HP UFT Web Add-in Extensibility

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

開発者ガイド

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。商用コンピュータソフトウェア、コンピュータ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 Corporation 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 を登録するには、https://softwaresupport.hp.com にアクセスし、**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。お客様の業務の管理に必要な対話型の技術支援ツールに素早く効率的にアクセスいただけます。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、https://softwaresupport.hp.com にアクセスして、［**My Software Support サインイン**］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 & Integrations and Best Practices

HP Software Solutions Now（https://h20230.www2.hp.com/sc/solutions/index.jsp）を参照してください。このサイトでは、HP ソフトウェアのカタログに記載された製品の説明を確認したり、情報を交換したり、ビジネスニーズを解決することができます。

Cross Portfolio Best Practices Library（https://hpln.hp.com/group/best-practices-hpsw）からは、様々なベストプラクティス文書や資料にアクセスすることができます。

目次

HP UFT Web Add-in Extensibility へようこそ 7
UFT Web Add-in Extensibility SDK について 7
UFT Web Add-in Extensibility 開発者ガイドについて 8
その他のオンライン・リソース 9

第1部: Web Add-in Extensibility を使った作業 11
第1章: UFT Web Add-in Extensibility の概要 12
UFT Web Add-in Extensibility について 13
Extensibility Accelerator 13
Web Add-in Extensibility の構成要素 14
どのような場合に Web Add-in Extensibility を使用するか 15
サンプル・カスタム・コントロールに対する標準設定の UFT サポートと Extensibility オプションの分析 17
Web Add-in Extensibility の実装方法 21
Web Add-in Extensibility のサンプル 22
第2章: ツールキットに対する UFT サポートの計画 24
ツールキットに対する UFT サポートの計画について 25
カスタム・ツールキットのサポートの作成の準備 25
ツールキット関連情報の決定 26
各カスタム・コントロール・タイプに対するサポート情報の決定 26
Web Add-in Extensibility 計画の理解チェックリスト 27
Web Add-in Extensibility の計画チェックリスト 29
その他の情報 30
第3章: ツールキットのサポートの開発 31
カスタム・ツールキットのサポートについて 32
ブラウザに依存しないサポートの開発 32
カスタム・ツールキット・サポート・セットの作成 33
テスト・オブジェクト設定ファイルについて 34
UFT がテスト・オブジェクト設定 XML をロードする方法 38
UFT がテスト・オブジェクト設定ファイルを結合する方法 38
既存のテスト・オブジェクト・クラスの拡張 40
カスタマイズしたテスト・オブジェクト・クラスに対するヘルプ・ファイルの作成 41
ツールキット設定ファイルについて 42
ツールキット・サポート・セット用の JavaScript 関数の設計 44
グローバル JavaScript メソッドとユーティリティ・メソッド 48
例： オブジェクトを識別するための兄弟コントロールの使用 49
カスタム Web コントロールに対して使用するテスト・オブジェクト・クラスを UFT に指示する方法 51

Conditions 要素の使用 52
ツールキット・サポート・セットの開発中のテスト 57
カスタム・サポートのログとデバッグ 60
Microsoft Windows イベント・ログの使用 60
JavaScript ファイルのデバッグ（Internet Explorer のみ） 62
テスト・オブジェクト・メソッドのサポートの実装 63
メソッド引数に対する値の動的なリストのサポート 65
認識プロパティのサポートの実装 67
テスト・オブジェクト名のカスタマイズ 68
子コントロールの学習のためのフィルタの実装 69
記録のサポートの実装 70
トラブルシューティングと制限事項 - サポートの開発 73
第4章: ツールキット・サポート・セットのデプロイ 74
カスタム・ツールキット・サポートのデプロイについて 75
カスタム・ツールキット・サポートのデプロイ 75
設計段階で使用する設定 77
デプロイ済みサポートの変更 77
テスト・オブジェクト設定ファイルで指定された認識プロパティ属性の変更 78
デプロイ済みサポートの削除 79

第2部: チュートリアル：Web カスタム・ツールキットのサポートの作成方法の学習 80
第5章: 簡単な Web カスタム・コントロールに対する UFT サポートの作成方法の学習 81
このレッスンの準備 82
Web Add-in Extensibility Book サンプル・ツールキットのサポートの計画 82
Web Add-in Extensibility の計画チェックリスト 88
ツールキット・サポート・セットの開発 90
第 1 段階：ツールキット・サポート・セットの作成 90
第 2 段階：WebExtSample 環境の UFT への導入 91
ツールキット設定ファイルの設計 91
テスト・オブジェクト設定ファイルの設計 92
第 3 段階：Book コントロールの識別，調査，学習の実行を UFT に指示 93
ツールキット・サポート・セットのデプロイとテスト（第 3 段階） 95
第 4 段階：WebExtBook テスト・オブジェクト・メソッドのサポートを実装 98
ツールキット・サポート・セットのデプロイとテスト（第 4 段階） 100
第 5 段階：WebExtBook の認識プロパティのサポートを実装 101
ツールキット・サポート・セットのデプロイとテスト（第 5 段階） 102
第 6 段階：テスト・オブジェクトの名前の変更 103
ツールキット・サポート・セットのデプロイとテスト（第 6 段階） 104
第 7 段階：子オブジェクトを学習対象から除外するフィルタを実装 104
ツールキット・サポート・セットのデプロイとテスト（第 7 段階） 105

第 8 段階：Book コントロール上の記録のサポートを実装 106
ツールキット・サポート・セットのデプロイとテスト（第 8 段階） 108
第 9 段階：AuthorName の動的な値リストのサポートを実装 109
ツールキット・サポート・セットのデプロイとテスト（第 9 段階） 111
レッスンのまとめ 111
その他の情報 111
第6章: 複雑な Web カスタム・コントロールに対する UFT サポートの作成方法の学習 113
このレッスンの準備 114
Web Add-in Extensibility のサンプル・コントロール UsedBooks のサポートの計画 114
Web Add-in Extensibility の計画チェックリスト 119
ツールキット・サポート・セットの開発 121
第 1 段階：追加のコントロールをサポートするためにツールキット・サポート・セットを拡張 121
第 2 段階：UsedBooks コントロールの識別，調査，学習の実行を UFT に指示 122
ツールキット・サポート・セットのデプロイとテスト（第 2 段階） 124
第 3 段階：WebExtUsedBooks テスト・オブジェクト・メソッドのサポートを実装 126
ツールキット・サポート・セットのデプロイとテスト（第 3 段階） 129
第 4 段階：WebExtUsedBooks 認識プロパティとテスト・オブジェクト名のサポートを実装 129
ツールキット・サポート・セットのデプロイとテスト（第 4 段階） 130
第 5 段階：子オブジェクトを学習対象から除外するフィルタを実装 130
ツールキット・サポート・セットのデプロイとテスト（第 5 段階） 131
第 6 段階：UsedBooks コントロール上の記録のサポートを実装 132
ツールキット・サポート・セットのデプロイとテスト（第 6 段階） 133
レッスンのまとめ 134
その他の情報 134

フィードバックをお送りください 135

HP UFT Web Add-in Extensibility へようこそ

HP UFT Web Add-in Extensibility は，UFT Web Add-in で標準ではサポートされない，サードパーティ製およびカスタムの Web コントロールを使用するテスト・アプリケーションをサポートできるようにする SDK（ソフトウェア開発キット）パッケージです。

また，Extensibility によって HP Sprinter を使用できるようになります。これは，標準ではサポートされない Web オブジェクトの手動テストを効率化する HP のソリューションです。Sprinter の詳細については，『HP Sprinter ユーザーズ・ガイド』を参照してください。

本章の内容

- UFT Web Add-in Extensibility SDK について 7
- UFT Web Add-in Extensibility 開発者ガイドについて 8
- その他のオンライン・リソース 9

UFT Web Add-in Extensibility SDK について

Extensibility Accelerator for HP Functional Testing をインストールすると，UFT Web Add-in Extensibility SDK もインストールされます。この SDK には次の内容が含まれます。

- UFT Web Add-in を拡張する API。これにより，Web のカスタム・コントロールをサポートします。

- Web Add-in Extensibility ヘルプ（以下を含む）
  - 開発者ガイド。詳しい手順を説明するチュートリアル形式で，サンプルのカスタム・コントロールのサポートを開発します。
  - API Reference
  - Toolkit Configuration Schema Help
  - UFT Test Object Schema Help

  このヘルプは［**スタート**］ > ［**すべてのプログラム**］ > ［**HP Software**］ > ［**HP Unified Functional Testing**］ > ［**Extensibility**］ > ［**Documentation**］から利用可能です。
- 印刷用（PDF）バージョンの開発者ガイド（**<Extensibility Accelerator インストール・ディレクトリ>\help** フォルダに収録）。
- Web Add-in Extensibility ツールキット・サポート・セット。次の Web 2.0 環境で UFT GUI テスト サポートを拡張します。
  - ASP.NET AJAX Control Toolkit
  - Google Web Toolkit

- Yahoo User Interface
- Dojo

- サンプル Web ツールキット。**Book** と **UsedBooksTable** という名前のコントロールが含まれます（Extensibility Accelerator をインストールした場合）。

Windows 8 オペレーティング・システムからの UFT Web Add-in Extensibility へのアクセス

以前の Windows バージョンの［**スタート**］メニューからアクセス可能な UFT ファイルは，Windows 8 では［**スタート**］画面または［**アプリ**］画面からアクセス可能です。

- **アプリケーション（.exe ファイル）**。Windows 8 では，UFT アプリケーションには［**スタート**］画面から直接アクセスできます。たとえば，UFT をスタートするには，［**HP Unified Functional Testing**］ショートカットをダブルクリックします。
- **プログラム以外のファイル**。ドキュメントには［**アプリ**］画面からアクセスできます。

> **注:** 以前の Windows バージョンと同じく，UFT のコンテキスト・ヘルプにアクセスするには **F1** キーを押します。また，［**Help**］メニューからすべてのドキュメントと外部リンクにアクセスできます。

UFT Web Add-in Extensibility 開発者ガイドについて

このガイドでは，UFT Web Add-in Extensibility をセットアップし，UFT GUI テスト を拡張することによってサードパーティが提供またはカスタマイズされた Webコントロールをサポートする方法について説明します。

このガイドを使用するには UFT 機能に関する知識が必要です。さらに，Web Add-in Extensibility ヘルプ（［**スタート**］ > ［**すべてのプログラム**］ > ［**HP Software**］ > ［**HP Unified Functional Testing**］ > ［**Extensibility**］ > ［**Documentation**］ > ［**Web Add-in Extensibility Help**］）で提供されている，以下のドキュメントを参照してください。

- API Reference
- Toolkit Configuration Schema Help
- Test Object Schema Help

このドキュメントに併せて，次の UFT ドキュメント（UFT と一緒にインストールされます。UFT メイン・ウィンドウの［**ヘルプ**］ > ［**HP Unified Functional Testing ヘルプ**］からアクセスできます）も参照してください。

- 『HP Unified Functional Testing ユーザーズ・ガイド』
- 『HP Unified Functional Testing アドイン・ガイド』のWeb の項
- 『HP UFT Object Model Reference for GUI Testing』

注:

本書の情報，例，画面キャプチャは，特に UFT GUI テストで作業するものに的を絞っています。ただし，Web Add-in Extensibility を使用して開発したツールキット・サポート・セットは，すべて Sprinter の機能拡張に使用でき，標準ではサポートされない Web コントロールのサポートが可能になります。このガイドに記載されている UFT に関する内容は，すべて UFT と Sprinter の両方に適用されます。

さらに，このガイドの内容のほとんどはコンポーネントにも適用できます。

ビジネス・コンポーネントは HP Business Process Testing の一部です。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。

Windows 8 で作業する場合，［**アプリ**］画面から UFT のドキュメントとその他のファイルにアクセスしてください。

このガイドをより効果的にトピックまたはキーワードで検索できるようにするには，次のオプションを使用します。

- **AND**，**OR**，**NEAR**，**NOT** の各論理演算子。［検索］ボックスの横にある矢印をクリックして選択します。
- **以前の結果から検索：**［**検索**］タブの下部から使用できます。
- **類似する文字に合致：**［**検索**］タブの下部から使用できます。
- **タイトルのみ検索：**［**検索**］タブの下部から使用できます。

ヒント: 検索結果からヘルプ・ページを開く場合，検索文字列を含む箇所が折りたたまれていることがあります。ページで文字列が見つからない場合は，ドロップダウン・セクションをすべて展開して Ctrl + F キーで文字列を検索してください。

最新の更新のチェック，またはご使用のドキュメントが最新版かどうかのご確認には，HP ソフトウェア製品マニュアル Web サイト（http://h20230.www2.hp.com/selfsolve/manuals）をご利用ください。

その他のオンライン・リソース

次の追加オンライン・リソースが利用可能です。

リソース	説明
トラブルシューティングとナレッジ・ベース	セルフソルブ・ナレッジベースを検索できる HP ソフトウェア・サポート Web サイトのトラブルシューティング・ページ。この Web サイトの URL は，http://h20230.www2.hp.com/troubleshooting.jsp です。
HP ソフトウェア・サポート	HP ソフトウェア・サポート Web サイト。このサイトで，セルフソルブ・ナレッジ・ベースを参照できます。また，英語版のサイトでは，ナレッジ・ベースの参照，独自の項目の追加，ユーザ・ディスカッション・フォーラムへの書き込みや検索，パッチや更新されたドキュメントのダウンロードなどを行うこともできます。この Web サイトの URL は，

リソース	説明
	www.hp.com/go/hpsoftwaresupport です。 • 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。 • アクセスレベルに関する詳細は，以下の Web サイトにアクセスしてください。 http://h20230.www2.hp.com/new_access_levels.jsp • HP Passport ユーザ ID の登録は，次の場所で行います。 http://h20229.www2.hp.com/passport-registration.html
HP ソフトウェア Web サイト	HP ソフトウェア Web サイト。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。この Web サイトの URL は，www.hp.com/go/software です。

第1部: Web Add-in Extensibility を使った作業

第1章: UFT Web Add-in Extensibility の概要

UFT Web Add-in Extensibility を使用すると，購入時の UFT Web アドインではサポートされていないサードパーティ製およびカスタムの Web コントロールに対する高いレベルのサポートを提供できるようになります。

本章の内容

- UFT Web Add-in Extensibility について 13
- Extensibility Accelerator 13
- Web Add-in Extensibility の構成要素 14
- どのような場合に Web Add-in Extensibility を使用するか 15
- Web Add-in Extensibility の実装方法 21
- Web Add-in Extensibility のサンプル 22

UFT Web Add-in Extensibility について

UFT Web Add-in は，広く用いられるいくつかの Web コントロールに対する組み込みのサポートを備えています。UFT Web Add-in Extensibility を使用すると，このサポートを拡張し，UFT に追加の Web コントロールを認識させることができます。また，Web Add-in Extensibility を使用して，UFT にコントロールのグループや Web 要素をひとつのオブジェクトとして認識するように指示することもできます。たとえば，検索ボタンと編集ボックスが検索ボックスを構成している場合，UFT がそれらをひとつの関数ユニットとして表示するようにしたいことがあります。

UFT がアプリケーション内のオブジェクトを学習する場合，オブジェクトを特定のテスト・オブジェクト・クラスに属するものとして認識します。これによって，UFT 内でアプリケーションのオブジェクトを表すテスト・オブジェクトの認識プロパティとテスト・オブジェクト操作が決定されます。

UFT が Extensibility なしで Web ページ上のコントロールを学習する場合，いくつかの種類の要素は無視され，それによって定義されるコントロールを表すテスト・オブジェクトは作成されません。

購入時の Web Add-in でサポートされないその他の Web コントロールに対しては， UFT は汎用の WebElement テスト・オブジェクトを作成します。このタイプのテスト・オブジェクトには，テストしている Web コントロールに固有の特性がない可能性があります。このため，このテスト・オブジェクトを使用したテスト・ステップの作成を試みる場合，使用可能な認識プロパティとテスト・オブジェクト操作では不十分なことがあります。

たとえば，特別な種類のテーブルであるカスタム Web コントロールが，UFT からは通常の WebElement として認識されたとします。WebElement テスト・オブジェクトは **GetCellData** 操作をサポートしません。テーブルのセルからデータを取得するテスト・ステップを作成するには，テーブルの各セルを表すテスト・オブジェクトを作成し，該当するセルのテスト・オブジェクトにアクセスしてデータを取得する複雑なテストを作成する必要があります。

また，追加した複数の html テーブル要素で構成される Web ページ上のテーブルについて考えます。UFT がこの集計テーブルをひとつのオブジェクトとして認識できるようにするルールを定義し，関連するテスト・オブジェクト・メソッドを提供できます。このようなメソッドの例として，集計テーブルに関する行番号を受け取り，実際にその行が含まれる html テーブルを認識するようなメソッドがあります。

Web Add-in Extensibility を使用して Web コントロールのサポートを作成することで，UFT に特定のテスト・オブジェクト・クラスに属するコントロールまたはコントロールのグループを認識することを指示したり，テスト・オブジェクトの動作を指定したりできます。また，UFT が認識する使用可能なテスト・オブジェクト・クラスのリストを拡張することもできます。これにより，カスタム Web コントロール固有の動作を完全にサポートするテストを作成することが可能です。

Extensibility Accelerator

ますます多くの Web アプリケーションが Web 2.0 ベースのツールキット（ASP.NET AJAX，Dojo，YahooUI，GWT，JQueryUI など）を使用して，動的で対話型のコンテンツを自身のサイトに追加して

います。これらのツールキットのコントロールは複雑で，高度で柔軟性の高いテスト機能が求められます。

UFT Web Add-in Extensibility を使用すると，Web Add-in を拡張して，UFT が各種コントロールを認識してやり取りする方法をカスタマイズできます。これまで，Web Add-in Extensibility を使用するということは，手動でツールキット・サポート・セットを開発し保守することでした。

Extensibility Accelerator for HP Functional Testing は，Visual Studio ライクな IDE で，これらのサポート・セットの設計，開発，デプロイメントを容易にします。必要な拡張 XML ファイルを素早く簡単に設計，開発できるため，UFT とカスタムの Web コントロールとの連携動作を可能にする JavaScript 関数の開発に専念することができます。

Extensibility Accelerator ユーザ・インタフェースは，新しいテスト・オブジェクト・クラス，操作，プロパティを定義するときに役に立ちます。このインタフェースには，定義したテスト・オブジェクト・クラスをアプリケーション内のコントロールに割り当てるときに使用できるポイント・アンド・クリック・メカニズムも用意されています。Extensibility Accelerator のデプロイメント機能を使用すると，新しいツールキット・サポート・セットを UFT に自動的にデプロイするか，パッケージ化してほかの UFT ユーザと共有できます。

Unified Functional Testing のセットアップ・プログラムの［**アドインによる機能拡張と Web 2.0 ツールキット**］オプションから利用可能な Extensibility Accelerator for HP Functional Testing のインストール。

> **注:** この処理の一環として，html ページがブラウザで開きます。インストールを完了するには，このページは Internet Explorer で開く必要があります。

Web Add-in Extensibility の構成要素

次の各項では，UFT オブジェクト・サポートを構成する主な要素について説明します。これらの要素は，Web Add-in Extensibility の構成要素です。これらの要素の既存のサポートを拡張することで，必要なサポートを開発して，有意義で保守が容易なテストを作成できます。

テスト・オブジェクト・クラス

UFT では，アプリケーションの各オブジェクトは，それぞれのテスト・オブジェクト・クラスのテスト・オブジェクトで表されます。テスト・オブジェクト・クラスは，このテスト・オブジェクトに対して UFT で使用可能な認識プロパティとテスト・オブジェクト・メソッドのリストを決定します。UFT のキーワード・ビューやオブジェクト・リポジトリでテスト・オブジェクトを表すために使用されるアイコンも，テスト・オブジェクト・クラスによって決定されます。

テスト・オブジェクト名

UFT がオブジェクトを学習するときには，ページ上の各テスト・オブジェクトに対して一意の名前を作成します。わかりやすいテスト・オブジェクト名を使用すれば，同じクラスのテスト・オブジェクトの間の区別が可能になり，オブジェクト・リポジトリやテストでの識別が容易になります。

テスト・オブジェクトに標準設定で付けられる名前は，テスト・オブジェクト・クラスの名前です（ページ上に同じクラスのテスト・オブジェクトが複数存在する場合はインデックスが追加されます）。多くの場合，これは必ずしもカスタム・コントロールに最適な名前ではありません。

テスト・オブジェクト名は，UFT のユーザにとって意味がある必要があり，ツールキットに関連する用語を使用しているのが望ましいといえます。UFT はこの名前を，キーワード・ビュー，エディタ，オブジェクト・リポジトリに表示します。

テスト・オブジェクトの認識プロパティ

Web コントロールを表すのに使用されるテスト・オブジェクト・クラスによって，テスト・オブジェクトで使用できる認識プロパティのリストが決定されます。また，コントロールを一意に認識するために使用される認識プロパティ，チェックポイントおよび出力値に使用できる認識プロパティ（［チェックポイントのプロパティ］および［出力値のプロパティ］ダイアログ・ボックス），チェックポイントに対して標準設定で選択される認識プロパティも，これによって決まります。ただし，認識プロパティの実際の値は，Web コントロールから得られます。したがって，同じテスト・オブジェクト・クラスのテスト・オブジェクトで表される複数の Web コントロールが，同じ認識プロパティに対して異なる定義を持つ場合もあります。

テスト・オブジェクト・メソッド

テスト・オブジェクトのテスト・オブジェクト・メソッドのリストは，Web コントロールを表すテスト・オブジェクト・クラスによって決定されます。ただし，同じテスト・オブジェクト・クラスのテスト・オブジェクトで表される Web コントロールの間でも，同じテスト・オブジェクト・メソッドの動作が異なる可能性があります。その理由は，Web コントロールのタイプに応じて，UFT がテスト・オブジェクト・メソッドを異なる方法で実行する場合があるからです。

イベントの記録

UFT GUI テストの作成方法の 1 つに，アプリケーションの操作を記録する方法があります。記録セッションを開始すると，UFT はアプリケーション内のオブジェクトで発生するイベントをリッスンし，対応するテスト・ステップを書き込みます。Web コントロールに対して UFT がリッスンできるイベントと，発生したイベントに対して記録するテスト・ステップは，Web コントロールを表すテスト・オブジェクト・クラスによって決まります。

どのような場合に Web Add-in Extensibility を使用するか

UFT Web アドインは，ほとんどの Web コントロールに対してある程度のレベルのサポートを提供しますが，DIV または SPAN 要素として定義されたコントロールは無視します。カスタム Web コントロールに対するサポートを拡張する前に，UFT の観点からコントロールを分析して，このサポートの範囲を調べ，サポートのどの要素を変更するかを決定します。

カスタム Web コントロールを分析するには，オブジェクト・スパイ，キーワード・ビュー，エディタ，記録オプションを使用します。「Web Add-in Extensibility の構成要素」を参考に，要素のチェックを行ってください。

既存のオブジェクトの認識や動作が不十分な場合は，以下に示す例のように，Web コントロールは Web Add-in Extensibility の候補となります。

- UFT がニーズに一致しないテスト・オブジェクト・クラスを使用してコントロールを認識する場合。Web Add-in Extensibility を使用して，作成した新しいテスト・オブジェクト・クラスに属するものとしてカスタム・コントロールを認識するように UFT に指示することができます。
- UFT がコントロールに対して使用するテスト・オブジェクト・クラスはニーズを満たしているが，一部のテスト・オブジェクト・メソッドや認識プロパティの動作をカスタマイズする必要がある場合。Web Add-in Extensibility を使用して，UFT が使用しているクラスを拡張する新しいテスト・オブジェクト・クラスを作成し，該当するプロパティやメソッドの実装をカスタム実装でオーバーライドして，新しいテスト・オブジェクト・クラスを使用するように UFT に指示することができます。
- 特定のコントロール・タイプのすべてのオブジェクトに対して UFT が生成するテスト・オブジェクト名が同一（一意のカウンタを除く）である場合や，テスト・オブジェクトに使用される名前が対応するコントロールを明確に示さない場合。Web Add-in Extensibility を使用して，UFT が使用しているクラスを拡張する新しいテスト・オブジェクト・クラスを作成し，新しいクラスのテスト・オブジェクトの名前を UFT が付ける方法を変更し，新しいテスト・オブジェクト・クラスを使用するように UFT に指示することができます。
- UFT がカスタム・コントロール内の個々のサブコントロールを認識するが，メイン・コントロールを適切に認識しない場合。たとえば，メイン・カスタム・コントロールが時と分の数字を表示したエディット・ボックスを持つデジタル時計である場合，時刻の変更をエディット・ボックスに対する **Set** 操作でなく時計コントロールに対する **SetTime** 操作として認識させるようにします。Web Add-in Extensibility を使用して，子コントロールに発生するイベントの処理方法を変更できます。
- 記録セッション中に，コントロールに対して操作を実行するかイベントをトリガしたときに，UFT が全くステップを記録しないか，コントロールの動作に固有でないステップを記録する場合。あるいは，1 つの操作とみなすべきイベントに対して UFT が複数のステップを記録したり，ステップを記録すべきでないときに記録する場合もあります。

  Web イベントの設定を変更することにより，既存の Web オブジェクトの各タイプに対して記録するイベントを設定できます。詳細については，『HP Unified Functional Testing アドイン・ガイド』で Web イベント記録の設定に関する項を参照してください。

  Web イベントの設定では記録を十分にカスタマイズできない場合，たとえば特定のイベントが発生したときに UFT が記録するステップを変更する必要がある場合には，Web Add-in Extensibility を使用することができます。

サンプル・カスタム・コントロールに対する標準設定の UFT サポートと Extensibility オプションの分析

次の例は，Web Add-in Extensibility を使用して UFT のカスタム・コントロールのサポートを改善する方法を示します。

次に示す Book コントロールは，インターネット上で販売される本を表します。このコントロール固有のサポートは UFT にはありません。

このコントロールには，本の題名，著者，新品価格，古本の最低価格といった情報が含まれます。

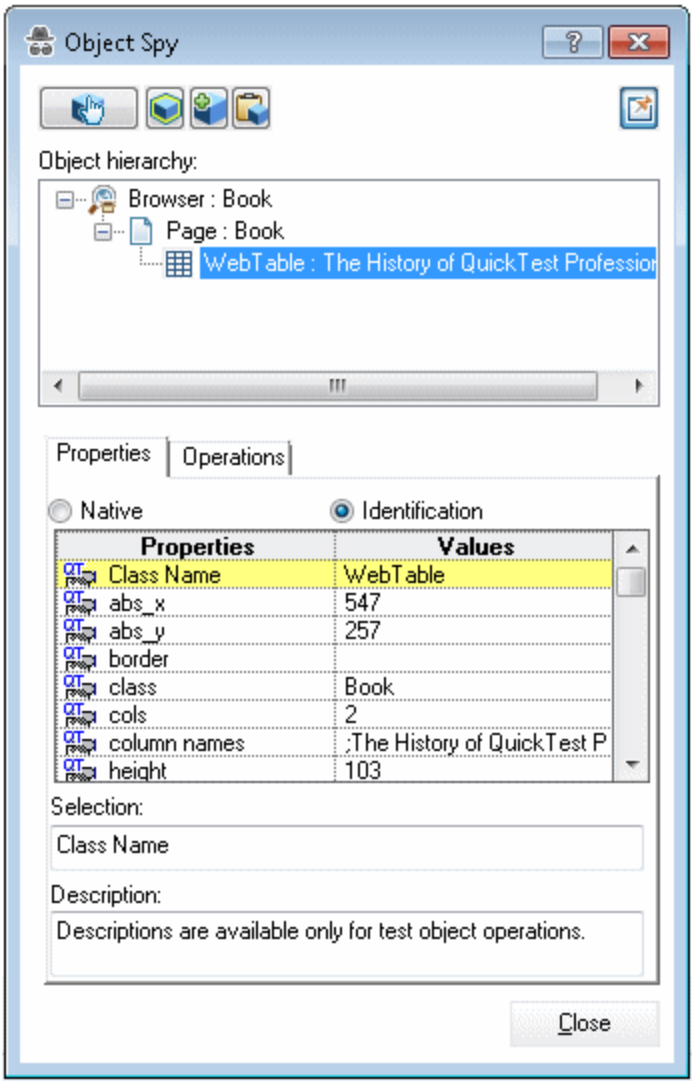
本の題名をクリックすると，詳細を表示するページが開きます。著者名をクリックすると，同じ著者の本のリストを表示するページが開きます。［**Used**］をクリックすると UsedBooks ページが開き，この本の入手可能なすべての古本とその価格のリストが表示されます。

Book コントロールは，Web テーブルを使用して次のように実装されています。

```
<table class="Book">
  <tr>
    <td class="BookImageCell" rowspan="4">
      <a href=".\QtpHistory.htm">
        <img class="BookImage" alt="Book" src=".\Res\Book.jpg"/>
      </a>
    </td>
    <td class="BookCell">
      <a class="BookTitle" href=".\QtpHistory.htm" > The History of QuickTest Professional </a>
    </td>
  </tr>
  <tr>
    <td class="BookCell">
        By:<a href=".\JaneDoe.htm">Jane Doe</a>,
 <a href=".\JohnDoe.htm">John Doe</a>
    </td>
  </tr>
  <tr>
    <td class="BookCell">
    </td>
  </tr>
  <tr>
    <td class="BookCell">
```

```
        New:<strong>59.99$</strong>   <a href=".\UsedBooks.htm">Used:</a> from
<strong>29.99$</strong>
      </td>
   </tr>
</table>
```

したがって，オブジェクト・スパイを使用してこのコントロールをポイントすると，UFT はこれを本の題名に基づく名前を持つ WebTable オブジェクトと認識します。テスト・オブジェクトに使用されるアイコンは，標準の WebTable クラスのアイコンです。



Book コントロールに対するサポートを実装せずに記録を実行すると，キーワード・ビューは次のようになります。

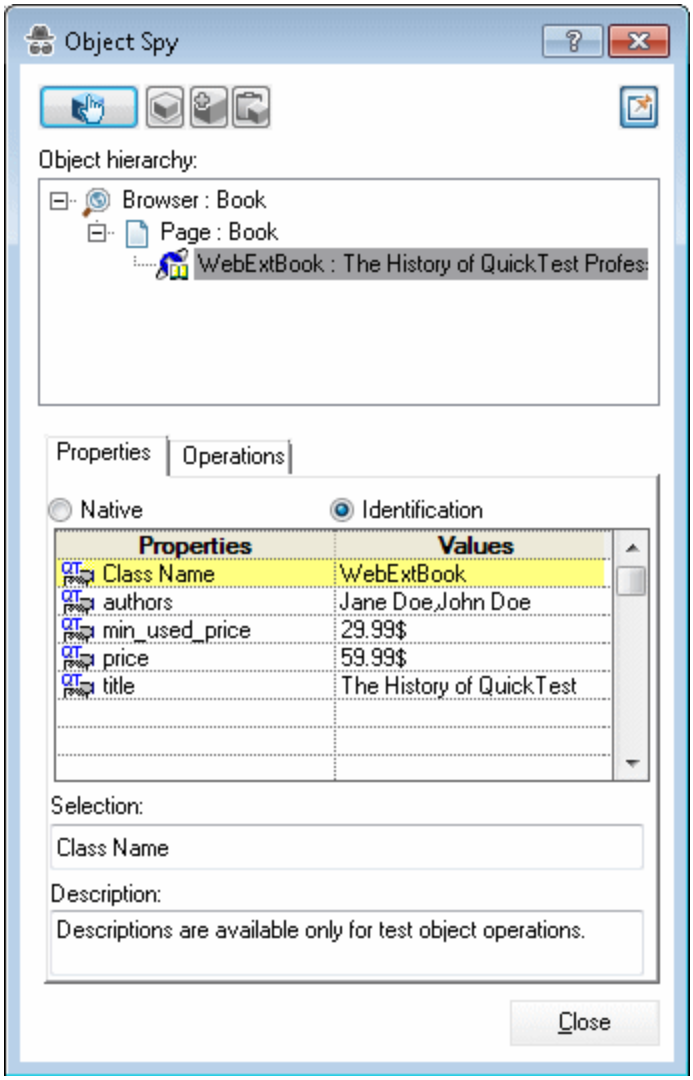
Item	Operation	Value	Documentation
Action1			
Book			
Book			
The History of Quick...	Click		Click the "The History of QuickTest" link.
Book	Click		Click the "Book" image.
Jane Doe	Click		Click the "Jane Doe" link.
Used:	Click		Click the "Used:" link.

エディタでは，記録されたテストは次のように表示されます。

```
Browser("Book").Page("Book").Link("The History of QuickTest").Click
Browser("Book").Page("Book").Image("Book").Click
Browser("Book").Page("Book").Link("Jane Doe").Click
Browser("Book").Page("Book").Link("Used:").Click
```

単純な **Click** ステップが記録されているだけで，各ステップは Book コントロール内部に定義された別のオブジェクトに対応付けられています。**Click** 操作は，異なる名前の Web Link テスト・オブジェクトまたは Book 画像テスト・オブジェクトに対して独立に記録されています。これらのステップは，このコントロールのコンテキストでは有用な意味を持ちません。

Web Add-in Extensibility を使用して Book コントロールをサポートすると，より意味のある結果が得られます。UFT はコントロールを **WebExtBook** テスト・オブジェクト（名前はやはり本の題名に基づく）として認識し，別のアイコンを使用します。認識プロパティには，**authors** や **min_used_price** といった関連情報が含まれます。これらは，本のすべての著者の名前と，古本の最低価格を表します。



コントロールに対するテストを作成する準備ができると，**Select**，**GoToAuthorPage**，**GoToUsedBookPage** メソッドがサポートされます。これらのメソッドは，記録することも，キーワード・ビューの［**操作**］カラムで手動で選択することもできます。テストを記録する際には，本の画像をクリックしても，題名をクリックしても，どちらも **Select** ステップが記録されます。

また，チェックポイントを作成して，**authors** （本のすべての著者から構成される文字列）などの認識プロパティの値をチェックすることもできます。

キーワード・ビューでは，上に示すテストと同じユーザ操作を記録して作成したテストは，次のようになります。

Item	Operation	Value	Documentation
Action1			
Book			
Book			
The History of QuickTest	Select		Select the "The History of QuickTest" book.
The History of QuickTest	Select		Select the "The History of QuickTest" book.
The History of QuickTest	GoToAuthorPage	"Jane Doe"	Open the Web page for "Jane Doe".
The History of QuickTest	GoToUsedBooksPage		Open the "The History of QuickTest" used books page.

エディタでは，テストは次のように表示されます。

```
Browser("Book").Page("Book").WebExtBook("The History of QuickTest").Check CheckPoint("The History of QuickTest")
Browser("Book").Page("Book").WebExtBook("The History of QuickTest").Select
Browser("Book").Page("Book").WebExtBook("The History of QuickTest").GoToAuthorPage "Jane Doe"
Browser("Book").Page("Book").WebExtBook("The History of QuickTest").GoToUsedBooksPage
```

このテストはより意味があり，Book コントロールの機能に即したものになっています。

Web Add-in Extensibility の実装方法

Web Add-in Extensibility を使用すると，コントロールに対するすべての UFT 機能の完全なサポートを実装できます。**ツールキット・サポート・セット**を開発することにより，コントロールのセット（**ツールキット**または**カスタム・ツールキット**と呼びます）に対する Web Add-in Extensibility サポートを実装できます。

Web Add-in Extensibility の実装は，次の段階で構成されます。

1. **ツールキット・サポート・セットの計画。**
   - カスタム・ツールキットを構成する Web コントロールのセットを決定します。
   - ツールキットによってサポートされるコントロールとビジネスプロセスに基づいてサポートするテスト・オブジェクトと操作を決定することにより，テスト・オブジェクト・モデルを定義します。
2. **ツールキット・サポート・セットの開発。**

   Web Add-in Extensibility ツールキット・サポート・セットは，次のファイルから構成されます。
   - 1 つの**テスト・オブジェクト設定**ファイル。これはツールキットに対して UFT が使用するテスト・オブジェクト・モデルを記述します。
   - 1 つの**ツールキット設定**ファイル。これは，ツールキット内の各コントロールを表すテスト・オブジェクト・クラスの指定と，UFT と各コントロールの間のやり取りの方法を記述します。
   - JavaScript 関数を含む 1 つまたは複数のファイル。UFT はこれらの関数を呼び出して，カスタム・コントロールに対する操作を実行します。

3. **ツールキット・サポート・セットのデプロイとテスト。**

   ツールキット・サポート・セットをデプロイして，UFT がコントロールをサポートできるようにするには，作成したファイルを UFT インストール・フォルダの特定の場所にコピーする必要があります。

   ツールキット・サポート・セットをデプロイした後で UFT を開くと，ツールキット名がアドイン・マネージャの［Web アドイン］ノードの下に子ノードとして表示されます。ツールキットのチェック・ボックスを選択すると，UFT は開発したツールキット・サポート・セットを使用してこのツールキットのコントロールをサポートします。

4. **ツールキット・サポート・セットの拡張。**

   コントロールに対する基本的な Web Add-in Extensibility サポートを作成してテストしたら，より複雑なオプションを使用してサポートを微調整することにより，ツールキット・サポート・セットを拡張できます。

各段階の詳細については，以下を参照してください。

- 「ツールキットに対する UFT サポートの計画」(24ページ)
- 「ツールキットのサポートの開発」(31ページ)
- 「ツールキット・サポート・セットのデプロイ」(74ページ)

Web Add-in Extensibility ツールキット・サポート・セットを開発する際には，まず単純で基本的なサポート・セットを作成して，UFT にデプロイする方法があります。これにより，UFT はコントロールを正しく認識できるようになり，UFT のユーザはコントロールに対するテストを作成して実行できるようになります。その後，より複雑な機能を実現するために，サポートを拡張します。たとえば，コントロールから学習された子オブジェクトのフィルタリングや，テストを記録する際のイベントのより高度な処理などです。

ツールキット・サポート・セットの開発方法をレッスンによって学ぶには，「チュートリアル：Web カスタム・ツールキットのサポートの作成方法の学習」(80ページ) のレッスンを実行してください。

Web Add-in Extensibility のサンプル

Web 2.0 Feature Pack for HP Functional Testing には，Web Add-in Extensibility についての学習に役立つ完成したツールキット・サポート・セットがいくつか付属しています。

- この機能パックから Web 2.0 Toolkit Support for Unified Functional Testing を UFT コンピュータにインストールすると，Web アドインの下に，いくつかの公開 Web 2.0 ツールキット（ASP.NET AJAX，GWT，YahooUI，Dojo）をサポートする子アドインがインストールされます。これらのアドインは，Web Add-in Extensibility を使用して開発されたツールキット・サポート・セットです。これらのツールキット・サポート・セットを構成するファイルは，**<UFT インストール・フォルダ>\dat\Extensibility\Web** フォルダと，このフォルダ内の **Toolkits** サブフォルダにあります。
- Extensibility Accelerator をインストールすると，上記の Web 2.0 ツールキット・サポート・セットを含むサンプル Web Add-in Extensibility プロジェクトがインストールされます。

  サンプル・プロジェクトは，**%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples** フォルダにインストールされており，Extensibility Accelerator 開始ページか

らもアクセスできます。これらのプロジェクトを Extensibility Accelerator で開いて，ファイル，関数，コメントを調べることで，サポート・セットの設計について詳しく知ることができます。また，サンプル・プロジェクトを変更して実験してみることもできます。サンプル・プロジェクトのバックアップ・コピーが，**<Extensibility Accelerator インストール・フォルダ>\Help\Samples** フォルダにインストールされています。

- Extensibility Accelerator をインストールすると，サンプル Book アプリケーションと，このアプリケーション用のツールキット・サポート・セットを含む Web Add-in Extensibility プロジェクトもインストールされます。
  - アプリケーションは，**%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample\Application\Book.htm** にインストールされます。
  - Book アプリケーション用のツールキット・サポート・セットを含む Web Add-in Extensibility プロジェクトは，**%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample** にインストールされます。

  「チュートリアル：Web カスタム・ツールキットのサポートの作成方法の学習」(80ページ) では，Book アプリケーション用のこれと同じようなツールキット・サポート・セットを作成して，UFT に（Extensibility Accelerator を使用せずに）デプロイします。

  『HP UFT Extensibility Accelerator for HP Functional Testing ユーザーズ・ガイド』 には，Extensibility Accelerator を使用してこのコントロールに対するサポートを作成するチュートリアルが含まれています。

第2章: ツールキットに対する UFT サポートの計画

カスタム・ツールキットのサポートの作成を始める前に，サポートを注意深く計画する必要があります。UFT にカスタム・コントロールをどのように認識させるかを詳細に計画し，カスタム・ツールキット・サポートの基本要素を正しく構築します。

本章の内容

- ツールキットに対する UFT サポートの計画について 25
- カスタム・ツールキットのサポートの作成の準備 25
- ツールキット関連情報の決定 26
- 各カスタム・コントロール・タイプに対するサポート情報の決定 26
- その他の情報 30

注: 本章は，「UFT Web Add-in Extensibility の概要」(12ページ)に記載されている概念について理解していることが前提となっています。

ツールキットに対するUFTサポートの計画について

カスタム・コントロールを認識する UFT の Web アドインのサポートを拡張するプロセスには，詳細な計画が必要です。この作業に役立つように，この章の各項には，カスタム・ツールキットおよびそのコントロールのサポートの実装に関連するいくつかの確認事項が記されています。ツールキット・サポート・セットを作成する際には，これらの質問に対する答えに基づいて実装を行います。

最初に，カスタム・ツールキットに関する一般的な情報を決定し，その後に，サポートする各コントロールに関する具体的な情報を定義します。

カスタム・ツールキットのサポートの作成の準備

カスタム Web コントロールのサポートの計画を開始する前に，コントロールに対する完全なアクセスが可能であることと，コントロールの動作を十分に理解していることを確認します。コントロールの実際の動作を確認できるアプリケーションまたは Web ページが利用でき，コントロールを実装しているソースを参照できることが必要です。

UFT でカスタム・コントロールをサポートするためにコントロールのソースを変更する必要はありませんが，ソースを理解しておく必要はあります。コントロールにどのような要素と属性が含まれ，どのような HTML プロパティが存在し，どのようなイベントをリッスンできるかといったことを知っておく必要があります。

ツールキット関連情報の決定

ツールキット・サポート・セットを計画する際には，まずツールキット関連の一般的情報を決定します。

- サポートを作成するツールキットまたは環境に一意の名前を付けます。

  UFT は，アドインまたはサポートされる環境のリストを表示するすべてのダイアログ・ボックスに，環境の名前を表示します。たとえば，UFT が開いたときには，［アドイン マネージャ］ダイアログ・ボックスに Web Add-in の子として環境の名前が表示され，UFT ユーザはその環境のサポートをロードするかどうかを指定できます。

- このツールキット・サポート・セットがサポートするコントロールを決定します。
- ツールキット・サポート・セット用に作成する JavaScript 関数を保存するファイルを決定します。
  - 複数の UFT 機能と複数のテスト・オブジェクト・クラスに対するサポートを実装する JavaScript 関数のための標準設定のファイルを 1 つ指定できます。これに加えて，各機能やテスト・オブジェクト・クラスに対する関数を実装する個別のファイルも定義できます。
  - ほかの関数から呼び出す共通の JavaScript 関数を含むファイルを 1 つ指定できます。
- テスト・オブジェクト・クラスをカスタム・コントロールに対応付けるためにツールキット全体で 1 つの JavaScript 関数を使用するかどうかを決定します。詳細については，「カスタム Web コントロールに対して使用するテスト・オブジェクト・クラスを UFT に指示する方法」(51ページ)を参照してください。

ツールキット・サポート・セットを設計する際には，この情報をツールキット設定ファイルに指定します。詳細については，「ツールキット設定ファイルについて」(42ページ)を参照してください。

各カスタム・コントロール・タイプに対するサポート情報の決定

特定のタイプのコントロールのカスタム・サポートを計画する際には，UFT でコントロールを認識する方法（UFT GUI テストでコントロールを表すテスト・オブジェクトのタイプ，使用する認識プロパティとテスト・オブジェクト・メソッドなど）をよく検討する必要があります。このような決定は，このタイプのコントロールを使用してテストされる可能性があるビジネス・プロセスと，ユーザがこれらのコントロールに対して実行すると想定される操作に基づいて行います。

カスタム・コントロールを含むアプリケーションを実行し，オブジェクト・スパイ，キーワード・ビュー，記録オプションを使用して，UFT の観点からコントロールを分析することができます。カスタム・サポートがない状態で UFT がコントロールを認識する方法を把握できるので，どのような変更が必要なのかを判断できます。

注: Web Add-in Extensibility を使用すると，Web ページおよびフレーム内の Web コントロールに対するサポートを作成できます。Web ページまたはフレーム自身に対するカスタム・サポート

は開発できません。

UFT を使用してカスタム・コントロールを分析する例については，「サンプル・カスタム・コントロールに対する標準設定の UFT サポートと Extensibility オプションの分析」(17ページ)を参照してください。

本項の内容

- 「Web Add-in Extensibility 計画の理解チェックリスト」(27ページ)
- 「Web Add-in Extensibility の計画チェックリスト」(29ページ)

Web Add-in Extensibility 計画の理解チェックリスト

特定のタイプのコントロールのサポートを計画する際には，いくつかの質問に対する答えを考える必要があります。ここでは，これらの確認事項について説明します。また，確認事項を簡単にまとめた印刷形式の「チェックリスト」（29 ページ）も利用できます。

1. UFT がインストールされたコンピュータ上でカスタム・コントロールを実行するアプリケーションが使用できることを確認します。
2. 拡張することでカスタム・コントロールを表ることができる既存の Web テスト・オブジェクトがあるか？ある場合，それはどれか？ない場合，新しいテスト・オブジェクト・クラスが WebElement クラスを拡張する必要があります。
3. 新しいテスト・オブジェクト・クラスが WebElement 以外の基本テスト・オブジェクト・クラスを拡張する場合，コントロールには通常基本テスト・オブジェクト・クラスで表されるタイプの要素（基本要素とも呼ぶ）が含まれているか？
   - 含まれていない場合，基本クラスから継承されたテスト・オブジェクト・メソッドのうち，WebElement でサポートされないものを実装する必要があります。さらに，テスト・クラスの認識プロパティのうち，WebElement でサポートされないものすべての値を返すメソッドを設計する必要があります。
   - コントロールに基本要素が含まれる場合，それはコントロールのルート Web 要素か？
     - ルート要素の場合，継承されたテスト・オブジェクト・メソッドと認識プロパティのうち，ユーザがオーバーライドしないものに対しては，UFT はその内部実装を使用します。
     - 基本要素がルート要素でない場合，基本要素を返す JavaScript 関数を実装する必要があります。
4. UFT GUI テストでカスタム・コントロールを表す新しいテスト・オブジェクト・クラスの詳細を定義します。

   ツールキット・サポート・セットを設計する際には，この情報をテスト・オブジェクト設定ファイルに指定します。詳細については，「テスト・オブジェクト設定ファイルについて」(34ページ)を参照してください。

   a. テスト・オブジェクト・クラス名を指定します。
   b. 新しいテスト・オブジェクト・クラスが，基本クラスが属するのとは異なる汎用タイプに属するようにするか？そうする場合は，汎用タイプを指定します（たとえば，新しいテスト・

オブジェクト・クラスが WebElement（汎用タイプは **object**）を拡張しているが，このテスト・オブジェクト・クラスを **edit** テスト・オブジェクト・クラスと同じグループに分類させる場合）。

c. 新しいテスト・オブジェクトに対して UFT で異なるアイコンを使用するか？

使用する場合，アイコン・ファイルは非圧縮の **.ico** 形式である必要があります。推奨される場所：**<UFT インストール・フォルダ>\dat\Extensibility\<UFT アドイン名>\Toolkits\<環境名>\Res**

d. コントロールを一意に識別するために使用できる 1 つまたは複数の認識プロパティを（テスト・オブジェクト・クラスに加えて）指定します。

e. このクラスのテスト・オブジェクトに対してステップが生成されたときにキーワード・ビューとステップ・ジェネレータに表示される標準設定のテスト・オブジェクト・メソッドを指定します。

f. キーワード・ビューまたはエディタでこのクラスのテスト・オブジェクトに対して **F1** キーが押されたときに UFT が開くヘルプ・ファイルを作成するか？

作成する場合は，ヘルプ・ファイルが **.chm** 形式であることを確認します。推奨される場所：**<UFT インストール・フォルダ>\dat\Extensibility\<UFT アドイン名>\Toolkits\<環境名>\Help**

5. このコントロールに対して使用するテスト・オブジェクト・クラスを識別するプロセスの設計方法を決定します。

- 特定の HTML タグを持つ HTML 要素のみにプロセスを限定するか？そうする場合は，それはどのタグか？カスタム・コントロールに関連するすべての HTML タグを含める必要があります。
- このタイプのコントロールを表すテスト・オブジェクト・クラスを決定するために使用する HTML プロパティは何か？
- コントロールを異なるブラウザで実行するときに，異なる認識ルールを作成する必要があるか？

詳細については，「カスタム Web コントロールに対して使用するテスト・オブジェクト・クラスを UFT に指示する方法」(51ページ)を参照してください。

6. コントロールを表すテスト・オブジェクトの名前を付けるための基礎を指定します。
7. サポートする認識プロパティはどれか？UFT の［チェックポイントのプロパティ］および［出力値のプロパティ］ダイアログ・ボックスに表示するプロパティはどれか？また，このダイアログ・ボックスで標準設定で選択されるプロパティはどれか？スマート認識に使用できる認識プロパティはどれか？
8. サポートするテスト・オブジェクト・メソッドはどれか？メソッドの引数タイプと名前，およびメソッドがリターン・コードに加えて値を返すかどうかを指定します。

オプションで，キーワード・ビューまたはエディタで **F1** キーが押されたとき，またはステップ・ジェネレータでテスト・オブジェクト・メソッドに対して［操作ヘルプ］ボタンがクリックされたときに UFT が開くヘルプ・ファイルの場所を指定します。

9. テスト・オブジェクト・メソッドのいくつかの引数に対して可能な値のリストを動的に提供するか？提供する場合，どの引数か？

10. UFT がコントロールから学習するのはどのタイプの子か？
11. オブジェクト・スパイにこのクラスのテスト・オブジェクトを表示させるか？
12. 記録による UFT GUI テストの作成のサポートを提供するか？

    提供する場合，UFT 記録セッション中にカスタム・コントロールに対して記録するイベントのリストを指定します。
13. サポートのどの部分をツールキット設定ファイルで設計し，どの部分に JavaScript 関数が必要かを決定します。

Web Add-in Extensibility の計画チェックリスト

このチェックリストは，カスタム・コントロールのサポートを計画するために使用します。

☑	カスタム・コントロールのサポート計画チェックリスト	ツールキット XML で指定	JavaScript 関数でサポート
❑	このカスタム・コントロールのソースの場所：	該当なし	該当なし
❑	新しいテスト・オブジェクト・クラスが拡張する Web テスト・オブジェクト基本クラスを指定：（標準設定—WebElement）	該当なし	該当なし
❑	基本テスト・オブジェクト・クラスは WebElement か？ **はい/いいえ** 「いいえ」の場合，基本要素（基本テスト・オブジェクト・クラスに対応する要素）があるか？ **はい/いいえ** 基本要素がある場合，それを返す JavaScript 関数が必要か？ **はい/いいえ**	**はい/いいえ**	**はい/いいえ**
❑	新しい Web テスト・オブジェクト・クラスの詳細を指定： • テスト・オブジェクト・クラス名： • 汎用タイプ（オプション）： • アイコン・ファイルの場所（任意）： • 記述のための認識プロパティ： • 標準設定のテスト・オブジェクト・メソッド： • ヘルプ・ファイルの場所：	該当なし	該当なし
❑	コントロールに使用するテスト・オブジェクト・クラスを識別するための基準を指定（異なるブラウザを考慮）：	**はい/いいえ**	**はい/いいえ**
❑	テスト・オブジェクトの名前付けの基礎を指定：	該当なし	**はい**
❑	サポートする認識プロパティのリスト。チェックポイントと出力値に使用可能なもの（およびチェックポイントで標準設定で選択されるもの）と，スマート認識に使用されるもの（ある場合）をマー	**はい/いいえ**	**はい/いいえ**

☑	カスタム・コントロールのサポート計画チェックリスト	ツールキット XML で指定	JavaScript 関数でサポート
	ク。		
❑	サポートするテスト・オブジェクト・メソッドのリスト（必要な場合，引数，戻り値，ヘルプ・ファイルの場所，ヘルプ ID を指定）：	**はい/いいえ**	**はい/いいえ**
❑	テスト・オブジェクト・メソッド引数に対して動的な値のリストを指定するか？ **はい/いいえ**（標準設定） 指定する場合，引数をリスト：	該当なし	**はい/いいえ**
❑	UFT がコントロールで学習する子のタイプを指定：	**はい/いいえ**	**はい/いいえ**
❑	オブジェクト・スパイにこのクラスのテスト・オブジェクトを表示させるか？ **はい**（標準設定）**/いいえ**	**はい/いいえ**	該当なし
❑	記録をサポートするか？ **はい/いいえ** その場合，記録をトリガするイベントを列挙：	**はい/いいえ**	**はい/いいえ**

その他の情報

カスタム・ツールキット・サポートの計画が終了したら，計画に基づいて，カスタム・ツールキットをサポートするツールキット・サポート・セットを作成します。ツールキット・サポート・セットの開発方法は，「ツールキットのサポートの開発」(31ページ) で説明しています。

第3章: ツールキットのサポートの開発

本章では，カスタム Web ツールキットに対するサポートの作成方法を説明します。ツールキット・サポート・セット用に作成するファイル，これらのファイルの構造と内容，UFT のさまざまな機能を環境に対してサポートするためのファイルの使用方法を説明します。

設計したサポートをアクティブにするためにツールキット・サポート・セットのファイルを置く場所については，「ツールキット・サポート・セットのデプロイ」(74ページ) を参照してください。

本章の内容

- カスタム・ツールキットのサポートについて 32
- カスタム・ツールキット・サポート・セットの作成 33
- テスト・オブジェクト設定ファイルについて 34
- ツールキット設定ファイルについて 42
- ツールキット・サポート・セット用の JavaScript 関数の設計 44
- カスタム Web コントロールに対して使用するテスト・オブジェクト・クラスを UFT に指示する方法 51
- ツールキット・サポート・セットの開発中のテスト 57
- カスタム・サポートのログとデバッグ 60
- テスト・オブジェクト・メソッドのサポートの実装 63
- 認識プロパティのサポートの実装 67
- 子コントロールの学習のためのフィルタの実装 69
- 記録のサポートの実装 70
- トラブルシューティングと制限事項 - サポートの開発 73

カスタム・ツールキットのサポートについて

Web Add-in Extensibility を実装するには，サポートする各 Web ツールキットに対して，**ツールキット・サポート・セット**を作成します。ツールキット・サポート・セットは，XML 設定ファイルと JavaScript 関数から構成されます。XML 設定ファイルは，カスタム Web コントロールをサポートするために作成するテスト・オブジェクト・クラスを定義し，コントロールにマップします。また，UFT がカスタム・コントロールに対してどのように動作するかも定義します。JavaScript 関数は，UFT とテスト対象のアプリケーションとの間のインタフェースの役割を果たし，コントロールに関する情報を取得したり，コントロールに対する操作を実行したりします。

本章では，カスタム・サポート・セットに必要なさまざまなファイル，定義，関数について説明します。詳細については，『Toolkit Configuration Schema Help』，『API Reference』，『JavaScript Function Reference』，『UFT Test Object Schema Help』（『UFT Web Add-in Extensibility ヘルプ』から利用可能）を参照してください。

Extensibility Accelerator で提供されるサンプル・プロジェクトには，いくつかの公開 Web 2.0 ツールキットに対する UFT のサポートを拡張するツールキット・サポート・セットが含まれています。これらは，ASPAjax，GWT，YahooUI，Dojo に対するサポート・セットであり，Web 2.0 Toolkit Support for Unified Functional Testing をインストールしたときに UFT にインストールされます。サンプルには，これらのツールキットからのコントロールのテストをサポートするために UFT が使用するテスト・オブジェクト・モデルに関するヘルプ・ファイルも含まれます。

サンプル・プロジェクトは **%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples** にインストールされます。プロジェクトを Extensibility Accelerator で開き，ツールキット・サポート・セットのファイルを調べることで，Web Add-in Extensibility の実装方法の例を見ることができます。この章の例のいくつかは，これらのファイルから取られています。

注: ツールキット・サポート・セットの作成を実際に始める前に，注意深く計画を立てる必要があります。詳細については，「ツールキットに対する UFT サポートの計画」(24ページ)を参照してください。

ブラウザに依存しないサポートの開発

Web コントロールの中には，ブラウザによって実装や動作が異なるものが数多くあります。これは，さまざまな DOM プロパティの実装がブラウザによって異なる場合があるからです。

カスタム・サポートが Microsoft Internet Explorer，Google Chrome，Mozilla Firefox で動作するようにするには，次のことを行う必要があります。

- コントロールの DOM 構造がブラウザによって異なる場合，オブジェクトの学習またはステップの実行の際に，UFT はコントロールに対して使用するオブジェクト・クラスを識別するためにブラウザごとに異なるロジックを使用しなければならない可能性があります。ブラウザ固有の識別条件の定義方法については，「Conditions 要素の使用」(52ページ) を参照してください。
- JavaScript 関数を作成する際には，jQuery 関数呼び出しを使用して，さまざまなタイプとバージョンのブラウザでスムーズに動作するコードを作成します。詳細については，「ブラウザに依存しないサポートの作成」(45ページ)を参照してください。
- Web Add-in Extensibility に用意されている **GetBrowserType** および **GetBrowserVersion** ユーティリティ・メソッドを使用すると，現在コントロールを実行しているブラウザのタイプとバージョンを知ることができます。これにより，JavaScript コードは，現在コントロールを実行しているブラウザに応じて異なる処理を，コントロールに対して実行できます。

カスタム・ツールキット・サポート・セットの作成

Web Add-in Extensibility ツールキット・サポート・セットは，次のファイルから構成されます。

- 1 つの**テスト・オブジェクト設定**ファイル。これはツールキットのテスト・オブジェクト・モデルを記述します。これには，ツールキット内のコントロールを表すために UFT が使用するテスト・オブジェクト・クラスと，そのテスト・オブジェクトに対してサポートが必要な認識プロパティおよびテスト・オブジェクト・メソッドが含まれます。詳細については，「テスト・オブジェクト設定ファイルについて」(34ページ)を参照してください。

  注: UFT が開いたときに，すべてのテスト・オブジェクト・クラス定義が（すべてのテスト・オブジェクト設定ファイルから），対象となるカスタム・ツールキットに関わらずロードされます。このため，複数のカスタム・ツールキットのサポートに同じテスト・オブジェクト・クラス定義を使用できます。

- 1 つの**ツールキット設定**ファイル。これは，ツールキット内の各コントロールを表すテスト・オブジェクト・クラスの指定と，UFT と各コントロールの間のやり取りの方法を記述します。詳細については，「ツールキット設定ファイルについて」(42ページ)を参照してください。
- JavaScript 関数を含む 1 つまたは複数のファイル。UFT はこれらの関数を呼び出して，カスタム・コントロールから情報を取得したり，カスタム・コントロールに対する操作を実行したりします。
- オプションで，定義するテスト・オブジェクト・クラスを表すために UFT で使用されるアイコンを含むアイコン・ファイルと，これらのテスト・オブジェクト・クラスとそのメソッドおよびプロパティの説明を含むヘルプ・ファイル。

ツールキット・サポート・セットに必要なファイルの名前と，ファイルが保存されるフォルダの構造は，「ツールキット・サポート・セットのデプロイ」(74ページ)に記されています。

カスタム・ツールキット・サポート・セットを作成するには，次の手順を実行します。

1. サポートを作成するツールキットまたは環境を表す一意の名前を選択します。

   カスタム・ツールキット名は，ツールキット・フォルダとツールキット設定ファイルの名前を作成するために使用されます。名前の先頭は英字で，英数字とアンダースコアだけが使用できます。

   一意のツールキット名を指定することで，1 つの UFT で多数のカスタム・ツールキット・サポート・セットを同時にサポートすることが可能になります。そのため，MyToolkit といった名前は推奨されません。

2. ツールキット・サポート・セットのフォルダを作成します。

   このフォルダの名前と場所は任意に決めることができます。

3. 次のフォルダ構造を作成します。
   - **<ツールキット・サポート・セット・フォルダ>\Toolkits\<ツールキット環境名>\JavaScript**
   - **<ツールキット・サポート・セット・フォルダ>\Toolkits\<ツールキット環境名>\Res**
   - **<ツールキット・サポート・セット・フォルダ>\Toolkits\<ツールキット環境名>\Help**

4. ツールキット・サポート・セット・フォルダの下に，**<ツールキット環境名>TestObjects.xml** というファイルを作成します。これはテスト・オブジェクト設定ファイルです。

5. **<ツールキット・サポート・セット・フォルダ>\Toolkits\<ツールキット環境名>** フォルダの下に，**<ツールキット環境名>.xml** というファイルを作成します。これはツールキット設定ファイルです。

6. **<ツールキット・サポート・セット・フォルダ>\Toolkits\<ツールキット環境名>\JavaScript** フォルダの下に，設計する JavaScript 関数を含める 1 つまたは複数のファイルを作成します。

   推奨される方法としては，各テスト・オブジェクト・クラスに対して JavaScript ファイルを 1 つ作成します。その他に，ほかのファイル内の JavaScript 関数から呼び出される JavaScript 関数を含める JavaScript ファイルを 1 つ作成することもできます。

7. **<ツールキット・サポート・セット・フォルダ>\Toolkits\<ツールキット環境名>\Res** フォルダには，UFT でテスト・オブジェクト・クラスを表すために使用するアイコンを置くことができます。

8. **<ツールキット・サポート・セット・フォルダ>\Toolkits\<ツールキット環境名>\Help** フォルダには，環境内のテスト・オブジェクトを記述するヘルプ（**.chm**）ファイルを置くことができます。

テスト・オブジェクト設定ファイルについて

カスタム・ツールキットのサポート開発の第 1 段階では，アプリケーションとコントロールのテストに UFT が使用するテスト・オブジェクト・モデルを導入します。このためには，テスト・オブジェクト設定 XML ファイルでテスト・オブジェクト・モデルを定義します。UFT サポートを拡張または変更するカスタム・コントロールのすべてのタイプに対して，テスト・オブジェクト・クラスを作成する必要があります。

テスト・オブジェクトの設定 XML では，テスト・オブジェクト・クラス（サポートされるテスト・オブジェクト・メソッド，認識プロパティなど）を定義します。

定義するテスト・オブジェクト・クラスごとに **ClassInfo** 要素を作成します。さらに，テスト・オブジェクト・クラスを使用する環境やカスタム・ツールキットの名前（**TypeInformation** 要素の **PackageName** 属性）と，このテスト・オブジェクト・クラスが拡張する対象となる UFT アドイン（**TypeInformation** 要素の **AddinName** 属性）を定義します。

UFT が開いたときに該当するアドインがロードされていない場合は，UFT はこの XML の情報をロードしません。同様に，環境またはカスタム・ツールキットの名前が［アドイン マネージャ］ダイアログ・ボックスに表示されていて，対応するチェック・ボックスが選択されていない場合，この XML の情報はロードされません。

詳細については，「UFT がテスト・オブジェクト設定 XML をロードする方法」(38ページ)を参照してください。

次の項では，テスト・オブジェクト・クラスで定義できる内容について説明します。

クラス名と基本クラス

新しいテスト・オブジェクトの名前と属性です。これには，基本クラスも含まれます。基本クラスとは，新しいテスト・オブジェクト・クラスが拡張するスト・オブジェクト・クラスです。新しいテスト・オブジェクト・クラスを定義することにより，既存の WebUFT テスト・オブジェクト・クラスを直接的または間接的に拡張できます。基本クラスには UFT で提供されているクラスのほかに，Web Add-in Extensibility を使用して定義したクラスも含まれます。

標準設定では，基本クラスは WebElement です。

テスト・オブジェクト・クラス名は，UFT ユーザがサポートを同時にロードする可能性があるすべての環境の間で一意である必要があります。たとえば，新しいテスト・オブジェクト・クラスを定義する場合，既存の UFT アドインのテスト・オブジェクト・クラスの名前（WebButton，WebEdit， など）は使用しないでください。

注:

- テスト・オブジェクト・クラスは，基本クラスのテスト・オブジェクト操作（メソッドとプロパティ），汎用タイプ，標準設定操作，アイコンを継承します。ただし認識プロパティは，継承されません継承されます。
- 他のツールキット・サポート・セットで定義されたテスト・オブジェクト・クラスを拡張するテスト・オブジェクト・クラスを作成した場合，2 つのツールキット・サポート・セットの間に依存関係が生じます。拡張元となるツールキット・サポート・セットを UFT アドイン・マネージャにロードする場合には，拡張対象となるツールキット・サポート・セットもロードする必要があります。

汎用タイプ

新しいテスト・オブジェクト・クラスの汎用タイプとして基本クラスとは異なるタイプを選択したい場合に，新しいテスト・オブジェクト・クラスに指定する汎用タイプです（たとえば，新しいテスト・オブジェクト・クラスが WebElement（汎用タイプは **object**）を拡張する場合に，UFT ではこの

テスト・オブジェクト・クラスを **edit** として分類したいケースなど)。

汎用タイプは，オブジェクトのフィルタ処理（ステップ・ジェネレータの［ステップでオブジェクトを選択］ダイアログ・ボックスでの操作や，複数のテスト・オブジェクトをオブジェクト・リポジトリに追加する操作など）で使用します。また，キーワード・ビューの［注釈］カラムで使用する文字列を作成する際にも使用します（テスト・オブジェクト設定ファイルで別途定義されていない場合）。

テスト・オブジェクト操作

テスト・オブジェクト・クラスに対する操作のリスト。各操作に関する次の情報を含みます。

- 引数とそのタイプ（String，Integerなど），方向（InまたはOut），引数が必須かどうか，必須でない場合の標準設定値。
- この引数の可能な値の動的なリストを，サポートされるコントロールから取得して，キーワード・ビュー，エディタ，ステップ・ジェネレータに表示できるかどうか。
- 操作の説明（オブジェクト・スパイと，キーワード・ビューおよびステップ・ジェネレータのツールヒントで表示）。
- 注釈の文字列（キーワード・ビューの［**注釈**］カラムと，ステップ・ジェネレータに表示）。
- 戻り値のタイプ。
- キーワード・ビューまたはエディタでテスト・オブジェクト操作に対して **F1** キーが押されたとき，またはステップ・ジェネレータで操作に対して［**操作ヘルプ**］ボタンがクリックされたときに表示されるコンテキスト・ヘルプ・トピック。ヘルプ・ファイルのパスとファイル内の該当するヘルプ ID を定義します。

標準設定操作

このクラスのオブジェクトに対してステップが生成されたときに，キーワード・ビューとステップ・ジェネレータで標準設定で選択されるテスト・オブジェクト操作。

認識プロパティ

テスト・オブジェクト・クラスの認識プロパティのリストです。また，次の情報も定義できます。

- オブジェクト記述で使用する認識プロパティ。

  ヒント: オブジェクト記述に **html tag** プロパティを含めておくと，UFT でオブジェクトの認識にかかる時間を短縮できます。

- **スマート認識**で使用する認識プロパティ（この情報が使用されるのは，テスト・オブジェクト・クラスでスマート認識が有効になっている場合のみです。スマート認識を有効にするには，UFT の［オブジェクトの認識］ダイアログ・ボックスを使用します）。
- チェックポイントと出力値で使用できる認識プロパティ。
- チェックポイントに対して標準設定で選択される認識プロパティ（UFT の［チェックポイントのプロパティ］ダイアログ・ボックス）。

アイコン・ファイル

このテスト・オブジェクト・クラスに使用するアイコン・ファイルのパス（任意。定義しない場合は，基本クラスのアイコンが使用されます）。使用できるファイルの種類は，**.dll**，**.exe**，**.ico** です。

ヘルプ・ファイル

キーワード・ビューまたはエディタでテスト・オブジェクトに対して **F1** キーが押されたときに表示されるコンテキスト・ヘルプ・トピック。設定ファイルでは，**.chm** ヘルプ・ファイルのパスとそのファイル内の該当するヘルプ ID を定義します。

テスト・オブジェクト設定 XML の構造と構文については，『UFT Test Object Schema Help』（『UFT Web Add-in Extensibility ヘルプ』から利用可能）を参照してください。

テスト・オブジェクト設定 XML の例

次の例は，このガイドのチュートリアル・セクションで使用される **WebExtSample** ツールキットの一部である **WebExtUsedBooks** テスト・オブジェクトの定義を示します。

```
<ClassInfo BaseClassInfoName="WebTable" GenericTypeID="Table"
    DefaultOperationName="SelectBook" Name="WebExtUsedBooks">
   <IconInfo
      IconFile="INSTALLDIR\Dat\Extensibility\Web\Toolkits\
   WebExtSample\Res\WebBookList.ico"/>
   <TypeInfo>
      <Operation ExposureLevel="CommonUsed" Name="SelectBook" PropertyType="Method">
         <Description>
            Selects the radio button for the specified book and clicks Select.
         </Description>
         <Documentation>
            <![CDATA[Select the radio button for the book with index %a1 and click Select.]]>
         </Documentation>
         <Argument Name="BookIndex" IsMandatory="true" Direction="In">
            <Type VariantType="Integer"/>
         </Argument>
      </Operation>
   </TypeInfo>
   <IdentificationProperties>
      <IdentificationProperty ForDefaultVerification="true" ForVerification="true"
ForDescription="true" Name="title"/>
   </IdentificationProperties>
</ClassInfo>
```

この例は，**WebExtUsedBooks** テスト・オブジェクト・クラスが **WebTable** テスト・オブジェクト・クラスを拡張していることを示しています。**WebExtUsedBooks** テスト・オブジェクト・クラスは，**WebBookList.ico** アイコン・ファイルを使用し，標準設定のテスト・オブジェクト・メソッドは

SelectBook（Integer タイプの必須入力パラメータ**BookIndex** を持つ）で，追加の認識プロパティ **title** を持ちます。

WebExtUsedBooks テスト・オブジェクト・クラスに対しては，認識プロパティ **title** が定義されています。この認識プロパティは，オブジェクトの記述に用いられ，チェックポイントと出力値に使用でき，UFT の［チェックポイントのプロパティ］ダイアログ・ボックスで標準設定で選択されます。

その他の参照項目：

- 「UFT がテスト・オブジェクト設定 XML をロードする方法」(38ページ)
- 「既存のテスト・オブジェクト・クラスの拡張」(40ページ)
- 「カスタマイズしたテスト・オブジェクト・クラスに対するヘルプ・ファイルの作成」(41ページ)

UFT がテスト・オブジェクト設定 XML をロードする方法

UFT が実行されると，すべてのテスト・オブジェクト設定 XML が読み込まれ，さまざまな XML に記述された各テスト・オブジェクト・クラスの情報が，1 つのテスト・オブジェクト・クラス定義に結合されます。詳細については，「UFT がテスト・オブジェクト設定ファイルを結合する方法」(38ページ)を参照してください。

テスト・オブジェクト設定ファイル中の**IdentificationProperty**要素には，UFT で（［オブジェクトの認識］ダイアログ・ボックスを使用して）変更される可能性がある情報を指定する属性があります。その属性とは，**AssistivePropertyValue**，**ForAssistive**，**ForBaseSmartID**，**ForDescription**，**ForOptionalSmartID**，**OptionalSmartIDPropertyValue** です。この属性に基づいて，UFT で各種用途に使用する認識プロパティのリストが決まります。

したがって標準設定では，UFT は属性値を XML ファイルから 1 度だけ読み込みます。これにより，［オブジェクトの認識］ダイアログ・ボックスでユーザが行った変更内容が上書きされることはなくなり，UFT では，このようにして，ユーザ定義のプロパティ・リストが変更されないようにしています。詳細については，「テスト・オブジェクト設定ファイルで指定された認識プロパティ属性の変更」(78ページ)を参照してください。

UFT がテスト・オブジェクト設定ファイルを結合する方法

UFT は起動のたびに，**<UFT インストール・フォルダ>\dat\Extensibility\<UFT アドイン名>** フォルダにあるテスト・オブジェクト設定ファイルをすべて読み込みます。次に，テスト・オブジェクト設定ファイルのそれぞれの優先度に応じて，各ファイルのテスト・オブジェクト・クラスの情報を 1 つのテスト・オブジェクト・クラス定義にマージします。

次の場合には，テスト・オブジェクト設定ファイル内の定義は無視されます。

- **TypeInformation** 要素の **Load** 属性が false に設定されている場合。
- テスト・オブジェクト設定ファイルに関連する環境は［アドイン マネージャ］ダイアログ・ボックスに表示されているが，UFT ユーザが環境をロードしない選択を行っている場合。

テスト・オブジェクト設定ファイルの優先度を，**TypeInformation** 要素の **Priority** 属性で定義します。

テスト・オブジェクト設定ファイルの優先度が既存のクラス定義よりも高い場合，既存のテスト・オブジェクト・クラス定義は，UFT の組み込みの情報を含めて，オーバーライドされます。したがって，テスト・オブジェクト設定ファイルの優先度を変更する場合には，組み込み関数がオーバーライドされる可能性がある点に注意してください。

複数のテスト・オブジェクト・クラス定義が存在する場合，UFT は競合を解決する必要があります。次の各項では，**ClassInfo**，**ListOfValues**，**Operation** の各要素が複数のテスト・オブジェクト設定ファイルで定義されている場合に，UFT で実行されるプロセスについて説明します。各テスト・オブジェクト・クラスの **IdentificationProperty** 要素はすべて 1 つのテスト・オブジェクト設定ファイルで定義する必要があります。

ClassInfo 要素

- **ClassInfo** 要素がテスト・オブジェクト設定ファイルで定義されていて，既存の定義よりも優先度が高い場合，この内容は既存の定義に追加されます。複数のファイルで定義されている **ClassInfo** 間で競合が発生した場合，優先度の高いファイルの定義で低いファイルの定義がオーバーライド（置換）されます。
- **ClassInfo** 要素がテスト・オブジェクト設定ファイルで定義されていて，優先度が既存の定義と同じかまたは低い場合，差分が既存の定義に追加されます。複数のファイルで定義されている **ClassInfo** 間で競合が発生した場合，優先度がより低いファイルの定義は無視されます。

ListOfValues 要素

- 複数のファイルで定義されている **ListOfValues** 間で競合が発生した場合，優先度の高いファイルの定義で低いファイルの定義がオーバーライド（置換）されます（マージされません）。
- **ListOfValues** の定義で既存のリストがオーバーライドされる場合，新しいリストには，このテスト・オブジェクト設定ファイルに含まれるクラスの操作で定義されている **Enumeration** タイプの引数すべてが反映されます。
- **ListOfValues** が既存の定義より低い優先度の設定ファイルで定義されている場合，優先度の低い定義は無視されます。

Operation 要素

- **Operation** 要素の定義は，テスト・オブジェクト設定ファイルの優先度に基づいて追加，無視，オーバーライドされます。
- **Operation** 要素が既存の定義よりも優先度が高いテスト・オブジェクト設定ファイルで定義されている場合，クラスの既存の定義にこの操作が追加されます。複数のファイルの **Operation** 定義の間で競合が発生した場合，優先度の高いファイルの定義で低いファイルの定義がオーバーライド（置換）されます（マージされません）。

詳細については，『HP UFT Test Object Schema Help』（Web Add-in Extensibility SDK ヘルプで利用可能）を参照してください。

既存のテスト・オブジェクト・クラスの拡張

コントロールに対する部分的なサポートを備える既存のテスト・オブジェクト・クラスが存在するが，何らかの変更が必要な場合には，コントロールを表す新しいテスト・オブジェクト・クラスを作成します。たとえば，クラス内のテスト・オブジェクトの名前付け規則を変更する場合や，テスト・オブジェクト・メソッドを追加したり変更したりする場合がこれにあたります。新しいテスト・オブジェクト・クラスを作成する場合，このテスト・オブジェクト・クラスは既存のテスト・オブジェクト・クラスに基づき，そのテスト・オブジェクト・メソッドをすべて継承します。

その後，追加のテスト・オブジェクト・メソッドと認識プロパティを実装することで，このテスト・オブジェクト・クラスの機能を拡張できます。さらに，既存のテスト・オブジェクト・メソッドをオーバーライドする別の実装を用意することもできます。メソッドやプロパティの新規作成または変更はテスト・オブジェクト設定ファイルで行い，その実装は JavaScript 関数で設計します。

既存のテスト・オブジェクト・クラスを拡張するには，テスト・オブジェクト設定ファイル内の新しいテスト・オブジェクト・クラスの **ClassInfo** 要素の **ClassInfo\BaseClassInfoName** 属性で，基本テスト・オブジェクト・クラスの名前を定義します。これは，新しいテスト・オブジェクト・クラスが，基本テスト・オブジェクト・クラスのすべてのテスト・オブジェクト・メソッドを，このクラスに対して定義するメソッドに加えてサポートすることを宣言します。

基本クラスを選択する方法

拡張するテスト・オブジェクト・クラスを選択する際には，次のことを考慮します。

- 新しいテスト・オブジェクト・クラスは，基本テスト・オブジェクト・クラスから操作の定義を継承します。したがって，コントロールに対して適切な機能のセットを持つテスト・オブジェクト・クラスを選択する必要があります。
- 基本クラス操作の実装を継承するには，基本テスト・オブジェクト・クラスに一致するタイプの要素がコントロールに含まれる必要があります。

すべてのメソッドとプロパティの実装を確認する方法

継承されたすべてのテスト・オブジェクト・メソッドは，単に宣言されているだけでなく，実装されていることが必要です。これを行う方法の 1 つは，継承されたテスト・オブジェクト・メソッドのそれぞれをサポートする JavaScript 関数を作成することです。もう 1 つの，もっと簡単な方法は，基本テスト・オブジェクト・クラスに一致するタイプの要素をコントロールに含めることです。この要素は**基本要素**と呼ばれます。継承されたテスト・オブジェクト・メソッドのうち，ユーザがカスタム・コントロール専用の実装を定義していないものについては，UFT は基本要素と通信することによって，その内部実装を使用できます。

また，コントロールに基本要素が含まれる場合，次の条件が満たされたときには，UFT は基本テスト・オブジェクト・クラスの実装を使用して，認識プロパティの値を取得します。

- テスト・オブジェクト設定ファイルで，基本テスト・オブジェクト・クラスのプロパティと同じ名前を持つ認識プロパティが新しいテスト・オブジェクト・クラスに対して定義されている。
- これらのプロパティの値を取得する JavaScript 関数を実装していない。

基本要素を UFT に返す必要がある場合

基本要素が，サポートするコントロールのルート要素である場合，UFT は基本要素を認識し，ユーザが実装していないテスト・オブジェクト・メソッドと認識プロパティに対しては基本テスト・クラスの実装を使用します。

基本要素がコントロールのルート要素でない場合，基本要素を UFT に返す 「JavaScript 関数」を作成し，その関数の名前と場所をツールキット設定ファイルに指定する必要があります。この情報は，ツールキット設定ファイルの **Control\Settings** 要素で指定します。たとえば，コントロールが特別な種類のテーブルであり，テーブル要素（通常は UFT によって WebTable 要素として表される）を含む DIV 要素（通常は UFT に無視される）で定義されている場合，WebTable を拡張する MyWebTable テスト・オブジェクト・クラスを作成し，DIV 要素をこの MyWebTable テスト・オブジェクト・クラスにマップすることができます。

基本要素を返すには，**get_base_table** という名前の JavaScript 関数を **HPTable.js** という名前のファイルで実装します。MyWebTable コントロール要素のツールキット設定の **Settings** 要素で，**func_to_get_base_elem** を次のように定義します。

```
<Control TestObjectClass=MyWebTable>
  <Settings>
    <variable name="default_imp_file " value="HPTable.js"/>
    <variable name="func_to_get_base_elem" value="get_base_table"/>
  </Settings>
</Control>
```

カスタマイズしたテスト・オブジェクト・クラスに対するヘルプ・ファイルの作成

UFT では，UFT オンライン・ヘルプの一部として，定義済みのテスト・オブジェクト・クラスに対するオブジェクト・モデル・リファレンスが用意されています。このリファレンスは，UFT ユーザが UFT テスト・オブジェクト，メソッド，プロパティをテストで使用する際に役立ちます。さらに，キーワード・ビューまたはエディタでテスト・オブジェクトまたはテスト・オブジェクト・メソッドに対して F1 キーが押されるか，ステップ・ジェネレータでテスト・オブジェクト・メソッドに対して［操作ヘルプ］ボタンがクリックされると，UFT はオブジェクト・モデル・リファレンスの関連する場所を開きます。

テスト・オブジェクト設定 XML 定義を使用してテスト・オブジェクト・クラスを変更したり新しく定義したりする場合に，定義するテスト・オブジェクト，メソッド，プロパティに対して同様の（**.chm**）ヘルプ・ファイルを作成することができます。これらのヘルプ・ファイルは，ツールキット・サポート・セットの一部としてデプロイし，その場所をユーザに通知します。ヘルプ・ファイルは任意の場所に置くことができます。たとえば，**<UFT インストール・フォルダ>\dat\Extensibility\<UFT アドイン名>\Toolkits\<環境名>\help** フォルダなどに格納します。テスト・オブジェクト設定 XML では，テスト・オブジェクト・クラスとテスト・オブジェクト・メソッドで使用する **HelpInfo** 要素を定義し，ヘルプ・ファイルのパスとファイル内の該当するヘルプ ID を指定します。

> **注意:** ヘルプ・ファイル名は，**<UFT インストール・フォルダ>\help** フォルダにある付属のヘルプ・ファイルの名前とは異なっている必要があります。

HelpInfo 要素が定義されているテスト・オブジェクト・クラスまたはテスト・オブジェクト・メソッドに対して F1 キーまたは［操作ヘルプ］ボタンが押されると，UFT は指定されたコンテキスト・ヘルプ・トピックを開きます。**HelpInfo** 要素を持たない継承されたテスト・オブジェクト・メソッドに対しては，UFT は基本テスト・オブジェクト・クラスに対する付属のヘルプを開きます。

テスト・オブジェクト・クラスに対して作成するヘルプでは，各テスト・オブジェクト・クラスに対して次の情報を指定すると役立ちます。

- このテスト・オブジェクト・クラスが拡張している基本テスト・オブジェクト・クラス（このヘルプに記述されていない継承されたテスト・オブジェクト・メソッドの説明は，基本テスト・オブジェクト・クラスのヘルプにあるという説明を追加するようにします）。
- 使用可能なテスト・オブジェクト・メソッドのリスト（継承されたメソッドを含む）。
- 定義したテスト・オブジェクト・メソッドの説明（メソッドの目的，構文，引数，戻り値，その他の関連情報）。
- 使用可能な認識プロパティのリストとプロパティの説明。

ツールキット設定ファイルについて

ツールキット・サポート・セットの開発で最初に行うのは，基本ツールキット設定ファイルの定義です。設計したツールキット設定ファイルの検証には，**<Extensibility Accelerator インストール・フォルダ>\dat\Toolkit.xsd** ファイルが使用できます（このファイルは **<UFT インストール・フォルダ>\dat\Extensibility\Web\Toolkits** フォルダにもあります）。

ツールキット設定 XML では，ツールキット内のコントロールのサポートに使用する各テスト・オブジェクト・クラスについて，**Control** 要素を定義する必要があります。各 **Control** 要素には，適用されるテスト・オブジェクト・クラスの名前を指定する **TestObjectClass** 属性が含まれる必要があります。

これらの **Control** 要素は，ツールキット全体を表す 1 つの **Controls** 要素の中に含まれます。

ツールキット設定ファイルには，ツールキット内の各コントロールに対して使用するテスト・オブジェクト・クラスを UFT が識別できるようにするための情報が含まれている必要があります。この情

報は，ツールキット・レベルまたはコントロール・レベルのいずれかで提供されます。詳細については次の項で説明します。

ツールキット設定 XML では，他の情報も記述できます。この XML の内容の簡単なまとめを次に示します。また，ツールキット設定 XML の設計と使用の方法の詳細な説明は，このガイドのこの後の各項に記します。

ツールキット固有の情報：

- ツールキットの説明。

  UFT のアドイン・マネージャで，使用可能なアドインのリストからツールキット・サポート・セットの環境名を選択すると，この説明が表示されます。このツールキット・サポート・セットを配布用に開発する場合は，関連する人物または会社を指定する **Provided by** 句をこの説明に含めます。
- ツールキットの優先度。カスタム・コントロールにマップされたテスト・オブジェクト・クラスを識別する際に，UFT はさまざまなツールキットを優先度に従って検索します（優先度が高いものが先）。
- ツールキット内の各コントロールで使用するテスト・オブジェクト・クラスを特定する識別関数（さらにオプションで関数を含むファイル名）。
- 実装関数の呼び出しに使用する標準ファイルの名前。テスト・オブジェクト・クラスで特にファイル名が定義されていない場合に使用します。
- UFT が Web ページに挿入する JavaScript ライブラリ。設計したサポート関数から，このライブラリの関数を呼び出せるようになります。

テスト・オブジェクト・クラス要素：

- このテスト・オブジェクト・クラスが示すカスタム・コントロールの識別に使用する HTML タグと認識条件または識別関数（さらにオプションで関数を含むファイル名）。
- 実装関数の呼び出しに使用する標準ファイルの名前。関数で特にファイル名が定義されていない場合に使用します。
- 基本クラスから継承されているがこのコントロールでは実装されていないプロパティとテスト・オブジェクト・メソッドを実装するテスト・オブジェクトについて，その要素を返す関数（さらにオプションで関数を含むファイル名）。
- このテスト・オブジェクト・クラスのテスト・オブジェクト・メソッドを実装する関数（およびオプションで関数を含むファイルの名前）。UFT はコントロールに対するテスト・ステップを実行するためにこれらの関数を呼び出します。関数が定義されていない場合，UFT はテスト・オブジェクト・メソッドと同じ名前の実装関数を呼び出します。
- コントロールから認識プロパティを取得する関数（さらにオプションで関数を含むファイル名）。関数が定義されていない場合，UFT は **get_property_value** という名前の関数を呼び出します。
- コントロールとその子に関するイベントの処理のために標準設定の Web イベント設定を使用するかどうかを示す要素。カスタマイズしたイベントのリスニングと処理を行うために UFT が実行する関数を指定できます。
- コントロールとその子を学習するタイミングを示す要素と，学習対象となる子を示す関数（任

意）。

- オブジェクト・スパイでこのクラスのテスト・オブジェクトを表示するかどうかを示す要素。
- テスト・オブジェクト・メソッドの引数の有効値リストをコントロールから取得する関数（さらにオプションで関数を含むファイル名）。関数が定義されていない場合，UFT は **get_list_of_values** という名前の関数を呼び出します。
- サポートされているツールキットの状態を取得し，ページが現在の Web ページに完全にロードされているかどうかを確認する関数（さらにオプションで関数を含むファイル名）。

> **注:** UFT は，コントロールと一致するテスト・オブジェクト・クラスを検索する際，このファイル内の **Control** 要素の順序に基づいて検索を行うので，その点を考慮して要素を指定してください。検索では，最初に検出されたテスト・オブジェクト・クラスが使用されます。

この XML の構造と構文の詳細については，『Toolkit Configuration Schema Help』（UFT Web Add-in Extensibility ヘルプから利用可能）を参照してください。

ツールキット・サポート・セット用の JavaScript 関数の設計

ツールキット・サポート・セットの一部として，UFT が呼び出す JavaScript 関数を設計します。実装する JavaScript 関数の説明は，UFT Web Add-in Extensibility JavaScript Function Reference』（『Web Add-in Extensibility ヘルプ』から利用可能)にあります。

JavaScript 関数を作成する際には，次の項の情報も考慮する必要があります。

- 「使用可能な規則，ユーティリティ・オブジェクト，グローバル関数」(44ページ)
- 「ブラウザに依存しないサポートの作成」(45ページ)
- 「グローバル変数スコープ」(47ページ)
- 「グローバル JavaScript 関数の作成」(47ページ)
- 「外部 JavaScript ライブラリの関数の呼び出し」(47ページ)
- 「例外の処理」(48ページ)

使用可能な規則，ユーティリティ・オブジェクト，グローバル関数

Web Add-in Extensibility API には，JavaScript 関数で使用できる次の規則，定数，ユーティリティ・オブジェクト・メソッド，グローバル JavaScript 関数が用意されています。

- **_elem :** UFT が現在処理している Web 要素を表すトークン。識別関数では，_elem トークンは学習対象の HTML 要素を表します。ほかのすべてのコンテキストでは，カスタム・コントロールのルート要素を表します。これには次のコンテキストが含まれます。
  - イベント・ハンドラおよびイベント登録関数。
  - 基本要素，プロパティ値，可能な値のリストを返す関数。

- フィルタ関数。

  たとえば，次の JavaScript 関数は，現在の Web 要素の **id** 認識プロパティの値を返します。

```
function get_property_value(prop)
{
   if (prop == "id")
   {
      return _elem.id;
   }
}
```

- **_util**。さまざまな操作を UFT に実行させるために使用できるメソッドを含むユーティリティ・オブジェクト。詳細については，「グローバル JavaScript メソッドとユーティリティ・メソッド」(48ページ)を参照してください。
- **<Extensibility Accelerator インストール・フォルダ>\bin\PackagesToLoad** フォルダにある **common.js** ファイルで定義されているグローバル JavaScript 関数および定数（このファイルは **<UFT インストール・フォルダ>\dat\Extensibility\Web\Toolkits** フォルダにもあります）。詳細については，「グローバル JavaScript メソッドとユーティリティ・メソッド」(48ページ)を参照してください。
- **window**：テスト対象の Web ブラウザ・ウィンドウに対応する Window オブジェクトにアクセスするために使用します。このオブジェクトを使用して，ブラウザの名前空間に存在するオブジェクトや関数にアクセスし，サポートを作成しているコントロールを管理することができます。

  たとえば，Web ページにグローバル JavaScript 関数 **getTree()** が含まれ，それが特定のツリー・コントロールを管理する **Tree** オブジェクトを返すとします。この **Tree** オブジェクトには，**Collapse**，**Expand**，**Select** メソッドがあります。**Collapse** テスト・オブジェクト・メソッドの実行をサポートするために，次のような JavaScript メソッドを作成できます。

```
function Collapse (treeBranch)
{
  var tree = window.getTree(_elem);
  tree.Collapse(treeBranch);
}
```

ブラウザに依存しないサポートの作成

異なるタイプやバージョンのブラウザでツールキット・サポート・セットがスムーズに動作するようにするには，開発する拡張コードで次のいずれかまたは両方を行います。

- ブラウザごとにコントロールの処理方法を変えるには，**GetBrowserType** または **GetBrowserVersion** ユーティリティ・メソッドを呼び出して，関連するブラウザ情報を取得します。詳細については，「グローバル JavaScript メソッドとユーティリティ・メソッド」(48ページ)を参照してください。
- 異なるタイプやバージョンのブラウザでスムーズに動作するコードを作成するために，**jQuery** などのブラウザに依存しない外部 JavaScript 関数ライブラリの関数を使用します。

jQuery（バージョン 1.3.2）ライブラリおよび **jQuery.simulate** ライブラリは，UFT インストールに付属しており，ブラウザに依存しない関数を備えています。**jQuery** 関数呼び出しは情報の取得と操作の実行に，**jQuery.simulate** 関数呼び出しはイベントのディスパッチの処理に使用します。

たとえば，コントロールに対してクリック操作を実行する場合，現在のブラウザに応じて異なる方法でイベントを送信する複雑なコード・ブロックの代わりに，**jQuery** を使用する 1 行のコードで同じ処理を実現できます。

jQuery を使用した場合：

```
$(_elem).simulate("click");
```

jQuery を使用しない場合：

```
if ( _util.GetBrowserType()==QtpConstants.FireFox)
{
   var myEvent = window.document.createEvent("MouseEvents")
   myEvent.initEvent("click", true, true)
   _elem.dispatchEvent(myEvent)
}
else
   _elem.click();
```

jQuery JavaScript ライブラリと **jQuery.simulate.js** の詳細については，http://jquery.com/ および http://code.google.com/p/jqueryjs/source/browse/trunk/plugins/simulate/jquery.simulate.js?r=6163 を参照してください。

jQuery ライブラリおよびその他の外部 JavaScript 関数ライブラリの使用方法については，「外部 JavaScript ライブラリの関数の呼び出し」(47ページ)を参照してください。

- Mozilla Firefox で動作する必要がある JavaScript 関数では，グローバル名前空間にあるすべての項目にアクセスするために，**window** オブジェクトを明示的に使用する必要があります。たとえば，window.document は Internet Explorer でも Mozilla Firefox でも **document** オブジェクトに正しくアクセスしますが，document は Internet Explorer でしか動作しません。
- Chrome 用に開発する場合は，次の点を考慮してください。
  - **<body>** タグを使用してコントロールを識別していて，**<body>** が blank または about:blank **SRC** 値を持つ **<iframe>** の子である場合，UFT がコントロールを識別できるのは，Chrome 37 以降のみです。
  - Chrome では，カスタム・コントロールが 2 つ以上の兄弟コントロールで構成されている場合，オブジェクトの識別にそれらのすべてを含める必要があります。たとえば，カスタム・コントロールがひとつの編集ボックスおよびそれに隣接する送信ボタン，またはチェックボックスとそのラベルで構成されている場合が，これに該当します。コード・サンプルについては，「例： オブジェクトを識別するための兄弟コントロールの使用」(49ページ)を参照してください。
- コードをブラウザに依存しないように設計していたとしても，各ブラウザは異なるロジックで動作するため，スクリプトはすべてのブラウザで検証してください。

グローバル変数スコープ

JavaScript 関数でグローバル変数を定義した場合，変数はブラウザで動作する JavaScript エンジンのスコープ内に存在します。UFT が JavaScript エンジンに接続するには，UFT がブラウザより前に開かれている必要があります。したがって，グローバル JavaScript 変数がアクセス可能なのは，ブラウザまたは UFT のどちらかが閉じられるまでです。

グローバル JavaScript 関数の作成

推奨される方法としては，各テスト・オブジェクト・クラスに対して JavaScript ファイルを 1 つ作成します。ただし，同じ JavaScript 関数を複数のテスト・オブジェクトの JavaScript 関数から呼び出す場合には，これらのグローバル関数を別のファイルに保存し，そのファイルをツールキット・サポート・セットの共通 JavaScript ファイルとして使用することができます。共通 JavaScript ファイルのパスは，ツールキット設定ファイルの **Controls\Settings\Variable** 要素で指定します。詳細については，『Toolkit Configuration Schema Help』（UFT Web Add-in Extensibility ヘルプから利用可能）を参照してください。

外部 JavaScript ライブラリの関数の呼び出し

設計した JavaScript 関数内で，UFT がテストしている Web ページにインクルードされている任意の JavaScript 関数を呼び出すことができます。

ツールキット・サポート・セットがロードされている場合，作成したコードは，ブラウザで開かれるすべての Web ページで UFT から呼び出される可能性があることに注意してください。たとえば，Web ページ上のオブジェクトを UFT が学習する際に，そのページに自分のコントロールが含まれておらず，関数ライブラリもインクルードされていなくても，設計した識別関数が呼び出される可能性があります。

作成した JavaScript ライブラリがインクルードされていないページからもそのライブラリを使用できるようにするには，その JavaScript ライブラリをすべてのページに挿入するように UFT に指示します。このためには，ツールキット設定ファイル内の **JSLibrariesToInject\JSLibrary** 要素に，ライブラリのファイル・システム・パスを指定します。詳細については，『Toolkit Configuration Schema Help』（UFT Web Add-in Extensibility ヘルプから利用可能）を参照してください。

ツールキット・サポート・セットがロードされた状態で UFT が開かれると，UFT は指定された JavaScript ライブラリをブラウザで開かれるすべての Web ページに挿入します。

JavaScript ライブラリは，まだ Web ページにインクルードされていない場合にのみロードされるように設計することが推奨されます。

ツールキット・サポート・セットを UFT にデプロイする際には，参照している JavaScript ライブラリが，ツールキット設定ファイルに指定したパスに存在することを確認してください。

UFT に付属する jQuery ライブラリの使用

jQuery（バージョン 1.3.2）ライブラリおよび **jQuery.simulate** ライブラリは，UFT インストールに付属しています。これらは次の場所にあります。**<UFT インストール・フォルダ>\bin\JSFiles**

jQuery ライブラリ関数を JavaScript 関数から呼び出すには，ライブラリの場所をツールキット設定ファイルで指定します。

```
<Controls>
  <JSLibrariesToInject>
    <JSLibrary path="INSTALLDIR\bin\JSFiles\jQuery-1.3.2.js" />
    <JSLibrary path="INSTALLDIR\bin\JSFiles\jQuery.simulate.js" />
  </JSLibrariesToInject>
</Controls>
```

これらの jQuery ライブラリは，このライブラリのどれかのバージョンが Web ページにすでにインクルードされている場合はロードされないように設計されています。

例外の処理

設計した JavaScript 関数で例外が発生した場合，UFT は実行時エラー・メッセージを表示し，テスト・ステップは失敗します。設計時に例外に対して指定したメッセージは，失敗したステップのテスト・レポート詳細に追加されます。

実行時エラーを発生させずに，ステップで失敗した実行結果にメッセージを追加するには，JavaScript 関数から **_util.Report** メソッドを呼び出します。

注: 例外に含めるメッセージが単純な文字列の場合は，UFT が表示する実行時エラーに例外が発生したコード行が含まれます。メッセージが書式を使用して文字列に変数を含めている場合には，実行時エラーにコード行は含まれません。

グローバル JavaScript メソッドとユーティリティ・メソッド

ツールキット・サポート・セットの一部として，UFT が呼び出す JavaScript 関数を設計します。JavaScript 関数は，UFT で提供されているグローバル・メソッドとユーティリティ・メソッドを呼び出すことができます。グローバル・メソッドは，**common.js** ファイル内の JavaScript 関数として提供されています。ユーティリティ・メソッドは，**_util** オブジェクトによって公開されています。

_util ユーティリティ・オブジェクトは，JavaScript 関数で呼び出せる次のメソッドを公開しています。

- **Alert (**message**) :** 指定されたメッセージを表示するモーダル・メッセージ・ボックスを開きます。
- **GetBrowserType** / **GetBrowserVersion () :** 現在コントロールを実行しているブラウザの名前またはバージョンを返します。
- **LogLine (**text**,** severity, Id, category**) :** Microsoft Windows イベント・ログ（イベント・ビューアを使用して表示できる）にエントリを追加します。Chrome の場合，Chrome コンソールにエントリを追加します。このメソッドは，サポート・セットのパフォーマンスの分析や機能のデバッグに使用できます。

イベント・ビューアによるツールキット・サポート・セットのデバッグの詳細については，「Microsoft Windows イベント・ログの使用」(60ページ)を参照してください。

- **Record (**method**,** arrParams**,** delay**) :** テストにステップを追加し，テスト・オブジェクトがオブジェクト・リポジトリにない場合はそこに追加します。このメソッドは，コントロールでイベントが発生した後にテストでステップを記録する JavaScript 関数で使用します。記録のサポートの開発の詳細については，「記録のサポートの実装」(70ページ)を参照してください。
- **RegisterForEvent (**element**,** eventName**,** handler**,** [HandlerParam]**):** 特定の Web 要素に対する特定のイベントをリッスンするように登録します。このメソッドは，記録をサポートするためにイベントのリスニングを制御する JavaScript 関数で使用します。このメソッドはイベント・ハンドラから呼び出すこともできます。記録のサポートの開発の詳細については，「記録のサポートの実装」(70ページ)を参照してください。
- **Report (**status**,** method**,** arrParams**,** details**) :** ステップの結果に関する情報を実行結果に追加します。このメソッドは，コントロールに関するステップを実行する JavaScript 関数で使用します。テスト実行のサポートの開発の詳細については，「テスト・オブジェクト・メソッドのサポートの実装」(63ページ)を参照してください。
- **Wait (**milliseconds**)** : 指定した時間（ミリ秒単位）だけ実行を中断します。

common.js の関数のうち最も一般的に使用されるものを次に示します。

- **getFrameElement :** 現在の要素を含むフレーム要素を返します。
- **toSafeArray(inArray) :** JavaScript 配列オブジェクトを COM の SafeArray に変換します。
- **trim(inString) :** 文字列の先頭と末尾の空白を除去して返します。

common.js ファイルには，値が設定された変数も含まれており，コードで定数として使用できます。

common.jsファイルは，**<Extensibility Accelerator インストール・フォルダ>\bin\PackagesToLoad** フォルダと，**<UFT インストール・フォルダ>\dat\Extensibility\Web\Toolkits** フォルダにあります。

これらのメソッドと定数の詳細については，『UFT Web Add-in Extensibility API Reference』（Web Add-in Extensibility ヘルプから利用可能）を参照してください。

例： オブジェクトを識別するための兄弟コントロールの使用

Chrome では，カスタム・コントロールが 2 つ以上の兄弟コントロールで構成されている場合，オブジェクトの識別にそれらのすべてを含める必要があります。

以下は，入力とラベルで構成されているカスタム・コントロールを示しています。

```
<input type="checkbox" role="hidden-checkbox-input" class="ui-helper-hidden"/><label
for="check" class="ui-toggle-button" role="button">Toggle</label>
```

別のブラウザでは，次のように**input**タグのみを指定することで，コントロールを識別できます。

SampleToolkit.xml:

```
<Control TestObjectClass="SampleToggleButton">
   ...
   <Identification type="javascript" function="isSampleToggleButton">
      <Browser name="*">
         <HTMLTags>
            <Tag name="input" />
         </HTMLTags>
      </Browser>
   </Identification>
   ...
</Control>
```

SampleToggleButton.js:

```
function isSampleToggleButton ()
{
   return _elem.getAttribute("role") === "hidden-checkbox-input";
}
```

Chrome では，入力タグとラベル・タグの両方でコントロールを識別します。

このコントロールを識別するには，次に示すようにさらにタグをひとつ **SampleToolkit.xml** に追加します。

```
   <Control TestObjectClass="SampleToggleButton ">
      ...
      <Identification type="javascript" function=" isSampleToggleButton ">
         <Browser name="*">
            <HTMLTags>
               <Tag name="input" />
               <Tag name="label" />
            </HTMLTags>
         </Browser>
      </Identification>
      ...
   </Control>
```

次に，これに応じて次のように **SampleToggleButton.js** 関数を更新します。

```
function isSampleToggleButton ()
{
   if (_elem.getAttribute("role") === "hidden-checkbox-input" && _elem.tagName.toUpperCase()
=== "INPUT")
```

```
      return true;

   if (_elem.tagName.toUpperCase() === "LABEL" && _elem.getAttribute("role") == "button")
    {
        var sibling = _elem.previousSibling;
        if (sibling && sibling.tagName.toUpperCase() === "INPUT" && sibling.getAttribute("role")
=== "hidden-checkbox-input")
        {
            return sibling; // use the sibling input element to replace label to serve as _elem.
        }
    }

   return false;
}
```

カスタム Web コントロールに対して使用するテスト・オブジェクト・クラスを UFT に指示する方法

UFT がコントロールを表すために使用するテスト・オブジェクト・クラスを定義したら，各タイプのコントロールを特定のテスト・オブジェクト・クラスにマップする必要があります。

この識別は，JavaScript 関数または Conditions 要素を使用して実行できます。これらは，コントロールのプロパティをチェックして，それを表すテスト・オブジェクト・クラスを決定します。UFT によるカスタム・コントロールの学習とそれらに対するステップの実行のパフォーマンスを改善するには，JavaScript 関数呼び出しをできる限り避けるように識別要素を定義します。

HTML タグを指定することによって，カスタム・コントロールの識別プロセスを HTML 要素に制限することができます。これによりパフォーマンスをさらに改善でき，**tagName** プロパティをチェックする条件を定義するよりも効果的になります。

各コントロールに対してどのテスト・オブジェクト・クラスを使用するかを認識するための情報を UFT に指定するには，ツールキット設定ファイルの **Controls\CommonIdentification** 要素か，**Control\Identification** 要素を使用します。これらの要素には，次の情報を定義できます。

- **テスト・オブジェクト・クラスごとの HTML タグのセット：** UFT が各コントロールを処理する際に，指定した HTML タグを持つ HTML 要素に対してのみ，**Identification** 要素の定義に基づく識別プロセスが実行されます。

- **テスト・オブジェクト・クラスごとの条件のセット：**UFT が各コントロールをチェックする際に，UFT はコントロールの HTML プロパティを，各 **Control** 要素に定義された条件と照合し，最初に条件が満たされる要素を特定します。

  注: 異なるブラウザで実行する際に，異なる条件や HTML タグを UFT が使用するように定義できます。

- **ツールキット全体に対する 1 つの JavaScript 識別関数：**UFT が各コントロールを処理する際に，UFT はこの関数を呼び出し，関数がコントロールのプロパティをチェックして，適切なテスト・オブジェクト・クラスの名前を返します。

  注: ツールキット・レベル（**Controls\CommonIdentification** 要素内）で識別関数を指定し，テスト・オブジェクト・クラス・レベル（**Control\Identification** 要素内）で条件も指定した場合，UFT は条件をチェックした後で JavaScript 関数を呼び出します。これは不要な関数呼び出しを避けるためです。

- **テスト・オブジェクト・クラスごとの JavaScript 識別関数：**UFT が各コントロールを処理する際に，UFT はこれらの JavaScript 関数を，ツールキット設定ファイルでテスト・オブジェクトが定義されている順序で呼び出します。各関数は，対応するテスト・オブジェクト・クラスがコントロールを表すかどうかをチェックします。UFT は，最初に **true** を返した関数に対応するテスト・オブジェクト・クラスを使用します。

  この識別方法は，各コントロールに対して JavaScript 関数を何度も呼び出すため，パフォーマンスの大幅な低下につながります。したがって，なるべく避けることが推奨されます。

  注: ツールキット・レベル（**Controls\CommonIdentification** 要素内）で識別関数を指定した場合，テスト・オブジェクト・クラス・レベル（**Control\Identification** 要素内）で指定された識別関数は呼び出されません。

- 条件と JavaScript 関数の使用を組み合わせて，コントロールのプロパティに基づいて JavaScript 関数の呼び出しを制限する条件を定義することもできます。詳細については，「Conditions 要素の使用」(52ページ)を参照してください。

Web Add-in Extensibility 用の JavaScript 関数の作成方法の詳細については，「ツールキット・サポート・セット用の JavaScript 関数の設計」(44ページ)を参照してください。

識別要素の構造と構文については，『Toolkit Configuration Schema Help』（『UFT Web Add-in Extensibility ヘルプ』から利用可能）を参照してください。

カスタム・コントロールに対して使用するテスト・オブジェクト・クラスの識別方法を UFT に指示したら，ツールキット・サポート・セットの基本機能をテストできます。詳細については，「ツールキット・サポート・セットの開発中のテスト」(57ページ)を参照してください。

Conditions 要素の使用

Conditions 要素は，テスト・オブジェクト・クラスに対して定義された **Control\Identification\Browser** 要素内で定義できます。これにより，UFT は，コントロールのプロパ

ティに基づいて，識別関数を呼び出さずに，このテスト・オブジェクト・クラスで表されるコントロールを識別することができます。

別の方法として，識別関数を（テスト・オブジェクト・クラスごとに，またはツールキット全体に対して）定義し，UFT が識別関数を呼び出す回数を条件を使用して制限することもできます。このためには，コントロールのプロパティが特定の条件を満たしたときだけ識別関数が呼び出されるように定義します。

異なるブラウザで実行する際に UFT が異なる条件セットを使用するように定義することも，1 つの条件セットを定義して，すべてのブラウザにそれを使用するように指定することもできます。詳細については，『Toolkit Configuration Schema Help』（UFT Web Add-in Extensibility ヘルプから利用可能）で **Browser** 要素の説明を参照してください。

Conditions を作成するには，**Condition** 要素を **and** ロジックまたは **or** ロジックで結合します。各 **Condition** 要素には，HTML コントロールの特定のプロパティと，その期待される値を指定します。コントロールのプロパティの値が指定された値に一致した場合に，条件が満たされます（値が等しい場合に満たされるか等しくない場合に満たされるかを条件の中で指定できます）。**Conditions** 要素をネストして複雑なロジックを作成することもできます。

条件セット，すなわちトップレベルの **Conditions** 要素のそれぞれに対して，次のいずれかのタイプを指定することにより，コントロールのプロパティが要素の条件に一致した場合にコントロールをどのように処理するかを UFT に指示することができます。

- **IdentifyIfPropMatch：**この要素の条件が満たされた場合，現在のテスト・オブジェクト・クラスをコントロールを表すのに使用します。
- **CallIDFuncIfPropMatch：**この要素の条件が満たされた場合，識別関数を呼び出してこのコントロールをチェックします。満たされなかった場合は，現在のテスト・オブジェクト・クラスをコントロールを表すのに使用しません。
- **SkipIfPropMatch：**この要素の条件が満たされた場合，現在のテスト・オブジェクト・クラスをコントロールを表すのに使用しません。

識別関数が定義されており，条件のチェックによって現在のテスト・オブジェクト・クラスをコントロールを表すのに使用するかどうかを決定できない場合は，UFT は識別関数を呼び出します。

- タイプが **IdentifyIfPropMatch** の Conditions 要素が定義されている場合は，ほかのタイプよりも先にチェックされます。
- **CallIDFuncIfPropMatch** タイプと **IdentifyIfPropMatch** タイプの両方の Conditions 要素が定義されている場合，タイプ **CallIDFuncIfPropMatch** の Conditions は，タイプ **IdentifyIfPropMatch** の Conditions が満たされなかった場合のみチェックされます。
- **SkipIfPropMatch** タイプと **IdentifyIfPropMatch** タイプの両方の Conditions 要素が定義されている場合，タイプ **SkipIfPropMatch** の Conditions は，タイプ **IdentifyIfPropMatch** の Conditions が満たされなかった場合のみチェックされます。
- タイプ **CallIDFuncIfPropMatch** の Conditions 要素が定義されている場合，タイプ **SkipIfPropMatch** の Conditions 要素はチェックされません。
- **Conditions** 要素をネストした場合，ネストされた要素のタイプ属性は無視されます。

UFT による識別の実行方法について

UFT は，処理するカスタム・コントロールのそれぞれに対して，次の手順を実行することにより，使用するテスト・オブジェクト・クラスを決定します。

1. 識別関数がツールキット・レベル（ツールキット設定ファイルの **CommonIdentification** 要素内）で定義されている場合，まずすべての **Control** 要素内の **Identification\Conditions** 要素をチェックして，関数の呼び出しを避けられるかどうかを判断します。

   各 **Control** 要素に対して次のことを行います。

   a. **IdentifyIfPropMatch** 条件が定義されており，カスタム・コントロールのプロパティが条件の要件を満たす場合，この **Control** 要素のテスト・オブジェクト・クラスを使用します。満たさない場合は，次の手順に進みます。

   b. **CallIDFuncIfPropMatch** 条件が定義されており，カスタム・コントロールのプロパティが条件の要件を満たさない場合，この **Control** 要素のテスト・オブジェクト・クラスを使用しません。

      この **Control** 要素に **CallIDFuncIfPropMatch** 条件が含まれない場合，次の手順に進みます。

   c. **SkipIfPropMatch** 条件が定義されており，カスタム・コントロールのプロパティが条件の要件を満たす場合，この **Control** 要素のテスト・オブジェクト・クラスを使用しません。

   すべての条件をチェックした後で，一致する **Control** 要素が見つかるか，**すべて**の **Control** 要素の条件がこの **Control** 要素のテスト・オブジェクト・クラスを使用しないことを示す場合を除いて，ツールキット・レベルの識別関数を呼び出します。

   識別関数は，このカスタム・コントロールを表すのに使用するテスト・オブジェクト・クラスの名前を返します。

2. ツールキット・レベルで識別関数が定義されていない場合，**Control** 要素を 1 つずつチェックします。各 **Control** 要素は，1 つのテスト・オブジェクト・クラスに対して定義されています。

   各 **Control** 要素に対して，次の手順を実行して，この要素のテスト・オブジェクト・クラスをカスタム・コントロールに対して使用するかどうかを決定します。最初に一致した **Control** 要素を使用します。

   a. **IdentifyIfPropMatch** 条件が定義されており，カスタム・コントロールのプロパティが条件の要件を満たす場合，この **Control** 要素のテスト・オブジェクト・クラスを使用します。満たさない場合は，次の手順に進みます。

   b. **CallIDFuncIfPropMatch** 条件が定義されており，カスタム・コントロールのプロパティが条件の要件を満たす場合，手順 d に進みます。カスタム・コントロールのプロパティが条件の要件を満たさない場合，この **Control** 要素のテスト・オブジェクト・クラスを使用しません。

      この **Control** 要素に **CallIDFuncIfPropMatch** 条件が含まれない場合，次の手順に進みます。

   c. **SkipIfPropMatch** 条件が定義されており，カスタム・コントロールのプロパティが条件の要件を満たす場合，この **Control** 要素のテスト・オブジェクト・クラスを使用しません。満たさない場合は，次の手順に進みます。

   d. この **Control** 要素内に識別関数が定義されている場合は，それを実行して，この要素のテスト・オブジェクト・クラスをカスタム・コントロールに対して使用するかどうかを決定しま

す。関数は true または false を返します。この **Control** 要素に識別関数が含まれない場合，この **Control** 要素のテスト・オブジェクト・クラスを使用しません。

条件における HTML プロパティの比較方法について

条件に指定された HTML プロパティの値を，指定された期待される値と比較する場合，次の規則が適用されます。

- 文字列値の比較では，大文字と小文字は区別されません。
- 期待される値に指定された文字列を正規表現として扱うように UFT に指示できます。そのためには，**is_reg_exp** 属性を **true** に設定します。
- 数値の比較では，単に数値が比較されます。
- ブール値の比較では，値 **true**，**1**，**yes** はすべて true と見なされます。値 **false**，**0**，**no** はすべて false と見なされます。
- チェックする HTML プロパティがオブジェクトを返す場合，期待される値として valid および null を使用します。プロパティが正常にオブジェクトを返した場合は **valid** と見なされ，オブジェクトを返さなかった場合は **null** と見なされます。
- **Condition** 要素の **equal** 属性を **false** に設定すると，プロパティが特定の値を持たないことをチェックできます。
- **Condition** 要素の **trim** 属性を **true** に設定すると，条件が評価される前に，プロパティの値と期待される値の先頭と末尾のスペースが削除されます。

次の例は，**Conditions** 要素のさまざまな使用法を示します。

例 1：識別関数と条件

カスタム・コントロールに **CalendarBehavior** または **PopupBehavior** プロパティが定義されている場合，UFT は **isCalendar** JavaScript 関数を呼び出して，**TSGCalendar** テスト・オブジェクトをコントロールを表すのに使用するかどうかを決定します。定義されていない場合，ツールキット設定ファイル内の次の **Control** 要素がカスタム・コントロールに一致するかどうかがチェックされます。

```
<Control TestObjectClass="TSGCalendar">
  <Identification type="javascript" function="isCalendar">
    <Browser name="*">
      <Conditions type="CallIDFuncIfPropMatch" logic="or">
        <Condition prop_name="CalendarBehavior" expected_value="valid"/>
        <Condition prop_name="PopupBehavior" expected_value="valid"/>
      </Conditions>
    </Browser>
  </Identification>
</Control>
```

例 2：条件のみ

カスタム・コントロールに **AccordionBehavior** プロパティが定義されている場合，UFT は **TSGAccordion** テスト・オブジェクトを使用してそのコントロールを表します。定義されていない場合，ツールキット設定ファイル内の次の **Control** 要素がカスタム・コントロールに一致するかどうかがチェックされます。

```
<Control TestObjectClass="TSGAccordion">
  <Identification>
   <Browser name="*">
    <Conditions type="IdentifyIfPropMatch" logic="and">
      <Condition prop_name="AccordionBehavior" expected_value="valid"/>
    </Conditions>
   </Browser>
  </Identification>
</Control>
```

例 3：ネストした条件ノード

カスタム・コントロールの **className** プロパティの値が tsg に等しく，コントロールの **id** プロパティの値が tsg_table に等しいか，コントロールの **myType** プロパティの値が table に等しい場合，UFT は **TSGTable** テスト・オブジェクトを使用してそのコントロールを表します。

カスタム・コントロールの **className** プロパティの値が tsg に等しいが，コントロールの **id** プロパティが tsg_table でなく，かつ **myType** プロパティの値が table でない場合，UFT は **isTable** JavaScript 関数を呼び出して，**TSGTable** テスト・オブジェクトを使用するかどうかを決定します。

カスタム・コントロールの **className** プロパティの値が tsg に等しくない場合，UFT は，**isTable** 識別関数を呼び出さずに，ツールキット設定ファイル内の次の **Control** 要素に進みます。

```
<Control TestObjectClass="TSGTable">
  <Identification type="javascript" function="isTable">
   <Browser name="*">
    <Conditions type="IdentifyIfPropMatch" logic="and">
      <Condition prop_name="className" expected_value="tsg"/>
      <Conditions logic="or">
        <Condition prop_name="id" expected_value="tsg_table"/>
        <Condition prop_name="myType" expected_value="table"/>
      </Conditions>
    </Conditions>
    <Conditions type=" SkipIfPropMatch" logic="and">
      <Condition prop_name="className" expected_value="tsg" equal="false"/>
    </Conditions>
   </Browser>
  </Identification>
</Control>
```

Conditions と **Condition**要素の構造と構文については，『Toolkit Configuration Schema Help』（UFT Web Add-in Extensibility ヘルプから利用可能）を参照してください。

ツールキット・サポート・セットの開発中のテスト

テスト・オブジェクト設定ファイルでテスト・オブジェクト・モデルを定義し，各コントロールに対してどのテスト・オブジェクト・クラスを使用するかを UFT が識別するための基本ツールキット設定ファイルを定義したら，ツールキット・サポート・セットのその時点での機能をテストできます。このためには，ツールキット・サポート・セットをデプロイし，環境内のコントロールと UFT との間のやり取りをテストします。

環境サポートの開発のこの後の段階を完了した後で，ツールキット・サポート・セットをもう一度デプロイして，UFT とコントロールの間のその他の種類のやり取り（テスト・オブジェクトの学習，テストの実行など）をテストできます。

異なるブラウザで動作するようにサポートを設計した場合は，それに関するテストも行う必要があります。

テスト・オブジェクト・クラスを定義し，カスタム Web 要素にマップした後で，ツールキット・サポート・セットをテストするには，次の手順で行います。

1. テスト・オブジェクト設定ファイルで，**TypeInformation** 要素の **DevelopmentMode** 属性を true に設定します。これは，UFT が開かれるたびに，すべてのテスト・オブジェクト・クラス情報がファイルから読み込まれるようにするためです。ツールキット・サポート・セットの開発が完了したら，この属性を false に設定します。
2. ツールキット・サポート・セットを UFT コンピュータにデプロイするために，サポート・セットのファイルを UFT インストール・フォルダの下の適切な場所にコピーします。詳細については，「ツールキット・サポート・セットのデプロイ」(74ページ)を参照してください。
3. UFT を開き，GUI テストを開きます。ツールキット・サポート・セットに対して定義した環境名が，［アドイン マネージャ］ダイアログ・ボックスに Web アドインの子として表示されることを確認します（UFT を開いたときに［アドイン マネージャ］ダイアログ・ボックスが表示されない場合の手順については，『HP Unified Functional Testing アドイン・ガイド』を参照してください）。

   ヒント: アドイン・マネージャに表示される環境名は，ツールキット設定ファイル名と同じです。アドイン・マネージャに環境が正しく表示されない場合は，ツールキット設定ファイルの名前が正しく，ファイルが正しいフォルダに存在し，構文が正しいかどうかを確認します。

4. 環境のチェック・ボックスを選択し（これにより Web アドインも自動的に選択されます），［**OK**］をクリックしてツールキットのサポートをロードします。
5. ［新規テストオブジェクトの定義］ダイアログ・ボックスを使用して，定義したテスト・オブ

ジェクト・クラスから新しいテスト・オブジェクトを作成します。環境の名前が［**環境**］リストに表示されます。環境の名前をリストから選択すると，テスト・オブジェクト設定ファイルで定義したテスト・オブジェクト・クラスが［**クラス**］リストに表示されます。

詳細については，『HP Unified Functional Testing ユーザーズ・ガイド』のオブジェクト・リポジトリへの新しいテスト・オブジェクトの定義に関する項を参照してください。

6. カスタム・コントロールを使用しているアプリケーションを開きます。
7. UFT はすでにこのコントロールを認識して学習でき，ユーザはカスタム・テスト・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（UFT で次に示すオプションを使用する方法の詳細については，『HP Unified Functional Testing ユーザーズ・ガイド』を参照してください）。

   a. オブジェクト・スパイを使用して，認識プロパティを表示し，コントロールに対してサポートされているテスト・オブジェクト操作をテストします。

      - 定義した各テスト・オブジェクト・クラスに対して，テスト・オブジェクト設定ファイルで定義したすべての認識プロパティが，オブジェクト・スパイに表示されます。新しい認識プロパティは値なしで表示されます。これは，コントロールから値を取得するメソッドをまだ実装していないからです。

        基本テスト・オブジェクト・クラスのプロパティと同じ名前を持つ認識プロパティに対しては，コントロールのルート Web 要素が基本テスト・オブジェクト・クラスに一致する場合，UFT は基本クラスの実装を使用してプロパティの値を取得します。詳細については，「認識プロパティのサポートの実装」(67ページ)を参照してください。

        > **注:** UFT は，認識プロパティ名として英小文字だけを使用します。テスト・オブジェクト設定ファイル内の認識プロパティ名に英大文字が使用されている場合は，小文字に変換されます。

      - オブジェクト・スパイには，コントロールを表すために定義されている各テスト・オブジェクト・クラスに対して使用できるすべてのテスト・オブジェクト操作が表示されます。これには，テスト・オブジェクト設定ファイルで定義されているテスト・オブジェクト・メソッドに加えて，基本テスト・オブジェクト・クラスから継承されたテスト・オブジェクト・メソッドとプロパティが含まれます。

   b. コントロールを学習するには，［オブジェクト リポジトリ］ダイアログ・ボックスの［**ローカルへオブジェクトを追加**］ボタンを使用します。オブジェクト・リポジトリでテスト・オブジェクトを表すのに正しいアイコンが使用されているのを確認してください。

c. キーワード・ビューで，定義したクラスのテスト・オブジェクトを使用するテスト・ステップを作成します。
   - このクラスに対して標準設定操作を定義してある場合，［**項目**］カラムでテスト・オブジェクトを選択したときに［**操作**］カラムにその操作が表示されます。
   - ［**操作**］カラムの使用可能な操作のリストは，テスト・オブジェクト設定ファイル内の定義を反映し，基本テスト・オブジェクト・クラスから継承された操作も含みます。
   - 操作を選択すると，選択した操作の引数の数に基づいて［**値**］セルが分割され，可能な値を（ListOfValues 要素で）定義してある場合は，値のリストが表示されます。
   - テスト・オブジェクト設定ファイルでテスト・オブジェクト・メソッドに対して定義した説明の文字列はツールヒント，注釈の文字列は［**注釈**］カラムに表示されます。

d. エディタで，定義したクラスのテスト・オブジェクトを使用するテスト・ステップを作成します。ステートメント自動補完機能では，テスト・オブジェクトに対して使用できるすべての操作と，これらの操作に対して使用できる入力値（該当する場合）が，テスト・オブジェクト設定ファイル内の定義に基づいて表示されます（継承されたテスト・オブジェクト・メソッドも表示されます）。

e. ステップ・ジェネレータで，定義したクラスのテスト・オブジェクトを使用するテスト・ステップを作成します。テスト・オブジェクト設定ファイルで定義した操作が［操作］リストに表示され，操作に対して定義した説明がツールヒントに表示されます（継承されたテスト・オブジェクト・メソッドも表示されます）。

追加の UFT 機能で使用するサポートの開発が完了し，サポート・セットをテストするには，次の手順で行います。

1. 前の手順で記述されている手順を実行して，ツールキット・サポート・セットをデプロイし，UFT を開き，サポートをロードして，対象の環境のコントロールを使用するアプリケーションを実行します。
2. サポートを開発している UFT 機能によっては，サポートをテストするためにアプリケーション上で関連する UFT 操作を実行します。たとえば，アプリケーションでコントロールを学習したり，アプリケーションに対してテストを実行したり，アプリケーションでテスト・ステップを記録したりします。

   標準設定では，UFT は，開発するツールキット・サポート・セットの JavaScript ファイルを開いたときに Web ページに挿入します。JavaScript ファイルで行ったその後の変更をテストするには，カスタム・コントロールを含む Web ページを閉じて再度開く必要があります。

   または，ツールキット設定ファイル内で **Controls\Settings\Variable** 要素を定義し，その **name** 属性を cache scripts に，その **value** 属性を false に設定できます。これは，JavaScript ファイルが更新されるたびに，Web ページに再挿入するように UFT に指示します。したがって，JavaScript 関数を変更およびデプロイ後に，XML ファイルで何も変更しなかったか，UFT を再度開いた場合は，Web ページを更新して変更をテストするだけで済みます。

   ツールキット・サポート・セットの開発が完了したら，この要素を削除するか，value 属性を true に設定して，パフォーマンスの低下を回避します。

カスタム・サポートのログとデバッグ

ツールキット・サポート・セットに対する JavaScript 関数を設計する際には，**_util.LogLine** メソッドを使用して，Microsoft Windows のイベントログにログ・メッセージを追加できます。設計したサポートを使用して UFT がテストまたはコンポーネントを実行する際に，これらのメッセージをイベント・ビューアで見ることができます。詳細については，「Microsoft Windows イベント・ログの使用」(60ページ)を参照してください。

設計したツールキット・サポート・セットをテストする際に，JavaScript ファイルをデバッグするには，ほかの JavaScript ファイルの場合と同様に，Microsoft Script Debugger や Microsoft Visual Studio デバッガが使用できます。詳細については，「JavaScript ファイルのデバッグ（Internet Explorer のみ）」(62ページ)を参照してください。

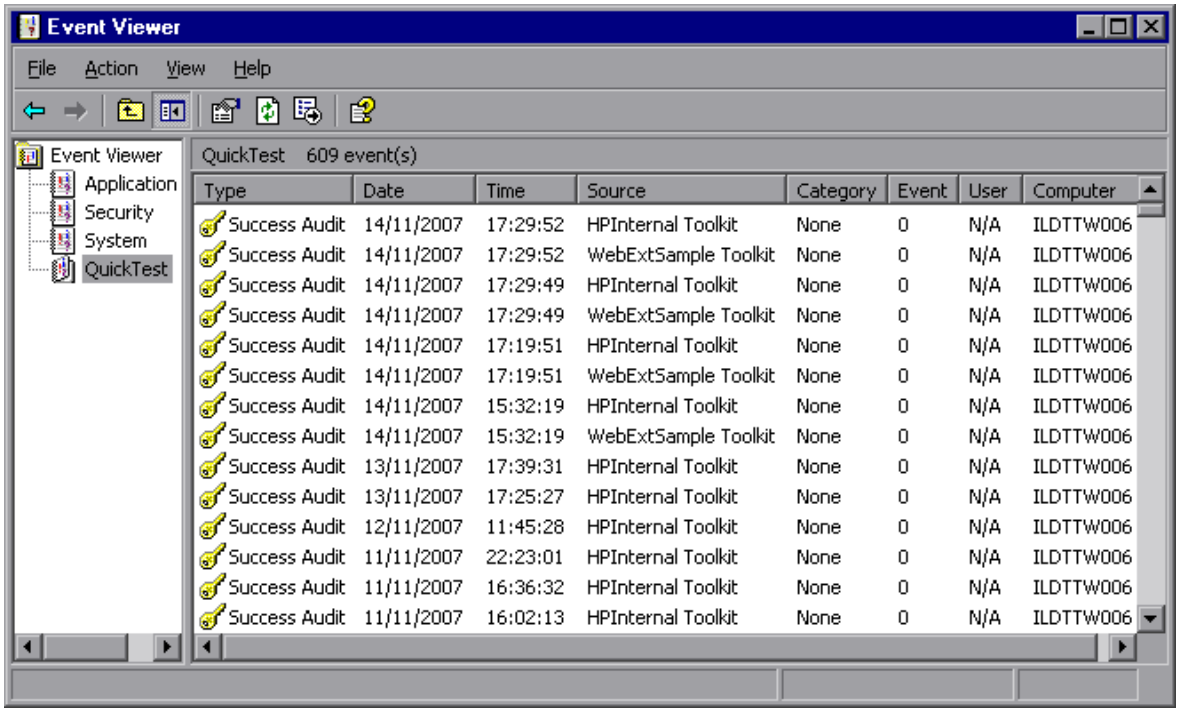
Microsoft Windows イベント・ログの使用

JavaScript 関数で **_util.LogLine** メソッドを使用することにより，Microsoft Windows イベント・ログにメッセージを追加できます。詳細については，UFT Web Add-in Extensibility API リファレンス（Web Add-in Extensibility ヘルプから利用可能）を参照してください。

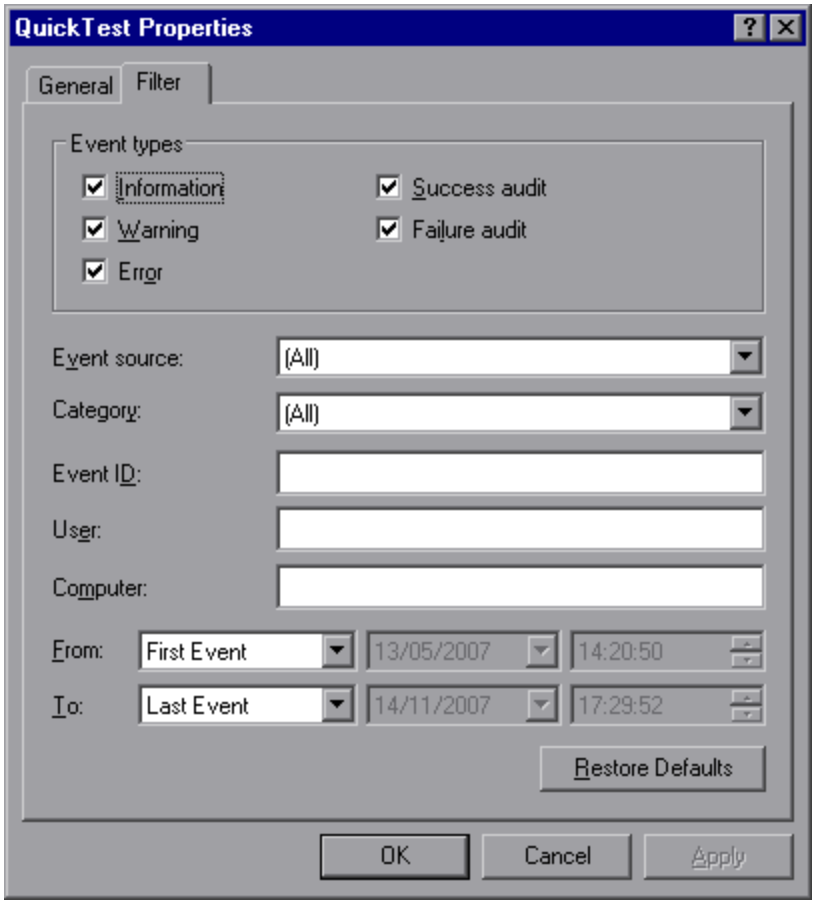
ログ・メッセージのテキストとログ・エントリの重要度，およびオプションで ID とカテゴリ番号を指定します。UFT は，指定された情報にツールキット名と時刻および日付を付加してから，エントリをイベント・ログに追加します。

これに加えて，Web Add-in Extensibility でサポートされるオブジェクトを認識してテストを実行する際にも，UFT はログとエラー・メッセージをイベント・ログに書き込みます。

イベント・ログを表示してツールキット・サポート・セットのパフォーマンスを分析するには，イベント・ビューアを開き（Windows XP および Windows 2000 では，［**スタート**］ > ［**設定**］ > ［**コントロール パネル**］ > ［**管理ツール**］ > ［**コンピュータの管理**］を選択し，［**コンピュータの管理**］ツリーで［**イベント ビューア**］ノードを展開），UFT ノードを選択します。特定のログ・エントリをダブルクリックすると，テキストが表示されます。



イベント・ビューアに表示されるログ・メッセージは，重要度やその他のメッセージ・フィールドに基づいてフィルタできます。［コンピュータの管理］ツールバーで，**［表示］** > **［フィルタ］**を選択します。詳細については，イベント・ビューアのヘルプを参照してください（イベント・ビューアで［**操作**］ > **［ヘルプ**］を選択）。



JavaScript ファイルのデバッグ（Internet Explorer のみ）

Microsoft Script Debugger または Microsoft Visual Studio デバッガを使用して，ツールキット・サポート・セットに対して作成した JavaScript ファイルをデバッグできます

デバッグを有効にするには，Internet Explorer の詳細設定オプション（［**ツール**］ > ［**インターネット オプション**］ > ［**詳細設定**］）で，［**スクリプトのデバッグを使用しない (Internet Explorer)**］オプションと［**スクリプトのデバッグを使用しない (その他)**］オプションをクリアする必要があります。このオプションを変更した後，変更を有効にするには Internet Explorer を再起動する必要があります。

Just-In-Time デバッガの使用可能なデバッガのリストに Microsoft Visual Studio デバッガが表示されるようにするには，Visual Studio の［オプション］ダイアログ・ボックス（［**Visual Studio**］ > ［**ツール**］ > ［**オプション**］ > ［**デバッグ**］ > ［**Just-In-Time**］）で［**スクリプト**］オプションを選択する必要があります。

これらのデバッグ・オプションを有効にすると，JavaScript 関数をデバッグするための標準的な方法がすべて使用可能になります。例：

- Internet Explorer プロセスにアタッチして，関数にブレークポイントを設定できます。
- JavaScript 関数に debugger; ステートメントを含めることで，関数が実行されたときに Just-In-Time デバッガを起動することができます。
- 例外が発生したときに Just-In-Time デバッガが開いて使用可能になります。

テスト・オブジェクト・メソッドのサポートの実装

UFT がカスタム・コントロールを認識できるようにしたら，テスト・オブジェクト・メソッドの実行のサポートを作成する必要があります。メソッドを実装する前に，カスタム・テスト・オブジェクトに対して実行されるステップを持つテストを実行した場合，テストは失敗し，実行時エラーが発生します。

そのためには，テスト・オブジェクト設定ファイルで定義したテスト・オブジェクト・メソッドごとに，コントロールでのステップ実行に必要な JavaScript 関数を記述します。ステップの実行のほかに，実行結果への行の追加，イベント・ログへのログ・メッセージの追加，UFT ユーザへのメッセージ・ボックスの表示といった作業を行う JavaScript 関数を，必要に応じて設計することもできます。詳細については，「ツールキット・サポート・セット用の JavaScript 関数の設計」(44ページ)を参照してください。

ツールキット設定ファイルで，UFT が JavaScript 関数を探すために使用する JavaScript ファイルと，オプションで各テスト・オブジェクト・メソッドに対して使用する関数の名前を指定します。

ツールキット設定ファイル内の各テスト・オブジェクト・メソッドに対して JavaScript ファイルと関数を指定することも，ツールキットごと（**Controls** 要素）またはテスト・オブジェクト・クラスごと（**Controls** 要素）に標準設定の JavaScript ファイルを定義することもできます。UFT は，実装ファイルが指定されていないすべてのテスト・オブジェクト・メソッドに対して，標準設定のファイルを使用します。標準設定では，UFT は指定されたファイル内で，テスト・オブジェクト・メソッドと同じ名前の JavaScript 関数を探します。したがって，関数名が対応するテスト・オブジェクト・メソッド名と異なる場合以外は，ツールキット設定ファイルに関数名を指定する必要はありません。詳細については，（『Toolkit Configuration Schema Help』（『UFT Web Add-in Extensibility ヘルプ』から利用可能）を参照してください。

注意: get_property_value または **get_list_of_values** という名前の JavaScript 関数は作成しないでください。これらの名前は，UFT が（標準設定で）呼び出す特別な JavaScript 関数のために予約されています。それぞれの役割は，テスト・オブジェクトの認識プロパティの実行時の値を取得することと，テスト・オブジェクト・メソッド引数の可能な値のリストを取得することです。

テスト・オブジェクト・メソッドの実行のためのサポートを作成したら，カスタム・テスト・オブジェクトに関する UFT GUI テストを実行し，ツールキット・サポート・セットが正しく動作することを検証できます。ツールキット・サポート・セットのテスト方法の詳細については，「ツールキット・サポート・セットの開発中のテスト」(57ページ)を参照してください。

次の例は，ASPAjax ツールキット・サポート・セットから取られています。これには，ASPAjaxTabs テスト・オブジェクトの **Select** メソッドに対するサポートが含まれています。ASPAjax テスト・オブジェクト設定ファイルでは，これは次のように宣言されています。

```
<ClassInfo BaseClassInfoName="WebElement" Name="ASPAjaxTabs"
        DefaultOperationName="Select" >
   <TypeInfo>
      <Operation ExposureLevel="CommonUsed" Name="Select"            PropertyType="Method">
         <Description>Selects the specified tab.</Description>
         <Documentation>
            <![CDATA[Select the tab with index %a1.]]>
         </Documentation>
         <Argument Name="Index" IsMandatory="true" Direction="In">
            <Type VariantType="Integer"/>
            <Description>
               The index value of the tab to select.
            </Description>
         </Argument>
      </Operation>
   </TypeInfo>
</ClassInfo>
```

ツールキット設定ファイルで，ASPAjaxTabs テスト・オブジェクト・クラスの **Control\Settings** 要素は次のように定義されています。

```
<Control TestObjectClass="ASPAjaxTabs">
   <Settings>
      <!-- Indicates the location of the JavaScript file that contains the implementation of the script -
->
      <variable name="default_imp_file" value="Tabs.js"/>
   </Settings>
</Control>
```

このテスト・オブジェクト・クラスには **Run** 要素は定義されていません。したがって，UFT が **Select** テスト・オブジェクト・メソッドを持つステップを実行する際には，UFT は **Tabs.js** ファイル内で（ツールキット設定ファイルの定義に基づいて），**Select** という名前の JavaScript 関数を探します。

Tabs.js ファイルで，**Select** JavaScript 関数は，インデックスを受け取ってそれに対応するタブを選択するように実装されています。

メソッド引数に対する値の動的なリストのサポート

UFT ユーザがテスト・オブジェクト・メソッドを使用してテスト・ステップを作成する際に，UFT はそのメソッドの引数に使用できる値として，あらかじめ決められた値の組を表示することができます。たとえば，Boolean 型の引数の場合は **true** と **false** が表示され，**month** 型の場合はすべての月の名前が表示されます。ただし，場合によっては，引数に使用できる値の組が限定されていても，その内容がステップが実行される特定のオブジェクトに依存することがあります。次に例を示します。

- **Table(**<table_name>**).SetCellData (**row**,** column**)** 関数の整数型引数である **row** と **column** が実際にとる値は，テーブルの行と列の数によって制限されます。
- また，**Tree(**<tree_name>**).Select (**path**)** 関数の String 型引数である **path** の値は，ツリー内に存在するパスによって制限されます。

Extensibility を使えば，テスト・オブジェクト・メソッドの引数の値のリストを UFT に動的に作成させることができます。UFT がこのリストを作成するのは，エディタでステートメント自動補完機能が使用された場合だけです。

値の動的なリストをサポートするには，次の手順で行います。

1. テスト・オブジェクト設定ファイルで，**Argument** 要素の **DynamicListOfValues** 属性を **true** に設定します。
2. ツールキット設定ファイルで，値のリストを取得するために UFT が呼び出す JavaScript 関数のファイル名と関数名を指定できます。標準設定では， UFT は値のリストを要求するために，テスト・オブジェクト・クラスに対して指定された標準設定の実装ファイル内の **get_list_of_values** JavaScript 関数を呼び出します。標準設定の実装ファイルは，**Control\Settings** 要素の **default_imp_file** 変数で指定されます。詳細については，『Toolkit Configuration Schema Help』（UFT ヘルプから利用可能）を参照してください。

   UFT は，テスト・オブジェクト設定ファイルで **DynamicListOfValues** 属性が **true** に設定されているすべての引数に対して，この JavaScript 関数を呼び出します。この関数に渡されるパラメータには，値を要求するテスト・オブジェクト・メソッドと引数を指定します。
3. テスト・オブジェクト・メソッドと引数の名前を受け取って，現在の要素に関して指定された引数に適切な値のリストを返す JavaScript 関数を作成します。それぞれの値を引用符で囲み，全体を 1 つの文字列に結合した文字列値を返します。

   > **注:** 値の動的なリストは，テストしているアプリケーション内のコントロールから取得されます。したがって，テストの編集中に値の動的なリストを表示するには，対応するコントロールがアプリケーションに表示されている必要があります。

たとえば，WebExtSample 環境で設定したツールキット・サポート（**<Web Add-in Extensibility SDK インストール・フォルダ>\samples\WebExtSample** フォルダ）の場合，動的な値リストは，テスト・オブジェクト・クラス WebExtBook の **GoToAuthorPage** テスト・オブジェクト・メソッドにある **AuthorName** 引数でサポートされます。

- **WebExtSampleTestObjects.xml** テスト・オブジェクト設定ファイルで，この引数は次のように定義されています。

```
<Operation Name="GoToAuthorPage" PropertyType="Method">
   <Description>
      Opens the Web page for the specified author.
   </Description>
   <Argument Name="AuthorName" IsMandatory="true" Direction="In"
         DynamicListOfValues="true">
      <Type VariantType="String" />
      <Description>The author.</Description>
   </Argument>
</Operation>
```

- **WebExtBook.js** ファイル（**WebExtSample.xml** ツールキット設定ファイルで WebExtBook テスト・オブジェクト・クラスの標準設定の実装ファイルとして定義）では，本の著者をそれぞれ引用符で囲んで返す JavaScript 関数が次のように設計されています。

```
// Dynamic list of values implementation
// ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
function get_list_of_values( method, argIndex )
{
   // When creating a step with the GoToAuthorPage test object
   // method, provide a list of the authors of this book, that
   // can be used for the method's argument.
   if (method == "GoToAuthorPage")
   {
      return get_GoToAuthorPage_list_of_values(argIndex);
   }
   return null;
}
function get_GoToAuthorPage_list_of_values(argIndex)
{
   var arr = new Array();
   if( argIndex > 1 )
      return toSafeArray(arr);
   // Retrieve all authors
   var AuthorsCount = 0;
   var authors = window.$(_elem.rows[1].cells[0]).children("A");
   for( var i = 0 ; i < authors.length ; ++i )
   {
      arr[AuthorsCount] = "\"" + authors.eq(i).text() + "\"";
      AuthorsCount++;
   }
   return toSafeArray(arr);
}
```

注: $ は，このコードがブラウザに依存しないサポートを提供するために jQuery JavaScript 関数ライブラリを使用していることを示します。

認識プロパティのサポートの実装

テスト・オブジェクト設定ファイルで，テスト・オブジェクト・クラスの認識プロパティを定義しました。UFT がテストを実行する際には，これらのプロパティの値を取得する必要があります。UFT は，**GetROProperty** などのさまざまなテスト・オブジェクト・メソッドで，認識プロパティの実行時の値を使用します。認識プロパティの実行時の値は，チェックポイントと出力値の作成機能など，さまざまな基本機能を実行する際にも必要になります。

認識プロパティの実行時の値を提供するには，**PropertyName** パラメータを受け取り，UFT が要求したプロパティの値を返す JavaScript 関数を実装する必要があります。(UFT は，認識プロパティ名として英小文字だけを使用します。テスト・オブジェクト設定ファイル内の認識プロパティ名に英大文字が使用されている場合は，小文字に変換されます）。

この関数が返すプロパティ値の形式は，**String**，**Integer**，**Boolean**，**配列**のいずれかです。配列を返す場合，**toSafeArray** 関数を使用して，UFT が受け付ける形式に配列を変換します。認識プロパティの値を配列形式で指定した場合，UFT は配列をセミコロンで区切られた文字列に変換します。

Web Add-in Extensibility 用の JavaScript 関数の作成方法の詳細については，「ツールキット・サポート・セット用の JavaScript 関数の設計」(44ページ)を参照してください。

次の条件が満たされた場合，UFT は基本テスト・オブジェクト・クラスの実装を使用して認識プロパティの値を取得します。

- コントロールに基本要素が含まれる（詳細については「既存のテスト・オブジェクト・クラスの拡張」(40ページ)を参照）。
- テスト・オブジェクト設定ファイルで，認識プロパティが基本テスト・オブジェクト・クラスのプロパティと同じ名前で定義されている。
- その認識プロパティの値を返す関数を定義していない。

次の場合には，テスト・オブジェクト設定ファイルで定義された認識プロパティの値を返す JavaScript 関数を実装します。

- この認識プロパティの値を取得する基本テスト・オブジェクト・クラスの実装が使用できない。
- 基本テスト・オブジェクト・クラスの実装がニーズに一致しない。

認識プロパティの値を返す関数を設計して，**logical_name** プロパティの値を返すようにすると，このテスト・オブジェクト・クラスのテスト・オブジェクトに UFT が名前を付ける方法を制御できます。詳細については，「テスト・オブジェクト名のカスタマイズ」(68ページ)を参照してください。

ツールキット設定ファイルでは，プロパティの値を取得する JavaScript 関数が実装されている JavaScript ファイルを指定できます。この目的のために実装した関数の名前を（**Control\Run\Properties** 要素で）指定することもできます。ただし，関数名を指定しない場合は，UFT は **get_property_value (**PropertyName**)** を呼び出すので，この関数を実装する必要があります。ファイル名を指定しない場合，**Control\Settings** 要素（テスト・オブジェクト・クラス・レベル）または **Controls\Settings** 要素（ツールキット・レベル）で指定した標準設定の JavaScript ファイル内の関数が呼び出されます。詳細については，（『Toolkit Configuration Schema Help』（『UFT Web Add-in Extensibility ヘルプ』から利用可能）を参照してください。

認識プロパティの実行時の値の取得のサポートを作成したら，このツールキット・サポート・セットを使って，Web 要素に対するチェックポイントの実行，プロパティ値の出力，オブジェクト・スパイでのプロパティ値の表示，**GetROProperty** 操作によるテスト・ステップの実行を UFT が正しく実行できるかどうかをテストできます。ツールキット・サポート・セットのテスト方法の詳細については，「ツールキット・サポート・セットの開発中のテスト」(57ページ)を参照してください。

テスト・オブジェクト名のカスタマイズ

UFT がオブジェクトを学習するときには，ページ上の各テスト・オブジェクトに対して一意の名前を作成します。わかりやすいテスト・オブジェクト名を使用すれば，同じクラスのテスト・オブジェクトの間の区別が可能になり，オブジェクト・リポジトリやテストでの識別が容易になります。

テスト・オブジェクトに標準設定で付けられる名前は，テスト・オブジェクト・クラスの名前です（ページ上に同じクラスのテスト・オブジェクトが複数存在する場合はインデックスが追加されます）。多くの場合，これは必ずしもカスタム・コントロールに最適な名前ではありません。

テスト・オブジェクト名は，UFT のユーザにとって意味がある必要があり，ツールキットに関連する用語を使用しているのが望ましいといえます。UFT はこの名前を，キーワード・ビュー，エディタ，オブジェクト・リポジトリに表示します。テスト・オブジェクト名は，オブジェクトの認識には用いられないので，アプリケーション内で一定である必要はありません。

たとえば，言語によって異なるテスト・オブジェクト名を付けることもできます。UFT ユーザは，ある言語で動作するアプリケーションのテストを作成する際に，その言語の名前を持つテスト・オブジェクトを作成することができます。ユーザは，別の言語で同じアプリケーションのテストを実行できます。テスト内のテスト・オブジェクトの名前は元の言語のままですが，UFT はテスト・オブジェクトを記述に基づいて正しく認識し，それに対する操作を実行できます。

あるテスト・オブジェクト・クラスのテスト・オブジェクトに UFT が名前を付ける方法を制御するには，認識プロパティの値を返す関数を設計して，**logical_name** プロパティの値を返すようにします。UFT は，この値をテスト・オブジェクト名に使用します。

たとえば，WebExtSample 環境の **WebExtBook** テスト・オブジェクトは，本の題名に基づいて名付けられます。WebExtSample 環境向けのサンプル・ツールキット・サポート・セットは，**%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample** フォルダにあります。**WebExtBook.js** ファイルは，**WebExtSample.xml** ツールキット設定ファイルで標準設定の実装ファイルとして定義されています。認識プロパティの実行時の値を返す JavaScript 関数は，**WebExtBook.js** ファイルで次のように定義されています。

```
function get_property_value(prop)
{
   if ( prop == "logical_name" || prop == "title" )
   // For the "title" identification property, as well as the
   // "logical_name" property, return the inner text of the
   // second cell in the first row.
   {
      return window.$(_elem.rows[0].cells[1]).text();
   }
```

```
}
```

注: $ は，このコードがブラウザに依存しないサポートを提供するために jQuery JavaScript 関数ライブラリを使用していることを示します。

子コントロールの学習のためのフィルタの実装

UFT に Web ページを学習するように指示すると，［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，Web ページのどの子孫をページとともに学習するかを指定できます。［**すべてのオブジェクト タイプ**］を選択して，親 Web ページとともにカスタム・コントロールを学習するように UFT に指示すると，カスタム・コントロールに含まれるすべてのコントロールも，その Web ページの子（およびコントロール自身の兄弟）として学習されます。

場合によっては，コントロールのすべての子に対してテスト・オブジェクトを作成する必要がないこともあります。たとえば，子に対して実行する重要な操作や取得するプロパティがない場合や，テストの目的では子に対する操作が親に対する操作と見なされる場合などです。例として，計算機コントロールにボタン・コントロールが含まれる場合，数字ボタンに対してテスト・オブジェクトを作成する必要はありません。数字ボタンを押すと，計算機オブジェクト自身に対して **Set** 操作が実行され，計算機を操作するための数値が入力されます。

UFT がどのコントロールを学習するかは，作成したテスト・オブジェクト・クラスに対して学習フィルタを定義することで決定できます。ツールキット設定ファイルの **Control\Filter\Learn** 要素を使用して基本的なフィルタを定義することも，「JavaScript 関数」を作成して複雑なフィルタを実装することもできます。JavaScript 関数でフィルタを設計した場合，関数の名前と場所をツールキット設定ファイルで指定します。

ツールキット設定ファイルの **Control\Filter\Learn** 要素で，次の情報を定義できます。

- このテスト・オブジェクト・クラスで表されるコントロールを学習するかどうか。このタイプのコントロールに子がある場合のみ学習するように UFT に指示することもできます。
- このテスト・オブジェクト・クラスで表されるコントロール内に含まれるコントロールを学習するかどうか。どの子孫を学習するかを決定するために JavaScript 関数を呼び出すように指定することもできます。

  フィルタを実装する JavaScript 関数を作成する場合，関数は UFT が学習する必要がある子孫の Web 要素すべてを含む SafeArray を返す必要があります。詳細については，「ツールキット・サポート・セット用の JavaScript 関数の設計」(44ページ)を参照してください。

詳細については，（『Toolkit Configuration Schema Help』（『UFT Web Add-in Extensibility ヘルプ』から利用可能）を参照してください。

学習フィルタの定義の例は，WebExtSample 環境のサンプル・ツールキット・サポート・セットに含まれています。このサンプルは **%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample** フォルダにあります。

- **WebExtBook** テスト・オブジェクト・クラスの **Filter** 要素は（**WebExtSample.xml**ファイルで）次のように定義されています。

```
<Filter>
  <Learn learn_control="Yes" learn_children="No"/>
</Filter>
```

これは，**WebExtBook** テスト・オブジェクトを学習し，子孫は学習しないように UFT に指示します。

- **WebExtUsedBooks** テスト・オブジェクト・クラスの **Filter** 要素は次のように定義されています。

```
<Filter>
  <Learn learn_control="Yes" learn_children="CallFilterFunc"
      type="javascript" function="GetChildrenToLearn" />
</Filter>
```

これは，**WebExtUsedBooks** テスト・オブジェクトを学習し，**CallFilterFunc** JavaScript 関数を呼び出して学習する子孫を決定するように，UFT に指示します。

GetChildrenToLearn JavaScript 関数は，**WebExtUsedBooks.js** ファイルにあります。このファイルは，**WebExtSample.xml** ツールキット設定ファイルで標準設定の実装ファイルとして定義されています。**GetChildrenToLearn** JavaScript 関数は，UsedBooks テーブル・コントロールの子孫のラジオ・ボタン・コントロールをすべて返します。

```
function GetChildrenToLearn()
{
// Return all of the radio buttons in the used books table
return toSafeArray(window.$(_elem).children()[0].getElementsByTagName("input") );
}
```

注: $ は，このコードがブラウザに依存しないサポートを提供するために jQuery JavaScript 関数ライブラリを使用していることを示します。

学習フィルタを実装したら，カスタム・コントロールを学習するように UFT に指示し，コントロールのどの子が学習されるかをツールキット・サポート・セットが正しく制御することを確認できます。ツールキット・サポート・セットのテスト方法の詳細については，「ツールキット・サポート・セットの開発中のテスト」(57ページ)を参照してください。

記録のサポートの実装

UFT でオブジェクト・リポジトリにオブジェクトを追加してテストを作成する方法の 1 つは，記録を使用することです。テストを記録する場合，UFT は Web 要素に対するイベントをリッスンするように登録し，イベントが発生すると，対応するステップをテストに追加します。標準設定では，UFT は

標準の Web イベント設定を使用して，各 Web 要素に対してリッスンするイベントと，各イベントが発生したときにテストに記録するステップを決定します。

定義したテスト・オブジェクト・クラスに対する記録をカスタマイズするには，記録するイベントと，そのイベントが発生したときにテストに追加するステップを指定する必要があります。

記録をカスタマイズする各テスト・オブジェクト・クラスに対して，ツールキット設定ファイルで **Control\Record\EventListening** 要素を定義します。この要素では，このテスト・オブジェクト・クラスで表されるコントロールに対するイベントの処理に標準の Web イベント設定を使用するかどうかを指定できます。さらに，これらのコントロールの子に対して発生するイベントの処理に標準の Web イベント設定を使用するかどうかも指定できます。

標準の Web イベント設定を使用するかどうかの指定のほかに，より具体的なイベント登録を実行する JavaScript 関数（およびオプションで関数を含むファイルの名前）も指定できます。詳細については，（『Toolkit Configuration Schema Help』（UFT Web Add-in Extensibility ヘルプから利用可能）を参照してください。

ツールキット設定ファイル内の定義に加えて，次の種類の JavaScript 関数を作成します。

- イベントのリッスンを登録する 1 つの JavaScript 関数（ツールキット設定ファイルに名前を指定）。これは，**_util** ユーティリティ・オブジェクトの **RegisterForEvent** 関数を使用して，適切な要素に対する適切なイベントのリッスンを登録します。この関数の引数によって，各イベントが発生したときに UFT が呼び出す JavaScript 関数が決まります。

  UFT は，標準の Web イベント設定に基づいてイベントのリッスンを登録した後で，この関数を呼び出します。この関数によって実行されるイベント登録は，同じイベントに対するそれ以前のすべての登録をオーバーライドします。この関数によって処理されないイベントに対しては，標準の登録が使用されます。

- イベントを処理する 1 つまたは複数の JavaScript functions。これは，**_util** ユーティリティ・オブジェクトの **Record** 関数を呼び出して，テストに追加するステップを UFT に指示します。

  Record 関数などのユーティリティ・オブジェクト関数は，**SafeArray** タイプの引数を渡す必要があります。配列を **SafeArray** に変換するには，Web Add-in Extensibility に用意されている toSafeArray (array) 関数を使用します。この関数は，**<Extensibility Accelerator インストール・フォルダ>\bin\PackagesToLoad\common.js**で定義されています（このファイルは，**<UFT インストール・フォルダ>\dat\Extensibility\Web\Toolkits** フォルダにもあります）。

ユーティリティ・オブジェクト関数の構文については，『API Reference』の『UFT Web Add-in Extensibility API Reference』（Web Add-in Extensibility ヘルプから利用可能）の **_util** の項を参照してください。Web Add-in Extensibility 用の JavaScript 関数の作成方法の詳細については，「ツールキット・サポート・セット用の JavaScript 関数の設計」(44ページ)を参照してください。

記録のカスタマイズの例は，WebExtSample 環境のサンプル・ツールキット・サポート・セットに含まれています。このサンプルは **%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample** フォルダにあります。

- ツールキット設定ファイルで，**WebExtBook** テスト・オブジェクト・クラスの **Control** 要素内で，次の **Record\EventListening** 要素が定義されています。

```
<Record>
  <EventListening
    use_default_event_handling_for_children="false"
    use_default_event_handling="false"
    type="javascript" function="RegisterToEvents"/>
</Record>
```

これによって UFT は，Book コントロールとその下位コントロールで発生したイベントを処理する際，標準の Web イベント定義を使用するのではなく，JavaScript 関数 **RegisterToEvents** を呼び出します。JavaScript ファイルは定義されていないので，UFT は JavaScript 関数を，WebExtBook テスト・オブジェクト・クラスに対して **Control** レベルで指定されている **WebExtBook.js** ファイルで探します。

- **WebExtBook.js** ファイルには，次の **RegisterToEvents** 関数が定義されています。

```
function RegisterToEvents( elem )
{
  // Connect to the "Select" event:When the book name or the
  // book icon is clicked, call OnSelectClicked.
  _util.RegisterForEvent( window.$(_elem.rows[0].cells[0]).
      children[0], "onclick", "OnSelectClicked" );
  _util.RegisterForEvent( window.$(_elem.rows[0].cells[0]),
      "onclick", "OnSelectClicked");
}
```

注: $ は，このコードがブラウザに依存しないサポートを提供するために jQuery JavaScript 関数ライブラリを使用していることを示します。

この関数は，本の題名と画像に対するクリック・イベントをリッスンするように UFT に登録します。イベントに対する登録の際に，この関数はイベント発生時に UFT から呼び出される JavaScript 関数を指定します。

- **WebExtBook.js** ファイルに，次のイベント・ハンドラ JavaScript 関数を追加します。

```
function OnSelectClicked( handlerParam , eventObj )
{
  // Record the "Select" step
  var arr = new Array();
  _util.Record( "Select", toSafeArray(arr) , 0 );
  return true;
}
```

この関数は，本の題名または画像がクリックされたときに，WebExtBook テスト・オブジェクトに対する **Select** テスト・オブジェクト・メソッドを記録します。

記録に対するサポートを実装したら，環境内のコントロールに関するテストを記録し，ツールキット・サポート・セットが正しく動作することを確認できます。ツールキット・サポート・セットのテ

スト方法の詳細については，「ツールキット・サポート・セットの開発中のテスト」(57ページ)を参照してください。

トラブルシューティングと制限事項 - サポートの開発

本項では，Web Add-in Extensibility サポートの開発に関するトラブルシューティングと制限事項について説明します。

- Web Add-in Extensibility を使用すると，Web ページおよびフレーム内の Web コントロールに対するサポートを作成できます。Web ページまたはフレーム自身に対するカスタム・サポートは開発できません。すなわち，Web Add-in Extensibility を使用して開発したテスト・オブジェクト・クラスに，Web ページまたはフレームをマップすることはできません。
- Web Add-in Extensibility で開発されたサポートを UFT が正しく使用するためには，ブラウザのセキュリティ設定がアクティブ・スクリプトを許可するように設定されている必要があります（たとえば，Internet Explorer 7 では，このセキュリティ設定は次の場所にあります：［**ツール**］ > ［**インターネット オプション**］ > ［**セキュリティ**］ > ［**レベルのカスタマイズ**］ > ［**スクリプト**］ > ［**アクティブ スクリプト**］）。
- JavaScript 関数を設計する際には，Mozilla Firefox ページで単純な **Click** または **FireEvent** メソッド呼び出しを使用してリンクをクリックすることはできません。この場合，リンクをクリックするには，Web Add-in Extensibility の **_util** オブジェクトに用意されている特別な **FireEvent** メソッドが使用できます（このメソッドはこのような場合の回避策専用なので，API リファレンスには記述されていません）。

  このメソッドの使用方法の例を次に示します。

```
function clickOnLink(link) {
   if (_util.GetBrowserType() == QtpConstants.IE) {
      link.click();
   }
   else {
      //Firefox.
      var evObj = window.document.createEvent("MouseEvents");
      evObj.initEvent("click", true, true);
      _util.FireEvent(link, "click", evObj);
   }
}
```

第4章: ツールキット・サポート・セットのデプロイ

カスタム・ツールキットの UFT サポートを拡張する作業で最後に行う手順が，ツールキット・サポート・セットのデプロイです。この手順では，作成したすべてのファイルを，UFT をインストールしたコンピュータ上の正しい場所に配置します。これにより，UFT はツールキット内のコントロールを識別し，コントロール上でテストを実行できるようになります。

ツールキット・サポート・セットの開発作業では，ツールキット・サポート・セットを UFT にデプロイすることによって，作成したサポートのテストとデバッグを実行できます。完成したツールキット・サポート・セットは，UFT がインストールされているコンピュータにデプロイすると，Web Add-in を拡張できます。

本章の内容

- カスタム・ツールキット・サポートのデプロイについて 75
- カスタム・ツールキット・サポートのデプロイ 75
- デプロイ済みサポートの変更 77
- デプロイ済みサポートの削除 79

カスタム・ツールキット・サポートのデプロイについて

UFT がインストールされているコンピュータにツールキット・サポート・セットをデプロイすると，UFT ユーザはツールキット・サポート・セットを UFT アドインとして使用できるようになります。

UFT が起動するとアドイン・マネージャが開き，ツールキット・サポート・セットの環境名が Web アドイン・ノードの下に子ノードとして表示されます。環境のチェック・ボックスを選択すると，開発したツールキット・サポート・セットを使用して，その環境のサポートがロードされます。

環境のサポートがロードされると，次のようになります。

- UFT は，環境内のコントロールを認識し，そのコントロール上でテストを実行できるようになります。
- UFT は，アドインまたはサポートされる環境のリストを表示するすべてのダイアログ・ボックスに，環境の名前を表示します。
- UFT は，ツールキット・サポート・セットによって定義されたテスト・オブジェクト・クラスのリストを，各アドインで使用可能なテキスト・オブジェクト・クラスのリストを表示するダイアログ・ボックス内に表示します（たとえば，（［新規テスト オブジェクトの定義］ダイアログ・ボックス，［オブジェクトの認識］ダイアログ・ボックス）。

カスタム・ツールキット・サポートのデプロイ

作成したツールキット・サポート・セットをデプロイするには，UFT インストール・フォルダ内の特定の場所にファイルを格納する必要があります。

注: 作業を始める前に，**<UFT インストール・フォルダ>\dat\Extensibility\Web** フォルダ）にカスタム・ツールキットと同じ名前のフォルダを作成します。

ツールキット・サポート・ファイルを配置する場所を次の表にまとめます。

ファイル名	場所
<カスタム・ツールキット名>TestObjects.xml **注:** 上記は，ファイル名で推奨される命名規則です。テスト・オブジェクト設定 XML ファイルを複数作成して，それぞれに任意の名前を付けることができます。	• <UFT インストール・フォルダ>\dat\Extensibility\ Web • <ALM 用 UFT アドインのインストール・フォルダ>\dat\Extensibility\Web （任意。このフォルダが存在する場合にのみ必要です。つまり，ALM 用 UFT アドインは，ALM アドイン ページとは関係なくインストールされており，UFT インストールの一部には含まれません。）

ファイル名	場所
<カスタム・ツールキット名>.xml	<UFT インストール・フォルダ>\dat\Extensibility\Web\Toolkits\<カスタム・ツールキット名>
JavaScript ファイル	ファイルの場所は，**<カスタム・ツールキット名>.xml** 内の で指定します。
新しいテスト・オブジェクト・クラスのアイコン・ファイル（任意）	**.dll**，**.exe** または **.ico** ファイルのいずれかを，UFT がインストールされているコンピュータまたはネットワーク上のアクセス可能な場所に格納します。 ファイルの場所は，<カスタム・ツールキット名>TestObjects.xml で指定します。
テスト・オブジェクト・クラスのヘルプ・ファイル（任意）	**.chm** ファイルを，UFT がインストールされているコンピュータに格納します。 ファイルの場所は，<カスタム・ツールキット名>TestObjects.xml で指定します。
外部 JavaScript ライブラリ このライブラリはツールキット・サポート・セットには含まれていませんが，サポート・セットのデプロイメント時に，ツールキット設定ファイルで指定された場所にこのライブラリ・ファイルが格納されていることを確認してください。	**.js** ファイルは，UFT がインストールされているコンピュータまたはネットワーク上のアクセス可能な場所に格納します。 ファイルの場所は，<カスタム・ツールキット名>.xml で指定します。

推奨されるファイルの格納場所

JavaScript ファイル，ヘルプ・ファイル，アイコン・ファイルの場所は，ツールキット・サポート・セットの設定ファイルで指定します。場所の指定には，相対パスを使用できます。詳細については，『Test Object Schema ヘルプ』および『Toolkit Configuration Schema Help』（Web Add-in Extensibility ヘルプで利用可能）を参照してください。

各ファイルの格納場所として推奨される場所を次の表にまとめます。

ファイル名	場所
JavaScript ファイル	<UFT インストール・フォルダ>\dat\Extensibility\Web\Toolkits\<カスタム・ツールキット名>\JavaScript
アイコン・ファイル	<UFT インストール・フォルダ>\dat\Extensibility\Web\Toolkits\<カスタム・ツールキット名>\Res
ヘルプ・ファイル	<UFT インストール・フォルダ>\dat\Extensibility\Web\Toolkits\<カスタム・ツールキット名>\Help

設計段階で使用する設定

- テスト・オブジェクト設定ファイルの**Identification Property** 要素の属性を変更する場合，カスタム・ツールキット・サポートの設計段階では，**TypeInformation** 要素の **DevelopmentMode** 属性の値を true にしてください。通常の用途にカスタム・ツールキット・サポート・セットをデプロイする場合は，この属性を削除（または false に設定）してから作業を開始してください。詳細については，「テスト・オブジェクト設定ファイルで指定された認識プロパティ属性の変更」(78ページ)を参照してください。
- JavaScript ファイルが更新されるたびに，Web ページに再挿入するように UFT に指示するには，ツールキット設定ファイル内で **Controls\Settings\Variable** 要素を定義し，その **name** 属性を cache scripts に，その **value** 属性を false に設定します。こうすると，JavaScript 関数を変更およびデプロイ後に，XML ファイルで何も変更しなかったか，UFT を再度開いた場合は，カスタム・コントロールを含む Web ページを更新して変更をテストするだけで済みます。Web ページを閉じて再度開く必要はありません。

  通常の用途にカスタム・ツールキット・サポート・セットをデプロイする場合は，この属性を削除するか，**value** 属性を true に設定して，パフォーマンスの低下を回避します。

デプロイ済みサポートの変更

デプロイ済みのツールキット・サポート・セットを変更する場合，UFT を再度開いてから Web アプリケーションを再起動すると，変更内容が有効になります。

UFT で使用する認識プロパティの機能に関する定義を変更する場合は，「テスト・オブジェクト設定ファイルで指定された認識プロパティ属性の変更」を参照してください。

テスト・オブジェクト設定ファイルで指定された認識プロパティ属性の変更

テスト・オブジェクト設定ファイル中の**IdentificationProperty**要素には，UFT で（［オブジェクトの認識］ダイアログ・ボックスを使用して）変更される可能性がある情報を指定する属性があります。その属性とは，**AssistivePropertyValue**，**ForAssistive**，**ForBaseSmartID**，**ForDescription**，**ForOptionalSmartID**，**OptionalSmartIDPropertyValue** です。この属性に基づいて，UFT で各種用途に使用する認識プロパティのリストが決まります。詳細については，『UFT Test Object Schema Help』（UFT Web Add-in Extensibility ヘルプで利用可能）を参照してください。

したがって標準設定では，UFT は属性値を XML ファイルから 1 度だけ読み込みます。これにより，［オブジェクトの認識］ダイアログ・ボックスでユーザが行った変更内容が上書きされることはなくなり，UFT では，このようにして，ユーザ定義のプロパティ・リストが変更されないようにしています。

［オブジェクトの認識］ダイアログ・ボックスの［**テスト オブジェクトを元に戻す**］ボタンをクリックすると，属性値が XML から再ロードされます。

前回のロード以降 XML が変更されている場合（システムのファイル変更日時から判断），属性が XML から読み込まれます。さらにこれらの属性の値に基づいて，認識プロパティが関連のリストに追加されます（必要に応じてリスト内の順序も調整されます）。ただし，リストから既存の認識プロパティが削除されることはありません。

UFT を起動するたびに XML 内で定義されている属性値に基づいて認識プロパティ・リスト全体を更新するには，テスト・オブジェクト設定ファイルの **TypeInformation** 要素の **DevelopmentMode** 属性を true に設定します。

認識プロパティの属性の変更に関する注意事項

- テスト・オブジェクト設定ファイルの**Identification Property** 要素の属性を変更する場合，カスタム・ツールキット・サポートの設計段階では，**TypeInformation** 要素の **DevelopmentMode** 属性の値を true にしてください。これにより，ファイルへの変更内容すべてが UFT で有効になります。
- 実際の使用のためにツールキット・サポート・セットをデプロイする場合は，事前に **TypeInformation** 要素の **DevelopmentMode** 属性を削除（または false に設定）してください。削除または無効にしないと，UFT が起動するたびに，テスト・オブジェクト設定ファイルでの定義に基づいてプロパティ・リストが更新されます。UFT ユーザが［オブジェクトの認識］ダイアログ・ボックスでプロパティ・リストを変更した後，UFT を再起動すると，変更内容は破棄されます。
- テスト・オブジェクト設定ファイルの読み込み時，既存のプロパティがプロパティ・リストから削除されることはありませんが（ただし，**DevelopmentMode** 属性が true に設定されている場合を除く），プロパティがリストに追加され，ファイル内の定義に基づいて順序が調整されます。UFT ユーザが［オブジェクトの認識］ダイアログ・ボックスでリストからプロパティを削除した場合や，順序を変更した場合には，変更後のファイルをロードした時点で変更内容は破棄されます。

  カスタム・ツールキット・サポート・セットを第三者に提供し，変更済みのテスト・オブジェクト設定ファイルを含むアップグレードを行う場合には，認識プロパティのリストが変更される可能性があることを UFT ユーザに通知してください。

デプロイ済みサポートの削除

UFT では，特定の環境またはツールキット向けに提供されているサポートを起動時にロードするかどうかをアドイン・マネージャで指定できます。

デプロイメントの完了後，カスタム・ツールキットのサポートを UFT から削除するには，次のツールキット設定ファイルを削除する必要があります。**UFT インストール・フォルダ>\dat\Extensibility\Web\Toolkits\<カスタム・ツールキット名>**

テスト・オブジェクト設定ファイル内に，カスタム・コントロールに対応するテスト・オブジェクト・クラス定義が存在しない場合（つまりファイルが不要），次の場所からファイルを削除できます。**UFT インストール・フォルダ>\dat\Extensibility\Web** および
UFT Add-in for ALM インストール・フォルダ>\dat\Extensibility\Web（該当する場合）

第2部: チュートリアル：Web カスタム・ツールキットのサポートの作成方法の学習

このチュートリアルでは，Internet Explorer を使用します。作成する JavaScript コードは Internet Explorer のスタイルで記述されるので，サンプルのカスタム・コントロール用に作成するサポートは Microsoft Internet Explorer のみで正しく動作します。Web Add-in Extensibility サポートは，Mozilla Firefox や Google Chrome でも動作するように設計することも可能です。詳細については，「ブラウザに依存しないサポートの開発」(32ページ)を参照してください。

第5章: 簡単な Web カスタム・コントロールに対する UFT サポートの作成方法の学習

このレッスンでは，Extensibility Accelerator for HP Functional Testing と一緒にインストールされる Web Add-in Extensibility Book サンプル・ツールキットを使用して Book コントロールのサポートを手動で作成します。Book コントロールのサポートの作成では，最小限のカスタマイズのみを行うため，ツールキット・サポート・セットの作成に関する基本事項を習得できます。

コントロールのサポートの作成では，Extensibility Accelerator を使用してサポートを設計します。ツールキット・サポート・セット・ファイルの作成と XML ファイルのセットアップは Extensibility Accelerator によって自動実行されるので，JavaScript 関数の設計に専念できます。このレッスンでサポートを作成することで，Web Add-in Extensibility ツールキット・サポート・セットの構造とコンテンツの理解を深めることができます。

%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample フォルダにこのサンプルで使用する完全なツールキット・サポート・セットが含まれているので，参考にしてください。

レッスンを始める前に，「UFT Web Add-in Extensibility の概要」(12ページ)をよく読んでおいてください。

このレッスンの内容

- このレッスンの準備 82
- Web Add-in Extensibility Book サンプル・ツールキットのサポートの計画 82
- ツールキット・サポート・セットの開発 90
- レッスンのまとめ 111

このレッスンの準備

カスタム・コントロールの UFT サポートを拡張する作業には，ソース・ファイルへのアクセス権限が必要です。UFT でカスタム・コントロールをサポートするためにコントロールのソースを変更する必要はありませんが，ソースを理解しておく必要はあります。コントロールを構成する要素と属性，コントロールで発生する可能性のあるイベントなどに関する情報を準備しておいてください。この情報は，サポート・クラスを設計するときに使用します。

Book コントロールのソース・ファイルは，**%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample\Application\Book.htm** にあります。

ファイルを開いてコントロールを実行します。

コントロールを実行し，ソース・ファイルを開いてコントロールの動作と実装を確認します。

Book コントロールには，本の題名，著者，新品価格，古本の最低価格といった情報が含まれます。

タイトルまたは本の画像をクリックすると，本に関する詳細情報が表示されます（ただし，このサンプルでは実装されていません）。著者名をクリックすると，その著者の著書が一覧表示されます（ただし，このサンプルでは実装されていません）。［**Used**］をクリックすると UsedBooks ページが開き，この本の入手可能なすべての古本とその価格のリストが表示されます。UsedBooks テーブルはより複雑なコントロールです。このコントロールのサポートは，レッスン「複雑な Web カスタム・コントロールに対する UFT サポートの作成方法の学習」(113ページ)で学習します。

Web Add-in Extensibility Book サンプル・ツールキットのサポートの計画

ここでは，UFT による Book コントロールの認識について，現在の方法とカスタマイズ後の方法を比較検討します。次に，「Web Add-in Extensibility 計画の理解チェックリスト」(27ページ)の質問に回答し，「Web Add-in Extensibility の計画チェックリスト」(29ページ)に記入します。

この一連の作業を効率的に行うためには，UFT が Book コントロールを認識する方法をオブジェクト・スパイ，キーワード・ビュー，記録オプションで確認し，コントロールの実装方法と用途を検討する必要があります。

1. **UFT を開いて Book コントロールを実行します。**

   UFTを開いて Web Add-in をロードします。

Book コントロールの開いているインスタンスがある場合は，閉じてから開きます。これには，**%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample\Application\Book.htm** ファイルを開いてください。

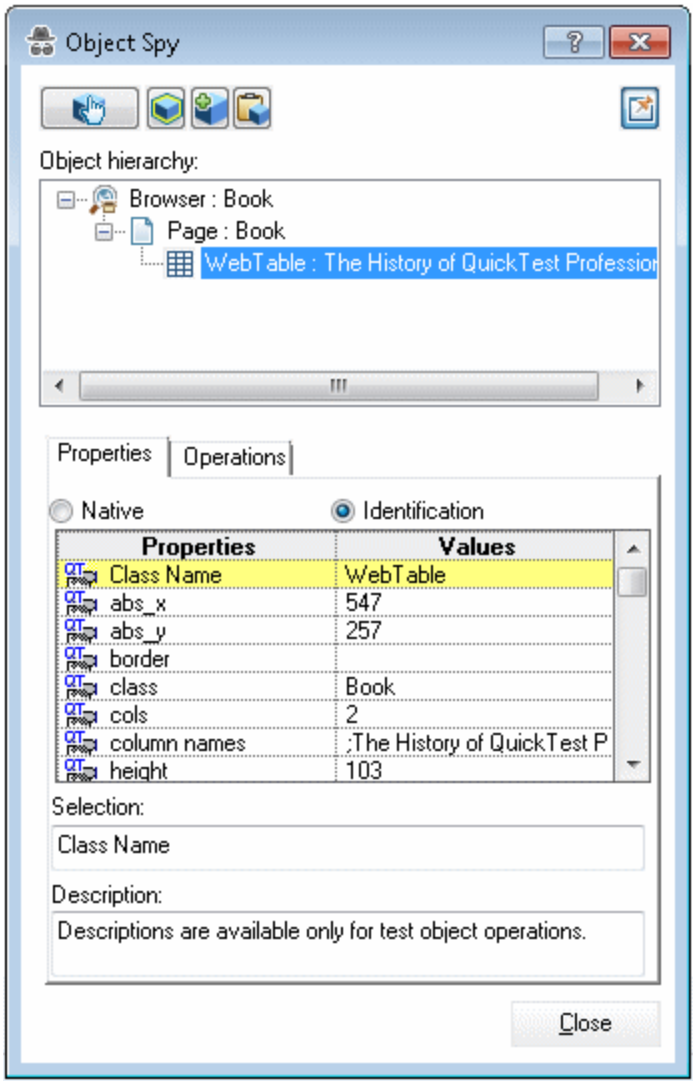
2. **オブジェクト・スパイを使用して Book のプロパティと操作を表示します。**

   UFT で，GUI テストを開き，［**ツール**］ > ［**オブジェクト スパイ**］を選択するか，［**オブジェクト スパイ**］ツールバー・ボタンをクリックして，［**オブジェクト スパイ**］ダイアログ・ボックスを開きます。［**プロパティ**］タブをクリックし，［**認識プロパティ**］を選択します。

   ［オブジェクト スパイ］ダイアログ・ボックスで，指差しアイコンをクリックしてから，Book コントロールをクリックします。

   Book コントロールは Web テーブルとして実装され，その UFT サポートは標準で組み込まれています。したがって，UFT は本の題名が名前として割り当てられた **WebTable** としてコントロールを認識します。テスト・オブジェクトに使用されるアイコンは，標準の WebTable クラスのアイコンです。

次の「チェックリスト」には，上記の回答が反映されています。



オブジェクト・スパイを閉じます。

3. **Book コントロール上の操作を記録します。**

   UFT で，［**記録**］ > ［**記録と実行環境設定**］を選択して，［記録と実行環境設定］ダイアログ・ボックスを開きます。［Web］タブで，［**任意の開いているブラウザを対象にテストを記録および実行**］を選択します。［**OK**］をクリックします。

   ［**記録**］ボタンをクリックするか，［**記録**］ > ［**記録**］を選択します。Book コントロール内のさまざまなリンクをクリックします（クリックした後は，Book コントロールに戻るために前のページに戻る必要があります）。本の題名，コントロール内の画像，著者名，［**Used**］リンクなどをクリックしてみます。

クリックするたびに，新しいステップがテストに追加されます。

Item	Operation	Value	Documentation
Action1			
Book			
Book			
The History of QuickTest	Click		Click the "The History of QuickTest" link.
Not Implemented	Sync		Wait for the Web page to synchronize before continuing the run.
Book	Back		Navigate back to the previous page of the browser.
Book			
Book	Click		Click the "Book" image.
Not Implemented	Sync		Wait for the Web page to synchronize before continuing the run.
Book	Back		Navigate back to the previous page of the browser.
Book			
Jane Doe	Click		Click the "Jane Doe" link.
Not Implemented_2	Sync		Wait for the Web page to synchronize before continuing the run.
Book	Back		Navigate back to the previous page of the browser.
Book			
Used:	Click		Click the "Used:" link.

［**停止**］ボタンをクリックするか，［**記録**］ > ［**停止**］を選択して記録セッションを終了します。

ここで記録されるのは単純な **Click** ステップであり，各ステップは Book コントロール内で定義されている各オブジェクトに対応付けられています。**Click** 操作は，異なる名前の Web Link テスト・オブジェクトまたは Book 画像テスト・オブジェクトに対して独立に記録されています。これらのステップは，このコントロールのコンテキストでは有用な意味を持ちません。

4. **Book コントロールの定義先となるが属するカスタム・ツールキットを決定します。**

   コントロールに対する UFT サポートを拡張する場合は，必ずツールキットのコンテキストで行います。このチュートリアルの目的では，2 つのカスタム Web コントロールがグループ化されて，WebExtSample という名前のカスタム・ツールキットを構成しています。2 つのカスタム・コントロールとは，Book と UsedBooksTable です。

   このレッスンでは，WebExtSample ツールキットのサポートを作成します。まず最初に，Book コントロールのみをサポートします。

5. **カスタム・クラス・サポート計画のチェックリストを完成させます。**

   Book コントロールは，Web テーブルを使用して次のように実装されています。

```
<table class="Book">
  <tr>
    <td class="BookImageCell" rowspan="4">
      <a href=".\QtpHistory.htm">
        <img class="BookImage" alt="Book"   src=".\Res\Book.jpg"/>
      </a>
    </td>
    <td class="BookCell">
      <a class="BookTitle" href=".\QtpHistory.htm" >   The History of QuickTest Professional
</a>
```

```
      </td>
    </tr>
    <tr>
      <td class="BookCell">
         By:<a href=".\JaneDoe.htm">Jane Doe</a>,   <a href=".\JohnDoe.htm">John Doe</a>
      </td>
    </tr>
    <tr>
      <td class="BookCell">
      </td>
    </tr>
    <tr>
      <td class="BookCell">
         New:<strong>59.99$</strong>   <a href=".\UsedBooks.htm">Used:</a> from
<strong>29.99$</strong>
      </td>
    </tr>
</table>
```

ここでは，Book コントロールのサポートを計画する際に決定すべき項目について説明し，サポート計画チェックリストを使って各項目をまとめます。

a. カスタム・コントロールを表すテスト・オブジェクト・クラスを選択します。

   Book コントロールは，表示をサポートする Web テーブル・コントロールとして実装されます。このコントロールに関するテストを実行するという目的であれば，UFT で Book コントロールをテーブルとして認識する必要はありません。ただし，WebElement オブジェクトを使用して汎用の Web 要素に提供される基本的なサポートでは Book コントロールに対しては不十分です。したがって，**WebExtBook** という名前のテスト・オブジェクト・クラスを新規作成して **WebElement** を拡張し，このテスト・オブジェクト・クラスを Book コントロールを表すクラスとして認識するように UFT に指示する必要があります。

b. UFT がコントロールを表すのに使用するテスト・オブジェクト・クラスを識別する方法を次のように定義します。

   コントロールの **tagName** プロパティが **table**，**className** プロパティが **Book** の場合，**WebExtBook** テスト・オブジェクトを使用してそのコントロールを表します。

c. 新しいテスト・オブジェクト・クラスの詳細を決定します。
   - 新しいテスト・オブジェクト・クラスは，次のアイコン・ファイルで表されます。**<UFT インストール・フォルダ>\Dat\Extensibility\Web\Toolkits\WebExtSample\Res\WebBook.ico**
   - ヘルプ・ファイルは提供されません。
   - サポートが必要な新しい認識プロパティは，**title**，**authors**，**price**，**min_used_price** です。これらのプロパティは UFT の［チェックポイントのプロパティ］ダイアログ・ボックスと［出力値のプロパティ］ダイアログ・ボックスに表示され，チェックポイントでは標準で選択されます。スマート認識に使用されるプロパティはありません。

     オブジェクトを一意に定義する認識プロパティは，本の題名と著者名です。
   - テスト・オブジェクト自身には，title 認識プロパティと同じ名前を付けてください。

d. カスタム・コントロールをサポートするテスト・オブジェクト・メソッドを決定します。

   WebExtBook テスト・オブジェクト・クラスは，次のテスト・オブジェクト・メソッドを提供します。
   - **Select：**本の題名または画像のクリックをシミュレートします。これは標準設定のテスト・オブジェクト・メソッドです。
   - **GoToAuthorPage：**指定した著者名のクリックをシミュレートします（使用可能な著者名は，実行時に特定のコントロールから取得されます）。
   - **GoToUsedBooksPage：**［**Used**］リンクのクリックをシミュレートします。

e. メソッド引数の値の動的リストのサポートが必要かどうかを決定します。

   GoToAuthorPage メソッドの **AuthorName** 引数では，動的リストが必要です（ツールキット設定ファイルを修正して，値を提供する JavaScript 関数を指定し，該当する JavaScript 関数を設計する必要があります）。

f. このコントロールの学習時に UFT の学習対象とするコントロールの子を定義します。

   技術的にはその内部のコントロールで実行されるとしても，テストの目的のためには，すべての操作が Book コントロール自身の上で実行されているかのようにUFT が認識する必要があります。したがって，コントロールを学習させたり，テスト・オブジェクトで表す必要はありません。

g. **WebExtBook** テスト・オブジェクトをオブジェクト・スパイで表示するかどうかを決定します。表示します。

h. 記録をサポートするかどうかと，記録の対象となるイベントを定義します。

   コントロールの要素に発生するマウス・クリックをリッスンします。対象となる要素は，title，image，authors，Used です。この要素でクリックが発生すると，テストに関連するステップが記録されます。

i. サポートのどの部分をツールキット設定ファイルで設計し，どの部分に JavaScript 関数が必要かを決定します。
   - シンプルな Book コントロールでは，テスト・オブジェクト・クラスは html プロパティ値のみで識別されます。したがって，ツールキット設定ファイルの変更だけで対応でき，JavaScript 関数を作成する必要はありません。
   - JavaScript 関数は，標準の命名規則でテスト・オブジェクト・メソッドと認識プロパティをサポートできるので，ツールキット設定ファイルの変更は不要です。
   - UFT による学習でコントロールの子を学習対象から除外する設定は，ツールキット設定ファイル内で行うことができ，JavaScript 関数は必要ありません。
   - 記録をサポートするには，ツールキット設定ファイルにおいて，標準の Web イベントの設定を無効にし，JavaScript 関数を指定して UFT が正しいイベントをリッスンするようにします。さらに，イベント登録を処理する JavaScript 関数と，イベント発生時に関連ステップを記録するように UFT に指示する JavaScript 関数を作成します。

次の「チェックリスト」には，上記の回答が反映されています。

Web Add-in Extensibility の計画チェックリスト

☑	カスタム・コントロールのサポート計画チェックリスト	ツールキット XML で指定	JavaScript 関数でサポート
☑	このカスタム・コントロールのソースの場所：%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample\Application\Book.htm	該当なし	該当なし
☑	新しいテスト・オブジェクト・クラスが拡張する Web テスト・オブジェクト基本クラスを指定：（標準設定—WebElement）WebElement	該当なし	該当なし
☑	基本テスト・オブジェクト・クラスは WebElement か？ はい 「いいえ」の場合，基本要素（基本テスト・オブジェクト・クラスに対応する要素）があるか？ 該当なし 基本要素がある場合，それを返す JavaScript 関数が必要か? 該当なし	いいえ	いいえ
☑	新しい Web テスト・オブジェクト・クラスの詳細を指定： • テスト・オブジェクト・クラス名： WebExtBook • アイコン・ファイルの場所（任意）：<UFT インストール・フォルダ>\Dat\Extensibility\Web\Toolkits\WebExtSample\Res\WebBook.ico • 記述のための認識プロパティ： title，authors • 標準設定のテスト・オブジェクト・メソッド： Select • ヘルプ・ファイルの場所：該当なし	該当なし	該当なし
☑	コントロールに使用するテスト・オブジェクト・クラスを識別するための基準を指定：	はい	いいえ

☑	カスタム・コントロールのサポート計画チェックリスト	ツールキット XML で指定	JavaScript 関数でサポート
	tagName = table, className = Book.		
☑	テスト・オブジェクトの名前付けの基礎を指定： 本の題名を使用	該当なし	はい
☑	サポートする認識プロパティのリスト。チェックポイントで使用するプロパティ（および標準で選択されるプロパティ）とスマート認識で使用するプロパティ（該当する場合）をマーク。 title，authors，price，min_used_price （すべてチェックポイントで使用可能，標準で選択，ただしスマート認識では使用されない）	いいえ	はい
☑	サポートするテスト・オブジェクト・メソッドのリスト（必要な場合，引数，戻り値，ヘルプ・ファイルの場所，ヘルプ ID を指定）： Select () GoToAuthorPage (AuthorName) GoToUsedBooksPage ()	いいえ	はい
☑	テスト・オブジェクト・メソッド引数に対して動的な値のリストを指定するか？ はい 指定する場合，引数をリスト： GoToAuthorPage メソッドの AuthorName	該当なし	はい
☑	UFT がコントロールで学習する子のタイプを指定： なし	はい	いいえ
☑	オブジェクト・スパイにこのクラスのテスト・オブジェクトを表示させるか？ はい	いいえ	該当なし
☑	記録をサポートするか？ はい その場合，記録をトリガするイベントを列挙： 題名，画像，著者名，Used のクリック	はい	はい

ツールキット・サポート・セットの開発

次の手順に従って，WebExtSample ツールキットのツールキット・サポート・セットを開発し，Web Add-in Extensibilityの基本について学習します。ツールキット・サポート・セットの開発作業は，次の工程に従って行います。

- 「第 1 段階：ツールキット・サポート・セットの作成」（90 ページを参照）
- 「第 2 段階：WebExtSample 環境の UFT への導入」（91 ページを参照）
- 「第 3 段階：Book コントロールの識別，調査，学習の実行を UFT に指示」（93 ページを参照）
- 「第 4 段階：WebExtBook テスト・オブジェクト・メソッドのサポートを実装」（98 ページを参照）
- 「第 5 段階：WebExtBook の認識プロパティのサポートを実装」（101 ページを参照）
- 「第 6 段階：テスト・オブジェクトの名前の変更」（103 ページを参照）
- 「第 7 段階：子オブジェクトを学習対象から除外するフィルタを実装」（104 ページを参照）
- 「第 8 段階：Book コントロール上の記録のサポートを実装」（106 ページを参照）
- 「第 9 段階：AuthorName の動的な値リストのサポートを実装」（109 ページを参照）

第 1 段階：ツールキット・サポート・セットの作成

ここでは，WebExtSample ツールキット用のツールキット・サポート・セットを構成するファイルとフォルダを作成します。

ツールキット・サポート・セットを作成するには，次の手順で行います。

1. ツールキット・サポート・セットのフォルダを作成します。

   このフォルダの名前と場所は任意に決めることができます。
2. ツールキット・サポート・セット・フォルダの下に，**WebExtSampleTestObjects.xml** という名前のファイルを作成します。これはテスト・オブジェクト設定ファイルです。
3. ツールキット・サポート・セットのフォルダに，**Toolkits** という名前のフォルダを作成します。
4. **Toolkits** フォルダに，**WebExtSample** という名前のフォルダを作成します。
5. **Toolkits\WebExtSample** フォルダに，次の項目を作成します。
   - **WebExtSample.xml** という名前のファイル（ツールキット設定ファイルです）。
   - **WebExtBook.js** という名前のファイル（Book コントロールのサポート用に設計する JavaScript 関数をすべて記述したファイルです）。
   - **WebBook.ico** アイコン・ファイルを含む **Res** という名前のフォルダ（アイコン・ファイルは **%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample\Res** からコピーできます）。

第 2 段階：WebExtSample 環境の UFT への導入

ここでは，ツールキット設定ファイルとテスト・オブジェクト設定ファイルを使用して，WebExtSample 環境を UFT に導入します。Web Add-in Extensibility が最初に行う処理は，環境を UFT に通知する処理です。ツールキット設定ファイルでは新しい環境（および名前）を UFT に通知し，テスト・オブジェクト設定ファイルでは環境向けに設計したテスト・オブジェクト・モデルを記述します。

ツールキット設定ファイルの設計

ツールキット設定ファイルではまず，新しくサポートされた環境を UFT に通知する設定を行います。

新しいサポート対象環境を UFT に通知する処理は，基本的なツールキット設定ファイルを作成し，環境名と同じ名前を付けるだけで完了です。作成する基本的なツールキット設定ファイルには，**Controls** 要素 1 つと，これに含まれる **Control** 要素（それぞれが 1 つのテスト・オブジェクト・クラスを記述）が 1 つ以上存在します。ツールキット設定ファイル内の要素と属性の詳細については，『UFT Java Add-in Extensibility Toolkit Configuration Schema Help』（Web Add-in Extensibility ヘルプから利用可能）を参照してください。

UFT に WebExtSample 環境を通知するには，次の手順で行います。

「第 1 段階：ツールキット・サポート・セットの作成」(90ページ)で作成した **WebExtSample.xml** ファイルに次のテキストを入力します。

```
<?xml version="1.0" encoding="UTF-8"?>
<Controls>
    <Control TestObjectClass="WebExtBook"/>
</Controls>
```

このファイルを UFT コンピュータ上の正しい場所に格納すると，UFT の起動時に，WebExtSample 環境がアドイン・マネージャで Web アドインの下の子ノードとして表示されます。WebExtSample のチェック・ボックスを選択すると，UFT はこの環境用のサポートをロードします。

このレッスンでは，この **Control** 要素にさらに要素を追加することにより，ツールキット・サポート・セット全体で使用する JavaScript 関数の場所と，次の UFT 機能のサポートに関する情報を指定します。

- コントロールを表すのに使用するテスト・オブジェクト・クラスを識別する機能（オブジェクト・スパイとコントロールの学習をサポート）
- コントロールの学習時に子コントロールをフィルタ処理する機能
- テスト・ステップを記録するために，コントロール上のイベントをリッスンする機能

テスト・オブジェクト設定ファイルの設計

テスト・オブジェクト設定ファイルを使用して，WebExtSample 環境とテスト・オブジェクト・モデルを UFT に導入します。

TypeInformation 要素の **PackageName** 属性は，このテスト・オブジェクト設定ファイル（およびファイル内で定義したテスト・オブジェクト）を WebExtSample 環境に関連付けます。UFT の起動時に WebExtSample 環境を選択しないと，UFT はこのファイルのテスト・オブジェクト・クラス定義を無視します。

テスト・オブジェクト設定ファイル内の要素と属性の詳細については，『UFT Test Object Schema Help』（『Web Add-in Extensibility ヘルプ』から利用可能）を参照してください。

WebExtSample テスト・オブジェクト・モデルをテスト・オブジェクト設定ファイルで定義するには，次の手順で行います。

「第 1 段階：ツールキット・サポート・セットの作成」(90ページ) で作成した **WebExtSampleTestObjects.xml** ファイルに，次のテキストを追加します。このテキストは，「Web Add-in Extensibility の計画チェックリスト」(88ページ)の内容に従って，**WebExtSample** 環境と **WebExtBook** テスト・オブジェクト・クラス（テスト・オブジェクト・メソッドと認識プロパティを含む）を定義します。

```
<?xml version="1.0" encoding="UTF-8"?>
<TypeInformation Load="true" AddinName="Web"
            PackageName="WebExtSample">
    <ClassInfo BaseClassInfoName="WebElement"
           Name="WebExtBook"
           DefaultOperationName="Select" >
       <IconInfo IconFile= "INSTALLDIR\dat\Extensibility\Web\
            Toolkits\WebExtSample\Res\WebBook.ico"/>
       <TypeInfo>
         <Operation ExposureLevel="CommonUsed"
                Name="Select"
                PropertyType="Method">
            <Description>Selects the book.</Description>
            <Documentation>
               <![CDATA[Select the %l book.]]>
            </Documentation>
         </Operation>
         <Operation ExposureLevel="CommonUsed"
                Name="GoToAuthorPage"
                PropertyType="Method">
            <Description>
               Opens the Web page for the specified author.
            </Description>
            <Documentation>
               <![CDATA[Open the Web page for %a1.]]>
```

```
          </Documentation>
          <Argument Name="AuthorName"
                IsMandatory="true"
                Direction="In"
                DynamicListOfValues="true">
            <Type VariantType="String" />
            <Description>The author.</Description>
          </Argument>
        </Operation>
        <Operation ExposureLevel="CommonUsed"
              Name="GoToUsedBooksPage"
              PropertyType="Method">
          <Description>
            Opens the UsedBooks page.
          </Description>
          <Documentation>
            <![CDATA[Open the %l UsedBooks page.]]>
          </Documentation>
        </Operation>
      </TypeInfo>
      <IdentificationProperties>
        <IdentificationProperty ForDefaultVerification="true"
                        ForVerification="true"
                        ForDescription="true"
                        Name="title"/>
        <IdentificationProperty ForDefaultVerification="true"
                        ForVerification="true"
                        ForDescription="true"
                        Name="authors"/>
        <IdentificationProperty ForDefaultVerification="true"
                        ForVerification="true"
                        ForDescription="false"
                        Name="price"/>
        <IdentificationProperty ForDefaultVerification="true"
                        ForVerification="true"
                       ForDescription="false"
                        Name="min_used_price"/>
      </IdentificationProperties>
    </ClassInfo>
  </TypeInformation>
```

第 3 段階：Book コントロールの識別，調査，学習の実行を UFT に指示

カスタム・コントロールをサポートするには，各コントロールを表すためにどのテスト・オブジェクト・クラスを使用するかを UFT が識別できる必要があります。したがって，Web Add-in Extensibility

で最も基本的な要素となるのが **Identification** 要素であり，ツールキット設定ファイル内で **Control** 要素ごとに定義します。**Control** 要素は，それぞれが 1 つのテスト・オブジェクト・クラスを定義します。**Identification** 要素は，各テスト・オブジェクト・クラスを使用してどのコントロールを表すかを指定します。

テスト中のアプリケーション内のコントロールを UFT が認識する必要がある場合，UFT はテスト・オブジェクト・クラスごとに定義された **Identification** 要素をチェックします。コントロールを表すのに使用されるクラスは，**Identification** 定義が最初に一致したテスト・オブジェクト・クラスです。

「Web Add-in Extensibility Book サンプル・ツールキットのサポートの計画」(82ページ)で説明したように，**tagName** プロパティが **table** であり，**className** プロパティが **Book** であるコントロールは，**WebExtBook** テスト・オブジェクトによって表されます。この定義はツールキット設定ファイルで簡単に行うことができ，JavaScript 関数は必要ありません。

WebExtBook テスト・オブジェクト・クラスの認識ルールを定義するには，次の手順で行います。

WebExtSample.xml ファイル内のテキストを，次のテキストで置き換えます。

```
<?xml version="1.0" encoding="UTF-8"?>
<Controls>
  <Control TestObjectClass="WebExtBook">
    <Identification>
     <Browser name="*">
      <Conditions type="IdentifyIfPropMatch" logic="and">
        <Condition prop_name="tagName" expected_value="TABLE"/>
        <Condition prop_name="className" expected_value="Book" />
      </Conditions>
     </Browser>
    </Identification>
  </Control>
</Controls>
```

これによって，**WebExtBook** テスト・オブジェクト・クラスを定義する **Control** 要素に，**Identification** 要素が追加されます。要素には **Conditions** 要素が 1 つあり，これには 2 つの条件が含まれています。コントロールが **WebExtBook** としてみなされるには，この 2 つの条件を両方満たす必要があります。**Conditions** 要素内の **Condition** 要素で，それぞれ 1 つずつ条件を指定します。コントロールの HTML プロパティの値と，指定された値が一致すると（大文字と小文字を区別しないで比較），条件が満たされたとみなされます。

このチュートリアルでは，上記の定義を使用することにより，1 つの **Conditions** 要素に含まれる複数の **Condition** 要素を使用する方法を紹介しています。ただし，アプリケーションのページ上にコントロールが多数存在する場合や，DOM 構造が大きい場合には，次のように識別ルールを定義することをお勧めします。

```
  <Identification>
   <Browser name="*">
    <HTMLTags>
```

```
      <Tag name="TABLE"/>
    </HTMLTags>
    <Conditions type="IdentifyIfPropMatch">
      <Condition prop_name="className" expected_value="Book" />
    </Conditions>
  </Browser>
</Identification>
```

この定義では，TABLE 要素のみで識別プロセスが実行されるのでパフォーマンスが向上します。

テスト・オブジェクト・クラスで **Identification** 要素を定義する方法の詳細については，「カスタム Web コントロールに対して使用するテスト・オブジェクト・クラスを UFT に指示する方法」(51ページ) および 『UFT Java Add-in Extensibility Toolkit Configuration Schema Help』（Web Add-in Extensibility ヘルプで利用可能）を参照してください。

ツールキット・サポート・セットのデプロイとテスト（第 3 段階）

WebExtBook テスト・オブジェクト・クラスをテスト・オブジェクト設定ファイルで定義し，このテスト・オブジェクト・クラスの識別ルールをツールキット設定ファイルで定義する作業が完了したら，UFT でツールキット・サポート・セットを使用することによる影響をテストできる準備が整います。

注: ツールキット・サポート・セットのカスタマイズでは，テスト・オブジェクト設定ファイルで **Identification Property** 要素の属性を変更する場合，カスタム・ツールキット・サポートの設計段階では，**TypeInformation** 要素の **DevelopmentMode** 属性を true に設定しておいてください。通常の用途にカスタム・ツールキット・サポート・セットをデプロイする場合は，この属性を削除（または false に設定）してから作業を開始してください。ただし，この操作はこのチュートリアルのレッスンでは必要ありません。詳細については，「テスト・オブジェクト設定ファイルで指定された認識プロパティ属性の変更」(78ページ)を参照してください。

ツールキット・サポート・セットをデプロイするには，次の手順を実行します。

1. **WebExtSampleTestObjects.xml** ファイルを **<UFT インストール・フォルダ>\dat\Extensibility\Web** にコピーします。
2. **<UFT インストール・フォルダ>\dat\Extensibility\Web\Toolkits** フォルダに，**WebExtSample** という名前のフォルダを作成します。
3. **WebExtSample.xml** ファイルを **<UFT インストール・フォルダ>\dat\Extensibility\Web\Toolkits\WebExtSample** フォルダにコピーします。
4. **<UFT インストール・フォルダ>\dat\Extensibility\Web\Toolkits\WebExtSample** フォルダに，**Res** という名前のフォルダを作成します。
5. **WebBook.ico** ファイルを **<UFT インストール・フォルダ**

>\dat\Extensibility\Web\Toolkits\WebExtSample\Res フォルダに置きます。

ツールキット・サポート・セットをテストするには，次の手順を実行します。

1. ツールキット・サポート・セットのデプロイが完了した後，UFT を開きます。

   注: UFT は，起動時にツールキット・サポート・ファイルを読み込みます。UFT が開いている場合には，UFT を終了してから再度起動する必要があります。

   ［アドイン マネージャ］ダイアログ・ボックスで，使用可能なアドインのリストに，**WebExtSample** が Web 環境の子として表示されます（［アドイン マネージャ］ダイアログ・ボックスが開かない場合は，『HP Unified Functional Testing アドイン・ガイド』の手順を参照してください）。

2. **WebExtSample** のチェック・ボックスを選択し，［**OK**］をクリックします。UFT が開き，設計したサポートをロードします。

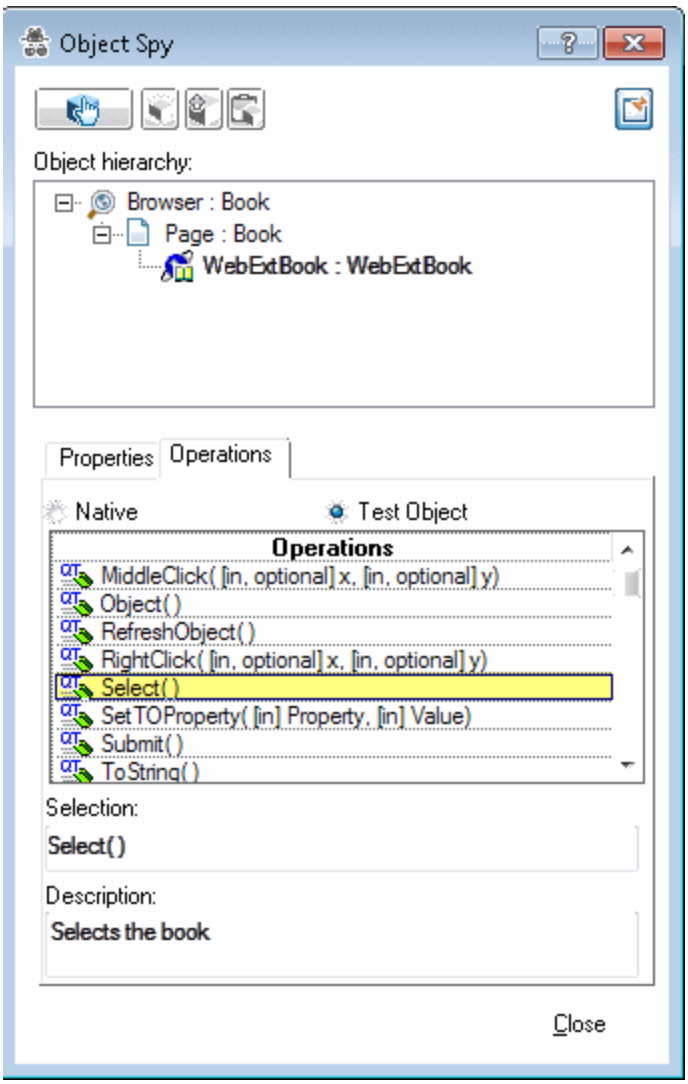
3. ［オブジェクト リポジトリ］ダイアログ・ボックスの［**新規テストオブジェクトの定義**］ボタン をクリックすると，［新規テストオブジェクトの定義］ダイアログ・ボックスが開きます。WebExtSample 環境が［**環境**］リストに表示されます。リストで WebExtSample 環境を選択すると，テスト・オブジェクト設定ファイルで定義した **WebExtBook** テスト・オブジェクト・クラスが［**クラス**］リストに表示され，**WebExtSampleTestObjects.xml** ファイルで指定したアイコンが表示されます。

4. サンプル・コントロールを実行するために，**%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample\Application\Book.htm** ファイルを開きます。

   注: UFT は，アプリケーションが開いたときにアプリケーションとの接続を確立します。したがって，Book コントロールが開いている場合には，終了してから再度起動する必要があります。

   次の操作を Book コントロールで実行し，UFT がコントロールをどのように認識するかを確認します（UFT での作業の詳細については，『HP Unified Functional Testing ユーザーズ・ガイド』を参照してください）。

- Book コントロールでサポートされる認識プロパティとテスト・オブジェクト操作は，オブジェクト・スパイで表示できます。
  - Book コントロール用に作成された **WebExtBook** テスト・オブジェクトには，テスト・オブジェクト・クラスの名前が付いています。このレッスンの後のほうで，ツールキット・サポート・セットをカスタマイズして，わかりやすい名前を指定します。
  - テスト・オブジェクト操作のリストには，**WebElement** 基本クラスから継承したすべての操作（メソッドとプロパティ）と，**WebExtSampleTestObjects.xml** テスト・オブジェクト設定ファイルで定義したすべてのメソッドが含まれます。



  - 認識プロパティのリストには，**WebExtSampleTestObjects.xml** テスト・オブジェクト設定ファイルで定義したプロパティがすべて表示されます。アプリケーションからプロパティ値を戻すメソッドの実装が完了していないので（さらに，**WebElement** 基本クラスではこのプロパティをサポートしないので），プロパティの値は表示されません。このレッスンの後のほうで，そのようなメソッドを実装します。

- Book コントロールの学習には，［オブジェクト リポジトリ］ダイアログ・ボックスの［**ローカルへオブジェクトを追加**］ボタン を使用します。オブジェクト・リポジトリでテスト・オブジェクトを表すのに正しいアイコンが使用されているのを確認してください。
- キーワード・ビューで，オブジェクト・リポジトリの **WebExtBook** オブジェクトを［**項目**］カラムで選択してテスト・ステップを作成します。
  - ［**操作**］カラムで使用可能な操作のリストは，テスト・オブジェクト設定ファイルの定義を反映しています。
  - 操作を選択すると，操作の引数の Name 属性が［**値**］セルのツールヒントに表示されます。操作の引数が複数ある場合，引数の数に応じて［**値**］セルが分割されます。
  - テスト・オブジェクト設定ファイルでテスト・オブジェクト・メソッドに対して定義した説明の文字列はツールヒント，注釈の文字列は［**注釈**］カラムに表示されます。
- エディタを開き，WebExtBook テスト・オブジェクトを使用するテスト・ステップを作成します。ステートメントの自動補完機能により，テスト・オブジェクト設定ファイルの内容に応じて，テスト・オブジェクトで使用可能な操作がすべて表示されます。

  操作の引数で指定可能な値が定義されている場合は（ListOfValues 要素），これもステートメントの自動補完機能によって表示されます。このレッスンの後のほうで，コントロールで動的な引数値リストを表示する方法を学習します。**GoToAuthorPage** 操作の **AuthorName** 属性で使用可能な著者リストを表示するサポートを作成します。

5. WebExtBook テスト・オブジェクト上で新しく作成したテスト・オブジェクト・メソッドを実行するステップを含むテストを実行します。UFT は，コントロール上でテスト・オブジェクト・メソッドを実行する JavaScript 関数を検索します。テスト・オブジェクト・メソッド実行のサポートを実装していないので，実行時エラーが発生します。このサポートの実装は，次の項で行います。

第 4 段階：WebExtBook テスト・オブジェクト・メソッドのサポートを実装

UFT がカスタム・コントロールを認識できるようにしたら，テスト・オブジェクト・メソッドの実行のサポートを作成する必要があります。そのためには，テスト・オブジェクト設定ファイルで定義したテスト・オブジェクト・メソッドごとに，コントロールでのステップ実行に必要な JavaScript 関数を記述します。

ツールキット設定ファイルで，UFT が JavaScript 関数を探すために使用する JavaScript ファイルと，オプションで各テスト・オブジェクト・メソッドに対して使用する関数の名前を指定します。

この項では，WebExtBook のテスト・オブジェクト・メソッドに対するサポートを作成します。**Select**，**GoToAuthorPage (**AuthorName**)**，**GoToUsedBooksPage** です。

ツールキット設定ファイルでは，各テスト・オブジェクト・メソッドで使用する JavaScript ファイルと関数を指定することが可能です。ただしこのレッスンでは，テスト・オブジェクト・メソッドを実行するサポートをできるだけ簡単な方法で作成する方法を紹介します。**Control** 要素ごとに JavaScript ファイルを 1 つ定義します。UFT は，各要素内で定義されたすべてのテスト・オブジェク

ト・メソッドに対して標準でこのファイルを使用し，JavaScript 関数名については，UFT は標準でテスト・オブジェクト・メソッドの名前と同じ名前を持つ JavaScript 関数を，指定のファイル内で検索します。したがって，ツールキット設定ファイル内で関数名を指定する必要はなく，作成した JavaScript 関数に正しい名前を付けるだけで問題ありません。

WebExtBook テスト・オブジェクト・メソッドのサポートを作成するには，次の手順で行います。

1. **WebExtSample.xml** ファイルで，**WebExtBook** テスト・オブジェクト・クラスに対して定義されている **Control** 要素内に，次の **Settings** 要素を追加します。

```
<Settings>
  <Variable name="default_imp_file" value="WebExtBook.js"/>
</Settings>
```

これにより，UFT は JavaScript 関数を **WebExtBook.js** ファイル（**<UFT インストール・フォルダ>\dat\Extensibility\Web\Toolkits\WebExtSample** フォルダ）で検索します。

注: ツールキット・サポート・セット・フォルダにある **WebExtSample.xml** ファイルを変更してからテストのために UFT にデプロイする方法と，**<UFT インストール・フォルダ>\dat\Extensibility\Web\Toolkits\WebExtSample.xml** を直接変更する方法があります。

2. 「第 1 段階：ツールキット・サポート・セットの作成」(90ページ)で作成した **WebExtBook.js** ファイルに次のテキストを貼り付けて，**Select**，**GoToAuthorPage (**AuthorName**)**，**GoToUsedBooksPage** の各テスト・オブジェクト・メソッド用の JavaScript 関数を作成します。

注: _elem オブジェクトは，現在処理中の HTML コントロールを参照するために UFT が使用する予約オブジェクトです。

```
// Run implementation
// ~~~~~~~~~~~~~~~~~~~~~~~~~~~~
// This section contains the functions that carry out the
// test object methods.
function Select()
{// Click the link in the second cell of the first row.
    _elem.rows[0].cells[1].children[0].click();
}
function GoToAuthorPage(AuthorName)
{   // Look for the specified author name among the children
    // of the first cell in the second row and click it.
    var bWasFound = false;
    for( var i = 0 ; i &lt; _elem.rows[1].cells[0].children.length ; ++i )
   {
       if( _elem.rows[1].cells[0].children[i].innerText == AuthorName )
      {
          _elem.rows[1].cells[0].children[i].click();
          bWasFound = true;
          break;
```

```
      }
    }
    if( bWasFound == false )
        throw ("Author name not found !");
}
function GoToUsedBooksPage()
{// Click the link in the first cell of the third row.
    _elem.rows[3].cells[0].children[1].click();
}
```

ツールキット・サポート・セットのデプロイとテスト（第 4 段階）

テスト・オブジェクト・メソッドを実行するためのサポートの作成が完了したら，更新したツールキット・サポート・セットを UFT にデプロイし，テストします。

テスト・オブジェクト・メソッド実行のサポートをテストするには，次の手順で行います。

1. 更新したツールキット・サポート・セットを UFT にデプロイするには，**WebExtBook.js** ファイル（および必要に応じて **WebExtSample.xml**）を **<UFT インストール・フォルダ >\dat\Extensibility\Web\Toolkits\WebExtSample** にコピーします。
2. UFT を閉じてから再び開きます。［アドイン マネージャ］ダイアログ・ボックスで **WebExtSample** のチェック・ボックスを選択し，［**OK**］をクリックします。UFT が開き，設計したサポートがロードされます。
3. サンプル・コントロールを終了してから再度実行します。
4. Book コントロールの学習には，GUI テストを開き，［オブジェクト リポジトリ］ダイアログ・ボックスの［**ローカルへオブジェクトを追加**］ボタン を使用します。
5. 次のステップを含むテストを作成し，テストを実行します。

```
Browser("Book").Page("Book").WebExtBook("WebExtBook").GoToAuthorPage "Jane Doe"
```

> **注: GoToAuthorPage** テスト・オブジェクト・メソッドを実行する際，コントロールに存在しない著者名をコントロールに入力すると，JavaScript 関数で例外が発生し，UFT で実行時エラー・メッセージが表示されてテスト・ステップは失敗します。

Select および **GoToUsedBooksPage** テスト・オブジェクト・メソッドをテストするには，同様のテストを作成します。

第 5 段階：WebExtBook の認識プロパティのサポートを実装

この項では，テスト実行中に認識プロパティの値を取得するためのサポートを実装します。UFT は，認識プロパティの実行時の値を，**GetROProperty** などのさまざまな標準テスト・オブジェクト・メソッドで使用します。認識プロパティの実行時の値は，チェックポイントと出力値の作成機能など，さまざまな基本機能を実行する際にも必要になります。

認識プロパティの実行時の値を提供するには，**PropertyName** パラメータを受け取り，UFT が要求したプロパティの値を返す JavaScript 関数を実装する必要があります。このメソッドを実装すると，テスト・オブジェクト設定ファイルで定義した各認識プロパティの値を返すことができます。

ツールキット設定ファイルでは，プロパティの値を取得する JavaScript 関数が実装されている JavaScript ファイルを指定できます。また，この目的のために実装した関数の名前を指定することもできます。ただし，関数名を指定しないと，UFT は **get_property_value (PropertyName)** を呼び出すので，この関数の実装が必要になります。ファイル名を指定しないと，UFT は **Control\Settings** 要素で指定した JavaScript ファイルから関数を呼び出します。したがってこのレッスンでは，**get_property_value** 関数を **WebExtBook.js** ファイル内に作成します。

WebExtBook の認識プロパティの実行時の値を提供するには，次の手順で行います。

次のテキストを **WebExtBook.js** ファイルに追加します。

```
// Property retrieval implementation
// ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
// The function provides values for all of the identification
// properties defined in the test object configuration XML file.
function get_property_value(prop)
{
   if ( prop == "title" )
   // For the "title" identification property,
   // Return the inner text of the second cell in the first row
   {
      return _elem.rows[0].cells[1].innerText;
   }
   if ( prop == "authors" )
   // To return a list of all the authors, look for all the
   // children of the first cell in the second row.
   {
      var BookAuthors = "";
      var AuthorsCount = 0;
      for( var i = 0 ; i < _elem.rows[1].cells[0].children.length ; ++i )
      {
         if( _elem.rows[1].cells[0].children[i].tagName == "A" )
       {
            if( AuthorsCount > 0 )
```

```
            BookAuthors += ",";
            BookAuthors += _elem.rows[1].cells[0].children[i].innerText;
            AuthorsCount++;
        }
      }
       return BookAuthors;
   }
   if ( prop == "price" )
   // To return the price of the book, return the innerText
   // property of the first cell in the fourth row.
   {
      return _elem.rows[3].cells[0].children[0].innerText;
   }
    if ( prop == "min_used_price" )
   // To return the lowest price available for a used copy of the
   //  book, return the innerText property of the second child of
   // the first cell in the fourth row.
   {
      if( _elem.rows[3].cells[0].children.length > 2 )
         return _elem.rows[3].cells[0].children[2].innerText;
   }
}
```

注: ツールキット・サポート・セット・フォルダにある **WebExtBooks.js** ファイルを変更してから UFT にデプロイする方法と，**<UFT インストール・フォルダ >\dat\Extensibility\Web\Toolkits\WebExtSample\WebExtBook.js** を直接変更する方法があります。

ツールキット・サポート・セットのデプロイとテスト（第 5 段階）

認識プロパティの実行時の値の取得のサポートを作成したら，更新したツールキット・サポート・セットを UFT にデプロイしてテストします。

認識プロパティの実行時の値の取得のサポートをテストするには，次の手順で行います。

1. 最新のファイルが **<**UFT **インストール・フォルダ >\dat\Extensibility\Web\Toolkits\WebExtSample** に格納されていることを確認します。
2. UFT を閉じてから再び開きます。［アドイン マネージャ］ダイアログ・ボックスで **WebExtSample** のチェック・ボックスを選択し，［**OK**］をクリックします。UFT が開き，設計したサポートがロードされます。
3. サンプル・コントロールを終了してから再度実行します。
4. GUI テストを新規作成し，WebExtBook 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て，このテスト・オブジェクトを使用するテスト・ステップを作成します。オブジェクト

を右クリックして［**標準チェックポイントの挿入**］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。テスト・オブジェクト設定ファイルで定義した認識プロパティ（**title**，**authors**，**price**，**min_used_price**）がプロパティ・リストに表示されていて，選択されている状態であることを確認します。

5. それぞれの認識プロパティを取得し，値をチェックするかメッセージ・ボックスで表示するテストを作成して実行します。たとえば，次のようなテストを実行します。

Item	Operation	Value
Action1		
Book		
Book		
WebExtBook	CheckProperty	"authors","Jane Doe,John Doe"
WebExtBook	Check	CheckPoint("WebExtBook")
Function Call	MsgBox	Browser("Book").Page("Book").WebExtBook("WebExtBook").GetROProperty("title")

最初のステップで，authors プロパティの値をチェックします。2 番目のステップで，チェックポイントで選択したプロパティ（ここでは price と min_used_price）をチェックします。3 番目のステップで，本の題名をメッセージ・ボックスに表示します。

6. ［**OK**］をクリックし，メッセージ・ボックスを閉じます。テストが完了すると，結果が表示されます。実行結果のツリーを展開すると，ステップの詳細情報が表示されます。

第 6 段階：テスト・オブジェクトの名前の変更

この項では，ツールキット・サポート・セットを変更します。これにより，「Web Add-in Extensibility Book サンプル・ツールキットのサポートの計画」(82ページ)で行った計画に従って，WebExtBook テスト・オブジェクトの名前に本の題名を割り当てます。

UFT は，コントロールを表すテスト・オブジェクトを作成する際，**get_property_value** 関数（テスト・オブジェクト認識プロパティの値を取得）を引数 **logical_name** で呼び出し，テスト・オブジェクトの名前を決定します。したがって，**get_property_value** 関数を実装することにより，UFT がテスト・オブジェクトに割り当てる名前をカスタマイズできます。**get_property_value** 関数が **logical_name** プロパティをサポートしない場合には，テスト・オブジェクトにはテスト・オブジェクト・クラスの名前が割り当てられます（1 つのページに同じテスト・オブジェクト・クラスを持つオブジェクトが複数存在する場合は，インデックスが付加されます）。

テスト・オブジェクトの名前をカスタマイズするには，次の手順で行います。

WebExtBook.js ファイルの **get_property_value** 関数には次の行があります。

```
if ( prop == "title" )
// For the "title" identification property,
```

これを，次の行で置き換えます。

```
if ( prop == "logical_name" || prop == "title" )
// For the "title" identification property, as well as
```

```
// the "logical_name" property
```

注: ツールキット・サポート・セット・フォルダにある **WebExtBooks.js** ファイルを変更してから UFT にデプロイする方法と，**<UFT インストール・フォルダ>\dat\Extensibility\Web\Toolkits\WebExtSample\WebExtBook.js** を直接変更する方法があります。

上記の変更により，**get_property_value** 関数は，UFT がテスト・オブジェクトの名前として使用する **logical_name** プロパティのテキストを返します。これは，**title** 認識プロパティと同じテキストです（上記の変更内容を説明するために，この関数のコメントを変更します。コメントの最後に，次の内容を追加してください。「as well as the hard coded "logical_name" property that UFT uses to name the test object.」）。

ツールキット・サポート・セットのデプロイとテスト（第 6 段階）

コントロールを表すテスト・オブジェクトに名前を割り当てるサポートの作成が完了したら，更新したツールキット・サポート・セットを UFT にデプロイしてテストします。

WebExtBook テスト・オブジェクトに名前を割り当てるサポートをテストするには，次の手順で行います。

1. 最新のファイルが **<UFT インストール・フォルダ>\dat\Extensibility\Web\Toolkits\WebExtSample** に格納されていることを確認します。
2. UFT を閉じてから再び開きます。［アドイン マネージャ］ダイアログ・ボックスで **WebExtSample** のチェック・ボックスを選択し，［**OK**］をクリックします。UFT が開き，設計したサポートがロードされます。
3. サンプル・コントロールを終了してから再度実行します。
4. GUI テストを開いて，オブジェクト・リポジトリを開き，［ローカルへオブジェクトを追加］ボタンをクリックして Book コントロールを学習します。UFT が作成したテスト・オブジェクトの名前が **History of QuickTest Professional** であることを確認します。

第 7 段階：子オブジェクトを学習対象から除外するフィルタを実装

UFT に Web ページを学習するように指示すると，［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，Web ページのどの子孫をページとともに学習するかを指定できます。［**すべてのオブジェクト タイプ**］を選択すると，UFT は親の Web ページと一緒に WebExtBook コントロールを学習するので，WebExtBook コントロールに含まれるすべてのコントロールもその Web ページの子（および WebExtBook コントロールの兄弟）として学習の対象になります。

ただし，Book コントロールの場合は，すべての子に対してテスト・オブジェクトを作成する必要はありません（「Web Add-in Extensibility Book サンプル・ツールキットのサポートの計画」(82ページ)を参照してください）。

Web Add-in Extensibility がサポートするコントロールの下位にあるコントロールを学習対象から除外するには，学習フィルタを定義します。JavaScript 関数を使用すれば複雑なフィルタを実装することができ，この場合は関数の格納場所と名前をツールキット設定ファイルで指定します。簡単なフィルタであればツールキット設定ファイルに直接実装できるので，JavaScript 関数は必要ありません。

Book コントロールに含まれるコントロールを学習対象から除外する処理は，簡単なフィルタで実行できます。このフィルタを実装する前に，Book コントロールとその下位コントロールがすべて含まれる Web ページを学習し，Book の子がすべて学習対象になることを確認してください。手順は，「ツールキット・サポート・セットのデプロイとテスト（第 7 段階）」(105ページ)を参照してください。

Book コントロールに含まれるコントロールを学習対象から除外するには，次の手順で行います。

WebExtSample.xml ファイルで，**WebExtBook** テスト・オブジェクト・クラスに対して定義されている **Control** 要素に，次の **Filter** 要素を追加します。

```
<Filter>
  <Learn learn_control="Yes" learn_children="No"/>
</Filter>
```

これによって UFT は，親の Web ページの学習時に WebExtBook テスト・オブジェクトを学習しますが，子のコントロールは学習しなくなります。

注: ツールキット・サポート・セット・フォルダにある **WebExtSample.xml** ファイルを変更してから UFT にデプロイする方法と，**<UFT インストール・フォルダ >\dat\Extensibility\Web\Toolkits\WebExtSample\.xml** を直接変更する方法があります。

ツールキット・サポート・セットのデプロイとテスト（第 7 段階）

子を学習対象から除外するフィルタの定義が完了したら，更新したツールキット・サポート・セットを UFT にデプロイしてテストします。

WebExtBook テスト・オブジェクトの学習で子を学習対象から除外するサポートをテストするには，次の手順で行います。

1. 最新のファイルが **<UFT インストール・フォルダ >\dat\Extensibility\Web\Toolkits\WebExtSample** に格納されていることを確認します。
2. UFT を閉じてから再び開きます。［アドイン マネージャ］ダイアログ・ボックスで **WebExtSample** のチェック・ボックスを選択し，［**OK**］をクリックします。UFT が開き，設計したサポートがロードされます。
3. サンプル・コントロールを終了してから再度実行します。

4. GUI テストを開き，オブジェクト・リポジトリを開きます。Book コントロールを含む Web ページを学習するには，［オブジェクト リポジトリ］ダイアログ・ボックスの［**ローカルへオブジェクトを追加**］ボタン を使用します。［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。
5. ［**すべてのオブジェクト タイプ**］を選択して［**OK**］をクリックします。**The History of QuickTest Professional** という名前の WebExtBook オブジェクトがオブジェクト・リポジトリに追加されますが，下位コントロールは追加されません。

第 8 段階：Book コントロール上の記録のサポートを実装

チュートリアルがここまで進んだ時点で，作成したツールキット・サポート・セットでは，すでに UFT のすべての機能が使用できます。UFT は Book コントロールを認識し，学習し，テストを実行することができます。

また，UFT でテストを作成する別の方法として，アプリケーションでユーザが実行する操作を記録する方法があります。「Web Add-in Extensibility Book サンプル・ツールキットのサポートの計画」(82 ページ)に示されているように，標準設定では，UFT は Book コントロール内のさまざまな Web リンクや画像オブジェクトに対して，単なる **Click** 操作だけを記録します。また，クリックに応じて Book コントロール自身で実行される **Select**，**GoToAuthorPage**，**GoToUsedBooksPage** の各操作を記録すれば，さらに詳細なテストの実行が可能になります。

コントロール上の記録のカスタマイズをサポートするには，関連イベントをリッスンし，各イベント発生時に記録するテスト・ステップを UFT に指示する必要があります。

この操作を行うために，次の 2 つの JavaScript 関数を記述します。

- イベントのリッスンを登録する 1 つの JavaScript 関数。これは，UFT が Web Add-in Extensibility SDK で公開している **_util** ユーティリティ・オブジェクトの **RegisterForEvent** 関数を使用して，適切な要素に対する適切なイベントのリッスンを登録します。この関数の引数によって，各イベントが発生したときに UFT が呼び出す JavaScript 関数が決まります。

  ツールキット設定ファイルで，この JavaScript 関数の名前と格納場所（オプション）を指定します。
- イベントを処理する 1 つまたは複数の JavaScript 関数。これは，**_util** ユーティリティ・オブジェクトの **Record** 関数を呼び出して，テストに追加するステップを UFT に指示します。

> **注: Record** 関数などのユーティリティ・オブジェクト関数は，**SafeArray** タイプの引数を渡す必要があります。配列を **SafeArray** に変換するには，Web Add-in Extensibility に用意されている **toSafeArray (array)** 関数を使用します。この関数は，**<Extensibility Accelerator インストール・フォルダ>\bin\PackagesToLoad\common.js**で定義されています（このファイルは **<UFT インストール・フォルダ>\dat\Extensibility\Web\Toolkits** フォルダにもあります）。

ユーティリティ・オブジェクト関数の構文の詳細については，『UFT Java Add-in Extensibility API Reference』の「**_util**」の項（『Web Add-in Extensibility ヘルプ』から利用可能）を参照してください。

Book コントロール上の記録のサポートを作成するには，次の手順で行います。

注: ツールキット・サポート・セット・フォルダにある **WebExtSample.xml** ファイルと **WebExtBook.js** ファイルを変更してから UFT にデプロイする方法と，**<UFT インストール・フォルダ>\dat\Extensibility\Web\Toolkits\WebExtSample** にあるファイルを直接変更する方法があります。

1. ツールキット設定ファイルの **Control** 要素に，次の **Record\EventListening** 要素を追加します。

```
<Record>
  <EventListening use_default_event_handling_for_children="false"
            use_default_event_handling="false"
            type="javascript"
            function="ListenToEvents"/>
</Record>
```

   これによって UFT は，Book コントロールとその下位コントロールで発生したイベントを処理する際，標準の Web イベント定義を使用するのではなく，JavaScript 関数 **ListenToEvents** を呼び出します。JavaScript ファイルの指定は行っていないので，WebExtBook テスト・オブジェクト・クラスの **Control** レベルで指定した **WebExtBook.js** ファイル内で JavaScript 関数を検索します。

2. **WebExtBook.js** ファイルに，次の **ListenToEvents** 関数を追加します。

```
function ListenToEvents( elem )
{
   // Connect to the "Select" event:When the book name or the
   // book icon is clicked, call OnSelectClicked.
   _util.RegisterForEvent( _elem.rows[0].cells[0].children[0], "onclick", "OnSelectClicked");
   _util.RegisterForEvent( _elem.rows[0].cells[1].children[0], "onclick", "OnSelectClicked");
   // Connect to the "Author" event:When an author name is
   // clicked, call OnAuthorClicked.
   for( var i = 0 ; i < _elem.rows[1].cells[0].children.length ; ++i )
   {
      if( _elem.rows[1].cells[0].children[i].tagName == "A" )
      {
         _util.RegisterForEvent( _elem.rows[1].cells[0].children[i], "onclick", "OnAuthorClicked" );
      }
   }
   // Connect to the "UsedBooks" event:When "Used" is
   // clicked, call OnUsedBooksClicked.
   if(  _elem.rows[3].cells[0].children.length > 1 )
      _util.RegisterForEvent( _elem.rows[3].cells[0].children[1], "onclick", "OnUsedBooksClicked"
);
   return true;
}
```

この関数は，本の題名，画像，著者，［**Used**］リンクで発生するクリック・イベントをリッスンするように UFT に指示します。イベントに対する登録の際に，この関数はイベント発生時に UFT から呼び出される JavaScript 関数を指定します。

3. **WebExtBook.js** ファイルに，次のイベント・ハンドラ JavaScript 関数を追加します。

```
function OnSelectClicked( handlerParam , eventObj )
{
   // Record the "Select" step
   var arr = new Array();
   _util.Record( "Select", toSafeArray(arr) , 0 );
   return true;
}
```

```
function OnAuthorClicked( handlerParam , eventObj )
{
   // Record the "GoToAuthorPage" step
   var arr = new Array();
   arr[0] = eventObj.srcElement.innerText;
   _util.Record( "GoToAuthorPage", toSafeArray(arr) , 0 );
   return true;
}
```

```
function OnUsedBooksClicked( handlerParam , eventObj )
{
   // Record the "GoToUsedBooksPage" step
   var arr = new Array();
   _util.Record( "GoToUsedBooksPage", toSafeArray(arr) , 0 );
   return true;
}
```

これらの関数は，「Web Add-in Extensibility Book サンプル・ツールキットのサポートの計画」(82ページ) の計画に従って，WebExtBook テスト・オブジェクトに対して **Select**，**GoToAuthorPage**，**GoToUsedBooksPage** を記録します。

ツールキット・サポート・セットのデプロイとテスト（第 8 段階）

Book コントロール上の記録のサポートの作成が完了したら，更新したツールキット・サポート・セットを UFT にデプロイしてテストします。

Book コントロールで実行される操作の記録のサポートをテストするには，次の手順で行います。

1. 最新のファイルが **<UFT インストール・フォルダ>\dat\Extensibility\Web\Toolkits\WebExtSample** に格納されていることを確認します。
2. UFT を閉じてから再び開きます。［アドイン マネージャ］ダイアログ・ボックスで

WebExtSample のチェック・ボックスを選択し，［**OK**］をクリックします。UFT が開き，設計したサポートがロードされます。

3. サンプル・コントロールを終了してから再度実行します。
4. GUI テストを開き，［**記録**］ボタンをクリックするか，［**記録**］ > ［**記録**］を選択します。Book コントロール内のさまざまなリンクをクリックします（クリックした後は，Book コントロールに戻るために前のページに戻る必要があります）。本の題名，コントロール内の画像，著者名，［**Used**］リンクなどをクリックしてみます。

   クリックするたびに，新しいステップがテストに追加されます。

Item	Operation	Value	Documentation
Action1			
Book			
Book			
The History of QuickTest	Select		Select the "The History of QuickTest" book.
Not Implemented	Sync		Wait for the Web page to synchronize before continuing
Book	Back		Navigate back to the previous page of the browser.
Book			
The History of QuickTest	Select		Select the "The History of QuickTest" book.
Not Implemented	Sync		Wait for the Web page to synchronize before continuing
Book	Back		Navigate back to the previous page of the browser.
Book			
The History of QuickTest	GoToAuthorPage	"Jane Doe"	Open the Web page for "Jane Doe".
Not Implemented_2	Sync		Wait for the Web page to synchronize before continuing
Book	Back		Navigate back to the previous page of the browser.
Book			
The History of QuickTest	GoToUsedBooksPage		Open the "The History of QuickTest" used books page.

   ［**停止**］ボタンをクリックするか，［**記録**］ > ［**停止**］を選択して記録セッションを終了します。

第 9 段階：AuthorName の動的な値リストのサポートを実装

Web Add-in Extensibility を使用することにより，テスト・オブジェクト・メソッドの引数として使用可能な値のリストを，コントロールの実行時の値に基づいて UFT ユーザに提供することができます。たとえば，**WebExtBook** テスト・オブジェクト・クラスの **GoToAuthorPage** テスト・オブジェクト・メソッドは，**AuthorName** 引数を受け取ります。UFT ユーザの操作を考えた場合，著者名を入力するのではなくリストから著者名を選択できれば操作は簡単になります。ただし，このリストは固定ではなく，**WebExtBook** コントロールによって変動します。

テスト・オブジェクト設定ファイルで **AuthorName** 引数の **DynamicListOfValues** 属性を **true** に設定すると，UFT はテスト・ステップの作成時に，使用可能な引数値のリストをコントロールに要求します。

ツールキット設定ファイルで，値のリストを取得するために UFT が呼び出す JavaScript 関数のファイル名と関数名を指定できます。標準設定では，UFT は WebExtBook テスト・オブジェクト・クラスの **Control** レベルで指定した**WebExtBook.js** ファイルから，JavaScript 関数 **get_list_of_values** を呼び出して，値リストを要求します。UFT は，テスト・オブジェクト設定ファイルで **DynamicListOfValues**

属性が **true** に設定されているすべての引数に対して，この JavaScript 関数を呼び出します。この関数に渡されるパラメータには，値を要求するテスト・オブジェクト・メソッドと引数を指定します。

この項では，JavaScript 関数 **get_list_of_values** を実装し，Book コントロールから著者名を返します。

GoToAuthorPage テスト・オブジェクト・メソッドの AuthorName 引数について，使用可能な値の動的リストを取得するには，次の手順で行います。

WebExtBook.js ファイルに，次の JavaScript 関数を追加します。

```
// Dynamic list of values implementation
// ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
function get_list_of_values( method, argIndex )
{
   // When creating a step with the GoToAuthorPage test
   // object method, provide a list of the authors of this book
   // that can be used for the method's argument.
   if (method == "GoToAuthorPage")
   {
      return get_GoToAuthorPage_list_of_values(argIndex);
   }
   return null;
}
function get_GoToAuthorPage_list_of_values(argIndex)
{
   var arr = new Array();
   if( argIndex > 1 )
      return toSafeArray(arr);
   // Retrieve all authors
   var AuthorsCount = 0;
   for( var i = 0 ; i < _elem.rows[1].cells[0].children.length ; ++i )
   {
      if( _elem.rows[1].cells[0].children[i].tagName == "A" )
      {
         arr[AuthorsCount]="\""+_elem.rows[1].cells[0].children[i].innerText+"\"";
         AuthorsCount++;
      }
   }
   return toSafeArray(arr);
}
```

これにより，本の著者名リストが返されます。著者名は引用符で囲まれます。

以上で，カスタム・サポートの計画時の仕様に従って，Book カスタム・コントロールが完全にサポートされました。

注: ツールキット・サポート・セット・フォルダにある **WebExtSample.xml** ファイルを変更してから UFT にデプロイする方法と，**<UFT インストール・フォルダ**

>\dat\Extensibility\Web\Toolkits\WebExtSample\.xml を直接変更する方法があります。

ツールキット・サポート・セットのデプロイとテスト（第 9 段階）

JavaScript 関数 **get_list_of_values** の実装が完了したら，更新したツールキット・サポート・セットを UFT にデプロイし，著者名の動的リストが正しく提供されることを確認します。

Book コントロールで実行される操作の記録のサポートをテストするには，次の手順で行います。

1. 最新のファイルが **<UFT インストール・フォルダ >\dat\Extensibility\Web\Toolkits\WebExtSample** に格納されていることを確認します。
2. UFT を閉じてから再び開きます。［アドイン マネージャ］ダイアログ・ボックスで **WebExtSample** のチェック・ボックスを選択し，［**OK**］をクリックします。UFT が開き，設計したサポートがロードされます。
3. サンプル・コントロールを終了してから再度実行します。
4. エディタで GUI テストを開き，**WebExtBook** テスト・オブジェクトと **GoToAuthorPage** テスト・オブジェクト・メソッドでステップを作成します。メソッド名の後に空白文字を入力すると，Book コントロールのメソッド引数で使用できる著者名リストがドロップダウン・リストで表示されます。

レッスンのまとめ

このレッスンでは，認識プロパティとテスト・オブジェクト・メソッドを定義して，新しいテスト・オブジェクト・クラス（WebExtBook）を作成しました。また，Book コントロールのサポートを作成し，UFT で WebExtBook テスト・オブジェクトとして認識されるようにしました。

- テスト・オブジェクト設定ファイルについて学習しました。
- ツールキット設定ファイルの理解について学習しました。
- 新しい認識プロパティとテスト・オブジェクト・メソッドのサポートについて学習しました。
- 子コントロールを学習対象から除外するフィルタの作成について学習しました。
- 記録のサポートと，**Record**，**RegisterForEvent** の各ユーティリティ・メソッドの使用について学習しました。
- テスト・オブジェクト引数で使用可能な値の動的リストの表示について学習しました。

その他の情報

ツールキット・サポート・セットの構造と内容の詳細は，「ツールキットのサポートの開発」(31 ページ)を参照してください。

テスト・オブジェクト設定ファイルの構造と内容の詳細については，『UFT Test Object Schema Help』（Web Add-in Extensibility ヘルプから利用可能）を参照してください。

ツールキット設定ファイルの構造と内容の詳細については，『UFT Java Add-in Extensibility Toolkit Configuration Schema Help』（『Web Add-in Extensibility ヘルプ』から利用可能）を参照してください。

_util ユーティリティ・オブジェクトとグローバル JavaScript メソッドの詳細については，『UFT Java Add-in Extensibility API Reference』（『Web Add-in Extensibility ヘルプ』から利用可能）を参照してください。

次のレッスンでは，UsedBooks カスタム・コントロールのサポートを作成する方法を学習します。UsedBooks コントロールを表すテスト・オブジェクト・クラスは，既存の WebTable テスト・オブジェクト・クラスを拡張します。このコントロールの UFT サポートを作成する作業を通して，Web Add-in Extensibility で提供されるさらに高度なオプションの使用方法をいくつか学習します。

第6章: 複雑な Web カスタム・コントロールに対する UFT サポートの作成方法の学習

このレッスンでは，Extensibility Accelerator for HP Functional Testing と一緒にインストールされる Web Add-in Extensibility Book サンプル・ツールキットの中の UsedBooks コントロールのサポートを作成します。UsedBooks コントロールを表すテスト・オブジェクト・クラスは，既存の WebTable テスト・オブジェクト・クラスを拡張します。UsedBooks コントロールのサポートの作成を通じて，Web Add-in Extensibility で提供されている高度なオプションの使用方法を学習します。

レッスン「簡単な Web カスタム・コントロールに対する UFT サポートの作成方法の学習」(81ページ)では，シンプルなカスタム・コントロールのサポートの作成方法について学習しました。Web Add-in Extensibility の基本事項については習得済みであるため，このレッスンでは，より詳細な内容についてのみ説明します。

%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample フォルダにこのサンプルで使用する完全なツールキット・サポート・セットが含まれているので，参考にしてください。サンプル・サポート・セットは Firefox 上でも Internet Explorer 上と同様に動くように設計されていて，**jQuery** JavaScript ライブラリを使用しているので，ここで作成する JavaScript コードと同じものではありません。

このレッスンの内容

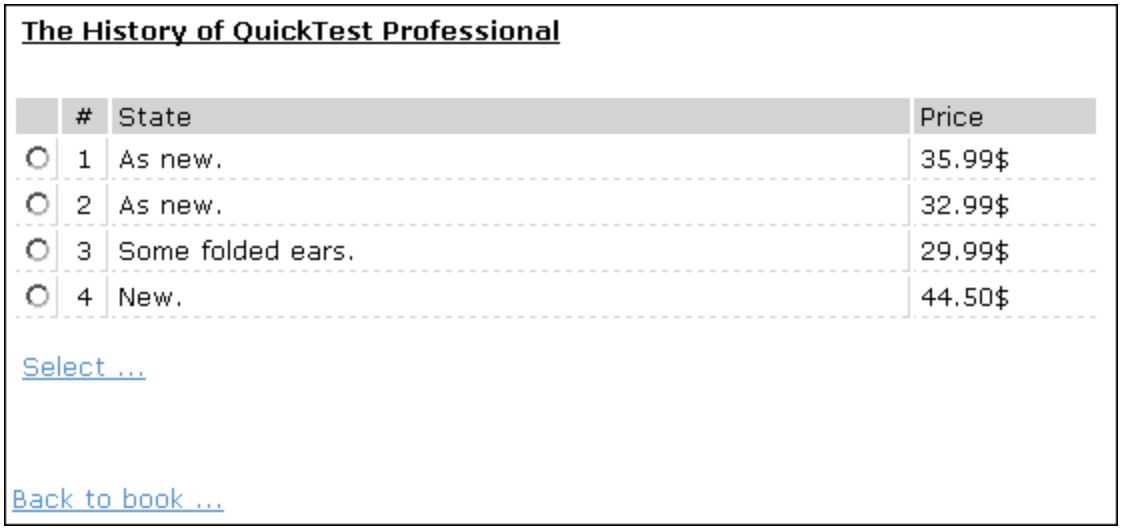
- このレッスンの準備 114
- Web Add-in Extensibility のサンプル・コントロール UsedBooks のサポートの計画 114
- ツールキット・サポート・セットの開発 121
- レッスンのまとめ 134

このレッスンの準備

カスタム・コントロールの UFT サポートを拡張する作業には，ソース・ファイルへのアクセス権限が必要です。UFT でカスタム・コントロールをサポートするためにコントロールのソースを変更する必要はありませんが，ソースを理解しておく必要はあります。コントロールを構成する要素と属性，コントロールで発生する可能性のあるイベントなどに関する情報を準備しておいてください。この情報は，サポートの設計で必要になります。

UsedBooks コントロールのソース・ファイルは，**%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample\Application\UsedBooks.htm** ファイルを開きます。

サンプル・アプリケーションを実行するには，**%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample\Application\Book.htm** ファイルを開きます。これにより，Book コントロールが開きます。Book コントロールで［**Used**］をクリックすると，UsedBooks コントロールが実行されます。



コントロールを実行し，ソース・ファイルを開いてコントロールの動作と実装を確認します。

UsedBooks コントロールは，この本の古本の情報とラジオ・ボタンが格納された Web テーブルを構成される **div** 要素と，リストから本を選択できる［**Select**］リンク（table 要素の外側）として実装されます。本を選択してその本に関するページを開くには（このサンプルでは実装されていません），テーブルの該当する行にあるラジオ・ボタンを選択してから［**Select**］をクリックします。

Web Add-in Extensibility のサンプル・コントロール UsedBooks のサポートの計画

ここでは，UFT による UsedBooks コントロールの認識について，現在の方法とカスタマイズ後の方法を比較検討します。次に，「Web Add-in Extensibility 計画の理解チェックリスト」(27ページ)の質問

に回答し，「Web Add-in Extensibility の計画チェックリスト」(29ページ)に記入します。

この一連の作業を効率的に行うためには，UFT が UsedBooks コントロールを認識する方法をオブジェクト・スパイ，キーワード・ビュー，記録オプションで確認し，コントロールの実装方法と用途を検討する必要があります。

1. **UFT を開いて UsedBooks コントロールを実行します。**

   UFTを開いて Web Add-in をロードします。

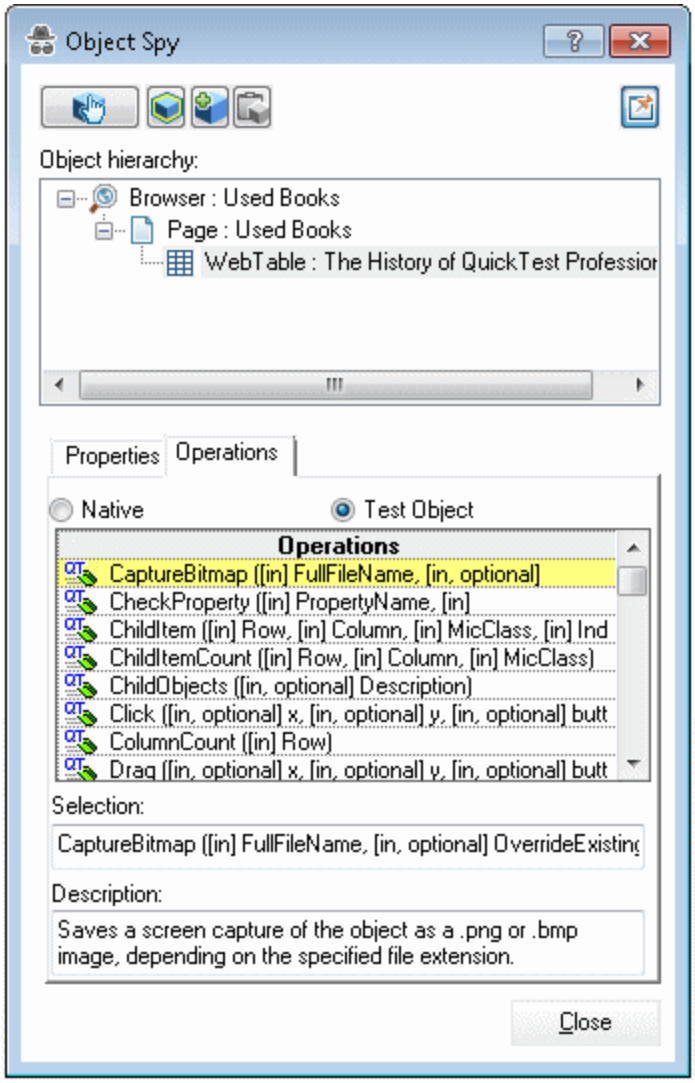
   UsedBooks コントロールの開いているインスタンスをすべて開じて，開きなおします。これを行うには，**%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample\Application\Book.htm** ファイルを開き，Book コントロールが開いたら［**Used**］をクリックします。

2. **オブジェクト・スパイを使用して UsedBooks テスト・オブジェクトの操作を表示します。**

   UFT で，GUI テストを開き，［**ツール**］ > ［**オブジェクト スパイ**］を選択するか，［**オブジェクト スパイ**］ツールバー・ボタンをクリックして，［**オブジェクト スパイ**］ダイアログ・ボックスを開きます。［**操作**］タブをクリックし，［**テスト オブジェクト操作**］を選択します。

   ［オブジェクト スパイ］ダイアログ・ボックスで，指差しアイコンをクリックしてから，UsedBooks テーブルをクリックします。

UsedBooks コントロールは Web テーブルを含みますが，これの UFT サポートは組み込まれています。したがって，テーブル内の題名が名前として割り当てられた **WebTable** としてコントロールが認識されます。テスト・オブジェクトに使用されるアイコンは，標準の WebTable クラスのアイコンです。UFT は **div** 要素を無視します。この要素は，UsedBooks コントロールのルートです。



オブジェクト・スパイを閉じます。

3. **UsedBooks コントロール上の操作を記録します。**

   UFT で，［**実行**］ > ［**実行の設定**］または［**記録**］ > ［**記録の設定**］を選択して，［記録と実行環境設定］ダイアログ・ボックスを開きます。［Web］タブで，［**任意の開いているブラウザを対象にテストを記録および実行**］を選択します。［**OK**］をクリックします。

   ［**記録**］ボタンをクリックするか，［**記録**］ > ［**記録**］を選択します。UsedBooks テーブルでラジオ・ボタンを 1 つ選択し，［**Select**］をクリックします。

クリックするたびに，新しいステップがテストに追加されます。

Item	Operation	Value	Documentation
Action1			
Used Books			
Used Books			
SelUsedBook	Select	"#0"	Select the "#0" radio button in the "SelUsedBook" radio button group.
Select ...	Click		Click the "Select ..." link.

［**停止**］ボタンをクリックするか，［**記録**］ > ［**停止**］を選択して記録セッションを終了します。

記録されたステップには，ラジオ・ボタンの選択とリンクのクリックが別々に反映されるので，ボタンの選択やクリックを UsedBooks コントロールに関連したものとして認識しません。

4. **UsedBooks コントロールが属するカスタム・ツールキットを決定します。**

   コントロールに対する UFT サポートを拡張する場合は，必ずツールキットのコンテキストで行います。このチュートリアルでは，Book と UsedBooks という 2 つの Web カスタム・コントロールを作成し，WebExtSample という名前のカスタム・ツールキットを作成します。

   前のレッスンでは，このツールキット用にツールキット・サポート・セットを作成しました。このレッスンでは，UsedBooks コントロールのサポートを WebExtSample ツールキット・サポート・セットに追加します。

5. **カスタム・コントロール・サポート計画のチェックリストを完成させます。**

   ここでは，UsedBooks コントロールのサポートを計画する際に決定すべき項目について説明し，サポート計画チェックリストを使って各項目をまとめます。

   a. カスタム・コントロールを表すテスト・オブジェクト・クラスを選択します。

      UsedBooks コントロールの内部コンテンツは，それが含む情報のタイプに従って Web テーブル・コントロールとして実装されます。UsedBooks コントロールでテストを実行して内容をチェックするという目的に限れば，UFT でこのコントロールをテーブルとして認識する方が適切ですが，UsedBooks コントロールのサポートを最適化するには，このコントロールを表すテスト・オブジェクトにおいて，テーブル行のラジオ・ボタンを選択して［**Select**］をクリックすることによって本を選択する **SelectBook** テスト・オブジェクト・メソッドをサポートする必要があります。

      さらに，UsedBooks テーブルの先頭行にはカラム名が格納されているので，WebTable オブジェクトでサポートされる **RowCount** テスト・オブジェクト・メソッドを置換（またはオーバーライド）することにより，テーブルの行数を減らして入手可能な古本の冊数を返すこともできます。**SelectBook** テスト・オブジェクト・メソッドをサポートし，**RowCount** の実装をオーバーライドするには，新しいテスト・オブジェクト・クラスを作成して **WebExtUsedBooks** という名前を付け，**WebTable** を拡張します。次に，このテスト・オブジェクト・クラスを UsedBooks コントロールを表すクラスとして認識するように UFT に指示します。

b. UFT がコントロールを表すのに使用するテスト・オブジェクト・クラスを識別する方法を次のように定義します。

   次の条件が満たされる場合，**WebExtUsedBooks** テスト・オブジェクトを使用してそのコントロールを表します。

   - コントロールの **tagName** プロパティが **div**。
   - コントロールの最初の子の **tagName** プロパティが **table**。
   - コントロールの最初の子の **className** プロパティが **UsedBooks**。

c. 新しいテスト・オブジェクト・クラスの詳細を決定します。

   - 新しいテスト・オブジェクト・クラスは，次のアイコン・ファイルで表されます。**<UFT インストール・フォルダ>\dat\Extensibility\Web\Toolkits\WebExtSample\Res\WebBookList.ico**
   - ヘルプ・ファイルは提供されません。
   - **WebExtUsedBooks** テスト・オブジェクト・クラスは，**title** 認識プロパティ（コントロールを一意に認識し，［チェックポイントのプロパティ］ダイアログ・ボックスで標準で選択され，スマート認識では使用されない）をサポートする必要があります。
   - テスト・オブジェクト自身には，**title** 認識プロパティと同じ名前を付けてください。

d. カスタム・コントロールをサポートするテスト・オブジェクト・メソッドを決定します。

   WebExtUsedBooks テスト・オブジェクト・クラスは，**WebTable** テスト・オブジェクト・クラスがサポートするテスト・オブジェクト操作をすべてサポートする必要があります。さらに，**SelectBook** テスト・オブジェクト・メソッドのサポートも必要です。

   - 基本クラスである WebTable から継承された操作はすべて，UsedBooks コントロールに含まれる **table** 要素でサポートできます。ただし，**table** 要素は UsedBooks コントロールのルート要素ではないので，UFT はこの要素を基本要素としては認識しません。したがって，UsedBooks コントロールの基本要素として **table** 要素を返す JavaScript 関数を実装する必要があります。この実装によって UFT は，**table** 要素を使用して WebTable 基本テスト・オブジェクト・クラスから継承された操作をサポートします。
   - **SelectBook** テスト・オブジェクト・メソッドは，各本に対応するラジオ・ボタンを選択して［**Select**］をクリックする操作をシミュレートします。
   - WebTable テスト・オブジェクト・メソッドである **RowCount** は，オーバーライドすることにより，テーブルの行数ではなく実際の冊数を返すことができます。

e. このコントロールの学習時に UFT の学習対象とするコントロールの子を定義します。

   このチュートリアルでは，**WebExtUsedBooks** テスト・オブジェクトが Web ページの一部として学習される際に，ラジオ・ボタンも学習されます。

f. **WebExtUsedBooks** テスト・オブジェクトをオブジェクト・スパイで表示するかどうかを決定します。表示します。

g. 記録をサポートするかどうかと，記録の対象となるイベントを定義します。

   ［**Select**］リンクで発生するマウス・クリックをリッスンします。ラジオ・ボタンを選択してこのリンクをクリックすると，そのラジオ・ボタンに対応する本を選択するテスト・ステップが記録されます。

h. サポートのどの部分をツールキット設定ファイルで設計し，どの部分に JavaScript 関数が必要かを決定します。
   - UsedBooks コントロールでは，ツールキット設定ファイルで指定した JavaScript 関数によってテスト・オブジェクトが認識されます。したがって，不要な JavaScript 関数呼び出しを避けるために，Conditions 要素をツールキット設定ファイルで定義します。これにより，コントロールが **div** 要素として定義されている場合にのみ，UFT は JavaScript 関数を呼び出します。
   - **table** 基本要素は，ツールキット設定ファイルで指定した JavaScript 関数によって返されます。
   - JavaScript 関数は，標準の命名規則でテスト・オブジェクトの認識プロパティをサポートできるので，ツールキット設定ファイルの変更は不要です。
   - WebTable の **RowCount** テスト・オブジェクト・メソッドは，**BookCount** という名前の JavaScript が提供する新しい実装によってオーバーライドされます。したがって，関数の名前をツールキット設定ファイルで指定しておく必要があります。
   - UsedBooks コントロールによる学習対象となる子のフィルタ処理は JavaScript 関数の呼び出しで実行するので，この関数をツールキット設定ファイルで指定しておく必要があります。
   - 記録をサポートするには，ツールキット設定ファイルにおいて，標準の Web イベントの設定を無効にし，JavaScript 関数を指定して UFT が正しいイベントをリッスンするようにします。さらに，イベント登録を処理する JavaScript 関数と，イベント発生時に関連ステップを記録するように UFT に指示する JavaScript 関数を作成します。

次の「チェックリスト」には，上記の回答が反映されています。

Web Add-in Extensibility の計画チェックリスト

☑	カスタム・コントロールのサポート計画チェックリスト	ツールキット XML で指定	JavaScript 関数でサポート
☑	このカスタム・コントロールのソースの場所：%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample\Application\UsedBooks.htm	該当なし	該当なし
☑	新しいテスト・オブジェクト・クラスが拡張する Web テスト・オブジェクト基本クラスを指定：（標準設定—WebElement）WebTable	該当なし	該当なし
☑	基本テスト・オブジェクト・クラスは WebElement か？ いいえ 「いいえ」の場合，基本要素（基本テスト・オブジェクト・クラスに対応する要素）があるか？ はい 基本要素がある場合，それを返す JavaScript 関数が必要か？ はい	はい	はい
☑	新しい Web テスト・オブジェクト・クラスの詳細を指定：	該当なし	該当なし

☑	カスタム・コントロールのサポート計画チェックリスト	ツールキット XML で指定	JavaScript 関数でサポート
	• テスト・オブジェクト・クラス名： WebExtUsedBooks • アイコン・ファイルの場所（任意）： <UFT インストール・フォルダ>\dat\Extensibility\Web\Toolkits\WebExtSample\Res\WebBookList.ico • 記述のための認識プロパティ： title • 標準設定のテスト・オブジェクト・メソッド： SelectBook • ヘルプ・ファイルの場所：該当なし		
☑	コントロールに使用するテスト・オブジェクト・クラスを識別するための基準を指定： tagName = div 最初の子の tagName = table 最初の子の className = UsedBooks	はい	はい
☑	テスト・オブジェクトの名前付けの基礎を指定： UsedBooks テーブルのタイトルを使用	該当なし	はい
☑	サポートする認識プロパティをリストし，チェックポイントで使用可能とするプロパティ（および標準で選択するプロパティ），スマート認識で使用するプロパティ（該当する場合）をマーク。 title（チェックポイントで使用可能，標準で選択，ただしスマート認識では使用されない）	いいえ	はい
☑	サポートするテスト・オブジェクト・メソッドのリスト（必要な場合，引数，戻り値，ヘルプ・ファイルの場所，ヘルプ ID を指定）： SelectBook (BookIndex) RowCount	はい	はい
☑	テスト・オブジェクト・メソッド引数に対して動的な値のリストを指定するか？ いいえ 指定する場合，引数をリスト：	該当なし	いいえ
☑	UFT がコントロールで学習する子のタイプを指定： ラジオ・ボタン	はい	はい
☑	オブジェクト・スパイにこのクラスのテスト・オブジェクトを表示させるか？ はい	いいえ	該当なし
☑	記録をサポートするか？ はい その場合，記録をトリガするイベントを列挙： ［Select］のクリック	はい	はい

ツールキット・サポート・セットの開発

次の手順に従って，WebExtSample ツールキットのツールキット・サポート・セットを開発し，Web Add-in Extensibilityについてさらに学習します。ツールキット・サポート・セットの開発作業は，次の工程に従って行います。

- 「第 1 段階：追加のコントロールをサポートするためにツールキット・サポート・セットを拡張」（121 ページを参照）
- 「第 2 段階：UsedBooks コントロールの識別，調査，学習の実行を UFT に指示」（122 ページを参照）
- 「第 3 段階：WebExtUsedBooks テスト・オブジェクト・メソッドのサポートを実装」（126 ページを参照）
- 「第 4 段階：WebExtUsedBooks 認識プロパティとテスト・オブジェクト名のサポートを実装」（129 ページを参照）
- 「第 5 段階：子オブジェクトを学習対象から除外するフィルタを実装」（130 ページを参照）
- 「第 6 段階：UsedBooks コントロール上の記録のサポートを実装」（132 ページを参照）

第 1 段階：追加のコントロールをサポートするためにツールキット・サポート・セットを拡張

UsedBooks コントロールのサポートを追加するには，まず **WebExtUsedBooks** テスト・オブジェクト・クラスの定義を **WebExtSampleTestObjects.xml** ファイルに追加します。

UsedBooks コントロールをサポートするためにツールキット・サポート・セットを拡張するには，次の手順で行います。

1. UsedBooks コントロールのアイコン・ファイルである **WebBookList.ico** を，**%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample\Res** から **<ツールキット・サポート・セット・フォルダ>\Toolkits\WebExtSample\Res** フォルダにコピーします。
2. **WebExtUsedBooks** テスト・オブジェクト・クラスの次の定義を，**WebExtSampleTestObjects.xml** ファイルに追加します（**TypeInformation** 要素内，WebExtBook テスト・オブジェクト・クラスの ClassInfo 要素の後に追加します）。

```
<ClassInfo BaseClassInfoName="WebTable"
        GenericTypeID="Table"
        Name="WebExtUsedBooks"
        DefaultOperationName="SelectBook">
 <IconInfo IconFile="INSTALLDIR\dat\Extensibility\Web\Toolkits\
        WebExtSample\Res\WebBookList.ico"/>
  <TypeInfo>
    <Operation ExposureLevel="CommonUsed"
```

```
            Name="SelectBook"
            PropertyType="Method">
        <Description>
          Selects the radio button for the specified book and clicks Select.
        </Description>
        <Documentation>
          <![CDATA[Select the radio button for the book with index %a1 and click Select.]]>
        </Documentation>
        <Argument Name="BookIndex"
              IsMandatory="true"
              Direction="In">
          <Type VariantType="Integer"/>
        </Argument>
      </Operation>
    </TypeInfo>
    <IdentificationProperties>
      <IdentificationProperty ForDefaultVerification="true"
                      ForVerification="true"
                       ForDescription="true"
                      Name="title"/>
    </IdentificationProperties>
  </ClassInfo>
```

ここでは，**WebExtUsedBooks** テスト・オブジェクト・クラスを 「Web Add-in Extensibility の計画チェックリスト」(119ページ) の内容に従って定義しています。

テスト・オブジェクト設定ファイル内の要素と属性の詳細については，『UFT Test Object Schema Help』（『Web Add-in Extensibility ヘルプ』から利用可能）を参照してください。

第 2 段階：UsedBooks コントロールの識別，調査，学習の実行を UFT に指示

新しいテスト・オブジェクト・クラスの定義が完了したら，このテスト・オブジェクト・クラスを使用する Web コントロールを UFT が識別できるようにします。

「Web Add-in Extensibility のサンプル・コントロール UsedBooks のサポートの計画」(114ページ)で説明したように，**tagName** プロパティが **div** であるコントロールの最初の子の **tagName** プロパティ，**className** プロパティがそれぞれ **table**，**UsedBooks** である場合，このコントロールは **WebExtUsedBooks** テスト・オブジェクトで表されます。

WebExtUsedBooks テスト・オブジェクト・クラスの認識は，ツールキット設定ファイルで定義されている **Condition** 要素と JavaScript 関数を組み合わせて行います。

WebExtUsedBooks テスト・オブジェクト・クラスの認識ルールを定義するには，次の手順で行います。

1. **WebExtSample.xml** ファイルで，**Controls** 要素にこのテスト・オブジェクト・タイプの**Control** 要素を次のように追加します。

```
<Control TestObjectClass="WebExtUsedBooks">
  <Settings>
    <variable name="default_imp_file"    value="WebExtUsedBooks.js"/>
  </Settings>
  <Identification type="javascript"
            function="IsWebExtUsedBooks">
   <Browser name="*">
    <Conditions type="CallIDFuncIfPropMatch" logic="and">
      <Condition prop_name="tagName"    expected_value="div"/>
    </Conditions>
   </Browser">
  </Identification>
</Control>
```

   この定義によって UFT は，別のファイルが指定されていない場合，**WebExtUsedBooks.js** ファイル内で JavaScript 関数を検索します。**Identification** 要素には **Conditions** 要素が 1 つあります。この要素で指定されている条件では，コントロールの **tagName** プロパティが **div** の場合（大文字と小文字を区別しないで比較），JavaScript 関数 **IsWebExtUsedBooks** を呼び出して，コントロールを表すためにこのテスト・オブジェクト・クラスを使用するかどうかを決定します。

   このチュートリアルでは，上記の定義のように，**Conditions** 要素の **Type** 属性に CallIDFuncIfPropMatch 値を使用しています。ただし，アプリケーションのページ上にコントロールが多数存在する場合や，DOM 構造が大きい場合には，次のように識別ルールを定義することをお勧めします。

```
  <Identification type="javascript"
            function="IsWebExtUsedBooks">
    <HTMLTags>
      <Tag name="div"/>
    </HTMLTags>
  </Identification>
```

   これにより，UFT は div 要素のみに対して **IsWebExtUsedBooks** 識別関数を呼び出すので，機能的には同じですが，カスタム・コントロールの学習とステップ実行でのパフォーマンスが向上します。

2. ツールキット・サポート・セット・フォルダの **Toolkits\WebExtSample** フォルダに，**WebExtUsedBooks.js** という名前のファイルを作成します（このファイルは，UsedBooks コントロールのサポートのために作成するすべての JavaScript 関数で使用します）。

3. **WebExtUsedBooks.js** に次の JavaScript 関数を追加します。

```
function IsWebExtUsedBooks()
{
   // Verify that the tagName property is "div" and the
   // className property of the first child (a TABLE element)
   // is "UsedBooks".
   var firstChild = _elem.children[0];
   if   ( _elem.tagName == "DIV" &&
      firstChild.tagName == "TABLE" &&
      firstChild.className == "UsedBooks" )
      return true;
   return false;
}
```

この JavaScript 関数は，**WebExtUsedBooks** テスト・オブジェクトの適用先となる条件をコントロールが満たしているかどうかをチェックします。

ツールキット・サポート・セットのデプロイとテスト（第 2 段階）

WebExtUsedBooks テスト・オブジェクト・クラスをテスト・オブジェクト設定ファイルで定義し，このテスト・オブジェクト・クラスの認識ルールをツールキット設定ファイルと JavaScript 関数で定義する作業が完了したら，このツールキット・サポート・セットの動作を UFT でテストします。

ツールキット・サポート・セットをテストするには，次の手順を実行します。

1. テスト・オブジェクト設定ファイル，ツールキット設定ファイル，アイコン・ファイル，JavaScript ファイルを，UFT インストール・フォルダ内の所定の場所にデプロイします。
2. UFT を開いて **WebExtSample** サポートをロードします（［アドイン マネージャ］で選択）。
3. GUI テストを開き，［オブジェクト リポジトリ］ダイアログ・ボックスの［**新規テストオブジェクトの定義**］ボタン をクリックすると，［新規テストオブジェクトの定義］ダイアログ・ボックスが開きます。［**環境**］リストで WebExtSample 環境を選択し，テスト・オブジェクト設定ファイルで定義した **WebExtUsedBooks** テスト・オブジェクト・クラスが［**クラス**］リストに表示されていることを確認します。
4. サンプル・コントロールを実行するために，**%ALLUSERSPROFILE%\Documents\ExtAccTool\Samples\WebExtSample\Application\Book.htm** ファイルを開いて **Used** をクリックします。

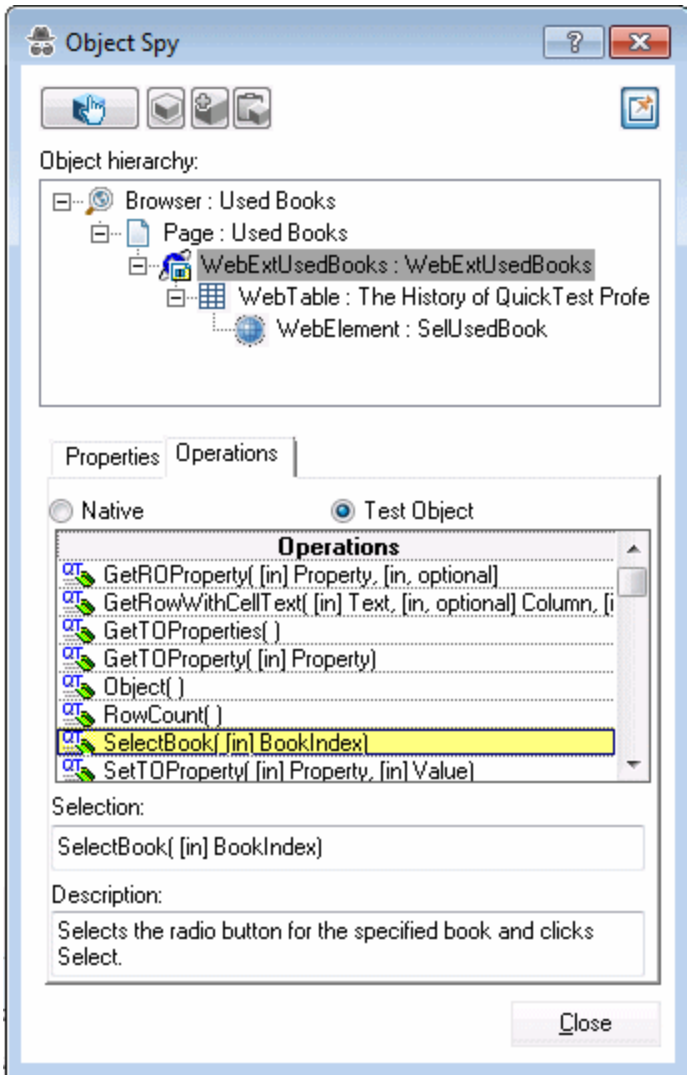
> **注:** UFT は，アプリケーションが開いたときにアプリケーションとの接続を確立します。したがって，UsedBooks コントロールが開いている場合には， 終了してから再度起動する必要があります。

5. UsedBooks コントロールで次の操作を実行し，UFT がコントロールを認識する方法を確認します（UFT での作業の詳細については，『HP Unified Functional Testing ユーザーズ・ガイド』を参照してください）。

   - オブジェクト・スパイを開き，UsedBooks コントロールでサポートされている認識プロパティおよびテスト・オブジェクト操作を確認します。**title** プロパティの値を返す JavaScript 関数は実装していないので，このプロパティの値は表示されません。

     UsedBooks コントロールに対して作成されたテスト・オブジェクトにはテスト・オブジェクト・クラスの名前が割り当てられており，定義したカスタム・アイコンが使用されます。このレッスンの後のほうで，ツールキット・サポート・セットをカスタマイズして，わかりやすい名前を指定します。

     WebExtUsedBooks テスト・オブジェクトには，WebTable テスト・オブジェクトで使用可能なすべてのテスト・オブジェクト操作と，テスト・オブジェクト設定ファイルで定義した SelectBook メソッドが含まれています。



- UsedBooks コントロールを学習するには，［オブジェクト リポジトリ］ダイアログ・ボックスの［**ローカルへオブジェクトを追加**］ボタン を使用します。オブジェクト・リポジトリでは，カスタム・アイコンを使用してテスト・オブジェクトが表示されます。
- キーワード・ビューを開き，**WebExtUsedBooks** オブジェクトを［**項目**］カラムで選択してテスト・ステップを作成します。
  - ［**操作**］カラムで使用可能な操作のリストは，テスト・オブジェクト設定ファイルの定義を反映しています。WebExtUsedBooks テスト・オブジェクトが WebTable テスト・オブジェクト・クラスを拡張（したがって，継承）するようにテスト・オブジェクト設定ファイルで定義したので，WebTable テスト・オブジェクトがサポートするテスト・オブジェクト操作はすべてサポートされます。
  - 操作を選択すると，選択した操作の引数の数に基づいて［**値**］セルが分割されます。たとえば，**SelectBook** 操作を実行するステップを作成すると，引数が 1 つあるので，引数の Name 属性がツールヒントに表示されます。
  - テスト・オブジェクト設定ファイルでテスト・オブジェクト・メソッドに対して定義した説明の文字列はツールヒント，注釈の文字列は［**注釈**］カラムに表示されます。
- エディタを開き，WebExtBook テスト・オブジェクトを使用するテスト・ステップを作成します。ステートメントの自動補完機能によって，テスト・オブジェクトで使用可能な操作がすべて表示され，WebTable から継承した操作もこれに含まれます。

6. WebExtUsedBooks テスト・オブジェクトで SelectBook テスト・オブジェクト・メソッドを実行するステップを含むテストを実行します。UFT は，コントロール上でテスト・オブジェクト・メソッドを実行する JavaScript 関数を検索します。テスト・オブジェクト・メソッド実行のサポートを実装していないので，実行時エラーが発生します。このサポートの実装は，次の項で行います。

第 3 段階：WebExtUsedBooks テスト・オブジェクト・メソッドのサポートを実装

テスト・オブジェクト設定ファイルでは，WebExtUsedBooks テスト・オブジェクトで使用可能なテスト・オブジェクト・メソッドを定義しました。このテスト・オブジェクト・メソッドを UFT で実行するには，メソッドを実際に実装する必要があります。

次のタイプのテスト・オブジェクト・メソッドの実装が必要です。

- WebTable 基本テスト・オブジェクト・クラスから継承されたテスト・オブジェクト・メソッド
- 新しいテスト・オブジェクト・クラスに追加されたテスト・オブジェクト・メソッド
- 別途実装が必要な基本クラスから継承されたテスト・オブジェクト・メソッド

WebTable から継承されたテスト・オブジェクト・メソッドの実装

テスト・オブジェクト設定ファイルでは，WebExtUsedBooks テスト・オブジェクト・クラスが基本クラス WebTable を拡張するように定義しました。継承した WebTable のテスト・オブジェクト・メソッドは，オーバーライドしない場合，UsedBooks コントロール内で定義されている **table** 基本要素

とやりとりをすることによって，内部実装を使用することが可能です。**table** 要素は UsedBooks コントロールのルート・レベルにはないので，**table** 要素が基本要素であることを UFT に通知する必要があります。手順としては，基本要素を返す JavaScript 関数を記述し，ツールキット設定ファイルで関数名を指定します。

table 要素を基本要素として使用するように UFT に指示するには，次の手順で行います。

1. **WebExtSample.xml** ファイルで，WebExtUsedBooks テスト・オブジェクト・クラスの **Control** 要素で定義した **Settings** 要素に，次の **Variable** 要素を追加します。

```
<Control TestObjectClass="WebExtUsedBooks">
  <Settings>
    <variable name="func_to_get_base_elem"
          value="GetTableElem"/>
  </Settings>
</Control>
```

   これによって UFT は，**GetTableElem** という名前の JavaScript 関数（**WebExtUsedBooks.js** ファイル）を呼び出して，継承された WebTable テスト・オブジェクト・メソッドをサポートする基本要素を返します。

2. **WebExtUsedBooks.js** ファイルに，次の JavaScript 関数を追加します。

```
function GetTableElem()
{
  // Get the <table> element (the first child of the <div>
  // element which is the root of the UsedBooks control)
  return _elem.children[0];
}
```

   この JavaScript 関数は **table** 要素を返します。これは，UsedBooks コントロールを定義する **div** 要素内の最初の要素です。この要素がサポートするのは，WebTable から継承されたテスト・オブジェクト・メソッドのうち WebExtUsedBooks で実装されていないメソッドです。

   このファイル内で記述されているほかの JavaScript 関数も，**GetTableElem()** 関数を使用して UsedBooks コントロール内の table 要素にアクセスします。

新しいテスト・オブジェクト・メソッド SelectBook の実装

WebExtUsedBooks テスト・オブジェクト・クラスで **SelectBook** テスト・オブジェクト・メソッドをサポートするには，**SelectBook** という名前の JavaScript 関数を **WebExtUsedBooks.js** で記述します。これは，**SelectBook** テスト・オブジェクト・メソッドが実行するときに UFT が呼び出す関数です。これにより，指定された本に対応するラジオ・ボタンを選択して［**Select**］をクリックする操作がシミュレートされます。

次の JavaScript 関数を **WebExtUsedBooks.js** ファイルに追加します。

```
function SelectBook( BookIndex )
// Select the radio button for the specified index
// and clicks the "Select" link.
{
   if( BookIndex > BookCount() )
      throw "Book index is out of range !"
   //Select the radio button corresponding to the specified index
   GetTableElem().rows[1+BookIndex].cells[0].children[0].click();
   // Click the "Select" link (the 3rd child of the <div> element)
   _elem.children[2].click();
   // Add a log message to the event log to assist in debugging
   _util.LogLine("Book Selected",1);
}
```

継承された RowCount テスト・オブジェクト・メソッドの実装をオーバーライド

1. **WebExtSample.xml** ファイルで，**WebExtUsedBooks** サポートを定義する **Control** 要素に次の **Run** 要素を追加します。

   ```
   <Run>
     <Methods>
       <Method name="RowCount" type="javascript"
           function="BookCount" />
     </Methods>
   </Run>
   ```

   これにより，**RowCount** テスト・オブジェクト・メソッドが JavaScript 関数 **BookCount** によって実装されることが定義されます。

2. **WebExtUsedBooks.js** ファイルに JavaScript 関数 **BookCount** を追加します。この関数は，UsedBooks コントロールの行数を減らしてテーブル内の本の冊数を返します。

   ```
   function BookCount()
   // This function overrides the RowCount test object method
   // inherited from WebTable, so that it counts only book rows.
   {
     var table = GetTableElem();
     if( table.rows.length < 2 )
       return 0;
     return table.rows.length - 2;
   }
   ```

ツールキット・サポート・セットのデプロイとテスト（第 3 段階）

テスト・オブジェクト・メソッドを実行するサポートの作成が完了したら，更新したツールキット・サポート・セットを UFT にデプロイしてテストします。

テスト・オブジェクト・メソッド実行のサポートをテストするには，次の手順で行います。

1. 更新したツールキット・サポート・セットを UFT にデプロイするには，**WebExtUsedBooks.js** ファイル（および必要に応じて **WebExtSample.xml**）を **<UFT インストール・フォルダ>\dat\Extensibility\Web\Toolkits\WebExtSample** にコピーします。
2. UFT を閉じてから再び開きます。［アドイン マネージャ］ダイアログ・ボックスで **WebExtSample** のチェック・ボックスを選択し，［**OK**］をクリックします。UFT が開き，設計したサポートがロードされます。
3. **UsedBooks** サンプル・コントロールを終了してから再度実行します。
4. UsedBooks コントロールの学習には，GUI テストを開き，［オブジェクト リポジトリ］ダイアログ・ボックスの［**ローカルへオブジェクトを追加**］ボタン を使用します。
5. **SelectBook** メソッドと **RowCount** メソッドを実行するテストを作成し，正しく実行されることを確認します。また，Microsoft Windows イベント・ビューアを開くと，**SelectBook** メソッドによって記録されたログ・メッセージを表示することもできます（詳細については「Microsoft Windows イベント・ログの使用」(60ページ)を参照してください）。

第 4 段階：WebExtUsedBooks 認識プロパティとテスト・オブジェクト名のサポートを実装

WebExtUsedBooks.js ファイルで，**get_property_value** を次のように実装します。

```
function get_property_value(prop)
// The function provides values for all of the identification
// properties defined in the test object configuration XML file, as
// well as the hard coded "logical_name" property that UFT
// uses to name the test object.
{
   if ( prop == "logical_name" || prop == "title" )
   // For the "title" identification property, as well as the
   // "logical_name" property, return the inner text of the
   // first cell in the first row
   {
      return GetTableElem().rows[0].cells[0].innerText;
   }
}
```

この関数は，**title** プロパティとテスト・オブジェクト名で使用するオブジェクトのタイトルを返します。

注: ツールキット・サポート・セット・フォルダにある **WebExtUsedBooks.js** ファイルを変更してから UFT にデプロイする方法と，**<UFT インストール・フォルダ >\dat\Extensibility\Web\Toolkits\WebExtSample\WebExtUsedBooks.js** を直接変更する方法があります。

ツールキット・サポート・セットのデプロイとテスト（第 4 段階）

認識プロパティの実行時の値の取得のサポートを作成したら，更新したツールキット・サポート・セットを UFT にデプロイしてテストします。

認識プロパティの実行時の値の取得のサポートをテストするには，次の手順で行います。

1. 最新のファイルが **<UFT インストール・フォルダ >\dat\Extensibility\Web\Toolkits\WebExtSample** に格納されていることを確認します。
2. UFT を閉じてから再び開きます。［アドイン マネージャ］ダイアログ・ボックスで **WebExtSample** のチェック・ボックスを選択し，［**OK**］をクリックします。UFT が開き，設計したサポートがロードされます。
3. **UsedBooks** サンプル・コントロールを終了してから再度実行します。
4. GUI テストを新規作成し，WebExtUsedBooks 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て，このテスト・オブジェクトを使用するテスト・ステップを作成します。テスト・オブジェクトの名前がテーブルのタイトルに基づいていることを確認します。オブジェクトを右クリックして［**標準チェックポイントの挿入**］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。テスト・オブジェクト設定ファイルで定義した **title** 認識プロパティがプロパティ・リストで表示され，選択されていることを確認します。
5. それぞれの認識プロパティを取得し，値をチェックするかメッセージ・ボックスで表示するテストを作成して実行します。

第 5 段階：子オブジェクトを学習対象から除外するフィルタを実装

この項では，UFT による学習対象から，UsedBooks コントロールの子をすべて除外するフィルタを作成します。

このフィルタの実装には，ツールキット設定ファイルと JavaScript ファイルを使用します。

UsedBooks コントロールの子を学習対象から除外するフィルタを作成するには，次の手順で行います。

1. **WebExtSample.xml** ファイルで，**WebExtBook** テスト・オブジェクト・クラスに対して定義されている **Control** 要素に，次の **Filter** 要素を追加します。

```
<Filter>
   <Learn learn_control="Yes" learn_children="CallFilterFunc"
       type="javascript" function="GetChildrenToLearn" />
</Filter>
```

   これによって UFT は，親の Web ページと一緒に **WebExtUsedBooks** テスト・オブジェクトを学習し，JavaScript 関数 **GetChildrenToLearn** を呼び出すことによって学習対象に含める子オブジェクトを決定します。この JavaScript 関数は，コントロールでの学習対象となる子孫の SafeArray を返します。

   注: ツールキット・サポート・セット・フォルダにある **WebExtSample.xml** ファイルを変更してから UFT にデプロイする方法と，**<UFT インストール・フォルダ >\dat\Extensibility\Web\Toolkits\WebExtSample\.xml** を直接変更する方法があります。

2. **WebExtUsedBooks.js** ファイルに，次の関数を追加します。

```
// Learn filtering
// This function instructs UFT which child objects of a
// UsedBooksTable should be learned with the object is learned.
function GetChildrenToLearn()
{
   // Return all of the radio buttons in the UsedBooks table
   return toSafeArray(GetTableElem().getElementsByTagName("input") );
}
```

   このレッスンの最初で計画したとおり，上記の関数によってラジオ・ボタンのみが学習されます。

ツールキット・サポート・セットのデプロイとテスト（第 5 段階）

子の学習をカスタマイズするフィルタの定義が完了したら，更新したツールキット・サポート・セットを UFT にデプロイしてテストします。

WebExtUsedBooks テスト・オブジェクトの学習で子を学習対象から除外するサポートをテストするには，次の手順で行います。

1. 最新のファイルが **<UFT インストール・フォルダ >\dat\Extensibility\Web\Toolkits\WebExtSample** に格納されていることを確認します。
2. UFT を閉じてから再び開きます。［アドイン マネージャ］ダイアログ・ボックスで

WebExtSample のチェック・ボックスを選択し，［**OK**］をクリックします。UFT が開き，設計したサポートがロードされます。

3. **UsedBooks** サンプル・コントロールを終了してから再度実行します。
4. GUI テストを開き，オブジェクト・リポジトリを開きます。UsedBooks コントロールを含む Web ページを学習するには，［オブジェクト リポジトリ］ダイアログ・ボックスの［**ローカルへオブジェクトを追加**］ボタン を使用します。［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。
5. ［**すべてのオブジェクト タイプ**］を選択して［**OK**］をクリックします。**The History of QuickTest Professional** という名前の WebExtUsedBooks オブジェクトがオブジェクト・リポジトリに追加され，SelUsedBook ラジオ・ボタン・グループも追加されます。ただし，コントロールに含まれるほかの要素は追加されません。

第 6 段階：UsedBooks コントロール上の記録のサポートを実装

この項では，UsedBooks コントロール上の記録のサポートを実装します。

1. **WebExtSample.xml** ファイルで，WebExtUsedBooks クラスを定義する **Control** 要素に次の **Record** 要素を追加します。

```
<Record>
   <EventListening use_default_event_handling_for_children="false"
              use_default_event_handling="false"
              type="javascript" function="ListenToEvents"/>
</Record>
```

これによって UFT は，UsedBooks コントロールで発生したイベントの記録に，標準の Web イベントの定義を使用するのではなく，JavaScript 関数 **ListenToEvents** を呼び出します。

WebExtUsedBooks.js ファイルに JavaScript 関数 **ListenToEvents** を追加します。

```
function ListenToEvents( elem )
{
   // Connect to the "Select" event:
   //When "Select" is clicked, call OnSelectUsedBooksClicked.
   _util.RegisterForEvent(_elem.children[2], "onclick",
                    "OnSelectUsedBooksClicked" );
   return true;
}
```

この関数によって，UFT は［Select］リンクのクリックをリッスンし，イベント発生時に適切なイベント・ハンドラを呼び出します。

2. **WebExtUsedBooks.js** ファイルに，イベントを処理する JavaScript 関数 **OnSelectUsedBooksClicked** を追加します。

```
function OnSelectUsedBooksClicked( handlerParam , eventObj )
{
   var arr = new Array();
   var booksCount = BookCount();
   // Find the index of the selected radio button and record
   // a step that runs the SelectBook test object method
   // with that index.
   var BookIndex = -1;
   for( var i = 0 ; i < booksCount ; i++ )
   {
      if( _elem.rows[2+i].cells[0].children[0].status == true )
      {
         // This is the selected item
         arr[0] = i+1;
         _util.Record( "SelectBook", toSafeArray(arr) , 0 );
         _util.LogLine("SelectBook Recorded",1);
         break;
      }
   }
   return true;
}
```

この関数は，どの本のラジオ・ボタンが選択されたかをチェックし，その本を選択するステップを記録するように UFT に指示します（さらに，ログ・メッセージをイベント・ログに書き込みます）。

ツールキット・サポート・セットのデプロイとテスト（第 6 段階）

UsedBooks コントロール上の記録のサポートの作成が完了したら，更新したツールキット・サポート・セットを UFT にデプロイしてテストします。

UsedBooks コントロールで実行される操作の記録のサポートをテストするには，次の手順で行います。

1. 最新のファイルが **<UFT インストール・フォルダ>\dat\Extensibility\Web\Toolkits\WebExtSample** に格納されていることを確認します。
2. UFT を閉じてから再び開きます。［アドイン マネージャ］ダイアログ・ボックスで **WebExtSample** のチェック・ボックスを選択し，［**OK**］をクリックします。UFT が開き，設計したサポートがロードされます。
3. **UsedBooks** サンプル・コントロールを終了してから再度実行します。
4. GUI テストを開き，［**記録**］ボタンをクリックするか，［**記録**］ > ［**記録**］を選択します。UsedBooks テーブルでラジオ・ボタンを 1 つ選択し，［**Select**］をクリックします。

［**Select**］をクリックしないと，テストには新しいステップは追加されません。

Item	Operation	Value	Documentation
Action1			
Used Books			
Used Books			
The History of QuickTest	SelectBook	2	Select the radio button for the book with index 2 and click Select.

［**停止**］ボタンをクリックするか，［**記録**］ > ［**停止**］を選択して記録セッションを終了します。

以上で，カスタム・サポートの計画時の仕様に従って，Book カスタム・コントロールが完全にサポートされました。

レッスンのまとめ

このレッスンでは，以下のことを学びました。

- テスト・オブジェクト・クラスを 1 つ作成し（WebExtUsedBooks），WebTable テスト・オブジェクト・クラスを拡張しました。
- UsedBooks コントロールのサポートを作成し，UFT で WebExtUsedBooks テスト・オブジェクトとして認識されるようにしました。
- ツールキット設定ファイルの各種オプションについて学習しました。
- より複雑な JavaScript 関数を使用し，ツールキット設定ファイルで格納場所を指定することにより，サポートを実装する方法を学習しました。

その他の情報

これで，このチュートリアルのレッスンはすべて終了しました。このレッスンで学習した Web Add-in Extensibility の考え方やスキルを各自のカスタム・ツールキット・サポートの作成に活用してください。

ツールキット・サポート・セットの構造と内容の詳細は，「ツールキットのサポートの開発」(31 ページ)を参照してください。

テスト・オブジェクト設定ファイルの構造と内容の詳細については，『UFT Test Object Schema Help』（Web Add-in Extensibility ヘルプから利用可能）を参照してください。

ツールキット設定ファイルの構造と内容の詳細については，『UFT Java Add-in Extensibility Toolkit Configuration Schema Help』（『Web Add-in Extensibility ヘルプ』から利用可能）を参照してください。

_util ユーティリティ・オブジェクトとグローバル JavaScript メソッドの詳細については，『UFT Java Add-in Extensibility API Reference』（『Web Add-in Extensibility ヘルプ』から利用可能）を参照してください。

フィードバックをお送りください

開発者ガイド について何かお気づきのことがありませんか?

ご意見をお聞かせください。sw-doc@hp.com

Java™
COMPATIBLE