

# Getting Started with TestComplete 15 $f_{2}$ , Web $f_{3}$ , $f_{3}$

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. Worrd 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

自動う	テストおよび TESTCOMPLETE について	5
白重	動テスト	
テス	マト タイプ	5
Tes	stComplete プロジェクトとプロジェクト項目	6
Tes	stComplete ユーザー インターフェイス	8
Tes	stComplete テスト オブジェクト モデル	9
チュ	ェックポイントと Stores	
デスク	フトップ アプリケーションのテスト	
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	
4.	テストするアプリケーションの定義	
5.	テストのレコーディング	
6.	レコーディングしたテストの分析	
7.	レコーディングしたテストの実行	
8.	テスト結果の分析	
WFB	マプリケーションのテスト - チュートリアル	38
1	Web ブラウザーの進備	39
2	テストの計画	40
<u>-</u> . 3.	テストのレコーディングについて	
4.	テストのレコーディング開始	
5.	テストする Web ページの定義	
6.	テストのレコーディング	
7.	レコーディングしたテストの分析	
8.	レコーディングしたテストの実行	
9.	テスト結果の分析	
10.	複数のブラウザーでのテストの実行	63
	ROID 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	
2.	テストのレコーディング開始	
3.	テストするアプリケーションの定義	
4.	デバイスへ接続してテスト セッションを開く	
5.	テストのレコーディング	
6.	レコーディングしたテストの分析	
7.	テストの実行	
8.	テスト結果の分析	

IOS 7	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	
2.	テストのレコーディング開始	
3.	テストするアプリケーションの定義	
4.	デバイスへ接続してテストセッションを開く	
5.	テストのレコーディング	
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	
8.	テスト結果の分析	111
次のス	ペテップ	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分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。

TestComplete - C:\Users\Public\Documents\TestComplete 15 Samples\Desktop\Orders	\C#\TCProjects\JavaScrip	t\Orders_C#_JavaScript.pjs							□ ×
<u>File Edit View Test Debug Tools Help</u>									
New • 🕞 📴 • 🐑 🗘 • : 📦 🚳 📲 • : 🍔 • 🕼	🕶 📰 Stop 🐞 🔋	Show Mobile Screen 👻							$\bigcirc$
Project Workspace Object Browser		Object Browser panel							
□ \$\$ \$\$ \$\$ C \$\$	Selected object:								
Objects	Aliases.Orders.Mai	nForm							· ۲
	Properties Methods	Search	Q	000	¥.				
Sys									
Process("Orders")	Basic view View m	ore members (Advanced view)							
	✓ Standard Eastland	hu a							
AppDomain("Orders.exe")	Height	408							
Window ("IME", "Default IME", 2)	MappedName	Aliases, Orders, MainForm	<u>ن</u>						
Window("IME", "Default IME", 1)	Name	WinFormsObject("MainForm")	-						
Window ("MSCTFIME UI", "MSCTFIME UI", 1)	Parent	(Object)							
Window ("GDI+ Hook Window Class", "GDI+ Window (Orders.exe)", 1)	Visible	true		😨 Orders - C:\User	rs\Public\Docume	nts\TestComp	lete 15 Samples\Des	ktop\Orde –	
Window (".NET-BroadcastEventWindow.4.0.0.0.141b42a.0", ".NET-BroadcastEv	Width	608		File Orders Repo	ort View	_			
Window("tooltips_dass32", "", 2)	<ul> <li>WndCaption</li> </ul>	Orders - C:\Users\Public\Documents\TestComplete 15 Samples\Desktop\Orders\C#\TCProjects\MyTable.tbl	5	Curtomer Name	Product	at in a	Date	Oreat	Ch-
WinFormsObject("ToolBar")	✓ .NET			John Smith Jr	MyMoney	1	7/5/2009	12, Orange Blvd	Grovetown, CA
WinFormsObject("StatusBar")	CirFullClassName	Orders.MainForm		Clare Jefferson Susan Mol area	FamilyAbum Moderan	2	5/4/2009	23, Owk Street 7, Flower Street	Greentown, CA
WinFormsObject("OrdersView")	WinFormsControlNar	me MainForm		Charles Dodgeson	MyMoney	i	12/5/2009	45, Stone st.	Bringtone, TX
5				Steve Johns Samuel Clemens	ScreenSaver MyMoney	2	4/4/2008 12/12/2009	17, Park Avenue 3, Garden st.	Salmon Island Hilsberry, UT
9 9				Bob Feather	FamilyAlbum	1	3/12/2010	14, North av.	Miltown, WI
0				Mark Smtn	PamityAbum		2/2/2010	s, maple valley	whitestone, bitts
ž –									
5									
S.									
2 C									
F									
Yst									
2	Description								
5									
8									
4									
Bookmarks Search/Replace Results To Do									
CAPS NUM SCR									

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るので、 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コンセプトに対応するため、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):

Project Workspace	Objec	t Browser			
i 🔂 👯 The Objects (m	execı inus	utable n extensio	ame on)		*
Sys	ss("Ord	ers")		0	
Proces	indow(" indow(" indow("	epad", 2) 'IME", <sup>(*</sup> Def 'MSCTF ME 'Notepa <mark>d</mark> ",	ault IME" UI", "MS "Untitled	Process na , 1) CTFIME UI", 1 - Notepad", 1	ame
>- <u>()</u> Proces	pro	Index of	the stance	32", "", 1)	

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。<u>C:\testing-with\object-identification\name-mapping\about.htm</u>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。

Pr	oject Workspace	Object Browser		
: [	💽 🙆 🐞	🗞 📿 🍣		
	Objects			
	m 📘 Mobile			
	🗸 🍂 Sys			
	>  Proces	ss("Orders") Window:	ext or title	
	Y 🧾 Proces	ss("note	text of the	
	- 🗖 W	indow("IME", "Default	IME", 1)	
	W	indow("MSCTFIME UI"	"MSCTFIME UI", 1)	
Window name	🔶 🍾 🗖 w	indow("Notepad", "Un	titled - Notepad", 1)	
	> 🐂 Proces	ss("explore <mark>r</mark> ")	$\uparrow$	
	Window	class name	Index	

### 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。

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。

💀 Orders - C:\Users\Public\Documents\TestComplete 14 Samples\Desktop\Orde 🛛 🗙										
File Orders Report View										
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.	Hillsberry, UT					
Bob Feather	FamilyAlbum	1	3/12/2010	14, North av.	Milltown, 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

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。

注意: ファイル マネージャーによっては、Public Documents フォルダーを 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。

- 他のコンパイラー (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 <u>https://support.smartbear.com/downloads/testcomplete</u> 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。

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ー)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、<sup>∨</sup> 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。

🚳 TestCor	mplete Rec	ording					¢	^	×
Record	Pause	Stop	Add Check	Run App +	Add Comment	Test Type ∓		Scrip! Ext	t T
Parent: Object:									

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、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ー)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。
- 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。
- 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。

🗾 Workspace	
A Start Page ×	
MARTBEAR TestComplete <sup>™</sup>	DOCUMENTATION
RECORD TE ST	NEW PROJECT
Getting Started Create your first test with TestComplete	Test M for An

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 X							
Select project name, location and type of test to record.							
Project name: *							
Orders							
Project location: *							
C: \Users \tester \Documents \TestComplete 14 Projects \							
Keyward							
O Python							
Participation of the sector							
Next > C	ancel						

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。

web	Application file:*
Desktop	C: \Users\Public\Documents\TestComplete 14 Samples\Desktop\Orders\C#\bin\Release\Orders.exe
Mobile	✓ Restart application (recommended) ● Command-line parameters:
	Working folder:

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。

- **注意:** ファイル マネージャーによっては、Public Documents フォルダーを 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. 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。



 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。

💀 Orders - Untitled					_		х
File Orders Report	t View						
🗅 🗲 🖬 🖻 😭	X <u>a</u> <u>s</u>						
Customer Name	Product	Quantity	Date	Street		City	
<							>
							.:

- 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 フォルダーにあります。ファイルマネージャー によっては、Public Documents フォルダーを 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ーが表示された場 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。

💀 Orders - C:\Users\Public\Documents\TestComplete 14 Samples\Desktop\Orde — 🛛 🛛 🗡 File Orders Report View									
🗅 🚔 🖥 🛍	X <u>°</u> 10 10	i							
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.	Hillsberry, UT				
Bob Feather	FamilyAlbum	1	3/12/2010	14, North av.	Milltown, WI				
Mark Smith	FamilyAlbum	1	2/2/2010	9, Maple Valley	Whitestone, Britis				
1			_						
×					2				

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。

Order				×
Product: MyMor	ney			~
Quantity: 2	🗧 , price per unit	\$100	, discount	0%
			Total:	200
Date:	12/12/2009			
Customer Name:	Samuel Clemens			
Street	3, Garden st.	City:	Hillsberry,	UT
State:	US	Zip:		
Card:	🔿 Visa			
	MasterCard			
	◯ American Express			
Card No	123456789012			
Expiration Date:	02/03/2012	-		
			ОК	Cancel

- 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 <sup>●</sup> 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って、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。

Image: Orders - C:\Users'       File     Orders       Report       Image: Orders       <	\Public\Documents t View ' X ≞≞ ∷- ☷	\TestCom	plete 14 Samples\De	sktop\Orde	-		×
Customer Name John Smith Jr Clare Jefferson Susan McLaren Charles Dodgeson Steve Johns Mark Twain Bob Feather Mark Smith	Product MyMoney FamilyAlbum MyMoney MyMoney ScreenSaver MyMoney FamilyAlbum FamilyAlbum	Quantity 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Date 7/5/09 Quick Checkpoints witemCount = 8 witem wChecked Enabled = True Check Other Proper Disable Quick Chec	Street 12, Orange Blvd s		City Grovetov Greentov Earlcastle Bringtone Salmon I: Hillsberry Milltown, Whitesto	vn, CA vn, CA e, TX sland , UT WI ne, Britiź
<							>

 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。

Available Properties (0 se	lected)		Search	م		
Property	<u>=</u>	Condition	Value	Category	Select	-
wItem	PAR	Equals (case-sei	-	Extended property		1
		Item		Parameter		1
		SubItem	0	Parameter		
wItemBounds	PAR	Equals	*	Extended property		
wItemCount		Equals	- 8	Extended property		
wLParam	PAR	Equals	*	Extended property		
wListSeparator		Equals (case-sei	<b>*</b>	Extended property		
wListViewStyleEx	OBJ	Equals	• (Object)	Extended property		
wSelectBounds	PAR	Equals	<b>*</b>	Extended property		
wSelected	PAR	Equals	<b>*</b>	Extended property		
wSelectedItems		Equals (case-sei	- Samuel Clemens	Extended property		
wStateImage	PAR	Equals	<b>*</b>	Extended property		
wTextBounds	PAR	Equals	<b>*</b>	Extended property		
width_2		Equals	- 592	Private field		
x		Equals	<del>~</del> 0	Private field		-

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。

,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	pic prop				
Available Properties (1 select	ed)	1	Search	م	
Property	<u>=</u>	Condition	Value	Category	Select *
wItem	PAR	Equals (case-sei	<ul> <li>Samuel Cleme</li> </ul>	ns Extended property	✓
		Item	5	Parameter	
		SubItem	0	Parameter	
wItemBounds	PAR	Equals	*	Extended property	
wItemCount		Equals	* 8	Extended property	
wLParam	PAR	Equals	*	Extended property	
wListSeparator		Equals (case-sei	-	Extended property	
wListViewStyleEx	OBJ	Equals	- (Object)	Extended property	
wSelectBounds	PAR	Equals	-	Extended property	
wSelected	PAR	Equals	-	Extended property	
wSelectedItems		Equals (case-sei	- Samuel Cleme	ens Extended property	
wStateImage	PAR	Equals	-	Extended property	
wTextBounds	PAR	Equals	Ŧ	Extended property	
width_2		Equals	<del>*</del> 592	Private field	
x		Equals	<del>~</del> 0	Private field	
inne Orders MainForm Or	1 r	Equals	<del>~</del> 0	Private field	Paris

- [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。[しいえ] 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 テストを編集用に開き、Keyword Test エディターにテストの内容を表示します。

🗾 Workspace					? □ ×
🛧 Start Page × 🔂 To	est1 ×				•
Operations -		🕨 Record 📑 Run 📑 Add Par	ameter 🄏 Add Varia	ble 🗋 🚺 🕇 🗍 🔶	$\rightarrow$
Search P			•		
Test Actions	1	Item	Operation	Value	Description
hp On-Screen Action		Run TestedApp     Orders	Orders		Runs the "Orders" teste
Run TestedApp		. <u></u> MainForm			
Menu Action		MainMenu	Click	"File  Open"	Moves the mouse curso
ho Process Action		dgOpen ✓ mainForm	OpenFile	Variables.Var1, "Table (*.tbl)"	Opens the KeywordTest
Run Keyword Test		CrdersView	ClickItem	"Samuel Clemens", 0	Clicks the 0 subitem of t
Run Script Routine		ToolBar	ClickItem	5, false	Clicks the 5 item of the '
🗅 Run Test		V .net OrderForm			
{···} Bun Code Spinnet		Customer	Click		Clicks the 'Customer' ob
Run Code Snipper		👆 Customer	ClickR		Clicks the 'Customer' ob
Call Object Method		PopupMenu	Click	"Select All"	Moves the mouse curso
K Find Object		http://www.customer	SetText	"Mark Twain"	Enters the text 'Mark T
🍋 If Object	🔼	ButtonOK	ClickButton		Clicks the 'ButtonOK' bu
Image Based Action		Property Checkpoint		Aliases.Orders.MainForm.OrdersVi	Checks whether the 'wI
OCR Action		MainForm	Close		Closes the 'MainForm' w
Logging		🗸 🗖 dlgConfirmation			
Web -		btnNo	ClickButton		Clicks the 'btnNo' button.
🔁 Test Steps 🕄 Variab	les [	Parameters			
🔁 Test Visualizer					-

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。

Run Code Snippet	) Property Checkpoint Orders	Aliases.Orders.MainForm.Ord	er Checks whether the 'wItem(5, 0
Call Object Method	MainForm Close		Closes the 'MainForm' window.
F If Object	btnNo ClickButton		Clicks the 'btnNo' button.
Test Steps 🔓 Variables 📑 Param	eters		
🔊 Test Visualizer			-
Update 💿	Update	Update	Update
€ down-Color-Alachonouring factoring billing billing billing               Ξ Σ               Ξ Σ               Ξ Σ               Ξ             Ξ               Ξ             Ξ	Der S Neis Nime Der S Neis Nime Der S Seine No S	ber standard bereiten standar	Sate         x           Backs         the max           Backs         the max
<			>

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			
✓ .net 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
24	🚽 🔤 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			
23	MainMenu	Click	"File Open"	Moves the mouse curs
	👆 🐂 dlgOpen	OpenFile	Variables.Var1, "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実行するア プ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 Ma	in Form Ir1, "Table (*.tbl)"	Opens the KeywordTest
MainForm	IVIC	in tothi	
🛛 🍋 OrdersView	ClickItem	"Samuel Clemens", 0	Clicks the 0 subitem of t
ToolBar	ClickItem	5. false	Clicks the 5 item of the '
.net OrderForm	Edit (	Order Form	
Customer	Click		Clicks the 'Customer' ob
Customer	ClickR		Clicks the 'Customer' ob
PopupMenu	Click	"Select All"	Moves the mouse curso
http://www.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
V 🔲 dlgConfirmation			
ka 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		
2	MainForm	Close	Closes the 'MainForm' w
	dlgConfirmation		
2	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。

<u> </u>	- Them		Operation	Value	Description
Grouping	nodes	Run TestedApp	Orders		Runs the "Orders" teste
		Orders			
		.net MainForm			
	2	MainMenu	Click	"File  Open"	Moves the mouse curso
	2	📲 dlgOpen	OpenFile	Variables.Var1, "Table (*.tbl)"	Opens the KeywordTest
		NordersView	ClickItem	"Samuel Clemens", 0	Clicks the 0 subitem of t
		NoolBar	ClickItem	5, false	Clicks the 5 item of the '
		net OrderForm			
	2	ko Customer	Click		Clicks the 'Customer' ob
	2	ko Customer	ClickR		Clicks the 'Customer' ob
	2	PopupMenu	Click	"Select All"	Moves the mouse curso
	2	http://www.customer	SetText	"Mark Twain"	Enters the text 'Mark T
	2	ButtonOK	ClickButton		Clicks the 'ButtonOK' bu
	×	Property Checkpoint		Aliases.Orders.MainForm.OrdersVi	Checks whether the 'wI
	× 🗆	Orders			
	2	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お気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。

🗾 Workspace							? □ ×
🔺 Start Page 🛛 🐻	Tes	t1 ×					*
Operations	<b>7</b>	: 👩	Record 🕞 Run	📲 Add Par	ameter 🄏 Add Variat	ole 🔂 🚺 🕇 🔶 🔶	$\rightarrow$
Search			Item		Operation	Value	Description
Non-Screen Action			Run TestedAp	p	Orders	•••	Runs the "Orders" teste

このテスト エンジンは 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ーを表示します。

mples	s\Desktop\Orders\C#\TCProjects	\JavaScript\Orders_C	#_JavaScript.pjs	>	<
:	■ ■ ■ 🕸	<u>成我走</u> 日	Show Mobile Screen 👻	Waiting for the diaOpen object	-
				Orders	5
t1 ×					-
	Record 🕞 Run 🗮 Add Par	rameter 🄏 Add Varia	able 🕞 🚺 🕇 🗍	$\leftarrow \rightarrow$	
	Item	Operation	Value	Description	*
	Run TestedApp	Orders		Runs the "Orders" test	

このインジケーターは、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。

🗾 Workspace						? □ ×			
🛧 Start Page × 🔝 Test1 × 🖓 Keyword Test Log [Test1] 3/23/20 ×									
🕴 Go to Next: 😢 Error 🔻 🔀 💠 Create Jira Issue 👻 🏂 👔 🗲 🔶 Select <u>P</u> anel 🗸									
Log Items	TestLog								
🛛 🕑 Keyword Test Log [Test1]	g [Test1] 😮 🗸 Error 🙏 🗸 Warning 😋 🗸 Message 🔓 🗸 Event 🤗 🗸 Checkpoint Search 🔎								
	Type Message	Time	Priority	Has Pi	Link	Time Diff (sec)			
	The application "C:\Users\Public\Documents\TestComplete 14 Samples\Desktop\Orders\C#\	12:25:44	Normal			0.00			
	The menu item 'File Open' was clicked.	12:25:45	Normal			0.73			
	The file 'MyTable.tbl' was selected in the Open dialog.	12:25:46	Normal			1.13			
	The list view item ('Samuel Clemens', 0) was dicked with the left mouse button.	12:25:46	Normal			0.40			
	Toolbar button 5 was clicked.	12:25:46	Normal			0.06			
	The window was clicked with the left mouse button.	12:25:47	Normal			0.37			
	> The window was clicked with the right mouse button.	12:25:47	Normal			0.68			
	The menu item 'Select All' was clicked.	12:25:48	Normal			0.63			
	The text 'Mark Twain' was entered in the text editor.	12:25:48	Normal			0.04			
	> The button was dicked with the left mouse button.	12:25:48	Normal			0.09			
	The property checkpoint passed: wItem(5, 0) equals (case-sensitive) "Mark Twain".	12:25:48	Normal			0.34			
	The window 'Orders - C:\Users\Public\Documents\TestComplete 14 Samples\Desktop\Orders	12:25:51	Normal			2.16			
	The button was clicked with the left mouse button.	12:25:51	Normal			0.17			
	Picture Duta								
						Actual Image			
		😡 🏑 🏑 Toom In Selection							
		Order X							
	Product MyMoney								
	Quantly: 2 🗘 . price per unit 1100 . discont 🕅 Quantly: 2 🖓					. price per unit (\$100 . discount (%)			
	Tetal 200		Tetat 200						
	Date: 12/12/2020 (c)								
	Streat 3.Goden (	et 3.Ga	Unde Jenis						
	State: US Copy State	er US	Cut Copy						
Information	Card: Ovise Parts Delete Card	t OVs	a Pasta Delete						
	Market     All     Arerice	⊛ Ma ⊖ Am	ericz Select Al						
Start Time: 3/23/20 12:25 PM	CandNo 123456776 Right to left Reading order Show Unicode control characters	d No 12345	6785 Right to Show Un	left Reading order licode control chara	acters				
End Time: 3/23/20 12:25 PM	Expiration Date: 22:03/201 Inset Unicode control character >	iration Date: 02/03	201. Insert Un	ricode control chari	acter >				
Run Time: 0:00:07	Open IME Incel		Open IM Reconve	* rsion	ncel				

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。

Workspace					? □ ×		
👬 Start Page × 🔂 Test1 × 🚳 Keyword Test Log [Test1] 3/23/20 ×							
Operations 무	:	Record 📑 Run 溝 Add Paran	neter 🄏 Add Variable	🕞 🚺 🕇 ↓ ← −	Þ.		
Search P		Item	Operation	Value	Description		
Test Actions		Dun TestedAnn	Operation	Value	Description		
hp On-Screen Action		✓ ☐ Orders	Orders		Runs the Orders tested		
Run TestedApp		MainForm					
Menu Action	2	MainMenu	Click	"File  Open"	Moves the mouse cursor		
ho Process Action	2	dlgOpen	OpenFile	Variables.Var1, "Table (*.tbl)"	Opens the KeywordTest		
🕿 Run Keyword Test		OrdersView	ClickItem	"Samuel Clemens",	Clicks the 0 subitem of th		
Run Script Routine	24	ToolBar	ClickItem	5, false,	Clicks the 5 item of the 'T		
Run Test		V .net OrderForm					
{···} Dun Cada Caisaat	2	Customer	Click		Clicks the 'Customer' obj		
Run Code Snippet	2	Customer	ClickR		Clicks the 'Customer' obj		
Call Object Method	2	PopupMenu	Click	"Select All"	Moves the mouse cursor		
🗞 Find Object	2	🛛 🧤 Customer	SetText	"Mark Twain"	Enters the text 'Mark Tw		
🍋 If Object	2	ButtonOK	ClickButton		Clicks the 'ButtonOK' butt		
Image Based Action		Property Checkpoint		Aliases.Orders.MainForm.OrdersVie	Checks whether the 'wIt		
OCR Action		MainForm	Close		Closes the 'MainForm' wi		
Logging		✓ ☐ dlgConfirmation					
Web +	2	btnNo	ClickButton		Clicks the 'btnNo' button.		
Test Steps 🕄 Variables		Parameters					
🔊 Test Visualizer					-		

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。 ほかの種類の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。

Message The application The application The menu it The file 'My' The list view Toolbar but	ation "C: \Users\F item 'File Open								
Image: Constraint of the second se	ation "C: \Users\F item 'File Open				Time	Priority	Has Pic	. Link	Time Diff (sec)
The menu it The file 'My' The list view Toolbar but	item 'File Open	Public\Documents\Test(	Complete 14 San	nples \Desktop \Orders \C#\bin \release \	14:01:22	Normal			0.00
The file 'My' The list view Toolbar but		.' was clicked.			14:01:23	Normal			0.77
The list view	yTable.tbl' was s	elected in the Open dia	log.		14:01:24	Normal			1.13
Toolbar but	ew item ('Samuel	Clemens', 0) was clicke	d with the left m	nouse button.	14:01:24	Normal			0.47
	utton 5 was clicke	:d.			14:01:24	Normal		_	0.05
The window	w was clicked wit	th the left mouse butto	n.		14:01:25	Normal			0.38
The window	w was clicked wit	th the right mouse butt	on.		14:01:25	Normal			0.69
The menu it	item 'Select All' v	ras clicked.			14:01:26	Normal		_	0.63
The text Ma	Mark Twain' was	entered in the text edit	or.		14:01:26	Normal			0.04
Ine Button	onOK <sup>-</sup> object was	replaced with a similar	object.	<u></u>	14:01:41	Normal			14.81
Ine button	n was clicked with	n the left mouse button	la International and a state of the	to a bland Transfer	14:01:41	Normal			0.09
The propert	rty checkpoint pa	assed: witem(5, 0) equ	ais (case-sensit	IVe) Mark Wain .	14:01:41	Normal	-		0.28
The window	w orders - C: ju	sers y ublic y ocuments	(restcomplete 1	-+ samples (pesktop (orders (c # (r CProj	14:01:45	Normal			2,13
ure Details our test was self- intime.	lf-healed auto	matically! The map	ping criteria i	for the above object was replaced	with crite	ria of a :	similar ob	ject four	nd during
ture Details	lf-healed auto Sys.Orders	matically! The map .OrderForm. <u>But</u>	ping criteria l <u>tonOK</u>	for the above object was replaced TestComplete used this obje of the object it was not able	with crite ct instea e to find.	ria of a : d	similar ob	ject four	nd during
ture Details 'our test was self- untime. IameMapping.S . similar object ha	lf-healed auto Sys.Orders has been foun	matically! The map . OrderForm. <u>But</u> Id – see the Picture	ping criteria f	for the above object was replaced TestComplete used this obje of the object it was not able	with crite ct instea e to find	ria of a s	similar ob	ject four	nd during
ture Details 'our test was self- untime. IameMapping . S . similar object ha Property	lf-healed auto Sys.Orders has been foun	matically! The map .OrderForm. <u>But</u> d – see the Picture Mapping Value	ping criteria f tonOK tab. Found	for the above object was replaced TestComplete used this obje of the object it was not able Action	with crite ct instea e to find.	ria of a s	imilar ob	ject four	nd during
ture Details 'our test was self- untime. IameMapping.5 similar object ha Property WinFormsCon	lf-healed auto Sys.Orders has been foun ontrolName	matically! The map .OrderForm. <u>But</u> d – see the Picture Mapping Value OK	ping criteria l tonOK tab. Found ButtonOK	for the above object was replaced TestComplete used this obje of the object it was not able Action Update	with crite ct instea e to find.	ria of a s	imilar ob	ject four	nd during

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/

<sup>8</sup> 

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。



Online shopping is the process consumers go through to purchase products or services over the Internet. You can edit this in the admin site.



**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るとします。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。

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ー)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。

TestComplete Recording							0	^	×
ecord	Pause	Stop	Add Check	Run App +	Add Comment	Test Type ∓		Scrip! Ext	t T
Parent: Object:									

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、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ー)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。

# 4. 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		
select project name, location and type of test to record.		
Project name: *		
Orders		
Project location: *		
C:\Users\tester\Documents\TestComplete 14Projects\		
2 Learn more about these settings		
	Next >	Cancel
	George S	Contect

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。

Reco	rd Test		×
	Select applica	tion type and settings to record.	
	Web Desktop Mobile	Browsers: ● Google Chrome ● Firefox ● Microsoft Edge ● Internet Explorer URL (optional): https://services.smartbear.com/samples/TestComplete 14/smartstore/ ■ Restart the selected browser (recommended) ● ■ Restart the selected browser (recommended) ● ■ Use XPath and CSS selectors for web objects (required for cross-platform web tests) ● Create tests to be run on Windows, macOS, Linux, Android and iOS Configure your browser	]
	Learn more about the second		
		< <u>B</u> ack Record Cance	<u>ا</u> د

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

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。

**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。

#### 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。

Shop X	+			- 0	×
$\leftrightarrow$ $\rightarrow$ C $$ services.smartbear	.com/samples/TestComplete14/smart	store/	☆	<b>◎ * 0</b>	:
USD (\$) ~	со	NTACT US HELP & SERVICES $\sim$	NEWS BLOG FORUMS	8 LOG IN	<b>^</b>
ball		Q	Compare Wish Products List	Cart Shopping	
Books ~ Furniture Mobile	SALE phone Sports ~ Fashion ~	Gaming Watches Gift	t Cards		
Welcome to our	store				
Online shopping is the process cor site.	isumers go through to purchase p	roducts or services over the I	nternet. You can edit this in	n the admin	
Mobile phone	Sports	Furniture	Sungla	isses	Ţ

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。



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。

Shop. Torfabrik official game ball 🗙 🕂					-		×
← → C	mplete14/smartstore/t	orfabrik-official-game-ball		☆		* 6	) :
cer   Official game ball		SHOPPING CART 🜖	WISHLIST		C	OMPAR	E
	Official c Available in	The product Official     added to your cart	<b>al game ball</b> has be	en suc	cessfu	lly	×
	<b>\$59.</b> excl. VAT ;	Official ga Available Color: Blue	ame ball in different colors e, Size: 3				
	Color * Size * () 3 () 4 () 5	- 1 + \$59.90	excl tax			$\bigcirc$	Î
	O Add to I						
	Add to L	SUBTOTAL: \$59.90 excl tax	Go to ca		~	Checko	out

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 <sup>♣</sup> 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。

SHOPPING CART		WISHLIST	COMPARE			
~	<ul> <li>The product Official game ball has been successfully added to your cart</li> </ul>					
Ą	<u>()</u>	Official gam Available in Color: Blue, S Quick Checl	e ball different colors Size: 3 cpoints			
-	1 +	Enabled = Tr Check Other Disable Quic	= Color: Blue, Size: 3 ue Properties k Checkpoints			

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。

🕈 Start Page 🛛 🔂 Tes	st1 ×				<b>•</b>
Operations 무		Record 📑 Run 🗮 Add Parameter 🤈	🖕 Add Variable 🛛 🔂 📑	$\uparrow  \downarrow  \leftarrow \rightarrow$	
Search P			0	Mahar	Description
Test Actions	1	Etem	Operation	bttps://services.smartbear.com/sampl	Launches the specified browser and on
h On-Screen Action		v @ browser	chronic	https://scivices.sinar.actin.com/sampi	Edulates are specified browser and op
Run TestedApp		✓ □ pageShop			
Menu Action		textboxInstasearch	Click		Clicks the 'textboxInstasearch' object.
Process Action		hp textboxInstasearch	SetText	"ball"	Sets the text 'ball' in the 'textboxInsta
Or Due Keywood Test		textnode	Click		Clicks the 'textnode' object.
		pageTorfabrikOfficialGameBall	Wait		Waits until the browser loads the page
Run Script Routine		pageTorfabrikOfficialGameBall	-		
🕒 Run Test		label	Click		Clicks the 'label' object.
Run Code Snippet		radiobutton3	ClickButton		Clicks the 'radiobutton3' radio button.
Call Object Method		buttonAddToCart	ClickButton		Clicks the 'buttonAddToCart' button.
		Property Checkpoint		Aliases.browser.pageTorfabrikOfficialG	Checks whether the 'contentText' prop
		browser			
Version If Object			ClickButton		Clicks the 'link' button.
Image Based Action		BrowserWindow	Close		Closes the 'BrowserWindow' window.
OCR Action					
📔 Test Steps 🔓 Variable	s 📑	Parameters			
Test Visualizer					в

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、ブラウザーを起動し、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、ブラウザー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.
texthoyInetasearch	CatTavt	"hall"	Sate the text 'hall' in the 'texthevIneta

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			
	Y pageShop			
2	textboxInstasearch	Click		Clicks the 'textboxInstasearch' object.
2	textboxInstasearch	SetText	"ball"	Sets the text 'ball' in the 'textboxInsta
2	textnode	Click		Clicks the 'textnode' object.
	🗝 🖳 pageTorfabrikOfficialGameBall	Wait		Waits until the browser loads the page
	nageTorfabrikOfficialGameBall			

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.
2	http://www.instasearch	SetText	"ball"	Sets the text 'ball' in the 'textboxInsta
2	ky textnode	Click		Clicks the 'textnode' object.
	hpppageTorfabrikOfficialGameBall	Wait		Waits until the browser loads the page
	pageTorfabrikOfficialGameBall			

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
	http://www.commonsteinscomm	Wait		Waits until the
	🗸 🗖 pageTorfabrikOfficialGameBall			
2	🗏 📐 label	Click		Clicks the 'labe
2	ka radiobutton3	ClickButton		Clicks the 'radi
2	buttonAddToCart	ClickButton		Clicks the 'but!
	Yroperty 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 ClickButton		Clicks the 'radiobutton3' radio button. Clicks the 'buttonAddToCart' button.
	Property Checkpoint		Aliases.browser.pageTorfabrikOfficialG	Checks whether the 'contentText' prop
	🗸 🌐 browser			
	👻 🔲 pageTorfabrikOfficialGameBall			
24	ko link	ClickButton		Clicks the 'link' button.
2	🐂 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。

	Property Checkpoint     Browser     pageTorfabrikOfficialGameBall	ClickButton	Aliases.browser.pageTorfabrikOfficialG	Clicks the 'buttonAddToCart' button. Clicks the 'buttonAddToCart' button. Checks whether the 'contentText' prop
2	link	ClickButton		Clicks the 'link' button.
2	🔤 🗤 BrowserWindow	Close		Closes the 'BrowserWindow' window.
そし	て、Web ブラウザーを閉じる動	動作です。		
-	Property Checkpoint		Aliases.browser.pageTorfabrikOfficialG	Checks whether the 'contentText' prop

	🗸 🌐 browser		
	pageTorfabrikOfficialGameBall		
25	link	ClickButton	Clicks the 'link' button.
2	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 つのブラウザーのユーザー アクションをレコーディングしました。そのため、ブラウザーのグルー プ ノードがあります。これらには、ブラウザー 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 s
	textboxInstasearch	Click		Clicks the 'text
Grouping r	10des http://www.instasearch	SetText	"ball"	Sets the text '
	textnode	Click		Clicks the 'text
	pageTorfabrikOfficialGameBall	Wait		Waits until the
	💳 🔶 🛁 pageTorfabrikOfficialGameBall			
	🔼 🛛 👆	Click		Clicks the 'labe
	radiobutton3	ClickButton		Clicks the 'radi
	buttonAddToCart	ClickButton		Clicks the 'butt
	Property Checkpoint		Aliases.browser.pageTorfabrikOfficialG	Checks whethe
	💳 🔶 👻 🔲 pageTorfabrikOfficialGameBall			
	🔼 🔤 🍡 link	ClickButton		Clicks the 'link'
	BrowserWindow	Close		Closes the 'Bro

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。



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、ブラウザーを起動し、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。

0.	Record 📭 Run 🚎 Add Parameter	X Add Variable 🕞 [	$\leftarrow \rightarrow \downarrow \uparrow \downarrow \leftarrow \rightarrow$	
	Item	Operation	Value	Description
	💿 Run Browser	Chrome	"https://services.smartbear.com/sampl	Launches the specified browser and op
	✓ ■ browser			
	🗸 🗖 pageShop			
2	👆 textboxInstasearch	Click		Clicks the 'textboxInstasearch' object.
2	http://www.instasearch	SetText	"ball"	Sets the text 'ball' in the 'textboxInsta
2	textnode	Click		Clicks the 'textnode' object.
	👦 pageTorfabrikOfficialGameBall	Wait		Waits until the browser loads the page

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ーを表示します。



このインジケーターは、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ーの ■ [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。

Workspace						2 🗆 🗙
- Honopuce	•					
📥 Start Page 🛛 🔂 Test1 🛛	🔏 Keyword Test Log [Test1] 3/23/20 ×					-
Go to Next: 🙁 Error 👻 📘	y 🔷 Create Jira Issue 🔻 🏂 🚹 🗲 🌧 Select <u>P</u> anel 🔹					
Log Items	TestLog					□ ×
Keyword Test Log [Test1]	😢 🗹 Error 🔺 🗸 Warning 😋 🗸 Message 🖙 🖓 Event 🥑 🗸 Checkpoint			5	Search	م
	Type Message	Time	Priority	Has Pic	Link	Time Diff (sec)
	- 🤤 Google Chrome was launched.	17:07:26	Normal			0.00
	Navigating to the http://services.smartbear.com/samples/TestComplete 14/smartst page.	17:07:30	Normal			3.48
	The window was dicked with the left mouse button.					0.36
	The text 'ball' was entered in the text editor.	17:07:30	Normal			0.26
	The window was dicked with the left mouse button.	17:07:31	Normal			1.18
	The window was clicked with the left mouse button.	17:07:33	Normal			1.88
	The radio button was clicked with the left mouse button.	17:07:34	Normal	<b></b>		0.53
	The button was dicked with the left mouse button.	17:07:36	Normal			1.84
	The property checkpoint passed: contentText equals (case-sensitive) "Color: Blue, Size: 3".	17:07:36	Normal	<b>1</b>		0.79
	The button was clicked with the left mouse button.	17:07:37	Normal	<b></b>		0.84
	The 'Shop. Torfabrik official game ball - Google Chrome' window was closed.	17:07:38	Normal			0.47
	Picture Details					
	Expected Image		Actual	Image		
	🗄 🕝 🖾 🕰 🔈 💊 🔀 Zoom In Selection 🕴 🗔 🖾 🗛 🖇	ି \Lambda ି	🕻 Zoom li	n Selection	-	
	C - C Alternal interestionare which interest     C - C Alternal interestionare which interest     C - C Alternal interestionare which interest     C - C Alternal interestionare which interestionare interestinterestionare interestionare interestionare interestinterestiona	n fan Lander (de Lander en la 1. de geleter - Stater en Stater en la	minaration in the second	anninge i ver 🗮 en 🔘 enne Constat i statut statut i	- NEW ROC (DAME 2000)	
	м <b>О</b> ад Фат Фат м			۹.	0 aa 0 a 6 ar	
	No Northering	Al-Court Enterine				
		MIDA				
	Contracting for the large kinetic and early age     Contracting age (1) and (1) age (1) a	lanadar High lahad Cara Kala <b>rka</b> Tarat politika analifereta matéh di Oftaid para <b>tat</b>			in the administra	
	Autors and and a second an	Nation of Control of States				
	No Approximate App	Lepane and the having had, only particulying more hybrid <b>bios</b>			. 🗧	
Tafarmatian	S of the second s	n different adars Name <b>Bel</b> Indeptment arming bell.			loor keirtai	
Information				0		
Start Time: 3/23/20 5:07 PM	M Mm Geny Who Office	ad here		Yan De	# Cris	
End Time: 3/23/20 5:07 PM	Featured products		Featured	products		
Run Time: 0:00:14		-		•		
Information Start Time: 3/23/20 5:07 PM End Time: 3/23/20 5:07 PM Run Time: 0:00:14	Februrd products	ingenerations and Ward to Akad any parters, ingeneration Wards Back States and a result back State and a result back and a result back State and a result back and a result back and a result back and a result back and a result back and a result back State and a result back and	Featured	products		

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 [2] [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。

🚔 Start Page 🛛 🔂 Test	1 ×				-
Operations +	: 🧑	Record 💦 Run 🗮 Add Parameter	X Add Variable 🔝	$\rightarrow$ 1 1	
Search P	1				
Test Actions	i 🖂	Item	Operation	Value	Description
Con Company Anting		💿 Run Browser	Chrome	"https://services.smartbear.com/samples/TestC	Launches the specified browser and opens the s
M ON-Screen Action		✓ ⊕ browser			
Run TestedApp		pageShop			
henu Action		ko textboxInstasearch	Click		Clicks the 'textboxInstasearch' object.
Process Action		http://www.instasearch	SetText	"ball"	Sets the text 'ball' in the 'textboxInstasearch' te
		textnode	Click		Clicks the 'textnode' object.
Run Keyword Test		hap page Torfabrik Official Game Ball	Wait		Waits until the browser loads the page and is re
Run Script Routine		<ul> <li>pageTorfabrikOfficialGameBall</li> </ul>			
🕒 Run Test		ko label	Click		Clicks the 'label' object.
Run Code Snippet		hadiobutton3	ClickButton		Clicks the 'radiobutton3' radio button.
P. cal object with a		buttonAddToCart	Click		Clicks the 'buttonAddToCart' object.
Call Object Method		Property Checkpoint		Aliases.browser.pageTorfabrikOfficialGameBall	Checks whether the 'contentText' property of t
Note: Find Object		✓ ⊕ browser			
🍋 If Object		<ul> <li>pageTorfabrikOfficialGameBall</li> </ul>			
Image Based Action		ko link	ClickButton		Clicks the 'link' button.
		http://www.actionality.com/act	Close		Closes the 'BrowserWindow' window.
CR ACTION					

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「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。 ほかの種類の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、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。

Go to Next: 😢 Error 👻 [	👌 🔷 Create Jira Issue 👻 🇚 🏌 🐟 🄶 Select <u>P</u> anel 🗸					
Log Items	Test Log					$\Box$ ×
Keyword Test Log [Test1]	😢 🗸 Error 🔺 🗸 Warning 😳 🗸 Message 🍃 🖌 Event 🥪 🗸 Checkpoint			Searc	ħ	م
	Type         Message           Image: Google Chrome was launched.         Google Chrome was launched.           Image: Navigating to the https://services.smartbear.com/samples/TestComplete 14/smarts page.           Image: The 'Shop - Google Chrome' window was maximized.           Image: The object "textboxInstasearch" does not exist.           Image: The test run has stopped because your test project is configured to stop on errors.	Time 10:41:37 10:41:41 10:41:41 10:41:51 10:41:51	Priority Normal Normal Normal Higher Normal	Has	Link	Time D           0.00           3.45           0.19           10.52           0.06
	Picture       Details       Call Stack         Unable to find the following object when executing the Click command:       Image: Click command:         Aliases.browser.pageShop.header.formSearch.textboxInstasearch       Image: Click command:         Time       Image: Click command:	Nga Entertement Marana Santa Karali Ana Mara	- A 2 € 15 A - A 2 € 15 A - A 10.1 A 2 € 15 A - A 10.1 A 2 € 15 A		2 9 20°	·
Information Start Time: 8/30/2021 10:41 AM End Time: 8/30/2021 10:41 AM Run Time: 0:00:18	<ul> <li>On the <u>Picture</u> tab, check if the object existed during the test run.</li> <li><u>Update the search criteria</u> for the object or view them in the <u>NameMapping editor</u>.</li> <li><u>Learn more</u> about possible causes of the error.</li> </ul>	Feeture	ed products	2077 2070	<b></b>	~

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. 複数のブラウザーでのテストの実行

レコーディングしたブラウザー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 ザーで実行することができます (これは、**クロスブラウザー 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ー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ー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。

🚳 Browser Loop	×
Select Loop Mode Specify the browser loop mode.	
Iterate Through All Browsers Iterates through all supported browsers that are available on the computer. Iterate Through Specific Browsers Select some of the supported browsers for iteration.	
<u>Back</u> <u>Next</u> > <u>Finish</u> Cancel	Help

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。

	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.
2	hp textboxInstasearch	SetText	"ball"	Sets the text 'ball' in the 'textboxInstasearch' te
	textnode	Click		Clicks the 'textnode' object.
	http://www.ageTorfabrikOfficialGameBall	Wait		Waits until the browser loads the page and is re
	pageTorfabrikOfficialGameBall			
2	ko label	Click		Clicks the 'label' object.
2		ClickButton		Clicks the 'radiobutton3' radio button.
2	buttonAddToCart	Click		Clicks the 'buttonAddToCart' object.
	Property Checkpoint		Aliases.browser.pageTorfabrikOfficialGameBall	Checks whether the 'contentText' property of t
	✓			
	Y 🔲 pageTorfabrikOfficialGameBall			
2	- ko 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、ほかのブラウザー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、ブラウザーの数だけテスト操作を繰り返します。毎回、異なるブラウザーでテスト アクションが実行されます。

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

Test Log	1					□×
8	Error 🔥 🗸 Warning 🤤 🗸 Message 🔓 🖌 Event 🥥 🗸 Checkpoint			Se	arch	Q
Туре	Message	Time	Priority	Has Picture	Link	Time Diff (sec)
e 😳	Mozilla Firefox was launched.	17:11:41	Normal			0.00
- D	Navigating to the https://services.smartbear.com/samples/TestComplete14/smarts page.	17:11:45	Normal		-	4.09
	The window was clicked with the left mouse button.	17:11:46	Normal	2	-	1.09
1.						
- 🔓	The button was clicked with the left mouse button.	17:11:54	Normal	<b></b>		0.59
🕞	The 'Shop. Torfabrik official game ball - Mozilla Firefox' window was closed.	17:11:55	Normal			0.58
9	Microsoft Internet Explorer was launched.	17:11:56	Normal			0.86
	Navigating to the https://services.smartbear.com/samples/TestComplete14/smarts page.	17:12:06	Normal			10.55
- D	The window was clicked with the left mouse button.	17:12:07	Normal	<b></b>		0.60
	The property checkpoint passed: contentText equals (case-sensitive) "Color: Blue, Size: 3".	17:12:18	Normai			2.13
Ň	The button was clicked with the left mouse button.	17:12:19	Normal			0.85
	The 'Shop. Torfabrik official game ball - Internet Explorer' window was closed.	17:12:19	Normal			0.54
	Google Chrome was launched.	17:12:20	Normal			0.82
	Navigating to the https://services.smartbear.com/samples/TestComplete14/smarts page.	17:12:31	Normal			10.49
	The window was dicked with the left mouse button.	17:12:31	Normal			0.49
10						
Ň	The button was clicked with the left mouse button.	17:12:42	Normal			0.78
	The 'Shop. Torfabrik official game ball - Google Chrome' window was closed.	17:12:43	Normal		-	0.61
	Microsoft Edge was launched.	17:12:44	Normal			0.79
	Navigating to the https://services.smartbear.com/samples/TestComplete14/smarts page.	17:12:48	Normal			4.58
	The window was clicked with the left mouse button.	17:12:49	Normal			0.79
	The text 'ball' was entered in the text editor.	17:12:50	Normal			0.50
	The property checkpoint passed: contentText equals (case-sensitive) "Color: Blue, Size: 3"	17:13:15	Normal			1 79
	The property encoded with the left mouse button	17:13:15	Normal			0.09
	The Second was decided with the left mode bottom.	17.13.10	Normal			0.50
- <b>L</b>	The shop, for ablik 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ー 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

- 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。

   <u>https://support.smartbear.com/testcomplete/downloads/samples/</u>
- 2. インストールを実行します。
- 3. サンプルは、<TestComplete Samples>\Mobile\Android\Orders\フォルダー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する方法を紹介します。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 b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。

Workspace				
🕈 Start Page	×			
Ø	<sup>smartbear</sup> TestComple	ete™	D	OCUMENTATION
	RECORD TEST		Ľ	NEW PROJECT
Ci	Getting reate your first test	Started with TestC	omplete	Test M for An

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	
beleer project name, location and type of test to record.	
Project name: *	
Orders	
Project location: *	
C:\Users\tester\Documents\TestComplete 14Projects\	
lest type:	
2 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.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。

Recor	d Test		×
	Select applic	ation type and settings to record.	
	Web	Mobile application file:	
	Desktop	$\label{eq:sers} \end{tabular} Isers \end{tabular} ublic \end{tabular} Samples \end{tabular} where \end{tabular} be a simple \end{tabular} where $	
	Mobile	$\checkmark$ Reupload the tested application to BitBar when connecting to a device, if needed	
		BitBar API Key:	
		Do not have a BitBar account? Sign up <u>here</u> or <u>learn more</u> .	
		You can connect to your local Annium servers or use BitBar	
		BitBar is an on-demand mobile device cloud providing you access to latest Android and iOS devices.	
	<u>Learn more a</u>	about these settings	
		< <u>B</u> ack Record Cance	el

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]  $\mathcal{F}$ xy $\rho$ xy $\rho$ xi $\mathcal{F}$  $\gamma$ xuhout  $\mathcal{F}$ xuhout  $\mathcal{F}$ 

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のステップ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。

Device Type:       OS Version:       Resolution:       Quick Filter:         Android       All OS Versions       All Resolutions       Search         iound devices:       All Resolution       Search         Name       OS Version       Resolution       Status         Google Nexus 5 6.0.1       6.0.1 (API level: 29)       1080 x 2220       Ready         Actorola Google Nexus 5 6.0.1       6.0.1 (API level: 23)       Full HD (1920 x 1080)       Ready         Actorola Google Nexus 6 6.0.1       6.0.1 (API level: 23)       QHD (2560 x 1440)       Ready         Actorola Google Nexus 6 7.1.1       7.1.1 (API level: 22)       QHD (2560 x 1440)       Ready         Actorola Google Nexus 6 7.1.1       7.1.1 (API level: 21)       Full HD (1920 x 1080)       Subscription only         Gony Xperia ZL C6502       5.1.1 (API level: 22)       Pull HD (1920 x 1080)       Subscription only         Gony Xperia ZL C6503       5.0.2 (API level: 21)       HD (1920 x 1080)       Subscription only         Gony Xperia ZL C6503       5.0.2 (API level: 22)       Full HD (1920 x 1080)       Subscription only         Gony Xperia ZL C6503       5.0.2 (API level: 21)       HD 720 (1280 x 720)       Subscription only         Gony Xperia ZL C6503       5.0.2 (API level: 22)       Full HD (1920 x 1080)       Subscription o	Parameters Custom parameters				
Android       All OS Versions       All Resolutions       Search         iound devices:       Image: Constraint of the second	Device Type: OS Ver	sion:	Resolution:	Qui	ick Filter:
Name         OS Version         Resolution         Status           Boogle Pixel 3a Android 10         10 (API level: 29)         1080 x 2220         Ready           G Google Nexus 5 6.0.1         6.0.1 (API level: 23)         Full HD (1920 x 1080)         Ready           Atotrola Google Nexus 6 6.0.1         6.0.1 (API level: 23)         QHD (2560 x 1440)         Ready           Atotrola Google Nexus 6 7.1.1         7.1.1 (API level: 25)         QHD (2560 x 1440)         Ready           Siony Xperia ZL C6502         5.1.1 (API level: 21)         Full HD (1920 x 1080)         Subscription only           Siony Xperia ZL C6503 (Honami)         5.1.1 (API level: 22)         Full HD (1920 x 1080)         Subscription only           G G3 D855         5.0 (API level: 21)         HD (1920 x 1080)         Subscription only           Gony Xperia Tablet Z         5.1.1 (API level: 22)         Full HD (1920 x 1080)         Subscription only           Gony Xperia ZI C6903 (Honami)         5.1.1 (API level: 21)         QHD (2560 x 1440)         Subscription only           Gony Xperia ZI C6904 (Honami)         5.1.1 (API level: 21)         QHD (2560 x 1440)         Subscription only           Gony Xperia Tablet Z         5.0.2 (API level: 21)         QHD (2560 x 1440)         Subscription only           Gony Xperia Tablet Z         5.0.2 (API level: 21) <td< th=""><th>Android 🔹 All OS</th><th>Versions -</th><th>All Resolutions</th><th>▼ Sec</th><th>arch</th></td<>	Android 🔹 All OS	Versions -	All Resolutions	▼ Sec	arch
Name         OS Version         Resolution         Status           Google Pixel 3a Android 10         10 (API level: 29)         1080 x 2220         Ready           Google Nexus 5 6.0.1         6.0.1 (API level: 23)         Full HD (1920 x 1080)         Ready           Aotorola Google Nexus 6 6.0.1         6.0.1 (API level: 23)         QHD (2560 x 1440)         Ready           Aotorola Google Nexus 6 7.1.1         7.1.1 (API level: 25)         QHD (2560 x 1440)         Ready           Garange Galaxy A7 SM-A700F         5.0.2 (API level: 21)         Full HD (1920 x 1080)         Subscription only           Sony Xperia ZL C6502         5.1.1 (API level: 22)         Full HD (1920 x 1080)         Subscription only           Sony Xperia ZL C6503         5.0.2 (API level: 21)         HD 720 (1280 x 720)         Subscription only           Sony Xperia ZL C6503         5.0.2 (API level: 22)         Full HD (1920 x 1080)         Subscription only           Sony Xperia ZL C6503         5.0 (API level: 21)         HD 720 (1280 x 720)         Subscription only           Sony Xperia ZL C6903 (Honami)         5.1.1 (API level: 22)         Full HD (1920 x 1080)         Subscription only           Sony Xperia Tablet Z         5.0 (API level: 21)         UP (2560 x 1440)         Subscription only           Sony Xperia Tablet Z         5.0.2 (API level: 21) <t< th=""><th>Found devices:</th><th></th><th></th><th></th><th>1</th></t<>	Found devices:				1
Soogle Pixel 3a Android 10         10 (API level: 29)         1080 x 2220         Ready           G Google Nexus 5 6.0.1         6.0.1 (API level: 23)         Full HD (1920 x 1080)         Ready           Aotorola Google Nexus 6 6.0.1         6.0.1 (API level: 23)         QHD (2560 x 1440)         Ready           Aotorola Google Nexus 6 7.1.1         7.1.1 (API level: 25)         QHD (2560 x 1440)         Ready           Samsung Galaxy A7 SM-A700F         5.0.2 (API level: 21)         Full HD (1920 x 1080)         Subscription only           Siony Xperia ZL C6502         5.1.1 (API level: 22)         Full HD (1920 x 1080)         Subscription only           Siony Xperia ZL C6503         5.0.2 (API level: 21)         HD 720 (1280 x 720)         Subscription only           Siony Xperia ZL C6503         5.0.2 (API level: 22)         Full HD (1920 x 1080)         Subscription only           Siony Xperia ZL C6503         5.0.2 (API level: 21)         HD 720 (1280 x 720)         Subscription only           Siony Xperia Tablet Z         5.0.2 (API level: 22)         Full HD (1920 x 1080)         Subscription only           Siony Xperia Tablet Z         5.0.2 (API level: 21)         UP (2560 x 1440)         Subscription only           Siony Xperia Tablet Z         5.0.2 (API level: 21)         UP (2560 x 1440)         Subscription only           Siony Xperia Tablet Z	Name	OS Version	Resolu	ition	Status
G Google Nexus 5 6.0.1         6.0.1 (API level: 23)         Full HD (1920 x 1080)         Ready           Aotorola Google Nexus 6 6.0.1         6.0.1 (API level: 23)         QHD (2560 x 1440)         Ready           Aotorola Google Nexus 6 7.1.1         7.1.1 (API level: 25)         QHD (2560 x 1440)         Ready           Gamsung Galaxy A7 SM-A700F         5.0.2 (API level: 21)         Full HD (1920 x 1080)         Subscription only           Gony Xperia ZL C6502         5.1.1 (API level: 22)         Full HD (1920 x 1080)         Subscription only           Gony Xperia ZL C6503         5.0.2 (API level: 21)         HD 720 (1280 x 720)         Subscription only           Gony Xperia ZL C6503         5.0.2 (API level: 22)         Full HD (1920 x 1080)         Subscription only           Gony Xperia ZL C6503         5.0.2 (API level: 21)         HD 720 (1280 x 720)         Subscription only           Gony Xperia ZL C6903 (Honami)         5.1.1 (API level: 22)         Full HD (1920 x 1080)         Subscription only           Gony Xperia Tablet Z         5.0 (API level: 21)         QHD (2560 x 1440)         Subscription only           Gony Xperia Tablet Z         5.1.1 (API level: 22)         WUXGA (1920 x 1200)         Subscription only           Gony Xperia Tablet Z         5.0.2 (API level: 21)         HD 720 (1280 x 720)         Subscription only           Gony X	Google Pixel 3a Android 10	10 (API leve	el: 29) 1080 x	2220	Ready
Antorola Google Nexus 6 6.0.1         6.0.1 (API level: 23)         QHD (2560 x 1440)         Ready           Antorola Google Nexus 6 7.1.1         7.1.1 (API level: 25)         QHD (2560 x 1440)         Ready           Gamsung Galaxy A7 SM-A700F         5.0.2 (API level: 21)         Full HD (1920 x 1080)         Subscription only           Sony Xperia ZL C6502         5.1.1 (API level: 22)         Full HD (1920 x 1080)         Subscription only           Sony Xperia ZL C6503         5.0.2 (API level: 21)         HD 720 (1280 x 720)         Subscription only           Sony Xperia ZL C6503         5.0.1 (API level: 22)         HD (1920 x 1080)         Subscription only           Sony Xperia ZL C6903 (Honami)         5.1.1 (API level: 21)         HD (2560 x 1440)         Subscription only           Sony Xperia Tablet Z         5.1.1 (API level: 22)         WUXGA (1920 x 1080)         Subscription only           Sony Xperia Tablet Z         5.1.1 (API level: 21)         UP (2560 x 1440)         Subscription only           Sony Xperia Tablet Z         5.1.1 (API level: 21)         WUXGA (1920 x 1200)         Subscription only           Sony Xperia Tablet Z         5.0.2 (API level: 21)         HD 720 (1280 x 720)         Subscription only           Subscription Sony Xperia Tablet Z         5.0.2 (API level: 21)         HD 720 (1280 x 720)         Subscription only	LG Google Nexus 5 6.0.1	6.0.1 (API	evel: 23) Full HD	(1920 x 1080)	Ready
Antorola Google Nexus 6 7.1.1         7.1.1 (API level: 25)         QHD (2560 x 1440)         Ready           Bamsung Galaxy A7 SM-A700F         5.0.2 (API level: 21)         Full HD (1920 x 1080)         Subscription only           Sonry Xperia ZL C6502         5.1.1 (API level: 22)         Full HD (1920 x 1080)         Subscription only           Sonry Xperia ZL C6503         5.0.2 (API level: 21)         HD 720 (1280 x 720)         Subscription only           Sonry Xperia ZL C6503         5.0.2 (API level: 21)         HD 720 (1280 x 720)         Subscription only           Gony Xperia ZL C6903 (Honami)         5.1.1 (API level: 21)         HD 1920 x 1080)         Subscription only           GG 3D 855         S.0 (API level: 21)         QHD (2560 x 1440)         Subscription only           Gony Xperia Tablet Z         S.1.1 (API level: 22)         WUXGA (1920 x 1200)         Subscription only           Huwei Ascend P8 Lite         S.0.2 (API level: 21)         HD 720 (1280 x 720)         Subscription only           Gamsung Galaxy S6 Edge + SM-G928V         S.1.1 (API level: 22)         QHD (2560 x 1440)         Subscription only	Motorola Google Nexus 6 6.0.1	6.0.1 (API le	evel: 23) QHD (2	2560 x 1440)	Ready
Samsung Galaxy A7 SM-A700F         5.0.2 (API level: 21)         Full HD (1920 x 1080)         Subscription only           Sony Xperia ZL C6502         5.1.1 (API level: 22)         Full HD (1920 x 1080)         Subscription only           Sony Xperia ZR C5502         5.0.2 (API level: 21)         HD 720 (1280 x 720)         Subscription only           Sony Xperia Z1 C6903 (Honami)         5.1.1 (API level: 22)         Full HD (1920 x 1080)         Subscription only           GG 3 D855         5.0 (API level: 21)         QHD (2560 x 1440)         Subscription only           Sony Xperia Tablet Z         5.1.1 (API level: 22)         WUXGA (1920 x 1200)         Subscription only           sony Xperia Tablet Z         5.1.1 (API level: 21)         QHD (2560 x 1440)         Subscription only           subscription Galaxy S6 Edge + SM-G928V         5.1.1 (API level: 21)         HD 720 (1280 x 720)         Subscription only	Motorola Google Nexus 6 7.1.1	7.1.1 (API	evel: 25) QHD (2	2560 x 1440)	Ready
Sony Xperia ZL C6502         5.1.1 (API level: 22)         Full HD (1920 x 1080)         Subscription only           Sony Xperia ZR C5502         5.0.2 (API level: 21)         HD 720 (1280 x 720)         Subscription only           Sony Xperia Z1 C6903 (Honami)         5.1.1 (API level: 22)         Full HD (1920 x 1080)         Subscription only           GG 3 D855         5.0 (API level: 21)         QHD (2560 x 1440)         Subscription only           Sony Xperia Tablet Z         5.1.1 (API level: 22)         WUXGA (1920 x 1200)         Subscription only           Huwei Ascend P8 Lite         5.0.2 (API level: 21)         HD 720 (1280 x 720)         Subscription only           Samsung Galaxy S6 Edge + SM-G928V         5.1.1 (API level: 22)         QHD (2560 x 1440)         Subscription only	Samsung Galaxy A7 SM-A700F	5.0.2 (API le	evel: 21) Full HD	(1920 x 1080)	Subscription only
Single Service         Solution of the service	Sony Xperia ZL C6502	5.1.1 (API	evel: 22) Full HD	(1920 x 1080)	Subscription only
Sony Xperia Z1 C6903 (Honami)         5.1.1 (API level: 22)         Full HD (1920 x 1080)         Subscription only           .G G3 D855         5.0 (API level: 21)         QHD (2560 x 1440)         Subscription only           .Gory Xperia Tablet Z         5.1.1 (API level: 22)         WUXGA (1920 x 1200)         Subscription only           .dwwei Ascend P8 Lite         5.0.2 (API level: 21)         HD 720 (1280 x 720)         Subscription only           .dwwei Galaxy S6 Edge + SM-G928V         5.1.1 (API level: 22)         QHD (2560 x 1440)         Subscription only	Sony Xperia ZR C5502	5.0.2 (API le	evel: 21) HD 720	) (1280 x 720)	Subscription only
G G3 D855         5.0 (API level: 21)         QHD (2560 x 1440)         Subscription only           Sony Xperia Tablet Z         5.1.1 (API level: 22)         WUXGA (1920 x 1200)         Subscription only           Huawei Ascend P8 Lite         5.0.2 (API level: 21)         HD 720 (1280 x 720)         Subscription only           Gamsung Galaxy S6 Edge + SM-G928V         5.1.1 (API level: 22)         QHD (2560 x 1440)         Subscription only	Sony Xperia Z1 C6903 (Honami)	5.1.1 (API	evel: 22) Full HD	(1920 x 1080)	Subscription only
Sony Xperia Tablet Z         5.1.1 (API level: 22)         WUXGA (1920 x 1200)         Subscription only           Huawei Ascend P8 Lite         5.0.2 (API level: 21)         HD 720 (1280 x 720)         Subscription only           Samsung Galaxy S6 Edge + SM-G928V         5.1.1 (API level: 22)         QHD (2560 x 1440)         Subscription only	LG G3 D855	5.0 (API lev	el: 21) QHD (2	2560 x 1440)	Subscription only
Huawei Ascend P8 Lite         5.0.2 (API level: 21)         HD 720 (1280 x 720)         Subscription only           Samsung Galaxy S6 Edge + SM-G928V         5.1.1 (API level: 22)         QHD (2560 x 1440)         Subscription only	Sony Xperia Tablet Z	5.1.1 (API	evel: 22) WUXG/	A (1920 x 1200)	Subscription only
Gamsung Galaxy S6 Edge + SM-G928V 5.1.1 (API level: 22) QHD (2560 x 1440) Subscription only	Huawei Ascend P8 Lite	5.0.2 (API le	evel: 21) HD 720	) (1280 x 720)	Subscription only
	Samsung Galaxy S6 Edge + SM-G928	V 5.1.1 (API le	evel: 22) QHD (2	2560 x 1440)	Subscription only
Samouno Calavu Moto 2.5C 0.1E E. 0 (ADT Jouraly 2.1) Eull HD (1020 yr 1080). Surbocriation poly	Computer Colover Note 2 CC 01E	E O (ADT lov	ملايات (11 يام	(1020 x 1090)	Eubocription only
	Application:				
Application:					

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 .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。

	Mobile Screen				×
€	🖕 🎝 र 🖻 Google Pixel За	Android	10 - 🚡	टे ि 🔽	
1	🗊 Orders				÷.,
	🕒 New 🖉	Ed	lit	Delete	
	John Smith Jr 1, MyMoney, 12.09.201	3			
	Clare Jeffersor 2, FamilyAlbum, 03.03.2	<b>ו</b> 2012			
	Susan McLarer 1, MyMoney, 04.04.201	ן 3			
	Charles Dodge 1, MyMoney, 11.07.201	son ₃			
	Steve Johns 1, ScreenSaver, 03.03.2	:011			
	Samuel Cleme 2, MyMoney, 02.03.201	ns 2			
	Bob Feather 1, FamilyAlbum, 12.06.2	2014			
\$	d)) <	1 0			Ů

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。

Mobile Screen	×
🔍 🖕 🖧 • 🖻 Google Pixel 3a Android 10 • 💼 🖬 🕼 🕻 🕻	
Orders	
Product:	
MyMoney	,
Quantity:	
2	
Price per unit (\$):	
100	
Discount (%):	
0	
Total (\$):	
200	
Customer Name:	
Samuel Clemens	
Street:	
3, Garden st.	
City:	_
Hillsberry, UT	
State:	
US	
Oradi	
Visa	
Master Card	
- American Express	
4 <sup>3</sup> 4 <sup>3</sup> 4 <sup>3</sup> 4 <sup>3</sup>	Ċ

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。

Sector TestComplete Recording									×
Record	Pause	Stop	Add Check	Run App +	Add Comment	Test Type 👻		Scrip Ext	t Ŧ
Parent: Object:									

[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。

Checkpoint Wizard		×
Select a checkpoint:		
$\rightarrow$ Text or property (most popular)		
$\rightarrow$ Image		
$\rightarrow$ File		
$\rightarrow$ Table or Grid		
→ Database		
$\rightarrow$ Web metrics		
Learn more about these settings		
	Next >	Cancel

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。

Checkpoint Wizard	×
Specify an object to check	
Drag the target to an object     Click the target button and drag it to the desired object. Release the mouse button to select it.     Show me how it works.	
Point and press the shortcut (Shift + Ctrl + A) Click this button. Make the desired window or control visible, hover the mouse pointer over it and press Shift + Ctrl + A. Show me how it works.	
Object:	
Aliases.Device.process_smartbear_example_orders.listview1.textview_text1	•
Preview:	
Mark Twain	
Learn more about these settings	
< <u>B</u> ack <u>N</u> ext >	Cancel

[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。

	Properties	OCR		
Specify one or mu	tiple properties to chec	k		
Property	E Condition	Value	Category	Select
Enabled	Equals	▼ true	Standard	
Text	Equals (case	<ul> <li>Mark Twain</li> </ul>	Extended	✓

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。

kpoint Wizard					
Select data to	check				
	Properti	es	OCR		
Specify one o	r multiple prope	erties to check			
Property		Condition	Value	Category	Select
Enabled		Equals 👻	true	Standard	
Text		Equals (case 👻	Mark Twain	Extended	$\checkmark$
Aliases.Device.proc	ess_smartbear_exam	ple_orders.listview1.	textview_text1.T	iext Mor	e Properties
Aliases.Device.proc	ess_smartbear_exam	ple_orders.listview1.	textview_text1.T	iext Mor	e Properties

- 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。

🕈 Start Page 🛛 🔂 Tes	t1 ×			
Operations -	🗄 👧 Record 📑 Run 🚔 Add Parameter X	Add Variable	${}         $	
Search P	Item	Operation	Value	Description
Test Actions	Run Remote Device	- percent	"https://appium.bitbar.com/wd/hub/", "{\"automationName\":\"U	Connect to BitBar device
hp On-Screen Action	V S Device			
Run TestedApp	v S process_smartbear_example_orders			
Menu Action	textview_text12	Touch		Simulates a touch on the
Process Action	button_editbutton	Touch		Simulates a touch on the
C Run Keyword Test	edittext_cust_name	Touch	"Mark Twain"	Simulates a touch on the
Pup Script Poutipe	button ok	Touch		Simulates a touch on the
Run Test	Property Checkpoint		Aliases.Device.process_smartbear_example_orders.listview1.te	Checks whether the 'Texi
Run Code Snippet				
Call Object Method				
rind Object				
🚭 If Object				
Image Based Action				
OCR Action				
Close Application				
📄 Drag Dialog				
Object Reference				
Logging				
Web				
Checkpoints _				
Test Steps 🔓 Variable	s 📑 Parameters			
Test Visualizer				

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。

🕈 Start Page 🛛 🔂 Test	×			
Operations 무	Record 💦 Run 🚋 Add Parameter 🄏	Add Variable		
Search P		, <b>_</b> ,		
	Item	Operation	Value	Description
Test Actions	Run Remote Device		"https://appium.bitbar.com/wd/hub/", "{\"automationName\":\"U	Connect to BitBar device
hp On-Screen Action	V S Device			
Run TestedApp	v S process_smartbear_example_orders			
Menu Action	textview_text12	Touch		Simulates a touch on the
	button_editbutton	Touch		Simulates a touch on the
ho Process Action	edittext_cust_name	Touch		Simulates a touch on the
약 Run Keyword Test	edittext_cust_name	Keys	"Mark Twain"	Simulates keypresses on
Run Script Routine	htton_ok	Touch		Simulates a touch on the
Run Test	Property Checkpoint		${\it Aliases. Device. process\_smartbear\_example\_orders. listview 1. te}$	Checks whether the 'Tex'

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。

🕈 Start Page ×	Т	est1	×						
Operations	Ą	11		Record	Run	🗮 Add Parameter X	Add Variable		
Search	م ر			•.	_	· · ·			
Tool Astron		1		Item			Operation	Value	Description
Test Actions	_			- j 🖓 🖡 R	un Remote De	evice		"https://appium.bitbar.com/wd/hub/", "{\"automationName\":\"U C	Connect to BitBar device
hp On-Screen Action				~ 🔇 D	evice				
Run TestedApp				- V (	process sm	nartbear example orders			
Menu Action					http://www.com/	w_text12	Touch	S	Simulates a touch on the
					h button	_editbutton	Touch	S	Simulates a touch on the
ho Process Action					h edittex	t_cust_name	Touch	S	Simulates a touch on the
嘴 Run Keyword Test					k. edittex	t_cust_name	Keys	"Mark Twain" S	Simulates keypresses on
Run Script Routine					htton_	_ok	Touch	S	Simulates a touch on the
🕒 Run Test				Pi	operty Check	φoint		Aliases.Device.process_smartbear_example_orders.listview1.te C	Checks whether the 'Texi
( )									

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。

Start Page 🛛 🔂 Test	1 X			
Operations 7	Record 💦 Run 🛼 Add Parameter 🄏	Add Variable	$\Rightarrow \boxed{1}  (\uparrow  \downarrow  \leftarrow \rightarrow  \downarrow  (\uparrow  \downarrow  (\downarrow  \downarrow  \downarrow  (\downarrow  \downarrow  (\downarrow  \downarrow  (\downarrow  \downarrow  $	
Search P				
	Item	Operation	Value	Description
Test Actions	Run Remote Device		"https://appium.bitbar.com/wd/hub/", "{\"automationName\":\"U	Connect to BitBar device
hp On-Screen Action	V S Device			
Run TestedApp	v S process_smartbear_example_orders			
Manu Ankan	textview_text12	Touch		Simulates a touch on the
Menu Action	button editbutton	Touch		Simulates a touch on the
ho Process Action	edittext_cust_name	Touch		Simulates a touch on the
👕 Run Keyword Test	edittext_cust_name	Keys	"Mark Twain"	Simulates keypresses on
Run Script Routine	button_ok	Touch		Simulates a touch on the
Run Test	- Yoperty Checkpoint		Aliases.Device.process_smartbear_example_orders.listview1.te	Checks whether the 'Tex
6.3 _ · · · ·				

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。

🕈 Start Page 🗴 🔂 T	est1 ×				
Operations -	: 🦲	Record 💦 Run 🗮 Add Parameter 🔏	Add Variable	$\mathbb{E} \stackrel{1}{\models} 1 \leftarrow \rightarrow$	
Search P			0	Neture	Description
Test Actions	1	Item	Operation	value	Description
Test Actions		Run Remote Device		"https://appium.bitbar.com/wd/hub/", "{\"automationName\":\"U	Connect to BitBar device
hp On-Screen Action		V 🔇 Device			
Run TestedApp		v S process_smartbear_example_orders			
Manu Antina		textview_text12	Touch		Simulates a touch on the
		button editbutton	Touch		Simulates a touch on the
ho Process Action		👆 edittext_cust_name	Touch		Simulates a touch on the
📭 Run Keyword Test		edittext_cust_name	Keys	"Mark Twain"	Simulates keypresses on
Run Script Routine		htton_ok	Touch		Simulates a touch on the
Run Test		Property Checkpoint		Aliases.Device.process_smartbear_example_orders.listview1.te	Checks whether the 'Tex'

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検索式と、

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。

Project Workspace Object Browser		
C Project Explorer ? □ 쿠 ×	Workspace	? 🗆 🗙
: 🗳 📮 🗟 🖳 🕼 🌾 -	🚔 Start Page × 🔂 Test1 × 🔯 NameMapping ×	•
Project Suite 'Orders' (1 project)     V      Orders	🗄 🙀 Add Object 🔞 Ed <u>i</u> t 🍖 <u>R</u> efresh 🍋 <u>H</u> ighlig	jht 🔹 🎼 Iemplates 🏶 Search 🖉
Execution Plan	Mapped Objects 🛛 🕹	Aliases.Device.process_smartbear_example_orders.listview1.textview_text1
KeywordTests +     Test1     NameMapping     TestedApps     Project Suite Logs	Alases. Drag mapped objects here. Device     v process_smartbear_example_orders     button_editbutton     v listview1     textview_text1     textview_text12     v scrollview     button_ok     edittext_cust_name	Image: Constraint of the second se
		Selectors
		i 🚳 🙀 💫 🖓 🔀
		Constraints of the constrai

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ーを表示します。



このインジケーターは、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ーの■[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。

Project Workspace Object Browser								
Correct Explorer ? □ + ×	□ 🕂 × 🚺 Workspace ? □ ×							
: 🕰 📭 🕞 🙋 🕼 🔶 -	😫 📴 😰 🕼 🏚 Test1 × 🎄 Keyword TestLog [Test1] 4/1/202 ×							
Project Suite 'Orders' (1 project)     Orders	🤅 Go to Next: 🙁 Error 👻 [	lext: 8 Error 🝷 🕞 🔷 Create Jira Issue 🝷 🎉 📜 🗲 🄶 Select <u>P</u> anel 🔹						
Execution Plan	Log Items	Test Log						□ ×
KeywordTests +	🗸 🗸 Keyword Test Log [Test1]	💈 🗸 Error 🔥 🗸 Warning 🗠 🗸 M	1essage 💫 🗸 Event 🥑 🗸 Checkpoint				Search	م
NameManning		Type Message		Time	Priority	Has Pict	Link	Time Diff (sec)
TestedAnne		Device "Pixel" was connected.		15:33:59	Normal			0.00
Project Suite Logs		The control was touched.		15:34:02	Normal			12.34
V Corders Logs		The control was touched.		15:34:10	Normal			7.54
Keyword Test Log [Test1] 4/1/2		The control was touched.		15:34:18	Normal	<b></b>		8.05
		The control was touched.		15:34:27	Normal			8.98
		The keystrokes 'Mark Twain' were seen and the second se	ent to the control.	15:34:38	Normal			11.20
		The control was touched.		15:34:47	Normal			9.16
		The control was touched.		15:34:55	Normal			8.03
		I he control was touched.		15:35:04	Normal			9.23
		line control was touched.		15:35:12	Normal			8.18
		3 The control was touched.	the state of the s	15:35:20	Normal			7.50
		The property diecepoint passed: Te	xt equais (case-sensitive) Mark Twart .	15:55:22	Normal			2.17
Picture Details		Picture Details						
		Result The property checkpoint Spassed: A Details	liases.Device.process_sistview1.textview_text1.Text	equals (cas	e-sensitiv	<b>e)</b> "Mark Tw	vain".	
		Object	Aliases.Device.process smartbear example or	ders.listview	1.textviev	w text1		
		Property	Text					
		Condition	equals (case-sensitive)					
Information Actual value "Mark Twain"								
	End Time: 4/1/2021 3:33 PM End Time: 4/1/2021 3:35 PM	Expected value	"Mark Twain"					
<	Run Time: 0:01:29							

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

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。

### 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 iOS 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。

Edit	Orders	+
Bob Feath 1, FamilyAlbun	ner n, 2010-12-03	>
Samuel Cl 2, MyMoney, 2	lemens 2009-12-12	>
Charles D 1, MyMoney, 2	odgeson 2009-05-12	>
John Smit 1, MyMoney, 2	th Jr 2009-05-07	>
Clare Jeff 2, FamilyAlbur	<b>erson</b> n, 2009-04-05	>
Susan Mc 1, MyMoney, 2	Laren 2008-05-12	>
Steve Joh	I <b>NS</b> r, 2008-04-04	>
/	ريجي	6
Orders	Settings	About

### アプリケーションを入手するには

- 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\iOS\Orders\フォルダーにインストールされます。 アプリケーションは Mac 上でコンパイルする必要があります。

91

### 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する方法を紹介します。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 Dをクリックして、API Key をクリップボードにコピーします。

SMARTEEAR         BitBar         «         MOBILE APP TESTING         Pashboard	\$	Automation Testing Test an app using your own automation scripts	Live Te Start testi our many	sting ng your app on one of devices	Al-drive Use Al Test Use Al Test Access Groups	8	
Automation	-	Create Automated Test	Security Center	M Preferences	Start Al 1] Log out		
-		Account Details		General	Security Center		0
		This data is needed for the order p	rocess.	Time Zone	Info Panel Ch	ange password Session Settings	API Delete account
Basic Inform Basic doe that we First Name Last Name	ation need to know about you	Core Enter country state Dry Enter chy Enter address Egy Code Enter address Egy Code		Running Tests Default set timeout System Default: 30 User Interface Datableard Datableard Default Reset to default		API key C foroad Avit kay C oncode Avit kay C on	ia D
Email Phone Fields marked with *	are required	_	Save changes	My Integrations	Annual Sector		

# 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。

1 Workspace	
A Start Page ×	
MARTBEAR TestComplete <sub>™</sub>	DOCUMENTATION
RECORD TEST	NEW PROJECT
Getting Started Create your first test with TestComplete	Test M for An

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	
Select project name, location and type of test to record.	
Project name: *	
Orders	
Project location: *	
C:\Users\tester\Documents\TestComplete 14 Projects\	
• Keyword	
O P <u>y</u> thon	
Participation of the 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。

94

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。

Record Test			×
Select a	applic	ation type and settings to record.	
Web		Mobile application file:	
Desk	top	Users\Public\Documents\TestComplete Samples\Mobile\iOS\Orders\Orders Application\Orders.ipa	]
Mobi	ile	$\checkmark$ Reupload the tested application to BitBar when connecting to a device, if needed	
		BitBar API Key:	
			]
		Do not have a BitBar account? Sign up here or learn more.	
		You can connect to your local Appium servers or use BitBar	
		BitBar is an on-demand mobile device cloud providing you access to latest Android and iOS devices.	
? Learr	n more al	bout these settings	
		< Back Record Cance	I

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]  $\mathcal{F}_{xy}$ 

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のステップ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。

Parameters Custom p	arameters			
Device Type:	OS Version	:	Resolution: Q	uick Filter:
IOS	<ul> <li>All OS Vers</li> </ul>	ions -	All Resolutions     Search	
Found devices:				Q
Name		OS Version	Resolution	Status
Apple iPhone 7 A 1778 1	3.4.1	13.4.1	1334 x 750	Ready
Apple iPhone SE 2020 A	2296 13.4.1	13.4.1	1334 x 750	Ready
Apple iPhone 4S A 1387	9.3.5	9.3.5	DVGA (960 x 640)	Subscription only
Apple iPhone 6S Plus A1	687 10.2	10.2	Full HD (1920 x 1080)	) Subscription only
Apple iPhone 6S Plus A1	687 12.0.1	12.0.1	Full HD (1920 x 1080)	) Subscription only
Apple iPad Pro A1584 11	1.2.6	11.2.6	2048 x 2732	Subscription only
Apple iPhone XR A 1984	12.1.2	12.1.2	1792 x 828	Subscription only
Apple iPhone 6S A 1688	12.4.1	12.4.1	1334 x 750	Subscription only
Apple iPhone 6 A 1586 1	2.4.1	12.4.1	Full HD (1920 x 1080)	) Subscription only
Apple iPhone 7 14.0.1 -	JS	14.0.1		Subscription only
Apple iPhone 11 A2221	13.4.1	13.4.1	1792 x 828	Subscription only
Apple iPad Air A1474 12	.4.4	12.4.4	QXGA (2048 x 1536)	Subscription only
Apple iDhope VC May A1	021 12 1	10.1	10/0 / 0600	Subscription only

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。

Select BitBar device	×
BitBar Local Appium	
Parameters Custom parameters	
Parameters (JSON):	
<pre>{     "bundleID": "    </pre>	
OK Cancel <u>H</u> elp	

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。

Select BitBar device	×
BitBar Local Appium	
Parameters Custom parameters	
Parameters (JSON):	
{ "bundleID": "Orders", "autoAcceptAlerts": false	
ОК Са	ncel Help

6. 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。

ł	
	"bundleId": " <your_bundle_id>",</your_bundle_id>
	"autoAcceptAlerts": false
}	

7. [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。

Edit	Orders	+
Bob Feather 1, FamilyAlbum, 2010	-12-03	>
Samuel Clemer 2, MyMoney, 2009-12	<b>15</b> 2-12	>
Charles Dodge 1, MyMoney, 2009-05	<b>SON</b> 5-12	>
John Smith Jr 1, MyMoney, 2009-05	5-07	>
Clare Jefferson 2, FamilyAlbum, 2009	) -04-05	>
Susan McLarer 1, MyMoney, 2008-05	<b>1</b> 5-12	>
Steve Johns 1, ScreenSaver, 2008	-04-04	>
/	Je.	6

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

🞯 Mobile Screen	×
🔍 🝙 Apple iPhone 7 A1778 13.4.1 • 📓 🖸	le le 🕼
No Service 🗢 9:41	AM 🛃
Edit Ord	lers +
Bob Feather 1, FamilyAlbum, 2010-12-03	>
Samuel Clemens 2, MyMoney, 2009-12-12	>
Charles Dodgeson 1, MyMoney, 2009-05-12	>
John Smith Jr 1, MyMoney, 2009-05-07	>
Clare Jefferson 2, FamilyAlbum, 2009-04-05	>
Susan McLaren 1, MyMoney, 2008-05-12	>
Steve Johns 1, ScreenSaver, 2008-04-04	>
Orders Sett	C Dings About

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。

🔯 Mobile	Screen					
🔍 🖬 Ap	ople iPhone 7 A17	18 13.4.1 • 🛅	0 6	C 🔽		
No Serv	vice 穼	9	:41 AM			
Don	е	0	rders			+
•	Bob Fe	ather Ibum, 2010	)-12-03			>
•	Samue 2, MyMon	l Cleme ey, 2009-1	<b>ns</b> 2-12			>
•	Charles	s Dodge ey, 2009-0	eson 5-12			>
•	John Si 1, MyMone	mith Jr 9, 2009-0	5-07			>
•	Clare J 2, FamilyA	effersoi Ibum, 200	<b>n</b> 9-04-05			>
•	Susan I 1, MyMone	McLare	n 5-12			>
•	Steve J 1, ScreenS	lohns aver, 2008	3-04-04			>
	Orders	s	<b>%</b> iettings		About	
			•			

5. 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。

- 6. [Edit Order] パネルの [Save] ボタンをクリックし、変更を確定します。
- 7. 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。

8. [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。

9. 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。

ckpoint Wizard	×
Select a checkpoint:	
$\rightarrow$ Text or property (most popular)	
→ Image	
$\rightarrow$ File	
$\rightarrow$ Table or Grid	
→ Database	
$\rightarrow$ Web metrics	
Learn more about these settings	
Next >	Cancel

ターゲットアイコン(<sup>(</sup>)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。

🞯 Mobile Screen			×
🕀 🖻 Apple iPhone 7 A1778	13.4.1 • 🚡 🖸 🔀 🕻		
No Service 奈	9:41 AM		
Edit	Orders		+
Bob Feather 1, FamilyAlbum, 20	10-12-03		>
Mark Twain 2, MyMoney, 2009	-12-12		>
Charles Dodo 1, MyMoney, 2009	<b>geson</b> -05-12		>
John Smith J 1, MyMoney, 2009	<b>r</b> -05-07		>
Clare Jeffers 2, FamilyAlbum, 20	<b>on</b> 009-04-05		>
Susan McLar 1, MyMoney, 2008	<b>en</b> -05-12		>
Steve Johns 1, ScreenSaver, 20	08-04-04		>
	J	6	
Orders	Settings	About	
	۲		

• 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。

point		
Speci	fy an object to check	
•	Drag the target to an object Click the target button and drag it to the desired object. Release the mouse button to select it. Show me how it works.	
	Point and press the shortcut (Shift + Ctrl + A) Click this button. Make the desired window or control visble, hover the mouse pointer over it and press Shift + Ctrl + A. <u>Show me how it works</u> .	
Object	t:	
Aliase	es.Device.process_smartbear_orders.table.statictext_mark_twain	<b>≣</b> - (
	W:	
	Mark Twain	
3	Mark Twain	

#### [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。

heckpoint Wizard						×
Select data to	check					
Select data to	check					
	Properti	es	oc	ર	]	
Specificana a	multiple prop	arties to she	-k			
specify one of	multiple prope	erties to che	JK .			
Property	<u>1</u>	Condition	Value	Category	Select	
Text		Equals (case	<ul> <li>Mark Twain</li> </ul>	Extended	$\checkmark$	
Aliases Device proc	acc cmarthear order	e table etatictevt	mark twain Taxt	[	More Properties	i
Allases.Device.proc	ess_sinal deal_order:	s. table is table if	_mark_twain.rext	l	Hore Properces	
P Learn more about the second seco	out these settings					
				Back F	-inish Can	cel

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。

Properties       OCR         Decision or multiple properties to check         Property       Education         Category       Select         Text       Equals (case • Mark Twain         Extended       Image: Category         Select data to check       Mark Twain         Extended       Image: Category         Select data to check       Mark Twain         Extended       Image: Category         Select data to check       Mark Twain         Extended       Image: Category         Select data to check       Image: Category         Select data to check       Mark Twain         Extended       Image: Category         Select data to check       Mark Twain         Mark Twain       Image: Category         Alases.Device.process_smartbear_orders.table.statictext_mark_twain.Text       More Properties	Checkpoint Wizard						х
Propertie   Category   Select     Text   Category   Select     Text     Equals (case • Mark Twain   Extended   Mark Twain     Extended     Text     Rest.Device.process_smartbear_orders.table.statictext_mark_twain.Text     More Properties	Select data to	check					
Specify one or multiple properties to check         Image: the two the two the two the two the two two the two two the two two the two		Propertie	es	OCR			
Property       Endition       Value       Category       Select         Text       Equals (case > Mark Twain       Extended       Image: Category       Select         Aliases.Device.process_smartbear_orders.table.statictext_mark_twain.Text       More Properties	Specify one or	r multiple prope	erties to chec	k			
Text       Equals (case • Mark Twain       Extended         Image: Comparison of the second state of the se	Property	<u>E</u>	Condition	Value	Category	Select	
Aliases.Device.process_smartbear_orders.table.statictext_mark_twain.Text       More Properties         Image: Learn more about these settings	Text		Equals (case	<ul> <li>Mark Twain</li> </ul>	Extended	✓	
Aliases.Device.process_smartbear_orders.table.statictext_mark_twain.Text       More Properties         Image: Control of the section of							
Aliases.Device.process_smartbear_orders.table.statictext_mark_twain.Text       More Properties         Image: Comparison of the section							
Aliases.Device.process_smartbear_orders.table.statictext_mark_twain.Text       More Properties         Image: Comparise about these settings							
Aliases.Device.process_smartbear_orders.table.statictext_mark_twain.Text       More Properties         C       Learn more about these settings							
Aliases.Device.process_smartbear_orders.table.statictext_mark_twain.Text       More Properties         Common about these settings							
Aliases.Device.process_smartbear_orders.table.statictext_mark_twain.Text       More Properties         Image: Control of the section of							
Aliases.Device.process_smartbear_orders.table.statictext_mark_twain.Text       More Properties         Image: Control of the section of							
Aliases.Device.process_smartbear_orders.table.statictext_mark_twain.Text       More Properties         Image: Comparison of the section							
Aliases.Device.process_smartbear_orders.table.statictext_mark_twain.Text       More Properties         Image: Control of the section of							
Aliases.Device.process_smartbear_orders.table.statictext_mark_twain.Text       More Properties         Image: Control of the section of							
Aliases.Device.process_smartbear_orders.table.statictext_mark_twain.Text       More Properties         Image: Contract and the settings       Image: Contract and the settings							
Aliases.Device.process_smartbear_orders.table.statictext_mark_twain.Text       More Properties         Image: Contract the section of t							
Aliases.Device.process_smartbear_orders.table.statictext_mark_twain.Text       More Properties         Image: Control of the section of							
Learn more about these settings	Aliases.Device.proc	ess_smartbear_order	s.table.statictext_	mark_twain.Text	N	Nore Properties	
Learn more about these settings							
	? Learn more abo	out these settings					
< Back Einish Cancel				< 8	ack <u>F</u> inis	sh Cance	el

- [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。

🕈 Start Page 🗴 🔂 Tes	st1 ×		
Operations 🕂	: 👧 Record 📭 Run 溝 Add	Parameter Ҳ Add Vari	able 🕞 💽 $\uparrow$ 🗼 $\leftarrow$ $\rightarrow$
Search P	Itam	Operation	Value
Test Actions	Bun Remote Device	Operation	"https://appium.bitbar.com/wd/hub/", "{\"autoAcceptAlerts\":f
h On-Screen Action	Device		
Run TestedApp	process_smartbear_or	ders	
Menu Action	button_edit	Touch	
	▶ statictext_samuel_	demens Touch	
Process Action	http://www.interview.com/	Touch	
Run Keyword Test	button_clear_text	Touch	
Run Script Routine	textfield	Keys	"Mark Twain"
🕒 Run Test	button_return	Touch	
Run Code Snippet	button_save	Touch	
R. Call Object Method	button_yes	Touch	
	button_orders	Touch	
	Dutton_done	louch	
Notest If Object	Property Checkpoint		Aliases.Device.process_smartbear_orders.table.statictext_ma
Image Based Action			
CR Action			
Logging			
Web			
Checkpoints			
Mobile			
Statements			
Miscellaneous			
Performance			
📔 Test Steps 🔀 Variable	s 😫 Parameters		

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。

🔺 Start Page 🛛 🔒	Test1 ×	c		
Operations	æ : 👩	💂 Record 📑 Run 🗦 Add Parameter	X Add Varia	able 🕞 🚺 $\uparrow$ \downarrow $\leftarrow$ $\rightarrow$
Search .	<u>م</u>	Itom	Operation	Value
Test Actions	· -	Run Remote Device	Operation	"https://appium.bitbar.com/wd/hub/", "{\"autoAcceptAlerts\":f
hp On-Screen Action		V Device		
🖏 Run TestedApp		process_smartbear_orders		
Menu Action		button_edit	Touch	
Dana Antina		h_ statictext_samuel_demens	Touch	
Process Action		hp textfield	Touch	
🏆 Run Keyword Test		http://www.alion_dear_text	Touch	
🖶 Run Script Routine		textfield	Keys	"Mark Twain"
🕒 Run Test		button_return	Touch	
		http://www.save	Touch	
		htton_yes	Touch	
Call Object Method		http://www.angle.com	Touch	
🏹 Find Object		http://www.weighted.com	Touch	
🌄 If Object		Property Checkpoint		${\sf Aliases.Device.process\_smartbear\_orders.table.statictext\_ma}$

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。

🚔 Start Page 🛛 🔒	Test1 ×	c		
Operations	æ : 👩	💂 Record 📑 Run 📑 Add Parameter 🗸	X Add Variat	ble 🕞 🚺 $\uparrow$ \downarrow $\leftarrow$ $\rightarrow$
Search	<u>م</u>	74	Onersting	M-b
Test Actions	<u> </u>	Pun Remote Device	Operation	"https://appium.hithar.com/wd/bub/"_"/\"autoAcceptAlerte\":f
Con-Screen Action				https://applain.bibai.com/wa/hdb/ , {{autoAcceptaicits{
Run TestedApp		v process smartbear orders		
Menu Action		button_edit	Touch	
		▶ statictext_samuel_demens	Touch	
Ro Process Action		textfield	Touch	
🙄 Run Keyword Test		button_clear_text	Touch	
Run Script Routine		textfield	Keys	"Mark Twain"
Run Test		button_return	Touch	
{***} Pun Code Spinnet		button_save	Touch	
		button_yes	Touch	
Call Object Method		button_orders	Touch	
🗞 Find Object		button_done	Touch	
😼 If Object		Property Checkpoint		${\sf A} iases. {\sf D} evice. {\sf process\_smartbear\_orders.table.statictext\_ma}$

🔺 Start Page 🗴 🔂 T	est1 ×			
Operations 4		Record 💦 Run 🗮 Add Parameter	X Add Varia	able 🔂 🔁 🕇 👃 🔶 →
Search P			•	
Test Actions	ī	Item	Operation	Value
Test Actions		Run Remote Device		"https://appium.bitbar.com/wd/hub/", "{\"autoAcceptAlerts\":f
hp On-Screen Action		V Device		
📮 Run TestedApp		process_smartbear_orders		
Menu Action		hp button_edit	Touch	
		h_ statictext_samuel_clemens	Touch	
ho Process Action		http://www.interview.com/www.com/www	Touch	
🏆 Run Keyword Test		button_dear_text	Touch	
🖶 Run Script Routine		textfield	Keys	"Mark Twain"
🕒 Run Test		htton_return	Touch	
		button_save	Touch	
Run Code Snippet		button_yes	Touch	
💫 Call Object Method		button_orders	Touch	
Notest Find Object		button_done	Touch	
🍋 If Object				Aliases.Device.process_smartbear_orders.table.statictext_ma

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。

🚔 Start Page 🛛 🔂 Te	est1 ×			
Operations 7	:	Record 📑 Run 🗮 Add Parameter	X Add Varia	able $\bigcirc$ $\uparrow$ $\downarrow$ $\leftarrow$ $\rightarrow$
Search P			•	
Test Actions	i	Item	Operation	Value
		Run Remote Device		"https://appium.bitbar.com/wd/hub/", "{\"autoAcceptAlerts\":f
hp On-Screen Action		Y Device		
🔍 Run TestedApp		process_smartbear_orders		
		htton_edit	Touch	
Menu Action		statictext samuel demens	Touch	
ho Process Action		textfield	Touch	
👕 Run Keyword Test		button_clear_text	Touch	
Run Script Routine		textfield	Keys	"Mark Twain"
🕒 Run Test		http://www.intervectors.com	Touch	
{***} Run Code Sninnet		htton_save	Touch	
- Kurr coue snipper		button_yes	Touch	
Call Object Method		htton_orders	Touch	
Reference Find Object		htton_done	Touch	
🊱 If Object		Property Checkpoint		Aliases.Device.process_smartbear_orders.table.statictext_ma

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。

🚔 Start Page ×	📜 Te	st1 ×				
Operations	<b></b>	:	Record 📑 Run	🟣 Add Parameter	r 🄏 Add Varia	able [ 🛅 👔 🕇
Search	Q	_				
Test Astron			Item		Operation	Value
Test Actions	_		🖳 🖵 Run Remote D	)evice		"https://appium.bitbar.com
hp On-Screen Action			Y 📃 Device			
Run TestedApp			process_s	martbear_orders		
			httor	n_edit	Touch	
Menu Action			h static	text samuel demens	Touch	

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の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 インジケーターは、実行されたテスト操作に関する情報を表示します。



このインジケーターは、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ーの [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。

インジケーターの **[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。

Project Workspace Object Browser								
Co Project Explorer ? □ ₽ ×	🗾 Workspace	∑ Workspace ? □ ×						
4 4 6 2 4 • •	🚔 Start Page × 🔒 Test1 ×	🔭 StatfAge × 👼 Testi × 🙆 Keyword Testlog [Testi] 7/22/20 ×						
<ul> <li>Project Suite 'Orders' (1 project)</li> <li>Project Suite 'Orders' (1 project)</li> </ul>	🕴 Go to Next: 😆 Error 👻 [	🔶 Create Jira Issue 👻 🏂 🅇	Select Panel -					
Execution Plan	Log Items	Testlog						
<ul> <li>KeywordTests +</li> <li>Test1</li> </ul>	Keyword Test Log [Test1]	😒 🗸 Error 🛕 🗸 Warning 😳 🗸 Message 🖕 🗸 Event 🥥 🗸 Checkpoint 🦉						
NameMapping		Type Message		Time	Priority	Has Picture	Link	Time Diff (sec)
> TestedApps		Device "Apple iPhone" was	connected.	17:34:33	Normal			0.00
Project Suite Logs		The control was touched.		17:34:44	Normal			10.70
Orders Logs		The control was touched.		17:34:52	Normal			7.72
Seyword Test Log [Test1] 7/22/		The control was touched.		17:35:01	Normal			0.90
		The keystrokes 'Mark Twai	n' were sent to the control.	17:35:25	Normal			11.46
		The control was touched.		17:35:37	Normal			11.60
		The control was touched.		17:35:45	Normal			8.19
		The control was touched.		17:35:52	Normal			7.27
		The property checkpoint p.	assed: Text equals (case-sensitive) "Mark Twain".	17:35:59	Normal			6.84
		Picture Details Result The property checkpoint ③ pa Details	<b>ssed:</b> Aliases.Device.process_se.statictext_mark_twain.Text	equals (case-sensitin	ve) "Mark ī	īwain".		^
		Object	Aliases Device, process, smartbear, orders, table,	tatictext mark twai	in			
		Property	Text		-			
	Information	Condition	equals (case-sensitive)					
	Start Time: 7/22/2021 5:32 PM	Actual value	"Mark Twain"					
	End Time: 7/22/2021 5:35 PM Run Time: 0:03:48	Expected value "Mark Twain"					~	

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。

Project Workspace Object Browser	
CompositionProject Explorer? $\square \not = \times$	🗾 Workspace
: 🕰 📮 🔂 🖳 🕼 🌾 -	🚔 Start Page 🛛 🔂 Test1 🗡
<ul> <li>Project Suite 'Orders' (1 project)</li> <li>Orders</li> </ul>	Go to Next: 😢 Error 👻 [
🖳 Execution Plan	Log Items
Y 🔂 KeywordTests +	Keyword Test Log [Test1]
Test1	
NameMapping	
TestedApps	
🔨 🏠 Project Suite Logs	
🗸 🚱 Orders Logs	
Keyword Test Log [Test1] 4/1/2	

### 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。

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

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。
- ▶ エキ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.xlsoft.com/jp/services/xlsoft\_form.html

# 索引

Α	す
Android アプリケーションのテスト67	スクリプト5
Ι	ち
iOS アプリケーションのテスト90	チェックポイント
Naming オブジェクト	作成 23 チェックポイント 13
	チェックポイント (Checkpoints) 23
<b>O</b> Open アプリケーション12	<b>て</b> テクニカル サポートとリソース115
S	デスクトップ アプリケーションのテスト
Stores 13 U	テスト レコーディング
UI テスト5	テスト結果
Wab アプリケーションのテスト 38	モバイル (iOS)111 テスト
	結果の分析 34, 87, 111
お	テスト
オブジェクト naming9	作成20, 45, 71, 73, 99
オブジェクト モデル9 オブジェクト名のマッピング31, 57	テスト 実行32, 58, 86, 109 テスト
き	うって 自動テストについて5
ー キーワード テスト5 機能テスト5	テスト テスト タイプ 5
ę	テスト テスト タイプ5
サポートとリソース 115	テストプロジェクト6
<b>し</b> 自動テスト5	テスト プロジェクト Web プロジェクトの作成
	アスト フロジェクト デスクトップ プロジェクトの作成17
	アスト プロジェクト

モバイル プロジェクトの作成 (Android)
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
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
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
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
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
テストの作成
Web
テストの作成
デスクトップ20

<b>ね</b> ネーム マッピング
テストの実行 一時停止110 停止110
テストのレコーディング Web45 デスクトップ20
テストの実行 モバイル
テストの実行 別のブラウザー63
テストの実行 バッチ実行 33
テストの実行 デスクトップ
テストの実行 停止
テストの実行 初期状態
テストの実行 一時停止33, 59, 86
テストの実行 Web 58
テストのYF成 モバイル (Android)
テストの作成

ム マッピング	31

ネー

### は

ペネル
-----

### 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

モバイル (Android)	67
プロジェクト	
モバイル (iOS)	
プロジェクト項目	6

### ほ

ホロイトボックフ	アプリケーション	19
ホシイトホックス	///////////////////////////////////////	12

### ŧ

モバイル	アプリケーションのテスト	 90
<u> </u>		 00

### Þ

ユーザー	アクションのシミュレーション	80
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	
レコーディングしたテストの分析	
モバイル	106

## ろ

ロノ	ゲ	
	Web	60
ロノ	ゲ	
	ソースにジャンプ	61
ロノ	Ґ	
	デスクトップ	34