

HP Application Lifecycle Intelligence

HP ALM プラットフォーム向け

ソフトウェア・バージョン: 2.6

ユーザーズ・ガイド

ドキュメント・リリース日: 2012年8月(英語版)

ソフトウェア・リリース日: 2012年6月(英語版)



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、FAR12.211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 - 2012 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) の商標です。

Microsoft®およびWindows®は、米国におけるMicrosoft Corporationの登録商標です。

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

目次

ユーザース・ガイド	1
目次	5
はじめに	7
本書の構成	9
ALI の設定	10
ALI の前提条件	11
ALI 拡張機能の有効化	11
ALI への移行	11
SCM システムの統合	13
サポート対象のSCMシステム	14
SCM エージェント	15
リポジトリの設定	15
リポジトリの追加と編集	16
外部リポジトリ・ビューアの設定	17
ブランチと実施の設定	19
ブランチの設定	20
コミット・パターンの設定	21
変更の検出の設定	23
ビルド・システムの統合	25
サポート対象のビルド・システム	26
Hudson/Jenkins のインストール	26
HP ALI Hudson/Jenkins プラグインのインストール	26
HP ALI Hudson/Jenkins プラグインの設定	27
ビルド・サーバの追加	28
ビルド構成の追加	29
ビルド構成にあるSCM構成の再利用	29
ビルド構成の不具合フィルタの設定	30
ALI プロジェクト・リストのカスタマイズ	30
コード問題の追跡	31
コード問題の追跡の設定	31

ビルド・サーバの検出の設定.....	32
Force.com の統合.....	33
前提条件.....	34
プロジェクトのデプロイメント, テスト, レポート作成.....	34
Hudson/Jenkins に関する Force.com の構成.....	36
SCM の変更とトレーサビリティのモニタ.....	37
コード変更テーブルの表示.....	38
コード変更の詳細.....	38
変更影響レポートの表示.....	39
プロジェクト・レポートの作成.....	39
グラフの生成.....	39
開発 アクティビティのモニタ.....	42
開発 アクティビティ - リリース.....	43
開発 アクティビティ - 不具合.....	43
開発 アクティビティ - 要件.....	43
[不具合詳細]での開発 アクティビティのモニタ.....	43
ビルドとテストのトレーサビリティ.....	45
ビルドとテストのトレーサビリティの表示.....	46
ビルド・アクティビティのモニタ.....	47
ビルドの表示.....	48
ビルドの詳細.....	48
ビルド・レポート.....	50
ビルド・グラフの作成.....	51

はじめに

Application Lifecycle Intelligence(ALI) は、ALM のトレーサビリティを実現する各種機能、レポート、メトリックを提供します。ALM のユーザは、ALI を使用することによって適切な情報に基づいた意志決定が可能になります。

ALI は、ソース・コード管理(SCM) とビルド管理の統合を通じて HP ALM プラットフォーム・モデルを拡張し、ALM エンティティ(リリース、要件、不具合、テスト・セット、テスト実行など)、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・アナリシスを相互に関連付けます。その結果、HP ALM は、ビジネス・アクティビティと開発アクティビティの両方の情報を提供できる集中システムとして機能します。さらに ALI が提供するビルド・テストのトレーサビリティでは、ビルドとテストを関連付け、追跡することが可能です。

SCM にコミットする変更の管理

ALI では、SCM ポリシーを定義して、コミットする変更を制御することが可能です。SCM ポリシーを実施することにより、開発者は事前に定義したガイドラインとベスト・プラクティスに従って作業を進めることができます。また、開発チームのリーダーは、適切な機能を実装し、コミットされた変更セットに必要なメタデータを追加するなど、開発者の作業が適切に行われていることを確認できます。さらに、重大な不具合の修正や、リリースのコード・ベースの完全ロックなど、コードの安定化において大きなメリットがあります。

詳細については、「[ブランチと実施の設定](#)」(19ページ)を参照してください。

開発者、QA マネージャ、プロジェクト・マネージャは、次の項目を簡単にレビューできます。

- 要件または不具合を実装するコード変更
- 特定のビルドで実装されたコード変更
- 特定の期間、リリース、ビルド、担当者によって実装された内容
- 変更の影響を受ける要件と不具合
- 特定の要件に関連付けられた変更の量
- 変更セットに含まれる差異の行ごとの比較
- 要件または不具合に関連しない変更
- ビルドと品質のメトリック
- コード・カバレッジ、テスト結果、変更の量のサマリー情報
- ビルドとテストのトレーサビリティ

詳細については、次の項を参照してください。

- 「[SCM の変更とトレーサビリティのモニタ](#)」(37ページ)
- 「[開発アクティビティのモニタ](#)」(42ページ)
- 「[ビルドとテストのトレーサビリティ](#)」(45ページ)
- 「[ビルドの詳細](#)」(48ページ)

Application Lifecycle Intelligence では、ALM の要件、テスト、不具合と、オープンソースや市販の SCM、ビルド・システム、ユニット・テスト・ツール、コード・カバレッジ・アナリシス・ツールを統合することによって、ALM トレサビリティを実現します。

ALI 2.6 では、次との統合をサポートします。

- **SCM システム**

- Subversion(SVN)
- Concurrent Versions System(CVS)
- Microsoft Team Foundation Server(TFS)
- Perforce
- Git

この統合は SCM クライアントに依存しません。したがって、コマンド・ライン・ユーティリティや Tasktop プラグインを使用した Eclipse IDE など、現在使用している SCM クライアントから変更をコミットできます。詳細については、「[サポート対象の SCM システム](#)」(14ページ)を参照してください。

- **ビルド・システム**

- Jenkins
- Hudson

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、ビルドの情報とメトリックを自動抽出し、ALM モデルに読み込みます。詳細については、「[ビルド・システムの統合](#)」(25ページ)を参照してください。

- **ユニット・テスト**

- JUnit
- TestNG
- NUnit

- **コード・カバレッジ・アナリシス**

- Cobertura
- NCover

注: 本書では、Application Lifecycle Intelligence を ALM と組み合わせて使用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。

本書の構成

本書は、次の章で構成されます。

章	説明
第 1 章「ALI の設定」(10ページ)	Application Lifecycle Intelligence を使用する上で必要となる前提条件と設定情報について説明します。
第 2 章「SCM システムの統合」(13ページ)	Application Lifecycle Intelligence を SCM システムと統合する方法について説明します。
第 3 章「ビルド・システムの統合」(25ページ)	Application Lifecycle Intelligence をビルド・システムと統合する方法について説明します。
第 4 章「Force.com の統合」(33ページ)	Application Lifecycle Intelligence を Force.com と統合する方法について説明します。
第 5 章「SCM の変更とトレーサビリティのモニタ」(37ページ)	変更セットのデータを表示する方法について説明します。
第 6 章「開発 アクティビティのモニタ」(42ページ)	開発 アクティビティを表示する方法について説明します。
第 7 章「ビルドとテストのトレーサビリティ」(45ページ)	ビルドとテストの関連付けとその追跡を行う方法について説明します。
第 8 章「ビルド・アクティビティのモニタ」(47ページ)	ビルド・データを表示する方法について説明します。

第1章

ALI の設定

本章では, Application Lifecycle Intelligence(ALI) のインストール要件とセットアップ手順について説明します。

ALI の旧バージョンからのアップグレードについては, 「ALI への移行」(11ページ)を参照してください。

本章の内容

ALI の前提条件.....	11
ALI 拡張機能の有効化.....	11
ALI への移行.....	11

ALI の前提条件

ALI では、統合の対象となる SCM とビルド・システムが必要です。詳細については、「サポート対象の SCM システム」(14ページ)および「サポート対象のビルド・システム」(26ページ)を参照してください。

ALI 拡張機能の有効化

ALM サイト管理では、ALI を必要とする ALM プロジェクトごとに、ALI 拡張機能を有効にする必要があります。拡張機能は、既存のプロジェクトで有効にするほかに、プロジェクトの新規作成時に有効にすることもできます。

ALI 拡張機能を有効にする場合は、次の点に注意してください。

- プロジェクトに対して有効にした拡張機能を、後で無効にすることはできません。
- プロジェクトがアクティブな状態の場合、非アクティブにしてから有効にしてください。拡張機能が有効になったら、プロジェクトを再度アクティブにします。
- プロジェクトで拡張機能を有効化する処理には時間がかかることがあります。

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。

ALI への移行

ALI 2.6 への移行

旧バージョンからのアップグレード:

- 「**ALI 2.0 への移行**」にある手順を実行します。
- コード変更モジュールで、すべてのユーザは[メッセージ]カラムを追加する必要があります。移行が完了すると、このカラムは表示されなくなります。

ALI 2.0 への移行

Perforce の移行: Application Lifecycle Intelligence バージョン 1.1 からバージョン 2.0 にアップグレードする場合、ALM で処理する Perforce データを移行するには、最新のデータを Perforce から削除してから再ロードする必要があります。

Perforce SCM からすべてのデータを移行するには、各 Perforce リポジトリで次の手順を実行します。

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。

第2章

SCM システムの統合

本章の内容

サポート対象のSCMシステム.....	14
リポジトリの設定.....	15
ブランチと実施の設定.....	19
コミット・パターンの設定.....	21
変更の検出の設定.....	23

サポート対象のSCM システム

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。サポート対象のシステムでは、プッシュ機構を使用してコミット・メッセージ・ポリシーを実施する場合を除き、特に追加インストールをしなくても変更セットをロードできます。プッシュ機構が必要な場合は、SCM エージェントを SCM システムにインストールします。詳細については、「SCM エージェント」(15ページ)を参照してください。

TFS システムもネイティブ・サポートの対象ですが、TFS システムを完全サポートするには、TFS エージェント Web サービスのインストールが必要になります。たとえば、変更された行数をカウントする Web サービスがありますが、このサービスをインストールしないと、[行]フィールドの値は常に「0」のままになります。

注: ALI 2.6 は、SvnBridge を使用した旧バージョンの TFS システムのアプローチをサポートしますが、このアプローチの使用は ALI の最新バージョンではお勧めしません。TFS システムのネイティブ・サポートの使用をお勧めします。

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。

- Subversion バージョン: 1.6.*、1.7*(1.6.11、1.6.16、1.7.0、1.7.3 でテストを実施)
- CVS バージョン: 1.11.*、1.12.*(1.11.22、1.11.23、1.12.13 でテストを実施)
- TFS バージョン: Team Foundation Server 2010 (TFS 2010、Windows のみでテストを実施)
- Perforce バージョン 2010.2 (2010.2/334844 でテストを実施)
- Git バージョン: 1.7.9*、1.7.10*(1.7.8.5、1.7.9.4、1.7.10.0、Linux のみでテストを実施)

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 Server (32 ビット、64 ビット)
- Windows 2008 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。

SCM エージェント

PUSH 機構で必要となる SCM システムに関連するアプリケーションの 1 つです。エージェントは、スクリプト群や独自設計のアプリケーションで構成され、SCM システムでリッスンする設定を行った SCM サーバ上にインストールされます。設定済みのリポジトリとブランチに対して変更がコミットされると、エージェントはポリシーをチェックし、コミット可能であれば変更セットを 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、「[変更の検出の設定](#)」(23ページ)を参照してください。

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`

CVS_Linux

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

CVS_Windows

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

TFS エージェントのインストール手順

`agents\scm-integration\windows\scm-agent-tfs.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`

リポジトリの設定

Application Lifecycle Intelligence (ALI) で SCM システムからコード変更をロードし、作業項目 (要件/不具合)、コード変更、定義済みリリースの自動トレーサビリティを開始するには、SCM リポジトリの設定が必要です。

本項の内容

リポジトリの追加と編集..... 16

リポジトリの追加と編集

ALI は、Subversion, CVS, TFS, Perforce, Git をサポートします。詳細については、「サポート対象のSCM システム」(14ページ)を参照してください。

リポジトリを追加または編集するには、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. 新しいリポジトリを追加するには、[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. 新しいウィンドウが開いたら、リポジトリの名前、場所、ユーザ名とパスワードを必要に応じて入力します。

注:場所には、リポジトリの完全パス(`http://host/svn/repo` など)を指定します。不明な場合は、システム管理者に確認してください。

Perforce: ALI リポジトリは、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、デポ名で構成され、`host:port//depot_name`のような形式で指定されます。

Subversion: リポジトリの場所は、実際のリポジトリのルートをポイントする必要があります。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。

TFS: 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構成され、`http://host:port//path_to_project_collection`のような形式で指定されます。

CVS: エイリアス・プロパティに完全パス: `CVSROOT`(例えば、`:pserver:username:password/cvsrepo` など)を入力します。これは、ビルド・システムでの設定と同一の内容です。CVSRoot プロパティは、pserver プロトコルでは必須です。

Git: リポジトリの場所には、Git URL 全体を指定します。サポート対象のプロトコルは、`http`, `https`, `Git`, `ssh` です。たとえば、`https://github.com/hp/ali.git` などと指定します。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。

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。

4. 変更が必要なリポジトリ・プロパティがある場合は、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。

CVS: CVS リポジトリでは、CVS プロトコルの `pserver`、変更セットのロードを開始する `initial date` 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。

TFS: 行数のカウント・サービスの URL がデフォルト値 (`repository_protocol://repository_host:9080/AliTfsServices/Service/CountLinesService.svc`) 以外の場合、ブランチの生成で警告が発生し、ロードされた変更セットの変更行数は 0 となります。

Git: 必須の "directory for clone" プロパティの設定が必要です。これは、リモート・リポジトリのミラーの生成先となる絶対パスを示し、パスは ALM リポジトリ・フォルダ内で指定する必要があります。場所ごとに異なるフォルダの指定が必要です。リポジトリの場所を変更する場合、クローン・ディレクトリのプロパティを変更するか、フォルダの内容を削除してください。

5. [送信] をクリックします。ALI はリポジトリへの接続をテストしてから、リポジトリをリストに追加します。

リポジトリの追加では、変更の検出オプションとコミット・パターン・オプションを設定できます。

詳細については、「変更の検出の設定」(23ページ)および「コミット・パターンの設定」(21ページ)を参照してください。

外部リポジトリ・ビューアの設定

ALI にはリポジトリ・ビューアが付属しているので、特に設定を行わなくてもファイルの差分や詳細を表示できます。また、ViewVC など事前定義されたリポジトリ・ブラウザの使用も可能です。

次に示すプロパティは、SCM リポジトリの設定時に入力され、リポジトリの詳細タブで変更できます。

差分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、ファイルの差分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 変更セット・テーブルからリンクを作成できます。変更セット内の各ファイルには、差分ビュー (現在の変更セットのリビジョンと旧リビジョンの差分) 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が含まれ、現在選択している変更セットのコンテンツから、実行時に UI で展開されます。

代入または展開可能なタグ変数

`${filePath}` ... リポジトリ内のファイルのパス。

`${revision}` ... 最新の変更セットのリビジョン。

`${fromRevision}` ... 最新の変更セット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: (TFS) TFS に格納されている 2 つのファイルの差分を表示する場合、次のアドレスが指定されます。

```
http://TFSServer:8080/tfs/_COLLECTION_  
/web/diff.aspx?opath={originalItemPath}&ocs={originalItemChangeset}&mpath=  
{modifiedItemPath}&mcs={modifiedItemChangeset}
```

説明:

- TFSServer: TFS サーバのアドレス。
- _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 のオリジナルの変更セット 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 の変更後の変更セット ID

注: (Git) 差分リンク用 Git テンプレートには、次の変数が含まれます。

`${blobId}` - blob SHA

`${commitId}` - commit SHA

`${prevBlobId}` - 旧バージョンの blob SHA

`${prevCommitId}` - 親コミットの commit SHA

`${branchName}` - ブランチの名前

`${filePath}` - Git リポジトリ内のファイル・パス(先頭に "/" なし)

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 変更セット・テーブルからリンクを作成できます。変更セット内の各ファイルには、ファイル・ビュー(最新の変更セットの範囲に含まれる、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/変数が含まれ、現在選択している変更セットのコンテンツから、実行時に UI で展開されます。

代入または展開可能なタグ/変数

`${filePath}` ... リポジトリ内のファイルのパス。

`${revision}` ... 最新の変更セットのリビジョン。

注: (TFS) TFS に格納されているファイルのソースを表示する場合、次のアドレスが指定されます。

`http://TFSServer:8080/tfs/_COLLECTION_/web/view.aspx?&path={itemPath}&cs={itemChangeset}`

説明:

- TFSServer: TFS サーバのアドレス。
- _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 変更セット ID

注: (Git) ファイル・リンク用 Git テンプレートには、次の変数が含まれます。

`${blobId}` - blob SHA

`${commitId}` - commit SHA

`${branchName}` - ブランチの名前

`${filePath}` - Git リポジトリ内のファイルパス(先頭に "/" なし)

ブランチと実施の設定

ブランチの設定には、既存のリポジトリの[ブランチ]タブを使用します。ブランチとは、独立したリポジトリのラインです。

ブランチは、リリースと関連付ける必要があります。さらに変更の検出オプションを設定することにより、変更を変更セットの形式でコード変更テーブルに表示できるようになります。

また、コミット・パターンでは、ブランチの変更をコミットする操作を許可または許可しない条件を指定できます。

SCM リポジトリで設定されているエージェントがあると、ポリシーが実施されます。

エージェントの詳細については、「SCM エージェント」(15ページ)を参照してください。

本項の内容

ブランチの設定.....	20
--------------	----

ブランチの設定

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. [ブランチ]タブを開いて[追加]をクリックします。
 - a. ブランチのパス名を指定します。オプションで、*Branch* プロパティと *Last Change Read* プロパティを指定します。

CVS: ブランチ名が適用されるのは、一部の CVS リポジトリのみです。SVNブリッジを使用する SVN または TFS では、このプロパティを指定しないでください。[最終変更読み取り]フィールドは、SVN では最終リビジョン、CVS では変更セットを読み取った日時です。所定のブランチから変更セットをすべて読み取る必要がある場合、このフィールドは空白のままにしてください。

Perforce: Perforce を使用する場合、デポ名なしのブランチ・パスを指定します。たとえば、ブランチの場所が `//depot/HelloWorld/releases/release-1.0/...` である場合、パスは `/HelloWorld/releases/release-1.0` となります。ブランチに名前が割り当てられている場合でも、`branch` 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'` となります。

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: Git を使用する場合、ブランチ・パスは必ず `/` に設定します。フィールド名は、実際の Git ブランチ名に設定します。簡易ブランチ名は、`"refs/head/master"` ではなく、`"master"` という形式のみを使用します。

2. [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. [既存のリリースのリンク]ボタンをクリックし、ドロップダウン・メニューで、必要なリリースのラジオ・ボタンをチェックします。
3. デフォルトでは、開始日と終了日がリリースから取得されます。変更するには、日付フィールドをクリックして日付を入力します。

選択した期間に含まれるブランチの変更セットが、指定したリリースに関連付けられます。この変更セットは、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。これ以外のリリースもブランチで指定することが可能ですが、通常はそれぞれ別の期間を指定します。

ブランチでコミットを実施する条件となるチェックイン・ポリシーを指定するには、次の手順を実行します。

注: この機能は、Git リポジトリでは使用できません。設定しても無視されます。

1. [SCM ブランチの詳細] ウィンドウの左パネルにある[実施]をクリックします。
2. 次のオプションを選択します。
 - [チェックイン ポリシー] タブの[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]は、エージェントが拒否する対象として事前定義したパターンと、コミット・メッセージが一致しない場合にコミットするオプションです。
 - [変更セットの要件参照]は、コミットごとに既存の要件を必ず参照します。
 - [要件タイプが存在]とドロップダウン・メニューからタイプ、または[優先度が存在]とドロップダウン・メニューから優先度を選択すると、選択した要件タイプまたは優先度に関連付けられた変更のみをコミットします。
 - [変更セットの不具合参照]は、コミットごとに既存の不具合を必ず参照するオプションです。
 - [重大度]では、選択した重大度を持つ変更のみがコミットの対象になります。
 - [次のメモをシステム メッセージに追加]は、メモによってコミットがブロックされている場合、カスタム・システム・メッセージをユーザに送信します。
3. [OK]をクリックします。

注: 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」(15ページ)を参照してください。

ブランチでコミットを実施する条件となるロック・ポリシーを指定するには、次の手順を実行します。

注: この機能は、Git リポジトリでは使用できません。設定しても無視されます。

1. [ロックポリシー] タブでは、ブランチに関連付けられているロック・ポリシーを管理します。
2. 次のオプションを選択します。
 - [次の例外を除いて、コミットを許可しない]を選択すると、次の例外を除き、所定のブランチを対象としたコミットをすべて拒否します。
 - ブランチへのコミットを許可するユーザのリストを指定します。コミットを許可されている SCM ユーザのユーザ名を適用することにより、ブランチがロックされます。
 - [不具合の追加]をクリックして不具合 ID を指定することにより、変更のコミットを許可する不具合リストを指定します。[不具合の削除]をクリックすると、テーブルから不具合が削除されます。
 - [次のメモをシステム メッセージに追加]は、メモによってコミットがブロックされている場合、カスタム・システム・メッセージをユーザに送信します。
3. [OK]をクリックします。

コミット・パターンの設定

変更セットと要件/不具合間のトレーサビリティを検出および維持する作業には、コミット・メッセージを使用します。このメッセージは、ユーザ(開発者)が変更を SCM システムにコミットする際に作成されます。ALI では、ユーザのさまざまなコミット・パターンに対応したカスタマイズが可能です。このメッセージは、変更セットのコミット時に、コミット・メッセージに記載されている要件または不具合 ID を転送することを目的とします。

コミット・パターンの設定画面は、[コミット パターン]ビューで新しいリポジトリを設定中に表示されます。また、既存のリポジトリで[コミット パターン]タブをクリックしてもアクセスできます。

パターンは、「基本」モードまたは「詳細」モードで定義できます。

基本:

1. 不具合または要件で使用するキーワードを追加するには、[追加]ボタンをクリックし、値を入力して[OK]をクリックします。指定したキーワードは、不具合または要件に関連するコミットとしてマークします。

キーワードを削除するには、キーワードを選択して[削除]ボタンをクリックします。

2. ID 番号を入力してプレフィクス ID をパターンに追加します。ラジオ・ボタンをクリックすると、オプションになります。
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** – デフォルトで有効になります。Taskop プラグインを ALM に統合すると、Tasktop はデフォルトのコミット・メッセージを、ALI が認識するパターンに応じて生成します。
- **大文字と小文字が区別されるコミット メッセージ** – デフォルトでは無効です。コミット・メッセージで大文字と小文字を区別する場合、有効にします。
- **複数の不具合または要件の区切りマークアップ** – 1つのコミット・メッセージに不具合や要件が複数含まれている場合、このマークアップは区切り文字として認識されます。
- **コミット メッセージ内のキーワードの位置** – キーワードは、コミット・メッセージの先頭のみで識別されるか、任意の位置で識別されます。
- **ユーザコミット メッセージの区切り文字** – ユーザ・コミット・メッセージと、不具合または要件キーワードを区切る区切り文字。

■ コミット・メッセージのオプションの使用例:

fixing defect #100, #101:fix caching and enhance functionality

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 = ユーザ・コミット・メッセージ

4. タブから出て設定をコミットします。

詳細:

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。
 - a. [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。
3. このフィールドにカスタム・コミット・メッセージを入力すると、メッセージをテストできます(オプション)。
4. 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。

次に、詳細設定の基本的な例をいくつか示します。

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. 次に示すデフォルトの 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"
```

変更の検出の設定

ALI は、事前設定した SCM リポジトリで行った変更を検出し、変更セットと関連ファイルの情報を ALM サーバにロードして、ロードしたコード変更と作業項目 (要件, 不具合) 間のトレーサビリティを自動生成します。ALI は、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ます。

ポーリング機構を有効にすると、ALI は所定の SCM リポジトリで新しい変更が発生していないかを定期的にチェックし、検出した場合、変更をロードします。ポーリング機構は、SCM システムで設定を行わなくても使用でき、SCM エージェントを実行する必要もありません。ポーリング機構は、[変更の検出] タブの [SCM から変更を読み取る] オプションで設定します。

プッシュ機構は、SCM エージェントを SCM システムに接続します。エージェントは SCM システムでリスンします。事前構成のリポジトリとブランチに対して変更がコミットされると、エージェントはポリシーをチェックして、コミットが許可される場合には変更セットを ALM サーバにプッシュします。プッシュ機構は、[変更の検出] タブの [SCM エージェントから送信された変更を受信する] オプションで設定します。

変更検出の設定画面は、[変更の検出] ビューで新しいリポジトリを設定中に表示されます。また、既存のリポジトリで [変更の検出] タブをクリックしてもアクセスできます。

次の2つの検出オプションがあります。

SCM から変更を読み取る

- 読み取り間隔を秒単位で設定します。また、[再スケジュール] ボタンをクリックして日時を選択することで、実行スケジュールを設定します。
- [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。

注: 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。

SCM エージェントから送信された変更を受信する

- SCM エージェントが設定済みの場合、変更を直ちに処理します。詳細については、「サポート対象の SCM システム」(14 ページ) を参照してください。

第3章

ビルド・システムの統合

ビルドとは、ソフトウェア開発の主要な成果物です。Application Lifecycle Intelligence では、トレーサビリティ機能として、ビルドとその他の ALM エンティティとの関係を追跡します。

このモデルは、次の情報を提供します。

- ビルド・アーティファクト(ビルド・プロセスで生成されたバイナリ・データ)。
- ビルドのコンテンツ(コンポーネント、パッケージ、ファイルなど)。
- ビルドに新しく追加された変更(前回ビルド以降に新しく追加された変更であり、ビルドを構成する変更セット)。
- 実装された作業項目(ビルドで実装または変更された要件、不具合、ストーリー)。
- ビルドとリリースの関係
- ビルドとテストの関係

ビルド・システムとの統合とは、コード変更がソフトウェア成果物とビルド・アーティファクトにどのような影響を与えるかを測定するツールです。ビルドのレポートでは、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(テスト結果が5%低下した原因となったコミットなど)で示します。

本章の内容

サポート対象のビルド・システム	26
ビルド・サーバの追加	28
ビルド構成の追加	29
ビルド構成にある SCM 構成の再利用	29
ビルド構成の不具合フィルタの設定	30
ALI プロジェクト・リストのカスタマイズ	30
コード問題の追跡	31
コード問題の追跡の設定	31
ビルド・サーバの検出の設定	32

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。

- Jenkins - 長期サポート・リリース - 1.424.6, 1.447.1
- Hudson - 最新製品版 - 2.1.2, 2.2.0

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、ビルドの情報とメトリックを自動抽出し、ALM モデルに読み込みます。

ALI は、次の開発メトリクス・ツールをサポートします。

- ユニット・テスト
 - 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 0.32 でテスト済み
 - NUnit - Hudson 0.10, Jenkins 0.14, NUnit framework 2.5.10 バージョンでテスト済み
- コード・カバレッジ・アナリシス
 - 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

Hudson/Jenkins のインストール

Hudson/Jenkins ビルド・サーバを使用している場合は、次のコンポーネントを追加インストールする必要があります。

- **Perforce, TFS, Git:** SCM の使用をサポートする Hudson/Jenkins プラグイン。<http://hudson-ci.org> または <http://jenkins-ci.org> からダウンロード可能です。このプラグインが必要になるのは Perforce, TFS, Git を使用する場合のみであり、Hudson/Jenkins プラグインのパブリック・リポジトリで提供されています。SVN と CVS はデフォルトでサポートされます。
- HP ALI Hudson/Jenkins プラグイン(ALI バンドル - agents\build-integration\hudson\にある ali-hudson-plugin.hpi)。このプラグインは、Hudson と Jenkins の両方のシステムをサポートします。
- **Perforce, TFS, Git:** 各 SCM 用に追加の HP ALI Hudson/Jenkins プラグインを ALI バンドルから取得する必要があります。

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 プラグインのインストール](#)」(26ページ)を参照してください。

HP ALI Hudson/Jenkins プラグインのインストール

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 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、ベース・プラグインである `ali-hudson-plugin.hpi` のほかに、`ali-hudson-tfs-plugin.hpi` もインストールする必要があります。

注意: `ali-hudson-tfs-plugin.hpi` を使用するには、Hudson/Jenkins TFS プラグインを Hudson/Jenkins からダウンロードして先にインストールしておく必要があります。詳細については、「[Hudson/Jenkins のインストール](#)」(26ページ)を参照してください。

- **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、ベース・プラグインである `ali-hudson-plugin.hpi` のほかに、`ali-hudson-perforce-plugin.hpi` もインストールする必要があります。

注意: `ali-hudson-perforce-plugin.hpi` を使用するには、Hudson/Jenkins Perforce プラグインを Hudson/Jenkins からダウンロードして先にインストールしておく必要があります。詳細については、「[Hudson/Jenkins のインストール](#)」(26ページ)を参照してください。

- **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、ベース・プラグインである `ali-hudson-plugin.hpi` のほかに、`ali-hudson-git-plugin.hpi` もインストールする必要があります。

注意: `ali-hudson-git-plugin.hpi` を使用するには、Hudson/Jenkins Git プラグインを Hudson/Jenkins からダウンロードして先にインストールしておく必要があります。詳細については、「[Hudson/Jenkins のインストール](#)」(26ページ)を参照してください。

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。

注:

- CVS または TFS SCM を使用する場合は、「[リポジトリの追加と編集](#)」(16ページ)を参照して、ALM で SCM の追加設定を行ってください。
- インストールしたプラグインは、[プラグインマネージャ]の[インストール済み]タブに表示されません。
- ALI プラグインが正しくインストールされていると、Hudson/Jenkins の左側メニューに[ALI 統合]リンクが表示されます。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 プラグインには、グローバルな[システムの設定]から行うグローバル設定と、ジョブごとに行うジョブ・スコープ設定があります。詳細については、Hudson/Jenkins サーバの ALI 統合プラグインにアクセスし、各プロパティ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] セクションを選択します。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: このオプションを有効にすると、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 - このオプションを選択すると、ビルドの開始後と終了時に、ビルド情報が HP ALM サーバに即時送信されます。

注: グローバル設定で指定したプロパティはすべて、後でジョブごとに上書きできます。特定のビルド・サーバでプッシュ機構を使用する場合、このプロパティが必要になりますが、プッシュ機構を使用し、さらにジョブごとに ALM プロパティを指定する場合、このオプションを有効にする必要があります。

ジョブごとの設定を行うには、次の手順を実行します。

目的の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ます。このジョブで ALI 統合を有効にするには、このオプションを選択する必要があります。

- テストソースのマッピングパターン - テスト結果から、テスト・ソースの場所を特定します。詳細と例については、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 ALI 設定プロパティを上書きします。指定したビルド・ジョブの HP ALM 内にあるビルド情報を更新します。
- NCover - .NET 設定に関する 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 Basedir は、ルート・ワークスペースです。
- Force.com – Force.com 統合に関する設定です。詳細については、「Force.com の統合」(33 ページ)を参照してください。

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 にログインします。ALM サイドバーで、[管理]にある[ビルド サーバ]を選択します。[サーバ]ページが開きます。
2. [新規サーバ]をクリックし、ドロップダウン・リストで[ビルド サーバの構成]を選択してから[OK]をクリックします。
3. [新規ビルド サーバ]ダイアログ・ボックスが開いたら、サーバの名前と場所、サーバの説明、ユーザ名とパスワードを必要に応じて入力します。

注: サーバの場所には、完全パス(http://xx.xx.xxx.xxx:xxxx など)を入力します。不明な場合は、システム管理者に確認してください。

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。
 - [ビルド サーバから変更を読み取る]を選択し、実行間隔を指定したり、実行回数をスケジュールします。

- 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・サーバの検出」を参照してください。
5. [送信]をクリックすると、サーバが追加されます。

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]タブを開き、[新規ビルド構成]をクリックします。
2. 表示されたリストでビルド構成を選択して[OK]をクリックすると、詳細ウィンドウが開きます。
注:このリストには、管理対象サーバの未使用構成が表示されます。
注意:このリストには、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が表示されます。
3. [リリース]ドロップダウン・メニューをクリックし、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・メニューをクリックし、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:
 - 構成を有効化 -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せん。
 - 構成をデフォルトに設定 -デフォルトのビルド構成に従って、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

既存のビルド構成は ALM によって保存されており、新しく作成した構成に SCM ブランチとリポジトリを適用することが可能です。

ビルド・システム上にあるビルド構成では、一般的に、SCM 関連情報 (SCM リポジトリ、ブランチ、資格情報) がすでに指定されています。ALI には、ビルド・システムに保存されているこのような情報を再利用する機能があり、ALI 内で別の SCM システムを構成する作業を簡素化できます。

特定のビルド構成にある既存の SCM 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]ページでサーバをクリックします。
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すると、[ビルド構成詳細]ポップアップ・ウィンドウが開きます。
4. [SCM]タブを選択すると、使用可能なリポジトリとブランチが表示されます。
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。

リポジトリまたはブランチの場所の横にステータス・メッセージが表示され、リポジトリまたはブランチが定義済みかどうかを示されます。

注:[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。

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ことにより、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、リリースに関して表示される未解決の不具合と新規不具合の入力セットの数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。

特定の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]ページでサーバをクリックします。
2. [ビルド構成]タブを選択します。
3. 構成名をクリックすると、[ビルド構成詳細]ウィンドウが開きます。
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、内容を確認してから[はい]をクリックします。

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を定義します。
ALI 終了ステータス	[ステータス]フィールドの値のサブセットを定義します。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。[終了ビルド]フィールドに値を設定できるのは、不具合が解決済みとみなされた場合のみです。
ALI コード問題定義	ALI がコード問題として不具合を識別するときに使用する値を定義します。 コード問題の詳細については、「コード問題の追跡」(31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]フィールドで指定できる値を定義します。
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を定義します。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]タブで表示されます。重大度のプロジェクト・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

ALI には、**コード問題**を追跡する機能があります。コード変更の結果、新しい不具合が作成されると(コード分析またはテスト・プロセスの一部など)、ALI は、ユーザが設定した条件に基づいて、不具合をコード問題として識別します。コード問題の追跡機能を使用すると、コード変更に関連付けられている要件、不具合、ビルドから、コード問題の不具合にリンクを追加できます。

コード問題の不具合を追跡することにより、次の操作を実行できます。

- 要件に関連するコード変更が原因で発生する不具合を特定
- 不具合に関連するコード変更が原因で新しく発生した不具合を特定
- ビルドで検出される不具合を特定

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。

コード問題の追跡の設定方法の詳細については、「[コード問題の追跡の設定](#)」(31ページ)を参照してください。

コード問題の追跡の設定

本項では、ALM プロジェクトでコード問題の追跡を設定する方法について説明します。コード問題の詳細については、「[コード問題の追跡](#)」(31ページ)を参照してください。

次の条件を満たす不具合は、要件、不具合、ビルドのコード問題と認識されます。

- 不具合が終了していないこと。つまり、不具合の[ステータス]フィールドは、[ALI 終了ステータス]のプロジェクト・リストで定義されている値のいずれにも一致しません。
- 不具合の[優先度]フィールドが、[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にも一致しないこと。
- コード問題の追跡用に、ユーザ定義のフィールドを定義していること(次を参照)。このフィールドを不具合で設定し、[ALI コード問題定義]プロジェクト・リストのサブ項目として定義されている値にする必要があります。
- 次のいずれかに該当する必要があります。
 - 要件または不具合の場合: 不具合または要件に関連するコード変更が原因で発生した不具合であること。
 - ビルドの場合: ビルド内で検出された不具合であること。

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. [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定義します。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。
2. [プロジェクトリスト]ページで、ユーザ定義のフィールドの[名前]フィールドの値を、[ALI コード問題定義]プロジェクト・リスト内にある唯一のリスト項目として追加します(例: BG_USER_01)。

3. 新しいユーザ定義フィールドの値を、[ALI コード問題定義]プロジェクト・リストのサブ項目として追加します。

ビルド・サーバの検出の設定

Application Lifecycle Intelligence は、ビルド・サーバに新しく追加されたビルドを検出し、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・カバレッジをロードします。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 との統合により、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ます。

変更検出の設定画面は、変更の検出ビューで新しいビルド・サーバを設定中に表示されます。また、既存のビルド・サーバで[変更の検出]タブをクリックしてもアクセスできます。

ポーリング:

ポーリング機構では、ALI はビルド・サーバに新しいビルドが追加されたかどうかを定期的にチェックし、ビルドが検出されると、ALM サーバ上に存在しない場合はロードします。

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]を選択します。実行のスケジュール設定、新規ビルドの有無をチェックする時間帯の指定、変更のロードには、同期コマンドを使用します。

プッシュ:

プッシュ機構は、ポーリング機構とは逆の方法で動作します。ビルド・サーバの ALI 統合プラグインで[Update build information in HP ALM]オプションを選択すると(詳細については「[HP ALI Hudson/Jenkins プラグインの設定](#)」(27ページ)を参照してください)、このプラグインはビルド・サーバでリスンします。新しいビルドが開始すると、そのビルドの情報が ALM サーバにプッシュされます。ビルドが完了すると、ALM サーバ上にあるステータスなどのビルド情報が更新されます。

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。

第4章

Force.com の統合

この統合により、Force.com プラットフォームで開発を行うチームは、ALI の標準的な開発機能をすべて利用可能になります。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。

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

本章の内容

前提条件.....	34
プロジェクトのデプロイメント、テスト、レポート作成.....	34
Hudson/Jenkins に関する Force.com の構成.....	36

前提条件

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

- Force.com ソース・コードを SCM システムに格納します。HP ALI でサポートされる SCM システムの完全なリストは、「サポート対象のSCMシステム」(14ページ)を参照してください。
- Hudson または Jenkins
 - SCM の使用をサポートする Hudson/Jenkins プラグイン(SVN と CVS はデフォルトでサポート)。
 - HP ALI Hudson プラグイン(`ali-bundle.zip/build/Hudson/ali-hudson-plugin.hpi`)。詳細については、「HP ALI Hudson/Jenkins プラグインのインストール」(26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ために、ビルド管理サーバを構成します。
- Apache Ant(<http://ant.apache.org/>(英語 サイト) からダウンロード)。
- HP force-deploy-task が、ALI アーカイブに格納されている(`/tools/force-deploy-task/force-deploy-task-bundle.zip` ファイルの内容が `ant_install_dir/lib` に展開)。

プロジェクトのデプロイメント, テスト, レポート作成

ソース・コードのデプロイメント, テスト, レポートの作成には、専用の Ant タスクである HP force-deploy-task を使用します。次の例のように、「build.xml」という名前の Ant ビルド・スクリプトを Force.com プロジェクトのルート・フォルダに作成する必要があります(存在しない場合)。

1. 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、レポートにプロジェクト全体のコード・カバレッジを含める必要があります。

```
<project name="Sample usage of force-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>
```

2. 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 は完全なコード・カバレッジを実現しません。

```
<project name="Sample usage of force-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 -->
    <batchtest>
      <fileset dir="src/classes">
        <include name="*Test.cls"/>
      </fileset>
    </batchtest>
  </target>
</project>
```

HP force-deploy-task の説明 (sfdeploy で定義される例) :

- username 属性 – Force.com 環境にログインする際のログイン名
- password 属性 – Force.com 環境にログインする際のパスワード
- severurl 属性 – Force.com 環境へのログイン・ページの URL
- deployRoot 属性 – ソース・コード・ディレクトリへのパス(フォルダ・クラス、トリガなどの格納場所)
- runalltest 属性 – (true/false) true の場合、すべてのテストを開始してプロジェクトのコード・カバレッジを報告します。false の場合、batchtest 要素で指定されたテストのみが開始されます(コード・カバレッジは実現されません)。
- reportDir 属性 – すべてのレポートが格納されるディレクトリ
- batchtest 要素 – 開始するテストを指定する要素 (runalltests=true の場合のみ有効)
 - fileset 要素 – 実行するテストのファイル・セット
 - dir 属性 – テストが格納されているディレクトリ
 - include 要素
 - name 属性 – 実行するテストのクラス名のパターン

Hudson/Jenkins に関する 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、開始するターゲットを指定します (deployAndTestAndReport を参照してください)。
3. ビルド後の処理セクションで、次の図に示す設定を行います。

The screenshot shows the configuration interface for the 'force-deploy-task' in Hudson/Jenkins. It features two main sections: 'Publish JUnit test result report' and 'ALI Integration'. The 'Publish JUnit test result report' section has a checked checkbox and a text input field for 'Test report XMLs' containing 'test-report-xml/**/TEST-*.xml'. Below this is a help link for 'Fileset 'includes'' and a checkbox for 'Retain long standard output/error'. The 'ALI Integration' section also has a checked checkbox and a text input field for 'Test sources mapping pattern' containing '.*//src/classes//cls'. Below this is a help link for the 'Regex pattern for locating test sources' and a '高度な設定...' (Advanced Settings) button.

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 (force-deploy-task の reportDir 属性) で置換します。
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、Report 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:

Force.com	
Report directory	<input type="text" value="test-report-xml"/> <small>Relative path to directory where reports are generated by HP force-deploy-task Ant task (attribute reportDir). Consider workspace root as basedir.</small>
Project root	<input type="text" value="src"/> <small>Relative path to directory with source codes (consider workspace root as basedir). Default value is src.</small>

第5章

SCM の変更とトレーサビリティのモニタ

リポジトリ、ブランチ、実施ルール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の[変更セット]に表示します。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。

プロジェクトの効率的な管理ではレポート機能が重要な役割を果たすので、ALI では、コード変更エンティティを網羅したレポートを追加することによって HP ALM プラットフォーム・レポートを拡張しています。

プロジェクトのレポートとグラフ(サマリ, 進行状況, トレンド) では、チームと開発者の開発アクティビティをレビューおよび比較できます。また、変更影響レポートでは、不具合と要件への対応として実装された変更が与える影響を確認できます。

本章の内容

コード変更テーブルの表示.....	38
コード変更の詳細.....	38
変更影響レポートの表示.....	39
プロジェクト・レポートの作成.....	39
グラフの生成.....	39

コード変更テーブルの表示

コード変更モジュールは、ALM サイドバーの[開発]の下に表示されます。

初期設定では、所定の期間中、選択したリリースで発生したすべてのコード変更が表示されます。表示内容は、テーブルの一番上にあるドロップダウンメニューで変更およびフィルタ処理できます。

日時	開発者	ファイル	行	メッセージ	ステータス メッセージ
2011/12/07 14:46:44	admin	1	1	不具合 #3: サイクル2	
2011/12/21 14:13:22	admin	1	17		カラム 0 に がありました、'修正中不具合'、'不具合
2012/08/23 13:37:18	admin	1	216	不具合 #1	カラム 1 に '不具合 #1' がありました、'修正中不具合
2012/08/23 13:47:31	admin	1	238	不具合 #3: サイクル2	
2012/08/23 13:48:35	admin	1	93	不具合 #3: サイクル2	
2012/08/23 14:00:53	admin	1	1		カラム 0 に がありました、'修正中不具合'、'不具合
2012/08/24 15:21:38	admin	1	1	要件 #2: テスト	
2012/08/24 15:35:59	admin	1	1	不具合 #2: 修正しました	
2012/08/24 15:48:45	admin	1	1	不具合 #3: 要件 #21	
2012/09/10 16:00:32	admin	1	162	不具合 #3:	カラム 9 に がありました、TEXT が必要です
2012/09/10 16:08:00	admin	1	722	不具合 #3: 要件 #21	

テーブルには、変更日時、変更を行った開発者、変更されたファイル数と行数、開発者によるコミット・メッセージが表示されます。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、エラーまたは警告がステータス・メッセージで表示されます。[表示]メニューにあるフィルタを有効にすると、カラムを変更またはフィルタ処理できます。


変更セットに関連の要件が存在する場合、黒で表示されます。

変更セットに関連の不具合が存在する場合、緑で表示されます。

変更セットに関連の不具合と要件のいずれも存在しない場合、赤で表示されます。

リリースと時刻のほかに、要件または不具合のいずれかのみに関連するデータや未割り当ての変更も表示可能です。


注: 未割り当ての変更とは、要件と不具合のいずれにも関連しないコミットです。計画外の変更を担当するチームのマネージャが参照すると、役立つ情報です。

フィルタ・ボタン  をクリックすると、詳細フィルタを適用できます。

[表示]メニューの下に[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、変更セットに関連するファイル、要件、不具合がタブで表示されます。このパネルには、コミット・メッセージまたはステータス・メッセージも表示されます。

コード変更の詳細

[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]をダブルクリックするか、[コード変更の詳細]ボタン  をクリックすると、新しいウィンドウが開いてコード変更が表示されます。

- 基本的な属性に加えて関連ファイルとステータスのリストが表示され、ファイルの旧バージョンとの相違点を比較する機能もあります。
- 変更セットが要件または不具合と関連付けられている場合、左側メニューの[要件]または[不具合]項目をクリックすると、リンクが表示されます。

変更影響レポートの表示

ALI のレポートでは、変更が与える影響を、ファイルとモジュールだけでなく、開発者に割り当てられた作業項目(要件および不具合)についても確認できます。コード・カバレッジ、テスト結果、変更の規模、影響を受ける作業項目をコミットした担当者などの情報が表示されます。

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。

1. 表示期間を選択し、[レポートの表示]をクリックします。
 - 新しいウィンドウが開き、指定した表示期間でレポートが表示されます。
 - 選択した期間中に要件と不具合で発生した変更内容が表示されます。
 - レポート右上にあるグラフでは、プロジェクトの工数の分布がLOC(Kilo lines of Code)単位で表示されます。
2. [未割り当ての変更を表示]をクリックすると、未割り当ての変更が表示されます。
3. [このレポートを印刷]をクリックすると、印刷デバイスにレポートが送信されます。
4. [このレポートを電子メールで送信]をクリックすると、電子メール・クライアントが開き、レポートのURLが指定アドレスに送信されます。

プロジェクト・レポートの作成

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 ユーザーズ・ガイド』を参照してください。

グラフの生成

ALI 拡張機能を使用すると、コード変更のグラフを作成できます。

コード変更について、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	<p>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。</p> <p>X 軸で示す時間間隔と、ALM がデータを分類する際の基準となる要件情報を指定します。表示内容は、開発者ごとのコード変更の数、変更したファイル数の合計、変更した行数を選択できます。</p>
コード変更 - サマリグラフ	<p>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。</p> <p>X 軸に表示するデータのタイプと、ALM でデータを分類する際の基準となるコード変更情報を指定します。</p>
コード変更 - トレンドグラフ	<p>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。</p> <p>変更数を表示する対象となるフィールドと、データの表示期間を指定します。</p>

グラフを作成するには、次のいずれかを実行します。

1. グラフ・ウィザードを起動します。

グラフ・ウィザードでは、グラフの作成と設定に必要な作業を手順に沿って実行できます。グラフ・ウィザードは、アナリシス・ビュー・モジュールから起動する方法と、コード変更モジュールでの作業中に起動する方法があります。

a. アナリシス・ビュー・モジュールから:

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。

b. コード変更モジュールから:

[アナリシス] > [グラフ] > [グラフ ウィザード]を選択します。

c. [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「コード変更」に変更します。グラフ・タイプが3つ表示されるので、適切なグラフを選択してウィザードの指示に従います。

2. 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、次の定義済みグラフのいずれかを選択します。

- コード変更 - サマリグラフ - [開発者]によりグループ化
- コード変更 - サマリグラフ - [ファイル]の合計, [開発者]によりグループ化
- コード変更 - サマリグラフ - [行]の合計, [開発者]によりグループ化
- コード変更 - 進捗グラフ - [開発者]によりグループ化
- コード変更 - トレンド グラフ - [開発者]によりグループ化

3. 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。

4. アナリシス・ビュー・モジュールの[アナリシス メニュー]タブで, 定義済みのグラフを作成します。

コード変更のメイン アナリシス メニューフォルダには, 次の定義済みグラフが表示されます。

- コード変更 - サマリグラフ - [開発者]によりグループ化
- コード変更 - サマリグラフ - [ファイル]の合計, [開発者]によりグループ化
- コード変更 - サマリグラフ - [行]の合計, [開発者]によりグループ化
- コード変更 - 進捗グラフ - [開発者]によりグループ化
- コード変更 - トレンド グラフ - [開発者]によりグループ化

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6章

開発アクティビティのモニタ

ビルド・サーバのセットアップと設定が完了すると、Application Lifecycle Intelligence(ALI) では、リリース、不具合、要件の各 ALM モジュールで[開発アクティビティ]タブが表示されます。このタブでは、プロジェクト内で起こっていることについての情報が、ビルドおよび開発者の視点から表示されます。

タブ内のリンクをクリックすると、選択したビルドの詳細情報と統計が表示されます。

タブの一番下には、ビルドの変更情報が開発者別に表形式で表示されます。

[変更]をクリックすると、その他の構成済みビルドのアクティビティが表示されます。

本章の内容

開発アクティビティ-リリース.....	43
開発アクティビティ-不具合.....	43
開発アクティビティ-要件.....	43
[不具合詳細]での開発アクティビティのモニタ.....	43

開発 アクティビティ - リリース

リリース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タブに表示されます。タブをクリックすると、アクティビティが表示されます。

このタブでは、ビルド・ステータスと成功率、ビルド回数、修正回数、コードベース統計(単位はKLOC(Kilo Lines of Code))などが表示され、ビルドの進行状況を的確に把握できます。タブのリンクをクリックすると、詳細な内容が表示されます。

[**変更**]をクリックすると、デフォルト以外のビルド構成を選択できます。

[**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**」(29ページ)を参照してください。

タブの一番下には、リリースに担当している開発者とそれぞれの分担レベルが表形式で表示されます。

開発 アクティビティ - 不具合

不具合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タブに表示されます。タブをクリックすると、アクティビティが表示されます。

このタブでは、[開発メトリクス]でユニット・テストとカバレッジの情報、[不具合の実装]で要件の情報が表示されます。[ユニット テスト]リンクをクリックすると[**不具合 ユニット テスト レポート**]が表示され、[カバレッジ]リンクをクリックすると[**不具合 カバレッジ レポート**]が表示されます。

[**不具合の実装**]の下にあるリンクをクリックすると、ビルドの詳細または各コード変更が表示されます。

タブの一番下には、不具合に関連するコード変更とアクティブな開発者が表形式で表示されます。

コード変更のメッセージ・リンクをクリックすると、[**不具合詳細**]ページが開きます。

カラム見出しをクリックすると、テーブルをソートまたはフィルタ処理できます。

開発 アクティビティ - 要件

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タブに表示されます。タブをクリックすると、アクティビティが表示されます。

タブには、要件について、開発メトリクスと要件の実装が表示されます。[ユニット テスト]リンクをクリックすると[**要件 ユニット テスト レポート**]が表示され、[カバレッジ]リンクをクリックすると[**要件 カバレッジ レポート**]が表示されます。

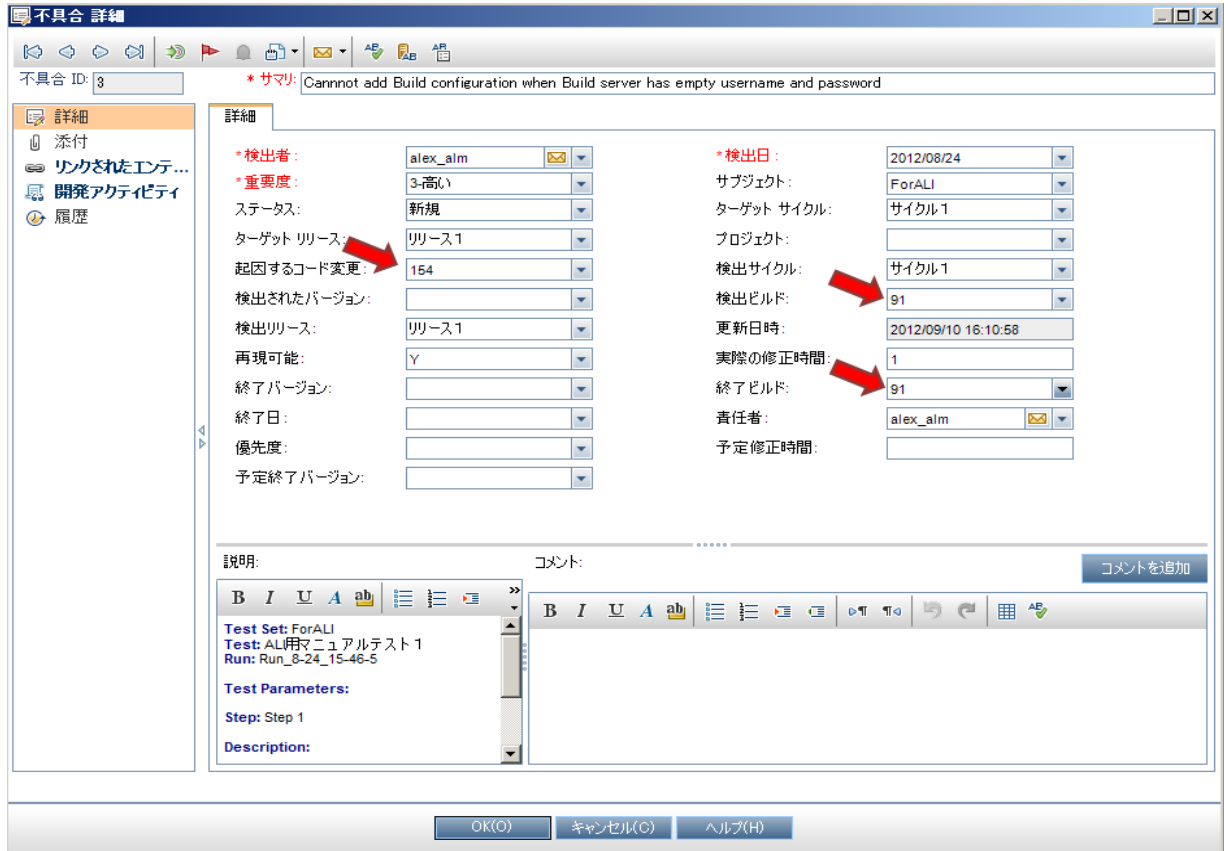
タブの一番下には、要件に関連するコード変更とアクティブな開発者が表形式で表示されます。

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**]ページが開きます。

カラム見出しをクリックすると、テーブルをソートまたはフィルタ処理できます。

[不具合詳細]での開発 アクティビティのモニタ

Application Lifecycle Intelligence の[不具合詳細]ページには、新しい値が3つ追加されています。



検出ビルド – 所定の不具合が検出されたビルド。

終了ビルド – 所定の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セット。

HP ALM プラットフォーム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表示されます。ツールの設定を行っていない場合は、ドロップダウン・リストから手動で入力します。

これらのフィールドのデータは、[ビルド 詳細] ページの[不具合] タブに入力され、[ビルド サマリレポート] で表示されます。詳細については、「[ビルドの詳細](#)」(48ページ)および「[ビルド・レポート](#)」(50ページ)を参照してください。

第7章

ビルドとテストのトレーサビリティ

ビルドとテストを関連付け、相互の関係を追跡します。

本章の内容

ビルドとテストのトレーサビリティの表示.....	46
--------------------------	----

ビルドとテストのトレーサビリティの表示

QA エンジニアまたはマネージャは、ビルドとテストのトレーサビリティ機能を使用することにより、ビルド、テストセットとテスト実行の関連付けを作成し、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、テストセットのテストに使用するビルドを定義します。したがって、すべてのテストが、選択したビルドでテストされることになります。テストとビルドをこのような方法で関連付けることにより、ビルドの品質をQAの視点からモニタすることができます。具体的には、特定のビルドでテストを計画し、ビルドで行ったテストの結果をレビューします。QA エンジニアが実行するテストセットとテスト実行の結果は、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します。

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

ビルド・アクティビティのモニタ

ビルド・システムのセットアップが完了すると、Application Lifecycle Intelligence 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、エラーが報告された場合には適切に対応できます。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。

さらに、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することもできます。詳細については、「ビルド・レポート」(50ページ)および「ビルド・グラフの作成」(51ページ)を参照してください。

本章の内容

ビルドの表示.....	48
ビルドの詳細.....	48
ビルド・レポート.....	50
ビルド・グラフの作成.....	51

ビルドの表示

ALI 拡張機能が有効になっているプロジェクトにログインすると、ビルド・モジュールを使用できます。


デフォルトのビューでは、選択したリリースと所定の期間に含まれるビルドがすべて表示されます。表示内容は、テーブルの一番上にあるドロップダウン・メニューで変更およびフィルタ処理できます。

日時	番号	ビルド ステータス	ラベル	QA ステータス	ユニットテスト	キャッシュ	経過時間	ビルドレポート	ビルド
2012/08/23 14:01:00	59						36 秒	レポートの表示	詳細の表示
2012/08/23 14:02:55	60				100%		24 秒	レポートの表示	詳細の表示
2012/08/23 14:38:12	61				100%		29 秒	レポートの表示	詳細の表示
2012/08/23 14:41:32	62				100%		24 秒	レポートの表示	詳細の表示
2012/08/23 14:47:38	63				100%		24 秒	レポートの表示	詳細の表示
2012/08/23 16:31:42	64				100%		47 秒	レポートの表示	詳細の表示
2012/08/24 15:25:43	65				75%		1分 16 秒	レポートの表示	詳細の表示
2012/08/24 15:36:08	66				100%		25 秒	レポートの表示	詳細の表示
2012/08/24 15:48:53	67				100%		25 秒	レポートの表示	詳細の表示
2012/08/27 12:17:01	68				100%		1分 14 秒	レポートの表示	詳細の表示
2012/09/10 16:00:48	69				75%		1分 25 秒	レポートの表示	詳細の表示
2012/09/10 16:08:23	70			テスト不可	100%		23 秒	レポートの表示	詳細の表示

このテーブルでは、ビルドの日時など、ビルドについて記録されている情報が表示されます。[表示]メニューにあるフィルタを有効にすると、カラムを変更またはフィルタ処理できます。

[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。

- 緑色 = 正常。ビルドが正常であることを示します。
- 黄色 = 警告。ビルドに問題があることを示します。
- 赤 = 失敗。ビルドが失敗したことを示します。

フィルタ・ボタン  をクリックすると、詳細フィルタを適用できます。


[ビュー]メニューで[情報パネル]を選択すると、テーブルの下にパネルが表示され、ビルドの要件、不具合、コード変更、テスト、配布、ステータス・メッセージがタブで表示されます。

各ビルドの詳細表示については、「ビルドの詳細」(48ページ)を参照してください。

ビルドの詳細

[ビルド詳細]ダイアログ・ボックスでは、選択したビルドの詳細情報が表示されます。



アクセス方法	ビルド・モジュールでビルドをダブルクリックするか、ビルドを選択して[ビルド詳細]ボタン  をクリックします。
参照情報	「ビルド・アクティビティのモニタ」(47ページ)

次の UI 要素が表示されます。

UI 要素	説明
詳細	ビルドに関するフィールドが表示されます。 [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作成されます。詳細については、「ビルド・レポート」(50ページ)を参照してください。 [ビルドシステム]をクリックすると、ビルド・システム全体のビューが作成されます。 コード問題のリンクをクリックすると、そのコード問題に関する不具合の詳細な内容が表示されます。詳細については、「コード問題の追跡」(31ページ)を参照してください。
[要件]タブ	ビルドに関連付けられている要件が表示されます。 カラム見出しをクリックすると、リストをソートできます。 要件をクリックすると、[要件の詳細]ページが開きます。

<p>[不具合] タブ</p>	<p>ビルドに関連付けられている不具合が表示されます。</p> <p>検出された不具合、解決済みの不具合、実行された修正が表示されます。緑色の星型アイコンが付いているタブは、グリッドに新しいエントリが追加されたことを示します。</p> <p>カラム見出しをクリックすると、リストをソートできます。</p> <p>不具合サマリのリンクをクリックすると、[不具合詳細]ダイアログ・ボックスが開きます。</p>
<p>[コード変更] タブ</p>	<p>ビルドに関連付けられているコード変更が表示されます。</p> <p>カラム見出しをクリックすると、リストをソートできます。</p> <p>コード変更メッセージをクリックすると、[コード変更の詳細]ダイアログ・ボックスが開きます。</p>
<p>[アクティブな開発者] タブ</p>	<p>ビルドに関連付けられている開発者と、分担のレベルが表示されます。</p> <p>カラム見出しをクリックすると、リストをソートできます。</p>
<p>[テスト] タブ</p>	<p>ビルドに関連付けられているテスト実行の詳細が表示されます。緑色の星型アイコンが付いたタブは、入力済みであることを示します。</p> <p>すべてのテスト実行のステータスを参照できます。テスト数をクリックすると、テスト・インスタンスがすべて一覧表示されます。</p> <p>サマリ・ボックスの最後の行には、テスト実行のステータスと回数が、ステータスごとの内訳で表示されます。</p>

ビルド・レポート

[ビルド - サマリレポート]では、指定の期間とビルド・カテゴリに基づいてビルド情報をサマリで表示します。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。

1. Application Lifecycle Intelligence の開発モジュールからビルドを選択します。
2. [レポートの表示]ボタンをクリックすると、新しいウィンドウが開いてサマリ・レポートが表示されます。

[詳細なビルドレポート]では、要件、通知された不具合、新しい不具合、解決済みの不具合など、選択した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]リンクをクリックすると、未割り当ての変更が表示されます。

[ビルド変更レポート]では、選択したビルドが影響を与える要件と、選択したビルドによって変更、終了、作成された不具合の情報がサマリで表示されます。

[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. **開発** モジュールの[ビルド]ページでビルドを選択します。これにより, [ビルド詳細]ページが開きます。
2. [レポートの表示] ボタンをクリックすると, 新しいウィンドウが開いて変更レポートが表示されます。

ビルド・グラフの作成

ALI 拡張機能では, ビルドのグラフを作成できます。

次のグラフを作成できます。

グラフ	説明
ビルド - サマリ グラフ	ALM プロジェクトに現在存在するビルドの数を示します。 X 軸に表示するデータのタイプと, ALM によるデータの分類基準として使用するビルド情報を指定します。
ビルド - トレンド グラフ	ALM プロジェクトで選択したビルド・フィールドについて, 履歴情報を表示期間ごとに示します。 ビルドの数を表示する対象となるフィールドと, データの表示期間を指定します。

グラフを作成するには, 次のいずれかを実行します。

1. グラフ・ウィザードを起動します。

グラフ・ウィザードでは, グラフの作成と設定に必要な作業を手順に沿って実行できます。グラフ・ウィザードは, アナリシス・ビュー・モジュールから起動する方法と, ほかのビルド・モジュールでの作業中に起動する方法があります。

- アナリシス・ビュー・モジュールから:

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。

- ビルド・モジュールから:

[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] ウィンドウが開いたら, エンティティ名を「ビルド」に変更します。グラフ・タイプが2つ表示されるので, 適切なグラフを選択してウィザードの指示に従います。

2. 定義済みのグラフを作成します。

ビルド・モジュールでの作業中は, モジュール・データの分析用グラフをその場で作成できます。

[アナリシス] > [グラフ]を選択し, 次の定義済みグラフのいずれかを選択します。

- ビルド - サマリ グラフ - 'ビルド・カテゴリ' 別
- ビルド - サマリ グラフ - 'ビルド・カテゴリ' 別, 'ビルド・ステータス' 別
- ビルド - サマリ グラフ - 'QA ステータス' 別

- ビルド -トレンド グラフ -'ビルド カテゴリ' 別
3. **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。
4. **アナリシス・ビュー・モジュールの[アナリシス メニュー]タブで, 定義済みのグラフを作成します。**
- [ビルドのメイン アナリシス メニュー]フォルダには, 次の定義済みグラフが表示されます。
- ビルド - サマリ グラフ -'ビルド・カテゴリ' 別
 - ビルド - サマリ グラフ -'ビルド・カテゴリ' 別, 'ビルド・ステータス' 別
 - ビルド - サマリ グラフ -'QA ステータス' 別
 - ビルド -トレンド グラフ -'ビルド カテゴリ' 別

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・レポートの作成](#)」(39ページ)を参照してください。