



Getting Started

with

TestComplete 15

デスクトップ、Web およびモバイル テストの
チュートリアル

A4 サイズの用紙をお使いの方へ: このドキュメントは US Letter サイズで提供されています。そのまま A4 サイズの用紙に印刷することもできますが、下の余白が多くなります。

ドキュメントのサイズを A4 に変更する手順:

1. Word で [レイアウト] に進み、A4 サイズを選択します。
2. 目次ページ (目次の任意の場所) に移動して F9 を押し、目次のページ番号を更新します。
3. 最後のページ (索引の任意の場所) に移動して F9 を押し、索引のページ番号を更新します。
4. 新しい名前でファイルを保存します。

チュートリアルについて

TestComplete を用いることで、デスクトップ、Web、モバイルの 3 種類のアプリケーションをテストすることができます。

- **デスクトップ アプリケーション** - これらのアプリケーションは、Windows オペレーティング システムが動作しているデスクトップ マシンで実行します。
- **Web アプリケーション** - これらのアプリケーションは、Web ブラウザー (デスクトップ アプリケーションに組み込まれた Web ブラウザーを含む) で実行します。
- **モバイル アプリケーション** - これらのアプリケーションは、Android デバイスまたは iOS デバイスで実行します。

このドキュメントは、初めて TestComplete を使用するユーザー向けに用意されたもので、自動テストおよび製品の概要を紹介します。また、主なアプリケーション タイプのテストの作成方法を説明するチュートリアルも含まれています。これらのチュートリアルをお読みになることで、デスクトップ、Web、モバイル アプリケーションのテストを作成、修正、実行することができます。

2021.10.14 - TestComplete 15.0

© 2021 SmartBear Software

Translated by XLsoft Corporation.

目次

自動テストおよび TESTCOMPLETE について	5
自動テスト.....	5
テスト タイプ	5
TestComplete プロジェクトとプロジェクト項目	6
TestComplete ユーザー インターフェイス	8
TestComplete テスト オブジェクト モデル.....	9
チェックポイントと Stores	13
デスクトップ アプリケーションのテスト.....	14
1. テストの計画.....	15
2. レコーディングについて.....	15
3. テストのレコーディング開始.....	17
4. テストするアプリケーションの定義.....	19
5. テストのレコーディング	20
6. レコーディングしたテストの分析	28
7. レコーディングしたテストの実行	32
8. テスト結果の分析	34
WEB アプリケーションのテスト - チュートリアル.....	38
1. Web ブラウザーの準備.....	39
2. テストの計画.....	40
3. テストのレコーディングについて.....	40
4. テストのレコーディング開始.....	41
5. テストする Web ページの定義	43
6. テストのレコーディング	45
7. レコーディングしたテストの分析	54
8. レコーディングしたテストの実行	58
9. テスト結果の分析.....	60
10. 複数のブラウザでのテストの実行	63
ANDROID アプリケーションのテスト.....	67
1. 無料の BitBar アカウントの申し込み	68
2. テストのレコーディング開始.....	69
3. テストするアプリケーションの定義.....	71
4. デバイスへ接続してテスト セッションを開く.....	72
5. テストのレコーディング	73
6. レコーディングしたテストの分析	83
7. テストの実行.....	86
8. テスト結果の分析	87

IOS アプリケーションのテスト	90
1. 無料の BitBar アカウントの申し込み	92
2. テストのレコーディング開始	93
3. テストするアプリケーションの定義	94
4. デバイスへ接続してテストセッションを開く	95
5. テストのレコーディング	99
6. レコーディングしたテストの分析	106
7. レコーディングしたテストの実行	109
8. テスト結果の分析	111
次のステップ	113
テクニカル サポートとリソース	115
エクセルソフト株式会社へのお問合せ	115
索引	116

自動テストおよび TestComplete について

自動テスト

ソフトウェア テストとは、アプリケーションを調査し、そのアプリケーション内のエラーを検出するプロセスです。テストと単なる調査との違いは、テストはアプリケーションの出力と期待値を比較し、アプリケーションが期待通りに機能するかどうかの判定を含んでいることにあります。言い換えると、テスターはアプリケーションが値のリストを表示するのを確かめるだけでなく、そのリストが適切な値を含んでいるかを検証する必要があります。

このことから、基本的なテスト シーケンスは次のようになります。

- 期待する出力の定義。
- テスト動作の実行 (適切な入力データの付与)。
- アプリケーション出力の収集および期待する結果 (基準データ) との比較。
- 比較に失敗した場合は、開発者またはマネージャーに通知。

自動テストとは、若干あるいは全く人間の操作のない特別なプログラムによるソフトウェア テストの自動実行です。自動実行によって、テスト動作を 1 つたりともスキップしないことが保障されます (これは、テスターを退屈な同じ手順の繰り返し作業から解放します)。

TestComplete は、テスト動作の自動化、テストの作成、基準データの定義、テストの実行、テスト結果の記録を行うための特別な機能を提供します。たとえば、ビジュアルにテストを作成できる特別な "**テストのレコーディング**" 機能を持っています。レコーディングを開始して、テストするアプリケーションに対して、必要とされるアクションを実行するだけで、TestComplete が自動的にすべての "記録された" アクションをテストに変換します。また、TestComplete は、テスト内の比較コマンドや**チェックポイント**の自動化を簡単にする特別なダイアログやウィザードも含んでいます。

テスト タイプ

TestComplete はさまざまなテスト タイプとテスト方法 (単体テスト、機能および GUI テスト、リグレッション テスト、分散テストなど) をサポートしています (TestComplete ヘルプの「Different Ways of Testing」を参照)。このチュートリアルでは、最もよく用いられる機能テストを作成します。機能テストは、アプリケーションの GUI とその背後のシステムおよびユーザー間のインターフェイスをチェックします。アプリケーション機能が予想通りであるか検証します。

典型的な機能テストは、クリックやキーストロークのシミュレーション、ループ内でのテスト コマンドの実行、オブジェクトの内容の検証などのさまざまな動作を実行するテスト コマンドから構成されます。

TestComplete では、機能テストは**キーワード テスト**と**スクリプト**の形式で作成できます。このテストは、**レコーディング**あるいは内蔵エディターで**ゼロから作成**することができます。キーワード テストの作成は視覚的で、簡単であり、プログラミングの知識を必要としません。スクリプトはスクリプト コマンドの理解が必要

になりますが、より強力で柔軟性のあるテストの作成が可能になります。TestComplete は以下のスクリプトをサポートしています。

- JavaScript
- Python
- VBScript
- その他の言語

馴染みのある言語でスクリプトを作成できます。サポートされている言語とスクリプト作成言語の選択方法については、TestComplete ヘルプの「*Selecting the Scripting Language*」を参照してください。

このチュートリアルでは、キーワード テスト機能を使用します。

TestComplete プロジェクトとプロジェクト項目

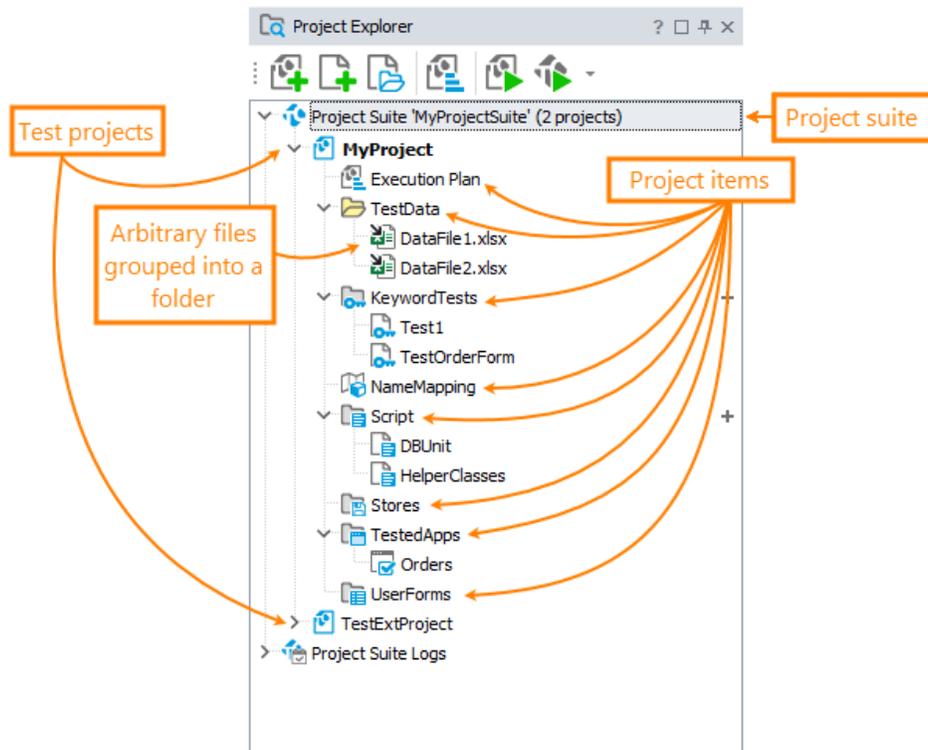
TestComplete はテスト プロジェクトおよびプロジェクト スイートで動作します。**プロジェクト**はテスト作成の開始地点です。プロジェクトには、テスト、チェックポイントの基準データ、テストするアプリケーションについての情報、その他のテスト実行に必要な項目が含まれます。プロジェクトは、テストの実行シーケンスも定義します。

1 つのプロジェクトに、テストするアプリケーションのすべてのテストを含めることができます。複雑なアプリケーションの場合、アプリケーションの一部のみでプロジェクトを作成し、他の部分は他のプロジェクトでテストすることができます。通常は、モジュール単位でプロジェクトを作成します。

関連するプロジェクトは**プロジェクト スイート**にグループ化できます。TestComplete は、新しいプロジェクトを作成するときにプロジェクト スイートを自動的に作成します。空のプロジェクト スイートを作成してプロジェクトを追加することもできます。

プロジェクト項目は、さまざまなテストを実行する、あるいは実行を支援するためのプロジェクト要素です。プロジェクト項目以外にも、プロジェクトには、テストデータ付きの Excel や .csv ファイルのようなヘルパーファイルを含めることができます。

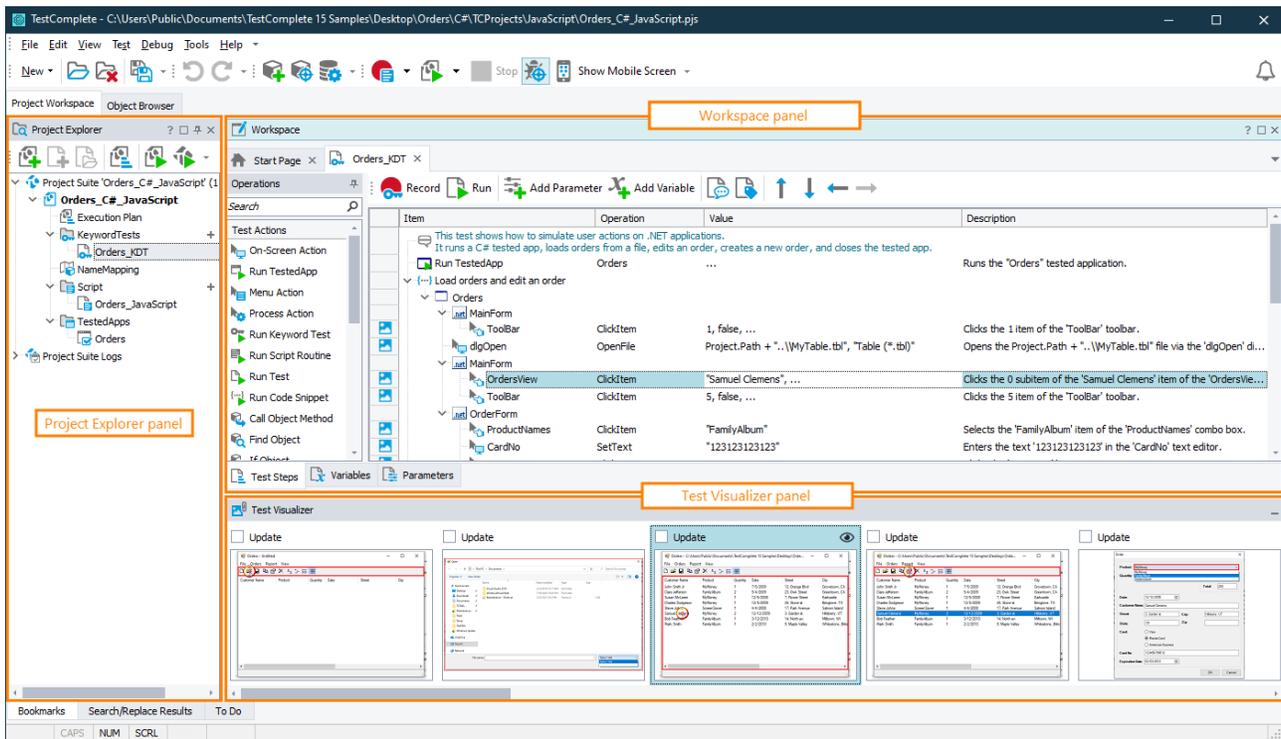
Project Explorer で、プロジェクト、プロジェクト スイート、プロジェクト項目を表示、管理することができます。



TestComplete のプロジェクト項目に関する情報は、TestComplete ヘルプの「About Project Items」を参照してください。

TestComplete ユーザー インターフェイス

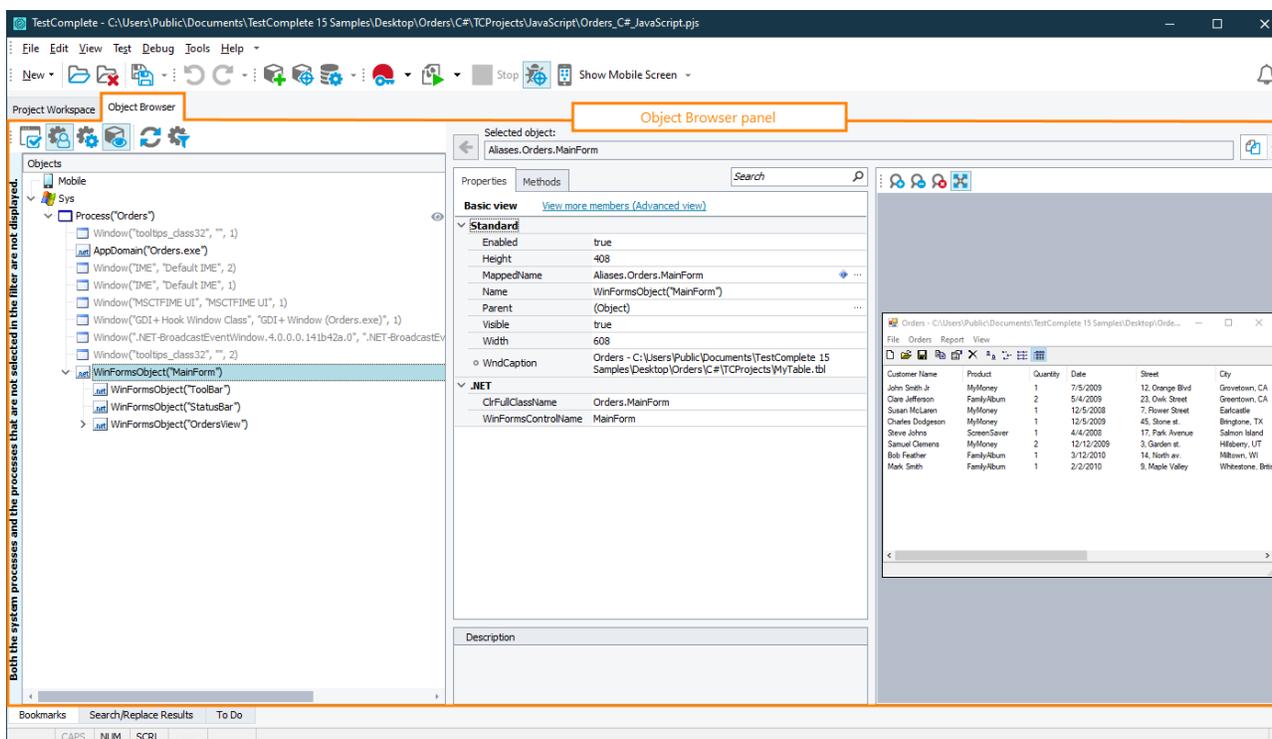
TestComplete のメイン ウィンドウのサンプル画像を次に示します。



TestComplete のユーザー インターフェイスは、いくつかのパネルで構成されています。

- **[Project Explorer]** (ウィンドウの左側) は、プロジェクトとプロジェクトスイートの内容を表示します。また、テスト ログへのアクセスも提供します。
- **[Workspace]** は、作業用デスクトップです。エディターを表示して、テストの作成と修正、テスト結果の確認を行うことができます。たとえば、上記の画像では、**Keyword Test** エディターが **Workspace** 内で開かれています。
- エディターの下側には、**[Test Visualizer]** パネルがあり、テスト コマンドのレコーディング中にテスト エンジンが撮った画像を表示します。これらの画像は、テスト コマンドが実行する動作を理解するのに役立ちます。
- **[Watch List]**、**[Locals]**、**[Breakpoints]**、**[Call Stack]** パネルはテストのデバッグに使用されます。**[To Do]** パネルは、実行すべきタスクを管理します。
- **[Object Browser]** には、すべてのプロジェクトで利用可能なコンテンツが表示されます。マシンに存在し、テスト エンジンで利用可能なすべてのプロセス、ウィンドウおよびコントロール、モバイル デバイスのリストが、アプリケーション内の実際のオブジェクト階層と同じツリー階層で表示されます。

テスト レコーディング中、TestComplete はこれらのオブジェクトとそのメソッドやプロパティを使用します。言い換えると、Object Browser から、テストで利用可能なオブジェクト、メソッド、プロパティと、その取得方法がわかります。詳細は、TestComplete ヘルプの「Exploring Application Properties」を参照してください。



パネルの説明を表示するには、パネル内をクリックして F1 キーを押します。これにより、パネルの解説が開きます。

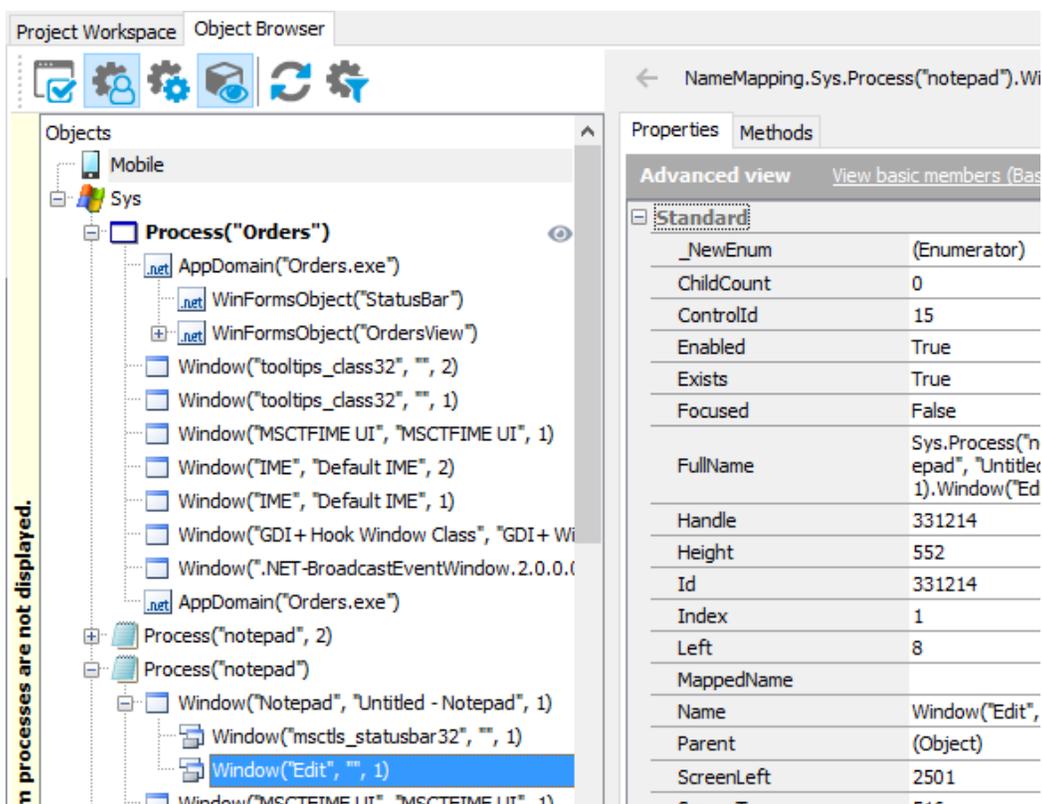
TestComplete には、ユーザー インターフェイスの外観を制御するいくつかのビジュアル テーマがあります。たとえば、ライト (デフォルト) テーマやダークテーマなどです。テーマの切り替えは、**[Tools] > [Options] > [General] > [User Interface]** ダイアログで行います。また、TestComplete のパネルのレイアウトをカスタマイズすることもできます。パネルの非表示、パネルの移動、サイズの変更などを手動で行うことができます。詳細は、TestComplete ヘルプの「Customizing TestComplete」を参照してください。

TestComplete に特定のアクションを実行するように指示するには、メニューとツールバーを使用します。TestComplete のメニュー サブシステムは、Microsoft Visual Studio または他の主要な Windows アプリケーションのメニューとツールバーに類似しています。ツールバーの位置を変更したり、メニューまたはツールバーの項目を他に移動したり、項目を隠したり、元に戻したり、他のタスクを実行することができます。詳細については、TestComplete ヘルプの「Working With TestComplete Toolbars and Menus」を参照してください。

TestComplete テスト オブジェクト モデル

自動テストでは、テストするアプリケーションのオブジェクト (ウィンドウ、コントロール、Web ページ要素) に対するユーザー アクションをシミュレートするため、最初にオブジェクトを識別する必要があります。TestComplete にオブジェクトを見つける方法を指示します。TestComplete はアプリケーションの個々のウィンドウやコントロールを認識でき、それらのオブジェクトへのアクセスを含むテストを提供します。

認識されるオブジェクトは、TestComplete の **[Object Browser]** パネルに表示されます。

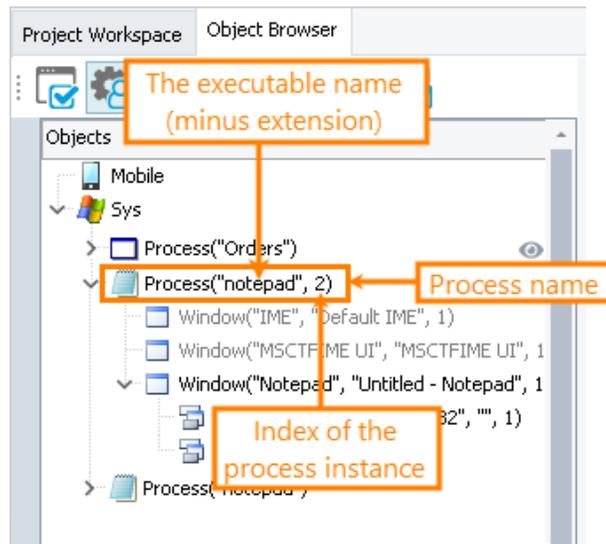


注意: この画像では、デスクトップ アプリケーションのオブジェクト モデルを使用しています。Web およびモバイル アプリケーションのオブジェクト モデルもほぼ同じです。

TestComplete は、テスト オブジェクトにツリー形式のモデルを使用しています。デスクトップと Web アプリケーションのツリーのルート ノードは **Sys** で、モバイル アプリケーションのルート ノードは **Mobile** です。

Process オブジェクトは オペレーティング システムで実行中のアプリケーションに対応します。このオブジェクトは、Windows ドキュメント内のプロセスの概念に対応するため、ここでは**アプリケーション**ではなく**プロセス**という用語を使用しています。

プロセス オブジェクトの名前は、実行可能なプロセスの名前とそのインデックスから構成されています (インデックスはいくつかのアプリケーション インスタンスを実行している場合のみ):

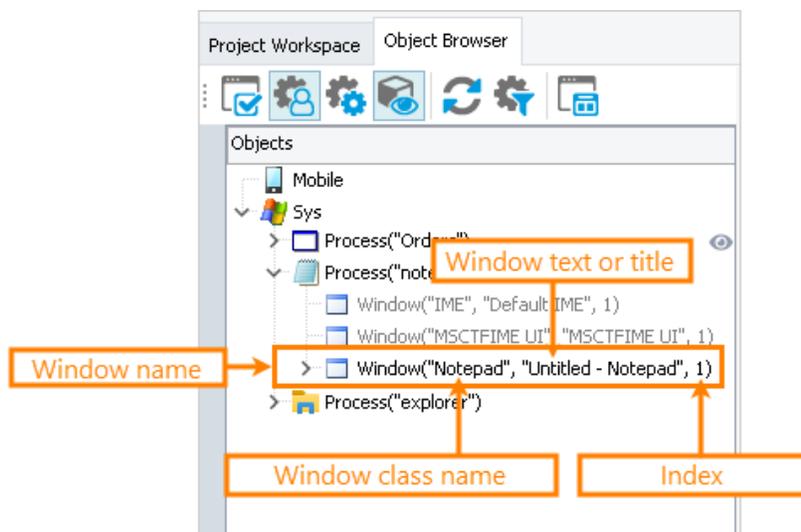


プロセスには、子オブジェクト (最上位ウィンドウに対応するウィンドウ) があり、子オブジェクトには、コントロールに対応する子ウィンドウ オブジェクトがあります。ウィンドウやコントロールの名前は、テストするアプリケーションの内部メソッドやプロパティにテスト エンジンがアクセスできるかどうかによって依存します。TestComplete は、両方のタイプのアプリケーションで動作しますが、それらに異なる方法で名前を付けます。

この構造のオブジェクトの完全な名前は長く、読みづらいため、オブジェクトに短いカスタム名を割り当てることができます。 <C:\testing-with\object-identification\name-mapping\about.htm> そうすることで、テストが分かりやすく、保守しやすくなります。

- **ブラックボックス アプリケーション**

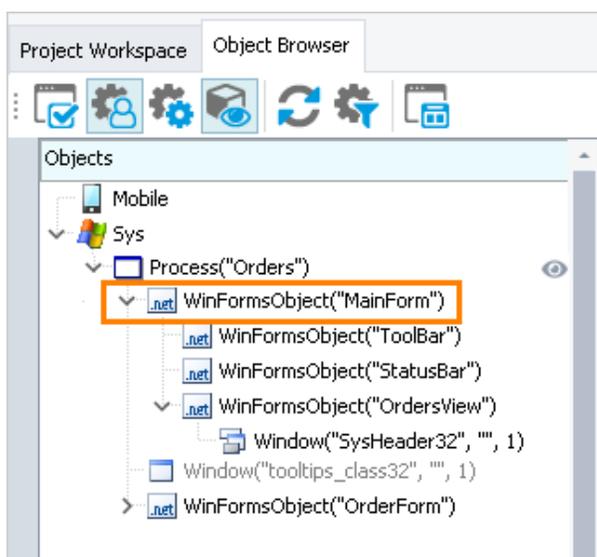
内部メソッドやプロパティにアクセスできないアプリケーションは**ブラックボックス アプリケーション**と呼ばれます。このようなアプリケーションのウィンドウの名前は、ウィンドウのクラス名、ウィンドウのテキストあるいはタイトル (キャプション)、インデックスから構成されます。オペレーティング システムの観点から言えば、コントロールはウィンドウの一種なので、コントロールもウィンドウと同様に名前が付けられます。



- **ホワイトボックス アプリケーション**

内部のオブジェクト、メソッド、プロパティに TestComplete がアクセスできるアプリケーションは、**ホワイトボックス アプリケーション**あるいは **Open アプリケーション**と呼ばれます。このアプリケーションには、Object Browser 内で  アイコンが表示されます (次に示す画像を参照)。

Open アプリケーションのウィンドウやコントロールに対して、TestComplete は ウィンドウやコントロールの種類を反映した名前およびアプリケーションのソースで定義した名前を付けます。たとえば、Microsoft WinForms ライブラリで作成された C# アプリケーションの MainForm という名前のフォームの場合、TestComplete は WinFormsObject("MainForm") という名前を付けます。



プロセス、ウィンドウ、コントロールの名前の付け方についての詳細は、TestComplete ヘルプの「Naming Objects」を参照してください。

注意: 可能な限り、テストにはブラックボックス アプリケーションではなく **Open** アプリケーションを使用することをお勧めします。**Open** アプリケーションで動作させることによって、テスト エンジンがアプリケーションの内部メソッドやプロパティにアクセスできるようになり、より強力で柔軟性のあるテストを作成することができます。

.NET、WPF、Visual Basic、Java、Web などのアプリケーションは、TestComplete では常に "Open" です。そのほかのアプリケーションは特別な方法でコンパイルが必要になることがあります。これについての詳細情報は、TestComplete ヘルプの「Open Applications」を参照してください。

チェックポイントと Stores

典型的なテストでは、たくさんの比較が実行されます。たとえば、アプリケーションのデータをファイルにエクスポートするユーザー アクションをシミュレートする場合、有効なデータがファイルに含まれるかどうかを確認する必要があります。この確認を実行するには、その基準のコピーと結果ファイルを比較します。これは、実行する必要がある比較の 1 つの例に過ぎません。実際のテストには、数千とまではいかなくても数百の比較が含まれます。すべてのテスト (リグレッション、単体、機能など) は、オートメーション中に有効なリファレンスを必要とします。

TestComplete を用いることで、テストに比較コマンド (あるいは**チェックポイント**) を簡単に追加することができます。テストのレコーディング中およびデザイン時の両方でチェックポイントを作成することができます。TestComplete は、異なる種類のデータ (画像、ファイル、オブジェクト テキストやプロパティ、XML ドキュメント、データベース テーブルなど) を比較するチェックポイントを提供します。TestComplete は、これらのチェックポイントで用いる基準データを格納する **Stores** プロジェクト項目を含んでいます。このプロジェクト項目は、プロジェクトと一緒に保存される比較用の画像、ファイル、その他の要素を格納します。唯一の例外は、オブジェクト プロパティを検証するチェックポイントです (このオブジェクト プロパティの基準データはテスト内で指定されます)。

チェックポイントと検証用コードの作成に関する詳細は、TestComplete ヘルプの「About Checkpoints」を参照してください。

デスクトップアプリケーションのテスト

TestComplete は、C++、C#、VB.NET、Java、Delphi、C++ Builder およびその他多くの開発ツールで作成された 32 ビットおよび 64 ビット Windows デスクトップアプリケーションのテストをサポートします。

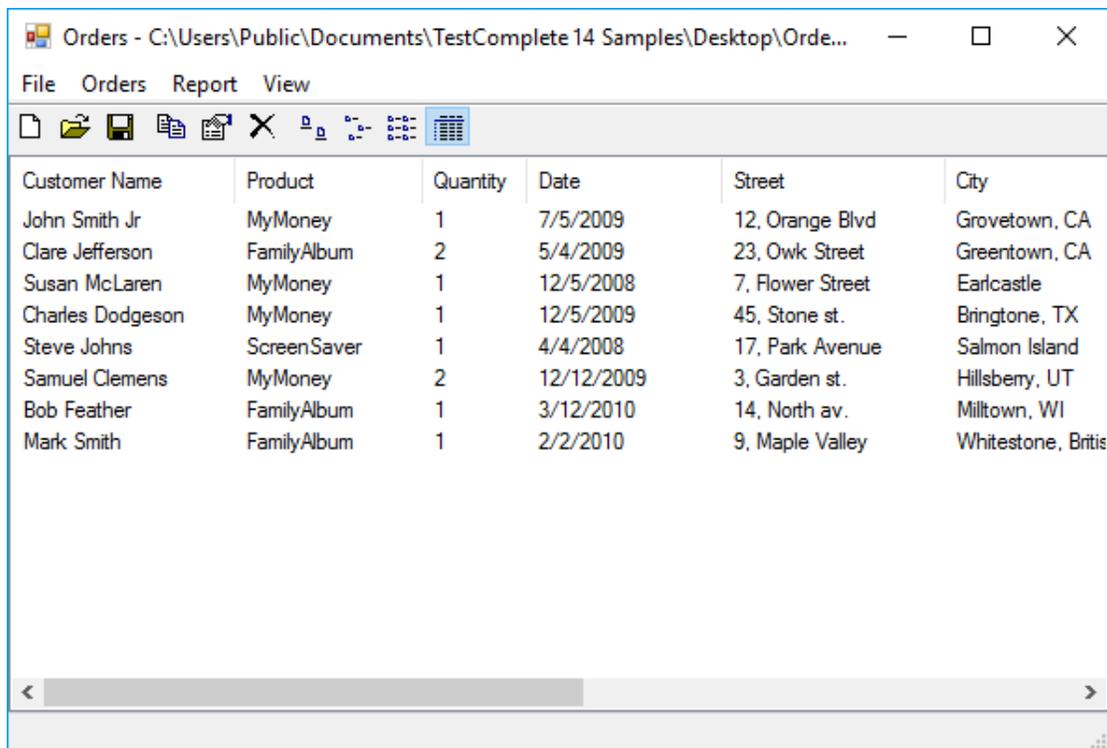
このチュートリアルは、デスクトップアプリケーション (デスクトップマシンで実行するアプリケーション) のテストの基本を説明します。これは、自動テストの一般原則に精通しており、TestComplete IDE についての最低限の知識を持っていることを前提としています。

! 初心者の方は、上記の自動テストの紹介を読むことをお勧めします。

このチュートリアルのセクションには、TestComplete でテストプロジェクトを作成し、簡単なテストをレコーディングおよび再生実行して、結果を分析する方法の説明が含まれています。このテストは、テストするアプリケーション上のユーザーアクションをエミュレートし、データを検証します。検証用コマンドはテストレコーディング中に作成されます。

テストするアプリケーションについて

この説明では、TestComplete に含まれるサンプルの *Orders* アプリケーションを使用します。このアプリケーションは注文 (orders) のリストを表示し、注文を追加、削除、編集、エクスポートするための特別な機能を持っています。



The screenshot shows a window titled "Orders - C:\Users\Public\Documents\TestComplete 14 Samples\Desktop\Orde...". The window contains a table with the following data:

Customer Name	Product	Quantity	Date	Street	City
John Smith Jr	MyMoney	1	7/5/2009	12, Orange Blvd	Grovetown, CA
Clare Jefferson	FamilyAlbum	2	5/4/2009	23, Owk Street	Greentown, CA
Susan McLaren	MyMoney	1	12/5/2008	7, Flower Street	Earlcastle
Charles Dodgeson	MyMoney	1	12/5/2009	45, Stone st.	Bringtone, TX
Steve Johns	ScreenSaver	1	4/4/2008	17, Park Avenue	Salmon Island
Samuel Clemens	MyMoney	2	12/12/2009	3, Garden st.	Hillsbery, UT
Bob Feather	FamilyAlbum	1	3/12/2010	14, North av.	Miltown, WI
Mark Smith	FamilyAlbum	1	2/2/2010	9, Maple Valley	Whitestone, Britis

このアプリケーションは下記の場所にインストールされます。

`C:\Users\Public\Public Documents\TestComplete 15 Samples\Desktop\Orders`

注意: ファイル マネージャーによっては、Public Documents フォルダを Documents として表示するものがあります。

ここでは、Visual C# で作成された Orders アプリケーションを使用します。

-  他のコンパイラ (Delphi、C ++ Builder、Swing など) で作成された Orders プロジェクトの場合は、Web サイトの <https://support.smartbear.com/downloads/testcomplete> ページから TestComplete Samples インストール パッケージをダウンロードして実行してください。

1. テストの計画

サンプルの Orders アプリケーションは、注文リストを管理します。ここでは、アプリケーションの Edit Order フォームが正しく機能し、注文リストのデータが変更されるかどうかをテストします。この場合、次のように定義します。

- **テストの目的:** このテストは、Edit Order フォームで変更されたデータが保存され、その変更内容が注文リストに反映されるかどうかをチェックします。
- **テストの手順:** テストでは、注文の詳細情報の変更、注文リスト内のデータの検証をシミュレートします。アプリケーション上のユーザー アクションをシミュレートしたテストをレコーディングします。簡単にするために、このテストでは1つの注文の1つのプロパティのみを "変更" します。
- **チェックおよびテスト結果の記録:** 注文への変更が正しく保存されたのであれば、注文リストにその変更内容が反映されるべきです。これをチェックするために、今回のテストはリスト内のデータと期待される値を比較します。テストに特別な比較コマンドを追加します。このコマンドは、テスト ログに比較結果を出力するので、これによって検証が成功したか失敗したかを確認することができます。

TestComplete を使用したテストの計画についての詳細は、TestComplete ヘルプの「Planning Tests」を参照してください。

2. レコーディングについて

TestComplete では、2 通りの方法でテストを作成することができます。

- **テストを手動で作成する** - スクリプト オブジェクトまたはキーワード テスト コマンドを使用して、必要なすべてのコマンドとアクションを入力します。このアプローチは、強力で柔軟なテストを作成する必要があるとき、またはテストの作成で豊富な経験がある場合、役に立ちます。しかし、手動でテストを作成することは、多くの時間を必要とし、別の問題が生じる可能性があります。たとえば、使用するアプリケーション オブジェクトのクラスや名前を知っていなければなりません。
- **テストを記録する** - テストを簡単に作成することができます。一度、テストするアプリケーションに対していくつかのアクションを実行すると、TestComplete は自動的にこれらのアクションを認識し、それらをスクリプトまたはキーワード テスト操作に変換します。テストをビジュアルに作成し、実行したアク

ションをスクリプトまたはキーワード テストに記録できます。これは、非常に役に立つアプローチであり、テストの作成であまり経験を必要としません。

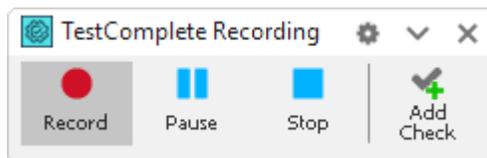
このチュートリアルでは、TestComplete でテストを記録する方法を説明します。

レコーディングは次の 3 つの手順を含みます。

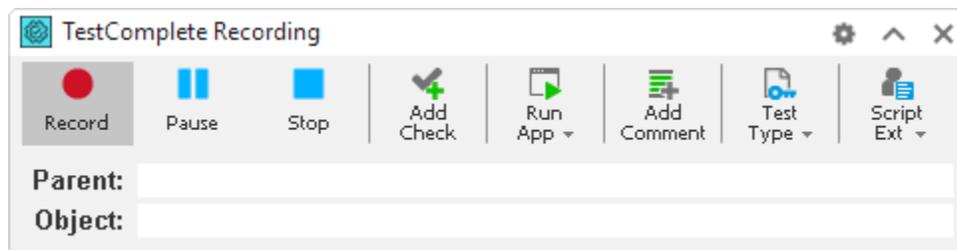
1. レコーディングを開始します。TestComplete のメイン メニューあるいは [Test Engine] ツールバーから **[Test] > [Record] > [Record Keyword Test]** または、**[Test] > [Record] > [Record Script]** を選択してレコーディングを開始します。スタートページの **● [Record Test]** をクリックして、レコーディングを開始することもできます。

さまざまな種類のテスト (キーワード テスト、スクリプト、低レベルのプロシージャ) をレコーディングすることができます。レコーディングの開始に使用するメニュー項目によって、メインのレコーディングするテスト (キーワード テストまたはスクリプト コード) が定義されます。そのほかのテストはそのレコーディングが開始された後でレコーディングされます。メインのレコーディングするテストには、これらのテストを実行する特別なコマンドが含まれます。

TestComplete は、レコーディング モードに切り替わり、画面上に [Recording] ツールバーが表示されます。デフォルトでは、ツールバーは折りたたまれ、レコーディング中に必要となるよく利用されるコマンドだけが表示されます。



[Recording] ツールバーを展開するには、▼ 矢印ボタンをクリックします。すると、すべてのボタンが表示されます。



このツールバーを使用して、レコーディング中に追加の動作を実行したり、レコーディングの一時停止や停止、レコーディングするテスト タイプ (キーワード テスト、スクリプト コード、低レベルのプロシージャ) を変更することができます。

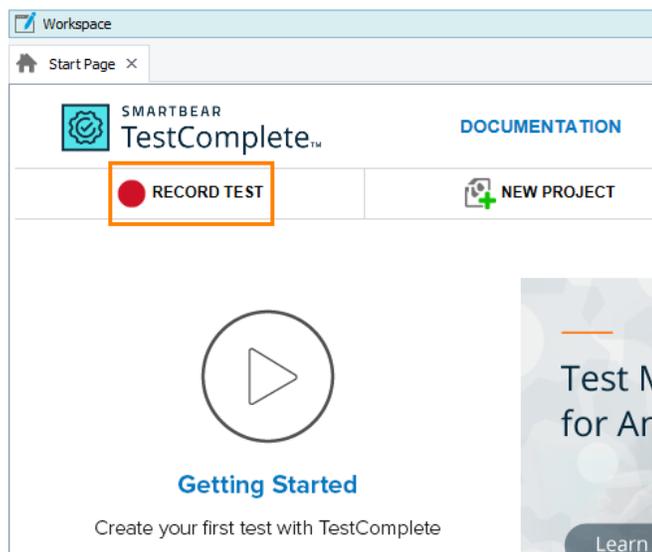
2. レコーディング開始後、必要なテスト動作を実行します。テストするアプリケーションを起動し (必要な場合)、コマンド ボタンをクリックしたり、メニュー項目を選択したり、テキストを入力するなどして使用します。
3. すべてのテスト動作が終了した後、[Recording] ツールバーから **■ [Stop]** を選択し、レコーディングを停止します。

テストのレコーディングに関する詳細は、TestComplete ヘルプの「*Recording in TestComplete*」を参照してください。

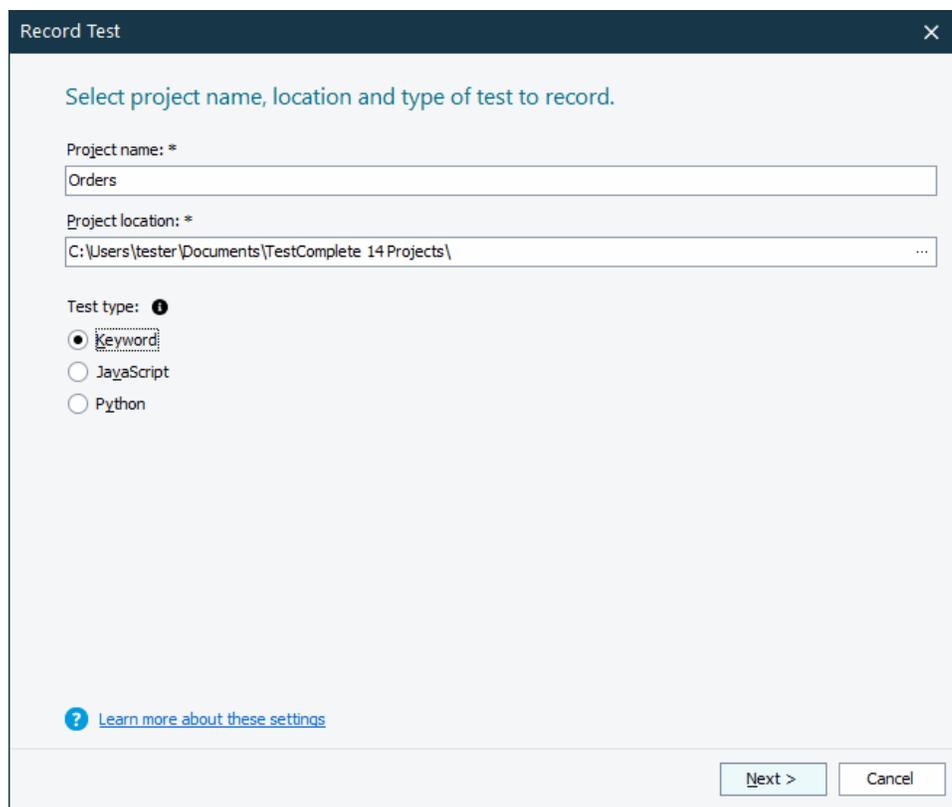
最初のテストのレコーディング方法は、次の「3. テストのレコーディング開始」を参照してください。

3. テストのレコーディング開始

1. TestComplete でプロジェクトあるいはプロジェクト スイートを開いている場合は、閉じてください。これをするには、**[File] > [Close]** を TestComplete のメインメニューから選択します。
2. TestComplete **スタート** ページに切り替えます。このページが隠れている場合、TestComplete の **[Help]** メニューから **[Start Page]** を選択します。



3. スタートページで、● **[Record Test]** をクリックします。TestComplete は、**[Record Test]** ウィザードを表示します。



ウィザードの最初のページで、プロジェクトの名前、場所、テストタイプを指定します。

[Project name] テキストボックスに Orders と入力します。

[Location] テキストボックスはデフォルトの値のままにします。

テストタイプを選択します。キーワードテストまたは JavaScript、Python スクリプト テストのいずれかを作成することができます。

- キーワードテストは、マウスのクリック、テキストの入力など、ユーザーの操作を定義する一連のキーワードです。視覚的にキーワード テストを作成します。スクリプトの知識は必要ではありません。
- JavaScript と Python テストは、ユーザー アクションをシミュレートする命令を含むスクリプト関数です。

このチュートリアルでは、キーワード テストの作成方法を示します。 **[Keyword]** をクリックします。

4. **[Next]** をクリックします。

ウィザードを使って、テストするアプリケーションをプロジェクトに追加するため、ウィザードで作業を続行します。

4. テストするアプリケーションの定義

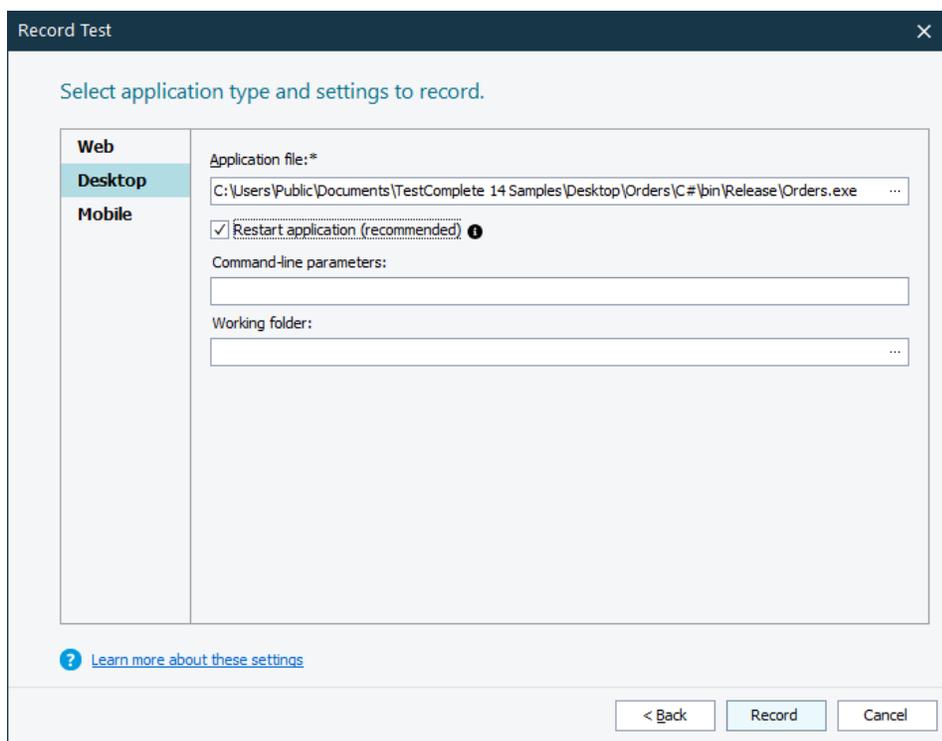
各 TestComplete プロジェクトには、テストするアプリケーションのリストが含まれます。これにより、プロジェクトで使用するアプリケーションとそのテスト設定を追跡できます。

プロジェクトにアプリケーションを追加する方法はいくつかあります。

- プロジェクトの作成中にこれを行うことができます。
- これは、後で Project Explorer でいつでも行うことができます。
- テストのレコーディング中に自動的に追加することができます。レコーダーはコマンドライン、エクスプローラーなどからのアプリケーションの開始を検出することが可能です。

テストする Orders アプリケーションをプロジェクトに追加します。

1. ウィザードでは、テストするアプリケーションを選択できる 2 番目のページが表示されます。



2. テストする Orderes アプリケーションは、スタンドアロンの実行ファイルとして実行される .NET アプリケーションなので、デスクトップ アプリケーションのカテゴリに該当します。

[Desktop] をクリックします。

3. **Application file** ボックスで、[...] ボタンをクリックします。表示される **[Select Tested Application]** ダイアログで、Orders 実行ファイルの場所を見つけます。以下のパスにあります。

C:\Users\Public\Public Documents\TestComplete 15 Samples\Desktop\Orders\C#\bin\Release\Orders.exe

注意: ファイル マネージャーによっては、Public Documents フォルダを Documents として表示するものがあります。

4. テストの実行を成功させるには、テスト実行の開始時およびテストのレコーディングの開始時に、テストするアプリケーションの状態が同じである必要があります。テストのレコーディングを開始するときにテストするアプリケーションが実行されている場合、その状態は後続のテスト実行中の状態と異なることがあります (たとえば、データが読み込まれているか、ダイアログが開いているなど)。そのため、テストするアプリケーションのすべての既存のインスタンスを閉じ、テストするアプリケーションの起動からテストをレコーディングすることを推奨します。

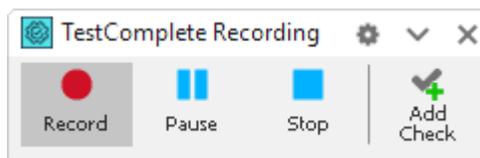
TestComplete が Orders アプリケーションの既存のインスタンスをすべて自動的に再起動するようにするには、[Restart application] チェックボックスを選択します。

5. [Record] をクリックし、プロジェクトの作成を完了し、レコーディングを開始します。[Record] をクリックし、プロジェクトの作成を完了し、レコーディングを開始します。

5. テストのレコーディング

! レコーディング中に TestComplete ヘルプに切り替えしないでください。レコーディング エンジンではすべてのユーザー アクションをトレースし、レコーディングします。つまり、レコーディングしたテストに、この "切り替え" をシミュレートするコマンドが含まれてしまいます。操作手順は、レコーディングを開始する前に印刷しておくことができます。あるいは、2 つのモニターがある場合、TestComplete ヘルプをもう一方のモニターへ移動しておくことができます。

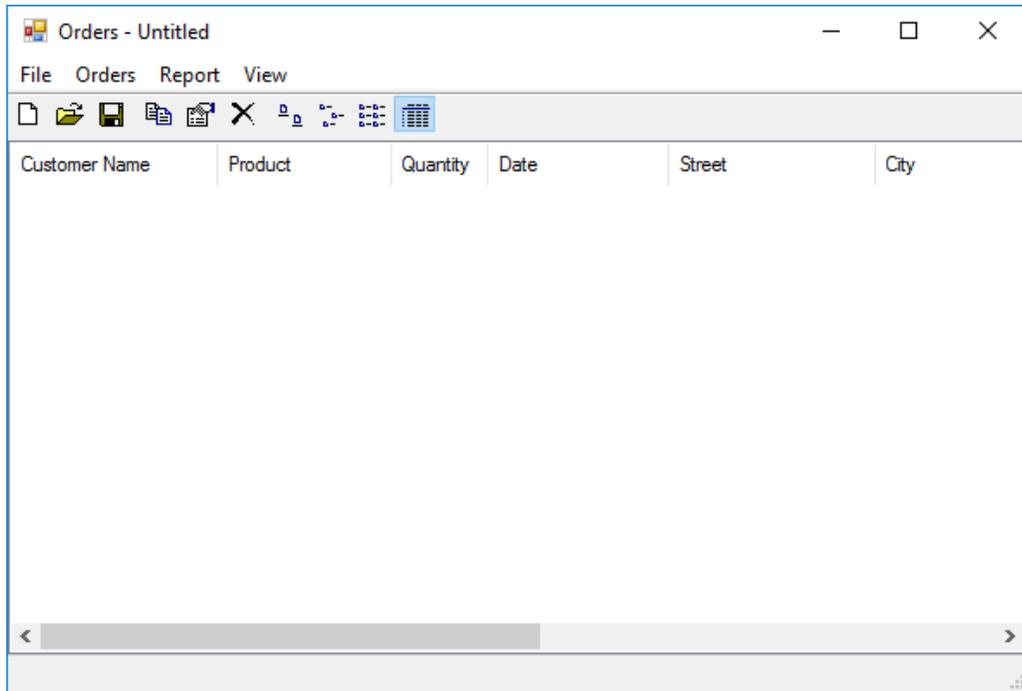
1. TestComplete は、テストのレコーディングを開始し、レコーディング モードに切り替わり、画面上に [Recording] ツールバーが表示されます。



2. レコーディングが開始されると、TestComplete は自動的にテストするアプリケーション Orders を起動します。アプリケーションが起動しない場合は、[Recording] ツールバーの [Run App] メニューからアプリケーションを選択して手動で起動できます。または、Windows Explorer や任意のファイル マネージャーからアプリケーションを起動できます。テストするアプリケーションがリストにない場合は、追加することができます。

TestComplete は、特別なアプリケーションの起動テストコマンドを使用して、アプリケーションの開始を記録します。後でレコーディングしたテストを分析するときに、このコマンドを参照します。

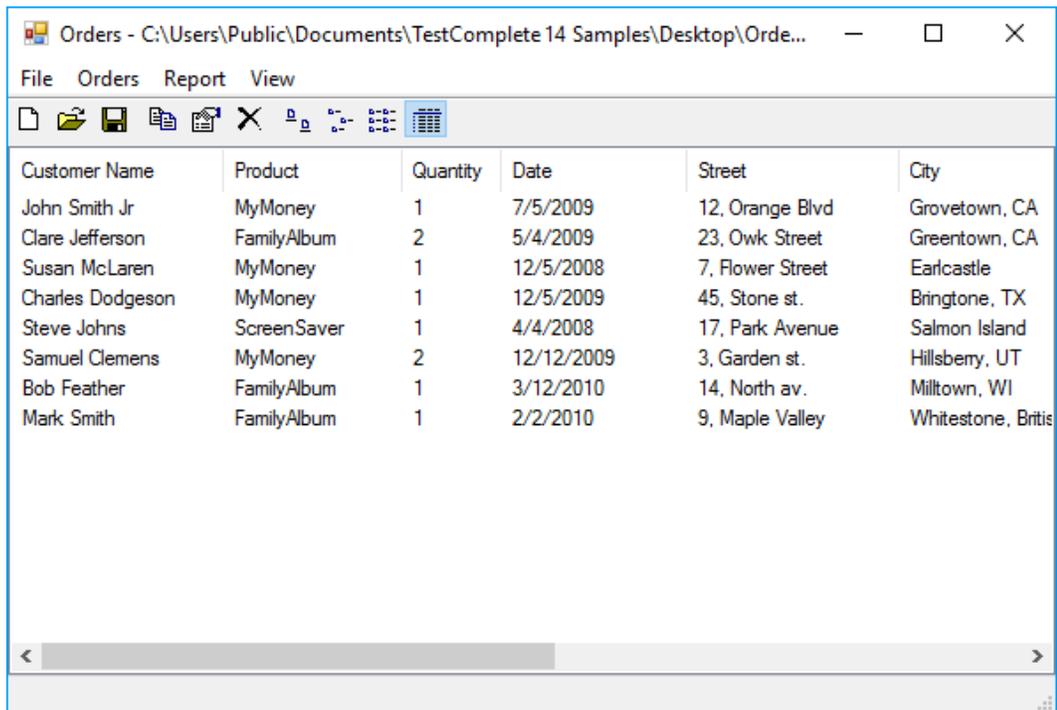
3. アプリケーションが開始するまで待ちます。アプリケーションのメイン ウィンドウが表示されます。



4. Orders アプリケーションに切り替えて、そのメイン メニューから **[File] > [Open]** を選択します。
[ファイルを開く] ダイアログが表示されます。
5. このダイアログで、*MyTable.tbl* ファイルを開きます。これは、*C:\Users\Public\Public Documents\TestComplete 15 Samples\Desktop\Orders* フォルダにあります。ファイル マネージャーによっては、Public Documents フォルダを Documents として表示するものがあります。

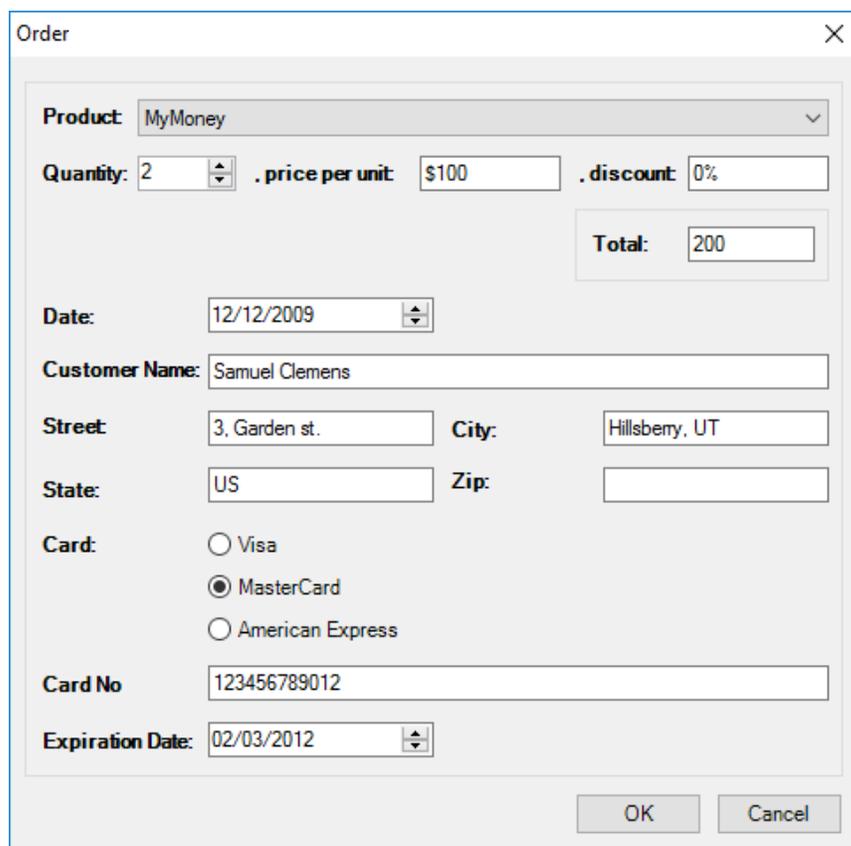
! [ファイルを開く] ダイアログの **[ファイル名]** ボックス内には省略していないファイル名を入力することをお勧めします。マウスを使用する代わりにキーボード入力することで、異なるオペレーティング システムでテストを再生実行したり、後でテストを実行したときに [ファイルを開く] ダイアログで異なる初期フォルダが表示された場合に問題を回避することができます。

6. **【ファイル名】** ボックス でファイルを指定した後、**【開く】** をクリックします。Orders アプリケーションは、そのファイルからデータをロードし、アプリケーションのメイン ウィンドウにデータを表示します。



7. 注文リストの *Samuel Clemens* 行をクリックします。

8. [Orders] ツールバーで、 [Edit order] をクリックします。[Orders] ダイアログが表示されます。



The screenshot shows a dialog box titled "Order" with a close button (X) in the top right corner. The dialog contains the following fields and controls:

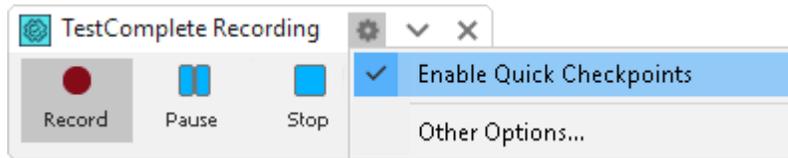
- Product:** A dropdown menu showing "MyMoney".
- Quantity:** A spinner box with the value "2".
- . price per unit:** A text box with the value "\$100".
- . discount:** A text box with the value "0%".
- Total:** A text box with the value "200".
- Date:** A date picker showing "12/12/2009".
- Customer Name:** A text box containing "Samuel Clemens".
- Street:** A text box containing "3, Garden st.".
- City:** A text box containing "Hillsbery, UT".
- State:** A text box containing "US".
- Zip:** An empty text box.
- Card:** Radio buttons for "Visa", "MasterCard" (which is selected), and "American Express".
- Card No:** A text box containing "123456789012".
- Expiration Date:** A date picker showing "02/03/2012".
- Buttons:** "OK" and "Cancel" buttons at the bottom right.

9. ダイアログで、挿入ポイントを [Customer Name] ボックスにするため、[Customer Name] ボックス内をクリックします。[Customer Name] ボックス内を右クリックして、コンテキストメニューから **[すべて選択]** を選択した後に顧客名として Mark Twain と入力します。
10. [OK] をクリックしてこのダイアログを閉じます。アプリケーションのメイン ウィンドウ内の顧客リストが更新されます。
11. テストに比較コマンドを挿入します。この比較コマンドは、アプリケーションの顧客リストに変更した名前 (*Mark Twain*) が表示されることを検証します。

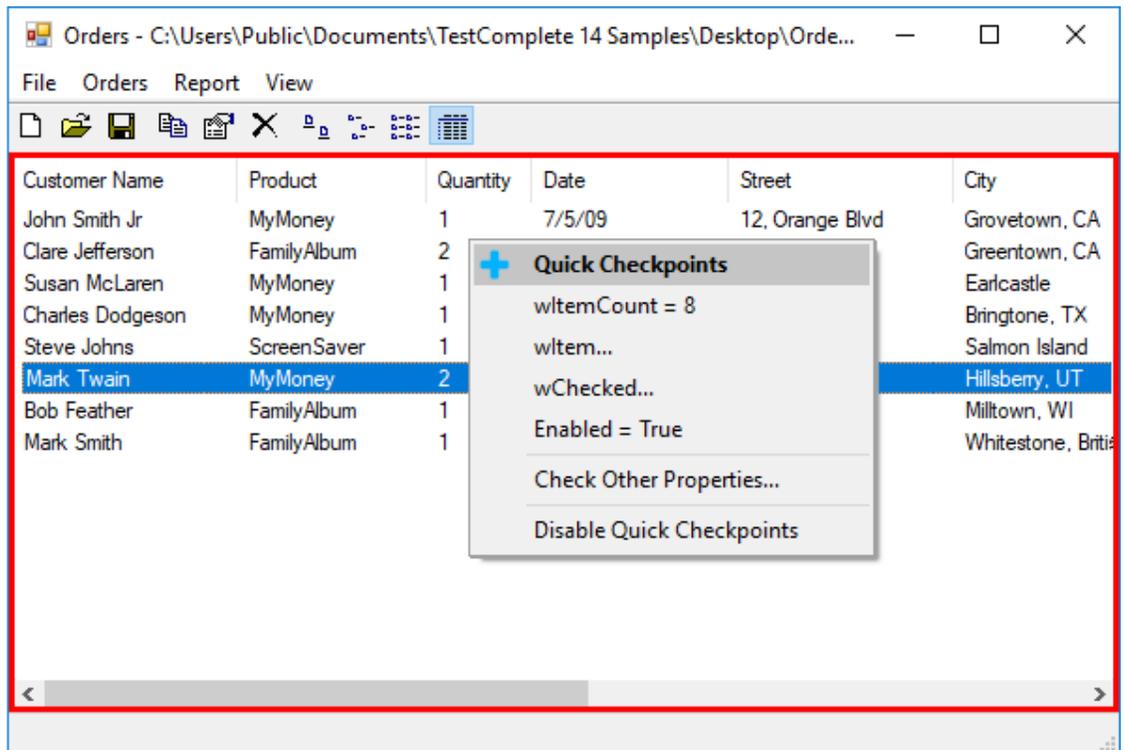
比較コマンドを**チェックポイント**と呼びます。TestComplete には、異なる種類のデータの検証に適したさまざまな種類のチェックポイントが用意されています (TestComplete ヘルプの「Checkpoints」を参照)。最もよく使用されるチェックポイントの 1 つは、**プロパティ チェックポイント**です。アプリケーションのコントロールのデータのチェックに使用されます。このチュートリアルでは、このプロパティチェックポイントを使用します。

プロパティチェックポイントを作成するには、**Create Checkpoint** ウィザードを使用するは、**Quick Checkpoint** を作成します。このチュートリアルでは、Quick Checkpoints の作成方法を示します。

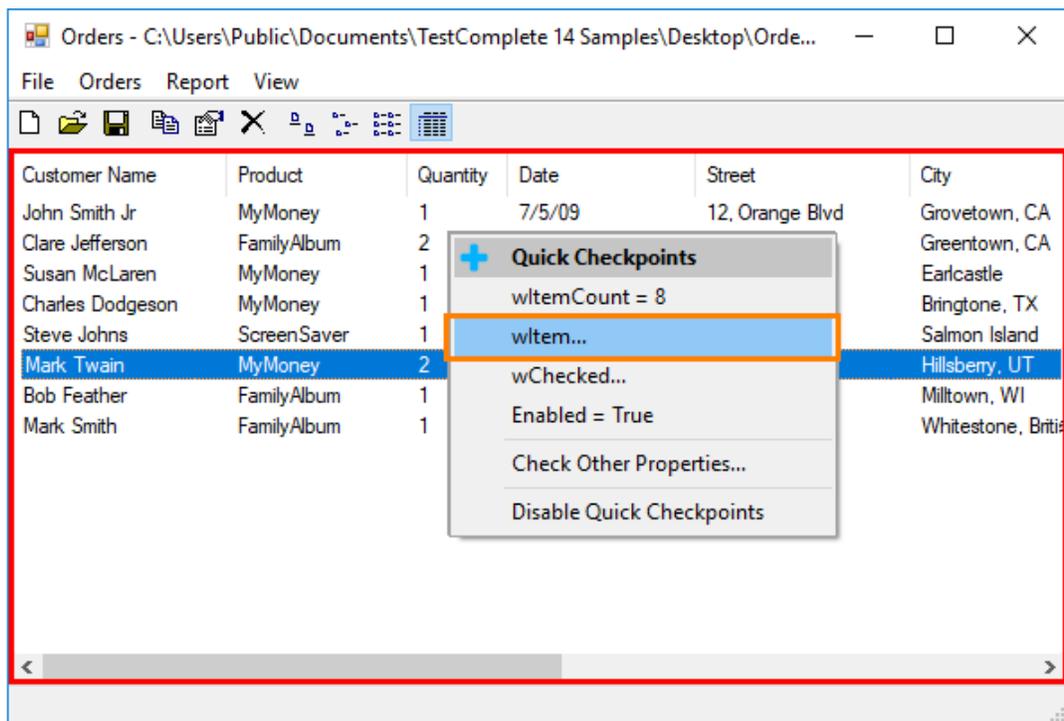
- **[Recording]** ツールバーの  をクリックし、Quick Checkpoints が有効になっていることを確認します。



- マウス ポインターを顧客リストの上に移動します。TestComplete はそれを赤いフレームでハイライトします。+ アイコンが不透明なるまで待つ、マウス ポインターをそれに移動します。TestComplete は、Quick Checkpoints を作成するために最も一般的に使用されるプロパティのリストを表示します。

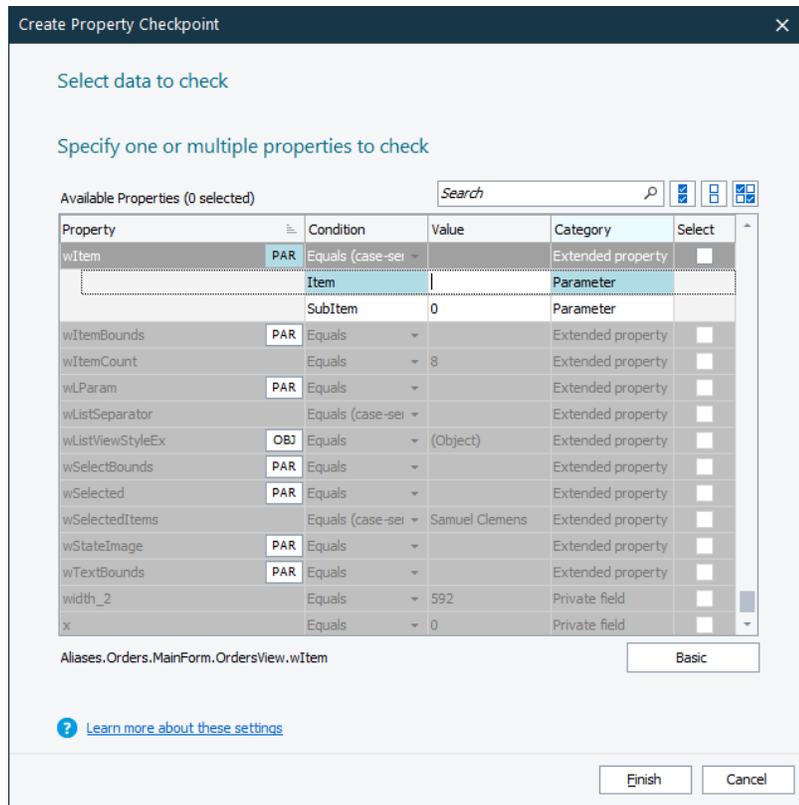


- そのリストにある wItem をクリックします。このプロパティでは、ツリー表示コントロールの個々の項目にアクセスできます。



- TestComplete は、オブジェクトプロパティのリストを表示する **Create Checkpoint** ウィザードを開きます。このリストには、テストするアプリケーションによって定義されたプロパティと TestComplete によって提供されるプロパティが含まれます。たとえば、テストするアプリケーションが C# で作成されている場合は、該当する .NET クラスのプロパティが含まれます。これらは、.NET カテゴリです (**Available Properties** テーブルを参照)。

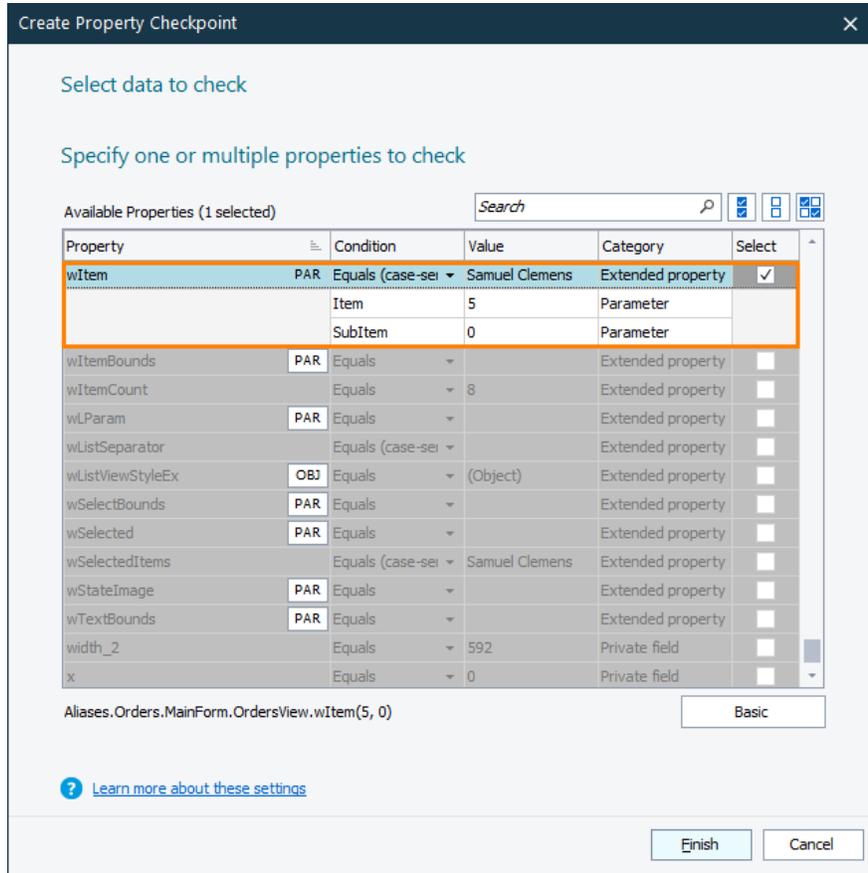
クリックした wItem プロパティがハイライトされます。



- Mark Twain 文字列を保持しているセルを指定するには、**Item** ボックスに **5** を入力します (5 はツリー表示での Mark Twain 項目のインデックスです。インデックスはゼロベースです。) **0** を **SubItem** ボックスに入力します。

テストエンジンによって項目のデータが読み取られ、プロパティリスト内に表示されます。

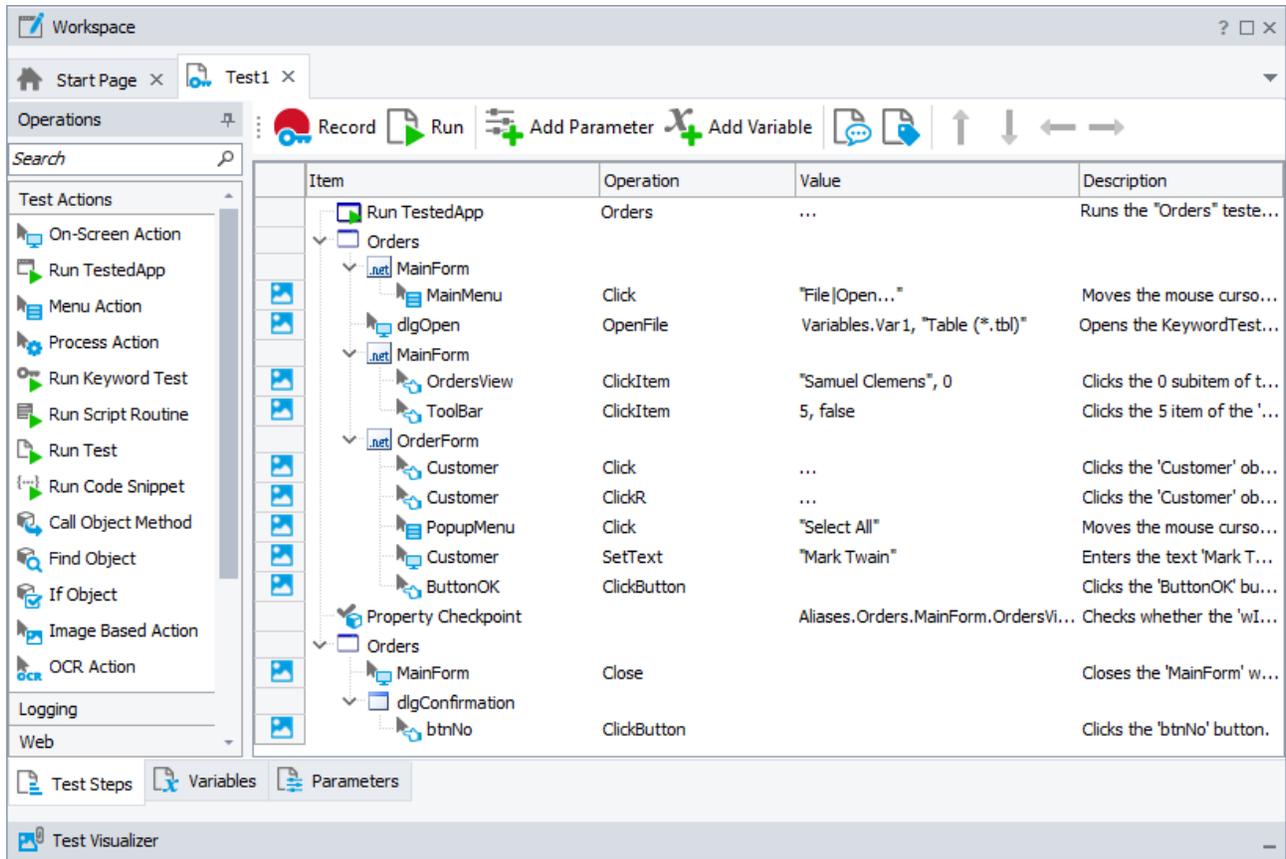
- **[Condition]** カラムで、デフォルトの比較条件 Equals (大文字と小文字を区別) をそのままにします。
- **[Value]** カラムには、テスト実行中にチェックポイントが実際のデータを比較する基準となるデータを指定します。現在の値をそのままにします。
- プロパティのチェックボックスを選択します。



- **[Finish]** をクリックしてチェックポイント作成を完了します。
 - レコーディングしたテストにチェックポイントコマンドが追加され、そのことについて通知が表示されます。ユーザーアクションのレコーディングを継続できます。
12. ウィンドウのキャプション バーにある **[X]** ボタンをクリックして **[Orders]** ウィンドウを閉じます。変更を保存するかどうかを確認するダイアログが表示されます。**[いいえ]** をクリックします。
 13. **[Recording]** ツールバーの **[Stop]** を押して、レコーディングを停止します。レコーディングしたテストコマンドが処理され、テストに保存されます。

6. レコーディングしたテストの分析

レコーディング終了後、TestComplete はレコーディングしたキーワード テストを編集用に関き、Keyword Test エディターにテストの内容を表示します。

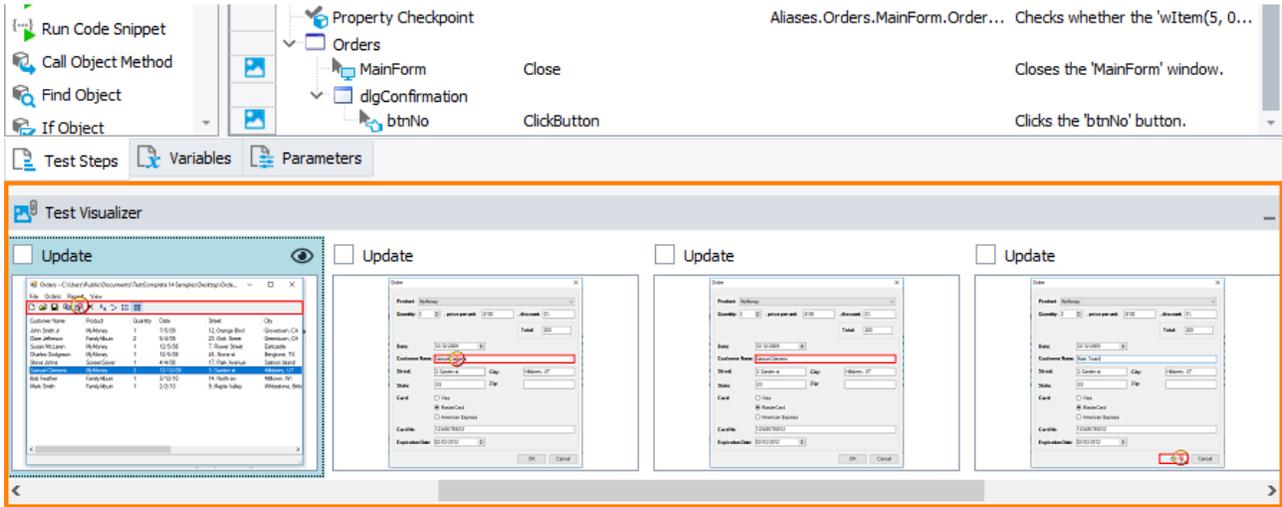


Item	Operation	Value	Description
Run TestedApp	Orders	...	Runs the "Orders" teste...
Orders			
MainForm			
MainMenu	Click	"File Open..."	Moves the mouse curso...
dlgOpen	OpenFile	Variables.Var1, "Table (*.tbl)"	Opens the KeywordTest...
MainForm			
OrdersView	ClickItem	"Samuel Clemens", 0	Clicks the 0 subitem of t...
ToolBar	ClickItem	5, false	Clicks the 5 item of the '...
OrderForm			
Customer	Click	...	Clicks the 'Customer' ob...
Customer	ClickR	...	Clicks the 'Customer' ob...
PopupMenu	Click	"Select All"	Moves the mouse curso...
Customer	SetText	"Mark Twain"	Enters the text 'Mark T...
ButtonOK	ClickButton		Clicks the 'ButtonOK' bu...
Property Checkpoint		Aliases.Orders.MainForm.OrdersVi...	Checks whether the 'wI...
Orders			
MainForm	Close		Closes the 'MainForm' w...
dlgConfirmation			
btnNo	ClickButton		Clicks the 'btnNo' button.

レコーディングしたテストは上記の画像に示すテストのようになります。実行環境によってはこれとは異なることがあります。たとえば、C++ Builder または Delphi アプリケーションでテストをレコーディングした場合は、ほかのオブジェクト名やウィンドウ インデックスを含んでいる可能性があります。

このテストには、レコーディング中に Orders アプリケーション上で実行した動作に対応するコマンドが含まれています。このテストコマンドを **operation** と呼びます。

コマンドの下に **[Test Visualizer]** パネルがあり、TestComplete によってテストのレコーディング中に operation でキャプチャーされた画像が表示されます。



これらの画像は、レコーディングした operation を表し、operation が実行する動作を理解するのに役立ちます。TestComplete は、ユーザー アクション (マウス クリック、テキスト入力など) に対応した operation の画像のみをキャプチャーします。

エディター内でこの操作を選ぶと、Test Visualizer は自動的に適切な画像を選択するため、operation が実行される前にアプリケーションの状態を簡単に見ることができます。画像をさらに詳しく見たい場合は、[Test Visualizer] パネルでその画像をダブルクリックします。

画像の作業についての詳細は、TestComplete ヘルプの「Test Visualizer」を参照してください。

このテストの最初の operation は、**Run TestedApp** です。これは、テストするアプリケーション (今回の場合、Orders アプリケーション) を、キーワードテストから起動するのに使用します。TestComplete は、自動的にアプリケーションを起動するときに、あるいは [Recording] ツールバーやオペレーティング システムの UI からアプリケーションが起動されたことを検出すると、自動的にこの操作を記録します。

Item	Operation	Value	Description
Run TestedApp	Orders	...	Runs the "Orders" test...
Orders			
MainForm			
MainMenu	Click	"File Open..."	Moves the mouse curs...

次の operation は、**[File] > [Open]** メニュー項目の選択に対応します。

Item	Operation	Value	Description
Run TestedApp	Orders	...	Runs the "Orders" test...
Orders			
MainForm			
MainMenu	Click	"File Open..."	Moves the mouse curs...
dlgOpen	OpenFile	Variables.Var1, "Table (*.tbl)"	Opens the KeywordTest...
MainForm			

次の operation は、[ファイルを開く] ダイアログでファイルを開く動作をシミュレートします。

Item	Operation	Value	Description
Run TestedApp	Orders	...	Runs the "Orders" test...
Orders			
MainForm			
MainMenu	Click	"File Open..."	Moves the mouse curs...
dlgOpen	OpenFile	Variables.Var 1, "Table (*.tbl)"	Opens the KeywordTest..
MainForm			

特定のケースでは、TestComplete は、[ファイルを開く] ダイアログ コントロールを操作するときに行うアクションをシミュレートする一連の操作をレコーディングできます。

注意: [ファイルを開く] ダイアログのコントロールを使用してファイルをナビゲートする代わりに、このダイアログの **[ファイル名]** ボックスに開きたいファイルのフルネームを入力することを推奨します。そうすることで、オペレーティング システム、ダイアログで表示されるパス、利用可能なナビゲーション バーやパネルに関係なく、正常に実行されるテストを記録できます。

テストに [ファイルを開く] ダイアログに対する一連の操作が含まれる場合、手動でテストを編集して、これらの操作を **OpenFile** メソッドの呼び出しに置換することができます。

その後、アプリケーションのメイン ウィンドウと Order フォームでのユーザー アクションをシミュレートする operation が続きます。

Item	Operation	Value	Description
Run TestedApp	Orders	...	Runs the "Orders" teste...
Orders			
MainForm			
MainMenu	Click	"File Open..."	Moves the mouse curso...
dlgOpen	OpenFile	Variables.Var 1, "Table (*.tbl)"	Opens the KeywordTest...
MainForm			
OrdersView	ClickItem	"Samuel Clemens", 0	Clicks the 0 subitem of t..
ToolBar	ClickItem	5, false	Clicks the 5 item of the '...
OrderForm			
Customer	Click		Clicks the 'Customer' ob..
Customer	ClickR	...	Clicks the 'Customer' ob..
PopupMenu	Click	"Select All"	Moves the mouse curso...
Customer	SetText	"Mark Twain"	Enters the text 'Mark T...
ButtonOK	ClickButton		Clicks the 'ButtonOK' bu...
Property Checkpoint		Aliases.Orders.MainForm.OrdersVi...	Checks whether the 'wi...
Orders			
MainForm	Close		Closes the 'MainForm' w...
dlgConfirmation			
btnNo	ClickButton		Clicks the 'btnNo' button.

マウス イベント、キーボード入力、スクリプトからのその他の動作についての詳細は、TestComplete ヘルプの「Simulating User Actions」を参照してください。

次は、テストレコーディング中に追加した比較動作です。

	Customer	SetText	"Mark Twain"	Enters the text 'Mark T...
	ButtonOK	ClickButton		Clicks the 'ButtonOK' b...
	Property Checkpoint		Aliases.Orders.MainForm.Orde...	Checks whether the 'w...
	Orders			
	MainForm	Close	...	Closes the 'MainForm' ...
	dlgConfirmation			
	btnNo	ClickButton		Clicks the 'btnNo' button.

最後に、Orders アプリケーションを閉じる動作とメッセージ ボックス内で [いいえ] ボタンを押した動作です。

	Property Checkpoint		Aliases.Orders.MainForm.OrdersVi...	Checks whether the 'wi...
	Orders			
	MainForm	Close		Closes the 'MainForm' w...
	dlgConfirmation			
	btnNo	ClickButton		Clicks the 'btnNo' button.

TestComplete は、自動的に operation を使用したプロセスやウィンドウに対応するグループにまとめます。グループ化することで、テスト構造を簡単に理解できるようにし、さらにアプリケーションのオブジェクト階層の情報を提供しています。

ここでは、1 つのプロセス (Orders) のユーザー アクションをレコーディングしました。そのため、"process" グループ ノードは、1 つだけです。このグループ ノードには、プロセス ウィンドウおよびコントロールでシミュレートしたすべての動作が含まれています。Orders プロセスのコントロールやウィンドウで実行した動作はいくつかの "window" グループ ノードにまとめられています。

Item	Operation	Value	Description
Run TestedApp	Orders	...	Runs the "Orders" teste...
Orders			
MainForm			
MainMenu	Click	"File Open..."	Moves the mouse curso...
dlgOpen	OpenFile	Variables.Var 1, "Table (*.tbl)"	Opens the KeywordTest...
MainForm			
OrdersView	ClickItem	"Samuel Clemens", 0	Clicks the 0 subitem of t...
ToolBar	ClickItem	5, false	Clicks the 5 item of the '...
OrderForm			
Customer	Click	...	Clicks the 'Customer' ob...
Customer	ClickR	...	Clicks the 'Customer' ob...
PopupMenu	Click	"Select All"	Moves the mouse curso...
Customer	SetText	"Mark Twain"	Enters the text 'Mark T...
ButtonOK	ClickButton		Clicks the 'ButtonOK' bu...
Property Checkpoint		Aliases.Orders.MainForm.OrdersVi...	Checks whether the 'wi...
Orders			
MainForm	Close		Closes the 'MainForm' w...
dlgConfirmation			
btnNo	ClickButton		Clicks the 'btnNo' button.

テストするプロセスやそのウィンドウ、およびコントロールの名前が、これまでの手順で [Object Browser] パネルに表示されていた名前と異なることに気が付きかもしれません。たとえば、Object Browser では、テストするプロセスは Process("Orders") という名前であるのに対し、テスト内では Orders と呼ばれます。また Object Browser で、メイン ウィンドウは WinFormsObject("MainForm") と呼ばれるのに対し、テストでは MainForm という名前です。

これには、論理的な理由があります。デフォルトでは、TestComplete は、テスト レコーディング中に使用したオブジェクトに対して自動的にカスタム ネームを生成し、使用します。カスタム ネームの生成および割り

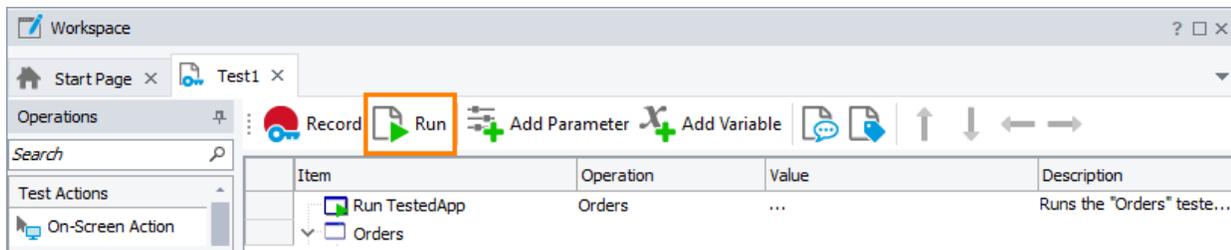
当ては、名前のマッピングと呼ばれます。デフォルトの名前は理解しにくいことがあるので、TestComplete では名前のマッピングを行っています。デフォルトの名前では、名前に対応するウィンドウやコントロールがどれなのかの判断が難しいことがあります。マッピングした名前はテストをよりわかりやすく、安定したものにします。名前のマッピングに加えて、TestComplete はマップされたオブジェクトのイメージを Name Mapping リポジトリに格納します。これは、1 つまたは別のマップされたオブジェクトが一致するウィンドウまたはコントロールを理解するのに役立ちます。名前のマッピングに関する詳細情報は、TestComplete ヘルプの「Name Mapping」を参照してください。

7. レコーディングしたテストの実行

TestComplete がユーザー アクションをどのようにシミュレートするか、ここで簡単なテストを実行してみましょう。

レコーディングしたテストを実行する前に、レコーディングしたときと同じ初期状態でテストが開始されることを確認してください。たとえば、ほとんどテストは、テストするアプリケーションの実行を必要とします。つまり、ユーザー アクションをシミュレートする前に、そのアプリケーションを起動する必要があります。今回の例では、テストするアプリケーションを起動するため、テストの始めに **Run TestedApp** operation を使用し、テストによってアプリケーションが起動されるようにしました。代替手段として、テストするアプリケーションを手動で TestComplete の IDE から実行することも可能です。

レコーディングしたテストを実行するには、テスト エディターのツールバーにある  **[Run Test]** をクリックします。



このテスト エンジンには TestComplete のウィンドウを最小化して、テスト コマンドの実行を開始します。このチュートリアルでのテストでは、レコーディングした動作が実行されます。

注意: テスト実行中は、マウスを動かしたり、キーを押したりしないでください。TestComplete でシミュレートされる動作を妨げたり、テスト実行が正しく行われない可能性があります。

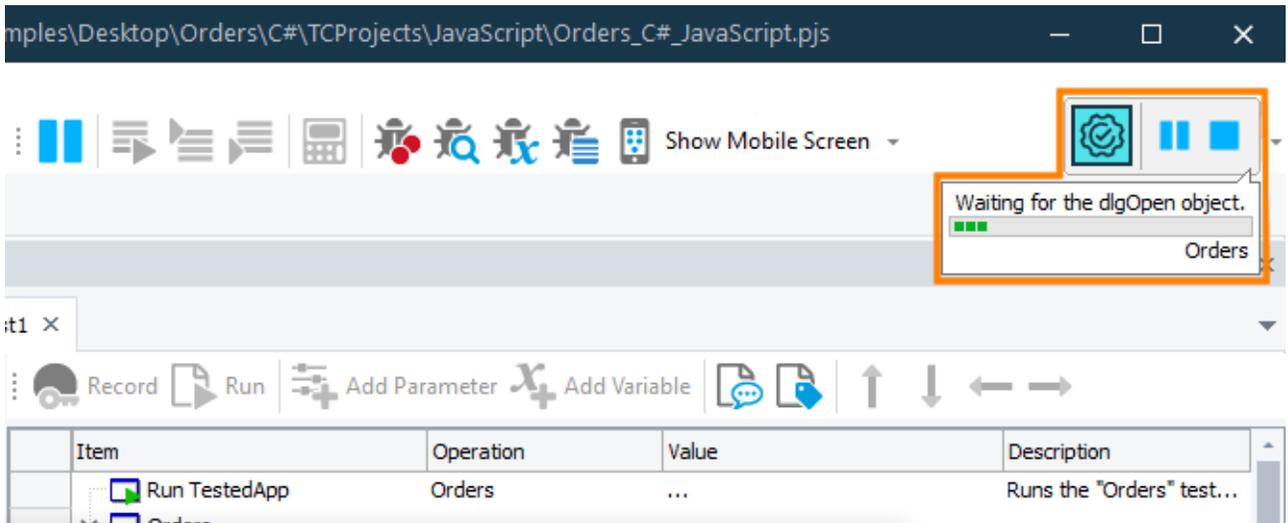
テスト実行の終了後、TestComplete はテスト結果を表示します。次の手順では、これを分析します。

テスト実行に関するいくつかの注意:

- 作成したテストは、テスト実行用の実行ファイルにはコンパイルされません。TestComplete から直接テストを実行します。TestComplete がインストールされていないマシンでテストを実行するには、TestExecute と呼ばれるリソース フレンドリーなユーティリティを使用することができます。また、外部のアプリケーションにスクリプトをエクスポートして、それを

実行することもできます。詳細については、TestComplete ヘルプの「Connected and Self-Testing Applications」を参照してください。

- テストの実行中、TestComplete は、画面の右上にインジケータを表示します。



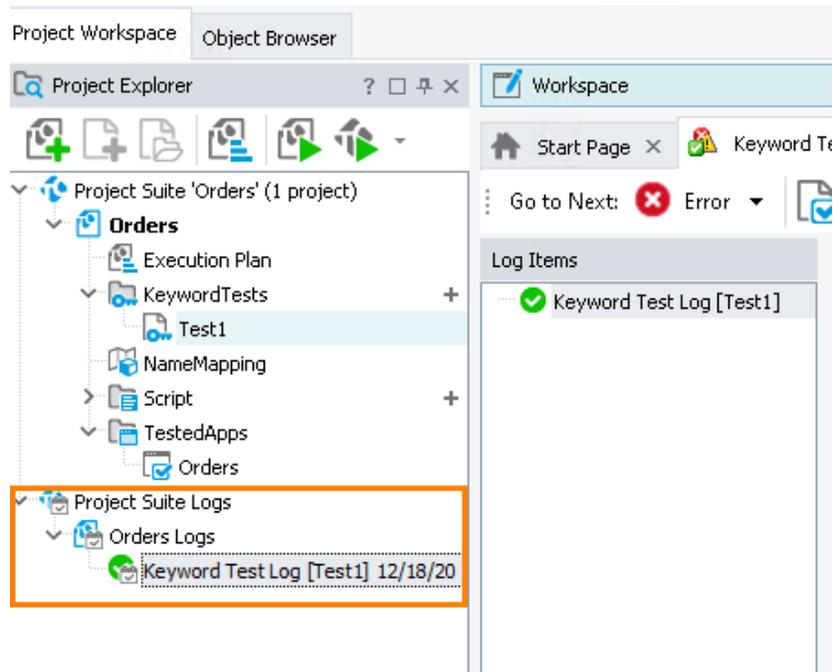
このインジケータは、シミュレートしたテスト動作に関する通知メッセージを表示します。

- TestComplete は、テストが終了するまでこのテスト コマンドを実行します。[Test Engine] ツールバーの ■ **[Stop]** を押すか、または TestComplete メインメニューから **[Test] > [Stop]** を選択することで、テスト実行をいつでも停止することができます。
 - **[Pause]** をクリックすることでテスト実行を一時停止することができます。一時停止の間は、必要な任意の動作を実行することが可能です。たとえば、テスト ログを表示したり、TestComplete の **[Watch List]** や **[Locals]** パネルあるいは **[Evaluate]** ダイアログを使用してテストの変数やオブジェクトを確認することが可能です (TestComplete ヘルプの「Debugging Tests」を参照)。
- テストを実行するには、テスト エディターのツールバーにある **[Run Test]** ボタンを使用します。このほかにも、いくつかの方法でテストを実行することができます。Project Explorer あるいは、ほかのテストから、テストを実行することも可能です。プロジェクト エディターのテスト項目ページを使用して、バッチの実行を作成することもできます。

TestComplete のテスト実行、実行に影響するプロジェクト設定、テスト実施に関する詳細は、TestComplete ヘルプの「Running Tests」を参照してください。

8. テスト結果の分析

TestComplete は、テスト中に実行されたすべての操作の完全なログを保存します。テスト結果へのリンクは、**[Project Explorer]** パネルの **[Project Suite Logs] > [Orders Log]** ノードに表示されます。これは、プロジェクトおよびプロジェクト スイートのテスト履歴を調べるための主なワークスペースです。各ノードは 1 つのテスト実行に対応します。ノードの左の画像は、対応するテストの実行が成功したかどうかを表しています。

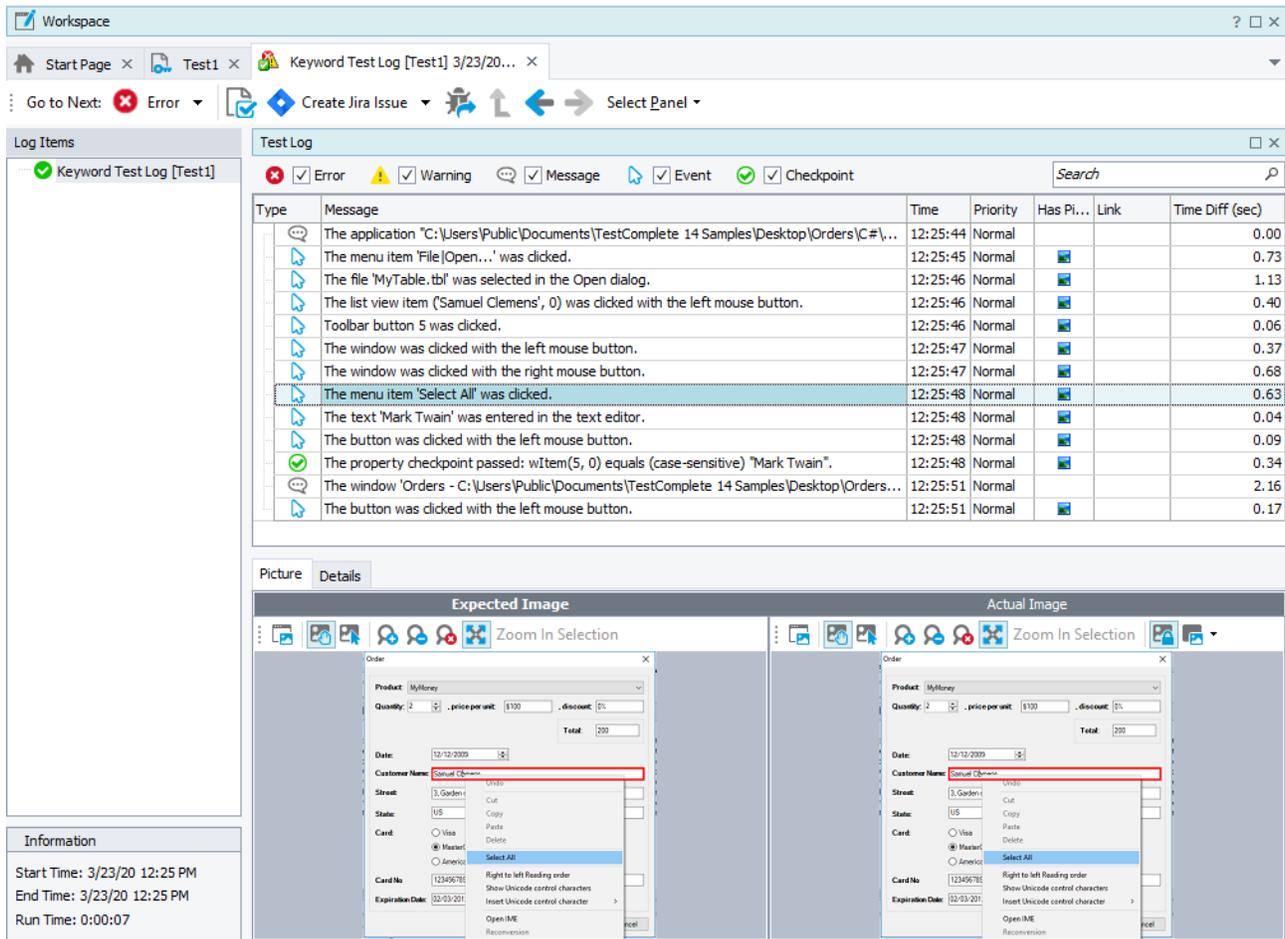


テスト実行の**終了後**、TestComplete は自動的に最終結果のノードを追加します。つまり、テストが実行中のときは、結果は表示されません (テストの実行を一時停止した場合は、中間結果を表示することができます)。

ここまでに実行したテストは 1 つのみなので、Project Explorer には 1 つのログ ノードだけが表示されます。デフォルトでは、**[Workspace]** パネルにこのノードの内容が自動的に表示されます。このログは、いつでも確認することができます。ログを見るには、**[Project Explorer]** パネルで表示したい結果を右クリックし、コンテキストメニューから **[Open]** を選択します。

注意: デフォルトで、TestComplete はログ ファイルにすべてのテスト結果を格納します。テスト実行ごとにログ ファイル数が増えて、メモリ消費量が増加します。メモリ使用量を減らすために、手動でログ ファイルを削除したり、保存するログ ファイルの数を制限することができます。

この例では、ログは次のようになります。



ログ ウィンドウは、1 度に 1 つのテスト実行の結果を表示します。ウィンドウの左側は、実行されたテストのツリー構造で、それぞれのテストのノードを選択してその結果を表示することができます。たとえば、今回の例のように 1 つのテストを実行した場合、このツリーには 1 つのノードのみ含まれます。このノードのアイコンは、テストが成功したかどうかを示します。

テスト ログには、エラー、警告、情報および他の種類のメッセージが含まれています。左のアイコンは、メッセージの種類を示します。メッセージ リストの上のチェック ボックスを使用することで、種類によってメッセージを表示/非表示にすることができます。

それぞれのメッセージでは、アクションが行われた時間が **Time** 列に表示されます。

TestComplete は、メッセージとともに追加テキストや画像を出力することがあります。これらを確認するには、ログ内でメッセージを選択し、メッセージ リストの下にある **[Details]** や **[Picture]** ペインを参照します。たとえば、上記画像の **[Picture]** ペインは、"The menu item 'Orders|Edit order...' was clicked" メッセージに関連するスクリーンショットを表示しています。

[Picture] パネルでは、選択したテスト コマンドを実行する前に、期待される状態と実際のアプリケーションの状態を示す画像が表示されます("期待される"とはレコーディング中にそのテストコマンドでキャプチャーされた画像を指し、"実際"とはテストの実行中にキャプチャーされた画像を指します)。画像の違いを見るた

め、**[View Comparison Result]** をクリックすることができます。そのため、テストでエラーが発生した場合、簡単に見つけることができます。詳細については、TestComplete ヘルプの「Test Visualizer」を参照してください。

ログにメッセージを出力したテスト操作を表示するには、ログ内でメッセージをダブルクリックします。すると、エディターでキーワードテストが開き、該当する操作がハイライトされます。たとえば、ログの "Toolbar button 5 was clicked" メッセージをダブルクリックした場合、この動作が行われたキーワードテスト操作がハイライトされます。

Item	Operation	Value	Description
Run TestedApp	Orders	...	Runs the "Orders" tested...
Orders			
MainApp			
MainMenu	Click	"File Open..."	Moves the mouse cursor ...
dlgOpen	OpenFile	Variables.Var1, "Table (*.tbl)"	Opens the KeywordTest...
MainForm			
OrdersView	ClickItem	"Samuel Clemens", ...	Clicks the 0 subitem of th...
ToolBar	ClickItem	5, false, ...	Clicks the 5 item of the T...
OrderForm			
Customer	Click	...	Clicks the 'Customer' obj...
Customer	ClickR	...	Clicks the 'Customer' obj...
PopupMenu	Click	"Select All"	Moves the mouse cursor ...
Customer	SetText	"Mark Twain"	Enters the text 'Mark Tw...
ButtonOK	ClickButton		Clicks the 'ButtonOK' butt...
Property Checkpoint		Aliases.Orders.MainForm.OrdersVie...	Checks whether the 'wit...
Orders			
MainForm	Close	...	Closes the 'MainForm' wi...
dlgConfirmation			
btnNo	ClickButton		Clicks the 'btnNo' button.

テスト ログ パネル、ログへメッセージを出力、その結果の使用に関する詳細情報は、TestComplete ヘルプの「Test Results」を参照してください。

注意: ここで説明したログは、TestComplete キーワードテストおよびスクリプト特有のログです。ほかの種類テストは異なった構造のログになります。ログに関する詳細情報は、適当なプロジェクト項目の解説を参照するか、そのログ ページ内をクリックして、F1 を押してください。

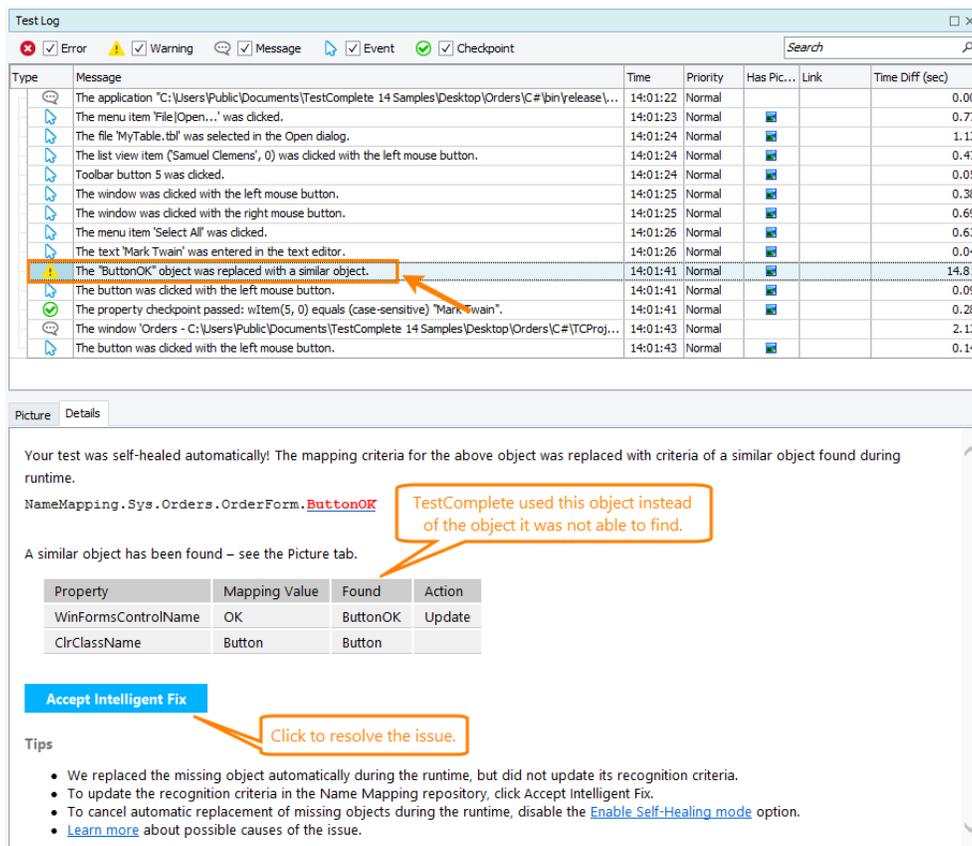
エラーの解決

テストが失敗することがあります。これにはいくつかの可能性がります。たとえば、開発者がアプリケーションの振る舞いを変更したり、ウィンドウとコントロールの認識属性の変更によりテスト エンジンが必要なオブジェクトの検出に失敗したり、サードパーティのアプリケーションがテストするアプリケーションのウィンドウに重なっているためにテスト エンジンがアクションのシミュレートに失敗するなどの可能性があります。

初めてのユーザーが直面する最も典型的なものは、テスト作成中と再生実行中でアプリケーションの状態が異なることです。この問題を避けるには、テストを作成したときと同じ初期状態でテスト実行しているか確認してください。たとえば:

- テストするアプリケーションがテストを記録する前に実行されていた場合は、テストを実行する前に実行している必要があります。
- テストアクションがアプリケーションの特定のウィンドウで実行された場合は、テストを実行するときにもそのウィンドウを開く必要があります。
- アプリケーション内のデータを編集して保存した場合は、変更を元に戻す必要があります。

テストするアプリケーションが変更され、テストするオブジェクトの検出に使用される識別プロパティが無効になると、TestComplete は不足オブジェクトを見つけるため変更の検出を試みます。テスト ログが "*Object <Object_Name> was replaced with a similar object*" 警告を報告した場合、TestComplete はテストの実行中にテストするオブジェクトを見つけることができず、代わりに類似するオブジェクトを使用したことを意味します。



この警告は無視しないでください。不足オブジェクトと代替オブジェクトを確認して、問題を解決するためオブジェクトの識別プロパティを更新してください。

エラーの原因を探し、典型的な問題を解決するための情報は、TestComplete ヘルプの「Handling Playback Errors」を参照してください。

Web アプリケーションのテスト - チュートリアル

TestComplete は、Internet Explorer、Edge、Chrome、Firefox、または組み込みの WebBrowser または Chromium で動作する Web アプリケーションの機能テストをサポートします。TestComplete は、Web アプリケーションのテストをサポートし、HTML5、AJAX、ASP.NET、Flash、Flex、AIR、および Silverlight テクノロジーのサポートを強化しています。

このセクションでは、TestComplete でテストプロジェクトを作成し、簡単な Web テストをレコーディング、再生実行し、結果を分析する方法について説明します。これは、自動テストの一般原則に精通しており、TestComplete IDE についての最低限の知識を持っていることを前提としています。

! 初心者の方は、上記の自動テストの紹介を読むことをお勧めします。

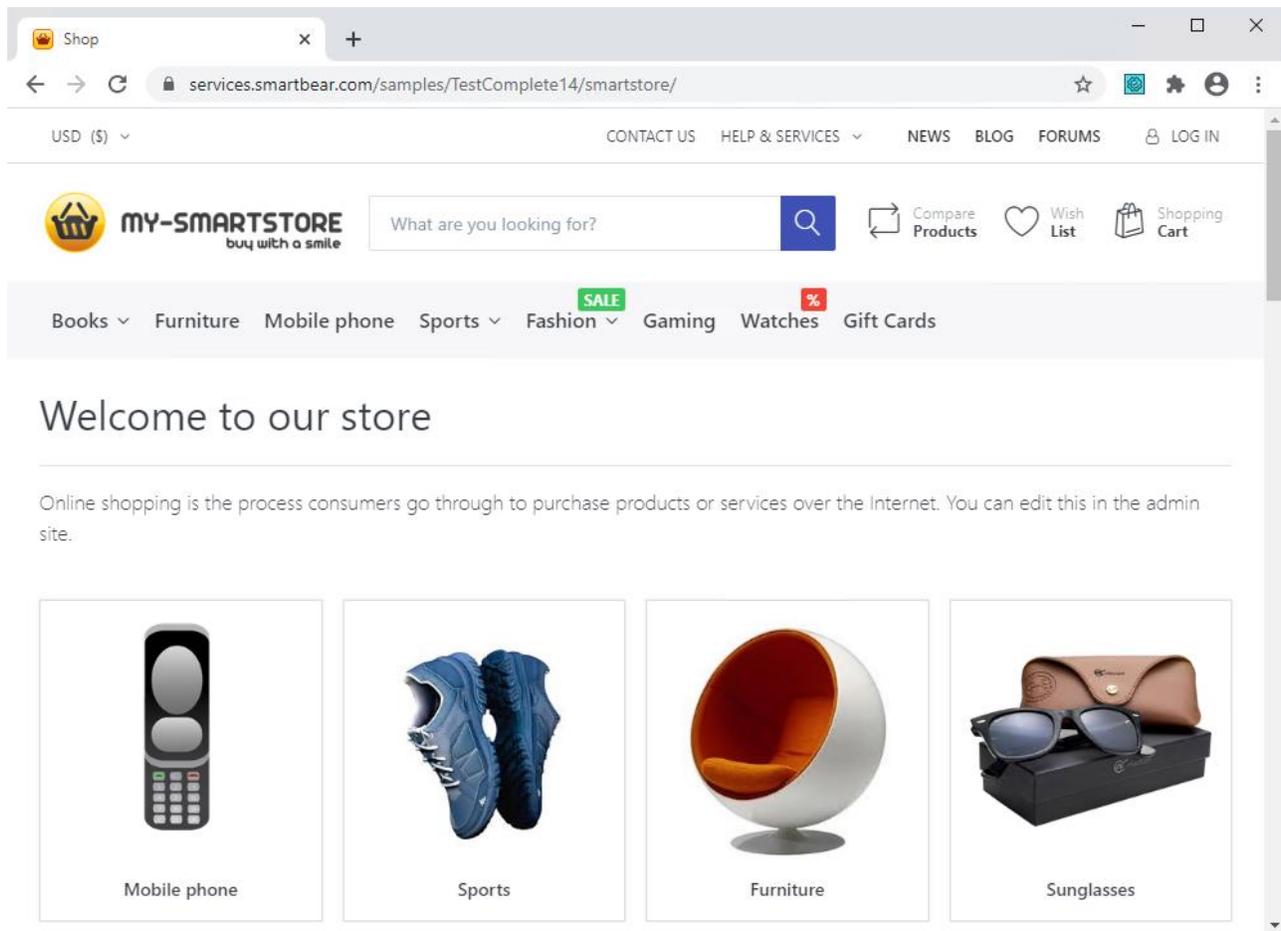
このテストは、Web ページ上のユーザー アクションをエミュレートし、データを検証します。検証用コマンドはテストレコーディング中に作成されます。

テストする Web ページについて

このチュートリアルでは、SmartBear 社の Web サイトにあるサンプルの *SmartStore* アプリケーションを使用します。

<https://services.smartbear.com/samples/TestComplete14/smartstore/>

このアプリケーションは、アイテムをブラウズしたり、ウィッシュリストとショッピングカートに追加したり、アイテムを比較したりできるサンプル オンラインストアです。アカウントを作成して、ログインし、モックアップオーダーを作成することができます。



注意: このチュートリアルでは、Google Chrome Web ブラウザーを使用します。チュートリアルのすべての画像は、テストするアプリケーションを Chrome で開いた状態で作成されています。サポートされている任意の Web ブラウザーを使用できます。

1. Web ブラウザーの準備

TestComplete で Web テストをレコーディングして再生実行するには、Web ブラウザーを特別な方法で設定する必要があります。また、クロスブラウザー テストを簡単にするため、ブラウザー固有の動作を排除することを推奨します。ブラウザーの設定についての詳細は、TestComplete ヘルプの「Preparing Web Browsers」を参照してください。

ブラウザーの設定が完了したら、テストを作成します。

2. テストの計画

このサンプル SmartStore アプリケーションは、オンライン ストアです。アプリケーションがショッピング カートにアイテムを正しく追加するかどうかをテストする必要があります。この場合、次のように定義する必要があります。

- **テスト目的:** テストでは、選択したアイテムがショッピング カートに正しく追加されているかどうかを確認する必要があります。つまり、ショッピングカートに追加されたアイテムが表示されます。
- **テスト手順:** テストでは、ショッピング カートにアイテムを追加し、ショッピング カートにこのアイテムが含まれていることを確認する必要があります。アプリケーション上のユーザー アクションをシミュレートしたテストをレコーディングします。
- **テスト結果の確認とロギング:** アイテムが正しく追加されている場合は、そのアイテムがショッピングカート内にある必要があります。これをチェックするには、ショッピングカートのデータの期待値を比較します。この目的のために、特別な比較コマンドをテストに追加します。このコマンドは、比較結果をテストログに出力するので、検証が失敗したかどうかを見れます。

TestComplete を使用したテストの計画についての詳細は、TestComplete ヘルプの「Planning Tests」を参照してください。

3. テストのレコーディングについて

TestComplete では、2 通りの方法でテストを作成することができます。

- **テストを手動で作成する** - スクリプト オブジェクトまたはキーワード テスト コマンドを使用して、必要なすべてのコマンドとアクションを入力します。このアプローチは、強力で柔軟なテストを作成する必要があるとき、またはテストの作成で豊富な経験がある場合、役に立ちます。しかし、手動でテストを作成することは、多くの時間を必要とし、別の問題が生じる可能性があります。たとえば、使用するアプリケーション オブジェクトのクラスや名前を知っていなければなりません。
- **テストを記録する** - テストを簡単に作成することができます。一度、テストするアプリケーションに対していくつかのアクションを実行すると、TestComplete は自動的にこれらのアクションを認識し、それらをスクリプトまたはキーワード テスト操作に変換します。テストをビジュアルに作成し、実行したアクションをスクリプトまたはキーワード テストに記録できます。これは、非常に役に立つアプローチであり、テストの作成であまり経験を必要としません。

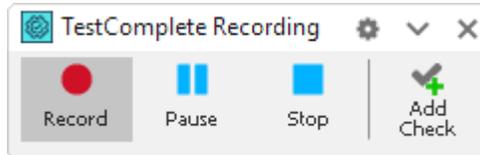
このチュートリアルでは、TestComplete でテストを記録する方法を説明します。

レコーディングは次の 3 つの手順を含みます。

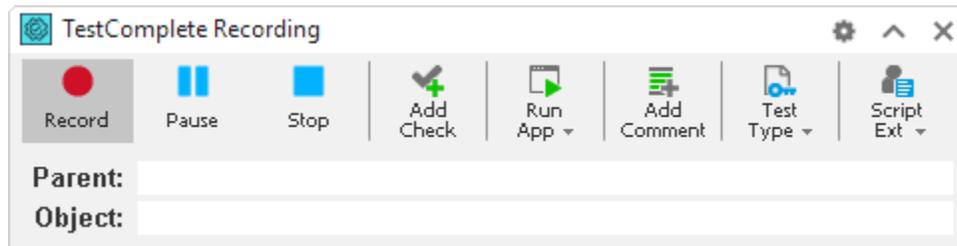
1. レコーディングを開始します。TestComplete のメイン メニューあるいは [Test Engine] ツールバーから **[Test] > [Record] > [Record Keyword Test]** または、**[Test] > [Record] > [Record Script]** を選択してレコーディングを開始します。スタートページの  **[Record Test]** をクリックして、レコーディングを開始することもできます。

さまざまな種類のテスト (キーワード テスト、スクリプト、低レベルのプロシージャ) をレコーディングすることができます。レコーディングの開始に使用するメニュー項目によって、メインのレコーディングするテスト (キーワード テストまたはスクリプト コード) が定義されます。そのほかのテストはそのレコーディングが開始された後でレコーディングされます。メインのレコーディングするテストには、これらのテストを実行する特別なコマンドが含まれます。

TestComplete は、レコーディング モードに切り替わり、画面上に [Recording] ツールバーが表示されます。デフォルトでは、ツールバーは折りたたまれ、レコーディング中に必要となるよく利用されるコマンドだけが表示されます。



[Recording] ツールバーを展開するには、▼ 矢印ボタンをクリックします。すると、すべてのボタンが表示されます。



このツールバーを使用して、レコーディング中に追加の動作を実行したり、レコーディングの一時停止や停止、レコーディングするテスト タイプ (キーワード テスト、スクリプト コード、低レベルのプロシージャ) を変更することができます。

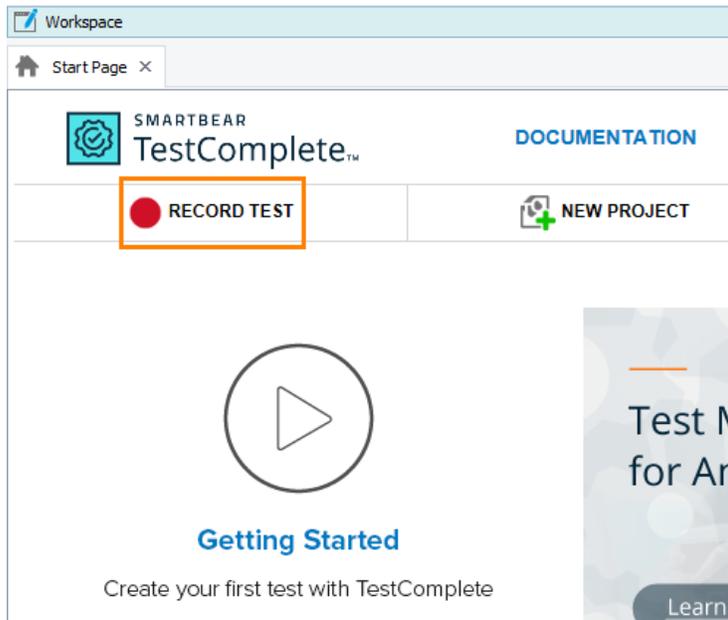
- レコーディング開始後、必要なテスト動作を実行します。テストするアプリケーションを起動し (必要な場合)、コマンド ボタンをクリックしたり、メニュー項目を選択したり、テキストを入力するなどして使用します。
- すべてのテスト動作が終了した後、[Recording] ツールバーから ■ [Stop] を選択し、レコーディングを停止します。

テストのレコーディングに関する詳細は、TestComplete ヘルプの「*Recording in TestComplete*」を参照してください。

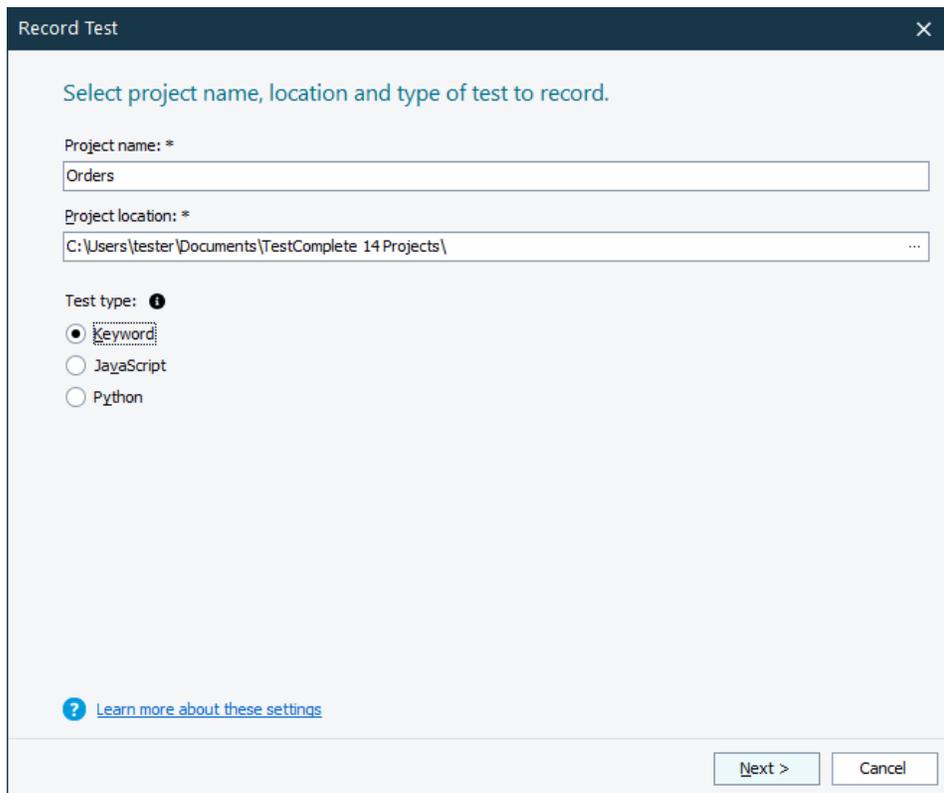
4. テストのレコーディング開始

- TestComplete でプロジェクトあるいはプロジェクト スイートを開いている場合は、閉じてください。これをするには、[File] > [Close] を TestComplete のメインメニューから選択します。

2. TestComplete **スタート** ページに切り替えます。このページが隠れている場合、TestComplete の **[Help]** メニューから **[Start Page]** を選択します。



3. スタートページで、**[Record Test]** をクリックします。TestComplete は、**[Record Test]** ウィザードを表示します。



4. ウィザードの最初のページで、プロジェクトの名前、場所、テストタイプを指定します。

[Project name] テキスト ボックスに **Orders** と入力します。

[Location] はデフォルトの値をそのままにします。

テストタイプを選択します。キーワードテストまたは JavaScript、Python スクリプト テストのいずれかを作成することができます。

- キーワードテストは、マウスのクリック、テキストの入力など、ユーザーの操作を定義する一連のキーワードです。視覚的にキーワード テストを作成します。スクリプトの知識は必要ではありません。
- JavaScript と Python テストは、ユーザー アクションをシミュレートする命令を含むスクリプト関数です。

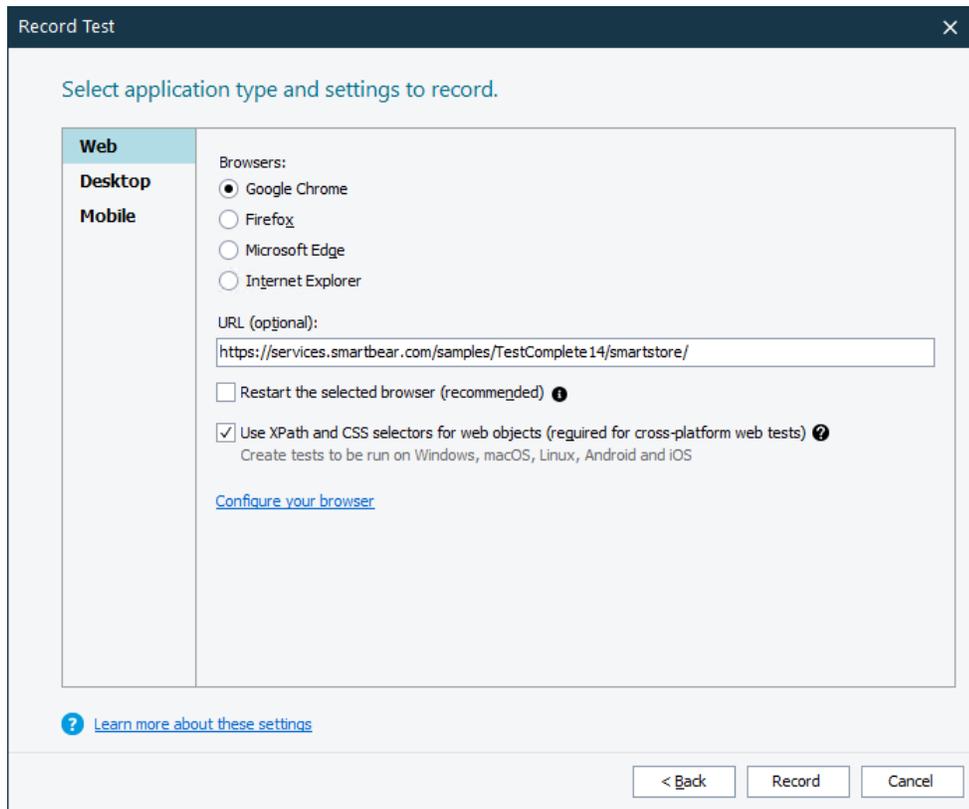
このチュートリアルでは、キーワード テストの作成方法を示します。**[Keyword]** をクリックします。

5. **[Next]** をクリックします。

ウィザードを使って、テストするアプリケーションをプロジェクトに追加するため、ウィザードで作業を続行します。

5. テストする Web ページの定義

1. ウィザードでは、テストするアプリケーションを選択できる 2 番目のページが表示されます。



2. Web ページにある SmartStore アプリケーションをテストするので、**[Web]** をクリックします。

3. サポートしている Web ブラウザーの 1 つを選択します。

- Internet Explorer
- Firefox
- Google Chrome
- Microsoft Edge

注意: TestComplete は Edge Chromium のみサポートします。古いバージョンの Edge ではレコーディングを利用できません。

このチュートリアルでは **Google Chrome** を使用します。サポートされているその他の Web ブラウザーを使用できます。

4. **[URL]** テキスト ボックスで、次の URL を入力します。

<https://services.smartbear.com/samples/TestComplete14/smartstore/>

5. テストを正常に実行するには、テスト実行の開始時にテストするアプリケーションの状態をテストのレコーディング時の開始時と同じにする必要があります。テストのレコーディングを開始するときにテストするアプリケーションが実行されている場合、その状態は後続のテスト実行中の状態と異なることがあります (たとえば、データが読み込まれているか、ダイアログが開いているなど)。そのため、テストのレコーディングを開始する前に、Web ブラウザーの実行中のすべてのインスタンスを閉じることをお勧めします。

サポートされている Web ブラウザーのいずれかが実行されている場合、ウィザードは、**[Restart the selected browser]** チェックボックスを表示します。このチェックボックスをオンにすると、選択した Web ブラウザーを閉じて、レコーディングが開始されると自動的に再起動するようにウィザードに指示します。

(システム内で他の Web ブラウザーが実行されている場合、Web ブラウザーは引き続き実行されます。)

6. **[Use XPath and CSS selectors for web objects]** チェックボックスをオンにすると、CSS セレクターの XPath 式を使用して Web オブジェクトを識別できます。

これは**クロスプラットフォーム Web テスト**では必須です。これらのテストは、Safari と MacOS など、リモート環境のさまざまな Web ブラウザーとプラットフォームで実行できます。そのようなテストの作成は、このチュートリアルではカバーしませんが、

この**チェックボックス**はオンにしたままにすることを推奨します。

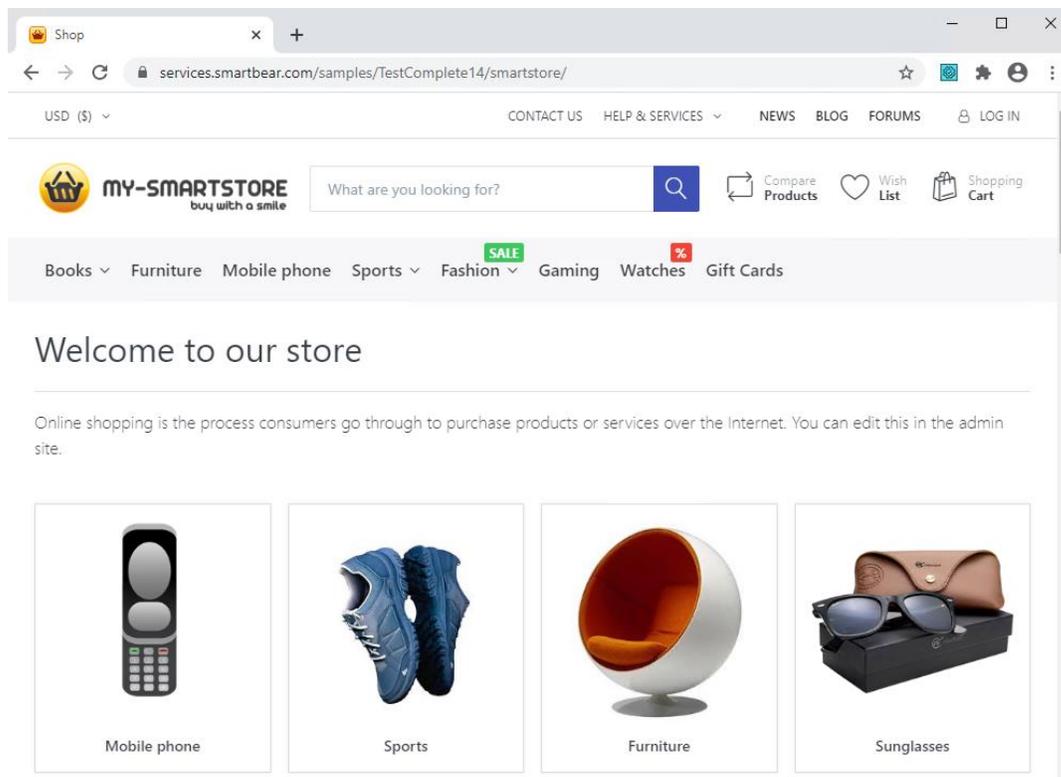
7. **[Record]** をクリックし、プロジェクトの作成を完了し、レコーディングを開始します。

6. テストのレコーディング

注意: レコーディング中に TestComplete ヘルプに切り替えしないでください。レコーディング エンジン はすべてのユーザー アクションをトレースし、レコーディングします。つまり、レコーディングしたテストに、この "切り替え" をシミュレートするコマンドが含まれてしまいます。

操作手順は、レコーディングを開始する前に印刷しておくことができます。あるいは、2 つのモニターがある場合、TestComplete ヘルプをもう一方のモニターへ移動しておくことができます。

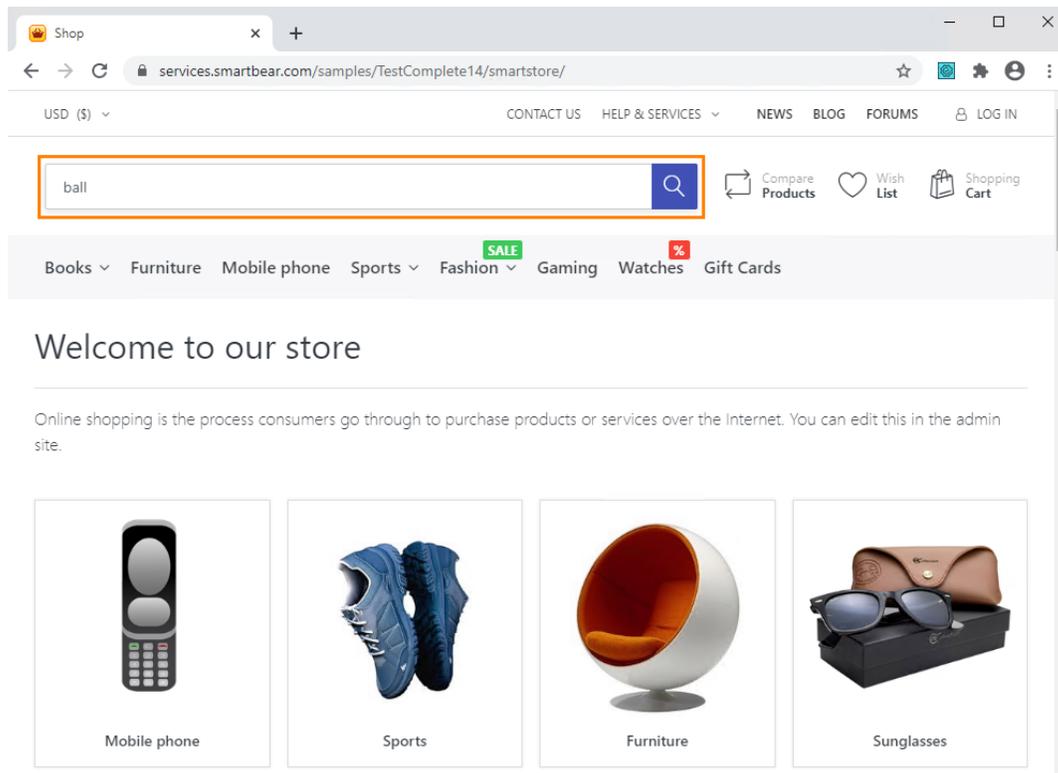
1. 選択した Web ブラウザーが起動し、SmartStore アプリケーションのメインページに移動するまで待ちます。



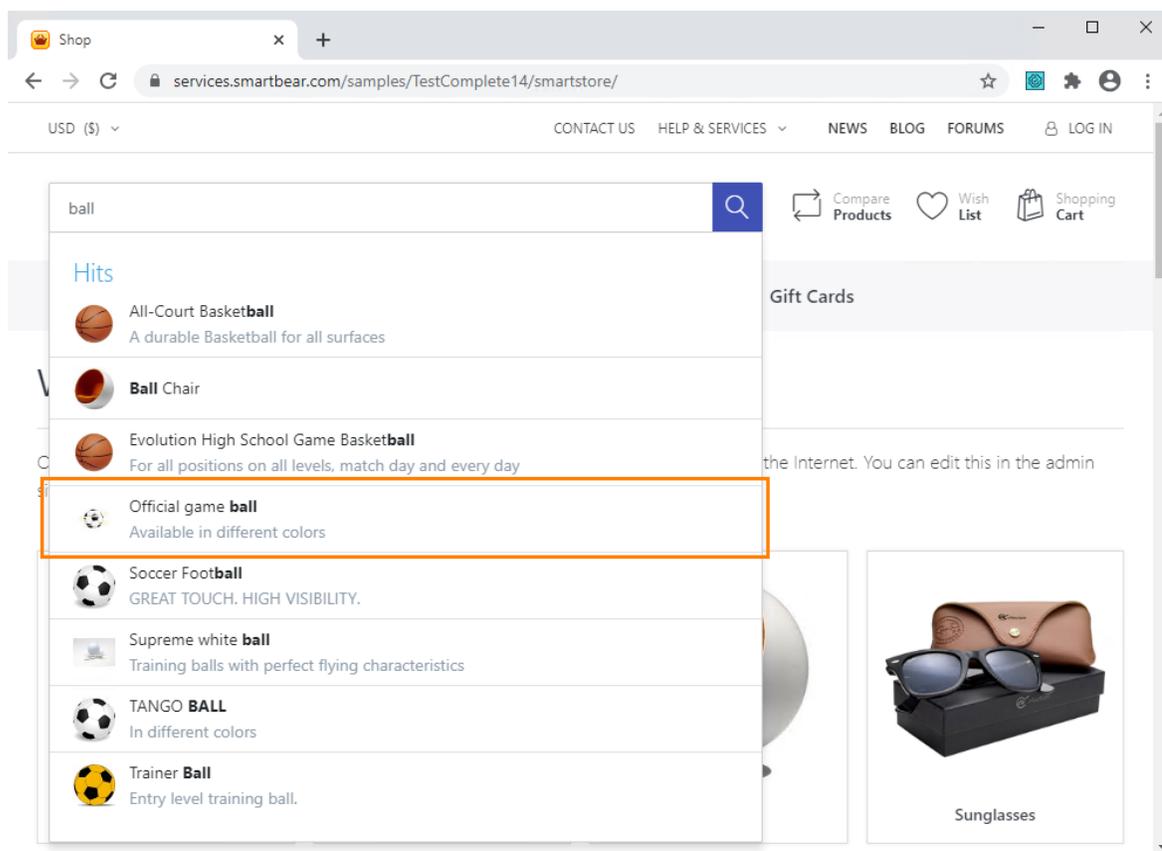
ブラウザーが起動しない場合は、起動して必要な Web ページを手動で開くことができます。

TestComplete は、特別なテスト コマンドを使用して、Web ブラウザーの開始をレコーディングします。後でレコーディングしたテストを分析するときに、このコマンドを参照します。

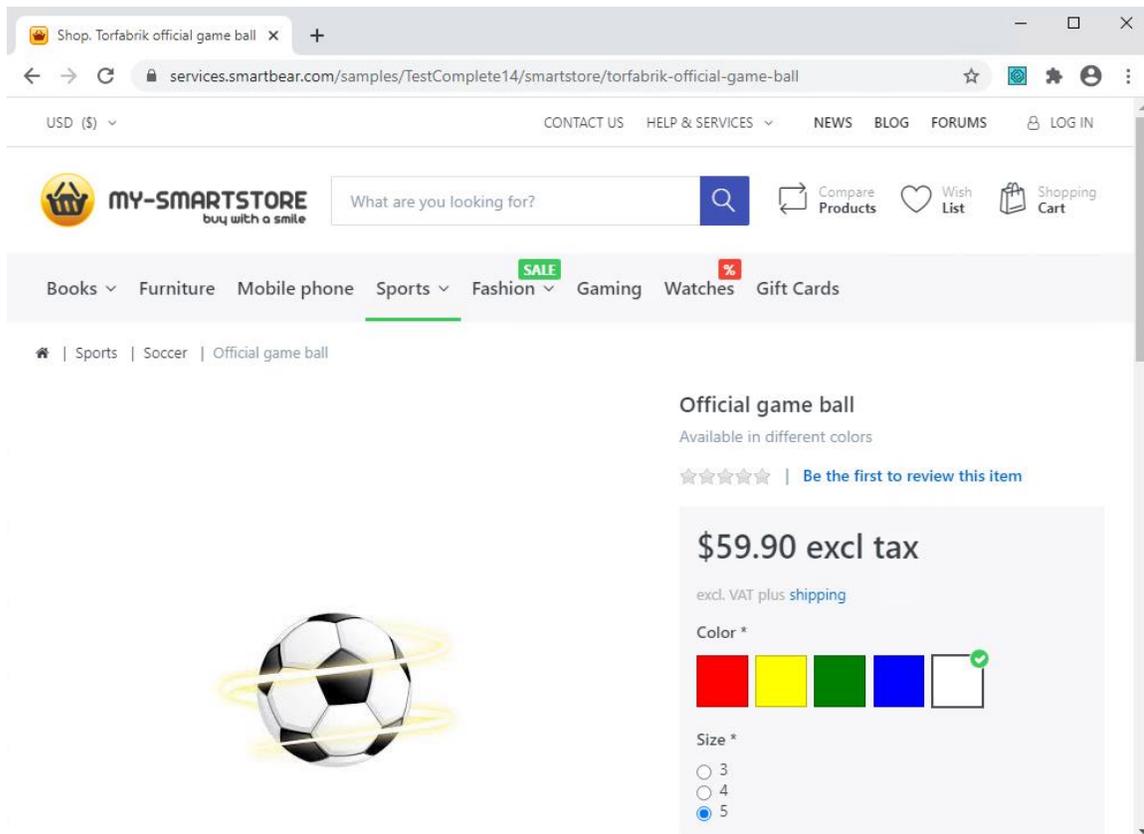
2. **[検索]** テキスト ボックスに、*ball* を入力します。



3. 検索結果で、**Official game ball** アイテムを見つけて、クリックします。

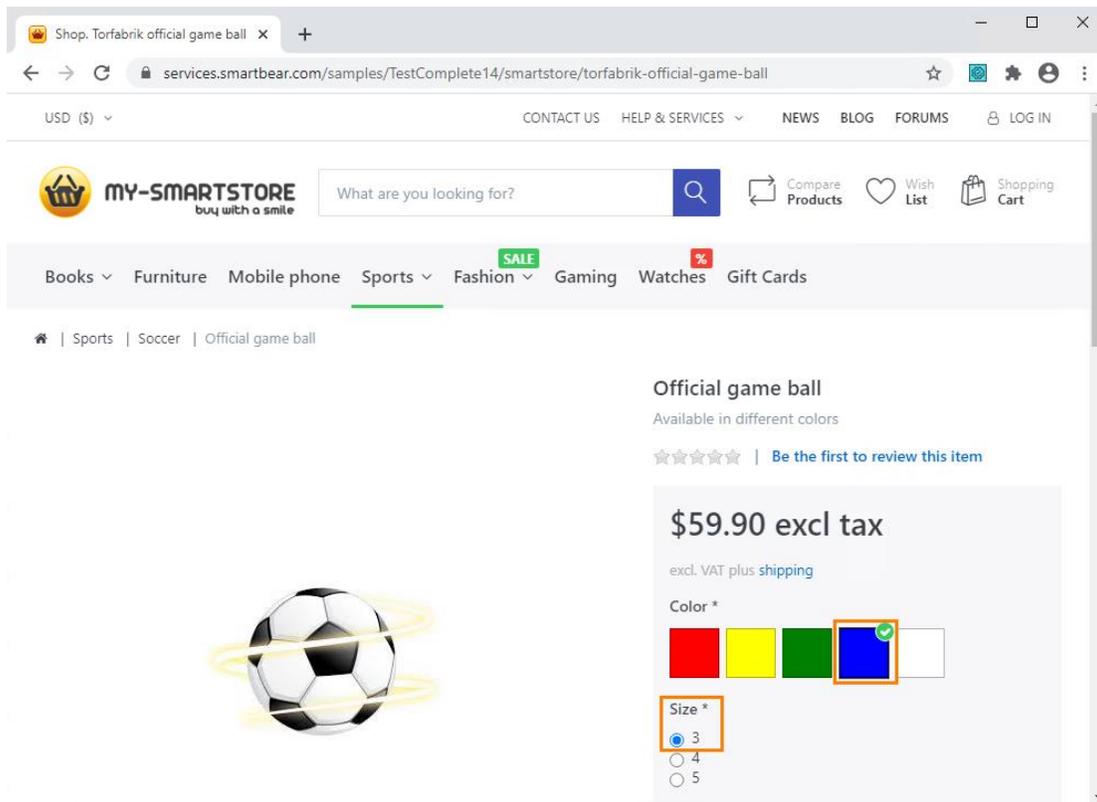


4. アプリケーションが、選択したアイテムページを開きます。

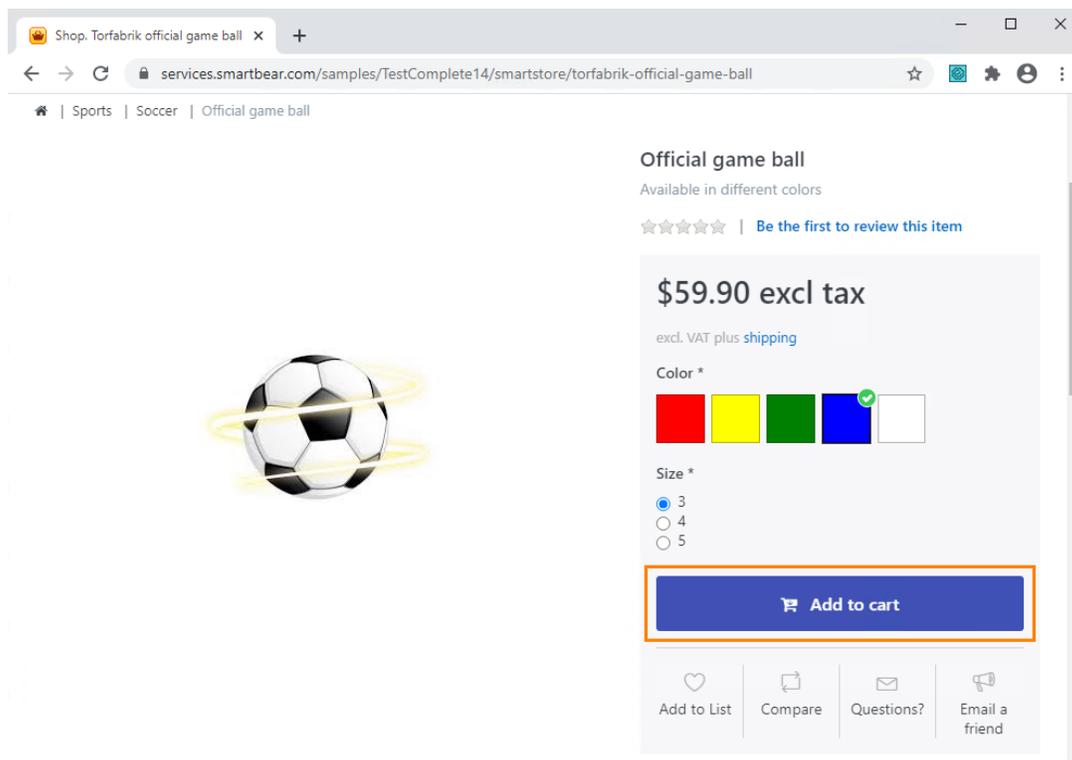


The screenshot displays a web browser window with the URL `services.smartbear.com/samples/TestComplete14/smartstore/torfabrik-official-game-ball`. The page header includes the My-SmartStore logo, a search bar, and navigation links for 'CONTACT US', 'HELP & SERVICES', 'NEWS', 'BLOG', 'FORUMS', and 'LOG IN'. The main navigation menu lists categories: Books, Furniture, Mobile phone, Sports (highlighted), Fashion (with a 'SALE' tag), Gaming, Watches (with a '%' tag), and Gift Cards. The breadcrumb trail shows 'Sports | Soccer | Official game ball'. The product title is 'Official game ball', with a note 'Available in different colors' and a prompt 'Be the first to review this item'. The price is '\$59.90 excl tax', with a sub-note 'excl. VAT plus shipping'. The 'Color *' selection area shows five color swatches: red, yellow, green, blue, and white (selected with a green checkmark). The 'Size *' selection area shows three radio buttons for sizes 3, 4, and 5, with size 5 selected.

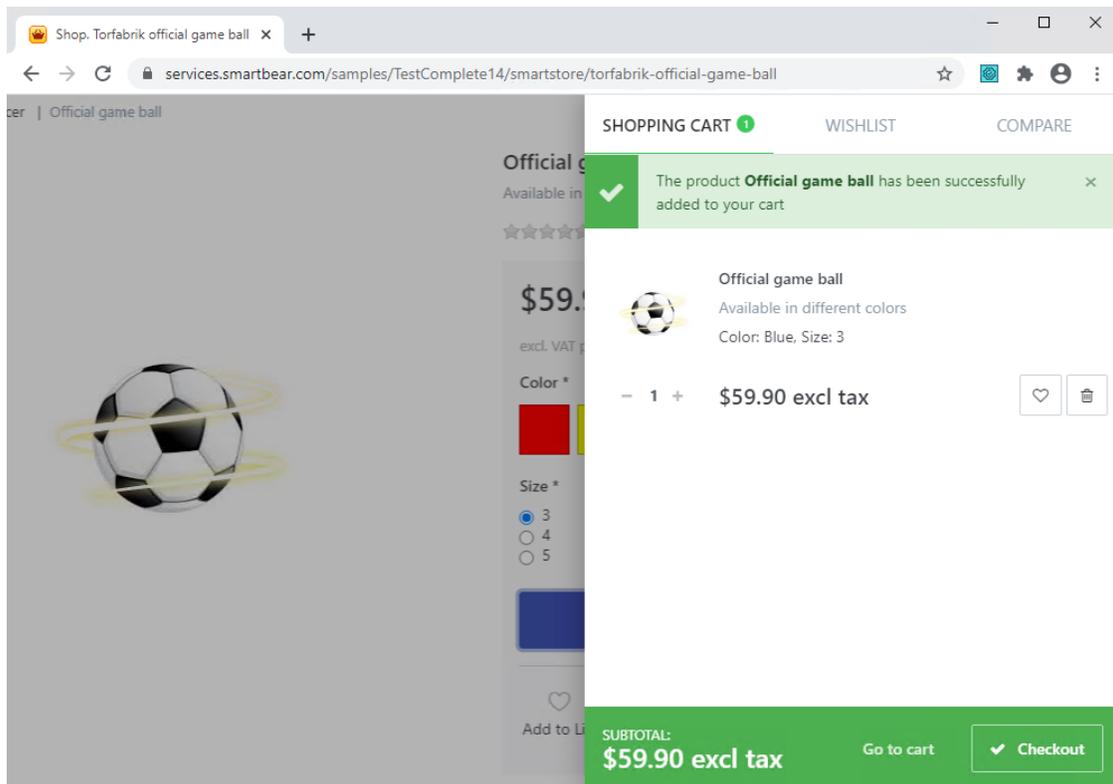
5. そのページで、アイテムのカラー: **青色**と、サイズ: **3** を選択します。



6. **[Add to cart]** をクリックします。



7. アプリケーションの **Shopping Cart** にアイテムが追加されます。

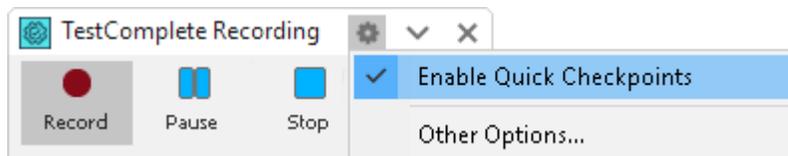


8. テストに比較コマンドを挿入します。ショッピングカートに選択した機能 (カラー: 青色、サイズ: 3) のアイテムが含まれていることが確認されます。

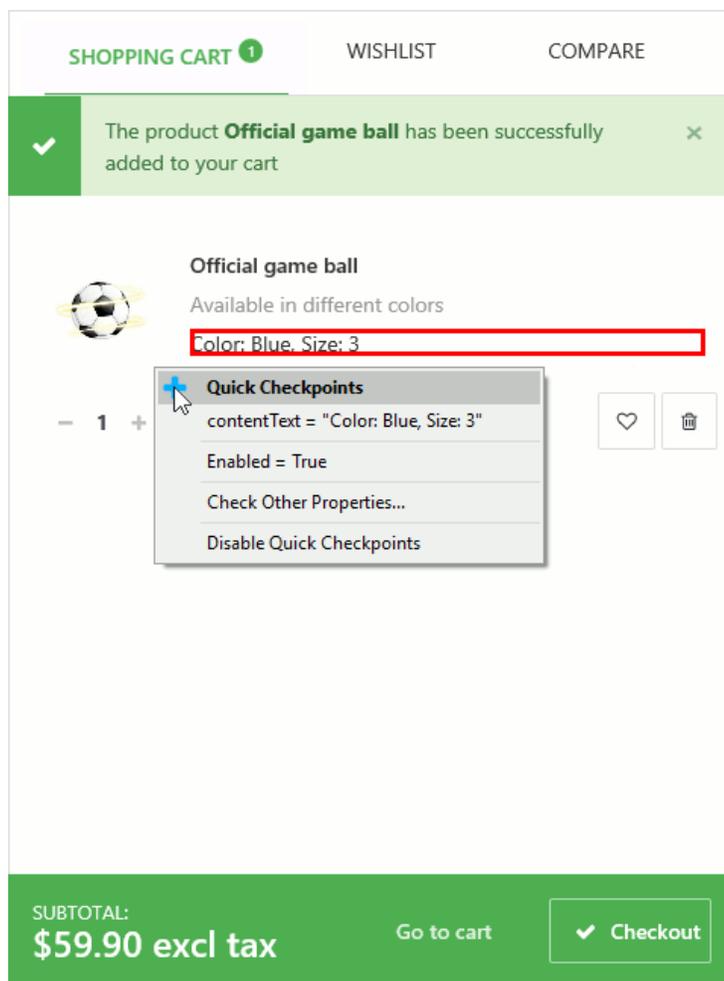
比較コマンドを**チェックポイント**と呼びます。TestComplete には、異なる種類のデータの検証に適したさまざまな種類のチェックポイントが用意されています (TestComplete ヘルプの「Checkpoints」を参照)。最もよく使用されるチェックポイントの 1 つは、**プロパティ チェックポイント**です。アプリケーションのコントロールのデータのチェックに使用されます。このチュートリアルでは、このプロパティチェックポイントを使用します。

プロパティチェックポイントを作成するには、**Create Checkpoint** ウィザードを使用するか、**Quick Checkpoint** を作成します。このチュートリアルでは、Quick Checkpoints の作成方法を示します。

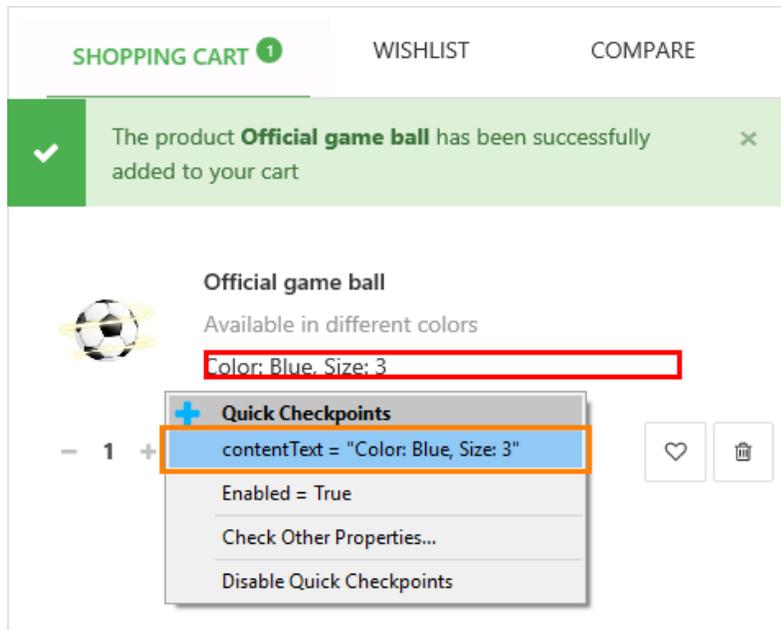
- **[Recording]** ツールバーの  をクリックし、Quick Checkpoints が有効になっていることを確認します。



- マウスポインターを **Color: Blue, Size:3** の行に移動します。TestComplete はそれを赤いフレームでハイライトします。+ アイコンが表示されるまで待ち、マウスポインターをそのアイコンに移動します。TestComplete は、Quick Checkpoints を作成するために最も一般的に使用されるプロパティを表示します。



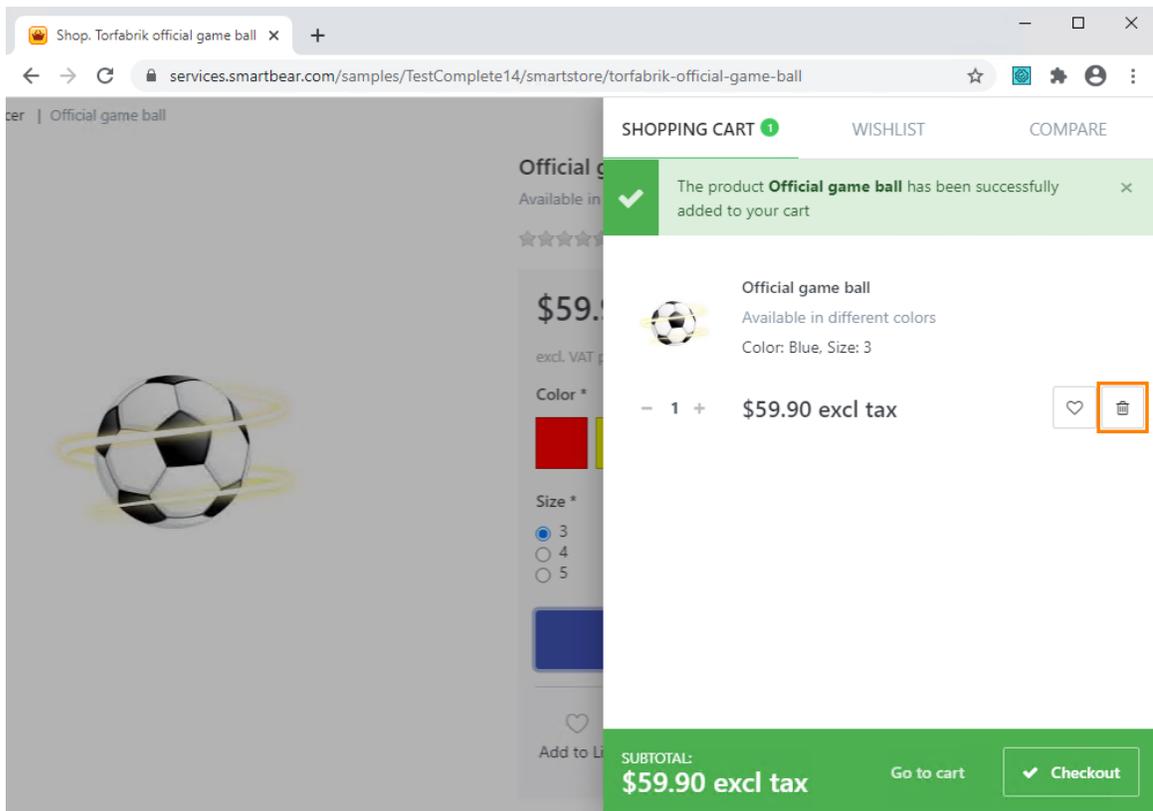
- リストで、**[contentText]** プロパティを選択します。



このプロパティは、選択した Web 要素が表示するテキストへのアクセスを提供します。このプロパティは、ブラウザーに依存しません。つまり、サポートされているすべての Web ブラウザーで同じように動作し、さまざまなブラウザーでテストを実行再生する際の問題を回避できます。

- **TestComplete** は、レコーディングしたテストにチェックポイント コマンドを追加し、それに関する通知を表示します。
- ユーザーアクションのレコーディングを継続できます。

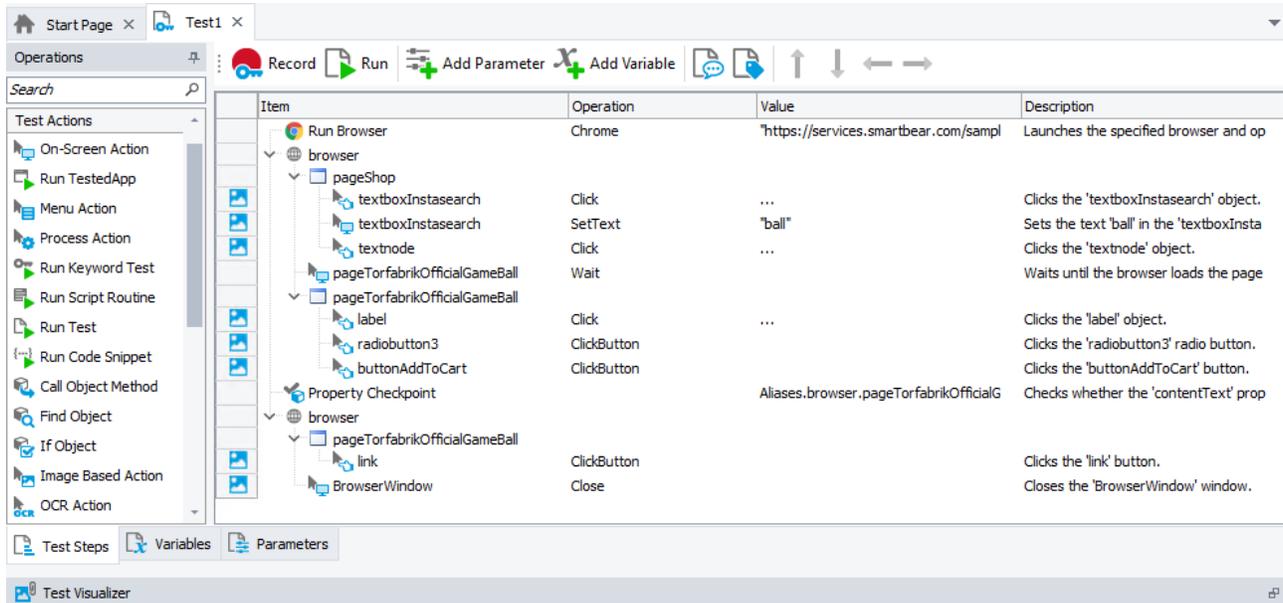
9. アイテムの横にある **[削除]** ボタンをクリックし、ショッピングカートをクリアします。



10. ブラウザーのキャプション バーにある **[X]** ボタンをクリックしてブラウザーを閉じます。
11. [Recording] ツールバーの **[Stop]** を押して、レコーディングを停止します。レコーディングしたテストコマンドが処理され、テストに保存されます。

7. レコーディングしたテストの分析

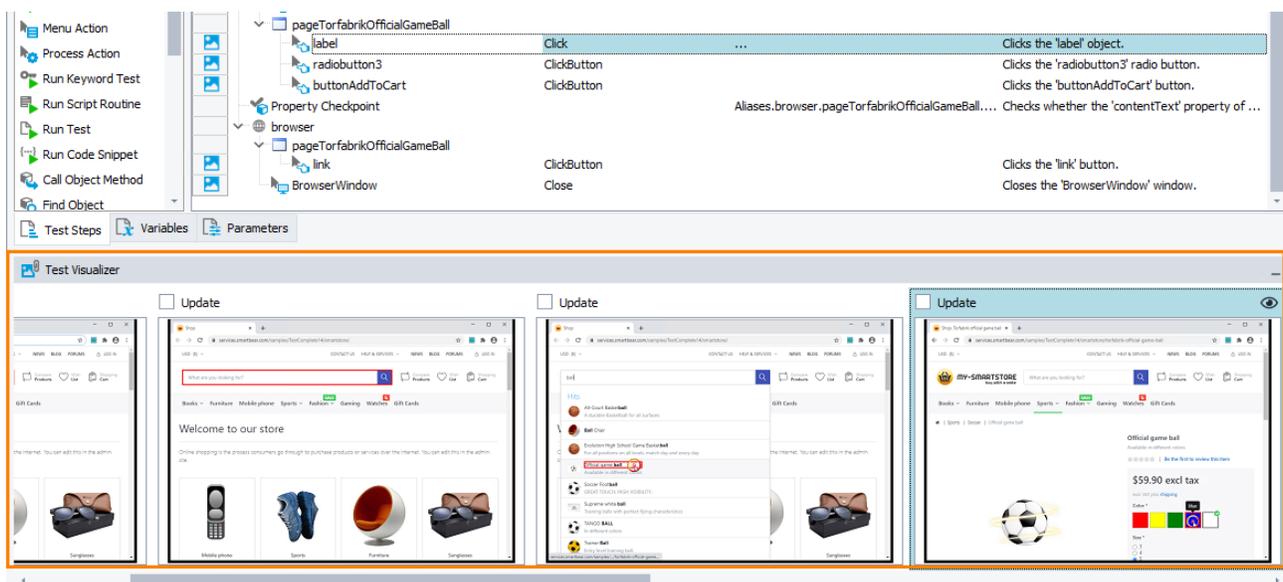
レコーディング終了後、TestComplete はレコーディングしたキーワードを Keyword Test エディターで開きます。



レコーディングしたテストは上記の画像に示すテストのようになります。実行環境によってはこれとは異なることがあります。たとえば、不要なクリック操作が含まれていることがあります。

このテストには、レコーディング中に SmartStore アプリケーション上で実行した動作に対応するコマンドが含まれています。このテストコマンドを **operation** と呼びます。

コマンドの下に **[Test Visualizer]** パネルがあり、TestComplete によってテストのレコーディング中に operation でキャプチャーされた画像が表示されます。



これらの画像は、レコーディングした **operation** を表し、**operation** が実行する動作を理解するのに役立ちます。TestComplete は、ユーザーアクション (マウスクリック、テキスト入力など) に対応した **operation** の画像のみをキャプチャーします。エディター内でこの操作を選ぶと、Test Visualizer は自動的に適切な画像を選択するため、**operation** が実行される前にアプリケーションの状態を簡単に見ることができます。画像の作業についての詳細は、「Test Visualizer」を参照してください。

このテストの最初の **operation** は、**Run Browser** です。これは、ブラウザを起動し、キーワードリストから指定されたページに移動します。TestComplete は、ブラウザが起動されたことを検出すると、自動的にこの操作をレコーディングします。

Item	Operation	Value	Description
Run Browser	Chrome	"https://services.smartbear.com/sampl	Launches the specified browser and op
browser			
pageShop			
textboxInstasearch	Click	...	Clicks the 'textboxInstasearch' object.
textboxInstasearch	SetText	"ball"	Sets the text 'ball' in the 'textboxInsta

次の動作は、検索ボックスと検索結果に連動します。

Item	Operation	Value	Description
Run Browser	Chrome	"https://services.smartbear.com/sampl	Launches the specified browser and op
browser			
pageShop			
textboxInstasearch	Click	...	Clicks the 'textboxInstasearch' object.
textboxInstasearch	SetText	"ball"	Sets the text 'ball' in the 'textboxInsta
textnode	Click	...	Clicks the 'textnode' object.
pageTorfabrikOfficialGameBall	Wait		Waits until the browser loads the page

次に、選択されたアイテムの説明を持つページを待つ動作があります。

textboxInstasearch	Click	...	Clicks the 'textboxInstasearch' object.
textboxInstasearch	SetText	"ball"	Sets the text 'ball' in the 'textboxInsta
textnode	Click	...	Clicks the 'textnode' object.
pageTorfabrikOfficialGameBall	Wait		Waits until the browser loads the page

それに続いて、アイテムの色とサイズを選択し、そのアイテムをカートに追加する動作があります。

textnode	Click	...	Clicks the 'text
pageTorfabrikOfficialGameBall	Wait		Waits until the
pageTorfabrikOfficialGameBall			
label	Click	...	Clicks the 'label
radiobutton3	ClickButton		Clicks the 'radi
buttonAddToCart	ClickButton		Clicks the 'but
Property Checkpoint		Aliases.browser.pageTorfabrikOfficialG	Checks wheth

次は、テストのレコーディング中に追加した比較動作です。

radiobutton3	ClickButton		Clicks the 'radiobutton3' radio button.
buttonAddToCart	ClickButton		Clicks the 'buttonAddToCart' button.
Property Checkpoint		Aliases.browser.pageTorfabrikOfficialG	Checks whether the 'contentText' prop
browser			
pageTorfabrikOfficialGameBall			
link	ClickButton		Clicks the 'link' button.
BrowserWindow	Close		Closes the 'BrowserWindow' window.

その後、ショッピングカートをクリアする動作があります。

buttonAddToCart	ClickButton	Clicks the 'buttonAddToCart' button.
Property Checkpoint	Aliases.browser.pageTorfabrikOfficialG	Checks whether the 'contentText' prop
browser		
pageTorfabrikOfficialGameBall		
link	ClickButton	Clicks the 'link' button.
BrowserWindow	Close	Closes the 'BrowserWindow' window.

そして、Web ブラウザーを閉じる動作です。

Property Checkpoint	Aliases.browser.pageTorfabrikOfficialG	Checks whether the 'contentText' prop
browser		
pageTorfabrikOfficialGameBall		
link	ClickButton	Clicks the 'link' button.
BrowserWindow	Close	Closes the 'BrowserWindow' window.

マウス イベント、キーボード入力、スクリプトからのその他の動作についての詳細は、TestComplete ヘルプの「Simulating User Actions」を参照してください。

TestComplete は、自動的に operation を使用したプロセスやウィンドウに対応するグループにまとめます。グループ化することで、テスト構造を簡単に理解できるようになります。さらにアプリケーションのオブジェクト階層の情報を提供しています。

ここでは、1 つのブラウザのユーザー アクションをレコーディングしました。そのため、ブラウザのグループ ノードがあります。これらには、ブラウザ ウィンドウおよびコントロールで実行したすべての動作が含まれています。さまざまな Web ページで実行した動作はいくつかの "page" グループ ノードにまとめられています。

Item	Operation	Value	Description
Run Browser	Chrome	"https://services.smartbear.com/sampl	Launches the :
browser			
pageShop			
textboxInstasearch	Click	...	Clicks the 'text
textboxInstasearch	SetText	"ball"	Sets the text 'l
textnode	Click	...	Clicks the 'text
pageTorfabrikOfficialGameBall	Wait		Waits until the
pageTorfabrikOfficialGameBall			
label	Click	...	Clicks the 'labe
radiobutton3	ClickButton		Clicks the 'radi
buttonAddToCart	ClickButton		Clicks the 'butt
Property Checkpoint	Aliases.browser.pageTorfabrikOfficialG		Checks whethr
browser			
pageTorfabrikOfficialGameBall			
link	ClickButton		Clicks the 'link'
BrowserWindow	Close		Closes the 'Brc

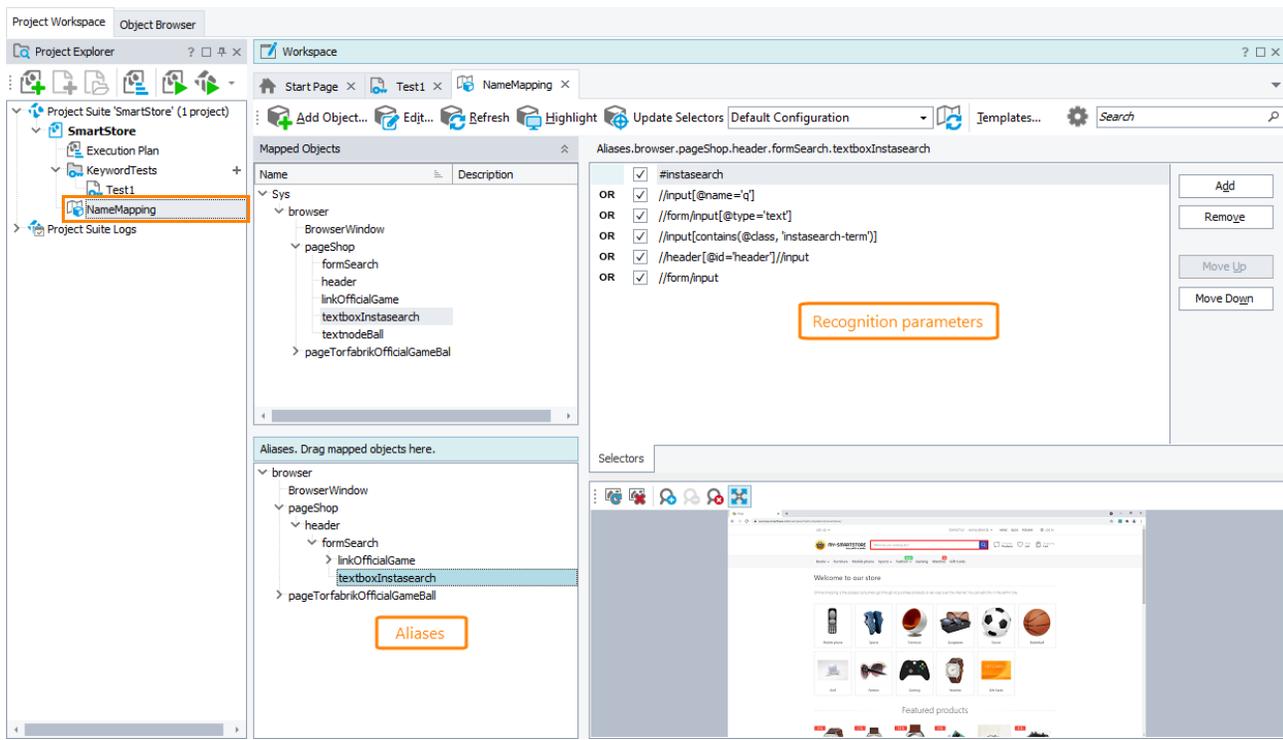
レコーディングしたテストの Web ブラウザー、およびページ要素の名前に注意する必要があります。

	Operation	Value	Description
Browser name	browser	Chrome	Launches the browser
	pageShop		
	textboxInstasearch	Click	Clicks the 'text
	textboxInstasearch	SetText	Sets the text 'ball'
	textnode	Click	Clicks the 'text
	pageTorfabrikOfficialGameBall	wait	Waits until the
	pageTorfabrikOfficialGameBall	Click	Clicks the 'labe
	label	Click	Clicks the 'radi
	radiobutton3	ClickButton	Clicks the 'butt
	buttonAddToCart	ClickButton	Clicks the 'butt
	Property Checkpoint	Aliases.browser.pageTorfabrikOfficialG	Checks wheth
	browser		
	pageTorfabrikOfficialGameBall		
	link	ClickButton	Clicks the 'link'
	BrowserWindow	Close	Closes the 'Brc

これらは、**[Object Browser]** パネルに表示される名前と異なります。たとえば、**[Object Browser]** では、Web ブラウザーの名前は（使用するブラウザーに応じて）**Browser("iexplore")**、**Browser("edge")**、**Browser("firefox")**、または **Browser("chrome")** であるのに対し、テストでは単に **browser** と呼ばれます。別の例は、SmartStore アプリケーションのメインページです。**Object Browser** では、**Page("https://services.smartbear.com/samples/TestComplete14/smartstore/")** と呼ばれ、テストでは、その名前ははるかに短い **pageShop** と呼ばれます。

これは、デフォルトで、テストをレコーディングしている間、**TestComplete** は自動的にすべての項目を **Name Mapping** リポジトリに追加しているためです。項目ごとに、**TestComplete** は後でアプリケーション内のオブジェクトを見つけるために使用する認識パラメーターと、**alias** を格納します。**alias** はテスト内のオブジェクトを見つけるために使用する短い名前です。

Project Explorer の **Name Mapping** 項目をダブルクリックして、Name Mapping リポジトリを開き、TestComplete がテストのレコーディング中にオブジェクト用に作成した認識パラメーターと **alias** を表示できます。



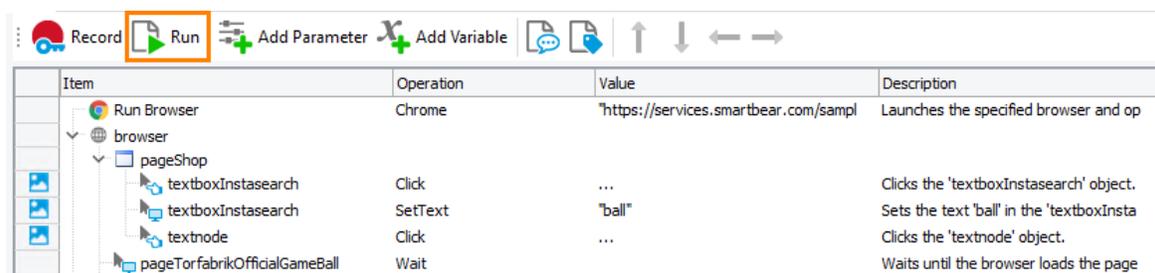
Name Mapping と **alias** の使用は、テストをより分かりやすく、安定したものにします。

8. レコーディングしたテストの実行

TestComplete がユーザー アクションをどのようにシミュレートするか、ここで簡単なテストを実行してみましょう。

レコーディングしたテストを実行する前に、初期状態がレコーディングしたときと同じであることを確認してください。たとえば、ほとんどの Web テストでは、Web ブラウザーが起動され、必要なページが表示されている必要があります。今回の例では、ブラウザを起動し、テストする Web ページを開くため、テストの初めに **Run Browser operation** を使用します。

レコーディングしたテストを実行するには、テスト エディターのツールバーにある  **[Run Test]** をクリックします。



TestComplete がテスト コマンドの実行を開始します。このチュートリアルのテストでは、レコーディングした動作が実行されます。

注意: テスト実行中は、マウスを動かしたり、キーを押したりしないでください。TestComplete でシミュレートされる動作を妨げたり、テスト実行が正しく行われない可能性があります。

テスト実行の終了後、TestComplete はテスト結果を表示します。次の手順では、これを分析します。

テスト実行に関するいくつかの注意:

- テストの実行中、TestComplete は、画面の右上にインジケータを表示します。



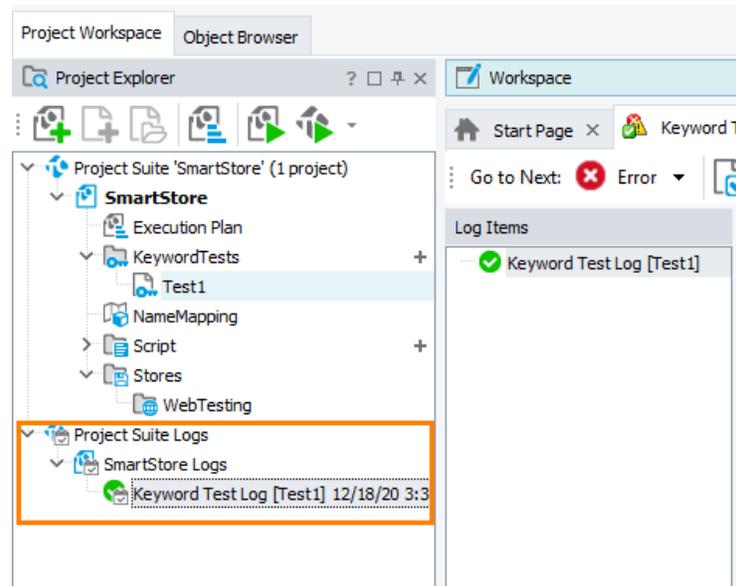
このインジケータは、シミュレートしたテスト動作に関する通知メッセージを表示します。

- TestComplete は、テストが終了するまでこのテスト コマンドを実行します。[Test Engine] ツールバーまたはインジケータの ■ [Stop] をクリックするか、または TestComplete メインメニューから [Test] > [Stop] を選択することで、テストの実行をいつでも停止することができます。
 - [Pause] をクリックすることでテスト実行を一時停止することができます。一時停止の間は、必要な任意の動作を実行することが可能です。たとえば、テスト ログを表示したり、TestComplete の [Watch List] や [Locals] パネルあるいは [Evaluate] ダイアログを使用してテストの変数やオブジェクトを確認することが可能です (TestComplete ヘルプの「Debugging Tests」を参照)。
- TestComplete がインストールされていないマシンでテストを実行するには、TestExecute と呼ばれるリソース フレンドリーなユーティリティを使用することができます。

TestComplete のテスト実行、実行に影響するプロジェクト設定、テスト実施に関する詳細は、TestComplete ヘルプの「Running Tests」を参照してください。

9. テスト結果の分析

TestComplete は、テスト中に実行されたすべての操作の完全なログを保存します。Project Explorer は、テストの結果を **Project Suite Logs > SmartStore Log** ノードの下に維持しています。これは、プロジェクトおよびプロジェクト スイートのテスト履歴を表示するための場所です。各ノードは 1 つのテスト実行に対応します。ノードの左側のアイコンは、対応するテストの実行が成功したかどうかを表します。



テスト実行の**終了後**、TestComplete は自動的に最終結果のノードを追加します。つまり、テストが実行中のときは、結果は表示されません (テストの実行を一時停止した場合は、中間結果を表示することができます)。

ここまでに実行したテストは 1 つのみなので、Project Explorer には 1 つのログ ノードだけが表示されます。デフォルトで、TestComplete は、自動的にそのログを開きます。テスト ログを手動で開くには、[Project Explorer] パネルで表示したログ ノードをダブルクリックします。

注意: デフォルトで、TestComplete はログ ファイルにすべてのテスト結果を格納します。テスト実行ごとにログ ファイル数が増えて、メモリ消費量が増加します。メモリ使用量を減らすために、手動でログ ファイルを削除したり、保存するログ ファイルの数を制限することができます。

この例では、ログは次のようになります。

The screenshot shows the TestComplete interface. At the top, there's a 'Workspace' window with tabs for 'Start Page', 'Test1', and 'Keyword Test Log [Test1] 3/23/20...'. Below that is a toolbar with icons for 'Go to Next', 'Error', 'Create Jira Issue', and 'Select Panel'. The main area is divided into 'Log Items' on the left and 'Test Log' on the right. The 'Log Items' panel shows a tree view with a single node for 'Keyword Test Log [Test1]'. The 'Test Log' panel has a table with columns: Type, Message, Time, Priority, Has Pic..., Link, and Time Diff (sec). The table contains 11 rows of log entries. Below the table, there are 'Picture' and 'Details' tabs. The 'Picture' tab shows two side-by-side images: 'Expected Image' and 'Actual Image', both showing a webpage with a red box highlighting a specific element.

Type	Message	Time	Priority	Has Pic...	Link	Time Diff (sec)
Message	Google Chrome was launched.	17:07:26	Normal			0.00
Message	Navigating to the http://services.smartbear.com/samples/TestComplete14/smartst... page.	17:07:30	Normal			3.48
Message	The window was clicked with the left mouse button.	17:07:30	Normal			0.36
Message	The text 'ball' was entered in the text editor.	17:07:30	Normal			0.26
Message	The window was clicked with the left mouse button.	17:07:31	Normal			1.18
Message	The window was clicked with the left mouse button.	17:07:33	Normal			1.88
Message	The radio button was clicked with the left mouse button.	17:07:34	Normal			0.53
Message	The button was clicked with the left mouse button.	17:07:36	Normal			1.84
Checkpoint	The property checkpoint passed: contentText equals (case-sensitive) "Color: Blue, Size: 3".	17:07:36	Normal			0.79
Message	The button was clicked with the left mouse button.	17:07:37	Normal			0.84
Message	The 'Shop. Torfabrik official game ball - Google Chrome' window was closed.	17:07:38	Normal			0.47

Information

Start Time: 3/23/20 5:07 PM
End Time: 3/23/20 5:07 PM
Run Time: 0:00:14

ログ ウィンドウは、1 度に 1 つのテスト実行の結果を表示します。ウィンドウの左側は、実行されたテストのツリー構造で、それぞれのテストのノードを選択してその結果を表示することができます。1 つのテストを実行した場合、このツリーには 1 つのノードのみ含まれます。このノードのアイコンは、テストが成功したかどうかを示します。

テスト ログには、エラー、警告、情報および他の種類のメッセージが含まれています。左のアイコンは、メッセージの種類を示します。メッセージ リストの上のチェック ボックスを使用することで、種類によってメッセージを表示/非表示にすることができます。

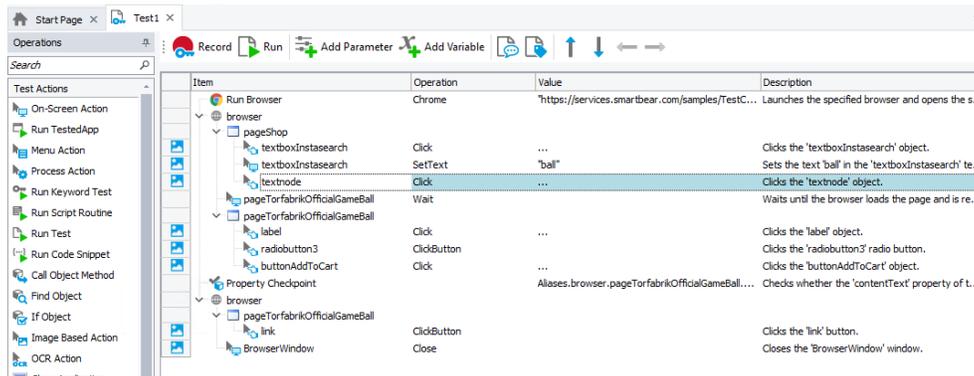
それぞれのメッセージでは、アクションが行われた時間が **Time** 列に表示されます。

TestComplete は、メッセージとともに追加テキストや画像を出力することがあります。これらを確認するには、ログ内でメッセージを選択し、メッセージリストの下にある **[Details]** や **[Picture]** ペインを参照します。

[Picture] パネルでは、選択したテスト コマンドを実行する前に、期待される状態と実際のアプリケーションの状態を示す画像が表示されます("Expected" (期待される) とはレコーディング中にそのテスト コマンドでキャプチャーされた画像を指し、"Actual" (実際) とはテストの実行中にキャプチャーされた画像を指します)。この画像を比較するには、**[View Comparison Results]** をクリックして、違いを簡単に見ることができます。そのため、テストでエラーが発生した場合、簡単に見つけることができます。詳細については、TestComplete ヘルプの「Test Visualizer」を参照してください。

ログにメッセージを出力したテスト操作を表示するには、ログ内でメッセージをダブルクリックします。すると、エディターでキーワード テストが開き、該当する操作がハイライトされます。たとえば、ログの "The button

was clicked with the left mouse button" メッセージをダブルクリックした場合、この動作が行われたキーワードテスト操作がハイライトされます。



テスト ログ パネル、ログヘッメッセージを出力、その結果の使用に関する詳細情報は、TestComplete ヘルプの「*About Test Log*」を参照してください。

注意: ここで説明したログは、TestComplete キーワード テストおよびスクリプト特有のログです。ほかの種類テストは異なった構造のログになります。ログに関する詳細情報は、適当なプロジェクト項目の解説を参照するか、そのログ ページ内をクリックして、**F1** を押してください。

エラーの解決

テストが失敗することがあります。これにはいくつかの可能性がありま。たとえば、開発者がアプリケーションの振る舞いを変更したり、ページと Web 要素の認識属性の変更によりテスト エンジンが必要なオブジェクトの検出に失敗したり、サードパーティのアプリケーションがテストするアプリケーションの Web ブラウザーに重なっているためにテスト エンジンがアクションのシミュレートに失敗するなどの可能性があります。

初めてのユーザーが直面する最も典型的なものは、テスト作成中と再生実行中でアプリケーションの状態が異なることです。この問題を避けるには、テストを作成したときと同じ初期状態でテスト実行しているか確認してください。

例:

- Web ブラウザーがテストを記録する前に実行されていた場合は、テストを実行する前に実行している必要があります。
- テストする Web ブラウザーがテストを記録する前に実行されていた場合は、テストを実行する前に実行している必要があります。
- アプリケーション内のデータを編集して保存した場合は、変更を元に戻す必要があります。

テストするアプリケーションが変更され、テストするオブジェクトの検出に使用される識別プロパティが無効になると、TestComplete は不足オブジェクトを見つけることができず、テスト ログに "The object does not exist" エ

ラーを記録します。画面上の実際のオブジェクトをポイントすることで、不足オブジェクトの識別プロパティをログから直接更新できます。

The screenshot shows the TestComplete interface. On the left, there is a 'Log Items' panel with a red 'x' icon and the text 'Keyword Test Log [Test1]'. The main area is the 'Test Log' window, which has a search bar and a table of log entries. The table has columns for Type, Message, Time, Priority, Has..., Link, and Time D... The entries are:

Type	Message	Time	Priority	Has ...	Link	Time D...
Message	Google Chrome was launched.	10:41:37	Normal			0.00
Message	Navigating to the https://services.smartbear.com/samples/TestComplete14/smarts... page.	10:41:41	Normal			3.45
Message	The 'Shop - Google Chrome' window was maximized.	10:41:41	Normal			0.19
Error	The object "textboxInstasearch" does not exist.	10:41:51	Higher			10.52
Error	The test run has stopped because your test project is configured to stop on errors.	10:41:51	Normal			0.06

Below the table, there are tabs for 'Picture', 'Details', and 'Call Stack'. The 'Details' tab is selected, showing a message: 'Unable to find the following object when executing the Click command: Aliases.browser.pageShop.header.formSearch.textboxInstasearch'. There are also 'Tips' listed, including 'Update the search criteria' and 'Learn more about possible causes of the error.' To the right of the text is a screenshot of a web browser showing a 'Welcome to my store' page with various product images.

At the bottom left, there is an 'Information' panel with the following details:

- Start Time: 8/30/2021 10:41 AM
- End Time: 8/30/2021 10:41 AM
- Run Time: 0:00:18

エラーの原因を探し、典型的な問題を解決するための詳細情報は、TestComplete ヘルプの「Handling Playback Errors」を参照してください。

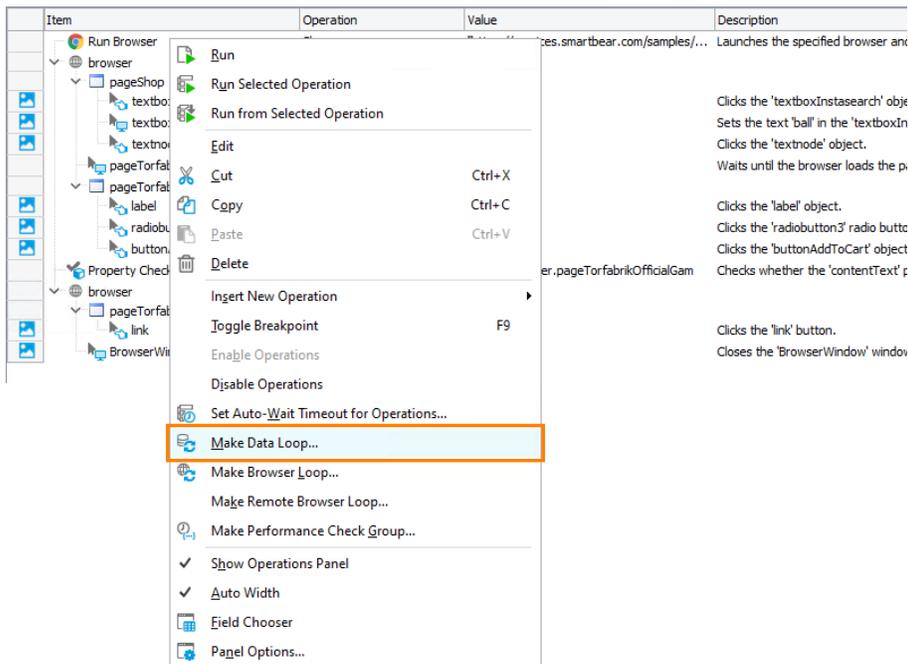
10. 複数のブラウザでのテストの実行

レコーディングしたブラウザでテストが正常に実行されることを確認したら、テストを変更して、別のブラウザで実行することができます (これは、**クロスブラウザ テスト**と呼ばれます)。これにより、Web アプリケーションがさまざまなブラウザで正しく動作することを確認できます。

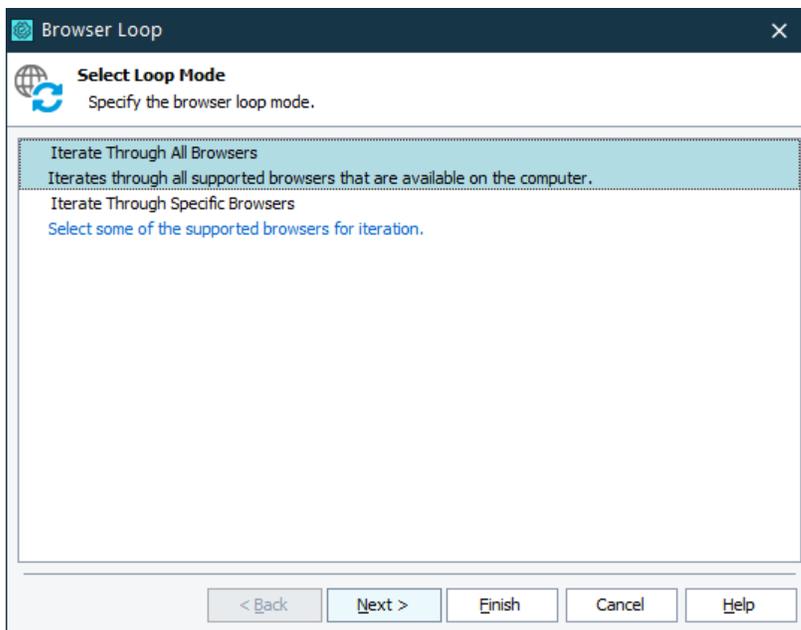
ここで、システムにインストールされているサポートされているすべてのブラウザでテストが実行されるようにテストを修正します。

1. Keyword Test エディターでテストを開きます。

2. Run Browser operation を右クリックして、コンテキストメニューから **[Make Browser Loop]** を選択します。



3. 操作パラメーターのダイアログで、**[Iterate Through All Browsers]** を選択し、**[Finish]** をクリックします。



Run Browser operation が **Browser Loop** operation に変換されます。

4. Browser Loop operation の後のすべてのテスト操作を選択し、➡️をクリックしてこれらの操作をループの内側に移動します。これらの操作はループの各反復で実行されます。

変更後のテストは次のようになります。

Item	Operation	Value	Description
Browser Loop	All Installed browsers	"https://services.smartbear.com/samples/TestC...	Iterates through the specified browsers.
browser			
pageShop			
textboxInstasearch	Click	...	Clicks the 'textboxInstasearch' object.
textboxInstasearch	SetText	"ball"	Sets the text 'ball' in the 'textboxInstasearch' te...
textnode	Click	...	Clicks the 'textnode' object.
pageTorfabrikOfficialGameBall	Wait		Waits until the browser loads the page and is re...
pageTorfabrikOfficialGameBall			
label	Click	...	Clicks the 'label' object.
radiobutton3	ClickButton		Clicks the 'radiobutton3' radio button.
buttonAddToCart	Click	...	Clicks the 'buttonAddToCart' object.
Property Checkpoint		Aliases.browser.pageTorfabrikOfficialGameBall...	Checks whether the 'contentText' property of t...
browser			
pageTorfabrikOfficialGameBall			
link	ClickButton		Clicks the 'link' button.
BrowserWindow	Close		Closes the 'BrowserWindow' window.

5. TestComplete のメインメニューから **[File] > [Save]** を選択してテストを保存します。

TestComplete ヘルプの「Preparing Web Browsers」の説明に従って、ほかのブラウザの準備と設定を行います。

変更したテストを実行します。

TestComplete は、ブラウザの数だけテスト操作を繰り返します。毎回、異なるブラウザでテストアクションが実行されます。

テスト ログには、使用したブラウザに関する情報と、各ブラウザで実行されたテスト操作の結果が含まれます。

Type	Message	Time	Priority	Has Picture	Link	Time Diff (sec)
Message	Mozilla Firefox was launched.	17:11:41	Normal			0.00
Navigation	Navigating to the https://services.smartbear.com/samples/TestComplete14/smarts... page.	17:11:45	Normal			4.09
Click	The window was clicked with the left mouse button.	17:11:46	Normal			1.09
Click	The button was clicked with the left mouse button.	17:11:54	Normal			0.59
Close	The 'Shop. Torfabrik official game ball - Mozilla Firefox' window was closed.	17:11:55	Normal			0.58
Message	Microsoft Internet Explorer was launched.	17:11:56	Normal			0.86
Navigation	Navigating to the https://services.smartbear.com/samples/TestComplete14/smarts... page.	17:12:06	Normal			10.55
Click	The window was clicked with the left mouse button.	17:12:07	Normal			0.60
Click	The button was clicked with the left mouse button.	17:12:18	Normal			2.13
Close	The 'Shop. Torfabrik official game ball - Internet Explorer' window was closed.	17:12:19	Normal			0.54
Message	Google Chrome was launched.	17:12:20	Normal			0.82
Navigation	Navigating to the https://services.smartbear.com/samples/TestComplete14/smarts... page.	17:12:31	Normal			10.49
Click	The window was clicked with the left mouse button.	17:12:31	Normal			0.49
Click	The button was clicked with the left mouse button.	17:12:42	Normal			1.54
Close	The 'Shop. Torfabrik official game ball - Google Chrome' window was closed.	17:12:43	Normal			0.78
Message	Microsoft Edge was launched.	17:12:44	Normal			0.61
Navigation	Navigating to the https://services.smartbear.com/samples/TestComplete14/smarts... page.	17:12:48	Normal			0.79
Click	The window was clicked with the left mouse button.	17:12:49	Normal			4.58
Text	The text 'ball' was entered in the text editor.	17:12:50	Normal			0.79
Click	The button was clicked with the left mouse button.	17:12:50	Normal			0.50
Checkpoint	The property checkpoint passed: contentText equals (case-sensitive) "Color: Blue, Size: 3".	17:13:15	Normal			1.79
Click	The button was clicked with the left mouse button.	17:13:16	Normal			0.98
Close	The 'Shop. Torfabrik official game ball - Microsoft Edge' window was closed.	17:13:17	Normal			0.65

TestComplete を使用したクロスブラウザ テストについての詳細は、TestComplete ヘルプの「Cross-Browser Testing - Overview」を参照してください。

Android アプリケーションのテスト

このチュートリアルでは、TestComplete で Android アプリケーションをテストする方法を説明します。これは、自動テストの一般原則に精通しており、TestComplete IDE についての最低限の知識を持っていることを前提としています。

! 初心者の方は、上記の自動テストの紹介を読むことをお勧めします。

システム要件

以下の要件を満たす場合、このチュートリアルの手順に従って Android アプリケーションのテストを作成できます。

- TestComplete がインストールされた Windows マシンと TestComplete Mobile モジュールのアクティブなライセンス。
- Appium サーバーが管理する Android デバイスへのアクセス。次のいずれかです。
 - BitBar が提供するモバイル デバイス クラウド。このチュートリアルでは、こちらを使用します。無料の評価用 BitBar アカウントを作成して、モバイル デバイスにアクセスする方法を紹介します。
 - または -
 - ローカル マシンまたはローカル ネットワークのリモート マシンで実行しているプライベート Appium サーバー。プライベート Appium サーバーの使用法については、このチュートリアルではカバーしません。

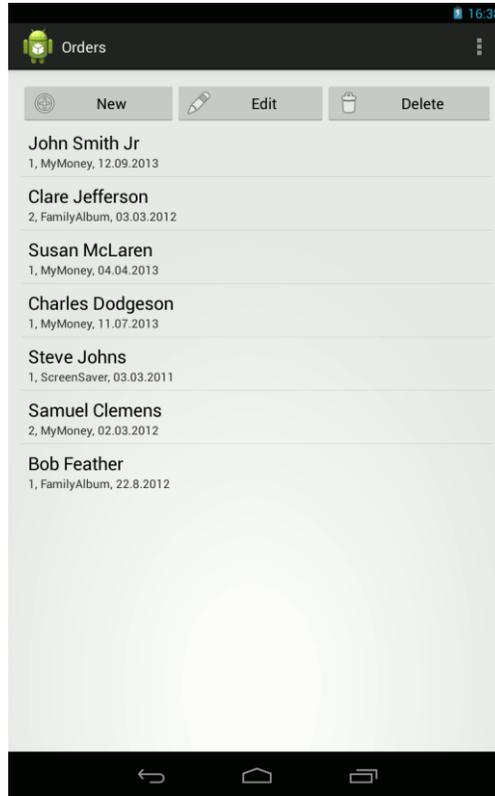
注意: このチュートリアルには、Java や Android SDK などの他のソフトウェアは必要ありません。

チュートリアルについて

評価用 BitBar アカウントの作成、BitBar デバイス クラウド上の Android モバイル デバイスへの接続、テストするアプリケーションの展開、簡単なテストの作成と実行、結果の分析を行います。

テストするアプリケーションについて

この説明では、Orders アプリケーションの Android バージョンを使用します。このアプリケーションは、注文テーブルを管理します。新しい注文の追加、既存の注文の変更または削除、既存の注文リストの確認を行うことができます。



アプリケーションを入手するには

1. 以下の Web サイトから TestComplete Samples インストール パッケージをダウンロードします。
⇒ <https://support.smartbear.com/testcomplete/downloads/samples/>
2. インストールを実行します。
3. サンプルは、<TestComplete Samples>\Mobile\Android\Orders\ フォルダにインストールされます。

1. 無料の BitBar アカウントの申し込み

TestComplete では、Appium サーバーが管理する Android デバイスに接続して、自動テストを作成して実行できます。このチュートリアルでは、SmartBear のクラウド モバイル テスト サービス「BitBar」が提供する、すぐに使えるソリューションを使用します。

無料の評価用 BitBar アカウントを作成して使用方法を紹介します。評価用アカウントの申し込みは数分で完了し、必要なのは有効なメール アドレスのみです。

チュートリアルを開始する前に申し込むことも、スキップして後で実際にテストのレコーディングを開始する際に申し込むこともできます。

1. 評価用 BitBar アカウントの申し込み

1. 以下のリンクをクリックして、BitBar Web サイトで無料の評価用アカウントの申し込み手続きを行ってください。

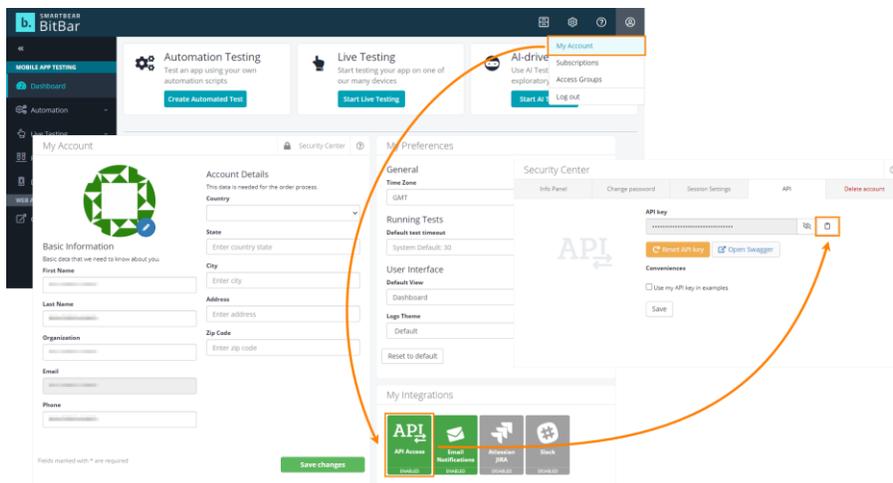
➔ bitbar.com/signup

2. Web フォームの指示に従ってください。

2. BitBar API Key の入手

TestComplete のテストから BitBar リソースにアクセスするには、**BitBar API Key** を使用します。キーを取得するには

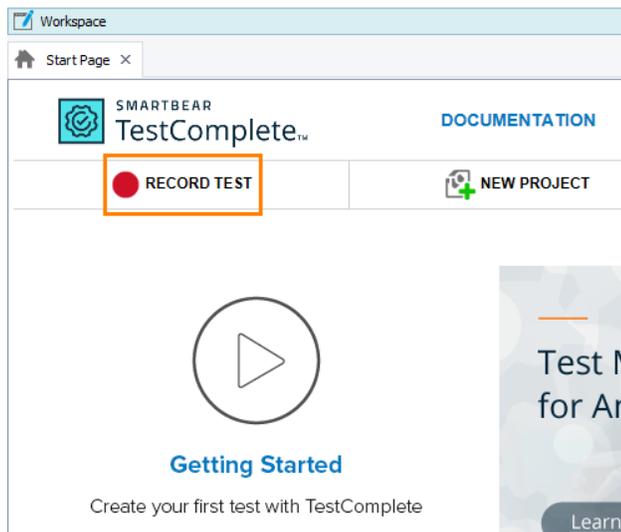
1. 作成した BitBar アカウントで BitBar にログインします。
2. ページの右上にある  > **[My Account]** をクリックします。
3. **[My Integrations]** セクションで **[API]** をクリックします。
4. 結果ページで  をクリックして、API Key をクリップボードにコピーします。



2. テストのレコーディング開始

1. TestComplete でプロジェクトあるいはプロジェクト スイートを開いている場合は、閉じてください。これをするには、**[File] > [Close]** を TestComplete のメインメニューから選択します。

2. TestComplete **スタート** ページに切り替えます。このページが隠れている場合、TestComplete の **[Help]** メニューから **[Start Page]** を選択します。



3. スタートページで、**[Record Test]** をクリックします。TestComplete は、**[Record Test]** ウィザードを表示します。

4. ウィザードの最初のページで、プロジェクトの名前、場所、テスト タイプを指定します。
[Project name] テキスト ボックスに *Orders* と入力します。
[Location] テキスト ボックスはデフォルトの値のままにします。

テストタイプを選択します。キーワードテストまたは JavaScript、Python スクリプトテストのいずれかを作成することができます。

- キーワードテストは、マウスのクリック、テキストの入力など、ユーザーの操作を定義する一連のキーワードです。視覚的にキーワード テストを作成します。スクリプトの知識は必要ではありません。
- JavaScript と Python テストは、ユーザー アクションをシミュレートする命令を含むスクリプト関数です。

このチュートリアルでは、キーワード テストの作成方法を示します。[Keyword] をクリックします。

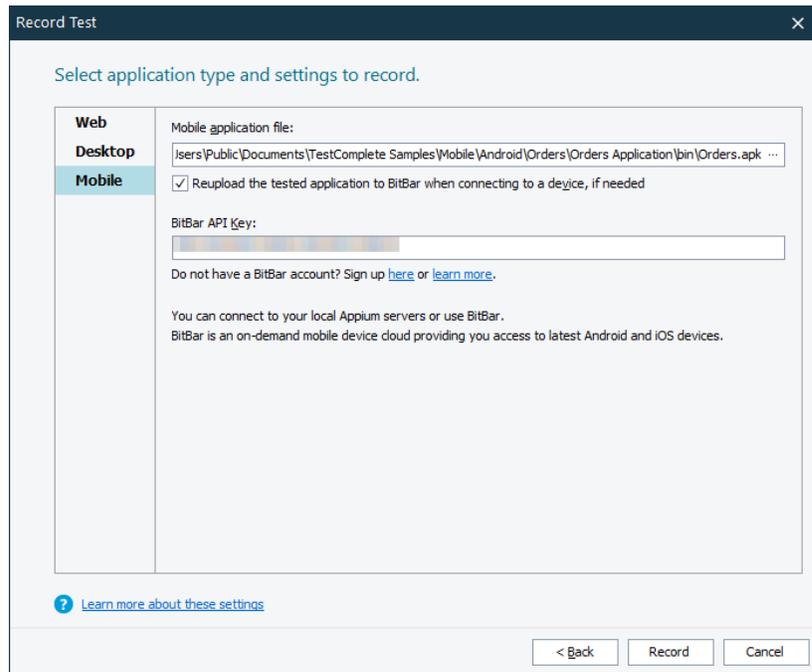
5. [Next] をクリックします。

ウィザードを使って、テストするアプリケーションをプロジェクトに追加するため、ウィザードで作業を続行します。

3. テストするアプリケーションの定義

モバイル デバイスで実行するアプリケーションのテストを作成するには、そのデバイスへアプリケーションを展開する必要があります。BitBar モバイル デバイス クラウド プロバイダーを使用する場合は、アプリケーション ファイル (*Orders.apk*) をアカウントのファイル ライブラリにアップロードします。TestComplete からファイルをアップロードする最も簡単な方法は、テストのレコーディングの開始時などに、TestComplete プロジェクトにファイルを追加することです。

1. ウィザードでは、テストするアプリケーションを選択できる 2 番目のページが表示されます。



2. テストする Orders アプリケーションは、.apk ファイルとして出荷される Android アプリケーションなので、Mobile アプリケーションのカテゴリに該当します。

[Mobile] をクリックします。

3. **[Mobile application file]** ダイアログで、[...] ボタンをクリックします。表示される **[Select Tested Application]** ダイアログで、*Orders.apk* ファイルの場所を見つけます。

[Reupload the tested application to BitBar when connecting to a device, if needed] チェックボックスはデフォルトのままオンにします。

4. すでに BitBar アカウントを持っている場合は、アカウントに割り当てられた API Key を **[BitBar API Key]** テキストボックスに入力します。

API Key の取得方法は、前述の手順を参照してください。

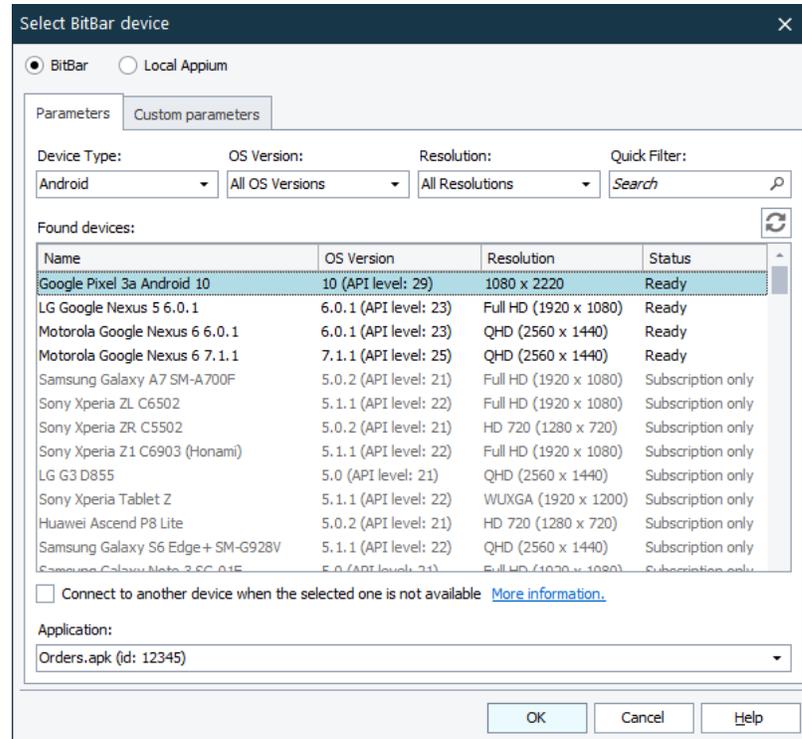
このチュートリアルステップ 1 をスキップした場合は、ウィザードの **[Sign up]** リンクをクリックすると無料の評価用アカウントを申し込むことができます。

5. **[Record]** をクリックし、プロジェクトの作成を完了し、レコーディングを開始します。

4. デバイスへ接続してテスト セッションを開く

レコーディングを開始する前に、テストをレコーディングするデバイスを選択して、Orders アプリケーションのテストセッションを開きます。

1. **[Select BitBar Device]** ダイアログが表示されます。BitBar が提供するすべてのモバイル デバイスがリストされます。



2. 任意の利用可能な Android デバイスを選択します。デバイスが利用可能かどうかは **[Status]** カラムで確認できます。

評価ユーザー向けにいくつかのデバイスが用意されています。このチュートリアルでは、Google Pixel を使用します。

適切なデバイスをすばやく見つけるには、プラットフォーム、バージョン、名前がデバイス リストをフィルタリングできます。

3. **Orders** アプリケーションのテスト セッションを開くには、アプリケーションがターゲット デバイスにアップロードされている必要があります。前のステップで有効な `.apk` ファイルを指定した場合、TestComplete はファイルを自動的にアップロードします。
4. **[OK]** をクリックして続行します。

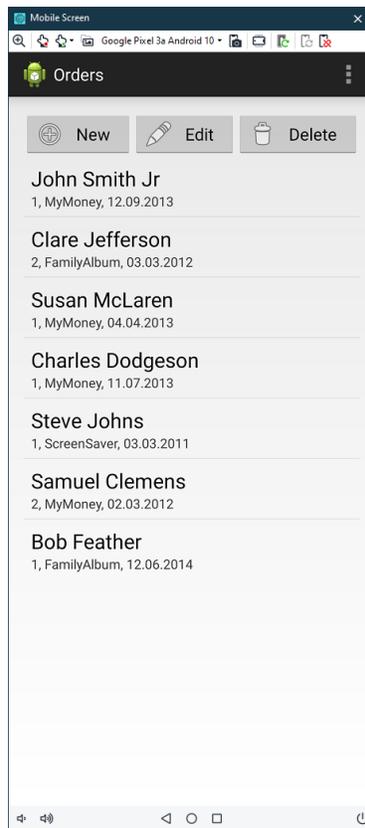
TestComplete は選択されたデバイスへ接続して、デバイスに Orders アプリケーションをインストールし、アプリケーションのテスト セッションを開きます。TestComplete の **[Mobile Screen]** ウィンドウに接続デバイスの画面が表示されます。テストするアプリケーションのユーザー アクションをレコーディングするには、そのウィンドウでアプリケーションを操作します。

5. テストのレコーディング

注意: レコーディング中に TestComplete ヘルプに切り替えしないでください。レコーディング エンジンはすべてのユーザー アクションをトレースし、レコーディングします。つまり、レコーディングしたテストに、この "切り替え" をシミュレートするコマンドが含まれてしまいます。

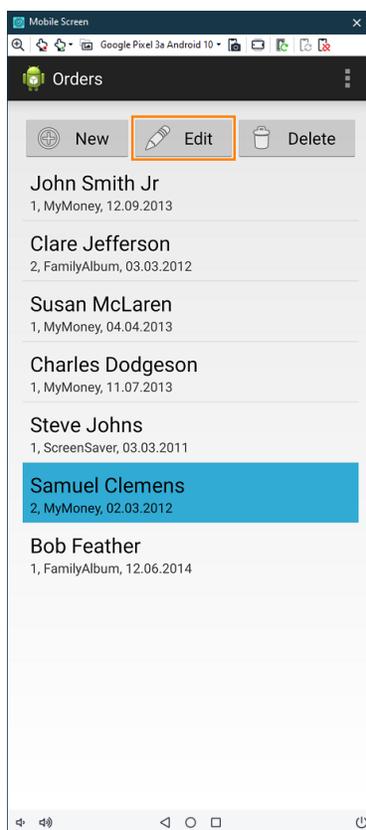
操作手順は、レコーディングを開始する前に印刷しておくことができます。あるいは、2 つのモニターがある場合、TestComplete ヘルプをもう一方のモニターへ移動しておくことができます。

1. **[Mobile Screen]** にアプリケーションの初期ウィンドウが表示されるまで待機します。

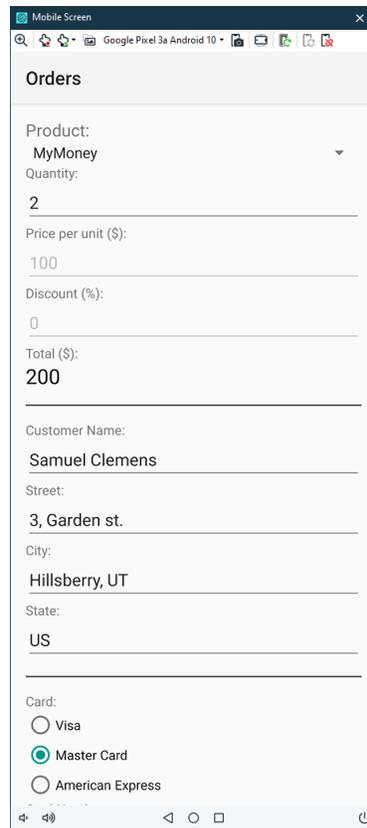


アプリケーションが開始されない場合は、手動で起動します。

2. [Mobile Screen] ウィンドウで Samuel Clemens の注文をクリックして、[Edit] ボタンをクリックします。



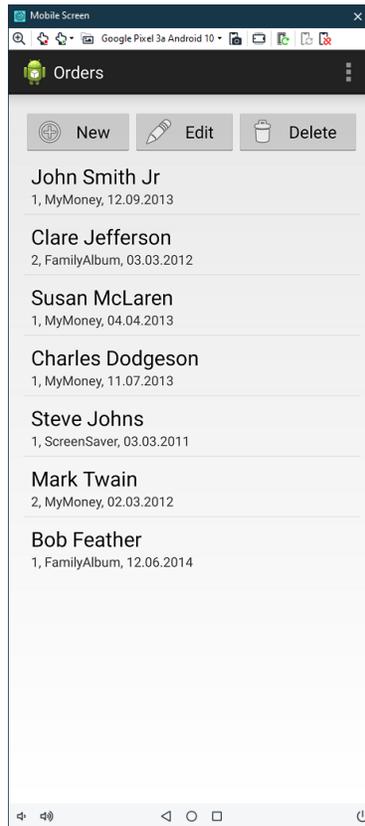
Orders アプリケーションが編集モードに切り替わります。



3. 注文の詳細情報で顧客名を変更してみましょう。

Samuel Clemens を消去して、*Mark Twain* と入力し、**Enter** キーを押します。デスクトップのキーボードを使用して、[Mobile Screen] ウィンドウでテキストを入力します。

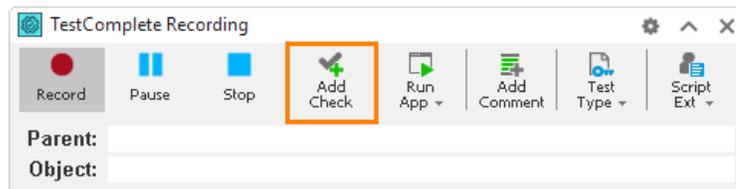
4. **[Edit Order]** ページをスクロール ダウンして、**[OK]** をクリックします。これにより、注文の変更が保存され、注文リストに戻ります。



5. テストに比較コマンドを挿入します。この比較コマンドは、アプリケーションの顧客リストに変更した名前 (*Mark Twain*) が含まれることを検証します。

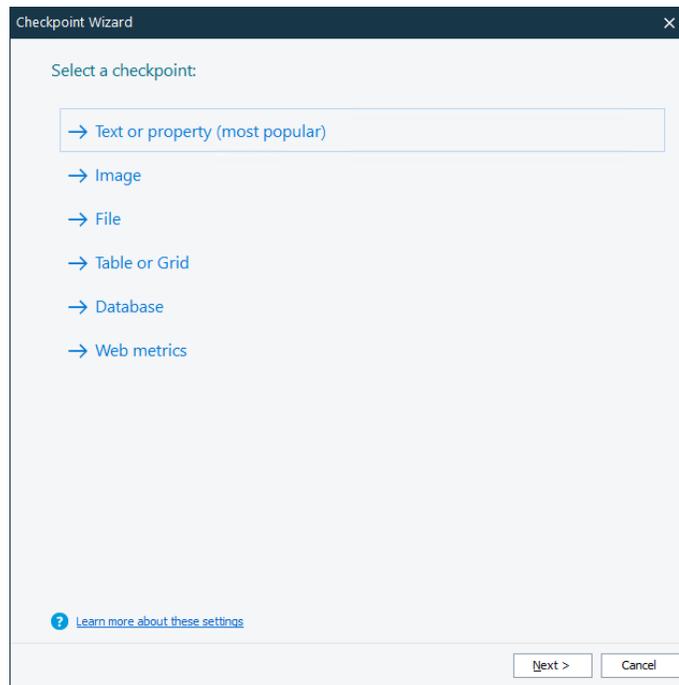
比較コマンドを**チェックポイント**と呼びます。TestComplete には、異なる種類のデータを検証するさまざまな種類のチェックポイントが用意されています。最もよく使用されるチェックポイントの 1 つは、**プロパティ チェックポイント**です。アプリケーション コントロールのデータのチェックに使用されます。このチュートリアルでは、このチェックポイントを使用して **[Customer Name]** テキスト ボックスの値を検証します。

- a. [Recording] ツールバーの  **[Add Check]** をクリックします。



[Checkpoint] ウィザードが呼び出されます。このウィザードを利用して、チェックポイントを作成します。

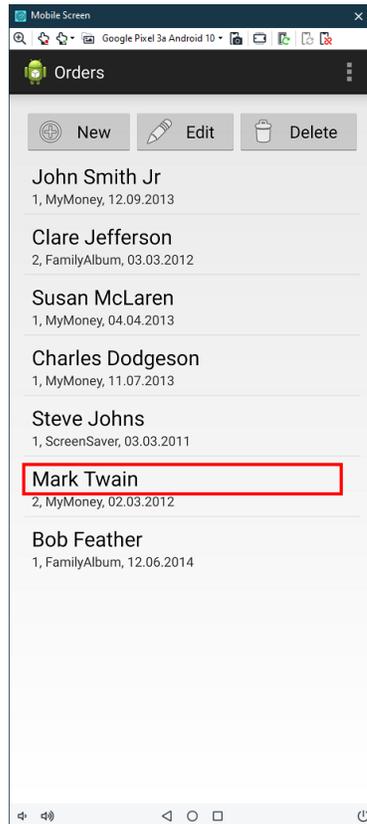
- b. ウィザードの最初のページで、**[Text or property]** をクリックします。



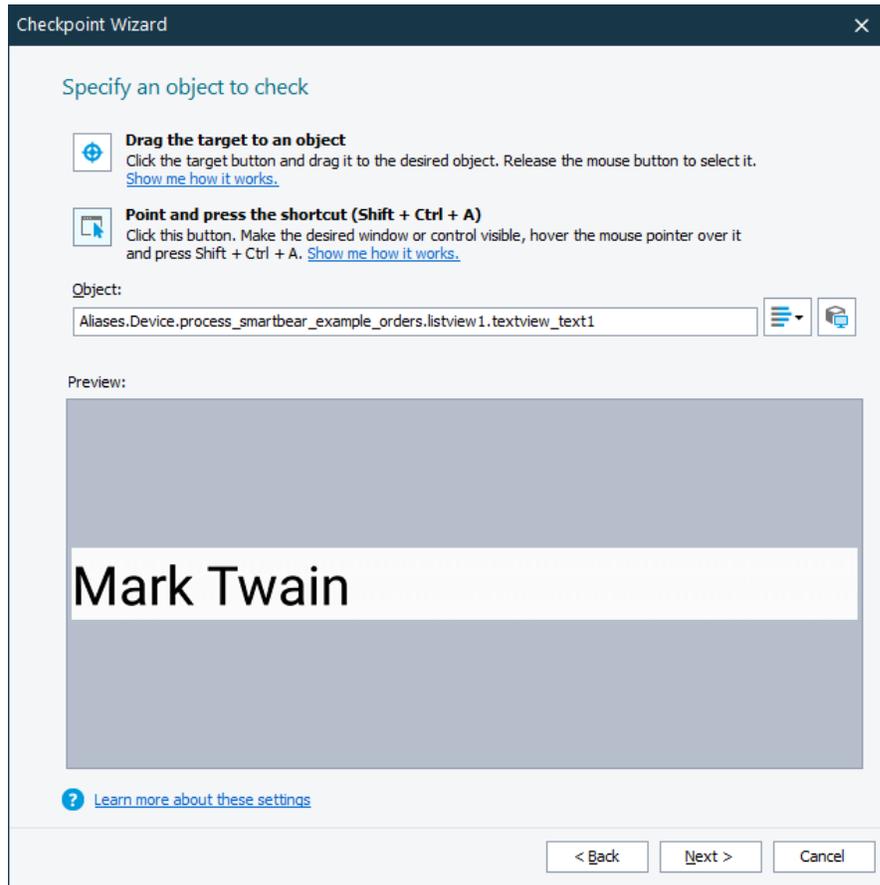
- c. ターゲットアイコン (⊕) をマウスの左ボタンでクリックして押したままにします。

ウィザードが最小化されるまで待ってから、そのアイコンを **Orders** アプリケーションの注文リストにドラッグします。ドラッグしている間、マウスカーソル下のコントロールとウィンドウが赤いフレームでハイライトされます。

[ターゲットアイコン] を **Mark Twain** の注文の上に移動し、その項目全体が赤いフレームでハイライトされたら、マウスのボタンを離します。

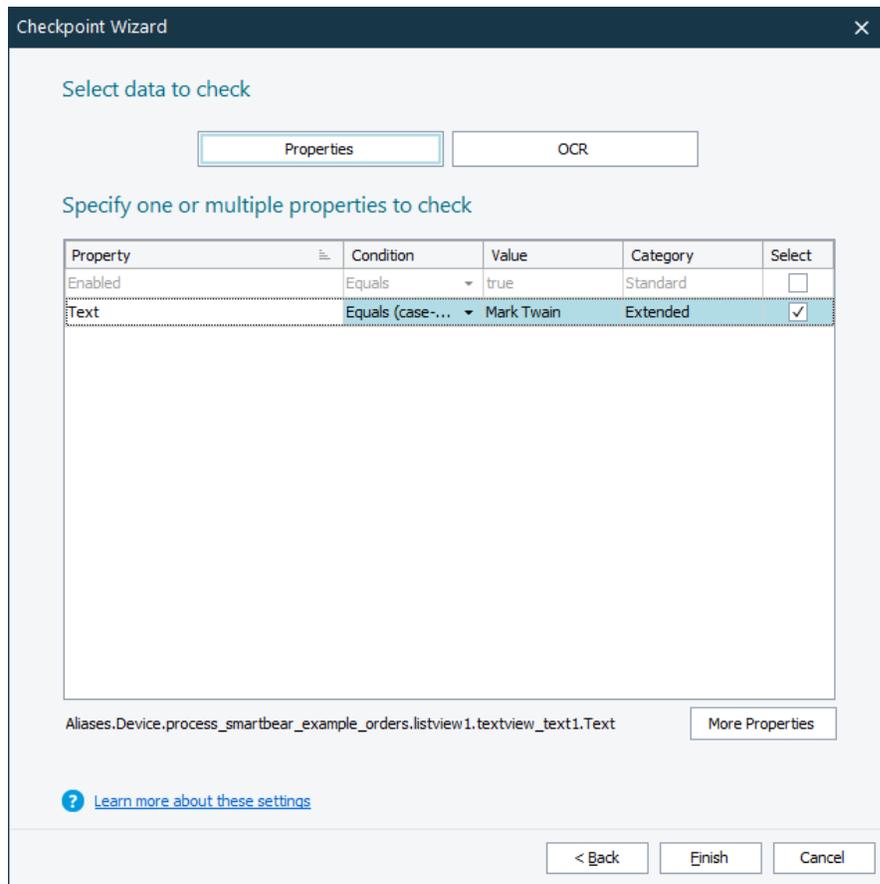


- d. マウスのボタンを離すと、ウィザードに戻り、選択したオブジェクトの名前が [Object] ボックスに表示され、その下にオブジェクトの画像が表示されます。



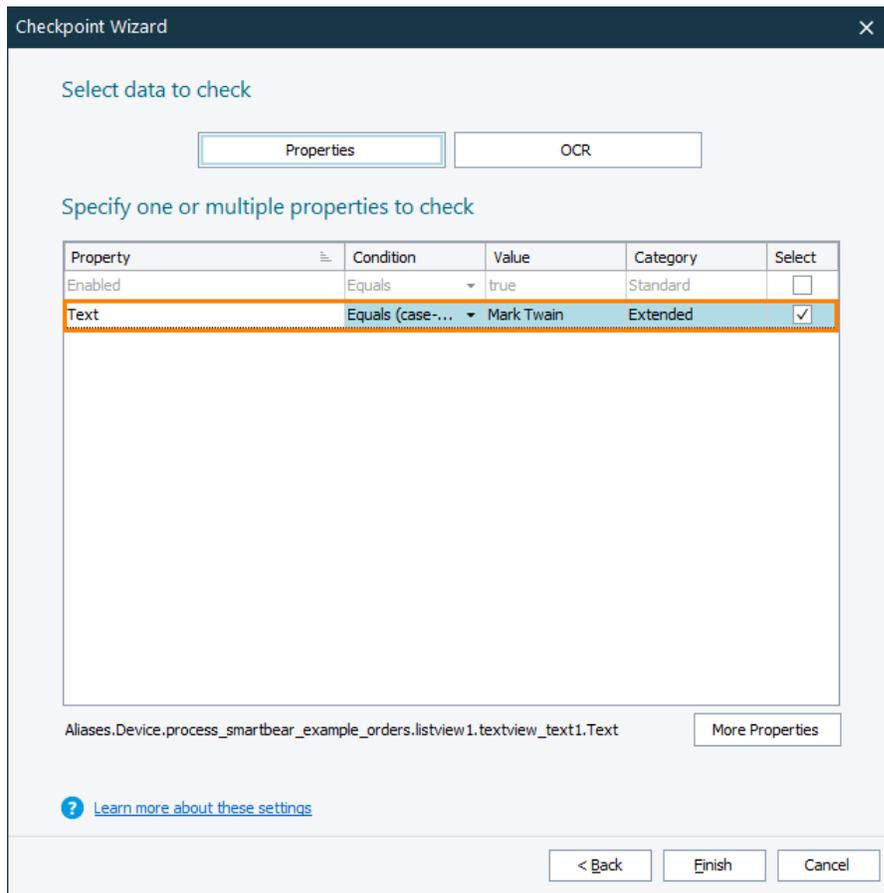
[Next] をクリックします。

- e. ウィザードの次のページには、チェックのための提案されたプロパティのリストが表示されます。



このリストには、テストするアプリケーションによって定義されたプロパティと TestComplete によって提供されるプロパティが含まれます。利用可能なすべてのプロパティを表示するには、**[More Properties]** をクリックします。

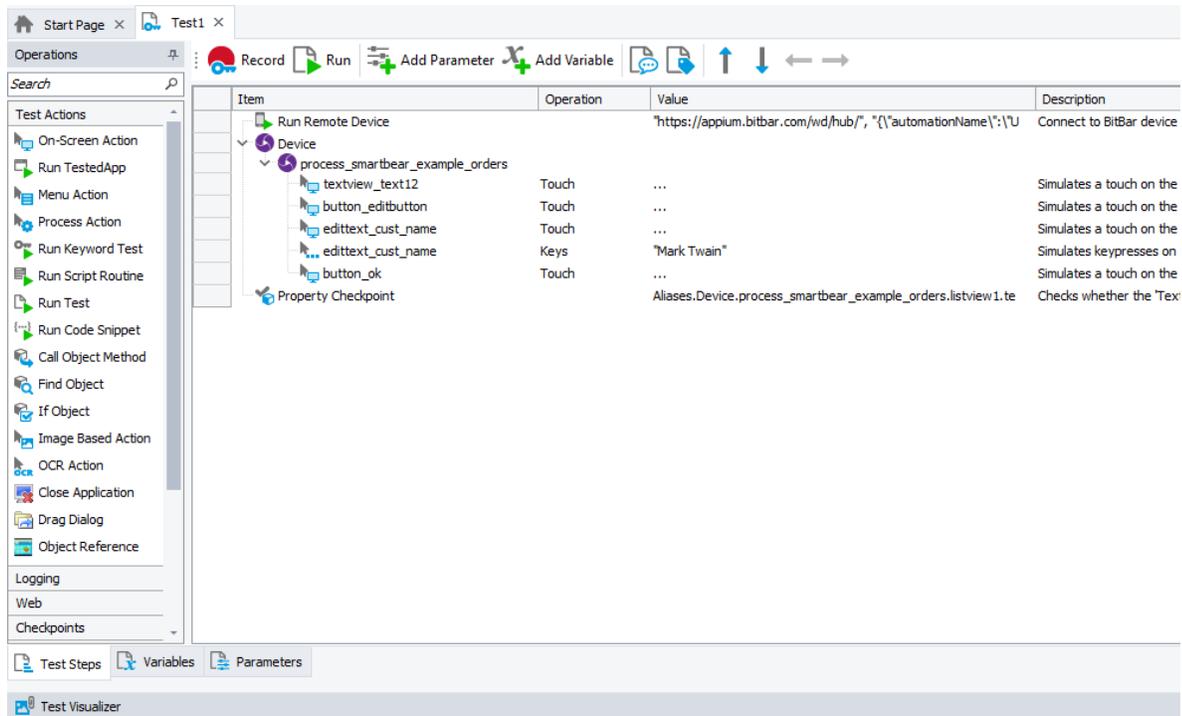
- f. テーブルで、**Text** プロパティが選択されていることを確認します。**[Condition]** カラムで **[Equals (case-sensitive)]** 条件が選択されていることを確認します。



- g. **[Finish]** をクリックしてチェックポイント作成を完了します。レコーディングしたテストにチェックポイントコマンドが追加されます。
6. **[Recording]** ツールバーの **[Stop]** を押して、レコーディングを停止します。レコーディングしたテストコマンドが処理され、キーワードテストに保存されます。

6. レコーディングしたテストの分析

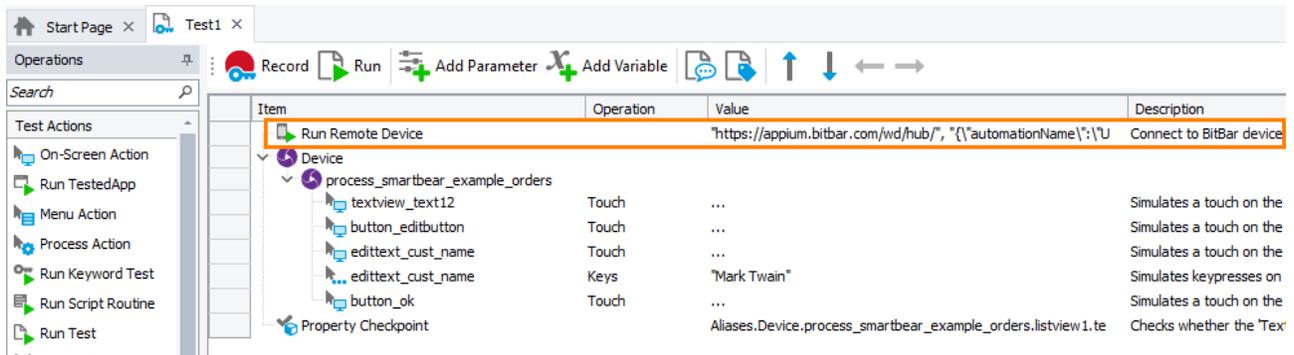
レコーディング終了後、TestComplete はレコーディングしたキーワードを Keyword Test エディターで開きます。



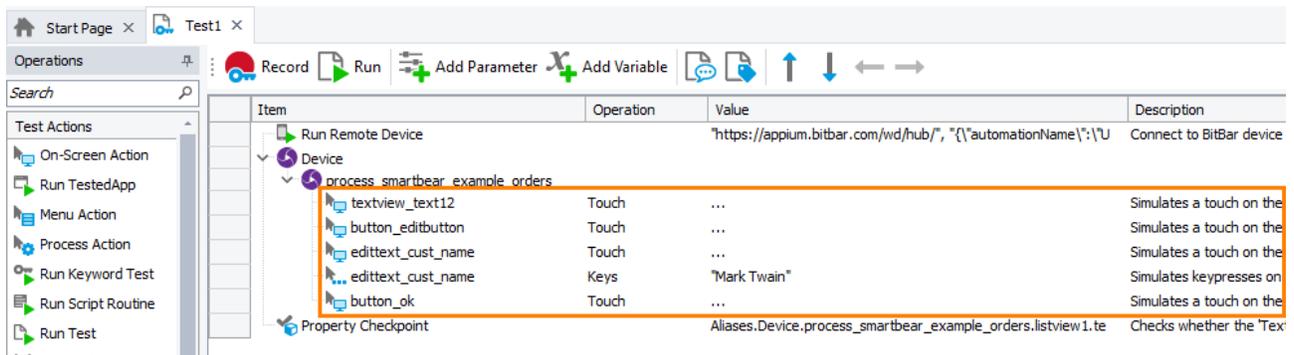
レコーディングしたテストは上記の画像に示すテストのようになります。実行環境によってはこれとは異なることがあります。たとえば、不要なタッチ操作が含まれていることがあります。

このテストには、レコーディング中に **Orders** アプリケーション上で実行した動作に対応するコマンドが含まれています。このテストコマンドを **operation** と呼びます。

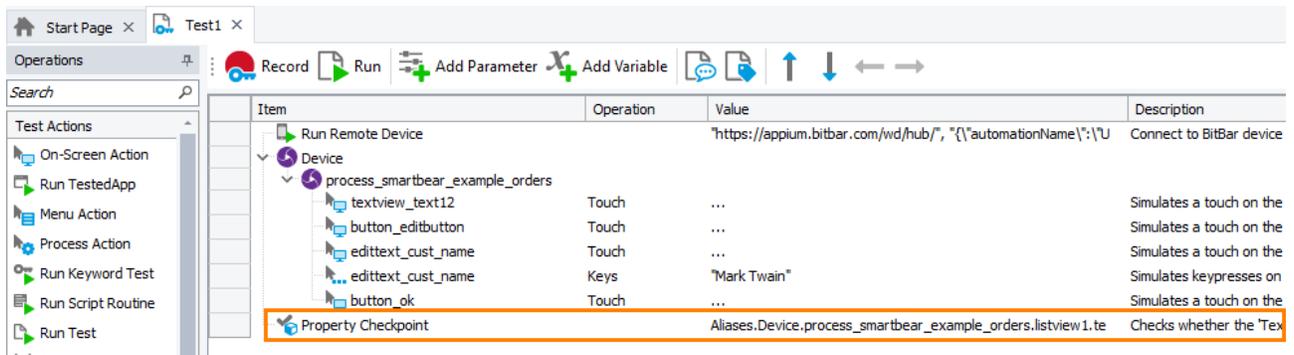
このテストの最初の operation は、**Run Remote Device** です。この operation は、BitBar モバイル デバイス クラウドのモバイル デバイスに接続して、Orders アプリケーションを指定したデバイスに展開し、アプリケーションのテストセッションを開きます。以降のテスト操作はすべてこのデバイスに関するものです。



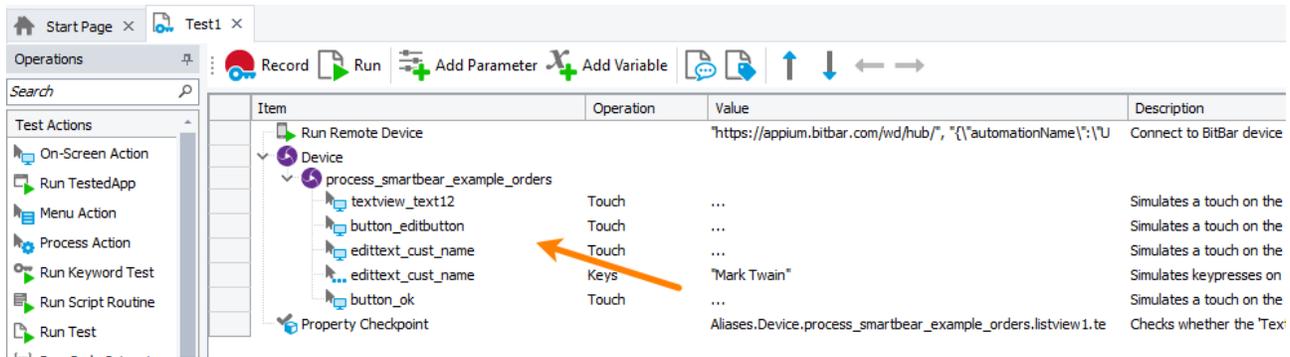
その後、アプリケーションでのユーザー アクションをシミュレートする operation が続きます。これらの operation は、[Edit] ボタンをクリックし、注文リストから項目を選択し、テキストフィールドの値を変更し、変更を保存して [Done] ボタンをクリックします。



最後に、テストレコーディング中に追加した比較動作です。

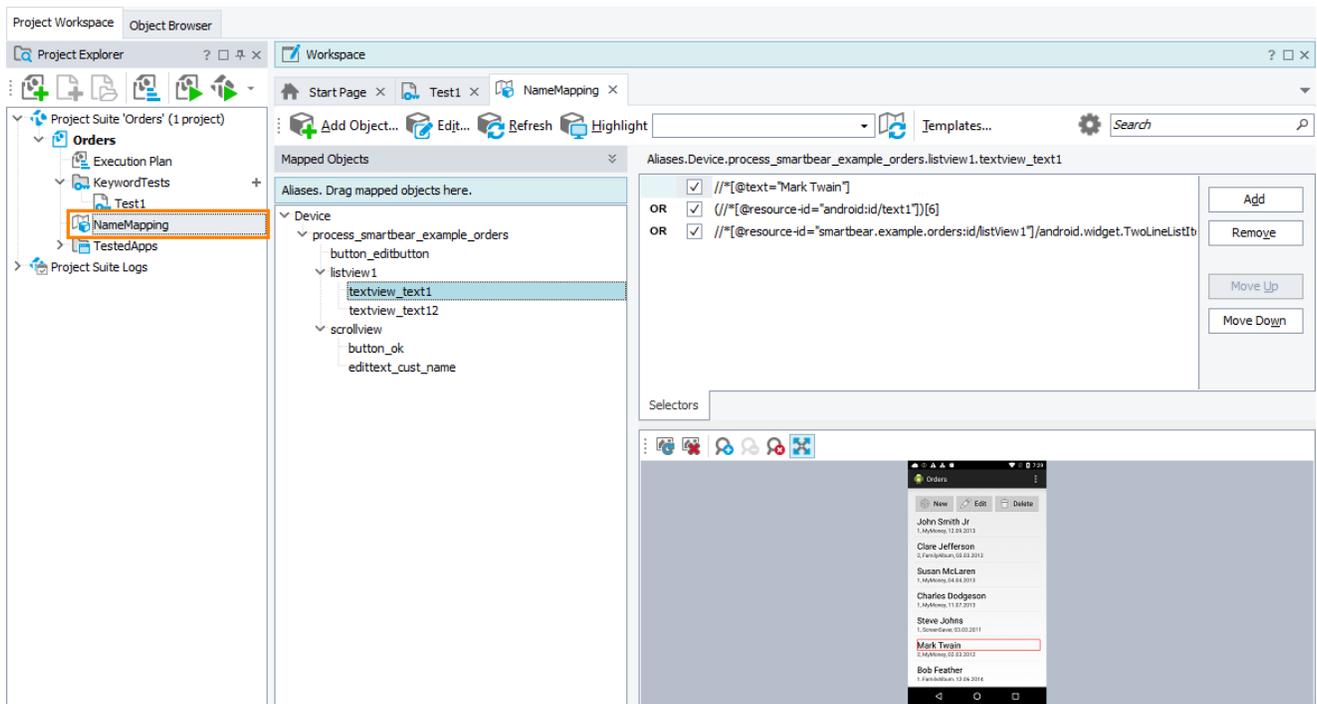


レコーディングしたテストのアプリケーションとオブジェクトの名前に注意する必要があります。



デフォルトでは、テストをレコーディングしている間、TestComplete は自動的にすべての項目を **Name Mapping** リポジトリに追加します。項目ごとに、TestComplete はテストの実行中にアプリケーション内のオブジェクトを見つけるために使用する検索式と、**alias** を格納します。**alias** はテスト内のオブジェクトを見つけるために使用する短い名前です。

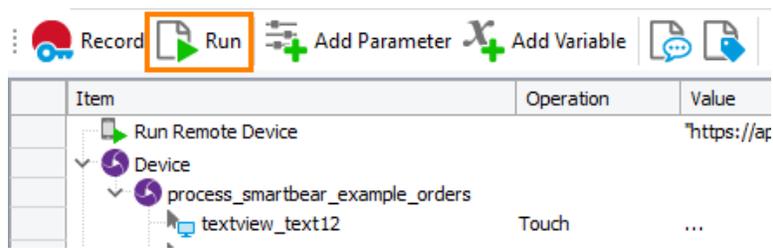
Project Explorer の **Name Mapping** 項目をダブルクリックして、Name Mapping リポジトリを開き、TestComplete がテストのレコーディング中にオブジェクト用に作成した認識パラメーターと **alias** を表示できます。



Name Mapping と alias の使用は、テストをより分かりやすく、安定したものにします。

7. テストの実行

レコーディングしたテストを実行するには、テスト エディターのツールバーにある  **[Run]** をクリックします。



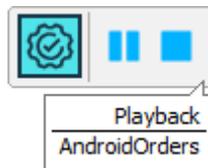
TestComplete は、BitBar デバイス クラウドの指定したデバイスに接続して、デバイスで Orders アプリケーションを起動し、レコーディングしたアクション (Samuel Clemens の注文を開き、顧客名を Mark Twain に変更する) を再生実行します。

注意: テスト実行中は、マウスを動かしたり、キーを押したりしないでください。TestComplete でシミュレートされる動作を妨げたり、テスト実行が正しく行われない可能性があります。

テスト実行の終了後、TestComplete はテスト結果を表示します。次の手順では、これを分析します。

テスト実行に関するいくつかの注意:

- テストの実行中、TestComplete は、画面の右上にインジケータを表示します。



このインジケータは、シミュレートしたテスト動作に関する通知をするためのメッセージを表示します。

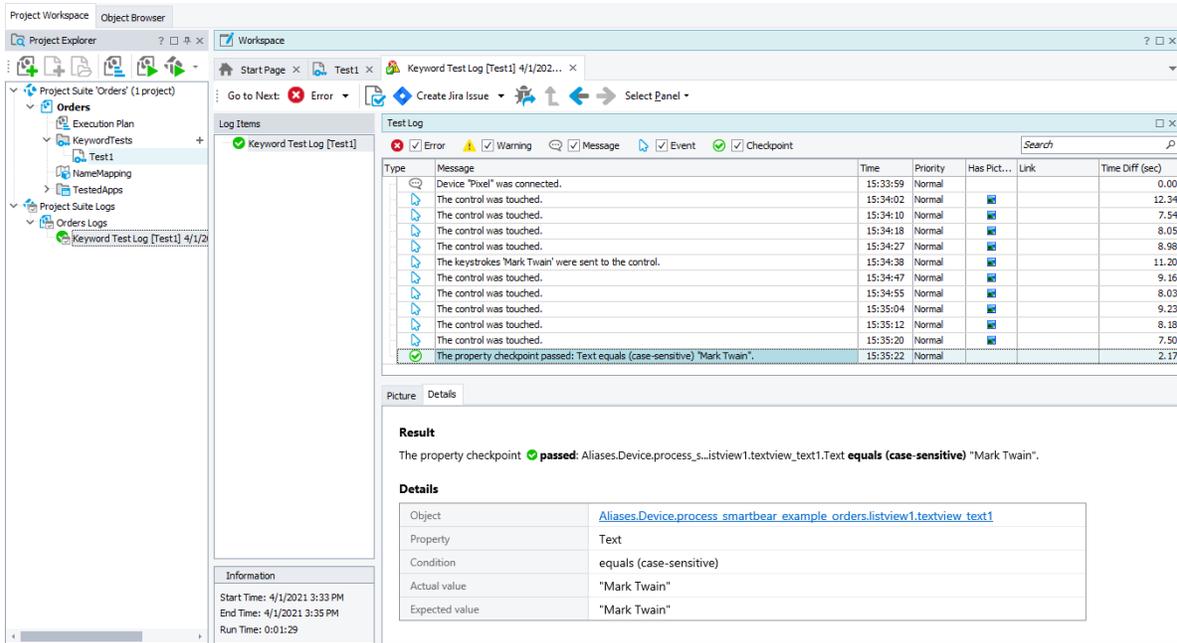
- TestComplete は、テストが終了するまでこのテスト コマンドを実行します。[Test Engine] ツールバーまたはインジケータの  **[Stop]** をクリックするか、または TestComplete メインメニューから **[Test] > [Stop]** を選択することで、テストの実行をいつでも停止することができます。

 **[Pause]** をクリックすることでテスト実行を一時停止することができます。一時停止の間は、必要な任意の動作を実行することが可能です。たとえば、テスト ログを表示したり、TestComplete の [Watch List] や [Locals] パネルあるいは **[Evaluate]** ダイアログを使用してテストの変数やオブジェクトを確認することが可能です (TestComplete ヘルプの「Debugging Tests」を参照)。

TestComplete のテスト実行、実行に影響するプロジェクト設定、テスト実施に関する詳細は、TestComplete ヘルプの「Running Tests」を参照してください。

8. テスト結果の分析

テスト終了後、すべてのテスト操作の結果を含むテスト ログが表示されます。この例では、ログは次のようになります。

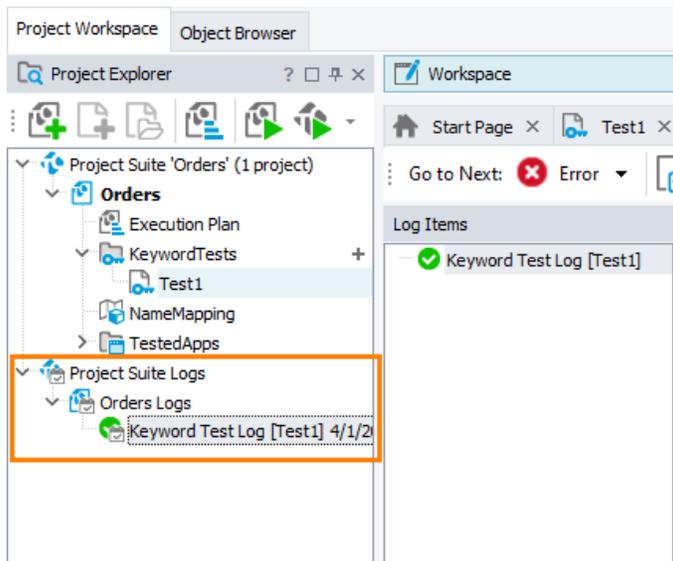


ログには、アクション、イベント、チェックポイントなどに関するさまざまなメッセージが含まれています。メッセージリストの上のツールバーを使用してメッセージをフィルタリングできます。ログメッセージをダブルクリックすると、そのメッセージを出力したテスト操作が表示されます。これは、エラーの原因となった操作を把握するのに便利です。

[Picture] パネルには、選択したテスト コマンドが実行されると、アプリケーションの状態を示す画像が含まれます。これらの画像は、テスト実行中のアプリケーションの動作を理解し、エラーを迅速に見つけるのに役立ちます。

[Details] パネルには、実行された操作の詳細が表示されます。たとえば、プロパティ チェックポイントの [Details] には、チェックタイプ (equals、contains、matches a regular expression など) およびその他の詳細が含まれます。

すべてのログが **Project Explorer** の **[Project Suite Logs] > [ProjectName Logs]** にあるため、以前のログを見ることもできます。



エラーの解決

テストが失敗することがあります。これにはいくつかの可能性があります。たとえば、開発者がアプリケーションの動作を変更したり、アプリケーション コントロールの認識属性が変わりテスト エンジンが必要なオブジェクトを見つけられなくなったりする可能性があります。

初めてのユーザーが直面する最も典型的なものは、テスト作成中と再生実行中でアプリケーションの状態が異なることです。この問題を避けるには、テストを作成したときと同じ初期状態でテスト実行しているか確認してください。以下に例を示します。

- テストするアプリケーションがテストを記録する前に実行されていた場合は、テストを実行する前に実行している必要があります。
- テストアクションがアプリケーションの特定の画面で実行された場合は、テストを実行するときにもその画面を開く必要があります。
- アプリケーション内のデータを編集して保存した場合は、変更を元に戻す必要があります。

特定のアプリケーションの応答を待つ必要があるため、テスト エンジンがアプリケーション オブジェクトを検出できないことがあります。この問題を解決するには、**Delay** コマンドを挿入し、テキストの実行を一時停止する時間を指定します。

- キーワードテストでは、指定した *Delay Time* (遅延時間) パラメーターで **Delay** オペレーションを使用します。
- スクリプトでは、ループ本体に `aqUtils.Delay` メソッドを使用します。

<C:\work\reference\program-objects\aqutils\delay.htm>

エラーの原因を探し、典型的な問題を解決するための情報は、TestComplete ヘルプの「[Handling Playback Errors](#)」を参照してください。

iOS アプリケーションのテスト

このチュートリアルでは、TestComplete で iOS アプリケーションをテストする方法を説明します。これは、自動テストの一般原則に精通しており、TestComplete IDE についての最低限の知識を持っていることを前提としています。

! 初心者の方は、上記の自動テストの紹介を読むことをお勧めします。

システム要件

以下の要件を満たす場合、このチュートリアルの手順に従って iOS アプリケーションのテストを作成できます。

- TestComplete がインストールされた Windows マシンと TestComplete Mobile モジュールのアクティブなライセンス。
- Appium サーバーが管理する iOS デバイスへのアクセス。次のいずれかです。
 - BitBar が提供するモバイル デバイス クラウド。このチュートリアルでは、こちらを使用します。無料の評価用 BitBar アカウントを作成して、モバイル デバイスにアクセスする方法を紹介します。
 - または -
 - ローカル マシンまたはローカル ネットワークのリモート マシンで実行しているプライベート Appium サーバー。プライベート Appium サーバーの使用法については、このチュートリアルではカバーしません。
- サンプル Orders アプリケーションをコンパイルするための、Xcode、iOS SDK、および iOS 開発ライセンスを備えた Mac マシン。

注意: このチュートリアルには、iTunes などの他のツールやユーティリティは必要ありません。

チュートリアルについて

評価用 BitBar アカウントの作成、BitBar デバイス クラウド上の iOS モバイル デバイスへの接続、テストするアプリケーションの展開、簡単なテストの作成と実行、結果の分析を行います。

1. 無料の BitBar アカウントの申し込み

TestComplete では、Appium サーバーが管理する iOS デバイスに接続して、自動テストを作成して実行できます。このチュートリアルでは、SmartBear のクラウド モバイル テスト サービス「BitBar」が提供する、すぐに使えるソリューションを使用します。

無料の評価用 BitBar アカウントを作成して使用方法を紹介します。評価用アカウントの申し込みは数分で完了し、必要なのは有効なメールアドレスのみです。

チュートリアルを開始する前に申し込むことも、スキップして後で実際にテストのレコーディングを開始する際に申し込むこともできます。

1. 評価用 BitBar アカウントの申し込み

1. 以下のリンクをクリックして、BitBar Web サイトで無料の評価用アカウントの申し込み手続きを行ってください。

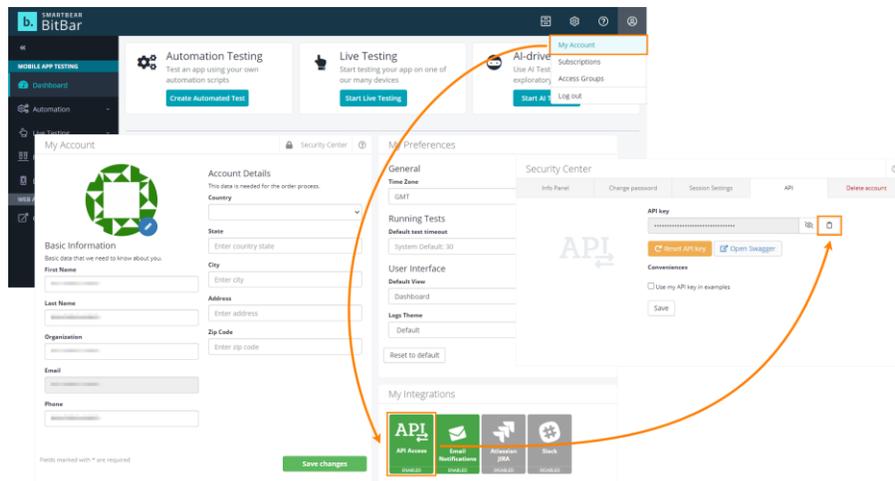
→ bitbar.com/signup

2. Web フォームの指示に従ってください。

2. BitBar API Key の入手

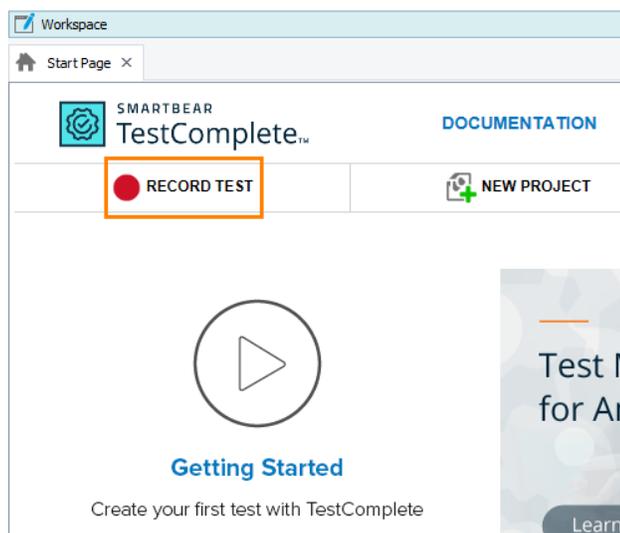
TestComplete のテストから BitBar リソースにアクセスするには、**BitBar API Key** を使用します。キーを取得するには

1. 作成した BitBar アカウントで BitBar にログインします。
2. ページの右上にある  > **[My Account]** をクリックします。
3. **[My Integrations]** セクションで **[API]** をクリックします。
4. 結果ページで  をクリックして、API Key をクリップボードにコピーします。



2. テストのレコーディング開始

1. TestComplete でプロジェクトあるいはプロジェクト スイートを開いている場合は、閉じてください。これをするには、**[File] > [Close]** を TestComplete のメインメニューから選択します。
2. TestComplete **スタート** ページに切り替えます。このページが隠れている場合、TestComplete の **[Help]** メニューから **[Start Page]** を選択します。



3. スタートページで、**[Record Test]** をクリックします。TestComplete は、**[Record Test]** ウィザードを表示します。

Record Test

Select project name, location and type of test to record.

Project name: *

Orders

Project location: *

C:\Users\tester\Documents\TestComplete 14 Projects\

Test type: ⓘ

Keyword

JavaScript

Python

[? Learn more about these settings](#)

Next > Cancel

4. ウィザードの最初のページで、プロジェクトの名前、場所、テストタイプを指定します。

[Project name] テキスト ボックスに *Orders* と入力します。

[Location] テキスト ボックスはデフォルトの値のままにします。

テスト タイプを選択します。キーワードテストまたは JavaScript、Python スクリプト テストのいずれかを作成することができます。

- キーワードテストは、マウスのクリック、テキストの入力など、ユーザーの操作を定義する一連のキーワードです。視覚的にキーワード テストを作成します。スクリプトの知識は必要ではありません。
- JavaScript と Python テストは、ユーザー アクションをシミュレートする命令を含むスクリプト関数です。

このチュートリアルでは、キーワード テストの作成方法を示します。**[Keyword]** をクリックします。

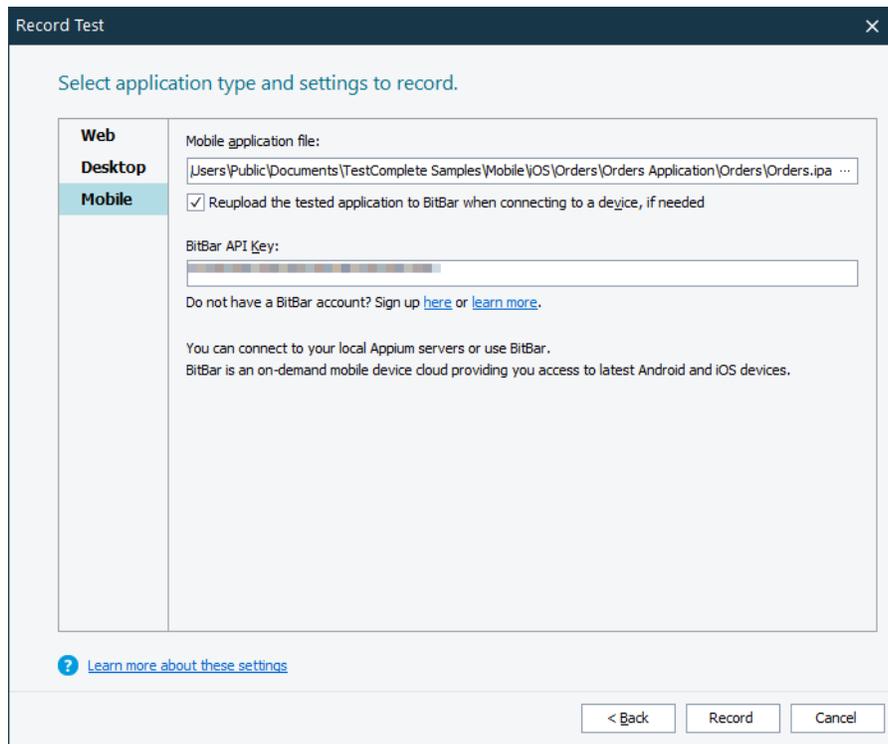
5. **[Next]** をクリックします。

ウィザードを使って、テストするアプリケーションをプロジェクトに追加するため、ウィザードで作業を続行します。

3. テストするアプリケーションの定義

モバイル デバイスで実行するアプリケーションのテストを作成するには、そのデバイスへアプリケーションを展開する必要があります。BitBar モバイル デバイス クラウド プロバイダーを使用する場合は、アプリケーション ファイル (*Orders.ipa*) をアカウントのファイル ライブラリにアップロードします。TestComplete からファイルをアップロードする最も簡単な方法は、テストのレコーディングの開始時などに、TestComplete プロジェクトにファイルを追加することです。

1. ウィザードでは、テストするアプリケーションを選択できる 2 番目のページが表示されます。



2. Orders アプリケーションは、.ipa ファイルとして出荷される iOS アプリケーションなので、Mobile アプリケーションのカテゴリに該当します。

[Mobile] をクリックします。

3. **[Mobile application file]** ダイアログで、[...] ボタンをクリックします。表示される **[Select Tested Application]** ダイアログで、Orders.ipa ファイルの場所を見つけます。

[Reupload the tested application to BitBar when connecting to a device, if needed] チェックボックスはデフォルトのままオンにします。

4. すでに BitBar アカウントを持っている場合は、アカウントに割り当てられた API Key を **[BitBar API Key]** テキストボックスに入力します。

API Key の取得方法は、前述の手順を参照してください。

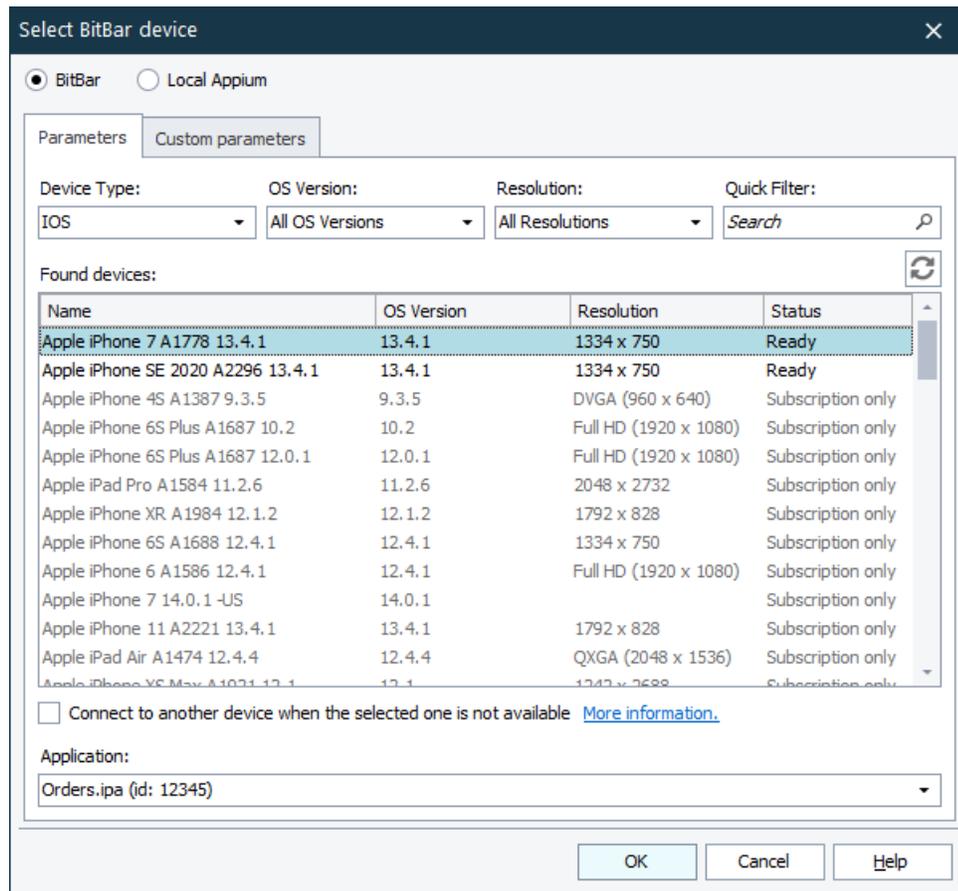
このチュートリアルステップ 1 をスキップした場合は、ウィザードの **[Sign up]** リンクをクリックすると無料の評価用アカウントを申し込むことができます。

5. **[Record]** をクリックし、プロジェクトの作成を完了し、レコーディングを開始します。

4. デバイスへ接続してテスト セッションを開く

レコーディングを開始する前に、テストをレコーディングするデバイスを選択して、Orders アプリケーションのテストセッションを開きます。

1. **[Select BitBar Device]** ダイアログが表示されます。BitBar が提供するすべてのモバイル デバイスがリストされます。



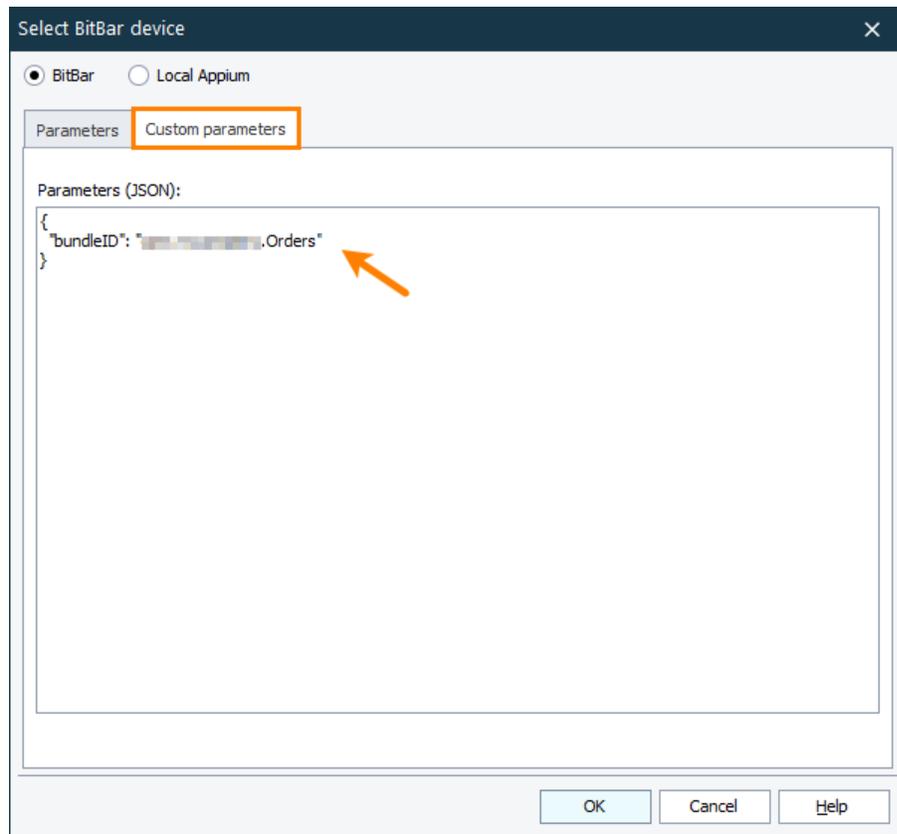
2. 任意の利用可能な iOS デバイスを選択します。デバイスが利用可能かどうかは **[Status]** カラムで確認できます。

評価ユーザー向けにいくつかのデバイスが用意されています。このチュートリアルでは、iPhone 7 を使用します。

適切なデバイスをすばやく見つけるには、プラットフォーム、バージョン、名前 でデバイス リストをフィルタリングできます。

3. **Orders** アプリケーションのテスト セッションを開くには、アプリケーションがターゲット デバイスにアップロードされている必要があります。前のステップで有効な **.ipa** ファイルを指定した場合、TestComplete はファイルを自動的にアップロードします。
4. レコーディングの開始時に選択したデバイスでテストするアプリケーションを自動的に開始するには、アプリケーションのバンドル ID を指定します。これは、指定した **.ipa** ファイルのコンパイルに使用されたバンドル ID です。

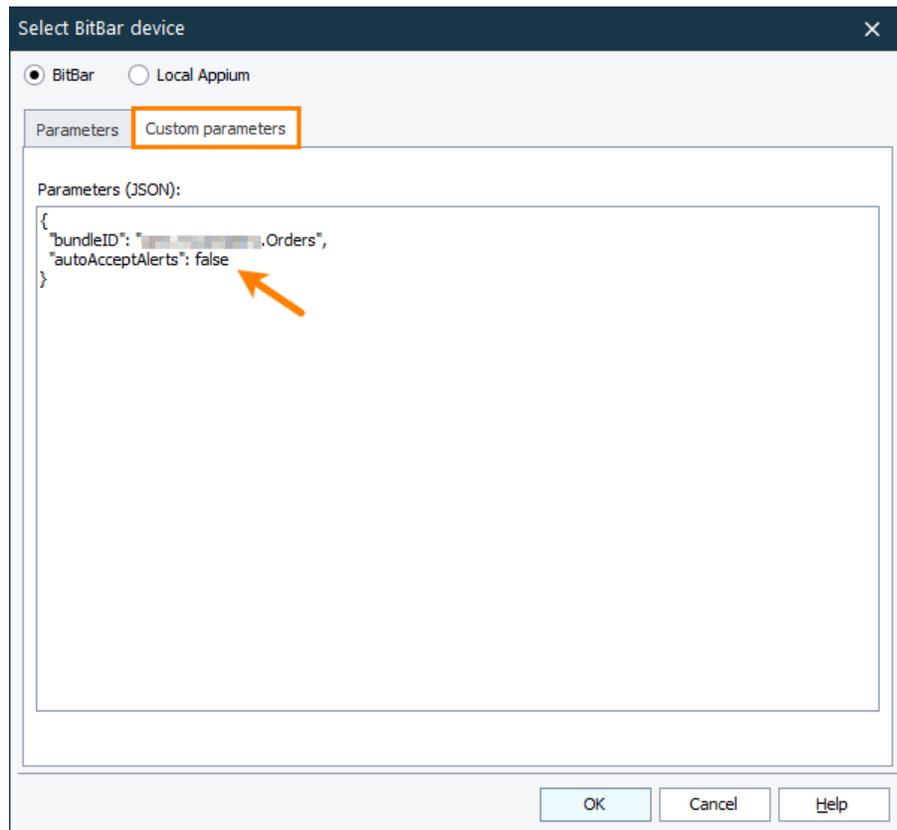
[Custom Parameters] タブに切り替えて、パラメーターリストに **bundleId** を追加します。



このステップをスキップすると、手動でアプリケーションを起動する必要があります。

5. デフォルトでは、**BitBar** クラウドで実行するモバイル デバイスは、すべての通知、メッセージ、およびアラートを自動的に閉じるように設定されています。このチュートリアルでは、**Orders** アプリケーションが表示するアラートに対するユーザー アクションをレコーディングするため、デバイスがアラートを自動的に処理しないように設定します。

[Custom Parameters] タブで、パラメーターリストに `autoAcceptAlerts` または `autoDismissAlerts` を追加して `false` に設定します。



- たとえば、次のようにカスタム パラメーターを設定します。テストするアプリケーションのコンパイルに使用した適切なバンドル ID を指定してください。

```
{
  "bundleId": "<Your_bundle_id>",
  "autoAcceptAlerts": false
}
```

- [OK]** をクリックして続行します。

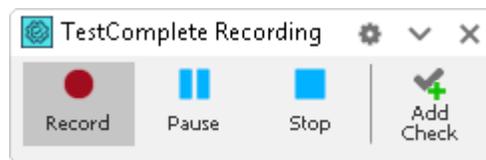
TestComplete は選択されたデバイスへ接続して、デバイスに Orders アプリケーションをインストールし、アプリケーションのテスト セッションを開きます。TestComplete の **[Mobile Screen]** ウィンドウに接続デバイスの画面が表示されます。テストするアプリケーションのユーザー アクションをレコーディングするには、そのウィンドウでアプリケーションを操作します。

5. テストのレコーディング

- !** テストのレコーディング中に TestComplete ヘルプに切り替えしないでください。レコーディングエンジンはすべてのユーザー アクションをトレースし、レコーディングするため、レコーディングしたテストにこの動作が含まれてしまいます。

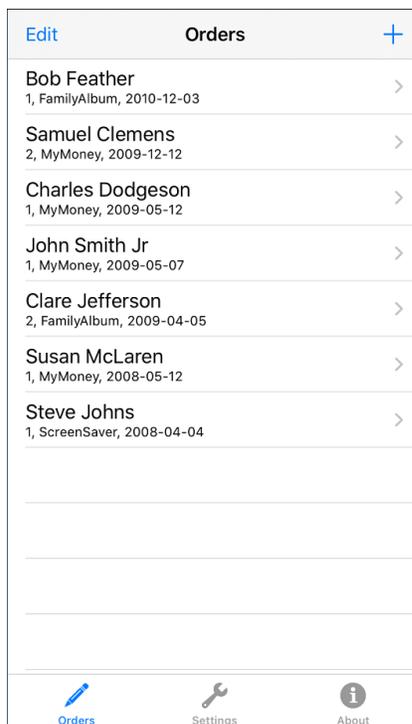
操作手順は、レコーディングを開始する前に印刷しておくことができます。あるいは、モニターが 2 つある場合は、TestComplete ヘルプをもう一方のモニターへ移動しておくことができます。

1. TestComplete は、テストのレコーディングを開始し、レコーディング モードに切り替わり、画面上に [Recording] ツールバーが表示されます。

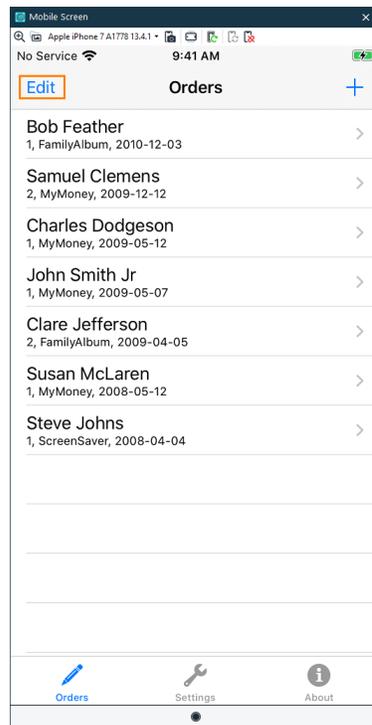


2. **[Mobile Screen]** にアプリケーションの初期ウィンドウが表示されるまで待機します。

注意: バンドル ID の指定をスキップした場合、アプリケーションは自動的に開始しません。手動で起動してください。

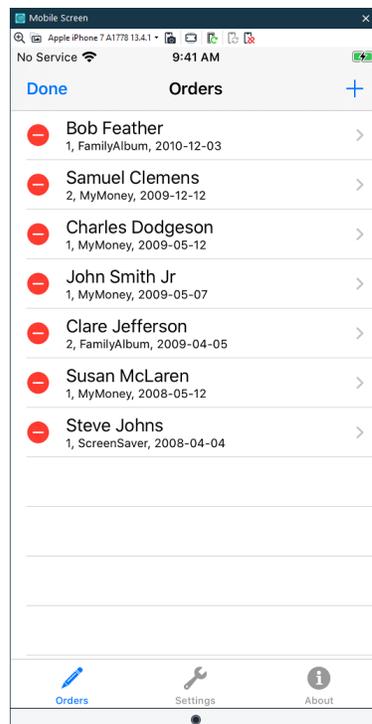


3. [Mobile Screen] ウィンドウで [Edit] ボタンをクリックします。



Orders アプリケーションが編集モードに切り替わります。

4. リストの2つめにある Samuel Clemens の注文をクリックします。[Edit Order] パネルが表示されます。



- 注文の詳細情報で顧客名を変更してみましょう。

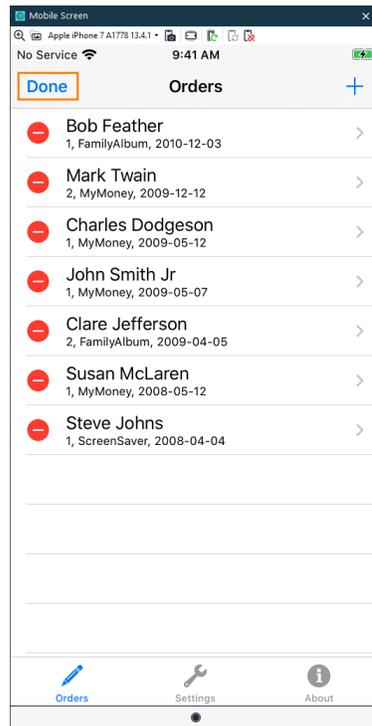
Samuel Clemens を消去して、*Mark Twain* と入力し、**Enter** キーを押します。デスクトップのキーボードを使用して、[Mobile Screen] ウィンドウでテキストを入力します。

- [Edit Order] パネルの **[Save]** ボタンをクリックし、変更を確定します。

- iPhone デバイス (一度に 1 つのパネルしか表示されない) でレコーディングする場合は、ナビゲーションバーにある **Orders** ボタンをタッチすると、[Orders List] パネルが表示されます。

iPad デバイスは両方のパネルを同時に表示できるため、このアクションは必要ありません。

- [Done]** ボタンをクリックして編集モードを終了します。

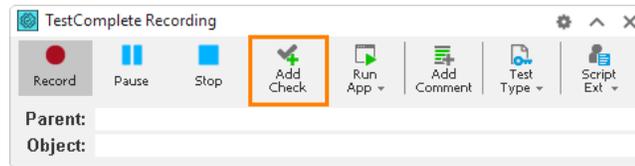


注文リストが通常モードに戻ります。

- テストに比較コマンドを挿入します。この比較コマンドは、アプリケーションの顧客リストに変更した名前 (*Mark Twain*) が表示されることを検証します。

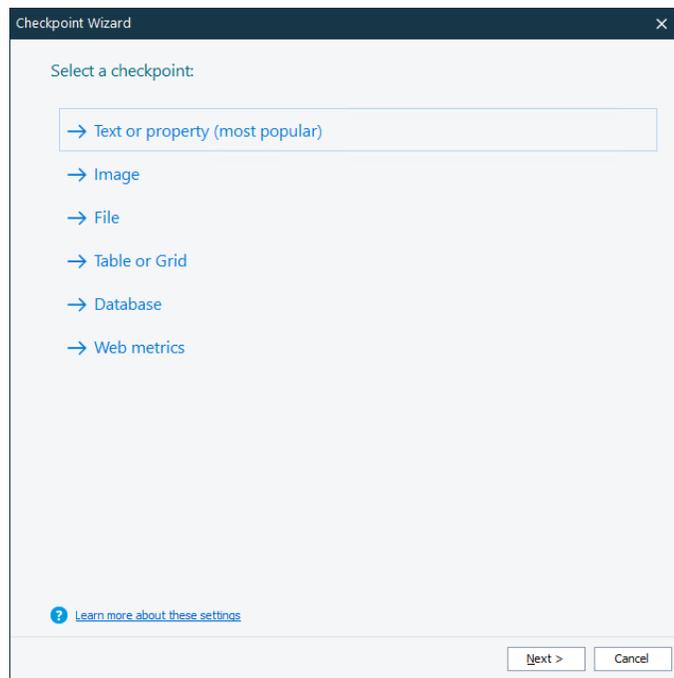
比較コマンドを**チェックポイント**と呼びます。TestComplete には、異なる種類のデータを検証するさまざまな種類のチェックポイントが用意されています。最もよく使用されるチェックポイントの 1 つは、**プロパティ チェックポイント**です。アプリケーション コントロールのデータのチェックに使用されます。このチュートリアルでは、このプロパティ チェックポイントを使用します。

- [Recording] ツールバーの  **[Add Check]** をクリックします。



[Checkpoint] ウィザードが呼び出されます。このウィザードを利用して、チェックポイントを作成します。

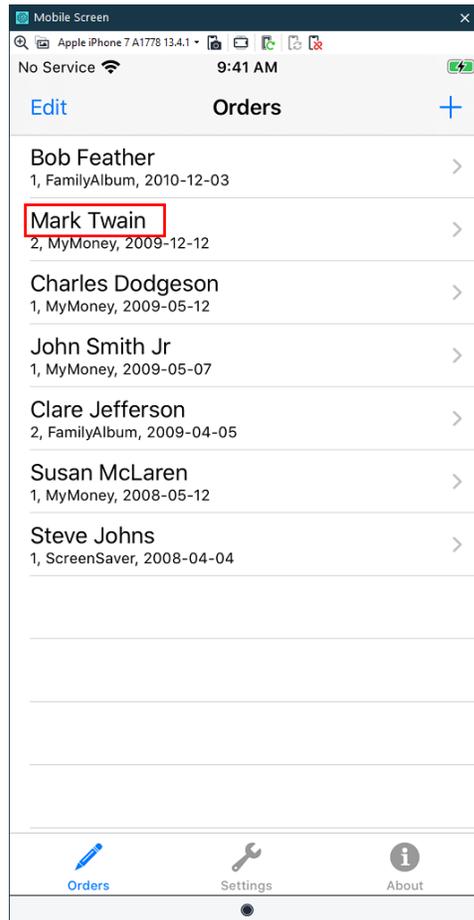
- ウィザードの最初のページで、**[Text or property]** をクリックします。



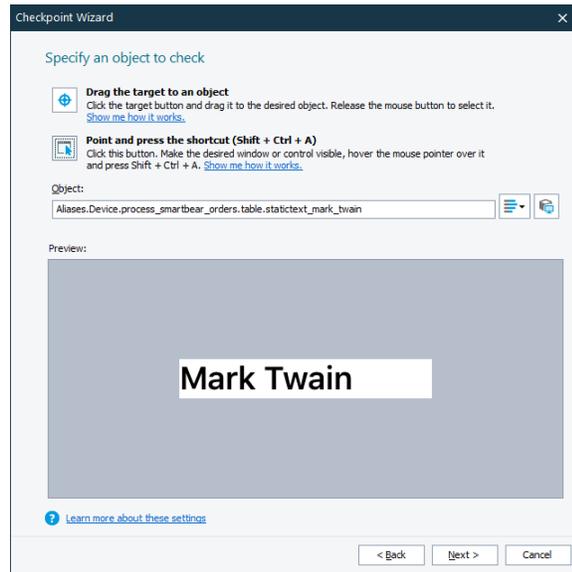
- ターゲット アイコン () をマウスの左ボタンでクリックして押したままにします。

ウィザードが最小化されるまで待ってから、そのアイコンを **Orders** アプリケーションの注文リストにドラッグします。ドラッグしている間、マウスカーソル下のコントロールとウィンドウが赤いフレームでハイライトされます。

[ターゲットアイコン]を Mark Twain の注文の上に移動し、その項目全体が赤いフレームでハイライトされたら、マウスのボタンを離します。

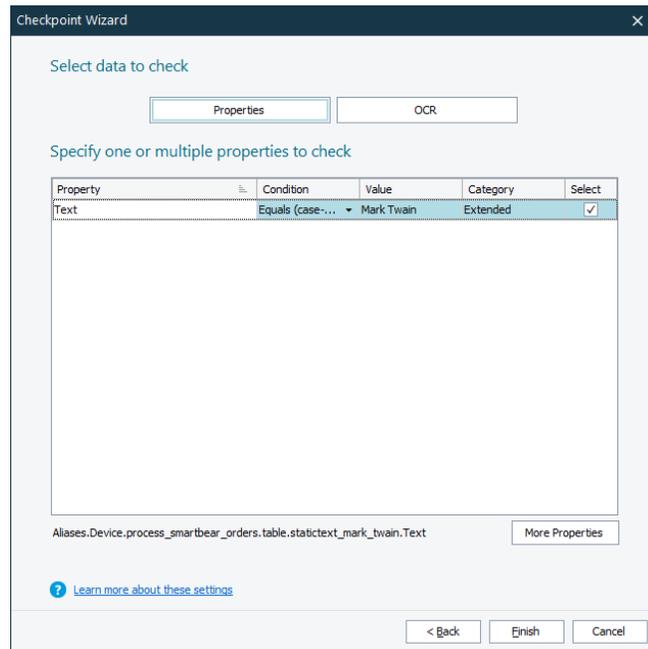


- マウスのボタンを離すと、ウィザードに戻り、選択したオブジェクトの名前が [Object] ボックスに表示され、その下にオブジェクトの画像が表示されます。



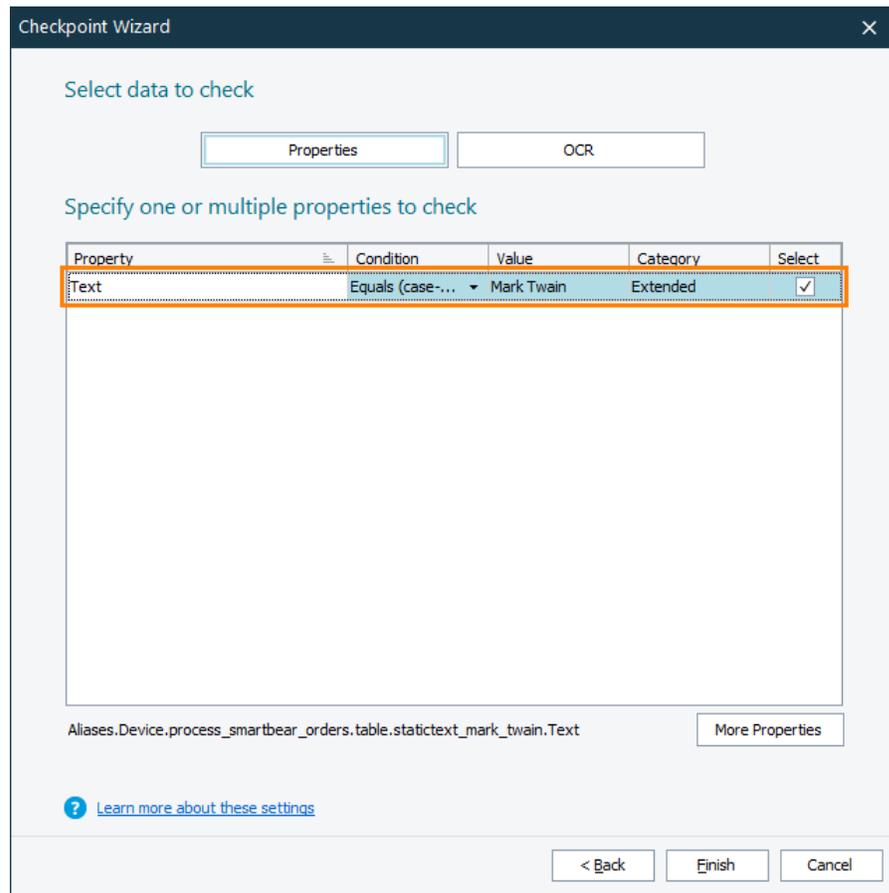
[Next] をクリックします。

- ウィザードの次のページには、チェックのための提案されたプロパティのリストが表示されます。



このリストには、テストするアプリケーションによって定義されたプロパティと TestComplete によって提供されるプロパティが含まれます。利用可能なすべてのプロパティを表示するには、**[More Properties]** をクリックします。

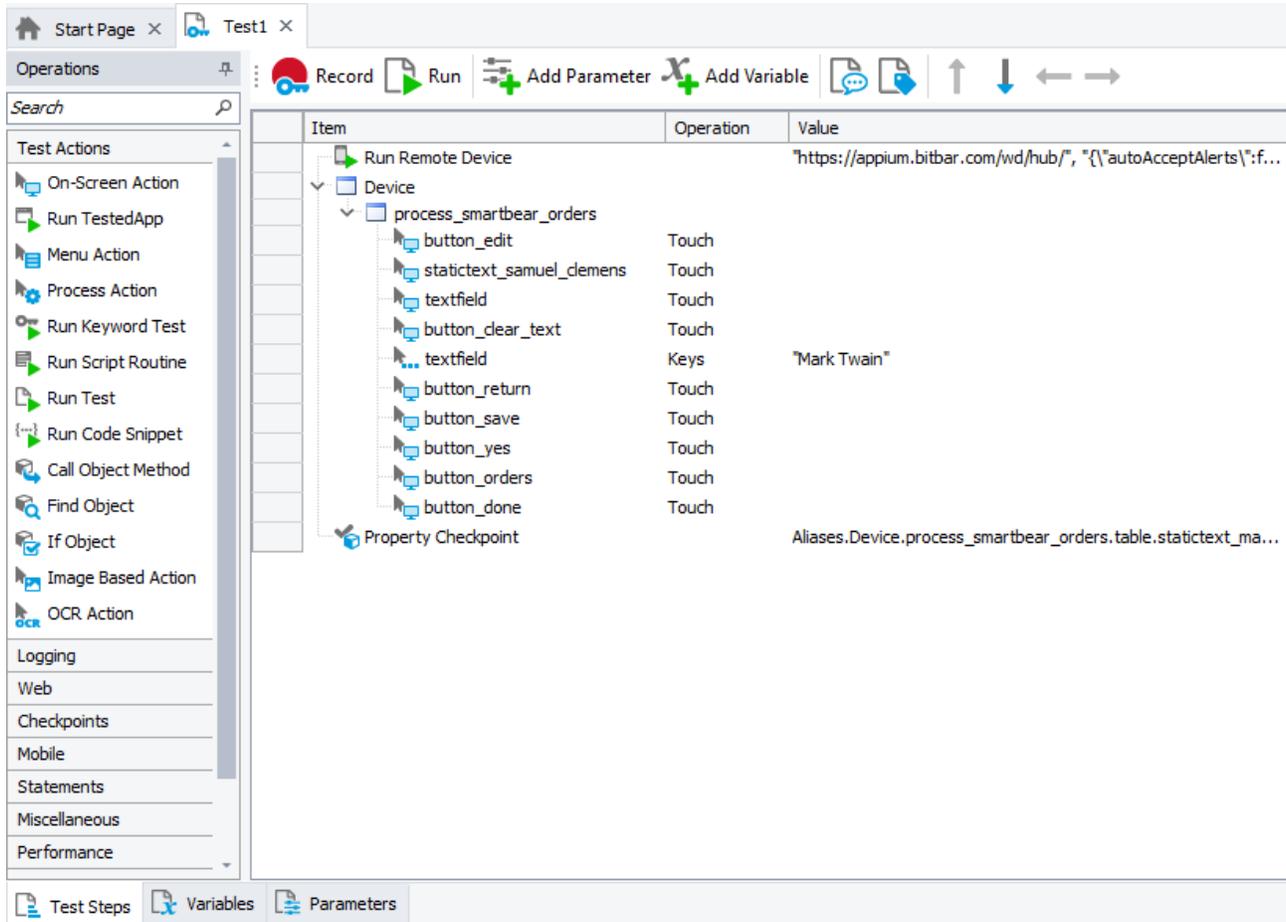
- **[Text]** プロパティが選択され、**[Condition]** カラムで **[Equals (case-sensitive)]** 条件が選択されていることを確認します。



- **[Finish]** をクリックしてチェックポイント作成を完了します。レコーディングしたテストにチェックポイントコマンドが追加されます。
10. **[Recording]** ツールバーの **[Stop]** を押して、レコーディングを停止します。レコーディングしたテストコマンドが処理され、キーワードテストに保存されます。

6. レコーディングしたテストの分析

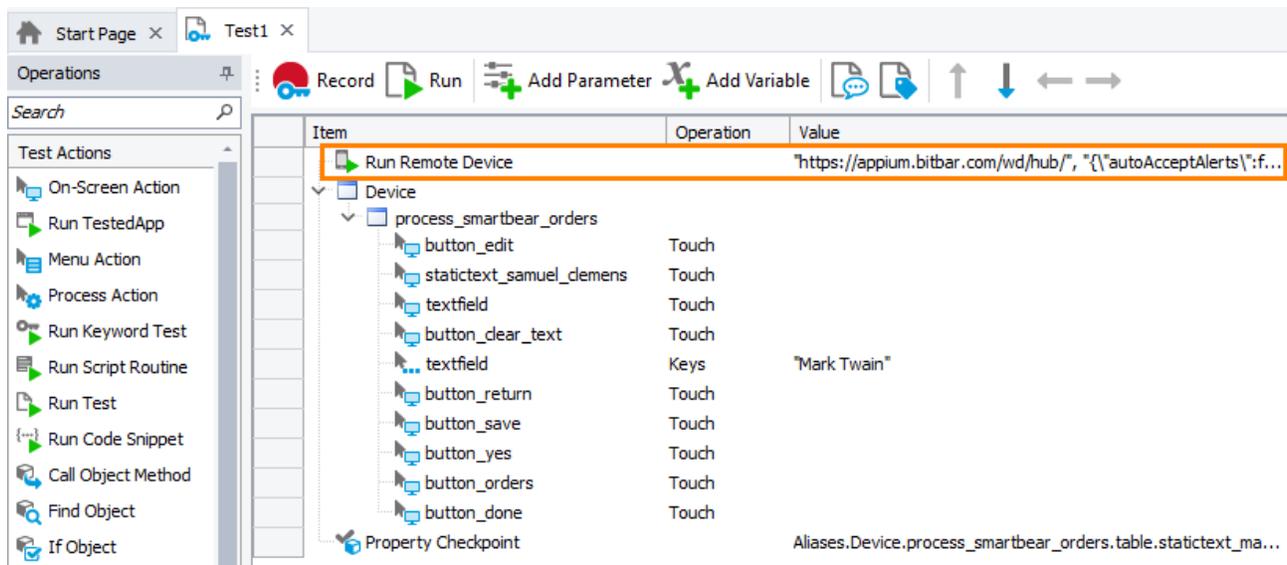
レコーディング終了後、TestComplete はレコーディングしたキーワードを Keyword Test エディターで開きます。



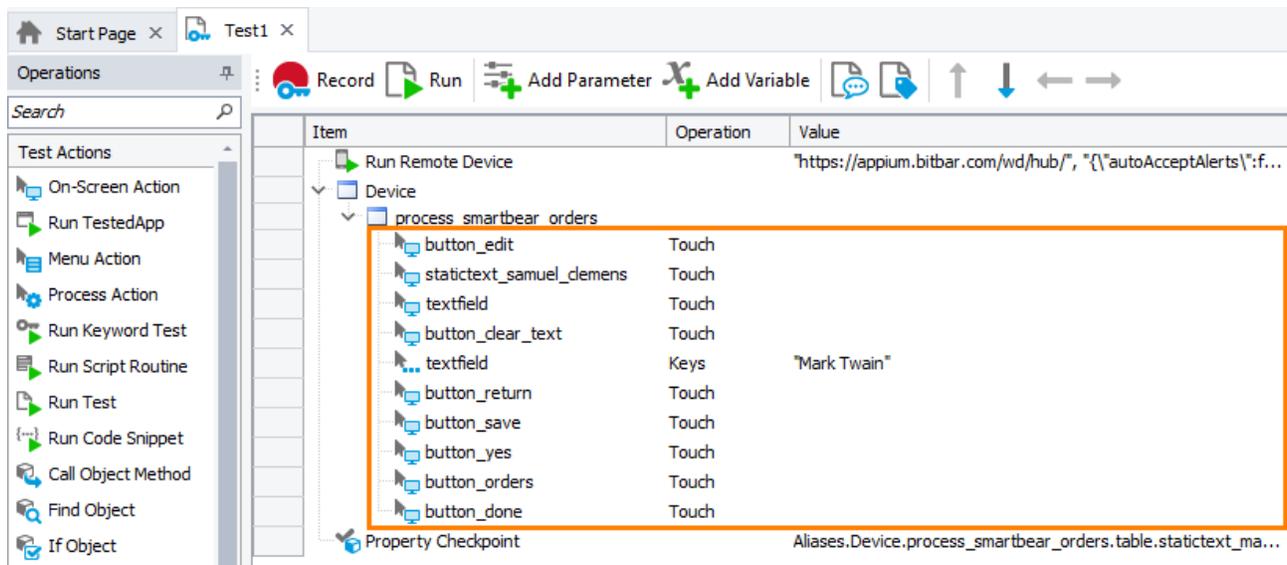
レコーディングしたテストは上記の画像に示すテストのようになります。実行環境によってはこれとは異なることがあります。たとえば、不要なタッチ操作が含まれていることがあります。

このテストには、レコーディング中に Orders アプリケーション上で実行した動作に対応するコマンドが含まれています。このテストコマンドを **operation** と呼びます。

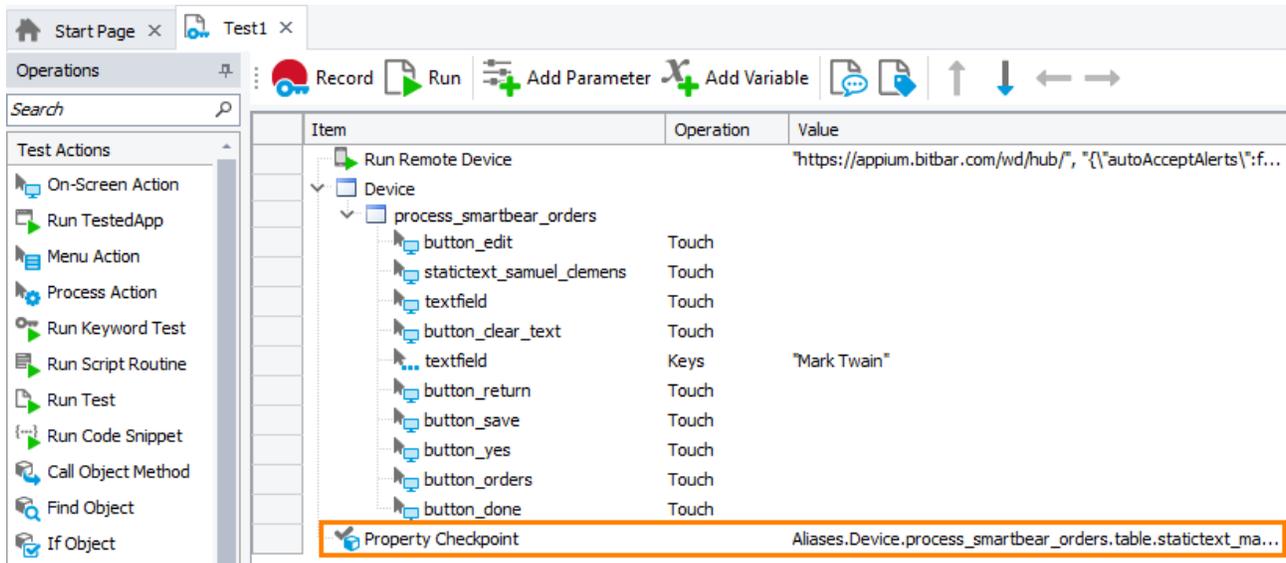
このテストの最初の operation は、**Run Remote Device** です。この operation は、BitBar モバイル デバイス クラウドのモバイル デバイスに接続して、Orders アプリケーションを指定したデバイスに展開し、アプリケーションのテストセッションを開きます。以降のテスト操作はすべてこのデバイスに関するものです。



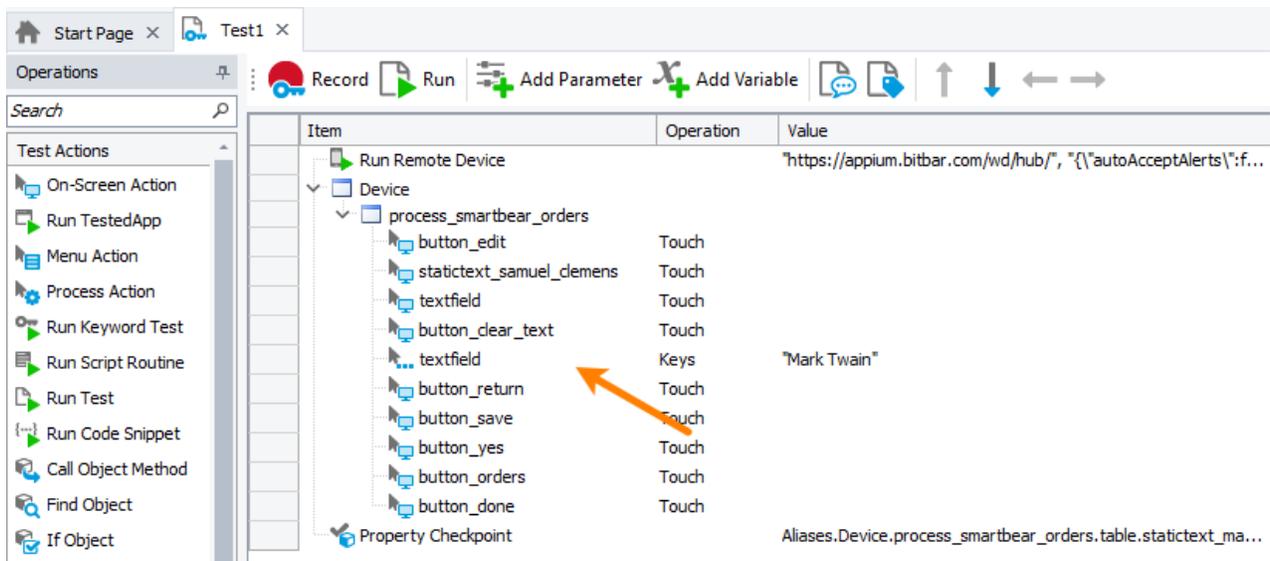
その後、アプリケーションでのユーザー アクションをシミュレートする operation が続きます。これらの operation は、[Edit] ボタンをクリックし、注文リストから項目を選択し、テキストフィールドの値を変更し、変更を保存して [Done] ボタンをクリックします。



最後に、テストレコーディング中に追加した比較動作です。

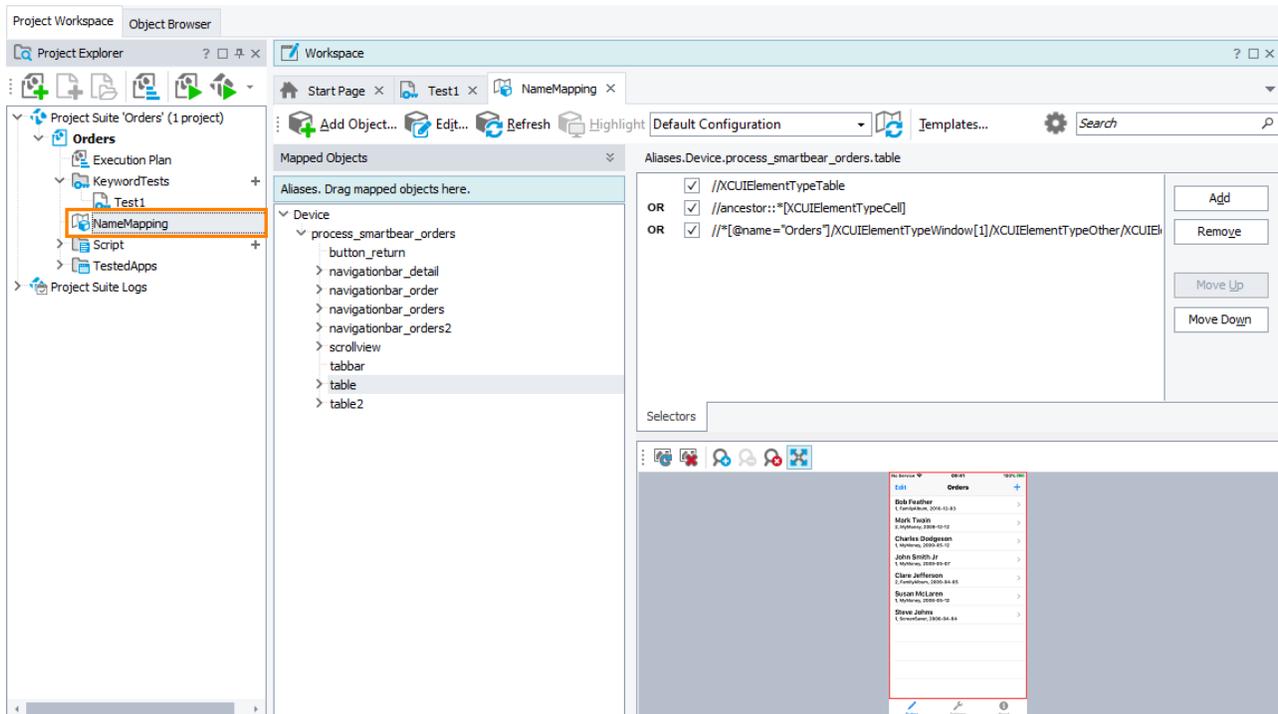


レコーディングしたテストのアプリケーションとオブジェクトの名前に注意する必要があります。



デフォルトでは、テストをレコーディングしている間、TestComplete は自動的にすべての項目を **Name Mapping** リポジトリに追加します。項目ごとに、TestComplete はテストの実行中にアプリケーション内のオブジェクトを見つけるために使用する検索式と、**alias** を格納します。**alias** はテスト内のオブジェクトを見つけるために使用する短い名前です。

Project Explorer の **Name Mapping** 項目をダブルクリックして、Name Mapping リポジトリを開き、TestComplete がテストのレコーディング中にオブジェクト用に作成した認識パラメーターと **alias** を表示できます。



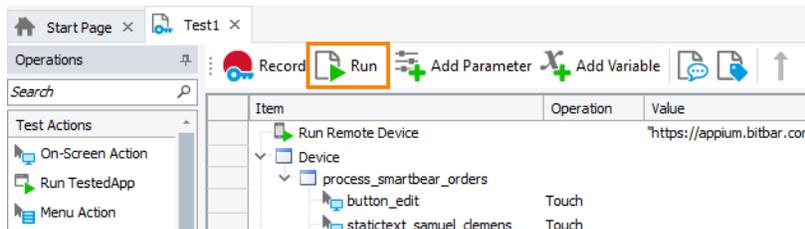
Name Mapping と **alias** の使用は、テストをより分かりやすく、安定したものにします。

7. レコーディングしたテストの実行

テストを実行してその動作を検証してみましょう。

テストの実行

テストを実行するには、テスト エディターのツールバーにある  **[Run Test]** をクリックします。



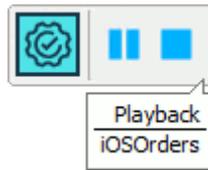
TestComplete は、BitBar デバイス クラウドの指定したデバイスに接続して、デバイスで Orders アプリケーションを起動し、レコーディングしたアクション (Samuel Clemens の注文を開き、顧客名を Mark Twain に変更する) を再生実行します。

テスト結果

テストの終了後、テスト結果が表示され、テストが成功したか、失敗したかを確認できます。テスト結果については、このチュートリアル次のステップで詳しく説明します。

テスト実行に関する注意

- **重要:** テスト実行中は、テスト動作の妨げとならないように、デバイスの画面に触れないでください。
- テスト実行中、画面の右上隅にある **TestComplete** インジケータは、実行されたテスト操作に関する情報を表示します。



このインジケータは、シミュレートしたテスト動作に関する通知をするためのメッセージを表示します。

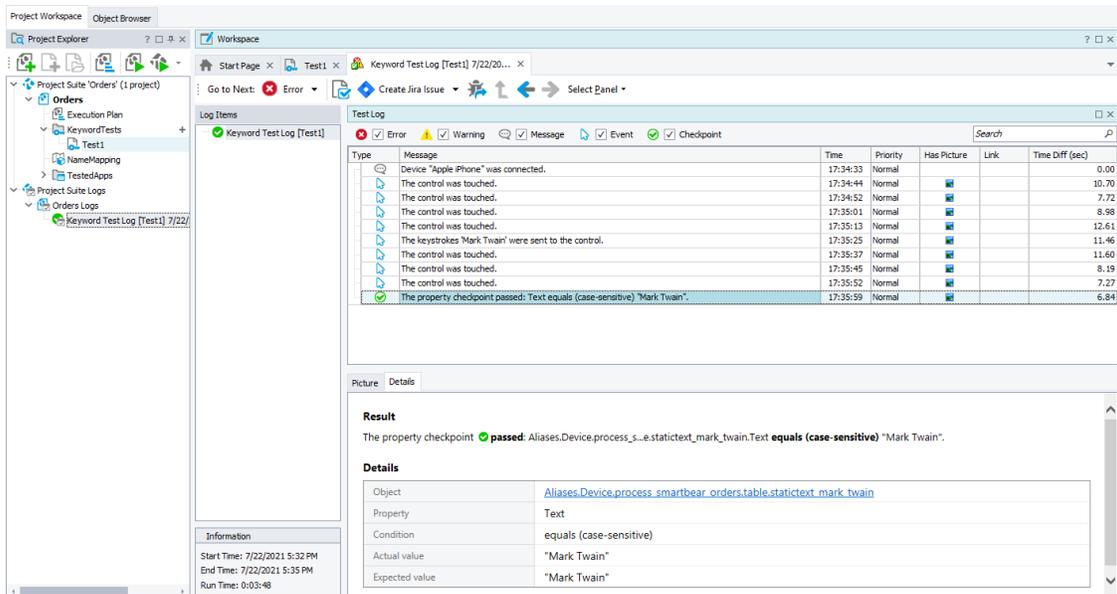
- [Test Engine] ツールバーまたはインジケータの **■ [Stop]** をクリックするか、または **TestComplete** メインメニューから **[Test] > [Stop]** を選択することで、テストの実行をいつでも停止することができます。

インジケータの **■ [Pause]** をクリックすることでテスト実行を一時停止することができます。一時停止の間は、必要な任意の動作を実行することが可能です。たとえば、テスト ログを表示したり、**TestComplete** の **[Watch List]** や **[Locals]** パネルあるいは **[Evaluate]** ダイアログを使用してテストの変数やオブジェクトを確認することが可能です (**TestComplete** ヘルプの「**Debugging Tests**」を参照)。

詳細に関する情報は、**TestComplete** ヘルプの「**Running Tests**」を参照してください。

8. テスト結果の分析

テスト終了後、すべてのテスト操作の結果を含むテスト ログが表示されます。この例では、ログは次のようになります。

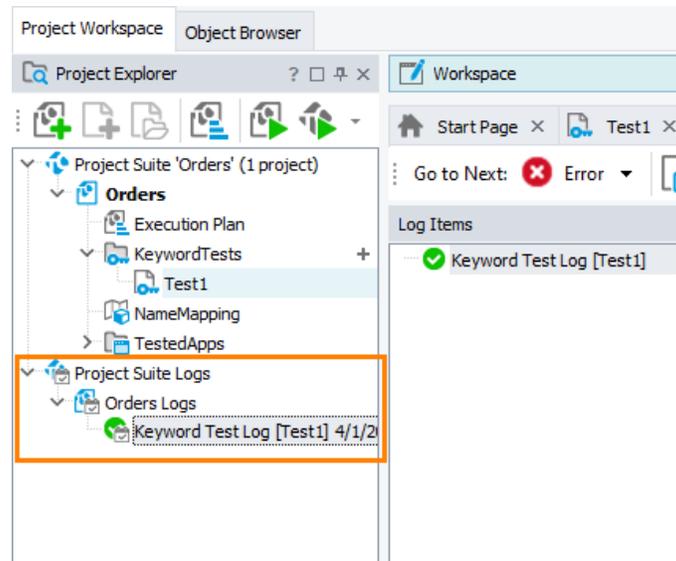


ログには、アクション、イベント、チェックポイントなどに関するさまざまなメッセージが含まれています。メッセージリストの上のツールバーを使用してメッセージをフィルタリングできます。ログメッセージをダブルクリックすると、そのメッセージを出力したテスト操作が表示されます。これは、エラーの原因となった操作を把握するのに便利です。

[Picture] パネルには、選択したテスト コマンドが実行されると、アプリケーションの状態を示す画像が含まれます。これらの画像は、テスト実行中のアプリケーションの動作を理解し、エラーを迅速に見つけるのに役立ちます。

[Details] パネルには、実行された操作の詳細が表示されます。たとえば、プロパティ チェックポイントの **[Details]** には、チェックタイプ (equals、contains、matches a regular expression など) およびその他の詳細が含まれます。

すべてのログが **Project Explorer** の **[Project Suite Logs] > [ProjectName Logs]** にあるため、以前のログを見ることもできます。



エラーの解決

テストが失敗することがあります。これにはいくつかの可能性がありま。たとえば、開発者がアプリケーションの動作を変更したり、アプリケーション コントロールの認識属性が変わりテスト エンジンが必要なオブジェクトを見つけられなくなったりする可能性があります。

初めてのユーザーが直面する最も典型的なものは、テスト作成中と再生実行中でアプリケーションの状態が異なることです。この問題を避けるには、テストを作成したときと同じ初期状態でテスト実行しているか確認してください。

例:

- テストするアプリケーションがテストを記録する前に実行されていた場合は、テストを実行する前に実行している必要があります。
- テストアクションがアプリケーションの特定の画面で実行された場合は、テストを実行するときにもその画面を開く必要があります。
- アプリケーション内のデータを編集して保存した場合は、変更を元に戻す必要があります。

特定のアプリケーションの応答を待つ必要があるため、テスト エンジンがアプリケーション オブジェクトを検出できないことがあります。この問題を解決するには、**Delay** コマンドを挿入し、テキストの実行を一時停止する時間を指定します。

- キーワードテストでは、指定した *Delay Time* (遅延時間) パラメーターで **Delay** オペレーションを使用します。
- スクリプトでは、ループ本体に `aqUtils.Delay` メソッドを使用します。

<C:\work\reference\program-objects\aqutils\delay.htm>

エラーの原因を探し、典型的な問題を解決するための情報は、TestComplete ヘルプの「Handling Playback Errors」を参照してください。

次のステップ

スタートガイドチュートリアルはこれで終了です。このスタートガイドが TestComplete の理解に役立つことを願っています。他のアプリケーションタイプのテストを作成する方法を習得したり、高度な機能を習得し独自のテストを作成してみてください。TestComplete とその機能についての詳細情報は、TestComplete ヘルプを参照してください。いくつかのヘルプトピックを次に示します。

共通トピック

➤ TestComplete によるレコーディング (Recording in TestComplete)

このセクションには、TestComplete によるテストのレコーディングに関する情報があります。

➤ チェックポイント (Checkpoints)

このセクションは、テスト エンジンによって提供されるさまざまなチェックポイントの種類と、テストのレコーディング中およびデザイン時にチェックポイントを作成する方法を説明します。

➤ テストの実行 (Running Tests)

このセクションには、テスト実行の方法、バッチ実行 (テストのグループを実行) の設定方法、テスト実行の予定の立て方などの情報があります。

➤ テストログ (Test Log)

TestComplete によるテスト結果の記録方法とテストログ パネルについて説明します。このセクションでは、メッセージ、画像、ファイルをログに投稿する方法についても説明します。

➤ 再生実行エラーの対応 (Handling Playback Errors)

テスト実行中に発生したエラーの対応方法を説明します。

➤ TestComplete の使用 (Working With TestComplete)

TestComplete プロジェクトをチーム メンバーと共有する方法、TestComplete テストをビルド、開発、品質保証プロセスに統合する方法がわかります。

デスクトップ アプリケーションに関するトピック

➤ オブジェクトの命名方法 (Naming Objects)

このセクションには、TestComplete によるプロセス、ウィンドウ、コントロールの名前の付け方に関する情報があります。

➤ ユーザー アクションのシミュレーション (Simulating User Actions)

このセクションは、TestComplete を用いたマウス クリック、キーストローク、メニュー項目の選択のシミュレートについて説明します。

- アプリケーションのオブジェクトとコントロールの使用 (Working With Applications' Object and Controls)

このセクションは、テスト オブジェクトに対する特定のアクションの実行方法と、そこからデータを取り込む方法を説明します。

- 詳細なタスク (Advanced Tasks)

テストの拡張に役立つさまざまな TestComplete 機能に関する情報が得られます (イベントの処理方法、ActiveX オブジェクト、ファイル、データベースの使用方法など)。

Web アプリケーションに関するトピック

- デフォルトの Web テスト (Default Web Testing)

TestComplete によるプロパティを利用して Web オブジェクトを識別する Web テストの作成方法について説明します。

- TestComplete のクロスブラウザー テスト (Cross-Browser Testing in TestComplete)

CSS セレクターの XPath 式を利用して Web オブジェクトを識別する Web テストの作成方法と、デバイスクラウドでのテストの実行方法を説明します。

- クロスプラットフォーム Web テストの並列実行 (Running Cross-Platform Web Tests in Parallel)

デバイスクラウドでクロスプラットフォーム Web テストを並列に実行する方法を説明します。

- エミュレーターを使用したモバイル Web アプリケーションのテスト (Testing Mobile Web Applications Using Emulator - Tutorial)

サンプルのクロスブラウザー モバイル Web テストを作成するためのステップ バイ ステップのチュートリアル

モバイル アプリケーションに関するトピック

- モバイル テストについて (About Mobile Tests)

TestComplete を使用したモバイル テストの作成方法を説明します。

- モバイル デバイスへ接続してテスト セッションを開く (Connecting to Mobile Devices and Opening Testing Sessions)

TestComplete からモバイル デバイスへの接続方法を示します。

- モバイル テストの作成と実行 (Creating and Running Mobile Tests)

モバイル デバイスで自動テストを作成して実行するステップ バイ ステップの手順を紹介します。

テクニカル サポートとリソース

TestComplete に関する質問、問題、その他のお問い合わせについては、下記のフォームから **サポートチーム**までご連絡ください。

➤ <https://support.smartbear.com/message/?prod=TestComplete>

サポートチームから電子メールで回答を送付します。また、以後の連絡は電子メールで行われます。

さらに、**フォーラム**で回答を探したり、意見を交換することもできます。また、**よくある質問**で質問に対する回答を見つけたり、**ビデオ チュートリアル**、**Web セミナー**、**ブログ**を見たり、**テクニカル ペーパー**を参照したり、SmartBear 社による TestComplete **トレーニング セミナー**に参加することも可能です。

開発元から提供されるサポート情報に関しては、次の Web サイトを参照してください。

<https://support.smartbear.com/>

エクセルソフト株式会社へのお問合せ

TestComplete に関する質問、問題、その他のお問い合わせについては、エクセルソフト株式会社の下記のお問合せページからお気軽にご連絡ください。

https://www.xlssoft.com/jp/services/xlsoft_form.html

索引

A	す
Android アプリケーションのテスト 67	スクリプト 5
I	ち
iOS アプリケーションのテスト 90	チェックポイント
N	作成 23
Naming オブジェクト 9	チェックポイント 13
O	チェックポイント (Checkpoints) 23
Open アプリケーション 12	て
S	テクニカル サポートとリソース 115
Stores 13	デスクトップ アプリケーションのテスト 14
U	テスト
UI テスト 5	レコーディング 20, 45
W	テスト オブジェクト モデル 9
Web アプリケーションのテスト 38	テスト結果
お	モバイル (iOS) 111
オブジェクト naming 9	テスト
オブジェクト モデル 9	結果の分析 34, 87, 111
オブジェクト名のマッピング 31, 57	テスト
き	作成 20, 45, 71, 73, 99
キーワード テスト 5	テスト
機能テスト 5	実行 32, 58, 86, 109
さ	テスト
サポートとリソース 115	自動テストについて 5
し	テスト
自動テスト 5	テスト タイプ 5
	テスト
	テスト タイプ 5
	テスト プロジェクト 6
	テスト プロジェクト
	Web プロジェクトの作成 38
	テスト プロジェクト
	デスクトップ プロジェクトの作成 17
	テスト プロジェクト

モバイル プロジェクトの作成 (Android)	69
モバイル プロジェクトの作成 (iOS)	90, 93
テスト	
レコーディングしたテストの分析	28, 54, 83, 106
テスト ログ	
Web	60
テスト ログ	
デスクトップ	34
テスト ログ	
ソースにジャンプ	36, 61
テスト結果	
Web	60
テスト結果	
エラーの解決	36, 62
テスト結果	
ソースにジャンプ	36, 61
テスト結果	
デスクトップ	34
テスト結果の分析	
モバイル	87
テスト結果	
モバイル (Android)	87
テスト	
結果の分析	60
テスト結果の分析	
Web	60
テスト結果の分析	
デスクトップ	34
テスト結果の分析	
モバイル	111
テストするアプリケーション	
デスクトップ	19
テストするアプリケーション	
モバイル	69, 93
テストの作成	
Web	45
テストの作成	
デスクトップ	20

テストの作成	
モバイル (Android)	71, 73
モバイル (iOS)	99
テストの実行	
Web	58
テストの実行	
一時停止	33, 59, 86
テストの実行	
初期状態	32, 58
テストの実行	
停止	33, 59, 86
テストの実行	
デスクトップ	32
テストの実行	
バッチ実行	33
テストの実行	
別のブラウザ	63
テストの実行	
モバイル	86, 109
テストのレコーディング	
Web	45
デスクトップ	20
テストの実行	
一時停止	110
停止	110

ね

ネーム マッピング	57
ネーム マッピング	31

は

パネル	8
-----------	---

ふ

ブラックボックス アプリケーション	11
プロジェクト	6
プロジェクト	
Web	38
プロジェクト スイート	6

プロジェクト	
デスクトップ	17
プロジェクト	
モバイル (Android)	67
プロジェクト	
モバイル (iOS)	90
プロジェクト項目	6

ほ

ホワイトボックス アプリケーション	12
-------------------------	----

も

モバイル アプリケーションのテスト	67, 90
-------------------------	--------

ゆ

ユーザー アクションのシミュレーション	30
ユーザー インターフェイス概要	8

れ

レコーディングしたテストの分析	
Android	83
レコーディングしたテストの分析	
Web	54
レコーディングしたテストの分析	
デスクトップ	28
レコーディングしたテストの分析	
モバイル	106

ろ

ログ	
Web	60
ログ	
ソースにジャンプ	36, 61
ログ	
デスクトップ	34