



# HP Unified Functional Testing

ソフトウェア・バージョン: 12.50

Windows® オペレーティング・システム

## チュートリアル

ドキュメント・リリース日: 2015 年 7 月 (英語版)  
ソフトウェア・リリース日: 2015 年 7 月

## ご注意

### 保証

HP製品、またはサービスの保証は、当該製品、およびサービスに付随する明示的な保証文によってのみ規定されるものとします。ここでの記載は、追加保証を提供するものではありません。ここに含まれる技術的、編集上の誤り、または欠如について、HPはいかなる責任も負いません。

ここに記載する情報は、予告なしに変更されることがあります。

### 権利の制限

機密性のあるコンピューターソフトウェアです。これらを所有、使用、または複製するには、HPからの有効な使用許諾が必要です。商用コンピューターソフトウェア、コンピューターソフトウェアに関する文書類、および商用アイテムの技術データは、FAR12.211および12.212の規定に従い、ベンダーの標準商用ライセンスに基づいて米国政府に使用許諾が付与されます。

### 著作権について

© Copyright 1992 - 2015 Hewlett-Packard Development Company, L.P.

### 商標について

Adobe® およびAcrobat® は、Adobe Systems Incorporated (アドビシステムズ社) の登録商標です。

Google™ およびGoogleマップ™ は、Google Incの商標です。

Intel® およびPentium® は、Intel Coporation の米国およびその他の国における商標です。

Microsoft®、Windows®、Windows® XPおよびWindows Vista® は、米国におけるMicrosoft Corporationの登録商標です。

OracleとJavaは、Oracle Corporationおよびその関連会社の登録商標です。

## ドキュメントの更新情報

このマニュアルの表紙には、以下の識別情報が記載されています。

- ・ソフトウェアバージョンの番号は、ソフトウェアのバージョンを示します。
- ・ドキュメントリリース日は、ドキュメントが更新されるたびに変更されます。
- ・ソフトウェアリリース日は、このバージョンのソフトウェアのリリース期日を表します。

更新状況、およびご使用のドキュメントが最新版かどうかは、次のサイトで確認できます。

<https://softwaresupport.hp.com>

このサイトを利用するには、HP Passportへの登録とサインインが必要です。HP Passport IDの登録は、次のWebサイトから行なうことができます。<https://softwaresupport.hp.com> にアクセスして **【Register】** をクリックしてください。

## サポート

HPソフトウェアサポートオンラインWebサイトを参照してください。 <https://softwaresupport.hp.com>

このサイトでは、HPのお客様窓口のほか、HPソフトウェアが提供する製品、サービス、およびサポートに関する詳細情報をご覧いただけます。

HPソフトウェアオンラインではセルフソルブ機能を提供しています。お客様のビジネスを管理するのに必要な対話型の技術サポートツールに、素早く効率的にアクセスできます。HPソフトウェアサポートのWebサイトでは、次のようなことができます。

- 関心のあるナレッジドキュメントの検索
- サポートケースの登録とエンハンスメント要求のトラッキング
- ソフトウェアパッチのダウンロード
- サポート契約の管理
- HPサポート窓口の検索
- 利用可能なサービスに関する情報の閲覧
- 他のソフトウェアカスタマーとの意見交換
- ソフトウェアトレーニングの検索と登録

一部のサポートを除き、サポートのご利用には、HP Passportユーザーとしてご登録の上、サインインしていただく必要があります。また、多くのサポートのご利用には、サポート契約が必要です。HP Passport IDを登録するには、次のWebサイトにアクセスしてください。<https://softwaresupport.hp.com> にアクセスし、**[Register]** をクリックしてください。

アクセスレベルの詳細については、次のWebサイトをご覧ください。

<https://softwaresupport.hp.com/web/softwaresupport/access-levels>

## HP Software Solutions統合とベストプラクティス

**HP Software Solutions Now** (<https://h20230.www2.hp.com/sc/solutions/index.jsp>) では、HPソフトウェアのカタログ記載製品がどのような仕組みで連携、情報の交換、ビジネスニーズの解決に対応するのかご確認いただけます。

**Cross Portfolio Best Practices Library** (<https://hpln.hp.com/group/best-practices-hpsw>) では、ベストプラクティスに関するさまざまなドキュメントや資料をご覧頂けます。

# 目次

UFTチュートリアルへようこそ .....	7
UFT のガイドとリファレンス .....	8
その他のオンライン・リソース .....	10
 第1部: Unified Functional Testing の概要 .....	11
自動テストの利点 .....	12
UFT テスト・プロセス .....	13
UFT メイン・ウィンドウ .....	15
 第2部: アプリケーションの分析とテストの作成 .....	21
テスト対象のアプリケーションについての考察 .....	22
フライト予約アプリケーションの確認 .....	23
すべてのテスト用のソリューションの作成 .....	25
 第3部: GUI 自動テストの作成と実行 .....	28
レッスン 1: GUI テストとアクションの作成 .....	29
レッスン 2: オブジェクト・リポジトリの作成 .....	33
UFT テスト・オブジェクトの認識 - 概要 .....	34
練習 2a: アプリケーションからのオブジェクトの追加 .....	35
練習 2b: [ナビゲートして学習] を使用したオブジェクト・リポジトリの作成 .....	42
レッスン 3: テストへのステップの追加 .....	46
GUI テストでのテスト・ステップの追加 - 概要 .....	47
練習 3a: キーワード・ビューでの Login アクションへのステップの追加 .....	47
キーワード・ビューおよびエディタでの Login アクションの分析 .....	52
練習 3b: 記録による FlightFinder アクションへのステップの追加 .....	54
練習 3c: [ツールボックス] 表示枠を使用した Select Flight アクションへのステップの追加 .....	58
練習 3d: ステップ・ジェネレータを使用した Book Flight アクションへのステップの追加 .....	62
高度な練習 3e (オプション) - エディタを使用したステップの追加 .....	67
レッスン 4: GUI テストの実行と分析 .....	73
練習 4a: テストの実行 .....	74
練習 4b: 実行結果の操作 .....	76
練習 4c: 実行結果の分析 .....	77
レッスン 5: ステップとオブジェクトのパラメータ化 .....	79
テスト, アクション, オブジェクトのパラメータ化 - 概要 .....	80
練習 5a: パラメータ化用のテストの作成 .....	80
練習 5b: データ・テーブル・パラメータの定義 .....	81

練習 5c : データ・テーブルへのパラメータ値の追加 .....	85
練習 5d : パラメータ化されたテストの実行 .....	87
レッスン 6 : チェックポイントおよび出力値の作成 .....	94
チェックポイントおよび出力値のタイプについて .....	95
練習 6a : チェックポイント・テストの作成 .....	97
練習 6b : オブジェクトの値の検査 .....	98
練習 6c : テーブルの値の検査 .....	101
練習 6d : テキストの値の検査 .....	107
練習 6e : オブジェクト・リポジトリ内のチェックポイントの管理 .....	113
練習 6f : チェックポイントがあるテストの実行および分析 .....	115
練習 6g : 出力値のテストの作成 .....	118
練習 6h : 出力値のステップの追加 .....	119
レッスン 7 : 関数と関数ライブラリの作成 .....	129
関数および関数ライブラリ - 概要 .....	130
練習 7a : 関数の作成 .....	130
練習 7b : 関数ライブラリのテストへの関連付け .....	132
練習 7c : 関数を使用した検査の実行 .....	133
レッスン 8 : テストでの Insight の使用 .....	139
Insight オブジェクトの識別 - 概要 .....	140
練習 8a : Insight オブジェクト用のテストの作成 .....	140
練習 8b : オブジェクト・リポジトリへの Insight オブジェクトの追加 .....	141
練習 8c : テストでの Insight オブジェクトの使用 .....	144
 第4部: API 自動テストの作成と実行 .....	 148
レッスン 1 : API テストの作成 .....	149
レッスン 2 : 単純な API テスト・ステップの作成 .....	150
レッスン 3 : 標準アクティビティを使用した API テスト・ステップの作成 .....	155
UFT API テストの標準アクティビティ - 概要 .....	156
練習 3a : 標準アクティビティによるテストの作成 .....	157
レッスン 4 : API テスト・ステップのパラメータ化 .....	161
API テスト・ステップのパラメータ化 - 概要 .....	162
練習 4a : データ・ソースを元にしたテスト・ステップのパラメータ化 .....	162
練習 4b : 前のステップの出力を元にしたテスト・ステップのパラメータ化 .....	173
練習 4c : カスタム式を使用した複数のソースによるテストのパラメータ化 .....	176
レッスン 5 : API テストの実行 .....	183
練習 5a : テストの実行 .....	184
練習 5b : 実行結果の操作 .....	185
練習 5c : 実行結果の分析 .....	186
レッスン 6 : Web サービスの API テストの作成と実行 .....	188
練習 6a : Web サービス・テストの作成 .....	189
練習 6b : Web サービスのインポート .....	189

練習 6c : Web サービスのテストの作成とパラメータ化 .....	192
練習 6d : Web サービスのテストの実行 .....	198
レッスン 7 : REST サービスの API テストの作成と実行 .....	202
練習 7a : REST サービスのテストの作成 .....	203
練習 7b : REST サービスの構造の作成 .....	203
練習 7c : REST サービス・メソッドを使用したテストの作成 .....	208
練習 7d : REST サービスのテストの実行 .....	211
練習 7e : REST サービスの競合の解決 .....	213
レッスン 8 : Web アプリケーション・サービスの API テストの作成と実行 (WADL) .....	217
練習 8a : Web アプリケーション・サービス用のテストの作成 .....	218
練習 8b : Web アプリケーション・サービス・モデルのインポート .....	219
練習 8c : Web アプリケーション・サービスのメソッドの編集 .....	221
練習 8d : Web アプリケーション・サービスのメソッドを使用したテストの構築 .....	224
練習 8e : Web アプリケーション・サービスのテストの実行 .....	229
 第5部: 1 つのテストでの GUI および API テストの作成と実行 .....	 232
レッスン 1 : GUI テストと API テストをまとめて実行するテストの作成 .....	233
レッスン 2 : API テストの GUI テストからの呼び出し .....	234
レッスン 3 : API テストを呼び出す GUI テストの実行 .....	238
 その他の情報 .....	 241
 フィードバックの送信 .....	 244

# UFTチュートリアルへようこそ

『UFT チュートリアル』は、UFT を使ってアプリケーションをテストするための基本的な方法をご自分のペースで学習できるガイドです。このチュートリアルでは、自動化された GUI テストおよびAPI テストの作成、実行、実行結果の分析という工程を学んでいきます。

チュートリアルを完了すれば、学んだ技術を自分のアプリケーションのテストで活用できます。

**注:** Web アプリケーション用 GUI テストの作成と実行の詳細については、<UFT インストール・フォルダ>\help フォルダにある『UFT Web アプリケーション用 GUI テスト・チュートリアル』を参照してください。

## チュートリアルの対象読者と範囲

このチュートリアルは、UFT を初めて使用するユーザを対象としています。UFT、QuickTest または Service Test の予備知識は必要ありません。テストのコンセプトや機能テストのプロセスについての一般的な知識が役立つ場合もありますが、必ず必要というわけではありません。UFT では、GUI テスト、API テスト、ビジネス・プロセス・テスト、また、GUI テストと API テストから成る複合テスト、さらに、テストの呼び出しも作成できます。

**注:** このチュートリアルで参照するファイル・システム・パスは、Windows 7 オペレーティング・システムのものであります。ほかのオペレーティング・システムでは、パスがわずかに異なる場合があります。

## UFT での BPT の使用

テストのほかに、BPT をサポートする ALM サーバに接続されていれば、UFT ではビジネス・プロセス・テストで使用するキーワード駆動、スクリプト、API の各ビジネス・コンポーネントも作成できます。このチュートリアルでは GUI テストと API テストの作成手順を説明しますが、大部分はキーワード駆動コンポーネント、スクリプト・コンポーネント、または API コンポーネントの作成手順にも適用できます。ビジネス・コンポーネントと BPT の詳細については、『HP Unified Functional Testing ユーザーズ・ガイド』および『HP Business Process Testing ユーザーズ・ガイド』を参照してください。

**注:** このガイドでは、別途記載のないかぎり、「**Application Lifecycle Management**」または「**ALM**」とは現在サポートされている ALM または Quality Center のすべてのバージョンを指します。一部の機能およびオプションは、使用している ALM または Quality Center のエディションではサポートされていない可能性があります。

サポートされている ALM または Quality Center のバージョンの一覧については、『HP Unified Functional Testing 使用可能製品マトリクス』（「[HP サポート・マトリクス](#)」ページ（要 HP passport 登録）から入手可能）を参照してください。

ALM または Quality Center のエディションの詳細については、『HP Application Lifecycle Management ユーザーズ・ガイド』 または『HP Quality Center ユーザーズ・ガイド』を参照してください。

## UFT のガイドとリファレンス

次の表は、UFT のガイド、オンライン・ヘルプ、リファレンスのリストです。

**注：**下記のガイドの更新状況のチェックには、HP ソフトウェア製品マニュアル Web サイト (<https://softwaresupport.hp.com/group/softwaresupport/search-result?keyword=>) をご利用ください。

### はじめに

リファレンス	説明
新機能	Unified Functional Testing の最新バージョンの最新機能を説明します。  [新機能] には、 <b>Unified Functional Testing</b> の [ヘルプ] メニューからもアクセスできます。
<a href="#">製品ムービー</a>	主要な製品機能を紹介するショート・ムービーを視聴するには、リンクをクリックするか、[ヘルプ] > [製品ムービー] を選択してください。
Readme	Unified Functional Testing に関する最新ニュースおよび最新情報が記載されています。  最新版の Readme ファイルは、HP ソフトウェア・マニュアル Web サイト <a href="https://softwaresupport.hp.com/group/softwaresupport/search-result?keyword=">https://softwaresupport.hp.com/group/softwaresupport/search-result?keyword=</a> (HP Passport が必要) から入手してください。
UFT PAM	使用可能製品マトリクス (PAM) には、このバージョンの UFT でサポートされるテクノロジーと統合に関する最新情報があります。
UFT チュートリアル	UFT チュートリアルは、印刷可能な自習用のガイドです。複合アプリケーションの GUI テスト、API テスト、フル UFT テストを手順に従って作成できます。
Web アプリケーション用 GUI テスト・チュートリアル	『Web アプリケーション用 GUI テスト・チュートリアル』は、印刷可能な自習用のガイドです。Web アプリケーションの GUI テストを手順に従って作成することで、テスト環境に習熟することができます。



## PDF ガイド

ガイド	説明
UFT ユーザーズ・ガイド	『HP Unified Functional Testing ユーザーズ・ガイド』では、UFT を使用してアプリケーションをテストする方法を説明します。テストの作成・デバッグ・実行の方法と、テスト・プロセス中に検出された不具合の報告の方法を順を追って説明します。
Run Results Viewer	『HP Run Results Viewer ユーザーズ・ガイド』では、Run Results Viewer を使用して、GUI または API テストのテスト結果を解釈して使用する方法について説明します。
UFT インストール・ガイド	『HP Unified Functional Testing インストール・ガイド』には、UFT をスタンドアロンのコンピュータにインストールしセットアップするための詳細な手順が順を追って記載されています。
UFT クイックスタート	『UFT インストール・クイックスタート・シート』は、UFT の基本インストールを実行する手順について説明しています。
UFT アドイン・ガイド	『HP Unified Functional Testing アドイン・ガイド』は、UFT アドインのサポートと標準的な Windows テストのサポートをセットアップする方法について説明しています。アドインにより、GUI テストとビジネス・コンポーネントを使用して、サポートされるあらゆる環境をテストできます。
新機能	Unified Functional Testing の最新バージョンの最新機能を説明します。
セキュリティ・リファレンス	UFT 使用の際のセキュリティ関連情報などを説明します。
Runtime Engine Guide	ランタイム・エンジンを使用するテストの実行方法についての情報を説明します。（UFT IDE なし）

## リファレンス

リファレンスへのリンクは、UFT オンライン・ヘルプのホーム・ページにあります。

リファレンス	説明
Object Model Reference	『Object Model Reference for GUI Testing』には、各 UFT テスト・オブジェクトの説明、メソッドとプロパティのリスト、構文、例、認識プロパティがあります。
VBScript Reference	VBScript スクリプトの作成に使用するオブジェクト、メソッド、プロパティ、関数などの要素について説明する Microsoft の Visual Basic Scripting のドキュメントです。
Automation Object Model Reference	UFT を別のアプリケーション内から制御できるようにするオブジェクト、メソッド、プロパティがリストされています。
Object Repository Automation Reference	UFT 共有オブジェクト・リポジトリとその内容を UFT の外側から操作するためのオブジェクトについて説明しています。
Run Results Schema Reference	実行結果の XML スキーマの構造を詳細に示し、その XML レポートで使用される要素と属性について説明しています。
Test Object Schema Reference	テスト・オブジェクト設定 XML のコンテンツの作成で使用する要素と属性について説明するリファレンスです。UFT Extensibility プロジェクトを作成するときに使用します。
Object Repository Schema Reference	オブジェクト・リポジトリ・スキーマのために定義される要素と複合型について説明しています。

その他のオンライン・リファレンスについては、PDF ガイドの「ようこそ」の項を参照してください。

## その他のオンライン・リソース

次のオンライン・リソースが利用可能です。

リソース	説明
HP ソフトウェア・サポート・オンライン	HP ソフトウェア・サポート Web サイト ( <a href="http://www.hp.com/go/hpsoftwaresupport">www.hp.com/go/hpsoftwaresupport</a> )。アクセスするには、[ヘルプ] > [HP ソフトウェア サポート] を選択します。  <b>注:</b> 一部のサポートを除き、サポートのご利用には HP Passport ユーザーとしてご登録の上、サインインしていただく必要があります。また、多くのサポートのご利用には、サポート契約が必要です。HP Passport ユーザ ID の登録は、次の場所で行います。 <a href="http://h20229.www2.hp.com/passport-registration.html">http://h20229.www2.hp.com/passport-registration.html</a> (英語サイト)
テスト・フォーラム	<ul style="list-style-type: none"><li>GUI テスト: <a href="http://h30499.www3.hp.com/t5/Unified-Functional-Testing/bd-p/sws-Fun_TEST_SF">http://h30499.www3.hp.com/t5/Unified-Functional-Testing/bd-p/sws-Fun_TEST_SF</a></li><li>API テスト: <a href="http://h30499.www3.hp.com/t5/Service-Test-Support-and-News/bd-p/sws-Serv_TEST_SF">http://h30499.www3.hp.com/t5/Service-Test-Support-and-News/bd-p/sws-Serv_TEST_SF</a></li><li>BPT: <a href="http://h30499.www3.hp.com/t5/Business-Process-Validation/bd-p/sws-BPT_SF">http://h30499.www3.hp.com/t5/Business-Process-Validation/bd-p/sws-BPT_SF</a></li></ul>
UFT 製品ページ	HP Unified Functional Testing 製品ページ ( <a href="http://www8.hp.com/us/en/software-solutions/unified-functional-testing-automated-testing/index.html">http://www8.hp.com/us/en/software-solutions/unified-functional-testing-automated-testing/index.html</a> )。ここには、UFT に関する情報や関連リンクがあります。アクセスするには、[ヘルプ] > [便利なリンク] > [製品ページ] を選択します。
トラブルシューティングとナレッジ・ベース	トラブルシューティング・ページ ( <a href="http://h20230.www2.hp.com/troubleshooting.jsp">http://h20230.www2.hp.com/troubleshooting.jsp</a> )。HP ソフトウェア・サポート Web サイトの HP ソフトウェア・セルフソルブ技術情報 で検索できます。アクセスするには、[ヘルプ] > [ナレッジ ベース] または [トラブルシューティング] を選択します。
HP ソフトウェア・コミュニティ	HP IT Experts Community サイト ( <a href="http://h10124.www1.hp.com/campaigns/IT_Experts/pages/home.html">http://h10124.www1.hp.com/campaigns/IT_Experts/pages/home.html</a> )。ここでは、他の HP ソフトウェア・ユーザと対話し、HP ソフトウェアに関する記事とブログを読み、他のソフトウェア製品のダウンロードにアクセスできます。
HP マニュアル・サイト	HP ソフトウェア製品マニュアル Web サイト ( <a href="http://support.openview.hp.com/selfsolve/manuals">http://support.openview.hp.com/selfsolve/manuals</a> )。ここでは、選択した HP ソフトウェア製品の最新のドキュメントを検索できます。アクセスするには、[ヘルプ] > [便利なリンク] > [HP マニュアル サイト] を選択します。
新機能	UFT の新機能のヘルプ。このバージョンの UFT の新機能と強化された点が記載されています。
製品ムービー	UFT HPLN (HP Live Networks) ページ ( <a href="https://hpln.hp.com/page/uft-120-videos">https://hpln.hp.com/page/uft-120-videos</a> )。ここには製品のムービーがリストされています。
HP ソフトウェア Web サイト	HP ソフトウェア Web サイト ( <a href="http://www.hp.com/go/software">http://www.hp.com/go/software</a> )。ここでは、HP ソフトウェアが提供する製品、サービス、サポートに関する詳細情報をご覧いただけます。新しいソフトウェアのリリース、セミナー、展示会、カスタマ・サポートなどの情報も含まれています。

# 第1部: Unified Functional Testing の概要

このレッスンでは、自動テストおよび UFT の概要について説明します。また、テストの準備にも着手します。

**注:** いくつかの項では、テスト・ステップをコピーして UFT に貼り付けるよう求めることがあるので、このチュートリアルはソフト・コピーの使用をお勧めします。ただし、このチュートリアルには、UFT によって操作が記録されたり、マウスでポイントしたオブジェクトが学習されるセッションがあることに注意してください。そのようなセッションでは、PDF またはヘルプ・ウィンドウにフォーカスを切り替えないようにしてください。

本項には、次の項目が含まれています。

- [自動テストの利点](#) ..... 12
- [UFT テスト・プロセス](#) ..... 13
- [UFT メイン・ウィンドウ](#) ..... 15

# 自動テストの利点

アプリケーションや Web サイトを手動でテストしたことがあれば、これらのアプリケーションを手動でテストすることの欠点にお気付きでしょう。

手動テストは時間のかかる退屈な作業であるうえ、相当量の人的資源を必要とします。何よりも問題なのは、アプリケーションがリリースされる前に手動ですべての機能を十分にテストすることが、時間的に不可能な場合がしばしばあるということです。このことが、重大なバグが検出されないままになっているのではないかと不安を抱かせるのです。

UFT を使ってアプリケーションの GUI および非 GUI（サービス）層のテストを自動化すれば、テスト・プロセスにかかる時間を短縮して手動テストの問題に対処できます。アプリケーションや Web サイトのあらゆる側面を検査するテストを作成でき、アプリケーションや Web サイトが変更されるたびにそのテストを実行できます。その結果、アプリケーションが更新されるたびにテストを更新する必要がなく、一度作成したテストをアプリケーションが変更されるたびに実行できます。

UFT ではこれらのテストを実行する際に、アプリケーションや Web サイトでカーソルを動かし、アプリケーションの GUI でオブジェクトをクリックし、キーボード入力を行い、アプリケーションの API プロセスを実行することで、実際のユーザ操作を再現します。ただし、手動テストと異なり、UFT は、実際のユーザよりも高速でこれらの操作を行います。

自動テストには、数多くの利点があります。

自動テストの利点	
速い	自動テストは実際のユーザが行う手動テストよりも大幅に高速です。
信頼できる	自動テストではまったく同じ操作を正確に繰り返し実行できるため、手動テストに伴う人的ミスを解消できます。
反復可能	同じ操作を繰り返し実行したときにアプリケーションや Web サイトがどのような反応をするかをテストできます。
プログラム可能	複雑で重要度の高いさまざまなシナリオをテストする高度なテストをプログラミングし、通常の手動テストでは簡単に見つからない問題や不具合を見つけることができます。
包括的	アプリケーションや Web サイトのすべての機能を網羅する一連のテストを構築できます。
再利用可能	ユーザ・インタフェースや内部アプリケーション API が変更されている場合でも、アプリケーションや Web サイトのさまざまなバージョンに対して自動テストを再利用できます。

# UFT テスト・プロセス

UFT テストの作成は、次の複数の手順で構成されます。

## 1. アプリケーションの分析

テスト計画の最初の手順は、アプリケーションを分析してテストのニーズを特定することです。

アプリケーションの開発環境は何か。	アプリケーションのユーザ・インタフェース・オブジェクトを使用するには、適切な UFT GUI テスト・アドインをロードする必要があります。たとえば、アプリケーションが Web、.NET、または Java 環境で構築されている場合、これらの環境を使用するのに UFT に Web Add-in、.NET Add-in、または Java Add-in が必要です。
テストの対象となるビジネス・プロセスと機能は何か。	ユーザがアプリケーションで実行するアクティビティと、これらのビジネス・プロセスを実行するためにアプリケーションで実行する必要がある内部アクションについて考えます。  ユーザ・インタフェースでユーザのアクションを再現するには、GUI テスト・ステップを作成します。アプリケーションがバックグラウンドで実行するプロセスを実行するには、API テスト・ステップを作成します。
アプリケーションで標準アプリケーション・アクティビティまたはカスタム設計サービスのいずれが使用されるか。	アプリケーションの API で実行される機能に応じて、UFT API テストで用意されている設定済みのアクティビティを使用するか、カスタム・アクティビティをインポート/作成します。
テストを実行可能な小さな単位に分割するにはどうすればよいか。	テストするプロセスと機能をより小さなタスクに分割することで、その GUI テストに基づいて UFT アクションを作成できます。アクションが小さくモジュール性が高いほどテストの可読性が高くなり、結果として保守も楽になります。

この段階でも、テストの骨組みを作成し、GUI テストへのアクションの追加を開始できます。

## 2. テスト・インフラストラクチャの準備

テストのニーズに基づいて必要なリソースを特定し、それに応じてリソースを作成する必要があります。

- GUI テストの場合、これらのリソースには、アプリケーションのオブジェクトを表すテスト・オブジェクトを含む**共有オブジェクト・リポジトリ**、テスト内で使用するカスタム関数を含む**関数ライブラリ**、UFT でアプリケーションに問題がある場合の対応方法を指定する**回復シナリオ**、一般的な環境変数の定義を含む**環境変数ファイル**、またはテスト・ステップのパラメータ化に使用する**外部データ・テーブル**などが含まれます。
- API テストの場合、これらのリソースには、アプリケーション・サービスのメソッドを記述する **WSDL** または **WADL** ファイル、プロトタイプやアプリケーションの REST プロセスとして使用するために作成する **REST サービス**、**外部データ・ソース**、サービス呼び出しで使用する**仮想化プロジェクト**、テスト・ステップで参照される **.NET アセンブリ**、またはテスト・ス

トップで使用する **Java クラス**などが含まれます。これらのリソースは、UFT でインポートまたは作成する必要があります。

UFT の設定を行って、必要な追加タスク（テストを実行するたびに結果レポートを表示するタスク、テストの実行に対してデバッグを有効または無効にするなど）を UFT で実行できるようにすることも必要です。

### 3. テストの作成と各テストへのステップの追加

テスト・インフラストラクチャおよびリソースの準備ができれば、テストの作成を開始できます。

- GUI テストの場合、1 つ以上の空のテストを作成してアクションを追加することで、テストの骨組みを作成できます。続いて、オブジェクト・リポジトリに適切なアクションを関連付け、キーワード駆動型の方式またはスクリプトの作成によってステップを挿入できるようにします。
- API テストの場合、1 つ以上の空のテストを作成し、アクティビティをテスト・キャンバスにドラッグしてこれらのテストにテスト・ステップを追加し、これらのステップに入力、出力、チェックポイントの各プロパティを定義できます。また、複数回実行するステップをグループ化して、個別のテスト・ステップとして実行できる1つのアクションにすることもできます。
- すべてのテストを1つのソリューションに追加することもできます。ソリューションを使用すると、関連するテストの格納、管理、編集をまとめて実行できるため、テストを閉じてから別のテストを開く必要がありません。
- テストの設定（GUI テストの場合）またはテスト固有のプロパティ（API テストの場合）を構成することもできます。

### 4. テストの拡張

テストはさまざまな方法で拡張できます。

<b>GUI テストの場合：</b>	<ul style="list-style-type: none"><li>テスト・ステップとして<b>チェックポイント</b>を挿入し、アプリケーションが正しく機能しているかどうかを検査します。たとえば、これらのチェックポイントで、ページ、個々のテスト・オブジェクト、またはテキスト文字列の特定の値を検査できます。</li><li>複数のデータ・セットを使ってテスト・ステップの値をパラメータ化することで、アプリケーションが異なる値で同じ操作をどのように実行するかを検査します。これは、テスト・ステップの固定値に代わり<b>パラメータ</b>を使用して行います。</li><li>VBScript を使用して、<b>プログラミングおよび条件文またはループ文</b>やその他のプログラミング・ロジックをテストに追加できます。</li></ul>
<b>API テストの場合：</b>	<ul style="list-style-type: none"><li><b>チェックポイントのプロパティ</b>を選択し、ステップのプロパティの期待値を入力することで、テスト・ステップおよびテスト・ステップの個別のプロパティを検証できます。</li><li>複数のデータ・セットを使ってテスト・ステップのプロパティをパラメータ化することで、アプリケーションが異なる値で同じプロセスをどのように実行するかを検査できます。これは、固定値の代わりに<b>パラメータ</b>を使用して行います。</li><li>UFT のアクティビティ・ウィザードを使って作成される<b>カスタム・コード・アク</b></li></ul>

ティビティ、イベント・ハンドラ、またはカスタム・アクティビティを使用して、テスト・ステップに機能を追加します。

## 5. テストのデバッグ，実行，分析

UFT のデバッグ機能を使ってテストをデバッグし、テストがスムーズに滞りなく実行できるようにします。テストが正常に動作したら、そのテストを実行してアプリケーションの動作を検査します。テストの実行時に、UFT はアプリケーションのユーザ・インタフェースで各ステップを実行するか（GUI テストの実行時）、アプリケーションの API プロセスを実行します（API テストの実行時）。

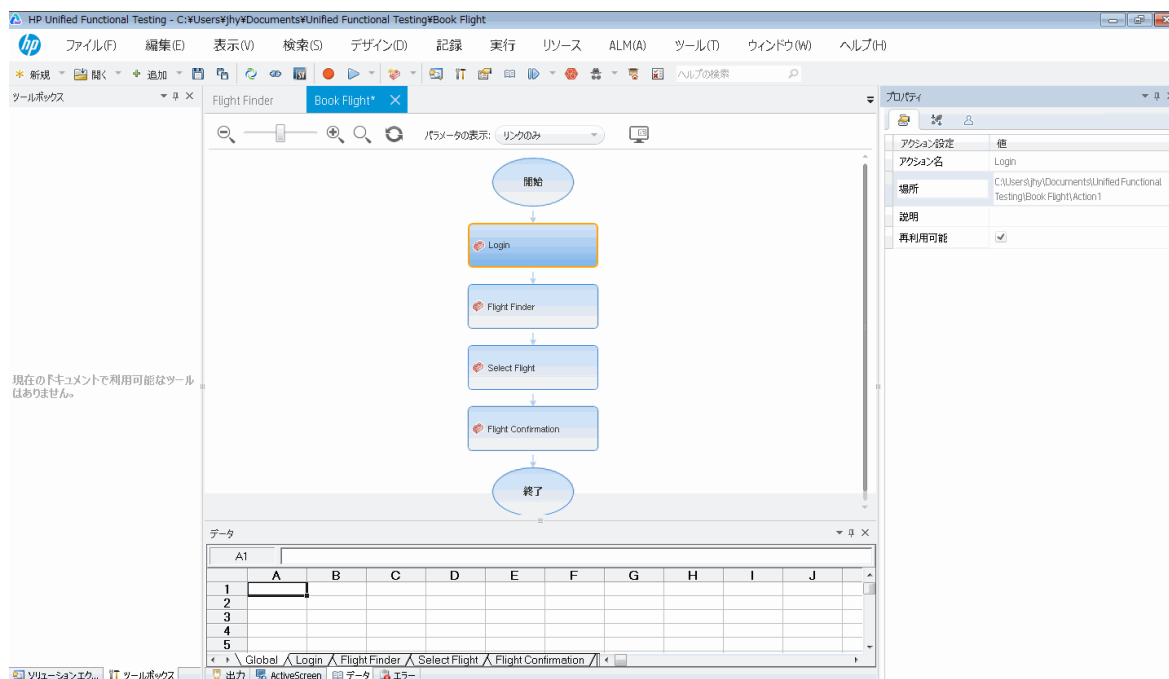
## 6. 不具合の報告

ALM サーバにアクセスできる場合は、発見した不具合をそれぞれの ALM プロジェクトに報告できます。サーバにアクセスできない場合は、不具合をそれぞれの専用の不具合データベースに手動で報告できます。

# UFT メイン・ウィンドウ

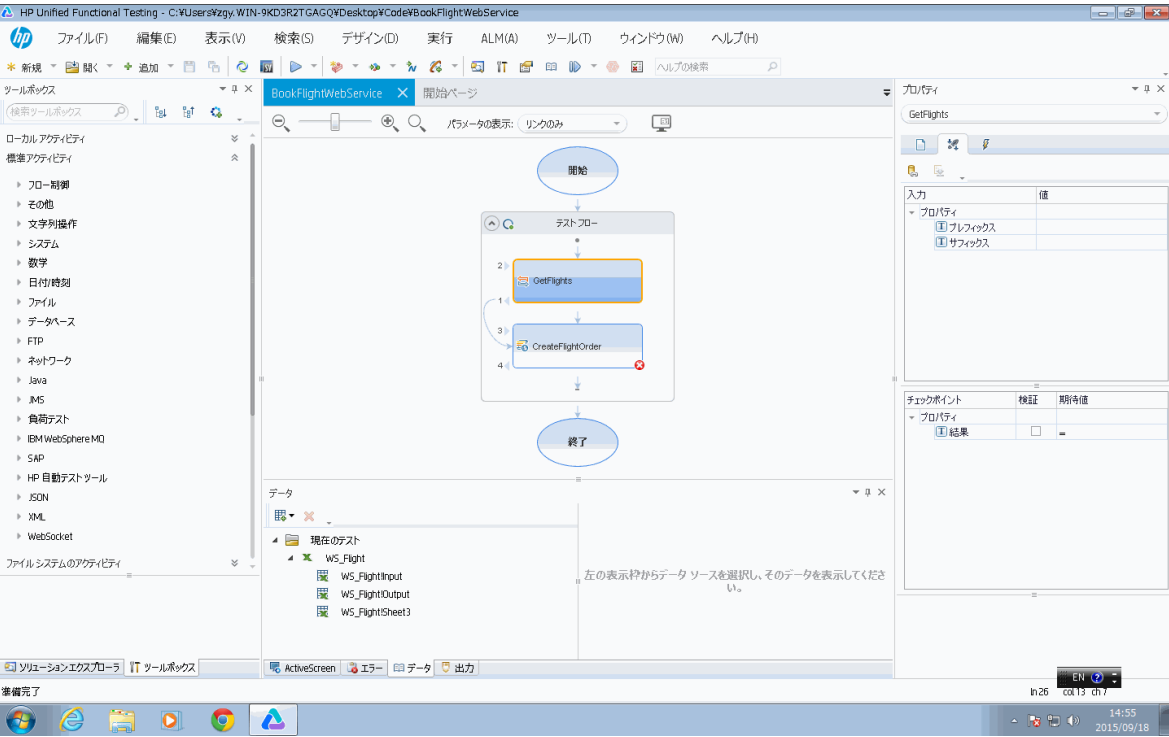
UFT を使用してテストの作成を始める前に、UFT メイン・ウィンドウに慣れておく必要があります。

次の画像は、GUI テストを作成した後に表示される UFT ウィンドウです。キャンバスには、テスト・フローが表示されています。ツールバー、ソリューション・エクスプローラ、[データ] 表示枠、[プロパティ] 表示枠も表示されます。



次の画像は、API テストを作成した後に表示される UFT ウィンドウです。キャンバスには、テスト・

フローが表示されています。ツールバー，[ツールボックス] 表示枠，[データ] 表示枠，[プロパティ] 表示枠も表示されます。



メイン・ウィンドウには，次に示すいくつかの要素が表示されます。

テスト・ドキュメント

開いているドキュメントは，ドキュメント表示枠（UFT ウィンドウの中央）に表示されます。開いているドキュメントに移動して，そのドキュメントにフォーカスを移すには，ツールバーのすぐ下にあるドキュメント・タブを使用します。

ドキュメント表示枠には，次の種類のファイルを表示できます。

テスト	GUI テストと API テストの両方が，キャンバスのそれぞれのタブに表示されます。GUI テスト表示には，テスト・フローとテストに含まれるアクションごとの個別の項目が表示されます。API テストには，テストに含まれる各種ステップのテスト・フローが表示されます。  BPT テストはグリッドに表示され，テストに含まれる個々のコンポーネント，グループ，またはフローがリスト表示されます。
GUI テストのアクション	各 GUI テストには，個別のアクションまたは他のアクションへの呼び出しが含まれます。各アクションは次のビューのいずれかで表示できます。 <ul style="list-style-type: none"><li>キーワード・ビュー：各ステップ（とオブジェクト階層）が，モジュール化されたアイコン・ベースの表形式で表示されます。</li><li>エディタ：各ステップが VBScript の行として，テキストおよびコード・エディタ内に表示されます。オブジェクト・ベースのステップの場合，この VBScript の行によってオブジェクト階層が定義されます。</li></ul>



<b>ビジネス・コンポーネント</b>	<p>ビジネス・コンポーネントでは、アプリケーション内の各ビジネス・プロセスに対するモジュール化された単体のテスト「ユニット」を設計できます。これらのコンポーネントをビジネス・プロセス・テストに追加し、アプリケーション・シナリオとして組み合わせて実行できます。</p> <p>次のいずれかのタイプのビジネス・コンポーネントを表示できます。</p> <ul style="list-style-type: none"> <li>・ <b>キーワードGUI・コンポーネント</b>：これらのコンポーネントは、キーワード・ビューにのみ表示されます。</li> <li>・ <b>スクリプトGUI・コンポーネント</b>：これらのコンポーネントは、キーワード・ビューまたはエディタに表示できます（これらは標準設定でエディタに表示されます）。</li> <li>・ <b>API コンポーネント</b>：これらのコンポーネントは、テスト・キャンバスで表示されます（API テストと同じ形式）。</li> </ul>
<b>関数ライブラリ</b>	<p>関数ライブラリでは、テスト内で使用する関数の作成、編集、変更を行うことができます。これらの関数は単一の関数ライブラリとして作成し、作成した関数ライブラリを各テストに関連付けることで、複数のテストで使用することができます。</p> <p>関数ライブラリはエディタに表示されます。</p>
<b>アプリケーション領域</b>	<p>各 GUI ビジネス・コンポーネントには、アプリケーション領域も含まれています。アプリケーション領域は、コンポーネントのオブジェクト・リポジトリ、関数ライブラリ、構成設定のコンテナとして機能します。各アプリケーション領域は、複数のビジネス・コンポーネントに関連付けることもできます。</p> <p>アプリケーション領域のユーザ・インタフェースは、一連のサイドバー・タブとして表示されます。</p>
<b>ユーザ・コード・ファイル</b>	<p>API テストでは、特別なイベント・ハンドラ・コードまたはカスタム・コードのファイルを追加できます。このコードにより、API テスト・ステップの標準機能の追加や拡張を行うことができます。イベント・ハンドラ・コードは、標準の API テストに含まれる <b>TestUserCode.cs</b> ファイル内に格納されます。</p> <p>これらのファイルはエディタに表示されます。</p>
<b>開始ページ</b>	<p>このページは、UFT の入口で、最近使ったファイル、新機能の説明、製品フォーラムなどのサポート・リンクが用意されています。ショートカット・ボタンを使用して、ドキュメントを作成し、既存ドキュメントを開くことができます。</p>
<b>内部ブラウザ・ページ</b>	<p>開始ページや <b>ヘルプ</b> メニューからアクセスできるような、フォーラムおよびその他の製品関連資料のインターネット・ページを表示できます。</p>

## ツールバーとメニュー

ドキュメント表示枠のほかに、UFT ウィンドウには次の要素が存在します。



- ・ **タイトル・バー**：現在のテストまたはソリューションのパスが表示されます。
- ・ **メニュー・バー**
- ・ **UFT ツールバー**

## 表示枠

UFT ウィンドウには、テスト・ドキュメントの作成と設計に役立つように構成された複数の表示枠が存在します。

これらの表示枠およびツールバー・オプションの一部については、これに続くレッスンで詳しく説明します。その他の表示枠やツールバー・オプションの詳細については、『HP Unified Functional Testing ユーザーズ・ガイド』を参照してください。

名前	ツールバー・ボタン	説明	標準設定の位置
ソリューション・エクスプローラ		現在開いているテスト、コンポーネント、アプリケーション領域、またはソリューションに含まれているテスト、コンポーネント、アプリケーション領域がすべて表示されます。また、現在のテストとコンポーネントに関連付けられているすべてのリソースも表示されます。ソリューション・エクスプローラを使用すると、これらのリソースを管理できます。	UFT ウィンドウの左側のタブ。
ツールボックス		<ul style="list-style-type: none"> <li>GUI テストとコンポーネントの場合：テストで使用するすべてのキーワードが表示されます。オブジェクト（関数の呼び出し）を [ツールボックス] 表示枠からテストにドラッグ・アンド・ドロップできます。</li> <li>API テストとコンポーネントの場合：テストで使用するすべてのアクティビティが表示されます。これらのアクティビティをキャンバスにドラッグ・アンド・ドロップできます。</li> </ul>	UFT ウィンドウの左側のタブ。
【ドキュメント】表示枠	なし	開いているすべてのドキュメントが表示されます。各ドキュメントにはタブがあり、そのタブをクリックすると、フォーカスをそのドキュメントに移動できます。	UFT ウィンドウ中央のラベルがない表示枠。各ドキュメント・タブのラベルには、ドキュメント名が表示されます。  表示方法：テスト・ドキュメントを開きます。
プロパティ		<p>GUI テストとコンポーネントの場合：現在選択しているテスト、アクション、コンポーネント、またはアプリケーション領域のすべてのプロパティが表示されます。</p> <p>API テストとコンポーネントの場合：選択したテスト・ステップ/テスト・フローのすべてのプロパティ、または選択したデータ・ソース（[データ] 表示枠内）のプロパティが表示されます。</p>	UFT ウィンドウの右側の表示枠。  表示方法： <ul style="list-style-type: none"> <li>・ [表示] &gt; [プロパティ] を選択します。</li> <li>・ ツールバーの [プロパティ] ボタンをクリックします。</li> <li>・ キャンバスで API テスト・ステップをダブルクリックします。</li> <li>・ キャンバスで API</li> </ul>

			テスト・ステップを右クリックし、 <b>「プロパティ」</b> を選択します。
<b>データ</b>		テストのパラメータ化を支援します。	UFT ウィンドウの下部のタブ。
<b>出力</b>	なし	実行セッション中に情報が表示されます。	UFT ウィンドウの下部のタブ。  表示するには、 <b>「表示」</b> > <b>「出力」</b> を選択します。
<b>エラー</b>	なし	テストまたはコンポーネントに関する問題のリストが表示されます：テストからの参照の欠落（GUI テストからのオブジェクト・リポジトリや回復シナリオの欠落、または API テストでの外部ファイルへの参照の欠落やプロパティ値の欠落など）	UFT ウィンドウの下部のタブ。  表示するには、 <b>「表示」</b> > <b>「エラー」</b> を選択します。
<b>ActiveScreen</b>	なし	記録セッション中に特定のステップを実行したときのアプリケーションの状態を示すスナップショットが表示されます。  この表示枠は API テストでは使用されません。	UFT ウィンドウの下部のタブ。  表示方法： <b>「表示」</b> > <b>「ActiveScreen」</b> を選択します。
<b>「デバッグ」表示枠</b>		テストのデバッグを支援します。 次の複数のデバッグ表示枠があります。  <ul style="list-style-type: none"> <li>・ <b>ブレークポイント</b></li> <li>・ <b>コール・スタック</b></li> <li>・ <b>ローカル変数</b></li> <li>・ <b>コンソール</b></li> <li>・ <b>ウォッチ</b></li> <li>・ <b>スレッド</b>（API テストのみ）</li> <li>・ <b>ロードされたモジュール</b>（API テストのみ）</li> </ul>	UFT ウィンドウの下部のタブ。
<b>タスク</b>	なし	現在のテストに定義されたタスクが表示され、それらのタスクを管理できます。この表示枠には、テストのアクション、関数ライブラリ、またはユーザ・コード・ファイルの TODO コメント・ステップも表示されます。	UFT ウィンドウの下部のタブ。  表示するには、 <b>「表示」</b> > <b>「タスク」</b> を選択します。
<b>検索結果</b>	なし	<b>「検索」</b> ダイアログ・ボックスや検索メニュー項目で定義する検索条件に一致した箇所がすべて表示されます。	UFT ウィンドウの下部のタブ。  表示方法： <ul style="list-style-type: none"> <li>・ <b>「表示」</b> &gt; <b>「検</b></li> </ul>

			<b>索結果</b> を選択します。 <ul style="list-style-type: none"><li>検索を実行します。</li></ul>
<b>ブックマーク</b>	なし	テスト・ドキュメントに含まれるブックマークのリストと場所が表示されます。	UFT ウィンドウの下部のタブ。  表示するには、 <b>表示</b> > <b>ブックマーク</b> を選択します。
<b>ステップの実行結果</b>	なし	個々の API テスト・ステップのテスト実行の実行結果が表示されます。  この表示枠は GUI テストでは使用されません。	UFT ウィンドウの下部のタブ。  表示方法： <ul style="list-style-type: none"><li><b>表示</b> &gt; <b>実行結果</b>を選択します。</li><li>ステップを実行するには、API テスト・ステップを右クリックし、<b>ステップの実行</b>を選択します。</li></ul>

# 第2部: アプリケーションの分析とテストの作成

「Unified Functional Testing の概要」(11ページ)では、自動化されたテストと UFT の概要について説明しました。

このレッスンでは、アプリケーションを分析してテストが必要な部分を確認します。

本項には、次の項目が含まれています。

- テスト対象のアプリケーションについての考察 ..... 22
- フライト予約アプリケーションの確認 ..... 23
- すべてのテスト用のソリューションの作成 ..... 25

# テスト対象のアプリケーションについての考察

アプリケーションのテストの作成を開始する前に、アプリケーションで何をテストするかを正確に特定する必要があります。これを行うには、アプリケーション・プロセスの点からアプリケーションを分析する必要があります。アプリケーション・プロセスとは、特定のタスクを完了するためにアプリケーションで実行される個々のアクティビティのことです。

このチュートリアルでは、フライト予約アプリケーションのテストを行います。このアプリケーションは、フライト情報と航空券予約のサービスをエミュレートするものです。

このアプリケーションは、次の2つのコンポーネントで構成されます。

- **Book Flights 層**：この層はアプリケーションのユーザ・インタフェースで、フライトの検索と予約を行うためのインタフェースを提供します。
- **Flights API 層**：このアプリケーションは、フライト予約アプリケーションのサービス (API) 側を提供します。サービスのメソッドおよび操作の詳細については、Flights API ウィンドウの **[Open Help Page]** ボタンをクリックします。

Book Flights 層を使用して、アプリケーションの GUI テストを作成します。Flights API 層を使用して、アプリケーションの API テストを作成します。

フライト予約アプリケーションのテストを計画する際には、次の確認事項を考慮します。

テストする必要があるビジネス・プロセスは何か。	これは基本的な確認事項です。次の点を考慮してください。 <ul style="list-style-type: none"><li>• アプリケーションで実行が想定されるプロセスは何か。</li><li>• 実行が想定されるプロセスに基づくと、プロセスを完了するためにユーザが行うアクションは何か。</li></ul>
アプリケーションはどのように整理されているか。	<ul style="list-style-type: none"><li>• 各ユーザ・アクティビティに対してアプリケーションのセクション/ページ/モジュールが個別に存在するか。</li><li>• ユーザが実行できるアクティビティは何か。</li><li>• これらのセクション/ページ/モジュールはアプリケーション内のどこに存在するか。</li><li>• これらのユーザ・アクティビティの期待される結果は何か。</li><li>• これらのユーザ・アクティビティをサポートするバックグラウンドのプロセスは何か。</li></ul>
アプリケーションの各セクション/ページ/モジュールでテストする必要があるユーザ・インタフェース要素は何か。	<ul style="list-style-type: none"><li>• 各領域でテストする必要があるユーザ・インタフェース・オブジェクトは何か。</li><li>• テストでシミュレートする必要があるユーザ・アクションは何か。</li></ul>
複数のシナリオで使われる可能性のあるアクティビティは何か。	繰り返し実行されるユーザ・アクションまたはアプリケーション・プロセスは何か。たとえば、アプリケーションへのログインはユーザが繰り返し行うアクションである可能性があります。また、ユーザ資格情報データベースへの接続はアプリケーションで繰り返し実行され

か。	るアクティビティです。
テスト用にサポート する必要がある開発 環境は何か。	UFT には、さまざまなテスト環境をサポートするアドインが用意されています。アプリケーションに適したアドインをロードするためには、アプリケーションの開発に使用されたテクノロジーを考慮して、そのテクノロジーが UFT でサポートされるようにする必要があります。また、UFT を開いてテストを作成するときにも、これらのアドインをロードする必要があります。これにより、テストを作成および実行する際に、UFT でアプリケーション内のオブジェクトを適切に認識することができます。

## フライト予約アプリケーションの確認

テストの作成を始める前に、サンプル・アプリケーションを参照して、アプリケーションに含まれるユーザ・アクションとアプリケーション・フローをサポートしているプロセスを確認する必要があります。

アプリケーションを操作しながら、[前のレッスン](#)で示した確認事項について考慮します。これらの確認事項を使用して、どのようにすればアプリケーションからテストを作成できるかを考えます。

### 1. フライト予約アプリケーションを開きます。

次の手順で、Book Flights (GUI) 層と Flights API (サービス) 層の両方を開きます。

- Book Flights 層 : [スタート] > [すべてのプログラム] > [HP Software] > [HP Unified Functional Testing] > [Sample Applications] > [Flight GUI]
- Flights API 層 : [スタート] > [すべてのプログラム] > [HP Software] > [HP Unified Functional Testing] > [Sample Applications] > [Flight API]

Windows 8 および Windows Server 2012 での UFT および UFT のツールとファイルへのアクセス方法の詳細については、[「Windows 8.X 以降のオペレーティングシステムでの UFT へのアクセス」\(243 ページ\)](#)を参照してください。

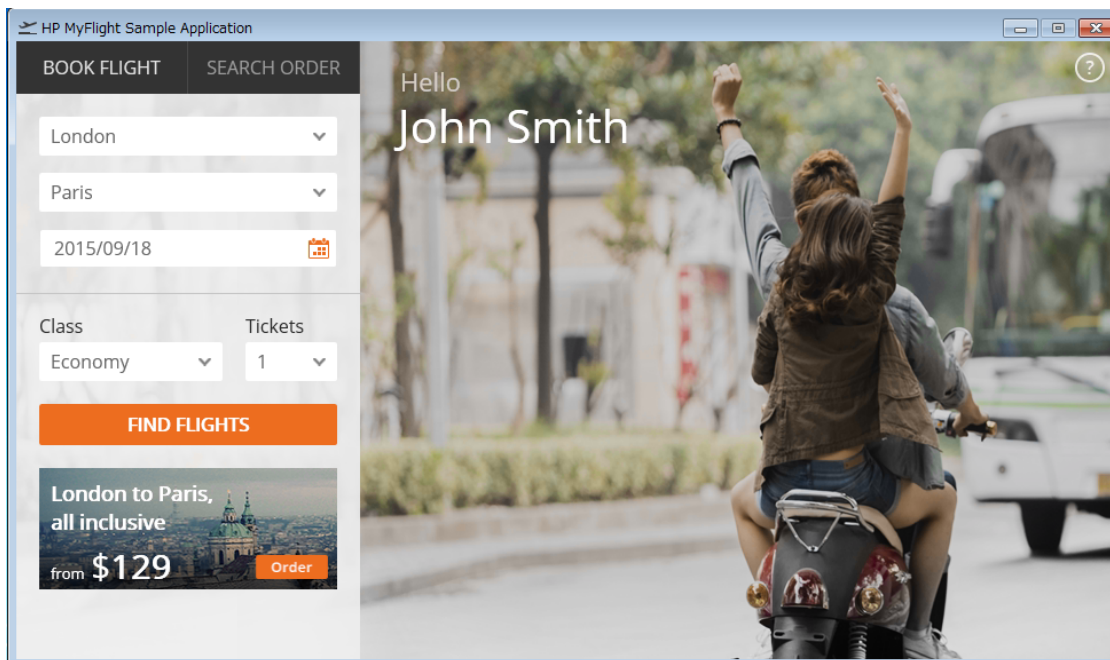
### 2. Book Flights アプリケーションにログインします。

Book Flights アプリケーションの開始ページで、ユーザ名に「John」、パスワードに「hp」と入力します。

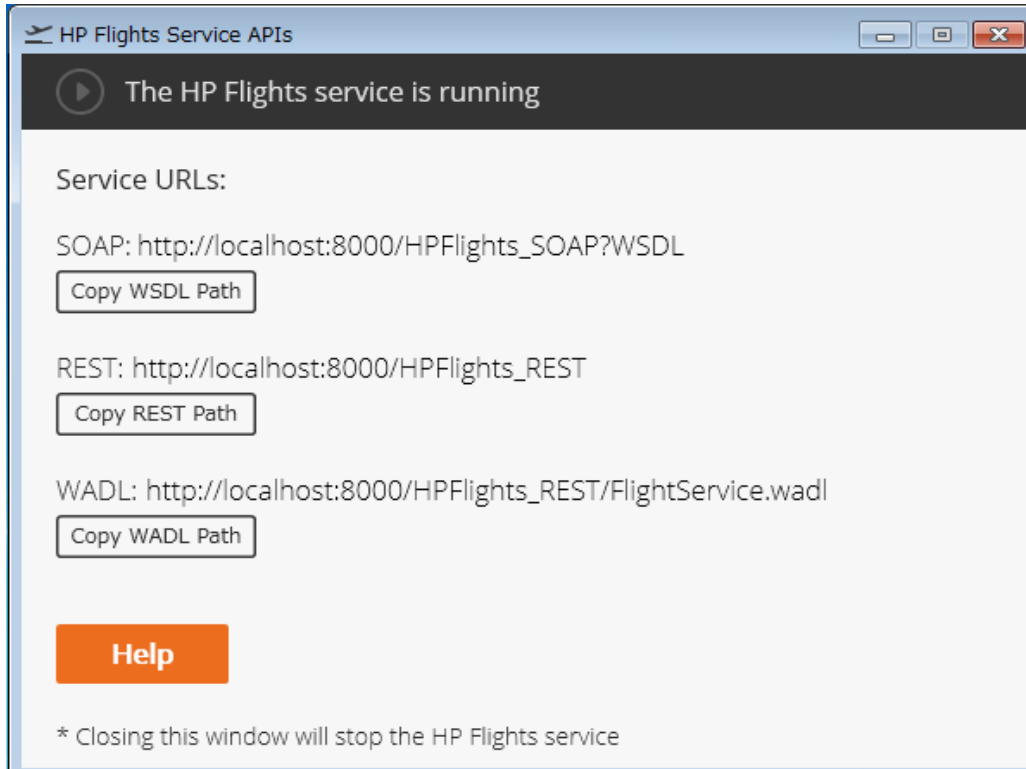
**注:** Flights API 層にはログイン情報は必要ありません。

正常にログインすると、アプリケーション層の表示は次のようになります。

### Book Flights (GUI 層)



### Flights API 層





### 3. アプリケーション層を確認します。

<b>Book Flights 層の場合...</b>	各ページで必要な情報の入力または選択を行い、予約プロセスをたどります。 アプリケーションを操作しながら、テストが必要になりそうなユーザ・アクションと、テストを構成するのに必要なオブジェクトについて考えます。
<b>Flights API 層の場合...</b>	【Help】ボタンをクリックして、アプリケーションに含まれるメソッドの説明を参照します。 アプリケーションに含まれているメソッドのリストを確認するときに、各メソッドに指定されているプロパティの詳細を書き留めます。この情報は、テストを設計する際に、これらのメソッドのプロパティ値を指定するのに使用します。

### 4. アプリケーションの参照を終了します。

- **Book Flights 層の場合**：フライト予約申し込みの詳細が【Order Details】ウィンドウに表示されたら、【New Search】をクリックしてアプリケーションの最初に戻ります。
- **Flights API 層の場合**：Flights API ウィンドウを最小化します。このウィンドウは閉じないでください。閉じると、サービスが停止します。

以上で、これらのアプリケーションを使用して、アプリケーションの各層のテストを作成する準備ができました。続いて「[すべてのテスト用のソリューションの作成](#)」(25ページ)では、このチュートリアルで使用するテスト用のソリューションを作成します。

## すべてのテスト用のソリューションの作成

UFT では、テストのコンテナとして機能するソリューションを作成できます。作成したソリューションには、テスト、ビジネス・コンポーネント、アプリケーション領域を追加できます。

ソリューションを使用すると、テストを簡単にグループ化できます。たとえば、特定のアプリケーションのすべてのテストを含むソリューションを作成したり、アプリケーションのパーツの GUI または API テストのみを含むソリューションを作成することができます。

すべてのタイプの UFT ドキュメントをソリューションに追加できます。また、テストを複数のソリューションに追加することができます。

この練習では、このチュートリアルで作成するテストを保持するためのソリューションを新規に作成します。

## 1. UFT を起動します。

次の手順のいずれかを実行します。


<b>UFT がまだ開いていない場合：</b>	<p>a. UFT を開きます。</p> <ul style="list-style-type: none"><li>デスクトップ上または <b>【スタート】</b> 画面（Windows 8.x または Windows Server 2012 の場合）の UFT アイコンをダブルクリックします。</li><li>Windows 7 または Windows Server 2008 R2 の場合は、 <b>【スタート】</b> &gt; <b>【すべてのプログラム】</b> &gt; <b>【HP Software】</b> &gt; <b>【HP Unified Functional Testing】</b> を選択します。</li></ul> <p>b. アドイン・マネージャで、 <b>WPF Add-in のみ</b> が選択されていることを確認します。他のアドインはすべてクリアします。</p> <p>c. <b>【OK】</b> をクリックしてアドイン・マネージャを閉じ、UFT を開きます。</p> <p>UFT のスプラッシュ画面が表示され、その間に、選択したアドインが UFT によってロードされます。</p>
<b>UFT が開いている場合：</b>	<p>a. <b>【ヘルプ】</b> &gt; <b>【HP Unified Functional Testing バージョン情報】</b> を選択し、ロードされているアドインを確認します。バージョン情報ダイアログ・ボックスで、ロードされているアドイン名の横にはチェックボックスが表示されます。</p> <p>b. WPF Add-in がロードされていない場合は、UFT を終了して再起動する必要があります。アドイン・マネージャが再表示されたら、WPF を選択します。</p> <p>UFT の開始時にアドイン・マネージャが開かない場合は、UFT が開いてから、 <b>【ツール】</b> &gt; <b>【オプション】</b> を選択します。続いて、 <b>【起動オプション】</b> 表示枠（ <b>【ツール】</b> &gt; <b>【オプション】</b> &gt; <b>【一般】</b> タブ &gt; <b>【起動オプション】</b> ノード）で、 <b>【起動時にアドイン マネージャを表示する】</b> オプションを選択します。</p>

## 2. 開始ページを探索します。

開始ページには、最近使ったファイル、UFT の新機能の情報、役に立つサポートやコミュニティ・フォーラムなどのリンクが用意されています。ページの右上角で、UFT の使用中に開始ページを表示する、または閉じるオプションを定義できます。


**注：** 開始ページが表示されていない場合は、 **【表示】** > **【開始ページ】** を選択して開始ページを表示できます。

## 3. 新しいソリューションを作成します。

- a. ツールバーで **【新規作成】** ドロップダウン矢印  をクリックし、 **【新規ソリューション】** を選択します。
- b. ソリューションの詳細を入力します。
- 名前：** Flight Reservation Application
  - 場所：** 標準設定では、すべてのソリューションとテストは **C:\%HOMEPATH%\My Documents\Unified Functional Testing** に保存されます。このチュートリアルでは、このパスを変更する必要はありません。

c. **【作成】** をクリックします。

【ソリューション エクスプローラ】 表示枠で、表示枠の上部にソリューション名が表示されているのを確認できます。このソリューションに追加するテストは、このソリューションのサブノードとして表示されます。

**注:** ソリューション・エクスプローラが表示されていない場合は、【ソリューション エクスプローラ】 ボタン  をクリックするか、**【表示】 > 【ソリューション エクスプローラ】** を選択して表示します。

続いて、GUI テストの作成（「[GUI 自動テストの作成と実行](#)」(28ページ)を参照）または API テストの作成（「[API 自動テストの作成と実行](#)」(148ページ)を参照）を開始できます。

## 第3部: GUI 自動テストの作成と実行

アプリケーションを分析してテスト目標を計画したら、アプリケーションのテストを作成します。この作業で大きな割合を占めるのがアプリケーションのユーザ・インタフェース (GUI) のテストの作成です。GUI テストを作成することで、アプリケーション内のコントロールやオブジェクトが設計されたとおりに動作することを確認できます。

GUI テストの作成には、次の個別のプロセスが含まれます。

- アプリケーション内のオブジェクトに対するテスト・オブジェクトを格納するオブジェクト・リポジトリの作成
- 関数ライブラリ内でのテストで使用する追加関数の作成
- アプリケーションでのユーザ・アクションを表すステップのテストへの追加
- アプリケーション内の特定のオブジェクトを検証するチェックポイントの作成
- アプリケーションがさまざまな入力値にどのように反応するかを確認するためのテスト・オブジェクトの値のパラメータ化
- テストの実行とテスト実行の結果の分析

以下のレッスンでは、これらのプロセスについて詳しく説明します。

本項には、次の項目が含まれています。

• レッスン 1 : GUI テストとアクションの作成 .....	29
• レッスン 2 : オブジェクト・リポジトリの作成 .....	33
• レッスン 3 : テストへのステップの追加 .....	46
• レッスン 4 : GUI テストの実行と分析 .....	73
• レッスン 5 : ステップとオブジェクトのパラメータ化 .....	79
• レッスン 6 : チェックポイントおよび出力値の作成 .....	94
• レッスン 7 : 関数と関数ライブラリの作成 .....	129
• レッスン 8 : テストでの Insight の使用 .....	139

# レッスン 1：GUI テストとアクションの作成

アプリケーションの GUI をテストするステップを作成する前に、最初にテストおよびテストを構成するアクションを作成する必要があります。

各 UFT GUI テストはアクションの呼び出しで構成されます。アクションは、テストを論理的なセクションに分割する（テスト内の）構成単位です。テストを複数のアクションに分割することにより、モジュール性が高く理解しやすい効率的なテストを設計できます。


テストには複数の異なるタイプのアクションを含めることができます。

<b>内部アクションと外部アクション</b>	<ul style="list-style-type: none"><li>• <b>内部アクション</b>は、ローカル・テスト（ソース・テストともいう）に格納されるアクションです。</li><li>• <b>外部アクション</b>は、別のテストに格納されているアクションに対する参照呼び出しです。</li></ul> <div><b>注:</b> テストによって呼び出される外部アクションは、ソリューション・エクスプローラでテスト・ノードの下に別のノードとして表示されます。</div> <p>たとえば、複数のテストで使用するアクションがある場合、このアクションをあるテストに内部アクションとして格納し、ほかのテストにはそのアクションへの呼び出しを挿入します。このアクションを呼び出す他のテストでは、このアクションを外部アクションとして利用できます。</p>
<b>再利用可能アクション</b>	<p>新しいアクションへの呼び出しを挿入する場合、そのアクションは標準で<b>再利用可能</b>となるため、任意のテストからこのアクションを呼び出すことができます。</p> <p>再利用可能アクションを使用する場合、元のテストに格納されている既存のアクションを更新するだけで済みます。元のアクションを変更すると、そのアクションを呼び出すすべてのテストが更新されます。再利用可能アクションを呼び出すテストでは、これらの再利用可能アクションは読み取り専用です。</p> <p>再利用可能アクションは、次の場合に役立ちます。</p> <ul style="list-style-type: none"><li>• 複数のテストに含める可能性があるプロセス（たとえば、アプリケーションへのログイン・プロセス）がある場合。</li><li>• たとえば、セキュリティが確保されたアプリケーション部分にユーザがアクセスしようとするたびに開くユーザ資格情報ダイアログ・ボックスへの入力のように、同じテストに複数回挿入することが必要なプロセスがある場合。</li></ul> <p>アクションが他のテストで利用されないようにする場合は、アクションを<b>再利用不可能</b>にできます。</p> <div><b>ヒント:</b> 多くのテストで同じアクションまたは類似したアクションを使用する場合は、再利用可能なアクションを格納するためにテスト・リポジトリの作成を検討してください。その後、他のテストにこれらの既存のアクションへの呼び出しを挿入できます。</div>
<b>アクションのコピー</b>	<p>アクション・ステップを変更する場合に、テストにアクションのコピーを挿入することもできます。アクションをコピーすると、そのアクションはコピー先のテストの<b>内部</b>アクションになります。これらのコピーはソース・テストにリンクしていないため、元のアクションを変更してもコピーされたアクションは更新されません。</p>

アクションをテストに追加するには、次の方法を使用します。

- **新規アクションの呼び出し** : 新しい空のアクションがテストに挿入されます。このアクションはテストの内部アクションになります。
- **アクションのコピーの呼び出し** : 既存のアクションのコピーがテストに挿入されます。
- **既存アクションの呼び出し** : 既存のアクションの呼び出しが挿入されます（現在のテストに対する外部アクション）。

## 1. 新規テストを作成し、ソリューションに追加します。

- a. UFT のツールバーで、**[追加]** ドロップダウン矢印  をクリックし、**[新規テストの追加]** を選択します。
- b. **[新しいテストをソリューションに追加]** ダイアログ・ボックスで、**[GUI テスト]** を選択します。
- c. テストに関する次の詳細情報を入力します。
  - **名前** : **Book Flights**
  - **場所** : 標準設定では、ドキュメントは **C:\%HOMEPATH%\My Documents\Unified Functional Testing** に保存されます。このレッスンでは、このパスを変更する必要はありません。
- d. **[追加]** をクリックします。

キャンバスに空のテストが開き、(**Book Flights** という名前の) テスト・フローのタブが表示され、(**Action 1** という名前の) 空のアクション・タブがドキュメント表示枠に開きます。

このテストは、[ソリューション エクスプローラ] 表示枠に [Flight Reservation Application] ソリューション・ノードのサブノードとしても表示されます。


## 2. Action 1 の名前をわかりやすい論理名に変更します。

- a. キャンバスで **Action1** ボックスを右クリックし、**[アクションのプロパティ]** を選択します。
- b. **[アクションのプロパティ]** ダイアログ・ボックスの **[一般]** タブで、**Action1** を **Login** に変更し、**[OK]** をクリックします。
- c. 警告が表示されたら、**[はい]** をクリックします。  
キャンバスに表示されるアクションのブロックの名前が **Login** になっているはずです。

## 3. Flight Finder という名前の新しいアクションを作成します。


- a. キャンバスの任意の場所を右クリックし、**[新規アクションの呼び出し]** を選択します。

【新規アクションの呼び出しの挿入】ダイアログ・ボックスが開きます。



- b. 【名前】ボックスに、アクション名として「**Flight Finder**」を入力します。
  - c. 【再利用可能アクション】オプションと【テストの末尾】オプションが選択されていることを確認します。
  - d. 【OK】をクリックします。
- 追加のブロックが **Flight Finder** という名前でキャンバスに追加されます。

#### 4. 追加アクションをテストに追加します。


- a. UFT ツールバーで、【新規アクションの呼び出しの挿入】ボタン  をクリックします。
  - b. 【新規アクションの呼び出しの挿入】ダイアログ・ボックスで、【名前】ボックスに「**Select Flight**」と入力します。
  - c. 【再利用可能アクション】オプションと【テストの末尾】オプションが選択されていることを確認します。
  - d. 【OK】をクリックしてアクションをテストに追加します。
- 別のブロックが **Select Flight** という名前でキャンバスに追加されます。

#### 5. 最後のアクションを作成します。

上記で使いたいいずれかの方法を使用して、**Flight Confirmation** という名前の別のアクションをテストに追加します。

これで、アプリケーションのテストに必要なすべてのアクションが揃いました。

## 6. テストを保存します。


UFT ツールバーで **保存** ボタン  をクリックします。

**保存** をクリックする前に、ドキュメント表示枠の **Book Flights** タブとソリューション・エクスプローラの **Book Flights** ノードにアスタリスク (\*) が表示されていたことに気付いたかもしれません。これらのアスタリスクは、ドキュメントに未保存の内容があることを示します。テストを保存すると、アクション・タブのすべての変更も保存されます。

## 7. キャンバス上のテスト・フローを確認します。

キャンバスの **Book Flights** タブをクリックして、テスト・フロー全体を表示します。

- テスト・フローは、マウスまたはキャンバス上にあるズーム・バーを使用して拡大/縮小表示できます。
- テスト・フローを最大に拡大表示し、キャンバスに入り切らなくなった場合は、キャンバスの左下隅にあるミニマップを使用して、テスト・フローを操作できます。

ミニマップが表示されていない場合は、キャンバスの上の **ミニマップ** ボタン  をクリックして表示します。ミニマップは、テストのアクションが多い場合や、アクションがネストされている場合に役立ちます。

テスト構造ができたなら、テストのオブジェクト・リポジトリの作成を開始できます。続いて **「レッスン 2 : オブジェクト・リポジトリの作成」 (33 ページ)** に進みます。



## レッスン 2：オブジェクト・リポジトリの作成

GUI テストは、アプリケーションのユーザ・インタフェースをテストするのに使用するテスト・オブジェクトのコレクションに基づいています。これらのテスト・オブジェクトは UFT によって学習され、テストに関連付けられたオブジェクト・リポジトリ内に格納されます。

（アクションを作成して）テストとテスト構造の作成が済んだら、テスト内で使用するテスト・オブジェクトを作成する必要があります。このレッスンでは、テスト内で使用するテスト・オブジェクト、実行環境オブジェクト、オブジェクト・リポジトリの基本概念について説明します。

このレッスンには、次の手順が含まれています。

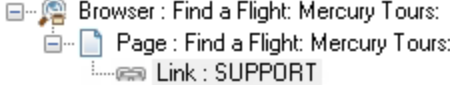
- [UFT テスト・オブジェクトの認識 - 概要](#) ..... 34
- [練習 2a：アプリケーションからのオブジェクトの追加](#) ..... 35
- [練習 2b：\[ナビゲートして学習\] を使用したオブジェクト・リポジトリの作成](#) ..... 42

## UFT テスト・オブジェクトの認識 - 概要

GUI テストの作成と実行を行う際に、UFT は**テスト・オブジェクト**を使用してアプリケーション内のオブジェクトを認識し、アプリケーションのオブジェクトに基づいたテスト・ステップを作成します。これらのテスト・オブジェクトは、UFT の**テスト・オブジェクト・モデル**に基づいています。

テスト・オブジェクト・モデルとは、UFT がアプリケーション内のオブジェクトを表すために使用する、オブジェクト・タイプまたはクラスの大規模なセットです。各テスト・オブジェクト・クラスには、UFT がオブジェクトについて学習できる認識プロパティのリスト、当該クラスのオブジェクトを一意に認識できるプロパティのサブセット、UFT がオブジェクトに実行できる関連操作のセットがあります。

テストを設計および実行する場合、オブジェクトには次の2つの種類があります。

<b>テスト・オブジェクト</b>	<p>テスト・オブジェクトは、アプリケーション内の実際のオブジェクトを表現するために UFT で作成される格納可能な表現形式です。UFT は、アプリケーション内のオブジェクトの特定のプロパティと値のセットを学習してテスト・オブジェクトを作成します。UFT は実行セッション中にオブジェクトの識別や検査に役立つオブジェクトに関する情報を保管し、その情報を使用して実行セッション中にアプリケーション・オブジェクトを認識します。</p> <p>各テスト・オブジェクトは、より大きな1つのテスト・オブジェクト階層に含まれています。たとえば、<b>Link</b> オブジェクトは、(Web) <b>Browser</b> オブジェクト内にある <b>Page</b> オブジェクト</p>  <p>に含まれています。</p> <p><b>Browser</b> オブジェクトのような最上位レベル・オブジェクトは、<b>Page</b> オブジェクトや <b>Frame</b> オブジェクトのような下位レベルのオブジェクトを含むことができるため、コンテナ・オブジェクトと呼ばれます。</p>
<b>実行環境オブジェクト</b>	<p>実行環境オブジェクトは、実行セッション中にアクション（メソッド）の実行対象となるアプリケーションに含まれる実際のオブジェクトです。UFT は実行環境オブジェクトのプロパティを学習し、テスト・オブジェクトに変換します。</p>

UFT はアプリケーション内のオブジェクトを学習し、対応するテスト・オブジェクトを**オブジェクト・リポジトリ**に追加します。このオブジェクト・リポジトリは、テスト・オブジェクトの保管庫として機能します。UFT はテストを実行する際に、テストのオブジェクト・リポジトリでテスト・ステップに含まれるオブジェクトを探します。

オブジェクトをオブジェクト・リポジトリに追加すると、UFT では次の処理が行われます。

- ・アプリケーション内の学習したオブジェクトを表す UFT テスト・オブジェクト・クラスが認識され、適切なテスト・オブジェクトが作成される。
- ・アプリケーションに含まれているオブジェクトのプロパティの現在値が読み取られ、認識プロパティのリストと値がテスト・オブジェクトと一緒に保管される。
- ・テスト・オブジェクトに一意の名を付ける。

オブジェクト・リポジトリには、次の2つの種類があります。

<b>共有オブジェクト</b>	共有オブジェクト・リポジトリは、個別のテストとは無関係に存在するオブジェクト・リポジトリ
-----------------	--

ト・リポジトリ	トリです。共有オブジェクト・リポジトリ内のテスト・オブジェクトは、複数のテスト/アクションで使用できます。テスト・オブジェクトを更新すると、その更新内容が共有オブジェクト・リポジトリを使用するすべてのテストに適用されるため、テスト・オブジェクトの保管および保守には、このタイプのオブジェクト・リポジトリをお勧めします。
ローカル・オブジェクト・リポジトリ	ローカル・オブジェクト・リポジトリには、特定のアクションのコンテキストで使用されるテスト・オブジェクトが含まれます。このタイプのオブジェクト・リポジトリは、別のアクションでは使用できません。標準設定では、すべてのアクションにローカル・オブジェクト・リポジトリが存在します。

オブジェクト・リポジトリを作成するときは、テストに必要なテスト・オブジェクトのみを含めることをお勧めします。これによりオブジェクト・リポジトリが比較的小さく保たれ、保守やオブジェクトの選択が簡単になります。また、テストの作成または変更時にほかの人が正しいオブジェクトを簡単に選択できるように、オブジェクトには必ず論理名をつけてください。

オブジェクト・リポジトリには、チェックポイントおよび出力オブジェクトを含めることもできます。チェックポイント・オブジェクト・タイプについては、「[レッスン5：ステップとオブジェクトのパラメータ化](#)」(79ページ)を参照してください。

## 練習 2a：アプリケーションからのオブジェクトの追加

「[UFT テスト・オブジェクトの認識 - 概要](#)」(34ページ)では、UFT のテスト・オブジェクト・モデルと UFT でオブジェクトを学習して保管する方法について学習しました。

この練習では、UFT のオブジェクト認識機能を使用したオブジェクトの学習と、オブジェクト・リポジトリの作成を行います。

### 1. UFT を起動し、Book Flights テストを開きます。

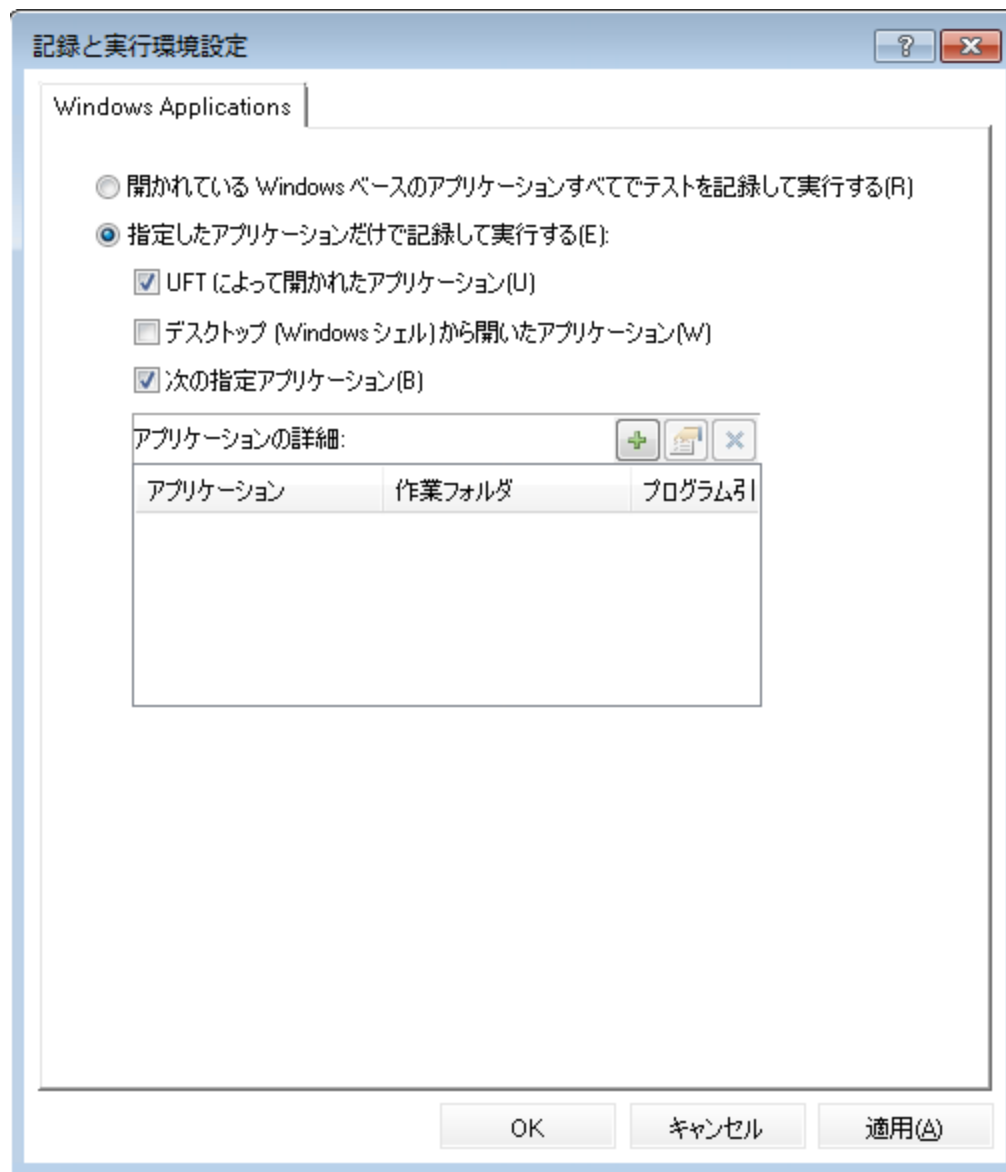
- UFT が開いていない場合は、「[すべてのテスト用のソリューションの作成](#)」(25ページ)の手順に沿って UFT を開きます。
- 開始ページの「[最近のテスト/コンポーネント](#)」領域で、「Book Flights」をクリックします。


Book Flights テストが開き、「[レッスン1：GUI テストとアクションの作成](#)」(29ページ)で作成した Book Flights テスト（とそのアクション）が表示されます。

### 2. UFT の学習設定をセットします。

- 「[記録](#)」 > 「[記録と実行環境設定](#)」を選択します。「記録と実行環境設定」ダイアログ・ボックスが開きます。
- 「記録と実行環境設定」ダイアログ・ボックスで、「Windows Applications」タブを選択します。
- 「Windows Applications」タブで、「[指定したアプリケーションだけで記録して実行する](#)」オプションを選択します。

- d. **【指定したアプリケーションだけで記録して実行する】** オプションの下に**【UFT によって開かれたアプリケーション】** オプションと**【次の指定アプリケーション】** オプションのチェックボックスを選択します。



- e. **【アプリケーションの詳細】** 領域で、**【追加】** ボタン  をクリックします。
- f. **【アプリケーションの詳細】** ダイアログ・ボックスに、次のようにアプリケーションのパスと作業フォルダを入力します。
- **アプリケーション**: <UFT installation folder>\samples\Flights Application\FlightsGUI.exe
  - **作業フォルダ**: <UFT installation folder>\samples\Flights Application
- g. **【アプリケーションを起動する】** オプションを選択し、**【OK】** をクリックします。


- h. メインの [記録と実行環境設定] ダイアログ・ボックスで, [OK] をクリックします。その後アプリケーションに対してステップの記録やテストの実行を行う際に, UFT とアプリケーションが連動できるようになります。

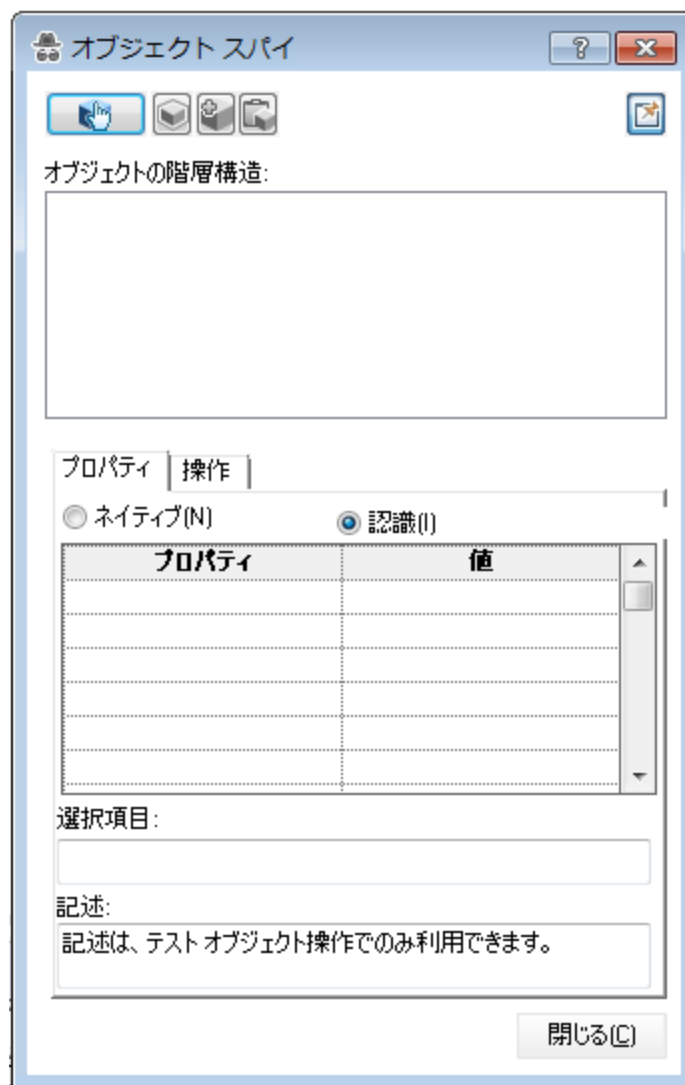
### 3. フライト予約アプリケーションを開きます。



まだ開いていない場合は, フライト予約アプリケーションを開きます。

**注:** フライト予約アプリケーションを開くタイミングは, UFT を開く前でも後でも構いません。

### 4. フライト予約アプリケーションの一部のオブジェクトで利用できるプロパティと操作を表示します。

- a. ツールバーで, [オブジェクト スパイ] ボタン  をクリックします。[オブジェクト スパイ] ダイアログ・ボックスが開きます。



- b. 「オブジェクト スパイ」ダイアログ・ボックスをアプリケーションの脇にドラッグします。こうすることで、調査の対象となるアプリケーションのオブジェクトをはっきりと見ることができます。
- c. 「スパイの実行中にオブジェクト スパイを常に手前に表示」トグル・ボタン  が押されていることを確認します。
- d. 指差しボタン  をクリックします。

指差しボタンを押すと、UFT が非表示になり、「オブジェクト スパイ」ダイアログ・ボックスがフライト予約アプリケーション上に表示されます。


**ヒント:** フライト予約アプリケーション、UFT、開いているほかのウィンドウを交互に切り替える必要がある場合は、**CTRL** を押して、指差しアイコンを通常の Windows ポインタに戻します。Windows ポインタが必要な間は **CTRL** ボタンを押したままにし、指差

しアイコンを使用する準備ができたなら離します。

- e. マウスのカーソルをページのさまざまなオブジェクトの上に移動し、[オブジェクト スパイ] ダイアログ・ボックスで何が起こるかを確認します。

**注:** UFT がオブジェクトの場所を正しく認識できない場合、アプリケーションまたはページが100%で表示され、現在のビューの拡大または縮小が行われていないことを確認してください。

たとえば、90% または 120% の倍率でページを表示すると、オブジェクトの右側または左側の部分をクリックまたは選択しないとオブジェクトが認識されない場合があります。

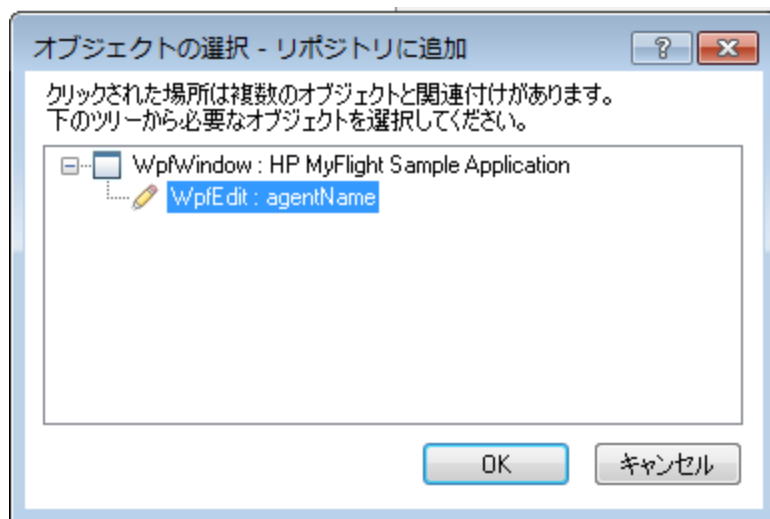
- f. [Username] エディット・ボックス内をクリックします。このオブジェクトのテスト・オブジェクト階層が表示されます。階層ボックスにオブジェクトの名前が「WpfEdit」と表示されています。これはオブジェクトのクラス・タイプです。
- g. アプリケーションの [Login] ウィンドウで、[Username] エディット・ボックス内をクリックします。これにより、アプリケーション内のオブジェクトがアクティブになります。
- h. UFT の [オブジェクト スパイ] ダイアログ・ボックスで、指差しボタン  をクリックします。[オブジェクト スパイ] の [オブジェクトの階層構造] ボックスに **agentName** が表示されていることを確認します。
- i. [オブジェクト スパイ] ダイアログ・ボックスを閉じます。

## 5. テストに必要なオブジェクトをオブジェクト・リポジトリに追加します。

この手順では、「スパイ」のプロセスを進めて、UFT に対してテストに必要なオブジェクトのみを学習し、それらをオブジェクト・リポジトリに追加するように指示します。

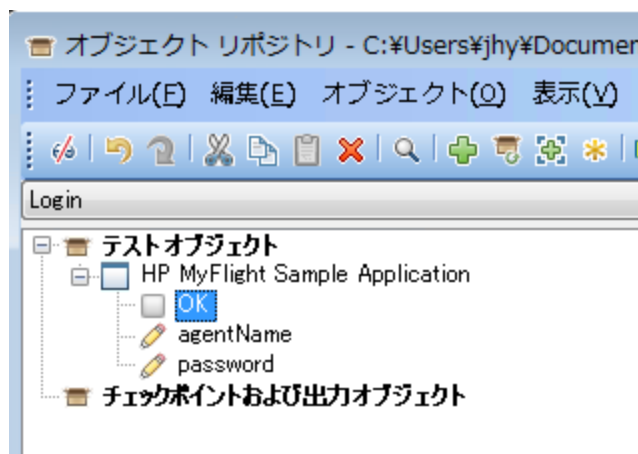
- a. アプリケーションの [Login] ウィンドウで、[Username] エディット・ボックス内をクリックします。
- b. UFT で、[リソース] > [オブジェクト リポジトリ マネージャ] を選択します。[オブジェクト リポジトリ マネージャ] ウィンドウが開きます。
- c. [オブジェクト リポジトリ マネージャ] ウィンドウで、[オブジェクト] > [オブジェクトの追加] を選択します。UFT とオブジェクト・リポジトリ・マネージャの両方が非表示になります。

- d. **[Login]** ページで, **[Username]** エディット・ボックス内を再度クリックします。[オブジェクトの選択 - リポジトリに追加] ダイアログ・ボックスが開きます。



- e. [オブジェクトの選択] ダイアログ・ボックスで, **[agentName]** オブジェクトを選択し, **[OK]** をクリックします。agentName オブジェクトが親オブジェクト (**Login** ウィンドウ・オブジェクト) とともにオブジェクト・リポジトリに追加されます。
- f. 上記の手順を繰り返し, **[Password]** エディット・ボックスのオブジェクトを追加して **[OK]** ボタンをクリックします。

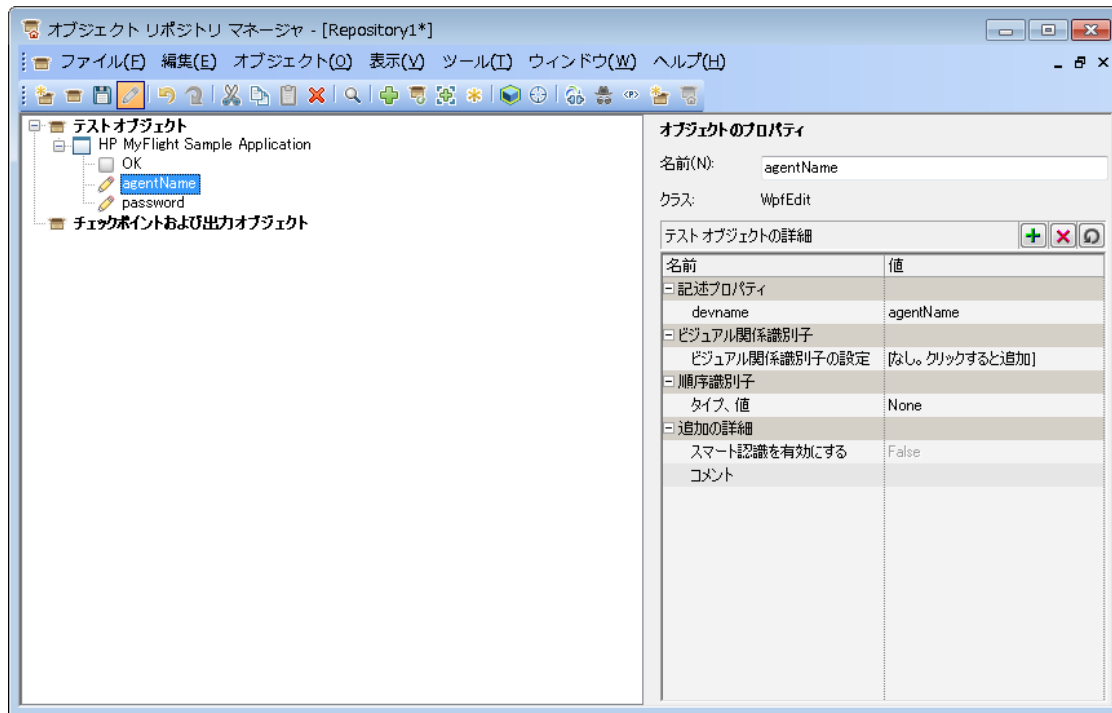
すべてのオブジェクトをオブジェクト・リポジトリに追加すると, リポジトリは次のようになります。




6. **UFT がオブジェクトについて学習した内容を確認します。**



【テスト オブジェクト】ツリーで、**agentName** オブジェクトを選択し、オブジェクト・リポジトリの右側の表示枠に表示されるオブジェクト・プロパティを確認します。これらは、UFT が実行セッション中にオブジェクトの識別に使用する記述プロパティです。



## 7. オブジェクト・リポジトリを保存します。

- 【オブジェクトリポジトリ マネージャ】ウィンドウで、【保存】を選択します。
- ソリューションとテストが保存されるフォルダ（C:\%HOMEPATH%\My Documents\Unified Functional Testing）を参照します。このフォルダ内に **Tutorial\_Object Repositories** という名前の新規フォルダを作成し、そのフォルダを開きます。
- Tutorial\_Object Repositories フォルダに、**Login** という名前でオブジェクト・リポジトリを保存し、【OK】をクリックします。

**注:** この手順では、オブジェクト・リポジトリの保存のみが行われます。テストで UFT のメイン・ウィンドウを開く場合でも、オブジェクト・リポジトリの保存時にオブジェクト・リポジトリとテストとの関連付けは行われません。

## 8. オブジェクト・リポジトリを Login アクションに関連付けます。

オブジェクト・リポジトリをアクションに関連付けることにより、関連付けたアクションの任意のステップでリポジトリの任意のオブジェクトを使用できるようになります。

**注:** 同じオブジェクト・リポジトリを複数のテストやアクションに関連付けることができます。

- a. UFT のメイン・ウィンドウを開きます。
- b. UFT のメイン・ウィンドウで、ソリューション・エクスプローラを開きます。
- c. ソリューション・エクスプローラで、[Book Flights] ノードの下に [Login] ノードを展開します。
- d. **Login** アクションを右クリックし、[リポジトリをアクションに関連付ける] を選択します。[共有オブジェクトリポジトリを開く] ダイアログ・ボックスが開きます。
- e. [共有オブジェクトリポジトリを開く] ダイアログ・ボックスで、前の手順でオブジェクト・リポジトリを保存したフォルダに移動し、**Login.tsr** ファイルを選択します。[開く] をクリックします。
- f. ダイアログ・ボックスが開くので、[はい] をクリックします。

これで、オブジェクト・リポジトリが **Login** アクションに関連付けられ、そのアクションの子要素としてソリューション・エクスプローラに表示されます。

## 9. テストを保存します。

[ファイル] > [保存] を選択してテストを保存します。

以上で UFT でのアプリケーション・オブジェクトの認識の設定、オブジェクトを格納するオブジェクト・リポジトリの作成、およびオブジェクト・リポジトリとテストとの関連付けが完了しました。続いて、アプリケーションの別の領域に対するオブジェクト・リポジトリを追加で作成できます。さらに学習するには、[「練習 2b：\[ナビゲートして学習\]を使用したオブジェクト・リポジトリの作成」\(42ページ\)](#)に進みます。

## 練習 2b：[ナビゲートして学習]を使用したオブジェクト・リポジトリの作成

[「練習 2a：アプリケーションからのオブジェクトの追加」\(35ページ\)](#)では、UFT を使用してアプリケーション内のオブジェクトを追加する方法、フライト予約アプリケーションのログイン・ページの共有オブジェクト・リポジトリを作成する方法、およびこのオブジェクト・リポジトリをテストに関連付ける方法について学習しました。

この練習では、サイトの残りの各ページ用の共有オブジェクト・リポジトリを作成します。[ナビゲートして学習]を使用することで、アプリケーションのページまたはセクション内のすべてのオブジェクトを一度に学習できます。


**ヒント:** 必ずアプリケーションまたは Web サイトのセクション/ページごとに、個別の共有オブジェクト・リポジトリを作成することをお勧めします。そうすることで、テスト・ステップを追加または変更するときや、保守作業を行うときに、正しいオブジェクトを見つけやすくなります。

この練習では、各ページのすべてのオブジェクトを学習するように UFT に指示します。この時点では、オブジェクト・リポジトリを特定のアクションに関連付ける必要はありません。

**1. フライト予約アプリケーションの [Flight Finder] ページにログインします。**


- まだ開いていない場合は、フライト予約アプリケーションを開きます。
- ログイン・ページで、次のログイン資格情報を入力します。
  - ユーザ名:** john
  - パスワード:** hp
- [OK]** をクリックします。 [Book Flight] ページが開きます。

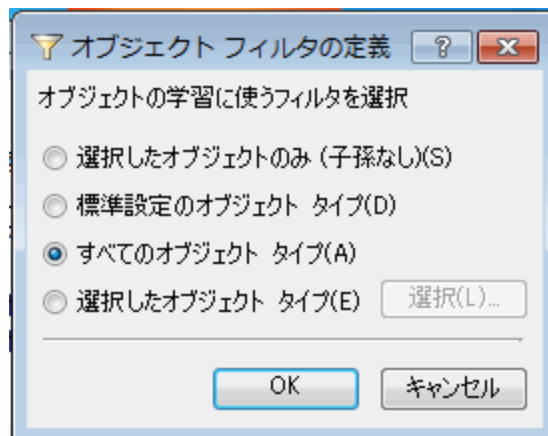
**2. 新しい共有オブジェクト・リポジトリを作成します。**

- オブジェクト・リポジトリ・マネージャが開いている場合は、UFT で [リソース] > [オブジェクト リポジトリ マネージャ] を選択して開きます。
- [オブジェクト リポジトリ マネージャ] ウィンドウで、**[新規作成]**  をクリックします。空の共有オブジェクト・リポジトリが開きます。

**3. [オブジェクト フィルタの定義] を設定して、ナビゲートして学習するプロセスを開始します。**

前の練習で学習したようにオブジェクトを個別に追加するだけでなく、[ナビゲートして学習] を使用してアプリケーション内のすべてのオブジェクトをまとめて学習して追加できます。

- [オブジェクト リポジトリ マネージャ] ウィンドウで、**[オブジェクト] > [ナビゲートして学習]** を選択します。UFT とオブジェクト・リポジトリ・マネージャの両方が非表示になります。
- [ナビゲートして学習] ツールバーの**[オブジェクト フィルタの定義]** ボタン  をクリックします。[オブジェクト フィルタの定義] ダイアログ・ボックスが開きます。
- [オブジェクト フィルタの定義] ダイアログ・ボックスで **[すべてのオブジェクト タイプ]** を選択し、**[OK]** をクリックします。

**4. 「FLIGHT FINDER」 ページのすべてのオブジェクトを学習します。**

この手順では、[Book Flight] ページでフィルタに合致するすべてのオブジェクトを学習し、それらを共有オブジェクト・リポジトリに追加するように UFT に指示します。

- a. フライト予約アプリケーションの [Book Flight] ページで、アプリケーション・タイトルバーをクリックし、UFT で学習するページとしてフォーカスを設定します。
- b. [ナビゲートして学習] ツールバーで、[学習] をクリックします。UFT がページ上のオブジェクトを新しいオブジェクト・リポジトリに追加し始めると、アプリケーションがちらつき、[オブジェクトを追加しています] メッセージ・ボックスが表示されます。

**注:** オブジェクトの追加には数秒かかります。[ナビゲートして学習] の実行中はアプリケーションを操作しないでください。

- c. [ナビゲートして学習] ツールバーを閉じます。UFT と [オブジェクト リポジトリ マネージャ] ウィンドウが再び表示されます。

## 5. 共有オブジェクト・リポジトリを保存します。

- a. [オブジェクト リポジトリ マネージャ] ウィンドウで、[保存]  を選択します。[共有オブジェクト リポジトリの保存] ダイアログ・ボックスが開きます。
- b. 前の練習で作成した **Tutorial\_Object Repositories** フォルダを参照します。
- c. このオブジェクト・リポジトリに「Flight Finder」と名前を付け、[OK] をクリックします。

## 6. 残りのアプリケーション・ページ用のオブジェクト・リポジトリを作成します。

- a. 前の手順で説明した方法で、次の各ページの共有オブジェクト・リポジトリを作成します。
  - **Select Flight**
  - **Flight Details**

**重要：** [Flight Details] ページのオブジェクトを学習する前に、必ず [Passenger Name] ボックスに文字列を入力してください。こうすることで、[Order] ボタンがアクティブになり、UFT で正しく学習することが可能になります。このボタンは他の練習で必要になります。

- b. オブジェクト・リポジトリはそれぞれ「Select Flight」および「Flight Confirmation」という名前にします。

## 7. Flight Finder オブジェクト・リポジトリを Flight Finder アクションに関連付けます。

- a. UFT ウィンドウに切り替えます。ソリューション・エクスプローラがまだ開いていない場合は、UFT ウィンドウの左下隅にある [ソリューション エクスプローラ] タブをクリックしてソリューション・エクスプローラを開きます。

- b. ソリューション・エクスプローラの [Book Flights] ノードで, [Flight Finder] ノードを右クリックし, [リポジトリをアクションに関連付ける] を選択します。[共有オブジェクトリポジトリを開く] ダイアログ・ボックスが開きます。
  - c. **Tutorial\_ObjectRepositories** フォルダを参照します。
  - d. Tutorial\_ObjectRepositories フォルダで, **Flight Finder.tsr** ファイルを選択し, [開く] をクリックします。
  - e. ダイアログ・ボックスが開くので, [はい] をクリックします。
- これで, オブジェクト・リポジトリが **Flight Finder** アクションに関連付けられ, そのアクションの子としてソリューション・エクスプローラに表示されます。


## 8. 残りのオブジェクト・リポジトリを適切なアクションに関連付けます。

次のように, オブジェクト・リポジトリをアクションに関連付けます。

アクション	オブジェクト・リポジトリ
Select Flight	Select Flight.tsr
Flight Confirmation	Flight Confirmation.tsr

以上で, 各アクションに後でステップを加える際に, 必要なすべてのテスト・オブジェクトが利用できます。

## 9. テストを保存します。

[保存]  をクリックします。

オブジェクト・リポジトリの作成とテストへの関連付けが済んだら, これらのオブジェクトを使用してテスト・ステップを作成できます。続いて「[レッスン3：テストへのステップの追加](#) (46ページ)」では, テスト・ステップの作成を開始します。

## レッスン3：テストへのステップの追加

「[レッスン1：GUIテストとアクションの作成](#)」では、フライト予約アプリケーションのテスト構造を提供するため、テストとアクションの作成を行いました。「[レッスン2：オブジェクト・リポジトリの作成](#)」では、アプリケーションのテスト・オブジェクトを含むオブジェクト・リポジトリを作成しました。

このレッスンで学習する内容は、アプリケーションのテストを作成するのに必要な最後の手順です。ここでは、テスト・ステップをGUIテストに追加する方法について学習します。これにより、ユーザ・インタフェースでユーザ・アクションの正確なテストを実行することが可能になります。

このレッスンには、次の手順が含まれています。

- [GUIテストでのテスト・ステップの追加 - 概要](#) .....47
- [練習 3a：キーワード・ビューでの Login アクションへのステップの追加](#) .....47
- [練習 3b：記録による FlightFinder アクションへのステップの追加](#) .....54
- [練習 3c：\[ツールボックス\] 表示枠を使用した Select Flight アクションへのステップの追加](#) . 58
- [練習 3d：ステップ・ジェネレータを使用した Book Flight アクションへのステップの追加](#) .....62
- [高度な練習 3e（オプション） - エディタを使用したステップの追加](#) .....67

## GUI テストでのテスト・ステップの追加 - 概要

GUI テストでテスト・ステップを作成するには、テスト・ステップ内でオブジェクトを使用し、テスト・オブジェクトに対して実行するアクションを UFT に設定する必要があります。これにより、テスト・オブジェクト・メソッド（アクション）をアプリケーション内の実行環境オブジェクトに対するアクションに変換し、アプリケーション上でアクションを再生できます。

これを支援するため、UFT にはテスト・ステップを追加するさまざまな方法が用意されています。

<b>キーワード・ビュー</b>	キーワード・ビューを使用すると、ステップのグリッドでテスト・オブジェクトを選択し、これらのテスト・オブジェクトに必要なアクション（メソッド）を追加できます。キーワード・ビューでは、オブジェクトの階層構造が必要に応じて自動的に並べ替えられます。  適切なオブジェクトとメソッドを選択すると、テスト・ステップがグリッドに表示され、オブジェクト名、オブジェクト・メソッド、追加したパラメータ、およびステップの注釈サマリが表示されます。
<b>エディタ</b>	エディタでは、オブジェクト・メソッドやパラメータとともにオブジェクト（必要に応じてオブジェクトに必要な階層構造を含む）を入力できます。アプリケーションのコード記述に慣れている場合は、エディタを使用することで容易にテスト・ステップを作成できます。
<b>記録</b>	UFT には、アプリケーション内のテスト・オブジェクトとアプリケーションのオブジェクトに対して実行するアクションを自動的に認識する記録メカニズムが組み込まれています。アプリケーションで記録を行うと、UFT でユーザのアクションがテスト・ステップに変換され、オブジェクト名とオブジェクトに対して実行するアクション（メソッド）が表示されます。これにより、ユーザが自身で行うテストを実行することが可能になり、UFT 内でテストを手動で編集することなく UFT でテストを自動的に作成できます。
<b>【ツールボックス】表示枠</b>	ドキュメント表示枠で GUI のアクション・タブを選択すると、そのアクションの関連するオブジェクトと関数が【ツールボックス】表示枠に自動的に表示されます。これらのテスト・オブジェクト（または関数）をキーワード・ビューやエディタにドラッグすると、UFT によってオブジェクトに対するステップが自動的に作成されます（ただし、【ツールボックス】表示枠からドラッグした後でオブジェクトに対するメソッドを指定する必要があります）。
<b>ステップ・ジェネレータ</b>	【ステップジェネレータ】ダイアログ・ボックスを使用すると、1つのダイアログ・ボックスでテスト・ステップのすべての詳細情報を選択または指定できます。ステップ・ジェネレータでは、選択したアクション、そのアクションのメソッド、および必要なパラメータと関連付けられている任意のテスト・オブジェクトを選択できます。この情報を選択すると、選択した場所にステップがすべての詳細情報とともに挿入されます。

以下の練習では、これらの各方法を使用してテスト・ステップを作成します。

### 練習 3a：キーワード・ビューでの Login アクションへのステップの追加

この練習では、キーワード・ビューを使用して Login アクションにステップを挿入します。


1. **UFT を起動し、Book Flights テストを開きます。**
  - a. UFT が開いていない場合は、「すべてのテスト用のソリューションの作成」(25ページ)の手順

に沿って開きます。WPF Add-in がロードされていることを確認します。

- b. 開始ページの [最近のソリューション] 領域で、[**Flight Reservation Application**] をクリックします。

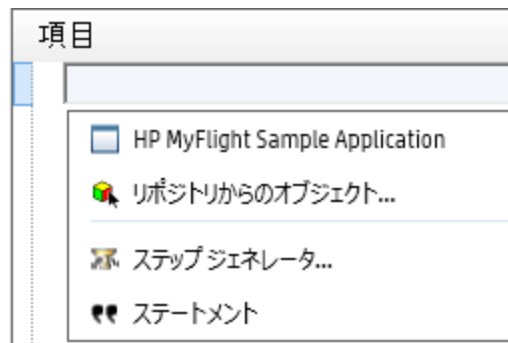
Flight Reservation Application ソリューションが開き、「[レッスン 1 : GUI テストとアクションの作成](#)」(29ページ)で作成した Book Flights テストが表示されます。

## 2. Login アクションを開いて、キーワード・ビューを表示します。

- a. テスト・フロー・キャンバスがまだ開いていない場合は、ドキュメント表示枠で [**Book Flights**] タブをクリックして表示します。
- b. キャンバスで、**Login** アクションをダブルクリックします。  
**Login** アクションがドキュメント表示枠に別のタブとして表示されます。
- c. エディタが表示されている場合は、ツールバーで [**キーワード ビューとエディタ間の切り替え**] ボタン  をクリックして、キーワード・ビューを表示します。

## 3. フライト予約アプリケーションにログインする最初のステップを追加します。

- a. キーワード・ビューの [項目] カラムで、[**新規ステップ**] ボタンをクリックします。[項目] カラムのグリッドに空の編集フィールドが現れます。
- b. [項目] カラムで、空の編集フィールド内をクリックします。[項目] ドロップダウン・リストが開き、関連付けられているオブジェクト・リポジトリの最上位の親オブジェクトが表示され、項目を選択するように求められます。

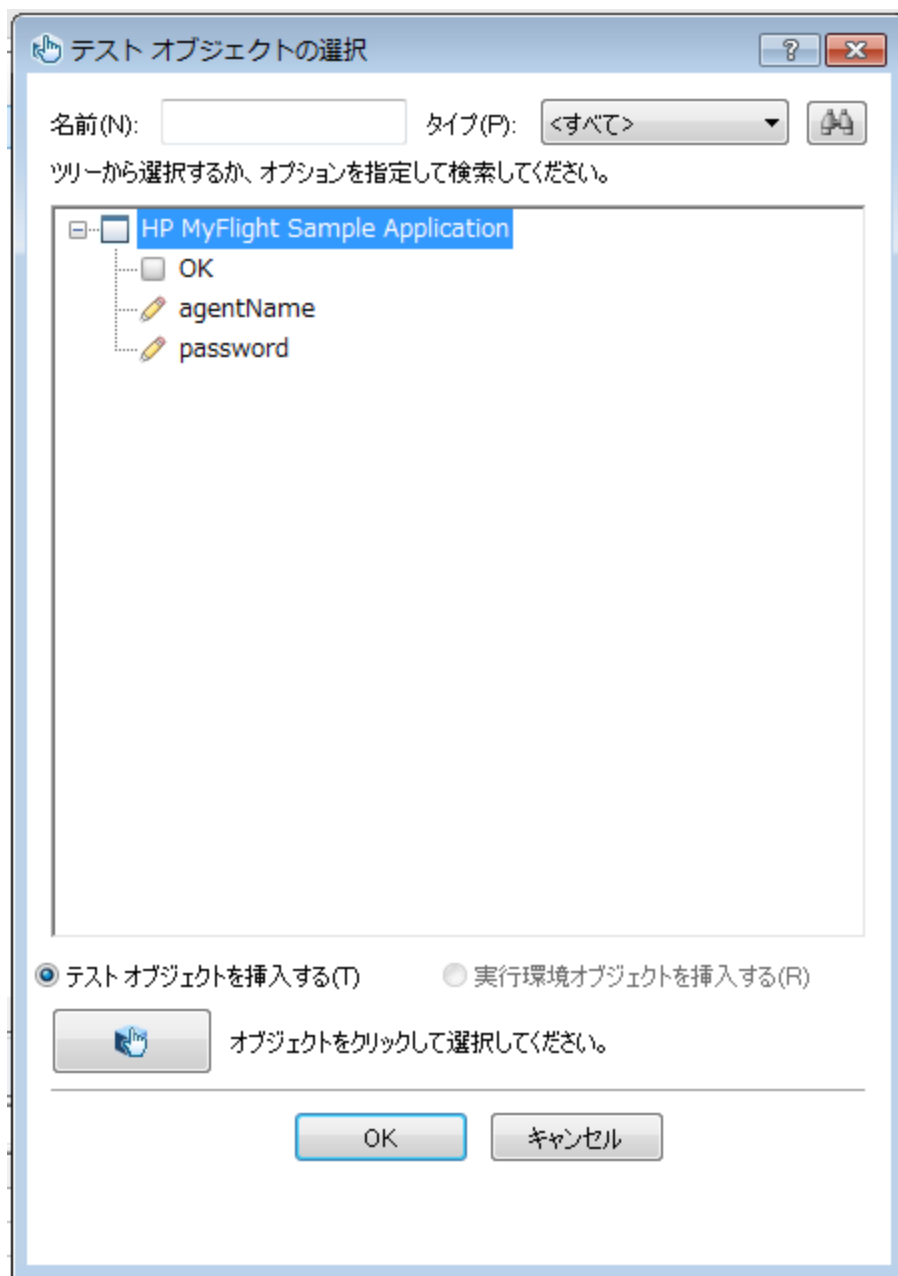


このステップで、親オブジェクトは **Login** ウィンドウです。メイン・ウィンドウに対する操作は必要ないため、このオブジェクトは選択しないでください。ステップを挿入する必要があるのは、操作を実行する対象のオブジェクトに対してのみです。

- c. [**リポジトリからのオブジェクト**] を選択して、[テストオブジェクトの選択] ダイアログ・ボックスを開きます。



- d. [テスト オブジェクトの選択] ダイアログ・ボックスで、テスト・オブジェクト・ツリーを展開します。



- e. テスト・オブジェクト・ツリーで、**agentName** を選択し、[OK] をクリックします。

[テスト オブジェクトの選択] ダイアログ・ボックスが閉じて、アクションにステップが1つ追加されます。

キーワード・ビューには3行が追加されることに注意してください。UFT 親テスト・オブジェクトごとに1行追加します（オブジェクトに対して操作が実行されない場合でも追加されます）。これら行はステップが実行されるオブジェクトへのパスの一部です。

実行セッション中、UFT は、操作を実行する必要がある実際の対象オブジェクトを親オブジェクトに基づいて識別します。

新しい3行の最後の行で表されるこのステップで：

- 選択した **agentName** WpfEdit オブジェクトが【項目】セルに追加されます。
- 標準設定のメソッド **Set** が【操作】セルに追加されます。
- テキストが【注釈】セルに追加され、このステップによってエディット・ボックスのテキストがクリアされることを示します。これは、【値】セルに必要な値がステップに欠けており、ユーザ名で更新する必要があるためです。

項目	操作	値	注釈
HP MyFlight Sample Application	Activate		"HP MyFlight Sample Application" window をアクティブにする。
agentName	Set		"agentName" edit box のテキストをクリアする。
+ 新規ステップ			

- f. 【値】セルをクリックし、「john」と入力します。文字列を入力したら、**ENTER** キーを押します。

この値を挿入するとステップが完成します。キーワード・ビューの別領域をクリックすると、このステップの注釈が【注釈】セルで更新されます。

項目	操作	値	注釈
HP MyFlight Sample Application	Activate		"HP MyFlight Sample Application" window をアクティブにする。
agentName	Set	"john"	"agentName" edit box に "john" を入力する。
+ 新規ステップ			


**ヒント:** 【値】カラムに入力した値は自動的に引用符で囲まれ、それが文字列値であることが示されます。メソッドがインデックス値をサポートしているときに、インデックス値を入力した場合は、引用符は追加されません。

- g. 【表示】 > 【エディタ】を選択してエディタを表示します（ステップの構文が VBScript で表示されます）。

```
WpfWindow("HP MyFlight Sample Application").WpfEdit("agentName").Set "john"
```

このステップは、agentName という名前の **WpfEdit**（エディット・ボックス）テスト・オブジェクトを対象に実行されます。次の点に注目してください。

- WpfEdit（エディット・ボックス）テスト・オブジェクトの前に、このテスト・オブジェクトの階層が表示されています。このステップの階層には、WpfWindow オブジェクトを含めます。
- テスト・オブジェクトの直後に、オブジェクトに対して実行されるメソッドが表示されています。このステップのメソッドは **Set** です。
- **agentName** エディット・ボックスに入力するテキストは、**Set** メソッドのすぐ後に表示されます。テキストは引用符に囲まれて表示され、その値が文字列であることが示されます。このステップで入力するテキストは「john」です。
- ステップの各部分は終止符（ピリオド）で分けられています。

- h. **【キーワード ビュー】** ボタン  をクリックして、キーワード・ビューに戻ります。

#### 4. 次のステップを追加します。

- 【項目】** カラムで、**agentName** 行のすぐ下の **【新規ステップ】** ボタンをクリックします。編集フィールドを含む行が追加されます。
- 空の編集フィールド内をクリックします。**【項目】** リストが開き、前のステップのテスト・オブジェクトの兄弟オブジェクトが一覧表示されます。
- 【項目】** リストから **password** を選択します。今回は新しい行が1つのみが追加されます。これは、このオブジェクトと前のステップで、親オブジェクトが同じためです。

このステップで：

- **password** WpfEdit テスト・オブジェクトが **【項目】** セルに追加されます。
- 標準設定のメソッド **Set** が **【操作】** セルに追加されます。パスワードは暗号化するためがあるので、このメソッドを変更します。
- テキストが **【注釈】** セルに追加され、このステップによってエディット・ボックスのテキストがクリアされることを示します。これは、**【値】** セルに必要な値がまだステップに欠けており、password で更新する必要があるためです。
- d. **【操作】** セルをクリックして下向き矢印を表示し、下向き矢印をクリックして、選択したテスト・オブジェクトで利用可能なメソッドのリストを表示します。
- e. メソッドのリストで、**SetSecure** を選択します。このメソッドで暗号化されたテキストが使用できるようになります。

#### 5. HP パスワード・エンコーダ・アプリケーションを使用して、エンコードされたパスワードを生成します。

- 【スタート】** > **【すべてのプログラム】** > **【HP Software】** > **【HP】** > **【HP Unified Functional Testing】** > **【ツール】** > **【パスワード エンコーダ】** または **＜インストール・フォルダ＞\bin\CryptonApp.exe** を選択します。
- パスワード・エンコーダで、**【パスワード】** ボックスに「hp」と入力します。
- 【生成】** をクリックします。**【パスワード エンコーダ】** によってパスワードが暗号化され、暗号化された値が **【エンコード文字列】** ボックスに表示されます。
- 【コピー】** をクリックします。
- UFT で、Login アクションに、暗号化した値を password ステップの **【値】** セルに貼り付けます。

このステップの **【注釈】** カラムに、更新された注釈が表示されます。

- 【パスワード エンコーダ】** ダイアログ・ボックスを閉じます。

この時点でアクションを実行した場合、UFT によってフライト予約アプリケーションが自動的に開かれ、指定した値が **【Username】** および **【Password】** ボックスに挿入されます。

#### 6. Login アクションに最後のステップを挿入します。

- a. 最後のステップの下の【項目】カラムで、【新規ステップ】をクリックして次のステップを挿入します。空の編集フィールドが開きます。
- b. 編集フィールド内をクリックします。【項目】リストが開き、前のステップのテスト・オブジェクトの兄弟オブジェクトが一覧表示されます。
- c. 【項目】リストから **OK** を選択します。  
このステップは、アプリケーションのユーザ名とパスワードを入力した後に【OK】をクリックするように UFT に指示します。

7. **テストを保存します。**

【ファイル】 > 【保存】を選択します。

キーワード・ビューとエディタについて詳しく学習するには、次の「[キーワード・ビューおよびエディタでの Login アクションの分析](#)」(52ページ)に進みます。

続いてテストにステップを追加する場合は、「[練習 3b：記録による FlightFinder アクションへのステップの追加](#)」(54ページ)に進みます。

# キーワード・ビューおよびエディタでの Login アクションの分析

いくつかのテスト・ステップの作成が完了したので、キーワード・ビューとエディタの両方でこれらのステップがどのように表示されるかを見てみましょう。

## キーワード・ビュー

ステップの追加後、キーワード・ビューは次のようになります。

項目	操作	値	注釈
HP MyFlight Sample Application			
agentName	Set	"john"	"agentName" edit box に "john" を入力する。
password	SetSecure	"55fc7bc5bc4efb3494b0"	暗号化されたパスワードを "password" edit box に入力する。
OK	Click		"OK" button をクリックする。
+ 新規ステップ			

図に示すように、テスト内のステップは、テスト・ステップの順序とオブジェクトの階層構造を表すアイコンベースのグリッド内に配置されます。キーワード・ビュー内の各行は、それぞれ異なる情報を表します。

行	説明
HP MyFlight Sample Application	<b>HP MyFlight Sample Application</b> ウィンドウ・オブジェクトは、このアクションに含まれるテスト・オブジェクトの親オブジェクトです。  以降のステップに含まれるすべてのテスト・オブジェクトが、このオブジェクトの

			サブノードとして表示されます。
 agentName	Set	"john"	<p><b>agentName</b> は UFT がアクションを実行するエディット・ボックスの名前です。</p> <p><b>Set</b> メソッドは <b>agentName</b> オブジェクトに対して実行されるアクションです。</p> <p>エディット・ボックスにテキストとして「john」が入力されます。</p>
 password	SetSecure		<p><b>password</b> は UFT がアクションを実行するエディット・ボックスの名前です。</p> <p><b>SetSecure</b> メソッドは passwordWatermark オブジェクトに対して実行されるアクションです。</p> <p>password エディット・ボックスにテキストとして数字文字列が入力されます。</p>
 OK	Click		<p><b>OK</b> は、agentName および password エディット・ボックスに情報が入力された後に UFT でクリックするボタンの名前です。</p> <p><b>Click</b> メソッドはこのボタンに対して実行されるアクションです。</p>

キーワード・ビューの各ステップごとに、複数の異なる要素が存在します。

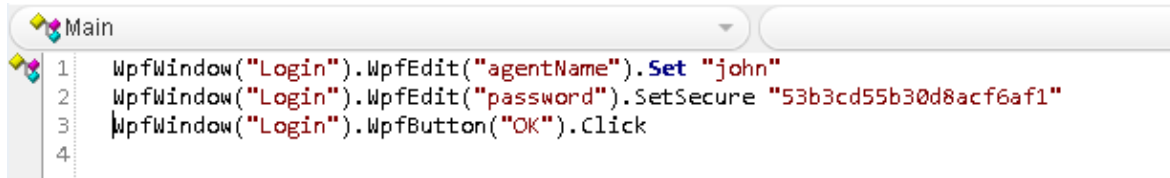
キーワード・ビューの要素	説明
項目	ステップの項目です（テスト・オブジェクト、ユーティリティ・オブジェクト、関数の呼び出し、またはステートメント）。この項目は、階層構造のアイコンベースのツリー形式で表示されます。
操作	項目に対して実行する操作（ <b>Click</b> 、 <b>Set</b> 、または <b>Select</b> など）。
値	選択した操作の引数値か（必要な場合。例：エディット・ボックスに入力するテキスト、画像をクリックするときに使うマウス・ボタンなど）。
注釈	自動的に提供されるステートメントです。ステップの動作を理解しやすい文で表します（例：“OK” ボタンをクリックする）。
割り当て	変数への値の割り当て、または変数からの値の割り当てです。これにより、後でテストの中でその値を使うことができます。このカラムは、標準設定では非表示です。
コメント	ステップに関して追加するテキスト形式の情報です。たとえば、「テストの最初のステップで使われたページに戻る」というコメントを追加できます。 このカラムは、標準設定では非表示です。

**ヒント:** キーワード・ビューでカラム見出しを右クリックし、リストからカラム名を選択することにより、個々のカラムを隠したり表示したりできます。

キーワード・ビューの詳細については、『HP Unified Functional Testing ユーザーズ・ガイド』の「キーワード・ビュー」の項を参照してください。

## エディタ

ステップの追加後、エディタは次のようになります。



```
1 WpfWindow("Login").WpfEdit("agentName").Set "john"
2 WpfWindow("Login").WpfEdit("password").SetSecure "53b3cd55b30d8acf6af1"
3 WpfWindow("Login").WpfButton("OK").Click
4
```

キーワード・ビューと異なり、エディタ内の各ステップは、次の形式のスクリプト行で表されます。

<オブジェクトの階層構造>.<メソッド> <メソッドのパラメータ>

そのため、各ステップ（およびスクリプトの各行）ごとに、次の複数の内容が表示されます。

テストのオブジェクトの階層構造	<p>エディタ内の各ステップごとに、以下を含むオブジェクトの階層構造<b>全体</b>が表示されます。</p> <ul style="list-style-type: none"><li>• テスト・オブジェクト・タイプ</li><li>• 各オブジェクトのオブジェクト名（UFT による識別名）</li></ul> <p>上記の例の最初の行には、次の両方の要素が存在します。</p> <ul style="list-style-type: none"><li>• <b>WpfWindow</b> はテスト・オブジェクト・タイプ</li><li>• <b>"Login"</b> はオブジェクト名（UFT による識別名）</li></ul>
オブジェクト・メソッド	<p>オブジェクトの階層構造の後には、メソッド（オブジェクトに対して実行するアクション）が存在します。各メソッドは太字で表示されます。</p> <p>上記の例の最初の行では、オブジェクトで <b>Set</b> メソッドが実行されます。</p>
メソッド・パラメータ	<p>数多くのメソッドに対して、必須パラメータまたはオプション・パラメータが存在します。エディタ内では、これらはメソッド名の後に表示されます。</p> <p>上記の例の最初の行では、<b>Set</b> メソッドでパラメータ John が入力されます。</p>

エディタの詳細については、UFTの「コードとテキスト・ドキュメントの編集」に関する項を参照してください。

## 練習 3b：記録による FlightFinder アクションへのステップの追加

「[練習 3a：キーワード・ビューでの Login アクションへのステップの追加](#)」では、キーワード・ビューを使用してステップを作成することで、フライト予約アプリケーションの [Login] ページで実行するステップをテストに追加しました。

この練習では、アプリケーションの [Flight Finder] ページ用に作成した **Flight Finder** アクションに関するステップを記録します。このアクションは、Flight Finder 共有オブジェクト・リポジトリに含まれるテスト・オブジェクトを使用します。

**ヒント:** 記録セッションを開始する前に、画面上でアプリケーション・ウィンドウとこのチュートリアル・ウィンドウを横に並べて配置できます。これにより、記録中にチュートリアルを読むことができます。

### 1. UFT を起動し、Book Flights テストを開きます。

- a. UFT が開いていない場合は、「すべてのテスト用のソリューションの作成」(25ページ)の手順に沿って UFT を開きます。WPF Add-in がロードされていることを確認します。
  - b. [ファイル] > [開く] > [ソリューション] を選択します。[ソリューションを開く] ダイアログ・ボックスが開きます。
  - c. [ソリューションを開く] ダイアログ・ボックスで、**C:\%HOMEPATH%\My Documents\Unified Functional Testing** にある **Flight Reservation Application.ftsln** ファイルに移動し、[開く] をクリックします。
- Flight Reservation Application ソリューションが開き、「[レッスン 1: GUI テストとアクションの作成](#)」(29ページ)で作成した Book Flights テストが表示されます。

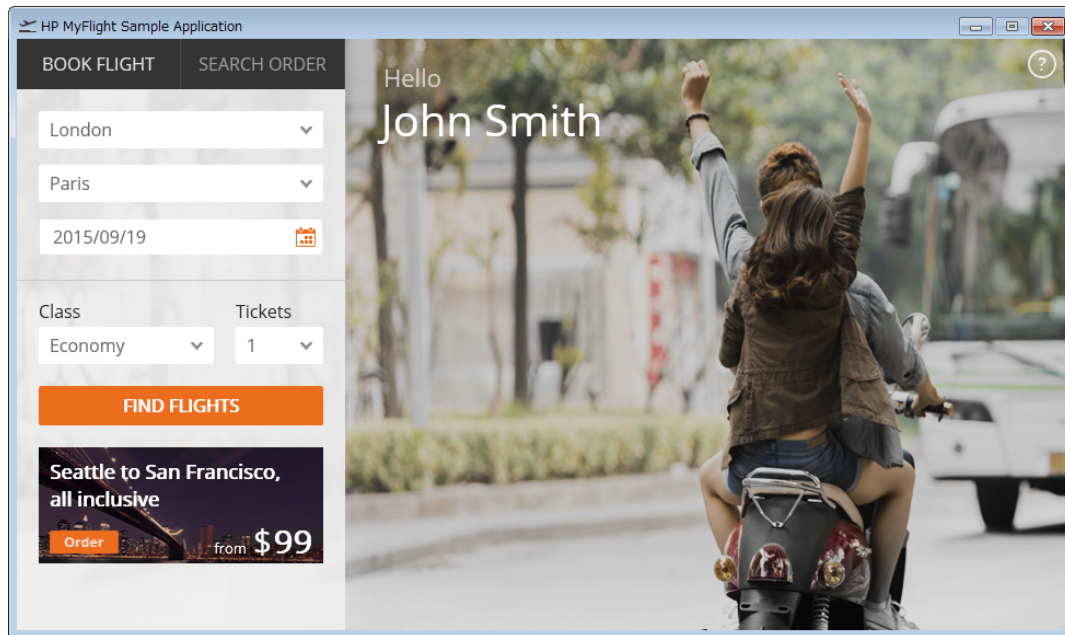
### 2. Flight Finder アクションを開きます。

- a. テスト・フロー・キャンバスがまだ開いていない場合は、ドキュメント表示枠で [Book Flights] タブをクリックして表示します。
  - b. キャンバスで、**Flight Finder** アクションをダブルクリックします。
- Flight Finder アクションがドキュメント表示枠に別のタブとして表示されます。

### 3. フライト予約アプリケーションを開きます。

- a. まだ開いていない場合は、フライト予約アプリケーションを開きます（「[フライト予約アプリケーションの確認](#)」(23ページ)を参照）。
- フライト予約アプリケーションの [ログイン] ページが開きます。
- b. アプリケーションにログインします。ユーザ名に john、パスワードに hp を使用します。

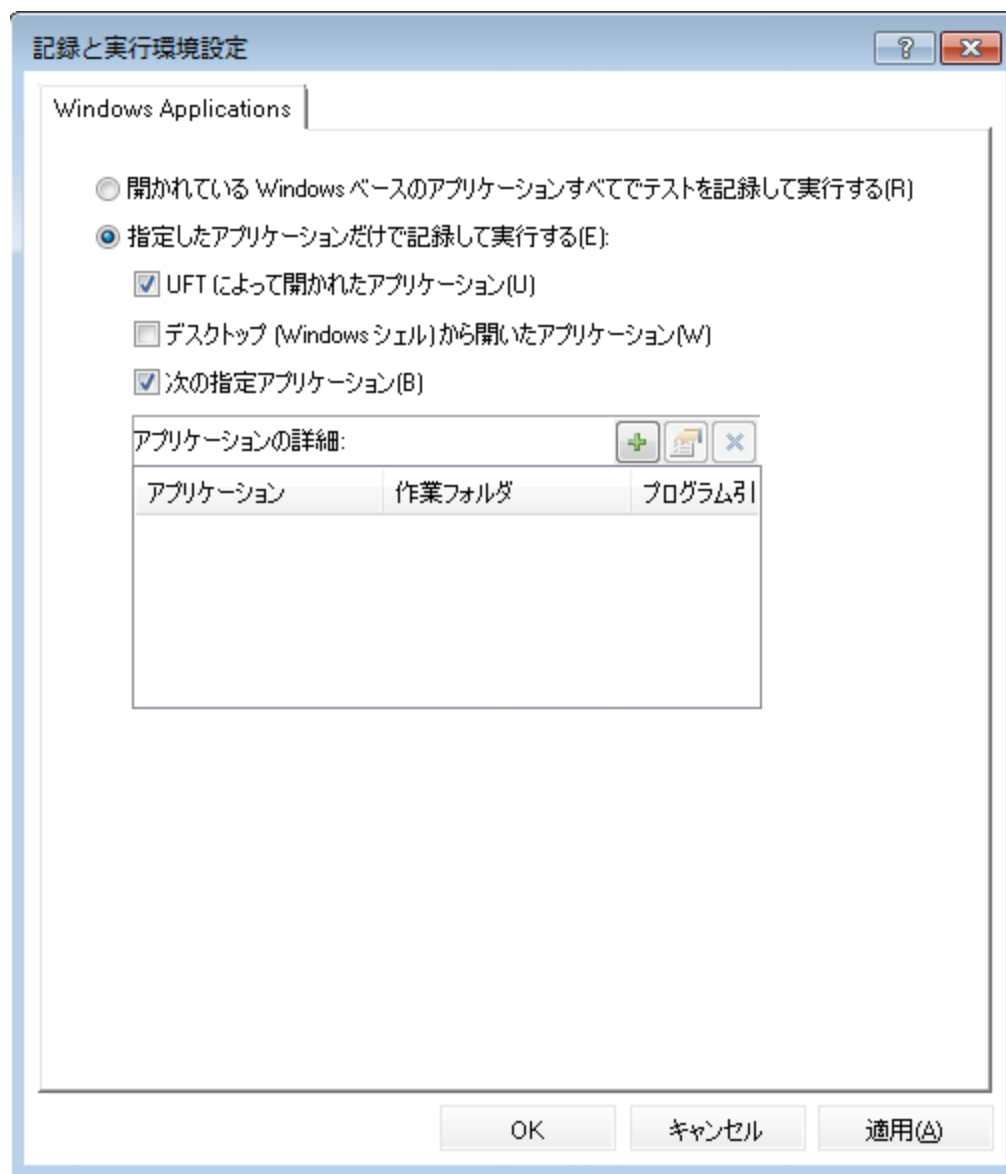
アプリケーションの [Flight Finder] ページが開きます。




- c. UFT で、**【記録】 > 【記録と実行環境設定】** を選択します。**【記録と実行環境設定】** ダイアログ・ボックスが開きます。



- d. [記録と実行環境設定] ダイアログ・ボックスの [Windows Applications] タブで、[開かれている Windows ベースのアプリケーションすべてでテストを記録して実行する] オプションを選択します。これにより、開かれているアプリケーションで UFT を実行できます。



- e. UFT で、[記録] ボタン  をクリックします。画面上部近くに記録ツールバーが表示され、UFT が記録を開始します。
- UFT ウィンドウが表示されなくなり、アプリケーションの [Flight Finder] ページが表示されます。
- f. [Flight Finder] ページで、各種項目を選択して次の選択内容を変更します。
- **From:** Los Angeles
  - **To:** Sydney

- **Date:** 明日の日付

**重要：** エディット・ボックスに日付を入力する際には、**M/D/YYYY** の形式を使用します。


- **Class:** Business
- **Tickets:** 2

**注:** このチュートリアルを月末または年末に行っている場合は、記録中に別の月または年を選択してください。UFT はアプリケーション内で変更が行われたときにだけ操作を記録するため、こうすることでステップを確実に記録できます。標準設定の値を受け入れる場合（または標準設定の値を選択しなす場合）は、UFT は操作を記録しません。

- g. 選択が済んだら、**[FIND FLIGHTS]** をクリックして続行します。「SELECT FLIGHT」ページが開きます。
- h. 記録ツールバーで、**[停止]** ボタンをクリックして、記録を停止します。

これで、ロサンゼルス発シドニー行きの架空のチケットを予約できました。UFT は、UFT で**[記録]** ボタンをクリックしてから記録ツールバーの**[停止]** をクリックするまでの間にアプリケーションで実行されたアクションを記録しました。

#### 4. テストを保存します。

**[保存]**  をクリックします。

引き続き別のアクションにステップを追加するため、テストは閉じないでください。続いて「[練習 3c：\[ツールボックス\] 表示枠を使用した Select Flight アクションへのステップの追加](#)」(58ページ)では、別のアクションにステップを追加します。

## 練習 3c：[ツールボックス] 表示枠を使用した Select Flight アクションへのステップの追加


「[練習 3b：記録による FlightFinder アクションへのステップの追加](#)」(54ページ)では、UFT の記録機能を使用して Flight Finder アクションにステップを追加しました。これにより、フライト予約アプリケーションのユーザ・インタフェースで実行したアクションを正確に反映したステップが作成されました。

この練習では、[ツールボックス] 表示枠から関連するテスト・オブジェクトをアクションにドラッグします。

**注:** [ツールボックス] 表示枠のアイテムは、ドキュメント表示枠でフォーカスのあるアクションに基づいて表示されます。テスト・フローまたは関数ライブラリのタブにフォーカスがある場合、または開いているテストが1つもない場合は、[ツールボックス] 表示枠は空です。

## 1. UFT を起動し、Book Flights テストを開きます。

- UFT が開いていない場合は、「すべてのテスト用のソリューションの作成」(25ページ)の手順に沿って UFT を開きます。


- 「開く」ボタンの下向き矢印をクリックし、「ソリューションを開く」を選択します。「ソリューションを開く」ダイアログ・ボックスが開きます。
- 「ソリューションを開く」ダイアログ・ボックスで、**C:\%HOMEPATH%\My Documents\Unified Functional Testing**にある**Flight Reservation Application.ftsln**ファイルに移動し、「開く」をクリックします。

Flight Reservation Application ソリューションが開き、「[レッスン 1 : GUI テストとアクションの作成](#)」(29ページ)で作成した Book Flights テストが表示されます。

## 2. Select Flight アクションを開きます。

ソリューション・エクスプローラで、「**Select Flight**」アクションをダブルクリックします。  
Select Flight アクションがドキュメント表示枠に別のタブとして表示されます。

## 3. 【ツールボックス】表示枠を表示します。

UFT のツールバーで、「**ツールボックス**」ボタンをクリックします。「ツールボックス」表示枠が UFT ウィンドウの右側に表示されます。

**注:** アクションがキーワード・ビューまたはエディタに表示されているかどうかに関係なく、「ツールボックス」表示枠には、アクションのテスト・オブジェクトと関数が表示されます。

## 4. グリッドから予約するフライトを選択します。

- 「ツールボックス」表示枠で、「**テスト オブジェクト**」ツリーを展開します。
- 「項目」カラムで「**新規ステップ**」ボタンをクリックします。空の編集フィールドが開きます。
- 「ツールボックス」表示枠で、**flightsDataGrid** オブジェクトを見つけて、キーワード・ビューまたはエディタ（開いているビューに依存）の編集フィールドにドラッグします。


**注:** このオブジェクト・リポジトリには、Login リポジトリよりはるかに多くのオブジェクトがあります。これは、このオブジェクト・リポジトリの作成時に関係のないオブジェクトを削除しなかったためです。

選択したテーブル・テスト・オブジェクトが標準メソッド **SelectCell** とともにステップに追加されます。


- メソッドの必要なパラメータを指定していないため、キーワード・ビューの「**注釈**」セルは空です。親テスト・オブジェクトがステップの一部であるため、このステップはキーワード・ビューで3つの行に表示されます。

- エディタでは、このステップは次のように表示されます。

```
WpfWindow("HP MyFlight Sample Application").WpfTable("flightsDataGrid").SelectCell
```

- キーワード・ビューが開いていない場合は、**【表示】** > **【キーワード ビュー】** を選択してキーワード・ビューを開きます。
- flightsDataGrid オブジェクトの**【値】** カラムの真ん中で、**【値の設定】** アイコン  をクリックします。

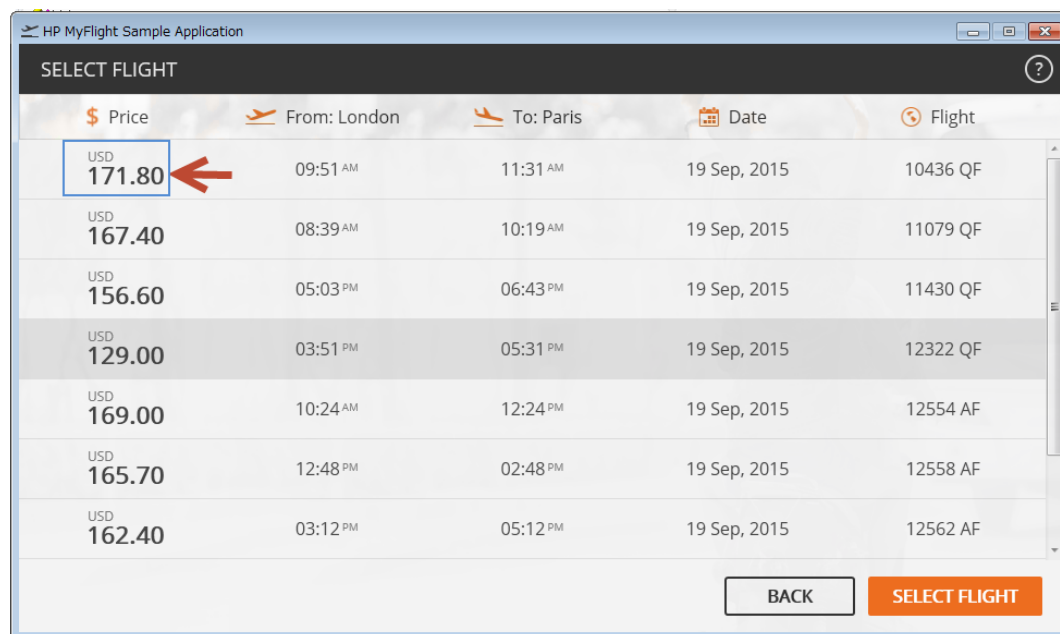
**注:** このアイコンをクリックすると、「row, Column」と書かれたツールヒントが表示されます。これは、row パラメータの値を設定していることを示します。メソッドのパラメータの値を設定する際には、常に設定中のパラメータを示すツールヒントが表示されます。

- row パラメータの値に「0」と入力します。
- 【値】** カラムの右側で、**【値の設定】** アイコン  をクリックします。

**注:** このアイコンをクリックすると、「row, Column」と書かれたを示すツールヒントが表示されます。これは **Column** パラメータの値を設定していることを示します。

- Column パラメータの値に「0」と入力します。

このパラメータを入力すると、最初の行のフライトを選択するように UFT が設定されます（下の例を参照）。



メソッドのパラメータを追加すると、キーワード・ビューとエディタでステップが更新されます。

- キーワード・ビューの【値】カラムには、ステップの値として "0", "0" が表示されます。
- エディタでは、**SelectCell** メソッドの後に "0", "0" が追加されます。エディタに表示されるステートメントは、次のようになります。

```
WpfWindow("HP MyFlight Sample Application").WpfTable("flightsDataGrid").SelectCell  
"0", "0"
```

- 【注釈】カラムには、ステップのアクションを説明するステートメントが表示されます。

#### 5. 【Select Flight】 ボタンをクリックするステップを追加します。

フライト・リストからセルを選択した後、フライト予約プロセスを続行するには、【Select Flight】 ボタンをクリックする必要があります。

- 【ツールボックス】表示枠で、**SELECT FLIGHT** オブジェクトを見つけます。
- 【項目】カラムで【新規ステップ】 ボタンをクリックします。空の編集フィールドが開きます。
- 【ツールボックス】表示枠から、**SELECT FLIGHT** オブジェクトを、キーワード・ビューの **flightsDataGrid** オブジェクトの下ステップ・グリッドのエディット・フィールドにドラッグします。

**SELECT FLIGHT** オブジェクトを含む新規ステップがテストに追加されます。

項目	操作	値	注釈
HP MyFlight Sample Application			
flightsDataGrid	SelectCell	"0", "0"	"flightsDataGrid" table で行 "0"、カラム "0" にあるセルを選択...
SELECT FLIGHT	Click		"SELECT FLIGHT" object をクリックする。
+ 新規ステップ			

この場合、**SELECT FLIGHT** オブジェクトの標準メソッド (**Click**) が、このテストで使用するメソッドです。

エディタでは、これらのアクション・ステップは次のようになります。

```
WpfWindow("HP MyFlight Sample Application").WpfTable("flightsDataGrid").SelectCell "0",  
"0"  
WpfWindow("HP MyFlight Sample Application").WpfObject("SELECT FLIGHT").Click
```

#### 6. テストを保存します。

【ファイル】 > 【保存】 を選択します。

引き続き別のアクションにステップを追加するため、テストは閉じないでください。続いて「[練習 3d：ステップ・ジェネレータを使用した Book Flight アクションへのステップの追加](#)」(62ページ)では、ステップ・ジェネレータを使用してステップを追加します。

## 練習 3d：ステップ・ジェネレータを使用した Book Flight アクションへのステップの追加

「[練習 3c：\[ツールボックス\] 表示枠を使用した Select Flight アクションへのステップの追加](#)」(58ページ)では、[ツールボックス] 表示枠に表示されるオブジェクトを使用してテスト・ステップを作成しました。

このレッスンでは、ステップ・ジェネレータを使用してテスト・ステップを作成します。ステップ・ジェネレータでは、キーワード・ビューのさまざまなカラムにステップの個別の部分挿入する方法の代わりに、1つのダイアログ・ボックスでステップ全体を定義できます。

### 1. **Flight Confirmation アクションを開きます。**

ソリューション・エクスプローラで、[Book Flights] テスト・ノードの下に [Flight Confirmation] アクションをダブルクリックします。

Flight Confirmation アクションがドキュメント表示枠に別のタブとして表示されます。

### 2. **ステップ・ジェネレータを使用してステップを追加します。**

- a. エディタが表示されていない場合は、[表示] > [エディタ] を選択してエディタを表示します。

- b. エディタの最初の行で、右クリックして **【ステップの挿入】** > **【ステップ ジェネレータ】** を選択します。 **【ステップ ジェネレータ】** ダイアログ・ボックスが開きます。

ステップ ジェネレータ

カテゴリ(Y): テストオブジェクト

オブジェクト(O): "HP MyFlight Sample Application"

☒ テストオブジェクト操作(T) ☐ ネイティブの操作(N)

操作(O): Activate

引数(A):

名前	タイプ	値
BUTTON	定数	<値なし>

☐ 戻り値(R)

ステップについてのコメント(D):

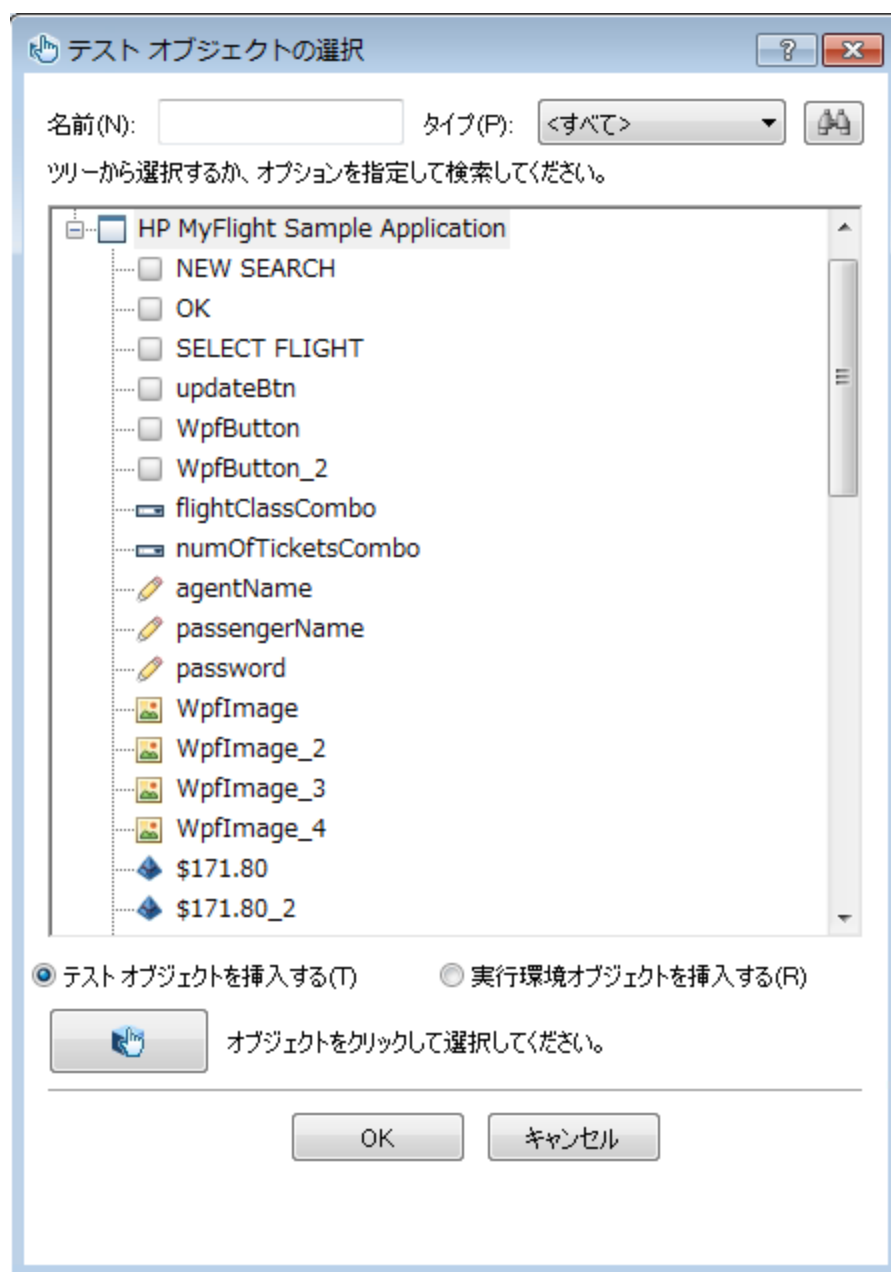
"HP MyFlight Sample Application" window をアクティブにする。


☐ 他のステップを挿入する(S)

OK キャンセル

- c. **【オブジェクト】** ドロップダウンで、 **"Book Flights"** を選択します。
- d. **【オブジェクトの選択】** ボタン をクリックします。 **【テストオブジェクトの選択】** ダイアログ・ボックスが開きます。

- e. [テスト オブジェクトの選択] ダイアログ・ボックスで、**Book Flights** ノードを展開します。



- f. **passengerName** エディット・ボックス  を選択し、[OK] をクリックします。



ステップ・ジェネレータに、**passengerName** オブジェクトの標準設定のオプションが表示されます。

ステップ ジェネレータ

カテゴリ(Y): テストオブジェクト

オブジェクト(O): "passengerName"

☒ テストオブジェクト操作(T) ☐ ネイティブの操作(N)

操作(O): Set

引数(A):

名前	タイプ	値
text *	String	

\* 必須引数

☐ 戻り値(R)

ステップについてのコメント(D):

必須引数をすべて定義してください。

☐ 他のステップを挿入する(S)

OK キャンセル

- g. キーワード・ビューの場合と同じように、ステップ・ジェネレータで引数と値を次のように定義します。
- **【操作】** ドロップダウン・リストで、操作を **Set** のままにします。 **【オブジェクト】** ドロップダウン・リストでテスト・オブジェクトを選択する際には、標準操作が表示されます。必要に応じて、このオブジェクトに対して別の操作を選択することもできます。ただし、この練習では、標準操作が適切な操作になります。
  - **【引数】** テーブルで、 **【値】** カラム内をクリックし（キーワード・ビューの場合と同様）、好みの名前を入力します。引数が必須の場合は、引数名の横に赤いアスタリスクが表示されます。
  - **【ステップについてのコメント】** にこのステップに対する指示が表示されます。キーワード・ビューの **【注釈】** セルに表示されるものと同様です。
  - **【他のステップを挿入する】** チェックボックスを選択し、このステップを追加した後に **【ステップ ジェネレータ】** ダイアログ・ボックスを再度開きます。

- h. **【OK】** をクリックします。ステップ・ジェネレータは開いたままですが、入力した詳細情報に基づいてバックグラウンドでエディタにステップが追加されます。

### 3. アプリケーション・ウィンドウで進行状況バーのロードを待機するステップを追加します。

【Flight Details】ウィンドウには、進行状況バー・オブジェクトが存在します。これは、オブジェクト・リポジトリ内で **progBar** と呼ばれます。テスト時にアプリケーションを正常に動作させるには、ボタンをクリックして注文を完了する前に進行状況バーを確実にロードするためのステップを追加する必要があります。

- a. ステップ・ジェネレータの **【テスト オブジェクト】** ドロップダウン・リストで、**progBar** オブジェクトを選択します。【ステップ ジェネレータ】ダイアログ・ボックスのフィールドに、**progBar** オブジェクトの標準プロパティ（標準メソッドの **Value** など）が反映されます。
- b. **【操作】** ドロップダウン・メニューで、**【WaitProperty】** を選択します。このメソッドは、テスト実行中に特定のプロパティが指定された状態に達するまで待機するように UFT を設定します。
- c. **【引数】** テーブルで、次の情報を入力します。

項目	値
PropertyName	value
PropertyValue	100

- d. **【他のステップを挿入する】** チェックボックスを選択し、**【OK】** をクリックします。  
バックグラウンドでキーワード・ビューにステップが挿入されます。【ステップ ジェネレータ】ボックスは開いたままです。

### 4. ステップ・ジェネレータを使用して注文を完了するステップをテストに追加します。

注文の詳細の定義が済んだら、【Flight Details】ウィンドウで **【ORDER】** ボタンと **【NEW SEARCH】** ボタンをクリックして注文を完了するステップを指定する必要があります。ここでもステップ・ジェネレータを使用して、このステートメントを挿入します。

**注:** **【ORDER】** ボタンをクリックするステップは、前のステップで作成した進行状況バー・オブジェクトのステップの前に挿入する必要があります。

- a. 前のステップの手順を使用して、次の詳細情報を入力します。

	ORDER ボタン・ステップ	NEW SEARCH ボタン・ステップ
オブジェクト	ORDER (WpfButton)	NEW SEARCH (WpfButton)
操作	Click	Click
引数	空のまま	空のまま
【他のステップを挿入する】 チェックボックス	Select	Clear

- b. **【OK】** をクリックします。ステップ・ジェネレータが閉じ、キーワード・ビューにステップが追加されます。


両方のステップをテストに追加すると、キーワード・ビューは次のようになります。

項目	操作	値	注釈
HP MyFlight Sample Application			
passengerName	Set	"John Smith"	"passengerName" edit box に "John Smith" を入力する。
ORDER	Click		"ORDER" button をクリックする。
progBar	WaitProperty	"value", "100"	次のステップに進む前に、"progBar" progress bar にある "valu..."
NEW SEARCH	Click		"NEW SEARCH" button をクリックする。
+ 新規ステップ			

エディタでは、これらのステップは次のように表示されます。

```
WpfWindow("HP MyFlight Sample Application").WpfEdit("passengerName").Set "John Smith"
WpfWindow("HP MyFlight Sample Application").WpfButton("ORDER").Click
WpfWindow("HP MyFlight Sample Application").WpfProgressBar("progBar").WaitProperty
"value", "100"
WpfWindow("HP MyFlight Sample Application").WpfButton("NEW SEARCH").Click
```

## 5. テストを保存します。

**【保存】**  をクリックします。

最初のテストが作成されたので、テストを実行できます。続いて「[レッスン 4 : GUI テストの実行と分析](#)」(73ページ)に進み、テストの実行の詳細について学習します。

さらに高度な内容を希望する場合は、「[高度な練習 3e \(オプション\) - エディタを使用したステップの追加](#)」(67ページ)に進み、エディタでステップを追加する方法について学習します。

## 高度な練習 3e (オプション) - エディタを使用したステップの追加

キーワード・ビュー、[ツールボックス] 表示枠、またはステップ・ジェネレータを使用してテストにステップを追加する以外に、エディタでステップを直接追加することもできます。

ただし、エディタでステップを追加する場合は、アプリケーションとテスト・オブジェクトの両方に関する豊富な知識が必要です。キーワード・ビュー、[ツールボックス] 表示枠、ステップ・ジェネレータでは、テスト・オブジェクトに関するすべての情報が UFT のダイアログ・ボックスに表示されます。エディタでは、次の内容を把握しておく必要があります。

- オブジェクトの完全なテスト・オブジェクト階層
- テスト・オブジェクトの名前（オブジェクト・リポジトリで記録した名前）
- テスト・オブジェクトのタイプ（WpfWindow, WpfButton など）
- 使用するメソッド

これらの情報を使用して、エディタで行を記述します。このレッスンでは、これらの情報を見つける場所およびこれらの情報をエディタに入力してテスト・ステップを作成する方法について学習します。

すでにテスト・ステップが存在するアクションに対してエディタでステートメントを作成します。

### 1. UFT を起動し、Book Flights テストを開きます。

- a. UFT が開いていない場合は、「すべてのテスト用のソリューションの作成」(25ページ)の手順に沿って UFT を開きます。WPF Add-in がロードされていることを確認します。
  - b. [ファイル] > [開く] > [ソリューション] を選択します。[ソリューションを開く] ダイアログ・ボックスが開きます。
  - c. [ソリューションを開く] ダイアログ・ボックスで、**C:\%HOMEPATH%\My Documents\Unified Functional Testing** にある **Flight Reservation Application.ftsln** ファイルに移動し、[開く] をクリックします。
- Flight Reservation Application ソリューションが開き、「[レッスン 1 : GUI テストとアクションの作成](#)」(29ページ)で作成した Book Flights テストが表示されます。

### 2. Flight Confirmation アクションを開きます。

[ソリューション エクスプローラ] 表示枠で、[Flight Confirmation] アクションをダブルクリックします。

Flight Confirmation アクションがドキュメント表示枠に別のタブとして表示されます。アクション内の既存のテスト・ステップを削除しないでください。

### 3. 親オブジェクトのオブジェクトの詳細を確認します。

- a. エディタが開いていない場合は、[表示] > [エディタ] を選択してエディタを開きます。既存のステップは次のように表示されます。

```
WpfWindow("HP MyFlight Sample Application").WpfEdit("passengerName").Set "John Smith"
WpfWindow("HP MyFlight Sample Application").WpfButton("ORDER").Click
```

これらのステートメントの各要素の説明については、「[キーワード・ビューおよびエディタでの Login アクションの分析](#)」(52ページ)の「エディタ」の項を参照してください。

- b. ソリューション・エクスプローラで、**[Book Flights]** ノードの下**[Flight Confirmation]** アクション・ノードを展開します。
- c. **Flight Confirmation.tsr** オブジェクト・リポジトリをダブルクリックします。オブジェクト・リポジトリ・マネージャが開いて、フライト予約アプリケーションの**[Flight Details]** ページのオブジェクトが表示されます。
- d. オブジェクト・リポジトリ・マネージャで、**HP MyFlight Sample Application** オブジェクト（最上位ノード）を選択します。これは、**[Flight Details]** ページのすべてのオブジェクトの親オブジェクトです。

オブジェクトの詳細とプロパティが**[オブジェクトのプロパティ]** 領域（ウィンドウの右側）に表示されます。

**オブジェクトのプロパティ**

名前(N):

クラス:

リポジトリ(R):

テストオブジェクトの詳細 + × ↺

名前	値
<b>記述プロパティ</b>	
wpftypename	window
regexwndtitle	HP MyFlight Sample Application
devname	HP MyFlight Sample Application
<b>ビジュアル関係識別子</b>	
ビジュアル関係識別子の設定	[なし。クリックすると追加]
<b>順序識別子</b>	
タイプ、値	None
<b>追加の詳細</b>	
スマート認識を有効にする	False
コメント	

- e. **HP MyFlight Sample Application** オブジェクトの以下の詳細を記録します。

- **名前** : HP MyFlight Sample Application
- **クラス** : WpfWindow

これはエディタでステートメントを作成するときに必要になります。この情報は、このページのテスト・オブジェクトを使用するステップのすべてのステートメントの最初の部分になります。

#### 4. 子オブジェクトのオブジェクトの詳細を確認します。

「練習 3d : ステップ・ジェネレータを使用した Book Flight アクションへのステップの追加」(62 ページ) (この練習の基になる練習) では、航空券申し込みの名前を入力するステップと **[ORDER]** ボタンをクリックするステップの2つのステップを作成しました。エディタでこれらのステップを作成するためには、これらのステップに含まれるオブジェクトの詳細について

も把握する必要があります。

- a. オブジェクト・リポジトリ・マネージャで、**passengerName** オブジェクトを選択します。  
テスト・オブジェクトの詳細が、オブジェクト・リポジトリ・マネージャの「**オブジェクトのプロパティ**」セクションに表示されます。

オブジェクトのプロパティ

名前(N): passengerName

クラス: WpfEdit

リポジトリ(R): Local

テスト オブジェクトの詳細

名前	値
記述プロパティ	
devname	passengerName
ビジュアル関係識別子	
ビジュアル関係識別子の設定	[なし。クリックすると追加]
順序識別子	
タイプ、値	None
追加の詳細	
スマート認識を有効にする	False
コメント	

- b. オブジェクトの以下のプロパティを記録します。
    - **名前** : passengerName
    - **クラス** : WpfEdit
  - c. **ORDER** オブジェクトについても、同じ手順を実行します。
5. **エディタでステップのステートメントを作成します。**

このアクションのステップに関連するオブジェクトのプロパティを表示すると、次の内容が含まれています。

オブジェクト	名前	クラス
HP MyFlight Sample Application ウィンドウ（親オブジェクト）	HP MyFlight Sample Application	WpfWindow
Passenger Name（エディット・ボックス）	passengerName	WpfEdit
ORDER ボタン	ORDER	WpfButton

オブジェクトの詳細を使用して、オブジェクトの階層構造とオブジェクトに対して実行するメソッド（アクション）を含むステートメントを作成する必要があります（オブジェクトごとに

テスト・オブジェクトで利用できる複数のメソッドが存在します。使用可能なすべてのオブジェクトとメソッドの詳細については、チュートリアルの練習を終了した後に『HP UFT Object Model Reference for GUI Testing』を参照してください。

- a. 最初の新規行には、次の形式を使用して、（注文の顧客名を入力する）最初のステップの親オブジェクトの階層構造を入力します。

```
<オブジェクト・クラス>("<オブジェクト名>").
```

このステップでは、次のように入力します。

```
WpfWindow("HP MyFlight Sample Application").
```

- b. 同じ形式を使用して、最初のステップの子オブジェクト（**passengerName**）を入力します。このステップでは、次のように入力します。

```
WpfEdit("passengerName").
```

- c. **WpfEdit("passengerName")** オブジェクトの後に、**passengerName** オブジェクトに対する **Set** メソッドを入力します。

ステップは次のようになります。

```
WpfWindow("HP MyFlight Sample Application").WpfEdit("passengerName").Set
```

- d. **Set** メソッドの後に、**passengerName** オブジェクトに対して入力（Set）する文字列として "John Smith" を入力します。

ステップは次のようになります。

```
WpfWindow("HP MyFlight Sample Application").WpfEdit("passengerName").Set "John Smith"
```

- e. 上記と同じ手順を使用して、（**ORDER** ボタンをクリックする）2 番目のステップのオブジェクトの階層構造を追加します。

2 番目のステップのオブジェクトの階層構造を追加すると、ステートメントは次のようになります。

```
WpfWindow("HP MyFlight Sample Application").WpfButton("ORDER").
```

- f. **WpfButton("ORDER")** オブジェクトの後に **Click** メソッドを追加します。ステートメントは次のようになります。

```
WpfWindow("HP MyFlight Sample Application").WpfButton("ORDER").Click
```

**注:** **Click** メソッドにはパラメータが必要ないため、メソッド名の後に情報を追加する必要はありません。

2つのステートメントの入力が完了すると、エディタには次のように表示されます。

```
WpfWindow("HP MyFlight Sample Application").WpfEdit("passengerName").Set "John Smith"  
WpfWindow("HP MyFlight Sample Application").WpfButton("ORDER").Click
```

#### 6. 余分な行を削除します。

上記の手順を実行したことで、2つの余分な行がアクションに追加されています（合計で4つのステートメント）。最後の2つの行をアクションから削除してテストが正常に実行されるようにします。

#### 7. テストを保存します。

【ファイル】 > 【保存】を選択します。



# レッスン 4：GUI テストの実行と分析

「[レッスン 3：テストへのステップの追加](#)」(46ページ)では、フライト予約アプリケーションをテストするため、複数のアクションと各アクション内のテスト・ステップを作成しました。テストが完成したら、テストを実行してフライト予約アプリケーションがどのように実行されるかを確認できます。

このレッスンでは、テストを実行して実行結果を表示する方法について学習します。

このレッスンには、次の手順が含まれています。

- [練習 4a：テストの実行](#) ..... 74
- [練習 4b：実行結果の操作](#) ..... 76
- [練習 4c：実行結果の分析](#) ..... 77

## 練習 4a：テストの実行

「レッスン3：テストへのステップの追加」(46ページ)では、フライト予約アプリケーションを実行してフライトを予約する基本的なテストを作成しました。

この練習では、作成が完了したテストを実行する方法について学習します。

### 1. UFT を起動し、Book Flights テストを開きます。

- a. UFT が開いていない場合は、「すべてのテスト用のソリューションの作成」(25ページ)の手順に沿って UFT を開きます。WPF Add-in がロードされていることを確認します。
- b. [ファイル] > [開く] > [ソリューション] を選択します。[ソリューションを開く] ダイアログ・ボックスが開きます。
- c. [ソリューションを開く] ダイアログ・ボックスで、**C:\%HOMEPATH%\My Documents\Unified Functional Testing** にある **Flight Reservation Application.ftsln** ファイルに移動し、[開く] をクリックします。  
ソリューション・エクスプローラに Flight Reservation Application ソリューションが開き、Book Flights テストがドキュメント表示枠に別のタブとして表示されます。


### 2. すべての画像を実行結果に保存するように UFT を設定します。

UFT では、テストを実行するときに、すべての画像をテスト結果に保存するかどうかを選択できます。

- a. [ツール] > [オプション] > [GUI テスト] タブ > [画面キャプチャ] ノードを選択します。[画面キャプチャ オプション] 表示枠が開きます。
- b. [画面キャプチャ オプション] 表示枠で、[静止画像キャプチャをテスト結果へ保存] チェックボックスを選択し、ドロップダウン・メニューから [常に] を選択します。
- c. [OK] をクリックして [オプション] ダイアログ・ボックスを閉じます。

### 3. テストの記録と実行環境設定を設定します。

場合によっては、テスト実行の開始時に UFT でアプリケーションを開く必要があります。このような場合には、[記録と実行環境設定] でこの操作を可能にすることができます。


- a. [記録] > [記録と実行環境設定] を選択します。[記録と実行環境設定] ダイアログ・ボックスが開きます。
- b. [記録と実行環境設定] ダイアログ・ボックスの [Windows Applications] タブで、[指定したアプリケーションだけで記録して実行する:] オプションを選択します。
- c. [指定したアプリケーションだけで記録して実行する] オプションの下にある [次の指定アプリケーション] オプションを選択します。
- d. [アプリケーションの詳細] 領域で、[追加] ボタン  をクリックします。[アプリケーションの詳細] ダイアログ・ボックスが開きます。

- e. 【アプリケーションの詳細】ダイアログ・ボックスで、アプリケーションの詳細を入力します。

アプリケーション	C:\%HOMEPATH%\Unified Functional Testing\samples\Flights Application\FlightsGUI.exe
作業フォルダ	C:\%HOMEPATH%\Unified Functional Testing\samples\Flights Application\

- f. 【OK】をクリックして【アプリケーションの詳細】ダイアログ・ボックスを閉じます。  
g. 【記録と実行環境設定】ダイアログ・ボックスで、【適用】をクリックしてから【OK】をクリックし、設定を有効にしてダイアログ・ボックスを閉じます。

#### 4. テストの実行を開始します。

- a. ソリューション・エクスプローラで、【Book Flights】テスト・ノードを選択します。  
b. ツールバーで【実行】ボタン  をクリックします。【テストの実行】ダイアログ・ボックスが開きます。  
c. 【テストの実行】ダイアログ・ボックスで、【オプション】バーをクリックしてテストの実行オプション領域を展開し、【新規実行結果フォルダ】オプションが選択されていることを確認します。標準設定フォルダ名をそのまま使用します。

実行

?

×

テスト名(E): 

GUITest2

結果保管場所: C:\Users\zgy.WIN-9KD3R2TGAGQ\Desktop\Feedback\SelectLinkDataSource\_DataSourceColumn\GUITest2\Res1

オプション: 

⌵

結果保管場所

入力パラメータ

結果の書き込み先:

☒ 新規実行結果フォルダ(N):

C:\Users\zgy.WIN-9KD3R2TGAGQ\Desktop\Feedback\SelectLinkDataSource\_DataSourceColumn\GUITest2\Res1

...

☐ 一時実行結果フォルダ (古い一時結果に上書き)(T)

C:\Users\zgy.WIN-9KD3R2TGAGQ\AppData\Local\Temp\TempResults

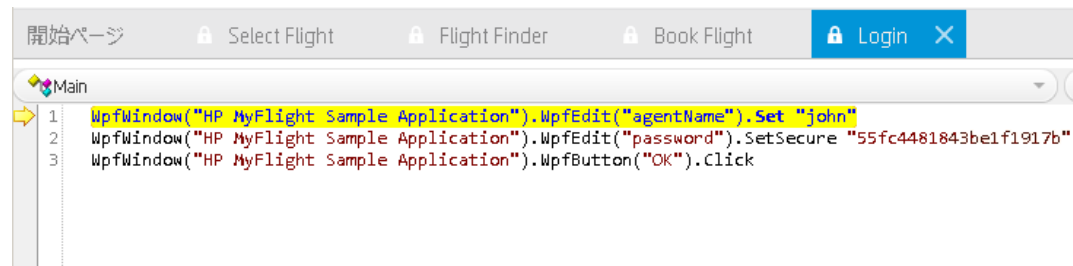
▶ 実行

キャンセル

- d. 【実行】をクリックして【実行】ダイアログ・ボックスを閉じ、テストの実行を開始します。

UFT によってアプリケーションが開かれ、テストの実行が開始されるので、その様子を注意深く観察します。挿入した各ステップが UFT によって実行される様子をアプリケーションで

見ることができます。キーワード・ビューまたはエディタの左マージンにある黄色い矢印と強調表示されている行は、UFT によって現在実行されているステップを示します。



エラーが表示された場合は、エラー・メッセージで示されるテスト内の箇所に移動し、関連する練習の説明どおりにステップが構成されているかどうかを確認します。

テスト実行が完了すると、実行結果がドキュメント表示枠に別のタブとして表示されます。続いて「[練習 4b : 実行結果の操作](#)」(76ページ)では、実行結果について詳しく学習します。

## 練習 4b : 実行結果の操作

「[練習 4a : テストの実行](#)」(74ページ)では、作成した Book Flights テストを実行しました。テスト実行が終了すると、実行結果にテスト実行の結果が自動的に表示されます。

**注:** 標準設定では、実行結果は HTML ベースのレポートに表示されます。実行結果は、[オプション] ダイアログ・ボックスの[実行セッション] 表示枠の Run Results Viewer に表示することもできます ([ツール] > [オプション] > [一般] タブ > [実行セッション] ノード)。このチュートリアルのレッスンは、HTML ベースのレポートに基づいています。

実行結果が開くと、次の内容が表示されます。



標準設定では、実行結果に次の内容が表示されます。

<b>テスト・フロー</b>	結果を視覚的に表現したツリーです。アクションとテストの実行中にアクセスしたアプリケーション
----------------	---

	ション・ページに応じて構成されています。実行中に実施されたステップはツリー上ではアイコンとして表示され、展開（矢印をクリック）すると、各ステップを表示できます。実行ごとに異なるデータ・セットを使用してテストやアクションを複数回実行するように UFT に指示することもできます。テストの各実行を反復といい、それぞれに番号が付きます（実行したテストの反復は1つのみです）。
エラー・リスト	すべてのエラーと警告のリストが表示されます。
ステップのサマリ情報	概要レベルの結果レポートで、テストに関する一般情報、成功または失敗したステップ、および各テスト・ステップに関する詳細が含まれます。
外部リソースへのリンク	テストで使用する外部リソースまたはテスト実行からの外部リソースへのリンク。次のものが含まれます。 <ul style="list-style-type: none"><li>データ・テーブル</li><li>実行環境ムービー</li><li>アプリケーション・ログ</li></ul>

この練習では、追加したステップに従って UFT でフライト予約アプリケーションを操作できたため、テスト実行は成功しました。エラーが発生し、テストが正常に実行されなかった場合、そのエラーは実行結果に表示されます。このような場合は、前に戻って、このチュートリアルの説明どおりにステップが設定されていることを確認してください。

実行結果の表示について理解できたら、続いて「[練習 4c : 実行結果の分析](#)」(77ページ)で実行結果の詳細について学習します。

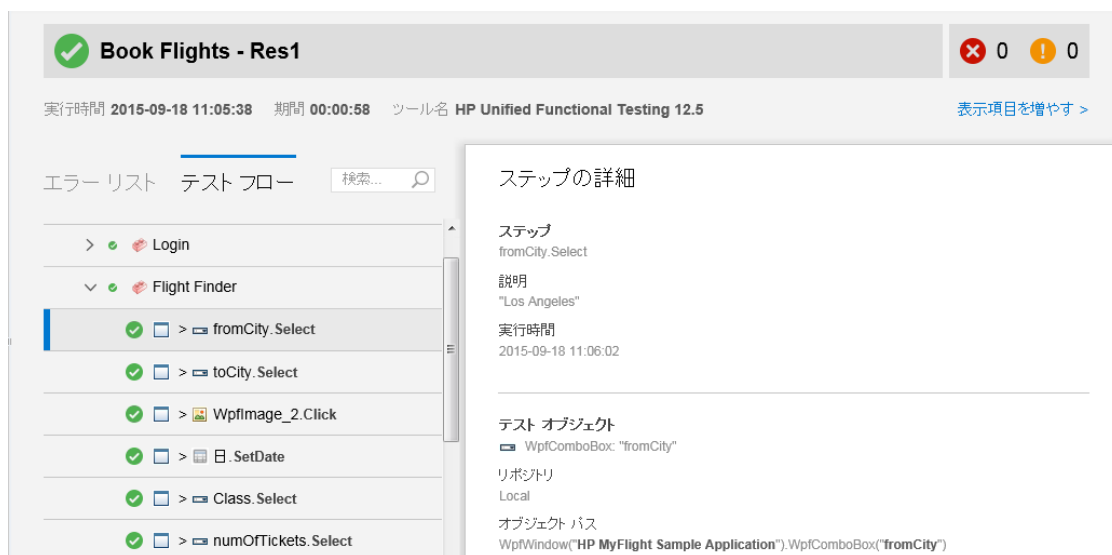
## 練習 4c : 実行結果の分析

この練習では、テストを「[練習 4a : テストの実行](#)」(74ページ)で実行したときに UFT によって実行されたステップをチェックします。アプリケーション・ウィンドウのスナップショットはステップごとに表示できます。

### 1. 特定のステップの実行結果を表示します。

テスト・フローの結果ツリーで、**Flight Finder** ノードを見つけて、フライト予約アプリケーションの [Flight Finder] ページで実行されたすべてのステップを確認します。

結果ツリーで **fromCity.Select** ステップを選択します。



実行結果に次の情報が表示されます。

- テスト・フロー：ステップが強調表示されます
- テスト・ステップのサマリ：強調表示されたステップの詳細が表示されます

## 2. 実行結果を閉じます。

ドキュメント表示枠で、実行結果を含むタブを閉じます。

最初のテストの設定と実行が済んだら、続いてテストを拡張するさまざまな方法について学習できます。次のいずれかを選択してください。

- 「レッスン 5 : ステップとオブジェクトのパラメータ化」(79ページ)
- 「レッスン 6 : チェックポイントおよび出力値の作成」(94ページ)
- 「レッスン 7 : 関数と関数ライブラリの作成」(129ページ)
- 「レッスン 8 : テストでの Insight の使用」(139ページ)

# レッスン 5：ステップとオブジェクトのパラメータ化

「[レッスン 3：テストへのステップの追加](#)」(46ページ)では、フライト予約アプリケーションで実行される一連のステップがスムーズに実行されることを検査するテスト・ステップを作成しました。

「[レッスン 4：GUI テストの実行と分析](#)」(73ページ)では、1 組のデータのみを使用してテストを実行しました。ところが、実際のアプリケーションをテストするときは、同じ操作を複数のデータ・セットで行うことが必要な場合もあります。

たとえば、10 種類の異なるデータ・セットを使用してアプリケーションのテストを実行することができます。その場合は、10 個のテストを別々に作成し、それぞれで独自のデータ・セットを使用するか、1 つのテストに 10 組のパラメータを追加することができます。パラメータを追加する場合は、毎回異なるデータ・セットを使用してテストを 10 回実行することになります。

このレッスンでは、パラメータをテストに追加し、複数のデータ・セットでテストを実行します。

このレッスンには、次の手順が含まれています。


- [テスト、アクション、オブジェクトのパラメータ化 - 概要](#) .....80
- [練習 5a：パラメータ化用のテストの作成](#) ..... 80
- [練習 5b：データ・テーブル・パラメータの定義](#) .....81
- [練習 5c：データ・テーブルへのパラメータ値の追加](#) .....85
- [練習 5d：パラメータ化されたテストの実行](#) .....87

## テスト、アクション、オブジェクトのパラメータ化 - 概要

データを使用してテスト、アクション、またはテスト・オブジェクトの値をパラメータ化するには、次の複数の場所からデータ・ソースを提供できます。

- ・ **データ・テーブル**：パラメータ名と値を含む Excel スプレッドシート
- ・ **環境変数**：テスト内で固定値が設定される変数
- ・ **乱数**：テスト実行で UFT によって生成される乱数

これらの中で最も一般的なのは、データ・テーブル・パラメータです。データ・テーブルは Excel スプレッドシートで、UFT ウィンドウの下部にある [データ] 表示枠に表示されます。

**注：** [データ] 表示枠が表示されない場合は、[表示] > [データ] を選択するか、ツールバーの [データ] ボタン  をクリックします。

データ・テーブルには、次の 2 種類のシートが存在します。

<b>グローバル・データ・シート</b>	<p>グローバル・データ・シートには、テスト内のすべてのアクションで使用される（使用可能な）データ・パラメータとデータが含まれます。グローバル・シートに挿入されたパラメータは、テスト内のアクションおよびアクションのステップのいずれでも使用できます。</p> <p>テストはグローバル・データ・シート内の行数と同じ反復回数だけ実行されます。たとえば、5つのデータ行が存在する場合、テストは5回反復実行されます。</p>
<b>アクション・シート</b>	<p>テスト内のアクションごとに、そのアクションに関するシートが（アクションと同じ名前で）追加されます。データ・パラメータとデータは、そのアクションのステップのみで使用できます。</p> <p>アクション・シート内で複数のデータ行を使用すると、UFT は（1回のテスト反復内で）データ・シート内の行数と同じ回数だけアクションを実行します。</p>

このレッスンでは、データ・テーブル・パラメータのみを使用します。ほかのタイプのパラメータの詳細については、『HP Unified Functional Testing ユーザーズ・ガイド』のパラメータ化に関する項を参照してください。

## 練習 5a：パラメータ化用のテストの作成

「練習 3b：記録による FlightFinder アクションへのステップの追加」(54ページ)では、ロサンゼルス発シドニー行きの航空券を予約しました。これらのステップでは、ロサンゼルスとシドニーの値は固定値です。これは、UFT でテストを実行するたびに、出発地および到着地として **Los Angeles** と **Sydney** が使用されることを意味します。

この練習で作成する新しいテストでは、出発地と到着地をパラメータとして定義し、テスト実行のたびに異なる出発地と到着地が使われるようにします。



### 1. UFT を起動し、Book Flights テストを開きます。

- a. UFT を開きます（「すべてのテスト用のソリューションの作成」(25ページ)を参照）。WPF Add-in がロードされていることを確認します。
- b. 開始ページの「最近のソリューション」領域で、「Flight Reservation Application」をクリックします。  
Flight Reservation Application ソリューションが開きます。これには、「レッスン 1：GUI テストとアクションの作成」(29ページ)で作成した **Book Flights** テストが含まれます。

### 2. Book Flights を Book Flights Parameter という名前で保存します。

- a. ソリューション・エクスプローラで、「Book Flights」ノードを選択します。
- b. 「ファイル」>「名前を付けて保存」を選択します。「テストに名前を付けて保存」ダイアログ・ボックスで、**C:\%HOMEPATH%\My Documents\Unified Functional Testing** に移動し、「Book Flights Parameter」という名前でテストを保存します。  
ソリューション・エクスプローラで、**Book Flights** テストが新しい **Book Flights Parameter** テストに置き換えられます。Book Flights テストは、今までと同様にファイル・システムに別に保存されています。

### 3. Book Flights テストをソリューションに再び追加します。

すべてのテストが同じソリューションから参照される場合は、それらのテストを同時に開いておくことができます。そうしておくで、テストの比較や編集を行うときに、交互に切り替えることができます。一度に実行できるテストは1つだけです。

- a. 「ファイル」>「追加」>「既存のテスト」を選択します。
- b. 「既存のテストの追加」ダイアログ・ボックスで、**C:\%HOMEPATH%\My Documents\Unified Functional Testing** に移動し、**Book Flights** テストを選択します。
- c. 「追加」をクリックしてテストをソリューションに追加します。  
Book Flights テストがソリューション・エクスプローラに再度表示されます。テストはアルファベット順に表示されるため、このテストは作成した Book Flights Parameter テストの上に表示されます。

パラメータ化用のテストの作成が済んだら、「練習 5b：データ・テーブル・パラメータの定義」(81ページ)に進み、テストのデータ・テーブル・パラメータを作成します。

## 練習 5b：データ・テーブル・パラメータの定義

このレッスンでは、出発地と到着地をパラメータとして定義し、テスト実行のたびに異なる出発地が使われるようにします。

### 1. UFT を起動し、必要に応じて Book Flights Parameter テストを開きます。

- a. UFT を開きます（「すべてのテスト用のソリューションの作成」(25ページ)を参照）。WPF Add-in がロードされていることを確認します。

- b. **【ファイル】 > 【開く】 > 【ソリューション】** を選択します。【ソリューションを開く】ダイアログ・ボックスが開きます。
- c. **C:\%HOMEPATH%\My Documents\Unified Functional Testing** にある **Flight Reservation Application.ftsln** ファイルに移動し、**【開く】** をクリックします。  
Flight Reservation ソリューションが開きます。これには、[「練習 5a：パラメータ化用のテストの作成」\(80ページ\)](#) で作成した Book Flights Parameter テストが含まれます。
- d. ソリューション・エクスプローラで、**【Book Flights Parameter】** ノードを選択します。


## 2. **【データ】表示枠が表示されていることを確認します。**

UFT ウィンドウの下部に【データ】表示枠が表示されていない場合は、**【表示】 > 【データ】** を選択します。

## 3. **Flight Finder アクションを開きます。**

- a. キャンバスで、**Flight Finder** アクションをダブルクリックします。Flight Finder アクションがドキュメント表示枠に別のタブとして表示されます。
- b. 必要に応じて、**【表示】 > 【キーワード ビュー】** を選択します。

## 4. **パラメータ化するテキストを選択します。**

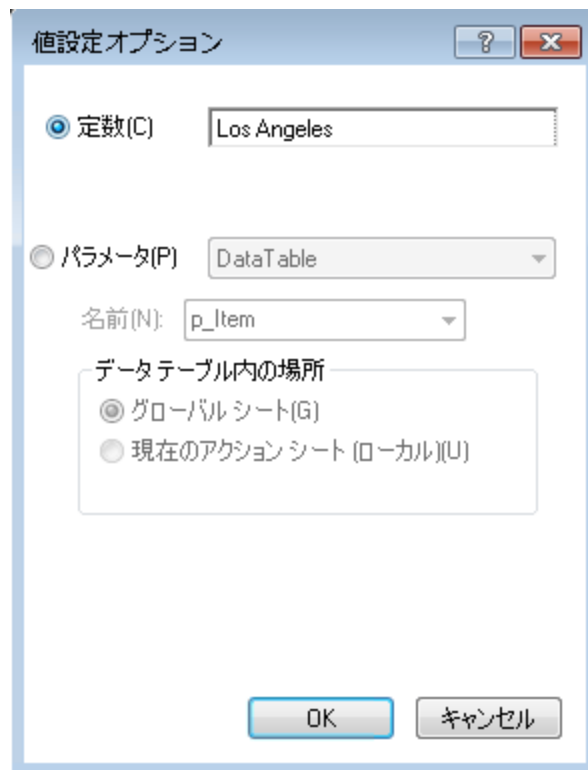
キーワード・ビューの **fromCity** 行で、**【値】** セルをクリックしてから、パラメータ化ボタン  をクリックします。

パラメータ・リストが表示されます。

テスト/アクション パラメータ (0)		DataTable (0)	Environment (21)	Random Number (0)
名前	値			
<div>+新規パラメータの追加</div>				

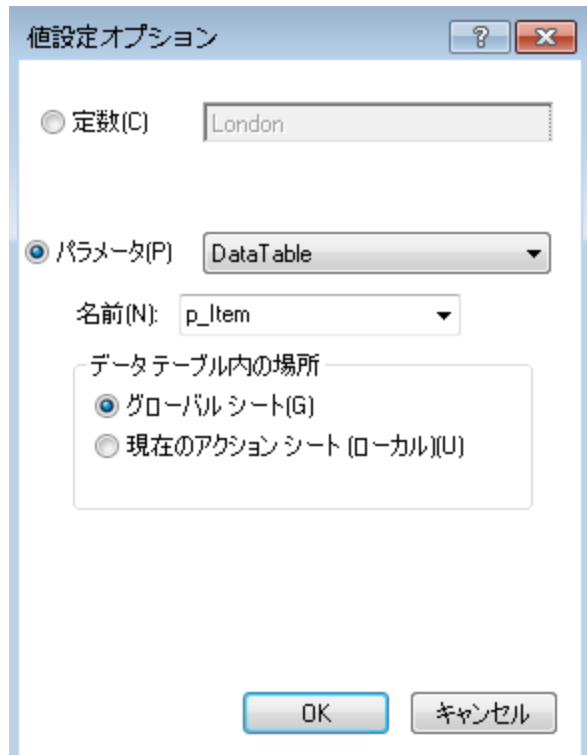
## 5. **パラメータ化のプロパティを設定します。**

- a. パラメータ・リストで、**【Data Table】** タブを選択します。こうすることで、定数値 (**London**) をパラメータで置き換えることができるようになります。  
データ・テーブル・パラメータをまだ作成していないため、パラメータは表示されていません。
- b. パラメータ・リストの **【Data Table】** タブで、**【新規パラメータの追加】** ボタンをクリックします。【値設定オプション】ダイアログ・ボックスが開きます。



- c. 【値設定オプション】ダイアログ・ボックスで、【パラメータ】ラジオ・ボタンを選択します。

- d. [パラメータ] ドロップダウン・メニューで [DataTable] オプションが選択されていることを確認します。これにより、パラメータの値が UFT の [データ] 表示枠から取得されます。[名前] ボックスが有効になり、**p\_Item** が表示されます。



- e. **p\_Item** パラメータを削除し、「**fromCity**」と入力します。
- f. [OK] をクリックして、ダイアログ・ボックスを閉じます。

**fromCity** パラメータが新規カラムとして [データ] 表示枠に追加され、Los Angeles（前の固定値）がカラムの最初の行に挿入されます。

**Los Angeles** が、アプリケーションのテスト実行中に UFT によって使用される複数の出発地のうちの最初の出発地となります。

項目	操作	値	注釈
HP MyFlight Sample Application			
fromCity	Select	DataTable("fromCity", dtGlobalS...	"fromCity" list から項目 <the 'fromCity' Data Table column の値> を選択す...
toCity	Select	"Sydney"	"toCity" list から項目 "Sydney" を選択する。
datePicker	SetDate	"3-Jul-2014"	"datePicker" calendar で日付または日付範囲を "3-Jul-2014" に設定する。
Class	Select	"Business"	"Class" list から項目 "Business" を選択する。
numOfTickets	Select	"2"	"numOfTickets" list から項目 "2" を選択する。
FIND FLIGHTS	Click		"FIND FLIGHTS" button をクリックする。
+ 新規ステップ			

キーワード・ビューのステップの表示が変わります。変更前のステップは「fromCity Select Los Angeles」と表示されていました。「値」セルをクリックすると、次の情報が表示されます。これは、fromCity というデータ表示枠パラメータを使って値がパラメータ化されていることを示します。

DataTable("fromCity", dtGlobalS...

## 6. toCity ステップ用のデータ・テーブル・パラメータを追加します。

前のステップで説明した手順を使用して、toCity オブジェクト用のデータ・テーブル・パラメータを toCity という名前で追加します。

完了すると、テストは次のようになります。

項目	操作	値	注釈
HP MyFlight Sample Application			
fromCity	Select	DataTable("fromcity", dtGlobalS...	"fromCity" list から項目 <the 'fromcity' Data Table column の値> を選択す...
toCity	Select	DataTable("toCity", dtGlobalShe...	"toCity" list から項目 <the 'toCity' Data Table column の値> を選択する。
datePicker	SetDate	"3-Jul-2014"	"datePicker" calendar で日付または日付範囲を "3-Jul-2014" に設定する。
Class	Select	"Business"	"Class" list から項目 "Business" を選択する。
numOfTickets	Select	"2"	"numOfTickets" list から項目 "2" を選択する。
FIND FLIGHTS	Click		"FIND FLIGHTS" button をクリックする。

## 7. テストを保存します。


【ファイル】 > 【保存】 を選択します。

続いて「練習 5c : データ・テーブルへのパラメータ値の追加」(85ページ)では、データ・テーブルにテスト実行で使用する値を入力する方法について学習します。

# 練習 5c : データ・テーブルへのパラメータ値の追加

「練習 5b : データ・テーブル・パラメータの定義」(81ページ)で学習したように、UFT では、【データ】表示枠にパラメータ値が表示されます。この練習では、(fromCity オブジェクトから)【データ】表示枠に別の出発地を追加し、そのデータを使用して UFT でアプリケーションをテストできるようにします。

## 1. UFT を起動し、必要に応じて Book Flights Parameter テストを開きます。

- UFT を開きます (「すべてのテスト用のソリューションの作成」(25ページ)を参照)。  
WPF Add-in がロードされていることを確認します。
- 【開く】 下向き矢印  をクリックし、【ソリューションを開く】を選択します。  
【ソリューションを開く】ダイアログ・ボックスが開きます。
- C:\%HOMEPATH%\My Documents\Unified Functional Testingにある Flight Reservation Application.ftsln ファイルに移動し、【開く】をクリックします。  
Flight Reservation ソリューションが開きます。これには、「練習 5a : パラメータ化用のテストの作成」(80ページ)で作成した Book Flights Parameter テストが含まれます。
- ソリューション・エクスプローラで、【Book Flights Parameter】ノードを選択します。

## 2. Flight Finder アクションを開きます。

ソリューション・エクスプローラで、【Flight Finder】アクションをダブルクリックします。

Flight Finder アクションがドキュメント表示枠に別のタブとして表示されます。

### 3. **[fromCity] カラムに追加する出発地を入力します。**

【データ】表示枠で、**fromCity** パラメータ用に次の内容を入力します。

行	値
2	Denver
3	Frankfurt
4	London

### 4. **toCity オブジェクト用のデータ・テーブル・パラメータと値を作成します。**

- a. 【データ】表示枠で、**B** カラムのヘッダ行をダブルクリックします。【パラメータ名の変更】ダイアログ・ボックスが開きます。

- b. 【パラメータ名の変更】ダイアログ・ボックスで、パラメータ名に「toCity」と入力し、【OK】をクリックします。

カラム・ヘッダ（以前の **B** カラム）に、入力したパラメータ名が反映されます。

	fromCity	toCity	C
1	Los Angeles		
2			
3			
4			
5			

Global Login Flight Finder

- c. **toCity** パラメータの値を、次のように入力します。


行	値
1	Sydney

2	Los Angeles
3	London
4	Frankfurt

2 つ目のパラメータと値の追加が完了すると、[データ] 表示枠は次のようになります。

	fromCity	toCity	C
1	Los Angeles	Sydney	
2	Denver	Los Angeles	
3	Frankfurt	London	
4	London	Frankfurt	
5			


## 5. toCity ステップをパラメータ化します。

- Flight Finder アクションの **toCity** 行で、[値] セルをクリックしてから、パラメータ化ボタン  をクリックします。[値設定オプション] ダイアログ・ボックスが開きます。
- [値設定オプション] ダイアログ・ボックスで、[パラメータ] ラジオ・ボタンを選択します。
- [パラメータ] タイプのドロップダウン・メニューで、[DataTable] を選択します。
- [データ テーブル内の場所] 領域で、[グローバル シート] を選択します。グローバル・データ・シートのパラメータを反映して、[名前] ドロップダウン・メニューが変更されます。
- [名前] ボックスで toCity パラメータを選択し、[OK] をクリックします。

キーワード・ビューで、**toCity** オブジェクトの [値] セルが更新され、パラメータ化されていることを示します。



## 6. テストを保存します。

ツールバーで [保存]  をクリックします。

パラメータと値を追加し、これらの値とテスト・ステップを関連付けたので、パラメータ化されたテストを実行する準備ができました。続いて「[練習 5d：パラメータ化されたテストの実行](#)」(87ページ)に進みます。

# 練習 5d：パラメータ化されたテストの実行


「[練習 5b：データ・テーブル・パラメータの定義](#)」(81ページ)および「[練習 5c：データ・テーブルへのパラメータ値の追加](#)」(85ページ)では、Flight Finder アクションの **toCity** オブジェクトと

**fromCity** オブジェクト用にデータ・テーブル・パラメータを作成しました。これにより、オブジェクトの固定値をテストのデータ・テーブルの可変値に置き換えることができます。

ただし、この時点でテストを実行すると、グローバル・データ・シートの最初の行のデータを使用して、テストが1回だけ実行されることになります。パラメータ化の目的は、さまざまなデータ・セットを使用してアプリケーションがどのように実行されるかを確認することであるため、テストを複数回実行するように UFT を設定する必要があります。

このレッスンでは、テスト全体を複数回実行してテストのデータ・テーブル内のデータを使用するように UFT とテストを構成します。

### 1. UFT を起動し、必要に応じて Book Flights Parameter テストを開きます。

- UFT を開きます（「すべてのテスト用のソリューションの作成」(25ページ)を参照）。WPF Add-in がロードされていることを確認します。
- 「開く」下向き矢印  をクリックし、[ソリューションを開く] を選択します。  
[ソリューションを開く] ダイアログ・ボックスが開きます。
- C:\%HOMEPATH%\My Documents\Unified Functional Testing にある **Flight Reservation Application.ftsln** ファイルに移動し、[開く] をクリックします。  
Flight Reservation ソリューションが開きます。これには、「練習 5a：パラメータ化用のテストの作成」(80ページ)で作成した Book Flights Parameter テストが含まれます。
- ソリューション・エクスプローラで、[Book Flights] テストのノードをクリックします。  
テスト・フローのキャンバスがドキュメント表示枠に別のタブとして表示されます。

### 2. 【記録と実行環境設定】を変更してフライト予約アプリケーションが自動的に開かないようにします。

Book Flights テスト（このテストを作成する元になったテスト）では、テスト実行の開始時にフライト予約アプリケーションを自動的に開くように【記録と実行環境設定】を構成しました。このテスト実行では、UFT でテスト・ステップの一部としてアプリケーションを開く必要があります。

- 【記録】 > 【記録と実行環境設定】を選択します。【記録と実行環境設定】ダイアログ・ボックスが開きます。
- 【記録と実行環境設定】ダイアログ・ボックスの【Windows Applications】タブで、[開かれている Windows ベースのアプリケーションすべてでテストを記録して実行する] オプションを選択し、[適用] をクリックします。
- [OK] をクリックして、ダイアログ・ボックスを閉じます。


### 3. アプリケーションを開く/閉じるためのアクションを追加します。

パラメータ化されたテストを実行する場合、UFT はデータ・テーブル内の行数に応じてテスト全体を複数回実行します。ただし、これを行うには、アプリケーションを開くステップと閉じるステップを追加して、アプリケーションの4つのページ（**Login**、**Flight Finder**、**Select Flight**、**Flight Details/Confirmation**）のそれぞれでこれらのアクションが実行されるようにする必要があります。

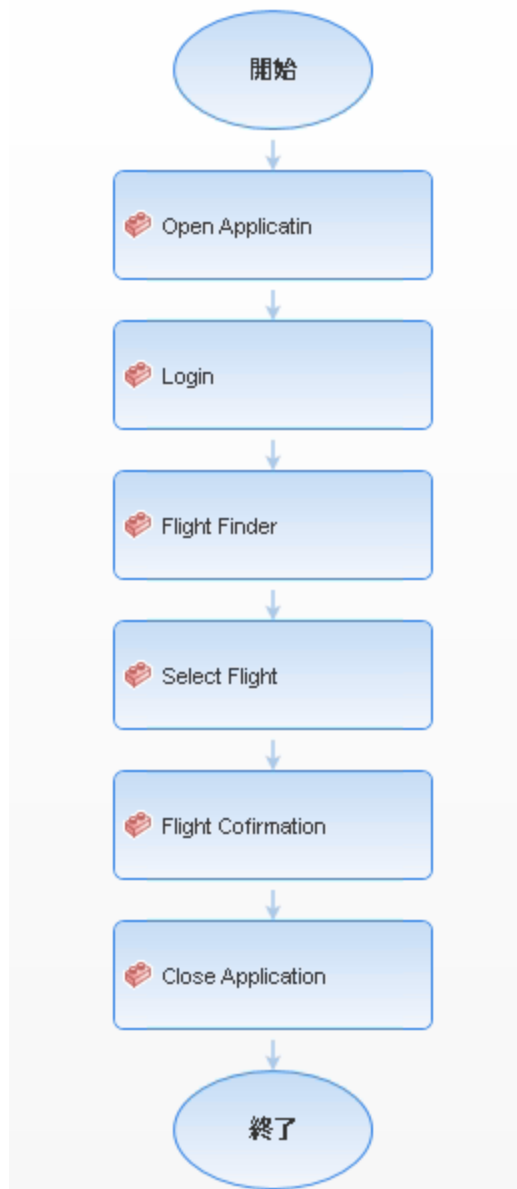


- a. ドキュメント表示枠で、**[Book Flights]** タブ (テスト・フロー・キャンバス) を選択します。
- b. ツールバーで、**[新規アクションの呼び出しの挿入]** ボタン  をクリックします。  
[新規アクションの呼び出しの挿入] ダイアログ・ボックスが開きます。
- c. [新規アクションの呼び出しの挿入] ダイアログ・ボックスで、新規アクションに「Open Application」という名前を付けます。その他の設定とオプションはすべて標準設定のままにします。  
**Open Application** という名前の新規アクションのブロックがテスト・フローの末尾に追加されます。
- d. [Book Flights] タブ (テスト・フロー・キャンバス) で、**Open Application** アクションを右クリックして **[上に移動]** を選択します。Open Application アクションのブロックが Flight Confirmation アクションの上に移動します。
- e. Open Application ブロックがテストの最初のアクションになるまで、右クリックして **[上に移動]** を選択する操作を行います。

**注:** 必要に応じて、テスト・フロー内でアクションのブロックをドラッグアンドドロップすることもできます。

- f. ツールバーで、**[新規アクションの呼び出しの挿入]** ボタン  をクリックします。
- g. [新規アクションの呼び出しの挿入] ダイアログ・ボックスで、新規アクションに「Close Application」という名前を付けます。その他の設定とオプションはすべて標準設定のままにします。

この2つの新規アクションの挿入が完了すると、テスト・フローは次のようになります。



#### 4. アプリケーションを開く/閉じるためのステートメントを追加します。

Book Flights テストを作成したときには、実行の設定を使用してアプリケーションを自動的に開くように UFT を設定しました。このテストでは、アプリケーションを開く操作と閉じる操作を別のステップとして追加する必要があります。これを行うには、**SystemUtil** ステートメントを使用します。

- ソリューション・エクスプローラで、**[Open Application]** アクション・ノードをダブルクリックします。Open Application アクションがドキュメント表示枠に別のタブとして表示されます。
- [表示] > [エディタ]** を選択してエディタを開きます。

- c. エディタで、次の行を貼り付けます。

```
SystemUtil.Run "C:\Program Files (x86)\HP\Unified Functional Testing\samples\Flights  
Application\FlightsGUI.exe"
```

**注:** この行をチュートリアルの PDF ファイルから貼り付ける場合は、貼り付けたテキストを編集し、このメソッドが1つの行に収まるようにしてください。

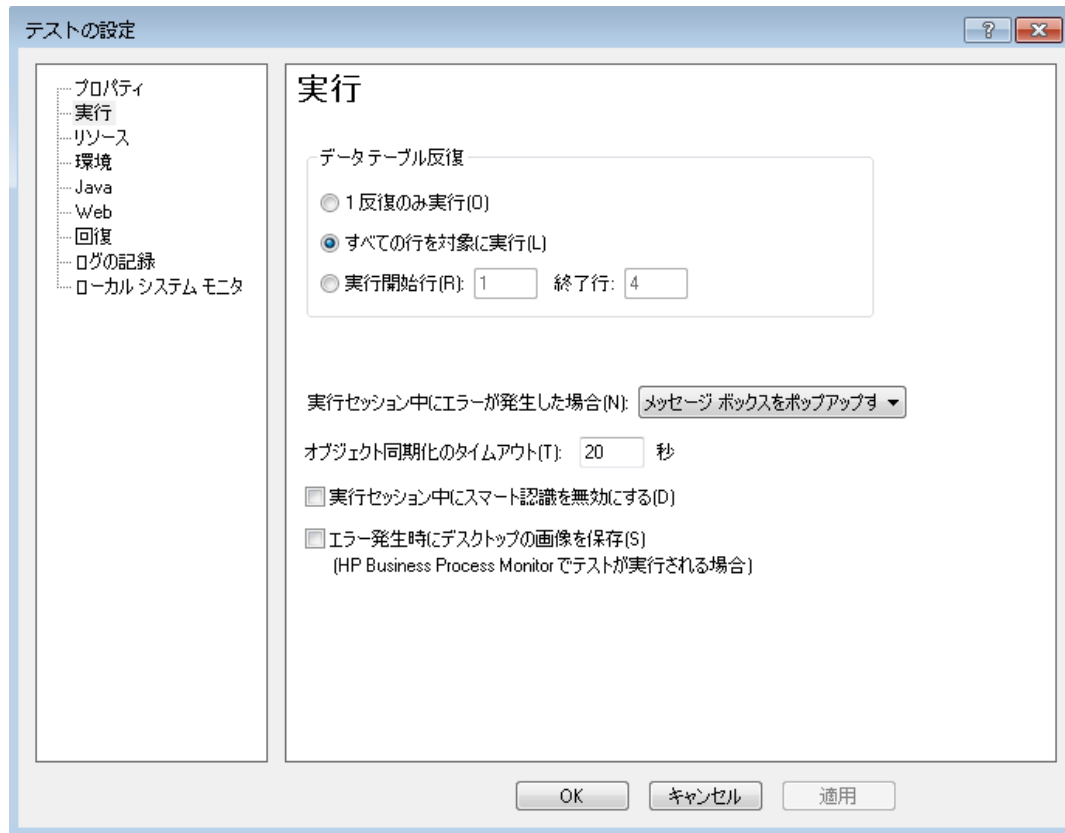
- d. ソリューション・エクスプローラで、**[Close Application]** アクション・ノードをダブルクリックします。Close Application アクションがドキュメント表示枠に別のタブとして表示されます。
- e. エディタで、次の行を貼り付けます。

```
SystemUtil.CloseDescendentProcesses
```

5. **データ・テーブル内の各行に対して反復実行するように UFT を設定します。**


- a. **[ファイル]** > **[設定]** を選択します。**[テストの設定]** ダイアログ・ボックスが開きます。
- b. **[テストの設定]** ダイアログ・ボックスで、**[実行]** ノードを選択します。

- c. **【データ テーブル反復】** セクションで、**【すべての行を対象に実行】** オプションを選択します。これにより、グローバル・データ・シート内の行ごとにテストの反復が実行されます。



これでテストを実行したときに、グローバル・データ・シートの4つの行に対応して、UFTでテストの反復が複数回実行されるようになります。

## 6. Book Flights Parameter テストを実行します。

- 【実行】** ボタン  をクリックします。【実行】ダイアログ・ボックスが開きます。
- 【実行】ダイアログ・ボックスの**【結果保管場所】** タブで、**【新規実行結果フォルダ】** を選択し、標準設定フォルダ名をそのまま使用します。
- 【OK】** をクリックします。テストの実行が完了すると、実行結果が開きます。

## 7. 実行結果を分析します。

Run Results Viewer で、結果ツリーの最上位ノードを右クリックし、**【すべて展開】** を選択します。

結果にはテストの各反復に対する異なる 5 つのノードが表示されます。これはグローバル・データ・シートの複数の行に対応しています。



Flight Finder サマリ・ノードの下で **fromCity.Select** または **toCity.Select** ステップを参照すると、データ・テーブル内の値に合わせてステップの詳細が変更されていることがわかります。

#### 8. 実行結果を閉じます。

ドキュメント表示枠で、実行結果を含むタブを閉じます。

以上でデータを使用してテストをパラメータ化する方法についての学習は終了です。続いて「[レッスン 6：チェックポイントおよび出力値の作成](#)」(94ページ)では、チェックポイントおよび出力値を使用してテストを拡張する方法について学習します。

# レッスン 6：チェックポイントおよび出力値の作成

「[レッスン 4：GUI テストの実行と分析](#)」(73ページ)では、それまでのレッスンで作成したテストを実行し、一連のステップがフライト予約アプリケーションでスムーズに正しく実行されるかどうかを検査しました。

基本的なテスト・ステップを作成したら、拡張機能としてテストおよびテスト・ステップに対してチェックポイントと出力値を追加できます。チェックポイントでは、テストの実行中に、アプリケーション内で期待どおりの情報が表示されるかどうかを検証します。出力値では、テスト内の他の場所で使用する値をパラメータとしてエクスポートします。

このレッスンでは、チェックポイントを挿入し、関数を使用してフライト予約アプリケーションのチェックポイントのオブジェクトの妥当性を検査します。

このレッスンには、次の手順が含まれています。

- [チェックポイントおよび出力値のタイプについて](#) .....95
- [練習 6a：チェックポイント・テストの作成](#) .....97
- [練習 6b：オブジェクトの値の検査](#) .....98
- [練習 6c：テーブルの値の検査](#) .....101
- [練習 6d：テキストの値の検査](#) .....107
- [練習 6e：オブジェクト・リポジトリ内のチェックポイントの管理](#) .....113
- [練習 6f：チェックポイントがあるテストの実行および分析](#) .....115
- [練習 6g：出力値のテストの作成](#) .....118
- [練習 6h：出力値のステップの追加](#) .....119

## チェックポイントおよび出力値のタイプについて

UFT では、**チェックポイント**を挿入してアプリケーションが正しく実行されていることを確認できます。これらのチェックポイントは、テスト・フロー全体の中で個別のテスト・ステップとして実行されます。また、**出力値**を使用し、特定のステップやオブジェクトで生成された値を取得して別のステップに渡すことができます。

### チェックポイント

チェックポイントを使用すると、さまざまなアプリケーション・オブジェクトを検査できます。

オブジェクト・タイプ	チェックポイントの説明	使用例
標準	オブジェクトのプロパティの値を検査します。	ラジオ・ボタンが選択されているかどうかを検査します。
画像	画像のプロパティ値を検査します。  画像を検査するには、 <b>【標準チェックポイント】</b> オプションを選択してから画像オブジェクトを選択します。	画像のソース・ファイルが正しいかどうかを検査します。
テーブル	テーブル内の情報を検査します。  テーブルを検査するには、 <b>【標準チェックポイント】</b> オプションを選択してからテーブル・オブジェクトを選択します。	テーブルのセルの値が正しいかどうかを検査します。
ページ	Web ページの特性について検査します。  テーブルを検査するには、 <b>【標準チェックポイント】</b> オプションを選択してからブラウザで Web ページを選択します。	Web ページをロードするのにかかる時間、または Web ページに壊れたリンクが含まれていないかなどを検査できます。
テキスト	テキスト文字列がアプリケーションの正しい場所に表示されるかどうかを検査します。	テキスト文字列がテスト・オブジェクトの期待される場所に表示されるかどうかを検査します。
テキスト領域	テキスト文字列が Windows ベース・アプリケーションの定義された領域に表示されるかどうかを検査します。	アプリケーションの別の部分に入力されたテキストがダイアログ・ボックスの領域に表示されるかどうかを検査します。
ビットマップ	アプリケーションの領域をビットマップとしてキャプチャしてから	Web ページ（またはその一部）が期待どおりに表示されているかどうかを検査します。

	検査します。	
<b>データベース</b>	Web サイトやアプリケーションがアクセスするデータベースの内容を検査します。	データベース・クエリの値が正しいかどうかを検査します。
<b>アクセシビリティ</b>	第 508 条に準拠しているかどうかを検査するため、Web サイトの領域を特定します。	W3C の「Web Content Accessibility Guidelines」で規定されている ALT 属性が Web ページの画像に含まれているかどうかを検査できます。
<b>ファイル内容</b>	実行セッション中に生成またはアクセスされたドキュメントのテキストを検査します。	動的に生成された PDF ファイルのヘッダに、地域企業本社の連絡先情報が表示されるかどうかを検査します。
<b>XML</b>	XML 文書のデータ内容を検査します。	要素の内容を検査して、タグ、属性、値が変更されていないことを確認します。  <b>注:</b> XML ファイル・チェックポイントは、指定された XML ファイルを検査するのに使用します。XML アプリケーション・チェックポイントは、Web ページ内の XML 文書を検査するのに使用します。

## 出力値

さまざまなタイプの出力値を使用できます。

オブジェクトの種類	説明	例
<b>標準</b>	アプリケーション内の大半のオブジェクトから値を取得して保存します。	編集フィールドの文字列出力を取得します。
<b>ファイル内容</b>	選択したファイルまたは選択したファイルの一部の出力を取得します。	HTML ページの出力を取得します。
<b>テーブル</b>	テーブル・オブジェクトのセルまたは選択したセルの出力を取得します。	テーブル・オブジェクトの行 1、列 1 のセルの出力を取得します。
<b>テキスト/テキスト領域</b>	アプリケーション内のオブジェクトまたは領域のテキスト出力を取得します。	エラー・メッセージのテキスト出力を取得します。
<b>データベース</b>	データベース・セルまたは選択したデータベース・セルの出力を取得します。	アプリケーション内のオブジェクトによってアクセスされるデータベースの出力を取得します。
<b>XML</b>	XML 文書内に含まれる要素の出力を取得します。	製品の価格を定義する XML の <price> 属性の出力を取得します。

ほとんどのチェックポイントと出力値は、キーワード・ビューまたはエディタでのステップの編集画面または記録中に追加できます。以下の練習では、上記の一部のチェックポイントの作成方法について説明します。

UFT によって作成されるチェックポイントまたは出力値の名前は、チェック対象の値など、チェックポイントまたは出力値内部の情報に基づいて割り当てられます。基となった情報が変更されても、チェックポイントまたは出力値の名前は変わりません。キーワード・ビューでチェックポイントまた



は出力値を探すときには、この点に留意してください。また、キーワード・ビューに表示されている名前は、UFT によって短くされている場合があります。

チェックポイントおよび出力値の詳細については、『HP Unified Functional Testing ユーザーズ・ガイド』を参照してください。

続いて「[練習 6a : チェックポイント・テストの作成](#)」(97ページ)に進み、チェックポイントを使用するテストを作成します。


出力値を追加する場合は、「[練習 6g : 出力値のテストの作成](#)」(118ページ)に進みます。

## 練習 6a : チェックポイント・テストの作成

このレッスンでは、「[レッスン 4 : GUI テストの実行と分析](#)」(73ページ)で実行した Book Flights テストを新しいテストとして保存し、チェックポイントを作成します。

**注:** チェックポイントを別のテストで管理する必要はありません。これから新しいテストを作成するのは、チュートリアルで使用するためです。通常の作業の場合、任意のテストにチェックポイントを追加できます。

### 1. UFT を起動し、Book Flights テストを開きます。

- UFT が開いていない場合は、「[すべてのテスト用のソリューションの作成](#)」(25ページ)の手順に沿って開きます。WPF Add-in がロードされていることを確認します。
- 「**開く**」ボタンの下向き矢印  をクリックし、「**ソリューションを開く**」を選択します。「**ソリューションを開く**」ダイアログ・ボックスが開きます。
- C:\%HOMEPATH%\My Documents\Unified Functional Testing** にある **Flight Reservation Application.ftsln** ファイルに移動し、「**開く**」をクリックします。  
Flight Reservation Application ソリューションが開きます。これには、「[レッスン 1 : GUI テストとアクションの作成](#)」(29ページ)で作成した **Book Flights** テストが含まれます。

### 2. テストを Book Flights Checkpoint として保存します。

- ソリューション・エクスプローラで、「**Book Flights**」テスト・ノードを右クリックし、「**名前を付けて保存**」を選択します。
- 「**テストに名前を付けて保存**」ダイアログ・ボックスで、**C:\%HOMEPATH%\My Documents\Unified Functional Testing** ディレクトリに移動し、「**Book Flight Checkpoint**」という名前でテストを保存します。  
ソリューション・エクスプローラで、Book Flights テストが **Book Flights Checkpoint** テストに置き換えられます。Book Flights テストは、今までと同様にファイル・システムに別に保存されています。

### 3. Book Flights テストをソリューションに再び追加します。

Book Flights テストと Book Flights Checkpoint テストが同じソリューションに含まれる場合、両方のテストを同時に開いておくことができます。こうしておく、テストの比較や編集を行うときに、交互に切り替えることができます。

**注:** 一度に実行できるテストは 1 つだけです。

- a. **【ファイル】 > 【追加】 > 【既存のテスト】** を選択します。
- b. **C:\%HOMEPATH%\My Documents\Unified Functional Testing** にある **Book Flights** テストに移動し、**【追加】** をクリックします。

Book Flights テストがソリューション・エクスプローラに再度追加されます。ソリューション・エクスプローラでは、テストがアルファベット順に表示されることに注意してください。

ソリューションが自動的に保存されます。

続いて「[練習 6b：オブジェクトの値の検査](#)」(98ページ)では、チェックポイントの挿入を開始します。

## 練習 6b：オブジェクトの値の検査

このレッスンでは、「[練習 6a：チェックポイント・テストの作成](#)」(97ページ)で作成したテストに標準チェックポイントを追加します。このチェックポイントは、**【Flight Details】** ウィンドウの **【Passenger Name】** フィールドに入力された値を検証します。

**注:** チェックポイントを挿入するには、事前にフライト予約アプリケーションの **【Flight Details】** ページを開いておく必要があります。

### 1. UFT を起動し、Book Flights Checkpoint テストを開きます。

- a. 必要に応じて、UFT を開きます（「[すべてのテスト用のソリューションの作成](#)」(25ページ)を参照）。WPF Add-in がロードされていることを確認します。
- b. 開始ページの **【最近のソリューション】** 領域で、**【Flight Reservation Application】** をクリックします。

Flight Reservation ソリューションが開きます。これには、「[練習 6a：チェックポイント・テストの作成](#)」(97ページ)で作成した Book Flights Parameter テストが含まれます。

- c. ソリューション・エクスプローラで、**【Book Flights Checkpoint】** ノードをダブルクリックします。

Book Flights Checkpoint テストがドキュメント表示枠に別のタブとして表示されます。

### 2. チェックポイントを追加するアクションを表示します。


乗客の名前がテストによって自動的に入力された後に [Passenger Name] エディット・ボックスのプロパティ値をチェックするチェックポイントを追加するには、このチェックポイントをテスト内の適切なアクションに追加する必要があります。

キャンバスで、[Flight Confirmation] アクションをダブルクリックして開きます。

### 3. フライト予約アプリケーションの [Flight Details] ページを開きます。

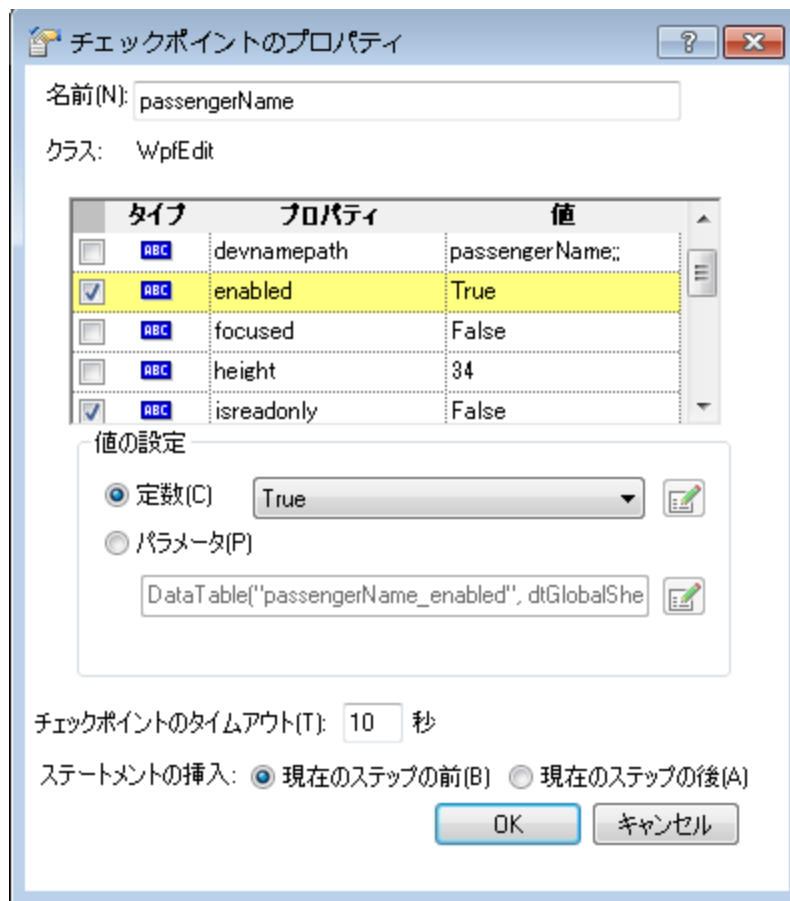
- a. 「[フライト予約アプリケーションの確認](#)」(23ページ)の説明に従って、フライト予約アプリケーションを開きます。
- b. ログイン情報を入力します。
  - **ユーザ名:** john
  - **パスワード:** hp
- c. [OK] をクリックします。[Flight Finder] ページが開きます。
- d. フライト検索の詳細を入力します。
  - **Departure City:** Los Angeles
  - **Arrival City:** Sydney
  - **Date:** 明日の日付
  - **Class:** Business
  - **Tickets:** 2
- e. [Find Flights] ボタンをクリックします。「SELECT FLIGHT」ページが開きます。
- f. [Select Flight] ページで、最初の行を選択して [Select Flight] をクリックします。  
[Flight Details] ページが開きます。

### 4. 標準チェックポイントを作成します。

- a. エディタが表示されている場合は、[キーワード ビューとエディタ間の切り替え] ボタン  をクリックして、キーワード・ビューを表示します。
- b. キーワード・ビューで、グリッドの右余白をクリックして [passengerName] 行を選択します。

**注:** [項目] カラム内をクリックしないでください (オブジェクトのみが選択されます)。チェックポイントを追加するには、ステップ全体を選択する必要があります。

- c. [デザイン] > [チェックポイント] > [標準チェックポイント] を選択します。[チェックポイントのプロパティ] ダイアログ・ボックスが表示されます。



このダイアログ・ボックスには、passengerName オブジェクトのプロパティが表示されます。

- [名前] はアプリケーション内で定義されているオブジェクトの名前です。この場合の名前は **passengerName** です。
- [クラス] はオブジェクトのタイプです。この場合のタイプは **WpfEdit** で、オブジェクトのタイプがエディット・ボックスであることを意味します。
- [タイプ] カラムの [ABC] アイコンは、プロパティの値が定数であることを示します。

チェックポイントを挿入する場合、UFT で各オブジェクト・クラスの標準のプロパティ検査が推奨されます。


プロパティ	値	説明
enabled	True	オブジェクトが現在有効になっているかどうかをチェックします。
isreadonly	False	エディット・ボックスに情報を入力できるかどうかを

		チェックします。現在、テキスト文字列の入力を許可するようにオブジェクトが設定されています。
<b>text</b>	標準設定の指定なし	オブジェクトに入力されたテキストをチェックします。現在、値は空です。このアクションの最初のステップで <b>[passengerName]</b> エディット・ボックスに指定した値と同じ値を入力する必要があります。

- d. [チェックポイントのプロパティ] ダイアログ・ボックスの[名前]ボックスに、新規チェックポイント名として「CheckName」と入力します。
- e. オブジェクト・プロパティの領域をスクロール・ダウンし、「**text**」というプロパティ名を含む行を選択します。選択した行が選択されていることを示す黄色に変わります。
- f. [**text**] プロパティ行で、[値] カラム内をクリックします。
- g. オブジェクト・プロパティのグリッドの下[値の設定]領域で、[定数] ラジオ・ボタンをクリックします。
- h. [定数] 値エディット・ボックスに、「John Smith」と入力します。（これは、アクションの最初のステップで [Passenger Name] ボックスに入力した値の名前です。）この値はオブジェクト・プロパティのグリッドでも更新されます。
- i. [チェックポイントのプロパティ] ダイアログ・ボックスの下部にある[ステートメントの挿入]領域で、[現在のステップの後]を選択します。これにより、**passengerName Set** ステップの後にチェックポイントが挿入されます
- j. その他の設定については標準の設定をそのまま受け入れて[OK]をクリックします。  
標準チェックポイント・ステップがテストの選択したステップの下に追加されます。

項目	操作	値	注釈
HP MyFlight Sample Application			
passengerName	Set	"John Smith"	"passengerName" edit box に "John Smith" ...
passengerName	Check	CheckPoint("CheckName")	選択されたプロパティで "passengerName" e...
ORDER	Click		"ORDER" button をクリックする。
progBar	WaitProperty	"value","100"	次のステップに進む前に、"progBar" progre...
Order 88 completed	Check	CheckPoint("CheckOrderCom...	"Order 88 completed" object にあるテキスト...
NEW SEARCH	Click		"NEW SEARCH" button をクリックする。
+ 新規ステップ			

## 5. テストを保存します。

ツールバーで[保存]  をクリックします。


この手順を使用すると、さまざまなタイプのチェックポイントを挿入できます。続いて「[練習 6c：テーブルの値の検査](#)」(101ページ)で、アプリケーション内のテーブル・オブジェクトをチェックする方法を学習します。

## 練習 6c：テーブルの値の検査

「[練習 6b：オブジェクトの値の検査](#)」(98ページ)では、アプリケーション内のオブジェクトに対してチェックポイントを追加しました。このレッスンでは、テストにテーブル・チェックポイントを追加

します。テーブル・チェックポイントでは、[Select Flights] ページのフライト・グリッド内の値を検査します。

### 1. UFT を起動し、Book Flights Checkpoint テストを開きます。

- a. 必要に応じて、UFT を開きます（「すべてのテスト用のソリューションの作成」(25ページ)を参照）。WPF Add-in がロードされていることを確認します。
- b. [開く] ボタンの下向き矢印  をクリックし、[ソリューションを開く] を選択します。[ソリューションを開く] ダイアログ・ボックスが開きます。
- c. [ソリューションを開く] ダイアログ・ボックスで、**C:\%HOMEPATH%\My Documents\Unified Functional Testing** にある **Flight Reservation Application.ftsln** ファイルに移動し、[開く] をクリックします。  
Flight Reservation Application ソリューションが開きます。これには、「練習 6a：チェックポイント・テストの作成」(97ページ)で作成した Book Flights Parameter テストが含まれます。
- d. ソリューション・エクスプローラで、[Book Flights Checkpoint] ノードをダブルクリックします。

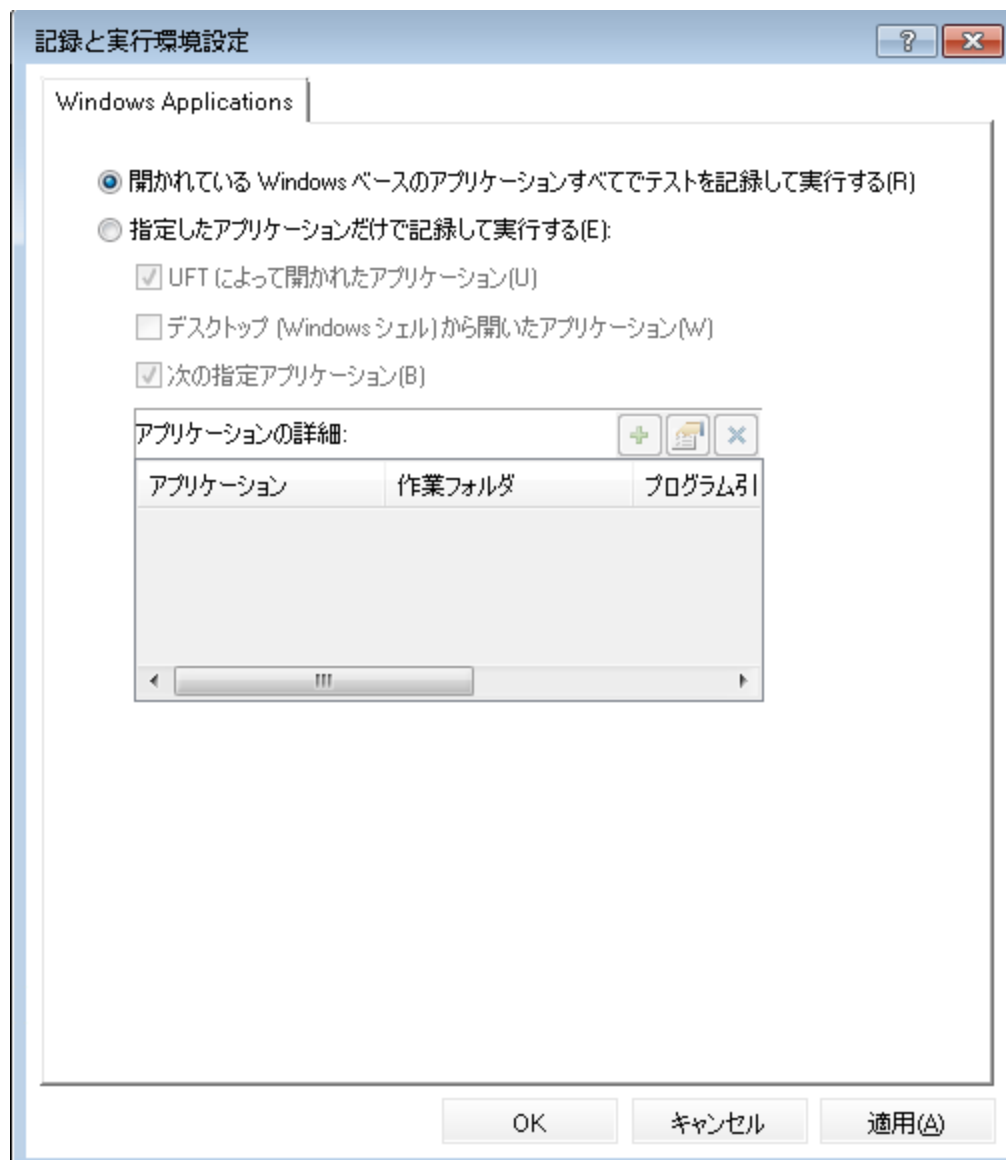
### 2. テーブル・チェックポイントを追加するステップを指定します。

- a. Select Flight アクションがまだ開いていない場合は、ソリューション・エクスプローラで [Select Flight] アクション・ノードをダブルクリックします。アクションがドキュメント表示枠に別のタブとして表示されます。
- b. エディタが表示されている場合は、[表示] > [キーワード ビュー] を選択して、キーワード・ビューを表示します。
- c. キーワード・ビューで、**flightsDataGrid** ステップ（予約するフライトを選択するステップ）を選択します。

### 3. フライト予約アプリケーションの [Select Flight] ページを開きます。



- a. 「フライト予約アプリケーションの確認」(23ページ)の説明に従って、フライト予約アプリケーションを開きます。
- b. ログイン情報を入力します。
  - **ユーザ名:** john
  - **パスワード:** hp
- c. [OK] をクリックします。[Flight Finder] ページが開きます。

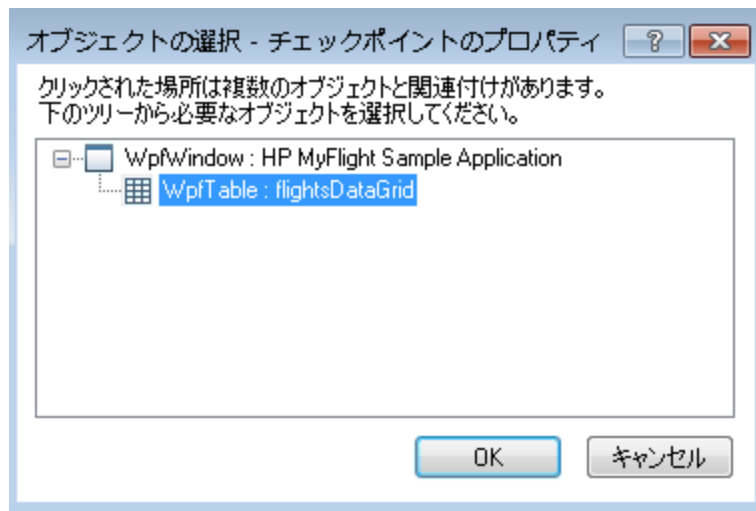
- d. フライト検索の詳細を入力します。
- **Departure City:** Los Angeles
  - **Arrival City:** Sydney
  - **Date:** 明日の日付
  - **Class:** Business
  - **Tickets:** 2
- e. **[Find Flights]** ボタンをクリックします。「SELECT FLIGHT」ページが開きます。
4. **開いているアプリケーション・ページを記録するように UFT を設定します。**
- a. UFT で、**[記録] > [記録と実行環境設定]** を選択します。**[記録と実行環境設定]** ダイアログ・ボックスが開きます。



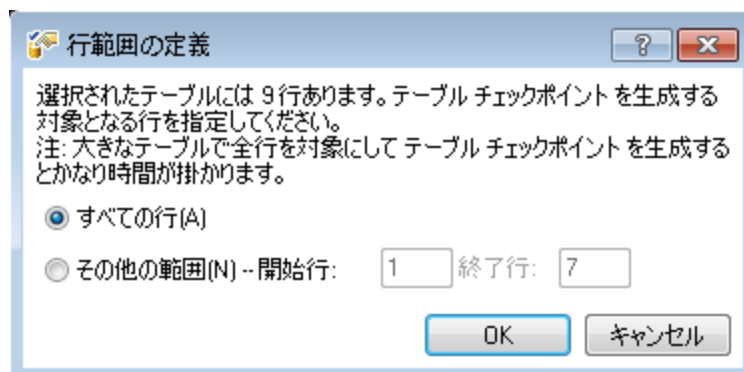
- b. [Windows Applications] タブで、**【開かれている Windows ベースのアプリケーションすべてでテストを記録して実行する】** オプションを選択します。
- c. **【OK】** をクリックして、ダイアログ・ボックスを閉じます。

## 5. テーブル・チェックポイントを作成します。

- a. ツールバーで、**【記録】** ボタン  をクリックします。UFT が記録セッションを開始し、UFT のメイン・ウィンドウが非表示になります。
- b. 記録ツールバーで、**【チェックポイントまたは出力値の挿入】** 下矢印  をクリックし、**【標準チェックポイント】** を選択します。マウス・ポインタが指差し型に変わります。
- c. フライト予約アプリケーションで、フライト・テーブルをクリックします。**【標準チェックポイント】** オブジェクト選択ダイアログ・ボックスが表示されます。

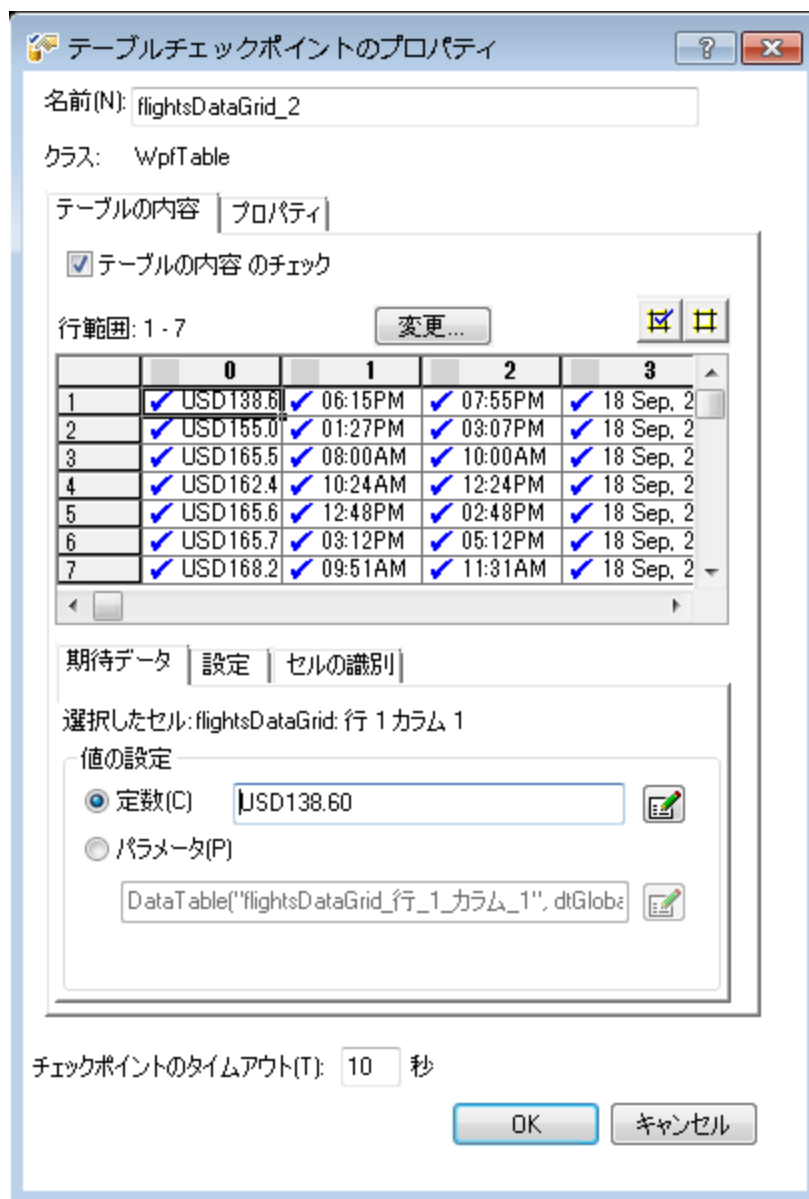


- d. **【標準チェックポイント】** 選択ダイアログ・ボックスで、**【WpfTable: flightsDataGrid】** オブジェクトを選択し、**【OK】** をクリックします。**【行範囲の定義】** ダイアログ・ボックスが開きます。





- e. [行範囲の定義] ダイアログ・ボックスで, [すべての行] ラジオ・ボタンを選択し, [OK] をクリックします。[テーブルチェックポイントのプロパティ] ダイアログ・ボックスが開きます。



標準では, すべてのセルにチェック・マークが入ります。セルをダブルクリックしてセルの選択または選択解除を行うか, 行またはカラムのヘッダをダブルクリックして選択した行やカラムにあるすべてのセルの選択または選択解除を行うことができます。

- f. [テーブルチェックポイントのプロパティ] ダイアログ・ボックスで, [名前] ボックスに新しいチェックポイントの名前として「CheckCost」と入力します。

- g. グリッドで各カラムのヘッダをダブルクリックして、すべてのチェック・マークを外します。

**注:** テーブル・オブジェクトのすべてのカラムを表示するには、グリッドで右方向にスクロールする必要があります。

- h. グリッドで行 1, カラム 0 をダブルクリックしてセルを選択します( UFT は、チェック・マークが入ったセルのみを検査します)。

	0	1	2	3
1	<input checked="" type="checkbox"/> USD138.6	06:15PM	07:55PM	18 Sep, 2
2	USD155.0	01:27PM	03:07PM	18 Sep, 2
3	USD165.5	08:00AM	10:00AM	18 Sep, 2
4	USD162.4	10:24AM	12:24PM	18 Sep, 2
5	USD165.6	12:48PM	02:48PM	18 Sep, 2
6	USD165.7	03:12PM	05:12PM	18 Sep, 2
7	USD168.2	09:51AM	11:31AM	18 Sep, 2

**注:** テーブルに表示されるデータは日付の影響を受けます。このチェックポイントを作成してしばらくしてから別の日に再度このテストを実行する場合は、チェックポイントをパスできるようにするため、上記の手順でこのチェックポイントを更新する必要があります。

- i. 行とカラムをスクロールし、行 1, カラム 0 のセルだけがチェックされていることを確認します。ほかにチェックされているセルがある場合は、ダブルクリックしてチェックを外します。
- j. その他の設定については標準の設定をそのまま受け入れて **[OK]** をクリックします。


## 6. 記録セッションを停止します。

記録ツールバーで、**[停止]**  をクリックします。

テーブル・オブジェクトのチェックポイントのプロパティの定義が完了し、テーブル・チェックポイント・ステップがテストに追加されました。これは、キーワード・ビューの flightsDataGrid オブジェクト・ステップの下に新規ステップとして表示されます。

項目	操作	値	割り当て	コメント	注釈
HP MyFlight Sample Application					
flightsDataGrid	SelectCell	2,2			"flightsDataGrid" table で行 2, カラム 2
flightsDataGrid	Check	CheckPoint("flightsDataGrid")			Check whether the content of specific...
SELECT FLIGHT	Click				"SELECT FLIGHT" button をクリックする。
+ 新規ステップ					

## 7. テストを保存します。

ツールバーで **[保存]**  をクリックします。

これで、テーブル・オブジェクトのチェックポイントの追加が完了しました。続いて「[練習 6d：テキストの値の検査](#)」(107ページ)でチェックポイントを追加します。

## 練習 6d：テキストの値の検査

前の練習では、通常のテスト・オブジェクトとテーブル・オブジェクトにチェックポイントを追加しました。このオブジェクトでは、テキスト・チェックポイントをテストに追加し、注文プロセスの終了時に表示されるオブジェクト内のテキストを検査します。

### 1. UFT を起動し、Book Flights Checkpoint テストを開きます。


- 必要に応じて、UFT を開きます（「[すべてのテスト用のソリューションの作成](#)」(25ページ)を参照）。WPF Add-in がロードされていることを確認します。
- 開始ページの「最近のソリューション」領域で、**[Flight Reservation Application]** ソリューションをクリックします。

Flight Reservation Application ソリューションが開きます。これには、「[練習 6a：チェックポイント・テストの作成](#)」(97ページ)で作成した Book Flights Parameter テストが含まれます。

- ソリューション・エクスプローラで、**[Book Flights Checkpoint]** ノードをダブルクリックします。

Book Flights Checkpoint テストがドキュメント表示枠に別のタブとして表示されます。

### 2. テキスト・チェックポイントを追加するステップを指定します。

- ソリューション・エクスプローラで、**[Flight Confirmation]** アクション・ノードをダブルクリックします。Flight Confirmation アクションがドキュメント表示枠に別のタブとして表示されます。
- エディタが開いている場合は、**[キーワード ビュー]** ボタン  をクリックして、キーワード・ビューを表示します。
- キーワード・ビューで、**progBar** ステップ（すべてのステップを完全に展開している場合は、最後から 2 つ目の行）を強調表示します。

### 3. フライト予約アプリケーションの [Flight Details] ページを開きます。

- 「[フライト予約アプリケーションの確認](#)」(23ページ)の説明に従って、フライト予約アプリケーションを開きます。
- ログイン情報を入力します。
  - ユーザ名:** john
  - パスワード:** hp
- [OK]** をクリックします。[Flight Finder] ページが開きます。

d. フライト検索の詳細を入力します。

- **Departure City** : Los Angeles
- **Arrival City** : Sydney
- **Date**: 明日の日付
- **Class**: Business
- **Tickets** : 2

e. **[Find Flights]** ボタンをクリックします。「SELECT FLIGHT」ページが開きます。

f. [Select Flight] ページで、最初の行を選択して **[Select Flight]** をクリックします。  
[Flight Details] ページが開きます。

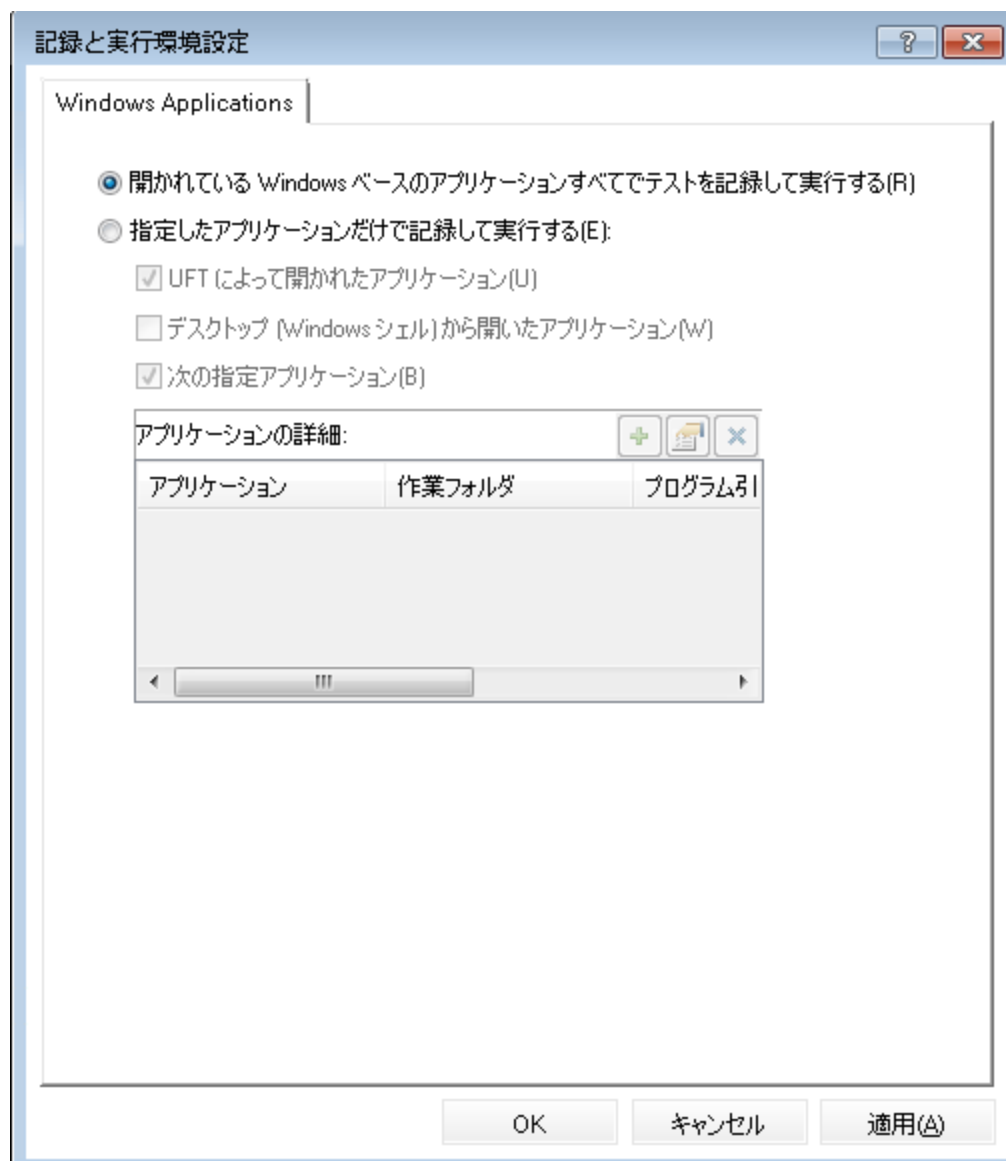
g. [Flight Details] ページの **[Passenger Name]** ボックスに、「John Smith」と入力して  
**[ORDER]** をクリックします。

ウィンドウの中央部分に注文の完了を示すボックスが表示されます。アプリケーションはこのままにしておきます。

#### 4. **開いているアプリケーション・ページを記録するように UFT を設定します。**



a. UFT で、**[記録]** > **[記録と実行環境設定]** を選択します。**[記録と実行環境設定]** ダイアログ・ボックスが開きます。

- b. [Windows Applications] タブで、**「開かれている Windows ベースのアプリケーションすべてでテストを記録して実行する」** が選択されていることを確認します。

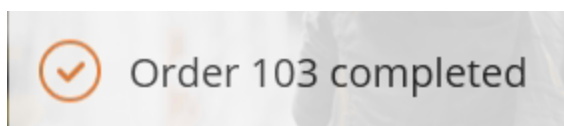


- c. **「OK」** をクリックして、ダイアログ・ボックスを閉じます。

## 5. テキスト・チェックポイントを作成します。

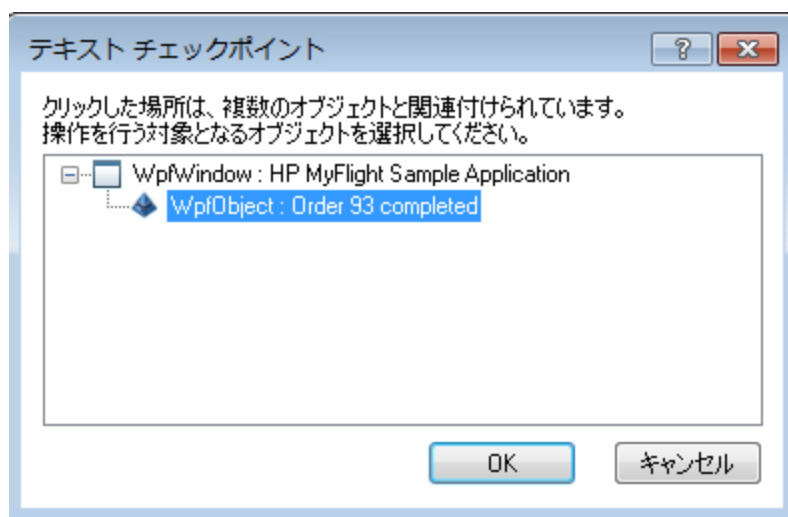
- a. ツールバーで **「記録」** ボタン  をクリックします。UFT ウィンドウが非表示になり、ウィンドウの上部に記録ツールバーが表示されます。
- b. 記録ツールバーで、**「チェックポイントまたは出力値の挿入」** ボタン  をクリックし、**「テキスト チェックポイント」** を選択します。マウス・ポインタが指差し型に変わります。

- c. フライト予約アプリケーションの [Flight Details] ウィンドウで, [Flight Details] ウィンドウの中央にある「Order # Completed」のグラフィックをクリックします。

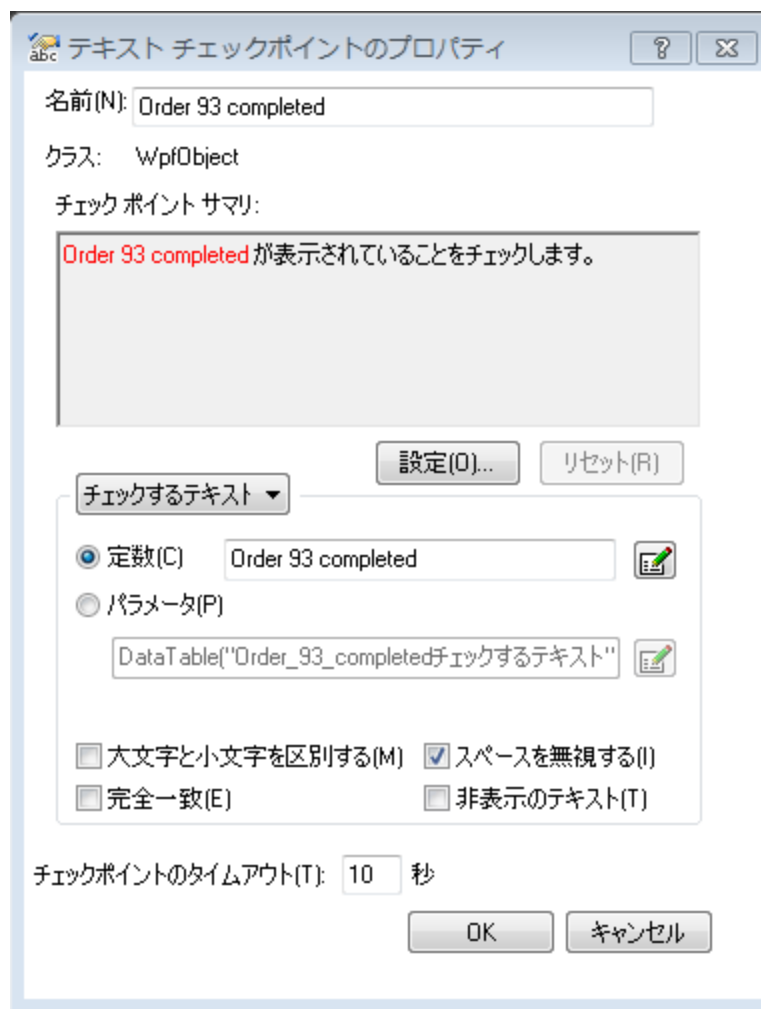


**注:** 注文番号はフライト予約アプリケーションを以前に実行したことがあるかどうかによって変わる場合があります。

[テキスト チェックポイント] オブジェクト選択ダイアログ・ボックスが表示されます。

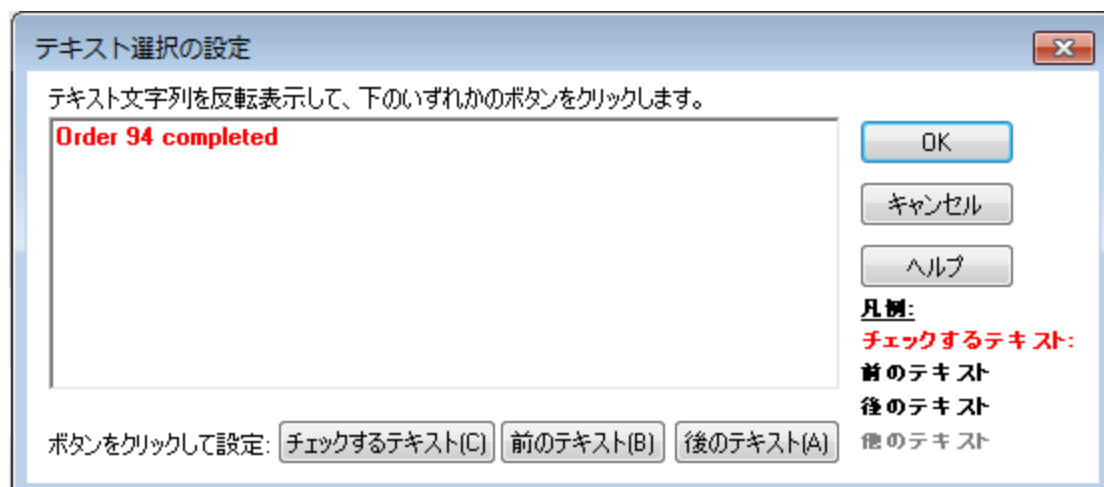


- d. [テキスト チェックポイント] 選択ダイアログ・ボックスで、[WpfObject: Order # Completed] オブジェクトを選択し、[OK] をクリックします。[テキスト チェックポイントのプロパティ] ダイアログ・ボックスが開きます。

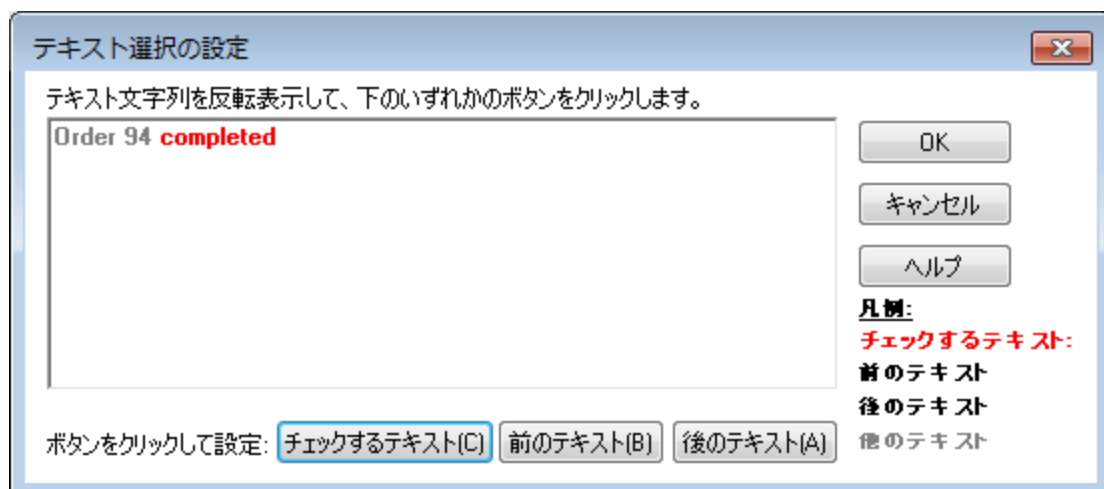


- e. [テキスト チェックポイントのプロパティ] ダイアログ・ボックスの [名前] ボックスに、新規チェックポイント名として「CheckOrderCompletedText」と入力します。

- f. **【チェック ポイント サマリ】** 領域の下にある**【設定】** ボタンをクリックします。**【テキスト選択の設定】** ダイアログ・ボックスが開きます。



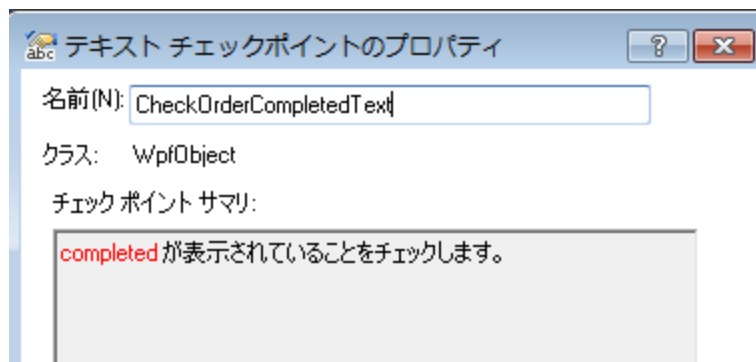
- g. **【テキスト選択の設定】** ダイアログ・ボックスで、文字列 **completed** を強調表示し、**【チェックするテキスト】** をクリックします。**Order #** テキスト文字列が赤からグレーに変わります。



- h. **【OK】** をクリックして、**【テキスト選択の設定】** ダイアログ・ボックスを閉じます。



「テキスト チェックポイントのプロパティ」ダイアログ・ボックスで、「チェックポイントのサマリ」領域が更新され、選択内容が反映されます。






i. その他の設定については標準の設定をそのまま受け入れて [OK] をクリックします。

## 6. 記録セッションを停止します。

記録ツールバーで、[停止]  をクリックして、記録を停止します。


UFT のテストにテキスト・チェックポイントを含むステップが、**progBar** オブジェクトを含むステップの下に追加されます。追加されたチェックポイントは、「**Order # Completed**」オブジェクトを対象としたチェックポイント操作としてキーワード・ビューに表示されます。

 progBar	WaitProperty	"value","100"		次のステップに進む前...
 Order 87 completed	Check	CheckPoint("Order 98 completed")		"Order 87 completed" o...
 NEW SEARCH	Click			"NEW SEARCH" button ...
+ 新規ステップ				

エディタでは、このステートメントは次のようになります。

```
WpfWindow("Book Flights").WpfObject("Order 89 completed").Check CheckPoint
("CheckOrderCompletedText")
```

## 7. テストを保存します。

[保存]  をクリックします。

以上でさまざまなタイプのチェックポイントの追加は完了です。「[練習 6e：オブジェクト・リポジトリ内のチェックポイントの管理](#)」(113 ページ)では、チェックポイントの管理について詳しく学習します。

# 練習 6e：オブジェクト・リポジトリ内のチェックポイントの管理

以前の練習では、アクションにさまざまなタイプのチェックポイントを追加しました。特定のアクションのコンテキストでチェックポイントを使用して表示するほか、オブジェクト・リポジトリ内の

チェックポイントを表示してプロパティを変更することもできます。

オブジェクト・リポジトリ内のチェックポイントを変更することで、同じチェックポイントをテスト内の複数の場所で使用できます。たとえば、組織のロゴがアプリケーションのすべてのページに表示されることを確認するには、チェックポイントを作成し、テスト内のさまざまなアクションや場所に挿入します。

この練習では、チェックポイントの再利用は行いません。

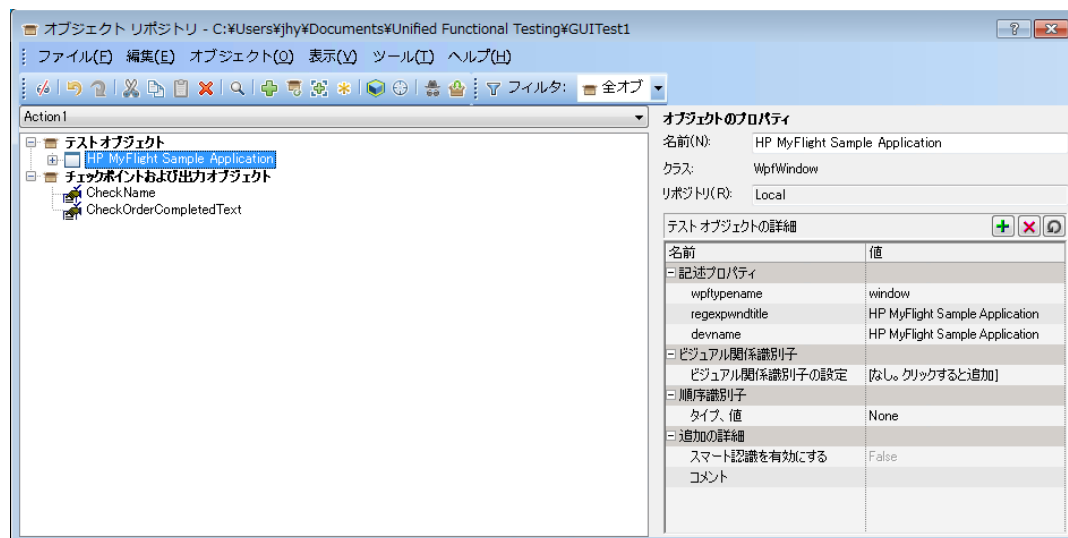
## 1. UFT を起動し、Book Flights Checkpoint テストを開きます。

- a. 必要に応じて、UFT を開きます（「[すべてのテスト用のソリューションの作成](#)」(25ページ)を参照）。WPF Add-in がロードされていることを確認します。
- b. **【ファイル】 > 【開く】 > 【ソリューション】** を選択します。【ソリューションを開く】ダイアログ・ボックスが開きます。
- c. 【ソリューションを開く】ダイアログ・ボックスで、**C:\%HOMEPATH%\My Documents\Unified Functional Testing**にある **Flight Reservation Application** ソリューションに移動し、**【開く】** をクリックします。  
Flight Reservation Application ソリューションが開きます。これには、「[練習 6a：チェックポイント・テストの作成](#)」(97ページ)で作成した Book Flights Parameter テストが含まれます。

## 2. 【オブジェクト リポジトリ】 ウィンドウを開きます。

- a. ソリューション・エクスプローラで、**【Flight Confirmation】** アクションをダブルクリックします。Flight Confirmation アクションがドキュメント表示枠に別のタブとして表示されます。
- b. ツールバーで**【オブジェクト リポジトリ】** ボタン  をクリックします。【オブジェクト リポジトリ】ウィンドウが開き、現在のアクションのすべてのテスト・オブジェクト、およびすべてのチェックポイントと出力オブジェクトのツリーが表示されます。

このツリーには、すべてのローカル・オブジェクトと、アクションに関連付けられている任意の共有オブジェクト・リポジトリ内のすべてのオブジェクトが含まれます。



### 3. チェックポイントを表示するアクションを選択します。

- オブジェクト・ツリーのすぐ上にあるアクション・ドロップダウン・メニューで、テスト・オブジェクト、チェックポイント・オブジェクト、出力値オブジェクトを表示するアクションを選択します。
- 終了したら、[オブジェクトリポジトリ] ウィンドウを閉じます。

**注:** この練習では、オブジェクトまたはチェックポイントのプロパティを変更する必要はありません。

これでチェックポイントを含むテストを実行する準備ができました。続いて「[練習 6f：チェックポイントがあるテストの実行および分析](#)」(115ページ)で、チェックポイントを含むテストの実行と実行結果の表示について学習します。

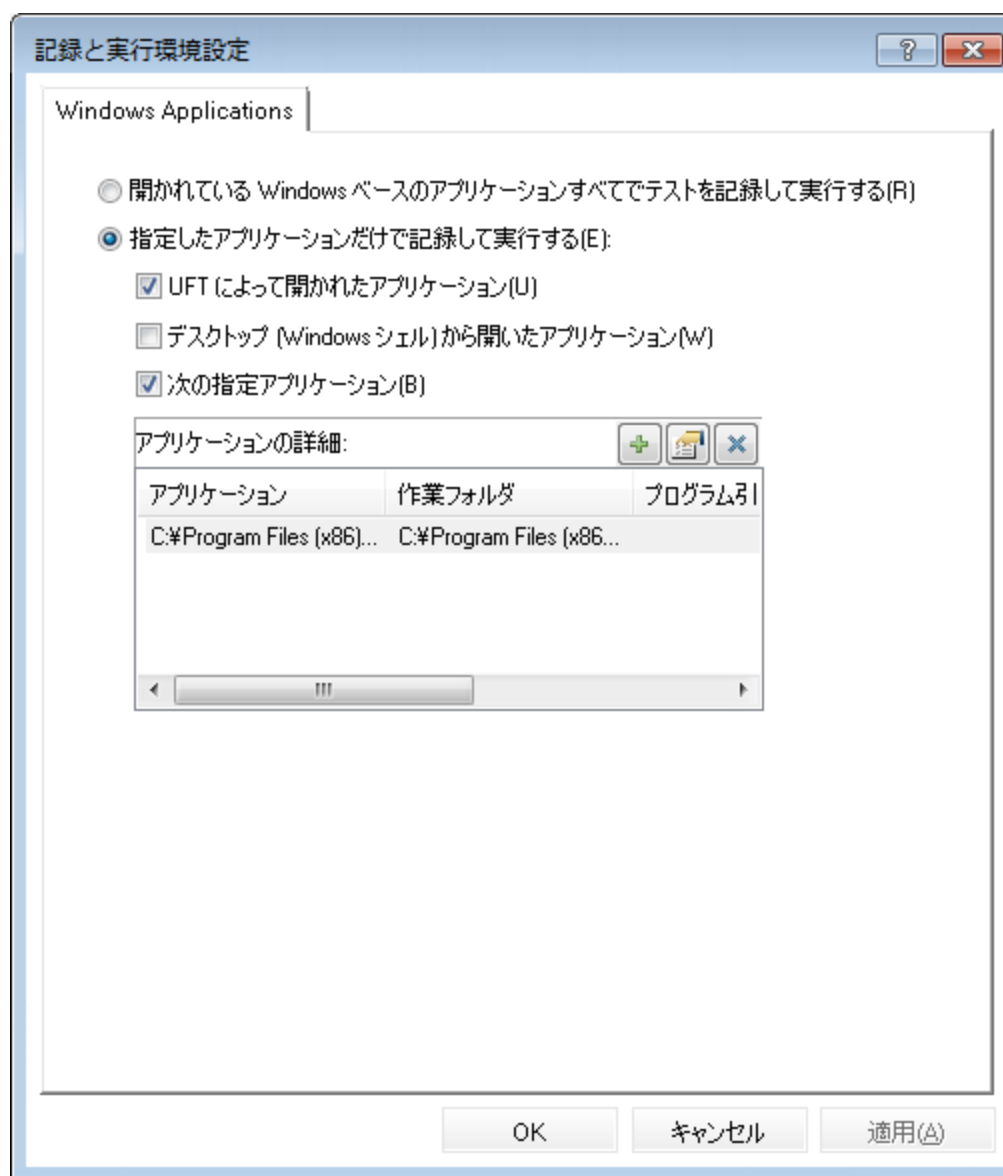
## 練習 6f：チェックポイントがあるテストの実行および分析

チェックポイントを使用するテストの作成が済んだら、テストを実行してチェックポイントがどのように実行されるかを確認する必要があります。このレッスンでは、テストを実行し、チェックポイントの結果を分析します。


### 1. UFT を設定し、フライト予約アプリケーションを開きます。

- UFT で、[記録] > [記録と実行環境設定] を選択します。[記録と実行環境設定] ダイアログ・ボックスが開きます。

- b. [Windows Applications] タブで、**指定したアプリケーションだけで記録して実行する:** オプションを選択します。



**注:** アプリケーションの詳細は、「[レッスン 4 : GUI テストの実行と分析](#)」(73ページ)で設定した状態で保存されています。

- c. **[OK]** をクリックして、ダイアログ・ボックスを閉じます。
2. **テストの実行を開始します。**
- a. ツールバーで**[実行]** ボタン  をクリックします。[実行] ダイアログ・ボックスが開きます。

- b. **【結果保管場所】** タブで, **【新規実行結果フォルダ】** を選択します。結果フォルダ名をそのまま受け入れます。

- c. **【OK】** をクリックします。

UFT でフライト予約アプリケーションが開き, ステップが実行されます。テスト実行が終了すると, 実行結果が開きます。

### 3. 実行結果を表示します。

表示される実行結果が**成功**になっている必要があります。これはすべてのチェックポイントがテストに成功したことを示します。チェックポイントが1つでも失敗した場合, 実行結果は**失敗**と表示されます。

### 4. 標準チェックポイントの結果を表示します。

- a. テスト・フローで, **【Flight Confirmation】** ノードを見つけます。

- b. **【Flight Confirmation】** ノードの下 **【passengerName.Set】** ノードで, **【標準チェックポイント: "CheckName"】** ノードを選択します。

サマりに, 標準チェックポイントの詳細が, 検査されたプロパティおよびその値とともに表示されます。オブジェクトのプロパティの実際の値が期待値と同じなので, このチェックポイントは成功となります。

### 5. テーブル・チェックポイントの結果を表示します。

- a. 結果ツリーで, **【Select Flight】** ノードを展開します。

- b. **【アクション: Select Flight】** ノードで, **【Check Cost】** ノードを見つけます。

- c. **【Check Cost】** ノードを展開し, **【標準チェックポイント: CheckCost】** ノードを選択します。

サマりに, テーブル・チェックポイントの詳細が表示されます。

#### 詳細

データベース チェックポイント

"CheckCost"

説明

Verification type: String Content.

Settings: Exact match - ON; Ignore space - ON; Match case - OFF.

Rows Range : 1 - 7.

Results: Checked 1 cells;

Succeeded: 1;

Failed: 0

実行時間

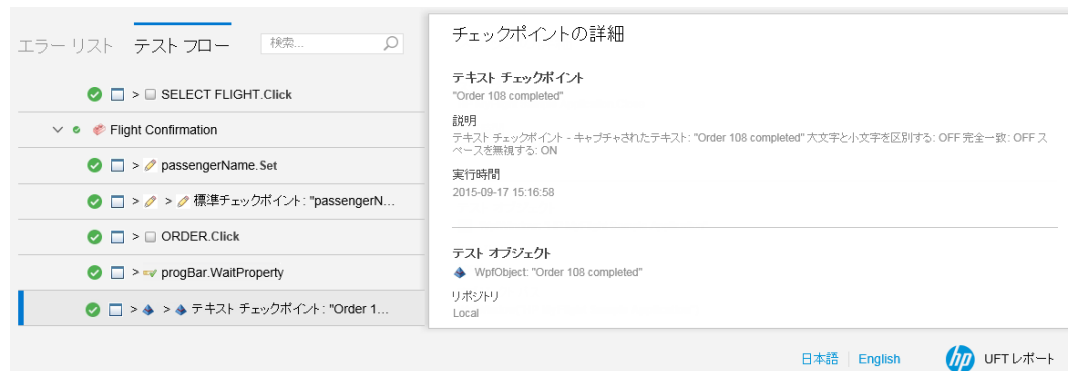
2015-04-02 21:01:30

場合によっては、ステップ・サマリーに、テーブル・オブジェクトからキャプチャしたデータなどのテーブル・チェックポイントに関する詳細も表示されます。

## 6. テキスト・チェックポイントの結果を表示します。

- テスト・フローで、**[Flight Confirmation]** ノードを見つけます。
- [Flight Confirmation]** ノードの下に **[progBar.Wait]** ステップで、**[テキスト チェックポイント: CheckOrderCompleted]** を選択します。

ステップ・サマリーに、チェックポイントの詳細が表示されます。実際のテキストが期待されるテキストと同じなので、このチェックポイントは成功となります。




## 7. 実行結果を閉じます。

ドキュメント表示枠で、実行結果を含むタブを閉じます。

# 練習 6g : 出力値のテストの作成

このレッスンでは、出力値のステップを追加するテストを作成します。このテストは、「[レッスン 1 : GUI テストとアクションの作成](#)」(29ページ)で作成した Book Flights テストに基づいています。

## 1. UFT を起動し、Book Flights テストを開きます。

- 必要に応じて、UFT を開きます（「[すべてのテスト用のソリューションの作成](#)」(25ページ)を参照）。WPF Add-in がロードされていることを確認します。
- 「**開く**」ボタンの下向き矢印  **開く** をクリックし、**[ソリューションを開く]** を選択します。**[ソリューションを開く]** ダイアログ・ボックスが開きます。
- C:\%HOMEPATH%\My Documents\Unified Functional Testing** にある **Flight Reservation Application.ftsln** ファイルに移動し、**[開く]** をクリックします。

Flight Reservation Application ソリューションが開きます。これには、「[レッスン 1 : GUI テストとアクションの作成](#)」(29ページ)で作成した **Book Flights** テストが含まれます。

## 2. テストを Book Flights Output Value として保存します。

- a. ソリューション・エクスプローラで、**[Book Flights]** テスト・ノードを選択し、**[ファイル]** > **[名前を付けて保存]** を選択します。
- b. **[テストに名前を付けて保存]** ダイアログ・ボックスで、**C:\%HOMEPATH%\My Documents\Unified Functional Testing** ディレクトリに移動し、「Book Flights Output Value」という名前でテストを保存します。

ソリューション・エクスプローラで、Book Flights テストが新しい Book Flights Output Value テストに置き換えられます。元の Book Flights テストは、今までと同様にファイル・システム内に保存されています。

### 3. **Book Flights テストをソリューションに再び追加します。**

**Book Flights** テストと **Book Flights Output Value** テストが同じソリューションに含まれる場合、両方のテストを同時に開いておくことができます。こうしておくと、テストの比較や編集を行うときに、交互に切り替えることができます。

**注:** 一度に実行できるテストは 1 つだけです。

- a. **[ファイル]** > **[追加]** > **[既存のテスト]** を選択します。
- b. **[既存のテストの追加]** ダイアログ・ボックスで、**C:\%HOMEPATH%\My Documents\Unified Functional Testing** に保存された Book Flights テストに移動し、**[追加]** をクリックします。

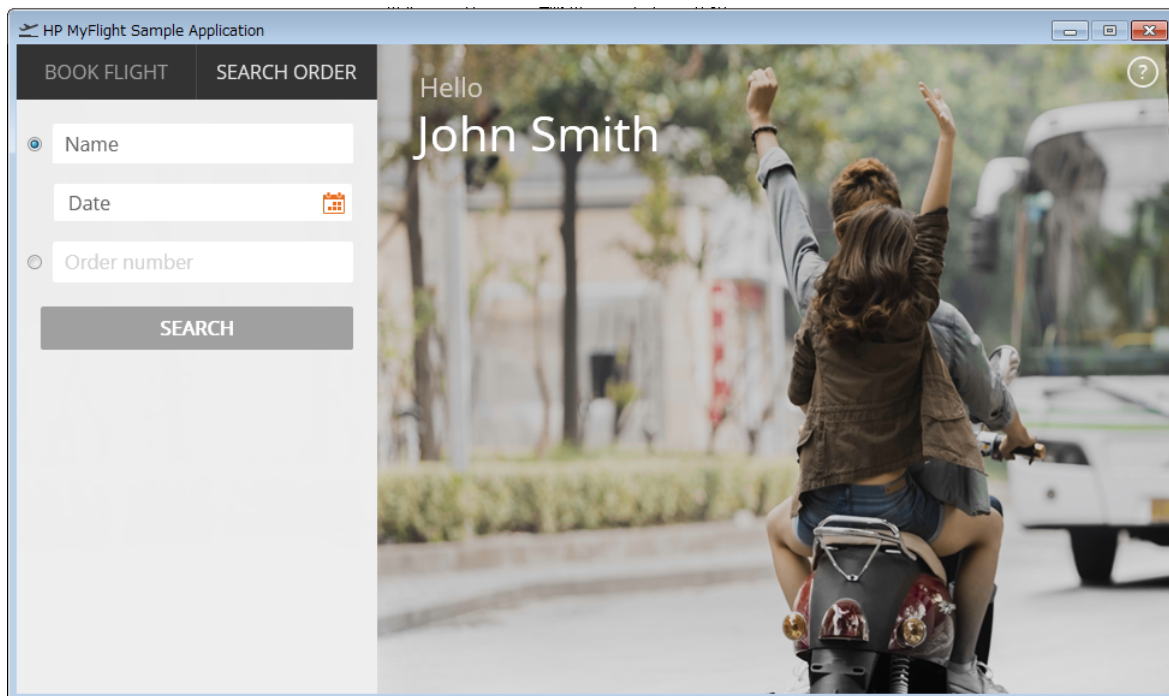
ソリューション・エクスプローラで、Book Flights テストが別のノードとして再度表示されます。

以上で出力値を追加するテストの作成は完了です。続いて「[練習 6h：出力値のステップの追加](#)」(119 ページ)では、出力値のステップをテストに追加します。

## 練習 6h：出力値のステップの追加


最初の Book Flights テストを作成したときには、アプリケーションへのログイン、出発と到着に関する詳細の入力、フライトの選択、および特定の顧客に対するフライト予約を実行するメイン・アプリケーション・フローのアクションを作成しました。

フライト予約アプリケーションには、以前に作成したすべてのフライト申し込みを検索できるアプリケーションの追加領域が存在します。



この練習では、（テストの Flight Confirmation アクションの）[Flight Details] ページでステップの出力を取得し、この出力を検索ページでオブジェクトのパラメータとして使用する出力値ステップを作成します。

### 1. 検索ページにテスト・ステップのアクションを作成します。

- ソリューション・エクスプローラで、[Book Flights] テスト・ノードをクリックします。テスト・フローのキャンバスがドキュメント表示枠に別のタブとして表示されます。
- ツールバーで、[新規アクションの呼び出しの挿入] ボタン  をクリックします。[新規アクションの呼び出しの挿入] ダイアログ・ボックスが開きます。
- [新規アクションの呼び出しの挿入] ダイアログ・ボックスで、新規アクションの名前に「Flight Order Search」と入力します。
- その他の設定は標準設定のままにして、[OK] をクリックします。

**Flight Order Search** という名前の新規アクションのブロックがキャンバスのテスト・フローの末尾に追加され、Flight Order Search アクションがドキュメント表示枠に別のタブとして表示されます。

### 2. 他のアプリケーション・ページ用のオブジェクト・リポジトリを作成します。

Book Flights テストを作成したときには、メイン・アプリケーション・ページ用のオブジェクト・リポジトリのみを作成しました。検索ページ用のテスト・ステップを作成するには、検索



ページ用のオブジェクト・リポジトリを追加で作成する必要があります。

- a. 「[フライト予約アプリケーションの確認](#)」(23ページ)の説明に従って、フライト予約アプリケーションを開きます。
  - b. アプリケーションのログイン資格情報を入力します。
    - **ユーザ名:** john
    - **パスワード:** hp
  - c. **[OK]** をクリックしてログインします。[Flight Finder] ページが開きます。
  - d. [Flight Finder] ページで、ウィンドウの左上にある **[Search Order]** ボタンをクリックします。[Search Details] ページが開きます。
  - e. UFT で、**[リソース] > [オブジェクト リポジトリ マネージャ]** を選択します。[オブジェクト リポジトリ マネージャ] ウィンドウが開きます。
  - f. オブジェクト・リポジトリ・マネージャで、**[ナビゲートして学習]** を使用します (「[練習 2b : \[ナビゲートして学習\] を使用したオブジェクト・リポジトリの作成](#)」(42ページ)を参照)。
  - g. このページ内のすべてのオブジェクトを学習したら、**[ファイル] > [保存]** をクリックします。
  - h. **C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial\_Object Repositories** フォルダに移動し、**[Search]** という名前でオブジェクト・リポジトリを保存します。
  - i. フライト予約アプリケーションの [Search Details] ページで、**[名前]** ボックスに「John Smith」と入力して **[検索]** をクリックします。John Smith で予約されているフライトのリストが別のページに表示されます。
  - j. UFT で **[オブジェクト リポジトリ マネージャ]** ウィンドウを再度開きます。
  - k. [Select Order] ページで **[ナビゲートして学習]** を使用し、このページ内のオブジェクトを学習します。
  - l. [Select Order] ページ内のすべてのオブジェクトを学習したら、**[ファイル] > [保存]** をクリックします。
  - m. **[オブジェクト リポジトリの保存]** ダイアログ・ボックスで、**C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial\_Object Repositories** フォルダに移動し、**[Search\_Results]** という名前でオブジェクト・リポジトリを保存します。
  - n. オブジェクト・リポジトリ・マネージャを閉じます。
3. **新しいオブジェクト・リポジトリを Flight Order Search アクションに関連付けます。**
- a. UFT メイン・ウィンドウのソリューション・エクスプローラで、**[Book Flights Output Value]** テスト・ノードの下に **[C:\%HOMEPATH%\My Documents\Unified Functional Testing]** アクション・ノードを右クリックし、**[リポジトリをアクションに関連付ける]** を選択します。

- b. [共有オブジェクト リポジトリを開く] ウィンドウで、**C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial\_Object Repositories** フォルダに移動し、**Search.tsr** ファイルを選択します。
- c. [開く] をクリックしてオブジェクト・リポジトリを関連付けます。
- d. この手順を繰り返して、**Search Results.tsr** オブジェクト・リポジトリ（同様に **C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial\_Object Repositories** フォルダに保存）を関連付けます。  
**Search.tsr** と **Search Results.tsr** の両方のオブジェクト・リポジトリ・ファイルが、Flight Order Search アクションのサブノードとして表示されます。

#### 4. 検索およびフライト申し込みページを使用するステップを追加します。

- a. ソリューション・エクスプローラで、[Flight Order Search] アクション・ノードをダブルクリックします。Flight Order Search アクションがドキュメント表示枠に別のタブとして表示されます。
- b. キーワード・ビューが表示されている場合は、[表示] > [エディタ] を選択して、エディタを表示します。
- c. エディタで、次の行を貼り付けます。

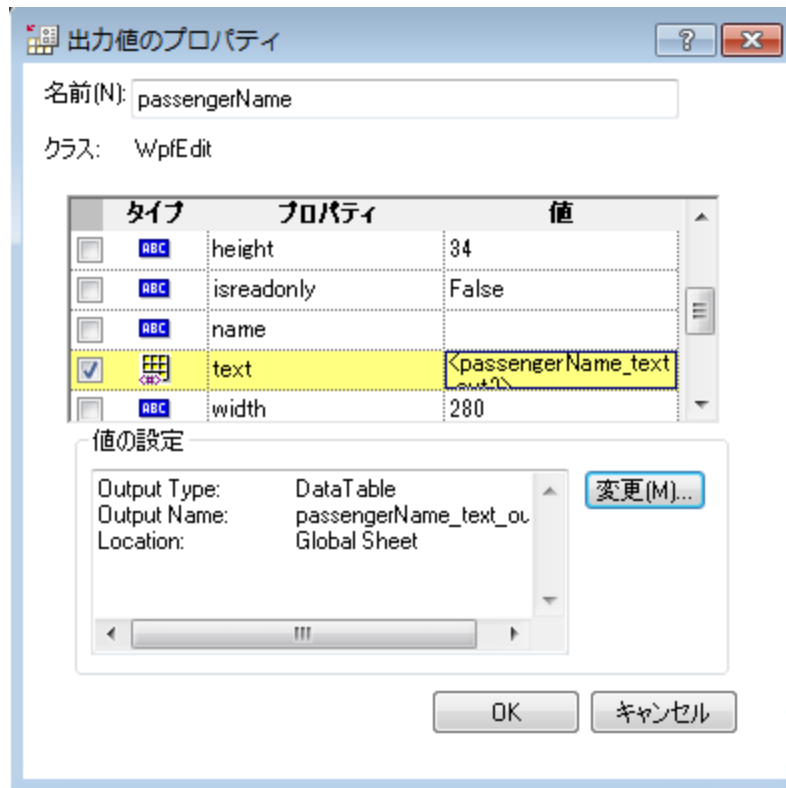
```
WpfWindow("HP MyFlight Sample Application").WpfTabStrip("WpfTabStrip").Select  
"SEARCH ORDER"  
WpfWindow("HP MyFlight Sample Application").WpfEdit("byName").Set "John Smith"  
WpfWindow("HP MyFlight Sample Application").WpfButton("SEARCH").Click  
WpfWindow("HP MyFlight Sample Application").WpfTable("ordersDataGrid").SelectCell 1, 1  
WpfWindow("HP MyFlight Sample Application").WpfButton("SELECT ORDER").Click
```

#### 5. フライト予約アプリケーションの [Flight Details] ページを開きます。

- a. フライト予約アプリケーションの [検索結果] ページで、[戻る] ボタンをクリックします。[Search Details] ページが表示されます。
- b. [Search Details] ページで、左上隅にある [Book Flight] ボタンをクリックします。[Flight Finder] ページが開きます。
- c. [Flight Finder] ページで、フライトの詳細を入力します。
  - **Departure City:** Los Angeles
  - **Arrival City:** Sydney
  - **Date:** 明日の日付
  - **Class:** Business
  - **Tickets:** 2
- d. [Find Flights] ボタンをクリックします。「SELECT FLIGHT」ページが開きます。
- e. [Select Flight] ページで、最初の行を選択して [Select Flight] をクリックします。[Flight Details] ページが開きます。

## 6. Flight Confirmation アクションに出力値のステップを追加します。

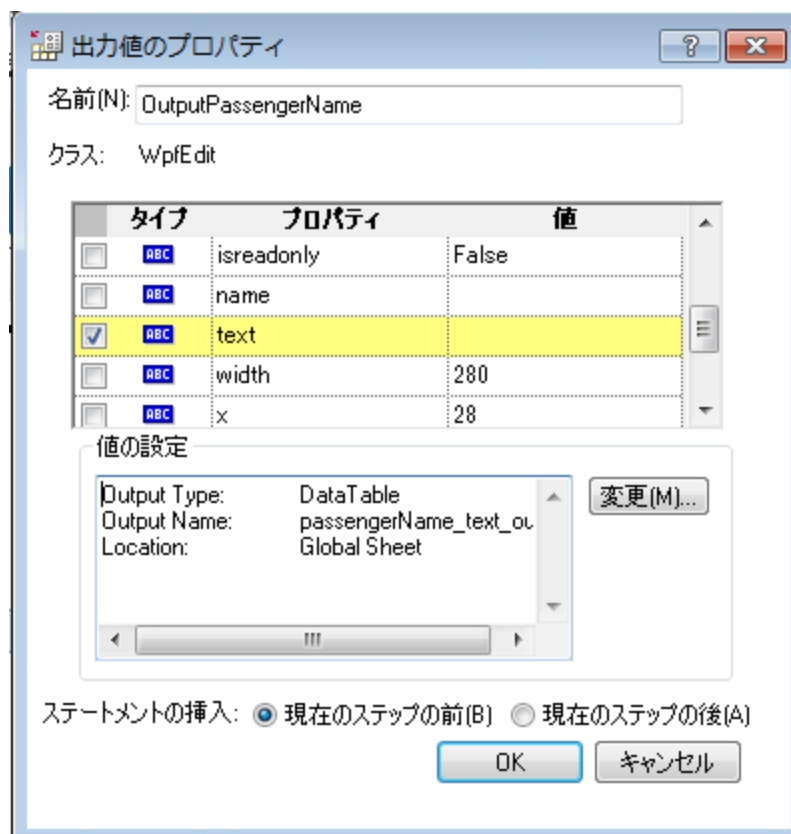
- a. UFT のソリューション・エクスプローラで、[Book Flights] テスト・ノードの下に **[Flight Confirmation]** アクション・ノードをダブルクリックします。Flight Confirmation アクションがドキュメント表示枠に別のタブとして表示されます。
- b. エディタが表示されている場合は、[表示] > [キーワード ビュー] を選択して、キーワード・ビューを表示します。
- c. キーワード・ビューで、byName ステップを右クリックし、[出力値の挿入] を選択します。[出力値のプロパティ] ダイアログ・ボックスが開きます。



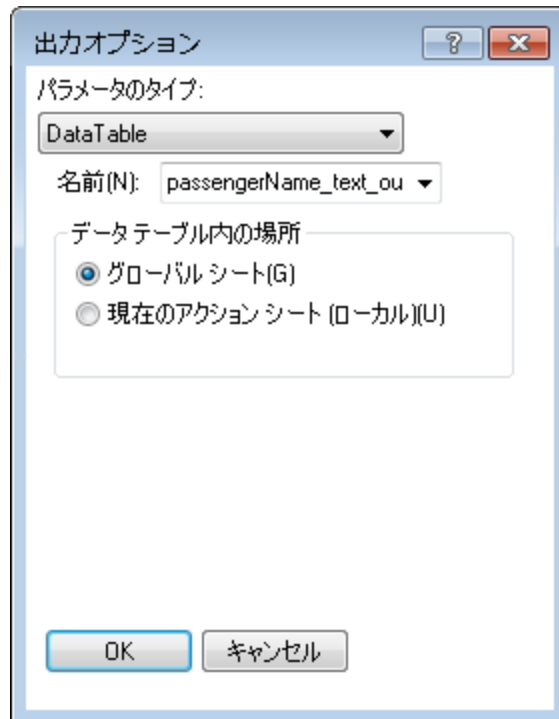
このダイアログ・ボックスには、出力値を挿入するプロパティが表示されます。

- **【名前】** はアプリケーション内で定義されたオブジェクトの名前です。この場合は **byName**。
  - **【クラス】** はオブジェクトの種類です。この場合は **WpfEdit** で、オブジェクトがエディット・ボックスであることを示します。
  - **【タイプ】** カラムの **【ABC】** アイコンは、プロパティの値が定数であることを示します。
  - グリッドには出力用に選択できるオブジェクト・プロパティがリスト表示されます。これには、出力として送信可能なオブジェクトの **【プロパティ】** 名と **【値】** が含まれます。
- d. **【名前】** ボックスに、新しい出力値の名前として「OutputPassengerName」と入力します。

- e. オブジェクトのプロパティのグリッドで、プロパティをスクロールし、**[text]** プロパティ行を選択します。このプロパティには指定された値が存在しないことに注意してください。



- f. プロパティ・グリッドの下に **【変更】** ボタンをクリックします。**【出力オプション】** ダイアログ・ボックスが開きます。（このダイアログでは、このテスト・ステップの出力を保存する場所を特定できます。）



出力値を作成するには、出力値を保存する場所を選択できます。

- DataTable パラメータ
- テスト/アクション・パラメータ（作成済みの場合）
- 環境変数
- コンポーネント・パラメータ（作成済みの場合）

このチュートリアルでは、データテーブルに出力値を保存します。

- g. **【パラメータのタイプ】** ドロップダウン・メニューで、**【DataTable】** を選択します。ダイアログ・ボックスのフィールドが更新され、パラメータの標準設定の名前が提示されます。
- h. **【名前】** フィールドに「passengerName\_text」と入力します。
- i. **【データ テーブル内の場所】** 領域で、**【グローバル シート】** を選択します。これにより、テスト内の他のアクションからアクセスが可能なグローバル・データ・シートに出力を保存できます。

- j. **【OK】** をクリックして、ダイアログ・ボックスを閉じます。[出力値のプロパティ] ダイアログ・ボックスの値が更新され、DataTable パラメータの保存場所オプションが反映されます。

タイプ	プロパティ	値
	isreadonly	False
	name	
	text	<passengerName_text
	width	280
	x	28

- k. **【ステートメントの挿入】** 領域で、**【現在のステップの後】** オプションを選択し、**【OK】** をクリックします。

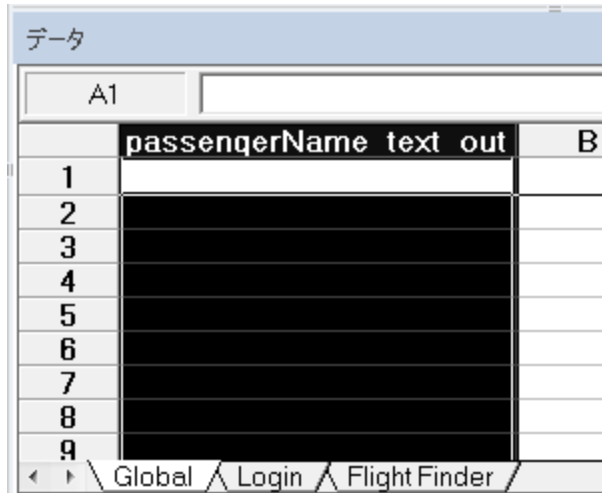
**passengerName.Set** ステップの直後に **Output** ステップが挿入されます。

項目	操作	値	割り当て	コメント	注釈
HP MyFlight Sample Application					
passengerName	Set	"jhy"			"passengerName" edit box に "jhy" を入力する。
passengerName	Output	Checkpoint("passengerName")			"passengerName" edit box の選択されたプロパティ...
ORDER	Click				"ORDER" button をクリックする。
progBar	WaitProperty	"value", "100"			次のステップに進む前に、"progBar" progress bar に...
NEW SEARCH	Click				"NEW SEARCH" button をクリックする。
+ 新規ステップ					


エディタでは、これらのステップは次のようになります。

```
WpfWindow("Book Flights").WpfEdit("passengerName").Set "John Smith"
WpfWindow("Book Flights").WpfEdit("passengerName").Output CheckPoint
("OutputPassengerName")
WpfWindow("Book Flights").WpfButton("ORDER").Click
WpfWindow("Book Flights").WpfProgressBar("progBar").WaitProperty "value", "100"
WpfWindow("Book Flights").WpfButton("NEW SEARCH").Click
```

データ・テーブルのグローバル・シートも変更に応じて更新されます。



## 7. 保存した出力値を使用して検索アクションをパラメータ化します。

- ソリューション・エクスプローラで、[Flight Order Search] アクション・ノードをダブルクリックします。Flight Order Search アクションがドキュメント表示枠に別のタブとして表示されます。
- [Flight Order Search] タブで、[byName] 行を選択します。
- [byName] 行の[値] カラムで、[値の設定] ボタン  をクリックします。[値設定オプション] ダイアログ・ボックスが開きます。
- [値設定オプション] ダイアログ・ボックスで、[パラメータ] ラジオ・ボタンを選択します。
- [パラメータ] ドロップダウン・リストで、[DataTable] を選択します。これに合わせてダイアログ・ボックスの他のフィールドが更新されます。
- [データ テーブル内の場所] 領域で、[グローバル シート] オプションを選択します。
- [名前] ドロップダウン・リストで、[passengerName\_text] パラメータを選択し、[OK] をクリックします。

[byName] 行が更新され、値が DataTable パラメータから提供されるようになります。


項目	操作	値	割り当て	コメント	注釈
HP MyFlight Sample Application					
WpfTabStrip	Select	"SEARCH ORDER"			"WpfTabStrip" tab strip で "SEARCH ORDER" タブを...
byName	Set	DataTable("passengerName_text...")			"byName" edit box に "the 'passengerName_text' Da...
SEARCH	Click				"SEARCH" button をクリックする。
ordersDataGrid	SelectCell	"1", "1"			"ordersDataGrid" table で行 "1"、カラム "1" にある...
SELECT ORDER	Click				"SELECT ORDER" button をクリックする。
+ 新規ステップ					

エディタでは、このステートメントは次のようになります。


```
WpfWindow("HP MyFlight Sample Application").WpfEdit("byName").Set DataTable
```

```
("passengerName_text", dtGlobalSheet)
```

## 8. テストを保存します。

【保存】をクリックします。

## 9. テストを実行して実行結果を表示します。

- a. 【記録】 > 【記録と実行環境設定】を選択します。【記録と実行環境設定】ダイアログ・ボックスが開きます。
- b. 【記録と実行環境設定】ダイアログ・ボックスの【Windows Applications】タブで、【**指定したアプリケーションだけで記録して実行する:**】オプションを選択します。アプリケーションの詳細は、前のテスト実行で保存されたものです。
- c. 【OK】をクリックし、設定を保存してダイアログ・ボックスを閉じます。
- d. ツールバーで【実行】ボタンをクリックします。
- e. 【結果保管場所】タブで、【新規実行結果フォルダ】オプションを選択します。結果フォルダ名をそのまま受け入れます。
- f. 【OK】をクリックします。テストの実行が完了すると、実行結果が開きます。
- g. テスト・フローで、【**Flight Confirmation**】アクション・ノードを見つけます。
- h. 【Flight Confirmation Summary】ノードの下の【標準出力: "**OutputPassengerName**"】ノードを選択します。実行結果に、テスト・ステップのサマリが表示されます。  
サマリには出力値の詳細が表示されます。
- i. テスト・フローで、【**Flight Order Search**】ノードを見つけます。
- j. 【Flight Order Search】ノードで、【**byName.Set**】ノードを選択します。実行結果に、テスト・ステップのサマリが表示されます。  
サマリには、このステップの結果が表示されます。これには **Set** 操作で使用する値も含まれます。この値が出力値になります。

## 10. 実行結果を閉じます。

実行結果の表示が済んだら、実行結果を表示しているタブを閉じます。



## レッスン7：関数と関数ライブラリの作成

UFT には、テストの多くのニーズを満たす組み込み関数と組み込みメソッドが用意されています。ところが、特定のテスト・オブジェクト・クラスに対して標準設定では用意されていないタスクを実行しなければならないこともあります。このような場合、このタスクに対応するユーザ定義関数を作成できます。作成した関数はテストに関連付けられた関数ライブラリ・ファイルに保存し、タスクを実行する必要がある場合に関数の呼び出しをステップとして挿入します。

「[レッスン2：オブジェクト・リポジトリの作成](#)」(33ページ)では、共有オブジェクト・リポジトリを作成し、テスト内のアクションに関連付けました。このレッスンでは、同様のプロセスで関数と関数ライブラリを作成し、関数ライブラリのテストへの関連付けを行います。この関数ライブラリをテストに関連付けることによって、テスト内で関数を任意に呼び出すことができます。

このレッスンには、次の手順が含まれています。

- [関数および関数ライブラリ - 概要](#) ..... 130
- [練習 7a：関数の作成](#) ..... 130
- [練習 7b：関数ライブラリのテストへの関連付け](#) ..... 132
- [練習 7c：関数を使用した検査の実行](#) ..... 133

## 関数および関数ライブラリ - 概要

UFT では、関数を作成して UFT の標準クラスや標準メソッドでサポートされていない特別なタスクを実行できます。関数は、標準設定では適切なメソッドが存在しない特定のタスクを実行するための、コード化されたステップのセットです。このようなタスクをテストに含め、さらにはそのタスクを複数回繰り返すことが必要な場合があります。そのため、関数は容易にアクセスできる必要があります。

作成した関数は関数ライブラリに格納できます。これらの関数ライブラリは、ユーザ定義関数のリポジトリとして機能します。各関数ライブラリは（複数の）テストに割り当てることができます。これにより、テストで関数をテスト・ステップとして呼び出すことが可能になります。

このレッスンでは、フライト予約アプリケーションによって生成されたページの日付の形式を検査する関数を作成し、その関数呼び出しをテストに追加します。

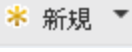
## 練習 7a：関数の作成

この練習では、テストから呼び出す関数を作成します。この関数では、日付が適切な形式で表示されるかどうかを検査します。また、月の値が 12 を超えないことや、日の値が 31 を超えないことなど、日付が有効であることも検査します。

### 1. UFT を起動し、Book Flights テストを開きます。

- UFT が開いていない場合は、「すべてのテスト用のソリューションの作成」(25ページ)の手順に沿って UFT を開きます。WPF Add-in がロードされていることを確認します。
- 【ファイル】 > 【開く】 > 【ソリューション】を選択します。【ソリューションを開く】ダイアログ・ボックスが開きます。
- 【ソリューションを開く】ダイアログ・ボックスで、**C:\%HOMEPATH%\My Documents\Unified Functional Testing**にある**Flight Reservation Application.ftsln**ファイルに移動し、【開く】をクリックします。  
ソリューション・エクスプローラにソリューションが表示されます。これには、Book Flights テストが含まれます。

### 2. 新規関数ライブラリを作成します。

- ツールバーで【新規作成】ドロップダウン矢印  をクリックし、【関数ライブラリ】を選択します。【新規関数ライブラリ】ダイアログ・ボックスが開きます。
- 【新規関数ライブラリ】ダイアログ・ボックスで、次のように関数ライブラリの詳細を入力します。

場所：

- C:\%HOMEPATH%\My Documents\Unified Functional Testing** フォルダを参照します。

	ii. このフォルダ内に <b>Tutorial_Function Libraries</b> という名前の新規フォルダを作成します。 iii. このフォルダを開きます。
ファイル名：	CheckDate Function

- c. **【作成】** をクリックします。空の関数ライブラリがドキュメント表示枠に別のタブとして表示されます。

### 3. 関数を作成します。

CheckDate 関数ライブラリに、次のコードを貼り付けます。

```
'次の関数は、日付文字列（dateStr）が
'DD-<MMM string>-YYYY を表す文字で構成されていることを検査します

Function check_data_validity( dateStr )
    Dim firstDashPos, secondDashPos
    Dim mmPart, ddPart, yyyyPart
    firstDashPos = InStr( dateStr, "-" )
    secondDashPos = InStrRev( dateStr, "-" )
    If ( (firstDashPos <> 2 and firstDashPos <> 3) or (secondDashPos <> 6 and secondDashPos
<> 7)) Then
        reporter.ReportEvent micFail, "Format check", "Date string is"&" missing at least one
dash ( - )."
        check_data_validity = False
        Exit function
    End If

    If firstDashPos = 2 Then
        ddPart = mid( dateStr, 1, 1)
    else
        ddPart = mid( dateStr, 1, 2 )
    End If
    mmPart = mid( dateStr, firstDashPos+1, 3 )
    yyyyPart = mid( dateStr, secondDashPos +1 , 4 )

    If InStr(mmPart, "Jan") and InStr(mmPart, "Feb") and InStr(mmPart, "Mar") and InStr
(mmPart, "Apr") and InStr(mmPart, "May") and InStr(mmPart, "Jun") and InStr(mmPart, "Jul")
and InStr(mmPart, "Aug") and InStr(mmPart, "Sep") and InStr(mmPart, "Oct") and InStr
(mmPart, "Nov") and InStr(mmPart, "Dec") Then
        reporter.ReportEvent micFail, "Format Check", "The month"&" value is invalid.It is not a
valid month string."
        check_data_validity = False
        Exit function
    End If

    If ddPart > 31 Then
```


```
        reporter.ReportEvent micFail, "Format Check", "The date"& " value is invalid.It exceeds
31."
        check_data_validity = False
        Exit function
    End If

    If yyyyPart < 2013 Then
        reporter.ReportEvent micFail, "Format Check", "The year"& " value is invalid.(Prior to
this year.)"
        check_data_validity = False
        Exit function
    End If

    check_data_validity = True

End Function
```

#### 4. 関数ライブラリを保存します。

【保存】  をクリックします。

#### 5. 関数ライブラリを閉じます。

【ファイル】 > 【閉じる】を選択します。関数ライブラリのタブが閉じます。テストは開いた状態のままです。

関数の作成が済んだら、これらの関数をテスト・ステップで使用するため関数をテストに関連付ける必要があります。続いて「[練習 7b：関数ライブラリのテストへの関連付け](#)」(132ページ)で、関数ライブラリとテストを関連付ける方法について学習します。

## 練習 7b：関数ライブラリのテストへの関連付け

「[練習 7a：関数の作成](#)」(130ページ)では、テスト・オブジェクトに対して日付の検査を実行する関数と関数ライブラリを作成しました。ただし、この関数をテストで使用するには、事前に関数ライブラリをテストに関連付ける必要があります。

この練習では、関数ライブラリのテストへの関連付けを行います。


#### 1. UFT を起動し、Book Flights テストを開きます。

- UFT が開いていない場合は、「[すべてのテスト用のソリューションの作成](#)」(25ページ)の手順に沿って UFT を開きます。WPF Add-in がロードされていることを確認します。
- 【ファイル】 > 【開く】 > 【ソリューション】を選択します。【ソリューションを開く】ダイアログ・ボックスが開きます。
- 【ソリューションを開く】ダイアログ・ボックスで、**C:\%HOMEPATH%\My**

**Documents\Unified Functional Testing** にある **Flight Reservation Application.ftsln** ファイルに移動し、**【開く】** をクリックします。

ソリューション・エクスプローラにソリューションが表示されます。これには、Book Flights テストが含まれます。

## 2. CheckDate Function.qfl を Book Flights テストに関連付けます。

- ソリューション・エクスプローラがまだ開いていない場合は、**【ソリューション エクスプローラ】** ボタン  をクリックします。
- ソリューション・エクスプローラで **【Book Flights】** テスト・ノードを右クリックし、**【追加】** > **【関数ライブラリの関連付け】** を選択します。**【関数ライブラリを開く】** ダイアログ・ボックスが開きます。
- 【関数ライブラリを開く】** ダイアログ・ボックスで、**C:\%HOMEPATH%\Unified Functional Testing\Tutorial\_Function Libraries** にある **CheckDate Function.qfl** ファイルに移動し、**【開く】** をクリックします。
- 【自動相対パス変換】** ダイアログ・ボックスで **【はい】** をクリックします。これにより、関数ライブラリのパスが相対パスに変換されます。

関数ライブラリが Book Flights テストに関連付けられ、ソリューション・エクスプローラで Book Flights テストの **【関数ライブラリ】** フォルダに表示されます。

**注:** 相対パスを使用すると、テストやその他のファイルを含むフォルダを、ある場所から別の場所に移動しても、フォルダ階層が同じである限りパスの有効性が保たれます。

## 3. テストを保存します。

ドキュメント表示枠で、キャンパスの **【Book Flights】** タブを選択し、**【保存】**  をクリックします。

関数ライブラリのテストへの関連付けが済んだら、テスト・ステップでその関数を使用できます。続いて「[練習 7c：関数を使用した検査の実行](#)」(133ページ)では、テスト・ステップで関数を使用します。

# 練習 7c：関数を使用した検査の実行

「[レッスン 6：チェックポイントおよび出力値の作成](#)」(94ページ)では、フライト予約アプリケーションのオブジェクトを検査する複数のチェックポイントを作成しました。

この練習では、UFT ユーザ・インタフェースを使用してチェックポイントを作成する代わりに、「[練習 7a：関数の作成](#)」(130ページ)で作成した関数を使用して、Flight Finder アクションのカレンダー・オブジェクトの日付の形式を検査します。

## 1. UFT を起動し、Book Flights テストを開きます。

- UFT が開いていない場合は、「[すべてのテスト用のソリューションの作成](#)」(25ページ)の手順

に沿って UFT を開きます。WPF Add-in がロードされていることを確認します。

- b. **【ファイル】 > 【開く】 > 【ソリューション】** を選択します。【ソリューションを開く】ダイアログ・ボックスが開きます。
  - c. 【ソリューションを開く】ダイアログ・ボックスで、**C:\%HOMEPATH%\My Documents\Unified Functional Testing** にある **Flight Reservation Application.ftsln** ファイルに移動し、**【開く】** をクリックします。
- ソリューション・エクスプローラにソリューションが表示されます。これには、Book Flights テストが含まれます。


## 2. テストを Book Flights Function として保存します。

- a. ソリューション・エクスプローラで、**【Book Flights】** ノードを右クリックし、**【名前を付けて保存】** を選択します。【テストに名前を付けて保存】ダイアログ・ボックスが開きます。
  - b. 【テストに名前を付けて保存】ダイアログ・ボックスで、**C:\%HOMEPATH%\My Documents\Unified Functional Testing** フォルダを参照します。
  - c. テスト名として「**Book Flights Function**」と入力します。
- ソリューション・エクスプローラで、**Book Flights** テストが **Book Flights Function** テストに置き換えられます。Book Flights テストは、今までと同様にファイル・システムに別に保存されています。

## 3. Book Flights テストをソリューションに再び追加します。

Book Flights テストと Book Flights Function テストが同じソリューションに含まれる場合、両方のテストを同時に開いておくことができます。こうしておくと、テストの比較や編集を行うときに、交互に切り替えることができます。

**注:** 一度に実行できるテストは1つだけです。

- a. ツールバーで、**【追加】** ドロップダウン矢印  **追加** をクリックし、**【既存のテストの追加】** を選択します。【テストをソリューションに追加】ダイアログ・ボックスが開きます。
  - b. 【テストをソリューションに追加】ダイアログ・ボックスで、**C:\%HOMEPATH%\My Documents\Unified Functional Testing** フォルダに移動します。
  - c. フォルダで **Book Flights** テストを選択し、**【開く】** をクリックします。
- Book Flights テスト・ノードがソリューション・エクスプローラに再度追加されます。テストはアルファベット順に表示されます。



## 4. フライト予約アプリケーションで【Flight Finder】ページを表示します。

- a. 「**フライト予約アプリケーションの確認**」(23ページ)の説明に従って、フライト予約アプリケーションを開きます。

- b. ログイン画面で、ログイン・ユーザの資格情報を入力します。
  - **ユーザ名**:john
  - **パスワード**:hp
- c. **[OK]** をクリックしてサインインします。[Flight Finder] ページが開きます。

5. **チェックポイント・オブジェクトのオブジェクトのプロパティを調査します。**

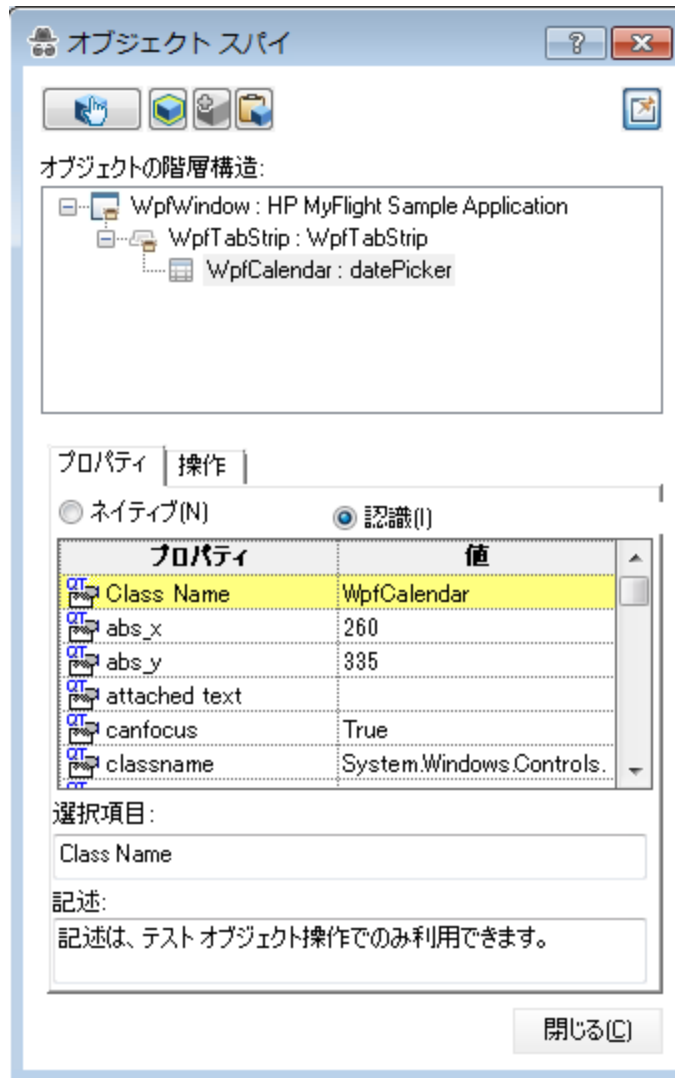
この練習では、カレンダー・オブジェクトの日付の形式を検査する関数を使用します。これを行うには、チェックポイントを設定するオブジェクトのプロパティを学習する必要があります。

- a. UFT ウィンドウのツールバーで、**[オブジェクト スパイ]** ボタン  をクリックします。  
[オブジェクト スパイ] ダイアログ・ボックスが開きます。
- b. [オブジェクト スパイ] ダイアログ・ボックスを開いて表示した状態でアプリケーションを調査できるように、[オブジェクト スパイ] ダイアログ・ボックスをウィンドウの端に移動します。
- c. [オブジェクト スパイ] ダイアログ・ボックスで、指差しボタン  をクリックします。指差しボタンを押すと、UFT が非表示になり、[オブジェクト スパイ] ダイアログ・ボックスがフライト予約アプリケーション上に表示されます。
- d. フライト予約アプリケーションで、日付入力フィールド・オブジェクトをクリックします。

2015/09/15



「オブジェクト スパイ」ダイアログに、オブジェクトのプロパティが表示されます。



- e. プロパティのグリッド（ダイアログ・ボックスの下部）で、日付の値（DD-MMM-YYYY の形式）を持つプロパティが見つかるまでプロパティをスクロールします。このプロパティの名前をメモします。これは、後でテストに関数呼び出しステップを挿入する際に必要になります。
  - f. 「閉じる」をクリックしてオブジェクト・スパイを閉じ、テストに戻ります。
6. **関数呼び出しのアクションを開きます。**
- ソリューション・エクスプローラで、**「Book Flights Function」** テスト・ノードを展開し、**「Flight Finder」** アクションをダブルクリックします。
- Flight Finder アクションがドキュメント表示枠に別のタブとして表示されます。
7. **エディタで関数の呼び出しを追加します。**



このステップでは、「[練習 7a : 関数の作成](#)」(130ページ)で関数ライブラリに追加した `check_data_validity` 関数を追加します。

- キーワード・ビューが表示されている場合は、**[表示]** > **[エディタ]** を選択して、エディタを表示します。
- エディタでカーソルを `WpfCalendar.SetDate` ステップの後に置き、**ENTER** キーを押します。
- エディタにステップを追加し、`datePicker` オブジェクトの日付プロパティを取得します。

```
departureDate = WpfWindow("HP MyFlight Sample"& "Application").WpfCalendar
("datePicker").GetROProperty("date")
```

このステップでは、UFT で該当するプロパティを検査する際にオブジェクトに対してチェックポイント関数を実行するために、日付プロパティの値を取得します。

- エディタに別のステップを追加し、チェックポイント関数を呼び出します。

```
If check_data_validity(departureDate) Then
    reporter.ReportEvent micPass, "Date is valid" , departureDate
End If
```

これらのステップの追加が完了すると、アクションは次のようになります（各自が使用する日付は異なります）。

```
WpfWindow("HP MyFlight Sample Application").WpfComboBox("fromCity").Select "Los
Angeles"
WpfWindow("HP MyFlight Sample Application").WpfComboBox("toCity").Select Sydney
WpfWindow("HP MyFlight Sample Application").WpfCalendar("datePicker").SetDate "17-
Jul-2014"
departureDate = WpfWindow("HP MyFlight Sample Application").WpfCalendar
("datePicker").GetROProperty("date")
If check_data_validity(departureDate) Then
    reporter.ReportEvent micPass, "Date is valid" , departureDate
End If
WpfWindow("HP MyFlight Sample Application").WpfComboBox("Class").Select "Business"
WpfWindow("HP MyFlight Sample Application").WpfComboBox("numOfTickets").Select "2"
WpfWindow("HP MyFlight Sample Application").WpfButton("FIND FLIGHTS").Click
```

- これらのステップを貼り付けた後で、後続のステップ - `WpfWindow("HP MyFlight Sample Application").WpfComboBox("Class").Select "Business"` - が貼り付けたステップの次の行から始まることを確認します。そうならない場合は、カーソルをこのステップの前に置いて、**ENTER** キーを押します。

この関数は、アプリケーションで次のフィールドが編集される前に実行する必要があります。

- それぞれのアクションで、コードの `departureDate` 行内の改行を確実に取り除き、この行が `departureDate` から始まり、`("date")` で終わるようにします。

## 8. キーワード・ビューでこれらのステップを確認します。

【表示】 > 【キーワード ビュー】を選択して、キーワード・ビューに切り替えます。


**GetROProperty** ステップの下に関数呼び出しステップが別のステップとして存在することに注意してください。

項目	操作	値	割り当て	コメント	注釈
HP MyFlight Sample Application					
fromCity	Select	"Los Angeles"			"fromCity" list から項目 "Los Angeles" ...
toCity	Select	Sydney			"toCity" list から項目 Sydney を選択...
datePicker	SetDate	"17-Jul-2014"			"datePicker" calendar で日付または...
datePicker	GetROProperty	"date"	保管先: dep...		"datePicker" calendar で "date" プロパ...
関数の呼び出し	check_data_...	departureDate			
reporter		reporter.ReportEvent mic...			
Class	Select	"Business"			"Class" list から項目 "Business" を選...
numOfTickets	Select	"2"			"numOfTickets" list から項目 "2" を選...
FIND FLIGHTS	Click				"FIND FLIGHTS" button をクリックする。
+ 新規ステップ					

## 9. テストを保存します。

【ファイル】 > 【保存】を選択します。

## 10. テストを実行します。

- a. ツールバーで【実行】ボタン  をクリックします。【実行】ダイアログ・ボックスが開きます。

**注:** テストを実行する前に、必ずフライト予約アプリケーションを閉じておきます。

- b. 【実行】ダイアログ・ボックスの【結果保管場所】タブで、【新規実行結果フォルダ】オプションが選択されていることを確認します。結果フォルダ名をそのまま受け入れます。
- c. 【実行】をクリックします。UFT でステップが順番に実行されます。  
テストの実行が完了すると、【実行結果】が開きます。

## 11. 実行結果を分析します。

- a. 実行結果のテスト・フローで、【Flight Finder】ノードの下の【Date is valid】ノードを選択します。

ステップ名の横に緑のチェックマークが表示されていることに注意してください。これは、追加した関数ごとにチェックポイントが成功したことを示します。

## レッスン 8 : テストでの Insight の使用

テストを行うためにテストやテスト・オブジェクトを作成していると、通常のオブジェクトの認識がアプリケーション内のオブジェクトの識別に役立たないことがあります。つまり、標準のオブジェクトの認識では、アプリケーションまたはブラウザ・ウィンドウ内の位置やウィンドウ内での表示の時間などのオブジェクトのプロパティを使用しますが、これがテストでアプリケーション内のオブジェクトを識別するのに役立たない場合があります。

通常のオブジェクトの認識が適切に機能しない場合やニーズに合わない場合に備えて、UFT には **Insight** という名前の画像に基づくオブジェクト認識メカニズムが用意されています。このメカニズムでは、テスト実行中に画像のスナップショットをキャプチャしてその画像をオブジェクトとして使用することで、オブジェクトを識別できます。

このレッスンでは、Insight を使用してオブジェクトを認識し、これらのオブジェクトをテスト内で使用する方法について学習します。

このレッスンには、次の手順が含まれています。

- [Insight オブジェクトの識別 - 概要](#) .....140
- [練習 8a : Insight オブジェクト用のテストの作成](#) .....140
- [練習 8b : オブジェクト・リポジトリへの Insight オブジェクトの追加](#) .....141
- [練習 8c : テストでの Insight オブジェクトの使用](#) ..... 144

## Insight オブジェクトの識別 - 概要

UFT では、画像ベースの認識機能である **Insight** を使用して、アプリケーション内のオブジェクトを認識できます。Insight では、アプリケーション/オブジェクト設計の一部であるオブジェクトのプロパティを使用する代わりに、オブジェクトがどう見えるかに基づいてオブジェクトを認識します。Insight オブジェクトの認識は、アプリケーションが UFT でサポートされていないテクノロジーを使用して設計されている場合や、アプリケーションがリモート・コンピュータで実行されている場合に特に便利です。


Insight オブジェクトの認識を使用する場合、UFT はオブジェクトの画像を Insight テスト・オブジェクトの一部として格納します。その後、テストの実行時に、UFT は画像をメイン・オブジェクト・プロパティとして使用し、実行環境でアプリケーション内のオブジェクトを認識します。

Insight オブジェクトは、オブジェクト・リポジトリ内または記録時に作成できます。このレッスンでは、オブジェクト・リポジトリへの Insight オブジェクトの追加のみを行います。

## 練習 8a : Insight オブジェクト用のテストの作成

この練習では、テストを作成して Insight オブジェクトを使用するテスト・ステップを追加します。このテストは、**Book Flights** テストを含むソリューションとは別に保存されます。

### 1. UFT を起動します。

- UFT が開いていない場合は、「[すべてのテスト用のソリューションの作成](#)」(25ページ)の手順に沿って開きます。WPF Add-in が開いていることを確認します。
- ツールバーで **[新規作成]** ボタン  **新規** をクリックします。[新規テスト] ダイアログ・ボックスが開きます。
- [新規テスト] ダイアログ・ボックスで、**[GUI テスト]** を選択します。
- テストに「Insight」という名前を付けます。標準設定の場所をそのまま使用します。

**注:** このテストではソリューション名を入力しないでください。このテストは、このチュートリアルの他のレッスンで使用するソリューションとは別に保存されます

- [作成]** をクリックしてテストを作成します。  
作成したテストがドキュメント表示枠に別のタブとして表示されます。


### 2. オブジェクト・リポジトリとアクションを関連付けます。

このテストでは、すべてのテスト・ステップを1つのアクション内に保持します。ただし、適切なテスト・オブジェクトを用意するためには、複数のオブジェクト・リポジトリが必要です。

このテストでは、フライト予約アプリケーションにログインした後に、プロモーション画像のリンクをクリックしてフライトを申し込みます。そのため、アプリケーションの [Login] ページと [Flight Finder] ページに対するオブジェクト・リポジトリを含める必要があります。

- a. ソリューション・エクスプローラで **Action1** ノードを右クリックし、**「リポジトリをアクションに関連付ける」**を選択します。[共有オブジェクト リポジトリを開く] ダイアログ・ボックスが開きます。
- b. [共有オブジェクト リポジトリを開く] ダイアログ・ボックスで、**C:\%HOMEPATH%\Unified Functional Testing\Tutorial\_Object Repositories** にある、**「レッスン 2 : オブジェクト・リポジトリの作成」(33ページ)** で作成した **Login.tsr** オブジェクト・リポジトリ・ファイルに移動し、**「開く」** をクリックします。  
**Login.tsr** オブジェクト・リポジトリが **Action1** ノードのサブノードとして表示されます。
- c. 上記の手順を繰り返し、**Flight Finder.tsr** オブジェクト・リポジトリ・ファイルを Action1 に追加します。

### 3. テストを保存します。


[保存]  をクリックします。

以上で Insight オブジェクトを使用するテストの作成は完了です。続いて **「練習 8b : オブジェクト・リポジトリへの Insight オブジェクトの追加」(141ページ)** に進み、Insight オブジェクトをオブジェクト・リポジトリに追加して Insight オブジェクトをテスト・ステップで使用できるようにします。

## 練習 8b : オブジェクト・リポジトリへの Insight オブジェクトの追加

**「練習 8a : Insight オブジェクト用のテストの作成」(140ページ)** では、Insight オブジェクトで使用するテストの構造を作成しました。ただし、これらのオブジェクトをテストに追加するには、事前にオブジェクト・リポジトリにオブジェクトを追加する必要があります。このレッスンでは、オブジェクト・リポジトリ機能を使用して、Insight オブジェクトをテストに追加する方法について学習します。

### 1. UFT を起動します。

- a. UFT が開いていない場合は、**「すべてのテスト用のソリューションの作成」(25ページ)** の手順に沿って開きます。WPF Add-in が開いていることを確認します。
- b. ツールバーで、**「開く」** ドロップダウン矢印  **開く** を選択し、**「テストを開く」** を選択します。[テストを開く] ダイアログ・ボックスが開きます。
- c. [新規テスト] ダイアログ・ボックスで、**C:\%HOMEPATH%\Unified Functional Testing** に保存した **Insight** テストに移動し、**「開く」** をクリックします。  
作成したテストがドキュメント表示枠に別のタブとして表示されます。

## 2. 新しいオブジェクト・リポジトリを作成します。


- a. [リソース] > [オブジェクト リポジトリ マネージャ] を選択します。[オブジェクト リポジトリ マネージャ] ダイアログ・ボックスが開きます。
- b. [オブジェクト リポジトリ マネージャ] ウィンドウで、[ファイル] > [保存] を選択します。[共有オブジェクト リポジトリの保存] ダイアログ・ボックスが開きます。
- c. [共有オブジェクト リポジトリの保存] ダイアログ・ボックスで、  
**C:%HOMEPATH%\Unified Functional Testing** にある **Tutorial\_Object Repositories** フォルダに移動します。
- d. オブジェクト・リポジトリに「**Insight**」という名前を付けて、[保存] をクリックします。

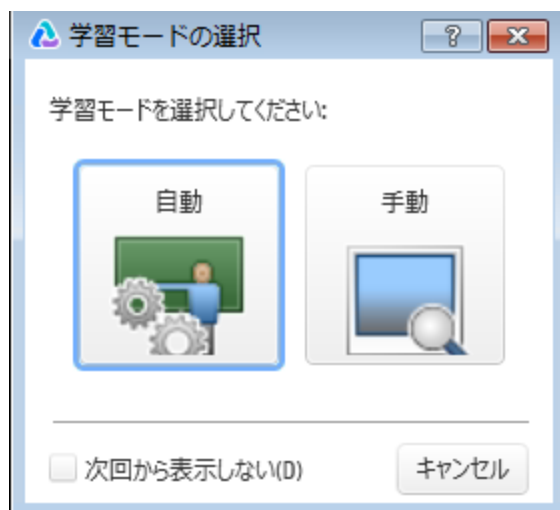
## 3. フライト予約アプリケーションの [Flight Finder] ページを開きます。

フライト予約アプリケーションの [Flight Finder] ページには、次のような特別な航空券の広告を表示する変化するオブジェクトが存在します。



この画像はアプリケーション内で周期的に変化するため、Insight オブジェクト認識を使用してテストする必要があるオブジェクトのタイプです。従来のオブジェクト認識では、画面上での他のオブジェクトに対する相対位置や順序識別子などを使用します。さらに、画像の中に配置された [Order] ボタンは、個別のオブジェクトとして識別できません。たとえば、オブジェクト・スパイを使用して [Order] ボタンを識別する場合、UFT はこのボタンを識別できません。

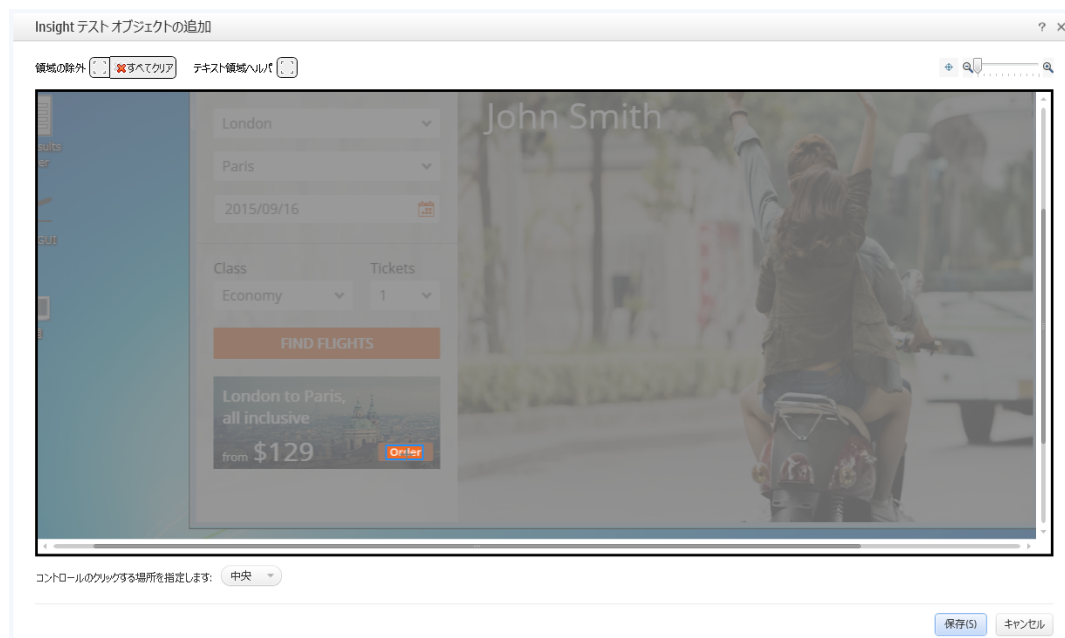
- a. 「[フライト予約アプリケーションの確認](#)」(23ページ)の説明に従って、フライト予約アプリケーションを開きます。
  - b. ログイン・ウィンドウで、次のユーザ資格情報を入力します。
    - ユーザ名: john
    - パスワード: hp
  - c. [OK] をクリックしてアプリケーションにログインします。[Flight Finder] ページが開きます。
- ## 4. Insight オブジェクトをオブジェクト・リポジトリに追加します。
- a. オブジェクト・リポジトリ・マネージャのツールバーで、[Insight オブジェクトを追加] ボタン  をクリックします。[学習モードの選択] ダイアログ・ボックスが開きます。



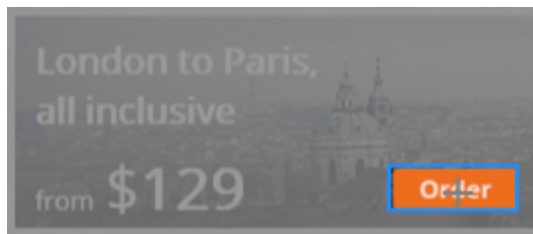
- b. 「学習モードの選択」ダイアログ・ボックスで、「自動」ボタンをクリックします。UFT が非表示になり、アプリケーションが表示されます。

**注:** 「自動」モードを選択すると、UFT は領域/オブジェクトを Insight オブジェクトとして自動的に選択できます。「手動」を選択した場合は、ユーザが Insight オブジェクトとして使用する領域/オブジェクトを指定できます。

- c. フライト予約アプリケーションで、航空券のプロモーション画像内の「Order」ボタンをクリックします。「Insight テストオブジェクトの追加」ダイアログ・ボックスが開きます。

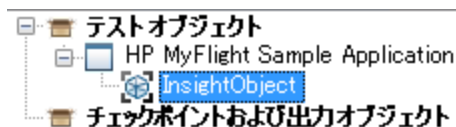



- d. 画像エディタ（ダイアログ・ボックスの中央）で、[Order] ボタン全体が含まれるようにオブジェクトの選択ボックスをドラッグします。



- e. [コントロールのクリックする場所を指定します] オプションで、[中央] を選択します。  
f. [保存] をクリックしてオブジェクトをオブジェクト・リポジトリに追加します。

Insight オブジェクトがオブジェクト・リポジトリの最上位オブジェクトとして追加されます。




- g. オブジェクト・リポジトリ・マネージャの[オブジェクトのプロパティ] 表示枠（右側）で、オブジェクトの名前を **InsightObject** から **Promotion Order** に変更します。  
h. ツールバーの[保存]  をクリックして、オブジェクト名を変更してオブジェクト・リポジトリを保存します。

Insight の画像を含むオブジェクト・リポジトリの作成が済んだら、テストでこれらのオブジェクトを使用できます。続いて「[練習 8c : テストでの Insight オブジェクトの使用](#)」(144ページ)では、テスト内で Insight オブジェクトを使用してテストを実行します。

## 練習 8c : テストでの Insight オブジェクトの使用

「[練習 8b : オブジェクト・リポジトリへの Insight オブジェクトの追加](#)」(141ページ)では、アプリケーション内のオブジェクト（画像）に対して Insight オブジェクトを追加しました。この練習では、テストでこれらのオブジェクトを使用して、Insight オブジェクトの認識を使用したときにテストがどのように実行されるかを確認します。


### 1. UFT を起動します。

- a. UFT が開いていない場合は、「[すべてのテスト用のソリューションの作成](#)」(25ページ)の手順に沿って開きます。WPF Add-in が開いていることを確認します。  
b. ツールバーで、[開く] ドロップダウン矢印  を選択し、[テストを開く] を選択します。[テストを開く] ダイアログ・ボックスが開きます。



- c. [新規テスト] ダイアログ・ボックスで、**C:%HOMEPATH%\Unified Functional Testing** に保存した **Insight** テストに移動し、[開く] をクリックします。  
作成したテストがドキュメント表示枠に別のタブとして表示されます。

## 2. Insight オブジェクト・リポジトリをテストに関連付けます。

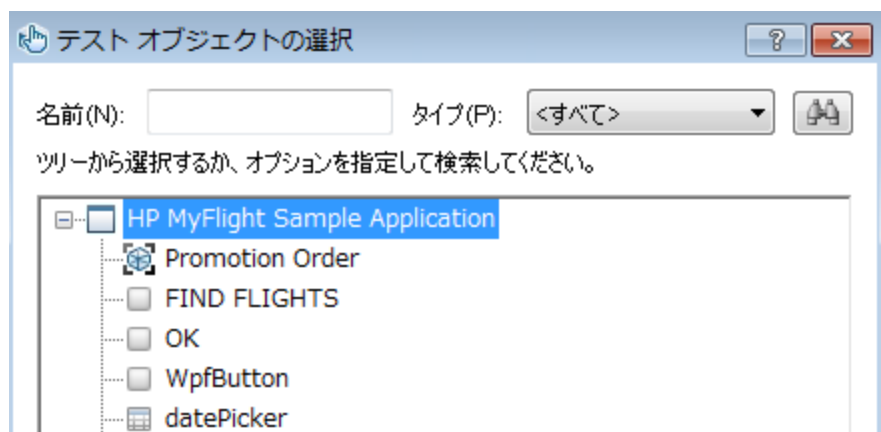
- a. ソリューション・エクスプローラで **Action1** ノードを右クリックし、[リポジトリをアクションに関連付ける] を選択します。[共有オブジェクトリポジトリを開く] ダイアログ・ボックスが開きます。
- b. [共有オブジェクトリポジトリを開く] ダイアログ・ボックスで、**C:%HOMEPATH%\Unified Functional Testing\Tutorial\_Object Repositories** に保存された **Insight.tsr** オブジェクト・リポジトリ・ファイルに移動し、[開く] をクリックします。  
ソリューション・エクスプローラに **Insight.tsr** ファイルが **Action1** ノードのサブノードとして表示され、関連するオブジェクトがテストで使用できるようになります。
- c. [保存]  をクリックして変更を保存します。

## 3. テストにログイン・ステップを追加します。

フライト予約アプリケーションにログインするステップをテストに追加します (「練習 3a : キーワード・ビューでの Login アクションへのステップの追加」(47ページ)を参照)。

## 4. Insight オブジェクトをテストに追加します。

- a. キーワード・ビューで、**OK** ボタン・ステップの下の [項目] カラム内をクリックし、ドロップダウン・リストから [リポジトリからのオブジェクト] を選択します。[テストオブジェクトの選択] ダイアログ・ボックスが開きます。
- b. [テストオブジェクトの選択] ダイアログ・ボックスのテスト・オブジェクト・ツリーで、[**Promotion Order**] ノードを選択します。




- c. [**OK**] をクリックしてステップをテストに追加します。  
Insight オブジェクトの新規ステップがテストに追加されます。

## 5. テストを実行して実行結果を表示します。

- a. **【記録】** > **【記録と実行環境設定】** を選択します。**【記録と実行環境設定】** ダイアログ・ボックスが開きます。
- b. **【Windows アプリケーション】** タブで, **【指定したアプリケーションだけで記録して実行する:】** オプションを選択します。
- c. **【指定したアプリケーションだけで記録して実行する】** オプションの下にある **【次の指定アプリケーション】** オプションを選択します。アプリケーションの詳細は, 前の実行セッションで保存されたものです。

**注:** アプリケーションの詳細を再入力する必要がある場合は, [「練習 4a : テストの実行」\(74ページ\)](#)で必要な詳細を確認してください。

- d. **【OK】** をクリックし, 変更を保存して設定を適用します。
- e. ツールバーで **【実行】** ボタン  をクリックします。**【実行】** ダイアログ・ボックスが開きます。
- f. **【実行】** ダイアログ・ボックスの **【結果保管場所】** タブで, **【新規実行結果フォルダ】** を選択します。標準設定フォルダ名をそのまま使用します。
- g. **【実行】** をクリックしてテスト実行を開始します。

**重要 :** テストを実行する前にフライト予約アプリケーションが閉じていることを確認してください。

UFT でフライト予約アプリケーションが開かれ, テスト・ステップが順番に実行されます。UFT は Insight オブジェクトを含むステップを実行する際に, Insight オブジェクトの認識メカニズムを使用してオブジェクトを見つけます。

テストの実行が完了すると, 実行結果が開きます。


- h. テスト・フローで, **Promotion Order.Click** ノードを選択します。実行結果に, ステップのサマリが表示されます。
- i. ステップの詳細で, UFT がテスト実行中に Insight オブジェクトの認識を使用して識別したオブジェクトを確認できます。

#### 実行時間

2015-09-15 14:05:16

---

#### テスト オブジェクト

 InsightObject: "Promotion Order"

#### リポジトリ

C:\User\brojerem\Desktop\UFT\_Tutorial\_Test\NET\_Tutorial\Tutorial\_Object  
Repositories\Insight Objects.tsr

#### オブジェクト パス

WpfWindow("HP MyFlight Sample Application").InsightObject("Promotion Order")

---

#### コンテキスト情報

 HP MyFlight Sample Application

WpfWindow

## 第4部: API 自動テストの作成と実行

アプリケーションのユーザ・インタフェースのテストのほかに、非 GUI（サービス）層をテストしてアプリケーションを実行する API が正しく機能することを確認する必要があります。UFT を使用すると、このための API 自動テストを作成できます。

API テストを作成する場合、アプリケーションの API プロセスを個別にテストするテスト・ステップで構成されたテスト・フロー全体を作成します。これらのテスト・ステップの入力プロパティとチェックポイント・プロパティを指定し、UFT でアプリケーションでテストを実行します。テスト実行が終了すると、結果をチェックしてアプリケーションがどのように機能しているかを確認できます。


このパートでは、さまざまなタイプのアプリケーションとサービスに対して API テストを作成する方法を学習します。

本項には、次の項目が含まれています。

- [レッスン 1：API テストの作成](#) ..... 149
- [レッスン 2：単純な API テスト・ステップの作成](#) .....150
- [レッスン 3：標準アクティビティを使用した API テスト・ステップの作成](#) .....155
- [レッスン 4：API テスト・ステップのパラメータ化](#) .....161
- [レッスン 5：API テストの実行](#) ..... 183
- [レッスン 6：Web サービスの API テストの作成と実行](#) ..... 188
- [レッスン 7：REST サービスの API テストの作成と実行](#) .....202
- [レッスン 8：Web アプリケーション・サービスの API テストの作成と実行（WADL）](#) ..... 217

# レッスン 1：API テストの作成

アプリケーションのサービス層のテストの内容を作成する前に、最初にテストの作成とテスト構造の作成を行う必要があります。

1. UFT のツールバーで、**【新規作成】** ドロップダウン矢印  をクリックし、**【新規テスト】** を選択します。
2. **【新規テスト】** ダイアログ・ボックスで、**【API テスト】** を選択します。
3. テストに関する次の詳細情報を入力します。

**名前：Basic**

**場所：**標準設定では、ドキュメントは **C:\%HOMEPATH%\My Documents\Unified Functional Testing** に保存されます。このレッスンでは、このパスを変更する必要はありません。

4. **【作成】** をクリックします。

空白のテストが (**Basic** という名前の) テスト・フローのタブのキャンバスに表示されます。このテスト・フローのキャンバス内には空のテスト・フローがあります。

このテストは、[ソリューション エクスプローラ] 表示枠に**無題のソリューション・ノード**のサブノードとしても表示されます（これはテストが作成されてソリューション名が指定されていない場合の一般ソリューション名です）。

これで API テストの設計を開始する準備ができました。続いて「[レッスン 2：単純な API テスト・ステップの作成](#)」(150 ページ)に進み、API テスト・ステップの作成方法について学習します。

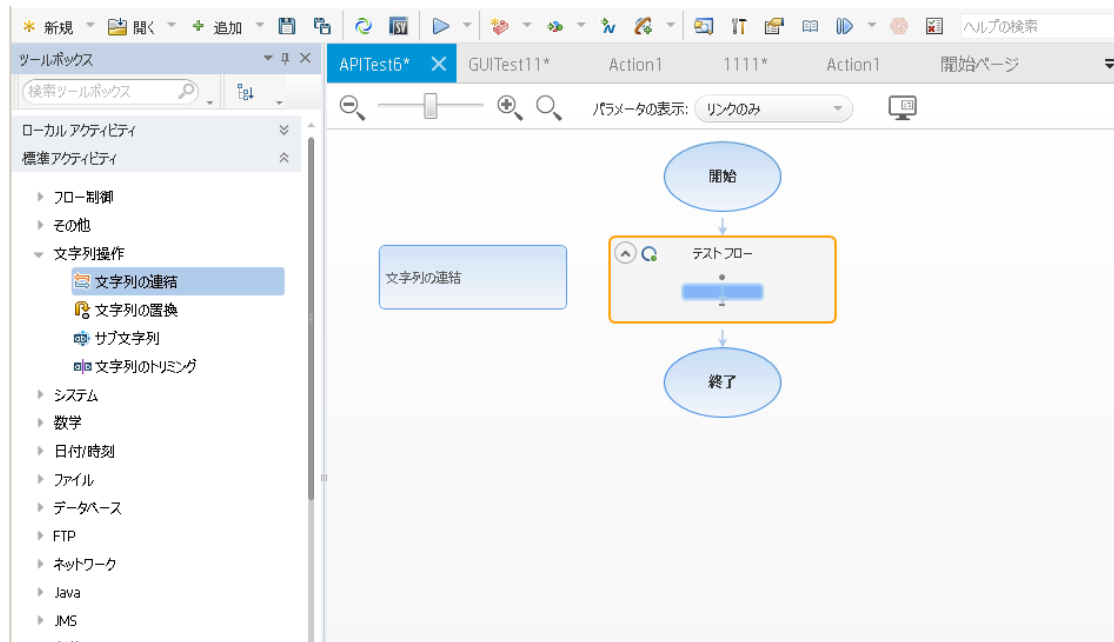
## レッスン 2：単純な API テスト・ステップの作成

UFT の API テストでは、テストを視覚的に作成できます。テスト・ステップは、マスタ・テスト・フロー全体を表示するキャンバスに表示されます。

実際のテスト・ステップの作成は、次の 2 つの部分で構成されます。

1. 適切なアクティビティをキャンバスのテスト・フローにドラッグしてテスト・ステップを作成します。

「ツールボックス」表示枠には、テストで利用できるすべてのアクティビティが含まれます。キャンバスに含まれるアクティビティのリストから、アクティビティをキャンバスにドラッグして、テスト・フローの任意の場所に追加します。



2. ステップのプロパティ（ステップの入力およびチェックポイント）を追加する

テスト・ステップをキャンバスにドラッグすると、キャンバスでステップを選択するたびにステップのプロパティが「プロパティ」表示枠に表示されます。

入力	値
▼ プロパティ	
プレフィックス	
サフィックス	

チェックポイント	検証	期待値
▼ プロパティ		
結果	<input type="checkbox"/>	=

各ステップには、次の 2 種類のプロパティが存在します。


入力	これらのプロパティは、UFT でステップを実行する際にアクティビティで使用する値を設定するのに使います。
チェックポイント	ステップが実行された後でこれらの値を実際の値と比較し、ステップが正しく実行されたこと（正しく実行されなかったこと）を確認します。 チェックポイントのプロパティは、テストを実行する際のオプションです。

このレッスンでは、これらの基本機能を使用して基本的なテストを作成します。


### 1. UFT を起動し、Book Flights テストを開きます。

- a. UFT が開いていない場合は、「すべてのテスト用のソリューションの作成」(25ページ)の手順に沿って開きます。
- b. 開始ページの「最近のテスト/コンポーネント」領域で、「Basic」をクリックします。  
「Basic」テストがドキュメント表示枠に表示されます。

### 2. アクティビティをキャンバスに追加し、プロパティを定義します。


- a. ツールバーで、「ツールボックス」ボタンをクリックします。「ツールボックス」表示枠が開き、アクティビティのカテゴリが表示されます。
- b. 「ツールボックス」表示枠で、「文字列操作」ノードを展開します。
- c. 文字列操作アクティビティのリストから、「文字列の連結」アクティビティをキャンバスにドラッグします。

文字列の連結という名前の新しいブロックがキャンバスに追加され、「プロパティ」表示枠でテスト・ステップのプロパティを指定できるようになります。

- d. 「プロパティ」表示枠で、「入力/チェックポイント」タブを選択します。
- e. 「入力/チェックポイント」タブの「入力」セクション（上部）で、ステップのプロパティとして次の値を入力します。
  - **A** : Hello（後ろにスペースを 1 つ入れる）
  - **B** : World.
  - **チェックポイント** : Hello world.


### 3. 別のアクティビティをキャンバスに追加し、そのプロパティを定義します。

入力とチェックポイントのプロパティに加え、テスト・ステップに対して別のプロパティを定義できます。

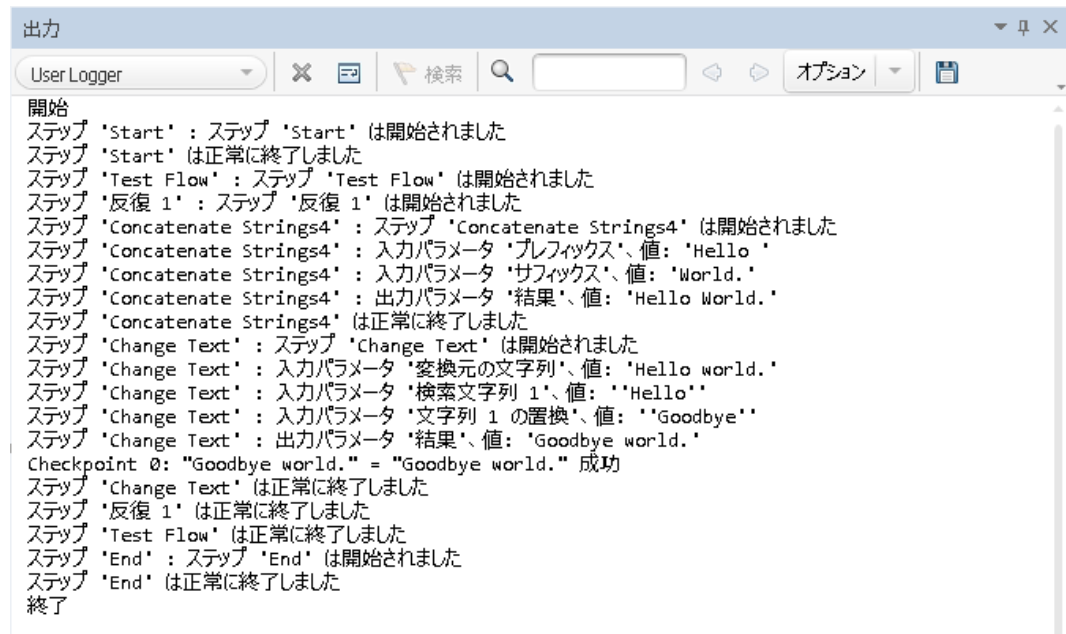
- a. 「ツールボックス」表示枠で、「文字列操作」ノードを展開します。
- b. 文字列操作アクティビティのリストから、「文字列の置換」アクティビティをダブルクリックしてキャンバスに追加します。「プロパティ」表示枠に「入力/チェックポイント」タブが開き、ステップの入力プロパティとチェックポイント・プロパティが表示されます。
- c. 「プロパティ」表示枠で、「一般」タブを開きます。
- d. 「名前」プロパティ行で、「文字列の置換」を「テキストの変更」に変更し、**Enter** キーを押します。キャンバスのステップの名前が「テキストの変更」に変わります。
- e. 「入力/チェックポイント」タブを開きます。



- f. **【入力/チェックポイント】** タブで、プロパティとして次の値を入力します。
  - **変換元の文字列**：Hello world.
  - **検索文字列**：Hello
  - **置換文字列**：Goodbye
  - **大文字と小文字を区別する**：False
4. **テキストの変更ステップにチェックポイントのプロパティを追加します。**
  - a. **【入力/チェックポイント】** タブの **【チェックポイント】** セクションの **【結果】** 行で、**【検証】** カラムのチェックボックスを選択します。これにより、このステップのチェックポイントが有効になります。
  - b. **【期待値】** カラムに、期待される結果「Goodbye world.」を入力します。
5. **テストを保存します。**

**【ファイル】** > **【保存】** を選択します。
6. **テストを実行し、結果を表示します。**
  - a. ツールバーで **【実行】** ボタン  をクリックします。**【実行】** ダイアログ・ボックスが開きます。
  - b. **【実行】** ダイアログ・ボックスで、**【オプション】** をクリックしてダイアログ・ボックスを展開します。
  - c. **【結果保管場所】** タブで、**【一時実行結果フォルダ】** オプションを選択します。
  - d. **【実行】** をクリックして、テストをコンパイルし実行します。

テストの実行中、[出力] 表示枠にコンパイル・ログが表示されます。コンパイル・ログに表示される各行で、前のステップで指定した入力およびチェックポイントのプロパティ値を確認できます。



テストの実行が完了すると、実行結果が開きます。

- e. 実行結果で、[テスト フロー] を選択してテスト内のステップを表示します。
- f. テスト・ステップのツリーで、[文字列の連結] ステップ・ノードを選択します。実行結果に、ステップのサマリが表示されます。
- g. ステップのサマリ領域で、[Concatenate Strings.xml] リンクをクリックします。ドキュメント表示枠に別のタブが開きます。
- h. 新しく開いたタブで、[プレフィックス] 行と [サフィックス] 行を確認します。以前の手順で指定したのと同じ入力値と出力値を使用してステップが実行されることを確認します。
- i. テスト・ステップのツリーで、[テキストの変更] ステップを選択します。
- j. ステップのサマリ領域で、[Change Text.xml] リンクをクリックします。
- k. 新しく開いたタブで、このステップで使用した入力プロパティを再度確認します。
- l. [テキストの変更] ステップの下に [チェックポイント] ステップを選択します。
- m. ステップのサマリ領域で、[Checkpoint.xml] リンクをクリックします。
- n. 新しく開いたタブで、実行されたテスト・ステップの詳細を確認します。結果にチェックポイントの成功（緑のチェックマークが表示される）と前の手順で入力した期待される文字列が表示されることを確認します。
- o. 結果の確認が終了したら、実行結果のタブを閉じます。

# レッスン 3：標準アクティビティを使用した API テスト・ステップの作成

API テストを作成する際には、アプリケーションの非 GUI（サービス）層が正常に動作していることを確認するためにテストを行います。データベースの呼び出し、Web サービスの呼び出し、プログラムのオープン、Web 経由でのメッセージの送信など、アプリケーションを実行する見えないプロセスがいくつも存在します。

テストの作成を支援するため、UFT には、テスト・ステップの設計時に使用する標準 API アクティビティが用意されています。このレッスンでは、これらの標準アクティビティを使用して基本的なテストを作成します。

このレッスンには、次の手順が含まれています。

- [UFT API テストの標準アクティビティ - 概要](#) .....156
- [練習 3a：標準アクティビティによるテストの作成](#) .....157

## UFT API テストの標準アクティビティ - 概要

API テストを作成する場合、すべてのテストで利用できる標準アクティビティが用意されています。次の標準アクティビティは一般的なアプリケーション・プロセスをテストします。

- **フロー制御**アクティビティ：待機，中断，条件ステップなど
- **文字列操作**アクティビティ：文字列の連結，文字列の置換など
- **ファイル・システム**・アクティビティ：ファイル・システムを使用して実行されるプロセス用
- **データベース**・アクティビティ
- **FTP** アクティビティ
- **ネットワーク**・アクティビティ：HTTP 要求，SOAP 要求など
- **JSON** および **XML** アクティビティ：XML または JSON 文字列/ファイルとのやり取りを伴うアプリケーション・プロセス用
- **数値演算**アクティビティと**日付/時刻**アクティビティ
- **その他の**アクティビティ：カスタム・コード・アクティビティ，プログラムの実行アクティビティとプログラムの終了アクティビティ，レポート・アクティビティが含まれます。

以下のテクノロジー固有のアクティビティも用意されています。

- **Java クラスの呼び出し**アクティビティ：アプリケーション内で使用される Java プロセスをテストする場合
- **JMS** (Java メッセージ・サービス) アクティビティ
- **IBM Websphere MQ** アクティビティ
- **SAP** アクティビティ：SAP サーバに格納された SAP iDOC または RFC にアクセスする場合
- **負荷テスト**・アクティビティ：(LoadRunner スクリプトへの変換後に) HP LoadRunner でテストを実行する場合
- **HP 自動テスト・ツール**・アクティビティ (テストの一部として使用するために、UFT，QuickTest Professional，Service Test，LoadRunner から GUI テストまたはアクション，API テストまたはアクション，または仮想ユーザ・ジェネレータ・スクリプトを呼び出す場合)


テストにインポートできるカスタム・アクティビティも存在しますが、これらのアクティビティについては、後のレッスンで詳しく説明します。

## 練習 3a：標準アクティビティによるテストの作成

「[レッスン 2：単純な API テスト・ステップの作成](#)」(150ページ)では、UFT API テスト用ユーザ・インタフェースの概要と、このユーザ・インタフェースを使用してテスト・ステップの作成と実行を行う手順について学習しました。

このレッスンでは、この知識を活用して、標準アクティビティを使って基本的な API テストを作成します。

### 1. 新しいテストを作成します。

- a. ツールバーで **新規作成** ボタン  **新規** をクリックします。[新規テスト] ダイアログ・ボックスが開きます。
- b. [新規テスト] ダイアログ・ボックスで、**[API テスト]** を選択します。
- c. テストに関する次の詳細情報を入力します。
  - **名前**：Standard
  - **場所**：C:\%HOMEPATH%\My Documents\Unified Functional Testing
- d. **[作成]** をクリックします。

キャンバスに空のテストが開き、空のテスト・フローが表示されます。このテストは、ソリューション・エクスプローラに**無題のソリューション**のサブノードとしても表示されます。


### 2. テスト・フローにステップを作成します。

このステップでは、特定の文字列を検索し、その文字列を置換して、文字列置換の結果をファイルに書き込むアプリケーション・プロセスのテストを作成します。

このテストでは、次の 3 つのアクティビティが必要になります。

- 文字列を置換する
- 結果を保存するファイルを作成する
- 結果をファイルに書き込む

これらのアクティビティはすべて、UFT の標準アクティビティで提供されています。

- a. [ツールボックス] 表示枠が開いていない場合は、UFT ウィンドウの左下にある **[ツールボックス]** タブをクリックして、[ツールボックス] 表示枠を開きます。
- b. [ツールボックス] 表示枠で、**[文字列操作]** アクティビティ・ノードを展開します。
- c. 文字列操作アクティビティで、**[文字列の置換]** アクティビティをキャンバスにドラッグします。新しいブロックがキャンバスのテスト・フローに追加され、[プロパティ] 表示枠に文字列の置換ステップに対する **[入力/チェックポイント]** タブ  が表示されます。

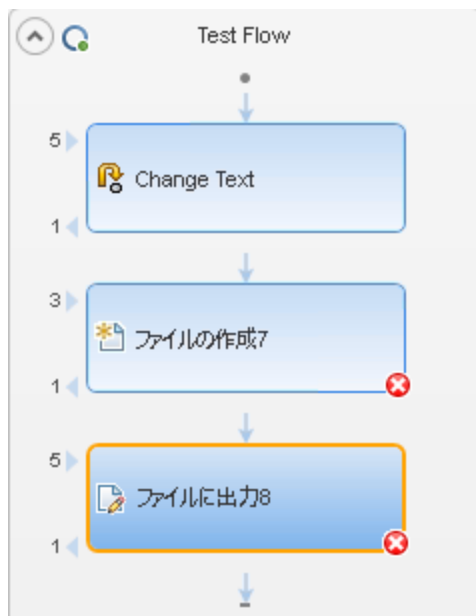
- d. [ツールボックス] 表示枠で、[ファイル] アクティビティを展開します。
- e. [ファイル] アクティビティで、次のアクティビティをキャンバスにドラッグします。

- **ファイルの作成**
- **ファイルに出力**


各アクティビティに対応する 2 つのブロックがキャンバスに追加されます。これらのアクティビティは、赤いアラート・アイコン付きで表示されます。



アクティビティをキャンバスに追加した後のテスト・フローは、次のようになります。




### 3. 文字列の置換アクティビティのプロパティを入力します。

- a. キャンバスで、[文字列の置換] アクティビティを選択します。[プロパティ] 表示枠に [入力/チェックポイント] タブ  が開きます。
- b. [入力/チェックポイント] タブで、ステップの入力プロパティを入力します。
  - **変換元の文字列**：Hello world.
  - **検索文字列**：Hello
  - **置換文字列**：Goodbye

**注:** このアクティビティでは、チェックポイントのプロパティを使用する必要はありません。

### 4. ファイルの作成ステップのプロパティを入力します。

このステップにより、前のステップの文字列の置換操作の結果を出力するファイルを作成できます。次のステップで使用するため、指定されたディレクトリにファイルを作成します。

- a. キャンバスで、**【ファイルの作成】** ステップを選択します。**【プロパティ】** 表示枠に**【入力/チェックポイント】** タブ  が開きます。
- b. ファイル・システムで、**C:\%HOMEPATH%\My Documents\Unified Functional Testing** フォルダを開きます。
- c. 前のステップで示した Unified Functional Testing フォルダ内に、**Tutorial\_Files** という名前のフォルダを作成します。次のステップでこのフォルダを使用してファイルを作成します。
- d. **【入力/チェックポイント】** タブで、ステップの入力プロパティを入力します。
  - **フォルダ・パス** : C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial\_Files


**注:** 参照ボタンを押してフォルダに直接移動することで、フォルダを指定することもできます。

- **ファイル名** : Standard Test Result
  - **上書き** : true
- e. **【入力/チェックポイント】** タブの**【チェックポイント】** セクションの**【結果】** 行で、**【検証】** カラムのチェックボックスを選択します。標準設定値をそのまま使用します。

これにより、テストを実行して実行結果を表示し、ファイルの作成ステップが正常に実行されたかどうかを確認できます。

## 5. ファイルに出力ステップのプロパティを入力します。

このステップでは、**【文字列の置換】** ステップで生成された文字列を**【ファイルの作成】** ステップで作成したファイルに書き込みます。


- a. キャンバスで、**【ファイルに出力】** ステップを選択します。**【プロパティ】** 表示枠に**【入力/チェックポイント】** タブ  が開きます。
- b. **【入力/チェックポイント】** タブの**【入力】** セクションで、ステップの入力プロパティを入力します。
  - **コンテンツ** : Goodbye world.
  - **ファイル・パス** : 前のステップで作成したファイルを使用します。次の内容を手動で入力する必要があります : C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial\_Files\Standard Test Result
  - 他のプロパティはすべて標準設定のままとします。
- c. **【入力/チェックポイント】** タブの**【チェックポイント】** セクションの**【結果】** 行で、**【検証】** カラムのチェックボックスを選択します（標準設定値をそのまま使用します）。

これにより、テストを実行して実行結果を表示し、ファイルに出力操作が正常に実行されたかどうかを確認できます。

## 6. テストを保存します。

【ファイル】 > 【保存】を選択します。

## 7. テストを実行します。

- a. ツールバーで【実行】ボタンをクリックします。【実行】ダイアログ・ボックスが開きます。
- b. 【実行】ダイアログ・ボックスの【結果保管場所】タブで、【一時実行結果フォルダ】が選択されていることを確認します。
- c. 【実行】をクリックして、テストをコンパイルし実行します。  
テストの実行が完了すると、実行結果が開きます。
- d. ファイル・システムで、**C:\%HOMEPATH%\My Documents\Unified Functional Testing\Tutorial\_Files\Standard Test Result** ファイルを開きます。

このファイルのドキュメントの最初の行に文字列 "Goodbye world" が記載されているはずで  
す。これは、UFT で提供される標準 API アクティビティを使用して、UFT が実際のアプリ  
ケーション（いずれのアプリケーションの場合も同様）に対してテスト・ステップを実行し  
たことを示します。

以上で標準アクティビティを使用したアプリケーション・プロセスの基本的なテストの作成は  
完了です。続いて「[レッスン 4 : API テスト・ステップのパラメータ化](#)」(161ページ)で、テスト  
でのデータの使用とテスト・ステップのパラメータ化について学習します。



# レッスン 4 : API テスト・ステップのパラメータ化

「[レッスン 2 : 単純な API テスト・ステップの作成](#)」(150ページ)と「[レッスン 3 : 標準アクティビティを使用した API テスト・ステップの作成](#)」(155ページ)では、必要な入力およびチェックポイントの値を手動で入力することで、API テスト・ステップの値を指定する方法について学習しました。

ただし、次のように、入力およびチェックポイントの値を他のソースから提供することもできます。

- テストに含まれるデータ・ソース (Excel ファイル, XML, データベースのデータ・ソース, またはローカルに作成したテーブル)
  - 前のステップの出力
  - 上記のすべて (必要なデータの手動入力, データ・ソース, 前のステップの出力) の組み合わせ
- アプリケーション・プロセスの入力はデータ・ソースや前のアプリケーション・プロセスの結果などから取得されるため、これらのデータ・ソースを使用してステップの入力値と出力値を読み込むことで、アプリケーションの動作を再現できます。

このレッスンでは、さまざまな方法を使用して API テスト・ステップをパラメータ化する方法について学習します。

このレッスンには、次の手順が含まれています。

- [API テスト・ステップのパラメータ化 - 概要](#) ..... 162
- [練習 4a : データ・ソースを元にしたテスト・ステップのパラメータ化](#) ..... 162
- [練習 4b : 前のステップの出力を元にしたテスト・ステップのパラメータ化](#) ..... 173
- [練習 4c : カスタム式を使用した複数のソースによるテストのパラメータ化](#) ..... 176

## API テスト・ステップのパラメータ化 - 概要

API テストでテスト・ステップのプロパティの値を提供する場合、[プロパティ] 表示枠でこれらの値を手動で入力するのが標準的な方法です。

しかし、この方法で実地的なアプリケーションのテストを実行できるとは限りません。多くのアプリケーションでは、内部 API プロセスは、データ・ソース、他のテスト・ステップの出力、またはその両方などから動的に情報を受け取ります。

そのため、UFT には、次のように、ステップの値を提供する（パラメータ化する）異なる複数の方法が用意されています。

<b>手動入力</b>	ステップの値を手動で入力する場合、各ステップを選択し、入力プロパティとチェックポイント・プロパティの値に適した値を入力または選択します。  アプリケーションのプロパティが変わるたびにテスト内の個々のステップやプロパティを更新する必要があるため、この方法はテストのメンテナンスが容易ではありません。
<b>データ・ソースへのリンク</b>	ステップのプロパティをデータ・ソースにリンクする場合、UFT はテスト実行中にデータ・ソースから値を取得し、データ・ソースで提供される値を使用します。データ・ソースに複数のデータ・セットが含まれる場合は、テストを複数回反復して実行することで、入力とチェックポイントにさまざまな値を提供し、さまざまなデータ入力を使用してアプリケーションがどのように実行されるかを確認することができます。  個々のテスト・ステップを更新せずに、データ・ソースの値を更新するだけで済むため、この方法はテスト・メンテナンスが比較的容易です。
<b>前のステップの出力へのリンク</b>	ステップのプロパティを前のステップの出力にリンクする場合、UFT はステップの出力から値を取得し、これらの値をテスト実行中に使用します。これにより、アプリケーションの API プロセスの出力が別のプロセスに入力として値を渡す、実際のアプリケーション動作を再現できます。
<b>複数のソースへのリンク</b>	アプリケーションの入力とチェックポイントが複数のソース（静的な文字列、データ、他のステップ/プロセスの出力）から提供される場合は、カスタム式を作成してテスト内で実行できます。UFT はカスタム式を使用して値を提供し、テスト実行中にこれらの値を使用します。

## 練習 4a：データ・ソースを元にしたテスト・ステップのパラメータ化

「[API テスト・ステップのパラメータ化 - 概要](#)」(162ページ)で学習したように、テスト・ステップに値を提供する方法の1つとして、ステップのプロパティ値をデータ・ソースにリンクする方法があります。これにより、データ・ソースの構造に応じて、異なる複数の値でテスト・ステップを実行できます。

API テストでは、次に示す異なる複数の種類のデータ・ソースを追加できます。

- Excel シート
- XML ファイルまたはスキーマ

- データベース
- ローカルに保存したデータ・テーブル

これらをそれぞれテストに追加し、テスト内のすべてのステップで利用することができます。

このレッスンでは、テスト・ステップをデータ・ソースにリンクします。

### 1. UFT を起動し、標準テストを開きます。

- a. UFT が開いていない場合は、「すべてのテスト用のソリューションの作成」(25ページ)の手順に沿って開きます。
- b. 開始ページの「最近のテスト/コンポーネント」領域で、「Basic」をクリックします。


「レッスン 3 : 標準アクティビティを使用した API テスト・ステップの作成」(155ページ)で作成した標準テストが、ドキュメント表示枠に別のタブとして表示されます。このテストは、ソリューション・エクスプローラにソリューション **Untitled** のサブノードとしても表示されます。

### 2. 文字列の連結ステップをテストに追加します。

- a. 「表示」 > 「ツールボックス」を選択して、「ツールボックス」表示枠を開きます。
- b. 「ツールボックス」表示枠で、「文字列操作」ノードを展開します。
- c. 「文字列操作」ノードで、「文字列の連結」アクティビティをキャンバスにドラッグして、「文字列の置換」アクティビティの上に配置します。

### 3. テストにデータ・ソースを追加します。

- a. 必要に応じて、「表示」 > 「データ」を選択し、「データ」表示枠を開きます。

- b. [データ] 表示枠で, [新規データ ソース] ボタン  をクリックして [ローカル テーブル] を選択します。[新規ローカル テーブル データ ソース] ダイアログ・ボックスが開きます。

新規ローカルテーブルデータソース ? ×

データソース名:

名前	データタイプ	説明
----	--------	----



OK キャンセル

- c. [データ ソース名] フィールドで, テーブル名「Concatenated Strings」を指定します。

- d. [データソース名] フィールドの下に[追加] ボタンをクリックします。新しい行がテーブル・グリッドに追加されます。


新規ローカルテーブルデータソース

データソース名:

名前	データタイプ	説明
カラム1	String	

OK キャンセル

- e. テーブル・グリッドで、[名前] フィールドに「プレフィックス」と入力します。[データタイプ] は [String] のまま変更しません。
- f. [追加] ボタンをもう一度クリックします。2 つ目の行がテーブルに追加されます。
- g. テーブル・グリッドで、2 つ目の行の [名前] フィールドに「サフィックス」と入力します。[データタイプ] は [String] のまま変更しません。
- h. [OK] をクリックしてダイアログ・ボックスを閉じ、テーブルをテストに追加します。
- テーブルがテストに追加され、[データ] 表示枠に [現在のテスト] データ・ソースのサブノードとして表示されます。

データ

現在のテスト

- Concatenated Strings

Prefix	Suffix

データ エラー

#### 4. 値をデータ・テーブルに追加します。

テスト・ステップでテスト・ステップ用の値を使用するには、データ・ソースに使用可能なデータが存在している必要があります。ローカルに作成して保存したテーブルの場合、データを追加する必要があります。



- a. [データ] 表示枠で、[現在のテスト] ノードの下に [Concatenated Strings] ノードを選択します。[データ] 表示枠が更新され、選択したデータ・ソースのデータが表示されます（この時点では、テーブル内にデータは存在しません。）
- b. データ・グリッドで、[Prefix] カラム内をクリックします。
- c. [Prefix] カラムに、「Hello」と入力します。Hello の後には必ずスペースを 1 つ入れます。
- d. [Suffix] カラム内をクリックし、「World」と入力します。
- e. 次の行を追加で入力します。

Prefix	Suffix
Welcome	to UFT.
I am running	API tests.

注: A カラムの文字列の後に必ずスペースを 1 つ入れます。

## 5. テスト・ステップをデータ・ソースに接続します。

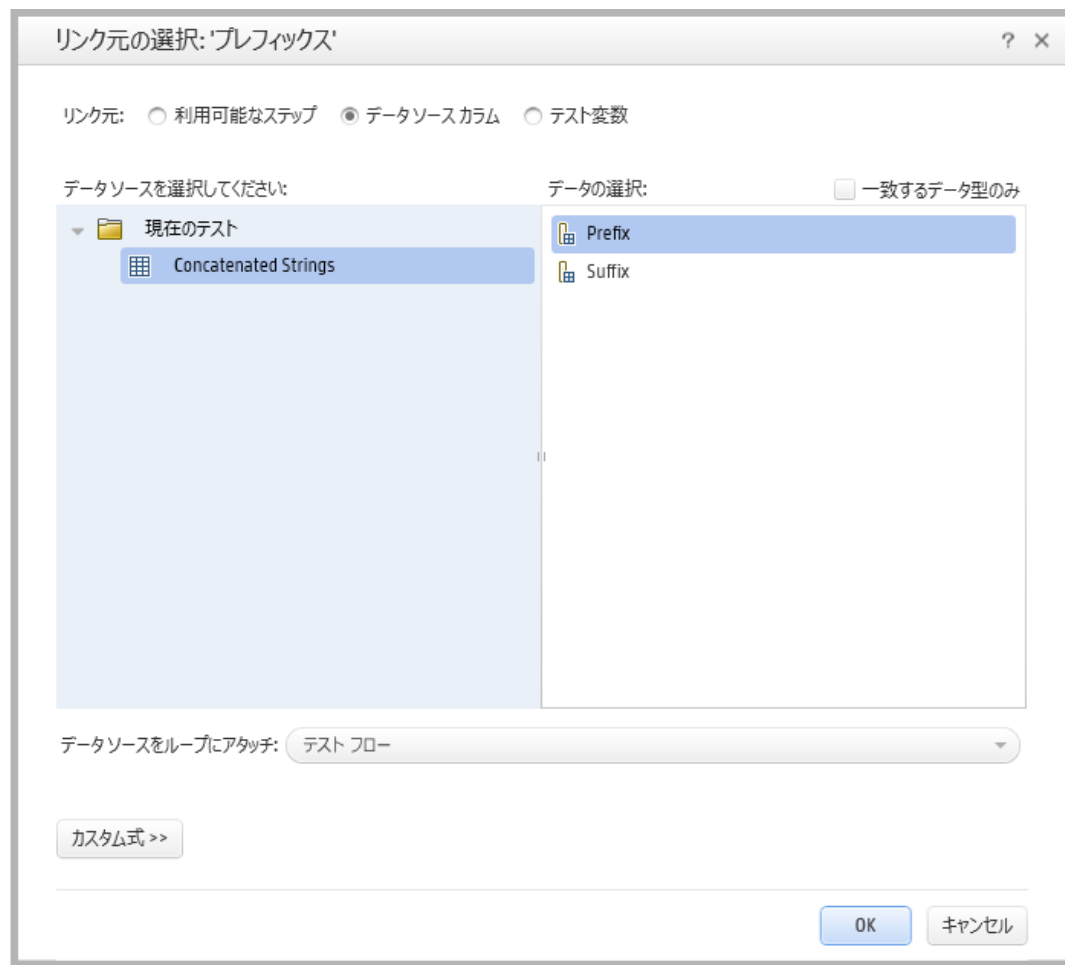
テストの実行時にデータ・ソースの値を使用するには、テスト・ステップのプロパティをデータ・ソースにリンクする必要があります。

- a. キャンバスで、[文字列の連結] ステップを選択します。[プロパティ] 表示枠に [入力/チェックポイント] タブ  が開きます。
- b. [入力/チェックポイント] タブの [入力] セクションで、[プレフィックス] プロパティの [値] セル内をクリックします。
- c. [値] セルで、[データ・ソースにリンク] ボタン  をクリックします。[リンク元の選択] ダイアログ・ボックスが開きます。

- d. [リンク元の選択] ダイアログ・ボックスで、[データ ソース カラム] ラジオ・ボタンを選択します。すべてのテストのデータ・ソースのリストが、ダイアログ・ボックスの左側に表示されます。



- e. **【データソースを選択してください】** 表示枠（左側）で，**【Concatenated Strings】** ノードを選択します（これは前のステップで作成したデータ・ソースです）。右の表示枠にデータ・ソース・パラメータのリストが表示されます。



- f. **【データの選択】** 表示枠（右側）で，**【Prefix】** ノードを選択し，**【OK】** をクリックします。

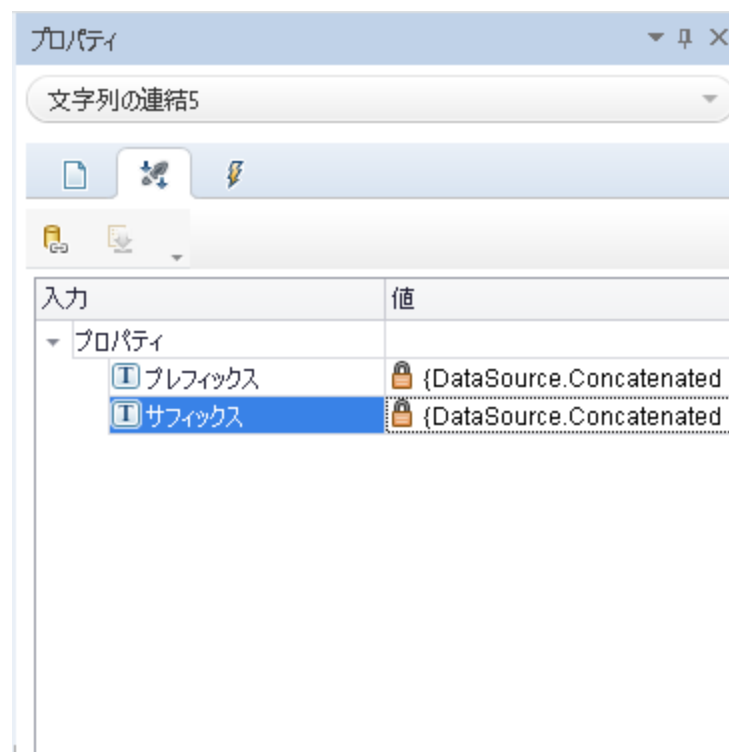
【入力/チェックポイント】タブの**【値】**カラムが更新され，データ・ソースへのリンクを示す式が表示されます。**【値】**カラム上にカーソルを置くと，次の式が表示されます。

```
{DataSource.Concatenated Strings.Prefix}
```

- g. 上記の手順を繰り返し，**【サフィックス】**プロパティをデータ・テーブルの**【Suffix】**カラムにリンクします。




完了すると、【入力/チェックポイント】タブにデータ・テーブルへのリンクが反映されます。

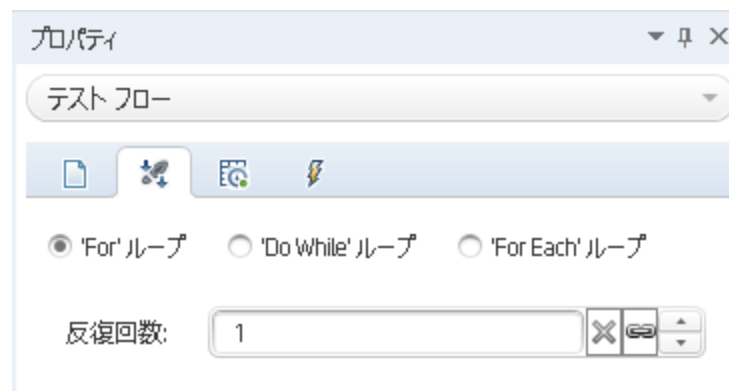


#### 6. テストの反復回数を設定します。

この後テスト全体を実行する場合、データ・テーブルの最初の行のデータを使用し、反復が1回のみ実行されます。

3つの行を入力したため、UFTによって異なるデータがどのように処理されるかを確認するには、テストを3回反復実行する必要があります。

- a. キャンバスで、**テスト・フロー**を選択します。【プロパティ】表示枠に【入力】タブ  が開きます。




- b. 【入力】タブで、【**For** ループ】ラジオ・ボタンを選択します。

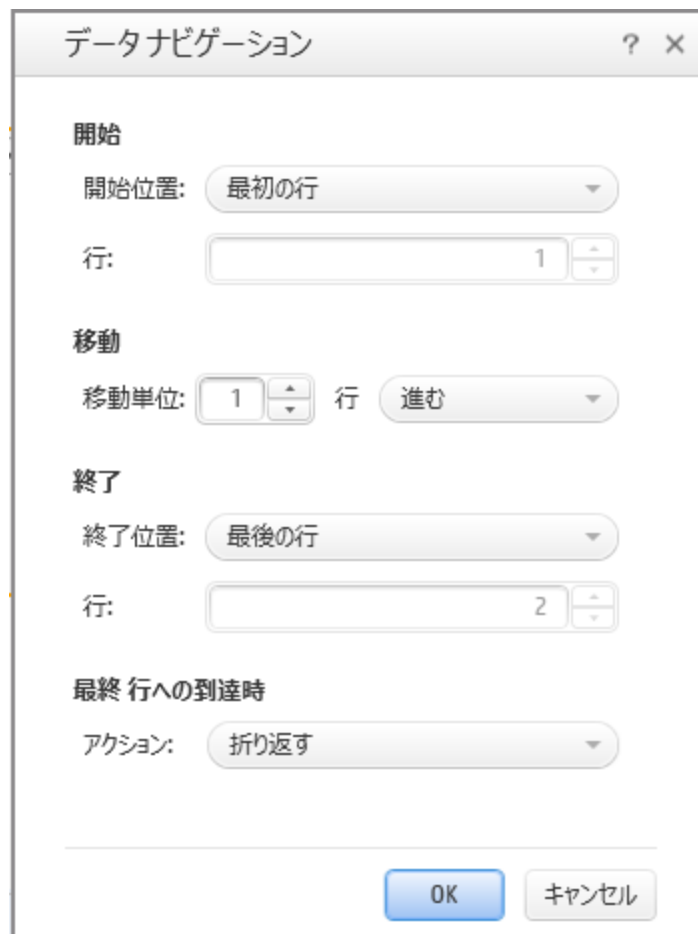
- c. **【反復回数】**に、反復回数として**3**を入力します。

ここで UFT でテストを実行すると、テストが 3 回反復実行され、1 回ごとにデータ・テーブルの新しい行が使用されます。

**注:** データ・ソースの行数と同じ反復回数を実行する必要はありません。ただし、標準設定で、UFT はデータ・テーブルの最初の行から実行を開始し、その後テストが終了するまで反復実行するたびに新しい行を使用します。

## 7. データ・テーブルのデータ・ナビゲーション・ポリシーを設定します。

- a. キャンバスで、テスト・フローを選択します。【プロパティ】表示枠に【入力】タブが開きます。
- b. 【プロパティ】表示枠で**【データ ソース】**タブを選択します。【データナビゲーション】グリッドが開きます。
- c. 【データ ソース】タブの【データナビゲーション】グリッドの**【データ ソース名】**コラムで、**【Concatenated Strings】**データ・ソースを選択し、**【編集】**をクリックします。【データナビゲーション】ダイアログ・ボックスが開きます。



The dialog box titled "データナビゲーション" (Data Navigation) contains the following settings:

- 開始 (Start):**
  - 開始位置 (Start position): 最初の行 (First row)
  - 行 (Row): 1
- 移動 (Move):**
  - 移動単位 (Move unit): 1 行 (1 row)
  - 進む (Advance): 進む (Advance)
- 終了 (End):**
  - 終了位置 (End position): 最後の行 (Last row)
  - 行 (Row): 2
- 最終 行への到達時 (When reaching the final row):**
  - アクション (Action): 折り返す (Wrap)


Buttons at the bottom: OK, キャンセル (Cancel).

- d. 【データナビゲーション】ダイアログ・ボックスで、次のプロパティを設定します。

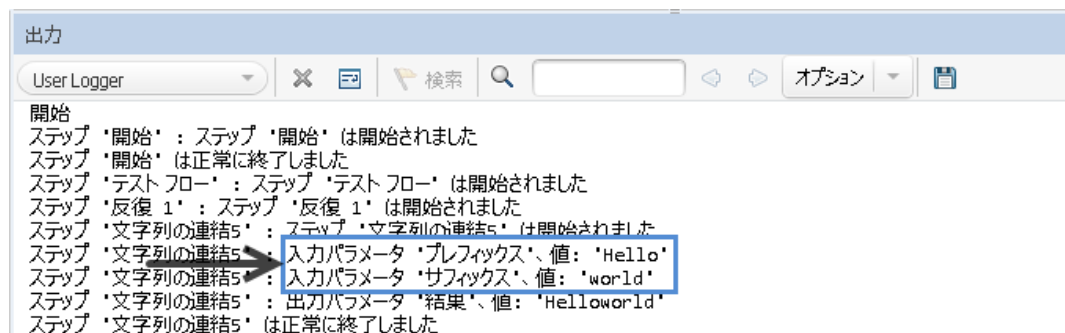
開始位置	最初の行
行	1
移動単位	1 行進む
終了位置	最後の行
最終行への到達時	折り返す

- e. 【OK】をクリックしてダイアログ・ボックスを閉じ、データ・ナビゲーション・ポリシーを更新します。

## 8. テストを実行します。

- ツールバーで【実行】 ボタンをクリックします。【実行】ダイアログ・ボックスが開きます。
- 【実行】ダイアログ・ボックスで、【オプション】をクリックしてダイアログ・ボックスを展開します。
- 【結果保管場所】タブで、【一時実行結果フォルダ】オプションを選択します。
- 【実行】をクリックしてテスト実行を開始します。

UFT でステップが順番に実行され、【文字列の置換】アクティビティの入力としてデータ・テーブル内の値が使用されます。UFT でテストを実行している間、UFT がデータ・テーブルから取得する値を【出力】表示枠で確認できます。



テストの実行が完了すると、【実行結果】が開きます。

## 9. 実行結果を分析します。

- 実行結果の【テスト フロー】で、テスト結果のツリーを確認します。

次のように、3つの【サマリ】ノードが存在します。



これは、前に設定した3つの反復に対応しています。

- b. 【アクション反復: 行 1】ノードを展開し、【文字列の連結】ノードを選択します。実行結果に、ステップのサマリが表示されます。
- c. キャプチャ・データに、【文字列の連結】アクティビティの入力値として使用されたデータが表示されます。

Name	Value
タイプ	HP.ST.Ext.BasicActivities.ConcatenateStringsActivity
ステップ ID	ConcatenateStringsActivity5
メッセージ	文字列は正常に連結されました
プレフィックス	'Hello'
サフィックス	'world.'
結果	'Helloworld.'
名前	'文字列の連結5'

使用されたデータは、データ・テーブルの最初の行と一致しています。

- d. 反復 2 および反復 3 の【文字列の連結】アクティビティで、このプロセスを繰り返します。ステップのキャプチャ・データの一部として表示される値がデータ・テーブルの 2 番目および 3 番目の行と一致していることを確認できます。

## 10. テストを保存します。

【ファイル】 > 【保存】を選択します。

これで、テスト・ステップをデータ・ソースに接続する方法の学習は終了です。引き続き「[練習 4b : 前のステップの出力を元にしたテスト・ステップのパラメータ化](#)」(173ページ)では、ステップのプロパティを前のステップの出力にリンクする方法を学習します。

## 練習 4b : 前のステップの出力を元にしたテスト・ステップのパラメータ化

「[練習 4a : データ・ソースを元にしたテスト・ステップのパラメータ化](#)」(162ページ)では、選択したステップのプロパティ値をデータ・ソースにリンクする方法を学習しました。



データからプロパティ値を提供する方法に加え、前のステップの出力からプロパティ値を取得することもできます。このレッスンでは、前のステップの出力を使用してステップの値をリンクする方法を学習します。

### 1. UFT を起動し、標準テストを開きます。

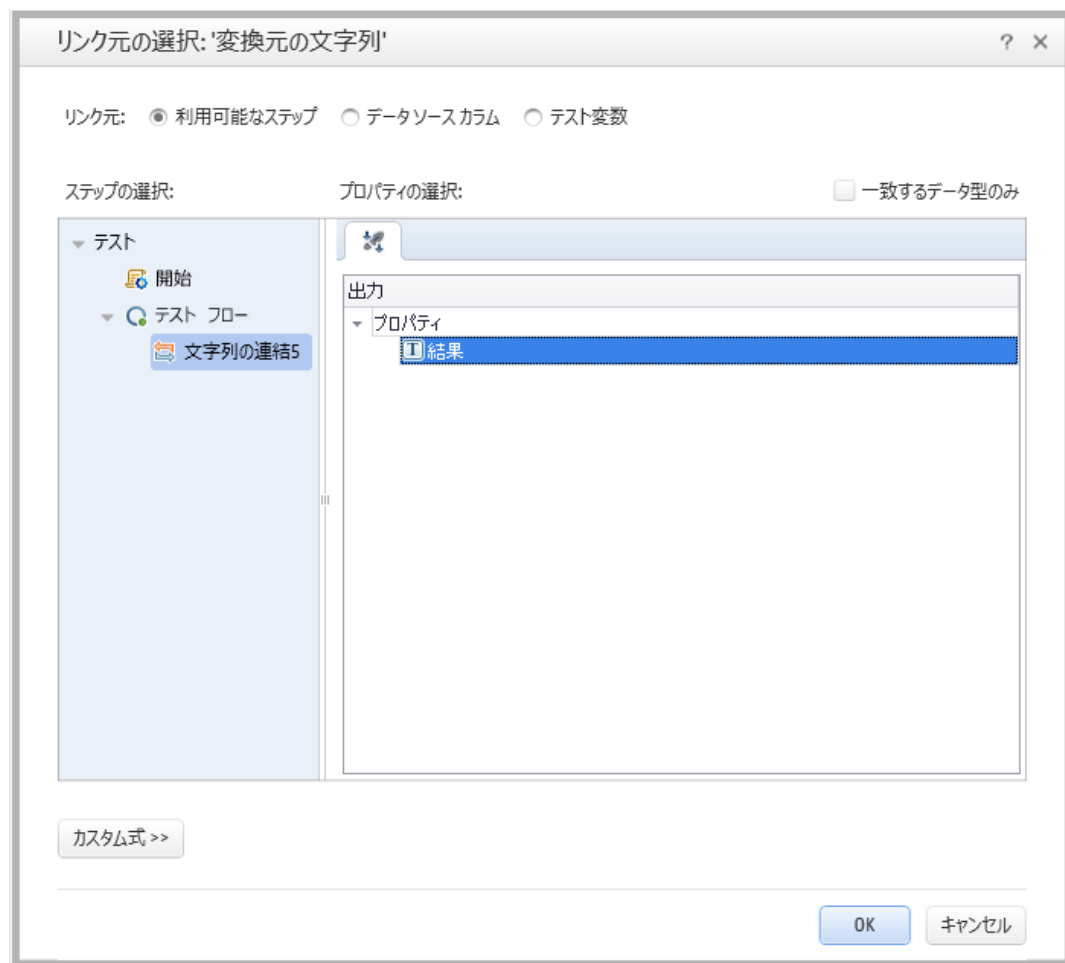
- a. UFT が開いていない場合は、「[すべてのテスト用のソリューションの作成](#)」(25ページ)の手順に沿って開きます。
- b. 開始ページの「[最近のテスト/コンポーネント](#)」領域で、「**Basic**」をクリックします。

「[レッスン 3 : 標準アクティビティを使用した API テスト・ステップの作成](#)」(155ページ)で作成した標準テストが、ドキュメント表示枠に別のタブとして表示されます。このテストは、ソリューション・エクスプローラに無題のソリューションのサブノードとしても表示されます。

### 2. 「**文字列の置換**」ステップのプロパティを「**文字列の連結**」テストにリンクします。

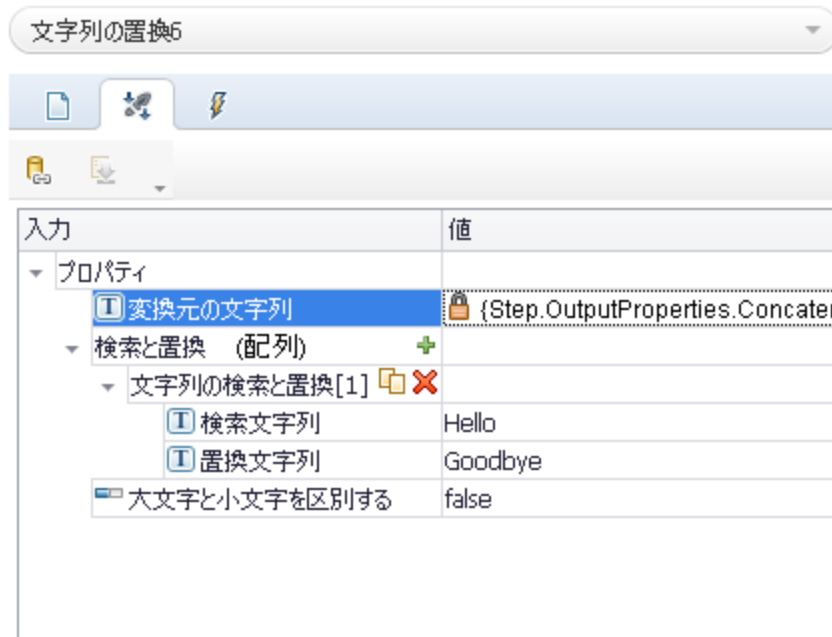
- a. キャンバスで、「**文字列の置換**」ステップを選択します。「プロパティ」表示枠に「**入力/チェックポイント**」タブ  が開きます。
- b. 「入力/チェックポイント」タブの「**入力**」領域で、「**変換元の文字列**」行を選択します。
- c. 「変換元の文字列」行の「**値**」セルで、「**データ ソースにリンク**」ボタン  をクリックします。「リンク元の選択」ダイアログ・ボックスが開きます。

- d. [リンク元の選択] ダイアログ・ボックスで, [利用可能なステップ] ラジオ・ボタンを選択します。ダイアログ・ボックスの左側に, 利用可能なステップのリストが表示されます。



- e. [ステップの選択] 表示枠 (左側) で, [文字列の連結] を選択します。[文字列の連結] アクティビティの出力プロパティのリストが表示されます。
- f. [プロパティの選択] 表示枠 (右側) で, [結果] 行を選択します。
- g. [OK] をクリックし, プロパティをリンクしてダイアログ・ボックスを閉じます。



【変換元の文字列】行の【値】カラム（【プロパティ】表示枠）が更新され、前のステップの出力へのリンクが反映されます。



【値】カラム上にカーソルを置くと、次の完全なリンク・ステートメントが表示されます。

```
{Step.OutputProperties.ConcatenateStringsActivity7.Result}
```

### 3. テストを実行し、リンクされたステップの結果を確認します。

- キャンバスで、テスト・フローを選択します。【プロパティ】表示枠に【入力】タブ  が開きます。
- 【入力】タブで、【For ループ】オプションが選択されていることを確認します。
- 【反復回数】に、1 と入力します。
- ツールバーで【実行】ボタン  をクリックします。【実行】ダイアログ・ボックスが開きます。
- 【実行】ダイアログ・ボックスの【結果保管場所】タブで、【一時実行結果フォルダ】オプションが選択されていることを確認します。
- 【実行】をクリックしてテスト実行を開始します。  
UFT でテスト・ステップが実行され、【文字列の置換】アクティビティの入力として【文字列の連結】ステップの出力が使用されます。  
テスト実行が完了すると、実行結果が開きます。
- 実行結果で、【テスト フロー】を表示します。
- 【サマリ】ノードの下に、【文字列の置換】ノードが表示されているのを見つけて選択します。実行結果に【文字列の置換】ステップの詳細が表示されます。

- i. ステップの詳細で、このテスト実行で使用された変換元の文字列を確認します。

#### Step Properties

Name	
タイプ	HP.ST.Ext.BasicActivities.ReplaceStringAct
ステップ ID	ReplaceStringActivity6
変換元の文字列	'Hello world.' ←
検索文字列 1	'Hello'
文字列 1 の置換	'Goodbye'

変換元の文字列は、「[練習 4a : データ・ソースを元にしたテスト・ステップのパラメータ化](#)」(162 ページ)で指定した【文字列の連結】アクティビティの出力です。

#### 4. テストを保存します。

【ファイル】 > 【保存】を選択します。

これで、テスト・ステップを相互にリンクする方法の学習は終了です。続いて、「[練習 4c : カスタム式を使用した複数のソースによるテストのパラメータ化](#)」(176 ページ)に進み、値の手動入力、データ・ソースへのリンク、前のステップの出力へのリンクを組み合わせるステップをパラメータ化する方法について学習します。

## 練習 4c : カスタム式を使用した複数のソースによるテストのパラメータ化

以前の練習では、テスト・ステップのプロパティ値をデータ・ソースまたは前のステップの出力にリンクする方法について学習しました。

しかし、静的な値の手動入力、データ・ソース、前のステップの出力など、ステップの値はその時々に応じてさまざまな場所から取得されます。このような場合には、複数のソースにリンクするカスタム式を作成できます。

この練習では、テストのデータ・テーブルのデータを使用する文字列置換操作の実行結果、静的なテキスト文字列の手動入力、および別のテスト・ステップの出力をファイルに書き込むカスタム式を作成します。

#### 1. UFT を起動し、標準テストを開きます。

- UFT が開いていない場合は、「[すべてのテスト用のソリューションの作成](#)」(25 ページ)の手順に沿って開きます。
- 開始ページの【最近のテスト/コンポーネント】領域で、【Basic】をクリックします。

「[レッスン 3 : 標準アクティビティを使用した API テスト・ステップの作成](#)」(155 ページ)で作成した標準テストが、ドキュメント表示枠に別のタブとして表示されます。このテスト



は、ソリューション・エクスプローラにソリューション **Untitled** のサブノードとしても表示されます。


## 2. **【ファイルに出力】テスト・ステップの入力プロパティをリンクします。**

「[レッスン 3 : 標準アクティビティを使用した API テスト・ステップの作成](#)」(155 ページ)では、次の 3 つのアクティビティを含むテストを作成しました。

- **文字列の置換** : 文字列を取得し、その一部を別の文字列で置き換える
- **ファイルの作成** : 置換した文字列を書き込むファイルを作成する
- **ファイルに出力** : 置換した文字列を書き込む

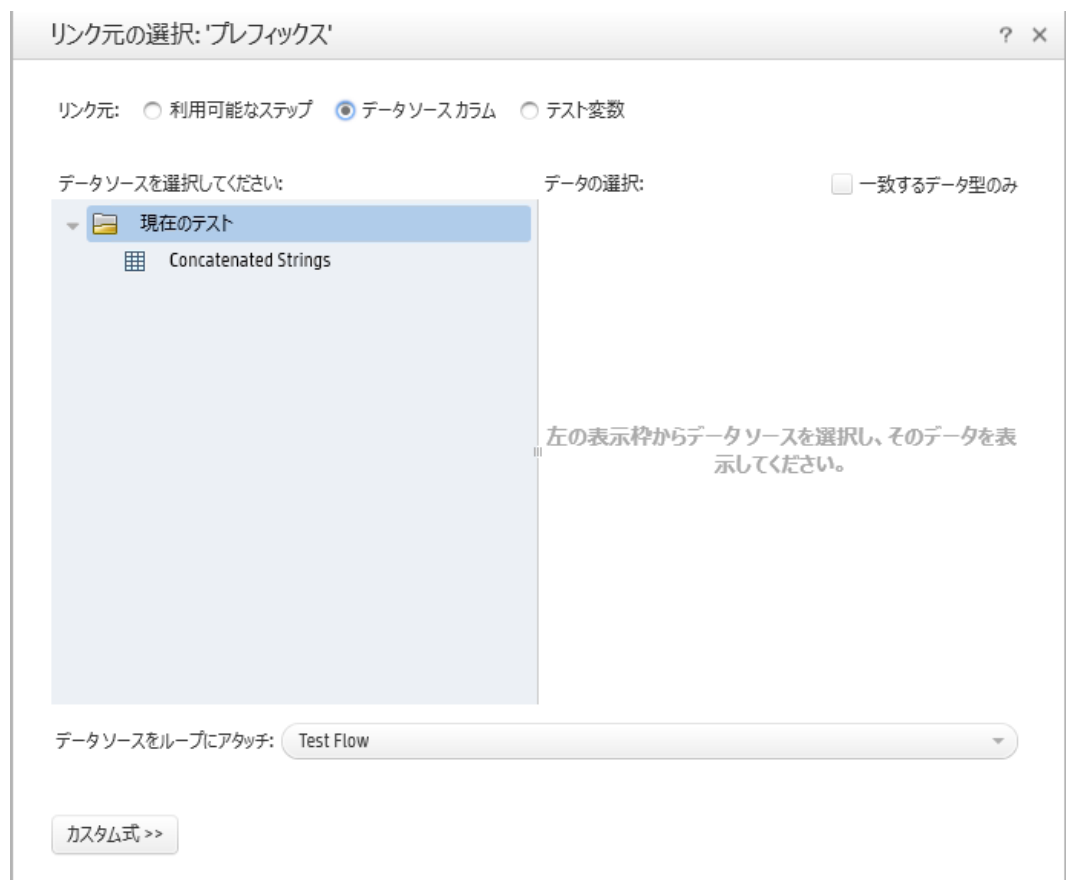
**注:** 「[練習 4a : データ・ソースを元にしたテスト・ステップのパラメータ化](#)」(162 ページ)では、文字列の連結ステップを追加しました。

このステップおよび後続のステップでは、**【ファイルに出力】** アクティビティを使用します。

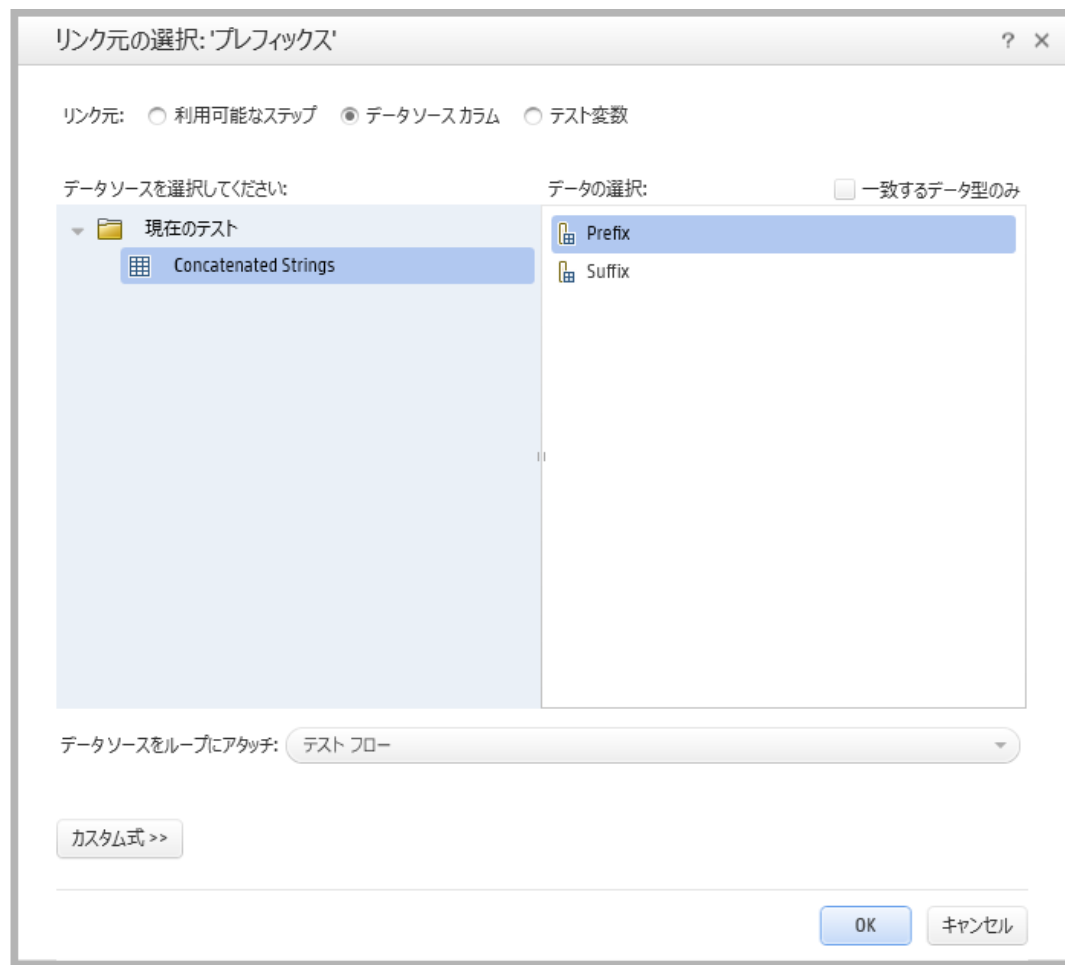
- a. キャンバスで、**【ファイルに出力】** ステップを選択します。**【プロパティ】** 表示枠に**【入力/チェックポイント】** タブ  が開きます。
- b. **【入力/チェックポイント】** タブの**【入力】** セクションで、**【コンテンツ】** 行を選択します。
- c. **【コンテンツ】** 行の**【値】** カラムで、**【データ ソースにリンク】** ボタンをクリックします。**【リンク元の選択】** ダイアログ・ボックスが開きます。

## 3. **データ・テーブルからのコンテンツ・プロパティ値に対応するカスタム式の最初の部分を作成します**

- a. **【リンク元の選択】** ダイアログ・ボックスで、**【データ ソース カラム】** オプションを選択します。**【リンク元の選択】** ダイアログ・ボックスに、すべてのテストのデータ・ソースのリストが表示されます。



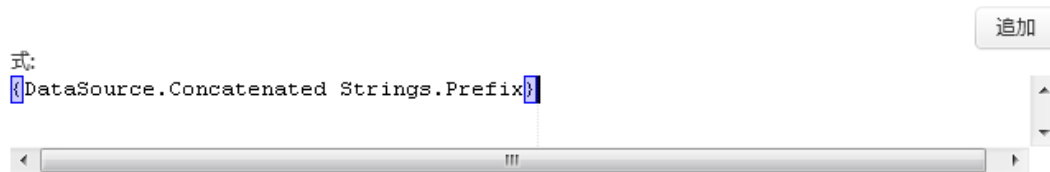
- b. **【データソースを選択してください】** 表示枠（左側）で，**【Concatenated Strings】** ノードを選択します。ダイアログ・ボックスに，Concatenated Strings データ・テーブル内のデータ・パラメータのリストが表示されます。



- c. **【データの選択】** 表示枠（右側）で，**【Prefix】** ノードを選択します。
- d. ダイアログ・ボックスの下部にある**【カスタム式】** ボタンをクリックします。ダイアログ・ボックスが拡張され，**【式】** 領域が表示されます。



- e. **【Prefix】** ノードが選択された状態で、**【式】** 領域の上にある**【追加】** ボタンをクリックします。式にプレフィックス・パラメータの式が追加されます。



- f. **【データの選択】** 表示枠で、**【Suffix】** ノードを選択します。  
 g. **【式】** 領域で**【追加】** ボタンを再度クリックします。式にサフィックス・パラメータの式が追加されます。



#### 4. 文字列を手動で入力してカスタム式の中央部分を追加します

カスタム式の中央部分では、静的なテキスト文字列を手動で追加します。

- a. **【リンク元の選択】** ダイアログ・ボックスの**【式】** 領域で、**{DataSource.Concatenated Strings.Suffix}** 式の後にスペースを1つ入力します。  
 b. テキスト「was replaced by」を入力し、続けてスペースを1つ入力します。

**注:** この文字列を入力した後で**【追加】** をクリックしないでください。**【追加】** ボタンをクリックすると、ダイアログ・ボックスの上の部分の表示枠で選択されている要素が追加されます。

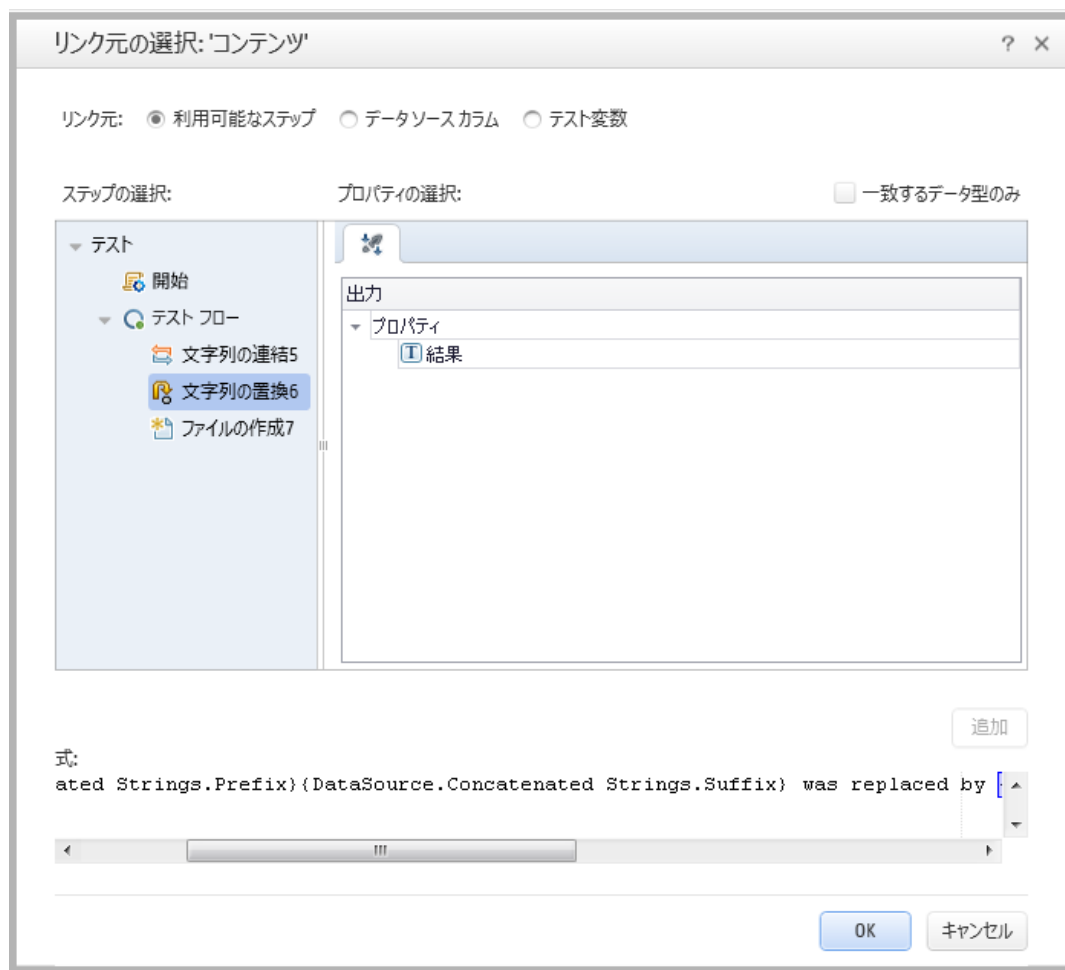
静的なテキスト文字列を入力した後のカスタム式は、次のようになります。

```
{DataSource.Concatenated Strings.Prefix}{DataSource.Concatenated Strings.Suffix} was replaced by
```

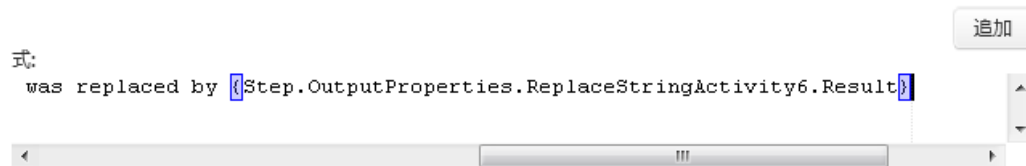
#### 5. 前のステップの入力にリンクしてカスタム式の最後の部分を追加します。

カスタム式の最後の要素では、前のステップの出力にリンクします。

- a. **【リンク元の選択】** ダイアログ・ボックスの上部の**【利用可能なステップ】** オプションを選択します。ダイアログ・ボックスに、すべての前のステップのリストが表示されます。



- b. **【ステップの選択】** 表示枠（左側）で、**【文字列の置換】** ノードを選択します。使用可能なプロパティのリストが表示されます。
- c. **【プロパティの選択】** 表示枠（右側）で、**【結果】** 行を選択します。
- d. **【式】** 領域で **【追加】** をクリックします。追加の式が前の式に追加され、**【文字列の置換】** ステップの出力へのリンクが反映されます。




- e. **【OK】** をクリックしてダイアログ・ボックスを閉じ、この式を**【ファイルに出力】** ステップのコンテンツ・プロパティの値として追加します。

コンテンツ・プロパティの**【値】** カラムに、更新された式が表示されます。


```
{DataSource.Concatenated Strings.Prefix){DataSource.Concatenated Strings.Suffix} was
```

```
replaced by {Step.OutputProperties.ReplaceStringActivity4.Result}
```

## 6. テスト実行の反復回数を設定します。


- キャンバスで、テスト・フローを選択します。[プロパティ] 表示枠に [入力] タブ  が開きます。
- [入力] タブで、[**For** ループ] オプションが選択されていることを確認します。
- [反復回数] に、**1** と入力します。

## 7. テストを実行して実行結果を表示します。

- ツールバーで [実行] ボタン  をクリックします。[実行] ダイアログ・ボックスが開きます。
- [実行] ダイアログ・ボックスの [結果保管場所] タブで、[一時実行結果オプション] が選択されていることを確認します。
- [実行] をクリックしてテスト実行を開始します。  
カスタム式で作成したリンクから [ファイルに出力] ステップのコンテンツ・プロパティの値を取得するステップが実行されます。  
テスト実行が完了すると、実行結果が開きます。
- 実行結果で、[テスト フロー] を表示します。
- テスト・フローで、[ファイルに出力] ノードを見つけます。実行結果に、ステップに関する関連情報のサマリが表示されます。
- ステップのサマリで、**Write to File.xml** リンクをクリックします。ステップのキャプチャ・データを含む別のタブが表示されます。
- 表示された別のタブで、このテスト実行で使用される [コンテンツ] プロパティの値を確認します。

Name	Value
タイプ	HP.ST.Ext.BasicActivities.FileWriteActivity
ステップ ID	FileWriteActivity8
コンテンツ	'Helloworld. was replaced by Goodbyeworld.'
ファイルパス	'True'
エンコード	'ASCII'
モード	'既存のファイルに追加'
新しい行を追加	False

## 8. テストを保存します。

[保存]  をクリックします。

## レッスン 5：API テストの実行

「[練習 3a：標準アクティビティによるテストの作成](#)」(157ページ) では、標準アクティビティを使用して基本的な API テストを作成しました。このテストの作成が済んだら、続いてテストを実行できます。

このレッスンでは、テストを実行して実行結果を表示する方法について学習します。

このレッスンには、次の手順が含まれています。

- [練習 5a：テストの実行](#) ..... 184
- [練習 5b：実行結果の操作](#) ..... 185
- [練習 5c：実行結果の分析](#) ..... 186

## 練習 5a：テストの実行

「練習 3a：標準アクティビティによるテストの作成」(157ページ)では、標準 API テスト・アクティビティを使用してテストを作成しました。「レッスン 4：API テスト・ステップのパラメータ化」(161ページ)では、異なる複数の方法を使用してテストのパラメータ化を行いました。

このレッスンでは、UFT をテスト実行用に準備する方法、および API テストを実行する方法について学習します。

### 1. UFT を起動し、Book Flights ソリューションを開きます。


- a. UFT を開きます（「すべてのテスト用のソリューションの作成」(25ページ)を参照）。
- b. 開始ページの「最近のテスト」領域で、「Flight Reservation Application」をクリックします。  
Flight Reservation Application ソリューションが開きます。これには、「練習 3a：標準アクティビティによるテストの作成」(157ページ)で作成した「Book Flights」テストが含まれます。

### 2. テストの実行モードを設定します。

テストを実行する前に、UFT でテストの実行方法を指定する必要があります。[リリース] モードまたは[デバッグ] モードでテストを実行できます。[リリース] モードではテストはすばやく実行されますが、[デバッグ] モードでは、UFT でテスト実行の前にデバッグ・ツールをロードする必要があるため、実行速度は遅くなります。

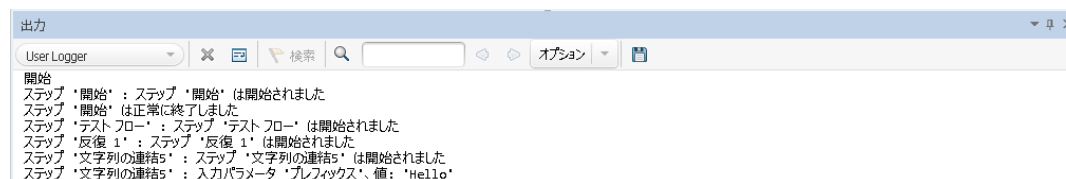
- a. [ツール] > [オプション] を選択します。[オプション] ダイアログ・ボックスが開きます。
- b. [オプション] ダイアログ・ボックスで、[API テスト] タブを選択します。
- c. [API] タブで、[一般] ノードを選択します。
- d. [一般] 表示枠で、[デバッグ モードでテストを実行] オプションを選択します。

### 3. テストの実行を開始します。

- a. ツールバーで[実行] ボタン  をクリックします。[実行] ダイアログ・ボックスが開きます。
- b. [実行] ダイアログ・ボックスで、[オプション] バーをクリックしてダイアログ・ボックスを展開します。
- c. [オプション] 領域で、[結果保管場所] タブをクリックします。これにより、テスト結果を保存する場所を指定できます。
- d. [結果保管場所] で、[新規実行結果フォルダ] オプションを選択します。標準設定フォルダ名をそのまま使用します。
- e. [実行] をクリックして[実行] ダイアログ・ボックスを閉じ、テストの実行を開始します。



UFT でテストが実行され、実行されたステップのログが出力表示枠に表示され（入出力パラメータを含む）、実行されたチェックポイントの結果が示されます。



テスト実行中にエラーが発生した場合は、ログの中で報告されます。これらのエラーを修正するために、関連するステップに戻ることができます。

テスト実行が完了すると、実行結果がドキュメント表示枠に別のタブとして表示されます。続いて「[練習 5b：実行結果の操作](#)」(185ページ)では、実行結果について詳しく学習します。

## 練習 5b：実行結果の操作

「[練習 5a：テストの実行](#)」(184ページ)では、作成した標準テストを実行しました。テスト実行が終了すると、実行結果にテスト実行の結果が自動的に表示されます。

**注:** 標準設定では、実行結果は HTML ベースのレポートに表示されます。実行結果は、[オプション] ダイアログ・ボックスの [実行セッション] 表示枠の Run Results Viewer に表示することもできます（[ツール] > [オプション] > [一般] タブ > [実行セッション] ノード）。このチュートリアルのレッスンは、HTML ベースのレポートに基づいています。

実行結果が開くと、次の内容が表示されます。



最初は、実行結果に次の内容が表示されます。

テスト・フロー	結果をツリー形式でグラフィカルに表したもので、テスト内のステップに合わせて構成されます。実行ごとに異なるデータ・セットを使用してテストを複数回実行するように UFT に指示できます。テストの各実行を反復といい、それぞれに番号が付きます。
エラー・リスト	すべてのエラーと警告のリストが表示されます。
ステップのサマリ情報	概要レベルの結果レポートで、テストに関する一般情報、成功または失敗したステップ、および各テスト・ステップに関する詳細が含まれます。 サマリには、テスト・ステップのキャプチャ・データへのリンクも含まれています。

作成したステップと指定したプロパティに従って UFT ですべてのステップを正しく実行できたので、このテストは成功となります。エラーが発生し、テストが正常に実行されなかった場合は、エラーが「出力」表示枠のログに表示されます。このような場合は、前に戻って、このチュートリアルの説明どおりにステップが設定されていることを確認してください。

実行結果の表示について理解できたら、続いて「[練習 5c：実行結果の分析](#)」(186 ページ)で実行結果の詳細について学習します。

## 練習 5c：実行結果の分析

このレッスンでは、テストを「[練習 5a：テストの実行](#)」(184 ページ)で実行したときに UFT によって実行されたステップをチェックします。

## 1. 特定のステップの結果を表示します。

- 「テスト フロー」の結果のツリーで、「テスト フロー」>「サマリ」>「反復 1」ノードを見つけ、このテストで実行されたすべてのステップを参照します。
- 「反復 1」ノードの下にある「文字列の置換」ノードを選択します。



実行結果に次の情報が表示されます。

- テスト・フロー：ステップが強調表示されます
- テスト・ステップのサマリ：強調表示されたステップの詳細が表示されます
- 選択したステップのキャプチャ・データを表示するリンク

## 2. 実行結果を閉じます。

ドキュメント表示枠で、実行結果を含むタブを閉じます。

# レッスン 6 : Web サービスの API テストの作成と実行

「[レッスン 3 : 標準アクティビティを使用した API テスト・ステップの作成](#)」(155ページ)では、標準 API アクティビティを使用してテストを作成する方法を学習しました。

ただし、標準アクティビティがアプリケーションで実行されるプロセスに適合しない場合もあります。このような場合、カスタム・アクティビティを UFT にインポートして使用する必要があります。広く使用されるサービス・アクティビティの種類の 1 つに Web サービスがあります。UFT では、サービスとそれぞれのメソッドを UFT にインポートして、これらをテストで使用できます。

このレッスンでは、Web サービス・テストを作成してテストを実行する方法について学習します。

このレッスンには、次の手順が含まれています。

- [練習 6a : Web サービス・テストの作成](#) ..... 189
- [練習 6b : Web サービスのインポート](#) ..... 189
- [練習 6c : Web サービスのテストの作成とパラメータ化](#) ..... 192
- [練習 6d : Web サービスのテストの実行](#) ..... 198

## 練習 6a : Web サービス・テストの作成

「すべてのテスト用のソリューションの作成」(25ページ)では、フライト予約アプリケーションのテスト用のソリューションを作成しました。「GUI 自動テストの作成と実行」(28ページ)では、フライト予約のユーザ・インタフェースのパフォーマンスをテストするさまざまな GUI テストを作成しました。

このレッスンでは、フライト予約アプリケーションのサービス (API) 層のテストに必要な最初の API テストを作成します。

### 1. UFT を起動し、フライト予約アプリケーション・ソリューションを開きます。

- UFT を開きます (「すべてのテスト用のソリューションの作成」(25ページ)を参照)。
- 開始ページの「最近のソリューション」領域で、「**Flight Reservation Application**」をクリックします。

Flight Reservation Application ソリューションが開きます。これには、「[レッスン 1 : GUI テストとアクションの作成](#)」(29ページ)で作成した **Book Flights** テストが含まれます。

### 2. 新規 API テストをソリューションに追加します。

- ツールバーで、「追加」ドロップダウン矢印  **追加** を選択し、「**新規テストの追加**」を選択します。「新規テストの追加」ダイアログ・ボックスが開きます。
- 「新規テストの追加」ダイアログ・ボックスで、「**API テスト**」を選択します。
- 「名前」フィールドで、テストに「Book Flights Web Service」という名前をつけます。
- 「場所」フィールドで「参照」ボタンをクリックし、**C:\%HOMEPATH%\Unified Functional Testing** フォルダに移動します。
- 「追加」をクリックしてテストを作成し、ソリューションに追加します。

フライト予約アプリケーション・ソリューションに Book Flights Web Service テストが個別ノードとして追加され、ドキュメント表示枠に別のタブとして表示されます。

**注:** ソリューションが自動的に保存されます。

テストの作成が完了し、Web サービスとメソッドを使用する準備ができました。続いて「[練習 6b : Web サービスのインポート](#)」(189ページ)で、Web サービスをテストにインポートする方法を学習します。

## 練習 6b : Web サービスのインポート

Web サービスをテストする前に、サービス記述とメソッドを UFT にインポートする必要があります。通常、サービス記述は WSDL (Web Service Description Language) ファイルに保存されます。こ

のファイルでは、サービスのメタデータとサービスの操作/メソッドを定義します。UFT はこの WSDL ファイルを読み取り、[ツールボックス] 表示枠にサービスのメソッドをアクティビティとして作成します。

この練習では、フライト予約アプリケーションのサービス WSDL ファイルを UFT にインポートします。

## 1. Flights API アプリケーションを起動します。

「[フライト予約アプリケーションの確認](#)」(23ページ)の説明に従って、HP Flights Service API アプリケーションを起動します。

**注:** テストを編集および実行する際に UFT からアプリケーションにアクセスできるように、チュートリアルを使用する際には、このアプリケーションを開いたままにしておいてください。

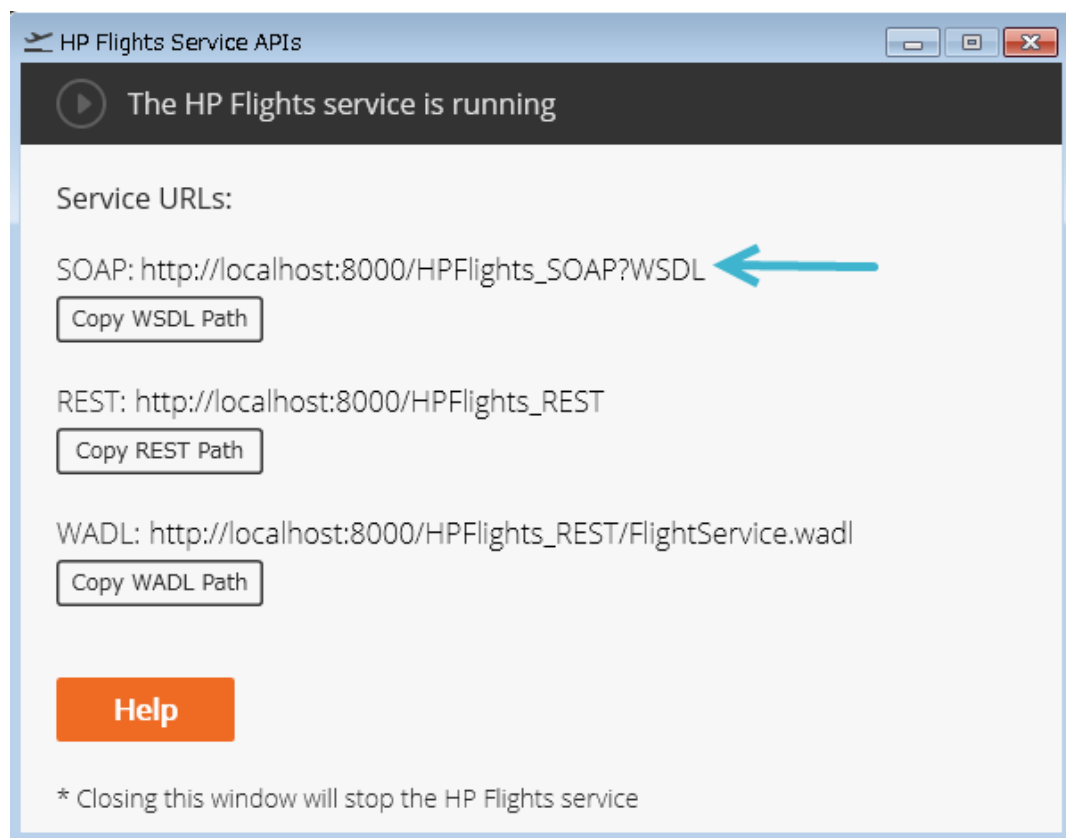
## 2. WSDL ファイルをインポートします。

- a. UFT のツールバーで、[WSDL のインポート] ボタンを押し、[WSDL を URL または UDDI からインポート] を選択します。[WSDL を URL または UDDI からインポート] ダイアログ・ボックスが開きます。

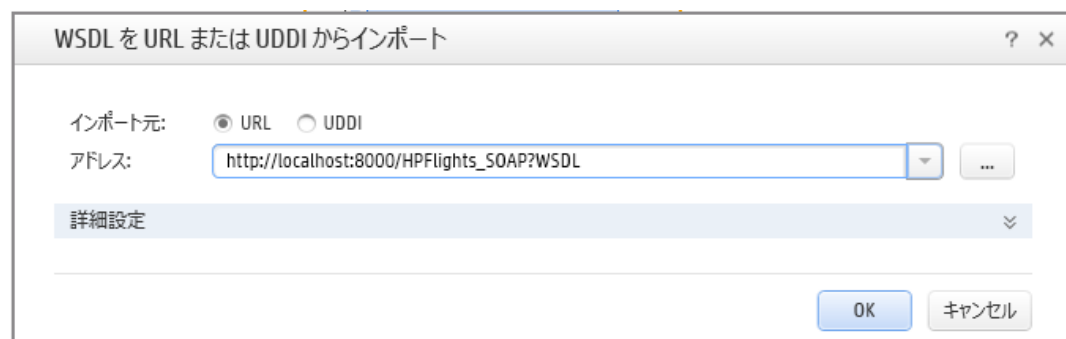
**注:** WSDL ファイルをローカルまたは ALM プロジェクトに保存している場合は、ファイルを UFT に直接インポートすることもできます。

- b. [WSDL を URL または UDDI からインポート] ダイアログ・ボックスで、[URL] オプションを選択します。

- c. HP Flights Service API アプリケーション・ウィンドウで、SOAP ベースのサービスの URL を見つけます。

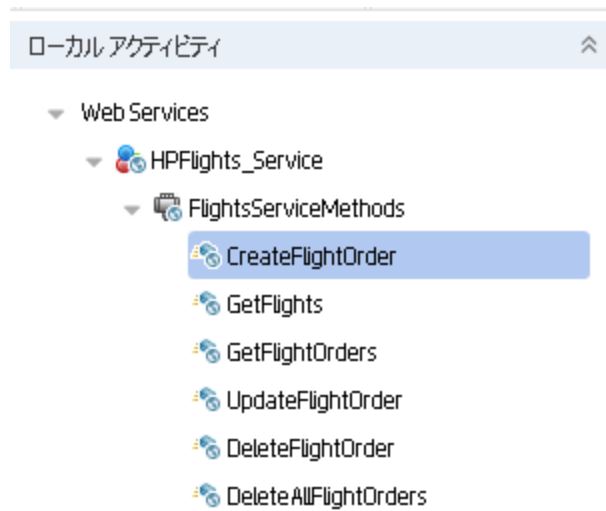


- d. HP Flights Service API アプリケーション・ウィンドウで、**[Copy WSDL Path]** ボタンをクリックします。これで、WSDL ファイルの URL を [WSDL を URL または UDDI からインポート] ダイアログ・ボックスにコピーできます。
- e. UFT の [WSDL を URL または UDDI からインポート] ダイアログ・ボックスの [アドレス] フィールドに、アプリケーション・ウィンドウからコピーした URL を貼り付けます。



- f. **[OK]** をクリックしてサービスを UFT にインポートします。

サービスを UFT にインポートすると，[ツールボックス] 表示枠の [ローカル アクティビティ] セクションの下にこのサービスのメソッドが表示されます。



これでサービスとメソッドがテストにインポートされ，Web サービスのテストを作成する準備ができました。続いて「[練習 6c：Web サービスのテストの作成とパラメータ化](#)」(192ページ)で，Web サービスのテストの構築とパラメータ化を行います。

## 練習 6c：Web サービスのテストの作成とパラメータ化

「[練習 6b：Web サービスのインポート](#)」(189ページ)では，Web サービスの詳細を含む WSDL ファイルをインポートしました。UFT にサービスをインポートすると，[ツールボックス] 表示枠にメソッドが表示されます。[ツールボックス] 表示枠で，これらのメソッドを使用してテストを作成できます。

このレッスンでは，Web サービスのテストを作成してパラメータ化を行い，UFT API テスト・インタフェースを使用して Web サービスのテストを作成する方法を学習します。



「[レッスン 3 : テストへのステップの追加](#)」(46ページ)で作成した Book Flights GUI テストでは、アプリケーション・ウィンドウの順序は次のようになっていました。



同じアプリケーションの API テストを作成する際には、ステップがアプリケーションのフローにできるだけ一致するようにする必要があります。WSDL ファイルからインポートしたメソッドのリストには、次の内容が含まれます。

- **CreateFlightOrder**
- **GetFlights**
- **GetFlightOrders**
- **UpdateFlightOrder**
- **DeleteFlightOrder**
- **DeleteAllFlightOrders**

ユーザ・インタフェースのフローと一致させるには、フライトを見つけた後に顧客の入力に基づいてフライトの申し込みを作成する API テスト・ステップを作成する必要があります。

この練習では、**GetFlights** と **CreateFlightOrder** の 2 つのステップを作成します。


### 1. テスト・ステップを作成します。

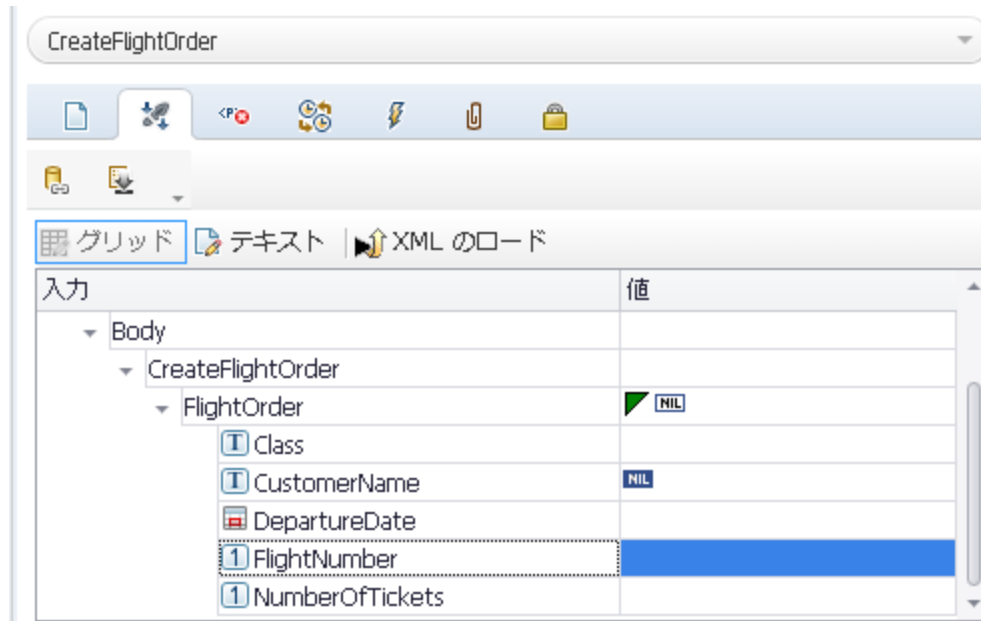
- 「[ツールボックス] 表示枠の [ローカル アクティビティ] セクションで、[Web Services] ノード、[HPFlights\_Services] ノード、[FlightsServiceMethods] ノードの順に展開します。[ツールボックス] 表示枠に、利用可能なすべてのメソッドのリストが表示されます (全部で 6 つ)。
- [FlightsServiceMethods] のリストから、**GetFlights** メソッドをキャンバスにドラッグします。


GetFlights という名前の新しいステップのブロックがキャンバスに追加されます。[プロパティ] 表示枠に [入力/チェックポイント] タブが開きます。

- c. [ツールボックス] 表示枠で、再度 [FlightsServiceMethods] ノードから **CreateFlightOrder** メソッドをキャンバスにドラッグします。

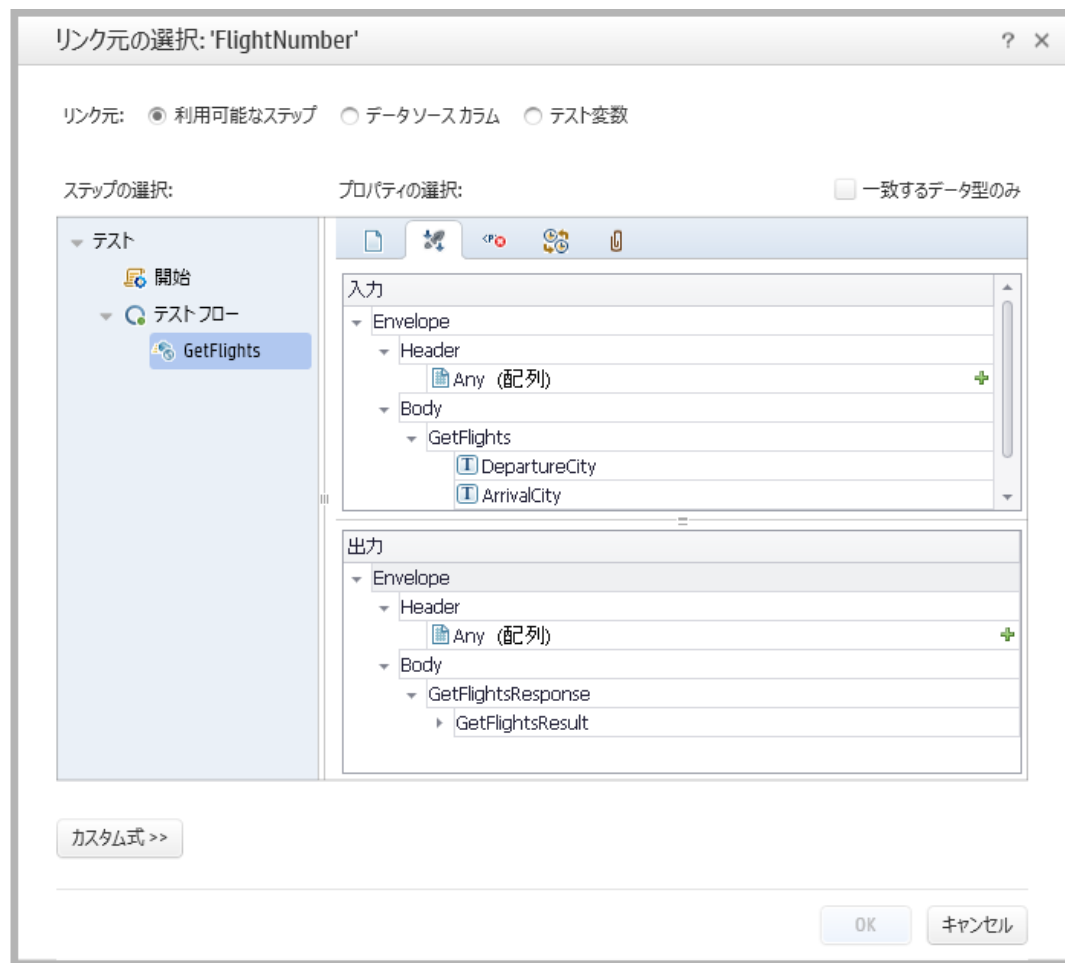
## 2. **[CreateFlightOrder] ステップの [FlightNumber] プロパティを、[GetFlights] ステップの出力にリンクします。**



- a. キャンバスで、[CreateFlightOrder] ステップを選択します。[プロパティ] 表示枠に [入力/チェックポイント] タブ  が開きます。
- b. [入力/チェックポイント] タブの [入力] セクションで、[FlightNumber] 行を選択します。






- c. [FlightNumber] 行の [値] カラムで、[データ ソースにリンク] ボタン  をクリックします。[リンク元の選択] ダイアログ・ボックスが開きます。
- d. [リンク元の選択] ダイアログ・ボックスで、[利用可能なステップ] オプションを選択します。[ステップの選択:] 表示枠 (左側) が更新され、利用可能なステップのリストが表示されます。

- e. [ステップの選択:] 表示枠で, [GetFlights] ステップを選択します。[プロパティの選択:] 表示枠 (右側) が更新され, ステップのプロパティが表示されます。




- f. [プロパティの選択:] 表示枠の [入力/チェックポイント] タブ  の [出力] セクションで, [GetFlightsResult] ノードを展開します。
- g. [GetFlightsResult] ノードの下にある [Flight (配列)] 行で, [追加] ボタン  をクリックします。新しい出力の配列が出力プロパティに追加されます。
- h. [Flight (配列)] 行の下に [Flight[1]] 配列を展開します。[GetFlights] ステップのすべての出力プロパティのリストが表示されます。
- i. 出力プロパティのリストで, [FlightNumber] プロパティを選択し, [OK] をクリックします。ターゲットのステップをループで囲むかどうかを確認するメッセージが表示されたら, [いいえ] を選択します。



【FlightNumber】プロパティの【値】カラムが更新されてリンクが反映されます。

入力	値
Body	
CreateFlightOrder	
FlightOrder	 NIL
Class	Business
CustomerName	 John Smith
DepartureDate	2015/09/17 12:00:00.000
FlightNumber	 {Step.OutputProperties.StServiceCallActivity...
NumberOfTickets	2

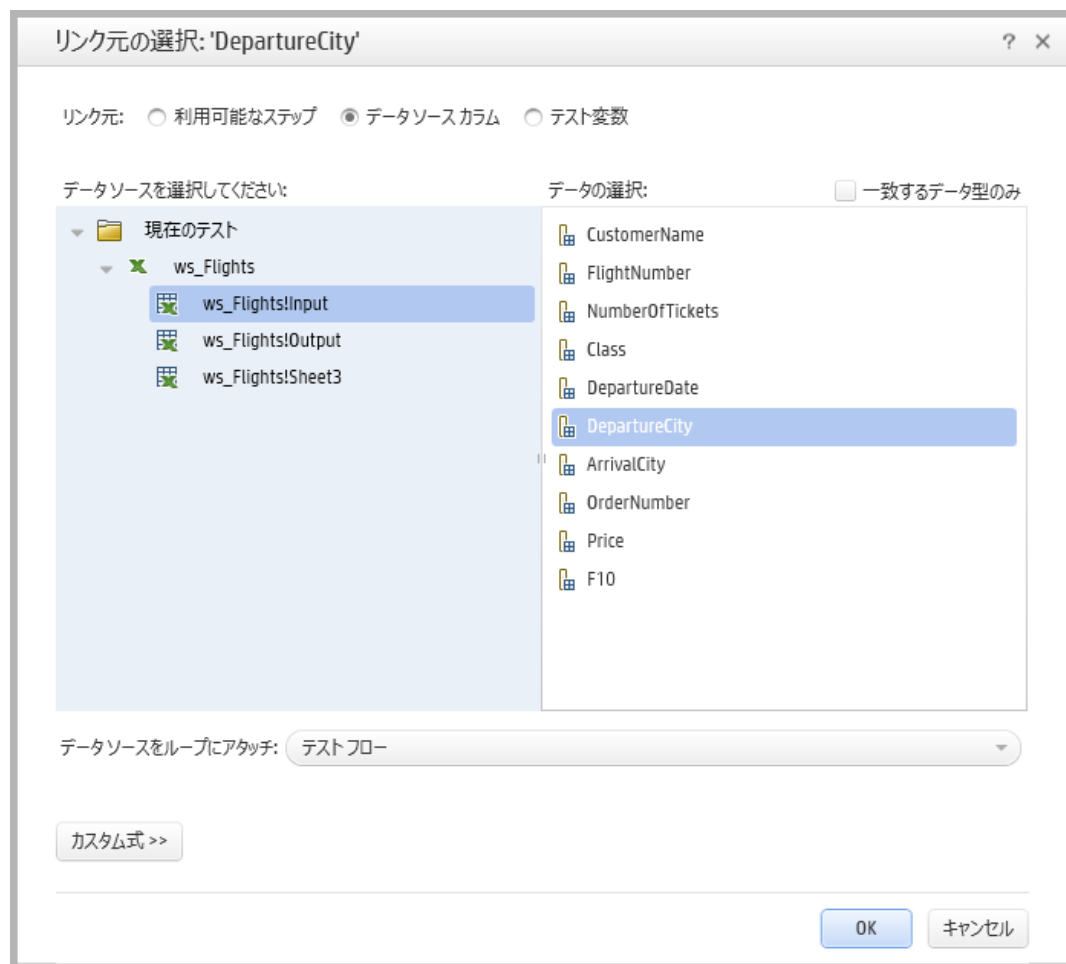
### 3. テストで使用するデータ・ソースを追加します。

- 必要に応じて、【表示】 > 【データ】を選択し、【データ】表示枠を表示します。
- 【データ】表示枠で、【新規データ ソース】ボタン  をクリックして【Excel】を選択します。【新規 Excel データ ソース】ダイアログ・ボックスが開きます。
- 【新規データ ソース】ダイアログ・ボックスの【Excel ファイルのパス】フィールドで、【参照】ボタンをクリックします。
- 【開く】ダイアログ・ボックスで、<UFT インストール・ディレクトリ>\samples\Flights Application に保存されたアプリケーションの Excel ファイルに移動し、【OK】をクリックします。
- 【新規 Excel データ ソース】ダイアログ・ボックスで、ファイルに「WS\_Flights」という名前を付けます。
- 【元の場所にある Excel ファイルへのリンク】オプションを選択します。
- 【OK】をクリックしてデータ・ソース情報を保存し、Excel データをテストに追加します。

### 4. テスト・ステップの入力プロパティをデータ・ソースにリンクします。

- キャンバスで、【GetFlights】ステップを選択します。【プロパティ】表示枠に【入力/チェックポイント】タブ  が開きます。
- 【入力/チェックポイント】タブの【入力】セクションで、【GetFlights】ノードを展開します。
- 【GetFlights】ノードで、【DepartureCity】行を選択します。
- 【DepartureCity】行の【値】カラムで、【データ ソースにリンク】ボタン  をクリックします。【リンク元の選択】ダイアログ・ボックスが開きます。
- 【リンク元の選択】ダイアログ・ボックスで、【データ ソース カラム】オプションを選択します。データ・ソースのリスト（この場合は Excel ファイル）が【データ ソースを選択してください】表示枠（左側）に表示されます。

- f. **【データ ソースを選択してください】** 表示枠で、**【WS\_Flights!Input】** ノードを選択します。利用可能なすべてのデータ・パラメータ（カラム）のリストが、**【データの選択】** 表示枠（右側）に表示されます。
- g. **【データの選択】** 表示枠で、**【DepartureCity】** カラムを選択します。



- h. **【OK】** をクリックして、データ・テーブルのこのカラムに対してプロパティをリンクします。
- 【値】カラムが更新され、データ・ソースへのリンクを示すステートメントが表示されます。
- i. ステップの他のプロパティに対して、上記の手順を繰り返します。
- **ArrivalCity**
  - **FlightDate**


j. CreateFlightOrder ステップの次のプロパティに対して、上記の手順を繰り返します。

- **Class**
- **CustomerName**

**注:** このプロパティの【値】カラムには、NIL という文字を含む青いボックスが存在します。プロパティをリンクする前に、このボックスをクリックして NIL を削除する必要があります（ボックスが白になります）。

- **DepartureDate**
- **NumberOfTickets**

## 5. データ・ソースのナビゲーション設定を指定します。

- a. キャンバスで、（テスト・フロー内のステップではなく）【テスト フロー】を選択します。
- b. 【プロパティ】表示枠で【データ ソース】タブを選択します。
- c. 【データ ソース】タブの関連するデータ・ソースのリストで、テーブルの【WS\_FlightsInput】エントリを選択し、【編集】をクリックします。【データナビゲーション】ダイアログ・ボックスが開きます。
- d. 【データナビゲーション】ダイアログ・ボックスで、データ・ナビゲーションの詳細を指定します。

開始位置 :	最初の行
移動単位 :	3 行進む
終了位置 :	最後の行
最終行への到達時 :	折り返す

- e. 【OK】をクリックし、データ・ナビゲーション・プロパティを割り当ててダイアログ・ボックスを閉じます。

以上でフライト予約アプリケーションの Web サービスのテストの作成が完了しました。テストを実行し、UFT がどのようにテストを実行して Web サービスの実行結果をレポートするかを確認できます。続いて「[練習 6d : Web サービスのテストの実行](#)」(198ページ)に進みます。

## 練習 6d : Web サービスのテストの実行

「[練習 6c : Web サービスのテストの作成とパラメータ化](#)」(192ページ)では、インポートしたメソッドから Web サービスのテストを作成し、ステップのパラメータ化を行いました。このレッスンでは、テストを実行し、Web サービス・アプリケーションのテスト中に UFT がレポートする実行結果を確認します。

### 1. UFT を起動し、Book Flights ソリューションを開きます。


- a. UFT を開きます (「すべてのテスト用のソリューションの作成」(25ページ)を参照)。  
WPF Add-in がロードされていることを確認します。
- b. 開始ページの [最近のソリューション] 領域で、[Flight Reservation Application] をクリックします。  
  
Flight Reservation Application ソリューションが開きます。これには、「レッスン 1 : GUI テストとアクションの作成」(29ページ)で作成した **Book Flights** テストが含まれます。

### 2. Flights API アプリケーションを起動します。


必要に応じて、「フライト予約アプリケーションの確認」(23ページ)の説明に従って、HP Flights Service API アプリケーションを起動します。

**注:** テストを編集および実行する際に UFT からアプリケーションにアクセスできるように、チュートリアルを使用する際には、このアプリケーションを開いたままにしておいてください。

### 3. テストの反復回数を設定します。

- a. キャンバスで、(テスト・フロー内のステップではなく) [テスト フロー] を選択します。[プロパティ] 表示枠に [入力] タブ  が開きます。
- b. [入力] タブで、['For' ループ] オプションを選択します。
- c. [反復回数] フィールドに「4」と入力します。

### 4. テストを実行します。

- a. ツールバーで [実行] ボタン  をクリックします。[実行] ダイアログ・ボックスが開きます。
- b. [実行] ダイアログ・ボックスの [結果保管場所] タブで、[一時実行結果フォルダ] が選択されていることを確認します。
- c. [実行] をクリックして、テストをコンパイルし実行します。  
テストの実行が完了すると、実行結果が開きます。

### 5. 実行結果を表示します。

- a. テスト・フローで、反復に対応するいずれかのノードの下に [GetFlights] ノードを選択します。ステップ・サマリに詳細が表示されます。

- b. キャプチャ・データで, **[Web Service Call HTTP Snapshot]** 領域が表示されるまで下にスクロールします。

Web Service Call HTTP Snapshot

Request	Response
HTTP Header	HTTP Header
SOAPAction: HP.SOAQ.SampleApp/IFlightsSoapService/GetFlights Content-Type: text/xml; charset=utf-8 Host: localhost:8000 Content-Length: 210 Expect: 100-continue Connection: Close	Connection: close Content-Length: 14696 Content-Type: text/xml; charset=utf-8 Date: Mon, 28 Jul 2014 07:21:49 GMT Server: Microsoft-HTTPAPI/2.0
SOAP	SOAP
<pre>&lt;Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/"&gt;   &lt;Body&gt;     &lt;GetFlights xmlns="HP.SOAQ.SampleApp"&gt;       &lt;DepartureCity&gt;Los Angeles&lt;/DepartureCity&gt;       &lt;ArrivalCity&gt;Sydney&lt;/ArrivalCity&gt;     &lt;/GetFlights&gt;   &lt;/Body&gt; &lt;/Envelope&gt;</pre>	<pre>&lt;DepartureCity&gt;Los Angeles&lt;/DepartureCity&gt; &lt;DepartureTime&gt;09:04 AM&lt;/DepartureTime&gt; &lt;FlightNumber&gt;13930&lt;/FlightNumber&gt; &lt;Price&gt;117&lt;/Price&gt; &lt;/Flight&gt; &lt;Flight&gt;   &lt;Airline&gt;LH&lt;/Airline&gt;   &lt;ArrivalCity&gt;Sydney&lt;/ArrivalCity&gt;   &lt;ArrivalTime&gt;10:02 AM&lt;/ArrivalTime&gt;</pre>

この領域で, UFT は Web サービス呼び出しの HTTP 要求および応答の情報を提供します。

**[Request]** の **[SOAP]** ウィンドウでは, GetFlights ステップに対して送信された入力プロパティを参照できます。

```
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <GetFlights xmlns="HP.SOAQ.SampleApp">
      <DepartureCity>Los Angeles</DepartureCity>
      <ArrivalCity>Sydney</ArrivalCity>
    </GetFlights>
  </Body>
</Envelope>
```



同様に，[応答] 領域では，ステップの出力プロパティを参照できます。

SOAP

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetFlightsResponse xmlns="HP.SOAQ.SampleApp">
      <GetFlightsResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <Flight>
          <Airline>AF</Airline>
          <ArrivalCity>Sydney</ArrivalCity>
          <ArrivalTime>10:02 AM</ArrivalTime>
          <DepartureCity>Los Angeles</DepartureCity>
          <DepartureTime>09:04 AM</DepartureTime>
          <FlightNumber>13930</FlightNumber>
          <Price>117</Price>
        </Flight>
      </GetFlightsResult>
    </GetFlightsResponse>
  </s:Body>
</s:Envelope>
```

- c. [GetFlights] ノードの下に [チェックポイント] ノードを選択します。実行結果に，チェックポイントのサマリが表示されます。
- d. チェックポイントのステップの詳細には，結果（チェックポイントが成功したか失敗したか，実際値と期待値，チェックポイントのタイプ）が表示されます。

Name	Result	Property	Actual Result	Evaluation Style	Expected Values	Details
"チェックポイント 1"	✓	...	...	構造的検証	...	

- 6. **テストを保存します。**  
[ファイル] > [保存] を選択します。

# レッスン 7：REST サービスの API テストの作成と実行

UFT での Web サービスのテストに加え、API テストを使用してアプリケーションの REST ベース・サービスまたは REST ベース・サービス層をテストすることもできます。UFT でサービスのプロトタイプ・モデルを作成し、作成したメソッドを使用してテストを構築できます。

このレッスンでは、REST サービスのプロトタイプ・モデルを作成し、これらのメソッド・モデルを使用してテストを作成する基本的な手順について説明します。

このレッスンには、次の手順が含まれています。

- [練習 7a：REST サービスのテストの作成](#) ..... 203
- [練習 7b：REST サービスの構造の作成](#) ..... 203
- [練習 7c：REST サービス・メソッドを使用したテストの作成](#) ..... 208
- [練習 7d：REST サービスのテストの実行](#) ..... 211
- [練習 7e：REST サービスの競合の解決](#) ..... 213

## 練習 7a：REST サービスのテストの作成


「[レッスン 6：Web サービスの API テストの作成と実行](#)」(188ページ)では、フライト予約アプリケーションの API 側の Web サービス・テストを作成しました。この練習では、フライト予約アプリケーションの API 側の REST サービス・コンポーネントのテストを追加します。

### 1. UFT を起動し、フライト予約アプリケーション・ソリューションを開きます。

- UFT を開きます（「[すべてのテスト用のソリューションの作成](#)」(25ページ)を参照）。
- 開始ページの「最近のソリューション」領域で、「**Flight Reservation Application**」をクリックします。

Flight Reservation Application ソリューションが開きます。

### 2. 新規 API テストをソリューションに追加します。

- ツールバーで、「追加」ドロップダウン矢印  を選択し、「**新規テストの追加**」を選択します。「新規テストの追加」ダイアログ・ボックスが開きます。
- 「新規テストの追加」ダイアログ・ボックスで、「**API テスト**」を選択します。
- 「名前」フィールドで、テストに「Book Flights REST Service」という名前をつけます。
- 「場所」フィールドで「参照」ボタンをクリックし、**C:\%HOMEPATH%\Unified Functional Testing** フォルダに移動します。
- 「追加」をクリックしてテストを作成し、ソリューションに追加します。

Flight Reservation Application ソリューションに Book Flights REST Service テストが個別ノードとして追加され、ドキュメント表示枠に別のタブとして表示されます。

**注：**ソリューションが自動的に保存されます。

テストの作成が完了し、REST サービス・モデルとメソッドを使用する準備ができました。続いて「[練習 7b：REST サービスの構造の作成](#)」(203ページ)で、テスト内で REST サービス・モデルを作成する方法について学習します。

## 練習 7b：REST サービスの構造の作成

テストで REST サービス・アクティビティを使用するには、必要なメソッドとそれらのプロパティから成るモデルを UFT で作成する必要があります。UFT はサービスとメソッドに関する情報を取得し、実際のサービスのパフォーマンスをテストするためのテスト・ステップとして使用します。

### 1. UFT を起動し、Book Flights ソリューションを開きます。

- UFT を開きます（「[すべてのテスト用のソリューションの作成](#)」(25ページ)を参照）。

- b. 開始ページの **【最近のソリューション】** 領域で、**【Flight Reservation Application】** をクリックします。

Flight Reservation Application ソリューションが開きます。これには、**「練習 7a：REST サービスのテストの作成」** (203 ページ) で作成した **【Book Flights REST Service】** テストが含まれています。

## 2. Flights API アプリケーションを起動します。

HP Flights Service API アプリケーションが実行中であることを、**「フライト予約アプリケーションの確認」** (23 ページ) の説明に従って確認します。


## 3. REST サービス・メソッドのプロパティのヘルプ・ドキュメントを開きます。

HP Flights Service API ウィンドウで、**【ヘルプ】** ボタンをクリックします。ブラウザ・ウィンドウが開いて、メソッド情報が表示されます。


## 4. REST サービス・モデルを作成します。

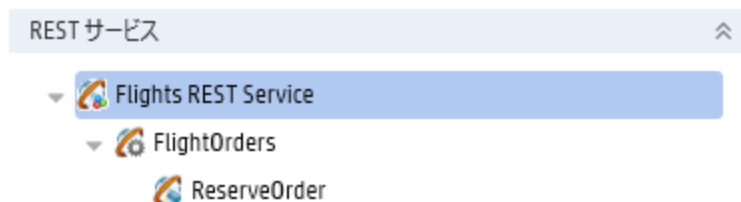
- a. ツールバーで、**【REST サービスの追加】** ボタンをクリックします。[REST サービスの追加] ダイアログ・ボックスが開きます。
- b. [REST サービスの追加] ダイアログ・ボックスで、**【新規サービス】** の名前を「Flights REST Service」に変更します。

## 5. リソースを REST サービス・モデルに追加します。

- a. [REST サービスの追加] ダイアログ・ボックスで、ツールバーの **【リソースの追加】** ボタン  をクリックします。サブノードが [Flights REST Service] ノードに追加されます。
- b. リソースの名前を **FlightOrders** に変更します。





## 6. メソッドを REST サービス・モデルに追加します。

- a. [REST サービスの追加] ダイアログ・ボックスで、ツールバーの **【メソッドの追加】** ボタン  をクリックします。サブノードが FlightOrders リソースに追加されます。
- b. リソースの名前を ReserveOrder に変更します。  
サービス、リソース、メソッドが追加されて、次のような 3 階層の階層構造が構成されます。





## 7. REST サービス・メソッド・モデル URL を設定します。



REST サービス・モデルのメソッドでアプリケーションを正確にテストするには、アプリケーションのサービスの URL を指定する必要があります。この URL は、**サービス**、**リソース**、**メソッド**のレベルで指定します。

- a. [REST サービスの追加] ダイアログ・ボックスで、[**Flights REST Service**] ノードを選択します。右側の表示枠に [一般] タブ  が開きます。
- b. [一般] タブで、[URL] プロパティの [値] カラムに、「http://localhost:8000」と入力します。
- c. 左の表示枠で、[**Flight Orders**] 行を選択します。右の表示枠に、[一般] 表示枠  が再度開きます。Flights REST Service を選択していたときに [一般] 表示枠で追加した URL が表示されます。
- d. [一般] タブで、[**相対 URL**] プロパティの [値] カラムに、「HPFlights\_REST」と入力します。  
この部分の URL を追加すると、UFT によって [相対 URL] の値が [URL] の値に追加され、連結された URL : http://localhost:8000/HPFlights\_REST が作成されます。
- e. 左の表示枠で、[**ReserveOrder**] ノードを選択します。右の表示枠に [一般] 表示枠  が開きます。
- f. 右の表示枠で、[**HTTP 入力/チェックポイント**] タブ  を選択します。
- g. [**相対 URL**] プロパティの [値] カラムに、「/FlightOrders」と入力します。  
この部分の URL は、[**Flights REST Service**] および [**Flight Orders**] のレベルから渡された URL と連結されます。

#### 8. ReserveOrders メソッド・モデルの追加の HTTP プロパティを設定します。

- a. [REST サービスの追加] ダイアログ・ボックスで、[**ReserveOrder**] ノードを選択します。右の表示枠に [一般] 表示枠  が開きます。
- b. 右の表示枠で、[**HTTP 入力/チェックポイント**] タブ  を選択します。
- c. [**HTTP メソッド**] プロパティの [値] カラムで、HTTP タイプを POST に設定します。

#### 9. ReserveOrder メソッド・モデルの応答の情報を追加します。

- a. [REST サービスの追加] ダイアログ・ボックスで、[**ReserveOrder**] ノードを選択します。右側の表示枠に [一般] タブ  が開きます。
- b. 右の表示枠で、[**HTTP**] タブ  を選択します。


- c. **【要求本文】** セクションの **【要求】** ドロップダウン・リストで、**【XML】** を選択します。テキスト・エディタが開き、XML の入力が可能になります。

- d. 要求本文タイプで、**【XML】** が選択されていることを確認します。  
 e. テキスト・エディタ領域に、次の XML を入力します。

```
<FlightOrderDetails xmlns="HP.SOAQ.SampleApp">
  <Class>Business</Class>
  <CustomerName>John Parker</CustomerName>
  <DepartureDate>27.05.15</DepartureDate>
  <FlightNumber>1042</FlightNumber>
  <NumberOfTickets>1</NumberOfTickets>
</FlightOrderDetails>
```

**注:** この XML をファイルに保存し、**【XML のロード】** ボタンをクリックして XML を入力することもできます。

【グリッド】 ボタンをクリックすると、XML で入力したプロパティをグリッド・フォームで表示して確認することもできます。

- f. 右の表示枠で、**【HTTP 入力/チェックポイント】** タブ  を再度選択します。  
 g. **【HTTP 入力/チェックポイント】** タブの **【入力】** セクションで、**【RequestHeaders】** ノード、**【RequestHeaders[1]】** ノードの順に展開します。

次の応答の設定を確認します。

- **名前**：Content - Type
- **値**：text/xml

入力	値
相対 URL	/FlightOrders
HTTP メソッド	POST
HTTP バージョン	1.1
RequestHeaders (配列) +	
RequestHeaders[1] ✖	
名前	Content-Type
値	text/xml

#### 10. ReserveOrder メソッド・モデルの出力プロパティを作成します。

- 【REST サービスの追加】ダイアログ・ボックスで，【ReserveOrder】ノードを選択します。右側の表示枠に【一般】タブが開きます。
- 右の表示枠で，【カスタム入力/チェックポイント】タブを選択します。
- 【カスタム入力/チェックポイント】タブで，【追加】ボタンをクリックし，【出力プロパティの追加】を選択します。【出力プロパティの追加】ダイアログ・ボックスが開きます。
- 【出力プロパティの追加】ダイアログ・ボックスで，【名前】フィールドに「Total\_Price」と入力します。
- 【タイプ】ドロップダウン・メニューで **Int** を選択し，【OK】をクリックして出力プロパティを追加します。【カスタム入力/チェックポイント】タブの【出力】セクションに，新しい出力プロパティが追加されます。
- 同じ手順を使用し，タイプ **Int** の **Order\_Number** という名前の別の出力プロパティを追加します。

次のように，ReserveOrder メソッドの【カスタム入力/チェックポイント】タブに，作成したすべての出力プロパティが表示されます。

チェックポイント	検証	値
プロパティ	<input type="checkbox"/>	
Total_Price	<input type="checkbox"/>	= 0
Order_Number	<input type="checkbox"/>	= 0

#### 11. ReserveOrder メソッド・モデルをテストします。

- 【REST サービスの追加】ダイアログ・ボックスで，【ReserveOrder】メソッド・ノードを選択します。

- b. ツールバーで、**[メソッドの実行]** ボタン  をクリックします。

メソッドが実行され、**[REST サービスの追加]** ダイアログ・ボックスの下部の表示枠に結果が表示されます。

通信時間	0:0:37
応答本文 <a href="#">レンダリングされた HTML の表示</a>	<pre> 1 &lt;!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd"&gt; 2 &lt;HTML&gt;&lt;HEAD&gt;&lt;TITLE&gt;Length Required&lt;/TITLE&gt; 3 &lt;META HTTP-EQUIV="Content-Type" Content="text/html; charset=us-ascii"&gt;&lt;/HEAD&gt; 4 &lt;BODY&gt;&lt;h2&gt;Length Required&lt;/h2&gt; 5 &lt;hr&gt;&lt;p&gt;HTTP Error 411. The request must be chunked or have a content length.&lt;/p&gt; 6 &lt;/BODY&gt;&lt;/HTML&gt; 7 </pre>
XML 応答	
応答タイプ	Text

## 12. サービス・モデルとメソッドをツールボックスに追加します。

**[REST サービスの追加]** ダイアログ・ボックスで、**[OK]** ボタンをクリックします。

REST サービス・モデル、リソース、メソッドが、**[ツールボックス]** 表示枠の**[ローカル アクティビティ]** ノードの下に追加されます。



**[ツールボックス]** 表示枠から、メソッドをキャンバスにドラッグし、ステップのプロパティを編集します。

REST サービスのプロトタイプ・モデルが作成され、メソッドを使用してテストを作成する準備ができました。続いて「[練習 7c：REST サービス・メソッドを使用したテストの作成](#)」(208ページ)では、テスト内でメソッドを使用します。

## 練習 7c：REST サービス・メソッドを使用したテストの作成

「[練習 7b：REST サービスの構造の作成](#)」(203ページ)では、テスト・ステップの作成で使用する REST サービスのプロトタイプ・モデル（メソッドとプロパティを含む）を作成しました。

この練習では、テスト・フローで REST サービス・モデルのメソッドを使用してテストを作成します。

### 1. UFT を起動し、Book Flights ソリューションを開きます。


- a. UFT を開きます（「[すべてのテスト用のソリューションの作成](#)」(25ページ)を参照）。




- b. 開始ページの [最近のソリューション] 領域で、[Flight Reservation Application] をクリックします。

Flight Reservation Application ソリューションが開きます。これには、「[練習 7a：REST サービスのテストの作成](#)」(203ページ)で作成した [Book Flights REST Service] テストが含まれています。

## 2. テストで使用するデータ・ソースをインポートします。



- a. 必要に応じて、[表示] > [データ] を選択し、[データ] 表示枠を表示します。
- b. [データ] 表示枠で、[新規データ ソース] ボタン  をクリックして [Excel] を選択します。[新規 Excel データ ソース] ダイアログ・ボックスが開きます。
- c. [新規データ ソース] ダイアログ・ボックスの [Excel ファイルのパス] フィールドで、[参照] ボタンをクリックします。
- d. [開く] ダイアログ・ボックスで、<UFT インストール・ディレクトリ>\samples\Flights Application に保存されたアプリケーションの Excel ファイルに移動し、[OK] をクリックします。
- e. [新規 Excel データ ソース] ダイアログ・ボックスで、ファイルに「Flights\_REST」という名前を付けます。
- f. [元の場所にある Excel ファイルへのリンク] オプションを選択します。
- g. [OK] をクリックしてデータ・ソース情報を保存し、Excel データをテストに追加します。

## 3. ReserveOrder メソッドをテストするステップを作成します。

- a. ツールバーで、[ツールボックス] ボタン  をクリックします。[ツールボックス] 表示枠が開きます。
- b. [ツールボックス] 表示枠の [ローカル アクティビティ] セクションで、[Flights REST Service] ノードの下のノードを展開します。
- c. [ローカル アクティビティ] ノードで、[ReserveOrder] ステップをキャンバスにドラッグします。

メソッド名 (ReserveOrder) を使用して、新しいブロックがテスト・フローに追加されます。

## 4. メソッドの HTTP 要求プロパティをデータ・ソースにリンクします。


- a. キャンバスで、[ReserveOrder] ステップを選択します。
- b. [プロパティ] 表示枠で、[HTTP] タブ  を選択します。
- c. [HTTP] タブの [要求] セクションで、[グリッド] ボタンをクリックします。
- d. [Class] プロパティの [値] カラムで、[データ ソースにリンク] ボタン  をクリックします。[リンク元の選択] ダイアログ・ボックスが開きます。

- e. [リンク元の選択] ダイアログ・ボックスで、[データ ソース カラム] オプションを選択します。[データ ソースを選択してください:] 表示枠（左の表示枠）に、利用可能なすべてのデータ・ソースのリストが表示されます。
- f. [データ ソースを選択してください] 表示枠で、[Flights\_REST!Input] ノードを選択します。[データの選択:] 表示枠（右の表示枠）に、すべてのデータ・カラム/パラメータのリストが表示されます。
- g. [データの選択:] 表示枠で、[Class] ノードを選択し、[OK] をクリックします。  
[Class] プロパティの値が更新され、データ・ソースへのリンクが反映されます。

スキーマ	値
▼ FlightOrderDetails	
Class	{DataSource.Flights_REST!Input.Class} ✕
CustomerName	John Parker
DepartureDate	2115/05/27
FlightNumber	1042
NumberOfTickets	1

- h. 他の HTTP 要求プロパティについても、同様の手順を繰り返します。
  - **CustomerName**
  - **DepartureDate**
  - **FlightNumber**
  - **NumberOfTickets**

## 5. ReserveOrder ステップのチェックポイントを設定します。

- a. キャンバスで、[ReserveOrder] ステップを再度選択します。
- b. [プロパティ] 表示枠で、[HTTP] タブ  を選択します。
- c. テキスト・エディタで、次の XML を貼り付けます。

```
<?xml version="1.0"?>
<CreatedOrderParams xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="HP.SOAQ.SampleApp">
  <OrderNumber>93</OrderNumber>
  <TotalPrice>374.400024</TotalPrice>
</CreatedOrderParams>
```

- d. テキスト・エディタでファイルに **response.xml** という名前を付けてそれぞれが選択したディレクトリに保存します。
- e. UFT の [プロパティ] 表示枠で、[HTTP] タブの [応答] セクションのドロップダウン・リストから [XML] を選択します。
- f. [応答本文] セクションで、[XML のロード] ボタンをクリックします。

- g. [開く] ダイアログ・ボックスで、前のステップで保存した **response.xml** ファイルに移動し、[開く] をクリックします。

[HTTP] タブの [応答] 本文セクションに **response.xml** ファイルから XML スキーマがロードされます。

スキーマ	検証	値
CreatedOrderParams	<input checked="" type="checkbox"/>	=
OrderNumber	<input checked="" type="checkbox"/>	> 93
TotalPrice	<input checked="" type="checkbox"/>	< 374.400024

- h. [OrderNumber] プロパティの [値] カラムで、ドロップダウン矢印をクリックして > を選択します。
- i. 値に **10** と入力します。
- j. [TotalPrice] プロパティの [値] カラムで、同様の手順を繰り返し、< と値 **500** を入力します。

以上でテスト・ステップの作成および入力プロパティとチェックポイント・プロパティの指定が完了し、テストを実行して実行結果を表示する準備ができました。続いて「[練習 7d：REST サービスのテストの実行](#)」(211ページ)に進みます。

## 練習 7d：REST サービスのテストの実行

「[練習 7c：REST サービス・メソッドを使用したテストの作成](#)」(208ページ)では、REST サービス・モデルのメソッドを使用してテストを作成しました。このレッスンでは、テストを実行して実行結果を表示する方法について学習します。


### 1. UFT を起動し、Book Flights ソリューションを開きます。

- UFT を開きます（「[すべてのテスト用のソリューションの作成](#)」(25ページ)を参照）。
- 開始ページの [最近のソリューション] 領域で、[Flight Reservation Application] をクリックします。



Flight Reservation Application ソリューションが開きます。これには、「[練習 7a：REST サービスのテストの作成](#)」(203ページ)で作成した [Book Flights REST Service] テストが含まれています。

### 2. テストの反復回数を設定します。

テストで使用するデータ・ソースには複数の行が含まれるため、実行する反復回数を指定する必要があります。


- a. キャンバスで、**テスト・フロー**を選択します。[プロパティ] 表示枠に [入力] タブ  が開きます。
- b. [入力] タブで、[**For** ループ] オプションを選択します。
- c. [**反復回数**] フィールドに、8 と入力します。

### 3. データ・ソースのデータ・ナビゲーション・プロパティを設定します。

- a. キャンバスで、テスト・フローを選択します。[プロパティ] 表示枠に [入力/チェックポイント] タブ  が開きます。
- b. [プロパティ] 表示枠で [**データ ソース**] タブ  を選択します。テスト・フローに関連付けられたすべてのデータ・ソースのリストが表示されます。
- c. データ・ソースのリストで、[**Flights\_REST!Input**] データ・ソースを選択し、[編集] をクリックします。[データナビゲーション] ダイアログ・ボックスが開きます。
- d. [データナビゲーション] ダイアログ・ボックスで、次のデータ・ナビゲーション・プロパティを設定します。

開始位置：	最初の行
移動単位：	1行進む
終了位置：	最後の行
最終行への到達時	折り返す

### 4. テストを実行します。

- a. HP Flights Service API アプリケーションが開いていることを確認します。
- b. ツールバーで [**実行**] ボタン  をクリックします。  
UFT でテスト・ステップが実行され、データ・ソースからプロパティ値が提供されます。テスト実行ログが [出力] 表示枠に表示されます。  
テストの実行が終了すると、実行結果が開きます。

### 5. 実行結果を分析します。

- a. [実行結果ツリー] 表示枠（左の表示枠）で、[アクション: Book Flights REST Service] ノードを展開します。
- b. 実行結果ツリーで、[ReserveOrder] ノードを選択します。ステップの詳細が表示されます。

- c. キャプチャ・データで、要求と応答の情報を確認します。[要求本文] および [応答本文] セル内のリンクをクリックすると、XML 要求と応答の情報をブラウザ・ウィンドウで開くことができます。

```
<?xml version="1.0"?>
- <CreatedOrderParams xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns="HP.SOAQ.SampleApp">
  <OrderNumber>96</OrderNumber>
  <TotalPrice>374.400024</TotalPrice>
</CreatedOrderParams>
```

- d. [ReserveOrder] ノードの下に [チェックポイント] ノードを選択します。

キャプチャ・データで、チェックポイントのステータスを確認します。この場合は、実際値が期待値の制限範囲内にあるため、チェックポイントをパスしています。

Name	Result	Property	Actual Result	Evaluation Style	Expected Values	Details
"チェックポイント 1"	✓	"CreatedOrderParams[1]/OrderNumber[1]"	"96"	>	"10"	""
"チェックポイント 2"	✓	"CreatedOrderParams[1]/TotalPrice[1]"	"374.400024"	<	"500"	""

## 練習 7e：REST サービスの競合の解決

「練習 7b：REST サービスの構造の作成」(203ページ)では、メソッド ReserveOrder を使用してプロトタイプの REST サービス・モデルを作成しました。このメソッドには、URL やプロパティ名などの特定のプロパティが含まれていました。テストを作成した後にサービス・モデルのプロパティが変更された場合、テストがモデルと一致しなくなります。そのため、UFT では競合の解決ウィザードを使用して、メソッドのプロパティの変更を解決できます。


この練習では、競合の解決ウィザードを使用してこれらの差異を解決します。




### 1. UFT を起動し、Book Flights ソリューションを開きます。

- UFT を開きます（「すべてのテスト用のソリューションの作成」(25ページ)を参照）。
- 開始ページの [最近のソリューション] 領域で、[Flight Reservation Application] をクリックします。

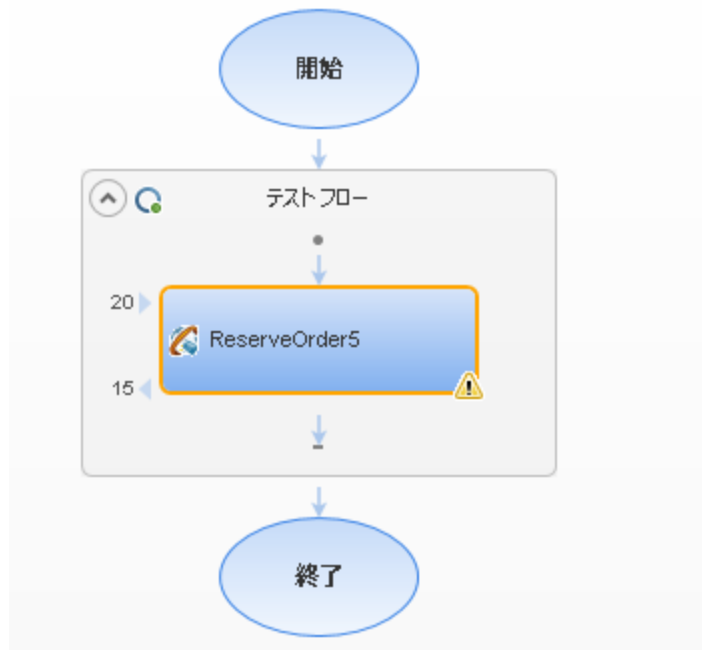
Flight Reservation Application ソリューションが開きます。これには、「練習 7a：REST サービスのテストの作成」(203ページ)で作成した [Book Flights REST Service] テストが含まれています。

### 2. サービス・モデルのプロパティを編集します。

- ツールバーで、[ツールボックス] ボタン  をクリックします。
- [ツールボックス] の [ローカル アクティビティ] セクションで、[REST サービス] ノードの下にノードを展開します。

- c. **[Flights REST Service]** ノードを右クリックし、**[サービスの編集]** を選択します。  
[REST サービスの編集] ダイアログ・ボックスが開きます。
- d. [REST サービスの編集] ダイアログ・ボックスで、**[ReserveOrder]** ノードを選択します。右側の表示枠に **[一般]** タブ  が開きます。
- e. 右の表示枠で、**[カスタム入力/チェックポイント]** タブ  を選択します。
- f. [カスタム入力/チェックポイント] タブの **[チェックポイント]** セクションで、**[Total\_Price]** プロパティを選択し、**[プロパティの編集]** ボタン  をクリックします。[プロパティの編集] ダイアログ・ボックスが開きます。
- g. [プロパティの編集] ダイアログ・ボックスで、プロパティ名を「TotalPrice」に変更し、**[OK]** をクリックします。[チェックポイント] セクションのプロパティ名が変更されます。
- h. 同様の手順を繰り返し、**[Order\_Number]** プロパティを「OrderNumber」に変更します。
- i. [REST サービスの編集] ダイアログ・ボックスで、**[OK]** をクリックしてサービスへの変更を保存します。

REST サービス・モデルへの変更が保存されます。ただし、キャンバスの ReserveOrder ステップにアラートが表示されます。



これはサービス・モデルとテスト内のステップとの間に競合が存在し、解決が必要であることを示しています。

### 3. 競合を解決します。

- a. キャンバスで、アラート・アイコンをクリックします。次のメッセージが表示されます：このステップは解決する必要があります。ステップの解決。

- b. アラートのテキストをクリックします。REST メソッドの解決ウィザードが開きます。



- c. [ステップの選択] 画面（最初の画面）で，[ReserveOrder] ステップのチェックボックスを選択し，[次へ] をクリックします。[競合の解決] 画面が開きます。

**注:** このウィザードを使用する場合，[ステップの選択] 画面に競合が存在するすべてのステップが表示されます。そのため，複数のステップの競合を一度に解決できます。

- d. [競合の解決] 画面の[出力プロパティ] セクション（下部の領域）の[変更後] ボックスで，[Total\_Price] プロパティ（赤で表示）を選択します。

変更後:

スキーマ	検証	値
プロパティ		
1 Total_Price	<input type="checkbox"/>	0
1 Order_Number	<input type="checkbox"/>	0
1 TotalPrice	<input type="checkbox"/>	0

- e. [削除] をクリックします。
- f. 同様の手順を繰り返し，[Order\_Number] プロパティ（赤で表示）を削除します。
- g. [次へ] をクリックして続行します。[完了] 画面が開きます。
- h. [完了] 画面には，サービス・モデル内の競合のステータスが表示されます。この場合，競合は存在しません。

【完了】をクリックしてウィザードを終了します。REST メソッドの解決ウィザードで選択した変更内容に合わせてテストが更新されます。この練習では，【プロパティ】表示枠で【ReserveOrder】ステップの出力プロパティが更新されます。

チェックポイント	検証	値
▼ 属性		
1 TotalPrice	<input type="checkbox"/>	= 0
1 OrderNumber	<input type="checkbox"/>	= 0

4. **テストを保存します。**  
【ファイル】 > 【保存】を選択します。



# レッスン 8 : Web アプリケーション・サービスの API テストの作成と実行 (WADL)

「[レッスン 3 : 標準アクティビティを使用した API テスト・ステップの作成](#)」(155ページ)では、標準 API アクティビティを使用してテストを作成する方法を学習しました。

ただし、標準アクティビティがアプリケーションで実行されるプロセスに適合しない場合もあります。このような場合、カスタム・アクティビティを UFT にインポートして使用する必要があります。他の種類のサービス・アクティビティの 1 つに、Web アプリケーション・サービスがあります。UFT では、サービス記述とそれぞれのメソッドを UFT にインポートし、これらをテストで使用できます。

このレッスンでは、Web アプリケーション・サービス記述を UFT にインポートして使用方法について学習します。

このレッスンには、次の手順が含まれています。

- [練習 8a : Web アプリケーション・サービス用のテストの作成](#) ..... 218
- [練習 8b : Web アプリケーション・サービス・モデルのインポート](#) ..... 219
- [練習 8c : Web アプリケーション・サービスのメソッドの編集](#) ..... 221
- [練習 8d : Web アプリケーション・サービスのメソッドを使用したテストの構築](#) ..... 224
- [練習 8e : Web アプリケーション・サービスのテストの実行](#) ..... 229

## 練習 8a : Web アプリケーション・サービス用のテストの作成

「すべてのテスト用のソリューションの作成」(25ページ)では、フライト予約アプリケーション用のソリューションを作成し、アプリケーションのユーザ・インタフェースの GUI テスト、およびアプリケーションの Web サービスと REST サービスの API テストを追加しました。

この練習では、フライト予約アプリケーションの Web アプリケーション・サービス用のテストを作成します。

### 1. UFT を起動し、フライト予約アプリケーション・ソリューションを開きます。

- UFT を開きます (「すべてのテスト用のソリューションの作成」(25ページ)を参照)。
- 開始ページの「最近のソリューション」領域で、**[Flight Reservation Application]** をクリックします。  
ソリューション・エクスプローラに Flight Reservation Application ソリューションが開きます。

### 2. 新規 API テストをソリューションに追加します。

- ツールバーで、**[追加]** ドロップダウン矢印  **追加** を選択し、**[新規テスト]** を選択します。
- 「新規テストの追加」ダイアログ・ボックスで、**[API テスト]** を選択します。
- [名前]** フィールドで、テストに「Flights WADL」という名前をつけます。
- [場所]** フィールドで **[参照]** ボタンをクリックし、**C:\%HOMEPATH%\Unified Functional Testing** フォルダに移動します。
- [追加]** をクリックしてテストを作成し、ソリューションに追加します。

Flight Reservation Application ソリューションに Book Flights WADL テストが個別ノードとして追加され、ドキュメント表示枠に別のタブとして表示されます。

**注:** ソリューションが自動的に保存されます。

テストの作成が完了し、Web アプリケーション・サービス・モデルとメソッドを使用する準備ができました。続いて「練習 8b : Web アプリケーション・サービス・モデルのインポート」(219ページ)で、サービス記述を UFT にインポートする方法を学習します。

## 練習 8b：Web アプリケーション・サービス・モデルのインポート

Web アプリケーション・サービスをテストする前に、サービス記述（リソースとメソッドの構造を含む）をインポートする必要があります。Web アプリケーション・サービス記述は、WADL（Web Application Description Language）ファイルに保存されます。UFT はこのファイルを読み取り、（REST サービスの階層に似た）サービス、リソース、メソッドの階層を作成します。サービス記述のインポートが済んだら、メソッドを使用してテストを作成できます。

この練習では、フライト予約アプリケーションのサービス WADL ファイルを UFT にインポートします。

### 1. Flights API アプリケーションを起動します。

「[フライト予約アプリケーションの確認](#)」(23ページ)の説明に従って、HP Flights Service API アプリケーションを起動します。

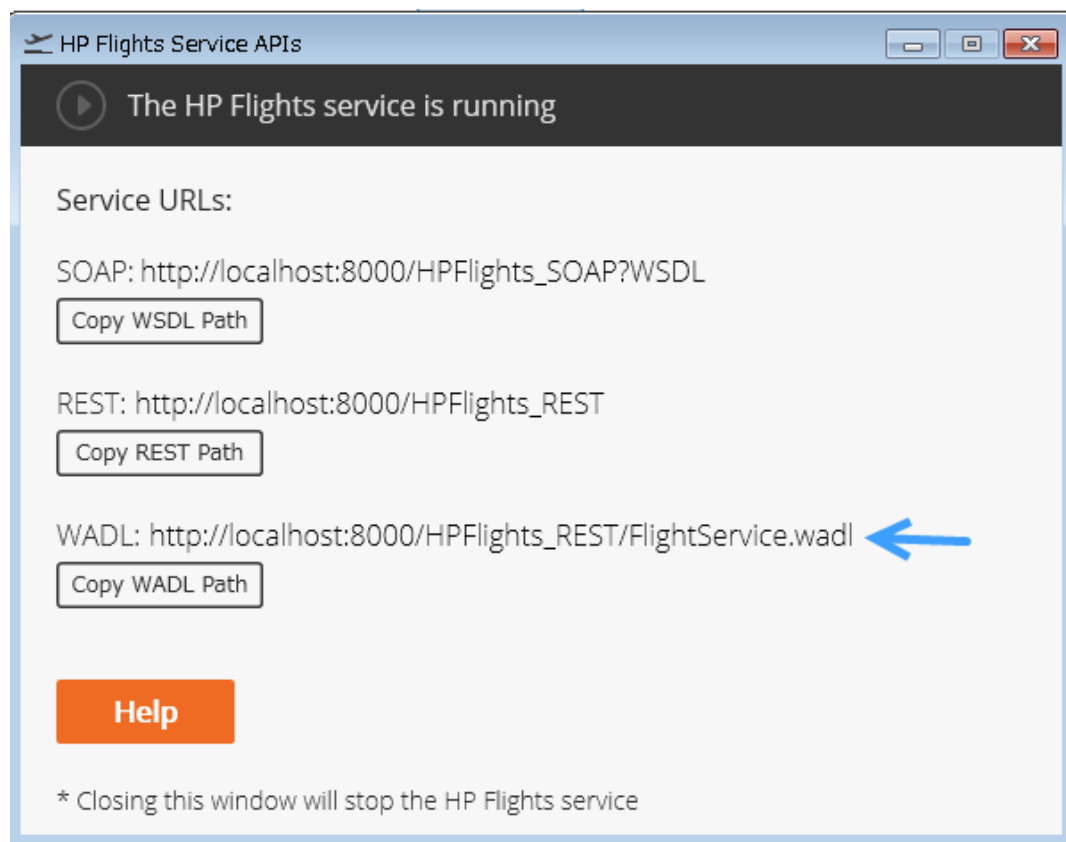
**注:** テストを編集および実行する際に UFT からアプリケーションにアクセスできるように、チュートリアルを使用する際には、このアプリケーションを開いたままにしておいてください。

### 2. WADL ファイルをインポートします。

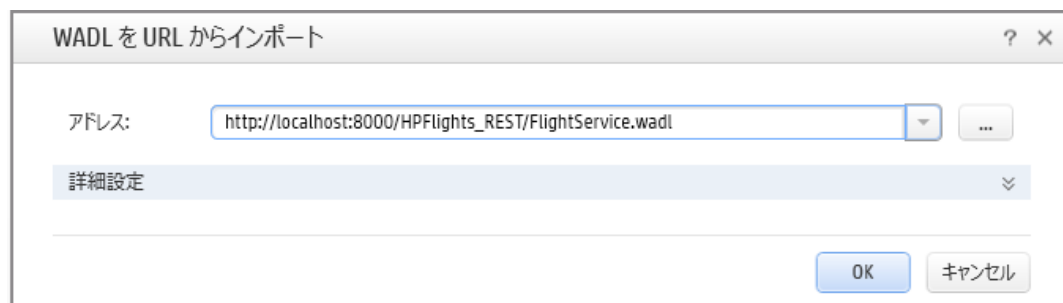
- a. UFT のツールバーで、**[REST サービスの追加]** ドロップダウン矢印をクリックして、**[WADL を URL からインポート]** を選択します。**[WADL を URL からインポート]** ダイアログ・ボックスが開きます。

**注:** ローカルに保存した WADL ファイルがある場合は、WADL ファイルを UFT にインポートできます。

- b. HP Flights Service API アプリケーション・ウィンドウで、WADL サービスの URL を見つけます。

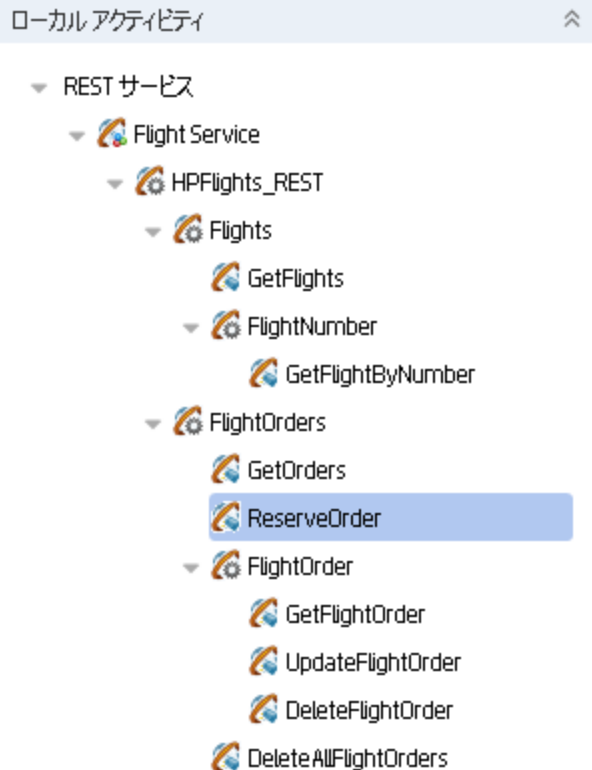


- c. HP Flights Service API アプリケーション・ウィンドウで、**[Copy WADL Path]** ボタンをクリックします。これで、WADL ファイルの URL を [WADL を URL からインポート] ダイアログ・ボックスにコピーできます。
- d. UFT の [WADL を URL からインポート] ダイアログ・ボックスの **[アドレス]** フィールドに、アプリケーション・ウィンドウからコピーした URL を貼り付けます。



- e. **【OK】** をクリックしてサービス記述を UFT にインポートします。

サービス記述が UFT にインポートされ、リソースとメソッドの階層が [ツールボックス] 表示枠の [ローカル アクティビティ] セクションに表示されます。



これでサービス記述とメソッドが UFT にインポートされ、これらのメソッドを使用してテストを作成する準備ができました。続いて [「練習 8c : Web アプリケーション・サービスのメソッドの編集」](#) (221 ページ) に進み、UFT でサービス記述のメソッドを編集する方法を学習します。

## 練習 8c : Web アプリケーション・サービスのメソッドの編集

「練習 8b : Web アプリケーション・サービス・モデルのインポート」 (219 ページ) では、Web アプリケーションのサービス記述とメソッドを UFT にインポートし、サービス、リソース、メソッドの階層を作成しました。

インポートする WADL には、サービスとメソッドの URL や各種メソッドのパラメータなど、サービスのリソースとメソッドに関する詳細な記述が含まれます。WADL ファイルを UFT にインポートした後に、これらのプロパティを変更することはできません。

ただし、メソッドの追加パラメータやメソッドの要求および応答情報など、サービス記述に情報を追加することはできます。この追加情報はサービス・モデルの一部として保存され、メソッドをキャンバスにドラッグする際にプロトタイプの情報として使用されます。

このレッスンでは、UFT でサービス・モデルのプロパティを編集する方法について学習します。また、GetFlights メソッドと ReserveOrder メソッドに情報を追加します。これらは、次の練習でテストを作成する際に使用します。

### 1. UFT を起動し、フライト予約アプリケーション・ソリューションを開きます。

- a. 必要に応じて、UFT を開きます (「すべてのテスト用のソリューションの作成」(25ページ)を参照)。
- b. 開始ページの [最近のソリューション] 領域で、[Flight Reservation Application] をクリックします。


ソリューション・エクスプローラに Flight Reservation Application ソリューションが開きます。これには、「練習 8a : Web アプリケーション・サービス用のテストの作成」(218ページ)で作成した Flights WADL テストが含まれます。

### 2. Web アプリケーションのサービス・モデル記述を開きます。

Web アプリケーションのサービス記述を編集する場合は、REST サービス・モデル記述の作成と編集に使用するのと同じダイアログ・ボックスとツールを使用します。

[ツールボックス] 表示枠の [ローカル アクティビティ] セクションで、[Flight Service] ノードを右クリックし、[サービスの編集] を選択します。[REST サービスの編集] ダイアログ・ボックスが開きます。

### 3. GetFlights メソッドの応答情報を編集します。

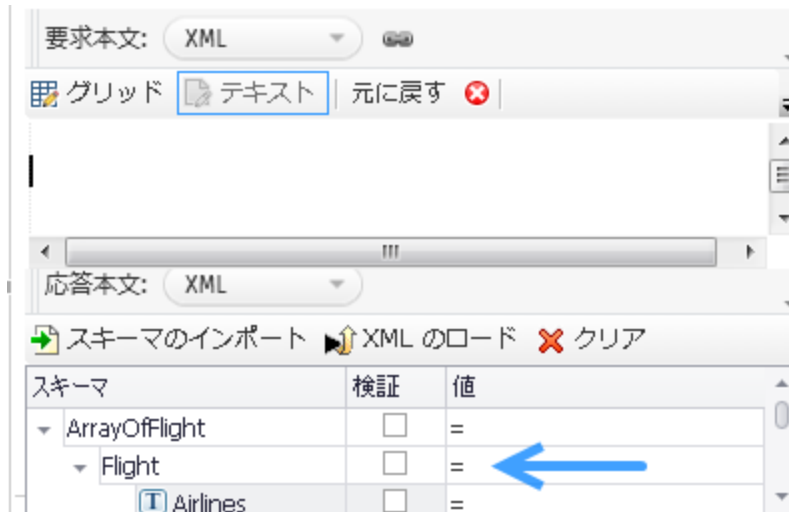
- a. [REST サービスの編集] ダイアログ・ボックスで、[Flights] ノードの下で [GetFlights] ノードを選択します。右の表示枠が更新され、メソッドの情報が反映されます。
- b. 右の表示枠で、[HTTP] タブ  を選択します。
- c. テキスト・エディタで、次の XML をコピーします。

```
<ArrayOfFlight xmlns="HP.SOAQ.SampleApp"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <Flight>
    <Airlines>AA</Airlines>
    <ArrivalCity>Denver</ArrivalCity>
    <ArrivalTime>01:23 PM</ArrivalTime>
    <DepartureCity>London</DepartureCity>
    <DepartureTime>06:12 AM</DepartureTime>
    <FlightNumber>20279</FlightNumber>
    <Price>112.2</Price>
  </Flight>
</ArrayOfFlight>
```


**注:** この場合、値は XML の一部のままにしておく必要があります。これにより、各パラ

メータで使用する値のタイプが指定されます。

- d. ファイルに **response.xml** という名前を付けてそれぞれが選択したディレクトリに保存します。
- e. UFT の [REST サービスの編集] ダイアログ・ボックスで [GetFlights] ノードを選択し、[HTTP] タブの [応答] セクションで、ドロップダウン・リストから [XML] を選択します。
- f. [応答本文] セクションで、[XML のロード] ボタンをクリックします。[開く] ダイアログ・ボックスが開きます。
- g. [開く] ダイアログ・ボックスで、**response.xml** ファイルを保存したディレクトリに移動し、**response.xml** ファイルを選択します。
- h. [開く] をクリックし、XML スキーマをサービス記述に追加します。  
応答グリッドの XML に基づいて応答プロパティが追加されます。



#### 4. ReserveOrders メソッドの要求情報を追加します。

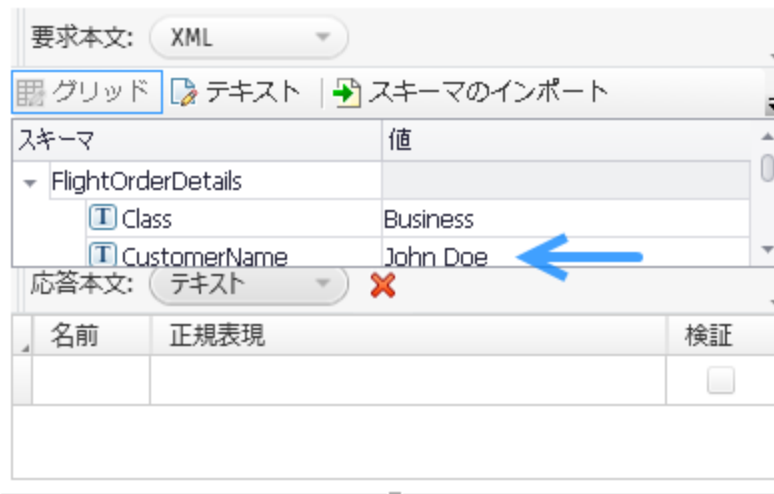
- a. [REST サービスの編集] ダイアログ・ボックスで、[FlightOrders] ノードの下に [ReserveOrder] ノードを選択します。右の表示枠が更新され、ReserveOrder メソッドの情報が反映されます。
- b. 右の表示枠で、[HTTP] タブ  を選択します。
- c. [HTTP] タブの [要求] セクションで、[要求本文] ドロップダウン・リストから [XML] を選択します。テキスト・エディタが開き、Insert XML here というメッセージが表示されます。
- d. テキスト入力領域に、次の XML を入力します。

```
<?xml version="1.0" encoding="utf-8"?>
<FlightOrderDetails xmlns="HP.SOAQ.SampleApp">
```

```
<Class>Business</Class>
<CustomerName>John Doe</CustomerName>
<DepartureDate>2012-12-12</DepartureDate>
<FlightNumber>1304</FlightNumber>
<NumberOfTickets>21</NumberOfTickets>
</FlightOrderDetails>
```

**注:** この場合、値は XML の一部のままだしておく必要があります。これにより、各パラメータで使用する値のタイプが指定されます。

【グリッド】 ボタンをクリックすると、プロパティ・グリッドに表示された XML 要求のパラメータを参照できます。



- e. 【OK】 をクリックして変更を保存します。

変更内容がサービス・モデルに保存され、【ツールボックス】表示枠でインポート済みのメソッドのプロパティが更新されます。

以上で Web アプリケーションのメソッドの編集と追加情報を設定が完了し、これらをテストで使用する準備ができました。続いて「[練習 8d : Web アプリケーション・サービスのメソッドを使用したテストの構築](#)」(224ページ)で、WADL からインポートしたメソッドを使用してテストを作成します。

## 練習 8d : Web アプリケーション・サービスのメソッドを使用したテストの構築

「[練習 8b : Web アプリケーション・サービス・モデルのインポート](#)」(219ページ)では、Web アプリケーション・サービスで使用するメソッドの記述を含む WADL ファイルをインポートしました。

「[練習 8c : Web アプリケーション・サービスのメソッドの編集](#)」(221ページ)では、WADL ファイル内で定義されていない追加のプロパティ情報をこれらのメソッドに追加しました。



「[レッスン 3 : テストへのステップの追加](#)」(46ページ)で作成した Book Flights GUI テストでは、アプリケーション・ウィンドウの順序は次のようになっていました。



同じアプリケーションの API テストを作成する際には、ステップがアプリケーションのフローにできるだけ一致するようにする必要があります。WADL ファイルからインポートしたメソッドのリストには、次のいくつかが含まれます。

- **GetFlights**
- **GetFlightOrders**
- **UpdateFlightOrder**
- **DeleteFlightOrder**
- **DeleteAllFlightOrders**
- **ReserveOrder**

ユーザ・インタフェースのフローと一致させるには、フライトを見つけた後に顧客の入力に基づいてフライトの申し込みを作成する API テスト・ステップを作成する必要があります。

この練習では、**GetFlights** と **ReserveOrder** の 2 つのステップを作成します。




1. **UFT を起動し、フライト予約アプリケーション・ソリューションを開きます。**
  - a. 必要に応じて、UFT を開きます (「[すべてのテスト用のソリューションの作成](#)」(25ページ)を参照)。
  - b. 開始ページの「最近のソリューション」領域で、「**Flight Reservation Application**」をクリックします。

ソリューション・エクスプローラに Flight Reservation Application ソリューションが開きます。これには、「練習 8a : Web アプリケーション・サービス用のテストの作成」(218 ページ) で作成した Flights WADL テストが含まれます。

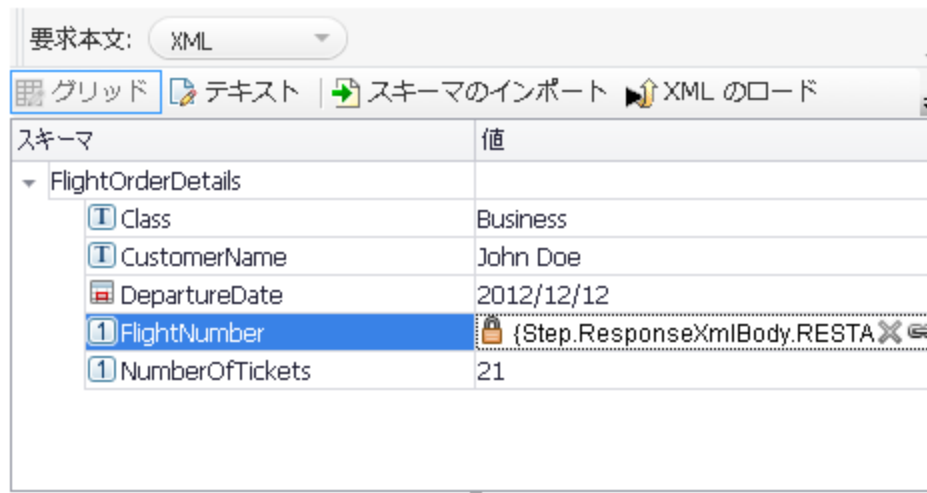
## 2. テスト・ステップを作成します。

- a. [ツールボックス] 表示枠の [ローカル アクティビティ] セクションで、( [Flight Service] ノードと [HPFlights\_REST] ノードの下にある ) [Flights] ノードを展開します。
- b. [Flights] ノードの下で、**GetFlights** メソッドをキャンバスにドラッグします。テスト・フローに **GetFlights** というブロックが追加されます。
- c. [FlightOrders] ノードの下で、[ReserveOrder] ノードをキャンバスにドラッグします。テスト・フローの GetFlights ブロックの下に **ReserveOrder** という別のブロックが追加されます。

## 3. [ReserveOrder] ステップの [FlightNumber] プロパティを、[GetFlights] ステップの出力にリンクします。

- a. キャンバスで、[ReserveOrder] ステップを選択します。[プロパティ] 表示枠に [入力/チェックポイント] タブ  が開きます。
- b. [プロパティ] 表示枠で、[HTTP] タブ  を選択します。
- c. [HTTP] タブの [要求本文] セクションで、[FlightNumber] プロパティの [値] セル内の [データ ソースにリンク] ボタンをクリックします。[リンク元の選択] ダイアログ・ボックスが開きます。
- d. [リンク元の選択] ダイアログ・ボックスで、[利用可能なステップ] オプションを選択します。[ステップの選択:] 表示枠 (左の表示枠) が更新され、利用可能なステップのリストが表示されます。
- e. [ステップの選択:] 表示枠で、[GetFlights] ステップを選択します。[プロパティの選択:] 表示枠 (右の表示枠) が更新され、利用可能なプロパティのリストが表示されます。
- f. [プロパティの選択:] 表示枠で、[HTTP] タブ  を選択します。HTTP プロパティのリストが表示されます。
- g. プロパティのリストで、[FlightNumber] プロパティを選択し、[OK] をクリックします。

【ReserveOrder】ステップの【FlightNumber】プロパティの値が更新され，【GetFlights】ステップの出力へのリンクが反映されます。






4. **テストで使用するデータ・ソースを追加します。**
  - a. 必要に応じて，【表示】 > 【データ】を選択し，【データ】表示枠を表示します。
  - b. 【データ】表示枠で，【新規データ ソース】ボタン をクリックして【Excel】を選択します。【新規 Excel データ ソース】ダイアログ・ボックスが開きます。
  - c. 【新規データ ソース】ダイアログ・ボックスの【Excel ファイルのパス】フィールドで，【参照】ボタンをクリックします。
  - d. 【開く】ダイアログ・ボックスで，<UFT インストール・ディレクトリ>\samples\Flights Application に保存されたアプリケーションの Excel ファイルに移動し，【OK】をクリックします。
  - e. 【新規 Excel データ ソース】ダイアログ・ボックスで，ファイルに「WADL\_Flights」という名前を付けます。
  - f. 【元の場所にある Excel ファイルへのリンク】オプションを選択します。
  - g. 【OK】をクリックしてデータ・ソース情報を保存し，Excel データをテストに追加します。
5. **【GetFlights】ステップの入力プロパティをデータ・ソースにリンクします。**
  - a. キャンバスで，【GetFlights】ステップを選択します。【プロパティ】表示枠に【入力/チェックポイント】タブ が開きます。
  - b. 【入力/チェックポイント】タブで，【DepartureCity】プロパティの【値】セルの【データ ソースにリンク】ボタン をクリックします。【リンク元の選択】ダイアログ・ボックスが開きます。
  - c. 【リンク元の選択】ダイアログ・ボックスで，【データ ソース カラム】オプションを選択します。【データ ソースを選択してください:】表示枠（左の表示枠）に，利用可能なすべてのデータ・ソースのリストが表示されます。

- d. [データソースを選択してください:] 表示枠で、[WADL\_Flights!Input] ノードを選択します。[データの選択:] 表示枠（右の表示枠）に、利用可能なすべてのデータ・パラメータ（カラム）のリストが表示されます。
- e. [データの選択:] 表示枠で、[DepartureCity] パラメータを選択し、[OK] をクリックします。[プロパティ] 表示枠の [DepartureCity] プロパティの [値] カラムが更新され、データ・ソースへのリンクが反映されます。
- f. 同様の手順を繰り返して、[ArrivalCity] プロパティと [Date] プロパティをデータ・ソースにリンクします。

GetFlights のプロパティの値にデータ・ソースへのリンクが表示されます。

入力	値
▼ プロパティ	
DepartureCity	{DataSource.WADL_Flights!Input.Departur
ArrivalCity	{DataSource.WADL_Flights!Input.ArrivalCit
Date	{DataSource.WADL_Flights!Input.Departur

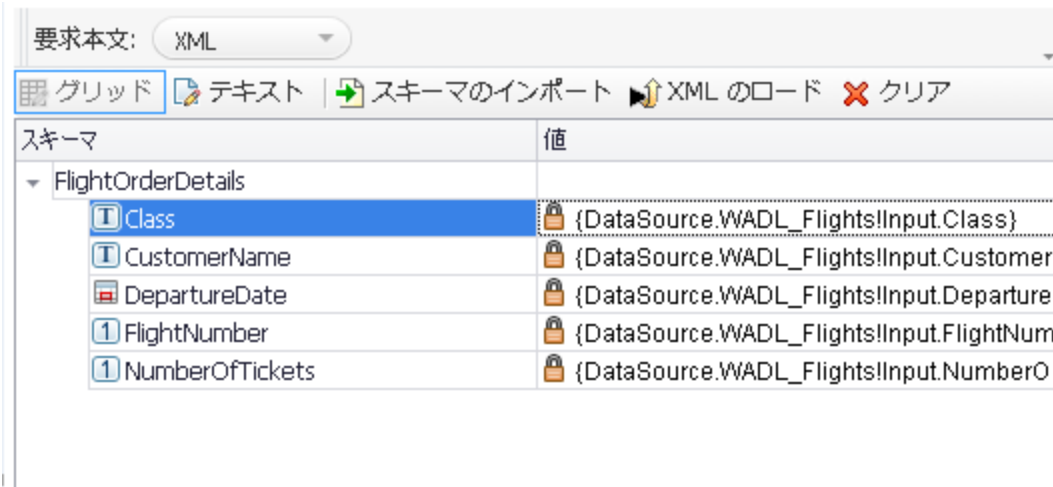
## 6. [ReserveOrder] ステップの HTTP 要求プロパティをデータ・ソースにリンクします。

- a. キャンバスで、[ReserveOrder] ステップを選択します。[プロパティ] 表示枠に [入力/チェックポイント] タブ  が開きます。
- b. [プロパティ] 表示枠で、[HTTP] タブ  を選択します。HTTP 要求プロパティと応答プロパティのリストが表示されます。
- c. [Class] プロパティの [値] セルで、[データソースにリンク] ボタン  をクリックします。[リンク元の選択] ダイアログ・ボックスが開きます。
- d. [リンク元の選択] ダイアログ・ボックスで、[データソース カラム] オプションを選択します。[データソースを選択してください:]（左の表示枠）に、利用可能なすべてのデータ・ソースのリストが表示されます。
- e. [データソースを選択してください:] 表示枠で、[WADL\_Flights!Input] ノードを選択します。[データの選択:] 表示枠（右の表示枠）に、利用可能なすべてのデータ・パラメータ（カラム）のリストが表示されます。
- f. [データの選択:] 表示枠で、[Class] カラムを選択し、[OK] をクリックします。[Class] プロパティの値が更新され、データ・ソースへのリンクが反映されます。


g. 他の HTTP 要求プロパティについても、同様の手順を繰り返します。

- **CustomerName**
- **DepartureDate**
- **NumberOfTickets**

すべてのプロパティ（[FlightNumber] プロパティを除く）のデータ・ソースへのリンクが完了すると、[プロパティ] 表示枠の値が更新されリンクが反映されます。



## 7. テストを保存します。

ツールバーで **保存**  をクリックします。

以上で WADL ファイルからインポートしたメソッドを使用したテストの作成が完了し、テストを実行して実行結果を表示する準備ができました。続いて「[練習 8e : Web アプリケーション・サービスのテストの実行](#)」(229ページ)でテストを実行します。

## 練習 8e : Web アプリケーション・サービスのテストの実行

「[練習 8d : Web アプリケーション・サービスのメソッドを使用したテストの構築](#)」(224ページ)では、以前の練習でインポートして編集した Web アプリケーション・サービス・モデルのメソッドを使用してテストを作成しました。

この練習では、作成したテストを実行し、結果を確認します。


### 1. UFT を起動し、フライト予約アプリケーション・ソリューションを開きます。

- a. 必要に応じて、UFT を開きます（「[すべてのテスト用のソリューションの作成](#)」(25ページ)を参照）。



- b. 開始ページの **【最近のソリューション】** 領域で、**【Flight Reservation Application】** をクリックします。

ソリューション・エクスプローラに Flight Reservation Application ソリューションが開きます。これには、**「練習 8a：Web アプリケーション・サービス用のテストの作成」** (218 ページ) で作成した Flights WADL テストが含まれます。

## 2. テストの反復回数を設定します。


- a. キャンバスで、テスト・フローを選択します。[プロパティ] 表示枠に **【入力】** タブ  が開きます。
- b. **【入力】** 表示枠で、**【'For' ループ】** オプションを選択します。
- c. **【反復回数】** フィールドに、8 と入力します。

## 3. データ・ソースのデータ・ナビゲーション・プロパティを設定します。

- a. キャンバスで、テスト・フローを選択します。[プロパティ] 表示枠に **【入力/チェックポイント】** タブ  が開きます。
- b. [プロパティ] 表示枠で **【データ ソース】** タブ  を選択します。テスト・フローに関連付けられたすべてのデータ・ソースのリストが表示されます。
- c. データ・ソースのリストで、**【WADL\_Flights!Input】** データ・ソースを選択し、**【編集】** をクリックします。[データナビゲーション] ダイアログ・ボックスが開きます。
- d. [データナビゲーション] ダイアログ・ボックスで、次のデータ・ナビゲーション・プロパティを設定します。

開始位置：	最初の行
移動単位：	1 行進む
終了位置：	最後の行
最終行への到達時	折り返す

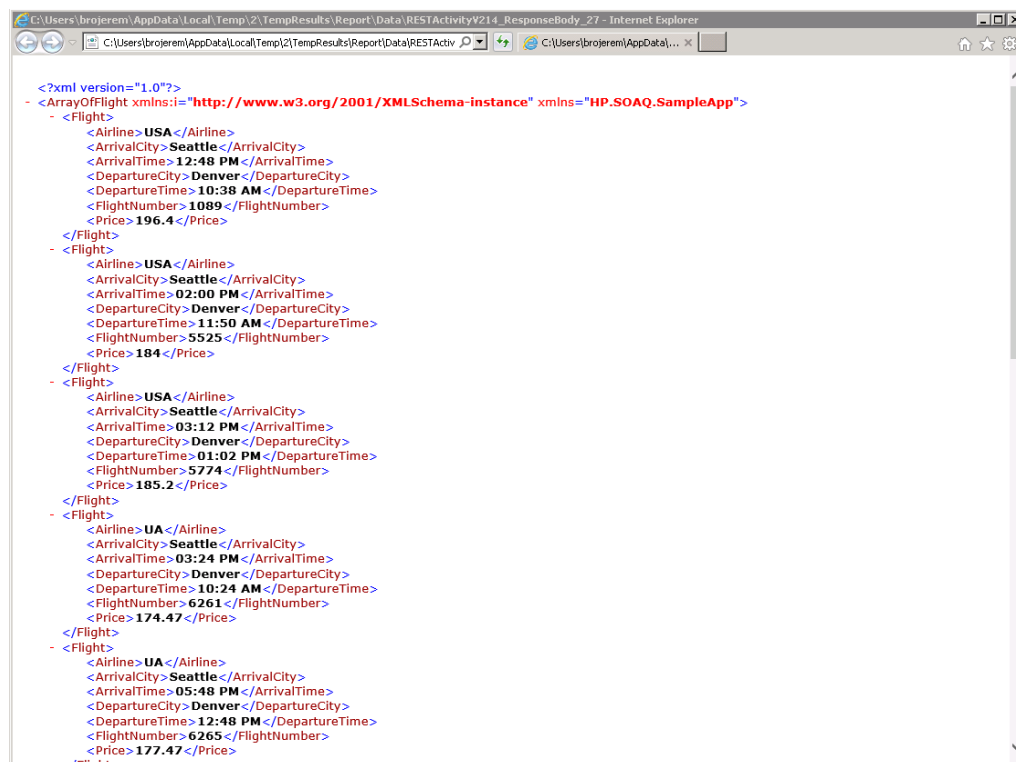
## 4. テストを実行します。

- a. HP Flights Service API アプリケーションが開いていることを確認します。
- b. ツールバーで **【実行】** ボタン  をクリックします。  
UFT でテスト・ステップが実行され、データ・ソースからプロパティ値が提供されます。テスト実行ログが **【出力】** 表示枠に表示されます。  
テストの実行が終了すると、実行結果が開きます。

## 5. 実行結果を分析します。

- a. テスト・フローで、**【ReserveOrder】** ステップを選択します。実行結果に、ステップのサマリが表示されます。

- b. キャプチャ・データで、要求と応答の情報を確認します。[要求本文] および [応答本文] セル内のリンクをクリックすると、XML 要求と応答の情報をブラウザ・ウィンドウで開くことができます。



# 第5部: 1 つのテストでの GUI および API テストの作成と実行

**注:** チュートリアルはこのパートを始める前に、「[GUI 自動テストの作成と実行](#)」(28ページ)および「[API 自動テストの作成と実行](#)」(148ページ)に記載されているレッスンを学習する必要があります。

アプリケーションをテストする際に包括的なテストを行うには、ユーザ・インタフェース (GUI) とサービス層 (API) の両方をテストする必要があります。これを行う場合の課題の 1 つは、アプリケーションのそれぞれの部分に対して別のテストを用意して実行することです。

UFT では、アプリケーションの GUI テストと API テストの両方を作成して管理する必要がありますが、1 つの統合テストでアプリケーションの GUI と API の両方をテストする統合テストを実行できます。GUI テストから API テストを呼び出すだけで、UFT が 1 つのテスト実行内でアプリケーションの両方の層を実行します。テスト実行が完了すると、実行結果が統合されて表示され、1 つのレポートで GUI 層と API 層の両方のパフォーマンスが報告されます。

このパートでは、1 つのテスト実行に GUI テストと API テストの両方を含むテストを作成して実行する方法を学習します。

本項には、次の項目が含まれています。

- [レッスン 1 : GUI テストと API テストをまとめて実行するテストの作成](#) ..... 233
- [レッスン 2 : API テストの GUI テストからの呼び出し](#) ..... 234
- [レッスン 3 : API テストを呼び出す GUI テストの実行](#) ..... 238



# レッスン 1：GUI テストと API テストをまとめて実行するテストの作成

このレッスンでは、GUI テストと API テストの統合テストを実行するため、別のテストを作成します。

## 1. UFT を起動し、Book Flights テストを開きます。

- a. UFT を開きます（「[すべてのテスト用のソリューションの作成](#)」(25ページ)を参照）。WPF Add-in がロードされていることを確認します。

- b. 開始ページの「最近のソリューション」領域で、「**Flight Reservation Application**」をクリックします。

Flight Reservation Application ソリューションが開きます。これには、**Book Flights** の GUI テスト（「[GUI 自動テストの作成と実行](#)」(28ページ)で作成）と API テスト（「[API 自動テストの作成と実行](#)」(148ページ)で作成）が含まれます。

## 2. Book Flights テストを Flight Reservation Application として保存します。

- a. ソリューション・エクスプローラで、「**Book Flights**」ノードを右クリックし、「名前を付けて保存」を選択します。「名前を付けて保存」ダイアログ・ボックスが開きます。
- b. 「テストに名前を付けて保存」ダイアログ・ボックスで、**C:\%HOMEPATH%\My Documents\Unified Functional Testing** に移動し、「Flight Reservation Application」という名前でテストを保存します。

ソリューション・エクスプローラで、Book Flights テストが新しい Flight Reservation Application テストに置き換えられます。Book Flights テストは、今までと同様にファイル・システムに別に保存されています。

## 3. Book Flights テストをソリューションに再び追加します。

- a. 「**ファイル**」>「**追加**」>「**既存のテスト**」を選択します。「テストをソリューションに追加」ダイアログ・ボックスが開きます。
- b. 「テストをソリューションに追加」ダイアログ・ボックスで、**C:\%HOMEPATH%\My Documents\Unified Functional Testing** に移動し、**Book Flights** テストを選択します。
- c. 「**追加**」をクリックして Book Flights テストをソリューションに戻します。

Book Flights テストがソリューション・エクスプローラに再度表示されます。

以上で GUI テストと API テストをまとめて実行するためのテストが用意できました。次に両方のタイプのテストを含むテストを作成できます。続いて「[レッスン 2：API テストの GUI テストからの呼び出し](#)」(234ページ)では、両方のタイプのテストを含むテストを作成する方法について学習します。

# レッスン 2：API テストの GUI テストからの呼び出し

API テストを GUI テストから実行するには、最初に API テストを呼び出す必要があります。このレッスンでは、API テストと GUI テストを 1 つの統合テストとして実行するため、GUI テストから API テストへの呼び出しを追加する方法について学習します。


## 1. UFT を起動し、Book Flights テストを開きます。

- a. 必要に応じて、UFT を開きます（「すべてのテスト用のソリューションの作成」(25ページ)を参照）。WPF Add-in がロードされていることを確認します。
- b. 開始ページの「最近のソリューション」領域で、「Flight Reservation Application」をクリックします。

Flight Reservation Application ソリューションが開きます。これには、「レッスン 1：GUI テストと API テストをまとめて実行するテストの作成」(233ページ)で作成した Flight Reservation Application テストが含まれます。


## 2. API テスト呼び出しで使用する新規アクションを作成します。

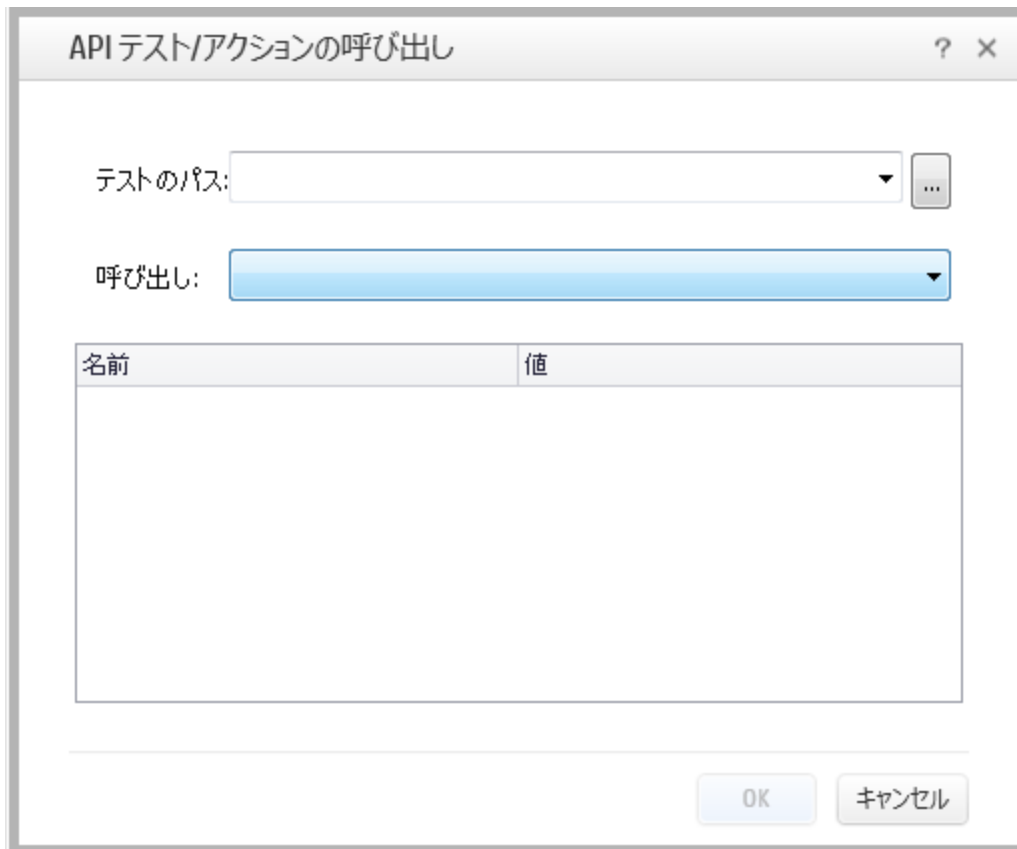
API テストを呼び出す場合、UFT は呼び出された API テスト全体を実行します。API テスト呼び出しは、アクション内の個別のステップとみなすことも、独立したアクションとみなすこともできます。

- a. ツールバーで、「新規アクションの呼び出し」ドロップダウン矢印  をクリックし、「新規アクションの呼び出し」を選択します。「新規アクションの呼び出しの挿入」ダイアログ・ボックスが開きます。
- b. 「新規アクションの呼び出しの挿入」ダイアログ・ボックスで、アクションに「API Test Call」という名前を付けます。
- c. 「テストの末尾」オプションが選択されていることを確認します。その他のオプションは標準設定のままにします。
- d. 「OK」をクリックしてアクションをテストに追加します。
- e. キャンバスで、API Test Call アクションを右クリックして「上に移動」を選択します。テスト・フローで API Test Call アクションが Flight Confirmation アクションの上に移動します。

## 3. API テスト呼び出しを追加します。

- a. キャンバスで、API Test Call アクションのブロックをダブルクリックします。アクションがドキュメント表示枠に別のタブとして表示されます。

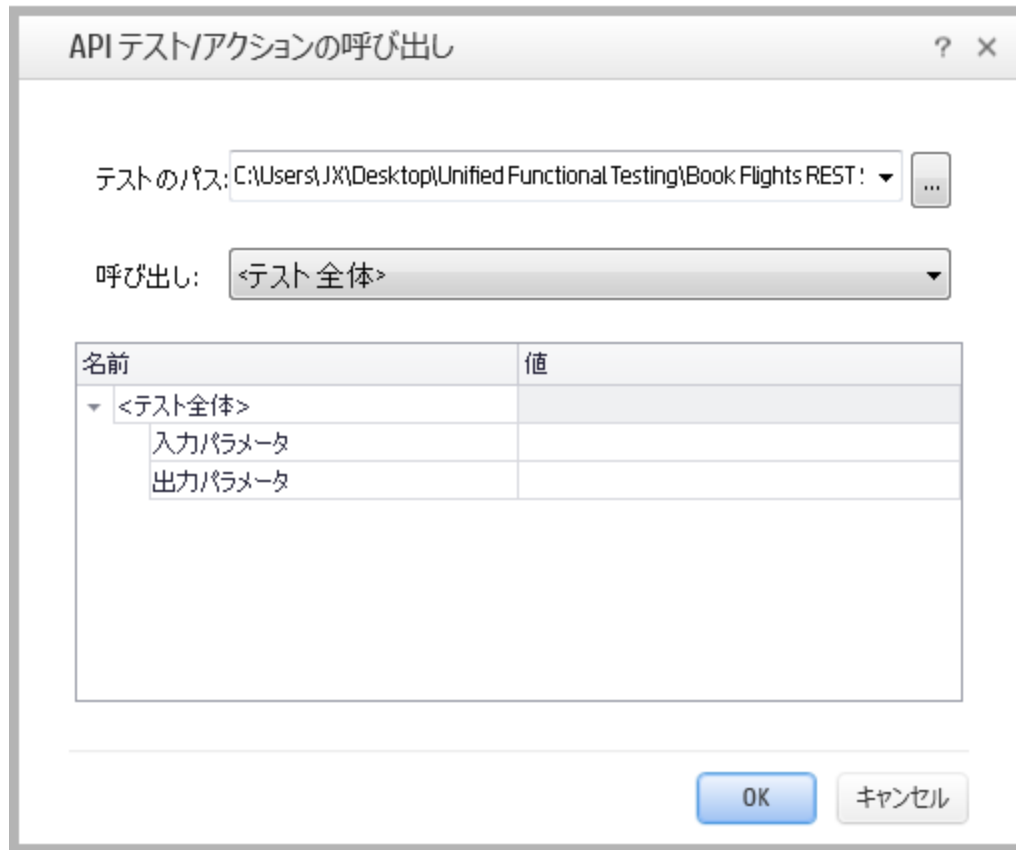
- b. ツールバーで、**【新規アクションの呼び出し】** ドロップダウン矢印  をクリックし、**【既存の API テスト/アクションの呼び出し】** を選択します。【API テスト/アクションの呼び出し】 ダイアログ・ボックスが開きます。



名前	値
----	---

- c. 【API テスト/アクションの呼び出し】 ダイアログ・ボックスの **【テストのパス】** フィールドで、**【参照】** ボタンをクリックします。【テストを開く】 ダイアログ・ボックスが開きます。
- d. 【テストを開く】 ダイアログ・ボックスで、**C:\%HOMEPATH%\Unified Functional Testing** に保存されている **Flight Reservation Application** ソリューションのテストを含むフォルダに移動します。

- e. このディレクトリで **Book Flights Web Service** テストを選択し、**開く** をクリックします。テストの名前とパラメータが、**API テスト/アクションの呼び出し** ダイアログ・ボックスに追加されます。



The dialog box titled "API テスト/アクションの呼び出し" (API Test/Action Call) contains the following elements:

- A text field for "テストのパス:" (Test Path) with the value "C:\Users\JX\Desktop\Unified Functional Testing\Book Flights REST!" and a browse button (...).
- A dropdown menu for "呼び出し:" (Call) with the selected option "<テスト全体>" (All Tests).
- A table with two columns: "名前" (Name) and "値" (Value).
 

名前	値
<テスト全体>	
入力パラメータ	
出力パラメータ	
- Buttons for "OK" and "キャンセル" (Cancel) at the bottom right.

**注:** テストに出力パラメータが含まれる場合は、このダイアログ・ボックスを使用して API テスト出力パラメータを格納する場所を指定できます。この場合、API テストに出力パラメータが含まれていないため、この手順は実行しません。

- f. **OK** をクリックして呼び出しをステップとしてテストに追加します。

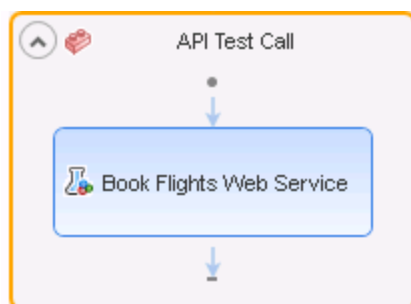
UFT は API テスト呼び出しのステップを API Test Call アクションに追加します。エディタでは、このステップは次のように表示されます。

RunAPITest "Book Flights Web Service"

キーワード・ビューでは、このステップは次のようになります。

項目	操作	値	注釈
関数の呼び出し + 新規ステップ	RunAPITest	"Book Flights Web Service"	"Book Flights Web Service" API テスト/アクションを実行します

また、UFT のキャンパスの API Test Call アクションにアクション呼び出しの視覚的表示が追加されます。



#### 4. テストを保存します。

【ファイル】 > 【保存】を選択します。

以上で API テストを呼び出すテスト・ステップの作成が完了し、テストを実行する準備ができました。続いて「[レッスン 3 : API テストを呼び出す GUI テストの実行](#)」(238ページ)では、テストを実行して実行結果を表示します。

# レッスン 3 : API テストを呼び出す GUI テストの実行

「[レッスン 2 : API テストの GUI テストからの呼び出し](#)」(234ページ)では、フライト予約アプリケーションの統合テストを 1 つのテストで実行できるように、API テストの呼び出しを既存の GUI テストに追加しました。

このレッスンでは、テストを実行し、実行結果を確認します。


## 1. UFT を起動し、Book Flights テストを開きます。

- a. 必要に応じて、UFT を開きます（「[すべてのテスト用のソリューションの作成](#)」(25ページ)を参照）。WPF Add-in がロードされていることを確認します。
- b. 開始ページの「**最近のソリューション**」領域で、「**Flight Reservation Application**」をクリックします。

Flight Reservation Application ソリューションが開きます。これには、「[レッスン 1 : GUI テストと API テストをまとめて実行するテストの作成](#)」(233ページ)で作成した Flight Reservation Application テストが含まれます。

## 2. Book Flights Web Service テストの反復回数を更新します。

Book Flights Web Service テストを作成したときには、テストを複数回反復して実行しました。ただし、（Book Flights Web Service テストを呼び出す）Flight Reservation Application テストでは、テストの反復は 1 回だけです。そのため、同じ反復回数を実行するように、API テストを変更する必要があります。

- a. ソリューション・エクスプローラで、「**Book Flights Web Service**」ノードの下ノードを展開します。
- b. 「Book Flights Web Service」ノードの下「**フロー**」ノードをダブルクリックします。Book Flights Web Service のテスト・フローがドキュメント表示枠に別のタブとして表示されます。
- c. キャンバスで、**テスト・フロー**を選択します。「プロパティ」表示枠に「**入力**」タブ  が開きます。
- d. 「入力」タブの「**反復回数**」フィールドで、回数を「1」に変更します。
- e. 「**ファイル**」 > 「**保存**」を選択して変更した設定を保存します。

## 3. Flight Reservation Application テストの実行の設定を設定します。

- a. ドキュメント表示枠で、「**Flight Reservation Application**」タブを選択します。  
「Flight Reservation Application」タブが閉じている場合は、ソリューション・エクスプローラで「**Flight Reservation Application**」ノードをダブルクリックして開きます。

- b. **【記録】 > 【記録と実行環境設定】** を選択します。**【記録と実行環境設定】** ダイアログ・ボックスが開きます。
- c. **【Windows アプリケーション】** タブで、**【指定したアプリケーションだけで記録して実行する:】** オプションと**【次の指定アプリケーション】** オプションが選択されていることを確認します。アプリケーションの詳細は、[「レッスン 4 : GUI テストの実行と分析」\(73ページ\)](#)で元の Book Flights テストを実行したときに保存されたものです。
- d. **【OK】** をクリックして、ダイアログ・ボックスを閉じます。

#### 4. **Flight Reservation Application テストを実行します。**

- a. テストを実行する前に、**HP MyFlight Sample Application** ウィンドウ（フライト予約アプリケーションのユーザ・インタフェース）が閉じていることを確認します。
- b. **HP Flights Service APIs** ウィンドウが開いていることを確認します。
- c. ツールバーで**【実行】** ボタン  をクリックします。**【実行】** ダイアログ・ボックスが開きます。
- d. **【実行】** ダイアログ・ボックスの**【結果保管場所】** タブで、**【新規実行結果フォルダ】** オプションを選択します。標準設定フォルダ名をそのまま使用します。
- e. **【実行】** をクリックしてテスト実行を開始します。

UFT で HP MyFlight Sample Application ウィンドウが開き、GUI テストで作成したとおりにアプリケーション・オブジェクトに対してステップが実行されます。テストが API テスト呼び出しに到達すると、GUI テストが一時停止し、API テストにフォーカスが移ります。

API テストの実行中は、出力表示枠に API テスト実行の進行状況を表示できます。

API テスト実行が終了すると、残りの GUI テストが実行されます。

すべてのテスト実行が終了すると、実行結果が開き、テスト結果が表示されます。

#### 5. **実行結果を表示します。**

実行結果で、**【テスト フロー】** を表示します。

GUI テスト内のアクションごとに個別のノードが存在することに注意してください。ただし、API テストの実行結果も実行結果の一部として表示されます。



API テスト結果内のノードを選択すると、ステップ・サマリにステップの詳細を表示できます。

6. **実行結果を閉じます。**

実行結果を表示した後に、実行結果を含むタブを閉じます。



# その他の情報

GUI 自動テストの作成、API 自動テストの作成、および GUI テストと API テストの両方を実行するテストの作成など、UFT の使用方法についての学習が完了し、UFT を使用してそれぞれのアプリケーションをテストする準備ができました。

アプリケーションをテストするときには、次の手順を踏むことをお勧めします。

## 1. アプリケーションを分析します。

- 開発環境を明らかにします。これにより、適切な UFT アドインをロードして、アプリケーションのオブジェクトをサポートできます。
- ユーザが実行するビジネス・プロセスを明らかにします。それに応じてテストとアクションを計画してください。
- テストの構成および含める操作を決定します。テストの目的を検討し、自分のアプリケーションと UFT がテストの目的に合うように設定されていることを確認してください。

この段階で、アプリケーションをテストする際に使用するテストとアクションの骨組みを作成できます。

## 2. テストのインフラストラクチャを準備します。

テスト内のオブジェクトを格納する方法を決定します。個々のアクションのオブジェクトは、対応するローカル・オブジェクト・リポジトリ、または1つ以上の共通の（共有）オブジェクト・リポジトリに格納できます。また、複数のアクションで、同じ共有オブジェクト・リポジトリを使用することもできます。

- **テストの経験があまりない場合：**アクションごとにローカル・オブジェクト・リポジトリを使用するとよいでしょう。これは標準の設定であり、オブジェクトはすべて自動的に各アクションのローカル・レポジトリに追加されます。
- **テストに慣れている場合：**多くの場合、1つまたは複数のアクションで使用可能な共有オブジェクト・リポジトリを使用するのが最も効率的です。オブジェクト情報は1か所に集められて保管されます。アプリケーションのオブジェクトに変更があった場合には、その1か所で情報を更新すれば複数のテスト内の複数のアクションに反映されます。

このチュートリアルでは説明していませんが、ローカル・オブジェクト・リポジトリから共有オブジェクト・リポジトリにテスト・オブジェクトをエクスポートすることもできます。また、オブジェクト・リポジトリはマージできます。

関数ライブラリを作成して UFT の機能を拡張することもできます。

詳細については、『HP Unified Functional Testing ユーザーズ・ガイド』を参照してください。

## 3. テストを作成します。

テスト・ステップの作成中は、ユーザがアプリケーションを利用するときに実行すると予想されるステップを実行します。

**4. テストを拡張します。**

- ページ、オブジェクト、テキスト文字列またはテーブルの特定の値を検索し検査するためのチェックポイントを追加します。
- テストの定数値をパラメータで置き換え、アプリケーションが、異なる複数のデータ・セットを使った場合に、同じ操作をどのように実行するかを検査します。

テストにロジックを追加するプログラミングと条件文およびループ文によって、テストをさらに拡張できます。詳細については、『HP Unified Functional Testing ユーザーズ・ガイド』を参照してください。

**5. テストをデバッグします。**

テストをデバッグし、テストをスムーズに滞りなく実行できるかどうかを検査します。詳細については、『HP Unified Functional Testing ユーザーズ・ガイド』を参照してください。

**6. テストを実行します。**

アプリケーションを対象にテストを実行し、アプリケーションが期待どおりに機能するか検査します。

**7. 実行結果を分析します。**

テスト結果を検証し、アプリケーションの不具合を特定します（チェックポイントやパラメータのテスト結果のどこに注目するかについては、このチュートリアルの該当箇所を参照してください）。

# Windows 8.X 以降のオペレーティングシステムでのUFTへのアクセス

**注:** 標準設定では、Windows 8.x 以降の [スタート] 画面と [アプリ] 画面は、Internet Explorer をメトロ・モードで開くように設定されています。ただし、コンピュータのユーザー・アカウント制御がオフになっている場合、Windows 8 は Internet Explorer をメトロ・モードで開きません。このため、[スタート] または [アプリ] 画面から HTML ショートカット (UFT ヘルプや Readme ファイルなど) を開こうとすると、エラーが表示されます。

この問題を解決するには、Internet Explorer の標準設定の動作を変更して、メトロ・モードで開かないようにできます。[インターネットのプロパティ] ダイアログ・ボックス > [プログラム] タブで、[リンクの開き方を選択] オプションの [デスクトップ上には常に Internet Explorer を表示] を選択します。詳細については、<http://support.microsoft.com/kb/2736601> および <http://blogs.msdn.com/b/ie/archive/2012/03/26/launch-options-for-internet-explorer-10-on-windows-8.aspx> を参照してください。

# フィードバックの送信



チュートリアルの改善点について、フィードバックをお寄せください。

フィードバックの送信先 : [sw-doc@hp.com](mailto:sw-doc@hp.com)

