

## SilkTest Workbench 13.0

---



.NET スクリプト入門  
ガイド

**Micro Focus**  
575 Anton Blvd., Suite 510  
Costa Mesa, CA 92626

Copyright © 2012 Micro Focus. All rights reserved. SilkTest は Borland Software Corporation に由来する成果物を含んでいます, Copyright © 2012 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。

2012-05-07

# 目次


<b>SilkTest Workbench スクリプト チュートリアル</b> .....	<b>4</b>
スクリプトの記録：概要 .....	4
サンプル Web アプリケーションの起動 .....	4
サンプル Web アプリケーションに対するスクリプトの記録 .....	5
記録されたスクリプトの確認 .....	6
記録されたテストの再生 .....	7
結果の分析：概要 .....	7
[結果] ウィンドウの概要 .....	7
スクリプトの拡張：概要 .....	10
検証の挿入 .....	10
アプリケーション データの作成とローカル変数への保存 .....	10
拡張したスクリプトの再生と分析 .....	11
スクリプト内でのスクリプトの実行：概要 .....	12
モジュール式テスト .....	12
2 つめのスクリプトの記録 .....	12
別のスクリプトへのスクリプトの挿入 .....	13
再生エラーへの対応：概要 .....	14
モジュール式スクリプトの再生 .....	14
結果の確認 .....	14
エラーが含まれているスクリプトの変更 .....	14


# SilkTest Workbench スクリプト チュートリアル

SilkTest Workbench スクリプト チュートリアルによろこそ。このチュートリアルでは、スクリプトの作成、スクリプトの再生、および再生結果の分析に必要な基本ステップについて学習します。また、多数の機能を使用して、記録したスクリプトをすばやく更新および拡張する方法についても学習します。

このチュートリアルでは、Microsoft Visual Basic および Microsoft .NET Framework の基本的な知識があることが前提になっています。 .NET Framework についての知識をお持ちでない場合は、Microsoft の Web サイトを参照してください。

このチュートリアルでは、SilkTest Workbench を使用して繰り返し可能なテストを作成する練習をするための SilkTest サンプル Web アプリケーション <http://demo.borland.com/InsuranceWebExtJS/> を使用します。

 **注:** このチュートリアルで使用されているサンプル アプリケーションは、Windows Internet Explorer に合わせて設計され、最適化されています。チュートリアルのレッスンと同じユーザー体験を得るには、Micro Focus は Windows Internet Explorer 以外のサポート対象ブラウザでチュートリアル サンプル アプリケーションを実行することをお勧めしません。

 **注:** Web アプリケーションを記録または再生する前に、システムにインストールされているすべてのブラウザアドオンを無効にします。Windows Internet Explorer でアドオンを無効にするには、**ツール > インターネット オプション** をクリックし、**プログラム** タブをクリックし、**アドオンの管理** をクリックし、アドオンを選択してから **無効にする** をクリックします。

各レッスンは前のレッスンの出力に基づいて進められるため、チュートリアルのレッスンは順番に完了してください。

## スクリプトの記録：概要

サンプル Web アプリケーションで保険の見積もり要求を作成するための操作を行うと、SilkTest Workbench はその操作を記録します。スクリプトに必要な操作の記録を完了すると、記録されたスクリプトがコード ウィンドウに表示されます。

## サンプル Web アプリケーションの起動

このチュートリアルでは、SilkTest サンプル Web アプリケーションを使用します。この Web アプリケーションは、デモ用に提供されています。

Windows Internet Explorer で、SilkTest サンプル Web アプリケーションを使用します。チュートリアルのレッスンと同じユーザー体験を得るには、Windows Internet Explorer 以外のサポート対象ブラウザでサンプル Web アプリケーションを実行しないでください。

1. テストの速度と信頼性を高めるために DOM 関数を記録するには、以下のステップを実行します。
  - a) **ツール > オプション** をクリックします。
  - b) **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。 **記録** オプションが右側のパネルに表示されます。
  - c) **xBrowser** をクリックします。
  - d) **ネイティブなユーザー入力を記録する** リスト ボックスから、**いいえ** を選択します。
  - e) **OK** をクリックします。



**注:** 通常、Web アプリケーションをテストする場合、DOM 関数ではなく、ユーザーの入力そのものを使用します。Flash および Java アプレットなどのプラグインや、AJAX を使用するアプリケーションは、ネイティブなユーザー入力ではサポートされますが、高レベルの API 記録ではサポートされません。

2. ブラウザのすべてのアドオンを確実に無効にするには、以下のステップを実行します。  
Web アプリケーションの記録または再生を行う前に、ブラウザのアドオンをすべて無効にする必要があります。
  - a) Windows Internet Explorer で**ツール > インターネット オプション** を選択します。 **インターネット オプション** ダイアログ ボックスが開きます。
  - b) **プログラム** タブをクリックし、**アドオンの管理** をクリックします。 **アドオンの管理** ダイアログ ボックスが開きます。
  - c) アドオンのリストの **状態** 列で、各アドオンの状態が **無効** になっていることを確認します。  
**状態** 列に **有効** と表示されている場合は、アドオンを選択してから **無効にする** をクリックします。
  - d) **閉じる** をクリックし、**OK** をクリックします。
3. サンプル アプリケーションにリモートでアクセスするには、<http://demo.borland.com/InsuranceWebExtJS> をクリックします。 サンプル アプリケーションの Web ページが開きます。
4. Web アプリケーションをローカルにインストールした場合は、以下のステップを実行します。
  - a) **スタート > すべてのプログラム > Borland > Borland Demo Application > Startup Demo Environment** を選択します。 データベースが起動します。
  - b) **スタート > すべてのプログラム > Borland > Borland Demo Application > Demo Application** を選択します。 サンプル アプリケーションの Web ページが開きます。Web アプリケーションをローカルにインストールするには、**スタート > プログラム > Silk > SilkTest > Sample Applications > xBrowser > Test Application** をクリックしてから EXE ファイルを抽出して実行します。

## サンプル Web アプリケーションに対するスクリプトの記録

記録中は、記録を停止するまでテスト アプリケーションでのすべての操作 (SilkTest Workbench の操作は除く) が SilkTest Workbench によって記録されます。記録が完了したら、生成したスクリプトを変更して、ステップを追加したり、不要なステップを削除したりできます。

1. **ファイル > 新規作成** を選択します。 **資産の新規作成** ダイアログ ボックスが開きます。
2. [資産の種類] リストから **.NET スクリプト** を選択し、**資産名** テキスト ボックスにスクリプトの名前を入力します。  
たとえば、タイトルに AutoQuote と入力します。
3. 記録をすぐに開始するために、**記録の開始** チェック ボックスをオンにします。
4. **OK** をクリックして、スクリプトを資産として保存し、記録を開始します。 **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。
5. リストから **InsuranceWeb: Home - Windows Internet Explorer** を選択し、**OK** をクリックします。 SilkTest Workbench が最小化され、**記録中** ダイアログ ボックスが開きます。
6. Insurance Company Web サイトでは、次のステップのいずれかを行います：  
記録中は、タスクバーの SilkTest Workbench アイコンが点滅します。 **記録中** ダイアログ ボックスでは、現在操作中のオブジェクト、および最後に記録された操作を確認できます。
  - a) **Select a Service or login** リスト ボックスから **Auto Quote** を選択します。 **Automobile Instant Quote** ページが開きます。
  - b) 郵便番号と電子メール アドレスを適切なテキスト ボックスに入力し、**自動車タイプ** をクリックして、**Next** をクリックします。  
このチュートリアル各ステップに従うため、郵便番号に 92121、電子メール アドレスに [jsmith@gmail.com](mailto:jsmith@gmail.com) をそれぞれ入力し、**自動車タイプ** として Car を指定します。
  - c) 年齢を指定し、性別と運転履歴タイプをクリックして、**Next** をクリックします。

たとえば、年齢に 42 を入力し、性別と運転履歴タイプに Male および Good をそれぞれ指定します。

- d) 製造年、車種、モデルを指定し、財務情報タイプをクリックして、**Next** をクリックします。  
たとえば、製造年に 2010 と入力し、車種とモデルに Lexus および RX400 をそれぞれ指定し、財務情報タイプとして Lease を指定します。

指定した情報の概要が現れます。

- e) **Purchase** をクリックします。

**Purchase A Quote** ページが開きます。

- f) ページ上部にある **Home** をクリックして、記録を開始したホーム ページに戻ります。

7. Alt+F10 を押すか、**記録中** ダイアログ ボックスで **記録の停止** をクリックするか、タスクバーの SilkTest Workbench アイコンをクリックして、記録を停止します。**記録完了** ダイアログ ボックスが開きます。**記録完了** ダイアログ ボックスで **次回からこのメッセージを表示しない** チェックボックスがオンになっている場合、記録が停止したあとにこのダイアログ ボックスは表示されません。この場合、スクリプトが表示されます。

8. **.NET スクリプトへ移動** をクリックします。スクリプトが **コード** ウィンドウに表示されます。

9. **保存** をクリックします。

## 記録されたスクリプトの確認

SilkTest Workbench では、それ以外のすべてのアプリケーション内の操作がすべて記録されます。指示に忠実に従っていれば、サンプル アプリケーションの Web サイトで実行した操作だけが SilkTest Workbench によってキャプチャされます。SilkTest Workbench は再生中にこれらの操作を繰り返します。

スクリプトは、以下のサンプルのようになります。

```
Imports SilkTest.Ntf.XBrowser
Public Module Main
    Dim _desktop As Desktop = Agent.Desktop

    Public Sub Main()
        _desktop.Control("control1").TypeKeys("9")
        With _desktop.BrowserApplication("webBrowser")
            With .BrowserWindow("browserWindow")
                .DomTextField("autoquoteZipcode").SetText("92121")
                .DomTextField("autoquoteEMail").SetText("jsmith@gmail.com")
                .DomRadioButton("autoquoteVehicle0").Select()
                .DomButton("autoquoteNext").Select()
                .DomTextField("autoquoteAge").SetText("42")
                .DomRadioButton("autoquoteGender0").Select()
                .DomRadioButton("autoquoteType1").Select()
                .DomButton("autoquoteNext").Select()
                .DomTextField("autoquoteYear").SetText("2010")
                .DomElement("img").DomClick(MouseButton.Left, New Point(8, 9))
                .DomElement("lexus").DomClick(MouseButton.Left, New Point(87, 7))
                .DomElement("img3").DomClick(MouseButton.Left, New Point(11, 10))
                .DomElement("rX400").DomClick(MouseButton.Left, New Point(96, 11))
                .DomRadioButton("autoquoteFinInfo2").Select()
                .DomButton("autoquoteNext").Select()
                .DomLink("home").Select()
            End With
        End With
    End Sub
End Module
```

スクリプトが前述の例と完全に一致しないことがあります。各ユーザーは、それぞれ異なる方法でアプリケーションと対話するためです。たとえば、フォームに情報を入力するとき、各フィールドをクリックし

ていくユーザーもいれば、Tab キーを使用して移動していくユーザーもいます。結果は同じですが、これらの操作の SilkTest Workbench による記録内容は異なります。このような違いはありますが、スクリプトは正しく再生されるはずで

## 記録されたテストの再生

スクリプトを記録し、保存したら、再生して、スクリプトが適切に動作することを確認できます。

1. 以下のいずれかのステップを実行します。

- **操作** > **再生** を選択します。
- ツールバーで **再生** をクリックします。

**再生** ダイアログ ボックスが開きます。このダイアログ ボックスで、結果をどのように保存するかを指定できます。

2. **結果の説明** テキスト ボックスに Initial test results for the recorded test と入力します。

3. **OK** をクリックします。

各結果は、一意のテスト実行番号によって識別されます。

SilkTest Workbench が最小化され、スクリプトが再生されます。再生中、サンプル アプリケーションに対して、スクリプトの記録中に行った操作が画面上で再生されます。再生が正常に完了すると、**再生完了** ダイアログ ボックスが開きます。

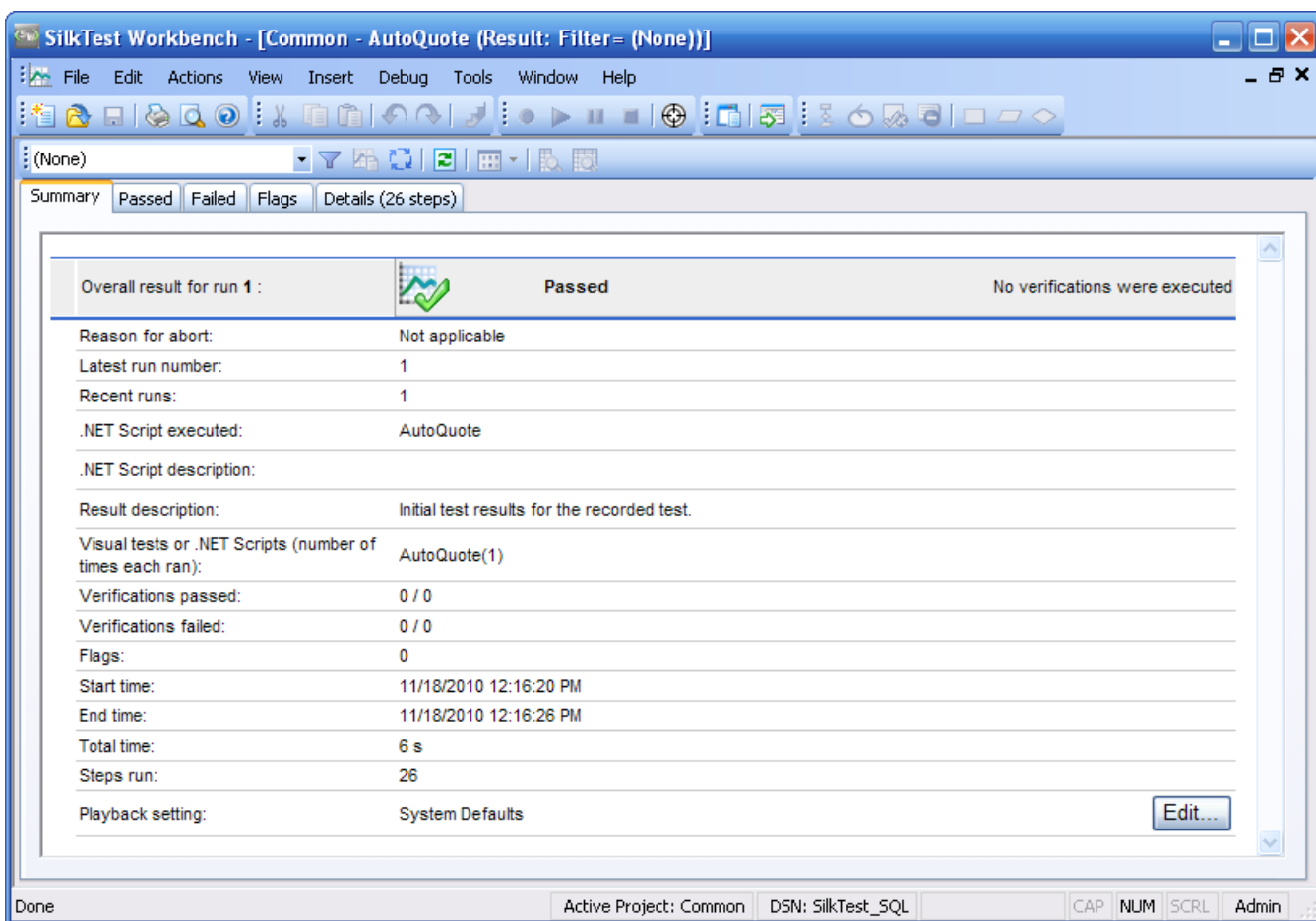
4. **結果へ移動** をクリックします。 **結果** ウィンドウが表示されます。

## 結果の分析：概要

テストを再生すると、SilkTest Workbench によってテスト結果が生成されます。テスト結果にはスクリプトの再生に関する情報が含まれています。スクリプトの名前、実行番号、各ステップが実行された日付と時刻、各ステップの合格/失敗のステータスなどの重要な情報が含まれます。

### [結果] ウィンドウの概要

ビジュアルテストまたはスクリプトを再生したあと、再生結果を **結果** ウィンドウに表示することができます。 **結果** ウィンドウには [ビジュアル ナビゲータ] があり、テストの再生をあらゆる面からすばやく確認できます。 **結果** ウィンドウには、[ビジュアル ナビゲータ] 以外に以下の機能もあります。



## [結果] ウィンドウのツールバー

**結果** ウィンドウのツールバーには、結果に含まれる内容の表示と種類をカスタマイズするための以下のボタンとリストボックスがあります。

- **種類の選択によるフィルタ処理の結果**：すべての事前定義済みおよびユーザー定義の結果フィルタにすばやくアクセスできます。リストからフィルタを選択します。
- **フィルタの管理**：**フィルタの管理** ダイアログ ボックスを開き、そこから結果フィルタを作成、編集、および適用できます。
- **条件**：**条件** ダイアログ ボックスが表示されます。このダイアログ ボックスでは、今後のすべての実行の成功を定義する条件として、合格した検証のパーセンテージを設定できます。たとえば、合格条件を 90% に設定した場合は、ビジュアルテストまたはスクリプトにおける 10 件の検証のうち 9 件以上に合格しないと、再生の結果が合格になりません。このオプションを設定すると、再生結果オプションの**結果合格条件 (パーセンテージ)** が更新されます。このパーセンテージは、今後のすべての結果に適用されます。
- **すべての実行を表示**：各実行結果の詳細を表示する**実行の詳細** ダイアログ ボックスを開きます。このダイアログ ボックスで、以前の実行を開いたり削除したりすることができます。
- **更新**：現在の結果を更新します。
- **表示**：ビジュアルテストについて、**テストステップ** ペインに表示するステップの種類を設定します。このボタンの横にあるドロップダウン矢印をクリックして、ステップのみを表示するか、画面のステップのみを表示するか、またはその両方を表示するかを選択します。**ステップの説明** 列を表示するように選択することもできます。選択された表示は、**結果** ウィンドウの各タブ (**合格**、**失敗**、**フラグ**、および **詳細**) に適用されます。
- **基本表示**：標準的な **テストステップ** ペインの情報、および **結果** と **結果の詳細** の追加の列を表示します。**要約** タブを表示している間は無効になります。



- **詳細表示**：各ステップの詳細情報を表示します。 **要約** タブを表示している間は無効になります。

## [結果] ウィンドウのタブ

**結果** ウィンドウには、結果の内容を特定の種類に分類する 5 つのタブがあります。

- **[要約] タブ**：以下の情報を示す全体的な概要レポートを表示します。
  - **総実行結果**：ビジュアル テストまたはスクリプトが正常に再生されて、結果合格条件（パーセンテージ）を満たしている場合は、[合格] と表示されます。結果合格条件（パーセンテージ）を満たしていない場合は、[失敗] と表示されます。ステップを正しく実行できなかった場合は、[再生エラー] と表示されます。
  - **中止の理由**：ビジュアル テストまたはスクリプトの再生が中止された理由を表示します。
  - **最新の実行番号**：最新の結果の実行番号を表示します。
  - **最近の実行**：最近の実行を表示します。 前の実行を表示するには、表示する実行をクリックします。このフィールドに表示されていない以前の実行を開くには、ツールバーの **すべての実行を表示** をクリックして **実行の詳細** ダイアログ ボックスを開きます。ここで、前の実行を開いたり削除したりできます。
  - **実行されたビジュアル テスト/.NET スクリプト**：結果のビジュアル テストまたはスクリプトの名前を表示します。
  - **ビジュアル テスト/.NET スクリプトの説明**：結果のビジュアル テストまたはスクリプトの説明を表示します。
  - **結果の説明**：結果の説明を表示します。
  - **ビジュアル テスト または .NET スクリプト**：再生の一部として正しく実行されたビジュアル テストまたはスクリプトをすべてリストします。これには、Workbench.RunScript( ) メソッドを使用して実行された、挿入されたビジュアル テストまたはスクリプトも含まれます。たとえば、ドライバ スクリプトでは、一度の再生により複数のスクリプトが実行されることがあります。
  - **合格した検証**：すべてのビジュアル テストまたはスクリプトで、正常に実行され合格した検証の合計数を表示します。このフィールドの数字をクリックすると **合格** タブが開くため、合格したすべての検証を確認できます。
  - **失敗した検証**：すべてのビジュアル テストまたはスクリプトで、正常に実行されたが失敗した検証の合計数を表示します。このフィールドの数字をクリックすると **失敗** タブが開くため、失敗したすべての検証を確認できます。
  - **フラグ**：ビジュアル テストについて、結果でフラグを付けられた検証ステップの総数です。このフィールドの数値をクリックして、**フラグ** タブを開きます。フラグは、スクリプトでは使用できません。
  - **開始時刻**：最初のビジュアル テストまたはスクリプトの再生が開始された時刻です。
  - **終了時刻**：最後のビジュアル テストまたはスクリプトの再生が終了した時刻です。
  - **合計時間**：ビジュアル テストまたはスクリプトを再生した合計時間です。
  - **実行したステップ数**：実行したステップまたはコード行の総数です。
  - **再生設定**：結果の作成に使用される再生オプションのグループである再生設定を表示します。 **編集** をクリックして再生オプションを表示し、再生設定を設定することができます。
- **[合格] タブ**：合格した検証をすべて表示します。
- **[失敗] タブ**：失敗した検証をすべて表示します。再生エラーとなったステップはこのタブには表示されません。
- **[フラグ] タブ**：検証ロジックで作成された、フラグの付いたステップのみを表示します。ビジュアル テストのフラグ付きのステップは、再生後はこのタブに表示されません。フラグは、スクリプトでは使用できません。
- **[詳細] タブ**：ビジュアル テストの各ステップまたはスクリプトの各コード行に関する情報を表示します。テストステップやコード行の名前、合格/失敗のステータス、説明、フラグのステータスなどの情報が表示されます。

## ビジュアルナビゲータのペイン

スクリプトでは、**結果** ウィンドウには **テストステップ** ペインおよび **プロパティ** ペインのみ表示されます。**テストステップ** ペインには、各テストステップの再生ステータスと結果の詳細情報を示す列があります。

## スクリプトの拡張：概要

テストの拡張には、既存のテストが新しいバージョンのテストアプリケーションで使用できるようにテストを更新することなどが含まれます。たとえば、テストアプリケーションのさまざまな状態を処理し、検証するために、テストロジックを挿入することができます。さらに、テストの内容をわかりやすくしたり、重要な点について自分自身や他のユーザーに注意を促すために、メッセージボックスを挿入することができます。

これらは、SilkTest Workbench で既存のテストを拡張し、よりパワフルで堅牢かつ柔軟なテストを作成する、ほんの数例です。

## 検証の挿入

検証とは、ユーザー定義の条件を評価し、合格/失敗のメッセージを再生結果に送信するテストロジックです。

このレッスンでは、検証を挿入して、見積もりで正しい車両モデルが使用されていることを確認します。

1. 自動見積もりのモデルタイプを定義する、以下のテキストを選択します。

```
.DomElement("rX400").DomClick(MouseButton.Left, New Point(96, 11))
```

2. **Model** タイプを指定する Instant Quote ウィザードのページに移動し、異なるモデルタイプに注目します。

たとえば、**GS430** は車種 **Lexus** のモデルタイプです。

3. SilkTest Workbench でモデルタイプを変更します。

以下の textContents コードを

```
.DomElement("rX400").DomClick(MouseButton.Left, New Point(96, 11))
```

から

```
.DomElement("GS430").DomClick(MouseButton.Left, New Point(96, 11))
```

に変更します。

4. 期待値と実際の値を比較して、コメントを追加するには、以下のように入力します。

```
Workbench.Verify ("GS430", .DomTextField("GS430").Text, "The model type is correct")
```

サンプルアプリケーションのプロパティの値を検証するテストロジックを挿入することで、記録したスクリプトを拡張できました。

## アプリケーションデータの作成とローカル変数への保存

変数を使用すると、スクリプトの別の場所で使用するデータ値を保存できるため、テストを拡張できます。データは他の種類のファイルに出力することもできます。

サンプルアプリケーションの **Get Instant Auto Quote** ページには、一意の電子メールアドレスが表示されます。このページの電子メールアドレスを含むテキストは、このページのコントロールのプロパティ値です。

このレッスンでは、ローカル変数 `stremailAddr` に、このテキストを保存します。

1. スクリプトで、電子メールの値に移動します。

コードは以下のようになります。

```
.DomTextField("autoquoteEMail").SetText("jsmith@gmail.com")
```

2. 電子メールの値のあとに、以下のコードを挿入します。

```
Dim StremailAddr As String  
StremailAddr = .DomTextField("autoquoteEMail").Text
```

このステップによって、新しいローカル変数 *StremailAddr* が作成されます。このローカル変数に、**Get Instant Auto Quote page** ページの電子メール アドレスのテキストを保存します。

3. 再生中に変数テキストを表示する出力を含めるには、スクリプトに `Console.Write` コマンドを含めます。  
以下に例を示します。

```
Console.Write (StremailAddr)
```

4. コンソール出力を表示するには、**表示 > 出力** を選択します。**出力** ウィンドウが開きます。スクリプトを再生すると、**出力** ウィンドウが入力されます。

テストでプロパティ値が正しくキャプチャされ、保存されることを確認するために、テストを再生して結果を確認します。

## 拡張したスクリプトの再生と分析

記録したスクリプトにいくつかの拡張を行ったので、スクリプトを再生し、結果を分析してみます。

1. 次のいずれか 1 つのステップを行います：

- **操作 > 再生** を選択します。
- ツールバーで **再生** をクリックします。

**再生** ダイアログ ボックスが開きます。

2. **結果の説明** テキスト ボックスに、Enhanced test results for the script と入力します。
3. **OK** をクリックします。SilkTest Workbench により、拡張されたテストが再生されます。
4. **再生完了** ダイアログ ボックスで **結果へ移動** をクリックします。

**結果** ウィンドウが開き、デフォルトで **要約** タブが表示されます。

**要約** タブに、スクリプトが合格した（エラーなしで正常に再生されたことを意味します）、または失敗した検証が表示されます。

5. **合格 (1)** タブをクリックします。

カッコ内の数は、合格した検証の合計数を示します。**テスト ステップ** ペインには検証ステップが表示され、**結果の詳細** 列には検証の合格の説明が表示されます。

6. **詳細** タブをクリックすると、各操作の結果が表示されます。

7. **出力** ウィンドウで、変数に電子メール アドレスを保存した結果を示すステップまでスクロールします。

テキストは以下のようなものです。

```
jsmith@gmail.com
```

8. **合格** タブをクリックすると、モデル タイプの検証結果が表示されます。

テキストは以下のようなものです。

```
Main:Verify 合格: The model type is correct
```

検証結果は、**結果** 列および **結果の詳細** 列にも表示されます。

お疲れ様でした。これで、サンプル アプリケーションを確実にテストするスクリプトを作成できました。次のレッスンでは、スクリプトを他のスクリプトですばやく簡単に実行する方法など、より高度なテストの概念と機能について学習します。

## スクリプト内でのスクリプトの実行：概要

このチュートリアルでは、Web アプリケーションから自動車保険の見積もりを取得するために必要な各操作を実行する単一のスクリプトを作成しました。単一のスクリプトは、単純なアプリケーションに対して基本的なテスト ケースを実行する場合に便利です。ただし、ほとんどのソフトウェア テストでは、アプリケーションのすべての側面をテストするなど、より厳格なアプローチが必要となります。さらに、テストアプリケーションが変化したときに既存のスクリプトをすばやく更新できることが必要です。

これらのテストの課題を解決する効率的な手段を提供するために、SilkTest Workbench はモジュール式テストをサポートしており、特定のテスト ソリューションの共通の操作を単一のテストにまとめ、同じ操作セットを必要とする他のスクリプトでスクリプトを再利用できるようにしています。

## モジュール式テスト

ビジュアル テストやスクリプトなどの SilkTest Workbench 資産を作成してアプリケーションのテスト ソリューションを構築する前に、テスト戦略を立てることをお勧めします。

1 つのビジュアル テストまたはスクリプトに特定のテスト ソリューションのすべての部分を含める必要はありません。また、通常、そうすることは有益ではありません。

通常、最も効率のよいテスト方法は、モジュール式アプローチを採用したものです。アプリケーション テストを一連のトランザクション単位として考えます。

たとえば、オンライン発注システムのテストには、以下のようなトランザクション単位が含まれるかもしれません。

- オンライン システムへのログオン
- 顧客プロフィールの作成
- 注文の発注
- オンライン システムのログオフ

1 つのテストでこれらの単位をすべて処理し、このテストを使用するシナリオが 10 通りある場合、それらのシナリオを処理するために 10 個の別々のテストを記録する必要があります。アプリケーションに何らかの変更があった場合、たとえば、ログオン ウィンドウにフィールドが 1 つ追加された場合、新しいフィールドへのデータ入力を処理するために 10 個の別々のテストに変更が必要になります。

これらのトランザクション単位をすべてテストするビジュアル テストまたはスクリプトを 1 つ作成して、それをシナリオごとに 10 個作成するよりも、これらのトランザクション単位を 1 つずつ処理する別個のテストをテスト「モジュール」として作成する方が有益です。トランザクション単位ごとに別々のテストを作成し、それをテスト シナリオごとに再利用すれば、ログオン トランザクション単位を処理するテストだけを変更すればよいことになります。

モジュール式テストの基本が理解できたので、2 つめのテストを作成し、前のレッスンで作成したテストに追加してみます。

## 2 つめのスクリプトの記録

このセクションでは、チュートリアルの 2 つめのスクリプトを記録し、スクリプトの別の作成方法を学習します。

1. **ファイル > 新規作成** を選択します。 **資産の新規作成** ダイアログ ボックスが開きます。
2. [資産の種類] リストから **.NET スクリプト** を選択し、 **資産名** テキスト ボックスにスクリプトの名前を入力します。  
このチュートリアルでは、名前として AddAccount と入力します。
3. 記録をすぐに開始するために、 **記録の開始** チェック ボックスをオンにします。
4. **OK** をクリックして、スクリプトを資産として保存し、記録を開始します。 **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。

- リストから **InsuranceWeb: Home - Windows Internet Explorer** を選択し、**OK** をクリックします。 SilkTest Workbench が最小化され、**記録中** ダイアログ ボックスが開きます。
- サンプル アプリケーションの **Home** ページで、**Login** セクションの **Sign Up** をクリックします。  
**Create A New Account** ページが開きます。
- 適切なフィールドに、以下の情報を入力します。  
次のフィールドに移動するには Tab キーを押します。

フィールド名	値
名 (First Name)	Pat
姓 (Last Name)	Smith
生年月日 (Birthday)	1990/02/12
	 <b>注:</b> カレンダー コントロールの月と年の隣にある下矢印をクリックして月と年を変更し、カレンダーで 12 を選択します。
電子メール アドレス (Email Address)	smith@test.com
住所 (Mailing Address)	1212 Test Way
市 (City)	San Diego
州 (State)	CA
郵便番号 (Postal Code)	92121
パスワード (Password)	test

- Sign Up** をクリックします。
- Continue** をクリックします。 連絡先情報が表示されます。
- ページ上部にある **Home** をクリックして、記録を開始したホーム ページに戻ります。
- Log Out** をクリックします。
- Alt+F10 を押して記録を完了します。 **記録完了** ダイアログ ボックスが開きます。
- 保存** をクリックします。 スクリプトが **コード** ウィンドウに表示されます。

## 別のスクリプトへのスクリプトの挿入

このセクションでは、ユーザー アカウントを追加する 2 つめのスクリプトを、元のスクリプトの自動見積もりの要求を行うコードの前に挿入します。

スクリプト内でスクリプトを実行すると、スクリプトで同じ基本操作を効率的にテストできます。



**ヒント:** スクリプトを別のスクリプトに挿入する際、テスト アプリケーションが正しい初期再生状態になっていることを確認することが重要です。

- ファイル > 開く** を選択します。 **アセット ブラウザ** が開きます。
- 左側のペインで **.NET スクリプト** を選択し、スクリプトのリストを表示します。
- リストから **AutoQuote** をダブルクリックして開きます。  
AutoQuote はこのチュートリアルで最初に作成したテストです。
- コード** ウィンドウの `Public Sub Main()` コードの後ろにカーソルを置き、Enter を押して、新規行を追加し、以下のように入力します。

```
Workbench.RunScript ("AddAccount")
```

ここで、`AddAccount` は、作成した 2 つめのスクリプトの名前です。

見積もりステップを実行する前にアカウント情報を追加するため、With ステートメントの前に `Workbench.RunScript` コマンドが追加してあります。 見積もりステップのあとで `AddAccount` スク

リプトを実行するには、Workbench.RunScript コマンドを End With ステートメントのあとに追加します。

## 再生エラーへの対応：概要

再生中のエラーは、テスト アプリケーションの変更、不適切なワークフローなど、さまざまな要因により発生します。このようなエラーをすばやく診断し、修正することにより、テストの保守を最小限にし、チームによるテスト作業の効率を上げることができます。

まず、前のレッスンで作成したモジュール式スクリプトを再生します。

## モジュール式スクリプトの再生

前のレッスンでは、AddAccount を AutoQuote スクリプト挿入することにより、モジュール式スクリプトを作成しました。

このセクションでは、このモジュール式スクリプトを再生し、再生中のエラーを確認します。

1. AutoQuote スクリプトが開いている状態で、ツールバーの **再生** をクリックします。再生 ダイアログボックスが開きます。
2. **結果の説明** テキスト ボックスに Responding to errors in a modular test と入力します。
3. **OK** をクリックします。

再生中に、**Create A New Account** ページでテストが停止し、エラー メッセージが表示されます。

データベースでは、各顧客レコードに対して一意の電子メール アドレスが必要なため、このエラーが発生します。AddAccount スクリプトの記録中にすでに電子メール アドレスを入力しているため、この電子メール アドレスはデータベースにすでに存在しており、テストが失敗します。

## 結果の確認

スクリプトの結果を確認します。

1. **終了** をクリックして、再生を停止します。再生完了 ダイアログ ボックスが開きます。
2. **結果へ移動** をクリックします。AutoQuote の結果が表示され、デフォルトで **要約** タブが表示されます。

**要約** タブにはテスト実行の全体的な詳細が表示されます。**ビジュアル テストまたは .NET スクリプト (実行回数)** フィールドには、「AutoQuote(1)」と挿入されたスクリプト「AddAccount(1)」がリストされます。

3. **詳細** タブをクリックします。
4. 青色のテキストのステップまでスクロールします。

**結果** 列と **結果の詳細** 列を確認すると、再生中に発生したエラーに関する情報をすばやく参照できます。



**注: 失敗** タブには、再生エラーを含むステップは表示されません。失敗した検証のみが表示されます。

このレッスンでは、再生エラーを診断する方法を学習しました。次に、スクリプトを変更してエラーを修正する追加ステップを記録する方法を学習します。

## エラーが含まれているスクリプトの変更

データベースには顧客レコードごとに一意の電子メール アドレスが必要であるため、スクリプト内でエラーが発生します。AddAccount スクリプトの記録中にすでに電子メール アドレスを入力しているため、この電子メール アドレスはデータベースにすでに存在しており、テストが失敗します。テスト内に追加のステップを記録してデータベース重複エラーを解決するか、または別のモジュール式テストを追加できます。たとえば、データベースをリセットする別のスクリプトを記録し、新しいスクリプトを AddAccount

スクリプトから呼び出すことができます。モジュール式スクリプトを作成すると、他のスクリプト内でも Reset Database ステップを使用できます。

1. Internet Explorer で、サンプル Web アプリケーションの **Home** ページに移動します。  
**Home** ページから開始すると、テストが正しい再生状態で開始されることが保証されます。
  2. SilkTest Workbench で、**ファイル > 新規作成** を選択します。 **資産の新規作成** ダイアログ ボックスが開きます。
  3. [資産の種類] リストから **.NET スクリプト** を選択し、**資産名** テキスト ボックスにスクリプトの名前を入力します。  
このチュートリアルでは、名前として ResetDatabase と入力します。
  4. 記録をすぐに開始するために、**記録の開始** チェック ボックスをオンにします。
  5. **OK** をクリックして、スクリプトを資産として保存し、記録を開始します。 **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。
  6. リストから **InsuranceWeb: Home - Windows Internet Explorer** を選択し、**OK** をクリックします。 SilkTest Workbench が最小化され、**記録中** ダイアログ ボックスが開きます。
  7. サンプル アプリケーションの **Home** ページで、ページの右下隅にある **Settings** をクリックします。
  8. **Reset Database** セクションで **Reset** をクリックします。 このステップは、以降のテストで重複するデータが入力されたときにエラーが発生しないようにするために必要です。
  9. ページ上部にある **Home** をクリックして、記録を開始したホーム ページに戻ります。
  - 10 **記録の停止** をクリックし、**.NET スクリプトへ移動** をクリックします。スクリプトが **コード** ウィンドウに表示されます。
  - 11 **保存** をクリックします。
  - 12 AddAccount スクリプトを開きます。
  - 13 **コード** ウィンドウの Public Sub Main() コードの後ろにカーソルを置き、Enter を押して、新規行を追加し、以下のように入力します。  

```
Workbench.RunScript ("ResetDatabase")
```
- ここで、ResetDatabase は、作成した 2 つめのスクリプトの名前です。
- 14 AutoQuote スクリプトを開き、ツールバーの **再生** をクリックします。 3 つのテストがすべてエラーなしで完了します。

# 索引

## け

### 結果

エラーの確認 14

### [結果] ウィンドウ

概要 7

### 検証

スクリプトへの追加 10

## さ

### 再生

記録されたスクリプト 7

### サンプル アプリケーション

起動 4

スクリプトの記録 5

## す

### スクリプト

概要 4

確認 6

記録 12

再生 7, 11

再生エラー 14

サンプル アプリケーション 4

サンプル アプリケーションに対する記録 5

挿入 13

モジュール式 12

### スクリプトの拡張

概要 10

検証の追加 10

再生 11

変数の追加 10

### スクリプトの記録

サンプル アプリケーション 5

### スクリプトの再生

拡張したスクリプト 11

## て

### テスト

モジュール式 12

### テスト結果

ウィンドウ 7

## ひ

### ビジュアル テスト

モジュール式の概要 12

## へ

### 変数

追加 10

## も

### モジュール式スクリプト

エラー 14

概要 12

挿入 13

### モジュール式テスト

概要 12