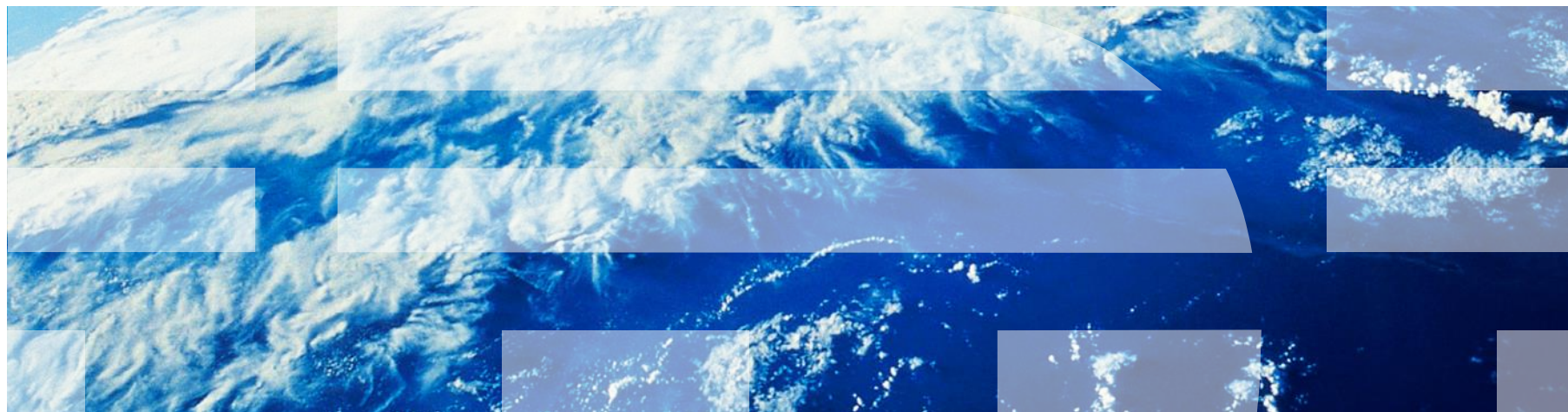


# ***IBM Worklight V6.1.0*** **入門**

**Rational Team Concert を使用してアプリケーションをビルド**



## 商標

- IBM、IBM ロゴおよび [ibm.com](http://www.ibm.com) は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。
- Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。
- この資料は、事前に IBM の書面による許可を得ずにその一部または全部を複製することは禁じられています。

## IBM® について

- <http://www.ibm.com/ibm/us/en/> を参照してください。

# アジェンダ

- Rational Team Concert ビルド・システムを使用して Worklight プロジェクトをビルド
  - ネイティブ・バイナリー (iOS 用) のビルド
  - ネイティブ・バイナリー (Android 用) のビルド
  - ハイブリッド・リソースのビルドおよびデプロイ
- サンプル Worklight ビルド・スクリプト

# Rational Team Concert ビルド・システムを使用して Worklight プロジェクトをビルド (1/2)

- Worklight® プロジェクト内に含まれているコードを Rational Team Concert™ に送信できます。
- Rational Team Concert (RTC) ビルド・システムを使用すれば、以下のいずれかの方法を使用して Worklight プロジェクトをビルドできます。
  - 個人用ワークスペースのプライベート・コンテンツに基づいて個人用ビルドを要求します。
  - ストリーム内で配信されるコンテンツに基づいて定期的なビルドをスケジュールします。
- IBM® Worklight の Consumer Edition および Enterprise Edition には Ant タスクが含まれています。この Ant タスクにより、Worklight プロジェクト内の Web コードがビルドされ、ビルドの結果が Worklight Server にデプロイされます。
- Android SDK や Xcode など、モバイル・オペレーティング・システムのネイティブ SDK にはコマンド・ライン・インターフェースが含まれています。このコマンド・ライン・インターフェースと Ant を併用すれば、デバイス上で実行するネイティブ実行可能ファイル (Android の .apk ファイルや、iOS の .ipa ファイルなど) をビルドできます。
- ヒント: IBM Worklight の Enterprise Edition で利用できる Application Center を使用している場合は、提供されている Ant タスクを使用してネイティブ実行可能ファイルをアプリケーションにデプロイできます。

# Rational Team Concert ビルド・システムを使用して Worklight プロジェクトをビルド (2/2)

- Rational Team Concert ビルド・システムでは、以下のビルド・タスクおよびデプロイ・タスクがサポートされています。
  - **ネイティブ・バイナリー (iOS 用) のビルド:** Xcode で提供されるコマンド・ライン・インターフェースを使用して、iPhone および iPad のネイティブ実行可能ファイルをビルドするためのビルド定義を作成できます。
  - **ネイティブ・バイナリー (iOS 用) を Application Center にデプロイ:** iOS 用のネイティブ・バイナリーを Application Center にデプロイする方法については、IBM Worklight ユーザー文書の『アプリケーションをアップロードするためのコマンド・ライン・ツール』を参照してください。
  - **ネイティブ・バイナリー (Android 用) のビルド:** Android SDK で提供されるコマンド・ライン・インターフェースを使用して、Android のネイティブ実行可能ファイルをビルドするためのビルド定義を作成できます。
  - **ネイティブ・バイナリー (Android 用) を Application Center にデプロイ:** Android 用のネイティブ・バイナリーを Application Center にデプロイする方法については、IBM Worklight ユーザー文書の『アプリケーションをアップロードするためのコマンド・ライン・ツール』を参照してください。
  - **ハイブリッド・リソースのビルドおよびデプロイ:** Worklight プロジェクト内に含まれているハイブリッド Web コードをビルドするためのビルド定義を作成できます。
    - ビルド定義では、ハイブリッド・コードをビルドしてデプロイするために **worklight-ant.jar** ファイルからの Ant タスクが使用されます。

# アジェンダ

- Rational Team Concert ビルド・システムを使用して Worklight プロジェクトをビルド
  - ネイティブ・バイナリー (iOS 用) のビルド
  - ネイティブ・バイナリー (Android 用) のビルド
  - ハイブリッド・リソースのビルドおよびデプロイ
- サンプル Worklight ビルド・スクリプト

## ネイティブ・バイナリー (iOS 用) のビルド (1/10)

- iOS 用のネイティブ・ビルドの一環として、最初に Worklight プロジェクトの共通層および環境層において (またオプションでスキン層においても) Web リソースをビルドする必要があります。さらに、Xcode や他の Worklight クライアント・サイド・ランタイム・ライブラリー (Cordova や JSON Store API など) で必要となるネイティブ成果物を生成する必要があります。
- Xcode で提供されるコマンド・ライン・インターフェースを使用して、iPhone および iPad のネイティブ実行可能ファイルをビルドするためのビルド定義を作成できます。
- 作業を開始する前に、適切なバージョンの Xcode が Mac ワークステーション上にインストールされていることを確認します。
- iOS ビルド用の Jazz™ ビルド・エンジンが実行される環境を Mac ワークステーション上に準備するには、後続のスライドにある手順に従います。

## ネイティブ・バイナリー (iOS 用) のビルド (2/10)

1. 以下のいずれかの方法を使用して、Mac ワークステーションを開発者の証明書およびプロビジョニング・プロファイルでセットアップします。
  - A. プロビジョニング・ポータル
  - B. 手動セットアップ



## ネイティブ・バイナリー (iOS 用) のビルド (3/10)

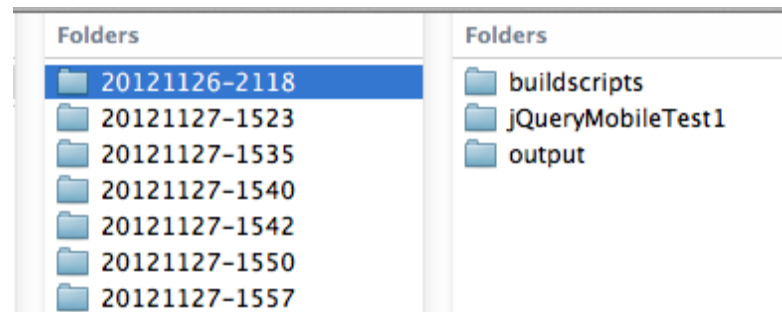
- A. プロビジョニング・ポータル: この方法は、Mac を特定の開発者アカウント専用にすることができる場合に使用します。
  - a. iOS Developer Program メンバー・センターにログオンします。
  - b. iOS Provisioning Portal の「Development Provisioning Assistant」セクション内にある「**Launch Assistant**」をクリックします。
  - c. Development Provisioning Assistant に従ってワークステーションをセットアップします。

## ネイティブ・バイナリー (iOS 用) のビルド (4/10)

- B. 手動セットアップ:** この方法は、開発アカウントの 1 次開発ワークステーションとして Mac を構成できない場合に使用します。
- a. Xcode を開き、「**Window**」 > 「**Organizer**」 > 「**Library**」 > 「**Provisioning Profiles**」を選択します。
  - b. 「**Automatic Device Provisioning**」オプションの横にある「**Refresh**」をクリックします。
  - c. プロビジョニング・ポータルにログオンするように求められたら、iOS Enterprise Developer Program アカウントを使用してログオンします。
  - d. プロビジョニング・プロファイルが正しくインポートされていることを確認します。
  - e. プロビジョニング・プロファイルに関する様々な警告が表示され、適合する秘密鍵でプロファイルを検証できないことが示されます。この問題を解決するには、以下の手順を実行します。
    - i. プロビジョニング・ポータルを使用してセットアップした Mac ワークステーション上で「**Keychain Access**」を開きます。
    - ii. login や System などの iOS プログラムからインストールされた証明書を見つけます。
    - iii. その証明書を展開し、その証明書に関連付けられている秘密鍵を表示します。
    - iv. 秘密鍵を右クリックし、「**Export to a .p12 file**」を選択します。
    - v. 当該ファイルをビルド・ワークステーションに転送して Keychain にインポートします。
    - vi. プロビジョニング・プロファイルが認識されるようになったことを確認します。

## ネイティブ・バイナリー (iOS 用) のビルド (5/10)

2. IBM Worklight インストール・フォルダー <WLInstall> からビルド・ホスト上のディレクトリーに以下のファイルをコピーします。
  - <WLInstall>/WorklightServer にある **worklight-ant.jar**
  - <WLInstall>/ApplicationCenter/tools にある **applicationcenterdeploytool.jar**
  - <WLInstall>/ApplicationCenter/tools にある **json4j.jar**
3. ビルド・フォルダー (例: /Users/username/Documents/wlbuild) を作成します。
  - このフォルダーに RTC ビルド・エンジンがチェックアウトする一時リソースが保持され、出力がビルドされます。
  - RTC ビルド・エンジンは、ビルドごとに、プロジェクト・リソースを保持して出力をビルドするための固有のフォルダーを作成します。このフォルダーは以下のようになります。



## ネイティブ・バイナリー (iOS 用) のビルド (6/10)

4. このモジュールに関連する圧縮ファイルからサンプル・スクリプトをダウンロードします。詳しくは、このモジュールの終わりにあるスライド『サンプル・ビルド・スクリプト』を参照してください。
5. 以下のファイルをプロジェクトにコピーし、**buildscripts** と命名し、RTC にチェックインします。
  - **ios.build.xml**: iOS ビルド用のメイン・スクリプトです
  - **hybrid.build.xml**: ハイブリッド・ビルドを実行するためにメイン・スクリプトで使用されます
  - **appcenter.build.xml**: ビルドの結果を Application Center にアップロードするためにメイン・スクリプトで使用されます

## ネイティブ・バイナリー (iOS 用) のビルド (7/10)

### 6. Rational Team Concert でビルド定義を作成します。

**ヒント:** このステップを実行するには Eclipse Rational Team Concert クライアントを使用する必要があります。

- a. Rational Team Concert Eclipse クライアントを開き、プロジェクト域に接続します。
  - b. 「チーム成果物 (Team Artifacts)」ビューで、ターゲット Rational Team Concert サーバーへの接続が存在することを確認します。
  - c. Worklight プロジェクトを含むプロジェクト域に接続します。
  - d. このプロジェクト域内で、「ビルド (Builds)」を右クリックし、「新規ビルド定義 (New Build Definition)」をクリックします。
  - e. ターゲット・プロジェクト域が選択されていることを確認し、「次へ (Next)」をクリックします。
  - f. 「コマンド・ライン - Jazz ビルド・エンジン (Command Line-Jazz Build Engine)」テンプレートを選択し、「次へ (Next)」をクリックします。
  - g. プレビルド用に「Jazz ソース制御 (Jazz Source Control)」チェック・ボックスを選択します。
  - h. ポストビルド・アクション用のすべてのチェック・ボックスがクリアされていることを確認します。
  - i. 「終了 (Finish)」をクリックします。
- 新規ビルド定義用のエディターが開きます。

# ネイティブ・バイナリー (iOS 用) のビルド (8/10)

## 7. 新規ビルド定義にある以下の値を更新します。

オプション	説明
サポートするビルド・エンジン (Supporting Build Engines)	iOS ビルドを実行するようにセットアップされているビルド・エンジンの ID を指定します。
ロード・ディレクトリー (Load directory)	/Users/username/Documents/wlbuild/\${buildLabel}
ビルド・ファイル (Build file)	/Users/username/Documents/wlbuild/\${buildLabel}/buildscripts/ios.build.xml
ビルド・ターゲット (Build target)	「すべて (all)」を指定します。
Jazz ビルド・ツールキット (Jazz build toolkit)	「Ant ライブラリー・パスに Jazz ビルド・ツールキット・タスクを含める (Include the Jazz build toolkit tasks on the Ant library path)」にチェック・マークを付けます。
Ant 引数 (Ant arguments)	次のように指定します。 「-verbose -lib <path to worklight-ant.jar> -lib <path to applicationcenterdeploytool.jar> -lib <path to json4j.jar>」

## ネイティブ・バイナリー (iOS 用) のビルド (9/10)

8. 「プロパティ (Properties)」タブで以下のプロパティの値を指定します。

プロパティ	説明
<i>appcenterAdminId</i>	Application Center の管理者 ID を指定します。
<i>appcenterAdminPwd</i>	Application Center の管理者パスワードを指定します。
<i>appcenterContext</i>	Application Center のコンテキスト・ルート値を指定します。デフォルトは「applicationcenter」です。
<i>appVersion</i>	アプリケーションのバージョン (例: 1.0) を指定します。
<i>certificate</i>	ステップ 3 または 4 で構成した証明書の名前を指定します。通常、この名前の先頭は「iPhone Developer:」または「iPhone Distribution:」です。
<i>configuration</i>	構成設定を指定します。 <ul style="list-style-type: none"> <li>- QA のために結果の ipa を無線でインストールできるように、デフォルトは「アドホック (Ad Hoc)」です。</li> <li>- プロジェクトのニーズに応じて、ビルド定義内で「デバッグ (Debug)」または「リリース (Release)」でオーバーライドできます。</li> <li>- 詳しくは、iOS 開発ガイドを参照してください。</li> </ul>
<i>mobilePlatform</i>	「 <b>iphone</b> 」を指定します。
<i>provisioning.profile</i>	<i>configuration</i> 設定に対応する .mobileprovision ファイルへのパスを指定します。
<i>rtcPassword</i>	ビルド・エンジンで使用される RTC アカウントのパスワードを指定します。
<i>rtcUserId</i>	ビルド・エンジンで使用される RTC アカウントのユーザー ID を指定します。

# ネイティブ・バイナリー (iOS 用) のビルド (9/10)

## 8. (続き)

プロパティ	説明
<i>wlAdapter</i>	ビルドする Worklight アダプターを指定します。
<i>wlApp</i>	ビルドする Worklight アプリケーションを指定します。
<i>wlContext</i>	Application Center のコンテキスト・ルート値を指定します。デフォルトは「applicationcenter」です。
<i>wlHost</i>	Web アーカイブを公開する Worklight サーバーのホスト名とポート番号を指定します。
<i>wlProject</i>	アダプターおよびアプリケーションを含む Worklight Studio プロジェクトの名前を指定します。
<i>xcodeSDK</i>	シミュレーターではなくデバイス上で実行するためのビルドとして「 <b>iphoneos</b> 」を指定します。
<i>xcodeSDKver</i>	iOS SDK のバージョン (例: 6.0) を指定します。



## ネイティブ・バイナリー (iOS 用) のビルド (10/10)

9. 更新したビルド定義を保存します。

10. ビルドをテストします。

- a. iOS ネイティブ・ビルド用に構成されたワークステーション上のビルド・エンジンが開始され、現在実行中であり、ビルド要求を受け取るために待機していることを確認します。
  - Jazz ビルド・エンジンのセットアップについては、Rational Team Concert のユーザー文書を参照してください。
- b. Eclipse Rational Team Concert クライアントで新規ビルド定義を右クリックし、「**ビルドの要求 (Request Build)**」をクリックします。
  - ビルドの状況は、「ビルド (Builds)」ビューでモニターできます。

# アジェンダ

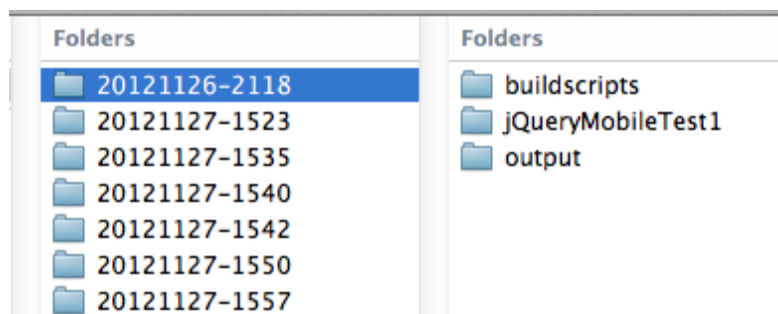
- Rational Team Concert ビルド・システムを使用して Worklight プロジェクトをビルド
  - ネイティブ・バイナリー (iOS 用) のビルド
  - ネイティブ・バイナリー (Android 用) のビルド
  - ハイブリッド・リソースのビルドおよびデプロイ
- サンプル Worklight ビルド・スクリプト

## ネイティブ・バイナリー (Android 用) のビルド (1/8)

- Android 用のネイティブ・ビルドの一環として、最初に Worklight プロジェクトの共通層および環境層において (またオプションでスキン層においても) Web リソースをビルドする必要があります。さらに、Android や他の Worklight クライアント・サイド・ランタイム・ライブラリー (Cordova や JSON Store API など) で必要となるネイティブ成果物を生成する必要があります。
- Android SDK で提供されるコマンド・ライン・インターフェースを使用して、Android のネイティブ実行可能ファイルをビルドするためのビルド定義を作成できます。
- 作業を開始する前に、以下のソフトウェアがビルド・ワークステーション上にインストールされていることを確認します。
  - Oracle Java™ Development Kit (JDK)
  - Android SDK (<http://developer.android.com/sdk> から入手可能)
  - Ant (Apache.org から入手可能)
    - **重要:** PATH システム変数に bin ディレクトリーを必ず追加してください。
- Android ビルド用の Jazz ビルド・エンジンが実行される環境をワークステーション上に準備するには、後続のスライドにある手順に従います。

## ネイティブ・バイナリー (Android 用) のビルド (2/8)

1. IBM Worklight インストール・フォルダー <WLInstall> からビルド・ホスト上のディレクトリーに以下のファイルをコピーします。
  - <WLInstall>/WorklightServer にある **worklight-ant.jar**
  - <WLInstall>/ApplicationCenter/tools にある **applicationcenterdeploytool.jar**
  - <WLInstall>/ApplicationCenter/tools にある **json4j.jar**
2. ビルド・フォルダー (例: /Users/username/Documents/wlbuild) を作成します。
  - このフォルダーに RTC ビルド・エンジンがチェックアウトする一時リソースが保持され、出力がビルドされます。
  - RTC ビルド・エンジンは、ビルドごとに、プロジェクト・リソースを保持して出力をビルドするための固有のフォルダーを作成します。このフォルダーは以下のようになります。



## ネイティブ・バイナリー (Android 用) のビルド (3/8)

3. このモジュールに関連する圧縮ファイルからサンプル・スクリプトをダウンロードします。詳しくは、このモジュールの終わりにあるスライド『サンプル・ビルド・スクリプト』を参照してください。
4. 以下のファイルをプロジェクトにコピーし、「**buildscripts**」と命名し、RTC にチェックインします。
  - **android.build.xml**: Android ビルド用のメイン・スクリプトです
  - **hybrid.build.xml**: ハイブリッド・ビルドを実行するためにメイン・スクリプトで使用されます
  - **appcenter.build.xml**: ビルドの結果を Application Center にアップロードするためにメイン・スクリプトで使用されます

## ネイティブ・バイナリー (Android 用) のビルド (4/8)

5. Android 実行可能ファイル (.apk) に署名するように環境を構成します。
  - <http://developer.android.com/guide/publishing/app-signing.html> にある手順を参照してください。
  - keytool コマンドを使用して有効な鍵ペアを生成します。

## ネイティブ・バイナリー (Android 用) のビルド (5/8)

6. Rational Team Concert でビルド定義を作成します。

ヒント: このステップを実行するには Eclipse Rational Team Concert クライアントを使用する必要があります。

- a. Rational Team Concert Eclipse クライアントを開き、プロジェクト域に接続します。
- b. 「チーム成果物 (Team Artifacts)」ビューで、ターゲット Rational Team Concert サーバーへの接続が存在することを確認します。
- c. Worklight プロジェクトを含むプロジェクト域に接続します。
- d. このプロジェクト域内で、「ビルド (Builds)」を右クリックし、「**新規ビルド定義 (New Build Definition)**」をクリックします。
- e. ターゲット・プロジェクト域が選択されていることを確認し、「**次へ (Next)**」をクリックします。
- f. 「Ant - Jazz ビルド・エンジン (Ant-Jazz Build Engine)」テンプレートを選択し、「**次へ (Next)**」をクリックします。
- g. プレビルド用に「**Jazz ソース制御 (Jazz Source Control)**」チェック・ボックスを選択します。
- h. ポストビルド・アクション用のすべてのチェック・ボックスがクリアされていることを確認します。
- i. 「**終了 (Finish)**」をクリックします。
- 新規ビルド定義用のエディターが開きます。

# ネイティブ・バイナリー (Android 用) のビルド (6/8)

## 7. 新規ビルド定義にある以下の値を更新します。

オプション	説明
サポートするビルド・エンジン (Supporting Build Engines)	Android ビルドを実行するようにセットアップされているビルド・エンジンの ID を指定します。
ロード・ディレクトリー (Load directory)	/Users/username/Documents/wlbuild/\${buildLabel}
ビルド・ファイル (Build file)	/Users/username/Documents/wlbuild/\${buildLabel}/buildscripts/android.build.xml
ビルド・ターゲット (Build target)	「すべて (all)」を指定します。
Jazz ビルド・ツールキット (Jazz build toolkit)	「Ant ライブラリー・パスに Jazz ビルド・ツールキット・タスクを含める (Include the Jazz build toolkit tasks on the Ant library path)」にチェック・マークを付けます。
Ant ホーム (Ant home)	Ant 1.8.3 以降を含むディレクトリーを指定します。
Ant 引数 (Ant arguments)	次のように指定します。 「-verbose -lib <path to worklight-ant.jar> -lib <path to applicationcenterdeploytool.jar> -lib <path to json4j.jar>」



# ネイティブ・バイナリー (Android 用) のビルド (7/8)

8. 「プロパティ (Properties)」 タブで以下のプロパティの値を指定します。

プロパティ	説明
<i>androidSDKPath</i>	Android SDK を含むディレクトリーを指定します。
<i>androidTarget</i>	SDK コンパイル・ターゲットの ID (例: android-15) を指定します。ID のリストはコマンド・ライン「<android sdk>/tools/android list targets」から取得できます。
<i>appcenterAdminId</i>	Application Center の管理者 ID を指定します。
<i>appcenterAdminPwd</i>	Application Center の管理者パスワードを指定します。
<i>appcenterContext</i>	Application Center のコンテキスト・ルート値を指定します。デフォルトは「applicationcenter」です。
<i>configuration</i>	構成設定 (「debug」または「release」) を指定します。
<i>rtcPassword</i>	ビルド・エンジンで使用される RTC アカウントのパスワードを指定します。
<i>rtcUserId</i>	ビルド・エンジンで使用される RTC アカウントのユーザー ID を指定します。
<i>wlAdapter</i>	ビルドする Worklight アダプターを指定します。
<i>wlApp</i>	ビルドする Worklight アプリケーションを指定します。
<i>wlContext</i>	Application Center のコンテキスト・ルート値を指定します。デフォルトは「applicationcenter」です。
<i>wlHost</i>	Web アーカイブを公開する Worklight サーバーのホスト名とポート番号を指定します。
<i>wlProject</i>	アダプターおよびアプリケーションを含む Worklight Studio プロジェクトの名前を指定します。

## ネイティブ・バイナリー (Android 用) のビルド (8/8)

9. 更新したビルド定義を保存します。
10. ビルドをテストします。
  - a. Android ネイティブ・ビルド用に構成されたワークステーション上のビルド・エンジンが開始され、現在実行中であり、ビルド要求を受け取るために待機していることを確認します。
    - Jazz ビルド・エンジンのセットアップについては、Rational Team Concert のユーザー文書を参照してください。
    - ビルド・エンジンを開始する前に、環境変数 JAVA\_HOME が存在すること、およびその環境変数で Oracle JDK インストール・ロケーションが識別されることを確認します。
  - b. Eclipse Rational Team Concert クライアントで新規ビルド定義を右クリックし、「**ビルドの要求 (Request Build)**」をクリックします。
    - ビルドの状況は、「ビルド (Builds)」ビューでモニターできます。

# アジェンダ

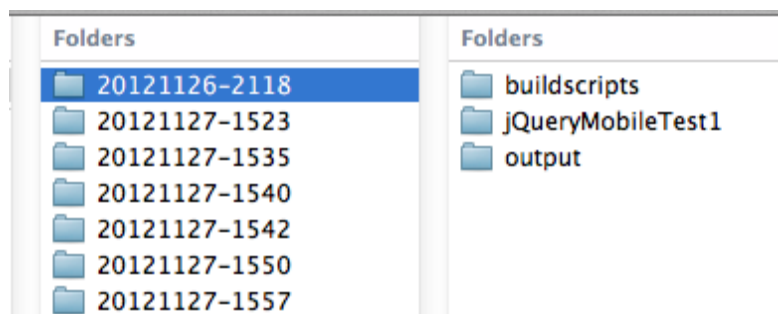
- Rational Team Concert ビルド・システムを使用して Worklight プロジェクトをビルド
  - ネイティブ・バイナリー (iOS 用) のビルド
  - ネイティブ・バイナリー (Android 用) のビルド
  - ハイブリッド・リソースのビルドおよびデプロイ
- サンプル Worklight ビルド・スクリプト

## ハイブリッド・リソースのビルドおよびデプロイ (1/7)

- Worklight プロジェクト内に含まれているハイブリッド Web コードをビルドするためのビルド定義を作成できます。このビルド定義は、アプリケーションの更新によって Web コードのみが変更される (ネイティブ・コードが変更されることはない) 場合に使用されます。
- ビルド定義では、ハイブリッド・コードをビルドしてデプロイするために **worklight-ant.jar** ファイルからの Ant タスクが使用されます。
- ハイブリッド・ビルド用の Jazz ビルド・エンジンが実行される環境をワークステーション上に準備するには、後続のスライドにある手順に従います。

## ハイブリッド・リソースのビルドおよびデプロイ (2/7)

1. IBM Worklight インストール・フォルダー <WLInstall> からビルド・ホスト上のディレクトリーに以下のファイルをコピーします。
  - <WLInstall>/WorklightServer にある **worklight-ant.jar**
2. ビルド・フォルダー (例: /Users/username/Documents/wlbuild) を作成します。
  - このフォルダーに RTC ビルド・エンジンがチェックアウトする一時リソースが保持され、出力がビルドされます。
  - RTC ビルド・エンジンは、ビルドごとに、プロジェクト・リソースを保持して出力をビルドするための固有のフォルダーを作成します。このフォルダーは以下のようになります。



## ハイブリッド・リソースのビルドおよびデプロイ (3/7)

3. このモジュールに関連する圧縮ファイルからサンプル・スクリプトをダウンロードします。詳しくは、このモジュールの終わりにあるスライド『サンプル・ビルド・スクリプト』を参照してください。
4. 以下のファイルをプロジェクトをコピーし、「**buildscripts**」と命名し、RTC にチェックインします。
  - **hybrid.build.xml**: ハイブリッド・ビルドを実行するためにメイン・スクリプトで使用されます
  - アプリケーション内で Dojo Toolkit を使用する場合は、**app-builder** エレメントに **skinBuildExtensions="build-dojo.xml"** 属性を追加します。この属性は、必須 Dojo リソースをアプリケーション出力フォルダーにコピーするために必要となる Ant タスクを実行するために要求されます。

# ハイブリッド・リソースのビルドおよびデプロイ (4/7)

5. Rational Team Concert でビルド定義を作成します。

ヒント: このステップを実行するには Eclipse Rational Team Concert クライアントを使用する必要があります。

- a. Rational Team Concert Eclipse クライアントを開き、プロジェクト域に接続します。
- b. 「チーム成果物 (Team Artifacts)」ビューで、ターゲット Rational Team Concert サーバーへの接続が存在することを確認します。
- c. Worklight プロジェクトを含むプロジェクト域に接続します。
- d. このプロジェクト域内で、「ビルド (Builds)」を右クリックし、「**新規ビルド定義 (New Build Definition)**」をクリックします。
- e. ターゲット・プロジェクト域が選択されていることを確認し、「**次へ (Next)**」をクリックします。
- f. 「Ant - Jazz ビルド・エンジン (Ant-Jazz Build Engine)」テンプレートを選択し、「**次へ (Next)**」をクリックします。
- g. プレビルド用に「**Jazz ソース制御 (Jazz Source Control)**」チェック・ボックスを選択します。
- h. ポストビルド・アクション用のすべてのチェック・ボックスがクリアされていることを確認します。
- i. 「**終了 (Finish)**」をクリックします。
- 新規ビルド定義用のエディターが開きます。

# ハイブリッド・リソースのビルドおよびデプロイ (5/7)

## 6. 新規ビルド定義にある以下の値を更新します。

オプション	説明
サポートするビルド・エンジン (Supporting Build Engines)	ハイブリッド・ビルドを実行するようにセットアップされているビルド・エンジンの ID を指定します。
ロード・ディレクトリー (Load directory)	/Users/username/Documents/wlbuild/\${buildLabel}
ビルド・ファイル (Build file)	/Users/username/Documents/wlbuild/\${buildLabel}/buildscripts/hybrid.build.xml
ビルド・ターゲット (Build target)	「すべて (all)」を指定します。
Jazz ビルド・ツールキット (Jazz build toolkit)	「Ant ライブラリー・パスに Jazz ビルド・ツールキット・タスクを含める (Include the Jazz build toolkit tasks on the Ant library path)」にチェック・マークを付けます。
Ant 引数 (Ant arguments)	次のように指定します。 [-verbose -lib <path to worklight-ant.jar>]



## ハイブリッド・リソースのビルドおよびデプロイ (6/7)

7. 「プロパティ (Properties)」 タブで以下のプロパティの値を指定します。

プロパティ	説明
<i>rtcPassword</i>	ビルド・エンジンで使用される RTC アカウントのパスワードを指定します。
<i>rtcUserId</i>	ビルド・エンジンで使用される RTC アカウントのユーザー ID を指定します。
<i>wlAdapter</i>	ビルドする Worklight アダプターを指定します。
<i>wlApp</i>	ビルドする Worklight アプリケーションを指定します。
<i>wlContext</i>	Application Center のコンテキスト・ルート値を指定します。デフォルトは「applicationcenter」です。
<i>wlHost</i>	Web アーカイブを公開する Worklight サーバーのホスト名とポート番号を指定します。
<i>wlProject</i>	アダプターおよびアプリケーションを含む Worklight Studio プロジェクトの名前を指定します。

## ハイブリッド・リソースのビルドおよびデプロイ (7/7)

8. 更新したビルド定義を保存します。
9. ビルドをテストします。
  - a. ハイブリッド・ビルド用に構成されたワークステーション上のビルド・エンジンが開始され、現在実行中であり、ビルド要求を受け取るために待機していることを確認します。
    - Jazz ビルド・エンジンのセットアップについては、Rational Team Concert のユーザー文書を参照してください。
  - b. Eclipse Rational Team Concert クライアントで新規ビルド定義を右クリックし、「**ビルドの要求 (Request Build)**」をクリックします。
    - ビルドの状況は、「ビルド (Builds)」ビューでモニターできます。

# アジェンダ

- Rational Team Concert ビルド・システムを使用して Worklight プロジェクトをビルド
  - ハイブリッド・リソースのビルドおよびデプロイ
  - ネイティブ・バイナリー (iOS 用) のビルド
  - ネイティブ・バイナリー (Android 用) のビルド
- サンプル Worklight ビルド・スクリプト

## サンプル・ビルド・スクリプト

- この研修モジュール用のサンプル・ビルド・スクリプトは、IBM Worklight 文書 Web サイトの「Getting Started」ページにあります (<http://www.ibm.com/mobile-docs>)。
- このビルド・スクリプトをダウンロードして、以下の項目を探索します。
  - Rational Team Concert ビルド・ツールキット: ビルド結果ページでビルドの結果をダウンロード・リンクとして公開するタスクなど、多くのタスクが含まれています。
  - Worklight Application Center ビルド・タスク: ネイティブ・アプリケーションを Application Center カタログに公開するタスクです。

# 特記事項

- これらの資料は、以下のご使用条件に同意いただける場合に限りご使用いただけます。
- 本書は米国 IBM が提供する製品およびサービスについて作成したものです。
- 本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、または サービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。
- IBM は、本書に記載されている内容に関して特許権（特許出願中のものを含む）を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。
  - 〒103-8510  
東京都中央区日本橋箱崎町19番21号  
日本アイ・ビー・エム株式会社  
法務・知的財産  
知的財産権ライセンス渉外

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

- この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。
- 本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。
- IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。
- 本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム（本プログラムを含む）との間で情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。
  - IBM Corporation  
Dept F6, Bldg 1  
294 Route 100  
Somers NY 10589-3216  
USA

- 本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。
- 本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。
- IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお問い合わせください。

## 著作権使用許諾:

- 本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。
- それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。
  - © (お客様の会社名) (西暦年) このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。© Copyright IBM Corp. \_年を入れる\_ All rights reserved.

## プライバシー・ポリシーの考慮事項

- サービス・ソリューションとしてのソフトウェアも含めた IBM ソフトウェア製品（「ソフトウェア・オファリング」）では、製品の使用に関する情報の収集、エンド・ユーザーの使用感の向上、エンド・ユーザーとの対話またはその他の目的のために、Cookie ははじめさまざまなテクノロジーを使用することがあります。多くの場合、ソフトウェア・オファリングにより個人情報が収集されることはありません。IBM の「ソフトウェア・オファリング」の一部には、個人情報を収集できる機能を持つものがあります。ご使用の「ソフトウェア・オファリング」が、これらの Cookie およびそれに類するテクノロジーを通じてお客様による個人情報の収集を可能にする場合、以下の具体的事項を確認ください。
- このソフトウェア・オファリングは、展開される構成に応じて、(アプリケーション・サーバーが生成する) セッション情報を収集するセッションごとの Cookie を使用する場合があります。これらの Cookie は個人情報を含まず、セッション管理のために要求されるものです。加えて、匿名ユーザーの認識および管理のために持続的な Cookie が無作為に生成される場合があります。これらの Cookie も個人情報を含まず、要求されるものです。
- この「ソフトウェア・オファリング」が Cookie およびさまざまなテクノロジーを使用してエンド・ユーザーから個人を特定できる情報を収集する機能を提供する場合、お客様は、このような情報を収集するにあたって適用される法律、ガイドライン等を遵守する必要があります。これには、エンドユーザーへの通知や同意の要求も含まれますがそれらには限られません。このような目的での Cookie を含む様々なテクノロジーの使用の詳細については、IBM の『IBM オンラインでのプライバシー・ステートメント』(<http://www.ibm.com/privacy/details/jp/ja/>) の「クッキー、ウェブ・ビーコン、その他のテクノロジー」および『IBM Software Products and Software-as-a-Service Privacy Statement』(<http://www.ibm.com/software/info/product-privacy/>) を参照してください。

# サポートおよびコメント

- IBM Worklight の一連の文書、トレーニング資料、および質問をポストできるオンライン・フォーラムはすべて、次の IBM Web サイトからご覧になれます。
  - <http://www.ibm.com/mobile-docs>
- サポート
  - ソフトウェア・サブスクリプション & サポート (ソフトウェア・メンテナンスと呼ばれる場合もあります) は、パスポート・アドバンテージおよびパスポート・アドバンテージ・エクスプレスから購入されたライセンスに含まれています。International Passport Advantage Agreement および IBM International Passport Advantage Express Agreement の追加情報については、次のパスポート・アドバンテージ Web サイトを参照してください。
    - <http://www.ibm.com/software/passportadvantage>
  - ソフトウェア・サブスクリプション & サポートが有効になっている場合、IBM は、インストールおよび使用法 (ハウツー) に関する短期間の FAQ に対するサポートや、コード関連の質問に対するサポートを提供します。詳しくは、次の IBM ソフトウェア・サポート・ハンドブックを参照してください。
    - <http://www.ibm.com/support/handbook>
- ご意見
  - 本資料に関するご意見をお寄せください。本資料の具体的な誤りや欠落、正確性、編成、題材、または完成度に関するご意見をお寄せください。お寄せいただくご意見は、本マニュアルまたは製品の情報、およびその情報の提示方法に関するもののみとしてください。
  - 製品の技術的な質問および情報、および価格については、担当の IBM 営業所、IBM ビジネス・パートナー、または認定リマーケットアーにお問い合わせください。
  - IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。IBM またはいかなる組織も、お客様から提示された問題についてご連絡を差し上げる場合にのみ、お客様が提供する個人情報を使用するものとします。
  - どうぞよろしくお願いいたします。
  - 次の IBM Worklight Developer Edition サポート・コミュニティにご意見をお寄せください。
    - <https://www.ibm.com/developerworks/mobile/worklight/connect.html>
  - IBM からの回答を希望される場合は、以下の情報をご連絡ください。
    - 氏名
    - 住所
    - 企業または組織
    - 電話番号
    - Eメール・アドレス

ありがとうございました

