

Borland®

Silk Test 16.0

モバイル Web アプリ
ケーションのテスト

**Borland Software Corporation
700 King Farm Blvd, Suite 400
Rockville, MD 20850**

Copyright © Micro Focus 2015. All rights reserved. Silk Test は Borland Software Corporation に由来する成果物を含んでいます, Copyright © 2015 Borland Software Corporation (a Micro Focus company).

MICRO FOCUS, Micro Focus ロゴ、及びその他は Micro Focus IP Development Limited またはその米国、英国、その他の国に存在する子会社・関連会社の商標または登録商標です。

その他、記載の各名称は、各所有社の知的所有財産です。

2015-02-24


目次

モバイル Web アプリケーションのテスト	4
Android 上のモバイル Web アプリケーションのテスト	4
物理 Android デバイス上のモバイル Web アプリケーションのテスト	4
Android エミュレータ上のモバイル Web アプリケーションのテスト	5
USB ドライバのインストール	6
USB デバッグの有効化	6
Android エミュレータのプロキシとして Open Agent を手動で設定する	7
Android デバイスの推奨設定	7
Silk Test 用に Android エミュレータを設定する	7
iOS 上のモバイル Web アプリケーションのテスト	9
物理 iOS デバイス上のモバイル Web アプリケーションのテスト	9
Silk Test アプリケーションを iOS デバイスにインストールする	10
Silk Test アプリケーションを iOS デバイスに自動的にインストールする	11
iOS デバイスのプロキシの設定	11
iOS デバイスの推奨設定	12
モバイル アプリケーションの記録	12
モバイル デバイスの操作	13
モバイル Web アプリケーションのテスト時のトラブルシューティング	13
セキュアな Web アプリケーションをテストするためにルート証明書を手動で追加する	15
セキュアな Web アプリケーションをテストするためにルート証明書をインストールする	16
モバイル Web アプリケーションのテストにおける制限事項	17
モバイル Web サイトでのオブジェクトのクリック	18

モバイル Web アプリケーションのテスト

Silk Test では、モバイル アプリケーション (アプリ) を自動的にテストすることができます。Silk Test を使用した自動テストには、次のメリットがあります。

- モバイル アプリケーションのテスト時間を大幅に減少させることができます。
- テストを一旦作成すれば、数多くの異なるデバイスやプラットフォーム上でモバイル アプリケーションをテストできます。
- エンタープライズ モバイル アプリケーションに要求される信頼性とパフォーマンスを確保できます。
- QA チームのメンバーおよびモバイル アプリケーションの開発者の効率を向上できます。
- モバイル アプリケーションは、多くのモバイル デバイスとプラットフォームで動作することを要求されるため、アジャイルにフォーカスした開発環境にとって手動テストは十分効率的とは言えない場合があります。

 **注:** Silk Test は、Android および iOS デバイスの両方でのモバイル Web アプリとハイブリッド モバイル アプリのテストをサポートします。

モバイル アプリケーションのテストをサポートするオペレーティング システムとサポートするブラウザについての情報は、『[リリース ノート](#)』を参照してください。

Android 上のモバイル Web アプリケーションのテスト


Silk Test では、Android デバイスまたは Android エミュレータ上のモバイル アプリケーションをテストすることができます。

物理 Android デバイス上のモバイル Web アプリケーションのテスト


物理 Android デバイス上のモバイル アプリケーションをテストするには、次のタスクを実行します。

1. Silk Test をインストールしたマシンにデバイスを接続します。
2. このマシンで、この Android デバイスをはじめてテストしている場合、適切な Android USB ドライバをマシンにインストールします。
詳細については、『[USB ドライバのインストール](#)』を参照してください。
3. USB デバッグを Android デバイスで有効化します。
詳細については、『[USB デバッグの有効化](#)』を参照してください。
4. Android デバイスが接続しているマシン上で Open Agent が実行していることを確認します。

モバイル Web アプリケーションをテストする場合、Open Agent は Android デバイスに対するプロキシとして自動的に使用されます。

 **注:** ネットワーク接続が Android デバイスでアクティブである必要があります。

5. **Silk Test Web Tunneler** アプリが Android デバイスにインストールされていない場合、Open Agent とデバイス間の USB 接続を有効化するために、Silk Test はアプリをインストールします。
6. HTTPS を使用したセキュアなモバイル Web アプリケーションをテストするために、Silk Test はフック中にルート証明書をデバイスまたはエミュレータにコピーします。証明書がインストールされていない場合、**Silk Test Web Tunneler** アプリはルート証明書がインストールされていないことを示すメッセージ ボックスを表示します。メッセージをクリックして、証明書をインストールします。

 **注:** 証明書がフック中に自動的にインストールされない場合、「[モバイル Web アプリケーションのテスト時のトラブルシューティング](#)」または「[セキュアな Web アプリケーションをテストするためにルート証明書を手動で追加する](#)」を参照してください。

7. デバイスまたはエミュレータ上ですべてのブラウザを閉じ、Web アプリケーションのすべての必要な証明書がインストールされ使用されているかどうかを Silk Test が確認できるようにします。
8. モバイル アプリケーション用の Silk Test プロジェクトを作成します。
9. モバイル アプリケーション用のテストを作成します。
10. **モバイルの記録** 機能を使用して、モバイル アプリケーションに対するテストを記録します。
11. **モバイルの記録** 機能が開始すると、**アプリケーションの選択** ダイアログ ボックスが開きます。使用するモバイル ブラウザーを選択し、記録を開始します。
12. 選択したブラウザが Web に接続できない場合、**Silk Test Web Tunneler** アプリにプロキシ設定が正しくないことを示すメッセージが表示されていないか、確認してください。プロキシ設定を手動で変更するには：
 - a) Android デバイスで使用しているワイヤレス接続のプロキシ設定を開きます。プロキシ設定の開き方についての詳細は、Android デバイスのドキュメントを参照してください。
 - b) **プロキシ** または **プロキシ ホスト名** フィールドに「localhost」を入力します。
 - c) **ポート** フィールドに「9999」を入力します。
 - d) **OK** をクリックします。
13. テストを再生します。


Android デバイスまたはエミュレータの画面が、テスト中にロックされないようにしてください。マシンに接続中にデバイスがロックされないようにするには、**開発者向けオプション** を開きます。**スリープモードにしない** または **充電中に画面をスリープにしない** をチェックします。
14. テスト結果を分析します。

Android エミュレータ上のモバイル Web アプリケーションのテスト

Android エミュレータ上のモバイル Web アプリケーションをテストするには、次のタスクを実行します。


1. Silk Test のエミュレータ設定を構成します。

詳細については、「[Silk Test 用に Android エミュレータを設定する](#)」を参照してください。
2. Android エミュレータを開始します。
3. モバイル アプリケーションをテストするには、Open Agent を Android エミュレータのプロキシとして設定します。

 **注:** エミュレータがインストールされているマシン上で Open Agent が実行していることを確認します。

詳細については、「[Android デバイスまたはエミュレータのプロキシとして Open Agent を手動で設定する](#)」を参照してください。
4. HTTPS を使用したセキュアなモバイル Web アプリケーションをテストするには、Web アプリケーションのルート証明書をエミュレータ上にインストールします。

詳細については、「[セキュアな Web アプリケーションをテストするためにルート証明書をインストールする](#)」を参照してください。

 **注:** Open Agent をプロキシとして設定してからルート証明書を直接インストールしてください。これは、Android エミュレータを使用したときにルート証明書をインストールできないという Android エミュレータの問題があるためです。
5. デバイスまたはエミュレータ上ですべてのブラウザを閉じ、Web アプリケーションのすべての必要な証明書がインストールされ使用されているかどうかを Silk Test が確認できるようにします。
6. モバイル アプリケーション用の Silk Test プロジェクトを作成します。

7. モバイル アプリケーション用のテストを作成します。
8. **モバイルの記録** 機能を使用して、モバイル アプリケーションに対するテストを記録します。
9. テストを再生します。
Android デバイスまたはエミュレータの画面が、テスト中にロックされないようにしてください。マシンに接続中にデバイスがロックされないようにするには、**開発者向けオプション** を開きます。**スリープモードにしない** または **充電中に画面をスリープにしない** をチェックします。
10. テスト結果を分析します。

USB ドライバのインストール

モバイル アプリケーションをテストするために、ローカル マシンに最初に Android デバイスに接続するには、適切な USB ドライバをインストールする必要があります。

デバイスの製造元は、そのデバイスに必要なすべてのドライバをもった EXE を提供している可能性があります。この場合、ローカル マシンにその EXE をインストールするだけです。製造元がこのような EXE を提供していない場合、マシン上にデバイスに対する単一の USB ドライバをインストールできます。

Microsoft Windows 7 上に Android USB ドライバをインストールするには：

1. デバイス用の適切なドライバを探します。
USB ドライバを探してインストールする方法についての詳細は、『<http://developer.android.com/tools/extras/oem-usb.html>』を参照してください。
2. Android デバイスをローカル マシンの USB ポートに接続します。
3. デスクトップ、または **Windows Explorer** から、**コンピュータ** を右クリックし、**管理** を選択します。
4. 左側のペインで、**デバイス マネージャ** を選択します。
5. 右側のペインで、**その他のデバイス** を探して展開します。
6. デバイス名 (*Nexus S* など) を右クリックして、**ドライバ ソフトウェアの更新** を選択します。**ハードウェアの更新ウィザード** が開きます。
7. **コンピュータを参照してドライバ ソフトウェアを検索します** を選択して、**次へ** をクリックします。
8. **参照** をクリックして、USB ドライバ フォルダを探します。
デフォルトでは、Google USB ドライバは、<sdk>%extras%google%usb_driver% にあります。
9. **次へ** をクリックしてドライバをインストールします。

既存の USB ドライバのアップグレード、または他のオペレーティング システムに USB ドライバをインストールする方法については、『<http://developer.android.com/tools/extras/oem-usb.html>』を参照してください。

USB デバッグの有効化

Android Debug Bridge (adb) 上で Android デバイスと通信するために USB デバッグを有効化します。

1. Android デバイスで設定を開きます。
2. **開発者向けオプション** (Dev Settings) をタップします。
開発者向けオプションは、デフォルトでは表示されません。開発者向けオプションがデバイスの設定メニューに含まれていない場合：
 - a) 画面を下にスクロールさせて、デバイスが携帯電話の場合は **端末情報** を、タブレットの場合は **タブレット情報** をタップします。
 - b) 再度画面を下にスクロールさせて **ビルド番号** を 7 回タップします。
3. **開発者向けオプション** ウィンドウで、**USB デバッグ** をオンにします。
4. デバイスの USB モードをデフォルトの設定である **メディア デバイス (MTP)** に設定します。
詳細については、デバイスのドキュメントを参照してください。

Android エミュレータのプロキシとして Open Agent を手動で設定する

Android エミュレータのプロキシとして Open Agent を設定するには、エミュレータのテストを行いたいマシン上に Open Agent をインストールし、エミュレータ上で [USB デバッグ] を有効化します。

1. Android エミュレータを開始します。
2. Android エミュレータで設定を開きます。
3. **無線とネットワーク** セクションで、**その他** をクリックします。
4. **モバイルネットワーク > アクセスポイント名** を選択します。
5. 既存のアクセス ポイントを選択して編集するか、または新しいアクセス ポイントを作成します。
6. Open Agent がインストールされているマシンの IP アドレスを **プロキシ** または **プロキシ ホスト名** フィールドに入力します。
7. **ポート** をクリックします。
8. **ポート** フィールドに Open Agent のポート番号を入力します。デフォルトでは、ポート番号は動的なので、まず、固定ポート番号を設定する必要があります。ポート番号を変更するには、構成設定 `ext.http.proxy.port` (AppData¥Roaming¥Silk¥SilkTest¥conf¥silkproxy.properties.sample ファイル) を使用して、固定ポート番号を設定します。たとえば、ポート番号を 9999 に設定するには、`ext.http.proxy.port=9999` を設定します。ポート番号を **ポート** フィールドに入力し、`silkproxy.properties.sample` ファイルの名前を `silkproxy.properties` に変更します。
9. **OK** をクリックします。

以上で Open Agent が Android デバイスまたは Android エミュレータのプロキシとして設定されました。Android デバイスまたは Android エミュレータのプロキシの構成についての詳細は、デバイスまたはエミュレータのドキュメントを参照してください。



注: Open Agent が実行している限り、Open Agent をプロキシとして使用してモバイル デバイス上のインターネット接続を使用できます。Open Agent が実行していない場合、接続は機能しないため、モバイル デバイスからインターネットに接続するために他の接続を使用する必要があります。デバイスまたはエミュレータが実行している間にワイアレス ネットワーク接続が削除されると、Open Agent との接続はデバイスまたはエミュレータをシャットダウンするまで開放されません。



Android デバイスの推奨設定

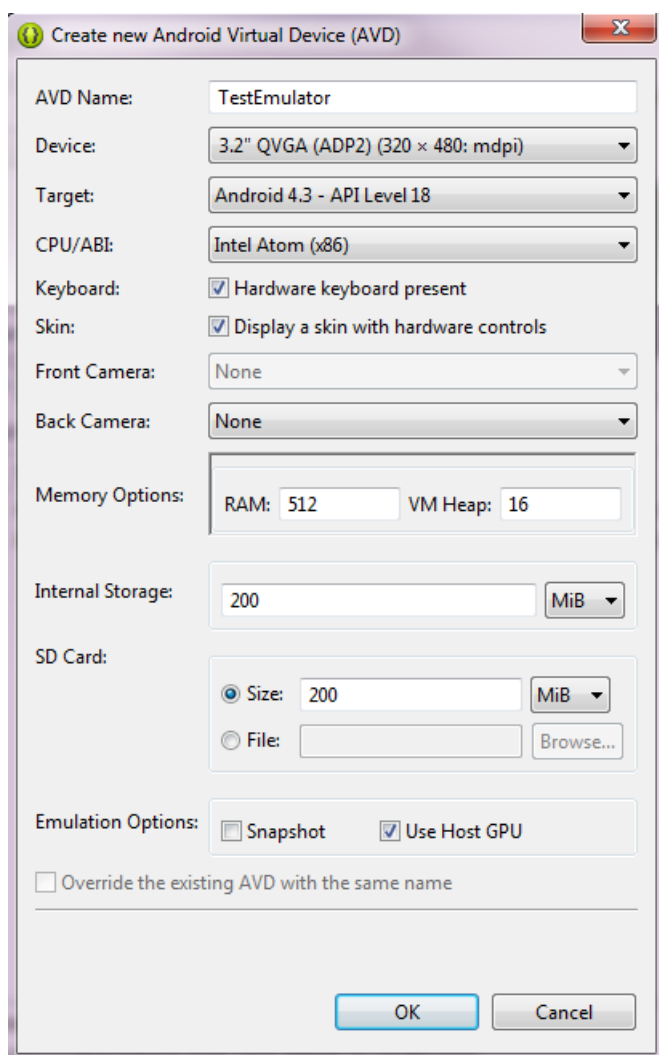
Silk Test を使用したテストを最適化するために、テストしたい Android デバイスで次の設定を行ってください。

- USB デバッグを Android デバイスで有効化します。詳細については、「[USB デバッグの有効化](#)」を参照してください。
- Android デバイスの画面をロックするパターンまたは PIN を設定します。
- Android デバイスは、Open Agent を実行しているマシンに、メディア デバイスとして接続されている必要があります。Android デバイスの USB モードは、**メディア デバイス (MTP)** を設定します。
- Android デバイスまたはエミュレータの画面が、テスト中にロックされないようにしてください。マシンに接続中にデバイスがロックされないようにするには、**開発者向けオプション** を開きます。**スリープモードにしない** または **充電中に画面をスリープにしない** をチェックします。
- Android エミュレータに対する変更 (プロキシの設定など) を保存するには、エミュレータの **Launch Options** ダイアログ ボックスにある **Wipe user data** チェック ボックスをオフにします。

Silk Test 用に Android エミュレータを設定する

Silk Test を使用して Android エミュレータ上でモバイル アプリケーションをテストする場合、テスト用にエミュレータを設定する必要があります。

1. Android SDK をインストールします。
Android SDK のインストールと設定についての詳細は、「[Get the Android SDK](#)」を参照してください。
2. Eclipse 上で、**ウィンドウ > Android SDK Manager** をクリックして、**Android SDK Manager** を起動します。
3. エミュレータを使ってテストするすべての Android のバージョンに対して、バージョン ノードを展開し、**Intel x86 Atom System Image** の隣のチェック ボックスをオンにします。
4. **Install** をクリックして、選択したパッケージをインストールします。
5. **Extras** ノードを展開し、**Intel x86 Emulator Accelerator (HAXM)** の隣にあるチェック ボックスをオンにします。
6. **Install** をクリックして、選択したパッケージをインストールします。
7. *Intel Corporation license agreement* を確認します。条項に同意できる場合は、**Accept** を選択して、**Install** をクリックします。**Android SDK Manager** は、メイン SDK ディレクトリの下での extras ディレクトリにインストーラをダウンロードします。**Android SDK Manager** は、Installed というステータスを表示しますが、これは Intel HAXM 実行可能ファイルがダウンロードされたことを意味します。extras ディレクトリにあるインストーラを実行してインストールする必要があります。
8. extras ディレクトリにあるインストーラを実行し、プラットフォームごとのインストール手順に従います。
9. Eclipse 上で、**ウィンドウ > Android Virtual Device Manager** をクリックし、新しい Android Virtual Device (AVD) を追加します。
- 10 **Android Virtual Devices** タブを選択します。
- 11 **New** をクリックします。
- 12 要件に従って仮想デバイスを設定します。
- 13 エミュレータが使用する RAM サイズを対象のマシンで管理可能な量に設定します。
たとえば、エミュレータの RAM サイズを 512 に設定します。
- 14 SD カードのサイズを設定します。
 **注:** SD カードのサイズを設定しないと場合は、内部ストレージ (Internal Storage) の値を 50 MB 以上に設定する必要があります。そうしないと、証明書ファイルをエミュレータにコピーできません。
- 15 エミュレータのトランザクション速度を向上させるには、**CPU/ABI** フィールドの **Intel Atom (x86)** CPU を選択します。
- 16 省略可能 : エミュレータのトランザクション速度を向上させるには、エミュレーション オプション (Emulation Options) の **Use Host GPU** チェック ボックスをオンにすることもできます。
 **注:** **Use Host GPU** を設定すると、スクリーンショットが撮影できなくなり、**モバイルの記録** ダイアログ ボックスに空のイメージが表示されます。しかし、**モバイルの記録** ダイアログ ボックスのコントロールはハイライトされます。詳細については、<https://code.google.com/p/android/issues/detail?id=58724> を参照してください。



17 OK をクリックします。

18 省略可能 : Android エミュレータに対する変更 (プロキシの設定など) を保存するには、エミュレータの **Launch Options** ダイアログ ボックスにある **Wipe user data** チェック ボックスをオフにします。

iOS 上のモバイル Web アプリケーションのテスト

Silk Test では、iOS デバイス上のモバイル アプリケーションをテストすることができます。

物理 iOS デバイス上のモバイル Web アプリケーションのテスト

物理 iOS デバイス上のモバイル アプリケーション (アプリ) をテストするには、次のタスクを実行します。


1. ハイブリッド アプリをテストする場合は、アプリにアクセスできるようにします。詳細については、[Making Your iOS App Accessible](#) を参照してください。
2. このマシンで初めて iOS デバイス上のモバイル アプリケーションをテストする場合、そのマシンに iTunes をインストールしてください。


iOS デバイス上でテストするために必要なデバイス ドライバが iTunes に含まれているため、必要となります。

3. Silk Test アプリケーションを iOS デバイスにインストールします。詳細については、「[Silk Test アプリケーションを iOS デバイスにインストールする](#)」を参照してください。
4. iOS デバイスのプロキシとして「localhost:9999」を設定します。
iOS デバイスのプロキシを設定する場合の詳細については、「[iOS デバイスのプロキシを設定する](#)」を参照してください。
5. Silk Test をインストールしたマシンにデバイスを接続します。
6. iOS デバイスが接続しているマシン上で Open Agent が実行していることを確認するために、簡単なテストを実行します。
7. Silk Test アプリケーションを iOS デバイスで開きます。
8. HTTPS を使用したセキュアなモバイル Web アプリケーションをテストするには、モバイル Web アプリケーションのルート証明書を Silk Test アプリケーションを使用してインストールします。
9. デバイスまたはエミュレータ上ですべてのブラウザーを閉じ、Web アプリケーションのすべての必要な証明書がインストールされ使用されているかどうかを Silk Test が確認できるようにします。
10. モバイル アプリケーション用の Silk Test プロジェクトを作成します。
11. モバイル アプリケーション用のテストを作成します。
12. **モバイルの記録** 機能を使用して、モバイル アプリケーションに対するテストを記録します。
13. テストを再生します。
iOS デバイスのテスト中に、スリープ モードに移行すべきではありません。画面ロックとパスワードをオフにするには、**設定 > 一般 > パスコードロック** を選択します。iOS 7 で、**設定 > パスコード** を選択します。
14. テスト結果を分析します。


Silk Test アプリケーションを iOS デバイスにインストールする

Open Agent と iOS デバイス間の USB 接続を有効化するため、iOS デバイスに Silk Test アプリケーションをインストールします。

 **注:** iOS デバイスを Silk Test でテストするには、iOS デバイスの UDID が会社の Apple Developer Account に登録されている必要があります。


1. Xcode をダウンロードし (たとえば、<https://developer.apple.com/xcode/downloads/> など)、Mac にインストールします。
Mac は、Silk Test アプリケーションを iOS デバイスにインストールするためだけに必要なもので、高速なものである必要はありません。たとえば、最低限の構成の Mac Mini で十分です。
2. iOS デバイスを Mac に接続します。
3. ダイアログ ボックスが iOS デバイスで開いたら、**信頼** をクリックします。これで Xcode と組み合わせでデバイスを使用できるようになります。最初にアプリを起動すると、会社のデベロッパ プロファイルに一致するプロビジョニング プロファイルがデバイスにインストールされます。
4. SilkTestiOS.zip アーカイブ (デフォルトでは、Open Agent をインストールした Windows マシンの C:\Program Files (x86)\Silk\SilkTest\ng\iOS の下にありますが) を Mac にコピーして、アーカイブを解凍します。
 **注:** アーカイブを解凍するパスワードを入手するには、[SupportLine サイト](#) にログインして、「iOS Password」という件名のインシデントをレポートしてください。
5. **File > Open** をクリックして、Xcode にプロジェクトをインポートするか、.xcodproj ファイルをクリックしてプロジェクトを開きます。
6. Xcode で、デフォルトのターゲットとして設定されている iOS シミュレータの代わりに、使用するデバイスをターゲットとして選択します。
7. プロジェクトの設定で、会社の Developer Program を選択します。

8. 左上隅の矢印をクリックするか、または **Product > Run** を選択します。
9. Silk Test アプリケーションを会社で使用する追加の iOS デバイスに自動的にインストールするには、「[Silk Test アプリケーションを iOS デバイスに自動的にインストールする](#)」を参照してください。
10. Silk Test アプリケーションが iOS デバイス上で、初めて開始されます。

 **注:** Silk Test アプリケーションが iOS デバイス上で正しく開始できたら、iOS デバイス上でアプリケーションのアイコンを単にタップすれば、アプリケーションを開始できます。

Silk Test アプリケーションを iOS デバイスに自動的にインストールする

IPA ファイルを生成し、Silk Test アプリケーションを iOS デバイスに自動的にインストールするように配布します。

 **注:** iOS デバイスを Silk Test でテストするには、iOS デバイスの UDID が会社の Apple Developer Account に登録されている必要があります。

1. Xcode をダウンロードし (たとえば、<https://developer.apple.com/xcode/downloads/> など)、Mac にインストールします。
Mac は、Silk Test アプリケーションを iOS デバイスにインストールするためだけに必要なもので、高速なものである必要はありません。たとえば、最低限の構成の Mac Mini で十分です。
2. iOS デバイスを Mac に接続します。
3. ダイアログ ボックスが iOS デバイスで開いたら、**信頼** をクリックします。これで Xcode と組み合わせてデバイスを使用できるようになります。最初にアプリを起動すると、会社のデベロッパ プロファイルに一致するプロビジョニング プロファイルがデバイスにインストールされます。
4. Xcode で Silk Test アプリケーションをコンパイルします。
5. **Products > Archive** をクリックして、Silk Test アプリケーションの IPA ファイルを生成します。
6. 生成した IPA ファイルと、テストしたいすべての iOS バージョンの DeveloperDiskImage を使用する配布フォルダにコピーします。
 - a) DeveloperDiskImage のデフォルトの場所は、Xcode をインストールしたフォルダの下にある `xCode/Contents/Developer/Platforms/IOS.platform/DeviceSupport/<iOS_バージョン番号>/` です。ここで、`iOS_バージョン番号` は、テストしたいデバイスの iOS バージョンです。
 - b) DeveloperDiskImage には DeveloperDiskImage.dmg と DeveloperDiskImage.dmg.signature の 2 つのファイルがあり、両方ともコピーする必要があります。
7. iOS デバイスのテストを開始するすべてのマシンで、`%APPDATA%\SilkTest\Conf` フォルダを開きます。
8. `iosApp.properties.sample` ファイルの名前を、`iosApp.properties` に変更します。
9. `iosApp.properties` ファイルを開き、ファイルの場所を、IPA ファイルと DeveloperDiskImage をコピーした配布フォルダの場所に変更します。

DeveloperDiskImage をコピーした iOS バージョンの iOS デバイスを **アプリケーションの選択** ダイアログで選択すると、Silk Test アプリケーションが iOS デバイスにインストールされます。


iOS デバイスのプロキシの設定

iOS デバイスのプロキシとして `localhost` を設定するには、デバイスをテストしたいマシン上に Open Agent をインストールします。

1. iOS デバイスで、**設定 > Wi-Fi** をクリックします。
2. アクティブなワイアレス ネットワークの情報ボタン (i) をクリックします。
3. **HTTP プロキシ** セクションで、**手動** を選択します。
4. [サーバ] フィールドに「localhost」を入力します。

5. [ポート] フィールドに「9999」を入力します。

iOS デバイスのプロキシの構成についての詳細は、デバイスのドキュメントを参照してください。


 **注:** Open Agent が実行している限り、モバイル デバイス上のインターネット接続を使用できます。Open Agent が実行していない場合、接続は機能しないため、モバイル デバイスからインターネットに接続するために他の接続を使用する必要があります。デバイスが実行している間にワイアレス ネットワーク接続が削除されると、Open Agent との接続はデバイスをシャットダウンするまで開放されません。

iOS デバイスの推奨設定

Silk Test を使用したテストを最適化するために、テストしたい iOS デバイスで次の設定を行ってください。


- iOS デバイスが Xcode で、開発者モードで実行されていることを確認します。
- Apple Safari が正しく開始されることを確実にするため、**設定 > Safari** をタップして、**Cookie とデータを消去** を選択します。
- 実際にユーザーが行った操作をテストに反映させるために、Apple Safari の自動入力とパスワードの保存を無効化します。**設定 > Safari > パスワードと自動入力** をタップし、**ユーザー名とパスワード** 設定をオフにします。
- iOS デバイスのテスト中に、スリープ モードに移行すべきではありません。画面ロックとパスワードをオフにするには、**設定 > 一般 > パスコードロック** を選択します。iOS 7 で、**設定 > パスコード** を選択します。

モバイル アプリケーションの記録

 **注:** 一部の低レベル メソッドとクラスは、モバイル Web アプリケーションではサポートされません。モバイル Web アプリケーションに対して記録したテストを正しく再生できるようにするためには、モバイル Web アプリケーションに対して記録を行う前に、Silk Test のブラウザ オプションで、**ネイティブなユーザー入力を記録する** オプションをオフにします。詳細については、モバイル Web アプリケーションのテストの制限事項を参照してください。

Silk Test とモバイル デバイスまたはエミュレータとの間の接続が一旦確立すると、デバイス上のモバイルブラウザで実行する操作を記録してテストを作成できます。モバイル Web アプリケーションを記録するには、Silk Test は **モバイルの記録** 機能を使用します。この機能は、標準アプリケーションや Web アプリケーションに対して使われる記録よりもさらに多くの機能を提供します。

モバイルの記録 機能は、テストするモバイル デバイスまたは Android エミュレータの画面を表示します。

 **注:** モバイル デバイスがマシンに接続してなかったり、エミュレータが開始されていなかった場合は、**モバイルの記録** 機能はエラー メッセージを表示します。モバイル デバイスをマシンに接続するか、エミュレータを起動してから、**モバイルの記録** ウィンドウの **更新** をクリックします。

モバイルの記録 機能で操作を実行すると、モバイル デバイス上でも同じ操作が実行されます。

画面上のコントロールを操作すると、**モバイルの記録** 機能はデフォルトの操作を事前に選択します。コントロールに対して有効なすべての操作がリストで表示されるので、実行したい操作を選択するか、単に **OK** をクリックして事前に選択された操作を受け入れます。選択した操作のパラメータの値をパラメータ フィールドに入力することができます。Silk Test は自動的にパラメータを検証します。

コントロールを直接扱うことができない場合 (たとえば、コントロールが他のコントロールで隠されている場合)、**モバイルの記録** ウィンドウの **階層ビューの切り替え** をクリックして、コントロール階層ツリーからコントロールを選択できます。

記録を一時停止すると、画面上での操作は記録されないため、デバイスを記録を続けたい状態に変更することができます。

記録を停止すると、記録した操作でスクリプトが生成されるため、続いてテストの再生を行うことができます。

モバイルデバイスの操作

モバイル デバイスを操作したり、テスト対象アプリケーションでスワイプのような操作を実行するには、次の手順を実行します。

1. **モバイルの記録** ウィンドウで、**モバイル デバイス操作の表示** をクリックします。モバイル デバイスに対して実行できるすべての操作がリストされます。
2. リストからリストから実行したい操作を選択します。
3. Android デバイスまたはエミュレータで、スワイプを記録するには、マウスの左ボタンをクリックしながらマウスを動かします。
4. テストの記録を続行します。

モバイル Web アプリケーションのテスト時のトラブルシューティング

[アプリケーションの選択] ダイアログ ボックスにモバイル ブラウザーが表示されない理由

Silk Test が、次の何れかの理由でモバイル デバイスまたはエミュレータを認識していない可能性があります。

理由	解決策
モバイル デバイスがローカル マシンに接続されていない。	モバイル デバイスをローカル マシンに接続します。
エミュレータが実行されていない。	エミュレータを開始します。
Android Debug Bridge (adb) がモバイル デバイスを認識しない。	モバイル デバイスが adb によって認識されているかどうか確認するには： <ol style="list-style-type: none">1. C:\Program Files (x86)\Silk\SilkTest\ng\agent\plugins\com.microfocus.silktest.adb_15.0.0.6733\bin に移動します。2. Shift を押しながら、ファイル エクスプローラ ウィンドウで右クリックします。3. コマンド ウィンドウをここで開く を選択します。4. コマンド ウィンドウで、adb devices を入力して、アタッチしたすべてのデバイスのリストを得ます。5. デバイスがリストされない場合、USB デバッグがデバイスで有効化されていることを確認します。
デバイスのオペレーティング システムのバージョンを Silk Test がサポートしていない。	サポートするモバイル オペレーティング システムのバージョンについては、 リリース ノート を参照してください。
デバイスの USB ドライバがローカル マシンにインストールされていない。	デバイスの USB ドライバをローカル マシンにインストールしてください。詳細については、「 USB ドライバをインストールする 」を参照してください。
USB デバッグがデバイスで有効化されていない。	USDB デバッグをデバイスで有効化してください。詳細については、「 USB デバッグの有効化 」を参照してください。

モバイル デバイスまたはエミュレータがインターネットに接続できない理由

モバイル デバイスまたはエミュレータのすべてのネットワーク接続のプロキシを構成し、現在どんなテストも記録または再生していない場合、モバイル デバイスまたはエミュレータはインターネットに接続できません。物理モバイル デバイスの場合、**Silk Test Web Tunneler** アプリケーションで接続ステータスを確認できます。

モバイル デバイスが接続され、Open Agent が実行中にも関わらず、モバイル デバイスがまだインターネットに接続できない場合には、プロキシ設定が正しいかどうか確認してください。

Open Agent が実行していない状態でインターネットに接続できるようにするためには、プロキシを一時的に無効化してください。

URL に移動せずに Silk Test が Chrome for Android で URL を検索する理由

アドレスバーに入力された URL を、Chrome for Android が検索として解釈する場合があります。回避策として、URL に移動するコマンドをスクリプトに手動で追加できます。

Android 4.3 の Android エミュレータで記録できない理由

Android バージョン 4.3 の Android エミュレータで記録するには、エミュレータ設定 (Emulator Settings) の **Use Host GPU** チェック ボックスをオフにします。

プロキシを構成したときにモバイル アプリケーションが機能しなくなる理由

WiFi 接続に対して設定できるグローバルプロキシを使用しないモバイル アプリケーションがあります。ブラウザや Gmail のようなアプリケーションは、プロキシ設定を使用しますが、多くのほかのモバイル アプリケーションは、プロキシ設定を無視するため、プロキシが設定されている間、インターネットに接続できません。

adb サーバーが正しく起動しない場合にすべきこと

Android Debug Bridge (adb) サーバーが開始するとき、ローカル TCP ポート 5037 にバインドし、adb クライアントから送信されてくるコマンドをリッスンします。すべての adb クライアントは、ポート 5037 を使用して、adb サーバーと通信します。adb サーバーは、5555 から 5585 の範囲 (エミュレータやデバイスで使用される範囲) で奇数のポートをスキャンしてエミュレータやデバイス インスタンスを探します。adb はこれらのポートの変更を許しません。adb 開始中に問題が発生した場合、これらの範囲のポートの 1 つが、他のプログラムによって既に使用されているかどうか確認します。

詳細については、<http://developer.android.com/tools/help/adb.html> を参照してください。

エラーが発生する理由メモリの割り当てに失敗しました : :8?

エミュレータを開始しているときに、システムが十分なメモリを割り当てることができない場合に、このエラーが表示されます。以下を行ってみてください。

1. エミュレータのメモリ オプションの RAM サイズを下げる
2. Intel HAXM の RAM サイズを下げる RAM サイズを下げるには、IntelHaxm.exe を再度実行して、**Change** を選択します。
3. **タスク マネージャ** を開き、十分なフリー メモリが利用可能かどうかを確認します。不足している場合、プログラムを閉じてメモリを開放してください。

セキュア Web サイトに対して動作しない理由

物理モバイル デバイス上でセキュア Web サイト (HTTPS) をテストできない場合、以下を行っててください。

1. モバイル デバイス上で **Silk Test Web Tunneler** アプリケーションを開き、以下を確認します。
 - セキュアな Web サイトに対して証明書がインストールされている。

- 証明書が、Open Agent がインストールされているマシンのルート証明書と一致している。


証明書がインストールされていない、または Open Agent がインストールされているマシンのルート証明書と一致していない場合、警告メッセージが黄色で表示されます。

2. 警告をクリックして、**OK** を選択し、証明書をインストールします。モバイル デバイスに対してパスワードやスクリーン ロックを設定するために、証明書をインストールする必要があります。パスワードやスクリーン ロックが設定されていない場合、このステップ中に設定するように指示されます。
3. 証明書がデバイス上に見つからない場合、インストールは失敗し、エラー メッセージが表示されます。ファイル `root.crt` が `sdcard/silk/certs/` の下に存在するかどうか確認してください。
4. ファイル `root.crt` が存在しない場合、**ファイル エクスプローラ** を使用して、ファイルを手動でコピーします。モバイル デバイス上に書き込み権を持たない場合、証明書が見つからない可能性があります。
5. 証明書をデバイスにコピーした後に、**Silk Test Web Tunneler** アプリケーションを使うか、ファイル システムで証明書ををクリックして、証明書をインストールできます。

エミュレータ上でセキュア Web サイト (HTTPS) をテストできない場合、Web サイトのルート証明書を手動で追加します。詳細については、「[セキュアな Web アプリケーションをテストするためにルート証明書を手動で追加する](#)」を参照してください。

セキュアな Web アプリケーションをテストするためにルート証明書を手動で追加する

Android バージョン 4.4 以降の Android エミュレータをテストする場合は、このトピックに記述された手順を行うことはできません。Android バージョン 4.4 以降の Android エミュレータでセキュアな Web アプリケーションをテストするためにルート証明書を追加する方法についての情報は、「[セキュアな Web アプリケーションをテストするためにルート証明書をインストールする](#)」を参照してください。


 **注:** このトピックで述べるステップを実行するには、Open Agent を Android デバイスまたは Android エミュレータのプロキシとして設定する必要があります。

Android デバイスや Android エミュレータ上で HTTPS を使ったモバイル Web アプリケーションをテストする場合、特定のサイトを開くリクエストごとに Open Agent がインストールされているマシンでこのサイトに対する証明書が自動的に生成されます。この新しい証明書は元の証明書と同じドメインに対して発行され、元の証明書を置き換えることで SSL 接続によるテストを可能にします。


生成される最初の証明書は、モバイル Web アプリケーションに対するルート証明書です。


Silk Test を使用してアプリケーションをテストできるようにするためには、このルート証明書を Android デバイスまたは Android エミュレータにインストールしなければなりません。デフォルトでは、ルート証明書はフック中にデバイスにコピーされます。ただし、ルート証明書が自動的にインストールされない場合、テストしたいモバイル Web アプリケーションそれぞれに対して一度、ルート証明書をインストールする必要があります。

1. Android 4.4 以降の Android エミュレータでモバイル Web アプリケーションをテストする場合、次の手順を実行します。
 - a) Android デバイスまたは Android エミュレータから、テストしたいモバイル Web アプリケーションを開きます。
 - b) たとえば、www.borland.com を開きます。
 - c) 次のパスを URL に追加します： `/_st_/dynamic/certificate`。たとえば、モバイル ブラウザで `www.borland.com` に対する新しい URL は、次のようになります： `www.borland.com/_st_/dynamic/certificate`。
2. テストしたいモバイル Web アプリケーションを開きます。初めてモバイル Web アプリケーションを開くときに、Open Agent はアプリケーション用の修正したルート証明書を生成します。
3. Open Agent がインストールされているマシン上で、ルート証明書が生成されたフォルダに移動します。
デフォルトでは、フォルダは `%Appdata%¥Silk¥SilkTest¥certs¥authority` になります。

4. ルート証明書ファイル root.crt をコピーします。
5. Android デバイスのストレージのルート フォルダにルート証明書ファイルを貼り付けます。
Android エミュレータ上でテストする場合、Open Agent は、エミュレータのルート ディレクトリに証明書を自動的にコピーします。
 **注:** エミュレータへの証明書のコピーを Open Agent で有効にするには、エミュレータの設定で SD カードのサイズを設定します。
6. 物理 Android デバイス上でテストする場合、ストレージから Android デバイスに証明書をインストールします。
ストレージから証明書をインストールする方法に関する詳細については、Android デバイスまたは Android エミュレータのドキュメントを参照してください。
7. Android エミュレータ上でテストする場合：
 - a) エミュレータ上で **設定 > セキュリティ > SD カードからインストール** に移動します。
 - b) **OK** をクリックして証明書をインストールします。
 - c) 省略可能：**設定 > セキュリティ > 信頼できる認証情報 > ユーザー** に移動して、認証情報がエミュレータにインストールされていることを確認します。
8. デバイスまたはエミュレータ上ですべてのブラウザーを閉じ、Web アプリケーションのすべての必要な証明書がインストールされ使用されているかどうかを Silk Test が確認できるようにします。

セキュアな Web アプリケーションをテストするためにルート証明書をインストールする

 **注:** 物理 Android デバイス、または Android 4.4 より前の Android エミュレータでテストする場合は、「[セキュアな Web アプリケーションをテストするためにルート証明書を手動で追加する](#)」を参照してください。

 **注:** このトピックで述べるステップを実行するには、Open Agent を Android デバイスまたは Android エミュレータのプロキシとして設定する必要があります。

Android デバイスや Android エミュレータ上で HTTPS を使ったモバイル Web アプリケーションをテストする場合、特定のサイトを開くリクエストごとに Open Agent がインストールされているマシンでこのサイトに対する証明書が自動的に生成されます。この新しい証明書は元の証明書と同じドメインに対して発行され、元の証明書を置き換えることで SSL 接続によるテストを可能にします。

生成される最初の証明書は、モバイル Web アプリケーションに対するルート証明書です。

Silk Test を使用してアプリケーションをテストできるようにするためには、このルート証明書を Android デバイスまたは Android エミュレータにインストールしなければなりません。デフォルトでは、ルート証明書はフック中にデバイスにコピーされます。ただし、ルート証明書が自動的にインストールされない場合、テストしたいモバイル Web アプリケーションそれぞれに対して一度、ルート証明書をインストールする必要があります。

1. Android エミュレータから、テストしたいモバイル Web アプリケーションを開きます。
たとえば、www.borland.com を開きます。
2. URL に `/_st_/dynamic/certificate` を追加して、新しい URL に移動します。
たとえば、モバイル ブラウザで www.borland.com/_st_/dynamic/certificate に対する URL は、次のようになります。
3. 証明書のダウンロード ダイアログ ボックスの **証明書の名前** フィールドに証明書の名前を入力します。
4. **証明書の使用** リスト ボックスの **VPN とアプリ** は、デフォルトの設定のままにします。
5. **OK** をクリックします。証明書がエミュレータにインストールされます。
6. デバイスまたはエミュレータ上ですべてのブラウザーを閉じ、Web アプリケーションのすべての必要な証明書がインストールされ使用されているかどうかを Silk Test が確認できるようにします。

モバイル Web アプリケーションのテストにおける制限事項

モバイル ブラウザ上でのテストの再生とロケータの記録のサポートは、サポートされている他のブラウザほど完全なものではありません。以下のリストに、モバイル ブラウザ上でのテストの再生とロケータの記録の既知の制限事項をリストします。

- 次のクラス、インターフェイス、メソッド、プロパティは、モバイル Web アプリケーションでは現時点ではサポートされません。
 - BrowserApplication クラス。
 - CloseOtherTabs メソッド
 - CloseTab メソッド
 - ExistsTab メソッド
 - GetActiveTab メソッド
 - GetSelectedTab メソッド
 - GetSelectedTabIndex メソッド
 - GetSelectedTabName メソッド
 - GetTabCount メソッド
 - OpenTab メソッド
 - SelectTab メソッド
 - DomElement クラス。
 - DomDoubleClick メソッド
 - DomMouseMove メソッド
 - GetDomAttributeList メソッド
 - DomForm クラス。このクラスのすべてのメソッドとプロパティは、モバイル Web アプリケーションではサポートされません。
 - DomRadioButton クラス。
 - RadioListItemCount プロパティ
 - RadioListItems プロパティ
 - RadioListSelectedIndex プロパティ
 - RadioListSelectedItem プロパティ
 - DomTable クラス。このクラスのすべてのメソッドとプロパティは、モバイル Web アプリケーションではサポートされません。
 - DomTableRow クラス。このクラスのすべてのメソッドとプロパティは、モバイル Web アプリケーションではサポートされません。
 - IClickable インターフェイス。
 - Click メソッド。Android デバイスで実行中の Web アプリケーションにクリックを使用できますが、iOS デバイスには使用できません。
 - DoubleClick メソッド
 - PressMouse メソッド
 - ReleaseMouse メソッド
 - IKeyable インターフェイス。このインターフェイスのすべてのメソッドとプロパティは、モバイル Web アプリケーションではサポートされません。
 - イメージ解決は iOS ではサポートされません。iOS デバイスで Web アプリケーションをテストする場合は、イメージ検証のみ使用できます。
- XPath 論理演算子は、標準 HTML 属性に対してのみサポートされており、プロパティやカスタム Silk Test 属性に対してはサポートされていません。たとえば、textContents 属性と innerText 属性に対して論理演算子はサポートされません。これらの演算子を使用して構成される式は、Silk Test の設定に関係なく常に大文字小文字が区別されます。

- バージョン 4.4 以前の Android に搭載されている Android Stock Browser では、XPath 論理演算子がサポートされません。
- 横固定モードでの記録はシステム バーに仮想ボタンを含むエミュレータに対してサポートされません。このようなエミュレータは、回転を正しく検出せずに、横固定モードのシステム バーを画面の下部ではなく画面の右側に配置します。ただし、このようなエミュレータは縦固定モードで記録することができます。

モバイル Web サイトでのオブジェクトのクリック

自動テストの記録と再生中にオブジェクトをクリックするとき、モバイル Web サイトではデスクトップ Web サイトと比較して、次のような困難があります。

- 拡大/縮小率やデバイス ピクセル比が異なる
- さまざまなモバイル デバイスによって画面サイズが異なる
- モバイル デバイス間でのフォントとグラフィックサイズが異なる (通常、デスクトップ ブラウザの Web サイトよりも小さい)。
- さまざまなモバイルデバイスによってピクセル サイズと解像度が異なる

Silk Test は、このような困難をものともせず、モバイル Web サイトの適切なオブジェクトをクリックできます。

モバイル デバイスでテストを記録するときに、Silk Test は Click の記録時に座標を記録しません。ただし、クロス ブラウザ テストの場合、再生中に座標が許されています。また、Click に座標を手動で追加することもできます。Silk Test は、これらの座標をオブジェクトの HTML 座標として解釈します。モバイル デバイスのテストの再生時に BrowserWindow の内側の適切なオブジェクトをクリックするために、Silk Test はオブジェクトの HTML 座標に現在の拡大/縮小率を適用します。デバイスのピクセル座標は、オブジェクトの HTML 座標に現在の拡大/縮小率をかけた座標です。

モバイル Web サイトの現在表示されている領域にオブジェクトが表示されていない場合、Silk Test は Web サイトの適切な位置にスクロールします。

例

HTML ページで 100 x 20 ピクセルの固定サイズの DomButton をテストするコードを以下に示します。

異なるモバイル デバイスまたは異なる拡大/縮小率で再生すると、たとえば DomButton は、デバイス画面上では実際は 10 ピクセルの幅かもしれません。しかし、現在の拡大/縮小率の影響は受けず、上記のコードを使用したときに Silk Test は要素の中央をクリックします。これは、Silk Test が座標を HTML 座標として解釈し、現在の拡大/縮小率を適用するためです。

索引

A

Android

- 物理デバイス上でのテスト 4
- USB デバッグの有効化 6
- USB ドライバのインストール 6
- エミュレータ上でのテスト 5
- エミュレータのプロキシの設定 7
- エミュレータを設定する 7
- 推奨設定 7
- 前提条件 15
- テスト 4
- トラブルシューティング 13

Android エミュレータ

- 前提条件 16

I

iOS

- Silk Test アプリケーションのインストール 10
- Silk Test アプリケーションの自動インストール 11
- 推奨設定 12
- テスト 9
- 物理デバイス上でのテスト 9
- プロキシの設定 11

U

USB ドライバのインストール

- Android 6

き

記録

- モバイル アプリケーション 12

く

クリック

- モバイル Web 18

て

デバイスが接続されていません

- モバイル 13

と

トラブルシューティング

- モバイル 13

ふ

プロキシ サーバー

- Android エミュレータの設定 7
- iOS の設定 11

も

モバイル

- トラブルシューティング 13

モバイル アプリケーション

- 記録 12
- テスト 4

モバイル Web

- クリック 18

モバイル Web アプリケーション

- 制限事項 17

モバイル テスト

- 物理 Android デバイス 4
- Android 4
- Android エミュレータ 5
- iOS 9
- 概要 4
- 物理 iOS デバイス 9

モバイルデバイス

- 操作する 13
- に対して操作を実行する 13

モバイルの記録

- について 12

モバイルブラウザ

- 制限事項 17

る

ルート証明書

- 生成する 15
- 生成する、Android エミュレータ 16
- 追加する 15
- 追加する、Android エミュレータ 16
- ルート証明書を追加する
- Android 15
- Android エミュレータ 16