンソールのバージョン番号の判別 . . . . .</b>	<b>647</b>	データベース (database) . . . . .	658
<b>第 39 章 診断 . . . . .</b>	<b>649</b>	ダイナミック (dynamic) . . . . .	658
Build Forge サービス層のデバッグ診断 . . . . .	649	エンジン (engine) . . . . .	658
Build Forge のデフォルトのログ . . . . .	650	環境変数 (environments) . . . . .	658
Java エンジン・ログの構成 . . . . .	650	ハンドシェーク (handshake). . . . .	659
Build Forge エージェント・アクティビティ・ログの有効化 . . . . .	650	インターセプター (interceptor). . . . .	659
パフォーマンス問題の診断 . . . . .	651	インターフェース (interface) . . . . .	659
<b>第 40 章 パフォーマンスとスケーラビリティ . . . . .</b>	<b>653</b>	ジョブ (job). . . . .	659
<b>第 41 章 製品によりインストールされる実行可能なコマンド . . . . .</b>	<b>655</b>	ライブラリー (library) . . . . .	659
<b>第 42 章 用語集 . . . . .</b>	<b>657</b>	Lightweight Directory Access Protocol . . . . .	659
アクセス・グループ (access group) . . . . .	657	マニフェスト (manifest) . . . . .	659
アダプター (adaptor) . . . . .	657	管理コンソール (Management Console) . . . . .	660
エージェント (agent) . . . . .	657	通知テンプレート (notification template). . . . .	660
アーカイブ (archive) . . . . .	657	プラグイン (plug-in) . . . . .	660
BOM . . . . .	657	プロジェクト (project) . . . . .	660
クラス (class) . . . . .	658	セレクター (selector) . . . . .	660
強制削除 (clobber). . . . .	658	セマフォ (semaphore) . . . . .	660
		サーバー (server) . . . . .	661
		サービス (service) . . . . .	661
		スナップショット (snapshot) . . . . .	661
		静的 (static) . . . . .	661
		ステップ (step). . . . .	661
		ステップ・ログ (step log) . . . . .	662
		スレッド (threading) . . . . .	662
		ユーザー (user). . . . .	662



---

## 第 1 章 IBM® Rational® Build Forge® について

Rational® Build Forge® をインストールおよび使用する前に、基本概念を読み、理解しておいてください。また、各種製品エディションを確認することもできます。

---

### Rational Build Forge の概要

IBM® Rational® Build Forge® は、ソフトウェアとシステムのデリバリーを促進および合理化する、ビルドの自動化機能および管理機能を備えています。

Rational® Build Forge® 製品には、ソフトウェアの開発とデリバリーを自動化および促進するためのソフトウェア・アセンブリー・ラインを作成する適応実行フレームワークが用意されています。アセンブリー・ライン内のすべてのプロセスのオーケストレーション、管理、および追跡も行います。反復タスクを標準化および自動化し、開発ボトルネックを検出し、特定プロジェクトの傾向を判別し、コンプライアンス委任を管理することが可能になりました。

Build Forge® 製品は、ご使用の現行環境に統合され、主要な開発言語、スクリプト、ツール、およびプラットフォームをサポートします。資産、スクリプト・ソフトウェア、ハードウェアなどの既存の投資を引き続き使用しながら、プロセスの自動化、促進、通知、およびスケジューリングに関する有用な機能を追加することができます。

Rational® Build Forge® の適応実行フレームワークにより、ソフトウェアの開発プロセスおよびビルド・プロセスが自動化されます。

- 分野サイロ間の遅延およびハンドオフをなくすために、開発およびテストにわたる自動化を統一します。
- 標準の自動化システムにより、複数分野にわたって既存ツールを制御します。
- 複雑な、または古いスクリプトへの依存を体系的に低減または除去します。
- 一貫性があり、測定可能で、改善可能なプロセスを、複数のプラットフォームおよび製品にわたって実現します。
- ビルド・チームへの依存をなくすために、開発者セルフ・サービスを可能にします。
- マウスをクリックするだけで、自動化されたパラレル・アクティビティーを作成します。
- 状況の確認および問題のトラブルシューティングを簡単に行うことができるように、プロセスおよび結果の一貫性のあるビューを提供します。
- 一貫性のある自動的生成レポートにより、コンプライアンスの表示プロセスを容易にします。
- 同じ時間内でより多くのプロセス・サイクルを可能にすることで、品質を向上させます。
- 一般的なプラットフォームのエージェントに、他のプラットフォームで提供およびサポートされるソース・キットを提供します。

---

## Build Forge 製品エディション

この製品の使用可能なエディションは以下のとおりです。

- IBM Rational Build Forge Standard Edition
- IBM Rational Build Forge Enterprise Edition
- IBM Rational Build Forge Enterprise Plus Edition

表にはエディション固有のコンポーネントまたは機能がリストされています。

コンポーネントまたは機能	Standard Edition	Enterprise Edition	Enterprise Plus Edition
管理コンソール	Microsoft Windows、UNIX、または Linux	Windows、UNIX、または Linux	Windows、UNIX、または Linux
データベース	サポートされるすべてのデータベース	サポートされるすべてのデータベース	サポートされるすべてのデータベース
エージェント	サポートされるすべてのオペレーティング・システム	サポートされるすべてのオペレーティング・システム	サポートされるすべてのオペレーティング・システム
ライセンス・サーバー	必要 (同時ユーザー数 25)	必要 (同時ユーザー数 150)	不要 (同時ログイン数 250)
Adaptor Toolkit	サポート	サポート	サポート
クイック・レポート	サポート	サポート	サポート
API (Perl、Java)	サポート	サポート	サポート
動的サーバー管理	未サポート	サポート	サポート

---

## プロセスの自動化

プロセスの自動化により、ビジネスは、さまざまなアプリケーション、ユーザー、およびシステム全体でプロセスのコレオグラフィー (振り付け) を実施することで、非効率性を取り除き、コストを最適化し、コンプライアンスを確保して、生産性を向上させることができます。

Build Forge は、製品開発のアセンブリー・ライン内でプロセスの自動化、オーケストレーション、管理、および追跡を実行します。これは、ソフトウェア・ビルドおよびパッケージ・プロセスを自動化する場合に一般的に使用されます。

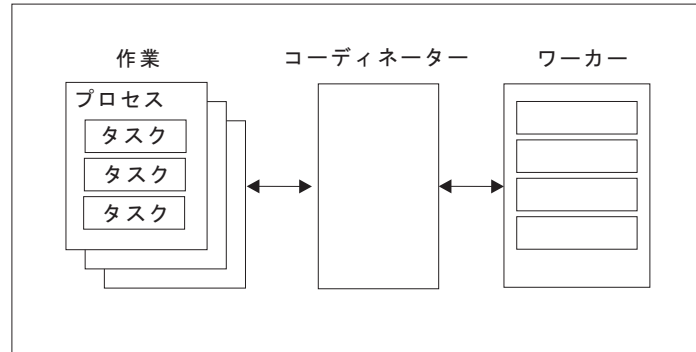
このセクションでは、プロセスのヒューマン・モデルを説明し、Build Forge 機能にマップします。

### ヒューマン・モデル

次のロールから、プロセスの自動化のヒューマン・ベースのモデルが構成されます。

- **作業:** 実行対象の有効な作業は、一連の文書化されたプロセス です。各プロセスは複数のタスク で構成されます。

- **コーディネーター:** コーディネーター が作業を「実行」します。コーディネーターは、実行するプロセスを選択し、タスクを順番に調べます。タスクごとに、プロセスではどのワーカーがそのタスクを実行するかが定義されています。コーディネーターは、タスクをワーカーに渡し、その結果を待機します。
- **ワーカー:** ワーカー は、プロセスに定義されているタスクを実行します。



## Build Forge モデル

ヒューマン・ベースのモデルと Build Forge モデルのマッピングは、次のようになります。

- **作業:** 作業プロセスは、Build Forge のプロジェクト・オブジェクトに定義されます。各プロジェクトに 1 つ以上のステップ・オブジェクトが含まれます。プロジェクト内のステップは、ヒューマン・モデルではプロセス内のタスクに対応します。
- **コーディネーター:** コーディネーターは、実行中の Build Forge ソフトウェアです。ここには、プロセス・エンジンと呼ばれるコンポーネントが含まれます。Build Forge でプロジェクトを開始すると、プロセス・エンジンはそのプロジェクトをジョブ・オブジェクトとして実行します。ジョブは、単なる実行中のプロジェクトです。
- **ワーカー:** ワーカーはホスト・コンピューターです。Build Forge ではサーバー・オブジェクトで表されます。ホストには、エージェント がインストールされている必要があります。

これは非常に簡素化したモデルです。

- Build Forge には、プロセスの自動化をサポートするためのその他の多くのオブジェクト・タイプが含まれています。サーバーとプロジェクトをサポートするユーザー、許可、およびその他のオブジェクトはすべて、データベースに格納されます。UI ウィジェットなど、Build Forge のパーツもデータベースに格納されます。Build Forge を稼働するためにはデータベースを実行する必要があります。
- 複雑な依存関係とワークフローを含む複雑なプロセスをサポートするために、プロジェクトとステップはさまざまな方法で実行するように構成できます。プロジェクトは他のプロジェクトを実行でき、ステップそのものがプロジェクトを実行することもあります。
- ステップとプロジェクトは、サーバー定義の一部である基準データを基に、動的にサーバーを選択できます。例えば、ステップは、Windows または UNIX/Linux のどちらが実行環境であるかに基づいて、実行するサーバーを選択できます。ま

た、サーバーは、プール化リソースとして構成することもできるため、その場合、ステップは、サーバーの現在の可用性や負荷を基にサーバーを選択します。

## インストールする Build Forge コンポーネント

Build Forge を使用するには、次のコンポーネントをインストールする必要があります。

- データベース: Build Forge は、データベースを使用して、使用するすべてのオブジェクトを格納します。
- Build Forge: データベースへのアクセスを制御し、UI コードを実行し、実行中のジョブの作業を実行する、一連のテクノロジーです。
- エージェント: 作業を実行する各ホストにエージェント・ソフトウェアをインストールします。
- Web ブラウザー: 一般に、ユーザーは Web ブラウザーに表示されるコンソールを介して Build Forge にアクセスします。

---

## Build Forge<sup>®</sup> の概念および目的

Build Forge は単純なオブジェクト・セットを使用してプロセス自動化の記述を構成します。

このセクションでは、主な概念とオブジェクトの概要、およびそれらの関係について説明します。わかりやすいように、より詳しい説明へのリンクが含まれています。

### ユーザーとロール

Build Forge にアクセスするには、ユーザーは、対応するユーザー・オブジェクトをセットアップしておく必要があります。実動システムでこれを行うには、ID 管理システムである LDAP にアクセスするように Build Forge を構成します。

Build Forge は、許可システムを使用して、すべてのオブジェクトへのユーザー・アクセス権を制御します。アクセス・グループ・オブジェクトには、アクセス権のリストが含まれています。ユーザー は、1 つ以上のアクセス・グループに属します。

独自のアクセス・グループを作成したり、用意されているアクセス・グループを変更したりできます。システムに定義されているアクセス・グループは、次のとおりです。

- ビルド・エンジニア - プロセスを定義します (プロジェクトとステップを作成します)。
- システム管理者 - サーバーおよびその他のシステム全体の機能を管理します。
- セキュリティー - ユーザー、アクセス、およびセキュリティを管理します。
- 開発者 - プロセスの開発の支援、ジョブの実行、および結果の表示を行います。
- オペレーター - プロジェクトのコピー、ジョブの実行、結果の表示を行います。
- ゲスト - プロジェクトのコピー、ジョブの実行、結果の表示を行います。



## サーバー

サーバー・オブジェクトでは、プロジェクトおよびステップを実行できる場所を定義します。プロジェクトおよびステップは、同じサーバーを使用することも、個別にサーバーを選択することもできます。

サーバー・オブジェクトは、作業を実行するホストを表します。ホストは、システムから作業を受け取って結果を返すために、Build Forge エージェントを実行している必要があります。

その他のオブジェクトは、サーバーに関連するため、プロジェクトおよびステップを定義する前にセットアップしておく必要があります。

- **サーバー認証:** サーバー認証 には、サーバーで指定されたホストにアクセスする場合に使用する、サーバーのログイン情報を保管します。サーバー認証は、そのサーバー認証を使用するサーバーを作成する前に作成しておく必要があります。サーバー定義で、システムに定義されているすべてのサーバー認証のリストからサーバー認証を選択してください。
- **コレクター:** コレクター・オブジェクトは、サーバーの指定されたプロパティを収集します。データは、マニフェスト に保管されます。サーバーには、プロパティのデフォルト・セットが割り当てられています。これらのビルトイン・プロパティには、ホスト・アーキテクチャー、ネットワーク接続、およびリソース (CPU、メモリー、負荷) に関する情報が含まれます。その他のプロパティを追加するには、コレクター・オブジェクトを定義します。コレクターをサーバーに追加するには、事前にコレクターを作成しておく必要があります。サーバー定義で、システムに定義されているすべてのコレクターのリストからコレクターを選択してください。
- **セクター:** セクター・オブジェクトでは、プロジェクトまたはステップで使用するサーバーを選択する方法を定義します。セクターを、そのセクターを使用するプロジェクトに追加するには、事前に作成しておく必要があります。プロジェクト定義で、システムに定義されているすべてのセクターのリストからセクターを選択してください。
- **環境変数:** 環境変数 オブジェクトは、ステップで使用できる一連の変数です。ジョブの際に、ステップが実行される前にサーバー・ホスト上で変数が設定されます。環境変数は、サーバー・オブジェクト、プロジェクト・オブジェクト、およびステップ・オブジェクトに関連付けることができます。同じ変数が環境変数に依って異なる値に設定されている場合は、継承スキームによって、使用する値が判別されます。環境変数をサーバー、プロジェクト、またはステップに追加するには、事前に作成しておく必要があります。それらのオブジェクト定義で、システムに定義されているすべての環境変数のリストから環境変数を選択してください。

## 環境変数

環境変数 は、変数のセットです。環境変数は、サーバー・オブジェクト、プロジェクト・オブジェクト、およびステップ・オブジェクトに対して指定できます。ステップの実行時に、それらの各オブジェクトで設定されている環境変数が組み合わされて、使用するステップに対する変数が提供されます。 327 ページの『環境変数について』を参照してください。

ステップが実行されると、変数が変更される場合があります (337 ページの『ステップ実行時の変数値の変更』を参照してください。) 変更の範囲は、ステップに対してローカルの場合、プロジェクトに対してローカルの場合、または永続 (変数は格納されている環境で変更されます) の場合があります。

定義済みのシステム変数だけでなく、ユーザー定義の変数も使用可能です。

## プロジェクト

プロジェクト では、プロセスで実行する作業を定義します。プロジェクトが開始されると、プロジェクトはジョブとして実行します。

実行する作業は、ステップのリストに含まれます。

プロジェクトに関連するその他のオブジェクトは、次のとおりです。

- **セレクトター:** セレクトター では、プロジェクトを開始する場所を指定します。プロジェクトのセレクトターが指定されていない場合、そのプロジェクトは独立して実行できず、ライブラリーと呼ばれます。セレクトターをジョブに割り当てるには、事前に定義しておく必要があります。
- **環境変数:** 環境変数 オブジェクトは、ステップで使用できる一連の変数です。ジョブの際に、ステップが実行される前にサーバー・ホスト上で変数が設定されます。環境変数は、サーバー・オブジェクト、プロジェクト・オブジェクト、およびステップ・オブジェクトに関連付けることができます。同じ変数が異なる値に設定されている場合は、継承スキームによって、使用する値が判別されます。環境変数を使用するサーバー、プロジェクト、またはステップを作成する場合は、事前にその環境変数を作成しておく必要があります。それらのオブジェクト定義で、システムに定義されているすべての環境変数のリストから環境変数を選択してください。
- **通知テンプレート:** 通知テンプレート では、ジョブ・アクティビティー (ジョブの開始、ジョブの成功、ジョブの失敗など) に関する通知の送信方法を定義します。通知オブジェクトでは、アクセス・グループを介して通知担当者を定義します。SMTP サーバーのロケーションは、システム設定で指定します。
- **クラス:** クラス ・オブジェクトを使用して、メンテナンスのためのプロジェクトのグループ化を行います。一般に、クラスは、完了済みのジョブを定期的にアーカイブしたりアーカイブしたりする場合に使用します。プロジェクトを使用して実行されたジョブはすべて、クラスの影響を受けます。
- **アダプター:** アダプター では、外部システム (一般的には、ソース・コード管理システム) との統合を定義します。開始点として、複数のサンプル・テンプレートが用意されています。接続を管理し、指定されたアクションをシステムで実行するには、さらに細かく構成する必要があります。
- **アダプター・リンク:** アダプター・リンク では、アダプターとそのアダプターを使用するプロジェクト間の関係を定義します。

## ステップ

ステップ では、実行する最小作業単位を定義します (407 ページの『ステップについて』を参照してください)。そのキー・コンポーネントは、選択したサーバーで実行するコマンドが含まれている、コマンド・プロパティーです。

コマンド・プロパティを使用して、ドット・コマンドを実行することもできます。ドット・コマンドとは、プロセス・エンジン上で実行して、追加機能を提供するコマンドです。

プロジェクトに関連するその他のオブジェクトは、次のとおりです。

- セレクター: ステップには、独自のセレクターを割り当てることができます。セレクターが指定されていない場合は、プロジェクトのセレクターが使用されます。
- 環境変数: ステップには、独自の環境変数を割り当てることができます。サーバー、プロジェクト、およびステップに用意されている環境変数が組み合わせられます。デフォルトでは、環境変数はその順序で適用されるため、ステップの環境変数で定義されている変数はすべて、同じ変数の定義に優先します。優先順位はシステム設定で制御できます。
- ログ・フィルター: ステップには、結果プロパティでログ・フィルターを割り当てることができます。ログ・フィルター・オブジェクトは、ステップの成否を示す条件を指定するために使用されます。Perl エンジンを実行している場合は、正規表現を使用してログをスキャンし、特定のパターンを見つけます。また、Java エンジンを実行している場合も、正規表現を使用してログをスキャンし、特定のパターンを見つけます。通常はコマンドの終了状況が使用されますが、ログ・フィルターでは別の手段も提供しています。ステップでログ・フィルターを指定するには、ログ・フィルターを事前に作成しておく必要があります。ステップの結果プロパティで、システムに定義されているすべてのログ・フィルターのリストからログ・フィルターを選択します。
- 通知テンプレート: 通知テンプレート では、ステップ・アクティビティ (ステップの開始、ステップの成功、ステップの失敗など) に関する通知の送信方法を定義します。通知オブジェクトでは、アクセス・グループを介して通知担当者を定義します。SMTP サーバーのロケーションは、システム設定で指定します。
- Build Catalyst: ステップは、Build Catalyst の主要ユーティリティ `rafmake` を実行できます。Build Catalyst は、Make ベースのビルドを高速化する手段を提供します。Build Forge エージェントのほかに、Make ビルドを実行する同一のホストに Build Catalyst もインストールする必要があります。

## ジョブ

ジョブは実行中のプロジェクトです。ジョブが開始されると、プロセス・エンジンはそのジョブをキューに入れてから実行します。ジョブの実行状況は、「ジョブ」パネルで確認できます。ジョブが完了すると、次の情報を入手できます。

- 結果: ジョブを開くことによって、すべてのステップの結果を確認できます。また、実行中のジョブを開いてその進行状況をモニターすることもできます。
- ステップ・ログ: ステップ・ログには、実行結果のほかに、マニフェスト、環境変数の設定に関する情報など、ステップの実行内容に関する広範な情報が記録されます。「結果」ページでステップの結果のリンクをクリックすると、ログが表示されます。
- 部品構成表: 部品構成表 (BOM) には、ジョブ・ステップおよびステップのマニフェストに関する情報が含まれています。`.bom` ドット・コマンドでは、BOM

の追加情報を書式設定し、データを BOM に書き込むことができます。..scan ドット・コマンドでは、ベースラインおよびチェックポイントを BOM に追加することができます。

ジョブは、取り消したり、再始動したりできます。プロジェクトをスケジュールに追加して、定期的にプロジェクトを実行させることができます。

---

## 第 2 章 Build Forge® の入門

これらのトピックでは、システムについて簡単に紹介します。

### インストール

まだインストールしていない場合、最初にシステムをインストールしてください。システム（管理コンソールおよびエージェント）を単一のホストにインストールすると、簡単に始めることができます。

1. 以下の要件を確認します。
  - 23 ページの『データベースの要件』
2. 以下のステップを完了します。
  - a. 33 ページの『第 5 章 インストールの計画』（スタンドアロン、通常の実動、分散など）
  - b. 37 ページの『第 6 章 インストール前のセットアップ』（国際データ、データベース、およびセキュリティ）
  - c. 63 ページの『第 7 章 管理コンソールのインストール』
  - d. 157 ページの『第 10 章 エージェントのインストール』

---

### コンソールへのアクセスおよびその使用

このトピックでは、システム・メニューおよびタブの使用に関する基本的な情報について説明します。

#### コンソールへのアクセス

コンソールにアクセスするには、次の手順を実行します。

1. ブラウザー・ウィンドウを開きます。
2. コンソールの URL を入力します。

`http://host:port/`

*host* には、コンソールが稼働しているホストの完全修飾ネットワーク名または IP アドレスを指定します。

*port* を指定する必要があるのは、HTTP のデフォルト・ポートが使用されていない場合のみです。デフォルト・ポートは、SSL に対応していない場合は 80 です。デフォルトの HTTPS ポートは 443 です。HTTPS は、SSL を使用するようコンソールが構成されている場合にのみ使用されます。

例:

- `http://localhost/` または `http://127.0.0.1/`: コンソールがインストールされているホスト上でブラウザーを実行している場合に使用できます。
- `http://my.company.com/`

- `http://my.company.com:81`: コンソールはポート 81 を使用するよう構成されているため、ポート番号を指定する必要があります。
3. ログインします。次の情報を入力して、「**ログイン**」をクリックします。
- ユーザー名
  - パスワード
  - ドメイン

デフォルトでは、資格情報入力用の別の URL にリダイレクトされ、ログインが正常に行われると、元の場所に再度リダイレクトされます。この URL は、サービス・コンポーネントの URL です。これは、認証のためのセキュアなサブレットを提供します。その使用は構成可能で、オフになっている場合またはユーザー認証に LDAP が使用される場合は、使用されません。詳しくは、121 ページの『デフォルト・ログイン・セキュリティについて』を参照してください。

root、または Build Forge® でのみ定義されている別のユーザーとしてログインすると、「ドメイン」の入力を求めるプロンプトは出されません。LDAP を使用してユーザーを認証するように Build Forge を構成することをお勧めします。このようにセットアップされている場合は、「ドメイン」フィールドが表示されません。この場合のドメインとは、ユーザーの認証に使用される LDAP ドメインのことです。

254 ページの『LDAP の統合』も参照してください。

## ユーザー・セッション

ユーザー・セッションは、最初に Cookie で生成されるセッション ID によって追跡されます。生成されたセッション ID は、データベースに保管されます。すべてのユーザー対話が、このセッション ID に対して検査されます。

通常は、同じログインを使用して複数のブラウザー・ウィンドウを開くことができますが、それらのウィンドウは、同じホスト上にあり、同じタイプのブラウザー (Firefox、Internet Explorer) を使用している必要があります。

セッションは、以下のいずれかの状態になると閉じられます。

- 別のホストから同じユーザー・アカウントを使用してログインが行われた。
- 同じホストから同じユーザー・アカウントを使用してログインが行われたが、使用されたブラウザーのタイプが異なっていた。
- 管理コンソール・セッションを含むブラウザーのウィンドウまたはタブをユーザーが閉じた。
- セッションの制限時間に達した。制限時間は、「自動ログオフ」システム設定で設定します。デフォルトでは、制限時間なしを表す 0 に設定されています。

## 制約およびヒント

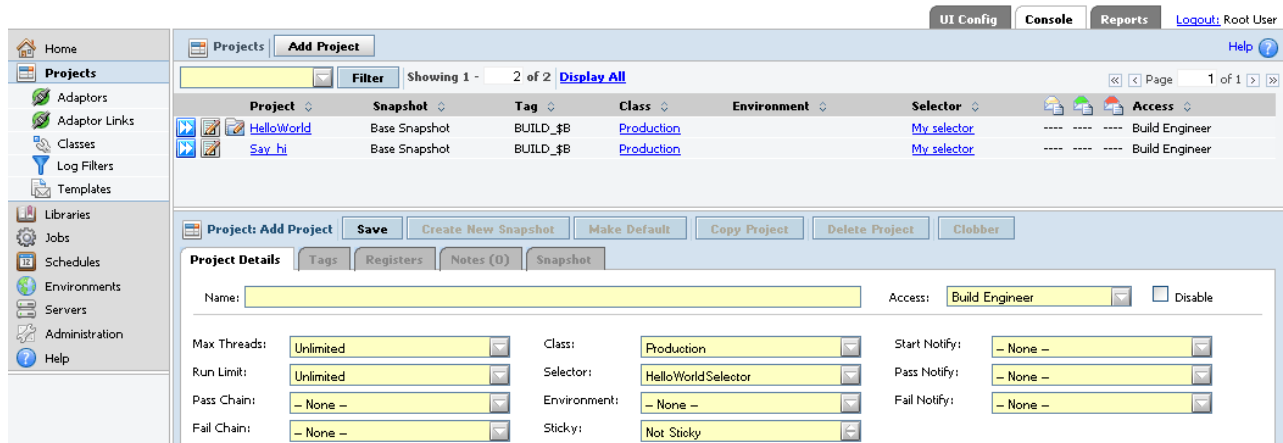
次の使用上の制約およびヒントに注意してください。

- 1024 x 768 以上のディスプレイを設定してください。最適な結果を得るためには、1280 x 1024 以上を使用してください。
- ブラウザー・ウィンドウを 1024 x 768 より小さくしないでください。



- ・ フォント・サイズ、色、およびその他のユーザー補助機能を制御するには、ご使用のブラウザの設定を使用してください。新しいブラウザ設定を適切に表示するために、必要に応じてページを最新表示してください。
- ・ Microsoft Internet Explorer では、フォントが大きすぎる場合や、Internet Explorer を縮小したときに、「ジョブ (Job)」 > 「開始」 ページに管理コンソール (MC) が表示されません。


## コンソール・ウィンドウの使用



右上のタブをクリックして、アプリケーションを表示します。「ダッシュボード」(デフォルト)、「UI 構成」、「コンソール」、または「レポート」をクリックできます。システムは、ログアウト時に表示されていたタブを記憶していて、次回のログイン時にそれを表示します。

「コンソール」アプリケーションでは、次のようにナビゲートします。

- ・ メニューが左側に表示され、メイン表示パネルが右側に表示されます。
- ・ メニュー項目をクリックすると、パネルが表示されるか、パネルのサブメニューが開きます。
- ・ パネルが表示領域より長い場合は、右上のページング・コントロールを使用します。
 

<< < Page 1 of 4 > >>
- ・ リストを表示する場合は、「フィルター」フィールドを使用します。ストリングを入力して「フィルター」をクリックすると、リストが更新されます。リストには、名前にそのストリングを含む項目のみが表示されます。
- ・ メニューのサイズを変更するには、メニューの右端をドラッグします。
- ・ 通常、データの作成または編集に使用できるパネルでは、次のコントロールが使用できます。
  - パネルの上部で、項目を表示および選択することができます。項目の名前をクリックすると、その項目の内容が表示されます。項目を編集するには、「編集」アイコン  をクリックします。
  - パネルの下部に、選択された項目の内容が表示されます。
- ・ リストの項目を選択すると、左側のメニューの下に追加情報が表示される場合があります。例:

1. 「管理」 > 「ユーザー」の順に選択して開きます。
2. ユーザーをクリックします。

そのユーザーの権限が、左側のメニューの下に表示されます。

---

## リストのフィルタリングおよびソート

リストは、素早くフィルタリングしたり、ソートしたりすることができます。

リストをフィルタリングするには、以下のようにします。

1. 「フィルター」テキスト・ボックスにテキストを入力します。この入力には大/小文字の区別があります。
2. 「フィルター」をクリックします。
3. 結果が表示されます。すべての結果を表示することも、それらをページ編集することもできます。

ヒント:

- フィルターは、項目がリンク (青の下線) になっているすべての列に対して適用されます。値がただの黒いテキストで表示されている列にフィルターを適用することはできません。
- 単一の列だけにフィルターを適用するには、フィルター・ストリングの前に列名を入力します。例えば、Selector 列の場合は、Selector: Select All などと入力します。
- 入力したフィルター・ストリングはシステムに保存されます。「フィルター」ボックスの横にある矢印をクリックすると、自分または他のユーザーが入力したフィルターのリストが表示されます。フィルター・ストリングを削除するには、ストリングを強調表示して、ストリングの右側にある「ごみ箱」アイコンをクリックします。
- フィルター・ストリングのリストには、常に「すべて表示」選択項目が表示されています。
- フィルター・テキストを入力した際、そのフィルターに一致する項目がない場合は、どの項目も表示されません。

リストをソートするには、以下のようにします。

1. 列ラベルの隣にある二重矢印をクリックします。
2. 結果が表示されます。すべての結果を表示することも、それらをページ編集することもできます。

ヒント:

- 二重矢印を再度クリックすると、ソートの順序が逆になります。

---

## Hello World プロジェクトの作成

このトピックでは、単純なプロジェクトを作成して実行し、ビルド・システムが適切にセットアップされているかどうかを確認する方法について説明します。

次の項目は、「Hello World」プロジェクトの作成に必要なタスクの概要を示しています。

1. サーバーをセットアップする:
  - a. サーバー認証 (ログイン名とパスワード) を作成して、Rational Build Forge がサーバー上のコマンドにアクセスして実行できるようにする。
  - b. エージェントがインストールされているサーバーを指定するサーバー定義を作成する。
  - c. 使用する HelloWorld プロジェクト用のセレクトターを作成して、実行するサーバーを決定する。
  - d. サーバー接続をテストして、サーバー上のマスター・コンソールとエージェントが相互に通信できることを確認する。
2. プロジェクトを定義する:
  - a. プロジェクトに「HelloWorld」という名前を付け、セレクトターを使用して、プロジェクトを実行するサーバーを指定する。
  - b. Hello World メッセージを表示するコマンドを実行するためのステップを作成する。
3. ジョブを実行する:
  - a. プロジェクトのインスタンス (ジョブとも呼ばれます) を開始する。
  - b. 左パネルで「ジョブ」項目をクリックして、ジョブ状況を確認する。
4. ジョブの結果を表示する:
  - a. 完了ジョブ・リストで、ジョブ・タグをクリックして、ジョブのステップのリスト (または、HelloWorld の例の場合は単一のステップ) を表示する。
  - b. ジョブのステップをクリックして、ステップ・ログを表示する。
  - c. ステップ・ログでコマンドの Hello World 出力を検索する。

## サーバーのセットアップ

このトピックでは、「Hello World」演習で使用するようにサーバーをセットアップして、ビルド・システムが適切にセットアップされているかどうかを確認する方法について説明します。

### サーバー認証の作成

Rational Build Forge は、サーバー認証 (ログイン名とパスワード) を使用して、エージェントがインストールされているサーバーに接続します。

### このタスクについて

そのサーバーを Rational Build Forge のリソースとして使用するよう定義する前に、サーバーに対するサーバー認証を作成する必要があります。ログイン名およびパスワードに関連付けられた特権で、Rational Build Forge プロジェクトがサーバー上で実行されるときに使用する特権が決定します。

### 手順

1. Rational Build Forge の左パネルで、「サーバー」 > 「サーバー認証」をクリックします。

管理コンソールの下部に、ブランクの「サーバー認証の詳細 (Server Authentication Details)」パネルが表示されます。

2. 「サーバー認証の追加」をクリックします。
3. 「名前」に認証の名前を入力します。認証名にはログイン名を使用します。
4. 「ログイン」で、サーバー上のアカウントのログイン名を指定します。

注: ログイン名がドメイン・ユーザー名である場合は、このフィールドにドメイン名も含めてください。例えば、MYDOMAIN/joeuser のように入力します。

5. 「パスワード」にパスワードを入力します。
6. 「パスワードの確認」にパスワードを再度入力します。
7. 「サーバー認証の保管」をクリックします。

## タスクの結果

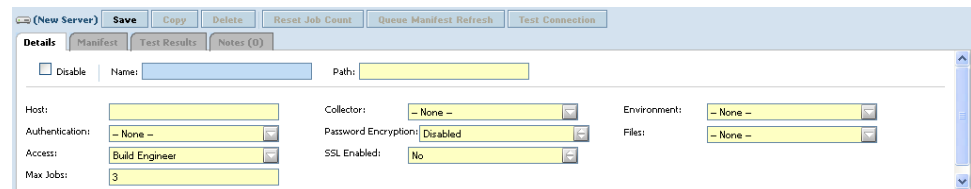
新しいサーバー認証が一番上のパネルに表示され、サーバーを定義するときに使用できるようになります。

## サーバーの作成

管理コンソールのサーバーは、プロジェクトまたはステップを実行できるホストを表します。

## 手順

1. Rational Build Forge の左パネルで、「サーバー」をクリックします。
2. 「サーバーの追加」をクリックします。
3. サーバーの詳細を指定します。



- a. **名前:** サーバーに HelloWorldServer という名前を付けます。後でセクターを定義するときに、これと同じ名前を使用します。
- b. **パス:** サーバーがプロジェクトまたはジョブのディレクトリーを作成するときに使用するディレクトリーを指定します (例: C:\%buildforgeprojects)。このパスの値は、ビルド・ディレクトリーが作成される際の開始点として使用されます。

**ヒント:** システムはサーバー・パスを作成しません。パスは、ビルドがサーバーへのアクセスを試みる前に存在している必要があります。パスが存在していないと、ビルドは失敗します。

- c. **ホスト:** エージェントを実行している物理コンピューターのホスト名を指定します。管理コンソールのコンピューターをサーバーとして定義する場合は、値 localhost を使用します。(また、エージェントも管理コンソールにインストール済みでなければなりません。)

注: ホスト名の前にプロトコルを付けないでください。例えば、`http://` を使用しないでください。

- d. 認証: このサーバーで使用するサーバー認証を選択します。
  - e. その他のフィールドはデフォルト値のままにします。
4. 「保存」をクリックします。新規サーバーはコンテンツ・パネルの上部のサーバー・リストに表示されます。
  5. サーバーが正しく構成されていることを確認するには、リストからサーバーを選択し、「接続のテスト (Test Connection)」をクリックします。

サーバーと通信できない場合は、エラーがレポートされます。エラーを受信した場合は、エージェントがサーバー上で実行されていることを確認してください。

## セレクトターの作成

セレクトターは、プロジェクトまたはステップを実行するサーバーを決定します。

### このタスクについて

HelloWorld プロジェクトがそのステップをどこで完了するかを決定できるように、セレクトターを作成します。

#### 手順

1. Rational Build Forge の左パネルで、「サーバー」 > 「セレクトター」をクリックします。
2. 「セレクトターの追加」をクリックします。
3. 「名前」に HelloWorldSelector を入力します。
4. 「保存」をクリックします。パネルの下部が変更されて、どのサーバーを選択するかを決定するセレクトター・プロパティを定義できるようになります。
5. 14 ページの『サーバーの作成』で指定したサーバー名を使用して、実行するサーバーを設定します。「名前」を BF\_NAME、「演算子」を EQ (デフォルト) に設定して、「値」に HelloWorldServer を入力します。
6. 「保存」をクリックします。

## プロジェクトの定義

このトピックでは、「Hello World」演習の一部として単純なプロジェクトを作成して、ビルド・システムが適切にセットアップされているかどうかを確認する方法について説明します。

## プロジェクトの作成

### 始める前に

HelloWorld プロジェクト定義で使用するサーバーとセレクトターをセットアップする必要があります。

#### 手順

1. 「プロジェクト」を選択します。メイン・コンテンツ・パネルの下部に、「プロジェクト詳細」パネルが表示されます。
2. 「名前」フィールドに「HelloWorld」と入力します。

3. 「セレクトター」を HelloWorldSelector に設定します。
4. 「保存」をクリックします。プロジェクト用の空のステップ・リスト、およびブランクの「ステップ詳細」ページが表示されます。『ステップの作成』で説明されているとおりに、ステップを定義します。

## ステップの作成

### 始める前に

HelloWorld プロジェクトの最初のステップを定義するには、プロジェクトの定義をすでに開始している必要があります。

### 手順

1. 「ステップ詳細」ページの「名前」フィールドで、EchoHelloWorld と入力します。
2. 「コマンド」フィールドに、選択したサーバーの標準出力に Hello World と書き込むコマンドを入力します。例えば、次のコマンドは、Windows<sup>®</sup>、Solaris、Linux<sup>®</sup>、UNIX<sup>®</sup>、および Apple Macintosh OS X の各オペレーティング・システムで機能します。  

```
echo Hello World
```
3. 「ステップの保存」をクリックします。そのステップがステップ・リストに表示されます。

## ジョブの実行


このトピックでは、「Hello World」演習の一部として、ジョブ (実行中のプロジェクトのインスタンス) を開始し、その状態を確認して、ビルド・システムが適切にセットアップされているかどうかを確認する方法について説明します。

## ジョブの開始

### 始める前に

HelloWorld プロジェクトを定義する必要があります。

### 手順

1. 「プロジェクト」を選択してプロジェクト・リストを再表示します。
2. HelloWorld プロジェクトの横にある「クイック・スタート」アイコン  をクリックします。

## ジョブ状況の確認

### 手順

1. 「ジョブ」をクリックしてから「実行中」タブをクリックして、実行中としてリストされている HelloWorld ジョブを表示します。ジョブは、完了すると「完了」タブのページに移動します。

注: HelloWorld ジョブがリストされていない場合は、次のステップに進みます。

2. 「完了」タブをクリックします。完了したジョブのリストに HelloWorld ジョブが表示されます。



## ジョブ結果の表示

このトピックでは、「Hello World」演習の一部としてジョブの結果を表示して、ビルド・システムが適切にセットアップされているかどうかを確認する方法について説明します。

### 始める前に

HelloWorld プロジェクトを定義し、実行する必要があります。

### 完了ジョブ・リストの表示

#### 手順

1. 「ジョブ」をクリックします。
2. 「完了」タブをクリックします。完了ジョブ・リストに、HelloWorld ジョブが正常に完了したものとして表示されます。

### ステップ・ログの表示

#### 手順

1. 「ジョブ」をクリックします。
2. 「完了」タブをクリックします。
3. HelloWorld ジョブのジョブ・タグをクリックします。初期ジョブのデフォルトのジョブ・タグは BUILD\_1 です。システムは、ジョブのステップの名前を表示します。この例では、ジョブには EchoHelloWorld という 1 つのステップしかありません。
4. EchoHelloWorld ステップをクリックして、そのログを調べます。ほとんどの Hello World の例では、コンソール・ウィンドウまたはポップアップ・ウィンドウに「Hello World」というテキストが表示されます。管理コンソールは、ターゲット・サーバー上のエージェント・プロセスにコマンドを送信することにより、処理を行います。次に、エージェントはそれらのコマンドからの出力を管理コンソールに戻します。その結果、ステップ・ログにその出力が保管されます。ログには数多くのセクションがあります。関係するセクションは、最後の EXEC セクションです。EXEC セクションのみを表示するには、すべてのチェック・ボックスのチェック・マークを外し、EXEC チェック・ボックスを選択してから「リフレッシュ」リンクをクリックします。このコマンドの結果は次のとおりです。

```
80 04/19/10 11:06AM EXEC Locale set to 'English_United States.1252'
253 04/19/10 11:06AM EXEC Locale set to 'English_United States.1252'
354 04/19/10 11:06AM EXEC Performing variable expansion on command line
356 04/19/10 11:06AM EXEC start [C:\buildforgeprojects\HelloWorld\BUILD_1@mcsystem]
357 04/19/10 11:06AM EXEC Hello World
358 04/19/10 11:06AM EXEC end [C:\buildforgeprojects\HelloWorld\BUILD_1@mcsystem]
```

### タスクの結果

このプロジェクトは、システムが正しく構成されており、プロジェクトがサーバーに正常にアクセスして、サーバー上で実行され、出力を生成できることを示しています。echo コマンドは、ターゲット・サーバー上で実行可能な任意のコマンドに置き換えることができます。

---

## プロジェクト・サンプル

システムに習熟するために役立つプロジェクトのサンプルが含まれています。

プロジェクト・サンプルは以下のディレクトリーにあります。

`<bfinstall>/samples/projects/`

デフォルトでは、Build Forge インストール・ディレクトリー (つまり、`<bfinstall>`) は、Windows の場合は `C:\Program Files\IBM\Build Forge`、UNIX および Linux の場合は `/opt/buildforge` です。

サンプル・プロジェクトを使用するには、以下のようにします。

- `bfimport` コマンドを使用して、管理コンソール内にインポートします。
- プロジェクトを実行します。

---

## 第 3 章 IBM Rational Build Forge の資料に関する特記事項

© Copyright IBM Corporation 2003, 2014.

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510

東京都中央区日本橋箱崎町19番21号

日本アイ・ビー・エム株式会社

法務・知的財産

知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

*Intellectual Property Dept. for Rational Software*  
*IBM Corporation*  
*5 Technology Park Drive*  
*Westford, MA 01886*  
*U.S.A.*

本プログラムに関する上記の情報は、適切な使用条件の下で 사용할 수 있지만、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

## 著作権使用許諾

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、

利便性もしくは機能性があることをほのめかしたり、保証することはできません。これらのサンプル・プログラムは特定物として現存するままの状態を提供されるものであり、いかなる保証も提供されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。 © Copyright IBM Corp. 2003, 2013.

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

## プライバシー・ポリシーに関する考慮事項

サービス・ソリューションとしてのソフトウェアも含めた IBM ソフトウェア製品 (「ソフトウェア・オファリング」) では、製品の使用に関する情報の収集、エンド・ユーザーの使用感の向上、エンド・ユーザーとの対話またはその他の目的のために、Cookie はじめさまざまなテクノロジーを使用することがあります。多くの場合、これらの「ソフトウェア・オファリング」により個人情報が収集されることはありません。一部の「ソフトウェア・オファリング」では、個人情報を収集することができます。ご使用の「ソフトウェア・オファリング」が、これらの Cookie およびそれに類するテクノロジーを通じてお客様による個人情報の収集を可能にする場合、以下の具体的事項をご確認ください。

このソフトウェア・オファリングは、展開される構成に応じて、セッション管理または認証のために、それぞれのお客様のお名前、ユーザー名、パスワード、プロファイル名、または役職を収集するセッション Cookie を使用場合があります。これらの Cookie を無効にすることはできません。

この「ソフトウェア・オファリング」が Cookie およびさまざまなテクノロジーを使用してエンド・ユーザーから個人情報を収集する機能を提供する場合、お客様は、個人情報を収集するにあたって適用される法律、ガイドライン等を遵守する必要があります。これには、エンドユーザーへの通知や同意取得の要求も含まれますがそれらには限られません。

このような目的での Cookie を含む様々なテクノロジーの使用の詳細については、IBM の『IBM オンラインでのプライバシー・ステートメント』 (<http://www.ibm.com/privacy/details/jp/ja/>) の『クッキー、ウェブ・ビーコン、その他のテクノロジー』および『IBM Software Products and Software-as-a-Service Privacy Statement』 (<http://www.ibm.com/software/info/product-privacy>) を参照してください。

## 商標の帰属表示

IBM、IBM ロゴおよび <a href="http://www.ibm.com">ibm.com</a> は、世界の多くの国で登録された International Business Machines Corp. の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、 <a href="http://www.ibm.com/legal/copytrade.shtml">www.ibm.com/legal/copytrade.shtml</a> をご覧ください。
---





---

## 第 4 章 要件

Rational® Build Forge® 製品のコンポーネントには、ハードウェア要件およびソフトウェア要件があります。

要件は、以下の分野に関するものです。

- ハードウェア、ソフトウェア、およびデータベース
- Installation Manager
- Licensing
- IPv6 サポートのネットワーキング要件
- 統合
- 各国語サポート

---

### ハードウェア、ソフトウェア、およびデータベースの要件

ハードウェア、ソフトウェア、およびデータベースの要件は、照会によるレポートとして、「Detailed system requirements for a specific product」サイトで入手できます。ご使用の製品、製品エディション、バージョン、およびオペレーティング・システムを指定してください。ハードウェア、オペレーティング・システム、データベース、エージェント・ホスト、および Web クライアントに関する要件が記載されています。

### データベースの要件

Rational® Build Forge® を使用するには、データベースがインストールされている必要があります。サポートされているデータベースについては、クリアリングハウスのハードウェア要件およびソフトウェア要件を確認してください。

### 本製品で提供されているデータベース・ライセンス

DB2 Workgroup バージョン 10.1 データベースは、パスポート・アドバンテージから別途ダウンロードすることで入手できます。ご使用の製品のライセンスには、この製品のライセンスが含まれています。製品で DB2 Workgroup 10.1 を使用する予定の場合は、これをパスポート・アドバンテージからダウンロードして、コンソールをインストールする前にインストールする必要があります。Rational Build Forge での使用に関して、プラットフォームまたはホストの場所に関する制約事項はありません。

---

### Installation Manager の要件

Installation Manager がインストールされている場合、Build Forge をインストールするには、バージョン要件を満たしている必要があります。

インストーラーにより、インストール済みの Installation Manager が検索されます。バージョン要件を満たしている必要があります。

検出されない場合、インストーラーにより Installation Manager がインストールされます。

サポートされている最小バージョンは、1.4.4 です。

## 製品エディションごとのライセンス要件

使用するライセンス交付メカニズムは、所有する製品のエディションによって異なります。 2 ページの『Build Forge 製品エディション』を参照してください。

エディション	ライセンス・メカニズム
Rational Build Forge Standard Edition	ライセンス・サーバー
Rational Build Forge Enterprise Edition	ライセンス・サーバー
Rational Build Forge Enterprise Plus Edition	ライセンス・ファイル

このトピックでは、Build Forge 用の Rational License Server 設定について詳しく説明します。 Build Forge の UI を使用して、新規ライセンス・サーバーを指定することができます。 IBM Installation Manager を使用したライセンス管理は、サポートされていません。

## インストール中のライセンス・ファイルの指定

このトピックは、Build Forge Enterprise Plus Edition (ファイル・ベースのライセンス交付を必要とします) を使用するお客様のみを対象としています。

インストール中にライセンス・ファイルを指定するには、以下のステップを実行します。

1. ライセンス・キー・ファイル **irbf\_license.properties** をパスポート・アドバンテージから取得します。これをルート・インストール・ディレクトリーに置きます。

Windows	C:\Program Files\IBM\Build Forge
UNIX/Linux	/opt/buildforge

2. Installation Manager インストーラーの「ライセンス・サーバー構成」ページで「**ライセンス・ファイル (License File)**」をクリックします。
3. 「**参照**」をクリックして、ルート・インストール・ディレクトリー内のライセンス・キー・ファイルを探します。
4. 「**irbf\_license.properties**」をダブルクリックして選択し、「**次へ**」をクリックしてインストールを続行します。

## Build Forge 用の Rational License Server の構成

ライセンス管理者が Rational® License Server をセットアップします。インストール時に指定されたライセンス・サーバー・ホスト名をライセンス管理者から受け取ってください。

このセクションは、サーバー・ベースのライセンス交付が必要な製品エディションのみに当てはまります。

- Build Forge Standard Edition
- Build Forge Enterprise Edition

## Build Forge のライセンス・サーバーの構成要件

Build Forge のライセンス・サーバーを構成する前に、以下の要件を検討してください。

- ご使用の環境のすべての管理コンソール・ホスト・コンピューターは、Rational License Server ホスト・コンピューターに接続可能でなければなりません。
- UNIX/Linux または Windows Rational License Server は、自動的に始動してサービスとして稼働するように、セットアップする必要があります。
- ライセンス・サーバーを使用するように製品をインストールして構成するには、Rational License Server のホスト名およびポート番号をライセンス管理者から取得する必要があります。

注: サポートされる Rational License Key Server のバージョンが、Rational® Build Forge® バージョン 8.0 で 7.0 から 8.1.3に変更されました。8.0 より前のバージョンからアップグレードを行う予定の場合は、ライセンス・サーバーのアップグレードも必要となります。詳しくは、「Detailed System Requirements」レポートを確認してください。

## インストール時のライセンス・サーバーの指定

インストール時に、Rational License Server のホスト名および TCP/IP ポートを指定します。この情報は、ライセンス管理者から入手してください。

サーバー・ベースのライセンスを構成するには、次のようにします。

1. Installation Manager インストーラーの「ライセンス・サーバー構成」ページで、「**ライセンス・サーバー**」をクリックします。
2. デフォルトのユーザーである *root* を受け入れるか、または Build Forge を実行する別の UNIX/Linux ユーザーを指定します。別のユーザーを指定する場合は、Build Forge を *root* で開始する必要があります。Build Forge は、インストール・ステップで指定されたユーザーでエンジンと Apache Tomcat を始動します。Installation Manager の「データベース構成」ページで指定したデータベース・ライブラリーと JDBC jar ファイルに対する読み取り権限と実行権限が必要です。
3. Rational License Server に有効なホスト名を「**ライセンス・サーバー**」フィールドに入力します。

注: ホスト名を指定すると、その情報が Build Forge システム設定に自動的に追加されます。ホスト名が分からない場合は、このフィールドに任意の文字や値を入力して、インストール完了後にユーザー・インターフェースで Build Forge システム設定を更新してください。

4. 「**ポート**」で、ライセンス・サーバーの TCP/IP ポートを入力します。デフォルト・ポートは 27000 です。
5. 「**次へ**」をクリックして、インストールを続けます。

## 管理コンソール用のライセンス・サーバーの変更

管理コンソール用の Rational License Server を変更するには、製品のライセンス・サーバー構成に以下の修正を加えます。この手順は、以下の条件が満たされている場合に実行しなければなりません。

- インストール中に、間違ったホスト名または他の値を入力した場合。
  - ライセンス・サーバーのホスト名がインストール中にグレー表示された場合。これは、FLEXlm クライアントにホストのライセンス・サーバーがすでに登録されていることを示しています。
1. このステップは、Build Forge のバージョンが 7.1 から 7.1.3.3 までの場合にのみ適用されます。これより後のバージョンの場合は、ステップ 2 に進んでください。ご使用のオペレーティング・システムに関して、**RATIONAL\_LICENSE\_FILE** 変数の値を変更してください。

Windows	RATIONAL_LICENSE_FILE は、 HKEY_LOCAL_MACHINE\SOFTWARE\FLEXlm License Manager のレジストリーにあります。
UNIX/Linux	RATIONAL_LICENSE_FILE は .flexlmrc ファイルに設定されており、これは、Build Forge を実行しているユーザーのホーム・ディレクトリーにあります。

以下のようにして、変数をライセンス・サーバーの正しいホスト名に設定します。

`port@hostname | @hostname`

ライセンス・サーバーのポートが 27000 (デフォルト) の場合、ポートを設定する必要はありません。

2. 管理コンソールの UI で、「管理」→「システム」と選択します。
3. ライセンス・サーバーの設定を見つけ、その値を、新しい Rational License Server のホスト名に設定します。

以下のフォーマットの 1 つを使用します。

`<host_name>:<port> | <host_name> | <port>:<host_name>`

ライセンス・サーバーのポートが 27000 (デフォルト) の場合、ポートを設定する必要はありません。

4. Web ブラウザーを最新表示して、管理コンソールが新しいライセンス・サーバーに接続することを確認します。

## ライセンス・キーの取得と、Rational License Server の設定

既存のライセンス・サーバーを使用できない場合、以下のテーブルによって、ライセンス管理者がライセンス・タイプを取得し、Rational License Server をセットアップするために実行する一般タスクが指定されます。

Rational License Server をインストールおよび構成するには、Rational License Server ソフトウェアのバージョンに対応した資料を確認してください。

<http://www-306.ibm.com/>に移動し、「サポートとダウンロード (Support and

downloads)」を選択して、「Rational ライセンス管理ガイド (Rational License Management Guide)」を検索してください。

ライセンス管理者のタスク	リソース
Rational License Key Center から、ライセンス・タイプを取得します。	Rational ライセンス管理ガイド (Rational License Management Guide)  Rational License Key Center のクイック・スタート・ガイド ( <a href="http://www-306.ibm.com/software/rational/support/licensing">http://www-306.ibm.com/software/rational/support/licensing</a> )
管理コンソール・ホストおよび Rational License Server のネットワーク接続を確認します。	Rational ライセンス管理ガイド (Rational License Management Guide)
以下のいずれかの方法で、Rational License Server ソフトウェアの必要なバージョンを取得します。 <ul style="list-style-type: none"> <li>製品の配布時に含まれているソフトウェアにアクセスする。</li> <li>次の IBM Rational Download and Licensing Center から、ソフトウェアをダウンロードする。</li> </ul> <a href="http://www-01.ibm.com/software/rational/support/upgrades/full-product.html">http://www-01.ibm.com/software/rational/support/upgrades/full-product.html</a>	Rational ライセンス管理ガイド (Rational License Management Guide)
Rational License Server をインストールします。	Rational ライセンス管理ガイド (Rational License Management Guide)
以下のようにして、ライセンス・タイプを Rational License Server にインストールまたはインポートします。 <ul style="list-style-type: none"> <li>Windows の場合、IBM Rational License Key Administrator (LKAD) を使用します。これは、多数の IBM Rational 製品および Rational License Server ソフトウェアとともにインストールされています。</li> <li>UNIX/Linux の場合、license_setup スクリプトおよびライセンス交付実行可能ファイルを使用します。これらは、以下の IBM Rational Download and Licensing Center から入手できます。</li> </ul> <a href="http://www-01.ibm.com/software/rational/support/upgrades/full-product.html">http://www-01.ibm.com/software/rational/support/upgrades/full-product.html</a>	Rational ライセンス管理ガイド (Rational License Management Guide)

ライセンス管理者のタスク	リソース
<p>以下のようにして、Rational License Server を開始し、Rational License Server が自動的に開始し、サービスとして実行するようにセットアップします。</p> <ul style="list-style-type: none"> <li>Windows では、Rational License Server ソフトウェアは、コンピューターが起動すると自動的にサービスとして開始するように設定されます。自動的に開始しない場合は、Windows でライセンス・サーバーを自動的に開始するための説明を確認してください。</li> <li>UNIX/Linux では、提供されるテンプレート開始スクリプトを使用して、開始スクリプトを作成し、インストール済み環境に応じてそれを修正します。次の IBM Rational Download and Licensing Center から、テンプレートを入手します。</li> </ul> <p><a href="http://www-01.ibm.com/software/rational/support/upgrades/full-product.html">http://www-01.ibm.com/software/rational/support/upgrades/full-product.html</a></p>	Rational ライセンス管理ガイド (Rational License Management Guide)

## IPv6 サポートのネットワーキング要件

Rational® Build Forge® システムは、Internet Protocol バージョン 6 (IPv6) ネットワークおよび IPv6-IPv4 混合のネットワークで稼働することができますが、いくつかの制約事項があります。

IPv6 サポートでは、ご使用のコンピューターおよびネットワークが IPv6 をサポートするように正しく構成されている必要があります。ネットワーク構成に問題があると、Build Forge® システム内で指定されたホスト名およびアドレスが正しく解決されません。

IPv6 で稼働するように、Rational® Build Forge® を手動で構成する必要があります。この方法については、104 ページの『[httpd.conf の変更](#)』を参照してください。

Build Forge® で IP アドレスを入力する際には、正しいアドレス・フォーマットを使用してください。Build Forge® で、管理者あるいはユーザーがホスト名または IP アドレスを入力するのは、以下の数箇所のみです。

- インストール時には、管理コンソールがライセンス・サーバー (Standard Edition および Enterprise Edition の場合) およびデータベースと通信するために使用するホスト名または IP アドレスを、管理者が指定します。
- エージェント構成 (bfagent.conf ファイル) に、エージェントへの接続を、特定のアドレスまたはアドレス範囲に制限するためのオプション設定があります。
- 管理コンソールのユーザー・インターフェースを表示するためには、ユーザーがブラウザに URL を入力します。この URL は、管理コンソールが実行されているサーバーのホスト名または IP アドレスによって構成されます。例えば、IPv4 と IPv6 の両方のアドレスが構成されている BFmachine という名前のサー

バーにインストールされた管理コンソールにアクセスする場合は、Web ブラウザーで以下のいずれかのアドレスを入力します。

- http://BFmachine/
- http://localhost/
- http://127.0.0.1/ (IPv4 のループバック・アドレス)
- http://::127.0.0.1/, http://0:0:0:0:0:0:127.0.0.1/ (IPv4 のループバック・アドレスの IPv6 における省略形)、または単に http://::1/ (IPv6 のループバック・アドレスの短縮表記)

IPv4 アドレスと IPv6 アドレスは、フォーマットと長さが異なります。

- **IPv4 フォーマット:** 長さは 32 ビットです。アドレスは、4 つの小数点付き 10 進数値として指定します (例えば、255.255.255.255)。
- **IPv6 フォーマット:** 長さは 128 ビットです。アドレスは、8 つの 16 進値をコロンで区切って指定します (例えば、FE80:0000:0000:0202:B3FF:FE1E:8329)。高位フィールドの使用に関しては、いくつかの規則があります。省略に関しても規則があります。Build Forge<sup>®</sup> では IP アドレスの変換処理は実行しません。それらはネットワークに直接渡されます。したがって、正しく有効なすべての省略形が機能します。IPv6 のアドレスに関する規則について詳しくは、他の解説書を参照してください。

## **/etc/hosts では、localhost 別名の使用を避ける**

IPv4 と IPv6 の両方を使用している場合、UNIX および Linux システムでは、/etc/hosts で別名として localhost を使用しないでください。サーバーの完全修飾ドメイン・ネームまたは IP アドレスを使用してください。

localhost をホスト名として /etc/hosts で使用しなければならない場合は、IPv4 アドレス 127.0.0.1 のみ定義する必要があります。

## **IPv6 をサポートしていないコンポーネント**

IPv6 をサポートしていないコンポーネントは、IPv4 アドレスを持つホスト・コンピュータにインストールする必要があります。以下のコンポーネントは、IPv4 アドレスを持つコンピュータにインストールしてください。

- Rational License Key Server 8.1.3 (製品の配布に組み込まれている Build Forge<sup>®</sup> Standard Edition および Enterprise Edition のライセンス・サーバー)。

**注:** IPv6 アドレスをサポートするには、Rational License Key Server 8.1.3 をインストールする必要があります。これは、IBM<sup>®</sup> Rational Support または IBM Rational Download and Licensing Center から入手できます。

- DB2<sup>®</sup> データベース。PHP データベース・ドライバーは、現時点では IPv6 をサポートしていません。
- MySQL データベース。PHP データベース・ドライバーは、現時点では IPv6 をサポートしていません。
- 11g 以外の Oracle データベース。PHP データベース・ドライバーは、現時点では IPv6 をサポートしていません。



---

## 各国語サポート要件

Build Forge 製品では、以下の言語についてローカライズされたサポートを提供しています: フランス語、ドイツ語、イタリア語、ブラジル・ポルトガル語、スペイン語、日本語、韓国語、中国語 (簡体字)、中国語 (繁体字)。

このセクションでは、Build Forge での言語サポートについて説明します。

### 管理コンソールおよびエージェントの言語設定

このトピックでは、管理コンソール、エージェント、およびエンジンの各コンポーネントの言語設定を、Build Forge で構成する方法について説明します。

#### 管理コンソールの言語設定

管理コンソールでのインターフェース・コントロールの表示に使用される言語は、現行ユーザーの言語設定によって決まります。

ユーザー・アカウントに対する管理コンソールの言語設定は、以下のように設定します。

##### root ユーザー

管理者が初めてコンソールにログインする際に使用する root ユーザー・アカウントは、インストール・プログラムによって作成されます。

最初のログイン時、root ユーザーの言語設定は、Build Forge エンジン・ホストのオペレーティング・システムの言語を基に決定されます。

##### 他のユーザーの言語のデフォルト

管理者によって作成されたユーザー・アカウントの場合、デフォルトのコンソール言語は、ユーザーがログイン時に使用する Web ブラウザーの言語設定を使用して最初に設定されます。

**注:** これは、Web ブラウザーの構成設定であって、Build Forge の構成設定ではありません。

管理者がユーザーの言語設定を変更しない場合は、Web ブラウザーの設定が継承されます。

管理者に言語を変更してもらいたい場合は、次の手順を実行するところまで進みます。UI で、「管理」>「ユーザー」>「言語」とクリックし、言語を選択します。

LDAP によって作成されたユーザー・アカウントでは、常に、Web ブラウザーに対して構成された言語設定が使用されます。

##### デフォルトのユーザー言語の変更

コンソールにログインした後、管理者は、Build Forge ウィンドウの左側にあるナビゲーション・パネルで「管理」>「ユーザー」>「言語」とクリックして、Build Forge ユーザーごとに異なる言語設定を選択できます。

ユーザーの言語設定をこの方法で構成すると、管理コンソールでは、Web ブラウザーに対して構成された言語とは無関係に、ユーザーに対して選択された言語でインターフェース・コントロールが表示されます。

## エージェントおよび Build Forge エンジンの言語設定

Build Forge エンジン・ホストのオペレーティング・システム言語は、以下を制御します。

- Build Forge エンジンで使用する言語
- Build Forge エージェントのデフォルト言語

エージェントの言語設定は、システム・メッセージおよびジョブ出力の言語を制御します。

言語が管理コンソールに対して設定されているか、Web ブラウザーに対して設定されているか、Build Forge 設定として設定されているかにかかわらず、エージェントは、Build Forge エンジン・ホストのオペレーティング・システム言語でシステム・メッセージとジョブ出力のデータをログに記録します。

管理コンソール・インターフェースで言語が混合しないようにするには、管理コンソール用に選択する言語が、Build Forge エンジン・ホストで使用される言語と一致するようにしてください。

### 言語のセットアップ

管理コンソールに表示される言語が、エージェントがシステム・メッセージやジョブ出力のデータを記録する際に使用する言語と一致するようにするには、Web ブラウザー、管理コンソール・ホスト上のオペレーティング・システム、および Build Forge ユーザーで同じ言語を使用してください。

## データベース・ホストでの国際データのサポート

国際データを表示および操作するには、管理コンソール・データベースのホスト・コンピュータを以下のように構成してください。

- Unicode UTF-8 文字セットを使用してください。
- データ表示に使用するフォントをインストールしてください。

## 管理コンソールにおけるユーザー言語設定の変更

すべての管理コンソール・ユーザーのデフォルトの言語は、最初は Web ブラウザーの言語に設定されます。

Build Forge ユーザーの言語設定を変更するには、「管理」>「ユーザー」>「言語」と選択します。

管理コンソールには、ユーザーについて選択された言語でユーザー・インターフェース・コントロールが表示されますが、エージェントは、システム・メッセージおよびジョブ出力のデータを、引き続き Build Forge エンジン・ホストのオペレーティング・システム言語で記録します。

## UNIX/Linux ホストの言語/文字セットの特定

管理コンソールまたはエージェントが UNIX/Linux ホストにインストールされている場合は、次のように locale コマンドを使用してください。

- オペレーティング・システムで現在使用されている言語/文字セットを特定する場合:  
\$ locale
- オペレーティング・システムで使用可能な言語/文字セットの組み合わせを特定する場合:  
\$ locale -a
- ログイン時にロケールを設定するには、rc またはプロファイル・スクリプトを使用します。

## Windows ホストの言語コード・ページの特定

管理コンソールまたはエージェントが Windows ホストにインストールされている場合は、次のように chcp コマンドを使用してください。

- アクティブなコード・ページ番号を特定するには、次のように入力します。  
> chcp
- コード・ページを設定するには、言語の番号を入力します。  
> chcp code\_page

次の表に、Build Forge でサポートされる NLV1 言語の Windows 文字エンコードをリストします。

言語	コード・ページ
英語	1252
フランス語	
スペイン語	
イタリア語	
ドイツ語	
ポルトガル語	
日本語 Shift-JIS	932
韓国語	949
中国語 (簡体字) GBK	936
中国語 (繁体字) Big5	950

---

## 第 5 章 インストールの計画

Rational® Build Forge® のインストールでは、いくつかの製品コンポーネントについて考慮する必要があります。

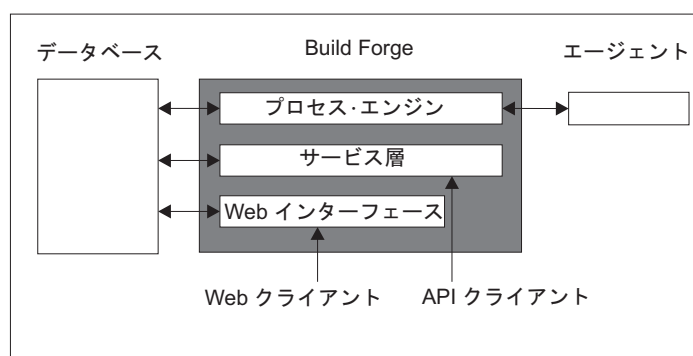
このトピックでは、Rational® Build Forge® 製品コンポーネントのインストールに必要な計画について説明します。これは、インストールと構成の方法と対象について可能な選択のロードマップとして機能します。

653 ページの『第 40 章 パフォーマンスとスケーラビリティ』に記載されているリソースの計画も参照してください。

---

### コンポーネント

Rational® Build Forge® システムには、以下のコンポーネントがあります。



- **Web クライアント:** Web ブラウザーを使用してシステムにアクセスするユーザーおよび管理者が使用します。ブラウザー・クライアントは **Web インターフェース**・コンポーネントにアクセスします。
- **API クライアント:** Rational® Build Forge® にアクセスするために Java™ API または Perl API を使用する任意のプログラム。API クライアントは、**サービス層**に直接アクセスします。
- **Build Forge:** このシステムの総称。このシステムは、インストール時に以下の中核的な製品機能を表示します。
  - **Web インターフェース:** 管理コンソールまたはコンソールと呼ばれます。このコンポーネントは、一連の PHP モジュールから構成されます。
  - **プロセス・エンジン:** エンジンと呼ばれます。エンジンは、ジョブのスケジューリングや実行を管理します。
  - **サービス層:** API クライアント、Web インターフェース、およびプロセス・エンジンが要求を行うときに使用するデータベース抽象化層。

単純なインストールでは、これらの 3 つのコンポーネントはすべて同じホスト上にインストールされます。これらのコンポーネントは別々のホストにインストー

ルすることができます。 Rational® Build Forge® 構成ファイル (例: buildforge.conf) の場合は、コンポーネントが通信できるように、このファイルを変更する必要があります。

- **データベース:** システムの情報ストレージ。データベースには、プロジェクト定義、システム構成、およびユーザー構成が保管されます。
- **エージェント:** ホストにインストールされたプログラム。管理コンソールでサーバー・リソースとして使用するすべてのホスト上にエージェントをインストールする必要があります。エージェントは、作業 (すなわち、ステップ) の実行要求を受け取って、そのエージェントがインストールされているホスト上でその作業を実行します。

コンポーネントは、さまざまな方法でデプロイできます。単一ホスト上のすべてのコンポーネントから、クラスター・コンソールおよび多くの分散サーバー・リソースを使用するシステムまで、デプロイメントのバリエーションは多岐にわたります。

---

## デプロイメントの種類

このセクションでは、以下のデプロイメントの種類について説明します。

- スタンドアロン
- 通常の実動インストール
- 段階的インストール

### スタンドアロン

スタンドアロン・デプロイメントでは、データベース、管理コンソール、および 1 つのエージェントが、単一のホストにデプロイされます。このデプロイメントは、主として評価または開発の目的で使用されます。

### 通常

通常のインストールでは、データベース、管理コンソール、および複数のエージェントが、さまざまなホストに配置されます。このデプロイメントは、主として実稼働環境に使用されます。

### 段階的インストール

段階的デプロイメントには、通常は以下の 3 つの標準的なデプロイメントがあります。

- **開発 (スクラッチまたはサンドボックスとも呼ばれます):** 開発スタッフが新規プロジェクトを作成するために使用します。アップグレードの初期領域としても使用されます。
- **テスト:** 品質保証および開発スタッフが、システムをテストするために使用します。
- **実動:** すべてのスタッフが日常業務においてジョブを実行するために使用します。

Build Forge データは 1 つの段階からエクスポートされ、次の段階にインポートされます。

**重要:** これら 3 つのデプロイメントは対称的にする必要があります。各デプロイメントで同じオペレーティング・システム、データベース、およびコンポーネントを使用してください。

---

## インストール方法

このセクションでは、ユーザーが使用できるインストール方法について説明します。

- **Installation Manager、対話式:** インストールを実行しながら入力します。
- **Installation Manager、サイレント:** 入力をファイルに保存してから、インストール時にそのファイルを使用して入力します。
- **独自のコンポーネントを使用したインストール:** Build Forge に必要な既存のコンポーネントを引き続き使用します。
- **仮想イメージへのインストール:** VMware へのインストールのガイドラインが提示されています。
- **IBM System z へのインストール:** コンソールを System z 上の SUSE Linux にインストールするために必要なツールと情報を理解します。

---

## 構成オプション

このトピックでは、製品の構成を変更するために必要な機能について説明します。

- **ポートの割り当て:** デフォルトのポート 80 以外のポートの使用方法について学習します。
- **121 ページの『デフォルト・ログイン・セキュリティについて』:** インストールの際に鍵ストアのパスワードを指定します。鍵ストアは、セキュア・ログインのメカニズム (デフォルトで指定) および SSL (追加セットアップが必要) の両方で使用されます。セキュア・ログインは、必要に応じて無効にできます。
- **パスワード暗号化:** Build Forge エンジン、エージェント、サービス層、およびデータベースで使用されるパスワードを暗号化します。
- **SSL および HTTPS:** Build Forge システム全体で SSL を使用するために必要なステップを理解します。
- **シングル・サインオン:** セキュリティを維持しながらサインオンを単純化します。
- **コンソールの冗長化を使用した高スループットのサポート:** 1 つのデータベースを使用する複数のコンソールを設定します。
- **Build Catalyst を使用した Make ベースのビルドの高速化:** 単一システムの並行ビルドと複数システムの分散ビルドをセットアップします。

---

## 他の製品との統合

- **WebSphere:** Apache Tomcat ではなく WebSphere Application Server を使用
- **Rational Team Concert IDE:** Build Forge を RTC ビルド・サーバーとしてセットアップしてその他のタスクを実行
- **Eclipse および Rational Application Developer の IDE:** Eclipse および Rational Application Developer IDE 内からコンソールにアクセスする方法を学習

- ソースにアクセスするためのアダプターの使用: 他の製品との相互作用のための、提供されているアダプターの使用と独自の設計



---

## 第 6 章 インストール前のセットアップ

このセクションでは、インストーラーを実行する前に行う必要のある、インストール前のセットアップについて説明します。

注: すでに Build Forge バージョン 7.1 以前がインストールされている場合には、223 ページの『第 12 章 以前のバージョンからのアップグレード』を参照してください。

- インストール役割 (必須)。ソフトウェアは、管理レベルのユーザーとしてインストールする必要があります。
- 国際データのサポート (必須)。Build Forge をインストールする前に、Web ブラウザーとデータベースで UTF-8 文字セットがサポートされるように設定する必要があります。
- データベースのセットアップ (必須)。Build Forge で使用するデータベースをセットアップします。通常、データベース・オブジェクト (データベース・ユーザー、パスワードなど) を作成し、Build Forge が稼働するホスト上にデータベース・クライアントをインストールして、Build Forge のインストール時に入力求められる情報を収集する必要があります。
- セキュリティー (任意)。SSL/HTTPS を使用する場合は、証明書を提供するか、Build Forge で自己署名証明書をインストールする必要があります。

---

### 国際データのセットアップ

国際データをサポートできるように、Build Forge コンポーネントをセットアップする必要があります。

#### 手順

1. Web ブラウザーを設定します。
  - a. 言語を設定します。
  - b. 正しいフォントがインストールされていることを確認してください。
2. エージェント・ホストで、UTF-8 文字セットが使用されるように設定します。

Windows では、chcp コマンドを使用してコード・ページを確認します。

```
> chcp
```

UNIX または Linux では、次のコマンドを使用して、ロケールと文字セットを確認します。

```
locale
```

ここで、使用する言語と文字セットを指定する値を確認する必要があります。以下は、言語に「米国英語」、文字セットに UTF-8 を指定した Solaris システムを使用する場合の例です。

```
LANG=en_US.UTF-8  
LC_CTYPE="en_US.UTF-8"
```

3. データベースで、国際データをサポートする UTF-8 文字セットとフォントが使用されるように設定します。

- **DB2:**

- a. コード・セットと地域を設定します。例えば、CREATE DATABASE USING CODESET UTF-8 TERRITORY US と設定します (または、コントロール・センターで適切なコード・セットと地域を選択します)。
- b. 管理コンソール・コンピュータで DB2CODEPAGE 環境変数を 1208 に設定します。

Windows の場合は以下のコマンドを使用します。

```
set DB2CODEPAGE=1208
```

UNIX または Linux の場合は以下のコマンドを使用します。

```
export DB2CODEPAGE=1208
```

UTF-8 に移行する必要があるデータが既存のデータベースにある場合は、次のサイトにある文書を参照してください。 <http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.admin.doc/doc/t0024033.htm>

- **Microsoft SQL Server** (Windows で管理コンソールを使用する場合のみ): 国際データのサポートはありません。Microsoft SQL Server では、Unicode データを保管するために UCS-2 を使用します。UTF-8 はサポートされていません。
- **MySQL:** サーバーの文字セットと照合を設定します。 インストールした MySQL で国際データが現在サポートされていない場合は、ソースから再コンパイルして、`./configure --with-charset=utf8 --with-collation=utf8_bin` を使用することができます。このサポートが構成されていない場合、Build Forge エンジンが始動しません。
- **Oracle:** インストール時に、インスタンスの文字セットを **UTF8 - Unicode 3.0** に設定します。 Database Configuration Assistant では、「文字セット (Character Sets)」タブの「初期化パラメーター (Initialization Parameters)」ステップで設定されます。コマンド行を使用する場合は、文字セットを **AL32UTF8** に設定してください。

4. (オプション) Build Catalyst 機能のメッセージを構成します。

Microsoft® Windows® プラットフォームの場合、Build Catalyst のメッセージは、自動的に適切な翻訳文の形で表示されます。

UNIX プラットフォームおよび Linux プラットフォームでこれらのメッセージを表示するには、NLSPATH 環境変数を以下の値に設定します。

```
/opt/rational/buildforge/buildcatalyst/lib/nls/%L/utf8/%N.cat:$NLSPATH
```

各部の詳細は以下のとおりです。

- %L は、LANG 環境変数の値に置き換わります。
- %N は、catopen(3C) に渡される名前パラメーターの値に置き換わります。

また、翻訳を明示的に指定することもできます。以下は、ja\_JP 翻訳の場合の値です。

---

## データベースのバックアップ (必須)

フレッシュ・インストール、再インストール、またはアップグレードを進める前に、**データベースをバックアップ**することが非常に重要です。

インストール中に問題が生じた場合は、データベース・バックアップを使用して、インストールを試行する前の状態にデータベースを復元することができます。データベース・バックアップは、ロールバックを行うために唯一提供されている手段です。

---

## データベースのセットアップ

このセクションでは、サポートされる各データベースに対応したセットアップ手順を説明します。

データベースは、管理コンソールと同じホストに置くことも、別のホストに置くこともできます。

**注:** Rational Build Forge 7.1 リリースでは、空のデータベースを使用してインストールする必要があります。データベースが空ではない場合、インストールは失敗します。

セットアップでは、データベースに応じて、以下のタスクの一部またはすべてを完了する必要があります。

- Build Forge で使用するデータベース・オブジェクト (データベース、データベース・ユーザー) を作成する。
- Build Forge で使用する必須クライアント・ソフトウェアをインストールする。
- インストール中に必要になる追加情報を判別する。通常、Apache Tomcat アプリケーション・サーバーで使用される JDBC ドライバーのロケーションなどが必要になります。
- UTF-8 文字セットおよび適切な照合がサポートされるように設定する。通常、国際データのサポートはデータベースを作成するときに指定します。国際データのサポートは、データベースの作成後に設定することはできません。データ表示に使用するフォントをインストールする必要があります。 **Build Forge** では、**国際データ (UTF-8 文字セット)** を使用する必要があります。

## DB2 のセットアップ

DB2 のサポートをセットアップするには、この手順に従ってください。

### DB2 の Red Hat および SuSe Linux 6 要件 始める前に

Red Hat Linux 6 上の DB2 では、Rational Build Forge をインストールする前に追加のセットアップが必要になります。

## 手順

- 互換性共用ライブラリーをインストールします。以下のパッケージをインストールする必要があります。

- libgcc-4.4.4-13.i686
- glibc.i686 (nss-softokn-freebl.i686 を同時にインストールします)
- gtk2.i686 (多くの追加ファイル: atk.i686、 audit-libs.i686、 avahi-libs.i686、 cairo.i686、 cups-libs.i686、 dbus-libs.i686、 expat.i686、 fontconfig.i686、 freetype.i686、 gamin.i686、 glib2.i686、 gnutls.i686、 jasper-libs.i686、 keyutils-libs.i686、 krb5-libs.i686、 libX11.i686、 libXau.i686、 libXcomposite.i686、 libXcursor.i686、 libXdamage.i686、 libXext.i686、 libXfixes.i686、 libXft.i686、 libXi.i686、 libXinerama.i686、 libXrandr.i686、 libXrender.i686、 libcom\_err.i686、 libgcrypt.i686、 libgpg-error.i686、 libjpeg.i686、 libpng.i686、 libselinux.i686、 libstdc++.i686、 libtasn1.i686、 libthai.i686、 libtiff.i686、 libxcb.i686、 pango.i686、 pixman.i686、 zlib.i686 を同時にインストールします)
- libXtst.i686
- libuuid.i686

- ファイルのアルファベット順リストは以下のとおりです。ファイル拡張子は Intel プラットフォームに対応していることに注意してください。 System z プラットフォームであれば、拡張子 .s390 を使用してください。

```
atk-1.28.0-2.el6.i686
audit-libs-2.0.4-1.el6.i686
avahi-libs-0.6.25-8.el6.i686
cairo-1.8.8-3.1.el6.i686
cups-libs-1.4.2-35.el6_0.1.i686
dbus-libs-1.2.24-4.el6_0.i686
expat-2.0.1-9.1.el6.i686
fontconfig-2.8.0-3.el6.i686
freetype-2.3.11-6.el6_0.2.i686
gamin-0.1.10-9.el6.i686
glib2-2.22.5-5.el6.i686
glibc-2.12-1.7.el6_0.5.i686
gnutls-2.8.5-4.el6.i686
gtk2-2.18.9-4.el6.i686
jasper-libs-1.900.1-15.el6.i686
keyutils-libs-1.4-1.el6.i686
krb5-libs-1.8.2-3.el6_0.7.i686
libcom_err-1.41.12-3.el6.i686
**libgcc-4.4.4-13.el6.i686
libgcrypt-1.4.5-3.el6.i686
libgpg-error-1.7-3.el6.i686
libjpeg-6b-46.el6.i686
libpng-1.2.44-1.el6.i686
libselinux-2.0.94-2.el6.i686
libstdc++-4.4.4-13.el6.i686
libtasn1-2.3-3.el6.i686
libthai-0.1.12-3.el6.i686
libtiff-3.9.4-1.el6_0.3.i686
libuuid.i686
libX11-1.3-2.el6.i686
libXau-1.0.5-1.el6.i686
libxcb-1.5-1.el6.i686
libXcomposite-0.4.1-2.el6.i686
libXcursor-1.1.10-2.el6.i686
```

```
libXdamage-1.1.2-1.el6.i686
libXext-1.1-3.el6.i686
libXfixes-4.0.4-1.el6.i686
libXft-2.1.13-4.1.el6.i686
libXi-1.3-3.el6.i686
libXinerama-1.1-1.el6.i686
libXrandr-1.3.0-4.el6.i686
libXrender-0.9.5-1.el6.i686
libXtst-1.0.99.2-3.el6.i686
nss-softokn-freebl-3.12.8-1.el6_0.i686
pango-1.28.1-3.el6_0.5.i686
pixman-0.18.4-1.el6_0.1.i686
zlib-1.2.3-25.el6.i686
```

- SuSe Linux 64 ビット・システムを操作している場合、pam.i686 パッケージもインストールする必要があります。

## DB2 のデータベース・オブジェクト

### 手順

1. オペレーティング・システムで ユーザーを作成します。管理コンソールは、この名前を使用してデータベースにログインします。例: ユーザー名 **BUILD**、パスワード **BUILD**

DB2 で残りのステップを実行します。重要: データベース名とスキーマ名には、すべて大文字を使用する必要があります。

2. **BUILD** という名前のデータベースを作成します。
3. **BUILD** という名前のスキーマを作成します。

### タスクの結果

注: Build Forge は、このユーザーのスキーマを使用してデータベースにアクセスします。

## DB2 SQL コマンド・スクリプトの例

DB2 コマンド行プロセッサで以下のコマンドを使用して、データベースを作成します。

```
// Create database and schema
db2 "CREATE DATABASE BUILD USING CODESET UTF-8
    TERRITORY US PAGESIZE 16 K AUTOCONFIGURE
    USING MEM_PERCENT 20 APPLY DB ONLY"
db2 "CONNECT TO BUILD"
db2 "CREATE SCHEMA BUILD"
db2 "CONNECT RESET"

// Grant permissions to the user
db2 "CONNECT TO BUILD"
db2 "GRANT CREATEIN,DROPIN,ALTERIN ON SCHEMA BUILD
    TO USER BUILD WITH GRANT OPTION"
db2 "GRANT DBADM,CREATETAB,BINDADD,CONNECT,
    CREATE_NOT_FENCED_ROUTINE,IMPLICIT_SCHEMA, LOAD,
    CREATE_EXTERNAL_ROUTINE,QUIESCE_CONNECT
    ON DATABASE TO USER BUILD"
db2 "commit work"
db2 "CONNECT RESET"
db2 "terminate"
```

注: Build Forge をインストールする場合、または Build Forge を前バージョンからアップグレードして DB2 データベースを使用する場合は、DB2 データベース・ページ・サイズを 16K 以上にする必要があります。ページ・サイズの変更方法については、DB2 インフォメーション・センター ([http://www.ibm.com/support/publications/us/library/index.shtml#letter\\_D](http://www.ibm.com/support/publications/us/library/index.shtml#letter_D)) を参照してください。

## DB2 で推奨されるチューニング・パラメーター このタスクについて

DB2 チューニング・パラメーターを設定すると、既存の DB2 データベースを使用した Build Forge システムのパフォーマンスおよびスケーラビリティが向上する可能性があります。

注: Build Forge をインストールして実行した後に、これらのパラメーターを変更する場合は、変更を行う前に Build Forge を停止してください。DB2 を再始動した後に Build Forge を再始動します。

### 手順

1. チューニング・パラメーターを設定します。 次の DB2 コマンドを実行します。

```
db2set DB2_EVALUNCOMMITTED=ON
db2set DB2_SKIPDELETED=ON
db2set DB2_SKIPINSERTED=ON
```

これらの設定の影響に関して詳しくは、DB2 の資料を参照してください。

2. DB2 を再始動します。 パラメーターを有効にするために、再始動する必要があります。まず、データベース上で実行中のセッションがないことを確認してください。

```
db2stop force
db2start
```

## DB2 クライアント・ドライバ

製品をインストールする前に、IBM DB2 データベース・クライアント・ドライバをホスト上にインストールする必要があります。 UNIX や Linux の場合は、32 ビットのドライバを使用します。

### 手順

1. DB2 データ・サーバー・クライアントをインストールして、ドライバを取得します。
2. DB2 クライアントをインストールした後、コンピューターを再始動します。これを行わないと、製品のインストールは失敗します。

## インストール中に必要な DB2 情報 このタスクについて

インストール中に、「データベース構成」パネルに以下の情報を入力するように要求されます。

データベース構成

- データベース・ホスト: DB2 がインストールされるホストです。

- **データベース・ポート:** Build Forge では、DB2 用に、このフィールドにデフォルトのポート 50000 が入力されます。別のポートを使用する場合は、使用するポート番号を入力してください。
- **データベース名:** Build Forge で使用されるデータベースの名前です。このデータベースは、以前のセットアップ・ステップで作成しました。
- **データベース・スキーマ名:** Build Forge で使用されるスキーマの名前。
- **データベース・ユーザー名:** データベースにアクセスする際に、Build Forge で使用されるユーザー名です。このユーザーは、以前のセットアップ・ステップで作成しました。
- **パスワード:** データベース・ユーザー名のパスワードです。

#### データベース構成のテスト

- **DB2 クライアント・ライブラリーへのパス** - DB2 クライアント・ライブラリーが置かれているディレクトリーです。

**重要:** Build Forge を UNIX または Linux にインストールする場合、このディレクトリーに 32 ビットのクライアント・ドライバー・ライブラリーが含まれている必要があります。

- **JDBC ドライバー・ロケーション** - JDBC ドライバーが置かれているディレクトリーです。Apache Tomcat は、このドライバーを使用してデータベースにアクセスします。標準的なロケーションは以下のとおりです。
  - Windows: <db2install>/IBM/SQLLIB/java
  - UNIX または Linux: システムの資料を参照してください。

以下の情報が表示されます。

- **必須ドライバー JAR ファイル** - 必須ドライバー JAR ファイルが表示されます。DB2 の場合、db2jcc.jar と db2jcc\_license\_cu.jar があります。
- **必須 JDBC ドライバー・クラス** - 必須 JDBC ドライバー・クラスが表示されます。DB2 の場合、com.ibm.db2.jcc.DB2Driver です。

## Microsoft SQL Server のセットアップ

Microsoft SQL Server データベースを Windows® ベースの管理コンソールからアクセスできるように、Microsoft SQL Server をインストールおよび構成するには、以下の手順に従います。

### Microsoft SQL Server のデータベース・オブジェクト

#### このタスクについて

以下のステップでは、データベース所有者となるユーザー、および Build Forge で使用するデータベースを作成します。必要であれば、Build Forge で使用するスキーマも作成することができます。

#### 手順

1. **データベース所有者となるユーザーを作成します。** Build Forge は、このユーザー名を使用してデータベースにログオンします。別の名前を使用する必要がある場合以外は、**build** を使用してください。ユーザーにはすべての特権が必要です。



- a. SQL Server Management Studio を開きます。
  - b. 「オブジェクト エクスプローラ」(左パネル) でデータベース・サーバーを開きます。
  - c. 「セキュリティ」フォルダーを右クリックして、「新規」 > 「ログイン」と選択します。
  - d. 「ログイン - 新規作成」ダイアログで、ログイン名を指定し、以下のようにオプションを選択します。重要: 「ユーザーは次回ログイン時にパスワードを変更する」のチェック・マークを外してください。
    - 「SQL Server 認証」を選択して、パスワードを指定します。
    - 「パスワードの期限を適用する」のチェック・マークを外します。
    - 「ユーザーは次回ログイン時にパスワードを変更する」のチェック・マークを外します。
2. データベースを作成します。 混合モードの認証を使用する必要があります。以下の対話式の例およびスクリプトの例では、デフォルト・スキーマである DBO を使用することを想定しています。
- a. SQL Server Management Studio を開きます。
  - b. 「オブジェクト エクスプローラ」(左パネル) でデータベース・サーバーを開きます。
  - c. 「データベース」フォルダーを右クリックして、「新しいデータベース」を選択します。
  - d. 「新しいデータベース」ダイアログで、以下のようにデータベースのパラメーターを指定します。
    - 「データベース名」を指定します。別の名前を使用する必要がある場合以外は、**build** を使用してください。名前は大文字小文字が区別されます。「データベース ファイル」ボックスで、データ・ファイルおよびログ・ファイルの名前が自動的に更新されます。
    - 「データベース所有者」を指定します。
      - フィールドの右側にある [...] コントロールをクリックします。
      - 「データベース所有者の選択」ダイアログで、「参照」をクリックします。
      - 作成したユーザーの名前にチェック・マークを付けて、「OK」をクリックします。
      - 「データベース所有者の選択」ダイアログで、「OK」をクリックします。
    - 「データベース ファイル」のパラメーターを指定します。「データベース ファイル」表で、以下のようになります。
      - どちらのファイルについても、「初期サイズ」を 500 (MB) に設定します。
      - どちらのファイルについても、「自動拡張」を設定します。「自動拡張」列の [...] コントロールをクリックして、ダイアログを開きます。「自動拡張を有効にする」ボックスにチェック・マークを付け、拡張サイズを 500 MB に設定します。「無制限に拡張 (Unrestricted Growth)」を選択して、「OK」をクリックします。

または、以下のスクリプトを使用してデータベースを作成することもできます。

```
CREATE DATABASE [build] ON PRIMARY
( NAME = N'build', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL.2
\MSSQL\DATA\build.mdf' , SIZE = 2048KB , FILEGROWTH = 1024KB )
LOG ON
( NAME = N'build_log', FILENAME = N'C:\Program Files\Microsoft SQL Server
\MSSQL.2\MSSQL\DATA\build_log.ldf' , SIZE = 1024KB , FILEGROWTH = 10%)
GO
EXEC dbo.sp_dbcmtlevel @dbname=N'build', @new_cmptlevel=90
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [build].[dbo].[sp_fulltext_database] @action = 'disable'
end
GO
ALTER DATABASE [build] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [build] SET ANSI_NULLS OFF
GO
ALTER DATABASE [build] SET ANSI_PADDING OFF
GO
ALTER DATABASE [build] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [build] SET ARITHABORT OFF
GO
ALTER DATABASE [build] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [build] SET AUTO_CREATE_STATISTICS ON
GO
ALTER DATABASE [build] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [build] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [build] SET CURSOR_CLOSE_ON_COMMIT ON
GO
ALTER DATABASE [build] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [build] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [build] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [build] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [build] SET READ_COMMITTED_SNAPSHOT ON
GO
ALTER DATABASE [build] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [build] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [build] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [build] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [build] SET READ_WRITE
GO
ALTER DATABASE [build] SET RECOVERY FULL
GO
ALTER DATABASE [build] SET MULTI_USER
GO
ALTER DATABASE [build] SET PAGE_VERIFY CHECKSUM
GO
USE [build]
GO
IF NOT EXISTS (SELECT name FROM sys.filegroups WHERE is_default=1 AND name = N'PRIMARY') ¥
ALTER DATABASE [build] MODIFY FILEGROUP [PRIMARY] DEFAULT
GO
```

注: READ\_COMMITTED\_SNAPSHOT 属性はオンに設定する必要があります。

READ\_COMMITTED\_SNAPSHOT がオンになっているかどうかをテストするには、次の SQL ステートメントを実行してください。

```
SELECT is_read_committed_snapshot_on FROM sys.databases WHERE name='build'
```

build 以外のデータベース名を使用する場合には、build に代えてその名前を指定します。このステートメントの結果として「1」が戻されるはずです。

ステートメントの結果として「0」が戻された場合には、以下の SQL ステートメントを実行します。

```
ALTER DATABASE [build] SET READ_COMMITTED_SNAPSHOT ON
```

### 3. ユーザーのデフォルト・データベースを設定します。

- a. SQL Server Management Studio を開きます。
- b. 「オブジェクト エクスプローラ」(左パネル) でデータベース・サーバーを開きます。
- c. 「オブジェクト エクスプローラ」で、「セキュリティ」 > 「ログイン」と開きます。
- d. 作成したユーザーを右クリックして、「プロパティ」を選択します。
- e. 「全般」ページで、「既定のデータベース」を選択します。作成したデータベースを選択します。
- f. 「OK」をクリックします。

### 4. 必要であれば、スキーマを作成してください。 デフォルト・スキーマ DBO を使用できる場合、スキーマを作成する必要はありません。デフォルトを使用することをお勧めします。 異なるスキーマを使用する場合は、そのスキーマを Build Forge データベースに作成し、所有者を作成済みのデータベース・ユーザーに割り当ててください。次の例では、スキーマ bf\_schema が作成され、build が所有者として設定されます。

```
USE build;  
GO  
CREATE SCHEMA bf_schema AUTHORIZATION build;  
GO
```

## Microsoft SQL Server での TCP/IP のセットアップ このタスクについて

管理コンソールで Microsoft SQL Server を使用するには、Microsoft SQL Server で TCP/IP を有効にしなければなりません。

MS SQL Server で TCP/IP を有効にするには、以下のようにします。

### 手順

1. MSSQLServer Configuration Manager を開きます。
2. 「SQL Server のネットワーク構成」で、「MSSQLSERVER のプロトコル」をクリックします。
3. 「TCP/IP」を右クリックし、メニューから「有効 (Enable)」を選択します。

## Microsoft SQL Server クライアントと JDBC ドライバーのインストール

### このタスクについて

Microsoft SQL Server クライアントと JDBC ドライバーを、管理コンソールのホスト上にインストールする必要があります。管理コンソールは、それらを使用してデータベースにアクセスします。

### 手順

1. SQL Native Client をインストールします。バージョン 2005.90.4035.00 が必要です。これは、Microsoft SQL Server の Service Pack 3 に含まれています。
2. JDBC ドライバーをインストールします。
  - バージョン 2005: JDBC ライブラリーのバージョン 1.2 が必要です。Microsoft の以下のサイトからダウンロードします。

<http://www.microsoft.com/downloads/details.aspx?FamilyId=C47053EB-3B64-4794-9500-81E1EC91C1BA&displaylang=en>

Microsoft のインストール処理が完了した後、JAR は以下の場所に置かれています。

`/sqljdbc_1.2/enu/sqljdbc.jar`

- バージョン 2008: JDBC ライブラリーのバージョン 3.0 が必要です。

<http://www.microsoft.com/downloads/en/details.aspx?FamilyID=%20a737000d-68d0-4531-b65d-da0f2a735707&displaylang=en>

- バージョン 2012: JDBC ライブラリーのバージョン 4.0 が必要です。

## Microsoft SQL Server の ODBC データ・ソースのセットアップ

### このタスクについて

以下のステップでは ODBC データ・ソースを作成します。このデータ・ソースは、Microsoft SQL Server で作成したデータベースにアクセスするために Windows ベースの管理コンソールによって使用されるものです。32 ビットの Windows システムまたは 64 ビットの Windows システムのどちらが稼働しているかに関係なく、32 ビットの ODBC データ・ソースを作成する必要があります。

### 手順

1. データ ソース アドミニストレータを開始します。
  - 32 ビットの Windows システムの場合: Windows の「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」>「データ ソース」と選択します。
  - 64 ビットの Windows システムの場合: `C:\Windows\SysyWOW64\odbcad32.exe` ファイルをダブルクリックします。

「データ ソース アドミニストレータ」のダイアログ・ボックスが表示されます。ここで、必要な 32 ビットの ODBC データ・ソースを 64 ビットのシステム上に作成することができます。
2. 「システム DSN」タブで、「追加」をクリックします。ドライバーのリストが表示されます。

3. ドライバーのリストから「**MS SQL Server**」を選択し、「完了」をクリックします。表示された「**ODBC セットアップ (ODBC Setup)**」ダイアログ・ボックスに、以下の情報を入力します。
  - **データ ソース名:** このデータ・ソースの名前。「データベース名」と同じでなければなりません、データベースに関連付けられたスキーマ名と同じではありません。
  - **説明:** このデータ・ソースの説明です。
  - **サーバー名:** MS SQL Server データベースがインストールされているホストの名前です。
  - **データベース名:** 上で作成したデータベース名です。

「OK」をクリックして「**ODBC セットアップ (ODBC Setup)**」を閉じてから、「OK」をクリックして「**ODBC データ ソース アドミニストレータ**」を閉じます。データ・ソースが作成されます。

## タスクの結果

以下の情報を書き留めておいてください。管理コンソールをインストールする際、インストール・プログラムで必要になります。

- **データ・ソース名:** ODBC データ・ソースの作成時に割り当てた名前です。
- **データ・ソースのタイプ:** SQL Server
- **ユーザー名:** データベース用に作成したユーザー名 (例えば、**build**) です。
- **パスワード:** ユーザー名に設定したパスワード (例えば、**build**) です。

**重要:** 「データベース名」と「データ ソース名」には同じ値を使用してください。JDBC ドライバーに関する制限により、この制約が生じます。これらが同じでないと、クイック・レポートのレポート機能およびサービス層の Java API および Perl API が機能しません。

## インストール中に必要な Microsoft SQL Server の情報 このタスクについて

インストール中に、「データベース構成」パネルに以下の情報を入力するように要求されます。

### データベース構成

- **データベース・ホスト:** SQL Server がインストールされるホストです。
- **データベース・ポート:** Build Forge では、SQL Server 用に、このフィールドにデフォルトのポート 1521 が入力されます。別のポートを使用する場合は、使用するポート番号を入力してください。
- **データベース名:** Build Forge で使用されるデータベースの名前です。このデータベースは、以前のセットアップ・ステップで作成しました。
- **データベース・スキーマ名:** Build Forge で使用されるスキーマの名前。デフォルト・スキーマ名 DBO (推奨) または以前のセットアップ・ステップで作成したスキーマの名前です。
- **ODBC データ・ソース名:** ODBC データ・ソースの名前です。

- **データベース・ユーザー名:** データベースにアクセスする際に、Build Forge で使用されるユーザー名です。このユーザーは、以前のセットアップ・ステップで作成しました。
- **パスワード:** データベース・ユーザー名のパスワードです。

データベース構成のテスト

- **SQL Server クライアント・ライブラリーへのパス** - SQL Server クライアント・ライブラリーが置かれているディレクトリーです。

**重要:** Microsoft SQL Server は、Windows プラットフォームでサポートされています。

- **JDBC ドライバー・ロケーション** - JDBC ドライバーが置かれているディレクトリーです。Apache Tomcat は、このドライバを使用してデータベースにアクセスします。
- **2005 バージョン:** 2005 バージョンで提供されるドライバを使用します。
- **2008 バージョン:** バージョン 3.0 のドライバを使用します。
- **2012 バージョン:** バージョン 4.0 のドライバを使用します。

以下の情報が表示されます。

- **必須ドライバ JAR ファイル** - 必須ドライバ JAR ファイルが表示されます。
  - SQL Server 2005 の場合、このファイルは `sqljdbc.jar` です。
  - SQL Server 2008 および 2012 の場合、このファイルは `sqljdbc4.jar` です。
- **必須 JDBC ドライバー・クラス** - 必須 JDBC ドライバー・クラスが表示されます。SQL Server の場合、ドライバ・クラスは `com.microsoft.sqlserver.SQLServerDriver` です。

## MySQL のセットアップ

MySQL のサポートをインストールして構成するには、この手順に従ってください。

### 始める前に

以下の項目をインストールおよび構成します。以下の各セクションの説明に従ってください。

## Red Hat Linux 4 での MySQL 要件

### 始める前に

Red Hat Linux 4 上の MySQL では、Rational Build Forge を構成する前に追加セットアップが必要になります。

### 手順

- `ncurses-libs.i686` をインストールします。次のステップでファイルをインストールするには、これが存在していなければなりません。
- 互換性共用ライブラリーをインストールします。パッケージまたは RPM をインストールする必要があります。両方をインストールする必要はありません。

```
MySQL-client-community-5.0.51a-0.rhel5.i386
MySQL-shared-compat-5.0.51a-0.rhel5.i386
```

- ・ 開始スクリプトを名前変更します。 開始プロセス内のデフォルトよりも早い位置で MySQL がロードされるように開始スクリプトを名前変更する必要があります。次の例のようにすれば、MySQL が十分な位置に繰り上がります。

```
mv /etc/rc3.d/S99mysql /etc/rc3.d/S50mysql
```

## MySQL のデータベース・オブジェクト

### 手順

1. **build** という名前の空のデータベースを作成します。
2. そのデータベースに関連付けられるユーザーを作成します (ユーザー名 **build**、パスワード **build**)。

### タスクの結果

データベース **build** を作成して、ユーザー **build@localhost** およびパスワード (「identified by」) **build** を作成するには、以下のコマンドを使用することができます。

```
mysql -u root
mysql> create database build;
mysql> grant all on build.* to build@localhost
-> identified by "build";
```

## MySQL ライブラリーおよびクライアント・ドライバー

### 始める前に

Build Forge をインストールする前に、MySQL ネイティブ・クライアント・ドライバーを Build Forge ホスト上にインストールする必要があります。UNIX または Linux の場合、32 ビットのドライバーを使用します。

MySQL 5.1 を使用している場合は、MySQL 5.0 クライアントを使用する必要があります。

Build Forge と同じホストに MySQL をインストールする場合は、以下をインストールします。

```
MySQL database server
MySQL-client-community-5.0.51a-0.rhel5.i386 - client drivers
MySQL-shared-compat-5.0.51a-0.rhel5.i386 - required compatibility libraries
Connector/J JDBC driver
```

異なるホストに MySQL をインストールする場合は、以下を Build Forge ホストにインストールします。

```
MySQL-client-community-5.0.51a-0.rhel5.i386 - client drivers
MySQL-shared-compat-5.0.51a-0.rhel5.i386 - required compatibility libraries
Connector/J JDBC driver
```

### AIX システムの場合のみ:

Connector/J ドライバーおよび 5.0-51a クライアント・ドライバーをインストールする必要があります。これらは AIX バージョン 6.1 と互換性があります。これらは、MySQL Web サイトの「Archives」域にあります。



インストーラーは、MySQL クライアント共用ライブラリーの再パッケージ化を試みます。ファイルが `mysql.com` にポストされている場合は、それらのファイルのパッケージ化を Build Forge で使用することはできません。インストーラーの実行ユーザーには、クライアント・ドライバー・ファイルがインストールされているディレクトリー（通常は `/opt/mysql`）への書き込み権限が必要です。インストーラーは、`libmysqlclient.a` ファイルおよび `libmysqlclient.so.15` ファイルを置換します。ファイルを再パッケージ化できない場合、インストーラーはインストール中にユーザーに通知して、インストールを続行します。その場合は、Build Forge を実行する前に、ファイルを手動で再パッケージ化しておく必要があります。

**注:** この MySQL クライアントを使用するその他のアプリケーションを実行している場合は、別個のディレクトリーにファイルを手動で再パッケージ化することが必要になる場合があります。

ファイルを再パッケージ化するには、以下の手順を実行します。

1. MySQL をインストールしたディレクトリーから、新しいディレクトリー `libbf` を作成します。

```
root@myaix:/opt/mysql/> mkdir libbf
```

2. この新しいディレクトリーに `lib/libmysqlclient.so.15` ファイルをコピーします。

```
root@myaix:/opt/mysql/> cp lib/libmysqlclient.so.15 libbf
```

3. 新しいディレクトリーに移動し、新しいアーカイブ・ファイルをビルドします。

```
root@myaix:/opt/mysql/> cd libbf
```

```
root@myaix:/opt/mysql/libbf/> ar -q libmysqlclient.a libmysqlclient.so.15
```

これで、このディレクトリーをインストール・プロセスのクライアント・ライブラリー・パスとして使用できるようになります（インストール・プロセスがまだ実行されていない場合）。サイレント・メソッドまたはコマンド行メソッドのいずれかが使用されている場合は、このパスを使用するように `$BFROOT/rc/buildforge` で `LIBPATH` を更新してください。

## MySQL の構成

### 手順

1. データベースの最大接続数を 200 に引き上げます。 `<mysql-installdir>/my.ini` (Windows の場合) または `/etc/my.cnf` (UNIX/Linux の場合) の `[mysqld]` セクションを以下のように編集します。

```
max_connections=200
```

この値は、少なくとも「**最大コンソール・プロセス**」システム設定と「**実行キュー・サイズ**」システム設定（管理コンソールの「**管理**」→「**システム**」ページにあります）の合計と等しくなければなりません。

2. ご使用の MySQL インスタンスが InnoDB ストレージ・エンジンを使用していることを確認します。 Rational Build Forge データベースには外部キー制約があります。 MySQL MyISAM ストレージ・エンジンでは、外部キー制約は検査されません。 MyISAM を使用している場合は、新しいデータベース・オブジェクトの作成時（特に API を使用時）に問題が発生するおそれがあります。

## インストール中に必要な MySQL 情報

### このタスクについて

インストール中に、「データベース構成」パネルに以下の情報を入力するように要求されます。

#### データベース構成

- **データベース・ホスト:** MySQL がインストールされるホストです。
- **データベース・ポート:** Build Forge では、MySQL 用に、このフィールドにデフォルトのポート 3306 が入力されます。別のポートを使用する場合は、使用するポート番号を入力してください。
- **データベース名:** Build Forge で使用されるデータベースの名前です。このデータベースは、以前のセットアップ・ステップで作成しました。
- **データベース・ユーザー名:** データベースにアクセスする際に、Build Forge で使用されるユーザー名です。このユーザーは、以前のセットアップ・ステップで作成しました。
- **パスワード:** データベース・ユーザー名のパスワードです。

#### データベース構成のテスト

- **MySQL クライアント・ライブラリーへのパス** - MySQL クライアント・ライブラリーが置かれているディレクトリーです。

**重要:** Build Forge を UNIX または Linux にインストールする場合、このディレクトリーに 32 ビットのクライアント・ドライバー・ライブラリーが含まれている必要があります。

- **JDBC ドライバー・ロケーション** - JDBC ドライバーが置かれているディレクトリーです。Apache Tomcat は、このドライバーを使用してデータベースにアクセスします。

以下の情報が表示されます。

- **必須ドライバー JAR ファイル** - 必須ドライバー JAR ファイルが表示されます。MySQL の場合、これは `mysql-connector-java-5.*-bin.jar` です。
- **必須 JDBC ドライバー・クラス** - 必須 JDBC ドライバー・クラスが表示されます。MySQL の場合、これは `com.mysql.jdbc.Driver` です。

## MySQL のトラブルシューティング

### このタスクについて

このタスクでは、MySQL データベースを使用しているコンソールで、メッセージ「データベース拡張 'mysqli' が見つかりません」が出されて障害が発生した場合の回避策について説明します。この問題は、Red Hat Linux バージョン 6.x で次のファイルが使用されている場合に発生します。

`/opt/buildforge/server/apache/lib/php/extensions/no-debug-non-zts-20090626`

#### 手順

1. 欠落しているライブラリーおよびアクセス権について確認します。 `ldd` コマンドを使用してください。応答の例:

```
ldd: warning: you do not have execution permission for './mysql.so'
linux-gate.so.1 => (0x00413000)
libmysqlclient.so.15 => not found
.....
libssl.so.6 => not found
libcrypto.so.6 => not found
```

2. libmysqlclient.so.15 ライブラリーを見つけられるよう、以下の作業のいずれかを実行します。

- libmysqlclient.so.15 の絶対パスを LD\_LIBRARY\_PATH に追加します。
- libmysqlclient.so.15 を /usr/lib にコピーします。

3. 古いライブラリーへのリンクを追加します。古いライブラリーへのリンクは、手動で追加する必要があります。例:

```
ln -s /usr/lib/libcrypto.so.1.0.0 /usr/lib/libcrypto.so.6
ln -s /usr/lib/libssl.so.1.0.0 /usr/lib/libssl.so.6
```

4. コンソールを再始動します。再始動が必要です。

## Oracle のセットアップ

Oracle データベースのサポートをセットアップするには、この手順に従ってください。

### 始める前に

以下の項目をインストールまたは構成します。以下の各セクションの説明に従ってください。

## Oracle 10 の Red Hat Linux 6 要件

### 始める前に

Red Hat Linux 6 上の Oracle 10 では、Rational Build Forge をインストールする前に追加のセットアップが必要になります。

### 手順

互換性共用ライブラリーをインストールします。以下のパッケージをインストールする必要があります。

- compat-libs-5.2-1.i386.rpm
- compat-libstdc++-33-3.2.3-68.i686

## Oracle のデータベース・オブジェクト

### このタスクについて

Oracle ホスト上にローカル・ユーザー (ユーザー名 **build**、パスワード **build**) を作成します。

- 所要の権限 (CREATE SESSION や CREATE TABLE など) を追加します。
- DEFAULT TABLESPACE に所要の QUOTA サイズを追加し、システムがデータを保管できるだけの十分なスペースを用意します。

```
create user build
  identified by password
  default tablespace users
```

```
quota unlimited on users;  
  
grant create session, create table  
to build;
```

**重要:** インストール時、データベース名と Oracle SID には同じ値が使用されます。JDBC ドライバーに関する制限により、この制約が生じます。これらが同じでない、クイック・レポートのレポート機能およびサービス層の Java API および Perl API が機能しません。

## Oracle で推奨されるチューニング・パラメーター このタスクについて

Build Forge を正常に稼働させるには、Oracle パラメーターの一部をデフォルト値から変更する必要があります。

**注:** Build Forge をインストールして実行した後に、これらのパラメーターを変更する場合は、変更を行う前に Build Forge を停止してください。データベース・サーバーを再始動した後に、Build Forge を再始動します。

### 手順

1. チューニング・パラメーターを設定します。 次のコマンドを実行してください。

```
ALTER SYSTEM SET open_cursors=1000 SCOPE=BOTH  
ALTER SYSTEM SET processes=500 SCOPE=BOTH
```

**注:** ALTER SYSTEM SET processes=500 SCOPE=BOTH を実行した際に「SQL エラー : ORA-02095: 指定された初期設定パラメーターは変更できません」というメッセージが表示された場合は、代わりに ALTER SYSTEM SET processes=500 SCOPE=SPFile を実行してください。

これらの設定が及ぼす影響について詳しくは、Oracle の資料を参照してください。

2. データベース・サーバーを再始動します。 パラメーターを有効にするために、再始動する必要があります。まず、データベース上で実行中のセッションがないことを確認してください。

## インストール中に必要な Oracle 情報 このタスクについて

インストール中に、「データベース構成」パネルに以下の情報を入力するように要求されます。

### データベース構成

- **データベース・ホスト:** Oracle がインストールされるホストです。
- **データベース・ポート:** Build Forge では、Oracle 用に、このフィールドにデフォルトのポート 1521 が入力されます。別のポートを使用する場合は、使用するポート番号を入力してください。
- **データベース名:** Build Forge で使用されるデータベースの名前です。このデータベースは、以前のセットアップ・ステップで作成しました。

- **データベース・ユーザー名:** データベースにアクセスする際に、Build Forge で使用されるユーザー名です。このユーザーは、以前のセットアップ・ステップで作成しました。
- **パスワード:** データベース・ユーザー名のパスワードです。

データベース構成のテスト

- **Oracle クライアント・ライブラリーへのパス** - Oracle クライアント・ライブラリーが置かれているディレクトリーです。

**重要:** Build Forge を UNIX または Linux にインストールする場合、このディレクトリーに 32 ビットのクライアント・ドライバ・ライブラリーが含まれている必要があります。

- **ORACLE\_HOME 環境変数** - Oracle がインストールされるディレクトリーです。
- **tnsnames.ora ファイル (TNS\_ADMIN) へのパス** - tnsnames.ora ファイルが含まれているディレクトリーです。 tnsnames.ora ファイルに関する全アクセス権限が設定されていることを確認してください。
- **JDBC ドライバ・ロケーション** - JDBC ドライバが置かれているディレクトリーです。Apache Tomcat は、このドライバを使用してデータベースにアクセスします。

以下の情報が表示されます。

- **必須ドライバ JAR ファイル** - 必須ドライバ JAR ファイルが表示されます。
  - Oracle 10 の場合、このファイルは ojdbc14.jar です。
  - Oracle 11 の場合、このファイルは ojdbc5.jar です。
- **必須 JDBC ドライバ・クラス** - 必須 JDBC ドライバ・クラスが表示されます。Oracle の場合、これは oracle.jdbc.driver.OracleDriver です。

## Oracle クライアントの構成 このタスクについて

Oracle のクライアントをインストールして構成するには、以下のようになります。

### 手順

1. 管理コンソール・ホストに Oracle Instant Client をインストールします。実行対象のプラットフォーム (32 ビットまたは 64 ビット) に関係なく、32 ビットのバージョンをインストールする必要があります。ご使用のオペレーティング・システムに対応する **Instant Client Package - Basic** のバージョン 10.2 または 11.2 のみを使用してください。
  - Oracle 10.2 にはバージョン 10.2 クライアントを使用してください。
  - Oracle 11g にはバージョン 11.2 クライアントを使用してください。

クライアントは、Oracle (<http://www.oracle.com/technology/software/tech/oci/instantclient/index.html>) からダウンロードしてください。

2. 管理コンソール・ホスト上で環境をセットアップします。いくつかの環境変数を設定しなければなりません。

- **LD\_LIBRARY\_PATH:** クライアントのインストール・ディレクトリーを含むように設定します。

注: Installation Manager から、この値を「データベース構成」ページに設定することができます。

- **NLS\_LANG:** 国際言語をサポートする適切な値に設定します。
  - UNIX システムおよび Linux システムの場合: AL32UTF8 が値に含まれている必要があります。例: AMERICAN\_AMERICA.AL32UTF8。
  - Windows システムの場合: Windows レジストリーの「現在のコード ページ」の設定に対応する文字マップ仕様が値に含まれている必要があります。『Windows における Oracle クライアントの構成例』を参照してください。

**NLS\_LANG** は、説明のとおり明示的に設定する必要があります。クライアントのインストール時に設定されるデフォルトの文字セットは、*Build Forge* での使用には適していません。

- **ORACLE\_HOME:** Oracle クライアントのインストール・ディレクトリーのパスに設定します。

注: Installation Manager から、この値を「データベース構成」ページに設定することができます。

- **ORA\_NLS10:** サーバー上の 文字セット・データが置かれているパスに設定します。
- **PATH:** クライアントのインストール・ディレクトリーを含むように設定します。
- **TNS\_ADMIN:** サーバー上の tnsnames.ora ファイルが置かれているパスに設定します。tnsnames.ora ファイルに関する全アクセス権限が設定されていることを確認してください。

注: Installation Manager から、この値を「データベース構成」ページに設定することができます。

Oracle サーバーの現行の言語設定を確認するには、Oracle にログインして、次のコマンドを実行します。

```
SQL> host echo $NLS_LANG
```

## 次のタスク

**重要:** インストール時、tnsnames.ora は、データベース名と Oracle SID に同じ値を使用するようにセットアップされます。JDBC ドライバーに関する制限により、この制約が生じます。これらが同じでないと、クイック・レポートのレポート機能、およびサービス層の Java API と Perl API が機能しません。

### Windows における Oracle クライアントの構成例: このタスクについて

環境例:

- Instant Client - Basic (C:\instantclient\_11\_2)。米国英語を使用。

- Windows システム上の Oracle 11.2 (C:\oracle\product\11.2.0\db\_1)。国際データに対応。

クライアントおよび Build Forge がインストールされているシステムの変数設定:

- LD\_LIBRARY\_PATH (C:\instantclient\_11\_2% を含める)
- NLS\_LANG=AMERICAN\_AMERICA.WE8MSWIN1252
- ORACLE\_HOME=C:\instantclient\_11\_2%
- ORA\_NLS10=C:\oracle\ocommon\nls\admin\data
- PATH (C:\instantclient\_11\_2% を含める)
- TNS\_ADMIN=C:\oracle\product\11.2\db\_1\network\admin

### Windows システム上の NLS\_LANG について

Windows 上の Oracle クライアントは、Windows の設定を使用してローカル文字マッピングを実行します。

HKKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Nls\CodePage\ACP

クライアントがサーバーにデータを送信する場合、サーバーの文字マッピングで文字が送信されます。

上記の NLS\_LANG 設定の例は、ACP 設定の 1252 に対応しています。これは、米国向け Windows のデフォルトです。その他の言語については、Oracle InstantClient の資料を参照して、NLX\_LANG の適正な言語、ロケール、および文字マップの部分を確認してください。Windows システム上で実際の ACP 設定を調べ、設定の文字マッピング部分を一致させてください。

その他の言語の文字マップは、以下のとおりです。適正な言語およびロケールと組み合わせで使用し、NLS\_LANG の設定を作成してください。

```
1250 EE8MSWIN1250
1251 CL8MSWIN1251
1252 WE8MSWIN1252
1253 EL8MSWIN1253
1254 TR8MSWIN1254
1255 IW8MSWIN1255
1256 AR8MSWIN1256
1257 BLT8MSWIN1257
1258 VN8MSWIN1258
874 TH8TISASCII
932 JA16SJIS
936 ZHS16GBK
949 KO16MSWIN949
950 ZHT16MSWIN950
```

**注:** 文字マッピングが正しく設定されていない場合でも、警告は表示されません。最初のうちは、動作やパフォーマンスに大きな変化はありません。クライアントは、メモリーで文字変換を試みます。メモリー不足の場合や、変換を実行するためにスワップする必要がある場合は、クライアントとサーバー間の通信のパフォーマンスは著しく低下します。

### UNIX または Linux における Oracle クライアントの構成例:



## このタスクについて

環境例:

- Instant Client - Basic (/usr/local/instantclient\_11\_2)。米国英語を使用。
- UNIX システム上の Oracle 11.2 (/usr/local/oracle/product/11.2.0/db\_1)。国際データに対応。

クライアントおよび Build Forge がインストールされているシステムの変数設定:

- LD\_LIBRARY\_PATH (Solaris、Linux) または LIBPATH (AIX)  
(/usr/local/instantclient\_11\_2 を含める)
- NLS\_LANG=AMERICAN\_AMERICA.AL32UTF8
- ORACLE\_HOME=/usr/local/instantclient\_11\_2
- ORA\_NLS10=/usr/local/oracle/ocommon/nls/admin/data
- PATH (/usr/local/instantclient\_11\_2 を含める)
- TNS\_ADMIN=/usr/local/oracle/product/11.2/db\_1/network/admin

## Oracle Instant Client 11.2 を使用する UNIX システムおよび Linux システムの場合

Build Forge アプリケーションは、バージョン 10.2 クライアントの一部である libclntsh.so.10.1 ライブラリーに依存しています。インストールの際に、インストーラーはこのファイルが ORACLE\_HOME にあるかどうかを調べます。このファイルがない場合、インストーラーはバージョン 11.2 のクライアントが使用されていると見なし、libclntsh.so.10.1 から対応するバージョン 11.2 ライブラリー libclntsh.so.11.1 へのシンボリック・リンクの作成を試みます。

```
ln -s libclntsh.so.11.1 libclntsh.so.10.1
```

**重要:** Build Forge インストーラーの実行ユーザーには、ORACLE\_HOME ディレクトリへの書き込み権限が必要です。これがサイトで問題になる場合は、Build Forge をインストールする前に、権限があるユーザーが手動でリンクを作成するようにしてください。一部のサイトでは、読み取り専用としてマウントされているファイル・システムにクライアントがインストールされていることがあります (Solaris システムではよく見られます)。

## Oracle RAC を使用したフェイルオーバーの実装

Oracle RAC を使用している場合にコンソール・フェイルオーバーを有効にする手順を説明します。

### 始める前に

Oracle RAC 用の基本的なデータベース・セットアップは、他の Oracle データベースのセットアップと同じです。それらの設定値を使用してコンソールをインストールします。コンソールがインストールされた後で、以下の手順を使用してフェイルオーバーを実装します。

## このタスクについて

Oracle RAC フェイルオーバーを Build Forge と一緒に使用できるようにするには、追加構成が必要です。次の 2 つの方式が使用可能です。

- SCAN 方式: buildforge.conf を変更します。SCAN (単一クライアント・アクセス名 (single client access name)) は Oracle 11g の機能です。これを使用した場合、クラスター内のすべてのノードがフェイルオーバー適格になります。少なくとも 3 つのノードが使用可能であり、ラウンドロビン・ベースでフェイルオーバーするように設定されている必要があります。
- 明示接続ストリング方式: Oracle Instant Client 用に tnsnames.ora ファイルを変更し、buildforge.conf を変更します。この方式では、クラスターおよび必要な各クラスター・ノードを表すサービスを指定します。少なくとも 3 つのノードが使用可能であり、ラウンドロビン・ベースでフェイルオーバーするように設定されている必要があります。LOAD\_BALANCE パラメーターと FAILOVER パラメーターを設定する必要があります。

Oracle DBA から、サービスの名前と使用するノードのリストの提供を受けることが必要な場合があります。Oracle Instant Client 用の tnsnames.ora ファイルを構成し、buildforge.conf で JDBC 接続ストリングを設定します。

どちらの方式でも、クラスター全体を表す SERVICE\_NAME を指定する必要があります。この点で、非 RAC 構成と異なります。非 RAC 構成では、SERVICE\_NAME は一般的には SID と同じです。

## 手順

### • SCAN 方式

1. buildforge.conf を編集します。buildforge.conf に db\_url パラメーターを追加し、JDBC 接続ストリングをパラメーターとして使用します。サービスは、Oracle (mysvc.mycorp.com) でセットアップする必要があり、複数のノードを必要とします。見やすくするために改行が挿入されています。buildforge.conf では改行を行わないでください。

```
db_hostname myoracledbhost.mycompany.com
db_url jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)
(HOST=myoracledbhost.mycompany.com)(PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=mysvc.mycorp.com)))
db_tcp_port 1521
db_database orclrac
db_username
db_password
db_type odbc
db_provider oracle
```

2. すべての場所で buildforge.conf を更新します。

#### – Windows

```
<bfinstall>
<bfinstall>%Apache%tomcat%work%Catalina%localhost%jas%eclipse%plugins
%com.ibm.jas-xxx.jar
```

#### – UNIX または Linux

```
<bfinstall>/Platform
bfinstall>/server/tomcat/work/Catalina/localhost/jas/eclipse/plugins/
com.ibm.jas-xxx.jar
```

- 明示接続ストリング方式 この例では、以下の名前が使用されています。

- サービス名: myorasvc.mycorp.com
- ノード名: clusnode-1vip、clusnode-2vip、clusnode-3vip、clusnode-4vip

1. tnsnames.ora を編集します。

```
myorasvc.mycorp.com=
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=TCP)(HOST=clusnode-1vip)(PORT=1521))
  (ADDRESS=(PROTOCOL=TCP)(HOST=clusnode-2vip)(PORT=1521))
  (ADDRESS=(PROTOCOL=TCP)(HOST=clusnode-3vip)(PORT=1521))
  (ADDRESS=(PROTOCOL=TCP)(HOST=clusnode-4vip)(PORT=1521))
  (LOAD_BALANCE=yes)(FAILOVER=ON)
  (CONNECT_DATA=(SERVER=DEDICATED)
  (SERVICE_NAME=myorasvc.mycorp.com)
  (failover_mode=(type=select)
  (method=basic)(retries=180)(delay=5)))
)
```

2. buildforge.conf を編集します。db\_url パラメーターを JDBC 接続ストリングに設定します。見やすくするために改行が挿入されています。buildforge.conf では改行を行わないでください。

```
db_url jdbc:oracle:thin:@(DESCRIPTION=(LOAD_BALANCE=on)
  (ADDRESS=(PROTOCOL=TCP)(HOST=clusnode-1vip)(PORT=1521))
  (ADDRESS=(PROTOCOL=TCP)(HOST=clusnode-2vip)(PORT=1521))
  (ADDRESS=(PROTOCOL=TCP)(HOST=clusnode-3vip)(PORT=1521))
  (ADDRESS=(PROTOCOL=TCP)(HOST=clusnode-4vip)(PORT=1521))
  (CONNECT_DATA=(SERVICE_NAME=myorasvc.mycorp.com)))
```

3. すべての場所で buildforge.conf を更新します。

- Windows

```
<bfinstall>
<bfinstall>%Apache%tomcat%work%Catalina%localhost%jas%eclipse%plugins
%com.ibm.jas-xxx.jar
```

- UNIX または Linux

```
<bfinstall>/Platform
bfinstall>/server/tomcat/work/Catalina/localhost/jas/eclipse/plugins/
com.ibm.jas-xxx.jar
```

## インストール中に必要な Oracle 情報

### このタスクについて

インストール中に、「データベース構成」パネルに以下の情報を入力するように要求されます。

#### データベース構成

- **データベース・ホスト:** Oracle がインストールされるホストです。
- **データベース・ポート:** Build Forge では、Oracle 用に、このフィールドにデフォルトのポート 1521 が入力されます。別のポートを使用する場合は、使用するポート番号を入力してください。
- **データベース名:** Build Forge で使用されるデータベースの名前です。このデータベースは、以前のセットアップ・ステップで作成しました。
- **データベース・ユーザー名:** データベースにアクセスする際に、Build Forge で使用されるユーザー名です。このユーザーは、以前のセットアップ・ステップで作成しました。
- **パスワード:** データベース・ユーザー名のパスワードです。

#### データベース構成のテスト

- **Oracle クライアント・ライブラリーへのパス - Oracle クライアント・ライブラリーが置かれているディレクトリーです。**

**重要:** Build Forge を UNIX または Linux にインストールする場合、このディレクトリに 32 ビットのクライアント・ドライバ・ライブラリーが含まれている必要があります。

- **ORACLE\_HOME 環境変数** - Oracle がインストールされるディレクトリです。
- **tnsnames.ora ファイル (TNS\_ADMIN) へのパス** - tnsnames.ora ファイルが含まれているディレクトリです。 tnsnames.ora ファイルに関する全アクセス権限が設定されていることを確認してください。
- **JDBC ドライバ・ロケーション** - JDBC ドライバが置かれているディレクトリです。Apache Tomcat は、このドライバを使用してデータベースにアクセスします。

以下の情報が表示されます。

- **必須ドライバ JAR ファイル** - 必須ドライバ JAR ファイルが表示されます。
  - Oracle 10 の場合、このファイルは ojdbc14.jar です。
  - Oracle 11 の場合、このファイルは ojdbc5.jar です。
- **必須 JDBC ドライバ・クラス** - 必須 JDBC ドライバ・クラスが表示されます。Oracle の場合、これは oracle.jdbc.driver.OracleDriver です。

---

## セキュリティのセットアップ

インストール中に、セキュリティをどのようにセットアップするかに関して質問されます。

- **鍵ストアのパスワード:** 鍵ストアのパスワードを指定する必要があります。これは、セキュア・ログイン (資格情報の暗号化) の有効化、および HTTPS/SSL を有効化するための開始点として使用されます。
- **証明書:** 個人証明書をインストールするか、すでに所有している証明書をインポートすることができます。
- **セキュア HTTP:** HTTPS/SSL が有効に設定された Apache サーバをインストールするかどうかを質問されます。選択した証明書が使用されます。デフォルトの 443 以外のポート番号を使用する必要がある場合、この時点でポート番号を入力する必要があります。

### 提供される個人証明書の使用

提供される証明書には、以下の属性セットが含まれています。

- **Subject DN:** "CN=hostname"。 *hostname* はインストールを実行するホストの完全修飾名です。
- **有効期限:** 10 年間 (3650 日として表示)。 この値は変更できます。有効期限は、1 年間から 2 年間が一般的です。有効期限をこれより長く設定すると、キーの推測を試みるセキュリティ・アタックに対するぜい弱性が高まります。

提供される証明書を変更することができます。証明書を変更する場合、以下のフィールドを指定することができます。

- 共通名 (必須)

- 地域
- 都道府県
- 組織名
- 国名/地域名 (必須)
- 住所

共通名および国名/地域名は X500Principal タイプの Subject DN に連結され、証明書の作成中に指定されます。

インストーラーが作成する鍵ストアに使用するパスワードを入力するように、プロンプトが出されます。このパスワードを記録します。HTTPS/SSL のセットアップを完了する必要があります。

**重要:** 後でパスワードを変更することは可能ですが、かなり長いプロセスを経る必要があります。ローカル側の要件を十分に満たす、複雑で安全性の高いパスワードを使用してください。

## 独自の証明書の使用

証明書を所有している場合、SSL を使用するシステムで、すべてのコンポーネントと接続で使用されるように、その証明書をインポートすることができます。

- 証明書は、Build Forge をインストールするホスト上で有効である必要があります。証明書を一時ディレクトリーにコピーします。インストール中に、完全修飾パスを入力するようにプロンプトが出されます。
- インストール中に、鍵ストアのパスワードを指定する必要があります。
- 証明書は、PKCS12 鍵ストア・タイプである必要があります。証明書が別のタイプである場合、OpenSSL openssl ユーティリティー、または JDK keytool ユーティリティーを使用して、コピーを PKCS12 に変換することができます。
- インストーラーが作成する鍵ストアに使用するパスワードを入力するように、プロンプトが出されます。このパスワードを記録します。HTTPS/SSL のセットアップを完了する必要があります。

**重要:** 後でパスワードを変更することは可能ですが、かなり長いプロセスを経る必要があります。ローカル側の要件を十分に満たす、複雑で安全性の高いパスワードを使用してください。

---

## 第 7 章 管理コンソールのインストール

このセクションでは、Windows、UNIX、および Linux の各プラットフォームに管理コンソールをインストールする方法について説明します。Installation Manager でサポートされているインストール・シナリオに関しては、以下の手順を使用してください。

1. 37 ページの『第 6 章 インストール前のセットアップ』に記載されているインストール前のセットアップを実行します。次のような作業を行います。
  - 国際データのサポートのセットアップ (必須)。
  - データベースのセットアップ (必須)。通常、このステップには、データベース・オブジェクトの作成、データベース・クライアントのインストール、および Installation Manager におけるインストール・ステップ中に必要な情報の収集が含まれます。
  - セキュリティーのセットアップ (任意、必要に応じて実行)。
2. ランチパッドを開始して、Installation Manager を実行します。

注: ランチパッドにより、Installation Manager を実行するホスト上に Installation Manager があるかどうか探索されます。Installation Manager が存在しない場合、ランチパッドにより、パッケージ化された Installation Manager が実行され、ホストに Installation Manager がインストールされます。次に、インストールされた Installation Manager を使用して、Build Forge がインストールされます。必要に応じて、ランチパッドを使用せずに、手動で Installation Manager をインストールすることもできます。

3. Installation Manager で、インストール・ステップを実行します。
4. 必須のインストール後の確認を行います。

以下のシナリオに対するインストール手順に関しては、75 ページの『第 8 章 代替のインストール方法』を参照してください。

- Build Forge で提供およびインストールされるアプリケーションではなく、以下の必須アプリケーションの独自のインストールを使用する場合
  - Apache HTTP Server
  - PHP
  - Apache Tomcat
- IBM Installation Manager を使用して、製品コンポーネントのサイレント・インストールを行う場合
- VMware にインストールする場合
- System z 上の SUSE Linux に管理コンソールをインストールする場合

---

## ランチパッドを使用した Installation Manager の開始

ランチパッドを使用して、Installation Manager を開始し、Build Forge をインストールします。

以下のようにして、ランチパッドを開始します。

- 製品 DVD からランチパッドを開始します。
- ダウンロードしたファイル・パッケージから、ランチパッドを開始します。

ランチパッドにより、Installation Manager がホストにインストールされているかどうかを検出されます。

- Installation Manager が存在する場合、ランチパッドにより Installation Manager が開始されます。Installation Manager で、インストールする **Build Forge** パッケージを選択してください。

**重要:** Installation Manager のバージョンが最小バージョン要件を満たしている必要があります。23 ページの『Installation Manager の要件』を参照してください。

- Installation Manager が存在しない場合、ランチパッドはパッケージ化された Installation Manager を使用します。この中で、インストールする **Installation Manager** パッケージと **Build Forge** パッケージの両方を選択してください。パッケージ化された Installation Manager により、Installation Manager がインストールされ、次に Installation Manager を使用して、Build Forge がインストールされます。

Windows ホストの場合、ランチパッドを使用して、ローカル・ホストにエージェントをインストールすることもできます。Windows 以外のオペレーティング・システムでは、ランチパッドを使用してエージェントをインストールすることはできません。エージェントのインストールに関して詳しくは、157 ページの『第 10 章 エージェントのインストール』を参照してください。

### 製品 DVD からのランチパッドの開始

以下の手順に従って、製品 DVD からランチパッドを開始します。

ご使用のオペレーティング・システムに適した手順を選択してください。

- UNIX または Linux
  1. Build Forge をインストールするホストに、最初の DVD を挿入します。
  2. ドライブをマウントします。
  3. ドライブのルート・ディレクトリーで、launchpad.sh を実行します。
  4. Installation Manager にインストールするパッケージを選択します。
    - IBM Installation Manager がホストで検出されると、ランチパッドにより Installation Manager が開始されます。

最初の「パッケージのインストール (Install Packages)」ページで、「**Build Forge**」パッケージを選択し、「次へ」をクリックします。



- IBM Installation Manager が検出されない場合、パッケージ化された Installation Manager が始動し、Installation Manager がインストールされます。次に、Installation Manager を使用して、Build Forge がインストールされます。

最初の「パッケージのインストール (Install Packages)」ページで、「Installation Manager」パッケージと「Build Forge」パッケージを選択し、「次へ」をクリックします。

- Windows

1. Build Forge をインストールするホストに、最初の DVD を挿入します。
2. 自動実行が有効になっている場合、ランチパッドは自動的に開始します。自動実行が無効になっている場合、ドライブのルート・ディレクトリーで、launchpad.exe を実行します。
3. Installation Manager にインストールするパッケージを選択します。
  - IBM Installation Manager がホストで検出されると、ランチパッドにより Installation Manager が開始されます。

最初の「パッケージのインストール (Install Packages)」ページで、「Build Forge」パッケージを選択し、「次へ」をクリックします。

- IBM Installation Manager が検出されない場合、パッケージ化された Installation Manager が始動し、Installation Manager がインストールされます。次に、Installation Manager を使用して、Build Forge がインストールされます。

最初の「パッケージのインストール (Install Packages)」ページで、「Installation Manager」パッケージと「Build Forge」パッケージを選択し、「次へ」をクリックします。

## ダウンロードしたパッケージからのランチパッドの開始

以下の手順に従って、インストール・パッケージをダウンロードし、ランチパッドを開始します。

1. IBM パスポート・アドバンテージから、Build Forge をインストールするホストの一時ディレクトリーに、ご使用のオペレーティング・システムに対応したインストール・パッケージをダウンロードします。
2. ダウンロードしたファイルからローカル・ディレクトリーに、インストール・イメージを抽出します。ファイルの内容は、ローカル・ディレクトリーに抽出されます。
3. 以下を実行して、ファイルを抽出したディレクトリーからランチパッド・プログラムを開始します。
  - Windows の場合: launchpad.exe を実行
  - UNIX/Linux の場合: launchpad.sh を実行
4. Installation Manager にインストールするパッケージを選択します。
  - IBM Installation Manager がホストで検出されると、ランチパッドにより Installation Manager が開始されます。

最初の「パッケージのインストール (Install Packages)」ページで、「**Build Forge**」パッケージを選択し、「次へ」をクリックします。

- IBM Installation Manager が検出されない場合、パッケージ化された Installation Manager が始動し、Installation Manager がインストールされます。次に、Installation Manager を使用して、Build Forge がインストールされます。

最初の「パッケージのインストール (Install Packages)」ページで、「**Installation Manager**」パッケージと「**Build Forge**」パッケージを選択し、「次へ」をクリックします。

---

## Installation Manager でのインストール・ステップ

IBM Installation Manager を使用して、大部分のプラットフォーム上に製品コンポーネントをインストールします。

### 始める前に

Installation Manager を開始し、インストールする **Build Forge** パッケージを選択してから、以下の手順を実行してください。

### このタスクについて

プロンプトに従って、目的のパッケージをインストールします。

### 手順

1. パッケージのインストール 「**Build Forge**」チェック・ボックスおよび「**バージョン**」チェック・ボックスを選択します。 Installation Manager がまだインストールされていない場合は、「**Installation Manager**」チェック・ボックスを選択します。選択したら、「次へ」をクリックします。
2. ご使用条件のページ – ご使用条件に同意することを指定して、「次へ」をクリックします。
3. パッケージのインストール: ロケーション - パッケージ・グループ - インストール・パッケージをインストールするディレクトリーを選択し、「次へ」をクリックします。デフォルトのロケーションは以下のとおりです。
  - Windows: C:\Program Files\IBM\Build Forge
  - Linux および AIX: /opt/buildforge
  - Solaris: /opt/buildforge\_0
4. パッケージのインストール: フィーチャー - デフォルトで、Web インターフェース、プロセス・エンジン、およびサービス層の 3 つの中核的な製品モジュールのすべてがインストールされます。「次へ」をクリックします。
5. パッケージのインストール: ライセンス・サーバー構成 - 「**Run-As**」ユーザー、およびコンソールで使用するライセンスのタイプを選択します。以下の情報を入力して、「次へ」をクリックします。
  - UNIX および Linux – プロンプト「**Build Forge を稼働させるユーザーはどれですか? (Which user should Build Forge run as?)**」で、デフォルトのユーザー (root) を受け入れるか、別のユーザーを指定します。このユーザーが、Build Forge エンジンと、指定された Apache Tomcat を開始します。こ

のユーザーには、Installation Manager でこの後表示されるページで指定されるデータベース・ライブラリーと JDBC JAR ファイルの読み取り権限および実行権限が必要です。

- **Rational License Server** ベース: Rational License Server のホスト名を入力します。
  - ライセンス・サーバーの有効なホスト名を入力します。

**重要:** 後でホスト名を入力する場合でも、このフィールドを空白のままにしないでください。このフィールドに文字または値を入力してください。このフィールドを空白のままにすると、製品が不完全なものになり、使用できなくなることがあります。

インストールが完了した後で、正しいホスト名を入力してください。詳しくは、24 ページの『Build Forge 用の Rational License Server の構成』を参照してください。

- ライセンス・サーバーが表示されていても、グレー表示されている場合、FLEXlm ライセンス・クライアントはホストのライセンス・サーバーをすでに登録しています。

インストールが完了した後で、正しいホスト名を入力してください。詳しくは、26 ページの『管理コンソール用のライセンス・サーバーの変更』を参照してください。

- **ファイル・ベース:** ライセンス・ファイルをダウンロードしたロケーションを参照します。

6. データベース構成 データベースを構成するには、以下の手順を実行します。

- a. 構成をテストするには、JAS のデータベース接続を構成し、外面の下部付近にある「**接続のテスト**」をクリックします。テストに合格したら、「**次へ**」をクリックします。
- b. データベース・クライアント・ライブラリーを構成して、「**接続のテスト**」をクリックします。テストに合格したら、「**次へ**」をクリックします。
- c. Build Forge をインストールする OS プラットフォーム、およびインストールするデータベースに応じて、特定の情報を指定する必要があります。以下に関しては、39 ページの『データベースのセットアップ』のデータベース・セットアップ手順を参照してください。

- DB2 の場合は、39 ページの『DB2 のセットアップ』を参照してください。
- Microsoft SQL Server の場合は、43 ページの『Microsoft SQL Server のセットアップ』を参照してください。指定する JDBC ドライバーは、選択するバージョン (SQL Server 2005 または SQL Server 2008) に応じて異なります。
- MySQL の場合は、49 ページの『MySQL のセットアップ』を参照してください。
- Oracle の場合は、53 ページの『Oracle のセットアップ』を参照してください。インストールした Oracle Instant Client のバージョンに対応するバージョン (Oracle 10 または Oracle 11) を選択する必要があります。

注: UNIX および Linux プラットフォームで、Oracle、DB2、または MySQL データベースを使用している場合は、32 ビット・データベース・クライアント・ドライバーをインストールして使用する必要があります。Installation Manager のデータベース構成ページで、使用する特定のデータベース・タイプに対応するように、データベース構成ページの「[DB2|Oracle|MySQL] クライアント・ライブラリーへのパス」フィールドに 32 ビット・バージョンのドライバー・ライブラリーが指定されていることを確認します。また、DB2 の場合は、db2profile をロードします。

7. パッケージのインストール: アプリケーションおよび Web サーバー構成 – 4 ページのそれぞれで要求された情報を入力し、各ページを入力するごとに「次へ」をクリックします。

a. Web サーバー/PHP 構成

- **独自の Web サーバーを指定しますか?** 独自の Web サーバーを指定する場合は、「はい」を選択します。デフォルトは「いいえ」です。
- **Web サーバーが使用するポートはどれですか?** Web サーバーでポート 80 を使用しない場合は、別のポート番号を入力します。デフォルトは 80 です。
- **セキュア HTTP を使用しますか?** セキュア HTTP を使用する場合は、「はい」を選択します。デフォルトは「いいえ」です。
- **PHP のメモリーの上限:** デフォルトを使用しない場合は、PHP に対するメモリーの上限を入力します。デフォルトは 256 MB です。

b. アプリケーション・サーバー構成

- **独自のアプリケーション・サーバーを指定しますか?** 既にインストールされているアプリケーション・サーバーを使用するように Build Forge を構成する場合は、「はい」を選択します。デフォルトは「いいえ」です。

注: 「はい」を選択するのは、既にインストールされている 1 つ以上のコンポーネントを使用するように Build Forge を構成する場合のみです。通常、これらのコンポーネントは、Build Forge と一緒にインストールされます。75 ページの『独自のコンポーネントを使用したインストール』を参照してください。

- **アプリケーション・サーバーが HTTP に使用するポートはどれですか?** アプリケーション・サーバーが使用するポートを入力します。デフォルトは **8080** です。
- **アプリケーション・サーバーが SSL に使用するポートはどれですか?** アプリケーション・サーバーが SSL に使用するポートを入力します。デフォルトは **8443** です。
- **アプリケーション・サーバーが AJP に使用するポートはどれですか?** アプリケーション・サーバーが AJP に使用するポートを入力します。デフォルトは **8009** です。
- **アプリケーション・サーバーがシャットダウンに使用するポートはどれですか?** アプリケーション・サーバーがシャットダウン時に使用するポートを入力します。デフォルトは **8005** です。

- **アプリケーション・サーバーの完全修飾ドメイン・ネーム (Fully qualified Domain Name of the Application Server)** アプリケーション・サーバーの完全修飾ドメイン・ネームを入力します。例:  
**WIN-TB4MVA22E7P.cn.ibm.com。**
- c. サービス構成 「サービス構成」パネルで、以下のフィールドを入力します。
- **listen するポート** - このチェック・ボックスを選択します。サービス・ポートとして、一方または両方のポートを選択する必要があります。デフォルトは **3966** です。
  - **listen するセキュア・ポート** - このチェック・ボックスを選択します。サービス・ポートとして、一方または両方のポートを選択する必要があります。デフォルトは **49150** です。
- d. セキュリティー構成 インストーラーによって Build Forge 用に作成される鍵ストアのパスワードを、以下のフィールドに入力して確認します。

注: これらのフィールドが表示されない場合は、表示されるまでスクロールダウンしてください。

- **鍵ストアのパスワード** 証明書を含む鍵ストアのパスワードを入力します。
- **パスワードの確認** 証明書を含む鍵ストアのパスワードを再入力します。デフォルトのセキュア・ログイン (資格情報の暗号化) を有効にするには、パスワードが必要です。HTTPS/SSL を使用するように Build Forge を構成する場合にも必要になります。

**重要:** 後でパスワードを変更することは可能ですが、かなり長いプロセスを経る必要があります。ローカル側の要件を十分に満たす、複雑で安全性の高いパスワードを使用してください。

プロンプト「**デフォルトを変更するか、またはカスタム証明書をアップロードしますか?**」で、以下のオプションから選択します。

- インストーラーが作成する現状のままの自己署名証明書を使用する場合:
  - 1) プロンプト「**デフォルトを変更するか、またはカスタム証明書をアップロードしますか?**」で「**いいえ**」を選択します。以下のプロンプトが表示されます。
  - 2) **既存のセキュア証明書はありますか?** 「**いいえ**」を選択します。
- インストーラーが作成する自己署名証明書を使用するが、フィールドを変更する場合:
  - ・ 証明書用の追加フィールドが表示されます。フィールドに値を入力し、「**次へ**」をクリックします。
  - 1) プロンプト「**デフォルトを変更するか、またはカスタム証明書をアップロードしますか?**」で「**はい**」を選択します。
  - 2) プロンプト「**既存のセキュア証明書はありますか?**」で「**いいえ**」を選択します。

- 自分の証明書のロケーションを指定する。証明書はホスト上にあり、pkcs12 フォーマットである必要があります。また、証明書がある鍵ストアの既存パスワードを入力する必要があります。以下を実行します。
  - 1) **鍵ストアのパスワード** 証明書を含む鍵ストアのパスワードを入力します。
  - 2) **パスワードの確認** 証明書を含む鍵ストアのパスワードを再入力します。
  - 3) タイプ **pkcs12** の鍵ストアで署名済み証明書を指定します。

**注:** 自己署名証明書を持つセキュアなサイトへのアクセスに関して、通常、ブラウザーは警告を出します。通常、ユーザーは続行するかどうかを選択することができますが、例外を許可する必要がある場合もあります。

8. パッケージのインストール: コンソール開始オプション - 必要な情報を入力します。

- a. データベース・ポートを指定します。デフォルトはポート **3306** です。
  - **デスクトップ上にショートカットを作成しますか?** デフォルトで、デスクトップ上にショートカットが作成されます (Windows の場合)。

**注:** Linux では、デスクトップ・ショートカットは作成されません。

- b. 「**データベース名**」フィールドにデータベース名を入力します。
- c. 「**データベース・ユーザー名**」フィールドで、データベースへのログオン時に使用するユーザー名を入力します。
- d. 「**パスワード**」フィールドに、データベースへのログオンに必要なパスワードを入力します。
- e. 「**パスワードの確認**」フィールドで、パスワードを再入力します。
- f. 「**JDBC ドライバー・ロケーション**」フィールドで「**参照**」をクリックし、JDBC ドライバーのロケーションを探して選択します。 必須ドライバ JAR ファイルおよび必須 JDBC ドライバー・クラスが表示されます。
- g. データベース構成の接続をテストするには、「**接続のテスト**」をクリックします。
- h. テストに合格したら、「**次へ**」をクリックします。 コンソール開始オプションの次のパネルが表示されます。
- i. 「データベース・クライアント・ライブラリー構成情報 (Database Client Library Configuration Information)」パネルで「**参照**」をクリックして、ご使用の SQL クライアント・ライブラリーへのパスを検索して選択します。
- j. データベース接続を再度テストするには、「**接続のテスト**」をクリックします。 接続のテストの結果が表示されます。
- k. 「**次へ**」をクリックします。

9. パッケージのインストール: 要約の検討 - このページの要約情報を検討して、Build Forge コンポーネントがインストールされる場所を確認し、「**インストール**」をクリックします。

- a. ターゲット・ロケーション
  - **パッケージ・グループ名:** デフォルトのパッケージ名は、`buildforge.console` です。



- **インストール・ディレクトリー:** デフォルトのインストール・ディレクトリーは、C:\Program Files\IBM\Build Forge です。
  - **共用リソース・ディレクトリー:** デフォルトの共用リソース・ディレクトリー。デフォルトは以下のとおりです。
    - Windows: C:\Program Files\IBM\SDP70Shared
    - UNIX および Linux: /opt/IBM/SDP70Shared
  - b. **フィーチャー**
    - **Build Forge のフィーチャー** インストールするフィーチャーまたはモジュールを確認します。例えば、中核的な製品モジュールは、Web インターフェース、プロセス・エンジン、およびサービス層です。
  - c. **環境**
    - デフォルトの環境は英語です。
  - d. **リポジトリ情報**
    - **次のロケーションからファイルが取得されます。** このセクションを使用して、リポジトリ・ロケーションを検討および確認してください。
  - 10. **管理コンソールにアクセスします。** ブラウザーを起動します。管理コンソールの URL にアクセスします。
    - 一般的な形式: `http://<hostname>[:<portnumber>]` のような形式になります。HTTP のデフォルトであるポート 80 を使用している場合は、ポート番号は任意です。
    - ローカル: 管理コンソールを実行している同一ホスト上でブラウザーを実行する場合は、`http://localhost` を使用します。
- 注:** インストール後にすぐにログインできない場合は、しばらく待ってから再試行してください。最初の起動時には、エンジン (bfengine) がデータベース・スキーマをロードする必要があります。
- 重要:** インストール直後に bfengine を停止しないでください。そのようにすると、データベース・スキーマが壊れる場合があります。データベース・スキーマが壊れている場合、データベースからすべての Build Forge® テーブルをドロップし、Build Forge を再インストールする必要があります。
11. **ログインします。** ユーザー名 **root** とパスワード **root** を使用してログインします。root パスワードはすぐに変更してください。

---

## インストール後のチェックリスト

このセクションでは、Rational® Build Forge® システムをインストールした後にすべきことについて説明します。

- **PATH 変数を確認します。**
- **管理コンソールがプロキシ・サーバーを介してデータベースにアクセスする必要がある場合、PHP が使用するプロキシ・サーバーを指定します。**
- **JVM メモリーを設定します。**
- **ログ・メッセージをマイグレーションします。**



## Windows の PATH 変数の確認

Windows の PATH 環境変数には、データベース・クライアントまたはドライバ DLL へのパスが組み込まれていなければなりません。次のデータベースに適した PATH を手動で確認してください。

- DB2 - db2cli.dll および sqlar.dll を含むディレクトリー
- MySQL - libmysql.dll を含むディレクトリー
- Oracle - oci.dll を含むディレクトリー

Microsoft SQL Server の場合は、ODBC 接続をセットアップすることによって、この要件が対処されます。

## プロキシ・サーバーの指定

このステップはオプションです。このステップは、管理コンソールがプロキシ・サーバーを使用してデータベースにアクセスしなければならない場合にのみ必要です。プロキシ・サーバーを使用するように PHP を構成する必要があります。

- `php.ini` ファイルを編集します。このファイルは、`bfinstall/Apache/php` にあります。例えば、`C:\Program Files\IBM\Build Forge\Apache\php` などです。

次の項目を追加します。

```
bf_proxyHost=<使用するプロキシ・サーバーのホスト名>
bf_proxyPath=<プロキシのパス>
bf_symlinkPath=<symlink からプロキシへのパス>
```

## JVM メモリーの設定 (クイック・レポートに必要)

このステップはオプションです。ただし、このステップは、ご使用のエディション (Standard Edition、Enterprise Edition、Enterprise Plus Edition) にクイック・レポートが含まれている場合は必要です。

- JVM の最大メモリーを 1 MB 以上に設定します。レポートの実行には、最小で 1 GB (1024 MB) のヒープ・サイズが必要です。レポート実行中 (大容量のレポートを実行する場合など) に「メモリー不足」エラーが発生した場合は、この設定値を増やしてください。この設定値を大きくする場合は、ホストにメモリーを追加しなければならないことがあります。

```
JAVA_OPTS -Xmx1024M
```

## ログ・メッセージのマイグレーション

Rational® Build Forge® は、以前のリリースとは異なる方法でジョブ・メッセージを保管します。そのため、Rational® Build Forge® のインストール済み環境が以前のリリースからのアップグレードである場合、ジョブ出力ページの「メッセージ」列は、そのジョブが完全にマイグレーションされるまで一時的にブランクになることがあります。小規模で優先度の低いバッチでこのようなことが起こると、マイグレーションに時間がかかる場合があります。ジョブにブランク・メッセージがある場合は、そのジョブのメッセージが完全にマイグレーションされるまで待つことをお勧めします。

# Linux のファイル・ハンドル数の増加

## 始める前に

**重要:** 最適な結果を得るため、Rational 製品を使用する前に、Rational Build Forge で使用できるファイル・ハンドル数を増やします。この変更は、必要に応じてシステム管理者が行うようにしてください。

以下の手順に従って Linux でファイル記述子を増やす際には、慎重に行ってください。手順に従わないと、コンピューターが正常に開始しなくなる場合があります。最適な結果を得るために、システム管理者がこの手順を行うようにしてください。

ファイル記述子を増やすには、以下の手順に従います。

## 手順

1. root としてログインします。root 特権を持っていない場合は、取得してから続行してください。
2. etc ディレクトリーに移動します。
3. initscript シェル・スクリプトを探します。ファイルを開くか、Linux テキスト・エディターを使用して作成します。

**重要:** コンピューター上で、initscript ファイルを空のままにしないでください。このファイルを空のままにすると、コンピューターは、次に起動するとき、または再起動するときに開始しなくなります。

4. 最初の行に、大部分の Linux コンピューターのデフォルトである 1024 より確実に大きい数字を ulimit として設定します。

```
ulimit -n 4096
```

**注意:** ulimit の値が大きすぎると、システム全体のパフォーマンスに影響を及ぼす場合があります。

5. 2 番目の行に、「eval exec "\$@"」と入力します。
6. シェル・スクリプトを保存して、閉じます。

## タスクの結果

ulimit コマンドに関して詳しくは、ulimit のマニュアル・ページを参照してください。



---

## 第 8 章 代替のインストール方法

このセクションでは、管理コンソールをインストールするための代替方法について説明します。以下のシナリオについての説明があります。

- 以下のアプリケーションについて、Build Forge によって提供され、インストールされるものではなく、独自のインストール済み環境を使用する場合
  - Apache HTTP Server および PHP
  - Apache Tomcat
- IBM Installation Manager を使用して、製品コンポーネントのサイレント・インストールを行う場合
- VMware にインストールする場合
- System z 上の SUSE Linux にインストールする場合

---

### 独自のコンポーネントを使用したインストール

以下のコンポーネントをインストール済みであり、Rational® Build Forge® によって提供されるものではなく、既にあるコンポーネントを使用したい場合は、このセクションの説明に従って、必要なテクノロジーをセットアップします。

Build Forge は、以下の必要なコンポーネントおよびテクノロジーを自動的にインストールして、構成します。

- Apache HTTP Server および PHP
- Apache Tomcat

**重要:** 使用する各コンポーネントのバージョンは、製品で提供されるコンポーネントと同じでなければなりません。詳しくは、23 ページの『ハードウェア、ソフトウェア、およびデータベースの要件』を参照してください。

Build Forge コンポーネントをインストールするには、Build Forge 要件に合わせて各コンポーネントのインストール済み環境を構成し、Installation Manager を実行してください。インストール時に、既にセットアップしてあるテクノロジーを使用するためのオプションが提供されます。

以下のセクションでは、各テクノロジーを Build Forge で使用するためのセットアップ方法、およびそれらを使用するための Build Forge のインストール方法について説明します。

### 前提条件

UNIX または Linux でインストールを行うには、以下が必要です。

- インターネット・アクセス。Build Forge をインストールするコンピューターからインターネットにアクセスできない場合は、アクセス権のあるコンピューターからファイルをダウンロードして、Build Forge コンピューターにそのファイルを転送してからステップを完了する必要があります。

- ご使用のプラットフォーム上で機能している有効な C コンパイラー (例えば、Linux 上の gcc コンパイラー)。
- ご使用のコンパイラーの製造元が推奨する make 機能 (例えば、gcc で使用する gnu-make)。
- root としての特権。
- SSL を使用するには、Open SSL をコンパイルする必要があります。

## 国際データのサポート

管理コンソールで国際データをサポートするように Build Forge をセットアップする必要があります。

### 始める前に

- **Web ブラウザー:**
  - 言語が設定されている必要があります。
  - データ表示に使用するフォントがインストールされている必要があります。
- **エージェント**

Build Forge では、エージェント・サーバーで UTF-8 文字セットを使用することをお勧めします。

UNIX/Linux 上では、次のコマンドを使用して、ロケールと文字セットを確認します。

```
locale
```

ここで、使用する言語と文字セットを指定する値を確認する必要があります。以下は、言語に「米国英語」、文字セットに UTF-8 を指定した Solaris システムを使用する場合の例です。

```
LANG=en_US.UTF-8
LC_CTYPE="en_US.UTF-8"
```

- **すべてのデータベース:**

国際データのサポートは通常、データベースを作成するときに指定します。データベースの作成後に、国際データのサポートを構成することはできません。

データ表示に使用するフォントが、データベース・ホスト・コンピューターにインストールされている必要があります。

Build Forge では、国際データ (UTF-8 文字セット) を使用する必要があります。

- **DB2:**
  1. コード・セットと地域を設定します。例えば、CREATE DATABASE USING CODESET UTF-8 TERRITORY US と設定します (または、コントロール・センターで適切なコード・セットと地域を選択します)。
  2. 管理コンソール・コンピューターで DB2CODEPAGE 環境変数を 1208 に設定します。

Windows の場合は以下のコマンドを使用します。

```
set DB2CODEPAGE=1208
```

UNIX または Linux の場合は以下のコマンドを使用します。

```
export DB2CODEPAGE=1208
```

UTF-8 に移行する必要があるデータが既存のデータベースにある場合は、次のサイトにある文書を参照してください。 <http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.admin.doc/doc/t0024033.htm>

- **MySQL:** サーバーの文字セットと照合を設定します。 インストールした MySQL で国際データが現在サポートされていない場合は、ソースから再コンパイルして、`./configure --with-charset=utf8 --with-collation=utf8_bin` を使用することができます。このサポートが構成されていない場合、Build Forge エンジンが始動しません。
- **Oracle:** インストール時に、インスタンスの文字セットを **UTF8 - Unicode 3.0** に設定します。 Database Configuration Assistant では、「文字セット (Character Sets)」タブの「初期化パラメーター (Initialization Parameters)」ステップで設定されます。

## データベースのインストールと構成

このセクションでは、独自のコンポーネントを使用して、インストール済みの Build Forge 管理コンソールで使用するデータベースのインストールについて説明します。

### 始める前に

データベースは、他のテクノロジーおよび Build Forge をインストールして構成する前に、データベース・オブジェクトを使用してインストールおよび構成する必要があります。 以下を実行してください。

通常は、次の作業を完了する必要があります。

- 使用するデータベース・システムを識別します。そのデータベース・システムが Build Forge でサポートされているかどうかを確認し (39 ページの『データベースのセットアップ』を参照)、データベース・ホストと Build Forge ホストの間で必要なネットワーク接続が確立されているかどうかを検証します。データベースへのアクセスにプロキシ・サーバーが必要な場合は、プロキシ・サーバー名とパスを取得してください。
- 国際データを使用する場合は、UTF-8 文字セットを使用できるようにデータベースが構成されていることを検証してください。
- データベース・オブジェクトおよび権限を作成します。通常は、以下のような項目になります。
  - データベース: 例では、**build** という名前を付けましたが、別の名前を使用してもかまいません。
  - データベース・ユーザー: 管理コンソールではこのユーザー名を使用して、データベースにアクセスします。 例では、**build** という名前を付けましたが、別のユーザー名を使用してもかまいません。
  - データベース・ユーザーのパスワード
  - **build** データベースに表スペースを作成するための **build** ユーザー用の権限。データの作成、変更、削除には、所有者権限が必要になります。

データベースのタイプに応じた具体的な手順が提供されています。

## Apache HTTP Server のインストールと構成

Installation Manager は、Build Forge 用の Web サーバーとして Apache HTTP Server をインストールおよび構成します。提供された Apache HTTP Server を使用することが、Build Forge 用の Web サーバーを構成する最も速い方法です。

この標準構成に代わる方法として、Build Forge によってインストールおよび構成された Apache HTTP Server の代わりに、既存の Apache HTTP Server を構成することができます。この説明では、ユーザーが自身のオペレーティング・システム上に Apache HTTP Server をセットアップおよび構成した経験があることを前提としています。

既存の Apache HTTP Server を使用するには、ご使用のインストール済み環境を以下のように変更します。

1. ご使用の Apache HTTP Server 構成ファイル (httpd-vhosts.conf) を、Build Forge アプリケーションを指すように変更します。
2. PHP をインストールし、Apache HTTP Server、ご使用の Build Forge データベース、およびパスワード暗号化 (パスワード暗号化のセキュリティー機能を使用したい場合) に必要な PHP モジュールを構成します。
3. ご使用のデータベース用に Apache を構成します。

### Installation Manager を使用した Build Forge のインストール

Installation Manager の「アプリケーションおよび Web サーバー構成」ページで、「独自の Web サーバーを指定しますか」というプロンプトが出されたら、「はい」を選択します。

### 前提ソフトウェア

- Apache HTTP Server 2.2.4
- PHP 5.2.4

### Apache サーバー構成ファイルの編集

1. ご使用のサーバーの extras ディレクトリーにある Apache httpd-vhosts.conf ファイルを見つけます。

```
cd <apache-dir>/conf/extras/  
vi httpd-vhosts.conf
```

2. Apache httpd-vhosts.conf ファイルを編集します。Build Forge に関する情報を httpd-vhosts.conf に追加するには、以下の行を追加します。

```
<VirtualHost *:80>  
    ServerAdmin build@yourdomain.com  
    DocumentRoot /opt/buildforge/webroot/public  
    ServerName ausbuild01.yourdomain.com  
    ServerAlias build.yourdomain.com mc.yourdomain.com  
    ErrorLog logs/ausbuild.error_log  
    CustomLog logs/ausbuild.access_log common  
</VirtualHost>
```

3. Build Forge Web アプリケーションを指すように DocumentRoot 設定を変更します。この例では、Build Forge のインストール・ディレクトリーは /opt/buildforge です。



4. ポートを 80 のままにするか、または、ローカルで Apache HTTP Server を実行しているポートに変更します。

```
<VirtualHost *:80>
```

**重要:** ポート 8080 は Apache Tomcat のデフォルト・ポートであるため、使用しないでください。

5. 以下のような http-vhosts.conf の他の設定を、ご使用の Apache HTTP Server の要件に合わせて変更します。
  - ServerAdmin: Build Forge 管理者の E メール・アドレス
  - DocumentRoot: Build Forge アプリケーションの入力ページがある場所
  - ServerName: Build Forge アプリケーションがインストールされるサーバー
  - ServerAlias: Build Forge ServerName URL のオプションの別名
  - ErrorLog: Build Forge アプリケーションに関する Apache エラー・ログ
  - CustomLog: Build Forge アプリケーションへのアクセスを記録するための Apache エラー・ログ

## Apache HTTP Server 用の PHP のインストールおよび構成

PHP は Apache HTTP Server と一緒にインストールされません。PHP 5.2.4 をインストールして、Apache HTTP Server の httpd-vhosts.conf を指すように構成する必要があります。

## ご使用の Build Forge データベース用の PHP のインストールおよび構成

PHP のインストール時に、Build Forge データベースとして使用しているデータベース・タイプ用の PHP 拡張を選択してインストールします。

## (オプション) パスワード暗号化をサポートするための PHP OpenSSL モジュールの構成

SSL をサポートするために、Build Forge は PHP OpenSSL モジュールを使用します。このサポートは PHP 5.2.4 で提供されるため、追加の構成は必要ありません。

パスワード暗号化をサポートするには、追加の構成がいくつか必要になります。この構成をサポートするには PHP 5.2.4 が必要になります。以下のようにして、OpenSSL 拡張のパッチ・ファイルを見つけて、それらを OpenSSL ディレクトリーにインストールし、PHP を再コンパイルする必要があります。

1. Build Forge のインストール・ディレクトリーにある misc ディレクトリーで php\_openssl.h および openssl.c パッチ・ファイルを見つけます。例えば、次のとおりです。

Windows	C:\Program Files\Build Forge\misc
UNIX/Linux	/opt/buildforge/Platform/misc

2. Build Forge のインストール・ディレクトリーにある openssl ディレクトリーにパッチ・ファイルをコピーします。

3. `--with-openssl=<path_to_openssl>` 構成オプションを使用して PHP をコンパイルします。 `<path_to_openssl>` は Build Forge openssl ディレクトリーです。

## データベース用の Apache の構成

ご使用のデータベースに応じて、`httpd.conf` に特定の情報を追加する必要があります。

### DB2 用の Apache の構成

1. 次の行を Apache 開始スクリプト (通常は `/etc/init.d/httpd` または `/etc/init.d/apache2`、配布バージョンにより異なります) の先頭に追加します。

```
source /home/db2bf/sql1lib/db2profile
```

2. 次の行を `httpd.conf` に追加します。

```
PassEnv LD_LIBRARY_PATH
PassEnv CLASSPATH
PassEnv LIBPATH
PassEnv VWSPPATH
```

### MySQL 用の Apache の構成

追加構成は必要ありません。

### Oracle 用の Apache の構成

1. 次の行を `httpd.conf` に追加します。

```
PassEnv LD_LIBRARY_PATH
PassEnv NLS_LANG
PassEnv ORACLE_HOME
PassEnv ORA_NLS
PassEnv ORA_NLS32
PassEnv TNS_ADMIN
```

2. 次の行を、ブート時に Apache を開始するスクリプト (通常は `/etc/init.d/httpd` または `/etc/init.d/apache2`) に追加して、以下の設定に値を指定します。

```
export LD_LIBRARY_PATH=<値>
export NLS_LANG=<値>
export ORACLE_HOME=<値>
export ORA_NLS=<値>
export ORA_NLS32=<値>
export TNS_ADMIN=<値>
```

## Apache HTTP Server の始動

Build Forge エンジンを開始して、管理コンソールを開始する前に、Apache HTTP Server を始動します。

## PHP のインストールと構成

管理コンソールで使用するために PHP をセットアップするには、次の手順に従ってください。

### 始める前に

要件:

- バージョン: バージョン 5.2.4 以降の PHP が必要です

- データベース・ドライバー: インストールされた Build Forge データベース用の PHP モジュール

## このタスクについて

このセクションの説明に従って、Apache HTTP Server またはその他の Web サーバー用に PHP を構成します。以下の説明では、必要なバージョンの PHP が既にダウンロードされていることが前提となります。

- PHP のダウンロード
- PHP のインストール
- PHP の構成
- Apache 構成ファイルの編集
- (オプション) データベースへのアクセスに使用するプロキシ・サーバーの識別 (管理コンソール・ホストがプロキシ・サーバーを介してデータベースにアクセスする場合にのみ必要です)

## PHP のインストール

### このタスクについて

このセクションでは、ソースから PHP のコンパイルおよびインストールする方法を説明します。すでに PHP がインストールされており再コンパイルをしない場合は、適切なデータベース・ドライバーがインストールされているかどうかを判別するだけでかまいません。データベース・ドライバーをインストールする必要がある場合は、使用するインストール方法に関して PHP の資料を参照してください。以下のように Build Forge で使用するデータベースに対応したデータベース・ドライバーをインストールします。

- DB2 の場合: `ibm_db2` ドライバー
- MySQL の場合: `mysqli` ドライバー
- Oracle Instant Client の場合: `oci8` ドライバー

注: 現時点では、完全な Oracle クライアントは PHP `oci8` と互換性がありません。必ず Oracle Instant Client を使用してください。

### 手順

1. 上で作成した作業ディレクトリーにインストールする PHP を構成します。

```
$ ./configure --prefix=/usr/local/php-5.2.4 --with-<database>=shared ¥
--with-apxs2 --with-ldap=shared --enable-mbstring --enable-shmop ¥
--with-xml --with-zlib=shared
```

`--with-<データベース名>` は以下のように置き換えてください。

- DB2 の場合: `--with-ibm_db2[=dir]`。=`dir` が指定されない場合は、デフォルト値は `/home/db2inst1/sqlllib` が使用されます。
- MySQL の場合: `--with-mysqli[=file]`。オプション・ファイルのパラメーターは `mysql_config` へのパス名です。
- Microsoft SQL Server の場合: `--with-mssql[=dir]`
- Oracle の場合: **PHP `oci8` を使用するための Oracle Instant Client の別個のインストールが必要になります。** Oracle Instant Client を使用してデータベース

に接続する場合は、`--with-oci8=instantclient,lib` を使用します (*lib* は Instant Client の lib ディレクトリーへのパスです)。

コード・ブロック内の行継続文字は `¥` であることに注意してください。このステップでは、PHP のインストール先、およびそれと一緒にインストールされるオプションを指定します。デフォルトでは、`/usr/local` にインストールされます。この例では、`/usr/local/php-5.2.4` にインストールする方法を説明します。この場所は後述の例で使用します。

2. PHP をコンパイルします。

```
$ make
```

このステップでは、ローカル側のディレクトリーに実行可能ファイルをコンパイルします。

3. PHP を (root として) インストールします。

```
# make install
```

このステップは、Apache をインストールするディレクトリー (この例では、`/usr/local/apache-2.2.4`) に対する書き込み権限を持つユーザーとして実行する必要があります。通常は、root として実行します。ローカル側の管理セットアップによっては、これと異なる場合があります。

## PHP の構成

### 手順

1. 拡張ファイルを拡張ディレクトリーにコピーします。データベースの拡張ファイルは、リポジトリからアクティブ拡張ディレクトリーまでコピーする必要があります。以下は、PHP を `/usr/local/php-5.2.4` にインストールした場合の例です。ここで `<日付スタンプ>` は数字のストリングになることに注意してください。

```
$ cd /usr/local/php-5.2.4/lib/php/extensions/no-debug-non-zts-<datestamp>/
$ cp <db-extensions> ..
```

`<db-extensions>` ファイルは、以下のように Build Forge で使用するデータベースに対応します。

- DB2 の場合: `ibm_db2.so`
- MySQL の場合: `mysql.so` および `mysql_i.so`
- Oracle の場合: `oci8.so`

2. PHP 構成ファイル `php.ini` を編集します。以下は、PHP を `/usr/local/php-5.2.4` にインストールした場合の例です。

```
$ cd /usr/local/php-5.2.4/lib/
$ vi php.ini
```

次の項目を追加します。

```
extension_dir=/usr/local/php-5.2.4/lib/php/extensions
upload_tmp_dir=<directory>
extension=<db-extension-so>
```

以下のように使用するデータベースに対応した `<db-extensions-so>` ファイル名 (複数可) を使用します。

- DB2 の場合: extension=ibm\_db2.so
- MySQL の場合: 以下の 2 つを使用します。  
extension=mysql.so  
extension=mysqli.so
- Oracle の場合: extension=oci8.so

注: upload\_tmp\_dir に使用するディレクトリーは、Apache Web サーバーを実行するユーザーによって書き込み可能でなければなりません。通常、このユーザーは nobody ですが、ローカル側の管理慣行によってはこれと異なる場合があります。

## Apache 構成ファイルの編集

### 手順

Apache 構成ファイルを編集します。PHP に関する情報を httpd.conf に追加します。

```
cd <apache-dir>
vi httpd.conf
```

次の行を追加します。

```
LoadModule php5_module modules/libphp5.so
AddHandler php5-script .php
AddType text/html .php
DirectoryIndex index.php
```

## プロキシ・サーバーの指定

### このタスクについて

オプション: このステップは、管理コンソールでプロキシ・サーバーを使用してデータベースにアクセスする場合にのみ必要です。

### 手順

PHP 構成ファイル php.ini を編集します。ファイルは <php-install>/lib にあります。例えば、/usr/local/php-5.2.4 などです。

次の項目を追加します。

```
bf_proxyHost=<使用するプロキシ・サーバーのホスト名>
bf_proxyPath=<プロキシのパス>
bf_symlinkPath=<symlink からプロキシへのパス>
```

## Apache Tomcat のインストールと構成

Installation Manager は、Build Forge 用のアプリケーション・サーバーとして Apache Tomcat をインストールおよび構成します。提供された Apache Tomcat アプリケーション・サーバーを使用することが、Build Forge 用のアプリケーション・サーバーを構成する最も速い方法です。

この標準構成に代わる方法として、Build Forge によって提供される Apache Tomcat の代わりに、既存の Apache Tomcat を使用することができます。このセクションでは、この代替方法についての前提ソフトウェア、インストール前のセットアップ、

インストール、およびインストール後の要件について説明します。この説明では、ユーザーが Apache Tomcat のセットアップおよび構成経験があることを前提とします。

## ソフトウェア前提条件

- Apache Tomcat Server:
  - 5.5.28 (Solaris プラットフォームの場合)
  - 5.5.9 (その他すべてのプラットフォームの場合)
- J2SE 5: IBM または Sun
- Build Forge データベースの JDBC データベース・ドライバ: Apache Tomcat 用の Java Database Connectivity (JDBC) ドライバが必要になります。Sun が提供している JDBC ベンダー・リスト (<http://developers.sun.com/product/jdbc/drivers>) を参照してください。

## JDBC ドライバーの JAR ファイルのインストール

ご使用のデータベースに適した JDBC ドライバーをダウンロードして unzip します。

**重要:** JDBC ドライバーのダウンロード・データには、多数のファイルおよびサブディレクトリが含まれている場合があります。JDBC ドライバーの JAR ファイルを見つけて、その JAR ファイルを `$CATALINA_HOME/common/lib`

にのみコピーします。`$CATALINA_HOME` は Tomcat のインストール・ルートです。これは環境変数として設定する必要があります。ご使用の JDBC ドライバーのインストール資料を参照してください。

- DB2 - <http://www-306.ibm.com/software/data/db2/express/download.html>

にアクセスし、「**DB2 Driver for JDBC and SQLJ**」の横にあるダウンロード・リンクをクリックします (IBM のアカウント登録が必要です)。ライセンス .jar ファイル `db2jcc_license_cu.jar` を検索してインストールする必要もあります。

- MySQL - <http://www.mysql.com/products/connector/j/>

にアクセスし、「**MySQL Connector/J 5.0 or 5.1**」のリンクをクリックします。ご使用の MySQL バージョンに対応する JDBC ドライバー・バージョンを選択します。

- Oracle - [http://www.oracle.com/technology/software/tech/java/sqlj\\_jdbc/index.html](http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html)

にアクセスし、ご使用の Oracle バージョンの横にあるダウンロード・リンクをクリックします (アカウント登録が必要です)。

- Microsoft SQL Server - <http://msdn.microsoft.com/en-us/data/aa937724.aspx>

にアクセスし、『**Download SQL Server JDBC Driver**』リンクをクリックします。

## Installation Manager での Apache Tomcat サーバーの構成

ここでは、Installation Manager を介して Apache Tomcat を構成するために必要な情報を確認します。

1. Apache Tomcat をシャットダウンします。

**重要:** Installation Manager を開始する前に、Apache Tomcat を停止しておく必要があります。

2. Installation Manager を開始します。
3. 「開始」ページで、「インストール」をクリックします。
4. Installation Manager ウィザードの指示に従って、製品をインストールします。
5. 「アプリケーション・サーバー構成」ページで、「はい」をクリックして、ユーザー自身のアプリケーション・サーバーを構成します。

### チェックリスト: アプリケーション・サーバー構成

✓	フィールド	説明
	リダイレクト URL	ご使用のアプリケーション・サーバーのホスト名とポート番号を入力します。コンテキスト・パスとして <code>jas</code> を指定する必要があります。例えば、次のようになります。 <code>http:   https://&lt;app_server_host&gt;:&lt;app_server_port&gt;/jas.</code>
	Build Forge サービス・プラグインをインストールするディレクトリーの指定 (Specify the directory where you want to install the Build Forge services plug-ins)	アプリケーション・サーバー・ホストに対してローカルなディレクトリーを指定します。Installation Manager は、このディレクトリーに Build Forge サービス層アプリケーション・プラグイン拡張をインストールします。アプリケーション・サーバーを実行するユーザーは、このディレクトリーに対する読み取り、書き込み、および実行権限を持っている必要があります。

**注:** 以前のバージョンの Build Forge では、WAR デプロイメント・ディレクトリーを指定する必要がありました。WAR デプロイメント・ディレクトリーは、自動的に `bfinstall/PrepForExternal` に設定されるようになりました。

6. Installation Manager を使用したインストールを完了します。

## Apache Tomcat のインストール後の構成

Installation Manager を使用したインストールが完了したら、以下のインストール後のステップを実行します。

1. `buildforge.conf` ファイルを `jas.war` ファイルに手動で追加します。

`jas.war` のロケーションは、ご使用のオペレーティング・システムによって異なります。UNIX および Linux では、このファイルは `/opt/buildforge/PrepForExternal` にあります。Windows では、このファイルは `C:\Program Files\IBM\Build Forge\PrepForExternal` にあります。99 ページの『`buildforge.conf` ファイルの更新』を参照して、必要なステップを実行します。



2. Apache Tomcat サーバーの JVM ヒープ・サイズを増やします。

JVM 最大ヒープ・サイズ・オプション `-Xmx` を 1024 M に設定します。

`catalina.bat` または `catalina.sh` の `CATALINA_OPTS` 環境変数または `JAVA_OPTS` 環境変数を使用して、この JVM オプションを設定します。

3. 次のように設定して、Build Forge の開始前に、Apache Tomcat を始動します。

```
$CATALINA_HOME/bin/catalina.sh start
```

---

## Installation Manager の手動インストール

ランチパッド・プログラムを使用して製品のインストールを開始すると、IBM Installation Manager? が自動インストールされるか更新されます。64 ページの『ランチパッドを使用した Installation Manager の開始』を参照してください。

Installation Manager に慣れているユーザーや、サイレント・インストールのセットアップを検討しているユーザーは、Installation Manager を手動でインストールすることもできます。次のステップを実行します。

1. パスポート・アドバンテージからダウンロードするか、製品 DVD を使用して、製品のインストール・パッケージを入手します。
2. ご使用のプラットフォーム用の以下のいずれかの Installation Manager ファイルを見つけます。
  - `InstallerImage_linux`
  - `InstallerImage_solaris`
  - `InstallerImage_win32`
3. 以下のいずれかのコマンドを入力して、インストール・プログラムを開始します。
  - 管理ユーザーとしてインストールを実行する場合は、次のコマンドを実行します。

```
install
```
  - 非管理ユーザーとしてインストールを実行する場合は、次のコマンドを実行します。

```
userinst
```
4. インストールの指示に従って、Installation Manager をインストールします。

インストールが完了すると、Installation Manager または Installation Manager インストーラーを使用して、パッケージをサイレント・インストールすることができます。

## Installation Manager の開始

Installation Manager を Windows または UNIX/Linux 上で開始します。

## 始める前に

ランチパッド・プログラムを使用して製品のインストールを開始する場合、Installation Manager は自動的に開始されます。Installation Manager が既にインストールされている場合、以下のいずれかの方法で開始できます。

- Windows の場合: 「スタート」 > 「すべてのプログラム」 > 「IBM Installation Manager」 > 「IBM Installation Manager」をクリックします。
- <IM-installdir> に移動して、./IBMIM を実行します。

## リポジトリ URL の指定

IBM Installation Manager は、各製品パッケージの埋め込み URL を使用して、インターネット経由でリポジトリ・サーバーに接続したり、最新の製品インストール・パッケージを検索したりします。

### 始める前に

Installation Manager の「設定 (Preferences)」ウィンドウにある「リポジトリ (Repositories)」ページで、リポジトリ・ロケーションを設定できます。場合によっては、イントラネット・サイトを使用するために、リポジトリをリダイレクトするよう組織から要請されることがあります。

**注:** インストール・プロセスを開始する前に、必ずインストール・パッケージ・リポジトリ URL を管理者または IBM から入手してください。

リポジトリを指定するには、以下のステップを実行します。

1. IBM Installation Manager を開始します。
2. 「開始」ページで、「ファイル」>「設定 (Preferences)」をクリックします。
3. 「設定 (Preferences)」ウィンドウで、「リポジトリ (Repositories)」をクリックします。「リポジトリ (Repositories)」ページが開き、使用可能なりポジトリ、そのロケーション、およびリポジトリが接続されているかどうかが表示されます。
4. 「リポジトリ (Repositories)」ページで、「リポジトリの追加 (Add Repository)」をクリックします。
5. 「リポジトリの追加 (Add repository)」ダイアログ・ボックスで、リポジトリ・ロケーションの URL を入力するか、「参照」を使用して、リポジトリ、diskTag.inf ファイル、または拡張リポジトリの repository.config ファイルを含む .zip または JAR ファイルを見つけ、「OK」をクリックします。

新規リポジトリ・ロケーションがリストされます。リポジトリが接続されていない場合、赤い x 印が「接続 (Connection)」列に表示されます。

**注:** 更新済みパッケージを検索する場合、「インストールおよび更新時にサービス・リポジトリを検索 (Search service repositories during installation and updates)」が選択されていることを確認します。このオプションはデフォルトで選択されています。

6. 「OK」をクリックして「設定 (Preferences)」ウィンドウを閉じます。

## 製品コンポーネントのサイレント・インストールの実行

Installation Manager をサイレント・インストール・モードで実行することにより、Rational Build Forge 製品コンポーネントをサイレントでインストールできます。サイレント・モードでは、ユーザー・インターフェースは使用できません。その代わりに、応答ファイルにより、製品パッケージのインストールに必要なコマンドを入力します。

サイレント・インストールを実行するには以下の作業が必要になります。

1. Installation Manager のインストール。
2. 応答ファイルの作成。
3. サイレント・インストール・モードでの Installation Manager の実行。

注: 次の場合は、サイレント・インストールを使用できません。

- X11 がインストールされていない Linux サーバーにインストールする場合

Installation Manager およびサイレント・インストールについて詳しくは、Installation Manager Information Center (<http://www.ibm.com/software/awdtools/installmanager/support/index.html>) を参照してください。

## Installation Manager での応答ファイルの作成

Installation Manager を使用して製品パッケージをインストールするときにアクションを記録することにより、応答ファイルを作成することができます。応答ファイルを記録する際、Installation Manager UI で行った選択はすべて XML ファイルに保管されます。サイレント・モードで Installation Manager を実行する場合、Installation Manager は XML 応答ファイルを使用して、インストールを実行します。

応答ファイルを作成して製品をインストールすることも、`-skipInstall` `<agentDataLocation>` 引数を使用して、製品のインストールを行わずに、応答ファイルの作成のみを行うこともできます。以下の説明では、両方のオプションの構文例を示します。

インストールの応答ファイルを作成するには、以下のようになります。

1. コマンド行で、Installation Manager をインストールしたディレクトリーの `eclipse` サブディレクトリーに移動します。例えば、次のようになります。

Windows	C:\Program Files\IBM\Installation Manager\eclipse
UNIX/Linux	/opt/IBM/InstallationManager/eclipse

2. コマンド行で、以下のいずれかのコマンドを使用して、Installation Manager を開始します。応答ファイルおよびログ・ファイル (オプション) のファイル名と場所はご使用のものを指定してください。

入力したファイル・パスが存在していることを確認します。Installation Manager は、応答ファイルとログ・ファイルのディレクトリーを作成しません。`-skipInstall` オプションを使用する場合、`<agentDataLocation>` は書き込み可能ディレクトリーでなければなりません。

- 応答ファイルを記録して、製品をインストールする場合:

IBMIM -record <応答ファイルのパスおよび名前> -log <ログ・ファイルのパスおよび名前>

- 製品をインストールせずに応答ファイルを記録する場合:

IBMIM -record <応答ファイルのパスおよび名前> -log <ログ・ファイルのパスおよび名前> -skipInstall <agentDataLocation>

3. 「パッケージのインストール (Install Packages)」ウィザードの説明に従って、インストールの選択を行います。
4. 「完了」をクリックしてから、Installation Manager を閉じます。

XML 応答ファイルが作成されて、コマンドで指定した場所に置かれます。

## サイレント・モードでの Installation Manager のインストールおよび実行

Installation Manager を使用して、コマンド行から製品パッケージをサイレント・インストールします。

サイレント・インストールを実行するには以下の作業が必要になります。

サイレント・モードで Installation Manager を実行するには、以下に示すように、ご使用のプラットフォーム用のコマンドを eclipse サブディレクトリーから実行します。

Windows	IBMIMc.exe --launcher.ini silent-install.ini -input <応答ファイルのパスおよび名前> -log <ログ・ファイルのパスおよび名前>  例えば、次のようになります。IBMIMc.exe --launcher.ini silent-install.ini -input C:%mylog%responsefile.xml -log C:%mylog%silent_install_log.xml
UNIX/Linux	IBMIM --launcher.ini silent-install.ini -input <応答ファイルのパスおよび名前> -log <ログ・ファイルのパスおよび名前>  例えば、次のようになります。IBMIM --launcher.ini silent-install.ini -input /root/mylog/responsefile.xml -log /root/mylog/silent_install_log.xml

Installation Manager は、サイレント・インストール・モードで実行されると、応答ファイルを読み取って、指定したディレクトリーにログ・ファイルを書き込みます。応答ファイルは必須ですが、ログ・ファイルはオプションです。このコマンドの実行に成功した場合はゼロ、失敗した場合はゼロ以外の数が返されます。

下の表で、サイレント・インストール・コマンドで使用する引数について説明します。

引数	説明
-vm	Java ランチャーを指定します。サイレント・モードの場合は必ず、Windows では java.exe を、他のプラットフォームでは java を使用してください。
-nosplash	スプラッシュ画面を抑制します。
--launcher.suppressErrors	JVM エラー・ダイアログ・ボックスを抑制します。

引数	説明
-silent	Installation Manager インストーラーをサイレント・モードで実行します。
-input	Installation Manager への入力として使用する応答ファイルを指定します。応答ファイルには、インストーラーまたは Installation Manager が実行するコマンドが含まれています。
-log	(オプション) サイレント・インストールの結果を記録するログ・ファイルを作成します。ログ・ファイルは XML ファイルです。

## 製品コンポーネントのサイレント・アップグレードの実行

Installation Manager をサイレント・インストール・モードで実行することにより、Rational Build Forge 製品コンポーネントをサイレントでアップグレードできます。

サイレント・アップグレードを実行するには、以下の前提条件が必要です。

- 既存の Build Forge コンソールのインストール済み環境は、Installation Manager ベースのサイレント・インストールを使用してインストールされたものである必要があります。
- Installation Manager は、Build Forge コンソール・ホストと同一のホストにインストールする必要があります。

サイレント・インストールを実行するには以下の作業が必要になります。

1. アップグレード用の応答ファイルを作成します。
2. 入力として応答ファイルを指定し、Installation Manager をサイレント・モードで実行します。

Installation Manager およびサイレント・インストールについて詳しくは、Installation Manager Information Center (<http://www.ibm.com/software/awdtools/installmanager/support/index.html>) を参照してください。

## Installation Manager での更新用応答ファイルの作成

Installation Manager を使用して製品パッケージをインストールする際のアクションを記録することにより、応答ファイルを作成します。

応答ファイルを記録する際、Installation Manager UI で行った選択はすべて XML ファイルに保管されます。

更新インストール用の応答ファイルを作成するには、以下のようになります。

1. Installation Manager を実行します。「設定 (Preferences)」で、更新インストール用の製品リポジトリの URL を IM リポジトリのリストに追加して、それが選択されていることを確認します。
2. Installation Manager を終了します。
3. コマンド行で、Installation Manager をインストールしたディレクトリーの eclipse サブディレクトリーに移動します。例えば、次のようになります。

Windows	C:\Program Files\IBM\Installation Manager\eclipse
---------	---

UNIX/Linux	/opt/IBM/InstallationManager/eclipse
------------	--------------------------------------

4. 実際に製品をインストールせずに、インストール内容の記録を開始します。

`response_file` および `log_file` の絶対パス (ファイル名も含む) を入力します。入力したファイル・パスが存在することを確認してください。Installation Manager は、応答ファイルとログ・ファイルのディレクトリーを作成しません。  
`agentDataLocation` は書き込み可能なディレクトリーでなければなりません。

`IBMIM -record response_file -log log_file -skipInstall agentDataLocation`

5. Installation Manager が始動します。Installation Manager で、「更新」をクリックして、プロンプトに対応します。
6. Installation Manager が完了したら、「完了」をクリックします。
7. Installation Manager を終了します。

XML 応答ファイルが作成されて、コマンドで指定した場所に置かれます。

## サイレント・モードでの更新インストールの実行

Installation Manager を使用して、コマンド行から製品パッケージをサイレント・インストールします。

Installation Manager をサイレント・モードで実行するには、以下のような一般的な形式のコマンドを使用します。

`IBMIMc.exe --launcher.ini silent-install.ini -input response_file -log log_file`

`response_file` および `log_file` には、絶対パスおよびファイル名を使用してください。

- Windows の例

```
IBMIMc.exe --launcher.ini silent-install.ini -input
C:\mylog\responsefile.xml
-log C:\mylog\silent_install_log.xml
```

- UNIX または Linux の例

```
IBMIM --launcher.ini silent-install.ini -input /root/mylog/responsefile.xml
-log /root/mylog/silent_install_log.xml
```

Installation Manager は、サイレント・インストール・モードで実行されると、応答ファイルを読み取って、指定したディレクトリーにログ・ファイルを書き込みます。応答ファイルは必須ですが、ログ・ファイルはオプションです。このコマンドの実行に成功した場合はゼロ、失敗した場合はゼロ以外の数が返されます。

下の表で、サイレント・インストール・コマンドで使用する引数について説明します。

引数	説明
-vm	Java ランチャーを指定します。サイレント・モードの場合は必ず、Windows では <code>java.exe</code> を、他のプラットフォームでは <code>java</code> を使用してください。
-nosplash	スプラッシュ画面を抑制します。
--launcher.suppressErrors	JVM エラー・ダイアログ・ボックスを抑制します。

引数	説明
-silent	Installation Manager インストーラーをサイレント・モードで実行します。
-input	Installation Manager への入力として使用する応答ファイルを指定します。応答ファイルには、インストーラーまたは Installation Manager が実行するコマンドが含まれています。
-log	(オプション) サイレント・インストールの結果を記録するログ・ファイルを作成します。ログ・ファイルは XML ファイルです。

## VMware への Build Forge システムのインストール

Build Forge を VMWare にインストールして実行することができます。

以下のガイドラインに従ってください。

- Build Forge が別個のホスト (VMWare イメージではなく物理ホストが望ましい) で使用するデータベースをインストールします。
- VMware Workstation が仮想マシンの実行に使用するメモリーを少なくとも 1GB に設定します。この値を調整するには、VMWare Workstation で「編集」 > 「設定 (Preferences)」 > 「メモリー (Memory)」をクリックします。
- VMWare でのパフォーマンスを最大化するために、他のシステム・リソース・パラメーターの操作が必要になる場合があります。

## System z 上の Linux への管理コンソールのインストール

インストール・メディアに用意されている mc-<version>-<build>.tar.gz tar ファイルを使用して、z/Linux に管理コンソールをインストールして構成します。このインストールに IBM Installation Manager は使用しません。

z/Linux のコンソールは、Apache Web サーバーではなく IBM HTTP サーバーに同梱されています。

管理コンソールのインストール後、エージェントの rpm パッケージ (zlinux-bfagent-<version>.rpm) を z/Linux にインストールし、Build Forge® 用の z/Linux サーバーをセットアップします。インストールの説明については、159 ページの『UNIX および Linux システムへのエージェントのインストール』を参照してください。

### インストール中に必要な情報

インストール中に、以下の情報を入力するように要求されます。

1. インストール・ディレクトリー
  - Build Forge® をインストールするロケーションの絶対パスを選択します。
2. データベース情報
  - Build Forge® が使用するデータベースのタイプ (DB2、Oracle、または MySQL)
  - データベース・サーバーのホスト名



- データベースのポート番号
  - 使用するデータベース名
  - Build Forge<sup>®</sup> でデータベースへの接続に使用するデータベース・ユーザー名
  - データベース・ユーザー名のパスワード
  - データベースへのアクセスに使用されるクライアント・ライブラリーの場所
  - JDBC ドライバー JAR ファイルの場所
3. アプリケーション・サーバー情報
- 使用するアプリケーション・サーバー (提供される Tomcat、またはセットアップ済みの WebSphere<sup>®</sup> Application Server インストール済み環境)
- WebSphere Application Server を選択する場合は、さらに以下の情報を提供します。
- Build Forge<sup>®</sup> サービス・コンポーネントの場所 (WebSphere Application Server にインストールされるように)。ドメイン、ポート、および jas へのパスが必要です。例:  
`http://mydomain.com:9080/jas`
  - プラグイン用に使用するディレクトリー (読み取り/書き込みが可能なディレクトリー)。現在のところ、Rational Team Concert との統合を行うときにのみ使用されます。
  - WAR デプロイメント・ディレクトリー: Build Forge<sup>®</sup> jas.war ファイルの一時ロケーション。インストール後に、これをアプリケーション・サーバーにデプロイします。
  - Java 実行可能ファイル (java.jar) へのパス
  - サービス層の一時ストレージ・ディレクトリー。このディレクトリーは読み取り/書き込みが可能である必要があります。これは、一時情報を保管するために、サービス・コンポーネントによって使用されます。
  - Build Forge<sup>®</sup> サービス用の HTTP ポート (デフォルト: 3966)
  - Build Forge<sup>®</sup> サービス用の SSL ポート (デフォルト: 49150)
4. Web サーバー情報
- 使用する Web サーバー (提供される IBM HTTP サーバー、またはセットアップ済みの Web サーバー)

提供される IHS サーバーを選択する場合は、SSL に関する情報をさらに提供します (SSL を使用するかどうか、どのポートとメモリーを使用するか、既存のセキュア証明書を使うか新規に作成するか、など)。

## インストーラーの実行

1. コンソールのパッケージを tar ファイルから解凍した際の、解凍先のディレクトリーに移動します。
2. 以下のコマンドを実行します。  
`./cmdline-install.sh`

## 例

以下の注釈付きリストには、インストーラーにおける処理手順が示されています。このリストは注釈付きの実行であり、実際のインストールを反映したものではありません。デフォルト設定が使用可能な場合には、大括弧付きで示されます。例: [y]。デフォルトを受け入れるには、**Enter** キーを押します。

インストール・ディレクトリー [/opt/buildforge]

どのデータベースを使用しますか?  
使用するデータベースの番号を入力してください  
1) DB2  
2) Oracle  
3) MySQL

注: リストの残りの部分では、Oracle が選択されたものと想定されています。

データベースのホスト名は? 127.0.0.1  
データベースのポート番号は? [1521]  
データベース名は? build  
データベースのユーザー名は? build  
データベースのユーザー・パスワードは?  
データベースのユーザー・パスワードを確認しますか?

このインストールで、Build Forge データベース・スキーマを作成しますか? (y|n) [y]

この時点でのクライアントのライブラリーと情報を指定します。選択したデータベースに応じて、以下のようなプロンプトが出されます。絶対パスを使用してください。

- DB2

32 ビット DB2 クライアント・ライブラリー (libdb2.so.1) の場所は?

注: このライブラリーは、32 ビットのライブラリーである必要があります。絶対パスを入力してください。

DB2 (DB2\_HOME) のインストール場所は?

- MySQL

MySQL クライアント・ライブラリー (libmysqlclient.so) の場所は?

注: このライブラリーは、32 ビットのライブラリーである必要があります。絶対パスを入力してください。

- Oracle

Oracle クライアント・ライブラリー (libclntsh.so) の場所は?

注: このライブラリーは、64 ビットのライブラリーである必要があります。絶対パスを入力してください。

Oracle Instant Client (ORACLE\_HOME) のインストール場所は?

注: これは、Instant Client ライブラリーのルートへの絶対パスです。

tns.names ファイル (TNS\_ADMIN) の場所は?

注: これは、tns.names ファイルを含むディレクトリーです。絶対パスを入力してください。

データベース JDBC JAR ファイルを含むディレクトリーを入力してください。

注: ojdbc14.jar への絶対パスを入力してください。

指定された Tomcat アプリケーション・サーバーを使用しますか? (y|n) n

注: リストの残りの部分では、アプリケーション・サーバーとして WebSphere Application Server を使用するものと想定しています。

アプリケーション・サーバー上のサービス層への接続に使用する完全 URL を入力してください:

http://mydomain.com:9080/jas

Build Forge サービス・プラグインのインストール先ディレクトリーを入力してください。

注: これは、アプリケーション・サーバー・ホスト上の読み取り/書き込みが可能な必要がある場合があります。サービス・コンポーネントは、Rational Team Concert と Build Forge<sup>®</sup> が統合される際に、これを使用します。

WAR デプロイメント・ディレクトリーの指定:

注: ローカル・ホスト上のディレクトリーを指定してください。インストーラーは、インストールの完了時に、jas.war ファイルをここに配置します。次に、これをアプリケーション・サーバーにデプロイすることができます。

JAR 実行可能ファイルへのパスの入力 (JDK に含まれていてはなりません):

注: データベース・ドライバーの .jar ファイルへのパスを入力してください。

サービス層の一時ストレージ・パスの入力:

注: サービス・コンポーネントが使用できるディレクトリーへのパスを入力してください。これは、読み取り/書き込みが可能である必要があります。

Build Forge サービス層が使用する HTTP ポートは? [3966]

Build Forge サービス層が使用する SSL ポートは? [49150]

指定された Apache Web サーバーを使用しますか? (y|n)[y]

注: リストの残りの部分では、y を入力したものと想定されています。

Apache で SSL を使用しますか? (y|n)[n]

Apache で SSL を構成するには、「y」を入力してください。

注: リストの残りの部分では、y を入力したものと想定されています。

Apache で使用する SSL ポートは? [443]

PHP のメモリーの上限を入力してください (MB 単位): [256]

カスタムの SSL 証明書を変更または指定しますか? (y|n) [n]

注: リストの残りの部分では、y を入力して、カスタムの証明書を作成することにしたものと想定されています。

既存のセキュア証明書はありますか? (y|n)[n]

この証明書には有効期間が必要です。[数値][ピリオド] の形式で入力してください。

例: 10Y = 10 年、6M = 6 カ月、350D = 350 日。この証明書の有効期間を入力してください。

証明書の共通名を入力してください

(通常は、サーバーの名前) [linux142.rtp.raleigh.ibm.com]。

所在地の市区町村を入力してください。

所在地の都道府県を入力してください。

組織名を入力してください。

組織単位を入力してください。

所在地がある国を以下のリストの中から選んでください。

フランス

台湾  
イタリア  
ドイツ  
韓国  
米国  
中国  
ブラジル  
スペイン  
日本

所在地の住所を入力してください。

鍵ストアのパスワードが必要です。パスワードは 6 文字以上の長さでなければなりません。

鍵ストアのパスワードを入力してください。

Web サーバーとして Apache ではなく IBM HTTP サーバーを使用する場合は、セットアップの方法と SSL を有効化する方法について、635 ページの『Apache HTTP Server の代わりに IBM HTTP Server を使用』を参照してください。

## コンソールの開始

1. 管理コンソールを開始します。

```
<bfinstall>/rc/buildforge start
```

2. サービス・コンポーネント (Apache Tomcat サーバー) が開始されたことを確認します。 `catalina.out` を開いて、開始メッセージが記録されているかを確認してください。

```
<bfinstall>/server/tomcat/logs/catalina.out
```

3. Web ブラウザーを開いて、完全修飾 `z/Linux` ホスト名を入力します。例:  
`http://myhost.mycompany.com`。

管理コンソールが開始され、ログイン・プロンプトが表示されます。

4. **root/root** としてログインします。

## ライセンス・ファイルのインストール

`z/Linux` のライセンス・ファイルは、`<bfinstall>` ディレクトリーにあります。ライセンス・ファイルの名前は、`IRBF_license` です。インストール後、ここに含まれるテキスト・メッセージには、パスポート・アドバンテージから実際のライセンス・ファイルをダウンロードする手順が記載されています。

ライセンス・ファイルをダウンロードして、`<bfinstall>` ディレクトリーに置いてから、このファイルを使用するように Build Forge® を構成します。

1. 管理コンソールを開始します。
2. **root/root** としてログインします。
3. 「管理」>「システム」を選択します。
4. 「ライセンス・サーバー」設定を見つけ、その値をライセンス・ファイルの完全修飾パスに設定します。

例えば、`<bfinstall>/IRBF_license` などです。

## 管理コンソールでの SSL の有効化

SSL を有効にして、以下の Build Forge® コンポーネント間で転送されるデータを暗号化することができます。

- Web ブラウザー・クライアントと Apache HTTP サーバー

- Apache Tomcat サーバーと Apache HTTP サーバー

プロンプトに対して「はい」を選択すると、SSL の有効化に必要な作業の一部がインストール・プログラムによって実行されます。SSL を有効にするには、以下の作業を実行します。

1. 個人証明書および鍵ストアの確認。
2. IBM HTTP サーバーの SSL 用の構成。
3. 管理コンソールのユーザー・インターフェースでの SSL の有効化。
4. SSL のデバッグの有効化。

注: パスワード暗号化やシングル・サインオン (SSO) などのその他のセキュリティー・フィーチャーは、本リリースの Build Forge® on z/Linux ではサポートされていません。

### 個人証明書および鍵ストアの確認

以下の鍵ストアが、インストール・プログラムによって作成されます。

鍵ストア	説明
buildForgeKeyStore.p12	パスワードで保護された keyEntry (公開鍵と秘密鍵のペアを使用した個人証明書) が入ります。
buildForgeTrustStore.p12	パスワードで保護された trustedCertEntry (公開鍵のみを使用した証明書) が入ります。
buildForgeKey.pem	パスワードで保護された秘密鍵が入ります。
buildForgeCert.pem	buildForgeKey.pem の秘密鍵に対応する公開鍵を使用した、パスワードで保護されていない証明書が入ります。
buildForgeCA.pem	最初は buildForgeCert.pem と同じ情報が入りますが、信頼を確立するために他のピア証明書が追加されます。
buildForgeKeyForApache.pem	この鍵ストアは、Apache HTTP サーバーの SSL を有効にするために必要です。buildForgeKey.pem と異なり、これはパスワードで保護されていないため、パスワード・プロンプトなしで Apache HTTP サーバーを始動することができます。

既存の PEM 証明書の変換、および証明書の管理については、134 ページの『証明書の管理』を参照してください。

### IBM HTTP Server の SSL 用の構成

635 ページの『Apache HTTP Server の代わりに IBM HTTP Server を使用』の『IHS 用 SSL の構成』を参照してください。鍵ストアのフォーマットを変更して、項目を httpd.conf に追加する必要があります。

### 管理コンソール UI での SSL の有効化

管理コンソールのユーザー・インターフェース設定を使用して、管理コンソールで SSL を有効にし、Build Forge® データベースを更新します。これらの設定値にアクセスするには、「管理」 > 「セキュリティー」と移動します。続いて、bfclient.conf 構成ファイルで必要なプロパティ値が更新されていることを確認します。

1. Build Forge® を開始します。

2. ユーザー・インターフェースにログインします。
3. 「管理」 > 「セキュリティ」に移動します。
4. 「SSL 有効化」を「はい」に変更します。
5. 「保存」をクリックします。
6. 「マスター **BFClient.conf** の更新」をクリックします。

### SSL のデバッグの有効化

管理コンソールで SSL に関する問題をデバッグするには、以下の手順に従って、SSL に必要な追加情報をログに記録します。

1. エンジンでデバッグを有効にします。Build Forge<sup>®</sup> エンジンを開始する前に、以下の環境変数を設定します。

```
export BFDEBUG_SECURITY=1
```

- a. Build Forge<sup>®</sup> エンジンを再始動します。
  - b. IHS を再始動します。IHS を再始動すると、PHP がこのデバッグ・パラメーターを使用できるようになります。
2. Tomcat でデバッグを有効にします。<bfinstall>/server/tomcat/common/classes/logging.properties で以下の変更を行います。
    - a. 次の行を追加します。

```
com.buildforge.level = ALL
```
    - b. ハンドラー・セクションで、その他のすべてのレベルを FINE から ALL に変更します。

Tomcat を再始動して変更を有効にします。

## 第 9 章 管理コンソールでの追加機能の構成

このセクションでは、追加機能を使用可能にしたり、デフォルト構成に代わる構成を提供したりするように Build Forge® を構成する方法について説明します。

### Build Forge 構成ファイル (buildforge.conf)

buildforge.conf ファイルは Build Forge 製品の構成ファイルです。このファイルには、Build Forge データベースを開始して、そのデータベースと通信するために、さまざまな Build Forge コンポーネントで使用する構成設定が含まれています。

buildforge.conf ファイルは以下の 2 つの場所に保管され、インストール後に buildforge.conf ファイルを変更する必要がある場合は、両方の場所で更新しなければなりません。『buildforge.conf ファイルの更新』を参照してください。

- アプリケーション・サーバーで使用される jas.war ファイル (サービス層とも呼ばれる) 内。
- インストール・ルート・ディレクトリー内。下の表に、製品のデフォルトまたは標準のインストール・ディレクトリーを示します。

Windows	C:\Program Files\IBM\Build Forge
UNIX/Linux	/opt/buildforge/Platform

### buildforge.conf ファイルの更新

buildforge.conf ファイルは、2 つの場所にあります。インストール後にこのファイルを更新する場合は、両方の場所でファイルを更新する必要があります。

Build Forge® データベースが別のホスト・コンピューターに移動された場合、データベース・ホストを更新するために buildforge.conf ファイルを編集することが必要になる場合があります。このファイルを編集するもう 1 つの一般的な理由は、データベース・パスワードを更新するためです。データベース・パスワードは、ネットワーク・セキュリティー・ポリシーに準拠するよう定期的に変更する必要があります。

以下の手順を使用して、buildforge.conf ファイルを更新してから、buildforge.conf の更新済みコピーを使用して jas.war ファイルを再作成します。

1. Build Forge® エンジン进行停止します。
2. Build Forge® のインストール・ディレクトリーにある buildforge.conf ファイルを見つけてます。

Windows	<bfinstall>
UNIX および Linux	<bfinstall>/Platform

3. テキスト・エディターを使用してファイルを開き、構成設定を変更して、ファイルを保存します。



注: このファイルを編集するには、ルート特権または管理者特権が必要になります。

4. jas.war ファイルを含むディレクトリーに移動します。例えば、次のとおりです。

Apache Tomcat Server	<code>&lt;bfinstall&gt;/Apache/tomcat/webapps</code> <code>\$CATALINA_HOME/webapps</code>
----------------------	--

5. 任意のアーカイブ・マネージャー・ツールを使用して jas.war ファイルを開き、/WEB-INF/eclipse/plugins/ の下で com.ibm.jas-xxx.jar を見つけます。
6. com.ibm.jas-xxx.jar で buildforge.conf ファイルを更新し、保存します。

注: jas.war ファイルが自動的に更新されるように設定されていることを確認してください。

7. \$CATALINA\_HOME/webapps および \$CATALINA\_HOME/work/Catalina/localhost の下で、次の例を使用して jas ディレクトリーを削除します。

Windows	<code>rmdir jas /s /q</code>
UNIX/Linux	<code>rm -rf jas</code>

8. Build Forge<sup>®</sup> エンジンを実行します。

jas ファイルが再配置され、更新された buildforge.conf ファイルを使用して jas ディレクトリーが自動的に再作成されます。

## Buildforge.conf の参照

buildforge.conf ファイルには、Build Forge 管理コンソールの動作に関する設定が保管されています。

このファイルはインストール・ディレクトリーにあります。これは、インストーラーによって自動的に作成されます。編集する必要がある場合は、ファイルを ASCII テキスト・ファイルまたは XML ファイルとして保存できます。構文は次のとおりです。

- ・ キーワードとその値は 1 行に入力します。
- ・ キーワードと値は、スペースで区切ります (等号は使用しない)。

キーワード	値
bf_file_storage	一時 Build Forge ファイルが存在するディレクトリー。例: <code>C:\Program Files\IBM\BuildForge\temp</code>
bf_plugin_dir	管理コンソールに接続する IDE プラグインが存在するディレクトリー。
birt_home	Eclipse レポート作成ツール (BIRT) のファイルの場所。
db_database	コンソールを使用するために作成したデータベースの名前。
db_hostname	データベースを実行中のコンピューターのホスト名/IP アドレス。 db_hostname の値を入力するときは、実際の名前または IP アドレスを使用してください。デフォルトの localhost を使用しないでください。

キーワード	値
db_password	データベース・ユーザー名用に作成したパスワード。
db_provider	Build Forge のインストール先として選択したデータベース。この値は編集しないでください。
db_schema	ご使用のデータベースのスキーマの名前 (通常は db_username と同じですが、別のスキーマ名が選択されている場合もあります)。
db_tcp_port	使用中のデータベース接続ポート。
db_type	Build Forge がインストールされたデータベースのタイプ。デフォルトは <code>odbc</code> です。この値は編集しないでください。
db_username	データベースのユーザー名。これは、インストーラーを実行する前にセットアップされます。  DB2 および DB2 Express では、Windows (DB2 ではありません) でユーザーを作成します。  その他のすべてのデータベース・タイプでは、データベース・ユーザー名を作成します。  39 ページの『データベースのセットアップ』を参照してください。  この値は、すべてのコンソール・タイプで必須です。
services_hostname	Build Forge サービス層を実行しているコンピューターのホスト名/IP アドレス。これは、ご使用のサービス構成における完全修飾ドメイン・ネーム (FQDN) です。
services_ssl_port	Build Forge サービスに安全に接続するための SSL ポート。
services_tcp_port	SSL が指定されていない場合に Build Forge サービスに接続するための TCP ポート。
services_url	サービス層で使用されるポートを指定する URL。例:  <code>services_url http://mybfhost.com:8080</code>

## 代替ポートを使用するように管理コンソールを構成

デフォルト・ポート (80) 以外のポートで管理コンソールを実行できます。

### このタスクについて

以下の 2 とおりの方法で、管理コンソールを代替ポートで実行するように構成できます。

- インストール時に、ポートを任意の値に設定します。
- 管理コンソールが既にインストール済みである場合には、以下のステップを完了してください。
  1. コンソールを開始し、root ユーザー、または管理特権を持つユーザー名でログインします。
  2. 「管理」 > 「システム」を選択し、「コンソール・ポート」のシステム構成設定値を、管理コンソールが実行されているポートに変更します。
  3. `httpd.conf` (Windows インストール済み環境では `<bfinstall>/Apache/conf/`、\*nix インストール済み環境では `<bfinstall>/server/apache/conf/` に配置)

の 2 つの設定を変更します。例えば、ローカル・コンピューターが `myHost` であり、ポート 81 を使用する場合には、以下のように設定します。

```
Listen 81
ServerName myHost:81
```

4. エンジンを停止してから再始動します。

- Windows の場合: 「スタート」 > 「プログラム」 > 「IBM Rational Build Forge 管理コンソール」 > 「エンジン・サービスの停止」とクリックして、「エンジン・サービスの開始」をクリックします。

Build Forge がフォアグラウンドで実行されている場合は、実行先の Windows コンソールに移動して、Ctrl-C を入力します。

- UNIX または Linux の場合: rc ファイルに提供されているスクリプトを使用します。

```
$ /opt/buildforge/rc/buildforge start
```

```
$ /opt/buildforge/rc/buildforge stop
```

手動でコマンドを使用することもできます。

- a. 停止するには、プロセス ID を見つけてプロセスを強制終了します。

```
$ ps aux | grep buildforge
$ kill ${<PID>}
```

- b. 開始するには、以下のコマンドを使用します (<bfinstall> はインストール・ディレクトリーへのパスです)。

```
<bfinstall>/Platform/buildforge
```

---

## 冗長化構成

複数のコンピューターで Build Forge を実行するようにセットアップし、同一の Build Forge データベースと通信させることができます。このセットアップを冗長化と呼びます。

### 冗長化について

冗長化はジョブ処理のバランスを取るために役立ち、1 つのインストールで障害が起きた場合の可用性を向上させます。

**重要:** 冗長化は、フェイルオーバー機能やその他の高可用性機能を提供するわけではありません。単に、ジョブの処理能力を向上させるだけです。冗長化インストールでは、いずれか 1 つのインストールで障害が起きた場合、そのインストールで管理されていた実行中のジョブはすべて失われますが、その他のインストールでは、実行中のジョブが引き続き処理され、新しいジョブが受け入れられます。

ユーザーがジョブを開始すると、ジョブに関する項目がデータベース内に作成されます。プロセス・エンジンはデータベースをポーリングし、新規のジョブの有無を調べます。複数のプロセス・エンジンがある場合、各エンジンが非ビジー・サイクル中に個別にポーリングを行うため、自動的にロード・バランシングが行われます。

冗長化をセットアップする際には、管理コンソールの通常のインストールを実行し、続いてその他のホストに管理コンソールをインストールします。すべてのインストールは、同じ Build Forge データベースにアクセスするように構成されます。

**重要:** インストールはそれぞれ独自のホストにインストールする必要があります。同一ホスト上に複数の管理コンソールをインストールすることはできません。

## 冗長化システムのインストール

### このタスクについて

以下の説明は、データベースのセットアップと、データベースを使用する管理コンソールの最初のインストールが完了していることを前提としています。別のホストに追加のインストールを作成するには、以下の手順を実行します。

### 手順

1. ご使用のデータベースに応じた手順に従って、ホスト上でインストール前のセットアップを実行します。データベースによっては、ホストにデータベース・クライアントをインストールしたり、その他の構成を行うなどのセットアップが必要になる場合があります。37 ページの『第 6 章 インストール前のセットアップ』を参照してください。
2. データベース・サーバーが外部接続 (TCP) を受け入れるように構成されていることを確認します。
3. インストール手順に従って操作します。63 ページの『第 7 章 管理コンソールのインストール』から開始して、以下を実行します。
  - Installation Manager をインストールします (必要な場合)。
  - Installation Manager を開始します。
  - インストールを実行します。

以下の手順では、特定の入力が必要とするパネルについてのみ説明します。

4. 「パッケージのインストール: フィーチャー」パネルで、すべてのフィーチャーが選択されていることを確認します。
5. 「データベース構成」パネルで、以下を実行します。
  - データベース名およびスキーマ名を指定します。これらは、最初のコンソールで指定した名前と同じでなければなりません。
  - データベースのユーザー名およびパスワードを指定します。最初のコンソールで指定した名前と同じものを使用してください。
  - 「インストール時にこのデータベースにデータを取り込みますか?」で「いいえ」を選択します。
  - 「テストの実行」をクリックします。テストが成功したら、「次へ」をクリックして続行します。

**注:** 「いいえ」を選択した場合、JDBC ドライバーへのパスが正しいかどうかに関してのみテストが行われます。

6. 「アプリケーション・サーバー構成」パネルで、最初のコンソールで入力したものと同一アプリケーション・サーバーの完全修飾ドメイン・ネームを入力します。

注: ポート番号はすべて、最初のコンソールで指定したものと同じでなければなりません。

7. 「コンソール開始オプション」パネルで、「**Build Forge を開始しない**」オプションが選択されていることを確認します。このオプションはぼかし表示されているため、変更できません。インストール後に手動でコンソールを開始します。
8. インストール後の必須構成を行います。213 ページの『第 11 章 インストール後のタスク』を参照してください。必要に応じて、データベース・クライアントから接続できるようにデータベースをカタログします。

## 冗長化での処理

冗長化をセットアップしたら、以下の手順で処理を行います。

- ユーザーに最初のインストールの URL を示します。
- ユーザーが追加のインストールにアクセスできないようにするには、追加のインストールで Apache サーバーを停止します。
- HTTP 要求の処理能力を強化する場合は、すべてのインストールで Apache を実行します。インストール間で要求を分散するには、ロード・バランサーをインストールします。

注: 冗長化でサポートされるジョブ・コントローラーのタイプは同時に 1 つのみです。Perl ジョブとマスター・ジョブ・コントローラー (MJC) ジョブを両方とも同時に実行することはできません。別のジョブ・コントローラーを実行に使用するには、「システム構成」ページでジョブ・コントローラー・タイプを切り替える必要があります。

---

## IPv6 ネットワーク・サポートの有効化

管理コンソールを IPv6 および IPv6/IPv4 混合ネットワークで構成できます。

### このタスクについて

管理コンソールを IPv6 用に構成する際には、以下を行う必要があります。

1. httpd.conf のサーバー項目を変更します。
2. IPv6 ネットワークをサポートするために必要な FLEXlm ライセンス・クライアントをセットアップします。

Build Forge を IPv6 で使用するための要件および構成手順を確認してください。28 ページの『IPv6 サポートのネットワーキング要件』を参照してください。

## httpd.conf の変更

### このタスクについて

IPv6 サポートでは、ご使用のコンピューターおよびネットワークが IPv6 をサポートするように正しく構成されている必要があります。ネットワーク構成に問題があると、Build Forge システム内で指定されたホスト名およびアドレスが正しく解決されません。

IPv6 で稼働するように、**Build Forge** を手動で構成する必要があります。この構成を行うには、Apache の主要構成ファイル httpd.conf 内の項目を変更します。

1. httpd.conf ファイル (buildforge/server/apache/conf/httpd.conf) にナビゲートします。
2. 完全修飾ドメイン・ネーム (FQDN) を参照する ServerName を追加します。例:  
ServerName qlnx500-v6.ipv6.lexma.ibm.com
3. Listen ディレクティブを 0.0.0.0:80 から以下のように変更します。
  - Windows: [::]:80
  - UNIX または Linux:80

---

## セキュリティ・フィーチャー

このセクションでは、Build Forge®で以下のセキュリティ・フィーチャーを有効にする方法について説明します。

- セキュア・ログイン。インストール時にデフォルトで有効にされます。
- シングル・サインオン (SSO)
- HTTPS および SSL の有効化
- パスワード暗号化の有効化
- bfclient.conf ファイルによるセキュリティ構成

これらのフィーチャーは、管理コンソールの「管理」 > 「セキュリティ」での選択項目の組み合わせと、構成ファイルの手動セットアップにより有効になります。このセクションには、セキュリティ・フィーチャーの有効化に使用される構成ファイル bfclient.conf の参照セクションが含まれています。

**注:** このセクションは、Build Forge on z/Linux を実行するユーザー向けではありません。Build Forge on z/Linux で使用可能なセキュリティ・フィーチャーについては、92 ページの『System z 上の Linux への管理コンソールのインストール』を参照してください。

## シングル・サインオンの実装

Build Forge には、シングル・サインオンのフレームワークが備わっています。

シングル・サインオンとは認証方式の 1 つであり、これを使用すると、ユーザーは毎回ユーザー名とパスワードを入力しなくても、アプリケーションにアクセスできるようになります。Build Forge には、シングル・サインオンを実装するためのサード・パーティー HTTP インターセプターと組み合わせて使用できるフレームワークが同梱されています。

### シングル・サインオン・フレームワークについて

Build Forge® SSO フレームワークは、多くの市販の SSO ソリューションとの統合を可能にする機能を提供します。SSO フレームワークはインターセプター・ベースであり、HTTP 要求をインターセプトして、それを処理するためのメソッドを提供します。カスタム・インターセプターを作成し、HTTP 要求に含まれるセキュリティ成果物を受信したり検証したりすることができます。具体的には、インターセプターによって HTTP 応答にトークンを設定し、そのトークンを後続の要求内で探することができます。

Build Forge® では、以下の 2 つの SSO ソリューションを提供します。



- SPNEGO 用インターセプター (保護された単純なネゴシエーション・プロトコル)。 110 ページの『Active Directory ドメインにおける SPNEGO を使用したシングル・サインオンの実装』を参照してください。
- WebSphere SSO との統合用インターセプター。 116 ページの『カスタム・インターセプターを使用した WebSphere Application Server セキュリティーとの統合』を参照してください。

**SSO フレームワーク・メソッド:** SSO インターセプターは、Build Forge® SSO フレームワークが使用するインターフェースをインプリメントした、以下のような Java クラスです。

```
com.buildforge.services.server.sso.ISSOInterceptor
```

これは、以下のサービス層コンポーネントに配置されています。

```
<bfinstall>/Apache/tomcat/webapps/jas/WEB-INF/eclipse/plugins/com.ibm.jas-1.0.jar
```

インターフェースは以下のようなメソッドを提供します。

#### **initInterceptor**

インターセプターのロード時に呼び出されます。構成プロパティのマップは、`initInterceptor()` メソッドに渡されます。構成プロパティは、Build Forge コンソールの「管理」 > 「セキュリティ」 > 「SSO」で作成します。

#### **isTargetInterceptor**

インバウンド要求の属性を確認し、このインターセプターがそれらの属性に対処する必要があるかどうかを判別します。対処が必要な場合、インターセプターは `authenticateRequest()` メソッドで要求を認証します。対処が不要な場合、このインターセプターはスキップされます。インターセプターの選択では、複数のインターセプターが構成されて実行されることを想定しています。これらは順次対処されます。

#### **authenticateRequest**

要求内のデータを使用して、要求を認証します。応答属性を使用して、データをクライアントに返送します。

#### **logoutRequest**

要求の処理後に、ユーザー関連のセキュリティ情報をすべてクリーンアップします。

**インターセプターの構成および配列:** インターセプターの構成は、「管理」 > 「セキュリティ」 > 「SSO」で定義します。Build Forge® には、以下の構成が組み込まれています。

- フォーム SSO インターセプター - デフォルトでアクティブになっており、単純なログイン・フォームをインプリメントします。
- SPNEGO SSO インターセプター - デフォルトでは非アクティブになっており、SPNEGO をインプリメントして認証を実行します。

インターセプター・クラスをインプリメントして、Build Forge® Apache Tomcat アプリケーション・サーバーに配置した後、ここで新規 SSO 構成を構成します。このクラスは、SSO 構成の 1 プロパティです。



このリストの配列は、インターセプターが要求処理の打診を受ける順序を決定します。複数のインターセプターを構成し、要求を処理することができます。ログインの間に、各インターセプターは順次打診を受けます。要求内の属性に適した属性を持っているインターセプターで、最初にアクティブになっているものが要求を処理します。1 つの要求は、単独のインターセプターで処理します。これは常に、`isTargetInterceptor` に対して最初に `true` を返したインターセプターです。

**注:** エラー発生時にフォールバックを提供できるように、フォーム SSO インターセプターをアクティブな状態にしておいてください。リストでは、フォーム SSO インターセプターの前にカスタム・インターセプターを配置してください。

**カスタム SSO インターセプターの追加:** Build Forge® にカスタム・インターセプターを作成するには、以下の手順を実行します。

1. カスタム Java クラスを作成します。

このクラスには、`SSOInterceptor` インターフェースをインプリメントする必要があります。

2. 作成したカスタム・クラスを、サービス層コンポーネント WAR ファイルにデプロイします。

a. コンパイル済みのカスタム SSO インターセプター・クラスを含む JAR ファイルを作成します。

b. JAR ファイルを、Build Forge® サービス層 JAR ファイル (場所は `<bfinstall>/server/tomcat/work/Catalina/localhost/jas/eclipse/plugins/com.ibm.jas-xxx.jar`) にマージします。

c. Build Forge® を再始動します。

3. 環境を定義します。環境は、プロパティー・オブジェクトとして `initInterceptor()` メソッドに渡すことができます。

a. 管理コンソールで、「環境変数」に移動します。

b. 「環境変数の追加」をクリックします。

c. SSO インターセプターを初期化するために必要なプロパティーをすべて定義します。

4. SSO インターセプターを Build Forge® に追加します。

a. 管理コンソールで、「管理」 > 「セキュリティ」 > 「SSO」と移動します。

b. 「SSO 構成の追加」をクリックして、プロパティーを入力します。

- 「名前」 - SSO 構成の名前を入力します。

- 「アクティブ」 - 「はい」に設定します。認証要求の間に、すべてのアクティブな構成はアクセスを受けます。アクセスを受ける順序は、パネルで表示される順序と同じです。

- 「Java クラス」 - クラスの完全なパッケージ名を入力します。所定のクラスは、1 つの SSO インターセプターにのみ割り当てることができます。

- 「環境変数」 - この SSO インターセプターで使用する環境を定義した場合には、その環境を選択します。

c. 「保存」をクリックします。

追加した SSO インターセプターがリストに表示されるはずです。

5. SSO インターセプターの左側にあるアイコンをクリックしてから「最上部に移動」を選択し、SSO 構成の順序を設定します。

要求の間、アクティブな SSO 構成は、パネルに表示されている順序でアクセスを受けます。この構成はデフォルトでアクティブであり、アクセスを受けると必ず true を戻すため、カスタムの構成は「フォーム SSO」構成の前に配置する必要があります。SPNEGO SSO 構成は、デフォルトでは非アクティブです。

**authenticateRequest インプリメンテーションの例:** 以下は、WebSphere SSO インターセプターを使用して、WebSphere Application Server セキュリティーを Build Forge® に統合した例です。

このインターセプターは、リフレクションを使用して WebSphere クラス WSSubject を検出します。このクラスには、AuthServlet へのログインに使用するプリンシパルを返す、getCallerPrincipal メソッドがあります。AuthServlet は、WebSphere Application Server が認証に使用する前に保護する必要があります。

さらに多くの情報を返すことができる、その他のメソッドが使用可能です。任意のアプリケーション・サーバーで動作可能な、同様のメソッドが使用可能です。

```
public Result authenticateRequest
    (Request requestAttributes, Response responseAttributes)
    throws SSOException {

    Result result = null;

    try {
        Class<?> c1 =
            Class.forName("com.ibm.websphere.security.auth.WSSubject");
        Method theMethod = c1.getMethod("getCallerPrincipal",
            (Class[])null);
        String principal = (String)theMethod.invoke((Object[])null,
            (Object[])null);

        if (principal != null
            && principal.length() > 0
            && !principal.equals("UNAUTHENTICATED")) {
            result = new Result(Result.UseridOnlyOID, domain, principal);
            responseAttributes.setStatus(HttpServletResponse.SC_OK);
        } catch (Exception e) {
            throw new SSOException(e);
        }

        return result;
    }
}
```

authenticateRequest のインプリメンテーションの間に、結果を返す前の応答状況を以下のように設定する必要があります。

- リダイレクトを実行する必要がなく、検出された情報が良好である場合には、次の情報を返します。

```
responseAttributes.setStatus(HttpServletResponse.SC_OK);
```

- 要求内の情報が、有効なログインを続行する上で不十分である場合は、次の情報を返します。

```
responseAttributes.setStatus(HttpServletResponse.SC_FORBIDDEN);
```

- 追加情報を収集するためにリダイレクトを実行する必要がある場合は、次の情報を返します。

```
responseAttributes.setStatus(HttpServletResponse.SC_MOVED_TEMPORARILY);
responseAttributes.sendRedirect(url);
```

使用可能な状況値はその他にもあります。HttpServletResponse に関する JavaDoc を参照してください。

**ログイン・エラーからのリカバリー:** カスタム・インターセプターがテスト時に正しく動作しない場合、最も問題がある可能性が高いのは認証です。以下の情報を示すエラー・ページが表示されます。

Build Forge エラー

```
Access is denied to the Build Forge console
```

```
"Error authenticating:
com.buildforge.services.common.api.APIException - API:
Authentication Error."
```

同じタイプのログインを再試行する場合は、[ここをクリックするか](#)、強制的にフォーム・ログイン（ユーザー ID/パスワード）を実行する場合は[ここをクリックしてください](#)。

リカバリーには以下の 2 つのオプションがあります。

- 「ログインの再試行 (Retry the login)」。これは、構成済みのインターセプターのリストを、同じ方法で再度処理する方法です。
- 「フォーム・ログインの強制 (Force a form login)」。これを選択すると、カスタム・インターセプターが迂回され、フォーム・ログイン・ページが使用されます。

**メソッド・ソースのリスト:** 以下のコメントおよびソースのリストは、SSOInterceptor インターフェース内のメソッドに関する詳細情報を提供します。

#### initInterceptor

```
/**
 * This method is called when the interceptor is loaded. A map of the
 * configuration properties is passed into the init method. You can create
 * the configuration properties from a BuildForge Environment and associate
 * it with the SSO configuration.
 *
 * @param initializationProps used to configure the implementation
 * @return true if successful, false if an error should be reported.
 * @throws SSOException if the initialization fails
 */
public boolean initInterceptor (Properties initializationProps) throws SSOException;
```

#### isTargetInterceptor

```
/**
 * This methods will review the attributes in the requestAttributes Map
 * to determine if there is something that this interceptor should
 * act on. If the interceptor return is "true", then the interceptor will
 * be responsible for authenticating the request and the authenticateRequest
 * method is invoked. If the interceptor return is "false", then this
 * interceptor is skipped and the next isTargetInterceptor in the list will
 * be called. Ordering of the interceptors during the configuration will
 * return which interceptor has the first shot at authenticating a request.
 *
 * @param requestAttributes attributes found in the inbound request
 * @return true if this interceptor will authenticate the request,
 *         false if it will not.
 * @throws SSOException
```

```

*
**/
public boolean isTargetInterceptor(Request requestAttributes) throws SSOException;

```

#### **authenticateRequest**

```

/**
 * This method is called on an interceptor that returns true for the
 * isTargetInterceptor method. The Request will contain data used
 * to perform the authentication. The Response is for the interceptor
 * to send information back to the client. The Result returned will contain
 * the following information if the status code is 200:
 *
 *
 * OID: an object identifier of the SecurityContext that can process token
 * information stored in this map when going to an Agent.
 * Domain: a valid BF domain name or <default> if not known
 * (the username must be valid in the configured realm).
 * Username: a valid BF username. This will be used to lookup BFUser attributes
 * that are used in checking authorization policy.
 * @see com.buildforge.services.common.security.context.Result
 *
 *
 * @param requestAttributes attributes found in the inbound request
 * @param responseAttributes sent back in the outbound response
 * @return com.buildforge.services.common.security.context.Result - result
 *         information that tells BF how to handle the authentication request.
 * @throws com.buildforge.services.server.sso.SSOException
 */
public Result authenticateRequest(
    Request requestAttributes,
    Response responseAttributes)
    throws SSOException;

```

#### **logoutRequest**

```

/**
 * This method is called to logout a request. The first interceptor that
 * returns true for the isTargetInterceptor method will perform the logout.
 * The main point is to clean up any user-related security information that
 * should not be kept. The interceptor can inspect the request and response
 * objects to determine what needs to be removed.
 *
 *
 * @param requestAttributes attributes found in the inbound request
 * @param responseAttributes sent back in the outbound response
 * @return boolean - true if request redirect to exit page,
 *                 false if redirect to login page.
 * @throws com.buildforge.services.server.sso.SSOException
 */
public boolean logoutRequest(
    Request requestAttributes,
    Response responseAttributes)
    throws SSOException;

```

## **Active Directory ドメインにおける SPNEGO を使用したシングル・サインオンの実装**

Active Directory ドメインでシングル・サインオンを実装する際に、Simple and Protected GSS-API Negotiation (SPNEGO) 機構を使用することができます。

### **始める前に**

この作業を行うには、ネットワーク上に次の要素が必要です。

- Active Directory ドメイン
- ディレクトリー・サーバーのホスト名
- Kerberos 鍵配布センター (KDC) のホスト名

- Active Directory ドメインのホスト上の Build Forge インストール済み環境
- Active Directory ドメイン上のクライアント・ホスト
- 各クライアント上の Kerberos 構成ファイル
- ディレクトリー・サーバー・ホストにインストールされた Windows Server 2003 SP2 リソース・ツールキット
- サポートされている Web ブラウザー

注: SPNEGO を使用する場合には、Internet Explorer 6 はサポートされません。サポートされているブラウザを使用してください。

以下の手順にはいくつかの例が含まれており、次のようにセットアップされています。

- mycompany.com は、ドメインのすべてのホストが使用する TCP/IP ドメインの名前です。
- ITDEV.COM は、Active Directory ドメインの名前です。
- it\_directory.mycompany.com は、ディレクトリー・サーバーが稼働しているホストです。ここでは、Kerberos KDC も稼働しています。
- it\_domain.mycompany.com は、Active Directory ドメイン・コントローラーが稼働しているホストです。
- it\_buildforge.mycompany.com は、Build Forge がインストールされているホストです。
- bfuser は、Build Forge システムのドメイン・ユーザー名です。
- happy\_user は、ドメイン・ユーザー名です。例えば Web ブラウザーで SSO を使用して Build Forge にアクセスするユーザーなどです。

## このタスクについて

Active Directory ドメインおよび KDC に SPNEGO SSO を実装するには、次の作業を行います。各作業はセクションに分けられており、それぞれ詳細な手順が記載されています。

注: SPNEGO インターセプターは、Active Directory 以外の KDC で使用できません。

## 手順

1. Active Directory のユーザーとサービス・プリンシパルをセットアップします。
2. Kerberos ファイルをセットアップします。
3. Build Forge で Active Directory と SPNEGO が使用されるように構成します。
4. セキュアなアクセスができるように、ブラウザ・クライアントを構成します。
5. SSO を使用して Build Forge にアクセスします。

### Active Directory ユーザーとサービス・プリンシパルのセットアップ:

Build Forge サーバーと Build Forge クライアントを、Active Directory ドメインにセットアップする必要があります。

## 始める前に

以下の手順を実行するには、Windows 2003 SP2 に対応したサポート・ツールが必要です。これらのツールには、`setspn` コマンドが含まれており、Active Directory でサービス・プリンシパルを設定する必要があります。Windows Server 2003 製品 CD、または Microsoft Download Center から、サポート・ツールをインストールします。

## このタスクについて

Build Forge クライアントおよびサーバーが Active Directory ドメインにある場合、ユーザーは、Windows ホストにログインするときに、Kerberos 証明書トークンを生成します。生成後、ユーザーが Build Forge サーバーにアクセスしようすると、SPNEGO インターセプターはユーザー・トークンを受け取り、それを検証します。検証済みの ID は Build Forge に渡され、構成された Microsoft Active Directory LDAP サーバーを介してログインが行われます。

## 手順

1. ドメイン・コントローラー・ホストへのログオン 例では、ホストは `it_example.mycompany.com` です。
2. Build Forge ホストがまだメンバーではない場合は、これを Active Directory ドメインに追加します。この例では、ホスト `it_buildforge` をドメイン `ITDEV.COM` に追加します。このドメインにおけるホストの完全修飾名は、`it_buildforge.ITDEV.COM` になります。
3. Build Forge ユーザーを Active Directory ドメインに追加します。この例では、ユーザー `bfuser` を作成します。

### 重要:

- 「パスワード期限なし (Password never expires)」を選択します。他のパスワード管理を選択することもできます。ただし、期限が切れるたびに、Build Forge サーバーで新しいパスワードを入力する必要があります。
  - 「アカウント」タブで、「アカウントは委任に対してトラステッド (Account is trusted for delegation)」を選択します。
4. ユーザー・アカウントがない場合、すべてのクライアントに対して、Microsoft Active Directory でアカウントを作成します。この例では、ユーザー `happy_user` を作成します。
  5. Build Forge のサービス・プリンシパル名 (SPN) を作成します。以下の例では、Active Directory ユーザーの `bfuser` が、サービス名 `HTTP/it_buildforge.mycompany.com` に関連付けられ、Build Forge サーバーの SPN `it_buildforge` が作成されます。

```
setspn -A HTTP/it_buildforge.mycompany.com bfuser
```

HTTP は、Build Forge サービスのサービス名です。

## Kerberos 認証用のファイルのセットアップ:

スタートアップ・ファイル (Kerberos クライアント構成ファイル) および keytab ファイルを、Build Forge ホスト上にセットアップする必要があります。



## 手順

1. スタートアップ・ファイルを、Build Forge が稼働しているホスト上にセットアップします。

- Windows システム
  - krb.ini ファイルを指定し、それを C:¥winnt に置きます。まだ作成していない場合は、C:¥winnt を作成します。
  - default\_keytab\_name を FILE:C:¥winnt¥krb5.keytab に設定します。
- UNIX システムおよび Linux システム
  - krb.conf ファイルを指定し、それを C:¥winnt に置きます。
  - default\_keytab\_name を FILE:/etc/krb5.keytab に設定します。

サンプル・システムのドメインおよび領域設定を使用して、Windows 用に以下のサンプル・ファイルがセットアップされます。

```
[libdefaults]
default_realm = ITDEV.COM
default_keytab_name = FILE:C:¥winnt¥krb5.keytab
default_tkt_enctypes = rc4_hmac
default_tgs_enctypes = rc4_hmac
# kdc_default_options = 0x40800000
forwardable = true
renewable = true
noaddresses = true
clockskew = 300
[realms]
ITDEV.COM = {
    kdc = it_directory.itdev.com:88
    default_domain = mycompany.com
}
[domain_realm]
.mycompany.com = ITDEV.COM
```

注: クライアント・ホストと Build Forge サーバー・ホスト間のクロック・スキューが 300 秒を超えている場合、トークンは作動しません。クライアントおよびサーバー・ホスト上に、スキュー制限内で、時刻、日付、およびタイムゾーンを設定してください。

2. Kerberos keytab ファイルをセットアップします。Build Forge サーバーは、keytab ファイルを使用して、クライアントが Build Forge サーバーの URL にアクセスしようとするときに、Kerberos トークンを検証します。ドメイン・コントローラー・ホストで ktpass コマンドを使用して、このファイルを作成します。ktpass コマンドは、前提条件である Windows リソース・ツールキットに含まれています。以下の例では、Build Forge サービスのプリンシパル名と、サンプル・シナリオで Build Forge 用にセットアップされた Active Directory ユーザー名が使用されています。-pass Rational を自分のパスワードに置き換えます。分かりやすくするために、例では改行が表示されています。ktpass コマンドでは、改行を使用しないでください。

```
ktpass -out C:¥it_buildforge.keytab
-princ HTTP/it_buildforge.mycompany.com@ITDEV.COM
-mapuser bfuser -mapop set
-pass Rational /crypto RC4-HMAC-NT /rndpass /ptype KRB5_NT_SRV_HST
```

it\_buildforge.keytab を krb5.keytab に名前変更し、Kerberos スタートアップ・ファイルが置かれているディレクトリーにある Build Forge ホストに置きます。

- Windows の場合: C:¥winnt¥



- UNIX および Linux の場合: /etc

## Build Forge で Active Directory と SPNEGO が使用されるための構成:

### 手順

1. Build Forge で、LDAP が Active Directory ドメイン・コントローラーを指定するように設定します。
  - a. Build Forge で、「管理」 > 「LDAP」とクリックします。
  - b. 新規 LDAP 構成を作成し、以下のようにプロパティを設定して、ドメイン・コントローラーへのアクセスをセットアップします。
    - 名前: Active Directory ドメインの名前に設定します。例の環境では、itdev になります。
    - 管理者 DN: ドメイン内の管理者ユーザーに設定します。
    - アクセス・グループのマップ: いいえ
    - ホスト: ドメイン・コントローラー・ホストの IP アドレスに設定します。
    - ユーザー・アカウントのバインド: はい
    - プロトコル: LDAP
    - 表示名: displayname
    - 識別名: distinguishedname
    - グループ名: memberof
    - メール名: displayname
    - 検索基準: on=users、do=domainname、do=domainextension。この例の環境では、on=users,do=itdev,do=.com になります。
    - 検索フィルター: sAMAccountNames=%
  - c. 「デフォルトの設定」をクリックします。この構成を、デフォルトの LDAP 構成にする必要があります。
2. Build Forge 環境変数を SPNEGO に対応するように設定します。
  - a. Build Forge で、「環境変数」 > 「SPNEGO SSO の環境変数」と移動します。
  - b. bf\_spnego\_service\_name を HTTP に設定します。これは、サービス・プリンシパル名と同一です。
  - c. bf\_spnego\_server\_name を it\_buildforge.mycompany.com (Build Forge サーバー・ホストの完全修飾ホスト名) に設定します。この変数が設定されていないと、INetAddress API はホスト名の場所を探索しようとします。
  - d. bf\_spnego\_realm を ITDEV.COM (Kerberos 領域名) に設定します。この変数が設定されていない場合、Kerberos スタートアップ・ファイル内の値が使用されます。
3. SPNEGO インターセプターを有効に設定します。
  - a. Build Forge で、「管理」 > 「セキュリティ」 > 「SSO」 > 「SPNEGO SSO インターセプター」と移動します。
  - b. 「アクティブ・プロパティ (Active property)」を「はい」に設定し、「保存」をクリックします。

- c. 「管理」 > 「セキュリティー」 > 「SSO」とクリックして、「SPNEGO SSO インターセプター」をリストの最上部に移動します。「SSO オプション」メニューの「最上部に移動」を選択して、SPNEGO SSO インターセプターを移動し、「保存」をクリックします。

#### SSO に対応したクライアント・ブラウザーの構成:

クライアント・ブラウザーで、セキュリティー設定に SPNEGO が使用されるようにセットアップする必要があります。

#### このタスクについて

Build Forge へのアクセスに使用されるブラウザー (Microsoft Internet Explorer または Mozilla Firefox) のクライアント・セットアップ手順に従います。

#### 手順

- Internet Explorer の場合、以下の手順に従います。
  1. Active Directory のドメインにログインします。 例の構成では、itdev.com にログインします。
  2. Internet Explorer を開きます。
  3. 「ツール」 > 「インターネット オプション」 > 「セキュリティ」とクリックします。
  4. 「ローカル イン트라ネット」を選択し、「サイト」をクリックします。
  5. 「ローカル イン트라ネット」ダイアログで、「ほかのゾーンにないローカル (イントラネット) のサイトをすべて含める」にチェックマークを付け、「詳細設定」をクリックします。
  6. Build Forge を実行するホストを「Web サイト」リストに追加し、「OK」をクリックします。
  7. 「OK」をクリックして、「ローカル イン트라ネット」ダイアログを閉じます。
  8. 「インターネット オプション」ウィンドウで、「詳細設定」タブをクリックします。
  9. 「セキュリティ」グループまでスクロールダウンして、「統合 Windows 認証を使用する」を見つけ、これが選択されていない場合は、選択します。
  10. 「OK」をクリックします。
  11. Internet Explorer を再始動します。
- Mozilla Firefox の場合、以下の手順に従います。
  1. Active Directory のドメインにログインします。 例の構成では、itdev.com にログインします。
  2. Firefox を開きます。
  3. アドレス・フィールドに「about:config」と入力します。
  4. 「フィルタ」ボックスに「network.n」と入力します。 リストが更新されます。
  5. 「network.negotiate-auth.trusted-uris」をダブルクリックします。 トラステッド・ドメインのリストを入力します。このリストには、ディレクトリー・サーバー・ホストと Build Forge サーバー・ホスト (この例では、

it\_directory.mycompany.com および it\_buildforge.mycompany.com) が含まれている必要があります。「OK」をクリックします。

6. 委任をセットアップします。「network.negotiate-auth.delegation-uris」をダブルクリックして、ブラウザがユーザー認証を委任するサイトのリストを入力します。

### SSO を使用した Build Forge へのアクセス:

サーバー URL を入力して、SSO を使用したログインをテストします。

#### 手順

1. ユーザーの Active Directory リストにあるユーザー名を使用して、Active Directory ドメインにあるホストにログインします。
2. ブラウザーを開きます。
3. Build Forge サーバー・ホストの URL を入力します。 サンプルの構成では、これは http://it\_buildforge.mycompany.com です。 SSO が正しく設定されていると、Build Forge 管理コンソールが表示されます。
4. Build Forge コンソールの右上に表示されているユーザー名が、クライアントの Windows ログイン名と一致していることを確認します。

## カスタム・インターセプターを使用した WebSphere Application Server セキュリティーとの統合

このセクションでは、SSO インターセプターを作成して、WebSphere Application Server セキュリティーと統合する方法を説明します。

### 始める前に

ユーザーは、LDAP ユーザー資格情報を使用して WebSphere Application Server で構成されている必要があります。

注: WebSphere Application Server ユーザーを含む Build Forge LDAP ドメインを「デフォルト」の LDAP サーバーにする必要があります。これを行うには、この Build Forge LDAP ドメインに移動し、「デフォルトの設定」を選択します。

### このタスクについて

提供されるフォーム SSO インターセプターは、フォーム・ベースのログイン・ページを使用して、ユーザーを認証します。以下に、カスタム SSO インターセプターの作成方法の例を示します。カスタム・インターセプターは、カスタム・インターセプター・クラスを使用します。

このインターセプター・クラスは、WebSphere Application Server にアクセスして、認証済みのユーザー資格情報を取得します。これらの資格情報は、取得後にキャッシュされます。その後のログインには、キャッシュされた資格情報が使用されます。

### 許可サービス (AuthServlet) の保護:

Build Forge では、通常、指定された Apache Tomcat アプリケーション・サーバーのアプリケーションとして、サービスを実行します。

## このタスクについて

以下の説明では、Tomcat ではなく WebSphere Application Server を使用するように Build Forge を構成する方法を示します。629 ページの『Apache Tomcat の代わりに WebSphere Application Server を使用』セクションに記載された説明に従ってください。ただし、セキュリティ制約を追加するために、`jas.war` ファイルに含まれている `web.xml` という名前のファイルを抽出および変更する必要がある点だけが異なります。このアプリケーションを WebSphere Application Server にインストールする前に、アプリケーションが `war` ファイルの保護されたバージョンを使用するようにアプリケーションを修正してから、このファイルを再生成する必要があります。

`war` ファイルを再生成するには、以下の手順を実行します。

### 手順

1. `jas.war` ファイルを含むディレクトリー (Tomcat サーバー・ルート of `webapps` ディレクトリー) に移動します。このファイルを一時的なロケーション (例えば、`C:\jas`) にコピーします。
2. `WAR` ファイルを展開します。コマンド行から、`%IBM_JAVA_HOME%\jar -xvf jas.war` コマンドを実行して、`war` ファイルの内容を展開します。

注: Java が使用可能になっている必要があります。また、`IBM_JAVA_HOME` 環境変数が作成されている必要があります。

3. 後でできるように、`jas.war` ファイルを保存しておきます。
  - a. Windows の場合: `copy jas.war jas.war.bak`
  - b. UNIX または Linux の場合: `cp jas.war jas.war.bak`
4. `WEB-INF` ディレクトリーで `web.xml` ファイルを (`war` ファイルから展開されたファイルから) 探します。このファイルを編集して、セキュリティ・コンテキストを追加します。例えば、ファイルの末尾の `</web-app>` タグのすぐ前に次の行を追加します。

```
<security-constraint id="SecurityConstraint_1">
  <web-resource-collection id="WebResourceCollection_1">
    <web-resource-name>*/</web-resource-name>
    <url-pattern>/AuthServlet/*</url-pattern>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
    <http-method>PUT</http-method>
    <http-method>DELETE</http-method>
  </web-resource-collection>
  <auth-constraint id="AuthConstraint_1">
    <description>myconstraint:+:</description>
    <role-name>User</role-name>
  </auth-constraint>
  <user-data-constraint id="UserDataConstraint_1">
    <transport-guarantee>NONE</transport-guarantee>
  </user-data-constraint>
</security-constraint>
<login-config id="LoginConfig_1">
  <auth-method>BASIC</auth-method>
  <realm-name>full-qualified-domain</realm-name>
</login-config>
<security-role id="SecurityRole_1">
  <role-name>User</role-name>
</security-role>
```

注: <auth-method> は、WebSphere Application Server でサポートされる任意の J2EE 認証メソッドにすることができます。最も一般的な認証メソッドは FORM ですが、これを使用するには、いくつかの構成パラメーターを追加する必要があります。アプリケーションで FORM を構成するための説明は、WebSphere Application Server の資料を参照してください。

5. コマンド行から次のコマンドを実行して、war ファイルを再生成します (ファイルの抽出先と同じディレクトリーから呼び出します)。

```
%IBM_JAVA_HOME%¥jar -cvf jas.war
```

これで、jas の新しいバージョンが作成されました。このバージョンは、J2EE 制約を使用して AuthServlet を保護するように修正されています。WebSphere Application Server で実行するためのインストール手順を完了し、「アプリケーション」->「新規アプリケーションのインストール (Install New Application)」とクリックして、jas.war のこの新しいバージョンをインストールします。WebSphere Application Server の構成中に、Build Forge が実行していないことを確認してください。

インストール後に、「アプリケーション」>「エンタープライズ・アプリケーション (Enterprise Applications)」の順に進みます。アプリケーション名をクリックして構成します。「詳細プロパティー (Detail Properties)」の下で、「ユーザー/グループ・マッピングのセキュリティー・ロール (Security role to user/group mapping)」リンクをクリックします。ユーザーに対して、「すべて認証対象 (All Authenticated)」チェック・ボックスを選択します。この変更を行ったら、マスター構成に保存してください。

アプリケーション・セキュリティーも WebSphere Application Server で有効に設定される必要があります。これを行うには、「セキュリティー」>「管理、アプリケーション、およびインフラストラクチャーの保護 (Secure administration, applications, and infrastructure)」と移動します。「アプリケーション・セキュリティーを有効に設定 (Enable application security)」にチェック・マークが付けられていることを確認します。

この時点で、WebSphere Application Server を再始動し、続いて Build Forge サーバーを再始動します。

注: これで、jas が WebSphere Application Server の始動プロセスの一環として開始されるようになったため、Build Forge を開始する前に WebSphere Application Server を開始してください。

### 新規 SSO 構成の作成:

新規 SSO 構成を作成して、インターセプターを使用します。

#### 手順

1. Build Forge コンソールで、「管理」>「セキュリティー」>「SSO」と移動します。
2. 「SSO 構成の追加」をクリックします。
3. 構成用に、プロパティーを設定します。
  - **Name** - この構成の名前を入力します。

- **Java Class** -  
com.buildforge.services.server.sso.was.WebSphereSSOInterceptor を入力します。
- **Active** - 「はい」を選択します。

4. 「保存」をクリックします。

5. この構成がリストの最初に表示されるように移動させます。 構成名の左にあるメニューから、「**最上部に移動**」を選択します。

#### **LDAP ユーザーまたは LDAP グループのローカル・ユーザーへのマッピング:**

LDAP ユーザーまたは LDAP グループをローカル root ユーザーなどのローカル・ユーザーにマップします。

#### **このタスクについて**

WebSphere Application Server SSO インターセプターの作成後は、以下を実行するようにインターセプターを構成できます。

- LDAP ユーザーを root などのローカル・ユーザーにマップする
- LDAP グループを root などのローカル・ユーザーにマップする

マッピングをセットアップするには、環境変数を定義して、インターセプターでその環境変数を参照します。

#### **環境変数の定義:**

##### **手順**

1. Build Forge コンソールで、「**環境変数**」をクリックします。
2. 環境変数の名前を指定して、「**環境変数の保存**」をクリックします。
3. LDAP ユーザーをローカル・ユーザーにマップするには、以下の変数を設定して保存します。
  - a. ldap\_user\_list\_mapping を  
LDAP\_user1|local\_user1;LDAP\_user2|local\_user2;LDAP\_user3|local\_user3  
に設定します。
  - b. ldap\_realm\_name を LDAP\_domain\_name に設定します。

例えば、以下の環境変数が設定されたとします。

```
ldap_user_list_mapping = user1|root;user2|root;user3|root
ldap_realm_name = bluepages.ibm.com:389
```

この場合、LDAP の user1、user2、および user3 がすべてローカル root ユーザーにマップされます。

4. 特定の LDAP グループのメンバーをローカル・ユーザーにマップするには、以下の変数を設定して保存します。
  - a. ldap\_group\_name\_mapping を LDAP\_group|local\_user に設定します。
  - b. ldap\_realm\_name を LDAP\_domain\_name に設定します。

**注:** 両方のタイプのマッピングを使用した場合は、グループ・マッピングよりもユーザー・マッピングが優先されます。

グループ・マッピングの例として、以下の環境変数が設定されたとします。



```
ldap_group_name_mapping = cn=bf_admin,ou=memberlist,ou=ibmgroups,o=ibm.com|root
ldap_realm_name = bluepages.ibm.com:389
```

この場合、LDAP グループ `cn=bf_admin,ou=memberlist,ou=ibmgroups,o=ibm.com` のメンバーであるユーザーが、ローカル `root` ユーザーにマップされます。

### 環境変数の参照:

#### 手順

1. Build Forge コンソールで、「管理」 > 「セキュリティ」 > 「SSO」とクリックします。
2. WebSphere Application Server SSO インターセプターの名前をクリックします。
3. 構成の「環境変数」プロパティを、上で定義した環境変数に設定します。
4. 「保存」をクリックします。
5. この構成がリストの最初に表示されるように移動させます。 構成名の左にあるメニューから、「最上部に移動」を選択します。

### SSO カスタム・インターセプターの実行:

新しい構成を使用してログインできるようになりました。

#### このタスクについて

このカスタマイズされた SSO インターセプターにより、WebSphere Application Server セキュリティ技法でユーザーを認証できるようになりました。認証は、AuthServlet 要求がユーザーとして Build Forge に渡されることで行われます。

#### 手順

1. ご使用の Web ブラウザーを開き、アドレス `http://localhost` を入力します。 Build Forge ログイン・フォームではなく、認証ページが表示されます。
2. 自分の資格情報を入力して、**Enter** をクリックします。

注: 認証後、ログインが自動的行われます。

ログアウトすると、ログイン・フォームではなく、デフォルトの `jsp` ページが表示されます。これ以降、ユーザーが認証されれば、ログインは自動的行われます。

### フォーム・ベースの SSO への復帰:

SSO ログイン・フォームを使用するように戻すことができます。

#### このタスクについて

フォーム・ログインを使用するようにシステムを再構成するには、WebSphere Application Server で `jas` をアンインストールし、元の `jas.war` ファイルを再インストールする必要があります。Build Forge で、フォーム・ベースの SSO インターセプターが有効に設定され、最初の項目としてリストされていることを確認します。カスタム WebSphere Application Server インターセプターを無効に設定します。行った変更を有効にするには、WebSphere Application Server および Build Forge を再始動する必要があります。



## 手順

1. jas をアンインストールし、元の jas.war ファイルを再インストールします。
2. Build Forge で、フォーム SSO インターセプターが有効に設定され、最初の項目としてリストされていることを確認します (上記の「Build Forge SSO セキュリティー構成 (Build Forge SSO security configurations)」を参照)。
3. カスタム WebSphere Application Server インターセプターを無効に設定します。
4. WebSphere Application Server を再始動します。
5. Build Forge を再始動します。

## SSL および HTTPS の有効化

SSL および HTTPS を使用するように Build Forge システムを構成すると、システムのセキュリティが向上します。SSL では、エンドポイント認証およびデータ暗号化を実行します。

**注:** Rational Build Forge 8.0.0.1 以降では、2048 ビット長の暗号化された自己署名 SSL 証明書がサポートされています。

デフォルトでは、Build Forge システムは、Apache Tomcat の認証サーブレットを使用するログイン・フォームの場合のみ SSL を使用します。Build Forge システム全体に SSL 保護を拡張するには、以下のセットアップが必要です。

1. Apache サーバーでの SSL の有効化。このステップは、インストール時にこの SSL の有効化を指定しなかった場合のみ必要です。
2. クライアントおよび内部通信用の SSL の有効化
3. エージェントの SSL の有効化

**注:** WebSphere コンポーネントと統合する場合は、SSL サポートの前提条件を満たしていることを確認してください。

- WebSphere Application Server と統合する場合は、629 ページの『Apache Tomcat の代わりに WebSphere Application Server を使用』を参照してください。

## デフォルト・ログイン・セキュリティについて

Build Forge には、デフォルトでログイン・セキュリティが組み込まれています。ユーザーのログイン時に、要求は認証サーブレットにリダイレクトされます。入力されたユーザー名およびパスワードは、サーブレットで使用するために暗号化されます。ログインが正常に行われると、コンソール・ユーザー・インターフェースのホームが表示されます。この後にクライアントとコンソールの間で行われるセッション通信は、http (デフォルト) または https を介して行われます。https を使用するには、追加のシステム構成が必要です。『SSL および HTTPS の有効化』を参照してください。

インストール時に、暗号化のために使用される鍵ストア用のパスワードを指定します。自己署名証明書をインストールするというオプションもあります。

**自己署名証明書に関するメッセージの認証:** Build Forge に自己署名証明書がインストールされると、セキュリティ対応のブラウザー経由でシステムにアクセスしているユーザーは、その証明書に関する警告メッセージを受け取ります。

この警告は、自己署名証明書をユーザーに配布して、それぞれのブラウザーにインストールすることで表示されなくなります。証明書のインストールの詳細は、ブラウザーごとに異なります。ブラウザーの資料を参照してください。

証明書は `<bfinstall>/keystore` にあります。

**デフォルト・ログイン・セキュリティの無効化:** ログイン・セキュリティが無効になっている場合、ユーザー資格情報はログイン時に、暗号化されていない平文でコンソールに通知されます。HTTPS/SSL を使用するようにコンソールが構成されている場合にログイン・セキュリティを無効にしても、コンソールによるHTTPS/SSL での使用には影響を与えません。

認証サーブレットを無効にするには、以下の手順を実行します。

1. Build Forge が稼働中であれば停止します。
2. `<bfinstall>/buildforge.conf` を編集し、サービス層との通信に HTTP およびポート 8080 を指定します。

以下の行を見つけます。

```
services_url https://hostname:8443/jas
```

これを次のように変更します。

```
services_url http://hostname:8080/jas
```

3. サービス層構成ファイルを強制 SSL がオフになるように編集します。  
`<bfinstall>/Apache/tomcat/webapps/jas/WEB-INF/web.xml` を編集します。以下のように、ForceHttps 設定を `false` に変更します。設定がファイルにない場合は、太字で示したとおりに追加してください。

```
<display-name>A Services Layer Bootstrap Servlet</display-name>
<servlet>
  <servlet-name>ServicesBootstrap</servlet-name>
  <servlet-class>com.buildforge.services.server.web.BootstrapServlet
  </servlet-class>
  <init-param>
    <param-name>port</param-name>
    <param-value>3966</param-value>
  </init-param>
  <init-param>
    <param-name>sslPort</param-name>
    <param-value>49150</param-value>
  </init-param>
  <init-param>
    <param-name>ForceHttps</param-name>
    <param-value>false</param-value>
  </init-param>
  <load-on-startup>0</load-on-startup>
</servlet>
```

4. Build Forge を開始します。

**注:** 認証サーブレットが無効になっている場合、ユーザーの資格情報はネットワーク上で平文で転送されますが、これはセキュリティ・リスクの要因となります。

## SSL および Build Forge コンポーネントについて

Build Forge コンポーネントはデフォルトでセットアップされ、SSL が有効である場合には特定のポートとセキュリティ設定を使用します。

**デフォルトの SSL セットアップ:** デフォルトの証明書を使用している場合、SSL の有効化は比較的単純です。このセクションに示す手順は、そのシナリオに基づいています。

しかし、一般に、各システムで同じ証明書 (秘密鍵) を使用することはお勧めできません。1 つのシステムで秘密鍵が漏えいすると、インフラストラクチャー全体が危険にさらされる可能性があります。物理的セキュリティを強化することで、漏えいの危険性は削減できます。

比較的安全なシステムでは、プロセスごとに証明書を使用します。これは、Build Forge では以下を行うことを意味します。

- 各エージェントに証明書を作成します。
- 各エンジンに証明書を作成します。これは、冗長化がセットアップされている場合に該当します。102 ページの『冗長化構成』を参照してください。

このセットアップには、追加の証明書管理が必要です。以下のような選択肢があります。

- CA (認証局) を使用して証明書を生成します。こうすることで、署名者の交換数を削減できます。
- すべてのトラストストアまたは CA ストアに、接続の確立に必要な署名者を確保させることができます。

以下のセクションで、SSL セキュリティーを強化する Build Forge システムのインターフェースを識別します。

**クライアント・インターフェース:** ユーザーはクライアント・インターフェースを使用して Build Forge システムにアクセスします。

#### **Build Forge に対する Web クライアント**

Web クライアントは Apache Web サーバーを使用して Build Forge にアクセスします。SSL が有効化され、セキュリティが有効になった Web ブラウザーを使用する場合、以下のインターフェースが使用されます。

##### **• Apache Web サーバーのポート 443**

Web クライアントは URL を使用して Build Forge にアクセスします。SSL が有効化された場合の URL は、以下のとおりです。

`https://host/`

*host* は Build Forge が稼働するホストです。Apache へのセキュア・アクセス用に、443 以外のポートをセットアップする場合、以下のポートも指定する必要があります。

`https://host:port/`

Web クライアントは、Apache Tomcat サーバーで稼働中の認証サーブレットにリダイレクトされます。

##### **• Apache Tomcat アプリケーション・サーバーのポート 8443**

認証サーブレットはログイン証明書を受諾し、ユーザーを認証します。サーブレットは資格情報を暗号化し、ネットワーク上に平文で表示されないようにします。

Apache Tomcat サブレットが使用するリスナー・ポートの構成は、構成ファイルを使用して管理します。このファイルは `<bfinstall>/Apache/tomcat/conf/server.xml` にあります。以下のコネクター構成を探します。

```
<Connector port="8443" maxHttpHeaderSize="8192" algorithm="IbmX509"
  maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
  enableLookups="false" disableUploadTimeout="true"
  acceptCount="100" scheme="https" secure="true"
  clientAuth="false" sslProtocol="SSL_TLS"
  keystoreFile="C:\BuildForge71.536\keystore\buildForgeKeyStore.p12"
  keystorePass="password"
  keystoreType="PKCS12"
  truststoreFile="C:\BuildForge71.536\keystore\buildForgeTrustStore.p12"
  truststorePass="password"
  truststoreType="PKCS12"/>
```

## Build Forge に対する API プログラム・クライアント

### • Apache Tomcat アプリケーション・サーバーのポート 49150

API クライアントは、サービス層コンポーネント、つまり Apache Tomcat アプリケーション・サーバー上で稼働しているアプリケーションを介して、Build Forge にアクセスします。API クライアントには、有効な `bfclient.conf` ファイルが必要です。

サービス層コンポーネントは、インバウンド通信に SSL 構成を使用します。これは、Build Forge コンソールの「管理」 > 「セキュリティ」 > 「SSL」で定義します。デフォルトでは「デフォルト JSSE インバウンド SSL」が使用されます。

**内部インターフェース:** Build Forge は、Web インターフェース・コンポーネント (Apache Web サーバーおよび PHP)、サービス層コンポーネント、およびエンジン・コンポーネントで構成されます。Web インターフェース・コンポーネントおよびエンジン・コンポーネントは、サービス層コンポーネントのクライアントです。また、API プログラム・クライアントもサービス層コンポーネントのクライアントです。

## サービス層インバウンド

### Apache Tomcat アプリケーション・サーバーのポート 49150

サービス層コンポーネントは、インバウンド通信に SSL 構成を使用します。これは、Build Forge コンソールの「管理」 > 「セキュリティ」 > 「SSL」で定義します。デフォルトでは「デフォルト JSSE インバウンド SSL」が使用されます。

## サービス層クライアント・アウトバウンド

**Apache Tomcat アプリケーション・サーバーのポート 49150** Web インターフェース・コンポーネント (PHP 経由) およびエンジン・コンポーネントは、いずれも、サービス層コンポーネントに対してアウトバウンド通信専用の SSL 構成を使用します。これは、Build Forge コンソールの「管理」 > 「セキュリティ」 > 「SSL」で定義します。デフォルトでは「デフォルト JSSE アウトバウンド SSL」が使用されます。

SSL ハンドシェイクを正常に実行できるように、クライアント・アウトバウンド構成用の SSL プロパティと、サービス層インバウンド構成用の SSL プロパティ

との間に互換性が必要です。「管理」 > 「セキュリティ」 > 「SSL」では、「タイプ」プロパティと「ハンドシェーク・プロトコル」プロパティが一致していなければなりません。

各 SSL 構成には、参照鍵ストア構成があります。

- 鍵ストア構成: プライベート証明書が入っている鍵ストアのプロパティを定義します。
- トラストストア構成: 信頼できる署名者が含まれる鍵ストアのプロパティを定義します。

構成は名前で指定されます。これは「管理」 > 「セキュリティ」 > 「SSL」で定義します。いくつかのデフォルト値が指定されています。

**外部インターフェース:** 外部インターフェースとは、Build Forge が外部システムと通信する際に使用するインターフェースのことです。

- Build Forge エンジン、エージェントと通信します。
- Build Forge サービス層コンポーネントはデータベースと通信します。

#### Build Forge エンジンからエージェントへの通信

このインターフェース用の SSL を有効にするには、以下の操作を行う必要があります。

- エージェントの構成。エージェント構成ファイルに変更を加えて、エージェント・ホストに証明書を置く必要があります。
- エージェントを使用するサーバー・リソースごとに SSL 通信を有効化。これはコンソールの「サーバー」パネルで実行します。

128 ページの『エージェント通信への SSL の有効化』を参照してください。

#### Build Forge サービス層コンポーネントからデータベースへの通信

このインターフェース用の SSL 構成は、データベースのデバイス・ドライバで定義されます。

### Apache サーバーでの SSL の有効化

Build Forge コンポーネントはデフォルトでセットアップされ、SSL が有効である場合には特定のポートとセキュリティ設定を使用します。

インストール中に、Web およびアプリケーション・サーバーのパネルで、「**セキュア HTTP を使用しますか?**」という質問がたずねられます。

「はい」と答えると、Apache サーバーの SSL は有効になります。その場合、このセクションは省略してください。

「いいえ」と答えた場合、Build Forge ホストにログオンして、以下のように Apache サーバーを構成することで、SSL を有効にできます。

- httpd.conf ファイル

<bfinstall>/Apache/conf/httpd.conf を、以下の設定を使用するように編集します。

```
Listen 0.0.0.0:443
ServerName localhost:443
```

- ssl.conf ファイル

<bfinstall>/Apache/conf/ssl/ssl.conf を、以下の設定を使用するように編集します。

```
<VirtualHost *:443>
ServerName localhost:443
SSLEngine on
```

ホストを選択する際には、完全修飾ドメイン名を指定できます。

## クライアントおよび内部接続用の SSL の有効化

Build Forge コンポーネントはデフォルトでセットアップされ、SSL が有効である場合には特定のポートとセキュリティ設定を使用します。

クライアントから Build Forge までの接続と、Build Forge の内部コンポーネント間での接続において SSL を有効にするには、以下の操作を指定の順序どおりに実行します。

- コンソールで、「管理」 > 「セキュリティ」と移動します。
- 「SSL 有効化」を「はい」に設定します。その他のプロパティがいくつか表示されますが、デフォルトのままにしておいてください。これらの設定は、後で必要に応じてカスタマイズできます。
- 「保存」をクリックします。SSL 構成が Build Forge データベースに保存されます。
- 「マスター BFClient.conf の更新」をクリックします。SSL 構成を使用して BFClient.conf ファイルが更新されます。この設定は、Build Forge が使用できるように、このファイル内で行っておく必要があります。
- Build Forge を停止して再始動します。これは必須です。

デフォルト設定でシステムが正常に動作することを確認したら、SSL に影響を与えるプロパティを編集します。

**注:** クライアントでは SSL を使用し、内部クライアント (Web インターフェース・コンポーネントおよびエンジン・コンポーネント) やサービス層コンポーネントの間では SSL を使用しない場合は、以下を実行します。

- BFClient.conf ファイルを手動で編集します。このファイルは <bfinstall> にあります。
- bf\_services\_preferred\_protocol 設定を tcp に変更します。  
bf\_services\_preferred\_protocol to tcp

Build Forge ホストが物理的に保護されている場合には、この構成によりセキュリティ・リスクが最低限に抑えられ、パフォーマンスが改善されます。

## ロック済みシステムでの TCP 通信の再有効化

SSL の構成に誤りがある場合、ユーザーはシステムによりロックアウトされます。

ロックされたシステムへのアクセス権限を得るには、以下を実行します。

- Build Forge を停止します。



- `<bfinstall>` ディレクトリー内の `bfclient.conf` ファイルをテキスト・エディターで開きます。
- プロトコルのプロパティーを以下のように変更します。  
`bf_services_preferred_protocol=tcp`
- Build Forge を開始します。

これで、ログインできるようになるはずです。

**注:** プロトコルを変更しても、デフォルトで有効になっているセキュアなログイン認証は無効になりません。ユーザーは、Build Forge にログイン資格情報を安全に通信できるセキュア接続にリダイレクトされます。

## API クライアント (Perl または Java) の SSL の有効化

Perl または Java で作成された API クライアント用の SSL 接続を構成するには、`bfclient.conf` を使用します。

### このタスクについて

API クライアント・プログラムとの SSL 接続を作成するには、`bfclient.conf` をセットアップします。このファイルには、SSL 構成プロパティーが含まれています。この手順を単純化するために、以下のステップでは、最初のステップで言及しているクライアント・ディレクトリーでクライアントを実行するものとします。

### 手順

1. スクリプトを実行する予定のクライアント・ディレクトリーを作成します。

Windows: `mkdir c:\%client`

UNIX または Linux: `mkdir /temp/client`

2. `bfclient.conf` をクライアント・ディレクトリーにコピーします。

Windows: `copy c:\%BuildForge%\bfclient.conf c:\%client`

UNIX または Linux: `cp /opt/BuildForge/Platform/bfclient.conf /temp/client`

3. クライアント・ディレクトリーに `keystore` サブディレクトリーを作成します。

Windows: `mkdir c:\%client%\keystore`

UNIX または Linux: `mkdir /temp/client/keystore`

4. BuildForge インストール済み環境からクライアントの `keystore` ディレクトリーに鍵ストアをコピーします。

Windows: `copy %BuildForge%\keystore\*.pem %client%\keystore`

UNIX または Linux: `cp /opt/BuildForge/Platform/keystore/*.pem /temp/client/keystore`

5. Perl の場合は、`bfclient.conf` の場所が検出されるように、以下の OS 環境変数を設定します。



Windows: `set BFCLIENT_CONF=c:\client\bfclient.conf`

UNIX または Linux: `export BFCLIENT_CONF="/temp/client/bfclient.conf"`

6. Java の場合は、以下のステップを実行します。

- a. `bfclient.conf` を使用した SSL 接続を作成するために、コード内で `APIClientConnection` ではなく `SecureAPIClientConnection` を使用します。
- b. スクリプトを実行する際に Java のコマンド行で以下のシステム・プロパティを設定します。

Windows: `-Dcom.buildforge.client.config=c:\client\bfclient.conf`

UNIX または Linux: `-Dcom.buildforge.client.config=/temp/client/bfclient.conf`

7. `bfclient.conf` を編集して、以下のプロパティを確認します。

- `bf_services_preferred_protocol=ssl`

値が `tcp` ではなく、`ssl` であることを確認してください。

- `bf_services_ssl_port=49150`

サービス層の SSL ポートが 49150 になっていることを確認してください。

- `bf_keystore_location=./keystore/buildForgeKey.pem`

いくつかの鍵ストアの場所があります。クライアント・ディレクトリー以外のディレクトリーでスクリプトを実行する場合は、完全修飾パスを使用するように各鍵ストアの場所を変更してください。

8. スクリプトが `bfclient.conf` を正しく使用しているかを確認するには、環境変数に以下のデバッグ・プロパティを設定します。スクリプトを実行すると、SSL 接続のプロパティに関する追加出力が表示されます。

Windows: `set BFDEBUG_SECURITY=1`

UNIX または Linux: `export BFDEBUG_SECURITY=1`

## エージェント通信への SSL の有効化

Build Forge<sup>®</sup> コンポーネントはデフォルトでセットアップされ、SSL が有効である場合には特定のポートとセキュリティ設定を使用します。

Build Forge<sup>®</sup> とエージェントとの間の SSL 通信を有効にするには、以下を実行する必要があります。

- UNIX コンピューターで、エージェントをホストするコンピューターに GCC ライブラリーがインストールされているようにします。
- 前提条件: クライアント通信および内部通信用に SSL を有効にします。126 ページの『クライアントおよび内部接続用の SSL の有効化』を参照してください。
- 各エージェントを構成します。このタスクには、以下が含まれます。
  - エージェント・ホストへの証明書の追加
  - エージェント用に `bfagent.conf` ファイルを編集

- コンソールで、エージェントに接続するサーバー定義ごとに SSL を有効にします。

## エージェントでの SSL の構成

1. エージェントが稼働中である場合は、停止してください。
2. エージェントのルート・インストール・ディレクトリーに、証明書の .PEM ファイルを配置します。

SSL を迅速にインプリメントおよびテストするには、Build Forge インストールから .PEM ファイルをコピーします。このファイルは `<bfinstall>/keystore` にあります。

**注:** エージェント・バージョン (bfagent -v) が 8.0 ではない、または 8.0 より低い場合、エージェントのインストール・フォルダーで古いエージェント `openssl` ユーティリティーを使用して、新しい `buildForgeKeyStore.p12` から `buildForgeKey.pem` を再生成し、古いエージェントを使用できるようにする必要があります。次に例を示します。

```
openssl pkcs12 -in buildForgeKeyStore.p12 -passin pass:<bfpasword>
-passout pass:<bfpasword> -out buildForgeKey.pem
```

.

SSL のベスト・プラクティスは、各エージェントに個別の証明書を使用することです。

- a. 以下のそれぞれについて、個別の鍵ストア・ファイル (.PEM) を作成します。
    - 秘密鍵 (key)
    - 秘密鍵用のパブリック証明書 (cert)
    - 信頼された署名者 (ca つまり認証局)
  - b. エージェントに (Build Forge® エンジンの証明書のコピーではなく) 固有の証明書を使用している場合は、そのエージェントの証明書を Build Forge® の認証局鍵ストア `<bfinstall>/keystore/buildForgeCA.pem` に追加します。複数のエンジン (冗長化エンジン) を実行している場合は、各エンジンの認証局鍵ストアに証明書を追加します。
  - c. kdb を System z で使用する場合は、gskkeyman で kdb を作成してください。
  - d. System Authorization Facility (SAF) 鍵リングを使用する場合は、RACF で鍵リングを作成し、その鍵リングに証明書を結び付けてください。
3. BFAgent.conf を編集します。以下の行は、ファイル内でコメント化されています。コメント接頭部を削除してください。

```
ssl_key_location buildForgeKey.pem
ssl_key_password password
ssl_cert_location buildForgeCert.pem
ssl_ca_location buildForgeCA.pem
ssl_protocol TLSv1
ssl_cipher_group ALL
```

`password` は、`buildForgeKey.pem` 鍵ストアのパスワードです。暗号化する場合は、148 ページの『`buildforge.conf` および `bfagent.conf` におけるパスワードの暗号化』を参照してください。

エージェントに接続する際にクライアント認証を必須にする場合は、以下の行のコメントを外してください。

```
ssl_client_authentication true
```

この設定では、エージェントの認証局鍵ストア `buildForgeCA.pem` に、エンジンの証明書を追加する必要があります。

固有の暗号を使用する場合は、以下の行のコメントを外して、使用する暗号リストを追加します。

```
ssl_cipher_override cipher_list
```

4. System z で稼働しているエージェントの場合は、`BFAgent.conf` を編集し、以下の行のコメントを外してください。

```
gsk_ssl_key_location <root/cert SAF>
gsk_ssl_key_location /etc/key.kdb
gsk_ssl_kdb_password <password>
gsk_keyring_label <certificate label>
gsk_ssl_protocol ALL
gsk_ssl_cipher_v2 6321
gsk_ssl_cipher_v3 0906030201
gsk_ssl_client_authentication true
gsk_password_encrypt false
```

`gsk_ssl_key_location` は、1 つだけ使用してください。SAF 鍵リングと `kdb` のいずれかを選択できます。SAF 鍵リングは、`<userid>/<keyring>` で指定します。

5. エージェントを開始します。コンソールからの接続をテストするには、エージェントが稼働中である必要があります。

## サーバー定義での SSL の有効化

コンソールは、サーバー定義を使用してエージェントに接続します。

SSL 有効化エージェントに接続する各サーバー定義について、以下を実行します。

- コンソールで、「サーバー」パネルに移動します。
- サーバー定義名をクリックします。
- サーバー定義の「詳細」タブで、以下を行います。
  - 「SSL 有効化」を「はい」に設定します。
- 「保存」をクリックします。
- 「接続のテスト」をクリックします。

## エージェントとの SSL 通信のトラブルシューティング

以下のチェックリストは、SSL 有効化の際に共通する問題について説明します。

- エージェント
  - エージェント SSL は構成されていませんが、セキュリティーおよびサーバー定義において SSL が有効化されています。

- エージェント用の証明書は、Build Forge<sup>®</sup> エンジンによって信頼されません。以下のようにして、エンジンの CA 鍵ストアにエージェントの証明書を追加する必要があります。

```
<bfinstall>/keystore/buildForgeCA.pem
```

- BFAgent.conf に指定された鍵ストアのパスワードが間違っています。
- BFAgent.conf にクライアント認証が指定されていますが、エンジンの証明書がエージェントの認証局 buildForgeCA.pem に追加されていません。
- Build Forge<sup>®</sup> コンソール
  - コンソールで SSL が有効化されました。保存しなかったか、保存前に「マスター BFClient.conf の更新」をクリックしています。または、保存後に「マスター BFClient.conf の更新」をクリックしていません。
- サーバー定義
  - SSL が正常に有効化されませんでした。「保存」をクリックしてから「接続のテスト」をクリックしてください。
- エンジンとエージェントの設定の一致
  - ハンドシェーク・プロトコルが一致していません。エンジンの SSL 構成とエージェントの SSL 構成の両方で、ハンドシェーク・プロトコルは同じ値 (TLSv1 または SSLv3) に設定する必要があります。デフォルト値は **TLSv1** です。
  - 暗号スイートが一致していません。エンジンの SSL 構成とエージェントの SSL 構成に指定された暗号スイートでは、共通する暗号が必要です。デフォルトの暗号スイート・グループは **ALL** です。

## デバッグ・メッセージの有効化

エンジンおよびエージェントでデバッグを有効にできます。デバッグを有効にすると、構成上の問題識別に役立つ、追加の明細出力が生成されます。

- エンジンのデバッグを有効にする手順は、以下のとおりです。
  1. SSL 有効化サーバー定義によって使用される環境で、以下の変数を追加します。

```
BFDEBUG_SECURITY=1
```

2. エンジンを停止し、再始動します。
  - Windows の場合は、エンジンをフォアグラウンドで開始してください。出力はコマンド・ウィンドウに表示されます。
  - UNIX または Linux の場合は、デバッグをオンにしてエンジンを開始してください。

```
cd <bfinstall>/rc
./buildforge start
```

エンジンからの出力は、<bfinstall>/log 内のエンジン・ログ・ファイルに記載されます。

- エージェントのデバッグを有効化する手順は、以下のとおりです。

1. エージェントを停止します。
2. BFAgent.conf に以下の行を追加します。

```
activity_log bfaagent.log
```

この例の場合、エージェントは出力を `bfagent.log` に書き込みます。出力先として、別のファイル名を指定できます。

注: エージェントがサービスとして稼働する場合は、絶対パスを指定してください。

3. エージェントを開始します。

### SSL 接続が正しく確立された場合のエンジンのデバッグ出力例

エンジンがエージェントへの接続に成功すると、以下のような出力が生成されます。

```
SSL_ca_file: ./keystore/buildForgeCA.pem
SSL_cert_file: ./keystore/buildForgeCert.pem
SSL_key_file: ./keystore/buildForgeKey.pem
SSL_verify_mode: 0x01
SSL_version: TLSv1
SSL_cipher_list: ALL
SSL_use_cert: 1
Making as SSL connection using socket IO::Socket::INET=GLOB(0x1e8f0f4).
SSL connection to agent.
DEBUG: .../IO/Socket/SSL.pm:1387: new ctx 80662848
DEBUG: .../IO/Socket/SSL.pm:880: dont start handshake: IO::Socket::SSL=GLOB(0x1e8f0f4)
DEBUG: .../IO/Socket/SSL.pm:284: ssl handshake not started
DEBUG: .../IO/Socket/SSL.pm:327: Net::SSL::connect -> 1
DEBUG: .../IO/Socket/SSL.pm:382: ssl handshake done
Socket is of type: ref(IO::Socket::SSL=GLOB(0x1e8f0f4))
ReadyLine: 202 HELLO TLS - BuildForge Agent v_VERSION_
.
Storing Agent Version [999.999.999.999-999-9999] for [08974C8E-6C3B-1014-972D-D9B2901D9F42]
cmd ping
username pbirk
encpass c1713f4a31af3f1300f7b2414a24559c4d6097e07310cf9c412e
go
Sending agent request...
```

### SSL 接続が正常に確立された場合のエージェントのデバッグ出力例

正常に稼働しているエージェントが SSL 接続を確立すると、以下のような出力が生成されます。

```
[2256] main.c : 409: === NEW AGENT ===
[2256] io.c : 264: In start SSL
[2256] io.c : 89: Key location: buildForgeKey.pem
[2256] bfpwdlocloader.c: 134: Looking for password locator: ssl_key_password_locator
[2256] bfpwdlocloader.c: 244: Looking for password for prop
ssl_key_password from bfagent.conf.
[2256] bfcryptloader.c : 202: Loading password encryption module.
[2256] bfcryptloader.c : 276: Password encryption property
password_encrypt_module is not configured.
[2256] bfcryptloader.c : 539: Password decoded.
[2256] io.c : 98: Cert location: buildForgeCert.pem
[2256] bfpwdlocloader.c: 134: Looking for password locator:
ssl_cert_password_locator
[2256] bfpwdlocloader.c: 244: Looking for password for prop
ssl_cert_password from bfagent.conf.
[2256] io.c : 153: Setting key password in default userdata.
[2256] io.c : 160: Getting private key from PEM.
[2256] io.c : 166: Checking private key from PEM.
[2256] io.c : 172: Getting CA store information.
[2256] bfpwdlocloader.c: 134: Looking for password locator:
ssl_ca_password_locator
[2256] bfpwdlocloader.c: 244: Looking for password for prop
ssl_ca_password from bfagent.conf.
```

```

[2256] io.c          : 178: CA location: buildForgeCert.pem
[2256] io.c          : 184: Checking the CA store.
[2256] io.c          : 230: Returning from init_CTX.
[2256] io.c          : 281: Calling SSL_new
[2256] io.c          : 294: Calling SSL_accept.
[2256] io.c          : 346: Cipher chosen: AES256-SHA
[2256] io.c          : 367: ssl_state = SS_CERTIFIED

```

## エージェントで不適切な鍵ストアのパスワードが使用された場合の出力例

エージェント・サイドの鍵ストアのパスワード構成が間違っていると、エンジンとエージェント両方の出力に示されます。

エンジンの出力 (excerpt):

```

SSL_use_cert: 1
Making an SSL connection using socket IO::Socket::INET=GLOBAL(0x1e8f0f4).
SSL connection to agent.
DEBUG: .../IO/Socket/SSL.pm:1387: new ctx 80662848
DEBUG: .../IO/Socket/SSL.pm:880: dont start handshake: IO::Socket::SSL=GLOBAL(0x1e8f0f4)
DEBUG: .../IO/Socket/SSL.pm:284: ssl handshake not started
DEBUG: .../IO/Socket/SSL.pm:327: Net::SSLeay::connect -> -1
DEBUG: .../IO/Socket/SSL.pm:1135: SSL connect attempt failed with unknown error
error:1408F10B:SSL routines:SSL3_GET_RECORD:wrong version number

DEBUG: .../IO/Socket/SSL.pm:333: fatal SSL error: SSL connect attempt failed with unknown error
error:1408F10B:SSL routines:SSL3_GET_RECORD:wrong version number
DEBUG: .../IO/Socket/SSL.pm:1422: free ctx 80662848 open=80662848 80566656
DEBUG: .../IO/Socket/SSL.pm:1425: OK free ctx 80662848

```

エージェントの出力 (excerpt):

```

[ 5272] io.c          : 98: Cert location: buildForgeCert.pem
[ 5272] bfpwdlocloader.c: 134: Looking for password locator: ssl_cert_password_locator
[ 5272] bfpwdlocloader.c: 244: Looking for password for prop ssl_cert_password from bfagent.conf.
[ 5272] io.c          : 153: Setting key password in default userdata.
[ 5272] io.c          : 160: Getting private key from PEM.
[ 5272] io.c          : 218: Failure reason: SSL_ERROR_BAD_KEY_FILE
[ 5272] io.c          : 221: OpenSSL error string: error:00000000:lib(0):func(0):reason(0)
[ 5272] io.c          : 281: Calling SSL_new
[ 5272] platform.c     :2693: platform_release_credentials
[ 5272] main.c         : 412: --- EXITING ---

```

## エージェント出力のエラー・コード

以下は、検出される可能性のあるその他いくつかのエラー・コードと、その理由を示すリストです。

- **SSL\_ERROR\_BAD\_CA**: buildForgeCA.pem に署名者をロード中に問題が発生しました。これは、署名者の追加方法に起因する、ファイル・フォーマット上の問題である可能性があります。
- **SSL\_ERROR\_BAD\_CERT**: buildForgeCert.pem に証明書をロード中に問題が発生しました。証明書の秘密鍵が一致しないか、PEM 内の秘密鍵が破損しています。
- **SSL\_ERROR\_BAD\_KEY\_FILE**: BFAgent.conf 内の ssl\_key\_password プロパティに指定された、buildForgeKey.pem パスワードで問題が発生しました。
- **SSL\_ERROR\_BAD\_KEY**: SSL\_CTX\_check\_private\_key から 1 以外の値が返されました。秘密鍵のフォーマットが無効であるか、秘密鍵が証明書と一致しません。



- **SSLERrorFIPSEnablement:** FIPS 有効化中にエラーが発生しました。このエラーは通常、FIPS のセルフ・チェック中に検出された問題が原因です。このエラーは内部エラーであると考えられます。
- **SSLERrorInvalidCipher:** 暗号の仕様が、OpenSSL で許可される内容に一致していません。BFAgent.conf の `ssl_cipher_group` プロパティまたは `ssl_cipher_override` プロパティに指定された暗号を確認してください。
- **SSLERrorNoCtx:** 新規 SSL CTX オブジェクトの作成中に問題が発生しました。このエラーは内部エラーであると考えられます。

## 証明書の管理

Build Forge によって使用される証明書と鍵ストアは、インストール後に変更できません。

インストール中に、使用する証明書 (独自の証明書または Build Forge で生成された証明書) と、鍵ストアのパスワードを指定できます。このセクションでは、以下の手順について説明します。

- Build Forge で必要となる OpenSSL および JSSE 鍵ストアに PEM 鍵ストアを変換する
- 鍵ストアのパスワードを変更し、その新規パスワードを使用するように Build Forge を変更する
- 新規自己署名証明書の作成
- クライアント証明書を使用したログイン
- 強力に暗号化された証明書に使用する無制限のポリシー・ファイルをダウンロードする

### PEM 鍵ストアの Build Forge 鍵ストアへの変換:

認証局から受け取った PEM 鍵ストアは、Build Forge で使用するための鍵ストアに変換できます。

### 始める前に

SDK 用に、無制限のポリシー・ファイルをダウンロードします。この前提条件は、使用する鍵サイズが制限付きポリシー・ファイルに対して大きすぎる場合にのみ該当します。ファイルは、[https://www14.software.ibm.com/webapp/iwm/web/reg/signup.do?source=jcesdk&lang=en\\_US&S\\_PKG=142ww](https://www14.software.ibm.com/webapp/iwm/web/reg/signup.do?source=jcesdk&lang=en_US&S_PKG=142ww) からダウンロードします。

注: IBM が提供する keytool ユーティリティーを使用する必要があります。

### このタスクについて

認証局から受け取った一連の PEM ファイルがある場合は、それらのファイルを使用して、Build Forge 用の一連の OpenSSL および JSSE 鍵ストアを作成する必要があります。

### 手順

1. パスに Build Forge ツール・ディレクトリーを含めます。
  - `<bfinstall>/openssl`
  - `<bfinstall>/ibmjdk/bin` (Windows の場合)



- `<bfinstall>server/ibmjdk/bin` (UNIX または Linux の場合)

UNIX および Linux の場合は、LD\_LIBRARY\_PATH に以下のディレクトリを組み込みます。

`<bfinstall>/openssl`

2. PEM ファイルを 1 つの PKCS12 鍵ストアに変換します。

以下のコマンドを使用します。

```
openssl pkcs12
  -export
  -name "buildforge"
  -out buildForgeKeyStore.p12
  -inkey <key.pem>
  -passin pass:<pempassword>
  -in <crt.pem>
  -password pass:<bfpasword>
```

3. 証明書が追加され、読み取り可能になっているかを確認します。

```
keytool -v
  -list
  -keystore buildForgeKeyStore.p12
  -storepass <bfpasword>
  -storetype pkcs12
```

無効な鍵サイズに関するエラーが発生した場合は、無制限のポリシー・ファイルをダウンロードします。このセクションの冒頭に記載されている指示に従ってください。

4. パブリック証明書をエクスポートします。

コマンド・ウィンドウで、`<bfinstall>/keystore` に移動し、以下のコマンドを実行します。

```
keytool -export
  -alias buildforge
  -file cert.der
  -keystore buildForgeKeyStore.p12
  -storepass <bfpasword>
  -storetype pkcs12
```

- 証明書は `cert.der` ファイルに保管されています。
  - インストール時に鍵ストアに指定したものと同一 `<bfpasword>` を使用してください。これを使用しない場合、構成を変更する必要があります。
5. トラストストアを作成し、パブリック証明書をインポートします。

コマンド・ウィンドウで、`<bfinstall>/keystore` に移動し、以下のコマンドを実行します。

```
keytool -import
  -noprompt -trustcacerts
  -alias buildforge
  -file cert.der
  -keystore buildForgeTrustStore.p12
  -storepass <bfpasword>
  -storetype pkcs12
```

6. パブリック・クライアント証明書を `buildForgeCert.pem` に入れます。

コマンド・ウィンドウで、`<bfinstall>/keystore` に移動し、以下のコマンドを実行します。

```
openssl pkcs12 -clcerts -nokeys
-in buildForgeKeyStore.p12
-passin: pass:<bfpasword>
-out buildForgeCert.pem
```

7. 証明書および鍵を buildForgeKey.pem に入れます。

コマンド・ウィンドウで、<bfinstall>/keystore に移動し、以下のコマンドを実行します。

```
openssl pkcs12
-in buildForgeKeyStore.p12
-passin pass:<bfpasword>
-passout pass:<bfpasword>
-out buildForgeKey.pem
```

8. PEM 認証局 buildForgeCA.pem を作成します。

- a. CA ルート証明書を <bfinstall>/keystore にダウンロードします。この証明書は CARootCert.crt と命名されます。これを PEM 鍵ストアに追加し、buildForgeTrustStore.p12 にインポートできるようにする必要があります。
- b. コマンド・ウィンドウで、<bfinstall>/keystore に移動し、以下のコマンドを実行します。

```
cat CARootCert.crt > buildForgeCA.pem
keytool -import -noprompt -v -trustcacerts
-alias "CA Root"
-file CARootCert.crt
-keystore buildForgeTrustStore.p12
-storepass <bfpasword>
-storetype pkcs12
```

## タスクの結果

Build Forge は、パスワードで保護された PEM 鍵ストア buildForgeKey.pem を使用します。Apache サーバーは、始動時にこのパスワードを求めるプロンプトを出します。

始動時にパスワードを求めるプロンプトが表示されないようにするには、パスワード保護されていない PEM 鍵ストアを生成し、Apache サーバーがその鍵ストアを使用するようにします。以下のコマンドはその例です。

```
openssl rsa -in buildForgeKey.pem
-passin pass:<password>
-out buildForgeKeyForApache.pem
```

必ず、この無保護の PEM 鍵ストアを、Build Forge を実行するプロセスの ID にアクセスする必要のある、すべてのユーザーに対して読み取り可能にしてください。

**鍵ストアのパスワードの変更:** インストール中に、鍵ストアのパスワードを指定することができます。パスワードを変更するには、以下を実行する必要があります。

1. 新規パスワードを使用するように Build Forge 鍵ストアを変更します。
2. 新規パスワードを使用するように Build Forge 構成を変更します。

**鍵ストアのパスワードの変更:**

## このタスクについて

Build Forge には、以下の 3 つのパスワード保護されたデフォルト鍵ストアがあり、いずれも Build Forge エンジンインストールしたホストの `<bfinstall>/keystore` にインストールされます。

- `buildForgeKey.pem` - OpenSSL によって使用され、パスワードを変更するために `openssl` ツールが必要です。
- `buildForgeKeyStore.p12` - JSSE によって使用され、パスワードを変更するために `ibmjdk` ツールが必要です。
- `buildForgeTrustStore.p12` - JSSE によって使用され、パスワードを変更するために `ibmjdk` ツールが必要です。

これらのツールは、Build Forge ソフトウェアに組み込まれています。

注: コマンド例では、見やすくするために改行を挿入しています。実際のコマンドでは、改行は挿入しないでください。1 つのストリングとして入力するか、行継続文字 (Windows の場合は `^`、UNIX または Linux の場合は `¥`) を使用してください。

**重要:** すべての鍵ストアに同じパスワードが使用されます。この例では、`newpassword` として示されます。

## 手順

1. Build Forge エンジンがインストールされているホストにログオンします。
2. パスにツール・ディレクトリーを含めます。
  - `<bfinstall>/openssl`
  - `<bfinstall>/ibmjdk/bin`
3. SSL を無効にします。コンソールで、「管理」 > 「セキュリティ」と移動します。「SSL 有効化」を「いいえ」に設定します。
4. 「保存」をクリックします。
5. 「マスター `BFClient.conf` の更新」をクリックします。
6. Build Forge エンジンを停止します。
7. 既存の鍵ストアをバックアップします。既存の Build Forge 鍵ストアを一時ディレクトリーにコピーします。バックアップした鍵ストアは、変更後のファイルが破損した場合に使用できます。
8. `buildForgeKey.pem` を変更します。ディレクトリー `<bfinstall>/keystore` で、以下のコマンドを実行します。

```
openssl rsa
-in buildForgeKey.pem
-passin pass:oldpassword
-out buildForgeKey.pem
-passout pass:newpassword -aes128
```
9. `buildForgeKeyStore.p12` を変更します。ディレクトリー `<bfinstall>/keystore` で、以下のコマンドを実行します。

```
keytool -storepasswd -all
-new newpassword
-keystore buildForgeKeyStore.p12
-storepass oldpassword
-storetype pkcs12
```

10. buildForgeTrustStore.p12 を変更します。ディレクトリー `<bfinstall>/keystore` で、以下のコマンドを実行します。

```
keytool -storepasswd -all
-new newpassword
-keystore buildForgeTrustStore.p12
-storepass oldpassword
-storetype pkcs12
```

## タスクの結果

パスワードを変更したら、新規パスワードを使用するように Build Forge 構成を変更する必要があります。

### 新規パスワードを使用するように Build Forge 構成を変更:

変更した鍵ストアのパスワードを使用するには、Build Forge 構成を変更する必要があります。

## 始める前に

前提条件:

- SSL をオフにして、Build Forge エンジンを停止し、鍵ストアのパスワードを変更してから、Build Forge エンジンを開始していません。

## このタスクについて

Apache Tomcat アプリケーション・サーバーで、`server.xml` 構成ファイルに鍵ストアのパスワードが入っています。パスワードは平文として保管されています。この設定では、Apache Tomcat はエンコードまたは暗号化されたパスワードをサポートしません。以下の手順で、`server.xml` および Build Forge コンソールのセキュリティー・プロパティーを変更します。

## 手順

1. Tomcat 構成に新規パスワードを入力します。 `<bfinstall>/Apache/tomcat/conf/server.xml` を編集します。以下の SSL の Connector ステートメントは、コメント `<!-- Define a SSL HTTP/1.1 Connector on port 8443 -->` のすぐ下にあります。

```
<Connector port="8443" maxHttpHeaderSize="8192" algorithm="IbmX509"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" disableUploadTimeout="true"
acceptCount="100" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"
keystoreFile="C:\Program Files\IBM\Build
Forge\keystore\buildForgeTrustStore.p12"
keystorePass="newpassword"
keystoreType="PKCS12"
truststoreFile="C:\Program Files\IBM\Build
Forge\keystore\buildForgeTrustStore.p12"
truststorePass="newpassword"
truststoreType="PKCS12"/>
```

2. Build Forge を開始します。
3. コンソールにログインします。 root あるいはセキュリティー・アクセス・ロールを持つログイン名を使用します。
4. 鍵ストアに新規パスワードを入力します。「管理」 > 「セキュリティー」 > 「鍵ストア」の「パスワード」プロパティで、新規パスワード を使用するよう以下項目を編集します。
  - デフォルト JSSE 鍵ストア
  - デフォルト JSSE トラストストア
  - デフォルト OpenSSL 鍵ストア
5. SSL を有効にします。
  - a. コンソールで、「管理」 > 「セキュリティー」と移動します。
  - b. 「SSL 有効化」を「はい」に設定します。
  - c. 「保存」をクリックします。
6. 変更内容を BFClient.conf にエクスポートします。「マスター BFClient.conf の更新」をクリックします。
7. Build Forge エンジンを開始します。

#### 新規自己署名証明書の作成:

提供されているツールを使用して、新しい自己署名証明書を作成します。

#### 始める前に

インストール時に鍵ストアに対して指定されたパスワードが必要です。パスワードが分からない場合は、bfinstall/Apache/tomcat/conf/server.xml を調べてください。SSL/HTTP コネクターの keystorePass 属性に、パスワードが含まれています。

#### このタスクについて

この手順では、Rational® Build Forge® のインストール時に自動生成された証明書を置き換える方法について説明します。以下のプロパティが指定された証明書を作成します。

- 鍵ストア: buildForgeKeyStore.p12
- 有効期限: 15 年 (5475 日として設定)
- サブジェクト DN: CN=hostname (hostname は完全修飾ホスト名)。

証明書を作成するには、openssl ツールおよび ibmjdk ツールを使用します。これらのツールは、Rational® Build Forge® ソフトウェアに組み込まれています。

以下の 5 つの鍵ストアが必要です。

- buildForgeKeyStore.p12 - 証明書および鍵のコンテナとなる鍵ストア
- buildForgeTrustStore.p12 - 証明書および鍵のコンテナとなるトラストストア
- buildForgeKey.pem - PEM 鍵ストア
- buildForgeCert.pem - パブリック証明書
- buildForgeCA.pem - PEM 認証局 (CA)

注: コマンド例では、見やすくするために改行を挿入しています。実際のコマンドでは、改行は挿入しないでください。1つのストリングとして入力するか、行継続文字 (Windows の場合は ^、UNIX または Linux の場合は ¥) を使用してください。

**重要:** すべての鍵ストアに同じパスワードが使用されます。この例では、*password* として示されます。

#### 手順

1. Rational® Build Forge® サーバーがインストールされているホストにログオンします。
2. 次のリストにあるようにファイル・パスにツールのディレクトリーを追加します。
  - `<bfinstall>/openssl`
  - `<bfinstall>/ibmjdk/bin`
3. openssl ディレクトリーを LD\_LIBRARY\_PATH ファイル・パスに追加します。例えば、`<bfinstall>/openssl` を追加します。
4. 鍵ストア buildForgeKeyStore.p12、証明書、および公開鍵と秘密鍵のペアを作成します。
  - a. 一時ディレクトリーで、次のスクリプトを使用して鍵ストアを作成します。

```
keytool -genkey -alias buildforge
-keyalg RSA -keysize 1024 -validity 5475 -dname "CN=hostname"
-keystore buildForgeKeyStore.p12
-storepass password
-storetype pkcs12
```
  - b. 鍵ストア・ファイル buildForgeKeyStore.p12 を `<bfinstall>/keystore` にコピーします。既存のファイルは上書きされます。
5. `<bfinstall>/keystore` ディレクトリーで、次のコマンドを入力してパブリック証明書をエクスポートします。

```
keytool -export -alias buildforge
-file cert.der -keystore buildForgeKeyStore.p12
-storepass password
-storetype pkcs12
```
6. トラストストアを作成します。
  - a. 一時ディレクトリーで、keytool を実行してトラストストアを作成します。

```
cd /temp
keytool -import -noprompt -trustcacerts -alias buildforge
-file cert.der -keystore buildForgeTrustStore.p12
-storepass password
-storetype pkcs12
```
  - b. トラストストア・ファイル buildForgeTrustStore.p12 を `<bfinstall>/keystore` にコピーします。既存のファイルは上書きされます。
7. パブリック・クライアント証明書を buildForgeCert.pem に入れます。ディレクトリー `<bfinstall>/keystore` で、以下のコマンドを実行します。

```
openssl pkcs12 -clcerts -nokeys
-in buildForgeKeyStore.p12 -passin pass:password
-out buildForgeCert.pem
```
8. 証明書および鍵を buildForgeKey.pem に入れます。ディレクトリー `<bfinstall>/keystore` で、以下のコマンドを実行します。

```
openssl pkcs12
-in buildForgeKeyStore.p12 -passin pass:password
-passout pass:password -out buildForgeKey.pem
```

9. ディレクトリー <bfinstall>/keystore で、次のコマンドを実行して、PEM 認証局 buildForgeCA.pem を作成します。 次のコマンドは、buildForgeKey.pem のコピーを作成します。

- Windows:  
copy buildForgeKey.pem buildForgeCA.pem
- UNIX および Linux  
cat buildForgeKey.pem > buildForgeCA.pem

## 次のタスク

buildForgeKey.pem は、パスワード保護された PEM 鍵ストアです。Apache サーバーから始動時にパスワードを求めるプロンプトが出されます。このようなパスワードを求めるプロンプトが出されないようにする場合は、パスワード保護されていない PEM 鍵ストアを Apache サーバー用に生成します。

秘密鍵からパスワードを削除するには、次のコマンドを入力します。 Build Forge を実行するプロセスの ID にアクセスする必要のあるユーザーが buildForgeKeyForApache.pem ファイルを読み取れるようにしてください。

```
openssl rsa -in buildForgeKey.pem -passin pass:password
-out buildForgeKeyForApache.pem
```

## クライアント証明書を使用したログイン:

クライアント証明書を使用したログイン機能の構成には、2 つの選択肢があります。

### WebSphere Application Server を使用して Build Forge をホストする場合のログイン:

#### このタスクについて

WebSphere Application Server を使用して Build Forge をホストする場合、変更を行った上で使用可能なオプションがあります。この方法を使用する場合には、WebSphere が認証を実行した後に認証済みのプリンシパルを取得する WebSphereSSOInterceptor が必要です。

この方法を使用する場合は、以下の手順で変更を行ってください。

#### 手順

1. WebSphere Application Server のコンテナ管理認証が jas への要求を認証できるように、Build Forge の WAR ファイルを保護します。 この保護をセットアップする方法については、116 ページの『カスタム・インターセプターを使用した WebSphere Application Server セキュリティーとの統合』を参照してください。
2. jas Web アプリケーション用にクライアント証明書認証のマッピングをサポートするように WebSphere Application Server を構成します。 このサポートは、以下の項目を構成することによってセットアップします。
  - クライアント証明書用の SSL



- クライアント証明書用の jas Web アプリケーション
- クライアント証明書用の Web サーバー
- クライアント証明書用の LDAP サーバーのマッピング

WebSphere Application Server を構成する際のガイドとなる WebSphere Portal でクライアント証明書認証を構成する方法については、<http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/index.jsp?topic=/com.ibm.wp.ent.doc/wpf/certauth.html> を参照してください。ステップ 1 から 3 に従ってください。この手順では、116 ページの『カスタム・インターセプターを使用した WebSphere Application Server セキュリティーとの統合』で変更する web.xml ファイルについて言及しています。\${WAS\_INSTALL\_ROOT}/profiles/\${PROFILE\_NAME}/installedApps/\${CELL\_NAME}/jas\_war.ear/jas.war/WEB-INF/web.xml にあるファイルに変更を行います。

### カスタム Build Forge SSO インターセプターに基づくログイン: このタスクについて

この方法を使用する場合は、105 ページの『シングル・サインオン・フレームワークについて』で説明しているとおり、カスタム SSO インターセプターを追加する必要があります。

#### 手順

1. 以下のように、カスタム SSO インターセプターが要求属性から X509 証明書を受け取るようにします。

```
X509Certificate[] certs =
(X509Certificate[])request.getAttribute("javax.net.ssl.peer_certificates");
```

2. カスタム SSO インターセプターがサブジェクト DN の一部を LDAP 属性にマップするようにします。通常は、証明書の共通名 (CN) を、LDAP ディレクトリ内のユーザー ID にマップします。Build Forge SSO インターセプターの authenticateRequest メソッドで結果を返します。

```
responseAttributes.setStatus(HttpServletResponse.SC_OK);
principal = mapCert(certs); /* Custom method to map from certificate to
                             Build Forge LDAP principal. */
result = new Result(Result.UseridOnlyOID, domain, principal); /* Specify
the domain name of the LDAP server the principal exists in. */
return result;
```

### 無制限のポリシー・ファイルのダウンロード:

証明書が強力に暗号化されている場合、JCE 用の無制限のポリシー・ファイルをダウンロードする必要があります。

エクスポートおよびインポートの規定のため、Installation Manager および Build Forge に付属しているデフォルトの JCE 管轄権ポリシー・ファイルでは、強力ではあるが制限のある暗号方式しか使用できません。エクスポートに許可されている暗号化の強度より強力な暗号化が行われている証明書を使用する場合は、無制限の JCE ポリシー・ファイルをダウンロードして、Installation Manager および Rational Build Forge の JVM を更新する必要があります。

インストール中に、使用する証明書を指定した際に次のエラー・メッセージが表示される場合は、JCE ポリシー・ファイルを更新しなければならないことがあります。

この証明書を検査できません。証明書なしでインストールし、インストール後に追加することを試みてください。

**重要:** Software Development Kit (SDK) の更新が含まれる Build Forge のアップグレードを行うと、ポリシー・ファイルが上書きされる場合があります。この場合は、アップグレードを行った後に、無制限のポリシー・ファイルを SDK に再度適用してください。

**重要:** ご使用になっている国によっては、暗号ソフトウェアの輸入、所持、使用、または他国への再輸出が規制されている可能性があります。制限されていないポリシー・ファイルをダウンロードまたは使用する前に、ご使用になっている国における、暗号化ソフトウェアのインポート、所有、使用、再エクスポートに関する法律、規則、およびポリシーを確認して、その行為が許可されるかどうかを判断する必要があります。

1. バージョン 1.4.2+ の SDK 用の無制限の JCE ポリシー・ファイルをダウンロードします。
2. ファイルを unzip します。
3. unzip したファイル local\_policy.jar を、適切な Build Forge インストール・ディレクトリーにコピーします。既存のファイルを上書きします。ご使用のオペレーティング・システムに対応するディレクトリーを使用してください。

- Windows システム

```
<bfinstall>%ibmjdk%jre%lib%security
```

- UNIX システムおよび Linux システム

```
<bfinstall>/Platform/ibmjdk/jre/lib/security
```

4. Build Forge の今後のインストールで証明書が確認されるように、Installation Manager を更新します。unzip したファイル local\_policy.jar を、Installation Manager とともにインストールされる IBM JRE にコピーします。既存のファイルを上書きします。

- Windows システム

```
<IInstall>%eclipse%jre_version%jre%lib%security
```

- UNIX システムおよび Linux システム

```
<IInstall>/eclipse/jre_version/jre/lib/security
```

## パスワード暗号化の有効化

パスワード暗号化を使用するように Build Forge システムを構成すると、システムのセキュリティが向上します。

また、システムへの無許可アクセスを防ぐため、物理的セキュリティを強化することも重要です。

注: WebSphere コンポーネントと統合する場合は、パスワード暗号化を構成する前に、パスワード暗号化のための前提条件を満たしているかを確認してください。

- WebSphere Application Server と統合する場合は、629 ページの『Apache Tomcat の代わりに WebSphere Application Server を使用』を参照してください。
- IBM HTTP Server (IHS) と統合する場合は、635 ページの『Apache HTTP Server の代わりに IBM HTTP Server を使用』を参照してください。

## Build Forge でのパスワード・セキュリティについて

Build Forge システムでは、エンコードしたパスワードをデフォルトで使用しますが、追加のセキュリティとして暗号化パスワードを使用できます。

パスワード暗号化を有効にすると、対称鍵パスワード方式が使用されます。パスワードを使用しているクライアントと、アクセスを受けるサービスの両方で同じ鍵を使用する必要があります。

- Build Forge エンジンおよび Build Forge エージェント
- Build Forge サービス層、および Build Forge が使用するデータベース

Build Forge システムでは、鍵は `bfpwcrypt.conf` ファイルに入れられます。このファイルは、Build Forge のインストール・ディレクトリー (エンジンの場合) およびエージェントに置かれます。

パスワード暗号化では、対称鍵を使用します。共通のデータベース・パスワードを暗号化解除しなければならないすべてのシステムは、同じ鍵を必要とします。また、エンジンから暗号化パスワードを受け取るすべてのエージェントは、そのエンジンの鍵を必要とします。複数のエンジンが稼働している (冗長化構成) 場合、エージェントは各エンジンの鍵を必要とします。

あるホストに 1 つの Build Forge 管理コンソールを、別のホストに 1 つのエージェントをインストールした単純な環境で、パスワード暗号化を有効にするために必要な手順は以下のとおりです。

- Build Forge コンソール (「管理」 > 「セキュリティ」) でパスワード暗号化を有効にします。
- 現行の鍵をファイルにエクスポートします。(これはエージェントによって使用される鍵であり、他のエンジンでも使用されます。)
- エージェント用に新規の鍵を生成します。その鍵をファイルにエクスポートします。(この鍵は、エージェントによって鍵ストア・パスワードの暗号化に使用されます。)
- エージェントの `bfpwcrypt.conf` を、両方の鍵を使用して更新します。新規の鍵を最後に配置します。
- Build Forge コンソールで、そのエージェントを使用するすべてのサーバー定義に対してパスワード暗号化を有効にします。(「サーバー」パネル)
- Build Forge ホストで `bfpwcrypt` ユーティリティを使用して、Build Forge がデータベースへのアクセスに使用するパスワードを暗号化します。  
`buildforge.conf` で、現行パスワード (エンコード済み) を暗号化したパスワードに置き換えます。
- サービス層の `buildforge.conf` のコピーを更新します。99 ページの『Build Forge 構成ファイル (buildforge.conf)』を参照してください。

## 管理コンソールのパスワード暗号化の有効化

管理コンソールでのパスワード暗号化の有効化は、その他すべてのコンポーネントでパスワード暗号化を有効化するための前提条件です。

- 冗長化管理コンソールでは、`bfpwcrypt.conf` ファイルに入れるパスワード鍵を最初の管理コンソールからエクスポートする必要があります。すべての管理コンソールで同じ鍵を使用しなければなりません。通常、データベース・パスワードの暗号化にもその鍵を使用する必要があります。
- エンジンが暗号化パスワードを送信している場合、エージェントでパスワード鍵のエクスポートが必要となります。エンジンとエージェントの両方で、同じ鍵を使用するようにしてください。管理コンソールからエクスポートした鍵を使用することで、暗号化されたサーバー認証パスワードを暗号化解除できます。

エージェントで、暗号化パスワードを受信していないが、`BFAgent.conf` ファイルで使用する暗号化パスワードを生成する必要がある場合は、エージェントごとに異なるパスワード鍵を使用しなければなりません。各鍵を、管理コンソールから個別に生成してください。

管理コンソールでパスワード暗号化を有効化する手順は、以下のとおりです。

1. コンソールで、「管理」 > 「セキュリティ」と移動します。
2. 「パスワード暗号化の有効化」を「はい」に設定します。
3. 「保存」をクリックします。この手順により、構成が Build Forge データベースに保存されます。
4. 「マスター `BFClient.conf` の更新」をクリックします。この手順により、構成が Build Forge の `bfclient.conf` ファイルに保存されます。
5. Build Forge を再始動します。実行中の Build Forge プロセスが `bfclient.conf` ファイルの新規設定を使用できるようにするには、この手順が必要です。

構成内で「パスワード暗号化の有効化」が「はい」に指定されている場合、以下を実行できます。

- パスワード鍵のファイルへのエクスポート
- 新規パスワード鍵の生成
- `bfpwencrypt` コマンドおよび `bfagent` コマンドを実行して、コンソールおよびエージェントの構成ファイルに含める暗号化パスワードを作成。

## パスワードの鍵の管理

Build Forge システムでは、エンコードしたパスワードをデフォルトで使用しますが、追加のセキュリティとして暗号化パスワードを使用できます。

パスワード暗号化の鍵ファイル:

`bfpwcrypt.conf` ファイルには、パスワード暗号化の構成プロパティが含まれます。

Build Forge は、初回始動時に `bfclient.conf` ファイルと同じ場所に `bfpwcrypt.conf` ファイルを自動で生成します。

- Windows: `<bfinstall>`
- UNIX または Linux: `<bfinstall>/Platform`

**重要:** このファイルの名前は変更しないでください。名前は常に `bfpwcrypt.conf` でなければなりません。

このファイルには以下のようなプロパティーが含まれます。

#### **bfpwcrypt\_key\_alias**

暗号鍵の別名。別名は、この鍵を使用して暗号化されたすべてのパスワードの一部になります。システムは、この別名を使用して、使用する鍵を判別します。このプロパティーの定義は複数ある場合があります (鍵ごとに 1 つずつ)。最新の定義が、パスワードの暗号化に使用されます。その他の定義はすべて、暗号化されたパスワードを読み取る際に、その暗号化解除に使用されます。暗号化されたパスワードは、構成ファイル、データベース、またはエージェントとエンジン間の通信の中で見つかります。

#### **bfpwcrypt\_key**

128 ビット AES 暗号化を使用して暗号化される暗号化マスター・キー。この鍵を使用してパスワードを暗号化します。

#### **bfpwcrypt\_key\_password**

`bfpwcrypt_key` の暗号化解除に必要なパスワード。このパスワードはエンコードされています。

`bfpwcrypt.conf` ファイルの例:

```
***** Password Encryption Configuration Properties *****
bf_pwcrypt_key=MKuoiwD+MsWBFgl/2xeG0TEtpY+hAzXQu21fBcofM0M=
bf_pwcrypt_key_alias=8a679d430c401000b55e00007d1a7d1a
bf_pwcrypt_key_password=Tq0eDXc4G/bdaWeatKTYUx6Sw4S3i6wX
# Creation date=Thu Nov 20 03:44:48 CST 2008
# Origination host=myhost.mycompany.com
```

#### **パスワード鍵のエクスポート:**

別の場所に配置できるように、Build Forge コンソールからパスワード鍵をエクスポートします。

Build Forge エンジンで使用されるパスワード鍵をエクスポートして、以下の場所に配置します。

- 同じデータベースにアクセスするその他の Build Forge コンソール
- パスワード暗号化が有効になっているサーバー定義に指定されているエージェントをホストするコンピューター

パスワード鍵をエクスポートする手順は、以下のとおりです。

1. コンソールで、「管理」 > 「セキュリティ」と移動します。

注: 「パスワード暗号化の有効化」が既に「はい」に設定済みで、この設定が `BFCClient.conf` のマスター・ファイルで保存および更新済みになっている必要があります。

2. 「鍵ファイルのエクスポート」をクリックします。ファイルの保存場所をたずねられます。
3. 場所を指定し、「保存」をクリックします。

パスワードを暗号化解除する鍵を必要とするホスト (他の冗長化管理コンソール、エージェント) か、パスワードを暗号化する鍵を必要とするホスト (エージェント)

にファイルを置きます。ファイルの内容 (3 つのプロパティー設定すべて) をホスト上の `bfpwcrypt.conf` ファイルに追加します。下の表で、いつ鍵が使用されるかが、内容の配置によってどのように決定されるかについて説明します。

bfpwcrypt.conf ファイル内の配置	パスワード鍵が使用される状況
最後	<ul style="list-style-type: none"><li>その鍵の別名が含まれているパスワードの暗号化解除時</li><li>新規パスワードの暗号化時</li></ul>
それ以外の場所	<ul style="list-style-type: none"><li>その鍵の別名が含まれているパスワードの暗号化解除時</li></ul>

注: 新規の鍵を生成すると、追加の鍵が `bfpwcrypt.conf` ファイルの末尾に配置されます。鍵をエクスポートすると、最新の鍵だけがエクスポートされます。

**新規パスワード鍵の生成:** 管理コンソールから新規の鍵を生成して、システム全体の鍵を変更できます。この操作は、システム・セキュリティを確保するために、定期的に行う必要があります。新規の鍵は、すべてのエンジンおよびエージェントで更新する必要があります。

新規パスワードの鍵を生成すると、`bfpwcrypt.conf` ファイルに新規の鍵が追加されます。

- ユーザーまたはコンソールのパスワードが古い鍵を使用して生成または保存されている場合でも、パスワードは引き続き機能します。
- **重要:** 以前の鍵がエクスポートされて、エージェントの `bfpwcrypt.conf` ファイルに取り込み済みである場合は、新しく生成した鍵を追加するまで、そのエージェントとの通信は失敗します。
- Build Forge に保存される、または `bfpwencrypt` で生成されるすべての新規パスワードは、この新規の鍵を使用します。
- 鍵ファイルをエクスポートすると、最新の鍵のみがエクスポートされます。

新規パスワード鍵を生成する手順は、以下のとおりです。

1. コンソールで、「管理」 > 「セキュリティ」と移動します。

注: 「パスワード暗号化の有効化」が既に「はい」に設定済みで、この設定が `BFCClient.conf` のマスター・ファイルで保存および更新済みになっている必要があります。

2. 「新規鍵の生成 (Generate New Key)」をクリックします。確認が求められます。
3. 「はい」をクリックします。

## エージェントのパスワード暗号化の有効化

エージェントのパスワード暗号化は、エージェントの構成ファイルで有効化します。

エージェントでパスワード暗号化を有効化する手順は、以下のとおりです。

1. エージェントが実行中であれば停止します。
2. エージェントのインストール先ディレクトリーに移動します。



3. 以下のように `bfagent.conf` を編集し、暗号化設定をオンにします。

```
password_encrypt_module ./bfcrypt.dll;./bfpwcrypt.conf
```

エージェントがインストール・ディレクトリー以外のディレクトリーで起動されている場合、直接そのファイルを参照するようにこのパスを変更してください。

4. 管理コンソール暗号鍵を入手します。その鍵をファイルにエクスポートします。暗号化されたサーバー認証パスワードをエージェントが暗号化解除する際に、この鍵が必要となります。この鍵は、ローカル鍵ストアのパスワードを暗号化するためにも使用されます。
5. この鍵をエージェントの `bfpwcrypt.conf` ファイルに配置します。エージェント用に生成した鍵を、ファイルの最後に置きます。パスワードを手動で暗号化するには、ファイル内の最後の項目が使用されます。
6. エージェントを開始します。
7. 管理コンソールで「サーバー」に移動します。このエージェントを使用するすべてのサーバー定義について、「パスワード暗号化構成」プロパティを「有効」に設定します。
8. 「接続のテスト」をクリックして、接続が暗号化されたパスワードで動作することを確認します。

## buildforge.conf および bfagent.conf におけるパスワードの暗号化

エクスポートしたパスワード鍵を使用して、`buildforge.conf` および `bfagent.conf` で使用する暗号化パスワードを作成します。

`buildforge.conf` ファイルには、Build Forge がデータベースへのアクセスに使用するユーザー名とパスワード (`db_password`) が含まれます。このパスワードは通常エンコードされていますが、暗号化することが可能です。管理コンソール用にパスワードを暗号化する手順は、以下のとおりです。

1. 管理コンソールのルート・ディレクトリーに移動します。

- Windows:<*bfinstall*>
- UNIX または Linux: <*bfinstall*>/Platform

2. 以下のコマンドを実行します。

```
bfpwncrypt -e password
```

*password* には、暗号化する平文のパスワードを指定します。

暗号化したパスワードは `stdout` に送信されます。

`bfagent.conf` ファイルには、エージェントが鍵ストアへのアクセスに使用するパスワード鍵 (`ssl_key_password`) が含まれます。このパスワードは通常平文ですが、暗号化することが可能です。エージェント用にパスワードを暗号化する手順は、以下のとおりです。

1. エージェントのルート・ディレクトリーに移動します。
2. 以下のコマンドを実行します。

```
bfagent -e password
```

*password* には、暗号化する平文のパスワードを指定します。暗号化したパスワードは `stdout` に送信されます。



注: AIX を使用しており、ご使用の GCC ライブラリーが `/lib` または `/usr/lib` がない場合、「モジュール `/usr/local/bin/bfcrypt.dll` をロードできない」ことを示すエラーを受け取る可能性があります。この問題には、`inetd.conf` を更新することで対応できます。

エラーを修正するには、次の手順を実行します。

- a. 次の行を見つけます。

```
bfagent stream tcp nowait root /usr/local/bin/bfagent bfagent
```

- b. この行を以下の行に変更します。

```
bfagent stream tcp nowait root /usr/bin/env env LIBPATH=path /usr/local/bin/bfagent
```

`path` は GCC ライブラリーの場所です。

- c. `refresh -s inetd` コマンドを使用して `inetd.conf` を再ロードします。

暗号化パスワードは、ストリング `bfcrypt:` と、中括弧で囲まれたパスワード鍵の別名で始まり、その後ろにパスワードが続きます。このパスワードは暗号化 (AES 128 ビット) されてから、エンコード (Base63) されます。以下は、エンコードされたパスワードと暗号化されたパスワードの例です。

エンコードされたパスワード:

```
dd8b42eed5cc051500f5bffe2b82b1aa6a67baee028a85d0cefa
```

暗号化されたパスワード:

```
{bfcrypt:7427ab360c4010008f9d000049664966}drAIT1zLDGX/xRcvw65+B8aFpTqvmAdbmnh6FpwkHjU=
```

## パスワード暗号化にともなう問題のデバッグ

暗号化パスワードに問題がある場合のデバッグ方法については、以降のセクションで説明します。

### コンソールでのパスワード暗号化に関する問題のデバッグ:

サービス層、Web インターフェース、およびエンジンは、同じ鍵ファイルを共有します。

これら 3 つのコンポーネントは、すべてが同じホスト上にインストールされると、同じ鍵ファイルを使用します。

- Windows:

```
<bfinstall>/bfpwcrypt.conf
```

- UNIX または Linux:

```
<bfinstall>/Platform/bfpwcrypt.conf
```

パスワード暗号化を有効にした後で問題が発生する場合、以下を確認してください。

- パスワード暗号化を有効にした後で、Build Forge<sup>®</sup> を再始動したことを確認します。すべてのプロセス (Apache、Apache Tomcat、エンジン) の停止および再始動が適切に行われたことを確認してください。
- 冗長コンソール: 同じデータベースを使用する管理コンソールのインストールが複数ある場合、それらはすべての同じ `bfpwcrypt.conf` ファイルを使用する必要があります。最も安全な方法は、このファイルをネットワーク経由ではなく手動で配布することです。

- サーバー定義: コンソールで「**接続のテスト**」が失敗した場合、鍵がエクスポートされて `bfagent.conf` ファイルに正しく配置されたことを確認します。問題がパスワードにあることを確認するには、パスワード暗号化を無効にしてから「**接続のテスト**」を試みます。
- ログイン: パスワード暗号化を有効化した後でログインできない場合は、Build Forge® が `bfclient.conf` と `buildforge.conf` の両方で、`bfencrypt.conf` の正しい鍵を使用していることを確認してください。 `<bfinstall>` ディレクトリーと、サービス層にあるディレクトリーのコピーで、`buildforge.conf` を更新する必要があります。99 ページの『Build Forge 構成ファイル (`buildforge.conf`)』を参照してください。

これらすべての項目を確認しても問題が解決しない場合は、トレースを有効にして出力ログを調べてみてください。

- Web インターフェース (UI): 環境変数 `BFDEBUG_SECURITY=1` を設定します。

Web インターフェース: 出力がファイル内に示されます。

– Windows:

`<bfinstall>/Apache/logs/php_error.log`

– UNIX または Linux:

`<bfinstall>/server/apache/logs/php_error.log`

- エンジン: エンジンをフォアグラウンドで始動します。インストール・ディレクトリーで、`bfengine -d` を実行します。UNIX または Linux の場合、`bfengine -d 2>&1 | tee out.txt` を使用して、これをファイルにパイピングできます。Windows の場合、tee ユーティリティーを保持していれば、同様の処理を実行できます。

- サービス: 以下を実行します。

1. Build Forge を停止します。

2. ログ・ファイルをエディターで開きます。

– Windows:

`<bfinstall>/Apache/tomcat/common/classes/logging.properties`

– UNIX または Linux:

`<bfinstall>/server/apache/tomcat/common/classes/logging.properties`

3. ファイルの末尾に以下の行を追加します。

`com.buildforge.services.common.security.level=ALL`

4. Build Forge® を開始します。

5. 出力を検査します。

– Windows:

`<bfinstall>/Apache/tomcat/logs/catalina.out`

– UNIX または Linux:

`<bfinstall>/server/apache/tomcat/logs/catalina.out`

### エージェントにおけるパスワード暗号化の問題のデバッグ:

エージェント通信のデバッグには、エージェントとエンジンのコンポーネントが関連しています。

パスワード暗号化を有効にした後で問題が発生する場合、以下を確認してください。

- `bffagent.conf` を確認します。以下の行のコメントは外す必要があります。  
`password_encrypt_module ./bfcrypt.dll;./bfpcrypt.conf`
- `bfpcrypt.conf` がエージェントの起動元ディレクトリに存在することを確認します。このファイルには少なくとも 1 つの鍵項目が含まれていなければなりません。
- エンジンの `bfpcrypt.conf` の最終項目が、エージェントの `bfpcrypt.conf` 内の任意の場所に存在することを確認します。

これらすべての項目を確認しても問題が解決しない場合は、トレースを有効にして出力ログを調べてみてください。トレースを有効にする手順は、以下のとおりです。

1. `bffagent.conf` を開きます。
2. 以下の行のコメントを外します。

```
activity_log bffagent.log
```

`bffagent.log` の代わりに別のパスを指定できます。

### **bfcrypt.dll のパスの問題**

`bffagent.conf` では、`password_encrypt_module` プロパティが `bfcrypt.dll` への正確なパスを指していなければなりません。例:

```
password_encrypt_module /opt/buildforge/bfcrypt.dll
```

トレースをオンにすると、このパスの問題点が、出力に以下のように示されます。

```
[ 8928] bfcryptloader.c : 208: Loading password encryption module.
[ 8928] bfcryptloader.c : 223: module: bfcrypt
[ 8928] bfcryptloader.c : 232: Loading module: C:/BuildForge71.181.Agent/bfcrypt.dll
[ 8928] bfcryptloader.c : 262: Failed loading DLL, error code = 0
```

正常にロードされると、出力は以下のようになります。

```
[ 12248] bfpwdlocloader.c: 134: Looking for password locator: ssl_key_password_locator
[ 12248] bfpwdlocloader.c: 244: Looking for password for prop ssl_key_password from bffagent.conf.
[ 12248] bfcryptloader.c : 208: Loading password encryption module.
[ 12248] bfcryptloader.c : 223: module: bfcrypt
[ 12248] bfcryptloader.c : 232: Loading module: ./bfcrypt.dll
[ 12248] bfcryptloader.c : 269: Loading procedure bfcrypt_init.
```

### **パスワード暗号化解除の失敗**

鍵が間違っているなどの理由でパスワードの暗号化解除が失敗すると、ログには以下のような行が示されます。

```
[ 4912] agent.c : 237: AUTH failed
```

パスワードが正しいことが分かっている場合は、さらに問題を診断することができます。 `bfcrypt.dll` モジュールに対するデバッグを有効にします。デバッグを有効にするには、以下の環境変数を設定します。

```
BFDEBUG_SECURITY=1
```

エージェントがサービスとして稼働している場合は、この設定を全体的に適用する必要があります。

デバッグ出力は、エージェントの起動元ディレクトリー内の `bfcrypt.txt` に置かれます。

以下の出力は、エージェントの `bfpcrypt.conf` に正しい鍵が入っていないことを示しています。

```
load_keys_from_file: Parsed 1 key configurations.decrypt:
  Looking for key matching info: 922492fe0c4010008304c3670e1e0e1e, length=32
decrypt: Comparing against: 4d553f110c401000ac08000051f651f6, length=32
decrypt: Warning! No matching key found.
```

以下の出力は、鍵が正常に一致していることを示しています。

```
load_keys_from_file: Parsed 2 key configurations.decrypt:
  Looking for key matching info: 922492fe0c4010008304c3670e1e0e1e, length=32
decrypt: Comparing against: 7427ab360c4010008f9d000049664966, length=32
decrypt: Comparing against: 922492fe0c4010008304c3670e1e0e1e, length=32
decrypt: Found match: 922492fe0c4010008304c3670e1e0e1e, length=32
```

## bfclient.conf の参照

`bfclient.conf` ファイルには、Build Forge セキュリティーに関する設定が保管されます。このファイルは、Build Forge のインストール・ルート・ディレクトリーにあります。

`bfclient.conf` ファイルには、セキュア通信 (SSL) とパスワード暗号化を有効にするために使用する設定が含まれています。以下のようなセクションがあります。

- 接続プロパティー
- ログイン・プロパティー
- OpenSSL と JSSE の両方が使用する SSL プロパティー
- OpenSSL でのみ使用される SSL プロパティー
- JSSE でのみ使用される SSL プロパティー
- 鍵ストア・プロパティー (`bf_keystore_*`)
- 暗号プロパティー

表 1. 接続プロパティー

属性名	デフォルト	指定可能な値	必須	説明
<code>bf_services_hostname</code>	インストール中に指定されます。	該当なし	はい	Build Forge サービス層が配置されているホスト名
<code>bf_services_tcp_port</code>	インストール中に指定されます。	該当なし	はい	Build Forge サービスに接続するための TCP ポート。これは、SSL が指定されていない場合に使用します。
<code>bf_services_ssl_port</code>	インストール中に指定されます。	該当なし	はい	Build Forge サービスに安全に接続するための SSL ポート。

表 1. 接続プロパティ (続き)

属性名	デフォルト	指定可能な値	必須	説明
bf_services_preferred_protocol	tcp	ssl, tcp	はい	Perl クライアントまたは PHP クライアントの場合、SSL または TCP を指定して接続を行います。Java クライアントの場合、SecureAPIClient Connection オブジェクトが SSL を指定し、APIClient Connection が TCP を指定します。

表 2. ログイン・プロパティ

属性名	デフォルト	指定可能な値	必須	説明
bf_login_user	なし	Build Forge ユーザー・リスト内のユーザー ID	いいえ	ログイン ID として使用されます。このログイン ID は、クライアント・プログラム内にも指定できます。
bf_login_password	なし	bf_login_user のパスワード	bf_login_user を使用している場合は必須	bf_login_user のパスワード
bf_login_realm	なし	LDAP ドメイン名	いいえ	ユーザーがまだユーザー・テーブルに指定されていない場合に照会する LDAP ドメイン。

表 3. OpenSSL と JSSE の両方が使用する SSL プロパティ

属性名	デフォルト	指定可能な値	必須	説明
bf_ssl_usage	なし	jsse, openssl	はい	SSL インプリメンテーションを選択します。この選択に応じて、異なるプロパティが使用可能になります。
bf_ssl_cipher_group	ALL	ALL, HIGH, MEDIUM, LOW	いいえ	SSL ハンドシェークの間に提供される暗号グループを指定します。HIGH は最もセキュア、LOW はパフォーマンスが最良、ALL は相互運用性が最も高くなります。
bf_ssl_cipher_override	なし	ユーザーの指定する暗号スイート	いいえ	bf_ssl_cipher_group をオーバーライドします。SSL ハンドシェークの間に使用する、比較的小規模な暗号セットを選択するために使用できます。
bf_ssl_protocol	TLSv1	TLSv1, SSLv3. インプリメンテーションに応じて異なります。	いいえ	SSL が使用するハンドシェーク・プロトコル。推奨されるプロトコルは TLSv1 です。
bf_ssl_cert_alias	なし	構成された鍵ストア内の、有効な証明書の別名。	いいえ	使用する証明書を指定します。これは、同じ鍵ストアに複数の証明書がある場合に指定可能です。

表 4. OpenSSL でのみ使用される SSL プロパティ

属性名	デフォルト	指定可能な値	必須	説明
bf_ssl_key_ref	openssl_key	秘密鍵を含む任意の有効な PEM 鍵ストア参照。	いいえ	クライアントがサーバーに接続するとき に使用される秘密鍵を含んだ、鍵ストア 構成への参照。使用時には、この秘密鍵 用の有効な証明書を bf_ssl_cert_ref に 指定することも必要です。サーバーが個 人証明書を要求するようにセットアップ されている場合にのみ使用されます。
bf_ssl_cert_ref	openssl_cert	指定された秘密鍵の証明 書を含んだ、有効な PEM 鍵ストア参照。	いいえ	上記の秘密鍵の証明書を含む、鍵ストア 構成への参照。サーバーが個人証明書を 要求するようにセットアップされている 場合にのみ使用されます。
bf_ssl_ca_ref	openssl_ca	このクライアントの接続 先のサーバー証明書を検 証するために使用され る、1 つ以上の証明書を 含んだ有効な PEM 鍵ス トア参照。	はい	SSL ハンドシェイク中にサーバー証明書 の検証に使用される、1 つ以上の署名者 証明書を含んだ鍵ストア構成への参照。 証明書は、CA ルート、中間、または自 己署名のいずれであってもかまいません。

表 5. JSSE でのみ使用される SSL プロパティ

属性名	デフォルト	指定可能な値	必須	説明
bf_ssl_keystore_ref	jsse_keystore	keyEntry (秘密鍵と証明 書) を含む、有効な PKCS12、JKS、または JCEKS 鍵ストア参照。	いいえ	サーバーへの接続時に使用する、クラ イアント用の個人証明書 (秘密鍵と関 連する証明書) を含む鍵ストア構成へ の参照。これは、サーバーがクライア ント認証に個人証明書を要求する場合 にのみ必要です。
bf_ssl_truststore_ref	jsse_truststore	keyEntry (秘密鍵と証明 書) を含む、有効な PKCS12、JKS、または JCEKS 鍵ストア参照。	はい	SSL ハンドシェイク中にサーバー証明 書の検証に使用される、署名者証明書 を含む鍵ストア構成への参照。鍵ストア には、他の証明書署名の検証に使用 される、証明書 trustedCertEntries が 1 つ以上含まれています。

表 6. 鍵ストア・プロパティ

属性名	デフォルト	指定可能な値	必須	説明
bf_keystore_alias	各種	ストリング	はい	これは、SSL 構成が鍵ストア構成を参照 する際に使用する名前です。
bf_keystore_location	各種	指定されたタイプの鍵ス トアへの相対パスまたは 完全修飾パス。	はい	これは、指定されたタイプの鍵ストアを 参照するパスおよびロケーションです。 パスは相対パスであってもかまいません が、開始ディレクトリーに対して正確で なければなりません。

表 6. 鍵ストア・プロパティ (続き)

属性名	デフォルト	指定可能な値	必須	説明
bf_keystore _type	openssl の場合は PEM、jsse の場合は PKCS12。	openssl の場合は PEM。jsse の場合は PKCS12、 JCEKS、または JKS。	はい	鍵ストアのタイプ。bf_keystore _location プロパティが参照する実際の鍵ストア・タイプと一致していなければなりません。
bf_keystore _password	インストール中に指定されます。	鍵ストア・タイプでサポートされるストリング。一部の鍵ストアでは、非 ASCII ストリングをサポートしません。	いいえ	鍵ストアにアクセスするためのパスワード。 OpenSSL の場合、このパスワードは、秘密鍵のみを含んでいる Cert 鍵ストアおよび CA 鍵ストアでは必要ありません。

表 7. 暗号プロパティ

属性名	デフォルト	指定可能な値	必須	説明
bf_pw_crypt _enabled	false	true、false	いいえ	パスワードがエンコードされる (false) または暗号化される (true) ことを指定します。有効にされると、パスワード暗号化インプリメンテーションは bfclient.conf と同じディレクトリーに入っている bfpwcrypt.conf という名前のファイルを使用します。





---

## 第 10 章 エージェントのインストール

このセクションでは、エージェントのインストール、実行、構成、およびトラブルシューティングを行う方法について説明します。

管理コンソールからサーバー・リソースとして使用するそれぞれのホストごとにエージェントをインストールします。エージェントとは、管理コンソールから要求を受け取ってプロジェクトおよびステップを実行するサービスのことです。

---

### Windows プラットフォームへのエージェントのインストール

Windows プラットフォームにエージェントをインストールするには、以下のようになります。

1. インストール・メディアからエージェント・インストール・プログラムを見つけて開始します。インストール・プログラムのファイル名は、`win-bfagent-version.exe` です。

**ヒント:** 「**Rational Build Forge エージェントのインストール**」を選択すると、ランチ・パッドはこのインストール処理を開始します。

2. エージェントの既存バージョンを検出すると、インストーラーは、上書きの確認を求めるプロンプトを出します。「**OK**」をクリックします。デフォルト: **OK**。
3. 「ようこそ (Welcome)」パネルが開いたら、「**次へ**」をクリックします。
4. ご使用条件の条項に同意する場合は、「**同意する (I Agree)**」をクリックします。
5. 「インストール場所の選択 (Choose Install Location)」ウィンドウで、「**宛先フォルダー (Destination Folder)**」を設定し、「**次へ**」をクリックします。デフォルトの場所 `C:\Program Files\IBM\Build Forge\Agent` を使用して、ファイルが容易に見つかるようにします。
6. 「構成 (Configuration)」ウィンドウで、任意の「**エージェント・オプション (Agent Options)**」を選択し、「**インストール (Install)**」をクリックします。
7. 以下のいずれかのインストール方式を選択します。
  - ・ サービスとしてインストールする
  - ・ ユーザー・モード・エージェントをインストールする (**Install User Mode Agent**)

エージェントで GUI アプリケーションを実行する必要がある場合のみ、ユーザー・モード・エージェントを選択します。

**注:** Build Forge エージェントが Windows 7 にインストールされている場合に「**ユーザー・モード (User Mode)**」を選択するときは、リブートが必要です。

8. オプション: 「**Cygwin サポートを使用可能にする**」をクリックします。

**ヒント:** Cygwin Linux<sup>®</sup> エミュレーション環境を使用する場合は、エージェントをインストールする際に Cygwin サポートのインストールを選択することができます。Cygwin サポートをインストールする場合は、以下のステップを実行します。

- a. Cygwin のインストール時、**DOS/text** 行終了を選択する。
- b. プロジェクトで、UNIX<sup>®</sup> スタイルのコマンド構文を使用する。

**重要:**

Cygwin は US ASCII のみで動作します。UTF-8 はサポートしていません。したがって、その他のシステムでは使用できません。

9. エージェントが管理コンソールとの通信で使用する「**ポート**」を指定します。デフォルト・ポートは 5555 です。
10. 「セットアップの完了 (Completing Setup)」パネルで、「**完了**」をクリックします。

**注:** インストール時にポップアップ・ウィンドウを閉じないでください。インストールの実行中に、ポップアップ・ウィンドウが表示されたり非表示になったりしますが、そのままの状態にしてください。

## ユーザー・モード・エージェント

「ユーザー・モード」としてインストールされたエージェントでは、ユーザーが、プロジェクトによって起動されたアプリケーションと対話することができます。

ユーザー・モードは、Windows のエージェント専用用意されているオプションです。このオプションはインストール時に設定されます。インストール後に構成することはできません。ユーザー・モードの用途には、以下のようなものがあります。

- ジョブの実行時に GUI アプリケーションを使用して手動で入力を収集する。これを行うと、ジョブは人による入力に依存したものになります。
- プロジェクトおよびステップのトラブルシューティングを行う。サービス・モード操作時には非表示にされる出力が、ユーザー・モードでは表示されます。ステップの実行中、各ステップごとにコンソール・ウィンドウが生成されます。

**重要:** Windows 7 システム。 Windows 7 システムでユーザー・モード・エージェントをインストールまたは再インストールした場合は、オペレーティング・システムをリブートする必要があります。このエージェントは、オペレーティング・システムがリブートされない限り機能しません。

ユーザー・モードでコンピューターを使用するプロジェクトをセットアップする際には、以下の違いに注意してください。

- ユーザー・モード・エージェントは、システム上の現行ログイン・ユーザーとして動作します。エージェントは、このユーザーがコンピューターにログインしている間だけアクティブになります。ユーザーがログアウトすると、ユーザー・モード・エージェントを実行しているサーバーは使用できなくなります。
- ユーザー・モードのコンピューター上で実行されるステップは、そのコンピューターのどのユーザーにも表示されます。

- このエージェントが実行されている Windows コンピューターでは、各ステップごとにコンソール・ウィンドウが開きます。そこには、管理コンソールからのコマンド・アクティビティーが表示されます。
- ステップから GUI アプリケーションを起動すると、エージェントが実行されている Windows コンピューター上にアプリケーション・ウィンドウが表示されます。管理コンソールは、アプリケーションが終了するまで待機してから、ステップを続行します。

代替方法: ジョブが待機することなく続行されるようにするには、start コマンドを使用します。

- `_USE_BFCREDS` 変数は使用しないでください。この変数を使用するステップはすべて失敗します。
- ユーザーには次の特権が必要です。通常、これらの特権はデフォルトでは使用できません。明示的に追加する必要があります。

```
SeInteractiveLogonRight
SeAssignPrimaryTokenPrivilege
SeImpersonatePrivilege
SeIncreaseQuotaPrivilege
SeTcbPrivilege
```

## Windows オペレーティング・システムでのサイレント・エージェント・インストールの実行

Windows 上でエージェントの自動インストールまたはサイレント・インストールを実行するには、`/S` (大文字の S) オプションを使用します。例えば、コマンド・プロンプトで以下のコマンドを入力します。オプションは大/小文字を区別します。

```
win-bfagent-7.0.1.2305.exe /S
```

サイレント・インストールでは、以下の設定を使用します。変更することはできません。

- 既存インストールの上書き: はい
- インストール場所: `C:\Program Files\IBM\Build Forge\Agent`
- サービスとしてインストールする: はい
- Cygwin サポート: いいえ
- ポート: 5555

---

## UNIX および Linux システムへのエージェントのインストール

プラットフォームごとのインストール指示に従ってください。

### • AIX

1. `aix5-bfagent-<version>.tar.gz` ファイルまたは `aix5np-bfagent-<version>.tar.gz` ファイルを使用します。

`aix5np` ファイルには、PAM 認証のサポートが含まれていません。

#### 重要:

`aixnp` エージェントを `root` として実行するようにインストールする場合、このエージェントは実行時に AIX の認証呼び出しを使用して、指定のサーバー

認証資格情報による認証を行います。 root として実行するようにこのエージェントをインストールしない場合は、bfaagent.conf ファイルで magic\_login 設定を使用して、このエージェントへのアクセスを制限する必要があります。

2. 次のコマンドを入力して、ファイルを解凍します。

```
gzip -d gzipfilename.gz
```

3. 次のコマンドを入力して、ファイルを解凍します。

```
tar xvf tarfilename.tar
```

4. 次のコマンドを入力して、エージェントをインストールします。

```
cd extracted-agent-directory  
./install.sh
```

**重要:** 構成オプション --without-pam を使用して AIX 用にエージェントをコンパイルした場合、そのエージェントの認証は**オフ**になります。

このエージェントが root 特権を使用してインストールされている場合は、ユーザーは、自分で指定したパスワードに関係なく、正当なユーザーとして接続を行うことができます。

PAM を使用しない AIX システム上で実行するようにエージェントをコンパイルする必要がある場合は、このエージェントに専用のアカウントを使用して、そのユーザーとして実行するようにインストールしてください。また、bfaagent.conf ファイルの magic\_login 設定を使用して、このエージェントへのアクセスを制限してください。

#### • HP-UX

1. Rational® Build Forge® エージェントが、プロジェクトに指定されているすべてのオペレーティング・システム・コマンドを確実に検出できるようにします。この作業は、次のどちらかのオプションを使用して実行することができます。

- 必須ディレクトリーをすべて含むようにシステムの /etc/PATH ファイルを編集します。

- HP-UX システム上の bfaagent.conf ファイルに nologonshell 設定を指定します。この設定により、Build Forge® は /etc/profile を処理できなくなります。各プロジェクトに適用する Build Forge® 環境変数で /etc/profile から PATH の値およびその他のすべての必須設定を指定する必要があります。

2. インストール・メディアから hpux11-bfaagent-<version>.tar.gz ファイルを入手します。エージェントのインストール先にこのファイルを置きます。

3. ファイルを解凍します。

```
gzip -d gzipfilename.gz
```

4. tar ファイルを解凍します。

```
tar xvf tarfilename.tar
```

5. エージェントをインストールします。

```
cd extracted-agent-directory  
./install.sh
```

6. /etc/profile の以下の行を変更し、エージェントが非対話式ログイン・シェルでコマンドを実行できるようにします。

```
if [ ! $VUE]
```

この行を以下のように変更します。

```
if [ -z "$VUE" -a -n "$PS1" ]
```

- **Mac OS**

1. インストール・メディアから `mac-bfagent-<version>.dmg` ファイルを入手し、任意の場所に置きます。
2. ファイルをダブルクリックして、そのコンポーネントを解凍します。

- **Red Hat Linux および SuSE Linux**

1. インストール・メディアから、`rh9-bfagent-<version>.rpm` ファイル (Red Hat Enterprise Linux 4 または SuSE Linux 10) または `rhel5-bfagent-<version>.rpm` ファイル (Red Hat Enterprise Linux 5) を入手します。
2. RPM ユーティリティを使用して、ファイルをインストールします。

```
rpm -iUvh rh9-bfagent-version.rpm
```

- **Solaris**

1. SPARC 上の Solaris 9 または Solaris 10 の場合は、`sol9-bfagent-<version>-sparc-opt.gz` ファイルを使用します。
2. パッケージを解凍します。

```
gzip -d solN-bfagent-<version>-platform-opt.gz
```

3. コマンド・プロンプトで次のコマンドを入力します。

```
pkgadd -d ./unzipped-package
```

- **その他のプラットフォーム - ソースからのコンパイル**

別のプラットフォーム用のエージェントが必要な場合は、以下の手順に従って、`src-bfagent-<version>.tar.gz` ファイルを使用して、ソースからエージェントをコンパイルします。

1. ダウンロードした `tar` ファイルを解凍します。
2. `src` ディレクトリーにある構成スクリプトを実行します。

SSL 用に構成を行うには、以下のオプションを使用します。

```
./configure --with-ssl=/usr/include/openssl
```

SSL およびパスワード暗号化の両方に対して構成するには、オプションは次のようになります。

```
./configure --with-ssl=/usr/include/openssl --enable-bfcrypt-dll
```

3. `src` ディレクトリーで **make** コマンドを実行します。

ソース・パックには、GNU C コンパイラーまたはご使用のシステム用の C コンパイラーが必要です。ソース・パック、およびローカル・コンピューター用のインストーラーを含んでいない事前ビルド・エージェントには、コンピューターのシステムの `inetd/xinetd` 構成ファイルにエージェントをインストールするインストーラーが含まれています。

---

## System i プラットフォームへのエージェントのインストール

以下の説明に従って、System i に手動でエージェントをインストールします。

ステップ 7 のコマンド・スクリプトにより、始動時に実行されるジョブ記述が作成され、\*ALLOBJ 特殊権限を持つ BFAAGENT ユーザーとしてエージェントが開始されます。

- \*ALLOBJ 特殊権限を持つすべてのユーザーまたは QSECOFR ユーザーは、管理コンソールで指定したサーバーの認証資格情報を使用して認証できます。
- これらの特権を持たないユーザーを認証するには、bfagent.conf ファイルに magic\_login 設定を構成する必要があります。詳しくは、193 ページの『bfagent.conf の参照』を参照してください。

System i プラットフォームにエージェントをインストールするには、以下のようになります。

1. 製品のインストール・メディアまたはダウンロード・イメージを使用して、iseries-bfagent-<version>.tar.gz ファイルを見つけます。
2. 次のコマンドを入力して、アーカイブから tar ファイルを解凍します。  
gzip -d iseries-bfagent-<version>.tar.gz
3. tar ファイルからファイルを実行可能ファイルに解凍します。  
tar xvf iseries-bfagent-<version>.tar
4. iSeries サーバーで、エージェントのインストール・ディレクトリー (例えば、/bin) に bfagent 実行可能ファイルを置きます。
5. iSeries サーバーで、/etc に bfagent.conf ファイルを置きます。
6. bfagent.conf ファイルで、シェル・オプションのコメントを外し、以下の例のように PASE のデフォルト・シェルを指定するか、自分の好みのシェルを指定します。  
shell /bin/sh
7. 始動時に BFAAGENT ユーザーとして実行されるように System i を構成します。

以下のコマンドを入力して、\*ALLOBJ 特殊権限を持つ BFAAGENT ユーザーを作成し、BFAAGENT ユーザーとして始動時に実行されるジョブ記述を作成します。以下の例では、bfagent 実行可能ファイルが /bin にインストールされています。

```
CRTLIB BFAAGENT
```

```
CRTSBSD SBSD(BFAAGENT/BFAAGENT) POOLS((1 *BASE)) TEXT('Build Forge Agent subsystem')
```

```
CRTJOBQ JOBQ(BFAAGENT/BFAJOBQ) TEXT('Build Forge Agent job queue')
```

```
CRTUSRPRF USRPRF(BFAAGENT) PASSWORD(*NONE) INLMNU(*SIGNOFF) LMTCPB(*YES)  
SPCAUT(*ALLOBJ) TEXT('Build Forge Agent user profile')
```

```
CRTJOBQ JOBQ(BFAAGENT/BFAJOBQ) JOBQ(BFAAGENT/BFAJOBQ)  
TEXT('Build Forge Agent autostart')USER(BFAAGENT) RQSDTA('CALL PGM(QP2SHELL)  
PARM('/bin/bfagent' ' ' -s'))
```

```
CRTCLS CLS(BFAAGENT/BFACLS) TEXT('Build Forge Agent job class')
```

```
ADDRTGE SBSD(BFAAGENT/BFAAGENT) SEQNBR(1) CMPVAL(*ANY) PGM(QCMD) CLS(BFAAGENT/BFACLS)
```

```
ADDJOBQE SBSD(BFAAGENT/BFAAGENT) JOBQ(BFAAGENT/BFAJOBQ) MAXACT(*NOMAX) SEQNBR(10)
```

```
ADDAJE SBSD(BFAAGENT/BFAAGENT) JOB(BFAAGENT) JOBQ(BFAAGENT/BFAJOBQ)
```



## System z プラットフォームでのエージェントのインストールおよび実行

以下の説明に従って、System z 上で Build Forge エージェントのソース・コードを手動で解凍し、コンパイルします。z/OS 用のエージェント・ソース・コードは、コンパイルされていないソースとしてのみ提供されています。バイナリー配布は利用できません。

以下のソフトウェアおよびプログラムが必要になります。

- c89 コンパイラーおよび Unix ヘッダー・ファイル。z/OS システムでは、エージェントは Unix System Services (USS) 環境で実行されます。
- z/OS UNIX シェル・インターフェース。インストール中は、z/OS 上のすべてのコマンドを z/OS UNIX シェルで実行します。
- gzip ユーティリティ。

注: z/OS システムで gzip が使用可能な場合、ソース・パックを z/OS システムに転送してから、z/OS システム上で tar ファイルを解凍できます。z/OS システムで gzip が使用できない場合は、最初にファイルを非 z/OS コンピューターに解凍してから、解凍したファイルを z/OS システムに転送します。

- z/OS 用の以下の Build Forge エージェント・ソース・パックが必要です:  
src-bfagent-<version>.tar.gz

### 前提条件

- クライアントおよび内部通信の SSL を有効にします。126 ページの『クライアントおよび内部接続用の SSL の有効化』を参照してください。
- システム SSL ライブラリーをインストールしておく必要があります。
- エージェント・コードをコンパイルする際には、最新バージョンの SystemSSL を使用してください。

System z プラットフォームにエージェントをインストールするには、以下のようになります。

1. 製品のインストール・メディアまたはダウンロード製品イメージを使用して、エージェント・ソース・パックのファイル src-bfagent-<version>.tar.gz を見つけます。

z/OS 以外のコンピューター上のディレクトリーにソース・パックをコピーまたはダウンロードします。

2. 非 z/OS コンピューターのシェル・プロンプトで、次のコマンドを入力して、エージェント・ソース・パックから tar ファイルを解凍します。

```
gzip -d src-bfagent-<version>tar.gz
```

3. FTP または別の転送方法を使用して、バイナリー・イメージとして tar ファイルを z/OS システムに転送し、専用の HFS サブディレクトリー (通常は、ユーザー・アカウントの USS ホーム・ディレクトリー) に置きます。
4. z/OS システム上で、次のコマンドを実行してエージェント・ソース・コードを作成します。
  - a. 最初のステップは ./configure-zos です。-with-system-ssl を使用することに注意してください。システム SSL のパスを指定します。最新バージョンの SystemSSL を使用してください。

- BuildForge 8.0.0.1 から、bfagent は TLS v1.2 および TLS v1.1 をサポートします。
- TLS v1.2 をサポートする最低限の z/OS バージョンはバージョン 1 リリース 13 です。最低限のバージョン要件を満たしていない場合は、システム SSL のない z/OS で bfagent をコンパイルする必要があります。

```
pax -rf src-bfagent-<version>.tar -ofrom=ISO8859-1,to=IBM-1047
cd bfagent-<version>/src
tcsch
./configure-zos --with-system-ssl=<path>
```

- b. ./configure-zos スクリプトが完了したら、以下のコマンドを実行します。  
./build-zos

注: ./build-zos スクリプトの実行後にエラーが発生する場合は、165 ページの『z/OS でのエージェント・インストールのトラブルシューティング』を参照します。

5. z/OS システム上で、bfagent.conf ファイルを /etc に置きます。

bfagent.conf が /etc にない場合、-f オプションを指定してエージェントを開始する必要があります。190 ページの『bfagent の参照』を参照してください。

6. z/OS システム上で、bfagent 実行可能ファイルを適切な場所 (例えば、/usr/bin または /usr/local/bin) に置きます。
7. z/OS システム上で、以下のコマンドを root として実行します。

```
# extattr +p -s bfagent
```

8. システム SSL を使用している場合は、システム SSL ライブラリーを PROGRAM CONTROL に追加してください。次のコマンドを使用します。

```
ralter program * addmem('SYS1.SIEALNKE'//nopadchk) uacc(read)
```

9. z/OS システム上で root としてログインし、-s オプションを使用してエージェントを手動で開始します。

```
bfagent -s
```

root としてのログインがセキュリティー・ポリシーによって許可されていない場合は、bfagent.conf の magic\_login 設定に関する説明 (193 ページの『bfagent.conf の参照』に記載されています) を参照してください。

エージェントはスタンドアロン・デーモンとして実行され、デフォルト・エージェント・ポート 5555 を使用します。デフォルト・ポートを変更するには、bfagent.conf のポート設定を使用します。190 ページの『bfagent の参照』を参照してください。

注: z/OS システム上に Unix TCP/IP デーモン (inetd または xinetd) がインストールされていて、アクティブな場合は、サービスとして実行され、自動的に開始するように Build Forge エージェントをセットアップできます。187 ページの『UNIX、Linux、および MacOS でのエージェントの実行』を参照してください。

10. z/OS システムで、Telnet コマンドを使用して、接続をテストします。208 ページの『接続のテスト』を参照してください。

## z/OS でのエージェント・インストールのトラブルシューティング

z/OS でエージェント・ソース・コードを作成した後、エラー・メッセージを受け取る場合があります。このトピックでは、いくつかの一般的なエラーの修正について説明します。

configure-zos スクリプトは、一部の共通の値を設定し、システムで使用可能なヘッダーと関数を特定するためのいくつかの基本的なチェックを実行します。

z/OS システム構成にはさまざまな種類があるため、./configure-zos スクリプトはエラーなしで実行されても、./build-zos スクリプトを実行すると以下のエラーが表示される場合があります。

**CEE3501S モジュール CCNDRVR は見つかりませんでした。(The module CCNDRVR was not found.)**

FSUM3066 COMPILE ステップは以下の戻りコードで終了しました。(The COMPILE step ended with the following return code:)

-1: EDC5083I モジュールをストレージにロードしようとしてエラーが発生しました。(An error occurred attempting to load a module into storage.)

このエラーは、必要な動的ライブラリーがコンパイラーによってロードできないことを示します。

次のコマンドを実行してください。% export STEPLIB="SYS1.SCCNCMP"  
./build-zos コマンドを再実行します。コマンドが再度失敗する場合、システム管理者に連絡して、必要なライブラリーを見つけるための支援を要請してください。

**IKJ56228I データ・セット CEE.SCEE0BJ がカタログにないか、カタログにアクセスできません (DATA SET CEE.SCEE0BJ NOT IN CATALOG OR CATALOG CAN NOT BE ACCESSED)**

FSUM3066 COMPILE ステップは以下の戻りコードで終了しました。(The COMPILE step ended with the following return code:)

FSUM3052 データ定義名 C8961 を解決できません。(The data definition name C8961 cannot be resolved.) データ・セットが見つかりませんでした。(The data set was not found.) データ・セット名 CEE.SCEE0BJ が正しく指定されていることを確認してください。(Ensure that data set name CEE.SCEE0BJ is specified correctly.)

このエラーは、コンパイルを完了するために必要なシステム・ライブラリーをリンカーが検出できないことを示します。次のコマンドを実行します。

```
% export _C89_LSYSLIB=SYS1.SCEELKED:SYS1.SCEELKEX
```

```
% export _C89_PSYSLIB=SYS1.SCEE0BJ
```

./build-zos コマンドを再実行します。コマンドが再度失敗する場合、システム管理者に連絡して、必要なライブラリーを見つけるための支援を要請してください。

**IEW2456E 9207 シンボル xxx が解決されません (SYMBOL xxx UNRESOLVED)**

解決されないシンボルのエラーは、ビルド時にシステムの C ライブラリーによってシンボルが定義されることが予期されていましたが、実際にはこの

ライブラリーが存在しない場合に発生します。ほとんどの場合、これは他のシステムからも欠落していることが多いシンボルで、問題を回避するための設定が `config.h` に存在します。

例えば、ご使用のシステムで `unsetenv` 関数を定義しないことがあります。通常は、`configure-zos` スクリプトはこのことを検出するはずですが、検出しない場合は、エージェント・ソース・パックに用意されている `config.h` ファイルを以下のように編集します。

```
#define HAVE_UNSETENV 1 を #undef HAVE_UNSETENV に変更します。
```

```
./build-zos コマンドを再実行して、問題を修正します。
```

注: 他の関数用に同様の `#define` ステートメントが存在します。

---

## 「エージェント更新およびデプロイメント」ユーティリティー

「エージェント更新およびデプロイメント」ユーティリティー **agentupdate** を使用すると、複数のコンピューター上での IBM Rational Build Forge エージェントのインストールと更新が容易になります。

Rational Build Forge エージェントは、Rational Build Forge がタスクを自動化するすべてのコンピューターで必要です。エージェントのインストールまたは更新を大量のコンピューターに対して行う場合、作業に時間がかかることがあります。このユーティリティーを使用すると、すべてのターゲット・コンピューター上でエージェントを同時にインストールまたは更新することができます。ユーティリティーは、コマンド行で開始することも、Rational Build Forge で提供されるプロジェクトを使用して開始することもできます。このユーティリティーでは、ターゲット・コンピューター、ターゲット・コンピューターのログイン資格情報、接続方法、および関連情報を特定するためにユーザーが指定した XML ファイルを使用します。

**重要:** ツールでは、片方向のインストールまたはアップグレード操作がサポートされます。

- ツールでは、インストールまたはアップグレードの実行後のロールバックはサポートされません。
- ツールでは、既存のエージェントのバージョンと、インストールするバージョンとの照合は行われません。既にインストールされているバージョンよりも古いバージョンをインストールするように指定した場合、古いバージョンがインストールされます。

ユーティリティーを開始すると、以下のアクションがターゲット・コンピューターごとに実行されます。

1. コンピューターにログインします。
2. オペレーティング・システムを決定します。
3. 実行中のエージェントがあれば停止します。

ユーティリティーで使用する XML ファイル内の **StopCommand** パラメーターを使用して、Rational Build Forge エージェントを停止するためにカスタマイズしたコマンドを使用できます。

4. コンピューターに使用できる十分な一時スペースがあるかどうかを判断します。

5. オペレーティング・システム用のエージェント・インストーラーをターゲット・コンピューターにコピーします。
6. インストーラーを実行して、コンピューター上でエージェントを更新またはインストールします。

エージェントがデフォルトの場所にある場合は、このプロセスにより `BFAgent.conf` ファイルが保持されます。ユーティリティーで使用する XML ファイル内の **ExistingAgentPath** パラメーターを使用して指定した別の場所にエージェントがある場合も、ファイルは保持されます。

7. 一時スペースからインストーラーを削除します。

注: ユーティリティーによって、操作履歴を取り込むためのログ・ファイルも作成されます。このログ・ファイルは、ユーティリティーが実行されるコンピューターに保管されます。

## サポートされるオペレーティング・システム

「エージェント更新およびデプロイメント」ユーティリティーは、2 つのオペレーティング・システムで動作します。

ユーティリティーは以下のオペレーティング・システムで稼働します。

- Windows
- Linux

### メモ:

- ユーティリティーを実行するコンピューターには、IBM Rational Build Forge に付属する IBM Java をインストールしておく必要があります。
- ユーティリティーには、Rational Build Forge に付属する IBM Tivoli Remote Execution and Access が必要です。

ユーティリティーを使用して、以下のオペレーティング・システム上で Rational Build Forge エージェントのインストールおよび更新を行うことができます。

- Windows
- Linux
- IBM AIX
- HP-UX
- Solaris

## インストール

IBM Rational Build Forge インストーラーにより、「エージェント更新およびデプロイメント」ユーティリティーが以下のファイルにインストールされます。

- Windows システム

```
<bfinstall>%java%agentupdate.jar
<bfinstall>%agentupdate.bat
```
- UNIX システムおよび Linux システム

```
<bfinstall>/Platform/java/agentupdate.jar
```

## UNIX または Linux が稼働するターゲット・コンピューターの接続のセットアップ

UNIX または Linux が稼働するターゲット・コンピューターを、IBM Tivoli Remote Execution and Access を使用してアクセスできるように構成する必要があります。

Tivoli Remote Execution and Access を使用してアクセスできるようにするには、ターゲット・コンピューターが以下の要件を満たしていることを確認します。

**SSH:** 「エージェント更新およびデプロイメント」ユーティリティーがアクセスするすべての UNIX または Linux ターゲット・コンピューターで、SSH を確実に使用することができる。

## Windows が稼働するターゲット・コンピューターの接続のセットアップ

Windows が稼働するターゲット・コンピューターを、IBM Tivoli Remote Execution and Access を使用して検出できるように構成する必要があります。

Tivoli Remote Execution and Access を使用してアクセスできるようにする場合、以下の 2 つのオプションがあります。

- **SSH/RSH:** 「エージェント更新およびデプロイメント」ユーティリティーがアクセスするターゲット・コンピューターで、SSH/RSH を確実に使用できるようにします。

Windows オペレーティング・システムを実行するターゲット・コンピューターで SSH または RSH を使用している場合は、残りの構成ステップをスキップしてもかまいません。

- **Windows オペレーティング・システムを実行するターゲット・コンピューター**を、後続のセクションで示されている Tivoli Remote Execution and Access 要件を満たすように構成します。最初のセクションは、すべての Windows オペレーティング・システム用です。その後のセクションは、特定の Windows オペレーティング・システム用です。

**注:** このトピック内の Windows に関する情報は、以下のリンク先で入手可能な情報から得られたものです。最新情報については、以下のリンク先に示されている Windows オペレーティング・システムでの IBM Tivoli Remote Execution and Access の構成に関する情報を参照してください。

- [http://publib.boulder.ibm.com/infocenter/tivihelp/v11r1/index.jsp?topic=/com.ibm.tivoli.tpm.scenario.doc/sdi/rept\\_prereq\\_win.html](http://publib.boulder.ibm.com/infocenter/tivihelp/v11r1/index.jsp?topic=/com.ibm.tivoli.tpm.scenario.doc/sdi/rept_prereq_win.html)
- [http://publib.boulder.ibm.com/infocenter/rafwhelp/v7r1/index.jsp?topic=/com.ibm.rafw.install.doc/topics/r\\_network\\_connection\\_reqs\\_Windows.html](http://publib.boulder.ibm.com/infocenter/rafwhelp/v7r1/index.jsp?topic=/com.ibm.rafw.install.doc/topics/r_network_connection_reqs_Windows.html)

### すべての Windows オペレーティング・システム

- ターゲット・コンピューターで、リモート・レジストリー管理を有効にする必要があります (これはデフォルトの構成です)。以下のように、「サービス」を開きます。



- Windows XP、Windows Vista の場合: 「スタート」 > 「コントロール パネル」 > 「管理ツール」 > 「サービス」の順に選択します。
- Windows Server 2003、Windows Server 2008 の場合: 「スタート」 > 「設定」 > 「コントロール パネル」 > 「管理ツール」 > 「サービス」の順に選択します。

「リモート レジストリ」サービスで、サービス状況が「開始」になっていることを確認します。

- Tivoli Remote Execution and Access を使用して Windows が稼働するターゲット・コンピュータに接続するには、管理共有を使用可能にする必要があります。デフォルトの管理ディスク共有の例として、C\$ および D\$ があります。共有を使用不可にすると、Tivoli Remote Execution and Access はドライブ内にあるディレクトリを非表示と見なします。この場合、次のメッセージが表示されます。

XCIM0009E: リモート・ターゲット <host\_name> へ接続中にエラーが発生しました。

例外: java.io.FileNotFoundException:

CTGRI0003E 指定されたリモート・パス名が見つかりません: <file\_or\_directory\_path>。

原因: com.starla.smb.SMBException: ネットワーク名が正しくありません。

(The network name is incorrect.)

以下のステップに従って、管理共有を有効にします。

1. 「マイ コンピュータ」をクリックします。
  2. 管理共有を有効にするディスク・ドライブを右クリックします。
  3. 「共有とセキュリティ」をクリックします。
  4. 「このフォルダを共有する」を選択します。
  5. C\$ または D\$ などの共有名を指定して、「OK」をクリックします。
- Windows Scripting Host (WSH) または WMI サービスがターゲットで使用不可にされているか、そうでなければ VBScript が使用不可にされている場合、Windows プロトコルのメソッドのいくつかが機能しません。

## Windows XP Professional

- Windows XP オペレーティング・システムを実行するターゲット・コンピュータでは、Tivoli Remote Execution and Access を機能させるために、簡易ファイルの共有を無効にする必要があります。単純なネットワーキングでは、すべてのログインが強制的にゲストとして認証されます。ゲスト・ログインでは、Tivoli Remote Execution and Access を機能させるために必要な権限がありません。簡易ファイルの共有を無効にするには、以下の手順を実行します。
  1. Windows エクスプローラのウィンドウで、「ツール」 > 「フォルダ オプション」とクリックして、「表示」 タブをクリックします。
  2. 「詳細設定」リストで、「簡易ファイルの共有を使用する」チェック・ボックスの選択を解除します。
  3. 「適用」をクリックし、次に「OK」をクリックします。
- Tivoli Remote Execution Access を使用して正常に通信が行われるようにするには、ターゲット・コンピュータで 135 (RPC) と 445 (TCP) の両方のポートを有効にする必要があります。Tivoli Remote Execution and Access がポート 445 を無効と判断した場合、ポート 139 (NetBIOS over TCP/IP) が使用されます。



- ファイアウォール: Windows XP オペレーティング・システムには、インターネット接続ファイアウォール (ICF) と呼ばれるファイアウォールが組み込まれており、デフォルトでは無効になっています。また、Windows XP Service Pack 2 (SP2) には、Windows ファイアウォールが用意されており、デフォルトでは有効になっています。Windows XP を実行するターゲット・コンピュータでファイアウォールが有効になっている場合、ファイアウォールによって Tivoli Remote Execution and Access からのアクセスがブロックされます。

Windows XP SP2 では、「Windows ファイアウォール」構成の「例外」ページで「ファイルとプリンタの共有」チェック・ボックスを選択することで、アクセスを許可できます。

## Windows Vista

- Guest または Everyone アカウントのファイル共有を有効にして、パスワード保護共有を無効にする必要があります。

パスワード保護の共有を無効にするには、以下のようにします。

1. 「スタート」 > 「コントロール パネル」の順にクリックします。
  2. 「ネットワークとインターネット」をクリックしてから、「共有と探索」をクリックします。
  3. 「パスワード保護共有」の横の下矢印をクリックします。
  4. 「パスワード保護の共有を無効にする」をクリックします。
  5. 「適用」をクリックして、「コントロール パネル」を閉じます。
- 以下のように、レジストリー項目の変更が必要になる場合があります。
    1. 「スタート」 > 「検索の開始」フィールド (画面の左下隅にあります) で、「regedit」と入力してから Enter キーを押します。
    2. 左ペインで、HKEY\_LOCAL\_MACHINE¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥policies¥system フォルダを参照します。
    3. 右ペインで空白の領域を右クリックします。
    4. 「新規」をクリックします。
    5. 「DWORD 値」をクリックします。
    6. 「LocalAccountTokenFilterPolicy」と入力します。
    7. 作成した項目をダブルクリックします。
    8. ボックスに「1」と入力します。
    9. 「OK」をクリックします。
    10. コンピューターを再始動します。
  - ファイアウォール: Windows Vista には、インターネット接続ファイアウォール (ICF) と呼ばれるファイアウォールが組み込まれており、デフォルトでは無効になっています。Windows Vista オペレーティング・システムを実行するターゲット・コンピュータでファイアウォールが有効になっている場合、ファイアウォールによって Tivoli Remote Execution and Access からのアクセスがブロックされます。

## Windows Server 2003

- Tivoli Remote Execution Access を使用して正常に通信が行われるようにするには、ターゲット・コンピュータで 135 (RPC) と 445 (TCP) の両方のポートを有効にする必要があります。Tivoli Remote Execution and Access がポート 445 を無効と判断した場合、ポート 139 (NetBIOS over TCP/IP) が使用されます。

### Windows Server 2008

- Guest または Everyone アカウントのファイル共有を有効にして、パスワード保護共有を無効にする必要があります。

パスワード保護の共有を無効にするには、以下のようにします。

1. 「スタート」 > 「コントロール パネル」の順にクリックします。
  2. 「ネットワークとインターネット」をクリックしてから、「共有と探索」をクリックします。
  3. 「パスワード保護共有」の横の下矢印をクリックします。
  4. 「パスワード保護の共有を無効にする」をクリックします。
  5. 「適用」をクリックして、「コントロール パネル」を閉じます。
- 以下のように、レジストリー項目の変更が必要になる場合があります。
    1. 「開始」 > 「実行」をクリックします。
    2. regedit と入力して、Enter キーを押します。
    3. 左ペインで、HKEY\_LOCAL\_MACHINE¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥policies¥system フォルダを参照します。
    4. 右ペインで空白の領域を右クリックします。
    5. 「新規」をクリックします。
    6. 「DWORD (32 ビット) 値」をクリックします。
    7. 「LocalAccountTokenFilterPolicy」と入力します。
    8. 作成した項目をダブルクリックします。
    9. ボックスに「1」と入力します。
    10. 「OK」をクリックします。
    11. コンピューターを再始動します。
  - ファイアウォール: Windows Server 2008 には、インターネット接続ファイアウォール (ICF) と呼ばれるファイアウォールが組み込まれており、デフォルトでは無効になっています。Windows Server 2008 オペレーティング・システムを実行するターゲット・コンピュータでファイアウォールが有効になっている場合、ファイアウォールによって Tivoli Remote Execution and Access からのアクセスがブロックされます。

## 使用法の概要

エージェント更新およびデプロイメント (**agentupdate**) ユーティリティーを使用するには、以下の 3 つのステップを実行する必要があります。

IBM Rational Build Forge エージェントを複数のコンピュータでインストールまたは更新するには、以下のようにします。

1. **agentupdate** ユーティリティーを実行するコンピュータ上のディレクトリーにエージェント・インストーラーを配置します。

インストーラーは、インストール・メディア、および Jazz Community サイト (Jazz.net) の Rational Build Forge プロジェクト・エリアから取得できます。

2. XML ファイルを作成して、Rational Build Forge エージェントのインストールまたは更新を行うコンピューターを指定します。

**注:** XML ファイルにログイン資格情報を組み込む場合、その資格情報を平文で指定します。パスワードの暗号化については、176 ページの『agentupdate ユーティリティの実行』を参照してください。

3. コマンド行、または提供される Rational Build Forge プロジェクトを使用して、**agentupdate** コマンドを実行します。

XML ファイルについて詳しくは、『XML ファイルでのターゲット・システムの指定』を参照してください。

ユーティリティの実行方法について詳しくは、176 ページの『agentupdate ユーティリティの実行』を参照してください。

## XML ファイルでのターゲット・システムの指定

XML ファイルを使用して、エージェントの更新またはインストールを行うターゲット・システムを指定します。

### XML ファイルの概要

「エージェント更新およびデプロイメント」ユーティリティには、エージェントの更新またはインストールを行うコンピューターを決定する XML ファイルが必要です。このファイルでは、ユーティリティで必要なログイン資格情報やその他の項目も指定する必要があります。

ファイルには以下の要件があります。

- 他のエレメントを格納する 1 つの <AgentDeploymentAndUpdate> エレメント
- 各ターゲット・コンピューターを指定する 1 つ以上の <Target> エレメント
- <Globals> エレメントを使用する場合は、1 つのみ使用できます。このエレメントを、すべてのターゲット・コンピューターに適用する項目に対して使用します。

以下の例に、単一の <Target> エレメントと <Globals> エレメントを示します。

<Globals> エレメントは、単一の <Target> エレメントと共に使用しても効果的ではありません。<Globals> エレメントは、複数の <Target> エレメントがある場合に使用するのが効果的です。

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentDeploymentAndUpdate>

  <Target id="9.184.112.152">
    <Parameter Name="ConnectionType" Value="ssh"/>
    <Parameter Name="UserName" Value="root"/>
    <Parameter Name="Password" Value="passWord"/>
    <Parameter Name="ExistingAgentPath" Value="/usr/local/bin/bfagent"/>
    <Parameter Name="GZIPPath" Value="/usr/contrib/bin"/>
  </Target>

  <Globals>
```

```
<Parameter Name="KeyFile" Value="..¥test¥keyfile.txt"/>
</Globals>
```

```
</AgentDeploymentAndUpdate>
```

## XML ファイルの構文

各 <Parameter> エレメントは、<Target> と <Globals> の両方のエレメントで有効です。この柔軟性により、パラメーターをすべてのターゲット・コンピューターに適用する場合に、<Target> エレメントを使用して id 属性のみを指定して、各 <Parameter> エレメントを <Globals> エレメント内に配置することができます。

パラメーターで Value="" と指定することは、パラメーターを指定しないことと同じ意味になります。

下の表で、ファイル構文について説明します。

ターゲット属性	値
id	ターゲット・コンピューターの名前または IP アドレス。

パラメーター	値
<b>AgentInstallLocation</b>	IBM AIX、HP-UX、および Windows の各システムの場合のみ。その他のプラットフォームでは無効になります。  新規エージェントの配置場所。指定がない場合は、システム・デフォルトの一時ディレクトリが使用されます。
<b>BFAgentPort</b>	IBM AIX および HP-UX の各システムの場合のみ。  使用するエージェント用のポート。応答ファイルの生成時に使用されます。
<b>BFSolarisAdminFile</b>	Solaris システムの場合のみ。  サイレント・インストール用の応答ファイルを保管するターゲット・システム上の場所。指定がない場合は、システム・デフォルトの一時ディレクトリが使用されます。
<b>BFSolarisResponseFile</b>	Solaris システムの場合のみ。  応答ファイルを保管するターゲット・システム上の場所。指定がない場合は、システム・デフォルトの一時ディレクトリが使用されます。
<b>ConnectionType</b>	UNIX または Linux の場合: SSH または RSH  Windows の場合: SSH、RSH、SMB、または CIFS  (RSH はリモート・シェルです。SMB は Server Message Block です。CIFS は Common Internet File System です。)
<b>ConnectionPort</b>	(オプション) ターゲットでデフォルト・ポートを使用しない場合に、( <b>ConnectionType</b> パラメーターで) 指定した接続タイプに対して使用するポート番号。ターゲットでデフォルト・ポートを使用する場合は、このパラメーターを指定する必要はありません。
<b>ExistingAgentPath</b>	IBM AIX、HP-UX、および Windows の各システムで必要。  ターゲット・コンピューター上の既存のエージェントの場所。ユーティリティは、指定した場所にある BFAgent.conf ファイルのバックアップを作成し、インストールまたはアップグレードの完了後にリストアします。他のシステムの場合、デフォルトのシステム・インストール場所が使用されます。

パラメーター	値
<b>GZIPPPath</b>	IBM AIX、HP-UX、および Solaris の各システム。  エージェント・インストーラーによって (インストーラーを解凍する) <b>gzip</b> コマンドが検出されるディレクトリー。
<b>IsSudoEnabled</b>	ターゲット・システムで有効になっている <b>sudo</b> を使用する場合は <b>yes</b> に設定します。 <b>SudoPassword</b> を適切な <b>sudo</b> パスワードに設定します。設定されていない場合、ユーティリティは <b>sudo</b> ログインに対して <b>Password</b> の使用を試みます。
<b>KeyFile</b>	ユーティリティを実行するコンピューター上にある、パスワードの暗号化と暗号化解除に使用する鍵を保管するためのファイル。
<b>LocalTempLocation</b>	IBM AIX、HP-UX、および Windows の場合のみ。  応答ファイルを保管するターゲット・システム上の場所。指定がない場合は、システム・デフォルトの一時ディレクトリーが使用されます。Solaris システムの場合は、 <b>BFASolarisResponseFile</b> または <b>BFASolarisAdminFile</b> を使用します。
<b>ManagedScriptsDir</b>	ユーティリティが実行されるディレクトリーに設定します。このディレクトリーは、 <b>agentupdate</b> ユーティリティを実行するコンピューター上にあります。
<b>SourceDirectory</b>	ターゲット・コンピューター用の (.exe、.rpm、および .gz フォームの) インストーラーを保管するディレクトリー。このディレクトリーは、 <b>agentupdate</b> ユーティリティを実行するコンピューター上にあります。
<b>StartCommand</b>	(オプション) ターゲット・コンピューター上の Rational Build Forge エージェントを開始するためにカスタマイズしたコマンド。  このコマンドを必ず <b>PATH/path</b> 変数に組み込むか、または XML ファイルに絶対パスを指定してください。
<b>StopCommand</b>	(オプション) ターゲット・コンピューター上の Rational Build Forge エージェントを停止するためにカスタマイズしたコマンド。  このコマンドを必ず <b>PATH/path</b> 変数に組み込むか、または XML ファイルに絶対パスを指定してください。
<b>SudoPassword</b>	<b>sudo</b> 用に使用するパスワード。 <b>IsSudoEnabled</b> を参照してください。
<b>パスワード</b>	ターゲット・コンピューターへのログイン時に <b>UserName</b> 用に使用するパスワード。 <b>重要:</b> 最初に、XML ファイルに <b>Password</b> パラメーターを平文で定義します。次に、176 ページの『agentupdate ユーティリティの実行』に示されているように、 <b>-encrypt</b> コマンド行オプションを使用して、パスワードを暗号化します。
<b>TemporaryLocation</b>	ユーティリティを実行する前にインストーラーを保管するターゲット・コンピューター上の場所。
<b>UseManagedScripts</b>	<b>yes</b> または <b>no</b> に設定します。 <b>yes</b> に設定した場合、ユーティリティは、 <b>ManagedScriptsDir</b> ディレクトリー内の <b>StartCommand</b> および <b>StopCommand</b> の検索を試みます。
<b>UserName</b>	ターゲット・コンピューターへのログイン時に使用するユーザー名。

## サンプル XML ファイル

以下のサンプルで、XML ファイルの使用法を示します。

- サンプル 1

このサンプルでは、ターゲットは 1 つしかありません。ターゲット IP アドレスは 9.164.259.143 です。<Target> エレメントには、オプション・パラメーターの

**SourceDirectory** も **TemporaryLocation** も含まれていません。ただし、**<Globals>** エレメントには **SourceDirectory** パラメーターが含まれています。ユーティリティは、このパラメーターの値をターゲットに使用します。

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentDeploymentAndUpdate>

  <Globals>
    <Parameter Name="SourceDirectory" Value="..%test"/>
    <Parameter Name="KeyFile" Value="..%test%keyfile.txt"/>
  </Globals>

  <Target id="9.164.259.143">
    <Parameter Name="ConnectionType" Value="ssh"/>
    <Parameter Name="UserName" Value="root"/>
    <Parameter Name="Password" Value="toor"/>
    <Parameter Name="ExistingAgentPath" Value="/usr/local/bin/bfagent"/>
    <Parameter Name="AgentInstallLocation" Value="/usr/local/bin/bfagent"/>
  </Target>

</AgentDeploymentAndUpdate>
```

- サンプル 2

このサンプルでは、ターゲットが 4 つあります。ただし、どのターゲットにもそれぞれに固有のパラメーターは必要ありません。したがって、**UserName**、**Password**、および **ConnectionType** を含むすべてのパラメーターが、**<Globals>** セクションに指定されています。

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentDeploymentAndUpdate>

  <Globals>
    <Parameter Name="SourceDirectory" Value="..%test"/>
    <Parameter Name="ConnectionType" Value="ssh"/>
    <Parameter Name="UserName" Value="root"/>
    <Parameter Name="Password" Value="toor"/>
    <Parameter Name="TemporaryLocation" Value="/tmp"/>
    <Parameter Name="ExistingAgentPath" Value="/usr/local/bin/bfagent"/>
    <Parameter Name="AgentInstallLocation" Value="/usr/local/bin/bfagent"/>
    <Parameter Name="KeyFile" Value="..%test%keyfile.txt"/>
  </Globals>

  <Target id="9.164.259.143">
  </Target>

  <Target id="9.164.102.169">
  </Target>

  <Target id="9.164.102.196">
  </Target>

  <Target id="9.106.189.157">
  </Target>

</AgentDeploymentAndUpdate>
```

- サンプル 3

ターゲット・システム用の IP アドレスの範囲を指定できます。以下は、各システムで同じログイン資格情報 (ログイン: root、パスワード: password) を使用した、IP アドレスの範囲が 9.184.112.152 から 9.184.112.160 までの場合の例です。

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentDeploymentAndUpdate>
<Target id="9.184.112.152-160">
<Parameter Name="ConnectionType" Value="ssh">
<Parameter Name="ConnectionPort" Value=""/>
<Parameter Name="UserName" Value="root">
<Parameter Name="Password" Value="passWord">
</Target>
</AgentDeploymentAndUpdate>
```

#### • サンプル 4

すべてのターゲットのプロパティ（接続タイプ、接続ポート、ユーザー名、パスワード）が同じである場合は、それらのプロパティをファイルの `Globals` セクションで指定できます。

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentDeploymentAndUpdate>
<Globals>
<Parameter Name="ConnectionType" Value="ssh">
<Parameter Name="ConnectionPort" Value=""/>
<Parameter Name="UserName" Value="root">
<Parameter Name="Password" Value="passWord">
</Globals>

<Target id="9.184.112.152-160">
</Target>

<Target id="9.184.80.12-16">
</Target>
</AgentDeploymentAndUpdate>
```

## agentupdate ユーティリティーの実行

XML ファイルを作成したら、コマンド行または提供される `BuildForgeAgentUpdateUtilityProject` を使用して、「エージェント更新およびデプロイメント」ユーティリティーを実行することができます。

### インストールの注意点

ツールでは、片方向のインストールまたはアップグレード操作がサポートされません。

- ツールでは、インストールまたはアップグレードの実行後のロールバックはサポートされません。
- ツールでは、既存のエージェントのバージョンと、インストールするバージョンとの照合は行われません。既にインストールされているバージョンよりも古いバージョンをインストールするように指定した場合、古いバージョンがインストールされます。

### コマンド行でのユーティリティーの実行

以下の例では、コマンド行でのユーティリティーの実行方法を示します。ユーティリティーの構文については、178 ページの『agentupdate ユーティリティーのリファレンス』を参照してください。



以下のコマンドでは **-preview** オプションを使用します。このオプションを指定すると、ユーティリティはターゲット・コンピューターに接続して、そのコンピューター上で実行するコマンドを表示します。この場合、コマンドは実際には実行されません。

```
java -jar agentupdate.jar -preview -targets agent_deploy_config.xml  
-keyfile mykeyfile.txt
```

次のコマンドでは、入力 XML ファイル内の平文のパスワードをすべて暗号化します。このコマンドでは、どのリモート・コンピューターにも接続されません。

```
java -jar agentupdate.jar -encrypt -targets agent_deploy_config.xml -keyfile mykeyfile.txt  
-logfile LogFile.txt
```

このコマンドでは、ターゲット・コンピューターでの IBM Rational Build Forge エージェントの更新またはインストールを行います。**-sourceDir** オプションは、すべてのターゲットのエージェント・インストーラーを含むディレクトリーを示します。ユーティリティの実行時にこのオプションを指定しない場合は、XML ファイル内に対応するパラメーターを指定してください。XML ファイル内のすべてのターゲットに共通のディレクトリーにインストーラーがある場合は、XML ファイル内の **<Globals>** エlement に **SourceDirectory** パラメーターを指定します。ただし、ターゲット・コンピューターのいずれかのソース・ディレクトリーが異なる場合は、そのターゲット・コンピューターの **<Target>** Element で **SourceDirectory** パラメーターを定義してください。例については、174 ページの『サンプル XML ファイル』の『サンプル 2』を参照してください。

```
java -jar agentupdate.jar -sourceDir .\resources -targets agent_deploy_config.xml -logfile LogFile.txt
```

## BuildForgeAgentUpdateUtilityProject を使用したユーティリティの実行

IBM Rational Build Forge には、ユーティリティの使用法を示す **bfagentupdate.xml** サンプル・プロジェクトが用意されています。

サンプル・プロジェクトを使用するには、管理コンソールのインポート機能を使用するか、**bfimport** コマンドを使用して、プロジェクトをインポートします。このプロジェクトは、**C:\Program Files\IBM\Build Forge\samples\agentupdate\BFproject\** (Windows の場合) または **/opt/buildforge/samples/agentupdate/BFproject/** (Linux の場合) からインポートしてください。

プロジェクトは、プレビュー、暗号化、実行という 3 つのステップを必要とします。これらのうち、プレビューと暗号化のステップは、サンプル・プロジェクトに含まれていません。

- **プレビュー・ステップ:** ビルド・システムはターゲット・コンピューターに接続して、エージェントを更新またはインストールする各ターゲットで実行するコマンドを表示します。この場合、コマンドは実際には実行されません。
- **暗号化ステップ:** ビルド・システムはすべてのパスワードを暗号化します。この場合、ターゲット・コンピューター上ではエージェントの更新もインストールも行われません。このステップで、入力 XML ファイル内の平文のパスワードが、暗号化されたパスワードに置き換えられます。
- **実行ステップ:** ビルド・システムはターゲット・コンピューターに接続して、オペレーティング・システムの識別、実行中のすべてのエージェントの停止、イン

ストーリー用の一時スペースの検査、ターゲット・コンピューターへのインストーラーのコピー、更新またはインストールの実行、および一時スペースからのインストーラーの除去を行います。

これらのステップを要件に応じて有効にしてから、プロジェクトを実行してください。

## agentupdate ユーティリティーのリファレンス

ユーティリティーには、動作を制御するためのコマンド行オプションおよび環境変数がいくつか用意されています。

下の表に、コマンド行オプションをアルファベット順に示します。

オプション	必須	説明
<b>-debug</b>	いいえ	拡張デバッグ出力を有効にします。  DEBUG 環境変数を任意の値に設定して、デバッグを有効にすることもできます。優先順位について詳しくは、179 ページの『XML ファイル・パラメーター、コマンド行オプション、および環境変数の優先順位』を参照してください。
<b>-encrypt</b>	いいえ	XML ファイル内の平文のパスワードを、暗号化されたフォームのパスワードに置き換えます。
<b>-genkey &lt;key_file&gt;</b>	いいえ	パスワードの暗号化と暗号化解除に使用する秘密鍵を生成します。  <b>-genkey</b> オプションを指定する場合は、 <b>-keyfile &lt;key_file&gt;</b> オプションも指定して、指定されたファイルに鍵を保管する必要があります。
<b>-keyfile &lt;key_file&gt;</b>	いいえ	パスワードの暗号化と暗号化解除に使用する鍵を保持します。  このファイルは、 <b>KeyFile</b> XML ファイルのパラメーターを設定して指定することもできます。優先順位について詳しくは、179 ページの『XML ファイル・パラメーター、コマンド行オプション、および環境変数の優先順位』を参照してください。
<b>-logFile &lt;log_file&gt;</b>	いいえ	ユーティリティーがメッセージを記録するファイルを指定します。  デフォルトの名前は <code>RemoteAgentDeployerUpdater_Log.txt</code> です。
<b>-maxThreads</b>	いいえ	ユーティリティーが実行時に使用するスレッドの最大数を設定します。デフォルトでは、ホスト・ハードウェア内のプロセッサ・コアの数の 2 倍の数に設定されます。
<b>-preview</b>	いいえ	各ターゲットに対してユーティリティーが実行するコマンドを (実行せずに) リストします。

オプション	必須	説明
<b>-sourceDir &lt;path&gt;</b>	いいえ	<p>さまざまなプラットフォーム用の (.exe、.rpm、および .gz フォームの) エージェント・インストーラーを保存するディレクトリーを指定します。</p> <p><b>agentupdate</b> ユーティリティーは、各ターゲットのオペレーティング・システムを決定してから、このディレクトリー内にある対応するエージェント・インストーラーを使用します。</p> <p><b>注:</b> このユーティリティーではエージェントのインストーラー・プログラムを使用します。エージェントのソース・コードは使用しません。</p> <p>このディレクトリーは、<b>SourceDirectory</b> XML ファイルのパラメーター、または <b>SOURCEDIR</b> 環境変数を設定して指定することもできます。優先順位について詳しくは、『XML ファイル・パラメーター、コマンド行オプション、および環境変数の優先順位』を参照してください。</p>
<b>-targets &lt;XML_file&gt;</b>	はい	<p>IBM Rational Build Forge エージェントの更新またはインストールを行うターゲット・コンピューターを示すために作成した XML ファイルのパスとファイル名を指定します。</p>

下の表で、環境変数について説明します。

環境変数	必須	説明
DEBUG	いいえ	<p>拡張デバッグ出力を有効にします。</p> <p><b>-debug</b> コマンド行オプションを指定して、デバッグを有効にすることもできます。優先順位について詳しくは、『XML ファイル・パラメーター、コマンド行オプション、および環境変数の優先順位』を参照してください。</p>
SOURCEDIR	いいえ	<p>さまざまなプラットフォーム用の (.exe、.rpm、および .gz フォームの) エージェント・インストーラーを保存するディレクトリーを指定します。</p> <p><b>agentupdate</b> ユーティリティーは、各ターゲットのオペレーティング・システムを決定してから、このディレクトリー内にある対応するエージェント・インストーラーを使用します。</p> <p><b>注:</b> このユーティリティーではエージェントのインストーラー・プログラムを使用します。エージェントのソース・コードは使用しません。</p> <p>このディレクトリーは、<b>SourceDirectory</b> XML ファイルのパラメーター、または <b>-sourceDir</b> コマンド行オプションを設定して指定することもできます。優先順位について詳しくは、『XML ファイル・パラメーター、コマンド行オプション、および環境変数の優先順位』を参照してください。</p>

## XML ファイル・パラメーター、コマンド行オプション、および環境変数の優先順位

ユーティリティーは複数の方法で制御することができます。ユーティリティーに実際にどの方法が適用されるかは、優先順位によって決まります。

「エージェント更新およびデプロイメント」ユーティリティーでは、動作を制御する方法が以下の優先順位で用意されています。

1. <Target> エlement内のパラメーター
2. コマンド行オプション

### 3. <Globals> エlement内のパラメーター

### 4. 環境変数

ユーティリティーは、それぞれの方法を順次検討します。方法が有効な値で定義されている場合、ユーティリティーはその値を使用して、後続の方法をすべて無視します。

## 例

以下の例では、ユーティリティーがインストーラーを見つける方法について説明します。<Target> と <Globals> の両方のElementには、**SourceDirectory** パラメーターが定義されています。これらのパラメーターは、インストーラーが存在する可能性のある場所を示しています。ただし、ユーティリティーが検査するのは、そのうちのいくつかの場所です。ユーティリティーはインストーラーを見つけるとすぐにそのインストーラーを使用します。

1. ユーティリティーは、最初に <Target> Element内のパラメーターを検査します。..`%test%wind` ディレクトリーが存在する場合、ユーティリティーはそのディレクトリー内のインストーラーを検査します。インストーラーがそのディレクトリーにある場合、ユーティリティーはそのインストーラーを使用します。
2. ディレクトリーが存在しないか、存在はするがインストーラーがない場合、ユーティリティーは **-sourceDir** コマンド行オプションが指定されているかどうかを検査します。有効なディレクトリーでオプションが指定されている場合、ユーティリティーはそのディレクトリーを検査して、そのディレクトリーにあるインストーラーを使用します。
3. **-sourceDir** が指定されなかったか、ディレクトリーが有効ではなかったか、あるいはディレクトリーにインストーラーがなかった場合、ユーティリティーは <Globals> Element内のパラメーターを検査します。この時点で、ユーティリティーは..`%test` ディレクトリーを検査します。
- 4...`%test` ディレクトリーでインストーラーが作成されていない場合、ユーティリティーは **SOURCEDIR** 環境変数を検査します。そのディレクトリーが存在しないか、存在はするがインストーラーがない場合、ユーティリティーはエラーを生成します。

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentDeploymentAndUpdate>

<Globals>
<Parameter Name="SourceDirectory" Value=".."%test"/>
<Parameter Name="KeyFile" Value=".."%test%keyfile.txt"/>
</Globals>

<Target id="9.164.102.169">
<Parameter Name="SourceDirectory" Value=".."%test%wind"/>
<Parameter Name="ConnectionType" Value="ssh"/>
<Parameter Name="UserName" Value="root"/>
<Parameter Name="Password" Value="toor123"/>
<Parameter Name="ExistingAgentPath" Value="/usr/local/bin/bfagent"/>
<Parameter Name="AgentInstallLocation" Value="/usr/local/bin/bfagent"/>
</Target>

</AgentDeploymentAndUpdate>
```

## ログ・ファイルを使用したユーティリティー・アクションの表示

ログ・ファイルにはユーティリティー・アクションの履歴が示されます。

**-logFile** オプションを指定して、メッセージを記録するファイルに名前を付けることができます。 **-logFile** オプションを指定しない場合、メッセージはユーティリティーによって RemoteAgentDeployerUpdater\_Log.txt に記録されます。このファイルは、**agentupdate** ユーティリティーを保管するディレクトリーに作成されます。

ログ・ファイルの形式は [MSGTYPE] [IP address/MachineName] [Msg] です。

すべてのターゲットのメッセージが同じログ・ファイルに保管されます。

## トラブルシューティング

**agentupdate** ユーティリティーには、問題のトラブルシューティングに役立つ情報を表示できる方法がいくつか用意されています。

デフォルトでは、接続の問題やディスク・スペースの不足などの一般的な問題を理解できるだけの十分な情報が、ユーティリティーによって提供されます。ただし、詳しいデバッグ情報が必要な場合は、**-debug** コマンド行オプションを使用してください。あるいは、DEBUG 環境変数を任意の値で定義して、ツールからの追加のログ・メッセージ (詳しいデバッグ情報) を取得できるようにします。

### agentupdate のメッセージ参照

**agentupdate** ユーティリティーは、操作中にエラー・メッセージおよび情報メッセージを表示します。

以下のトピックでは、すべての **agentupdate** メッセージへの参照が示されています。

---

**CRRBU0001I** エージェントの更新およびデプロイメント・ユーティリティーが正常に開始されました。

---

**CRRBU0002I** デバッグ・モードがオンです。

---

**CRRBU0003I** パスワード暗号化が完了しました。

---

**CRRBU0004I** XML 構成ファイルが構文解析されました。ターゲットを処理中...

---

**CRRBU0005I** ターゲット [{0}] を処理中

---

**CRRBU0006I** [{0}] プロトコル = {1} ユーザー = {2}

---

**CRRBU0007I** [{0}] 暗号化されたパスワード = {1}

---



---

**CRRBU0008I** 管理対象のスクリプト・ディレクトリー {0} が正常に作成されました。

---

**CRRBU0009I** I[{0}] 暗号化解除されたパスワード = #{1}#

---

**CRRBU0010I** パラメーター鍵ファイル = {0}。

---

**CRRBU0011I** 子の数 : {0}。

---

**CRRBU0012I** パラメーター genkey = {0}。

---

**CRRBU0013I** [{0}] sudo により暗号化されたパスワード = {1}。

---

**CRRBU0014I** テキスト・ノード

---

**CRRBU0015I** コメント・ノード

---

---

CRRBU0016I 属性ノード

---

CRRBU0017I 文書ノード

---

CRRBU0018I 文書タイプ・ノード

---

CRRBU0019I エLEMENT・ノード

---

CRRBU0020I 不明ノード

---

CRRBU0021I ターゲットの総数：{0}

---

CRRBU0022I エLEMENT数 = {0}。

---

CRRBU0023I {0} (ユーザー ID={1}) の Windows プロ  
トコルを試行しています。

---

CRRBU0024I RXA インターフェースを使用してターゲ  
ット・マシンの空き領域を検査していま  
す。

---

CRRBU0025I ターゲット・システムの空き領域 = {0}  
KB。

---

CRRBU0026I OS {0} (インストール場所 {1}) のエー  
ジェントを検査しています。

---

CRRBU0027I エージェントのインストールが以下の場所  
に見つかりました: {0}。

---

CRRBU0028I [プロセスの停止中]

---

CRRBU0029I プロセス ID = {0}、プログラム名 = {1}

---

CRRBU0030I ステージの構文解析中: エLEMENT:{0} =  
{1}

---

CRRBU0031I ターゲットのエージェント・プロセス  
(bfagent) を停止できません。

---

CRRBU0032I ターゲットで実行中のエージェントを停止  
しています: {0}。

---



---

CRRBU0033I Windows 上のサービス {0} を停止しよう  
としています。

---

CRRBU0034I ターゲットのエージェント・サービスを停  
止しています (RXA インターフェースを  
使用)。

---

CRRBU0035I ターゲットのエージェント・サービスを停  
止しています: {0}

---

CRRBU0036I ターゲットの一時ディレクトリー {0} を  
使用します。

---

CRRBU0037I ディレクトリー内の接頭部の一致を検査し  
ています。{0}

---

CRRBU0038I <{0}> でインストーラーを検索していま  
す。

---

CRRBU0039I インストーラー <{0}> をターゲットにコ  
ピーしています (RXA インターフェース  
を使用)。

---

CRRBU0040I 鍵ファイルを作成しました:{0}

---

CRRBU0041I ユーザー構成ファイルのバックアップ:  
{0}

---

CRRBU0042I ターゲットの {0} をクリーンアップして  
います。

---

CRRBU0043I ディレクトリー [{0}] を削除しています

---

CRRBU0044I エージェントのインストール場所 {0}

---

CRRBU0045I インストール進行中 ({0})

---

CRRBU0046I インストール/アップグレードのターゲッ  
ト: {0}

---

CRRBU0047I プレビュー・モードは有効です。

---

CRRBU0048I response.txt をディレクトリー {0} に作成  
しました。

---



---

**CRRBU0049I** ユーザー構成ファイルをリストアしていません: {0}。

---

**CRRBU0050I** コマンドの実行中: {0}

---

**CRRBU0051I** 現行作業ディレクトリーは [{0}] です。

---

**CRRBU0052I** {0} にコピーしています。

---

**CRRBU0053I** ターゲットで実行中の bfdispatch プロセスを強制終了中: {0}

---

**CRRBU0054I** ターゲットの新規スレッドが開始されました。

---

**CRRBU0055I** エージェントをターゲットにインストール中です。

---

**CRRBU0056I** リモート・アクセスの問題: プレビュー・モードを続行できません。

---

**CRRBU0057I** ターゲットで実行中のエージェントを停止しています ({0} を使用)

---

**CRRBU0058I** ターゲット上のエージェントが正常に停止しました。

---

**CRRBU0059I** エージェントがターゲットにインストールされていません。

---

**CRRBU0060I** ターゲットの一時ストレージ・ロケーションは {0} です。

---

**CRRBU0061I** すべてのパスワードを暗号化してください。

---

**CRRBU0062I** インストーラー {0} ({1} 用) を {2} から使用します。

---

**CRRBU0063I** 一時ロケーションへの {0} のコピーが正常に行われました。

---

**CRRBU0064I** 構成ファイルのバックアップが成功しました。

---



---

**CRRBU0065I** 既存のエージェントのインストール場所: {0}。

---

**CRRBU0066I** ターゲットで実行中のエージェントを開始します ({0} を使用)

---

**CRRBU0067I** スレッドを終了しています。

---

**CRRBU0068I** admin ファイル (名前は {0}) を作成しました。

---

**CRRBU0069I** ノード: {0}

---

**CRRBU0070I** ノード属性: {0}

---

**CRRBU0071I** パラメーター logFile = {0}

---

**CRRBU0072I** パラメーター sourceDir = {0}

---

**CRRBU0073I** パラメーター sourceFile= {0}

---

**CRRBU0074I** パラメーター・ターゲット = {0}

---

**CRRBU0075I** getOS: [{0}/{1}]]

---

**CRRBU0084I** ターゲットの情報を含むファイルへのパス (必須)。

---

**CRRBU0085I** {0} から鍵が作成されました。

---

**CRRBU0088I** 最大スレッドを {0} に設定

---

**CRRBU1000W** [{0}] プロトコルが指定されていません。

---

**CRRBU1001W** [{0}] サポートされていないターゲット・プロトコル: {1}

---

**CRRBU1002W** Globals セクションが XML ファイルにありません。

---

**CRRBU1003W** Targets セクションが XML ファイルにありません。

---



---

**CRRBU1004W** 無効なターゲット項目: ターゲット・セクションの ID が XML ファイル内に見つかりません。

---

**CRRBU1005W** [stderr] {0}

---

**CRRBU1006W** [stdout] {0}

---

**CRRBU1007W** ターゲット上の一時ロケーションがヌルに設定されています。

---

**CRRBU1008W** サポート対象外: Sun Solaris バージョン {0}

---

**CRRBU1009W** XML ファイルのターゲット・セクションの指定されたソース・ディレクトリでインストーラーが見つかりませんでした。

---

**CRRBU1010W** インストーラーの場所がコマンド行に指定されていませんでした。

---

**CRRBU1011W** インストーラーの場所が XML ファイルの Globals セクションに指定されていませんでした。

---

**CRRBU1012W** インストーラーの場所が環境変数に指定されていませんでした。

---

**CRRBU1013W** ターゲット・エラー: 一時的な admin ファイルを生成できません。

---

**CRRBU1014W** ターゲット・エラー: 一時的な応答ファイルを生成できません。

---

**CRRBU1015W** [戻りコード] {0}

---

**CRRBU2000E** 例外がキャッチされました。

---

**CRRBU2001E** 始動スクリプトがローカル/リモートいずれのロケーションにも存在しません。

---

**CRRBU2002E** 管理対象のスクリプト・ディレクトリ {0} を作成できません。

---



---

**CRRBU2003E** 必須の引数が欠落しています: {0}

---

**CRRBU2004E** TargetInfoTable データ構造体が不正に使用されています。

---

**CRRBU2005E** 複数の Globals タグが XML ファイル内に見つかりました。1 つだけ指定してください。

---

**CRRBU2006E** 未定義の ManagedScriptsDir パラメーターです。

---

**CRRBU2007E** 始動コマンド <{0}> の実行中に問題が発生しました。

---

**CRRBU2008E** [{0}] 平文のパスワードが指定されていません。鍵ファイルが指定されていません。

---

**CRRBU2009E** 始動/停止スクリプトのファイル・オブジェクトの作成中にエラーが発生しました。

---

**CRRBU2010E** 無効なターゲット項目: XML ファイル内のターゲットのホスト名またはアドレスがヌルです。

---

**CRRBU2011E** 停止スクリプトがローカル/リモートいずれのロケーションにも存在しません。

---

**CRRBU2012E** XML ファイルへの書き込み中にエラーが発生しました。

---

**CRRBU2013E** 重複するターゲット項目を無視します: XML ファイル内のターゲット {0}。

---

**CRRBU2014E** 使用できる領域が十分にありません。使用可能な空き領域 = {0} 必要な空き領域 = {1} ({2} 上)

---

**CRRBU2015E** 接続例外がキャッチされました: {0}

---

**CRRBU2016E** File not found 例外がキャッチされました: {0}

---

---

CRRBU2017E I/O 例外がキャッチされました: {0}

---

CRRBU2018E ターゲットのエージェント・プロセス (bfagent) の停止中にエラーが発生しました。

---

CRRBU2019E ターゲットのエージェント・サービスの停止中にエラーが発生しました。

---

CRRBU2020E ディレクトリーが無効か、またはインストーラーが存在しません: {0}

---

CRRBU2021E ディレクトリー内で接頭部が一致するファイルが多すぎます: {0}

---

CRRBU2022E ディレクトリー内で接頭部が一致するファイルがありません: {0}

---

CRRBU2023E インストーラーのソース・ディレクトリーが指定されていないか、または指定されたディレクトリー内にインストーラーが見つかりませんでした。

---

CRRBU2024E エラー: ターゲットの bfdispatch プロセスを強制終了できません。

---

CRRBU2025E 停止コマンド <{0}> の実行中に問題が発生しました。

---

CRRBU2026E 停止コマンドを完了できません。{0} での処理は継続されません。

---

CRRBU2027E ターゲット上のエージェントの停止中に問題が発生しました。

---

CRRBU2028E 停止コマンドを完了できません。処理は継続されません。

---

CRRBU2029E ターゲットのディスク・スペースが不足しています: {0}

---

CRRBU2030E インストーラーが見つかりません。

---



---

CRRBU2031E 一時ロケーションへの {0} のコピーが失敗しました。

---

CRRBU2032E 構成ファイルのバックアップが失敗しました。

---

CRRBU2033E 構成ファイルのリストアが失敗しました。

---

CRRBU2034E {0} の認証エラーのため、リモート・アクセスが失敗しました。

---

CRRBU2035E サポートされないオペレーティング・システムです。

---

CRRBU2036E 始動コマンドを完了できません。{0} での処理は継続されません。

---

CRRBU2037E 未定義の ManagedScriptsDir パラメーターです。

---

CRRBU2038E サポートされないプラットフォームです。

---

CRRBU2039E StopScript のローカル・ファイルが存在しないか、または接頭部が StopScript\_ の項目が複数あります。

---

CRRBU2040E StartScript のローカル・ファイルが存在しないか、または接頭部が StartScript\_ の項目が複数あります。

---

CRRBU2041E サーバーで認証できません。

---

CRRBU2042E 鍵ファイル {0} を作成できませんでした。

---

CRRBU2043E 暗号化アルゴリズムのエラー。ご使用の環境で DES が使用可能であるかを確認してください。鍵ファイルの作成に失敗しました。

---

CRRBU2044E 鍵ファイル {0} は有効な鍵ファイルではありません。鍵ファイルの設定に失敗しました。

---

---

**CRRBU2045E** 暗号化アルゴリズムのエラー。ご使用の環境で DES が使用可能であるかを確認してください。鍵ファイルの設定に失敗しました。

---

**CRRBU2046E** 鍵ファイル {0} にアクセスできませんでした。鍵ファイルの設定に失敗しました。

---

**CRRBU2047E** 暗号化アルゴリズムのエラー。ご使用の環境で DES が使用可能であるかを確認してください。暗号化に失敗しました。

---

**CRRBU2048E** DES を使用した暗号化のパティंगは、この環境では利用できません。暗号化に失敗しました。

---

**CRRBU2049E** 鍵ファイルから生成された鍵が無効です。正しい鍵ファイルを使用しているかを確認してください。暗号化解除に失敗しました。

---

**CRRBU2050E** パスワードに使用されている文字エンコードはサポートされていません。暗号化に失敗しました。

---

**CRRBU2051E** 暗号に指定されているデータの長さが正しくありません。暗号化に失敗しました。

---

**CRRBU2052E** 予期されたデータ・パティंगが存在しません。暗号化に失敗しました。

---

**CRRBU2053E** 暗号化アルゴリズムのエラー。ご使用の環境で DES が使用可能であるかを確認してください。暗号化解除に失敗しました。

---

**CRRBU2054E** DES を使用した暗号化のパティंगは、ご使用の環境では利用できません。暗号化解除に失敗しました。

---

**CRRBU2055E** 鍵ファイルから生成された鍵が無効です。正しい鍵ファイルを使用しているかを確認してください。暗号化解除に失敗しました。

---

---

**CRRBU2056E** 暗号に指定されているデータの長さが正しくありません。暗号化解除に失敗しました。

---

**CRRBU2057E** 予期されたデータ・パティंगが存在しません。暗号化解除に失敗しました。

---

**CRRBU2058E** パスワードに使用されている文字エンコードはサポートされていません。暗号化解除に失敗しました。

---

**CRRBU2059E** 鍵ファイルが指定されていません。

---

**CRRBU2060E** {0} は有効な応答ではありません。

---

---

## エージェントの実行

このセクションでは、エージェントをセットアップして実行する方法について説明します。通常、エージェントは自動開始サービスまたはデーモンとして実行されます。

### Windows でのエージェントの実行

通常、エージェントはサービスとしてインストールされ、コンピュータの起動時に開始されるよう、「自動」に設定されます。エージェントを開始および停止するには、エージェントのインストールされているコンピュータにログオンしなければなりません。

エージェントを開始および停止するには、「スタート」メニューを使用できます。

- エージェントを開始するには、「スタート」 > 「プログラム」 > 「IBM Rational Build Forge」 > 「エージェント・サービスの開始 (Start Agent Service)」をクリックします。
- エージェントを停止するには、「スタート」 > 「プログラム」 > 「IBM Rational Build Forge」 > 「エージェント・サービスの停止 (Stop Agent Service)」をクリックします。

コマンド・プロンプトで以下のコマンドを使用することもできます。

- net start bfaagent
- net stop bfaagent

### UNIX、Linux、および MacOS でのエージェントの実行

エージェントはサービスとして実行することを目的としており、システム再始動時には自動的に再始動するようにする必要があります。

必要に応じて、inetd または xinetd の構成に bfaagent 項目を追加します。以下に、Linux システム上の xinetd.d にある bfaagent 項目の例 (エージェントは /usr/local/bin にインストールされています) を示します。

```
# description: The IBM Rational Build Forge Agent serves build requests
#   from the IBM Rational Build Forge Management Console.
service bfaagent
{
    disable           = no
    flags             = REUSE
    socket_type       = stream
    wait              = no
    user              = root
    server             = /usr/local/bin/bfaagent
    log_on_failure    += USERID
}
```

必要に応じて、エージェントは inetd/xinetd 環境の外で実行することができます。スタンドアロン・デーモンとして実行するには、-s オプションを使用します。

```
bfaagent -s
```

このオプションを使用すると、エージェントはバックグラウンドに移動し、接続の listen を開始します。このコマンドは、コンピュータの始動時に自動的にエージェントが開始されるよう、始動スクリプトの中に置く必要があります。

## System i でのエージェントの実行

エージェントを System i プラットフォームで実行する計画がある場合は、このトピックの情報を確認してください。

### エージェントのポート番号が固有であることの確認

標準の Build Forge エージェント・ポートであるポート 5555 は、System i サーバー上の他のエージェントに事前に割り当てられていることがあります。この場合、まだ割り当てられていないポートに Build Forge エージェントのポートを変更してから、エージェントを開始してください。これを行うには、bfagent.conf ファイルを直接編集します。詳しくは、191 ページの『エージェントのポートの変更』を参照してください。

### 手動でのエージェントの開始

161 ページの『System i プラットフォームへのエージェントのインストール』のインストール手順のステップ 7 を完了すると、System i の始動時に、エージェントが BFAGENT ユーザーとして開始されます。

または、以下のコマンドを使用することにより、System i 上でエージェントを手動で開始することができます。

```
bfagent -s
```

注: bfagent.conf ファイルが /etc (デフォルトの場所) にインストールされていない場合は、-f オプションを使用して、bfagent.conf の場所を指定します。

bfagent コマンドを実行してエージェントを手動で開始した場合、エージェントは、エージェントを開始するユーザーとして開始されます。

- QSECOFR ユーザーまたは \*ALLOBJ 特殊権限を持つユーザーがエージェントを開始する場合、そのユーザーは、管理コンソールで指定したサーバー認証を使用して認証されます。
- 別のユーザーがエージェントを開始する場合は、bfagent.conf ファイルで magic\_login 設定を構成することによりユーザーを認証します。詳しくは、193 ページの『bfagent.conf の参照』を参照してください。

### i5/OS PASE プログラムがインストールされていることの確認

エージェントは、i5/OS Portable Application Solution Environment (PASE) プログラムとして実行されます。PASE は i5/OS に組み込まれており、AIX バイナリーとコマンドを実行できるようにします。通常、PASE はデフォルトでインストールされます。

PASE プログラムがインストールされているかどうかを確認するには、コマンド行で DSPSFWRSC を実行します。

PASE プログラムがインストールされていない場合、インストール CD からロードします。

## PASE でのエージェントの使用

i5/OS でアプリケーションを作成するために必要なタスクのほとんどは、PASE 環境からアクセスできます。iSeries プラットフォームを対象とするプロセスの自動化を計画および定義する場合は、この事実に留意することが重要です。

ステップ内のコマンドは、PASE シェルによって解釈されます。次の構文を使用して、ネイティブ・コマンドを実行することもできます。

```
system -biOE "<native commands>"
```

**重要:** ステップ内の各 system コマンドは、それぞれ独自のプロセスを実行します。これは、各コマンドが、それ自身のプロセスの中でのみ実行されるということです。

例えば、一連のステップに対してライブラリー・リストを設定するとします。

- CHGSYSLIBL または ADDLIBL をステップ・コマンドとして使用することはできません。これらはネイティブ・コマンド (PASE が認識しないコマンド) であるためです。
- サポートされているネイティブ・コマンド構文 (例えば、system -biOE "ADDLIBL FLGHT400") は、コマンド自体のプロセスのライブラリー・リストを変更するだけなので、ステップ内では使用できません。残りのコマンドおよびステップに、変更による影響はありません。

1 つのステップ、一連のステップ、またはプロジェクトに対してライブラリー・リストを設定することはできませんが、BFAGENT ユーザーの開始コマンド・スクリプトでライブラリー・リストを設定することはできます。161 ページの『System i プラットフォームへのエージェントのインストール』の開始スクリプトの例を参照してください。開始コマンド・スクリプトでライブラリー・リストを設定すると、例の中で BFAGENT ユーザーとして実行されるすべてのプロジェクトおよびステップのライブラリー・リストが設定されます。プロジェクトとステップを実行するユーザーは、必要なライブラリーに対するアクセス権限を持っている必要があります。

ライブラリー・リストを設定するには、必要なライブラリーをリストした、エージェント用のジョブ記述を追加します。以下のジョブ記述の例には、ライブラリー FLGHT400 と FLGHT400M が含まれています。

```
10    UTLIB
20    QGPL
30    QTEMP
40    FLGHT400
50    FLGHT400M
```

エージェントは、その開始ルーチンの中にこのジョブ記述を指定します。例えば、ジョブ記述が BFAJOB の場合、システムの開始ルーチン内の行は次のようになります。

```
ADDAJE SBSDB(BFAGENT/BFAGENT) JOB(BFAGENT) JOBD(BFAGENT/BFAJOB)
```

この解決策は、このエージェントと関連付けられている System i サーバー上で実行される (すべてのステップおよびプロジェクトの) すべてのコマンドに影響を与えます。



## bfagent の参照

bfagent 実行可能ファイルは Build Forge エージェントを開始します。このファイルは、同じディレクトリ内の BFAgent.conf ファイルから構成を読み取ります。

コマンドの構文は次のとおりです。

**bfagent** [-f configfile | -s]

### オプション

#### -f configfile

BFAgent.conf ではなく、configfile の構成ファイルを使用して実行します。これは、UNIX または Linux ではランタイム・オプションです。Windows では、エージェントを手動で実行する際のデバッグ・オプションです。Windows でこれをサービスの開始用に使用することはできません。

**-s**     スタンドアロン・サービスとして開始します。このオプションは、UNIX または Linux でのみ使用できます。これは、**inetd** または **xinetd** のいずれかを使用して bfagent を開始する方法の代わりとなるものです。

---

## エージェントの構成

このセクションでは、インストール後にエージェントを構成する方法について説明します。

## エージェント構成ファイルの配置

エージェント構成ファイルである BFAgent.conf には、エージェントのオペレーションのランタイム構成が指定されています。使用可能なすべてのオプションについて、説明コメントが記載されています。このファイルは、以下に示すエージェントのインストール・ディレクトリにあります。

- Windows のデフォルト: C:\Program Files\IBM\Build Forge\Agent\BFAgent.conf
- UNIX および Linux のデフォルト: /etc/bfagent.conf

**重要:** インストール・ディレクトリにある BFAgent.conf ファイルに変更を加えた場合は、その後、エージェントを再インストールまたはアップグレードした後に、繰り返しそれらの変更を行わなければなりません。インストール時には、毎回、構成ファイルが上書きされます。

以下のようにして、代替構成ファイルを指定することができます。

- UNIX または Linux システムの場合、エージェントのインストール・ディレクトリの外にある構成ファイルを使用して、エージェントの構成を保持することができます。これを行う際は、エージェントの開始コマンドに -f コマンド行オプションを使用してください。例:

```
bfagent -f /opt/bfagent.conf
```

- Windows システムでは、このオプションを指定してサービスを開始することはできません。サービスを使用できるのは、エージェントを手動で実行する場合のみです。これはデバッグ用のツールです。



## エージェントのポートの変更

すでにポート 5555 が使用されているサーバーにエージェントをインストールする場合は、インストール後にエージェントのポートを変更することができます。

Windows オペレーティング・システムでポートを変更するには、以下のようになります。

1. 次の手順に従って、レジストリー・エディターを開きます。「開始」 > 「実行」をクリックしてから、「regedit」と入力します。
2. HKEY\_LOCAL\_MACHINE¥SOFTWARE¥BuildForge Agent に移動します。
3. AgentPort の値を任意のポート番号に変更します。

UNIX、Linux、および Macintosh オペレーティング・システムの場合は、以下のようになります。

1. /etc/services ファイルを開きます。
2. BuildForge Agent Port の値を任意のポート番号に変更します。

## 異なるシェルの構成

BFAgent.conf ファイルのパラメーターを編集することによって、デフォルト・シェル以外のシェルを使用するようにエージェントを構成することができます。

例えば、MKSTool の Korn シェルを使用するように Windows システムを変更するには、以下のコマンドで shell パラメーターを変更できます。

```
shell C:¥MKSTools¥mksnt¥ksh.exe -L -c ¥"%s¥"
```

このコマンドの % の部分は、システムがサーバーにコマンドを送信する際に、ステップ・コマンドによって置き換えられます。この場合、コマンド内で円記号 (¥) のエスケープ文字を使用して、リテラルとして引用符を含めます。

## ネットワーク・ファイル共有システムでのエージェント・コマンドの実行 (Windows)

Build Forge エージェントは、最初に、Windows のシステム・アカウントの資格情報を使用して開始されます。コマンドを実行するために、エージェントは後から Build Forge サーバーの認証資格情報を使用して、Windows で認証を行います。

サーバーの認証資格情報は、ローカル・コマンドの場合には受け入れられますが、エージェントが外部のネットワーク共有ドライブで実行する必要がある一部のコマンドの場合には、受け入れられないことがあります。例えば、ClearCase の動的ビューでファイルを変更する場合、エージェントは、ネットワーク共有ドライブ上の ClearCase ファイルにアクセスする必要があります。

外部ファイル・システムではエージェント・サーバーの認証資格情報が無視されるため、コマンドは失敗します。このシステムでは、エージェントの初期システム・アカウントの資格情報のみが認識されます。

ネットワーク共有ドライブでのコマンド実行に関して問題が発生した場合は、以下のアクションを試行します。

### サーバーの認証資格情報を使用してコマンドを実行する

ネットワーク共用に対するアクセス権限を持つ Build Forge サーバーの認証資格情報を使用してコマンドを実行するには、BFagent.conf ファイルに win\_reexec\_after\_auth 設定を追加します。

Build Forge サーバーの認証資格情報を使用してネットワーク共用へのアクセスを確立する場合には、この設定を追加することが前提条件となります。

win\_reexec\_after\_auth 設定を使用すると、エージェントが Windows での認証の後に新規のプロセスを開始するようになります。新規プロセスにより、エージェントがユーザーの資格情報を変更したことを、ファイル共用システムが認識するようになります。

win\_reexec\_after\_auth を設定すると、エージェントがサービスとして実行され、ネットワーク共用にアクセスするコマンドとアクセスしないコマンドを区別しなくなるため、パフォーマンスに影響が出る場合があります。

### エージェントをシングル・ユーザー・モードで実行する

エージェントのインストール時に、Build Forge サーバーの認証資格情報を使用せずに、シングル・ユーザー・モードでコマンドを実行するようにエージェントをセットアップします。「ユーザー・モード・エージェントをインストールします」オプションを選択します。

指定したユーザーが管理者グループのメンバーである場合は、サーバーの認証資格情報を使用してそのユーザーの資格情報を指定する必要があります。

ユーザーが管理者でない場合は、BFagent.conf の magic\_login 設定を使用して、エージェントに無許可アクセスが行われないようにします。

管理コンソールにログオンすると、エージェントが開始され、指定のユーザー一名として実行されます。すぐに、そのユーザーの資格情報を使用したネットワーク共用へのアクセスが許可されます。

### エージェントを専用のユーザー・アカウントを持つサービスとして実行する

専用のユーザー・アカウントを持つ Windows サービスとして実行されるようにエージェントをセットアップします。この方法を選択した場合、単一ユーザー・アカウントとしてエージェントを実行すること以外できませんが、エージェントで再認証を行う新規プロセスを開始する必要があるため、パフォーマンスへの影響がありません。

専用のユーザー・アカウントを持つサービスとしてエージェントを実行するには、以下のようにします。

1. Build Forge サーバーで、「管理ツール」 > 「サービス」をクリックして、Windows のコントロール パネルを開きます。サービスのリストが開きます。
2. IBM Rational Build Forge Agent のサービスを開きます。
3. エージェント・コマンドを実行させるユーザーのユーザー・アカウント情報を指定します。例えば、ClearCase の管理ユーザー、または ClearCase の動的ビューおよび VOB に対するアクセス権限を持つその他のユーザーの情報を指定します。

## エージェント間の直接ファイル転送を構成する

`_XSTREAM_PROTOCOL` トリガー変数によって、エージェント間の直接ファイル転送がジョブの環境にあるとき、その転送が可能になります。送信側エージェント、受信側エージェント、およびエンジンのすべてが、直接ファイル転送に対応している必要があります。いくつかのパラメーターが、エンコード方式、ネットワーク、およびファイル転送特性を制御します。

いくつかのオペレーティング・システム上のエージェントでは、直接ファイル転送のサポートが制限されています。

- `System z`: 直接ファイル転送はサポートされていません。
- `System i`: 直接ファイル転送は、`_XSTREAM_PROTOCOL` の値が `PRNG` または `PLAIN` である場合のみサポートされます。ファイル転送に `SSL` を使用することはサポートされていません。(`_XSTREAM_PROTOCOL` の値が `AES-CBC`)

344 ページの『トリガー変数のリファレンス』を、また 『`bfagent.conf` の参照』で `xstream_*` パラメーターを参照してください。

## トリガー変数とエージェントのパフォーマンス

2 つのトリガー変数は、ステップ・ログ用に生成されたメッセージの数を削減することによってエージェント全体のパフォーマンスに影響を与えることができます。

- `_SUPPRESS_ENV_OUTPUT`: 設定した場合、ステップ・ログで `ENV` メッセージの出力が抑止されます。
- `_SUPPRESS_LOG_OUTPUT`: 設定した場合、ステップ・ログでほぼすべてのメッセージの出力が抑止されます。

『環境変数の処理』の『トリガー変数のリファレンス』を参照してください。トリガー変数は、プロジェクト環境変数またはステップ環境変数に含まれている場合にジョブに影響します。

## `bfagent.conf` の参照

`bfagent.conf` ファイルには、Build Forge エージェントの動作に関する設定が保管されています。このファイルは、`bfagent` 実行可能ファイルと同じディレクトリーにあります。

このファイルには、すべての設定と内部デフォルト値がリストされています。非アクティブな設定はコメント化されています。

### 設定

#### `activity_log path`

アクティビティー・ロギングをオンにします。情報は、`path` で指定されたファイルに追加されます。パスの存在が必要であり、またエージェントのユーザーがそのパスに対する書き込み権限を所有している必要があります。

注: パスが存在しないか、ファイルへの書き込みができない場合、エージェントはエラーを報告しません。

重要: ファイル・サイズに制限はありません。ファイルの削除は手動で行う必要があります。この設定は、エージェントのデバッグを行うために一時的

に使用することを目的としています。稼働中のエージェントの永続ログとして使用するものではありません。

#### **allow IP-address-or-range [...]**

以下の場合にのみ、この設定を使用します。

- エージェントが Windows 上で実行されている場合。
- 始動時に `-s` オプションが使用されている場合に、エージェントが UNIX または Linux 上でスタンドアロン・モードで実行されている場合。

この設定は、エージェントに対する接続に制限を加えます。

IP-address-or-range に一致する IP アドレスからの接続のみが許可されます。デフォルトではすべてのアドレスからの接続が許可されています。

以下の項目のいずれかまたは両方を指定します。

- **IP アドレス:** IPv4 または IPv6 の完全修飾アドレス。例えば、IPv4 の 255.192.192.003 などです。特定の IP アドレスが許可されます。
- **IP アドレス範囲:** IPv4 または IPv6 の部分修飾アドレス。例えば、IPv4 の 192.168 または 192.168.63 などです。この制限に一致するすべての IP アドレスが許可されます。

**注:** inetd や xinetd などのスーパーサーバー上でエージェントを実行している場合は、別の方法でアクセスを制御してください。ファイアウォール、TCP ラッパー (hosts.allow および hosts.deny)、または xinetd のビルトイン・フィルター機能を使用できます。

**bind** この設定を使用して、エージェントの明示的なバインド・アドレスを指定できます。この設定と「port」設定を一緒に使用することにより、`-s` コマンド行オプションを指定してエージェントを開始したときに、エージェントが接続を listen する方法を決定できます。bfaagent.conf ファイルに指定された値により、エージェントは IPv4 ローカル・ホスト・アドレスにバインドされます。これにより、エージェントは同じコンピューター上にあるコンソールからの接続だけを受け入れます。例えば、bind 255.192.192.003 などです。

**注:** この設定は、システムのサービス・アーキテクチャー (inetd、xinetd、または launchd など) によって開始された Windows または UNIX のエージェントには作用しません。

#### **ccviewroot root-path**

この設定は、このホストのデフォルトのビュー・ルート指定します。詳しくは、ClearCase の資料に記載されている init の説明を参照してください。内部デフォルト値は以下のとおりです。

- Windows: ccviewroot M:
- UNIX または Linux: ccviewroot /view

#### **cc\_suppress\_server\_root**

これを設定した場合、ビューのパスは ccviewroot によって設定されたパスになります。設定していない場合は、ccviewroot によって設定されたパスに、サーバー定義で設定されたパスが付加されます。この設定に値は必要ありません。bfaagent.conf にこの設定があれば、設定されていることになります。

### **command\_output\_cache\_size**

この設定により、指定のサイズ (バイト単位) に達するまで、エージェントが出力をキャッシュに入れるようになります。キャッシュ用の内部デフォルト値はありません。キャッシュを使用すると、エージェントのパフォーマンスが大きく向上し、ネットワークのオーバーヘッドが削減されます。キャッシュ・サイズは、コマンドが生成する出力の量によって異なります。

最小値: 2048。2048 という値は、設定値がそれより小さい場合に内部的に使用されます。

**cygwin** この設定は、エージェントが Windows 上にある場合にのみ使用します。

この設定により、Windows ホスト上で、Linux に似た環境である Cygwin を使用してエージェントが動作するようになります。Cygwin を使用すると、エージェントで数多くの Linux 用ツールを使用することができます。

この設定を使用する際、`cygwin_script_magic` およびシェルの設定も必要になる場合があります。これらの設定を構成する方法の 1 つとして以下の例を示します。

```
cygwin
shell C:¥cygwin¥bin¥bash.exe --login -c "%s"
cygwin_script_magic #!/bin/bash
```

shell の設定は、Cygwin のインストール場所と一致していなければなりません。

### **cygwin\_script\_magic**

この設定は、エージェントが Windows 上にある場合で **cygwin** が設定されているときにのみ使用します。

この設定は、ステップを実行するときに使用する `#!` 行を指定します。デフォルトは `#!/bin/bash` です。

### **default\_logon\_domain**

認証要求にドメインが含まれていない場合に使用するドメインを指定します。指定がない場合、エージェント・コンピューターのドメインが使用されます。

### **digest\_algorithm SHA2**

パスワード暗号化が有効である場合、`bfagent` はデフォルトで SHA1 を使用します。SHA2 を使用するには、この設定を有効にしてください。この設定は、バージョン 8.0 で新規導入されました。

`bfagent` を 8.0 にアップグレードしても、この設定は自動的に `bfagent.conf` に追加されません。SHA2 を使用したい場合は、この設定を `bfagent.conf` に追加する必要があります。 `bfagent` 8.0 を直接インストールする場合、この設定は `bfagent.conf` に含まれています。

既にパスワードを暗号化した場合は、この設定を有効にした後で再度パスワードを暗号化する必要があります。

SHA2 を使用するには、Build Forge® コンソールでの「ダイジェスト・アルゴリズム」のシステム構成設定値が SHA2 に設定されていることを確認してください。この設定については、262 ページの『システム構成の設定』のトピックを参照してください。



注: Secure Hash Algorithm 2 (SHA2) を使用して管理コンソールおよびエージェントでパスワード暗号化を有効にする場合は、パスワード暗号化プロパティー・ファイル `bfpwcrypt.conf` を更新してください。

注: Perl エンジンでは、メッセージ・ダイジェストでの Secure Hash Algorithm 2 (SHA2) はサポートされていません。SHA2 ダイジェスト・サポートは、Build Forge 8.0 および Build Forge Agent 8.0 からの Java MJC 用です。

パスワード暗号化が構成プロパティー・ファイル `bfpwcrypt.conf` で有効になっていない場合は、以下の手順で SHA2 のパスワード暗号化を有効にしてください。

1. パスワード暗号化構成プロパティー・ファイルを名前変更します。例えば、`bfpwcrypt.conf` ファイルの名前を `bfpwcrypt.conf.sha1` に変更します。
2. 「管理」 > 「システム」にナビゲートし、「ダイジェスト・アルゴリズム」設定を **SHA2** に変更します。
3. 管理コンソールを再始動します。新しい `bfpwcrypt.conf` ファイルが作成されます。
4. パスワード暗号化を有効にします。「管理」 > 「セキュリティ」にナビゲートし、「パスワード暗号化の有効化」設定を「はい」に設定します。「保存」をクリックし、「マスター `BFClient.conf` の更新」をクリックします。
5. オプション: 「管理」 > 「セキュリティ」 > 「鍵ストア」にナビゲートし、鍵ストアのパスワードを更新します。「管理」 > 「サーバー」 > 「サーバー認証、管理 (Server Auth, Administration)」 > 「LDAP」および「管理」 > 「ユーザー」に適宜ナビゲートし、それぞれのパスワードを更新します。
6. エージェントでもパスワード暗号化を有効にする場合は、新たに生成された `bfpwcrypt.conf` ファイルを、エージェントのあるサーバーにコピーします。
7. `ssl_key_password` が、`bfagent.conf` ファイルで以前に設定されたときに暗号化されていた場合は、パスワードを再暗号化する必要があります。例えば、`bfagent.conf` ファイル内の行の前にある `#` を削除することによって `digest_algorithm SHA2` を有効にしてから、コマンド行で `bfagent -e <your password>` を使用して再暗号化します。これにより、新しい暗号化パスワードが作成されます。この新しい暗号化パスワードを使用して、`bfagent.conf` ファイル内の `ssl_key_password` をリセットし、保存します。
8. `bfagent` を再始動します。

パスワード暗号化が構成プロパティー・ファイル `bfpwcrypt.conf` で有効になっている場合は、以下の手順で SHA2 のパスワード暗号化を再度有効にしてください。

1. パスワード暗号化を無効にします。「管理」 > 「セキュリティ」にナビゲートし、「パスワード暗号化の有効化」設定を **No** に設定します。「保存」をクリックし、「マスター `BFClient.conf` の更新」をクリック

ックして、bfclient.conf ファイル内の bf\_keystore\_password が {bfcrypt:xxx..}yyy... の形式ではないことを確認してください。

2. パスワード暗号化が初めて有効化されて以降にサーバー認証、LDAP、およびユーザー・パスワードが追加または更新された場合は、「管理」 > 「セキュリティ」 > 「鍵ストア」にナビゲートし、各パスワードを更新します。
3. パスワード暗号化構成プロパティ・ファイルを名前変更します。例えば、bfpwcrypt.conf ファイルの名前を bfpwcrypt.conf.sha1 に変更します。
4. 「管理」 > 「システム」にナビゲートし、「ダイジェスト・アルゴリズム」設定を **SHA2** に変更します。
5. 管理コンソールを再始動します。新しい bfpwcrypt.conf ファイルが作成されます。
6. パスワード暗号化を有効にします。「管理」 > 「セキュリティ」にナビゲートし、「パスワード暗号化の有効化」設定を「はい」に設定します。「保存」をクリックし、「マスター BFClient.conf の更新」をクリックします。
7. オプション: 「管理」 > 「セキュリティ」 > 「鍵ストア」にナビゲートし、鍵ストアのパスワードを更新します。「管理」 > 「サーバー」 > 「サーバー認証、管理 (Server Auth, Administration)」 > 「LDAP」および「管理」 > 「ユーザー」に適宜ナビゲートし、それぞれのパスワードを更新します。
8. エージェントでもパスワード暗号化を有効にする場合は、新たに生成された bfpwcrypt.conf ファイルを、エージェントのあるサーバーにコピーします。
9. ssl\_key\_password が、bfagent.conf ファイルで以前に設定されたときに暗号化されていた場合は、パスワードを再暗号化する必要があります。例えば、bfagent.conf ファイル内の行の前にある # を削除することによって digest\_algorithm SHA2 を有効にしてから、コマンド行で bfagent -e <your password> を使用して再暗号化します。これにより、新しい暗号化パスワードが作成されます。この新しい暗号化パスワードを使用して、bfagent.conf ファイル内の ssl\_key\_password をリセットし、保存します。
10. bfagent を再始動します。

#### **disable\_telnet\_support**

最良の結果を得るために、エージェント接続のテストには Telnet を使用します。

エージェントの場合、Telnet の制御シーケンスの処理および正しい取り扱いに関連して、いくつかの処理オーバーヘッドが必然的に発生します。

この設定を使用して、エージェントが特殊な Telnet 文字コードを処理しないようにすることで、わずかにパフォーマンスを向上させることができます。実稼働環境では、パフォーマンスの向上というメリットを得るためにこの設定を使用します。

#### **disable\_transcode**

オペレーティング・システムが UTF-8 エンコードを使用していない場合



に、エージェントが国際データを変換するために実行する処理をオフにします。エンコードの混合とデータ破壊を防ぐには、エージェントのオペレーティング・システムで UTF-8 を使用します。

オペレーティング・システムが UTF-8 エンコードを使用していない場合、エージェントが、オペレーティング・システムのロケール設定に合った正しいエンコードにデータを変換する必要があります。

ご使用のオペレーティング・システムが UTF-8 を使用していない場合、この設定を使用して、最良の結果を得るとともに、エージェントのパフォーマンスの向上を図ります。

#### **enable\_agent\_dll**

この設定により、DLL プロセスのトレース (デバッグ・ツール) が有効になります。

#### **env\_recursion\_limit number-of-recursions**

事前構文解析における変数置換の再帰上限数を設定します。設定されていない場合の上限数は 32 です。

#### **extensions**

この設定は、外部関数ライブラリーのパスを指定します。関数は、ステップ内でドット・コマンドとして使用することができます。この設定が指定されていない場合は、外部ライブラリーはロードされません。

構文解析時、ステップ・コマンド内の最初のトークンは関数名として認識されます。2 番目のトークンはストリング、3 番目のトークンは整数によるタイムアウト値 (秒単位) です。

要件: オペレーティング・システムがダイナミック・ローダーをサポートしていること。例えば、UNIX または Linux では、/usr/include/dlfcn.h が必要になります。以下のデフォルト値は内部で使用されます。

- UNIX または Linux: /usr/local/bin/bfextensions.so
- Windows: C:\program files\ibm\build forge\agent\bfextensions.dll

#### **getaddrinfo\_using\_addrconfig**

この設定は、UNIX または Linux のオペレーティング・システム上で、エージェントをスタンドアロン・サービスとして実行する場合 (bfagent -s) にのみ使用します。この設定を使用すると、getaddrinfo() を呼び出して listen インターフェースを選択する際に、エージェントが AI\_ADDRCONFIG を使用するようになります。デフォルトでは、AI\_ADDRCONFIG は使用されません。

この設定を使用すると、アドレスが正しく構成されていないインターフェースをエージェントが無視するようになります。正しく構成されたアドレスを持つインターフェースのみが listen されます。

#### **gsk\_ssl\_key\_location [ <kdb\_path> | <SAF\_specification>]**

kdb ファイルの絶対パスまたは SAF 鍵リング仕様のいずれかを指定します。

#### **gsk\_ssl\_kdb\_password <password>**

kdb ファイルのパスワード。非暗号化テキストでも暗号化テキストでも構いません。SAF 鍵リングが使用されている場合は、NULL を使用してください。

い。暗号化されたパスワードを非暗号化テキストから作成するには、`bfagent -e <plaintext>` を使用してください。

**`gsk_ssl_protocol <protocol>`**

使用するプロトコルで、ALL (デフォルト)、SSLV2、SSLV3、TLSV1、または TLSV1\_1 のいずれかです。

**`gsk_ssl_cipher_v2 <seed>`**

システム SSL バージョン 2 (SSLV2) に使用する暗号スイート。デフォルト値は 6321 で、これが多くのアプリケーションに適しています。詳しくは、System z に関する資料を参照してください。

**`gsk_ssl_cipher_v3 <seed>`**

システム SSL バージョン 3 (SSLV3) に使用する暗号スイート。デフォルト値は 0906030201 で、これが多くのアプリケーションに適しています。詳しくは、System z に関する資料を参照してください。

**`gsk_keyring_label <label>`**

kdb ファイルの鍵ラベル。

**`gsk_password_encrypt [true | false]`**

暗号化されたパスワードを参照するために使用されます。true に設定されている場合、暗号化された値を `bfagent -e <plaintext>` で作成し、`gsk_ssl_kdb_password` を設定してください。デフォルトで、これは false に設定されています。

**`gsk_ssl_client_authentication [true | false]`**

クライアント証明書を検証するかどうかを指定します。デフォルトは false です。

**`lang lang-code`**

この設定は、管理コンソールで有効な言語が提供されない場合にのみ使用します。

この設定は、エージェントがメッセージおよびコマンド出力を書き込むために使用する言語を指定します。エージェントは、管理コンソールで指定する言語を使用するため、通常、これは明示的に設定されません。ただし、希望のロケールがコンピューターで使用できない場合は、言語の設定が役に立つことがあります。また、管理コンソールが言語を伝達できない場合や、無効な言語を伝達する場合には、バックアップとしても役立ちます。

内部のデフォルト値は en です。次のような明示的設定がある場合と同様の動作になります。

```
lang en
```

**`leave_tmp_file`**

トラブルシューティング時にのみこの設定を使用します。

この設定を使用すると、コマンド実行後に、ステップ・コマンドを保持するために使用する一時ファイルが削除されずに保存されるようになります。トラブルシューティングの際、このファイルは、管理コンソールにステップが表示されたときに、そのステップと比較することができます。

注: 通常の操作時にはこの設定を使用しないでください。

### **locale locale-code.charset-code**

この設定は、UNIX および Linux オペレーティング・システムの場合にのみ使用します。Windows によるロケールの処理は、異なります。

この設定は、ローカライズされたアプリケーションが使用する言語およびマルチバイト文字セットを指定します。この設定を行うには、エージェント・コンテキストの LANG 環境変数を設定します。

コマンド出力を米国英語の UTF-8 として処理するようにエージェントをセットアップするには、オペレーティング・システムで UTF-8 のロケールを使用します。例えば、Linux の場合は、以下の表記を使用します。

```
locale en_US.UTF-8
```

ご使用のオペレーティング・システムの UTF-8 ロケールの正しい表記を確認するには、**locale -a** コマンドを実行します。

この設定が指定されていない場合、エージェントはオペレーティング・システムのロケールを使用します。この設定は便利ですが、オペレーティング・システムのデフォルト・ロケールが、エージェントで使用したいロケールではない場合、この設定が特に役に立ちます。また、エージェントの要件を満たすようにシステム・ロケールを変更することが実用的ではない場合にも、この設定は非常に有用です。

### **magic\_login user:encoded-password**

エージェントは通常、管理特権 (root や admin など) を使用してオペレーティング・システムにログオンします。magic\_login 設定は、標準的なシステム認証の代替となる方法です。この設定により、単一のユーザー名およびパスワードを使用したユーザーのログインを、システムが認証できるようになります。

エージェントが root ユーザーまたは admin ユーザーとして実行された場合、この設定は無視され、通常の認証が試行されます。

エージェントは、ログインに使用されたユーザー名ではなく、そのエージェントを開始したユーザーの権限を使用して、すべてのコマンドを実行します。

この設定は、以下の状態の場合のみ使用します。

- 管理特権を持ったエージェントの実行が不可能な場合。例えば、PAM を使用しない UNIX システムでこの設定を使用します。
- 管理特権を持ったエージェントの実行が許可されない場合 (セキュリティ・ポリシー上の理由)。

エージェントのログインを構成するには、以下のようになります。

1. ユーザー名およびパスワードを使用するサーバー認証を作成します。管理コンソールで、「サーバー」 > 「サーバー認証」とクリックします。
2. この例の場合、ユーザー名は build、パスワードは MySecretPassword です。
3. エージェントを使用するサーバーを作成します。「認証」フィールドで、サーバー認証をこのサーバーと関連付けます。

4. エージェントのエンコード済みパスワードを生成します。エージェントのインストール・ディレクトリーで、選択したパスワードを指定して **bfagent -P** を実行します。

以下のように、SMD5 でハッシュ・エンコードされたパスワードが返されます。

```
bfagent -P "MySecretPassword"  
eca0b7f2f4fbf110f7df570c70df844e1658744a4871934a
```

5. BFAgent.conf で、希望のユーザー名およびエンコード済みパスワードを使用するように magic\_login を設定します。

```
magic_login build:eca0b7f2f4fbf110f7df570c70df844e1658744a4871934a
```

6. エージェントを開始します。
7. サーバー接続をテストします。「サーバー」でサーバーを選択し、「サーバーのテスト (Test Server)」をクリックします。

#### **map drive-and-user-spec[; ...]**

この設定は、マップされたドライブを指定します。一部のシステムでは、ドライブ・マッピングが必要になることがあります。例えば、シェルを共有ドライブから実行するという理由から、ドライブ・マッピングが必要になる場合があります。エージェントに指定されたマッピングは、管理コンソールで \_MAP 環境変数に指定されたマッピングよりも先に実行されます。2 つのドライブ・マッピングの例を次に示します。

```
map X:>//host1/share;Z:>//host2/share(username,password)
```

#### **map\_drive\_is\_failure**

この設定を指定した場合、マップされていないドライブ指定が検出された際に、ステップが実行前に失敗します。この設定が指定されていない場合、エージェントはドライブの障害を無視して、ステップの実行を試みます。その場合、障害によって重要なエラー・メッセージが生成されることを確認します。

#### **no\_preparse\_command**

この設定は、コマンドをシェルに渡す前に、通常エージェントがコマンドに対して実行する変数展開構文解析を無効にします。単一のプロジェクトまたはステップに対して使用可能な \_NO\_PREPARSE\_COMMAND 環境変数も参照してください。

**no\_pty** この設定は、エージェントが UNIX または Linux システム上で実行されている場合にのみ使用します。

この設定を使用して、シェルがエージェントの疑似端末と相互作用する際に、システム・シェルがロックされるのを防ぐことができます。この設定は、通常、HP-UX および z/OS で使用します。このタイプのロックを防止するには、この設定の他に、以下の 2 つの方法を使用できます。

- 代替シェルを使用する
- nologonshell 設定を使用する

**no\_pty** 設定は、疑似端末の割り振りを無効にします。

**注:** `no_pty` の使用は、一部のコマンドに影響を及ぼします。例えば、`ls` コマンドでは、出力が 3 列ではなく、1 列で返されます。この設定を使用する場合は、実稼働環境にジョブをデプロイする前に、入念なテストを行ってください。

### **nologonshell**

この設定は、エージェントが UNIX または Linux 上で実行されている場合にのみ使用します。

この設定は、エージェントで実行するシェルが、ログオン・シェルではなく、通常のシェルになるようにします。この設定が役に立つのは、主に以下の場合です。

- ログオン・シェルが冗長出力を生成する場合
- ログオン・シェルが望ましくない方法で環境設定を変更する場合
- ログオン・シェルがユーザーと対話式に通信しようとする場合

これを設定した場合、シェルがログオン・シェルではなく通常のシェルであることを要求する標準的な方法が使用されます。これは、必ずしもすべてのプラットフォームで機能するとは限りません。このような場合は、`shellflag` 設定を使用して、その動作を変更するためにフラグをシェルに渡すことができます。

これらの動作は、エージェントにとって好ましくありません。ユーザーが対話式ユーザーでないユーザーとして実行されるためです。

**注:** Mac OS X 10.5 システムは、`/bin/bash` を使用します。`/bin/bash` は、`nologonshell` には応答しません。 `shellflag -l` を使用してください。

**注:** z/OS オペレーティング・システムでは、ログオン・シェルと非ログオン・シェルの両方に対して、常に `/etc/profile` スクリプトを使用します。シェルの動作がエージェントでうまく機能しない場合、必要に応じてスクリプトの内容を変更したり、別のシェルを使用してください。

**shellflag** 設定も参照してください。ログオン・スクリプトの動作の変更には、フラグを使用することができます。

### **password\_encrypt\_module *dll\_path;conf\_path***

エージェントで SSL を使用可能にする場合に必要になります。この設定は、DLL および構成ファイルのパスを指定します。

- `dll_path` は `bfcrypt.dll` のパスです (通常、これは `./bfcrypt.dll` です)。
- `conf_path` は `bfpwccrypt.conf` のパスです (通常、これは `./bfcrypt.conf` です)。

### **port *port-number-or-range* [...]**

この設定は、始動時に `-s` オプションを実行したときに、UNIX または Linux 上でスタンドアロン・モードで実行されているエージェントでのみ使用します。

この設定は、エージェントが管理コンソールとの接続を `listen` するために使用するポートを指定します。

エージェントが管理コンソールとの接続で `listen` するポートを指定します。

注: このポートは、デフォルトで 5555 に設定されています。 UNIX または Linux の場合、この設定は `/etc/services` にあります。

### **read\_timeout**

エージェントが切断されるまでに要求を待つ時間 (秒)。デフォルト値は 1800 秒 (30 分) です。タイムアウトを無効にするには、この値を 0 に設定します。

このディレクティブは、正規のエンジン要求を受け取っていない場合に、クライアント接続接点がポートを開いたままにしないようにするために役立ちます。一部のネットワーク・ポート・スキャンニング・ソフトウェアがこのような動作をします。

このディレクティブには小さすぎる値を設定しないでください。通常のエンジン動作には、要求間に数分のギャップが含まれる場合があります。

### **shell shell\_name [options]**

この設定は、デフォルトのシェルを指定します。内部デフォルト値は以下のとおりです。

- Windows: 以下の設定値が使用されない場合は `shell cmd.exe /q /c "%s"`。
  - cygwin 設定値が使用される場合は、デフォルトは `shell C:%cygwin%bin%bash.exe --login -c "%s"` です。
  - cygwin 設定値が使用されない場合は、デフォルトは `shell cmd.exe /u /q /c "%s"` です。
- UNIX または Linux: ユーザー・アカウントに対して設定されたシェル、または、ユーザーのシェルを判別できない場合には、`/bin/sh`。この設定ではパラメーターを指定できないことに注意してください。ただしパラメーターを受け渡す `shellflag` 設定は使用できます。エージェントは、ハイフンを挿入することによって、自動的にデフォルト値がログオン・シェルになるようにします。例えば、`/bin/ksh` は `-ksh` として送信されます。明示的に `shell` を設定すると、暗黙的に `nologonshell` が設定されます。 `nologonshell` を参照してください。
- *System i*: `shell` 値を `/bin/sh` に設定します。

この設定は、ステップ内から指定変更することができます。 `#!` を含む行から始まるステップによってシェルの設定が指定変更され、`nologonshell` 設定を使用してステップ・コマンドが実行されます。

### **shell\_compatible\_undef\_vars**

この設定は、未定義の変数を強制的に空ストリングとして表示します。これが設定されていない場合、表示は、`$VAR`、`${VAR}`、または `%VAR%` 形式の変数の場合は変数名、`[$VAR]` の場合は空ストリングになります。

### **shellarg**

この設定は、エージェントが UNIX または Linux 上で実行されている場合にのみ使用します。

コマンドがごちゃまぜになっているように見える場合は、この設定を使用します。 Red Hat Linux Enterprise の一部のシェルは、この設定を必要としています。



この設定によりシェルにコマンド・スクリプトを渡す方法が変更されます。通常、スクリプトは、次のように標準入力を使用して渡されます。

```
/bin/sh < /tmp/bfshellscript.sh
```

この設定を使用すると、次のように、スクリプトをパラメーターとして渡すことでスクリプトが実行されるようになります。

```
/bin/sh /tmp/bfshellscript.sh
```

#### **shellflag** *flag*

この設定は、エージェントが UNIX または Linux 上で実行されている場合にのみ使用します。

この設定は、シェルが実行されているときにフラグを追加します。指定できるフラグは 1 つのみです。通常、この設定は、出力または不要な処理を削減するため **rc** スクリプト処理を無効にする場合に使用されます。例:

- **csh** および派生物: **shellflag -f** を使用して、**rc** スクリプト処理を無効にします。
- **bash**: **shellflag --noprofile** を使用して、プロファイル・スクリプト処理を無効にします。

#### **ssl\_ca\_location** *path*

認証局を含む鍵ストア・ファイルを指定します。エージェントがサービスとして実行される場合は、絶対パスを使用します。

#### **ssl\_cert\_location** *path*

プライベート証明書を含む鍵ストアを指定します。エージェントがサービスとして実行される場合は、絶対パスを使用します。

#### **ssl\_client\_authentication** [**true** | **false**]

エージェントへの接続が行われるときにクライアント認証を必要とする場合に、**true** に設定します。 **true** に設定した場合、Build Forge エンジンの証明書をエージェントの認証局の鍵ストアに追加する必要があります。

#### **ssl\_cipher\_group** [*grouplist* | **ALL**]

使用する個々の暗号グループを指定します。ALL (すべて) に設定することができます。

#### **ssl\_cipher\_override** *cyphers*

暗号グループを指定変更します。使用する暗号を指定してください。

#### **ssl\_key\_location** *path*

鍵を含む鍵ストア・ファイルを指定します。エージェントがサービスとして実行される場合は、絶対パスを使用します。

#### **ssl\_key\_password** *password*

鍵のパスワードです。このプロパティは、デフォルトでは平文で保管されます。独自の鍵または Build Forge サーバーの鍵を使用してこのパスワードを暗号化するようにエージェントを構成することができます。

#### **ssl\_protocol** *protocol*

使用する SSL ハンドシェーク・プロトコル。 TLSv1、TLSv1.1、TLSv1.2 のいずれかです。このプロトコルは、Build Forge<sup>®</sup> サーバーで使用されているプロトコルに一致していなければなりません。 TLSv1、TLSv1.1、または



TLSv1.2 を有効にすると、対応する接続のみが受け入れられます。例えば、TLSv1.2 を有効にすると、bfagent は TLSv1.2 接続のみを受け入れます。

#### **update\_path** *path*

この設定は、Build Forge エージェントの実行可能ファイルの絶対パスを指定します。これは、インストール時に自動的に設定されます。設定されるディレクトリーは、オペレーティング・システムのデフォルト・ディレクトリー、またはユーザーが指定したインストール・ディレクトリーです。

**注:** この設定は、Windows のエージェントでは無視されます。更新パスはレジストリー・キーから取得されます。このキーはエージェントのインストール時に設定されます。

#### **win\_reexec\_after\_auth**

Build Forge サーバーの認証資格情報を使用してネットワーク・ファイル共用システム上でエージェントのコマンドを実行する必要がある場合は、この設定を追加します。例えば、ClearCase の動的ビューでファイルを変更する場合、エージェントは、ネットワーク・ファイル共用システム上の ClearCase ファイルにアクセスする必要があります。

Build Forge エージェントは、最初に、Windows のシステム・アカウントの資格情報を使用して開始されます。コマンドを実行するために、エージェントは後から Build Forge サーバーの認証資格情報を使用して、Windows で認証を行います。

この設定を使用しなかった場合、ネットワーク共用は最初の Windows システム・アカウントの資格情報のみを認識し、ネットワーク・ファイル共用システム上のファイルに対するアクセスおよび書き込みに必要な、その後のサーバーの認証資格情報は無視します。

win\_reexec\_after\_auth は、Windows での認証の後に、サーバー認証資格情報を使用して新規プロセスを再度開始し、変更された資格情報をファイル共用システムが認識するようにします。

win\_reexec\_after\_auth 設定を使用すると、エージェントがサービスとして実行され、ネットワーク共用ファイルにアクセスするコマンドとアクセスしないコマンドを区別しなくなるため、パフォーマンスに影響が出る場合があります。

#### **xstream\_allow\_ssl\_mismatch**

OpenSSL でコンパイルされたエージェントと OpenSSL なしでコンパイルされたエージェントとの間でファイル転送が必要な場合は必須です。デフォルトでは、OpenSSL でコンパイルされたエージェントは AES\_CBC 暗号化ファイル転送を必要とします。この設定が使用されていない場合、このようなエージェントは PLAIN または PRNG エンコード方式を使用するファイル転送要求を拒否します。

#### **xstream\_bind** *ip\_address*

直接ファイル転送にのみ使用する IP アドレスを指定します。このアドレスは、ファイルを受け取るエージェントからアクセス可能でなければなりません。デフォルトでは、エージェントはすべてのネットワーク・インターフェースで listen します。bind も参照してください。

#### **xstream\_conn\_timeout** *seconds*

エージェントが接続を待つ時間 (秒)。この時間内に、エンジンは受信側エージェントに接続要求を転送する必要があり、受信側エージェントは送信側エージェントと接続を確立する必要があります。デフォルトでは 20 秒に設定されます。

#### **xstream\_listen\_range** *port-range*

エージェントが接続を listen するポートの範囲。この設定は、接続するホスト間にファイアウォールがある場合に役立ちます。ファイアウォール管理者は、接続を許可されたポート (例えば 22880-22889) を有効にするようにファイアウォールを構成できます。デフォルトのポート範囲は 16384-32767 です。ただし、xstream\_bind が使用され、xstream\_listen\_randomize が使用されない場合、エージェントは範囲を指定せず、オペレーティング・システムがどのポートを使用するかを決定します。

#### **xstream\_listen\_randomize**

xstream\_port\_range 内でランダムにポートを選択します。指定しない場合、エージェントは最小ポート番号からチェックを始めます。この設定は、セキュリティ対策として是非お勧めします。

#### **xstream\_rcv\_timeout** *seconds*

ファイル転送を待つ時間。ファイル転送中のいつでも、受信側エージェントが送信側エージェントからデータを取得することなくこの期間が経過した場合、転送は失敗し、接続は閉じます。デフォルトは 20 秒です。

#### **xstream\_send\_timeout** *seconds*

ファイル転送を待つ時間。ファイル転送中のいつでも、送信側エージェントが受信側エージェントにデータを書き込むことなくこの期間が経過した場合、転送は失敗し、接続は閉じます。デフォルトは 20 秒です。

## **パスワード暗号化が有効になっている場合**

パスワード暗号化が構成プロパティ・ファイル bfpwcrypt.conf で有効になっている場合は、SHA1 から SHA2 への変更後に、このトピックに記載された手順を使用してください。

### **手順**

1. 「管理」 > 「セキュリティ」に移動し、パスワード暗号化を無効にします。
2. 暗号化が有効にされて以降にサーバー認証、LDAP、およびユーザー・パスワードが追加または更新された場合は、「管理」 > 「セキュリティ」 > 「鍵ストア」に移動し、各パスワードを再度更新します。
3. パスワード暗号化構成プロパティ・ファイルを名前変更します。例えば、bfpwcrypt.conf ファイルの名前を bfpwcrypt.conf.sha1 に変更します。
4. 「管理」 > 「セキュリティ」に戻り、ダイジェスト・アルゴリズムを SHA2 に変更します。
5. Build Forge® 管理コンソールを再始動します。新しい bfpwcrypt.conf ファイルが作成されます。
6. 「管理」 > 「セキュリティ」に戻り、パスワード暗号化を有効にします。

7. オプション: 「管理」 > 「セキュリティ」 > 「鍵ストア」に戻り、サーバー認証、LDAP、およびユーザー・パスワードを更新します。
8. エージェントでもパスワード暗号化を有効にする場合は、新たに生成された `bfpcrypt.conf` ファイルを、エージェントのあるサーバーにコピーします。
9. `ssl_key_password` が暗号化されていた場合は、パスワードを再暗号化します。例えば、`re-encrypt bfaagent -e xxxxxx` のように指定します。この例では、`xxxxxx` はパスワードを表しています。
10. `bfaagent.conf` ファイルで、行の先頭にある `#` を除去するか、新たな行を追加して、`digest_algorithm SHA2` を有効にします。
11. `bfaagent` を再始動します。

## パスワード暗号化が有効になっていない場合

パスワード暗号化が構成プロパティ・ファイル `bfpcrypt.conf` で有効にされたことがない場合は、SHA1 から SHA2 への変更に、このトピックに記載された手順を使用してください。

### 手順

1. パスワード暗号化構成プロパティ・ファイルを名前変更します。例えば、`bfpcrypt.conf` ファイルの名前を `bfpcrypt.conf.sha1` に変更します。
2. 「管理」 > 「セキュリティ」に移動し、ダイジェスト・アルゴリズムを SHA2 に変更します。
3. Build Forge® 管理コンソールを再始動します。新しい `bfpcrypt.conf` ファイルが作成されます。
4. オプション: 「管理」 > 「セキュリティ」に移動し、パスワード暗号化を有効にします。
5. オプション: 「管理」 > 「セキュリティ」 > 「鍵ストア」に移動し、サーバー認証、LDAP、およびユーザー・パスワードを更新します。
6. エージェントでもパスワード暗号化を有効にする場合は、新たに生成された `bfpcrypt.conf` ファイルを、エージェントのあるサーバーにコピーします。
7. `ssl_key_password` が暗号化されていた場合は、パスワードを再暗号化します。例えば、`re-encrypt bfaagent -e xxxxxx` のように指定します。この例では、`xxxxxx` はパスワードを表しています。
8. `bfaagent.conf` ファイルで、行の先頭にある `#` を除去するか、新たな行を追加して、`digest_algorithm SHA2` を有効にします。
9. `bfaagent` を再始動します。

---

## エージェントのトラブルシューティング

このセクションでは、正しく機能しないエージェントをトラブルシューティングする際に使用可能な手順について説明します。手順は順序に従って進めてください。これらの手順を実行してもエージェントが作動しない場合には、サポートまでお問い合わせください。

## ホストの名前解決のテスト

管理コンソール・ホストからエージェント・ホストに到達可能であるかどうかを検査します。管理コンソール・ホストの `ping` ユーティリティを使用して、エージェント・ホストをテストします。

```
ping hostname
```

管理コンソールとエージェントの両方がインストールされている Windows で、以下のサンプル・セッションが実行されます。

```
C:\> ping localhost
```

```
Pinging somehost.city.company.com [127.0.0.1] with 32 bytes of data:
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

次のようなメッセージは、問題を示しています。

```
Unknown host
```

管理コンソール・ホストのネットワーク構成に問題があります。ネットワーク管理者に問い合わせてください。

## 接続のテスト

Telnet を使用するか、管理コンソールからのテストを使用して、エージェントに対する接続をテストすることができます。

コマンド行から接続をテストするには、以下のようになります。

1. Telnet コマンドを使用してエージェントに接続します。エージェントが実行されているホストにログオンしている場合は、ホスト名として `localhost` を使用できます。

```
telnet hostname 5555
```

この応答は、接続が成功したことを示しています。

```
200 HELLO - BuildForge Agent v7.0.1.buildnumber
```

2. 自分のログイン資格情報を使用して次のコマンドを実行し、認証を確認します。

```
telnet localhost 5555
username user name
password password
cmd ping
go
```

以下のメッセージは成功を示しています。

```
AUTH: set user account to <user name>
```

上記のテストは機能するが、ジョブが失敗し、サーバーのテストにユーザー認証エラーが表示される場合は、プラグ可能認証モジュール (PAM) の構成を確認してください。次のようなメッセージが表示される場合は、次のステップに進んでください。

AUTH: unable to set user account to <user name>: unknown account (1)

3. 次のコマンドを入力します。

```
cmd ping go
```

典型的な Telnet セッションの出力を以下に示します。特に、出力の最後の部分にある RESULT 0 を探してください。これはテストが成功したことを示します。このテストは、Windows 上で実行中のエージェントで実行されました。

```
300 DATA s 67
AUTH: Running as: [SYSTEM] in domain [NT AUTHORITY] SID Type [User]
300 DATA s 52
AUTH: Running with Privilege: [Lock pages in memory]
300 DATA s 66
AUTH: Running with Privilege: [Adjust memory quotas for a process]
300 DATA s 63
AUTH: Running with Privilege: [Create permanent shared objects]
300 DATA s 46
AUTH: Running with Privilege: [Debug programs]
300 DATA s 56
AUTH: Running with Privilege: [Bypass traverse checking]
300 DATA s 61
AUTH: Running with Privilege: [Back up files and directories]
300 DATA s 54
AUTH: Running with Privilege: [Change the system time]
300 DATA s 68
AUTH: Running with Privilege: [Remove computer from docking station]
300 DATA s 73
AUTH: Running with Privilege: [Impersonate a client after authentication]
300 HEARTBEAT 1
300 DATA s 16
PLAT: Windows XP
250 RESULT 0
PING: internal loopback test complete
260 EOR
```

管理コンソールから接続をテストするには、以下のようにします。

**注:** この方法は、サーバーがエージェントを使用するようにセットアップされている場合にのみ使用します。

1. 「サーバー」に移動します。
2. サーバーのリストで、テストするサーバーをクリックします。
3. 「接続のテスト」をクリックします。

テスト終了後の結果は、「テスト結果」タブに表示されます。

次の出力のようなメッセージは、問題を示しています。

```
Could not open a connection to host on port 5555
```

管理コンソールがホストに接続できないか、エージェントが機能していません。

## Windows でのエージェントのトラブルシューティング

Windows でエージェントのトラブルシューティングを実行するには、以下のようになります。

1. インストールされている実行可能ファイルを確認します。エージェントのインストール・ディレクトリーに以下のファイルがあることを確認します。

- bfaagent.exe
- bfdispatch.exe

注: 次に進む前に、カスタム構成があるかどうかを確認します。カスタム構成を使用している場合は、インストール・ディレクトリーの外に BFAgent.conf を保管しておき、インストール後にそれを復元してください。

2. エージェントを再インストールします。エージェントの再インストールにより、Windows エージェントのほとんどの問題を解決できます。再インストールにより、実行可能ファイルが更新されて、レジストリー・キーが復元されます。

## UNIX、Linux、または MacOS でのエージェントのトラブルシューティング

UNIX、Linux、または MacOS 上でエージェントのトラブルシューティングを行うには、以下の手順を実行します。

- シェルから **bfaagent** を実行します。正常な応答は、以下のメッセージのようになります。

```
200 HELLO - Build Forge Agent v7.0.1.122
```

この例のようなメッセージを受け取ったが、共用ライブラリーの問題が発生している場合は、それらの問題に関するメッセージが表示されます。共用ライブラリーの問題のほとんどは、パスを正しく設定することにより解決できます。

- エージェントが **listen** を実行しているか検査します。次のコマンドを実行します (ポートはデフォルトの 5555 であると仮定しています)。

```
telnet localhost 5555
```

200 HELLO 応答は、エージェントが **listen** していることを示しています。この応答が返されない場合は、システムのネットワーク構成を確認してください。

**inetd** の構成が正しいか検査するか、Linux または UNIX のシステム管理者に相談してください。

- 認証を検査します。自分のログイン資格情報を使用して、次のコマンドを実行します。

```
telnet localhost 5555
username <user name>
password <password>
cmd ping
go
```

次のようなメッセージは、認証が正しく行われていることを示しています。

```
AUTH: set user account to <user name>
```



上記のテストは機能するが、ビルドが失敗し、サーバーのテストにユーザー認証エラーが表示される場合は、プラグ可能認証モジュール (PAM) の構成を確認してください。次のようなメッセージが表示される場合は、次の手順に進んでください。

AUTH: unable to set user account to *user name*: unknown account (1)

- PAM 構成を確認します。PAM 構成の問題は、AIX プラットフォーム上の共通の問題です。オペレーティング・システムに応じて、PAM は、`pam.conf` ファイル内の行を使用した方法と、`pam.d` ディレクトリー内のファイルを使用した方法のいずれかによって構成されます。

**ヒント:** Solaris 10 は例外です。`pam_dial_auth` というモジュール (例えば、`pam_dial_auth.so.1` など) を指定するすべての行を削除してください。このモジュールが含まれていると、エージェント認証は機能しません。

1. `/etc/pam.conf` が存在するかを確認します。存在しない場合は、このトピックで後述する `pam.d` の説明を参照してください。このファイルが存在する場合は、次のステップに進みます。
2. ファイルで、`bfagent` の項目を作成します。
3. 別のアプリケーション (`sshd` や `login` など) の行をコピーして、`[application]` フィールドを `bfagent` に置き換えます。

`[application] [when] [mode] [module]`

フィールドは次のとおりです。

- `[application]` は、ユーザーの認証に必要なアプリケーションの名前です。
- `[when]` は認証要求のタイプです。
- `[mode]` は認証要求のモードです。
- `[module]` は起動する認証モジュールです。以下の例は、`login` から `bfagent` にコピーされる項目を示しています。モジュール名はシステムによって異なる場合があることに注意してください。

<code>bfagent auth requisite</code>	<code>pam_authtok_get.so.1</code>
<code>bfagent auth required</code>	<code>pam_dhkeys.so.1</code>
<code>bfagent auth required</code>	<code>pam_unix_cred.so.1</code>
<code>bfagent auth required</code>	<code>pam_unix_auth.so.1</code>

4. PAM 項目をセットアップした後、ステップ 3 の説明に従って再度ログインしてみてください。
5. 詳しくは、PAM の資料 (<http://www.sun.com/software/solaris/pam>) を参照してください。
- `pam.d` に構成されている PAM のトラブルシューティングを行うには、次のようにします。
  1. `/etc/pam.d` ディレクトリーを見つけて、アプリケーションごとに名前の付けられたいくつかのファイルが含まれていることを確認します。各ファイル内の各行のフォーマットは次のとおりです。

`[when] [mode] [module]`
  2. 別のアプリケーション (`sshd` または `login` など) からファイルをコピーして、`bfagent` という名前に変更します。
  3. PAM 項目をセットアップした後、ステップ 3 の説明に従って再度ログインしてみてください。



4. 詳しくは、PAM の資料 (<http://www.sun.com/software/solaris/pam/>) を参照してください。

---

## 第 11 章 インストール後のタスク

このセクションでは、正常にインストールが完了した後に行うタスクについて説明します。

以下のトピックについて説明します。

- エンジンの始動と停止
- ユーザーのセットアップ
- インストール済み環境の検査
- 共通問題のトラブルシューティング

---

### エンジンの始動と停止

以下のセクションでは、エンジンを始動および停止する方法について説明しています。

#### Windows 上でのエンジンの始動と停止

Windows の場合:

- 「スタート」 > 「すべてのプログラム」 > 「IBM Rational Build Forge 管理コンソール (IBM Rational Build Forge Management Console)」で、次のいずれかを選択する必要があります。
  - エンジン・サービスの開始
  - エンジン・サービスの停止

コントロール・パネル: 「管理ツール」 > 「サービス」 コントロール・パネル・オプションを使用しても、「IBM Rational Build Forge 管理コンソール (IBM Rational Build Forge Management Console)」サービスを開始したり停止したりできます。

フォアグラウンドでの実行: エンジンの実行時に問題が発生する場合は、そのエンジンをフォアグラウンドで実行できます。次のような状況とエラーのメッセージが生成されて表示されます: 「スタート」 > 「すべてのプログラム」 > 「IBM Rational Build Forge 管理コンソール (IBM Rational Build Forge Management Console)」 > 「エンジンの開始 (フォアグラウンド) (Start Engine (Foreground))」を実行してください。(Select **Start > Programs > IBM Rational Build Forge Management Console > Start Engine (Foreground)**.) 管理コンソールが実行されている間は、ログ出力がコンソール・ウィンドウに表示されます。(As the Management Console is running, the log output displays in a console window.) このモードでエンジンを停止する場合は、コンソール・ウィンドウで **Ctrl-C** を入力してください。

#### UNIX または Linux システムでのエンジンの開始および停止

rc ファイルがインストールされている場合は、以下のコマンドを使用して製品を開始および停止してください。

```
$ /opt/buildforge/rc/buildforge start
```

```
$ /opt/buildforge/rc/buildforge stop
```

rc ファイルがインストールされていない場合は、以下のコマンドを使用して製品を開始してください。

```
$ /<bfinstall>/Platform/buildforge &
```

プロセス ID を特定し、以下の強制終了コマンドを実行して、エンジンを停止します。

```
$ ps aux | grep buildforge  
$ kill ${<PID>}
```

---

## ユーザーのセットアップ

このセクションでは、Build Forge® システムのユーザー・アカウントのセットアップ方法について説明します。

以下のトピックについて説明します。

- root ユーザー
- ユーザー・アカウントの追加
- レポート用の読み取り専用ユーザー

### root ユーザー

root ユーザーでログインすると、特殊な特権と特性を持つことになります。特権と特性について以下に示します。

- **インストール時に作成:** root ユーザーは、インストール・プログラムが作成する唯一のデフォルト・ユーザーです。デフォルト・パスワードは *root* です。

注: パスワードはインストール直後に変更する必要があります。

- **ライセンス不要:** root ユーザーはユーザー・ライセンスを使用しません。ログイン・ユーザーの数に関係なく、いつでも root ユーザーでログインできます。

注: ログインしようとしたときに、他のユーザーが root ユーザーで現在ログインしている場合、そのユーザーは自動的にログオフされます。

- **システム・タイムゾーン:** root ユーザーのタイムゾーンは、管理コンソールのデフォルトのタイムゾーンになります。デフォルトでは、システム内のユーザーおよび LDAP ユーザーのタイムゾーンは root ユーザーのタイムゾーンと同じです。ユーザーは、ログインした後で独自のタイムゾーンを設定することができます。システムで報告された時刻とログはすべて、ユーザーのタイムゾーンで示されます。
- **すべての権限:** この場合、root ユーザーはすべての有効な権限を持ち、他のユーザー・プロパティを編集することができます。

注: root ユーザーのアクセス権は削除できません。root ユーザーはいずれのアクセス・グループにも属していません。root ユーザーは、システムにあるいずれのデータ・オブジェクトも表示、編集、使用できます。

- **優先順位:** root ユーザーは優先ユーザーです。

- ・ **現行ユーザーをログアウト:** この場合、root ユーザーは、「**ユーザーのログアウト**」をクリックすることにより、ユーザーをログアウトできます。「**管理**」 > 「**ユーザー**」をクリックしてからユーザー名をクリックして、このプロセスを完了します。

## ユーザーの作成および編集

ユーザーを作成してプロパティを割り当てるには、「**管理**」 > 「**ユーザー**」をクリックします。また、システムを LDAP/Active Directory データベースに接続して、ユーザー情報を取得することもできます。ユーザー・セキュリティ権限を管理するには、ユーザーをグループに割り当てます。セキュリティ機能をテストするには、ユーザーをいくつか作成することをお勧めします。

「**管理**」 > 「**ユーザー**」とクリックすると、現行ユーザーのリストが表示されます。リストの後にユーザー・パネルが表示されます。システムは、各ユーザーごとに、「名前」、「ログイン」、「E メール」、「上限」、「アクティビティ」(最後のユーザー・アクティビティからの経過時間)、および「タイムゾーン」を表示します。

- ・ ユーザーを編集する場合: ユーザー名をクリックして、プロパティを編集し、「**保存**」をクリックします。
- ・ 新規ユーザーを作成する場合: ユーザーが選択されていないときは、ユーザー・パネルでプロパティを指定します。ユーザーが選択されているときは、「**ユーザーの追加**」をクリックしてパネルをクリアし、ユーザー情報を編集して「**保存**」をクリックします。
- ・ ユーザーをログアウトする場合: ユーザー名をクリックしてから「**ユーザーのログアウト**」をクリックします。
- ・ 固定ライセンス・シートを使用可能にする場合: **root** でログインします。ユーザー名をクリックしてから「**シートのページ**」をクリックします。コンソールによって、その固定ライセンス・セットに記載された ID のリストからユーザーが除去されます。また、そのユーザーがログインしている場合は、そのユーザーのログオフも行われます。固定ライセンスの場合、コンソールは、これまでにログインしたことのあるユーザーの数をカウントします。上限に達すると、新規ユーザーはライセンスを取得できなくなります。別のユーザーが固定ライセンスを使用できるようにするには、既存のユーザーを削除またはページする必要があります。シートをページしても、ユーザーはコンソールから削除されません。ユーザーが再度ログインすると、固定ライセンスの数は増加します。フローティング・ライセンス・ユーザーで「**シートのページ**」を使用すると、「**ユーザーのログアウト**」をクリックした場合と同じ効果があります。
- ・ ユーザー・アカウントを削除する場合: ユーザー名をクリックしてから「**削除**」をクリックします。

「**削除**」が無効になっている場合は、そのユーザー・アカウントが、スケジュールに入れられたジョブを所有しています。スケジュールに入れられたジョブを持つユーザー・アカウントを削除するには、まず、スケジュールに入れられたジョブを削除する必要があります。

ユーザー・レコードによりデフォルト・プロパティが設定され、ユーザーがシステムと対話する方法が設定されます。また、ユーザー・ログイン名、パスワード、

およびパスワード有効期限も制御されます。ユーザー・レコードのデータは、管理コンソールを使用してシステムに渡したり、LDAP/Active Directory データベースから取り出したりできます。

**注:** LDAP データベースから取り出されたレコードを持つユーザーを編集するときは、「ユーザー」ページ上の多くのフィールドが使用不可になっています。対象となるプロパティは、ソース・データベースで変更できます。

新規ユーザーを追加する場合: 「**ユーザーの追加**」をクリックしてパネルを編集し、「**更新**」をクリックします。

ユーザー・レコードを表示すると、以下の 3 つのタブが表示されます。

- **詳細:** 編集可能なユーザー・プロパティが表示されます。使用可能なプロパティについては、このトピックの後のほうで説明します。
- **現行グループ:** 別のグループのメンバーとなっているグループを使用して、自分がメンバーになっているアクセス・グループが表示されます。
- **グループの変更:** 自分が直接メンバーとして属しているグループが表示されます。グループを追加したり削除したりできます。

「**詳細**」タブでは、ユーザーごとに以下のプロパティを設定することができます。

**名前** ユーザーの表示名およびラベルを指定します。

#### **E メール**

システムがこのユーザーに E メール通知を送信する際に使用する E メール・アドレスを指定します。

**注:** 通知対象として明示的に選択されたユーザーにのみ、E メールが送信されます。

#### **ユーザー名**

ユーザーが管理コンソールにログオンする際に使用する名前を指定します。

#### **パスワード**

ユーザーが管理コンソールにログオンする際に使用するパスワードを指定します。ログオン中のユーザーの場合、このフィールドは表示されません。新規パスワードを入力したり、パスワードを変更したりする場合に、このフィールドを使用してください。「**確認**」フィールドに同じパスワードを入力します。

**上限** ユーザーが 1 日に開始することのできるジョブの最大数を指定します。上限に達すると、システムによって、ユーザーの実行割り当て量を超えたことを示すメッセージが表示されます。

#### **タイムゾーン**

ユーザーのタイムゾーンを指定します。システムは、通知されるすべての時刻のデフォルトのタイムゾーンとして、root ユーザーのタイムゾーンを使用します。

デフォルトでは、システム内のユーザーと LDAP のユーザーに、root ユーザーと同じタイムゾーンが割り当てられます。ユーザーは、割り当てられたタイムゾーンを編集できます。

注: Build Forge を以前のバージョンからバージョン 7.1 にアップグレードする場合、root ユーザーのタイムゾーンを手動でリセットする必要があります。

**確認** パスワードを再入力して、最初に入力したパスワードが正しく入力されていたことを確認します。

#### 優先ログイン

ユーザーが優先ユーザーであるかどうかを指定します。優先ユーザーは常に、システムにログインすることができます。使用可能なユーザー・ライセンスの数が最大数に達している場合、システムは、最も古いセッションを持つユーザーをログアウトさせて、優先ユーザーがログオンできるようにします。root ユーザーは常に優先ユーザーです。

#### 日付書式

ユーザーが希望する日付書式を設定します。

**言語** ユーザーの言語を設定します。

#### パスワード期限切れ

ユーザー・パスワードが期限切れになることを指定します。このオプションが選択されている場合は、所定の日数が経過した後にユーザー・パスワードの有効期限が切れます。この情報は「パスワードの有効期限日数」システム設定に指定されています。

#### スクリーン・リーダーの使用

動的強調表示やフォーカスなどのスクリーン・リーダー機能がインターフェースで使用可能になります。

#### 週のカレンダー開始日

スケジュール・カレンダーで先頭に表示する曜日を選択します。デフォルトは日曜日 です。

---

## インストール済み環境の検査

このセクションでは、インストールおよび構成の完了した Build Forge<sup>®</sup> システムをテストする方法について説明します。

以下のトピックについて説明します。

- サーバーの構成
- テスト・プロジェクトの作成
- プロジェクトの実行

## サーバー認証

サーバー認証を使用して、ログイン資格情報をサーバーに関連付けます。多数のサーバーに対して同じ資格情報を使用し、サーバー認証のセットを管理することで、一括して資格情報を更新することができます。

サーバー認証では、ログイン名とパスワードが単一の名前付きオブジェクトとして保管されます。このオブジェクトは、1 つまたは複数のサーバーに関連付けることができます。サーバー認証を作成および編集するには、「サーバー認証」ページを使用してください。

## サーバーの構成

システム内で動的に選択できるサーバーの数を最大にするには、多数のデータ・オブジェクトを特定の順番でセットアップする必要があります。

このトピックでは、サーバーを使用する際の最小要件について簡単に説明します。

1. サーバー認証を作成します。

サーバー認証によって、サーバーのログイン名およびパスワードが提供されます。複数のサーバーに対して 1 つのサーバー認証を適用できます。

**注:** サーバーごとに固有のログインは必要ありません。ログインを変更することにした場合は、一連のシステムに対してログインを変更できます。

2. サーバー・グループのコレクターを作成します。

コレクターは、割り当てられたプロパティをサーバーから収集します。コレクターがなくてもサーバーを使用できます。サーバーは、*BF\_NAME* などのデフォルト・プロパティに基づいて選択する必要があります。

3. セレクターを作成します。

以下のタイプのセレクターの作成を検討してください。

- 名前ベース: サーバーごとに 1 つのセレクターを作成します。この場合、ホスト名に基づいてサーバーを選択します。サーバーを名前で選択できるようになります。
  - オペレーティング・システム・ベース: ご使用の環境におけるオペレーティング・システムのタイプごとにセレクターを 1 つ作成します。プロジェクトにおいて、オペレーティング・システムでサーバーを選択できるようになります。
  - 容量ベース: 使用可能な RAM またはハード・ディスク・スペースに基づいてサーバーを選択すれば、より具体的な選択を行うことができます。
4. Build Forge サーバーとして使用する予定のコンピューターごとに 1 つのエージェントをインストールします。
  5. 管理コンソールで、Build Forge で使用する予定のコンピューターごとに 1 つのサーバーを作成します。
  6. サーバーをテストします。
  7. 「**接続のテスト**」をクリックして、サーバーとの接続をテストします。

マニフェストを参照して、必要なプロパティが設定されていることを確認します。

## テスト・プロジェクトの作成

Build Forge<sup>®</sup> システムが正常に機能しているか検査するには、12 ページの『Hello World プロジェクトの作成』で説明されているとおりに、単純なプロジェクトを作成して実行してください。

## プロジェクトの実行


プロジェクトの立ち上げには、いくつかの異なる方法があります。



## 始める前に

このタスクは、すでにセレクトアー、サーバー、およびプロジェクトが作成されていることを前提としています。

## 手順

- プロジェクトのリストを表示しているときに、プロジェクトの前にある「クイック・スタート (Quick Start)」アイコン  をクリックすると、そのプロジェクトが即時に開始されます。プロジェクトにステップがない場合や、「変更が必要」プロジェクト動作アクションを使用した環境変数がある場合、この方法は使用できません。この方法でプロジェクトを実行すると、セレクトアー、クラス、タグ、および環境変数にデフォルト値が使用されます。
- プロジェクト・ステップを表示中に、「プロジェクトの開始」をクリックします。この方法を使用すると、プロジェクトの「プロジェクトの開始」ページが表示されます。このページで、プロジェクト・パラメーターおよび環境変数の値を変更したり、実行対象から除外するステップを選択したりできます。
  - プロジェクト・パラメーターの新しい値を選択します。
  - プロジェクトのタグ変数値を編集します。
  - プロジェクトの環境変数値を編集します。これらの変数の新しいデフォルトとして変更内容を保存したい場合は、「環境変数の保存」チェック・ボックスをクリックします。
  - 「ジョブ・ステップ」タブを選択して、プロジェクト・ステップのリストを表示します。個々のステップを選択すると、今回の実行に限り、選択したステップを除外することができます。選択後、「実行」をクリックしてプロジェクトを開始します。
- 「ジョブ」 > 「開始」と選択して、プロジェクト名をクリックします。この方法を使用すると、「プロジェクトの開始」を使用する場合と同様に、「プロジェクトの開始」ページが表示されます。

Job Details

Job Steps

Project Parameters

Project Environment

Save Environment ☐

Snapshot: Base Snapshot

Selector:

Class:

Tag Format:

Tag Example: BUILD\_9

Project Tags

☐ Editable Tags

B

## タスクの結果

プロジェクトの実行中、プロジェクトの状況を確認するには、「ジョブ」 > 「実行中」ページを参照してください。

ジョブの結果を表示するには、「ジョブ」 > 「完了」を選択して、完了したジョブを表示します。ジョブの結果を表示するオプションを使用するには、「タグ名」をクリックしてください。

---

## ログの場所およびセットアップ

Rational Build Forge は、いくつかのコンポーネントで構成されています。コンポーネントはそれぞれ、それ自体のログを持っています。

### Build Forge

<bfinstall>/db.log

### Apache

<bfinstall>/Apache/logs

### Apache Tomcat

<bfinstall>/Apache/tomcat/logs: Tomcat はアプリケーション・サーバーで、サービス層 (jas.war) およびヘルプ (BuildForgeHelp.war) の実行に使用されます。

**PHP** <bfinstall>/db.log: PHP は、ユーザー・インターフェースの実行に使用されます。

## ログの循環

いくつかの条件のもとで、コンポーネントは大量のデータを書き込みます。これは、Tomcat によって作成される Catalina ログ・ファイルにおいて発生する可能性があります。設計によって、Tomcat out ログは循環しません。一般的ではない条件において、out ログは、考えられるサイズよりも大きくなる場合があります。

この問題を回避するには、以下の 2 つの方法があります。

- サーバーのコンテキストで `swallowOutput` 属性を使用します。出力は、循環する Web アプリケーション・ログにリダイレクトされます。詳しくは、『[Apache Tomcat Configuration Reference](#)』を参照してください。
- Catalina 起動スクリプトを変更します。仕様は設定によって異なりますが、次のように行うことで、Catalina 出力がパイプされ、`cronolog` ファシリティーが使用されるため、ログは循環します。<bfinstall>/Apache/tomcat/bin/catalina.sh を変更し、次の (または同様の) コマンドを組み込みます。

```
start >> catalina.out 2>&1 & --replaced with
```

```
start |cronolog /tomcat/logs/%Y-%M-%D.catalina.out >> /dev/null 2>&1 &
```

### 関連概念:

649 ページの『[第 39 章 診断](#)』

このセクションでは、Build Forge が正しく機能しない原因の判別について説明します。

## 共通問題のトラブルシューティング

### Solaris でのエージェントのアップグレードで、pkgrm コマンドの実行が必要になる場合

pkgadd コマンドを実行する前に、pkgrm BFAgent コマンドを使用して、既存の Solaris Build Forge エージェントを除去してください。

### バージョン 7.0 の通知テンプレートの URL が、それより新しいバージョンで機能しない場合

通知 E メール内の通知テンプレート URL リンクをクリックすると、その URL によって Build Forge ジョブ・レポートが表示されます。

バージョン 7.0.1、および 7.1 では、通知テンプレートの URL は変更されています。このため、バージョン 7.0 から、それより新しいバージョンにアップグレードすると、URL が機能しない場合があります。

リンク・エラーが生じる場合、以下のステップを実行して、通知テンプレートを手動で編集してください。

1. 「プロジェクト」>「テンプレート」と選択します。
2. 通知テンプレート名をクリックして、「詳細」タブにそのプロパティを表示します。
3. 「本文」フィールドで、そのテンプレートの URL を見つけます。この URL は、以下の例のようになっているはずです。

```
http://${CONSOLEHOST}:${CONSOLEPORT}/fullcontrol/index.php?mod=projectruns&action=edit&bfid=${PID}&bfid=${BID}&bfid=${UID}
```

4. 以下の URL 要素を、7.0.1 バージョン以降の適切な URL 要素に置き換えます。

URL 要素	7.0.1 (以降の) URL 要素
projectruns	jobs
&amp;	&
action=edit	action=build.view
&bfid=\${PID}   &bfid=\${BID}   &bfid=\${UID}	&bf_id=\${BID}

### 製品が反応しない場合

Rational Build Forge が反応しなくなった場合は、<bfinstall>/Apache/tomcat/logs/ (UNIX または Linux) または <bfinstall>%Apache%tomcat%logs% (Windows) にあるログで、以下のようなメッセージを確認してください。

```
Services: 20075: CRRBF1381I: Established connection to Build Forge Services.  
DBD::DB2::st execute failed: [IBM][CLI Driver] SQL30081N A communication error has been detected.  
Communication protocol being used: "TCP/IP".  
Communication API being used: "SOCKETS".  
Location where the error was detected: "XXX.XXX.XXX.XXX".  
Communication function detecting the error: "recv".  
Protocol specific error code(s): "131", "*", "0". SQLSTATE=08001  
August 17, 2010 7:18:14 AM EDT
```

```

Database: 20075: CRRBFEEEE: DBD::DB2::st execute failed:
[IBM][CLI Driver] SQL30081N A communication error has been detected.
Communication protocol being used: "TCP/IP".
Communication API being used: "SOCKETS".
Location where the error was detected: "XXX.XXX.XXX.XXX".
Communication function detecting the error: "recv".
Protocol specific error code(s): "131", "*", "0". SQLSTATE=08001

Database: 20075: CRRBF0551I: StackTrace from Process id [20075] called from [
BuildForge::DB::db2(/PerlApp/BuildForge/DB/db2.pm:65)
BuildForge::Utilities::SysParams(/PerlApp/BuildForge/Utilities/SysParams.pm:57)
main(.buildforge.pl:305)
main(.buildforge.pl:213)
]
Database: 20075: CRRBF0556I: Trying to call [execute] on a non-existent database handle
DBD::DB2::st fetchrow_hashref failed: no statement executing at /PerlApp/BuildForge/DB/db2.pm
line 78, <$sock> line 1054.
20075: CRRBF0555E: Problem performing Database Operation [fetchrow_hashref] : DBD::DB2::st
fetchrow_hashref failed: no statement executing at /PerlApp/BuildForge/DB/db2.pm
line 78, <$sock> line 1054.

Database: 20075: CRRBFEEEE: DBD::DB2::st fetchrow_hashref failed: no statement executing
at /PerlApp/BuildForge/DB/db2.pm line 78, <$sock> line 1054.

Database: 20075: CRRBF0551I: StackTrace from Process id [20075] called from [
BuildForge::Utilities::SysParams(/PerlApp/BuildForge/Utilities/SysParams.pm:59)
main(.buildforge.pl:305)
main(.buildforge.pl:213)
]
[IBM][CLI Driver] CLI0106E Connection is closed. SQLSTATE=08003 at /PerlApp/BuildForge/DB/Handle.pm
line 385, <$sock> line 1054.
Database: 20075: CRRBF0555E: Problem performing Database Operation [prepare] :
query [SELECT * FROM bf_jobcount WHERE bf_engine_id=?]
DBD::DB2::db prepare failed: [IBM][CLI Driver] CLI0106E Connection is closed.
SQLSTATE=08003 at /PerlApp/BuildForge/DB/Handle.pm line 385, <$sock> line 1054.

Database: 20075: CRRBFEEEE: DBD::DB2::db prepare failed: [IBM][CLI Driver] CLI0106E
Connection is closed. SQLSTATE=08003 at /PerlApp/BuildForge/DB/Handle.pm
line 385, <$sock> line 1054.

```

XXX.XXX.XXX.XXX はデータベースをホストするサーバーです。

このメッセージは、Rational Build Forge とデータベースの間に接続がないことを示しています。

接続を再度確立するには、次のようにします。

1. 289 ページの『Perl エンジンの始動と停止』の説明に従って、Rational Build Forge をシャットダウンします。
2. データベースが実行中であることを確認します。
3. 289 ページの『Perl エンジンの始動と停止』の説明に従って、Rational Build Forge を起動します。

---

## 第 12 章 以前のバージョンからのアップグレード

以前のバージョンから現行バージョンにコンポーネントをアップグレードする方法について、以下の各トピックで説明します。

- アップグレードの前提条件

**重要:** 前提条件が守られなかった場合には、システムが損傷したり使用できなくなったりするおそれがあります。

- 管理コンソールのアップグレード
  - バージョン 7.1.x コンソールのアップグレード
  - バージョン 7.0.2.x コンソールのアップグレード
- エージェントのアップグレード

---

### アップグレードの前提条件

アップグレード・インストールを実行する前に、以下を行う必要があります。

- データベースをバックアップしてください。 39 ページの『データベースのバックアップ (必須)』も参照してください。
- データベースおよびデータベース・クライアントのバージョンをアップグレードする予定の場合は、そちらのアップグレードを先に実行してください。新規バージョンを使用するように、インストール済みのシステムを変更することはできません。システムでは、インストーラーで指定されたデータベースおよびデータベース・クライアントのファイルが使用されます。
- Apache および Tomcat のカスタマイズされた構成ファイルがあれば、そのファイルをバックアップしてください。

```
<bfinstall>/Apache/conf  
<bfinstall>/Apache/tomcat/conf
```

通常、httpd.conf ファイルおよび php.ini ファイルは Apache 用にカスタマイズされています。

- HTTPS と SSL を使用するようにシステムが構成されている場合は、証明書ディレクトリーをバックアップしてください。アップグレードによって、これらのファイルに影響が及ぶことはありません。

---

### バージョン 7.1 コンソールのアップグレード

このセクションは、7.1.x バージョンから現在のバージョンにアップグレードを行う場合にお読みください。

アップグレードを開始する前に、『アップグレードの前提条件』を確認してください。

アップグレードする場合は、更新インストールを実行してください。

注: バージョン 7.1 コンソールをアップグレードする場合は、パフォーマンス強化を有効にするために、追加の作業を実行する必要があります。詳しくは、jazz.net のリリース・ページを参照してください。

## 更新インストールの実行

このタスクでは、更新インストールの実行方法について説明します。

### 始める前に

- Build Forge が実行中の場合は、ここで停止します。
- Windows プラットフォームで作業を行っている場合は、インストール・ディレクトリーを調べて `bfengine.pid` ファイルを確認します。存在する場合は、それを除去します。ファイルがあると、Installation Manager は更新を実行できません。
- 更新を行う前に、リポジトリ URL として更新済みリポジトリの URL を指定する必要があります。手順については、87 ページの『リポジトリ URL の指定』を参照してください。

### このタスクについて

Build Forge® 8.0 では、バージョン 7.1.1.3 以降からのアップグレードのみをサポートします。7.1.1.3 より前のバージョンの場合、まず 7.1.1.3 にアップグレードしてから 8.0 にアップグレードします。

### 手順

1. IBM Installation Manager を開始します。
2. Installation Manager で、「更新」をクリックします。
3. 「パッケージの更新 (Update Packages)」パネルで、更新を検索するパッケージ・グループを選択し、「次へ」をクリックします。

注: 推奨される更新パッケージのみを表示するには、「推奨のみを表示 (Show recommended only)」チェック・ボックスを選択します。

4. インストールする更新または修正を選択します。
  - a. リストからインストールを選択します。
  - b. 「次へ」をクリックします。更新可能なバージョンが表示されます。
5. バージョンのリストからインストールを選択し、「次へ」をクリックします。
6. 「使用条件の条項に同意します (I accept the license terms)」チェック・ボックスを選択し、「次へ」をクリックします。「パッケージの更新 (Update Packages)」ページが表示され、ご使用の機能が自動的に選択されます。
7. 「次へ」をクリックします。
8. 構成をテストするには、JAS のデータベース接続を構成し、外面の下部付近にある「接続のテスト」をクリックします。テストに合格したら、「次へ」をクリックします。
9. データベース・クライアント・ライブラリーを構成して、「接続のテスト」をクリックします。テストに合格したら、「次へ」をクリックします。

`buildforge.conf` ファイルで編集した情報が表示されます。

10. 「パッケージの更新 (Update Packages)」の「コンソール開始オプション」パネルで、以下のプロンプトに情報を入力します。
- 「JAS データベース構成」パネルで、以下を実行します。
    - a. 「データベース・タイプ」フィールドに、更新するデータベースのタイプを入力します。
    - b. プロンプト「インストール時にこのデータベースにデータを取り込みますか?」で、「はい」または「いいえ」のいずれかのチェック・ボックスを選択します。
    - c. 「データベース・ホスト」フィールドに、ホストの IP アドレスを入力します。
    - d. 「データベース・ポート」フィールドに、このデータベース接続に使用するポートを入力します。デフォルトは **3306** です。
    - e. 「データベース・ユーザー名」フィールドで、データベースへのログオン時に使用するユーザー名を入力します。
    - f. 「パスワード」フィールドに、データベースへのログオンに必要なパスワードを入力します。
    - g. 「パスワードの確認」フィールドで、パスワードを再入力します。
    - h. 「JDBC ドライバー・ロケーション」フィールドで「参照」をクリックし、JDBC ドライバーのロケーションを探して選択します。必須ドライバ JAR ファイルおよび必須 JDBC ドライバー・クラスが表示されます。
    - i. 「次へ」をクリックします。「Update Packages (パッケージの更新)」の「コンソール開始オプション」が再び表示されます。
  - 「コンソール開始オプション」パネル
    - a. 「データベース名」フィールドに、データベース名を入力します。**bfdotcmd** がデフォルトです。
    - b. 「パスワード」フィールドに、データベースへのログオンに必要なパスワードを入力します。
    - c. 「パスワードの確認」フィールドで、パスワードを再入力します。
    - d. 「JDBC ドライバー・ロケーション」フィールドで「参照」をクリックし、JDBC ドライバーのロケーションを探して選択します。必須ドライバ JAR ファイルおよび必須 JDBC ドライバー・クラスが表示されます。
    - e. データベース構成の接続をテストするには、「接続のテスト」をクリックします。
    - f. テストに合格したら、「次へ」をクリックします。
  - 「データベース・クライアントライブラリー構成情報 (Database Client Library Configuration Information)」パネルで、以下のフィールドに入力します。
    - a. 「SQL クライアント・ライブラリーへのパス (Path to the My SQL Client Libraries)」フィールドで「参照」をクリックして、ご使用のSQL クライアント・ライブラリーへのパスを検索して選択します。
    - b. データベース接続を再度テストするには、「接続のテスト」をクリックします。接続のテストの結果が表示されます。



- c. 「次へ」をクリックします。
11. 以下のプロンプトに情報を入力して、更新を続行します。
- a. UNIX および Linux – プロンプト「**Build Forge を稼働させるユーザーはどれですか? (Which user should Build Forge run as?)**」で、ユーザー名をするか、デフォルトのユーザー (root) を受け入れます。このユーザーが、Build Forge エンジンと、指定された Apache Tomcat を開始します。このユーザーには、Installation Manager でこの後表示されるページで指定されるデータベース・ライブラリーと JDBC JAR ファイルの読み取り権限および実行権限が必要です。
  - b. インストーラーがスキーマに対して必要な変更を行うようにするには、プロンプト「**必要なデータベース変更をインストーラーに行わせますか?**」でこのチェック・ボックスを選択します。このチェック・ボックスを選択しない場合は、インストール後に、スキーマの変更を適用するために `bfschema -u` を手動で実行する必要があります。
  - c. インストールの完了後にコンソールが自動的に開始するようにするには、プロンプト「**アップグレード後にコンソールを開始しますか?**」でこのチェック・ボックスを選択します。
  - d. 「次へ」をクリックします。 インストールすることを選択したフィーチャーがリストされます。
12. 「更新」をクリックすると、インストールが開始されます。
13. 更新が正常に終了したら、「完了」をクリックします。
14. Installation Manager を閉じます。
15. Build Forge を開始します。 バージョン 7.1.1.x からアップグレードする場合は、Character Large Object (CLOB) のマイグレーションが完了するまで、ビルドのステップ・ログは表示されません。Build Forge を開始すると、マイグレーションが開始されます。マイグレーションが完了すると、Build Forge から通知が届きます。

## バージョン 7.1.3.1 へのアップグレード後、アダプターをコピーする

7.1.1.x から 7.1.3.1 以降にアップグレードした後でアダプターをコピーすると問題が発生します。いくつかのアダプターは、バージョン 7.1.3.1 用に更新されたものです。バージョン 7.1.3.1 にアップグレードした後、このアップグレードの前に作成されたアダプターをコピーしようと試みると、次のエラーが報告されます。

CRRBF0789I: A valid Adaptor Template Identifier must be specified.

回避策: アダプターをコピーしようとする前に、以下の手順を実行してください。

1. アダプターを開きます。
2. アダプターの名前を変更します。
3. アダプターを保存します。

また、アダプターを開き、その内容をコピーして、新しいアダプターを作成することもできます。

---

## バージョン 7.0.2.x コンソールのアップグレード

このセクションは、バージョン 7.0.2.x システムを最新バージョンにアップグレードする場合に使用します。

アップグレードを行うには、以下の手順を実行します。

1. 223 ページの『アップグレードの前提条件』を確認します。
2. バージョン 7.0.2.x からバージョン 7.1.1.3 にアップグレードします。

IBM サポート・サイトからバージョン 7.1.1.3 のパッケージをダウンロードして、「*IBM Rational Build Forge* インストール・ガイド バージョン 7.1.1.3」にある説明に従ってください。データベース・スキーマの変更により、バージョン 7.0.2.x からバージョン 7.1.1.3 にアップグレードするには、構成データ（プロジェクト）をマイグレーションする必要があります。

**注:** 履歴データ（ジョブからのログ）もマイグレーションできます。

**重要:** すべての項目がマイグレーションされるわけではありません。特に、カスタマイズされた通知テンプレート項目（「プロジェクト」 > 「テンプレート」）はマイグレーションされません。

3. バージョン 7.1.1.3 で更新インストールを実行して、それを現在のバージョンに更新します。

**注:** バージョン 7.1 コンソールをアップグレードする場合は、パフォーマンス強化を有効にするために、追加の作業を実行する必要があります。詳しくは、[jazz.net](http://jazz.net) のリリース・ページを参照してください。

---

## エージェントのアップグレード

新しいエージェントを古いエージェントに上書きインストールすれば、エージェントをアップグレードできます。

エージェントをアップグレードするには、新しいエージェントを古いエージェントに上書きインストールします。次のインストール手順に従って、エージェントをインストールしてください。157 ページの『第 10 章 エージェントのインストール』を参照してください。



# 第 13 章 製品コンポーネントのアンインストール

IBM Installation Manager を使用してインストールした製品コンポーネントは、Installation Manager を使用してアンインストールします。Build Forge エージェント・ソフトウェアをアンインストールするには、このセクションに記載されているオペレーティング・システム・ツールおよびコマンドを使用してください。

## Installation Manager を使用した製品のアンインストール

このシナリオでは、Installation Manager を使用してインストールした Build Forge 製品コンポーネントをアンインストールします。

注: この手順では、Build Forge に対して作成されたデータベースまたはデータベース・オブジェクトはアンインストールされません。これらは、データベースのツールを使用して個別に削除する必要があります。Windows にインストールされたデータベースの場合は、「コントロール パネル」 > 「プログラムの追加と削除」を使用できる場合があります。操作手順については、データベースの資料を参照してください。データベースまたはデータベース・オブジェクトの削除は、Build Forge 製品のコンポーネントを削除した後に 行ってください。

製品コンポーネントをアンインストールするには、以下の手順を実行します。

- 1. 製品パッケージのインストール時に使用したユーザー・アカウントと同じユーザー・アカウントを使用して、オペレーティング・システムにログインします。

Installation Manager を使用してインストールした、実行中のプログラムを停止します。

- 2. Build Forge エンジンおよびサービスを実行中の場合は、以下のようにして停止します。

Windows	<ul style="list-style-type: none"><li>• エンジンをフォアグラウンドで実行中の場合: <b>Ctrl + c</b> を押して、エンジンおよび Build Forge サービスを停止します。</li></ul> <p><b>重要:</b> エンジンおよびサービスを停止する最も確実な方法は、<b>Ctrl + c</b> を使用する方法です。アンインストール時に、エンジンまたはいずれかの Build Forge サービスがまだ実行されていると、アンインストールは失敗します。</p> <ul style="list-style-type: none"><li>• エンジンをバックグラウンドで実行中の場合: コマンド・ウィンドウを開いて、以下のコマンドを入力し、エンジンおよびサービスを停止します。</li></ul> <pre>net stop bfengine&lt;major number&gt;&lt;Fix Pack Number&gt;-&lt;Build Number&gt;</pre> <p>例: net stop bfengine 8.0.0.0-0-0308</p> <p>これらのコマンドは、Build Forge エンジン、Apache HTTP サーバー、および Apache Tomcat アプリケーション・サーバーを停止します。</p>
---------	---

UNIX/Linux	<ol style="list-style-type: none"> <li>1. コマンド・ウィンドウで、Build Forge rc ディレクトリーに移動します。 <code>cd /opt/buildforge/rc</code></li> <li>2. エンジンを停止します。 <code>./buildforge stop</code></li> </ol> <p>このコマンドは、Build Forge エンジン、Apache HTTP サーバー、および Apache Tomcat アプリケーション・サーバーを停止します。</p>
------------	--

3. IBM Installation Manager を開始します。
4. 「開始」ページで、「アンインストール」をクリックします。
5. 「パッケージのアンインストール」ページの「インストール・パッケージ」リストで、アンインストールする製品パッケージを選択して、「次へ」をクリックします。
6. 「要約」ページで、アンインストールするパッケージのリストを確認し、「アンインストール」をクリックします。パッケージが除去された後、「完了 (Complete)」ページが表示されます。
7. 「完了」をクリックします。
8. Installation Manager ウィンドウを閉じて、Installation Manager を終了します。

Build Forge ファイルのクリーンアップ、DB2 Express データベースの除去 (オプション)、または製品の再インストールを行う前に、Installation Manager を終了する必要があります。

## Installation Manager によるインストールが失敗した場合の製品の手動アンインストール

Build Forge エンジンおよびサービスが停止する前に IBM Installation Manager を開始すると、アンインストールは失敗します。

Installation Manager が失敗した後、製品コンポーネントを手動でアンインストールするには、次のようにします。

1. コマンド・ウィンドウで、Build Forge インストール・ディレクトリー内の Manager ディレクトリーに移動します。例えば、次のようなディレクトリーになります。

Windows	C:\Program Files\IBM\Build Forge\Manager
UNIX/Linux	/opt/buildforge/Manager

2. コマンド・プロンプトで、`main.exe uninstall main.res` と入力します。
3. コマンドの実行が完了したら、Build Forge インストール・ディレクトリーを削除します。
4. Installation Manager を使用して製品を再インストールします。

## アンインストール後の Build Forge ファイルのクリーンアップ

アンインストールの後で、若干の手動クリーンアップが必要です。

Installation Manager を使用して製品コンポーネントをアンインストールした後、以下のディレクトリーを手動で削除します。これらのディレクトリーが存在していると、Installation Manager を使用して Build Forge を再インストールできません。

ディレクトリー	デフォルトの場所
Build Forge インストール・ディレクトリー	<b>Windows:</b> C:\Program Files\IBM\Build Forge <b>UNIX/Linux:</b> /opt/buildforge
共用ファイル・ディレクトリー	<b>Windows:</b> C:\Program Files\IBM\SDPShared <b>UNIX/Linux:</b> /opt/IBM/SDPShared

## Installation Manager を使用した製品の再インストールおよび既存の DB2 Express の使用

他の製品コンポーネントのアンインストール時に DB2 Express データベースを除去しなかった場合は、製品を再インストールして、既存の DB2 Express データベースを使用することができます。このシナリオでは、既存の Build Forge プロジェクトおよびジョブ・ログにアクセスできるようにします。

Installation Manager を開始する前に、以下の手順でチェックリストを使用して、必要な DB2 Express 情報を入手します。

**重要:** HTTPS と SSL を使用するようにシステムが構成されている場合は、`<bfinstall>/keystore` および `<bfinstall>/bfclient.conf` (証明書のパスワードが保管されている場所) にある証明書をバックアップする必要があります。UNIX システムおよび Linux システムの場合、これらのディレクトリーは `<bfinstall>/Platform` にあります。

再インストールして、既存の DB2 Express データベースを使用するには、次のようにします。

1. Installation Manager を開始します。
2. 「開始」ページで、「インストール」をクリックします。
3. Installation Manager ウィザードの指示に従って、製品コンポーネントを再インストールします。
4. 「データベース構成」ページで、以下を選択します。
  - a. 「DB2 Express をインストールしますか?」で「いいえ」を選択します。
  - b. 「インストール時にこのデータベースにデータを取り込みますか?」で「いいえ」を選択します。
  - c. 次の表の情報を使用して、残りのフィールドに入力します。

### チェックリスト: DB2 Express 構成情報

✓	フィールド	説明
	データベース・ホスト	製品提供の DB2 Express データベースは、ローカル・ホスト名 (127.0.0.1) にインストールされています。
	データベース名	データベース名は BUILD (すべて大文字) です。

✓	フィールド	説明
	データベース・スキーマ名	データベース・スキーマの名前は BUILD (すべて大文字) です。
	データベース・ユーザー名	DB2 Express データベースの作成時に指定したデータベース・ユーザー名。
	データベース・パスワード	データベース・ユーザー名に対して指定したパスワード。
	DB2 クライアント・ライブラリーへのパス	DB2 Express 用の DB2 クライアント・ライブラリー (db2cli.dll) へのパスは、C:\Program Files\IBM\SQLLIB\bin です。
	JDBC ドライバー・ロケーション	DB2 Express 用の JDBC ドライバー (db2jcc.jar) へのパスは、C:\Program Files\IBM\SQLLIB\java です。

5. 「接続のテスト」をクリックします。
6. 「次へ」をクリックして続行し、インストールを完了します。

## Windows Build Forge エージェントのアンインストール

Build Forge エージェント・ソフトウェアを除去するには、Windows の「プログラムの追加と削除」ツールを使用します。

注: 再インストールする場合は、同じバージョンのエージェント・ソフトウェアを上書きできます。

1. Windows で、「プログラムの追加と削除」を開きます。例えば、「スタート」→「すべてのプログラム」→「コントロール パネル」→「プログラムの追加と削除」の順に選択します。
2. 現在インストールされているプログラムのリストで、IBM Rational Build Forge エージェントを見つけます。
3. 「プログラムの変更と削除」をクリックします。
4. 指示に従って、エージェントをアンインストールします。

## UNIX または Linux の Build Forge エージェントのアンインストール

UNIX または Linux プラットフォームからエージェント・ソフトウェアをアンインストールする場合は、以下の説明に従ってください。

### Linux のエージェント

rpm パッケージを使用してインストールされたエージェント・ソフトウェアを除去するには、次のようにします。

1. エージェント・ソフトウェアを検索して、パッケージ名およびバージョンをリストします。  

```
rpm -qa | grep bfagent
```
2. エージェント・ソフトウェアを削除します。  

```
rpm -e bfagent-<version_number>
```



## Solaris のエージェント

pkgadd プログラムを使用してインストールされたエージェント・ソフトウェアを除去するには、次のようにします。

pkgrm BFAgent

## その他のエージェント

その他のプラットフォームの場合、アンインストール処理は手動で行われ、プラットフォームによって異なります。プラットフォームおよびスーパー・サーバーのインプリメンテーションに適用される説明に従ってください。

**注:** ほとんどのコマンドの実行には、root 特権が必要で、/sbin および /usr/sbin ディレクトリーをユーザーの現行の PATH 環境変数に設定しておく必要があります。

1. エージェント・サービス・デーモン bfaagent を除去します。ご使用のプラットフォームに適用されるスーパー・サーバー・インプリメンテーション (inetd、xinetd、launchd、または SMF) の説明をお読みください。

スーパー・サーバー	手順
inetd (古い UNIX システムで一般的)	<ol style="list-style-type: none"><li>1. /etc/inetd.conf ファイルを編集して、bfaagent の行を削除します。</li><li>2. inetd のプロセス ID を検索します。  ps -ef   grep [i]netd  FreeBSD や Mac OS/X 10.4 以前などの BSD 派生システムの場合は、ps -ef の代わりに ps auwwwx を使用してください。</li><li>3. 更新された inetd.conf を確認して、inetd を開始します。  kill -HUP &lt;PID&gt;</li></ol>
xinetd (新しい UNIX システムで一般的)	<ol style="list-style-type: none"><li>1. エージェント・サービスを除去するには、以下のコマンドを実行します。  rm /etc/xinetd.d/bfaagent</li><li>2. inetd のプロセス ID を検索します。  ps -ef   grep [i]netd  FreeBSD や Mac OS/X 10.4 以前などの BSD 派生システムの場合は、ps -ef の代わりに ps auwwwx を使用してください。</li><li>3. 更新された inetd.conf を確認して、inetd を開始します。  kill -HUP &lt;PID&gt;</li></ol>

スーパー・サーバー	手順
launchd (Mac OS/X および OpenBSD システム)	<ol style="list-style-type: none"> <li>1. launchctl を実行します。</li> <li>2. stop com.ibm.rational.bfagent と入力します。</li> <li>3. 次のコマンドを入力します: unload /Library/LaunchDaemons/com.ibm.rational.bfagent.plist。</li> <li>4. quit と入力します。</li> <li>5. 次のコマンドを実行します: rm Library/LaunchDaemons/com.ibm.rational.bfagent.plist。</li> </ol>
Solaris システム管理機能 (SMF) (Solaris 10)	<ol style="list-style-type: none"> <li>1. inetadm -d network /bfagent/tcp を実行します。</li> <li>2. svccfg delete -f network/bfagent/tcp を実行します。</li> </ol>

2. エージェント・サービスを PAM インターフェースから除去します。
  - a. /etc/pam.conf を編集して、bfagent で始まるすべての行を削除します。
  - b. rm /etc/pam.d/bfagent を実行します。
3. etc/services ファイルからプロトコル項目を削除します。  
/etc/services を編集して、 bfagent の行を削除します。
4. エージェントによってインストールされた以下のファイルを削除します。  
/etc/bfagent.conf  
/etc/bfagent.conf-example  
/usr/local/bin/bfagent  
/usr/local/bin/bfcrypt.dll

---

## 第 14 章 リリース・ノート

このセクションには、製品の各リリースに対するリリース・ノートが含まれています。リリース間でインフォメーション・センターが更新された場所を記録する項目が含まれていることもあります。リリース・ノートには、以下の内容が含まれています。

- そのリリースの新規機能および変更された機能のリスト。
- 要件、インストールの説明、サポート、および他の製品情報への参照情報。

---

### リリース・ノート - *IBM Rational Build Forge* バージョン 8.0.0.2

*Rational Build Forge* バージョン 8.0.0.2 が使用可能です。互換性、インストール、およびその他の開始の際に生じる問題について説明しています。

© Copyright International Business Machines Corporation 2003, 2014. All rights reserved.

#### 内容

- 『説明』
- 『新規機能および変更された機能』
- 236 ページの『このリリースで修正された APAR』
- 236 ページの『告知』
- 236 ページの『旧バージョンとの互換性の問題』
- 236 ページの『システム要件』
- 236 ページの『インストール』
- 236 ページの『既知の問題』

#### 説明

*IBM® Rational® Build Forge®* は、ソフトウェア開発のアセンブリー・ライン内の各ハンドオフ間のすべてのプロセスを自動化、編成、管理、追跡し、自動化されたソフトウェア・ファクトリーを作成する、適応力のあるプロセス実行フレームワークです。IBM Rational Build Forge は、現在の環境に統合され、主要な開発言語、スクリプト、ツール、およびプラットフォームをサポートします。既存の投資資産を引き続き使用しながら、プロセスの自動化、促進、通知、およびスケジューリングに関する便利な機能を追加することができます。

#### 新規機能および変更された機能

このリリースでの新規機能および変更された機能を以下のリストに示します。機能の最新のリストについては、Jazz.net (アカウントが必要) 上の GA リリースの「New & Noteworthy」ページを参照してください。

## このリリースで修正された APAR

このリリースで修正された APAR の最新のリストについては、Jazz.net (アカウントが必要) 上の GA リリースの「Release Notes」ページを参照してください。

## 告知

このバージョンのリリース告知は、[www.ibm.com/common/ssi/index.wss](http://www.ibm.com/common/ssi/index.wss) で行われます。以下の情報についての告知を参照してください。

- 新規機能の説明を含む詳細な製品説明
- ハードウェア要件およびソフトウェア要件
- ソフトウェア・サービスおよび利用できるサポート
- パッケージ化および順序付けの詳細
- 国際的な互換性情報

## 旧バージョンとの互換性の問題

バージョン 8.0 ではデータベース表アーキテクチャーが変更されました。Build Forge 構成データおよび履歴データを古いデータベース表アーキテクチャーからマイグレーションする必要があります。

バージョン 8.0 にアップグレードするための要件は、ご使用の現行バージョンによって異なります。

- 7.1.0.x、7.1.1.x、7.1.2.x: これらのいずれかのバージョンが稼働している場合は、Installation Manager の「更新」機能を使用してアップグレードできます。
- 7.0.2.x: これらのいずれかのバージョンが稼働している場合は、まずバージョン 7.1.1.3 にアップグレードしてから、バージョン 8.0 にアップグレードする必要があります。

## システム要件

ハードウェアおよびソフトウェア要件は、照会によるレポートとして、「Detailed system requirements for a specific product」サイトで入手できます。ご使用の製品、製品エディション、バージョン、およびオペレーティング・システムを指定してください。ハードウェア、オペレーティング・システム、データベース、エージェント・ホスト、および Web クライアントに関する要件が記載されています。

## インストール

インストールに関するステップバイステップの説明については、23 ページの『第 4 章 要件』のすぐ後に続く各トピックを参照してください。

## 既知の問題

既知の問題は、[http://www.ibm.com/support/entry/portal/overview//software/rational/rational\\_brand\\_support\\_\(general\)](http://www.ibm.com/support/entry/portal/overview//software/rational/rational_brand_support_(general)) のサポート知識ベースに個別の技術情報の形式で文書化されています。問題が見つかり、解決すると、IBM サポート・チームが知識ベースを更新します。知識ベースを検索することにより、回避策や解決策を手早く見つけることができます。

- 報告された問題、Rational Build Forge バージョン 8.0

<http://www.ibm.com/support/search.wss?rs=3099&tc=SS2MGB&q=RN8.0&rankprofile=8>


- 解決した問題、Rational Build Forge バージョン 8.0

<http://www.ibm.com/support/search.wss?rs=3099&tc=SS2MGB&q=RN8.0C&rankprofile=8>

「IBM Rational Build Forge Support サイト」で、照会を実行し、以下のようなタイプの情報をさらに要求することもできます。

- ダウンロード
- フラッシュ (アラート)
- ニュース
- 製品情報および資料
- トラブルシューティング (これらの URL が提供している情報より広い範囲)

#### 関連情報:

 [IBM サポートへのお問い合わせ](#)



---

## 第 15 章 ダッシュボード・タブ

ダッシュボード・タブには、システム状況とアクティビティの概要が表示されます。

### 成功ジョブ

過去 4 週間に成功または失敗したジョブの数 (週ごとに表示)。すべてのプロジェクトまたは単一プロジェクトの傾向を表示するように選択できます。

### スケジュール

今後実行する予定のジョブ (スケジュールしたランタイム別にリスト)。リストの表示方法 (すべて表示するか限られた数だけ表示するか)、およびリストのページ送りの方法を選択できます。

### サーバー状態

サーバーとその状況 (「使用可能 (Available)」または「停止 (Down)」)。すべてのサーバーの状況、または選択したコレクターに関連付けられているサーバーのみの状況を表示するように選択できます。リストの表示方法 (すべて表示するか限られた数だけ表示するか)、およびリストのページ送りの方法を選択できます。

### ジョブ・キュー

実行中のジョブと待機中のジョブの数。グラフは、一定の時間が経過すると自動的に更新されます。「最大同時ジョブ数 (Max Simultaneous Jobs)」システム設定値は、グラフにしきい値として表示されます。

### キューのページ

実行中のページと待機中のページの数。グラフは、一定の時間が経過すると自動的に更新されます。「最大同時ページ数」システム設定値は、グラフにしきい値として表示されます。

### ライセンス・シート

使用中のライセンス・シートの数。これは指標ダイヤルに表示され、キャプションには「使用されているライセンス・シートの数: (X)/(総数)」と示されます。

注: Internet Explorer 7 および 8 では、ブラウザーでの制約により、このダッシュボードに数値は表示されません。





## 第 16 章 管理

このトピックでは、Build Forge® システムの管理操作について説明します。

### 管理について

管理パネルを使用して、構成と設定を管理します。

管理パネル内では、ユーザー、セキュリティー特権、通知設定、およびシステム設定を処理して、管理コンソールを構成することができます。

「管理」パネルにアクセスするには、左側のメニューで「管理」をクリックします。

UI Config Console Reports Logout: Root User

System Help ?

Filter Showing 1 - 72 of 72 Auto Paginate << < Page 1 of 1 > >>

Name	Value
Alert Email Limiting	10/30
Apply inlined steps container environment	No
Apply server environment last	No
Auto-Logoff Minutes	0
AutoClean Audit Log Days	365

Alert Email Limiting Save Revert to the Default

Details

Alert Email Limiting: 10/30

Enter the value for;the maximum number of alert email messages per time in minutes (messages/minutes). 0/0 for unlimited.

Default Value: 10/30

### アクセス・グループ

アクセス・グループは、システムが権限の制御に使用するユーザーの集合です。

「管理」 > 「アクセス・グループ」パネルを使用して、新規アクセス・グループの作成、ユーザーの追加または除去、およびグループ・プロパティーの変更を行います。このパネルには、既存のアクセス・グループのリストが表示されます。グループの名前をクリックして選択すると、パネルの下部にそのプロパティーが表示され

ます。

Access Group Name	New User Default	LDAP Group DN
<a href="#">Build Engineer</a>	No	
<a href="#">Developer</a>	Yes	*
<a href="#">Guest</a>	Yes	*
<a href="#">Operator</a>	Yes	*

**New Group** Save Copy Delete

Details Users Subgroups Permissions

☐ Default    Name:     Owner: Security

LDAP Group DN:

- 新規グループを作成するには、「グループの追加 (Add Group)」をクリックして、パネルの下部のフィールドをクリアします (必要な場合)。次に、グループに「名前」を付けて、「所有者」グループを選択します。(「所有者」グループにより、この新規グループへのアクセスが制御されます。グループを編集するには、ユーザーは「所有者」グループのメンバーでなければなりません)。LDAP 認証を使用している場合、ご使用のグループにマップする LDAP グループを、LDAP グループ DN フィールドに指定します。例えば、Developer グループを SoftwareEngineers という LDAP グループにマップし、さらに Developer グループに必要な権限を割り当てることにより、ソフトウェア・エンジニアが必要とするタイプのアクセス権を付与することができます。
- LDAP グループ DN フィールドに、このアクセス・グループに関連付けられた管理コンソール・セキュリティ特権を付与したいメンバーを持つすべての LDAP グループの識別名をリストします。
- どのアクセス・グループにも、複数の LDAP グループをマップすることができます。このプロパティにアスタリスク (\*) 文字を指定して、すべての LDAP ユーザーにこのアクセス・グループのメンバーシップを提供することができます。複数の LDAP グループをリストし、それらをセミコロンで区切ることができます。

- アクセス・グループを削除するには、グループを選択して、「**削除**」ボタンをクリックします。

**注:** 他の場所で「**アクセス**」プロパティとして割り当てられているアクセス・グループを削除することはできません。例えば、アクセス・グループを作成し、そのグループを使用するようにプロジェクトの「**アクセス**」プロパティを設定した場合、そのグループを削除することはできません。まず、別のアクセス・グループを使用するようにプロジェクトを編集する必要があります。

- ユーザーを追加または除去するには、グループを選択してから、「**ユーザー**」タブをクリックします。システムには、左に非メンバー、右にメンバーのリストが表示されます。ユーザーを選択し、「**追加**」ボタンと「**除去 (Remove)**」を使用して、ユーザーを 1 つのリストから別のリストへ移動します。

**注:** 「アクセス・グループ・マッピングのマップ (Map Access Group Mapping)」を「はい」に設定した LDAP ユーザーは、「**ユーザー**」タブに表示されません。この動作により、Build Forge の手動設定と矛盾することなく、「アクセス・グループ・マッピングのマップ (Map Access Group Mapping)」の設定によって実際にグループ・マッピングが実行されます。

- グループをネストするには、グループを別のグループのサブグループとして追加します。これを行う場合、収容グループに適用されるすべての権限は、メンバー・グループ内のすべてのユーザーにも適用されます。1 つのグループを別のグループのサブグループにするには、希望する親グループを選択してから、パネルの下部の「**サブグループ (Subgroups)**」タブをクリックします。サブグループにするグループを選択し、「**追加**」をクリックします。グループを再帰的にネストすることができます。例えば、親を子に追加して、グループ A を含むグループ B が、グループ A に含まれるようにします。これを行う場合、システムはグループ A のすべてのメンバーをグループ B のメンバーとして扱います。その逆も同様です。
- グループの権限を管理するには、グループを選択して、「**権限**」タブをクリックします。グループの現在の権限を表示したり、権限を追加または除去したりできます。

## アクセスの概説

システムは、データベース内でユーザーを管理します。各ユーザーの特権は、各ユーザーの割り当て先グループによって制御されます。特権をグループに割り当ててから、各ユーザーを適切なグループのメンバーとします。

これは、役割ベースのシステムです。グループは、ユーザーが組織内で所有することができる役割を表し、役割は特権を備えています。ユーザーは、そのユーザーが属しているグループの特権をすべて持ちます。特権を割り当てられるのはグループに対してのみであり、個々のユーザーに直接割り当てることはできません。

また、システムでは、通知に関するアクセス・グループも使用します。通知メッセージを送信するようにシステムを構成する場合、メッセージのターゲットはアクセス・グループにする必要があります。 385 ページの『通知のセットアップ』を参照してください。

セキュリティー特権、すなわち **権限** は、グループが実行も表示もできる対象を定義します。これらは、システムでグループが実行できる操作のフィルターとして機能します。例えば、ゲスト・グループのメンバーである (かつ、その他のいずれのグループのメンバーでもない) ユーザーは、ゲスト・グループにアクセス・プロパティーとして割り当てられているプロジェクトのみを表示できます。そのユーザーは、ゲスト・アクセスを備えたプロジェクトのみを起動できます。また、ユーザーが開発者グループのメンバーでもある場合は、アクセス・プロパティーがゲストまたは開発者であるすべてのプロジェクトが表示されます。

**注:** ユーザー認証では、データベースの代わりに、既存の LDAP データベースを使用できます。LDAP を使用するときには、システムでユーザーを定義する代わりに、LDAP データベースのユーザーの一部またはすべてが、システムにアクセスできるようにします。アクセス・グループを LDAP グループにマップすることもできます。LDAP のセットアップについて詳しくは、254 ページの『LDAP の統合について』を参照してください。

アクセス・グループで制御できるアクティビティーおよびリソースは、権限、サーバー、プロジェクト、ステップ、およびアクセス・グループです。

- リソース (サーバー、プロジェクト、またはステップ) へのアクセスを特定グループに拡張するには、リソースを選択し、そのアクセス・フィールドを当該グループの名前に変更します。例えば、開発者グループが、Win234 という特定サーバーへアクセスできるようにするには、そのサーバーのアクセス・プロパティーを開発者に設定します。

**注:** サーバーまたはプロジェクトのアクセス・グループのメンバーではないユーザーには、「サーバー」または「プロジェクト」リストのページにリストされるオブジェクトは表示されません。ステップのアクセス・グループのメンバーではないユーザーは、プロジェクトのリストにあるステップを表示することはできませんが、そのステップを編集したり実行したりすることはできません。ユーザーが、そのステップを含むプロジェクトを実行した場合、システムは、ユーザーがアクセス権限を持っていないステップをスキップします。

- あるアクセス・グループのメンバーが、別のアクセス・グループを編集できるようにするには、別グループの制御グループとして、グループを設定します。例えば、グループ A のメンバーが、グループ B にメンバーを追加できるようにするには、グループ A をグループ B の制御グループに設定します。
- グローバル特権のあるグループに拡張するには、「管理」 > 「権限」ページを使用して、そのグループに対して特定の権限を使用可能に設定します。

この柔軟なモデルにより、あるタイプのユーザーに対して、1 つの特権 (ビルドを実行する機能など) を安全に付与し、一方では別の特権 (プロジェクトを編集したり、特定のサーバーを使用する権利など) を制限することができます。

## アクセスの例: グループへのジョブ実行権限の付与

セキュリティー機能を使用して、特定のジョブを実行する権利を、アクセス・グループの 1 つに拡張することができます。例えば、デバイス・ドライバ・プログラマーのグループがあり、プログラマーが自身の仕事に関連するジョブを実行できるようにするとともに、プログラマーが使用するシステムのビューがその他のジョブ

で乱雑にならないようにする場合について考えてみます。ただし、これらのプログラマーにジョブの編集は許可しません。このシナリオを作成するには、以下を行います。

1. 組織内に、この役割に関するアクセス・グループ (例えば、DeviceDriverDevs) を作成する。
2. ユーザーに実行できるようにさせたいすべてのプロジェクトの「アクセス」プロパティーとして、新規のアクセス・グループを割り当てる。
3. プロジェクトのステップも、適切な「アクセス」プロパティーを持つようにする。ユーザーがアクセス権を持っていないステップは、ジョブの実行時にスキップされます。
4. 権限「ビルドの実行 (Execute Builds)」をグループに割り当てる。
5. 新規の DeviceDriverDevs グループに、これらのビルドを起動する必要のあるすべてのユーザーを追加する。グループのシステム・メンバーの管理者を作成する必要がある場合もあります。プロジェクトのアクセス・プロパティーを変更すると、DeviceDriverDevs グループのメンバーでないユーザーは、プロジェクトの表示、実行、または編集を行う権利を失います。

ユーザーは、多くのグループのメンバーとなることができ、複数の権限が累積されることに注意してください。もう 1 つのプロジェクト・チーム (例えば、PlatformDevs) のグループがある場合、両方のグループのメンバーであるユーザーは、いずれかのグループをアクセス・プロパティーとして設定したプロジェクトを、表示したり起動することができます。

## チームおよびプロジェクトのセキュリティ・プラン

異なるプロジェクトで作業する多くのユーザーがいる場合、以下の一般的なプランによって、ユーザーを管理できます。このプランでは、個々のユーザーは必要な権限を取得するが、相互にやり取りする必要のある、プロジェクトおよびその他のリソースについては、表示のみが許可されます。

- ユーザーが実行するさまざまなアクティビティーについて、役割ベースのアクセス・グループを作成します。例えば、ビルド・マネージャーや開発者のグループを作成します。これらのグループのジョブに適した権限を、そのグループに割り当てます。ビルド管理者は、選択可能な権限のほとんどを所有する場合がありますが、開発者は、ジョブの実行に関する権限のみを持つことが考えられます。
- 部門間協力を行う組織内のチームごとに、追加グループを作成します。IDE チーム、PrinterDriver チームなどが考えられます。
- チーム・グループに対して、プロジェクト、サーバー、およびその他のリソースのアクセス・プロパティーを設定します。PrinterDriver チームに関連するすべてのプロジェクトは、アクセス・プロパティーとして、PrinterDriver のアクセス・グループを持つことになります。
- ユーザーをシステムに追加する場合に、適切なアクセス・グループすべてに、ユーザーを割り当てます。すべてのユーザーは、1 つ以上の役割グループと、1 つ以上のチーム・グループに割り当てられる必要があります。

これらのガイドラインに従った場合、ユーザーが表示できるのは、そのユーザーに関連するプロジェクトのみであり、ユーザーは、それらのプロジェクトでの役割に

適した権限を所有します。また、組織内でのユーザーのジョブが変更された場合には、それに合わせてユーザー権限を容易に変更することもできます。

## アクセス・プロパティの管理

ユーザーは、そのユーザーのアクセス・グループに基づいてアクセス・プロパティを設定します。ユーザーは、プロジェクト、サービス、ステップなどのデータ・オブジェクトの割り当て先となるアクセス・グループのメンバーである必要があります。

例えば、管理者グループのメンバーでない場合、プロジェクトをそのグループに割り当てることはできません。

アクセス・グループのリストは、自分がメンバーとして所属しているグループのみに限定されます。

ステップは親プロジェクトからアクセス・グループ・プロパティを継承します。ステップの作成者は、ステップがプロジェクトとは異なるアクセス・グループ・プロパティを持つように、ステップのアクセス・グループ・プロパティを変更できます。ステップに指定されたアクセス・グループのメンバーではないユーザーは、ステップを実行できません。これにより、ユーザーがプロジェクト内の特定のステップを実行できないようにすることができます。

---

## ユーザー

以下のトピックでは、システムにおけるユーザーの管理方法を説明します。

### ユーザー・プロパティ

ユーザーを作成し、「管理」 > 「ユーザー」とクリックして、それらのユーザーにプロパティを割り当てることができます。また、システムを LDAP/Active Directory データベースに接続して、ユーザー情報を取得することもできます。ユーザーをグループに割り当てることにより、ユーザー・セキュリティ許可を管理するので、セキュリティ機能をテストするために何人かのユーザーを作成する必要があります。

ユーザー・レコードを表示した場合には、次の 3 つのタブが使用できます。

- **詳細:** このタブを使用して、ほとんどのユーザー・プロパティを編集します。使用可能なプロパティを以下に説明します。
- **現在のグループ:** ユーザーがメンバーのアクセス・グループを表示します。アクセスは、直接行うか、または別のグループのメンバーである直接グループを通じて行われます。
- **グループの変更:** ユーザーが直接メンバーであるグループを表示し、グループにユーザーを追加したり、グループからユーザーを除去することができます。



The screenshot shows a web-based user management interface. At the top, there is a row of buttons: (New User), Save, Copy, Switch To User, Expire Password, Logout User, Purge Seat, Unlock, and Delete. Below this is a tabbed interface with three tabs: Details (selected), Current Groups, and Change Groups. The Details tab contains several input fields and dropdown menus arranged in a grid-like fashion. Fields include: User name, Email, Name, Password, Limit (set to Unlimited), Time Zone (set to US/Central), Verified, Priority Login (set to No), Date Format (set to 1/1/03 2:30 PM), Language (set to English US), Password Expires (set to Yes), Uses screen reader (set to No), Calendar Start Day Of Week (set to Sunday), Truncation (set to Default Value), User type (set to Normal), Searching (set to Case Sensitive), and Step Log Initial View (set to First Page).

各ユーザーについて、「詳細」タブで次のプロパティを設定できます。

**名前** ユーザーの表示名およびラベルです。

### E メール

システムがこのユーザーに E メール通知を送信できる E メール・アドレスです。

**注:** 通知対象として明示的に選択されたユーザーにのみ、E メールが送信されます。以下の E メール・アドレス・フォーマットがサポートされています。*username* は、Eメールのユーザー名です。*name* は、任意のストリング (通常、何らかの形で名前を表すもの) です。

```
username@host.com
name username@host.com
name "<username@host.com>"
name <username@host.com>
name "username@host.com"
```

### ユーザー名

管理コンソールへのログオンに使用する名前です。

### パスワード

管理コンソールへのログオンに使用するパスワードです。現在ログオン中のユーザーには、フィールドは表示されません。このフィールドを使用して新規パスワードを入力するか、既存のパスワードを変更します。「確認」フィールドに同じパスワードを入力します。

**上限** ユーザーが 1 日に起動することができるジョブの最大数を設定します。上限に達すると、システムはユーザーの実行割り当て量が超過したことを示すメッセージを表示します。

### タイムゾーン

ユーザーのタイムゾーンです。システムは、通知されるすべての時刻のデフォルトのタイムゾーンとして、root ユーザーのタイムゾーンを使用します。通知時刻は、サーバー・ホスト・クロックに基づいています。

デフォルトでは、システム内のユーザーと LDAP のユーザーに、root ユーザーと同じタイムゾーンが割り当てられます。ユーザーは、自分に割り当てられたタイムゾーンを編集できます。

**確認** 確認のため、ここでパスワードを再入力します。

### 優先ログイン

ユーザーを優先ユーザーにするには、このオプションを指定します。優先ユーザーは常に、システムにログインすることができます。使用可能なユーザー

ー・ライセンスの数が許容範囲に達すると、システムは最も古いセッションのユーザーをログアウトさせて、優先ユーザーのためのスペースを作ります。 root ユーザーも常にログインできますが、ライセンスを必要としません。

#### 日付書式

ユーザーが好きな表示書式を選択します。

言語 ユーザーの言語を選択します。

#### パスワード期限切れ

このオプションにチェック・マークが付いている場合、そのユーザーのパスワードは、「パスワードの有効期限日数」システム設定に指定された日数が経過した後には有効期限切れになります。

#### スクリーン・リーダーの使用

「はい」を設定すると、強調表示およびフォーカスを動的に行うなど、視覚障害を持ったユーザーに対するスクリーン・リーダー機能をサポートするインターフェースが使用可能になります。

#### ユーザー・タイプ

ユーザーは、次の 3 つのタイプのいずれかに属している場合があります。

- 通常: Web クライアントを使用して、コンソールにアクセスできます。属しているアクセス・グループのメンバーおよび権限に基づいて、システムを変更することができます。1 ユーザーにつき 1 つのセッションのみが許可されます。
- 読み取り専用: Web クライアントを使用して、コンソールにアクセスできます。個人情報 (ユーザーのパスワードや設定など) 以外の項目は変更できません。1 ユーザーにつき 1 つのセッションのみが許可されます。
- API: Web クライアントを使用して、コンソールにアクセスすることはできません。このタイプのユーザーは、Java API または Perl API で作成されたクライアントを使用して、システムにアクセスします。同じ API ユーザーが複数のログイン・セッションを保持する可能性があります。セッション情報は自動的に共有されます。

#### 週のカレンダー開始日

スケジュール・カレンダーで先頭に表示する曜日を選択します。デフォルトは「日曜日」です。

#### 切り捨て

リストおよびプルダウン・メニューに表示する文字数を制御します。例えば、20 に設定した場合、プロジェクト名の最初の 20 文字のみが表示されます。

#### 検索 (Searching)

「フィルター」フィールドが製品全体でどのように検索を実行するかを決定します。値は以下のとおりです。

- 大/小文字の区別 (Case Sensitive) (デフォルト) - 検索では各文字の大/小文字が考慮されます。
- 大/小文字を区別しない (Case Insensitive) - 検索では大/小文字は考慮されません。

## ステップ・ログ初期ビュー

ジョブの結果でステップ・ログが開かれるときに、ステップ・ログがどのように配置されて表示されるかを指定します。「最初のページ」(デフォルト)に設定された場合、ビューはログの最初のページに配置されます。「最後のページ」に設定された場合、ビューはログの最後のページに配置されます。

## ユーザーの管理

以下のトピックでは、基本的なユーザーの管理操作について説明します。

「管理」 > 「ユーザー」を選択して、現在のユーザーのリストとその下のユーザー・パネルを表示します。システムは、各ユーザーごとに名前、ユーザー名、Eメール、上限、アクティビティー (最後のユーザー・ログイン以降の、サーバー・ホスト・クロックに基づく経過時間)、およびタイムゾーンを表示します。

- 新規ユーザーを作成するには、ユーザーが選択されていないときにユーザー・パネルでプロパティの入力を開始します。ユーザーが選択されている場合は、「**ユーザーの追加**」をクリックして、パネルをクリアします。ユーザー情報の編集が完了したら、「**保存**」をクリックします。
- ユーザーを編集するには、ユーザーの名前をクリックしてユーザー・パネル内でプロパティを編集し、「**保存**」をクリックします。
- ユーザーをログアウトするには、そのユーザーの名前をクリックしてから、「**ユーザーのログアウト**」をクリックします。
- パスワードを使用せずにユーザーとしてログインするには、最初に root としてログインします。ユーザーの名前をクリックしてから、「**ユーザーの切り替え**」をクリックします。root ユーザーは、パスワードを使用することなく、どのようなユーザーとしてでもログインできます。root 以外のユーザーとしてログインしていて、別の root 以外のユーザーとしてログインしたい場合は、まず root としてログインする必要があります。
- 固定ライセンス・シートを解放するには、最初に root としてログインします。ユーザーの名前をクリックしてから、「**シートのページ**」をクリックします。コンソールは、固定ライセンスのセットに対してカウントされた ID のリストから、そのユーザーを削除します。また、そのユーザーがログインしている場合は、そのユーザーのログオフも行われます。固定ライセンスの場合、コンソールは、これまでにログインしたことのあるユーザーの数をカウントします。上限に達すると、新規ユーザーはライセンスを取得できなくなります。別のユーザーがライセンスを取得できるようにするには、既存のユーザーを削除またはページする必要があります。シートをページしても、ユーザーはコンソールから削除されません。ユーザーが再度ログインすると、固定ライセンスの数は増加します。フローティング・ライセンスのユーザーに対して使用した場合、「**シートのページ**」には「**ユーザーのログアウト**」と同じ効果があります。
- ユーザーをコピーするには、ユーザーの名前をクリックしてから、「**コピー**」をクリックします。「コピー」という単語が追加された新しいユーザーの名前が、リストに表示されます。

**注:** コピーされたユーザーのパスワードは password にリセットされます。パスワードを変更するには、root またはコピー・ユーザーとしてログインしてください。

- ユーザー・アカウントを削除するには、ユーザーの名前をクリックしてから、「**削除**」をクリックします。

「削除」を使用できない場合、スケジュールに入れられたジョブがユーザー・アカウントによって所有されており、そのユーザー・アカウントを削除できません。ジョブがスケジュールされているユーザー・アカウントを削除するには、スケジュールに入れられたジョブを先に削除する必要があります。

ユーザー・レコードはシステムのユーザー経験のデフォルト・プロパティを設定し、ユーザーのログイン名、パスワード、およびパスワードの有効期限を管理します。ユーザー・レコードのデータは、管理コンソールからシステムに入力すること、LDAP/Active Directory データベースから取り出すこともできます。

**注:** LDAP データベースから取り込んだレコードを持つユーザーを編集する場合は、「ユーザー」ページの多くのフィールドが使用不可になっています。これらのプロパティはソース・データベースで変更する必要があります。

## ユーザー・タイプを使用したアクセスの制御

「ユーザー・タイプ」プロパティの設定によって、ユーザーが持つアクセス権のタイプが決まります。

- **通常:** Web クライアントを使用して、コンソールにアクセスできます。属しているアクセス・グループのメンバーおよび権限に基づいて、システムを変更することができます。
- **読み取り専用:** Web クライアントを使用して、コンソールにアクセスできます。個人情報 (ユーザーのパスワードや設定など) 以外の項目は変更できません。
- **API:** Web クライアントを使用して、コンソールにアクセスすることはできません。このタイプのユーザーは、Java API または Perl API で作成されたクライアントを使用してシステムにアクセスします。同じ API ユーザーが複数のログイン・セッションを保持する可能性があります。セッション情報は自動的に共有されます。

LDAP を使用してログインしたユーザーは、「**書き込みアクセス・グループ識別名**」プロパティの設定に応じて、「通常」または「読み取り専用」のユーザー・タイプと判別される場合があります。254 ページの『LDAP の統合』を参照してください。

## アカウント・ロックアウト

システム設定を使用して、ロックアウト・ポリシーを適用することができます。

ロックアウト・ポリシーは、ログインの試行に複数回失敗した場合に、ユーザーをログインさせないようにする条件を決定します。この条件は、連続して失敗したログインの最大試行回数として設定します。一連のログインを連続した試行と見なすかどうかは、タイマーにより決定されます。システムは、このタイマーに基づいて、失敗したログインを「破棄」できます。

あるユーザー・アカウントについて、最後にログインが成功してからログインが複数回失敗した場合、次にログインが成功した際に、失敗したログイン試行回数を示すポップアップが表示されます。このポップアップは、ロックアウト条件を満たしていない場合にのみ表示されます。

ロックアウト条件が満たされると、ユーザーはロックアウト状態になります。ユーザーは、以下の 2 つのアクションのうちの 1 つを行う必要がある場合があります。

- 別のログインの試行が受け入れられるまで、あらかじめ決められた時間が経過するのを待ちます。
- アカウントをアンロックするには、システム管理者に連絡してください。

以下のシステム設定によってロックアウト・ポリシーが決定します。

- **アカウント・ロックアウト最大試行回数:** 0 の場合、アカウント・ロックアウトは無効になります。ロックアウトの発生条件となる、連続試行失敗の許容回数を設定できます。デフォルトは 3 です。
- **アカウント・ロックアウト・リセット・タイマー:** 負の値に設定した場合、ユーザーは再度ログインできるように、管理者に連絡してアカウントのロックを解除するよう依頼する必要があります。ユーザーが再度ログインできるようになるまでに必要な分数を設定できます。デフォルト: 120 分。
- **アカウント・ロックアウトの減衰 (Account Lockout Decay):** 負の値に設定すると、失敗した試行が破棄されることはありません。最後に失敗したログイン試行が破棄されて、試行失敗の最大許容回数としてカウントされなくなるまでの分数を設定できます。デフォルト: 60 分。

「管理」 > 「システム」を参照してください。

## root ユーザー

root ユーザー (root のログイン名を持つユーザー) には、システムにおいて以下のような特性があります。

- **インストール時に作成:** root ユーザーは、インストール・プログラムが作成する唯一のデフォルト・ユーザーです。デフォルトのパスワードは、**root** です (パスワードはインストール直後に変更します)。
- **ライセンス不要:** root ユーザーはユーザー・ライセンスを使用しません。ログイン中のユーザー数に関係なく、常時、root ユーザーとしてログインすることができます。(あるユーザーが root としてログインする際、root として既にログインしている他のユーザーはログオフされます。)
- **システム・タイムゾーン:** root ユーザーのタイムゾーンは、管理コンソールのデフォルトのタイムゾーンになります。他のユーザー (システム内のユーザーおよび LDAP ユーザー) 用のタイムゾーンは、デフォルトでは root ユーザーのタイムゾーンから取得されます。ユーザーは、ログインした後で独自のタイムゾーンを設定することができます。システムで報告された時刻とログはすべて、ユーザーのタイムゾーンで示されます。
- **すべての権限:** この場合、root ユーザーはすべての有効な権限を持ち、他のユーザーのプロパティを編集することができます。root ユーザーからは、どのようなアクセス権も除去することができません。root ユーザーはアクセス・グループのメンバーではありませんが、システム内のデータ・オブジェクトの表示、編集、または使用が可能です。
- **優先順位:** root ユーザーは常に優先ユーザーとなります。

- **任意のユーザーとしてログイン:** ルート・ユーザーは、「管理」 > 「ユーザー」 > <ユーザー名> ページにある「ユーザーの切り替え」をクリックすることにより、パスワードを使用せずにユーザーとしてログインすることができます。
- **現行ユーザーをログアウト:** この場合、root ユーザーは、「管理」 > 「ユーザー」 > <ユーザー名> ページにある「ユーザーのログアウト」をクリックすることにより、ユーザーをログアウトできます。
- デフォルトでは、LDAP ユーザーは root ユーザーと同じタイムゾーンに割り当てられています。ただし、一度ログインすれば、割り当てられているタイムゾーンを編集することができます。新規設定の内容はシステムに記憶されます。

## API ユーザー

Java API または Perl API で作成されたクライアントを使用してコンソールにアクセスするユーザーは、ユーザー・タイプを「API」としてセットアップする必要があります。このタイプのユーザーは、Web クライアントを使用して、コンソールにアクセスすることはできません。ただし、このタイプのユーザーは、複数のログイン・セッションを保持することができます。セッション情報は、セッション間で自動的に共有されます。

同じ API ユーザーによって開始された複数のセッションが使用するライセンスは 1 つだけです。ライセンスを解放するためには、すべてのセッションからログアウトする必要があります。root アカウントを使用してライセンスを解放することもできます。249 ページの『ユーザーの管理』を参照してください。

Web クライアントと API クライアントの両方を使用してコンソールにアクセスする必要があるユーザーがいる場合には、それぞれに別のユーザー ID をセットアップしてください。Web コンソールにアクセスするユーザーのユーザー・タイプは、「通常」または「読み取り専用」のいずれかです。これらタイプのユーザーは、複数セッションを保持することができません。これらのタイプのユーザーが 2 回ログインすると、最初のログイン・セッションは強制終了されます。

---

## 権限

権限は、システム内でユーザーができる操作を定義します。アクセスするための権限をグループに割り当てます。これらは、ユーザーに直接割り当てないでください。

権限を処理するには、「管理」 > 「権限」と選択します。



UI Config Console Reports Logout: Root User

Permissions Help ?

Filter Showing 1 - 4 of 105 Display All << < Page 1 of 27 > >>

Name	Group
<a href="#">Add Build Notes</a>	Projects
<a href="#">Add Child Access Groups</a>	Access
<a href="#">Add New Access Levels</a>	Access
<a href="#">Add New Classes</a>	Classes

**Add Build Notes**

**Details**

Name: Add Build Notes Group: Projects Select names in the list below and click Add to add them to the list of members on the right.

Guest	Add >>	Build Engineer
Operator		Developer
	<< Remove	Security
		System Manager

グループに権限を割り当てるには、権限を選択して、希望するグループがその権限を持つものとしてリストされることを確認してください。「管理」 > 「権限」を選択し、リストから権限を選択してから、パネルの下部で作業します。左側のグループには権限がなく、右側のグループには権限があります。グループに権限を付与するには、左側のグループ内でそれを選択してから、「追加」をクリックします。

## 権限の演習

この例では、Jane Doe というユーザーに対して、サーバーを追加および編集する排他的な権利を与えます。このために、該当の権限を新規アクセス・グループに与え、そのグループに彼女を割り当てます。

1. root としてログインします。
2. Server Admin という新規アクセス・グループを作成します。
3. Jane Doe をその新規アクセス・グループに追加します。
4. 「管理」 > 「アクセス権」に移動します。
5. スクロールして「新規サーバーの追加 (Add New Servers)」権限を表示し、それをクリックします。
6. 「詳細」タブで「追加」および「削除」を使用して、Server Admin を、この権限が与えられた唯一のアクセス・グループにします。
7. 「更新 (Update)」をクリックします。これで、Jane Doe のみが、サーバーをシステムに追加する権限を持つようになります。

注: root ユーザーは常に、すべての権限を持ちます。root ユーザーからは、どのようなアクセス権も除去することができません。



Jane のみがサーバーを追加できます。ただし、彼女は、サーバーのログイン情報を入力することはできません。ログイン情報を入力する権限は、Server Admin グループを「サーバー認証の編集」権限に追加することによって付与することができます。

---

## LDAP の統合

LDAP サーバーを使用してユーザーのログインを行うように Build Forge をセットアップすることができます。LDAP を使用することで、ユーザーは、組織内の他の場所で使用するものと同じログイン名およびパスワードを使用して Build Forge にログインできます。LDAP を使用する場合、Build Forge でユーザーを手動作成する必要はありません。各ユーザーは、Build Forge に最初にログインする際に Build Forge 内で作成されます。

LDAP グループを Build Forge のアクセス・グループにマップするオプションがあります。この統合により、LDAP でグループを管理して、ユーザーのログイン時にユーザーのアクセス権が自動的に更新されるようにすることができます。

手動でのユーザーの作成および保守は、引き続き Build Forge システムで実行できます。それらのユーザーのアクセスは手動で管理する必要があります。

この統合を有効にするには、「管理」 > 「LDAP」で項目を作成します。

注: Build Forge ユーザー・インターフェースでは、LDAP ドメイン項目を作成および編集できるのは root ユーザーのみです。

## LDAP の統合について

ユーザーは、LDAP 資格情報を使用して最初に Build Forge にログインする際、Build Forge 内で次のように認証され、セットアップされます。

**重要:** グループ・マッピングを使用する予定の場合、ユーザーがログインする前に LDAP グループ・マッピングを有効にしてください。

グループ・マッピングが無効の状態ではユーザーがログインし、後でグループ・マッピングを有効にしても、現在のユーザーに対してマッピングは実行されません。ユーザーがログインした後で LDAP グループ・マッピングを使用可能にする場合、Build Forge のユーザー・リストからユーザーを削除し、再度ログオンさせてください。これにより、Build Forge アクセス・グループへのそのユーザーのメンバーシップが、行った手動変更ではなく LDAP グループ・マッピングに基づいて設定されるようになります。

1. ログイン・パネルに「ドメイン」フィールドが表示されます。複数のドメインが構成されている場合、このフィールドはプルダウン・リストになります。ユーザーはドメインを選択して、ログインします。

注: 複数のドメインを構成する場合、個々の固有ユーザー ID は、ドメイン全体で固有でなければなりません。システムでは、固有のユーザーごとに許可されるログインは 1 つのみです。あるユーザーがログインし、その後、別のユーザーが同じ固有ユーザー ID を使用してログインすると、最初のユーザー・セッションは閉じられます。ユーザー・セッションについて詳しくは、9 ページの『コンソールへのアクセスおよびその使用』を参照してください。

2. Build Forge が LDAP サーバーでアカウントを検査します。通常ユーザーまたは管理ユーザーを使用して検査を実行するように Build Forge を構成することができます。
3. ユーザー名が見つかった場合、Build Forge は、Build Forge のログイン・パネルで (または API クライアントを使用するプログラムのログインから) ユーザーが指定した資格情報を使用して LDAP にログインしようとします。
  - 資格情報が一致しないか、ユーザー名が見つからない場合、ログインは失敗します。
  - 資格情報が一致した場合、ログインが続行されます。
4. ユーザーが以前にログオンしたことがない場合、Build Forge はそのユーザー・リスト内にユーザーを自動的に作成します。LDAP を介してログインするユーザーの場合、「ユーザー名」、「パスワード」、「ログイン」、「確認」、および「E メール」の各フィールドは使用不可になっています。これらの情報は LDAP によって提供されるためです。

注: 最初にログインする際、システムは LDAP ユーザーを *root* ユーザーのタイムゾーンに割り当てます。LDAP からタイムゾーン情報が得られないためです。タイムゾーンは後から手動で設定できます。

5. Build Forge がユーザーにアクセス・グループを適用します。
    - LDAP グループ・マッピングが有効になっている場合は、指定のアクセス・グループが適用されます。デフォルトの Build Forge アクセス・グループも適用されます。グループ・マッピングを有効にするには、Build Forge の LDAP ドメイン・プロパティ内での構成が必要です。
- 注: グループ・マッピングは、ユーザーがログインするたびに実行されます。これにより、LDAP 内のグループ・メンバーシップの変更と Build Forge の同期が維持されるようになります。
- LDAP グループ・マッピングが無効になっている場合は、Build Forge のデフォルト・アクセス・グループが適用されます。その後、アクセス・グループのメンバーシップは手動で管理できます。

## LDAP ドメインのプロパティ

作成された LDAP ドメインのプロパティを編集するには、次のようにします。

1. 「管理」 > 「LDAP<Domain Name>」と選択します。
2. 編集するドメインを選択します。LDAP ドメインのプロパティ・パネルにプロパティが表示されます。

3. 任意のフィールドの値を編集して、「保存」をクリックします。次のフィールドは必須です。
  - 名前
  - ホスト
  - ユーザー・アカウントのバインド
  - プロトコル
  - 表示名
  - 識別名
  - メール
  - 検索フィルター
4. そのドメインをデフォルトとして使用することを指定するには、「デフォルトの設定」をクリックします。

**名前** これは必須です。Build Forge 内の LDAP ドメインの名前です。少なくとも 1 つの LDAP ドメインが構成されている場合、Build Forge のログイン・フォームにそれらのドメインがこの名前でリストされます。

#### 管理識別名

LDAP サーバー・データベースに検索アクセスを行うために使用するアカウントです。サーバーで、データベースの検索に匿名バインドを使用することが許可されている場合は、このフィールドをブランクのままにしてください。

LDAP サーバーの中には、データベースを検索するために管理バインドを必要とするものがあります。この設定を行うと、次の例に示すように、管理者アカウントの DN を指定できます。

`cn=Administrator,cn=users,dc=example,dc=com`

「管理識別名」アカウントのパスワードを「パスワード」フィールドおよび「パスワードの確認」フィールドに指定してください。

#### アクセス・グループのマッピング

LDAP サーバーから管理コンソール内のアクセス・グループにグループ情報をマッピングするかどうかを決定します。デフォルトは「いいえ」です。Build

Forge 内の各アクセス・グループの「**LDAP グループ識別名**」プロパティが、LDAP 内の正しいグループ名に設定されている必要があります。

- 「**いいえ**」の場合、LDAP のグループは Build Forge のアクセス・グループにマップされません。Build Forge のアクセス・グループに割り当てることができるのは、少なくとも 1 回はログインしたユーザーです。このオプションを使用するということは、Build Forge 内でユーザーのアクセス・グループを管理するということです。ユーザーが最初にログインする際、そのユーザーの名前が Build Forge 内で作成されたものである場合には、デフォルトのアクセス・グループが適用されます。
- 「**はい**」の場合は、あるユーザーが Build Forge にログインするたびに、Build Forge は、そのユーザーのグループ・メンバーシップ情報を LDAP サーバーから更新します。前回のログイン以降に、Build Forge 内のそのユーザーに対して実行されたアクセス・グループ・メンバーシップの変更はすべて上書きされます。このオプションを使用するということは、すべてのグループ・メンバーシップを LDAP 内で管理するということです。LDAP のグループ・メンバーシップが、Build Forge のアクセス・グループに自動的にマップ (追加または除去) されます。グループのプロパティを次のように使用することで、ユーザーのグループ・メンバーシップを判別します。
  1. 「**グループ名**」がブランクでない場合は、指定されたキーワードの値を照会します。ユーザーのグループとして返された値を使用してください。
  2. 「**グループ名**」がブランクであるか、その照会で値が返されない場合は、「**グループ検索基準**」および「**グループ検索フィルター**」を使用して、そのユーザーが属する LDAP グループを照会してください。
  3. (1) および (2) でグループ情報が返されない場合、そのユーザーには、ログインの許可が与えられているほか、新規ユーザー用のデフォルト・アクセス・グループとして指定されたアクセス・グループのメンバーシップが割り当てられています。

**ホスト** これは必須です。LDAP サーバーのホスト名およびポートです。例:

```
ldapservers.mycompanyname.com  
ldap.mycompany.com:9000
```

### パスワード

「**管理識別名**」アカウントのパスワードです。「**管理識別名**」を指定する場合には必須です。

**確認** 「**管理識別名**」のパスワードの入力を繰り返します。

### ユーザー・アカウントのバインド

これは必須です。ログイン時に Build Forge が LDAP に対してユーザーの資格情報の検証を試行するかどうかを決定します。デフォルトは「**はい**」です。

- 「**はい**」の場合、Build Forge は、ログイン時に指定されたユーザー名とパスワードを LDAP サーバーで検査します。
- 「**いいえ**」の場合、Build Forge はユーザー名を検証せずに受け入れます。この設定は、Build Forge にシングル・サインオン (SSO) などの外部パスワード検証が実装されている場合に使用します。

## プロトコル

これは必須です。Build Forge ユーザーを認証する目的でディレクトリー・サービスのデータを読み書きするために Build Forge が使用するプロトコルを指定します。デフォルトは LDAP です。SSL を介した LDAP (LDAPS) を使用する場合には、LDAPS を入力します。このオプションには、追加のセットアップが必要です。『セキュア LDAP (LDAPS) の使用可能化』を参照してください。

**表示名** これは必須です。ユーザーの氏名を示すキー名を入力します。

**識別名** これは必須です。ユーザー・アカウントの識別名を示すキー名を入力します。

**メール** これは必須です。ユーザーの E メール・アドレスを示すキー名を入力します。

## グループ名

ユーザーが所属するグループのリストを保持する、LDAP スキーマ内のキー名を入力します。「アクセス・グループのマップ」が「はい」の場合、または「認証済みグループ識別名」が使用されている場合にのみ使用されます。

## 認証済みグループ識別名

LDAP グループの識別名です。これが設定されている場合は、指定されたグループのメンバーのみがログインを許可されます。ブランクの場合は、すべての有効な LDAP ユーザーがコンソールにログインできます。

## 書き込みアクセス・グループ識別名

ユーザーのアクセス権限が通常か読み取り専用かを判別します。値は以下のいずれかです。

- ブランク - 新規ログインの場合、ユーザー・タイプは「通常」に設定されます。既存のユーザーの場合は、割り当てられているユーザー・タイプ（「通常」、「読み取り専用」、または「API」）が保持されます。タイプは、「管理」 > 「ユーザー」で設定します。
- \* (アスタリスク) - すべてのログインで「通常」のユーザー・タイプが使用されます。
- LDAP グループ名 - ユーザーがグループに属している場合、そのユーザーのタイプは「通常」に設定されます。ユーザーがグループに属していない場合、そのユーザーのタイプは「読み取り専用」に設定されます。
- その他 - 上記以外の値を使用して、すべてのユーザーのタイプを強制的に「読み取り専用」に設定します。例えば、RO などです。

## 検索基準

これは必須です。LDAP レコードにユーザーを照会する際に使用される検索ストリングです。例:

```
cn=users,dc=buildforge,dc=com
```

## 検索フィルター

これは必須です。ログイン時にユーザーが入力するユーザー名と比較する、LDAP データベース内のフィールドを指定します。ユーザーが入力するログイン名には、% 文字を使用してください。例:

```
(sAMAccountName=%)
```

## グループ検索基準

「グループ検索フィルター」が必須です。「アクセス・グループのマップ」が「はい」の場合、または「認証済みグループ識別名」が使用されている場合にのみ使用されます。LDAP レコードにグループ・データを照会する際に使用される検索ストリングです。ご使用の LDAP データベースが、ユーザー・レコードの保管に使用するデータベースとは別のデータベースにグループ・メンバーシップを保管している場合に必要です。例:

```
cn=groups,dc=buildforge,dc=com
```

## グループ検索フィルター

「グループ検索基準」が必須です。「アクセス・グループのマップ」が「はい」の場合、または「認証済みグループ識別名」が使用されている場合にのみ使用されます。グループ・メンバーシップ情報を取得する際に使用する、LDAP ユーザー・データベース内のフィールドを指定します。このフィルターは、ユーザー・アカウントの任意のデータ・フィールドを、グループ・テーブルを調べるためのキーとして使用することができます。フィールドは、`%fieldname%` という構文を使用して指定します。グループ・テーブルでユーザーのキーとして `sAMAccountname` フィールドを使用する場合は、次の例のようにすると機能します。

```
sAMAccountName=%sAMAccountname%
```

## タスク

以下のトピックでは、LDAP ドメインを処理するタスクについて説明します。

### LDAP ドメイン項目の作成

LDAP ドメイン項目は、必要な数だけ作成することができます。ユーザーがログインを試行する際には、「ドメイン」を指定する必要があります。API クライアントのログインの場合は、ログイン呼び出し内でドメインを指定するか、使用するドメインを `bfcclient.conf` 内で構成する必要があります。

ドメイン項目を追加するには、次のようにします。

1. 「管理」 > 「LDAP」と選択します。
2. 「LDAP ドメインの追加」をクリックします。
3. ドメインのプロパティに入力するか、これを変更します。「名前」は、Build Forge の内部プロパティです。デフォルトで指定されている値は、LDAP または標準の Microsoft® Active Directory サーバーで動作するように設計されています。
4. 「保存」をクリックします。

### LDAP ドメイン項目のテスト

LDAP ドメインが正しくセットアップされているか検査するには、次のようにします。

1. 「管理」 > 「LDAP」と選択します。
2. リストからドメインを選択します。
3. 「LDAP ドメイン・テスト」をクリックします。



システムが、LDAP ドメイン項目のプロパティを使用して、LDAP サーバーを照会します。

## セキュア LDAP (LDAPS) の使用可能化

ご使用の LDAP サーバーが SSL を介した LDAP (LDAPS) をサポートしている場合は、Build Forge の LDAP ドメイン項目でも LDAPS を使用するように構成することができます。デフォルトでは、厳格な SSL が構成されます。厳格な SSL では、サーバー認証が必要になります。

1. Build Forge で LDAP ドメイン項目を作成します。
2. 「**プロトコル**」プロパティを LDAPS に設定します。これにより、LDAPS の暗号化専用方式が有効になります。
3. 「**ホスト**」を、ご使用の LDAP サーバーの完全修飾ドメイン・ネームおよび SSL ポートに設定します。厳格なセキュア LDAP のデフォルトとして定義されているのは、ポート 636 です。例えば、`myldap.mycompany.com:636` などです。
4. LDAP サーバーから署名者証明書を取得し、それを Build Forge のトラストストアに追加します。「**管理**」 > 「**セキュリティ**」で、アウトバウンド LDAP は次の設定を使用するようにデフォルトで構成されています。
  - SSL パネル: デフォルト JSSE アウトバウンド SSL
  - 鍵ストア・パネル: デフォルト JSSE トラストストア。このトラストストアは、デフォルトで `<bfinstall>/keystore/buildForgeTrustStore.p12` を使用するように設定されています。署名者証明書をここに置きます。
5. Build Forge を再始動します。
6. 「**管理**」 > 「**セキュリティ**」と進み、ご使用のセキュア LDAP 構成を選択します。
7. 「**接続のテスト**」をクリックします。

**注:** Build Forge では、デフォルトで厳格な LDAPS SSL が構成されます。厳格な構成では、サーバー証明書の検証が必要です。厳格な LDAP を使用しない場合は、次のようにします。

1. JAVA\_OPTS 環境変数で、Tomcat のシステム・プロパティ `-Dcom.buildforge.services.server.ldap.strict=false` を設定します。Tomcat のスクリプトはこの変数を読み取り、指定されたすべてのシステム・プロパティを Tomcat プロセスに適用します。
2. Build Forge を再始動します。

この構成では、LDAP のサーバー証明書を Build Forge のトラストストアに追加する必要はありません。ただし、この構成は、SSL プロトコル設計の脆弱な実装です。Build Forge は、LDAP サーバーとの通信中に LDAP サーバーの ID を検査しません。

## LDAPS SSL 構成の変更

アウトバウンド LDAP 要求が使用する SSL 構成は、デフォルトでセットアップされます。これについて、次の 2 つの側面を変更することができます。

- **SSL 構成。**これは、ご使用の LDAP サーバーがデフォルトのプロトコルまたはハンドシェイクを使用して Build Forge と通信できない場合に実行する必要があります。



- 鍵ストア構成。厳格な SSL では、クライアント (Build Forge) が LDAP サーバーと安全に通信するために使用するトラストストア内に署名者証明書を置くことが必要です。別のトラストストアを使用する場合、またはトラストストアを別の場所に置く場合は、Build Forge 内でそのトラストストア用の新しい鍵ストア構成を作成する必要があります。

次の説明では、Build Forge のセキュア LDAP はすでに有効にし、Build Forge コンポーネントの SSL は有効にしていない場合を想定しています。

LDAPS SSL 構成を変更するには、次のようにします。

1. トラストストアの場所または名前を変更する場合は、それを Build Forge ホスト上の希望する場所に置きます。そこに LDAP サーバーの署名者証明書を追加します。
2. 必要に応じて、「管理」 > 「セキュリティ」 > 「鍵ストア」で、トラストストア構成を作成します。トラストストア構成には、トラストストアの場所および名前のプロパティが含まれています。
3. 必要に応じて、「管理」 > 「セキュリティ」 > 「SSL」で、SSL 構成を作成します。新しいトラストストア構成を使用するように構成します (新しいトラストストア構成を作成済みの場合)。必要に応じて、その他の構成の調整を行います。
4. 「管理」 > 「セキュリティ」で、「SSL 有効化」がまだ設定されていない場合には、これを「はい」に設定します。追加のフィールドが表示されます。
5. 「アウトバウンド LDAP (Outbound LDAP)」リストで、作成した SSL 構成を選択します。その他の設定は変更しないでください。
6. 「保存」をクリックします。
7. 「マスター BFClient.conf の更新」をクリックします。
8. これまで SSL が有効にされていなかった場合には、次のようにします。
  - a. 「SSL 有効化」をクリックして、「いいえ」にします。
  - b. 「保存」をクリックします。
  - c. 「マスター BFClient.conf の更新」をクリックします。
9. Build Forge を再始動します。
10. 「管理」 > 「LDAP」で、LDAP 構成を選択します。
11. 「接続のテスト」をクリックします。

## LDAP/Active Directory のサポートをオフにする

LDAP または Active Directory を使用した認証を停止するには、以下のようになります。

1. 「管理」 > 「LDAP」と選択します。
2. すべてのドメイン項目を削除します。削除するドメイン項目の横にあるごみ箱のアイコンをクリックしてください。

ドメインが存在しない場合は、Build Forge に手動で追加されたユーザーのみがログインできます。

## システム構成の設定

さまざまな設定を使用して、管理コンソールを構成できます。これらの設定は、「管理」 > 「システム」 ページにあります。

「管理」 > 「システム」 をクリックすると、システムは、設定のリストを表示します。システム設定の名前をクリックすると、設定の編集パネルが表示されます。

**注:** 数値を取るシステム設定の場合、管理コンソールは、1 つ以上の整数 (0 から 9 まで) で構成される任意の値を受け入れます。数値のグループ化文字 (コンマ (,)、小数点 (.), 他の非整数分離文字など) はサポートされません。

パネルには、以下のボタンが含まれます。

- **保存** - 設定値に対する変更が保存されます。
- **デフォルトへ復帰** - 設定をデフォルト値にリセットします。

下の表に、使用可能な設定を示します。

設定	説明
アカウント・ロックアウトの減衰 (Account Lockout Decay)	デフォルト: 60 分。この期間が経過すると、システムは最後に失敗したログインの試行を破棄します。これが負の値に設定された場合、失敗したログインは破棄されません。
アカウント・ロックアウト最大試行回数	デフォルト: 3。アカウントがロックされるまでに連続して失敗できるログインの試行回数。これが 0 に設定された場合、アカウント・ロックアウトは使用不可です。
アカウント・ロックアウト・リセット・タイマー	デフォルト: 120 分。最小: 5 分。  ユーザーがロックアウトされてから再びログインを試行することが許可されるまでに必要な期間。これが負の値に設定された場合、ロックアウトは無期限となります。 <b>注:</b> ユーザーが再びログインできるようになるには、管理者が設定をリセットする必要があります。
アラート E メール上限	指定された分数の間にシステムが送信するアラート E メールの最大数を設定します。例えば、値 10/60 は、1 時間当たり最大 10 個のメッセージを設定します。デフォルト値 0/0 は、アラート E メール・メッセージに上限がないことを示します。
インライン化されたステップ・コンテナ環境変数を適用します	デフォルト: 「いいえ」。「はい」の場合は、インライン化されたステップを含むプロジェクトまたはライブラリーの環境変数が適用されます。 <b>注:</b> Java エンジンではサポートされていません。
サーバー環境変数の最後を適用します	デフォルトは「いいえ」です。「はい」の場合、最後にステップのサーバー環境変数を適用します。ステップ環境変数またはプロジェクト環境変数の後にサーバー環境変数が適用されます。これが適用されるのは、環境変数が指定された場合に限られます。 <b>注:</b> Java エンジンではサポートされていません。
監査のストリングの最大長	デフォルト: 128 文字。  これにより、詳細な変更について記録できる情報の量が制御されます。ストリング値がこの文字数を超えると、暗号ハッシュが元の値の代わりに保管されます。暗号ハッシュが使用されると、ストレージ・スペースが節約されます。パスワードなどの機密情報を含むフィールドでは、ストリング長に関係なく暗号ハッシュが使用されます。

設定	説明
監査ポリシー	<p>監査ログの量が決定されます。</p> <p><b>なし</b>      監査情報は記録されません。</p> <p><b>セキュリティ</b> セキュリティ関連情報のみが記録されます。以下の例が表示されます。</p> <ul style="list-style-type: none"> <li>• アクセス・グループの作成</li> <li>• ユーザー・セッション・アクティビティ</li> </ul> <p><b>基本</b> この設定がデフォルトです。これは、監査ポリシー値が無効な場合にも使用されます。最小限の情報 (例: ユーザー、オブジェクト・タイプ、ユーザー ID、アクション) のみが記録されます。</p> <p><b>詳細</b> 変更されたプロパティに関する詳細情報 (古い値と新しい値を含む) が記録されます。すべてのオブジェクトがこの情報を提供できるわけではありません。この情報を提供できないオブジェクトの場合、簡易ポリシーと詳細ポリシーに違いはありません。保管されるデータの量を制限するには、「監査の最大ストリング長 (Audit Max String Length)」システム構成設定を使用してください。</p>
自動ログオフまでの分数	アイドル状態のユーザーがシステムから自動的にログオフされます。この設定は、システムがユーザーをログオフするまでに経過する必要がある、アイドル時間の分数を指定します。設定値が 0 の場合、システムによるユーザーの自動ログオフは実行されません。
AutoClean エラー・ログ日数 AutoClean 情報ログ日数 AutoClean 警告ログ日数	これらの値は、エントリーの各カテゴリーが監査ログに残存する最大日数を示します。エントリーは古いものから自動的に削除されます。値が 0 の場合、システムは、そのカテゴリーのエントリーを削除しません。ストリング値は整数として 0 に評価されます。0 の代わりに「行わない」などの値を使用できます。
ビルド取り消しの確認頻度	システムがビルド取り消し要求を確認する間隔 (秒) が指定されます。
コンソール・ポート	Web サーバーが Build Forge の要求を listen するために使用するポート番号。
コンソール URL	<p>Web サーバーが Build Forge の要求を listen するために使用する URL。</p> <p>この値は、コンソールが 80 以外のポートで実行されている場合に設定する必要があります。これが設定されている場合は、その設定値でデフォルト・コンソール URL が指定変更されます。これは、<code>protocol://hostname[:port]</code> のような形式になります。例: <code>http://myHost:81</code>。</p>
失敗時のステップ部分の続行	同じステップでの複数のコマンド部分の実行動作を決定します。デフォルトの No に設定した場合、ステップ内のコマンド部分が失敗すると、そのステップは失敗します。Yes に設定した場合、コマンド部分が失敗してもステップは失敗しません。ステップ内の他のコマンド部分を実行できます。コマンド部分について詳しくは、インフォメーション・センターのトピック『システムがステップを部分に分割する方法』(「開発 (Developing)」 > 「ステップの処理 (Working with steps)」 > 「実行フローの制御 (Controlling execution flow)」) を参照してください。
欠落バスの作成	デフォルト: 「いいえ」。「はい」の場合は、プロジェクトのバスがまだ存在していなければ、そのバスがシステムによって作成されます。
データベース・サイズしきい値	<p>データベース・サイズのしきい値。この値に達するとコンソールは通知を送信します。</p> <p><b>要確認:</b> このしきい値に近づく、ユーザー・インターフェースのパフォーマンスが低下します。データベースのクリーンアップを実行するか、しきい値のサイズを増やすと、パフォーマンスが復元されます。</p> <p>デフォルト: 2G。</p> <p><b>注:</b> Java エンジンではサポートされていません。</p>

設定	説明
データベース・サイズしきい値の通知	<p>データベース・サイズしきい値に達した場合の E メール送信先となるユーザー名または通知グループ。ユーザー名が使用される場合、ユーザーの「E メール」フィールドのアドレスが使用されます。「管理」 &gt; 「ユーザー」を参照してください。</p> <p>注: Java エンジンではサポートされていません。</p>
日付を一度設定 (Date Once)	<p>.date コマンド値を、ジョブごとに一度設定するか、あるいはステップごとに一度設定するかを決定します。動作は、以下のように、割り当てられた値によって異なります。</p> <ul style="list-style-type: none"> <li>デフォルト: いいえ。 .date コマンド値はステップごとに一度設定されます。ステップごとに、エージェント・マシンのタイムゾーンを使用して .date 値が再評価されます。エージェントが値を提供します。再始動しても .date 値は維持されます。値は初期実行の場合と同じように再評価されます。変数が同じでも、値は使用されるステップごとに異なります。これは、.date 値がステップごとに再解釈されるためです。</li> <li>はい: 現行 .date コマンド値はジョブごとに 1 回設定されます。この日付値は、エージェントのタイムゾーンではなくエンジン・ホストのタイムゾーンを使用して生成されます。再始動しても .date 値は維持されます。</li> </ul> <p>.email コマンドや通知テンプレートなどの E メール通知に .date を含む環境変数を使用する場合は、「はい」を入力します。その際、E メールで使われる環境変数では、エンジンのタイムゾーンが使用されます。</p>
デフォルト・エージェント・ポート	<p>エージェントへの接続を確立するために使用するデフォルト・ポート番号を設定します。</p>
デフォルト・インポート・クラス	<p>インポートされたプロジェクトに、定義済みのクラスがない場合、または存在しない定義クラスがある場合に使用されるクラス。</p> <p>デフォルト: 「実動」</p>

設定	説明
ダイジェスト・アルゴリズム	<p>メッセージ・ダイジェストのアルゴリズムです。エージェント構成にも、対応する <code>digest_algorithm</code> 設定があります。有効な値は <i>SHA1</i> および <i>SHA2</i> です。</p> <p>デフォルトのアルゴリズム: <i>SHA1</i></p> <p>注: Secure Hash Algorithm 2 (SHA2) を使用して管理コンソールおよびエージェントでパスワード暗号化を有効にする場合は、パスワード暗号化プロパティ・ファイル <code>bfpwcrypt.conf</code> を更新してください。</p> <p>注: Perl エンジンでは、メッセージ・ダイジェストでの Secure Hash Algorithm 2 (SHA2) はサポートされていません。SHA2 ダイジェスト・サポートは、Build Forge 8.0 および Build Forge Agent 8.0 からの Java MJC 用です。</p> <p>パスワード暗号化が構成プロパティ・ファイル <code>bfpwcrypt.conf</code> で有効になっていない場合は、以下の手順で SHA2 のパスワード暗号化を有効にしてください。</p> <ol style="list-style-type: none"> <li>1. パスワード暗号化構成プロパティ・ファイルを名前変更します。例えば、<code>bfpwcrypt.conf</code> ファイルの名前を <code>bfpwcrypt.conf.sha1</code> に変更します。</li> <li>2. 「管理」 &gt; 「システム」にナビゲートし、「ダイジェスト・アルゴリズム」設定を <b>SHA2</b> に変更します。</li> <li>3. 管理コンソールを再始動します。新しい <code>bfpwcrypt.conf</code> ファイルが作成されます。</li> <li>4. パスワード暗号化を有効にします。「管理」 &gt; 「セキュリティ」にナビゲートし、「パスワード暗号化の有効化」設定を「はい」に設定します。「保存」をクリックし、「マスター <code>BFCClient.conf</code> の更新」をクリックします。</li> <li>5. オプション: 「管理」 &gt; 「セキュリティ」 &gt; 「鍵ストア」にナビゲートし、鍵ストアのパスワードを更新します。「管理」 &gt; 「サーバー」 &gt; 「サーバー認証、管理 (Server Auth, Administration)」 &gt; 「LDAP」および「管理」 &gt; 「ユーザー」に適宜ナビゲートし、それぞれのパスワードを更新します。</li> <li>6. エージェントでもパスワード暗号化を有効にする場合は、新たに生成された <code>bfpwcrypt.conf</code> ファイルを、エージェントのあるサーバーにコピーします。</li> <li>7. <code>ssl_key_password</code> が、<code>bfagent.conf</code> ファイルで以前に設定されたときに暗号化されていた場合は、パスワードを再暗号化する必要があります。例えば、<code>bfagent.conf</code> ファイル内の行の前にある <code>#</code> を削除することによって <code>digest_algorithm SHA2</code> を有効にしてから、コマンド行で <code>bfagent -e &lt;your password&gt;</code> を使用して再暗号化します。これにより、新しい暗号化パスワードが作成されます。この新しい暗号化パスワードを使用して、<code>bfagent.conf</code> ファイル内の <code>ssl_key_password</code> をリセットし、保存します。</li> <li>8. <code>bfagent</code> を再始動します。</li> </ol>

設定	説明
ダイジェスト・アルゴリズム (続き)	<p>注 (続き): パスワード暗号化が構成プロパティ・ファイル <code>bfpwcrypt.conf</code> で有効になっている場合は、以下の手順で SHA2 のパスワード暗号化を再度有効にしてください。</p> <ol style="list-style-type: none"> <li>1. パスワード暗号化を無効にします。「管理」 &gt; 「セキュリティ」にナビゲートし、「パスワード暗号化の有効化」設定を <b>No</b> に設定します。「保存」をクリックし、「マスター <code>BFCClient.conf</code> の更新」をクリックして、<code>bfcclient.conf</code> ファイル内の <code>bf_keystore_password</code> が <code>{bfcrypt:xxx..}yyy...</code> の形式ではないことを確認してください。</li> <li>2. パスワード暗号化が初めて有効化されて以降にサーバー認証、LDAP、およびユーザー・パスワードが追加または更新された場合は、「管理」 &gt; 「セキュリティ」 &gt; 「鍵ストア」にナビゲートし、各パスワードを更新します。</li> <li>3. パスワード暗号化構成プロパティ・ファイルを名前変更します。例えば、<code>bfpwcrypt.conf</code> ファイルの名前を <code>bfpwcrypt.conf.sha1</code> に変更します。</li> <li>4. 「管理」 &gt; 「システム」にナビゲートし、「ダイジェスト・アルゴリズム」設定を <b>SHA2</b> に変更します。</li> <li>5. 管理コンソールを再始動します。新しい <code>bfpwcrypt.conf</code> ファイルが作成されます。</li> <li>6. パスワード暗号化を有効にします。「管理」 &gt; 「セキュリティ」にナビゲートし、「パスワード暗号化の有効化」設定を「はい」に設定します。「保存」をクリックし、「マスター <code>BFCClient.conf</code> の更新」をクリックします。</li> <li>7. オプション: 「管理」 &gt; 「セキュリティ」 &gt; 「鍵ストア」にナビゲートし、鍵ストアのパスワードを更新します。「管理」 &gt; 「サーバー」 &gt; 「サーバー認証、管理 (Server Auth, Administration)」 &gt; 「LDAP」および「管理」 &gt; 「ユーザー」に適宜ナビゲートし、それぞれのパスワードを更新します。</li> <li>8. エージェントでもパスワード暗号化を有効にする場合は、新たに生成された <code>bfpwcrypt.conf</code> ファイルを、エージェントのあるサーバーにコピーします。</li> <li>9. <code>ssl_key_password</code> が、<code>bfaagent.conf</code> ファイルで以前に設定されたときに暗号化されていた場合は、パスワードを再暗号化する必要があります。例えば、<code>bfaagent.conf</code> ファイル内の行の前にある <code>#</code> を削除することによって <code>digest_algorithm SHA2</code> を有効にしてから、コマンド行で <code>bfaagent -e &lt;your password&gt;</code> を使用して再暗号化します。これにより、新しい暗号化パスワードが作成されます。この新しい暗号化パスワードを使用して、<code>bfaagent.conf</code> ファイル内の <code>ssl_key_password</code> をリセットし、保存します。</li> <li>10. <code>bfaagent</code> を再始動します。</li> </ol>

設定	説明
即時開始を使用可能に設定	<p>デフォルト: 「いいえ」。すべてのプロジェクトに  が表示されます。プロジェクトをクリックすると、そのプロジェクト用に組み込まれているすべての変数が「変更が必要」変数タイプについて検査されます。プロジェクトは、「変更が必要」タイプの変数が含まれていない場合に、開始されます。プロジェクトに「変更が必要」変数タイプが含まれている場合、そのプロジェクトは開始されません。プロジェクトが開始されない理由がダイアログ・ボックスに示されます。アイコンが  に変わります。</p> <p><b>注:</b> プロジェクトに定義されている変数のみ検査されます。プロジェクト内のステップに定義されている変数は検査されません。 329 ページの『ジョブの開始時に行われるプロジェクト変数の変更』も参照してください。</p> <p>この設定が「はい」に設定されている場合は、いずれかの変数が「変更が必要」タイプに設定されているかどうかを判別するために、「プロジェクト」ページですべての環境変数がプロジェクトに対して検査されます。即時に開始できるプロジェクトはアイコン  で示されます。</p>
ハード実行上限数	<p>デフォルト: 「いいえ」。「はい」の場合は、スケジュールされたプロジェクト・ビルドを起動してもプロジェクト設定「<b>実行上限数</b>」に違反しなければ、そのプロジェクト・ビルドがシステムから起動されます。「いいえ」の場合は、スケジュールされたビルドに関してプロジェクト設定「<b>実行上限数</b>」はシステムで無視されます。</p>
デフォルトのセキュア・アクセス・グループのインポート	<p>「<b>セキュア・アクセスによるインポート</b>」設定が「Y」に設定されている場合、この設定は、インポートされたプロジェクトのデフォルト・アクセス・グループを示します。</p>
セキュアでないデフォルトのアクセス・グループのインポート	<p>「<b>セキュア・アクセスによるインポート</b>」設定が「N」に設定されている場合、この設定は、インポートされたプロジェクトのデフォルト・アクセス・グループを指定します。デフォルト・グループは、インポート・ファイルにアクセス・グループがない場合にのみ使用されます。</p>
セキュア・アクセスによるインポート	<p>Y に設定された場合、システムは、前の設定でリストされたデフォルトのアクセス・グループを、インポートされたデータ・オブジェクトに割り当てます。この値は、データのインポートによってセキュリティーが指定変更されないように、インポート対象の XML ファイルに指定されているアクセス・グループを指定変更します。N に設定された場合、システムは、インポートされるファイルにあるアクセス・グループの設定を優先します。</p>
継承タグ	<p>これが「はい」に設定されている場合、チェーンによって起動されたジョブは呼び出し元と同じジョブ・タグを使用しなければなりません。プロジェクト MasterProject の BUILD_15 がプロジェクト ComponentProject を呼び出す場合は、ComponentProject の実行に対するジョブ・タグおよびジョブ・ディレクトリー名は BUILD_15 になります。</p> <p><b>注:</b> 呼び出されたプロジェクトは必ず呼び出し元のオリジナルのタグを常に (呼び出し元のタグが実行中に変わったとしても) 継承します。例えば、<b>.retag</b> コマンドの結果として、呼び出されたプロジェクトは、呼び出し元が開始されたときのタグを保持します。</p>
無効な相対ディレクトリー文字	<p>プロジェクト名で使用された場合にシステムが下線に変更する文字が設定されます。</p>
LASTRUN フォーマット	<p><b>.date</b> コマンドに対して定義された日付フォーマット文字を使用して <b>BF_LASTRUN</b> フォーマット環境変数の値を入力します。 438 ページの『<b>.date</b>』を参照してください。</p>



設定	説明
ライセンス・サーバー	ライセンス・サーバー・ホスト名。インストール中に設定されます。例: myhost.mycompany.com。この値には、ポート番号が含まれる場合があります。例: myhost.mycompany.com:80。ライセンス・サーバーを変更するには、26 ページの『管理コンソール用のライセンス・サーバーの変更』を参照してください。
デバッグ・モードのリンク	これが「はい」に設定されている場合、アダプター・リンクが定義されているジョブは、関連プロジェクトを実行するのではなく、リンクのテストを行います。ジョブの出力には、アダプターからの出力を持つ単一ステップが含まれます。このデータは、アダプターのインターフェースをトラブルシューティングする際に役立ちます。 <b>注:</b> アダプター・リンクの状態を「デバッグ」に設定することによって、個々のアダプター・リンクにデバッグを設定できます。この状態は、「デバッグ・モードのリンク」設定よりも優先されます。
マニュアル・ジョブのリンク	プロジェクトをスケジューラーから実行するのではなく手動でクイック・スタートする場合にシステムがアダプター・リンクを使用してアダプターを実行するかどうかが決定されます。  リンク検査により、ジョブの BOM に追加の出力が生成される場合があります。  これが「N」に設定されている場合は、ジョブの開始時にリンクは検査も実行もされません。
最大コンソール・プロセス	コンソールが同時に実行するプロセスの最大数を設定します。コンソールのアクティビティでは、一般的なスロットル (数を調節する弁) として使用されます。システムは、各プロセスの ID をデータベースに保管し、新規の外部プロセスを起動する前に合計を確認することで、プロセスを管理します。この値は「実行キュー・サイズ」設定よりも 5 以上大きい値になるようにしてください。そうしないと、システムは実行キューをサポートするだけの十分なプロセスを実行できません。 <b>注:</b> Java エンジンではサポートされていません。
インライン化の最大の深さ	これは、プロジェクトのインライン化に対してシステムが許可するレベル数を制御して、プロジェクトが極端に深くネストされないようにします。デフォルト値は 32 です。この値が 0 に設定された場合、システムは 32 を使用します。システムがインライン化の上限に達すると、インライン化されるプロジェクトのうち、その上限を超えるプロジェクトは実行されません。そのステップは、ステップの包含先のプロジェクトに挿入されません。「インラインは中止されました (inline abandoned)」というメッセージがシステム・メッセージ・リストに書き込まれます。 <b>注:</b> Java エンジンではサポートされていません。
サーバーの最大同時テスト数	同時に実行できるサーバー・テストの数が指定されます。システム・リソースによっては、同時に実行されるサーバー・テストが多すぎると、コンソールの速度が極端に低下したり、ロックしたりします。
最大同時ページ数	同時に実行できるページの数を制御します。希望数のビルドをページできます。ただし、「最大同時ページ数」フィールドの値を超えた分のビルドは一斉に削除されます。デフォルト: 20。
最大リフレッシュ数	ページが自動的にリフレッシュする最大回数。デフォルト: 50。
チェーニング時のクラスの指定変更	システムが、チェーニングされたプロジェクトのクラスを呼び出し元のクラスで置き換えるかどうか決定されます。デフォルト値「Y」の場合、システムは、チェーニングされたプロジェクトのクラスを指定変更し、代わりに呼び出し元のクラスを使用します。
パスワードの有効期限日数	パスワードの有効期限が設定されているユーザーが自分のパスワードを変更しなければならなくなるまでの日数が示されます。この日数が経過すると、ユーザーは次のログイン時に自分のパスワードを変更しなければなりません。

設定	説明
パスワード・フォーマット	<p>以下のように、ピリオドで区切られた、6 つのフィールドからなるフォーマット・ストリングを使用して、ユーザー・パスワードの要件を指定します。</p> <p><code>length.char_types.upper.lower.numeric.special</code></p> <p>例: <code>5.2.u1.l1.n1.s1</code>。</p> <p>最初の 2 つのフィールドでは、以下を指定します。</p> <ul style="list-style-type: none"> <li>パスワードの最小長 (文字)</li> <li>残りの 4 つのフィールドで使用する文字タイプの最小数 (1 から 4 までの整数)</li> </ul> <p>残りのフィールドは、文字タイプと文字数の要件を指定します。各フィールドは、タイプと数字を含みます。</p> <ul style="list-style-type: none"> <li>タイプ: <code>u</code> (大文字)、<code>l</code> (小文字)、<code>n</code> (数字)、または <code>s</code> (特殊文字) のいずれか 1 つ。大文字 (<code>U</code>、<code>L</code>、<code>N</code>、<code>S</code>) は、その文字が必須であることを示しています。小文字 (<code>u</code>、<code>l</code>、<code>n</code>、<code>s</code>) は、その文字がオプションであることを示しています。</li> <li>数字: 必須。その文字が使用されている場合に、必要となる文字の数が示されます。</li> </ul> <p>タイプは、以下のようになります。</p> <ul style="list-style-type: none"> <li><code>U</code> または <code>u</code> は、大文字を示します。大文字には、それぞれのロケールで文字とみなされる、小文字以外のすべての文字が含まれます。具体的に説明すると、大文字やタイトル・ケースの文字、または単一ケース言語 (例: 中国語) のすべての文字が含まれます。</li> <li><code>L</code> または <code>l</code> は、小文字を示します。このタイプには、それぞれのロケールで小文字とみなされるすべての文字が含まれます。</li> <li><code>N</code> または <code>n</code> は、数字を示します。数字には、それぞれのロケールで数字とみなされるすべての文字が含まれます。</li> <li><code>S</code> または <code>s</code> は、特殊文字を示します。上記の 3 つのカテゴリに当てはまらないすべての文字です。文字や数字ではないすべての文字が含まれます。</li> </ul> <p>例: ストリング <code>5.2.u1.l1.n1.s1</code> は、以下のパスワード要件を示しています。</p> <ul style="list-style-type: none"> <li>長さは 5 文字以上である。</li> <li>4 つのカテゴリ (大文字、小文字、数字、特殊文字) のうち、2 つ以上のカテゴリの文字を含まなければならない。</li> <li>タイプごとに、そのタイプの 1 文字は、要件を反映し一致していると見なされる。</li> </ul> <p><code>abClx</code> や <code>Abc2%</code> などのパスワードがこれらの要件を満たします。</p>
Build Forge エンジンの一時停止	<p><code>Y</code> に設定されると、システムは、現在のジョブをすべて完了してから、エンジンを一時停止します。<code>N</code> に設定されると、通常の実行に戻ります。</p>
共通ホスト名	<p>設定された場合、システムは、通知テンプレートでこの設定の値を、<code>CONSOLEHOST</code> 変数のサーバー・ホスト名に代入します。</p>
ページ確認時間	<p>システムがページ対象のジョブを検査する頻度が検査間隔 (分) で設定されます。</p>

設定	説明
QuickReport 共通ディレクトリー	<p>公開レポート設計のファイル・システム・ロケーション。</p> <p>7.1 では、このシステム設定を使用して、公開レポートの完全修飾ロケーションを指定します。レポート設計を自動的にマイグレーションするためには、レポート設計をこのディレクトリーに入れる必要があります。</p> <p>以前のリリースでは、デフォルトのファイル・ロケーション (<code>../reports/public</code>) は、アプリケーション・サーバーのインストール・ディレクトリー (例えば、<code>&lt;bfinstall&gt;/Apache/tomcat/webapps/quickReport</code>) を基準とする相対ロケーションでした。</p>
QuickReport 一時ディレクトリー	<p>7.1 では、このディレクトリーを使用して、サービス層コンポーネントと同じホスト上にある完全修飾ディレクトリーを指定します。サービス層では、この作業ディレクトリーを使用して、データベースに正常にマイグレーションされたレポート設計をリストします。</p> <p>以前のリリースでは、このディレクトリーは、クイック・レポートのレポート設計をファイル・システムの公開ディレクトリーまたはプライベート・ディレクトリーに保存する前に一時的に保管するために使用されていました。</p>
QuickReport ユーザー・ディレクトリー	<p>私用レポート設計のファイル・システム・ロケーション。</p> <p>7.1 では、このシステム設定を使用して、プライベート・レポートの完全修飾ロケーションを指定します。レポート設計を自動的にマイグレーションするためには、レポート設計をこのディレクトリーに入れる必要があります。</p> <p>以前のリリースでは、プライベート・レポートのデフォルトのファイル・ロケーション (<code>../reports/users</code>) は、アプリケーション・サーバーのインストール・ディレクトリー (例えば、<code>&lt;bfinstall&gt;/Apache/tomcat/webapps/quickReport</code>) を基準とする相対ロケーションでした。</p>
言語バックの再ロード	<p>デフォルト: 「いいえ」。これが「はい」に設定されている場合、コンソールは再始動時に言語バックを再ロードして、この値を「いいえ」にリセットします。これはバージョン 7.0.1 以降では不要になりました。</p>
アダプター・テンプレートのリセット	<p>この設定を使用して、(更新で生じる変更を構成にコピーするために) アダプター・テンプレートをリセットします。これを使用するには、値を「はい」に設定し、1 分待ちます。システムは、テンプレートをリセットしてから、値を「いいえ」に設定し直します。</p>
サーバーのジョブ数のリセット	<p>設定に「はい」を使用すると、すべてのサーバーのジョブ数 (BF_JOBS) を同時にゼロにリセットできます。このリセットは、マニフェストの確認間隔の実行時に行われます。デフォルトは 10 秒間隔です。</p> <p>すべてのサーバーの BF_JOBS がリセットされた後に、「サーバーのジョブ数のリセット」の値がデフォルトの「いいえ」に戻ります。</p>
レポート・マイグレーションの再開	<p>デフォルト: 「いいえ」。7.1 では、サービス層コンポーネントを再始動せずにマイグレーションを開始する場合は、この値を「はい」に設定します。</p>
チェーン・リンクの実行	<p>起動されたチェーン・プロジェクトが、接続されている任意のアダプター・リンクも起動するかどうかを制御します。</p>
実行キュー・サイズ	<p>この値は、システムが同時に実行しようとするジョブの数を制限します。キューにある実行数がこの数に等しいかそれを超えた場合、システムは、ジョブの数がこの値を下回るまで、待機キュー から実行キュー への実行の移動を停止します。実行キュー・サイズを変更した場合は、最大コンソール・プロセス設定を確認してください。この設定は、実行キュー・サイズより 5 以上大きくなければなりません。</p>
始動環境変数の保存	<p>プロジェクトの手動開始ページにある、「環境変数の保存」チェック・ボックスのデフォルト値を制御します。設定が「Y」の場合は、このチェック・ボックスが選択されます。これはデフォルトです。それ以外の場合は、このチェック・ボックスは選択されません。「環境変数の保存 (Save Env)」チェック・ボックスを選択した場合、変更はすべて「開始」ページ上の環境変数と、データベース内の環境変数レコードに保存されます。今後のビルド実行はデフォルトでこの値になります。</p>

設定	説明
チェーニング前のサーバー環境変数	<p>システムがステップ・サーバー環境変数を前 (Y) に設定するか後 (N) に設定するかを決定します。この値により、ステップ内でのチェーニング・プロジェクト環境変数が設定されます。2 番目に処理された環境変数内の変数は、最初の環境変数内の変数をオーバーライドします。デフォルト値は Y です。これは、チェーニング・プロジェクト環境変数が 2 番目に処理されて、ステップ・サーバー環境変数をオーバーライドすることを示します。</p> <p><b>注:</b> Java エンジンではサポートされていません。</p>
サーバー読み取りタイムアウト	<p>システムがエージェントの読み取り試行を停止するまでの待機秒数。</p> <p><b>注:</b> 8.0 の新機能です。Java マスター・ジョブ・コントローラーにのみ使用されます。</p>
サーバー再試行	<p>すべてのセレクト条件に合致するサーバーが見つかったときに、システムがステップを停止して失敗とする前にシステムがサーバーにステップを割り振ろうとする回数が設定されます。</p>
サーバー書き込みタイムアウト	<p>システムがエージェントへの書き込み試行を停止するまでの待機秒数。</p> <p><b>注:</b> 8.0 の新機能です。Java マスター・ジョブ・コントローラーにのみ使用されます。</p>
サーバー・テスト頻度	<p>これは、サーバーのマニフェスト・データをテストしてリフレッシュする間隔を決定するために、使用可能なサーバーの数と一緒に使用されます。デフォルトは 120 分 (すなわち 2 時間) です。値が 0 の場合は、サーバーを検査する必要がないことになります。</p> <p>これらの検査の際に、システムは使用可能なすべてのサーバーに問い合わせて次の内容を検証します。</p> <ul style="list-style-type: none"> <li>サーバーに接続可能であること</li> <li>サーバーのログイン情報が適正であること</li> <li>サーバーのマニフェスト・データが最新であること</li> </ul> <p>サーバー・テストは、1 分当たり少なくとも 1 台のサーバー で実行されます。サーバーの数がこの設定よりも多い場合は、1 分当たりのサーバー数は増えます。</p> <p>システムは、この間隔内でテストを均等に分散させます。例:</p> <ul style="list-style-type: none"> <li>120 台のサーバーがあり、間隔は 120 分に設定されています。システムは、1 分当たり 1 台のサーバー (120 台のサーバー/120 分) をテストしようとしています。</li> <li>12,000 台のサーバーがあり、間隔は 1200 分に設定されています。システムは、1 分当たり 10 台のサーバーのテストを試みます。</li> <li>10 台のサーバーがあり、間隔は 120 分に設定されています。1 分当たり 1 台のサーバーの検査が実行されます。このシナリオでは、リフレッシュ間隔中にサーバーが 12 回も検査されます。</li> </ul> <p>手動サーバー・テストは、コンソールから開始されます。このテストは、自動テストに優先します。</p> <p>サーバーのコレクターの複雑さは、スループットに影響することがあります。コレクターがデータを収集するために数多くの手動コマンドを実行する場合は、完了までの所要時間が 1 分を超えることがあります。</p>
サーバー使用接続タイムアウト	<p>エージェント接続が開始されてから接続が失敗するまでに管理コンソールが接続の確立を待機する最大時間 (秒) が示されます。また、ソケット・エラーが発生したり、タイムアウト期限に達する前にコードが失敗したりした場合も、接続は失敗する可能性があります。</p> <p>エージェントへの既存接続のタイムアウト値が示されます。エージェントは 15 秒おきに管理コンソールに問い合わせるように設計されています。タイムアウト期間中に問い合わせがない場合は、エージェントが停止しているか、またはネットワーク通信問題が発生している可能性があります。この値を超えると、ステップは失敗します。</p>

設定	説明
サーバー待機時間	サーバーが使用可能になったかどうかを判別する検査の実行間隔 (秒数) を設定します。 <b>注:</b> リリース 8.0 の新機能です。Java マスター・ジョブ・コントローラーにのみ使用されます。
サービス層認証サブリット URL	これを設定すると、プログラマチックに構成された URL が、サービス層認証サブリットに指定変更されます。  別名を使用している場合や、デフォルト以外のポートを使用している場合は、 <code>http://server:port/jas/AuthServlet</code> のフォーマットを使用して、この設定を更新する必要があります。
SMTP サーバー	E メール通知の送信時に SMTP サーバーとして使用されるコンピューターが設定されます。  デフォルト設定は <code>localhost</code> です。
スタック BuildForge 環境変数	通常システムは、チェーニングされたプロジェクトに渡される <code>BF_</code> 変数の名前を、 <code>BF_CALLER_</code> に変更します。設定は、チェーニングが 1 を超える深さのレベルに達している場合に、システムがその命名をスタックするかどうかを決定します。デフォルト値は <code>N</code> です。設定が <code>Y</code> に変更されると、深さのレベルが 2 の呼び出しプロジェクトから派生した <code>BF_TAG</code> 変数は、名前 <code>BF_CALLER_CALLER_TAG</code> を受け取ります。
ステップの最大再試行数	最初の試行が失敗した場合にステップがエージェントへの接続を試みる回数を制御します。指定された試行回数で接続しなかったステップは失敗します。
ユーザー認証のローカル保管	システムが LDAP または Active Directory のユーザー認証情報を、暗号化されたフォームでキャッシュに入れるかどうかが決まります。デフォルトは「はい」です。システムが関係するのは、LDAP またはアクティブ・ディレクトリーの認証を使用する場合のみです。この設定が「はい」の場合、システムは、ユーザー認証情報を、暗号化されたフォームでキャッシュに入れ、 <code>_USE_BFCREDS</code> 特殊変数および <code>_USE_BFCREDS_DOMAIN</code> 特殊変数で使します。 <b>注:</b> この値は、サーバーに対するユーザー認証に適用されます。この値を「いいえ」に変更すれば、キャッシングをオフにできます。ただし、キャッシングをオフにした場合、システムはサーバーへのログオン時に <code>_USE_BFCREDS</code> 特殊変数や <code>_USE_BFCREDS_DOMAIN</code> 特殊変数を利用してユーザー資格情報を使用することはできなくなります。
システム・アラート E メール	システムは、アラート E メール・メッセージを、この設定で定義されたアドレスに送信します。  デフォルトは <code>root@localhost</code> です。以下の E メール・アドレス・フォーマットがサポートされています。  <pre>username@host.com nameusername@host.com name "&lt;username@host.com&gt;" name &lt;username@host.com&gt; name "username@host.com"</pre> <b>説明:</b>  <code>username</code> は、E メールのユーザー名です。  <code>name</code> は、任意のストリング (通常、何らかの形で名前を表すもの) です。  複数のアドレスに送信するには、コンマ (,) またはセミコロン (;) でそれらを区切ってください。

設定	説明
システム・アラート送信元	<p>システムがアラート E メール・メッセージを送信するとき、この設定で定義されたアドレスを、送信者として使用します。</p> <p>デフォルトは <code>root@localhost</code> です。以下の E メール・アドレス・フォーマットがサポートされています。</p> <pre>username@host.com name username@host.com name "&lt;username@host.com&gt;" name &lt;username@host.com&gt; name "username@host.com"</pre> <p>説明:</p> <p><code>username</code> は、Eメールのユーザー名です。</p> <p><code>name</code> は、任意のストリング (通常、何らかの形で名前を表すもの) です。</p>
システム広域ログイン・メッセージ	ログイン・フォームの上に表示されるメッセージを定義できます。
システム広域ユーザー・メッセージ	各ページの最上部でナビゲーション・ボタンの直下に表示されるメッセージを定義できます。
タグ: 日付書式	<p>BF_D タグ変数で、日付を表示するために使用する書式を定義します。文字 <code>y</code>、<code>m</code>、および <code>d</code> を、年、月、および日を表す変数として使用します。特殊文字を分離文字として使用すれば、この値を希望のフォーマットで表示できます。例えば、2005 年 9 月 21 日は次のように表します。</p> <p>書式ストリング....出力</p> <pre>ymd...050921 m/d/y...09/21/05</pre>
タグ: 時刻書式	<p>BF_T タグ変数で、時間を表示するために使用する書式を定義します。「タグ: 時刻形式」では、時、分、および秒を表すために文字 <code>h</code>、<code>m</code>、および <code>s</code> が使用されます。 <code>h:m:s</code> という設定の場合、出力は <code>12:53:42</code> のようになります。</p>
メール・テンプレートのログ量のテール	<code>TAILNORMALLOG</code> 変数が通知テンプレートで使用されるときに、通知で表示される、ログの末尾からの行数を設定します。
スレッドの終了	スレッド化ステップがどのようなときに失敗するかを決定します。同じプロジェクトに含まれる他のすべてのアクティブ・スレッド・ブロックが停止します。
トレース・エージェント・データ・リンク	<p>コンソールとエージェント間通信の詳細トレースを、オンにするかどうかを示します。</p> <p>デフォルト: いいえ</p>
Java マスター・ジョブ・コントローラーをジョブ実行エンジンとして使用	<p>「はい」に設定すると、システムは Java マスター・ジョブ・コントローラーを使用してマスター・ジョブ・コントローラー・プロジェクトを実行します。 Perl エンジン・プロジェクトは開始できません。新規作成されたプロジェクトは、Perl エンジンではなく、デフォルトでマスター・ジョブ・コントローラーを使用するように設定されます。</p> <p>「いいえ」に設定すると、システムは Perl エンジン・プロジェクトを実行する場合にのみ Perl エンジンを使用します。</p>

## その他のシステム設定

システムの外部で設定されている構成設定値があります。

### 入出力タイムアウト

入出力タイムアウトは、デフォルトで 300 秒に設定されています。この値は、オペレーティング・システム変数として

`BF_SOCKET_SELECT_TIMEOUT` を作成し、それに値を (秒単位で) 指定することにより、調整可能です。この値を 60 秒未満に設定しないでくださ



い。この値は、始動時にのみ読み取られます。したがって、値を変更する場合には、システムを停止して値を変更した後で、再度システムを始動する必要があります。この値が小さすぎると、多数の要求が同時に発生した際に（複数のジョブを一度に開始した場合など）システムがハングしたように見えることがあります。ほとんどの場合は、デフォルト値で問題ありません。

---

## 監査ロギング

監査ロギングでは、監査ロギングの管理および監査ログの表示を行うコマンド行インターフェースが提供されます。監査ログには、システム内のオブジェクトに対する変更の履歴が示されます。これらはフルテキスト索引に保管されます。

監査ロギングを使用した場合、トラッキング対象のオブジェクトについて、メッセージング機能に代わってロギング機能が適用されます。ロギング機能が適用されない場合は、「管理」 > 「メッセージ」でメッセージが表示されます。

**重要:** 監査ロギングは、実行中のシステムで直接実行されている操作でのみ機能します。システムにインポートされたデータでは機能しません。bfimport 機能はデータベースを直接更新し、監査ロギング機能と相互作用しません。

### 監査ポリシーの管理

監査ポリシーは、監査がどの程度行われるかを制御します。これは、監査ポリシーのシステム構成の設定によって決まります。

なし 監査情報は記録されません。

#### セキュリティ

セキュリティに関する情報のみが記録されます。例えば、アクセス・グループの作成やユーザー・セッションのアクティビティに関する情報などです。

**基本** これはデフォルト設定です。監査ポリシーの値が無効な場合にも使用されます。ユーザー、オブジェクト・タイプ、UUID、アクションなどの最小限の情報のみが記録されます。

**詳細** 古い値と新しい値を含む、プロパティの変更に関する詳細情報が記録されます。すべてのオブジェクトがこの情報を提供できるわけではありません。この情報を提供できないオブジェクトでは、BRIEF ポリシーと DETAILED ポリシーの間に違いはありません。「監査の最大ストリング長 (Audit Max String Length)」システム構成設定を使用して、保管されるデータの量を制限できます。

「監査の最大ストリング長 (Audit Max String Length)」システム構成設定では、ストリング・データ・フィールドの長さをキャッピングすることで、詳細な (DETAILED) 変更情報として記録できる情報量を制御します。監査項目がこの文字数を超えた場合は、ストリング値の代わりに SHA メッセージ・ダイジェストが保管されます。パスワードなどの機密情報を含むフィールドでは、ストリング長に関係なく、メッセージ・ダイジェストが使用されます。



## 監査情報の表示

コマンド行インターフェースを使用して、表示する情報およびその情報のフォーマット方法を指定することができます。

「すべての監査ログの表示」権限が必要です。デフォルトでは、セキュリティー・アクセス・グループとビルド・エンジニア・アクセス・グループに対して、この権限が割り当てられます。

使用法: `bfauditlog [connection-options] [command-options] [query]`

リテラルの実行可能ファイルは以下のとおりです。

- Windows システムの場合: `bfauditlog.bat`
- UNIX システムおよび Linux システムの場合: `bfauditlog.sh`

### 接続オプション

**-H** *hostname* | **--hostname=hostname**

接続に使用するリモート・ホスト名。デフォルトは `localhost` です。

**-P** *port* | **--port=port**

接続に使用する通信ポート。デフォルトは `3966` です。

**-d** *domain* | **--domain=domain**

認証に使用するドメインまたは領域。デフォルト: なし。

**-u** *login* | **--user=login**

ユーザー認証に使用するユーザー名。デフォルトは `root` です。

**-p** *password* | **--password=password**

認証ユーザー用のパスワード。デフォルトは `root` です。

**-E** *encoding* | **--encoding=encoding**

使用する出力文字セット・エンコード。指定されたエンコードが無効な場合は、使用可能なすべてのエンコードがリストされて、プログラムが終了します。要求されたエンコードでテキストを表示できるかどうかは、クライアントで使用されている端末ソフトウェアの機能に依存します。デフォルト設定は、ご使用のオペレーティング・システムと JVM のサポートによって異なります。

### コマンド・オプション

**-h** ヘルプを表示します。

**-b** | **--bare**

最低限の情報を出力します (`bare`)。正確な出力結果は出力フォーマットによって異なります (『`-t`』を参照)。例えば、XML 出力の場合、開始タグ、終了タグ、および行数は省略されます。CSV の場合は、行数と列名が省略されます。

**-c** *auditClass[,auditClass ...]* | **--audit.class=class-list**

照会で使用する監査クラスのリストを指定します。複数のクラス名がある場合は、コンマで区切ります。複数回指定すると、指定されたすべての監査クラスの和集合が使用されます。オプションが指定されなかった場合は、ユーザーが指定できるすべての監査クラスが使用されます。使用可能な監査クラスをリストするには、`-L` オプションを使用します。デフォルトは、ユーザーが表示する権限があるすべての監査クラスです。

**-f** *field[,field ...]* | **--field=field-list**

出力に含めるフィールドを指定します。複数のフィールド名がある場合は、コンマで区切ります。使用可能なフィールドをリストするには、**-L** オプションを使用します。

**-s** *field[+|-][,field[+|-] ...]* | **--field=field-list**

出力のソートに使用するフィールドをリストします。デフォルトでは、`audit.stamp` の降順で出力がソートされます。+ マーカーを指定すると昇順になります。- マーカーを指定すると降順になります。複数のフィールド名がある場合は、コンマで区切ります。デフォルトは `-audit.stamp` です。

**-l** [*offset*,]*count* | **--limit=[offset,]*count***

結果のページ編集に使用します。オフセットとカウントは項目数で示されます。例えば、「`-l 100,50`」は 100 番目の項目の後に 50 個の項目が表示されることを意味します。`bare` 以外の出力フォーマットでは、一致する項目の総数が出力に含まれます。

**-a** *user[,user ...]* | **--audit.user=user-list**

出力を、指定されたユーザーに関連付けられた項目に制限します。複数のユーザーがある場合は、コンマで区切ります。**-a** オプションを複数回使用した場合は、そのオプションで指定されたすべてのユーザーの和集合が使用されます。ユーザーは、ログイン ID または UUID によって指定できます。**--audit-user** を指定するときに `user` 引数を使用しなかった場合は、認証に使用するユーザーが使用されます。

**-t** *format* | **--format=format**

出力に使用するフォーマット (CSV、XML、JSON、または TEXT)。デフォルトは TEXT です。

**-L** | **--list**

ユーザーが使用可能な監査クラスをリストします。アスタリスク (\*) のマークが付いているものが、ユーザーがアクセスできる監査クラスです。**-c** (監査クラスのリスト) と一緒に使用した場合は、各クラスの使用可能なフィールドのリストが代わりに表示されます。このオプションを指定するときに、その他のコマンド・オプションを使用しても効果はありません。`ViewAllAudits` の権限を持つユーザーは、任意のクラスの監査ログを表示できます。その他のユーザーは、**-a** オプションを使用することで、自身の監査ログのみを表示できます。

## 照会

照会は、Web 検索エンジンで一般的に使用されるフォーマット (仕様をスペースで区切ってリストする) を使用するストリングです。仕様は「`field:value`」という形式になっており、指定したフィールドの値が検索されます。照会は指定されていないが、少なくとも 1 つのオプションが指定されている場合は、一致するすべての監査ログが表示されます。監査ユーザーと監査クラスを照会で指定することはできません。それらを指定するには、コマンド・オプションを使用します。

注: Windows システムでは二重引用符を使用し、UNIX システムと Linux システムでは単一引用符を使用してください。

## 例

1. 次の例では、Project 監査クラスと Step 監査クラスの既知の照会フィールドがすべてリストされます。

- Windows

```
$ bfauditlog.bat -c Project,Step -L
```

- UNIX および Linux

```
$ bfauditlog.sh -c Project,Step -L
```

2. 次の例では、プロジェクト作成についての最新の監査ログ (コンマ区切り変数の出力フォーマットを使用し、列見出しと合計行数を含まないもの) が 10 個表示されます。レコードごとに、プロジェクトの固有 ID と名前のみが表示されます。

- Windows

```
$ bfauditlog.bat -c Project -t CSV -f audit.target,  
sxProjectName --bare -l 10 "audit.type:Create"
```

- UNIX および Linux

```
$ bfauditlog.sh -c Project -t CSV -f audit.target,  
sxProjectName --bare -l 10 'audit.type:Create'
```

3. 次の例では、指定されたドメイン、ユーザー名、およびパスワードを使用してログインした後の、'buildforge.example.com' マシン上のインストール済み環境から最新の監査ログが 100 個取得されます。

- Windows

```
$ bfauditlog.bat -H buildforge.example.com  
-d ExampleLDAP -u apiuser -p "My$ecret" -l 100
```

- UNIX および Linux

```
$ bfauditlog.sh -H buildforge.example.com  
-d ExampleLDAP -u apiuser -p 'My$ecret' -l 100
```

4. 次の例では、接続を認証したユーザーのタイムゾーン設定に従って、2011 年 5 月 23 日から 2011 年 5 月 27 日までに記録された監査ログが取得されます。

- Windows

```
$ bfauditlog.bat "audit.stamp:[2011/05/23 TO 2011/05/27]"
```

- UNIX および Linux

```
$ bfauditlog.sh 'audit.stamp:[2011/05/23 TO 2011/05/27]'
```

5. 次の例では、アクセス・グループ・レベルが古い値の 4、5、または 6 から新しい値の 3 に変更されたプロジェクトの監査ログが取得されます。大括弧を使用して包括的な範囲を示しているため、'[4 TO 6]' は値 4 と 6 を突き合わせに含めるという意味であることに注意してください。また、古いレベル値と新しいレベル値を検索に使用できるのは、監査ポリシーが 'DETAILED' に設定されている場合のみであることに注意してください。

- Windows

```
$ bfauditlog.bat -c Project "ioLevel:[4 TO 6] AND inLevel:3"
```

- UNIX および Linux

```
$ bfauditlog.sh -c Project 'ioLevel:[4 TO 6] AND inLevel:3'
```

6. 次の例では、アクセス・グループ・レベルが古い値の 4 または 5 から新しい値の 2 または 3 に変更されたプロジェクトの監査ログが取得されます。中括弧を使用して排他的な範囲を示しているため、'{3 TO 6}' は値 3 と 6 を突き合わせに含めないという意味であることに注意してください。また、古いレベル値と新

しいレベル値を検索に使用できるのは、監査ポリシーが 'DETAILED' に設定されている場合のみであることにも注意してください。

- Windows

```
$ bfauditlog.bat -c Project "ioLevel:{3 TO 6} AND inLevel:(2 OR 3)"
```

- UNIX および Linux

```
$ bfauditlog.sh -c Project 'ioLevel:{3 TO 6} AND inLevel:(2 OR 3)'
```

7. 以下の例では、照会は、監査ユーザーの root と sv の結果に制限され、出力は CSV ファイルに生成されます。以下のように、プロジェクト・クラスとステップ・クラスのデータが返され、クラス、タイプ、ターゲット、ユーザー、およびタイム・スタンプの情報が含まれます。

- Windows

```
$ bfauditlog.bat -c Project,Step -f audit.class,  
audit.type,audit.target,audit.user,audit.stamp  
-a root,sv -l 10,5 -t CSV
```

- UNIX および Linux

```
$ bfauditlog.sh -c Project,Step -f audit.class,  
audit.type,audit.target,audit.user,audit.stamp  
-a root,sv -l 10,5 -t CSV
```

## クラス

クラスは、監査ログ・エントリーのラベルです。ユーザーが使用可能なクラスをリストするには、-L オプションを使用します。

## フィールド

フィールドには、各監査ログ・レコードに割り当てられた情報が示されます。

- audit.class: プロジェクト (ストリング)
- audit.stamp: タイム・スタンプ (整数)
- audit.stamp.localized: audit.stamp 整数値を表すローカライズされたストリング
- audit.type: 更新 (ストリング)
- audit.target: プロジェクトの UUID (ストリング)
- audit.user: 更新を実行したユーザーの UUID

監査ログ・レコードの一部のフィールドには、その状態 (o: 古い, n: 新しい, x: 拡張) とデータ型 (s: ストリング, i: 整数) が反映されます。例:

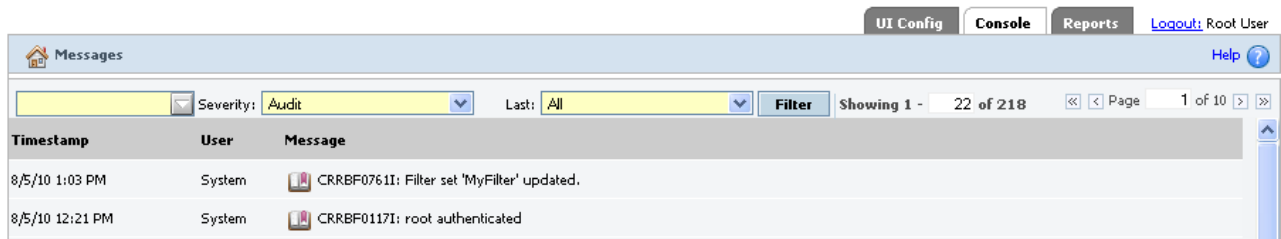
```
soDescription: 古いプロジェクトの説明 (ストリング)  
snDescription: 新しいプロジェクトの説明 (ストリング)  
soSelectorUuid: 古いセレクトターの UUID (ストリング)  
snSelectorUuid: 新しいセレクトターの UUID (ストリング)  
ioRunLimit: 古い実行上限数 (整数)  
inRunLimit: 新しい実行上限数 (整数)
```

---

## メッセージ

システム・メッセージ・ページには、システムのログとユーザー処置が表示されます。ユーザーは、これらを検討することができます。

「管理」 > 「メッセージ」を選択して、メッセージのリストを表示します。これらのメッセージは、処理の情報証跡を形成します。



Timestamp	User	Message
8/5/10 1:03 PM	System	CRRBF0761I: Filter set 'MyFilter' updated.
8/5/10 12:21 PM	System	CRRBF0117I: root authenticated

提供されている以下の 1 つ以上のコントロールを使用し、「フィルター」をクリックすることにより、リストをフィルターに掛けることができます。

- テキスト・ボックス: 検索対象のストリングを入力します。
- 重大度: 次のいずれかを選択します。
  - すべて (デフォルト)
  - エラー
  - 情報
  - 警告
  - 監査
- 期間 (Last): 次のいずれかを選択します。
  - すべて (デフォルト)
  - 12 時間 (12 hours)
  - 24 時間 (24 hours)
  - 2 日 (2 days)
  - 3 日 (3 days)
  - 4 日 (4 days)
  - 5 日 (5 days)

リストの更新時間は、メッセージ数およびフィルター基準によって異なります。基準の幅が狭いフィルターほど、リストはより迅速に更新されます。すべての重大度およびすべての期間でのストリングの検索は、処理に時間がかかることに注意してください。

メッセージごとに、以下の 3 つのフィールドが表示されます。

- タイム・スタンプ
- ユーザー
- メッセージ

「管理」 > 「システム」と選択すると、ログのサイズを制御するための以下の設定を使用できます。

- AutoClean エラー・ログ日数
- AutoClean 情報ログ日数
- AutoClean 警告ログ日数

詳しくは、262 ページの『システム構成の設定』を参照してください。

## 翻訳メッセージ

英語以外のメッセージは動的に翻訳されます。

en\_US 以外の言語で Build Forge を使用する場合、メッセージは動的に翻訳されます。「管理」 > 「メッセージ」を選択して、メッセージのリストを表示します。メッセージは、15 秒ごとに 250 個ずつ翻訳されます。

メッセージがまだ翻訳されていない場合、「メッセージ」フィールドに「未翻訳のメッセージ (Message Not Yet Translated)」と表示されます。

**重要:** 現在の設定と異なる言語を使用する新規ユーザーが追加された場合、メッセージにはそのユーザーの言語が使用されます。既存のメッセージは、以前の言語でも新規ユーザーの言語でも表示されません。

## ジョブ状況に関する RSS データ・フィードへの加入

RSS データ・フィードを使用して、個々のジョブの状況を追跡およびフィルター操作できるようになりました。Build Forge の、ジョブに関する RSS データ・フィードは、Build Forge 管理コンソールのサーバー状況に表示される情報と同じ情報を表示します。

ジョブ状況に関する RSS データ・フィードに加入するには、次のようにしてください。

1. Build Forge 管理コンソールで、「ジョブ」を選択します。

Web ブラウザーは RSS フィードを検出し、ブラウザーのアドレス・バーに「RSS」アイコンを表示します。

2. RSS 統合機能ツールで、Build Forge RSS データ・フィードをロードします。

例えば、URL をコピーして RSS データ・フィードのリストに追加したり、「RSS」アイコンをドラッグ・アンド・ドロップして RSS データ・フィードのリストに URL を追加します。

3. RSS データ・フィードに加入して URL を保存し、更新が行われたときに通知を受けるようにします。

注:

- URL のロードおよび RSS データ・フィードへの加入の詳細については、RSS 統合機能ツールの資料を参照してください。
- RSS データ・フィードを使用して、Build Forge のジョブ状況、システム・メッセージ、またはサーバー状況を英語以外の言語で表示するには、RSS 統合機能ツールが UTF-8 マルチバイト文字エンコードをサポートしている必要があります。
- RSS フィードは、タイム・スタンプを使用してアクティビティを報告します。タイム・スタンプは、ご使用の Build Forge ユーザー・アカウントに設定されているタイムゾーンに基づいています。ブラウザーが別のタイムゾーンに設定されている場合、正確なタイム・スタンプを得るためには、ご使用の Build Forge アカウントと同じタイムゾーンに設定する必要があります。



## システム・メッセージに関する RSS データ・フィードのフィルター操作

RSS データ・フィードで表示されるメッセージをフィルターに掛けることができます。そのためには、管理コンソールの「管理」 > 「メッセージ」ページでフィルター・プロパティを使用してください。

例えば、「重大度」プロパティを「監査」に変更したり、「最後」プロパティを 12 時間に変更してください。フィルター・プロパティを変更すると、「メッセージ」ページの RSS アイコンの URL が自動的に更新されます。

更新された URL を RSS 統合機能ツールにロードするには、「メッセージ」ページの「RSS」アイコンを右クリックしてリンクをコピーするか、更新された「RSS」アイコンをツール内の RSS データ・フィードのリストにドラッグ・アンド・ドロップしてください。

注:

- URL のロードおよび RSS データ・フィードへの加入の詳細については、RSS 統合機能ツールの資料を参照してください。
- RSS データ・フィードを使用して、Build Forge のシステム・メッセージまたはサーバー状況を英語以外の言語で表示するには、RSS 統合機能ツールが UTF-8 マルチバイト文字エンコードをサポートしている必要があります。
- RSS フィードは、タイム・スタンプを使用してアクティビティを報告します。タイム・スタンプは、ご使用の Build Forge ユーザー・アカウントに設定されているタイムゾーンに基づいています。ブラウザが別のタイムゾーンに設定されている場合、正確なタイム・スタンプを得るためには、ご使用の Build Forge アカウントと同じタイムゾーンに設定する必要があります。

---

## 「セキュリティ」パネル

「セキュリティ」パネルでは、次のセキュリティ・サービスを有効にすることができます。

- SSL: このパネルで SSL を有効にする作業は、システム全体で SSL セキュリティ・フィーチャーを有効にする作業の一部に過ぎません。これをオンにする前後に追加の作業が必要です。
- パスワード暗号化: このパネルでパスワード暗号化を有効にする作業は、システム全体でパスワード暗号化を有効にする作業の一部に過ぎません。

「セキュリティ」パネルを表示するには、「管理」 > 「セキュリティ」を選択します。

UI Config Console Reports Logout: Root User

Security Save Update Master BFClient.conf Export Key File New Key File Generate Key

Access: Security

SSL Enabled: No Password Encryption Enabled: No FIPS Enabled: No



**重要:** 「セキュリティー」パネルでは、セットアップの一部のみが実行されます。追加情報については、105 ページの『セキュリティー・フィーチャー』を参照してください。

## SSL の有効化

インストールの前提条件: インストール時に、SSL 構成で使用する次の 2 つの情報を指定してください。

- 「**SSL ポート (SSL port)**」。「Web サーバーおよびアプリケーション・サーバー (Web and Application Server)」パネルで指定します。このポートは、以下で選択する構成に指定されているポートと一致している必要があります。インストール時および構成におけるデフォルトは、ポート 8443 です。このポートは、Apache Tomcat 上の認証サブレットが、ログイン時にユーザーのログイン資格情報をエンコードまたは暗号化するために使用します。
- 「**証明書 (Certificate)**」。「Web サーバーおよびアプリケーション・サーバー (Web and Application Server)」パネルで指定します。独自の証明書を用意するか、インストーラーに自己署名証明書を作成させます。証明書はデフォルトの鍵ストアに保管されます。鍵ストアの場所は、名前付きの SSL 構成に定義されます。
- 「**SSL 有効化**」を「はい」に設定します。次の追加プロパティーが表示されます。
  - **LDAP アウトバウンド:** LDAP を使用したアウトバウンド通信で使用する構成を指定します。デフォルトは、「デフォルト JSSE アウトバウンド SSL」です。
  - **エンジンからエージェントへのデフォルトのアウトバウンド:** エンジン・コンポーネントからエージェントへの通信で使用する構成を指定します。デフォルトは、「デフォルト OpenSSL アウトバウンド SSL」です。
  - **サービス層インバウンド:** サービス層コンポーネントが Web インターフェース・コンポーネントおよびエンジン・コンポーネントからの通信を受け入れる際に使用する構成を指定します。デフォルトは、「デフォルト JSSE インバウンド SSL」です。
  - **サービス層アウトバウンド (JSSE):** エンジン・コンポーネントおよび Web インターフェース・コンポーネントのサービス層コンポーネントがデータベースと通信する際に使用する JSSE 構成を指定します。デフォルトは、「デフォルト JSSE アウトバウンド SSL」です。
  - **サービス層アウトバウンド (OpenSSL):** エンジン・コンポーネントおよび Web インターフェース・コンポーネントのサービス層コンポーネントがデータベースと通信する際に使用する OpenSSL 構成を指定します。デフォルトは、「デフォルト OpenSSL アウトバウンド SSL」です。
- 「**保存**」をクリックします。
- 「**マスター BFClient.conf の更新**」をクリックします。このステップを実行すると、これらのプロパティー設定を使用して BFClient.conf ファイルが編集されます。
- Build Forge を再始動します。システムがこれらの設定を使用し始めるまで、セキュア通信は有効になりません。

選択する構成は、「**SSL**」パネルで定義されます。

このパネルで SSL を有効にした後の要件:

- 証明書の配布: エージェント・ホストの鍵ストア、データベース・ホスト、および稼働中のすべての追加 Build Forge インストール済み環境 (冗長構成) に証明書をインストールする必要があります。
- エージェント SSL の有効化: エンジン・コンポーネントとエージェントの間の通信で SSL を使用する場合は、各エージェントが SSL を使用するように構成されている必要があります。
- API クライアントの有効化: サービス層コンポーネントと通信するために、すべての API クライアントで SSL を構成する必要があります。

## パスワード暗号化の有効化

前提条件:

- 「**SSL ポート (SSL port)**」。「Web サーバーおよびアプリケーション・サーバー (Web and Application Server)」パネルで指定します。このポートは、以下で選択する構成に指定されているポートと一致している必要があります。インストール時および構成におけるデフォルトは、ポート 8443 です。このポートは、Apache Tomcat 上の認証サブレットが、ログイン時にユーザーのログイン資格情報をエンコードまたは暗号化するために使用します。
- 「**証明書 (Certificate)**」。「Web サーバーおよびアプリケーション・サーバー (Web and Application Server)」パネルで指定します。独自の証明書を用意するか、インストーラーに自己署名証明書を作成させます。証明書はデフォルトの鍵ストアに保管されます。鍵ストアの場所は、名前付きの SSL 構成に定義されます。
- 「**パスワード暗号化の有効化**」を「はい」に設定します。
- 「**保存**」をクリックします。
- 「**マスター BFClient.conf の更新**」をクリックします。このステップを実行すると、これらのプロパティ設定を使用して BFClient.conf ファイルが編集されます。
- Build Forge を再始動します。システムがこれらの設定を使用し始めるまで、パスワード暗号化は有効になりません。
- これが有効になると、コンソールで入力されたすべての新規パスワードが暗号化されます。これには、コンソールで作成したユーザーのサーバー認証パスワードおよびユーザー・パスワードも含まれます。

追加要件:

暗号化を有効にした後で、以下を行う必要があります。

- すべてのエージェントで暗号化を有効にします。鍵をエクスポートし、その鍵を使用して、各エージェントの構成にあるサーバー認証パスワードを更新します。BFAgent.conf で、このパスワードを手動更新する必要があります。
- 暗号化されたパスワードをデータベース・アクセスに対して有効にします。鍵をエクスポートし、その鍵を使用して、Build Forge がデータベースにログオンする際に使用するデータベース・パスワードを更新します。buildforge.conf で、このパスワードを手動更新する必要があります。

## 「鍵ストア」パネル

「鍵ストア」パネルには、個々の鍵ストアの構成が含まれています。「管理」 > 「セキュリティ」 > 「SSL」パネルで SSL 構成を編集する際に、これらの個々の構成を、SSL 構成の一部として選択することができます。

構成を作成することも、用意された次の構成を使用することもできます。

- デフォルト JSSE 鍵ストア
- デフォルト JSSE トラストストア
- デフォルト OpenSSL CA ストア (認証局)
- デフォルト OpenSSL Cert ストア (証明書用)
- デフォルト OpenSSL 鍵ストア

各鍵ストアには、次のプロパティがあります。

### 名前

### アクセス

この鍵ストアを編集または削除できるユーザーを定義するアクセス・グループ。

**場所** 鍵ストア・ファイルの場所。すべてのデフォルト鍵ストアが、Build Forge の鍵ストアに関して、<bfinstall>/keystore というデフォルトの場所を使用します。ご使用のアプリケーション・サーバーが、提供された Tomcat アプリケーション・サーバーではなく WebSphere Application Server の場合は、絶対パスを指定します。

### 鍵ストア・タイプ

鍵ストアは、次のいずれかのタイプである必要があります。

- JKS
- PKCS12
- JCEKS
- PEM

### パスワード

鍵ストアへのアクセス時に使用する必要があるパスワードを指定します。

**確認** 確認のため、パスワードを再度ここに指定します。

## 鍵ストアおよび WebSphere Application Server

提供された Tomcat アプリケーション・サーバーではなく WebSphere Application Server を使用する場合は、鍵ストアの構成に以下の追加要件が適用されます。

- 「場所」フィールド: 相対パスではなく、絶対パスを指定する必要があります。
- 複数サービス・コンポーネント: 複数の Build Forge サービス・コンポーネントをインストールする場合、各コンポーネントは異なるホストにインストールされます。サービス・コンポーネントごとにセキュリティを構成します。「場所」フィールドに指定された鍵ストア・パスは、サービス・コンポーネントごとに同一でなければなりません。

## 「SSL」パネル

「SSL」パネルには、SSL の個々の構成が含まれています。「管理」 > **Security** パネルで「**SSL 有効化**」を「はい」に設定すると、これらの個々の構成を、SSL 有効化の一部として選択することができます。

独自の構成を作成することも、用意された次の構成を使用することもできます。

- デフォルト JSSE インバウンド SSL
- デフォルト JSSE アウトバウンド SSL
- デフォルト OpenSSL インバウンド SSL
- デフォルト OpenSSL アウトバウンド SSL

各構成には、次のプロパティがあります。

**名前** この構成の名前。

### アクセス

この鍵ストアを編集または削除できるユーザーを定義するアクセス・グループ。

**タイプ** JSSE または OpenSSL を選択します。

### クライアント認証

次のいずれかを選択します。

- 行わない
- サポート
- 必須

### サーバー認証別名

サーバー証明書の別名を入力します。

### クライアント認証別名

クライアント証明書の別名を入力します。

### 鍵ストア構成

鍵ストア構成の 1 つを選択します。これらは「鍵ストア」パネルで構成されます。

### トラストストア構成

トラストストア構成の 1 つを選択します。これらは「鍵ストア」パネルで構成されます。

### ハンドシェーク・プロトコル

次のいずれかを選択します。

- SSLv2
- SSLv3
- SSL
- TLSv1
- TLS
- SSL\_TLS - デフォルト OpenSSL インバウンド SSL およびデフォルト OpenSSL アウトバウンド SSL では使用不可
- TLSv1\_1

- TLSv1\_2

注:

以下の情報は、ハンドシェーク・プロトコルの TLSv1\_1 プロパティおよび TLSv1\_2 プロパティに適用されます。

- TLSv1.1 または TLSv1.2 を有効にするには、すべての構成 (デフォルト JSSE インバウンド SSL、デフォルト JSSE アウトバウンド SSL、デフォルト OpenSSL インバウンド SSL、デフォルト OpenSSL アウトバウンド SSL) を選択します。
- LDAP が有効であり、LDAP アウトバウンドで「デフォルト JSSE アウトバウンド SSL」が使用されている場合は、プロトコルを TLSv1\_1 または TLSv1\_2 に変更する前に、「デフォルト JSSE アウトバウンド SSL」をコピーします。LDAP はコピーされた構成を使用します。
- TLSv1\_1 および TLSv1\_2 は、BFagent と Java エンジン間、Java API とサービス層間、および Perl API とサービス層間でのみサポートされます。そのため、TLSv1\_1 または TLSv1\_2 が有効である場合、サーバーで TLSv1\_1 または TLSv1\_2 が有効になっていると、Perl エンジンはサーバーに接続できません。
- TLSv1\_1 および TLSv1\_2 は JDK 1.7 でのみサポートされます。そのため、バージョン 1.7 未満の JDK を引き続き使用している WebSphere Application Server 上に Build Forge .war ファイルをデプロイした場合、TLSv1.1 および TLSv1.2 はサポートされません。
- BFagent には、TLSv1.1 および TLSv1.2 サポートが Rational® Build Forge® 8.0.0.1 で追加されました。そのため、SSL ページで TLSv1\_1 または TLSv1\_2 を選択し、サーバーの「SSL 有効化」プロパティを「はい」に設定した場合は、エージェントのバージョンが 8.0.0.1 以上であることを確認してください。

#### 暗号スイート・グループ

「高」、「中」、「低」、または「すべて」のいずれかです。上位の暗号ほど安全性は高くなりますが、パフォーマンスの低下を伴います。

#### 暗号指定変更リスト

## 「SSO」パネル

「SSO」パネルには、シングル・サインオン (SSO) の構成が含まれています。新規の構成を作成することも、用意された次の構成を使用することもできます。

- フォーム SSO インターセプター
- SPNEGO SSO インターセプター

シングル・サインオン・サービスを有効にするには、追加のセットアップが必要です。それをすべてこのパネルから行うわけではありません。105 ページの『シングル・サインオンの実装』を参照してください。

各構成には、次のプロパティがあります。

名前 SSO 構成の名前。

### アクティブ

「はい」を選択すると、このインターセプターがアクティブになります。

### アクセス

この構成を編集または削除できるユーザーを定義するアクセス・グループ。

### Java クラス

この SSO 構成を実装する Java クラス。2 つ用意されています。

- フォーム・インターセプター:

`com.buildforge.services.server.sso.form.FORMSSOInterceptor`

- SPNEGO インターセプター:

`com.buildforge.services.server.sso.spnego.SPNEGOSOInterceptor`

---

## ライセンスの管理

ユーザーを作成すると、システムは、そのユーザーに関するデータをデータベースに保管します。ユーザーがシステムにログインした後で、システムはそのユーザーにライセンスを割り当てます。

### 許可ユーザー

ライセンスがユーザーに割り当てられています。ユーザーは常にログインできます。ログインにはライセンスが必要です。

### フローティング・ユーザー

フローティング・ライセンス・ユーザーは、ライセンスが有効である場合のみ、ログインできます。すべてのライセンスが使用されている場合、ログインできない可能性があります。

### トークン・ライセンス

複数の Rational 製品のライセンスを保有しているお客様であれば、トークン・ライセンスを使用するよう選択することができます。それらの製品にわたって使用できるトークン・ライセンスのプールを購入します。トークン・ライセンスは、ログインと同様、特定のフィーチャーを使用するときにも必要になる場合があります。

トークン・ライセンス・サポートは、フローティング・サーバー、フローティング・ユーザー、アダプター、および QuickReport の各ライセンスに対して利用可能です。このサポートは、固定ライセンス (許可サーバーや許可ユーザー) に対しては利用できません。サーバー、アダプター、および QuickReport の各ライセンスはチェックアウトされると、Build Forge がシャットダウンされるまでリリースされません。

トークン・ライセンス・サポートは、Rational Build Forge バージョン 7.1.3.4 以降に対応しています。トークン・ライセンス・サポートには、Rational Common Licensing バージョン 8.1.2 以降も必要です。

トークン・ライセンスについて詳しくは、IBM 営業担当員にお問い合わせください。

所有しているライセンスの数を超える数のユーザーを作成した場合、追加ライセンスを購入するか、root ユーザー・アカウントを使用して一部のユーザーからライセンスを除去 (あるいは、システムから一部のユーザーを削除) するまで、新しいユーザーはログインすることができません。



ユーザーは 1 回だけログインできます。あるコンピューターにログインしている場合に、同じ名前を使用して別のコンピューターからログインすると、元のセッションは無効になります。

注: バージョン 3.8 以前にシステムを購入した場合、お客様のライセンス体系はここで説明するものと異なる可能性があります。ライセンス交付に関する質問については、お客様サポートにお問い合わせください。

## 新規ライセンス・キーの入力

ライセンス・キーを変更するには、「管理」 > 「システム」を選択し、設定のリストでライセンス・サーバーの設定を見つけます。（「フィルター」ボックスに「License」と入力すると、リストでページ送りをすることなく、この設定が即時に表示されます。）

リストで「ライセンス・サーバー」項目をクリックすると、システムによって内容パネルの下部にタブが表示され、編集可能な「ライセンス・サーバー」フィールドも一緒に表示されます。フィールドを編集したら、「保存」ボタンをクリックします。

## トークン・ライセンス・ログ

Build Forge インストール・ディレクトリーには Rational Common Licensing 構成ファイル RCL\_Log\_Config.txt が入っています。デフォルトでは、インストール・ディレクトリーは以下のロケーションにあります。

- Windows システム  
C:\Program Files\IBM\Build Forge
- UNIX システムおよび Linux システム  
opt/buildforge/platform

デフォルトで、Rational Common Licensing ログ・ファイルはインストール・ディレクトリーの下に temp/rc1 サブディレクトリーに作成されます。

構成ファイルとログ・ファイルの両方を同じロケーションに保管するには、以下の手順に従います。

1. Build Forge を停止します。
2. ファイル RCL\_Log\_Config.txt で、Location = ./temp/rc1 の行を Location = ./token\_libs/rc1 に変更します。
3. Build Forge を再始動します。これで、すべてのトークン・ログが <bfinstall>/token\_libs/rc1 に作成されるようになりました。

---

## Perl エンジンの管理

このセクションでは、Perl エンジンを一時停止、開始、および停止する方法を説明します。

### Perl エンジンの一時停止

新規ジョブの処理を一時的に延期する場合は、Perl エンジンを一時停止します。



注: 例えば、データベースのバックアップやリストアのために、データベースを直接操作する場合は、まず、実行中の Build Forge サービスとプロセスをすべて停止する必要があります。この場合、エンジンを停止するだけでは不十分であり、不整合または破損 (あるいはその両方) を生じる可能性があります。

1. エンジンを一時停止するには、「管理」 > 「システム」をクリックし、「**Build Forge エンジンの一時停止**」プロパティを見つけます (「フィルター」ボックスを使用してください)。システムにより、「**Build Forge エンジンの一時停止**」設定を編集するためのパネルが表示されます。
2. プロパティを **Y** に変更し、「**構成の保存 (Save Config)**」をクリックします。エンジンは一時停止しますが、現在実行中のプロジェクトは続行されます。データがなにもデータベースに書き込まれていないことを確認するためには、アクティブになっているジョブが完了するまで待つ必要があります。

エンジンを再びアクティブにするには、プロパティの値を **N** に戻して、「**構成の保存 (Save Config)**」を再びクリックします。

## Perl エンジンの始動と停止

以下のセクションでは、Windows<sup>®</sup>、Linux<sup>®</sup>、および Solaris<sup>®</sup> 上で Perl エンジンを始動および停止する方法を説明します。

### Windows 上での Perl エンジンの始動と停止

Perl エンジンを開始および停止するには、Rational<sup>®</sup> Build Forge<sup>®</sup> ユーザー・インターフェースを使用します。Perl エンジンの開始および停止は、必ず手動で実行します。

#### Perl エンジンの開始

1. 次のようにして、Build Forge<sup>®</sup> ユーザー・インターフェースで Perl エンジンを使用可能にします。
  - a. 「管理」 > 「システム」をクリックします。
  - b. 「デフォルトを Java マスター・ジョブ・コントローラーに設定」フィールドで、「いいえ」を選択します。

注: Java エンジンのジョブが実行中である場合は、「デフォルトを Java マスター・ジョブ・コントローラーに設定」フィールドの値を「いいえ」に変更することはできません。これに対して、Perl エンジンのジョブにも同様のことが当てはまります。

2. コマンド行から Build Forge<sup>®</sup> インストール・パスに進み、コマンド `bfengine -p` を入力して Perl エンジンを開始します。

#### Perl エンジンの停止

Perl エンジンを停止するには、エンジンが実行中のウィンドウを手動で閉じます。

### Linux および Solaris での Perl エンジンの開始および停止

Perl エンジンを開始および停止するには、Rational<sup>®</sup> Build Forge<sup>®</sup> ユーザー・インターフェースを使用します。Perl エンジンの開始および停止は、必ず手動で行う必要があります。

### Perl エンジンの開始

1. 次のようにして、Build Forge<sup>®</sup> ユーザー・インターフェースで Perl エンジンを使用可能にします。
  - a. 「管理」 > 「システム」をクリックします。
  - b. 「デフォルトを Java マスター・ジョブ・コントローラーに設定」フィールドで、「いいえ」を選択します。

注: Java エンジンのジョブが実行中である場合は、「デフォルトを Java マスター・ジョブ・コントローラーに設定」フィールドの値を「いいえ」に変更することはできません。これに対して、Perl エンジンのジョブにも同様のことが当てはまります。

2. シェル・コマンド行から Build Forge<sup>®</sup> インストール・パスに進み、コマンド `rc/perl-engine start` を入力して Perl エンジンを開始します。

### Perl エンジンの停止

1. シェル・コマンド行から Build Forge<sup>®</sup> インストール・パスに進み、コマンド `rc/perl-engine stop` を入力して Perl エンジンを停止します。
2. Build Forge<sup>®</sup> ユーザー・インターフェースで、以下の手順を実行します。
  - a. 「管理」 > 「システム」をクリックします。
  - b. 「デフォルトを Java マスター・ジョブ・コントローラーに設定」フィールドで、「はい」を選択します。

---

## データベースの管理

このトピックでは、管理コンソール・データベースの設定時、特にデフォルト構成を変更したい場合に重要な問題について説明します。

### データベース・ログ・ファイルの削除

データベース・ログ・ファイルは定期的に削除してください。

システムは、管理コンソール・インストール・ディレクトリー内の `db.log` ファイルにデータベース・デバッグ情報を保管します。このファイルのサイズを毎月調べ、コンソール・コンピューターのスペースを解放する必要がある場合にはファイルを削除してください。

---

## エラー・メッセージ

このトピックでは、管理コンソールの使用中に検出される可能性のあるエラー・メッセージについて説明します。

### アクティブ・ステップが存在しません

このメッセージがスケジュール項目の「次の実行」列に表示された場合、関連するプロジェクト内のすべてのステップが使用不可になっています。プロジェクトに関するステップのリストを表示し、使用不可になっている各ステップのいくつかについて、その隣の赤い円をクリックしてステップを使用可能にしてください。

## ライセンス・キーが無効であるか、または Build Forge ライセンス・キーが破損または欠落しています。

ライセンス・キーの有効期限が切れているか、またはインストールされている製品バージョンでは無効です。新しいライセンス・キーを入力してください。288 ページの『新規ライセンス・キーの入力』を参照してください。

## データベース・ライセンスが必要です

LicMgr: 5140: データベース・ライセンスが必要です。  
(LicMgr: 5140: A database license is required.)

コンソール出力に上記のメッセージが表示される (フォアグラウンドでコンソールを実行する場合や、Linux<sup>®</sup> システムまたは UNIX<sup>®</sup> システムでコンソール・ログを表示する場合に表示されることがあります) のは、Enterprise ライセンスを取得せずに拡張データベース機能を使用しようとした場合です。例えば、DB2<sup>®</sup> 以外のデータベースを持つシステムを使用しようとしている可能性があります。ライセンスをアップグレードする場合は、サポートにお問い合わせください。



## 第 17 章 サーバー

このトピックでは、管理コンソールでのサーバー・リソースのセットアップと管理の方法について説明します。

### サーバー・リソースについて

コンソールのサーバー・リソースは、プロジェクトまたはステップを実行できるホストを表します。

- サーバーにはマニフェストがあります。マニフェストは、サーバー・プロパティのリストです。マニフェストの値は、コレクターの実行時に読み込まれます。サーバーに、割り当てられたコレクターがない場合、そのサーバー・マニフェストにあるいくつかのプロパティの値が自動的に設定されます。
- マニフェストは、コレクターによって読み込まれます。コレクターはサーバーに割り当てられます。コレクターは、プロパティ値を設定するほかに、プロパティの値をサーバーのエージェントから収集します。
- セレクターは、マニフェストからサーバー・プロパティを読み取ります。プロジェクトは、セレクターを使用して、ステップを実行するサーバーを決定できます。

管理者として、以下の手順でシステムをセットアップします。最初にサーバーを作成する必要があります。サーバーに割り当てられているコレクターを作成します。コレクターを実行して、サーバー・マニフェストを読み込みます。これらの作業の完了後、ビルド・エンジニアが、プロジェクト・ステップの実行場所を決定するセレクターを使用してプロジェクトを作成します。

各サーバー・リソースは、エージェントがインストールされたホストを示します。サーバー・リソースの追加時に、管理コンソールによる特定ホストのアクセス方法および使用方法を記述します。

サーバーの作成前に、そのサーバーに依存するデータ・オブジェクトが既に存在していることを確認してください。以下の項目をサーバーに割り当ててください。

項目	必須/オプション	説明
サーバー認証	必須	サーバーで使用するログイン名およびパスワードを指定します。
コレクター	オプション	デフォルトのプロパティ以外に、システムがサーバーから収集するプロパティを定義します。
環境変数	オプション	サーバーでプロジェクトが実行されるときに必ず割り当てられる環境変数が指定されます。

注: 単一の物理サーバーに対し、管理コンソール内に複数のサーバー・オブジェクトを作成できます。これらのサーバー・オブジェクトは、「論理サーバー」と呼ばれます。通常、論理サーバーを使用すると、プロジェクトは異なるプロパティを持つ同じハードウェアにアクセスできるようになります。例えば、2 つの論理サーバーは、異なるバスまたは異なる環境変数を使用する場合があります。

- 異なるバスを持つ 2 つの論理サーバーがあると、同じサーバー上に別々の作業ディレクトリーが作成されます。一方のサーバーを使用して実行した作業と、他方のサーバーを使用して実行した作業は区別できます。出力がすべて、別々のディレクトリーに保存されます。
- 異なる環境変数を持つ 2 つの論理サーバーでは、異なる開始環境変数値を使用してステップが実行されます。

## 「サーバー」パネルについて

「サーバー」パネルを使用して、プロジェクトを実行するサーバーを指定します。

管理コンソールでサーバーを管理するには、左側のメニューで「サーバー」をクリックします。

### 「詳細」タブ

「詳細」タブには、以下のフィールドがあります。

**ホスト** Rational Build Forge エージェントを実行しているホストの名前。

**コレクター**

サーバーで使用するコレクター。

**環境変数**

サーバーで使用する環境変数。

**サーバー・プロバイダー**

エンドポイント・サーバーとの通信を円滑に行うためのプロバイダー。デフォルトのサーバー・プロバイダーは、エージェント・サーバー・プロバイダーです。このプロバイダーによって、Build Forge エージェントのエンドポイント・サーバーとの通信が可能になります。

**認証** サーバー認証のログイン とパスワード。

**パスワード暗号化**

エージェントのパスワードを暗号化するかどうかを決定します。

**ファイル**

**.get** コマンドおよび **.put** コマンドで使用するファイル転送のタイプ。

**アクセス**

サーバー上のユーザーのアクセス・グループ。

**SSL 有効化**

Rational Build Forge システムが SSL を使用するように構成されているかどうかを示します。

**最大ジョブ数**

同時に実行されるジョブの最大数。

## 「マニフェスト」タブ

「マニフェスト」タブには、さまざまなサーバー・プロパティが表示されます。これにより、プロパティのデフォルト・セットが提供されます。コレクターを使用して、追加プロパティを含めることができます。

## 「テスト結果」タブ

「テスト結果」タブは、管理コンソールとサーバー間の最後の接続テストの状況を示します。

## 「メモ」タブ

「メモ」タブは、サーバーに関するコメントを保管および共有するための便利な場所を提供します。

---

## サーバーの作成

サーバー・リソースは、プロジェクトおよびステップを実行できるホストを表します。

サーバーを作成するには、次の手順に従ってください。

1. 左メニューで、「サーバー」を選択します。メイン・コンテンツ・パネルの下部に、「新規サーバー」パネルが表示されます。既存サーバーが選択されている場合は、「サーバーの追加」をクリックして、新規サーバーを追加できるようにパネルを消去してください。
2. サーバーの詳細を指定します。
  - 「名前」：サーバーの名前を入力します。この名前は、サーバーの `BF_NAME` プロパティになります。名前でサーバーを指定するには、セクター内でこの名前を参照します。
  - パス：サーバーがプロジェクトまたはジョブのディレクトリを作成するとき使用するディレクトリを指定します (例: `c:/buildforgeprojects`)。このパスの値は、ビルド・ディレクトリが作成される際の開始点として使用されます。

**ヒント：**システムはサーバー・パスを作成しません。パスは、ビルドがサーバーへのアクセスを試みる前に存在する必要があります。パスが存在しないと、ビルドは失敗します。

- **ホスト：**エージェントを実行している物理コンピューターのホスト名を入力します。管理コンソールのコンピューターをサーバーとして定義する場合は、値 `localhost` を使用します。また、エージェントも管理コンソールにインストール済みでなければなりません。

**注：**ホスト名と一緒にポート番号を使用することができます (例: `host_name:port_number`)。ホスト名と一緒にポート番号が指定されていると、「デフォルト・エージェント・ポート」システム設定で定義されているポート番号は、この番号でオーバーライドされます。「管理」 > 「システム」 > 「デフォルト・エージェント・ポート」とクリックします。

**注：**ホスト名の前にプロトコルを付けしないでください。例えば、`http://` を使用しないでください。



- **認証:** このサーバーで使用するサーバー認証を選択します。
- **アクセス:** このサーバーを使用できるユーザーのアクセス・グループを選択します。
- **コレクター:** このサーバーで使用するコレクターを選択します。
- **環境変数:** このサーバーがプロジェクトの実行に使用されるときに適用される、一群の環境変数を選択します。これらの変数は、他のすべての変数より先に適用されます。また、これらの変数は、サーバーに固有なパラメーターを設定します。
- **サーバー・プロバイダー:** エンドポイント・サーバーとの通信を円滑にするためのプロバイダーを選択します。デフォルトのサーバー・プロバイダーは、エージェント・サーバー・プロバイダーです。このプロバイダーによって、Build Forge エージェントのエンドポイント・サーバーとの通信が可能になります。
- **ファイル:** **.get** および **.put** コマンドを介して使用できる、このサーバーのファイル転送のタイプを定義します。転送なし (なし)、ファイルの読み取り (.get)、ファイルの書き込み (.put)、またはその両方 (.get および .put) を選択できます。
- **最大ジョブ数:** 同時に実行できるジョブの最大数を入力します。デフォルト値は 3 です。
- **パスワード暗号化構成:** エージェントのパスワードを暗号化する場合、「有効」を選択します。デフォルトは「無効」です。
- **SSL 有効化:** SSL を使用するように Build Forge<sup>®</sup> システムを構成した場合、およびこのサーバー・リソースを SSL を介してエージェントと通信させる場合は、「はい」を選択します。デフォルトは「いいえ」です。

注: 「はい」を選択した場合でも、Build Forge<sup>®</sup> が SSL 用に正しく構成されていないと、このサーバー・リソースは、エージェントと通信することができません。

3. 「保存」をクリックします。新規サーバーはコンテンツ・パネルの上部のサーバー・リストに表示されます。サーバーが正しく構成されていることを確認するには、リストからサーバーを選択し、「接続のテスト (Test Connection)」をクリックします。サーバーと通信できない場合は、エラーがレポートされます。

---

## サーバーのテスト

サーバーに対して、一連の診断テストを実行できます。

診断テストを実行するには、以下の手順を実行します。

1. 「サーバー」->「server\_name」をクリックします。
2. 「接続のテスト」をクリックします。

サーバーの接続テストが実行され、サーバー・マニフェストが更新されます。リフレッシュが完了するまで、「テスト結果」タブに進行状況表示バーが表示されます。

「テストの詳細の表示」をクリックすれば、詳細な結果を「テスト結果」タブに表示できます。

サーバーがテストに失敗した場合は、以下のいずれかのアクションを試行してください。

- サーバー認証でのユーザー名とパスワードが正しいことを確認する。
- 正しいホスト名を使用しているか検査する。
- サーバーにエージェントを再インストールするか、エージェントがインストールされていることを検査する。

---

## サーバーの変更

必要に応じて、サーバーのプロパティを編集することができます。

**重要:** 以下の条件に該当する場合は、コンソールを再始動する必要があります。

- ホスト・パラメーターでエージェント・ホストのホスト名を使用している。
- エージェント・ホストの IP アドレスが変更されている。

DNS から取得した IP アドレスは、Build Forge サービス・コンポーネントをホストする JVM にキャッシュされます。キャッシュを消去して、サービス・コンポーネントが新しい IP アドレスにあるエージェントを検出できるように再始動する必要があります。

ホスト・パラメーターで IP アドレスを使用する場合は、再始動する必要はありません。エージェント・ホストの IP アドレスが変更されている場合は、ホスト・パラメーターを新しいアドレスに変更して保存してください。

---

## サーバーの有効化および無効化

サーバーを一時的に無効にすることができます。無効の場合は、ジョブを実行することはできません。

サーバーを無効または有効にするには、次の手順に従ってください。

1. 「サーバー」 をクリックして、サーバーのリストを表示します。
2. 変更するサーバー・リソースの名前をクリックします。そのサーバーの詳細が表示されます。
3. メイン・パネルの下部にある「詳細」タブで、「無効」を選択します。
4. 「サーバーの保管」をクリックします。

---

## サーバーでの同時ジョブの制限

「最大ジョブ数」サーバー・プロパティを使用し、特定のサーバーが同時に実行できるジョブの最大数を指定します。

### このタスクについて

システムは、いずれかの単一サーバーで実行を試みるプロセスの数を制限します。

「最大ジョブ数」プロパティでは、プロセス数の上限を指定します。このプロパティのデフォルト値は 3 です。

注: 他のプログラムをこのサーバーで実行することができます。システムの制限値は、Build Forge ジョブのみに適用されます。

注: ビルトイン変数 `BF_RESERVE` および `BF_EXCLUSIVE` は、サーバーのジョブ・スロットの 1 つまたはすべての予約を制御します。詳しくは、317 ページの『事前設定されたプロパティ』を参照してください。

注: Linux または UNIX で `bfagent` を使用する場合は、`xinetd.conf` を構成してください。これは、同時に実行中のジョブによって多数のソケット接続が生成されるためです。「ERROR: Exception communicating with endpoint」というメッセージを受け取った場合は、インスタンスおよび `per_source` 値を増加してください。

サーバーにデフォルト以外の値を適用する場合には、次の手順を実行してください。

### 手順

1. 「サーバー」をクリックします。
2. 既存のサーバーを選択するか、新規サーバーを作成します。
3. 「最大ジョブ数」プロパティを希望の値に設定します。
4. 「保存」をクリックします。

---

## ジョブ数のリセット

### サーバーでのジョブ数のゼロへのリセット

「サーバー」ページの「ジョブ数のリセット」を使用すると、選択したサーバーのジョブ数 (`BF_JOBS`) がゼロにリセットされます。`BF_JOBS` はサーバーで現在実行中のステップまたはジョブの数です。この数値の設定が低すぎる場合、ジョブはジョブ競合によって失敗する可能性があります。

これを選択すると、ジョブの完了、失敗、または取り消し時に `BF_JOBS` が適切にリセットされない場合に、`BF_JOBS` をリセットすることができます。

例えば、複数のジョブを取り消すと、`BF_JOBS` がリセットされない場合があります。`BF_JOBS` がリセットされないと、最大ジョブ数の設定値に達してしまい、ステップまたはジョブが実行されなくなります。

1. 「サーバー」を選択して、サーバーのリストを表示します。
2. サーバーを選択します。
3. 「ジョブ数のリセット」をクリックします。

### すべてのサーバーでのジョブ数のゼロへのリセット

「サーバーのジョブ数のリセット」ページを使用して、すべてのサーバーのジョブ数 (`BF_JOBS`) を同時にゼロにリセットできます。`BF_JOBS` プロパティは、サーバーで実行中のステップまたはジョブの数です。この数値の設定が低すぎる場合、ジョブはジョブ競合によって失敗する可能性があります。

すべてのサーバーの `BF_JOBS` 値がリセットされた後、「サーバーのジョブ数のリセット」の値が「いいえ」(デフォルトの設定値)に戻ります。

例えば、複数のジョブをキャンセルすると、ビルド・システムが BF\_JOBS 値のリセットに失敗する場合があります。BF\_JOBS 値がリセットされないと、デフォルトの最大ジョブ数のシステム設定値に達してしまい、ステップまたはジョブが実行されなくなります。

1. 「管理」 > 「システム」をクリックして、システム構成設定のリストを表示します。
2. 「サーバーのジョブ数のリセット」を見つけます。
3. 「サーバーのジョブ数のリセット」をクリックします。
4. 「詳細」タブをクリックし、値として「はい」をクリックします。
5. 「保存」をクリックします。

---

## サーバー状況に関する RSS データ・フィード

管理コンソールは、サーバーが機能テストに合格できること、およびエージェントがログインできることを確認するために、サーバー状況検査を実行します。「テスト結果」タブは、状況検査の結果を表示します。サーバーが作成または編集されるたびに、管理コンソールによってサーバー状況が自動的に検査されます。また、ユーザーは、サーバー状況の検査をいつでも（例えば、プロジェクトを実行する前に）開始できます。

Build Forge の、サーバー状況に関する RSS データ・フィードは、Build Forge 管理コンソールの「テスト結果」タブに表示される情報と同じ情報を表示します。

サーバー状況に関する RSS データ・フィードに加入するには、次のようにしてください。

1. Build Forge 管理コンソールで、「サーバー」を選択します。

Web ブラウザーは RSS フィードを検出し、ブラウザーのアドレス・バーに「RSS」アイコンを表示します。

2. RSS 統合機能ツールで、Build Forge RSS データ・フィードをロードします。

例えば、URL をコピーして RSS データ・フィードのリストに追加したり、「RSS」アイコンをドラッグ・アンド・ドロップして RSS データ・フィードのリストに URL を追加します。

3. RSS データ・フィードに加入して URL を保存し、更新が行われたときに通知を受けけるようにします。

注:

- URL のロードおよび RSS データ・フィードへの加入の詳細については、RSS 統合機能ツールの資料を参照してください。
- RSS データ・フィードを使用して、Build Forge のシステム・メッセージまたはサーバー状況を英語以外の言語で表示するには、RSS 統合機能ツールが UTF-8 マルチバイト文字エンコードをサポートしている必要があります。
- RSS フィードは、タイム・スタンプを使用してアクティビティを報告します。タイム・スタンプは、ご使用の Build Forge ユーザー・アカウントに設定されているタイムゾーンに基づいています。ブラウザーが別のタイムゾーンに設定され

ている場合、正確なタイム・スタンプを得るためには、ご使用の Build Forge アカウントと同じタイムゾーンに設定する必要があります。

## サーバー認証

このトピックでは、サーバーで実行するプロジェクトへのアクセス権限を付与する方法について説明します。

### サーバー認証について

サーバー認証を使用して、ログイン資格情報をサーバーに関連付けます。

サーバー認証には、ログイン名とパスワードが単一の名前付きオブジェクトとして保管されます。ユーザーは、このオブジェクトを 1 つまたは複数のサーバーに関連付けることができます。サーバー認証を作成および編集するには、「サーバー認証」ページを使用してください。

一連のサーバー認証を管理することにより、多くのサーバーに同じクレデンシャルを使用して、クレデンシャルをグローバルに更新できます。

### 「サーバー認証」パネルについて

サーバー認証を作成および管理するには、「サーバー認証」パネルを使用してください。

パネルを表示するには、「サーバー」 > 「サーバー認証」を選択します。

The screenshot displays the 'Server Auth' management interface. At the top, there are tabs for 'UI Config', 'Console', 'Reports', and a 'Logout: Root User' link. The main section is titled 'Server Auth' and includes an 'Add Server Authentication' button. Below this is a table with two columns: 'Name' and 'User name'. The table contains two entries: 'jsmith' and 'jdoe'. To the right of the table, there are controls for 'Showing 1 - 2 of 2' and a 'Display All' link. Below the table, there are four buttons: '(New Server Authentication)', 'Save Server Authentication', 'Copy Server Authentication', and 'Delete Server Authentication'. The 'Details' section is expanded, showing a form with the following fields: 'Name' (text input), 'Access' (dropdown menu set to 'Build Engineer'), 'Login' (text input), 'Password' (text input), and 'Verified' (text input).

このパネルには、以下のフィールドがあります。

**名前** 認証自体に使用する名前。

**アクセス**

この認証を使用できるユーザーのアクセス・グループ。

**ログイン**

ログイン時に使用するアカウント名。

**パスワード**

「ログイン」に関連するパスワード。

**確認** パスワードの再入力。

## サーバー認証の作成

サーバー認証は、一連のサーバーに関するログイン情報を保管するために使用します。

### このタスクについて

管理コンソールが適切な特権でサーバーにログインできるように、各サーバーにサーバー認証を割り当てます。サーバー認証はサーバー・レコードからログイン情報を分離し、同じログイン情報を複数のサーバーに適用できるようにします。

サーバー認証を作成するには、次の手順に従ってください。

### 手順

1. Build Forge の左パネルで、「サーバー」 > 「サーバー認証」をクリックします。

ビルド・システムにより、メイン・コンテンツ・パネルの上部に既存のサーバー認証のリストが表示され、下部にブランクの「サーバー認証詳細 (Server Auth Details)」パネルが表示されます。

**ヒント:** サーバー認証を選択すると、「サーバー認証詳細 (Server Auth Details)」パネルに、選択されたサーバーの認証情報がシステムにより取り込まれます。新規認証を作成できるようにパネルをクリアするには、「サーバー認証の追加」をクリックします。

2. 「名前」に、認証名 (システム内のサーバー認証を識別するための論理名) を入力します。
3. 「ログイン」に、サーバー・ログイン名を入力します。

**注:** ログイン名がドメイン・ユーザー名である場合は、このフィールドにドメイン名も含めてください。例えば、MYDOMAIN/joeuser のように入力します。

4. 「パスワード」にパスワードを入力します。
5. 「パスワードの確認」にパスワードを再度入力します。
6. 「サーバー認証の保管」をクリックします。システムは、選択した名前で新規サーバー認証を保管します。

### タスクの結果

ビルド・システムは、選択した名前で新規サーバー認証を保管します。

## サーバー認証のオーバーライド

特別な環境変数を使用することにより、サーバーに割り当てられたサーバー認証ではなく、管理コンソールのログイン資格情報をサーバーに使用させることができます。割り当てられた認証をオーバーライドするには、変数 `_USE_BFCREDS` に値 `1` を指定して、プロジェクトまたはステップで使用される環境に追加してください。この変数をプロジェクト環境に追加すると、ビルド・システムはプロジェクトの各ステップでオーバーライドを行います。



`_USE_BFCREDS=1` が指定された環境では、ビルド・システムは、プロジェクトを開始したユーザーのコンソール・ログイン資格情報を使用して、ステップのコマンドを実行します。

**注:** LDAP/アクティブ・ディレクトリー認証を使用している場合、`_USE_BFCREDS` 変数を機能させるためには「**ユーザー認証のローカル保管**」システム設定を「はい」(デフォルト値) に設定する必要があります。この設定が「はい」の場合、システムは、ユーザー認証情報を暗号化された形式でキャッシュに入れます。これにより、`_USE_BFCREDS` で使用するためにユーザー認証情報にアクセスできるようになります。それ以外の場合、システムは LDAP 情報を保管せず、その情報を使用することができません。

**ヒント:** Windows の場合、変数 `_USE_BFCREDS_DOMAIN` の設定も検討してください。この変数の値を 1 に設定すると、ユーザーのドメインが組み込まれます。

## 制限されたサーバー認証の使用の許可

「アクセス不能なサーバー認証の実行」権限を使用すると、ユーザーが、自分がアクセス権を持っていないサーバー認証を使用して、サーバーでステップを実行できるようになります。

前提条件として、ユーザーにあらかじめサーバーへのアクセス権があるか、付与されている必要があります。(サーバーへのアクセス権を付与するには、「**サーバー**」>「**アクセス**」をクリックします。)

サーバー認証を使用して、サーバーにログインおよびアクセスします。サーバー認証はアクセス・グループと関連付けられます (「**サーバー**」>「**サーバー認証**」>「**アクセス**」)。

サーバーに関する以下のサーバー認証を作成することができます。

- `dev/dev` サーバー認証。 このサーバー認証をサーバーの開発者アクセス・グループと関連付けます。
- `qa/qa` サーバー認証。 このサーバー認証を QA アクセス・グループと関連付けます。
- `prod/prod` サーバー認証。 このサーバー認証を Build アクセス・グループと関連付けます。

`qa/qa` サーバー認証へのアクセス権限しか持たないユーザーに、`prod/prod` サーバー認証としてステップを実行することを許可するために、「アクセス不能なサーバー認証の実行」権限を QA アクセス・グループに追加します。

**注:** ユーザーがサーバーへのアクセス権を持っていて、「アクセス不能なサーバー認証の実行」権限を介したサーバー認証へのアクセス権を持っていない場合でも、ステップは実行されます。ただし、これは、環境変数 `_USE_BFCREDS` がサーバー認証をオーバーライドするように設定されている場合のみです。

---

## セクター

このトピックでは、セクターについて説明します。



## セレクトターについて

セレクトターは、プロジェクトまたはステップを実行するサーバー・リソースを選択します。

セレクトターには、変数のリストが含まれています。それぞれの変数について、値と比較演算子を指定します。例えば、プロパティ「`CompilerVersion = 1.1`」を指定して、そのプロパティを持つサーバー・リソースのみを選択することができます。また、「`CompilerVersion >= 1.1`」を指定して、バージョン 1.1、1.3、2、または 2.0 のサーバー・リソースを選択することもできます。

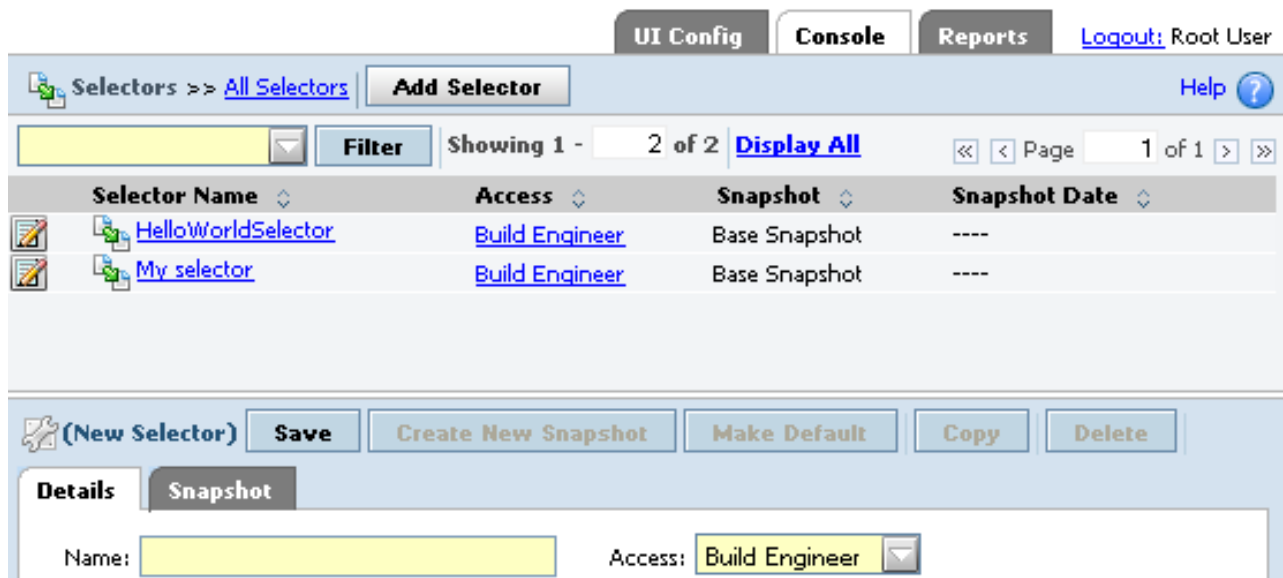
プロジェクトまたはステップの実行時、それらに割り当てられたセレクトターは、実行場所となるサーバー・リソースを判別します。

- 静的セレクトターは、BF\_NAME 変数を使用して、名前によってサーバー・リソースを識別します。
- 動的セレクトターは、セレクトターの変数により指定される基準を使用して、システム内のすべてのサーバー・リソースからサーバー・リソースを選択します。

### 「セレクトター」パネルについて

「セレクトター」パネルを使用して、プロジェクトまたはステップを実行するサーバー・リソースを選択するセレクトターを作成および管理します。

パネルを表示するには、「サーバー」 > 「セレクトター」を選択します。



「詳細」タブ: 「詳細」タブには、以下のフィールドがあります。

名前 セレクトターの名前。

アクセス

このセレクトターを使用できるユーザーのアクセス・グループ。

「スナップショット」タブ: セレクター・スナップショットは、セレクターのインスタンスです。「スナップショット」タブには、スナップショットの名前が表示されます。スナップショット名やスナップショットに関するコメントを表示または変更する場合は、このタブを使用します。

## セレクターのセットアップ方法

セレクターは、プロジェクトまたはステップに適したサーバーの種類を記述します。サーバーを名前、またはコレクターが収集してマニフェストに保管するプロパティで指定することができます。 必須およびオプションの複数のプロパティを指定することができます。

セレクターのセットアップには、以下のような複数のアプローチが可能です。

- サーバー・リソースを名前で選択します。 セレクターを作成し、サーバー・リソース名に倣って名前を付けます。セレクターは `BF_NAME` 値 (システムで 사용되는固有の名前) によってサーバー・リソースを指定します。プロジェクトまたはステップを実行するサーバー・リソースを指定するには、これらのセレクターの 1 つを使用します。
- サーバーをサーバー・プールで選択します。 指定されたプールにサーバーを編成し、各プールにコレクターを作成することができます。プール名をコレクターのプロパティ (設定値プロパティ) として定義します。その後で、プール名ごとにセレクターを作成します。プロジェクトまたはステップのサーバー・リソースが、現在の負荷に基づいて選択されます。
- サーバーをサーバー属性で選択します。 使用可能なハード・ディスク・スペース、オペレーティング・システム、CPU の個数など、機能に関するプロパティに基づいて、サーバーを選択できます。動的な選択を可能にするには、以下の手順を実行します。
  1. 適切なプロパティの収集と割り当てを行うコレクターを作成します。
  2. 次に、コレクターを適切なサーバーに割り当てます。
  3. 選択したサーバーを示すプロパティまたはプロパティ・セットごとに、セレクターを作成します。

例えば、以下の基準に従って選択を行うセレクターを作成することができます。

- 「Windows®」を含むオペレーティング・システムを使用したサーバー・リソース。
- 複数の CPU を搭載したサーバー・リソース。
- 指定した負荷より少ない負荷で実行されているサーバー・リソース。
- サーバー・リソースをネストしたコレクターで選択します。「組み込み」の「タイプ」プロパティを使用して、別のコレクターをポイントするコレクターを作成します。コレクター自体は、コレクター・ポインターのセットで構成される場合があります。サーバーごとに個別のコレクターを作成し、例えば、個々のサーバーにいくつかの固有のプロパティを持たせることができます。「組み込み」タイプを使用して、ユーティリティ・コレクターをポイントすることができます。例えば、Perl および Java などの環境で重要なリソースのバージョン番号を指定する、「バージョン」と呼ばれるコレクターを作成することができます。

## セレクター変数のタイプ

セレクターでは、実行時にプロジェクトまたはステップのサーバー・リソースを選択する方法を定義します。

次のタイプの変数をセレクターでの定義に使用できます。

- **標準プロパティ:** このタイプの変数には、以下を指定します。

- **名前:** 使用するプロパティの名前。すべてのビルトイン・プロパティ、およびコレクターで定義された任意の設定値プロパティで構成される自動生成リストから選択します。

**注:** 設定値プロパティからセレクターを作成するには、はじめに 1 つ以上のコレクターで設定値プロパティを定義する必要があります。

- **演算子:** 以下の比較演算子の 1 つを使用します。
    - EQ - Equal (等しい)。値が完全に一致する必要があります。この値には数値またはストリングを使用できます。
    - NE - Not Equals (等しくない)。値は指定した値と一致する必要はありません。この値には数値またはストリングを使用できます。
    - GT - Greater Than (より大きい)。
    - GE - Greater than or Equal (以上)。
    - LT - Less Than (より小さい)。
    - LE - Less than or Equal (以下)。
    - Contains (含む)。
  - **値:** 演算子で比較に使用する値。
  - **必須:** セレクターがこの変数と一致する必要がある場合は「はい」、オプションの場合は「いいえ」。
- **組み込み:** この「組み込み」タイプを指定すると、ネスティングを利用して単純なセレクターから作成した複雑なセレクターを定義できます。以下の項目を指定します。
    - **セレクター:** 組み込むセレクターを選択します。セレクターで指定したすべてのプロパティが組み込まれます。複数の単純なセレクターを組み込むことにより、複雑なセレクターを作成できます。

## セレクター変数の比較規則

システムはプロジェクトまたはステップに使用するサーバー・リソースを選択する際に、セレクター変数の値と、同じ名前のマニフェスト・プロパティの値を比較します。

システムはいずれか一方の値が以下の数値基準に一致していない場合に、ストリングの比較を行います。

- 値が数字で始まり、数字と、1 つ以上の数字が続く小数点のみで構成されている場合、システムは数値比較を実行します。
  - 5、5.5、0.5、5.0、および 5.5.5 は、数値と見なされます。
  - 5.、.5、5..、5..5、5.4.6\_05、および 5.6i5 は、すべてストリングと見なされます。

- 複数の小数点が含まれている数値の場合は、サブバージョンの数値比較が行われます。この場合、システムは小数点で区切られた各フィールドを比較します。通常の数値比較では 5.21 が 5.3 より小さい値と見なされますが、サブバージョンの数値比較では 5.21.0 が 5.3 より大きい値と見なされます。

注: Contains 演算子の場合、システムは常に大/小文字を区別せずに スtringの比較を行います。

下の表に、比較規則の適用例を示します。

プロパティ名	マニフェストのプロパティ値	演算子	セレクトー変数の値	比較タイプ	一致?
PerlVersion	v5.8.4	>=	5.2.1	String	はい
PerlVersion	v5.8.4	>=	v.5.2.1	String	はい
PerlVersion	v5.8.4	>=	v5.22.1	String	はい
OS_VERSION	1.15	>=	1.1	数値	はい
OS_VERSION	1.10	>=	1.1.0	サブバージョン数値	はい
BF_NAME	WinServer1	含む	win	String	はい
BF_NAME	Server123	=	123	String	いいえ

## セレクトーを用いた適格なサーバー・リソースの評価

プロジェクトまたはステップ用のサーバー・リソースを選択する場合、システムは、セレクトーを使用して、適格なすべてのサーバー・リソースを評価します。

1. システムは、セレクトー内のすべての必須 変数を含んだサーバーのリストをコンパイルします。必要なセレクトー基準と一致するサーバー・リソースがない場合、プロジェクトまたはステップは失敗し、システムはメモを作成します。
2. 必要な基準を満たすサーバー・リソースが複数存在する場合、システムはそれぞれの適格なサーバー・リソースを評価し、以下のようにポイントを割り当てます。
  - 基準を満たすオプション 変数ごとに 1 ポイントを割り当てます。セレクトーに同じ変数のコピーが複数含まれている場合、各コピーに 1 ポイントずつ割り当てます。
  - BF\_LOADRATIO の値が最も低いサーバー・リソースに 1 ポイントを割り当てます。
3. システムは、ポイント数が最大のサーバーを選択します。ポイント数が最大のサーバー・リソースが複数存在する場合、システムはそのうちのいずれかを選択します。

オプション変数はセレクトー内で反復させることができます。そうすることで、基準を満たすサーバーのスコアが増加します。例えば、必須のメモリーは 1 GB であるが、2 GB 以上のメモリーを強く推奨する場合は、セレクトー変数を以下のように設定します。

- 必須変数として MEM\_TOTAL GE 1GB を指定します。

- オプション変数として MEM\_TOTAL GE 2 GB を 3 回指定します。

## コレクター

### コレクターについて

コレクターは、どのプロパティを収集し、サーバー・リソースに割り当てるかを定義します。

「サーバー」パネルの「コレクター」セクションには、使用可能なコレクターがリストされ、新規コレクターを作成することができます。

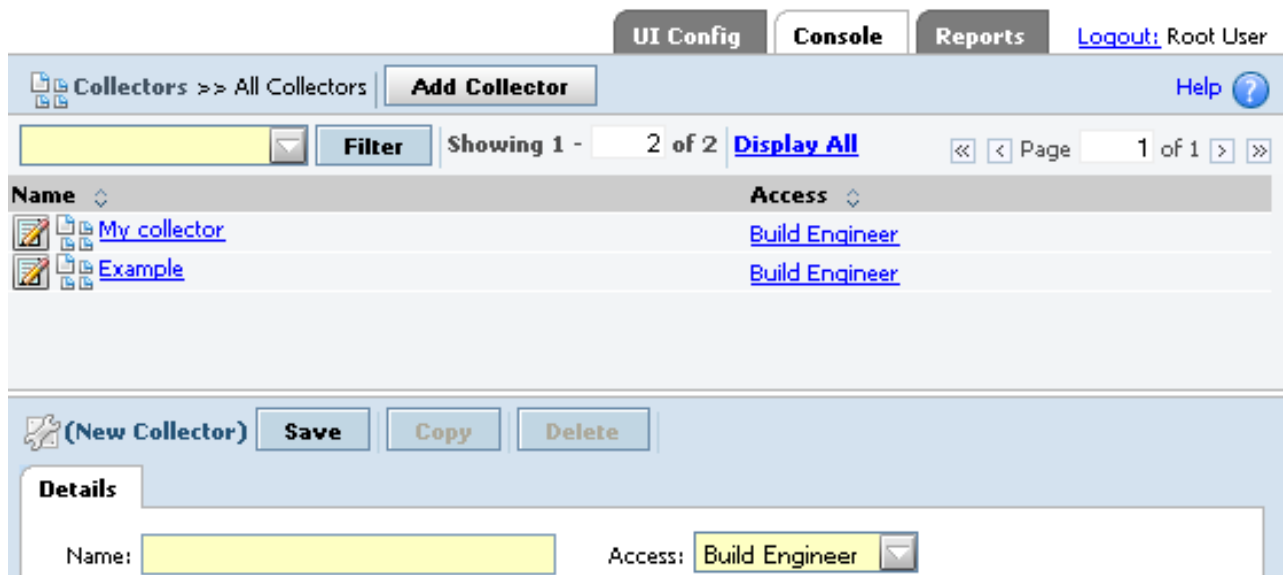
サーバー・パネルのコレクター・セクションを開くには、「サーバー」 > 「コレクター」をクリックします。

コレクターは、1 つ以上のプロパティから構成されます。各プロパティにより、マニフェストに組み込まれる情報が指定されます。

注: コレクターにより、ダイナミック・セクターの使用が可能になります。コレクターを使用するには、Enterprise Edition が必要です。

### 「コレクター」パネルについて

「コレクター」パネルを使用して、コレクターの作成、編集、および削除を行います。パネルを表示するには、「サーバー」 > 「コレクター」を選択します。



パネルには、以下のフィールドがあります。

**名前** コレクターの名前。

**アクセス**

このコレクターを使用できるユーザーのアクセス・グループ。

## コレクターの作成

コレクターを作成して、プロジェクト用のサーバー・リソースを動的に選択できるようにします。

### 手順

1. 「サーバー」 > 「コレクター」を選択します。
2. コレクターの名前を指定します。
3. コレクターに対するアクセス権限を指定します。
4. 「保存」をクリックします。 収集するデータを決定する変数を入力できるように、パネルが変わります。
5. 1 つ以上の変数を作成して保存します。 変数プロパティについて詳しくは、『コレクター変数のプロパティ・タイプ』を参照してください。

### コレクター変数のプロパティ・タイプ

コレクター変数のプロパティでは、サーバー・マニフェストの情報を収集する方法を指定します。

コレクターでは、以下のタイプのプロパティを定義できます。

**設定値** これらのプロパティは、指定された静的な値をサーバーに割り当てます。プロパティ名と値を指定します。

値に特殊値を使用して、事前定義された応答を取得できます。特殊値は、先頭に下線 ( \_ ) 文字が付いています。 319 ページの『コレクターの特殊なプロパティ』を参照してください。

### ビルトイン

これらのプロパティは、サーバー・リソースについての情報を戻します。ビルトイン・プロパティのリストについては、 313 ページの『ビルトイン・プロパティ』を参照してください。

### 実行コマンド

このタイプのプロパティは、システムが実行するコマンドを指定します。プロパティ値は、コマンドの実行により生成される出力に設定されます。デフォルトでは、出力の最初の 255 文字が使用されます。 正規表現を使用して、出力の特定の部分を抽出できます。

- **プロパティ:** プロパティの名前。
- **コマンド:** サーバー・リソースに割り当てられたホスト上で実行するコマンド。
- **正規表現:** オプション。出力をフィルターに掛けるために使用する正規表現。指定した場合、ビルド・システムは、コマンドの出力の各行 と正規表現との突き合わせを試行します。行が初めて一致した場合、システムは値 \$1 (Perl 規則) を検索し、 \$1 をそのプロパティの値として使用します。正規表現には、値を戻すための括弧が少なくとも一組含まれていなければなりません。 Perl 正規表現の構成について詳しくは、Perl の資料を参照してください。

### 組み込み

このタイプのプロパティは、コレクターのリストを指定します。コレクターは、ネストすることができます。「組み込み」タイプのコレクターを作成



する場合は、別のコレクターの名前をその値として指定します。ビルド・システムは、マニフェストの作成時または更新時に、参照したコレクターからプロパティを挿入します。

**ヒント:** システムは、コレクターにリストされる順序でコレクター変数プロパティを適用します。後で同じ名前のプロパティが古いプロパティをオーバーライドします。あるコレクターを別のコレクターに組み込む場合は、このフィーチャーを使用します。コレクターのすべての変数プロパティではなく、一部を使用する場合は、使用しない変数プロパティを無効にします。

システムは、サーバーの論理名が含まれている `BF_NAME` プロパティなどの、いくつかのプロパティも自動的に適用します。これらは、特殊なマニフェスト・プロパティの一部とみなされます。 317 ページの『事前設定されたプロパティ』を参照してください。

---

## マニフェストおよび動的サーバーの選択

コレクター、マニフェスト、およびセレクターを一緒に使用して、プロジェクトまたはステップの実行時にサーバー・リソースを選択します。

次の異なる 3 つのデータ・オブジェクトがあると、システムが動的にサーバーを選択することができます。

- コレクター は、システムがサーバー・リソースから収集したり、サーバー・リソースに割り当てるプロパティのセットを定義するオブジェクトです。システムは、サーバー・リソースのプロパティを検査した場合にコレクターを実行します。収集されたプロパティ値は、マニフェストに保管されます。
- マニフェスト は、特定のサーバーのプロパティのリストです。これにはコレクターの実行結果が含まれます。
- セレクター は、`MEM_TOTAL = 512` などのプロパティおよび比較のリストです。システムは、セレクターのプロパティをマニフェストと比較して、サーバーが特定のセレクターの要件を満たしているかどうかを確認します。プロジェクトおよびステップは、セレクターをプロパティの 1 つとして指定します。プロジェクトまたはステップが実行されると、セレクターは、実行場所となるサーバー・リソースを選択するために、定義されたすべてのサーバー・リソースのマニフェストと比較されます。

次の例は、単一のセレクターを作成して使用方法を示しています。

1. Mercury という名前のサーバー・リソースを作成し、エージェントと関連付けます。
2. Mercury という名前のセレクターを作成します。 `BF_NAME = Mercury` が指定されたサーバーを選択するようにそのセレクターを設定します。
3. Lincoln という名前のプロジェクトを作成します。 Mercury セレクターをそのプロジェクトに割り当てます。

Lincoln プロジェクトを実行すると、システムは Mercury という名前のサーバー・リソースを選択します。このサーバー・リソースが使用できない場合、プロジェクトは失敗します。



次の例は、一連のサーバーでの動的なサーバー選択のセットアップ方法を示しています。

1. RAMSIZE という名前のコレクターを作成します。ビルトイン・プロパティー MEM\_TOTAL を収集するようにそのコレクターを設定します。
2. ホストと関連付けるサーバー・リソースを作成します。 コレクター RAMSIZE を使用するように各リソースを設定します。
  - Mercury、512 MB RAM のホスト
  - Mars、1 GB RAM のホスト
  - Jupiter、3 GB RAM のホスト
3. BigRam という名前のセレクターを作成します。 標準プロパティー (プロパティー = MEM\_TOTAL、演算子 = GE (以上)、および値 = 2048) を選択するようにそのセレクターを設定します。 MEM\_TOTAL は、MB で表されます。 このセレクターは、2 GB 以上の RAM を持つホストのみを選択します。
4. SmallRam という名前のセレクターを作成します。 標準プロパティー (プロパティー = MEM\_TOTAL、演算子 = GE (以上)、および値 = 256) を選択するようにそのセレクターを設定します。このセレクターは、256 MB 以上の RAM を持つホストのみを選択します。
5. 2 つのプロジェクトを作成します。
  - HighMaint: セレクター BigRam を使用するようにこのプロジェクトを設定します。
  - LowMaint: セレクター SmallRam を使用するようにこのプロジェクトを設定します。

HighMaint を実行すると、システムにより、サーバー Jupiter が選択されます。これは、このサーバー以外に少なくとも 2 GB の RAM を持つというセレクター要件に合致するサーバーがないためです。

LowMaint を実行すると、システムにより、使用可能な 3 つのサーバー・リソースのいずれかが選択されます。

Neptune という名前のサーバー・リソースを 2 GB の RAM を持つホスト用に後で追加すると、次にプロジェクト HighMaint を実行するときに、そのプロジェクトに対して Neptune または Jupiter のいずれかが選択されます。何らかの理由で Jupiter がダウンしていると、Neptune が使用されます。このセレクターに合うのが Neptune のみになるためです。

## マニフェストの表示

サーバーのマニフェストを表示するには、次の手順を実行します。

1. 「サーバー」をクリックします。
2. サーバー・リソースの名前をクリックします。
3. 「マニフェスト」タブをクリックします。

サーバーのマニフェストを直接変更することはできません。マニフェストの内容は、コレクターによって定義されます。

マニフェストは、サーバー・プロパティまたはシステム設定値により決められた間隔で自動的にリフレッシュされます。各間隔の間にリフレッシュを行うには、「キュー・マニフェストの更新」をクリックします。

## 手動でのマニフェストのリフレッシュ

マニフェストを手動でリフレッシュするには、次の手順を実行します。

1. 「サーバー」をクリックして、サーバーのリストを表示します。
2. サーバーを選択します。
3. 「キュー・マニフェストの更新」をクリックします。

サーバーの接続テストが実行され、サーバー・マニフェストが更新されます。リフレッシュが完了するまで、「マニフェスト」タブに「マニフェストを更新しています」という進行状況表示バーが表示されます。

サーバー・マニフェストの自動リフレッシュ間隔は、「サーバー・テスト頻度」システム設定によって設定されます。キューでのマニフェスト手動リフレッシュは、キューでのすべてのマニフェスト自動リフレッシュの前に実行されます。

## サーバー・マニフェストの更新頻度の設定

システム設定値を使用して、システムがマニフェスト・プロパティについてサーバーに照会する頻度を制御できます。

### このタスクについて

コレクターに `OS_SYSNAME` という名前の変数を定義して、コレクターがアサインしたサーバーの Build Forge® エージェント・バージョンを収集することができます。この変数を作成するには、以下の手順を実行します。

### 手順

1. 「サーバー」 > 「コレクター」をクリックします。
2. 既存のコレクターを選択するか新規コレクターを作成します。
3. コレクターに変数を追加します。
  - a. タイプとして「ビルトイン (Built in)」を選択します。
  - b. 変数名 `OS_SYSNAME` を選択します。
  - c. 「保存」をクリックします。
4. 必要に応じて、複数のサーバーのコレクター・プロパティを変更し、編集したコレクターに一致させます。

## 静的および動的なサーバー選択のセットアップ例

次の例は、単一の静的セレクターを作成して使用方法を示しています。

1. Mercury という名前のサーバー・リソースを作成し、エージェントと関連付けます。
2. Mercury という名前のセレクターを作成します。標準プロパティ（プロパティ = `BF_NAME`、演算子 = `EQ` (等しい)、値 = Mercury、および「必須」）を選択するようにそのセレクターを設定します。

3. Lincoln という名前のプロジェクトを作成します。Mercury セレクターをそのプロジェクトに割り当てます。

Lincoln プロジェクトを実行すると、システムは Mercury という名前のサーバー・リソースを選択します。このサーバー・リソースが使用できない場合、プロジェクトは失敗します。

次の例は、一連のサーバーでの動的なサーバー選択のセットアップ方法を示しています。

1. RAMSIZE という名前のコレクターを作成します。ビルトイン・プロパティー MEM\_TOTAL を収集するようにそのコレクターを設定します。
2. ホストと関連付けるサーバー・リソースを作成します。コレクター RAMSIZE を使用するように各リソースを設定します。
  - Mercury、512 MB RAM のホスト
  - Mars、1 GB RAM のホスト
  - Jupiter、3 GB RAM のホスト
3. BigRam という名前のセレクターを作成します。標準プロパティー (プロパティー = MEM\_TOTAL、演算子 = GE (以上)、値 = 2048、および「必須」) を選択するようにそのセレクターを設定します。MEM\_TOTAL は、MB で表されます。このセレクターは、2 GB 以上の RAM を持つホストのみを選択します。
4. SmallRam という名前のセレクターを作成します。標準プロパティー (プロパティー = MEM\_TOTAL、演算子 = GE (以上)、および値 = 256) を選択するようにそのセレクターを設定します。このセレクターは、256 MB 以上の RAM を持つホストのみを選択します。
5. 2 つのプロジェクトを作成します。
  - HighMaint: セレクター BigRam を使用するようにこのプロジェクトを設定します。
  - LowMaint: セレクター SmallRam を使用するようにこのプロジェクトを設定します。

HighMaint を実行すると、システムにより、サーバー Jupiter が選択されます。これは、このサーバー以外に少なくとも 2 GB の RAM を持つというセレクター要件に合致するサーバーがないためです。

LowMaint を実行すると、システムにより、使用可能な 3 つのサーバー・リソースのいずれかが選択されます。

Neptune という名前のサーバー・リソースを 2 GB の RAM を持つホスト用に後で追加すると、次にプロジェクト HighMaint を実行するときに、そのプロジェクトに対して Neptune または Jupiter のいずれかが選択されます。何らかの理由で Jupiter がダウンしていると、Neptune が使用されます。このセレクターに合うのが Neptune のみになるためです。

## プロパティーの参照

以下のトピックは、マニフェストに影響を及ぼすプロパティーについて説明しています。

- 『ビルトイン・プロパティ』
- 317 ページの『事前設定されたプロパティ』
- 319 ページの『コレクターの特殊なプロパティ』

## ビルトイン・プロパティ

管理コンソールは、サーバーからビルトイン・プロパティを収集してから、それらの値をサーバー・マニフェストに割り当てます。

## ビルトイン・プロパティ

ビルトイン・プロパティは、ビルド・システム内にある、以下のような異なる複数のデータ・オブジェクトによって使用されます。

- **セクター**は、ビルトイン・プロパティをセクター変数として使用して、サーバーとこれらのプロパティの特定の値を突き合わせるすることができます。
- **コレクター**は、ビルトイン・プロパティを使用して、サーバーからデータを収集します。
- **マニフェスト**は、ビルトイン・プロパティの値が収集された場合に、その値を保管します。

ビルトイン・プロパティは自動的に追加されません。ビルトイン・プロパティをマニフェストに表示するには、プロパティ用のコレクターにビルトイン・プロパティを追加する必要があります。この表では、ビルトイン・プロパティをリストし、説明します。

注: プロパティの可用性は、プラットフォームによって異なります。

表 8. コレクターおよびマニフェストのビルトイン・プロパティ

プロパティ	説明
CPU_ARCH	アーキテクチャー名の戻り値は label で、以下に示すようになります。 <ul style="list-style-type: none"> <li>• HP-PA: HP Precision Architecture</li> <li>• IA-64: Intel Itanium</li> <li>• MVS: IBM S/390</li> <li>• PPC: PowerPC</li> <li>• PPC-64: PowerPC 64</li> <li>• SPARC: Sun SPARC</li> <li>• X86: Intel、AMD、Cyrix などで使用される x86 対応アーキテクチャー</li> </ul>
CPU_LOAD (Windows のみ)	CPU 負荷または CPU 使用量は容量のパーセンテージ (0 から 100 まで) として表されます。

表 8. コレクターおよびマニフェストのビルトイン・プロパティ (続き)

プロパティ	説明
CPU_LOAD1	<p>直前の 1 分間に実行されていたか、実行を待機していたプロセスの平均数 (ロード平均) が、戻り値によって報告されます。</p> <p>CPU_LOAD1 は CPU アクティビティの尺度です。活動停止中のコンピューターのロード数は 0 です。CPU を使用しているか、CPU の使用を待機している各プロセスにより、ロード数が 1 ずつ加算されます。</p> <p><b>注:</b> Windows では、すべてのプロセスは、アクティブであるかどうかにかかわらず、ロード数を追加します。また、この情報は、bfdispatch プロセスによって収集され、共用メモリ・セグメントを使用してエージェントに公開されます。エージェントに接続するために使用されるユーザー資格情報に特権がない場合、これらの統計は使用できません。</p>
CPU_LOAD5	<p>直前の 5 分間に実行を待機していたプロセスの平均数 (ロード平均) が、戻り値によって報告されます。</p> <p>CPU_LOAD5 は CPU アクティビティの尺度です。活動停止中のコンピューターのロード数は 0 です。CPU を使用しているか、CPU の使用を待機している各プロセスにより、ロード数が 1 ずつ加算されます。</p> <p><b>注:</b> Windows では、すべてのプロセスは、アクティブであるかどうかにかかわらず、ロード数を追加します。また、この情報は、bfdispatch プロセスによって収集され、共用メモリ・セグメントを使用してエージェントに公開されます。エージェントに接続するために使用されるユーザー資格情報に特権がない場合、これらの統計は使用できません。</p>
CPU_LOAD15	<p>直前の 15 分間に実行を待機していたプロセスの平均数 (ロード平均)。</p> <p>CPU_LOAD15 は CPU アクティビティの測定です。活動停止中のコンピューターのロード数は 0 です。CPU を使用しているか、CPU の使用を待機している各プロセスにより、ロード数が 1 ずつ加算されます。</p> <p><b>注:</b> Windows では、すべてのプロセスは、アクティブであるかどうかにかかわらず、ロード数を追加します。また、この情報は、bfdispatch プロセスによって収集され、共用メモリ・セグメントを使用してエージェントに公開されます。エージェントに接続するために使用されるユーザー資格情報に特権がない場合、これらの統計は使用できません。</p>
CPU_MHZ	<p>このプロパティにより、プロセッサ速度がメガヘルツで報告されます。このプロパティを正しく入力するには、以下の一定の条件が必要です。</p> <ul style="list-style-type: none"> <li>• Linux の場合: 周波数のスケーリングが使用可能である必要があります。</li> <li>• Windows の場合: ~MHz レジストリー項目が存在し、入力されている必要があります。</li> <li>• x86 および x86-64 プロセッサの場合: インライン・アセンブリが機能している必要があります。</li> </ul>

表 8. コレクターおよびマニフェストのビルトイン・プロパティ (続き)

プロパティ	説明
CPU_MANUFACTURER	<p>このプロパティにより、プロセッサ製造メーカーの企業名が戻されます。情報を直接入手できない場合、これらの名前はアーキテクチャーを基に想定されます。十分なプロセッサ情報が入手できなければ、値は戻されません。サポートしている値は、以下のとおりです。</p> <ul style="list-style-type: none"> <li>• AMD: x86 および AMD64 プロセッサの場合</li> <li>• Cyrix: x86 互換プロセッサの場合</li> <li>• DEC: Alpha および VAX の場合</li> <li>• HP: Hewlett-Packard Precision Architecture</li> <li>• IBM: IBM S/390 および PowerPC G5</li> <li>• Intel: Intel x86 (Intel64 を含む)、IA-64 Itanium</li> <li>• Motorola: PowerPC G4</li> <li>• NexGen: x86 互換プロセッサ</li> <li>• National: National Semiconductor x86 互換プロセッサ</li> <li>• Rise: Rise x86 互換プロセッサ</li> <li>• Sis: Sis x86 互換プロセッサ</li> <li>• Sun: Sun Microsystems SPARC</li> <li>• TransMeta: TransMeta x86 互換プロセッサ</li> <li>• UMC: UMC x86 互換プロセッサ</li> <li>• VIA: VIA Technologies x86 互換プロセッサ</li> </ul>
CPU_MODEL	<p>このプロパティにより、製造メーカー固有の CPU 型式番号が戻されます。報告される値は、以下のとおりです。</p> <ul style="list-style-type: none"> <li>• x86 アーキテクチャー <ul style="list-style-type: none"> <li>– 386</li> <li>– 486</li> <li>– 586</li> <li>– 686</li> <li>– X86_64</li> </ul> </li> <li>• PowerPC アーキテクチャー <ul style="list-style-type: none"> <li>– 6xx</li> <li>– POWER</li> <li>– RS64</li> <li>– G3</li> <li>– G4</li> <li>– G5</li> <li>– Cell</li> </ul> </li> </ul>

表 8. コレクターおよびマニフェストのビルトイン・プロパティ (続き)

プロパティ	説明
CPU_SERIAL	<p>このプロパティにより、CPU またはコンピューターのシリアル番号が戻されます。現在、この機能は以下のアーキテクチャーに制限されています。</p> <ul style="list-style-type: none"> <li>• x86: Intel または Transmeta シリアル番号のみ。注: ほとんどの x86 プロセッサは、シリアル番号を報告しません。このような場合に返される値はありません。</li> <li>• MacOS/X: 割り当て済みのシリアル番号は、入出力レジストリーから取得されます。このプロパティを報告するには、CoreFoundation フレームワークと IOKit フレームワークが入手可能でなければなりません。</li> </ul>
DISK_FREE	<p>UNIX および Linux の場合、サーバー・パスによって指定された、ファイル・システムのフリー・スペースの量 (MB 単位) が、このプロパティによって戻されます。</p> <p>Windows の場合、サーバー・パスによって指定された、ドライブ上のフリー・ディスク・スペース (MB) です。</p> <p>例えば、4 GB のフリー・スペースを持つディスクは、4096 MB と報告されます。</p>
DISK_TOTAL	<p>このプロパティにより、使用可能な空きディスク・スペースの合計が戻されます。この値はエージェントの基本パス用に報告されます。このパスには、残りのディスクまたは区画よりもサイズの小さいスペースが別個に割り振られている場合があります。ディスク・スペース管理は、オペレーティング・システムごとに大きく異なります。</p>
MEM_LOAD (UNIX/Linux のみ)	<p>UNIX および Linux の場合、現在使用中の RAM またはシステム・メモリーの量は、実メモリー合計のパーセンテージ (0 から 100 まで) として表されます。</p>
MEM_FREE	<p>このプロパティにより、RAM またはシステム・メモリーの量 (MB 単位) が戻されます。例えば、2 GB の空き RAM は、2048 MB として報告されます。</p>
MEM_PAGESIZE	<p>このプロパティにより、RAM またはシステム・メモリーのページ・サイズ (MB 単位) が戻されます。この数字は、ホスト・システムの標準のページ・サイズを表します。例えば、4 KB のページ・サイズを持つホスト・システムは、4096 MB として報告されます。</p>
MEM_TOTAL	<p>このプロパティにより、RAM またはシステム・メモリーの合計 (MB 単位) が戻されます。</p> <p>例えば、2 GB の RAM を持つコンピューターは、2048 MB として報告されます。</p>
NET_FQDN	<p>このプロパティにより、エージェントを実行しているコンピューターの完全修飾ドメイン名 (FQDN) が戻されます。FQDN は、エージェントが通信に使用しているアドレスを基にして報告されます。戻されるアドレスは、実際に使用されているアドレスに応じて、IPv4 または IPv6 に基づいたものになります。NET_IPV、NET_IPV4、および NET_IPV6 も参照してください。</p>
NET_HWADDR	<p>このプロパティにより、NET_IFACE で報告されたインターフェースのハードウェア・アドレスが戻されます。</p>
NET_IFACE	<p>このプロパティにより、エージェントが通信に使用するインターフェースの名前が戻されます。</p> <ul style="list-style-type: none"> <li>• Windows では、報告される名前は、ipconfig コマンドによって戻される名前です。例えば、Intel(R) PRO/100 VE Network Connection - Packet Scheduler Miniport です。</li> <li>• 他のオペレーティング・システムでは、報告される名前は、ifconfig によって戻される名前です。例えば、en0 または eth0 または OSA1 です。</li> </ul>
NET_IPV	<p>このプロパティにより、エージェントとの通信に使用される IP 接続のタイプ (IPv4 の場合は 4、IPv6 の場合は 6) が戻されます。</p>



表 8. コレクターおよびマニフェストのビルトイン・プロパティ (続き)

プロパティ	説明
NET_IPV4	このプロパティにより、エージェントが通信に使用する IPv4 アドレスが戻されます。IPv6 を介した接続で、エージェントが同じインターフェースで IPv4 アドレスを識別できる場合は、そのアドレスが報告されます。
NET_IPV6	このプロパティにより、エージェントが通信に使用する IPv6 アドレスが戻されます。
NET_SPEED (Windows のみ)	このプロパティにより、インターフェースの速度 (Mb/秒単位) が戻されます。例えば、ギガビット・イーサネットでは 1000 です。
NUM_CPU	このプロパティにより、コンピューター上の CPU の数が戻されます。
OS_HOSTID	このプロパティにより、gethostid() システム呼び出しの結果が戻されます。通常、この結果は、システム管理者が /etc/hostid を有益な値に設定していない限り、参考になりません。
OS_SYSNAME	このプロパティにより、サーバーのオペレーティング・システムの名前が戻されます。例えば、Microsoft Windows XP、AIX、Macintosh OS などです。
OS_RELEASE	このプロパティにより、サーバーのオペレーティング・システムのリリース・レベルが戻されます。  例えば、サーバーのオペレーティング・システムが Microsoft XP バージョン 5.1.2600 の場合、この戻り値は 5 です。
OS_VERSION	このプロパティにより、サーバーのオペレーティング・システムのバージョンが戻されます。  例えば、サーバーのオペレーティング・システムが Microsoft XP バージョン 5.1.2600 の場合、この戻り値は 1 です。
WIN_SERVICEPACK (Windows のみ)	このプロパティにより、サーバーにインストールされている Windows Service Pack のバージョン番号が戻されます。例えば、Service Pack 2 の場合、この値は 2 です。

## 事前設定されたプロパティ

いくつかのマニフェスト・プロパティは、自動設定されます。

自動設定されるマニフェスト・プロパティには以下があります。ビルトイン・プロパティとは異なり、これらのプロパティは、コレクターに追加しなくても値を取り込むことができます。

表の説明中で**セクター**というマークが付いているプロパティは、セクターで使うことができます。それ以外のものは、マニフェストの情報を提供するだけです。

表 9. 自動設定されたマニフェスト・プロパティ

プロパティ	説明
BF_AGENT_VERSION	<b>セクター</b> 。サーバーにインストールされているエージェントのバージョン番号。

表 9. 自動設定されたマニフェスト・プロパティ (続き)

プロパティ	説明
BF_EXCLUSIVE	<p><b>セレクト</b>。フラグであり、演算子または値はありません。セレクトにこのプロパティが含まれている場合、ジョブの実行中に、選択されたサーバー上のすべてのスロットが占有されます。</p> <p>ジョブ内のステップで別のサーバーが実行場所に指定されている場合、別のサーバー上でステップが実行されている間も、現行サーバー上のすべての現行スロットは引き続き占有されています。</p>
BF_JOBS	<p><b>セレクト</b>。サーバー・リソースが同時に実行されているジョブ (ステップ) の数。この値は、他のマニフェスト・プロパティの更新とは関係なく、コンソールがサーバーにステップを割り当てるたびに更新されます。</p>
BF_LAST_REFRESH	<p>マニフェストのビルトイン・プロパティの最終更新時刻。値は、UNIX<sup>®</sup> タイム・スタンプ (1970 年 1 月 1 日以降の秒数) として報告されます。</p>
BF_LASTJOBS	<p>マニフェストが最後にリフレッシュされたときに、サーバーで実行されていたジョブの数。</p>
BF_LAST_UPDATE	<p>実行コマンド・プロパティのマニフェストへの最終更新時刻。値は、UNIX<sup>®</sup> タイム・スタンプ (1970 年 1 月 1 日以降の秒数) として報告されます。</p>
BF_LOADRATIO	<p><b>セレクト</b>。比率で報告された計算値。ジョブの数 (BF_JOBS) をサーバーで許可されるジョブの最大数 (最大ジョブ数設定値) で割った値。1 つのジョブが実行中で、最大ジョブ数が 4 のサーバーの場合、そのロード比率は 0.25 です。</p>
BF_NAME	<p><b>セレクト</b>。実行場所となるサーバー・リソースを指定します。値は、サーバー・リソースの名前です。BF_NAME プロパティはマニフェスト・リストには表示されません。</p>
BF_RESERVE	<p><b>セレクト</b>。フラグであり、演算子または値はありません。セレクトにこのプロパティが含まれる場合、ジョブの実行中に、選択されたサーバー上でスロットが占有されます。</p> <ul style="list-style-type: none"> <li>ジョブ内のステップで別のサーバーが実行場所に指定されている場合、選択されたサーバー上のスロットは、別のサーバー上でステップが実行されている間、占有されます。</li> <li>選択されたサーバーをステップが明示的に指定している場合、サーバーは、そのステップ用に占有されたスロットを使用します。</li> </ul> <p>複数のステップが別のサーバー上で実行されると、プロジェクトはサーバー上のスロットを解放し、遅延が発生します。このフラグを設定すると、このような遅延を防ぐことができます。</p>

## コレクターの特殊なプロパティ

コレクターで一部の設定値プロパティを指定すると、システムで特定の動作が発生します。これらのプロパティの先頭には、下線文字が付きます。ビルド・システムは、これらのプロパティの値を使用して、コレクターからこれらのプロパティを取得するサーバーに、動作を適用します。

注: スtring「BF\_」で始まるプロパティ名はシステムで占有されているため、このような名前のプロパティは作成できません。

表 10. コレクターおよびマニフェストの特殊な設定値プロパティ

プロパティ	説明
<u>_AGE</u>	このプロパティは、マニフェストがリフレッシュされる頻度を定義する。デフォルト値は 86400 で、1 日に 1 回リフレッシュされます。3600 という値を設定すると、システムは、1 時間ごとにマニフェストを更新します。

## セクターの新規インスタンスを作成するためのスナップショットの使用

変更または修正するセクターの新規インスタンスを素早く作成するには、セクターのスナップショットを作成します。

### セクター・スナップショットの概要

以下のトピックを参照して、セクター・スナップショットについて学習し、その使用方法を理解してください。

### セクター・スナップショットの使用例

以下の例では、セクター・スナップショットの一般的な使用例を説明します。


- セクターのスナップショットを作成して、既存セクターでのジョブの実行を続行しながら、セクター構成を変更したり、新規ツールやスクリプトのテストを行ったりします。
- セクター・スナップショットを、一時バックアップまたは正式なアーカイブの一部として保管します。
- セクターのスナップショットを作成して、外部リリースや内部リリースのようなマイルストーンに対応するポイント・イン・タイム・セクター構成を取り込みます。

### セクター・スナップショットの概念および用語

スナップショットでは、セクターでの作業に関する新しい概念と用語がいくつか導入されています。


**セクターのスナップショット:** スナップショットとは、既存のセクターの新規インスタンスのことです。スナップショットに関して注意すべきいくつかのキーポイントを次に示します。

- スナップショットは、独立したセクター・オブジェクトです。 スナップショット・セット内の 1 つのスナップショットを変更しても、そのセット内の他のスナップショットは変更の影響を受けません。
- スナップショットはコピーではありません。セクターに関連したオブジェクトのスナップショットを作成すると、スナップショットによって独立したオブジェクトのインスタンスが生成されます。コピーではオブジェクト間の関係がマップされますが、オブジェクトは新規作成されません。
- スナップショットはセクターの改訂ではありません。
  - スナップショットでは、2 つのセクター・スナップショット間の変更点は比較されません。
  - セクター・スナップショットの変更点は、ソース制御システムの場合のようにバージョン番号でトラッキングまたは識別されません。 ただし、バージョン番号 (例えば、7.5.0, 3.4.01) を含むスナップショットの命名体系を使用することで、セクターのスナップショットをマイルストーンに関連付けることができます。


**スナップショット・セット:** スナップショット・セットは、基本スナップショットの子孫であるすべてのセクター・スナップショットのセットです。 このセットには少なくとも、基本スナップショットまたは親スナップショットと、子スナップショットが含まれます。 UI で、セクター名の横にある「スナップショット」アイコン  は、そのセクターにスナップショット・セットが存在することを示します。

**基本スナップショット:** 初期の状態では、すべてのセクターに基本スナップショットのスナップショット名が付けられています。 Base Snapshot は別の名前に変更できます。基本スナップショットは、スナップショット・セットの親です。

**デフォルトのセクター・スナップショット:** デフォルトのセクター・スナップショットは、現行の作業セクターです。セット内の 1 つのスナップショットのみをデフォルトにすることができます。デフォルト・スナップショットを指定しなかった場合、基本スナップショットがデフォルトになります。

- コンソールのデフォルト・スナップショットは、セクター・リストの最上位に表示されます。 セクター・リストを表示するには、「サーバー」>「セクター」と選択します。
- 複数のスナップショットでセクターを選択すると、リスト・ボックスで別のセクター・スナップショットを選択しない限り、デフォルトのセクター・スナップショットが使用されます。
- セクター・スナップショット・セットの他のスナップショットにアクセスして作業を行うには、「スナップショット」  アイコンをクリックする必要があります。

## セクターのスナップショット・ビュー

「スナップショット」ビューを表示するには、「スナップショット」アイコン  を選択します。「スナップショット」ビューには、セット内のスナップショットの階層が次のように表示されます。

- 基本スナップショットが最上位にあります。固有の名前を割り当てていない場合、これには `Base Snapshot` という名前が付いています。
- セレクター・スナップショットはすべて基本スナップショットの子です。同じ基本スナップショットの子は、「スナップショット」列の同じレベルにインデントされています。
- 子のスナップショットから作成されたセレクターのスナップショットは、子のスナップショットの子になり、「スナップショット」列内の次のレベルでインデントされます。

## セレクター・スナップショットに関する計画

デフォルトのセレクター・スナップショットの選択、およびセレクター・スナップショットの命名に関するベスト・プラクティスをいくつか検討します。

### • セット内のデフォルト・スナップショットの選択方法

UI では、スナップショット・セットに対して、1 つのデフォルトまたは現行のセレクター・スナップショットのみが認識されます。次のような一貫性のある方法で、デフォルト・スナップショットを選択してください。

- 基本スナップショットをデフォルト・スナップショットとして使用する

スナップショットをバックアップとして使用します。基本スナップショットのみに変更を行います。バックアップされたセレクター・スナップショットは変更しないでください。

- 最新のスナップショットをデフォルト・スナップショットとして使用する

新しいセレクター・スナップショットを作成するたびに、それを新しいデフォルトのセレクター・スナップショットにします。基本セレクター・スナップショットや以前のセレクター・スナップショットは変更しないでください。

### • セットのスナップショット命名体系を識別する

セレクターのスナップショット名は、セレクターのスナップショット・セット内で固有である必要があります。

以下の基準を使用して、セレクターのスナップショット名を作成します。

- 説明的な名前にし、スナップショットの使用法や目的が分かるようにします。
- 命名体系が、定義された標準に従うようにします。「スナップショット」タブの「コメント」ボックスを使用して、命名体系を説明することができます。




### • セットの単一セレクター名の使用

セレクター・スナップショットを作成した後で、セレクター名を変更するかどうかを選択できます。セレクター名を変更すると、セレクター・スナップショットごとに名前が更新されます。

## 既存のセレクターまたはセレクター・スナップショットからのセレクター・スナップショットの作成

セレクターのスナップショットを作成すると、セレクターの新規インスタンスが作成されます。スナップショットはコピーではなく、新規のセレクターです。

**ヒント:** スナップショットを作成し、デフォルト・スナップショットを変更するには、ユーザーは必要な権限を保持している必要があります。398 ページの『スナップショット権限のためのアクセス・グループの検証および編集』を参照してください。

1. スナップショットを作成するセレクトターまたはセレクトター・スナップショットの横にある「編集」アイコンをクリックします。
  - デフォルトのセレクトター・スナップショットのスナップショットを作成するには、セレクトター・リスト（「サーバー」>「セレクトター」）で、最上位のスナップショットの横にある「編集」アイコン  をクリックします。
  - デフォルト以外のセレクトター・スナップショットのスナップショットを作成するには、「スナップショット」アイコン  をクリックします。「スナップショット」ビューに、セレクトター・スナップショットがセットで表示されます。デフォルト以外のセレクトター・スナップショットの横にある「編集」アイコン  をクリックします。
2. 「新規スナップショットの作成」をクリックします。
3. 「スナップショット」タブの「名前」にスナップショット名を入力します。この名前は、セレクトターを使用してスナップショットを作成するすべてのオブジェクトに割り当てられます。

この名前は、セレクトターのスナップショット・セット内で固有である必要があります。

4. セレクトター・スナップショット作成時にスナップショットを作成する Build Forge オブジェクトを選択します。 選択可能なオブジェクトについては、下の表に説明があります。

オブジェクト	説明
デフォルト	UI のデフォルト・スナップショットは、セレクトター・リストの最上位に表示されます。  セレクトター・リストを表示するには、「サーバー」>「セレクトター」と選択します。
含まれるセレクトターに従う	セレクトターが「組み込み」プロパティ・タイプを使用して他のセレクトターを組み込む場合、組み込まれるセレクトターのスナップショットが作成されます。 <b>注:</b> 「組み込み」変数タイプは、これまでのリリースで提供されていた <code>.include</code> 機能に代わるものです。

5. 「保存」をクリックして、セレクトターのスナップショットを保存します。



## デフォルトのセレクトター・スナップショットの変更

デフォルトのセレクトター・スナップショットは、セレクトター・スナップショット・セット内の最上位のスナップショットです。このスナップショットはセレクトター・リスト（「サーバー」>「セレクトター」と選択）に表示されます。



**ヒント:** スナップショットを作成し、デフォルト・スナップショットを変更するには、ユーザーは必要な権限を保持している必要があります。398 ページの『スナップショット権限のためのアクセス・グループの検証および編集』を参照してください。

デフォルトのセクター・スナップショットを変更するには、新規デフォルトとして設定するスナップショットのスナップショット定義を以下の手順で編集します。

1. 「サーバー」>「セクター」を選択します。
2. セクター・リストで、デフォルトのセクター・スナップショットの「スナップショット」アイコン  をクリックします。
3. スナップショット・リストで、新規デフォルトとして設定するセクター・スナップショットの「編集」アイコン  をクリックします。
4. 「デフォルトの設定」をクリックします。
5. **重要:** ポップアップで「OK」または「取り消し」を選択します。



OK	参照を更新します: 前のデフォルト・セクターを参照するオブジェクトがある場合、新規のデフォルト・セクターを使用するようにオブジェクトを更新します。
取り消し	参照を更新しません: 前のデフォルトを参照するオブジェクトがある場合、新規のデフォルト・セクターを使用するようにオブジェクトを更新しません。

## セクター・スナップショットのスナップショット名の変更

セクター・スナップショットのスナップショット名を変更できます。また、セクターのスナップショット作成時に、スナップショットの作成対象として選択したオブジェクトのスナップショット名も変更できます。

基本スナップショットの場合は、このオプションを使用して、基本スナップショットのデフォルト名を別のスナップショット名に変更できます。これは、単一のセクター・スナップショット、または現行および新規のすべてのセクターに対して実行できます。

スナップショット名を変更するには、次の手順に従ってください。

1. 「サーバー」>「セクター」を選択します。
2. セクター・リストで、デフォルトのセクター・スナップショットの「スナップショット」アイコン  をクリックします。
3. スナップショットのリストで、セクター・スナップショットの「編集」アイコン  をクリックします。
4. 「スナップショット」タブを選択します。
5. 「名前」に、新規名を入力します。
6. **オプション:** 「コメント」にコメントを入力します。
7. **重要:** ポップアップで「OK」または「取り消し」を選択します。



OK	<p>セクターのスナップショット名と他のスナップショットのオブジェクト名を変更します: セクター・スナップショット作成時にスナップショットの作成対象として選択したオブジェクトについて、オブジェクト名とセクター・スナップショット名を変更します。</p> <p>基本スナップショットの場合: 現行のすべてのセクター・スナップショットと新規のすべてのセクター・スナップショットの基本スナップショット名を変更します。</p>
取り消し	<p>セクターのスナップショット名は変更しますが、他のスナップショットのオブジェクト名は変更しません: セクター・スナップショット作成時にスナップショットの作成対象として選択したオブジェクトについて、オブジェクト名を変更しません。セクター・スナップショット名のみを変更します。</p> <p>基本スナップショットの場合: 現行のすべてのセクター・スナップショットと新規のすべてのセクター・スナップショットの基本スナップショット名を保持します。</p>

## セクター・スナップショット・セット内のスナップショットのアクセスおよび表示


変更または修正するセクターの新規インスタンスを素早く作成するには、セクターのスナップショットを作成します。

セクター・スナップショットを作成すると、少なくとも 2 つの環境変数 (基本セクターと新規セクター・スナップショット) を含むスナップショット・セットが生成されます。



スナップショット・セット内のセクター・スナップショットをすべて表示するには、以下の手順を実行します。

1. 「サーバー」 > 「セクター」を選択します。

セクター・リストに、セクターおよびセクターのスナップショットのリストが表示されます。最上位のスナップショットがデフォルトのセクター・スナップショットです。

2. 「スナップショット」アイコン  をクリックすると、スナップショット・セット内のセクター・スナップショットが表示されます。

「スナップショット」ビューでは、以下を行うことができます。

- 新規のセクター・スナップショットを作成する。作業を開始するには、「編集」アイコン  をクリックします。
- セクターのデフォルト・スナップショットを変更する。「編集」アイコン  をクリックして、「デフォルトの設定」をクリックします。
- 標準セクターの場合と同じ要領で、セクター・スナップショット定義を編集する。

## セレクター・スナップショットの削除

「削除」オプションを使用して、セレクターのスナップショットを削除することができます。

別のオブジェクトで使用している場合は、セレクターを削除することはできません。例えば、別のセレクターに組み込まれている場合や、プロジェクト、ステップまたはスケジュールで使用している場合は、セレクターは削除できません。


セレクターのスナップショットを削除する場合は、次の手順を実行します。

1. 「サーバー」>「**セレクター**」を選択します。
2. セレクター・リストで、基本スナップショットの「**スナップショット**」アイコン



をクリックします。

「スナップショット」ビューに、セレクター・スナップショットがセットで表示されます。

3. 削除するセレクター・スナップショットの横にある「**編集**」アイコン  をクリックします。
4. 「**削除**」をクリックします。



---

## 第 18 章 環境変数の処理

このセクションでは、環境変数のセットアップ方法および管理方法について説明します。

---

### 環境変数について

環境変数は、変数の名前付きセットです。

定義された環境変数は、以下を行うために使用できます。

- ステップがジョブによって使用されるように変数を設定します。環境変数は、サーバー、プロジェクト、およびステップに割り当てられます。ジョブ実行時に、実行中のステップは、次の 3 つの環境から変数値を継承します。
  - ジョブが実行されるサーバー環境。
  - ステップが定義されているプロジェクト環境。
  - ステップに関連付けられているステップ環境。
- スケジュールに入れられたジョブが使用するように変数を設定します。スケジュールに入れられたジョブに対して設定された環境変数によって、プロジェクトに指定された環境変数が置き換えられます。
- アダプターが使用するように変数を設定します。環境変数は、アダプター・リンクに割り当てることができます。これは、プロジェクトの最初のアダプター・ステップによって使用されます。

### 環境の継承

システムは、ステップを実行する前に、ステップ環境を作成します。ステップ環境は、ステップに適用可能なすべての変数から構成されます。変数は、サーバー環境、プロジェクト環境、およびステップ環境から順に継承されます。以下は、基本的なケースです。

1. サーバー環境: サーバー環境変数は、ステップ環境にコピーされます。
2. プロジェクト環境: プロジェクト環境変数は、ステップ環境に適用されます。プロジェクト環境にサーバー環境の変数と同じ名前の変数が含まれている場合、値は両方の変数定義の変数アクション に従って更新されます。
3. ステップ環境: ステップ環境変数は、ステップ環境に適用されます。ステップ環境の変数が、サーバー環境およびプロジェクト環境から継承された変数と同じ名前を持つ場合、値は変数アクション にしたがって更新されます。

注: Java MJC では、Perl エンジンからの異なる環境の継承がサポートされていません。Java MJC では、変数はプロジェクト環境、サーバー環境、およびステップ環境から、この順に継承されます。

変数アクションは、継承時の値の適用方法に直接影響します。 例:

- ケース 1: 変数アクションが「設定」の場合に、値が継承により上書きされません。

- サーバー環境: X = 1、アクション: 設定
- プロジェクト環境: X = 100、アクション: 設定
- ステップ環境: X = 3、アクション: 設定
- ステップ実行中の最終値: X = 3

変数 X は、1、100、および 3 に設定されます。新規の値が適用されるたびに、「設定」変数アクションにより、変数値が置き換えられます。

- ケース 2: 変数アクション「設定 (未設定の場合)」により、値が継承されます。
  - サーバー環境: Y = 1、アクション: 設定
  - プロジェクト環境: Y = 100、アクション: 設定
  - ステップ環境: Y = 3、アクション: 設定 (未設定の場合)
  - ステップ実行中の最終値: X = 100

変数 Y は、サーバー環境およびプロジェクト環境での Y に対する「設定」アクションのために、1 および 100 に設定されます。Y でステップ環境の変数アクション「設定 (未設定の場合)」が使用されるため、プロジェクト環境で設定された値が継承されます。

## 継承の特殊なケース

以下のケースは、継承に影響します。

### インライン・プロジェクト

ステップでは、そのステップの「インライン」プロパティのプロジェクトを指定することにより、プロジェクトをインライン化します。ステップがプロジェクトをインライン化するとき、呼び出されたプロジェクトのサーバー環境およびプロジェクト環境は使用されません。継承は、次の順序で行われます。

1. 呼び出しステップ用のサーバー環境
2. 呼び出しステップ用のプロジェクト環境
3. 呼び出しステップ用のステップ環境
4. 呼び出されたプロジェクトの各ステップ用のステップ環境 (指定されている場合)

### チェーニングされたプロジェクト

プロジェクトまたはステップは、成功 または失敗 チェーンとしてプロジェクトを指定することができます。プロジェクトは、この方法で呼び出されると、それ自体の環境で実行されます。さらに、呼び出しプロジェクトまたはステップからすべての変数へアクセスします。変数は、接頭部

BF\_CALLER\_ を使用した新規名にコピーされます。例: 呼び出しプロジェクトまたは呼び出しステップの変数 BF\_NAME は、呼び出されたプロジェクトおよび呼び出されたステップでは BF\_CALLER\_BF\_NAME として使用できます。

### スケジュールに入れられたジョブ

「スケジュール」でプロジェクトをスケジュールに入れるときに、デフォルトとは異なる環境をプロジェクトに適用することを選択できます。環境が指定されると、「環境変数」タブを使用して、環境変数に値を設定できます。変数は、「プロジェクト動作」プロパティ設定に従い、「環境変数」タブ

に表示されます。スケジュールに入れられていないジョブを開始するために同じ規則が存在する場合は、その規則が遵守されます。

#### 継承順序のオーバーライド

システム設定「サーバー環境を最後に適用する (Apply server environment last)」を使用して、継承順序をオーバーライドします。値が「はい」の場合、継承順序は次のように設定されます。

1. プロジェクト環境
2. ステップ環境
3. サーバー環境

## ジョブの開始時に行われるプロジェクト変数の変更

ジョブを開始するときに、プロジェクトを変更して、プロジェクト環境変数に設定されている変数値をオーバーライドできます。

ジョブを開始すると、プロジェクト環境変数からの変数が**ジョブ開始**ページに示されます。各変数の「プロジェクト動作」プロパティに応じて、必要であれば、示された値を変更できます。

ジョブ開始時に加えられた変更は、プロジェクト環境変数に定義されている変数と同じ継承規則に従います。

例:

1. 初期値 (JavaVersion = 1.4) の変数を持つように JavaEnv 環境変数を定義します。
2. JavaEnv 環境変数を使用するようにプロジェクト MyBuild を定義します。
3. ジョブを起動してプロジェクト MyBuild を実行します。「**ジョブの開始**」パネルで、JavaVersion の値を 1.5 に変更します。

結果は、以下のようになります。

- プロジェクト環境変数 *JavaEnv* をオーバーライドしないステップは、変更された JavaVersion 値 1.5 を継承します。
- ステップ環境変数として JavaEnv 環境変数を明示的に使用するステップは、プロジェクト環境変数に定義されている JavaVersion の値 1.4 を使用します。

---

## 変数について

変数は、環境内で定義されます。

値に加えて、変数には、解釈されたときに振る舞いに影響する追加プロパティがあります。

UI Config Console Reports Logout: Root User

Environments > MyEnv Add Environment Variable Help ?

Filter Showing 1 - 3 of 3 Display All Page 1 of 1

Name	Type	Value	Action	On Project
MyStandardLevel	Standard Variable	High	Set	Normal
Flavor	Pulldown List		Set	Required
Include	Include	suppress ENV		

(New Variable) Save Variable Delete Variable

Details Pulldown Options

Type: Standard Variable Name: Value:

Action: Set On Project: Normal

新規変数を作成するには、次の手順を実行します。

1. 左メニューで、「環境変数」をクリックします。
2. 「環境変数値の追加」をクリックします。
3. 変数の名前を指定します。
4. 必要に応じて変数の他のプロパティを指定します。詳しくは、332 ページの『環境変数のプロパティ』を参照してください。
5. 「変数の保存」をクリックします。

変数が作成されると、クリックして編集することができます。

## ステップにおける変数の解釈

ステップ・コマンドや環境変数の定義で、UNIX® スタイルまたは Windows® スタイルのいずれかの変数構文を使用することができます。

システムはプリプロセッサを使用して、UNIX スタイル (\$VAR) または Windows スタイル (%VAR%) のいずれの構文も、ステップが実行されるサーバーに適した形式に解釈します。事前構文解析機能によって、Windows ベース・サーバーまたは UNIX ベース・サーバーのいずれかでステップを実行できます。

例:

- ステップにおいて、次の 2 つの割り当てステートメントは等価です。

```
echo %fooVar%      # Windows の構文
echo $fooVar       # UNIX または Linux の構文
```

- 変数割り当ては事前処理されません。そのため、コマンド行での直接割り当ては、特にオペレーティング・システムの制限を受けずにサーバーを選択できるような場合には、使用しないでください。環境変数の中で変数を使用します。

```
set fooVar=100     # Windows
fooVar=200         # UNIX または Linux の構文
```

変数の構文解析方法は、以下のとおりです。



1. 事前構文解析機能によって、変数割り当てが評価されます。特殊文字は、円記号によってエスケープされなければ、取り込まれます (\$、%、{、}、"、')。事前構文解析をオフにすると、すべての文字が渡されます。
2. 変数割り当ての各側は、ターゲットの環境変数によって評価されます。
3. 評価された変数割り当てが実行されます。

事前構文解析機能、Windows 環境変数、およびさまざまな UNIX と Linux のシェルでは、特殊文字をそれぞれ異なる方法で解釈します。特殊文字と、円記号のエスケープ文字の使用には、注意が必要です。

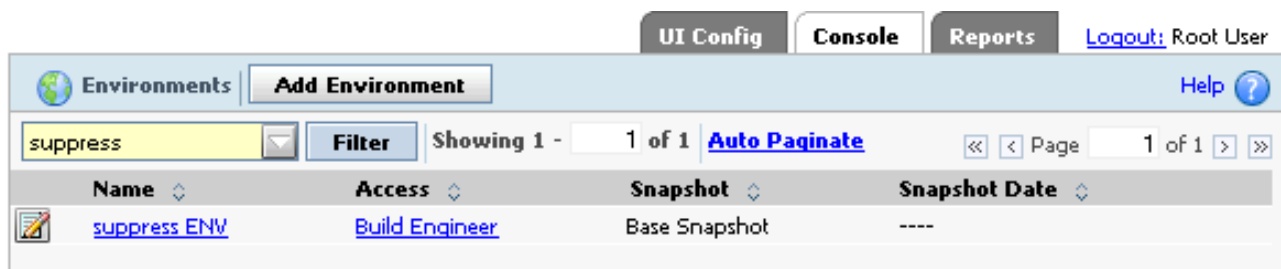
## 未宣言変数の解釈

ステップで呼び出された変数が未定義である場合、戻される値は、事前解析がオンになっている (デフォルト) かオフになっているかによって決まります。事前解析の動作は、bfaagent.conf ファイル内の no\_preparse コマンドまたは \_NO\_PREPARSE\_COMMAND 環境変数を編集することにより、設定します。344 ページの『トリガー変数のリファレンス』を参照してください。

変数フォーマット	戻される値 - 事前解析がオンの場合 (デフォルト)	戻される値 - 事前解析がオフの場合
echo %foo%	foo	Windows: %foo% UNIX または Linux: ブランク
echo \$foo	foo	Windows: \$foo UNIX または Linux: ブランク
echo \${foo}	foo	Windows: \${foo} UNIX または Linux: ブランク
echo \${foo}	空ストリング	Windows: \${foo} UNIX または Linux: システム・エラー

## 「環境変数」パネルについて

ご使用のプロジェクトの環境変数を作成したり管理したりするには「環境変数」パネルを使用します。パネルを表示するには、「環境変数」を選択します。



「環境変数」パネルでは、環境変数をクリックして、その変数を表示したり編集したりすることもできます。

## 「詳細」タブ

### 環境変数のプロパティ

次の環境変数のプロパティを指定します。

**名前** 環境変数の名前。

**アクセス**

この環境変数を使用できるユーザーのアクセス・グループ。

### 変数のプロパティ

「環境変数」パネルで、変数パネルに表示する変数を選択します。変数パネルを使用して、以下のプロパティを指定します。

**名前** 変数名には、英数字 (a-z、A-Z、0-9) と下線文字 ( \_ ) のみを使用できます。最大長は 255 バイトです。

- 1 バイト文字セットの場合: 255 文字
- 2 バイト文字セットの場合: 127 文字

**値** 変数値には、以下の特性があります。

- **長さ:** 値は任意の長さ (オペレーティング・システムで制限がある場合はその制限値まで) にできます。
- **特殊文字:** %, \$, [, ], {, }, ¥, ", および ' の各文字には、事前構文解析 (OS 評価前) 用とオペレーティング・システムでの評価用の特殊な意味があります。これらの文字は使用しないでください。これらを受け渡すには円記号 (¥) でエスケープします。330 ページの『ステップにおける変数の解釈』を参照してください。

値が単一引用符で囲まれている場合、変数名は解釈されるのではなく、リテラルとして渡されます。例えば、`MyEnv='$env'` という変数の代入では、変数 `$env` の値ではなく、ストリング `$env` が `MyEnv` に代入されます。

- **値の中の変数:** 変数に変数値内にある場合、その変数を使用するステップの実行時にその変数が解釈されます。
- **プルダウン値:** プルダウン変数に項目を指定するには、プルダウン・リストに変数タイプを設定し、変数を保存してから、編集します。「**プルダウン・オプション**」タブをクリックして、プルダウンに項目を追加します。336 ページの『変数のプルダウンの作成』を参照してください。
- **値としてのドット・コマンド:** ドット・コマンドの中には、環境変数の値として使用できるものがあります。そのような場合、システムはドット・コマンドを他の値で置き換えます。340 ページの『変数でのドット・コマンドの使用』を参照してください。
- **復帰:** 変数では、復帰は保管されません。次の例のように、1 つの変数に複数行の値を割り当てできます。ファイル `text.txt` の内容は変数 `test` に割り当てられています。

```
.bset env "test = `type text.txt`"
```

ファイル内では行は連結されます。例えば、ファイルの内容が以下であったとします。

```
A first line
And a second line
```

この変数の値は以下になります。

```
A first lineAnd a second line
```

**タイプ** 変数には、次のいずれかのタイプが割り当てられます。

- **標準** - デフォルトです。変数に値とアクションを割り当てることができます。
- **組み込み** - 変数値は別の環境変数を指し、その変数を組み込みます。環境変数内のすべての変数が組み込まれます。

注: 「組み込み」変数タイプは、これまでのリリースで提供されていた `.include` 機能に代わるものです。

- **プルダウン・リスト** - 変数には値セットが含まれ、ユーザーはこの中から選択できます。プルダウン・リスト変数が作成されたら、それを選択し、「プルダウン・オプション」タブをクリックして、値を追加できます。

## アクション

次のいずれかになります。

- **設定**: デフォルト・オプションです。指定された値が、変数に割り当てられます。変数が存在しない場合は作成されます。
- **設定 (未設定の場合)**: このアクションでは、変数にまだ値がない場合のみ、値を変数に割り当てます。 329 ページの『変数について』を参照してください。
- **付加**: 変数の現行値に値が付加されます。値の間には、OS 固有のパス区切り文字が追加されます。
  - Windows: セミコロン (;)
  - UNIX または Linux: コロン (:)
- **前に付加**: 現行値の前に値が挿入されます。値の間には、OS 固有のパス区切り文字が追加されます。
  - Windows: セミコロン (;)
  - UNIX または Linux: コロン (:)
- **クリア**: 値は空ストリングに設定されます。「値」プロパティに値が含まれる場合には、使用されません。
- **削除 / 設定解除**: 現在適用されている環境変数から変数が削除されます。「値」プロパティに値が含まれる場合には、使用されません。
- **非表示の割り当て**: 値はデータベースにおいて暗号化され、ログやユーザー・インターフェースでは「\*\*\*\*\*」のように表示されます。このオプションは、機密情報を保管する場合に使用します。

「非表示の割り当て」変数は、デフォルトでは編集できないように保護されています。「非表示タグの編集」権限が設定されていない場合は、値を編集できません。ただし、この権限が設定されていれば、値を削除したり

再作成したりできます。「非表示タグの編集」権限が設定されている場合は、値を編集して入力すれば、その値を使用できるようになります。この動作ではタイプが「設定」に変更されるため、「非表示の割り当て」変数の値を表示できなくなります。

環境変数の「アクション」値が「非表示の割り当て」である場合、変数の「タイプ」値および「アクション」値は変更できません。環境変数の「アクション」値が「非表示の割り当て」でない場合、「アクション」を「非表示の割り当て」に変更することはできません。

「非表示の割り当て」環境変数は、7.1.x エージェントでは機能しません。「非表示の割り当て」環境変数を使用したい場合は、ご使用のエージェントを対応するバージョンに更新してください。例えば、コンソール・バージョンが 8.0.0.1 である場合は、エージェント・バージョンを 8.0.0.1 に更新します。

注: 値を長くすれば、その値はよりセキュアになります。

注: 以前の「非表示の割り当て」変数がある場合は、Build Forge ユーザー・インターフェースの「環境変数」ページで、その変数を編集して保存し直し、暗号化できます。また、*bfassignhiddenmigration* ユーティリティーを使用すれば、バッチ・モードで暗号化を行うこともできます。Build Forge 8.0 インストール済み環境では、「非表示の割り当て」値は変更されません。Build Forge 7.x リリースからの古い「非表示の割り当て」変数値を保存し直したりアップグレードしたりしなかった場合は、新しい「非表示の割り当て」機能を使用できません。この動作は、7.x リリースで Perl エンジンを使用する場合と同じですが、「非表示の割り当て」値は新規 Java エンジン・ステップ・ログで平文として表されます。古い「非表示の割り当て」値を保存し直したり、*bfassignhiddenmigration* ユーティリティーを使用して、すべての「非表示の割り当て」値を暗号化したりすることをお勧めします。*bfassignhiddenmigration* ユーティリティーについて詳しくは、519 ページの『*bfassignhiddenmigration* リファレンス』トピックを参照してください。

注: ステップに含まれる変数の 1 つが「非表示の割り当て」に設定されると、そのステップに含まれる他のすべての変数がデフォルトで「非表示の割り当て」に設定されます。

注: 例えば、マルチバイト文字システム (MBCS) 文字 (中国語、日本語、または韓国語) を持つ「非表示の割り当て」変数を Build Forge 7.x で設定した場合、非表示マイグレーション・ユーティリティーは自動的にアップグレードを行いません。セキュリティを強化するために、この変数を ASCII 文字で置き換えることをお勧めします。これを行わないと、動作が Build Forge 7.x の場合と同じになるため、新規「非表示の割り当て」変数は使用できません。

### プロジェクト動作

ジョブを手動で開始したときの変数の使用方法を定義します。このプロパティーは、プロジェクトに割り当てられている環境変数内で使用される変数に

のみ有効です。このプロパティは、ジョブの実行中には変数に影響を与えません。値は、次のいずれかにできます。

- **通常:** プロジェクトへの割り当て時に、変数は通常どおり動作します。
- **必須:** 変数に値が存在する必要があります。このプロパティを持つ変数は、「開始」パネルで強調表示されます。変数定義に定義されている値で問題ありません。値を定義しないと、ジョブをクイック・スタートすることも、開始することもできません。

「必須」変数の含まれたジョブが、ユーザーではなくスケジューラーによって開始された場合、それらの変数は、現在値がある場合には変更されず、値がない場合にはブランクになります。

- **読み取り専用:** 値を変更することはできません。
- **表示の抑制:** 変数は「ジョブの開始」パネルに表示されません。しかし、変数は存在していて、ステップで使うことができます。
- **変更が必要:** 変数値を変更する必要があります。このプロパティを持つ最上位の変数は、「開始」パネルで強調表示されます。新規の値を入力しなければ、ジョブをクイック・スタートすることも、開始することもできません。

注: 「変更が必要」変数を含むジョブが、ユーザーではなくスケジューラーによって開始された場合、それらの変数値は変更されません。また、「変更が必要」変数を含むジョブがクラス・プロパティによって開始された場合も、それらの変数値は変更されません。クラス・プロパティには、「削除時に開始」、「設定時に開始」、「終了時に開始」などがあります。

## 「スナップショット」タブ

環境変数のスナップショットは、環境変数のインスタンスです。「スナップショット」タブには、スナップショットの名前が表示されます。スナップショット名やスナップショットに関するコメントを表示または変更する場合は、このタブを使用します。

---

## 環境変数の作成

### 手順

1. 左側のメニューで、「環境変数」をクリックします。
2. 「環境変数の追加」をクリックします。
3. 環境変数の名前を指定します。
4. この環境変数を使用できる必要のあるメンバーが属するアクセス・グループを指定します。
5. 「保存」をクリックします。 表示された変数パネルで、環境変数に変数を追加できます。

## 変数の使用

以下のセクションでは、変数を使用する共通のタスクを実効するための手順について説明します。

### 変数のプルダウンの作成 このタスクについて

変数には、複数の値を定義できます。指定した値は、プルダウンで選択可能なオプションとして表示されます。

環境変数用のプルダウンを作成するには、以下のようにします。


1. 「プロジェクト」 > 「環境変数」を選択します。
2. リストで、環境変数を選択します。
3. 「環境変数値の追加」をクリックします。プロパティには、次のように入力してください。
  - 名前: 環境変数の名前を入力します。
  - タイプ: 「プルダウン・リスト」を選択します。
  - アクション: アクションを選択します。
  - プロジェクト動作: プロパティを選択します。
4. 「保存」をクリックします。
5. 変数をクリックします。
6. 「プルダウン・オプション」タブをクリックします。
7. 次のようにプルダウンの値を追加します。
  - a. プルダウン・オプションの名前を指定します。これはユーザーには、プルダウンの選択項目名として表示されます。プルダウン名には、英数字 (a-z、A-Z、0-9) と下線文字 ( \_ ) のみを使用することができます。最大長は 255 バイトです。
    - 1 バイト文字セットの場合: 255 文字
    - 2 バイト文字セットの場合: 127 文字
  - b. プルダウン・オプションの値を指定します。ユーザーに、使用されている値を確認させたい場合は、変数名と同じ値を設定することができます。値で変数が使用されている場合は、ステップが実行されているオペレーティング・システムで解釈されます。ステップの実行前にこれが前処理されたり評価されたりすることはありません。
  - c. 「作成」をクリックします。オプションが、リストに追加されます。
  - d. 必要な値ごとに繰り返します。
8. 「変数の保存」をクリックします。
9. 表示されたデフォルト・オプションを設定します。この操作は、「プルダウン・オプション」に値を取り込んで、変数を保存した後で実行します。環境変数をクリックします。「詳細」タブで、「デフォルト・オプション」プロパティを、--NONE-- からデフォルトで表示したい値の名前に変更します。



変数の「詳細」タブを表示するときに、「デフォルト・オプション」が設定されていない場合 (その値が `--NONE--` の場合)、「プルダウン・オプション」リストの最初の項目がデフォルトで「値」列に表示されることに注意してください。

10. 「変数の保存」をクリックします。

次のようにさらにオプションを処理することができます。

- 各オプション名の左にある「編集」アイコン  を使用してオプションをリストに配置するか、オプションを削除します。
- 編集するには、オプションをクリックします。「名前」フィールドおよび「値」フィールドの両方を編集できます。編集が終了したら、「保存」をクリックします。
- 「名前」フィールドおよび「値」フィールドをクリアするには、「クリア」をクリックします。通常、新規オプションを作成するには、既存のオプションを表示してからこの操作を実行します。

## 他の環境変数の組み込み

「組み込み」変数タイプを使用して、既存の他の環境からすべての変数を組み込むことができます。

1. 「プロジェクト」 > 「環境変数」を選択します。
2. 新規の環境を作成し、「保存」をクリックします。
3. 環境の名前をクリックします。
4. 「環境変数値の追加」をクリックします。プロパティには、次のように入力してください。
  - タイプ: 「組み込み」を選択します。ユーザー・インターフェースが、「環境変数の組み込み」プルダウンを表示するように変更されます。「アクション」プロパティおよび「プロジェクト動作」プロパティが除去されます。
  - 環境変数の組み込み: リストから組み込む環境変数を選択します。
5. 「変数の保存」をクリックします。

## ステップ実行時の変数値の変更

ステップ、プロジェクト、または永続のスコープを使用して、変数を実行時に変更することができます。

- ステップ・スコープ: ステップ内でコマンドを使用する際、明示的に代入することで、変数値を指定変更することができます。それらの値は、現行ステップの間だけ有効になります。
- プロジェクト・スコープ: ステップ内で `.bset` コマンドを使用することにより、実行中のジョブをスコープとして変数値が変更されます。`.bset` を使用して、新規の変数を作成できます。それらのスコープは、ジョブのそれ以降の部分です。`.bset` を使用して行った変更は、`.bset` を使用したステップの後のステップで有効になります。
- 永続スコープ: ステップ内で `.set` コマンドを使用することにより、環境変数定義が変更されます。変数は、サーバー環境、プロジェクト環境、およびステップ環境で定義されます。`.set` を使用してサーバー変数またはプロジェクト変数を変更



しても、その変数の現行ジョブのコピーは変更されません。現行ジョブの後に実行されるジョブが、変更された変数を取得します。しかし、`.set` コマンドがステップ環境変数を変更した場合は、そのステップ環境変数を使用する後続のステップが、変更された変数を取得します。`.set` コマンドで新規の変数を作成することはできません。

例えば、変数 `JavaVersion = 1.4` が含まれている `Java` という名前のプロジェクト環境変数を使用するプロジェクトを起動し、`.bset` を使用してその値を `1.5` に変更したとします。この場合、そのプロジェクト環境変数を継承するすべてのステップは値 `1.5` を取得し、`Java` 環境変数を限定して参照するステップは元の値 `1.4` を取得します。

システムは、ジョブを開始するときに、そのジョブのために別に用意されたデータベース・レコードにプロジェクト環境変数をコピーし、その後は、プロジェクトのデフォルト値を取得するときにこのジョブ環境変数を参照します。ユーザーが、そのジョブを開始するときに何らかのプロジェクト変数の開始値を変更すると、それらの値がジョブ・レコードに記録されます。

## Windows ドライブのマッピング

Microsoft Windows<sup>®</sup> オペレーティング・システムでは、マップされたドライブを異なる方法で管理します。エージェントは、ユーザー・アカウントについて記憶している接続を再マップしようとしませんが、実行時にマッピングを正常に完了することができません。Windows でのドライブ・マッピングを支援するため、特別な環境変数である `_MAP` 変数を使用することができます。この変数を設定すると、Windows Agent は、ステップを実行する前に、ドライブをマップします。

`_MAP` 変数を使用するときの標準的な対処法は、プロジェクト環境変数でその変数を割り当てることで、同じドライブ・マッピングが、環境変数の継承によってすべてのステップ環境変数まで引き渡されるようにします。ステップ環境変数でも `_MAP` 変数を定義した場合、そのステップ環境変数の値によって、プロジェクトの環境変数がオーバーライドされます。これは、特定ステップについて 1 つの `_MAP` 値のみ定義できるからです。

これは Windows 環境での規則ですが、`_MAP` 変数でディレクトリー・パス名を区切るにはスラッシュを使用してください。パスが使用されるとき、エージェントは、必要な場合、自動的にスラッシュを訂正します。

例えば、`_MAP` を

```
X:="//server/share
```

に設定すると、X: ドライブを Windows UNC パス名 `¥¥server¥¥share` に接続する、ランタイム・マッピングが定義されます。

`MAP` 変数で追加のマッピング指定を (区切りにはセミコロンを使用して) 提供することで、複数のドライブをマップすることができます。

```
X:="//server/share;Y:="//server/share2
```

デフォルトでは、Windows 上のドライブ・マッピングは、論理サーバーで定義されているのと同じユーザー名とパスワードを使用して実行されます。異なるユーザー名のドライブをマップするには、次の例のように、ユーザー名とパスワードを括弧で囲んでマッピングの後に追加します。

```
X:=//server/share(alternateusername,password)
```

注: パスワードに \$ 文字を使用している場合は、もう 1 つ \$ 文字を使用してエスケープします。例: パスワード pas\$word を pas\$\$word として入力してください。パスワードに特殊文字 %, [, ], {, }, ", または ' を使用することは避けてください。

\_MAP 変数を使ってマップされたドライブは、コマンド完了時にマップ解除されます。

Windows 上でのドライブ・マッピングは、たとえ正常にマップされた場合でも、システムのコンソールにログオンしているユーザーがそのドライブを使用または共有していれば、アクセスできないことがあります。

## 次に使用可能なドライブ名を使用した Windows ドライブのマッピング

次に使用可能なドライブ名を、システムに選択させることができます。以下の構文を使用します。

```
<driveletter>?=//<directory path>
```

例えば、以下のようにして \_MAP を設定できます。

```
X?=//server/share
```

この場合、システムは、ドライブを X にマップ しません。代わりに、次に使用可能なドライブにマップし、\_MAP\_X という名前の変数に、選択したドライブ名を格納します。選択されたドライブが F の場合は、\_MAP\_X 変数の値は「F:」です。この変数を使用して、マッピングにアクセスできます。

次のように複数のマッピングを使用できます。

```
X?=//server/share; Y?=//server/public
```

希望する任意の文字を使用できるほか、以下の例のように、複数のマッピングを使用することも可能です。

- F: から //server/share へのマップ
- G: から //server/public へのマップ

この例では、以下の変数も作成されます。

- 値「F:」を持つ \_MAP\_X
- 値「G:」を持つ \_MAP\_Y

注: Cygwin を使用する Windows® システムをターゲットとするときに、次に使用可能なドライブの構文を使用する場合、以下のように、疑問符 (?) を円記号でエスケープする必要があります。

```
Y¥?=//server/share
```

## エージェント・ベースのドライブ・マッピング

ドライブは、エージェント内の構成パラメーターを使用してマップすることができます。`map` パラメーターは、`BFAgent.conf` ファイルに追加された場合、`_MAP` 変数の構文と同一の構文を使用します。このパラメーターを使用して、特定のサーバーのドライブ・マッピングを作成できます。`_MAP` 変数を使用した場合も、そのマッピングは、エージェント・マッピングをオーバーライドします。

## 変数でのドット・コマンドの使用

ドット・コマンドの中には、環境変数の名前または値として使用できるものがあります。

### `.source` によってコマンドの前にスクリプトを実行する

システムは、`.source` という名前の特別な環境変数を定義することにより、コマンドを実行する前にサーバーでスクリプトを実行する機能を提供します。これを使用することにより、サーバー上のソース・ファイルから環境変数のセットをロードしたり、カスタム準備コマンドを実行することができます。

この機能を試すには、次のようにしてください。

1. システムで、なんらかのセンテンスをエコーする `mybatch.bat` という名前のバッチ・ファイルを作成します。そのバッチ・ファイルを `C:%temp` に保存します。
2. `Step Variables` という名前の新しい環境変数を作成します。
3. 値が `C:%temp%mybatch.bat` の、`.source` という名前の変数を追加します。
4. ステップの環境変数を、新しく作成した `Step Variables` 環境変数に設定します。
5. プロジェクトを実行し、ログ出力を調べてそのステップを検査します。

`mybatch.bat` ファイルがステップのコマンドの前に実行されたことを示す、追加ログ・データを確認してください。次に、`.source` に関するいくつかの重要な注意事項を示します。

- 指定されたパスには引数を含めることができません。
- Windows® プラットフォームでは、このスクリプトは呼び出しを介して起動されます。
- UNIX® プラットフォームでは、このスクリプトは実行中のシェル内で入手されるため、ネイティブのシェルの構文規則に従っていなければなりません。

### `.date` コマンドで日付または時間を変数に保管する

`.date` 環境変数のドット・コマンドを使用して、現在の日付または時間によって、変数を指定できます。変数の値としてコマンドを使用します。変数を使用するプロジェクトの実行時に、システムは、`.date` コマンドの結果を変数に反映して更新します。

例えば、値 `.date %B` を持つ `MONTH` という名前の変数は、プロジェクトに含まれるときに、値「May」によって、5 月中のジョブに指定されます。

日付形式のコードのリストなど、このコマンドの使用について詳しくは、438 ページの『`.date`』を参照してください。

## システム変数のリファレンス

変数では、システム定義変数を使用することができます。

システムは、ジョブの各ステップで以下の変数の値を自動的に設定します。これらの変数は読み取り専用です。ジョブに対してのそれらの値は、ステップ・ログの ENV 行にリストされています。最初の 4 つはプロジェクト・レベルの通知です。その他のすべての BF\_ 変数は、ステップ・レベルで使用されます。

プロジェクト・レベルの変数	値
BF_D	日付。 タグで使用できます。フォーマットは、「タグ: 日付書式」システム設定で決定されます。
BF_J	通算で表す年の日。 タグで使用できます。
BF_T	時間。タグで使用できます。フォーマットは、Tag: Time Format システム設定で決定されます。
BF_W	曜日。0 (日曜) から 6 (土曜) までの数値で表されます。

ステップ・レベルの変数	値
BF_D	日付。 タグで使用できます。フォーマットは、「タグ: 日付書式」システム設定で決定されます。
BF_J	通算で表す年の日。 タグで使用できます。
BF_T	時間。タグで使用できます。フォーマットは、Tag: Time Format システム設定で決定されます。
BF_W	曜日。0 (日曜) から 6 (土曜) までの数値で表されます。
BF_AGENT_PLATFORM	エージェントを実行中のオペレーティング・システム・プラットフォームを識別するストリング。
BF_AGENT_VERSION	現行サーバーのエージェントのバージョン番号。
B	デフォルトのタグ変数。1 から始まり、ジョブごとに増分します。 BF_TAG で表されるタグの中で使用できます。
BF_BID	同一プロジェクトのジョブに対する固有のジョブ ID 番号。
BF_CALLER_	呼び出し側プロジェクトから、チェーニングされたプロジェクトに渡される変数に適用される接頭部。
BF_CLASS	プロジェクトの Build Forge クラス。

ステップ・レベルの変数	値
BF_ENGINE	エンジンを一意的に識別するストリング。この値は、インストール・ディレクトリーに格納されているファイル <code>engine.id</code> にも保管されます。例: D8531015-6C07-1014-8CA0-BD58317220B3。
BF_HOST	論理サーバーのホスト名 (TCP/IP ホスト名)。(この変数はサーバー環境変数の一部です。)
BF_ITERATION	While ループ内のステップが正常に開始された回数。これは、ステップの条件の評価結果が TRUE である場合に増分されます。ジョブの再開時には、再開すべき反復として、この変数の値が使用されます。
BF_ITERATION_MAX	While ループを実行可能な最大回数。これはステップのプロパティーで設定されます。
BF_ITERATION_MAX_REACHED	デフォルトでは作成または設定されません。While ループの反復回数が BF_ITERATION_MAX に達すると、この変数に While ループ・ステップのステップ固有の ID (BF_SSID) が付加されます。1 つのプロジェクト内の複数の While ループ・ステップがその BF_ITERATION_MAX に達した場合、この変数には、複数の値 (その最大反復回数に達したステップごとに 1 つずつ) が含まれます。
BF_LASTGOODRUN	同一プロジェクトの最終引き渡しジョブの日付。引き渡しジョブが存在しない場合は最終ジョブの日付。
BF_LASTGOODTAG	最後の引き渡しジョブ (または、同一プロジェクトの保管済み引き渡しジョブがない場合は、最後のジョブ) のタグ。
BF_LASTGOODUNIX	BF_LASTGOODRUN と同じ。ただし、日付は UNIX <sup>®</sup> フォーマットで表します。
BF_LASTRUN	現行ジョブの前回の実行日付。
BF_LASTTAG	同一プロジェクトの前回のジョブのタグ・ストリング。
BF_LASTUNIX	BF_LASTRUN と同じ。ただし、日付は UNIX フォーマットで表します。
BF_ONFAIL	ステップの停止/継続フラグ。
BF_PID	プロジェクト ID 番号。
BF_PROJECTNAME	このジョブのプロジェクトの名前。

ステップ・レベルの変数	値
BF_PROJECTNAME_PHYS	プロジェクト・ディレクトリーを作成するときに使用されるプロジェクト名。システムは、プロジェクト・ディレクトリーを作成するため、「無効な相対ディレクトリー文字」システム設定で指定された文字を下線文字に変更します。例えば、設定値にスペースが含まれる場合、「My Project」という名前のプロジェクトは、「My_Project」という名前のプロジェクト・ディレクトリーを受け取ります。
BF_ROOT	ジョブのプロパティーから取得された、ジョブの基本作業ディレクトリー。 BF_STEP_ROOT も参照してください。
BF_SERVER	現行ジョブを実行しているサーバーの名前 (この変数はサーバー環境変数の一部です)。
BF_SERVER_ROOT	サーバーのプロパティー内で論理サーバーに割り当てられたパス (この変数はサーバー環境変数の一部です)。
BF_SID	ビルド内のステップ結果のシーケンス番号。
BF_SPID	現行ジョブが別のジョブによって呼び出された場合、その呼び出し側プロジェクトの ID が含まれています。呼び出されていない場合、この値は BF_PID と同じです。
BF_STEPNAME	ステップ名。ステップのプロパティーで設定されます。
BF_STEP_ROOT	ステップのプロパティーから取得された、ステップの基本作業ディレクトリー。 BF_ROOT も参照してください。
BF_SSID	ステップ ID。プロジェクトの現行ステップに対する固有 ID です。
BF_TAG	ジョブのタグ。タグ定義には、変数を含めることができます。この変数には、ジョブの開始時にそれらの変数を解釈した結果として得られた値が含まれています。
BF_TAG_PHYS	BF_TAG 値内のすべてのスペースを下線で置き換えた、ジョブのタグ。ステップで「絶対」オプションが選択されている場合、BF_TAG_PHYS は BF_TAG と同じになります。
BF_USER	ジョブ所有者のユーザー名。

## トリガー変数のリファレンス

システムは、以下の変数名を監視します。ステップの環境変数が、これらの変数の 1 つを (具体的に指定して、あるいはプロジェクトまたはサーバーから継承して) 含む場合に、アクションが実行されます。

変数	内容
_CI_BUILD_DELETE	任意の値にこの変数を設定して、ジョブ実行後のビルドおよび関連のビルド・データを削除する。(その他のプロジェクト・ビルドが実行されない場合、タグ変数は、ビルドの削除の前に、その初期値にリセットされます。)
_CI_BUILD_KEEP	任意の値にこの変数を設定して、ジョブ実行後のビルドおよび関連のビルド・データを保持する。例えば、ジョブがアダプター・リンクを含み、そのアダプター・ステップが失敗した場合、その他のプロジェクト・ステップは実行されません。ジョブのビルド・レコードのコピーを、デバッグなどのため、保持してください。
CLEARCASE_VIEW	指定された ClearCase ビューを開始する。この変数で指定されるビューが存在しており、この変数を使用するステップが「絶対」に設定されている必要があります。Microsoft Windows が稼働しているシステムでは、bfagent.conf 内のエージェントの cc_suppress_server_root 構成オプションで、この変数を使用する必要があります。
_CLEARCASE_VIEWS	コマンドを実行する前に開始する ClearCase ビューのリストを指定する。ビューのコンマ区切りのリストに値を設定します (「View1,View2,View3」など)。
_CLEARCASE_VOBS	コマンドを実行する前にマウントする ClearCase VOB のリストを指定する。VOB のコンマ区切りのリストに値を設定します (「¥Vob1,¥Vob2,¥Vob3」など)。



変数	内容
_CONTEXT_LOG_RANGE	<p>この変数を使用して、ログ出力を、フィルターの突き合わせが行われた周辺の行に限定する。これは正整数の値をとり、この変数によってシステムは、各フィルター・ストリングでヒットした行の周辺にある行 (その範囲のサイズはこの変数の値に等しい) 以外のログ出力を省略します。例えば、変数に 5 を設定した場合、ログは、フィルターの突き合わせが行われた行と、その行の前後にある各 5 行のみを表示します。</p> <p><b>注:</b> この変数は、Perl エンジン・プロジェクトにのみ使用されます。</p>

変数	内容
<u>ERROR_THRESHOLD</u>	<p>エラー (定義した「失敗の設定」フィルターでキャッチされたもの) の最大許容数を設定する。この変数を使用すると、個々のステップまたはプロジェクトに対して、失敗およびメッセージのしきい値を設定することができます。</p> <p>次のいずれかの形式を使用します。</p> <ul style="list-style-type: none"> <li>• 値 5 または F5 は、5 個を超えるエラーが発生した場合にそのジョブが失敗となることを示します。</li> <li>• 値 N7 は、7 個を超えるエラーが発生した場合に、システムによりジョブの注釈にメッセージが追加されることを示します。そのメッセージとは、このしきい値に達したことを示すものです。</li> </ul> <p>ステップでこの変数を使用すると、システムは、個々のステップのエラーをカウントします。以下の追加フォームを使用できます。</p> <ul style="list-style-type: none"> <li>• W9 のような値は、9 つのエラーの後、フィルターでキャッチされるその後のエラーとは無関係に、ステップは警告状態となることを意味します。</li> <li>• C8 のような値は、8 つのエラーの後、ステップは失敗状況に設定されますが、任意の「失敗のクリア」フィルターで失敗をクリアできます。</li> </ul> <p>注: この変数でカウントされたエラーは、「失敗の設定」アクション付きのフィルターに一致するストリングとして定義され、プロジェクトのステップに割り当てられます。フィルターによって失敗として識別された各ストリングは、ステップ合計に対する 1 つのエラー、およびプロジェクトの合計に対する 1 つのエラーとしてカウントされます。</p>
<u>EXITCODE_MAP</u>	<p>システムが、ステップ成功の指標として受け入れる必要がある (コンマ、スペース、セミコロン、またはコロンで区切られた) 番号のリストを指定する。デフォルトでは、0 の終了コードは成功を示します。この変数が指定されると、この変数にリストされた値はすべて、成功を意味します。</p>

変数	内容
<code>_InterfaceLoggingLevel</code>	<p>アダプター・ステップの実行時に、Build Forge がログに記録するログ・データの数を制御する。<code>_InterfaceLoggingLevel</code> という名前で、環境変数を (アダプター環境変数に) 作成します。それに、0 から 8 の範囲にある整数値を割り当てます。ロギング・レベルは包括的で、例えば、レベル 2 は、レベル 1 および 0 の情報を含みます。</p> <ul style="list-style-type: none"> <li>• 0: 実行行およびサーバー接続エラー、またはキャンセル通知。他はなし。</li> <li>• 1: 構文解析済みのコマンド (サーバーに送信されるコマンド)</li> <li>• 2: 構文解析済みでないコマンド (ローカル変数が設定される前のコマンド)</li> <li>• 3: ビルドおよび環境変数の SET 行</li> <li>• 4: 一時および内部変数 SET 行</li> <li>• 5: 環境変数の評価、E メール・グループの追加、BOM テキスト・ロギングの行</li> <li>• 6: ブロックおよびサブブロックの開始/終了行</li> <li>• 7: (デフォルトのロギング・レベル) 一致パターンに対するチェックが行われるエージェント出力、およびパターンに一致した行</li> <li>• 8: すべてのエージェント出力</li> </ul>
<code>_LOG</code>	<p>Build Forge エージェントのロー出力が含まれるログ・ファイルを作成するパス名を指定します。</p> <p><b>注:</b> このログには、<code>_LOG_TIMESTAMP</code> がともに指定されていない限り、タイム・スタンプが含まれません。このファイル内のログ・データは通常、エージェント・コード、ログ・バケット、およびメッセージのようにフォーマット設定されています。</p> <p>この変数を使用して、ジョブ・ログのコピーをサーバーに保存します。ファイルが存在している場合、システムはそのファイルに追加します。</p>

変数	内容
_LOG_TIMESTAMP	<p>_LOG からの出力の各行の先頭にタイム・スタンプを付加する。この変数の値は、.date および .gmdate 環境変数コマンドにより使用される strftime 構文と同じ書式制御ストリングである必要があります。</p> <p>注: _LOG とともに指定する必要があります。</p>
_MAP	<p>この変数の使用法に関する考察については、338 ページの『Windows ドライブのマッピング』を参照してください。</p>
_NO_PREPARSE_COMMAND	<p>通常、システムは、コマンドをエージェントに送信する前に、環境変数の値を解決しようとする。_NO_PREPARSE_COMMAND 変数が (任意の値によって) 定義されると、システムは、変数を解決せずにそれをエージェントに送信します。この変数を使用すると、オペレーティング・システム・シェルで変数を操作できるようにします。</p>
_PRISM_DIR_POSTCMD	<p>IDE のプラグインで使用される。プロジェクト・ステップが実行された後でディレクトリーに対して実行される、コマンドを指定します。610 ページの『テスト・プロジェクト用の特殊変数』を参照してください。</p>
_PRISM_DIR_PRECMD	<p>IDE のプラグインで使用される。ディレクトリーがプロジェクト・ステップ用のサーバーにコピーされる前にそのディレクトリーに対して実行される、コマンドを指定します。610 ページの『テスト・プロジェクト用の特殊変数』を参照してください。</p>
_PRISM_FILE_POSTCMD	<p>IDE のプラグインで使用される。プロジェクト・ステップが実行された後でファイルに対して実行される、コマンドを指定します。610 ページの『テスト・プロジェクト用の特殊変数』を参照してください。</p>
_PRISM_FILE_PRECMD	<p>IDE のプラグインで使用される。ファイルがプロジェクト・ステップ用のサーバーにコピーされる前にそのファイルに対して実行される、コマンドを指定します。610 ページの『テスト・プロジェクト用の特殊変数』を参照してください。</p>

変数	内容
<code>_SUPPRESS_ENV_OUTPUT</code>	<p>システムがログに環境変数メッセージを出力しないように指定する。デフォルトではこの変数は設定されておらず、ステップ・コマンドの実行前に、環境変数のすべての変数値が出力されます。値は、ステップ・ログに ENV 項目として表示されます。変数に設定できる値は次のとおりです。</p> <ul style="list-style-type: none"> <li>• ALWAYS: ENV メッセージを常に省略します</li> <li>• その他の任意の値: ENV メッセージを省略します。ただし、コマンドが失敗した場合は、コマンド・メッセージの後に ENV メッセージが出力されます。この情報は、コマンド実行障害をデバッグする際に役立つことがあります。</li> </ul>
<code>_SUPPRESS_AGENT_LOG_OUTPUT</code>	<p>1 に設定すると、エージェントはログ・データをエンジンに送信できません。  <code>_SUPPRESS_LOG_OUTPUT</code> と比較してください。こちらの場合、ログ・データはエージェントから送信されますが、エンジンによって除去されます。  <b>注:</b> この変数を使用した場合、フィルターの突き合わせは実行されません。</p>
<code>_SUPPRESS_LOG_OUTPUT</code>	<p>この変数を任意の値に設定すると、エージェントから受信されたログ出力のほとんどすべてがエンジンによって除去されます。一部のコンソール・ログ・メッセージは残ります。  「フィルタリングする突き合わせ」が示されます。</p>
<code>_TIMEOUT</code>	<p>プロジェクトにある 1 つまたはすべてのステップのタイムアウト・プロパティを指定変更する値。</p>
<code>_TRAP</code>	<p>現在のステップが失敗した場合に実行されるストリング。このストリングには、実行可能ファイルまたはコマンドの名前を設定できます。<b>注:</b> コマンドの出力はコンソールには戻されません。これは、ステップが失敗すると、コンソールとエージェントの間の接続が閉じるためです。<code>_TRAP</code> を介して発行されたコマンドからの出力を保持したい場合は、後で取得できるように、コマンドを使って、その出力をファイルに書き出します。</p>

変数	内容
_USE_BFCREDS	<p>これが 1 に設定されると、システムは、サーバーに付与されたサーバー権限に保管された資格情報ではなく、ユーザーの ログイン資格情報を使用して、サーバーにログインする。システムは、プロジェクトを開始して、そのプロジェクトでコマンドを実行したユーザーの、管理コンソールのログイン資格情報を使用します。この変数を単一ステップに対して、またはプロジェクト全体に対して設定することができます。</p> <p><b>注:</b> LDAP/アクティブ・ディレクトリー認証を使用している場合、_USE_BFCREDS 関数を機能させるためには「<b>ユーザー認証のローカル保管</b>」システム設定を Yes (デフォルト値) に設定する必要があります。この設定が「はい」の場合、システムは、ユーザー認証情報を暗号化された形式でキャッシュに入れます。これにより、_USE_BFCREDS で使用するためにユーザー認証情報にアクセスできるようになります。</p> <p><b>ヒント:</b> Windows の場合、変数 _USE_BFCREDS_DOMAIN の設定も検討してください。</p>
_USE_BFCREDS_DOMAIN (Windows のみ)	<p>1 に設定すると、システムは、_USE_BFCREDS がサーバーへのログインに使用するログイン資格情報のほかに、ユーザーの ドメインを使用します。</p>

変数	内容
<code>_XSTREAM_PROTOCOL type</code>	<p>エージェント間の直接ファイル転送を有効にします。</p> <p><b>重要:</b> いくつかのオペレーティング・システム上のエージェントでは、直接ファイル転送のサポートが制限されているか、または直接ファイル転送がサポートされていません。</p> <p>193 ページの『エージェント間の直接ファイル転送を構成する』を参照してください。</p> <p>エンジン、送信側エージェント、および受信側エージェントのすべてで、直接ファイル転送がサポートされていることが必要です。いずれかでサポートされていないと、<code>_XSTREAM_PROTOCOL</code> は警告なしで無視され、通常のファイル転送方式が使用されます。</p> <p>受信側エージェントは、送信側エージェント・ホストで TCP 接続を作成可能であることが必要です。接続が存在する場合は、ファイアウォールが接続を許可するように構成されていなければなりません。</p> <p>プロトコル・タイプによって、データのエンコード方式が決まります。以下のタイプがあります。</p> <p><b>AES-CBC</b></p> <p>強力な暗号アルゴリズムがデータのエンコードに使用されます。エージェントはどちらも、OpenSSL でコンパイルされ、エンジンとの通信に SSL を使用する必要があります。暗号鍵はエンジンから得られます。</p> <p><b>PRNG</b> 疑似乱数生成プログラムがファイル内容の隠蔽に使用されます。</p> <p><b>PLAIN</b> ファイルはエンコードなしでそのまま転送されます。</p>

## 環境変数のスナップショット

変更する環境変数の新規インスタンスを手早く作成するには、環境変数のスナップショットを作成します。

### 環境変数のスナップショットの概説

環境変数のスナップショットについて学習し、それらの使用方法について理解するには、以下のトピックを参照してください。



## 環境変数のスナップショットの使用事例

環境変数のスナップショットを使用する一般的な事例には、次のようなものがあります。

- 環境変数の構成に変更を行う場合や、新しいツールまたはスクリプトのテストを実行する場合に環境変数のスナップショットを作成します。
- 一時的なバックアップとして、または正式なアーカイブの一部として、環境変数のスナップショットを保管します。
- 環境変数のスナップショットを作成して、外部リリースや内部リリースなどのマイルストーンに対応するポイント・イン・タイム環境変数構成を取り込みます。

## 環境変数のスナップショットの概念および用語

スナップショットに関して、環境変数を操作するための新しい概念および用語がいくつか UI の中で使用されています。

**環境変数のスナップショット:** スナップショットは、既存環境変数の新規インスタンスです。スナップショットに関して注意すべきいくつかのキーポイントを次に示します。

- スナップショットは、独立した環境変数オブジェクトです。スナップショット・セット内の 1 つのスナップショットを変更しても、そのセット内の他のスナップショットは変更の影響を受けません。
- スナップショットはコピーではありません。

環境変数と関連付けられたオブジェクトのスナップショットを作成した場合、そのオブジェクトとは別個のインスタンスが作成されます。コピーではオブジェクト間の関係がマップされますが、オブジェクトは新規作成されません。

- スナップショットは、環境変数の改訂ではありません。
  - スナップショットは、2 つの環境変数スナップショット間での変更の比較をサポートしていません。
  - 環境変数スナップショットに対する変更は、ソース制御システムの場合とは異なり、追跡されたり、バージョン番号によって識別されたりすることはありません。しかし、バージョン番号 (7.5.0, 3.4.01 など) を含むスナップショット命名体系を使用することによって、環境変数スナップショットとマイルストーンを相互に関連付けることができます。

**スナップショット・セット:** スナップショット・セットは、ある基本スナップショットの子孫であるすべての環境変数スナップショットのセットです。このセットには少なくとも、基本スナップショットまたは親スナップショットと、子スナップショットが含まれます。UI で、環境変数名の横にある「スナップショット」アイコン




は、その環境変数にスナップショット・セットが存在することを示します。


**基本スナップショット:** 最初、すべての環境変数には Base Snapshot というスナップショット名が付いています。Base Snapshot は別の名前に変更できます。基本スナップショットは、スナップショット・セットの親です。

**デフォルトの環境変数スナップショット (Default environment snapshot):** デフォルトの環境変数スナップショットは、現行の作業環境変数です。セット内の 1 つのス

ナップショットのみをデフォルトにすることができます。デフォルト・スナップショットを指定しなかった場合、基本スナップショットがデフォルトになります。

- UI で、デフォルト・スナップショットは、環境変数リストの最上位に表示されます。「環境変数」を選択すると、環境変数リストが表示されます。
- スナップショットを持つ環境変数を選択した場合、リスト・ボックスで別の環境変数スナップショットを選択しない限り、デフォルトの環境変数スナップショットが使用されます。
- 環境変数スナップショット・セット内の他のスナップショットにアクセスして操作するには、「スナップショット」アイコン  をクリックする必要があります。

## 環境変数のスナップショットのビュー

「スナップショット」ビューを表示するには、「スナップショット」アイコン  を選択します。UI の「スナップショット」ビューには、セット内のスナップショットの階層が次のように表示されます。

- 基本スナップショットが最上位にあります。固有の名前を割り当てていない場合、これには Base Snapshot という名前が付いています。
- すべての環境変数スナップショットは、基本スナップショットの子です。同じ基本スナップショットの子は、「スナップショット」列の同じレベルにインデントされています。
- 子スナップショットから作成された環境変数スナップショットは、その子スナップショットの子になり、「スナップショット」列の次のレベルにインデントされます。

## 環境変数のスナップショットの計画

デフォルトの環境変数スナップショットの選択、および環境変数スナップショットの命名に関するいくつかのベスト・プラクティスを紹介します。

### • セット内のデフォルト・スナップショットの選択方法

UI では、1 つのスナップショット・セットに対して、デフォルトまたは現行の環境変数スナップショットを 1 つだけ認識します。次のような一貫性のある方法で、デフォルト・スナップショットを選択してください。

- 基本スナップショットをデフォルト・スナップショットとして使用する

この方法では、ポイント・イン・タイム・バックアップとしてスナップショットを作成し、バックアップされた環境変数スナップショットに変更は行いません。基本スナップショットに変更を行います。

- 最新のスナップショットをデフォルト・スナップショットとして使用する

この方法では、新規の環境変数を作成する際に、それを新規のデフォルト環境変数スナップショットにします。基本スナップショットまたは以前の環境変数スナップショットに変更は行いません。

### • セットのスナップショット命名体系を識別する

環境変数スナップショット名は、環境変数スナップショット・セット内で固有の名前にする必要があります。

環境変数スナップショット名を作成する際には、次の基準が参考になります。

- 説明的な名前にし、スナップショットの使用法や目的が分かるようにします。
- 命名体系が、定義された標準に従うようにします。「スナップショット」タブの「コメント」ボックスを使用して、命名体系を説明することができます。

#### • セットに対する単一の環境変数名の使用

環境変数のスナップショットの作成後、その環境変数の名前を変更することができます。環境変数名を変更すると、すべての環境変数スナップショットでその名前が更新されます。

## 環境変数のスナップショットの作成

環境変数のスナップショットを作成すると、環境変数の新規インスタンスが作成されます。スナップショットはコピーではありません。新規の環境変数です。

環境変数のスナップショットは、環境変数または環境変数のスナップショットから作成できます。


**ヒント:** スナップショットを作成し、デフォルト・スナップショットを変更するには、ユーザーは必要な権限を保持している必要があります。398 ページの『スナップショット権限のためのアクセス・グループの検証および編集』を参照してください。


1. スナップショットを作成する環境変数または環境変数のスナップショットの横にある「編集」アイコンをクリックします。

- デフォルトの環境変数スナップショットのスナップショットを作成するには、環境変数のリスト（「環境変数」）で、最上位のスナップショットの横にある

「編集」アイコン  をクリックします。

- デフォルト以外の環境変数スナップショットのスナップショットを作成するに

は、「スナップショット」アイコン  をクリックします。「スナップショット」ビューにセット内の環境変数スナップショットが表示されます。デフォ

ルト以外の環境変数スナップショットの横にある「編集」アイコン  をクリックします。

2. 「新規スナップショットの作成」をクリックします。
3. 「スナップショット」タブの「名前」に名前を入力します。この名前は、環境変数とともにスナップショットを作成するすべてのオブジェクトに割り当てられます。

これは、環境変数のスナップショットのセット内で固有の名前にする必要があります。

4. 環境変数のスナップショットを作成する際に、スナップショットを作成する Build Forge オブジェクトを選択します。選択可能なオブジェクトについては、下の表に説明があります。

オブジェクト	説明
デフォルト	UI で、デフォルト・スナップショットは、環境変数リストの最上位に表示されます。  「環境変数」を選択すると、環境変数リストが表示されます。
含まれる環境変数に従う	環境変数が「組み込み」環境変数型を使用して組み込む環境変数のスナップショットを作成します。

5. 「保存」をクリックして、環境変数のスナップショットを保存します。

## デフォルトの環境変数スナップショットの変更

デフォルトの環境変数スナップショットは、スナップショット・セットの最上位スナップショットであり、環境変数リスト（「環境変数」）に表示されます。

**ヒント:** スナップショットを作成し、デフォルト・スナップショットを変更するには、ユーザーは必要な権限を保持している必要があります。398 ページの『スナップショット権限のためのアクセス・グループの検証および編集』を参照してください。

デフォルトの環境変数スナップショットを変更するには、新しいデフォルトにするスナップショットのスナップショット定義を次のように編集します。

1. 「環境変数」を選択します。
2. 環境変数リストで、デフォルトの環境変数スナップショットの「スナップショット」アイコンをクリックします。
3. スナップショットのリストで、新しいデフォルトにする環境変数スナップショットの「編集」アイコンをクリックします。
4. 「デフォルトの設定」をクリックします。
5. **重要:** ポップアップで「OK」または「取り消し」を選択します。



OK	参照を更新します。以前のデフォルトを参照するすべてのオブジェクトについて、以前のデフォルト環境変数スナップショットから新しいデフォルトへ参照を更新します。
取り消し	参照を更新しません。以前のデフォルトを参照するすべてのオブジェクトについて、新しいデフォルト環境変数スナップショットへ参照を更新しません。

## 環境変数のスナップショット名の変更

環境変数のスナップショット名、およびその環境変数のスナップショットを作成する際にスナップショットを作成するように選択したオブジェクトのスナップショット名は、変更することができます。

基本スナップショットについては、このオプションを使用して、その基本スナップショットのデフォルト名を別のスナップショット名に変更することができます。その際、単一の環境変数スナップショットのみについて変更することも、現在および今後のすべての環境変数について変更することもできます。

スナップショット名を変更するには、次の手順に従ってください。

1. 「環境変数」を選択します。
2. 環境変数リストで、デフォルトの環境変数のスナップショットの「スナップショット」アイコン をクリックします。
3. スナップショットのリストで、環境変数のスナップショットの「編集」アイコン をクリックします。
4. 「スナップショット」タブを選択します。
5. 「名前」に、新規名を入力します。
6. オプション: 「コメント」にコメントを入力します。
7. 重要: ポップアップで「OK」または「取り消し」を選択します。

OK	<p>環境変数のスナップショット名および他のスナップショット・オブジェクト名を変更します: 環境変数のスナップショットを作成した際にスナップショットを作成するように選択したオブジェクトについては、これらのオブジェクトの名前、および環境変数のスナップショットを変更します。</p> <p>基本スナップショットの場合: 現在のすべての環境変数スナップショットおよび今後のすべての環境変数スナップショットの基本スナップショット名を変更します。</p>
取り消し	<p>環境変数のスナップショット名は変更しますが、他のスナップショット・オブジェクト名は変更しません: 環境変数のスナップショットを作成した際にスナップショットを作成するように選択したオブジェクトについては、これらのオブジェクトの名前を変更しません。環境変数のスナップショット名のみを変更します。</p> <p>基本スナップショットの場合: 現在のすべての環境変数スナップショットおよび今後のすべての環境変数スナップショットの基本スナップショット名を保持します。</p>


## スナップショットへのアクセスおよびその表示

環境変数のスナップショットを作成すると、少なくとも 2 つの環境変数 (基本スナップショット環境変数と新規の環境変数スナップショット) を含むスナップショット・セットが作成されます。



スナップショット・セット内のすべての環境変数スナップショットを表示するには、次のようにします。

1. 「環境変数」を選択します。

環境変数リストに、環境変数および環境変数スナップショットのリストが表示されます。最上位のスナップショットは、デフォルトの環境変数スナップショットです。

2. 「スナップショット」アイコン をクリックすると、スナップショット・セット内の環境変数スナップショットが表示されます。

「スナップショット」ビューでは、以下を行うことができます。

- 新規の環境変数スナップショットを作成します。作業を開始するには、「編集」アイコン をクリックします。
- 環境変数のデフォルト・スナップショットを変更します。新しいデフォルト・スナップショットの「編集」アイコン をクリックして、「デフォルトの設定」をクリックします。
- 標準の環境変数の場合と同様に、環境変数スナップショットの定義を編集します。

## 環境変数のスナップショットの削除

「環境変数の削除」オプションを使用することによって、環境変数のスナップショットを削除することができます。


他のオブジェクトが使用中の環境変数を削除することはできません。例えば、他の環境変数に組み込まれている環境変数や、プロジェクト、ステップ、スケジュール、またはサーバーによって使用されている環境変数などです。

環境変数のスナップショットを削除するには、次のようにします。

1. 「環境変数」を選択します。

2. 環境変数リストで、基本スナップショットの「スナップショット」アイコン をクリックします。

「スナップショット」ビューにセット内の環境変数スナップショットが表示されます。

3. 削除する環境変数スナップショットの横にある「編集」アイコン をクリックします。
4. 「環境変数の削除」をクリックします。





---

## 第 19 章 プロジェクトの処理

このトピックでは、管理コンソールでのプロジェクトの作成と管理の方法について説明します。

---

### プロジェクトについて

プロジェクトは、独自の環境変数グループおよびサーバー・プロパティを備えた、実行可能なステップのセットです。

プロジェクトのほかに、ライブラリーを作成することができます。プロジェクトとは異なり、ライブラリーには、ステップが入力される場所を決定するセクターがありません。

### プロジェクト・パネルについて

「プロジェクト」パネルは、新規プロジェクトを作成したり、既存プロジェクトを表示または編集したりする場合に使用します。「プロジェクト」パネルにアクセスするには、左側のメニューで「プロジェクト」をクリックします。

「プロジェクト」パネルには以下のタブがあります。

- 『「詳細」タブ』
- 361 ページの『「タグ」タブ』
- 362 ページの『「レジスター」タブ』
- 362 ページの『「メモ」タブ』
- 363 ページの『「スナップショット」タブ』

#### 「詳細」タブ

「詳細」タブでは、さまざまなプロジェクト・プロパティを指定できます。

#### ジョブ・コントローラー

プロジェクト・エンジン・タイプ用のジョブ・コントローラーを選択します。選択項目は、Java マスター・ジョブ・コントローラーまたは Perl エンジンです。システムは、マスター・ジョブ・コントローラー・プロジェクトの実行に Java マスター・ジョブ・コントローラーを使用します。システムは、Perl エンジン・プロジェクトを実行する場合にのみ Perl エンジンを使用します。エンジン・タイプについて詳しくは、262 ページの『システム構成の設定』のトピックにある「**Java マスター・ジョブ・コントローラーをジョブ実行エンジンとして使用する (Use Java Master Job Controller as Job Execution Engine)**」の設定を参照してください。また、288 ページの『Perl エンジンの管理』のセクションにあるトピックも参照できます。

**名前** プロジェクトの名前。システムは、リスト内およびデータベース内にあるプロジェクトを参照する際に、この名前を使用します。

プロジェクトを実行した場合、プロジェクト・ディレクトリーの構成にはプロジェクト名が使用されます。プロジェクトには、さまざまなオペレーティ

ング・システムで実行される可能性のあるステップが含まれるため、プロジェクト名には、特殊な文字や記号は使用しないでください。無効なディレクトリー名を生成する文字をプロジェクト名に使用しなければならない場合は、「無効な相対ディレクトリー文字」システム構成設定にそれらの文字を追加してください。この設定にリストされた文字は、プロジェクト・ディレクトリーの作成時に下線 ( \_ ) に変換されます。アンパーサンド ( & ) を使用すると、意図せずにプロジェクト名の一部として HTML エンティティを生成してしまう可能性があります。

## アクセス

プロジェクトの表示と使用が許可されたアクセス・グループ。このプロパティは、アクセスできる対象を決定する権限とともに使用できます。例えば、ジョブを起動する場合は、プロジェクトに対して指定されたアクセス・グループのメンバーであると同時に、「ジョブの実行 (Execute Jobs)」権限を持つグループのメンバーでもなければなりません。アクセス・グループについて詳しくは、243 ページの『アクセスの概説』を参照してください。

**無効** プロジェクトを無効にするには、このチェック・ボックスにチェック・マークを付けます。無効に設定されたプロジェクトを実行しようとすると、Rational Build Forge はそのプロジェクトが非アクティブであることを示すメッセージが表示され、プロジェクトを実行できません。

## 最大スレッド

プロジェクトが起動を許される並列プロセスの最大数。このフィールドを使用して、プロジェクトがシステム・リソースを使いすぎないようにします。スレッド対応の各ステップと任意のインライン・プロジェクト (それ自体、スレッド対応のステップを起動できる) は、並列プロセスになることができるが、これらのプロセスはすべて、親プロジェクトの最大数に対してカウントされます。スレッド数の最大値に到達すると、新規並行プロセスの起動は停止され、プロジェクトの並列プロセス数がスレッド数の最大値を下回るまで待機した後、システムは続行されます。スレッド化について詳しくは、417 ページの『スレッド: ステップの並列実行』を参照してください。

## 実行上限数

「実行上限数」プロパティは、同時に許可されるプロジェクト・ジョブの最大数を設定するものです。

- プロジェクトを起動したときに、アクティブなジョブ数が「実行上限数」に等しい場合、新規ジョブは、少なくとも 1 つのジョブが完了するまで待機キューにとどまります。
- スケジュールによってプロジェクトの起動が試みられたときに、アクティブなジョブの数が「実行上限数」に等しく、「ハード実行上限数」システム構成が「はい」に設定されている場合、システムは新規ジョブを起動しません。「ハード実行上限数」が「いいえ」に設定されている場合、システムは、スケジュールされたビルドに関する「実行上限数」設定を無視します。
- インラインで起動されたプロジェクトは、オリジナル・プロジェクトのインスタンスとは見なされず、その「実行上限数」には算入されません。

**クラス** 各プロジェクトはクラスに割り当てる必要があり、クラスはグローバル・プロパティをジョブのグループに割り当てます。詳しくは、382 ページの『クラス』を参照してください。

## セレクトター

プロジェクトのサーバーを選択するときのセレクトターの名前。システムでは、このセレクトターが、独自のセレクトターを指定していない、プロジェクト内のすべてのステップのデフォルトとして使用されます。 302 ページの『セレクトター』を参照してください。セレクトターが指定されない場合、プロジェクトはプロジェクト・パネル ではなくライブラリー・パネル に追加されます。ライブラリーは、呼び出し元のステップのセレクトターを使用します。このステップにセレクトターがない場合、ライブラリーは呼び出し元のプロジェクトのセレクトターを使用します。

## 成功チェーン、失敗チェーン

プロジェクト・ビルドが成功または失敗したときに実行するプロジェクトを選択します。成功チェーンまたは失敗チェーンをプロジェクト・レベルで設定しておく、プロジェクトの成功状況または失敗状況に基づいて、別の成功アクションまたは失敗アクションを呼び出すことができます。この機能は、成功または失敗アクションをプロジェクト内でステップ・レベルで設定することに似ています。プロジェクト・レベルの場合は、成功または失敗アクションは、ステップ状況ではなくプロジェクト実行状況によってトリガーされます。

## 環境変数

サーバー環境変数の後、かつステップ環境変数の前に適用される環境変数です。環境変数を一緒に機能させる方法について詳しくは、329 ページの『変数について』を参照してください。

## スティッキー

デフォルトのプロジェクト・セレクトターを使用するプロジェクトのすべてのステップを強制的に同一サーバー上にとどまらせるには、「スティッキー」チェック・ボックスを有効にします。その際、サーバーがビジーな場合はサーバーが使用可能になるまで待機する必要があります。このオプションについて詳しくは、367 ページの『ステップをサーバーにスティックさせる』を参照してください。

## デフォルトのステップ・プロバイダー

ステップ・コマンドの実行を実装します。このプロバイダーは、エンドポイント・エージェント・サーバーでコマンド・テキストを実行するという、期待どおりのレガシー動作を提供します。デフォルトのステップ・プロバイダーは、Java マスター・ジョブ・コントローラーのみです。Perl エンジン・プロジェクトに使用できるステップ・プロバイダーはありません。

## 開始通知 (Start Notify)、成功通知 (Pass Notify)、失敗通知 (Fail Notify)

これらのフィールドは、プロジェクトの開始、成功、または失敗のときに通知 E メールを送信するようにシステムに指示する場合に使用します。それには、これらのフィールドの 1 つまたはすべてでアクセス・グループを選択します。

## 「タグ」タブ

「タグ」タブは、プロジェクトのビルド・タグを管理する場合に使用します。

## タグ形式

プロジェクトのタグを定義するストリング。プレーン・テキストとタグ変数

参照が使用されます。タグ形式について詳しくは、427 ページの『ジョブ中のビルド・タグの変更』を参照してください。

### タグ同期

2 つのプロジェクトのタグ変数を同期化します。現行プロジェクトと同期化するタグ変数を持つプロジェクトを選択します。2 つのプロジェクトが同期化されると、同一プールから変数が選択されます。2 つのプロジェクトが順番に実行されると、最初のプロジェクトの値は 1 になり、次のプロジェクトの値は 2 になります。詳しくは、370 ページの『タグの同期化』を参照してください。

**タグ名** 変数名。タグ形式でタグ変数を使用する場合は、フォーム `$<Tag Name>` を使用してその名前を参照します。例えば、`MainVer` 変数と `B` 変数を使用するタグを作成するには、タグ・フォーマット `Build_${MainVer}.${B}` を使用します。例えば、`Build_005.1` のようなタグが取得されます。

**初期値** タグ変数の値を設定します。「自動インクリメント」オプションを使用しないと、変数は変更されるまでこの値を保持します。

### パディング

「なし」以外の「パディング」値を選択した場合に、タグで変数が使用されていると、システムはその変数の値に先行ゼロを追加します。これは、桁数が「パディング」値と同じになるように行われます。例えば、変数が現在 2 で、パディングが 3 であると、「002」の値が作成されます。パディングの範囲は 1 から 8 までです。

### 自動インクリメント

これが「はい」に設定されている場合、システムは変数値をプロジェクト・ジョブごとに 1 ずつ増やします。

## 「レジスター」タブ

プロジェクト・レジスターは、複数のビルド間で持続する情報を保管する場合に使用します。

### レジスター

プロジェクト・レジスターの名前。

**内容** レジスターの値。

追加情報については、430 ページの『プロジェクト・レジスター』を参照してください。

## 「メモ」タブ

「メモ」タブは、プロジェクトについて関心のある事項を保管する場合に使用します。

メモごとにタブに以下の内容が表示されます。

- ユーザーがメモを作成した日時
- メモを入力したユーザー
- メモ

## 「スナップショット」タブ


プロジェクト・スナップショットは、プロジェクトのインスタンスです。「スナップショット」タブには、スナップショットの名前が表示されます。

このタブは、スナップショット名の表示や変更を行ったり、スナップショット名にコメントを追加したりする場合に使用します。

---

## 「プロジェクト」プロパティーの変更

プロジェクト・レベルのプロパティーを変更するには、「プロジェクト」を選択し

てから、対象プロジェクト名の横にある「編集」アイコン  をクリックします。

### プロジェクト名

プロジェクトの名前。システムは、リスト内およびデータベース内にあるプロジェクトを参照する際に、この名前を使用します。

プロジェクトを実行した場合、プロジェクト・ディレクトリーの構成にはプロジェクト名が使用されます。プロジェクトには、さまざまなオペレーティング・システムで実行される可能性のあるステップが含まれるため、プロジェクト名には、特殊な文字や記号は使用しないでください。無効なディレクトリー名を生成する文字をプロジェクト名に使用しなければならない場合は、「無効な相対ディレクトリー文字」システム構成設定にそれらの文字を追加してください。この設定にリストされた文字は、プロジェクト・ディレクトリーの作成時に下線 ( \_ ) に変換されます。アンパーサンド ( & ) を使用すると、意図せずにプロジェクト名の一部として HTML エンティティーを生成してしまう可能性があります。

### アクセス

プロジェクトの表示と使用が許可されたアクセス・グループ。このプロパティーは、アクセスできる対象を決定する権限とともに使用できます。例えば、ジョブを起動する場合は、プロジェクトに対して指定されたアクセス・グループのメンバーであると同時に、「ジョブの実行 (Execute Jobs)」権限を持つグループのメンバーでもなければなりません。アクセス・グループについて詳しくは、243 ページの『アクセスの概説』を参照してください。

### タグ形式

プレーン・テキストとタグ変数参照を使用して、プロジェクトのタグを定義するストリング。タグ形式について詳しくは、427 ページの『ジョブ中のビルド・タグの変更』を参照してください。

### タグ同期

2 つのプロジェクトのタグ変数を同期化します。現行プロジェクトと同期化するタグ変数を持つプロジェクトを選択します。2 つのプロジェクトが同期している場合は、それらの変数は同じプールから取得されます。したがって、順番に実行される場合は、1 つのプロジェクトが値 1 を取得すると、次のプロジェクトは値 2 を取得し、以下同様に実行されます。詳しくは、370 ページの『タグの同期化』を参照。

### 最大スレッド

プロジェクトが起動を許される並列プロセスの最大数。このフィールドを使



用して、プロジェクトがシステム・リソースを使いすぎないようにします。スレッド対応の各ステップと任意のインライン・プロジェクト（それ自体、スレッド対応のステップを起動できる）は、並列プロセスになることができるが、これらのプロセスはすべて、親プロジェクトの最大数に対してカウントされます。スレッド数の最大値に到達すると、新規並行プロセスの起動は停止され、プロジェクトの並列プロセス数がスレッド数の最大値を下回るまで待機した後、システムは続行されます。スレッド化について詳しくは、417 ページの『スレッド: ステップの並列実行』を参照してください。

### 実行上限数

「実行上限数」プロパティは、同時に許可されるプロジェクト・ジョブの最大数を設定するものです。

- プロジェクトを起動したときに、アクティブなジョブ数が「実行上限数」に等しい場合、新規ジョブは、少なくとも 1 つのジョブが完了するまで待機キューにとどまります。
- スケジュールによってプロジェクトの起動が試みられたときに、アクティブなジョブの数が「実行上限数」に等しく、「ハード実行上限数」システム構成が「はい」に設定されている場合、システムは新規ジョブを起動しません。「ハード実行上限数」が「いいえ」に設定されている場合、システムは、スケジュールされたビルドに関する「実行上限数」設定を無視します。
- インラインで起動されたプロジェクトは、オリジナル・プロジェクトのインスタンスとは見なされず、その「実行上限数」には算入されません。

**クラス** 各プロジェクトはクラスに割り当てる必要があり、クラスはグローバル・プロパティをプロジェクトのグループに割り当てます。詳しくは、382 ページの『クラス』を参照してください。

### セクター

プロジェクトのサーバーを選択するときのセクターの名前。システムでは、このセクターが、独自のセクターを指定していない、プロジェクト内のすべてのステップのデフォルトとして使用されます。302 ページの『セクター』を参照してください。セクターが指定されない場合、プロジェクトはプロジェクト・パネルではなくライブラリー・パネルに追加されます。ライブラリーは、呼び出し元のステップのセクターを使用します。このステップにセクターがない場合、ライブラリーは呼び出し元のプロジェクトのセクターを使用します。

### 成功チェーン、失敗チェーン

プロジェクト・ビルドが成功または失敗したときに実行するプロジェクトを選択します。成功チェーンまたは失敗チェーンをプロジェクト・レベルで設定しておく、プロジェクトの成功状況または失敗状況に基づいて、別の成功アクションまたは失敗アクションを呼び出すことができます。この機能は、成功または失敗アクションをプロジェクト内でステップ・レベルで設定することに似ています。プロジェクト・レベルの場合は、成功または失敗アクションは、ステップ状況ではなくプロジェクト実行状況によってトリガーされます。

## 環境変数

サーバー環境変数の後、ステップ環境変数の前に、適用される環境変数。環境変数を一緒に機能させる方法については、329 ページの『変数について』を参照してください。

## スティッキー

デフォルトのプロジェクト・セクターを使用するプロジェクトのすべてのステップを強制的に同一サーバー上にとどまらせるには、「スティッキー」チェック・ボックスを有効にします。その際、サーバーがビジーな場合はサーバーが使用可能になるまで待機する必要があります。このオプションについて詳しくは、367 ページの『ステップをサーバーにスティックさせる』を参照してください。

## 開始通知 (Start Notify)、成功通知 (Pass Notify)、失敗通知 (Fail Notify)


これらのフィールドは、プロジェクトの開始、成功、または失敗のときに通知 E メールを送信するようにシステムに指示する場合に使用します。それには、これらのフィールドの 1 つまたはすべてでアクセス・グループを選択します。

---

# プロジェクトのコピー

このトピックでは、既存のプロジェクトをコピーする方法について説明します。

**注:** Java マスター・ジョブ・コントローラー・プロジェクトを Perl プロジェクトにコピーすることはできません。

1. 「プロジェクト」を選択します。
2. コピーするプロジェクトの隣にある「編集」アイコン  を選択します。
3. 「プロジェクトのコピー」をクリックします。エンジンの選択ダイアログ・ボックスが表示されます。
4. ドロップダウン・リストから、コピーするプロジェクトに応じたエンジン・タイプを選択し、「OK」をクリックします。

プロジェクトのコピー時に、システムは既存プロジェクトの以下の参照を新規プロジェクトにコピーします。ソース・プロジェクトにチェーニングされたプロジェクトがある場合、または他のプロジェクトにチェーニングされたりインライン化されたステップがソース・プロジェクトに含まれている場合、警告メッセージが表示されます。このメッセージは、チェーニングまたはインライン化されたプロジェクトが正常にビルドされるように、これらのプロジェクトをコピーする際の注意喚起のためのものです。

- ステップの「詳細」タブにリストされているステップとそのすべてのプロパティ。
- プロジェクトのクラス、セクター、他のプロパティなど、プロジェクトの「詳細」タブにリストされているすべてのプロジェクト・プロパティ。
- 「タグ」タブにあるプロジェクト・タグ・フォーマット。

**注:** このタブはプロジェクト・プロパティ・パネルにあります。

- ソース・プロジェクトで Perl エンジンを使用しており、ターゲット・プロジェクトが Java エンジンを使用している場合、Perl プロジェクトの成功/失敗のチェー



ン・プロジェクトも Java エンジンとしてコピーされます。Perl プロジェクトのステップ・インライン・プロジェクトは、Java エンジンとしてコピーされます。

注: Java エンジン・プロジェクトを Perl エンジン・プロジェクトとしてコピーすることは、サポートされていません。

次のプロパティはコピーされません。

- 「**タグ**」タブにあるタグ変数。

注: このタブはプロジェクト・プロパティ・パネルにあります。タグ・フォーマットで使用されるタグ変数がコピーされます。

- 「**レジスター**」タブにあるプロジェクト・レジスター。

注: このタブはプロジェクト・プロパティ・パネルにあります。

- 「**メモ**」タブにあるステップ・メモ。

注: このタブはプロジェクト・プロパティ・パネルにあります。

- 「**メモ**」タブにあるプロジェクト・メモ。

注: このタブはプロジェクト・プロパティ・パネルにあります。

---

## プロジェクトの削除


プロジェクトを削除するには、プロジェクトに関連したジョブがあるかどうかに応じて 2 つの方法があります。

プロジェクトを削除するために、次のオプションのいずれかを選択します。

### 「プロジェクトの削除」ボタン

注: プロジェクトは、削除すると元に戻すことはできません。


「**プロジェクトの削除**」ボタンで、ジョブのないプロジェクトを削除します。このボタンでプロジェクトを削除するには、最初にプロジェクトのジョブをすべて削除しておく必要があります。このボタンは、プロジェクト・プロパティの編集ページとプロジェクト・ステップのリストにあります。プロジェクト・プロパティの編集ページを表示するには、以下のステップを実行します。

1. 左側のメニューで、「**プロジェクト**」を選択します。
2. 対象プロジェクトの名前の隣にある「**編集**」アイコン  をクリックします。

### 「強制削除」ボタン

注: 強制削除操作では、ロックされたプロジェクトも削除されます。また、プロジェクトの強制削除を元に戻すことはできません。


「**強制削除**」ボタンでは、プロジェクトとそれに関連したすべてのジョブを Build Forge データベースから削除します。プロジェクトを強制削除する前に、確認が求められます。このボタンは、プロジェクト・プロパティの編集ページにあります。プロジェクト・プロパティの編集ページを表示するには、以下のステップを実行します。

1. 左側のメニューで、「プロジェクト」を選択します。
2. 対象プロジェクトの名前の隣にある「編集」アイコン  をクリックします。

---

## ステップをサーバーにスティックさせる

プロジェクト内のステップは、セクターによって許される場合には、さまざまなサーバーで実行することができます。しかし、前もって特定のサーバーを指定したかどうかにかかわらず、プロジェクトの全部または大半のステップをその同じサーバーで実行することができます。プロジェクト・レベルの「スティッキー」プロパティを指定すると、これを行うことができます。

プロジェクト・レベルのプロパティを表示するには、「プロジェクト」を選択してから、対象プロジェクトの名前の隣にある「編集」アイコン  をクリックします。

「スティッキー」プロパティは、プロジェクト内の、独自のセクターが指定されていないステップにのみ適用されます。あるステップに「デフォルト」以外のセクター・オプションが指定されている場合、システムは、そのセクターを使用してそのステップ用のサーバーを選択します。そのセクターがプロジェクトのセクターと同じであっても同様です。

「スティッキー」プロパティが設定されていると、プロジェクトは、セクター・フィールドが「デフォルト」に設定されているすべてのステップについて、同じサーバーを使用します。このプロパティは、プロジェクトを再始動しても持続します。

インライン・プロジェクトを開始する際、システムはインラインのステップのデフォルトのセクターとして、そのインライン・プロジェクトのセクターを使用します。呼び出し元プロジェクトの「スティッキー」プロパティはインライン・プロジェクトに影響を与えず、インライン・プロジェクトは、独自の「スティッキー」プロパティが設定されている場合にはそれに従います。

システムは、インライン・ライブラリーを開始するときに、以下の規則に従います。

- 「スティッキー」プロパティにチェックが付いていないインライン・ライブラリー: 呼び出し元ステップのセクターをインライン・ステップのデフォルト・セクターとして使用します。
- 「スティッキー」プロパティにチェックが付いているインライン・ライブラリー: 呼び出し元ステップのサーバーをインライン・ステップのデフォルト・サーバーとして使用します。

注: `.bset` サーバー・コマンドを使用することにより、ジョブの実行中にプロジェクトのデフォルト・サーバーを変更できます。`.bset` コマンドの発行後に実行されるステップは、そのコマンドで設定された新規デフォルトを使用し、新規サーバーにスティックします。

---

## チェーン: 別のプロジェクトまたはライブラリーの条件付き実行

プロジェクト・レベルで呼び出し可能な以下の 2 つのタイプのチェーンがあります。

- 成功チェーン: プロジェクトが成功したときに実行するプロジェクトまたはライブラリーを指定します。
- 失敗チェーン: プロジェクトが失敗したときに実行するプロジェクトまたはライブラリーを指定します。

この機能には、いくつかの用途があります。

- プロジェクト・レベルで条件付き実行を使用します。他のフロー制御機能は、ステップ・レベルで提供されます。また、独自の「成功チェーン」と「失敗チェーン」をステップに指定することができます。407 ページの『第 20 章 ステップの処理』の 413 ページの『実行フローの制御』を参照してください。
- 頻繁に使用されるステップのグループを、それらに依存するプロジェクトから切り離して保守します。また、この目的にはライブラリーも使用できます。
- プロジェクトが成功または失敗した後にファイルをクリーンアップします。
- ソフトウェア・ビルド・プロジェクトが成功した際に、自動化されたテスト・プロジェクトおよび自動化されたデプロイメント・プロジェクトを呼び出します。

### 呼び出し元のプロジェクトまたはライブラリーからのチェーンの継承

チェーンングされたプロジェクトまたはライブラリーは、いくつかの特性を呼び出し元のプロジェクトから継承します。

- チェーニングされたプロジェクトは、デフォルトで呼び出し元のプロジェクトのクラスを継承します。この動作は、「管理」 > 「システム」で、「チェーニング時のクラスの指定変更」を「いいえ」に設定することによって変更できます。
- チェーニングされたライブラリーは、呼び出し元ステップのセクターを継承します。ライブラリーには独自のセクターがないためです。そのステップにセクターがない場合、ライブラリーはビルドのセクターを継承します。チェーニングされたライブラリーの各ステップでは、それらのステップに明示的なセクターがない限り、継承されたセクターを使用します。

それ以外の場合、チェーニングされたプロジェクトまたはライブラリーは、以下のように独自の特性を使用して実行されます。

- サーバーは、独自のセクターによって指定されます。
- 独自の通知設定およびチェーン設定など、独自のプロパティを使用します。
- 呼び出し元プロジェクトの環境変数の後で、独自の環境変数が適用されます。呼び出し元プロジェクトからの変数は名前変更され、呼び出し先プロジェクトで使用できます。369 ページの『チェーン・プロジェクトでの環境変数の継承』も参照してください。

### チェーン・ネスティング

プロジェクトをチェーニングする際、呼び出し先プロジェクトは呼び出し元プロジェクト内にネストされます。ネスティングの最大レベルは 32 レベルです。ネスティングのレベルには、管理コンソールを実行しているホスト上の使用可能メモリーに基づいた制限が課される場合もあります。

## チェーンの実行および中断

チェーンリングされたプロジェクト内で `.break` コマンドを使用する場合、システムにより、チェーンリングされたプロジェクトは停止しますが、呼び出しプロジェクトにコントロールが返され、処理は続行されます。 437 ページの『`.break`』を参照してください。

## チェーン・プロジェクトでの環境変数の継承

成功/失敗チェーンを使用してプロジェクトを起動すると、システムは呼び出し元のプロジェクトから環境変数を適用します。呼び出し先のプロジェクトは、以下の順序で呼び出し元のプロジェクトの環境変数および自身の環境変数から変数をセットアップします。

1. 呼び出し先のプロジェクトのサーバー環境変数。
2. 呼び出し元のプロジェクトの変数 (セット)。「BF\_」変数名は「BF\_CALLER」に変更されます。
3. 呼び出し先のプロジェクトの環境変数。
4. ステップ環境変数 (指定されている場合)。それらの実行順序に従ってセットアップします。

注: 変数セットアップの順序を制御するには、「チェーンリング前のサーバー環境変数」構成設定を使用できます。 262 ページの『システム構成の設定』を参照してください。

## 待機が使用可能な場合のチェーンリングされたプロジェクトの取り消し

通常、システムはチェーンリングされたプロジェクトを取り消すことはありません。「成功待機」または「失敗待機」属性を「はい」に設定して、システムが「成功チェーン」または「失敗チェーン」の呼び出し先プロジェクトを自動的に取り消すようにします。システムは、呼び出し元プロジェクトまたは呼び出し元ステップが取り消されたときに、呼び出し先プロジェクトを取り消します。

---

## タグの定義

システムは、タグを使用して、プロジェクトの特定のジョブを識別し、デフォルトでプロセス・アクティビティーが起こるジョブ・ディレクトリーの名前を構成します。システムは、プロジェクトの `タグ形式 プロパティー` から、ジョブのタグを作成します。この形式には、静的テキストのほか、数値の `タグ変数` も含めることができます。

プロジェクトのデフォルトのタグ形式は `BUILD_$B` です。これは、デフォルトのタグ変数 `B` を使用します。この変数は、自動的に増加する値で、プロジェクトごとにシステムによって定義されます。このデフォルトのタグ形式は、以下のようなビルド・タグのストリームになります。

`BUILD_1`

`BUILD_2`

## BUILD\_3

ただし、これらのタグに制限されることはありません。独自のタグ変数を定義して、独自のタグ形式をセットアップし、さまざまなタグ・タイプを生成できます。また、ジョブ中に `.retag` コマンドを使用して、タグを任意のストリングに変更することもできます。(詳しくは、454 ページの『`.retag`』を参照してください。)

現在のジョブのタグは、ジョブ中にシステムによって定義される環境変数 (BF\_TAG) として使用できます。このため、ソース・リポジトリのラベル付けのために、またはその他のトラッキングやラベリングの目的で、このタグにアクセスして使用することができます。(これらの変数について詳しくは、341 ページの『システム変数のリファレンス』を参照してください。)

2 つのプロジェクトのタグ変数を同期化することができます。これにより、いずれか一方が実行されると同じタグ変数の値が使用されるようなリンクが作成されます。(詳しくは、『タグの同期化』を参照してください。)

このセクションのトピックでは、タグ形式およびタグ変数をセットアップして、希望する値を反映する動的なタグを生成する方法を説明します。

## プロジェクトのタグ形式の編集

タグ形式には、システムによるタグの作成方法が定義されています。タグ形式は、プレーン・テキストおよび \$ 記号によって示される変数の参照から成っています。タグ形式で使用する変数はすべて、前のセクションのシステム定義タグ変数のリストにあるものか、プロジェクトの実行前にプロジェクト用に定義されたものでなくてはなりません。未定義の変数は、静的テキストとして扱われます。

タグ形式は、プロジェクト・プロパティの 1 つです。編集するには、「プロジェクト」ボタンをクリックしてプロジェクトのリストを表示し、編集したいプロジェクトのプロジェクト名をクリックします。すると、プロジェクト内のステップのリストが表示されます。ページ上部のプロジェクト名をクリックすると、プロジェクトのプロパティが表示されます。

タグ形式では、\$ 記号でタグ変数の開始を示します。必要に応じて、いくつかのタグ変数を使用することができます。例えば、プロジェクトの大幅な改訂 (\$MAJ) については非増分変数を定義し、軽微な改訂 (\$MIN) については増分変数を定義して、タグ形式でプロジェクトのバージョン番号を反映させることができます (例えば、Version\$MAJ.\$MIN)。これにより、メジャー・バージョン番号は手動で制御できますが、マイナー・バージョン番号はリリース毎に次に示すようなタグを生成するため、自動的に増分できます。

Version1.1

Version1.2

## タグの同期化

複数のプロジェクトで同じ変数値を使用できるように、プロジェクト・レベルの「タグ同期」プロパティを使用して、異なるプロジェクト間でタグを同期化する

ことができます。プロジェクト A と同等のプロジェクト B に対して、タグ同期プロパティを設定すると、親子関係がプロジェクト A (親) とプロジェクト B (子) の間に確立します。

「タグ同期」プロパティを設定してプロジェクトを実行すると、システムは、子プロジェクトのタグ変数に、親プロジェクトのタグ変数と一致するものがないか検査します。見つかった場合、子プロジェクトの変数が、親プロジェクトで最後に使用した値に設定されます。

子プロジェクトのタグ形式にある変数に、親プロジェクトのタグ形式にある変数と一致するものがない場合、タグ同期プロパティは無効です。

同期の対象となるのは、タグ変数の値のみです。「自動インクリメント」プロパティおよび「パテイング」プロパティは同期化されません。

タグの変数のみが同期化されるので、異なるプロジェクトを区別することができます。

例えば、次の表に定義されているような 2 つのプロジェクトについて考えてみます。

プロジェクト	タグ形式	自動インクリメント	タグ同期
プロジェクト A	Project_A_\$B	はい	-- なし --
プロジェクト B	Project_B_\$B	はい	プロジェクト A

これらのプロジェクトを交互に (プロジェクト A から開始) 実行すると、完了ジョブ・リストにタグが以下のように表示されます。最後に実行されたものが一番上に表示されます。完了ジョブ・リストのジョブの表示方法も同様です。

プロジェクト	タグ
プロジェクト B	Project_B_4
プロジェクト A	Project_A_3
プロジェクト B	Project_B_2
プロジェクト A	Project_A_1

親プロジェクトの「自動インクリメント」プロパティのみを「はい」に設定した場合、結果は異なります。プロジェクトのセットアップは以下のようになります。

プロジェクト	タグ形式	自動インクリメント	タグ同期
プロジェクト A	Project_A_\$B	はい	-- なし --
プロジェクト B	Project_B_\$B	いいえ	プロジェクト A



これらのプロジェクトを交互に (プロジェクト A から開始) 実行すると、完了ジョブ・リストにタグが以下のように表示されます。最後に実行されたものが一番上に表示されます。完了ジョブ・リストのジョブの表示方法も同様です。

プロジェクト	タグ
プロジェクト B	Project_B_2
プロジェクト A	Project_A_2
プロジェクト B	Project_B_1
プロジェクト A	Project_A_1

## タグのシステム定義変数

以下の定義済み変数をジョブ・タグで使用できます。

変数	値
B	ジョブ番号: 1 から始まり、新規のジョブごとに増加する整数値。
BF_D	日付。「タグ: 日付書式」システム設定で設定されるフォーマットで表される。これは実行時に評価されるため、プロジェクト開始ページの「タグ・サンプル」では評価できない。使用する場合は、{ } で囲む。
BF_J	通算で表す年の日。これは実行時に評価されるため、プロジェクト開始ページの「タグ・サンプル」では評価できない。使用する場合は、{ } で囲む。
BF_T	時刻。「タグ: 時刻書式」システム設定で設定されるフォーマットで表される。これは実行時に評価されるため、プロジェクト開始ページの「タグ・サンプル」では評価できない。使用する場合は、{ } で囲む。
BF_W	曜日 (0 から 6 で表した数値)。これは実行時に評価されるため、プロジェクト開始ページの「タグ・サンプル」では評価できない。使用する場合は、{ } で囲む。

## システム定義の変数、および開始ページの「タグ・サンプル」

いくつかの変数は、ジョブの開始時に評価されるため、開始ページの「タグ・サンプル」では評価できません。これらを使用する場合は、{ } で囲んでください。

例:




```
BUILD_$B.${BF_T}  
BUILD_$B.${BF_D}.${BF_T}
```

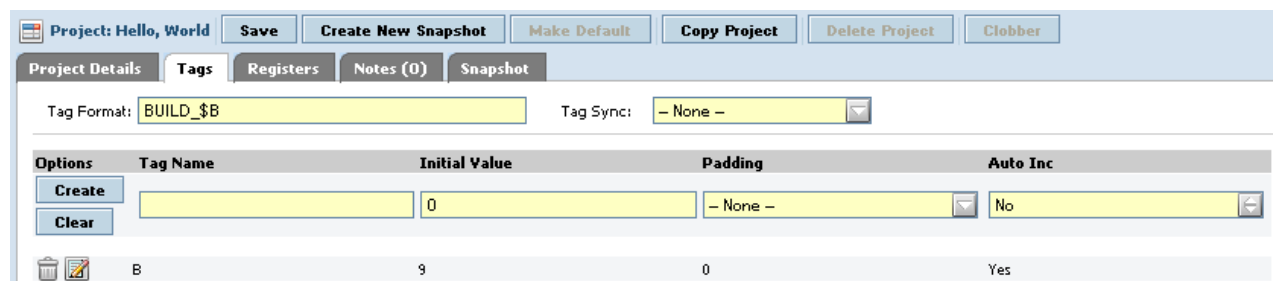
## タグ変数の作成または編集


独自のタグ変数を定義して、タグ形式に含めることができます。タグ変数は数値をとり、必要に応じてジョブ毎に自動的に増分させることができます。

プロジェクトで新規の変数を追加したり、既存の変数を編集したりするには、「プロジェクト」を選択してから、対象プロジェクトの名前の隣にある「編集」アイコン

 をクリックします。パネルの下部にプロジェクト・プロパティが表示されます。「タグ」タブをクリックすると、プロジェクトのタグ変数が表示されます。

プロジェクトのタグ変数のリストが表示されます。



- ・タグ変数を編集するには、タグ変数名の隣にある「編集」アイコン  をクリックします。パネルにタグ変数の値が表示され、「作成」が「保存」ボタンに変更されます。値を変更し、「保存」をクリックして、変更内容を保管します。
- ・タグ変数を削除するには、タグ変数名の隣にあるごみ箱アイコンをクリックします。
- ・新規変数を追加するには、変数のプロパティを入力し、「作成」ボタンをクリックします。

「タグ名」、「初期値」、「パディング」、および「自動インクリメント」の各プロパティについては、361 ページの『「タグ」タブ』を参照してください。

## ライブラリー

ライブラリーは、「セレクトター」プロパティを「なし」に設定したプロジェクトです。ライブラリーは、他のプロジェクト内で実行されるように意図されており、それらのプロジェクトを呼び出すステップのサーバー・リソース上で実行されます。

セレクトターが「なし」のプロジェクトを保存するときに、システムはそれがライブラリーとして保存されると警告します。ライブラリーは、「ライブラリー」パネルにリストされます。

ステップからライブラリーを呼び出すには、ステップの「インライン」、「成功チェーン」、または「失敗チェーン」のいずれかのプロパティで選択します。

## ライブラリーについて

「ライブラリー」パネルには、ライブラリーが含まれ、それらのライブラリーはセクターが指定されていないプロジェクトです。

ライブラリーは、呼び出し元のステップのセクターを使用します。呼び出し元のステップにセクターがない場合、ライブラリーはそのステップのプロジェクトのセクターを使用します。通常、ライブラリーは、他のプロジェクトによってステップのインラインとして呼び出されるか、ステップの成功チェーンまたは失敗チェーンとして呼び出されます。

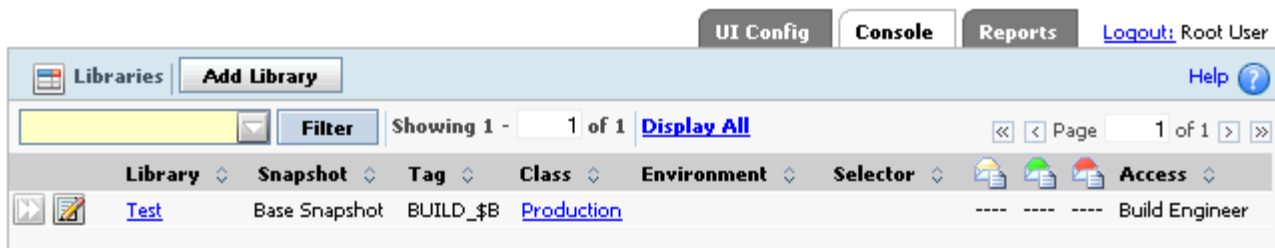
「ライブラリー」パネルからは、ライブラリーを表示、編集、作成、または起動することができます。ライブラリーは単独で実行することができますが、その場合にはセクターを指定する必要があります。

**注:** ライブラリーの起動時に、デフォルト設定により、リストの最初のセクターが割り当てられます。別のデフォルト・セクターを指定する場合は、そのセクターをプロジェクトに指定します。

ライブラリーを編集してそのプロジェクトのためのセクターを選択することにより、ライブラリー・プロジェクトをプロジェクトに変更することができます。セクターが指定されているライブラリーを保存すると、そのライブラリーはプロジェクトになり、「ライブラリー」リストから消去されます。


ライブラリーは、セクターがないことを除くと、他のすべてのプロジェクトと同様に扱われます。

「ライブラリー」パネルにアクセスするには、左側のメニューで「ライブラリー」をクリックします。



## ライブラリーのコピー

既存ライブラリーのコピーを作成するには、以下のステップを実行します。

1. 「ライブラリー」を選択します。
2. コピーするライブラリーの隣にある「編集」アイコン  を選択します。
3. 「ライブラリーのコピー」をクリックします。

ライブラリーのコピー時には、既存ライブラリーの次の特徴が新規ライブラリーにコピーされます。

- ステップの「詳細」タブにリストされているステップおよびそれらのすべてのプロパティ
- クラスおよび他のプロパティなど、ライブラリーの「詳細」タブにリストされているライブラリーの全プロパティ
- ライブラリーのタグ形式 (ライブラリー・プロパティ・パネルの「タグ」タブにあります)

次のプロパティはコピーされません。

- タグ変数 (ライブラリー・プロパティ・パネルの「タグ」タブにあります)。ただし、タグ形式で使用されるタグ変数はコピーされます。
- ライブラリー・レジスター (ライブラリー・プロパティ・パネルの「レジスター」タブにあります)
- ステップ・メモ (ステップ・プロパティ・パネルの「メモ」タブにあります)
- ライブラリー・メモ (ライブラリー・プロパティ・パネルの「メモ」タブにあります)

---

## ログ・フィルター

このトピックでは、ログ・フィルターの作成方法および使用方法について説明します。

### ログ・フィルターについて

ログ・フィルターを使用して、ステップの成功基準を指定します。フィルターには、1 つ以上の正規表現が保管されます。

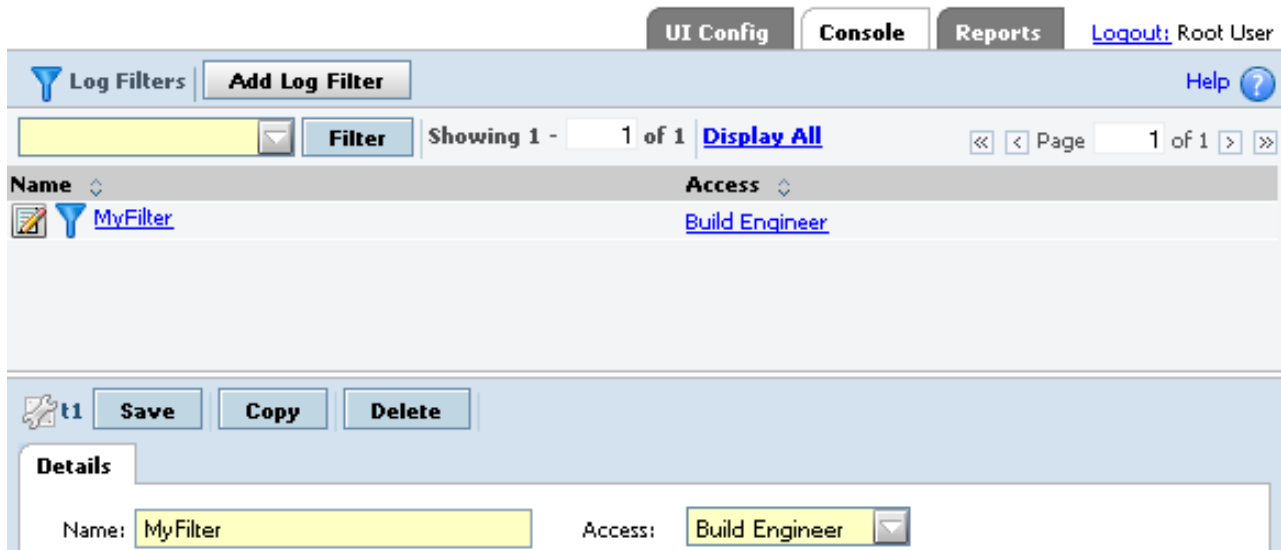
フィルターがセットアップされていないと、Build Forge ではステップ・コマンドが成功したか失敗したかをコマンドの終了状況で判別します。0 が成功で、1 が失敗です。コマンド・プロパティに複数のコマンドが使用された場合、最後に発行されたコマンドの終了状況のみが、ステップ結果状況に影響します。いくつかのコマンドは、常に 0 の終了状況を返すことに注意してください。net use などのレポート・コマンドは、マップされたネットワーク・ドライブのリストを出力します。リストに該当するドライブがない場合にも、コマンドは常に成功します。

ログ・フィルターによって、終了状況ではなく、ステップ内のコマンドからの出力を評価することができます。

例えば、net use コマンドでは、ログ・フィルターを使用することで、特定のドライブを検索し、そのドライブが見つかった場合にステップに成功のマークを付けることができます。

### 「ログ・フィルター」パネルについて

「ログ・フィルター」パネルは、ログ・フィルターを作成、編集、および削除する場合に使用します。パネルを表示するには、「プロジェクト」 > 「ログ・フィルター」を選択します。



パネルには、以下のフィールドがあります。

**名前** ログ・フィルターの名前。

**アクセス**

このフィルターを使用できるユーザーのアクセス・グループ。

## ログ・フィルターの作成

ログ・フィルターには、1 つ以上のフィルター・パターンを入れることができます。それぞれのフィルター・パターンはアクションと関連付けられていますが、必要に応じて、通知のためにアクセス・グループにも関連付けます。まずログ・フィルターを定義し、次にそのログ・フィルターをプロジェクト内のステップと関連付けます。

ログ・フィルターを作成するには、次の手順を実行してください。

1. 「プロジェクト」 > 「ログ・フィルター」を選択します。管理コンソールに、ログ・フィルターのリストと「新規ログ・フィルター」パネルが表示されます。
2. 「名前」に、ログ・フィルターの名前を入力し、「保存」をクリックします。管理コンソールでログ・フィルターが保存され、「新規パターン」パネルが表示されます。
3. ログ・フィルターに定義するフィルター・パターンごとに、次の手順を実行してください。
  - a. 「パターン」フィールドに正規表現を入力します。この正規表現は、Java エンジン・プロジェクトの場合は Java 互換でなければならず、Perl エンジン・プロジェクトの場合は Perl 互換でなければなりません。プロジェクトの実行時に、Build Forge はステップ出力でそのパターンを検索します。詳しくは、378 ページの『フィルター・パターン』を参照してください。
  - b. 「アクション」で、フィルター・パターンが検出されたときにとるフィルター・アクションを選択します。デフォルトのプロパティ「失敗の設定」は、ステップ状況を「失敗」に設定します。詳しくは、379 ページの『フィルター・アクション』を参照してください。

- c. 必要に応じて、「通知」でアクセス・グループを選択して、フィルターが活動化しているときに、メンバーに通知の E メールが送信されるようにします。
- d. 「保存」をクリックします。

ログ・フィルターを使用するには、プロジェクト・ステップを選択し、そのステップの「結果」プロパティを新規ログ・フィルターに設定します。『ステップへのログ・フィルターの割り当て』を参照してください。

## ステップへのログ・フィルターの割り当て

ログ・フィルターを使用するには、ステップの「結果」プロパティを使用して、ログ・フィルターをプロジェクト・ステップに割り当てる必要があります。ログ・フィルターをステップに割り当てると、プロジェクトが実行されるときは常に、ログ・フィルターのフィルター・パターンがステップ出力に対し実行されます。ただし、アダプター出力には、ログ・フィルターが適用されません。

ログ・フィルターをステップに割り当てると、ログ・フィルターで設定されているステップ結果が、ステップの成否を判別するための他のすべての基準 (ステップ・コマンドの終了状況やすべてのステップ・プロパティを含む) をオーバーライドします。例えば、ステップの実行時間がステップの「タイムアウト」プロパティで指定されている時間を超えた場合、ステップは停止します。ただし、その状況は、関連付けられたログ・フィルター・アクションによって「失敗」に設定されない限り、失敗とは見なされません。

ログ・フィルターをステップに割り当てるには、次の手順を実行してください。

1. ステップにアクセスするために、「プロジェクト」または「ライブラリー」を選択します。
2. ステップを含むプロジェクトまたはライブラリーを選択します。
3. ステップを選択して、ステップの「詳細」パネルを開きます。
4. 「結果」で、ステップが実行されるたびに実行するログ・フィルターを選択します。

## ログ・フィルターを有効にしてエントリーをステップに組み込む

ログ・フィルターは以下のステップ・ログ行タイプに適用できます。

- EXEC
- RESULT
- ERROR
- READ
- WRITE
- MANIFEST
- TMO

適用すると、対応するステップ・ログ出力行が、フィルターに掛けられた結果に組み込まれます。ログ・フィルターは STEP タイプには適用できません。STEP タイプは、ログ・フィルターでは常にスキップされます。

## フィルター・パターン

フィルター・パターンには、ステップ出力内で突き合わせる文字ストリングまたは式を定義します。作成するそれぞれのフィルター・パターンは、1 つのフィルター・アクションと関連付けられます。フィルター・ログ・セットには、フィルター・パターンとアクションの両方が定義されます。ログ・フィルターに複数のフィルター・パターンを含め、それを単一ステップからの出力に適用することができます。これにより、複雑な式を作成することなく、複数の検索条件を使用することができます。

ログ・フィルターを作成するには、「プロジェクト」 > 「ログ・フィルター」を選択します。詳しくは、375 ページの『ログ・フィルター』を参照してください。

### フィルター・パターンの構文

フィルター・パターンを作成するために、次のガイドラインを読んでください。

- フィルター・パターンは正規表現として定義されます。フィルター・パターンでは、Java エンジン・プロジェクトに対しては Java 互換の構文が使用され、Perl エンジン・プロジェクトに対しては Perl 互換の構文が使用されなければなりません。互換性のある正規表現を作成する方法について詳しくは、Java 正規表現の資料、または Perl の資料を参照してください。
- 区切り文字となるスラッシュはシステムによって追加されるため (`<expression>/`)、式を指定する際は、スラッシュの区切り文字で囲まない てください (`expression`)。
- 式にメタキャラクター (例: `a /b`) が使用されている場合、メタキャラクターの前にエスケープ文字となる円記号 (`¥`) を付加する必要があります (`a¥/b`)。

次の表には、正規表現の式の構文がいくつか示されています。

式	一致
<code>Production</code>	<code>Production</code> がストリング内の任意の場所にある場合に一致します。
<code>^Production</code>	<code>Production</code> がストリングの先頭にある場合に一致します。
<code>Error:.*[0-9]\$</code>	<code>Error</code> を含み、その後の一連の任意の文字が続き、ストリングの末尾が数字で終わる行と一致します。
<code>[Ww]arning</code>	<code>Warning</code> または <code>warning</code> と一致します。
<code>.*</code>	任意の文字と 0 回以上一致します。ドット ( <code>.</code> ) は任意の文字と一致し、アスタリスク ( <code>*</code> ) は 0 回以上一致します。

### 同じ行にある複数のパターン一致

パターン・フィルターを作成するにあたり、システムがどのようにパターンの一致を検索しているか理解することは重要です。

システムは、出力の各行を、すべてのフィルター・パターンと順番に照らし合わせて、一致がないか検査します。一致が見つかったらシステムは停止し、次のパターン

に移ります。したがって、同じ行にそのパターンが 2 回出現しても、システムはそれを検出しない可能性があります。以下の出力行を例に取ります。

```
exception retrying exception
```

システムは、次の表に記載されているフィルター・パターンを使用して、最初の *exception* を突き合わせてステップ結果を「失敗」に設定し、*retrying* を突き合わせてステップ結果を「成功」に設定した後、2 番目の *exception* を突き合わせることなく次の行へ移ります。

フィルター・パターン	フィルター・アクション	例の説明
[Ee]xception [Rr]etrying	失敗の設定 - 失敗 失敗のクリア - 成功	このパターンは、Java プロジェクトで便利です。例外でステップが失敗しますが、再試行でその失敗がクリアされます。再試行が失敗した場合は、新しい例外が生成され、コマンドの最終状態は妥当なものになります。

この問題の解決法の 1 つは、表に記載されているフィルター・パターンを次のフィルター・パターンで置換することです。

```
retrying.*exception
```

## フィルター・アクション

フィルター・アクションには、ステップ出力からフィルター・パターンが検出されたときにとるアクションを定義します。作成するそれぞれのフィルター・パターンは、1 つのフィルター・アクションと関連付けられます。ログ・フィルターには、フィルター・アクションとフィルター・パターンの両方が定義されます。

ログ・フィルターを作成するには、「プロジェクト」 > 「ログ・フィルター」を選択します。詳しくは、375 ページの『ログ・フィルター』を参照してください。

フィルター・アクション	定義	ステップ結果
失敗の設定 (デフォルト)	フィルター・パターンが検出されると、ステップ結果状況が「失敗」に設定され、その後も引き続き、セット内のフィルター・パターンがないか現在行を検索します。	失敗
失敗/停止の設定	フィルター・パターンが検出されると、ステップ結果状況が「失敗」に設定され、セット内のフィルター・パターンを対象とした現在行の検索は停止し、次の行にスキップしてパターン検索を再開します。	失敗
失敗のクリア	フィルター・パターンが検出されると、ステップ結果状況が「成功」に設定され、その後も引き続き、セット内にそのフィルター・パターンがないか現在行を検索します。	成功
失敗/停止のクリア	フィルター・パターンが検出されると、ステップ結果状況が「成功」に設定され、セット内のフィルター・パターンを対象とした現在行の検索は停止し、次の行にスキップしてパターン検索を再開します。	成功



フィルター・アクション	定義	ステップ結果
停止	フィルター・パターンが検出されると、セット内のフィルター・パターンを対象とした現在行の検索が停止し、次の行にスキップしてパターン検索を再開します。ステップ結果状況は変更されません。	適用外
組み込み	「組み込み」を使用すると、あるログ・フィルター内にある 1 つ以上のログ・フィルターを参照することができます。 「パターン」フィールドに、組み込むログ・フィルターを指定し、「アクション」フィールドで「組み込み」を選択します。	適用外
警告	フィルター・パターンが検出されると、ステップ結果状況が「警告」に設定され、その後も引き続き、セット内のフィルター・パターンがないか現在行を検索します。(注: 「警告」状況は成功途中の状況であるため、このステップに割り当てられたすべての成功チェーンが実行されます。)	警告
警告のクリア	フィルター・パターンが検出されると、ステップ結果状況が「成功」に設定され、その後も引き続き、セット内のフィルター・パターンがないか現在行を検索します。	成功
警告/停止のクリア	フィルター・パターンが検出されると、ステップ結果状況が「成功」に設定され、セット内のフィルター・パターンを対象とした現在行の検索は停止し、次の行にスキップしてパターン検索を再開します。	成功
通知チェンジャー	<p>通知チェンジャーを使用するには、関係リストを作成するアダプターをプロジェクト内に組み込む必要があります、そのアダプター・ステップは、通知チェンジャーのログ・フィルターを含むステップの前に実行する必要があります。</p> <p>アダプターの関係リストは、ユーザーとオブジェクト (変更されたファイルなど) を組み合わせてペアにします。詳しくは、『アダプター XML 参照』を参照してください。</p> <p>アダプターが実行され、関係リストが作成された後、通知チェンジャーのアクションを含むログ・フィルターがステップ出力行のフィルター・パターンと一致する場合、この行は、関係リスト内のオブジェクトとの照合を試みるために再度スキャンされます。オブジェクトの一致が見つかったら、オブジェクトとペアになっているユーザーに E メール通知が送信されます。</p> <p>以下のステップ出力の行の例では、フィルター・パターン <code>Error</code> のオブジェクト一致は <code>MyFile.c</code> です。そのため、関係リスト内で <code>MyFile.c</code> オブジェクトとペアになっているユーザーに、E メールのエラーの通知が送信されます。</p> <p>Error: Invalid token on line 55 of MyFile.c</p>	適用外

フィルター・アクション	定義	ステップ結果
失敗の結果でビルドを停止	<p>フィルター・パターンが検出されると、ジョブ結果が「失敗」に設定され、ジョブが終了します。</p> <p>Java エンジン・プロジェクトの場合、ステップ結果は「失敗」です。Perl エンジン・プロジェクトの場合、ステップ結果はその結果に応じて設定されます。それ以降のステップは実行されません。</p>	Java エンジン・プロジェクトの場合、ステップ結果は「失敗」です。Perl エンジン・プロジェクトの場合、ステップ結果はその結果に応じて設定されます。
成功の結果でビルドを停止	<p>フィルター・パターンが検出されると、ジョブ結果が「成功」に設定され、ジョブが終了します。</p> <p>Java エンジン・プロジェクトの場合、ステップ結果は「成功」です。Perl エンジン・プロジェクトの場合、ステップ結果はその結果に応じて設定されます。それ以降のステップは実行されません。</p>	Java エンジン・プロジェクトの場合、ステップ結果は「成功」です。Perl エンジン・プロジェクトの場合、ステップ結果はその結果に応じて設定されます。
警告の結果でビルドを停止	<p>フィルター・パターンが検出されると、ジョブ結果が「警告」に設定され、ジョブが終了します。</p> <p>Java エンジン・プロジェクトの場合、ステップ結果は「警告」です。Perl エンジン・プロジェクトの場合、ステップ結果はその結果に応じて設定されます。それ以降のステップは実行されません。</p>	Java エンジン・プロジェクトの場合、ステップ結果は「警告」です。Perl エンジン・プロジェクトの場合、ステップ結果はその結果に応じて設定されます。

## フィルター通知

ログ・フィルターのフィルター・パターン別に、必要に応じて通知を設定し、アクセス・グループに E メールを送信して、ステップ用のパターン・フィルターが活動状態であることをメンバーに通知するようにすることができます。

## エラーしきい値

特殊な環境変数 `_ERROR_THRESHOLD` を使用して、個々のステップまたはプロジェクト、あるいはその両方にしきい値を設定できます。システムはフィルターの一一致数をカウントし、しきい値に達したときにステップまたはプロジェクトを失敗とするか、しきい値に達したことをジョブ・メモで示します。

詳しくは、344 ページの『トリガー変数のリファレンス』を参照してください。

## エラー・カウントおよび警告カウント

ステップが成功したか失敗したかを判別するためにフィルターをステップに関連付けると、フィルターに掛かったエラーおよび警告の数がシステムによって表示されます。これらは、「ジョブ」 > 「完了」タブにある「結果」列に、ジョブ結果の後に括弧で囲んだ数字として示されます。形式は、(<失敗一致数> / <警告一致数>) です。

例として、失敗 (1 / 0) のジョブ結果は、ジョブが失敗したことを示し、1 個の失敗フィルターが一致し、警告フィルターは一致しなかったことを示します。

---

## クラス

このトピックでは、クラスの作成方法および使用方法について説明します。

## クラスについて

クラスとは、ジョブのグループのことです。各ジョブは 1 つのクラスのメンバーでなければならず、またメンバーであるのはそのクラスだけでなければなりません。クラスを使用して、クラス内のジョブごとに異なるグローバル管理動作を適用できます。ジョブは、そのプロジェクトのプロパティからデフォルト・クラスを取得します。ジョブを「ジョブ」 > 「開始」ページから起動すると、手動でジョブに別のクラスを選択することもできます。

クラスには以下のアクティビティを管理するためのプロパティがあります。

- ジョブを自動的に削除する。
- システムがこのクラスのジョブをパージしたとき、または既存のジョブがこのクラスに変更されたか、または既存のジョブがこのクラスから変更されたときに、ジョブを起動する。

**注:** ジョブが完了したら、ジョブのクラスを変更することができます。ジョブのクラスを変更するには、「ジョブ」 > 「完了」と選択してから、ジョブ・タグ (例えば、BUILD\_5) をクリックして、ジョブを表示します。「クラス」フィールドで、別のクラスを選択します。

## 「クラス」パネルについて

「クラス」パネルは、クラスを追加、編集、および削除する場合に使用します。パネルを表示するには、「プロジェクト」>「クラス」を選択します。システムはクラスのリストを表示します。クラスを選択してそのプロパティを編集します。

Build Class	Access	Delete	Days	Count	Which
<a href="#">Production</a>	<a href="#">Guest</a>	Console Data	Forever	Unlimited	Any Build
<a href="#">Scratch</a>	<a href="#">Guest</a>	Everything	5	Unlimited	Any Build

**(New Class)**

**Details**

Name:  Access:

Delete Files:  Start on purge:

Days:  Start on entry:

Count:  Start on exit:

Which:

クラスの「アクセス」プロパティでは、ユーザーが割り当てられているアクセス・グループを基にして、どのユーザーがクラスの表示や変更を行えるかを制御します。

### ジョブを自動的に削除するプロパティ:

クラスに関するほとんどのプロパティは、削除されるプロジェクト・データの種類とその条件を制御します。

システムは、「パージ検査時間 (Purge Check Time)」のシステム設定で定義されている間隔に従って、削除するジョブがないか検査します。デフォルトは 15 分間になっています。

注: また、スケジュールを使用して、いつパージを実行するか指示することができます。これにより、システムは、他の理由で使用中有るときにパージの実行を試みません。この機能を使用すると、例えば、夜間のみ、あるいは週に 1 回、パージを実行させることができます。382 ページの『クラス』を参照してください。

パージ・ジョブを実行すると、システムはジョブをアーカイブし、クラスの設定値に従ってデータを削除します。

### ファイルの削除

削除するデータの種別を判別します。次のオプションがあります。

すべて ジョブに関する情報をすべてデータベースから削除し、それを実行したサーバーからジョブ・ディレクトリーを削除します。

### コンソール・データ

ジョブに関する情報をすべてデータベースから削除するが、サーバー上のジョブ・ディレクトリーはそのまま残します。

### ログおよびファイル

ジョブ・ディレクトリーおよびログを削除しますが、ステップの成功および失敗情報は、「ジョブ」 > 「アーカイブ」ページに保存します。

### ログのみ

ジョブ・ログのみを削除します。

### ファイルのみ

ジョブを実行したサーバー上にあるジョブ・ディレクトリーを削除します。ログおよびその他のいくつかの情報（ステップの成功および失敗状況）は、データベース内に残ります。ジョブ・レコードは「ジョブ」 > 「アーカイブ」ページに移動します。

### 日数\* (Days\*)

この日数が経過するまでジョブを保存しておく必要があります。

### カウント\*

最大許容ジョブ数です。ジョブの数が「カウント」の値を超えると、システム・スケジュールでジョブがバージされ、余分なビルドが削除されます。デフォルト値である「無制限」では、存在するジョブの数のためにジョブが削除されることはありません。

**条件** 「条件」プロパティには、ジョブを削除するにあたって、事前に満たされている必要のある追加条件を設定します。次のオプションがあります。

### いずれかのビルド

このオプションが選択されている場合、「条件」プロパティはジョブの削除に影響しません。

### 失敗のみ

失敗したジョブのみが削除されます。

### 成功のみ

成功したジョブのみが削除されます。

### 1 パスの保持

他の削除基準を満たしている場合にも、直近の成功したジョブは必ず保存されます。

\*「日数」または「カウント」の値の**いずれかが** 設定値を超えた場合に、ジョブを削除します。例えば、「カウント」が 10、「日数」が 2 に設定されている場合に、8 個のジョブがあるが、そのうち 3 個が 2 日間を超えていると、それら 3 つのジョブが削除されます。同様に、12 個のジョブがあり、すべて経過日数が 2 日間未満である場合は、最も古いジョブ 2 つが削除されます。

ページ・ジョブを実行すると、システムはジョブをアーカイブし、クラスの設定値に従ってデータを削除します。

### イベント発生時にプロジェクトを起動するためのプロパティ:

クラスに関連した特定のイベントが発生したときに、プロジェクトを起動 (チェーニング) することができます。これらのプロパティを使用することにより、プロセス内の状態の進行をモデル化することができます。

以下のクラスのプロパティを設定すると、特定のイベントが発生したときにジョブを起動することができます。

#### 削除時に開始

このプロパティは、クラス内のなんらかのジョブが削除されたときに（つまり、システムがこのクラスのジョブに関してパージ・ジョブを開始したときに）、指定されたプロジェクトを起動します。このプロパティを使用すると、パージに伴って自動的に削除されない一部の特定ファイルが削除されるようになります。

#### 設定時に開始

このプロパティは、ジョブのクラス・プロパティがこのクラスに変更されたときに、指定されたプロジェクトを起動します。このプロパティを使用すると、プロセスをジョブの再分類と結び付けることができます。例えば、Test クラスを作成して、ジョブが Test クラスにプロモートされたときになんらかの標準テストを起動することができます。

#### 終了時に開始

このプロパティは、ジョブのクラス・プロパティがこのクラスから別のクラスに変更されたときに、指定されたプロジェクトを起動します。

これらのプロパティは、プロジェクトをチェーンとして起動します。

**注：**「変更が必要」変数を含むジョブがこれらのプロパティによって起動された場合、変数値は変更されません。

---

## 通知のセットアップ

システムは、プロジェクトまたはステップが成功または失敗したとき、あるいは特定の別イベントが発生したときに、E メール通知を送信できます。このセクションでは、E メール通知の構成方法と、E メール通知の外観を制御する通知テンプレートの変更方法を説明します。

通知はアクセス・グループに送信されるので、通知と同様にセキュリティも念頭に置いて、アクセス・グループを設計してください。通知は、個々のユーザーに直接ではなく、常にグループに対して送信されますが、必要があれば、ユーザーが 1 人しかいないグループもセットアップできます。

アクセス・グループを階層状（アクセス・グループにサブグループが含まれている）にセットアップした場合、通知は以下のように機能します。

- 通知のアクセス・グループが、1 人以上のユーザーを含む親グループである場合、通知は、その親グループ内のユーザーにのみ送信されます。
- 通知のアクセス・グループが、ユーザーは含まず、サブグループのみを含む親グループである場合、通知は、すべてのサブグループ・ユーザーに送信されます。

通知イベントを作成するには、通知プロパティのアクセス・グループを選択します。

- プロジェクトの場合、開始通知、成功通知、および失敗通知のグループを選択できます。
- ステップの場合、成功通知および失敗通知のグループを選択できます。



これらのプロパティに対してアクセス・グループが選択された場合、該当するイベントが発生すると、システムは、必ずそのグループ・メンバーに E メールを送信します。これらの E メール・メッセージにおける時刻の参照は、通知を送る管理コンソールのタイム・ゾーンに基づいて行われます。

プロジェクトが、インライン・プロジェクトとして別のプロジェクトを含む場合、インライン・プロジェクトの開始通知、成功通知、および失敗通知の設定は無視されますが、そのステップの通知設定は守られています。詳しくは、394 ページの『インライン化されたプロジェクトの通知』を参照してください。

通知を使用できるようにするには、以下を行う必要があります。

- **SMTP** サーバー・システム設定を構成して、Eメールの送信に使用する **SMTP** サーバーをシステムが判別できるようにします。デフォルトは `localhost` です。また、「**システム・アラート・ソース**」に関するシステム設定も必要な場合があります。このアドレスはソース・アドレスとして使用されます。ほとんどの **SMTP** サーバーには有効なソース・アドレスが必要です。デフォルトは `root@localhost` です。
- 1 つ以上の通知グループを作成し、通知グループにユーザーを割り当てます。
- 個々のプロジェクトまたはステップ (あるいはプロジェクトとステップの両方) について、通知するグループを選択します。

さらに、通知テンプレートを編集して、システムが送信する通知 E メールを構成できます。390 ページの『通知テンプレートのカスタマイズ』を参照してください。

## 通知テンプレートについて

通知テンプレートは、システムで発生したイベントに関して、カスタマイズされたメッセージを送信するために使用されます。

### 用意されているテンプレート・タイプ

システムには、次のような通知テンプレート・タイプのセットが含まれています。

- 変更成果物マッチング・メッセージ
- フィルター・マッチング・メッセージ
- ジョブ中断メッセージ
- ジョブ失敗メッセージ
- ジョブ成功メッセージ
- ジョブ開始メッセージ
- ジョブ警告メッセージ
- パージ失敗メッセージ
- ステップ Eメール・メッセージ
- ステップ失敗メッセージ
- ステップ成功メッセージ
- ステップ警告メッセージ
- システム・アラート・メッセージ

各タイプには一連の翻訳済みメッセージが含まれています。メッセージは、コンテキストを提供するために変数を使用します。

### グローバルな使用



提供されているテンプレートは、次のようにグローバルに使用します。

- プロジェクト: 各プロジェクト内のプロパティを使用して、プロジェクトの開始、成功、および失敗の通知タイプを通知できます。ジョブが警告を出して終了している場合、ジョブ警告の通知が行われます。
- ステップ: 各ステップ内のプロパティを使用して、成功および失敗の通知タイプを通知できます。ステップの失敗後にジョブが続行されている場合、ステップ警告の通知が行われます。
- その他の通知タイプには以下があります。
  - 変更成果物マッチング・メッセージ - ユーザーに関連する成果物が変更されました。
  - ジョブ .break メッセージ - ジョブで、.break コマンドを含むステップが検出されました。
  - ステップ .email メッセージ - ステップが .email コマンドを実行します。
  - システム・アラート・メッセージ

### トランスレーションに保管されるメッセージ

使用される通知メッセージは、テンプレート・タイプに関連したトランスレーションに保管されます。それぞれのトランスレーションでは、その通知メッセージに関する情報が指定されます。

- 言語
- 説明
- タイトル (通知 E メールの件名として使用されます)
- 本文 (通知 Eメールのメッセージ本文として使用されます)

用意された通知テンプレートおよびユーザーが作成する通知テンプレートの両方で、トランスレーションを編集することができます。

### メッセージ・フォーマット

メッセージは、次の 2 つのいずれかのメッセージ・タイプで送信できます。

- プレーン・テキスト・メッセージ (デフォルト)
- MIME - 本文フィールドに特定の HTML タグが含まれている通知は、自動的に MIME メッセージとして送信されます。391 ページの『本文フィールドでの HTML の使用』を参照してください。

### テンプレート内の変数

システムは、テンプレートを構文解析して、いくつの変数があるかどうか調べます (392 ページの『環境変数およびレジスター変数の使用』を参照してください)。

### 特定のプロジェクトおよびステップ用のカスタム・テンプレート

用意された通知タイプに基づいて、カスタマイズされた通知テンプレートを作成できます。カスタマイズされたテンプレートは、プロジェクト、またはプロジェクト内の個別ステップに関連付けられている必要があります。390 ページの『特定のプロジェクトおよびステップ用の新規テンプレートの作成』を参照してください。

## 「テンプレート」パネルについて

「テンプレート」パネルは、イベントに関する通知をセットアップする場合に使用します。パネルを表示するには、「プロジェクト」 > 「テンプレート」を選択します。

The screenshot shows the 'Templates' panel in the IBM Rational Build Forge interface. At the top, there are tabs for 'UI Config', 'Console', 'Reports', and 'Logout: Root User'. Below these is a 'Templates' section with an 'Add Template' button. A filter bar shows 'Showing 1 - 4 of 13' and a 'Display All' link. The main area lists four templates: 'Change Artifact Match Message', 'Filter Match Message', 'Job .break Message', and 'Job Fail Message'. Below the list is a 'Details' section with three dropdown menus: 'Type' (set to 'Job .break Message'), 'Project' (set to 'HelloWorld'), and 'Step' (set to '- All -'). There are also buttons for '(New Template)', 'Save', and 'Delete'.

Rational® Build Forge® に用意されているすべてのテンプレートがパネルにリストされます。新規テンプレートを作成する場合は、以下のフィールドに情報を入力します。

**タイプ** メッセージのタイプ。既存のテンプレート・タイプを選択します。

**プロジェクト**

メッセージを送信するプロジェクトを指定します。

**ステップ**

メッセージを送信するプロジェクト内にあるステップを指定します。

## 「トランスレーション (Translations)」パネルについて

テンプレートで使用するメッセージをセットアップするには、「トランスレーション (Translations)」パネルを使用します。このパネルを表示するには、「プロジェクト」 > 「テンプレート」と選択して、リストからテンプレートを選択してください。そのテンプレートのトランスレーションのリストが表示されます。リスト内の各項目には、以下の情報が表示されます。

**言語** これは必須です。メッセージの言語。

**説明** リスト内でトランスレーションのタイトルとして使用されます。

**記入者** オプション: メッセージの「記入者」フィールドで使用する E メール・アドレス。以下の E メール・アドレス・フォーマットがサポートされています。

```
username@host.com
name username@host.com
name "<username@host.com>"
name <username@host.com>
name "username@host.com"
```

説明:

`username` は、E メールของผู้ใช้名です。

`name` は、任意のストリング (通常、何らかの形で名前を表すもの) です。

#### タイトル

必須: メッセージの「タイトル」フィールドで使用されるストリング。これには、変数を含めることができます。

本文 必須: メッセージの本文。これには、変数を含めることができます。

## SMTP サーバーの構成

SMTP サーバーを構成するには、「管理」 > 「システム」 > 「SMTP サーバー」を選択します。SMTP サーバーの値を編集するためのパネルが表示されます。サイトの SMTP サーバー名を入力します。デフォルトは `localhost` です。

また、「システム・アラート・ソース」パラメーターの設定が必要になることもあります。ほとんどの SMTP サーバーには有効なソース・アドレスが必要であり、このパラメーターの値がソース・アドレスとして使用されるためです。デフォルトは `root@localhost` です。

詳しくは、262 ページの『システム構成の設定』を参照してください。

## プロジェクトおよびステップの通知プロパティの設定

通知に関する SMTP およびグループの構成が適切に行われていると、プロジェクトおよびステップを構成して、特定のイベントが発生した際に通知を送信することができます。

- プロジェクトの場合、開始通知プロパティ、成功通知プロパティ、および失敗通知プロパティを設定できます。これらはプロジェクト・プロパティです。
- ステップの場合、成功通知プロパティまたは失敗通知プロパティを設定できます。これらはステップ・プロパティです。

単一ステップで複数の通知を設定してある場合、最初の一致のみが E メール通知を生成します。

## 通知の演習

次の手順は、E メール通知をセットアップして試行する方法を示しています。この演習には、SMTP サーバーと E メール・アカウントが必要です。

1. 『SMTP サーバーの構成』で説明されているように、SMTP サーバーをセットアップします。

アクセスできる E メール・アカウントに E メールを送信するための、ユーザー・アカウントを所有していることを確認します。

2. 「管理」 > 「アクセス・グループ」を選択します。
3. 選択したユーザーを「初期メンバー (Initial Member)」として含む、「Email Test」という新規アクセス・グループを作成します。
4. プロジェクト (例えば、Hello World プロジェクト) を選択し、そのプロジェクト・プロパティを編集します。「開始通知」フィールド、「成功通知」フィールド、および「失敗通知」フィールドで「Email Test」グループを選択します。

5. プロジェクトを実行します。
6. 2 つの E メール (プロジェクトの開始を示す E メールと、その成功または失敗を示す E メール) を受信したことを確認します。E メールを受信しなかった場合は、正しい SMTP サーバー値を使用したかを確認してください。

## 通知テンプレートのカスタマイズ

次のように通知テンプレートをカスタマイズすることができます。

- プロジェクトおよびステップのカスタム・テンプレートの作成: 個々のプロジェクトおよびプロジェクト内の個々のステップ用に、カスタム・テンプレートを作成することができます。カスタム・テンプレートは、用意されているテンプレート・タイプを使用して作成します。作成したカスタム・テンプレート内でトランスレーションを作成することができます。
- 用意されているテンプレートのトランスレーションの編集: 用意されている各通知テンプレート内でトランスレーションを追加または編集することができます。用意されているテンプレートの基本特性を編集することはできません。

### 特定のプロジェクトおよびステップ用の新規テンプレートの作成

システムには、システムに発生しうる多くのイベント用にテンプレートが付属しています。特定のプロジェクト、またはプロジェクト内の特定のステップに固有の新規テンプレートを作成できます。このプロジェクトまたはステップは、カスタム通知を作成する前にシステム内に作成されている必要があります。

新規テンプレートを作成するには、次の手順を実行してください。

1. 「プロジェクト」 > 「テンプレート」と選択します。システムは現在のテンプレートのリストを表示します。
2. 「テンプレートの追加」をクリックします。
3. テンプレートに関する情報を指定します。
  - **タイプ:** 新規テンプレートの基礎として使用する、既存の通知タイプを選択します。
  - **プロジェクト:** 新規テンプレートを適用するプロジェクトを選択します。テンプレートは、選択したプロジェクトを実行したときに生成される、通知メッセージに対してのみ使用されます。

**注:** プロジェクトに複数のスナップショットが定義されている場合は、スナップショットのリストが表示されます。使用するスナップショットを選択してください。通知は、そのスナップショットを使用して実行されたプロジェクトにのみ適用されます。「デフォルト・スナップショット」を指定した場合は、デフォルトとして割り当てられたスナップショットが使用されます。

- **ステップ:** 特定のステップを選択できます (この結果、そのステップに関する通知に対してのみ、テンプレートが適用されます)。または、「プロジェクト・イベント (Project Events)」を選択して、選択したプロジェクトに関するすべての通知にテンプレートを適用することもできます。
4. 「保存」をクリックします。
  5. テンプレートのトランスレーションを作成します。
    - a. 「プロジェクト」 > 「テンプレート」からテンプレートを選択します。

- b. 「**トランスレーションの追加**」をクリックします。
- c. テンプレートに関する情報を指定します。
  - **言語:** 必須 - メッセージで使用する言語。
  - **説明:** 必須 - このフィールドは、トランスレーションのタイトルとして使用されます。
  - **記入者:** オプション - メッセージの「記入者」フィールドで使用する E メール・アドレス。
  - **タイトル:** 必須 - メッセージの「タイトル」フィールドで使用するストリング。これには、変数を含めることができます。
  - **本文:** 必須: メッセージの本文。これには、変数を含めることができます。特定の HTML タグを含むメッセージは、プレーン・テキストではなく MIME メッセージとして送信されます。
- d. 「**保存**」をクリックします。
- e. 必要に応じて、他のトランスレーションについてもこの処理を繰り返します。

## 通知テンプレート内のトランスレーションの編集

各通知テンプレートは、その通知のメッセージを保管するためにトランスレーションのリストを使用します。

用意されている通知テンプレート、または個々のプロジェクトおよびステップについてユーザーが作成した通知テンプレートのいずれでも、トランスレーションを編集することができます。

トランスレーションを編集するには、以下のようになります。

1. 「**プロジェクト**」 > 「**テンプレート**」から、編集するテンプレートを選択します。トランスレーションのリストが表示されます。
2. 編集するトランスレーションをクリックします。
3. テンプレートに関する情報を指定します。
  - **言語:** 必須 - メッセージで使用する言語。
  - **説明:** 必須 - このフィールドは、トランスレーションのタイトルとして使用されます。
  - **記入者:** オプション - メッセージの「記入者」フィールドで使用する E メール・アドレス。
  - **タイトル:** 必須 - メッセージの「タイトル」フィールドで使用するストリング。これには、変数を含めることができます。
  - **本文:** 必須: メッセージの本文。これには、変数を含めることができます。特定の HTML タグを含むメッセージは、プレーン・テキストではなく MIME メッセージとして送信されます。
4. 「**保存**」をクリックします。

## 本文フィールドでの HTML の使用

通知は、デフォルトではプレーン・テキストとして送信されます。以下の HTML タグまたはタグの一部のうちのいずれかを使用する場合、通知は MIME メッセージとして送信され、すべての HTML タグはインタープリットされます。

```

<html
<body
<a
<b>
<p>
<table

```

トランスレーションを作成すると、システムが本文フィールドを構文解析します。その時点で、使用するフォーマットが決定されます。システムは、リストされたタグまたはタグの一部のみをスキャンします。

スキャン対象の要素をまったく含まないメッセージは、プレーン・テキストとして送信されます。そのメッセージ内の HTML タグはインタープリットされません。それらのタグはメッセージに組み込まれます。

## 環境変数およびレジスター変数の使用

`${VAR}` 構文を使用するかぎり、通知テンプレートで環境変数（標準のシステム変数のほかに、ユーザー独自に定義したもの）を参照することができます。

また、通知テンプレートにプロジェクトのレジスター変数を組み込むこともできます。空のレジスターを参照すると、システムは空ストリングを返します。

## 特殊な通知テンプレート変数

下の表に、通知テンプレートで使用可能な特殊変数を示します。変数の中には文脈依存のものがあり、それらは関連性がある場合にのみ使用できます（例えば、STEPNAME 変数はプロジェクト・レベルの通知には設定されず、ステップ・レベルの通知のみを対象とします）。

変数	内容
ACTION	ビルド・パージの場合、実行された削除のタイプを記述する。
BID	ジョブ ID 番号を記載する。レポートにアクセスするための、管理コンソールへ戻るリンクを構成するために使用します。
CONSOLEHOST	管理コンソール・コンピューターのホスト名。
CONSOLEPORT	管理コンソールで使用されるポート番号。これにより、通知テンプレート内で有効な URL を構成できます。
CONTEXTLOGLINKS	先頭が「FILT:」のログの行を、エントリーごとに 3 行のコンテキストでリストする。システムは、このメッセージに、管理コンソールのログ・エントリーへのリンクを記述します。

変数	内容
DURATION	ステップの場合、ステップとそのステップがインライン化したすべてのステップの実行時間 (秒) を記載する。
EID	環境 ID 番号を記載する。レポートにアクセスするための、管理コンソールへ戻るリンクを構成するために使用します。
FULLNORMALLOG	ジョブのステップごとに、ログ情報を表示する。ただし、詳細なログに表示される環境セットアップ・アクションは除外されます。
LINK	ビルドの場合、リンク名を記載する。
MESSAGE	失敗またはアラート・メッセージのエラーまたはメッセージ・テキストを含む。
ONFAIL	ステップの場合、そのステップの継続プロパティを保持する。
PATH	ステップの場合、必要に応じて、サーバーまたはステップのようなデータ項目のパスを記載する。
PID	プロジェクト ID 番号を記載する。レポートにアクセスするための、管理コンソールへ戻るリンクを構成するために使用します。
PROJECTNAME	プロジェクトの名前を含む。
RUNACTION	E メール・テンプレートを活用する変数を指定する。
SELECTOR	ステップまたはプロジェクトのセクター名を含む。
SERVER	ステップまたはプロジェクトのセクター名を含む。
SID	ステップの場合、ステップ ID 番号を記載する。レポートにアクセスするための、管理コンソールへ戻るリンクを構成するために使用します。
SRVRHOST	ステップの場合、ステップに対するサーバーの TCP/IP ホスト名を含む。
START	ジョブが開始した日付/時間を含む。



変数	内容
STEPNAME	ステップの場合、そのステップの名前を含む。
STEPNORMALLOG	ステップの場合、ジョブでの現在のステップに関するログ情報を表示する。ただし、詳細なログに表示される環境セットアップ・アクションは除外されます。
TAG	ジョブのタグ・ストリングを含む。\$BF_TAG と同じ値です。
TAILNORMALLOG	ステップの場合、STEPNORMALLOG のように機能するが、ログの最後のみを表示する。  ビルドの場合、FULLNORMALLOG のように機能するが、各ステップのログの最後のみを表示する。  表示される行数は、「メール・テンプレートのログ量のテール」システム設定で制御されます。
USEREMAIL	ジョブ/イベントの所有者の E メール・アドレスを含む。
USERNAME	ジョブ/イベントの所有者のフルネームを含む。

## インライン化されたプロジェクトの通知

システムは、インライン化されたプロジェクトの通知を、インライン化されたプロジェクトからのステップが呼び出し元プロジェクトに組み込まれている場合と同様に取り扱います。

- あるプロジェクトにインライン化されたプロジェクトが含まれる場合、インライン化されたプロジェクトのプロジェクト・レベルの通知設定は無視されます。プロジェクト A 内にプロジェクト B がインライン化されている場合、プロジェクト B の開始、成功、または失敗に関するメッセージは送信されません。
- ステップ・レベルの通知設定は影響を受けません。あるステップで成功通知または失敗通知アクセス・グループが設定されている場合、そのステップがトップレベル・プロジェクトであるかインライン化されたプロジェクトであるかに関係なく、該当のメッセージが送信されます。
- インライン化されたステップは、呼び出し側プロジェクトの成否を判別するために役立ちます。例えば、インライン化されたステップで障害が起こると、呼び出し元プロジェクトが失敗するか、呼び出し元プロジェクトの状態が「失敗しましたが続行されました」に変更されます (そのステップが「障害時に続行 (Continue On Failure)」に設定されている場合)。

---

## スナップショットを使用したプロジェクトの新規インスタンスの作成

変更が必要なプロジェクトの新規インスタンスを素早く作成するには、プロジェクトのスナップショットを作成します。プロジェクトのスナップショットは、別個の実行可能プロジェクトです。スナップショットは、ライブラリーの新規インスタンスの作成にも使用できます。

### プロジェクト・スナップショットの概要

以下のトピックを参照して、プロジェクトのスナップショットについて学習し、それらの使用法を理解してください。

### プロジェクト・スナップショットのユース・ケース

以下の例では、プロジェクト・スナップショットの一般的なユース・ケースをいくつか示します。

- 既存のプロジェクトでジョブの実行を続けながら、プロジェクト構成を変更したり、新しいツールまたはスクリプトのテストを行ったりするためにプロジェクトのスナップショットを作成する。
- 一時的なバックアップとして、または正式なアーカイブの一部としてプロジェクトのスナップショットを保管する。
- 外部または内部リリースなどの、マイルストーンに対応する特定時点プロジェクト構成を取り込むために、プロジェクトのスナップショットを作成する。

### プロジェクト・スナップショットの概念と用語

ユーザー・インターフェースでは、スナップショットを作成すると、プロジェクトで使用されるいくつかの新しい概念および用語が紹介されます。


**プロジェクト・スナップショット:** スナップショットは、既存のプロジェクトの新規インスタンスです。スナップショットに関して注意すべきいくつかのキーポイントを次に示します。

- スナップショットは独立したプロジェクトです。スナップショット・セット内の1つのスナップショットを変更しても、そのセット内の他のスナップショットは変更の影響を受けません。
- スナップショットは実行可能なプロジェクトです。これは、プロジェクト・スナップショットの作成時にスナップショットの対象として選択したオブジェクトに対して、またはソース・プロジェクトに関連するオブジェクトに対して、実行されます (このスナップショットは基本スナップショットとも呼ばれます)。
- スナップショットはコピーではありません。

プロジェクトに関連したオブジェクトのスナップショットを作成すると、そのオブジェクトの別個のインスタンスが作成されます。プロジェクトをコピーした場合には、オブジェクト間の関係がコピーされ、セレクター、環境、インライン・プロジェクト、またはチェーニングされたプロジェクトの新規インスタンスは作成されません。

- スナップショットはプロジェクトの改訂ではありません。
  - スナップショットは、2つのプロジェクト・スナップショット間の変更内容の比較をサポートしません。

- プロジェクト・スナップショットに対して行われた変更は、ソース制御システムなどの場合と異なり、バージョン番号を使用して追跡または識別されることはありません。ただし、7.5.0 や 3.4.01 などのバージョン番号を含むスナップショット命名体系を使用して、プロジェクト・スナップショットをマイルストーンに関連付けることができます。

**スナップショット・セット:** スナップショット・セットは、1 つの基本スナップショットから派生したすべてのプロジェクト・スナップショットの集合です。このセットには少なくとも、基本スナップショットまたは親スナップショットと、子スナップショットが含まれます。UI では、プロジェクト名の隣にある「スナップショット」アイコン  により、そのプロジェクトに関するスナップショット・セットが作成されたことが示されます。


**基本スナップショット:** 最初は、すべてのプロジェクトのスナップショット名は「Base Snapshot」になっています。Base Snapshot は別の名前に変更できます。基本スナップショットは、スナップショット・セットの親です。

**デフォルト・プロジェクト・スナップショット:** デフォルト・プロジェクト・スナップショットは、現行の作業中のプロジェクトです。セット内の 1 つのスナップショットのみをデフォルトにすることができます。デフォルト・スナップショットを指定しなかった場合、基本スナップショットがデフォルトになります。

- UI では、デフォルト・スナップショットはプロジェクト・リストの最上位に表示されます。「プロジェクト」または「ジョブ」>「開始」を選択して、プロジェクト・リストを表示します。
- スナップショット付きのプロジェクトをインライン・プロジェクトまたはチェーニングされたプロジェクトとして選択すると、リスト・ボックスで別のプロジェクト・スナップショットを選択しないかぎり、デフォルト・プロジェクト・スナップショットが使用されます。
- プロジェクト・スナップショット・セット内の他のスナップショットにアクセス

して操作するには、「スナップショット」アイコン  をクリックする必要があります。

## プロジェクト・スナップショットのビュー

「スナップショット」ビューを表示するには、「スナップショット」アイコン  を選択します。UI の「スナップショット」ビューには、セット内のスナップショットの階層が次のように表示されます。

- 基本スナップショットが最上位にあります。固有の名前を割り当てていない場合、これには Base Snapshot という名前が付いています。
- すべてのプロジェクト・スナップショットは基本スナップショットの子です。同じ基本スナップショットの子は、「スナップショット」列の同じレベルにインデントされています。
- 子スナップショットから作成されたプロジェクト・スナップショットは、その子スナップショットの子になり、「スナップショット」列でその次のレベルにインデントされます。

## プロジェクト・スナップショットの計画

デフォルト・プロジェクト・スナップショットを選択してプロジェクト・スナップショットの名前を付けるために、いくつかのベスト・プラクティスを検討してください。

### • セット内のデフォルト・スナップショットの選択方法

UI は、1 つのスナップショット・セットごとに 1 つのデフォルトまたは現行プロジェクト・スナップショットしか認識しません。次のような一貫性のある方法で、デフォルト・スナップショットを選択してください。

- 基本スナップショットをデフォルト・スナップショットとして使用する

この方針を使用する場合、特定時点のバックアップとしてスナップショットを作成し、バックアップとして取られたプロジェクト・スナップショットには変更を行いません。基本スナップショットに対して変更を加えながら、基本スナップショット・プロジェクトのみを使用してジョブの実行を続けます。

- 最新のスナップショットをデフォルト・スナップショットとして使用する

この方法では、新規のプロジェクトを作成する際に、それを新規のデフォルト・プロジェクト・スナップショットにします。基本スナップショットや前のプロジェクト・スナップショットに変更は行いません。ジョブの実行には、最新のスナップショットが使用されます。

### • セットのスナップショット命名体系を識別する

プロジェクト・スナップショット名は、プロジェクト・スナップショット・セット内で固有でなければなりません。

プロジェクト・スナップショット名を作成するためには、以下の基準に従ってください。

- 説明的な名前にし、スナップショットの使用法や目的が分かるようにします。
- 命名体系が、定義された標準に従うようにします。「スナップショット」タブの「コメント」ボックスを使用して、命名体系を説明することができます。

### • セットで単一のプロジェクト名を使用する

プロジェクト・スナップショットを作成した後で、プロジェクトの名前を変更することができます。プロジェクト名を変更すると、各プロジェクト・スナップショットの名前が更新されます。

## プロジェクト・スナップショット・オプション

スナップショットを作成するときには、スナップショットに含めるオブジェクトを選択する必要があります。下の表に、Build Forge オブジェクトで使用可能なオプションを示します。表には、以下のオブジェクトに対する列があります。

- プロジェクト・スナップショットに自動的に含まれるオブジェクト。
- プロジェクト・スナップショットの作成時に選択した場合にオプションとして作成され、含まれるオブジェクト。

それらのオブジェクトについて、プロジェクト・スナップショットと同じスナップショット名の新規オブジェクトが UI に作成されます。例えば、プロジェク

ト・スナップショットの名前が release\_7.1 であれば、環境変数、セクター、インライン・プロジェクトまたはライブラリー、およびチェーニングされたプロジェクトまたはライブラリーのスナップショット名も release\_7.1 になります。

- プロジェクト・スナップショットに含まれないオブジェクト。手動でオブジェクトを作成して、プロジェクトに追加する必要があります。

	オプションとしてプロジェクト・スナップショットに含まれるもの	
プロジェクト・スナップショットに自動的に含まれるもの	* はコピーのみで、別個のインスタンスは作成されない	プロジェクト・スナップショットに含まれないもの
プロジェクト・ステップおよびそれらの設定 (ログ・フィルター、通知グループなど)	プロジェクトとそのステップの環境変数	プロジェクト・メモ
プロジェクト・タグ	スナップショット環境変数の「組み込み」タイプの環境変数によって追加される環境変数	
	インライン・プロジェクトまたはインライン・ライブラリーと、そのステップ	
	チェーニングされたプロジェクトまたはチェーニングされたライブラリーと、そのステップ	
	プロジェクトとそのステップのセクター	
	スナップショット・セクターの「組み込み」タイプのセクター・プロパティーによって追加されるセクター	
	* プロジェクト・レジスター (コピー)	
	* プロジェクト・タグ変数値 (コピー)	
	* 通知用のテンプレート (コピー)	
	* アダプター・リンク (コピー)	

## スナップショット権限のためのアクセス・グループの検証および編集

ユーザーにスナップショットの作成に必要な権限、およびデフォルト・スナップショットの設定に必要な権限があることを検証します。ない場合は、アクセス・グループを使用して、ユーザーに権限を割り当てます。

権限は、アクセス・グループを通じてユーザーに割り当てられ、このアクセス・グループは、Build Forge が提供するか、Build Forge 管理者が作成することができます。

スナップショット権限に割り当てられたアクセス・グループを検証および編集するには、以下の手順を実行します。




1. 「管理」>「権限」を選択します。
2. 権限リストから「すべてを表示」を選択して、すべての権限をリストします。
3. 適切なアクセス・グループおよびユーザーが、以下のスナップショット権限にアクセスできることを検証します。

スナップショットの作成	プロジェクト、環境変数、およびセクター用のスナップショットの作成に必要なユーザー権限。
デフォルトのスナップショットの設定	プロジェクト、環境変数、およびセクター用のデフォルト・スナップショットの設定または変更に必要なユーザー権限。

## 既存プロジェクトまたは既存プロジェクト・スナップショットからのプロジェクト・スナップショットの作成

プロジェクトのスナップショットを作成すると、プロジェクトと、スナップショットの対象として選択したオブジェクトの新規インスタンスが作成されます。スナップショットはコピーではありません。これは、プロジェクトの新しい実行可能インスタンスです。

注: 管理コンソールには、最初の 2999 個のプロジェクトより後にあるプロジェクトのスナップショットは表示されません。

1. スナップショットを作成するプロジェクトまたはプロジェクト・スナップショットの横にある「編集」アイコンをクリックします。
  - デフォルト・プロジェクト・スナップショットのスナップショットを作成するには、プロジェクトのリスト（「プロジェクト」）で、最上位のスナップショットの横にある「編集」アイコン  をクリックします。
  - デフォルト以外のプロジェクト・スナップショットのスナップショットを作成するには、「スナップショット」アイコン  をクリックします。スナップショット・ビューに、プロジェクト・スナップショットがセットで表示されます。デフォルト以外のプロジェクト・スナップショットの横にある「編集」アイコン  をクリックします。
2. 「新規スナップショットの作成」をクリックします。
3. 「スナップショット」タブの「名前」にスナップショット名を入力します。

名前は、プロジェクト・スナップショット・セット内で固有のものにする必要があります。この名前は、プロジェクトでスナップショットを作成するすべてのオブジェクトに割り当てられます。

4. プロジェクト・スナップショットを作成するときにスナップショットを作成する Build Forge オブジェクトを選択します。選択可能なオブジェクトについては、下の表に説明があります。



オブジェクト	説明
デフォルト	<p>UI で、デフォルトのプロジェクト・スナップショットがプロジェクト・リストのトップレベルに表示されます。</p> <p>「プロジェクト」または「ジョブ」&gt;「開始」を選択して、プロジェクト・リストを表示します。</p>
プロジェクト環境変数の組み込み	プロジェクトおよびプロジェクト内のステップ環境変数のスナップショットを作成します。
含まれる環境変数に従う	「プロジェクト環境変数の組み込み」が選択されている場合、「組み込み」タイプの環境変数によって組み込まれるその他すべての環境変数のスナップショットも作成します。
プロジェクト・セクターの組み込み	プロジェクトおよびプロジェクトに組み込んだステップ・セクターのスナップショットを作成する。
含まれるセクターに従う	「プロジェクト・セクターの組み込み」が選択されている場合、「組み込み」タイプのセクター・プロパティによって組み込まれるその他すべてのセクターのスナップショットも作成します。
プロジェクト・アダプター・リンクの複製	<p>アダプター・リンクをスナップショットの一部としてコピーします。</p> <p>アダプター・リンクは、プロジェクトにアダプターを追加します。アダプターは、プロジェクト内の最初のステップ (ステップ 0) として実行されます。</p>
プロジェクト・レジスターの複製	プロジェクト・レジスターをスナップショットの一部としてコピーします。
プロジェクト・タグ変数値の複製	プロジェクト・タグ変数のタグ値をコピーします。タグ変数は自動的にコピーされますが、その値はコピーされません。タグ値をコピーしなければ、1 にリセットされます。
プロジェクト・テンプレートの複製	プロジェクト・レベルとステップ・レベルで設定される成功と失敗の各通知イベント用の通知テンプレートをコピーします。
チェーニングされたプロジェクトの組み込み	<p>プロジェクト・レベルまたはステップ・レベルで参照される、チェーニングされたプロジェクトまたはライブラリーと、そのステップのスナップショットを作成します。</p> <p>チェーンは、プロジェクトの成功/失敗またはステップの成功/失敗の条件によって起動されます。</p>
プロジェクト・インラインの組み込み	<p>ステップ・レベルで参照される、インライン・プロジェクトまたはライブラリーと、そのステップのスナップショットを作成します。</p> <p>インラインは、ステップによって起動され、ステップの完了後に実行されます。</p>



5. 「保存」をクリックして、プロジェクト・スナップショットを保存します。

## デフォルトのプロジェクト・スナップショットの変更

デフォルトのプロジェクト・スナップショットは、プロジェクト・スナップショット・セット内のトップレベルのスナップショットで、プロジェクト・リスト（「プロジェクト」）に表示されます。



デフォルトのプロジェクト・スナップショットを変更するには、以下のようにして、新しいデフォルトにするスナップショットのスナップショット定義を編集します。

1. 「プロジェクト」を選択します。
2. プロジェクト・リストで、デフォルトのプロジェクト・スナップショットの「スナップショット」アイコン をクリックします。
3. スナップショット・リストで、新規デフォルトとして設定するプロジェクト・スナップショットの「編集」アイコン をクリックします。
4. 「デフォルトの設定」をクリックします。
5. **重要:** ポップアップで「OK」または「取り消し」を選択します。



OK	参照を更新する: 前のデフォルトを参照しているオブジェクトがある場合に、前のデフォルトのプロジェクト・スナップショットから新しいデフォルトへ参照を更新します。
取り消し	参照を更新しない: 前のデフォルトを参照しているオブジェクトがある場合に、新しいデフォルトのプロジェクト・スナップショットへ参照を更新しません。

## プロジェクト・スナップショットのスナップショット名の変更

プロジェクト・スナップショットのスナップショット名、およびプロジェクト・スナップショットの作成時にスナップショットの対象として選択したオブジェクトのスナップショット名も変更できます。

基本スナップショットの場合、このオプションを使用して、単一のプロジェクト・スナップショットのみ、または現在と将来のすべてのプロジェクトについて、基本スナップショットのデフォルト名を別のスナップショット名に変更できます。

スナップショット名を変更するには、次の手順に従ってください。

1. 「プロジェクト」を選択します。
2. プロジェクト・リストで、デフォルトのプロジェクト・スナップショットの「スナップショット」アイコン をクリックします。
3. スナップショットのリストで、プロジェクト・スナップショットの「編集」アイコン をクリックします。
4. 「スナップショット」タブを選択します。
5. 「名前」に新規名を入力します。
6. **オプション:** 「コメント」にコメントを入力します。
7. **重要:** ポップアップで「OK」または「取り消し」を選択します。

OK	<p>プロジェクト・スナップショット名およびその他のスナップショット・オブジェクト名を変更する: プロジェクト・スナップショットの作成時にスナップショットの対象として選択したオブジェクトについて、それらのオブジェクトおよびプロジェクト・スナップショットの名前を変更します。</p> <p>基本スナップショットの場合: 現在のすべてのプロジェクト・スナップショットおよび将来のすべてのプロジェクト・スナップショットについて、基本スナップショットの名前を変更します。</p>
取り消し	<p>プロジェクト・スナップショット名を変更するが、その他のスナップショット・オブジェクト名は変更しない: プロジェクト・スナップショットの作成時にスナップショットの対象として選択したオブジェクトについては、それらのオブジェクトの名前を変更しません。プロジェクト・スナップショット名のみを変更します。</p> <p>基本スナップショットの場合: 現在のすべてのプロジェクト・スナップショットおよび将来のすべてのプロジェクト・スナップショットについて、基本スナップショットの名前を保持します。</p>


## プロジェクト・スナップショット・セット内のスナップショットのアクセスと表示

プロジェクト・スナップショットを作成すると、少なくとも基本プロジェクト・スナップショットと新規プロジェクト・スナップショットを含むスナップショット・セットが作成されます。



スナップショット・セット内のすべてのプロジェクト・スナップショットを表示するには、次の手順を実行します。

1. 「プロジェクト」を選択します。

プロジェクト・リストには、プロジェクトとプロジェクト・スナップショットのリストが表示されます。最上位のスナップショットは基本プロジェクト・スナップショットです。

2. 「スナップショット」アイコン  をクリックすると、スナップショット・セット内のプロジェクト・スナップショットが表示されます。


「スナップショット」ビューでは、以下の作業を行うことができます。

- 新規プロジェクト・スナップショットを作成できます。「編集」アイコン  をクリックしてください。
- プロジェクトのデフォルト・スナップショットを変更できます。「編集」アイコン  をクリックしてから「デフォルトの設定」をクリックしてください。
- 標準プロジェクトの場合と同様にプロジェクト・スナップショット定義を編集できます。

## デフォルトのプロジェクト・スナップショットのジョブの開始

「クイック・スタート」アイコンまたは「プロジェクトの開始」ページを使用して、デフォルトのプロジェクト・スナップショットを開始します。

「クイック・スタート」アイコンを使用して、デフォルトのプロジェクト・スナップショットを開始するには、以下の手順を実行します。

1. 「プロジェクト」を選択します。
2. プロジェクト・リストで、「クイック・スタート」アイコン  をクリックします。

セクター、クラス、タグ形式、および環境変数のデフォルト値を使用してジョブが実行されます。

「プロジェクトの開始」ページを使用して、デフォルトのプロジェクト・スナップショットを開始するには、以下の手順を実行します。

1. 「ジョブ」→「開始」と選択します。
2. プロジェクト・リストで、デフォルト・スナップショットのプロジェクト名をクリックすると、「プロジェクトの開始」ページが表示されます。

「プロジェクトの開始」ページで、環境変数、セクター、クラス、およびタグ形式を変更できます。

セクターにセクターのスナップショットがあれば、「スナップショット」フィールドの下フィールドに、選択できるセクターのスナップショットがリストされます。


**注:** デフォルトのセクターのスナップショットを素早く選択するには、「デフォルト・スナップショット」を選択します。「デフォルト・スナップショット」では、デフォルトのセクターのスナップショットの名前が割り当てられます。

3. 「実行」をクリックします。


## デフォルト以外のプロジェクト・スナップショットのジョブの開始

「クイック・スタート」アイコンまたは「プロジェクトの開始」ページを使用して、デフォルト以外のプロジェクト・スナップショットを開始できます。

「クイック・スタート」アイコンを使用して、デフォルト以外のプロジェクト・スナップショットを開始するには、以下の手順を実行します。


1. 「プロジェクト」を選択します。
2. プロジェクト・リストで、基本スナップショットの「スナップショット」アイコン  をクリックします。

スナップショット・ビューに、プロジェクト・スナップショットがセットで表示されます。

3. デフォルト以外のプロジェクト・スナップショットの横にある「高速開始 (Fast Start)」アイコン  をクリックします。

セレクトター、クラス、タグ形式、および環境変数のデフォルト値を使用してジョブが実行されます。

「プロジェクトの開始」ページを使用して、デフォルト以外のプロジェクト・スナップショットを開始するには、以下の手順を実行します。

1. 「ジョブ」→「開始」と選択します。
2. プロジェクト・リストで、基本スナップショットの「スナップショット」アイコン  をクリックします。

スナップショット・ビューに、プロジェクト・スナップショットがセットで表示されます。

3. デフォルト以外のプロジェクト・スナップショットのプロジェクト名をクリックすると、「プロジェクトの開始」ページが表示されます。

「プロジェクトの開始」ページで、環境変数、セレクトター、クラス、およびタグ形式を変更できます。

セレクトターにセレクトターのスナップショットがあれば、「スナップショット」フィールドの下フィールドに、選択できるセレクトターのスナップショットがリストされます。

注: デフォルト・セレクトター・スナップショットを素早く選択するには、「デフォルト・スナップショット」を選択します。「デフォルト・スナップショット」では、デフォルトのセレクトターのスナップショットの名前が割り当てられます。


4. 「実行」をクリックします。

## プロジェクト・スナップショットの削除


「プロジェクトの削除」または「プロジェクトの強制削除 (Clobber Project)」オプションを使用して、プロジェクト・スナップショットを削除します。

- プロジェクト・スナップショットにジョブがない場合、およびプロジェクト・スナップショットが他のオブジェクトによって参照されていない場合は、「プロジェクトの削除」オプションが使用可能になります。プロジェクト・スナップショットは、クラスによって参照することも、インライン・プロジェクトまたはチェーン・プロジェクトとして他のプロジェクトによって参照することもできます。
- 「強制削除」オプションは、Build Forge データベースからプロジェクトとそのジョブを削除し、他のオブジェクトによるプロジェクトへの参照をすべて除去します。

プロジェクト・スナップショットを削除するには、以下の手順を実行します。

1. 「プロジェクト」を選択します。
2. プロジェクト・リストで、基本スナップショットの「スナップショット」アイコン  をクリックします。

スナップショット・ビューに、プロジェクト・スナップショットがセットで表示されます。

3. 削除するプロジェクト・スナップショットの横にある「編集」アイコン  をクリックします。
4. 「強制削除」または「プロジェクトの削除」をクリックします。



---

## 第 20 章 ステップの処理

このトピックでは、管理コンソールでのステップの作成と管理の方法について説明します。

---

### ステップについて

ステップは、プロジェクトのコンポーネントです。プロジェクトをジョブとして実行すると、各ステップが順番に実行されます。ステップには、1 つ以上のコマンドが含まれ、その動作に影響を及ぼすステップ・プロパティーがあります。

### 「ステップ」パネルについて

#### 「詳細」タブ

ステップ・プロパティーでは、ステップの実行方法、その出力の処理方法、およびステップ完了時に実行すべき操作を指定します。ステップでは、別のプロジェクトまたはライブラリーを実行することもできます。

ステップ・プロパティーを表示するには、プロジェクト内のステップを選択します。「詳細」タブはデフォルトで表示されます。ここにステップ・プロパティーが表示されます。

ステップ・プロパティーが明示的に設定されていない場合、その値はプロジェクトから継承されます。ステップに設定されたステップ・プロパティーによって、継承された値がオーバーライドされます。

ステップ・プロパティーには以下が含まれます。

**名前** ステップの名前。システムおよびログ内でステップのラベルとして使用されます。

#### アクティブ

ステップを実行するかどうかを指定します。デフォルトでは、ステップは有効です。ステップを実行しないようにするには、「無効」を選択します。無効なステップは、ジョブで実行することはできません。

#### ディレクトリー

ステップ・コマンドが実行される場所を設定します。システムは、ジョブごとに固有のディレクトリーを自動的に作成します。「ディレクトリー」フィールドにより、ジョブ中にプロジェクトが構成したディレクトリーでコマンドを実行することが容易になります。(Build Forge は、「ディレクトリー」フィールドに示されるディレクトリーを構成しません。)

**パス** 「ディレクトリー」を絶対パスにするか、相対パスにするかを指定します。

- **相対:** ステップ・コマンドは、サーバー、プロジェクト、ジョブおよびステップ・ディレクトリーと一緒に追加することで検出されるパスで実行されます。



- **絶対:** ステップ・コマンドは、サーバーおよびステップ・ディレクトリーを一緒に追加することで検出されるパスで実行されます。このオプションを使用すれば、プロジェクト・ディレクトリー構造にないディレクトリーにアクセスすることができます。例えば、サーバー上の常設アプリケーションを起動する場合に使用できます。

### ステップ・タイプ

ステップの実行方法を決定します。このプロパティーは、「コマンド」の内容および「インライン」で指定されたプロジェクト (ある場合) に影響します。

- **正規:** ステップは一度実行されます。
- **条件付き:** ステップは、「条件」プロパティーの式が `true` と評価された場合、一度実行されます。「条件付き」を選択すると、「条件」、「Else インライン」および「Else コマンド」の各プロパティーが表示されます。「条件」プロパティーで `false` と評価された場合は、コマンドとインラインは実行されません。代わりに、Else コマンドと Else インラインが実行されます (指定されている場合)。
- **While ループ:** ステップを複数回実行できます。「条件」プロパティーの式が `false` になるまで、または反復最大数に達するまで実行されます。「While ループ」を選択すると、「条件」と「最大反復」の各プロパティーが表示されます。

セレクトターは「While ループ」の各反復で評価され、反復に使用するサーバーが決定されます。

### インライン

現在のプロジェクトで、インラインで実行するプロジェクトまたはライブラリーを指定します。プロジェクトまたはライブラリーのステップは、現在のプロジェクトの環境変数と大部分のプロパティーを使用して実行されます。ただしシステムは、インラインのステップのデフォルトのセレクトターとして、そのインライン・プロジェクトのセレクトターを使用します。これは、指定したプロジェクトのステップが現行ステップの後にコピーされたように動作します。

### アクセス

アクセス・グループを選択して、ステップの使用が許可されるユーザーを定義します。このプロパティーを使用して、プロジェクト内にある特定ステップへのアクセスを制限できます。あるステップに関するアクセス・グループのメンバーでないユーザーが、そのステップを含むプロジェクトを開始した場合、ステップはスキップされます。

「プロジェクト・デフォルト」を選択すると、ステップはプロジェクトのアクセス・プロパティーを継承します。

### ステップ・プロバイダー

ステップ・コマンド実行の実装環境。デフォルトのステップ・プロバイダーは MJC ステップ・プロバイダーです。このプロバイダーは、エンドポイント・エージェント・マシンでコマンド・テキストを実行するという、期待どおりのレガシー動作を提供します。

### 最大反復

「ステップ・タイプ」が「While ループ」の場合にのみ表示されます。ル

ープ内でステップを反復して実行可能な最大数を指定します。システム指定のデフォルト値は 100 です。正常に完了 (成功) したステップはステップ・ログに表示されます。最大反復値に達した場合はステップが失敗するように設定するには、「最大に達した場合ステップを失敗する」を使用します。

ジョブの実行中、読み取り専用の `BF_ITERATION` という変数には、正しく入力された反復数が格納されます。ジョブを停止してから再始動した場合は、`BF_ITERATION` に格納された反復から再開されます。

#### 最大に達した場合ステップを失敗する

「はい」に設定すると、「最大反復」値に達した場合に「While ループ」ステップが失敗します。「いいえ」に設定すると、ステップは成功します。

#### Else インライン

「ステップ・タイプ」が「条件付き」の場合にのみ表示されます。指定した条件が `false` の場合はインラインで実行するようにプロジェクトを指定します。デフォルト値は「いいえ」です。

#### コマンド

1 つ以上のコマンドです。コマンドは、オペレーティング・システム・コマンド、ドット・コマンド、またはそれらを組み合わせたものにすることができます。414 ページの『ステップの実行方法』を参照してください。

**条件** ステップ・タイプに「条件付き」または「While ループ」を選択した場合にのみ表示されます。

- 条件付き: コマンドは、条件が `true` と評価された場合に実行されます。
- While ループ: コマンドは、条件が `true` と評価されている限り、複数回実行できます。上限は「最大反復」を使用して設定できます。

条件として、選択したサーバー・リソースで実行される関数またはコマンドを指定することができます。

- 関数 を使用する場合には、「条件」フィールドの最初に指定する必要があります。これは、Build Forge エンジンによって評価されます。サーバー・リソースには送信されません。関数のリスト、および使用方法についての説明は、421 ページの『条件関数』を参照してください。
- コマンド は、選択したサーバーで実行されます。ここで使用するコマンドはすべて、エージェントのシェル環境で有効である必要があります。実行時の戻りコードによって、条件の合格と不合格を判別します。

プロジェクトの Build Forge 変数は、条件式に使用できます。変数の表現方法および評価方法について詳しくは、330 ページの『ステップにおける変数の解釈』を参照してください。

#### Else コマンド

ステップ・タイプに「条件付き」を選択した場合にのみ表示されます。条件が `false` と評価された場合に実行するコマンドを指定します。

#### 環境変数

コマンドを実行する前に適用する環境変数を指定します。この環境変数の値によって、サーバー環境変数、プロジェクト環境変数、およびステップ環境変数から継承されたすべての値がオーバーライドされます。

## セレクトター

このステップのサーバーを選択する際に使用するセレクトターを指定します。「デフォルト」のままにすると、ステップはプロジェクトのセレクトターが決定したサーバーで実行されます。

## ブロードキャスト

チェック・マークを付けた場合、現行のセレクトター (指定した場合はステップ・セレクトター。それ以外の場合はプロジェクト・セレクトター) に適合するすべてのサーバーでステップが実行されます。実行時にシステムは、ブロードキャスト・ステップを、サーバーごとに 1 つずつ一連のステップと置き換え、ブロードキャスト・ステップの「スレッド」プロパティーに応じて、連続的または同時にそれらを実行します。

再始動時のブロードキャスト・ステップ動作: ブロードキャスト・ステップが再始動されたとき、ブロードキャストは行われません。その設定は、ステップの新規の開始にのみ適用されます。再始動時に、エンジンによってステップ用の単一サーバーがランダムに選出されます。

## タイムアウト (分単位)

現行コマンドが出力を生成するまでシステムが待機する時間を分単位で指定します (デフォルト値は 5 分です)。値 0 を指定すると、ステップがエージェントに正しく接続されている場合にはそのステップはタイムアウトになりません。タイムアウト値に達すると、システムでのステップは失敗します。ステップを「障害時に続行 (Continue on Fail)」に設定しない限り、プロジェクトも失敗します。

**結果** 結果プロパティーは、ステップの成功または失敗をシステムがどのように判断するかを決定します。終了コードのデフォルト値を使用して、コマンド・シェルによって戻される終了コードに基づいて、成功を判別します。コマンド出力を確認する「ログ・フィルター」を選択することもできます。選択する場合は、まず「ログ・フィルター」を作成する必要があります。

## 失敗動作

ステップが失敗した場合にジョブを停止するか継続するかを指定します。デフォルトでは、システムはジョブを停止します。

## スレッド

「はい」に設定した場合、このステップは他のステップと並行して実行されます。このプロパティーを「はい」に設定して、このステップのスレッド化 (その他のステップと同時にステップを実行) を許可します。プロパティーを「いいえ」に設定すると、スレッド化されません。プロパティーを「結合」に設定すると、スレッド化ステップのブロックは分離されます。ステップの最初のセットが完了してからでないと、「結合」ステップに続く、スレッド化ステップの次のセットを開始できません。

## 成功通知

ステップが成功した場合に通知されるアクセス・グループを指定します。

## 成功チェーン

現行のステップが成功した場合に起動するプロジェクトを指定します。  
(「警告」状況のステップは成功とみなされ、成功チェーンを起動します)。

## 成功待機

チェック・マークを付けた場合、成功チェーン・プロジェクトが完了するま

で、システムは現行プロジェクトを中断します。このステップ (またはそのプロジェクト) を取り消した場合は、チェーン・プロジェクトも取り消されます。これにチェック・マークを付けないと、チェーン・プロジェクトが非同期的に開始され、現在のプロジェクトは次のステップに進みます。

#### 失敗通知

ステップが失敗した場合に通知されるアクセス・グループを指定します。

#### 失敗チェーン

現在のステップが失敗すると起動するプロジェクトを指定します。(障害時に続行するように設定されたステップは失敗とみなされ、そのステップに割り当てられたすべての失敗チェーンを起動します。)

#### 失敗待機

チェック・マークを付けた場合、失敗チェーン・プロジェクトが完了するまで、システムは現行プロジェクトを中断します。このステップ (またはそのプロジェクト) を取り消した場合は、チェーン・プロジェクトも取り消されます。


### 「メモ」タブ

「メモ」タブには、ステップについて作成されたメモのタイム・スタンプ付きリストが含まれます。メモは手動で作成します。ステップ自体に対する編集内容は自動的に記録されません。タブには、メモの現行数 (例えば、**メモ (2)**) が表示されます。


メモを追加するには、次の手順を実行します。

1. 「メモ」タブをクリックします。
2. テキスト・フィールドに新規メモを書き込みます。
3. 「作成」をクリックします。

メモを編集するには、次の手順を実行します。

1. 「メモ」タブをクリックします。
2. 編集するメモの隣にある「編集」アイコン  をクリックします。編集を行います。
3. 「作成」をクリックします。

メモを削除するには、次の手順を実行します。

1. 「メモ」タブをクリックします。
2. 「ごみ箱」アイコン  をクリックします。メモを削除してよいかどうかを確認するプロンプトが表示されます。
3. 「OK」をクリックします。

---

## ステップの追加

### このタスクについて

この手順を実行すると、プロジェクトの最後にステップが追加されます。最初のステップとしてステップを挿入する方法、または既存ステップの間にステップを挿入

する方法について詳しくは、413 ページの『追加のステップ操作』を参照してください。

### 手順

1. 「プロジェクト」を選択し、プロジェクト名をクリックします。システムは、選択されたプロジェクトのステップのリストを表示します。
2. メインパネルの上部にある「ステップの追加」をクリックします。システムは、空のステップ詳細パネルを表示します。
3. プロパティに値を入力します。「名前」および「コマンド」は必須です。
4. 「ステップの保存」をクリックします。

---

## ステップの編集


### 手順

1. 「プロジェクト」を選択し、プロジェクト名をクリックします。システムは、選択されたプロジェクトのステップのリストを表示します。
2. ステップ名をクリックします。システムは、ステップのプロパティを、パネルの下部に表示します。
3. プロパティを編集します。
4. 「ステップの保存」をクリックします。

---

## ステップの無効化

### 手順

1. 「プロジェクト」を選択し、プロジェクト名をクリックします。システムは、選択されたプロジェクトのステップのリストを表示します。
2. ステップ名の前にあるチェック・ボックスをクリックします。ステップに  のようにチェック・マークを付けると、プロジェクトに対して以下の影響があります。
  - プロジェクトをジョブとして実行するときに、このステップは実行されません。
  - 通常どおりにジョブを開始する（「ジョブ」 > 「開始」を選択して、ジョブ名をクリックしてから、「ジョブ・ステップ」をクリックする）と、ステップのリストでそのステップがグレー表示になります。ステップは表示されていますが、開始しているジョブに対してはアクティブにできません。


### 注:

ステップの「詳細」タブで「アクティブ」プロパティを「無効」に設定することによっても、ステップを無効にすることができます。そのステップを保存すると、チェック・ボックスにチェック・マークが付けられます。

---

## 追加のステップ操作

ステップで作業するには、「プロジェクト」を選択し、次にプロジェクト名をクリックします。システムは、選択されたプロジェクトのステップのリストを表示します。

ステップ名の前にある「アクション」アイコン  をクリックして、以下の追加オプションを表示します。

- **新規ステップの挿入:** 選択したステップの上にステップを挿入します。
- **ステップの複製:** ステップとそのすべてのプロパティをコピーします。名前が変更されて、ステップの最後に番号が付加されます。Build というステップをコピーすると、Build COPY 0 という新規ステップができます。番号は自動的に設定されます。以下の場所にコピーできます。
  - 最上部: リストの先頭
  - 上: 現在のステップの直前
  - 下: 現在のステップの直後
  - 最下部: リストの最後
- **ステップの移動:** ステップをリスト内の別の位置に移動します。以下の場所に移動できます。
  - 最上部: リストの先頭
  - 上: 上に 1 つ移動
  - 移動...: ステップ番号を要求するダイアログ。ステップがその位置に移動し、それ以外のステップ番号は必要に応じて調整されます。
  - 下: 下に 1 つ移動
  - 最下部: リストの最後
- **ステップの削除**

---

## 実行フローの制御

以下のステップには、プロジェクト内の実行フローの制御に使用可能なフィーチャーがいくつかあります。

- **インライン:** ステップの「インライン」プロパティを使用して、プロジェクトまたはライブラリーを指定します。プロジェクトまたはライブラリーのステップは、このステップのコマンドの直後にインラインで実行されます。インライン・プロジェクトまたはライブラリーのステップは、ステップ・ログでインデントされます。
- **成功チェーンおよび失敗チェーン:** プロジェクトに指定されたチェーンとは異なる独自の「成功チェーン」と「失敗チェーン」をステップに指定することができます。
- **スレッド:** スレッド化のマークが付いた複数のステップを並行して実行することができます。ステップの「スレッド」プロパティを使用して、ステップにスレッド化のマークを付けます。



- **ブロードキャスト:** 複数のサーバー上でブロードキャストのマークが付いたステップを実行できます。ステップの「ブロードキャスト」プロパティを使用します。
- **条件付き:** 条件が `true` の場合にのみ実行するステップを設定できます。また、条件が `false` の場合に実行するコマンドの代替セット、およびインライン・プロジェクトまたはライブラリーを設定できます。「ステップ・タイプ」プロパティを「条件付き」に設定し、関連する「条件」プロパティと「Else」プロパティを用いてこのフィーチャーを使用します。
- **While ループ:** 条件が `true` と評価されている場合は、常にループ内でステップを実行することができます。「ステップ・タイプ」プロパティを「While ループ」に設定し、関連するプロパティを用いてこのフィーチャーを使用します。
- **ドット・コマンド:** `.run` コマンドや `.runwait` コマンドを使用して、ライブラリーまたはプロジェクトをステップのコマンドから起動します。

複雑な実行フローは、一般的にジョブを最適化する (つまり、必要な場合にのみステップを実行する) 場合に使用します。

ソフトウェアのビルド・エンジニアリングでは、ジョブの最適化は、アプリケーション全体の構築ではなく、アプリケーションの必要な一部分のみの構築を意味する場合があります。ジョブを実行してソースの状況と最新のコンパイル済みバイナリーを照合し、ソースに変更が見つかった場合にのみコンパイルを実行することができます。複雑なアプリケーションの場合は、実行フローをモジュールの依存関係とソースの状況に対応させることができます。

## ステップの実行方法

ステップ定義では、コマンド・プロパティにはオペレーティング・システム・コマンド、ドット・コマンド、またはそれらを組み合わせたものが含まれます。

個々のステップで、複数のコマンドを実行することができます。個々のコマンドを、別々の行に配置して切り離します。

**注:** ステップの「結果」プロパティでデフォルトの「終了コード」設定を使用する場合、ステップの最後のコマンドによって返される終了コードに基づいて、ステップ全体の成功または失敗が決定します。任意のコマンドで障害を検出するには、ログ・フィルターを作成して、その使用について「結果」プロパティに指定します。

システムは、ステップを実行する前に、ステップ環境を作成します。変数は、サーバー環境、プロジェクト環境、およびステップ環境で指定された値を使用して設定されます。327 ページの『環境の継承』を参照してください。デフォルトで、変数は構文解析され、その後コマンドで使用できるようになります。330 ページの『ステップにおける変数の解釈』を参照してください。

## シェルの指定

#! ディレクティブを使用して、コマンドの実行に使用されるシェルを指定できます。これは、Windows<sup>®</sup>、Linux<sup>®</sup>、および UNIX<sup>®</sup> システム (Windows エージェントは、指定されたインタープリターへのコマンドの引き渡しを扱います) で有効です。Windows で、ご使用のステップから `C:\perl\bin` にある Perl のコピーへコマ



ンドを送信するには、`#!C:\perl\bin\perl.exe` を使用します。Cygwin と共に Windows エージェントを使用しているが、コマンドを Windows シェル `cmd.exe` に送信する必要がある場合は、以下の行を使用できます。これには、Windows の暗黙パスを利用します。

```
#!cmd.exe /C
```

`/C` オプションは、`cmd.exe` では必須であることに注意してください。これは、このオプションがないと、ステップ・コマンドが送信された後に、追加のコマンドを待機するためです。UNIX または Linux コンピューターでは、`#!/bin/perl` コマンドを使用する場合があります。

**注:** Linux または UNIX システムで `#!` コマンドを使用する場合、システムは、標準のデフォルト・ディレクトリー (サーバー・パス、プロジェクト、名前、およびステップ・パス・フィールドの組み合わせで構成されるパス) に移動しません。これは、システムが、この必須の構文を予測できないためです。ユーザーは独自のディレクトリー移動コマンドを含める必要があります。これを実行するには、システムで作成された、`BF_SERVER_ROOT` および `BF_PROJECTNAME_PHYS` のような特殊な環境変数を使用します。

## システムがステップを部分に分割する方法

Rational® Build Forge® は、1 つのステップを複数の部分に分割します。各部分は、オペレーティング・システム・コマンドのセット、または単一ドット・コマンドで形成されます。例えば、以下のステップには 6 つの部分があります。

```
cmd1          # Part 1
cmd2
.dot_cmd1      # Part 2
cmd3          # Part 3
cmd4
.dot_cmd2      # Part 4
.sleep 30      # Part 5
.dot_cmd3      # Part 6
```

ステップの各部分は順番に実行されます。各部分の環境変数は、次の部分に渡されます。いずれかの部分でエラーが発生すると、処理は即時に停止します。

前のリリースでは、上記ステップの各部分は単一ステップである必要がありました。

**注:** 部分間に参照を作成しないでください。部分間にジャンプを作成しないでください。例えば、ある部分で `GOTO` を使用し、その宛先ラベルを別の部分に指定することは避けてください。

## インライン: プロジェクトまたはライブラリーのステップの組み込み

ステップの「インライン」プロパティーを使用して、指定したプロジェクトまたはライブラリーからすべてのステップを組み込みます。

ステップでインラインを使用するには、「インライン」プロパティーをプロジェクトまたはライブラリーの名前に設定します。ステップが実行されると、以下が行われます。

1. ステップによって、「コマンド」プロパティのコマンド (複数可) が実行されます。
2. ステップによって、「インライン」プロパティに指定されたプロジェクトまたはライブラリーのステップが実行されます。

インライン化されたステップを実行したいが、「コマンド」プロパティを使用しない場合、「コマンド」プロパティで `.sleep 0` を使用します。

## 呼び出しステップからのインライン継承

呼び出されたプロジェクトまたはライブラリーからのすべてのステップは、呼び出しステップのコンテキストで実行されます。インライン化されたステップでは、呼び出しステップの環境変数を継承します。

ただしシステムは、インラインのステップのデフォルトのセクターとして、そのインライン・プロジェクトのセクターを使用します。

## 呼び出しステップの状況に対するインライン・ステップの状況の影響

呼び出しステップのコマンドが成功した場合は、インライン化ステップの実行状況が以下のように判断されます。

- ステップのコマンドが失敗すると、ジョブは正常に停止します。ただし、ステップの「失敗動作」プロパティが「継続」に設定され、何らかのインラインが指定されている場合、そのインラインは実行されます。
- すべてのステップが実行状況として「成功」を返した場合、その呼び出しステップには「成功」のマークが付けられます。呼び出しステップに「成功チェーン」が指定されている場合は、それが実行されます。
- いずれかのインライン化されたステップが実行状況として「失敗」を返した場合、その呼び出しステップには「失敗」のマークが付けられます。呼び出しステップに「失敗チェーン」が指定されている場合は、それが実行されます。

この振る舞いにより、深くネストされたインラインおよびチェーンまで実行状況を簡単にトレースすることができます。

## インライン・ネスティング

プロジェクトまたはライブラリーのステップをインライン化すると、呼び出されたステップは呼び出し元のステップにネストされます。

ネスティングの最大レベルは、システム設定「インライン化の最大の深さ」によって設定されます。デフォルトは 32 です。ネスティングは、ジョブの開始時にテストされません。実行中のジョブが制限値を超えると、その制限値を超えた時点でジョブは失敗します。

ネスティングのレベルには、管理コンソールを実行しているホスト上の使用可能メモリに基づいた制限が課される場合もあります。

## ステップの成功チェーンおよび失敗チェーン

個々のステップには、成功チェーンと失敗チェーンを含めることができます。

ステップの成功チェーンと失敗チェーンは、プロジェクトの成功チェーンと失敗チェーンとは独立して実行されます。

「成功チェーン」および「失敗チェーン」プロパティは、プロジェクトまたはライブラリーの名前に設定されます。これらは、プロジェクト用に設定されたチェーンと同じように機能します。 369 ページの『待機が使用可能な場合のチェーニングされたプロジェクトの取り消し』を参照してください。

## スレッド: ステップの並列実行

スレッドを指定することにより、同じサーバーまたは別のサーバーでのステップの並列実行が可能になります。スレッドは、ステップの「スレッド」プロパティの設定値で制御されます。デフォルトでは、「スレッド」プロパティは「いいえ」に設定されています。スレッドは、プロジェクトの各部が単独で実行可能な場合に、プロジェクトの実行時間の短縮に役立ちます。

隣接する複数のステップの「スレッド」プロパティが「はい」に設定されている場合、システムは、ステップを並列して実行しようとします。このようなステップはスレッド化可能と見なされ、残りのジョブを続行する間に、各ステップを別個に実行することができます。スレッドは、以下の規則に従います。

- スレッド化を行うには、順次実行される 2 つ以上のステップで「スレッド」プロパティを「はい」に設定する必要があります。順次実行される一連のスレッド化ステップは、スレッド・ブロックと呼ばれます。スレッド・ブロックは、インラインの一部であるステップへと続けることができます。例えば、プロジェクト内のあるステップにインラインが含まれていて、そのインラインの最初のステップもスレッド化されている場合、その 2 つのステップは、同じスレッド・ブロックの一部になります。これらは同時に実行されます。スレッド・ブロックは、「結合」ステップまたは非スレッド化ステップが現れるまで、スレッド化ステップ (ネストされたインライン・ステップを含む) の後に続きます。ネストされたインライン・ステップを使用する際は、競合状態にならないように注意してください。親スレッド化ステップからの結果またはデータにより、インラインになったスレッド化ステップで競合状態が発生する可能性があります。
- スレッド・ブロックは、「スレッド」プロパティが「結合」に設定されているステップによって終了するか、スレッド化されていないステップを検出すると終了します。この時点で、ステップは再び順次実行されるようになります。
- システムは、スレッドが有効であるステップを見つけた場合、そのステップを開始しようとします。以降のステップがスレッド化される場合も、システムはそのステップを開始しようとし、次のステップに進みます。これは、スレッドが有効なステップがなくなるまで、またはジョブの上限に達するまで繰り返されます。プロジェクトのセクターでサーバー・プールを指定している場合は、ジョブの上限は概念的にプール内にある複数のサーバーのジョブの上限を合計した値になります。

**注:** スレッド化ステップの開始時刻は、ステップを実行するサーバーの可用性によって決まります。ステップを開始できない場合、システムは待機してから再度開始しようとします。最初に開始するステップを明示的に制御することはできません。

- 複数のステップを、(そのサーバーの能力によっては) 1 つのサーバーで、または (セレクトに一致するサーバーの数によっては) 複数のサーバーで、同時に実行することになる場合があります。
- すべてのステップを 1 つのサーバーで強制的に実行させる場合は、プロジェクトの「スティッキー」プロパティを使用してください。
- スレッド・ブロックが複数ある場合は、最初のスレッド・ブロックを完了してから、次のスレッド・ブロックを開始する必要があります。

以下の例では、ステップ 2、3、および 4 が完了してから、ステップ 5 および 6 が開始される必要があります。

プロジェクト	ステップのスレッド・プロパティ
ステップ 1	いいえ
ステップ 2	はい
ステップ 3	はい
ステップ 4	結合
ステップ 5	はい
ステップ 6	はい
ステップ 7	いいえ

- プロジェクトの「最大スレッド」プロパティを使用して、同時に実行可能なスレッドの数を制限します。スレッド化可能なステップとそのインライン・プロジェクトがある場合は、それぞれ並行して処理される可能性があります。すべての処理は、親プロジェクトの最大数に達するまでカウントされます。「最大スレッド」値に達すると、システムは新規並列処理の起動を停止します。プロジェクトの並列処理数が「最大スレッド」値を下回るまで待機した後で、システムは処理を続行します。

## 複数サーバーへのステップのブロードキャスト

多くのサーバーで実行できる有用なアクティビティがある場合は、ブロードキャスト・フィーチャーを使用して、多くのサーバーで同じステップを反復できます。

通常、ステップは 1 つのサーバーでしか実行できません。しかし、各ステップには「ブロードキャスト」チェック・ボックスがあります。ステップの「ブロードキャスト」ボックスにチェックマークを付けると、実行時にシステムがステップを一連の非ブロードキャスト・ステップ (ステップのセレクトに一致する各サーバーごとに、1 つ) に置換します。

**注:** ステップのセレクトが 1 つのサーバーにしか一致しない場合、ステップは一度だけ実行されます。

ブロードキャストの使用法として考えられるのは、以下のとおりです。

- サーバーのグループをリポートします。
- サーバーのグループでテストを実行します。
- サーバーのグループ全体に同じファイルのセットをコピーします。
- 同じソース・コードのセットを複数サーバーへチェックアウトし、後でそれらを単一の保守が容易なステップを持つ個々のタスクに使用できるようにします。

## ブロードキャスト・ステップの再始動

475 ページの『ブロードキャスト・ステップの再始動』を参照してください。

## ブロードキャスト・ステップにおけるスレッド化

システムは、実行時にブロードキャスト・ステップの置換ステップを作成する場合、次のようにしてステップをスレッド化します。

- ブロードキャスト・ステップの「スレッド化 (Thread)」プロパティが「いいえ (No)」に設定されている場合、置換ステップは同じ「スレッド」の値を取得し、これによりすべて連続して実行されます。各ステップは、次のステップが開始する前に完了する必要があります。
- ブロードキャスト・ステップの「スレッド」プロパティが「はい (Yes)」に設定されている場合も、置換ステップは、同じ「スレッド」の値を取得します。この結果、一連のステップは相互に平行して実行され、なおかつ、スレッド化されたステップは、ブロードキャスト・ステップに前後して実行されます。
- ブロードキャスト・ステップの「スレッド」プロパティが「結合 (Join)」に設定されている場合、システムは「スレッド」を「はい (Yes)」に設定して、置換ステップを作成します。ただし、最後のステップは「結合 (Join)」とマークされます。この結果、一連のステップが相互に平行して実行され、スレッド化されたステップはそれらに先行して実行されます。しかし、ステップ全体は、ブロードキャスト・ステップの後のステップを開始する前に完了する必要があります。

## ブロードキャスト・ステップからの別のプロジェクトの起動

インライン・プロジェクトが組み込まれているステップ、またはステップの受け渡しや失敗の際にプロジェクトをチェーンするステップをブロードキャストできます (「成功チェーン」/「失敗チェーン (Fail Chain)」)。別のプロジェクトを起動する (チェーンする) ステップをブロードキャストする場合は、ブロードキャスト・ステップは起動したプロジェクトのセクターを指定変更しないので注意してください。一般に、ブロードキャスト・ステップのセクターに一致する各サーバーでプロジェクトを起動する必要がある場合に、ブロードキャスト・ステップからプロジェクトを起動するときは、ライブラリーを使用します (独自のセクターを持たないプロジェクト)。

ライブラリーを使用しない場合は、ブロードキャスト・ステップの各コピーが異なるサーバーで実行されますが、インライン化またはチェーンされたプロジェクトは、その独自のセクターに従います。この場合、セクターは、ブロードキャスト・ステップのコピーと同じサーバーを選択しないことがあります。各ブロードキャスト・ステップは、異なるサーバーで実行されることになりますが、インライン化されたプロジェクトのすべてのステップは同じサーバーで複数回実行されます。

**注:** 「ブロードキャスト」を使用してライブラリーを、セクターに一致する各サーバーで 1 回起動したい場合は、ライブラリーに「スティッキー」オプションを設定します。これにより、そのすべてのステップ (デフォルト・プロジェクト・サーバーを使用) が同じサーバーで実行されます。



## 条件付きステップの実行

条件付き実行では、ステップの if-then-else 分岐を実行します。

簡単な If-Then 実行の場合:

1. ステップ・タイプを「条件付き」に設定します
2. 条件を評価可能な式に設定します。
3. 実行するコマンドを入力します。
4. 必要に応じて、実行するインラインを指定します (「インライン」を設定した場合は、コマンドはブランクのままでもかまいません)。

ジョブの実行時に「条件」が true と評価された場合は、ステップが実行されます。「インライン」プロジェクトまたはライブラリー (指定した場合) も実行されます。式が false と評価された場合は、スキップされ、次のステップのジョブの実行に進みます。

If-Then-Else 実行の場合:

「条件」で false が返されたときに別のコマンドおよび/またはインラインを実行する場合は、以下のようにさらにプロパティに入力します。

- 実行する Else コマンドを入力します。
- 必要に応じて、実行する「Else インライン」プロジェクトまたはライブラリーを指定します (「Else インライン」を設定した場合は、Else コマンドはブランクのままでもかまいません)。

ジョブの実行中に、条件が正常に評価された場合や「コマンド」または「Else コマンド」のコマンドが正常に実行された場合は、ステップ結果に「成功」のマークが付きます。使用されたパスを判別するには、ログを確認する必要があります。

421 ページの『条件関数』も参照してください。

## While ループの実行

While ループを実行することで、条件に基づいてステップを繰り返すことができます。

ステップを While ループとして実行するには、以下のようにします。

1. ステップ・タイプを「While ループ」に設定します。
2. 「条件」をコマンドまたは評価可能な式に設定します。
3. 実行するコマンドを入力します。
4. 必要に応じて、実行するインラインを指定します (「インライン」を設定した場合は、コマンドはブランクのままでもかまいません)。
5. 必要に応じて、「最大反復」を、ステップを実行する最大回数に設定します。開発中は、指定した条件が希望どおり正常に機能するまで、この上限を使用します。デフォルトは 100 です。
6. 「最大反復」値に達した場合にステップが失敗するようにするには、「最大に達した場合ステップを失敗する」を「はい」に設定します。それ以外の場合は、「最大反復」値に達したときにステップは成功します。

ステップの反復はそれぞれログに記録されます。各反復の結果は、「結果」基準に基づいて「成功」または「失敗」に設定されます。

『条件関数』も参照してください。

## 条件関数

条件関数は、「条件」ステップ・プロパティおよびアダプター XML エLEMENT の条件属性で使用されます。

- 「条件」プロパティを使用するステップの場合: 以下の関数が「条件」フィールドの先頭で使用されている場合、それらの関数はエンジンによって評価され、条件が `true` と評価されない限り、選択したサーバーに情報が送信されません。条件が `true` と評価された場合は、選択したサーバーでステップが実行されます。

### 重要:

- サーバー・リソースのシェル環境変数に設定されている変数に基づいて関数を使用しないでください。関数の評価は Build Forge エンジンで行われるため、関数は、ステップの Build Forge 環境変数で定義されている変数のみに基づいて作動します。
- 評価されるストリングに演算子を使用しないでください。演算子は `=`、`<`、`>`、`!` です。
- アダプター・テンプレートの場合: 以下の関数が、条件属性を持つアダプター XML ELEMENT で使用できます。関数は、アダプターの実行方法を指定するために使用されます。重要: `condition` 属性内の条件関数は、`condition="condition_function"` のように二重引用符で囲む必要があります。

以下の関数を使用できます。

### **true(expression)**

*expression* が `true` の場合は `true` が返されます。

### **false(expression)**

*expression* が `false` の場合は `true` が返されます。

### **contains(a,b)**

ストリング *a* にストリング *b* が含まれている場合に `true` が返されます。*a* および *b* のパラメーターは、リテラル・ストリングまたは変数です。リテラル・ストリングは、引用符で囲まないとはいけません。リテラル・ストリングが引用符で囲まれている場合、引用符は、評価されるストリングの一部になります。

注: Rational Build Forge バージョン 7.1.2 より前では、ストリング *a* がストリング *b* に含まれている場合に `true` が返されました。

### **hastext(var)**

変数が空でない場合は `true` が返されます。*var* は、Build Forge 内の変数セットです。

### **isempty(var)**

変数が指定されていない場合は `true` が返されます。*var* は、Build Forge 内の変数セットです。

*a eq b* *a* が *b* と等しい場合は `true` が返されます。*a* および *b* パラメーターは、



Build Forge 内の変数セットであるか、またはリテラル値です。文字および数値タイプも使用できます。パラメーターと演算子の間にはスペースを使用します。

***a ne b*** *a* が *b* と等しくない場合に **true** が返されます。*a* および *b* パラメーターは、Build Forge 内で設定された変数か、リテラル値です。文字および数値タイプも使用できます。文字および数値タイプも使用できます。パラメーターと演算子の間にはスペースを使用します。

***a contains b***

ストリング *b* がストリング *a* で検出された場合は **true** が返されます。リテラル・ストリングは引用符で囲まないでください。リテラル・ストリングが引用符で囲まれている場合、引用符は、評価されるストリングの一部になります。文字および数値タイプも使用できます。パラメーターと演算子の間にはスペースを使用します。

## 関数の式

**true()** および **false()** 関数の *expression* パラメーターでは、以下の演算子を使用できます。

***a==b*** 等しいことをテストします。パラメーターは、ストリングまたは数値です。パラメーターに、Build Forge で定義されているリテラルまたは変数を指定することができます。

***a eq b*** 等しいことをテストします。パラメーターは、ストリングまたは数値です。パラメーターに、Build Forge で定義されているリテラルまたは変数を指定することができます。パラメーターと演算子の間にはスペースを使用します。

***a!=b*** 等しくないことをテストします。パラメーターは、ストリングまたは数値です。パラメーターに、Build Forge で定義されているリテラルまたは変数を指定することができます。

***a ne b*** 等しくないことをテストします。パラメーターは、ストリングまたは数値です。パラメーターに、Build Forge で定義されているリテラルまたは変数を指定することができます。パラメーターと演算子の間にはスペースを使用します。

***a>b*** *a* が *b* より大きいことをテストします。パラメーターは数値でなければなりません。パラメーターに、Build Forge で定義されているリテラルまたは変数を指定することができます。リテラルでは、2+2 のように、算術演算子を使用できます。

***a<b*** *a* が *b* より大きくないことをテストします。パラメーターは数値でなければなりません。パラメーターに、Build Forge で定義されているリテラルまたは変数を指定することができます。リテラルでは、2+2 のように、算術演算子を使用できます。

***a>=b*** *a* が *b* より大きいまたは等しいことをテストします。パラメーターは数値でなければなりません。パラメーターに、Build Forge で定義されているリテラルまたは変数を指定することができます。リテラルでは、2+2 のように、算術演算子を使用できます。

***a<=b*** *a* が *b* より大きくないか、または等しいことをテストします。パラメータ

ーは数値でなければなりません。パラメーターに、 Build Forge で定義されているリテラルまたは変数を指定することができます。リテラルでは、2+2のように、算術演算子を使用できます。

#### ***a contains b***

文字列 *b* が文字列 *a* で検出されることをテストします。パラメーターに、 Build Forge で定義されているリテラルまたは変数を指定することができます。リテラル・文字列は、引用符で囲まないでください。

### **条件関数の例**

下の表の例では、変数は次のように設定されています。

- \$AVAL には、値 String が含まれています。
- \$BVAL には、値 3 が含まれています。

条件	評価結果	メモ
A String contains \$AVAL	TRUE	文字列比較
A String contains "String"	FALSE	String を囲む引用符は、比較対象の一部となります。
true(A String contains \$AVAL)	TRUE	文字列比較
\$AVAL contains String	TRUE	文字列比較
\$AVAL contains "String"	FALSE	String を囲む引用符は、比較対象の一部となります。
contains(A String,\$AVAL)	TRUE	文字列比較
true(A String contains "\$AVAL")	FALSE	\$AVAL を囲む引用符は、比較対照の一部となります。"A String" の "String" 部分は引用符で囲まれていません。
A String != \$AVAL	TRUE	文字列比較
A String ne \$AVAL	TRUE	文字列比較
false("Not Here" contains \$AVAL)	TRUE	テスト・文字列比較
true(2+1 == \$BVAL)	TRUE	数式 (等式)
false(2+2 < \$BVAL)	TRUE	数式 (不等式)
\$AVAL eq \$AVAL	TRUE	テスト・文字列比較
true(\$AVAL ne Linus)	TRUE	テスト・文字列比較
true(\$BVAL > 2+2)	FALSE	(3 > 2+2) は true ではありません。
contains(Not Here, \$AVAL)	FALSE	テスト・文字列比較

## **ステップからのプロジェクトの起動**

ステップからプロジェクトを起動するには、.run コマンドまたは .runwait コマンドを使用します。

この方法で起動されたプロジェクトは、チェーンのように機能します。プロジェクトは、独自のセレクターと環境変数を使用して実行されます。 456 ページの『`.run` および `.runwait`』を参照してください。

---

## ログ出力のカスタマイズ

以下のトピックでは、ステップでコマンド・フィーチャーを使用してログ出力をカスタマイズする方法をいくつか示します。

### ステップでのログ出力のラベル付け

ステップ・ログでステップ出力を各自のカテゴリにリストする場合は、ラベルを作成します。

#### 始める前に

このタスクは、すでにセレクター、サーバー、およびプロジェクトが作成されていることを前提としています。また、**echo** コマンドを受け入れるオペレーティング・システム (例えば、Windows、Linux、UNIX など) でサーバーを使用していることを前提としています。

#### このタスクについて

任意の出力行の先頭に大文字のラベルを組み込むことができます。このラベルは、ステップが終了するまで、または新規ラベルが検出されるまで使用されます。

ラベルには、以下の構文規則があります。

- 大文字のみが含まれ、その後にコロンが付けられます (ラベルには、スペース、句読点、数字、または小文字を含めることはできません。 "SPACESHIPS:" は有効です。 "Space Ships:" は有効ではありません)。
- 少なくとも 3 つの文字が含まれます。

ステップで **echo** コマンドを使用してラベルを作成することができます。システムは、**echo** コマンドに対する最初の引数がラベル構文規則に従う場合、その引数をラベルとして認識します。

注: システムは、任意のビルド出力における行の最初のテキストが構文規則に従う場合、**echo** コマンドの引数ではない場合でも、そのテキストをラベルとして認識します。

以下の例に示すように、ラベルは新規だけでなく既存の出力ラベル名にも設定できます。

#### 手順

1. プロジェクトで新規ステップを作成します。この例では、プロジェクトに `Say_hi` という名前を付けます。
2. ステップに `LabeledLogOutput` という名前を付けます。
3. 「コマンド」フィールドに次のテキストを入力します。

```
echo SPACESHIPS: Voyager I
echo Voyager II
echo EXEC: 既存のカテゴリにテキストを追加することもできます
```

4. プロジェクトを実行します。
5. ジョブが完了したら、ログを表示します。

## タスクの結果

カテゴリー・ヘッダーに **SPACESHIPS** というチェック・ボックスがあり、出力行の 354 と 355 に **SPACESHIPS** というラベルが付けられていることに注目してください。

```

79      7/9/10 10:24 AM EXEC      Locale set to 'English_United States.1252'
250    7/9/10 10:24 AM EXEC      Locale set to 'English_United States.1252'
349    7/9/10 10:24 AM EXEC      Performing variable expansion on command line
353    7/9/10 10:24 AM EXEC      start [C:%data%BuildForge%buildforge_projects%Say_hi%BUILD_1@RBF-14]
354    7/9/10 10:24 AM SPACESHIPS      Voyager I
355    7/9/10 10:24 AM SPACESHIPS      Voyager II
356    7/9/10 10:24 AM EXEC      You can add text to existing categories as well
357    7/9/10 10:24 AM EXEC      end [C:%data%BuildForge%buildforge_projects%Say_hi%BUILD_1@RBF-14]
```

注: **SPACESHIPS** チェック・ボックスを選択またはクリアすることにより、該当カテゴリーを表示したり隠したりすることができます。

注: テキストを **EXEC** カテゴリーに追加するときは、特殊文字 (例えば、括弧) を使用しないでください。システムによっては、引用符で囲むとこれらの文字が実行可能になることがあります。通常は、単にプレーン・テキストのみを使用することをお勧めします。

## カラー・リンクまたはアクティブ・リンクを使用したステップ出力の強調表示

ログ・ビューアーでは、出力テキストの **[STATUS]** コマンドおよび **[URL]** コマンドが認識されます。これらのコマンドでは大文字小文字が区別されません。これらのコマンドは、**echo** コマンドまたは出力を生成する任意のコマンドで使用できます。開始タグおよび終了タグは、出力の同じ行に表示されなければなりません。

- **[STATUS=condition]** タグおよび **[/STATUS]** タグを使用して、強調表示するテキストにマークを付けます。 *condition* には、以下のように強調表示カラーで設定します。
  - **PASS** - 緑
  - **WARN** - 黄
  - **FAIL** - 赤
  - **RUN** - 青
- **[URL]** タグと **[/URL]** タグは、テキストをアクティブ・ハイパーリンクとしてマーク付けする場合に使用します。

**[status]** を使用した場合の例を以下に示します。

```
echo [STATUS=WARN]Access to source control timed out[/STATUS]
```

**[url]** を使用した場合の例を以下に示します。

```
echo See the support forums at [url]http://www.ibm.com[/url]
```

ログには、指定した **URL** へのアクティブ・ハイパーリンクとしてテキストが表示されます。ハイパーリンクをクリックすると、新規ブラウザ・ウィンドウまたは

タブが開き、該当ページが表示されます。ただし、[URL] アドレスのログ・フィルターとの一致部分によって、リンクが機能しなくなる場合があることに注意してください。

---

## ジョブ・データの処理

以下のトピックでは、コマンド・フィーチャーを使用してプロジェクトやジョブを変更する方法をいくつか示します。

### プロジェクト・ファイルへのビルド番号の組み込み

`.strsub` コマンドを使用して、1 つのストリングを、ファイルにある別のストリングと交換できます。一般には、標準のトークンを、現在のジョブ番号を指定する `$B` 変数などのシステム変数に置き換えます。

`.strsub` ドット・コマンドを使用して、コード・ファイルに、ビルドまたはバージョン番号を組み込むことができます。`.strsub` コマンドをプロジェクトの前方に配置することにより、後のステップは、更新された情報を含むファイルをコンパイルできます。

例えば、以下のステップでプロジェクトをセットアップして、ビルド番号を組み込みます。

1. `_BUILD_` のような固有のストリングを、プロジェクトのファイルに追加します。  
例えば、ファイル `README.TXT` を変更し、以下のようにバージョン宣言を変更します。

```
Application version 5.0.123
Application version 5.0._BUILD_
```

2. プロジェクトの前の方にあるステップが、処理されるファイルをチェックアウトします。`README.TXT` がチェックアウトされてから、`_BUILD_` を `$B` システム変数に置き換えるステップを追加します。コマンドでは、以下を使用します。

```
.strsub _BUILD_ $B README.TXT
```

3. プロジェクトを実行し、`README.TXT` ファイルが現在のジョブ番号を含んでいるか検査します。プロジェクトの 3 回目の実行で、`README.TXT` ファイルは次の行を含んでいるはずです。

```
Application version 5.0.3
```

### 機能拡張

以下のような方法で、この操作を機能拡張することができます。

- 追加の環境変数を使用する。例えば、名前が `$MAJORVERSION` および `$MINORVERSION` の変数を作成して、以下のようにそれらを使用します。

```
.strsub _MAJORVERSION_ $MAJORVERSION README.TXT
.strsub _MINORVERSION_ $MINORVERSION README.TXT
```

- プロジェクトの開始時に、環境変数を更新する。「ジョブ」 > 「開始」を選択してプロジェクトを開始すると、現在の環境変数を表示し、それらの値を編集してから、プロジェクトを起動できます。例えば、変数として、ジョブにコメントを含めることができます。コメント変数で、プロジェクト・アクションの「変更が必要」を使用して、プロジェクトの実行時に、ユーザーに強制的に新規の値を入力させます。

## ジョブ中のビルド・タグの変更

ジョブの間にタグの値を完全に新しい値に設定することができます。それには、次の構文の `.retag` コマンドを使用します。

```
.retag <new tag value>
```

次の例は、簡単な使用法です。

```
.retag MyProject
```

次は、より複雑な使用法です。

```
.retag Job_${B}_${BF_D}
```

この例では、実行増分と現在日付のシステム変数を使用するようにタグを設定しています。サーバーのコマンド・インタープリターに対するコマンドを使用して、結果を設定することができます。ドット・コマンド内でコマンドを使用するには、バックチックまたはバッククオート文字 ( ``` ) でコマンドを囲みます。

```
.retag `hostname`
```

この例では、ステップを実行中のサーバーでホスト名コマンドを実行した結果にタグを設定します。

注: コマンドのバックチック形式と標準割り当て形式を混用してはなりません。

## ジョブ中の環境変数値の変更

`.set` コマンド、`.bset` コマンド、および `.unset` コマンドを使用して、ステップ内から環境変数を変更できます。これらのコマンドにより、既存環境変数の値を以下のように変更できます。

- 環境変数のマスター・レコードを変更するには、`.set` コマンドを使用します。システムはプロジェクトを実行するときに、プロジェクト環境変数のコピーをマスター・レコードから作成し、そのコピーをプロジェクトのデフォルトとして使用します。これにより、以下のような影響があります。
  - `.set` コマンドでプロジェクト環境変数を変更しても、デフォルトの環境変数を使用する後続ステップでは、その変更は認識されません。これは、システムがマスター・レコードにまで戻って参照しないためです。
  - `.set` コマンドを使用して環境変数を変更し、後続ステップで同じ環境変数を明示的に使用する場合、そのステップでは変更された内容が認識されます。ステップで特定の環境変数が選択されると、システムはその環境変数のマスター・レコードにまで戻って参照するからです。指定されたグループがプロジェクトのデフォルト・グループと同じ場合でも、ステップの環境設定が「デフォルト」でない限り、これは機能します。
  - `.set` コマンドで行った変更内容は、ジョブが終了した後も存続します。新しく開始されるジョブでは、直前に実行した `.set` コマンドで作成された値が使用されます。

次の基本構文を使用します。

```
.set env <EnvGroupName>[(<SnapshotName>)] "<VariableName>=<DesiredValue>"
```

- `.bset` コマンドを使用して、ジョブの実行中に変数の値を追加または変更します。変更は、`.bset` が表示されたステップの後のステップで有効になります。それらの変更は、ジョブの残りの部分に対して有効です。



```
.bset env "<VariableName>=<DesiredValue>"
```

注: .set コマンドとは異なり、.bset コマンドに指定する変数は、設定するときに存在していなくてもかまいません。したがって、.bset コマンドを使用してジョブの中で新規変数を作成することができます。変数の値は、現行ジョブの後では存続していません。

- .tset コマンドを使用して、ジョブの実行中に変数の値を追加または変更します。変更は、現行のステップで有効になります。それらの変更は、ステップ内のその他のコマンドと、ステップで指定されたすべてのインラインに対して有効です。変数の値は、現行ステップの後では存続していません。

```
.tset env "<VariableName>=<DesiredValue>"
```

注: .set コマンドとは異なり、.tset コマンドに指定する変数は、設定するときに存在していなくてもかまいません。したがって、.tset コマンドを使用してジョブの中で新規変数を作成することができます。

## 複数の変数の設定

以下の例にあるように、追加の変数と値のペアをスペースで区切って組み込むことにより、これらのコマンドで複数の変数を同時に設定できます。

```
.set env MyGroup "X=5" "X2=45"  
.bset env "Y=7" "CompilerVersion=4.511"  
.tset env "Z=9" "Z2=54"
```

## 値を設定するためのコマンド出力の使用

.set または .bset コマンドの場合、コマンドをサーバーのコマンド・インタープリターに送信することで、変数の値を生成することができます。ドット・コマンド内でコマンドを使用するには、次のようにバックチック文字でコマンドを囲みます。コマンド例を以下に示します。

```
.set env SetupGroup "PerlVer=`perl --version`"
```

変数 PerlVer を perl --version コマンドの出力に設定します。

変数に保管できるのは 256 文字のみです。それ以上の文字が変数に割り当てられると、値は切り捨てられます。

デフォルトでは、システムはコマンドの出力全体をバックチックで囲んで変数に割り当てますが、大括弧で囲んだ範囲コマンドを使用すると、コマンド出力から、変数に割り当てたい行を選択することができます。範囲の数値では、0 インデックス (最初の行の番号がゼロ、2 番目の行は 1、以下同様) を使用して、出力の行を指定します。次の例では、

```
.set env SetupGroup "WindowsIPinfo[0,5-8]=`ipconfig`"
```

変数 WindowsIPinfo は、ipconfig コマンドの出力の 6 行目から 9 行目までを受け取ります。

以下はすべて、単一行、行のグループまたは組み合わせを選択する、有効な範囲修飾子です。

[5]

[4-6]



[1,2,5,8-11]

システムは、区切りなしで行を結合します。スペースや復帰は追加されません。

注: コマンドのバックチック形式と標準割り当て形式を混用してはなりません。

## レジスターの使用

レジスターは、ステップが永続データを保管するために使用できる汎用バッファです。通常のレジスターは、文字で始まる単一文字の名前を持つことも、複数文字の名前を持つこともできます。

基礎となるデータベースの大/小文字の区別によって、レジスター名を作成する際に使用できる大/小文字が決定されます。

通知テンプレートにレジスター変数を組み込むことができます。通知テンプレート内のレジスターを参照する場合は、`${X}` というように中括弧で囲んだ形式を使用します。空のレジスターを参照すると、空ストリングが戻されます。

`.push` および `.pop` の各ドット・コマンドを使用して、レジスターに情報を保管したり、レジスターから情報を取り出したりします。現在のジョブ・タグをレジスターの内容に等しいものにする `.poptag` コマンド (451 ページの『`.poptag`』) も参照してください。

注: レジスターは、変数のようなコマンドでは使用できません。レジスターの値を使用する前に、まず、その値をファイルにポップする必要があります。

表 11. 特殊レジスター

レジスター	内容
!	「失敗 (Fail)」フィルター・パターンに一致したコマンド出力行が含まれます。 注: このレジスターは、フィルターが適用されるステップの範囲内でのみ表示されます。 Rational Build Forge がフィルターを使用したステップの処理をすると、そのステップ・フィルターによって設定される可能性のあったレジスターの内容は、今後、表示されなくなります。
@	「成功 (Pass)」フィルター・パターンに一致したコマンド出力行が含まれます。 注: このレジスターは、フィルターが適用されるステップの範囲内でのみ表示されます。 Rational Build Forge がフィルターを使用したステップの処理をすると、そのステップ・フィルターによって設定される可能性のあったレジスターの内容は、今後、表示されなくなります。

表 11. 特殊レジスター (続き)

レジスター	内容
=	<p>ジョブの注釈データベースを指定します。ステップが、ジョブへの注釈としてファイルからデータを追加できるようになります。このレジスターは、他のレジスターとは以下の点で異なります。</p> <ul style="list-style-type: none"> <li>• このレジスターには書き込み (プッシュ) のみが可能で、読み取りを行うことはできません。</li> <li>• このレジスターにプッシュされたデータは、前のデータに上書きされるのではなく、必ずレジスターに追加されます。</li> <li>• システムは、タイム・スタンプとユーザー ID に追加されたデータを提供します。これにより、ジョブの注釈の監査証跡が保存されます。</li> </ul>

## プロジェクト・レジスター

プロジェクト・レジスターは、通常のレジスターとは異なります。プロジェクト・レジスターは複数のビルドにわたって存続し、管理コンソール・インターフェースを使用して作成および表示できるため、ある種の構成情報を保管するには理想的な手段です。

例えば、プロジェクト・レジスターとして IBM® Rational® ClearCase® の構成仕様を保管し、ステップに `.pop -p` コマンドを使用させて、その仕様を抽出し、それを `cleartool setcs` コマンドで使用するビルドを構成することができます。これにより、プロジェクトと一緒に構成を管理することができます。

ALPHA というプロジェクト・レジスターがある場合、まったく異なる内容の、ALPHA という通常のレジスターも使用することができます。プロジェクト・レジスターは、別個の値の集合です。


プロジェクト・レジスターの作成およびアクセスは、次の 2 つの方法で行うことができます。

- `-p` オプションを追加したドット・コマンド (`.push` および `.pop`) を使用して行います。`-p` オプションを使用すると、そのコマンドは通常のレジスターではなくプロジェクト・レジスターを参照します。

例えば、


```
.push -p ALPHA register.txt
```

というコマンドは、ファイル `register.txt` の内容を ALPHA というプロジェクト・レジスターに入れます。

- 管理コンソール・インターフェースを使用して行います。「プロジェクト」を選択してから、対象プロジェクトの名前の隣にある「編集」アイコン  をクリ

ックします。プロジェクト・プロパティが下部パネルに表示されます。「レジスタ」タブをクリックしてそのプロジェクトのレジスタを表示してください。このタブで提供されるパネルを使用して、次のようにレジスタを管理します。

- 新規レジスタを作成するには、名前および内容を入力してから「作成」をクリックします。
- レジスタを削除するには、パネルの右側のリストに表示されたレジスタの名前の隣にある、ごみ箱アイコンをクリックします。
- レジスタを編集するには、リストに表示されたレジスタの名前の隣にある

「編集」アイコン  をクリックします。システムは、レジスタの内容をレジスタ・パネルに取り込みます。変更を行ってから、「編集済みレジスタの保存 (Save Edited Register)」ボタンをクリックします。

プロジェクトへのアクセス権限を持つ人は誰でも、そのプロジェクト・レジスタを表示および編集することができます。

注: 基礎となるデータベースの大/小文字の区別によって、レジスタ名を作成する際に使用できる大/小文字が決定されます。

---

## ステップでのサーバー・リソース間のファイルのコピー

ドット・コマンドを使用すると、ファイルのあるサーバー・リソースから別のサーバー・リソースにコピーできます。このトピックでは、`.get` コマンドと `.put` コマンド (単一ファイルの場合) および `.rget` コマンドと `.rput` コマンド (ディレクトリ・ツリー全体をコピーする場合) の使用方法について説明します。

**重要:** サーバー・リソース間でのファイルのコピーを有効にする必要があります。デフォルトでは有効になっていません。『サーバー・リソースでのファイルのコピーの有効化』を参照してください。

### サーバー・リソースでのファイルのコピーの有効化

サーバーのデフォルト設定では、ドット・コマンドを使用してファイルをコピーすることはできません。プロジェクトにおいて、サーバー・リソース間でファイルをコピーできるようにするには、サーバーの「ファイル」プロパティを変更します。

設定値を変更する場合は、次の手順を実行します。

1. 「サーバー」 > `<ServerName>` を選択します。
2. 「ファイル」プロパティで「なし」以外の値を選択します。サーバーから、またはサーバーへ、あるいはその両方向のファイルのコピーを有効にすることができます。

注: `.get`、`.put`、`.rput`、または `.rget` の使用時には、オペレーティング・システムに十分な一時ディスク・スペースがあることを確認してください。Windows 上のエージェントでは、デフォルトで Server Auth User の temp ディレクトリが

使用されます。Linux 上または UNIX 上のエージェントでは、デフォルトで /tmp が使用されます。bfagent.conf ファイルを変更して bfagent を再始動することにより、これを変更することができます。

## サーバーからのファイルの取得

サーバーからファイルのコピーを取得して、それを現行ステップの作業ディレクトリーと相対的な場所にある宛先に置くには、.get コマンドを使用します。例えば、winbuildserver1 という名前のサーバーの config ディレクトリーにファイル config.txt がある場合は、次のステップをプロジェクトに追加して、このファイルを現行サーバーの config ディレクトリーにコピーできます。

```
.get winbuildserver1:/config/config.txt ./config/config.txt
```

詳しくは、以下を参照してください。

- .getの参照エントリー
- 454 ページの『.rget』の参照エントリー
- ステップのパスの解決方法に関する説明 (485 ページの『ジョブ用の作業ディレクトリー』)

## サーバーでのファイルの配置

現行サーバーから別のサーバーにファイルをコピーする場合は、.put コマンドを使用してください。以下のサンプル・ステップでは、現行サーバー上の config ディレクトリーに config.txt というファイルがあり、そのファイルに現行パスからアクセスできることを前提にしています。

```
.put ./config/config.txt winbuildserver1:/config/config.txt
```

詳しくは、以下を参照してください。

- 『.put』の参照エントリー
- 『.rput』の参照エントリー
- ステップのパスの解決方法に関する説明 (485 ページの『ジョブ用の作業ディレクトリー』)

## 直接ファイル転送を構成する

\_XSTREAM\_PROTOCOL トリガー変数によって、エージェント間の直接ファイル転送がジョブの環境にあるとき、その転送が可能になります。送信側エージェント、受信側エージェント、およびエンジンのすべてが、直接ファイル転送に対応している必要があります。いくつかのパラメーターが、エンコード方式、ネットワーク、およびファイル転送特性を制御します。

344 ページの『トリガー変数のリファレンス』を、また 193 ページの『bfagent.conf の参照』で xstream\_\* パラメーターを参照してください。

---

## ステップ処理のトラブルシューティング

ステップ処理で問題が発生した場合は、このトピックの情報を参照し、適用可能な回避策または解決方法があるかどうかを確認します。

## ANT ビルド・コマンド以降のステップ・コマンドがジョブで処理されない

### 問題記述:

ANT ビルド・コマンド以降のステップ内のコマンドが処理されません。

以下のステップ例では、**echo** コマンドが実行されません。

```
<path to ant bin directory> ant -f <path to Java project>%build.xml build  
echo "Ant build complete"
```

**説明:** ANT ビルドでは、その失敗と成功に関係なく、エラー・コード 1 が返されます。

ステップの「コマンド」プロパティで、複数のコマンドが使用されている場合は、最後に実行されたコマンドの終了状況のみがステップ結果状況に影響します。ANT ビルド・コマンドを含むステップのコマンド・スクリプトをサーバーで実行したときに、エラー状況が 1 になった場合、ANT ビルド以降のすべてのコマンドが失敗します。

### 解決方法:

ステップ・ログ・フィルターを作成して、ANT ビルドで生成されたステップ出力を処理します。ステップ・ログ・フィルターでは、ステップ結果を設定し、ジョブの次のステップを確実に処理できるようにします。

1. ANT ビルドは、ステップ内の唯一のコマンドまたはステップ内の最後のコマンドである必要があります。

ログ・フィルターを使用しないと、ANT ビルドからエラー・コード 1 が返され、ステップ結果は失敗に設定されます。

2. ログ・フィルターを作成して、ステップ出力の該当する失敗テキスト・ストリング (BUILD FAILED) を検索し、ステップ処理を効果的に制御します。

テキスト・ストリングが検出された場合は、「失敗の設定」アクションを使用して、ステップ結果を失敗に設定します。ステップ・ログ・フィルターを使用したときにテキスト・ストリングが検出されなかった場合は、ステップ結果は常に成功に設定されます。

ステップでのログ・フィルター設定について詳しくは、375 ページの『ログ・フィルター』を参照してください。

## Windows バッチ・コマンド以降のステップ・コマンドが実行されない

### 問題記述:

Windows バッチ・コマンド以降のステップ・コマンドは処理されません。

以下のステップ例では、最初のバッチ・ファイル以降の 2 つのコマンドが実行されません。

```
C:%script1.bat  
C:%script2.bat  
echo "Performed both batch commands"
```

**説明:** ステップ内のすべてのコマンドは、サーバーで実行される Windows バッ

チ・ファイルに配置されます。ステップ・コマンドにバッチ・ファイルへの参照が含まれている場合、そのステップはバッチ・ファイルの実行後に終了します。バッチ・ファイル参照より後のステップ・コマンドは実行されません。

#### 解決方法:

ステップ内で `call` コマンドを使用してバッチ・ファイルを実行します。`call` コマンドは、ステップのバッチ・ファイル内で実行されます。

```
call C:%script1.bat
call C:%script2.bat
echo "Performed both batch commands"
```

---

## ドット・コマンドのリファレンス

ステップの「コマンド」フィールドでは、ドット・コマンドを使用することができます。これらは、システム内の特殊な機能と関数へのアクセスを提供します。

1 つのステップでドット・コマンドを通常のコマンドと混用することができ、また単一ステップで複数のドット・コマンドを使用できます。ただし、単一ステップで複数の `.scan` コマンドは使用しないでください。そのように使用しても、システムが正確にコマンドの結果をレポートできないからです。

環境変数で使用するドット・コマンドを示したリストが別に用意されています。340 ページの『変数でのドット・コマンドの使用』を参照してください。

## ドット・コマンドの構文

構文仕様: 各ドット・コマンドの記述には、次の表記を使用したコマンド指定が示されています。

- ユーザーが指定する値は、不等号括弧で囲まれています: `<value>`
- オプションのテキストは大括弧で囲まれています: `[optional text]`

特に断りがない限り、コマンド・パラメーターの環境変数を使用できます。

ドット・コマンドでパラメーターの環境変数を受け入れる場合は、以下の表記も使用できます。

- コマンド実行用のバックチック (```)
- 範囲を示すための大括弧 (`[ および ]`) (範囲表記とも呼ばれます)

次の `.set` を例にして考えてみましょう。

```
.set envgroupname "FOO[1,3-5]=`cat foofile`"
```

この場合、エージェントは `cat foofile` を実行し、これにより `foofile` のコンテンツがリストされます。範囲表記を使用することで、行 1 と、行 3 から 5 が抽出されます。次に、`.set` コマンドではこれらの行を使用して、`envgroupname` という名前の環境変数グループ内にある既存の変数「`FOO`」を更新します。

## .bom

```
.bom addcategory "category"
.bom setcolumn "category" "section" "column" [...]
```



```
.bom data category "section" "column=value" [...]
```

.bom コマンドは、ビルドの部品表 (BOM) にデータを追加します。これを使用して、カテゴリ、セクション、およびデータを追加できます。

### カテゴリ

カテゴリは、BOM に印刷される見出しです。カテゴリを指定するには、addcategory オプションを使用します。

```
.bom addcategory "category"
```

### セクション

セクションは、あるカテゴリ内のデータの列を定義します。セクションおよびセクション内の列を指定するには、setcolumn オプションを使用します。セクション名は印刷されません。セクションの一連の列見出しは、セクションの先頭に印刷されます。-p オプションを使用して親セクションを指定することにより、セクションをネストできます。

セクションおよび列は、data オプションがそれらにデータを追加しようとする前に、定義する必要があります。

```
.bom setcolumn "category" "section" "columnheader" [...]  
.bom setcolumn "category" "section" -p parentsection"columnheader" [...]
```

**データ** データは、セクションに定義された列にデータを追加します。セクションと列は、setcolumn を使用して既に定義されている必要があります。

```
.bom data category "section" "column=value" [...]
```

カテゴリ、セクションおよび列、およびデータは、次の例に示された順序で指定する必要があります。

```
.bom addcategory "Spaceships"  
.bom setcolumn "Spaceships" "Section1" "ShipName" "WarpSpeed" "Tonnage"  
.bom setcolumn "Spaceships" "Subsection1" -p "Section1" "ShippingDate" "ShippingManifest"  
.bom data "Spaceships" "Section1" "ShipName=SpaceShipOne" "WarpSpeed=9" "Tonnage=10000"  
.bom data "Spaceships" "Subsection1" "ShippingDate=123" "ShippingManifest=456"  
.bom data "Spaceships" "Section1" "ShipName=Freighter" "WarpSpeed=6" "Tonnage=20000"
```

この例は、BOM では次のように表示されます。

ShipName	WarpSpeed	Tonnage	ShippingDate	ShippingManifest
SpaceShipOne	9	10000	123	456
Freighter	6	20000		

他のドット・コマンドと同様に、コマンドで環境変数を使用することができます。

```
.bom data "Spaceships" "${SECTION}" "ShipName=${NAME}" "WarpSpeed=${SPEED}" "Tonnage=${TONNAGE}"
```

のようなコマンドは、以前のコマンドによって環境変数にロードされたデータを BOM に設定します。

任意の数の列を作成できますが、最後の行が設定されるまで、システムは BOM に行を書き込みません。

データ行から列を省略する場合、以下の例に示すように、システムは前の行からの値を使用します。



```
.bom addcategory "Spaceships"
.bom setcolumn "Spaceships" "Section1" "ShipName" "WarpSpeed" "Tonnage"
.bom data "Spaceships" "Section1" "ShipName=SpaceShipTwo" "WarpSpeed=3" "Tonnage=30000"
.bom data "Spaceships" "Section1" "ShipName=Tanker" "Tonnage=50000"
```

結果として、SpaceShipOne からの WarpSpeed の値が繰り返されます。

ShipName	WarpSpeed	Tonnage
SpaceShipTwo	3	30000
Tanker	3	50000

## .bomexport

### 説明

.bomexport ドット・コマンドは、ジョブの BOM を XML ファイルにエクスポートします。BOM 情報を収集したら、.bomexport はそれを指定したファイルおよびロケーションに保存します。

パスおよびファイル名は任意に指定できます。デフォルトでは、Build Forge は BOM レポートをサーバー上のステップの作業ディレクトリーに保存し、タグ名をファイル名として使用します (<build\_tagname>.xml)。

.bomexport コマンドをプロジェクトの最後のステップとして指定します。

注: Java MJC では、bomexport ユーティリティーまたは .bomexport コマンドで、ジョブのステップ結果およびステップ・ログを取得することはできません。ただし、サーバー・マニフェスト・セクションを取得することは可能です。Java MJC スキーマと Perl スキーマとの相違のため、この問題は Perl エンジンでのジョブには該当しません。

### 構文

```
.bomexport [path_name][file_name]
```

### オプション

オプション	説明
path_name	任意指定のパス名。指定する場合、パスは、Build Forge サーバー上のステップの作業ディレクトリーへの相対パスである必要があります。省略する場合、ファイルはステップの作業ディレクトリーに保存されます。
file_name	任意指定のファイル名。ジョブの BOM は、XML フォーマットのファイルに保存されます。ファイル名が指定されていない場合、ファイル名はビルド・タグ名および _BOM というストリングを使用して作成されます (<build_tagname>_BOM.xml)。

### 例

```
.bomexport
.bomexport myproj.xml
.bomexport path/to/myproj.xml
.bomexport /path/to/myproj.xml
```

## .break

```
.break [<notification_group_name>]
```

.break コマンドを使用して、ジョブを再開するまで一時停止します。システムが .break コマンドを持つステップを見つけると、その実行は完了し、結果的に停止します。「再始動」アイコンを使用してジョブを再開し、.break ステップ後のステップを続けます。

.break コマンドがチェーン・ジョブで発生する場合、システムはチェーン・ジョブを停止しますが、制御を呼び出しジョブに戻し、ステップの処理を継続します。

アクセス・グループをコマンドのオプション引数として含めることができます。その場合、システムはジョブを停止するときに、指定されたアクセス・グループに E メール・メッセージを送信します。

## .bset

```
.bset env "<VarName>=<Value>" [...]  
.bset selector <SelectorName> [(SelectorSnapshotName)]  
.bset server <ServerName>  
.bset buildserver <ServerName>
```

.bset コマンドは、ジョブの実行中にプロジェクト設定を一時的に変更します。

注: .bset コマンドは、後のステップで使用される基本ビルド環境変数に影響します。前の .bset コマンドに依存する .bset コマンドを、同じステップ内に指定しないでください。例えば、次のような組み合わせのステップを、同じステップ内に指定しないでください。

```
.bset env "VAR1=VALUE1"  
.bset env "VAR2=$VAR1"
```

このコマンドには以下のオプションがあります。

- **env** は、実行中のジョブの 1 つ以上のプロジェクト環境変数の値を変更します。この変更は、現行ステップでただちに有効になります。この方法でのコマンドの使用について詳しくは、427 ページの『ジョブ中のビルド・タグの変更』を参照してください。まだ存在しない変数を設定することができます。.bset コマンドで設定された値は、ジョブ・レコードに書き込まれます。プロジェクト環境変数またはステップ環境変数でも定義されている変数に新規の値を設定した場合、その新規の値は、ジョブの間だけ有効になります。プロジェクトまたはステップの環境変数は変更されません。
- **selector** は、ジョブ中にプロジェクト・セレクターを変更します。新規のセレクターは、次のステップで有効になります (現行ステップに関する何らかのインライン・ステップが含まれる場合があります)。この新規のセレクターは、明示的なセレクター設定を持たないステップにのみ使用されます。このステップのセレクターは「プロジェクト・デフォルト」に設定されます。インライン・ステップで **selector** オプションを使用すると、インライン化されたプロジェクト内の後続のすべてのステップ (インライン・ステップでインライン化されるステップなど) に影響が及ぼされます。呼び出しステップや、呼び出しステップと同じレベルまたは上位レベルにあるその他のステップには、影響は及びません。

明示的なセクター設定を持つステップは影響を受けません。

特定のセクターのスナップショットを指定するには、オプションの `[(SelectorSnapshotName)]` を使用します。

- **server** は、ジョブ中にデフォルトのプロジェクト・サーバーを変更します。新規のサーバー設定は、次のステップで有効になります (インライン・ステップは含まれません)。ステップは、明示的なセクター設定を持たないため、指定されたサーバーのみを使用します。

バックチック構文: バックチックを使用して、サーバー名をコマンドの出力に設定することができます。例えば、次のコマンドは、`SelectAServer.sh` スクリプトを実行し、その出力を `.bset server` コマンドのサーバー名として提供します。

```
.bset server `SelectAServer.sh`
```

注: `.bset` をバックチック、およびゼロ以外の終了コードを持つターゲット・スクリプト出力とともに使用すると、MJC はエラー・メッセージ `Exception communicating with endpoint server` を報告します。

- **buildserver** は、ジョブ中にデフォルトのプロジェクト・サーバーを変更します。新規のサーバー設定は、次のステップで有効になります (何らかのインライン・ステップが含まれます)。また、`.bset` がループ・シナリオで適用される場合、この設定は次のループでも有効になります。これは、インライン化されたステップ、次のループ・ステップ、およびプロジェクト・レベルのステップに適用されるため、`server` オプションとは異なります。

注: Perl エンジンの場合、`.bset server` および `buildserver` はインライン・ステップに影響しません。

重要: スレッド化ステップでは複数の `.bset` コマンドを使用しないようにしてください。

## **.buildstatus**

```
.buildstatus <result>
```

このコマンドは、ビルド終了後に、ビルド・ステップの結果とは無関係に強制的にビルドを指定の `<result>` になるようにします。

`<result>` に指定できる有効な値は次のとおりです。

- P**      結果を「成功」に設定します。
- F**      結果を「失敗」に設定します。
- W**      結果を「警告」に設定します。
- B**      結果を「停止」に設定します (`.break` コマンドを指定した場合と同様)。

## **.date**

```
.date <conversion_specifier>
```

プロジェクトの実行時に現在の日時情報を生成するには、1 つ以上の変換指定子文字を引数として指定した `.date` コマンドを使用します。

.date コマンドおよびその引数は、環境内の環境変数として定義する必要があります。その後、その環境変数をプロジェクトまたはステップに割り当てることができます。

例えば、環境内で環境変数 DayofWeek を定義し、それに .date %A という値を割り当てます。そしてその環境変数をプロジェクトまたはステップに割り当てます。プロジェクトが水曜日に実行された場合、ジョブは環境変数 DayofWeek に Wednesday というテキストを割り当てます。

**重要:** ステップの「コマンド」フィールドで .date コマンドを直接参照することはできません。

**注:** .date コマンドは、すべてのステップを再評価します。特定の日付を保持するには、環境変数 .date と .bset を組み合わせて使用します。例えば、  
\$ORIGTIMESTAMP = .date %d-%b-%Y.%H:%M:%S を使用した後に .bset 環境変数 "TIMESTAMP=`echo \$ORIGTIMESTAMP`" を使用します。

.date コマンドは POSIX の strftime 関数を基にして作成されており、ANSI C89 規格で指定された変換指定子を受け入れます。次の表に、変換指定子の日時値を示します。

**注:** すべての変換指定子が、ロケール間およびオペレーティング・システム間で移行可能なわけではありません。 .date コマンドを使用するサーバーのオペレーティング・システムおよびロケールで、.date コマンドの結果をテストしてください。

日付変換指定子	説明
%a	現在のロケールに応じた省略された週日名。
%A	現在のロケールに応じた完全な週日名。
%b	現在のロケールに応じた省略された月名。
%B	現在のロケールに応じた完全な月名。
%c	現在のロケールの、設定済みの日時表記。
%d	10 進数で表した月の日付 (01 から 31 の範囲)。
%H	24 時間クロックを使用した、10 進数で表した時間 (00 から 23 の範囲)。
%I	12 時間クロックを使用した、10 進数で表した時間 (01 から 12 の範囲)。
%j	10 進数で表した、年の日付 (001 から 366 の範囲)。
%m	10 進数で表した月 (01 から 12 の範囲)。

日付変換指定子	説明
%M	10 進数で表した分 (00 から 59 の範囲)。
%p	指定された時間値に応じて、または現在ロケールの対応するストリングに応じて、『AM』 または 『PM』。正午は 『pm』、真夜中は 『am』 と処理されます。
%S	10 進数で表した秒 (00 から 61 の範囲)。
%U	10 進数で表した、現在の年の週番号で、範囲は 00 から 53。最初の日曜日が週 01 の最初の日。
%w	10 進数で表した曜日。範囲は 0 から 6 で、日曜日は 0。
%W	10 進数で表した、現在の年の週番号で、範囲は 00 から 53。最初の月曜日が週 01 の最初の日。
%y	10 進数で表した、世紀なしの年 (範囲は 00 から 99)。
%Y	10 進数で表した、世紀を含む年。
%Z	時間帯、名前、または省略語。
%%	リテラル 『%』 文字。

## .defect

**説明** `.defect` コマンドを使用して、不良トラッキング・アプリケーションのアダプターをプロジェクト・ステップに追加します。不良アダプターは **Build Forge** オブジェクトの 1 つであり、不良トラッキング・アプリケーションのアダプター・テンプレートを基にしています。ステップのアダプター・コードは、プロジェクトの実行時に実行されます。

### 構文

```
.defect <adaptor_name> [entry_name]
```

`<adaptor_name>` が必要です。これは、管理コンソールのアダプターに割り当てられた名前です。`<adaptor_name>` の大/小文字は、管理コンソールで使用されている大/小文字と一致させる必要があります。

アダプター・テンプレートに複数のインターフェース機能がある場合、`entry_name` オプションを使用して実行する機能を指定してください。`entry_name` は、アダプター・テンプレートの<インターフェース>エレメントに指定した名前属性と一致させる必要があります。`entry_name` で指定した<インターフェース>・エレメントが存在しない、または見つからない場

合、代わりにデフォルトの<インターフェース>・エレメントが実行されます。次の例での項目名は `DefectFunction` です。

アダプター・リンクを使用している場合、アダプターが自動的に呼び出され、アダプター・テンプレートの最初のインターフェース機能が実行されます。アダプター・テンプレートの別のインターフェースを実行するには、実行するインターフェースでデフォルト属性を `true` (`default="true"`) に設定してください。

#### 例

```
.defect MyClearCaseQuestAdaptor
.defect MyClearCaseQuestAdaptor DefectFunction
```

**メモ** アダプターを作成、またはアダプター・リストを表示するには、「プロジェクト」 > 「アダプター」を選択します。

Build Forge 製品で提供されるアダプター・テンプレートは、次の場所にあります。

```
<bfinstall>/interface
```

## .drill

```
.drill [through]
    <"var1,var2,var3"|${EnvVar}|~r[p] Register>
    [grouped by] "{}"
    [separated by] ","
    [exec] "Command $1 $2"
```

`.drill` コマンドによりコマンドをループすることができ、一連の値の各メンバーに対してコマンドを一度実行します。値はコマンド行で指定するか、環境変数またはレジスターから引き出すことができます。システムが `.drill` コマンドを実行する場合、システムは `.drill` 構文を使用して一連のコマンド行を構成し、実行するためにエージェントに送信します。

例えば、コマンド `.drill "A,B,C,D" "echo value $1"` は以下のコマンドを作成します。

```
echo value A
echo value B
echo value C
echo value D
```

## グループ化

`$n` 構文を使用して、値をグループ化し、各グループで複数の値を参照することができます。`$1` がグループ内の最初の値を参照し、`$2` がグループ内の 2 番目の値を参照する、などです。例えば、`.drill through "(A,B,C,D,E),(B,C,D,E,F),(C,D,E,F,G)" grouped by "(" separated by "," exec "echo 1[$1] 2[$2] 3[$3] 4[$4] 5[$5]"` は、以下のコマンドを作成します。

```
echo 1[A] 2[B] 3[C] 4[D] 5[E]
echo 1[B] 2[C] 3[D] 4[E] 5[F]
echo 1[C] 2[D] 3[E] 4[F] 5[G]
```

**注:** デフォルトのグループ化文字はありません。デフォルトの分離文字はコンマです。`grouped by` を指定しない場合、システムでは、提供された値は分離文字で分離

されたものとして調べられ、そのような各ストリングは単一値とみなされます。例えば、コマンド `.drill "(A,B),(C,D)" "echo $1 $2"` は以下のコマンドに解決されます。

```
echo (A 2
echo B) 2
echo (C 2
echo D) 2
```

## データ・ソース

ループするデータを `.drill` コマンドが取得する際に指定できる、いくつかのオプションがあります。このコマンドの最初のパラメーターはデータ・ソースです。オプションのコマンド・ワード `"through"` を含めて、データ・ソースを示すことができます。

- ・ コマンド行にデータを明示的にリストすることができ、以下に示すコマンドでは、値 `one`、`two`、および `three` をループします。

```
.drill through "one,two,three" exec "echo $1"
```

- ・ データは環境変数から引き出すことができます。以下のコマンドでは、環境変数 `FILENAMES` はコンマで区切られたファイルのリストであると想定され、`DOS` コマンドを使用してリスト内のすべてのファイルが削除されます。

```
.drill through $FILENAMES exec "del $1"
```

- ・ データはレジスターまたはプロジェクト・レジスターから引き出すことができます。`RegisterA` にコンマで区切られたファイル名のリストが含まれる場合、`Linux` システムに発行された以下のコマンドは各ファイルの内容を書き出します。

```
.drill -r RegisterA exec "cat $1"
```

以下の例は同じことを実行しますが、プロジェクト・レジスターを使用しています。

```
.drill -rp ProjectRegisterA exec "cat $1"
```

## **.edit**

```
.edit /<search_expression>/<replace_expression>/ [<relative_path>/]file [file ...]
```

`.edit` コマンドを使用して、1 つ以上のファイル内のテキスト・ストリングを検索および置換します。`.edit` コマンドは、指定した各ファイルのすべての行で、ストリングの最初のインスタンス (`search_expression`) を置き換えます。ユーザーが相対パスを指定しない限り、ファイルはステップの作業ディレクトリーにあると想定されます。

`.edit` コマンドは、置換パターンでの `()` サブストリング選択および `¥N` 置換の使用を含む、マッチングおよび置換用の標準 `POSIX` 正規表現を実装します。

`.edit` コマンドは、デフォルトで `POSIX` の拡張正規表現構文を使用します。エージェントが **Perl 互換正規表現サポート** を使用してコンパイルされている場合は、置換表現の後に、`PCRE` 構文を指定する文字「`p`」を続けます。

円記号をエスケープ文字として使用する場合、エージェント・ホストのオペレーティング・システムに応じて異なる処理が必要になります。



- UNIX および Linux では、式はエージェントの処理によって 2 回解釈されるため、1 つの円記号を表すには 4 つの円記号を使用します。例えば、リテラルのピリオド文字と一致させるためには、4 つの円記号と 1 つのピリオドを使用します。

```
¥¥¥¥.
```

- Windows の場合、同じリテラルのピリオドと一致させるために必要な円記号は 2 つのみです。

```
¥¥.
```

区切り文字をエスケープすることはできません。区切り文字をエスケープした方がよいと思われる場合は、代替の区切り文字を使用してください。例: 文字 `abc` のみを含むファイル `abc.txt` があります。これをストリング `a/c` で置き換えるとします。

- 誤った方法 - 以下の場合、区切り文字がエスケープ文字に優先するため、ストリングは変更されません。

```
.edit /abc/a¥/c/ abc.txt
.edit /abc/a¥¥/c/ abc.txt
.edit /abc/a¥¥¥¥/c/ abc.txt
```

- 正しい方法 - 区切り文字を変更すると `a/c` を置き換えることができます。

```
.edit "abc"a/c" abc.txt
```

注: 1 つ以上のファイル名を、ワイルドカードなしで明示的にリストする必要があります。

例えば、次のコマンドは、`drivermakefile` という名前のファイルで、`winXPdriver` および `win2000driver` などのストリングを置換します。

```
.edit /win.*driver/linuxdriver/ drivermakefile
```

`.edit` コマンドは `.strsub` コマンドと似ていますが、以下の点が異なります。

- `.strsub` コマンドは、大規模なテキスト・ファイルまたは複数ファイルでの置換を `.edit` より速く実行できます。
- `.edit` コマンドは正規表現検索および置換を実行できます。
- `.edit` コマンドは、各ファイルのすべての行で、ストリングの最初のインスタンスのみ (`search_expression`) を置き換えます。
- `.strsub` コマンドは、各ファイルのすべての行で、ストリングのすべてのインスタンス (`source`) を置き換えます。

## **.email**

```
.email <recipient>
```

このコマンドは、`step_email` メール・テンプレートを使用して、E メールを *recipient* に送信します。指定されたアドレスは Build Forge ユーザーであることが理想ですが、`.email` は有効なすべての E メール・アドレスに同様に送信を行います。

## **.expand**

```
.expand <var1 var2 ... varn>
```

このコマンドは、ブランク・スペース区切りの変数を受け入れ、その変数をサーバー（エージェント）上で解析し、展開した変数を下記のように表示します。

```
variable[1] = expanded value  
variable[2] = expanded value  
...  
variable[n] = expanded value
```

## **.export**

```
.export [path_name][file_name]
```

**.export** コマンドは、呼び出し側プロジェクトのプロジェクト定義を、ステップの作業ディレクトリーにある XML ファイルに保存します。XML ファイルは、プロジェクトとそのステップを説明します。サーバーなど、他の関連するオブジェクトは記述されません。

エクスポート後の XML ファイルを使用して、プロジェクト定義を管理コンソールにインポートすることができます。

**.export** コマンドは、オプションのパスまたはファイル名、あるいはその両方を取ることができます。パスは相対パスでなければなりません。この相対パスは、ステップの作業ディレクトリーを基準とします。

ファイル名を指定しない場合、ファイル名は現行のプロジェクト・タグ \$BF\_TAG.xml を使用して作成されます。

## **.get**

```
.get server:[[<relative_path>/]file/]file [[<relative_path>/]file/]file]
```

**.get** コマンドを使用して、1 つの論理サーバーから別の論理サーバーにファイルを転送します。**.get** 操作は、現行サーバー・パスから実行され、指定されたサーバー・パスからファイルを取得します。宛先パス名は、ステップの現行作業ディレクトリーに対する相対パス名です。ソース・パス名は、指定されたサーバーの基本パスに対する相対パス名です。サーバーでは、ファイルに対して **.get** 操作を許可する論理サーバーを指定する必要があります。詳しくは、431 ページの『サーバー・リソースでのファイルのコピーの有効化』を参照してください。単一ファイルのみ転送できます。

パスの指定には、環境変数を組み込むことができます。この機能により、特定のジョブが使用するパスに対しての相対ファイルを指定できます。ジョブが使用するパスの説明については、485 ページの『ジョブ用の作業ディレクトリー』を参照してください。

使用しているサーバー名にスペースが含まれている場合は、そのサーバー名とそのファイルへのパスを引用符で囲みます。例: **.get** [**<relative\_path>/**]file server:[**<relative\_path>/**]file]"

注: ファイル転送を行う前に、使用するパスが存在することを確認してください。

この転送は高速ではないため、大きなファイルを転送するには他の方式を選択したほうが良い場合があります。予想される転送速度は、1 秒あたり 40 KB 以下です。70 MB のファイルを転送するには、45 分から 1 時間程度かかる場合があります。

注: 宛先ファイルが既に存在する場合は、警告なしで上書きされます。

Windows ファイル・パス分離文字である円記号 (¥) は、Rational® Build Forge® で使用されているエスケープ文字と同じであるため、dotcommand の構文解析時に、コマンドが「エスケープ」を意味しているかどうかを判別するのは困難です。

Windows ファイル・パス分離文字とエスケープを区別するには、以下の文字を使用してください。

- エスケープを有効にするには、/E を入力します。
- エスケープを無効にするには、/e を入力します。

次の例は .put コマンドを示しています。

```
.put[/E/e] [<relative_path>/]file server:[[<relative_path>/]file]
```

/E と /e のいずれも指定しない場合、システムはヒューリスティック方式を使用して、dotcommand に Windows ファイル・パスのように見える何かが含まれているかどうかを確認します。ファイル・パスが検出された場合、エスケープは無効になります。そうでない場合、エスケープは有効になります。

## .load

```
.load [-o] [-e] [-v] [-j] [<relative_path>/]<filename>  
.load -r|-p <registername>  
.load -s <command name>
```

.load コマンドは XML ファイルからプロジェクトをロードし、ロードしたプロジェクトのステップを現行プロジェクトに追加します。ロードしたプロジェクトのステップは、.load コマンドを実行したステップの次に追加され、これにより、プロジェクトは実行時に動的にステップを作成およびロードすることができます。オプションを使用し、.load コマンドで、レジスターまたはコマンドの出力からデータを引き出すことができます。

.load コマンド用の XML ファイルを作成する場合は、既存プロジェクトのエクスポート・ファイルを雛形として使用することをお勧めします。システム内でプロジェクトを作成してから、エクスポートして .load コマンドで使用することもできます。このトピックには、サンプル XML コードが含まれます。

.load コマンドでロードされたステップは、インライン化またはチューニングされたプロジェクトへの参照を含むことができます。デフォルトでは、システムは XML ファイル内にあるインライン化されたプロジェクトの定義を検索し、そのステップをロードします。インライン化されたプロジェクト定義をデータベースから取得する方法については、このトピックで後述する -e オプションを参照してください。成功チェーンまたは失敗チェーンのプロジェクトの場合、システムは常に、データベースでプロジェクト定義を検索します。

注: .load から実行される JPO ステップの場合、Else インラインが指定されたプロジェクトまたはライブラリーは、すでにシステム内に存在するプロジェクトまたはライブラリーである必要があります。そうでない場合、そのインラインは実行されません。

## XML ファイル内の複数プロジェクト

システムはインライン化されたプロジェクトをそれらの呼び出しプロジェクトとともにエクスポートするため、XML ファイルにはいくつかのプロジェクトが含まれる場合があります。 .load コマンドは、ファイル内で primary とラベルが付けられたプロジェクトを実行します。このプロジェクトには、 <project> エlement 上に属性 primary="1" があります。

## コマンド・オプションおよびパラメーター

最も単純なコマンド・フォームは .load <filename> です。ファイル名の前に、オプションの (ジョブ・ディレクトリーに対する相対) パス名を含めることができます。例えば、次のコマンドは、

```
.load ../../project.xml
```

サーバー・ディレクトリー (そのプロジェクトとジョブ・ディレクトリーが含まれるディレクトリー) からファイル project.xml をロードし、ステップのパス・プロパティーが "/" (デフォルト) であると想定します。

注: 通常のステップがインライン化されたプロジェクトを起動する場合、システムはデータベースにアクセスして、そのプロジェクトの現在の定義を取得します。 .load コマンドによってインポートされたステップがインライン化されたプロジェクトを起動する場合、システムは XML ファイル内でインライン化されたプロジェクトの定義を探します。この状態を回避する方法については、以下の -e オプションの説明を参照してください。

注: バージョン 7.1 より前の Build Forge から生成された XML のステップはサポートされておらず、これらのElementは ID ではなく名前参照されるため、失敗する可能性が高くなります。

注: .load ではインポート・ユーティリティーを使用するため、インポート・ユーティリティーの設定に従ってアクセス・グループ所有権を適用します。詳しくは、531 ページの『インポート・オブジェクトへのアクセス・グループの割り当て方法』を参照してください。

このコマンドには以下のオプションがあります。

### -r または -p

これらのオプションにより、システムはレジスターからステップをロードします。これらのオプションを指定してコマンド行を使用します。

```
.load -r|-p <registername>
```

-r オプションは通常のレジスターからステップをロードしますが、-p オプションはプロジェクト・レジスターからステップをロードします。プロジェクトの以前のステップでレジスターに日付を作成してから、このコマンドを使用して、そのレジスターからステップをロードすることができます。

- s このオプションにより、システムはコマンドを実行し、そのコマンドの出力をロード対象のデータとして使用します。次のコマンド行を使用します。  
`.load -s <command name>`
- e オプションが設定されている場合、システムは、ロードされた XML ファイルからではなく、データベースからインライン化されたプロジェクトを取得します。これは値 `chainID` を、データベース内のプロジェクト ID への参照として処理します。これにより、XML ファイルは、XML ファイル内のインライン化されたプロジェクトではなく、最新バージョンのインライン化されたプロジェクトを参照したり、XML ファイルに含まれないプロジェクトを参照したりすることができます。
- o オプションを使用し、XML ファイル内でインライン化されたプロジェクトを使用不可にします。このオプションが使用される場合、システムはメイン・プロジェクト内にあるインライン化されたプロジェクトを無視します。インライン化されたプロジェクトへの参照が含まれるステップは、コマンドは実行しますが、インラインは無視します。
- j オプションは、XML ファイル内の最後のステップ・セットがスレッド化され、`.load` コマンドに続くステップもスレッド化される場合に使用します。-j オプションは、最後にスレッド化されたステップを結合ステップに変えます。それ以外の場合、スレッド化されたステップは、`.load` コマンドの後の、スレッド化されたステップ・ブロックの一部になります。
- v 表示用にディスプレイ端末 (STDOUT) にロードされる XML ファイルの内容を送信します。

## サンプル XML

以下の例に、`.load` コマンドとともに使用する XML ファイルを示します。XML は、`HelloWorldPlusInline` というプロジェクトをエクスポートすることによって作成されます。

以下のサンプル XML の詳細に注意してください。

- XML には 2 つの `<project>` エLEMENTが含まれます。
- XML の最初のプロジェクトは基本プロジェクトで、属性 `name="HelloWorldPlusInline"` および `primary="1"` を持ちます。
- XML の 2 番目のプロジェクトは `Sleepytime` と呼ばれ、それが基本ではないことを示す属性 `primary="0"` を持ちます。
- `HelloWorldPlusInline` の最初のステップは、エコー・コマンドと `chainID` 属性が含まれる `EchoHelloWorld` というステップです。`chainID` 属性には値 2 が含まれ、システムは ID 2 を持つプロジェクト (`Sleepytime` プロジェクト) をインライン化することを示します。

注: ステップ属性 `inline` は無視してください。これは推奨されない属性で、使用されなくなりました。すべてのステップには、値 `N` を持つこの属性があります。ステップにインライン化されたプロジェクトがあるかどうかを判別する場合、属性 `chainID` を検索してください。値 `chainID` は、プロジェクトの ID を表します。デフォルトでは、システムは XML ファイル内でインライン化されたプロジェクトを検索しますが、`.load` コマンドで `-e` オプションを使用する場合、

システムはその値をデータベース内でプロジェクト ID として扱います。これにより、インライン化されたプロジェクトを内部に組み込むことなく、独自の .load ファイルを作成できます。

注: .load を、Build Forge バージョン 7.0.x 以前で生成された XML ファイルとともに使用することはサポートされておらず、古い構文では UUID ではなく名前でもオブジェクトを参照するため、適切に機能しません。

- 各プロジェクトには id 属性があります。この ID 値は、データベース内のプロジェクトの ID と同じです。プロジェクト ID のリストは、インストール・ディレクトリーから以下のコマンドを実行することによって取得できます。

```
bfexport -l
```

- commandStore 属性には、特定の文字が XML 形式でエスケープされた状態で、<command> エレメントの内容のコピーが含まれています。例えば、<command> 内には "、'、&、<、>、または改行などが含まれていることがあります。commandStore 内では、これらの文字はそれぞれ、&quot;、&apos;、&amp;、&lt;、&gt;、および &#10; と表現されます。

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<buildforge schema="7.115014" comment="">
  <project access="6" active="Y" name="HelloWorldPlusInline" primary="1"
    selectorId="Choose_local" maxthread="0" increment="Y" tagsync="0"
    buildclass="Production" sticky="N" envId="0" tag="BUILD_$B" id="19" exclusive="0">
    <tagvar autoincrement="Y" name="B" id="1">2</tagvar>
    <step absolute="N" failwait="N" selectorId="" dir="/" broadcast="N"
      timeout="300" id="1" passwait="N" inline="N" threadable="N" chainId="2"
      access="6" active="Y" passnotify="0" description="EchoHelloWorld" onfail=" "
      failnotify="0" envId="0" commandStore="echo Hello World">
      <command>echo Hello World</command>
    </step>
    <step absolute="N" failwait="N" selectorId="" dir="/" broadcast="N"
      timeout="300" id="2" passwait="N" inline="N" threadable="N" access="6"
      active="Y" passnotify="0" description="export proj to build and server folders"
      onfail=" " failnotify="0" envId="0"
      commandStore=".export $BF_PROJECTNAME_PHYS.xml&#10;copy /Y $BF_PROJECTNAME_PHYS.xml ..¥..">
      <command>.export $BF_PROJECTNAME_PHYS.xml
copy /Y $BF_PROJECTNAME_PHYS.xml ..¥..</command>
    </step>
  </project>
  <project access="6" active="Y" name="Sleepytime" primary="0"
    selectorId="Choose_local" maxthread="0" increment="Y" tagsync="0"
    buildclass="Production" sticky="N" envId="0" tag="SLEEP_$B" id="2" exclusive="0">
    <tagvar autoincrement="Y" name="B" id="1">21</tagvar>
    <step absolute="N" failwait="N" selectorId="" dir="/" broadcast="N"
      timeout="300" id="1" passwait="N" inline="N" threadable="N" access="6"
      active="Y" passnotify="0" description="Sleep, perchance to dream" onfail=" "
      failnotify="0" envId="0" commandStore=".sleep 0">
      <command>.sleep 0</command>
    </step>
  </project>
  <class maxdays="0" access="1" entranceprojectId="1" name="Production" keepfiles="B"
    deletechangedata="N" purgeprojectId="2" exitProjectId="5" candidates="AnyBuild "
    maxbuilds="0"></class>
  <selector operator="" required="" access="6" value="" name="Choose_local"
    selectorId="" property=""></selector>
</buildforge>
```

**.lock**

.lock



.lock コマンドにより、システムはジョブの完了後にジョブをロックします。これにより、ジョブが、そのクラスのプロパティに基づいて自動的に削除されることはなくなります。また、ロックされた実行は「ジョブ (Jobs)」 > 「完了」タブにはリストされず、代わりに「ロック」タブに表示されます。コマンドはパラメーターを取りません。コマンドは、それが使用されるジョブをロックします。

## .mkdir

```
.mkdir <relative_path>
```

.mkdir コマンドは、ディレクトリーを作成します。<relative\_path> パラメーターは、現行ステップ・ディレクトリーからの相対パスとして解釈されます。パス名を指定したディレクトリーがない場合は、作成されます。絶対パスおよびドライブ名を含むパス (C:¥ など) は使用できません。

## .monitor

```
.monitor [-c] [-w] <interval> [<relative_path>/]<filename>
```

.monitor コマンドにより、システムはプロジェクトを一時停止し、ファイル・サイズ変更の終了時を確認するためにファイルを監視します。ステップがこのコマンドを実行すると、システムは示されたファイルを検査し、その後、そのファイルを<interval> 秒ごとに再検査します。ファイル・サイズが 2 つの間隔の間で変更に失敗すると、システムは続けて次のステップを行います。

-c オプションを使用する場合、システムはモニター対象ファイルが変更を終了したと判断した後、そのファイルの内容をステップ・ログに書き込み、続けて次のステップを行います。

ファイルが存在しない場合、システムは待機しないで、最初の間隔のすぐ後に続行して作業を行います。-w オプションを使用して、システムに待機させてファイルが作成された後で、モニター処理を開始してください。

## .pack

**説明** .pack コマンドを使用して、パッケージ化アプリケーションのアダプターをプロジェクト・ステップに追加します。パッケージ化アダプターは Build Forge オブジェクトの 1 つであり、パッケージ化アプリケーションのアダプター・テンプレートを基にしています。ステップのアダプター・コードは、プロジェクトの実行時に実行されます。

### 構文

```
.pack <adaptor_name> [entry_name]
```

<adaptor\_name> が必要です。これは、管理コンソールのアダプターに割り当てられた名前です。<adaptor\_name> の大/小文字は、管理コンソールで使用されている大/小文字と一致させる必要があります。

アダプター・テンプレートに複数のインターフェース機能がある場合、entry\_name オプションを使用して実行する機能を指定してください。entry\_name は、アダプター・テンプレートの<インターフェース>エレメントに指定した名前属性と一致させる必要があります。entry\_name で指定した<インターフェース>・エレメントが存在しない、または見つからない場



合、代わりにデフォルトの<インターフェース>・エレメントが実行されます。次の例での項目名は `PackageFunction` です。

アダプター・リンクを使用している場合、アダプターが自動的に呼び出され、アダプター・テンプレートの最初のインターフェース機能が実行されます。アダプター・テンプレートの別のインターフェースを実行するには、実行するインターフェースでデフォルト属性を `true` (default="true") に設定してください。

#### 例

```
.pack MyPackagingAdaptor
.pack MyPackagingAdaptor PackageFunction
```

注: アダプターを作成、またはアダプター・リストを表示するには、「プロジェクト」 > 「アダプター」を選択します。

Build Forge 製品で提供されるアダプター・テンプレートは、次の場所にあります。

```
<bfinstall>/interface
```

## .pop

```
.pop [-p] <register_name> [+] [<relative_pathname>|-]
.pop [-p] <register_name> [>|>>]<register_name>
```

.pop コマンドは、レジスターの内容をファイル、ステップ・ログ、または別のレジスターに書き込みます。

オプションの -p パラメーターは、コマンドにプロジェクト・レジスターを参照させます。プロジェクト・レジスターは通常のレジスターとは分離されており、プロジェクト・レジスターはジョブの終了後も存続します。

以下の例に、.pop コマンドのさまざまな使用法を示します。

- .pop A data.txt

レジスター A が、ステップの作業ディレクトリーにあるファイル data.txt に書き込まれます。

- .pop ver +data.txt

レジスター ver の内容が、ファイル data.txt に追加されます。

- .pop Alpha

レジスター Alpha の内容が、ステップのログに書き込まれます。

- .pop ALPHA > BETA

レジスター BETA の内容が、レジスター ALPHA の内容と同じになります。

- .pop A >> B

レジスター A の内容が、レジスター B に追加されます。

注: レジスターをポップしても、空にはなりません。レジスターの内容を変更するには、.push コマンドを使用して、レジスターに新規値をプッシュします。

Windows ファイル・パス分離文字である円記号 (¥) は、Rational® Build Forge® で使用されているエスケープ文字と同じであるため、dotcommand の構文解析時に、コマンドが「エスケープ」を意味しているかどうかを判別するのは困難です。Windows ファイル・パス分離文字とエスケープを区別するには、以下の文字を使用してください。

- エスケープを有効にするには、/E を入力します。
- エスケープを無効にするには、/e を入力します。

次の例は .put コマンドを示しています。

```
.put[/E/e] [<relative_path>/]file server:[[<relative_path>/]file]
```

/E と /e のいずれも指定しない場合、システムはヒューリスティック方式を使用して、dotcommand に Windows ファイル・パスのように見える何かが含まれているかどうかを確認します。ファイル・パスが検出された場合、エスケープは無効になります。そうでない場合、エスケープは有効になります。

## .poptag

```
.poptag [-p]<registername>
```

.poptag コマンドは、現行タグを指定されたレジスターの内容で置き換えて変更します。

オプションの -p パラメーターは、コマンドにプロジェクト・レジスターを参照させます。プロジェクト・レジスターは通常のレジスターとは分離されており、プロジェクト・レジスターはジョブの終了後も存続します。

## .purge

```
.purge
```

ジョブの完了後にすぐにジョブがパージされるようにするフラグを設定するには、.purge コマンドを使用します。.purge コマンドの後で実行する .lock コマンドで、ジョブは保存されます。このコマンドを使用して、すべてのステップを正常に完了した場合にのみ保存されるジョブを作成することができます。このようなジョブを作成するには、.purge コマンドをプロジェクトの最初のステップにし、.lock コマンドを最後のステップにします。

## .push

```
.push [-p] [+]<register_name> [<relative_pathname> | -]
```

.push コマンドは、<relative\_pathname> の内容をレジスター <register\_name> に入れます。<register\_name> の現在の内容は置き換えられます。

置き換えるのではなく、付加するためには、<register\_name> の前に正符号 (+) を付けます。

レジスターを初期化するには、<relative\_pathname> の代わりにハイフンを使用します。

オプションの `-p` パラメーターは、コマンドにプロジェクト・レジスターを参照させます。プロジェクト・レジスターは通常のレジスターとは分離されており、プロジェクト・レジスターはジョブの終了後も存続します。

ステップの「絶対」プロパティーが使用可能になっていない限り、`<relative_pathname>` はプロジェクトまたはタグのパスに対する相対パスです。

以下の例では、ステップに対して、「絶対」プロパティーが使用可能になっていないことを前提としています。

- `.push ALPHA data.txt`

ステップの作業ディレクトリーにある `data.txt` ファイルの内容が、レジスター `ALPHA` に入れられます。

- `.push +B ..¥newdata.txt`

ステップの作業ディレクトリーの親ディレクトリーにある `newdata.txt` ファイルの内容が、レジスター `B` に付加されます。

- `.push ALPHA -`

レジスター `ALPHA` が初期化されます。

Windows ファイル・パス分離文字である円記号 (¥) は、Rational® Build Forge® で使用されているエスケープ文字と同じであるため、`dotcommand` の構文解析時に、コマンドが「エスケープ」を意味しているかどうかを判別するのは困難です。

Windows ファイル・パス分離文字とエスケープを区別するには、以下の文字を使用してください。

- エスケープを有効にするには、`/E` を入力します。
- エスケープを無効にするには、`/e` を入力します。

次の例は `.put` コマンドを示しています。

```
.put[/E/e] [<relative_path>/]file server:[[<relative_path>/]file]
```

`/E` と `/e` のいずれも指定しない場合、システムはヒューリスティック方式を使用して、`dotcommand` に Windows ファイル・パスのように見える何かが含まれているかどうかを確認します。ファイル・パスが検出された場合、エスケープは無効になります。そうでない場合、エスケープは有効になります。

## **.put**

```
.put [<relative_path>/]file server:[[<relative_path>/]file]
```

1 つの論理サーバーから別の論理サーバーへファイルを転送するには、`.put` コマンドを使用します。`.put` 操作は、現行サーバー・パスから実行し、指定されたファイルをリモート・サーバーに送信します。宛先パス名は、ターゲット・サーバーの基本パスに対する相対パス名です。ソース・パス名は、ステップの現行作業ディレクトリーに対する相対パス名です。リモート・サーバーでは、ファイルに対して `.put` 操作を許可する論理サーバーを指定する必要があります。詳しくは、431 ページの『サーバー・リソースでのファイルのコピーの有効化』を参照してください。単一ファイルのみ転送できます。

パスの指定には、環境変数を組み込むことができます。この機能により、特定のジョブが使用するパスに対しての相対ファイルを指定できます。ジョブが使用するパスの説明については、485 ページの『ジョブ用の作業ディレクトリー』を参照してください。

注: ファイル転送を行う前に、使用するパスが存在することを確認してください。

使用しているサーバー名にスペースが含まれている場合は、その名前を引用符で囲みます。

この転送は高速ではないため、大きなファイルを転送するには他の方式を選択したほうが良い場合があります。予想される転送速度は、1 秒あたり 40 KB 以下です。70 MB のファイルを転送するには、45 分から 1 時間程度かかる場合があります。

注: 宛先ファイルが既に存在する場合は、警告なしで上書きされます。

Windows ファイル・パス分離文字である円記号 (¥) は、Rational® Build Forge® で使用されているエスケープ文字と同じであるため、dotcommand の構文解析時に、コマンドが「エスケープ」を意味しているかどうかを判別するのは困難です。

Windows ファイル・パス分離文字とエスケープを区別するには、以下の文字を使用してください。

- ・ エスケープを有効にするには、/E を入力します。
- ・ エスケープを無効にするには、/e を入力します。

次の例は .put コマンドを示しています。

```
.put[/E/e] [<relative_path>/]file server:[[<relative_path>/]file]
```

/E と /e のいずれも指定しない場合、システムはヒューリスティック方式を使用して、dotcommand に Windows ファイル・パスのように見える何かが含まれているかどうかを確認します。ファイル・パスが検出された場合、エスケープは無効になります。そうでない場合、エスケープは有効になります。

## **.rem**

### **説明**

.rem ドット・コマンドはコメント用です。ステップでコマンドに関するメモを作成する場合に役立ちます。

### **例**

```
.rem ここにはコメントのテキストが入ります。  
.rem 「.rem」の後に続くテキストは、すべて無視されます。
```

## **.report**

Quick Report で作成するレポートの出力をジョブの BOM に追加します。Quick Report は、個別にライセンスの交付を受ける、Rational® Build Forge® のフィーチャーです。

注: Build Forge® 8.0 では、Java エンジン上で .report コマンドはサポートされません。

**説明** プロジェクトにレポートを追加するには、`.report` コマンドを使用します。  
`.report` コマンドはレポートを実行し、その出力をジョブの BOM に追加します。ジョブを実行するたびに、現行データを基にしたレポート結果が BOM に表示されます。

#### 構文

```
.report <report_name>
```

`<report_name>` の制約事項は次のとおりです。

- `<report_name>` は必須です。
  - 名前は大文字小文字が区別されます。
  - この名前にスペースが含まれている場合は、名前を引用符で囲ってください。例えば、"my\_report" のようにします。
- 指定する `<report_name>` は、公開レポートとして保存されている必要があります。
- BOM レポート・タイプはサポートされていません。

## .retag

```
.retag <new_tag>
```

ステップで `.retag` コマンドを使用し、ジョブ中にそのジョブのタグを変更します。変数またはコマンドを新規タグ値として使用できます。

## .retry

```
.retry <count> <command>
```

失敗したコマンドの再試行を許可するには、`.retry` コマンドを使用します。`.retry` コマンドは、コマンドを再試行する回数を指定する単一のカウンタ引数を取ります。実行するコマンドは引数の残りとして取られるので、`.retry` コマンドはステップの最終ドット・コマンドでなければなりません。以下のコマンドを例に取ります。

```
.retry 3 myscript.sh arg1 arg2 arg3
```

『myscript.sh arg1 arg2 arg3』を、失敗するまで最大 3 回そのステップを実行します。正常な状況に戻すコマンドを最初に呼び出すと、再試行処理は停止されます。

## .rget

```
.rget server:[<path>] [<path>]
```

`.rget` コマンドは `.get` コマンドに類似した動作をしますが、ディレクトリー・ツリー全体を再帰的にコピーします。ディレクトリー名をパラメーターとして提供する必要があります。例えば、次のコマンドは、

```
.rget winbuildserver1:config myconfig
```

サーバー winbuildserver 上のディレクトリー構成の内容を、現行サーバーの myconfig ディレクトリーにコピーします。

注: ファイル転送を行う前に、使用するパスが存在することを確認してください。

注:

- このコマンドでは環境変数を使用しないでください。
- このコマンドではシンボリック・リンクを使用しないでください。
- コマンドの使用を試みるには、指定のディレクトリーを事前に作成しておく必要があります。指定のディレクトリーが存在しない場合、このコマンドは失敗します。

Windows ファイル・パス分離文字である円記号 (¥) は、Rational® Build Forge® で使用されているエスケープ文字と同じであるため、dotcommand の構文解析時に、コマンドが「エスケープ」を意味しているかどうかを判別するのは困難です。

Windows ファイル・パス分離文字とエスケープを区別するには、以下の文字を使用してください。

- エスケープを有効にするには、/E を入力します。
- エスケープを無効にするには、/e を入力します。

次の例は .put コマンドを示しています。

```
.put[/E/e] [<relative_path>/]file server:[<relative_path>/]file]
```

/E と /e のいずれも指定しない場合、システムはヒューリスティック方式を使用して、dotcommand に Windows ファイル・パスのように見える何かが含まれているかどうかを確認します。ファイル・パスが検出された場合、エスケープは無効になります。そうでない場合、エスケープは有効になります。

## .rmdir

```
.rmdir <relative_path>
```

.rmdir コマンドは、<relative\_path> で指定されたディレクトリーを除去します。システムは、パス名で指定された基本ディレクトリーを、すべての内容および下位ディレクトリーを含めて除去します。

## .rput

```
.rput [<relative_path>] server:[<relative_path>]
```

.rput コマンドは .put コマンドに類似した動作をしますが、ディレクトリー・ツリー全体を再帰的にコピーします。ユーザーが提供する相対パスは、ファイルではなくディレクトリーでなければなりません。例えば、次のコマンドは、

```
.rput myconfig linuxserver5:feb2005
```

現行サーバーのディレクトリー myconfig の内容を、サーバー linuxserver5 の feb2005 ディレクトリーにコピーします。

注: ファイル転送を行う前に、使用するパスが存在することを確認してください。

ソース・パスはステップの作業ディレクトリーに対する相対パスであるため、そのステップの「絶対」プロパティの値に基づくプロジェクトおよびタグ・ディレクトリーを含む場合と含まない場合があります。宛先パスは、宛先サーバーの「パ



ス」プロパティーのみに対する相対パスです。システムによるパスの構成方法について詳しくは、485 ページの『ジョブ用の作業ディレクトリー』を参照してください。

注:

- このコマンドでは環境変数を使用しないでください。
- このコマンドではシンボリック・リンクを使用しないでください。
- コマンドの使用を試みるには、指定のディレクトリーを事前に作成しておく必要があります。指定のディレクトリーが存在しない場合、このコマンドは失敗します。

Windows ファイル・パス分離文字である円記号 (¥) は、Rational® Build Forge® で使用されているエスケープ文字と同じであるため、dotcommand の構文解析時に、コマンドが「エスケープ」を意味しているかどうかを判別するのは困難です。

Windows ファイル・パス分離文字とエスケープを区別するには、以下の文字を使用してください。

- エスケープを有効にするには、/E を入力します。
- エスケープを無効にするには、/e を入力します。

次の例は .put コマンドを示しています。

```
.put[/E/e] [<relative_path>/]file server:[<relative_path>/]file]
```

/E と /e のいずれも指定しない場合、システムはヒューリスティック方式を使用して、dotcommand に Windows ファイル・パスのように見える何かが含まれているかどうかを確認します。ファイル・パスが検出された場合、エスケープは無効になります。そうでない場合、エスケープは有効になります。

## .run および .runwait

注: Build Forge 8.0 の Java エンジンでは、.run および .runwait に対する条件 [<condition>] はサポートされません。条件に応じた起動にするには、代わりに条件付きステップを使用します。

```
.run [-c "<condition>"] "<ProjectName>" ["<ProjectSnapshotName>"]
```

```
.runwait [-c "<condition>"] "<ProjectName>" ["<ProjectSnapshotName>"]
```

.run コマンドと .runwait コマンドを使用して、チェーニングされたプロジェクトをステップ・コマンドから起動することができます。プロジェクトのスナップショットを指定するには、オプションの <ProjectSnapshotName> パラメーターを使用します。

これらのコマンドは、プロジェクトを起動した後の振る舞いに違いがあります。

- .run コマンドは指定されたプロジェクトをチェーンとして起動し、チェーンされたプロジェクトの環境変数継承の規則に従います。
- .runwait コマンドは、指定されたプロジェクトを起動します。起動したプロジェクトが完了する間、起動ステップは待機します。起動したプロジェクトが完了すると、システムは、起動したプロジェクトの完了状況に従って、起動ステップの結果値を pass または fail に設定します。



**重要:** `.runwait` を使用するステップを含むプロジェクトは、実行時に 2 つのジョブ・スロットを消費します。ジョブ・スロットを十分に使用できない場合、ステップはエラーで失敗します。

**重要:** `.run` コマンドまたは `.runwait` コマンドを使用して起動したプロジェクトでは、「ビルド結果」ページにチェーン・リンク・アイコンが生成されません。

## 条件付き起動

オプションの `-c` パラメーターを使用して、条件別の起動を実行することができます。条件には環境変数を使用することができます。条件には以下のいくつかの形式があります。

### ストリング比較

等号 (`=`) または非等号 (`!=`) 演算子を使用して、ストリングを評価することができます。比較が真と評価すれば、チェーンが起動されます。

### 数値比較

`<`、`>`、`<=`、`>=`、または `=` 演算子を使用し、2 つの数値を比較します。

### コマンド成功

`backtick` に囲まれたコマンドを、`-c` パラメーターの値として使用することができます。システムがコマンドを実行し、成功するとチェーンが起動されます。

## 例

```
.run "BuildWindowsDriver"
```

システムは `BuildWindowsDriver` プロジェクトを起動します。起動プロジェクトはすぐに次のステップで続行します。

```
.runwait "BuildWindowsDriver"
```

システムは `BuildWindowsDriver` プロジェクトを起動します。システムは、`.runwait` ステップで起動プロジェクトを一時停止します。`BuildWindowsDriver` プロジェクトが完了して成功すると、`.runwait` ステップの状況は `pass` に設定されます。

```
.run -c "$HOMEDRIVE=C:" "Simple Echo"
```

`HOMEDRIVE` 変数に値 `C` がある場合のみ、システムはプロジェクト `Simple Echo` を実行します。

このコマンドは以下のように (ステップ・ログの `EXEC` セクションに) ログ出力を生成します。

- `HOMEDRIVE` が `C` の場合:

```
.run Condition: 'C:' = 'C:' satisfied.
```

```
Queueing Project "Simple Echo" on server [WinBox].  
Queued Build 'BUILD_202' of project 'Simple Echo'.
```

- `HOMEDRIVE` が `C` でない場合:

```
.run -c "$HOMEDRIVE=C:" "Simple Echo"
```

```
.run Condition: 'D:' = 'C:' unsatisfied, no project queued.
```

ストリングに数値が含まれる場合、システムはストリングを数値的に比較することができます。例えば、以下の場合には、次に示すように処理します。

```
.runwait -c "a12b<c42d" "Simple Echo"
.run Condition: '12' < '42' satisfied.
Queueing Project "Simple Echo" on server [WinBox].
Waiting for .run build (4411) to complete.
.run build is now running.
.run build has finished.
Build 'BUILD_203' of project 'Simple Echo' completed.

.runwait -c "f43g<>h43i" "Simple Echo"
.run Condition: '43' <> '43' unsatisfied, no project queued.
```

以下の例に、コマンドを条件として使用する方法を示します。コマンドは引用符と backtick の両方で囲む必要があることに注意してください。

```
.run -c "`exit 1`" "Simple Echo"
Env .run encountered an error during variable expansion,
parameter [`exit1`] expanded to [].
Expansion returned non-zero exit, project will not be queued.

.run -c "`exit 0`" "Simple Echo"
Expansion returned zero exit, project will be queued.
Queueing Project "Simple Echo" on server [WinBox].
Queued Build 'BUILD_204' of project 'Simple Echo'.
```

.runwait を使用し、ビルドが失敗する場合、ログは次のように表示されます。

```
.runwait "Fail Build"
Queueing Project "Fail Build" on server [WinBox].
Waiting for .run build (4413) to complete.
.run build is now running.
.run build has finished.
Build 'BUILD_3' of project 'Fail Build' Failed, setting step status to fail.
```

## **.scan**

```
.scan [-v][-i <ignorepattern>] baseline | checkpoint
```

ジョブの BOM に保管されるデータを拡張するには、.scan コマンドを使用します。ステップの作業ディレクトリー内のファイルを、各ファイルの MD5 値とともに追跡します。

### **.scan baseline**

ステップの作業ディレクトリー内のすべてのファイルのリストを保管します。システムは、そのリストをカテゴリーとしてジョブの BOM に表示します。ジョブには基本的なコマンドが複数ありますが、コマンドが実行されると、それぞれがリストをステップの作業ディレクトリーの状態にリセットします。最後の BOM は、1 つの基本的なカテゴリーのみを表示します。

### **.scan checkpoint**

ジョブ内での最後の .scan baseline コマンドまたは .scan checkpoint コマンド以降の、すべての新規ファイル、変更済みファイル、および削除済み

ファイルのリストを保管します。システムは、BOM にリストを表示します。各 checkpoint コマンドにより、BOM 内に新規カテゴリが作成されます。

ジョブで最初の `.scan checkpoint` コマンドの前に、`.scan baseline` コマンドを使用する必要があります。`.scan baseline` コマンドより前にある `.scan checkpoint` コマンドは無視されます。

コマンド・オプション:

- v** 変更情報のコピーをジョブ・ログに記録します。
- i** 指定されたパターンと一致するディレクトリーを無視します。このパターンは、パスの最初、最後、またはディレクトリーの任意部分と突き合わせることができます。このオプションを使用して、変更リストからソース制御ディレクトリーを除去できます。

CVS の場合の例:

```
.scan -i CVS checkpoint
```

このコマンド例の場合、CVS ディレクトリーはレポートに含められません。

サブバージョンの場合の例:

```
.scan -i .svn baseline
```

`-i` とともに `-v` を使用した場合、システムは、ソース制御ディレクトリーには変更を記録しますが、BOM には変更を含めません。

**注:** 単一ステップで複数の `.scan` コマンドを使用しないでください。単一ステップで複数のコマンドを使用すると、システムは `.scan` コマンドの正確な出力を提供できません。

これらのコマンドの使用について詳しくは、476 ページの『`.scan` コマンドでのベースラインとチェックポイントの追加』を参照してください。

## **.semget**

```
.semget <semaphore_name>
```

ステップがこのコマンドを実行すると、システムは示された名前を持つセマフォが存在するかどうかを検査します。

- そのようなセマフォが存在しない場合、システムが作成し、ステップのジョブにそれを割り当てます。その後、次のステップの実行が続行されます。
- 他のジョブがすでにこのセマフォ名を要求している場合、他のプロジェクトがセマフォをリリースするまで、ジョブは `.semget` ステップを停止します。

このコマンドの使用について詳しくは、660 ページの『セマフォ (semaphore)』を参照してください。

## **.semput**

```
.semput <semaphore_name>
```

<semaphore\_name> という名前のセマフォを解放するには、`.semput` コマンドを使用します。このコマンドの使用について詳しくは、660 ページの『セマフォ (semaphore)』を参照してください。

## **.set**

```
.set env <EnvGroupName>[(SnapshotName)] "<VariableName>=<DesiredValue>" [...]
```

`.set` コマンドは、環境変数に値を割り当てます。追加の変数および値を指定することができます。各変数とその値を引用符で囲ってください。このコマンドは、スナップショットが指定されていない場合は、環境変数グループのデフォルト・スナップショットの変数を設定します。スナップショットを指定する場合には、環境変数グループ名とスナップショット名の間にスペースを入れずに、スナップショットを括弧で囲ってください。

注: このコマンドで設定される変数は、すでに存在していなければなりません。

環境変数のマスター・レコード を変更するには、`.set` コマンドを使用します。システムはプロジェクトを実行する際、マスター・レコードからプロジェクト環境変数のコピーを作成し、そのコピーをジョブ・レコードに保存してプロジェクトのデフォルトとして使用します。

実行されたステップは、マスター・レコードではなく、環境変数のジョブのコピーを使用します。その結果、`.set` を使用すると、次のような影響があります。

- `.set` コマンドをステップで実行した場合、デフォルトの ステップ環境変数を使用する後続のステップでは、その変更が認識されません。システムは、ステップのデフォルト環境変数のジョブのコピーを使用します。
- `.set` コマンドを指定の 環境変数で実行した場合、その環境変数を指定する後続のステップは、行われた変更を認識します。ステップで環境変数が指定されると、システムはその環境変数のマスター・レコードを読み取ります。これは、指定されたステップ環境変数がプロジェクト・デフォルトと同じ環境変数である場合にも当てはまります。
- `.set` コマンドで行った変更内容は、ジョブが終了した後も存続します。その後の実行では、直前に実行した `.set` コマンドで作成された値が使用されます。

このコマンドの使用について詳しくは、426 ページの『ジョブ・データの処理』を参照してください。類似コマンド 437 ページの『`.bset`』も参照してください。

## **.sleep**

```
.sleep <seconds>
```

ステップを一時停止する秒数を指定するには、`.sleep` コマンドを使用します。このコマンドは管理コンソールが処理するため、リモート・サーバーへの接続は作成されません。『`.sleep 0`』 コマンドを、プラットフォームに依存しないヌル・コマンドとして使用することもできます。

## .snapshot

呼び出し側プロジェクトの新規インスタンスを作成し、そのインスタンスをプロジェクトのスナップショットとしてデータベースに保管するには、`.snapshot` コマンドを使用します。プロジェクトのスナップショットは、実行可能なプロジェクトです。

**説明** `.snapshot` コマンドは、プロジェクトのスナップショットを作成します。その際、そのプロジェクトに関連付けられたオブジェクトについても、スナップショットまたはコピーを作成するように選択することができます。

スナップショットまたはコピーを作成するオブジェクトを指定するには、次の表の説明に従って `.snapshot` のオプションを使用します。どのオプションも指定しなかった場合は、プロジェクトの定義、ステップ、およびタグのみが含まれます。

スナップショット名は必須です。これは、そのプロジェクト・スナップショット・セットに固有の名前にする必要があります。スナップショット名は、スナップショットを作成する他のオブジェクトにも割り当てられます。

`.snapshot` を実行するプロジェクトが完了すると、プロジェクトのスナップショットが、呼び出し側プロジェクトの子として UI に表示されます。その他のスナップショット・オブジェクトも、それらの基本スナップショットまたは親オブジェクトの子として UI に表示されます。

### 構文

```
.snapshot -v <"snapshot_name"> [-c <"comment">] [-e[f]] [-s[f]] [-pI] [-pC]
[-a] [-t] [-r] [-g]
```

オプション	説明
- v <"snapshot_name">	プロジェクトのスナップショットの名前は必須です。スナップショット名は、そのプロジェクトに固有の名前にする必要があります。名前は引用符で囲む必要があります。
- c <"comment">	オプションのコメントをスナップショットの一部として保存します。コメントは引用符で囲む必要があります。
-e -ef	プロジェクトのスナップショットを作成する際に、プロジェクト環境変数およびステップ環境変数のスナップショットを作成します。  f オプションを追加すると、スナップショット環境変数が「組み込み」環境変数型を使用して組み込むすべての環境変数のスナップショットも作成されます。
-s -sf	プロジェクトのスナップショットを作成する際に、プロジェクト・セレクターおよびステップ・セレクターのスナップショットを作成します。  f オプションを追加すると、スナップショットのセレクターが「組み込み」セレクター・プロパティー型を使用して組み込むすべてのセレクターのスナップショットも作成されます。
-pI	プロジェクトのスナップショットを作成する際に、インライン・プロジェクトまたはインライン・ライブラリー、およびそれらのステップのスナップショットを作成します。  インライン化されたプロジェクトまたはライブラリーはステップによって起動され、ステップの完了後に実行されます。

オプション	説明
-pC	プロジェクトのスナップショットを作成する際に、チェーニングされたプロジェクトまたはチェーニングされたライブラリー、およびそれらのステップのスナップショットを作成します。  プロジェクトまたはライブラリーは、プロジェクトの成功または失敗条件、あるいはステップの成功または失敗条件によって起動することができます。スナップショットは、条件付きでチェーニングされるプロジェクトまたはライブラリーの両方のタイプに対して作成されます。
-a	プロジェクトのスナップショットを作成する際に、アダプター・リンクをコピーします。  アダプター・リンクは、プロジェクトにアダプターを追加し、アダプター・コードを実行します。
-t	プロジェクト・レベルおよびステップ・レベルで設定された成功および失敗の通知イベント用の通知テンプレートをコピーします。
-r	プロジェクトのスナップショットを作成する際に、プロジェクト・レジスターをコピーします。
-g	プロジェクト・タグ変数のタグ値をコピーします。  タグ変数は自動的にコピーされますが、それらの値は自動的にコピーされません。タグ値をコピーしなければ、1 にリセットされます。

## .source

**説明** .source コマンドを使用して、ソース・コード・アプリケーションのアダプターをプロジェクト・ステップに追加します。ソース・コード・アダプターは Build Forge オブジェクトの 1 つであり、ソース・コード・アプリケーションのアダプター・テンプレートを基にしています。ステップのアダプター・コードは、プロジェクトの実行時に実行されます。

### 構文

```
.source <adaptor_name> [entry_name]
```

<adaptor\_name> が必要です。これは、管理コンソールのアダプターに割り当てられた名前です。<adaptor\_name> の大/小文字は、管理コンソールで使用されている大/小文字と一致させる必要があります。

アダプター・テンプレートに複数のインターフェース機能がある場合、  
entry\_name オプションを使用して実行する機能を指定してください。  
entry\_name オプションは、アダプター・テンプレートの<インターフェース>エレメントに指定した名前属性と一致させる必要があります。 entry\_name で指定した<インターフェース>・エレメントが存在しない、または見つからない場合、代わりにデフォルトの<インターフェース>・エレメントが実行されます。次の例での項目名は By Date です。

アダプター・リンクを使用している場合、アダプターが自動的に呼び出され、アダプター・テンプレートの最初のインターフェース機能が実行されま

す。アダプター・テンプレートの別のインターフェースを実行するには、実行するインターフェースでデフォルト属性を `true` (`default="true"`) に設定してください。

#### 例

```
.source MyClearCaseAdaptor
.source MyClearCaseAdaptor "By Date"
```

**メモ** アダプターを作成、またはアダプター・リストを表示するには、「プロジェクト」 > 「アダプター」を選択します。

Build Forge 製品で提供されるアダプター・テンプレートは、次の場所にあります。

```
<bfinstall>/interface
```

## .stop

```
.stop <state of build>
```

`.stop` は、ビルドの処理を強制的に即時停止します。ビルドを強制終了する場合には、このコマンドを使用してください。 `states of build` に指定可能な値は、成功 (P)、失敗 (F)、警告 (W)、または中断 (B) です。

## .strsub

```
.strsub <source> <replacement> file [file ...]
```

`.strsub` コマンドを使用し、1 つ以上のテキスト・ファイル内の基本ストリング置換を実行します。システムはターゲット・ファイルで `<source>` ストリングをスキャンし、一致が見つかると、システムは `<source>` ストリングを `<replacement>` に置き換えます。 `.strsub` コマンドは、各ファイルのすべての行で、ストリングのすべてのインスタンス (`source`) を置き換えます。

`.strsub` コマンドはオペレーティング・システムで機能し、サーバーで使用可能な特定のコマンドに依存することはありません。

ファイル `about.c` 内のストリング `_VERSION_` を置換するには、次のようなコマンドを使用します。

```
.strsub _VERSION_ 2.34 about.c
```

1 つ以上のファイル名を、ワイルドカードを使用せずに、正確に指定する必要があります。例えば、次のようなコマンドは失敗します。

```
.strsub _VERSION_ 2.34 *.txt
```

ただし、コマンド内で変数を使用することはできるため、環境変数に `VERSION` 変数と `FILENAME` 変数が定義されている場合は、次のようなコマンドが機能します。

```
.strsub _VERSION_ ${VERSION} ${FILENAME}
```

**注:** コマンド内でパラメーターを分離するにはスペースを使用してください。

`.strsub` コマンドは `.edit` コマンドと似ていますが、以下の点が異なります。



- `.strsub` コマンドは、大規模なテキスト・ファイルまたは複数ファイルでの置換を `.edit` より速く実行できます。
- `.edit` コマンドは正規表現検索および置換を実行できます。
- `.edit` コマンドは、各ファイルのすべての行で、ストリングの最初のインスタンスのみ (`search_expression`) を置き換えます。
- `.strsub` コマンドは、各ファイルのすべての行で、ストリングのすべてのインスタンス (`source`) を置き換えます。

`.edit` コマンドは、デフォルトで POSIX の拡張正規表現構文を使用します。エージェントが Perl 互換正規表現サポートを使用してコンパイルされている場合は、置換表現の後に (代わりに PCRE 構文を使用することを示す) 文字「p」を続けます。

どちらの場合にも、表現はエージェントの処理によって 2 回解釈されます。そのため、通常は単一の円記号が使用されるすべての箇所でも 4 つの円記号が使用されます。例:

4 つの円記号が、リテラルのピリオドをエスケープしています。  
`¥¥¥¥.`

また、`/x` フラグを使用して円記号を非表示にすることもできます。

`.strsub/x`

## **.test**

**説明** `.test` コマンドを使用して、テスト・アプリケーションのアダプターをプロジェクト・ステップに追加します。テスト・アダプターは Build Forge オブジェクトの 1 つであり、テスト・アプリケーションのアダプター・テンプレートを基にしています。ステップのアダプター・コードは、プロジェクトの実行時に実行されます。

### **構文**

```
.test <adaptor_name> [entry_name]
```

`<adaptor_name>` が必要です。これは、管理コンソールのアダプターに割り当てられた名前です。`<adaptor_name>` の大/小文字は、管理コンソールで使用されている大/小文字と一致させる必要があります。

アダプター・テンプレートに複数のインターフェース機能がある場合、`entry_name` オプションを使用して実行する機能を指定してください。`entry_name` は、アダプター・テンプレートの<インターフェース>エレメントに指定した名前属性と一致させる必要があります。`entry_name` で指定した<インターフェース>・エレメントが存在しない、または見つからない場合、代わりにデフォルトのインターフェース・エレメントが実行されます。次の例での項目名は `TestFunction` です。

アダプター・リンクを使用している場合、アダプターが自動的に呼び出され、アダプター・テンプレートの最初のインターフェース機能が実行されます。アダプター・テンプレートの別のインターフェースを実行するには、実行するインターフェースでデフォルト属性を `true` (`default="true"`) に設定してください。

### **例**

```
.test MyTestAdaptor
```

```
.test MyTestAdaptor TestFunction
```

**メモ** アダプターを作成、またはアダプター・リストを表示するには、「プロジェクト」 > 「アダプター」を選択します。

Build Forge 製品で提供されるアダプター・テンプレートは、次の場所にあります。

```
<bfinstall>/interface
```

## .tset

```
.tset env "<VariableName>=<DesiredValue>" [...]
```

.tset コマンドは、ステップ中に一時的にプロジェクト設定を変更します。 .tset を使用して、まだ存在しない変数を設定することができます。

このコマンドは、現行ステップで有効になります。ステップ内のすべてのコマンド、およびステップで指定されたすべてのインラインに対して有効になります。ただし、この動作はスレッド化による影響を受ける場合があります。例:

1. ステップがスレッド化され、インラインが指定される。
2. インラインの最初のステップもスレッド化される。

この例では、.tset コマンドは最初のステップ (1) では有効になりますが、インラインの最初のステップ (2) では有効にはなりません。両方のステップとも同じスレッド・ブロックの一部です。これらのステップは独立して実行されます。417 ページの『スレッド: ステップの並列実行』を参照してください。

.tset コマンドで設定された値は、ジョブ・レコードに書き込まれます。これらの値は、環境変数のデータベース・レコードは更新しません。後続のジョブは、この変更の影響を受けません。

## .unlock

```
.unlock
```

.unlock を使用すると、.lock コマンドを使用してロックされたジョブをシステムが解放します。アンロックされた実行は、「ジョブ」 > 「完了」と選択して表示されるページにリストされ、「ロック済み」ページのリストからは除去されます。



## 第 21 章 ジョブの処理

このトピックでは、管理コンソールでのジョブの実行、表示、および管理の方法について説明します。

### ジョブについて

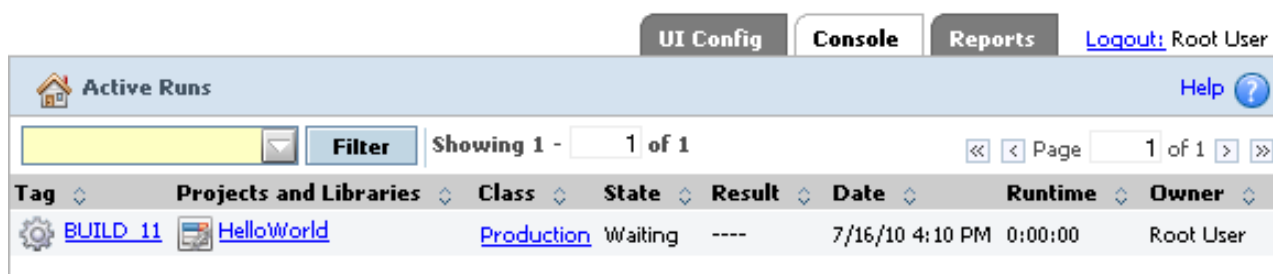
ジョブとは、実行中のプロジェクト、または実行が終了しているプロジェクトのことです。

ホーム・パネルと「ジョブ」パネルを使用して、ジョブをモニターできます。また、「ジョブ」パネルを使用して、ジョブの開始、ジョブの結果の表示、およびセマフォの管理を行うことができます。

### ホーム・パネルについて

ホーム・パネルは、最新のジョブおよびシステム・メッセージに関する情報を提供します。

ホーム・パネルにアクセスするには、左メニューで「ホーム」をクリックします。



ホーム・パネルを使用すると、最新または現行のジョブが表示されます。また、これを使用して通知およびシステム・メッセージを表示することもできます。メニュー項目を選択して、以下を表示します。

- ・「アクティブ実行」 - 現在実行中のジョブ
- ・「実行の完了」 - 完了したジョブ
- ・「システム・メッセージ」 - システム・メッセージ・ログ (ただし、メッセージを表示するには、「管理」 > 「メッセージ」を使用します。)

### 「ジョブ」パネルについて

実行中のジョブを処理したり、ジョブの結果を表示したりするには、「ジョブ」パネルを使用します。

「ジョブ」パネルにアクセスするには、左メニューで「ジョブ」をクリックします。

Jobs

UI Config

Console

Reports

Logout: Root User

Help

All

Completed

Running

Archived

Locked

Lock

Purge

Filter

Showing 1 - 5 of 5

Page 1 of 1

<input type="checkbox"/> Tag	<input type="checkbox"/> Projects and Libraries	<input type="checkbox"/> Class	<input type="checkbox"/> State	<input type="checkbox"/> Result	<input type="checkbox"/> Date	<input type="checkbox"/> Runtime	<input type="checkbox"/> Owner
<input type="checkbox"/> BUILD_1	<a href="#">Say_hi</a>	<a href="#">Production</a>	Completed	Passed	7/9/10 10:23 AM	0:00:03	Root User
<input type="checkbox"/> BUILD_4	<a href="#">HelloWorld</a>	<a href="#">Production</a>	Completed	Passed	7/8/10 4:43 PM	0:00:04	Root User
<input type="checkbox"/> BUILD_3	<a href="#">HelloWorld</a>	<a href="#">Production</a>	Completed	Passed	6/15/10 8:41 AM	0:00:06	Root User

## 「ジョブ」タブ

「ジョブ」タブでは、ジョブのリストを「すべて」、「完了」、「実行中」、「アーカイブ」、または「ロック済み」の状況ごとに表示できます。

タブをクリックして、タイプ別にジョブを表示します。各タブには、リスト内のジョブを管理するための各種のオプションがあります。

**すべて** すべての実行中のジョブが最初に表示され、その開始時刻によってソートされます。すべての完了ジョブがそれに続き、その完了時刻によってソートされます。チェック・ボックスはグレーになっています。いずれのリンクもクリックできます。特殊なアクションは使用できません。

**完了** すべての完了ジョブが表示され、その開始時刻によってソートされます。実行中、アーカイブ、およびロック済みの各ジョブは表示されません。完了ジョブはパージまたはロックできます。希望するジョブのチェック・ボックスを選択して、「パージ」または「ロック」をクリックします。完了ジョブがパージされると、完全に削除されていなければ「アーカイブ」に移動します。

**実行中** すべての実行中のジョブが表示され、その開始時刻によってソートされます。1 つ以上の実行中のジョブを取り消すことができます。希望するジョブのチェック・ボックスを選択して、「取り消し」をクリックします。

### アーカイブ

すべてのアーカイブされたジョブが表示され、その開始時刻によってソートされます。アーカイブされたジョブは、通常はジョブのクラスのクラス定義に指定された自動パージによって一部の情報が削除されたジョブです。アーカイブされたジョブをパージできます。希望するジョブのチェック・ボックスを選択して、「パージ」をクリックします。

### ロック済み

すべてのロック済みジョブが表示され、その開始時刻によってソートされます。1 つ以上のロック済みジョブをアンロックできます。希望するジョブのチェック・ボックスを選択して、「アンロック」をクリックします。

**注:** ロック済みジョブを直接削除することはできません。削除するには、まずアンロックしてから、「完了」タブに移動して、「パージ」を使用します。

## ジョブのリスト・フィールド

各ジョブの情報を表示するフィールド。

いずれのタブでも、完了時にジョブのリストが提供され、最新のジョブが最初に表示されます。ジョブごとに、以下の情報が列に表示されます。

**タグ** リンクとして表示されるビルド・タグ。クリックすると、ジョブ・ステップのログが表示されます。

**プロジェクトおよびライブラリー**

リンクとして表示されます。クリックすると、ジョブに使用されるプロジェクトが表示されます。

**クラス** リンクとして表示されます。クリックすると、ジョブに使用されるクラスのクラス定義が表示されます。

**状態** 「待機中」、「実行中」、「完了」、「アーカイブ」、または「ロック済み」のいずれかになります。

- 「待機中」とは、「実行中」キューが満杯であるか、ジョブがセマフォを待機しているために、ジョブが開始されていない状態のことです。「実行中」キューのサイズは、システム設定「**実行キュー・サイズ**」によって定義されます。
- 「実行中」キューにあるジョブは、通常は実行中です。しかし、「実行中」キューに同じプロジェクトに対する複数のジョブが含まれている場合、それらのジョブの 1 つ以上が現在実行中であるかどうかは、プロジェクトの「**実行上限数**」プロパティによって決まります。 363 ページの『「プロジェクト」プロパティの変更』を参照してください。

**結果** 「成功」、「失敗」、「失敗しましたが続行されました」、「警告」、「停止」、または「取り消し」のいずれかになります。

**日付** ジョブの開始前: ジョブが要求された日時です。

ジョブの開始後: ジョブが開始された日時です。

**実行時間**

実効中のジョブについては、現在の経過時間です。完了したジョブについては、経過時間の合計です。

**所有者** ジョブを開始したユーザーです。

## ジョブ・リストのフィルタリングとソート

任意のタブのジョブ・リストをフィルターに掛けてソートできます。

フィルターによって、指定したストリングと一致するものだけにジョブ・リストを制限します。ソートでは、任意のいずれかの列の値によってリストを順序付けします。

- フィルター: ジョブのリストをフィルターに掛けて、指定したストリングを含むジョブのみを表示できます。フィルターが一致するのは、いずれかの列にそのストリングが存在する場合です。
  - 新規フィルター: ボックスにストリングを入力して、「**フィルター**」をクリックします。
  - 既存のフィルター: ボックスの隣にある矢印をクリックして、使用するフィルター・ストリングを選択します。

- ソート: 単一列の値によってジョブ・リストをソートできます。列名の隣にある矢印をクリックします。複数回クリックすると、ソートは以下のソート順で繰り返し表示されます。
  1. 昇順 (列のヘッダーで上矢印が強調表示されます)
  2. 降順 (列のヘッダーで下矢印が強調表示されます)
  3. フィルターしない (no filter)

## ジョブの実行および結果の表示

ジョブの実行、スケジュール、取り消し、および再始動を行い、ジョブの結果を表示することができます。

### ジョブの開始

ジョブを開始するには、いくつかの方法があります。

#### 始める前に

ジョブを実行する前に、プロジェクトで環境変数とセレクターが定義されていなければなりません。

#### 手順

- 「ジョブ」パネルから: 「ジョブ」 > 「開始」 > 「プロジェクトの開始」を選択して、プロジェクト名をクリックします。「プロジェクトの開始」をクリックすると、「プロジェクトの開始」ページが表示されます。

Job DetailsJob Steps

Project Parameters

Snapshot: Base Snapshot

Selector: My selector

Class: Production

Tag Format: BUILD\_\$B

Tag Example: BUILD\_9

Project Environment


Job will start with the default environment.

Save Environment ☐

Project Tags

☐ Editable Tags


B 9

- 「プロジェクト」パネルから: 「クイック・スタート」アイコン  をクリックします。

以下の条件が検査されます。検査にパスすると、即時にプロジェクトが開始されます。

- プロジェクトに 1 つ以上のステップが含まれている。
- 「プロジェクト動作」変数プロパティーで、プロジェクトの環境内にあるどの変数も「変更が必要」に設定されていない。



検査に不合格だった場合は、「クイック・スタート」アイコンは無効になります。

この方法でプロジェクトを実行すると、セクター、クラス、タグ、および環境変数のデフォルト値が使用されます。

「即時開始を使用可能に設定」システム設定が「はい」に設定されている場合は、「プロジェクト」ページですべてのプロジェクトが検査されます。これにより、それらのプロジェクトがクイック・スタートの対象となるかどうかは判別されます。多数のプロジェクトが定義されている場合は、「プロジェクト」ページでのプロジェクト・リストの表示に長時間を要する場合があります。プロジェク

トが開始可能であることは、別のアイコン  によって示されます。バージョン 7.1.1.1 までは、これがデフォルトの動作でした。

- 「ステップ」パネルから：プロジェクトのステップを表示している間に「**プロジェクトの開始**」をクリックします。この方法を使用すると、プロジェクトの「プロジェクトの開始」ページが表示されます。このページで、プロジェクト・パラメーターおよび環境変数の値を変更したり、実行対象から除外するステップを選択したりできます。
  - プロジェクト・パラメーターの新しい値を選択します。
  - プロジェクトのタグ変数値を編集します。
  - プロジェクトの環境変数値を編集します。これらの変数の新しいデフォルトとして変更内容を保存したい場合は、「**環境変数の保存**」チェック・ボックスを選択します。
  - 「**ジョブ・ステップ**」タブを選択して、プロジェクト・ステップのリストを表示します。列ヘッダーにあるチェック・ボックスをクリックすれば、組み込むステップを選択できます。ステップを除外する場合は、そのステップのチェック・ボックスをクリアします。
    1. ステップが条件タイプ (Java エンジンでの IF 条件と ELSE 条件) の場合は、両方の条件が表示され、どちらの条件 (IF または ELSE) が実行されるかを選択できます。上記イメージを参照してください。
    2. IF 条件のみを選択した場合は、条件評価なしで IF が実行されます。
    3. ELSE 条件のみを選択した場合は、条件評価なしで ELSE が実行されます。
    4. IF 条件と ELSE 条件の両方を選択した場合は、条件評価を使用してコマンドを実行します。
    5. IF 条件と ELSE 条件の両方が選択解除されている場合は、選択したステップの条件を実行します。(IF 条件と ELSE 条件の両方が選択解除されている場合は、一方の条件のみ (IF または ELSE) が選択可能です。
- 選択が完了したら、「**実行**」をクリックしてプロジェクトを開始します。

## タスクの結果

プロジェクトの実行中、プロジェクトの状況を確認するには、「**ジョブ**」 > 「**実行中**」ページを参照してください。

ジョブの結果を表示するには、「ジョブ」 > 「完了」を選択して、完了したジョブを表示します。ジョブの結果を表示するオプションを使用するには、「タグ名」をクリックしてください。

## ジョブ結果の表示

### このタスクについて

すべてのジョブのタブでジョブ結果を表示できます。

#### 手順

1. 「ジョブ」をクリックします。「完了」タブが選択されます。
2. 希望するタブをクリックします。
3. 表示するジョブのタグをクリックします。以下のジョブ結果が表示されます。
  - ステップのリストがメインパネルに表示されます。各ステップには、情報の列があります。
    - ステップ: ステップ番号
    - ステップ名: ステップ名。ステップ名の前にあるグラフィックは、ステップ・タイプを示します。以下のステップ・タイプが表示されます。
      - a. ステップ
      - b. スレッド化
      - c. ブロードキャスト
      - d. 結合ステップ名をクリックすると、ステップ・ログが表示されます。
    - 結果: ステップ結果。以下のステップ結果が表示されます。
      - a. 成功
      - b. 失敗
      - c. 取り消し
    - ログ・バージョン: ステップが再始動された場合、そのステップのログ・バージョンがドロップダウン・リストに表示されます。ログ・バージョン 1 は、初回実行のステップ・ログを示しています。再始動後は、ログ・バージョン番号が 1 ずつ増えます。
    - サーバー (セレクトター) (Server (Selector)): サーバーがいつ稼働するかを示すステップの名前。また、サーバーを選択したセレクトターも表示されます。
    - ランタイム: ステップの経過時間 (時間、分、または秒)。
    - チェーン: ステップが別のジョブを開始した場合、そのジョブへのリンクが表示されます。
  - 左側メニューの下部にメニューが表示され、以下の項目が示されます。
    - 結果: メイン・パネルにステップのリストを表示します。
    - 部品構成表: メインパネルに部品構成表 (BOM) を表示します。BOM には、以下の項目を表示するリンクが含まれています。
      - a. ジョブ・ステップ
      - b. ステップ・マニフェスト

- c. チェックポイント (チェックポイントが使用された場合)
- メモ: このジョブに入力されたメモを表示します。
- ステップ・ログ (この項目が開いていて、ステップ名が表示されている。): ステップのリストを表示します。ステップをクリックすると、ステップ・ログが表示されます。
- チェーン: チェーンとして起動されたすべてのジョブのジョブ番号を表示します。

注: `.run` と `.runwait` の一方または両方を使用して起動されたジョブの場合、このアイコンは表示されません。

4. ステップ・リストからステップを 1 つ選択し、ステップのログ・バージョンを選んで、ステップ名をクリックします。当初は、すべてのカテゴリが選択されています。

実行中ジョブのステップを表示しようとしていて、そのステップが完了していない場合は、ログの一部が表示され、一部のカテゴリは表示されない可能性があります。実行中ステップのログ表示を更新するには、ビルド・タグ (上部に、「**ジョブ**」>>「**タグ**」として表示されます) をクリックして、そのステップをクリックします。ステップ・ログを表示するには、「**すべてを表示**」をクリックしてすべてのステップを表示します。表示されるステップの数を減らすには、「***n* のうち 1 から *m* までを表示 (Showing 1 -*m* of *n*)**」コントロールに *m* の値を入力し、**Enter** を押してください。

5. ステップ・ログをフィルターに掛ける場合、カテゴリにチェック・マークを付けるか、またはクリアして、「**最新表示**」をクリックします。

```

1 7/8/10 4:43 PM STEP      Step using selector 'My selector'.
2 7/8/10 4:43 PM MANIFEST  BF_LAST_UPDATE=1278618783
3 7/8/10 4:43 PM MANIFEST  BF_NAME=rbf-14
4 7/8/10 4:43 PM MANIFEST  BF_LOADRATIO=0.3333333333333333
5 7/8/10 4:43 PM MANIFEST  BF_JOBS=1
6 7/8/10 4:43 PM MANIFEST  BF_AGENT_VERSION=version
7 7/8/10 4:43 PM MANIFEST  BF_LAST_REFRESH=1278618783

```

## ジョブ状況に関する RSS データ・フィード

RSS データ・フィードを使用して、個々のジョブの状況を追跡およびフィルター操作できます。Build Forge の、ジョブに関する RSS データ・フィードは、Build Forge 管理コンソールのサーバー状況に表示される情報と同じ情報を表示します。

ジョブ状況に関する RSS データ・フィードに加入するには、次のようにしてください。

1. Build Forge 管理コンソールで、「**ジョブ**」を選択します。

Web ブラウザーは RSS フィードを検出し、ブラウザーのアドレス・バーに「RSS」アイコンを表示します。

2. RSS 統合機能ツールで、Build Forge RSS データ・フィードをロードします。

例えば、URL をコピーして RSS データ・フィードのリストに追加します。あるいは、「RSS」アイコンをドラッグ・アンド・ドロップして RSS データ・フィードのリストに URL を追加します。

3. RSS データ・フィードに加入して URL を保存し、更新が行われたときに通知を受けるようにします。

注:

- URL のロードおよび RSS データ・フィードへの加入の詳細については、RSS 統合機能ツールの資料を参照してください。
- RSS データ・フィードを使用して、Build Forge のジョブ状況、システム・メッセージ、またはサーバー状況を英語以外の言語で表示するには、RSS 統合機能ツールが UTF-8 マルチバイト文字エンコードをサポートしている必要があります。
- RSS フィードは、タイム・スタンプを使用してジョブ・アクティビティを報告します。タイム・スタンプは、ご使用の Build Forge ユーザー・アカウントに設定されているタイムゾーンに基づいています。ご使用のブラウザーが別のタイムゾーンに設定されている場合、正確なタイム・スタンプを受け取るには、ご使用の Build Forge アカウントと同じタイムゾーンにブラウザーを設定します。

## 失敗したジョブの再始動

ジョブが失敗した場合、そのジョブを再始動できます。再始動では、同じタグの下で新規実行を開始します。失敗した時点から続行されます。

ジョブを再始動するには、「ジョブ」 > 「完了」タブにあるビルドのリスト内の「ジョブ」タグをクリックします。システムは、そのビルドに関する情報を表示し、パネルの上部の周辺に「ジョブの再始動」ボタンを組み込みます。

1. 「ジョブの再始動」をクリックします。

「再始動」ページを表示します。

2. オプションを選択します。ステップ内でコマンドに対して行われた更新をプロジェクト・レコードからシステムで受信する場合は、「同期コマンド」プロパティを設定します。このプロパティを設定しない場合、コマンドは、ジョブが開始されたときと同じように入力されます。
3. 「再始動」ページで「再始動」ボタンをクリックします。

ジョブ再始動では、新規ジョブと比較して以下のような明らかな違いがあります。

- 失敗した実行と同じタグ番号を使用し、「完了」リスト内の失敗した実行を置き換えます。
- デフォルトでは、再始動ジョブは、失敗したステップから開始され、前の実行ステップをいっさい繰り返しません。ただし、実行ステップは、ジョブの再始動時に選択できます。
- デフォルトでは、システムは、前の実行で指定された環境変数値と同じ環境変数値を指定します。ただし、環境変数値は、ジョブの再始動前に変更できます。
- システムは、再始動したジョブで実行するステップのみに基づいて、ジョブの成否を評価します。前の実行における失敗は、再始動したジョブの状況には影響を与えません。

アダプター・リンク・ステップを伴うジョブ再始動については、561 ページの『アダプター・リンクについて』を参照してください。

## ブロードキャスト・ステップの再始動

ブロードキャスト・ステップとしてマークされたステップは、実行時に、実行先のサーバーごとにステップのセットとして表示されます。再始動ページと開始ページでは、このステップは単一のステップとして表示されます。

ブロードキャスト・ステップが失敗すると、すべてのステップが失敗ステップとして表示される可能性があります。再始動後に以下の操作を実行してください。

1. 単一のブロードキャスト・ステップを選択します。
2. 「同期コマンド」プロパティを設定します。
3. 他の望ましいオプションを設定します。
4. 「再始動」ページで「再始動」をクリックします。

ブロードキャスト・ステップが、使用可能なブロードキャスト・サーバーで実行し直されます。

## While ループの再始動

While ループ・タイプのステップの場合、反復時のステップは、BF\_ITERATION システム変数の値に基づいて失敗した場所から再始動されます。ジョブ・フローの例を示します。

1. ジョブが While ループ・タイプのステップに入ります。
2. 条件が true と評価されます。
3. BF\_ITERATION が 1 に設定されます。
4. コマンドおよびインラインが正常に実行されます。
5. ステップがループします。
6. 条件が true と評価されます。
7. BF\_ITERATION が 2 に設定されます。
8. コマンドまたはインラインの実行中にジョブが停止します。

上記のジョブを再始動すると、反復 2 で再始動されます。ステップのコマンドおよびインラインの実行を試行します。

## BOM (Bill of Materials) の使用

システムは、各ジョブの後に Bill of Materials (BOM) を生成します。BOM には、ジョブのステップについての情報と、その結果得られたファイルへの変更についての情報が含まれます。BOM は、新規のジョブの内容についての理解を助けるため、例えば、品質保証部門など、ジョブの利用者に提供することができます。BOM は、ビルドおよびリリース・プロセスについての監査解決策として、サービスを提供できます。BOM によって、ジョブの内容についての完全な資料が得られます。これには、結果、メモ、環境変数、ファイルのリスト、およびコード変更を組み込むことができます。これを使用して、企業全体でビルドの状態を比較および要約できます。

システムはジョブごとに自動的に BOM を生成しますが、各ステップでドット・コマンドを使用すると、ビルドの前後のファイルの状態についての追加情報をシステムに保管することができます。

## 部品構成表 (BOM) の表示

完了したビルドを表示すると (「ジョブ」 > 「完了」)、デフォルトで「ステップ」タブが表示されます。「BOM」タブをクリックすると、部品構成表が表示されます。

任意のカテゴリの横にある「+」をクリックして、カテゴリを拡張します。表示される実際のカテゴリは、プロジェクトおよびシステムの構成方法によって異なります。

- 「プロジェクト・ステップ (Project Steps)」カテゴリは、すべてのジョブに表示され、そのジョブで実行されたステップに関する情報が提供されます。
- 「ソースの変更 (Source Changes)」カテゴリは、システムにソース・コード・アダプターが含まれており、そのアダプターへのリンクがプロジェクトにある場合に限り、表示されます。詳しくは、561 ページの『アダプターおよびジョブ結果』を参照してください。「ソースの変更」カテゴリのフォーマットおよび名前も、アダプターを構成する際に変更できます。
- 「ベースライン」および「チェックポイント」セクションは、.scan コマンドがプロジェクトに含まれている場合に限り、表示されます。

## .scan コマンドでのベースラインとチェックポイントの追加

.scan コマンドを使用して、BOM に情報を追加することができます。 .scan コマンドが実行されると、システムは、ステップの作業ディレクトリーにファイルの状況に関する情報を保管します。このセクションには、使用方法の例を示します。458 ページの『.scan』の参照情報も参照してください。

このコマンドには 2 つの形式があります。

### .scan baseline

それぞれに MD5 値を使用して、ステップの作業ディレクトリー・ツリーにすべてのファイルのリストを保管します。システムは、そのジョブの BOM にリストを表示します。このコマンドは、いくつかのセットアップ・ステップを実行し、ファイルの適切なセットをチェックアウトした後で実行される場合があります。プロジェクトに複数の .baseline コマンドを持つことができますが、各コマンドが実行されると、その .baseline コマンドは、リストをステップの作業ディレクトリーの状態にリセットします。

### .scan checkpoint

プロジェクト内での最後の .scan baseline または .scan checkpoint 以降の、すべての新規ファイル、変更済みファイル、および削除済みファイルのリストを、各ファイルに MD5 値を使用して保管します。 .scan baseline コマンドと同様、システムは BOM にリストを表示します。プロジェクトで最初の .scan checkpoint コマンドの前に、.scan baseline コマンドを実行する必要があります。 .scan baseline コマンドより前にある .scan checkpoint コマンドは無視されます。

以下の例に、.scan baseline コマンドと .scan checkpoint コマンドがどのように連動するか示します。



番号	ステップ	ステップ後のファイル	BOM データ
1	初期ファイルのチェックアウト	config.c execute.c	
2	.scan baseline	config.c execute.c	ベースライン: config.c execute.c
3	データ・ファイルの追加	config.c execute.c data.txt	
4	.scan checkpoint	config.c execute.c data.txt	チェックポイント 1: 追加された data.txt
5	データ・ファイルの追加	config.c execute.c data.txt data2.txt data3.txt	
6	data.txt の削除	config.c execute.c data2.txt data3.txt	
7	.scan checkpoint	config.c execute.c data2.txt data3.txt	チェックポイント 2: 追加された data2.txt、 data3.txt  削除された data.txt

## XML ファイルとしての BOM のエクスポート

このトピックでは、bf bomexport コマンドの構文、使用法、およびオプションについて説明します。

Build Forge コマンドは、Windows の場合は Build Forge インストール・ディレクトリー、UNIX および Linux の場合は `<bfinstall>/Platform` ディレクトリーにあります。

### 説明

bf bomexport コマンドにより、ジョブの部品構成表 (BOM) は XML ファイルにエクスポートされます。BOM 情報が収集されると、bf bomexport により BOM は指定した場所およびファイル名で保管されます。

XML ファイルに保管したい BOM を指定するには、プロジェクトおよびジョブのビルドを指定する必要があります。

bf bomexport コマンドは、管理コンソールの Build Forge インストール・ディレクトリーや、UNIX または Linux 上の `/Platform` ディレクトリーから実行する必要があります。



## 構文

```
bfexport [-f filename] [-p projectID | -P projectName]
        [-b buildID | -t buildTag] [-L] [-H]
```

## オプション

オプション	説明
-f filename	オプションのパスまたはファイル名、あるいはその両方。ジョブの BOM は XML 形式で保管されます。ファイル名を指定しない場合、BOM は標準出力 (stdout) に書き込まれます。パス名を指定しない場合は、現行作業ディレクトリーが使用されます。
-p projectID	ジョブのプロジェクト ID。( -l オプションを指定して bfexport コマンドを実行すると、プロジェクト ID がリストされます。)
-P projectName	プロジェクトの名前。
-b buildID	ビルド ID。
-t buildTag	ビルド・タグ名。
-L	ステップ・ログを含めます。
-H	ヘルプ・メッセージ。

---

## ジョブのスケジューリング

今後のある一時刻に実行したり、一定間隔で定期的に繰り返し実行したりするようにプロジェクトをスケジュールに入れるには、「**スケジュール**」パネルを使用します。例えば、あるプロジェクトを毎時または毎日実行するようにセットアップすることができます。

ジョブ・スケジュールを表示するには、「**スケジュール**」をクリックします。今月のカレンダーが表示されます。その下には、スケジュール項目の作成および変更使用するパネルがあります。

カレンダーの各日付には、その日のスケジュールに入っているプロジェクトの数が表示されます。日付にマウス・カーソルを重ねると、その日のスケジュールに入っているジョブの名前およびスケジュール・パラメーターが表示されます。

複数のプロジェクトがスケジュールに入っている場合は、プロジェクトを基準にしてカレンダーをフィルタリング可能なドロップダウン・ボックスおよび「**フィルター**」ボタンがシステムによって表示されます。

UI Config Console Reports Logout: Root User

Schedules Add Scheduled Run

Schedule List Calendar

Filter

Zero pages to display

July 2010						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

(New Entry) Save Schedule Copy Schedule Resync Environment Delete Schedule

Schedule Details Environment

Description: Access: Build Engineer Owner: Root User

Project: - Class Purge Schedule - Class: Production Mode: Active

Environment: - Project Default - Auto-Sync Environment: No

Selector: - Project Default - Minutes: 0 Hours: 0 Dates: \* Months: \* Days: \*

プロジェクトをスケジューラーに入れるには、以下のようになります。

1. 左メニューで、「スケジュール」をクリックします。
2. 「スケジュール実行の追加」をクリックします。
3. 「説明」にスケジュール項目の説明を入力します。
4. 「プロジェクト」リストからプロジェクトを選択します。
5. プロジェクトの現在のスナップショットが、「プロジェクト」の下にあるフィールドに表示されます。スケジュールに入れられたジョブのために別のスナップショットを実行させるには、スナップショットをリストから選択します。
6. 「モード」で、「アクティブ」(デフォルト)、「非アクティブ」、または「1度」を選択します。「1度」に設定した場合、そのジョブは、次に時刻設定が一致したときに一度だけ実行されます。「アクティブ」に設定した場合、そのジョブは、時間設定が一致するたびに実行されます。
7. オプション: 以下のプロジェクト・プロパティをオーバーライドすることができます。ここで選択した設定は、スケジュールに入れられたジョブのこのインスタンスに対してのみ有効になります。
  - **アクセス:** これを指定すると、スケジュールに入れられたジョブが指定したアクセス・プロパティを使用ようになります。
  - **所有者:** これを指定すると、スケジュールに入れられたジョブの実行は、指定の所有者が手動で起動したものとして実行されます。
  - **クラス:** これを指定すると、スケジュールに入れられたジョブが指定したクラスを使用ようになります。
  - **環境変数:** スケジュールに入れられたこのジョブに対する環境変数のコピーが、このフィールドに作成されます (値が明示的に指定されていても、「プロジェクト・デフォルト」のままであっても同じです)。環境変数を指定すると、変数の開始値も設定できます。

**重要:** コピーが作成されると、オリジナルの環境変数とその変数に対する変更内容は、スケジュールに入れられたジョブ用に作成された環境変数のコピーで

自動的に更新されることはありません。オリジナルの環境変数に加えられた変更内容が反映されるように、コピーを手動または自動で更新することができます。手動更新する場合には、「環境変数の再同期」をクリックしてください。自動更新する場合には、「環境変数の自動同期 (Auto-Sync Environment)」を「はい」に設定してください。自動更新を設定すると、スケジュールされたジョブが実行されるたびに環境変数が更新されます。ただし、「環境変数」タブは、「環境変数の自動同期 (Auto-Sync Environment)」が「いいえ」に設定されたスケジュールでのみ使用できます。

- **セレクトター**: これを指定すると、ジョブが指定のセレクトターを使用するようになります。
8. 時刻設定を指定します。「分」、「時」、「日」、「月」、および「曜日」に値を入力します。使用する値については、『スケジュール・パラメーター』を参照してください。
  9. 「スケジュールの保存」をクリックします。

「次の実行」列に「計算」と表示された後、そのジョブが実行される次のスケジュール時刻が表示されます。

スケジュールが作成されると、カレンダーにスケジュール名が表示されます。その隣にあるアイコンにより、モードが表示されます。モードは、このアイコンをクリックすることによって変更できます。クリックするたびにモードが変更されます。

- 緑の円: アクティブ
- 青の円: 1 度
- 赤の円: 非アクティブ

スケジュールに入れられたジョブの実行が試行されると、システムは、ジョブ・キューおよび「ハード実行上限数」システム設定、さらに場合によってはプロジェクトの「実行上限数」プロパティを检查します。あるプロジェクトの 1 つ以上のジョブが、スケジュールのアクティブ化時点で既に実行中である場合、システムの動作は、次の表で説明する「ハード実行上限数」システム設定によって決まります。

「ハード実行上限数」値	システムは「実行上限数」値を检查するか?	システムはジョブを起動するか?
はい	はい	実行中のジョブの数が「実行上限数」値より少ない場合は、はい。  実行中のジョブの数が「実行上限数」値に等しい場合は、いいえ。
いいえ	いいえ	はい

注: 前の実行が完了していない場合に実行をスキップするためには、「実行上限数」値を 1 に設定し、「ハード実行上限数」を「はい」に設定してください。

## スケジュール・パラメーター

このセクションでは、ジョブの実行時を指定するために使用できるパラメーターについて説明します。

一連のフィールドを使用して、ジョブを実行する時間を指定します。

フィールド	説明	範囲
分	分数。	0-59
時	1 日の時間。	0-23
日 (日付)	月の中の日付。	1-31
月	年の中の月。	1-12
曜日 (平日)	曜日 (日曜日 = 0)。	0-6

フィールドに入力する数値は、次のように表すことができます。

- 範囲内の有効なすべての値を示すには、アスタリスク (\*) を使用します。アスタリスクの後にスラッシュ (/) とステップ値を続けることができます。例えば、「時」フィールドの \*/2 の値は、ジョブを 2 時間ごとに実行します。

注: アスタリスクは、特に「日」および「曜日」の各フィールドで解釈されます。UNIX/Linux の cron 機能での意味と一致しています。あるフィールドがリテラル値で、別のフィールドがアスタリスクである場合、リテラル値の頻度のみが使用されます。例えば、「月」が \*、「日」が \*、「曜日」が 1 の場合、毎週月曜日に実行されます。下記の日/曜日の例も参照してください。

- 数値範囲は、ハイフンで区切って使用します。例えば、「時」フィールドの 8-11 は、8 時、9 時、10 時、および 11 時を指定します。範囲の後にスラッシュ (/) とステップ値を続けることができます。例えば、「時」フィールドの 0-23/2 は、ジョブを 1 時間おきに実行します。
- コンマで区切った数値 (または範囲) セットのコンマ区切りリストを使用します。例えば、1、2、3-5、8 などです。

スケジュールは、これらのフィールドに指定された値から構成されます。例:

値 希望するスケジュール	分	時	日	月	曜日
毎日午後 5 時にジョブを実行する。	0	17	*	*	*
毎週月曜日の午後 4 時 30 分にジョブを実行する。	30	16	*	*	1

値 希望するスケ ジュール	分	時	日	月	曜日
平日の毎日 30 分ごとに ジョブを実行 する (週末は 飛ばす)。	*/30	*	*	*	1-5
1 日おきに、 午前 12 時 30 分にジョ ブを実行す る。	30	0	*/2	*	*

「曜日」および「日」の各フィールドには、特殊な方法でアスタリスクを使用しま  
す。

- 一方がアスタリスク値であって、もう一方がリテラル値である場合、ジョブはリ  
テラル設定値のフィールドに従って実行されます。
- いずれも非アスタリスク値である場合、ジョブはいずれかの 条件を満たしたとき  
に実行されます。
- いずれもアスタリスク値である場合、ジョブは毎日実行されます。

日/曜日の例:

値 希望するスケジュール	分	時	日	月	曜日
各月の最初の日の午前 1 時 01 分にジョブを実行する。 「日」はリテラル値を使用 し、「曜日」はアスタリス クを使用する。	1	1	1	*	*
その月の毎月曜日の午前 1 時 01 分にジョブを実行す る。「日」はアスタリスク を使用し、「曜日」はリテ ラル値を使用する。	1	1	*	*	1

値					
希望するスケジュール	分	時	日	月	曜日
月の毎月曜日の午前 1 時 01 分と、月の最初の日の午前 1 時 01 分 (その日が月曜日かどうかにかかわらず) にジョブを実行する。 「日」と「曜日」はいずれもリテラル値を使用する。	1	1	1	*	1
毎日午前 1 時 01 分にジョブを実行する。「日」と「曜日」はいずれもアスタリスク値を使用する。	1	1	*	*	*

## ジョブのクラスのページのスケジューリング

ジョブのクラスのスケジュールを作成することにより、システムに古いジョブをいつページさせるかを制御できます。これらのスケジュールは、プロジェクトを起動するスケジュールを作成する場合と同じように作成できます。ただし、プロジェクトの代わりに「クラス・ページ・スケジュール」オプションを選択してください。これを行うと、システムは、スケジュールで選択された時刻に、選択されたクラスをページするジョブがないかどうか調べます。システムは、条件に一致するジョブごとにページ・ジョブを作成し、そのページ・ジョブを待機キューに入れます。

デフォルトでは、システムは、「ページ確認時間」システム設定値で設定された間隔 (デフォルトでは 15 分ごと) で、ページする必要のあるジョブがないかどうかを (自動削除の規則を定義するクラス・プロパティに基づいて) 調べます。この動作により、システム・リソースと通常のジョブとで、ページが競合する可能性があります。

ジョブのクラスのスケジュールを作成する場合、システムは、スケジュールが活動化されたときにのみ、ページすべきジョブの有無を確認します。特定のクラスに関してスケジュールが存在しない場合、システムは、そのクラスのジョブに関するデフォルトの動作を適用します。すべてのページが特定の時刻に行われるように限定したい場合には、クラスごとに少なくとも 1 つのスケジュールを作成する必要があります。

ページのスケジュールを定義するには、以下のようにします。

1. 通常どおりにスケジュールを作成してください。ただし、「プロジェクト」フィールドで「クラス・ページ・スケジュール」を選択してください。
2. 「クラス」フィールドで、クラスを選択します。

## ジョブの管理

ジョブをロック、アーカイブ、および削除することができます。

## ジョブのロック

プロジェクトが自動的に削除されないようにするために、プロジェクトをロックすることができます。

ジョブをロックするには、「ジョブ」パネル内の「完了」タブをクリックし、ビルドを選択して、「ロック」をクリックしてください。

ジョブをアンロックするには、「ジョブ」パネル内の「ロック」タブをクリックし、ロックされているビルドを選択して、「アンロック」をクリックしてください。

システムは、(実行のためにクラス・ページ・プロパティーが削除を呼び出す場合のように) ロックされているジョブを自動的にパージしません。ロック済みジョブを手動で削除することもできます。

ロック状態になっているジョブを表示するには、「ロック」タブをクリックしてください。そのようなプロジェクトは「完了」ページには表示されません。

## ジョブの削除

以下のトピックでは、ジョブを削除するためのいくつかの方法について説明します。

### 完了タブからのジョブの削除

ジョブは、「ジョブ」パネルの「完了」タブから手動で削除できます。

このリストから、1 つ以上のジョブを手動で削除できます。これを行うと、ページ・ジョブが各実行のクラスに基づき起動します。このプロセスは、自動削除がクラス・プロパティーによってトリガーされた場合と同じです。システムはジョブをアーカイブし (完全に削除されていない場合)、そのクラスのジョブに指定されているデータを削除します。382 ページの『クラス』を参照してください。

ジョブを削除するには、次の手順を実行してください。

1. 「ジョブ」 > 「完了」 ページを選択して、完了した実行リストを表示します。
2. テーブルの右端にあるチェック・ボックスを 1 つ以上クリックして、削除するビルドを選択します。
3. リスト下部にあるリスト・ボックスから、「ページ」オプションを選択します。
4. 「実行 (Go)」ボタンをクリックします。

出力ファイルは削除するが、コンソール・データは保存しておくようクラスを設定した場合は、「完了」リストから実行を削除すると、出力ファイルが削除され、ジョブのエントリーが「完了」リストから「アーカイブ」リストに移動します。

### アーカイブ・リストからのジョブの完全な削除

「ジョブ」 > 「アーカイブ」を選択して、アーカイブ・リストを表示します。このリストには、ファイルが削除されたジョブについてのデータが表示されます。このリストからジョブを削除できます。「ジョブ」 > 「完了」のリストから行うのと同



様に削除できます。アーカイブ・リストからジョブを削除すると、ジョブのすべてのトレースがデータベースから除去され、アプリケーションによってレポートされる統計からも除去されます。

## ジョブの自動削除

システムは、ジョブのクラス・プロパティがその削除を決定した場合、ジョブを自動的に削除します。このフィーチャーを使用して、データが集積するのを防いだり、ジョブのグループを削除したりすることができます。

クラスのスケジュールを作成する場合、システムは、スケジュールが活動化されたときにジョブをパージするのを確認するだけです。 483 ページの『ジョブのクラスのパージのスケジュールリング』を参照してください。

必要な場合にジョブが削除されたことを確認するには、ジョブを起動またはスケジュールする前に以下の設定を確認します。

1. 1 つ以上のクラスの削除プロパティを設定して、一定の日数の経過後、または一定のジョブ数の累積後、あるいはその両方の場合に、システムがジョブを削除できるようにします。
2. 処理するプロジェクトの「**クラス**」プロパティを該当するクラスに設定します。

いくつかのジョブを生成してから、それらを削除する場合は、関連するクラスの削除プロパティを一時的に変更することができます。あるいは、「完了」タブで複数のビルドを選択し、それらを削除することができます (各ジョブの隣にあるチェック・ボックスをクリックして選択し、「**パージ (Purge)**」ボタンをクリックします。)

例えば、前日に非常に多くの実動ジョブを生成した場合は、次のプロセスを使用してそれらを削除します。

1. 実動クラスの現在の設定をメモします。
2. 実動クラスの「日数」プロパティを 1 に変更してから、「**クラスの保存 (Save Class)**」ボタンをクリックします。15 分後に、システムは 1 日より長く経過したジョブの削除を開始します。
3. ジョブが削除されると、実動クラスのプロパティが元の設定に戻されます。

## ジョブ用の作業ディレクトリー

システムはジョブごとに作業ディレクトリーを構成するため、それぞれの実行に、ラベルが付けられた独立の作業領域が割り当てられます。作業ディレクトリー名は、サーバー・パス、プロジェクト名、およびタグ用に提供された値を使用して付けられます。

システムは、コマンドを実行するときに、該当ステップに指定されたディレクトリー内でそのコマンドを開始します。デフォルトでは、そのディレクトリーはジョブの作業ディレクトリーですが、サーバーのパスプロパティに関連するその他のディレクトリーを指定することもできます。このセクションのトピックでは、パスおよびディレクトリーを作成する方法を説明します。

注: システムは、プロジェクトを実行するときに、プロジェクト・ディレクトリー (まだ存在しない場合) およびジョブ・ディレクトリーを構成します。サーバー・ディレクトリー (サーバーの「パス」プロパティーで指定されます) とステップ・ディレクトリー (ステップの「ディレクトリー (Dir)」プロパティーで指定されます) は構成されません。

## ジョブ用の作業ディレクトリー名

次の例では、単一サーバーで発生するジョブで、システムがいくつかの値を使用してジョブ・ディレクトリーを作成する方法を示します。

システム値	作成されるディレクトリー
サーバーの「パス」フィールド: C:¥BuildForge  プロジェクト名: My Project  タグ: Job 5	C:¥BuildForge¥My_Project¥Job_5¥  システムによって作成されるのは、パスの太字部分のみです。プロジェクトを実行する前にお客様自身でサーバー・ディレクトリーを作成する必要があります。そうでないと、プロジェクトは失敗します。

注: プロジェクト・ディレクトリーを作成すると、システム設定の「無効な相対ディレクトリー文字」に指定されている文字が、下線文字に変更されます。デフォルトでは、この設定にスペースおよびバックチック文字が含まれており、「My Project」という名前のプロジェクトは、「My\_Project」という名前のプロジェクト・ディレクトリーを受け取ります。

ジョブが複数サーバーで実行される場合、システムはサーバーごとにジョブ・ディレクトリーを作成します。プロジェクトの各ステップで異なるサーバーを指定できるので、多数のディレクトリーが作成される可能性があります。次の例は、2 つのサーバーを使用するプロジェクトを示しています。

システム値	作成されるディレクトリー
プロジェクト・サーバー: ServerA。「パス」の値は C:¥BuildForge。  プロジェクトの 3 番目のステップには、ServerB が指定されています。「パス」の値は C:¥deployments です。(他のすべてのステップでは、デフォルト (プロジェクト) サーバーである ServerA を使用します。)  プロジェクト名: My Project  タグ: Job_6	ServerA 上:  C:¥BuildForge¥My_Project¥Job_6¥  ServerB 上:  C:¥deployments¥My_Project¥Job_6¥  システムによって作成されるのは、パスの太字部分のみです。

上記の例では、C:¥deployments¥My\_Project¥Job\_6 ディレクトリーを使用する 3 番目のステップを除いて、C:¥BuildForge¥My\_Project¥Job\_6 ディレクトリーにデフォルトで作成されるステップから、出力ファイルがあることが予想されます。

## ステップ用のディレクトリー・パスの作成

ステップを実行する場合、システムは、ジョブ用に作成したディレクトリーからステップを開始できます。また、ステップの「パス」オプションが「絶対」に設定されている場合は、プロジェクトとタグの各ディレクトリーを無視できます。

- ステップの「パス」が「絶対」に設定されている場合、システムは、**サーバー・パス**とステップの「ディレクトリー」フィールドを一緒に追加することによって、ステップのパスを作成します。「ディレクトリー」フィールドの値は、サーバーの作業ディレクトリーを基準とする相対パスです。

ステップの値	コマンドの結果生じるパス
サーバーの「パス」フィールド: C:/BuildForge  ステップの「ディレクトリー」フィールド: /bin  ステップの「パス」の設定: 絶対	C:¥BuildForge¥bin  このフォームを使用して、サーバー・ディレクトリーにあるディレクトリーにアクセスします。
サーバーの「パス」フィールド: C:/BuildForge  ステップの「ディレクトリー」フィールド: / (デフォルト値)  ステップの「パス」の設定: 絶対	C:¥BuildForge
サーバーの「パス」フィールド: C:/BuildForge  ステップの「ディレクトリー」フィールド: C:/temp  ステップの「パス」の設定: 絶対	C:¥BuildForge¥C:¥temp  (この例を使用すると、エラーが発生し、ステップが失敗します。)

**注:** パスの値を、円記号またはスラッシュを使用して入力できます。システムは、円記号を含むパスを保管し、必要に応じて Windows® コンピューターではそれらをスラッシュに変更します。

**重要:** コマンドのパスをルート・ディレクトリーに設定すると、システム・ファイルが意図せずに削除または変更されることがあります。できるだけ別のディレクトリーでコマンドを実行してください。ただし、システム管理タスクを自動化しているため、パスをルート・ディレクトリーにする必要がある場合は、サーバーの「パス」をルート・ディレクトリーに設定し、ステップの「ディレクトリー」を「../」に設定し、ステップの「パス」を「絶対」に設定してください。

- ステップの「パス」オプションが「相対」に設定されている場合、システムは、サーバー・パス、プロジェクト名、タグ、およびステップの「ディレクトリー」フィールドを一緒に追加することによって、ステップのパスを作成します。「ディレクトリー」フィールドの値は、ジョブの作業ディレクトリーを基準とする相対パスになります。

ステップの値	コマンドの結果生じるパス
サーバーの「パス」フィールド: C:/BuildForge プロジェクト名: My Project タグ: Job_5 ステップの「ディレクトリー」フィールド: /bin ステップの「パス」の設定: 相対	C:¥BuildForge¥My_Project¥Job_5¥bin パスの太字の部分がまだ存在していない場合、その部分がシステムによって作成されます。 注: プロジェクト・ディレクトリーを作成すると、システム設定の「無効な相対ディレクトリー文字」に指定されている文字が、下線文字に変更されます。デフォルトでは、この設定にスペースおよびバックチック文字が含まれており、「My Project」という名前のプロジェクトは、「My_Project」という名前のプロジェクト・ディレクトリーを受け取ります。

サーバーの「パス」フィールドまたはステップの「ディレクトリー」フィールドで指定されたディレクトリーが存在しない場合、ステップは失敗します。システムは、これらのディレクトリーを作成しません。ステップの「ディレクトリー」フィールドで指定されるパスの部分は、プロジェクト中の直前のステップで明示的に作成しておく必要があります。

通常、プロジェクト初期のステップは、ソース・コード制御からディレクトリーのツリーをチェックアウトし、後続のステップがそれらのディレクトリーで機能します。

新規ステップを追加すると、システムは「パス」に対して選択した最後の設定（「相対」または「絶対」のどちらか）を記録し、その設定を新規ステップのデフォルトとして使用します。

## パスのスラッシュ

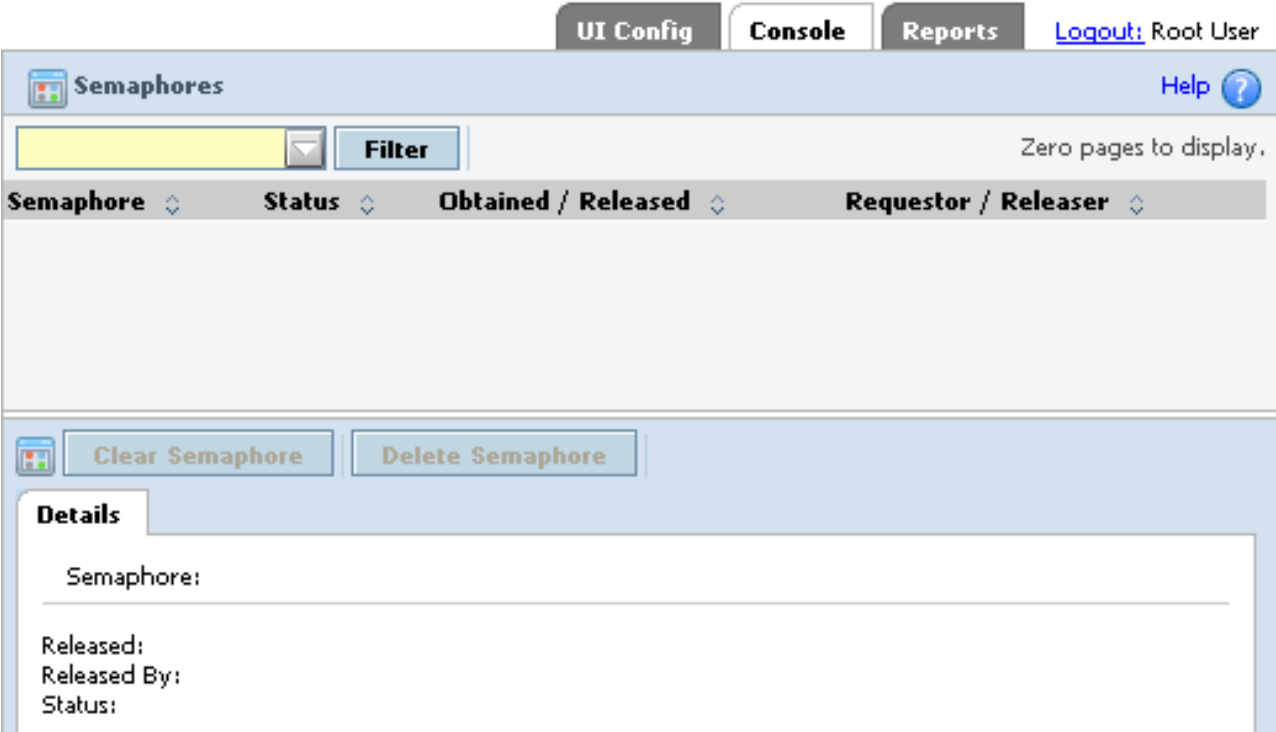
「パス」または「ディレクトリー」フィールドにパスを入力すると、システムは、円記号（またはバックスラッシュ）を普通のスラッシュに変換します。システムが実パスを作成するときには、使用されているサーバーが Windows® コンピューターであれば、システムはスラッシュを円記号（またはバックスラッシュ）に変換します。このため、いずれのタイプのスラッシュも、パスで使用できます。

注: システムは、ステップのコマンド・フィールドにあるスラッシュは変更しません。ターゲット・サーバーでの必要に応じて、バック・スラッシュまたはスラッシュを使用してください。

## セマフォ

セマフォは、相互に排他的な (mutex) リソースをセットアップする、システムのグローバルな信号フラグです。セマフォを使用して、いくつかのプロセスに、その他のプロセスが完了するのを待機させます。

「ジョブ」 > 「セマフォ」を使用して、使用中のジョブ・セマフォを表示できます。また、セマフォをクリアすることもできます。セマフォのクリアは、ハングしたジョブまたは取り消されたジョブがそのセマフォをリリースできないときに必要になることがあります。



.semget コマンドと .semput コマンドというドット・コマンドのペアによって、セマフォを実装します。 .semget コマンドを使用して、ラベルを「確保します」。ステップがラベルを取得した後、同一ラベルを取得しようとする (任意のプロジェクトにある) その他のステップは、初めの要求者がそのラベルを解放するまで待機する必要があります。

注: .semget コマンドを含むステップは、セマフォがリリースされるまで待機します。ジョブが失敗して、セマフォがアクティブのままになっている場合は、そのセマフォを使用するジョブを再び実行できるように、セマフォを手動でクリアする必要があります。

例えば、プリンター・ドライバーを作成するプログラムがあり、常に 1 つのプロセスだけが、そのプログラムを使用するようにしたい場合を考えます。プログラムを呼び出す各プロジェクト内で、以下のコマンド行によって、3 つのステップをセットアップします。

ステップ	コマンド行
セマフォの取得	.semget \$BF_PROJECTNAME_PHYS
ドライバー作成プログラムの実行	printdrivermaker.exe windows

ステップ	コマンド行
セマフォの解放	<code>.semput \$BF_PROJECTNAME_PHYS</code>

負荷の高いサーバーや、単一ユーザー・ライセンスのソフトウェア・プログラムなど、組織の重要なリソースに対して、セマフォを確立できます。保護したいリソースを使用する各ステップを、`.semget` コマンドと `.semput` コマンドでラップしてください。

セマフォは、以下の規則に従います。

- 使用時に、セマフォのドット・コマンドは、ステップで唯一のドット・コマンドでなければなりません。
- システムは、ラベルをそのままの状態で使用します。ラベル・パーサーを混乱させる可能性があるので、ラベルで特殊文字を使用したり、末尾に空白文字を付けたりしないでください。
- セマフォはグローバルであり、スレッド・ブロックに加えて、個々のプロジェクトを同期化させるために使用できます。
- セマフォは、最初にアクセスされたときに作成されます。
- プロジェクトによって取得されたセマフォは、そのプロジェクトの終了時に、自動的に解放されます。
- 2 つのステップが、同時にセマフォを要求した場合、どちらのステップがロックを取得するかについては、保証されている順序はありません。

## セマフォの手動でのクリア

システムは、プロジェクトが終了したときに、そのプロジェクトによって作成されたすべてのセマフォを自動的にリリースします。ただし、プロジェクトが異常終了したためにセマフォがリリースされなかった場合には、手動でセマフォをクリアできます。

1. 「ジョブ」 > 「セマフォ」を選択して、セマフォのリストを表示します。
2. クリアするセマフォを選択します。
3. 「クリア」をクリックします。

---

## 第 22 章 レポートの処理

このトピックでは、Rational Build Forge で提供されている組み込みレポートについて説明します。また、別個にライセンス交付されるレポート・ツールである、クイック・レポートを使用してレポートを作成および実行する方法についても説明します。

---

### レポートについて

レポートには、パフォーマンス・モジュールおよび照会パネルからアクセスする組み込みレポートと、クイック・レポート・ツールを使用して作成するレポートの、2つのカテゴリがあります。


最初に、右上隅にある「レポート」タブをクリックして、作成するレポートの選択項目を表示します。

- 標準レポートを表示するには、「パフォーマンス」パネルを選択します。
- 定義済みの照会レポートを表示して実行するには、「照会」パネルを選択します。
- 独自のレポートを作成したい場合には、「クイック・レポート (Quick Report)」パネルを選択します。クイック・レポートは、Rational Build Forge からライセンス交付されるオプションです。

### パフォーマンス・パネルと照会パネル

パフォーマンス・パネルは、Build Forge データを使用していくつかの標準レポートを作成します。パフォーマンス・レポートを開くときに、結果が自動的に更新され、常に現行データが表示されるようになります。

パフォーマンス・パネルには、定義済み照会レポートを含む照会パネルもあります。これらのレポートを実行するには、必要なデータを指定して、「クイック・ス

タート」アイコン  をクリックします。照会レポートは、現行データに対して実行されます。最新データに基づく結果を見るためには、レポートを再実行してください。

パフォーマンス・パネルと照会パネル内のレポートは変更できません。設計上、フィールドを変更したりレポート表示を変更したりできないようになっています。

### クイック・レポート・ツール

クイック・レポートは、BIRT (Eclipse ベースのオープン・ソース・レポート・システム) に基づくレポート・ツールです。クイック・レポートを使用する場合には、独自のレポート設計を作成できます。用意されたレポート・タイプを使用してレポート作業を開始して、後で必要なレポート・フィールドを選択することができます。どのレポート・タイプでも、データ・ソースとして Rational Build Forge データベース内のデータが使用されます。



クイック・レポートで設計するすべてのレポートについて、表やグラフなどのいくつかのレポート・フォーマットから選択することができます。また、グループ化機能およびフィルター機能を使用して、レポートの表示を制御することができます。

## レポート出力にデータを表示するための前提条件

サンプル・レポートを含めて、すべてのレポートは Build Forge データベースから得られたデータを表示します。データベースにデータを取り込むには、最初にプロジェクトを作成してジョブを実行する必要があります。

**注:** データベースにデータが含まれていない状態でレポートを作成して実行すると、レポートの結果は空になります。さらに、レポート・エラーが発生する可能性があります。

## レポート内のデータを表示するためのアクセス権限

レポートを実行しているユーザーのアクセス権限により、レポート結果に組み込まれ、表示されるプロジェクト・データが決まります。

次のように、ユーザーのプロジェクト・アクセス権限により、レポート結果に組み込まれるプロジェクト・データが決まります。

- パフォーマンス・パネル内の標準レポートでは、レポート・ページを開いたユーザーのアクセス権限によって決まります。
- 照会パネル内の定義済みレポートでは、レポートを実行したユーザーのアクセス権限によって決まります。
- クイック・レポートでは、レポート設計者のアクセス権限ではなく、レポートを実行したユーザーのアクセス権限によって決まります。

そのほかに、レポート設計を変更してレポートを実行することのできるアクセス権限があります。 499 ページの『クイック・レポートのためのレポート・グループ権限』を参照してください。

## 組み込みレポートの結果を CSV ファイルにエクスポートする

いくつかの組み込みレポートでは、レポート結果を CSV ファイルとしてエクスポートできます。結果を保存したり、別のレポート・アプリケーションにインポートしたりしたい場合には、この機能を使用してください。

「CSV に保存 (Save to CSV)」リンクがある組み込みレポートでは、レポート結果をエクスポートできます。

**注:** クイック・レポートでは、この機能は提供されません。

レポート結果を CSV ファイル・フォーマットにエクスポートするには、次のようにします。

1. レポートを実行して結果を生成します。
  - 「分析レポート」リンクを選択してステップおよびサーバー・パフォーマンス・レポート（「レポート > パフォーマンス」）を開きます。
  - 照会パネル（「レポート > 照会」）で任意のレポートを実行します。
2. レポート結果ページから「CSV に保存 (Save to CSV)」をクリックします。

3. 「ファイル・ダウンロード (File Download)」ダイアログで、「開く (Open)」または「保存」をクリックします。
  - CSV ファイル・フォーマットをサポートする任意のアプリケーションを使用して、CSV ファイルを開きます。例えば、Microsoft Excel または普段使用しているテキスト・ベースのエディターを使用します。
  - ファイルをローカル・ワークステーションまたはネットワーク上の別の場所に保存します。

---

## パフォーマンスに関する標準レポート

標準レポートには、標準レポート・ページを開いた時点での現行データに基づく結果が表示されます。

パフォーマンス・パネルは、現行のプロジェクトおよびジョブのデータを使用して、以下のようないくつかの標準レポートを作成します。

- すべてのプロジェクトに関するジョブ・パフォーマンス・レポート
- プロジェクトごとのジョブ所要時間レポート
- プロジェクトごとのステップおよびサーバー分析レポート

## プロジェクトのジョブ・パフォーマンス統計の表示

パフォーマンス・パネルの重要な部分として、すべてのプロジェクトのジョブ統計の全体像を要約するジョブ・パフォーマンス・レポートが含まれています。

- このレポートは、最新のジョブに関する完了日時とジョブ所要時間を表示します。
- すべてのジョブに関する統計として、成功したジョブ、失敗したジョブ、および警告が出されて成功したジョブの数が表示されます。

データ・ソース: Rational Build Forge データベース

レポート・フォーマット: テーブル

表示オプション: リストに表示するプロジェクトを制御するフィルターを作成します。

フィールド:

- 信頼性間隔: ジョブ平均所要時間よりどの程度長いかわかりやすい値の範囲 (秒)。次回のジョブの所要時間は、95% の確率でこの値の範囲内に収まります。例えば、信頼性間隔が 5.88 秒である場合、次回のジョブの所要時間は、ジョブの平均所要時間より 5.88 秒短い時間から 5.88 秒長い時間までの間に 95% の確率で収まります。

ジョブ・パフォーマンス統計レポートを表示するには、以下の手順を実行します。

1. 「レポート」タブを開きます。
2. 「パフォーマンス」パネルをクリックします。

すべてのプロジェクトに関するジョブ・パフォーマンス・レポートで、結果の表が表示されます。

## プロジェクトのジョブ所要時間の表示

パフォーマンス・パネルには、プロジェクトのすべてのジョブの所要時間をグラフで示すジョブ所要時間レポートが含まれます。

データ・ソース: Rational Build Forge データベース

レポート・フォーマット: グラフ

プロジェクト・ジョブ所要時間レポートを表示するには、以下の手順を実行します。

1. 「レポート」タブを開きます。
2. 「パフォーマンス」パネルをクリックします。
3. リスト内のプロジェクト名をクリックします。

ジョブの所要時間をグラフで表す、プロジェクト・ジョブ所要時間レポートが表示されます。

## プロジェクトごとのステップおよびサーバー・パフォーマンスの表示

パフォーマンス・パネルには、プロジェクトごとのステップおよびサーバー分析レポートが含まれます。また、プロジェクト内のステップに関するクリティカル・パス分析レポートも含まれます。

ステップおよびサーバー分析レポートは、各プロジェクト・ステップごとに、そのステップを実行したサーバーをリストし、異なる実行（例えば、最も速い実行と最も遅い実行）についてそのステップの実行に要した時間を示します。ステップ所要時間は、成功、失敗、または警告にカテゴリー化されます。カテゴリーにより、時間を誤って比較する（例えば、成功した時間と失敗した時間を比較するなど）ことがなくなります。

ステップのクリティカル・パス分析レポートは、成功したジョブに関するいくつかの重要なメトリックをリストします。

データ・ソース: Rational Build Forge データベース

レポート・フォーマット: テーブル

キー: -/-/ というキーは、ステップの 2 つの部分の情報を示します。つまり、以下の例のように、ステップ結果と秒単位のステップ所要時間を示します。

14/-/	そのステップは成功し、合計の実行時間は 14 秒でした。
成功/警告/失敗	
-/-/1	そのステップは失敗し、合計の実行時間は 1 秒でした。
成功/警告/失敗	
-/1/-	そのステップは警告付きで成功し、合計の実行時間は 1 秒でした。
成功/警告/失敗	

フィールド:

- 成功したステップ: すべてのジョブについてそのステップが成功した回数。
- 平均実行: 成功したジョブの平均ジョブ所要時間。
- 信頼性間隔: ジョブ平均所要時間よりどの程度長いかわかりやすい範囲 (秒)。次のジョブの所要時間は、95% の確率でこの値の範囲内に収まります。例えば、信頼性間隔が 5.88 秒である場合、次のジョブの所要時間は、ジョブの平均所要時間より 5.88 秒短い時間から 5.88 秒長い時間までの間に 95% の確率で収まります。
- 最長所要時間の確率 (Probability Longest Duration): 次のジョブの所要時間が成功したジョブの最長所要時間と同じになる確率。
- 最短所要時間の確率 (Probability Shortest Duration): 次のジョブの所要時間が成功したジョブの最短所要時間と同じになる確率。

分析レポートを表示するには、以下の手順を実行します。

1. 「レポート」タブを開きます。
2. 「パフォーマンス」パネルをクリックします。
3. リストからプロジェクトを選択して、「分析」をクリックします。

ステップおよびサーバーに関する分析レポートとクリティカル・パス分析レポートにより、結果が表で示されます。

---

## 照会のための定義済みレポート

現行のプロジェクトおよびジョブ・データを使用して定義済みレポートを実行するために、日付範囲またはその他の必須データを提供します。

照会パネルは、現行データを使用して、以下のようないくつかの定義済みレポートを作成します。

- プロジェクトおよびステップのセレクト使用率履歴
- すべてのサーバーの現行サーバー・マニフェスト
- ジョブ結果成功/失敗/警告履歴
- 日付範囲内のジョブのサーバー使用率履歴
- MD5 値に基づくジョブ・ファイルの検索

## セレクト使用率履歴の表示

すべてのプロジェクトのセレクト使用率を表示するレポート。


照会パネルには、すべてのプロジェクトのセレクト使用率を表示するレポートが含まれます。このレポートを使用して、プロジェクトとステップに割り当てられたセレクトを調べることができます。このレポートは、プロジェクトまたはステップを実行するサーバーを自動的に選択するためにセレクトを使用している場合に、特に便利です。

データ・ソース: Rational Build Forge データベース

レポート・フォーマット: テーブル

表示オプション: 「レポート出力のフラット化 (Flatten report output)」を選択して、ツリー階層を展開し、すべてのレポート結果をリストに表示します。

ステップおよびジョブのサーバー使用率レポートを表示するには、次のようにします。

1. 「レポート」タブを開きます。
2. 「照会」パネルをクリックします。
3. プロジェクト・セクターおよびステップ・サーバー・レポートが必要であれば、「クイック・スタート」アイコン  をクリックしてください。

セクター使用率レポートで、結果の表が表示されます。

## サーバーごとに現行サーバーのマニフェストを表示する

すべてのサーバーのサーバー・マニフェスト・プロパティと値をリストするレポート。

照会パネルには、すべてのサーバーのサーバー・マニフェスト・プロパティと値をリストするレポートが含まれます。このレポートを使用して、サーバー・プロパティを比較したり、同じセクターによって選択される可能性のあるサーバーを識別したり、サーバーに必要なプロパティを識別したりできます。


このレポートは、そのサーバーに割り当てられたすべてのマニフェスト・プロパティを、サーバーに割り当てられたコレクター別に表示します。すべてのサーバーに自動的に割り当てられる特別なマニフェスト・プロパティを含めるためには、「BF\_ プロパティの表示」を選択します。

データ・ソース: Rational Build Forge データベース

レポート・フォーマット: テーブル

表示オプション: 「レポート出力のフラット化 (Flatten report output)」を選択して、ツリー階層を展開し、すべてのレポート結果をリストに表示します。

ご使用のサーバーの現行サーバー・マニフェストのレポートを表示するには、次のようにします。

1. 「レポート」タブを開きます。
2. 「照会」パネルをクリックします。
3. (オプション) 特別なマニフェスト・プロパティをレポート結果に含めるために「BF\_ プロパティの表示」を選択します。
4. 現行サーバー・マニフェスト・レポートが必要であれば、「クイック・スタート」アイコン  をクリックしてください。

現行サーバー・マニフェスト・レポートで、結果の表が表示されます。

## ジョブの成功/失敗/警告結果の表示

指定された日時範囲内に完了したジョブについて、成功/失敗/警告結果をリストするレポートが生成されます。

照会パネルには、指定された日時範囲内に完了したジョブについて、成功/失敗/警告結果をリストするレポートが含まれます。このレポートは、すべてのプロジェクトのすべてのジョブに適用されます。このレポートを使用すると、完了したジョブのクイック・ヘルス・チェックを行うことができます。

データ・ソース: Rational Build Forge データベース

レポート・フォーマット: テーブル


表示オプション: 「レポート出力のフラット化 (Flatten report output)」を選択して、ツリー階層を展開し、すべてのレポート結果をリストに表示します。

プロジェクトのジョブ結果レポートを表示するには、以下の手順を実行します。

1. 「レポート」タブを開きます。
2. 「照会」パネルをクリックします。
3. 結果レポートの対象範囲として、デフォルトの時刻範囲を使用するか、時刻範囲を指定します。時刻の指定には、UNIX の時刻形式を使用してください。UNIX の時刻形式は、1970 年 1 月 1 日を起点とした秒数です。標準的な日時形式から UNIX 時刻を計算する際には、[http://www.onlineconversion.com/unix\\_time.htm](http://www.onlineconversion.com/unix_time.htm) にあるようなツールを使用することができます。

日付は、その日の深夜 0 時 (00:00) から始まり、翌日の深夜 0 時に終わります。例えば、2009 年 7 月 21 日に完了したすべてのジョブを返すには、次の UNIX 時刻を入力します。

開始時刻: 07/21/09 00:00 AM	1248134400
終了時刻: 07/22/09 00:00 AM	1248220800

4. ジョブ結果レポートが必要であれば、「クイック・スタート」アイコン  をクリックしてください。

ジョブ結果レポートで、結果の表が表示されます。

## 日付範囲内のジョブのサーバー使用率の表示

そのプロジェクトのすべてのジョブに関するサーバーをリストするサーバー使用率レポート。

照会パネルには、そのプロジェクトのすべてのジョブに関するサーバーをリストするサーバー使用率レポートが含まれます。このレポートは、ステップごとに、そのステップを実行したサーバーを識別し、ステップ所要時間を示します。このレポートを使用して、さまざまなサーバーのステップ所要時間を比較できます。

データ・ソース: Rational Build Forge データベース

レポート・フォーマット: テーブル




表示オプション: 「レポート出力のフラット化 (Flatten report output)」を選択して、ツリー階層を展開し、すべてのレポート結果をリストに表示します。

ステップのサーバー使用率レポートを表示するには、以下の手順を実行します。

1. 「レポート」タブを開きます。
2. 「照会」パネルをクリックします。
3. 「開始 (From)」と「終了 (To)」の日付を選択します。

日付は、その日の深夜 0 時 (00:00) から始まり、翌日の深夜 0 時に終わります。

4. サーバー使用率レポートが必要であれば、「クイック・スタート」アイコン  をクリックしてください。

サーバー使用率レポートで、結果の表が表示されます。

## MD5 値を使用して行うジョブ・ファイルの検索

ジョブ BOM を調べて MD5 値が一致するファイルを検索するレポート。 MD5 ハッシュ値は、ファイルのデジタル指紋です。

照会パネルには、ジョブ BOM を調べて MD5 値が一致するファイルを検索し、そのファイルが見つかった場合にジョブ・タグとファイルの場所をリストするレポートが含まれます。あるファイルの特定のバージョンを探したい場合にその MD5 値が分かっているときには、このレポートを使用してください。

前提条件として、プロジェクトに、MD5 値を保管するための .scan ベースラインまたは .scan チェックポイント・コマンドが含まれている必要があります。詳しくは、458 ページの『.scan』の参照情報を参照してください。 .scan コマンドは、以下のタスクを行います。

- あるステップの作業ディレクトリー内にあるファイルのリストを作成する
- その作業ディレクトリー内のファイルの MD5 値を生成し、MD5 値をジョブ BOM に保管する

データ・ソース: Rational Build Forge データベース

レポート・フォーマット: テーブル

表示オプション: 「レポート出力のフラット化 (Flatten report output)」を選択して、ツリー階層を展開し、すべてのレポート結果をリストに表示します。

ジョブ BOM から MD5 値を検索するには、以下のようになります。

1. 「レポート」タブを開きます。
2. 「照会」パネルをクリックします。
3. 「MD5 検索値 (MD5 search value)」フィールドに、探したい MD5 値を入力します。
4. MD5 ファイル・レポートが必要であれば、「クイック・スタート」アイコン



をクリックしてください。



MD5 ファイル・レポートで、結果の表が表示されます。

## クイック・レポートでのレポートの作成

クイック・レポートによって提供されるレポート・タイプを使用して、独自のレポート設計を作成します。クイック・レポートは、Rational® Build Forge® からライセンス交付されるオプションです。

クイック・レポートで、レポート設計の作成、保存、編集、および実行を行うことができます。レポートを他のユーザーと共用するには、共通ディレクトリーに保存します。

クイック・レポート内のレポート・タイプは、Build Forge® データベース内のデータを使用してレポートを作成します。さまざまなレポート・タイプにさまざまなレポート・フィールドが含まれます。レポート・タイプには、機能を表す名前 (例えば、Capacity や Build) が付いています。

レポートを作成するときは、表やグラフなど、いくつかのレポート・フォーマットからレポート・フォーマットを選択できます。グループ化機能とソート機能を使用すれば、レポートの表示を制御できます。

オプションで、ジョブ BOM ファイルでレポートを実行し、レポート結果を表示できます。 453 ページの『.report』の参照情報を参照してください。

最初に、「レポート」タブをクリックして、レポート作成のための選択項目を表示して、「クイック・レポート (Quick Report)」を選択します。

注: 8.0 用のクイック・レポートでは Perl エンジンによって生成されたレポートのみが表示されます。

## クイック・レポートのためのレポート・グループ権限

レポートをパブリック・リポジトリまたはプライベート・リポジトリに保管します。これにより、クイック・レポートで作成されたレポートを実行、変更、保存、および削除するユーザーを制御できます。

プライベート・レポートには、そのレポートを作成して保存した Rational Build Forge ユーザーのみがアクセスできます。

共通レポートには、適切な権限を持つアクセス・グループに所属するすべてのユーザーがアクセスできます。デフォルトでは、ビルド・エンジニア・グループに属するユーザーは、パブリック・レポートとして保存されたレポートに対する読み取り権限、書き込み権限、編集権限、および 3 つすべてのアクセス権限を持っています。

アクセス権を付与するには、ユーザーのアクセス・グループに以下のレポート権限のうちの 1 つ以上を割り当てます。

共通レポートの読み取り	クイック・レポートの共通レポート・リポジトリに保存されたレポートを実行するための権限。
-------------	---

レポートの保存、編集、または削除	クイック・レポートの共通リポジトリに保存されたレポートを保存、編集、または削除するための権限。
------------------	---

下の表に、レポート権限グループの権限と、アクセス・グループに割り当てられたデフォルト権限を示します。

権限を管理するために、コンソールの UI で「管理」>「権限」の順に選択します。

	ビルド・エンジニア	開発者	ゲスト	演算子	セキュリティー	システム管理者
共通レポートの読み取り	はい	はい			はい	はい
共通レポートの保存	はい					
共通レポートの編集	はい					
共通レポートの削除	はい					

## クイック・レポートのためのレポート・タイプ参照

レポート・タイプには、その内容（例えば、容量 やビルド）を説明する機能名が付いています。このトピックでは、レポート・タイプごとにレポートの目的、レポート・フィールド、およびレポート例について説明します。

**分析**    **説明:** ステップ所要時間およびステップ実行順序を使用してステップ・レベルでジョブ・パフォーマンスをレポートするには、分析レポート・タイプを使用します。

**レポート例:**

- ジョブに含まれるステップごとにステップ所要時間とステップ順序を表示する、表形式のレポートを作成します。ジョブをプロジェクト名によってグループ化します。
- 各ビルド・タグのステップ所要時間を表示する、折れ線グラフのレポートを作成します。ビルド・タグをステップ名によってグループ化します。

**フィールドの説明:**

フィールド名	説明
ビルド・タグ	このジョブ・タグは、プロジェクト・タグ・フォーマットに基づいた固有の ID です。
プロジェクト名	ユーザーが割り当てたプロジェクト名。
ステップ期間	ステップとそのステップがインライン化したすべてのステップの合計実行時間（秒）。
ステップ名	ユーザーが割り当てたステップ名。
ステップ・シーケンス	ステップ実行順序を識別する番号。

**ビルド 説明:** プロジェクト・レベルでジョブ・パフォーマンスをレポートするには、ビルド・レポート・タイプを使用します。

**レポート例:**

- 各ビルド・タグのビルド結果、開始時刻、および所要時間を表示する、表形式のレポートを作成します。ビルド・タグをプロジェクトによってグループ化します。
- 各プロジェクトのビルド・カウントを表示する、棒グラフのレポートを作成します。プロジェクトによってグループ化します。

**フィールドの説明:**

フィールド名	説明
ビルド・カウント (集約フィールド)	完了したジョブと失敗したジョブを含む、合計ジョブ・カウント。
ビルド期間	ジョブの合計実行時間 (秒)。
ビルド結果	ジョブの結果: 成功、成功 (警告あり)、または失敗。
ビルド開始時刻	ジョブの開始時刻。
ビルド状態	ビルドの状態 (実行中、完了、保存済み、またはロック)。
ビルド・タグ	このジョブ・タグは、プロジェクト・タグ・フォーマットに基づいた固有の ID です。
プロジェクト名	ユーザーが割り当てたプロジェクト名。
セクター名	ユーザーが割り当てたセクター名。
ユーザー・ログイン	ジョブを開始したユーザーの Build Forge ログイン名またはユーザー名。
ユーザー名	ジョブを開始したユーザーの名前。

**容量 説明:** プロジェクト別にジョブ・パフォーマンスをレポートするには、容量レポート・タイプを使用します。

**レポート例:**

- 各ビルド・タグのビルド開始、ビルド期間、ビルド平均期間、およびビルド結果を表示する、表形式のレポートを作成します。ビルド・タグをプロジェクトによってグループ化します。
- 各プロジェクトの平均ビルド時間を表示する、グラフ形式のレポートを作成します。プロジェクトをセクターによってグループ化します。

**フィールドの説明:**

フィールド名	説明
ビルド平均期間 (集約フィールド)	ジョブの合計数 (完了したジョブと失敗したジョブの両方) に基づく、平均ジョブ実行時間。
ビルド期間	ジョブの合計実行時間 (秒)。
ビルド結果	ジョブの結果: 成功、成功 (警告あり)、または失敗。
ビルド開始時刻	ジョブの開始時刻。
ビルド・タグ	このジョブ・タグは、プロジェクト・タグ・フォーマットに基づいた固有の ID です。

フィールド名	説明
最後のビルド期間 (集約フィールド)	最後のジョブの実行時間。最後のジョブを完了させるために要した合計時間 (秒単位)。
プロジェクト名	ユーザーが割り当てたプロジェクト名。
セレクトター名	ユーザーが割り当てたセレクトター名。

## プロジェクト

**説明:** プロジェクトごとのサーバー、環境、およびステップの使用に関して報告を行い、プロジェクトごとのステップ・パフォーマンスを報告するには、プロジェクト・レポートを使用します。

### レポート例:

- プロジェクト、クラス名、およびプロジェクト環境を表示する、表形式のレポートを作成します。プロジェクトをサーバーによってグループ化します。
- ステップ、ステップ結果、ステップの順序、ステップ環境、およびサーバーを表示する、表形式のレポートを作成します。プロジェクトによってステップをグループ化し、順序によってソートします。
- 各ステップのステップ・カウントを表示する、棒グラフのレポートを作成します。ステップをステップ結果によってグループ化します。

### フィールドの説明:

フィールド名	説明
クラス名	ユーザーがプロジェクトに割り当てたクラス (例えば、実動またはテスト)
失敗したステップ・カウント (集約フィールド)	選択したグループ・フィールド (例えば、プロジェクト、サーバー、またはその他のフィールド名) に関する、失敗したステップの数。
成功したステップ・カウント (集約フィールド)	選択したグループ・フィールド (例えば、プロジェクト、サーバー、またはその他のフィールド名) に関する、成功したステップの数。
プロジェクト環境変数名	プロジェクトの環境変数を定義するために使用するプロジェクト環境変数の名前。
プロジェクト・レベル	ユーザーが割り当てた、プロジェクトのアクセス・グループ。
プロジェクト名	ユーザーが割り当てたプロジェクト名。
サーバー名	ユーザーが割り当てたサーバー名。
ステップ・カウント (集約フィールド)	選択したグループ・フィールド (例えば、プロジェクトまたはサーバー) に関するステップの総数。
ステップ環境変数名	ステップの環境変数を定義するために使用するステップ環境変数の名前。
ステップ・レベル	ユーザーが割り当てた、ステップのアクセス・グループ。
ステップ名	ユーザーが割り当てたステップ名。
ステップ結果	ステップの結果: 成功、成功 (警告あり)、または失敗。
ステップ・シーケンス	ステップ実行順序を識別する番号。

## ステップ測定基準

**説明:** プロジェクトごとにステップの成功および失敗統計を報告するためには、ステップ測定基準レポートを使用します。

### レポート例:

- ステップ名、ステップ・カウント、および成功したステップと失敗したステップのパーセントを表示する、表形式のレポートを作成します。ステップをプロジェクトによってグループ化します。
- ビルド・タグ別にステップ所要時間を表示する、折れ線グラフのレポートを作成します。ビルド・タグをステップ名によってグループ化します。

### フィールドの説明:

フィールド名	説明
ビルド・タグ	このジョブ・タグは、プロジェクト・タグ・フォーマットに基づいた固有の ID です。
失敗したステップのパーセント (集約フィールド)	合計ステップ・カウントに対する失敗したステップのパーセント。この合計ステップ数は、選択したグループ・フィールド (例えばプロジェクトなど) のものです。
成功したステップのパーセント (集約フィールド)	合計ステップ・カウントに対する成功したステップのパーセント。この合計ステップ数は、選択したグループ・フィールド (例えばプロジェクトなど) のものです。
プロジェクト名	ユーザーが割り当てたプロジェクト名。
サーバー名	ユーザーが割り当てたサーバー名。
ステップ平均期間 (集約フィールド)	ステップの合計数 (完了したステップと失敗したステップの両方) に基づく、平均ステップ実行時間。
ステップ・カウント (集約フィールド)	完了したステップと失敗したステップを含む、合計ステップ・カウント。この合計ステップ数は、選択したグループ・フィールド (例えばプロジェクトなど) のものです。
ステップ期間	ステップとそのステップがインライン化したすべてのステップの合計実行時間 (秒)。
ステップ名	ユーザーが割り当てたステップ名。
ステップ結果	ステップの結果: 成功、成功 (警告あり)、または失敗。
ステップ・シーケンス	ステップ実行順序を識別する番号。
ステップ開始時刻	ステップの開始時刻。

## 品質

**説明:** プロジェクトごとにジョブの成功および失敗統計を報告するためには、品質レポートを使用します。

### レポート例:

- ビルド・タグごとにビルド結果を表示する、表形式のレポートを作成します。ビルド・タグをプロジェクトによってグループ化します。
- プロジェクトごとにすべての成功したビルドと失敗したビルドのパーセントを報告書表示する、表形式のレポートを作成します。プロジェクトによってグループ化します。

- プロジェクトごとにビルド・カウントを表示する、棒グラフのレポートを作成します。プロジェクトをビルド結果によってグループ化します。

**フィールドの説明:**

フィールド名	説明
ビルド・カウント (集約フィールド)	完了したジョブと失敗したジョブを含む、合計ジョブ・カウント。
ビルド結果	ジョブの結果または状況: 成功、成功 (警告あり)、または失敗。
ビルド開始時刻	ジョブの開始時刻。
ビルド・タグ	このジョブ・タグは、プロジェクト・タグ・フォーマットに基づいた固有の ID です。
失敗したビルドのパーセント (集約フィールド)	合計ビルド・カウントに対する失敗したビルドのパーセント。合計ビルド・カウントは、選択したグループ・フィールド (例えばプロジェクトなど) を対象としています。
成功したビルドのパーセント (集約フィールド)	合計ビルド・カウントに対する成功したビルドのパーセント。合計ビルド・カウントは、選択したグループ・フィールド (例えばプロジェクトなど) を対象としています。
プロジェクト名	ユーザーが割り当てたプロジェクト名。

## リソース

**説明:** プロジェクトおよびサーバーごとにステップおよびジョブのパフォーマンスを報告するためには、リソース・レポートを使用します。

**レポート例:**

- サーバーごとにステップ実行時間を表示する、表形式のレポートを作成します。ステップ順序、ステップ名、サーバー名、ステップ開始時刻、およびステップ期間を選択してください。ステップ順序および開始時刻でソートします。
- サーバーごとにジョブ実行時間を表示する、表形式のレポートを作成します。ビルド・タグ、サーバー名、ビルド開始時刻、ビルド期間、およびビルド結果を選択してください。ビルド・タグをプロジェクトによってグループ化し、ビルド開始時刻によってソートします。

**フィールドの説明:**

フィールド名	説明
ビルド期間	ジョブの合計実行時間 (秒)。
ビルド結果	ジョブの結果: 成功、成功 (警告あり)、または失敗。
ビルド開始時刻	ジョブの開始時刻。
ビルド・タグ	このジョブ・タグは、プロジェクト・タグ・フォーマットに基づいた固有の ID です。
プロジェクト名	ユーザーが割り当てたプロジェクト名。

フィールド名	説明
セレクトター名	ユーザーが割り当てたセレクトター名。
サーバー名	ユーザーが割り当てたサーバー名。
ステップ期間	ステップとそのステップがインライン化したすべてのステップの合計実行時間 (秒)。
ステップ・シーケンス	ステップ実行順序を識別する番号。
ステップ開始時刻	ステップの開始時刻。

## BOM 説明:

ジョブ BOM 内の情報を使用してレポートを作成するには、BOM レポートを使用します。BOM レポートは、以下のいずれかのデータ・セットに対して実行することができます。

- データベース内のすべてのプロジェクト (デフォルト)
- 単一プロジェクト
- 単一プロジェクト内の 1 つ以上のビルド

選択できるレポート・フィールドは、データ・セットごとに異なります。レポート・フィールドには、ステップ・マニフェスト・プロパティ、**.scan** コマンドの出力、アダプターによって記録された出力、および **.bom** コマンドを使用して指定されたユーザー定義の列を含めることができます。

## レポート例:

- 複数のプロジェクトでの **.scan** コマンドの出力を表示する、表形式のレポートを作成します。ビルド・タグ、BOM データ、BOM パス、および BOM タイプのフィールドを選択してください。ビルド・タグをプロジェクトによってグループ化します。
- 複数のプロジェクトでフィルター・アクションが呼び出される回数を表示する、表形式のレポートを作成します。ステップ名、フィルター・イベントのカウント・タイプ、およびフィルター・カウントを選択してください。ステップをプロジェクトによってグループ化します。

## フィールドの説明:

フィールド名	説明
BOM データ	プロジェクトに <b>.scan checkpoint</b> コマンドを組み込むと、「BOM データ」フィールドにスキャン対象ファイルの MD5 値が表示されます。
BOM パス	プロジェクトに <b>.scan checkpoint</b> コマンドを組み込むと、「BOM パス」フィールドにスキャン対象ファイルのファイル・パスが表示されます。
BOM タイプ	プロジェクトに <b>.scan checkpoint</b> コマンドを組み込むと、スキャン対象ファイルについて、「BOM タイプ」フィールドにそのパスが D (ディレクトリー)、F (ファイル)、S (シンボリック・リンク) のいずれであるかが示されます。
ビルド・タグ	このジョブ・タグは、プロジェクト・タグ・フォーマットに基づいた固有の ID です。



フィールド名	説明
フィルター・イベント・カウント	ステップ出力内でパターン・マッチを検索した結果として、フィルターによってフィルター・アクションが呼び出された回数。
フィルター・イベント・カウント・タイプ	ステップ出力内でパターン・マッチが検出されたときにフィルターによって呼び出されたフィルター・アクション。
プロジェクト名	ユーザーが割り当てたプロジェクト名。
結果の説明	結果レコード内の説明フィールド。通常、ここにはステップ名のコピーが入ります。
ステップ名	ユーザーが割り当てたステップ名。
セクター名	ユーザーが割り当てたセクター名。
サーバー名	ユーザーが割り当てたサーバー名。
ステップ期間	ステップとそのステップがインライン化したすべてのステップの合計実行時間 (秒)。
ステップ結果	ステップの結果: 成功、成功 (警告あり)、または失敗。

## クイック・レポートのためのレポート・フォーマットおよび表示の参照

クイック・レポートには、テーブル、棒グラフ、線グラフ、円グラフの共通レポート・フォーマットが備わっています。


注: クイック・レポートで提供されるサンプル・レポートには、データが含まれていません。レポート・データを表示するには、最初に管理コンソールでプロジェクトを作成してジョブを実行する必要があります。

### 表レポート・フォーマット

#### サンプル

サンプルの表レポートを表示するには、クイック・レポートで「**サンプル分析-ステップ所要時間 (SampleAnalytic-StepDuration)**」レポートを選択してください。

表示オプションを表示するには、「サンプル分析-ステップ所要時間

(SampleAnalytic-StepDuration)」レポートの「**編集**」アイコン  を選択してください。

#### 要件

表の場合には、以下の要件に従ってください。

- 少なくとも 1 つのレポート・フィールドを選択します。
- 集約レポート・フィールドを選択する場合、グループ化フィールドを選択する必要があります。511 ページの『レポート・フォーマットと集約レポート・フィールドに関する選択要件』を参照してください。


#### レポート・フィールド

「レポート・フィールド (Report Field)」で、レポートのテーブル列を 1 つ以上選択します。

注: 選択リストにおいてフィールドの順序は重要です。

最初のフィールドは表の最初の列になり、最後のフィールドは表の最後の列になります。表の最初の列は、レポート結果をグループ化するためにも使用されます。

レポート・フィールドを試すには、「サンプル分析-ステップ所要時間


(Sample Analytic-StepDuration)」レポートの「編集」アイコン  を選択し、以下の手順を実行します。

1. 選択リストでフィールド順序を変更します。
2. 選択内容を保存します。
3. レポートを実行して、テーブル列の順序の変更を確認します。

### グループ・フィールド

フィールドをグループ・フィールドから選択できます。選択したフィールドは、表の最初の列にツリー・コントロールとして追加され、レポート結果をグループ化するために使用されます。複数のグループ・フィールドを選択したり、テーブル列として表示されないグループ・フィールドを選択したりできます。

グループ化を試すには、「サンプル分析-ステップ所要時間


(SampleAnalytic-StepDuration)」レポートの「編集」アイコン  を選択し、以下の手順を実行します。


1. グループ・フィールドとして「プロジェクト名」を選択します。
2. 選択内容を保存します。
3. レポートを実行します。

この結果、「プロジェクト名」のツリー・ノードが「ビルド・タグ」列に追加されます。


### ソート・フィールド

ソート・フィールドで、ソート可能なデータを含むテーブル列を選択できま

す。ソートの方向を指定するには、矢印を使用してください。上矢印  を使用した場合、データは昇順に（最低から最高の順に）ソートされます。

下矢印  を使用した場合、データは降順に（最高から最低の順に）ソートされます。


ソートを試すには、「サンプル分析-ステップ所要時間

(SampleAnalytic-StepDuration)」レポートの「編集」アイコン  を選択し、以下の手順を実行します。

1. ソート・フィールドとして「プロジェクト名」を選択し、矢印を使用してソート方向を指定します。
2. 選択内容を保存します。
3. レポートを実行します。

## 棒グラフ・レポート・フォーマット サンプル

サンプルの棒グラフを表示するには、クイック・レポートで「**サンプル容量-プロジェクト別の実行時間 (SampleCapacity-RunTimeByProject)**」レポートを選択してください。

表示オプションを表示するには、「サンプル容量-プロジェクト別の実行時間 (SampleCapacity-RunTimeByProject)」レポートの「**編集**」アイコン  を選択してください。

#### 垂直のグラフ・エレメント (y 軸)

データ・エレメントのセット (x 軸) に対して比較するデータ値 (y 軸) のフィールドを 1 つ選択します。フィールド名とデータ値の単位がグラフの垂直軸 (y 軸) に表示されます。

**注:** 垂直のグラフ・エレメントとして選択できるフィールドは 1 つのみです。

#### 水平のグラフ・エレメント (x 軸)

データ値 (y 軸) を使用して比較するデータ・エレメント・セット (x 軸) のフィールドを 1 つ選択します。そのフィールド名が x 軸ラベルとして表示されます。

データ値は、棒の頂点の位置に表示されます。

**注:** 水平のグラフ・エレメントとして選択できるフィールドは 1 つのみです。

#### グループ・フィールド

オプションとして、グループ・フィールドを選択することにより、グラフの水平軸 (x 軸) でデータ・エレメントをグループ化できます。

#### 要件

グラフの場合には、以下の要件に従ってください。

- x 軸と y 軸の両方のレポート・フィールドを選択します。
- x 軸と y 軸には別のレポート・フィールドを選択します。
- X 軸データ用のグループ化フィールドを選択します。
- 集約レポート・フィールドを選択する場合、グループ化フィールドを選択する必要があります。511 ページの『レポート・フォーマットと集約レポート・フィールドに関する選択要件』を参照してください。

#### 例:

棒グラフを試すには、以下のレポートを作成してみてください。

- 「ステップ」レポート・タイプ: ステップ名 (x 軸) ごとにステップ失敗率 (y 軸) を比較し、ビルド・タグで結果をグループ化します。
- 「容量」レポート・タイプ: プロジェクト (x 軸) ごとに最後のビルド期間 (y 軸) を比較し、プロジェクト名で結果をグループ化します。
- 「リソース」レポート・タイプ: ビルド・タグ (x 軸) ごとにビルド所要時間 (y 軸) を比較し、プロジェクト名でビルド・タグをグループ化します。

## 折れ線グラフ・レポート・フォーマット


### 説明

折れ線グラフは、時間の経過または一連のイベントに対応したデータの推移を表します。デフォルトのグループ化フィールドは x 軸レポート・フィールドです。複数のグループ化フィールドを選択した場合、レポートに複数の線が表示されます。

### サンプル

サンプルの折れ線グラフを表示するには、クイック・レポートで「**サンプル・リソース-持続時間 (SampleResource-DurationOverTime)**」レポートを選択してください。

表示オプションを表示するには、「サンプル・リソース-持続時間

(SampleResource-DurationOverTime)」レポートの「**編集**」アイコン  を選択してください。

### 垂直グラフ・エレメント (y 軸)

データ・エレメントのセット (x 軸) に対して比較するデータ値 (y 軸) のフィールドを 1 つ選択します。フィールド名とデータ値の単位がグラフの y 軸に表示されます。

注: 垂直のグラフ・エレメントとして選択できるフィールドは 1 つのみです。

### 水平グラフ・エレメント (x 軸)

データ値 (y 軸) を使用して比較するデータ・エレメント・セット (x 軸) のフィールドを 1 つ選択します。そのフィールド名がグラフの x 軸のラベルとして表示されます。

x 軸データ・エレメントのデータ値が、y 軸上の値と同じ値を示すドットの横に表示されます。連続線がドットを結び、折れ線グラフを形成します。

注: 水平のグラフ・エレメントとして選択できるフィールドは 1 つのみです。

### グループ・フィールド

オプションとして、グループ・フィールドを選択することにより、グラフの水平軸 (x 軸) でデータ・エレメントをグループ化できます。デフォルトのグループ化フィールドは x 軸レポート・フィールドです。複数のグループ化フィールドを選択した場合、レポートに複数の線が表示されます。

### 要件

グラフの場合には、以下のガイドラインに従ってください。

- x 軸と y 軸の両方のレポート・フィールドを選択します。
- x 軸と y 軸には別のレポート・フィールドを選択します。
- X 軸データ用のグループ化フィールドを選択します。
- 集約レポート・フィールドを選択する場合、グループ化フィールドを選択する必要があります。511 ページの『レポート・フォーマットと集約レポート・フィールドに関する選択要件』を参照してください。

## 例

折れ線グラフを試すには、以下のレポートを作成してみてください。

- 「ステップ」レポート・タイプ: ビルド・タグ (x 軸) ごとにステップ期間 (y 軸) を表示し、ステップ名で結果をグループ化します。
- 「ビルド」レポート: ビルド開始時刻 (x 軸) ごとに期間 (y 軸) を表示し、プロジェクト名で結果をグループ化します。
- 「容量」レポート・タイプ: ビルド・タグ (x 軸) ごとにビルド期間 (y 軸) を表示し、サーバー名で結果をグループ化します。

## 円グラフ・レポート・フォーマット

### Y 系列エレメント

円グラフのデータ値のフィールドを 1 つ選択します。扇形のサイズはデータ値を表します。フィールド名とデータ値の単位が円グラフに表示されます。

注: y 系列エレメントとして選択できるフィールドは 1 つのみです。

### X 系列エレメント

データ値 (y 系列) を使用して評価するデータ・エレメント・セット (x 系列) のフィールドを 1 つ選択します。扇形の数にはデータ・エレメントの数を表します。

注: x 系列エレメントとして選択できるフィールドは 1 つのみです。

### グループ・フィールド

円グラフにはグループが必要です。デフォルト・グループには x 系列エレメントが使用されます。別のグループを選択すると、データ値 (y 系列) に対して別の円グラフが生成されます。

### 要件

グラフの場合には、以下のガイドラインに従ってください。

- x 系列と y 系列の両方のレポート・フィールドを選択します。
- x 系列と y 系列には別のレポート・フィールドを選択します。
- X 軸データ用のグループ化フィールドを選択します。
- 集約レポート・フィールドを選択する場合、グループ化フィールドを選択する必要があります。511 ページの『レポート・フォーマットと集約レポート・フィールドに関する選択要件』を参照してください。

## 例

円グラフを試すには、以下のレポートを作成してみてください。

- 「プロジェクト」レポート・タイプ: 失敗したステップ・カウント (y 系列) とステップ名 (x 系列) を選択します。
- 「容量」レポート・タイプ: ビルド平均期間 (y 系列) とプロジェクト名 (x 系列) を選択します。
- 「分析」レポート・タイプ: ステップ期間 (y 系列) とステップ名 (x 系列) を選択し、ステップ名でグループ化を行います。

## レポート・フォーマットと集約レポート・フィールドに関する選択要件

有効なレポートを作成するためには、表、グラフ、および集約レポートのフィールドに関するいくつかの基本的な要件に従う必要があります。

### 集約レポート・フィールド

#### 要件

表またはグラフに集約レポート・フィールドを組み込む場合には、グループ化フィールドを含める必要があります。

#### 定義

集約レポート・フィールドには、Build Forge データベースの 1 つ以上のオリジナル・データ・フィールドから派生したデータが含まれます。集約レポート・フィールドのデータは、データベースには保管されません。

以下のレポート・フィールドは集約フィールドです。

分析: 最後のビルド期間	プロジェクト: 失敗したステップ・カウント	プロジェクト: 成功したステップ・カウント
プロジェクト: ステップ・カウント	ステップ測定基準: 失敗したステップのパーセント	ステップ測定基準: 成功したステップのパーセント
ステップ測定基準: ステップ平均期間	ステップ測定基準: ステップ・カウント	品質: ビルド・カウント
品質: 失敗したビルドのパーセント	品質: 成功したビルドのパーセント	ビルド: ビルド・カウント

### 表レポート・フォーマット

#### 要件

表の場合には、以下の要件に従ってください。

- 少なくとも 1 つのレポート・フィールドを選択します。
- 集約レポート・フィールドを選択する場合、グループ・フィールドを選択する必要があります。

### グラフ・レポート・フォーマット

#### 要件

棒グラフ、折れ線グラフ、および円グラフの場合には、以下の要件に従ってください。


- x 系列と y 系列の両方のレポート・フィールドを選択します。
- x 系列と y 系列には別のレポート・フィールドを選択します。
- X 軸データ用のグループ化フィールドを選択します。
- 集約レポート・フィールドを選択する場合、グループ・フィールドを選択する必要があります。


## サンプル・レポートの参照

サンプル・レポートは、指定されたレポート・タイプで作成できるレポートのサンプルです。

注: サンプル・レポートでは、Rational Build Forge データベースのデータを使用し、レポート出力を作成します。結果は、プロジェクトを既に作成し、ジョブを実行している場合のみに表示されます。

サンプル・レポートのリストを表示するには、「**クイック・レポート**」をクリックします。

サンプル・レポートのフィールドとフォーマット・オプションを表示するには、レポートの「**編集**」アイコン  を選択してください。

サンプル・レポートをコピーするには、レポートの「**編集**」アイコン  を選択し、「**レポートのコピー**」をクリックしてください。

サンプル・レポートを実行するには、該当のレポート名を選択します。

サンプル・レポートの定義は、以下のテーブルに示されています。

サンプル・レポート名	説明
SampleAnalytic-StepDuration	説明: ビルドのステップの詳細に関して報告するテーブルを作成する  レポート・タイプ: 分析  レポート・フォーマット: テーブル  レポート・フィールド: ビルド・タグ、プロジェクト名、ステップ名、ステップ期間  レポート・オプション: オプションのグループ化またはソートなし
SampleBuild-BuildsByState	説明: プロジェクトごとにビルドおよびビルド状態をリストするテーブルを作成する  レポート・タイプ: ビルド  レポート・フォーマット: テーブル  レポート・フィールド: ビルド・タグ、プロジェクト名、ビルド・カウント、ビルド状態  レポート・オプション: プロジェクト名によってグループ化
SampleCapacity- RuntimeByProject	説明: プロジェクトごとにジョブのビルド平均期間を表示する棒グラフを作成する  レポート・タイプ: 容量  レポート・フォーマット: 棒グラフ  レポート・フィールド: ビルド平均期間 (Y 軸) およびプロジェクト名 (X 軸)  レポート・オプション: プロジェクト名によってグループ化



サンプル・レポート名	説明
SampleProject-TotalsByProject	<p>説明: プロジェクトごとに成功または失敗したステップ数を報告するテーブルを作成する</p> <p>レポート・タイプ: プロジェクト</p> <p>レポート・フォーマット: テーブル</p> <p>レポート・フィールド: プロジェクト名、成功したステップ・カウント、失敗したステップ・カウント、ステップ・カウント</p> <p>レポート・オプション: プロジェクト名によってグループ化</p>
SampleQuality-PercentSuccess	<p>説明: プロジェクトごとに成功または失敗したジョブのパーセンテージを報告するテーブルを作成する</p> <p>レポート・タイプ: 品質</p> <p>レポート・フォーマット: テーブル</p> <p>レポート・フィールド: プロジェクト名、ビルド・カウント、失敗したビルドのパーセンテージ、成功したビルドのパーセンテージ</p> <p>レポート・オプション: プロジェクト名によってグループ化</p>
SampleResource-DurationOverTime	<p>説明: 各ビルドのビルド開始時刻とビルド期間を表示するグラフを作成する</p> <p>レポート・タイプ: リソース</p> <p>レポート・フォーマット: テーブル</p> <p>レポート・フィールド: ビルド期間 (Y 軸) およびビルド開始時刻 (X 軸)</p> <p>レポート・オプション: 昇順ソート順序</p>

## 用意されたレポート・タイプを使用したレポート作成

使用したいレポート・タイプ、レポート・フォーマット、およびレポート・フィールドを選択するのみで、独自のレポートを作成できます。

### 手順

1. 「レポート」タブを選択して、「クイック・レポート」を選択します。
2. 「レポート名」に固有の名前を入力します。レポート名は必須です。このレポート名は、データベースにレポートを保存するために使用され、固有の名前でなければなりません。
3. 「レポートのタイトル」に、レポートの分かりやすいタイトルを入力します。レポートのタイトルは必須です。タイトルは、レポートの先頭に表示されます。
4. 「可視性」で「公開」または「プライベート」を選択します。 プライベート・レポートは共用できません。公開レポートは、必要なアクセス権限を持つユーザ

一間で共用できます。詳しくは、499 ページの『クイック・レポートのためのレポート・グループ権限』を参照してください。

5. 「**レポート・フォーマット**」で、表または棒グラフ、折れ線グラフ、または円グラフのいずれかの形式を選択します。レポート・フォーマットの詳細については、506 ページの『クイック・レポートのためのレポート・フォーマットおよび表示の参照』を参照してください。
6. 「**レポート・タイプ**」でレポート・タイプを選択します。レポート・タイプにより、レポートに組み込まれる内容が決まります。詳細および例については、500 ページの『クイック・レポートのためのレポート・タイプ参照』を参照してください。
  - 表またはグラフ形式のレポートで表示するレポート・フィールドを選択します。
  - レポート表示を制御するためのグループ・オプションおよびソート・オプションを選択します。

**重要:** BOM レポート・タイプを選択した場合には、「プロジェクト」ボックスが表示されます。特定のプロジェクトに関する BOM レポートを作成するためには、プロジェクトを選択してレポート・フィールドの完全なリストを表示してから、レポートに含めるフィールドを選択してください。

7. (BOM レポート・タイプの場合のみ) 「**プロジェクト**」で、以下のオプションのうちの 1 つを選択します。
  - 標準の BOM レポート・フィールドのみを表示するには、「すべてのプロジェクト」を選択します。（「すべてのプロジェクト」はデフォルト設定です。）

レポート出力には、データベース内のすべてのプロジェクトに関する BOM 情報が含まれます。

  - 標準の BOM レポート・フィールドのほかにプロジェクト固有の BOM フィールドを表示するには、該当のプロジェクトを 1 つ選択します。

レポート出力には、単一のプロジェクトのみに関する BOM 情報が含まれます。

8. 「**レポートの保存**」をクリックして選択内容を保存します。そのレポートがレポート・リストに表示されます。

## ジョブ BOM へのレポート出力の追加

オプションで、`.report` コマンドを使用してレポート結果をジョブ BOM に追加します。

### 手順

1. クイック・レポート・ツールを使用してレポートを作成します。513 ページの『用意されたレポート・タイプを使用したレポート作成』を参照してください。

**注:** この機能は、BOM レポート・タイプまたはプライベート・レポートではサポートされません。

2. `.report` コマンドを使用してレポートをジョブ BOM に追加します。453 ページの『`.report`』の参照情報を参照してください。

## クイック・レポートでのレポートの変更および管理

クイック・レポートでは、レポートを実行し、レポート結果を表示し、レポート設計を編集することができます。

レポートを編集してレポート設計を変更することができます。フィルター機能はレポート定義の一部であり、任意のレポートに追加して出力をフィルターに掛けることができます。

### レポートの実行

クイック・レポートで作成されるレポートを実行するためには、レポート名をクリックします。

### このタスクについて

レポート結果はクイック・レポートの結果ビューに表示されます。

### タスクの結果

レポート・リストに戻るには、Web ブラウザーの「戻る」矢印をクリックします。

### レポートの編集

クイック・レポートで作成されるレポートを編集するためには、レポート名の隣にある「編集」アイコンをクリックします。

### このタスクについて

レポートに関する選択項目が「レポートの詳細」に表示されます。変更を行った後で、「レポートの保存」をクリックして変更内容を保存します。

### レポートのコピー

サンプル・レポートまたはその他のレポートをコピーすると、そのレポート・フィールドおよびフォーマット・オプションが新規レポート設計にコピーされ、そのレポートに対して固有の名前が割り当てられます。

### 始める前に

レポート名に `Copy` を追加することにより固有のレポート名 (例えば、`<Report_Name> Copy`) が作成されます。

あるレポートのコピーを複数作成した場合、`<Report_Name> Copy <Copy_Number>` という構文に従って番号が付加されます。(例えば、`<Report_Name> Copy 2。`)

### 手順

1. コピー対象レポートの「編集」アイコンをクリックします。
2. 「レポートのコピー」をクリックします。

コピーされたレポートがレポート・リストに表示されます。

### レポート出力のフィルターの作成

レポート・フィルターを作成して、レポート出力に表示する情報を制御することができます。

## 始める前に

フィルターを作成する前に、以下の要件と制約事項を検討してください。

- 最初にレポートを作成します。フィルター操作では、レポート定義を使用してフィルター・オプションを指定します。
- レポート・フィルターは、レポート定義の一部として保存され、単一のレポートのみに適用されます。
- レポート・フィルターを作成した後で、フィルターを変更または削除するまで、そのレポートを実行するたびにフィルターが適用されます。
- BOM レポート・タイプの場合、プロジェクトごとにフィルター操作したり、ビルドごとにフィルター操作したり、特定のレポート・フィールドのフィルター基準を指定したりできます。
- BOM 以外のレポート・タイプの場合、特定のレポート・フィールドのフィルター基準を指定する必要があります。

## 手順

1. レポート・リストからレポートを選択し、レポートの「編集」アイコンを選択します。
2. (BOM レポート・タイプの場合のみに必要) 「プロジェクト」で、以下のオプションのうちの 1 つを選択します。
  - 標準の BOM レポート・フィールドのみを表示するには、「すべてのプロジェクト」を選択します。（「すべてのプロジェクト」はデフォルト設定です。）

レポート出力には、データベース内のすべてのプロジェクトに関する BOM 情報が含まれます。

- 標準の BOM レポート・フィールドのほかにプロジェクト固有の BOM フィールドを表示するには、該当のプロジェクトを 1 つ選択します。

レポート出力には、単一のプロジェクトのみに関する BOM 情報が含まれます。

3. 「フィルター」タブを選択します。
4. (BOM レポート・タイプの場合のみのオプション) 「ビルド・フィルターの表示 (Show Build Filter)」をクリックし、以下のオプションから選択を行って、レポートに含めるビルド情報を指定します。
  - 現在および将来のすべてのビルドに関するデータを含める場合には、選択を行いません。
  - すべての現行ビルドに関するデータのみを含める場合には、「すべての現行ビルドのみ (All current builds only)」をクリックします。
  - 選択されたビルドのみの現在および将来のすべてのビルドに関するデータを含める場合には、リストからビルド・タグを選択します。
5. (その他のすべてのレポート・タイプで必須) 「フィルターの追加」をクリックして、フィルターとして使用するために「レポートの詳細」タブで選択したレポート・タイプからレポート・フィールドを選択します。

フィルター式を作成するには、次のようにします。

- a. 「フィルタリングするフィールド」で、レポート・データをフィルターに掛けるために使用するレポート・フィールドを選択します。
  - b. 「フィルタリングする演算子」で関係演算子を選択します。
  - c. 「フィルタリングする値」にレポート・フィールド値を入力します。
6. 「レポートの保存」をクリックしてレポート・フィルターの選択内容を保存します。

## クイック・レポートのよくある問題のトラブルシューティング

Rational Software Analyzer の使用の際に問題が生じた場合は、このトピックの情報を参照し、適用可能な回避策または解決方法があるかどうかを確認します。

### ポートの競合

クイック・レポートは、インストール時に指定したアプリケーション Web サーバーを使用してレポートを表示します。 ポートの競合が生じた場合、割り当てられていないポートをアプリケーション Web サーバー用に構成する必要があります。



---

## 第 23 章 ユーティリティーの処理

このトピックでは、Rational Build Forge で提供されるコマンド行ユーティリティーのセットアップと実行の方法について説明します。

---

### コマンド行ユーティリティーのアクセスと実行

コマンド行ユーティリティーは、Build Forge のインストール・ディレクトリー (Windows の場合は *<bfinstall>*、UNIX および Linux の場合は *<bfinstall>/Platform*) にあります。

ご使用の環境変数 (またはシステム・レベルの環境変数) をエンジン・レベルのシェル・コマンドが機能するように適切に構成する必要があります。例えば、Oracle および UNIX では、ORACLE\_HOME、TNS\_ADMIN、および LD\_LIBRARY\_PATH は、手動で設定する必要があります。設定しなければ、ユーティリティーは実行されません。

bfexport や bfimport などのコマンド行ユーティリティーを使用する際、データベースにアクセスするには、コマンドで buildforge.conf ファイルを検出できる必要があります。そのため、インストール・ディレクトリーからコマンドを実行するか、または、環境変数 BF\_CONFIG\_FILE を buildforge.conf ファイルの絶対パスに設定する必要があります。

---

### 「非表示の割り当て」のマイグレーション

**bfassignhiddenmigration** 参照コマンドを使用すると、暗号化のために、非表示の割り当てのマイグレーション値をまとめてマイグレーションできます。

非表示の割り当てのマイグレーション変数が Build Forge 7.x 以前のバージョンで作成されている場合は、Build Forge ユーザー・インターフェースの「環境変数」ページで、その変数を編集して保存し直し、暗号化できます。バッチ・モードで暗号化を行うには、**bfassignhiddenmigration** ユーティリティーを使用してください。

### bfassignhiddenmigration リファレンス

このユーティリティーは、以前の「非表示の割り当て」マイグレーション値をバッチ・モードで暗号化する場合に使用します。

#### 構文

```
bfassignhiddenmigration  
connection-options command-options
```

#### 制約事項および考慮事項

以下のファイルを実行できます。

- Windows システム: bfassignhiddenmigration.bat
- UNIX システムおよび Linux システム: bfassignhiddenmigration.sh



このマイグレーションには「非表示タグの編集」権限が必要です。以下の接続オプションがあります。

- `-h | --help`

ヘルプ・テキストを表示します。

- `-S | --secure C /path/to/bfclient.conf | --config=/path/to/bfclient.conf`

このテキストは、クライアント構成ファイルのロケーションを指定します。デフォルトは現行ディレクトリ内の *bfclient.conf* です。

- `-H hostname | --hostname=hostname`

接続に使用するリモート・ホスト名。デフォルト: *localhost*。

- `-P port | --port=port`

接続に使用する通信ポート。デフォルト: *3966*。

- `-d domain | --domain=domain`

認証に使用するドメインまたは領域。デフォルト: なし。

- `-u login | --user=login`

認証ユーザー名。デフォルト: *root*。

- `-p password | --password=password`

認証ユーザー・パスワード。デフォルト: *root*。

- `-E encoding | --encoding=encoding`

出力文字セットで使用するエンコード。指定されたエンコードが無効な場合は、使用可能なすべてのエンコードがリストされて、プログラムが終了します。

注: 要求されたエンコードでテキストを表示できるかどうかは、クライアントで使用されている端末ソフトウェアの機能に依存します。デフォルト設定は、ご使用のオペレーティング・システムと JVM のサポートによって異なります。

コマンド・オプション:

- `-a | --all`

すべての「非表示の割り当て」環境変数がシステムでマイグレーションされます。このオプションが許可されると、`-e` オプションおよび `-v` オプションは無視されます。

- `-e Environment1[,Environment2[,...]] | --environment=Environment1[,Environment2[,...]]`

「非表示の割り当て」変数が特定の環境でマイグレーションされます。`-a` オプションが指定されていない場合に、すべての「非表示の割り当て」変数が特定の環境でマイグレーションされます。特定の環境と突き合わせるために Java 正規表現を使用できます。

- `-v Variable1[,Variable2[,...]] | --variable=Variable1[,Variable2[,...]]`

特定の「非表示の割り当て」環境変数がマイグレーションされます。'-a' オプションが指定されていない場合に、特定の「非表示の割り当て」変数がすべてシステムでマイグレーションされます。特定の変数と突き合わせるために Java 正規表現を使用できます。

## 例

1. 次の例は、システムですべての「非表示の割り当て」環境変数をマイグレーションする例です。

Windows

```
C:¥> bfassignhiddenmigration.bat -H localhost -u root -p root -a
```

UNIX および Linux

```
$bfassignhiddenmigration.sh -H localhost -u root -p root -a
```

2. 次の例は、*MyEnv* 環境ですべての「非表示の割り当て」変数をマイグレーションする例です。

Windows

```
$ C:¥> bfassignhiddenmigration.bat -H localhost -u root -p root -e MyEnv
```

UNIX および Linux

```
$ bfassignhiddenmigration.sh -H localhost -u root -p root -e MyEnv
```

3. 次の例は、*bf8* で始まる「非表示の割り当て」変数をすべてマイグレーションする例です。

Windows

```
$C:¥> bfassignhiddenmigration.bat -H localhost -u root -p root -v bf8.*
```

UNIX および Linux

```
$ bfassignhiddenmigration.sh -H localhost -u root -p root -v bf8.*
```

4. 次の例は、*bf8* を含む「非表示の割り当て」変数、あるいは非表示環境 *MyEnv* または *Com* で終わる「非表示の割り当て」変数をすべてマイグレーションする例です。

Windows

```
$C:¥> bfassignhiddenmigration.bat -H localhost -u root -p root -e MyEnv,Com.* -v .*bf8.*.*hid
```

UNIX および Linux

```
$ bfassignhiddenmigration.sh -H localhost -u root -p root -e MyEnv,Com.* -v .*bf8.*.*hid
```

---

## プロジェクトのエクスポート

`bfexport` コマンドまたは `.export` ドット・コマンドを使用して、プロジェクトとその他の Build Forge オブジェクトを XML ファイルにエクスポートすることができます。

エクスポートしたプロジェクトは XML ファイルに保管されます。エクスポートしたプロジェクトは、Build Forge にインポートして戻すことができます。

例えば、プロジェクト構成データを保存するために、bfexport を実行するプロジェクトの最後にステップを追加することができます。このステップはプロジェクト定義のバックアップとして使用できます。また、プロジェクトを別の Build Forge インストール済み環境に移動する際に使用することもできます。

## bfexport 参照

プロジェクト・データを指定された XML ファイルにエクスポートする場合、または、プロジェクト・データをディスプレイ端末 (STDOUT) に送信して表示する場合は、bfexport コマンドを使用します。エクスポート・ファイルには、単一のプロジェクトまたはプロジェクト・スナップショットのプロジェクト構成データが含まれています。

### 構文

bfexport

bfexport [-l]

bfexport [-l] <project\_name>

bfexport [-c "<comment>"] [-f <file\_name>] [-g] [-s] [-C] [-L ] [-n ]  
<project\_name> | <project\_name> <snapshot\_name> | <project\_id>

### 使用法

共通のプロジェクト・エクスポート・タスクを完了するには、次のコマンド・オプションを使用します。

- コマンド構文を表示する場合は、オプションを付けずに bfexport を使用します。
- Build Forge データベースに保存されているプロジェクト名とプロジェクト ID をリスト表示する場合は、bfexport -l を使用します。

スナップショット名は、<project\_ID>: <project\_name> - <snapshot\_name> のように、コマンド出力のプロジェクト名に付加されます。

- プロジェクト・データを XML ファイルに送信する場合は、bfexport -f <file\_name> となります。プロジェクト・データのインポートに使用できるファイルを生成する場合は、-f <file\_name> オプションを指定する必要があります。
- z/Linux プラットフォームでは、コマンドを bfexport.pl として実行する必要があります。他のすべてのプラットフォームでは、コマンドに拡張子は必要ありません。

### 前提条件および制約事項

bfexport ユーティリティーは、Build Forge のインストール・ディレクトリーにあります。

サーバー用の権限パスワードは、エクスポート・ファイルには含まれていません。したがって、インポート後、UI でサーバー用の権限パスワードを手入力する必要があります。

bfexport コマンドで buildforge.conf ファイルを検出して、Build Forge データベースにアクセスする必要があります。buildforge.conf が置かれているディレクトリー

(Windows の場合は `<bfinstall>`、UNIX および Linux の場合は `<bfinstall>/Platform`) から `bfexport` を実行してください。

例

ファイルに出力を書き込む場合は、`<file_name>` オプションを使用します。次の例では、`helloworld` が出力ファイル名で、プロジェクト ID がプロジェクト名の代わりに使用されています。

```
bfexport -c "Saving a copy of project before making changes"
-f helloworld 675B57CC-8366-11DD-B2E0-043C04E44E1A
```

デフォルトのプロジェクト・スナップショットのみをエクスポートする場合は、`<project_name>` を使用します。

```
bfexport -f helloworld test_project
```

プロジェクトのスナップショットを 1 つだけエクスポートする場合は、`<project_name> <snapshot_name>` を使用します。

```
bfexport -f helloworld test_project snapshot_1
```

親プロジェクトのスナップショットがデフォルト・プロジェクトでない場合は、`<project_name>` の後に `parent` というキーワードを指定して、親プロジェクトのスナップショットをエクスポートする必要があります。

```
bfexport -f helloworld test_project parent
```

オプションの説明

オプション	説明
<code>&lt;project_name&gt;</code>	エクスポートするプロジェクトの名前。プロジェクト名またはプロジェクト ID が必要です。プロジェクト名にスペースが含まれている場合は、名前を引用符で囲む必要があります。  コマンド・オプションの後にプロジェクト名を指定してください。
<code>&lt;snapshot_name&gt;</code>	エクスポートするプロジェクト・スナップショットの名前。以下の構文に示されているように、プロジェクト名が必要です。  <code>&lt;project_name&gt; &lt;snapshot_name&gt;</code>  コマンド・オプションの後にプロジェクト名とスナップショット名を指定してください。プロジェクト名またはスナップショット名にスペースが含まれている場合は、名前を引用符で囲む必要があります。
<code>&lt;project_id&gt;</code>	エクスポートするプロジェクトの ID。プロジェクト ID には UUID を使用します。プロジェクト ID またはプロジェクト名が必要です。  コマンド・オプションの後にプロジェクト ID を指定してください。

オプション	説明
-f <file_name>	<p>bfexport 出力の XML ファイル名。パス名を指定しない場合は、現在の作業ディレクトリーが使用されます。ファイル名にスペースが含まれている場合は、名前を引用符で囲む必要があります。</p> <p>ファイル名を指定しない場合は、bfexport 出力が STDOUT に送信されます。</p> <p><b>注:</b> 表示のみの場合は、STDOUT を使用してください。ファイルに STDOUT をリダイレクトしないでください。リダイレクトすると、結果ファイルにロギング・メッセージが含まれてしまい、bfimport コマンドまたは UI インポート・ユーティリティーでインポート・ファイルとして使用できなくなります。</p>
parent	<p>親がデフォルトのプロジェクト・スナップショットでない場合に、親のプロジェクト・スナップショットをエクスポートするために必要なキーワード。</p> <p>以下のように、プロジェクト・スナップショット名の後に parent というキーワードを指定してください。</p> <pre>bfexport -f helloworld test_project parent</pre>
-l	データベース内のプロジェクトを、名前またはプロジェクト ID でリストします。他のオプションと一緒にこの -l オプションを使用することはできません。
-c "<comment>"	コメントを組み込みます。コメントは引用符で囲む必要があります (例えば、"my project version 50")。コメントは、<buildforge> XML エlement に追加されます。
-g	通知の受信先に指定されているアクセス・グループのメンバーであるユーザーを XML ファイルに保存します。ユーザーとユーザーのプロパティーは、<user> XML エlement にリストされています。-s が必要です。
-s	管理コンソールで定義されたサーバーを XML ファイルに保存します。サーバーとサーバーのプロパティーは、関連付けられた <auth> および <collector> の情報と共に、<server> XML エlement にリストされています。
-L	UI で定義された LDAP ドメイン・コントローラーを XML ファイルに保存します。LDAP ドメイン・コントローラーとそれらのプロパティーは、<ldap> XML エlement にリストされています。
-n	プロジェクトとステップに割り当てられた通知テンプレートを XML ファイルに保存します。通知テンプレートとそれらのプロパティーは、<mail-template> XML エlement にリストされています。
-C	プロジェクト用のサーバーに割り当てられたコレクターを XML ファイルに保存します。コレクターとコレクターのプロパティーは、<collector> XML エlement にリストされています。-s が必要です。

## トラブルシューティング

ファイルまたはライブラリーが見つからないことを報告するエラーが検出された場合は、ライブラリー・パスが正しく設定されているかを確認してください。

- `LD_LIBRARY_PATH` (Linux システムおよび UNIX システムの場合)。
- `LIBPATH` (AIX システムの場合)。

正しい設定は `<bfinstall>/rc/buildforge` です。ライブラリー・パス変数およびその他の変数がそのファイルで設定されているかを確認してください。必要であればそれらをリセットし、エクスポートしてください。

## .export の使用

`.export` を使用して、プロジェクトをプロジェクト内のステップからエクスポートすることができます。

`.export` コマンドでは、その他のオブジェクト・データのエクスポート・オプションは提供されません。他の Build Forge オブジェクトをエクスポートするには、`bfexport` コマンドを使用する必要があります。

444 ページの『`.export`』を参照してください。

---

## プロジェクトのインポート

`bfimport` コマンドまたはコンソールのインポート機能を使用して、以前にエクスポートしたプロジェクトとその他の Build Forge オブジェクトをインポートすることができます。

さまざまなオプションを使用して、インストールするオブジェクトを選択できます。

インポートしたオブジェクトにアクセス・グループを適用する方法を選択できます。オプションは、「**セキュア・アクセスによるインポート**」システム設定値を使用して設定されます。

注: Build Forge ユーザー・インターフェースによって、2M より大きいファイルのインポートが拒否される場合があります。このような場合は、`bfimport` コマンド行ツールを直接使用するか、「`upload_max_filesize`」PHP 環境変数を調整します。

## インポート・ユーティリティを使用したプロジェクトおよびその他のオブジェクトのインポート

エクスポート・ファイルに保存したプロジェクトとその他のオブジェクトのオブジェクト定義をインポートするには、ユーザー・インターフェースのインポート・ユーティリティを使用します。インポート・ユーティリティを使用すると、エクスポート・ファイルからインポートするオブジェクトを選択できます。

注: インポートするプロジェクトに通知グループがあり、その通知グループが組み込み済みのグループ (ビルド・エンジニア、開発者、ゲスト、オペレーター、システム管理者など) ではない場合、インポートする前にアクセス・グループを作成する必要があります。

1. 「管理」 > 「インポート」と選択します。

UI Config Console Reports Logout: Root User

Import Help ?

Reset Import

Project XML File:  Browse...

Select entities to process:

- ☒ Import Projects
  - ☐ Include Chained Projects
- ☒ Import Server Definitions
- ☒ Import Environment Variables
- ☒ Import Project Class
- ☐ Import Users
- ☐ Import Project Specific Templates
- ☐ Import Filters
- ☐ Import Collectors
- ☐ Import Selectors
- ☐ Import LDAP Domain Settings

When the names are the same:

- ☐ Replace Entities
- ☒ Rename Entities

Reset Import

2. プロジェクト用のエクスポート XML ファイルを見つけるには、「参照」をクリックします。

注: エクスポート・ファイルは、**bfexport** コマンドまたは **.export** ドット・コマンドを使用して作成する必要があります。

3. エクスポート XML ファイルから、ユーザー・インターフェースにインポートするプロジェクトとその他のオブジェクトを選択します。



注: サーバー・オブジェクトをインポートする場合は、サーバー用の権限パスワードをインポートしてから、ユーザー・インターフェースで手入力する必要があります。**bfexport** コマンドと **.export** コマンドでは、サーバー用の権限パスワードはエクスポート・ファイルに保存されません。

4. インポート・ユーティリティを使用して、同じ名前のオブジェクトの置換、上書き、または名前変更のいずれを行うかを指定するには、「**エンティティの置換**」または「**エンティティの名前変更**」(デフォルト・オプション) を選択します。

重要: 名前変更および置換の方法については、532 ページの『インポート時のオブジェクトの名前変更および置換』を参照してください。

## bfimport 参照

bfimport ユーティリティを使用して、以前に XML ファイルにエクスポートしたプロジェクトおよびその他のオブジェクトの定義を、ユーザー・インターフェースにインポートします。 インポート・ユーティリティを使用して、XML ファイルから選択したオブジェクトをインポートすることもできます。

このトピックでは、**bfimport** コマンドの構文について説明し、その詳しい使用法を示します。

### 構文

bfimport

bfimport [-L] <file\_name>

bfimport [-p -I -s -S -e -c -C -u -T -f -d -r ] <file\_name.xml>

bfimport [-L | [-p -I -s -S -e -c -C -u -T -f -d -r ]] <file\_name.xml>

### 制約事項および考慮事項

サーバー用の権限パスワードは、エクスポート・ファイルには含まれていません。したがって、サーバー用の権限パスワードを手入力する必要があります。

インポート対象のオブジェクトと同じ名前のオブジェクトが存在する場合、デフォルトでは、インポート時にデータベース・オブジェクトを上書きしないように、インポート対象のオブジェクトの名前が変更されます。 また、同じ名前のオブジェクトが存在する場合、-r オプションを指定して、オブジェクトを置き換えることもできます。 スナップショット・オブジェクトの場合、オブジェクトは名前ではなく UUID で比較されます。

デフォルトでは、オブジェクトの名前は bfimport を使用して変更され、以下の命名規則が使用されます。

<object\_name>\_IMPORT\_<number>

名前変更の場合、スナップショット・オブジェクトのスナップショット名が変更され、スナップショット・オブジェクトが親スナップショットの子であっても、新規の基本スナップショットまたは親レベルのスナップショットとしてインポートされます。

名前変更について詳しくは、532 ページの『インポート時のオブジェクトの名前変更および置換』を参照してください。

オブジェクトを置き換えるには、`-r` オプションを指定する必要があります。置き換えオプションにより、既存のオブジェクトが上書きされます。7.0.2 以前のエクスポート・ファイルでは、スナップショット・オブジェクトは置き換えられません。代わりに、`<object_name>_IMPORT_<number>` の規則を使用して名前が変更されます。7.1 エクスポート・ファイルの場合、スナップショット・オブジェクトは、その UUID が一致する場合に置き換えられます。

## 前提条件

`bfimport` ユーティリティーを使用する前に、以下の前提条件を満たしていることを確認してください。

- `bfexport` コマンドまたは `.export` ドット・コマンドで作成されたエクスポート XML ファイル。
- `bfimport` ユーティリティーは Build Forge のインストール・ディレクトリにあります。
- `bfimport` コマンドで `buildforge.conf` ファイルを検出して、Build Forge データベースにアクセスする必要があります。 `buildforge.conf` が置かれているディレクトリ (Windows の場合は `<bfinstall>`、UNIX および Linux の場合は `<bfinstall>/Platform`) から `bfimport` を実行してください。

注: インポートするプロジェクトに通知グループがあり、その通知グループが組み込み済みのグループ (ビルド・エンジニア、開発者、ゲスト、オペレーター、システム管理者など) ではない場合、インポートする前にアクセス・グループを作成する必要があります。

## 使用法

共通のインポート・タスクを完了するには、以下のコマンド・オプションを使用します。

- コマンド構文を表示する場合は、オプションを付けずに **`bfimport`** を使用します。
- XML ファイル内の Build Forge オブジェクトとその名前の要約リストを表示する場合は、`bfimport -L <file_name.xml>` を使用します。
- z/Linux プラットフォームでは、コマンドを **`bfexport.pl`** として実行します。他のすべてのプラットフォームでは、コマンドに拡張子は必要ありません。
- オプションを指定しないと、オブジェクトはインポートされません。個々のオブジェクトをインポートする場合は、オプションを指定する必要があります。
- 存在しないオブジェクトにオプションを指定すると、インポート・ユーティリティーによって XML ファイル内にないオブジェクトがスキップされ、ファイル内にあるオブジェクトがインポートされます。

## 例

XML ファイル内の Build Forge® オブジェクトをリスト表示するには、`-L` オプションと XML ファイル名のみを指定します。以下の例に、コマンド出力の一部を示します。

```
C:\Program Files\IBM\Build Forge>bfimport -L samples\projects\basic.xml
10/07/2008 5:31:55 PM: Import: 7624: CRRBF200811: Importing export
file from a 7.0.10025 version console.
Project : [Basic Sample]
Tag Variable : [MAJ]
Tag Variable : [MIN]
Step : [Checkout Source]
Step : [Update Applet Version]
Step : [Create Baseline]
Environment : [Basic Environment]
Class : [Production]
Filter : []
Selector : [Web Server]
Selector : [Local Server]
```

XML ファイルにあるすべてのオブジェクトをインポートするには、以下の例のように、オブジェクトをインポートするためのオプションを指定します。 オブジェクトの名前はインポート時に変更されます。 インポートに成功した場合は、成功ステートメントが表示されます。

```
C:\Program Files\IBM\Build Forge>bfimport -p -I -s -S -e -c -C -u -T -d -f
"samples\projects\basic.xml"
```

## オプションの説明

オプション	説明
<file_name.xml>	<p>インポートする Build Forge<sup>®</sup> オブジェクトが含まれるエクスポート XML ファイルの名前。XML ファイルは、<b>bfexport</b> コマンドまたは <b>.export</b> ドット・コマンドを使用して作成する必要があります。この XML ファイル名は必須です。XML ファイルが現行ディレクトリー (<b>bfexport</b> コマンドの発行元ディレクトリー) にない場合は、パス名を指定してください。</p> <p>ファイル名にスペースが含まれている場合は、名前を引用符で囲む必要があります。</p>
-L	<p>エクスポート XML ファイルにあるオブジェクトとそのオブジェクト名をリスト表示します。 このオプションは単独で使用します。他の <b>bfimport</b> オプションと一緒に指定しないでください。 -L オプションからの出力は <b>STDOUT</b> に送信することも、XML またはテキスト・ファイルにリダイレクトすることもできます。</p>
-p	<p>XML ファイルからプロジェクトの構成データをインポートします。 プロジェクトの構成データには、ステップおよびプロジェクトの定義データ (タグ変数を含む) が含まれています。</p> <p>名前変更時に、プロジェクト名は &lt;project_name&gt;_IMPORT_&lt;number&gt; として UI にインポートされます。</p>
-I	<p>プロジェクトまたはステップ・レベルで参照されるチェーン・プロジェクトまたはライブラリーをインポートします。</p> <p>名前変更時に、チェーン・プロジェクトまたはライブラリーは &lt;project_or_library_name&gt;_IMPORT_&lt;number&gt; として UI にインポートされます。</p>

オプション	説明
-S	<p>UI で定義されているセレクター・オブジェクトをインポートします。</p> <p>名前変更時に、セレクターは <code>&lt;selector_name&gt;_IMPORT_&lt;number&gt;</code> として UI にインポートされます。</p>
-s	<p>-s オプションを <code>bfexport</code> に指定した場合は、管理コンソールで定義されているサーバー・オブジェクトをインポートします。</p> <p>名前変更時に、サーバーは <code>&lt;server_name&gt;_IMPORT_&lt;number&gt;</code> として UI にインポートされます。</p> <p>サーバー用の権限パスワードは、エクスポート XML ファイルには含まれていません。したがって、サーバー用の権限パスワードを手入力する必要があります。</p>
-e	<p>プロジェクトまたはステップ・レベルで参照される環境とその変数をインポートします。</p> <p>名前変更時に、環境は <code>&lt;environment_name&gt;_IMPORT_&lt;number&gt;</code> としてインポートされます。</p>
-c	<p>プロジェクトで参照されるクラスをインポートします。</p> <p>名前変更時に、クラスは <code>&lt;class_name&gt;_IMPORT_&lt;number&gt;</code> としてインポートされます。</p>
-C	<p>-C オプションを <code>bfexport</code> に指定した場合は、プロジェクトのサーバーに割り当てられているコレクターをインポートします。</p> <p>名前変更時に、コレクターは <code>&lt;collector_name&gt;_IMPORT_&lt;number&gt;</code> としてインポートされます。</p>
-u	<p>-g オプションを <code>bfexport</code> に指定した場合は、E メール通知の受信先に指定されているアクセス・グループのメンバーであるユーザーをインポートします。</p> <p>名前変更時に、ユーザーの情報は <code>&lt;users&gt;_IMPORT_&lt;number&gt;</code> としてインポートされます。</p>
-T	<p>-n オプションを <code>bfexport</code> に指定した場合は、プロジェクトとステップに割り当てられている、ユーザーによって作成された通知テンプレートをインポートします。</p> <p>名前変更時に、通知テンプレートは <code>&lt;template_name&gt;_IMPORT_&lt;number&gt;</code> としてインポートされます。</p>
-f	<p>-n オプションを <code>bfexport</code> に指定した場合は、プロジェクト・ステップに割り当てられているログ・フィルターをインポートします。</p> <p>名前変更時に、ログ・フィルターは <code>&lt;filter_name&gt;_IMPORT_&lt;number&gt;</code> としてインポートされます。</p>

オプション	説明
-d	<p>-L オプションを <code>bfexport</code> に指定した場合は、ユーザー・インターフェースで定義されている LDAP ドメイン・コントローラーをインポートします。</p> <p>名前変更時に、ログ・フィルタは  <code>&lt;LDAP_domain_controller&gt;_IMPORT_&lt;number&gt;</code> としてインポートされます。</p>
-r	<p>インポート対象オブジェクトの名前を変更せずに、オブジェクトを置き換えます。デフォルトでは、インポート対象オブジェクトの名前は変更され、以下の命名規則が使用されます。</p> <p><code>&lt;object_name&gt;_IMPORT_&lt;number&gt;</code></p> <p>7.1 オブジェクトでは、置き換えオプションを指定した場合、<b>bfimport</b> コマンドによって 7.1 オブジェクトのユーザー・インターフェースにあるオブジェクトが上書きされます。</p> <p>7.0.2 以前のオブジェクトでは、スナップショット・オブジェクトは置き換えられません。以下の命名規則を使用して、名前が変更されます。</p> <p><code>&lt;object_name&gt;_IMPORT_&lt;number&gt;</code></p> <p><b>重要:</b> スナップショット・オブジェクトは、名前ではなく UUID で、置き換えられるかどうかが評価されます。</p> <p>詳しくは、532 ページの『インポート時のオブジェクトの名前変更および置換』を参照してください。</p>

## インポート・オブジェクトへのアクセス・グループの割り当て方法

インポート・オブジェクトへのアクセス・グループの割り当て方法は、「セキュア・アクセスによるインポート」システム設定によって制御されます。

- ・「セキュア・アクセスによるインポート」が「はい」(デフォルト) に設定されている場合は、「デフォルトのセキュア・アクセス・グループのインポート」設定でアクセス・グループが指定されます。デフォルト値は「ビルド・エンジニア」です。
- ・「セキュア・アクセスによるインポート」が「いいえ」に設定されている場合は、「セキュアでないデフォルトのアクセス・グループのインポート」設定でアクセス・グループが指定されます。デフォルトは「開発者」です。

「セキュアでないデフォルトのアクセス・グループのインポート」のアクセス・グループが指定されていない場合は、直前に作成されたアクセス・グループが使用されます。アクセス・グループが作成されていない場合は、「デフォルト」アクセス・グループが使用されます。

## アクセス・グループの割り当ておよびセキュリティー

セキュリティー対策として、Build Forge ではアクセス・グループの直接エクスポートまたは直接インポートを許可していません。許可した場合には、コンソール・ホストへのアクセス権限を持つユーザーが、エクスポートされたオブジェクトの XML ファイルを直接操作して、それらをインポートすることができます。インポート時

にアクセス・グループを割り当てることにより、許可されたユーザーのみがアクセス・グループを割り当てることが保証されます。

## インポート時のオブジェクトの名前変更および置換

同じ名前のオブジェクトがデータベース内に既にある場合は、bfimport コマンドとインポート・ユーティリティーを使用して、インポート済みオブジェクトの名前を変更します。インポート時のオブジェクトの名前変更は、デフォルト動作です。

この動作を変更して、インポート時に既存のオブジェクトを置換するには、-r オプションを bfimport に指定するか、UI で「エンティティの置換」オプションを選択する必要があります。

以下のトピックでは、インポート済みオブジェクトの名前変更および置換を行う際に、bfimport コマンドとインポート・ユーティリティーで使用する命名規則について説明します。

スナップショット・オブジェクト (プロジェクト、セレクトーおよび環境変数) のスナップショット名はそのまま使用されます (名前がある場合)。それ以外の場合は、インポート時にデフォルトのスナップショット名が割り当てられます。

### 7.1 以前のエクスポート・ファイルにあるオブジェクトの名前変更

7.1 以前のエクスポート・ファイルにあるオブジェクトについては、bfimport コマンドとインポート・ユーティリティーで以下の命名規則を使用して、UI のオブジェクトの名前を変更します。

オブジェクトの状況	UI オブジェクト名	UI スナップショット名 (スナップショット・オブジェクトにのみ適用)
新規 (データベースにない)	<new_object_name>	<snapshot_name>
既存 (データベースにある)	<existing_object_name>_IMPORT_<n>	<snapshot_name>   Base Snapshot

スナップショット・オブジェクトは、それが親スナップショットの子であった場合でも、新規の親レベル・スナップショットとしてインポートされます。スナップショット・オブジェクトとして使用できるのは、プロジェクト、セレクトーおよび環境変数のみです。スナップショット名がある場合は、そのまま使用されます。それ以外の場合は、下の表に示すように、デフォルトの基本スナップショット名が割り当てられます。

### 7.1 以前のエクスポート・ファイルにあるオブジェクトの置換

7.1 以前のエクスポート・ファイルにあるオブジェクトについては、bfimport コマンドとインポート・ユーティリティーで以下の命名規則を使用して、UI のオブジェクトを置換します。スナップショット・オブジェクトは、名前ではなく UUID で、置き換えられるかどうかが評価されることに注意してください。

オブジェクトの状況	UI オブジェクト名	UI スナップショット名 (スナップショット・オブジェクトにのみ適用)
新規 (データベースにない)	<new_object_name>	基本スナップショット
既存 (データベースにある) スナップショット以外のオブジェクト	<existing_object_name	該当なし
既存 (データベースにある) スナップショット・オブジェクト	<existing_object_name>_IMPORT_<n>	基本スナップショット

スナップショット・オブジェクトの場合、そのスナップショット・オブジェクトは、親スナップショットの子であっても、新規の親レベル・スナップショットとしてインポートされます。スナップショット・オブジェクトとして使用できるのは、プロジェクト、セレクトーおよび環境変数のみです。 7.1 以前のエクスポート・ファイルには、スナップショット名が付けられているオブジェクトを含めることはできません。そのため、下の表で示すように、デフォルトの基本スナップショット名が割り当てられます。

## 7.1 エクスポート・ファイルにあるオブジェクトの置換

7.1 エクスポート・ファイルにあるオブジェクトについては、bfimport コマンドとインポート・ユーティリティーで以下の命名規則を使用して、UI のオブジェクトを置換します。

オブジェクトの状況	UI オブジェクト名	UI スナップショット名 (スナップショット・オブジェクトにのみ適用)
新規 (データベースにない)	<new_object_name>	<snapshot_name>   Base Snapshot
既存 (データベースにある)	<new_object_name>	<snapshot_name>   Base Snapshot

スナップショット・オブジェクトは、名前ではなく UUID で、置き換えられるかどうか評価されます。スナップショット・オブジェクトは、それが親スナップショットの子であった場合でも、新規の親レベル・スナップショットとしてインポートされます。スナップショット・オブジェクトとして使用できるのは、プロジェクト、セレクトーおよび環境変数のみです。 7.1 オブジェクトにはすべて固有のスナップショット名が付けられているか、デフォルトの基本スナップショット名が使用されます。



## スナップショットのインポートおよび置換のシナリオ

スナップショット・オブジェクトは、名前ではなく UUID で、置き換えられるかどうか評価されます。スナップショット・オブジェクトを何度もインポートするときには注意が必要です。

この動作について、以下のシナリオで説明します。

1. あるスナップショット・オブジェクトがシステム A からエクスポートされ、空のシステム B にインポートされます。システム A にインポートされているオブジェクトは、システム B での UUID と同じ UUID を持っています。
2. スナップショット・オブジェクト属性 (名前など) がシステム A で変更されます。
3. このスナップショット・オブジェクトが、もう一度システム A からエクスポートされ、置換オプション付きでシステム B にインポートされます。UUID が一致しているため、スナップショット・オブジェクトは置き換えられます。システム B で置き換えられたオブジェクトは、システム A で変更されたオブジェクト属性 (変更済みの名前など) すべてを引き継いでいます。

---

## レガシー Perl ジョブ・コントローラー・プロジェクトのマイグレーション

Java マスター・ジョブ・コントローラーを使用して、Build Forge 7.x 以前のリリースで作成されたレガシー Perl ジョブ・コントローラー・プロジェクトを実行するには、Perl プロジェクトを Java MJC プロジェクトとしてコピーする必要があります。

ユーザー・インターフェースでは、単一のプロジェクトまたはライブラリーをコピーすることができます。また、bfperlprojectmigration ユーティリティを使用して、チェーン関係を含むすべての Perl プロジェクトおよびライブラリーを、Java MJC プロジェクトおよびライブラリーに一度に複製することができます。

### Bfperlprojectmigration ユーティリティのリファレンス

bfperlprojectmigration ユーティリティを使用して、チェーン関係を含むすべての Perl プロジェクトおよびライブラリーを、Java MJC プロジェクトおよびライブラリーに一度に複製することができます。

#### 使用法

```
bfperlprojectmigration.sh [CONNECTION OPTIONS] [COMMAND OPTIONS]
```

#### 接続オプション

**-h**

**--help** このヘルプ・テキストを表示します。

**-S**

**--secure**

サーバーへのセキュアな SSL/TLS 接続を要求します。

デフォルト: 通常の TCP 接続 (非セキュア) が使用されます。

**-C /path/to/bfclient.conf**

**--config=/path/to/bfclient.conf**

使用するクライアント構成ファイルの場所をオーバーライドします。

デフォルト: bfclient.conf (現行ディレクトリー内)。

**-H Hostname**

**--hostname=Hostname**

接続のリモート・ホスト名を設定します。

デフォルト: localhost

**-P Port**

**--port=Port**

接続の通信ポートを設定します。

デフォルト: 3966

**-d Domain**

**--domain=Domain**

認証のドメイン (領域) を設定します。

デフォルト: なし

**-u Login**

**--user=Login**

認証のユーザー・ログインを設定します。

デフォルト: root

**-p Password**

**--password=Password**

認証のユーザー・パスワードを設定します。

デフォルト: root

**-E Encoding**

**--encoding=Encoding**

出力の文字セット・エンコードを設定します。無効なエンコードが指定された場合、使用可能なすべてのエンコードがリストされて、プログラムが終了します。

注: 要求されたエンコードでテキストを表示できるかどうかは、クライアントで使用されている端末ソフトウェアの機能に依存します。

デフォルト: UTF-8

## コマンド・オプション

**-a**

**--all**

システム内のすべてのレガシー Perl プロジェクトおよびライブラリーを、MJC プロジェクトおよびライブラリーとして複製します。このオプションを指定すると、"-o" は無視されます。

**-s NameSuffix**

**--suffix=NameSuffix**

複製された MJC プロジェクトまたはライブラリーの名前のサフィックス。  
これを指定しない場合、デフォルトでストリング "Copy" が名前のサフィックスとして使用されます。

デフォルト: TimeStamp

注: 同じサフィックスが複数回使用されると、システムは名前の最後に索引番号を付加して、重複を防ぎます。

**-o project1Name[, "project2 Name", projectUUID]**

**--one=project1Name[, "project2 Name", projectUUID]**

現在サポートされていません。指定された Perl プロジェクトまたはライブラリーを、チェーニングされたプロジェクトおよびライブラリーを含む MJC プロジェクトまたはライブラリーに再帰的に複製します。

例:

- UNIX/Linux: bfperlprojectmigration.sh -H localhost -u root -p root -a -s MJC
- Windows: bfperlprojectmigration.bat -H localhost -u root -p root -a -s MJC

システム内のすべての Perl プロジェクトおよびライブラリー (その間の関係も含む) を、origName\_MJC などの新しい名前を使用して、MJC プロジェクトおよびライブラリーとして複製します。

## 第 24 章 「UI 構成」タブの Web リソースへのリンク

Build Forge UI にタブを追加するには、「UI 構成」タブを使用します。

「UI 構成」タブを使用して、Build Forge ユーザー・インターフェースにタブを追加することができます。新しいタブには、それぞれ URL が含まれます。タブを使用することで、アプリケーション、オペレーティング・システム、サーバー、またはユーザーに関する情報などの外部リソースにリンクできます。

注: Build Forge 内部の URL にリンクするタブを作成しないでください。

タブを追加するには、以下を行います。

1. 「UI 構成」タブを選択します。
2. 「タブの追加」をクリックします。
3. 「名前」に、タブのタイトルまたは名前を入力します。
4. 「リンク」に、次のいずれかを入力できます。

オプション	使用法	例
プロトコルを指定した URL	<ul style="list-style-type: none"><li>• プロトコルが http でない場合は必須です。</li><li>• プロトコルが http の場合はオプションです。</li></ul>	http://www.ibm.com
プロトコルを指定しない URL	デフォルトの http プロトコルを使用します。	www.ibm.com

5. 「ターゲット」で、以下のオプションの 1 つを選択します。
  - 内部 (デフォルト): 既存のブラウザ・ウィンドウでリンクを開く場合に選択します。
  - 外部: 新しいブラウザ・ウィンドウでリンクを開く場合に選択します。
6. 「有効」で、以下のオプションの 1 つを選択します。
  - 有効 (デフォルト): タブを有効にする場合に選択します。有効にした場合、そのタブを選択すると URL への接続が行われます。
  - 無効: タブを無効にする場合に選択します。無効にされたタブは非表示になります。
7. 「可視」で、以下のオプションの 1 つを選択します。
  - True (デフォルト): UI にタブを表示する場合に選択します。
  - False: UI にタブを表示しない場合に選択します。
8. 「ルートのみ」で、以下のオプションの 1 つを選択します。
  - True (デフォルト): root ユーザーまたは root アクセス権を持つ Build Forge ユーザーのみが、タブを選択して URL リンクに接続できるようにします。
  - False: すべての Build Forge ユーザーが、タブを選択して URL リンクに接続できるようにします。

注: 新しいタブを保存した後、タブのタイトルがリストに表示されます。タブそのものが表示されない場合は、ページを最新表示してください。

---

## 第 25 章 Build Catalyst

Build Catalyst を使用すると Make ベースの C および C++ ソフトウェア・ビルドを高速化することができます。Build Catalyst は既存の Make ファイルを解釈および分析し、高速ビルドを実行します。高速ビルドでは、並列ビルドや分散ビルドなどの技法を使用して全体的なビルド時間を短縮できます。

Build Catalyst は IBM® Rational® Build Forge® で提供されますが、追加インストールが必要です。Build Catalyst を Rational Build Forge で使用するには、Rational Build Forge エージェントがインストールされているホストに Build Catalyst をインストールしてください。Build Catalyst をインストールして構成すると、Rational Build Forge のステップで Build Catalyst の高速ビルドを呼び出すことができます。Rational Build Forge プロジェクトと統合することにより、Rational Build Forge が提供する広範なビルド自動化ツールにアクセスすることができます。例えば、リファクタリングされた Make ベースのビルドの呼び出しを、他のビルドに関連したより大きなビルド・シナリオの一部に組み込むことができます。Rational Build Forge プロジェクトで、ビルド前のプロシーチャーとビルド後のプロシーチャーを実行することもできます。

Build Catalyst は、GNU Make 3.80 用に作成された Make ファイルとの互換性が高くなっています。GNU Make 3.80 用に作成された Make ファイルがある場合、Build Catalyst を使用することにより、多くの場合にその Make ファイルを変更することなくソース・コードをビルドできます。ただし、小規模な変更が必要な場合があります。Build Catalyst は以下のビルドをサポートします。

- **並列ビルド:** Build Catalyst は非依存ターゲットを識別して、それらを並列してビルドできます。並行してビルドを行うと、ビルド・コンピューターでのリソース (マルチコア・プロセッサなど) の使用効率が高くなります。リソースを効率的に使用することにより、Build Catalyst はビルドを高速化してビルド時間を短縮できます。
- **分散ビルド:** 分散ビルドは並列ビルドと類似していますが、ビルド・アクティビティが複数のコンピューター間で分散されます。

Build Catalyst 資料は、ソフトウェア・ビルドの概念に詳しい、Build Catalyst の新規ユーザーおよび熟練ユーザーを対象としています。

---

### サポートされるオペレーティング・システム

Build Catalyst は、いくつかの Linux®、Solaris、および Microsoft® Windows® オペレーティング・システムで使用できます。

以下のオペレーティング・システムは Build Catalyst をサポートします。

- Red Hat Enterprise Linux 4、5
- Solaris 9、10 (SPARC)
- Microsoft Windows XP Professional
- Microsoft Windows Server 2003

注: 分散ビルドをサポートするのは Linux および Solaris オペレーティング・システムのみです。

---

## インストールの概要

Build Catalyst のインストールは、Linux<sup>®</sup>、Solaris、または Microsoft<sup>®</sup> Windows<sup>®</sup> オペレーティング・システム上のアーカイブ・ファイルから行います。

詳しくは、539 ページの『サポートされるオペレーティング・システム』を参照してください。

Build Catalyst は、IBM<sup>®</sup> Rational<sup>®</sup> Build Forge<sup>®</sup> インストーラーとは別のインストーラーを備えています。Build Catalyst を Rational Build Forge で使用するには、Rational Build Forge エージェントがインストールされているコンピューターに Build Catalyst をインストールしてください。Build Catalyst は、サポートされるオペレーティング・システムのいずれかを実行するコンピューターであれば、ビルド・ホストとして使用する予定の任意のコンピューターにインストールできます。このコンピューターで Rational Build Forge ジョブに Build Catalyst を使用させる予定がなければ、コンピューターに Rational Build Forge エージェントをインストールする必要はありません。

IBM は、以下の項目を含むアーカイブ・ファイルとして Build Catalyst を配布しています。

- すべてのインストール可能ファイル (バイナリー・ファイル、さまざまなビルトイン Make ファイル、その他) を含む「ダンプ・ファイル」
- インストール・スクリプト

## Linux および Solaris オペレーティング・システムでの Build Catalyst のインストール

Build Catalyst を Linux<sup>®</sup> および Solaris オペレーティング・システムにインストールするには、Build Catalyst アーカイブ・ファイルを使用します。

### このタスクについて

Build Catalyst を分散ビルドに対して使用するには、分散ビルドで使用するすべてのコンピューターに Build Catalyst をインストールします。

### 手順

1. <http://hostname/buildcatalyst> (*hostname* はご使用の管理コンソール・ホストです) から Build Catalyst アーカイブ・ファイルをダウンロードします。
2. アーカイブ・ファイルを任意のディレクトリーに解凍します。
3. スーパーユーザーでない場合には `su` と入力してスーパーユーザーになります。
4. インストーラーを解凍したディレクトリーに移動してから、`install.sh` ファイルが含まれている新規サブディレクトリーに移動します。
5. そのディレクトリーで `./install.sh` を実行します。
  - a. インストール・ディレクトリーを指定します。(デフォルトのディレクトリーは `/opt/rational/buildforge/buildcatalyst` です。)



インストール済み環境が既に存在している場合には (それが、提供した場所以外に存在する場合であっても)、インストーラーは、インストレーションを続行すると既存のインストール済み環境が上書きされたり使用不可になったりすることを警告します。(現行インストール・プロセスは、デフォルトのインストール・ディレクトリーから新規に指定されたインストール・ディレクトリーへのシンボリック・リンクを作成するため、デフォルト・ディレクトリー以外の場所にある既存のインストール済み環境が使用不可になることがあります。) その場合、インストーラーはインストール・ディレクトリーの提供を求めるプロンプトを再び出します。インストール場所を提供すると、インストーラーは提供された場所にファイルを書き込みます。

- b. `rsh` または `ssh` のどちらのバイナリー・ファイルを分散ビルド用に使用するかを指定します (パスに `rsh` がある場合は、これがデフォルト・プログラムになります)。

Build Catalyst バイナリー・ファイルは、インストール・ディレクトリー内の `bin` ディレクトリーに `rafmake` としてインストールされます。インストールによって以下の項目が作成されます。

- `/opt/rational/buildforge/buildcatalyst` からインストール・ディレクトリーへのシンボリック・リンク。
- インストール・ディレクトリー内の `etc/rafbe_starter` からシェル・バイナリー・ファイルへのシンボリック・リンク。シェル・バイナリー・ファイルは `rsh` または `ssh` です。

## タスクの結果

これで、IBM® Rational® Build Forge® ジョブから `rafmake` を開始して、C または C++ ソフトウェアをビルドできるようになりました。また、Rational Build Forge ジョブとは別に `rafmake` を開始することもできます。

## Windows での Build Catalyst のインストール

Build Catalyst を Microsoft® Windows® にインストールするには、Build Catalyst アーカイブ・ファイルを使用します。

### 始める前に

Microsoft Visual C++ 2005 SP1 Redistributable Package バージョン 8.0.50727.762 がコンピューターにインストールされていることを確認してください。インストールされているパッケージは、「コントロール パネル」の「プログラムの追加と削除」を使用して確認できます。

**重要:** Build Catalyst は、このパッケージの他のバージョンを使用しません。バージョン 8.0.50727.762 がインストールされている必要があります。

必要な場合には、正しいバージョンを [www.microsoft.com/downloads/](http://www.microsoft.com/downloads/) からダウンロードすることができます。

Windows サポートの詳細については、539 ページの『サポートされるオペレーティング・システム』を参照してください。

## 手順

1. `http://hostname/buildcatalyst` (`hostname` はご使用の管理コンソール・ホストです) から Build Catalyst アーカイブ・ファイルをダウンロードします。
2. アーカイブ・ファイルを任意のフォルダーに解凍します。
3. そのフォルダーを開いて、`install.bat` ファイルを含む作成済みフォルダーを開きます。
4. `install.bat` を実行します。Build Catalyst バイナリー・ファイルは、`rafmake.exe` ファイルとしてインストールされます。次の表に、ファイルが保存される `bin` フォルダーを示します。

表 12. インストール・フォルダー

オペレーティング・システムのタイプ	場所
32 ビット・オペレーティング・システム	C:\Program Files\IBM\Build Forge\buildcatalyst
64 ビット・オペレーティング・システム	C:\Program Files (x86)\IBM\Build Forge\buildcatalyst

## タスクの結果

これで、IBM® Rational® Build Forge® ジョブから `rafmake` を開始して、C または C++ ソフトウェアをビルドできるようになりました。また、Rational Build Forge ジョブとは別に `rafmake` を開始することもできます。

## Build Catalyst の例

以下の例を参考に、Build Catalyst を使用して単純ビルド、並列ビルド、および分散ビルドを行う方法について理解を深めてください。

### 単純ビルド

単純ビルド (並列でも分散でもないビルド) の場合、**rafmake** コマンドは以下のようにして実行します。

```
rafmake [ -f makefile ] [ options ] ... [ targets ] ...
```

詳細は以下のとおりです。

- *makefile* は使用する Make ファイルです。
- *options* には、545 ページの『`rafmake` ユーティリティーのリファレンス』で説明されているオプションが入ります。
- *targets* はビルドするターゲット・ファイルです。

オペレーティング・システム別の例を以下にリストします。

- Linux および Solaris
  - 特定の Make ファイルにあるデフォルトのターゲットを、そのすべての依存関係とともに無条件にビルドします。

```
% rafmake -u -f project.mk
```
  - 特定の Make ファイルにある特定のターゲットをビルドし、詳細な出力を生成します。

```
% rafmake -v -f project.mk a.out
```

- Windows

- デフォルトの **Make** ファイルにあるデフォルトのターゲットを、**INCL\_DIR** マクロが設定する特定の値を使用してビルドします。

```
Y:¥> rafmake INCL_DIR=C:¥src¥include
```

## 並列ビルド

**rafmake** コマンドは並列ビルドをサポートしています。並列ビルドでは、ビルドを、同時にビルド可能な複数の独立したコンポーネントに分割して、全体的なビルド時間を短縮します。

並列ビルドを実行するための構文は以下のとおりです。

```
rafmake -J num [ -f makefile ] [ options ] ... [ targets ] ...
```

詳細は以下のとおりです。

- *num* は、それぞれ独立したターゲットを同時にビルドする並行ビルド・ジョブの数です。
- *makefile* は使用する **Make** ファイルです。
- *options* には、545 ページの『**rafmake** ユーティリティーのリファレンス』で説明されているオプションが入ります。
- *targets* はビルドするターゲット・ファイルです。

並列ビルドを実行すると、リソースを能率的に利用することができます。マルチコア・プロセッサの場合に特に効果的です。並列ビルドによってどれだけのパフォーマンス向上が得られるかは、ビルドのタイプによって異なります。例えば、CPU 集中型のビルドの場合、並行ビルド・ジョブの最適な数は、プロセッサ（またはコア）の数と同じと考えられます。それより多く指定した場合、パフォーマンスの大幅な向上は実現しない可能性があります。しかし、入出力集中型のビルドの場合、並列ジョブの数を増やすことによってビルド時間が短縮されます。異なる数値でビルドを数回試行して、ご使用のリソースに対する並行ビルド・ジョブの最適な数を判別してください。

並列ビルドの例をオペレーティング・システム別に以下にリストします。

- Linux および Solaris

- 3 つの並行ビルド・ジョブによって、特定の **Make** ファイルにあるデフォルトのターゲットを並列ビルドします。

```
% rafmake -J 3 -f project.mk
```

- Windows

- 5 つの並行ジョブによって、デフォルトの **Make** ファイルにある特定のターゲットを、**INCL\_DIR** マクロが設定する特定の値を使用して並列ビルドします。

```
Y:¥> rafmake -J 5 INCL_DIR=C:¥src¥include software.exe
```

## 分散ビルド

並列ビルドでは、1 つのコンピューター上にあるリソースのみを使用できます。ビルドの分散は並列ビルドをさらに一步押し進めたもので、ビルドの負荷を複数のコンピューターに分散します。

注: Build Catalyst を分散ビルドに対して使用するには、分散ビルドで使用するすべてのコンピューターに Build Catalyst をインストールします。

ビルドを分散させるためには、環境が以下の要件を満たしている必要があります。

- 各コンピューターがソース・コードを共有している。

ビルドの分散先となるすべてのコンピューターにおいて、同じパスで同じソース・コードにアクセスできる必要があります。通常は、NFS マウントされたシェア上にソース・コードを置くこと（およびそれらをすべてのコンピューター上で同じパスにマウントすること）で、すべてのビルド・コンピューター上の同じパスで同じソース・ディレクトリーを使用できるようになります。

- 選択したリモート・シェル (rsh または ssh) でのログインが、パスワードを要求しないように構成されている。

ユーザーがホスト・コンピューターから分散ビルドを開始すると、ビルドの分散先となるすべてのコンピューター上で、**rafmake** コマンドによりビルド・エンジン (rafbe) が内部的に開始されます。ビルド・エンジンは、Build Catalyst のインストール時に構成された **ssh** コマンドまたは **rsh** コマンドを使用して開始されます。/opt/rational/buildforge/buildcatalyst/etc/rafbe\_starter ファイルは、インストール時にユーザーが選択したリモート・シェルへのシンボリック・リンクです。ビルドを開始するユーザーは、この選択されたリモート・シェルを使用しているすべてのコンピューターに、パスワードなしでログインする必要があります。パスワードなしでログインすることにより、各コンピューター上でビルド・エンジンをサイレント状態で開始できます。パスワードを要求しないログインのセットアップ方法については、**rsh** コマンドまたは **ssh** コマンドの資料を参照してください。

- ビルド・ホスト・ファイルに、各コンピューターへのビルドの分散方法が記述されている。

通常、ビルド・ホスト・ファイルには、多くのコンピューターのホスト名または IP アドレスが記述されたいくつかの行が含まれています。これらのコンピューターが、ビルドの分散先となるコンピューターです。このファイルには、**-idle num** などの他のオプションを記述した行も含まれています。このタイプの行では、その下にリストされたコンピューターでビルドをスケジュールに入れる際の基準となるアイドル率が指定されます。明示されていない場合、アイドル率は 50 であると想定されます。ビルド・ホスト・ファイルの例を以下に示します。

```
### Contents of build hosts file
host1.mydomain.com
-idle 25
host2.mydomain.com
host2.mydomain.com
192.168.0.200
-idle 75
host3.mydomain.com
### End of build hosts file
```

この例では、host1、host2、host3、および 192.168.0.200 の 4 つのコンピューターにビルドが分散されます。host1.mydomain.com コンピューターのアイドル率は明示されていないため、50 であると想定されます。host2 コンピューターおよび 192.168.0.200 コンピューターのアイドル率は 25 に指定されています。これは、これらのいずれかのコンピューターの負荷率またはビジー率が 75% より低くな

った (つまり、少なくとも 25% のアイドル率になった) 場合に、そのコンピューターでビルドがスケジュールに入れられることを示します。コンピューターのアイドル率が 25% 未満の場合、それらのコンピューターでビルドはスケジュールに入れられません。また、この例では host2 コンピューターが 2 回リストされています。結果として、host2 コンピューターでは 2 つの並列ビルド・スレッドを開始できます。コンピューターの構成に応じて、ビルドでのリソース使用が最適化されるようにビルド・ホスト・ファイルを調整することができます。

ビルド・ホスト・ファイルを作成した後、以下のコマンド構文を使用することによって分散ビルドを開始できます。

```
rafmake -J num -B build_hosts_file [ -f makefile ]  
[ options ] ... [ targets ] ...
```

-J オプションの *num* 値には、全コンピューターのビルド・スレッドの総数を指定します。

-B オプションがない場合、**rafmake** コマンドは分散ビルドではなく並列ビルドを実行します。-B オプションおよび有効なビルド・ホスト・ファイルを指定した場合、ビルド・エンジンは、ビルド・ホスト・ファイルに指定された各コンピューターで分散ビルドをスケジュールに入れます。

---

## rafmake ユーティリティーのリファレンス

Build Catalyst のインターフェースは、**rafmake** という名前の **make** ユーティリティーです。このインターフェースには、その動作を制御する各種のオプションがあります。

このコマンドの構文は、下の表に示すように、オペレーティング・システムごとに異なります。

表 13. オペレーティング・システム別のオプション

オペレーティング・システム	オプション
Linux および Solaris (ターゲットのビルド)	rafmake [ -f <i>makefile</i> ] ... [ -ukinservwdpqUN ] [ -J <i>num</i> ] [ -B <i>bldhost-file</i> ] [ -c <i>compat-mode</i> ] [ -C <i>change-directory</i> ] [ -A <i>BOS-file</i> ] ... [ <i>macro=value</i> ... ] [ <i>target-name</i> ... ]
Windows (ターゲットのビルド)	rafmake [ -f <i>makefile</i> ] ... [ -ukinservwdpqUN ] [ -J <i>num</i> ] [ -c <i>compat-mode</i> ] [ -C <i>change-directory</i> ] [ -A <i>BOS-file</i> ] ... [ <i>macro=value</i> ... ] [ <i>target-name</i> ... ]
すべて (rafmake ユーティリティーのバージョン情報の表示)	rafmake { -ver/sion   -VerAll }

## オプションの説明

**rafmake** コマンドは、GNU バージョン 3.80 の **make** コマンドが持つ共通オプションの大部分をサポートしています。また、**rafmake** コマンドには追加オプションが用意されています。

**ヒント:** 引数を持たないオプションは、例えば `-rNi` のように結合することができます。

表 14. オプションの説明

オプション	説明
<code>-f makefile</code>	入力ファイルとして <b>Make</b> ファイルを使用します。このオプションを省略した場合、 <b>rafmake</b> コマンドは、現行作業ディレクトリーで <code>makefile</code> および <code>Makefile</code> という名前の入力ファイルを (この順番で) 検索します。 <code>-f makefile</code> 引数のペアは複数指定できます。複数の入力ファイルは事実上連結されます。
<code>-u</code>	(Unconditional (無条件)) 指定されたすべてのターゲット、およびその依存関係を、それらが再ビルドを必要とするかどうかに関係なく再ビルドします。 ( <code>-U</code> も参照してください。)
<code>-k</code>	障害が発生した場合、現在の項目での作業は中止しますが、その項目に依存しない他のターゲットでは作業を続行します。
<code>-i</code>	コマンドが返すエラー・コードを無視します。
<code>-n</code>	(No-execute (実行なし)) 再ビルドが必要なターゲットの <b>Make</b> ファイルにあるコマンド行を (アットマーク (@) で始まる行を含めて) リストしますが、実行はしません。例外として、ストリング <code>\$(MAKE)</code> を含むコマンドは常に実行されます。
<code>-s</code>	(Silent (サイレント)) コマンド行を実行する前にそれらの行をリストしません。
<code>-e</code>	環境変数により、 <code>makefile</code> ファイル内のマクロの代入がオーバーライドされます。 (ただし、環境変数は、コマンド行またはビルド・オプション指定にある <code>macro=value</code> の代入によりオーバーライドされます。)
<code>-r</code>	(No rules (規則なし)) <code>builddcatalyst-home-dir/etc/builtin.mk</code> ファイル (Linux および Solaris) または <code>builddcatalyst-home-dir¥etc¥builtin.mk</code> ファイル (Windows) 内の組み込み規則を使用しません。 <code>-c</code> オプションとともに使用した場合、 <code>-r</code> オプションは、プラットフォーム固有のスタートアップ・ファイルの読み取りも無効にします。詳しくは、 <code>-c</code> オプションを参照してください。
<code>-v</code>	(Verbose (詳細)) デフォルトの出力モードよりもわずかに詳細です。
<code>-w</code>	(Working directory (作業ディレクトリー)) <b>Make</b> ファイルの実行前と実行後の両方に、作業ディレクトリーを含むメッセージを表示します。
<code>-d</code>	(Debug (デバッグ)) 非常に詳細です。ビルド時に <b>rafmake</b> コマンドが読み取る環境変数のリストが含まれています。このオプションは、 <b>Make</b> ファイルをデバッグする場合にのみ使用してください。
<code>-p</code>	(Print (表示)) ターゲット固有のマクロ定義および暗黙的な規則を含む、すべてのターゲット記述およびすべてのマクロ定義をリストします。どれも実行はしません。
<code>-q</code>	(Query (照会)) <b>Make</b> ファイルのターゲットを評価しますが、ビルド・スクリプトは実行しません。 <b>rafmake</b> コマンドは、以下のいずれかの応答を返します。 <ul style="list-style-type: none"> <li>• 0 ターゲットが最新の場合</li> <li>• 1 再ビルドが必要なターゲットがある場合</li> </ul>



表 14. オプションの説明 (続き)

オプション	説明
-U	ゴール・ターゲットのみを無条件にビルドします。サブターゲットはビルドしません。コマンド行でどのターゲットも指定しなかった場合は、デフォルト・ターゲットがゴールです。 (-u オプションは、ゴール・ターゲットとビルド依存関係の両方を無条件にビルドします。)
-N	1 つ以上のビルド・オプション指定 (BOS) ファイルを読み取るデフォルト・プロシージャを無効にします。
-J <i>num</i>	並列ビルド機能を有効にします。並行ターゲット再ビルドの最大数は、整数 <i>num</i> に設定されます。 <i>num</i> =0 の場合、並列ビルドは無効になります。(これは -J オプションを指定しないことと同じです。)あるいは、 <i>num</i> に RAFMAKE_CONC 環境変数の値を指定することもできます。
-B <i>bldhost-file</i>	並列ビルドのビルド・ホスト・ファイルとして <i>bldhost-file</i> ファイルを使用します。 -B オプションを指定しなかった場合、 <b>rafmake</b> コマンドは、ホーム・ディレクトリー内の <i>.bldhost.\$RAFMAKE_HOST_TYPE</i> ファイルを使用します。 -B オプションを使用する場合は、同時に -J オプションを使用するか、RAFMAKE_CONC 環境変数を設定する必要があります。
-c <i>compat-mode</i>	<p>(Compatibility (互換性)) <b>rafmake</b> コマンドを互換モードで起動します。</p> <p>(あるいは、RAFMAKE_COMPAT 環境変数を使用して互換モードを指定することもできます。)</p> <p><i>compat-mode</i> 変数は、以下のいずれかの値に設定できます。</p> <p><b>gnu</b></p> <p>フリー・ソフトウェア財団の Gnu <b>make</b> プログラムをエミュレートします。組み込みの Make 規則を定義するために、<b>rafmake</b> コマンドは <i>builtin.mk</i> ではなく <i>gnubuiltin.mk</i> を読み取ります。</p> <p>このモードがデフォルトの互換モードです。</p> <p><b>std</b> IBM Rational® ClearCase® の <b>clearmake</b> コマンド互換モードを有効にして <b>rafmake</b> コマンドを実行します。</p> <p>Linux システムおよび Solaris システムの場合のみ、<i>compat-mode</i> 変数は以下のいずれかの値も持つことができます。 -c オプションは、Linux および Solaris プラットフォームから独立しています。しかし、一部のモードではシステム固有のファイルを読み取ろうとするため、それらのファイルが存在しない場合にコマンドが失敗します。</p> <p><b>sun</b> SunOS システムに用意された標準の <b>make(1)</b> コマンドをエミュレートします。</p> <p><b>aix</b> IBM AIX® システムに用意された標準の <b>make(1)</b> コマンドをエミュレートします。</p>
-C <i>dir</i>	ディレクトリーを変更します。ビルドを開始する前に、 <b>rafmake</b> コマンドが <i>dir</i> ディレクトリーに変更されます。



表 14. オプションの説明 (続き)

オプション	説明
-A <i>BOS-file</i>	<p>デフォルトで読み取られる BOS ファイルの直後に読み取られる BOS ファイルを指定します。複数の BOS ファイルを指定するには、このオプションを複数回使用してください。</p> <p>このオプションとともに -N を使用した場合、<b>rafmake</b> は、デフォルトの BOS ファイルではなく、指定の BOS ファイルを読み取ります。</p> <p>あるいは、環境変数 RAFMAKE_OPTS_SPECS の値として、コロンで区切った BOS ファイルのパス名リスト (Linux および Solaris) か、セミコロンで区切った BOS ファイルのパス名リストを指定することもできます。</p>
-ver/sion	<b>rafmake</b> コマンドのバージョン情報を表示します。
-VerAll	<b>rafmake</b> コマンド、および <b>rafmake</b> コマンドが使用するライブラリー (Linux および Solaris) または DLL (Windows) のバージョン情報を表示します。

## Build Catalyst の環境変数

Build Catalyst は、簡単に使用できるようにするために複数の環境変数をサポートしています。

これらの変数は、オペレーティング・システム別に以下の区分にグループ化されます。

- 『Linux、Solaris、および Microsoft Windows に共通の環境変数』
- 550 ページの『Linux および Solaris のみの環境変数』
- 550 ページの『Windows のみの環境変数』

### Linux、Solaris、および Microsoft Windows に共通の環境変数

#### RAFMAKE\_CONC

**rafmake** ビルドでの並行性レベルを設定します。この変数は、-J オプションと同じ値になります。**rafmake** コマンドの使用時に -J オプションを指定すると、この変数の設定値が指定変更されます。

#### RAFMAKE\_COMPAT

**rafmake** コマンドの互換モードを指定します。この変数は、-c オプションと同じ値になります。**rafmake** コマンドの使用時に -c オプションを指定すると、この変数の設定値が指定変更されます。

デフォルト: なし

#### RAFMAKE\_MAKEFLAGS

**rafmake** コマンド・オプションを指定するための代替または補足メカニズムを提供します。RAFMAKE\_MAKEFLAGS 環境変数には、コマンド行オプションに使用するものと同じキー文字のストリングを含めることができますが、引数を使用するオプションは無効です。**rafmake** コマンド行でオプションを指定すると、この環境変数の設定値と矛盾する場合には環境変数の設定値が指定変更されます。

デフォルト: なし

#### RAFMAKE\_OPTS\_SPECS

コロン (Linux および Solaris の場合) またはセミコロン (Windows の場合) で区切ってパス名のリストを提供します。各パス名は、**rafmake** コマンドが読み取るビルド・オプション仕様 (BOS) ファイルを指定します。**rafmake** コマンド行で 1 つ以上の **-A** オプションを使用して BOS ファイルを指定する代わりに、この変数を使用することができます。

デフォルト: 未定義

#### RAFMAKE\_SHELL\_FLAGS

ビルド・スクリプト・コマンドを実行するサブシェル・プログラムに渡す **rafmake** コマンド・オプションを指定します。

デフォルト:

- Linux および Solaris: **-e**
- Windows: なし

#### RAFMAKE\_SHELL\_REQUIRED

**rafmake** コマンドに、SHELL マクロで指定したシェル・プログラムでビルド・スクリプトを実行させます。**rafmake** コマンドにシェル・プログラムでビルド・スクリプトを実行させるには、この変数を **TRUE** に設定してください。ビルド・スクリプトを直接実行するように **rafmake** コマンドを構成するには、この変数を設定解除してください。

デフォルト: **rafmake** コマンドはビルド・スクリプトを直接実行します。

#### RAFMAKE\_VERBOSITY

**rafmake** コマンドのメッセージ・ロギング・レベルを以下のように指定します。

- 1 コマンド行の **-v** (詳細) と同等
- 2 コマンド行の **-d** (デバッグ) と同等
- 0 または未定義

標準メッセージ・ロギング・レベルと同等

コマンド行で **-v** または **-d** も指定した場合、高い方の値が有効になります。

デフォルト: 0

#### MAKEFLAGS

**make** コマンドに渡す 1 つ以上のフラグをリストします。**rafmake** コマンドは、開始時に **MAKEFLAGS** 環境変数の内容を読み取ります。コマンドはその後で、コマンド行で渡される、Build Catalyst 固有でないフラグを含めるように、変数を修正します。Build Catalyst 固有のフラグは **RAFMAKE\_MAKEFLAGS** 環境変数を介して渡されます。**rafmake** コマンドは、**MAKEFLAGS** 変数内でこれらのフラグを検出すると、該当フラグを **RAFMAKE\_MAKEFLAGS** 変数に移動します。

**MAKEFLAGS** 変数を介して渡されるフラグは、**-I**、**-p**、**-N**、**-w**、**-e**、**-r**、**-i**、**-k**、**-n**、**-q**、**-s** です。

RAFMAKE\_MAKEFLAGS 変数を介して渡されるフラグは、-A、-B、-N、-v、-c、-U、-M、-u、-d です。

デフォルト: なし

## Linux および Solaris のみの環境変数

### RAFMAKE\_BLD\_HOSTS

**rafmake** コマンドでターゲットをビルドできる、1 つ以上のビルド・ホストを指定します。

デフォルト: 未定義

### RAFMAKE\_HOST\_TYPE

並列ビルド (-J オプション) 時に使用するビルド・ホスト・ファイルの名前を判別します。このファイルは、ホーム・ディレクトリー内の `.blldhost.$RAFMAKE_HOST_TYPE` です。(ホーム・ディレクトリーは、パスワード・データベースを調べることによって判別されます。)

コマンド行で -B オプションを指定すると、この変数の設定値が指定変更されます。

C Shell ユーザー: この変数は、`.login` ファイル内ではなく `.cshrc` ファイル内で設定してください。並列ビルド機能で呼び出されるリモート・シェルは `.login` ファイルを読み取りません。

RAFMAKE\_HOST\_TYPE は **make** マクロとしてコーディングすることもできます。

デフォルト: なし

## Windows のみの環境変数

### RAFMAKE\_NO\_ESC\_PATT\_CHARS

**rafmake** コマンドの GNU 互換モードでエスケープ文字 (¥) を指定変更します。例えば、**rafmake** コマンドと GNU **make** コマンドはともに、¥% スtringがリテラル文字 % を表すものと想定します。これらのコマンドは、この規則をパターン規則としては扱いません。**rafmake** コマンドがリテラル文字を表すためにエスケープ文字を使用しないようにするには、この環境変数を任意の非ヌル値に設定してください。

### RAFMAKE\_PNAME\_SEP

**rafmake** コマンドが構成するパス名のパス名分離文字を設定します。この変数は、`Make` ファイル内、`BOS` ファイル内、またはコマンド行で設定することも、環境変数として設定することもできます。

デフォルト: この変数を設定しない場合、あるいはスラッシュ (/) または円記号 (¥) 以外の任意の値に設定した場合、**rafmake** コマンドはパス名分離文字として円記号 (¥) を使用します。

---

## 第 26 章 Rational Build Forge の統合方式

Build Forge は、以下の方式を使用して他の製品と統合します。

- コマンド行を使用した統合。Rational Build Forge ステップは、エージェント・ホストでコマンドを実行します。
- アダプターを使用した統合。アダプターは、ソース制御管理システムとの統合を支援する特別な機能を備えた、XML ベースのプログラムです。
- IDE プラグインを使用した統合。IDE プラグインを使用すると、IDE ユーザー・インターフェースからコンソールにアクセスできます。
- Websphere 製品との統合。Apache Tomcat に代わって、Websphere Application Server を Web アプリケーション・サーバーとして使用することができます。Apache HTTP サーバーに代わって、IBM HTTP サーバーを使用することができます。

---

### コマンド行インターフェースを使用した統合

Build Forge は、コマンド行インターフェースを持つアプリケーションと簡単に統合します。

このタイプの統合をセットアップするには、以下の手順を実行します。

1. ホストに Build Forge コンソールをインストールします。
2. アプリケーション・ホスト (またはアプリケーションにアクセスできるホスト) に Build Forge エージェントをインストールします。
3. コンソールで、サーバー・リソースおよびサーバー認証を作成します。インストールした Build Forge エージェントにアクセスするように、サーバー・リソースを構成します。
4. アプリケーションでコマンドが実行されるように、必要に応じてエージェント環境とホスト環境を構成します。これは、使用する Build Forge のユーザー・アカウントを作成し、実行するエージェントの PATH が正しくセットアップされるようにする、というような単純な処理で済む場合があります。また、アプリケーションでコマンドを実行するために使用するクライアントをインストールして構成する、という作業が必要になる場合もあります。例えば、Rational ClearCase および Rational ClearQuest では、コマンドを実行するためにクライアント・アプリケーションを使用する必要があります。

セットアップが完了すると、Build Forge で作成するプロジェクトに、アプリケーション・コマンドを実行するステップを組み込むことができます。ステップの成功または失敗状況に基づいてプロジェクトの実行を制御したり、ログ・フィルタを設定して Build Forge ログから、コマンドが返す出力のパターンをスキャンすることができます。アプリケーションを使用するためには、プロジェクト・ロジックでの追加セットアップが必要になることがあります。例えば、ClearCase と統合するには、ビューの作成、開始およびビューへのデータの取り込みが必要になります。

条件ステップとループ・ステップを使用して、特定のコマンドの成功または失敗に応じてどのようにするかを制御できます。

ソース制御アプリケーションの典型的な用途はビルド回避です。最後にビルドが実行されてからソース・コード更新がチェックインされたかどうかを照会するためのコマンドが実行されます。

- 更新がない場合、ビルドは実行されません。
- 更新が行われている場合、ビルドが実行されます。

モジュールの依存関係は、プロジェクト・ステップの実行ロジック内で表現できます。特定のモジュールがビルドされたときに他のモジュールまたはソフトウェア・プロジェクト全体がビルドされるようにするかどうかを制御できます。

ビルドの成功または失敗に応じてプロジェクト・メンバーのグループにどのように通知するかを制御するには、通知テンプレートを使用します。連続統合環境では、失敗した場合に通知して、問題となるコードを修正できるようにする必要があります。コマンド行を使用した統合では、プロジェクト・メンバーのグループ全体に通知することのみが可能です。プロジェクト・メンバーは、そのプロジェクトのステップ・ログを検査して、失敗の原因となったコードを判別する必要があります。

---

## アダプターを使用した統合

アダプターを使用すると、Rational Build Forge と外部アプリケーションを統合できます。

IBM Rational ClearCase および IBM Rational ClearQuest との統合用のサンプル・アダプター・テンプレートが用意されています。カスタム・アダプターの作成に使用できる Adaptor Toolkit は、別個にライセンス交付されるフィーチャーです。

アダプターの動作は XML ファイルによって定義されます。この XML ファイルの要素は、組み込まれた DTD によって指定されます。Build Forge は、プロジェクト・ステップと共同してアダプターを実行します。

アダプターは、統合ベースのビルド用に以下の追加ツールを提供します。

- 内部条件ロジック: 内部変数の設定値に基づきます。
- コマンド定義: コマンド・ステートメントおよび変数からアプリケーション・コマンドを作成できます。
- 応答スキャン: 各コマンドに応答してスキャンするパターンを定義できます。
- 動的通知グループ: アプリケーションから収集したデータに基づく通知が可能です。例えば、通知対象となるチーム・メンバーのグループを、コード変更をチェックインしたメンバーのみから構成することができます。
- スクリプトの実行: アダプターは、Build Forge ホストでスクリプトを実行できます。これは、アプリケーション・ホストで実行されるスクリプトおよびコマンドから独立させることができます。

## アダプター要件

このトピックでは、インストール要件および構成要件と、Build Forge アダプターのソフトウェア要件を説明します。

使用する環境変数が、Build Forge 製品で提供されるアダプターを使用するための要件を満たしているかを、次のように検証します。

- Build Forge エージェントを、外部アプリケーションが稼働中のコンピュータ上にインストールします。製品によっては、追加のセットアップ要件が必要なものがあります。Rational Clear Case および Rational ClearQuest では、追加のセットアップが必要となります。これらの製品についての統合に関するセクションを参照してください。
- ClearCase または ClearQuest 以外のアプリケーション・テンプレートを使用する場合、Build Forge アダプター・ツールキットのライセンス・キーをインストールします。
- Build Forge 製品でサポートされているバージョンの外部アプリケーションを実行します。

表 15. アダプターに対してサポートされているアプリケーション・バージョン

アプリケーション	バージョン
Rational ClearCase	6.0 以降
Rational ClearQuest	7.0 以降
CVS	1.1, 1.2
Microsoft Visual SourceSafe	6.0
Perforce	2005.1
StarTeam	2005 年、リリース 2
サブバージョン	1.3.1 以降

## アダプター

このトピックでは、アダプターに関する一般情報の一部を提供します。また、アダプターがその他の Build Forge オブジェクトおよび機能と相互作用する方法も説明します。

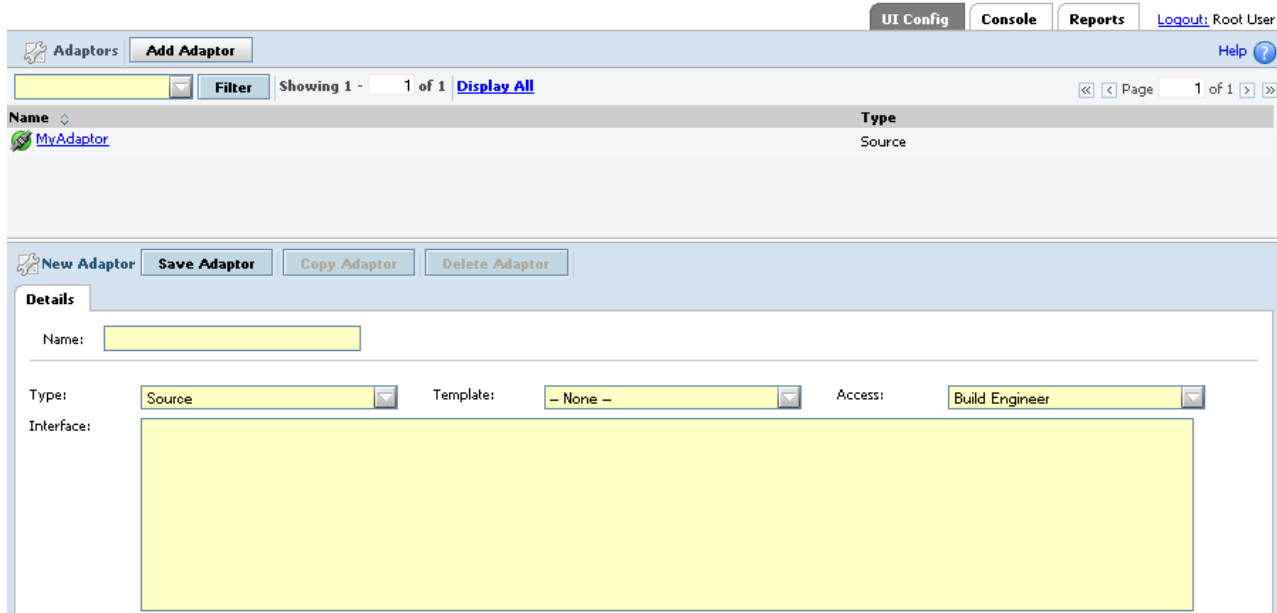
このセクションで提供される情報を検討することで、アダプターを使い始めるための知識を得ることができます。

### アダプターについて

アダプターは、外部アプリケーションへのインターフェースです。アダプターにより、Build Forge プロジェクトは外部アプリケーションと情報を交換して、目的を果たします。

例えば、ソース・コード・アプリケーションのアダプターは、Build Forge プロジェクトを実行する前提条件として、リポジトリにソース・コードの変更がないかをチェックします。ソース・コード・ファイルに変更があれば、プロジェクトを実行します。変更がなければ、プロジェクトは実行されません。

**アダプター・パネルについて:** アダプターの作成および編集には、「アダプター」パネルを使用します。パネルを表示するには、「プロジェクト」 > 「アダプター」を選択します。



パネルには、以下のフィールドがあります。

**名前** スペースが含まれていない固有のアダプター名

**タイプ** アダプター・タイプ

**テンプレート**

「なし」、または Rational Build Forge に組み込まれているアダプター・テンプレートの 1 つ

注: Rational Build Forge のバージョンを更新すると、この設定はクリアされます。

**アクセス**

アダプターの表示および編集をグループ・メンバーに制限するアクセス・グループ

**インターフェース**

ユーザーが提供する XML インターフェース、またはテンプレートに付属の XML インターフェース

## アダプター・テンプレート

アダプターは、アダプター・テンプレートのインスタンスです。アダプターを作成する場合は、アダプターに固有の名前を割り当て、テンプレートに関連付けます。

テンプレートは、XML ファイルです。この XML には、情報を収集するためのアプリケーション・コマンド、情報を分析するための指示、および BOM レポートで結果を表示するためのフォーマット詳細が含まれています。

Build Forge で提供されるテンプレートは、修正なしで使用されるように設計されています。ただし、新規アダプター・テンプレートを作成するために、テンプレートを変更したり、テンプレートをモデルとして使用したりすることができます。

アダプター・テンプレートは、次のディレクトリーにインストールされています。



*bfinstall¥interface* (Microsoft Windows オペレーティング・システムを実行しているコンピューターの場合)

*bfinstall/Platform/interface* (UNIX または Linux オペレーティング・システムを実行しているコンピューターの場合)

## サンプル・アダプター・テンプレート

いくつかのアプリケーション・タイプ (ソース・コード、障害追跡、テスト、およびパッケージ化) 用のサンプル・アダプター・テンプレートが用意されています。アダプターは、サポートするアプリケーション・タイプごとに分類されます。

Build Forge 製品では、次の表に示す、アプリケーション用サンプル・アダプター・テンプレートを用意しています。ClearCase および ClearQuest のテンプレートでは、別個のライセンス・キーは必要ありません。その他のアプリケーション・テンプレートは、Build Forge アダプター・ツールキットを介してライセンスが交付されます。

アダプター・テンプレートの XML ファイルは、次のディレクトリーに .xml ファイル拡張子を付けてインストールされます。

*bfinstall¥interface* (Microsoft Windows オペレーティング・システムを実行しているコンピューターの場合)

*bfinstall/Platform/interface* (UNIX または Linux オペレーティング・システムを実行しているコンピューターの場合)

### 注:

アダプター・リンクを介してアダプターが接続されたプロジェクトが実行された場合、そのアダプターはデバッグ・モードではないため、アダプター・ステップが失敗し、アダプターによりジョブが除去されます。

表 16. アダプター・テンプレートの説明

アダプター・テンプレート名	説明	タイプ
ClearCaseBaseline	ClearCase ビューのディレクトリーをスキャンします。  ClearCase からレポートされた分岐およびバージョン情報を BOM レポートに書き込みます。	source
ClearCaseByBaselineActivities	ClearCase ビューのコンテンツから新規ベースラインを作成します。  新規ベースラインと、前のアダプター実行でのベースラインを比較して、変更アクティビティーを特定します。  各変更アクティビティーに関して、次の情報を BOM レポートに書き込みます: アクティビティー、変更されたファイル、ユーザー、日付、コメント、およびバージョン。  変更されたファイルごとに、(diff コマンドの出力から得られる) 変更詳細を BOM レポートに書き込みます。	source

表 16. アダプター・テンプレートの説明 (続き)

アダプター・テンプレート名	説明	タイプ
ClearCaseByBaselineVersions	<p>ClearCase ビューのコンテンツから新規ベースラインを作成します。</p> <p>新規ベースラインと、前のアダプター実行でのベースラインを比較して、変更されたファイルを特定します。</p> <p>変更された各ファイルに関して、次の情報を BOM レポートに書き込みます: ファイル名、バージョン、日付、ユーザー、およびコメント。</p> <p>変更されたファイルごとに、(diff コマンドの出力から得られる) 変更詳細を BOM レポートに書き込みます。</p>	source
ClearCaseByDate	<p>2 つの日付の間に変更がないか、ClearCase ビューを照会します。デフォルトの日付は、現行のタイム・スタンプおよび前のアダプター実行時のタイム・スタンプです。</p> <p>変更された各ファイルに関して、次の情報を BOM レポートに書き込みます: ファイル名、バージョン、日付、ユーザー、およびコメント。</p> <p>変更されたファイルごとに、(diff コマンドの出力から得られる) 変更詳細を BOM レポートに書き込みます。</p>	source
ClearCaseByLabel	<p>新規ラベルを作成して、ClearCase ビューのコンテンツに適用します。</p> <p>新規ラベルと、前のアダプター実行でのラベルを比較して、変更されたファイルを特定します。</p> <p>変更された各ファイルに関して、次の情報を BOM レポートに書き込みます: ファイル名、バージョン、日付、ユーザー、およびコメント。</p> <p>変更されたファイルごとに、(diff コマンドの出力から得られる) 変更詳細を BOM レポートに書き込みます。</p>	source
ClearQuestBaseClearCaseByDate	<p>2 つの日付の間に変更がないか、ClearCase ビューを照会します。デフォルトの日付は、現行のタイム・スタンプおよび前のアダプター実行時のタイム・スタンプです。</p> <p>変更された各ファイルについて、ClearQuest 変更 ID を指定する CrmRequest ハイパーリンク属性を探します。ClearQuest 状況により解決が可能な場合は、ジョブ情報を追加して変更 ID の解決を試行し、ClearQuest 内の不良レコードを解決します。</p> <p>変更された各ファイルに関して、次の情報を BOM レポートに書き込みます: ファイル名、不良 ID、不良状況、および ClearQuest エラー。</p>	defect

表 16. アダプター・テンプレートの説明 (続き)

アダプター・テンプレート名	説明	タイプ
ClearQuestClearCaseByActivity	<p>ClearCase アクティビティーのリストに関連付けられた ClearQuest 不良レコードを検出します。検出された各不良レコードについて、ClearQuest 状況により解決が可能な場合は、ジョブ情報を追加して ClearQuest 内の不良レコードを解決します。</p> <p>ClearCase アクティビティー ID および ClearQuest 不良状況に関連付けられたファイルに関する情報を、BOM レポートに書き込みます。</p>	defect
ClearQuestUCMClearCaseByDate	<p>2 つの日付の間に変更がないか、ClearCase ビューを照会します。デフォルトの日付は、現行のタイム・スタンプおよび前のアダプター実行時のタイム・スタンプです。</p> <p>Rational 統一変更管理 (UCM) を使用して、その結果を生成します。</p> <p>変更された各ファイルに関して、次の情報を BOM レポートに書き込みます: ファイル名、不良 ID、不良状況、および ClearQuest エラー。</p>	defect
CVSv1Baseline	<p>変更されたファイルを探している Build Forge エージェントで、CVS ディレクトリーをスキャンします。</p> <p>次の情報を BOM レポートに書き込みます: 変更されたファイル名、状況、作業バージョン、リポジトリ・バージョン、およびスティッキー・タグ。</p>	source
CVSv1ByDate	<p>2 つの日付の間に変更がないか、CVS ビューを照会します。デフォルトの日付は、現行のタイム・スタンプおよび前のアダプター実行時のタイム・スタンプです。</p> <p>次の情報を BOM レポートに書き込みます: 変更タイプ、日付、ユーザー名、バージョン、およびファイル名。</p> <p>変更されたファイルごとに、(diff コマンドの出力から得られる) 変更詳細を BOM レポートに書き込みます。</p>	source
CVSv1ByTag	<p>新規タグを CVS モジュールに適用します。</p> <p>新しくタグ付けされたモジュールと、前のアダプター実行時にタグ付けされたモジュールの間の相違を比較します。</p> <p>次の情報を BOM レポートに書き込みます: ファイル名、改訂、状況、日付、時刻、変更の作成者、およびコミット・コメント。</p> <p>変更されたファイルごとに、(diff コマンドの出力から得られる) 変更詳細を BOM レポートに書き込みます。</p>	source

表 16. アダプター・テンプレートの説明 (続き)

アダプター・テンプレート名	説明	タイプ
CVSv2ByDate	<p>2 つの日付の間に変更がないか、CVS ビューを照会します。 デフォルトの日付は、現行のタイム・スタンプおよび前のアダプター実行時のタイム・スタンプです。</p> <p>次の情報を BOM レポートに書き込みます: 変更タイプ、日付、ユーザー名、バージョン、およびファイル名。</p> <p>変更されたファイルごとに、(diff コマンドの出力から得られる) 変更詳細を BOM レポートに書き込みます。</p>	source
JazzJBE	Rational Team Concert 3.x 以降のユーザー向け。詳しくは、Rational Team Concert の資料を参照してください。	source
JazzSCM	<p>変更されたソース・ファイルがないか、Rational Team Concert に照会します。</p> <p>次の情報を BOM レポートに書き込みます: 変更されたソース・ファイル、変更されたファイルのサイズ、変更されたコンポーネント、および変更セット。</p>	source
JazzSCMv2	<p>変更されたソース・ファイルがないか、Rational Team Concert バージョン 2.x に照会します。</p> <p>次の情報を BOM レポートに書き込みます: 変更されたソース・ファイル、変更されたストリーム、変更されたコンポーネント、変更セット、変更を行ったユーザー、チェックイン・コメント、および Jazz 作業項目。</p>	source
JUnit	<p>JUnit テストを実行します。</p> <p>次の情報を BOM レポートに書き込みます: 実行されたテスト、失敗、エラー、および経過時間。</p>	test
PerforceByDate	<p>アダプターの実行以降生じた変更がないか、Perforce クライアントを照会します。</p> <p>次の情報を BOM レポートに書き込みます: 変更、日付、時刻、ユーザー、Perforce クライアント、およびコメント。</p> <p>変更詳細 (diff コマンド出力による) を BOM レポートに書き込みます。</p>	source
PerforceByRev	<p>直前のリポジトリ改訂以降生じた変更がないか、Perforce クライアントを照会します。</p> <p>次の情報を BOM レポートに書き込みます: 変更、日付、時刻、ユーザー、Perforce クライアント、およびコメント。</p> <p>変更詳細 (diff コマンド出力による) を BOM レポートに書き込みます。</p>	source

表 16. アダプター・テンプレートの説明 (続き)

アダプター・テンプレート名	説明	タイプ
Quota	<p>Windows フォルダーを照会し、指定したしきい値サイズを超えるサブフォルダーがないかを判別します。</p> <p>各サブフォルダーに関して、次の情報を BOM レポートに書き込みます: フォルダー・サイズ、所有者、最終変更日。</p> <p>しきい値サイズを超えたサブフォルダーのリストを BOM レポートに書き込みます。</p> <p><b>重要:</b> 値を超えたディレクトリーを所有する各ユーザーに、次の順序で通知が送信されます: 最初に Build Forge システム内の名前順、その後にユーザー名順。通知を受けるユーザーが存在しない場合、&lt;notify&gt; は失敗します。</p>	source
StarTeamBaseline	<p>フォルダーを照会して StarTeam ビューを見つけ、ファイルに関する情報を収集します。</p> <p>次の情報を BOM レポートに書き込みます: ファイル名、状況、改訂、および分岐。</p>	source
StarTeamByDate	<p>StarTeam API を使用して、StarTeam ビューを照会し、現在日付と前のアダプター実行時の間の変更を特定します。</p> <p>次の情報を BOM レポートに書き込みます: 変更されたファイルおよびディレクトリー、ユーザー、バージョン、日付、および変更コメント。</p> <p>変更詳細 (diff コマンド出力による) を BOM レポートに書き込みます。</p>	source
SubversionByDate	<p>過去の日付と現在の日付の間で生じたりポジトリの変更がないか、サブバージョンを照会します。</p> <p>次の情報を BOM レポートに書き込みます: 変更タイプ、改訂、ユーザー、ファイルまたはディレクトリー、および変更日。</p> <p>次の情報を BOM レポートに書き込みます: ファイル名、状況、改訂、および分岐。</p>	source
SubversionByRev	<p>現在の改訂と前の改訂の間で生じたりポジトリへの変更がないか、サブバージョンを照会します。</p> <p>各変更に関して、次の情報を BOM レポートに書き込みます: 改訂、ユーザー、変更タイプ、ファイルまたはディレクトリー・パス、および変更日。</p> <p>変更詳細 (diff コマンド出力による) を BOM レポートに書き込みます。</p>	source

表 16. アダプター・テンプレートの説明 (続き)

アダプター・テンプレート名	説明	タイプ
SynergyByDate	<p>2 つの日付の間に変更がないか、Rational Synergy を照会します。デフォルトの日付は、現行のタイム・スタンプおよび前のアダプター実行時のタイム・スタンプです。</p> <p>変更された各ファイルに関して、次の情報を BOM レポートに書き込みます: ファイル名、バージョン、ユーザー、プロジェクト、および日付。</p> <p>変更されたファイルごとに、(diff コマンドの出力から得られる) 変更詳細を BOM レポートに書き込みます。</p>	source
VSSByDate	<p>前の日付と現在の日付の間で生じた変更がないか、Visual Source Safe ディレクトリーを照会します。</p> <p>プロジェクトおよびファイルに関して、次の変更情報を BOM レポートに書き込みます: プロジェクトまたはファイル、バージョン、ユーザー、日付、時刻、プロジェクト・アクティビティー、ファイル・プロジェクト、およびアクション。</p> <p>変更詳細 (diff コマンド出力による) を BOM レポートに書き込みます。</p>	source

## アダプターおよびプロジェクト

アダプター・コードを実行し、外部アプリケーションと接続するには、Build Forge プロジェクトにアダプターを追加する必要があります。

ドット・コマンドまたはアダプター・リンクを使用して、アダプターをプロジェクトに追加します。

アプリケーション・タイプに対応するドット・コマンド (.source、.defect、.test、または .pack) を使用して、アダプターをプロジェクトに追加することができます。

アダプター・リンクを使用して、ソース・コード・アダプターのみをプロジェクトに追加することができます。 .source コマンドの代わりにアダプター・リンクを使用して、アダプターをプロジェクトに接続します。

## アダプターおよび環境変数

アプリケーション・コマンドを実行するには、アダプターに環境変数が必要です。アダプター・テンプレートで、環境変数は、XML ファイルの <template> セクションにある、<env> エlement にリストされています。

例えば、ClearCaseBaseline アダプターの場合、次の環境変数が ClearCaseBaseline.xml ファイルにリストされています。

```
<template>
<!-- テンプレート・セクション。これらは最後の xml から構文解析されます。
通常の BuildForge ステップでこのインターフェースを統合している場合は、
次のリストを使用して、このインターフェースの実行に必要な変数を特定してください。
-->
<env name="VIEW" value="my_adaptor_view" />
<env name="VOB_PATH" value="%AdaptorVob" />
<env name="CCSERVER" value="BFServerName" /></template>
```

Build Forge では、環境変数は環境変数内に保管されます。アダプターを作成する前に、アプリケーション環境変数の環境を作成します。

## アダプターおよび通知

大部分のアダプター・テンプレートは、ユーザーに E メール通知を送信します。例えば、ClearCaseByDate アダプターの実行時に、ソース・コード・ファイルを変更したユーザーに成功の E メール通知を送信します。ファイルが全く変更されなかった場合、失敗の E メール通知を送信します。

次のように、アダプターの通知を任意に変更することができます。

- アダプター・テンプレートで、<adduser> エlementを複写して、ユーザーをアダプター通知グループに追加します。
- アダプター・テンプレートで、<notify> エlementを使用して、通知メッセージを追加または削除します。
- アダプター・プロジェクトでは、プロジェクト・レベルの通知をセットアップします。
- アダプター・ドット・コマンドのステップでは、ステップ・レベルの通知をセットアップします。

**重要:** <notify> ディレクティブが失敗すると (例えば、E メールの宛先ユーザーが存在しない場合など)、XML が失敗するほか、後続のすべての通知が失敗します。

## アダプターおよびジョブ実行

アダプター・ドット・コマンドを使用するアダプター・プロジェクトは、スケジュールに入れられたジョブとして開始したり、プロジェクトのいずれかの手動開始オプションを使用して開始することができます。

通常、アダプターにリンクされたプロジェクトは、スケジュールに沿って実行されます。ただし、いくつかの追加セットアップを完了すれば、アダプターにリンクされたプロジェクトを手動で開始することができます。 571 ページの『アダプターにリンクされたプロジェクトの手動開始』を参照してください。

## アダプターおよびジョブ結果

アダプター・ドット・コマンド・プロジェクトの場合、次のように、ジョブ結果がステップ・ログまたは BOM レポートに表示されます。

- 「ジョブ」 > 「完了」を選択します。ジョブのタグを選択して、そのステップ・ログを表示します。
- 「ジョブ」 > 「完了」を選択します。BOM タブを選択すると、BOM レポートとカテゴリ別のジョブ結果が表示されます。

アダプターにリンクされたプロジェクトの場合、次のように、ジョブ結果が BOM レポートのソース変更カテゴリーに表示されます。

- 「ジョブ」 > 「完了」を選択します。BOM タブを選択します。BOM レポートで、ソース変更カテゴリーを見つけます。

## アダプター・リンクについて

アダプター・リンクは、アダプターをプロジェクトに接続し、環境変数をアダプターに関連付けます。



アダプター・リンクは、ソース・コード・アダプターでのみ機能します。ドット・コマンドを使用して、あらゆるアダプター・タイプをプロジェクトに接続することができます。

アダプター・リンクは、次の機能を持ちます。

- アダプター・コードをステップ 0 (その他のプロジェクト・ステップを実行する前の、ソース・コード変更に対するテスト) としてプロジェクトに追加する。
- 環境変数のデータをアプリケーション環境変数に自動的に入力する。
- 「アクティブ」、「非アクティブ」、「デバッグ」から状態を選択することによって、アダプター・コードを実行するかどうかを制御することができます。
- アダプターを手動開始ジョブとして実行することができます。

アダプター・リンクには、次の制限があります。アダプター・リンクは、1 つのアダプターおよび 1 つのプロジェクトのみに対して定義されます。別のプロジェクトと同じアダプターを使用する場合は、そのアダプターの別のインスタンスを作成する必要があります。

ジョブを再開した場合、そのジョブの各アダプター・リンク・ステップの再開動作は、以下のようになります。

- アダプター・リンクの「状態」が「非アクティブ」である場合、そのアダプター・リンク・ステップは失敗し、再始動による実行も失敗します。
- アダプター・リンクの「状態」が「アクティブ」で、そのリンクが失敗した場合、そのアダプター・リンク・ステップは失敗し、再始動による実行も失敗します。

上記のいずれの場合にも、ビルドは削除されません。ただし、ビルドの状況は「失敗」に設定されます。

**アダプター・リンク・パネルについて:** アダプター・リンク・パネルを使用して、アダプターをプロジェクトに接続し、環境変数をアダプターに関連付けます。パネルを表示するには、「プロジェクト」 > 「アダプター・リンク」を選択します。

The screenshot shows the 'Adaptor Links' panel in the IBM Rational Build Forge interface. At the top, there are tabs for 'UI Config', 'Console', and 'Reports', along with a 'Logout: Root User' link. The main panel has a header with 'Adaptor Links' and an 'Add Adaptor Link' button. Below this is a filter section with 'Showing 1 - 2 of 2' and a 'Display All' link. The main content is a table with three columns: 'Adaptor', 'Project', and 'State'. There are two rows of data, both with 'Active' state. Below the table, there are buttons for 'New Adaptor Link', 'Save', 'Populate Env', and 'Delete'. At the bottom, there is a 'Details' section with several dropdown menus: 'State' (set to 'Active'), 'Adaptor' (set to 'Adaptor List'), 'Entry Point' (set to 'Default'), 'Project' (set to 'Projects'), and 'Environment' (set to 'None').

Adaptor	Project	State
MyAdaptor	Sav_hi	Active
MyAdaptor	HelloWorld	Active

パネルには、以下のフィールドがあります。

状態 「アクティブ」、「デバッグ」、または「非アクティブ」のいずれか

注: これらの状態は、1 つのアダプター・リンクにのみ適用されます。「デバッグ・モードのリンク」システム設定を使用することで、すべてのアダプター・リンクを「デバッグ」状態に設定できます。この状態は、システム構成設定よりも優先されます。

#### アダプター

プロジェクトにリンクするアダプター

#### エントリー・ポイント

XML インターフェースへのエントリー・ポイント

#### プロジェクト

アダプターを使用するプロジェクト

#### 環境変数

使用するプロジェクトの環境変数

## アダプター・タスクの概説

このトピックは、ソース・コード・アダプターを作成し、それをアダプター・リンクを持つプロジェクトに接続し、そのアダプターにリンクされたプロジェクトをテスト・モードで実行するために必要なすべてのタスクを網羅しています。

## テンプレートの選択によるアダプターの作成

テンプレートを選択してアダプターを作成するには、次のようにします。

1. 「プロジェクト」→「アダプター」と選択します。
2. 「アダプターの追加」をクリックします。
3. 「名前」に、アダプターの固有の名前を入力します。アダプター名は、アダプター・セット全体の中で固有でなければならず、スペースが含まれてはいけません。
4. 「タイプ」で、アダプター・タイプを選択します。
5. 「テンプレート」で、テンプレートを選択します。リストには、Build Forge 製品とともにインストールされたアダプター・テンプレートが含まれています。ClearCase および ClearQuest アダプターでは、別個のライセンス・キーは必要ありません。その他のアダプターは、アダプター・ツールキットを介して別個にライセンスが交付されます。
6. 「アクセス」で、アクセス・グループを選択します。アダプターの表示または編集は、これらのグループ・メンバーしかできません。
7. 「アダプターの保存」をクリックします。

## 空の環境変数の作成

空の環境変数を作成するには、次のようにします。

1. 「プロジェクト」→「環境変数」と選択します。
2. 「環境変数の追加」をクリックします。
3. 「名前」に、環境変数名を入力します。環境変数の目的を示す名前を割り当てます。
4. 「アクセス」で、アクセス・グループを選択します。環境変数の表示または編集は、これらのグループ・メンバーにしかできません。

5. 「環境変数の保存」をクリックします。


## プロジェクトへのアダプターの追加

プロジェクトにアダプターを追加するには、次のようにします。

1. 「プロジェクト」→「アダプター・リンク」と選択します。
2. 「アダプター・リンクの追加」をクリックします。
3. 「アダプター」で、作成したアダプター（およびアダプター・テンプレート）を選択します。
4. 「プロジェクト」で、プロジェクトを選択します。リストには、まだアダプターにリンクされていないプロジェクトが表示されています。
5. 「状態」で、「アクティブ」を選択します。
6. 「環境変数」で、アダプター・リンク用に作成した空の環境変数を選択します。
7. 「環境変数の有効化」で、「はい」を選択します。アダプター・テンプレートのアプリケーション環境変数が、環境変数に追加されます。
8. 「保存」をクリックして、アダプターをプロジェクトにリンクします。アダプターとプロジェクトが、アダプター・リンクのリストに追加されます。

## 環境変数の編集

環境変数を編集するには、次のようにします。

1. 「環境変数」を選択します。
2. 作成した環境変数に対して、「編集」アイコン  をクリックします。パネルに、環境変数に自動的に追加されるアダプター環境変数が表示されます。
3. アダプター・テンプレートで提供された環境変数のデフォルト値を検討します。
4. 必要に応じて、ソース・コード・アプリケーションのデフォルト値を変更し、アダプター・プロジェクトを実行します。

## condition 属性

condition 属性を使用すると、一部のアダプター・プロパティーに、Perl の比較演算子を使用して条件を適用することができます。比較には、ストリング・リテラル、数値、または変数を使用できます。

condition 属性の構文は次のとおりです。

```
condition="true(<lvalue> <operator> <rvalue>)"
condition="false(<lvalue> <operator> <rvalue>)"
condition="hastext(variable)"
condition="isempty(variable)"
```

次の 4 つのタイプのいずれかを指定します。

タイプ	説明
true	左辺値と右辺値が同じ場合に成功と評価されます。
false	左辺値と右辺値が異なる場合に成功と評価されます。
hastext	値の長さが 0 より大きい場合に成功と評価されます。
isempty	値の長さが 0 である場合に成功と評価されます。

左辺値および右辺値には、ストリング、数値、あるいはストリングまたは数値を含む変数のいずれかを使用できます。条件演算子は、任意の Perl 互換条件演算子です。ストリングおよび数値の条件演算子があります。適切な演算子を使用しないと、予測不能な結果を受け取ることになります。

ストリング演算子	数値演算子
eq	==
ne	!=
gt	>
lt	<
ge	>=
le	<=

ストリングに数値演算子を使用した場合、正しい結果は返されません。数値にストリングの比較演算子を使用した場合も同様です。

例: `condition="true("PASS"=="FAIL")`

上記の条件では常に TRUE が返されますが、これは正しくありません。

条件の例:

- `condition="true($BF_SERVER eq "TEST_BOX")` - ビルド・サーバー変数に TEST\_BOX が含まれている場合にのみ、項目を実行します。
- `condition="false($BF_BID <=141)` - ビルド・タグが 141 以上の場合にのみ、ビルドを実行します。

condition をサポートするアダプター・プロパティは、次のとおりです。

- adduser
- bom
- run
- setenv

Build Forge の現行インストールの DTD で、condition 属性をサポートするプロパティについての最新情報がないか二重に確認してください。

condition をサポートするプロパティに対しては、次のような項目があります。

<!ATTLIST adduser condition CDATA #IMPLIED>

アダプターの DTD は、Windows の場合は %BF\_HOME%\interface に、UNIX の場合は \$BF\_HOME/Platform/interface に、interface.dtd ファイルとして存在します。

## アダプターのテスト・モードでの実行

アダプターをテスト・モードで実行するには、次のようにします。

1. 「管理」→「システム」と選択します。
2. システム構成パラメーターのリストから、「デバッグ・モードのリンク」を選択します。

3. 「デバッグ・モードのリンク」で、「はい」を選択します。
4. 「保存」をクリックします。
5. 「ジョブ」→「開始」と選択します。
6. プロジェクト・リストから、「開始プロジェクト」ページで作成した、アダプターにリンクされたプロジェクトを選択します。
7. 「実行」をクリックします。

## ジョブ状況およびログの表示

アダプター・プロジェクトのジョブ状況とログ情報を表示するには、以下を行います。

1. 「ジョブ」を開きます。
2. プロジェクト・リストから、アダプターにリンクされたプロジェクトを見つけて、ジョブの成功または失敗状況を表示します。
3. ジョブ・ログを表示するには、以下を行います。
  - アダプター・プロジェクトのタグ名を選択して、ジョブ・ログ情報にアクセスします。
  - 部品構成表を選択して、BOM レポートにアクセスします。

## コア・アダプター・タスク

このトピックでは、アダプターの作成および構成に関する情報を提供します。

- **テンプレートの選択:** 各アダプター・アプリケーション・テンプレートを説明します。
- **環境変数の作成:** 環境変数とアダプターを関連付けるオプションを説明します。
- **アダプターの作成:** アダプターの作成と、テンプレートへ関連付ける方法を説明します。
- **アダプターのプロジェクトへの追加:** アダプターをプロジェクトへ追加するオプションを説明します。
- **アダプターのテスト:** アダプター構成のみをテストする方法を説明します。

このトピックでは、他のアダプター・タスクに関する情報も提供します。

- **アダプター・ログ・レベルの設定:** アダプターのログに記録される情報量を制御する方法を説明します。
- **アダプターにリンクされたプロジェクトのクイック・スタート:** 手動での開始に必要なセットアップを説明します。
- **アダプター・テンプレートのリセット:** テンプレート情報の更新にリセットが必要な場合を説明します。

## アダプターの環境変数の作成

アプリケーション・コマンドを実行するには、アダプターに環境変数が必要です。アダプター・テンプレートで、環境変数は、XML ファイルの `<template>` セクションにある、`<env>` エlement にリストされています。

アダプター・テンプレート・ファイルでは、環境変数を編集しないでください。Build Forge 製品では、環境変数を使用してアダプターの環境変数を定義します。

既存の環境変数を使用するか、アダプターのみ使用する環境変数を別途作成します。環境変数に特定の使用法および記述名を割り当てることができ、トラブルシューティングも容易になるため、環境変数を作成することをお勧めします。

このトピックでは、環境変数とアダプターを関連付ける次の方法を説明します。

- アダプター・リンクを使用する
- アダプター・ドット・コマンドと関連付けられたステップで行う

**アダプター・リンクの環境変数の作成:** アダプター・リンクを使用して、ソース・コード・アダプターをプロジェクトにリンクする場合は、このメソッドを使用します。

1. 「プロジェクト」 > 「環境変数」を選択します。
2. 「環境変数の追加」をクリックします。
3. 「名前」に、環境変数名を入力します。環境変数の目的を示す名前を割り当てます。
4. 「アクセス」で、アクセス・グループを選択します。環境変数の表示または編集は、グループ・メンバーしかできません。
5. 「環境変数の保存」をクリックします。この時点では、環境変数をグループに追加しないでください。アダプター・リンクを作成するときに、環境変数はアダプター・テンプレートから自動的にデータ入力されます。

**アダプター・ドット・コマンドの環境変数の作成:** アダプター・ドット・コマンドを使用して、アダプターをプロジェクトに追加する場合は、このメソッドを使用します。

このタスクを行うには、アダプター用の環境変数が必要です。

<adaptor\_name>.xml ファイルで、外部環境変数は <template><env> エlement にリストされています。

次のディレクトリーで、アダプター・テンプレートを見つけます。

```
<bfinstall>¥interface
```

1. 「プロジェクト」 > 「環境変数」を選択します。
2. 「環境変数の追加」をクリックします。
3. 「名前」に、環境変数名を入力します。環境変数の目的を示す名前を割り当てます。
4. 「アクセス」で、アクセス・グループを選択します。環境変数の表示または編集は、グループ・メンバーしかできません。
5. 「環境変数の保存」をクリックします。
6. 「環境変数値の追加」をクリックします。
7. 「名前」に、XML <env> Element に表示されている、環境変数名を入力します。
8. 「値」で、XML <env> Element の置換変数を、アプリケーションの実際の値に変更します。(正確な値が分からない場合は、後で入力することもできます。)
9. 「アクション」で、「設定」を選択します。



10. 「プロジェクト動作」で、「通常」を選択します。

## テンプレートからのアダプターの作成

すべてのアダプターは、アダプター・テンプレートを基にしています。アダプターを作成する前に、どのテンプレートを使用するかを決めます。

アダプターを作成するには、以下を行います。

1. 「プロジェクト」 > 「アダプター」を選択します。
2. 「アダプターの追加」をクリックします。
3. 「名前」に、アダプターの固有の名前を入力します。アダプター名は、アダプター・セット全体の中で固有でなければならず、スペースが含まれてはいけません。
4. 「タイプ」で、アダプター・タイプを選択します。
5. 「テンプレート」で、テンプレートを選択します。リストには、Build Forge 製品とともにインストールされたアダプター・テンプレートが含まれています。  
ClearCase および ClearQuest アダプターでは、別個のライセンス・キーは必要ありません。その他のアダプターは、アダプター・ツールキットを介して別個にライセンスが交付されます。
6. 「アクセス」で、アクセス・グループを選択します。アダプターの表示または編集は、これらのグループ・メンバーしかできません。
7. 「アダプターの保存」をクリックします。

## アダプターのプロジェクトへの追加

アダプター・コードを実行するには、アダプターをプロジェクトに追加する必要があります。新規プロジェクトを作成するか、アダプターを既存プロジェクトに追加します。

このセクションでは、次のメソッドを使用して、アダプターをプロジェクトに追加する方法を説明します。

- アダプター・リンクを使用して、ソース・コード・アダプターをプロジェクトに追加することができます。
- アダプター・ドット・コマンドを使用して、すべてのアダプター (ソース・コード・アダプターを含む) をプロジェクトに追加することができます。

**アダプター・リンクによるソース・コード・アダプターの追加:** アダプター・リンクは、アダプターをプロジェクトに接続し、アプリケーション環境変数をアダプターに接続します。

このタスクを開始する前に、アダプター用のプロジェクトと環境変数を作成します。

このタスクが完了した後、アプリケーション環境変数に実際の値をまだ指定していない場合は、環境変数を開いてその値を入力します。

1. 「プロジェクト」 > 「アダプター・リンク」を選択します。
2. 「アダプター・リンクの追加」をクリックします。
3. 「状態」で、次の状態から選択します。



状態	説明
アクティブ	プロジェクト実行時に、アダプター・コードを実行します。
非アクティブ	プロジェクト実行時に、アダプター・コードをスキップします。
デバッグ	アダプター・コードのみを実行します。プロジェクト実行時に、その他のステップをスキップします。

4. 「アダプター」で、アダプター・テンプレートを選択します。リストには、Build Forge 製品とともにインストールされているアダプター・テンプレートが表示されています。
5. 「プロジェクト」で、プロジェクトを選択します。リストには、まだアダプターにリンクされていないプロジェクトが表示されています。
6. 「保存」をクリックして、アダプターをプロジェクトにリンクします。アダプター名がリストに追加されます。
7. 「環境変数」で、アダプター・リンク用の環境変数を選択します。
8. 「環境変数の有効化」で、「はい」を選択します。 アダプター・テンプレートのアプリケーション環境変数が、環境変数に追加されます。
9. 「保存」をクリックして、アダプター・リンクを保存します。

**ドット・コマンドを使用したアダプターの追加:** アダプターのドット・コマンドを使用して、アダプターをプロジェクトに追加することができます。ステップの実行時に、ドット・コマンドは `<adaptor_name>.xml` ファイルを呼び出します。

このタスクを開始する前に、アダプター用のプロジェクトと環境変数を作成します。

このタスクが完了した後、アプリケーション環境変数に実際の値をまだ指定していない場合は、環境変数を開いてその値を入力します。

次のように、アダプター・ドット・コマンドをステップとしてプロジェクトに追加します。

1. 「ライブラリー」を選択します。
2. リストからプロジェクトを選択します。
3. 「ステップの追加」をクリックします。
4. 「名前」に、ステップ名を入力します。
5. 「コマンド」に、アプリケーション・タイプに対応するアダプター・ドット・コマンド (`.source`、`.defect`、`.test`、`.pack`) を入力します。
6. 「環境変数」で、アダプター用に作成した環境変数を選択します。
7. 「ステップの保存」をクリックします。

## アダプターのテスト

アダプター構成をテストするには、アダプター・プロジェクトを実行します。アダプターが外部アプリケーションと対話でき、期待された結果を返すことを確認するために、他のプロジェクト・ステップとは分離してアダプター・コードを実行します。

このトピックでは、次のアダプターのテスト方法について説明します。

- アダプター・リンクを介してプロジェクトに追加されたアダプター
- アダプター・ドット・コマンドを使用してプロジェクトに追加されたアダプター

**リンクされたアダプターのテスト:** ソース・コード・アダプターが、アダプター・リンクを通じてプロジェクトにリンクされている場合は、次の手順に従ってアダプター構成をテストしてください。

一般的な手順は、次のとおりです。

1. ソース・ファイルに変更を加える。
2. リンク・アダプターを使用して Build Forge プロジェクトを実行する。
3. 変更されたソース・ファイルについての情報があるか、BOM レポートを確認する。
4. 成功または失敗の通知があるか、E メールを確認する。

リンクされたアダプターをテストするには、以下のようにします。

1. 「プロジェクト」 > 「アダプター・リンク」を選択します。
2. リストで、リンクされたアダプターおよびプロジェクトを選択します。
3. 「状態」で「デバッグ」を選択します。
4. 「保存」をクリックします。
5. ソース・コード・アプリケーションで、1 つ以上のソース・ファイルに変更を加えます。変更を送信して、ソース・コード・リポジトリを更新します。
6. 次のように、アダプターとリンクされたプロジェクトを実行します。
  - a. 「管理」 > 「システム」を選択します。
  - b. リストで、「デバッグ・モードのリンク」を選択します。
  - c. 「デバッグ・モードのリンク」で、「はい」を選択します。
  - d. 「保存」をクリックします。
  - e. 「ジョブ」 > 「開始」を選択します。
  - f. プロジェクトのリストで、「プロジェクトの開始 (Start Project)」ページからアダプター・リンク・プロジェクトを選択します。
  - g. 「実行」をクリックします。
7. 次のようにして、ジョブの BOM レポートを検討します。
  - a. 「ジョブ」を開きます。
  - b. 「完了」タブを選択し、「BOM」タブを選択します。

**アダプター・ドット・コマンドのテスト:** ドット・コマンドを使用してプロジェクトに追加したアダプターについては、次のオプションの手順に従ってアダプター構成をテストします。

一般的な手順は、次のとおりです。

1. ソース・ファイルに変更を加える。
2. アダプター・ドット・コマンドを使用して Build Forge プロジェクトを実行する。
3. 変更されたソース・ファイルについての情報があるか、BOM レポートを確認する。

4. 成功または失敗の通知があるか、E メールを確認する。

アダプター・ドット・コマンドをテストするには、次の手順を実行してください。

1. 「ジョブ」 > 「開始」を選択します。
2. リストからプロジェクトを選択します。
3. 「ジョブ・ステップ」タブを開きます。
4. 「ステップ名」チェック・ボックスを使用して、アダプター・ドット・コマンド以外のすべてに対し、チェック・マークを外します。
5. 「実行」をクリックして、プロジェクトを実行します。
6. 次のようにして、ジョブの BOM レポートを検討します。
  - a. 「ジョブ」を選択します。
  - b. 「完了」タブを選択し、「BOM」タブを選択します。

## アダプター・ログ・レベルの設定

アダプターのステップ・ログに書き込まれる情報量を制御するには、`_InterfaceLoggingLevel` 環境変数を使用します。

1. `_InterfaceLoggingLevel` をアダプターの環境変数に追加します。
    - レベル 8 では大部分の情報をログに記録し、レベル 0 では最低限の情報をログに記録します。
    - ログ・レベルは包括的です。例えば、レベル 2 は、レベル 1 および 0 の情報を含みます。
    - デフォルトのログ・レベルは、レベル 7 です。
  2. `_InterfaceLoggingLevel` 変数の値として、次のログ・レベルを割り当てます。
- 0:** 実行する行およびサーバー接続エラー、またはキャンセル通知。他はなし。
- 1:** 構文解析済みのコマンド (サーバーに送信されるコマンド)
- 2:** 構文解析済みでないコマンド (ローカル変数が設定される前のコマンド)
- 3:** ビルドおよび環境変数の SET 行
- 4:** 一時および内部変数 SET 行
- 5:** 環境変数の評価、E メール・グループの追加、BOM テキスト・ロギングの行
- 6:** ブロックおよびサブブロックの開始/終了の行
- 7:** (デフォルトのロギング・レベル) 一致パターンに対するチェックが行われるエージェント出力、およびパターンに一致した行。
- 8:** すべてのエージェント出力

## アダプターにリンクされたプロジェクトの手動開始

アダプターにリンクされたプロジェクトは、スケジュールに沿って実行したり、クイック・スタート・オプションを使用して実行したり、「実行リンク」チェック・

ボックスにチェック・マークを付けた場合は手動で開始したりすることができます。「実行リンク」にチェック・マークを付けなければ、プロジェクトはアダプター・ステップなしで実行されます。


1. 「ジョブ」 > 「開始」を選択します。
2. プロジェクト・リストで、アダプターにリンクされたプロジェクトの名前を選択します。

「開始プロジェクト」ページが開きます。

3. 「実行リンク」チェック・ボックスにチェック・マークを付けます。
4. 「開始プロジェクト」ページで、「実行」をクリックします。

## アダプターにリンクされたプロジェクトのクイック・スタート

アダプターにリンクされたプロジェクトは、スケジュールに沿って実行したり、「プロジェクトの開始」ページ（「ジョブ」→「開始」）からプロジェクト名を選択して手動で開始したり、「即時開始を使用可能に設定」システム構成を設定した場合は「クイック・スタート」アイコンを使用して実行したりすることができます。次のステップでは、「即時開始を使用可能に設定」が設定済みであると見なします。

1. 「管理」 > 「システム」を選択します。
2. リストから、「マニュアル・ジョブのリンク」を選択します。
3. 「マニュアル・ジョブのリンク」で、「はい」を選択します。
4. 「保存」をクリックします。
5. 「ジョブ」 > 「開始」を選択します。
6. プロジェクトのリストで、アダプターにリンクされたプロジェクトの「クイック・スタート」アイコン  を選択すると、そのプロジェクトが即座に開始されます。ジョブが実行されているかどうかを確認するには、「ジョブ」 > 「実行中」タブを選択します。

## アダプター・テンプレートのリセット

### このタスクについて

アダプター・テンプレートをリセットすると、最新のアダプター・テンプレートが <bfinstall>%interface ディレクトリーから Build Forge データベースにコピーされます。

次の場合は、必ずアダプター・テンプレートをリセットしてください。

- 保守リリースまたは新規製品バージョンをインストールする場合
- (管理コンソール内ではなく) インターフェース・ディレクトリー内でバージョンを編集して、テンプレートを変更する場合
- 新規アダプター・テンプレートを作成する場合

### 手順

1. 「管理」 > 「システム」を選択します。
2. リストから、「アダプター・テンプレートのリセット」を選択します。
3. 「アダプター・テンプレートのリセット」で、「はい」を選択します。

4. 「保存」をクリックします。

## ClearQuest ビルド・レコードの更新

システムを Rational ClearQuest と統合すると、ClearQuest データベース内のビルド・レコードを自動的に作成および更新することができます。

IBM Rational® ClearQuest® データベース内に、ログ・データへのリンクを含むビルド・レコードが自動的に作成されます。さらに、ジョブが成功すると、システムは ClearQuest データベースを更新し、ジョブが完了したことを示してから、完遂したステップの終了時刻および要約を記録します。この機能には、Rational ClearQuest バージョン 7.0 以降が必要となります。

**注:** ビルド・レコードを正常に作成し、そのレコードにビルドに関する情報を追加するには、まず ClearQuest メンテナンス・ツールを開き、データベースへの接続をセットアップする必要があります。

ClearQuest データベースを更新するようにプロジェクトを構成すると、システムは以下を実行します。

### ジョブの開始

システムは、ジョブを起動すると、ClearQuest ビルド・レコードを作成します。ビルド・レコードは「実行依頼済み (Submitted)」状態になり、「ビルド XYZ が開始しました」というログ・エントリーに加えて、ジョブ・ログの URL、開始時刻、リリース名、および ID がレコードに含まれます。ソース制御アダプターがジョブをキャンセルすると (例えば、ソースの変更がなかったため)、ClearQuest ビルド・レコードは作成されません。

**注:** あるプロジェクトが別のプロジェクトをチェーニングすると、新規プロジェクトはそれ固有の ClearQuest ビルド ID を取得します。

### ジョブの成功または失敗

ジョブが成功または失敗すると、ClearQuest 内のビルド状態が「完了」または「失敗」に変更され、ビルド終了時刻が設定され、ClearQuest ビルド・ログにジョブのステップの要約が保管されます。要約には、各ステップの名前、結果状況、およびサーバーが含まれます。

### ジョブの再開

ジョブを再開すると、ClearQuest 内のビルド状態が「実行依頼済み (Submitted)」になり、「XYZ のビルドが再開しました」という内容の ClearQuest ビルド・ログ・エントリーが作成されます。

特定の環境変数を使用して、ClearQuest の統合を構成します。プロジェクトを ClearQuest データベースとリンクするには、下表の変数がプロジェクトの環境変数に組み込まれていることを確認してください。

**注:** これらの変数はプロジェクト環境変数内に存在する必要があります。ステップに追加するだけでは十分ではありません。ただし、別の環境変数を介してこれらの変数を含める「組み込み」タイプに設定された変数を使用することができます。また、プロジェクトごとに変わる可能性があるのは CQ\_RELEASE\_NAME の値のみであるため、その他の変数を含む環境変数を作成し、「組み込み」型の変数を

使用してプロジェクト環境変数内にその環境変数を含めることができます。その場合、CQ\_RELEASE\_NAME をプロジェクト固有の環境変数として指定することもできます。

表 17. Rational ClearQuest の統合に必要な環境変数

変数	説明
CQ_DBNAME	更新する ClearQuest データベースの名前。
CQ_INTERACTION	プロジェクト環境変数に定義済みの正しい環境変数があり、ClearQuest ビルド・レコードの作成が使用可能になっているが、ビルド・レコードを作成したくない場合は、この変数を OFF に設定して、ビルド・レコードの作成を使用不可にします。  ビルド・レコードの作成を使用可能にするには、この環境変数を ON に設定します。  <b>注:</b> ClearQuest アダプターのいずれかを使用している場合は、この環境変数を OFF に設定してください。
CQ_USER	ClearQuest データベースにログインするときに使用するユーザー名です。
CQ_RELEASE_NAME	ClearQuest データベース内にある更新するリリースの名前です。
CQ_PASSWORD	ClearQuest データベースにログインするときに使用するパスワードです。必須ではなく、デフォルトはブランクです。
CQ_DBSET	ClearQuest データベースの設定値です。必須ではなく、デフォルトはブランクです。

## 拡張アダプター・タスク

Build Forge アダプター・テンプレートを変更したり、Build Forge プロジェクトと接続する外部アプリケーション用の新規アダプター・テンプレートを作成したりすることができます。

テンプレートの変更および作成は、次の作業知識を必要とする拡張タスクです。

- XML 言語
- 外部アプリケーションのコマンド言語
- 正規表現

拡張タスクで作業する前に、次のセクションの情報をお読みください。

- 566 ページの『コア・アダプター・タスク』
- 582 ページの『アダプター・テンプレートの構造』
- 580 ページの『アダプター参照』

このセクションでは、次のタスクに関する情報を得ることができます。

- **アダプター・テンプレートの変更:** すべてのアダプターに対するテンプレートの変更方法を説明します。
- **単一アダプターのテンプレートの変更:** 1 つのアダプターに対するテンプレートの変更方法を説明します。
- **新規アダプター・テンプレートの作成:** 新規テンプレートを作成する一般的な手順を説明します。



- **例: アダプター通知へのユーザーの追加:** E メール通知へのアクセス・グループの追加方法を説明します。
- **例: BOM レポートからの変更詳細の削除:** BOM レポートから変更詳細を削除する方法を説明します。

## アダプター・テンプレートの変更

アダプター・テンプレートから作成される今後のすべてのアダプターで、テンプレート修正を選択するようにする場合は、この手順に従ってください。

始める前に、何を変更するかを把握します。例えば、通知を変更したり、BOM レポートのフォーマットを変更したりすることができます。

1. XML エディターを使用して、変更するアダプター・テンプレートを開きます。アダプター・テンプレートは、以下のディレクトリーにあります。

`bfinstall¥interface` (Microsoft Windows オペレーティング・システムを実行しているコンピューターの場合)

`bfinstall/Platform/interface` (UNIX または Linux オペレーティング・システムを実行しているコンピューターの場合)

2. テンプレートの変更を入力します。
3. アダプター・テンプレートを保存します。テンプレート名は、変更しないでください。アダプターおよびアダプター・テンプレートには固有の名前を付ける必要があります。

変更したアダプター・テンプレートを使用して次にアダプターを作成するときに、新しいアダプターにテンプレートへの変更が含まれるようになります。

## 単一アダプターのテンプレートの変更

- 1 つのアダプターの単一インスタンスのみのテンプレートを変更するときは、この手順に従います。

始める前に、何を変更するかを把握します。例えば、通知を変更したり、BOM レポートのフォーマットを変更したりすることができます。

**変更するアダプターが作成されている場合、**次のように、管理コンソールを使用してテンプレートへの変更を行います。

1. 「プロジェクト」 > 「アダプター」を選択します。
2. テキスト・ボックスで、テンプレートへの変更を入力します。
3. 「アダプターの保存」をクリックします。

**注:** 変更は、Build Forge データベースのアダプターと関連付けられているアダプター・テンプレートのインスタンスのみに保存されます。変更は、`interface` ディレクトリーのアダプター・テンプレート・ファイルへは保存されません。

**変更するアダプターが作成されていない場合、**アダプターを作成する前に、次のようにテンプレートを変更します。

1. XML エディターを使用して、アダプター・テンプレートを開きます。アダプター・テンプレートは、以下のディレクトリーにあります。



*bfinstall¥interface* (Microsoft Windows オペレーティング・システムを実行しているコンピューターの場合)

*bfinstall/Platform/interface* (UNIX または Linux オペレーティング・システムを実行しているコンピューターの場合)

2. テンプレートに変更を入力します。
3. アダプター名を変更して、アダプター・テンプレートを保存します。

注: アダプターおよびアダプター・テンプレートには固有の名前を付ける必要があります。

4. アダプター・テンプレートをリセットして、変更したテンプレートを選択し、それを管理コンソールの使用可能なテンプレートのリストに追加します。 572 ページの『アダプター・テンプレートのリセット』を参照してください。

## 新規アダプター・テンプレートの作成

この手順に従って、Build Forge プロジェクトと接続する外部アプリケーションの新規アダプター・テンプレートを作成します。

1. Build Forge 製品から提供されるアダプター・テンプレートの XML 構造およびエレメントを検討します。アダプター・テンプレートは、以下のディレクトリにあります。

*bfinstall¥interface* (Microsoft Windows オペレーティング・システムを実行しているコンピューターの場合)

*bfinstall/Platform/interface* (UNIX または Linux オペレーティング・システムを実行しているコンピューターの場合)

2. 新規アダプター利用して、次のどれを行うかを計画します。
  - どのコマンドを実行するかを認識する
  - コマンドがどのように構文解析されるかを決定する
  - 構文解析の結果から収集されたデータで何を行うかを決定する
  - どの外部環境変数が必要かを認識する
3. モデルとして使用するアダプター・テンプレートを、インターフェース・ディレクトリの中から 1 つ選択します。可能であれば、同じ外部アプリケーションを基にしているアダプターを選択します。または、作成しているものと同様の機能を持つアダプターを選択します。モデル内の XML 階層、エレメント、およびエレメント属性をガイドとして使用して、新規テンプレートを作成します。
4. XML エディターを使用して、モデル・アダプター・テンプレートを開きます。テンプレートに新規名をつけて、インターフェース・ディレクトリに保存します。
5. ユーザーの計画を使用して、新規アダプターの XML コードを作成します。
6. 新規アダプター・テンプレートを保存します。
7. インターフェース・ディレクトリの *interface.dtd* ファイルを使用して、アダプター・テンプレートを検証します。

8. アダプター・テンプレートをリセットして、新規テンプレートを選択し、それを管理コンソールの使用可能なテンプレートのリストに追加します。 572 ページの『アダプター・テンプレートのリセット』を参照してください。
9. アダプターのプロジェクトを作成します。
10. アダプターの環境変数を作成します。 566 ページの『アダプターの環境変数の作成』を参照してください。
11. 新規アダプター・テンプレートを使用して、アダプターを作成します。 568 ページの『テンプレートからのアダプターの作成』を参照してください。
12. アダプターをプロジェクトに追加します。 568 ページの『アダプターのプロジェクトへの追加』を参照してください。
13. アダプター・プロジェクトを実行して、アダプターをテストします。 569 ページの『アダプターのテスト』を参照してください。

## 複数エントリー・ポイント・アダプターの作成

Build Forge で提供されるアダプター・テンプレートは、単一エントリー・ポイント・アダプターです。

単一エントリー・ポイント・アダプターの場合、管理コンソール（「プロジェクト」 > 「アダプター」）で、実行するアプリケーションおよび機能のテンプレート名を選択します。例えば、ClearCaseBaseline または ClearCaseByDate などです。

必要に応じて、ClearCase 用の複数のインターフェースまたは機能を含む、ClearCase 用のアダプター・テンプレートを 1 つ作成できます。複数エントリー・ポイント・アダプターの場合、エントリー・ポイントと呼ばれる名前によって各インターフェースを識別します。

複数エントリー・ポイント・アダプターを作成するには、以下のようにします。

1. アダプター・テンプレートを作成します。テンプレートを作成する際に、モデルとして提供されるテンプレートの 1 つを使用して、必要に応じて XML を変更することができます。テンプレートでは、<interface> エlement に名前属性を追加して、テンプレートに追加する各インターフェースのエントリー・ポイントを識別する必要があります。次の例に、<インターフェース>Element に関連する構文を示します。

```
<interface name="By Date" default="true">
</interface>
```

2. スペースが含まれていない固有の名前でアダプターを作成し、それをアダプター・テンプレートに関連付けます。 568 ページの『テンプレートからのアダプターの作成』を参照してください。
3. アダプター・ドット・コマンドまたはアダプター・リンクを使用して、アダプターをプロジェクトに追加します。
  - 次の例では、.source アダプター・コマンドを使用してアダプターをプロジェクト・ステップに追加し、そのプロジェクト・ステップでは、ClearCase アダプターの By Date インターフェース機能呼び出します。

```
.source ClearCase "By Date"
```

- アダプター・リンクを使用して複数エントリー・ポイント・アダプターを呼び出すには、アダプター・テンプレートで次のいずれかのアクションを取り、プロジェクトの実行時にどのインターフェース機能を実行するかを指定します。

- 実行する機能の<インターフェース>・エレメント定義を、テンプレート・ファイルの最初の<インターフェース>・エレメントとして配置します。
- 実行する機能の <interface> エレメントの default 属性を true に設定します (default="true")。

## 例: E メール通知の使用可能化

アダプター・テンプレートを構成して、外部アプリケーションで変更を行うユーザーに E メール通知を送信することができます。次の例は、2 つのタイプの通知のセットアップ方法を示しています。

- 現行ビルドでファイルをチェックインしたすべてのユーザーに通知する
- Build Forge アクセス・グループのすべてのメンバーに通知する

次の手順は、ClearCaseByDate テンプレートにあるエレメントを参照しています。どのアダプター・テンプレートもエレメントを使用して、通知を使用可能に設定することができます。

### 現行ビルドでファイルをチェックインしたすべてのユーザーに通知する

<adduser> コマンドを使用して、ビルドのためにコードをチェックインしたユーザーのグループを動的に作成し、<notify> コマンドを使用して、そのグループに通知を送信することができます。

ClearCaseByDate テンプレートは、2 つのタイム・スタンプの間に変更がないか、ClearCase ビューを照会します。デフォルトのタイム・スタンプは、現行のアダプター実行と直前のアダプター実行を対象とします。実際には、現行ビルドでチェックインされる直前のビルド以降の、すべての変更のリストに変換します。

前提: ビューのすべてのユーザー名は、その名前での通知に使用する SMTP サーバーによって認識されている必要があります。つまり、ClearCase のユーザー名は、Eメールのユーザー名と揃える必要があります。

この通知を使用可能にするには、次を行います。

1. XML エディターで、ClearCaseByDate アダプター・テンプレートを開きます。
2. <adduser> を見つけて編集し、次のようにユーザー・グループを作成します。

```
<adduser group="MyChangers" user="$4">
```

定位置パラメーター \$4 は、ClearCaseByDate テンプレートで生成された ClearCase ビューに表示されるユーザー名フィールドを参照します。

3. このグループに E メールを送信するように、通知をセットアップします。次のセットアップでは、プロジェクトの失敗時と成功時の両方において、E メールが送信されます。一部の環境では、ビルドに失敗した場合にのみ通知するよう設定することができます。

```
<!-- ビルドが完了したときの通知をいくつか設定します -->
<onproject result="fail">
  <notify group="MyChangers" subject="Build $BF_TAG ($CurDate) Failed."
  message="{Changing}{Changes}"/>
</onproject>
<onproject result="pass">
  <notify group="MyChangers" subject="Build $BF_TAG ($CurDate) Passed." message="{Changing}{Changes}"/>
</onproject>
```

4. アダプター・テンプレートを保存します。

アダプターが実行している場合、MyChangers グループはビュー内のユーザー名から作成されます。ビルド・プロジェクトが完了すると、E メール通知がそのグループに送信されます。

### Build Forge アクセス・グループに属するすべてのユーザーに通知する

この例では、Build Forge アクセス・グループのすべてのメンバーに通知します。この例には、ClearCaseByDate アダプター・テンプレートを使用します。前提: Build Forge 内のすべてのユーザー名は、SMTP サーバーの E メール・ユーザー名に対応しています。

1. XML エディターで、ClearCaseByDate アダプター・テンプレートを開きます。
2. <adduser> を見つけて編集し、次のように Build Forge アクセス・グループからユーザー・グループを作成します。

```
<adduser group="Developer_Access_Group" user="Developer">
```

3. このグループに E メールを送信するように、通知をセットアップします。次のセットアップでは、プロジェクトの失敗時と成功時の両方において、E メールが送信されます。一部の環境では、ビルドに失敗した場合にのみ通知するよう設定することができます。

```
<!-- ビルドが完了したときの通知をいくつか設定します -->
<onproject result="fail">
  <notify group="Developer_Access_Group" subject="Build $BF_TAG ($CurDate) Failed."
message="{Changing}{Changes}"/>
</onproject>
<onproject result="pass">
  <notify group="Developer_Access_Group" subject="Build $BF_TAG ($CurDate) Passed."
message="{Changing}{Changes}"/>
</onproject>
```

4. アダプター・テンプレートを保存します。

アダプターが実行している場合、Developer\_Access\_Group グループは、その開発者アクセス・グループに属するユーザー名から作成されます。ビルド・プロジェクトが完了すると、E メール通知がそのグループに送信されます。

**重要:** <notify> ディレクティブが失敗する場合 (例えば、Eメールの宛先のユーザーが存在しない場合)、.xml が失敗し、それに続くすべての通知が失敗します。

### 例: BOM レポートからの変更詳細の削除

大部分のアダプター・テンプレートは、変更詳細を BOM レポートにログ記録します。(変更詳細のログ記録には、diff コマンドが使用されます。)

次のステップは、ClearCaseByDate テンプレート内のエレメントを参照しますが、すべてのアダプター・テンプレートに対する変更詳細を削除するために使用することができます。

BOM レポートの変更詳細を削除するには、以下のようにします。

1. XML エディターで、アダプター・テンプレートを開きます。
2. diff コマンドを呼び出す <run> エレメントを見つけて、次の行を削除します。

```
<run command="cc_diff" params="$VIEW $1 $2" server="$CCSERVER" dir="/" timeout="360"/>
```

3. diff コマンドの <command> エレメントを見つけます。次の行を削除します。

```
<!-- The cc_diff command does a generic clearcase diff, logging the full output
of the diff in the BuildForge BOM -->
<command name="cc_diff">
<execute>
pushd %view%$1 && cleartool diff -pred -diff_format "$2@@"$3"
</execute>
<resultsblock>
<match pattern=".">
<bom category="Source" section="diff">
<field name="diff" text="$_" />
</bom>
</match>
</resultsblock/>
</command>
```

4. <bomformat> セクションを探して、diff コマンド出力の <section> エレメントを見つけます。次の行を削除します。

```
<section name="diff">
<field name="diff" title="Change Details"/>
</section>
```

5. アダプター・テンプレートを保存します。

## アダプター参照

アダプターは、Build Forge プロジェクトに追加して、変更なしで実行するように設計されています。アダプターを変更したり、新規アダプターを作成したりするには、Build Forge アダプター・テンプレートで 사용되는 XML テンプレート構造およびエレメントを理解する必要があります。

**注:** このセクションでは、Build Forge アダプター・テンプレートで 사용되는外部アプリケーション・コマンドに関しては説明しません。これらのコマンドに関する情報は、外部アプリケーションの資料を参照してください。

Build Forge 製品とともにインストールされるアダプター・テンプレートは、次のディレクトリーにあります。

*bfinstall*\%interface (Microsoft Windows オペレーティング・システムを実行しているコンピューターの場合)

*bfinstall*/Platform/interface (UNIX または Linux オペレーティング・システムを実行しているコンピューターの場合)

このセクションでは、次の参照情報を提供します。

- **アダプター要件:** アダプターの使用に関する一般的な要件と、ClearQuest アダプター・テンプレートに関する特定の要件を説明します。
- **アダプターのドット・コマンド:** アダプター・ドット・コマンドの構文を説明します。
- **ClearCase および ClearQuest の環境変数:** ClearCase および ClearQuest アダプターで 사용되는環境変数を説明します。
- **Perforce の環境変数:** Perforce に必要ないくつかの追加環境変数を説明します。
- **アダプター・テンプレートの構造:** Build Forge アダプター・テンプレートの一般的な構造を説明します。

- **アダプター XML 参照:** Build Forge アダプター・テンプレートで使用される XML エlementを説明します。

## アダプターのドット・コマンド

いくつかのドット・コマンドにより、外部アプリケーションのアダプターを、プロジェクト・ステップとして Build Forge プロジェクトに追加することができます。

- **.source:** ソース・コード・アプリケーションのアダプターをプロジェクトに追加します。
- **.defect:** 不良トラッキング・アプリケーションのアダプターをプロジェクト・ステップに追加します。
- **.test:** テスト・アプリケーションのアダプターをプロジェクト・ステップに追加します。
- **.pack:** パッケージ化アプリケーションのアダプターをプロジェクト・ステップに追加します。

434 ページの『ドット・コマンドのリファレンス』も参照してください。

## Rational ClearCase および Rational ClearQuest の環境変数

Rational ClearCase および Rational ClearQuest アダプター・テンプレートでは、下の表にある環境変数を使用して cleartool コマンドを実行します。

下の表にある環境変数すべてが、各 ClearCase または ClearQuest アダプター・テンプレートで必要とは限りません。

各アダプター・テンプレートで、必要な環境変数は、<template> セクションの <env> Element にリストされます。

ClearCase または ClearQuest アダプター・プロジェクトを実行する前に、必須の変数に実際の値を指定するか、デフォルトを受け入れてください。アダプターに割り当てられた環境変数の変数値を編集します。

変数は、プロジェクトの環境内になければなりません。これらはステップ・レベルでは機能しません。

アダプター・テンプレートは、以下のディレクトリーにあります。

*bfinstall*¥interface (Microsoft Windows オペレーティング・システムを実行しているコンピューターの場合)

*bfinstall*/Platform/interface (UNIX または Linux オペレーティング・システムを実行しているコンピューターの場合)

表 18. ClearCase および ClearQuest の環境変数

環境変数名	代替変数	説明
VIEW	値=my_adaptor_view	この変数を、アダプターで使用する ClearCase ビューの名前に設定します。
VOB_PATH	値=¥c_vob	この値を、コンポーネント VOB の名前 (オプションで、そのサブディレクトリー) に設定します。



表 18. ClearCase および ClearQuest の環境変数 (続き)

環境変数名	代替変数	説明
PROJECT_VOB	値=¥ProjectVob	ByBaseline アダプターを使用する場合、この変数を、プロジェクト VOB (UCM ClearCase のみで使用される) の名前に設定します。
CCSERVER	値=BFServerName	この変数を、ClearCase がインストールされ実行されている Build Forge サーバーの名前に設定します。
CurDate	値=.date %d-%b-%y.%H:%M:%S	現在の日付をアダプターに指定し、.date コマンドを使用して、ClearCase が予期する書式で日付を生成します。この値は変更しないでください。
LAST_RUN	値=1-Jan-05.00:00:00	ByDate アダプターの場合、システムはこの値を使用して、何らかの変更が発生したか判別します。この値は、最後に正常に実行されたときの日付です。アダプターのテストの際には、この値を操作し、変更より前であると分かっている日付を取り出して、アダプターを強制的に実行させることができます。アダプターが実行の継続を許可した場合、この値は、現在の日付に自動的に更新されます。デフォルト値は、1-Jan-05.00:00:00 です。
LABEL	値=BUILD_1	ByLabel アダプターの場合、ご使用のアダプターを使用して、(ByLabel アダプターに付く) ラベルごとに違いが生じると、システムは、この値をラベルとして使用します。
BASELINE	値=BUILD_1	ByBaseline アダプターの場合、ご使用のアダプターを使用して、ベースラインごとに違いが生じると、システムは、この値をベースラインとして使用します。
ACTIVITIES	値=SAMPL0001@ ¥ProjectVob	ClearQuestClearCaseByActivity アダプターの場合、スペースで区切られた、アクティビティ ID のセット。

## Perforce 環境変数

次の Perforce 環境変数を、Perforce アダプターに割り当てられた Build Forge 環境変数グループに追加します。

- P4USER
- P4PASSWD

Perforce サーバーにアクセスするには、Build Forge で有効なユーザー名とパスワードが必要です。ステップ・ログに、Perforce ユーザー名とパスワードがプレーン・テキストで書き込まれます。

環境変数の「非表示の割り当て」プロパティを使用して、Perforce 認証情報を暗号化することはできません。

## アダプター・テンプレートの構造

このトピックでは、Build Forge アダプター・テンプレート内の一般的な XML 構造またはエレメント階層について説明します。

アダプター・テンプレートは、次のセクション・エレメントから構成されています。<template>、<interface>、<command>、および <bomformat>。これらの各セクションには子エレメントが含まれています。



エレメントの説明については、『アダプター XML 参照』を参照してください。

```
<PROJECT_INTERFACE>
<template>
<env/>
</template>

<interface>
<setenv/>
<run/>
<ontempenv>
<step/>
</ontempenv>
<onproject>
<notify/>
</onproject>
</interface>

<command>
<execute> または <command>
command line
</execute> または </command>
<resultsblock>
<match>
<bom>
<field/>
</bom>
<adduser/>
<setenv/>
<run/>
</match>
</resultsblock>
</command>

<bomformat>
<section>
<field/>
</section>
</bomformat>
</PROJECT_INTERFACE>
```

## アダプター XML 参照

このセクションでは、アダプターの XML エレメントをアルファベット順にリストします。このセクションは、アダプター XML 言語で使われるエレメントについてのリファレンスです。説明には、いくつかの例および疑似コードが含まれます。

### adduser

<adduser> エレメントは、変更コマンドの出力に基づいてユーザーを一時グループに追加するために、<interface>エレメントまたは <match> エレメント内で使用します。これにより、アダプターは、変更を行ったユーザーに通知を送信できるようになります。システムは、ユーザーがすでにグループのメンバーである場合、複数の通知が送信されるのを防ぐため、グループにそのユーザーを追加しません。

<adduser> エレメントは、空のエレメントです。グループ属性は、アダプター・ロジックの実行中に作成された一時グループを指定します。<notify> エレメント内の同じグループを参照して、実際の通知を送信させる必要があります。

```
<adduser group="MyChangers" user="$4"/>
```

condition="function" 属性を使用して、変更を行ったユーザーを <adduser> エレメントによって一時アクセス・グループに追加するかどうかを制御します。function の値

は、結果が TRUE または FALSE に評価される式です。式の評価結果が TRUE になる場合、ユーザーは一時グループに追加されます。それ以外の場合、ユーザーは追加されません。421 ページの『条件関数』を参照してください。

以下の E メール・アドレス・フォーマットがサポートされています。*username* は、Eメールのユーザー名です。*name* は、任意のストリング (通常、何らかの形で名前を表すもの) です。

```
username@host.com
name username@host.com
name "<username@host.com>"
name <username@host.com>
name "username@host.com"
```

## bom

<bom> エレメントは、ジョブの部品構成表 (BOM) に記録する情報を定義します。このエレメントは、<match> エレメントで囲む必要があります。<bom> エレメントでは、BOM 内のカテゴリとセクションを指定する必要があります。また、このエレメントは、<match> エレメントによって収集されたどの番号の変数 (\$1...\$n) を、BOM データのフィールドに変換するかを定義します。

```
<bom category="Source" section="changesets" >
  <field name="Change" text="$1"/>
  <field name="Date" text="$2"/>
  <field name="User" text="$4"/>
</bom>
```

condition="function" 属性を使用して、<bom> エレメントを BOM レポートに書き込むかどうかを制御します。function の値は、結果が TRUE または FALSE に評価される式です。式の評価が TRUE になる場合、<bom> エレメントの情報は BOM に書き込まれます。TRUE にならない場合は、書き込まれません。421 ページの『条件関数』を参照してください。

## bomformat

<bomformat> エレメントは、前述の <bom> エレメントで収集されたデータの表示方法を定義します。このエレメントは、BOM カテゴリの論理名を指定する category 属性と、カテゴリの表示名を指定する title 属性を取ります。<field> エレメントを含む <section> エレメントの構造には、レイアウトが定義されます。以下に例を示します。

```
<bomformat category="Detail" title="Change Details">
  <section name="descriptions" parent="section name" expandable="yes">
    <field name="Description" title="Change Description"/>
  </section>
  <section name="diff">
    <field name="Diff" title="Differences"/>
  </section>
```

## command

アダプター XML ファイルには、複数の <command> エレメントを入れることができます。各エレメントには、<interface> エレメント内の <run> エレメントによって参照可能な名前付きコマンドが定義されています。<command> エレメントは、<interface> エレメントの外に指定されているため、XML ファイル内の複数のインターフェースが同じコマンドを再利用できます。

<run> コマンドを、<resultsblock> エlement内の <match >Elementに組み込むと、コマンドで他のコマンドを呼び出すことができます。

次に示すように、<command> Elementは、<execute> および <resultsblock> Elementの構造をラップします。

```
<command name="p4_changes">
  <execute>
    command line
  </execute>
  <resultsblock>
    Has its own structure.
  </resultsblock>
</command>
```

あるいは、ブロック内の <execute> Elementを <integrate> Elementに置き換えることができます。

モード属性を使用して、<command> Elementのモードを指定します。モード属性 *exec* が使用されると、サーバーの始動がトリガーされ、コマンドが即時に処理されます。

注: Rational® Build Forge® 8.0 の場合、デフォルトの *exec* モードのみがサポートされます。

## env

<env> Elementは、環境変数を (初期値と一緒に) 定義するためにElement内で使用されます。それらの環境変数を、プロジェクト・リンクで使用する環境変数にコピーすることができます。各 <env> Elementには、名前 および値 プロパティを含める必要があります。値は、変数の初期値を提供します。

```
<env name="FILESPEC" value="//depot..." />
```

## execute

コマンドを指定するには、<command> Element内に <execute> Elementを使用します。Elementの内容は 1 行以上のテキストで、アダプターが使用するサーバーに送信されます。<execute> Element内でドット・コマンドを使用することはできません。<run> Elementが <command> Elementを呼び出すと、<execute> Elementの内容にある定位置パラメーターは、<run> Elementの呼び出しに指定されたパラメーターで置換されます。<execute> Elementの内容にある \$1 パラメーターは最初のパラメーターで置換され、\$2 パラメーターは 2 番目のパラメーターで置換されるといった具合です。

condition="function" 属性を使用して、<execute> Element内のコマンドの実行を制御します。function の値は、結果が TRUE または FALSE に評価される式です。式の評価が TRUE になる場合は、コマンドが実行されます。式の評価が FALSE になる場合は、コマンドは実行されません。421 ページの『条件関数』を参照してください。

```
<execute>
p4 changes -s submitted -t -i $2@$1,@now
</execute>
```

## field

フィールドを指定するには、<bom> エlementまたは <section> Element内のいずれかに <field> Elementを使用します。

<bom> Element内で使用する場合は、名前とテキストを指定します。テキスト属性では、どの変数を使用してフィールドにデータを表示するかを定義します。

<bomformat> <section> Elementで使用する場合は、名前とタイトルを指定します。名前が論理名を指定する一方、タイトルは表示に使用されます。<section> 内に複数のフィールドがある場合は、順序属性を使用してください。

```
<section name="changesets">
  <field order="1" name="Change" title="Change ID"/>
  <field order="2" name="Date" title="Date"/>
  <field order="3" name="Time" title="Time"/>
  <field order="4" name="User" title="User ID"/>
  <field order="5" name="Client" title="Client"/>
  <field order="6" name="Comment" title="Comment"/>
</section>
```

condition="function" 属性を使用して、<field> Elementを BOM レポートに書き込むかどうかを制御します。function の値は、結果が TRUE または FALSE に評価される式です。式の評価が TRUE になる場合、<field> Elementの情報が BOM に書き込まれます。式の評価が FALSE になる場合、情報は BOM に書き込まれません。 421 ページの『条件関数』を参照してください。

テンプレート属性を使用して、<field> Elementのテキスト・フォーマットを定義します。例えば、テキストがストリングである場合は、「Hello \$VALUE」をテンプレートの値にできます。フィールドが BOM レポートに書き込まれると、\$VALUE はフィールドのテキストで置換されます。

## integrate

<integrate> Elementは <execute> Elementと類似しています。<execute> Elementの代わりに <integrate> Elementを使用できます。 <execute> Elementと同じように、<integrate> Elementは実行されるコマンド行を指定します。ただし、次のような違いがあります。

- コマンド行は、アダプターを実行するサーバーではなく、管理コンソール・システムで実行されます。
- コマンド行は現行ディレクトリーとして %integration ディレクトリー (インストール・ディレクトリーのサブディレクトリー) を使用します。

<integrate> Elementは、管理コンソール・コンピューター上あるアプリケーションやスクリプト、特に %integration ディレクトリーにあるものを実行する場合に役に立ちます。

<run> Elementが <integrate> Elementを含む <command> Elementを呼び出すと、システムは、<integrate> Element内のすべての定位置変数を、呼び出し元 <run> Elementで指定されたパラメーターによって置き換えます。<integrate> Element内の \$1 は、最初のパラメーターによって置き換えられ、\$2 は 2 番目のパラメーターによって置き換えられます (以下同様)。

<execute> エレメントの場合と同様に、<integrate> エレメント内ではドット・コマンドを使用できません。

次の例は IBM Rational ClearQuest アダプターから抜粋したものです。CQperl コマンド (Perl コードを実行するための ClearQuest プログラム) を実行し、¥integration ディレクトリーにある Perl スクリプトの名前を割り当てることによって、データを ClearQuest に送信します。この例は、ClearQuest が管理コンソール・システムにインストールされていることを想定しています。

```
<integrate>
cqperl bfcqresolve.pl $2 Fixed "Fixed in build $BF_TAG"
</integrate>
```

## interface

<interface> エレメントは、アダプターへのエントリー・ポイントのコンテナです。ここに含まれるエレメントにより、アダプターのプログラム・ロジックが定義されます。<setenv>、<run>、<ontempenv>、<onproject>、および <adduser> の各エレメントが含まれます。

アダプター・テンプレートには複数のエントリー・ポイントを設定することができます。複数のエントリー・ポイントが設定されたアダプター・テンプレートを作成するには、次の手順を実行します。

- **name** 属性を使用して、それぞれのエントリー・ポイントを指定します。
- **default** 属性を使用して、名前が指定されずにアダプター・テンプレートが呼び出された場合に実行するエントリー・ポイントを指定します。

エントリー・ポイントを指定するには、テンプレートの .source 呼び出しのパラメーターとしてエントリー・ポイント名を使用します。

例: MyAdaptorTemplate というアダプター・テンプレートを作成します。このテンプレートに、次のコードを挿入して、エントリー・ポイントの名前を定義し、そのエントリー・ポイントをデフォルトに設定します。

```
<interface name="By Date" default="true">
</interface>
```

このインターフェースを名前で呼び出すには、プロジェクト・ステップで次のコマンドを使用します。

```
.source MyAdaptorTemplate "By Date"
```

属性:

- name** オプションです。このインターフェースの名前です。パラメーターとしてこの名前を使用してアダプターが呼び出された場合、そのパラメーターと一致する名前を持つインターフェースが使用されます。
- default** オプションです。Yes または No のいずれかに設定します。Yes の場合は、name パラメーターなしでアダプターが呼び出されたときに、このインターフェースが使用されます。

## match

<match> エレメントは、出力の行を処理するために <resultsblock> エレメント内で使用されます。<match> エレメントは、マッチング行を定義するパターン属性を取

ります。このパターンは正規表現です (Build Forge 8.0 では Java エンジンの Java 正規表現であり、Build Forge 8.0 およびすべての 7.1.x では Perl エンジンの Perl 正規表現)。

マッチング・パターンには、変数 \$1...\$n に保管されている括弧付きの式を含むことができます。

```
<match pattern="^Change (¥d+) on (.*) (.*) by (.*)@(.*?) '(.*)' '$">
```

<match> エレメントは、<adduser>、<setenv>、<bom>、および <field> をサブエレメントとして使用します。さらに詳しい例については、589 ページの『resultsblock』の参照情報を参照してください。

## notify

<notify> エレメントでは、E メールおよび配布リストを指定します。一般に、このエレメントは、<onproject> エレメント内で、実行結果に基づいて通知を指定する場合に使用されます。

属性:

**group** E メール宛の宛先です。これは、アダプターに定義されているユーザーのグループです。

**message**

Eメールの本文の内容です。テキストを入力します。

**subject**

Eメールの件名です。テキストを入力します。

## onproject

<onproject> エレメントは、システムがプロジェクト・ステップを実行した後で実行する通知アクションを定義します。このエレメントは、成功または失敗したジョブに関してアクションを実行するかどうかを指定する、必須の結果属性を使用します。一般には、アダプター XML ファイルには 2 つの <onproject> エレメントが含まれています。1 つは成功した場合用で、もう 1 つは失敗した場合用です。次の例は、プロジェクトが成功したか失敗したかに応じて異なるメッセージを送信するために notify エレメントを使用する、<onproject> エレメントのペアを示しています。

```
<onproject result="fail">
  <notify group="MyChangers"
    subject="Run $BF_TAG ($CurDate) Failed." message="$Changing$Changes"/>
</onproject>

<onproject result="pass">
  <notify group="MyChangers"
    subject="Run $BF_TAG ($CurDate) Passed." message="$Changing$Changes"/>
</onproject>
```

## ontempenv

<ontempenv> エレメントは <interface> エレメント内で使用され、if-then ステートメントと同様に機能します。このエレメントは、プロジェクトに pass (成功) または fail (失敗) の値を渡すために使用します。pass は、システムを続行してプロジェクトの残りの部分を実行することを指示し、fail はシステムの停止を指示します。これは通常、インターフェースがプロジェクトの新規実行に有益な関連変更内容を検出したかどうかを示すために使用されます。



システムは、<run> エlementで指定されたなんらかのコマンドを実行した後で、<ontempenv> エlementを処理します。一時環境変数を指定するには、このElementの名前属性を使用します。また、値を指定するには、状態属性を使用します。

<ontempenv> は <step> Elementをラップします。<step> Elementは、<run> Elementのコマンドの実行後に一時環境変数の name および state が存在する場合にのみ実行されます。

```
<ontempenv name="Changes" state="empty">
  <step result="FAIL"/>
</ontempenv>
```

## PROJECT\_INTERFACE

<PROJECT\_INTERFACE> Elementは、アダプター・テンプレート内にある他のすべてのタグをラップします。このElementは IFTYPE という 1 つの属性を使用します。この属性はアダプター・タイプを示します。有効なタイプは Source、Test、および Defect です。

```
<PROJECT_INTERFACE IFTYPE="Source">
...all other elements...
</PROJECT_INTERFACE>
```

## relate

<relate> Elementでは、成果物とユーザーとの関係を指定します。これは、「通知チェンジャー」タイプのログ・フィルターと組み合わせて使用するElementです。このフィルターは、プロジェクトで定義して、アダプターを呼び出すステップの直前に実行するステップの Result 属性で使用します。一般に、「通知チェンジャー」フィルターでは、予期されるログ行 (成功または失敗を示すもの) を指定します。ステップが実行されると、フィルターは、<relate> Elementで成果物として指定されるテキストと比較されます。一致した場合、その成果物に関連付けられたユーザーに E メールが送信されます。

379 ページの『フィルター・アクション』の「通知チェンジャー」フィルターの説明も参照してください。

属性:

### artifact

ログ・フィルターによって検索されたテキストと突き合わせるテキスト。

**user** 成果物に関連付けるユーザー。一致が発生すると、このユーザーに E メールが送信されます。

**text** ログ・フィルターとこの関係との間に一致が発生したときに BOM に記録するテキスト。

## resultsblock

<resultsblock> Elementは、関連する <execute> Elementから実行されたコマンド行の結果に対するシステムの処理方法を定義します。<resultsblock> Elementは <command> Element内でのみ使用されます。<resultsblock> Elementをネストして、結果を区切ることができます。

<resultsblock> Elementには、正規表現 (Build Forge 8.0 では Java エンジンの Java 正規表現であり、Build Forge 8.0 およびすべての 7.1.x では Perl エンジンの



Perl正規表現) を使用して処理対象の出力行の範囲を定義するためのオプション属性である `beginpattern` および `endpattern` を指定することができます。また、異なる `<resultsblock>` エレメントを使用することで、異なる範囲を処理することができます。以下の疑似コードは、`<resultsblock>` エレメントの構造を示したものです。

```
<resultsblock startpattern="" endpattern="" >
  <match>
    <bom>
      <field/>
    </bom>
    <adduser/>
    <setenv/>
    <run/> (<run> エレメントを使用すると、
    コマンドを別のコマンドの中で実行できます)
  </match>
  <setenv/>
</resultsblock>
```

次の例は、`<resultsblock>` エレメント、`<match>` エレメント、および `<bom>` エレメントと一緒に使用方法を示したものです。

```
<resultsblock
beginpattern="^Change (¥d+) by (.*?)@(.*?) on (.*?) (.*?)$"
endpattern="^Differences ...$"
  <match pattern="(?!^(?:(?!Differences ...).)*$).?">
    <bom category="Detail" section="descriptions">
      <field name="Description" text="$_" />
    </bom>
  </match>
</resultsblock>
```

## run

`<run>` エレメントは、実行するコマンドの名前を指定するために `<interface>` エレメント内で使用されます。このコマンドは、後で同じ XML ファイル内で定義されます。`<run>` エレメントは、空のエレメントです。

属性:

### condition

これはオプションです。 `condition="function"` の形式で指定します。 *function* の値は、結果が `TRUE` または `FALSE` に評価される式です。式の評価が `TRUE` になる場合、コマンドが実行されます。式の評価が `FALSE` になる場合、コマンドは実行されません。 421 ページの『条件関数』を参照してください。

### command

これは必須です。実行する定義済みコマンドの名前を指定します。コマンドは、`<command>` エレメント内で命名および定義されます。

**dir** これは必須です。コマンドを実行するディレクトリを指定します。 `dir` は、コマンドの実行に使用されるサーバー・リソースにおけるパス・セットの拡張と解釈されます。

**mode** オプションです。 `run` コマンドの実行モードを指定します。モード属性 `exec` が使用されると、サーバーの始動がトリガーされ、コマンドが即時に処理されます。

注: Rational® Build Forge® 8.0 の場合、デフォルトの `exec` モードのみがサポートされます。

### params

これは必須です。コマンドに渡すパラメーターを指定します。パラメーターの区切りには、スペースを使用します。

### server

これは必須です。コマンドの実行に使用されるサーバー・リソースを指定します。 NULL に設定された場合、アダプターを実行するステップで使用するサーバーで、コマンドが実行されます。 server を NULL に設定するには、server="" を使用します。

### timeout

これは必須です。コマンドがタイムアウトになるまでの秒数を指定します。

必須の属性を示す例を以下に示します。

```
<run command="UpdateEnv" params="" server="" dir="/" timeout="360"/>
```

例:

```
<run command="p4_changes" params="$LAST_RUN $FILESPEC $LAST_VER"  
server="$P4CLIENT" dir="/" timeout="360"/>
```

### section

BOM カテゴリーの一部をどのように表示するかを定義するには、<section> エレメントを使用します。これは、名前属性を使用します。<section> エレメントは、<bomformat> エレメント内でのみ使用できます。

### setenv

<setenv>エレメントを使用して、<interface> エレメントまたは <match>エレメント内の環境変数の値を初期化します。 <setenv> エレメントには、他のエレメントは含まれません。

エレメントは、以下の 3 つの異なる方法で使用することができます。

- グループ名を指定する場合、このエレメントは .set コマンドのように機能します。その際、現行のステップで使用されているコピーではなく、データベース内のマスター・レコードに変数値を設定します。このため、現行のステップで実行されているアダプターでは、変更内容は表示されません。この方法で新しい変数を作成することはできません。
- グループ名を指定しない場合、このエレメントは .bset コマンドのように機能します。その際、実行中のジョブの環境に変数値を設定します。変更は、そのジョブのすべてのステップで確認できます。この方法で新しい変数を作成することができます。
- グループ名を指定せず、なおかつ 一時変数 (type="temp") を指定した場合は、アダプター・ロジックを使用するためのみの一時変数がセットアップされます。この変数は、アダプター・ステップの実行後は存続しません。この方法で新しい変数を作成することができます。

属性:

### condition

これはオプションです。condition="function" の形式で指定します。function の値は、結果が TRUE または FALSE に評価される式です。式の評価が

TRUE になる場合、コマンドが実行されます。式の評価が FALSE になる場合、コマンドは実行されません。 421 ページの『条件関数』を参照してください。

**eval** オプションです。True または False に設定します。True の場合、アダプターは value 属性式を評価して結果を保管することを試みます。

**group** オプションです。変数が定義される Build Forge 環境を指定します。環境名を指定する場合は、指定された環境内にある既存の変数を参照する必要があります。

値として [ADAPTOR] を指定した場合、その値は実行時に設定されます。この値は、アダプターを呼び出すステップまたはアダプター・リンクの環境に設定されます。

Build Forge では、複数の環境で同じ名前の変数を使用することができます。環境継承および環境包含の優先順位は、実行時に変数に割り当てる値を決定する方法に影響することがあります。

**name** これは必須です。設定する変数の名前を指定します。値は変数でもかまいません。その場合、変数名は実行時まで設定されません。

**type** オプションです。変数を設定する方法を指定します。次のいずれかの値を取ります。

- **append text**: 既存の値の後に、指定された値を挿入します。オプションの *text* が指定された場合、そのテキストは値と値の間に挿入されます。
- **once**: 変数は、まだ設定されていない場合のみ設定されます。
- **prepend text**: 既存の値の前に値を挿入します。
- **temp**: 変数は、アダプターのコンテキストでのみ設定されます。オプションの *text* が指定された場合、そのテキストは値と値の間に挿入されます。次の例を参照してください。

**value** これは必須です。変数に設定する値を指定します。eval 属性も指定されている場合は、評価される式でもかまいません。評価の結果は、値として保管されます。

例:

次の例では、value 属性の式が評価され、その結果が変数 LAST\_VER に保管されます。値は、\$LAST\_VER、または \$1 変数の値のいずれか大きい方に設定されます。

```
<setenv group="Adaptor" name="LAST_VER"
value="$LAST_VER>$1?$LAST_VER:$1" eval="true" />
```

次の例では、Changes 変数にデータを付加する前に、改行文字 (¥n) が挿入されます。

```
<setenv name="Changes" value="$4 - $1 - $6" type="temp append¥n" />
```

次の例では、変数 INFOPATH の先頭に付加する値の後ろに、コロンが挿入されます。

```
<setenv name="INFOPATH" value="/usr/local" type="temp prepend:" />
```

## **step**

`/<step>` エLEMENTは、`<ontempenv>` エLEMENT内でのみ使用されます。このエLEMENTは、特殊なアダプター・ステップの結果を記述します。これは空のエLEMENTです。以下の例は、`<step>` エLEMENTの 2 つのフォームを示しています。

```
<step result="FAIL"/>
```

```
<step result="pass"/>
```



## 第 27 章 Rational Build Forge と Rational ClearCase

BuildForge は、コード・ソース管理のために Rational ClearCase と協働することができます。

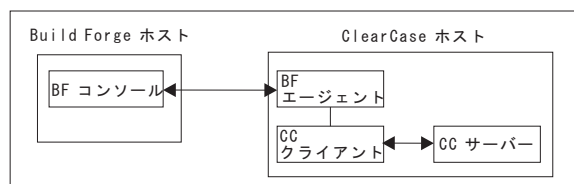
このトピックでは、Rational ClearCase との統合のための追加セットアップ要件について説明し、用意されているサンプル・アダプター・テンプレートの特性を要約します。

ClearCase のアダプター・テンプレート・サンプルでは、ベースラインに対する変更内容を分析する方法が提供されます。通常、変更分析はビルド回避のために使用されます。ベースライン・コンポーネントが変更されていない場合、再ビルドは行われません。

### ClearCase アダプターのための追加セットアップ要件

一般要件については、552 ページの『アダプター要件』を参照してください。アダプターの一般要件を満たす他に、以下を行ってください。

1. Build Forge エージェントを、ClearCase サーバーに接続可能なホスト上にインストールします。
2. エージェント・ホストに ClearCase のフル・クライアントをインストールします。
3. ClearCase クライアントからコマンドを実行できるように、エージェントの環境をセットアップします。
4. 必要な ClearCase ビューを実装する方法、およびそのビューをいつどのように開始するかを決定します。提供されるテンプレートでは、動的ビューを使用することを前提としており、cleartool コマンドの実行時にそのビューを開始します。エージェント・ホストに ClearCase のフル・クライアントをインストールします。



### ClearCase ビュー

ClearCase では、データにアクセスする手段としてビューを使用します。使用できるビューには次の 2 つのタイプがあります。

- 動的ビュー: 動的ビューを使用すると、指定した VOB 内の最新バージョンのエレメントを、ローカル・システムにコピーしなくても表示することができます。これは必ず起動してください。このビューでマウントする VOB を指定します。

- **スナップショット・ビュー:** スナップショット・ビューはリポジトリのコピーです。使用するオブジェクトのバージョンを指定できます。チェックインされた最新項目を取得するには、スナップショット・ビューを手動で更新する必要があります。

### 動的ビュー

動的ビューには、表示対象として選択したビュー・コンテキストの変更時に常に更新されるワークスペースが示されます。チェックアウトしたエレメントのみがワークスペースにコピーされます。他のユーザーが使用できるようにするにはそれらのエレメントをチェックインする必要があります。動的ビューを使用する場合、以下の 3 つを指定して、必要なデータにアクセスします。

- **ビュー:** (構成仕様を使用して) 表示するバージョンのリストを定義します。ビューはコマンドを使用して開始する必要があります。ビューは、コンテンツのルート・ディレクトリとして、ローカル・ディスク上に示されます。これはビュー・ルートとも言います。一般的に使用されるディレクトリは以下のとおりです。
  - Windows の場合: ドライブ M: (共有ドライブ)
  - UNIX または Linux の場合: /view (マウント済みファイル・システム)
- **ビュー・コンテキスト:** ビュー・タグを介して使用するコンテンツのディレクトリを定義します。ビュー・タグは ClearCase で定義されます。ビューを開始すると、ディレクトリが使用可能になります。
- **VOB (バージョン付きオブジェクト・ベース):** ビュー・コンテキストのサブディレクトリを定義します。VOB は ClearCase で定義されます。VOB はビュー・コンテキストで明示的にマウントします。このサブディレクトリには、バージョン付き成果物 (ソース・ファイルとその他の成果物) が含まれます。

ローカル・マシン上の VOB のパスは、以下のように、これらの 3 つの項目で構成されます。

- Windows システムの場合、デフォルトのパスは M:¥view\_tag¥vob\_name になります。
- UNIX システムおよび Linux システムの場合、デフォルトのパスは /view/view\_tag/vob\_name になります。

### スナップショット・ビュー

スナップショット・ビューを使用する場合は、ビュー・コンテキストを指定します。また、データをロードする必要もあります。これにより、指定したすべてのデータのローカル・コピーが作成されます。データを最後にロードしてからチェックインされた変更を反映させるには、スナップショット・ビューを再ロードする必要があります。

## 動的ビューをサポートする Rational Build Forge のフィーチャー

特別な環境変数を使用して、動的ビューを定義して Build Forge ジョブで開始することができます。

### CLEARCASE\_VIEW

指定された ClearCase ビューを開始する。この変数で指定されるビューが存在しており、この変数を使用するステップが「絶対」に設定されている必要



があります Microsoft Windows が稼働しているシステムでは、bfagent.conf 内のエージェントの cc\_suppress\_server\_root パラメーターとともにこの変数を使用する必要があります。

#### **\_CLEARCASE\_VIEWS**

コマンドを実行する前に開始する ClearCase ビューのリストを指定する。ビューのコンマ区切りのリストに値を設定します (「View1,View2,View3」など)。

#### **\_CLEARCASE\_VOBS**

コマンドを実行する前にマウントする ClearCase VOB のリストを指定する。VOB のコンマ区切りのリストに値を設定します (「¥Vob1,¥Vob2,¥Vob3」など)。

エージェント・ホスト上でビューを管理するには、エージェント・パラメーターを使用する必要がある場合もあります。ClearCase で使用しているエージェントが Windows システム上で稼働している場合は、エージェント・パラメーターを設定する必要があります。これらのパラメーターは、このエージェントを通じて使用するビュー・ルートがジョブで設定されたビュー・ルートとは異なる場合にも役立ちます。複数のエージェントでジョブを使用しており、エージェント・ホスト・システムごとに異なる仕様がが必要な場合などが考えられます。

#### **ccviewroot root-path**

このホストのデフォルトのビュー・ルートを指定します。詳しくは、ClearCase の資料に記載されている init の説明を参照してください。内部デフォルト値は以下のとおりです。

- Windows systems: ccviewroot M:
- UNIX または Linux システムの場合: ccviewroot /view

#### **cc\_suppress\_server\_root**

これを設定した場合、ビューのパスは ccviewroot によって設定されたパスになります。設定していない場合は、ccviewroot によって設定されたパスに、サーバー定義で設定されたパスが付加されます。この設定に値は必要ありません。bfagent.conf にこの設定があれば、設定されていることになります。

## **ClearCase のアダプター・テンプレート・サンプル**

以下のアダプター・テンプレート・サンプルが用意されています。

#### **ClearCaseBaseline**

1. ClearCase ビューのディレクトリーをスキャンします。
2. ClearCase からレポートされた分岐およびバージョン情報を BOM レポートに書き込みます。

使用される変数は以下のとおりです。

- INT\_STREAM
- VIEW
- PROJECT\_VOB
- CCSERVER
- UNIXCLIENT

### **ClearCaseByBaselineActivities**

1. ClearCase ビューのコンテンツから新規ベースラインを作成します。
2. 新規ベースラインと、前のアダプター実行でのベースラインを比較して、変更アクティビティを特定します。
3. 各変更アクティビティに関して、次の情報を BOM レポートに書き込みます: アクティビティ、変更されたファイル、ユーザー、日付、コメント、およびバージョン。
4. 変更されたファイルごとに、(diff コマンドの出力から得られる) 変更詳細を BOM レポートに書き込みます。

使用される変数は以下のとおりです。

- CurDate
- LAST\_RUN
- BASELINE
- VIEW
- VOB\_PATH
- PROJECT\_VOB
- CCSERVER
- UNIXCLIENT

### **ClearCaseByBaselineVersions**

1. ClearCase ビューのコンテンツから新規ベースラインを作成します。
2. 新規ベースラインと、前のアダプター実行でのベースラインを比較して、変更されたファイルを特定します。
3. 変更された各ファイルに関して、次の情報を BOM レポートに書き込みます: ファイル名、バージョン、日付、ユーザー、およびコメント。
4. 変更されたファイルごとに、(diff コマンドの出力から得られる) 変更詳細を BOM レポートに書き込みます。

使用される変数は以下のとおりです。

- CurDate
- LAST\_RUN
- LABEL
- BASELINE
- VIEW
- VOB\_PATH
- PROJECT\_VOB
- CCSERVER
- UNIXCLIENT
- \_CHAR\_NATIVE

### **ClearCaseByDate**

1. 2 つの日付の間に変更がないか、ClearCase ビューを照会します。デフォルトの日付は、現行のタイム・スタンプおよび前のアダプター実行時のタイム・スタンプです。

2. 変更された各ファイルに関して、次の情報を BOM レポートに書き込みます: ファイル名、バージョン、日付、ユーザー、およびコメント。
3. 変更されたファイルごとに、(diff コマンドの出力から得られる) 変更詳細を BOM レポートに書き込みます。

使用される変数は以下のとおりです。

- CurDate
- LAST\_RUN
- LABEL
- BASELINE
- VIEW
- VOB\_PATH
- PROJECT\_VOB
- CCSERVER
- UNIXCLIENT
- \_CHAR\_NATIVE

#### ClearCaseByLabel

1. 新規ラベルを作成して、ClearCase ビューのコンテンツに適用します。
2. 新規ラベルと、前のアダプター実行でのラベルを比較して、変更されたファイルを特定します。
3. 変更された各ファイルに関して、次の情報を BOM レポートに書き込みます: ファイル名、バージョン、日付、ユーザー、およびコメント。
4. 変更されたファイルごとに、(diff コマンドの出力から得られる) 変更詳細を BOM レポートに書き込みます。

使用される変数は以下のとおりです。

- CurDate
- LAST\_RUN
- LABEL
- BASELINE
- VIEW
- VOB\_PATH
- PROJECT\_VOB
- CCSERVER
- UNIXCLIENT
- \_CHAR\_NATIVE

### ClearCase のアダプター変数

この表は、アダプター・テンプレートの変数リストのリファレンスです。

環境変数名	説明
BASERLINE	ByBaseline アダプターの場合、ご使用のアダプターを使用して、ベースラインごとに違いが生じると、システムは、この値をベースラインとして使用します。

環境変数名	説明
CCSERVER	この変数を、ClearCase クライアントおよび Build Forge エージェントがインストールされているホストの名前に設定します。
CurDate	現在の日付をアダプターに指定し、.date コマンドを使用して、ClearCase が予期する書式で日付を生成します。この値は変更しないでください。
LABEL	ByLabel アダプターの場合、ご使用のアダプターを使用して、(ByLabel アダプターに付く) ラベルごとに違いが生じると、システムは、この値をラベルとして使用します。
LAST_RUN	ByDate アダプターの場合、システムはこの値を使用して、何らかの変更が発生したか判別します。この値は、最後に正常に実行されたときの日付です。アダプターのテストの際には、この値を操作し、変更より前であると分かっている日付を取り出して、アダプターを強制的に実行させることができます。アダプターが実行の継続を許可した場合、この値は、現在の日付に自動的に更新されます。デフォルト値は、1-Jan-05.00:00:00 です。
PROJECT_VOB	ByBaseline アダプターを使用する場合、この変数を、プロジェクト VOB (UCM ClearCase のみで使用される) の名前に設定します。例: ¥ProjectVob
UNIXCLIENT	プラットフォーム固有の情報を設定するために使用されます。クライアントが Windows 上で実行されている場合は、0 に設定します。クライアントが UNIX または Linux 上で実行されている場合は、1 に設定します。
VIEW	この変数を、アダプターで使用する ClearCase ビューの名前に設定します。
VOB_PATH	この値を、コンポーネント VOB の名前 (オプションで、そのサブディレクトリー) に設定します。名前が複数ある場合は、コンマ区切りのリストを使用します。
_CHAR_NATIVE	内部で使用され、常に 1 に設定されます。

ビューの開始および VOB のマウントのために、以下のトリガー変数が使用されることもあります。ただし、これらはアダプター変数で設定されるビューおよび VOB とは無関係です。

- CLEARCASE\_VIEW
- \_CLEARCASE\_VIEWS
- \_CLEARCASE\_VOBS

344 ページの『トリガー変数のリファレンス』を参照してください。

---

## 第 28 章 Rational Build Forge と Rational ClearQuest

BuildForge は、ビルド・レコードの更新のために Rational ClearQuest と協働することができます。

Build Forge と ClearQuest の統合には、以下の 2 つがあります。

- ジョブ状況に基づくビルド・レコードの自動作成または自動更新。この機能は、必要な環境変数がプロジェクト環境で設定されると、自動的にアクティブになります。
- ClearQuest アダプター

これらの機能は、それぞれ完全に独立しています。特に、アダプターは、そのアダプター用に作成された環境と関連付けられています。その環境内の変数は、自動ビルド・レコードをアクティブにするために設定される変数とは無関係です。

### ビルド・レコードの自動生成のセットアップ

IBM Rational® ClearQuest® データベース内に、ログ・データへのリンクを含むビルド・レコードが自動的に作成されます。さらに、ジョブが成功すると、システムは ClearQuest データベースを更新し、ジョブが完了したことを示してから、完遂したステップの終了時刻および要約を記録します。この機能には、Rational ClearQuest バージョン 7.0 以降が必要となります。

ClearQuest データベースを更新するようにプロジェクトを構成すると、システムはビルド・レコードを以下のように作成または更新します。

#### ジョブの開始

システムは、ジョブを起動すると、ClearQuest ビルド・レコードを作成します。ビルド・レコードは「実行依頼済み (Submitted)」状態になり、「ビルド XYZ が開始しました」というログ・エントリーに加えて、ジョブ・ログの URL、開始時刻、リリース名、および ID がレコードに含まれます。ソース制御アダプターがジョブをキャンセルすると (例えば、ソースの変更がなかったため)、ClearQuest ビルド・レコードは作成されません。

**注:** あるプロジェクトが別のプロジェクトをチェーニングすると、新規プロジェクトはそれ固有の ClearQuest ビルド ID を取得します。

#### ジョブの成功または失敗

ジョブが成功または失敗すると、ClearQuest 内のビルド状態が「完了」または「失敗」に変更され、ビルド終了時刻が設定され、ClearQuest ビルド・ログにジョブのステップの要約が保管されます。要約には、各ステップの名前、結果状況、およびサーバーが含まれます。

#### ジョブの再開

ジョブを再開すると、ClearQuest 内のビルド状態が「実行依頼済み (Submitted)」になり、「XYZ のビルドが再開しました」という内容の ClearQuest ビルド・ログ・エントリーが作成されます。

ビルド・レコードの自動更新は、特殊な環境変数を使用して構成します。プロジェクトを ClearQuest データベースとリンクするには、下表の変数がプロジェクトの環境変数に組み込まれていることを確認してください。

**注:** これらの変数はプロジェクト環境変数内に存在する必要があります。ステップに追加するだけでは十分ではありません。ただし、別の環境変数を介してこれらの変数を含める「組み込み」タイプに設定された変数を使用することができます。また、プロジェクトごとに変わる可能性があるのは **CQ\_RELEASE\_NAME** の値のみであるため、その他の変数を含む環境変数を作成し、「組み込み」型の変数を使用してプロジェクト環境変数内にその環境変数を含めることができます。その場合、**CQ\_RELEASE\_NAME** をプロジェクト固有の環境変数として指定することもできます。

Build Forge ジョブからビルド・レコードの自動更新をアクティブにするには、プロジェクトに対して以下の環境変数を設定する必要があります。これらはステップ・レベルでは機能しません。

変数	説明
CQ_DBNAME	これは必須です。更新する ClearQuest データベースの名前。
CQ_DBSET	ClearQuest データベースの設定値です。必須ではありません。デフォルトはブランクです。
CQ_INTERACTION	プロジェクト環境変数に定義済みの正しい環境変数があり、ClearQuest ビルド・レコードの作成が使用可能になっているが、ビルド・レコードを作成したくない場合は、この変数を OFF に設定して、ビルド・レコードの作成を使用不可にします。  ビルド・レコードの作成を使用可能にするには、この環境変数を ON に設定します。  <b>注:</b> ClearQuest アダプターのいずれかを使用している場合は、この環境変数を OFF に設定してください。このアダプターは、ビルド・レコードと直接対話します。
CQ_PASSWORD	これは必須です。ClearQuest データベースにログインするときに使用するパスワードです。必須ではなく、デフォルトはブランクです。
CQ_RELEASE_NAME	これは必須です。ClearQuest データベース内にある更新するリリースの名前です。
CQ_USER	これは必須です。ClearQuest データベースにログインするときに使用するユーザー名です。

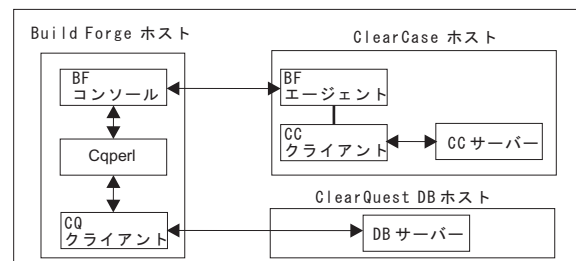
## ClearQuest アダプターのための追加セットアップ要件

ClearQuest アダプター・テンプレート・サンプルでは、ClearCase のスキャン方法と ClearQuest のビルド・レコードの更新方法が提供されます。通常は、これにより Build Forge で実行されるビルドの成否が左右されます。一般要件については、552 ページの『アダプター要件』を参照してください。また、以下の構成を行う必要があります。

- ソースをスキャンするための ClearCase へのアクセス権限。ジョブの実行時に、アダプターはエージェントと ClearCase クライアントを介して cleartool コマンドを実行します。アダプターは ClearQuest Perl API (cqperl) を使用してコマンドを実行します。
- ビルド・レコードを更新するための ClearQuest へのアクセス権限。ジョブの実行時に、アダプターはコンソール・ホスト上で直接 cqperl スクリプトを実行します。スクリプトは Cqperl ユーティリティによって解釈され、ClearQuest クライアントを介して実行されます。このユーティリティとクライアントは、両方ともコンソール・ホスト上にインストールされています。

以下を実行します。

1. Build Forge エージェントを、ClearCase サーバーに接続可能なホスト上にインストールします。
2. エージェント・ホストに ClearCase のフル・クライアントをインストールします。
3. ClearCase クライアントからコマンドを実行できるように、エージェントの環境をセットアップします。
4. Build Forge コンソール・ホストに ClearQuest のフル・クライアントをインストールします。
5. cqperl (ClearQuest Perl API) ディレクトリーをシステム・パスに追加します。
6. Build Forge ホスト上の ClearQuest クライアントが ClearQuest データベースにアクセスする際に使用できる接続を定義します。以下のアクションを ClearQuest クライアント・ホストで実行します。
  - a. cqreg コマンドを使用して、データベース・セットを追加する (cqreg add\_dbset)。
  - b. CQ メンテナンス・ツールを使用して、ClearQuest データベースへの接続をセットアップする。
7. 必要な ClearCase ビューを実装する方法、およびそのビューをいつどのように開始するかを決定します。



エージェントをインストールする必要はありません。 ClearQuest アダプターは、ClearQuest Perl API を使用して、クライアントを介して ClearQuest と直接通信します。

**重要:** ClearQuest アダプターは、ステップ内でドット・コマンドを使用することによってのみ呼び出すことができます。ソース・アダプターではないため、アダプター・リンクは使用できません。



## ClearQuest のアダプター・テンプレート・サンプル

以下のアダプター・テンプレート・サンプルが用意されています。

### ClearQuestBaseClearCaseByDate

1. 2 つの日付の間に変更がないか、ClearCase ビューを照会します。デフォルトの日付は、現行のタイム・スタンプおよび前のアダプター実行時のタイム・スタンプです。
2. 変更された各ファイルについて、ClearQuest 変更 ID を指定する CrmRequest ハイパーリンク属性を探します。ClearQuest 状況により解決が可能な場合は、ジョブ情報を追加して変更 ID の解決を試行し、ClearQuest 内の不良レコードを解決します。
3. 変更された各ファイルに関して、次の情報を BOM レポートに書き込みます: ファイル名、不良 ID、不良状況、および ClearQuest エラー。

このアダプター・テンプレートで定義される変数は以下のとおりです。

- CurDate
- LAST\_RUN
- VIEW
- VOB\_PATH
- CQ\_USER
- CQ\_PASSWORD
- BFSERVER
- UNIXCLIENT
- \_CHAR\_NATIVE

### ClearQuestClearCaseByActivity

1. ClearCase アクティビティのリストに関連付けられた ClearQuest 不良レコードを検出します。
2. 検出された各不良レコードについて、ClearQuest 状況により解決が可能な場合は、ジョブ情報を追加して ClearQuest 内の不良レコードを解決します。
3. ClearCase アクティビティ ID および ClearQuest 不良状況に関連付けられたファイルに関する情報を、BOM レポートに書き込みます。

このアダプター・テンプレートで定義される変数は以下のとおりです。

- CurDate
- VIEW
- VOB\_PATH
- ACTIVITIES
- CQ\_USER
- CQ\_PASSWORD
- PROJECT\_VOB
- BFSERVER
- UNIXCLIENT
- \_CHAR\_NATIVE

## ClearQuestUCMClearCaseByDate

1. 2 つの日付の間に変更がないか、ClearCase ビューを照会します。デフォルトの日付は、現行のタイム・スタンプおよび前のアダプター実行時のタイム・スタンプです。これは、Rational 統一変更管理 (UCM) を使用して、結果を生成します。
2. 変更された各ファイルに関して、次の情報を BOM レポートに書き込みます: ファイル名、不良 ID、不良状況、および ClearQuest エラー。

このアダプター・テンプレートで定義される変数は以下のとおりです。

- CurDate
- LAST\_RUN
- VIEW
- VOB\_PATH
- CQ\_USER
- CQ\_PASSWORD
- BFSERVER
- UNIXCLIENT
- \_CHAR\_NATIVE

## ClearQuest のアダプター変数

この表は、アダプター・テンプレートの変数リストのリファレンスです。

表 19. Rational ClearQuest の統合に必要な環境変数

変数	説明
ACTIVITIES	ClearQuestClearCaseByActivity アダプターの場合、スペースで区切られた、アクティビティ ID のセット。例: SAMPL0001@¥ProjectVob
BFSERVER	この変数を、Build Forge コンソールのホストの名前に設定します。
CQ_PASSWORD	これは必須です。ClearQuest データベースにログインするときに使用するパスワードです。必須ではなく、デフォルトはブランクです。
CQ_USER	これは必須です。ClearQuest データベースにログインするときに使用するユーザー名です。
CurDate	現在の日付をアダプターに指定します (日付の生成には .date コマンドを使用します)。この値は変更しないでください。
LAST_RUN	ByDate アダプターの場合、システムはこの値を使用して、何らかの変更が発生したか判別します。この値は、最後に正常に実行されたときの日付です。アダプターのテストの際には、この値を操作し、変更より前であると分かっている日付を取り出して、アダプターを強制的に実行させることができます。アダプターが実行の継続を許可した場合、この値は、現在の日付に自動的に更新されます。デフォルト値は、1-Jan-05.00:00:00 です。
UNIXCLIENT	プラットフォーム固有の情報を設定するために使用されます。クライアントが Windows 上で実行されている場合は、0 に設定します。クライアントが UNIX または Linux 上で実行されている場合は、1 に設定します。
VIEW	この変数を、アダプターで使用する ClearCase ビューの名前に設定します。

表 19. Rational ClearQuest の統合に必要となる環境変数 (続き)

変数	説明
VOB_PATH	この値を、コンポーネント VOB の名前 (オプションで、そのサブディレクトリー) に設定します。名前が複数ある場合は、コンマ区切りのリストを使用します。
_CHAR_NATIVE	内部で使用され、常に 1 に設定されます。

## ClearQuest 統合ジョブの再始動

ClearQuest 統合が完了すると、通常は Rational Build Forge で再始動することはできなくなります。これは、ジョブを新規ジョブとして開始すれば簡単に回避できます。

再始動を有効にするには、ClearQuest デザイナー・ツールで ClearQuest スキーマを編集してください。「完了」から「作成」の状態遷移を可能にするには、ビルド・レコードのワークフローを変更する必要があります。

---

## 第 29 章 Rational Build Forge と CVS

BuildForge は、ソース・コード管理のために CVS と協働することができます。

このトピックでは、CVS との統合のための追加セットアップ要件について説明し、用意されているサンプル・アダプター・テンプレートの特性を要約します。

CVS のアダプター・テンプレート・サンプルでは、ベースラインに対する変更内容を分析する方法が提供されます。通常、変更分析はビルド回避のために使用されます。ベースライン・コンポーネントが変更されていない場合、再ビルドは行われません。

### CVS アダプターのための追加セットアップ要件

一般要件については、552 ページの『アダプター要件』を参照してください。CVS のための追加要件はありません。

### CVS のアダプター・テンプレート・サンプル

以下のアダプター・テンプレート・サンプルが用意されています。

#### CVSv1Baseline

1. Build Forge エージェントの CVS ディレクトリーで変更されたファイルをスキャンします。
2. 次の情報を BOM レポートに書き込みます: 変更されたファイル名、状況、作業バージョン、リポジトリ・バージョン、およびスティッキー・タグ。ClearCase ビューのディレクトリーをスキャンします。

使用される変数は以下のとおりです。

- CVSROOT
- MODULE
- CVSCLIENT

#### CVSv1ByDate

1. 2 つの日付の間に変更がないか、CVS ビューを照会します。デフォルトの日付は、現行のタイム・スタンプおよび前のアダプター実行時のタイム・スタンプです。
2. 次の情報を BOM レポートに書き込みます: 変更タイプ、日付、ユーザー名、バージョン、およびファイル名。
3. 変更されたファイルごとに、(diff コマンドの出力から得られる) 変更詳細を BOM レポートに書き込みます。

使用される変数は以下のとおりです。

- LAST\_RUN
- CurDate
- CVSROOT
- MODULE

- BRANCH
- CVSCLIENT

#### **CVSv1ByTag**

1. 新規タグを CVS モジュールに適用します。
2. 新しくタグ付けされたモジュールと、前のアダプター実行時にタグ付けされたモジュールの間の相違を比較します。
3. 次の情報を BOM レポートに書き込みます: ファイル名、改訂、状況、日付、時刻、変更の作成者、およびコミット・コメント。
4. 変更されたファイルごとに、(diff コマンドの出力から得られる) 変更詳細を BOM レポートに書き込みます。

使用される変数は以下のとおりです。

- LAST\_RUN
- CurDate
- CVSROOT
- MODULE
- BRANCH
- LAST\_TAG
- CVSCLIENT

#### **CVSv2ByDate**

1. 2 つの日付の間に変更がないか、CVS ビューを照会します。デフォルトの日付は、現行のタイム・スタンプおよび前のアダプター実行時のタイム・スタンプです。
2. 次の情報を BOM レポートに書き込みます: 変更タイプ、日付、ユーザー名、バージョン、およびファイル名。
3. 変更されたファイルごとに、(diff コマンドの出力から得られる) 変更詳細を BOM レポートに書き込みます。

使用される変数は以下のとおりです。

- LAST\_RUN
- CurDate
- CVSROOT
- MODULE
- CVSCLIENT

---

## 第 30 章 IDE 統合

プラグインを使用すると、Rational Build Forge と統合開発環境 (IDE) を統合できます。

このセクションでは、統合開発環境 (IDE) から管理コンソール機能にアクセスできるようにするための、プラグインのインストールおよび使用方法について説明します。

---

### IDE 統合について

開発者は、統合開発環境 (IDE) プラグインを使用して、管理コンソールに接続することができます。

用意されたプラグインを使用することで、開発者は IDE から直接以下の作業を実行できます。

- プロジェクトを表示する
- ジョブを実行する
- ジョブ結果を検査する

その他の機能は、プラグインごとに異なります。

開発者はそれぞれ、Build Forge がユーザー・リストから確認するユーザー名を使用してコンソールに接続します。プロジェクトへのアクセスは、アクセス・グループ内のユーザー名メンバーシップによって制御されます。プロジェクト内のステップへのアクセスも、アクセス・グループによって制御されます。ステップには、アクセス・グループを明示的に指定できます。指定しなければ、プロジェクトのアクセス・グループが継承されます。

プラグインには、プロジェクトおよびステップを編集したり、削除したりする機能はありません。

ユーザーがブラウザーのクライアント・セッションを通じてライセンスを消費するように、IDE を通じて Build Forge にアクセスする各ユーザーはライセンスを消費します。

プラグインは、以下の IDE 向けに提供されます。

- Eclipse™
- Rational® Application Developer (Eclipse™ 上に構築される IBM IDE)
- Rational® Team Concert (Jazz.net の IBM 配布製品)

---

## テスト・プロジェクト用の特殊変数

プラグインを使用してプロジェクトのテスト・ビルドを実行するときに、いくつかの特殊な環境変数を使用することによって、システム内のファイルがサーバーにコピーされる前後にコマンドを実行するように指定できます。

すべてのコマンドはプロジェクト・ディレクトリーで実行されます。

- **PRECMD** 変数を使用して、開発者のコンピューターからビルドを実行しているサーバーにコピーされたディレクトリーおよびファイルに対してコマンドを実行します。このコマンドは、プロジェクト・ステップの前に実行されます。例: このコマンドを使用して、ソース制御システム内のファイルを、コピー前に確認することができます。
- **POSTCMD** 変数を使用して、プロジェクト・ステップが実行された後で、ディレクトリーおよびファイルに対してコマンドを実行します。例: このコマンドを使用して、(このような概念を使用する、Rational ClearCase などのソース制御システム内の) 検査済みの仮想ディレクトリーを解放することができます。

コマンドは、リフレクター・プラグイン内で Build Forge プロジェクト成果物としてマークされたディレクトリーおよびファイルに対して実行されます。これらのコマンドは、Reflector プラグインのディレクトリー・ツリーが全探索されるときに適用されます。

注: ディレクトリー・ツリーの全探索は、PRECMD コマンドの場合には幅優先で下方向に向かって実行され、POSTCMD コマンドの場合にはその逆になります。全探索中に、必要に応じてディレクトリー用のコマンドおよびファイル用のコマンドが実行されます。

### **\_PRISM\_DIR\_PRECMD**

ツリー全探索でディレクトリーが検出されたときに、そのディレクトリーに対して実行されるコマンドを指定します。このコマンドは、少なくとも 1 つのファイルを含むそれぞれのディレクトリーごとに、1 回実行されます。システムは、コマンドの最初の「\$1」を、ディレクトリー名に置き換えます。

### **\_PRISM\_FILE\_PRECMD**

ツリー全探索でファイルが検出されたときに、そのファイルに対して実行されるコマンドを指定します。このコマンドは、それぞれのファイルごとに、1 回実行されます。システムは、コマンドの最初の「\$1」を、ファイル名に置き換えます。

### **\_PRISM\_DIR\_POSTCMD**

ツリー全探索でディレクトリーが検出されたときに、そのディレクトリーに対して実行されるコマンドを指定します。このコマンドは、少なくとも 1 つのファイルを含むそれぞれのディレクトリーごとに、1 回実行されます。システムは、コマンドの最初の「\$1」を、ディレクトリー名に置き換えます。

### **\_PRISM\_FILE\_POSTCMD**

ツリー全探索でファイルが検出されたときに、そのファイルに対して実行さ



れるコマンドを指定します。このコマンドは、それぞれのファイルごとに、1 回実行されます。システムは、コマンドの最初の「\$1」を、ファイル名に置き換えます。

---

## Eclipse および Rational Application Developer のプラグイン

プラグインは、Eclipse™ および Rational® Application Developer IDE 内から管理コンソール機能へのアクセスを提供します。

Eclipse および Rational Application Developer 環境では、以下のプラグインが使用可能です。

### Frequency

Frequency プラグインを使用することで、開発者は以下の作業を実行できます。

- 1 つ以上の管理コンソールにアクセスし、プロジェクトを表示する
- ジョブを起動する
- ジョブの状況を表示する
- 実行中のジョブおよび完了したジョブのビルド・ログを表示する

### Reflector

Reflector プラグインは、ローカル環境内のファイルを使用してジョブを実行します。これらのジョブは通常、新規コードをテストしてから、他の開発者または実動ビルドが使用できるようにソース制御にチェックインするために実行されます。

Eclipse プラグイン・ユーザーは、プロジェクト環境変数の値をオーバーライドすることができます。Build Forge プロジェクトを開始すると、「ジョブ設定」ポップアップが表示されます。環境変数への変更はジョブのみに適用されます。プロジェクトのデフォルト変数値は、変更されません。

### Eclipse または Rational Application Developer でのプラグインの使用

プラグインをインストールした後、以下の方法でアクティブにすることができます。

- 管理コンソールにアクセスしてジョブを起動し、プロジェクト・ログを表示するには、Frequency プラグインを使用します。ご使用の IDE 内で、「ウィンドウ (Window)」 > 「パースペクティブを開く (Open Perspective)」 > 「その他 (Other)」を選択します。IDE に、パースペクティブ・タイプがリストされたダイアログ・ボックスが表示されます。Build Forge パースペクティブを選択します。システムに、「コンソール・エクスプローラー (Console Explorer)」、「ビルド情報 (Build Info)」、および「ビルド・ログ (Build Log)」ウィンドウが表示されます。「コンソール・エクスプローラー」を右クリックし、「新規コンソール」を選択して、管理コンソールへの接続を構成します。Frequency の使用について詳しくは、プラグインに提供されているオンライン・ヘルプを参照してください。

注: LDAP/Active Directory ドメインへのアクセスを構成する必要がある場合、ドメインの実際の名前ではなく、LDAP ドメイン・オブジェクトの Build Forge システム名を使用してください。

- テスト・ビルドを実行するには、Reflector プラグインを使用します。ご使用の IDE 内で、プロジェクトを選択し、右クリックすることによって、Reflector を構成します。ポップアップ・メニューから「プロパティー (Properties)」を選択します。「プロパティー (Properties)」ダイアログのプロパティー・オプションのリストで、「Build Forge プロジェクトの成果物」を選択します。ご使用のプロジェクトと連動させる Build Forge プロジェクトを使用してダイアログを構成し、システムにアップロードするファイルを選択します。詳しくは、プラグインに提供されているオンライン・ヘルプを参照してください。

注: Reflector プラグインには Frequency プラグインが必要になります。

## Eclipse または Rational Application Developer 用プラグインのインストール

ご使用の IDE 環境用のプラグインを Build Forge サーバーからインストールします。

### 始める前に

前提条件:

- Eclipse バージョン 3.0.2 以上または Rational Application Developer バージョン 7.0 以上
- Java 2 SE バージョン 5.0
- Build Forge システムが稼働している必要がある

### このタスクについて

プラグインをインストールするには、ご使用の IDE 内から以下の手順を実行してください。

### 手順

1. 「ヘルプ」 > 「ソフトウェア更新」 > 「検索およびインストール」と選択します。
2. 「インストールする新規フィーチャーを検索する (Search for new features to install)」ラジオ・ボタンをクリックして、「次へ」をクリックします。システムに「アクセスする更新サイト (Update Sites to Visit)」ダイアログが表示されます。
3. 「新規リモート・サイト (New Remote Site)」ボタンをクリックします。システムに「新規リモート・サイト (New Remote Site)」ダイアログが表示されます。
  - a. 名前フィールドに「Build Forge Update Site」と入力します。
  - b. ご使用の管理コンソール・コンピューターのホスト名を使用して、「URL」フィールドに次の更新サイト URL を入力します。 `http://<console_host_name>/prism/eclipse/updateSite/site.xml`

- Build Forge が稼働しているシステムと同じシステムで Eclipse を実行している場合、ホスト名として localhost を使用できます。
  - コンソールがポート番号 80 以外のポート上で実行されている場合、そのポート番号を含めます。例: `http://myhostname:11812/prism/eclipse/updateSite/site.xml`
- c. 「OK」をクリックします。
4. 「アクセスする更新サイト (Update sites to visit)」ダイアログで、「Build Forge Update Site」チェック・ボックスを選択して、「終了」をクリックします。
  5. システムは、「検索結果 (Search Results)」ダイアログに使用可能なプラグインのリストを表示します。提供されるすべてのプラグインを選択して、「次へ」をクリックします。
- 注: Reflector プラグインには Frequency プラグインが必要になります。単独でインストールすると、実行されません。
6. ご使用条件を読み、「使用条件の条項に同意します」を選択して、「次へ」をクリックします。
  7. フィーチャーをインストールする場所を選択します。新しい場所を追加するには、「新しい場所 (New Location)」をクリックして、希望する場所を表示します。
  8. 「完了」をクリックします。
  9. 「フィーチャー検証 (Feature Verification)」ダイアログが表示されたら、「インストール」をクリックします。このダイアログは、プラグインが署名されていないフィーチャーであるために表示されます。「すべてインストール」を選択しない限り、インストールするフィーチャーごとに 1 回ずつダイアログが表示されます。
  10. 変更を有効にするために Eclipse を再始動するように求められます。「はい」をクリックします。

## SSL が使用可能な場合の代替インストール

Build Forge システムで SSL が使用可能になっている場合に、代替のインストール方法を使用します。

### 始める前に

現行バージョンの Eclipse および Rational Application Developer では、SSL を使用できません。そのため、Eclipse および Rational Application Developer のプラグインは、SSL が使用可能である Build Forge システムからはインストールできません。次善策として、プラグイン・インストール・ファイルを非セキュア Web サーバー上で使用可能にするか、それらのファイルを手動でユーザーに配布します。

ファイルをパッケージするには、以下のようにします。

1. prism という名前のディレクトリーを作成します。
2. prism で、eclipse という名前のディレクトリーを作成します。
3. eclipse で、updateSite という名前のディレクトリーを作成します。

以下のファイルを、<bfinstall>/webroot/public/prism/eclipse/updateSite から、作成した updateSite ディレクトリーにコピーします。

- features ディレクトリー
- plugins ディレクトリー
- site.xml ファイル

## このタスクについて

prism ディレクトリーを使用可能にしたら、ユーザーは IDE 内から以下のステップを行います。

### 手順

1. 「ヘルプ」 > 「ソフトウェア更新」 > 「検索およびインストール」と選択します。
2. 「インストールする新規フィーチャーを検索する (Search for new features to install)」ラジオ・ボタンをクリックして、「次へ」をクリックします。システムに「アクセスする更新サイト (Update Sites to Visit)」ダイアログが表示されます。
3. 新規サイトを作成します。以下のいずれかの手順を選択します。

- リモート・サーバーからファイルを取得する場合

- a. 「新規リモート・サイト (New Remote Site)」ボタンをクリックします。システムに「新規リモート・サイト (New Remote Site)」ダイアログが表示されます。
- b. 名前フィールドに「Build Forge Update Site」と入力します。
- c. ファイルの場所を次のように入力します。

`http://host/path/prism/eclipse/updateSite/site.xml`

*host* は、Web サーバーのホスト名または IP アドレスです。

*path* は、サーバーのルートからファイルを置く場所へのパスです。

- d. 「OK」をクリックします。

- ローカル・ホストからファイルを取得する場合

- a. 「新規ローカル・サイト (New Local Site)」ボタンをクリックします。システムは「新規ローカル・サイト (New Local Site)」ダイアログを表示します。
- b. 名前フィールドに「Build Forge Update Site」と入力します。
- c. ファイルの場所を次のように入力します。

`file://path/prism/eclipse/updateSite/site.xml`

*path* には、ファイルの場所を指定します。

- d. 「OK」をクリックします。

4. 「アクセスする更新サイト (Update sites to visit)」ダイアログで、「Build Forge Update Site」チェック・ボックスを選択して、「終了」をクリックします。

5. システムは、「**検索結果 (Search Results)**」ダイアログに使用可能なプラグインのリストを表示します。提供されるすべてのプラグインを選択して、「**次へ**」をクリックします。

注: Reflector プラグインには Frequency プラグインが必要になります。単独でインストールすると、実行されません。
6. ご使用条件を読み、「**使用条件の条項に同意します**」を選択して、「**次へ**」をクリックします。
7. フィーチャーをインストールする場所を選択します。新しい場所を追加するには、「**新しい場所 (New Location)**」をクリックして、希望する場所を表示します。
8. 「**完了**」をクリックします。
9. 「**フィーチャー検証 (Feature Verification)**」ダイアログが表示されたら、「**インストール**」をクリックします。このダイアログは、プラグインが署名されていないフィーチャーであるために表示されます。「**すべてインストール**」を選択しない限り、インストールするフィーチャーごとに 1 回ずつダイアログが表示されます。
10. 変更を有効にするために Eclipse を再始動するように求められます。「**はい**」をクリックします。

## Eclipse および Rational Application Developer 用プラグインの使用

起動ジョブにアクセスして、プロジェクト・ログを表示するには (Frequency プラグイン)、以下の手順を実行します。

1. ご使用の IDE 内で、「**ウィンドウ (Window)**」 > 「**パースペクティブを開く (Open Perspective)**」 > 「**その他 (Other)**」を選択します。
2. 「**Build Forge**」パースペクティブを選択します。このパースペクティブには、以下のウィンドウが含まれます。
  - コンソール・エクスプローラー
  - ビルド情報
  - ビルド・ログ
3. 「**コンソール・エクスプローラー**」を右クリックし、「**新規コンソール**」を選択して、管理コンソールへの接続を構成します。
4. 「**Build Forge サービス層ホスト名**」にホスト名または IP アドレスを入力し、その他のフィールドを確認または編集して、「**OK**」をクリックします。

接続が確立されると、「**ビルド情報**」ウィンドウに実行可能なジョブが取り込まれます。ローカル・ファイルを使用してジョブを実行するには、使用するジョブと使用するファイルを構成します。

1. ご使用の IDE 内で、Build Forge に接続します。
2. 「**コンソール・エクスプローラー**」ウィンドウで、ジョブを右クリックして「**プロパティ**」を選択します。
3. 「**プロパティ**」で、「**Build Forge プロジェクトの成果物**」を選択します。
4. 「**Build Forge プロジェクトの成果物**」で、作業するプロジェクトを選択し、使用するローカル・ファイルを選択します。

詳しくは、プラグインに提供されているオンライン・ヘルプを参照してください。

---

## Rational Team Concert 用プラグイン

Rational Team Concert クライアント用のプラグインは、Rational Team Concert と Build Forge の統合のコンポーネントの 1 つです。サーバー拡張機能およびアダプター・テンプレートも必要です。Rational Team Concert との統合をセットアップすると、Rational Team Concert のユーザーが以下の作業を実行できるようになります。

- Build Forge を RTC ビルド・サーバーとしてセットアップする
- Build Forge プロジェクトを RTC ビルド定義としてセットアップする
- RTC クライアントからプロジェクトの表示、ジョブの実行、およびジョブ結果の表示を行う

統合のセットアップ手順については、623 ページの『第 33 章 Rational Build Forge と Rational Team Concert』を参照してください。

## Rational Team Concert プラグインの使用

ここでの説明では、Rational Team Concert 統合がセットアップされていて、ご使用の Rational Team Concert クライアントにプラグインがインストールされていることを前提としています。

統合のセットアップ手順については、623 ページの『第 33 章 Rational Build Forge と Rational Team Concert』を参照してください。

ビルド定義をセットアップしてビルドを実行するには、以下のようにします。

1. ビルド定義をセットアップします。
  - a. チーム成果物のビューで、プロジェクト・フォルダーを展開します。
  - b. 「ビルド」を右クリックして、「新規ビルド定義 (New Build Definition)」をクリックします。
  - c. 「新規ビルド定義 (New Build Definition)」で、「新規ビルドの作成 (Create a new build)」を選択して、「次へ」をクリックします。
  - d. 「一般情報」に、ビルド ID と説明を入力します。使用可能なテンプレート・リストから「**Rational Build Forge**」を選択します。「次へ」をクリックします。
  - e. 「追加構成 (Additional Configuration)」で、「一般 (General)」と「プロパティ」の両方を選択して、「終了」をクリックします。入力したビルド ID のラベルが付いたタブが作成されます。
  - f. 「**Build Forge**」タブをクリックします。
2. ビルド定義のプロジェクトを選択します。
  - a. 「Build Forge」タブをクリックします。Build Forge に接続するために必要な以下の情報を入力します。
    - ホスト名 - Build Forge が稼働するホストの名前。これは、「**コンソール URL**」システム設定が設定されている場合には、そのシステム設定と一致していなければなりません。「**管理**」 > 「**システム**」の設定にアクセスできない場合は、管理者にお問い合わせください。



- ポート - Build Forge との通信に使用されるポート。ポート 3966 がデフォルトです。「**Build Forge にセキュアに接続**」が選択されていれば、デフォルトでポート 49150 が表示されます。ご使用のインストールのポート番号が異なる場合は、入力してください。これは、「**コンソール URL**」システム設定が設定されている場合には、そのシステム設定のポート設定と一致していなければなりません。「**管理**」 > 「**システム**」の設定にアクセスできない場合は、管理者にお問い合わせください。
  - ユーザー名 - Build Forge への接続用のユーザー名。このユーザーは、Build Forge に既に存在している必要があります。
  - パスワード - ユーザー名のパスワード
  - パスワードの確認 - ユーザー名のパスワード
- b. 「**プロジェクトの取得**」をクリックします。
  - c. 表示される「**Build Forge プロジェクト**」リストから、このビルド定義のプロジェクトを選択します。
3. ビルドを要求します。
    - a. ビルド定義を右クリックして、「**ビルド要求 (Request build)**」を選択します。
    - b. 必要なビルド・オプションとビルド・プロパティを指定して、「**作成**」をクリックします。
  4. ビルド結果を確認します。
    - a. ビルドが終了したら、「**ビルド**」タブのリストから選択します。ウィンドウが表示されます。
    - b. 「**外部リンク (External Links)**」で、「**Build Forge 結果**」リンクをクリックします。
    - c. Build Forge のログイン・パネルが表示されます。ログインします。
    - d. 「**ホーム**」 > 「**完了した実行**」と進み、リストからビルドを選択します。
    - e. ビルド・ステップおよび結果が表示されます。ステップのログを表示するには、ステップ・リンクをクリックします。
    - f. 完了したら、「**ログアウト**」をクリックしてウィンドウを閉じます。

## Rational Team Concert プラグインのトラブルシューティング

このセクションでは、既知の問題とその対処方法について説明します。

### 「期限経過」の状況のジョブ

ジョブの状況が数分を超えて「期限経過」である場合、そのジョブはハングしている可能性があります。

#### 回避策

Build Forge エンジン进行を停止し、再始動してください。289 ページの『Perl エンジンの始動と停止』を参照してください。





---

## 第 31 章 Rational Build Forge と Perforce

BuildForge は、ソース・コード管理のために Perforce と協働することができます。

このトピックでは、Perforce との統合のための追加セットアップ要件について説明し、用意されているサンプル・アダプター・テンプレートの特性を要約します。

Perforce のアダプター・テンプレート・サンプルでは、ベースラインに対する変更内容を分析する方法が提供されます。通常、変更分析はビルド回避のために使用されます。ベースライン・コンポーネントが変更されていない場合、再ビルドは行われません。

### Perforce アダプターのための追加セットアップ要件

一般要件については、552 ページの『アダプター要件』を参照してください。

1. エージェントと同じホスト上に P4 コマンド行クライアントをインストールします。
2. Build Forge 用に使用する Perforce ユーザー・アカウントをセットアップします。
3. クライアントにアクセスできるようにエージェント環境変数をセットアップします。

### Perforce のアダプター・テンプレート・サンプル

以下のアダプター・テンプレート・サンプルが用意されています。

#### PerforceByDate

1. アダプターの実行以降生じた変更がないか、Perforce クライアントを照会します。
2. 次の情報を BOM レポートに書き込みます: 変更、日付、時刻、ユーザー、Perforce クライアント、およびコメント。
3. 変更詳細 (diff コマンド出力による) を BOM レポートに書き込みます。

このアダプターで設定される変数は以下のとおりです。

- LAST\_RUN
- CurDate
- P4PORT
- BFCLIENT
- P4CLIENT
- FILESPEC

この環境に必須の追加変数は以下のとおりです。

- P4USER
- P4PASSWD

これらの変数に「非表示の割り当て」プロパティを使用することはできません。ユーザーおよびパスワードは、ステップ・ログに平文で書き込まれます。

#### **PerforceByRev**

1. 直前のリポジトリ改訂以降生じた変更がないか、Perforce クライアントを照会します。
2. 次の情報を BOM レポートに書き込みます: 変更、日付、時刻、ユーザー、Perforce クライアント、およびコメント。
3. 変更詳細 (diff コマンド出力による) を BOM レポートに書き込みます。

このアダプターで設定される変数は以下のとおりです。

- LAST\_RUN
- CurDate
- P4PORT
- BFCLIENT
- P4CLIENT
- FILESPEC

この環境に必須の追加変数は以下のとおりです。

- P4USER
- P4PASSWD

これらの変数に「非表示の割り当て」プロパティを使用することはできません。ユーザーおよびパスワードは、ステップ・ログに平文で書き込まれます。

---

## 第 32 章 Rational Build Forge と Rational Quality Manager

Rational Build Forge は Rational Quality Manager と協働することができます。

Rational® Build Forge® を Rational Quality Manager と統合して、Rational Quality Manager からソフトウェアをビルドしてパッケージ化するプロジェクトを定義して実行することができます。

統合は Rational Quality Manager で行います。Rational Quality Manager のバージョンについては、インフォメーション・センターを参照してください。リンクについては、Rational Quality Manager の情報ページを参照してください。

- Rational Quality Manager バージョン 2 の場合は、『統合』 > 『**Rational Quality Manager** およびソフトウェア・インベントリーと自動化ツール』 > 『**Rational Build Forge** との統合 (Integrating with Rational Build Forge)』を参照してください。
- Rational Quality Manager バージョン 1 の場合は、『統合』 > 『ソフトウェア・インベントリーおよびプロビジョニング・ツールとの統合』 > 『**Rational Build Forge** との統合』を参照してください。



---

## 第 33 章 Rational Build Forge と Rational Team Concert

Build Forge は Rational Team Concert と統合することができます。

統合がセットアップされると、Rational Team Concert のユーザーは以下を行うことができます。

- Build Forge を RTC ビルド・サーバーとしてセットアップする
- RTC クライアントからプロジェクトの表示、ジョブの実行、およびジョブ結果の表示を行う
- Build Forge プロジェクトを RTC ビルド定義としてセットアップする

RTC 3.0 との統合に関する詳細情報については、[jazz.net](http://jazz.net) 上の Build Forge Team Wiki (Rational Build Forge wiki on jazz.net) を参照してください。

---

### Rational Team Concert バージョン 3.x、バージョン 4.x

Rational Team Concert™ をインストールした環境で、統合コンポーネントを構成します。

Rational Team Concert 3.x または 4.x を Rational Build Forge バージョン 7.1.1.3 以降に統合すると、Rational Team Concert の以前のバージョンを統合した場合に比べて大きな利点があります。Rational Build Forge との統合に使用するプラットフォームとして、Rational Team Concert 3.x または 4.x をお勧めします。

Rational Build Forge と Rational Team Concert をインストールしたら、Rational Team Concert で統合を構成してください。通常、2 つの製品はそれぞれ別のホストにインストールされます。

統合の構成およびトラブルシューティングを行う場合は、[jazz.net](http://jazz.net) の Rational Build Forge wiki に記載されている手順を参照してください。





---

## 第 34 章 Rational Build Forge と StarTeam

BuildForge は、ソース・コード管理のために StarTeam と協働することができます。

このトピックでは、StarTeam との統合のための追加セットアップ要件について説明し、用意されているサンプル・アダプター・テンプレートの特性を要約します。

StarTeam のアダプター・テンプレート・サンプルでは、ベースラインに対する変更内容を分析する方法が提供されます。通常、変更分析はビルド回避のために使用されます。ベースライン・コンポーネントが変更されていない場合、再ビルドは行われません。

### StarTeam アダプターのための追加セットアップ要件

一般要件については、552 ページの『アダプター要件』を参照してください。

1. エージェントと同じホスト上に StarTeam コマンド行クライアントをインストールします。
2. Build Forge 用に使用する StarTeam ユーザー・アカウントをセットアップします。
3. クライアントにアクセスできるようにエージェント環境変数をセットアップします。

### StarTeam のアダプター・テンプレート・サンプル

以下のアダプター・テンプレート・サンプルが用意されています。

#### StarTeamBaseline

1. フォルダを照会して StarTeam ビューを見つけ、ファイルに関する情報を収集します。
2. 次の情報を BOM レポートに書き込みます: ファイル名、状況、改訂、および分岐。

このアダプターで設定される変数は以下のとおりです。

- USER
- PASS
- HOST
- PORT
- PROJECT
- VIEW
- DIR
- BFCLIENT

この環境に必須の追加変数は以下のとおりです。

- P4USER
- P4PASSWD

これらの変数に「非表示の割り当て」プロパティを使用することはできません。ユーザーおよびパスワードは、ステップ・ログに平文で書き込まれます。

#### **StarTeamByDate**

1. StarTeam API を使用して、StarTeam ビューを照会し、現在日付と前のアダプター実行時の間の変更を特定します。
2. 次の情報を BOM レポートに書き込みます: 変更されたファイルおよびディレクトリー、ユーザー、バージョン、日付、および変更コメント。
3. 変更詳細 (diff コマンド出力による) を BOM レポートに書き込みます。

このアダプターで設定される変数は以下のとおりです。

- STA\_USER
- STA\_PASS
- STA\_HOST
- STA\_PORT
- STA\_PROJECT
- STA\_VIEW
- STA\_DIR
- BFCLIENT
- LASTRUN
- CURDATE
- EMAILCHANGES
- STARTEAM80JAR

この環境に必須の追加変数は以下のとおりです。

- P4USER
- P4PASSWD

これらの変数に「非表示の割り当て」プロパティを使用することはできません。ユーザーおよびパスワードは、ステップ・ログに平文で書き込まれます。

---

## 第 35 章 Rational Build Forge とサブバージョン

BuildForge は、ソース・コード管理のためにサブバージョン (SVN) と協働することができます。

このトピックでは、SVN との統合のための追加セットアップ要件について説明し、用意されているサンプル・アダプター・テンプレートの特性を要約します。

SVN のアダプター・テンプレート・サンプルでは、ベースラインに対する変更内容を分析する方法が提供されます。通常、変更分析はビルド回避のために使用されます。ベースライン・コンポーネントが変更されていない場合、再ビルドは行われません。

### SVN アダプターのための追加セットアップ要件

一般要件については、552 ページの『アダプター要件』を参照してください。SVN のための追加要件はありません。

### SVN のアダプター・テンプレート・サンプル

以下のアダプター・テンプレート・サンプルが用意されています。

#### SubversionByDate

1. 過去の日付と現在の日付の間で生じたりポジトリの変更がないか、サブバージョンを照会します。
2. 次の情報を BOM レポートに書き込みます: 変更タイプ、改訂、ユーザー、ファイルまたはディレクトリー、および変更日。
3. 次の情報を BOM レポートに書き込みます: ファイル名、状況、改訂、および分岐。

使用される変数は以下のとおりです。

- SVN\_CLIENT
- SVN\_OPTS
- SVN\_REPOSITORY
- SVN\_LAST\_REV
- SVN\_LAST\_DATE

#### SubversionByRev

このアダプターのセットアップ方法について説明している記事が [jazz.net](https://jazz.net/wiki/bin/view/Main/RationalBuildForge/SVNAdaptorExample) (<https://jazz.net/wiki/bin/view/Main/RationalBuildForge/SVNAdaptorExample>) にあります。

1. 現在の改訂と前の改訂の間で生じたりポジトリへの変更がないか、サブバージョンを照会します。
2. 各変更に関して、次の情報を BOM レポートに書き込みます: 改訂、ユーザー、変更タイプ、ファイルまたはディレクトリー・パス、および変更日。

3. 変更詳細 (diff コマンド出力による) を BOM レポートに書き込みます。

使用される変数は以下のとおりです。

- SVN\_CLIENT
- SVN\_OPTS
- SVN\_REPOSITORY
- SVN\_LAST\_REV
- SVN\_LAST\_DATE

---

## 第 36 章 Rational Build Forge と WebSphere 製品

このセクションでは、Build Forge® を WebSphere 製品と統合する方法について説明します。

- Apache Tomcat ではなく WebSphere Application Server を使用して、Build Forge サービスおよび Quick Report を実行
- Web アプリケーション・サーバーとして Apache ではなく IBM HTTP Server (IHS) を使用

---

### Apache Tomcat の代わりに WebSphere Application Server を使用

用意されている Apache Tomcat アプリケーション・サーバーの代わりに、WebSphere Application Server を Web アプリケーション・サーバーとして使用することができます。Build Forge サービスおよび Build Forge ヘルプ・アプリケーションを実行するように構成します。

WebSphere Application Server における構成には、以下の 3 つの領域があります。

- Build Forge サービスおよびヘルプ・アプリケーションを実行するように構成します。これは必須です。
- Java 2 セキュリティーを Build Forge サービス・アプリケーション用に構成します。Java 2 セキュリティーを WebSphere Application Server で実行する場合、これは必須です。
- Build Forge における SSL およびパスワード暗号化のサポートを構成します。これらのフィーチャーを Build Forge で構成した場合、これは必須です。

#### 前提条件

- 「Detailed system requirements for a specific product」サイトを調べてください。ご使用の Rational Build Forge エディションおよびバージョンを照会し、どのバージョンの WebSphere Application Server がサポートされているかを確認します。
- ライセンス・サーバーの制約により、Build Forge コンソールおよび WAS は、同一のオペレーティング・システムおよびハードウェア・プラットフォームで稼働する必要があります。
- Build Forge サービス層のデプロイを試行する前に、WebSphere Application Server に最新のフィックスパックが適用されていることを確認してください。

**重要:** この構成を使用するときは、Build Forge を開始する前に、Build Forge サービスと Build Forge Help アプリケーションを WebSphere Application Server で開始する必要があります。これらは自動的に開始されません。

### Build Forge® アプリケーションを実行するように Websphere Application Server を構成

サービス・アプリケーションとヘルプ・アプリケーションを実行するよう、WebSphere Application Server を構成します。

## このタスクについて

この手順にあるステップを WebSphere Application Server 管理コンソールで実行します。

WebSphere Application Server バージョン 8.0 では、WebSphere Application Server 管理コンソールのユーザー・インターフェースが多少変更されました。以下の説明は、バージョン 7 以前に対応しています。この手順に影響する、バージョン 8.0 における違い:

- アプリケーションにアクセスするには、「**アプリケーション (Applications)**」 > 「**アプリケーション・タイプ (Application Types)**」 > 「**エンタープライズ・アプリケーション (Enterprise Applications)**」をクリックします。
- アプリケーションを選択した後に、コンテキスト・ルートを設定します。

注: 「リソース用の MBean の作成」のチェック・ボックスは表示されません。

## 手順

1. WebSphere Application Server 管理コンソールを開きます。コンソールの URL:
  - `http://<was_host>:<was_port>/ibm/console`。デフォルト・ポートは 9060 です。
  - `https://<was_host>:<was_port>/ibm/console`。デフォルト・ポートは 9043 です。WebSphere Application Server 管理セキュリティが有効である場合はこの URL を使用します。
2. ご使用の JDBC ドライバーのサポートを構成します。
  - a. 新規変数 `RBF_JDBC_DRIVER_PATH` を作成します。変数は、「**環境変数**」 > 「**WebSphere 環境変数 (WebSphere Variables)**」で作成します。そのスコープは、WebSphere Application Server ノードおよびサーバーです。使用するデータベース・ドライバの JAR ファイルが入っているディレクトリーに値を設定します。
  - b. マスター構成に加えた変更を保存します。
  - c. WebSphere Application Server を停止および再始動して、新規変数を有効にします。
  - d. 新規共用ライブラリー `RBF_JDBC_LIBRARY` を作成します。

ライブラリーは、「**環境変数**」 > 「**共用ライブラリー (Shared Libraries)**」で作成します。作成した `RBF_JDBC_DRIVER_PATH` を使用して、JDBC デバイス・ドライバの JAR ファイル名を追加します。リストする JAR ファイルが複数ある場合は、各 JAR ファイルをそれぞれ独自の行に追加します (DB2 に関する下記の説明と同様)。

以下は、MySQL データベース・ドライバの場合の例です。

```
${RBF_JDBC_DRIVER_PATH}¥mysql-connector-java-5.0.5-bin.jar
```

DB2 の場合の例です。

```
${RBF_JDBC_DRIVER_PATH}¥db2jcc.jar  
${RBF_JDBC_DRIVER_PATH}¥db2jcc_license_cu.jar
```

UNIX または Linux の場合は、上記の例で、円記号 (¥) の代わりにスラッシュ (/) を使用します。

- e. マスター構成に加えた変更を保存します。
3. Build Forge<sup>®</sup> アプリケーション WAR ファイルをインストールします。
- 「アプリケーション」 > 「エンタープライズ・アプリケーション (Enterprise Applications)」を開きます。
  - 「インストール (Install)」をクリックします。
  - jas.war ファイルを参照します。 jas をコンテキスト・ルートとして使用します。
    - Windows システム  
`<bfinstall>%PrepForExternal%jas.war`
    - UNIX システムおよび Linux システム  
`<bfinstall>/Platform/PrepForExternal/jas.war`
  - 「次へ」をクリックし、「リソース用の MBean の作成 (Create MBeans for resources)」チェック・ボックスが選択されている場合はそのチェック・ボックスをクリアします。
  - 「次へ」をクリックし、「完了」をクリックします。
  - インストール・テキストの下部で「保存」をクリックします。
4. Build Forge<sup>®</sup> ヘルプ WAR ファイルをインストールします。
- 「アプリケーション」 > 「エンタープライズ・アプリケーション (Enterprise Applications)」をクリックします。
  - 「インストール (Install)」をクリックします。
  - Build Forge<sup>®</sup> インストール・ディレクトリー内の BuildForgeHelp.war ファイルを参照します。 /BuildForgeHelp をコンテキスト・ルートとして使用します。
  - 「次へ」をクリックし、「リソース用の MBean の作成 (Create MBeans for resources)」チェック・ボックスが選択されている場合はそのチェック・ボックスをクリアします。
  - 「完了」ボタンが表示されるまで「次へ」をクリックし、「完了」をクリックします。
  - インストール・テキストの下部で、「保存」をクリックします。
5. 共有ライブラリーの参照として RBF\_JDBC\_LIBRARY を設定します。
- 「アプリケーション」 > 「エンタープライズ・アプリケーション (Enterprise Applications)」をクリックします。
  - 「jas.war」リンクをクリックします。
  - 「共有ライブラリーの参照 (Shared library references)」をクリックします。
  - 「サービス層ログイン・サーブレット (A Services Layer Login Servlet)」ボックスを選択します。
  - 「参照共有ライブラリー (Reference Shared Libraries)」をクリックします。
  - リストに RBF\_JDBC\_LIBRARY を追加します。
6. クラス・ローダー順序を設定します。



- a. 「モジュールの管理 (Manage Modules)」の下で、「サービス層ログイン・サーブレット (A Services Layer Login Servlet)」を選択し、ドロップダウン・メニューから「クラス・ローダー順序 (Class loader order)」を選びます。
  - b. 値を「ローカル・クラス・ローダーによってロードされたクラスから先に (親が最後) (Classes loaded with local class loader first (parent last))」に変更します。
7. マスター構成に加えた変更を保存します。
  8. Build Forge<sup>®</sup> 構成ファイルを調べます。 ファイル内の `services_url` エントリが正しい URL を指していることを確認します。 WebSphere Application Server を使用している場合、`buildforge.conf` におけるこのプロパティの値は、`https://<hostname>:9443/jas` や `https://<hostname>:9080/jas` のようになります。 `buildforge.conf` ファイルは、以下の場所にあります。
    - Windows システム  
`<bfinstall>%`
    - UNIX システムおよび Linux システム  
`<bfinstall>/Platform/`
  9. UNIX および Linux システムの場合は、`bin` ディレクトリー内のすべてのファイルに対する、`jas` の実行アクセス権を追加します。  
  
 この例では、プロファイル `AppSrv01` が使用されています。 ご使用のプロファイル名が `AppSrv01` 以外の場合は、ご使用のプロファイル名を使用してください。  
  

```
chmod -R 755 <WAS-install>/AppServer/profiles/AppSrv01/installedApps/<node>/jas_war.ear/jas.war/bin
```
  10. ライブラリー `PATH` 変数を更新して、以下のディレクトリーを組み込みます。
    - Windows システムでは、`PATH` に以下を組み込みます。  
`<bfinstall>%token_libs`
    - AIX システムでは、`LIBPATH` に以下を組み込みます。  
`<bfinstall>/Platform/token_libs`
    - Linux/Solaris では、`LD_LIBRARY_PATH` に以下を組み込みます。  
`<bfinstall>/Platform/token_libs`
  11. 64 ビットの Linux または WebSphere Application Server を使用している場合は、以下の手順を実行して、提供されている 64 ビット・ライブラリーをコピーします。
    - a. ライブラリー・ファイルの名前を変更します。
      - Windows システム: `<bfinstall>%token_libs%rc1_rational.dll` を `<bfinstall>%token_libs%rc1_rational.dll.old` に名前変更します。
      - UNIX システムおよび Linux システム: `<bfinstall>/Platform/token_libs/librc1_rational.so` を `<bfinstall>/Platform/token_libs/librc1_rational.so.old` に名前変更します。
    - b. 64 ビット・ライブラリー・ファイルを `token_libs_64` から `token_libs` にコピーします。 コマンドの途中で改行しないでください。ここでは、読みやすくなるように改行を使用しています。

- Windows システム

```
copy <bfinstall>%token_libs_64%rc1_rational.dll
<bfinstall>%token_libs%rc1_rational.dll
```

- UNIX システムおよび Linux システム

```
cp <bfinstall>/Platform/token_libs_64/librc1_rational.so
<bfinstall>/Platform/token_libs/librc1_rational.so
```

## WebSphere Application Server で、Java 2 セキュリティーを Build Forge 用に構成

Java 2 セキュリティーを有効にして WebSphere Application Server を実行している場合、Build Forge サービス層で Java 2 セキュリティーを使用するように構成する必要があります。

### このタスクについて

WebSphere Application Server 管理コンソールを構成するには、以下の手順を実行します。

#### 手順

1. 「アプリケーション」 > 「エンタープライズ・アプリケーション (Enterprise Applications)」を開きます。
2. 「jas\_war」を選択し、「更新」をクリックします。
3. 「単一ファイルの置換または追加 (Replace or Add a Single File)」を選択します。
4. 「インストールしたアプリケーション・アーカイブ・ファイルで始まり、置換または追加されるファイルに続くパスを指定 (Specify the path beginning with the installed application archive file to the file to be replaced or added)」で、META-INF/was.policy を入力します。
5. 「ローカル・ファイル・システム」を選択し、ご使用のオペレーティング・システムの was.policy ファイルを参照して、「次へ」をクリックします。

- Windows システム

```
<bfinstall>/samples/projects/was.policy
```

- UNIX システムおよび Linux システム

```
<bfinstall>Platform/samples/projects/was.policy
```

6. 「OK」をクリックします。
7. マスター構成に加えた変更を保存し、Build Forge サービス・アプリケーションを停止して始動します。

## WebSphere Application Server で、SSL およびパスワード暗号化サポートを Build Forge 用に構成

WebSphere Application Server で、SSL またはパスワード暗号化、あるいはその両方を Build Forge アプリケーション用に有効にします。

## このタスクについて

SSL およびパスワード暗号化のサポートを有効にするには、構成がさらに必要になります。これらのいずれか、または両方を構成することができます。SSL を Build Forge で有効にする前に、SSL を WebSphere Application Server で構成しておく必要があります。

### 手順

1. `<bfinstall>%keystore` ディレクトリーを  
`<WAS_INSTALL_DIR>%AppServer%profiles%AppSrv01` にコピーします。AppSrv01 以外のプロファイルを使用している場合、keystore ディレクトリーをそのプロファイル・ディレクトリーにコピーしてください。

パスは、UNIX システムでも Linux システムでも同じです。このディレクトリーには、アプリケーション・サーバー・プロファイルが入っています。ご使用の WebSphere Application Server のバージョンに応じて、ディレクトリーは異なる場合があります。

2. WebSphere コンソールにログインします。WebSphere 管理用のコンソールを使用してください。
3. 「カスタム・プロパティー」ページに進みます。「サーバー」 > 「サーバー・タイプ」 > 「WebSphere Application Servers」 > 「server1」 > 「Java およびプロセス管理」 > 「プロセス定義」 > 「Java 仮想マシン」 > 「カスタム・プロパティー」を開きます。

デフォルト・サーバー名は **server1** です。ご使用のサーバーの名前が異なる場合、その名前を使用してください。

4. SSL を有効にします。bfclient.conf ファイルを指すようにカスタム・プロパティーを設定します。

- Windows

```
com.buildforge.client.config=<bfinstall>%bfclient.conf
```

- UNIX または Linux

```
com.buildforge.client.config=<bfinstall>/Platform/bfclient.conf
```

5. パスワード暗号化を有効にします。bfpassword.conf ファイルを指すようにカスタム・プロパティーを設定します。

- Windows

```
com.buildforge.password.encryption.file=<bfinstall>%bfpassword.conf
```

- UNIX または Linux

```
com.buildforge.password.encryption.file=<bfinstall>/Platform/bfpassword.conf
```

6. WebSphere Application Server を再始動します。再始動したときに変更は有効になります。

## WebSphere Application Server を使用して Build Forge アプリケーションを実行するときの始動順序

Build Forge を始動する前に、WebSphere Application Server で Build Forge アプリケーションを始動します。

## このタスクについて

Build Forge を始動する前に、WebSphere Application Server からサービスおよびヘルプ・アプリケーションを始動します。

### 手順

1. WebSphere Application Server 管理コンソールを開きます。
2. 「アプリケーション」>「エンタープライズ・アプリケーション (Enterprise Applications)」を選択します。
3. 「jas.war」および「BuildForgeHelp.war」を選択します。
4. 「開始」をクリックします。
5. Build Forge を開始します。

---

## Apache HTTP Server の代わりに IBM HTTP Server を使用

デフォルトでインストールされている Apache HTTP Server の代わりに、IBM HTTP Server (IHS) を構成して、管理コンソールで使えるようにすることができます。

### 始める前に

- ライセンス・サーバーの制約により、Build Forge コンソールおよび IBM HTTP Server は、同一のオペレーティング・システムおよびハードウェア・プラットフォームで稼働する必要があります。
- 提供されているプリコンパイル済み PHP モジュールを使用してください。自分でコンパイルした PHP を使用しないでください。リリース・ノートを調べて、ご使用のオペレーティング・システム用のモジュールが実際に組み込まれていることを確認してください。そうではない場合、サポート担当者と一緒にモジュールをコンパイルする必要があります。
- このセットアップで障害が発生する場合は、IBM サポートにお問い合わせください。
- 例では、以下のディレクトリーが IBM HTTP Server インストール・ディレクトリーとして使用されます。ご使用のインストール・ディレクトリーと異なる場合は、置き換えてください。
  - Windows: C:\Program Files\IBM\HTTPServer
  - Unix および Linux: /opt/IBM/HTTPServer

## このタスクについて

このセクションでは、以下の作業を行う方法について説明します。

1. プリコンパイルされた PHP ファイルをセットアップします。
2. Build Forge Web アプリケーションを指すように、IBM HTTP Server 構成ファイルを変更します。
3. データベースへのアクセスにプロキシ・サーバーを使用する場合には、プロキシ・サーバーを使用できるように PHP を変更します。
4. SSL を使用する場合には、SSL 経由で Build Forge を使用できるよう IHS を構成します。

5. Windows システム: IBM HTTP Server インストール・ディレクトリーにある libeay32.dll を置き換えてください。
6. AIX システム: 追加のセットアップ

## プリコンパイルされた PHP ファイルをセットアップする このタスクについて

この作業では、提供された PHP ファイルを見つけ、指定の場所にそれらをコピーする方法について説明します。ファイルは以下のディレクトリーにあります。

- Windows システム  
`<bfinstall>%ihs_modules`
- UNIX システムおよび Linux システム  
`<bfinstall>/Platform/ihs_modules`

### 手順

1. PHP モジュールをコピーします。 必要に応じて `modules` ディレクトリーを作成します。
  - Windows システム  

```
cd <bfinstall>%ihs_modules%modules%
copy *.dll C:%Program Files%IBM%HTTPServer%modules
```
  - UNIX システムおよび Linux システム  

```
cd <bfinstall>/Platform/ihs_modules/modules
cp libphp5.so /opt/IBM/HTTPServer/modules
```
2. PHP 拡張をコピーします。 必要に応じて `extensions` ディレクトリーを作成します。
  - Windows システム  

```
cd <bfinstall>%ihs_modules%lib%php%extension%
copy *.dll C:%Program Files%IBM%HTTPServer%extensions
```
  - UNIX システムおよび Linux システム  

```
cd <bfinstall>/Platform/ihs_modules/lib/php/extensions/no-debug-zts-20090626
cp *.so /opt/IBM/HTTPServer/extensions
```
3. 提供された `php.ini` ファイルをコピーします。 必要に応じて `conf` ディレクトリーを作成します。
  - Windows システム  

```
cd <bfinstall>%ihs_modules
copy php.ini C:%Program Files%IBM%HTTPServer%conf
```
  - UNIX システムおよび Linux システム  

```
cd <bfinstall>/Platform/ihs_modules
cp php.ini /opt/IBM/HTTPServer/conf
```
4. `conf` ディレクトリーにコピーした `php.ini` ファイルを編集して、`extensions` ディレクトリーへのポインターを入力します。 次の行を追加するか、または置き換えます。
  - Windows システム  

```
extension_dir = C:%Program Files%IBM%HTTPServer%extensions
```
  - UNIX システムおよび Linux システム

```
extension_dir = /opt/IBM/HTTPServer/extensions
```

5. php.ini ファイルを編集し、ご使用のシステムの情報で更新します。 Apache HTTP Server 用の php.ini ファイルをガイドとして使用することができます。このファイルは、次の場所にあります。

- Windows システム

```
<bfinstall>%Apache%php%php.ini
```

- UNIX システムおよび Linux システム

```
<bfinstall>/server/apache/conf/php.ini
```

6. 使用されているすべてのディレクトリーに対する書き込み権限を php.ini に追加します。 UNIX および Linux システムでは、次のコマンドをディレクトリーに対して実行してください。

```
chmod -R 777
```

## IBM HTTP Server 構成ファイルの編集

### 手順

1. サーバー・インストール済み環境の conf ディレクトリーで、IBM HTTP Server (IHS) 用の httpd.conf ファイルを見つけます。
2. 以下の例に示すように、Build Forge Web アプリケーションを指すように DocumentRoot 設定を変更します。この例では、Build Forge インストール・ディレクトリーは /opt/buildforge です。

```
<VirtualHost *:80>
    ServerAdmin build@yourdomain.com
    DocumentRoot /opt/buildforge/webroot/public
    ServerName ausbuild01.yourdomain.com
    ServerAlias build.yourdomain.com mc.yourdomain.com #optional server aliases
    ErrorLog logs/ausbuild.error_log
    CustomLog logs/ausbuild.access_log common
</VirtualHost>
```

3. PHP 関連項目を追加します。

- Windows システム

```
LoadModule php5_module "C:%Program
Files%IBM%HTTPServer\modules%php5apache2_2.dll"
```

```
<IfModule dir_module>
    DirectoryIndex index.html index.php
</IfModule>
```

```
AddType application/x-httpd-php .php
```

```
# PHPIniDir is the location of the php.ini file
PHPIniDir "C:%Program Files%IBM%HTTPServer%conf"
```

- UNIX システムおよび Linux システム

```
LoadModule php5_module "/opt/IBM/HTTPServer/modules/libphp5.so"
```

```
<IfModule dir_module>
    DirectoryIndex index.html index.php
</IfModule>
```

```
AddType application/x-httpd-php .php
```

```
# PHPIniDir is the location of the php.ini file
PHPIniDir "/opt/IBM/HTTPServer/conf"
```

4. デフォルト・ユーザーに関するステートメントを変更します。
  - AIX システムを使用している場合は、User daemon と Group daemon ではなく User daemon と Group staff を使用します。
  - Linux システムの場合は、<bfinstall>/webroot/templates\_c フォルダー内の templates\_c ファイルに対してステートメントをユーザー・デーモンとグループ・デーモンに変更します。例えば、Linux システムの場合は、chown daemon:daemon /opt/buildforge/webroot/templates\_c のようにします。
5. templates\_c ファイルのアクセス権を変更します。 Linux システムでは daemon: daemon を、AIX システムでは daemon: staff を使用してください。Linux システムでの例は次のとおりです。
 

```
chown daemon:daemon /opt/buildforge/webroot/templates_c
```
6. 必要に応じて IHS ポート番号を変更します。デフォルト・ポート番号は 80 です。その他の必要な変更をすべて httpd.conf に加えます。

## PHP でのプロキシー・サーバーの識別

### このタスクについて

このステップは、管理コンソールがプロキシー・サーバーを使用してデータベースにアクセスしなければならない場合にのみ実行します。

#### 手順

PHP 構成ファイル php.ini を編集します。 このファイルは <php-install>/lib にあります。例えば /usr/local/php-5.3.6 などです。  
次の項目を追加します。

```
bf_proxyHost=<使用するプロキシー・サーバーのホスト名>
bf_proxyPath=<プロキシーのパス>
bf_symlinkPath=<symlink からプロキシーへのパス>
```

## IHS 用 SSL の構成

### このタスクについて

IHS を Build Forge<sup>®</sup> と連携させるには、通常の SSL セットアップ以外に、追加の要件があります。

#### 手順

1. ご使用のパスに Build Forge<sup>®</sup> ツール・ディレクトリーを組み込みます。
  - Windows システム
    - *bfinstall\ibmjdk\bin*
    - *bfinstall\openssl*
  - UNIX システムまたは Linux システム
    - *bfinstall/server/ibmjdk/bin*
2. ご使用のライブラリー・パスに Build Forge<sup>®</sup> ツール・ディレクトリーを組み込みます。
  - AIX システム (LIBPATH)
    - *bfinstall/openssl*



- UNIX システムまたは Linux システム (LD\_LIBRARY\_PATH)  
– *bfinstall/openssl*

3. Build Forge® キーを PKCS12 から CMS に変換します。最新バージョンの GSKIT ツールを使用してください。gsk7¥bin (Windows) または bin (UNIX または Linux) で、以下のコマンドを実行します (見やすくするために改行が挿入されています)。

```
gsk7cmd -keydb
        -convert
        -db bfinstall¥keystore¥buildForgeKeyStore.p12
        -pw buildForgeKeyStore_password
        -old_format pkcs12
        -new_format cms
```

4. stash ファイルにパスワードを保管します。IHS はこのファイルを使用して、始動中にパスワードを取得します。ファイルがない場合、IHS はパスワードを求めるプロンプトを出します。最新バージョンの GSKIT ツールを使用してください。gsk7¥bin (Windows) または bin (UNIX または Linux) で、以下のコマンドを実行します (見やすくするために改行が挿入されています)。

```
gsk7cmd -keydb
        -stashpw
        -db bfinstall¥keystore¥buildForgeKeyStore.kdb
        -pw buildForgeKeyStore_password
```

5. httpd.conf を変更します。Windows では、以下の項目を組み込みます。ユーザーに https のみでアクセスを行わせたい場合は、Listen 80 をコメント化する必要があります。

```
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
# Listen 80
Listen 0.0.0.0:443
<VirtualHost *:443>
SSLEnable
SSLClientAuth None
SSLProtocolDisable SSLv2
SSLServerCert buildforge
KeyFile bfinstall¥keystore¥buildForgeKeyStore.kdb
SSLStashFile bfinstall¥keystore¥buildForgeKeyStore.sth
ErrorLog bfinstall¥Apache¥logs¥ssl_error.log
TransferLog bfinstall¥Apache¥logs¥transfer.log
</VirtualHost>
```

UNIX および Linux では、以下の項目を組み込みます。ユーザーに https のみでアクセスを行わせたい場合は、Listen 80 をコメント化する必要があります。

```
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
# Listen 80
Listen 0.0.0.0:443
<VirtualHost *:443>
SSLEnable
SSLClientAuth None
SSLProtocolDisable SSLv2
SSLServerCert buildforge
KeyFile bfinstall/Platform/keystore/buildForgeKeyStore.kdb
SSLStashFile bfinstall/Platform/keystore/buildForgeKeyStore.sth
ErrorLog bfinstall/server/apache/logs/ssl_error.log
TransferLog bfinstall/server/apache/logs/transfer.log
</VirtualHost>
```

*bfinstall* は、インストール・ディレクトリーのルートです。上記のステップに関しては、Unix と Linux の場合は、正しいディレクトリーにアクセスできるよう

に /Platform を *bfinstall* に付け足す必要があります。また、ディレクトリーの区切り文字として円記号 (¥) ではなくスラッシュ (/) を使用する必要があります。

## タスクの結果

詳しくは、SSL のセットアップに関する IHS 資料を参照してください。

## Windows: DLL の置換

### このタスクについて

バージョン 8.0 よりも古い IBM HTTP Server を使用している場合、DLL を Rational® Build Forge® に用意されているもので置き換える必要があります。

### 手順

IBM HTTP Server インストール・ディレクトリーの複数の場所にある既存の libeay32.dll ファイルを見つけ、置き換えます。

- <bfinstall>/openssl に用意されている libeay32.dll を使用してください。
- DLL のインスタンスをすべて検索し、それらを置き換えます。例えば、Windows 2003 のロケーションは次のようになります。

```
<ihs_install>%Plugins%gsk7%gsk7_32%icc%osslib  
<ihs_install>%gsk7%icc%osslib
```

## タスクの結果

DLL を置き換えた後、HTTP サーバーは正常に起動できるようになります。置き換えないと、HTTP サーバーは起動しません。

## AIX システム: 追加のセットアップ

AIX システムの場合、追加のセットアップが必要になります。

### 手順

- AIX 6 システム:
  - <bfinstall>/Platform/his\_modules/libxml2.a を /opt/IBM/HTTPServer/modules にコピーします。
  - /opt/IBM/HTTPServer/modules を LIBPATH に追加します。
- AIX 7 システム:
  - <bfinstall>/Platform/his\_modules/libxml2.a および <bfinstall>/Platform/his\_modules/libxml2.so.2 を /opt/IBM/HTTPServer/modules にコピーします。
  - /opt/IBM/HTTPServer/modules を LIBPATH に追加します。
  - 次の行を httpd.conf に追加します。  
LoadFile "/opt/IBM/HTTPServer/modules/libxml2.so.2"

---

## 第 37 章 API の処理

Build Forge では、Java クライアント API および Perl クライアント API を使用できます。

クライアント・ファイルは `<bfinstall>/webroot/public/clients/` に保管されています。

実行中の管理コンソールでそのファイルにアクセスすることができます。クライアント・ダウンロード・ディレクトリーは次の URL からアクセスできます:

`http://<hostname>:<portnumber>/clients/`

### Build Forge への API のアクセス

API を使用したプログラムは、サービス層と直接通信します。サービス層は、Apache Tomcat 上のアプリケーションです。インストール時に、Apache Tomcat サーバーは特定のポートで listen するように構成されます。API が Build Forge と通信するためには、これらのポートが開いている必要があります。デフォルトでは、ポートは次のように設定されています。

- 3966 (非セキュア)
- 49150 (セキュア - SSL 有効)

API クライアント用の SSL 構成は、`bfclient.conf` にあります。クライアントには、このファイルが組み込まれている必要があります。SSL を有効にする場合、クライアントが Build Forge と通信するためには、鍵ストアおよび証明書が必要になります。詳しくは、127 ページの『API クライアント (Perl または Java) の SSL の有効化』を参照してください。

---

## API プログラム用の Build Forge ユーザーの作成

プログラムが認証に使用するユーザーを、管理コンソールで作成します。

### このタスクについて

API プログラムが管理コンソールへのログインに使用するユーザーを作成します。そのユーザーが機能するかを、ログインしてテストします。

プログラムは、コンソールへのアクセス時に、必ずユーザー資格情報を使用してコンソールに対する認証を行う必要があります。

認証が行われた後、ユーザー・セッション用のセッション ID が生成され、データベースに保管されます。プログラムが既存ユーザーと同じログインを使用すると、そのユーザー・セッションは閉じられます。

ユーザー資格情報を使用できるスレッドまたはプロセスは 1 つのみです。別のスレッドまたはプロセスが同じユーザー資格情報を使用してセッションを確立しようとすると、最初に存在していたセッションは終了します。

注: LDAP/Active Directory 認証によって提供されるユーザーは使用しないでください。 管理コンソールでユーザーを作成します。

---

## Java クライアント API

Java クライアント API を使用して、管理コンソールへアクセスする Java プログラムを作成します。

Java クライアント API を使用して作成されたプログラムは、クライアント・ホスト上で稼働し、管理コンソール上のデータにアクセスします。 Java クライアント API は、管理コンソール・オブジェクトに命令を与える管理コンソール・オブジェクト・メソッドを定義するクラスを含む JAR ファイルで構成されます。

Java クライアント API を使用するには、Java SDK 1.5 または 1.6 が必要です。

資料は JavaDocs で提供されます。

注: プログラムが認証に使用する管理コンソール上に、Build Forge ユーザーを定義する必要があります。

## Java クライアント API パッケージの取得

Java クライアント・ソフトウェア・パッケージは、管理コンソール・ホストからダウンロードできます。

### 手順

Java API をダウンロードするには、以下のようにします。

1. クライアント・ダウンロード・ディレクトリーにアクセスします。 Web ブラウザーで、以下の URL にアクセスしてください。  
`http://<hostname>:<portnumber>/clients/`
2. JAR ファイルを保存します。 Java クライアントの下で、「JAR ファイル」リンクを右クリックし、「名前を付けてリンクを保存 (Save Link As)」を選択します。JAR ファイルを保存する場所を指定します。
3. JavaDocs を保存します。 Java クライアントの下で、「JavaDoc 参照 ZIP (JavaDoc reference ZIP)」リンクを右クリックし、「名前を付けてリンクを保存 (Save Link As)」を選択します。JAR ファイルを保存する場所を指定します。

### タスクの結果

管理コンソールを使用して資料にアクセスできます。「クライアント・ダウンロード・ディレクトリー」ページの Java クライアントの下で、「JavaDoc 参照 (JavaDoc reference)」をクリックします。

## Java クライアント API のセットアップ

Java API をクライアント・ホストに配置して、Java SDK が使用するようにセットアップします。

## このタスクについて

ホストは、管理コンソール・ホストに対するクライアントとして機能します。

注: TLS v1.1 または TLS v1.2 が有効になっている場合に、セキュア接続を使用して Build Forge® に接続するには、Java クライアントを JDK v1.7 以上によって開始する必要があります。

### 手順

1. .JAR ファイルを任意の場所に入れます。
2. CLASSPATH を更新します。CLASSPATH を、rbf-services-client-java.jar を入れたディレクトリーを含むように設定します。

---

## Perl クライアント API

Perl クライアント API を使用して、管理コンソールへアクセスする Perl プログラムを作成します。

Perl クライアントとは、管理コンソールのデータ・オブジェクトの抽象へのアクセスを提供する Perl モジュールとメソッドのセットです。

Perl クライアントのモジュールに関する資料は、クライアント API パッケージ内に 2 つの形式で含まれています。

- ファイル: apidoc.txt
- POD (Plain Old Documentation) 形式の Perl 文書。詳しくは、<http://www.perl.org> のオンライン資料を参照してください。

Perl クライアントを使用するには、以下を行う必要があります。

- 管理コンソール・コンピューターから、Perl クライアント・パッケージを取得する。
- そのパッケージをインストールする (Perl がインストールされていない場合は、Perl もインストールする)。

注: プログラムが認証に使用する管理コンソール上に、Build Forge ユーザーを定義する必要があります。

## Perl クライアント API パッケージの取得

管理コンソール・ホストから Perl クライアント API をダウンロードすることができます。

### 手順

Perl クライアント API をダウンロードするには、以下のようになります。

1. クライアント・ダウンロード・ディレクトリーにアクセスします。Web ブラウザーで、以下の URL にアクセスしてください。

`http://<hostname>:<portnumber>/clients/`

2. ZIP ファイルを保存します。Perl クライアントの下で、「ZIP ファイル」リンクを右クリックし、「名前を付けてリンクを保存 (Save Link As)」を選択します。ZIP ファイルを保存する場所を指定します。
3. 資料を保存します。Perl クライアントの下で、「PerlDoc 参照 tar.gz (PerlDoc reference tar.gz)」リンクを右クリックし、「名前を付けてリンクを保存 (Save Link As)」を選択します。ZIP ファイルを保存する場所を指定します。ファイルを unzip して、各モジュールの資料にアクセスします。

## タスクの結果

管理コンソールを使用して資料にアクセスできます。「クライアント・ダウンロード・ディレクトリー」ページの Perl クライアントの下で、「PerlDoc 参照 (PerlDoc reference)」をクリックします。

## Perl クライアント API のセットアップ

Perl クライアント API を使用するには、アプリケーションを実行するホスト上にその API をセットアップする必要があります。

### このタスクについて

ホストは、管理コンソール・ホストに対するクライアントとして機能します。

### 手順

1. ActiveState の ActivePerl バージョン 5.8.4 以降などのクライアント・ホスト上に、Perl インタープリターをインストールします。次の Perl 前提条件モジュールが必要です (ActivePerl バージョン 5.8.8 には含まれています)。

- Exporter
- LWP::UserAgent
- HTTP::Request

Perl モジュールのインストールについて詳しくは、Perl 資料を参照してください。

2. ダウンロードした Perl クライアント API パッケージを一時ディレクトリーに解凍します。
3. apidoc.txt ファイルで説明されているように、Perl クライアント API を標準 Perl 配布としてインストールします。

Windows の場合には nmake 1.5 が必要です。これは、Visual Studio に含まれているほか、Microsoft の Web サイトからダウンロードすることができます。これは、C:\Windows など、PATH 環境変数によって見つけることのできる場所にインストールする必要があります。Perl クライアント・パッケージを解凍した一時ディレクトリーで、以下のコマンドを実行します。

```
perl Makefile.PL
nmake
nmake install
```

UNIX または Linux システムの場合は、次のコマンドを実行します (または、Windows 上の Cygwin 環境の場合)。

```
perl Makefile.PL  
make  
make install
```

インストールが完了すると、上部 Perl クライアント・モジュールは `BuildForge::Services::DB0` となります。詳しくは、各モジュールの PerlDoc を参照してください。





---

## 第 38 章 管理コンソールのバージョン番号の判別

使用している管理コンソールのバージョンを確認するには、ページの左上隅にある Rational Build Forge のロゴの上にマウス・カーソルを置いてください。ポップアップのツールチップに、バージョン番号が表示されます。



---

## 第 39 章 診断

このセクションでは、Build Forge が正しく機能しない原因の判別について説明します。

以下のトピックを使用して、Rational® Build Forge® に関する問題の診断に役立ててください。問題が解決しない場合は、IBM® ソフトウェア・サポートに連絡し、支援を求めてください。また、追加情報について、以下の関連トピックのリンクを参照することもできます。

### 関連概念:

220 ページの『ログの場所およびセットアップ』

---

## Build Forge サービス層のデバッグ診断

この診断ツールでは、Java 仮想マシン (JVM) のすべてのスレッドの動作を示すスレッド・ダンプを 5 秒ごとに受け取ることができます。この診断ツールは、特にスレッド・デッドロックまたは JVM の予期しないシャットダウンが発生した場合に役立つデバッグ・ユーティリティです。

### このタスクについて

**重要:** デバッグを目的としたスレッド・ダンプが必要な場合を除いて、この診断ツールを有効にしないでください。

サービス層デバッグ診断ツールを有効にして使用するには、以下の手順に従ってください。

### 手順

1. アプリケーション・サーバーのサービス層を停止します。
2. enableBFSLThreadDump という名前の新規ファイルを当該ディレクトリーに作成します。
  - Windows: <bfinstall>/tmp
  - Linux: <bfinstall>/Platform/tmp
3. アプリケーション・サーバーのサービス層を開始します。開始時に、Build Forge サービス層は、enableBFSLThreadDump ファイルと同じディレクトリーに threadDump{Timestamp} という名前のファイルを作成します。
4. 新しいスレッド・ダンプのスナップショットを取得するには、以下に示すように enableBFSLThreadDump ファイルを更新します。
  - UNIX/Cygwin: コマンド touch enableBFSLThreadDump を入力します。
  - Windows: テキスト・エディターでファイルを開いてファイルを更新し、保存します。

---

## Build Forge のデフォルトのログ

このトピックでは、Build Forge® のデフォルトのログをリストします。これらのログは、さまざまなエラーや問題の有無を確認するために使用します。

ログ	説明
<bfinstall>/logs	Java エンジンの新しいログ・ファイル。
<bfinstall>/db.log	データベースおよび古い Perl エンジンのログ・ファイル。
<bfinstall>/server/apache/logs (UNIX/Linux) または <bfinstall>/Apache/logs (Windows)	アプリケーション・サーバー Apache のログ・ファイル。
<bfinstall>/server/tomcat/logs (Linux/UNIX) または <bfinstall>/Apache/tomcat/logs (Windows)	アプリケーション・サーバー Tomcat のログ・ファイル。

---

## Java エンジン・ログの構成

Java エンジン・ログのデフォルト設定を変更することができます。

Java エンジンの新しいログ・ファイルは、<bfinstall>/logs ディレクトリーにあります。各ログ・ファイルのデフォルト・サイズの上限は 20 MB であり、ファイルの最大数は 5 つで、デフォルトのロギング・レベルは **INFO** です。

デフォルト設定を変更するには、com.ibm.jas-1.0.jar のコメントを使用して logging.properties ファイルを更新します。 jar ファイルは、アプリケーション・サーバー・ディレクトリーにインストールされています。

---

## Build Forge エージェント・アクティビティー・ログの有効化

Build Forge エージェントに関する情報を収集するには、ロギングを有効にする必要があります。

### 始める前に

エージェントのロギングを有効にする前に、以下の前提条件を満たしていることを確認します。

- ・パスは bfagent.conf 内になければなりません。
- ・エージェントのデフォルトを変更するための書き込みアクセス権が必要です。

### このタスクについて

ロギング情報は、パスで指定されたファイルに追加されます。ファイルのサイズに上限はありません。Build Forge エージェント・アクティビティー・ログの設定は、エージェントのデバッグを行うために一時的に使用することを目的としています。稼働中のエージェントの永続ログとして使用するためのものではありません。ログ・ファイルの削除は手動で行う必要があります。

### 手順

1. Build Forge エージェント・アクティビティー・ログをオンにするには、bfagent.conf にある既存のパスを使用して、activity\_log をアンコメントします。

2. エージェントを再始動します。

---

## パフォーマンス問題の診断

Build Forge でパフォーマンス上の問題が発生した場合は、問題の診断および解決にこのトピックにある情報を使用することができます。

- データベースのパフォーマンス – データベース・ストレージとリソースの使用状況を確認します。
- ネットワーク通信 – Build Forge コンソールがデータベースおよびエージェントに使用しているネットワーク帯域幅を確認します。
- セキュリティーの向上 – 複雑なアルゴリズムがある場合は、確認します。Build Forge コンソールおよびエージェント上でセキュリティーを有効にしていると、アルゴリズムが複雑な場合にパフォーマンスに影響を及ぼす場合があります。
- ロギングによる影響 – 拡張ロギングが有効になっているかどうかを確認します。このロギングはパフォーマンスに影響する場合があります。拡張ロギングは、デバッグ目的でのみ一時的に使用します。
- ハードウェアによる影響 – ハードウェアでの変更点を確認します。CPU、メモリー、ストレージなどのハードウェアを変更すると、パフォーマンスに影響する可能性があります。





---

## 第 40 章 パフォーマンスとスケーラビリティ

システムは、さまざまな負荷状況の下で機能するように設計されています。高負荷の状態ではシステムを稼働させるには、構成およびリソース割り振りに関して何らかのスケーリングが必要です。

追加のリソースとして、以下の記事を参照可能です。 33 ページの『第 5 章 インストールの計画』も参照してください。

- [Deploying the IBM Rational Build Forge Management Console: Getting results and performance](#): この記事には、計画およびデプロイメントの各段階についての説明が記載されており、リソースの機能およびチューニングに関する推奨事項も紹介されています。
- [Rational Build Forge Performance test results: Evaluating the improved performance in 7.1.2 relative to 7.1.1.4](#): この記事には、負荷が増えていく状況下で、さまざまなシステム機能が通常はどのように動作するかが記載されています。オペレーティング・システムとデータベースのさまざまな組み合わせに関する情報が記載されています。ここでは、バージョン 7.1.1.4 と比較してバージョン 7.1.2 のパフォーマンスが向上していることが示されています。



## 第 41 章 製品によりインストールされる実行可能なコマンド

下の表で、Rational® Build Forge® で提供され、使用される実行可能コマンドについて説明します。

Windows の場合、コマンド・ファイルは、Build Forge® インストール・ディレクトリー (デフォルトでは、C:\Program Files\IBM\Build Forge) にあります。

UNIX および Linux の場合、コマンド・ファイルは <bfinstall>/Platform ディレクトリーにあります。<bfinstall> はデフォルトでは /opt/buildforge です。

注: 管理コンソールを z/Linux で実行している場合、コマンドを実行するには .pl 拡張子を指定する必要があります。

実行可能なコマンドのバージョン番号を表示する場合は、-v オプションを使用します。コマンドは、実行可能なコマンドがインストールされているディレクトリーから実行しなければなりません。

```
bfproject -v
```

すべてのコマンドにおいて、-v オプションを指定すると、以下の例に示されているように、コマンド名とバージョン番号が表示されます。

```
bfproject.exe 8.0.0.0-0-0306
```

実行可能ファイル	サービスであるかどうか	説明
bfdbdump	N	bfdbdump.exe が、このコマンドを実行して、データベースをパージします。
bfproject	N	buildforge.exe が、このコマンドを実行して、ジョブを開始します。
bfengine	Y	このコマンドは、buildforge.exe および Web サーバーを開始します。Windows 専用です。
bfexport	N	このユーティリティーは、データベースからデータをエクスポートするために使用します。
bf bomexport	N	このユーティリティーは、データベースから BOM をエクスポートするために使用します。
bfimport	N	このユーティリティーは、データベースにプロジェクト・データをインポートするために使用します。
bfstepcmd	N	bfproject は、長時間実行されているステップに対してこのコマンドを実行し、それらのステップにそれぞれ別個のプロセスを作成します。

実行可能ファイル	サービスで あるかどうか	説明
buildforge	N	このコマンドは、ビルド・プロセス、パージ・プロセス、およびスケジュール・プロセスを管理します。
bfdispatch	Y	このコマンドは、エージェント・サービスを開始します。Windows 専用です。
bfpwencrypt	N	パスワード暗号化ユーティリティー。

---

## 第 42 章 用語集

このトピックでは、このシステム全体で使用される概念および用語の定義を示します。

---

### アクセス・グループ (access group)

権限、通知、および LDAP グループのプロパティを共有する、ユーザーの集合。アクセス・グループを LDAP グループにマップすることができる。グループをネストすることもできる。ユーザーは、属しているグループの権限を継承する。

---

### アダプター (adaptor)

アダプターは、ソース制御システム、デバッグ・データベース、テスト・システムなどの外部システムと、Build Forge が相互作用できるようにするアドオンである。例えば、ソース・コード・アダプターを使用すると、システムは、IBM® Rational® ClearCase®、Perforce、Visual SourceSafe、および CVS などのソース制御システムでの変更をモニターおよびトラッキングして、それらの変更内容を基にしてアクションを実行することができる。アダプターを構成して、部品構成表 (BOM) でストレージの情報を収集したり、他の情報システムに情報をプッシュし直したりできる。

---

### エージェント (agent)

Build Forge® システムのコンポーネントの 1 つ。エージェントは、システム内でサーバー・リソースとして定義するコンピューターにインストールする必要がある。各エージェントは、管理コンソールと通信し、ステップに定義されたコマンドを実行する。また、エージェントは、ステップを実行した結果からの出力をまとめて、ステップ・ログに戻す。

---

### アーカイブ (archive)

出力ファイルが削除されたが、データベース内にはまだデータが残っているジョブのリスト。「ジョブ」パネルでこのリストを表示する。

---

## BOM

完了したジョブに関するデータのリスト。BOM は Bill of Materials (部品構成表) の頭字語である。ジョブを表示しているときに「BOM」タブに表示され、個々のステップの実行に関するデータは「ステップ」タブに表示される。BOM には、ジョブのステップについての情報と、その結果得られたファイルへの変更点についての情報が含まれる。一般的な使用法の 1 つとしては、ソース・ファイルへの変更の監査が必要な、ソフトウェア・ビルドにおけるソース・コード・アダプターでの使用

が挙げられる。 `.scan` コマンドを使用することで、ソースへの変更のベースラインを設定してからチェックポイントを設定し、最後の `.scan` コマンド以降の変更を要約できる。

---

## クラス (class)

グローバル・プロパティを持つプロジェクトのグループ化。これらのプロパティを使用して、完了したジョブを管理する。一般的には、定期的にそれらのジョブを削除したり、特定のクリーンアップ・タスクを実行する別のジョブを開始したりする。

---

## 強制削除 (clobber)

プロジェクトとそれに関連したすべてのジョブをデータベースから削除すること。

---

## コレクター (collector)

サーバー・リソースから収集する情報またはサーバー・リソースに割り当てる情報を決定するオブジェクト。情報は、コレクター内のプロパティによって指定される。サーバーに割り当てられたコレクターは、サーバーのマニフェストの仕様として機能する。コレクターは、「サーバー」 > 「コレクター」パネルで定義する。

---

## データベース (database)

データベースには、管理コンソールに入力されたすべての情報が保管される。また、システムがプロジェクトを実行する際やユーザー・アクションをログに記録する際にシステムによって作成されたデータもデータベースに保管される。

---

## ダイナミック (dynamic)

実行時または処理中に発生するイベントに付随する用語。

---

## エンジン (engine)

システムのコンポーネントの 1 つ。エンジンは、管理コンソールから入力され、データベースに保管された情報を使用して、プロジェクト実行の制御、通知用 E メール送信、および (サーバーで実行中の) エージェントとの通信を行う。

---

## 環境変数 (environments)

環境変数は、変数リストのコンテナである。環境変数は、サーバー、プロジェクト、およびステップに明示的に割り当てることができる。ステップの環境変数は、サーバー環境変数、プロジェクト環境変数、ステップ環境変数をこの順序で適用することによって構成される。これらの環境変数のうちの複数に変数が現れる場合、最後に指定された値が取られる。

---

## ハンドシェーク (handshake)

Secure Sockets Layer セッションの開始時に行うメッセージ交換のこと。これにより、クライアントは、公開鍵手法を使用してサーバーを認証し（および、任意で、サーバーがクライアントを認証し）、クライアントとサーバーが共同で、暗号化、暗号解除、および改ざんの検出のための対称鍵を作成することができます。

---

## インターセプター (interceptor)

Web サービスが着信メッセージの認証に使用するハンドラー。Build Forge では、インターセプターは、シングル・サインオンを実装するために提供されています。

---

## インターフェース (interface)

インターフェースは、アダプター・テンプレートのインスタンスである。アダプターを使用するには、インターフェースを作成する（そして編集する）必要がある。オリジナルのアダプター・テンプレートは、未変更のままである。また、インターフェースには、それぞれが個別の実行可能なアクションである、複数の <interface> エレメントを含むことができる点に注意する。

---

## ジョブ (job)

実行中のプロジェクトのインスタンス。システムは、完了したジョブごとに、ステップ・ログおよび BOM データを含むデータを保管する。

---

## ライブラリー (library)

ライブラリーは、作業の実行可能定義である。複数のステップで構成される。その動作は、プロパティーを使用して制御される。プロジェクトと異なる点は、実行場所となるサーバーを判別するためのセクターがないことである。ライブラリーは、プロジェクト内のステップから呼び出される。

---

## Lightweight Directory Access Protocol

TCP/IP を使用して、X.500 モデルをサポートするディレクトリーへのアクセスを提供するオープン・プロトコル。より複雑な X.500 Directory Access Protocol (DAP) のリソース要求は課されません。例えば、LDAP を使用して、インターネットまたはイントラネット・ディレクトリーに要員、組織、およびその他のリソースを配置することができます。

---

## マニフェスト (manifest)

コレクターによって収集された、サーバーに関するデータのリスト。マニフェスト・データは、セクターによってサーバーの選択に使用される。サーバーのマニフェストは自動的に更新される。手動で更新することもできる。「サーバー」 > 「*servername*」でサーバーを表示中に、「キュー・マニフェストの更新」ボタンを使用する。



---

## 管理コンソール (Management Console)

システムを調整するために単一のコンピューターにインストールされる、システムのコンポーネント。ユーザーは管理コンソールにログインして、プロジェクトを定義および開始したり、結果とレポートを表示したりする。管理コンソールは、エージェントに対し、ジョブを完了するための指示を出す。

---

## 通知テンプレート (notification template)

通知テンプレートは、特定のイベントの発生時にアクセス・グループへ送信される E メールの内容と形式を定義する。システムには、多くのデフォルト・テンプレートが用意されている。テンプレートは、編集したり、プロジェクトに固有の新しいものを作成したりすることができる。

---

## プラグイン (plug-in)

既存のプログラム、アプリケーション、またはインターフェースに機能を追加する、個別にインストール可能なソフトウェア・モジュール。

---

## プロジェクト (project)

プロジェクトは、作業の実行可能定義である。複数のステップで構成される。その動作は、プロジェクトのプロパティを使用して制御される。プロジェクトには、プロジェクトを実行できるサーバー（複数可）を特定するセクターが関連付けられる。プロジェクトには、独自の環境変数を割り当てることができる。実行中のプロジェクトはジョブである。セクターが割り当てられていないプロジェクトは、ライブラリーである。

---

## セクター (selector)

プロジェクトまたはステップが実行されるサーバーを選択する、プロジェクトまたはステップに関連付けられたオブジェクト。セクター内のプロパティによって、サーバーの選択方法が決定される。セクターは静的情報を使用できる。例えば、セクターはサーバー名を指定できる。サーバーは、動的情報を使用することもできる。例えば、セクターは、CPU タイプ、ディスク・サイズ、現在の負荷など、指定されたプロパティを持つサーバーを指定できる。実行時に、システムはセクターを使用して、一致するサーバーのリストを作成し、それらのサーバーのいずれかにプロジェクトまたはステップを割り当てる。「管理」 > 「セクター」パネルでセクターを定義する。プロジェクトを定義する前に、少なくとも 1 つのセクターを定義する必要がある。

---

## セマフォ (semaphore)

アクティビティが同時に発生するのを防ぐ、システム内のグローバル・フラグ。各セマフォは、システムが管理するラベルである。通常、リソースを排他使用する必要があるプロジェクトまたはステップは、セマフォを取得して、その排他使用を実現する。

`.semget` コマンドを使用して、ステップ内にセマフォを設定する。このセマフォは、`.semput` コマンドによって別のステップでリリースされる。セマフォを取得したら、他のステップはそのセマフォを取得できない。セマフォを取得しようとするステップは、セマフォがリリースされるまで待機する。

プロジェクトが完了すると、プロジェクトが使用したすべてのセマフォがシステムによって自動的にリリースされる。システム・エラーが原因でジョブが終了するような場合には、セマフォがリリースされないことがある。その場合、手動でリリースできる。

---

## サーバー (server)

Build Forge では、サーバーはホストに関連付けられたオブジェクトである。サーバー・リソース とも呼ばれる。プロジェクトまたはステップはホスト上で実行される。使用するサーバーは、プロジェクトまたはステップに関連付けられたセクターによって定義される。

Build Forge で、サーバーとして使用できるようにコンピューターをセットアップするには、以下を行う必要があります。

- エージェントをコンピューターにインストールする (詳しくは 157 ページの『第 10 章 エージェントのインストール』を参照)。
- 管理コンソールを使用して、サーバー・リソースを作成する。

「サーバー」パネルでサーバー・リソースを定義する。

---

## サービス (service)

システムのコンポーネントの 1 つ。クライアントとデータベースの間の抽象化層としても機能するため、サービス層とも呼ばれる。クライアントには、システム自体、および付属の Java API または Perl API を使用して構成されたクライアントが含まれる。

---

## スナップショット (snapshot)

ある時点におけるバックアップ・データの記録。

---

## 静的 (static)

あらかじめ決められた時間または定時に発生するオペレーションに付随する用語。

---

## ステップ (step)

ステップは、プロジェクトまたはライブラリーのコンポーネントである。これには、実行するコマンド行が 1 つ以上含まれる。ステップに関連付けられたセクターにより、使用するサーバーが決定される。何も指定されなければ、プロジェクトのセクターが使用される。ステップのプロパティによって、ステップの実行方

法および出力の処理方法が決定される。プロジェクトまたはライブラリーを作成するか、編集するときにステップを定義する。

---

## ステップ・ログ (step log)

完了したジョブ内の完了したステップに関するデータのリスト。ジョブを表示しているときに、「**ステップ**」タブにステップ・ログが表示される。各ステップに関する情報は列に表示される。「**ジョブ**」 > ジョブ名 と選択すると、ステップのリストが表示される。ステップ・ログを表示するステップをクリックする。

---

## スレッド (threading)

関連するさまざまなトランザクションを同時実行するための方法。

---

## ユーザー (user)

システム内のログイン。システムは、ユーザー設定および権限設定の独自セットを保守する。実動インストールにおいて、ユーザー管理に LDAP が使用され、LDAP エントリーおよびグループがシステムにマップされる。ユーザーには、アクセス・グループが関連付けられ、システム・リソースにアクセスするための特定の権限が付与される。