



HP Application Lifecycle Intelligence

ソフトウェアバージョン: 2.9
HP ALM 12.50

ユーザーズ・ガイド

ドキュメント・リリース日: 2015 年 9 月 (英語版)
ソフトウェア・リリース日: 2015 年 9 月

ご注意

保証

HP製品、またはサービスの保証は、当該製品、およびサービスに付随する明示的な保証文によってのみ規定されるものとします。ここでの記載で追加保証を意図するものは一切ありません。ここに含まれる技術的、編集上の誤り、または欠如について、HPはいかなる責任も負いません。

ここに記載する情報は、予告なしに変更されることがあります。

権利の制限

機密性のあるコンピュータソフトウェアです。これらを所有、使用、または複製するには、HPからの有効な使用許諾が必要です。商用コンピュータソフトウェア、コンピュータソフトウェアに関する文書類、および商用アイテムの技術データは、FAR 12.211および12.212の規定に従い、ベンダーの標準商用ライセンスに基づいて米国政府に使用許諾が付与されます。

著作権について

© Copyright 2003 - 2015 Hewlett-Packard Development Company, L.P.

商標について

Adobe™ は、Adobe Systems Incorporated (アドビシステムズ社) の登録商標です。

Microsoft®, Windows® は、Microsoft Corporation の米国登録商標です。

UNIX® は、The Open Groupの登録商標です。

ドキュメントの更新情報

このマニュアルの表紙には、以下の識別情報が記載されています。

- ソフトウェアのバージョン番号は、ソフトウェアのバージョンを示します。
- ドキュメントリリース日は、ドキュメントが更新されるたびに更新されます。
- ソフトウェアリリース日は、このバージョンのソフトウェアのリリース期日を表します。

最新の更新のチェック、またはご使用のドキュメントが最新版かどうかの確認には、次のサイトをご利用ください。

<http://support.openview.hp.com/selfsolve/manuals>

このサイトを利用するには、HP Passportへの登録とサインインが必要です。HP Passport IDの取得登録は、次のWebサイトから行なうことができます。

<http://h20229.www2.hp.com/passport-registration.html> (英語サイト)

または、HP Passportのログインページの [New users - please register] リンクをクリックします。

適切な製品サポートサービスをお申し込みいただいたお客様は、更新版または最新版をご入手いただけます。詳細は、HPの営業担当にお問い合わせください。

サポート

HPソフトウェアサポートオンラインWebサイトを参照してください。 <http://support.openview.hp.com>

HPソフトウェアが提供する製品、サービス、サポートに関する詳細情報をご覧ください。

HPソフトウェアオンラインではセルフソルブ機能を提供しています。お客様の業務の管理に必要な対話型の技術支援ツールに素早く効率的にアクセスいただけます。HPソフトウェアサポートWebサイトのサポート範囲は次のとおりです。

- 関心のある技術情報の検索
- サポートケースとエンハンスメント要求の登録とトラッキング
- ソフトウェアパッチのダウンロード
- サポート契約の管理
- HP サポート窓口の検索
- 利用可能なサービスに関する情報の閲覧
- 他のソフトウェアカスタマーとの意見交換
- ソフトウェアトレーニングの検索と登録

一部を除き、サポートのご利用には、HP Passportユーザとしてご登録の上、ログインしていただく必要があります。また、多くのサポートのご利用には、サポート契約が必要です。HP Passport IDの登録は、次の場所で行います。

<http://h20229.www2.hp.com/passport-registration.html> (英語サイト)

アクセスレベルに関する詳細は、以下のWebサイトにアクセスしてください。

http://support.openview.hp.com/access_level.jsp

HP Software Solutions Nowは、HPSWのソリューションと統合に関するポータルWebサイトです。このサイトでは、お客様のビジネスニーズを満たすHP製品ソリューションを検索したり、HP製品間の統合に関する詳細なリストやITILプロセスのリストを閲覧することができます。このサイトのURLは<http://h20230.www2.hp.com/sc/solutions/index.jsp>です。

目次

第1章: ALI の紹介	7
ALI の概要	8
ALI のワークフロー	9
第2章: 始める前に	11
ALI 拡張機能の有効化	12
旧バージョンからの移行	12
同期スケジュールの設定	13
TFS 統合サポートの設定	14
第3章: ビルド・システムとの統合のセットアップ	16
ビルド・システムのセットアップ・ワークフロー	17
Hudson/Jenkins との統合	18
HP ALI Hudson/Jenkins プラグインのデプロイメント	18
HP ALI Hudson/Jenkins プラグインの設定	20
TFS ビルド・サーバとの統合	21
ビルド・サーバの追加	21
ビルド構成の追加	22
ビルド構成にある SCM 構成の再利用	23
ビルド構成の不具合フィルタの設定	24
ビルド変更の検出の設定	24
ALI プロジェクト・リストのカスタマイズ	25
コード問題の追跡	26
第4章: ソース管理システムとの統合のセットアップ	28
SCM のセットアップのワークフロー	29
SCM との統合の前提条件と制限事項	30
SCM エージェント	30
SCM リポジトリの追加	31

Subversion リポジトリの追加	32
CVS リポジトリの追加	32
Perforce リポジトリの追加	33
TFS リポジトリの追加	34
Git リポジトリの追加	35
外部リポジトリ・ビューアの設定	36
コミット・パターンの設定	38
コード変更と代替 ID との関連付け	41
ブランチの追加とリリースへの関連付け	42
ブランチのチェックイン・ポリシーの設定	43
コード変更の検出の設定	45
第5章: Force.com との統合のセットアップ	46
前提条件	47
プロジェクトのデプロイメント, テスト, レポート生成のセットアップ	47
Hudson/Jenkins - Force.com の構成	49
第6章: SCM の変更とトレーサビリティの管理	51
SCM の変更とトレーサビリティのワークフロー	52
コード変更の表示	52
コード変更の影響の表示	53
プロジェクト・レポートの生成	54
グラフの生成	54
第7章: ビルド・アクティビティのモニタ	56
ビルドの表示	57
ビルド・レポートの表示	57
ビルド・グラフの生成	58
ビルドとテストのトレーサビリティの表示	59
第8章: 開発アクティビティのモニタ	60
リリース・モジュールでの開発アクティビティのモニタ	60
要件モジュールでの開発アクティビティのモニタ	60

不具合モジュールでの開発アクティビティのモニタ	61
付録A: サポート対象のシステム	62
サポート対象の SCM システム	63
サポート対象のビルド・システム	63
サポート対象の Force.com バージョン	64

第1章: ALI の紹介

本章では, ALI を概観します。

本章の内容

ALI の概要	8
ALI のワークフロー	9

ALI の概要

Application Lifecycle Intelligence (ALI) は、開発アクティビティを追跡して、Application Lifecycle Management (ALM) のエンティティに関連付けます。ALI は、ソース・コード管理 (SCM) ツールおよびビルド管理ツールを Application Lifecycle Management (ALM) と統合し、SCM ツールとビルド管理ツールのアクティビティ (コード変更, ビルド, ユニット・テストの結果, コード・カバレッジ・アナリシスなど) を ALM のエンティティ (リリース, 要件, 不具合, テストなど) にリンクします。

また、ALI を使用して、SCM ポリシー・コンプライアンスを作成し適用することもできます。たとえば、ある機能が特定のリリースに対して予定されているときにのみチェックインを許可したり、安定期には変更によって重大な不具合が修正される場合にのみチェックインを許可することができます。チェックインを許可する前にメタデータを要求したり、リリースのためにコードベースをロックしたりすることができます。

注 : 本書では、Application Lifecycle Intelligence を ALM と組み合わせて使用する方法について説明します。ALM の使用方法の詳細については、『HP Application Lifecycle Management ユーザーズ・ガイド』を参照してください。

使用例 :

- 要件または不具合にリンクされているコード変更や、初めてビルドに含まれたコード変更のレビュー
- ある期間に、リリースやビルドに対して実装された内容や、ある担当者によって実装された内容のチェック
- コード変更の行ごとの差異チェック
- 要件または不具合に関連付けられていない変更の特定
- ビルドと品質のメトリックのレビュー
- あるビルドが基になっているテストの確認

ALI のワークフロー

本項では、ALI ワークフローの全体について説明します。

1. ALI を設定します。
 - a. ALM で ALI 拡張機能を有効にします。

詳細については、[「ALI 拡張機能の有効化」\(12ページ\)](#)を参照してください。
 - b. 旧バージョンの ALI を使用していた場合は、[「旧バージョンからの移行」\(12ページ\)](#)の手順に従います。
 - c. SCM システムおよびビルド・システムとの同期を構成します。

詳細については、[「同期スケジュールの設定」\(13ページ\)](#)を参照してください。
 - d. ビルド・システムとの接続をセットアップし、設定します。

詳細については、[「ビルド・システムとの統合のセットアップ」\(16ページ\)](#)を参照してください。
 - e. SCM システムとの接続をセットアップし、設定します。

詳細については、[「ソース管理システムとの統合のセットアップ」\(28ページ\)](#)を参照してください。

f. Force.com プラットフォーム用に開発する場合は、Force.com 統合をセットアップします。

詳細については、[「Force.com との統合のセットアップ」 \(46ページ\)](#)を参照してください。

2. ソース・コードの変更を追跡します。

詳細については、[「SCM の変更とトレーサビリティの管理」 \(51ページ\)](#)を参照してください。

3. 開発アクティビティがリリース、要件、不具合に与える影響を追跡します。

詳細については、[「開発アクティビティのモニタ」 \(60ページ\)](#)を参照してください。

4. ビルドの品質を追跡します。

詳細については、[「ビルド・アクティビティのモニタ」 \(56ページ\)](#)を参照してください。

第2章: 始める前に

本章では、ALI のセットアップと設定を開始する前に完了しておくことが必要な作業について詳細に説明します。

本章の内容

ALI 拡張機能の有効化	12
旧バージョンからの移行	12
同期スケジュールの設定	13
TFS 統合サポートの設定	14

ALI 拡張機能の有効化

ALM のサイト管理で、ALI を使用するすべてのプロジェクトに対して ALI 拡張機能を有効にします。

ALM プロジェクトで拡張機能を有効にする方法の詳細については、『HP Application Lifecycle Management 管理者ガイド』を参照してください。

旧バージョンからの移行

本項の内容

- [「ALI 2.0 以前からの移行」](#)
- [「ALI 1.1 からの Perforce データの移行」](#)
- [「作業項目のアップグレード」](#)

使用中の ALI バージョンを確認するには、ALM のメイン・メニューから [ヘルプ] > [バージョン情報] を選択します。

ALI 2.0 以前からの移行

1. コード変更モジュールで、[メッセージ] カラムを追加します。2.0 より前のバージョンから ALI をアップグレードすると、自分で追加しない限り、このカラムは表示されなくなります。
2. ソース・コード管理に Perforce を使用する場合は、[「ALI 1.1 からの Perforce データの移行」](#) (12 ページ) の手順に従います。

ALI 1.1 からの Perforce データの移行

Application Lifecycle Intelligence バージョン 1.1 からアップグレードする場合は、で扱う Perforce データを次の手順で削除し再ロードします。

1. 管理モジュールで、[SCM リポジトリ] を選択してから Perforce リポジトリを選択します。
2. [リポジトリ] メニューから [クリーンアップ] を選択し、最新のコミットを削除するデータを選択します。
3. [ブランチ] タブを開き、ブランチごとに次の手順を実行します。
 - a. ブランチの詳細を選択します。

b. **【最終変更読み取り】** の値を削除します。

4. **【変更の検出】** タブで、同期を開始します。

作業項目のアップグレード

ALI に用意されているスクリプトを実行して、SYSTEM_FIELD プロジェクト・テーブルをアップグレードする必要があります。このスクリプトは、ALI バンドル内の **resources\work-items-update-scripts** フォルダにあります。データベース・コンソールを使用して、スクリプトを実行してください。

ALI 拡張機能が有効なプロジェクトごとに、**check-existence-of-has-changeset-linkage-fields.sql** を実行して、**has-changeset-linkage** フィールドが存在するかどうかを確認します。スクリプトの結果が 0 の場合は、**add-has-changeset-linkage-fields.sql** を実行してフィールドを追加してください。

同期スケジュールの設定

ALI は、事前設定された SCM リポジトリとビルド・サーバで行われた変更を検出し、コード変更と関連ファイル・ビルドの情報およびビルド・アーティファクトを ALM サーバにロードし、ロードしたコード変更、ビルド、作業項目（要件、不具合）間のトレーサビリティを自動生成します。

ALI スケジューラは、ALI が有効なプロジェクトの検出を一定間隔で実行し、各プロジェクトを同期します。プロジェクトの同期が実行されたときから、事前設定された同期間隔が経過していない場合、同期は実行されません。

ALI スケジューラの動作を設定するには、サイト管理にログインし **【サイト設定】** タブをクリックします。次の ALM サイト管理パラメータを追加または設定してください。パラメータが存在しない場合は、標準設定値が使用されます。

注意: これらの値を変更すると、サーバ・パフォーマンスに影響することがあります。

- **ALI_MAX_PROJECT_SYNC_JOB** : 同時に実行可能な同期ジョブの最大数。標準設定値は 5 です。
- **ALI_PROJECT_SYNC_DISCOVERY_INTERVAL** : プロジェクトの同期が必要かどうかを判断するチェック間隔。標準設定値は 10 分です。
- **ALI_MIN_PROJECT_SYNC_PERIOD** : 1つのプロジェクトの同期を行う最小時間間隔。実際の実行タイミングは、同期するプロジェクトと同期ジョブの数によります。標準設定値は 60 分です。

例 :

1. 11 時 30 分に、プロジェクト A は同期されました。
2. 11 時 45 分に、プロジェクト A に変更が加えられました。

3. 12 時 00 分に、スケジューラは検出チェックを実行します。結果：プロジェクト A について、同期は実行されません。

4. フリーの同期ジョブが利用可能であれば、プロジェクト A は 12 時 30 分に同期されます。

ALM 設定パラメータの設定の詳細については、『HP Application Lifecycle Management 管理者ガイド』の「ALM 設定パラメータの指定」の項を参照してください。

TFS 統合サポートの設定

Microsoft Team Foundation Server (TFS) システムを完全にサポートするため、HP ALI TFS Services を TFS サーバにインストールする必要があります。

このサービスをインストールするには、HP ALI TFS Services インストーラを TFS サーバ上で実行します。インストーラは、ALI バンドルに含まれています。インストーラによって、環境のセットアップと必要なすべての検証が行われます。HP ALI TFS インストーラは、ALI がサポートするすべての TFS バージョンで有効です。

サポート対象の TFS バージョンの詳細については、「[サポート対象の SCM システム](#)」(63 ページ)および「[サポート対象のビルド・システム](#)」(63 ページ)を参照してください。

本項の内容

- [「HP ALI TFS Services」](#)
- [「旧バージョンの ALI からのアップグレード」](#)
- [「SvnBridge 経由の TFS からネイティブの TFS 統合への移動」](#)

HP ALI TFS Services

HP ALI TFS Services には、次のサービスが含まれています。

- **行カウント・サービス**：コミット内の変更行を数えます。
- **ビルド・サービス**：ビルド情報を提示します。
- **ALI エージェント・サービス**：ALI エージェントが使用するサービスで、コミット・メッセージを検証し、コード変更を ALM にプッシュします。

旧バージョンの ALI からのアップグレード

ネイティブの TFS 統合を HP ALM 11.52 パッチ 01 以前で使用していた場合は、旧バージョンの HP ALI TFS Services をアンインストール/削除することを強くお勧めします。SvnBridge 経由の TFS を使用し

ていた場合、この処理は必要ありません。

旧バージョンの HP ALI TFS Services を削除するには、次の手順を実行します。

1. BisSubscribe /unsubscribe を使用して、前の CheckinEventService をアンインストールします。
2. 旧バージョンの ALI TFS Services に関連する IIS アプリケーション、IIS Web サイト、IIS アプリケーション・プールがあれば、IIS からすべて削除します。
3. TFS サーバの C:\inetpub にある **AliTfsServicesSite** フォルダを削除します。このフォルダは、前の ALI TFS Services IIS アプリケーションにマップされます。

SvnBridge 経由の TFS からネイティブの TFS 統合への移動

SvnBridge 経由の TFS はサポートされなくなりました。組み込みの TFS 統合を使用するには、次の手順を実行します。

1. ALM に構成されている以前の SCM リポジトリで、**SvnBridge 経由の TFS** タイプのものをすべて削除します。
2. プロジェクト・コレクションごとに、**TFS** タイプの新しい SCM リポジトリを追加します。
3. ALI に構成していた各プロジェクトの SCM ブランチを追加します。

第3章: ビルド・システムとの統合のセットアップ

本章の内容

ビルド・システムのセットアップ・ワークフロー	17
Hudson/Jenkins との統合	18
HP ALI Hudson/Jenkins プラグインのデプロイメント	18
HP ALI Hudson/Jenkins プラグインの設定	20
TFS ビルド・サーバとの統合	21
ビルド・サーバの追加	21
ビルド構成の追加	22
ビルド構成にある SCM 構成の再利用	23
ビルド構成の不具合フィルタの設定	24
ビルド変更の検出の設定	24
ALI プロジェクト・リストのカスタマイズ	25
コード問題の追跡	26

ビルド・システムのセットアップ・ワークフロー

ビルドとは、ソフトウェア開発の主要な成果物です。ALM は、ビルドに関する情報を追跡するとともに、ビルドとその他の ALI エンティティとの関係も追跡します。

ビルド・システムとの統合により、コード変更がソフトウェア成果物に与える影響を測定できます。表示するレポートには、どのようなコードが新しく実装され、プロジェクトにどのような影響を与えるのかを、主要なメトリックの差分で示すことができます。たとえば、あるコミットによって、テスト結果が 5% 低下したことが示されます。

ビルド・システムとの統合をセットアップするには、次の手順を実行します。

1. 使用するビルド・システムがサポートされていることを確認します。詳細については、[「サポート対象のビルド・システム」\(63ページ\)](#)を参照してください。
2. ビルド・サーバの要件を確認し、必要であれば構成します。

詳細については、次の項目を参照してください。

- [「Hudson/Jenkins との統合」\(18ページ\)](#)
- [「TFS ビルド・サーバとの統合」\(21ページ\)](#)

3. ビルド・サーバを ALI に追加します。詳細については、[「ビルド・サーバの追加」\(21ページ\)](#)を参照してください。
4. ビルド構成を追加します。詳細については、[「ビルド構成の追加」\(22ページ\)](#)を参照してください。

ビルド・サーバに定義されたビルド構成の既存の SCM 構成を再利用することもできます。詳細については、[「ビルド構成にある SCM 構成の再利用」\(23ページ\)](#)を参照してください。

5. 不具合フィルタを設定して、ビルド・レポート不具合トレンド・グラフに表示される不具合を制限します。詳細については、[「ビルド構成の不具合フィルタの設定」\(24ページ\)](#)を参照してください。
6. ビルド変更の検出を設定します。詳細については、[「ビルド変更の検出の設定」\(24ページ\)](#)を参照してください。
7. ALM プロジェクト・カスタマイズを使用して、ALI 固有のプロジェクト・リストをカスタマイズします。詳細については、[「ALI プロジェクト・リストのカスタマイズ」\(25ページ\)](#)を参照してください。

8. 不具合をコード問題としてタグ付けします。詳細については、「[コード問題の追跡](#)」(26ページ)を参照してください。

Hudson/Jenkins との統合

Hudson/Jenkins ビルド・サーバを使用する場合は、HP ALI Hudson/Jenkins プラグイン (ali-hudson-plugin.hpi) をインストールします。このプラグインは、ALI バンドル内の agents\build-integration\hudson\ にあります。

Perforce, TFS, Git :

Hudson/Jenkins プラグインをインストールします。また、適用可能な HP ALI Hudson/Jenkins プラグインが他にもある場合は、それもインストールします。

Perforce, TFS, Git SCM の使用に対応する Hudson/Jenkins プラグインは、<http://hudson-ci.org> または <http://jenkins-ci.org> からダウンロードできます。プラグインは、パブリックな Hudson/Jenkins プラグイン・リポジトリにあります。サポート状況は、次のとおりです。

- SVN は、標準設定でサポートされています。
- CVN は、Jenkins で標準設定でサポートされています。
- CVN は、Hudson バージョン 2.x で標準設定でサポートされています。

Hudson 3.x での CVS :

HP ALI Hudson プラグインの前提条件として udson CVS プラグインをインストールします。CVS プラグインは、Hudson バージョン 3.0 になるまでは標準設定でプリインストールされていましたが、削除されました。CVS プラグインは、<http://hudson-ci.org> からダウンロードできます。このプラグインは、パブリックな Hudson プラグイン・リポジトリにあります。

追加の HP ALI Hudson/Jenkins プラグインは、ALI バンドルにあります。使用する SCM に適用可能なものがあればインストールしてください。

Hudson/Jenkins 用の ALI プラグインをインストールする方法の詳細については、次の「[HP ALI Hudson/Jenkins プラグインのデプロイメント](#)」(18ページ)を参照してください。

HP ALI Hudson/Jenkins プラグインのデプロイメント

HP ALI Hudson/Jenkins プラグインをインストールするには、次の手順を実行します。

1. Hudson/Jenkins で [プラグインマネージャー] を開き、[高度な設定] タブをクリックします。

2. **【プラグインのアップロード】** セクションで、ALI バンドルにある `ali-hudson-plugin.hpi` または `ali-jenkins-plugin.hpi` を探して **【アップロード】** をクリックします。

- **TFS** : Hudson/Jenkins でビルドしたプロジェクト・ソース・コードが TFS に格納されている場合は、まず Hudson/Jenkins からダウンロードした Hudson/Jenkins TFS プラグインをインストールし、次に `ali-hudson-tfs-plugin.hpi/ali-jenkins-tfs-plugin` と、ベース・プラグインの `ali-hudson-plugin.hpi/ali-jenkins-plugin.hpi` をインストールします。

注: ソース管理用に ALI に TFS リポジトリを定義する場合、リポジトリ場所の書式には、プロジェクト・コレクションの名前が含まれます。詳細については、[「TFS リポジトリの追加」\(34ページ\)](#)を参照してください。

Hudson/Jenkins の使用には、プロジェクト・コレクションの名前は必要ありませんが、ALI と正しく連携するには、その名前が Hudson/Jenkins サーバ上に定義されている必要があります。Hudson/Jenkins サーバで、TFS サーバへのパスにプロジェクト・コレクション名が含まれていることを、**【Source Code Management (ソースコード管理)】** > **【サーバ URL】** で確認してください。

- **Perforce** : Hudson/Jenkins でビルドしたプロジェクト・ソース・コードが Perforce に格納されている場合は、まず Hudson/Jenkins からダウンロードした Hudson/Jenkins Perforce プラグインをインストールし、次に `ali-hudson-perforce-plugin.hpi/ali-jenkins-perforce-plugin.hpi` と、ベースの `ali-hudson-plugin.hpi/ali-jenkins-plugin.hpi` をインストールします。
- **Git** : Hudson/Jenkins でビルドしたプロジェクト・ソース・コードが Git に格納されている場合は、まず Hudson/Jenkins からダウンロードした Hudson/Jenkins Git プラグインをインストールし、次に `ali-hudson-git-plugin.hpi/ali-jenkins-git-plugin.hpi` と、ベースの `ali-hudson-plugin.hpi/ali-jenkins-plugin.hpi` をインストールします。

注: ALI Git プラグインでは、ワイルドカードを含むブランチ名はサポートされていません。Hudson/Jenkins サーバ上で、ワイルドカードが **【Source Code Management (ソースコード管理)】** > **【Branches to build (作成するブランチ)】** に定義されていないことを確認してください。

3. プラグインのアップロードが完了したら、Hudson/Jenkins サーバを再起動して変更内容を適用します。
4. インストールしたプラグインが **【プラグインマネージャ】** の **【インストール済み】** タブに表示されることを確認してください
5. Hudson/Jenkins の左側のメニューに **【ALI Integration】** リンクが表示されていることを確認してください。このリンクをクリックすると、プラグインの機能が表示されます。

Hudson/Jenkins プラグインのインストールと操作の詳細については、Hudson/Jenkins システムのドキュメントを参照してください。

HP ALI Hudson/Jenkins プラグインの設定

ALI Hudson プラグインのグローバルな設定は、グローバルな [システムの設定] からアクセスします。ジョブ固有の設定は、ジョブからアクセスします。

グローバル設定で指定したプロパティはすべて、ジョブごとに上書きできます。

プロパティの詳細な説明については、Hudson/Jenkins サーバ上の ALI 統合プラグインを参照してください。

グローバル設定を行うには、次の手順を実行します。

[**Hudson/Jenkins の管理 - システムの設定**] で [**ALI Integration**] を選択します。次のオプションを指定します。

- **Include the credentials in the SCM configuration** - SCM リポジトリ記述子にユーザ名とパスワードを含めるかどうかを指定します。このセキュリティ・モデルを有効にすると、ユーザは資格情報のリストのために「Extended Read」権限が必要になります。

注意: このオプションが有効な場合は、ビルド構成に関連する SCM リポジトリの資格情報が REST エンドポイントでプレーン・テキストとして公開されます。

- **Update build information in HP ALM** - このオプションが有効な場合は、ビルドの開始時と終了時に、ALM サーバに通知されます。

プッシュ機構を使用する場合に、個々のジョブの ALM プロパティも指定するときは、このオプションを有効にする必要があります。

- **Hudson/Jenkins URL** - Hudson または Jenkins サーバの URL を指定します。

注: ビルド・システムの詳細リンクに問題がある場合は、Hudson/Jenkins サーバの URL が正しく入力されていることを確認してください。

ジョブごとの設定を行うには、次の手順を実行します。

ジョブの左側メニューにある構成リンクで、[**ALI Integration**] オプションを選択して、ジョブの ALI 統合を有効にします。

- **テストソースのマッピングパターン** - 実際のテスト結果に基づいて、テスト・ソースの場所を設定します。詳細については、ALI 統合プラグインのヘルプを参照してください。

- **HP ALM 設定** – グローバルな ALI 設定プロパティを上書きします。ALM にあるビルド・ジョブのビルド情報を更新します。
- **NCover** - .NET 設定用の NCover コード・カバレッジです。[NCover レポート XML] では、未処理の XML レポート・ファイル (myproject/target/coverage-reports/*.xml など) の生成を指定します。ファイルセットのベースディレクトリは、ルート・ワークスペースです。
- **Force.com** – 詳細については、「[Force.com との統合のセットアップ](#)」(46ページ)を参照してください。

TFS ビルド・サーバとの統合

注: TFS をビルド・サーバとして完全にサポートするには、HP ALI TFS Services を TFS 2012 サーバにインストールする必要があります。設定と制限の詳細については、「[TFS 統合サポートの設定](#)」(14ページ)を参照してください。

TFS ビルドの機能と制限事項

TFS ビルド・サーバ・サポートは、定義済みのすべてのビルド定義、そのビルド、コード変更、テスト結果、ビルドごとのコード・カバレッジに関するデータを公開します。また、ビルド定義ごとのすべての定義済み SCM リポジトリに関する情報も公開します。ビルド統計は、テストとコード・カバレッジの結果を除き、制限なしで公開されます。

ビルド定義のために ALI で必要なテスト結果とコード・カバレッジ結果にアクセスするには、定義が次の条件を満たす必要があります。

- ビルド定義およびビルドに「格納フォルダ」が定義されていることが必要です。ビルドには、ログ・フォルダも必要です。
- ビルドの初めに、ワークスペースのクリーニングが必要です。コード変更は、すべてチェックアウトされている必要があります。
- テストとコード・カバレッジについて使用されたフレームワークの結果が、添付ファイルとしてそれぞれ *.trx ファイルと *.coverage ファイルに保存されている必要があります。
- TFS ビルド・サーバ構成は、プッシュ機構をサポートしません。ビルドは、すべてポーリング機構を使用して、HP ALM に保存されます。

ビルド・サーバの追加

ビルド・サーバを追加するには、次の手順を実行します。

1. 管理者権限で ALM にログインし、[管理] タブにある [ビルド サーバ] をクリックして、[サーバ] ページを開きます。
2. [新規サーバ] をクリックし、ドロップダウン・リストからビルド・サーバ・タイプを選択して [OK] をクリックします。
3. [新規ビルドサーバ] ダイアログ・ボックスで、詳細を入力します。

注:

- サーバの場所には、完全パス (http://xx.xx.xxx.xxx:xxxx/yyyy など) を入力します。この場所は、システム管理者に問い合わせてください。/yyyy の部分はオプションです。これは、サーバが実行されるコンテキスト (Jenkins サーバ上など) を指定するためです。
- ビルド・サーバの場所には、ダウンロードして TFS サーバにインストールする HP ALI TFS Services に含まれているサービスの URL を入力します。サービス・インストーラは、ALI バンドルに含まれています。TFS ビルド・サーバの場所としては、プロジェクト・コレクションごとに、次の書式の URL を使用します。

http://tfsServer:tfsPort/tfsali/Service/BuildService.svc/ali/ProjectCollectionName

説明 :

- tfsServer – TFS サーバのアドレス
- tfsPort – TFS サーバのポート
- tfsali – HP ALI TFS Services のインストール時に定義された、IIS サイト上のアプリケーションの名前
- ProjectCollectionName – プロジェクト・コレクションの名前

4. [変更の検出] タブを選択します。さらに、次の内容をオプションで選択します。
 - 必要に応じて、[ビルドサーバから変更を読み取る] を選択します。変更が、SCM リポジトリから定期的に取り込まれるようになります。
 - 必要に応じて、[ビルドサーバエージェントから送信されたビルドを受信する] を選択します。詳細については、「[ビルド変更の検出の設定](#)」(24ページ)を参照してください。

ビルド構成の追加

ビルド構成を追加するには、次の手順を実行します。

1. [ビルド構成] タブで, [新規ビルド構成] をクリックします。
2. リストからビルド構成を選択し, [OK] をクリックして, 詳細ウィンドウを開きます。
リストには, 管理対象サーバの未使用構成が表示されます。
リストに表示されるのは, ALI との統合が有効な Hudson ジョブまたは Jenkins ジョブで使用する構成のみです。
3. [リリース] ドロップダウン・リストをクリックし, 関連のリリースを選択します。
4. [ビルド カテゴリ] ドロップダウン・リストをクリックし, ビルド・タイプを選択します。この値に基づいて, [開発連携] タブのビルド・モジュールでビルドをフィルタ処理できます。
5. オプション:
 - 構成を有効にします。ビルド構成を有効にすると, ビルド・システムから新しいビルドがロードされます。
 - 構成を標準設定として設定します。標準設定のビルド構成に基づいて, 関連リリースに対して表示される統計データが計算されます。

ビルド構成にある SCM 構成の再利用

ビルド・サーバに保存されている既存のビルド構成とその SCM リポジトリ構成の設定は, HP ALM の新しいビルド構成に適用できます。

既存の SCM ビルド構成を適用するには, 次の手順を実行します。

1. 管理モジュールの [ビルド サーバ] ページで, サーバを選択します。
2. [ビルド構成] タブを選択します。
3. 構成を選択して, クリックします。
4. [ビルド構成詳細] ウィンドウで [SCM] タブを選択して, 使用可能なリポジトリとブランチを表示します。
5. リポジトリとブランチを選択して [+] ボタンをクリックすると, 選択した内容が SCM 構成に適用されます。[詳細] ボタンをクリックすると, ポップアップ・ウィンドウが開いて選択したリポジトリまたはブランチの詳細情報が表示されます。

リポジトリまたはブランチの場所の横のステータス・メッセージは, そのリポジトリまたはブランチが定義済みかどうかを示します。

ビルド構成の不具合フィルタの設定

不具合フィルタを設定すると、ビルド構成のビルド・レポート不具合トレンド・グラフにおいて、リリースに関連付けられた未解決の不具合と新規不具合の入力セットのサイズを絞り込むことができます。このフィルタでは、不具合のステータスとターゲット・リリースは指定しないでください。

ビルド構成に不具合フィルタを適用するには、次の手順を実行します。

1. 管理モジュールの [ビルド サーバ] ページで、サーバを選択します。
2. [ビルド構成] タブを選択します。
3. 構成名をクリックして、[ビルド構成詳細] ウィンドウを開きます。
4. [不具合フィルタ] を選択します。
5. [不具合フィルタの変更] をクリックすると、不具合フィルタのウィンドウが開きます。
6. フィルタ条件を追加または削除し、[OK] をクリックすると条件が適用されます。

ビルド構成から不具合フィルタを削除するには、次の手順を実行します。

[不具合フィルタのクリア] をクリックし、[はい] をクリックして確定します。

ビルド変更の検出の設定

ALI がビルド・サーバに新しく追加されたビルドを検出すると、新しいステータス情報、関連のコード変更、さらにオプションでユニット・テストの結果とコード・カバレッジがロードされます。ビルド、コード変更、作業項目（不具合、要件）間で、トレーサビリティが自動生成されます。ALI は、SCM との統合と同様に、ポーリングとプッシュという2つの検出機構をサポートしています。

新しいビルド・サーバを定義する場合は、変更の検出の設定値を [変更の検出] ビューで設定します。既存のサーバ定義については、[変更の検出] タブを使用します。

ポーリング：

ポーリングとは、ビルド・サーバに新しいビルドがあるかどうかを ALI が定期的にチェックし、新しいビルドを ALM サーバにロードする処理のことです。

ポーリングを有効にするには、[変更の検出] タブにある [ビルドサーバから変更を読み取る] オプションを選択します。必要時に変更をロードするには、[同期] コマンドを使用します。変更の自動ロードのスケジュールを設定するには、[「同期スケジュールの設定」\(13ページ\)](#)を参照してください。

プッシュ：

プラグインによるビルド・サーバのリスニングを開始するには、ビルド・サーバの ALI 統合プラグインで [Update build information in HP ALM] オプションを選択します。詳細については、「[HP ALI Hudson/Jenkins プラグインの設定](#)」(20ページ)を参照してください。新しいビルドが開始すると、そのビルドに関する情報が、プラグインによって ALM サーバにプッシュされます。ビルドが完了すると、ALM サーバにあるステータスなどのデータが更新されます。

プッシュ機構を有効にするには、「変更の検出」タブにある「ビルドサーバエージェントが送信したビルドオプションを受信」オプションを選択します。

ALI プロジェクト・リストのカスタマイズ

[ALM プロジェクトカスタマイズ] の [プロジェクトリスト] ページで ALI でデフォルトで提供されているプロジェクト・リストをカスタマイズできます。

ALM プロジェクト・リスト	説明
ALI ビルド カテゴリ	[ビルド カテゴリ] フィールドで指定できる値。
ALI 終了ステータス	不具合の [ステータス] フィールドにこのリストで定義されている値のいずれかが設定されると、ALI はこの不具合を解決済みとみなします。[ビルドで解決済み] の値を設定できるのは、不具合のステータスが [終了ステータス] リストの要素である場合のみです。[終了ステータス] の値は、[ステータス] プロジェクト・リストのサブセットです。
ALI コード問題定義	不具合をコード問題として識別する値。 コード問題の詳細については、「 コード問題の追跡 」(26ページ)を参照してください。
ALI QA ステータス	ビルドの QA ステータスとして有効な値。
ALI レポート 済み重大度	コード問題へのリンクで指定可能な値。この値は、要件モジュールまたは不具合モジュールの [開発アクティビティ] タブと [ビルド詳細] タブに表示されません。優先度のプロジェクト・リストと同じ値、またはサブセットを定義してください。

プロジェクト・リストのカスタマイズの詳細については、『HP Application Lifecycle Management 管理者ガイド』を参照してください。

コード問題の追跡

ソース・コードの問題を特定したら、問題があるコードに関連付けられる不具合、要件、ビルドで問題を追跡できます。

問題があるコードを修正するためにユーザが新しい不具合を作成すると、ALI は、設定された条件に基づいて、その不具合に**コード問題**のタグを付けることができます。プロジェクトでコード問題の追跡を設定すると、[要件]、[不具合]、[ビルド詳細] の各ダイアログ・ボックスから、コード問題の不具合へのリンクを表示できます。

ALM プロジェクトにコード問題の追跡を設定するには、次の手順を実行します。

1. **前提条件**：標準設定の ALI プロジェクト・リストをカスタマイズします。詳細については、「[ALI プロジェクト・リストのカスタマイズ](#)」(25ページ)を参照してください。
2. [プロジェクトのカスタマイズ] の [プロジェクトのエンティティ] ページで、コード問題の追跡用にユーザ定義のフィールドを作成します。サポートされるフィールド・タイプは、**文字列**または**ルックアップ・リスト**です。

ユーザ定義のフィールドを作成する詳細な手順については、『HP Application Lifecycle Management ユーザーズ・ガイド』を参照してください。

3. [プロジェクト リスト] ページの [**ALI コード問題定義**] プロジェクト・リストで、次の手順を実行します。
 - a. ユーザ定義フィールドの [**名前**] フィールドの値を、唯一のリスト項目として追加します。
 - b. ステップ 2 で作成したユーザ定義フィールドから、不具合にコード問題のタグを付ける際に使用する値を選択します。その値をサブ項目として追加します。

例：

ユーザ定義フィールドとして、たとえば**不具合カテゴリ**を作成します。このフィールドの値は、[バグ]、[問題]、[拡張機能]、[強化] などになります。

[**ALI コード問題定義**] プロジェクト・リストで、ユーザ定義フィールドの [**名前**] フィールドの値を唯一のリスト項目として追加します。たとえば、BG_USER_05 を追加します。

不具合にコード問題のタグを付ける際に使用するユーザ定義フィールドの値を、リストのサブ項目として追加します。たとえば、「強化」を追加します。

コード問題を追跡するには、次の手順を実行します。

1. **前提条件** : 次の条件を満たす不具合が、要件、不具合、ビルドのコード問題と認識されます。
 - 不具合が解決済みでないこと。不具合のステータスが、**[ALI 終了ステータス]** プロジェクト・リストで定義されている値のいずれかである場合、不具合は解決済みです。
 - 不具合の優先度が、**[ALI レポート済み重大度]** プロジェクト・リストで定義されている値のいずれかであること。
2. 不具合にコード問題のタグを付けます。**[不具合の詳細]** ダイアログ・ボックスで、上記でコード問題の追跡用に作成したユーザ定義フィールドに入力します。
3. 不具合を、問題のあるコード変更に関連付けられている不具合、要件、またはビルドに接続します。
 - a. **[不具合の詳細]** ダイアログ・ボックスで、ALI の **[起因するコード変更]** フィールドから適切な値を選択します（このフィールドには、コード変更とコミット・メッセージがリストされます）。
 - b. **[不具合の詳細]** ダイアログ・ボックスで、**[検出ビルド]** フィールドの値を選択します。

例 :

1. 特定の不具合、要件、またはビルドに関連するコード変更をコミットします。たとえば、不具合 124 を修正する際に行ったコード変更をコミットします。
2. レビューア、またはコード解析に使用するサードパーティ・ツールが、コード変更によって問題が生じることを認識します。新しい不具合を作成して、次の2つのフィールドを必ず設定します。
 - a. コード問題のユーザ定義フィールド（この例では**不具合カテゴリ**）で、適切な値を選択します。この例では、**[強化]** を選択します。
 - b. **[起因するコード変更]** フィールドで、この問題の原因となったコード変更を探し、選択します。たとえば、**[不具合 #124 : 認証の問題を修正]** を選択します。

結果 :

新しいコード問題の不具合が、関連する**[ビルドの詳細]**、**[要件の詳細]**、または**[不具合の詳細]** ダイアログ・ボックスにリストされます。

第4章: ソース管理システムとの統合のセットアップ

本章の内容

SCM のセットアップのワークフロー	29
SCM との統合の前提条件と制限事項	30
SCM エージェント	30
SCM リポジトリの追加	31
Subversion リポジトリの追加	32
CVS リポジトリの追加	32
Perforce リポジトリの追加	33
TFS リポジトリの追加	34
Git リポジトリの追加	35
外部リポジトリ・ビューアの設定	36
コミット・パターンの設定	38
コード変更と代替 ID との関連付け	41
ブランチの追加とリリースへの関連付け	42
ブランチのチェックイン・ポリシーの設定	43
コード変更の検出の設定	45

SCM のセットアップのワークフロー

ALI でコード変更についてレポートし、コード変更を ALM エンティティにリンクできるようにするには、ソース管理システム (SCM) との統合をセットアップしておく必要があります。

SCM との統合をセットアップするには、次の手順を実行します。

1. 使用する SCM システムがサポートされていることを確認します。詳細については、[「サポート対象の SCM システム」 \(63ページ\)](#)を参照してください。

2. 前提条件を確認します。

詳細については、[「SCM との統合の前提条件と制限事項」 \(30ページ\)](#)を参照してください。

3. リポジトリを追加します。

詳細については、[「SCM リポジトリの追加」 \(31ページ\)](#)を参照してください。

4. (オプション) SCM のエージェントをインストールします。

詳細については、[「SCM エージェント」 \(30ページ\)](#)を参照してください。

5. (オプション) 外部リポジトリ・ビューアを設定します。

詳細については、[「外部リポジトリ・ビューアの設定」 \(36ページ\)](#)を参照してください。

6. (オプション) チェックイン・メッセージに対して特定の書式を要求します。詳細については、[「コミット・パターンの設定」 \(38ページ\)](#)を参照してください。

7. ブランチを追加し、リリースに関連付けます。

詳細については、[「ブランチの追加とリリースへの関連付け」 \(42ページ\)](#)を参照してください。

8. (オプション) チェックイン・ポリシーとブランチのロックを設定して、コミットを SCM ブランチのレベルで制御します。詳細については、[「ブランチのチェックイン・ポリシーの設定」 \(43ページ\)](#)を参照してください。

9. ソース変更の検出を設定します。

詳細については、[「コード変更の検出の設定」 \(45ページ\)](#)を参照してください。

SCM との統合の前提条件と制限事項

Application Lifecycle Intelligence (ALI) は、Subversion, Perforce, GIT, CVS の各システムをネイティブでサポートします。ただし、チェックイン・ポリシーまたはプッシュ機構が必要な場合は、SCM システムに SCM エージェントをインストールする必要があります。詳細については、[「SCM エージェント」\(30ページ\)](#)を参照してください。

TFS システムはネイティブでサポートされます。プッシュ機構をサポートするには、TFS エージェント Web サービスをインストールします。また、TFS エージェント Web サービスがない場合、すべてのコード変更は変更行フィールドに “0” と表示されます。

前提条件：

MS PowerShell 2.0 以降を Windows 環境にインストールし、スクリプトの実行を有効に設定します。Linux/Unix エージェント・スクリプトは BASH を使用します。複数のリポジトリのプッシュをサポートするために、BASH のバージョンは 4.0 以降を使用してください。

制限事項：

Git : ALI は、コミットの履歴を変更する Git の機能をサポートしません。Git リポジトリ内でコミットの履歴が変更された場合、影響が発生したすべてのブランチについて、ALI が追跡しているコミット履歴全体を再ロードする必要があります。コミット履歴を再ロードするには、ブランチを削除してから追加します。

SCM エージェント

HP ALI エージェントは、SCM サーバにインストールされるスクリプトまたは専用のアプリケーションのセットです。ALI エージェントは、SCM サーバ上のアクティビティをリスンし、アクティビティを ALI にレポートするとともに、ALI との統合をサポートします。たとえば、設定済みのリポジトリとブランチに変更がコミットされるときに、エージェントは、コミットが許可されるかどうかをポリシーを基準にチェックし、許可される場合はコード変更を ALM サーバにプッシュします。

SCM エージェントの設定の詳細については、`readme.txt` ファイルを参照してください。このファイルは、メインの ALI 配布アーカイブ内のエージェント・アーカイブに収録されています。

SCM エージェントの操作方法の詳細については、次の項を参照してください。

- [「コード変更の検出の設定」\(45ページ\)](#)
- [「ブランチのチェックイン・ポリシーの設定」\(43ページ\)](#)

Linux 用 SVN エージェント

`agents\scm-integration\unix-linux\scm-agent-subversion.tgz`

Windows 用 SVN エージェント

agents\scm-integration\windows\scm-agent-subversion.zip

Linux 用 CVS エージェント

agents\scm-integration\unix-linux\scm-agent-cvs.tgz

Windows 用 CVS エージェント

agents\scm-integration\windows\scm-agent-cvs.zip

Windows 用 TFS Services

agents\scm-integration\windows\ali-tfs-services.zip

Linux 用 Perforce エージェント

agents\scm-integration\unix-linux\scm-agent-perforce.tgz

Windows 用 Perforce エージェント

agents\scm-integration\windows\scm-agent-perforce.zip

Linux 用 Git エージェント

agents\scm-integration\unix-linux\scm-agent-git.tgz

Windows 用 Git エージェント

agents\scm-integration\windows\scm-agent-git.zip

SCM リポジトリの追加

SCM システムからコード変更をロードし、次の内容の自動追跡を有効にするように、SCM リポジトリを設定します。

- 作業項目 (要件/不具合)
- コード変更
- 定義済みのリリース

本項の内容

Subversion リポジトリの追加	32
CVS リポジトリの追加	32
Perforce リポジトリの追加	33
TFS リポジトリの追加	34
Git リポジトリの追加	35

Subversion リポジトリの追加

サポートされるリポジトリのリストについては、「[サポート対象の SCM システム](#)」(63ページ)を参照してください。

リポジトリを追加するには、次の手順を実行します。

1. 管理者権限で ALM にログインします。ALM サイドバーで、[管理] にある [SCM リポジトリ] を選択すると [リポジトリ] ページが開きます。
2. [新規リポジトリ] をクリックします。ドロップダウン・リストから、SCM タイプを選択して [OK] をクリックします。
3. [新規 SCM リポジトリ] ウィンドウでフィールドに入力します。

場所は、リポジトリの完全パスです。たとえば、`http://host/svn/repo` のように指定します。この場所は、システム管理者に問い合わせてください。

リポジトリの場所は、実際のリポジトリのルートポイントをポイントします。リポジトリ内のパスの指定には、ブランチを使用します。SVN URL に不明なルートが含まれている場合は、`svn info <URL>` コマンドでルートを特定してください。

リポジトリを編集するには、次の手順を実行します。

1. [リポジトリ] ページの左の表示枠でリポジトリを選択します。
2. 右の表示枠でフィールドを編集します (別グループの設定に移動するには、タブを使用します)。
3. リポジトリ・プロパティの変更が必要な場合は、そのプロパティを選択して [プロパティの編集] ボタンをクリックします。[プロパティの編集] ウィンドウで値を追加し、[OK] をクリックします。プロパティは、詳細タブの下の領域にあります。

CVS リポジトリの追加

サポートされるリポジトリのリストについては、「[サポート対象の SCM システム](#)」(63ページ)を参照してください。

リポジトリを追加するには、次の手順を実行します。

1. 管理者権限で ALM にログインします。ALM サイドバーで、[管理] にある [SCM リポジトリ] を選択すると [リポジトリ] ページが開きます。

2. **【新規リポジトリ】** をクリックします。ドロップダウン・リストから、SCM タイプを選択して **【OK】** をクリックします。

3. **【新規 SCM リポジトリ】** ウィンドウでフィールドに入力します。

場所は、CVS サーバの完全なホスト名です。ホスト名は、システム管理者に問い合わせてください。

エイリアス・プロパティには、ビルド・システムで設定されているとおりに、完全な CVSROOT を入力します。たとえば、`pserver:username:password@host/cvsrepo` のように入力します。

CVSRoot プロパティは、pserver プロトコルでは必須です。

4. CVS リポジトリでは、CVS プロトコルの pserver、コード変更のロードを開始する開始日の指定が必要です。CVS ルートは必須ではありませんが、指定してください。

リポジトリを編集するには、次の手順を実行します。

1. **【リポジトリ】** ページの左の表示枠でリポジトリを選択します。
2. 右の表示枠でフィールドを編集します（別グループの設定に移動するには、タブを使用します）。
3. リポジトリ・プロパティの変更が必要な場合は、そのプロパティを選択して **【プロパティの編集】** ボタンをクリックします。**【プロパティの編集】** ウィンドウで値を追加し、**【OK】** をクリックします。プロパティは、詳細タブの下の領域にあります。

Perforce リポジトリの追加

サポートされるリポジトリのリストについては、[「サポート対象の SCM システム」\(63ページ\)](#)を参照してください。

リポジトリを追加するには、次の手順を実行します。

1. 管理者権限で ALM にログインします。ALM サイドバーで、**【管理】** にある **【SCM リポジトリ】** を選択すると **【リポジトリ】** ページが開きます。
2. **【新規リポジトリ】** をクリックします。ドロップダウン・リストから、SCM タイプを選択して **【OK】** をクリックします。
3. **【新規 SCM リポジトリ】** ウィンドウでフィールドに入力します。

ALI リポジトリは、Perforce デポに相当します。リポジトリの場所は、Perforce Server のホスト名とポート、およびデポ名です。書式は、`host:port//depot_name` です。

リポジトリを編集するには、次の手順を実行します。

1. [リポジトリ] ページの左の表示枠でリポジトリを選択します。
2. 右の表示枠でフィールドを編集します（別グループの設定に移動するには、タブを使用します）。
3. リポジトリ・プロパティの変更が必要な場合は、そのプロパティを選択して **[プロパティの編集]** ボタンをクリックします。 **[プロパティの編集]** ウィンドウで値を追加し、 **[OK]** をクリックします。プロパティは、詳細タブの下の領域にあります。

TFS リポジトリの追加

サポートされるリポジトリのリストについては、 [「サポート対象の SCM システム」 \(63ページ\)](#) を参照してください。

リポジトリを追加するには、次の手順を実行します。

1. 管理者権限で ALM にログインします。ALM サイドバーで、 **[管理]** にある **[SCM リポジトリ]** を選択すると **[リポジトリ]** ページが開きます。
2. **[新規リポジトリ]** をクリックします。ドロップダウン・リストから、SCM タイプを選択して **[OK]** をクリックします。
3. **[新規 SCM リポジトリ]** ウィンドウでフィールドに入力します。

ALI リポジトリは、TFS プロジェクト・コレクションに相当します。リポジトリの場所は、TFS サーバのホスト名とポート、プロジェクト・コレクションへのパスです。書式は、`http://host:port/tfs/path_to_project_collection` です。

行カウント・サービスの URL が標準設定値

(`http://tfsServer:tfsPort/tfsali/Service/LineCountService.svc`) と異なる場合は、行カウント・サービスの URL を入力する必要があります。これが設定されていない場合、ブランチは警告状態で作成され、ロードされるすべてのコード変更は変更行 0 で保存されます。行カウント・サービスは、ダウンロードして TFS Server にインストールする ALI TFS SCM エージェントに含まれます。HP ALI TFS Services は、ALI バンドルに含まれます。

注: . 以前のサービスである `CountLinesService.svc` は、ALI バージョン 2.0.0.139876 以降ではサポートされません。

リポジトリを編集するには、次の手順を実行します。

1. [リポジトリ] ページの左の表示枠でリポジトリを選択します。
2. 右の表示枠でフィールドを編集します（別グループの設定に移動するには、タブを使用します）。
3. リポジトリ・プロパティの変更が必要な場合は、そのプロパティを選択して **[プロパティの編集]** ボタンをクリックします。 **[プロパティの編集]** ウィンドウで値を追加し、 **[OK]** をクリックします。プロパティは、詳細タブの下の領域にあります。

Git リポジトリの追加

サポートされるリポジトリのリストについては、 [「サポート対象の SCM システム」 \(63ページ\)](#) を参照してください。

リポジトリを追加するには、次の手順を実行します。

1. 管理者権限で ALM にログインします。ALM サイドバーで、 **[管理]** にある **[SCM リポジトリ]** を選択すると **[リポジトリ]** ページが開きます。
2. **[新規リポジトリ]** をクリックします。ドロップダウン・リストから、SCM タイプを選択して **[OK]** をクリックします。
3. **[新規 SCM リポジトリ]** ウィンドウでフィールドに入力します。

リポジトリの場所は、Git URL の全体です。例： `https://github.com/hp/ali.git` などと指定します。URL にはユーザ名を含めないでください。

例： `ssh://git@server1.abc.com/home/git/gitrepo.git`。ユーザ名は、 **[ユーザ名]** フィールドに入力してください。

サポート対象のプロトコルには、 `http`、 `https`、 `Git`、 `ssh` があります。

秘密鍵を使用した `ssh` 認証には、 **[セキュリティ キー]** セクションを使用します。

注: `github.com` への接続に問題がある場合は、ALM でプロキシが設定されていることを確認します。

ALM プロキシを設定するには、次の行を `wrapper.conf` ファイルに追加します。このファイルは、ALM アプリケーションのデプロイメント・フォルダ (`C:\ProgramData\HP\ALM\`) にあります。

```
wrapper.java.additional.[nextFreeNumber]=-Dhttp.proxyHost=[proxyServerUrl] -  
Dhttp.proxyPort=[proxyPort] -Dhttps.proxyHost=[httpsProxyServerUrl] -Dhttps.proxyPort=  
[httpsProxyPort]
```

説明:

- [nextFreeNumber]: wrapper.conf パラメータの次の未使用番号
- [proxyServerUrl]: プロキシ・サーバの URL
- [proxyPort]: HTTP のポート番号
- [httpsProxyServerUrl]: HTTPS プロキシ・サーバの URL (オプションのパラメータ)
- [httpsProxyPort]: HTTPS プロキシのポート番号 (オプションのパラメータ)

4. GitHub リポジトリを設定している場合は, "GitHub view diff/file link templates" プロパティを選択して, ネイティブの GitHub ファイル表示および差異表示 Web ツールを使用します。

リポジトリを編集するには, 次の手順を実行します。

1. [リポジトリ] ページの左の表示枠でリポジトリを選択します。
2. 右の表示枠でフィールドを編集します (別グループの設定に移動するには, タブを使用します)。
3. リポジトリ・プロパティの変更が必要な場合は, そのプロパティを選択して [プロパティの編集] ボタンをクリックします。 [プロパティの編集] ウィンドウで値を追加し, [OK] をクリックします。プロパティは, 詳細タブの下の領域にあります。

外部リポジトリ・ビューアの設定

ALI には, ファイルの差異や詳細を表示するためのリポジトリ・ビューアが組み込まれていますが, ViewVC などの外部のリポジトリ・ビューアも使用できます。

外部ビューアを使用する場合は, SCM リポジトリの設定時に次に示すプロパティを入力します。このプロパティは, 既存のリポジトリの詳細タブから変更することもできます。

差異リンクのテンプレート-リポジトリ表示システム (ViewVC など) で, ファイルの差異ビューをポイントするリンクのテンプレート。このテンプレートでは, UI を使用して ALI コード変更テーブルからリンクを作成できます。コード変更内の各ファイルには, 差異ビュー (現在のコード変更のリビジョンと旧リビジョンの差異) を表示するリンクが含まれています。テンプレートにはタグ/変数を含めることが可能です。これは, 実行時に, その時点で UI で選択しているコード変更のコンテキストで展開されます。

置換または展開されるタグ/変数

`${filePath}` ... リポジトリ内のファイルのパス。

`${revision}` ... 現在のコード変更のリビジョン。

`${fromRevision}` ... 現在のコード変更のリビジョンの前のリビジョン。

`${fromFilePath}` ... 元の場所のコピー/移動。

注: 次のアドレスを使用すると、TFS に保存されている 2 つのファイルの差異を表示できます。

TFS

```
http://tfshost:tfSPORT/tfs/_COLLECTION_/_versionControl/changesets#opath=${fromFilePath}
&overion=${fromRevision}&mpath=${filePath}&mversion=${revision}&_a=compare
```

説明 :

- `tfshost` : TFS サーバのアドレス
- `tfSPORT` : TFS サーバが実行されているポート
- `_COLLECTION_` : 目的のファイルのプロジェクトがメンバーとして所属するコレクションの名前
- `opath` : オリジナルのソース・ファイルの完全パス (\$ 記号とプロジェクト名を含む)。
\$/ali-replica/alik/pom.xml など
- `ocs` : TFS のオリジナルのコード変更 ID
- `overion` : TFS のオリジナルのコード変更 ID
- `mpath` : 変更されたソース・ファイルの完全パス (\$ 記号とプロジェクト名を含む)。
\$/ali-replica/alik/pom.xml など
- `mcs` : TFS の変更後のコード変更 ID
- `mversion` : TFS の変更後のコード変更 ID

ファイル・リンクのテンプレート - リポジトリ表示システム (ViewVC など) で、ファイルのファイル・ビューをポイントするリンクのテンプレート。このテンプレートでは、UI を使用して ALI コード変更テーブルからリンクを作成できます。コード変更内の各ファイルには、ファイル・ビュー (最新のコード変更の範囲に含まれる、所定のリビジョンのファイル・テキスト・コンテンツ) を表示するリンクが含まれています。テンプレートにはタグ/変数を含めることが可能です。これは、実行時に、その時点で UI で選択しているコード変更のコンテキストで展開されます。

置換または展開されるタグ/変数

`${filePath}` ... リポジトリ内のファイルのパス。

`${revision}` ... 現在のコード変更のリビジョン。

注: 次のアドレスを使用すると、TFS に保存されているファイルのソースを表示できます。

TFS

```
http://tfshost:tfssport/tfs/_COLLECTION_/_versionControl/changesets#cs=${revision}
&path=${filePath}&version=${revision}&a=contents
```

説明 :

- tfshost : TFS サーバのアドレス
- tfssport : TFS サーバが実行されているポート
- _COLLECTION_ : 目的のファイルのプロジェクトがメンバーとして所属するコレクションの名前
- path : ソース・ファイルの完全パス (\$ 記号とプロジェクト名を含む)。
\$/alireplica/alik/pom.xml など
- cs : TFS のコード変更 ID
- version : TFS のコード変更 ID

コミット・パターンの設定

SCM の変更と ALM の要件および不具合とのトレーサビリティを検出し維持する処理は、SCM システムに入力されたコミット・メッセージに基づいて行われます。ALI では、コミット・メッセージのパターンを定義し、要件 ID または不具合 ID を含めることができます。

コミット・パターンを設定するには、次の手順を実行します。

1. 新しいリポジトリをセットアップしているときに **[コミットパターン]** ビューを選択するか、既存のリポジトリの **[コミットパターン]** タブを開きます。
2. **[基本]** または **[詳細]** のどちらかを選択して、パターンの定義方法を選択します。
3. **[基本]** を選択した場合は、すぐ下で説明するように、キーワード、ID プレフィックス文字などのオプションを追加します。
4. **[詳細]** を使用する場合は、後述する **「詳細パターンの定義」(39ページ)** の説明に従って、標準設定パターンを編集します。

基本パターンの定義

1. 不具合または要件のキーワードを追加します。これは、**[追加]** ボタンをクリックし、キーワードまたはフレーズを入力して **[OK]** をクリックすることで行います。キーワードにより、

コミットが不具合または要件のいずれかに関連するものとして認識されます。

コード変更を別の ID（外部追跡ツールの ID など）に関連付けるには、「[コード変更と代替 ID との関連付け](#)」(41ページ)を参照してください。

2. ID プレフィックス文字を入力して、パターンに追加します。プレフィックスが不要な場合は、**[オプション]** を選択します。
3. **[その他のオプション]** ボタンをクリックし、次のオプションを追加します。
 - 標準設定の Tasktop コミットパターンを含める：これを選択すると、Taskop プラグインが ALM と統合されている場合に、Tasktop は、ALI が認識するパターンに応じて標準設定のコミット・メッセージを生成します。
 - 大文字と小文字が区別されるコミット メッセージ：これを選択すると、コミット・メッセージ・パターンで大文字と小文字が区別されます。
 - 複数の不具合または要件の区切り文字：コミット・メッセージ内の複数の不具合または要件を区切る文字を入力します。
 - コミット メッセージ内のキーワードの位置：キーワードをコミット・メッセージの先頭のみで認識するか、任意の位置で認識するかを選択します。
 - ユーザ コミット メッセージの区切り文字：不具合または要件の定義と、フリーテキストのユーザ・コミット・メッセージとを区切る文字を入力します。
4. タブから出て設定をコミットします。

コミット・メッセージのオプションの使用例：メッセージ：fixing defect #100, #101:fix caching and enhance functionality

メッセージの要素：

- fixing defect - 不具合のキーワード
- #100 - 不具合 ID のプレフィックスと不具合 ID
- “.” - 複数の不具合を区切る区切り文字
- “:” - ユーザ・コミット・メッセージの区切り文字
- fix caching and enhance functionality - フリーテキストのユーザ・コミット・メッセージ

詳細パターンの定義

1. デフォルトのコード・パターンを追加または変更します。
2. **【既存のコミットに対してテスト】** をクリックして、変更したコードをテストします。
3. **【大文字と小文字を区別】** を選択すると、メッセージで大文字と小文字を区別します。
4. **【カスタム コミット メッセージのテスト】** ボックスにメッセージを入力すると、そのメッセージをテストできます。
5. タブから出て設定をコミットします。

【標準設定に戻す】 ボタンをクリックすると、変更内容を破棄してデフォルトのキーワードに戻します。

詳細設定の例：

1. 例 1

a. パターン

```
[fixing] REGEX('defects?')IDLIST(DEFECT) | [implementing] REGEX(requirements?')IDLIST(REQ) ):TEXT
```

b. サンプル・メッセージ：

```
"fixing defect #56721:something really serious was fixed"
```

```
"defects #57893,#61432:division by zero"
```

```
"requirement #1:domains"
```

2. 例 2

a. パターン

```
(UNTIL(RE '((BUG)|(REQ))#') (IDLIST(DEFECT lead='((BUG)?#)?' sep=',') | IDLIST(REQ lead='((REQ)?#)?' sep=',') ) {0,} [TEXT]
```

b. サンプル・メッセージ：

```
"This commit fixes BUG#1,#2 and implements REQ#4,REQ#5 making the product faster (resolving BUG#7)."
```

- c. このパターンは、'BUG#' および 'REQ#' というパターンを含む入力と抽出すべてと一致します。このようなパターンは、共通ポリシーの実施には適していませんが、レポートの目的などでレガシー・リポジトリのデータを「読み取り専用」モードでロードする際に使用すると便利です。

3. 例3

a. Tasktop パターン

```
(LISTITEM('Bug Status') - WORD IDLIST(DEFECT lead='DEF' sep='') | Incomplete - WORD IDLIST(REQ lead='REQ' sep='')):TEXT
```

b. 次のような標準設定の Tasktop メッセージと一致します。

```
"OPEN - task DEF10:http://host:9090/qcbin;DEFAULT;ALI_DEV-DEF10"
```

```
"Incomplete - task REQ42:http://host:9090/qcbin;DEFAULT;ALI_DEV-REQ42"
```

コード変更と代替 ID との関連付け

コード変更は、標準設定では ALM のエンティティ ID に基づいて、ALM の要件または不具合にリンクできます。要件や不具合に対して別の ID セットを管理している場合は、その代替 ID にコード変更を関連付けできます。

たとえば、プロジェクトを外部の不具合追跡ルールで同期する場合は、そのツールで生成された ID をコミット・メッセージで参照できます。ALM は、その代替 ID を使用して、コード変更を ALM の要件や不具合にリンクできます。

コード変更を代替 ID に関連付けるには、次の手順を実行します。

1. 前提条件:

- a. 代替 ID が、ALM ユーザー定義フィールドに保存されること。サポートされるフィールド・タイプは**数値**と**文字列**です。
- b. 代替 ID が一意であること。コード変更は、最初に一致した ALM エンティティにリンクされます。

2. SCM リポジトリ・モジュールで、リポジトリを選択し、[コミットパターン] タブをクリックします。基本ビューが選択されていることを確認します。

3. [不具合]または[要件] セクションで、[追加] をクリックします。

4. [キーワードの追加] ダイアログ・ボックスで:

- a. [キーワード] フィールドに、コミット・メッセージで代替 ID を使用していることを示すキーワードまたはフレーズを入力します。たとえば、「Resolve Jira issue」のように入力します。開発者は、そのキーワードまたはフレーズを自分のコミット・メッセージに含める必

必要があります。

- b. [フィールド] リストで、代替 ID を保存するユーザ定義フィールドを選択します。

ここで定義したキーワードまたはフレーズがコミット・メッセージに含まれる場合、ALM は、ユーザ定義フィールドを確認して、コード変更に関連付けるべきエンティティを見つけます。

ブランチの追加とリリースへの関連付け

本項の内容

- 新しいブランチの追加
- ブランチのリリースへの関連付け

新しいブランチを追加するには、次の手順を実行します。

1. ALM サイドバーで、[管理] にある [SCM リポジトリ] を選択し、[ブランチ] タブを開きます。

[追加] をクリックします。

2. ブランチのパスを入力します。リポジトリ・タイプで必要な場合は、[ブランチ] プロパティと [最終変更読み取り] プロパティを次のガイドラインに従って入力します (オプション)。

- SVN, TFS, または Perforce の場合は、[ブランチ] プロパティを定義しないでください。
- [最終変更読み取り] フィールドには、次のいずれかを指定できます。
 - CVS リポジトリの場合は、[最終変更読み取り] フィールドに日時が含まれている必要があります。
 - 他のすべてのリポジトリの場合は、[最終変更読み取り] フィールドにリビジョン番号が含まれている必要があります。

指定されたブランチからすべての変更セットを読み取る場合、このフィールドは空のままにします。

- **CVS :**
ブランチ名が適用されるのは、一部の CVS リポジトリの場合のみです。
- **Perforce :**
デポ名なしでブランチ・パスを設定します。たとえば、ブランチの場所が `//depot/HelloWorld/releases/release-1.0/...` の場合、パスは `/HelloWorld/releases/release-1.0`

とします。ブランチに名前が割り当てられている場合でも、ブランチ・パラメータは指定しないでください。

■ **TFS :**

プロジェクト・パスへのブランチ・パスを \$ 記号なしで指定します。たとえば、プロジェクトの場所が \$/TestApp の場合、ブランチ・パスは /TestApp とします。ブランチ・パスには、プロジェクトの名前のみを指定します。サブフォルダを含むパスはサポートされません。

■ **Git :**

ブランチ・パスは "/" に設定します。ブランチは、実際の Git ブランチ名に設定します。"refs/head/master" ではなく、"master" という形式の簡易ブランチ名のみを使用します。

3. **【送信】** をクリックすると、リンクがテストされてブランチが追加されます。

ブランチをリリースに関連付けるには、次の手順を実行します。

1. ブランチのパスをクリックするか [SCM ブランチの詳細] ボタンをクリックして、[SCM ブランチ 詳細] ウィンドウを開きます。
2. **【追加】** をクリックし、ドロップダウン・メニューでリリースを選択します。
3. 標準設定では、**【開始日】** と **【終了日】** はリリースから取得されます。これは変更できます。日付フィールドをクリックし、必要な日付を入力してください。

指定期間内のブランチのコード変更が、指定したリリースに関連付けられます。このセットは、指定したリリースを選択するとコード変更モジュールで表示されます。

ブランチには、複数のリリースを関連付けることができます。ブランチに複数のリリースが関連付けられる場合、それぞれのリリースは一般に期間が異なります。

ブランチのチェックイン・ポリシーの設定

ブランチでのチェックインは、チェックインのいくつかのプロパティに基づいて制限できます。ロックを使用すると、より細かなアクセス許可と制限を設定できます。

チェックイン・ポリシー機能を使用するには、エージェントのインストールが必要です。詳細については、[「SCM エージェント」\(30ページ\)](#)を参照してください。

注: この機能は、Git リポジトリではサポートされていません。設定が非表示になります。

ブランチのチェックイン・ポリシーを指定するには、次の手順を実行します。

1. **[SCM ブランチの詳細]** ウィンドウの左の表示枠にある **[実施]** をクリックします。
2. **[チェックイン ポリシー]** タブで、次のオプションを選択します。
 - **コミット メッセージと定義済みパターンとの一致が必要**：これは、エージェントによってチェックインが許可されるために、コミット・メッセージが一致する必要があるパターンです。
 - **コード変更の要件参照**：これを選択した場合、コミットは要件を参照している必要があります。
 - **要件タイプが存在**：このオプションを選択する場合は、ドロップダウン・リストから要件タイプを選択します。選択したタイプの要件に関連付けられたチェックインのみがエージェントによって許可されます。
 - **優先度が存在**：このオプションを選択する場合は、ドロップダウン・リストから優先度を選択します。選択した優先度を持つ要件に関連付けられたチェックインのみがエージェントによって許可されます。
 - **コード変更の不具合参照**：これを選択した場合、コミットは不具合を参照している必要があります。
 - **重要度**：このオプションを選択する場合は、ドロップダウン・リストから重要度を選択します。選択した重要度を持つ不具合に関連付けられたチェックインのみがエージェントによって許可されます。
 - **コミットがブロックされている場合、次のメモをシステム メッセージに追加**：コミットがブロックされたときにユーザに送信する、カスタムのシステム・メッセージを入力します。ALLは、**[チェックイン ポリシー]** タブで指定されたオプションに基づいて、ユーザが変更をコミットするのをブロックします。

ブランチのロック・ポリシーを指定するには、次の手順を実行します。

[ロック ポリシー] タブで次のオプションを使用すると、ブランチでのチェックインを制限できます。

- **次の例外を除いて、コミットを許可しない**：指定したブランチでのコミットを許可しないようにします。次を例外を指定できます。
- **ユーザの許可**：ブランチへのコミットを許可されるユーザのリスト。コミットを許可する SCM ユーザのユーザ名を入力します。
- **不具合の許可**：不具合のリスト。これらの不具合に関連付けられたコミットは許可されます。**[不具合の追加]** をクリックし、不具合 ID を入力します。リストから不具合を削除するには、その行を選択して **[不具合の削除]** をクリックします。

- ・ **コミットがブロックされている場合、次のメモをシステム メッセージに追加** : コミットがブロックされたときにユーザに送信する、カスタムのシステム・メッセージを入力します。

コード変更の検出の設定

ALI は、SCM リポジトリに加えられた変更を検出し、コード変更を作業項目（要件、不具合など）に関連付けます。ALI では、2つの変更検出機構（ポーリングとプッシュ）をサポートしています。

ポーリングが有効な場合は、ALI が、指定された SCM リポジトリに新しい変更がないか定期的にチェックします。ポーリングは、SCM エージェントがなくても機能します。

プッシュ機構が有効な場合は、SCM システム上のエージェントが変更を検出します。変更がリポジトリとブランチにコミットされるたびに、エージェントはポリシーをチェックします。コミットが許可されれば、エージェントはコード変更データを ALM にプッシュします。

変更検出を設定するには、次の手順を実行します。

- ・ 新しいリポジトリをセットアップしているときは、**[新規 SCM リポジトリ]** ダイアログ・ボックスの**[変更の検出]** ビューを使用します。
- ・ 既存のリポジトリの場合は、SCM リポジトリ・モジュールでリポジトリを選択し、**[変更の検出]** タブをクリックします。

検出オプションを選択します。

オプション	コメント
SCM から変更を読み取る	<p>このオプションを選択すると、ポーリングが有効になります。</p> <p>変更の自動ロードのスケジュールを設定するには、「同期スケジュールの設定」(13ページ)を参照してください。</p> <p>[同期] ボタンをクリックすると、読み取りを即時実行します。</p> <p>同期プロセスのモニタには、[ツール] メニューにある ALM の[タスク マネージャ] ツールを使用します。</p>
SCM エージェントから送信された変更を受信する	<p>このオプションを選択すると、変更のプッシュが有効になります。</p> <p>SCM エージェントが設定済みの場合にこのオプションを選択すると、変更の処理が直ちに開始されます。詳細については、「サポート対象の SCM システム」(63ページ)を参照してください。</p>

第5章: Force.com との統合のセットアップ

Force.com プラットフォーム用の開発を行うチームは、ALI の標準的な開発機能を利用できます。ソースコードは、クラウドで格納、コンパイル、テストされますが、ALI では、コード、作業項目（要件と不具合）、ビルド、ビルド・メトリック（テスト結果とカバレッジ）間のトレーサビリティが維持されます。

本章の内容

前提条件	47
プロジェクトのデプロイメント、テスト、レポート生成のセットアップ	47
Hudson/Jenkins - Force.com の構成	49

前提条件

Force.com と ALI との統合では、次の条件を満たす必要があります。

- 使用するバージョンがサポートされていることを確認します。詳細については、「[サポート対象の Force.com バージョン](#)」(64ページ)を参照してください。
- Force.com ソース・コードは SCM システムに保存されている必要があります。詳細については、「[サポート対象の SCM システム](#)」(63ページ)を参照してください。
- Hudson または Jenkins プラグインがインストールされている必要があります。
 - SCM の使用をサポートする Hudson/Jenkins プラグイン
 - HP ALI Hudson プラグイン/HP ALI Jenkins プラグイン (ali-bundle の ali-hudson-plugin.hpi/ali-jenkins-plugin.hpi)。
 - HP ALI Hudson/Jenkins Force.com プラグイン (ali-bundle の ali-hudson-salesforce-plugin.hpi/ali-jenkins-salesforce-plugin.hpi)

詳細については、「[HP ALI Hudson/Jenkins プラグインのデプロイメント](#)」(18ページ)を参照してください。

- ソース・コードを統合/ステージング環境にデプロイするビルド管理サーバが構成されている必要があります。
- Apache Ant がインストールされている必要があります (<http://ant.apache.org/> (英語サイト) からダウンロード)。
- HP force-deploy-task がインストールされている必要があります。このタスクは、ALI アーカイブ (/tools/force-deploy-task/force-deploy-task-bundle.zip) にあります。zip ファイルを <ant_install_dir > /lib に展開してください。

プロジェクトのデプロイメント, テスト, レポート生成のセットアップ

ソース・コード, テスト, レポート生成のデプロイメントは, Ant タスクによって実行されます。デプロイ・タスクに必要なクラスは, HP force-deploy-task の一部としてインストールされます。詳細については、「[前提条件](#)」(47ページ)を参照してください。

Ant ビルド・スクリプト (<Force.com プロジェクト・ルート>\build.xml) でタスクを定義します。タスクを既存の build.xml に追加するか, ファイルを作成します。

すべてのテストを実行する例 :

次の例では、構成済みの Force.com 環境にソース・コードをデプロイし、すべてのテストを実行しています。すべてのテストを実行するので、レポートにはプロジェクト全体のコード・カバレッジが含まれます。

```
<project name="Sample usage of deploy task"
  default="deployAndTestAndReport" basedir=".">
  <target name="deployAndTestAndReport">
    <taskdef name="sfdeploy" classname=
      "com.claimvantage.force.ant.DeployWithXmlReportTask"/>
    <delete dir="test-report-xml" quiet="true"/>
    <sfdeploy
      username="<username to force.com environment>"
      password="<password to force.com environment>"
      serverurl="<force.com server URL>"
      deployRoot="<path to source directory>"
      runalltests="true"
      reportDir="test-report-xml" />
  </target>
</project>
```

フィルタに一致するテストを実行する例 :

次の例では、構成済みの Force.com 環境にソース・コードをデプロイし、指定のパターンに一致するテストのみを実行しています。この場合、ALM に完全なコード・カバレッジは提供されません。

```
<project name="Sample usage of deploy task"
  default=" deployAndTestAndReport " basedir=".">
  <target name="deployAndTestAndReport">
    <taskdef name="sfdeploy" classname=
      "com.claimvantage.force.ant.DeployWithXmlReportTask"/>
    <delete dir="test-report-xml" quiet="true"/>
    <sfdeploy
      username="<username to force.com environment>"
      password="<password to force.com environment>"
      serverurl="<force.com server URL>"
      deployRoot="<path to source directory>"
      runalltests="false"
      reportDir="test-report-xml"/>
    <!-- Run only tests with file names that
      match this pattern.Applies only if
      runalltests is false.-->
    <batchtest>
      <fileset dir="src/classes">
        <include name="*Test.cls"/>
      </fileset>
    </batchtest>
```

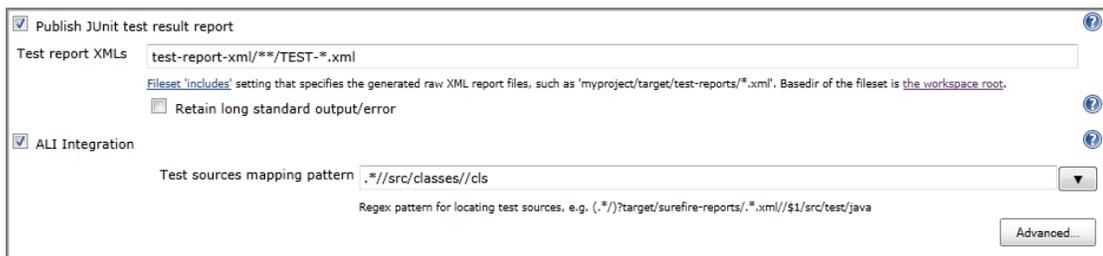
```
</target>
</project>
```

デプロイ・タスクの属性 (上の例の sfdeploy)

要素	属性	説明
sfdeploy		デプロイ定義
sfdeploy	username	Force.com 環境へのログオン名
sfdeploy	password	Force.com 環境へのパスワード
sfdeploy	serverurl	Force.com 環境へのログイン・ページの URL
sfdeploy	deployRoot	フォルダ, クラス, トリガなどを含むソース・コード・ディレクトリのパス
sfdeploy	runalltests	true の場合は, すべてのテストが開始され, プロジェクトのコード・カバレッジがレポートされます。 false の場合は, batchtest 要素で指定されるテストのみが開始されます。コード・カバレッジはレポートされません。
sfdeploy	reportDir	レポートのディレクトリ
batchtest		runalltests="false" の場合に開始するテスト
batchtest/fileset	dir	batchtest テスト・ファイルがある場所
batchtest/include	name	batchtest テストのクラス名パターン

Hudson/Jenkins - Force.com の構成

1. フリースタイルのジョブを作成し, SCM とビルドのトリガを必要に応じて作成します。
2. Invoke Ant というビルド・ステップを追加し, デプロイ・ターゲットを指定します。トピック「[プロジェクトのデプロイメント, テスト, レポート生成のセットアップ](#)」(47ページ)の例の target name="deployAndTestAndReport" を参照してください。
3. ビルド後の処理セクションで, 次のように設定します。

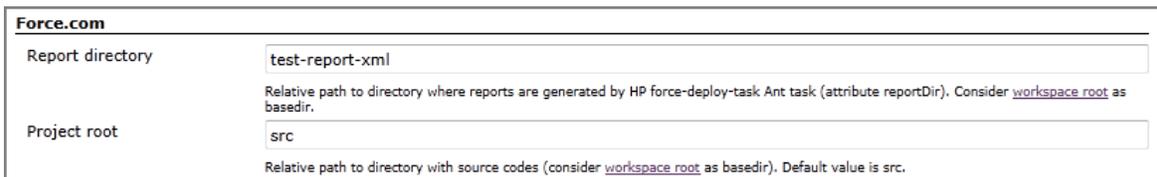


4. テスト・レポート XML で、文字列 test-report-xml を実際のレポートのディレクトリに置き換えます。これは、Ant デプロイ・タスクの reportDir 属性の値と同じです。
5. テスト・ソースのマッピング・パターンで、src を実際のソース・ディレクトリへのパスで置換します。

ほとんどの場合は上記の設定で問題ありませんが、次の場合にはそれぞれ必要な設定を行ってください。

- force-deploy-task が、メインの Ant スクリプトから呼び出される分散 Ant スクリプトで定義されている場合、レポート・ディレクトリ（Ant スクリプトの reportDir 属性の値）を Ali Integration/高度な設定で設定します。
- ソースコード・ディレクトリ（フォルダ・クラス、トリガなどを含む）がワークスペース・ルートの直下にある src ディレクトリにない場合、プロジェクト・ルートを設定します。

設定例：



第6章: SCM の変更とトレーサビリティの管理

リポジトリとブランチのセットアップが完了すると、Application Lifecycle Intelligence (ALI) はデータを集計して ALM のコード変更モジュールに提示します。このモジュールは、プロジェクト内のソース・コードの状態に関する情報を提供し、ソース・コード、要件/不具合、リリース間のトレーサビリティを維持します。

本章の内容

SCM の変更とトレーサビリティのワークフロー	52
コード変更の表示	52
コード変更の影響の表示	53
プロジェクト・レポートの生成	54
グラフの生成	54

SCM の変更とトレーサビリティのワークフロー

SCM の変更を管理しモニタするには、次の手順を実行します。

1. チェックイン・メッセージに対して特定の書式を要求します。詳細については、[「コミット・パターンの設定」\(38ページ\)](#)を参照してください。
2. チェックイン・ポリシーとブランチのロックを設定して、コミットを SCM ブランチのレベルで制御します。詳細については、[「ブランチのチェックイン・ポリシーの設定」\(43ページ\)](#)を参照してください。
3. コード変更を表示します。詳細については、[「コード変更の表示」\(52ページ\)](#)を参照してください。
4. コード変更が不具合と要件に与える影響を表示します。詳細については、[「コード変更の影響の表示」\(53ページ\)](#)を参照してください。
5. チームと開発者の開発アクティビティをレビューし比較します。

次を参照してください。

- [「プロジェクト・レポートの生成」\(54ページ\)](#).
- [「グラフの生成」\(54ページ\)](#).

コード変更の表示

コード変更モジュールにアクセスするには、ALM サイドバーで **開発連携** を選択し、**コード変更** をクリックします。

[コード変更] グリッドの各レコードは、SCM にコミットされたファイルが1つか複数かには関係なく、1つのコミット・アクションに関するデータを表します。

レコードの色は、コード変更がリンクされている作業項目のタイプを示します。

色	説明
青	要件にリンク
緑	不具合にリンク
赤	未割り当て（どの作業項目にも未リンク）

[メッセージ] カラムには、開発者が入力したコミット・メッセージが含まれます。[ステータスメッセージ] には、コミット・メッセージが事前定義のパターンに一致しないときに、エラーまたは警告が表示されます。

[表示] メニューの下の [情報パネル] を有効にすると、テーブルの下にパネルが開き、関連するファイルと作業項目を表示するためのタブが表示されます。このいずれかのタブにアスタリスク (*) が付いている場合、その表示枠にはデータがあります。単純なテキストを表示する表示枠のタブには、テキストがあってもアスタリスクは表示されません。

コード変更の詳細

[コード変更の詳細] ビューには、追加の詳細情報が表示されます。

[コード変更の詳細] を表示するには、次の手順を実行します。

[コード変更の詳細] ウィンドウを開くには、コード変更の [リビジョン] フィールドをクリックするか、[コード変更の詳細] ボタン  をクリックします。

[前回との差異] カラムの [変更の表示] をクリックすると、前バージョンのファイルとの差異が表示されます。

コード変更の影響の表示

変更影響レポートは、要件および不具合別にコード変更データを集計します。このレポートには、関連する変更のサイズと、コードをチェックインした開発者が表示されます。レポート右上にあるグラフには、工数の分布が要件、不具合、未割り当てのコード変更別に表示されます。工数は、KLOC (千行のコード) 単位で表現されます。

変更影響レポートを表示するには、次の手順を実行します。

コード変更モジュールで、ツールバーまたはグリッドから期間とリリースを選択し、[レポートの表示] をクリックします。他のフィルタの値はすべて無視されます。

変更影響レポートの下部には、次のオプションがあります。

- **未割り当ての変更を表示** : 不具合または要件に関連付けられていない変更を表示します。
- **このレポートを印刷**
- **このレポートを電子メールで送信** : 使用している電子メール・クライアントで、レポートの URL を含む電子メールを作成します。

プロジェクト・レポートの生成

ALI 拡張機能では、ALM の標準レポートに加えて、事前定義されたレポート・テンプレートが提供されています。

次のテンプレートを使用できます。

- ALI - コード変更テンプレート
- ALI - 不具合概要テンプレート
- ALI - 要件概要テンプレート
- ALI - ビルド・テンプレート

上記のテンプレートをベースに、コード変更とビルドに関するプロジェクト・レポートを作成できます。

プロジェクト・レポートの作成方法の詳細については、『HP Application Lifecycle Management ユーザズ・ガイド』を参照してください。

グラフの生成

次のグラフを作成できます。

グラフ	説明
コード変更 - 進捗グラフ	一定期間内のある時点において、それまでに ALM プロジェクトで行われてきたコード変更の累計数を示します。
コード変更 - サマリグラフ	ALM プロジェクトに現在存在するコード変更の数を示します。
コード変更 - トレンドグラフ	ALM プロジェクトのコード変更フィールドについて、履歴情報を表示間隔ごとに示します。

グラフを作成するには、次のいずれかを実行します。

- **グラフ・ウィザードを起動します。**

コード変更モジュールで、[アナリシス] > [グラフ] > [グラフ ウィザード] を選択します。

[グラフ ウィザード] ウィンドウで、適切なグラフを選択してウィザードの指示に従います。

- **コード変更モジュールで事前定義のグラフを作成します。**

[アナリシス] > [グラフ] を選択し、定義済みグラフを1つ選択します。

- **アナリシス・ビューでグラフを作成します。**

ALM サイドバーの [ダッシュボード] で、[アナリシス ビュー] を選択します。フォルダを右クリックし、[新規グラフ] を選択します。

グラフの作成方法の詳細については、『HP Application Lifecycle Management ユーザーズ・ガイド』を参照してください。

第7章: ビルド・アクティビティのモニタ

ビルド・システムとの統合のセットアップが完了すると、ALI は、関連コードと実装された作業項目と併せて、ビルドの結果をモニタします。

ビルド・モジュールでは、高速ビルド、夜間ビルド、統合ビルドのステータスと結果を参照し、報告されたエラーに対応できます。ビルド・システムの構成に応じて、ユニット・テストの結果、成功または失敗したテストの件数、コード・カバレッジ、ソース・コード、不具合などの統計情報のモニタが可能です。

ビルド情報を示すプロジェクト・レポートとグラフを作成できます。詳細については、「[ビルド・レポートの表示](#)」(57ページ)および「[ビルド・グラフの生成](#)」(58ページ)を参照してください。

本章の内容

ビルドの表示	57
ビルド・レポートの表示	57
ビルド・グラフの生成	58
ビルドとテストのトレーサビリティの表示	59

ビルドの表示

コード・ビルド・モジュールにアクセスするには、ALM サイドバーで **開発連携** を選択し、**ビルド** をクリックします。

[ビルドステータス] フィールドには、ステータスが次の色で表示されます。

色	説明
緑	成功したビルド
黄	警告：ビルドに問題があることを示します
赤	失敗したビルド

ビルドの詳細を表示するには、ビルド番号をクリックするか、ビルドを選択して **ビルド詳細** ボタン  をクリックします。

ビルド・レポートの表示

ビルド・サマリ・レポート

ビルド・サマリ・レポートは、指定された期間とビルド・カテゴリのビルドについて、その概要を表示します。グラフでは、SCM アクティビティの情報と、選択したリリースで開いている不具合をまとめて表示します。

ビルド・サマリ・レポートを表示するには、次の手順を実行します。

[**レポートの表示**] ボタンをクリックして、サマリ・レポートを生成します。

詳細なビルド・レポート

詳細なビルド・レポートには、要件、テスト、通知された不具合、新しい不具合、解決済みの不具合など、ビルドに関する情報が表示されます。

詳細なビルド・レポートを表示するには、次の手順を実行します。

1. サマリ・レポート内でビルドのリンクをクリックするか、ビルドの詳細ページを開いて [**レポートの表示**] ボタンをクリックします。
2. [**未割り当ての変更を表示**] をクリックすると、ビルド内のコード変更で、要件または不具合に関連付けられていないものが表示されます。

ビルド変更レポート

ビルド変更レポートでは、ビルドの影響を受ける要件と、ビルドによって変更、終了、作成された不具合の概要が表示されます。

ビルド変更レポートを表示するには、次の手順を実行します。

1. ビルドを選択し、**「ビルド詳細」**をクリックします。
2. **「ビルド詳細」**ページで**「レポートの表示」**をクリックして、ビルド変更レポートを生成します。

ビルド・グラフの生成

次のタイプのグラフを生成できます。

グラフ	説明
ビルド - サマリ グラフ	X軸として指定する各カテゴリのビルド数を表示します。
ビルド - トレンド グラフ	指定するフィールド別に、ビルドの履歴を表示期間ごとに示します。

グラフを作成するには、次のいずれかを実行します。

- **グラフ・ウィザードを使用します。**

ウィザードを起動するには、次のいずれかを実行します。

- ALM サイドバーの**「ダッシュボード」**で**「アナリシス ビュー」**を選択し、フォルダを右クリックして**「グラフ ウィザード」**を選択します。**「グラフ ウィザード」**ウィンドウが開いたら、エンティティ名を「ビルド」に変更します。

または：

- ビルド・モジュールで、**「アナリシス」** > **「グラフ」** > **「グラフ ウィザード」**を選択します。

- **定義済みのグラフを生成します。**

ビルド・モジュールで**「アナリシス」** > **「グラフ」**を選択し、定義済みグラフを1つ選択します。

- **アナリシス・ビューでグラフを生成します。**

ALM サイドバーの**「ダッシュボード」**で、**「アナリシス ビュー」**を選択します。フォルダを右クリックし、**「新規グラフ」**を選択します。

グラフの作成方法の詳細については、『HP Application Lifecycle Management ユーザーズ・ガイド』を参照してください。

ビルドでは、プロジェクト・レポートも提供されています。詳細については、「[プロジェクト・レポートの生成](#)」(54ページ)を参照してください。

ビルドとテストのトレーサビリティの表示

特定のビルドと、特定のテスト・セットおよびテスト実行を関連付けて、相互の関係を追跡できます。

テストを実行する前に、テストセットを定義し、テスト・セットに使用するビルドを定義します。つまり、すべてのテストは、選択したビルドで実行されることになります。テスト・セットとテスト実行の結果は、この関連付けに基づいてビルドごとにレビューできます。

ビルドとテストの関連付けは、次の場所で設定および表示できます。

場所	ビルドとテストのトレーサビリティ
【ビルド】モジュール > 【ビルド レポート】	【ビルド詳細】ダイアログ・ボックスで【レポートの表示】をクリックすると、ビルド変更レポートが表示されます。
【ビルド】モジュール > 【テスト】タブ	【ビルド】モジュール・ウィンドウまたは【ビルド詳細】ダイアログ・ボックスで、【テスト】タブをクリックします。
【テスト ラボ】モジュール > 【テストセットの詳細】ダイアログ・ボックス	【テスト ビルド】フィールドには、選択したテスト・セットに割り当てられているビルド番号が表示されます。ビルドを選択し、テスト・セットと関連付けます。
【テスト ラボ】モジュール > 【マニュアルランナー】ダイアログ・ボックス	【テスト ビルド】フィールドには、選択した実行に割り当てられているビルド番号が表示されます。
【テスト ラボ】モジュール > 【実行の詳細】ダイアログ・ボックス	【テスト ビルド】フィールドには、選択した実行に割り当てられているビルド番号が表示されます。

テスト・ラボ・モジュールの詳細については、『HP Application Lifecycle Management ユーザーズ・ガイド』を参照してください。

第8章: 開発アクティビティのモニタ

ビルド・サーバとビルド構成のセットアップが完了すると、ALM のリリース、不具合、要件の各モジュールに「開発アクティビティ」タブが表示されます。この表示枠には、選択した ALM エンティティの詳細情報と統計が表示されます。

リリース、不具合、または要件に関連付けられた開発アクティビティがあると、緑色の星型アイコンが「開発アクティビティ」タブに表示されます。

リリース・モジュールでの開発アクティビティのモニタ

「開発アクティビティ」タブで、次の操作を行います。

- 「**変更**」をクリックすると、他のビルドのアクティビティが表示されます（関連するビルドが複数ある場合）。
- 「**カバレッジ**」をクリックすると、カバレッジ・レポートが表示されます。
- 「**ユニットテスト**」をクリックすると、ユニット・テスト・レポートが表示されます。

「ビルドステータス」領域の成功率は、ステータスが成功または警告で完了したビルドの割合を示します。「**ビルドの構成**」をクリックすると、構成を変更できます。詳細については、「[ビルド構成の追加](#)」(22ページ)を参照してください。

要件モジュールでの開発アクティビティのモニタ

「開発アクティビティ」タブで、次の操作を行います。

- 「**変更**」をクリックすると、他のビルドのアクティビティが表示されます（関連するビルドが複数ある場合）。
- 「**カバレッジ**」をクリックすると、カバレッジ・レポートが表示されます。
- 「**ユニットテスト**」をクリックすると、ユニット・テスト・レポートが表示されます。
- コード変更のメッセージ・リンクをクリックすると、「**コード変更の詳細**」ページが表示されます。

不具合モジュールでの開発アクティビティのモニタ

不具合に関連付けられたコード変更とアクティブな開発者を表示するには、次のいずれかを実行します。

- グリッドの下のテーブルで **開発アクティビティ** タブをクリックします。
- **不具合の詳細** ウィンドウで **開発アクティビティ** を選択します。

コード変更の詳細を表示するには、**コード変更** のメッセージ・リンクをクリックします。

ALI の **不具合の詳細** ページには、3 つの値が追加されています。

- **検出ビルド** - 不具合が検出されたビルド。
- **終了ビルド** - 不具合が解決されたビルド。
- **起因するコード変更** - この不具合の原因となったコード変更。

HP ALM 12.50 との統合をサポートする静的コード分析ツールが設定されている場合、上記の値は自動的に設定されます。ツールの設定を行っていない場合は、ドロップダウン・リストを使用して設定できます。

上記のフィールドのデータは、**ビルド詳細** ページの **不具合** セクションに表示できます。このデータは、**ビルド・サマリ・レポート** にも含まれます。詳細については、[「ビルド・レポートの表示」\(57ページ\)](#)を参照してください。

付録A: サポート対象のシステム

この付録では、ALI がサポートするシステムについて詳しく説明します。

本付録の内容

サポート対象の SCM システム	63
サポート対象のビルド・システム	63
サポート対象の Force.com バージョン	64

サポート対象の SCM システム

ALI は、次の SCM システムをサポートします。

システム	バージョン
Subversion	1.7.x, 1.8.x (1.7.1, 1.7.3, 1.8.3 でテスト済み)
Concurrent Versions System (CVS)	1.11.x, 1.12.x (1.11.23, 1.12.13 でテスト済み)
Microsoft Team Foundation Server (TFS)	2012, 2013 (TFS 2012, 2013 でテスト済み)
Perforce	2013.1, 2014.2 (2013.1, 2014.2 でテスト済み)
Git	1.8.x, 2.1.x (1.8.5, 2.1.2 でテスト済み, Linux のみ)

SCM エージェントは、次のオペレーティング・システムへのデプロイメントをサポートします。

Red Hat Enterprise Linux 6.x (32 ビット, 64 ビット)
SuSE Linux Enterprise 11.x (32 ビット, 64 ビット)
Windows 2008 R2 Server (64 ビット)
Windows 2012 Server (64 ビット)
Windows 2012 R2 Server (64 ビット)

前提条件 :

MS PowerShell 2.0 以降を Windows 環境にインストールし、スクリプトの実行を有効に設定します。Linux/Unix エージェント・スクリプトは BASH を使用します。複数のリポジトリのプッシュをサポートするために、BASH のバージョンは 4.0 以降を使用してください。

制限事項 :

Git : ALI は、コミットの履歴を変更する Git の機能をサポートしません。Git リポジトリ内でコミットの履歴が変更された場合、影響が発生したすべてのブランチについて、ALI が追跡しているコミット履歴全体を再ロードする必要があります。コミット履歴を再ロードするには、ブランチを削除してから追加します。

サポート対象のビルド・システム

ALI は、次のビルド・システムをサポートします。

システム	バージョン
Jenkins - 長期サポート・リリース	1.532.1, 1.565.3 (1.532.1, 1.565.3 でテスト済み)
Hudson - 製品版	3.1.0, 3.2.1 (3.1.0, 3.2.1 でテスト済み)
Microsoft Team Foundation Server (TFS)	2012, 2013 (TFS 2012, 2013 でテスト済み)

ALI は、ビルド・サーバ向けのプラグインを提供しています。このプラグインは、ビルドの正確な情報とメトリックスを自動的に抽出し、ALM に保存します。

ALI は、次の開発メトリック・ツールをサポートします。

ツール	バージョン
ユニット・テスト	<ul style="list-style-type: none"> JUnit - サポート対象のビルド・システムにバンドルされているバージョンでテスト済み TestNG - Hudson 0.8, Jenkins 0.28, Jenkins 0.32 でテスト済み NUnit - Hudson 0.10, Jenkins 0.14, NUnit framework 2.5.10 バージョン, TFS 2012, 2013 でテスト済み Visual Studio Managed Unit Testing Framework (MSTest - VS 2012, 2013 の一部)
コード・カバレッジ・アナリシス	<ul style="list-style-type: none"> Cobertura - Hudson 1.1, Jenkins 1.3 でテスト済み NCover - Hudson 0.3, Jenkins 0.3, NCover 3.4.18.6937 x86 (試用) バージョンでテスト済み Visual Studio 2012, 2013 コード・カバレッジ

サポート対象の Force.com バージョン

Force.com との統合は、Force.com API バージョン 22.0 でテスト済みです。