

HP UFT Delphi Add-in Extensibility

ソフトウェア・バージョン : 11.50

開発者ガイド

ドキュメント・リリース日 : 2012 年 12 月 (英語版)

ソフトウェア・リリース日 : 2012 年 12 月 (英語版)



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。ここでの記載で追加保証を意図するものは一切ありません。ここに含まれる技術的、編集上の誤り、または欠如について、HPはいかなる責任も負いません。

ここに記載する情報は、予告なしに変更されることがあります。

権利の制限

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

著作権について

© 1992 - 2012 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。

Intel®, Pentium® および Intel® Xeon™ は、米国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

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

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

Oracle® は、カリフォルニア州レッドウッド市の Oracle Corporation の米国登録商標です。

Unix® は、The Open Group の登録商標です。

SlickEdit® は、SlickEdit Inc. 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。

<http://support.openview.hp.com/selfsolve/manuals>

このサイトを利用するには、HP Passport への登録とサインインが必要です。HP Passport ID の登録は、次の Web サイトから行なうことができます。

<http://h20229.www2.hp.com/passport-registration.html> (英語サイト)

または、HP Passport のサインインページの **[New users - please register]** をクリックします。

適切な製品サポートサービスをお申し込みいただいたお客様は、更新版または最新版をご入手いただけます。詳細は、HP の営業担当にお問い合わせください。

サポート

次のHPソフトウェアサポートのWebサイトを参照してください。

<http://support.openview.hp.com>

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

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

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

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

<http://h20229.www2.hp.com/passport-registration.html> (英語サイト)

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。

http://support.openview.hp.com/access_level.jsp

目次

Delphi Add-in Extensibility へようこそ	7
UFT Delphi Add-in Extensibility SDK について	8
このガイドについて	8
対象読者	9
Unified Functional Testing ヘルプ	10
その他のオンライン・リソース	10
第 1 章：カスタム Delphi コントロールのサポートの開発	13
用語	14
カスタム Delphi コントロールのサポートを作成する方法	15
テスト・オブジェクト設定 XML ファイルの作成	19
ツールキット設定 XML ファイルの作成	23
Delphi Extensibility コードの作成	24
カスタム・グリッド・コントロールのサポートの作成	30
カスタム Delphi コントロールをサポートする手順	35
第 2 章：ツールキット・サポート・セットのデプロイ	39
カスタム・ツールキット・サポートのデプロイについて	40
XML ファイルの格納場所	41
拡張コードのコンパイル	42
デプロイ済みサポートの変更	42
デプロイ済みサポートの削除	43
第 3 章：ツールキット設定 XML ファイルの構造について	45
ツールキット設定 XML ファイルについて	45
アプリケーション・コントロールをテスト・オブジェクト・クラスに マッピング	47
テスト・オブジェクト・クラスを内部オブジェクトにマッピング	49
ツールキット設定ファイル全体のサンプル	55

Delphi Add-in Extensibility へようこそ

HP UFT Delphi Add-in Extensibility は、Unified Functional Testing Delphi Add-in の標準でサポートされていないサードパーティ製やカスタマイズされた Delphi 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する SDK (Software Development Kit) パッケージです。

本章の内容

- ▶ UFT Delphi Add-in Extensibility SDK について (8ページ)
- ▶ このガイドについて (8ページ)
- ▶ 対象読者 (9ページ)
- ▶ Unified Functional Testing ヘルプ (10ページ)
- ▶ その他のオンライン・リソース (10ページ)

UFT Delphi Add-in Extensibility SDK について

UFT Delphi Add-in Extensibility SDK では、Delphi Add-in と一緒にインストールされ、次の内容が提供されます。

- ▶ Unified Functional Testing Delphi Add-in を拡張する API。これにより、Delphi のカスタム・コントロールをサポートします。
- ▶ 拡張コードの作成時に使用できるテンプレート。
- ▶ このヘルプ ([**スタート**] > [**すべてのプログラム**] > [**HP Software**] > [**HP Unified Functional Testing**] > [**Extensibility**] > [**Documentation**] でアクセス)。次の内容が含まれます。
 - ▶ 開発者ガイド
 - ▶ Unified Functional Testing Test Object Schema ヘルプ
- ▶ 印刷用 (PDF) バージョンの開発者ガイド (<インストール・ディレクトリ>\help\Extensibility フォルダに収録)。
- ▶ サンプルの Delphi Add-in Extensibility ツールキット・サポート・セット。UFT GUI テストを拡張することによって次の Delphi カスタム・コントロールをサポートします。
 - ▶ TrackBar
 - ▶ StringDrawGrid (テーブル・コントロール)

このガイドについて

このガイドでは、サードパーティが提供またはカスタマイズされた Delphi コントロールをサポートするために、UFT Delphi Add-in Extensibility を使用して UFT GUI テストを拡張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。また、API ファイル (<UFT インストール・フォルダ>\dat\Extensibility\Delphi\AgentExtensibilitySDK.pas) と拡張コード・テンプレート (<UFT インストール・フォルダ>\dat\Extensibility\Delphi\ExtensibilityImplementationTemplate.pas) を使用します (いずれもコメントと説明が提供されています)。さらに、Delphi 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**] > [**Delphi Add-in Extensibility Help**]) で提供されている『HP Unified Functional Testing Test Object Schema Help』も使用します。このドキュメントに併せて、『HP Unified Functional Testing ユーザーズ・ガイド』, 『HP Unified Functional Testing アドイン・ガイド』の Delphi の説明, 『HP Unified Functional Testing Object Model Reference』(UFT メイン・ウィンドウの[ヘルプ] > [**HP Unified Functional Testing ヘルプ**]) も参照してください。

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, 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・コンポーネントおよびスクリプト・コンポーネントは, HP Business Process Testing の一部で, アプリケーションのテストにキーワード駆動型の方法論が使用されます。詳細については, 『HP Unified Functional Testing ユーザーズ・ガイド』を参照してください。

対象読者

このガイドは, プログラマ, QA エンジニア, システム・アナリスト, システム・デザイナー, テクニカル・マネージャを対象に, UFT GUI テストを拡張することによって Delphi のカスタム・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, 次の内容に関する知識が必要になります。

- ▶ UFT の主要機能
- ▶ UFT オブジェクト・モデル
- ▶ Unified Functional Testing Delphi Add-in
- ▶ Delphi プログラミング
- ▶ XML (基本的な知識)

Unified Functional Testing ヘルプ

Unified Functional Testing ヘルプから、UFT に関するドキュメントにアクセスできます。

Unified Functional Testing ヘルプには、次の方法でアクセスできます。

- ▶ UFT で **[ヘルプ]** > **[HP Unified Functional Testing]** を選択します。
- ▶ UFT の **[スタート]** メニューから、**[すべてのプログラム]** > **[HP Software]** > **[HP Unified Functional Testing]** > **[Documentation]** > **[HP Unified Functional Testing ヘルプ]** を選択します。
- ▶ 選択した UFT ウィンドウおよびダイアログ・ボックスをクリックするか、F1 キーを押します。
- ▶ UFT テスト・オブジェクト、メソッド、またはプロパティの上にカーソルを置いて F1 キーを押すと、説明、構文、例が表示されます。

その他のオンライン・リソース

トラブルシューティング&ナレッジベース：問題の自己解決が可能な技術情報を検索できる、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のページにアクセスできます。**[ヘルプ]** > **[トラブルシューティング&ナレッジベース]** を選択します。このWebサイトのURLは、<http://support.openview.hp.com/troubleshooting.jsp> です。

HP ソフトウェアサポート：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。また、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、サポート要求の送信、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なうこともできます。**[ヘルプ]** > **[HPソフトウェア サポート]** を選択します。このWebサイトのURLは <http://support.openview.hp.com/> 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。

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。

http://support.openview.hp.com/access_level.jsp

HP Passport IDを登録するには、次のWebサイトにアクセスしてください。

<http://h20229.www2.hp.com/passport-registration.html> (英語サイト)

HPソフトウェアWebサイト : HPソフトウェアWebサイトにアクセスします。このサイトでは、HPソフトウェア製品に関する最新の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、カスタマーサポートなどの情報も含まれています。**[ヘルプ]** > **[HPソフトウェアWebサイト]** を選択します。このWebサイトのURLは、<http://support.openview.hp.com> です。

HP Software は、新しい情報を提供する目的で、製品の文書を継続的に更新しています。

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、HP Software 製品マニュアル (<http://support.openview.hp.com/selfsolve/manuals>) で確認できます。

はじめに

第 1 章

カスタム Delphi コントロールのサポートの開発

Unified Functional Testing Delphi Add-in を拡張するカスタム・ツールキット・サポート・セットを作成し、それを Unified Functional Testing (UFT) で指定することによって、Delphi Add-in が標準では対応できない Delphi ユーザ・インタフェース・コントロールの識別が可能になります。

本章では、UFT Delphi Add-in Extensibility を使用してユーザ定義の Delphi ユーザ・インタフェース・コントロールのサポートを作成する方法について説明します。具体的には、ツールキット・サポート・セットで作成するファイルとユニット、ファイルの構造と内容、カスタム Delphi コントロールを含むアプリケーションを UFT でテストする方法を取り上げます。

ツールキット・サポート・セットの格納場所や、作成したサポートをアクティブにするための手順については、第 2 章「ツールキット・サポート・セットのデプロイ」を参照してください。

本章の内容

- ▶ 用語 (14ページ)
- ▶ カスタム Delphi コントロールのサポートを作成する方法 (15ページ)
- ▶ テスト・オブジェクト設定 XML ファイルの作成 (19ページ)
- ▶ ツールキット設定 XML ファイルの作成 (23ページ)
- ▶ Delphi Extensibility コードの作成 (24ページ)
- ▶ カスタム・グリッド・コントロールのサポートの作成 (30ページ)
- ▶ カスタム Delphi コントロールをサポートする手順 (35ページ)

用語

- ▶ **Agent Object (AO)** : Delphi で記述されるクラスであり、Unified Functional Testing Delphi Add-in のプリコンパイル済みエージェントに組み込まれます。Agent Object は、プロパティの取得や、カスタム・コントロールでのテスト・オブジェクト・メソッドの記録と実行といった処理のサポートに必要なメソッドを提供します。Agent Object は、UFT とテスト対象の Delphi アプリケーションをつなぐインタフェースとして機能します。
- ▶ **カスタム・ツールキット** : 複数のカスタム・コントロールで構成されるセットであり、このセットに UFT サポートを実装します。
- ▶ **カスタム・ツールキット・サポート・セット** : カスタム・ツールキットに含まれるコントロールの UFT サポートを拡張するために作成される、ファイル、ユニット、オブジェクトのセット。
- ▶ **内部テスト・オブジェクト** : 標準 Windows テスト・オブジェクト・クラスです。作成した Agent Object で固有のサポートが提供されていないプロパティやメソッドがある場合、UFT はこの実装を使用します。

カスタム Delphi コントロールのサポートを作成する方法

Unified Functional Testing Delphi Add-in を拡張するカスタム・ツールキット・サポート・セットを作成し、これを UFT で指定することによって、Delphi Add-in が標準では対応できない Delphi ユーザ・インタフェース・コントロールの識別が可能になります。

カスタム・ツールキット・サポート・セットは、次の要素で構成されます。

- ▶ **必須**：ツールキット設定 XML ファイル。カスタム・コントロールをテスト・オブジェクト・クラス（既存の Delphi テスト・オブジェクト・クラスまたはユーザ定義の新しいテスト・オブジェクト・クラス）にマッピングします。
- ▶ **オプション**：テスト・オブジェクト設定 XML ファイル。UFT テストおよびコンポーネント内のカスタム・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・クラスを定義します。

必要なテスト・オブジェクト・メソッドと認識プロパティがすべて定義されている Delphi テスト・オブジェクト・クラスがすでに存在する場合、カスタム・コントロールを表すテスト・オブジェクト・クラスを定義する必要はありません。

- ▶ **オプション**：サポート対象となるコントロールのタイプごとに Agent Object を生成する拡張コードからなる Delphi ユニットと、Agent Object を生成するファクトリ関数。

既存の Delphi テスト・オブジェクト・クラスにマッピングすることで十分にサポートが提供されるコントロールには、Agent Object を作成する必要はありません。Agent Object の作成が必要になるのは、次の場合です。

- ▶ カスタム・コントロールを新しい（カスタム）テスト・オブジェクト・クラスにマッピングする場合。
- ▶ カスタム・コントロールを既存のテスト・オブジェクト・クラスにマッピングした状態で、テスト・オブジェクト・メソッドの実装をオーバーライドする場合、または認識プロパティの値を指定する場合。
- ▶ カスタム・グリッド・コントロールのサポートを作成する場合。

詳細については、次を参照してください。

- ▶ 19 ページ「テスト・オブジェクト設定 XML ファイルの作成」
- ▶ 23 ページ「ツールキット設定 XML ファイルの作成」
- ▶ 24 ページ「Delphi Extensibility コードの作成」
- ▶ 35 ページ「カスタム Delphi コントロールをサポートする手順」

Delphi Add-in Extensibility サンプルの使用

Delphi Add-in Extensibility SDK では次のサンプルが提供されています。Delphi Add-in Extensibility の学習や、Extensibility ファイルを作成するときの参考としてご活用ください。

- ▶ 基本的な Delphi Add-in Extensibility のサンプルには、標準の Delphi Add-in ではサポートされない **TTrackBar** コントロール用のツールキット・サポート・セットが含まれています。このツールキット・サポート・セットでは、カスタム・コントロール用の UFT サポートを拡張する方法を総合的にわかりやすく示す例が提供されています。
- ▶ Delphi Add-in Grid Extensibility のサンプルには、**TStringDrawGrid** コントロール用のツールキット・サポート・セットが含まれています。このコントロールは、**TCustomGrid** クラスから継承したカスタム・グリッド・コントロールです。このツールキット・サポート・セットでは、カスタム・グリッド・コントロールをテーブルとして処理するように UFT に指示する方法のみを紹介します。詳細については、30 ページ「カスタム・グリッド・コントロールのサポートの作成」を参照してください。

これらのサンプルは、<UFT インストール・フォルダ>\samplesの下にある

DelphiExtSample フォルダと **DelphiGridExtSample** フォルダに格納されています。各フォルダには、**Application** サブフォルダにカスタム・コントロールとそのソース・ファイル、**ToolkitSupportSet** サブフォルダにツールキット・サポート・セットのファイル（設定ファイルと拡張ユニット）が格納されています。

これらのサンプルは、次の用途に活用できます。

- ▶ ツールキット・サポート・セットのファイルを確認することで、独自のツールキット・セットを開発する方法を理解します。
- ▶ ツールキット・サポート・セットのファイル（またはその一部）をコピーし、これをベースにツールキット・サポート・セットを設計します。
- ▶ UFT とカスタム・コントロール間の処理に拡張機能がどのように影響するかを理解します。そのためには、サンプル・カスタム・コントロールに関する UFT テストを作成して実行し、サンプル・ツールキット・サポート・セットを UFT にデプロイする前と後で比較します。以下では、このプロセスを **TTrackBar** サンプルで行う手順を示しています。**TStringDrawGrid** サンプルでも同様のプロセスを実行できます。

TTrackBar Extensibility のサンプルが、UFT と TTrackBar カスタム・コントロール間の処理にどのような影響を与えるかを分析するには、次の手順で行います。

- 1** UFT オブジェクト・スパイを使用して、UFT が **TTrackBar** コントロールを認識する方法を確認します。コントロールに関する簡単なテストを作成して実行します。

サンプルでは、トラック・バー・コントロールを表すのに、汎用の **DelphiObject** テスト・オブジェクトを使用しています。トラック・バーのつまみの位置設定には、**Click**、**Drag**、**Drop** の各操作を使用する必要があります。

- 2** 第 2 章「ツールキット・サポート・セットのデプロイ」の手順に従って、**TTrackBar** コントロール用のツールキット・サポート・セットをデプロイします。
- 3** **Extensibility** のサポートを有効にした状態で、UFT が **TTrackBar** コントロールをどのように認識するのかを UFT オブジェクト・スパイで確認します。コントロールに関する簡単なテストを作成して実行します。

Extensibility のサポートを有効にすると、UFT は **DelphiTrackBar** テスト・オブジェクト（およびカスタマイズしたアイコン）を使用してコントロールを表します。**DelphiTrackBar** テスト・オブジェクトは **Set**、**Next**、**Prev** の各操作をサポートし、トラック・バーでのつまみの位置を変更します。さらに、記録セッション中にトラック・バーのつまみをドラッグすると、UFT は **Set** 操作を使用してテスト・ステップを記録します。

Delphi Add-in Grid Extensibility のサンプル・プロジェクトを Delphi Studio で開くには、次の手順で行います。

Delphi Add-in Grid Extensibility のサンプル・プロジェクトである **DelphiGridExtSample** は、**TStringDrawGrid** カスタム・コントロールを参照します。したがって、**DelphiGridExtSample** を Delphi Studio で開くには、まず **TStringDrawGrid** を Delphi Studio に登録する必要があります。

これを行うために、次の手順を実行します。

- 1 Delphi Studio で **[Component]** > **[Install Component]** を選択します。**[Install Component]** ダイアログ・ボックスが開きます。
- 2 **[Unit file name]** ボックスに **StringDrawGrid.pas** ファイルの完全パスを入力します。ファイルの格納場所は、**<UFT インストール・フォルダ>\samples\DelphiGridExtSample\Application** です。
- 3 **[OK]** をクリックします。確認または情報提供のダイアログ・ボックスが開いたら、**[Yes]** または **[OK]** をクリックします。
[Package] ダイアログ・ボックスが開きます。

注： Delphi Studio で **[Package]** ダイアログ・ボックスが開かず、**StringDrawGrid.pas** ファイルの編集画面が開いた場合は、1 ~ 3 の手順を再度実行してください。

- 4 **[Compile]** をクリックします。
- 5 **[Package]** ダイアログ・ボックスを閉じ、保存の確認ボックスが開いたら **[Yes]** をクリックします。
- 6 **DelphiGridExtSample** が開きます。

テスト・オブジェクト設定 XML ファイルの作成

このファイルでは、テストとコンポーネント内のカスタム・コントロールを表すために UFT が使用するカスタム・テスト・オブジェクト・クラスを定義します。テスト・オブジェクト・クラスの定義は、既存の Delphi テスト・オブジェクト・クラスでは十分に対応できないカスタム・コントロールごとに行います。

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。

既存のテスト・オブジェクト・クラスの定義をテスト・オブジェクト設定 XML で作成することが可能です。作成した定義はテスト・オブジェクト・クラスの既存の定義に追加され、このクラスのテスト・オブジェクトすべてに適用されます。したがって、この方法で既存のテスト・オブジェクト・クラスを変更することはお勧めしません。次に例を示します。

- ▶ 追加したテスト・オブジェクト・メソッドは、UFT でテスト・オブジェクト・メソッドのリストに表示されますが、このテスト・オブジェクト・メソッドをテストで使用していてもオブジェクトで実装していない場合には実行時エラーが発生します。

テスト・オブジェクト・メソッドを既存のテスト・オブジェクト・クラスに追加する場合は、メソッド名にプレフィックスを追加できます。このプレフィックスは、メソッドの追加先となるツールキット・サポートを示します (**CustomButtonClick** や **CustomEditSet** など)。テスト設計者はプレフィックスからカスタム・メソッドを簡単に特定し、各オブジェクトでサポートされるカスタム・メソッドのみをテスト・ステップで使用できるようになるので便利です。

- ▶ 追加した認識プロパティは、このクラスのすべてのテスト・オブジェクトで使用されるプロパティのリストに表示されますが、サポート対象オブジェクトで実装されていない場合、値はありません。

テスト・オブジェクト設定 XML ファイルでは、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 の起動時にアドインがロードされないと、UFT はこの XML 内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。

テスト・オブジェクト設定ファイルの構造が適切かどうかは、**ClassesDefintions.xsd** ファイルを使ってチェックできます。このファイルは、UFT のインストール時に **<UFT インストール・フォルダ>\dat** フォルダに格納され、(下位互換性を維持するために、UFT がサポートする XML 構造には、この XSD によるチェック条件を満たしていないものも含まれています)。

この XML の構造と構文の詳細については、UFT Test Object Schema ヘルプを参照してください。このヘルプは、UFT Delphi Add-in Extensibility ヘルプ (**<UFT インストール・フォルダ>\help\Extensibility** フォルダにある **DelphiExtensibility.chm** ヘルプ・ファイル) から利用できます。

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・クラスが拡張するテスト・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 Delphi UFT 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、Delphi 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 DelphiObject です。

テスト・オブジェクト・クラスの名前は、同時にロードする可能性のあるサポートについて、すべての環境内で一意になるように指定する必要があります。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 (DelphiButton, DelphiEdit, 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。

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 `DelphiObject`（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）で使われます。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われます（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。引数のタイプ（**String**, **Integer** など）、方向（**In** または **Out**）、引数の指定が必須かどうか、必須でない場合は標準設定値。
- ▶ 操作の説明（オブジェクト・スパイト、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・ビューとステップ・ジェネレータの [注釈] カラムで表示）。
- ▶ 戻り値のタイプ。
- ▶ キーワード・ビューまたはエディタで開いているテスト・オブジェクト操作で F1 キーを押すか、ステップ・ジェネレータで開いている操作で [操作ヘルプ] ボタンをクリックすると、コンテキスト・ヘルプ・トピックが開きます。設定内容には、ヘルプ・ファイルのパスとヘルプ ID が含まれます。

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。
- ▶ **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操作は、UFT の [オブジェクトの認識] ダイアログ・ボックスで行います）。
- ▶ 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標準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（オプション。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）。指定可能なファイルは、**.dll**、**.exe**、**.ico** のいずれかです。

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、ヘルプ・ファイルである **.chm** のパスとヘルプ ID を定義します。

ツールキット設定 XML ファイルの作成

このファイルでは、次のように 2 種類のマッピングを行います。

- ▶ **アプリケーション・コントロールをテスト・オブジェクトにマッピング** : Delphi アプリケーション内のカスタム・コントロールを Delphi テスト・オブジェクト・クラスにマッピングします。これにより、カスタム・コントロールは UFT テストおよびコンポーネント内で表されます。カスタム・コントロールのマッピング先は、テスト・オブジェクト設定ファイルで新しく定義した Delphi テスト・オブジェクト・クラス、または既存の Delphi テスト・オブジェクト・クラスです。

テスト・オブジェクト名、記述（アプリケーション内でオブジェクトを識別するために UFT が使用するプロパティ群）、コントロールに固有のプロパティとメソッドをカスタマイズするには、新規の Delphi テスト・オブジェクト・クラスを使用します。

注 : 特定のテスト・オブジェクト・メソッドのコントロールでの振舞いのみをカスタマイズするには（メソッドの構文はカスタマイズしない）、コントロールを既存のテスト・オブジェクト・クラスにマッピングします。作成した Agent Object で、テスト・オブジェクト・メソッドの実装をオーバーライドできます。

この操作は、認識プロパティに対しても実行できます。テスト・オブジェクト・クラスに、コントロールに関連する認識プロパティが含まれているにも関わらず UFT によって値が取得されていない場合、Agent Object を作成して必要な値を提供できます。詳細については、25 ページ「Extensibility コードの作成」を参照してください。

- ▶ **カスタム・テスト・オブジェクト・クラスを内部オブジェクトにマッピング**: 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 (ある場合) を、内部オブジェクトとして機能する既存の UFT 標準 Windows テスト・オブジェクト・クラスにマッピングします。UFT は、作成した Agent Object では固有のサポートが提供されないプロパティやメソッドに、内部オブジェクトの実装を使用します。

新しい Delphi テスト・オブジェクト・クラスを内部オブジェクトにマッピングすると、標準のメソッドとプロパティに対する既存の UFT GUI テスト サポートが使用可能になります。この機能は、TListView などのサブクラス化された標準 Windows コントロールのサポートを作成する際に便利です。既存の標準 Windows テスト・オブジェクト・クラスのリストについては、『HP Unified Functional Testing Object Model Reference』を参照してください。

UFT は、一部のダイアログ・ボックス (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、[ステップ ジェネレータ] など) で、使用可能なテスト・オブジェクト・クラスを環境ごとに表示します。このようなダイアログ・ボックスでカスタム Delphi テスト・オブジェクト・クラスを表示するには、ツールキット設定ファイルでも定義を行う必要があります。

この XML ファイルの構造と構文の詳細については、第3章「ツールキット設定 XML ファイルの構造について」を参照してください。

Delphi Extensibility コードの作成

Extensibility 用に作成した Delphi ユニットには、次の項目が必ず含まれています。

- ▶ 1 つまたは複数の Agent Object。Agent Object は、サポート対象となるコントロールのタイプごとに1つ含まれています。テスト対象アプリケーション内の Delphi コントロールと UFT を連携する Agent Object インタフェース。
- ▶ 各コントロールに対して適切な Agent Object を作成するファクトリ関数1つ。UFT は、コントロールと最初に対話する際に、コントロールに対応する Agent Object を生成するファクトリ関数を呼び出します。

一部のカスタム・コントロールでは、既存の Delphi テスト・オブジェクト・クラスにマッピングするだけで十分なサポートが提供される場合もあります。この場合、コントロール用に Agent Object を作成する必要はありません。

Extensibility コードの作成

カスタム・コントロールのサポートを目的に拡張 Delphi ユニットを作成するには、Delphi Add-in 付属のテンプレートを使用してください：

<UFT インストール・フォルダ>\dat\Extensibility\Delphi\ExtensibilityImplementationTemplate.pas

Extensibility コードで次の操作を行います。

- ▶ Unified Functional Testing Delphi Add-in 付属の AgentExtensibilitySDK ユニットをインポートして使用します：**<UFT インストール・フォルダ>\dat\Extensibility\Delphi\AgentExtensibilitySDK.pas**
- ▶ サポートするコントロールのタイプごとに Agent Object を作成します。Agent Object は、**TMicAO**、または Delphi Add-in Extensibility SDK で定義されている Agent Object 基本クラスを継承する必要があります。Agent Object で、コントロールに必要なテスト・オブジェクト・メソッドと認識プロパティをサポートする公開プロパティを作成します。詳細については、26 ページ「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する公開プロパティの操作」を参照してください。
- ▶ Delphi ユーザ・インタフェース・コントロールのオブジェクト参照を取得して新しい Agent Object を返すファクトリ関数を作成します。このファクトリ関数には、サポートの作成対象であるカスタム・コントロールを認識する機能と、適切な Agent Object を作成する機能が必要です。
- ▶ 拡張ユニットの初期化セクションで、**AddExtensibilityServer** API 関数を呼び出してファクトリ関数を Delphi Add-in に登録します。

詳細については、**<UFT インストール・フォルダ>\dat\Extensibility\Delphi\AgentExtensibilitySDK.pas** を参照してください。

作成したサポートを実行するには、作成した Extensibility ユニットとプリコンパイル済み Delphi Add-in エージェントを使用して、テスト対象の Delphi アプリケーションをコンパイルする必要があります。詳細については、第 2 章「ツールキット・サポート・セットのデプロイ」を参照してください。

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する公開プロパティの操作

UFT は、Agent Object とコントロール自体の公開プロパティを設定および取得することによって、アプリケーションのコントロールと対話します。UFT は、Agent Object の公開プロパティにアクセスしてから、必要に応じて Delphi オブジェクトの公開プロジェクトにアクセスします。

Agent Object の作成では、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とテスト・オブジェクト・メソッドをサポートする公開プロパティを作成します。たとえば、Agent Object で公開プロパティを作成することにより、コントロールの非公開の（パブリック）メンバ変数へのアクセスが可能になります。

次の予約プロパティは、テストおよびコンポーネントの記録と実行の実装で使用されます。

▶ **__QTPReplayMtd_**

このプレフィックスは、UFT テスト・オブジェクト・メソッドの実装を実装する Agent Object プロパティすべてに付加します。

▶ **__CellRect, __CellData, __TableContent**

グリッド・オブジェクトのサポートを実装するために使用します。詳細については、30 ページ「カスタム・グリッド・コントロールのサポートの作成」を参照してください。

▶ **__QTPRecording**

このプロパティは、UFT の記録機能のサポートを実装するために使用します。詳細については、28 ページ「UFT の記録機能のサポート」を参照してください。

テストとコンポーネントの記録と実行の実装については、次の項で説明します。

認識プロパティのサポート

サポート対象となる各認識プロパティに対して、Delphi コントロールまたは Agent Object に同じ名前の公開プロパティが存在することを確認してください。

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。したがって、認識プロパティをサポートする公開プロパティの名前には小文字のみを使用してください（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大文字が使用されている場合でも、小文字のみを使用します）。

UFT ユーザは、**GetROProperty** メソッドと **SetROProperty** メソッドを使用して Agent Object と Delphi コントロールの公開プロパティにアクセスします。さらにこの公開プロパティは、UFT テスト内のチェックポイントで検証でき、オブジェクト・スパイで表示できます。

2 つのアンダースコア（'__'）で始まるプロパティは、オブジェクト・スパイでは表示されず、チェックポイントおよび出力値で使用できません。このような隠しプロパティにアクセスするには、テスト・オブジェクト・メソッド **GetROProperty** および **SetROProperty** を使用するか、UFT で作成したユーザ定義関数で直接アクセスします。

テスト・オブジェクト・メソッドのサポート

サポート対象のテスト・オブジェクト・メソッドには、それぞれ **__QTPReplayMtd_<テスト・オブジェクト・メソッド名>** という名前の公開プロパティを Agent Object で作成します。たとえば、**__QTPReplayMtd_MyTOMethod** という名前の公開プロパティは、**MyTOMethod** テスト・オブジェクト・メソッドを実行するための実装を提供します。

UFT の記録機能のサポート

UFT の記録機能をサポートするには、Agent Object で **__QTPRecording** 公開プロパティを実装する必要があります。

記録セッション中、カスタム Delphi コントロールでイベントが発生すると、UFT はウィンドウ・メッセージを Agent Object に送信し、**__QTPRecording** プロパティを照会して対応する行を取得し、テストまたはコンポーネントに追加します。

ほとんどの場合、このプロパティを Extensibility コードで実装する必要はありません。記録機能をサポートする Agent Object を作成するには、**TMicRecordableAO** 基本クラスを継承する Agent Object を作成します。

Agent Object 基本クラス **TMicRecordableAO** は、**__QTPRecording** の実装によって次の操作を実行します。

- 1 UFT が渡すパラメータの処理。
- 2 **ProcessMessage** 関数を呼び出し、発生したイベントの記録の対象となるステップ（および記録モード）を特定します。
- 3 記録した情報を UFT で必要な形式に変換します。

作成する Agent Object では、**ProcessMessage** 関数の実装のみを行います。

UFT は、**__QTPRecording** プロパティにアクセスすると、ウィンドウ・メッセージ・パラメータとウィンドウ・ハンドルを Agent Object に渡し、記録したステップの情報（操作と引数）と記録モードが格納された配列（safearray タイプ）を取得します。記録モードの詳細については、<UFT インストール・フォルダ>\dat\Extensibility\Delphi\AgentExtensibilitySDK.pas を参照してください。

入力パラメータ

パラメータ・インデックス	値のタイプ	説明
0	VT_I4	ウィンドウ・ハンドル
1	VT_I4	メッセージ
2	VT_I4	IParam
3	VT_I4	wParam

出力の safearray 形式

パラメータ・インデックス	値のタイプ	説明
0	VT_I4	記録モード
1	VT_BSTR	記録するテスト・オブジェクト・メソッド
2..end		テスト・オブジェクト・メソッドの引数

注: カスタム・コントロールをサポートするカスタム・テスト・オブジェクト・クラスを作成すると、記録をトリガするイベントのレベルを、ツールキット設定ファイル内の **ExtObjRecFilter** 属性で指定できます。詳細については、49 ページ「テスト・オブジェクト・クラスを内部オブジェクトにマッピング」を参照してください。

カスタム・グリッド・コントロールのサポートの作成

カスタム・グリッド・コントロールのサポートを作成するには、コントロールをテスト・オブジェクト・クラスにマッピングし、サポートを実装する Agent Object を作成して、このコントロールをテーブルとして処理するように UFT に指示する（オプション）必要があります。

<UFT インストール・フォルダ>\samples\DelphiGridExtSample に収録されているツールキット・サポート・セットのサンプルでは、カスタム・グリッド・コントロール（TStringDrawGrid）のサポートを提供しています。この項の内容を読んだ後でサンプルを参照すると、カスタム・グリッド・コントロールのサポートを作成する方法をよく理解できます。詳細については、16 ページ「Delphi Add-in Extensibility サンプルの使用」を参照してください。

カスタム・グリッド・コントロールをテスト・オブジェクト・クラスにマッピング

カスタム・グリッド・コントロールは、DelphiTable テスト・オブジェクト・クラス、またはテスト・オブジェクト設定ファイルで定義したカスタム・グリッド・テスト・オブジェクト・クラスにマッピングできます。

カスタム・グリッド・コントロールを DelphiTable テスト・オブジェクト・クラスにマッピングする作業では、32 ページ「カスタム・グリッド・コントロールをテーブルとして処理することを UFT に指示」に説明されている定義を行う必要はありません。

カスタム・グリッド・コントロールをサポートする Agent Object の作成

Agent Object は、グリッド・コントロールにマッピングされるテスト・オブジェクト・クラスで定義されているすべてのテスト・オブジェクト・メソッドと認識プロパティに対してサポートを提供する必要があります。具体的には、SetCellData や GetCellData などのグリッド操作や、テスト・オブジェクト・クラスで定義されているメソッドとプロパティが含まれます。

Unified Functional Testing Delphi Add-in は、グリッド・コントロールのサポートに必要な設計を実装するテスト・オブジェクト拡張（Mercury.DelphiTableSrv）を提供します。

この拡張は、SetCellData、GetCellData、SelectCell などの基本的なグリッド操作を実装します。この拡張では、公開プロパティ __CellRect、__CellData、__TableContent、RowCount、ColCount を使用して、上記のテスト・オブジェクト・メソッドを Agent Object に委譲します。

カスタム・グリッド・コントロールのサポートを作成するには **TMicGridAOBase** または **TCustomGridAOBase** を継承し、上記の公開プロパティを実装する Agent Object の作成が必要です (**TMicGridAOBase** または **TCustomGridAOBase** 基本クラスの詳細は、**<UFT インストール・フォルダ>\dat\Extensibility\Delphi\AgentExtensibilitySDK.pas** を参照してください)。

さらに、カスタム・グリッドをサポートするグリッド・テスト・オブジェクト拡張の使用を UFT に指示する必要があります。詳しい使用方法については、32 ページ「カスタム・グリッド・コントロールをテーブルとして処理することを UFT に指示」を参照してください。

注：カスタム・コントロールをカスタム・テスト・オブジェクト・クラスにマッピングする場合、テスト・オブジェクト・クラスで定義済みのテスト・オブジェクト・メソッドと認識プロパティの中でほかに必要なものがあれば、それをサポートする Agent Object を作成します。

グリッドのサポートに必要な公開プロパティを実装

カスタム・グリッド・コントロール用に作成するサポートは、Delphi Add-in グリッド・テスト・オブジェクト拡張をベースとします。したがって、Agent Object では次の公開プロパティを実装する必要があります。

- ▶ **__CellRect** : セルの位置を示す矩形を、x;y;width;height;; (x と y は矩形の左上角の座標) という形式で返します。
- ▶ **__CellData** : セル内の値 (String 型) を設定および取得します。Agent Object 基本クラス **TMicGridAOBase** は、このプロパティを実装することによって抽象関数 **GetCellDataEx** および **SetCellDataEx** を呼び出します。これらの関数は、Agent Object 用に作成した派生クラスで実装します。
- ▶ **__TableContent** : テーブル全体のデータを指定のファイルに書き込み、処理の成否を true または false で返します。UFT は、パラメータで指定されたファイルを Agent Object に渡します。テーブル内のデータは String 形式でファイルに書き込まれ、セル・データはタブ区切り、行は改行文字で区切られます。

グリッド・テスト・オブジェクト拡張は、このプロパティを使用してテーブル・チェックポイントをサポートします。Agent Object 基本クラスである **TMicGridAObBase** はこのプロパティを実装することによって、**CaptureTableEx** 抽象関数を呼び出します。この関数は、Agent Object 用に作成した派生クラスで実装します。

カスタム・グリッド・コントロールをテーブルとして処理することを UFT に指示

注： カスタム・グリッド・コントロールを DelphiTable テスト・オブジェクト・クラスにマッピングした場合、この項で説明している定義を作成する必要はありません。

ツールキット設定 XML ファイルで次の内容を定義します。

- ▶ Delphi Add-in グリッド・テスト・オブジェクト拡張を使用して、カスタム・グリッド・コントロール（またはカスタム・グリッド・テスト・オブジェクト・クラスにマッピングされているすべてのコントロール）をサポートすることを UFT に指示します。
- ▶ カスタム・グリッド・テスト・オブジェクト・クラスが（テスト・オブジェクト設定ファイルで）定義した場合は、チェックポイントおよび出力値の生成時にこのタイプのテスト・オブジェクトをテーブル・テスト・オブジェクトとして処理することを UFT に指示します。

このタイプのコントロールのグリッド・テスト・オブジェクト拡張を使用することを UFT に指示するには、次の手順で行います。

次の定義をツールキット設定 XML ファイルに追加します (**太字**部分が追加が必要な箇所です)。

```
<MicTest>
  <Key Name="Packages">
    <Key Name="DelphiPackage">
      <Key Name="CustomServers">
        <Key Name="TCustomGridNativeClass">
          <Value Name="CustReplayProgID"
            Type="BSTR"> Mercury.DelphiTableSrv
          </Value>
        </Key>
      </Key>
    </Key>
  </Key>
  ...
</MicTest>
```

- ▶ **TCustomGridNativeClass** の部分には、サポート対象となるグリッド・コントロールのウィンドウ・クラス名を指定します。
- ▶ **Name="CustomServers"** の **Key** 要素内で、サポート対象となるカスタム・グリッド・クラスごとに **Key** 値を作成します。

または、**Key** 要素を 1 つ作成し、あるカスタム・テスト・オブジェクト・クラスにマッピングされているカスタム・コントロールすべてにグリッド・テスト・オブジェクト拡張を使用するよう UFT に指示することも可能です。これを行うには、**TCustomGridNativeClass** の部分にカスタム・グリッド・テスト・オブジェクト・クラスの名前を指定し、**MC2CSMapping_** という文字列をプレフィックスとして付加します (たとえば、**MC2CSMapping_DelphiCustomTable**)。

チェックポイントおよび出力値の生成時に、このタイプのテスト・オブジェクトをテーブル・テスト・オブジェクトとして処理することを UFT に指示するには、次の手順で行います。

ツールキット設定ファイルで、カスタム・テスト・オブジェクト・クラスを内部オブジェクトにマッピングする部分に次の定義を追加します (**太字**部分が追加が必要な箇所です)。

```
<MicTest>
...
<Key Name="Test Objects">
  <Key Name="TheDelphiCustomTestObject you are mapping">
...
  <!-- enables table checkpoint and output value -->
  <Key Name="CustomStepCfg">
    <Value Name="Checkpoint" Type="BSTR">Mercury.MultiVerUI</Value>
    <Value Name="Output Value" Type="BSTR">Mercury.MultiVerUI</Value>
  </Key>
  <!-- Enables use of the Define/Modify Row Range dialog box -->
  <Key Name="CustomStepCfgDlg">
    <Value Name="Checkpoint" Type="BSTR">
      Mercury.TableTOConcigUI
    </Value>
    <Value Name="Output Value" Type="BSTR">
      Mercury.TableTOConcigUI
    </Value>
  </Key>
</Key>
</Key>
</MicTest>
```

カスタム Delphi グリッド・テスト・オブジェクト・クラスを複数定義する場合は、それぞれのテスト・オブジェクト・クラスで定義した **Key** 要素内に上記の定義を追加してください。

これにより、このタイプのテスト・オブジェクトに対して、UFT は [テーブルチェックポイントのプロパティ] ダイアログ・ボックスと [テーブル出力値のプロパティ] ダイアログ・ボックスを表示します。

ツールキット設定 XML ファイルの構造と構文の詳細については、第 3 章「ツールキット設定 XML ファイルの構造について」を参照してください。

カスタム Delphi コントロールをサポートする手順

この項では、UFT Delphi Add-in Extensibility ツールキット・サポート・セットを作成する手順を詳しく説明します。

注：既存の Delphi テスト・オブジェクト・クラスにカスタム・コントロールをマッピングするだけで、サポートが提供されるケースもあります。このような場合には、以下のプロセスで手順 1～2 のみを実行し、**MicDelphiAgent.pas** モジュールを使ってテスト対象アプリケーションをコンパイルしてください（『HP Unified Functional Testing アドイン・ガイド』の Delphi の項の説明を参照）。

これ以外の場合には、必要なテスト・オブジェクト・メソッドと認識プロパティがテスト・オブジェクト・クラスにすべて含まれているとしても、カスタム・コントロールを既存のテスト・オブジェクト・クラスにマッピングするだけでは不十分です。この場合には、カスタム・コントロールをサポートするためのテスト・オブジェクト・クラスを新規作成する操作（手順 3～4）は不要ですが、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する extensibility Delphi ユニットの作成は必要です（手順 6～9）。

カスタム・コントロール用のサポート・セットを作成するには、次の手順で行います。

1 ツールキット設定ファイルの作成

- a UFT テストおよびコンポーネント内のカスタム・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・クラスを選択します。カスタム・コントロールは、既存の Delphi テスト・オブジェクト・クラス、または手順 3 で定義するカスタム・テスト・オブジェクト・クラスにマッピングできます。
- b カスタム・コントロールを表すために新規作成するカスタム・テスト・オブジェクト・クラス（該当する場合）と、内部オブジェクトとして機能する既存の標準 Windows テスト・オブジェクト・クラスを決定します。
- c サンプルのツールキット設定ファイル（<UFT インストール・フォルダ>**samples\DelphiExtSample\ToolkitSupportSet\TrackerSampleToolkitCfg.xml**）をコピーして、ユーザ定義のツールキット設定ファイルを作成します。この XML ファイルの構造と構文の詳細については、第 3 章「ツールキット設定 XML ファイルの構造について」を参照してください。

- d サポート対象となるカスタム・コントロールごとに、**TTrackBar** カスタム・コントロールをカスタム・テスト・オブジェクト・クラス **DelphiTrackBar** にマッピングする **Value** 要素をコピーします。**TTrackBar** と **DelphiTrackBar** は適切な名前に置き換えてください。
- e 作成するカスタム・テスト・オブジェクト・クラスごとに（該当する場合）カスタム・テスト・オブジェクト・クラス **DelphiTrackBar** の設定を含む **Key** 要素をコピーします。テスト・オブジェクト名、**InnerProgId**, **InnerMicClass** の値は適切なオブジェクト名に置き換えてください。

注：サポート対象のコントロールの機能を提供できる内部オブジェクトを選択する必要があります。ほとんどの場合、内部オブジェクトは **WinObject** のような汎用テスト・オブジェクト・クラスです。サポート対象となるコントロールのタイプに適したテスト・オブジェクト・クラスがわかっている場合は、それを使用してください（たとえば、**TListView** をサブクラス化するコントロールのサポートを作成するには、**WinListView** テスト・オブジェクトを内部オブジェクトとして使用します）。

内部オブジェクトに適切なテスト・オブジェクト・クラスを使用しているかどうかをチェックするには、UFT の [オブジェクトの割り当て] ダイアログ・ボックスでこのテスト・オブジェクト・クラスをコントロールにマッピングし、コントロールでオブジェクト・スパイを使用できることを確認してください。

[オブジェクトの割り当て] ダイアログ・ボックスの詳細については、『HP Unified Functional Testing ユーザーズ・ガイド』を参照してください。

2 ツールキット設定ファイルのデプロイ

ツールキット設定 XML ファイルを <UFT インストール・フォルダ>\dat\Settings フォルダにコピーします。

3 テスト・オブジェクト設定ファイルの作成

すべてのカスタム・コントロールを既存の Delphi テスト・オブジェクト・クラスにマッピングする場合は、手順 5 に進みます。

それ以外の場合は、テスト・オブジェクト設定 XML ファイルを作成して、カスタム・テスト・オブジェクト・クラスを定義します。各カスタム・テスト・オブジェクト・クラスの記述プロパティ、認識プロパティ、テスト・オブジェクト・メソッドを定義します。詳細については、19 ページ「テスト・オブジェクト設定 XML ファイルの作成」を参照してください。

4 テスト・オブジェクト設定ファイルのデプロイ

テスト・オブジェクト設定 XML ファイルを作成したら、**<UFT インストール・フォルダ>\dat\Extensibility\Delphi** フォルダにコピーします。

5 テスト・オブジェクトのマッピングの確認

UFT を開き、GUI テストを開いて次の内容を確認します。

- ▶ オブジェクト・スパイを開いた時に、カスタム・コントロールが UFT によって正しく認識される。
- ▶ UFT によるカスタム・コントロールの学習が完了すると、オブジェクト・リポジトリに対応するテスト・オブジェクトが追加される。

注：カスタム・コントロールを既存の Delphi テスト・オブジェクトにマッピングするだけで十分なサポートが提供され、コントロールに関するテストおよびコンポーネントの作成と実行に対応できる場合は、以下の手順を実行する必要はありません。

6 基本的な拡張 Delphi ユニットの作成

Agent Object 実装テンプレートである **<UFT インストール フォルダ>\dat\Extensibility\Delphi\Extensibility\ImplementationTemplate.pas** を使用して、拡張コードを作成します。

表示する公開プロパティをいくつか Agent Object で定義します。

7 拡張コードのコンパイル

プレコンパイル済みの Unified Functional Testing Delphi Add-in エージェントと拡張ユニットを使用して、テスト対象アプリケーションをコンパイルします。42 ページ「拡張コードのコンパイル」の手順を実行してください。

8 拡張コードの機能の確認

Delphi アプリケーションを実行し、Agent Object が正しく動作することをチェックします。これには、手順 6 で定義したプロパティが UFT オブジェクト・スパイで表示されることを確認してください。

9 Extensibility Delphi ユニットの作成の完了

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とテスト・オブジェクト・メソッドのサポートに必要なその他の Agent Object 公開プロパティを実装します。

第 2 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 GUI テスト 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。これは、作成したツールキット・サポート・セットを UFT が使用できるようにする手順であり、これによってツールキット内のコントロールを認識し、コントロールに関するテストの実行が可能になります。

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・セットを 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、Unified Functional Testing Delphi 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 (40ページ)
- ▶ XML ファイルの格納場所 (41ページ)
- ▶ 拡張コードのコンパイル (42ページ)
- ▶ デプロイ済みサポートの変更 (42ページ)
- ▶ デプロイ済みサポートの削除 (43ページ)

カスタム・ツールキット・サポートのデプロイについて

UFT Delphi Add-in Extensibility ツールキット・サポート・セットをデプロイするには、作成した XML ファイルを UFT がインストールされているコンピュータ上の正しい場所に配置し、プリコンパイル済みの Unified Functional Testing Delphi Add-in エージェントと作成した拡張ユニットを使ってテスト対象アプリケーションをコンパイルする必要があります。

詳細については、次を参照してください。

- ▶ 41 ページ「XML ファイルの格納場所」
- ▶ 42 ページ「拡張コードのコンパイル」

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、Delphi Add-in は UFT ユーザが作成したカスタム・コントロールをほかの Delphi オブジェクトと同様に認識できるようになります。

XML ファイルの格納場所

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 XML ファイルを格納する必要があります。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	場所
テスト・オブジェクト設定ファイル	<ul style="list-style-type: none"> ▶ <UFT インストール・フォルダ>\dat\Extensibility\Delphi ▶ <ALM 用 UFT Add-in のインストール・フォルダ>\dat\Extensibility\Delphi (任意。ALM 用 UFT アドインがインストールされている場合のみ必要です)
ツールキット設定ファイル	<UFT インストール・フォルダ>\dat\Settings
カスタム・テスト・オブジェクト・クラスのアイコン・ファイル (オプション)	<p>.dll または .ico ファイルを、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。</p> <p>推奨される場所: <UFT インストール・フォルダ>\dat\Extensibility\Delphi\Toolkits\ <カスタム・ツールキット名>\res</p> <p>テスト・オブジェクト設定ファイル内で場所を指定します。</p>
テスト・オブジェクト・クラスのヘルプ・ファイル (オプション)	<p>.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。</p> <p>推奨される場所: <UFT インストール・フォルダ>\dat\Extensibility\Delphi\Toolkits\ <カスタム・ツールキット名>\help</p> <p>テスト・オブジェクト設定ファイル内で場所を指定します。</p>

注: テスト・オブジェクト設定ファイルでは、格納場所を相対パスで指定できます。詳細については、UFT Test Object Schema ヘルプ (Delphi Add-in Extensibility ヘルプで利用可能) を参照してください。

拡張コードのコンパイル

拡張 Delphi ユニットを作成した場合、プリコンパイル済みの Unified Functional Testing Delphi Add-in エージェントと拡張ユニットを使って、テスト対象アプリケーションをコンパイルする必要があります。

これを行うために、次の手順を実行します。

- 1 <UFT インストール・フォルダ>\dat\Extensibility\Delphi フォルダをアプリケーションのプロジェクトの検索パスに追加するか、<UFT インストール・フォルダ>\dat\Extensibility\Delphi フォルダの内容をプロジェクト・フォルダにコピーします。
- 2 **MicDelphiAgent** をアプリケーションのプロジェクト・ファイルの **Uses** セクションに追加します。
- 3 InfoPower の **TwwDBGrid** がアプリケーションに含まれている場合、アプリケーションのプロジェクト・ファイル内の **Uses** セクション内で、**MicDelphiAgent** の後に **MicWWSupport** を追加します。
- 4 拡張コードの場所をアプリケーションのプロジェクトの検索パスに追加するか、プロジェクト・フォルダにファイルを配置します。
- 5 拡張ユニットの名前を、アプリケーションのプロジェクト・ファイルの **Uses** セクションに追加します。
- 6 Delphi アプリケーション・プロジェクトをコンパイルします。

デプロイ済みサポートの変更

作成した拡張 Delphi ユニットを変更する場合は、テスト対象アプリケーションを再コンパイルし（プリコンパイル済みの Unified Functional Testing Delphi Add-in エージェントと作成した拡張ユニットを使用）、Delphi アプリケーションを再度実行すると、変更内容が有効になります。

デプロイしたツールキット・サポート・セットの XML ファイルを変更する場合は、UFT を終了してから再度起動すると、変更内容が有効になります。

デプロイ済みサポートの削除

デプロイメントが完了した後でカスタム・ツールキットのサポートを UFT から削除するには、<UFT インストール・フォルダ>\dat\Settings からツールキット設定ファイルを削除し、作成した extensibility Delphi ユニットを使用しないでアプリケーションをコンパイルします。

テスト・オブジェクト設定ファイル内に、カスタム・コントロールに相当するテスト・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）、次のファイルを削除できます。

<UFTインストール・フォルダ>\dat\Extensibility\Delphi（および<UFT Add-in for ALM
インストール・フォルダ>\dat\Extensibility\Delphi（該当する場合））

第 3 章

ツールキット設定 XML ファイルの構造について

Delphi ツールキット・サポート・セットにはツールキット設定ファイルが必要です。このファイルは、カスタム Delphi コントロールを、UFT での表示およびサポートに使用するテスト・オブジェクト・クラスにマッピングします。

本章の内容

- ▶ ツールキット設定 XML ファイルについて (45ページ)
- ▶ アプリケーション・コントロールをテスト・オブジェクト・クラスにマッピング (47ページ)
- ▶ テスト・オブジェクト・クラスを内部オブジェクトにマッピング (49ページ)
- ▶ ツールキット設定ファイル全体のサンプル (55ページ)

ツールキット設定 XML ファイルについて

ツールキット設定 XML ファイルでは、次の 2 種類のマッピングを定義する必要があります (23 ページ「ツールキット設定 XML ファイルの作成」を参照してください)。

- ▶ アプリケーション・コントロールをテスト・オブジェクト・クラスにマッピング
- ▶ カスタム・テスト・オブジェクト・クラスを内部オブジェクトにマッピング

ツールキット設定 XML ファイルのルート要素は **MicTest** 要素です。これには **Key** 要素が 2 つあり、それぞれ異なる **Name** 属性 (**Packages** と **Test Objects**) が指定されています。

```
<?xml version="1.0"?>
<MicTest>
  <Key Name="Packages">
  ...
  </Key>

  <Key Name="Test Objects">
  ...
  </Key>
</MicTest>
```

MicTest 要素の 2 つの **Key** 要素により XML ファイルは 2 つのセクションに分かれ、それぞれで異なるタイプのマッピングが定義されます。

- ▶ **Name="Packages"** の **Key** 要素では、アプリケーション・コントロールをテスト・オブジェクトにマッピングします。

組み込みのグリッド・テスト・オブジェクト拡張を使用する場合には、このセクションでカスタム・グリッド・コントロール（またはカスタム・グリッド・テスト・オブジェクト・クラス）を **Mercury.DelphiTableSrv** にマッピングします。組み込みの拡張により、グリッド・コントロールのサポートに必要な設計のほとんどを実装できます。詳細については、30 ページ「カスタム・グリッド・コントロールのサポートの作成」を参照してください。

- ▶ **Name="Test Objects"** の **Key** 要素では、カスタム・テスト・オブジェクト・クラスを内部オブジェクトにマッピングします。

またこのセクションでは、カスタム・テスト・オブジェクト・クラスに関する追加設定を行います。これには、テスト・オブジェクト・クラスを特定のダイアログ・ボックスで表示する操作や、チェックポイントの作成時にテーブルとして処理する操作を UFT に指示する設定などがあります。

各セクションの構造と構文については、次を参照してください。

- ▶ 47 ページ「アプリケーション・コントロールをテスト・オブジェクト・クラスにマッピング」
- ▶ 49 ページ「テスト・オブジェクト・クラスを内部オブジェクトにマッピング」

アプリケーション・コントロールをテスト・オブジェクト・クラスにマッピング

ツールキット設定 XML ファイルのルート要素である **MicTest** では、**Name="Packages"** の **Key** 要素の下に次のような構造を作成します。

```
<?xml version="1.0"?>
<MicTest>
  <Key Name="Packages">
    <Key Name="DelphiPackage">
      <Key Name="ClassPatterns">
        <!-- One Value element for each custom control that you want to map -->
        <!-- This example shows the mapping for the TStringDrawGrid custom control.-->
        <Value Name="TStringDrawGrid" Type="BSTR">DelphiCustomTable</Value>
      </Key>
    <Key Name="CustomServers">
      <!-- One Key element for each custom grid control that you want to support -->
      <!-- This example shows the definition for the TStringDrawGrid custom control.-->
      <Key Name="TStringDrawGrid">
        <Value Name="CustReplayProgID"
              Type="BSTR"> Mercury.DelphiTableSrv
        </Value>
      </Key>
    </Key>
  </Key>
  ...
</MicTest>
```

Name 属性が **DelphiPackage** である **Key** 要素の下に、**ClassPatterns Key** 要素と、**CustomServers Key** 要素（オプション）を作成します。

ClassPatterns の Key 要素

Name 属性が **ClassPatterns** である **Key** 要素では、サポート対象となるカスタム・コントロールのタイプごとに **Value** 要素を1つ定義します。

各 **Value** 要素を次のように定義します。

- ▶ **Name** 属性には、カスタム・コントロールのウィンドウ・クラス名を指定します。
- ▶ **Type** 属性は **BSTR** に設定します。
- ▶ 要素の値には、テストおよびコンポーネントでコントロールを表すために **UFT** が使用するカスタム **Delphi** テスト・オブジェクト・クラスを指定します。

CustomServers の Key 要素

カスタム・グリッド・コントロールのサポートの作成では、**Name** 属性が **CustomServers** である **Key** 要素を作成します。この **Key** 属性内では、サポート対象となるカスタム・グリッド・コントロールごとに **Key** 要素を1つ定義します。

各カスタム・グリッド・コントロールの **Key** 要素は、次のように定義します。

- ▶ **Name** 属性には、カスタム・コントロールのウィンドウ・クラス名を含めます。
- ▶ **Value** 要素を次のように定義します。

```
<Value Name="CustReplayProgID" Type="BSTR"> Mercury.DelphiTableSrv </Value>
```

または、**Key** 要素を1つ作成し (**CustomServers** の **Key** 要素内)、あるカスタム・テスト・オブジェクト・クラスにマッピングされているカスタム・コントロールすべてにグリッド・テスト・オブジェクト拡張を使用する定義も可能です。これを行うには、**Key** 要素の **Name** 属性を **MC2CSMapping_<カスタム・グリッド・テスト・オブジェクト・クラス名>** に設定します。

カスタム・グリッド・コントロールのサポートの詳細については、30 ページ「カスタム・グリッド・コントロールのサポートの作成」を参照してください。

テスト・オブジェクト・クラスを内部オブジェクトにマッピング

ツールキット設定 XML ファイルのルート要素である **MicTest** 内で、**Name="Test Objects"** である **Key** 要素の下に、マッピング対象のカスタム Delphi テスト・オブジェクト・クラスごとに **Key** 要素を1つずつ定義します。

この **Key** 要素の **Name** 属性では、カスタム Delphi テスト・オブジェクト・クラスの名前を含んでいる必要があります。たとえば、下の例は、**DelphiTrackBar** テスト・オブジェクト・クラスを、内部オブジェクトとして機能する **WinObject** テスト・オブジェクト・クラスにマッピングしているテスト・オブジェクト設定ファイルの一部です。

```
<?xml version="1.0"?>
<MicTest>
...
  <Key Name="Test Objects">
    <!-- mapping for DelphiTrackBar -->
    <Key Name="DelphiTrackBar">
...
    </Key>
  </Key>
</MicTest>
```

第 3 章・ツールキット設定 XML ファイルの構造について

各カスタム Delphi テスト・オブジェクト・クラスの **Key** 要素では、次の **Value** 要素を定義します。

Name 属性	Type 属性	指定	値
BottomLevelObject	I4	オプション	このタイプのテスト・オブジェクトの下位にはかのオブジェクトを指定できるかどうかを示します。 取り得る値： ▶ 0 ：このテスト・オブジェクトは最下位のオブジェクトではありません。子オブジェクトを指定できます。(標準設定) ▶ 1 ：このテスト・オブジェクトは最下位のオブジェクトです。子オブジェクトを指定できません。
CLSID	BSTR	必須	必要な値： {A990252E-48C1-4d6c-9B55-4701BC29919C}
(Default)	BSTR	必須	マッピングの対象となるカスタム Delphi テスト・オブジェクト・クラスの名前。

Name 属性	Type 属性	指定	値
ExtObjRecFilter	I4	オプション	<p>記録操作で UFT が Agent Object に渡すウィンドウ・メッセージを指定します。</p> <p>取り得る値：</p> <ul style="list-style-type: none"> ▶ 0：コントロールのウィンドウに対するメッセージのみを、記録用として Agent Object に渡します（ほかのメッセージはすべて無視します）。 ▶ 1：すべてのウィンドウ・メッセージを記録用として Agent Object に渡します。 ▶ 2：コントロールとその子に対するメッセージのみを、記録用として Agent Object に渡します（ほかのメッセージはすべて無視します）。 <p>この要素が定義されていないと、すべての記録は内部オブジェクトによって処理されます。</p>
InnerMicClass	BSTR	必須	<p>内部オブジェクトとして使用する既存の標準 Windows テスト・オブジェクト・クラスの名前。</p> <p>使用可能なテスト・オブジェクト・クラスのリストは、『HP Unified Functional Testing Object Model Reference』の Standard Windows の項を参照してください。</p>

Name 属性	Type 属性	指定	値
InnerProgId	BSTR	必須	Mercury.<InnerMicClass> 例外 ：次の InnerMicClass の値には、 InnerProgId を次のように設定します。 Window :Mercury.StdWindow Dialog :Mercury.WinDialog Static :Mercury.WinStatic Desktop :Mercury.MicDesktop
tag query name	BSTR	必須	必要な値 : delphi_name

Test Objects Key 要素で行うその他の定義

ここでは、カスタム Delphi テスト・オブジェクト・クラスの **Key** 要素内で、オプションで定義できる要素について説明します。

- ▶ UFT は、一部のダイアログ・ボックス（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、[ステップジェネレータ] ダイアログ・ボックスなど）で、使用可能なテスト・オブジェクト・クラスを環境ごとに表示します。このようなダイアログ・ボックスでカスタム Delphi テスト・オブジェクト・クラスを表示するには、それぞれのカスタム Delphi テスト・オブジェクト・クラスの **Key** 要素に次の行を追加してください。

```
<Key Name="Info">
  <Value Name="package" Type="BSTR">DelphiPackage</Value>
</Key>
```

- ▶ UFT がカスタム Delphi テスト・オブジェクト・クラスをグリッド (テーブル) として処理するには、このテスト・オブジェクト・クラスの **Key** 要素に、さらに要素を追加する必要があります。詳細については、34 ページ「チェックポイントおよび出力値の生成時に、このタイプのテスト・オブジェクトをテーブル・テスト・オブジェクトとして処理することを UFT に指示するには、次の手順で行います。」を参照してください。

- ▶ UFT の [チェックポイントのプロパティ] ダイアログ・ボックスには、[**チェックポイントのタイムアウト**] という値があります (秒単位)。カスタム・テスト・オブジェクト・クラスでチェックポイントを新しく作成する際に標準設定値が適用されますが、この値のカスタマイズが可能です (カスタマイズしない場合の標準設定値は **0**)。チェックポイントのタイムアウトの標準設定値を設定するには、カスタム Delphi テスト・オブジェクト・クラスの **Key** 要素に、次に示す **Key** 要素を追加します。この例では **10** 秒に設定していますが、適用したい値を指定してください。

```
<Key Name="CheckpointTimeout">  
  <Value Name="DefaultTimeout" Type="I4">10<!--0xA--></Value>  
</Key>
```

Test Objects Key 要素

次の例では、**DelphiTrackBar** テスト・オブジェクト・クラスを既存の **WinObject** テスト・オブジェクト・クラスにマッピングする **Key** 要素全体を示しています。

```
<?xml version="1.0"?>
<MicTest>
...
<Key Name="Test Objects">
  <!-- The mapping definitions for the DelphiTrackBar test object -->
  <Key Name="DelphiTrackBar">
    <!-- The name of the custom Delphi test object class being mapped.-->
    <Value Name="(Default)" Type="BSTR">DelphiTrackBar</Value>
    <!-- The name (and ProgId) of the Standard Windows inner object.-->
    <Value Name="InnerProgId" Type="BSTR">Mercury.WinObject</Value>
    <Value Name="InnerMicClass" Type="BSTR">WinObject</Value>
    <!-- ExtObjRecFilter Value element is set to 0:only messages addressed to the
        control's window are passed to the Agent Object for recording.
        All other messages are ignored.-->
    <Value Name="ExtObjRecFilter" Type="I4">0</Value>
    <!-- BottomLevelObject Value element is set to 1:Test objects of this class
        do not have child objects.-->
    <Value Name="BottomLevelObject" Type="I4">1</Value>
    <!-- These elements are defined identically for every mapped test object class.-->
    <Value Name="tag query name" Type="BSTR">delphi_name</Value>
    <Value Name="CLSID" Type="BSTR">
      {A990252E-48C1-4d6c-9B55-4701BC29919C}
    </Value>
    <Key Name="Info">
      <Value Name="package" Type="BSTR">DelphiPackage</Value>
    </Key>
  </Key>
</MicTest>
```

ツールキット設定ファイル全体のサンプル

ツールキット設定ファイルの例を次に示します。このファイルでは、**TTrackBar** Delphi オブジェクトを新しい **DelphiTrackBar** テスト・オブジェクト・クラスにマッピングし、**DelphiTrackBar** テスト・オブジェクト・クラスを既存の **WinObject** テスト・オブジェクト・クラスにマッピングしています。

```
<?xml version="1.0"?>
<MicTest>
  <!-- Mapping the window class of the application controls to the
        custom Delphi test object classes that should represent them in UFT.-->
  <Key Name="Packages">
    <Key Name="DelphiPackage">
      <Key Name="ClassPatterns">
        <!-- Mapping the TTrackBar control to the DelphiTrackBar test object -->
        <Value Name="TTrackBar" Type="BSTR">DelphiTrackBar</Value>
      </Key>
    </Key>
  </Key>
  <!-- Mapping the custom Delphi test object classes to inner objects.-->
  <Key Name="Test Objects">
    <!-- The mapping definitions for the DelphiTrackBar test object -->
    <Key Name="DelphiTrackBar">
      <!-- The name of the custom Delphi test object class being mapped.-->
      <Value Name="(Default)" Type="BSTR">DelphiTrackBar</Value>
      <!-- The name (and ProgId) of the Standard Windows inner object.-->
      <Value Name="InnerProgId" Type="BSTR">Mercury.WinObject</Value>
      <Value Name="InnerMicClass" Type="BSTR">WinObject</Value>
      <!-- ExtObjRecFilter Value element is set to 0:only messages addressed to the
            control's window are passed to the Agent Object for recording.-->
      <Value Name="ExtObjRecFilter" Type="I4">0</Value>
      <!-- BottomLevelObject Value element is set to 1:Test objects of this class
            do not have child objects.-->
      <Value Name="BottomLevelObject" Type="I4">1</Value>
      <!-- These elements are defined identically for every mapped test object class.-->
      <Value Name="tag query name" Type="BSTR">delphi_name</Value>
      <Value Name="CLSID" Type="BSTR">
        {A990252E-48C1-4d6c-9B55-4701BC29919C}
      </Value>
      <Key Name="Info">
        <Value Name="package" Type="BSTR">DelphiPackage</Value>
      </Key>
    </Key>
  </Key>
</MicTest>
```

