HP QuickTest Professional for Business Process Testing

ソフトウェア・バージョン : 10.00

ユーザーズ・ガイド

製造部品番号 : T6511-99024
ドキュメント発行日 : 2009 年 1 月（英語版）
ソフトウェア・リリース日 : 2009 年 1 月（英語版）

利用条件

保証

HP の製品およびサービスの保証は，かかる製品およびサービスに付属する明示的な保証の声明において定められている保証に限ります。本ドキュメントの内容は，追加の保証を構成するものではありません。HP は，本ドキュメントに技術的な間違いまたは編集上の間違い，あるいは欠落があった場合でも責任を負わないものとします。

本ドキュメントに含まれる情報は，事前の予告なく変更されることがあります。

制限事項

本コンピュータ・ソフトウェアは，機密性があります。これらを所有，使用，または複製するには，HP からの有効なライセンスが必要です。FAR 12.211 および 12.212 に従って，商用コンピュータ ソフトウェア，コンピュータ ソフトウェアのドキュメント，および商用アイテムの技術データは，HP の標準商用ライセンス条件に基づいて米国政府にライセンスされています。

サードパーティ Web サイト

HP は，補足情報の検索に役立つ外部サードパーティ Web サイトへのリンクを提供します。サイトの内容と利用の可否は予告なしに変更される場合があります。HP は，サイトの内容または利用の可否について，いかなる表明も保証も行いません。

著作権

© 1992 - 2009 Hewlett-Packard Development Company, L.P.

商標

Adobe® および Acrobat® は，Adobe Systems Incorporated の商標です。

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，Sun Microsystems, Inc. の米国商標です。

Microsoft®，Windows®，Windows NT® および Windows XP® は，Microsoft Corporation の米国登録商標です。

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。

文書の更新

本書のタイトル・ページには，次の識別情報が含まれています。

- ソフトウェアのバージョンを示すソフトウェア・バージョン番号
- ドキュメントが更新されるたびに更新されるドキュメント発行日
- 本バージョンのソフトウェアをリリースした日付を示す，ソフトウェア・リリース日付

最新のアップデートまたはドキュメントの最新版を使用していることを確認するには，**http://h20230.www2.hp.com/selfsolve/manuals** を参照します。

このサイトでは，HP Passport に登録してサインインする必要があります。HP Passport ID の登録は，以下の Web サイトにアクセスしてください。

http://h20229.www2.hp.com/passport-registration.html

または，HP Passport のログイン・ページの［**New users - please register**］リンクをクリックしてください。

適切な製品サポート・サービスに登録すると，更新情報や最新情報も入手できます。詳細については HP の営業担当にお問い合わせください。

サポート

HP ソフトウェアのサポート Web サイトは，次の場所にあります。
support.openview.hp.com

HP ソフトウェアのオンライン・サポートは，インタラクティブな技術サポート・ツールにアクセスするための効率的な手段を提供します。サポート・サイトを利用することで，次のようなことができるメリットがあります。

- 関心のある内容のナレッジ・ドキュメントの検索
- サポート・ケースおよび機能強化要求の提出および追跡
- 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 サポートの連絡先の表示
- 利用可能なサービスに関する情報の確認
- ほかのソフトウェア顧客との議論の開始
- ソフトウェアのトレーニングに関する調査と登録

ほとんどのサポート・エリアは，HP Passport ユーザとしての登録およびサインインが必要です。また多くは，サポート契約も必要です。アクセス・レベルの詳細情報については，
http://h20230.www2.hp.com/new_access_levels.jsp を参照してください。

HP Passport ID の登録は，次の場所で行います。
http://h20229.www2.hp.com/passport-registration.html

目次

本書について 15
本書の構成 16
対象読者 17
QuickTest Professional オンライン・ドキュメント 18
その他のオンライン・リソース 21

第 I 部：BUSINESS PROCESS TESTING の概要

第 1 章： はじめに 25
QuickTest Professional for Business Process Testing の使用について 27
Business Process Testing について 28
必要なアクセス権の設定 38
サンプル・サイトの使用方法 39
ライセンス情報の変更 40
QuickTest ソフトウェアの更新 40

第 2 章： QuickTest の概要 43
QuickTest の開始 44
Quality Center プロジェクトへの接続 46
QuickTest ウィンドウ 47
キーワード・ビュー 51
アプリケーション領域 52
関数ライブラリ 54
スタート・ページ 55
［使用可能なキーワード］表示枠 57
［デバッグ ビューア］表示枠 59
［情報］表示枠 60
［欠落リソース］表示枠 61
［プロセス ガイダンス］表示枠 62
［リソース］表示枠 63
［タスク］表示枠 64
QuickTest コマンドの使用方法 65
QuickTest Professional プログラム・フォルダの参照 92

製品情報の表示 96

第 II 部 : テスト・オブジェクトおよびオブジェクト・リポジトリを使用した作業

第 3 章 : テスト・オブジェクト・モデルについて 103
テスト・オブジェクト・モデルについて 103
テスト・オブジェクト・モデルの概念の適用 107
オブジェクト・リポジトリの種類について 112
オブジェクト・スパイを使用したオブジェクト・プロパティと操作の表示 119
[オブジェクト スパイ] ダイアログ・ボックス 122

第 4 章 : コンポーネントのオブジェクト・リポジトリの使用方法 127
[オブジェクト リポジトリ] ウィンドウについて 128
[オブジェクトのプロパティ] ダイアログ・ボックス 143
リポジトリ・パラメータ値の割り当て 144
実行セッション中のテスト・オブジェクトでの作業 149

第 5 章 : オブジェクト・リポジトリ内のテスト・オブジェクトの管理 151
オブジェクト・リポジトリへのテスト・オブジェクトの追加 152
オブジェクト・リポジトリ内のオブジェクトのコピー，貼り付け，および移動 164
オブジェクト・リポジトリからのオブジェクトの削除 167
オブジェクトの検索 168
認識プロパティの保守 175

第 6 章 : オブジェクト認識の設定 193
オブジェクトの認識の設定について 193
[オブジェクトの認識] ダイアログ・ボックスについて 195
スマート認識の設定 209
ユーザ定義のテスト・オブジェクト・クラスの割り当て 219

第 7 章 : オブジェクト・リポジトリの管理 223
オブジェクト・リポジトリの管理について 224
オブジェクト・リポジトリ・マネージャ 226
オブジェクト・リポジトリを使った作業 233
共有オブジェクト・リポジトリでのオブジェクトの管理 239
リポジトリ・パラメータを使用した作業 245
オブジェクトの詳細の変更 250
テスト・オブジェクトの検索 254
結合操作の実行 255
インポート操作およびエクスポート操作の実行 256
オートメーションの使用によるオブジェクト・リポジトリの管理 259

第 8 章： 共有オブジェクト・リポジトリの結合.....261
共有オブジェクト・リポジトリの結合について 262
オブジェクト・リポジトリ結合ツールについて 264
オブジェクト・リポジトリ結合ツールのコマンドの使用方法 270
標準設定の定義 277
2 つのオブジェクト・リポジトリの結合 281
ローカル・オブジェクト・リポジトリから
共有オブジェクト・リポジトリを更新.....283
結合の統計情報の表示.....290
オブジェクトの矛盾について.....291
オブジェクトの矛盾の解決 294
[ターゲット リポジトリ] 表示枠に対するフィルタの設定.....296
特定のオブジェクトの検索 297
ターゲット・オブジェクト・リポジトリの保存 299

第 9 章： 共有オブジェクト・リポジトリの比較.....301
共有オブジェクト・リポジトリの比較について 302
オブジェクト・リポジトリ比較ツールについて 304
オブジェクト・リポジトリ比較ツールのコマンドの使用.....308
オブジェクトの相違点について 311
色の設定の変更 312
オブジェクト・リポジトリの比較.....314
比較統計情報の表示 316
[リポジトリ] 表示枠に対するフィルタ処理 318
オブジェクト・リポジトリ・ビューの同期 319
特定のオブジェクトの検索 320

第 III 部： 関数とその他のプログラミング・タスクの定義

第 10 章： 関数ライブラリ・ウィンドウを使用した作業.....325
関数ライブラリ・ウィンドウを使用した作業について 326
関数ライブラリでのステートメントの生成 326
関数ライブラリ内での操作 334
VBScript の基本的な構文の理解.....343
プログラム的記述の使用 350
プログラムによるアプリケーションの実行と終了 362
コメント，フロー制御，そのほかの VBScript ステートメントの
使用.....363
認識プロパティ値の取得と設定 370
ネイティブ・プロパティおよび操作へのアクセス 372
DOS コマンドの実行.....374
Windows API を使用したテストおよび関数ライブラリの拡張.....374
実行セッション中に報告するステップの選択.....377

第 11 章： 関数ライブラリ・ウィンドウのカスタマイズ.....379
関数ライブラリ・ウィンドウのカスタマイズについて 380
エディタの動作のカスタマイズ 381
エレメントの見映えのカスタマイズ 384
編集コマンドのカスタマイズ.....385

第 12 章： ユーザ定義関数および関数ライブラリを使用した作業 389
ユーザ定義関数および関数ライブラリの使い方について.....390
関数ライブラリの管理.....391
関連付けられている関数ライブラリを使用した作業.....402
関数定義ジェネレータの使用方法.....405
ユーザ定義関数のテスト・オブジェクト・メソッドとしての登録 419
ユーザ定義関数の使い方のヒント.....424

第 IV 部：アプリケーション領域を使った作業

第 13 章： アプリケーション領域の管理.....429
アプリケーション領域の使用について.....430
アプリケーション領域の作成.....433
アプリケーション領域を開く 436
アプリケーション領域の保存.....439
アプリケーション領域の削除.....443

第 14 章： アプリケーション領域の設定.....445
一般設定の定義 446
追加設定の定義 452
関数ライブラリの関連付け.....460
共有オブジェクト・ライブラリの関連付け 465
使用可能なキーワードの指定.....472

第 V 部：コンポーネントを使った作業

第 15 章： ビジネス・コンポーネントの使用.....481
ビジネス・コンポーネントの使用について 482
新規ビジネス・コンポーネントの作成.....483
ビジネス・コンポーネントを開く 488
ビジネス・コンポーネントの保存.....494
手動コンポーネントを使った作業.....500
コンポーネントに関連付けらているアプリケーション領域の変更 503
コンポーネントの印刷.....505

第 16 章： スクリプト・コンポーネントの作成 507
スクリプト・コンポーネントの使用について.....508
スクリプト・コンポーネントの作成 510
スクリプト・コンポーネントへの変換.....513

スクリプト・コンポーネントへのビジネス・コンポーネントの変換...514

第 17 章： キーワード・ビューを使った作業.....515
キーワード・ビューを使った作業について 516
キーワード・ビュー 518
コンポーネントへのステップの追加 523
コンポーネントへのその他のタイプのステップの追加 541
ステップの部分の変更.....541
パラメータの使用.....542
コメントの使用 551
コンポーネントステップの管理 553
キーワード・ビューでのキーボード・コマンドの使用 554
キーワード・ビューの表示オプションの定義.....555
キーワード・ビューのブレークポイントを使った作業 561

第 18 章： チェックポイントについて.....563
チェックポイントについて 563
コンポーネントへの新しいチェックポイントの追加.....564
チェックポイントの種類について.....565

第 19 章： 標準チェックポイントを使って オブジェクト・プロパティ値の検査 567
オブジェクトのプロパティ値の検査について.....567
標準チェックポイントの作成.....568
[チェックポイントのプロパティ] ダイアログ・ボックスについて 570
チェックポイントの変更 577

第 20 章： ビットマップの検査 579
ビットマップの検査について.....579
ビットマップ比較の微調整 580
ビットマップ・チェックポイントの作成と変更 582
ビットマップ・チェックポイントのプロパティ・ ダイアログ・ボックス 585

第 21 章： 値の出力 593
値の出力について.....593
出力値の作成.....594
プロパティ値の出力 595
出力タイプおよび設定の指定.....602

第 22 章： Windows ベースのオブジェクトのテキスト認識を 使用した作業 603
Windows ベースのオブジェクトのテキスト認識を使用した 作業について 604

［オプション］ダイアログ・ボックスの［一般］＞［テキスト認識］
表示枠 604
テキスト認識のガイドライン 608
テキスト認識および開発環境 609
使用事例シナリオ：画像内のテキストの検査 612

第 23 章： 正規表現の使い方 617
正規表現について 617
正規表現の概要と使い方 617
正規表現の定義 619

第 VI 部：設定

第 24 章： グローバル・テスト・オプションの設定 629
グローバル・テスト・オプションの設定について 629
［オプション］ダイアログ・ボックスの使用 630
テストの一般オプションの設定 632
テストのフォルダ・オプションの設定 635
テストの実行オプションの設定 638

第 25 章： ビジネス・コンポーネント設定にかかわる作業 647
ビジネス・コンポーネント設定にかかわる作業について 648
［ビジネス コンポーネントの設定］ダイアログ・ボックスの使用 648
コンポーネント・プロパティにかかわる作業 651
コンポーネントのスナップショットの定義 655
アプリケーション設定の表示 657
コンポーネント・リソースの表示 659
コンポーネントのパラメータの定義 660
回復シナリオの設定の表示 665

第 VII 部：コンポーネントの実行と分析

第 26 章： コンポーネントの実行 669
コンポーネントの実行について 669
コンポーネント全体の実行 670
コンポーネントの一部の実行 671
［実行］ダイアログ・ボックス：［結果の場所］タブ 674
［実行］ダイアログ・ボックス：［入力パラメータ］タブ 676

第 27 章： 実行セッション結果の表示 679
実行セッション結果の表示について 680
［テスト結果］ウィンドウ 681
実行セッションの結果の表示 689
実行結果の削除 710

Quality Center プロジェクトへの手動による実行セッション中に検出された不具合の送信 718
テスト結果の表示のカスタマイズ 719

第 28 章： 実行セッション結果の分析 723
テスト結果に含まれるスマート認識情報の分析 723
チェックポイントの結果の表示 727
［テスト結果］ウィンドウでのパラメータ化された値と出力値の結果の表示 733

第 VIII 部：コンポーネントのデバッグと保守

第 29 章： コンポーネントと関数ライブラリのデバッグ 739
コンポーネントと関数ライブラリのデバッグについて 740
デバッグ・セッションの速度調節 742
シングル・ステップ・コマンドの使用 742
［ステップまで実行］コマンドおよび［ステップからデバッグ］コマンドの使用 746
実行セッションの一時停止 748
ブレークポイントの使用 748
［デバッグ ビューア］表示枠 751
実行エラーの処理 762
関数のデバッグの練習 764

第 30 章： コンポーネントの保守 767
コンポーネントが失敗する理由 768
メンテナンス実行ウィザードを使ったコンポーネントの実行 770
更新モード・オプションを使ったコンポーネントの更新 787

第 IX 部：QUICKTEST IDE の使用

第 31 章： QuickTest ウィンドウのレイアウト 797
QuickTest ウィンドウのレイアウト変更 797
ツールバーとメニューのカスタマイズ 811
複数のドキュメントを使った作業 824

第 32 章： リソースの管理 827
［リソース］表示枠 827

第 33 章： コンポーネントへのキーワードの追加 831
［使用可能なキーワード］表示枠について 831

第 34 章： QuickTest のタスクとコメントの管理 835
タスクおよび TODO コメントを使った作業 835
［タスク］表示枠 836
［タスク エディタ］ダイアログ・ボックス 843

第 35 章： 欠落リソースの処理 845
欠落リソースの処理について 845
欠落している関数ライブラリの処理 848
欠落している共有オブジェクト・リポジトリの処理 850
欠落している回復シナリオの処理 851
未割り当ての共有オブジェクト・リポジトリ・パラメータ値の処理 854

第 36 章： プロセス・ガイダンスを使った作業 857
［プロセス ガイダンス］表示枠 858
プロセス・ガイダンスの開始 860
利用できるプロセスのリストの管理 862
［プロセス ガイダンス管理］ダイアログ・ボックス 862

第 X 部：高度な機能を使用した作業

第 37 章： 回復シナリオの定義と使用 867
回復シナリオの定義と使用について 867
回復シナリオを使用するタイミングの決定 869
回復シナリオの定義 870
回復シナリオ・ウィザードについて 874
回復シナリオの管理 901
アプリケーション領域への回復シナリオの関連付け 905
プログラムによる回復メカニズムの制御 909

第 38 章： QuickTest 操作のオートメーション 911
QuickTest 操作の自動化について 912
QuickTest オートメーション・スクリプトを使用する条件 913
オートメーション・スクリプトの設計と実行に使用する
プログラミング言語と開発環境の選択 914
QuickTest オートメーション・スクリプトの基本要素の学習 916
オートメーション・スクリプトの生成 917
QuickTest オートメーション・リファレンスの使用方法 918

第 XI 部：QUALITY CENTER を使用した作業

第 39 章： Quality Center との統合 921
Quality Center を使用した作業について 922
Quality Center との接続と切断 923
QuickTest の Quality Center との統合 929

第 40 章： リソースおよび依存関係モデルの使用 937
リソースおよび依存関係モデルの用語 938
リソースおよび依存関係モデルについて 939
アセットの依存関係を使用する利点 941
Quality Center でのリソースおよび依存関係モデルの使用 942

第 41 章： QuickTest アセットのバージョンの表示と比較.....949
アセット比較ツールおよびアセット・ビューアを使った作業 950
QuickTest アセット比較ツール 952
QuickTest アセット・ビューア 962

第 42 章： バージョン・コントロールを使ったアセット管理.....967
Quality Center でのアセットのバージョン管理 968
アセットのバージョンの履歴の表示 976
ベースラインの履歴の表示 978
バージョンの履歴とベースラインの履歴 982

第 XII 部：付録

付録 A： FAQ（よくある質問）.....985
コンポーネントの作成.....986
関数ライブラリを使用した作業 987
動的なコンテンツを使った作業 988
Web に関する高度な問題.....990
標準 Windows 環境.....993
コンポーネントの保守.....993
QuickTest のパフォーマンスの向上.....994

付録 B： カスタムのプロセス・ガイダンス・パッケージの作成.....997
プロセス・ガイダンス・パッケージについて.....997
パッケージ設定ファイルについて.....998
データ・ファイルの作成 1000
QuickTest でのカスタムのプロセス・ガイダンス・パッケージのインストール 1001

付録 C： ビットマップ・チェックポイントのカスタマイズ.....1003
ビットマップ・チェックポイントのカスタマイズについて 1004
カスタム・ビットマップ・コンペアラの開発.....1007
チュートリアル：カスタム・コンペアラの作成 1016
ビットマップ・チェックポイント・カスタマイズ・サンプルの使い方 1027

索引 1031

本書について

『**QuickTest Professional for Business Process Testing ユーザーズ・ガイド**』へようこそ。ここでは，QuickTest Professional を使用して HP Business Process Testing の作業を行う方法を説明します。QuickTest を使用してコンポーネントが基づいているアプリケーション領域を作成，管理する方法や，コンポーネントが使用するさまざまなリソース・ファイルを定義する方法も説明します。また，QuickTest Professional で Business Process Testing のキーワード駆動のビジネス・コンポーネントおよびスクリプト・コンポーネントを使用した作業の方法も説明します。

Business Process Testing は，QuickTest と Quality Center に完全に統合されており，お持ちのライセンスに Business Process Testing サポートが含まれる場合に有効になります。

本章では，次の項目について説明します。

- 本書の構成（16 ページ）
- 対象読者（17 ページ）
- QuickTest Professional オンライン・ドキュメント（18 ページ）
- その他のオンライン・リソース（21 ページ）

本書の構成

本書は以下の部で構成されています。

第 I 部　Business Process Testing の概要

QuickTest の概要と，Business Process Testing で作業を行う際のテスト・プロセスの主要な手順について説明します。

第 II 部　テスト・オブジェクトおよびオブジェクト・リポジトリを使用した作業

テスト・オブジェクト・モデルを紹介し，QuickTest によってアプリケーションのオブジェクトが識別される方法について説明します。オブジェクトの使用方法，オブジェクトの識別の設定方法，およびスマート認識定義の作成方法について説明します。また，オブジェクト・リポジトリを管理，マージ，および比較する方法についても説明します。

第 III 部　関数とその他のプログラミング・タスクの定義

関数ライブラリを使ったコンポーネントの拡張方法，関数ライブラリ・ウィンドウのカスタマイズ方法，および QuickTest でユーザ定義関数と関数ライブラリを使用する方法を説明します。

第 IV 部　アプリケーション領域を使った作業

アプリケーション領域を作成し管理する方法を説明します。アプリケーション領域には，コンポーネントが使用するすべてのリソースと設定が含まれます。また，ビジネス・コンポーネント，スクリプト・コンポーネント，ビジネス・コンポーネント・キーワード・ビューの作成方法と操作方法，およびチェックポイントと出力値の使用方法も説明します。

第 V 部　コンポーネントを使った作業

ビジネス・コンポーネント・キーワード・ビューおよびその他の QuickTest ツールやオプションを使用して，QuickTest でコンポーネントを作成，表示，変更およびデバッグする方法について説明します。

第 VI 部　設定

ビジネス・プロセス・テストのニーズに合うように QuickTest の設定を変更する方法を説明します。

第 VII 部　コンポーネントの実行と分析

コンポーネントとそれらに関連付けられた関数ライブラリを実行する方法，実行結果を表示，分析する方法を説明します。

第 VIII 部 コンポーネントのデバッグと保守

実行セッションを制御して，コンポーネントや関数ライブラリ内のバグを特定，検出する方法を説明します。

第 IX 部　QuickTest IDE の使用

QuickTest のレイアウトを変更する方法，テスト・リソースを管理する方法，およびプロセス・ガイダンスを使用する方法を説明します。

第 X 部　高度な機能を使用した作業

回復シナリオの使用方法，および QuickTest の操作を自動化する方法を説明します。

第 XI 部　Quality Center を使用した作業

HP Quality Center と統合することによって，テスト・プロセスの作成および管理する方法について説明します。

第 XII 部　付録

QuickTest に関するよくある質問の情報を提供し，カスタマイズされたプロセス・ガイダンス・パッケージの作成方法を説明します。

対象読者

本ガイドは，QuickTest Professional を Business Process Testing と組み合わせて使用するオートメーション・エンジニアを対象としています。オートメーション・エンジニアは，QuickTest Professional を使った自動テストの専門家であり，キーワード駆動型のテスト方法とテスト・プロセスに精通し，VBScript のプログラミングに熟練している必要があります。

オートメーション・エンジニアは，各分野のエキスパートと協力してビジネス・プロセス・テストを作成します。各分野のエキスパートは，オートメーション・エンジニアが作成したリソースと Quality Center のビジネス・コンポーネント・モジュールを使ってビジネス・プロセス・テストを作成します。Quality Center のビジネス・コンポーネント・モジュールについては，『**HP Business Process Testing ユーザーズ・ガイド**』を参照してください。

QuickTest Professional オンライン・ドキュメント

QuickTest Professional には，次のオンライン・ドキュメントがあります。

「**最初にお読みください**」では，QuickTest に関する最新ニュースと最新情報を入手できます。［**スタート**］＞［**プログラム**］＞［**QuickTest Professional**］＞［**Readme**］を選択してください。

『**QuickTest Professional インストール・ガイド**』では，QuickTest のインストールおよびセット・アップ方法を説明します。［**ヘルプ**］＞［**印刷用ドキュメント**］＞［**HP QuickTest Professional インストール ガイド**］を選択してください。

『**QuickTest Professional チュートリアル**』では QuickTest の基本スキルを学びアプリケーション向けのテストを設計する方法を説明します。［**ヘルプ**］＞［**HP QuickTest Professional チュートリアル**］を選択します。

「**製品の機能紹介ムービー**」（英語版）では，選択された QuickTest 機能の使用方法について，概要および手順ごとに説明します。［**ヘルプ**］＞［**製品の機能紹介ムービー**］を選択します。

「**印刷用ドキュメント**」では，Adobe の PDF（Portable Document Format）で全ドキュメントを表示します。オンライン・ドキュメントは Adobe Reader を使って読んだり印刷したりできます。Adobe Reader は，Adobe の Web サイト（http://www.adobe.com/jp/）からダウンロードできます。［**ヘルプ**］＞［**印刷用ドキュメント**］を選択します。

『**QuickTest Professional ヘルプ**』には次のドキュメントが含まれています。

- 『**QuickTest Professional 新機能**』では，最新バージョンの QuickTest でサポートされている環境，新機能および強化された点について説明します。
- 『**QuickTest Professional ユーザーズ・ガイド**』では，QuickTest を使用してアプリケーションをテストする方法を説明します。

- 『**QuickTest Professional for Business Process Testing ユーザーズ・ガイド**』では，QuickTest を使用して，ビジネス・プロセス・テスティングで使用するアセットの作成および管理を行う方法を手順ごとに説明します。
- 『**QuickTest Professional アドイン・ガイド**』では，サポート対象の環境で QuickTest のアドインを使って作業する方法について説明し，環境に固有の情報をアドインごとに示しています。
- 『**QuickTest Object Model Reference**』（英語版）では，QuickTest テスト・オブジェクトの説明，各オブジェクトに関連したメソッドおよびプロパティの一覧，メソッドおよびプロパティの構文情報と使用例を示します。
- 『**QuickTest Advanced References**』（英語版）には，次の QuickTest COM および XML リファレンスに関するドキュメントが含まれています。
  - 『**QuickTest Automation**』（英語版）では，オートメーション・オブジェクト，メソッドとプロパティの構文情報，詳細な情報，および例を入手できます。また，QuickTest のオートメーション・スクリプトを記述する際の詳しい概要も含まれます。オートメーション・オブジェクト・モデルは，QuickTest のほぼすべての機能を制御することを可能にするオブジェクト，メソッド，プロパティを提供することによって，テスト管理の自動化を支援します。
  - 『**QuickTest Test Results Schema**』（英語版）では，テスト結果のカスタマイズに必要な情報を提供する，テスト結果の XML スキーマについて説明します。
  - 『**QuickTest Test Object Schema**』（英語版）では，各種環境でテスト・オブジェクトのサポートを拡張するのに必要な情報を提供する，テスト・オブジェクトの XML スキーマについて説明しています。
  - 『**QuickTest Object Repository Schema**』（英語版）では，XML にエクスポートされたオブジェクト・リポジトリ・ファイルを編集するのに必要な情報を提供する，オブジェクト・リポジトリの XML スキーマについて説明しています。
  - 『**QuickTest Object Repository Automation**』（英語版）では，QuickTest の外部から QuickTest のオブジェクト・リポジトリやその内容を操作するのに必要な情報を提供する，オブジェクト・リポジトリ・オートメーション・オブジェクト・モデルについて説明します。
- 『**VBScript Reference**』（英語版）には，VBScript，Script Runtime，および Windows Script Host を含む Microsoft VBScript のドキュメントが含まれています。

QuickTest Professional ヘルプにアクセスするには，［**ヘルプ**］＞［**QuickTest Professional ヘルプ**］を選択します。選択した QuickTest ウィンドウおよびダイアログ・ボックスをクリックして F1 キーを押すことで QuickTest Professional ヘルプにアクセスすることもできます。また，QuickTest 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 キーを押すことで，それらの説明，構文，および例を参照できます。

その他のオンライン・リソース

Mercury Tours サンプル Web サイト（英語版）は，本書で説明する多くの例の基礎となります。この Web サイトの URL は http://newtours.demoaut.com です。［**スタート**］＞［**プログラム**］＞［**QuickTest Professional**］＞［**Sample Applications**］＞［**Mercury Tours Web サイト**］を選択します。

HP Software Web サイトからは，HP Software Web サイトにアクセスします。このサイトでは，HP Software 製品に関する最新情報を提供します。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 Software Web Site**］を選択します。この Web サイトの URL は http://welcome.hp.com/country/jp/ja/prodserv/software.html です。

QuickTest Professional の［**ヘルプ**］メニューから次のオンライン・リソースにアクセスできます。

トラブルシューティング & ナレッジベース：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します。このサイトからはセルフ・ソルブ技術情報を検索できます。［**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://h20230.www2.hp.com/troubleshooting.jsp です。

HP ソフトウェア・サポート：HP ソフトウェアのサポート Web サイトにアクセスします。このサイトでは，セルフ・ソルブ技術情報を閲覧できます。また，ユーザ・ディスカッション・フォーラムへの投稿や検索，サポート依頼の送信，パッチや更新されたドキュメントのダウンロードなども行えます。［**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，www.hp.com/go/hpsoftwaresupport です。

ほとんどのサポート・ページでは，HP Passport ユーザとして登録してログインすることを求められます。また，多く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，
http://h20230.www2.hp.com/new_access_levels.jsp を参照してください。

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

第 I 部

Business Process Testing の概要

第 1 章

はじめに

QuickTest Professional for Business Process Testing へようこそ。Business Process Testing を使用すると Quality Center で作業する専門知識のない，各分野のエキスパートが QuickTest Professional で作業を行っているオートメーション・エンジニアと効果的に共同作業を行うことができるようになります。各分野のエキスパートの側でプログラミングの知識がなくても，ビジネス・プロセス・テストの作成，ドキュメント化，および実行ができます。

注：QuickTest Professional は，Unicode 規格（http://www.unicode.org/standard/standard.html）の要件に従って Unicode に準拠しています。このため，各国の言語で開発されたアプリケーションをテストするための VBScript ステートメントの追加と更新が可能です。Unicode では，文字が 8 ビットまたは 16 ビットのコード値で表されます。QuickTest Professional がインストールされているコンピュータに適切な Windows 言語サポートがインストールされていれば，英語以外のアプリケーションをテストできます（［**スタート**］＞［**設定**］＞［**コントロール パネル**］＞［**地域と言語のオプション**］など）。ユニコードおよび多言語サポートに関する詳細については，『**HP QuickTest Professional 最初にお読みください**』を参照してください。

本書では，オートメーション・エンジニアが Business Process Testing に必要な自動リソースを作成および変更できるようにする QuickTest Professional の機能やオプションについて説明します。また，ビジネス・プロセス・テストの基本要素であるコンポーネントを作成するための機能やオプションについても説明します。

注：QuickTest を使用してビジネス・プロセス・テストで使用するスクリプト・コンポーネントを作成することもできますが，本書の対象は，主にビジネス・コンポーネントに関連する機能性や機能です。スクリプト・コンポーネントとビジネス・コンポーネントの違いの詳細については，第 16 章「スクリプト・コンポーネントの作成」の QuickTest を参照してください。

本章の内容

- QuickTest Professional for Business Process Testing の使用について（27 ページ）
- Business Process Testing について（28 ページ）
- 必要なアクセス権の設定（38 ページ）
- サンプル・サイトの使用方法（39 ページ）
- ライセンス情報の変更（40 ページ）
- QuickTest ソフトウェアの更新（40 ページ）

QuickTest Professional for Business Process Testing の使用について

Business Process Testing は，ロールベースのテスト・モデルです。アプリケーションの開発ライフサイクル中に**オートメーション・エンジニア**と**各分野のエキスパート**が協力してアプリケーションのビジネス・プロセスをテストできます。

オートメーション・エンジニアは，自動テストのエキスパートです。オートメーション・エンジニアは QuickTest を使用して，「**コンポーネント**」の作成に必要なリソースや設定を定義します。コンポーネントは「**ビジネス・プロセス・テスト**」の基本要素となるものです。

各分野のエキスパートは，テスト対象のアプリケーションのさまざまな部分や，テストを必要とするビジネス・プロセスを理解しています。ただし，必ずしも自動テストの作成に必要なプログラミング知識を持っているとはかぎりません。Quality Center のビジネス・コンポーネントとテスト計画モジュールを使用して，キーワード駆動のビジネス・プロセス・テストを作成します。

注：本書に記述されているいくつかの機能は，使用している Quality Center エディションではサポートされない可能性があります。詳細については，『**Mercury Quality Center ユーザーズ・ガイド**』を参照してください。

QuickTest と Quality Center の統合により，オートメーション・エンジニアは必要なリソースと設定を効果的に作成し保守できるようになり，同時に各分野のエキスパートはプログラミングの知識が必要のない，スクリプトが不要な環境でビジネス・プロセス・テストを作成し実装できるようになります。

注：組織では，それぞれのニーズに応じてオートメーション・エンジニアと各分野のエキスパートの役割を定めます。本書では，読者が上記に示したようなオートメーション・エンジニアの役割を担当しており，組織のほかの人員が各分野のエキスパートの役割を担当していることを想定しています。ただし，これらの役割は柔軟性が高く，Business Process Testing を使用する担当者の能力と時間リソースに依存します。特定の組織内で定義する必要がある役割や，（ユーザが正しい権限を持っているという条件で）あるユーザ・タイプが実行できる Business Process Testing タスクを制御する製品固有のルールや制限はありません。

Business Process Testing について

Business Process Testing ではテストの自動化と，自動生成されるわかりやすいテスト・ドキュメントを組み合わせることによって，構造化されたアプリケーション・テストを行うことができます。Business Process Testing は詳細なテスト・スクリプトの完成に依存しません。このため，自動テストの準備が整うまでの間，アプリケーションを手動でテストすることが可能です。また，ビジネス・プロセス・テストをほかの自動テストよりも短期間に作成し実装できるようになるため，開発プロセスの早い段階で，ダウンタイムが発生する前に，パフォーマンスの潜在的な問題を検出できます。

「**コンポーネント**」は，特定のタスクを実行する，保守しやすく再利用が可能なユニットです。コンポーネントはビジネス・プロセス・テストを構成する基本的な要素となります。各コンポーネントは，特定の順序で論理的にまとめて実行される，いくつかのアプリケーション・ステップで構成されています。たとえば，Web アプリケーションでは，ログイン・コンポーネントが 4 つのステップで構成されることが考えられます。最初のステップでは，アプリケーションを開きます。2 番目のステップは，ユーザ名の入力が考えられます。3 番目のステップはパスワードの入力，最後のステップは Web ページの［**送信**］ボタンのクリックが考えられます。関数を作成したり，関数ライブラリに格納されている関数を呼び出したりすることによって，コンポーネントを拡張し，ログイン・タスクの重要な詳細部分をテストするためのロジックを追加することができます。

仕様上，各コンポーネントはアプリケーションの特定の部分の機能をテストします。コンポーネントを組み合わせ，一連のフローとしてビジネス・プロセス・テストに組み込むことによって，特定のビジネス・プロセスの中で実行される主要なタスクを表現します。たとえば，フライト予約アプリケーション用のビジネス・プロセス・テストには，ログイン・コンポーネント，フライト情報検索コンポーネント，フライト予約コンポーネント，購入コンポーネント，およびログアウト・コンポーネントを含めることができます。フライト情報検索，フライト予約，購入の各コンポーネントは，複数の予約事例をテストする場合に，同じビジネス・プロセス・テストの中で複数回再利用されることが考えられます。また，各フライト予約の合間にアプリケーションをリセットするコンポーネントをテストに含めることによって，フライト予約を複数回繰り返して実行することも考えられます。コンポーネントやビジネス・プロセス・テストを作成し実行する作業は，通常は Quality Center で作業を行う各分野のエキスパートによって実行されます。

コンポーネントはモジュール化，再利用が可能なため，複数のビジネス・プロセス・テストで使用できます。たとえば，同じログインおよびログアウト・コンポーネントを，アプリケーションのレポートおよびグラフ生成プロセスをテストする分析（レポート）コンポーネントと組み合わせて使用したり，マイレージ・サービス制度の申し込みを行うビジネス・プロセスをテストするマイレージ・サービス制度コンポーネントと組み合わせて使用したりできます。

QuickTest は 2 つのタイプのコンポーネントを提供します。［**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**］です。ビジネス・コンポーネント（キーワード駆動コンポーネントとも呼ばれます）は QuickTest と Quality Center の両方に完全に統合されているので，オートメーション・エンジニアと各分野のエキスパートの双方がビジネス・コンポーネントを作成，変更，および実行できます。スクリプト・コンポーネントは Quality Center または QuickTest で作成できるプログラム・ロジックを含むより複雑なコンポーネントです。複雑であるため，スクリプト・コンポーネントは QuickTest でのみ変更ができます。各分野のエキスパートはスクリプト・コンポーネントをビジネス・プロセス・テストに組み込めますが，変更することはできません。

注：QuickTest を使用してビジネス・プロセス・テストで使用するスクリプト・コンポーネントを作成することもできますが，本書ではビジネス・コンポーネントに関連する機能と特徴について説明します。スクリプト・コンポーネントとビジネス・コンポーネントの違いやスクリプト・コンポーネントの使用の詳細については，第 16 章「スクリプト・コンポーネントの作成」を参照してください。

自動テスト・リソースが利用できるようになるまでの間，各分野のエキスパートは各コンポーネントの［ステップのデザイン］タブで手動ステップを定義できます（Quality Center のビジネス・コンポーネント・モジュールを使用します）。そして，それらの手動コンポーネントをビジネス・プロセス・テストに追加し，Quality Center マニュアル・ランナーを使用してステップを手動で実行できます。コンポーネントを定義する際，各分野のエキスパートは（Quality Center のビジネス・コンポーネント・モジュール内の）［詳細］タブの［ディスカッション エリア］にコメントを追加できます。ここでは，新しい操作の要求や，コンポーネントに予定されている今後の変更，コンポーネントを使用できるほかのテストなど，各分野のエキスパートがオートメーション・エンジニアに伝える必要のある補足情報や備考を入力できます。

この設計段階で，各分野のエキスパートと連携して作業し，どのようなリソースや設定が各コンポーネントに必要なのかを定義することができます。そして，テストの実際のニーズに基づいて，アプリケーションのさまざまな部分に対応する個々の「**アプリケーション領域**」を作成することができます。アプリケーション領域は，ビジネス・プロセス・テストでの作業時にコンポーネントが使用する設定とリソース・ファイルを指定するものです。各分野のエキスパートはコンポーネントを常に特定のアプリケーション領域に関連付け，それによってコンポーネントからそれらの設定やリソース・ファイルへのアクセスが可能になります。オートメーション・エンジニアがアプリケーション領域を作成し，その設定とリソース・ファイルを定義した後，各分野のエキスパートはそれらの自動テスト・リソースをビジネス・コンポーネント・ステップに組み込み，既存の手動コンポーネントを自動コンポーネントに変換して，新しい自動コンポーネントを作成することができます。

QuickTest のプロセス・ガイダンスを使用すると，ビジネス・コンポーネントを作成するプロセスを支援できます。詳細については，857 ページ「プロセス・ガイダンスを使った作業」を参照してください。

アプリケーション領域について

アプリケーション領域はコンポーネントの土台となるものです。アプリケーション領域によって，アプリケーションの特定部分のテストに関係するすべての要素の保守を一か所で行えるようになります。

アプリケーション領域では，アプリケーションの特定部分のテストに関係する具体的な設定を定義できます。たとえば，実行セッションの開始時に特定のアドインをロードする，指定したアプリケーションのみを対象としてコンポーネントを実行する，特定の条件のもとで回復シナリオを起動するなどの処理を QuickTest に指示する設定を定義できます。また，対象アプリケーション領域に関連付けられている任意のコンポーネントで使用できるキーワードを指定することもできます。

アプリケーション領域の重要な要素の 1 つに，コンポーネントが使用できるリソース・ファイルがあります。作成したリソース・ファイルは，特定のアプリケーションのビジネス・プロセス・テストを作成して実行する各分野のエキスパートが使用している，同じ Quality Center プロジェクトに格納します。通常，リソース・ファイルには関数ライブラリと共有オブジェクト・リポジトリが含まれます。

QuickTest がアプリケーション内のオブジェクトを認識するために使用する共有オブジェクト・リポジトリ・ファイルを作成，情報登録，および保守します。QuickTest オブジェクト・リポジトリ・マネージャを使用して，共有オブジェクト・リポジトリ内のテスト・オブジェクト情報を定義，変更します。共有オブジェクト・リポジトリ・ファイルをアプリケーション領域に関連付けた後，関連性に従って優先順位を付けることができます。共有オブジェクト・リポジトリをアプリケーション領域に関連付けることによって，そのアプリケーション領域をベースとするすべてのコンポーネントから，そのすべてのテスト・オブジェクトやほかの要素にアクセスできるようになります。詳細については，第 5 章「オブジェクト・リポジトリ内のテスト・オブジェクトの管理」および第 7 章「オブジェクト・リポジトリの管理」を参照してください。

コンポーネントによって呼び出される，関数や**操作**（キーワードとも呼ばれる）を含む関数ライブラリも作成します。これらの関数には，特定のタスクを実行するためのステップをカプセル化するプログラミング・ロジックが含まれており，呼び出し元のコンポーネントの機能はこれらの関数によって強化されます。これらの関数ライブラリは，QuickTest を使って作成できます。または，QuickTest 関数定義ジェネレータを使用して基本的な関数定義を挿入し，コードを追加して関数を完成させることもできます。

関数ライブラリ・ファイルをアプリケーション領域に関連付けた後，関連性に従って優先順位を付けることができます。関数ライブラリ・ファイルをアプリケーション領域に関連付けることによって，そのアプリケーション領域をベースとするすべてのコンポーネントから，その関数ライブラリの中で定義されているすべてのパブリック関数にアクセスできるようになります。関数ライブラリの使用方法の詳細については，第 10 章「関数ライブラリ・ウィンドウを使用した作業」を参照してください。

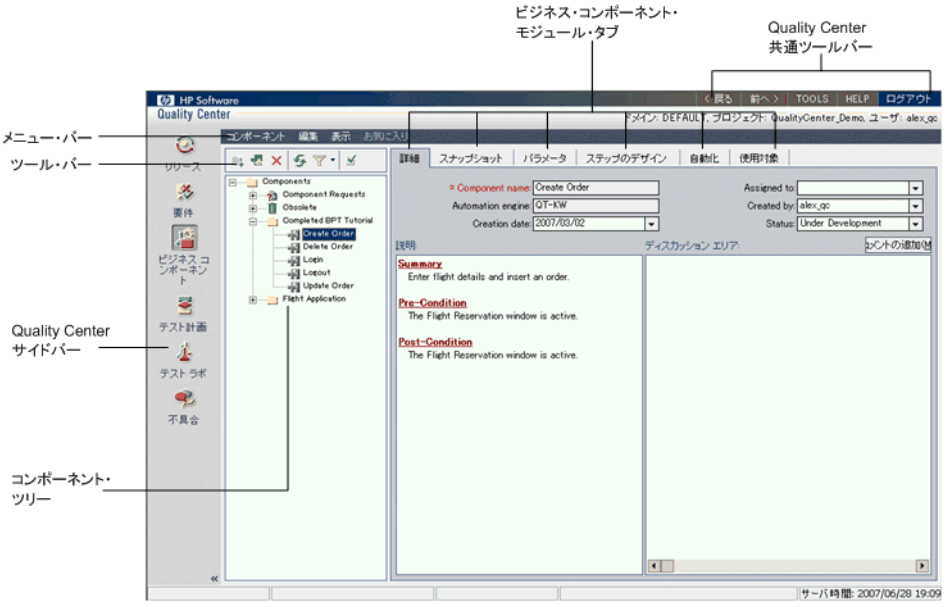
複数のアプリケーション領域を作成し，それぞれの領域がテスト対象アプリケーションの特定の部分（範囲）を対象とするようにできます。たとえば，フライト予約アプリケーションの場合，アプリケーション領域をログイン・モジュール用に 1 つ，フライト検索モジュール用に 1 つ，フライト予約モジュール用に 1 つ，請求書発行モジュール用に 1 つ，というように作成できます。アプリケーション領域の詳細については，第 13 章「アプリケーション領域の管理」を参照してください。

アプリケーション領域に関連付けられるリソース・ファイルの作成および保守に加えて，QuickTest では，コンポーネントとそれに関連付けられている関数ライブラリのデバッグも可能です。QuickTest でコンポーネントを作成することもできますが，この作業は多くの場合，各分野のエキスパートが Quality Center を使用して行います。詳細については，第 15 章「ビジネス・コンポーネントの使用」を参照してください。

QuickTest のプロセス・ガイダンスを使用すると，アプリケーション領域を作成するプロセスを支援できます。詳細については，857 ページ「プロセス・ガイダンスを使った作業」を参照してください。

Quality Center ビジネス・コンポーネント・モジュールでのコンポーネントの作成

各分野のエキスパートは，Quality Center ビジネス・コンポーネント・モジュールで新しいコンポーネントを作成し，定義します。



ビジネス・コンポーネント・モジュールには，次のタブがあります。

- ［**詳細**］タブ。コンポーネントの目的（ゴール）のサマリおよびコンポーネントの実行前後のアプリケーション状態（前提条件および後提条件）を指定します。現在選択しているビジネス・コンポーネントの詳細および実装要件を指定できます。
- ［**スナップショット**］タブ。コンポーネントの目的や操作について，視覚的な手掛かりや説明を提供するイメージです。アプリケーションからスナップショットのイメージをキャプチャし，現在選択しているビジネス・コンポーネントに添付することができます。
- ［**パラメータ**］タブ。ビジネス・コンポーネントの入力コンポーネント・パラメータ値および出力コンポーネント・パラメータ値を指定します。パラメータを実装および使用することで，コンポーネントは，外部ソースからデータを受け取り，ビジネス・プロセス・テスト・フローのほかのコンポーネントにデータを渡すことができるようになります。

- ［**ステップのデザイン**］タブ。ビジネス・コンポーネントの手動ステップを作成または表示できます。必要に応じて手動ステップを自動化できます。
- ［**自動化**］タブ。自動化されたコンポーネントを表示，またはアクセスできるようにします。キーワード駆動のコンポーネントの場合，キーワード駆動のテーブル形式で自動ビジネス・コンポーネントのステップを作成，変更でき，実装されているコンポーネントの各ステップについてわかりやすく書かれた説明が表示されます。
- ［**依存関係**］**タブ。**現在選択されているビジネス・コンポーネントにリンクされたアセットのリストを表示します。
- ［**履歴**］**タブ。**コンポーネントに加えられた変更のログを表示します。

Quality Center テスト計画モジュールでのビジネス・プロセス・テストおよびフローの作成

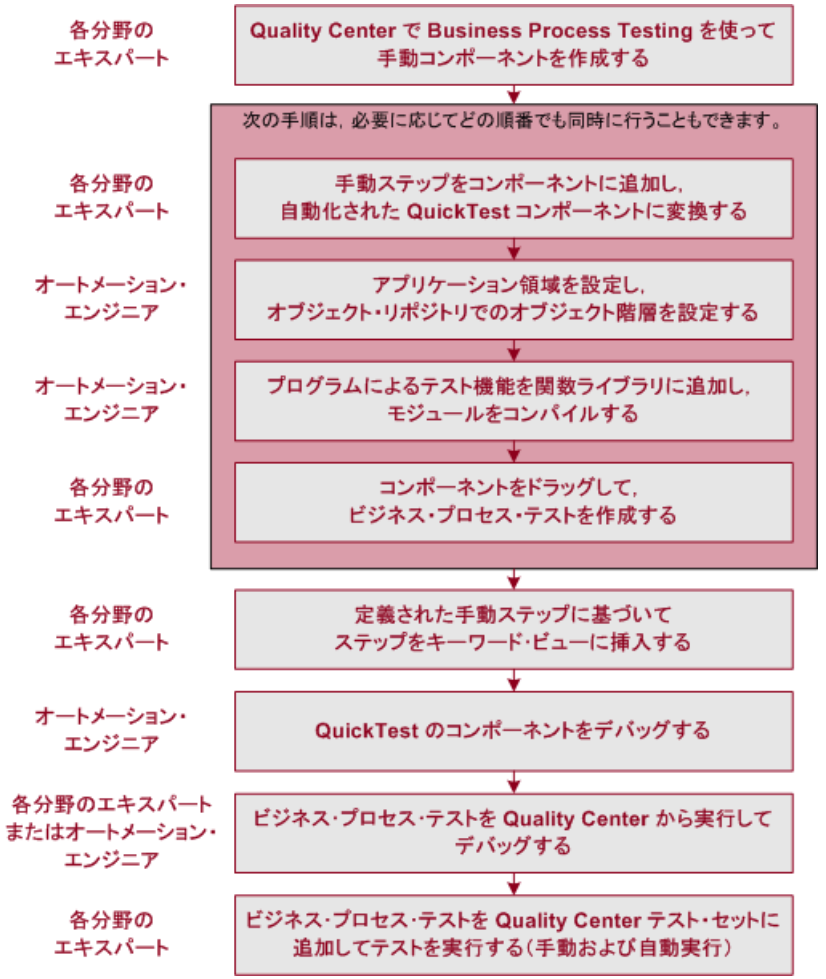
各分野のエキスパートは最初にテスト計画モジュールに，ビジネス・プロセス・テスト，またはフローを作成します。ビジネス・プロセス・テスト，またはフローを設定するために，各分野のエキスパートは次に関連するコンポーネントを選択（ドラッグ・アンド・ドロップ）し，それらの実行方法を設定します。

各コンポーネントは，異なるビジネス・プロセス・テストやフローで違う方法で使用できます。たとえば，コンポーネントに対してテストごとに異なる入力パラメータ値を使用したり，異なる反復回数で実行されるよう設定できます。

ビジネス・プロセス・テストまたはフローを作成しているときに，各分野のエキスパートが，ビジネス・プロセス・テストまたはフローに必要な要素に対してコンポーネントがまだ定義されていないことに気付いたとします。その場合，各分野のエキスパートはテスト計画モジュールからコンポーネント要求を送信できます。

Business Process Testing のワークフローについて

次は QuickTest を使用した一般的な Business Process Testing ワークフローの例です。組織の実際のワークフローは，別のプロジェクトまたは製品開発ライフサイクルの別の段階では異なる場合があります。



QuickTest Professional for Business Process Testing の用語について

本書では，次に示す QuickTest Professional for Business Process Testing 固有の用語を使用しています。

アプリケーション領域：ビジネス・コンポーネントの作成と実装に使用されるリソースと設定の集まり。これには，関数ライブラリ，共有オブジェクト・リポジトリ，キーワード，テスト設定，または回復シナリオなどのその他のテスト・リソースが含まれます。アプリケーション領域によって，アプリケーションの特定部分のテストに関係するすべての要素の保守を一か所で行えるようになります。アプリケーションの部分ごとに別々のアプリケーション領域を定義し，コンポーネントを該当するアプリケーション領域に関連付けることができます。

フロー：固定されたシーケンスのビジネス・コンポーネントの集合。フローは特定のタスクを実行するテストの種類の 1 つです。フローは複数のビジネス・プロセス・テストで使用できます。フローやそのコンポーネントを変更すると，そのフローを含むすべてのビジネス・プロセス・テストに変更が反映されます。

各分野のエキスパート は Quality Center テスト計画モジュールでフローを作成します。また［**実行条件**］を使用してフローの柔軟性を活かし，コンポーネントにフロー内の前のステージに基づいて選択的に実行させることができます。

ビジネス・コンポーネント（または**コンポーネント**）：特定のタスクを実行する 1 つ以上のステップで構成されている，保守が容易で再利用可能な単位です。ビジネス・コンポーネントは，外部ソースまたはほかのコンポーネントからの入力値が必要なことがあります。また，ほかのコンポーネントに出力値を返すことができます。

キーワード駆動コンポーネントとも呼ばれます。

手動コンポーネント：Quality Center で作成された自動化されていないビジネス・コンポーネント。QuickTest では，手動コンポーネントは自動ビジネス・コンポーネントに変換した後にのみ，表示および操作できます。

スクリプト・コンポーネント：プログラム・ロジックを含めることができ，QuickTest のキーワード・ビュー，エキスパート・ビュー，および QuickTest のその他のツールやオプションを使って編集できる自動コンポーネント。

キーワード・ビュー：キーワード駆動のモジュール化された表形式を使用して，テストとコンポーネントの作成，表示，およびデバッグを実行できる，スプレッドシート風のビュー。

関数ライブラリ：VBScript の関数，サブルーチン，モジュールなどが格納されているドキュメント。これらの関数はコンポーネントの中で操作（キーワード）として使用できます。QuickTest の関数ライブラリ・エディタを使用して，関数ライブラリ・ドキュメントの作成とデバッグができます。

ビジネス・プロセス・テスト：アプリケーションの特定のビジネス・プロセスをテストするように設計された，ビジネス・コンポーネントの一連のフローを構成するシナリオ。

コンポーネント入力パラメータ：ビジネス・コンポーネントが受け取り，コンポーネントの特定のパラメータ化ステップの値として使用する変数値。

コンポーネント出力パラメータ：ビジネス・コンポーネントが返す値。この値は，ビジネス・プロセス・テスト結果で表示できます。また，テスト内の後続のコンポーネントの入力値としても使用できます。

ローカル入力パラメータ：コンポーネント内で定義された変数値。これらの値は，同じコンポーネント内の後続のパラメータ化されたステップが受け取って使用できます。

ローカル出力パラメータ：同じコンポーネント内で使用するために，操作またはコンポーネント・ステップが返すことのできる値。これらの値は，ビジネス・プロセス・テストの結果内に表示され，コンポーネントの後のステップへの入力値としても使用できます。

役割：Business Process Testing に関係するさまざまなタイプのユーザ。

オートメーション・エンジニア：QuickTest Professional 自動テストのエキスパート。オートメーション・エンジニアは，ビジネス・コンポーネントの作成と使用に必要なリソースを定義し，管理します。オートメーション・エンジニアはアプリケーション領域を作成します。アプリケーション領域は，各分野のエキスパートが Quality Center においてビジネス・コンポーネントおよびビジネス・プロセス・テストを作成するのに必要なすべてのリソースと設定を指定します。オートメーション・エンジニアは，関数ライブラリを作成および変更でき，テスト対象アプリケーション内のさまざまなオブジェクトを表すテスト・オブジェクトを共有オブジェクト・リポジトリに登録できます。オートメーション・エンジニアは QuickTest の中でビジネス・コンポーネントの作成とデバッグも行えます。

各分野のエキスパート：各分野のエキスパートは，アプリケーション・ロジックに関する知識，システム全体の深い理解，テスト対象アプリケーションの基本となる個々の要素とタスクに関する深い理解を有しています。各分野のエキスパートは，Quality Center を使用してコンポーネントとビジネス・プロセス・テストを作成します。

必要なアクセス権の設定

QuickTest Professional の実行，または Quality Center で作業するには，下記のアクセス権の設定が必要です。

QuickTest Professional の実行に必要な権限

ファイル・システムに対する次のアクセス許可が必要です。

- QuickTest のインストール先の配下にあるすべてのファイルとフォルダに対する読み取りおよび書き込みの権限
- Temp フォルダに対する読み取りおよび書き込みの権限
- Windows フォルダおよび System フォルダの読み取り権限

レジストリ・キーに対する次の権限が必要です。

- **HKEY_CURRENT_USER¥Software¥Mercury Interactive** 以下のすべてのキーの読み取りおよび書き込みの権限
- **HKEY_LOCAL_MACHINE** と **HKEY_CLASSES_ROOT** のすべてのキーに対する読み取りおよび値照会の権限。

Quality Center の使用時に必要な権限

Quality Center に関する次の権限を持っている必要があります。

- Quality Center キャッシュ・フォルダへの完全な読み取りおよび書き込み権限。
- QuickTest の Quality Center アドインのインストール・フォルダに対する完全な読み取りおよび書き込み権限

Business Process Testing の使用に必要な権限

Quality Center のプロジェクト管理者は，プロジェクトにログインできるユーザを定義したり，各ユーザが実行できるタスクの種類を指定したりすることにより，プロジェクトへのアクセスを制御できます。Quality Center プロジェクト管理者は，Quality Center プロジェクトのビジネス・コンポーネント・モジュール内のフォルダ，コンポーネント，ステップ，およびパラメータに対して追加，変更，削除を実施する許可を割り当てることができます。

ビジネス・コンポーネントおよびアプリケーション領域を使用する前に，必要な Quality Center アクセス権限を持っていることを確認する必要があります。

- Quality Center のコンポーネント・ステップで作業するには，適切な［**ステップの追加**］，［**ステップの変更**］，または［**ステップの削除**］権限がなければなりません。コンポーネント・ステップで作業するのに［**コンポーネントの変更**］権限は必要ありません。［**コンポーネントの変更**］権限によりコンポーネント・プロパティ（コンポーネントの［詳細］タブのフィールド）の作業ができます。
- Quality Center またはテスト・ツールのパラメータを使用するには，Quality Center にすべてのパラメータ・タスク権限が設定されている必要があります。
- アプリケーション領域を変更するには，コンポーネントの変更，ステップの追加，変更，削除を実施するのに必要な権限が必要です。4 つの権限すべてが必要です。これらの権限のいずれかが割り当てられていない場合は，アプリケーション領域を読み取り専用形式でしか開くことができません。

ビジネス・コンポーネント・モジュールでのユーザ・グループ権限の設定に関する詳細は，Quality Center オンライン・ドキュメントで使用できる『**HP Business Process Testing ユーザーズ・ガイド**』を参照してください。

サンプル・サイトの使用方法

本書に出てくる例の多くは，サンプルの Web サイト，Mercury Tours に基づいています。この Web サイトの URL は，http://newtours.demoaut.com です。

このサイトを使用するには，ユーザ名とパスワードを登録する必要があります。

サンプルの Windows ベースのフライト・アプリケーションが，QuickTest Professional のインストールでも提供されています。このアプリケーションには，［**スタート**］＞［**プログラム**］＞［**QuickTest Professional**］＞［**Sample Applications**］＞［**Flight**］からアクセスできます。

ライセンス情報の変更

QuickTest を使用するにはライセンスが必要です。QuickTest をインストールする際に，次のライセンスの種類のいずれかを選択します。

- インストールしたコンピュータにのみ有効な無期限の**シート**・ライセンス。
- 複数の QuickTest ユーザが使用できるネットワーク・ベースの**コンカレント**・ライセンス。

ライセンスの種類は，コンピュータに管理者権限を持つユーザとしてログインしていれば，いつでも変更できます。たとえば，現在シート・ライセンスを使用している場合，コンカレント・ライセンス・サーバをネットワーク上で利用できる場合は，コンカレント・ライセンス・サーバに接続することを選択できます。

ライセンス情報の変更に関する詳細については，『**HP QuickTest Professional インストール・ガイド**』を参照してください。

QuickTest ソフトウェアの更新

標準設定では，QuickTest は 7 日ごとに自動的にオンライン・ソフトウェア・アップデートを確認します。QuickTest を開始すると，最後の自動の「更新の確認」が行われたのが 7 日より前かをチェックし，もしそうであればアップデートを確認します。

更新は，いつでも手動で確認することもできます。確認するには，QuickTest 内で［**ヘルプ**］＞［**更新の確認**］を選択するか，または［**スタート**］＞［**プログラム**］＞［**QuickTest Professional**］＞［**Check for Updates**］を選択します。

利用可能な更新がある場合は，ダウンロード（および必要に応じてインストール）する更新を選択できます。詳細については，画面に表示されている指示に従います。

ヒント：更新の自動チェックを無効にするには，［オプション］ダイアログ・ボックスの［一般］タブにある［**自動的にソフトウェアの更新を確認する**］チェック・ボックスをクリアします。［オプション］ダイアログ・ボックスを開くには，［**ツール**］＞［**オプション**］を選択します。

第 2 章

QuickTest の概要

本章では，QuickTest の開始方法を説明し，QuickTest ウィンドウを紹介します。

本章の内容

- QuickTest の開始（44 ページ）
- QuickTest ウィンドウ（47 ページ）
- キーワード・ビュー（51 ページ）
- アプリケーション領域（52 ページ）
- 関数ライブラリ（54 ページ）
- スタート・ページ（55 ページ）
- ［使用可能なキーワード］表示枠（57 ページ）
- ［デバッグ ビューア］表示枠（59 ページ）
- ［情報］表示枠（60 ページ）
- ［欠落リソース］表示枠（61 ページ）
- ［プロセス ガイダンス］表示枠（62 ページ）
- ［リソース］表示枠（63 ページ）
- ［タスク］表示枠（64 ページ）
- QuickTest コマンドの使用方法（65 ページ）
- QuickTest Professional プログラム・フォルダの参照（92 ページ）
- 製品情報の表示（96 ページ）

QuickTest の開始

QuickTest を起動するには，［**スタート**］メニューから［**プログラム**］＞［**QuickTest Professional**］＞［**QuickTest Professional**］を選択するか，デスクトップにある［**QuickTest Professional**］のショートカットをダブルクリックします。

初めて QuickTest を起動すると，［アドイン マネージャ］ダイアログ・ボックスが開いて現在インストールされているアドインが表示されます。ロードするアドインを選択します。

QuickTest はロードしたアドインを記憶していて，次回 QuickTest を起動するときは，前のセッションで選択したアドインが標準で選択されます。最高のパフォーマンスを得るには，特定のセッションで必要ないアドインは選択しないことをお勧めします。

ヒント：次回 QuickTest を開始したときにこのダイアログ・ボックスが開かないようにするには，［**起動時に表示する**］チェック・ボックスをクリアします。

アドインのインストール，ロード，および使用方法の詳細については，『**HP QuickTest Professional インストール・ガイド**』および『**HP QuickTest Professional アドイン・ガイド**』を参照してください。

［**OK**］をクリックします。QuickTest Professional ウィンドウが開いて［スタート ページ］と空のテストが表示されます。空のテストにアクセスするには，［**テスト**］タブをクリックします。

［スタート ページ］では，次の作業が可能です。

- QuickTest の使用に関するベスト・プラクティスについては，QuickTest プロセス・ガイダンスのリンクをクリックします。所属する組織に独自のプロセス・ガイダンスがある場合は，［**プロセス ガイダンス リスト**］で，そのリンクをクリックできる場合があります。
- ショートカット・ボタンをクリックして，新規または既存のテストまたは関数ライブラリを開きます。Business Process Testing が有効な場合は，新規または既存のビジネス・コンポーネントまたはアプリケーション領域を開くこともできます。
- **新機能情報**のセクションでリンクをクリックすると，このバージョンの QuickTest で提供される新機能に関する情報を入手できます。

［スタート ページ］の詳細については，55 ページ「スタート・ページ」を参照してください。

Quality Center プロジェクトへの接続

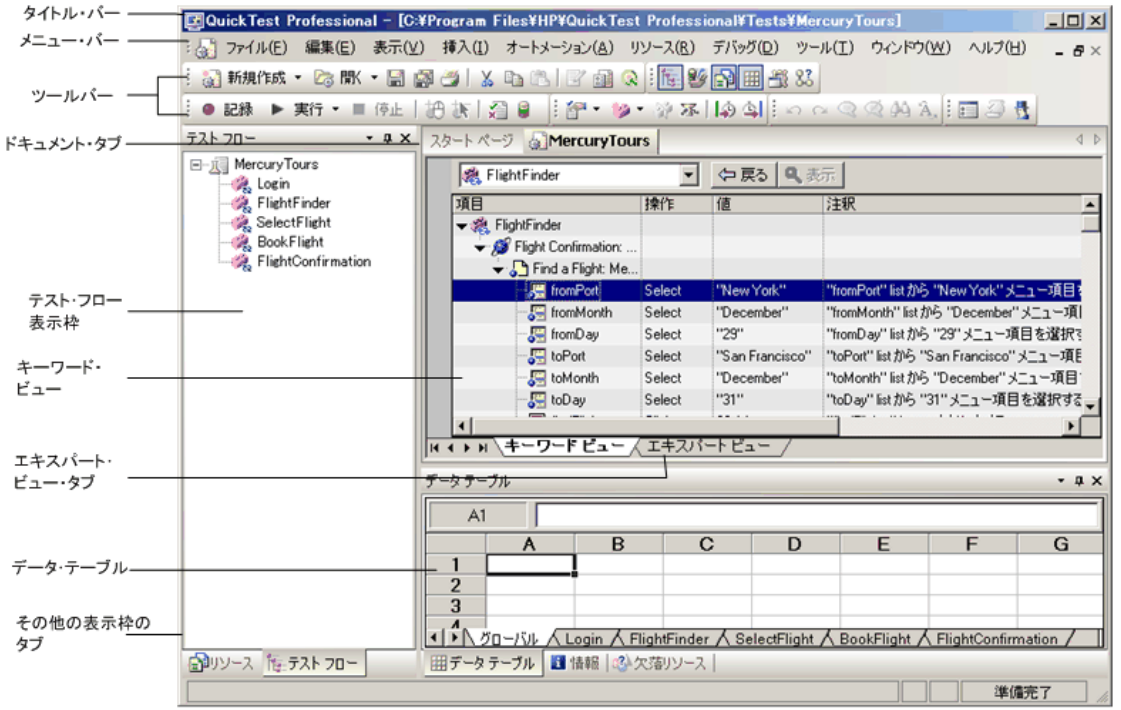
Business Process Testing を使用するには，Quality Center プロジェクトが格納されている Quality Center サーバに QuickTest を接続する必要があります。このサーバによって，QuickTest と Quality Center プロジェクト間の接続が処理されます。Business Process Testing では，QuickTest Professional 9.0 以降から，Quality Center 9.0 以降にのみ接続できます。

Quality Center プロジェクトでは，テスト対象アプリケーションのコンポーネントおよび実行セッションに関する情報が格納されます。この情報には，ビジネス・プロセス・テストを作成して実行するのに必要なリソース・ファイルと設定などがすべて含まれています。初めて Quality Center サーバおよびプロジェクトに QuickTest を接続すると，QuickTest によって標準の Business Process Testing フォルダとファイルがプロジェクトに設定されます。これでビジネス・コンポーネントに必要なリソースと設定を準備できるだけでなく，直観的なキーワード駆動型のキーワード・ビューを使ってビジネス・コンポーネントの作成，操作，デバッグを行うことができます。

注：Quality Center プロジェクトはパスワードで保護されるため，ユーザ名とパスワードを指定する必要があります。

QuickTest ウィンドウ

QuickTest ウィンドウのドキュメント領域には，テスト・ドキュメントが表示されます。



1 つのコンポーネントまたはアプリケーション領域で作業しながら，同時に 1 つ以上の関数ライブラリで作業できます（必要に応じて，アクティブ・ドキュメント 1 つをドキュメント領域に表示することも，または開いているドキュメントを重ねて表示または並べて表示することもできます）。詳細については，824 ページ「複数のドキュメントを使った作業」を参照してください。

ドキュメント領域

QuickTest ウィンドウのドキュメント領域には，次の項目を表示できます。

- **ビジネス・コンポーネント**：キーワードと操作を使用して，ビジネス・コンポーネントを作成，表示，変更できます。詳細については，第 17 章「キーワード・ビューを使った作業」を参照してください。
- **スクリプト・コンポーネント**：キーワード・ビューまたはエキスパート・ビュー（後述）。スクリプト・コンポーネントの詳細については，第 16 章「スクリプト・コンポーネントの作成」を参照してください。エキスパート・ビューの詳細については，『**HP QuickTest Professional ユーザーズ・ガイド**』を参照してください。
- **アプリケーション領域**：コンポーネントのリソースおよび設定を定義できます。詳細については，第 13 章「アプリケーション領域の管理」を参照してください。
- **関数ライブラリ**：コンポーネントで使用する関数およびサブルーチンを作成，表示，変更できます。詳細については，第 12 章「ユーザ定義関数および関数ライブラリを使用した作業」を参照してください。
- **スタート・ページ**：QuickTest の入口で，［プロセス ガイダンス］へのリンクが提供されています。ショートカット・ボタンを使用して，新規および既存のドキュメントを開けます。詳細については，55 ページ「スタート・ページ」を参照してください。

QuickTest ウィンドウの主な要素

ドキュメント領域に加えて，QuickTest ウィンドウには，次の主な要素が含まれています。

- **QuickTest タイトル・バー**：アクティブなドキュメントの名前が表示されます。前回保存したとき以降に加えられた変更がある場合は，タイトル・バーのドキュメント名の横にアスタリスク（*****）が表示されます。
- **メニュー・バー**：QuickTest コマンドのメニューが表示されます。
- **標準ツールバー**：ドキュメント管理を支援するボタンが含まれています。
- **オートメーション・ツールバー**：テスト・プロセスを支援するボタンが含まれています。
- **デバッグ・ツールバー**：ドキュメントのデバッグを支援するボタンが含まれています（標準では表示されません）。

- **編集ツールバー**：関数ライブラリの編集を支援するボタンが含まれています。
- **挿入ツールバー**：関数ライブラリ内のステートメントを使用する際に役立つボタンが含まれています。
- **ツール・ツールバー**：テスト・プロセス中に使用するツールのボタンが含まれています。
- **表示ツールバー**：ドキュメントの参照に使用するボタンが含まれています。
- **ドキュメント・タブおよびスクロール矢印**：アクティブにする（フォーカスを移動する）ドキュメントのタブを選択することで，ドキュメント領域に開いているドキュメント間を移動できます。すべてのタブを同時に表示するのに十分なスペースがドキュメント領域にない場合は，左右の矢印を使用して，開いているドキュメント間をスクロールできます。
- **キーワード・ビュー**：各ステップが，モジュール化されたアイコン・ベースの表形式で表示されます。詳細については，第 17 章「キーワード・ビューを使った作業」を参照してください。
- **ステータス・バー**：QuickTest アプリケーションのステータスおよびほかの関連情報が表示されます。

［**表示**］メニューから，次の表示枠の表示と非表示の切り替えができます。

- **使用可能なキーワード**：コンポーネントで使用できるすべてのキーワードが表示されます。コンポーネントにオブジェクトまたは関数に対する呼び出しをドラッグ・アンド・ドロップできます
- **デバッグ・ビューア**：ドキュメントのデバッグを支援します。［デバッグ ビューア］表示枠には，［**ウォッチ**］，［**変数**］，および［**コマンド**］タブがあります。
- **情報**：関数ライブラリのスクリプトで見つかった構文エラーのリストが表示されます。
- **欠落リソース**：未割り当ての共有オブジェクト・リポジトリ，共有オブジェクト・リポジトリに結び付いているパラメータなど，コンポーネントで指定されているのに見つからなかったリソースの一覧が表示されます。［欠落リソース］表示枠では，これらを見つけたり，アプリケーション領域から削除できます。
- **プロセス ガイダンス**：特定のプロセスを最も効果的に実行する手順と詳細を提供する 2 つの表示枠が表示されます。たとえば，QuickTest でのコンポーネントの作成に関する情報などが表示されます。プロセス・ガイダンスの［アクティビティ］表示枠には，コンポーネントへのステップの追加など，実行可能なアクティビティが表示されます。プロセス・ガイダンスの［詳細］表示枠には，

選択したアクティビティに対して実行する必要があるタスクについて説明が表示されます。また組織によっては，これらの表示枠からアクセスできるプロセス・ガイダンスが提供される場合があります

- **リソース**：現在のコンポーネントに関連付けられているすべてのリソースが表示され，これらのリソースを表示したり開いたりできます。
- **タスク**：現在のコンポーネントまたはアプリケーション領域に定義されたタスクが表示され，管理できます。［タスク］表示枠には，コンポーネントまたは現在開いている関数ライブラリの TODO コメント・ステップも表示されます。

QuickTest ウィンドウを変更して，ユーザ定義メニューを作成し，既存のメニューおよびツールバー の外観をカスタマイズできます。詳細については，811 ページ「ツールバーとメニューのカスタマイズ」を参照してください。

また，ほとんどの要素を移動，サイズ変更，表示，非表示することによって，QuickTest ウィンドウのレイアウトをカスタマイズできます。QuickTest は，お気に入りのレイアウト設定を記憶して，次のセッションはカスタマイズされたレイアウトを使って開きます。詳細については，797 ページ「QuickTest ウィンドウのレイアウト変更」を参照してください。

QuickTest ウィンドウの外観の変更

標準では，QuickTest ウィンドウには Microsoft Office 2003 のテーマが使用されます。必要に応じて，QuickTest のメイン・ウィンドウのルック・アンド・フィールは変更できます。

QuickTest のメイン・ウィンドウの外観を変更するには，次の手順を実行します。

QuickTest ウィンドウで，［**表示**］＞［**ウィンドウのテーマ**］を選択し，使用可能なテーマのリストからウィンドウの表示方法を選択します。たとえば，Microsoft Office 2000 または Microsoft Windows XP のテーマを適用できます。

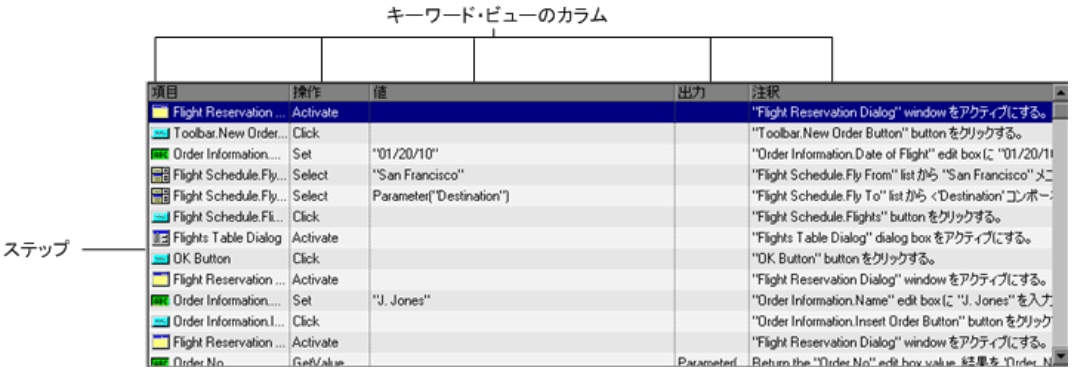
注：QuickTest ウィンドウに Microsoft Windows XP テーマを適用できるのは，コンピュータが Windows XP テーマを使用するように設定されている場合のみです。

ヒント：［テスト結果］ウィンドウに使用するテーマも変更できます。詳細については，688 ページ「［テスト結果］ウィンドウの外観の変更」を参照してください。

キーワード・ビュー

キーワード・ビューでは，コンポーネントのステップをキーワード駆動のモジュール化された表形式で作成および表示できます。キーワード・ビューは，表に似た表示形式で構成されます。各ステップは表内では個別の行として表示され，各カラムはステップの別々の部分を表します。表示されるカラムは必要に応じて変更できます。

コンポーネントを作成および変更するには，キーワード・ビューの中で項目と操作を選択し，必要に応じて情報を入力します。完了した各ステップは自動的にドキュメント化され，テストのステップの説明がわかりやすい言葉で表されます。



アプリケーション領域

各ビジネス・コンポーネントは，アプリケーション領域に基づいています。アプリケーション領域は，設定や特定のリソース・ファイル（アプリケーションによって使用されるテスト・オブジェクトを含む関数ライブラリ・ファイル，共有オブジェクト・リポジトリ，関連するアドイン，回復シナリオ・ファイルなど）へのリンクを提供します。これらのアセットは，アプリケーション領域ウィンドウで定義します。

アプリケーション領域ウィンドウには，左側のサイドバー上のボタンからアクセスする 4 つの表示枠があります。

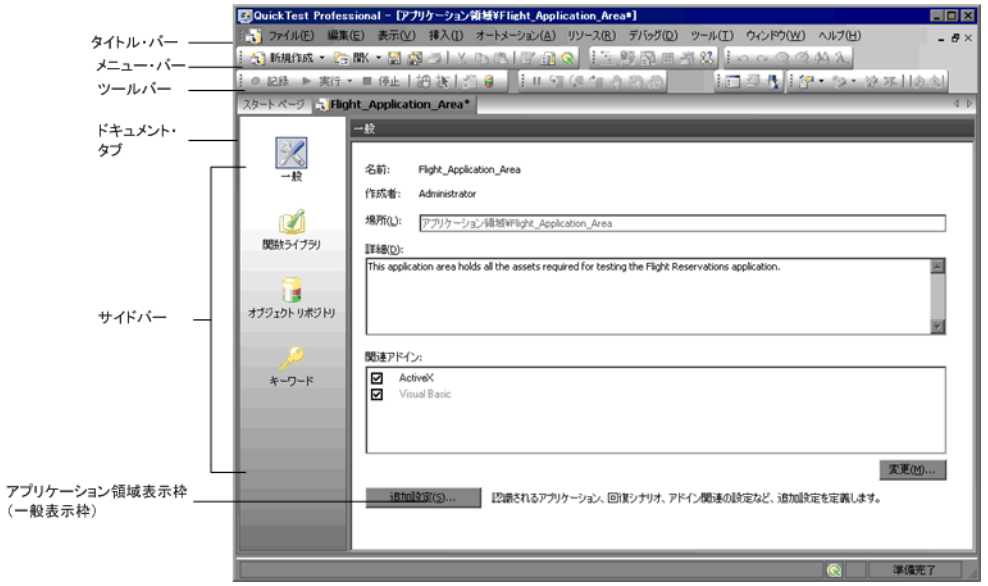
一般：アプリケーション領域に関する一般的な情報が表示され，関連するアドイン，回復シナリオ，ほかの設定の指定など，一般的な設定を変更できます。

関数ライブラリ：関数ライブラリをこのアプリケーション領域に関連付けて，優先順位を付けることができます。

オブジェクト・リポジトリ：共有オブジェクト・リポジトリをこのアプリケーション領域に関連付けて，優先順位を付けることができます。

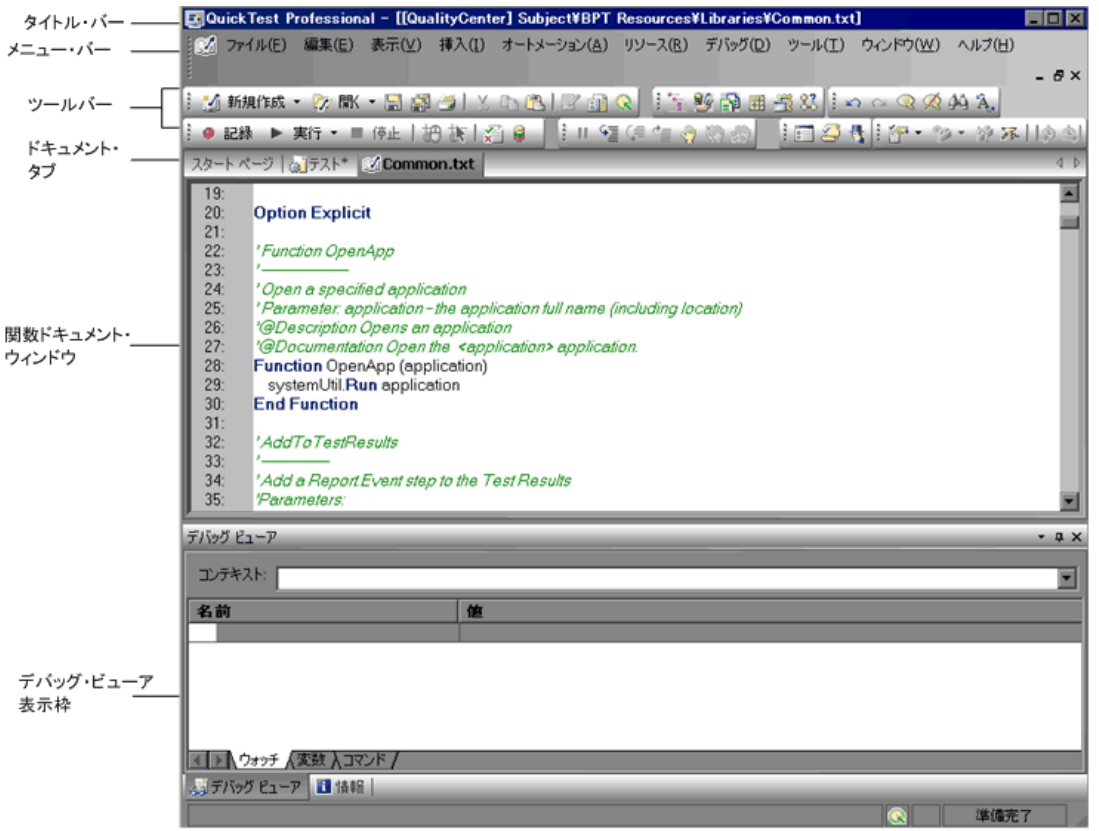
キーワード：このアプリケーション領域で使用可能なキーワードを設定し，個々のプロパティを表示できます。

詳細については，第 13 章「アプリケーション領域の管理」を参照してください。



関数ライブラリ

QuickTest の組み込みエディタでは，エキスパート・ビューで使用できるものと同じ編集機能を使用して，関数ライブラリを作成およびデバッグできます。各関数ライブラリは，VBscript 関数，サブルーチン，クラス，モジュールなどが含まれる独立した QuickTest ドキュメントです。個々の関数ライブラリは，すでに開いているコンポーネントに加えて，それぞれ別のウィンドウに開きます。関数ライブラリは複数を同時に開いて作業できます。関数ライブラリの編集が終了したら，関数ライブラリを閉じて，QuickTest セッションは開いたままにしておくことができます。また，開いているすべての関数ライブラリを同時に閉じることもできます 詳細については，第 12 章「ユーザ定義関数および関数ライブラリを使用した作業」を参照してください。



スタート・ページ

［スタート ページ］では QuickTest の［ようこそ］画面が表示され，新機能の詳細へのリンクなど，このリリースの新機能に関する情報を提供します。また，QuickTest の使用に関するベスト・プラクティスを提供するツールであるプロセス・ガイダンスへのリンクも提供します。所属する組織に独自のカスタム・プロセスに対する記述がある場合は，これらのプロセスは［**プロセス ガイダンス リスト**］からも使用できる場合があります 詳細については，857 ページ「プロセス・ガイダンスを使った作業」を参照してください。

［**最近使ったファイル**］の一覧からのドキュメントを開くことができますし，［**ようこそ**］領域でボタンをクリックして，新規または既存のドキュメントを開くこともできます。

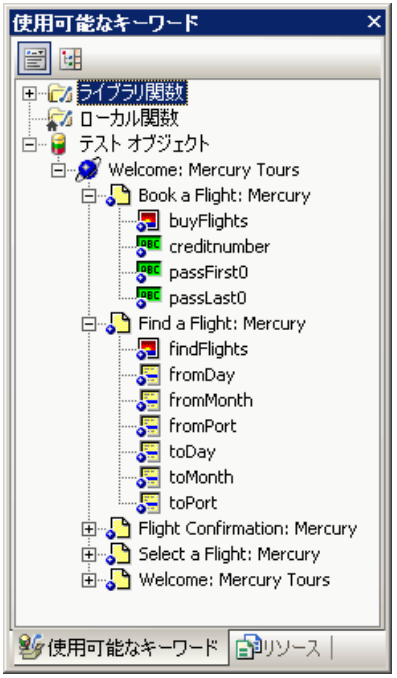
クリック	動作
	新規テストを開きます。
	新規ビジネス・コンポーネントを開きます。
	新規アプリケーション領域を開きます。
	新規関数ライブラリを開きます。
	既存のテストを開きます。
	既存のビジネス・コンポーネントを開きます。
	既存のアプリケーション領域を開きます。
	既存の関数ライブラリを開きます。

ヒント：次に QuickTest を起動するときに，QuickTest の［スタート ページ］を表示したくない場合は，［**起動時に［スタート ページ］ウィンドウを表示しない**］チェック・ボックスを選択します。このオプションを選択すると，別の QuickTest ドキュメントを開くとすぐに，現在の QuickTest セッションの［スタート ページ］も自動的に非表示になります。もう一度［スタート ページ］を表示するには，［**表示**］＞［**スタート ページ**］を選択します。

［使用可能なキーワード］表示枠

［使用可能なキーワード］表示枠では，オブジェクトまたは関数呼び出しをコンポーネントにドラッグ・アンド・ドロップできます。オブジェクトをコンポーネントにドラッグ・アンド・ドロップすると，QuickTest では，そのオブジェクトの標準操作を持つステップが挿入されます。コンポーネントに関数をドラッグ・アンド・ドロップすると，QuickTest では，その関数の呼び出しが挿入されます。［使用可能なキーワード］表示枠を表示するには，［**使用可能なキーワード**］表示枠ボタンをクリックするか，［**表示**］＞［**使用可能なキー**

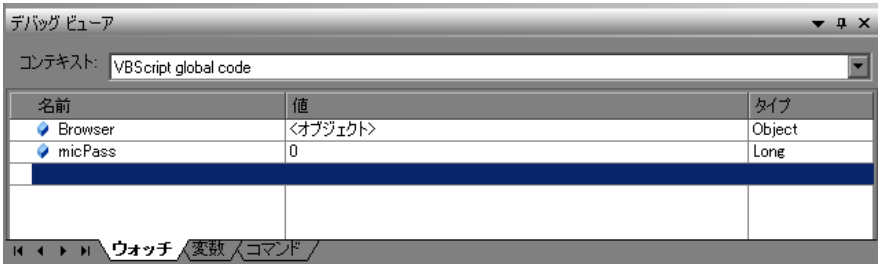
ワード］を選択します。



詳細については，831 ページ「[使用可能なキーワード］表示枠について」を参照してください。

［デバッグ ビューア］表示枠

［デバッグ ビューア］表示枠は，関数ライブラリのデバッグに役立ちます。［デバッグ ビューア］表示枠を表示するには，［**表示**］＞［**デバッグ ビューア**］を選択します。



この表示枠には，［ウォッチ］，［変数］，および［コマンド］の 3 つのタブがあります。

ウォッチ

［ウォッチ］タブには，このタブに追加した変数または VBScript 式の現在値と型が表示されます。また，表示されている変数およびプロパティの値を設定したり変更することもできます。

変数

［変数］タブには，デバッグしている実行セッション中に実行された最終ステップまでに認識されたすべての変数の現在値と型が表示されます。また，表示されている変数の値を設定したり変更することもできます。

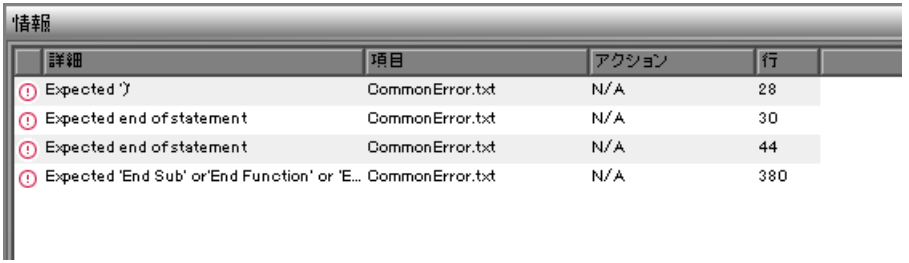
コマンド

［コマンド］タブでは，スクリプト行を実行して，関数ライブラリに含まれている変数または VBScript オブジェクトの現在値を設定したり変更できます。

詳細については，751 ページ「［デバッグ ビューア］表示枠」を参照してください。

［情報］表示枠

［情報］表示枠には，関数ライブラリのスクリプトで見つかった構文エラーのリストが表示されます。［情報］表示枠の表示 / 非表示を切り替えるには，［**表示**］＞［**情報**］を選択します。

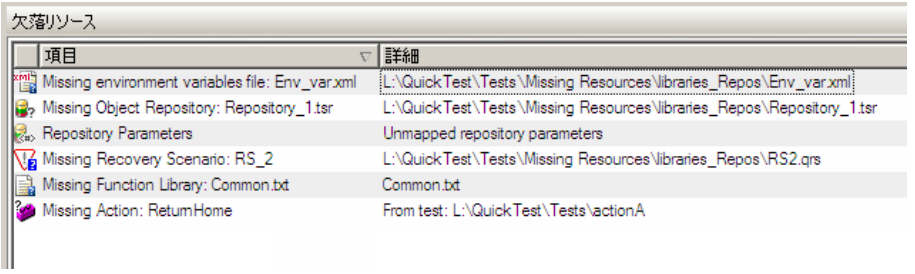


構文エラーをダブルクリックすると，関数ライブラリ内でそのエラーを見つけ，その場で修正できます。詳細については，348 ページ「VBScript 構文エラーの処理方法」を参照してください。

［欠落リソース］表示枠

［欠落リソース］表示枠には，テストで指定されているにもかかわらず見つからなかったリソースのリストが表示されます。欠落リソースには，欠落関数ライブラリ，欠落回復シナリオ，未割り当ての共有オブジェクト・リポジトリ，および共有オブジェクト・リポジトリに結び付いているパラメータが含まれます。

コンポーネントまたはアプリケーション領域を開くたびに，指定されたすべてのリソースがアクセス可能かどうか自動的にチェックされます。アクセスできないリソースが見つかった場合，QuickTest ではそのリソースが［欠落リソース］表示枠に一覧表示されます。［欠落リソース］表示枠が表示されていない場合でも，存在しないリソースが検出されると自動的に表示されます。［欠落リソース］表示枠の表示 / 非表示を切り替えるには，［**表示**］＞［**欠落リソース**］を選択するか，［**欠落リソース**］ボタンをクリックします。



［欠落リソース］表示枠には次のカラムがあります。

- ［**項目**］カラムには欠落リソースが表示されます。
- ［**詳細**］カラムには，QuickTest でリソースが見つかりそうな位置など，各欠落リソースに関する情報が表示されます。

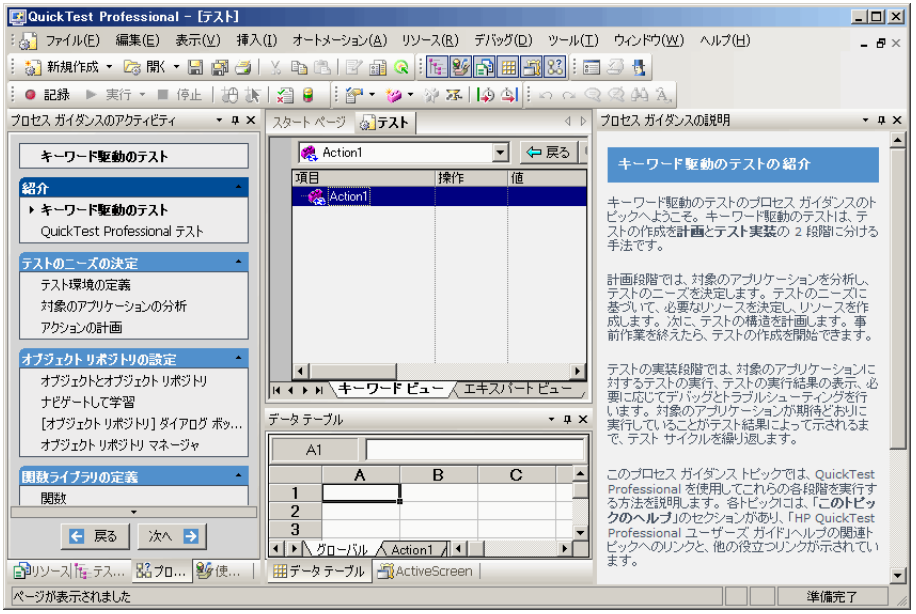
存在しないリソースをダブルクリックすると，そのリソースを再割り当て，または削除できます表示枠でフィルタを設定して，特定のタイプの欠落リソース（［欠落しているオブジェクト リポジトリ］など）を表示し，それ以外のタイプを非表示にすることもできます。

詳細については，845 ページ「欠落リソースの処理」を参照してください。

［プロセス ガイダンス］表示枠

プロセス・ガイダンスは，特定のプロセスを最も効果的に実行する方法に関する手順と詳細を提供するツールです。プロセス・ガイダンスを使用すると，新しいプロセスについて学習したり，すでに知っているプロセスを実行するための望ましい方法について学習できます。

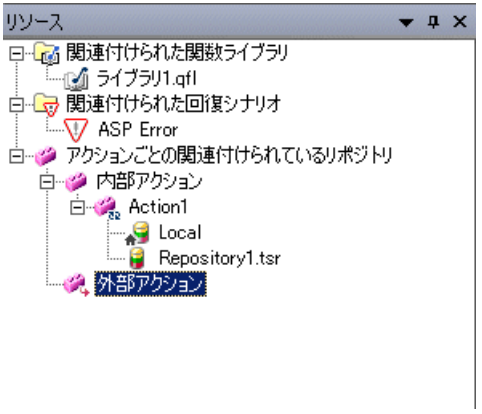
プロセス・ガイダンスは，［**プロセス ガイダンスのアクティビティ**］表示枠と［**プロセス ガイダンスの詳細**］表示枠という 2 つの表示枠に表示されます。これらの表示枠の表示と非表示を切り替えるには，［**表示**］＞［**プロセス ガイダンス**］を選択するか，または［**プロセス ガイダンス**］表示枠トグル・ボタンをクリックします。



［**プロセス ガイダンスのアクティビティ**］表示枠（左側に表示）には，選択したプロセスの一部であるアクティビティが表示されます。［**プロセス ガイダンスの説明**］表示枠（右側に表示）には，選択したアクティビティに関するトピック（説明）が表示されます。詳細については，第 36 章「プロセス・ガイダンスを使った作業」を参照してください。

［リソース］表示枠

コンポーネントは，関数ライブラリ，回復シナリオ，およびオブジェクト・リポジトリなどのリソースに関連付けられています（アプリケーション領域を経由）。QuickTest では，コンポーネントに関連付けられているすべてのリソースが［リソース］表示枠に表示されます。［リソース］表示枠では，コンポーネントのすべてのリソースを表示して開くことができます。［リソース］表示枠を表示するには，［**リソース**］表示枠ボタンをクリックするか，［**表示**］＞［**リソース**］を選択します。



詳細については，827 ページ「［リソース］表示枠」を参照してください。

［タスク］表示枠

［タスク］表示枠では，TODO タスクを作成，表示，管理できます。TODO タスクはコンポーネントで行う必要があることです。たとえば，テスト・ドキュメントの受け渡しに関連した情報を提供したり，アプリケーションで新しいページをテストするステップを追加するようにメモを追加することなどです。タスクはほかの人に割り当てることができ，終われば完了としてマークできます。TODO タスクはコンポーネントと一緒に保存されます。

［タスク］表示枠では，開いている関数ライブラリにある TODO コメントを表示できます（ステップの近くにある指示や注記など）。［タスク］表示枠の表示 / 非表示を切り替えるには，［**表示**］＞［**タスク**］を選択するか，［**タスク** ］表示枠ツールバー・ボタンをクリックします。

詳細については，835 ページ「QuickTest のタスクとコメントの管理」を参照してください。

QuickTest コマンドの使用方法

QuickTest のコマンドはメニュー・バーまたはツールバーから選択できます。QuickTest では，コンポーネント，およびアプリケーション領域に対して，それぞれ異なるコマンドとツールバーのセットが表示されます。各セットは，作成または変更されるドキュメントに合わせてカスタマイズされています。QuickTest のコマンドには，ショートカット・キーを押したり，ショートカット（右クリック）・メニューでコマンドを選択できるものがあります。メニューとツールバーはアクティブなドキュメントの種類に応じて有効になります。

ほとんどのコマンドはメニュー・バーから，またはショートカット・キーを押すことにより使用できます。よく使用する QuickTest コマンドは，ツールバーのボタンをクリックして実行できます。詳細については，次を参照してください。

- 66 ページ「QuickTest ツールバー」
- 69 ページ「[ファイル] メニューのコマンド」
- 72 ページ「[編集] メニューのコマンド」
- 77 ページ「[表示] メニューのコマンド」
- 78 ページ「[挿入] メニューのコマンド」
- 80 ページ「[オートメーション] メニューのコマンド」
- 83 ページ「[リソース] メニューのコマンド」
- 84 ページ「[デバッグ] メニューのコマンド」
- 85 ページ「[ツール] メニューのコマンド」
- 87 ページ「[ウィンドウ] メニューのコマンド」
- 87 ページ「[ヘルプ] メニューのコマンド」
- 89 ページ「[データ テーブル] メニューのコマンド」
- 91 ページ「その他の QuickTest コマンド」

QuickTest ツールバー

本項では，QuickTest の組み込みツールバーについて説明します。

QuickTest の外観は，［カスタマイズ］ダイアログ・ボックスおよび［ボタンの外観］ダイアログ・ボックスを使って，追加，削除，再配置，または変更できます。詳細については，811 ページ「ツールバーとメニューのカスタマイズ」を参照してください。

注：すべてのツールバーがすべてのドキュメントの種類に対応しているとはかぎりません。ここでは，コンポーネント，アプリケーション領域，および関数ライブラリに関連するツールバーについてのみ説明します。

標準ツールバー

標準ツールバーには，コンポーネント，アプリケーション領域，または関数ライブラリを管理するためのボタンがあります。

標準ツールバー・ボタンについては，69 ページ「［ファイル］メニューのコマンド」を参照してください。

注：［**新規**］および［**開く**］ボタン用のアイコンは，コンポーネント，アプリケーション領域，または関数ライブラリなど，アクティブなドキュメントの種類に応じて変わります。

ビジネス・コンポーネントの使用方法の詳細については，第 15 章「ビジネス・コンポーネントの使用」を参照してください。スクリプト・コンポーネントの使用方法の詳細については，第 16 章「スクリプト・コンポーネントの作成」を参照してください。アプリケーション領域を使った作業の詳細については，第 13 章「アプリケーション領域の管理」を参照してください。関数ライブラリの使用方法の詳細については，第 12 章「ユーザ定義関数および関数ライブラリを使用した作業」を参照してください。

オートメーション・ツールバー

オートメーション・ツールバーには，コンポーネントの記録と実行のためのボタンがあります。

オートメーション・ツールバー・ボタンの詳細については，80 ページ「[オートメーション] メニューのコマンド」を参照してください。

デバッグ・ツールバー

デバッグ・ツールバーには，コンポーネントおよび関連付けられている関数ライブラリのステップをデバッグする際に使用するコマンドのボタンがあります。

デバッグ・ツールバー・ボタンの詳細については，84 ページ「[デバッグ] メニューのコマンド」を参照してください。

編集ツールバー

編集ツールバーには，関数ライブラリの編集に使用するコマンドのボタンがあります。

編集ツールバー・ボタンの詳細については，72 ページ「[編集] メニューのコマンド」を参照してください。

挿入ツールバー

挿入ツールバーには，関数ライブラリの操作に使用するコマンドのボタンがあります。

挿入ツールバー・ボタンの詳細については，78 ページ「[挿入] メニューのコマンド」を参照してください。

注：［**ステップ ジェネレータ**］ボタンだけが関数ライブラリに関連しています。これらのボタンのいずれもコンポーネントには関係ありません。

ツール・ツールバー

ツール・ツールバーには，テストでの作業に役に立つツールにアクセスするために使用するコマンドのボタンがあります。

ツール・ツールバー・ボタンの詳細については，85 ページ「[ツール］メニューのコマンド」を参照してください。

構文チェック・ボタンは関数ライブラリ（および QuickTest テスト）にのみ関係し，コンポーネントには関係していません。

表示ツールバー

表示ツールバーには，QuickTest ウィンドウのさまざまな要素を表示するためのボタンがあります。

表示ツールバー・ボタンの詳細については，77 ページ「[表示］メニューのコマンド」を参照してください。

注：ActiveScreen ボタンと**データ・テーブル**・ボタンは，テストにのみ適用されます。

QuickTest コマンドの実行

ツールバー・ボタンをクリックして頻繁に使用するコマンドを実行することに加えて，対応するメニュー・オプションを選択することで，ほとんどの QuickTest コマンドを実行できます。QuickTest コマンドの一部は，該当するショートカット・キーを押して実行することもできます。

［ファイル］メニューのコマンド

次の［**ファイル**］メニューのコマンドを使用すると，コンポーネント，アプリケーション領域，または関数ライブラリを管理できます。

	コマンド	ショートカット・キー	機能
	［新規作成］＞［テスト］	CTRL+N	新規のテストを作成する。
	［新規作成］＞［ビジネス コンポーネント］	CTRL+SHIFT+N	新規のビジネス・コンポーネントを作成する。
	［新規作成］＞［スクリプト コンポーネント］		新規のスクリプト・コンポーネントを作成する。
	［新規作成］＞［アプリケーション領域］	CTRL+ALT+N	新規のアプリケーション領域を作成する。
	［新規作成］＞［関数ライブラリ］	SHIFT+ALT+N	新規の関数ライブラリを作成する。
	［開く］＞［テスト］	CTRL+O	既存のテストを開く。
	［開く］＞［ビジネス / スクリプト コンポーネント］	CTRL+SHIFT+O	既存のビジネス・コンポーネントまたはスクリプト化コンポーネントを開く。
	［開く］＞［アプリケーション領域］	CTRL+ALT+O	既存のアプリケーション領域を開く。
	［開く］＞［関数ライブラリ］	SHIFT+ALT+O	既存の関数ライブラリを開く。
	［閉じる］		アクティブな関数ライブラリを閉じる。
	［すべての関数ライブラリを閉じる］		開いているすべての関数ライブラリを閉じる。

	コマンド	ショートカット・キー	機能
	［Quality Center への接続］		［Quality Center への接続］ダイアログ・ボックスを開き，Quality Center プロジェクトに接続できます。 **ヒント**：接続を管理するにはステータス・バーの Quality Center アイコンをダブルクリックします。接続情報を表示するにはステータス・バーの Quality Center アイコンにカーソルを合わせます。
	Quality Center バージョン・コントロール		QuickTest アセットと Quality Center のベースラインのバージョンを管理するオプションのサブメニューを提供します。バージョン関連のサブメニューは，バージョン管理の可能な Quality Center プロジェクトに接続している場合にのみ利用できます。
	［上書き保存］	CTRL+S	アクティブなドキュメントを保存する。
	［名前を付けて保存］		対応する［保存］ダイアログ・ボックスを開き，開いているドキュメントを保存できる。
	［テストをリソースと保存］		現在のテストの独立したコピーをリソースファイルと一緒に保存します。（テストでのみ利用できます）。
	［すべて保存］		開いているすべてのドキュメントを保存する。
	［編集可能にする］		読み取り専用の関数ライブラリを編集できるようにする。

	コマンド	ショートカット・キー	機能
	[Zip ファイルにテストをエクスポート]	CTRL+ALT+S	アクティブなテストの zip ファイルを作成する。（テストでのみ利用できます）。
	[Zip ファイルからテストをインポート]	CTRL+ALT+I	zip ファイルからテストをインポートする。（テストでのみ利用できます）。
	[スクリプト コンポーネントに変換]	CTRL+ALT+C	ビジネス・コンポーネントをスクリプト・コンポーネントに変換する。
	[印刷]	CTRL+P	アクティブなドキュメントを印刷する。
	[プリンタの設定]		キーワード・ビューを印刷時の形式で表示して，ページ設定を変更できる。
	[設定]		[設定] ダイアログ・ボックスを開き，開いているドキュメントの設定を定義できる（関数ライブラリでは利用できません）。
	[プロセス ガイダンス管理]		[プロセス ガイダンス管理] ダイアログ・ボックスを開き，QuickTest で使用できるプロセスの一覧を管理できる。
	[アプリケーション領域の変更]		コンポーネントを別のアプリケーション領域に関連付けることができる。
	[ライブラリ '＜関数ライブラリ名＞' を '＜ドキュメント名＞' に関連付ける]		アクティブな関数ライブラリを開いているドキュメントに関連付ける（関数ライブラリからのみ使用可能）。
	[最近使用したファイル]		最近表示したファイルを表示する。
	[終了]		QuickTest セッションを閉じる。

［**ファイル**］メニューのコマンドの多くは，**標準**ツールバーからも使用できます（66 ページを参照）。

［編集］メニューのコマンド

次の［**編集**］メニューのコマンドを使用すると，コンポーネントまたは関数ライブラリ・ステップを管理できます。

	コマンド	ショートカット・キー	機能
	［元に戻す］	CTRL+Z	直前に実行したコマンドを元に戻すか，直前に入力したエントリを削除する。
	［やり直し］	CTRL+Y	直前に実行された［元に戻す］コマンドの操作をやり直す。
	［切り取り］	CTRL+X	選択した内容をドキュメントから切り取る。
	［コピー］	CTRL+C	選択した内容をドキュメントからコピーする。
	［貼り付け］	CTRL+V	選択した内容をドキュメントに貼り付ける。
	［削除］	DELETE	選択した内容をドキュメントから削除する。
	［注釈をクリップボードにコピーする］		キーワード・ビューの［**注釈**］カラムの内容をコピーして，外部アプリケーションに貼り付けられる。
	［アクション］＞［アクションの分割］		アクションを 2 つの兄弟アクションに分割したり，親子関係のネスト・アクションに分割したりできる。（テストでのみ利用できます）。
	［アクション］＞［アクション名の変更］	SHIFT+F2	アクション名を変更する。（テストでのみ利用できます）。

	コマンド	ショートカット・キー	機能
	［アクション］＞ ［アクションを削除］		選択したアクションに対する呼び出しを削除するか，またはアクティブなテストからアクションとその呼び出しを削除できる。（テストでのみ利用できます）。
	［アクション］＞ ［アクションのプロパティ］		格納されたアクションのオプション，パラメータ，および関連付けられているオブジェクト・リポジトリを指定できる。（テストでのみ利用できます）。
	［アクション］＞ ［アクション呼び出しプロパティ］		データ・テーブルの行数に応じて実行の反復数を指定したり，入力パラメータの値および出力パラメータの保管場所を定義したりできる。（テストでのみ利用できます）。
	［ステップのプロパティ］＞ ［コメントのプロパティ］	CTRL+ENTER; ALT+ENTER	コメント・ステップの［コメントのプロパティ］ダイアログ・ボックスを開く。選択したステップがコメントの場合にのみ使用できる。
	［ステップのプロパティ］＞ ［オブジェクトのプロパティ］	CTRL+ENTER; ALT+ENTER	選択したオブジェクトの［オブジェクトのプロパティ］ダイアログ・ボックスを開く。選択したステップがテスト・オブジェクトの場合にのみ使用できる。
	［ステップのプロパティ］＞ ［チェックポイントのプロパティ］		選択したオブジェクトの関連する［チェックポイントのプロパティ］ダイアログ・ボックスを開く。選択したステップがチェックポイント・ステップの場合にのみ使用できる。

	コマンド	ショートカット・キー	機能
	［ステップのプロパティ］＞［出力値のプロパティ］		選択したオブジェクトの関連する［出力値のプロパティ］ダイアログ・ボックスを開く。選択したステップが出力値ステップの場合にのみ使用できる。
	［ステップのプロパティ］＞［レポートのプロパティ］	CTRL+ENTER; ALT+ENTER	レポート・ステップの［レポートのプロパティ］ダイアログ・ボックスを表示する。選択したステップが **Reporter.ReportEvent** ステップの場合にのみ使用できる。
	［検索］	CTRL+F	特定の文字列を検索する。（テストと関数ライブラリでのみ使用可能）
	［置換］	CTRL+H	特定の文字列を検索して置換する。（テストと関数ライブラリでのみ使用可能）
	［移動］	CTRL+G	関数ライブラリの特定の行にカーソルを移動する。（テストでのみ利用できます）。
	［ブックマーク］	CTRL+B	簡単に移動できるように，関数ライブラリにブックマークを作成する。（テストでのみ利用できます）。
	［詳細設定］＞［コメント ブロック］	CTRL+M	現在の行または関数ライブラリで選択した行をコメント・アウトする。（テストでのみ利用できます）。
	［詳細設定］＞［コメント ブロック解除］	CTRL+SHIFT+M	現在の行または関数ライブラリで選択した行のコメント書式設定を削除する。（テストでのみ利用できます）。

	コマンド	ショートカット・キー	機能
	［詳細設定］＞［インデント］	TAB	［エディタ オプション］ダイアログ・ボックスで定義されたタブ間隔に従って，関数ライブラリのステップをインデントする。（テストでのみ利用できます）。
	［詳細設定］＞［インデント解除］	BACKSPACE	［エディタ オプション］ダイアログ・ボックスで定義されたタブ間隔に従って，関数ライブラリのステップをインデント解除する（インデントを減らす）。（テストでのみ利用できます）。
	［詳細設定］＞［関数定義に移動］	ALT+G	関数ライブラリで選択した関数の定義に移動する。
	［詳細設定］＞［単語入力候補］	CTRL+SPACE	VBScript メソッドまたは関数ライブラリにあるオブジェクトの最初の部分を入力すると，入力候補を表示する。
	［詳細設定］＞［引数詳細］	CTRL+SHIFT+SPACE	関数ライブラリにあるメソッドの構文を表示する。
	［詳細設定］＞［スクリプトに "With" を追加］	CTRL+W	エキスパート・ビューに表示されたアクションに **With** ステートメントを生成し，**With** ステートメント内で IntelliSense を有効にする。（テストでのみ利用できます）。
	［詳細設定］＞［"With" ステートメントを削除］	CTRL+SHIFT+W	エキスパート・ビューに表示されているアクションの **With** ステートメントを通常の（単独行の）VBScript ステートメントに変換する。（テストでのみ利用できます）。
	［オプション ステップ］		オプション・ステップ（実行セッションを正しく完了するのに必要ないステップ）を挿入する。（テストでのみ利用できます）。

［**編集**］メニューのコマンドの多くは，**編集**ツールバーからも使用できます（67 ページを参照）。

［表示］メニューのコマンド

次の［**表示**］メニューのコマンドを使用すると，QuickTest を画面に表示する方法を管理できます。

	コマンド	機能
	［スタート ページ］	スタート・ページを開きます（スタート・ページが閉じられているときのみ使用可能）。
	［ActiveScreen］	Active Screen を表示する （テストでのみ利用できます）。
	［使用可能なキーワード］	［使用可能なキーワード］表示枠 の表示 / 非表示を切り替える。
	［データ テーブル］	データ・テーブルを表示する （テストでのみ利用できます）。
	［デバッグ ビューア］	［デバッグ ビューア］表示枠 の表示 / 非表示を切り替える。
	［情報］	［情報］表示枠 の表示 / 非表示を切り替える。
	［欠落リソース］	［欠落リソース］表示枠 の表示 / 非表示を切り替える。
	［プロセス ガイダンス］	［プロセス ガイダンス］表示枠 の表示 / 非表示を切り替える。
	［リソース］	［リソース］表示枠 の表示 / 非表示を切り替える。
	［テスト フロー］	テスト・フロー表示枠の表示 / 非表示を切り替える （テストでのみ利用できます）。
	［タスク］	［タスク］表示枠 の表示 / 非表示を切り替える。
	［すべて展開］	キーワード・ビューですべてのステップを展開する。
	［すべて折りたたみ］	キーワード・ビューですべてのステップを閉じる。
	［キーワード ビュー］	エキスパート・ビューが表示されているときに，キーワード・ビューを表示する （テストでのみ利用できます）。
	［エキスパート ビュー］	キーワード・ビューが表示されているときに，エキスパート・ビューを表示する （テストでのみ利用できます）。
	［ツールバー］	QuickTest ツールバーの表示と非表示を切り替える。
	［ウィンドウのテーマ］	QuickTest ウィンドウに適用するテーマを選択できる。

［**表示**］メニューのコマンドの一部は，**表示**ツールバーからも使用できます（68 ページを参照）。

［挿入］メニューのコマンド

次の［**挿入**］メニューのコマンドを使用すると，さまざまな種類のコンポーネントおよび関数ライブラリのステップを挿入できます。

	コマンド	ショートカット・キー	機能
	［チェックポイント］＞［標準チェックポイント］	F12	［チェックポイントのプロパティ］ダイアログ・ボックスを開き，オブジェクト
	［出力値］＞［標準出力値］	CTRL+F12	［出力値のプロパティ］ダイアログ・ボックスを開き，オブジェクト
	［ステップ ジェネレータ］	F7	ステップ・ジェネレータを開く（テストと関数ライブラリでのみ使用可能）
	［関数定義ジェネレータ］		関数定義ジェネレータを開く。（テストと関数ライブラリでのみ使用可能）
	［同期化ポイント］		テストに同期化ポイントを挿入し，オブジェクト・プロパティ値が達成されるまで（またはタイムアウトするまで）テストを停止するように QuickTest に指示する。（テストでのみ利用できます）。
	［ステップの新規作成］	F8 キー，INSERT キー	キーワード・ビューに新規ステップを挿入する。
	［ブロックの後の新規ステップ］	SHIFT+F8	キーワード・ビューで，条件ブロックまたはループ・ブロックの後に新規ステップを挿入する。（テストでのみ利用できます）。
	［操作］		コンポーネントに操作（関数）ステップを挿入する。

	コマンド	ショートカット・キー	機能
	[コメント]		キーワード・ビューでコメント・ステップを挿入する。
	[レポート]		キーワード・ビューでレポート・ステップを挿入し，テスト結果にイベントを報告するように QuickTest に指示する。（テストでのみ利用できます）。
	[条件ステートメント]		選択に応じて，**If...Then**，**ElseIf...Then**，または **Else** ステートメントを挿入する。（テストと関数ライブラリでのみ使用可能）
	[ループ ステートメント]		選択に応じて，**While...Wend**，**For...Next**，**Do...While**，または **Do...Until** ステートメントを挿入する。（テストと関数ライブラリでのみ使用可能）
	[新規アクションの呼び出し]		新規アクションを作成し，指定した場所に挿入する。（テストでのみ利用できます）。
	[アクションのコピーの呼び出し]		既存のアクションの編集可能なコピーに対する呼び出しを挿入する。（テストでのみ利用できます）。
	[既存アクションの呼び出し]		再利用可能な既存のアクションに対する呼び出しを挿入する。（テストでのみ利用できます）。
	[WinRunner の呼び出し]		WinRunner テストまたはユーザ定義関数の呼び出しを挿入する （QuickTest コンピュータに WinRunner がインストールされる場合のみ使用可能）。

	コマンド	ショートカット・キー	機能
	［トランザクションの開始］		StartTransaction ステップをテストに挿入し，計測するトランザクションの開始を記録する （LoadRunner または Business Availability Center で使用されるトランザクションがテストに含まれている場合のみ使用できます）。
	［トランザクションの終了］		EndTransaction ステップをテストに挿入し，計測するトランザクションの終了を記録する （LoadRunner または Business Availability Center で使用されるトランザクションがテストに含まれている場合のみ使用できます）。

［**挿入**］メニューのコマンドの一部は，**挿入**ツールバーからも使用できます（67 ページを参照）。

［オートメーション］メニューのコマンド

次の［**オートメーション**］メニューのコマンドを使用すると，記録および実行セッションを管理できます。

	コマンド	ショートカット・キー	機能
	［記録］	F3	記録セッションを開始する。
	［実行］	F5	実行セッションを最初から，またはセッションが一時停止された行から開始する。

	コマンド	ショートカット・キー	機能
	[停止]	F4 （ショートカット・キーまたはキーの組み合わせも定義できます。638 ページ「テストの実行オプションの設定」を参照してください)。	記録または実行セッションを停止する。
	[現在のアクションの実行]		アクティブなアクションだけを実行する。(テストでのみ利用できます)。
	[ステップから実行]	CTRL+F5	選択したステップから実行セッションを開始する。
	[メンテスナンス実行モード]		アプリケーションでオブジェクトが見つからなかったために失敗したステップに対して［メンテナンス実行モード］ウィザードが開いている間に実行セッションを開始する（該当する場合)。
	[更新モード]		実行セッションを開始して，テスト・オブジェクトの説明およびほかのオプションを更新する（該当する場合)。
	アナログ記録	SHIFT+ALT+F3	**アナログ記録**モードで記録を開始する （テストでのみ利用できます)。
	[低レベル記録]	CTRL+SHIFT+F3	**低レベル記録**モードで記録を開始する （テストでのみ利用できます)。
	[記録と実行環境設定]		［記録と実行環境設定］ダイアログ・ボックスを開き，テストの記録と実行に関してブラウザの設定を定義できる （テストでのみ利用できます)。

	コマンド	ショートカット・キー	機能
	[プロセス ガイダンス リスト]		現在のドキュメントの種類および現在ロードされている QuickTest アドインで使用できるプロセスを表示して，開くことができる。
	[結果]		[テスト結果] ビューアを開いて，コンポーネントの実行セッションの結果を表示できる （テストでのみ利用できます)。

[**オートメーション**] メニューのコマンドの一部は，**オートメーション**・ツールバーからも使用できます（67 ページを参照してください)。

［リソース］メニューのコマンド

次の［**リソース**］メニューのコマンドを使用すると，オブジェクト・リポジトリなどのリソースを管理できます。

	コマンド	ショートカット・キー	機能
	［オブジェクト リポジトリ］	CTRL+R	［オブジェクト リポジトリ］ウィンドウを開き，現在のテストまたはコンポーネントのすべてのオブジェクトが含まれているツリーを表示する。
	［オブジェクト リポジトリ マネージャ］		［オブジェクト リポジトリ マネージャ］ダイアログ・ボックスを開き，複数の共有オブジェクト・リポジトリを開いて変更できる。
	［リポジトリの関連付け］		［リポジトリの関連付け］ダイアログ・ボックスを開き，テストのオブジェクト・リポジトリの関連付けを管理できる。
	［共有オブジェクト リポジトリ パラメータの割り当て］		［リポジトリ パラメータの割り当て］ダイアログ・ボックスを開き，必要に応じてリポジトリ・パラメータをマッピングできる。
	［回復シナリオ マネージャ］		［回復シナリオ マネージャ］ダイアログ・ボックスを開く。
	［関連付けのある関数ライブラリ］		アクティブなドキュメントに関連付けられている関数ライブラリを表示して，開くことができる。

［**オブジェクト リポジトリ**］メニューのコマンドは**オートメーション・**ツールバーからも使用できます（66 ページを参照）。

［デバッグ］メニューのコマンド

次の［**デバッグ**］メニューのコマンドを使用すると，コンポーネントおよび関連付けられている任意の関数ライブラリのステップをデバッグできます。

	コマンド	ショートカット・キー	機能
	［一時停止］		デバッグ・セッションを停止する。
	［ステップ イントゥ］	F11	スクリプトの現在のステップだけを実行する。現在の行がメソッドを呼び出した場合，そのメソッドがビューに表示されるが，実行されない。
	［ステップ オーバー］	F10	スクリプトの現在のステップだけを実行する。現在の行がメソッドを呼び出した場合，そのメソッド全体が実行されるが，ビューには表示されない。
	［ステップ アウト］	SHIFT+F11	メソッドの終了まで実行してから実行セッションを一時停止する （［**ステップ イントゥ**］を使ってメソッドを実行した後にのみ使用可能)。
	［ステップまで実行］	CTRL+F10	現在のステップまで実行する。
	［ステップからデバッグ］		コンポーネントの開始位置ではなく，選択したステップから実行する。
	［ウォッチに追加］	CTRL+T	選択した項目を［ウォッチ］タブに追加する。
	［ブレークポイントの設定 / 解除］	F9	コンポーネントでブレークポイントを設定または削除する。
	［ブレークポイントの有効化 / 無効化］	CTRL+F9	コンポーネントのブレークポイントを有効化または無効化する。
	［すべてのブレークポイントを削除］	CTRL+SHIFT+F9	コンポーネントのすべてのブレークポイントを削除する。
	［すべてのブレークポイントの有効化 / 無効化］		コンポーネントのすべてのブレークポイントを有効化または無効化する。

［**デバッグ**］コマンドの一部は，**デバッグ**・ツールバーからも使用できます(67 ページを参照してください)。

［ツール］メニューのコマンド

次の［**ツール**］メニューのコマンドを実行できます。

	コマンド	ショートカット・キー	機能
	オプション		［オプション］ダイアログ・ボックスを開き，グローバル・テスト・オプションを変更できる。
	［表示オプション］		［エディタ オプション］ダイアログ・ボックスを開き，テストおよび関数ライブラリをエキスパート・ビューおよび［関数ライブラリ］ウィンドウにどのように表示するかをカスタマイズできる。
	［構文チェック］	CTRL+F7	アクティブなドキュメントの構文をチェックする。（テストと関数ライブラリでのみ使用可能）
	［オブジェクトの認識］		［オブジェクトの認識］ダイアログ・ボックスを開き，QuickTest によって特定のテスト・オブジェクトを識別する方法を指定できる。
	［オブジェクト スパイ］		［オブジェクト スパイ］ダイアログ・ボックスを開き，開いているアプリケーションでオブジェクトのネイティブなプロパティと操作に加えて，テスト・オブジェクト階層，認識プロパティ，および QuickTest でそのオブジェクトを示すのに使用するテスト・オブジェクト操作を表示できる。
	［Web イベント記録の設定］		［Web イベント記録の設定］ダイアログ・ボックスを開き，記録設定レベルを指定できる （テストでのみ利用できます)。

	コマンド	ショートカット・キー	機能
	［データ ドライバ］		［データ ドライバ］ダイアログ・ボックスを開き，アクションの標準の定数一覧を表示する （テストでのみ利用できます)。
	［ActiveScreen の変更］		以前に記録された ActiveScreen を，選択した ActiveScreen に置き換える。（テストでのみ利用できます)。
	［仮想オブジェクト］＞［新規仮想オブジェクト］		仮想オブジェクト・ウィザードを開き，アプリケーションの領域を標準のテスト・オブジェクトと認識するように QuickTest を設定できる。（テストでのみ利用できます)。
	［仮想オブジェクト］＞［仮想オブジェクト マネージャ］		仮想オブジェクト・マネージャを開き，コンピュータで定義されているすべての仮想オブジェクト・コレクションを管理できる。（テストでのみ利用できます)。
	［カスタマイズ］		［カスタマイズ］ダイアログ・ボックスを開く。ここでは，ツールバーとメニューをカスタマイズしたり，新しいメニューを作成できる。

［**ツール**］メニューのコマンドの一部は，**ツール**・ツールバーからも使用できます（68 ページを参照)。

［ウィンドウ］メニューのコマンド

次の［**ウィンドウ**］メニューのコマンドを実行できます。

コマンド	機能
［重ねて表示］	開いているドキュメントを重ねて表示する。
［上下に並べて表示］	開いているドキュメントを縦に重ねて表示する。
［左右に並べて表示］	開いているドキュメントを横に並べて表示する。
［全関数ライブラリを閉じる］	開いているすべての関数ライブラリを閉じる。
［ファイル セクションを開く］	現在 QuickTest セッションで開かれているドキュメントを表示する。
［ウィンドウ］	［ウィンドウ］ダイアログ・ボックスを開き，開いているドキュメント・ウィンドウを管理できる。

［ヘルプ］メニューのコマンド

次の［**ヘルプ**］メニューのコマンドを実行できます。

コマンド	ショートカット・キー	機能
QuickTest Professional ［ヘルプ］	F1	QuickTest Professional のヘルプを開く。
［印刷用ドキュメント］		QuickTest の印刷用ドキュメントにリンクしているページを Adobe Acrobat Reader（PDF）形式で開く。
QuickTest Professional ［チュートリアル］		QuickTest Professional チュートリアルを開きます。このチュートリアルには，QuickTest の基本技術，およびアプリケーションのテストを開始する方法が説明されています。
［新機能］		QuickTest Professional ヘルプで新機能情報を開く。

コマンド	ショートカット・キー	機能
[製品の機能ムービー]		QuickTest のさまざまな機能を紹介するムービーを表示できる。
[トラブルシューティング & ナレッジ ベース]		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領域を開く。この領域では，製品固有の技術情報の記事など，いくつかの自助トラブルシューティング・オプションを選択できる （ログインが必要)。 URL は次のとおり。 http://support.openview.hp.com/troubleshooting.jsp
[HP ソフトウェア サポート]		HP ソフトウェア・サポート Web サイトを開く。このサイトでは，HP サポートの技術情報の閲覧や記事の投稿ができる。また，ユーザ・ディスカッション・フォーラムへの投稿や検索，サポート依頼の送信，パッチや更新されたドキュメントのダウンロードなども行える。 URL は次のとおり。 http://support.openview.hp.com
[フィードバックの送信]		HP の Web サイト，「QuickTest Professional Send Feedback and Win」を開く。このサイトでは，QuickTest に関する調査に回答して，特別な賞品の抽選に応募できる。 URL は次のとおり。 http://www.hpqtp.com
[更新の確認]		QuickTest Professional で使用できるアップデートをオンラインで確認する。ダウンロードしてインストール（任意）する更新を選択できる。

コマンド	ショートカット・キー	機能
HP QuickTest Professional［**ソフトウェア Web ページ**］		標準の Web ブラウザを使用して，HP の Web サイト内にある HP QuickTest Professional ソフトウェアの Web ページにアクセスする。このサイトでは，QuickTest に関する情報の概要，データシート，デモ，およびホワイト・ペーパーを提供するだけでなく，ほかの技術リソースにもアクセスできる。 URL は次のとおり。 https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-127-24^1352_4000_100__
［**QuickTest Professional のバージョン情報**］		インストールされている QuickTest Professional のバージョンに関する情報を表示する。

［データ テーブル］メニューのコマンド

データ・テーブルで 1 つ以上のセルを選択した場合に，データ・テーブルのセルを右クリックするか，対応するショートカット・キーを押すと，次の［**データ テーブル**］メニューのコマンドを実行できます。このセクションはテストにのみ関係しています。

コマンド	ショートカット・キー	機能
［**編集**］＞［**切り取り**］	CTRL+X	テーブルの選択範囲を切り取り，クリップボードに格納する。
［**編集**］＞［**コピー**］	CTRL+C	テーブルの選択範囲をコピーし，クリップボードに格納する。
［**編集**］＞［**値の貼り付け**］	CTRL+V	クリップボードの内容を，現在のテーブル選択範囲に貼り付ける。
［**編集**］＞［**クリア**］＞［**内容**］	CTRL+DEL	現在の選択範囲から内容を消去する。

コマンド	ショートカット・キー	機能
［**編集**］＞［**挿入**］	CTRL+I	現在選択している位置に空のセルを挿入する。挿入されたセルの隣のセルは，新しいセルの分だけ位置がずれる。
［**編集**］＞［**削除**］	CTRL+K	現在の選択範囲を削除する。削除されたセルの隣のセルは，消失したセルのスペースを埋めるために位置がずれる。
［**編集**］＞［**右方向へコピー**］	CTRL+R	選択範囲内の一番左にあるセルのデータを，その範囲内でそのセルの右にあるすべてのセルにコピーする。
［**編集**］＞［**縦方向にコピー**］	CTRL+D	選択された範囲内で一番上にあるセルのデータを，その範囲内でそのセルの下にあるすべてのセルにコピーする。
［**編集**］＞［**検索**］	CTRL+F	指定されたテキストを含むセルを検索する。テーブルの行ごとまたは列ごとの検索，大文字と小文字を区別した検索，セル全体が一致するものに限定した検索できる。
［**編集**］＞［**置換**］	CTRL+H	指定されたテキストを含むセルを見つけ，別のテキストに置き換える。テーブルの行ごとまたは列ごとの検索，大文字と小文字を区別した検索，セル全体が一致するものに限定した検索ができる。すべて置換することも可能。
［**データ**］＞［**再計算**］	F9	データ・テーブルの選択データを再計算する。
データ・テーブルのシート間の切り替え	CTRL+PAGE UP/ PAGE DOWN	データ・テーブルにフォーカスがあるときに，データ・テーブルのシートを切り替える。

その他の QuickTest コマンド

ショートカット・キーを使用して，次の特別なオプションを実行できます。

オプション	ショートカット・キー	機能
キーワード ビューとエキスパート ビューの切り替え	CTRL+PAGE UP/PAGE DOWN	キーワード・ビューとエキスパート・ビューを切り替える。
開いているドキュメント間の切り替え	CTRL+TAB	開いているほかのドキュメントのタイプに表示を切り替える。
ショートカット・メニューを開く	SHIFT+F10，または，アプリケーション・キー（）を押す **(Microsoft ナチュラル・キーボードのみ)**。	データ・テーブルで選択されているステップ・データ・セルに対応するショートカット・メニューを開く。
ツリーをすべて展開	* **（テンキーを使用）**	キーワード・ビューのすべてのツリーを展開する。
サブツリーを展開	+ **（テンキーを使用）**	キーワード・ビューで選択した項目のツリーと，それ以下のすべてのサブツリーを展開する。
サブツリーを折りたたむ	- **（テンキーを使用）**	キーワード・ビューで選択した項目のツリーと，それ以下のすべてのサブツリーを折りたたむ。
[**項目**]または[**操作**]リストを開く	SHIFT+F4 または SPACE（キーワード・ビューで[**項目**]または[**操作**]カラムが選択されている場合)。	[**項目**]または[**操作**]カラムが選択されているときに，キーワード・ビューで[**項目**]または[**操作**]リストが開く。

QuickTest Professional プログラム・フォルダの参照

QuickTest Professional セットアップ・プロセスが完了すると，QuickTest Professional プログラム・フォルダ（[**スタート**] > [**プログラム**] > [**QuickTest Professional**]）に次の項目が追加されます。

注：現在のバージョンをインストールする前に QuickTest Professional の旧バージョンをアンインストールした場合には，QuickTest Professional プログラム・フォルダに余計な（無効の）項目が追加されることがあります。さらに，QuickTest Professional のアドインまたは拡張 SDK をインストールした場合は，それらにのみ関連する項目がプログラム・フォルダに追加されることがあります。

- [**Documentation**]：よく使用されるドキュメントへの下記のリンクを提供します。
  - [**Printer-Friendly Documentation**]：QuickTest の印刷用ドキュメントにリンクしているページを Adobe Acrobat Reader（PDF）形式で開きます。
  - [**QuickTest Automation Reference**]：QuickTest Professional の『**Automation Object Model Reference**』を開きます。オートメーション・オブジェクト・モデルは，QuickTest の機能と設定を制御することを可能にするオブジェクト，メソッド，プロパティを提供することによって，テスト管理の自動化を支援します。『**Automation Object Model Reference**』には，構文，機能説明，およびオブジェクト，メソッド，プロパティの使用例が記載されています。また，QuickTest の自動スクリプトを記述する際の詳しい概要も含まれます。
  - [**QuickTest Professional Code Samples Plus**]：QuickTest Professional コード・サンプル・プラス・ヘルプを開きます。このヘルプでは，関数ライブラリのサンプル，コード，および SDK のサンプルを説明とともに提供されます。
  - [**QuickTest Professional Help**]：『**HP QuickTest Professional ユーザーズ・ガイド**』，『**HP QuickTest Professional for Business Process Testing ユーザーズ・ガイド**』，『**HP QuickTest Professional アドイン・ガイド**』，『**HP QuickTest Professional Object Model Reference**』（英語版）（インストールされているアドインに関連するセクションを含む），『**QuickTest Advanced References**』（オートメーション API および XML スキーマ・リファレンス），

および『**Microsoft VBScript Reference**』が含まれている包括的なヘルプ・ファイルが開きます。

- ［**Tutorial**］：QuickTest Professional チュートリアルを開きます。このチュートリアルには，QuickTest の基本技術，およびアプリケーションのテストを開始する方法が説明されています。

- ［**Extensibility**］：QuickTest Professional 10.00 で利用できるアドイン拡張 SDK に関するヘルプへのリンクを提供します。拡張 SDK をインストールすると，このプログラム・フォルダに追加の項目が含まれていることもあります。
- ［**Sample Applications**］：QuickTest でテストの練習に使用できる下記のサンプル・アプリケーションへのリンクが収められています。
  - ［**Flight**］：サンプルのフライト予約 Windows アプリケーションが開きます。このアプリケーションにアクセスするには，任意のユーザ名とパスワード（**mercury**）を入力します。
  - ［**Mercury Tours Web Site**］：サンプルのフライト予約 Web アプリケーションが開きます。この Web アプリケーションは，QuickTest のチュートリアルで使用します。詳細については，『**HP QuickTest Professional チュートリアル**』を参照してください。
- ［**Tools**］：テスト・プロセスを支援する下記のユーティリティとツールが収められています。

注：インストールする QuickTest アドインに応じて，ツールが追加されることがあります。

  - ［**Additional Installation Requirements**］：［追加のインストール要件］ダイアログ・ボックスを開き，QuickTest を使用するためにインストールまたは設定する必要があるソフトウェアを表示します。
  - ［**HP Micro Player**］：HP Micro Player を開き，QuickTest を開かずに実行セッションのキャプチャされたムービーを表示できます。詳細については，［HP Micro Playerr］ウィンドウで［**ヘルプ**］ボタンをクリックしてください。
  - ［**License Validation Utility**］：ライセンス検証ユーティリティを開き，ライセンス情報を取得および検証できます。詳細については，［ライセンス検証ユーティリティ］ウィンドウの［**ヘルプ**］ボタンをクリックしてください。

- ［**Password Encoder**］:［パスワード エンコーダ］ダイアログ・ボックスが開きます。このダイアログ・ボックスでは，パスワードを暗号化できます。生成された文字列は，メソッドの引数またはデータ・テーブルのパラメータ値として使用できます（テストのみ）。詳細については，536 ページ「で使用するパスワードを暗号化できます。メソッド引数への暗号化されたパスワードの挿入」を参照してください。
- ［[**QuickTest Script Editor**］: QuickTest スクリプト・エディタを開き，複数のテストおよび関数ライブラリのスクリプトを同時に開いて変更できます。詳細については，『**HP QuickTest Professional ユーザーズ・ガイド**』を参照してください。
- ［**Register New Browser Control**］:［新規ブラウザ・コントロールの登録］ユーティリティを開き，テストを記録または実行するときに QuickTest Professional で Web オブジェクトを認識できるように，ブラウザ・コントロール・アプリケーションを登録できます。詳細については，『**HP QuickTest Professional アドイン・ガイド**』でブラウザ・コントロールの登録に関する項を参照してください。
- ［**Remote Agent**］: QuickTest リモート・エージェントをアクティブにして，コンポーネントが，Quality Center などのリモート・アプリケーションによって実行されたときの QuickTest の動作を設定できます。詳細については，『**HP QuickTest Professional ユーザーズ・ガイド**』を参照してください。
- ［**Save and Restore Settings**］:［設定の保存と復元］ダイアログ・ボックスが開きます。このダイアログ・ボックスでは，QuickTest 9.2 以前のバージョンをアンインストールする前に既存の設定を保存して，新バージョンをインストールした後に復元できます。詳細については，『**HP QuickTest Professional ユーザーズ・ガイド**』を参照してください。
- ［**Silent Test Runner**］:（テストにのみ関連）。［サイレント テスト ランナー］ダイアログ・ボックスが開きます。このダイアログ・ボックスでは，LoadRunner および Business Availability Center から実行するのと同じように，QuickTest テストを実行できます。詳細については，『**HP QuickTest Professional ユーザーズ・ガイド**』を参照してください。
- ［**Test Batch Runner**］:（テストにのみ関連）。［Test Batch Runner］ダイアログ・ボックスが開きます。このダイアログ・ボックスでは，連続して数回テストが実行されるように QuickTest を設定できます。詳細については，『**HP QuickTest Professional ユーザーズ・ガイド**』を参照してください。
- ［**Test Results Deletion Tool**］:［テスト結果削除ツール］ダイアログ・ボックスが開きます。このダイアログ・ボックスでは，定義された特定の条件に

基づいて，不要な結果をシステムから削除できます。詳細については，710 ページ「テスト結果削除ツールを使った結果の削除」を参照してください。

➤［**Check for Updates**］：QuickTest Professional で使用できるアップデートをオンラインで確認する。ダウンロードしてインストール（任意）する更新を選択できます。

➤［**QuickTest Professional**］：QuickTest Professional アプリケーションが起動します。

➤［**Readme**］：『**HP QuickTest Professional 最初にお読みください**』を開きます。ここでは，QuickTest Professional および QuickTest Professional アドインに関する最新のニュースおよび情報が提供されます。

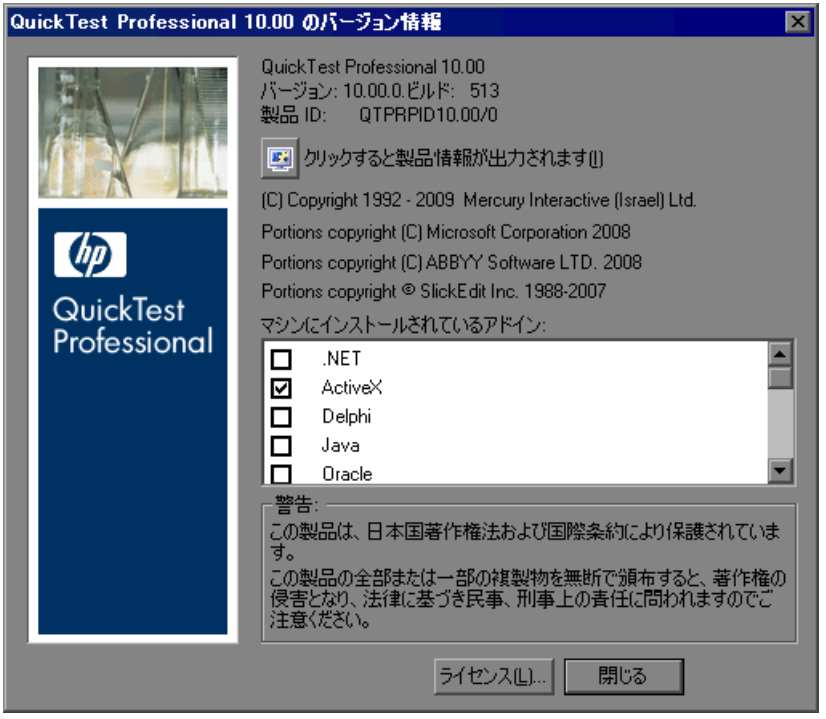
➤［**Test Results Viewer**］：［テスト結果］ウィンドウが開きます。このウィンドウでは，コンポーネントまたはビジネス・プロセス・テストを選択して，実行セッション中に実行されたステップに関する情報を表示できます。詳細については，681 ページ「［テスト結果］ウィンドウ」を参照してください。

製品情報の表示

コンピュータにインストールされている QuickTest アドイン，ホットフィックス，およびパッチに関する情報，コンピュータに関するその他の基本情報を表示できます。この情報はトラブルシューティングや HP ソフトウェア・サポートと協力するときに役立ちます。

製品情報を表示するには，次の手順を実行します。

1 QuickTest で，［**ヘルプ**］＞［**QuickTest Professional のバージョン情報**］を選択します。［QuickTest Professional のバージョン情報］ダイアログ・ボックスが開きます。



［QuickTest Professional のバージョン情報］ウィンドウには，次の情報が表示されます。

- コンピュータにインストールされている QuickTest のバージョン，ビルド番号，製品 ID 番号。
- コンピュータにインストールされている QuickTest アドインのリスト。アドイン名の横にあるチェック・マークは，そのアドインが現在ロードされていることを示します。QuickTest アドインの詳細については，『**HP QuickTest Professional アドイン・ガイド**』を参照してください。

ヒント：コンピュータにインストールされている QuickTest Professional ライセンスの詳細を表示する場合，またはライセンスを変更する場合は，［**ライセンス**］ボタンをクリックします。詳細については，『**HP QuickTest Professional インストール・ガイド**』を参照してください。

---

2 コンピュータにインストールされている QuickTest Professional 製品の詳細情報を表示するには，［**製品情報**］ボタンをクリックします。［製品情報］ウィンドウが開きます。

製品情報

製品名:	QuickTest Professional
製品バージョン:	10.00
製品 ID:	QTPRPID10.00/01
製品ビルド:	513
オペレーティング システム:	Service Pack 2 (Build 3790)
Internet Explorer のバージョン:	6.0.3790.3959
Quality Center 接続機能:	10.0.0.1946

アドイン情報:

名前
.NET
ActiveX
Delphi
Java
Oracle
PeopleSoft
PowerBuilder
SAP
Siebel
Stingray
Terminal Emulators
Visual Basic
VisualAge Smalltalk
Web
Web Services
WPF

ホットフィックスおよびパッチの情報:

名前
HP QuickTest Professional 10.0 User Interface Pack

注: インストール済みのホットフィックスおよびパッチの Readme ファイルは、すべて C:\Program Files\HP\QuickTest Professional\HotfixReadmes に格納されています。

© Copyright 1992 - 2009 Mercury Interactive (Israel) Ltd.

［製品情報］ウィンドウには，次の情報が表示されます。

- コンピュータにインストールされている QuickTest Professional のバージョン，製品 ID 番号，ビルド番号。
- **オペレーティング システム**：コンピュータにインストールされているオペレーティング・システムのバージョン。
- **Internet Explorer のバージョン**：コンピュータにインストールされている Microsoft Internet Explorer のバージョン。
- **Quality Center 接続機能**：コンピュータにインストールされている Quality Center 接続アドインのバージョン。
- **アドイン情報**：コンピュータにインストールされている QuickTest アドイン。
- **ホットフィックスおよびパッチ情報**：コンピュータにインストールされている QuickTest のホット・フィックスまたはパッチの名前，およびその readme ファイルへのリンク。

第 II 部

テスト・オブジェクトおよびオブジェクト・リポジトリを使用した作業

第 3 章

テスト・オブジェクト・モデルについて

本章では，QuickTest でアプリケーション内のオブジェクトの学習方法および識別方法について説明します。また，**テスト・オブジェクト**と**実行環境オブジェクト**の概念，オブジェクト・リポジトリのタイプについて説明し，オブジェクトで使用可能なメソッドとその構文を表示する方法について取り上げます。この情報を活用することで，コンポーネントの操作を作成するときに，［エキスパート ビュー］でスクリプトにステートメントを追加したり，または関数でテスト・オブジェクトおよびメソッドを使用したりできます。

本章の内容

- テスト・オブジェクト・モデルについて（103 ページ）
- テスト・オブジェクト・モデルの概念の適用（107 ページ）
- オブジェクト・リポジトリの種類について（112 ページ）
- オブジェクト・スパイを使用したオブジェクト・プロパティと操作の表示（119 ページ）
- ［オブジェクト スパイ］ダイアログ・ボックス（122 ページ）

テスト・オブジェクト・モデルについて

QuickTest は，テスト・オブジェクトとその期待プロパティおよび期待値を学習，識別することによって，動的に変化するアプリケーションをテストします。これを行うために，QuickTest は，人間が写真を見てその詳細を記憶するのとほぼ同じ方法で，アプリケーションの各オブジェクトを分析します。

次の各項では，テスト・オブジェクト・モデルに関する概念を紹介し，収集した情報を基に QuickTest によってアプリケーションがどのようにテストされるかを説明します。

QuickTest によるオブジェクトの学習方法について

QuickTest は人と同じように学習します。たとえば，ある実験で，Alex という被験者にピクニックの様子を写した写真を 2 ～ 3 秒間見て，その写真に写っているある何かを覚えてもらうと予告したとします。そして，1 週間後に同じ写真か別の似たような写真を見て，その何かを識別してもらうとします。

Alex は写真を見る前に，テストに備えて，テスト担当者に指示される記憶対象のどのような特徴を覚えるかを考えます。もちろん，Alex はそれが人，物，動物，植物のどれであるかは，特に意識することもなく覚えるでしょう。記憶する対象が人ならば，性別，皮膚の色，そして年齢を覚えるつもりでいます。そして，動物ならば，その動物の種類や色などといったことを覚えるつもりでいます。

テスト担当者は，Alex に写真を見せ，地面に敷いたシートの上に座っている 3 人の子供のうちの 1 人を指差します。Alex は女の子が白人で 8 歳ほどだと記憶します。しかし，写真のほかの部分を見て，その記述に当てはまる子供がもう 1 人いることに気付きます。そこで，覚える予定だった特徴に加え，その女の子の髪が茶色で長いことも覚えます。

これで，写真の中で Alex が覚えた特徴と一致するのは 1 人だけになりますから，1 週間後に見せられる写真が少し違っていても，かなり確実にその女の子を見分けられるでしょう。

それでもまだ少し写真を見る時間が残っているので，念のために，ほかの子供達とのもっと微妙な違いを見つけられるかどうか試みます。

もし，写真の中の似ている 2 人の子供が見分けのつかないような双子ならば，Alex はもっと一時的でしかない特徴も覚えようとするかもしれません。たとえば，シートの上で座っている位置です。別の見せられる写真でも子供達の位置関係が同じでありさえすれば，Alex は指定された女の子を見分けることができるでしょう。

QuickTest では，オブジェクトを学習するときに，これに非常に似た方法を使います。

まず，学習対象のオブジェクトを「見て」，これを「**テスト・オブジェクト**」として保存し，そのオブジェクトに合うテスト・オブジェクト・クラスを決定します。ちょうど Alex が，項目が人間，動物，植物，または無生物のどれであるかをすぐに検査したようにです。QuickTest は，テスト・オブジェクトを，たとえば標準 Windows ダイアログ・ボックス（Dialog），Web ボタン

（WebButton），あるいは Visual Basic スクロール・バー・オブジェクト（VbScrollBar）のようにクラス分けします。

次に，QuickTest では，テスト・オブジェクトの**認識プロパティ**が「考慮」されます。QuickTest には，テスト・オブジェクト・クラスごとに必ず学習する**必須**プロパティのリストがあります。これは Alex が写真を見る前に覚えようと考えていた特徴のリストに似ています。オブジェクトを学習するとき，QuickTest は必ずこれらの標準プロパティ値を学習し，ページ，ダイアログ・ボックス，その他の親オブジェクト上にある残りのオブジェクトを「見て」，その**記述**がオブジェクトを一意に識別するのに十分であるかどうかを確認します。十分でない場合，QuickTest は一意の記述ができるまで，記述に**補足**プロパティを 1 つずつ追加していきます。これは Alex がリストに髪の長さと色という特徴を追加したのと似ています。補足プロパティがない場合や，あっても一意の記述を作成するには足りない場合，QuickTest は，オブジェクトのページやソース・コードでの位置のような，特殊な**序数識別子**を追加して，一意の記述を作成します。これは，写真に写っている子供のうちの 2 人が，見分けのつかない双子であったならば，子供の座っているシート上での位置を覚えることとちょうど同じです。

QuickTest が実行セッション中にオブジェクトを識別する方法について

QuickTest は，非常に人間的なテクニックを使って実行セッション中にオブジェクトを識別します。

実験の続きとして，新しいけれども似たような環境で，最初に識別したものと同じ「項目」を識別するように Alex が指示されたとします。

最初に，元の写真を見せます。Alex は，記憶するように指示された，8 歳くらいの長い茶色の髪の同じ白人の女の子を探し，すぐに見つけます。2 番目の写真では，子供達は遊具を使って遊んでいますが，Alex はやはり，同じ基準を使ってその女の子を簡単に識別できます。

同様に，実行セッション中，QuickTest は以前に学習したテスト・オブジェクトの記述と正確に一致する**実行環境オブジェクト**を探します。オブジェクトの学習中に一意の記述を作成するのに使われた必須およびすべての補足プロパティが完全一致することが期待されます。アプリケーションのオブジェクトが大きく変更されていなければ，学習した記述は QuickTest がそのオブジェクトを識別するのにほとんど常に十分です。これは，ほとんどのオブジェクトに当てはまりますが，アプリケーションには以降の実行セッションで識別が困難なオブジェクトが含まれている可能性もあります。

Alex の実験の最終段階を考えてみましょう。この段階では，テスト担当者は Alex に，同じ場所で同じ家族を写したもう 1 枚の写真を見せますが，子供達は成長しており，運動場ではもっと多くの子供達が遊んでいます。Alex はまず，別の写真で女の子を見分けるために使ったのと同じ特徴を持った女の子を探します（テスト・オブジェクト）が，写真の中には長い茶色の髪の女の子は 1 人もいません。幸運にも Alex は，前の週に最初に写真を見たときにその女の子について，ほかの情報も覚えていました。彼は，その女の子の髪が短くブロンドになっているにもかかわらず彼女を見つけ出す（実行環境オブジェクト）ことができました。

どのようにやってのけたのでしょうか。まず，見つけなければならない，自分が知っている特徴は何であるかを考えました。Alex は，探しているのが白人の女性であることに変わりはないことをわかっており，この基準に一致する人が誰も見つからなければ，彼女は写真に写っていないと判断します。

この新しい写真に写っている 4 人の白人女性に的を絞ると，今度は問題の女の子を見分けるために使ってきたほかの特徴（年齢，髪の色，髪の長さ）について考えます。同じ人物であっても，年月が経過しており，覚えているほかの特徴も変わっているかもしれないことはわかっています。

したがって，白人の女の子は誰一人として長い茶色の髪ではありませんでしたが，そうした特徴は無視し，記憶にある目と鼻を持つ女の子を探します。目の似ている少女が 2 人いますが，元の写真で見たような小さな鼻をしているのは，そのうちの 1 人だけであることに気付きます。あまり目立つ特徴ではありませんが，これらを使って少女を識別できます。

QuickTest の**スマート認識**メカニズムでは，これに非常によく似た消去法を使って，学習時の記述がもはや正確とはいえなくなっているオブジェクトを識別します。認識プロパティの値が変更されていても，QuickTest はスマート認識でオブジェクトを識別することによって，コンポーネントの再利用可能性を維持します。スマート認識の詳細については，第 6 章「オブジェクト認識の設定」を参照してください。

次に，本書では，テスト・オブジェクト，実行環境オブジェクト，オブジェクト・プロパティ，必須プロパティ，補足プロパティ，スマート認識など，ここで示した概念について説明します。これらの概念を理解することにより，アプリケーションに対して適切に設計された機能コンポーネントを作成できるようになります。

テスト・オブジェクト・モデルの概念の適用

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

テスト・オブジェクトとは，アプリケーションに含まれる実際のオブジェクトを表現するために，QuickTest によってコンポーネント内に作成されるオブジェクトです。QuickTest によってオブジェクトに関する情報が格納されます。この情報は，実行セッション中にそのオブジェクトを識別し検査するために使用されます。

実行環境オブジェクトとは，実行セッション中にメソッドの実行対象となるアプリケーションに含まれる実際のオブジェクトです。

QuickTest がアプリケーションのオブジェクトを学習すると，対応するテスト・オブジェクトを**オブジェクト・リポジトリ**（オブジェクトの格納庫）に追加します。テスト・オブジェクトは，いくつかの方法でオブジェクト・リポジトリに追加できます。たとえば，QuickTest の［ナビゲートして学習］オプションを使用したり，テスト・オブジェクトを手動で追加したり，記録中にアプリケーション上で操作を実行することができます。オブジェクト・リポジトリの詳細については，第 5 章「オブジェクト・リポジトリ内のテスト・オブジェクトの管理」，第 7 章「オブジェクト・リポジトリの管理」を参照してください。

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

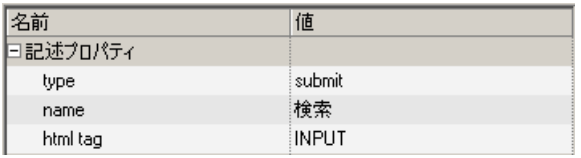
- 学習したオブジェクトを示す QuickTest テスト・オブジェクト・クラスが識別され，適切なテスト・オブジェクトが作成される。
- アプリケーションに含まれているオブジェクトのプロパティの現在値が読み取られ，**認識プロパティ**のリストと値がテスト・オブジェクトと一緒に保管される。

➤ テスト・オブジェクトに一意の名前を選択します。通常，代表的なプロパティの 1 つの値を使用します。

たとえば，次の HTML ソース・コードで［**Search**］ボタンを追加したとします。

```
<INPUT TYPE="submit" NAME="Search" VALUE="Search">
```

QuickTest ではオブジェクトが **WebButton** テスト・オブジェクトとして識別されます。オブジェクト・リポジトリで，QuickTest が Search という名前の WebButton オブジェクトを作成し，オブジェクトの一連の認識プロパティを学習して，**Search** WebButton を一意に識別する次のプロパティと値の使用を決定します。



名前	値
記述プロパティ	
type	submit
name	検索
html tag	INPUT

アプリケーションで記録することによってオブジェクトをオブジェクト・リポジトリに追加する場合は，QuickTest では，適切な QuickTest テスト・オブジェクト・メソッドを使用してオブジェクトで実行した操作が記録されます。たとえば，QuickTest によって，WebButton で **Click** メソッドが実行されたことが記録されます。

Search ボタンをクリックすることがコンポーネントの最初のステップと仮定すると，QuickTest によってステップが次のように表示されます。

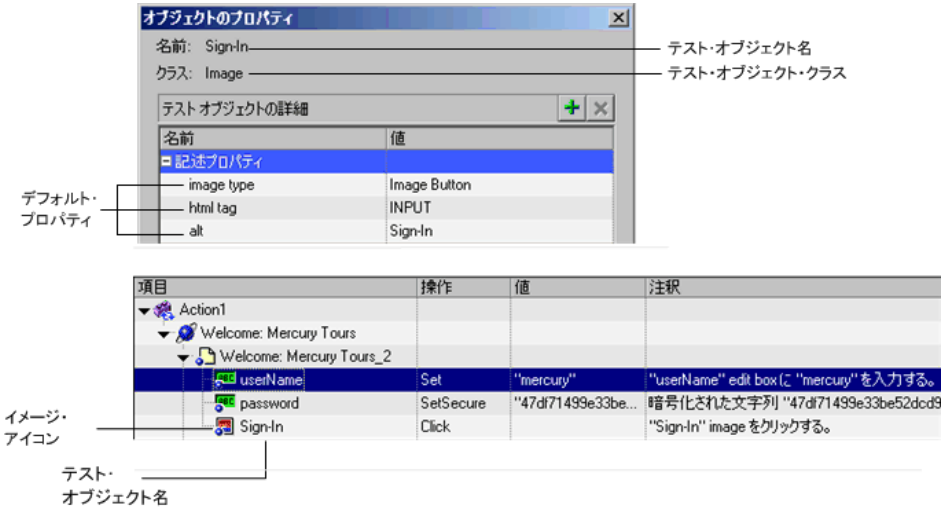
項目	操作	注釈
検索(F)	Click	"検索(F)" button をクリックする。

コンポーネントを実行すると，QuickTest により，テスト・オブジェクト・クラスと**記述**（オブジェクトを一意に識別する認識プロパティと値のセット）に基づいて，アプリケーションに含まれている各オブジェクトが識別されます。テスト・オブジェクトとそのプロパティおよび値のリストは，オブジェクト・リポジトリに格納されます。前の例では，実行セッション中，QuickTest によりオブジェクト・リポジトリ内で Search という名前を持つ WebButton オブジェクトが検索され，その記述が調べられます。次に，見つかった記述に基づいて，QuickTest によって，HTML タグが INPUT，タイプが submit，値が「Search」である WebButton オブジェクトがアプリケーションの中で検索されます。該当するオブジェクトが見つかると，そのオブジェクトに対して **Click** メソッドが実行されます。

テスト・オブジェクト記述について

各テスト・オブジェクト・クラスごとに，QuickTest はオブジェクトを学習すると，認識プロパティのセットを学習し，一意のオブジェクト記述として機能する認識プロパティのサブセットを選択します。QuickTest は次に，この記述を使用して，コンポーネントを実行するときにオブジェクトを識別します。

たとえば，標準設定の QuickTest では，学習対象となる各 Web 画像の画像タイプ（plain image や image button など），**HTML** タグ，**代替**テキストが認識されます。



親オブジェクト内のオブジェクトを一意に識別するのにこれら 3 つの必須のプロパティ値では十分ではない場合，QuickTest は一意の記述を作成するため，いくつかの補助プロパティまたは予約識別子，あるいはその両方を追加します。

コンポーネントを実行すると，学習された記述と一致するオブジェクトが QuickTest によって検索されます。記述と一致するオブジェクトが見つからなかったり，複数の一致するオブジェクトが見つかったりした場合，QuickTest ではオブジェクトを識別するため，スマート識別メカニズムが使用されることがあります。

QuickTest によってアプリケーション内のオブジェクトの記述の学習に使用される必須，補助，および序数識別子のプロパティは，ユーザが設定できます。また，スマート識別機能の有効化と設定もユーザが実施できます。詳細については，第 6 章「オブジェクト認識の設定」を参照してください。

テスト・オブジェクト，ネイティブなプロパティ，および操作について

各テスト・オブジェクトの認識プロパティ・セットは，QuickTest によって作成され，維持されます。また，各実行環境オブジェクトのネイティブなプロパティ・セットは，オブジェクト・クリエータ（Microsoft Internet Explorer オブジェクトの場合は Microsoft，Netscape Browser オブジェクトの場合は Netscape，ActiveX オブジェクトの場合は製品開発者など）によって作成され，維持されます。

同様に，テスト・オブジェクト操作は，QuickTest によって特定のテスト・オブジェクト・クラスに適用可能と認識されるメソッドまたはプロパティです。たとえば，**Click** メソッドは WebButton テスト・オブジェクトに適用できます。コンポーネントにステップを追加するときに，各テスト・オブジェクトで実行する操作を指定します。ステップを記録すると，関連する操作がオブジェクトで実行されるときに，QuickTest によって記録されます。

実行セッション中は，指定したテスト・オブジェクト操作が QuickTest によって実行環境オブジェクトで実行されます。ネイティブな操作は，オブジェクト・クリエータによって定義される，アプリケーション内のオブジェクトのメソッドです。

アプリケーションに含まれるオブジェクトのプロパティの値は，アプリケーションを起動するたび，または特定の条件に基づいて，動的に変化することがあります。認識プロパティの値を変更して，ネイティブなプロパティの値に一致させる必要があるかもしれません。認識プロパティは，コンポーネントの設計時に手作業で，あるいは実行セッション中に（関数ライブラリで定義した操

作を介して）**SetTOProperty** ステートメントを使って変更できます。また，正規表現を使って，定義した条件やパターンに基づいてプロパティの値を特定できます。オブジェクト・プロパティの変更の詳細については，第 5 章「オブジェクト・リポジトリ内のテスト・オブジェクトの管理」

コンポーネントに格納されている認識プロパティの値は，［オブジェクトのプロパティ］または［オブジェクト リポジトリ］ダイアログ・ボックスで確認したり変更できます。詳細については，176 ページ「プロパティ値の指定または変更」を参照してください。

また，デスクトップ上にある任意のオブジェクトの認識プロパティの現在値は，オブジェクト・スパイの［プロパティ］タブで確認できます。詳細については，119 ページ「オブジェクト・スパイを使用したオブジェクト・プロパティと操作の表示」を参照してください。

テスト・オブジェクト操作の構文，およびデスクトップ上にある任意のオブジェクトのネイティブな操作は，オブジェクト・スパイの［操作］タブで確認できます。詳細については，119 ページ「オブジェクト・スパイを使用したオブジェクト・プロパティと操作の表示」を参照してください。

関数ライブラリで定義された操作を使用すると，**GetTOProperty** および **SetTOProperty** ステートメントを追加して，実行セッション中にテスト・オブジェクトのプロパティ値を取得したり変更できます。実行セッション中に実行環境オブジェクトからプロパティ値を読み込むには，**GetROProperty** ステートメントを追加します。詳細については，370 ページ「認識プロパティ値の取得と設定」を参照してください。

使用可能なテスト・オブジェクト操作およびテスト・オブジェクトの認識プロパティによって必要な機能が提供されない場合は，**Object** プロパティを使って実行環境オブジェクトの内部操作およびプロパティにアクセスできます。また，オブジェクトのプロパティ**属性**を使用しても，ユーザ定義のプロパティに従って，アプリケーション内の Web オブジェクトを識別できます。詳細については，372 ページ「ネイティブ・プロパティおよび操作へのアクセス」を参照してください。

テスト・オブジェクト操作および認識プロパティの詳細については，『**HP QuickTest Professional Object Model Reference**』（英語版）を参照してください。

オブジェクト・リポジトリの種類について

オブジェクトは，2 種類のオブジェクト・リポジトリに格納できます。1 つは共有オブジェクト・リポジトリで，もう 1 つはローカル・オブジェクト・リポジトリです。**共有オブジェクト・リポジトリ**では，複数のコンポーネントから（アプリケーション領域を介して）読み取り専用モードでアクセス可能なファイルにオブジェクトが格納されます。**ローカル・オブジェクト・リポジトリ**では，ある特定のコンポーネントに関連付けられているファイルにオブジェクトが格納されるため，格納されたオブジェクトにアクセスできるのはそのコンポーネントのみです。

コンポーネントを計画し作成するときは，コンポーネント内のオブジェクトを格納する方法について検討する必要があります。各コンポーネントのオブジェクトを対応するローカル・オブジェクト・リポジトリに格納することも，コンポーネント内のオブジェクトを 1 つ以上の共有オブジェクト・リポジトリに格納することもできます。オブジェクトを共有オブジェクト・リポジトリに格納し，それらのリポジトリをコンポーネントのアプリケーション領域に関連付ければ，複数のコンポーネントでそれらのオブジェクトを使用できるようになります。各コンポーネントでは，必要に応じてローカル・オブジェクト・リポジトリのオブジェクトと共有オブジェクト・リポジトリのオブジェクトを組み合わせて使用できます。必要であれば，ローカル・オブジェクトを共有オブジェクト・リポジトリに移動することもできます。そうすることで，オブジェクトを複数の場所ではなく 1 つの共有の場所で管理できるため，保守が軽減され，コンポーネントの再利用性が向上します。詳細については，114 ページ「使用するオブジェクト・リポジトリ（ローカルか共有か）の決定」を参照してください。

QuickTest をまだ使い慣れていない場合は，ローカル・オブジェクト・リポジトリを使用することをお勧めします。これにより，対応するコンポーネントからアクセス可能なローカル・オブジェクト・リポジトリにすべてのオブジェクトが自動的に保存されるため，共有オブジェクト・リポジトリの作成，選択，変更を行わずにコンポーネントの記録と実行ができます。ローカル・オブジェクト・リポジトリ内のオブジェクトを変更した場合，その変更はほかのコンポーネントには影響しません。

テストに慣れている場合は，共有オブジェクト・リポジトリにオブジェクトを保存するのがおそらく最も効率的です。その場合，コンポーネントに含まれるオブジェクトが同じであれば，複数のコンポーネントで同じ共有オブジェクト・リポジトリを使用できます。複数のコンポーネントに適用されるオブジェクト情報が 1 か所に集められて保管されます。アプリケーション内のオブジェ

クトに変更があった場合は，この共有オブジェクト・リポジトリを使用するすべてのコンポーネントに対してオブジェクトを一元的に更新できます。

同じ名前のオブジェクトが，ローカル・オブジェクト・リポジトリと，同じコンポーネントに関連付けられた共有オブジェクト・リポジトリの両方に格納されている場合，そのコンポーネントではローカルのオブジェクト定義が使用されます。同じ名前のオブジェクトが同じコンポーネントに関連付けられた複数の共有オブジェクト・リポジトリに格納されている場合，各共有オブジェクト・リポジトリがそのコンポーネントに関連付けられた順序に従って，そのオブジェクトが最初に見つかった場所のオブジェクト定義が使用されます。共有オブジェクト・リポジトリの関連付けの詳細については，470 ページ「既存のオブジェクト・リポジトリのアプリケーション領域への関連付け」を参照してください。

ローカル・オブジェクトは，コンポーネントとともにローカルに保存され，そのコンポーネントからのみアクセスできます。共有オブジェクト・リポジトリを使用する場合は，複数のコンポーネントに対して同じオブジェクト・リポジトリを使用できます。また，各コンポーネントで複数のオブジェクト・リポジトリを使用することもできます。

既存のコンポーネントを開いて操作する場合は，コンポーネントが関連付けられたアプリケーション領域で指定したオブジェクト・リポジトリが常に使用されます。コンポーネントからアクセスしたときの共有オブジェクト・リポジトリは読み取り専用です。共有オブジェクト・リポジトリを編集するには，オブジェクト・リポジトリ・マネージャを使用します。

使用するオブジェクト・リポジトリ（ローカルか共有か）の決定

オブジェクトの保存先を選択するには，ローカル・オブジェクト・リポジトリと共有オブジェクト・リポジトリの違いを理解する必要があります。

一般に，ローカル・オブジェクト・リポジトリはシンプルなコンポーネントを作成する場合に使いやすく，特に次のような場合に使用します。

- 任意のアプリケーション，インタフェース，またはオブジェクト・セットについて，対応するコンポーネントの数が 1 つしかないか，または非常に少ない。
- オブジェクトのプロパティを頻繁に変更する可能性がない。

反対に，共有オブジェクト・リポジトリは，通常次の場合に使用します。

- （記録によってではなく）キーワード駆動の方法でコンポーネントを作成している。
- 同じアプリケーション，インタフェース，またはオブジェクト・セットの要素を検査するコンポーネントが複数ある。
- アプリケーション内のオブジェクトのプロパティを時々変更する可能性があり，オブジェクトのプロパティを頻繁に更新あるいは変更する必要がある。

ローカル・オブジェクト・リポジトリについて

ローカル・オブジェクト・リポジトリを使用する場合，QuickTest ではコンポーネントごとに個別のオブジェクト・リポジトリが使用されます（また，必要であれば 1 つ以上の共有オブジェクト・リポジトリを使用することもできます。詳細については，115 ページ「共有オブジェクト・リポジトリについて」を参照してください）。ローカル・オブジェクト・リポジトリに対しては，該当するコンポーネント内からすべての編集操作が可能です。

ローカル・オブジェクト・リポジトリを使って作業する場合は，次の点に注意してください。

- QuickTest は，コンポーネントごとに新規（空）のオブジェクト・リポジトリを作成します。
- （ローカル・オブジェクト・リポジトリにオブジェクトを追加するか，アプリケーション内のオブジェクトに対する操作を記録することで）QuickTest が新しいオブジェクトを学習すると，それらのオブジェクトに関する情報が対応するローカル・オブジェクト・リポジトリに自動的に格納されます（関連する共有

オブジェクト・リポジトリにそのテスト・オブジェクトがまだ存在しない場合)。

1 つ以上の共有オブジェクト・リポジトリがすでにコンポーネントに関連付けられている場合でも，QuickTest は，すべての新しいオブジェクトをローカル・オブジェクト・リポジトリに追加します（これは，関連付けられている共有オブジェクト・リポジトリの 1 つに同じ名前記述を持つオブジェクトが存在しないことが前提です)。

- 子オブジェクトがローカル・オブジェクト・リポジトリに追加され，その親が共有オブジェクト・リポジトリ内にある場合は，親がローカル・オブジェクト・リポジトリに自動的に追加されます
- 新しいコンポーネントを作成するたびに，QuickTest は対応する新しいローカル・オブジェクト・リポジトリを作成し，テスト・オブジェクトを学習するとリポジトリに追加します。
- QuickTest がアプリケーション内の同じオブジェクトを対象に 2 つの異なるコンポーネントを学習すると，そのテスト・オブジェクトはそれぞれのローカル・オブジェクト・リポジトリに個別のテスト・オブジェクトとして格納されます。
- コンポーネントを保存すると，ローカル・オブジェクト・リポジトリがテストとともに自動的に保存されます。このローカル・オブジェクト・リポジトリには，(共有オブジェクト・リポジトリと異なり）独立のファイルとしてアクセスできません。

共有オブジェクト・リポジトリについて

共有オブジェクト・リポジトリを使用すると，QuickTest は選択されたコンポーネントのアプリケーション領域に対して指定された共有オブジェクト・リポジトリを使用します。1 つ以上の共有オブジェクト・リポジトリを使用できます(特定のオブジェクトにアクセスする必要があるコンポーネントがかぎられている場合は，それらのオブジェクトを各コンポーネントのローカル・オブジェクト・リポジトリに保存することもできます。詳細については，114 ページ「ローカル・オブジェクト・リポジトリについて」を参照してください)。

コンポーネントの作成を開始した後も，追加の共有オブジェクト・リポジトリを指定できます。新しい共有オブジェクト・リポジトリを作成し，それらをコンポーネントに関連付けることもできます。コンポーネントを実行する前に，そのコンポーネントで使用するオブジェクト・リポジトリに，コンポーネント

内のすべてのオブジェクトが含まれていることを確認する必要があります。含まれていなければ，コンポーネントが失敗する場合があります。詳細については，152 ページ「オブジェクト・リポジトリへのテスト・オブジェクトの追加」を参照してください。

共有オブジェクト・リポジトリを変更するには，オブジェクト・リポジトリ・マネージャを使用します。詳細については，第 7 章「オブジェクト・リポジトリの管理」を参照してください。

ローカル・オブジェクト・リポジトリを使って作業する場合は，次の点に注意してください。

- QuickTest Professional は，共有オブジェクト・リポジトリまたはローカル・オブジェクト・リポジトリ内にすでに存在するテスト・オブジェクトを学習すると，既存の情報を使用し，そのオブジェクト・リポジトリにオブジェクトを追加しません。
- 子オブジェクトがローカル・オブジェクト・リポジトリに追加され，その親が共有オブジェクト・リポジトリ内にある場合は，親がローカル・オブジェクト・リポジトリに自動的に移動されます
- QuickTest は，テスト・オブジェクトを学習すると，関連付けられた共有オブジェクト・リポジトリに同じテスト・オブジェクトがすでに存在する場合を除き，そのテスト・オブジェクトを（共有オブジェクト・リポジトリではなく）ローカル・オブジェクト・リポジトリに追加します（この場合，QuickTest は共有オブジェクト・リポジトリ内の既存の情報を使用しています）。

  オブジェクトは，ローカル・オブジェクト・リポジトリから共有オブジェクト・リポジトリにエクスポートできます。これにより，ローカル・オブジェクトがほかのコンポーネントからアクセスできるようになります。詳細については，139 ページ「共有オブジェクト・リポジトリへのローカル・オブジェクトのエクスポート」を参照してください。

  ローカル・オブジェクト・リポジトリから同じコンポーネントに関連付けられている共有オブジェクト・リポジトリにオブジェクトを直接結合することもできます。これにより，オブジェクトを複数の場所ではなく 1 つの共有された場所で保守できるため，保守を軽減できます。詳細については，283 ページ「ローカル・オブジェクト・リポジトリから共有オブジェクト・リポジトリを更新」を参照してください。

次の表に機能と機能性をリストアップし，それらが［オブジェクト リポジトリ］ウィンドウまたは［オブジェクト リポジトリ マネージャ］で利用できるか示します。

機能性	**［オブジェクト リポジトリ］ウィンドウ**	**［オブジェクト リポジトリ マネージャ］**
152 ページ「オブジェクト・リポジトリへのテスト・オブジェクトの追加」	✓	✓
164 ページ「オブジェクト・リポジトリ内のオブジェクトのコピー，貼り付け，および移動」	✓	✓
167 ページ「オブジェクト・リポジトリからのオブジェクトの削除」	✓	✓
171 ページ「アプリケーション内のオブジェクトの強調表示」	✓	✓
172 ページ「オブジェクト・リポジトリ内のテスト・オブジェクトの検索」	✓	✓
176 ページ「プロパティ値の指定または変更」	✓	✓
178 ページ「アプリケーション内のオブジェクトから認識プロパティを更新」	✓	✓
181 ページ「テスト・オブジェクトの標準必須プロパティの復元」	✓	✓
182 ページ「テスト・オブジェクトの名前の変更」	✓	✓
184 ページ「テスト・オブジェクト記述へのプロパティの追加」	✓	✓
188 ページ「新規認識プロパティの定義」	✓	✓
190 ページ「テスト・オブジェクト記述からのプロパティの削除」	✓	✓
139 ページ「共有オブジェクト・リポジトリへのローカル・オブジェクトのエクスポート」	✓	✗
141 ページ「ローカル・オブジェクト・リポジトリへのオブジェクトのコピー」	✓	✗
233 ページ「新しいオブジェクト・リポジトリの作成」	✗	✓

機能性	[オブジェクト リポジトリ] ウィンドウ	[オブジェクト リポジトリ マネージャ]
234 ページ「オブジェクト・リポジトリを開く」	✗	✓
235 ページ「オブジェクト・リポジトリの保存」	✗	✓
237 ページ「オブジェクト・リポジトリを閉じる」	✗	✓
240 ページ「オブジェクト・リポジトリの編集」	✗	✓
242 ページ「オブジェクト・リポジトリ・マネージャの使用によるコンポーネントへのテスト・オブジェクトの追加」	✓	✓
242 ページ「[ナビゲートして学習] オプションの使用によるテスト・オブジェクトの追加」	✗	✓
246 ページ「リポジトリ・パラメータの管理」	✗	✓
247 ページ「リポジトリ・パラメータの追加」	✗	✓
249 ページ「リポジトリ・パラメータの変更」	✗	✓
249 ページ「リポジトリ・パラメータの削除」	✗	✓
251 ページ「プロパティ値の指定」	✗	✓
254 ページ「テスト・オブジェクトの検索」	✓	✓
255 ページ「結合操作の実行」	✗	✓
257 ページ「XML からのインポート」	✗	✓
258 ページ「XML へのエクスポート」	✗	✓

オブジェクト・スパイを使用したオブジェクト・プロパティと操作の表示

オブジェクト・スパイの指差しマークのメカニズムを使用すると，起動しているアプリケーションに含まれる任意のオブジェクトのサポートされているプロパティと操作を表示できます。アプリケーションのオブジェクト上に指差しマークを移動すると，オブジェクト・スパイに詳細が表示されます。表示される詳細には，テスト・オブジェクトの階層ツリー，その認識プロパティと値，およびオブジェクトに関連付けられた操作などがあります。操作については，構文も表示されます。実行環境オブジェクトの操作の使用方法やそのプロパティ値の取得については，370 ページ「認識プロパティ値の取得と設定」と 372 ページ「ネイティブ・プロパティおよび操作へのアクセス」を参照してください。

ほとんどの環境では，認識プロパティ，ネイティブ・プロパティ，テスト・オブジェクト操作，またはネイティブ操作の表示を選択できます。

認識プロパティ，ネイティブ・プロパティ，テスト・オブジェクト操作，またはネイティブ操作を表示するには，次の手順で行います。

1 アプリケーションを起動して，調査対象のオブジェクトが含まれるページを表示します。

2 ［**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。

3 表示するオブジェクトの詳細を選択します。詳細については，122 ページ「［オブジェクト スパイ］ダイアログ・ボックス」を参照してください。

4 調査対象のオブジェクトに深い階層，または長いプロパティ名および値がある場合は，スクロールせずにすべての情報を表示できるように［オブジェクト スパイ］ダイアログ・ボックスのサイズを変更します。

5 ［オブジェクト スパイ］ダイアログ・ボックスで指差しマークをクリックします。QuickTest が非表示になります。アプリケーションのオブジェクト上に指差しマークを移動すると，オブジェクトが強調表示され，［オブジェクト スパイ］ダイアログ・ボックスにテスト・オブジェクトの階層とプロパティまたは操作を表示できます。

注：アプリケーションのオブジェクトの強調表示は，一部の環境でのみサポートされています。

指差しマークの使用方法の詳細については，120 ページ「指差しマークを使用する際のヒント」を参照してください。

6 アプリケーション内でオブジェクト上に移動します。アプリケーションでオブジェクトが強調表示され，対応するテスト・オブジェクト，そのプロパティまたは操作，およびテスト・オブジェクト階層ツリーが［オブジェクト スパイ］に表示されます。アプリケーションのオブジェクト間でマウスを（クリックしないで）移動すると，各オブジェクトの情報を表示できます。

［オブジェクト スパイ］ダイアログ・ボックスでテスト・オブジェクトに関するさまざまな詳細を表示するには，左の CTRL キーを押したまま，このダイアログ・ボックスで関連オプションをクリックします。

テスト・オブジェクト階層ツリーに現在表示されている別のテスト・オブジェクトのプロパティと操作を表示するには，左の CTRL キーを押したまま，関連テスト・オブジェクトを選択します。

7 ［オブジェクト スパイ］で特定のオブジェクトとその親オブジェクトに関する情報をキャプチャするには，（アプリケーションの）オブジェクトをクリックします。選択したオブジェクト詳細タブとオブジェクト・タイプ・ラジオ・ボタンに応じて，選択したオブジェクトのテスト・オブジェクト階層ツリーと詳細が［オブジェクト スパイ］に表示されます。

オブジェクトをクリックすると，選択したラジオ・ボタンまたはタブを変更して，その他の詳細を表示できます。

テスト・オブジェクト階層ツリーに現在表示されているほかのテスト・オブジェクトのプロパティ，値，または操作を表示するには，ツリーでテスト・オブジェクトを選択します。

指差しマークを使用する際のヒント

➤ 左の CTRL キーを押したままにすると，指差しマークが標準ポインタに変わります。そのときに，QuickTest またはアプリケーションでウィンドウのフォーカスを変更したり，操作（右クリックしたり，スクロール・バーを使ったり，ポ

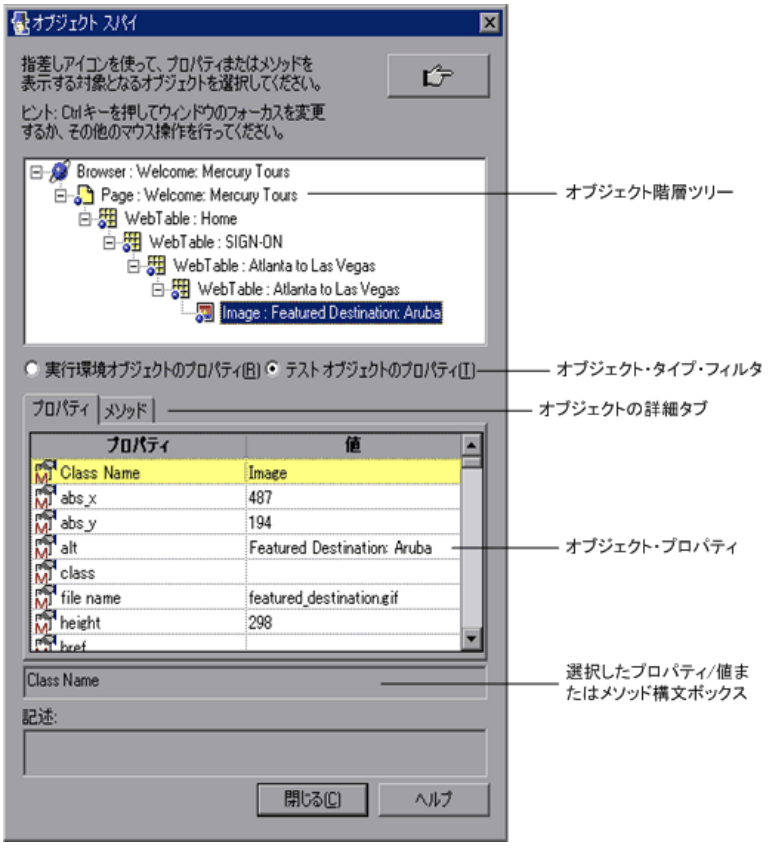
インタをオブジェクト上に移動してショートカット・メニューを表示するなど）を実行できます。

- 選択するオブジェクトがあるウィンドウが別のウィンドウで部分的に隠れている場合は，部分的に隠れているウィンドウが前面に表示されるまで，その上に指差しマークを数秒間置いたままにします。次に，必要なオブジェクトをポイントしてクリックします。ウィンドウを前面に出すのに必要な時間の設定は，［オプション］ダイアログ・ボックスの［一般］表示枠で行えます。
- 選択するオブジェクトがあるウィンドウが別のウィンドウで完全に隠れている場合，またはダイアログ・ボックスがウィンドウの背後に隠れている場合は，必要に応じて左の CTRL キーを押しながらウィンドウを配置します。
- 選択するオブジェクトがあるウィンドウが最小化されている場合は，左の CTRL キーを押したまま，Windows タスク・バーでアプリケーションを右クリックして，ショートカット・メニューから［**元のサイズに戻す**］を選択すると，そのウィンドウを表示できます。
- 特定のイベント（ショートカット・メニューを表示するために右クリックしたり，オブジェクト上にポインタを移動するなど）を実行するだけで，選択するオブジェクトを表示できる場合は，左の CTRL キーを押したままにします。指差しマークが一時的に標準ポインタに変わり，イベントを実行できます。選択するオブジェクトが表示されたら，左の CTRL キーを放します。ポインタが再び指差しマークに変わります。

[オブジェクト スパイ] ダイアログ・ボックス

詳細	開いているアプリケーションでオブジェクトのネイティブなプロパティと操作に加えて，テスト・オブジェクト階層，認識プロパティ，および QuickTest でそのオブジェクトを示すのに使用するテスト・オブジェクトの操作を表示できます。
アクセス方法	➤［**ツール**］＞［**オブジェクト スパイ**］メニュー・コマンドを選択する。 ➤［**オブジェクト スパイ**］ツールバー・ボタン をクリックする。 ➤ ALT+T+S キーを押す。 ［オブジェクト スパイ］ダイアログ・ボックスには，次の場所のどこからでも前述の方法でアクセスできます。 ➤ QuickTest ウィンドウ （47 ページを参照） ➤［オブジェクト リポジトリ］ウィンドウ （129 ページを参照） ➤ オブジェクト・リポジトリ・マネージャ （226 ページを参照）
詳細情報	**概念の概要**：110 ページ「テスト・オブジェクト，ネイティブなプロパティ，および操作について」 **主要なタスク**：119 ページ「オブジェクト・スパイを使用したオブジェクト・プロパティと操作の表示」 **その他の関連項目**：126 ページ「その他の参照」

以下に［オブジェクト スパイ］ダイアログ・ボックスの画像を示します。



［オブジェクト スパイ］ダイアログ・ボックスのオプション

オプション	説明
	指差しマーク・ボタンをクリックして，マウス・ポインタを指差しマークに変えます。次に，指差しマークを使って，表示するプロパティおよび / または操作が含まれているオブジェクトを強調表示またはクリックします。 アプリケーションでオブジェクト上に指差しマークを移動すると，（一部の環境では）アプリケーションのオブジェクトが強調表示され，それらの詳細が［オブジェクト スパイ］ダイアログ・ボックスに表示されます。 ［オブジェクト スパイ］で特定のオブジェクトとその親オブジェクトに関する情報をキャプチャするには，アプリケーションのオブジェクトをクリックします。 120 ページ「指差しマークを使用する際のヒント」を参照。
調査中はオブジェクト スパイを前面に表示する	アプリケーションでオブジェクトをスパイしているときに［オブジェクト スパイ］ダイアログ・ボックスを表示しておくには，このチェック・ボックスを選択します。 **注**：このチェック・ボックスをクリアすると，［オブジェクト スパイ］ダイアログ・ボックスは，画面上でアプリケーションの背後に隠れる場合があります。［オブジェクト スパイ］ダイアログ・ボックスを表示するには，左側の CTRL キーを押し，必要に応じてウィンドウを配置します。

オプション	説明
テスト・オブジェクト階層ツリー	選択したオブジェクトに関連するテスト・オブジェクト階層が表示されます。 オブジェクトが強調表示されている間は，テスト・オブジェクト・クラスがツリーに表示されますが，テスト・オブジェクト名は表示されません。テスト・オブジェクト名（前述の画像の Atlanta to Las Vegas や Featured Destinations など）は，［オブジェクト スパイ］でオブジェクトをクリックして情報をキャプチャした後でのみ表示されます。 表示されたツリー内で別のテスト・オブジェクトのプロパティ，値，または操作を表示するには，ツリーでそのテスト・オブジェクトを選択します。
ネイティブ・プロパティ / ネイティブ操作	このオプションを選択すると，［オブジェクト スパイ］のテスト・オブジェクト階層ツリーで選択したテスト・オブジェクトと関連する実行環境オブジェクトのネイティブ・プロパティまたは操作が表示されます。［**プロパティ**］または［**操作**］タブのどちらを選択するかによって，ラベルが変わります。
認識プロパティ / テスト・オブジェクト操作	このオプションを選択すると，［オブジェクト スパイ］のテスト・オブジェクト階層ツリーで選択したテスト・オブジェクトの認識プロパティまたはテスト・オブジェクト操作が表示されます。［**プロパティ**］または［**操作**］タブのどちらを選択するかによって，ラベルが変わります。
［**プロパティ**］タブ	選択したオブジェクトのネイティブ・プロパティまたは認識プロパティとプロパティの値が表示されます。 ➤［**プロパティ**］:［オブジェクト スパイ］のテスト・オブジェクト階層ツリーで現在選択されているテスト・オブジェクトのプロパティ名，またはそれと関連する実行環境オブジェクトが表示されます。 ➤［**値**］:［**プロパティ**］カラムにリストアップされたプロパティの値が表示されます。

オプション	説明
［**操作**］タブ：	［オブジェクト スパイ］のテスト・オブジェクト階層ツリーで現在選択されているテスト・オブジェクトに関するネイティブ操作またはテスト・オブジェクト操作，およびそれに対応する構文，またはそれと関連する実行環境オブジェクトが表示されます。
選択されたプロパティ，値，または操作ボックス	**［プロパティ］タブ：**ごく最近クリックしたプロパティ名または値が表示されます。 **［操作］タブ：**ごく最近クリックした操作の構文が表示されます。 **ヒント**：このボックスに表示されているテキストをクリップボードにコピーするには，そのテキストを強調表示して，CTRL+C キーを押すか，強調表示されたテキストを右クリックして，［**コピー**］を選択します。
［詳細］	可能な場合は，ごく最近クリックしたプロパティまたは操作について説明します。

その他の参照

関連する概念	➤ 372 ページ「ネイティブ・プロパティおよび操作へのアクセス」 ➤ 370 ページ「認識プロパティ値の取得と設定」

第 4 章

コンポーネントのオブジェクト・リポジトリの使用方法

本章では，コンポーネントに含まれるオブジェクト・リポジトリの使用方法について説明します。[オブジェクト リポジトリ] ウィンドウの使い方，共有リポジトリの関連付けの管理方法，リポジトリ・パラメータ値の割り当て方，実行セッションでテスト・オブジェクトを作成または変更する方法について述べます。

本章の内容

- ［オブジェクト リポジトリ］ウィンドウについて（128 ページ）
- ［オブジェクトのプロパティ］ダイアログ・ボックス（143 ページ）
- リポジトリ・パラメータ値の割り当て（144 ページ）
- 実行セッション中のテスト・オブジェクトでの作業（149 ページ）

［オブジェクト リポジトリ］ウィンドウについて

［オブジェクト リポジトリ］ウィンドウには，現在のコンポーネントに含まれるすべてのテスト・オブジェクト，チェックポイント・オブジェクト，および出力オブジェクトのツリーが表示されます（すべてのローカル・オブジェクト，および選択したコンポーネントと関連する共有オブジェクト・リポジトリのすべてのオブジェクトが含まれます）。

［オブジェクト リポジトリ］ウィンドウには，ツリー内で選択したオブジェクトごとに，テスト・オブジェクトに関する情報（オブジェクトのタイプ，オブジェクトが格納されているリポジトリ，およびオブジェクトの詳細）が表示されます。ローカル・オブジェクトは編集可能（黒色）ですが，共有オブジェクトは読み取り専用形式（灰色）です。

注： QuickTest がインストールされていない環境のテスト・オブジェクトは，オブジェクト・リポジトリに疑問符アイコン付きで表示されます。

［オブジェクト リポジトリ］ウィンドウを開いている間，QuickTest を引き続き使用することができます。また，オブジェクトおよびオブジェクト・リポジトリを引き続き変更することができます。また，必要に応じて［オブジェクト リポジトリ］ウィンドウのサイズ変更もできます。［オブジェクト リポジトリ］ウィンドウには，関連付けられているオブジェクト・リポジトリに加えたすべての変更がリアルタイムで反映されます。たとえば，ローカル・オブジェクト・リポジトリにオブジェクトを追加した場合，または追加のオブジェクト・リポジトリを現在のコンポーネントに関連付けた場合，［オブジェクト リポジトリ］ウィンドウには更新された内容が即座に表示されます。

［オブジェクト リポジトリ］ウィンドウを使って，リポジトリ（ローカルおよび共有オブジェクト・リポジトリ）内の任意のオブジェクトのオブジェクト記述を表示したり，ローカル・オブジェクトとそのプロパティを変更したり，ローカル・オブジェクト・リポジトリにテスト・オブジェクトを追加したりできます。［オブジェクト リポジトリ］ウィンドウからコンポーネントにテスト・オブジェクトをドラッグ・アンド・ドロップすることもできます。テスト・オブジェクトをコンポーネントにドラッグ・アンド・ドロップすると，QuickTest によって，そのテスト・オブジェクトの標準操作を含んだステップがコンポーネントに挿入されます。チェックポイント・オブジェクトや出力オブ

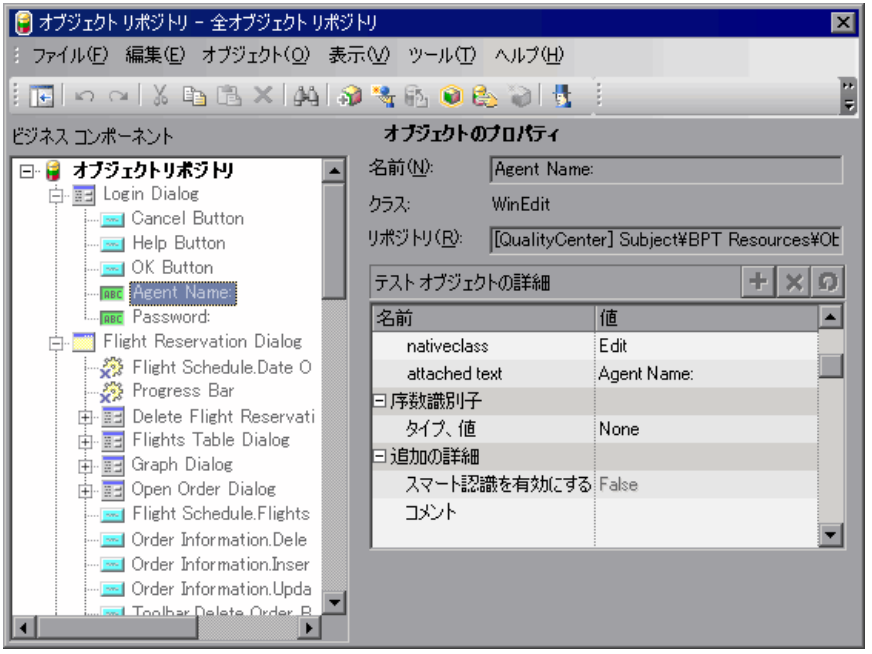
ジェクトは，［オブジェクト リポジトリ］ウィンドウからドラッグ・アンド・ドロップできません。

たとえば，ボタン・オブジェクトをコンポーネントにドラッグ・アンド・ドロップすると，そのボタン・オブジェクトを使って，**クリック**操作（ボタン・オブジェクトに対する標準の操作）を含むステップがコンポーネントに追加されます。

［オブジェクト リポジトリ］ウィンドウ

記述	コンポーネントに対する認識プロパティおよびオブジェクト・リポジトリの関連付けを管理できます。
アクセス方法	➤［**オブジェクト リポジトリ**］ボタン をクリックします。 ➤［リソース］表示枠でリポジトリをダブルクリックするか，リポジトリを右クリックして，［**リポジトリを開く**］を選択します。 ➤［使用可能なキーワード］表示枠でリポジトリのオブジェクトを右クリックし，［**リソースを開く**］を選択します。 ➤［**リソース**］＞［**オブジェクト リポジトリ**］を選択します。
詳細情報	**概念の概要**：128 ページ「［オブジェクト リポジトリ］ウィンドウについて」 **主要なタスク：** ➤ 152 ページ「オブジェクト・リポジトリへのテスト・オブジェクトの追加」 ➤ 164 ページ「オブジェクト・リポジトリ内のオブジェクトのコピー，貼り付け，および移動」 ➤ 167 ページ「オブジェクト・リポジトリからのオブジェクトの削除」 ➤ 168 ページ「オブジェクトの検索」 ➤ 175 ページ「認識プロパティの保守」 **その他の関連項目**：135 ページ「その他の参照」

以下に［オブジェクト リポジトリ］ウィンドウの画像を示します。



[オブジェクト リポジトリ] ウィンドウ－編集ツールバー

[オブジェクト リポジトリ] ウィンドウの編集ツールバーには，次のボタンがあります。

ボタン	名前	説明
	簡易表示	**簡易表示**モードでは，オブジェクト・リポジトリ・ツリーだけが表示されますが，**完全表示**モードでは，オブジェクト・リポジトリ・ツリーとともにオブジェクトの詳細領域も表示されます。
	全体表示	
	[元に戻す]	ローカル・オブジェクトに加えたすべての変更は，変更後直ちに，そのローカル・オブジェクトを使用するすべてのステップで自動的に更新されます。[**編集**] > [**元に戻す**] および [**編集**] > [**やり直し**] メニュー・オプション，または [**元に戻す**] および [**やり直し**] ツールバー・ボタンを使用すると，変更の取り消しや繰り返しができます。現在のコンポーネントを保存すると，保存操作の前に実行した操作を元に戻したり，やり直すことができなくなります。
	[やり直し]	
	[切り取り]	オブジェクト・リポジトリ・ツリーから選択したオブジェクトを切り取ります。詳細については，164 ページ「オブジェクト・リポジトリ内のオブジェクトのコピー，貼り付け，および移動」を参照してください。
	[貼り付け]	クリップボードにあるオブジェクトを，ツリーで選択したオブジェクトの子として，オブジェクト・リポジトリ・ツリーに貼り付けます。最下位のオブジェクトは子を持つことができません。詳細については，164 ページ「オブジェクト・リポジトリ内のオブジェクトのコピー，貼り付け，および移動」を参照してください。
	[コピー]	オブジェクト・リポジトリ・ツリーから選択したオブジェクトをクリップボードにコピーします。詳細については，164 ページ「オブジェクト・リポジトリ内のオブジェクトのコピー，貼り付け，および移動」を参照してください。
	[削除]	オブジェクト・リポジトリ・ツリーから選択したオブジェクトを削除します。詳細については，167 ページ「オブジェクト・リポジトリからのオブジェクトの削除」を参照してください。

ボタン	名前	説明
	[検索と置換]	オブジェクト・リポジトリでオブジェクトを検索したり置換します。詳細については，168 ページ「オブジェクト・リポジトリ内のオブジェクトの検索」を参照してください。
	[ローカルへオブジェクトを追加]	オブジェクトをローカル・オブジェクト・リポジトリに追加します。詳細については，152 ページ「オブジェクト・リポジトリへのテスト・オブジェクトの追加」を参照してください。
	[アプリケーションから更新]	アプリケーション内のオブジェクトから認識プロパティを更新します。詳細については，178 ページ「アプリケーション内のオブジェクトから認識プロパティを更新」を参照してください。
	[新規テストオブジェクトの定義]	新規テスト・オブジェクトを定義します。詳細については，161 ページ「新規テスト・オブジェクトの定義」を参照してください。
	[アプリケーションを強調表示]	オブジェクト・リポジトリ・ツリーで選択したオブジェクトをアプリケーションで強調表示します。詳細については，171 ページ「アプリケーション内のオブジェクトの強調表示」を参照してください。
	[リポジトリの中で検索]	テスト中のアプリケーションでオブジェクトを選択して，そのテスト・オブジェクトをオブジェクト・リポジトリで強調表示できます。詳細については，172 ページ「オブジェクト・リポジトリ内のテスト・オブジェクトの検索」を参照してください。
	[オブジェクトスパイ]	開いているアプリケーションでオブジェクトのネイティブなプロパティと操作に加えて，テスト・オブジェクト階層，認識プロパティ，および QuickTest でそのオブジェクトを示すのに使用するテスト・オブジェクトの操作を表示できる。詳細については，122 ページ「[オブジェクト スパイ] ダイアログ・ボックス」を参照してください。
	[リポジトリの関連付け]	共有オブジェクト・リポジトリとコンポーネントの関連付けを管理できます。詳細については，144 ページ「リポジトリ・パラメータ値の割り当て」を参照してください。

[オブジェクト リポジトリ] ウィンドウ－編集ツールバー

フィルタ・ツールバーには次のオプションがあります。

オプション	説明
	フィルタ・ツールバーを使って，[オブジェクト リポジトリ] ウィンドウに表示されるオブジェクトにフィルタを適用できます。 次のいずれかの条件に一致するオブジェクトを表示できます。 ➤ 現在のコンポーネントに含まれるすべてのオブジェクト ➤ 現在のコンポーネント含まれるローカル・オブジェクトのみ ➤ 現在のコンポーネントに関連付けられている特定の共有オブジェクト・リポジトリのオブジェクトのみ [オブジェクト リポジトリ] ウィンドウにフィルタを適用するには，次の手順を実行します。 [**フィルタ**] ツールバー・リストで，次のいずれかのオプションを選択します。 ➤ **[全オブジェクト]** ➤ **[ローカル オブジェクト]** ➤ 現在のコンポーネントに関連付けられている特定の共有オブジェクト・リポジトリの名前 選択した場所のオブジェクトだけを表示するようにオブジェクト・リポジトリ・ツリーにフィルタが設定されます。[オブジェクト リポジトリ] ウィンドウのタイトル・バーには，現在のフィルタが示されます。

［オブジェクト リポジトリ ウィンドウ］オプション

［オブジェクト リポジトリ］ウィンドウには次のオプションがあります。

オプション	説明
［**ビジネス コンポーネント**］	現在のテスト・ドキュメントがビジネス・コンポーネントであることを示します。
［**テスト オブジェクト**］ツリー	現在のコンポーネントに含まれるすべてのテスト・オブジェクト（すべてのローカル・オブジェクト，および選択されたコンポーネントに関連付けられたすべての共有オブジェクト・リポジトリ内のすべてのテスト・オブジェクト）が表示されます。 **注**：同じ名前，オブジェクト・クラス，および親階層を持つテスト・オブジェクトが，関連付けられている複数のオブジェクト・リポジトリに存在する場合は，定義されている優先順位に基づいて最初に見つかったオブジェクトだけがオブジェクト・リポジトリ・ツリーに表示されます。オブジェクト・リポジトリの優先順位の詳細については，465 ページ「共有オブジェクト・ライブラリの関連付け」を参照してください。 オブジェクト・リポジトリ・ツリーに表示されるオブジェクトにはフィルタを適用できます。詳細については，133 ページ「［オブジェクト リポジトリ］ウィンドウ－編集ツールバー」を参照してください。
［**チェックポイントおよび出力オブジェクト**］ツリー	現在のコンポーネントに含まれるすべてのチェックポイント・オブジェクトおよび出力オブジェクト（すべてのローカル・チェックポイントおよび出力オブジェクト，および選択されたコンポーネントに関連付けられているすべての共有オブジェクト・リポジトリ内のすべてのチェックポイントおよび出力オブジェクト）が表示されます。
［**名前**］	QuickTest によってオブジェクトに割り当てられる名前です。ローカル・オブジェクト・リポジトリ内のオブジェクトの名前は変更できます。詳細については，182 ページ「テスト・オブジェクトの名前の変更」を参照してください。
［**クラス**］	オブジェクトのクラスです。

オプション	説明
［**リポジトリ**］	オブジェクトが格納されているオブジェクト・リポジトリの場所（ファイル名とパス）です。オブジェクトがローカル・オブジェクト・リポジトリに格納されている場合は，［**ローカル**］と表示されます。
［**オブジェクトの詳細**］	実行セッション中にテスト・オブジェクトの識別に使用されるプロパティおよびプロパティ値，またはチェックポイントや出力オブジェクトのプロパティを表示します。ローカル・オブジェクト・リポジトリ内のオブジェクトの詳細は，変更することもできます。詳細については，136 ページ「オブジェクトの詳細領域について」を参照してください。オブジェクトの詳細領域は表示 / 非表示を選択できます。詳細については，131 ページ「[オブジェクト リポジトリ］ウィンドウ－編集ツールバー」を参照してください。

その他の参照

［**関連タスク**］	➤ 144 ページ「リポジトリ・パラメータ値の割り当て」 ➤ 139 ページ「共有オブジェクト・リポジトリへのローカル・オブジェクトのエクスポート」 ➤ 141 ページ「ローカル・オブジェクト・リポジトリへのオブジェクトのコピー」 ➤ 182 ページ「テスト・オブジェクトの名前の変更」 ➤ ほかの場所からテスト・オブジェクトをドラッグ・アンド・ドロップできます。詳細については，831 ページ「[使用可能なキーワード］表示枠について」および 242 ページ「オブジェクト・リポジトリ・マネージャの使用によるコンポーネントへのテスト・オブジェクトの追加」を参照してください。 ➤ テスト実行中に，テスト・オブジェクトのプロパティを変更できます。詳細については，149 ページ「実行セッション中のテスト・オブジェクトでの作業」を参照してください。 ➤ ほかの場所からオブジェクト・プロパティをド表示して変更できます。詳細については，175 ページ「認識プロパティの保守」を参照してください。

オブジェクトの詳細領域について

［オブジェクト リポジトリ］ウィンドウの右下にあるオブジェクトの詳細領域では，実行セッション中にオブジェクトの認識に使用されるプロパティとプロパティ値，またはチェックポイント・オブジェクトや出力オブジェクトのプロパティを表示および変更できます。

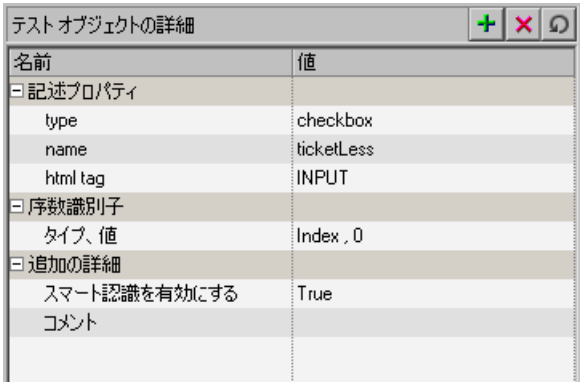
［オブジェクト リポジトリ］ウィンドウでは，共有オブジェクト・リポジトリ内のオブジェクトがオブジェクト・プロパティ表示枠（オブジェクトの詳細領域を含む）に読み取り専用形式で表示されます。共有オブジェクト・リポジトリ内のオブジェクトを変更するには，その共有オブジェクト・リポジトリをオブジェクト・リポジトリ・マネージャで開きます。詳細については，第 7 章「オブジェクト・リポジトリの管理」を参照してください。また，共有オブジェクト・リポジトリ内のオブジェクトをローカル・オブジェクト・リポジトリにコピーし，そのローカル・コピーを変更することもできます。詳細については，141 ページ「ローカル・オブジェクト・リポジトリへのオブジェクトのコピー」を参照してください。

ヒント：

- オブジェクトのプロパティおよびプロパティ値を表示するには，［オブジェクトのプロパティ］ダイアログ・ボックスを使用します。詳細については，143 ページ「［オブジェクトのプロパティ］ダイアログ・ボックス」を参照してください。

- いつでも［オブジェクト スパイ］を使用すると，テスト中のアプリケーションにあるオブジェクトのネイティブまたは認識プロパティと値を表示できます。オブジェクト・スパイを開くには，［**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ます。詳細については，119 ページ「オブジェクト・スパイを使用したオブジェクト・プロパティと操作の表示」を参照してください。

ローカル・オブジェクト・リポジトリに保存されたオブジェクトについては，テスト・オブジェクトの詳細を変更できます。



オブジェクトの詳細領域には，テスト・オブジェクトに関する次の項目が表示されます。

項目	説明
[記述プロパティ]	実行セッション中にオブジェクトの認識に使われるプロパティとおよびプロパティ値です。 プロパティはテスト・オブジェクト記述に追加したり削除できます。詳細については，184 ページ「テスト・オブジェクト記述へのプロパティの追加」を参照してください。 プロパティ値は定数として指定することも，その値をパラメータ化することもできます。詳細については，176 ページ「プロパティ値の指定または変更」を参照してください。
[序数識別子]	記述がまったく同じであるほかのオブジェクト（すべてのプロパティが同じ値のオブジェクト）に対する該当オブジェクトの相対的な順序または場所を示す数値。詳細については，191 ページ「序数識別子の指定」を参照してください。

項目	説明
[追加の詳細]	次のオプションが含まれています。 ➤ [**スマート認識を有効にする**]：**True** または **False** を選択することにより，QuickTest が実行セッション中にテスト・オブジェクト記述を使ってオブジェクトを認識できない場合にスマート認識を使ってテスト・オブジェクトを認識するかどうかを指定できます。 **注**：このオプションは，[オブジェクトの認識] ダイアログ・ボックスでテスト・オブジェクトのクラスにスマート認識プロパティが定義されている場合にのみ使用できます。スマート認識の詳細については，209 ページ「スマート認識の設定」を参照してください。 ➤ [**コメント**]：テスト・オブジェクトに関するテキスト情報を追加できます。

チェックポイント・オブジェクトと出力オブジェクトの場合は，オブジェクトの詳細領域にチェックポイント・オブジェクトまたは出力値オブジェクトのプロパティが表示されます。オブジェクトの詳細領域で，これらのプロパティを変更できます。

ヒント：

- ➤ ローカル・オブジェクト・リポジトリに保存したオブジェクトのチェックポイントおよび出力値の詳細は変更できます。
- ➤ 共有オブジェクト・リポジトリのオブジェクトをローカル・オブジェクト・リポジトリにコピーして変更できます。

詳細については，次を参照してください。

- ➤ 570 ページ「[チェックポイントのプロパティ] ダイアログ・ボックスについて」
- ➤ 585 ページ「ビットマップ・チェックポイントのプロパティ・ダイアログ・ボックス」
- ➤ 593 ページ「値の出力について」

共有オブジェクト・リポジトリへのローカル・オブジェクトのエクスポート

本項で説明する機能は，［**オブジェクト リポジトリ**］ウィンドウで作業している場合にのみ利用できます。

コンポーネントのローカル・オブジェクト・リポジトリに含まれているすべてのテスト・オブジェクト，チェックポイント・オブジェクト，および出力値オブジェクトをファイル・システム内の新しい共有オブジェクト・リポジトリまたは（QuickTest が Quality Center に接続されている場合は）Quality Center プロジェクトにエクスポートできます。これにより，ローカル・オブジェクトがほかのコンポーネントからアクセスできるようになります。

注 :［**ローカル オブジェクトのエクスポートと置換**］オプション（［**ファイル**］＞［**ローカル オブジェクトのエクスポートと置換**］）は，コンポーネントには使用できません。

ローカル・オブジェクトを共有オブジェクト・リポジトリにエクスポートすると，パラメータ化されたオブジェクトのパラメータがソース・パラメータと同じ名前を使ってリポジトリ・パラメータに変換されます。各リポジトリ・パラメータの標準設定値（割り当てられた値）は，対応するソース・パラメータです。

コンポーネント内で使用されている割り当てを変更するには，（144 ページ「リポジトリ・パラメータ値の割り当て」で説明している）［リポジトリ パラメータの割り当て］ダイアログ・ボックスを使用します。リポジトリ・パラメータの詳細については，第 7 章「オブジェクト・リポジトリの管理」を参照してください。

ヒント：ローカル・オブジェクトをエクスポートした後は，オブジェクト・リポジトリ結合ツールを使用して，エクスポートされたオブジェクトを含む共有オブジェクト・リポジトリのテスト・オブジェクトをほかの共有オブジェクト・リポジトリと結合できます。詳細については，第 8 章「共有オブジェクト・リポジトリの結合」を参照してください。

ローカル・オブジェクトを新しい共有オブジェクト・リポジトリにエクスポートするには，次の手順を実行します。

1 エクスポートするローカル・オブジェクトを含んでいるコンポーネントを開きます。

2 ［**リソース**］＞［**オブジェクト リポジトリ**］を選択するか，［**オブジェクト リポジトリ**］ボタンをクリックして，［オブジェクト リポジトリ］ウィンドウを開きます。

3 ［**ファイル**］＞［**ローカル オブジェクトのエクスポート**］を選択します。［共有オブジェクト リポジトリの保存］ダイアログ・ボックスが開きます。

4 サイドバーで，ファイルを保存する場所を選択します（File System や Quality Center Test Resources など）。

5 ファイルを保存するフォルダを参照し，そのフォルダを選択します。

6 ［**ファイル名**］ボックスに，ファイルの名前を入力します。ファイルを識別しやすいように，わかりやすい名前を使用します。オブジェクト・リポジトリの名前には，次の文字は使用できません。
¥ / : * " ? < > | '

オブジェクト・リポジトリを Quality Center に保存する場合は，ファイル・パスに連続した 2 つのセミコロン（;;）を含める必要があります。

7 ［**保存**］をクリックします。

［**ローカル オブジェクトのエクスポート**］を選択した場合は，指定した共有オブジェクト・リポジトリ（拡張子が **.tsr** のファイル）にローカル・オブジェクトがエクスポートされます。コンポーネントではローカル・オブジェクト・リポジトリ内のオブジェクトが引き続き使用され，新しい共有オブジェクト・リポジトリはテストに関連付けられません。

新しい共有オブジェクト・リポジトリは，ほかの共有オブジェクト・リポジトリと同じように使用できます。

ローカル・オブジェクト・リポジトリへのオブジェクトのコピー

本項で説明する機能は，［オブジェクト リポジトリ］ウィンドウで作業している場合にのみ利用できます。

共有オブジェクト・リポジトリに格納されているオブジェクトを変更するには，オブジェクト・リポジトリ・マネージャを使用して変更します。また，［オブジェクト リポジトリ］ウィンドウを使用してローカルに変更することも可能です。

オブジェクト・リポジトリ・マネージャを使用してオブジェクトを変更すると，その変更内容は共有オブジェクト・リポジトリを使用するすべてのコンポーネントに反映されます。オブジェクトのローカル・コピーを作成し，それを［オブジェクト リポジトリ］ウィンドウで変更すると，その変更内容は変更を行ったコンポーネントにのみ反映されます。後で共有オブジェクト・リポジトリ内の同じオブジェクトを変更しても，変更内容はコンポーネント内のオブジェクトのローカル・コピーに反映されません。

オブジェクトをローカル・オブジェクト・リポジトリにコピーする場合は，次のことを考慮してください。

- オブジェクトをローカル・オブジェクト・リポジトリにコピーすると，その親オブジェクトもローカル・オブジェクト・リポジトリにコピーされます。
- オブジェクトまたはその親オブジェクトの中で未割り当てのリポジトリ・パラメータが使用されている場合，当該オブジェクトはローカル・オブジェクト・リポジトリにコピーできません。オブジェクトをローカル・オブジェクト・リポジトリにコピーする前に，すべてのリポジトリ・パラメータが割り当て済みであることを確認する必要があります。
- オブジェクトまたはその親オブジェクトが 1 つ以上のリポジトリ・パラメータを使ってパラメータ化されている場合，リポジトリ・パラメータの値はオブジェクトをローカル・オブジェクト・リポジトリにコピーしたときに変換されます。たとえば，リポジトリ・パラメータがローカル・パラメータに割り当てられている場合，プロパティはローカル・パラメータを使用してパラメータ化されます。値が定数値の場合，プロパティも同じ定数値を受け取ります。
- 複数のオブジェクトをローカル・オブジェクト・リポジトリにコピーするときに，未割り当てのリポジトリ・パラメータを含むオブジェクトがある場合や，

割り当て済みのリポジトリ・パラメータの値を変換したくない場合は，コピー処理中に当該オブジェクトをスキップできます。その後，元の選択範囲の次のオブジェクトから引き続きコピーを継続できます。

オブジェクトをローカル・オブジェクト・リポジトリにコピーするには，次の手順を実行します。

1 ［オブジェクト リポジトリ］ウィンドウで，ローカル・オブジェクト・リポジトリにコピーするオブジェクトを共有オブジェクト・リポジトリから選択します。共有オブジェクト・リポジトリのオブジェクトは，灰色で表示されています。同じ親オブジェクトを持つオブジェクトであれば，複数のオブジェクトをコピー対象として選択できます。

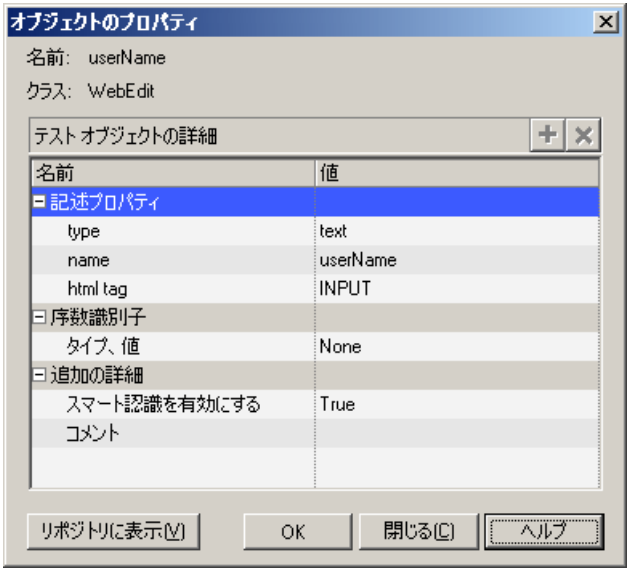
2 ［**オブジェクト**］＞［**ローカルへコピー**］を選択するか，オブジェクトを右クリックして［**ローカルへコピー**］を選択します。オブジェクト（および親オブジェクト）がローカル・オブジェクト・リポジトリにコピーされ，編集可能になります。

[オブジェクトのプロパティ] ダイアログ・ボックス

コンポーネントのステップに含まれるオブジェクトについて，認識プロパティおよびプロパティ値を表示できます。

オブジェクトのプロパティおよびプロパティ値を表示するには，次の手順を実行します。

コンポーネントで，プロパティを表示するオブジェクトのステップをクリックし，[**編集**] > [**ステップのプロパティ**] > [**オブジェクトのプロパティ**] を選択します。[オブジェクトのプロパティ] ダイアログ・ボックスが開きます。



注：選択したオブジェクトがローカル・オブジェクト・リポジトリに現在格納されているのか，あるいは現在のコンポーネントのアプリケーション領域に関連付けられた共有オブジェクト・リポジトリに格納されているのかに応じて，[オブジェクト プロパティ] ダイアログ・ボックスは若干異なります。本項では，ダイアログ・ボックスに表示されるオプションのうち，ローカル・オブジェクト・リポジトリ内のオブジェクトに関するオプションについて説明します。共有オブジェクト・リポジトリに格納されているオブジェクトの場合，この情報は読み取り専用形式です。

［オブジェクトのプロパティ］ダイアログ・ボックスでは，選択したオブジェクトの名前とクラスが表示され，次の操作を実行できます。

- オブジェクトのプロパティおよびプロパティ値（記述プロパティ，序数識別子，およびその他の設定）の表示。
- オブジェクトの認識に使われるプロパティおよびプロパティ値の変更（ローカル・オブジェクト・リポジトリに格納されているオブジェクトの場合）。［オブジェクトのプロパティ］ダイアログ・ボックスでプロパティと値を変更する方法は，［オブジェクト リポジトリ］ウィンドウでテスト・オブジェクトの詳細を変更する場合と同じです。詳細については，175 ページ「認識プロパティの保守」を参照してください。
- ［**オブジェクト内で表示**］ボタンをクリックして［オブジェクト リポジトリ］ウィンドウを開き，選択したオブジェクトをオブジェクト階層内に表示します（オブジェクト・リポジトリに格納されているオブジェクトの場合）。
- ［**リポジトリに追加**］ボタンをクリックして，選択したオブジェクトをローカル・オブジェクト・リポジトリに追加します（オブジェクト・リポジトリに格納されていないオブジェクトの場合）。

リポジトリ・パラメータ値の割り当て

コンポーネントに関連付けられている共有オブジェクト・リポジトリで使用されるリポジトリ・パラメータを割り当てることができます。リポジトリ・パラメータを値またはパラメータに割り当てることにより，実行セッション中にテスト・オブジェクトの認識に使用されるプロパティ値を指定します。プロパティ値は定数値から取得するように指定することも，ローカルまたはコンポーネント・パラメータを使ってプロパティ値をパラメータ化することもできます。

リポジトリ・パラメータを持つオブジェクト・リポジトリが関連付けられている各コンポーネント内の各リポジトリ・パラメータを，必要に応じて割り当てることができます。たとえば，あるコンポーネントでは，ユーザ名オブジェクトのテキスト・プロパティ値を環境変数パラメータから取得し，別のコンポーネントでは，同じオブジェクト・プロパティ値が定数，ローカル・パラメータを使用するといった使い分けが可能です。

リポジトリ・パラメータを割り当てるまでは，同じコンポーネントに関連付けられている複数の共有オブジェクト・リポジトリに，同じ名前の複数のリポジ

トリ・パラメータがある場合，（共有オブジェクト・リポジトリのリストで定義された）優先順位が最も高い共有オブジェクト・リポジトリのリポジトリ・パラメータが使用されます。リポジトリ・パラメータを割り当てると，QuickTest はユーザが定義した割り当てを使用します。また，パラメータの割り当て後に，優先順位または標準設定値を変更しても影響ありません。

標準設定値がないリポジトリ・パラメータを使ってパラメータ化されたオブジェクト・プロパティ値を持つオブジェクト・リポジトリを使用するコンポーネントを開くと，割り当てを必要とするリポジトリ・パラメータが存在することを示すメッセージが［欠落リソース］表示枠に表示されます。その場合は，そのコンポーネント内のリポジトリ・パラメータを必要に応じて割り当てることができます。また，標準設定値があるリポジトリ・パラメータの割り当てを行ったり，すでに割り当てのあるリポジトリ・パラメータの割り当てを変更したりもできます。

リポジトリ・パラメータを割り当てない場合は，そのパラメータに対して定義された標準設定値が（存在する場合は）コンポーネントの実行中に使用されます。パラメータが未割り当ての（つまり，パラメータの標準設定値が指定されていない）場合，コンポーネントの実行は，未割り当てのパラメータ値を含むテスト・オブジェクトを認識できずに失敗する可能性があります。

リポジトリ・パラメータ値を割り当てるには，次の手順を実行します。

1［**リソース**］＞［**リポジトリ パラメータの割り当て**］を選択します。［リポジトリ パラメータの割り当て］ダイアログ・ボックスが開きます。

ヒント：コンポーネント内に未割り当てのリポジトリ・パラメータ（標準設定値がないリポジトリ・パラメータ）がある場合は，［欠落リソース］表示枠の［**リポジトリ パラメータ**］行をダブルクリックしてこのダイアログ・ボックスを開くこともできます 詳細については，第 35 章「欠落リソースの処理」を参照してください。

［リポジトリ パラメータの割り当て］ダイアログ・ボックスには，次のオプションが含まれます。

オプション名	説明
［**次のパラメータの割り当て**］フィルタ	表示されるパラメータのリストに対してフィルタを設定します。次の表示方法を選択できます。 ➤［**未割り当ての全パラメータ**］: テスト内の未割り当ての値を持つすべてのパラメータが表示されます。 ➤［**<コンポーネント名>**］（たとえば，LogIn）: 指定したコンポーネント内のパラメータが（値の割り当ての有無に関係なく）すべて表示されます。
［**名前**］カラム	リポジトリ・パラメータの名前です。
［**値**］カラム	パラメータの現在の値です（存在する場合）。このカラムには，ユーザが定義した新しい値か，パラメータの作成時に定義した標準設定値のいずれかが表示されます。標準設定値が定義されていない場合，パラメータは現在未割り当てであり，［**{ 標準設定値なし }**］というテキストが表示されます。 次のいずれかを実行できます。 ➤ 新しい定数値を入力する。 ➤ 該当するパラメータの［**値**］セルをクリックし，パラメータ化ボタン をクリックして，値をパラメータ化する。 ➤ 該当するパラメータの［**値**］セルをクリックして［**標準設定値にリセット**］ボタン をクリックして，パラメータを標準設定値にリセットする。
［**詳細**］カラム	パラメータの説明文です（存在する場合）。
［**リポジトリの中を検索**］ボタン	［オブジェクト リポジトリ］ウィンドウが開き，選択したリポジトリ・パラメータを使用している最初のテスト・オブジェクトがオブジェクト・リポジトリ・ツリー内で強調表示されます。このボタンを再度クリックすると，選択したパラメータの次の出現箇所を検索できます。

注：リポジトリ・パラメータの名前，標準設定値，および記述は，［リポジトリ パラメータの管理］ダイアログ・ボックスで定義します。また，名前と記述については，このダイアログ・ボックスでのみ変更できます。詳細については，246 ページ「リポジトリ・パラメータの管理」を参照してください。

2 ［**次のパラメータの割り当て**］の矢印をクリックし，値を定義する対象となるパラメータ・グループのリストを選択します。次の表示方法を選択できます。

- ［**未割り当ての全パラメータ**］：テスト内の未割り当ての値を持つすべてのパラメータが表示されます。
- ［**<コンポーネント名>**］（たとえば，LogIn）：指定したコンポーネント内のパラメータが（値の割り当ての有無に関係なく）すべて表示されます。

3 割り当て対象パラメータの［**値**］セルをクリックします。値の割り当て方法として，次のいずれかを選択できます。

- 新しい定数値を入力する場合，または既存の定数値を変更する場合は，［**値**］セルに直接入力します。また，パラメータ化ボタンをクリックして，［値設定オプション］ダイアログ・ボックスに定数値を入力することもできます。このダイアログ・ボックスの使用法については，を参照してください。『**HP QuickTest Professional ユーザーズ・ガイド**』を参照してください。

- 値をパラメータ化する場合は，パラメータ化ボタンをクリックします。［値設定オプション］ダイアログ・ボックスが開きます。値のパラメータ化には，ローカル・パラメータまたはコンポーネント・パラメータを使用できます このダイアログ・ボックスの使用法については，を参照してください。『**HP QuickTest Professional ユーザーズ・ガイド**』を参照してください。

- 標準設定値を復元する場合は，［**標準設定値をクリア**］ボタンをクリックします。［リポジトリ パラメータの追加］ダイアログ・ボックスで定義した標準設定値が（存在する場合は）セルに表示されます。［リポジトリ パラメータの追加］ダイアログ・ボックスの詳細については，247 ページ「リポジトリ・パラメータの追加」を参照してください。

4 割り当てるパラメータ値がほかにもある場合は，手順 3 を繰り返します。次に［**OK**］をクリックして，［リポジトリ パラメータの割り当て］ダイアログ・ボックスを閉じます。

実行セッション中のテスト・オブジェクトでの作業

QuickTest は，実行セッション中に初めてオブジェクトに遭遇すると，その実行セッションの一時的なテスト・オブジェクト・バージョンを作成します。QuickTest は，オブジェクト記述を使ってこの一時的なオブジェクトのバージョンを作成します。以降，コンポーネントでは，オブジェクト・リポジトリ内のテスト・オブジェクトではなく，この一時的なバージョンのテスト・オブジェクトが参照されます。

注：記録および実行セッション中は，［オブジェクト リポジトリ］ウィンドウが読み取り専用になります。

実行セッション中のテスト・オブジェクトの作成

プログラムによる記述を使用して，アプリケーションのオブジェクトを表す一時的なバージョンのテスト・オブジェクトを作成できます。オブジェクト・リポジトリを参照せずに，それらのオブジェクトに対して操作を実行できます。たとえば，Web サイトのフォームにエディット・ボックスが追加されたとします。プログラムによる記述を使用して，新しいエディット・ボックスに値を入力するステートメントをユーザ定義関数に追加することで，オブジェクト・リポジトリへの追加もしていないオブジェクトを QuickTest に認識させることができます。プログラム的記述の詳細については，350 ページ「プログラム的記述の使用」を参照してください。

実行セッション中の認識プロパティの変更

ユーザ定義関数の中で SetTOProperty ステートメントを追加することで，オブジェクト・リポジトリ内の永続的な値に影響を与えることなく，実行セッション中に一時的なバージョンのオブジェクトのプロパティを変更できます。

SetTOProperty メソッドには，次の構文を使用します。

Object(description).SetTOProperty Property, Value

詳細については，『**HP QuickTest Professional Object Model Reference**』（英語版）を参照してください。

第 5 章

オブジェクト・リポジトリ内のテスト・オブジェクトの管理

本章では，オブジェクト・リポジトリに含まれるオブジェクトの管理と保守の方法について説明します。また，オブジェクトのプロパティを変更する方法，QuickTest によるオブジェクトの認識方法を変更する方法を取り上げます。これは，動的に変化するオブジェクトを対象に作業する場合に便利です。

本章の内容

- オブジェクト・リポジトリへのテスト・オブジェクトの追加（152 ページ）
- オブジェクト・リポジトリ内のオブジェクトのコピー，貼り付け，および移動（164 ページ）
- オブジェクト・リポジトリからのオブジェクトの削除（167 ページ）
- オブジェクトの検索（168 ページ）
- 認識プロパティの保守（175 ページ）

オブジェクト・リポジトリへのテスト・オブジェクトの追加

本項で説明する機能は，ローカル・オブジェクト・リポジトリに関しては［**オブジェクト リポジトリ**］ウィンドウで，共有オブジェクト・リポジトリに関しては［**オブジェクト リポジトリ マネージャ**］で利用できます。

キーワード駆動型のテスト・インフラストラクチャ用の共有オブジェクト・リポジトリを作成すると，さまざまな方法でテスト・オブジェクトを追加できます。選択したオブジェクトだけを追加するか，特定のタイプのすべてのオブジェクト（すべてのボタン・オブジェクトなど）を追加するか，特定のタイプおよびクラスのすべてのオブジェクト（すべての **WebButton** オブジェクトなど）を追加するかを選択できます。

たとえば，［ナビゲートして学習］オプションを使用すると，定義したフィルタに応じて共有オブジェクト・リポジトリにオブジェクトを追加できます。コンポーネントを記録すると，操作の対象となった各オブジェクトが QuickTest によってローカル・オブジェクト・リポジトリに追加されます（関連付けられている共有オブジェクト・リポジトリにまだ存在しないオブジェクトの場合）。コンポーネントの編集中にも，ローカル・オブジェクト・リポジトリにオブジェクトを追加できます。

たとえば，ユーザが，オブジェクト・リポジトリにはないオブジェクトを対象にステップを実行する必要がある場合があります。また，オブジェクト・リポジトリを構築した後で，テスト中のアプリケーションに追加されたオブジェクトがある場合もあります。オブジェクト・リポジトリ・マネージャを使うと，このオブジェクトを共有オブジェクト・リポジトリに直接追加できるため，その共有オブジェクト・リポジトリを使用するすべてのコンポーネントで利用可能になります。または，コンポーネントのローカル・オブジェクト・リポジトリにオブジェクトを追加することもできます。

注：テスト・オブジェクトをローカル・オブジェクト・リポジトリに追加できるのは，そのコンポーネントに関連付けられている共有オブジェクト・リポジトリに当該オブジェクトがまだ存在していない場合のみです関連付けられている共有オブジェクト・リポジトリにテスト・オブジェクトがすでに存在する場合は，［**ローカルにコピー**］オプションを使ってそのオブジェクトをローカル・オブジェクト・リポジトリに追加できます。詳細については，141 ページ「ローカル・オブジェクト・リポジトリへのオブジェクトのコピー」を参照してください。

必要に応じて，ローカル・オブジェクト・リポジトリのテスト・オブジェクトを共有オブジェクト・リポジトリに結合できます。詳細については，第 8 章「共有オブジェクト・リポジトリの結合」を参照してください。

アプリケーションをナビゲートしているときにテスト・オブジェクトを共有オブジェクト・リポジトリに追加することもできます。詳細については，242 ページ「[ナビゲートして学習］オプションの使用によるテスト・オブジェクトの追加」を参照してください。

ヒント：

- ［ローカルへオブジェクトを追加］ボタンを使用して，（キーワード・ビューの新規ステップ）［ステップでオブジェクトを選択］ダイアログ・ボックスでアプリケーションのテスト・オブジェクトを選択することで，テスト・オブジェクトをローカル・リポジトリに追加することもできます。詳細については，525 ページ「ステップの項目の選択」を参照してください。
- アプリケーションにまだ存在しない新しいテスト・オブジェクトをオブジェクト・リポジトリに追加できます。詳細については，161 ページ「新規テスト・オブジェクトの定義」を参照してください。

［ローカルへオブジェクトを追加］オプションまたは［オブジェクトの追加］オプションを使ったテスト・オブジェクトの追加

アプリケーションからローカルまたは共有オブジェクト・リポジトリにテスト・オブジェクトを直接追加できます。特定のテスト・オブジェクトをその子孫とともに追加するかどうかを選択できます。オブジェクト・フィルタで定義した選択に基づいて，オブジェクトやクラスのタイプに従って追加する子孫を制御することもできます。

注 :［オブジェクト リポジトリ］ウィンドウの［**ローカルへオブジェクトを追加**］ボタンやオブジェクト・リポジトリ・マネージャの［**オブジェクトの追加**］ボタンを使って WinMenu オブジェクトをオブジェクト・リポジトリに追加することはできません。WinMenu オブジェクトをオブジェクト・リポジトリに追加するには，［**オブジェクトの追加**］ボタンまたは［**ローカルへオブジェクトを追加**］ボタンを使ってその親オブジェクトを追加し，親オブジェクトをその子孫とともに追加することを選択するか，または WinMenu オブジェクトに対するステップを記録し，記録されたステップを削除してください。

［ローカルへオブジェクトを追加］または［オブジェクトの追加］オプションを使用してオブジェクト・リポジトリにテスト・オブジェクトを追加するには，次の手順を実行します。

1 次のいずれかを実行します。

➤［オブジェクト リポジトリ］ウィンドウで，［**オブジェクト**］＞［**ローカルへオブジェクトを追加**］を選択するか，［**ローカルへオブジェクトを追加**］ツールバー・ボタンをクリックします。このオプションを選択した場合，テスト・オブジェクトがローカル・オブジェクト・リポジトリに追加され，現在のコンポーネントでのみ使用できるようになります。

➤ オブジェクト・リポジトリ・マネージャで，［**オブジェクト**］＞［**オブジェクトの追加**］を選択するか，［**オブジェクトの追加**］ツールバー・ボタンをクリックします。このオプションを選択した場合，テスト・オブジェクトが共有オブジェクト・リポジトリに追加され，複数のコンポーネントで使用できるようになります。

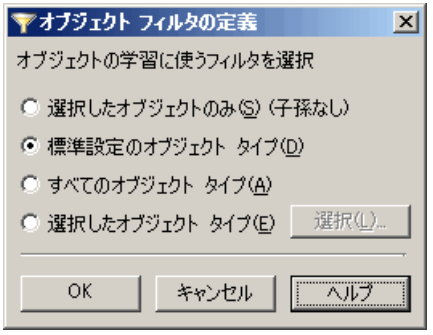
QuickTest と［オブジェクト リポジトリ］ウィンドウまたはオブジェクト・リポジトリ・マネージャが非表示になり，ポインタが指さし型に変わります。指差しマークの使用方法の詳細については，157 ページ「指差しマークの使用法に関するヒント」を参照してください。

2 オブジェクト・リポジトリに追加するオブジェクトをクリックします。

3 クリックした場所が複数のオブジェクトに関連付けられている場合は，［オブジェクトの選択］ダイアログ・ボックスが開きます。リポジトリに追加するオブジェクトを選択し，［**OK**］をクリックします。

［オブジェクトの選択］ダイアログ・ボックスで選択したオブジェクトがテスト・オブジェクト階層の最下位のオブジェクト（たとえば，WebButton オブジェクト）である場合は，オブジェクト・リポジトリに直接追加されます。

［オブジェクトの選択］ダイアログ・ボックスで選択したオブジェクトが，Web 環境のブラウザやページ，または標準 Windows アプリケーションのダイアログ・ボックスなどの親（コンテナ）オブジェクトである場合は，［オブジェクト フィルタの定義］ダイアログ・ボックスが開きます。［オブジェクト フィルタの定義］ダイアログ・ボックスには，前のオブジェクト追加セッションで定義した設定が保持されています。



次のオプションのいずれかを選択することができます。

➤ ［**選択したオブジェクトのみ（子孫なし）**］：事前に選択したオブジェクトのプロパティと値をオブジェクト・リポジトリに追加します。子オブジェクトは追加しません。

- ［**標準設定のオブジェクト タイプ**］：事前に選択したオブジェクトのプロパティと値をオブジェクト・リポジトリに追加します。標準設定のフィルタで指定したオブジェクトのタイプに従って，子孫オブジェクトのプロパティと値も追加します。標準設定のフィルタにどのオブジェクトが含まれているかを確認するには，［**選択**］ボタンをクリックし，［**標準設定**］ボタンをクリックします。
- ［**すべてのオブジェクト タイプ**］：事前に選択したオブジェクトのプロパティと値をオブジェクト・リポジトリに追加し，すべての子孫オブジェクトのプロパティと値も追加します。
- ［**選択したオブジェクト タイプ**］：事前に選択したオブジェクトのプロパティと値をオブジェクト・リポジトリに追加し，オブジェクト・フィルタで指定したオブジェクトのタイプおよびクラスに従って，子孫オブジェクトのプロパティと値も追加します。フィルタにオブジェクトやクラスを指定するには，［**選択**］ボタンをクリックし，［オブジェクト タイプの選択］ダイアログ・ボックスで必要な項目を選択します。［オブジェクト タイプの選択］ダイアログ・ボックスの詳細については，159 ページ「［オブジェクト タイプの選択］ダイアログ・ボックスについて」を参照してください。

4 必要なオプションを選択し，［**OK**］をクリックします。［オブジェクト フィルタの定義］ダイアログ・ボックスが閉じ，選択したオブジェクト・フィルタに従って指定した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。

5 ［オブジェクト リポジトリ］ウィンドウが再表示され，オブジェクト・リポジトリ内に新しいローカル・オブジェクトとそれらのプロパティが表示されます。オブジェクト・リポジトリ・マネージャを使用してオブジェクトを追加した場合は，アクティブな共有オブジェクト・リポジトリにオブジェクトが追加されます。

また，オブジェクト・リポジトリに新規オブジェクトの親オブジェクトが存在しない場合は，QuickTest によって親オブジェクトが追加されます。オブジェクト・リポジトリ・ツリーでは，ローカル・オブジェクトは編集可能であることを示す黒色で表示されます。共有オブジェクトは灰色で表示され，オブジェクト・リポジトリ・マネージャでのみ編集できます。

新しいテスト・オブジェクトの詳細の編集方法は，ローカルまたは共有オブジェクト・リポジトリ内のほかのオブジェクトと同じです。詳細については，175 ページ「認識プロパティの保守」を参照してください。

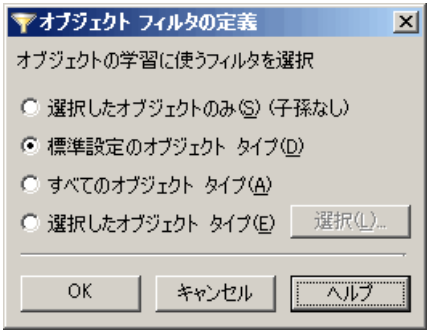
指差しマークの使用法に関するヒント

- 左の CTRL キーを押したままにすると，指差しマークが標準ポインタに変わります。そのときに，QuickTest またはアプリケーションでウィンドウのフォーカスを変更したり，操作（右クリックしたり，スクロール・バーを使ったり，ポインタをオブジェクト上に移動してショートカット・メニューを表示するなど）を実行できます。
- 選択するオブジェクトがあるウィンドウが別のウィンドウで部分的に隠れている場合は，部分的に隠れているウィンドウが前面に表示されるまで，その上に指差しマークを数秒間置いたままにします。次に，必要なオブジェクトをポイントしてクリックします。ウィンドウを前面に出すのに必要な時間の設定は，［オプション］ダイアログ・ボックスの［一般］表示枠で行えます。
- 選択するオブジェクトがあるウィンドウが別のウィンドウで完全に隠れている場合，またはダイアログ・ボックスがウィンドウの背後に隠れている場合は，必要に応じて左の CTRL キーを押しながらウィンドウを配置します。
- 選択するオブジェクトがあるウィンドウが最小化されている場合は，左の CTRL キーを押したまま，Windows タスク・バーでアプリケーションを右クリックして，ショートカット・メニューから［**元のサイズに戻す**］を選択すると，そのウィンドウを表示できます。
- 特定のイベント（ショートカット・メニューを表示するために右クリックしたり，オブジェクト上にポインタを移動するなど）を実行するだけで，選択するオブジェクトを表示できる場合は，左の CTRL キーを押したままにします。指差しマークが一時的に標準ポインタに変わり，イベントを実行できます。選択するオブジェクトが表示されたら，左の CTRL キーを放します。ポインタが再び指差しマークに変わります。

［オブジェクト フィルタの定義］ダイアログ・ボックスについて

オブジェクト・リポジトリにテスト・オブジェクトを追加するときに，追加するために選択したオブジェクトが一般に親オブジェクト（Web 環境のブラウザやページ，標準 Windows アプリケーションのダイアログ・ボックスなど）である場合は，［オブジェクト フィルタの定義］ダイアログ・ボックスが開きます。

オブジェクト・フィルタには，（［**ナビゲートして学習**］オプションや［**オブジェクトの追加**］オプションの使用中に）どのオブジェクトを学習する必要があるかを指定した定義済みの設定が含まれています。［オブジェクト フィルタの定義］ダイアログ・ボックスで選択したオプションは，その後の各学習セッションで保存され，使用されます。



次のオプションのいずれかを選択することができます。

- ［**選択したオブジェクトのみ（子孫なし）**］：事前に選択したオブジェクトのプロパティと値をオブジェクト・リポジトリに追加します。子オブジェクトは追加しません。
- ［**標準設定のオブジェクト タイプ**］：事前に選択したオブジェクトのプロパティと値をオブジェクト・リポジトリに追加します。標準設定のフィルタで指定したオブジェクトのタイプに従って，子孫オブジェクトのプロパティと値も追加します。標準設定のフィルタにどのオブジェクトが含まれているかを確認するには，［**選択したオブジェクト タイプ**］を選択し，［**選択**］ボタンをクリックして，［**標準設定**］ボタンをクリックします。
- ［**すべてのオブジェクト タイプ**］：事前に選択したオブジェクトのプロパティと値をオブジェクト・リポジトリに追加し，すべての子孫オブジェクトのプロパティと値も追加します。
- ［**選択したオブジェクト タイプ**］：事前に選択したオブジェクトのプロパティと値をオブジェクト・リポジトリに追加し，オブジェクト・フィルタで指定した

オブジェクトのタイプおよびクラスに従って，子孫オブジェクトのプロパティと値も追加します。フィルタにオブジェクトやクラスを指定するには，［**選択**］ボタンをクリックし，［オブジェクト タイプの選択］ダイアログ・ボックスで必要な項目を選択します。［オブジェクト タイプの選択］ダイアログ・ボックスの詳細については，159 ページ「［オブジェクト タイプの選択］ダイアログ・ボックスについて」を参照してください。

［オブジェクト タイプの選択］ダイアログ・ボックスについて

［オブジェクト タイプの選択］ダイアログ・ボックスを使って，（［**ナビゲートして学習**］オプションや［**オブジェクトの追加**］オプションの使用中に）オブジェクト・リポジトリにテスト・オブジェクトを追加するためのユーザ定義オブジェクト・フィルタを指定できます。

定義したオブジェクト・フィルタは，その後の（［**ナビゲートして学習**］オプションや［**オブジェクトの追加**］オプションを使って行われる）オブジェクト追加操作のために自動的に保存されます。

［オブジェクト タイプの選択］ダイアログ・ボックスを開くには，［オブジェクト フィルタの定義］ダイアログ・ボックスで［**選択**］ボタンをクリックします。

このリストに示されるオブジェクト・タイプは，オブジェクトを一般的な特性に基づいてグループ分けした一般的な分類です。たとえば，［**List**］タイプにはリスト・オブジェクトやリスト・ビュー・オブジェクトに加えコンボ・ボックスが含まれます。［**Table**］タイプには，テーブルとグリッドの両方が含まれています。

リストには，インストールされているアドインでサポートされるすべてのオブジェクトが表示されます。選択したオブジェクトに固有のリストではありません。一部のアドインでは，特定のオブジェクトのすべての子孫を追加しようとしたときに，それらのオブジェクト・タイプをリストで選択していても，特定の子オブジェクトが自動的にフィルタによって除外され，オブジェクト・リポジトリに追加されない場合があります。フィルタによって自動的に除外されるオブジェクトを追加する必要がある場合は，目的のオブジェクトを［オブジェクトの選択］ダイアログ・ボックスで選択することにより追加できます。使用しているアドインが特定のオブジェクトを自動的にフィルタで除外するかどうかを確認するには，『**HP QuickTest Professional アドイン・ガイド**』を参照してください。

ヒント：［オブジェクト タイプの選択］ダイアログ・ボックスですべてのチェック・ボックスを選択またはクリアするには，［**すべて選択**］または［**すべてクリア**］をクリックします。チェック・ボックスの選択を標準設定に戻すには，［**標準設定**］をクリックします。標準設定は，［オブジェクト フィルタの定義］ダイアログ・ボックスで［**標準設定のオブジェクト タイプ**］オプションを選択した場合と同じです。

オプションを選択して［**OK**］をクリックします。ユーザ定義オブジェクト・フィルタが定義され，［オブジェクト タイプの選択］ダイアログ・ボックスが閉じます。

新規テスト・オブジェクトの定義

アプリケーションにまだ存在しないテスト・オブジェクトをオブジェクト・リポジトリに定義できます。これによって，テスト対象アプリケーションの準備ができる前にオブジェクト・リポジトリを準備し，アプリケーション用のコンポーネントを構築できます。

たとえば，アプリケーション内の一部のオブジェクトについてはすでに名前，タイプ，および記述プロパティがわかっており，アプリケーション内のほかのオブジェクトについてはタイプのみがわかっている場合があります。アプリケーションの準備ができる前に，ログイン・ページの UserName および Password フィールド用の WebEdit オブジェクト（および対応する親の Page および Browser オブジェクト）を作成できます。これらのオブジェクトのプロパティ値がわかっている場合は，それらを追加することもできます。わかっていない場合は，アプリケーションの準備ができてテストが可能になってから値を追加できます。

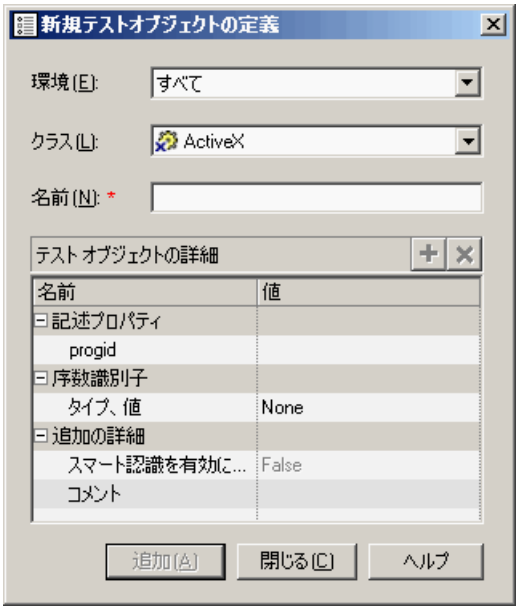
本項の説明に従って新規オブジェクトをオブジェクト・リポジトリに定義した場合，そのオブジェクトはローカル・オブジェクト・リポジトリに追加され，現在のコンポーネントでのみ使用できるようになります。オブジェクトを複数のコンポーネントで使用できるように，共有オブジェクト・リポジトリに追加するには，オブジェクト・リポジトリ・マネージャを使用する必要があります。詳細については，第 7 章「オブジェクト・リポジトリの管理」を参照してください。

新規テスト・オブジェクトを定義した後で，アプリケーション内のオブジェクトのプロパティが定義したテスト・オブジェクトの記述と一致しない場合や，アプリケーション内のオブジェクトが更新された場合は，いつでもオブジェクト記述を更新できます。詳細については，178 ページ「アプリケーション内のオブジェクトから認識プロパティを更新」を参照してください。

新規テスト・オブジェクトを定義するには，次の手順を実行します。

1 正しいオブジェクト階層に従って，新しくオブジェクトを定義する位置のすぐ上にあるオブジェクトを選択します。

2［**新規テスト オブジェクトの定義**］ボタンをクリックするか，［**オブジェクト**］＞［**新規テスト オブジェクトの定義**］を選択します。［新規テスト オブジェクトの定義］ダイアログ・ボックスが表示されます。



3［**環境**］ボックスで，適切な環境を選択します。選択した環境に関連付けられているテスト・オブジェクト・クラスが［**クラス**］ボックスに表示されます。

注 :

- ［**環境**］リストに含まれている環境は，読み込まれたアドインに対応するものです。アドインの読み込みの詳細については，『**HP QuickTest Professional アドイン・ガイド**』の QuickTest アドインの読み込みに関する項を参照してください。
- ［**環境**］リストには，ユーザまたはサードパーティがアドイン拡張機能を使って開発した追加の環境が含まれていることもあります。

4 ［**クラス**］ボックスで，定義するテスト・オブジェクトのクラスを選択します。

5 ［**名前**］ボックスに，新規テスト・オブジェクトの名前を入力します。名前を入力すると，［**テスト オブジェクトの詳細**］領域が有効になります。

6 ［**テスト オブジェクトの詳細**］領域で，テスト・オブジェクトのプロパティと値を定義します。［**テスト オブジェクトの詳細**］領域には，［オブジェクトの認識］ダイアログ・ボックスの中でオブジェクト・クラスに対して定義された必須のプロパティが自動的に含まれます。必要に応じてプロパティを追加または削除したり，プロパティに対しての値を定義したりできます。詳細については，175 ページ「認識プロパティの保守」を参照してください。

7 ［**追加**］をクリックします。新規テスト・オブジェクトが，ローカル・オブジェクト・リポジトリ内の選択した場所に追加されます。

8 手順 3 から 7 を繰り返して追加のテスト・オブジェクトを定義するか，［**閉じる**］をクリックして［新規テスト オブジェクトの定義］ダイアログ・ボックスを閉じます。

オブジェクト・リポジトリ内のオブジェクトのコピー，貼り付け，および移動

本項で説明する機能は，ローカル・オブジェクト・リポジトリのオブジェクトに関しては［**オブジェクト リポジトリ**］ウィンドウで，共有オブジェクト・リポジトリのオブジェクトに関しては［**オブジェクト リポジトリ マネージャ**］で利用できます。

注 :［**編集**］＞［**元にもどす**］および［**編集**］＞［**やり直し**］オプション，または［**元に戻す**］および［**やり直し**］ボタンを使用すると，変更の取り消しや繰り返しができます。オブジェクト・リポジトリを保存した後で，保存操作の前に実行した操作を元に戻したりやり直したりすることはできません。

以下の手順は，オブジェクトのコピー，貼り付け，および移動の方法について説明したものです。

オブジェクトをオブジェクト・リポジトリ内の別の場所に移動するには，次の手順を実行します。

オブジェクトをツリーの上または下にドラッグし，必要な場所でドロップします。標準設定では，オブジェクトをドラッグすると，子オブジェクトもすべて一緒に移動されます。

オブジェクトをオブジェクト・リポジトリ内の別の場所にコピーするには，次の手順を実行します。

CTRL キーを押しながらオブジェクトをドラッグし，ツリー内の必要な場所にドロップします。標準設定では，オブジェクトをドラッグすると，子オブジェクトもすべて一緒に移動されます。

子オブジェクトなしでオブジェクトを移動またはコピーするには，次の手順を実行します。

右マウス・ボタンを使ってオブジェクトをドラッグします。必要な場所にオブジェクトをドロップする際，子オブジェクトと一緒にオブジェクトをドロップするかどうかを選択できます。標準設定では，オブジェクトをドラッグすると，子オブジェクトもすべて一緒に移動またはコピーされます。

オブジェクト・リポジトリの内部でオブジェクトの切り取り，コピー，および貼り付けを行うには，次の手順を実行します。

対応するツールバー・ボタンまたは［**編集**］メニューのオプションを使用します。オブジェクトの切り取り，コピー，および貼り付けを行うと，その操作は選択したオブジェクトの子オブジェクト（あれば）にも実行されます。

複数の共有オブジェクト・リポジトリ間でオブジェクトの切り取り，コピー，および貼り付けを行うには，次の手順を実行します。

オブジェクト・リポジトリ・マネージャで，対応するツールバー・ボタンまたは［**編集**］メニューのオプションを使用します。オブジェクトの切り取り，コピー，および貼り付けを行うと，その操作は選択したオブジェクトの子オブジェクト（あれば）にも実行されます。

異なる共有オブジェクト・リポジトリ間でオブジェクトをコピーするには，次の手順を実行します。

オブジェクト・リポジトリ・マネージャで，必要な共有オブジェクト・リポジトリを開きます。一方のウィンドウからオブジェクトをドラッグし，もう一方のウィンドウの必要な場所にドロップします。

異なる共有オブジェクト・リポジトリ間でオブジェクトを移動するには，次の手順を実行します。

オブジェクト・リポジトリ・マネージャで，必要な共有オブジェクト・リポジトリを開きます。CTRL キーを押しながら一方のウィンドウからオブジェクトをドラッグし，もう一方のウィンドウの必要な場所にドロップします。オブジェクトを移動すると，そのオブジェクトは一方の共有オブジェクト・リポジトリから削除され，もう一方の共有オブジェクト・リポジトリに追加されます。

共有オブジェクト・リポジトリからローカル・オブジェクト・リポジトリにオブジェクトをコピーして，ローカルでオブジェクトを変更することもできます。詳細については，141 ページ「ローカル・オブジェクト・リポジトリへのオブジェクトのコピー」を参照してください。

オブジェクトのコピー，貼り付け，移動に関するガイドライン

オブジェクトのコピー，貼り付け，移動を行う場合は，次のガイドラインを考慮してください。

- オブジェクト・リポジトリのルート・ノードは変更できません。
- オブジェクト階層を変更する場合は，新しい階層が有効であることを確認してください。
- オブジェクトを別の階層レベルに貼り付けたり移動したりする場合は，共有の親オブジェクトまでのすべてのオブジェクトをコピーするかどうかを（この操作を実行したときに表示されるメッセージ内で）選択できます。
- ［オブジェクト リポジトリ］ウィンドウで，コンポーネントに関連付けられた共有オブジェクト・リポジトリからオブジェクトのコピー，貼り付け，移動を行うと，そのオブジェクトはコンポーネントのローカル・オブジェクト・リポジトリにコピー，貼り付け，または移動されます。
- オブジェクトを直接の親に移動する場合，QuickTest によってオブジェクトのコピー（1 つずつ大きくなる数字が名前の後ろに付加されます）が作成され，元のオブジェクトの兄弟として貼り付けられます。
- オブジェクトの切り取りまたはコピーを行い，それを親オブジェクトに貼り付けると，QuickTest によってそのオブジェクトのコピーが作成され（1 つずつ大きくなる数字が名前の後ろに付加されます），元のオブジェクトと同じレベルに挿入されます。
- オブジェクトをその子孫に移動することはできません。
- オブジェクトをオブジェクト階層の最下位にあるオブジェクト（子オブジェクトを持つことができないオブジェクト）の子になるようにコピーも移動もできません。
- 未割り当てのリポジトリ・パラメータを持つオブジェクトを共有オブジェクト・リポジトリからローカル・オブジェクト・リポジトリにコピー，貼り付け，または移動することはできません。共有オブジェクト・リポジトリからローカル・オブジェクト・リポジトリにオブジェクトをコピー，貼り付け，または移動し，そのオブジェクトまたはいずれかの親オブジェクトが 1 つ以上のリポジトリ・パラメータを使ってパラメータ化されている場合は，オブジェクトをコピー，貼り付け，または移動したときにリポジトリ・パラメータの値が変換されます。たとえば，リポジトリ・パラメータがローカル・パラメータに割り当てられている場合，プロパティはローカル・パラメータを使用してパラメータ化されます。値が定数値の場合，プロパティも同じ定数値を受け取ります。

オブジェクト・リポジトリからのオブジェクトの削除

本項で説明する機能は，ローカル・オブジェクト・リポジトリのオブジェクトに関しては［**オブジェクト リポジトリ**］ウィンドウで，共有オブジェクト・リポジトリのオブジェクトに関しては［**オブジェクト リポジトリ マネージャ**］で利用できます。

コンポーネントからステップを削除しても，対応するオブジェクトはオブジェクト・リポジトリから削除されずに残ります。

ローカル・オブジェクト・リポジトリを対象に作業しているときに，削除したステップに含まれるオブジェクトがコンポーネント内のほかのステップに一切含まれていなければ，そのオブジェクトをオブジェクト・リポジトリから削除できます。

共有オブジェクト・リポジトリを対象に作業している場合は，オブジェクト・リポジトリからオブジェクトを削除する前に，同じ共有オブジェクト・リポジトリを使用しているどのコンポーネントにもそのオブジェクトが含まれていないことを確認する必要があります。

オブジェクトを削除するには，ローカル・オブジェクト・リポジトリ内のオブジェクトの場合は［オブジェクト リポジトリ］ウィンドウを使用し，共有オブジェクト・リポジトリのオブジェクト内のオブジェクトの場合はオブジェクト・リポジトリ・マネージャを使用します。

注：オブジェクト・リポジトリから削除したオブジェクトへの参照がコンポーネントに含まれていると，コンポーネントの実行は失敗します。

オブジェクト・リポジトリからオブジェクトを削除するには，次の手順を実行します。

1 リポジトリ・ツリーの中で，削除するオブジェクトを選択します。

2［**削除**］ボタンをクリックするか，［**編集**］＞［**削除**］を選択します。

3［**はい**］をクリックし，オブジェクトの削除を確定します。オブジェクトがオブジェクト・リポジトリから削除されます。

ヒント：［**削除**］ボタンを使用すれば，テスト・オブジェクトではなく，オブジェクト・リポジトリ内の選択した値や項目を削除できます。たとえば，このボタンを使用して，オブジェクト名やプロパティ値の一部分を削除できます。

オブジェクトの検索

本項で説明する機能は，ローカル・オブジェクト・リポジトリのオブジェクトに関しては［**オブジェクト リポジトリ**］ウィンドウで，共有オブジェクト・リポジトリのオブジェクトに関しては［**オブジェクト リポジトリ マネージャ**］で利用できます。

オブジェクト・リポジトリ内の特定のオブジェクトを検索する方法は複数あります。オブジェクトのタイプに基づいて，オブジェクトを検索できます。たとえば，特定のエディット・ボックスを検索したり，アプリケーション内のオブジェクトにポイントすることで，リポジトリ内の同じオブジェクトを自動的に強調表示できます。オブジェクト・リポジトリ内でオブジェクトを選択すると，アプリケーション上でそのオブジェクトが強調表示され，どのオブジェクトに該当するかを確認できます。ローカル・オブジェクト（およびオブジェクト・リポジトリ・マネージャを使用する場合，編集可能な共有オブジェクト・リポジトリ内の共有オブジェクト）の場合，特定のプロパティ値をほかのプロパティ値で置換することもできます。たとえば，プロパティ値 userName を user name で置換できます。

オブジェクト・リポジトリ内のオブジェクトの検索

［検索と置換］ダイアログ・ボックスを使用して，オブジェクト・リポジトリ内のオブジェクト，プロパティ，またはプロパティ値を検索できます。指定したプロパティ値を検索して置換することもできます。

ローカル・オブジェクト・リポジトリ内のオブジェクトのプロパティ値を置換するには，［オブジェクト リポジトリ］ウィンドウを使用します。共有オブジェクト・リポジトリ内のオブジェクトのプロパティ値を置換するには，オブジェクト・リポジトリ・マネージャを使用します。

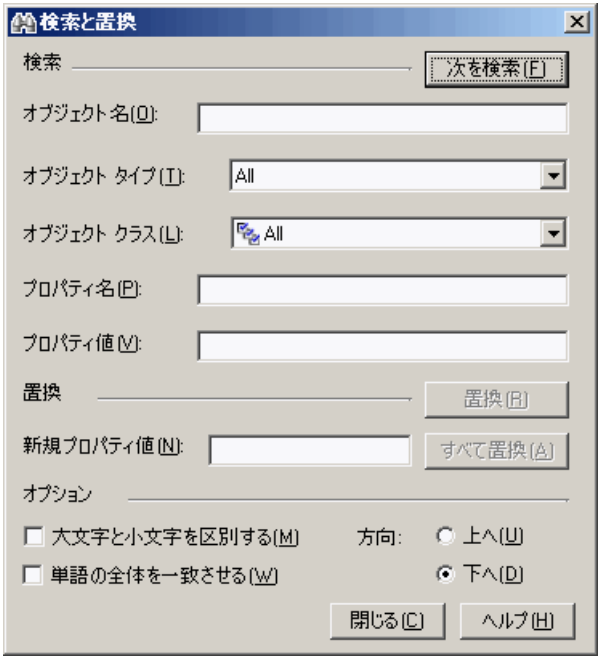
注：

- ➤［検索と置換］ダイアログ・ボックスでオブジェクト名を検索することによって検索できるのは，チェックポイントおよび出力値のみです。
- ➤［検索と置換］ダイアログ・ボックスを使ってプロパティやオブジェクトの名前を置換することはできません。読み取り専用のコンポーネントのプロパティ値を置換することはできません。

オブジェクト・リポジトリ内のオブジェクト，プロパティ，またはプロパティ値を検索するには，次の手順を実行します。

1 対象となるオブジェクト・リポジトリが（[オブジェクト リポジトリ］ウィンドウまたはオブジェクト・リポジトリ・マネージャで）開いていることを確認します。

2［**検索と置換**］ボタンをクリックするか，［**編集**］＞［**検索と置換**］を選択します。［検索と置換］ダイアログ・ボックスが開きます。



3 オブジェクト，プロパティまたはプロパティ値の検索に使用する条件を 1 つ以上指定します。

- ［**オブジェクト名**］：検索するオブジェクトの名前または名前の一部を入力します。
- **［オブジェクト タイプ］**：検索するオブジェクトのタイプ（**Button** など）を選択します。

注： このリストに示されるオブジェクト・タイプは，オブジェクトを一般的な特性に基づいてグループ分けした一般的な分類です。たとえば，［**List**］タイプにはリスト・オブジェクトやリスト・ビュー・オブジェクトに加えコンボ・ボックスが含まれます。［**Table**］タイプには，テーブルとグリッドの両方が含まれています。

- ［**オブジェクト クラス**］：検索するオブジェクトのクラス（**WebButton** など）を選択します。使用できるクラスは，［**オブジェクト タイプ**］ボックスでの選択に応じて異なります。
- ［**プロパティ名**］：検索するプロパティの名前または名前の一部を指定します。
- ［**プロパティ値**］：検索するプロパティ値またはプロパティ値の一部を指定します。

4 プロパティ値を指定し，その値を別の値に置換したい場合は，［**新規プロパティ値**］ボックスに新しいプロパティ値を入力します。

5 次のようにして，検索パラメータを指定します。

- 大文字と小文字を区別して検索するには，［**大文字と小文字を区別する**］を選択します。
- 入力した 1 つの単語に完全に一致する文字列だけを検索するには，［**単語の全体を一致させる**］を選択します。
- 検索する方向を指定します（［**上へ**］または［**下へ**］）を選択します。

6 次のいずれかの方法で検索操作または置換操作を実行します。検索は，オブジェクト・リポジトリ全体を対象に，現在選択されているオブジェクトを起点

に，指定した方向に向かって実行されます。次に一致する文字列を検索するには，再度［**次を検索**］をクリックします。

- 指定したオブジェクト，プロパティ，またはプロパティ値を検索するには，［**次を検索**］をクリックします。検索文字列に一致する最初の文字列が表示されます。
- 検索対象のプロパティ値が出現する個所を 1 つずつ検索して置換するには［**次を検索**］をクリックします。一致する文字列が見つかったら，［**置換**］をクリックします。プロパティ値が置換され，次に一致するプロパティ値が出現する箇所が（存在する場合は）強調表示されます。
- 指定したプロパティ値が出現するすべての箇所を新しいプロパティ値で置換するには［**すべて置換**］をクリックします。編集可能になっていない共有オブジェクト・リポジトリ内の出現箇所は，変更されません。

アプリケーション内のオブジェクトの強調表示

オブジェクト・リポジトリ内のテスト・オブジェクトを選択すると，テスト中のアプリケーション内で強調表示できます。テスト・オブジェクトを選択し強調表示すると，QuickTest は選択したオブジェクトの周りに一時的に枠を表示し，少しの間点滅させることによって，アプリケーション内のオブジェクトの位置を示します。オブジェクトが見えるよう正しいコンテキストでアプリケーションが開いている必要があります。

たとえば，Web ページ内の［User Name］エディット・ボックスを探すには，Web ブラウザで該当するページを開き，オブジェクト・リポジトリ内で「userName」テスト・オブジェクトを選択します。［**アプリケーションを強調表示**］オプションを選択すると，ブラウザ上の Web ページ内にある［User Name］エディット・ボックスが枠に囲まれ，数回点滅します。

注：枠の表示も点滅も一時的な動作です。

アプリケーション内のオブジェクトを強調表示するには，次の手順を実行します。

1 アプリケーションが正しいウィンドウまたはページに開いているか確認します。

2 オブジェクト・リポジトリの中で，強調表示するオブジェクトをクリックします。

3 ［**アプリケーションを強調表示**］ボタンをクリックするか，［**表示**］＞［**アプリケーションを強調表示**］を選択します。選択したオブジェクトがアプリケーション内で境界線で囲まれて強調表示されます。

注：正しいコンテキストでアプリケーションが開いていない場合は，オブジェクトが強調表示されず，メッセージが表示されます。

オブジェクト・リポジトリ内のテスト・オブジェクトの検索

テスト中のアプリケーション内のオブジェクトを選択して，そのテスト・オブジェクトをオブジェクト・リポジトリ内で強調表示できます。

たとえば，Web ページ内の「Find a Flight」画像を特定するには，指差しマークのメカニズムを使って Web ページ上でその画像を選択します。選択ダイアログ・ボックスで「Find a Flight」画像を選択し，［**OK**］をクリックすると，オブジェクト・リポジトリ・ツリー内の親階層が展開され，「Find a Flight」画像テスト・オブジェクトが強調表示されます。

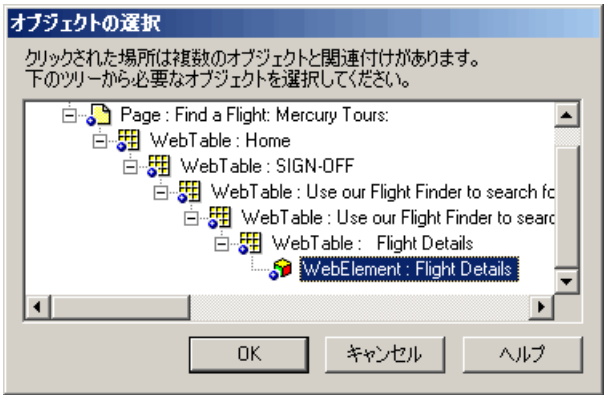
オブジェクト・リポジトリ内のオブジェクトの場所を特定するには，次の手順を実行します。

1 アプリケーションが正しいウィンドウまたはページに開いているか確認します。

2 ［**リポジトリの中で検索**］ボタンをクリックするか，［**表示**］＞［**リポジトリの中で検索**］を選択します。QuickTest が非表示になり，ポインタが指差しマークに変わります。

3 指差しマークを使用して，アプリケーション内の必要なオブジェクトをクリックします。指差しマークの使用方法の詳細については，173 ページ「指差しマークの使用法に関するヒント」を参照してください。

クリックした場所が複数のオブジェクトに関連付けられている場合，［オブジェクトの選択］ダイアログ・ボックスが開きます。



4 オブジェクト・リポジトリ内の場所を特定する対象となるオブジェクトを選択し，［**OK**］をクリックします。選択したオブジェクトが，オブジェクト・リポジトリ内で強調表示されます

ヒント：該当するオブジェクト・リポジトリが開いていない場合や，オブジェクトが見つからない場合，オブジェクトは強調表示されません。オブジェクト・リポジトリ・マネージャで，複数の共有オブジェクト・リポジトリを開いており，選択したオブジェクトがアクティブなオブジェクト・リポジトリ内で見つからない場合は，現在開いているすべてのオブジェクト・リポジトリの中でオブジェクトを検索するかどうかを選択できます。

指差しマークの使用法に関するヒント

➤ 左の CTRL キーを押したままにすると，指差しマークが標準ポインタに変わります。そのときに，QuickTest またはアプリケーションでウィンドウのフォーカスを変更したり，操作（右クリックしたり，スクロール・バーを使ったり，ポ

インタをオブジェクト上に移動してショートカット・メニューを表示するなど）を実行できます。

- 選択するオブジェクトがあるウィンドウが別のウィンドウで部分的に隠れている場合は，部分的に隠れているウィンドウが前面に表示されるまで，その上に指差しマークを数秒間置いたままにします。次に，必要なオブジェクトをポイントしてクリックします。ウィンドウを前面に出すのに必要な時間の設定は，［オプション］ダイアログ・ボックスの［一般］表示枠で行えます。
- 選択するオブジェクトがあるウィンドウが別のウィンドウで完全に隠れている場合，またはダイアログ・ボックスがウィンドウの背後に隠れている場合は，必要に応じて左の CTRL キーを押しながらウィンドウを配置します。
- 選択するオブジェクトがあるウィンドウが最小化されている場合は，左の CTRL キーを押したまま，Windows タスク・バーでアプリケーションを右クリックして，ショートカット・メニューから［**元のサイズに戻す**］を選択すると，そのウィンドウを表示できます。
- 特定のイベント（ショートカット・メニューを表示するために右クリックしたり，オブジェクト上にポインタを移動するなど）を実行するだけで，選択するオブジェクトを表示できる場合は，左の CTRL キーを押したままにします。指差しマークが一時的に標準ポインタに変わり，イベントを実行できます。選択するオブジェクトが表示されたら，左の CTRL キーを放します。ポインタが再び指差し型に変わります。

認識プロパティの保守

アプリケーションが変更されると，コンポーネントに含まれるステップのプロパティ値も変更しなければならない場合があります。アプリケーションに含まれるオブジェクトが変更された場合を考えてみます。そのオブジェクトがコンポーネントの一部であれば，引き続き QuickTest が認識できるように値を変更する必要があります。たとえば，ある企業の Web サイトに含まれる「**Contact Us**」というハイパーテキスト・リンクのテキスト文字列が「**Contact My Company**」に変更されたとします。このリンク・プロパティが引き続き QuickTest で正しく認識されるようにするためには，オブジェクト・リポジトリに含まれるこのオブジェクトの詳細を更新する必要があります。

認識プロパティはさまざまな方法で変更できます。ローカル・オブジェクト・リポジトリに格納されているオブジェクトの場合，そのプロパティは［オブジェクト リポジトリ］ウィンドウから直接変更できます。共有オブジェクト・リポジトリに格納されているオブジェクトの場合，オブジェクト・リポジトリ・マネージャでオブジェクトを開いてプロパティを変更できます。また，オブジェクトをローカル・オブジェクト・リポジトリにコピーしてからオブジェクトのプロパティを変更することもできます。

認識プロパティを変更するさまざまな方法の詳細については，次を参照してください。

- 176 ページ「プロパティ値の指定または変更」
- 178 ページ「アプリケーション内のオブジェクトから認識プロパティを更新」
- 181 ページ「テスト・オブジェクトの標準必須プロパティの復元」
- 182 ページ「テスト・オブジェクトの名前の変更」
- 184 ページ「テスト・オブジェクト記述へのプロパティの追加」
- 188 ページ「新規認識プロパティの定義」
- 190 ページ「テスト・オブジェクト記述からのプロパティの削除」
- 191 ページ「序数識別子の指定」

プロパティ値の指定または変更

本項で説明する機能は，ローカル・オブジェクト・リポジトリのオブジェクトに関しては［**オブジェクト リポジトリ**］ウィンドウで，共有オブジェクト・リポジトリのオブジェクトに関しては［**オブジェクト リポジトリ マネージャ**］で利用できます。

テスト・オブジェクト記述に含まれるプロパティの値を指定または変更できます。定数値（単純な値か，正規表現を含んだ定数値）を使って値を指定したり，値をパラメータ化したりできます。また，そのオブジェクトを識別するのに使用するプロパティのセットも変更できます。

オブジェクト・リポジトリに含まれる 1 つ以上のオブジェクトの記述を，アプリケーション内で実際に更新されたオブジェクト・プロパティに基づいて自動的に更新することもできます。詳細については，178 ページ「アプリケーション内のオブジェクトから認識プロパティを更新」を参照してください。

指定した認識プロパティ値を検索して置換することもできます。詳細については，168 ページ「オブジェクト・リポジトリ内のオブジェクトの検索」を参照してください。

注：場合によっては，テスト・オブジェクト・プロパティ値の一部が変更されても，スマート認識機能を使用することで QuickTest がテスト・オブジェクトを認識できることがあります。しかし，特定のテスト・オブジェクトのプロパティ値に変更があったことがわかっている場合は，QuickTest が基本のオブジェクト記述からテスト・オブジェクトを認識できるように，テスト・オブジェクトの定義を修正する必要があります。スマート認識メカニズムの詳細については，第 6 章「オブジェクト認識の設定」を参照してください。

ヒント：オブジェクト・スパイ使用して，テストするアプリケーションのオブジェクトのネイティブ・プロパティと値，またはそれらを示すテスト・オブジェクトの認識プロパティを表示できます。オブジェクト・スパイを開くには，［**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ます。詳細については，119 ページ「オブジェクト・スパイを使用したオブジェクト・プロパティと操作の表示」を参照してください。

認識プロパティ値を指定するには，次の手順を実行します。

1 ［オブジェクト レジストリ］ウィンドウまたはマネージャで，指定したい適切な値のテスト・オブジェクトを選択します。

2 ［**テスト オブジェクトの詳細**］領域で，必要なプロパティの値セルをクリックします。

Tips: ローカル・オブジェクト・リポジトリのテスト・オブジェクトの場合は，テスト・オブジェクトが含まれているステップを右クリックし，［オブジェクトのプロパティ］を選択して，［オブジェクトのプロパティ］ダイアログ・ボックスで次のようにプロパティ値を変更することもできます。

コンポーネントに含まれるすべてのオブジェクトを表示するには，［**リポジトリに表示**］ボタンをクリックします。［オブジェクト リポジトリ］ウィンドウが開き，リポジトリに格納されているすべてのオブジェクトがリポジトリ・ツリーに表示されます。

選択したコンポーネントのオブジェクト・リポジトリを開くには，［**リソース**］＞［**オブジェクト リポジトリ**］を選択する方法や，［**オブジェクト リポジトリ**］ツールバー・ボタンをクリックする方法もあります。

3 次のいずれかの方法でプロパティ値を指定します。

➤ 定数値を指定する場合は，値セルに定数値を入力します。

➤ 値をパラメータ化したり，正規表現を使った定数値を指定する場合は，値セル内のパラメータ化ボタンをクリックします。正規表現を使用して定数値を指定した場合， アイコンが値の隣に表示されます。値のパラメータ化の詳細については，542 ページ「パラメータの使用」を参照してください。

4 定数値を指定した場合は，その定数値が［**テスト オブジェクトの詳細**］領域の［**値**］カラムに表示されます。値をパラメータ化した場合は，パラメータ名が次のいずれかのアイコンとともに［**値**］カラムに表示されます。

パラメータ・アイコン	説明
	プロパティ値が現在はコンポーネント・パラメータであることを示します。
	プロパティ値が現在はローカル・パラメータであることを示します。
	プロパティ値が現在は（共有オブジェクト・リポジトリの）リポジトリ・パラメータであることを示します。

アプリケーション内のオブジェクトから認識プロパティを更新

本項で説明する機能は，ローカル・オブジェクト・リポジトリのオブジェクトに関しては［**オブジェクト リポジトリ**］ウィンドウで，共有オブジェクト・リポジトリのオブジェクトに関しては［**オブジェクト リポジトリ マネージャ**］で利用できます。

オブジェクト・リポジトリ内のオブジェクトを更新するには，アプリケーション内の対応するオブジェクトを選択し，そのオブジェクトのプロパティおよびプロパティ値をアプリケーションから学習し直します。このようにしてテスト・オブジェクトの記述を更新すると，記述のプロパティと値，序数識別子，スマート認識情報など，現在定義されているすべてのプロパティと値が上書きされます。更新されるオブジェクト記述は，［オブジェクトの認識］ダイアログ・ボックスの現在の定義に基づいたものになります。オブジェクト固有のコメントがあれば，保持されます。

これは，オブジェクトをオブジェクト・リポジトリに追加した後でオブジェクトのプロパティに変更があった場合に便利です。QuickTest では，変更があった場合，オブジェクトの記述を更新しないかぎりオブジェクトが認識されないためです。

また，このアプリケーションの開発が完了する前に（[**オブジェクト**] > [**新規テスト オブジェクトの定義**] オプションを使って）オブジェクトを定義した結果，認識プロパティと値の一部がテスト・オブジェクト記述に含まれていなかったり，オブジェクトを認識するのに十分でない場合は，このオプションを使ってそのオブジェクトを更新することもできます。[**新規テスト オブジェクトの定義**] オプションの詳細については，161 ページ「新規テスト・オブジェクトの定義」を参照してください。

注：変更したプロパティ値はそのままにして，元のテスト・オブジェクト記述のプロパティ・セットを復元したいだけなら，[**必須プロパティ セットを復元する**] オプションを使用できます。詳細については，181 ページ「テスト・オブジェクトの標準必須プロパティの復元」を参照してください。

アプリケーション内のオブジェクトから認識プロパティを更新するには，次の手順を実行します。

1 オブジェクト・リポジトリ・ツリーの中で，記述を更新するテスト・オブジェクトを選択します。

2 [**オブジェクト**] > [**アプリケーションから更新**] を選択するか，[**アプリケーションから更新**] ボタンをクリックします。QuickTest が非表示になり，ポインタが指差しマークに変わります。指差しマークの使用方法の詳細については，180 ページ「指差しマークの使用法に関するヒント」を参照してください。

3 オブジェクト・リポジトリでプロパティを更新するオブジェクトをアプリケーション内で見つけ，それをクリックします。オブジェクト・リポジトリ・ツリーの中で選択したテスト・オブジェクトと同じオブジェクト・クラスのオブジェクトを選択する必要があります。

アプリケーション内でオブジェクトをクリックしたときに QuickTest が学習した，オブジェクトの認識に必要なプロパティとプロパティ値に従って，選択したオブジェクトのプロパティとプロパティ値がオブジェクト・リポジトリ内で更新されます。[**テスト オブジェクトの詳細**] 領域のすべてのプロパティおよびプロパティ値が，序数識別子やスマート認識の選択とともに更新されます。すでに入力されているオブジェクト固有のコメントは削除されません。

指差しマークの使用法に関するヒント

- 左の CTRL キーを押したままにすると，指差しマークが標準ポインタに変わります。そのときに，QuickTest またはアプリケーションでウィンドウのフォーカスを変更したり，操作（右クリックしたり，スクロール・バーを使ったり，ポインタをオブジェクト上に移動してショートカット・メニューを表示するなど）を実行できます。
- 選択するオブジェクトがあるウィンドウが別のウィンドウで部分的に隠れている場合は，部分的に隠れているウィンドウが前面に表示されるまで，その上に指差しマークを数秒間置いたままにします。次に，必要なオブジェクトをポイントしてクリックします。ウィンドウを前面に出すのに必要な時間の設定は，［オプション］ダイアログ・ボックスの［一般］表示枠で行えます。
- 選択するオブジェクトがあるウィンドウが別のウィンドウで完全に隠れている場合，またはダイアログ・ボックスがウィンドウの背後に隠れている場合は，必要に応じて左の CTRL キーを押しながらウィンドウを配置します。
- 選択するオブジェクトがあるウィンドウが最小化されている場合は，左の CTRL キーを押したまま，Windows タスク・バーでアプリケーションを右クリックして，ショートカット・メニューから［**元のサイズに戻す**］を選択すると，そのウィンドウを表示できます。
- 特定のイベント（ショートカット・メニューを表示するために右クリックしたり，オブジェクト上にポインタを移動するなど）を実行するだけで，選択するオブジェクトを表示できる場合は，左の CTRL キーを押したままにします。指差しマークが一時的に標準ポインタに変わり，イベントを実行できます。選択するオブジェクトが表示されたら，左の CTRL キーを放します。ポインタが再び指差し型に変わります。

テスト・オブジェクトの標準必須プロパティの復元

本項で説明する機能は，ローカル・オブジェクト・リポジトリのオブジェクトに関しては［**オブジェクト リポジトリ**］ウィンドウで，共有オブジェクト・リポジトリのオブジェクトに関しては［**オブジェクト リポジトリ マネージャ**］で利用できます。

選択したテスト・オブジェクトの標準プロパティを復元できます。標準設定のプロパティを復元すると，オブジェクトを学習したときに［オブジェクトの認識］ダイアログ・ボックスでなされた設定に基づいて，選択したオブジェクト・クラスに定義された必須のプロパティ・セットが復元されます。プロパティを記述に追加したり削除すると，それらの変更内容が上書きされます。ただし，必須プロパティの値が定義または変更された場合は，このオプションを選択しても，それらの値は変更されません。また，標準設定の必須のプロパティ・セットを復元しても，テスト・オブジェクトの序数識別子やスマート認識の設定値は変更されません。

注 :［**必須プロパティ セットを復元する**］は，オブジェクトを学習したときにそのクラスに定義された必須プロパティに設定されたオブジェクト記述プロパティを復元します。［オブジェクトの認識］ダイアログ・ボックスの必須プロパティが現在では，オブジェクトを学習したときのテスト・オブジェクト・クラスに合わなくなり，新しい定義を使いたい場合は，［**アプリケーションから更新**］オプションを使用できます。このオプションは，［オブジェクトの認識］ダイアログ・ボックスの現在の定義に基づいてオブジェクト・プロパティと値を再学習します。詳細については，178 ページ「アプリケーション内のオブジェクトから認識プロパティを更新」を参照してください。

必須のプロパティ・セットを復元するには，次の手順を実行します。

1 オブジェクト・リポジトリ・ツリーの中で，記述を復元するテスト・オブジェクトを選択します。

2［**テスト オブジェクトの詳細**］領域で，［**必須プロパティ セットを復元する**］ボタンをクリックします。

3［**はい**］をクリックして操作を確定します。テスト・オブジェクトの記述プロパティが，オブジェクトを学習した時点で選択されたオブジェクト・クラスの必須プロパティ・セットに復元されます。

テスト・オブジェクトの名前の変更

本項で説明する機能は，ローカル・オブジェクト・リポジトリのオブジェクトに関しては［**オブジェクト リポジトリ**］ウィンドウで，共有オブジェクト・リポジトリのオブジェクトに関しては［**オブジェクト リポジトリ マネージャ**］で利用できます。

アプリケーション内のオブジェクトが変更された場合，または何らかの理由で現在のテスト・オブジェクト名に不満がある場合は，QuickTest によって，格納されているオブジェクトに割り当てられている名前を変更できます。また，ユーザがコンポーネント・ステップの中でテスト・オブジェクトを使用するときに識別しやすいように，テスト・オブジェクトにわかりやすい名前を付けることもできます。

たとえば，グラフィック・アプリケーションのツールバーのすべてのツールが WinObjects としてオブジェクト・リポジトリに保存され，それぞれオブジェクトに ToolChild1，ToolChild2，ToolChild3 という具合に名前が付けられているとします。この場合，すべてのボタンの名前を実際のラベル（Color_Picker，Eraser，Airbrush など）に変更すると，ボタンが識別しやすくなります。

共有オブジェクト・リポジトリを使用している場合は，その共有オブジェクト・リポジトリを使用するすべてのコンポーネントで，当該テスト・オブジェクトが出現するすべての箇所に変更が適用されます。

ローカル・オブジェクト・リポジトリを使用している場合，選択したコンポーネントの中で，当該テスト・オブジェクトが出現するすべての箇所に，変更が適用されます。当該ローカル・オブジェクトに対する操作が，ビジネス・プロセス・テスト内のほかのコンポーネントにも含まれている場合は，該当する各コンポーネントでテスト・オブジェクトの名前を変更する必要があります。

ローカル・オブジェクト・リポジトリのテスト・オブジェクトの名前を変更すると，そのテスト・オブジェクトのすべての出現箇所の名前が自動的に更新されます。共有リポジトリ内のテスト・オブジェクトの名前を変更すると，当該オブジェクト・リポジトリの変更をまだ保存していなくても，同じコンピュータ上で開いている，当該オブジェクト・リポジトリを使用しているすべてのコンポーネントの中の名前が，変更の直後に自動的に更新されます。変更を保存せずにオブジェクト・リポジトリを閉じると，変更時に開いていたすべてのコンポーネントにおいて，変更がロール・バックされます。保存した変更は，当該オブジェクト・リポジトリを使用しているコンポーネントにおいても，それらを開いた直後に自動的に更新されます。

現在別のコンピュータで開いているコンポーネントまたはオブジェクト・リポジトリに，保存した変更内容を読み込んで表示するには，使用しているコンピュータ上でそのオブジェクト・リポジトリを開くか，編集用にロックする必要があります。

ヒント：テスト・オブジェクトのすべての出現箇所のテスト・オブジェクト名を自動的に更新しないようにするには，［オプション］ダイアログ・ボックスの［一般］表示枠（［**ツール**］＞［**オプション**］＞［**一般**］ノード）で［**テスト オブジェクトの名称変更時にテストとコンポーネントのステップを自動的に更新する**］チェック・ボックスをクリアします。このオプションをクリアした場合は，該当するテスト・オブジェクトが使用されているすべてのステップでテスト・オブジェクト名を手作業で変更する必要があります。変更しないとコンポーネントの実行は失敗します。

注：共有オブジェクト・リポジトリ内のテスト・オブジェクト名を変更して変更を保存した場合，その同じ共有オブジェクト・リポジトリを使用する別のコンポーネントを開くと，当該コンポーネントの関連するすべてのステップでテスト・オブジェクト名が更新されます。この処理にはしばらくかかる場合があります。後者のコンポーネントに対する変更を保存すると，名前の変更されたステップが保存されます。しかし，保存せずに後者のコンポーネントを閉じると，次回同じコンポーネントを開いたとき，ステップ内のテスト・オブジェクト名を更新するのに再度時間がかかります。

テスト・オブジェクトの名前を変更するには，次の手順を実行します。

［オブジェクト リポジトリ］ウィンドウまたはマネージャのオブジェクト・リポジトリ・ツリーで，名前を変更するテスト・オブジェクトを選択し，次のいずれかを実行します。

➤ ［**編集**］＞［**名前の変更**］を選択し，ツリー内で選択したノードのテスト・オブジェクトに新しい名前を入力します。次に，ENTER キーを押すか，ほかの場所をクリックしてフォーカスをテスト・オブジェクトから移動します。

➤ F2 キーを押し，テスト・オブジェクトの新しい名前を入力します。

➤ オブジェクト・プロパティ表示枠の［**名前**］ボックスに，テスト・オブジェクトの新しい名前を入力します。次に，ほかの任意の場所をクリックしてフォーカスをオブジェクトから移動します。テスト・オブジェクトに割り当てる名前は，オブジェクト・リポジトリ内の同じクラスおよび階層内で一意である必要があります。オブジェクト名では，大文字と小文字は区別されません。

テスト・オブジェクト記述へのプロパティの追加

本項で説明する機能は，ローカル・オブジェクト・リポジトリのオブジェクトに関しては［**オブジェクト リポジトリ**］ウィンドウで，共有オブジェクト・リポジトリのオブジェクトに関しては［**オブジェクト リポジトリ マネージャ**］で利用できます。

QuickTest でオブジェクトの認識に使用されるプロパティのリストにプロパティを追加できます。QuickTest では，オブジェクト・クラスごとに，特定のテスト・オブジェクトのオブジェクト記述に使用する標準のプロパティ・セットがあります。［プロパティの追加］ダイアログ・ボックスを使用して，テスト・オブジェクト記述に含まれるプロパティを変更できます。

注：また，［プロパティの追加］ダイアログ・ボックスに表示されなくても，有効な認識プロパティであれば，テスト・オブジェクト記述に追加できます。詳細については，188 ページ「新規認識プロパティの定義」を参照してください。

プロパティのリストへの追加は，動的に変化するオブジェクトを対象にコンポーネントを作成し，実行する場合に便利です。頻繁に更新される場合，または動的コンテンツ（データベースなど）によってプロパティ値が設定される場合に，オブジェクトが動的に変化することがあります。

また，QuickTest がオブジェクトの学習中に自動的には学習されなかったプロパティを使ってオブジェクトを参照する場合は，オブジェクトを認識するプロパティを変更することもできます。たとえば，ニュースレターのアーカイブが含まれる Web サイトをテストするとします。アーカイブ・ページには，最新のニュースレターへのハイパーテキスト・リンクに加えて，過去のすべてのニュースレターへのハイパーテキスト・リンクがあります。

このページの最初のハイパーテキスト・リンクのテキストは，最新のニュースレターの変更に伴って変化しますが，常に **current.html** というページにリンクされています。コンポーネントに，常にアーカイブ・ページの最初のハイパーテキスト・リンクをクリックするステップを作成するとします。ニュースは常に変化しているので，このハイパーテキスト・リンクのテキストもそれに応じて変化していきます。そのため，QuickTest が常にこのリンクを見つけられるように，このハイパーテキスト・リンクを識別する方法を変更する必要があります。

Link オブジェクト（ハイパーテキスト・リンク）の標準プロパティは，「**text**」と「**HTML tag**」です。text プロパティは，リンク内のテキストです。HTML tag プロパティは常にリンクを表す「**A**」です。

QuickTest がリンクのテキストではなく，リンク先ページによってオブジェクトを認識できるように，学習したオブジェクトのハイパーテキスト・リンクの標準プロパティを変更できます。「**href**」プロパティを使用すると，「**text**」プロパティによってリンクのテキストでリンクを検査せずに，リンク先ページを検査できます。

ヒント：オブジェクト・スパイを使用して，テストするアプリケーションのオブジェクトのネイティブ・プロパティと値，またはそれらを示すテスト・オブジェクトの認識プロパティを表示できます。［オブジェクト スパイ］を開くには，［**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ます。詳細については，119 ページ「オブジェクト・スパイを使用したオブジェクト・プロパティと操作の表示」を参照してください。

注：また，［オブジェクトの認識］ダイアログ・ボックスを使用すれば，QuickTest が特定のオブジェクト・クラスのオブジェクトを学習する際に学習する一連のプロパティを変更できます。通常，そのような変更内容は，変更後に QuickTest が学習したオブジェクトにのみ反映されます。詳細については，193 ページ「オブジェクト認識の設定」を参照してください。また，［オブジェクトの認識］ダイアログ・ボックスで行った変更は，［**更新モード**］オプションを使用することで，既存のコンポーネント内にあるすべてのオブジェクトの記述に適用できます。詳細については，787 ページ「更新モード・オプションを使ったコンポーネントの更新」を参照してください。

テスト・オブジェクト記述にプロパティを追加するには，次の手順を実行します。

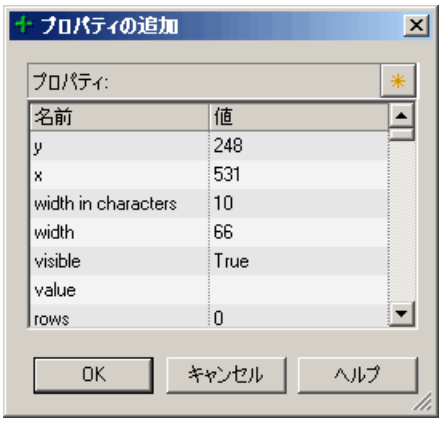
1 ［オブジェクト リポジトリ］ウィンドウまたはマネージャのオブジェクト・リポジトリ・ツリーで，記述を変更するテスト・オブジェクトを選択します。

2 ［**テスト オブジェクトの詳細**］領域で，［**記述プロパティを追加**］ボタンをクリックします。

ヒント：ローカル・オブジェクト・リポジトリ内のテスト・オブジェクトの場合は，必要なテスト・オブジェクトを選択し，［**編集**］＞［**ステップのプロパティ**］＞［**オブジェクトのプロパティ**］を選択して，［**記述プロパティを追加**］ボタンをクリックし，［プロパティの追加］ダイアログ・ボックスで以下の手順を実行することもできます。

［プロパティの追加］ダイアログ・ボックスが開き，オブジェクトの認識に使用できるプロパティ（テスト・オブジェクト記述にまだ含まれていないプロパティ）が一覧表示されます。

各プロパティの値は，［**値**］カラムに表示されます。



注：

- すべてのプロパティの値が表示されるのは，そのオブジェクトを含むアプリケーションが現在開いている場合だけです。アプリケーションが閉じている場合は，オブジェクトの学習時にオブジェクト記述の一部だったプロパティの値のみが表示されます。
- 長いプロパティ値を表示できるように，［プロパティの追加］ダイアログ・ボックスのサイズを変更できます。
-
［**新規プロパティを定義**］ボタンをクリックすると，このプロパティ・リストに有効な認識プロパティを追加できます。詳細については，188 ページ「新規認識プロパティの定義」を参照してください。

---

3 テスト・オブジェクト記述に追加する 1 つ以上のプロパティを選択して［**OK**］をクリックします。また，プロパティをダブルクリックすることで，テスト・オブジェクト記述にプロパティを追加することもできます。プロパティの最初の文字を入力して，パターンに一致するリスト内の最初のプロパティを強調表示させることができます。

ヒント：オブジェクト記述に新しいプロパティを追加した後は，その値を変更できます。オブジェクト・プロパティ値の変更の詳細については，176 ページ「プロパティ値の指定または変更」を参照してください。

新規認識プロパティの定義

本項で説明する機能は，ローカル・オブジェクト・リポジトリのオブジェクトに関しては［オブジェクト リポジトリ］ウィンドウで，共有オブジェクト・リポジトリのオブジェクトに関しては［オブジェクト リポジトリ マネージャ］で利用できます。

有効な認識プロパティであれば，［プロパティの追加］ダイアログ・ボックスに表示されなくても，テスト・オブジェクト記述に追加できます。

たとえば，QuickTest で特定のプロパティを使用してオブジェクトを認識したいが，そのプロパティが［プロパティの追加］ダイアログ・ボックスに表示されていないとします。このような場合は，［プロパティの追加］ダイアログ・ボックスを開いて，そのプロパティをリストに追加することができます。

ヒント：オブジェクト・スパイの［プロパティ］タブを使用すると，選択したオブジェクトに有効な認識プロパティの完全なリストを確認できます。［オブジェクト スパイ］を開くには，［**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ます。詳細については，119 ページ「オブジェクト・スパイを使用したオブジェクト・プロパティと操作の表示」を参照してください。

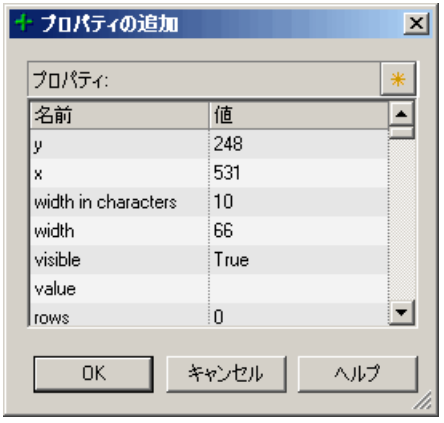
新規の認識プロパティを定義するには，次の手順を実行します。

1 ［オブジェクト リポジトリ］ウィンドウまたはマネージャのオブジェクト・リポジトリ・ツリーで，新しいプロパティを定義するテスト・オブジェクトを選択します。

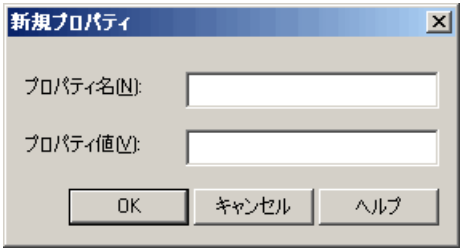
2 ［**テスト オブジェクトの詳細**］領域で，［**記述プロパティを追加**］ボタンをクリックします。

ヒント：ローカル・オブジェクト・リポジトリ内のテスト・オブジェクトの場合は，必要なテスト・オブジェクトを選択し，［**編集**］＞［**ステップのプロパティ**］＞［**オブジェクトのプロパティ**］を選択して，［**記述プロパティを追加**］ボタンをクリックし，［プロパティの追加］ダイアログ・ボックスで以下の手順を実行することもできます。

［プロパティの追加］ダイアログ・ボックスが表示されます。



3 ［**新規プロパティを定義**］ボタンをクリックします。［新規プロパティ］ダイアログ・ボックスが開きます。



4 有効な認識プロパティを指定します。

- ［**プロパティ名**］：プロパティ名を入力します。
- ［**プロパティ値**］：プロパティの値を入力します。

注：有効な認識プロパティを入力してください。無効なプロパティを入力し，オブジェクト記述にこのプロパティを含めると，実行セッションが失敗します。

---

5 ［**OK**］をクリックすることで，プロパティをリストに追加して［新規プロパティ］ダイアログ・ボックスを閉じます。［プロパティの追加］ダイアログ・ボックスの中で，新規プロパティが強調表示されます。

6 新規プロパティが強調表示されている状態で［**OK**］をクリックすると，オブジェクト記述にそのプロパティが追加され，［プロパティの追加］ダイアログ・ボックスが閉じます。

テスト・オブジェクト記述からのプロパティの削除

本項で説明する機能は，ローカル・オブジェクト・リポジトリのオブジェクトに関しては［**オブジェクト リポジトリ**］ウィンドウで，共有オブジェクト・リポジトリのオブジェクトに関しては［**オブジェクト リポジトリ マネージャ**］で利用できます。

記述に含める必要がなくなったプロパティは，テスト・オブジェクトの記述から削除できます。

テスト・オブジェクト記述からプロパティを削除するには，次の手順を実行します。

1 ［オブジェクト リポジトリ］ウィンドウまたはマネージャのオブジェクト・リポジトリ・ツリーで，記述を変更するテスト・オブジェクトを選択します。

2 ［**テスト オブジェクトの詳細**］領域で，テスト・オブジェクト記述から削除するプロパティを 1 つ以上選択します。

ヒント：ローカル・オブジェクト・リポジトリ内のオブジェクトの場合は，必要なテスト・オブジェクトを選択し，［**編集**］＞［**ステップのプロパティ**］＞［**オブジェクトのプロパティ**］を選択して，［オブジェクトのプロパティ］ダイアログ・ボックスで次の手順を実行することもできます。

3［**選択されている記述プロパティを削除**］ボタンをクリックします。選択したプロパティがテスト・オブジェクト記述から削除されます。

序数識別子の指定

本項で説明する機能は，ローカル・オブジェクト・リポジトリのオブジェクトに関しては［**オブジェクト リポジトリ**］ウィンドウで，共有オブジェクト・リポジトリのオブジェクトに関しては［**オブジェクト リポジトリ マネージャ**］で利用できます。

序数識別子は，同じ記述を持つほかのオブジェクト（すべてのプロパティが同じ値のオブジェクト）に対する該当テスト・オブジェクトの相対的な順序または場所を示すために割り当てる数値です。この順序値は，定義されているプロパティがオブジェクトを一意に認識するのに不十分な場合に，一意の記述を作成するための予備的なメカニズムとして使用できます。

序数識別子の詳細については，201 ページ「序数識別子の選択」を参照してください。

序数識別子を指定するには，次の手順を実行します。

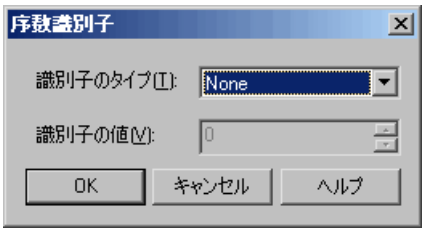
1［オブジェクト リポジトリ］ウィンドウまたはマネージャのオブジェクト・リポジトリ・ツリーで，序数識別子を指定するテスト・オブジェクトを選択します。

2［**テスト オブジェクトの詳細**］領域で，［**序数識別子**］行の下にある［**タイプ、値**］セルの右側にあるセルをクリックします。

ヒント：ローカル・オブジェクト・リポジトリ内のオブジェクトの場合は，必要なテスト・オブジェクトを選択し，［**編集**］＞［**ステップのプロパティ**］＞［**オブジェクトのプロパティ**］を選択し，［**序数識別子**］行の下にある［**タイプ、値**］セルの右側にあるセルをクリックして，［オブジェクトのプロパティ］ダイアログ・ボックスで次の手順を実行することもできます。

---

3 参照ボタンをクリックします。[序数識別子] ダイアログ・ボックスが開きます。



4 [**識別子のタイプ**] ボックスで，次のいずれかのオプションを選択します。

- [**Location**]：親ウィンドウ，フレーム，あるいはダイアログ・ボックス内においてオブジェクトが出現する順序を，その他の記述が同じであるほかのオブジェクトとの相対位置で表します。
- [**Index**]：アプリケーション・コード内においてオブジェクトが出現する順序を，その他の記述が同じであるほかのオブジェクトに対する相対関係で表します。
- [**CreationTime**]（ブラウザ・オブジェクトのみ）：ブラウザが開かれた順序を，その他の記述が同じであるほかのブラウザに対する相対位置で表します。この識別子タイプは，テスト・オブジェクトの学習時に複数のブラウザ・オブジェクトが開いていた場合にのみ使用できます。
- [**None**]：序数識別子は指定しません。これは QuickTest が序数識別子を学習しなかった場合の標準設定値です。

5 [**識別子の値**] ボックスに序数識別子の数値を入力します。

6 [**OK**] をクリックします。選択したオブジェクトの [**テスト オブジェクトの詳細**] 領域の該当する行に，序数識別子が表示されます。

第 6 章

オブジェクト認識の設定

QuickTest では，オブジェクトを学習するときにオブジェクト階層内のオブジェクトを一意に記述するプロパティと値のセットを学習します。多くの場合，この記述で，QuickTest が実行セッション中にオブジェクトを十分に識別できます。

特定のオブジェクト・クラスを表す記述が，アプリケーションのオブジェクトを最も論理的に記述するものでないことが判明した場合や，オブジェクト記述の中のプロパティの値が頻繁に変わることが予想される場合には，QuickTest によるオブジェクトの学習方法，識別方法を設定できます。また，ユーザ定義オブジェクトを標準のテスト・オブジェクト・クラスに割り当て，QuickTest によるユーザ定義オブジェクト・クラスのオブジェクトを学習する方法も設定できます。

本章の内容

- オブジェクトの認識の設定について（193 ページ）
- ［オブジェクトの認識］ダイアログ・ボックスについて（195 ページ）
- スマート認識の設定（209 ページ）
- ユーザ定義のテスト・オブジェクト・クラスの割り当て（219 ページ）

オブジェクトの認識の設定について

QuickTest には，各テスト・オブジェクトに対して学習するプロパティのセットがあらかじめ用意されています。学習したオブジェクトを一意に識別するのにこれらの必須プロパティ値では十分でない場合，QuickTest は何らかの補足プロパティまたは序数識別子を追加して，一意の記述を作成します。

「**必須プロパティ**」とは，QuickTest が特定のテスト・オブジェクト・クラスについて必ず学習するプロパティです。

「**補足プロパティ**」とは，QuickTest がアプリケーションの特定のオブジェクトについて学習する必須プロパティでは一意の記述を作成するのに不十分である場合にだけ QuickTest が学習するプロパティです。1 つのオブジェクト・クラスに複数の補足プロパティが定義されている場合，QuickTest は補足プロパティを 1 つずつ順番に学習し，オブジェクトの一意の記述ができたところで学習を止めます。QuickTest が学習した補足プロパティはテスト・オブジェクトの記述に追加されます。

注：すべての定義済みの必須および補足プロパティを組み合わせても一意のテスト・オブジェクト記述を作成するのに十分ではない場合，QuickTest は選択された序数識別子の値も学習します。詳細については，201 ページ「序数識別子の選択」を参照してください。

コンポーネントの実行の際，QuickTest は，学習した（序数識別子のない）記述に一致するオブジェクトを検索します。記述に一致するオブジェクトが見つからない場合や，複数のオブジェクトが記述に一致する場合，QuickTest は**スマート認識**メカニズム（有効な場合）でオブジェクトを識別します。多くの場合，スマート認識定義が存在すれば，学習した記述が 1 つ以上のプロパティ値が変更されたことで識別に失敗する場合に，QuickTest によるオブジェクトの識別に役立ちます。テスト・オブジェクト記述は，スマート認識メカニズムでもオブジェクトの候補を 1 つに絞り込めない場合にかぎり，序数識別子と組み合わせて使われます。

［オブジェクトの認識］ダイアログ・ボックス（［**ツール**］＞［**オブジェクトの認識**］）では，QuickTest がアプリケーションの中のオブジェクトの記述を学習するのに使う，必須プロパティ，補足プロパティ，および序数識別子プロパティを設定できるほか，スマート認識メカニズムの有効化と設定が可能です。

また，新しいユーザ定義クラスを設定し，それらを既存のテスト・オブジェクト・クラスにマップして，コンポーネント実行時にユーザ定義クラスからオブジェクトを QuickTest が認識するようにもできます。

[オブジェクトの認識] ダイアログ・ボックスについて

[オブジェクトの認識] ダイアログ・ボックスのメイン画面を使って，必須および補足プロパティの設定，序数識別子の選択，および各テスト・オブジェクトに対するスマート認識メカニズムを有効にするかどうかを指定します。

[オブジェクトの認識] ダイアログ・ボックスから，ユーザ定義オブジェクト・クラスの定義とそれらの Standard Windows オブジェクト・クラスへの割り当ても行えます。また，選択した環境の［**テスト オブジェクト クラス**］リストに表示される任意のオブジェクトにスマート認識メカニズムを設定することもできます。

注 :

- ［オブジェクトの認識］ダイアログ・ボックスで加えた変更は，すでにオブジェクト・リポジトリに追加されたオブジェクトには影響しません。
- WinMenu，VbLabel，および VbToolbar オブジェクトなど，特定のテスト・オブジェクトから学習するスマート認識プロパティは設定できません。したがって，これらのオブジェクトには選択した環境の［**テスト オブジェクト クラス**］リストが含まれません。

詳細については，次を参照してください。

- 195 ページ「必須および補足プロパティの設定」
- 201 ページ「序数識別子の選択」
- 206 ページ「スマート認識の有効化と無効化」
- 207 ページ「テスト・オブジェクトの標準設定のオブジェクト認識設定の復元」
- 208 ページ「オブジェクト認識設定用の自動スクリプトの生成」

必須および補足プロパティの設定

QuickTest が特定のオブジェクト・クラスのために使う記述が，アプリケーションのオブジェクトを記述するために十分には論理的ではないことが判明した場合や，オブジェクト記述の中で現在使われているプロパティの値が変わることが予想される場合には，そのクラスのオブジェクトを学習するときに QuickTest が学習する必須および補足プロパティを変更できます。

実行セッション中，QuickTest はテスト・オブジェクト記述中のすべてのプロパティに一致するオブジェクトを探します。このとき，必須プロパティとして学習したものと補足プロパティとして学習したものを区別しません。

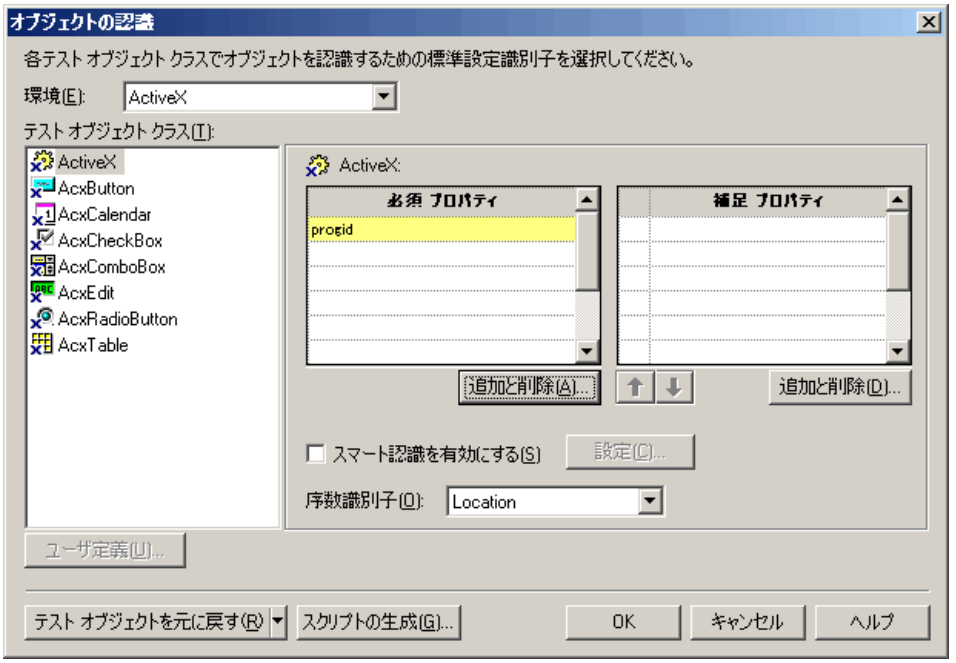
たとえば，Web Image オブジェクトの標準の必須プロパティは，**alt**，**html tag**，および **image type** プロパティです。標準の補足プロパティは定義されていません。Web サイトに複数の広告を循環して表示するいくつかの広告枠があるものとします。これらの広告枠のそれぞれのイメージをクリックするコンポーネントを作成するものとします。

しかし，それぞれの広告イメージの **alt** 値は異なるため，コンポーネントを作成すると 1 つの **alt** 値が追加され，そしてコンポーネントを実行するとほとんどの場合，別の **alt** 値がキャプチャされるため，テスト実行が失敗することになります。この場合，Web Image 必須プロパティ・リストから **alt** プロパティを削除できます。その代わり，サイトの特定の広告枠に表示される各広告イメージは，イメージの **name** プロパティの値が同じなので，必須プロパティにその **name** プロパティを追加して，QuickTest が一意にオブジェクトを識別できるようにします。

また，ページの複数の場所に表示される Web 画像に（たとえば，ロゴがページの一番上と下に表示されるなど），Web デザイナによって Image タグに特別な **ID** プロパティが追加されたとします。ページに一度だけ表示されるイメージであれば，一意の記述を作成するには必須プロパティで十分ですが，同じページにイメージが複数回表示される場合には，QuickTest に **ID** プロパティも学習させたいところです。このためには，**ID** プロパティを補足プロパティとして追加して，一意のテスト・オブジェクト記述の作成に必要な場合にだけ QuickTest に **ID** プロパティを学習させるようにします。

テスト・オブジェクト・クラスのための必須および補足プロパティを作成するには，次の手順を実行します。

1 ［**ツール**］ > ［**オブジェクトの認識**］を選択します。［オブジェクトの認識］ダイアログ・ボックスが表示されます。



2 ［**環境**］リストで適切な環境を選択します。選択した環境に関連付けられているテスト・オブジェクト・クラスが［**テスト オブジェクト クラス**］リストに

アルファベット順に表示されます（標準の Windows では，ユーザ定義のオブジェクトはリストの下部に表示されます）。

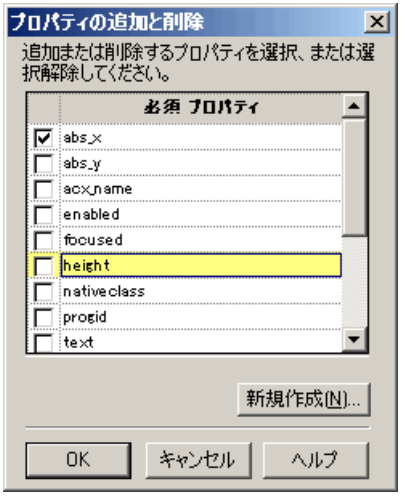
注：

- ［**環境**］リストに含まれている環境は，読み込まれたアドインに対応するものです。アドインの読み込みの詳細については，『**HP QuickTest Professional アドイン・ガイド**』の QuickTest アドインの読み込みに関する項を参照してください。
- ［**環境**］リストには，ユーザまたはサードパーティがアドイン拡張機能を使って開発した追加の環境が含まれていることもあります。

---

3 ［**テスト オブジェクト クラス**］リストで，設定するテスト・オブジェクト・クラスを選択します。

4 ［**必須プロパティ**］リストで，［**追加と削除**］をクリックします。必須プロパティのための［プロパティの追加と削除］ダイアログ・ボックスが表示されます。



5 必須プロパティ・リストに含めるプロパティを選択します。また，リストから削除するプロパティを消去します。

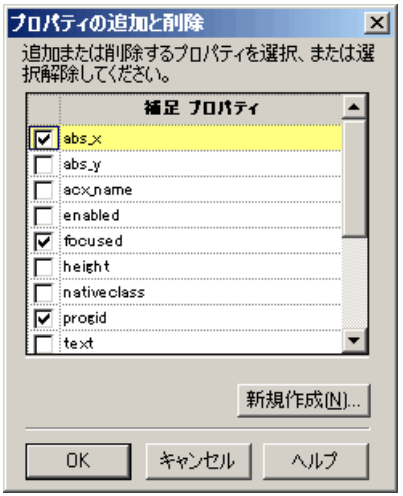
注：同じプロパティを必須と補足の両方のプロパティ・リストに含めることはできません。

［**新規作成**］をクリックして，表示されるダイアログ・ボックスに有効なプロパティ名を指定することにより，新規プロパティを指定できます。

ヒント：attribute/ **＜プロパティ名＞**の形式を使用して，Web オブジェクトに使用可能なプロパティの集合にプロパティ名を追加することもできます。これを行うためには，［**新規作成**］をクリックします。［新規プロパティ］ダイアログ・ボックスが開きます。有効なプロパティを，attribute/ **＜プロパティ名＞**の形式で入力して，［**OK**］をクリックします。新しいプロパティが［**必須プロパティ**］リストに追加されます。たとえば，MyColor というプロパティを追加するには，attribute/MyColor と入力します。

6［**OK**］をクリックし，［プロパティの追加と削除］ダイアログ・ボックスを閉じます。更新された必須プロパティの集合が［**必須プロパティ**］リストに表示されます。

7 ［**補足プロパティ**］リストで，［**追加と削除**］をクリックします。必須プロパティのための［プロパティの追加と削除］ダイアログ・ボックスが表示されます。



8 補足プロパティ・リストに含めるプロパティを選択します。また，リストから削除するプロパティを消去します。

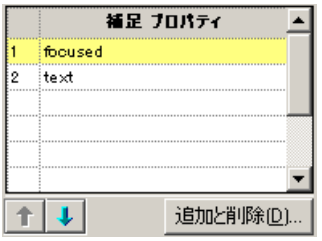
注：同じプロパティを必須と補足の両方のプロパティ・リストに含めることはできません。

［**新規作成**］をクリックして，表示されるダイアログ・ボックスに有効なプロパティ名を指定することにより，新規プロパティを指定できます。

ヒント：attribute/ **＜プロパティ名＞**の形式を使用して，Web オブジェクトに使用可能なプロパティの集合にプロパティ名を追加することもできます。これを行うためには，［**新規作成**］をクリックします。［新規プロパティ］ダイアログ・ボックスが開きます。有効なプロパティを，attribute/ **＜プロパティ名＞**の形式で入力して，［**OK**］をクリックします。新規プロパティが［**補足プロパティ**］リストに追加されます。たとえば，MyColor というプロパティを追加するには，attribute/MyColor と入力します。

---

9 [**OK**] をクリックし，[プロパティの追加と削除] ダイアログ・ボックスを閉じます。プロパティが [補足プロパティ] リストに表示されます。



10 上向き矢印と下向き矢印を使って，補足プロパティの順序を指定します。QuickTest ではオブジェクトを学習するときに一意のオブジェクト記述を作成するために補足プロパティが必要な場合，QuickTest は，一意の記述を作成するのに十分な情報が得られるまで，[補足プロパティ] リストでの順序に従って，記述に補足プロパティを 1 つずつ追加していきます。

序数識別子の選択

[オブジェクトの認識] ダイアログ・ボックスで指定した必須プロパティと補足プロパティを学習するのに加え，QuickTest は予備的に各テスト・オブジェクトの序数識別子も学習できます。[**序数識別子**] は，同じ記述を持つオブジェクト（必須および補足プロパティ・リストで指定されているすべてのプロパティの値が同じオブジェクト）を区別するために，ほかのオブジェクトとの相対的な順番を表す数値を割り当てます。この順位の値によって QuickTest は，必須および補足プロパティでは一意の記述を作成するのに不十分な場合でも，一意の記述を作成できます。

割り当てられた序数プロパティ値は相対的な値であり，QuickTest がオブジェクトを学習したときに表示されるほかのオブジェクトとの関連でのみ正確です。そのため，アプリケーション・ページまたは画面のレイアウトや構成が変わると，オブジェクト自体には少しも変化がなくても，その値が変わってしまう可能性があります。そのため，QuickTest はすべての利用可能な必須および補足プロパティを使っても一意の記述を作成できない場合にだけ，このバックアップ序数識別子の値を学習します。

さらに，QuickTest が序数識別子を学習しても，実行セッション中にその識別子を使うのは，学習した記述およびスマート認識メカニズムがアプリケーションのオブジェクトを識別するのに十分でない場合だけです。QuickTest がほかの認

識プロパティを使用して実行セッション中にオブジェクトを識別できれば，序数識別子は無視されます。

QuickTest では，次の種類の序数識別子を使用してオブジェクトを識別できます。

- ［**Index**］：アプリケーション・コード内においてオブジェクトが出現する順序を，その他の記述が同じであるほかのオブジェクトに対する相対関係で表します。詳細については，203 ページ「インデックス・プロパティを使用したオブジェクトの識別」を参照してください。
- ［**Location**］：親ウィンドウ，フレーム，あるいはダイアログ・ボックス内においてオブジェクトが出現する順序を，その他の記述が同じであるほかのオブジェクトとの相対位置で表します。詳細については，203 ページ「場所プロパティを使用したオブジェクトの識別」を参照してください。
- ［**CreationTime**］（ブラウザ・オブジェクトのみ）：ブラウザが開かれた順序を，その他の記述が同じであるほかのブラウザに対する相対位置で表します。詳細については，205 ページ「CreationTime プロパティを使用したオブジェクトの識別」を参照してください。

標準設定では，テスト・オブジェクト・クラスごとに序数識別子のタイプが存在します。標準設定の序数識別子を変更するには，［**序数識別子**］ボックスから，目的のタイプを選択できます。

ヒント：記録中に，必須および補足プロパティを使って一意のテスト・オブジェクト記述を作成できた場合，QuickTest は序数識別子を学習しません。［オブジェクトのプロパティ］または［オブジェクト リポジトリ］ダイアログ・ボックスの［**追加と削除**］オプションを使うと，後からオブジェクトの認識プロパティに序数識別子を追加できます。詳細については，第 5 章「オブジェクト・リポジトリ内のテスト・オブジェクトの管理」を参照してください。

インデックス・プロパティを使用したオブジェクトの識別

オブジェクトの学習中，QuickTest はオブジェクトを一意に識別できるように，テスト・オブジェクトの**インデックス**・プロパティに値を割り当てることができます。この値は，ソース・コード内のオブジェクトの順番に基づいています。最初の番号は 0 です。

インデックス・プロパティ値は，各オブジェクトに固有の値です。このため，ある WebEdit テスト・オブジェクトを記述するのに Index:=3 を使用すると，QuickTest はページ内の 4 番目の WebEdit を検索します。一方，WebElement オブジェクトを記述するのに Index:=3 を使用すると，WebElement オブジェクトはすべての Web オブジェクトに該当するため，QuickTest はタイプに関係なくページ内の 4 番目の Web オブジェクトを検索します。

たとえば，次のオブジェクトを含んだ次のページがあるとします。

- Apple という名前の画像
- UserName という名前の画像
- UserName という名前の WebEdit オブジェクト
- Password という名前の画像
- Password という名前の WebEdit オブジェクト

次のステートメントは，リストの 3 番目の項目を参照します。その項目が，ページ内で UserName という名前を持つ最初の WebEdit オブジェクトだからです。

```
WebEdit("Name:=UserName", "Index:=0")
```

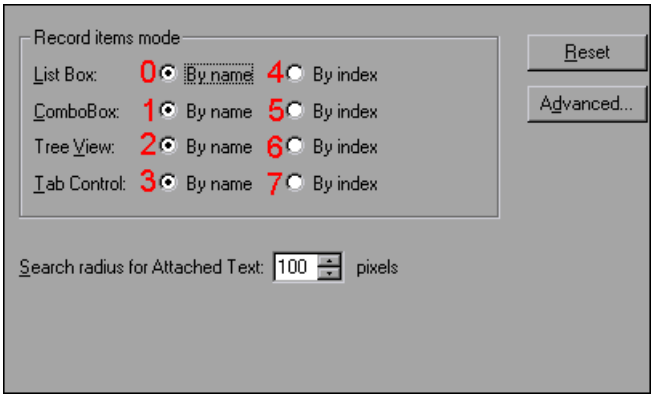
一方，次のステートメントは，リストの 2 番目の項目を参照します。その項目が，UserName という名前を持つ最初の任意のタイプ（WebElement）のオブジェクトだからです。

```
WebElement("Name:=UserName", "Index:=0")
```

場所プロパティを使用したオブジェクトの識別

オブジェクトの学習中，QuickTest はオブジェクトを一意に識別できるように，テスト・オブジェクトの**場所**プロパティに値を割り当てることができます。この値は，ウィンドウ，フレーム，ダイアログ・ボックス内に現れる同じプロパティを持つほかのオブジェクトとの相対的な順番に基づいて決まります。最初のオブジェクトの場合，値は 0 です。値はカラム内で上から下に，そして左から右への順序で割り当てられます。

次の例では，ダイアログ・ボックス内のラジオ・ボタンは，**位置**のプロパティに従って番号が付けられています。



場所プロパティ値は，各オブジェクトに固有の値です。このため，ある WinButton テスト・オブジェクトを記述するのに Location:=3 を使用すると，QuickTest は 4 番目の WinButton をページ内の上から下，左から右に検索します。一方，WinObject オブジェクトを記述するのに Location:=3 を使用すると，WinObject オブジェクトはすべての標準オブジェクトに該当するので，QuickTest はタイプに関係なくページ内の 4 番目の標準オブジェクトをページ内の上から下，左から右に検索します。

たとえば，次のオブジェクトを含んだ次のダイアログ・ボックスがあるとします。

- OK という名前のボタン・オブジェクト
- Add/Remove という名前のボタン・オブジェクト
- Add/Remove という名前のチェック・ボックス・オブジェクト
- Help という名前のボタン・オブジェクト
- Check spelling という名前のチェック・ボックス・オブジェクト

次のステートメントは，リストの 3 番目の項目を参照します。その項目がページ内で Add/Remove という名前を持つ最初のチェック・ボックス・オブジェクトだからです。

```
WinCheckBox("Name:=Add/Remove", "Location:=0")
```

一方，次のステートメントは，リストの 2 番目の項目を参照します。その項目が，Add/Remove という名前を持つ最初の任意のタイプ（WinObject）のオブジェクトだからです。

```
WinObject("Name:=Add/Remove", "Location:=0")
```

CreationTime プロパティを使用したオブジェクトの識別

ブラウザ・オブジェクトの学習中に，QuickTest は **CreationTime** 認識プロパティに値を割り当てます。この値は，開いているほかのブラウザに対して，そのブラウザが開かれた相対的な順番を示します。最初に開かれたブラウザは，CreationTime = 0 という値を受け取ります。

実行セッション中，テスト・オブジェクトの記述だけに基づいて Browser オブジェクトを一意に識別できないとき，QuickTest はブラウザが開いた順番を確認し，**CreationTime** プロパティを使用して正しいブラウザ・オブジェクトを識別します。

たとえば，QuickTest が 9:01 pm，9:03 pm，および 9:05 pm に開かれた 3 つのブラウザを学習すると，QuickTest は 9:01 pm のブラウザには CreationTime = 0，9:03 pm のブラウザには CreationTime = 1，9:05 pm のブラウザには CreationTime = 2 というように CreationTime 値を割り当てます。

10:30 pm にコンポーネントとこれらのブラウザ・オブジェクトを実行すると，ブラウザは 10:31 pm，10:33 pm，および 10:34 pm に開かれると推定されます。QuickTest は 10:31 pm のブラウザが CreationTime = 0 のブラウザ・テスト・オブジェクト，10:33 pm のブラウザが CreationTime = 1 のテスト・オブジェクト，10:34 pm のブラウザが CreationTime = 2 のテスト・オブジェクトであると識別します。

開いているブラウザが複数ある場合，CreationTime の最も低いものが最初に開いたものであり，最も高いものが最後に開いたものになります。たとえば，3 つ以上のブラウザが開いている場合，CreationTime = 2 のブラウザは 3 番目に開いたブラウザです。セッションの記録中に 7 つのブラウザを開いた場合，CreationTime = 6 のブラウザが最後に開いたブラウザです。

特定の CreationTime 値のブラウザ・オブジェクトを対象にステップが作成されても，実行セッションでこの CreationTime 値のブラウザが開いていないと，そのステップは CreationTime 値の最も高いブラウザで実行されます。たとえば，ステップが CreationTime = 6 のブラウザ・オブジェクトを対象に作成されても，実行セッションで CreationTime = 0 と CreationTime = 1 の 2 つのブラウザしか開いていないと，そのステップは最後に開いたブラウザで（この例では CreationTime = 1）実行されます。この例では，CreationTime = 1 のブラウザになります。

注：セッション中の特定の時間に使用できる CreationTime 値は連番になっていない可能性があります。たとえば，記録または実行セッション中に 6 つのブラウザを開いたとして，セッション中にそのうち 2 番目と 4 番目のブラウザ（CreationTime 値 1 と 3）を閉じたとすると，セッションの最後で開いているブラウザは CreationTime 値が 0，2，4，5 のブラウザになります。

スマート認識の有効化と無効化

特定のテスト・オブジェクト・クラスの［**スマート認識を有効にする**］チェック・ボックスを選択すると，［スマート認識プロパティ］ダイアログ・ボックスでオブジェクトの基本フィルタまたはオプション・フィルタ，あるいはその両方のプロパティで指定されているすべてのプロパティ値が QuickTest によって学習されます。

標準では，一部のテスト・オブジェクトはすでにスマート認識が設定されており，ほかは設定されていません。標準で設定されているものは，［**スマート認識を有効にする**］チェック・ボックスも標準で選択されています。

スマート認識設定が定義されているテスト・オブジェクト・クラスのみスマート認識メカニズムを有効にします。しかし，あるテスト・オブジェクト・クラスのスマート認識設定を定義した場合でも，スマート認識プロパティ値を学習したくないこともあります。スマート認識プロパティを学習しない場合は，［**スマート認識を有効にする**］チェック・ボックスをクリアします。

注 : オブジェクトのスマート認識プロパティを学習するように設定してある場合でも，［オブジェクトのプロパティ］または［オブジェクト リポジトリ］ダイアログ・ボックスで，特定のオブジェクトに対するスマート認識機能の使用を無効にできます。また，詳細については，第 5 章「オブジェクト・リポジトリ内のテスト・オブジェクトの管理」を参照してください。

ただし，スマート認識プロパティを学習しなければ，オブジェクトに対して後でスマート認識メカニズムを有効にすることはできません。

スマート認識メカニズムの詳細については，209 ページ「スマート認識の設定」を参照してください。

テスト・オブジェクトの標準設定のオブジェクト認識設定の復元

ロードされているすべての環境，現在の環境のみ，または選択したテスト・オブジェクトの，オブジェクトの認識設定およびスマート認識プロパティ設定を，標準設定に戻すことができます。

リセットできるのは組み込みオブジェクトのプロパティのみです。Standard Windows 環境の設定をリセットすると，ユーザ定義のオブジェクトも削除されます。ユーザ定義オブジェクトの詳細については，219 ページ「ユーザ定義のテスト・オブジェクト・クラスの割り当て」を参照してください。

注 : ［オブジェクトの認識］ダイアログ・ボックスの［環境］ボックスには，現在ロードされている環境のみが一覧表示されます。

標準では［**テスト オブジェクトを元に戻す**］ボタンが表示されますが，下矢印をクリックして，次のいずれかのオプションを選択することができます。

- ［**テスト オブジェクトを元に戻す**］：選択したテスト・オブジェクトの設定をリセットして，システム標準に戻します。
- ［**環境を元に戻す**］：現在の環境のすべてのテスト・オブジェクトの設定をリセットして，システム標準に戻します。
- ［**すべて元に戻す**］：現在ロードされている環境のすべての設定をリセットして，システム標準に戻します。

オブジェクト認識設定用の自動スクリプトの生成

［**スクリプトの生成**］ボタンをクリックして，現在のオブジェクトの認識設定を含む自動スクリプトを生成できます。詳細については，911 ページ「QuickTest 操作のオートメーション」，または『**QuickTest Professional Automation Object Model Reference** 』（［**ヘルプ**］＞［**QuickTest Professional ヘルプ**］＞［**HP QuickTest Professional 詳細リファレンス**］＞［**HP QuickTest Professional オートメーション オブジェクト モデル**］）を参照してください。

スマート認識の設定

スマート認識プロパティを設定すれば，学習したオブジェクト記述に含まれるプロパティの一部が変更されていても，QuickTest がアプリケーションのオブジェクトを識別するのに役立ちます。

QuickTest は，学習した記述を使ってオブジェクトを識別するとき，記述中のすべてのプロパティ値と一致するオブジェクトを検索します。ほとんどの場合，この記述はオブジェクトを識別する最も簡単な方法です。そしてオブジェクトの主要なプロパティが変更されないかぎり，この方法は有効です。

QuickTest が学習したオブジェクト記述に一致するオブジェクトを見つけられない場合や，複数のオブジェクトが記述に適合する場合，QuickTest は学習した記述を無視し，スマート認識メカニズムを使ってオブジェクトの識別を試みます。

スマート認識メカニズムはもう少し複雑ですが，柔軟性が優れています。このため，スマート認識定義を適格に設定すれば，学習した記述では識別できないときに，QuickTest がオブジェクト（存在していれば）を識別するのに役立ちます。

スマート認識メカニズムは次の 2 タイプのプロパティを使います。

- ［**基本フィルタ プロパティ**］: 特定のテスト・オブジェクト・クラスの最も基本的なプロパティです。その値は，元のオブジェクトの根本的な部分を変えなければ変わりません。たとえば，Web リンクのタグが <A> から何か別の値に変わった場合は，同じオブジェクトとはいえません。
- ［**オプション フィルタ プロパティ**］: 特定のクラスのオブジェクトを識別するのに役立つ別のプロパティです。これらのプロパティは通常は変わらないとみなされます。該当しなくなった場合には無視できます。

スマート認識の処理過程について

QuickTest が実行セッション中に（学習した記述を使ってオブジェクトを識別できなかったために）スマート認識メカニズムに切り替わると，スマート認識は次のプロセスでオブジェクトを識別します。

1 QuickTest は学習したテスト・オブジェクト記述を「忘れ」，［基本フィルタのプロパティ］リストに定義されているすべてのプロパティに適合するオブジェクト（親オブジェクト内のオブジェクト）を含んだ新しい**オブジェクト候補**リストを作成します。

2 QuickTest は，［オプション フィルタのプロパティ］リストの最初のプロパティに適合しないオブジェクトをオブジェクト候補リストからすべて除外します。残りのオブジェクトが新しいオブジェクト候補リストになります。

3 QuickTest が新しいオブジェクト候補リストを評価します。

- ➤ 新しいオブジェクト候補リストに，まだ複数のオブジェクトがある場合，QuickTest はこの新しい（より小さい）オブジェクト候補リストを使って，リスト中の次のオプション・フィルタ・プロパティを使って手順 2 を繰り返します。
- ➤ 新しくできたオブジェクト候補リストが空の場合，QuickTest はこのオプション・フィルタ・プロパティを無視し，前のオブジェクト候補リストに戻って手順 2 をリストの次のオプション・フィルタ・プロパティを使って繰り返します。
- ➤ オブジェクト候補リストにオブジェクトが 1 つだけ含まれている場合，QuickTest はそれが識別されたオブジェクトであると判断し，そのオブジェクトを含んでいるステートメントを実行します。

4 QuickTest は 1 つのオブジェクトを識別するか，オプション・フィルタ・プロパティを使い果たすまで，手順 2 と 3 で説明した処理を実行し続けます。

スマート認識の除外処理完了後も，QuickTest がまだオブジェクトを識別できない場合は，QuickTest は学習した記述に加え，序数識別子を使ってオブジェクトを識別します。

学習したスクリプトと序数識別子の組み合わせでもオブジェクトを識別するのに不十分な場合，QuickTest は実行セッションを中止し，実行エラー・メッセージを表示します。を参照してください。

テスト結果に含まれるスマート認識情報の参照

学習した記述を使っても QuickTest が指定されたオブジェクトを一度では識別できず，しかもオブジェクトに対してスマート認識定義が定義されている（そして有効である）場合，QuickTest はスマート認識メカニズムを使ってオブジェクトの識別を試みます。

QuickTest が学習した記述では一致するオブジェクトを見つけられず，スマート認識でオブジェクトを見つけるのに成功した場合，このステップはテスト結果で**警告**ステータスが割り当てられ，ステップの詳細結果にスマート認識メカニズムが使用されたことが示されます。

スマート認識メカニズムでオブジェクトを識別できない場合，QuickTest は学習した記述に加え，序数識別子を使ってオブジェクトを識別します。それでもオブジェクトが識別されない場合は，コンポーネントは失敗し，結果に通常の失敗ステップが表示されます。

詳細については，723 ページ「テスト結果に含まれるスマート認識情報の分析」を参照してください。

スマート認識の処理過程の例

以下では，あるオブジェクトの認識の過程を見ていきます。

コンポーネント中に次のステートメントがあるものとします。

```
Browser("Mercury Tours").Page("Mercury Tours").Image("Login").Click 22,17
```

コンポーネントを作成したとき，QuickTest は Login 画像について次のオブジェクト記述を学習しました。

名前	値
⊟ 記述プロパティ	
image type	Image Button
html tag	INPUT
alt	Login

しかし，コンポーネントを作成した後で，ページに 2 つ目のログイン・ボタン（Web サイトの VIP セクションにログインするためのもの）を追加したため，Web デザイナは元のログイン・ボタンの **alt** タグを basic login に変えました。

Web Image オブジェクトの標準の記述（**alt**，**html tag**，**image type**）はサイト内のほとんどの画像に使えますが，もはやログインの画像には使えません。その画像の **alt** プロパティが学習した記述とは一致しないからです。したがって，コンポーネントを実行すると，QuickTest は学習した記述に基づいてログイン・ボタンを識別することができません。しかし，QuickTest はスマート認識定義を使って，ログイン・ボタンをうまく識別できました。

下の例では，QuickTest がスマート認識を使って Login オブジェクトを見つける過程を示します。

1 Web Image オブジェクトに対するスマート認識定義によれば，QuickTest は Login 画像を学習したときに，次のプロパティの値を学習しています。

学習した値は以下のとおりです。

基本フィルタのプロパティ

プロパティ	[値]
html tag	INPUT

オプション・フィルタのプロパティ

プロパティ	[値]
alt	Login
image type	Image Button
name	login
file name	login.gif
class	<null>
visible	1

2 QuickTest は Mercury Tours ページの基本フィルタ・プロパティ定義（**html tag** = INPUT）に適合する 5 つのオブジェクトを識別することによってスマート認識処理を開始します。QuickTest はこれらをオブジェクト候補と考え，［**オプション フィルタのプロパティ**］リストを使ってオブジェクトの確認を開始します。

3 QuickTest は各オブジェクト候補の **alt** プロパティを確認しますが，どれも **alt** の値が Login ではありません。したがって，QuickTest はこのプロパティを無視し，次に移ります。

4 QuickTest は各オブジェクト候補の **image type** プロパティを確認しますが，どれも **image type** の値が Image Button ではありません。したがって，QuickTest はこのプロパティを無視し，次に移ります。

5 QuickTest は各オブジェクト候補の **name** プロパティを確認し，2 つのオブジェクト（基本および VIP ログイン・ボタン）の name が login であることを知ります。QuickTest はほかの 3 つのオブジェクトをリストから除外し，これら 2 つのログイン・ボタンを新しいオブジェクト候補とします。

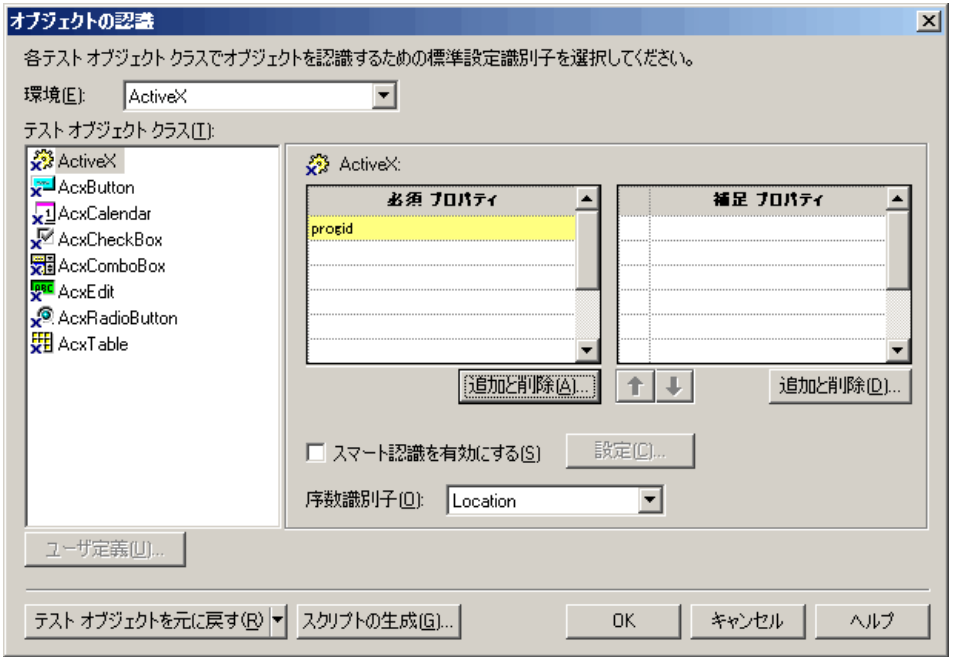
6 QuickTest は残った 2 つのオブジェクト候補の **file name** プロパティを確認します。ファイル名が login.gif であるのオブジェクトは 1 つだけなので，QuickTest はログイン・ボタンを見つけたと正しく結論を出し，それをクリックします。

スマート認識定義の設定手順

［オブジェクトの認識］ダイアログ・ボックスからアクセスできる［スマート認識プロパティ］ダイアログ・ボックスを使って，テスト・オブジェクト・クラスのスマート認識定義を設定できます。

スマート認識のプロパティを設定するには，次の手順を実行します。

1 ［**ツール**］＞［**オブジェクトの認識**］を選択します。［オブジェクトの認識］ダイアログ・ボックスが表示されます。



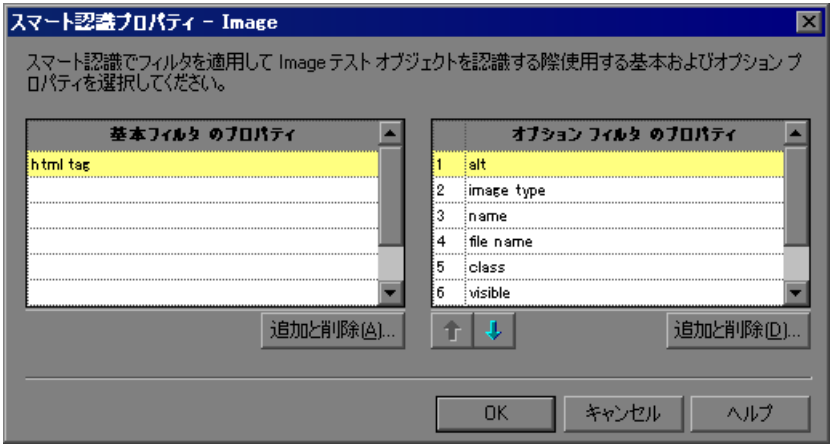
2 ［**環境**］リストで適切な環境を選択します。選択した環境に関連付けられているテスト・オブジェクト・クラスが［**テスト オブジェクト クラス**］リストに表示されます。

注：［環境］リストに含まれている環境は，読み込まれているアドインに対応しています。アドインの読み込みの詳細については，『**HP QuickTest Professional アドイン・ガイド**』の QuickTest アドインの読み込みに関する項を参照してください。

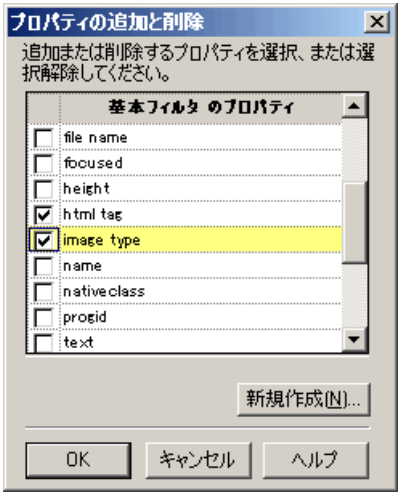
---

3 設定するテスト・オブジェクト・クラスを選択します。

4 ［**スマート認識を有効にする**］チェック・ボックスの横にある［**設定**］ボタンをクリックします。［**設定**］ボタンは，［**スマート認識を有効にする**］オプションが選択されている場合のみ有効になります。［スマート認識プロパティ］ダイアログ・ボックスが表示されます。



5［**基本フィルタのプロパティ**］リストで，［**追加と削除**］をクリックします。基本フィルタ・プロパティのための［プロパティの追加と削除］ダイアログ・ボックスが表示されます。



6［**基本フィルタのプロパティ**］リストに含めるプロパティを選択します。また，リストから削除するプロパティを消去します。

注：同じプロパティを基本とオプションの両方のプロパティ・リストに含めることはできません。

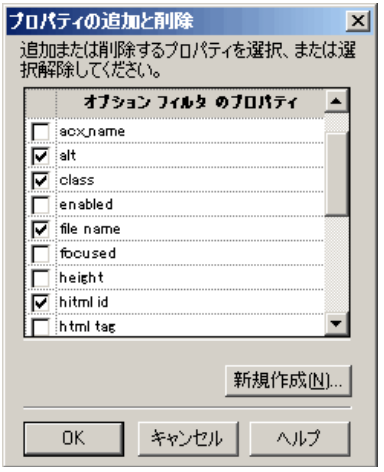
［**新規作成**］をクリックして，表示されるダイアログ・ボックスに有効なプロパティ名を指定することにより，新規プロパティを指定できます。

ヒント：attribute/ ＜プロパティ名＞の形式を使用して，Web オブジェクトに使用可能なプロパティの集合にプロパティ名を追加することもできます。これを行うためには，［**新規作成**］をクリックします。［新規プロパティ］ダイアログ・ボックスが開きます。有効なプロパティを，attribute/ ＜プロパティ名＞の形式で入力して，［**OK**］をクリックします。新規プロパティが［**基本フィルタのプロパティ**］リストに追加されます。たとえば，MyColor というプロパティを追加するには，attribute/MyColor と入力します。

---

7 ［**OK**］をクリックし，［プロパティの追加と削除］ダイアログ・ボックスを閉じます。更新された基本フィルタ・プロパティの集合が［**基本フィルタのプロパティ**］リストに表示されます。

8 ［**オプション フィルタのプロパティ**］リストで，［**追加と削除**］をクリックします。オプション・フィルタ・プロパティのための［プロパティの追加と削除］ダイアログ・ボックスが表示されます。



9 ［**オプション フィルタのプロパティ**］リストに含めるプロパティを選択します。また，リストから削除するプロパティを消去します。

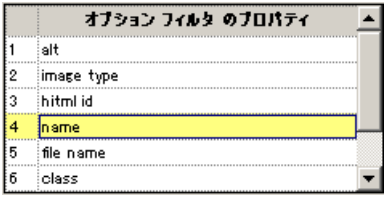
注：同じプロパティを基本とオプションの両方のプロパティ・リストに含めることはできません。

［**新規作成**］をクリックして，表示されるダイアログ・ボックスに有効なプロパティ名を指定することにより，新規プロパティを指定できます。

ヒント：attribute/ **＜プロパティ名＞**の形式を使用して，Web オブジェクトに使用可能なプロパティの集合にプロパティ名を追加することもできます。これを行うためには，［**新規作成**］をクリックします。［新規プロパティ］ダイアログ・ボックスが開きます。有効なプロパティを，attribute/ **＜プロパティ名＞**の形式で入力して，［**OK**］をクリックします。新規プロパティが［**オプション フィルタのプロパティ**］リストに追加されます。たとえば，MyColor というプロパティを追加するには，attribute/MyColor と入力します。

---

10 ［**OK**］をクリックし，［プロパティの追加と削除］ダイアログ・ボックスを閉じます。プロパティは［**オプション フィルタのプロパティ**］リストに表示されます。



11 上向き矢印キーと下向き矢印キーを使って，オプション・フィルタ・プロパティの順序を指定します。QuickTest はスマート認識メカニズムを使うとき，オプション・プロパティに対する残りのオブジェクト候補を［**オプション フィルタのプロパティ**］で設定した順序に従って，オブジェクト候補が 1 つになるまで 1 つずつチェックします。

ユーザ定義のテスト・オブジェクト・クラスの割り当て

オブジェクトの割り当て］ダイアログ・ボックスを使って未定義クラスまたはユーザ定義クラスを Standard Windows クラスに割り当てることができます。たとえば，アプリケーションに識別できないボタンがある場合，そのボタンは汎用の WinObject として学習されます。QuickTest に対して，そのオブジェクトが標準の Windows **button** クラスに属しているものとして識別するように指示できます。そのようにしておくと，記録中にそのボタンをクリックすると，QuickTest はその操作を標準の Windows ボタンをクリックしたのと同じように記録します。未定義オブジェクトまたはユーザ定義オブジェクトを標準オブジェクトに割り当てると，そのオブジェクトは Standard Windows テスト・オブジェクト・クラスのリストに，ユーザ定義のテスト・オブジェクト・クラスとして追加されます。オブジェクトの認識設定は，ほかのテスト・オブジェクト・クラスと同様に，ユーザ定義のテスト・オブジェクト・クラスに設定できます。

認識されないオブジェクトは，同等の動作をする Standard Windows クラスにのみ割り当てる必要があります。たとえば，ボタンと同等の動作をするオブジェクトを edit クラスに割り当ててはなりません。

注：

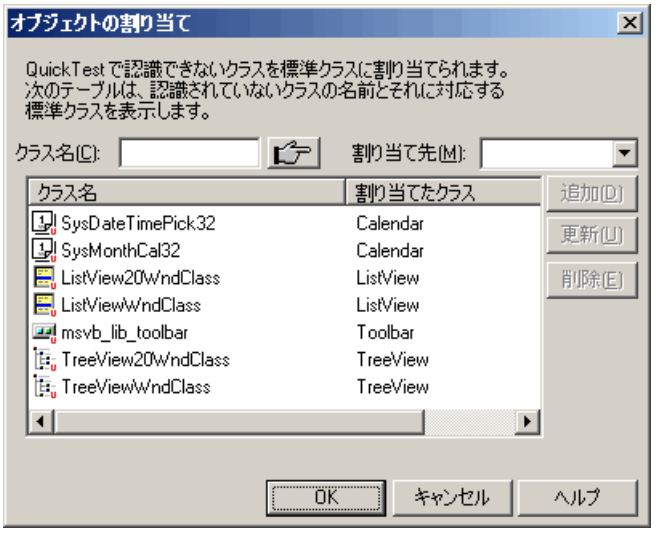
- ユーザ定義クラスを定義できるのは，［**環境**］ボックスで［**Standard Windows**］が選択されている場合だけです。
- ［**テスト オブジェクトを元に戻す**］ボタンの下向き矢印をクリックして［**環境を元に戻す**］を選択すると，［**Standard Windows**］が［**環境**］ボックスで選択されていれば，ユーザ定義のテスト・オブジェクト・クラスがすべて削除されます。

未定義クラスまたはユーザ定義クラスを Standard Windows クラスに割り当てるには，次の手順を実行します。

1 ［**ツール**］＞［**オブジェクトの認識**］を選択します。［オブジェクトの認識］ダイアログ・ボックスが表示されます。

2 ［**環境**］ボックスで［**Standard Windows**］を選択します。［**ユーザ定義**］ボタンが有効になります。

3［**ユーザ定義**］ボタンをクリックします。［オブジェクトの割り当て］ダイアログ・ボックスが表示されます。



4 指差しボタンをクリックしてから，ユーザ定義クラスに追加するクラスのオブジェクトをクリックします。ユーザ定義オブジェクトの名前が［**クラス名**］ボックスに表示されます。

指差しマーク機能の使用方法の詳細については，221 ページ「指差しマークの使用法に関するヒント」を参照してください。

5［**割り当て先**］ボックスで，ユーザ定義オブジェクト・クラスを割り当てる対象となる標準オブジェクト・クラスを選択して［**追加**］をクリックします。クラス名と割り当てがオブジェクト割り当てリストに追加されます。

6 標準クラスにさらにオブジェクトを追加する場合は，オブジェクトごとに手順 4 ～ 5 を繰り返します。

7 [**OK**] をクリックします。[オブジェクトの割り当て] ダイアログ・ボックスが閉じ，オブジェクトが Standard Window テスト・オブジェクト・クラスのリストにユーザ定義テスト・オブジェクトとして追加されます。追加したオブジェクトのアイコンの右下角には，ユーザ定義クラスであることを表す赤い U の文字が入ります。

8 ユーザ定義オブジェクト・クラスのオブジェクトの認識設定は，ほかのあらゆるオブジェクト・クラスと同様に設定できます。詳細については，195 ページ「必須および補足プロパティの設定」および 209 ページ「スマート認識の設定」を参照してください。

既存の割り当てを変更するには，次の手順を実行します。

1 [オブジェクトの割り当て] ダイアログ・ボックスのオブジェクト割り当てリストで，変更するクラスを選択します。そのクラス名と現在の割り当てが，[クラス名] および [割り当て先] ボックスに表示されます。

2 選択したユーザ定義オブジェクト・クラスを割り当てる対象となる標準オブジェクト・クラスを選択して [**更新**] をクリックします。オブジェクト割り当てリストのクラス名と割り当てが更新されます。

3 [**OK**] をクリックし，[オブジェクトの割り当て] ダイアログ・ボックスを閉じます。

既存の割り当てを削除するには，次の手順を実行します。

1 [オブジェクトの割り当て] ダイアログ・ボックスのオブジェクト割り当てリストで，削除するクラスを選択します。

2 [**削除**] をクリックします。そのクラス名と割り当てが，[オブジェクトの割り当て] ダイアログ・ボックスのオブジェクト割り当てリストから削除されます。

3 [**OK**] をクリックします。[オブジェクトの割り当て] ダイアログ・ボックスが閉じ，[オブジェクトの認識] ダイアログ・ボックスの Standard Windows テスト・オブジェクト・クラスのリストからクラス名が削除されます。

指差しマークの使用法に関するヒント

➤ 左の CTRL キーを押したままにすると，指差しマークが標準ポインタに変わります。そのときに，QuickTest またはアプリケーションでウィンドウのフォーカスを変更したり，操作（右クリックしたり，スクロール・バーを使ったり，ポインタをオブジェクト上に移動してショートカット・メニューを表示するなど）を実行できます。

- 選択するオブジェクトがあるウィンドウが別のウィンドウで部分的に隠れている場合は，部分的に隠れているウィンドウが前面に表示されるまで，その上に指差しマークを数秒間置いたままにします。次に，必要なオブジェクトをポイントしてクリックします。ウィンドウを前面に出すのに必要な時間の設定は，［オプション］ダイアログ・ボックスの［一般］表示枠で行えます。
- 選択するオブジェクトがあるウィンドウが別のウィンドウで完全に隠れている場合，またはダイアログ・ボックスがウィンドウの背後に隠れている場合は，必要に応じて左の CTRL キーを押しながらウィンドウを配置します。
- 選択するオブジェクトがあるウィンドウが最小化されている場合は，左の CTRL キーを押したまま，Windows タスク・バーでアプリケーションを右クリックして，ショートカット・メニューから［**元のサイズに戻す**］を選択すると，そのウィンドウを表示できます。
- 特定のイベント（ショートカット・メニューを表示するために右クリックしたり，オブジェクト上にポインタを移動するなど）を実行するだけで，選択するオブジェクトを表示できる場合は，左の CTRL キーを押したままにします。指差しマークが一時的に標準ポインタに変わり，イベントを実行できます。選択するオブジェクトが表示されたら，左の CTRL キーを放します。ポインタが再び指差しマークに変わります。

第 7 章

オブジェクト・リポジトリの管理

オブジェクト・リポジトリ・マネージャでは，オブジェクトの追加および定義，オブジェクトおよびその記述の変更，リポジトリの汎用性を高めるたのパラメータ化，リポジトリの保守および組織化，リポジトリの結合，XML 形式でのリポジトリのインポートおよびエクスポートなど，組織で使用されているすべての共有オブジェクト・リポジトリを一元管理することができます。

本章の内容

- オブジェクト・リポジトリの管理について（224 ページ）
- オブジェクト・リポジトリ・マネージャ（226 ページ）
- オブジェクト・リポジトリを使った作業（233 ページ）
- 共有オブジェクト・リポジトリでのオブジェクトの管理（239 ページ）
- リポジトリ・パラメータを使用した作業（245 ページ）
- オブジェクトの詳細の変更（250 ページ）
- テスト・オブジェクトの検索（254 ページ）
- 結合操作の実行（255 ページ）
- インポート操作およびエクスポート操作の実行（256 ページ）
- オートメーションの使用によるオブジェクト・リポジトリの管理（259 ページ）

オブジェクト・リポジトリの管理について

オブジェクト・リポジトリ・マネージャでは，共有オブジェクト・リポジトリの作成および保守ができます。ファイル・システムおよび Quality Center プロジェクトに保存されているオブジェクト・リポジトリのどちらも使用できます。

各オブジェクト・リポジトリには，QuickTest によるアプリケーション内のオブジェクトの識別を可能にする情報が含まれています。QuickTest により，テスト・オブジェクトに関するすべての情報を共有オブジェクト・リポジトリに格納することで，コンポーネントの再利用性を維持できますオブジェクト・リポジトリ・マネージャは，アプリケーションのオブジェクトに変更があった場合に，複数のコンポーネントのテスト・オブジェクト情報を 1 か所で集中して更新できる場所となります。

注：共有オブジェクト・リポジトリの代わりに，または共有オブジェクト・リポジトリに加えて，一部または全部のオブジェクトをコンポーネントごとにローカル・オブジェクト・リポジトリに保存するという選択肢もあります。ローカル・オブジェクト・リポジトリの詳細については，第 5 章「オブジェクト・リポジトリ内のテスト・オブジェクトの管理」を参照してください。

アプリケーション内のオブジェクトのプロパティ値の中に，QuickTest によって，オブジェクトの識別に使用されるプロパティ値と異なるものがあると，コンポーネントは失敗することがあります。そのため，アプリケーション内にあるオブジェクトのプロパティ値に変更があった場合，既存のコンポーネントを継続して使用するには，対応するオブジェクト・リポジトリ内にある対応する認識プロパティ値を修正する必要があります。

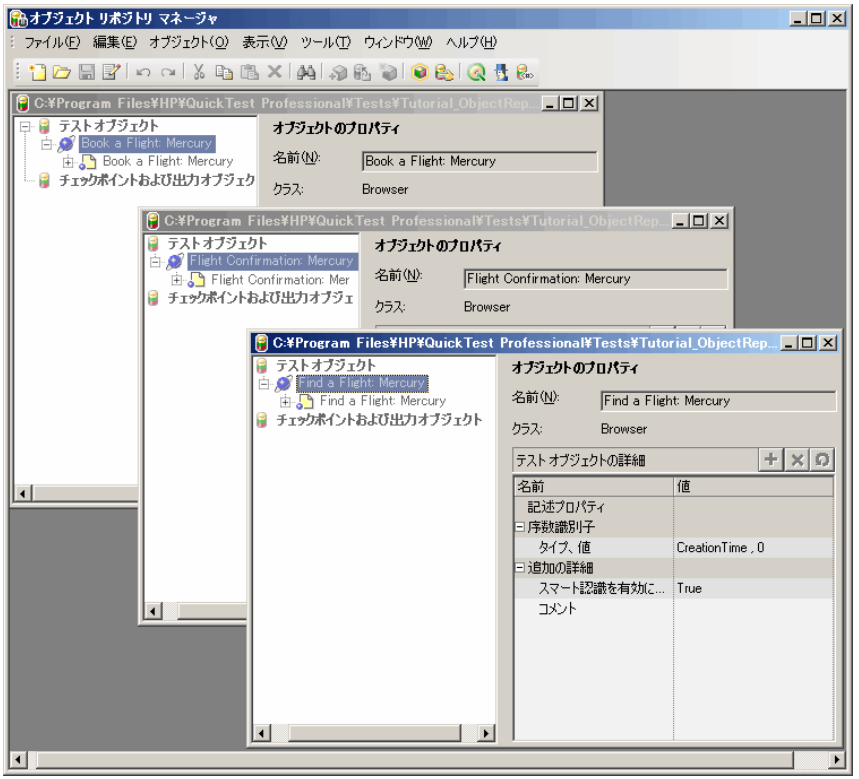
同じコンポーネントに関連付けられているローカル・オブジェクト・リポジトリと共有オブジェクト・リポジトリの両方に，同じ名前と記述を持つオブジェクトが存在する場合は，そのコンポーネントに対するローカルのオブジェクト定義が使用されます。同じコンポーネントに関連付けられている複数の共有オブジェクト・リポジトリ内に同じ名前と記述を持つオブジェクトがある場合，QuickTest では共有オブジェクト・リポジトリがコンポーネントに関連付けられている順序に従って，最初に出現したオブジェクトのオブジェクト定義が使用されます。共有オブジェクト・リポジトリの関連付けの詳細については，465 ページ「共有オブジェクト・ライブラリの関連付け」を参照してください。

複数のコンポーネントで同じ共有オブジェクト・リポジトリを使用できます。各コンポーネントで複数のオブジェクト・リポジトリを使用することもできます。また，コンポーネントと一緒に，オブジェクトをローカル・オブジェクト・リポジトリに直接保存することもできます。これにより，オブジェクトはそのコンポーネントからのみアクセスできるようになります。共有オブジェクト・リポジトリが Quality Center に保管されている場合は，それらにバージョン・コントロールを適用できます。詳細については，967 ページ「バージョン・コントロールを使ったアセット管理」を参照してください。

共有オブジェクト・リポジトリ内のオブジェクトは，本章で説明するように，オブジェクト・リポジトリ・マネージャを使用して変更できます。ローカルのオブジェクト・リポジトリに格納されているオブジェクトは，［オブジェクト リポジトリ］ウィンドウを使用して変更できます。［オブジェクト リポジトリ］ウィンドウの詳細については，第 5 章「オブジェクト・リポジトリ内のテスト・オブジェクトの管理」を参照してください。

オブジェクト・リポジトリ・マネージャ

オブジェクト・リポジトリ・マネージャを開くには，［**リソース**］＞［**オブジェクト リポジトリ マネージャ**］を選択します。オブジェクト・リポジトリ・マネージャでは，複数の共有オブジェクト・リポジトリを開き，必要に応じてそれらを変更できます。共有オブジェクト・リポジトリは，ファイル・システムと Quality Center プロジェクトのどちらからも開けます。



ヒント：オブジェクト・リポジトリ・マネージャを開いている間も，引き続き QuickTest のほかのウィンドウで作業が行えます。

共有オブジェクト・リポジトリは，必要な数だけ開けます。各共有オブジェクト・リポジトリは，それぞれ別のドキュメント・ウィンドウに開きます。開いたウィンドウは，必要に応じてサイズ変更，最大化，最小化して配置することで，共有オブジェクト・リポジトリ間でオブジェクトをコピーしたり，ドラッグして移動したりできるほか，オブジェクト・リポジトリを個別に操作できます。共有オブジェクト・リポジトリ・ウィンドウに表示される詳細については，231 ページ「[共有オブジェクト・リポジトリ］ウィンドウについて」を参照してください。

共有オブジェクト・リポジトリは，［共有オブジェクト リポジトリを開く］ダイアログ・ボックスから開きます。このダイアログ・ボックスの，［**読み取り専用モードで開く**］チェック・ボックスは標準設定で選択されています。このチェック・ボックスをクリアすると，共有オブジェクト・リポジトリは編集可能なモードで開きます。それ以外の場合，共有オブジェクト・リポジトリは読み取り専用モードで開きます。変更するには［**編集を有効化**］ボタンをクリックする必要があります。詳細については，240 ページ「オブジェクト・リポジトリの編集」を参照してください。

オブジェクト・リポジトリ・マネージャの中でメニュー項目を選択するか，ツールバー・ボタンをクリックすると，ウィンドウが現在アクティブな（フォーカスがある）共有オブジェクト・リポジトリを対象に選択した操作が実行されます。ウィンドウのタイトル・バーに，共有オブジェクト・リポジトリの名前およびファイル・パスが表示されます。オブジェクト・リポジトリ・マネージャのツールバー・ボタンの詳細については，228 ページ「オブジェクト・リポジトリ・マネージャ・ツールバーの使用について」を参照してください。

QuickTest が バージョン・コントロールの有効な Quality Center プロジェクトに接続されている場合は，共有オブジェクト・リポジトリのバージョンの確認と管理，2 つの共有オブジェクト・リポジトリ・バージョンの比較，ベースライン履歴の確認ができます。詳細については，967 ページ「バージョン・コントロールを使ったアセット管理」および 949 ページ「QuickTest アセットのバージョンの表示と比較」を参照してください。

オブジェクト・リポジトリ・マネージャの中で実行できる共有オブジェクト・リポジトリ操作の多くは，（［オブジェクト リポジトリ］ウィンドウを使用して）ローカルのオブジェクト・リポジトリに格納されているオブジェクトに変更を加える方法と同じような方法で行います。したがって，手順の多くは実際には第 5 章「オブジェクト・リポジトリ内のテスト・オブジェクトの管理」で説明しています。手順のほとんどは，オブジェクト・リポジトリ・マネージャと［オブジェクト リポジトリ］ウィンドウで同じですが，ウィンドウとオプションに若干の相違がある場合があります。

オブジェクト・リポジトリ・マネージャ・ツールバーの使用について

よく行う操作は，オブジェクト・リポジトリ・マネージャ・ツールバーから利用できますオブジェクト・リポジトリ・マネージャ・ツールバーには，次のボタンがあります。

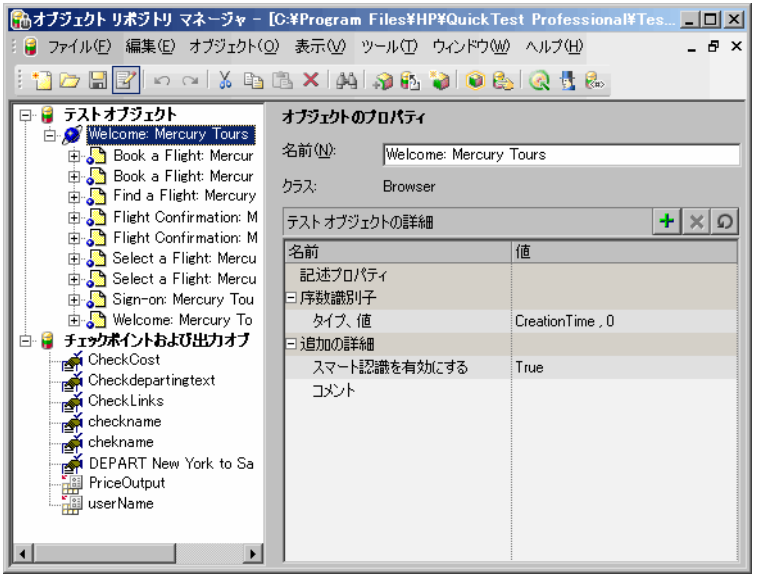
ボタン	詳細
	新規の共有オブジェクト・リポジトリを作成します。詳細については，233 ページ「新しいオブジェクト・リポジトリの作成」を参照してください。
	ファイル・システムまたは Quality Center から共有オブジェクト・リポジトリを開きます。詳細については，234 ページ「オブジェクト・リポジトリを開く」を参照してください。
	アクティブな共有オブジェクト・リポジトリをファイル・システムまたは Quality Center に保存します。詳細については，235 ページ「オブジェクト・リポジトリの保存」を参照してください。
	共有オブジェクト・リポジトリを編集可能にすることで，アクティブな共有オブジェクト・リポジトリを編集します。詳細については，240 ページ「オブジェクト・リポジトリの編集」を参照してください。
	アクティブな共有オブジェクト・リポジトリで行った前の操作を元に戻します。ローカル・オブジェクト・リポジトリでの場合と同じように行います。詳細については，164 ページ「オブジェクト・リポジトリ内のオブジェクトのコピー，貼り付け，および移動」を参照してください。
	アクティブな共有オブジェクト・リポジトリで前に元に戻した操作を再度実行します。ローカル・オブジェクト・リポジトリでの場合と同じように行います。詳細については，164 ページ「オブジェクト・リポジトリ内のオブジェクトのコピー，貼り付け，および移動」を参照してください。
	アクティブな共有オブジェクト・リポジトリで選択されている項目またはオブジェクトを切り取ります。ローカル・オブジェクト・リポジトリでの場合と同じように行います。詳細については，164 ページ「オブジェクト・リポジトリ内のオブジェクトのコピー，貼り付け，および移動」を参照してください。

ボタン	詳細
	アクティブな共有オブジェクト・リポジトリで選択されている項目またはオブジェクトをクリップボードにコピーします。ローカル・オブジェクト・リポジトリでの場合と同じように行います。詳細については，164 ページ「オブジェクト・リポジトリ内のオブジェクトのコピー，貼り付け，および移動」を参照してください。
	クリップボードのデータをアクティブな共有オブジェクト・リポジトリに貼り付けます。ローカル・オブジェクト・リポジトリでの場合と同じように行います。詳細については，164 ページ「オブジェクト・リポジトリ内のオブジェクトのコピー，貼り付け，および移動」を参照してください。
	アクティブな共有オブジェクト・リポジトリで選択されている項目またはオブジェクトを削除します。ローカル・オブジェクト・リポジトリでの場合と同じように行います。詳細については，167 ページ「オブジェクト・リポジトリからのオブジェクトの削除」を参照してください。
	アクティブな共有オブジェクト・リポジトリ内のオブジェクト，プロパティ，またはプロパティ値を検索します。指定したプロパティ値を検索して置換することもできます。ローカル・オブジェクト・リポジトリでの場合と同じように行います。詳細については，168 ページ「オブジェクト・リポジトリ内のオブジェクトの検索」を参照してください。
	アクティブな共有オブジェクト・リポジトリにオブジェクトを追加します。ローカル・オブジェクト・リポジトリでの場合と同じように行います。詳細については，152 ページ「オブジェクト・リポジトリへのテスト・オブジェクトの追加」を参照してください。
	アプリケーション内にあるオブジェクトの実際のプロパティに従って，アクティブな共有オブジェクト・リポジトリ内の認識プロパティを更新します。ローカル・オブジェクト・リポジトリでの場合と同じように行います。詳細については，178 ページ「アプリケーション内のオブジェクトから認識プロパティを更新」を参照してください。
	アプリケーションに存在しないテスト・オブジェクトを定義して，アクティブな共有オブジェクト・リポジトリに追加します。ローカル・オブジェクト・リポジトリでの場合と同じように行います。詳細については，161 ページ「新規テスト・オブジェクトの定義」を参照してください。

ボタン	詳細
	アクティブな共有オブジェクト・リポジトリ内でオブジェクトを選択すると，アプリケーション内でそのオブジェクトが強調表示されます。ローカル・オブジェクト・リポジトリでの場合と同じように行います。詳細については，171 ページ「アプリケーション内のオブジェクトの強調表示」を参照してください。
	アプリケーション内でオブジェクトを選択すると，アクティブな共有オブジェクト・リポジトリ内でそのオブジェクトが強調表示されます。ローカル・オブジェクト・リポジトリでの場合と同じように行います。詳細については，172 ページ「オブジェクト・リポジトリ内のテスト・オブジェクトの検索」を参照してください。
	Quality Center に接続し，Quality Center プロジェクトに格納されているオブジェクト・リポジトリ・ファイルを使って作業を行います。Quality Center には，Quality Center のメイン・ウィンドウまたは［オブジェクト リポジトリ マネージャ］から接続します。詳細については，46 ページ「Quality Center プロジェクトへの接続」を参照してください。
	［オブジェクト スパイ］ダイアログ・ボックスを開き，開いているアプリケーションでオブジェクトのネイティブなプロパティと操作に加えて，テスト・オブジェクト階層，認識プロパティ，および QuickTest でそのオブジェクトを示すのに使用するテスト・オブジェクト操作を表示できます。詳細については，119 ページ「オブジェクト・スパイを使用したオブジェクト・プロパティと操作の表示」を参照してください。
	アクティブな共有オブジェクト・リポジトリ内のリポジトリ・パラメータを追加，編集，および削除します。詳細については，246 ページ「リポジトリ・パラメータの管理」を参照してください。

[共有オブジェクト・リポジトリ] ウィンドウについて

オブジェクト・リポジトリ・マネージャで開いた各共有オブジェクト・リポジトリは，個別のドキュメント・ウィンドウに表示されます。各共有オブジェクト・リポジトリのウィンドウには，オブジェクト・リポジトリ内のすべてのオブジェクトのツリーが，選択したオブジェクトのオブジェクト情報とともに表示されます。



ツリーで選択したオブジェクトごとに，選択したオブジェクトに関する情報が [オブジェクト リポジトリ] ウィンドウに表示されます。共有オブジェクト・リポジトリ内の任意のオブジェクトのオブジェクト記述の表示，オブジェクトやそのプロパティの変更，共有オブジェクト・リポジトリへのオブジェクトの追加が可能です。

注：

- ➤ オブジェクト・リポジトリ・マネージャでは，チェックポイントまたは出力値のオブジェクトを共有オブジェクト・リポジトリに追加することはできません。
- ➤ QuickTest にインストールされていない環境のテスト・オブジェクトは，テスト・オブジェクト・ツリーでは疑問符アイコン付きで表示されます。

詳細については，239 ページ「共有オブジェクト・リポジトリでのオブジェクトの管理」および 250 ページ「オブジェクトの詳細の変更」を参照してください。

各オブジェクト・リポジトリ・ウィンドウには，次の情報が表示されます。

情報	詳細
[オブジェクト リポジトリ ツリー]	［オブジェクト リポジトリ］ウィンドウの左側にあります。共有オブジェクト・リポジトリのすべてのオブジェクトが含まれています。
[名前]	選択したオブジェクトに対して QuickTest によって割り当てられている名前を示します。オブジェクトの名前は変更できます。詳細については，182 ページ「テスト・オブジェクトの名前の変更」を参照してください。
[クラス]	選択したオブジェクトのクラスを示します。
[オブジェクトの詳細]	［オブジェクト リポジトリ］ウィンドウの右下にあります。実行セッション中にテスト・オブジェクトの識別に使用されるプロパティおよびプロパティ値，またはチェックポイントや出力オブジェクトのプロパティを表示します。詳細については，250 ページ「オブジェクトの詳細の変更」を参照してください。

注：オブジェクトが含まれるステップを，コンポーネントから削除しても，オブジェクトはオブジェクト・リポジトリから削除されません。共有オブジェクト・リポジトリのオブジェクトを削除するには，オブジェクト・リポジトリ・マネージャを使用して，ローカル・オブジェクト・リポジトリからオブジェクトを削除する場合とほぼ同じ方法で削除します。詳細については，167 ページ「オブジェクト・リポジトリからのオブジェクトの削除」を参照してください。

オブジェクト・リポジトリを使った作業

オブジェクト・リポジトリ・マネージャを使用して，オブジェクト・リポジトリを新規作成し，既存のオブジェクト・リポジトリを開いて変更し，終了時にはリポジトリを保存して閉じることができます。

新しいオブジェクト・リポジトリの作成

本項で説明する機能は，［**オブジェクト リポジトリ マネージャ**］ウィンドウで作業している場合にのみ利用できます。

新しいオブジェクト・リポジトリを作成してオブジェクトを追加し，保存することができます。その後，QuickTest で，1 つ以上のコンポーネントをオブジェクト・リポジトリに関連付けることができます。共有オブジェクト・リポジトリの関連付けの詳細については，465 ページ「共有オブジェクト・ライブラリの関連付け」を参照してください。

新しいオブジェクト・リポジトリを作成するには，次の手順を実行します。

［オブジェクト リポジトリ マネージャ］で，［**ファイル**］＞［**新規作成**］を選択するか，［**新規作成**］ボタンをクリックします。新しいオブジェクト・リポジトリが開きます。これで，オブジェクト・リポジトリへのオブジェクトの追加や，オブジェクト・リポジトリの変更および保存ができます。詳細については，239 ページ「共有オブジェクト・リポジトリでのオブジェクトの管理」および 235 ページ「オブジェクト・リポジトリの保存」を参照してください。

オブジェクト・リポジトリを開く

本項で説明する機能は，［**オブジェクト リポジトリ マネージャ**］ウィンドウで作業している場合にのみ利用できます。

既存のオブジェクト・リポジトリを開き，それらの表示または変更ができます。オブジェクト・リポジトリは，ファイル・システムまたは Quality Center プロジェクトから開くことができます。

QuickTest または［オブジェクト リポジトリ マネージャ］から Quality Center プロジェクトに接続するには，［**ファイル**］＞［**Quality Center への接続**］を選択するか，［**Quality Center への接続**］ボタンをクリックします。Quality Center への接続の詳細については，46 ページ「Quality Center プロジェクトへの接続」を参照してください。

QuickTest の以前のバージョンをお使いのユーザの方へ：

バージョン 9.0 以前の QuickTest を使用して作成され，ファイル・システムに保管されたオブジェクト・リポジトリを開く際，そのリポジトリを編集可能にすると，そのリポジトリは QuickTest によって最新の形式に変換されます。

オブジェクト・リポジトリにアドインからのテスト・オブジェクトが含まれている場合，オブジェクト・リポジトリを現在の形式に変換するためには関連するアドインがインストールされている必要があります。インストールされていない場合，読み取り専用形式でのみ開けます。

オブジェクト・リポジトリを変換したくない場合は，読み取り専用形式で表示します。ファイルを変換して保存すると，そのファイルは以前のバージョンの QuickTest では使用できません。

オブジェクト・リポジトリを開くには，次の手順を実行します。

1 ［オブジェクト リポジトリ マネージャ］で，［**ファイル**］＞［**開く**］を選択するか，［**開く**］ボタンをクリックします。［共有オブジェクト リポジトリを開く］ダイアログ・ボックスが開きます。

2 サイドバーで，オブジェクト・リポジトリ・ファイルの場所を選択します (File System や Quality Center Test Resources など)。開きたいオブジェクト・リポジトリ・ファイルを参照して選択し，［**開く**］をクリックします。オブジェクト・リポジトリが開きます。

標準設定では，オブジェクト・リポジトリは読み取り専用モードで開きます。編集可能な形式で開くには，［共有オブジェクト リポジトリを開く］ダイアログ・ボックスで［**読み取り専用モードで開く**］チェック・ボックスをクリアします。240 ページ「オブジェクト・リポジトリの編集」で説明している方法で，オブジェクト・リポジトリを編集することもできます。

オブジェクト・リポジトリが編集可能になっている場合は，オブジェクトの追加，オブジェクト・リポジトリの変更，および保存ができます。詳細については，239 ページ「共有オブジェクト・リポジトリでのオブジェクトの管理」および 235 ページ「オブジェクト・リポジトリの保存」を参照してください。

ヒント：オブジェクト・リポジトリは［**ファイル**］メニューの［**最近使用したファイル**］のリストからも開けます。

オブジェクト・リポジトリの保存

本項で説明する機能は，［**オブジェクト リポジトリ マネージャ**］ウィンドウで作業している場合にのみ利用できます。

オブジェクト・リポジトリを作成または変更を終了したら，保存する必要があります。オブジェクト・リポジトリに変更を加えると，オブジェクト・リポジトリを保存するまで，タイトル・バーにアスタリスク（*****）が表示されます。

オブジェクト・リポジトリは，ファイル・システムまたは Quality Center プロジェクト（Quality Center プロジェクトに接続している場合）に保存できます。共有オブジェクト・リポジトリをアプリケーション領域に関連付けてコンポーネントからアクセスできるようにするには，Quality Center プロジェクトに保存する必要があります。QuickTest または［オブジェクト リポジトリ マネージャ］から Quality Center プロジェクトに接続するには，［**ファイル**］＞［**Quality Center への接続**］を選択するか，［**Quality Center への接続**］ボタンをクリックします。Quality Center への接続の詳細については，46 ページ「Quality Center プロジェクトへの接続」を参照してください。

オブジェクト・リポジトリに加えた変更はすべて，まだ変更内容を保存していなくても，変更した直後に，オブジェクト・リポジトリを使用している同じコンピュータ上で開いているすべてのコンポーネントにおいて自動的に更新されます。変更を保存せずにオブジェクト・リポジトリを閉じると，変更時に開いていたすべてのコンポーネントにおいて，変更がロール・バックされます。

オブジェクト・リポジトリを変更した同じコンピュータ上でコンポーネントを開くと，コンポーネントは，関連付けられているオブジェクト・リポジトリに保存したすべての変更内容で自動的に更新されます。保存した変更を，別のコンピュータ上で開いているコンポーネントまたはリポジトリで表示するには，そのコンポーネントまたはオブジェクト・リポジトリ・ファイルを開くかコンピュータ上で編集用にロックすることによって，変更を読み込む必要があります。

オブジェクト・リポジトリを保存するには，次の手順を実行します。

1 保存するオブジェクト・リポジトリがアクティブ・ウィンドウであることを確認します。

2［**ファイル**］＞［**保存**］を選択するか，［**保存**］ボタンをクリックします。ファイルをすでに保存したことがあれば，加えた変更が保存されます。ファイルを保存したことがなければ，［共有オブジェクト リポジトリの保存］ダイアログ・ボックスが開きます。

3 サイドバーで，ファイルを保存する場所を選択します（File System や Quality Center Test Resources など）。

4 ファイルを保存するフォルダを参照し，そのフォルダを選択します。

5［**ファイル 名**］ボックスに，ファイルの名前を入力します。ファイルを識別しやすいように，わかりやすい名前を使用します。オブジェクト・リポジトリの名前には，次の文字は使用できません。
¥ / : * " ? < > | ’

オブジェクト・リポジトリを Quality Center に保存する場合は，ファイル・パスに連続した 2 つのセミコロン（;;）を含める必要があります。

6［**保存**］をクリックします。

注：ファイル・システムまたは Quality Center 9.x にあるリソースへのパスを指定すると，QuickTest によって，［オプション］ダイアログ・ボックスの［フォルダ］表示枠（［**ツール**］＞［**オプション**］＞［**フォルダ**］ノード）にパスまたはその一部があるかチャックされます。パスが存在する場合は，入力したパスの相対的な部分だけを使用してパスを定義するように求められます。パスが存在しない場合は，リソースの場所のパスを［フォルダ］表示枠に追加して，相対的なパスを定義するように要求されます。詳細については，629 ページ「グローバル・テスト・オプションの設定」を参照してください。

Quality Center 10.00 を使ってリソースおよび依存関係モデルを使用する場合は，Quality Center の絶対パスを指定してください。詳細については，940 ページ「Quality Center で相対パスを使用する場合の注意事項」を参照してください。

QuickTest はオブジェクト・リポジトリを指定された場所に拡張子 **.tsr** を付けて保存し，オブジェクト・リポジトリの名前とパスをリポジトリ・ウィンドウのタイトル・バーに表示します。

オブジェクト・リポジトリを閉じる

本項で説明する機能は，［**オブジェクト リポジトリ マネージャ**］ウィンドウで作業している場合にのみ利用できます。

オブジェクト・リポジトリの変更または使用が終了したら，そのリポジトリを閉じる必要があります。オブジェクト・リポジトリは，編集の間ほかのユーザによって変更されないようにロックされます。オブジェクト・リポジトリを閉じると，自動的にロックが解除されます。また，開いているすべてのオブジェクト・リポジトリを閉じることもできます。

注：QuickTest を終了すると，オブジェクト・リポジトリ・マネージャも終了します。まだ保存していない変更がある場合は，オブジェクト・リポジトリ・マネージャを終了する前に保存するように求められます。

オブジェクト・リポジトリを閉じるには，次の手順を実行します。

1 閉じる対象となるオブジェクト・リポジトリがアクティブ・ウィンドウであることを確認します。

2 ［**ファイル**］＞［**閉じる**］を選択するか，オブジェクト・リポジトリ・ウィンドウのタイトル・バーで［**閉じる**］ボタンをクリックします。オブジェクト・リポジトリが閉じ，ロックが自動的に解除されます。まだ保存していない変更がある場合は，ファイルを閉じる前に保存するよう求められます。

開いているすべてのオブジェクト・リポジトリを閉じるには，次の手順を実行します。

［**ファイル**］＞［**すべてのウィンドウを閉じる**］，または［**ウィンドウ**］＞［**すべてのウィンドウを閉じる**］を選択します。開いているすべてのオブジェクト・リポジトリが閉じ，ロックが自動的に解除されます。まだ保存していない変更がある場合は，ファイルを閉じる前に保存するよう求められます。

共有オブジェクト・リポジトリでのオブジェクトの管理

共有オブジェクト・リポジトリは，初めて使用するための準備を整えるために，または，テスト・プロセス全体を通じて共有オブジェクト・リポジトリを更新するために，さまざまな方法で変更することができます。共有オブジェクト・リポジトリのオブジェクトやオブジェクト・プロパティの追加および変更，オブジェクト・リポジトリ間でのオブジェクトのコピーまたは移動，階層内の別の場所へのオブジェクトのドラッグ，オブジェクトの削除，オブジェクト名の変更ができます。また，テスト・オブジェクトをオブジェクト・リポジトリ・マネージャからコンポーネントにドラッグ・アンド・ドロップすることもできます。共有オブジェクト・リポジトリを変更を加えると，オブジェクト・リポジトリを保存するまで，タイトル・バーにアスタリスク（*）が表示されます。

次に，［オブジェクト リポジトリ マネージャ］の使用に関するヒントとガイドラインを示します。

- 必要に応じて，［**編集**］＞［**元に戻す**］や［**編集**］＞［**やり直し**］オプションまたは［**元に戻す**］ボタンや［**やり直し**］ボタンを使用して，変更の取り消しや繰り返しを行うことができます。［**元に戻す**］および［**やり直し**］オプションは，アクティブなドキュメントを対象とします。オブジェクト・リポジトリを保存すると，保存操作の前にそのファイルに対して行った操作の取り消しや，やり直しはできません。
- オブジェクト・リポジトリを読み取り専用モードで開いた場合，変更を加えるにはオブジェクト・リポジトリの編集を有効にする必要があります。これによってオブジェクト・リポジトリがロックされ，複数のユーザが同時に変更できないようになります。
- オブジェクト・リポジトリに加えた変更はすべて，まだ変更内容を保存していなくても，変更した直後に，オブジェクト・リポジトリを使用している同じコンピュータ上で開いているすべてのコンポーネントにおいて自動的に更新されます。

  変更を保存せずにオブジェクト・リポジトリを閉じると，変更時に開いていたすべてのコンポーネントにおいて，変更がロール・バックされます。
- オブジェクト・リポジトリを変更した同じコンピュータ上でコンポーネントを開くと，コンポーネントは，関連付けられているオブジェクト・リポジトリに保存したすべての変更内容で自動的に更新されます。保存した変更を，別のコンピュータ上で開いているコンポーネントまたはリポジトリで表示するには，そのコンポーネントまたはオブジェクト・リポジトリ・ファイルを開くかコン

ピュータ上で編集用にロックすることによって，変更を読み込む必要があります。

- 共有オブジェクト・リポジトリは，別の共有オブジェクト・リポジトリと結合することでも変更できます。2 つの共有オブジェクト・リポジトリを結合すると，両方のオブジェクト・リポジトリの内容を含む，新しい共有オブジェクト・リポジトリが作成されます。ローカル・オブジェクト・リポジトリを共有オブジェクト・リポジトリに結合すると，共有オブジェクト・リポジトリが，ローカル・オブジェクト・リポジトリの内容で更新されます。詳細については，第 8 章「共有オブジェクト・リポジトリの結合」を参照してください。
- 共有オブジェクト・リポジトリが編集可能であり，アクティブ・ウィンドウであることを確認したら，ローカルのオブジェクト・リポジトリを変更するのと同じ方法で共有オブジェクト・リポジトリを変更できます。ローカル・リポジトリの場合と同じ方法で，共有オブジェクト・リポジトリにオブジェクトを追加できるだけでなく，［ナビゲートして学習］オプションを使用して共有オブジェクト・リポジトリにオブジェクトを追加することもできます。

詳細については，次を参照してください。

- 240 ページ「オブジェクト・リポジトリの編集」
- 242 ページ「オブジェクト・リポジトリ・マネージャの使用によるコンポーネントへのテスト・オブジェクトの追加」
- 152 ページ「オブジェクト・リポジトリへのテスト・オブジェクトの追加」
- 242 ページ「[ナビゲートして学習］オプションの使用によるテスト・オブジェクトの追加」
- 164 ページ「オブジェクト・リポジトリ内のオブジェクトのコピー，貼り付け，および移動」
- 167 ページ「オブジェクト・リポジトリからのオブジェクトの削除」

オブジェクト・リポジトリの編集

本項で説明する機能は，［**オブジェクト リポジトリ マネージャ**］ウィンドウで作業している場合にのみ利用できます。

標準設定では，オブジェクト・リポジトリは読み取り専用モードで開きます。編集可能な形式で開くには，開くときに［共有オブジェクト リポジトリを開

く］ダイアログ・ボックスで［**読み取り専用モードで開く**］チェック・ボックスをクリアします。

オブジェクト・リポジトリを読み取り専用モードで開いた場合，変更を加えるにはオブジェクト・リポジトリの編集を有効にする必要があります。オブジェクト・リポジトリを表示するだけなら，あるいは，別のオブジェクト・リポジトリにオブジェクトをコピーするだけなら，編集可能にする必要はありません。

オブジェクト・リポジトリを編集可能にすると，ほかのユーザが変更できないように，オブジェクト・リポジトリがロックされます。ほかのユーザがオブジェクト・リポジトリを変更できるようにするには，その前にロックを解除する必要があります（編集モードを無効にするか，オブジェクト・リポジトリを閉じます）。オブジェクト・リポジトリがすでに別のユーザによってロックされている場合，読み取り専用形式で保存されている場合，またはオブジェクト・リポジトリを開くために必要な権限がユーザにない場合，編集可能にすることはできません。

QuickTest の以前のバージョンをお使いのユーザの方へ：ファイル・システムに保管されたオブジェクト・リポジトリ，およびバージョン 9.0 以前の QuickTest を使用して作成されたオブジェクト・リポジトリを編集する場合は，編集する前に QuickTest で最新の形式に変換する必要があります。変換したくない場合は，読み取り専用形式で表示できます。ファイルを変換して保存すると，そのファイルは以前のバージョンの QuickTest では使用できません。

オブジェクト・リポジトリを編集可能にするには，次の手順を実行します。

1 編集対象オブジェクト・リポジトリがアクティブ・ウィンドウであることを確認します。

2［**ファイル**］＞［**編集を有効化**］を選択するか，［**編集を有効化**］ボタンをクリックします。オブジェクト・リポジトリが編集可能になります。

オブジェクト・リポジトリ・マネージャの使用によるコンポーネントへのテスト・オブジェクトの追加

本項で説明する機能は，ローカル・オブジェクト・リポジトリのオブジェクトに関しては［**オブジェクト リポジトリ**］ウィンドウで，共有オブジェクト・リポジトリのオブジェクトに関しては［**オブジェクト リポジトリ マネージャ**］で利用できます。

テスト・オブジェクトをオブジェクト・リポジトリ・マネージャからコンポーネントにドラッグ・アンド・ドロップできます。テスト・オブジェクトをコンポーネントにドラッグ・アンド・ドロップすると，QuickTest によって，そのテスト・オブジェクトの標準操作を含んだステップがコンポーネントに挿入されます。チェックポイント・オブジェクトまたは出力オブジェクトをオブジェクト・リポジトリ・マネージャからドラッグ・アンド・ドロップすることはできません。

たとえば，ボタン・オブジェクトをコンポーネントにドラッグ・アンド・ドロップすると，そのボタン・オブジェクトを使って，**クリック**操作（ボタン・オブジェクトに対する標準の操作）を含むステップがコンポーネントに追加されます。

ほかの場所からテスト・オブジェクトをドラッグ・アンド・ドロップすることもできます。詳細については，次を参照してください。

➤ 831 ページ「［使用可能なキーワード］表示枠について」

➤ 129 ページ「［オブジェクト リポジトリ］ウィンドウ」

［ナビゲートして学習］オプションの使用によるテスト・オブジェクトの追加

本項で説明する機能は，［**オブジェクト リポジトリ マネージャ**］ウィンドウで作業している場合にのみ利用できます。

［**ナビゲートして学習**］オプションを使用すると，アプリケーションをナビゲートしながら複数のテスト・オブジェクトを共有オブジェクト・リポジトリに追加できます。

学習対象のウィンドウを選択するたびに，定義済みのオブジェクト・フィルタに従って，選択したウィンドウとその子孫オブジェクトがアクティブな共有オブジェクト・リポジトリに追加されます。オブジェクト・フィルタの定義は，要件を満たすためにいつでも変更できます。オブジェクト・フィルタは，［**ナビゲートして学習**］オプションと［**オブジェクトの追加**］オプションの両方で使用されます。定義した設定は，QuickTest がオブジェクトを学習するときに，これらの両方で使用されます。フィルタ定義の変更の詳細については，158 ページ「[オブジェクト フィルタの定義］ダイアログ・ボックスについて」を参照してください。

注 :［ナビゲートして学習］オプションは，混合階層（異なる環境のオブジェクトが含まれているオブジェクト階層）の環境ではサポートされていません。たとえば，Browser("Homepage").Page("Welcome").AcxButton("Save") または Dialog("Edit").AcxEdit("MyEdit") などです。混合階層内のオブジェクトを追加するには，152 ページ「オブジェクト・リポジトリへのテスト・オブジェクトの追加」で説明している，ほかのオプションを使用してください。

［**ナビゲートして学習**］オプションを使用してオブジェクトを学習するときは，次のキーボード・ショートカットを使用できます。

- ［**フォーカス ウィンドウを学習**］: ENTER
- ［**オブジェクト フィルタの定義**］: CTRL+F
- ［**ヘルプ**］: F1
- ［**オブジェクト リポジトリ マネージャに戻る**］: ESC

注 :［**ナビゲートして学習**］オプションを使用しているときは，最小化されているウィンドウは学習対象になりません。

［ナビゲートして学習］オプションを使用してテスト・オブジェクトを追加するには，次の手順を実行します。

1 オブジェクト・リポジトリ・マネージャで，オブジェクトを追加する対象となるオブジェクト・リポジトリがアクティブ・ウィンドウであること，および編集可能であることを確認します。

2 ［**オブジェクト**］＞［**ナビゲートして学習**］を選択するか，F6 キーを押します。［**ナビゲートして学習**］ツールバーが表示されます。

注：オブジェクト・リポジトリに初めてオブジェクトを追加するとき，次に進む前にフィルタ定義を変更する必要がある場合があります。現在のフィルタ定義は，［**オブジェクト フィルタの定義**］ボタンのツールチップに表示できます（ボタン名に続いて括弧内に表示されます）。フィルタの定義は，［**オブジェクト フィルタの定義**］ボタンをクリックするか，CTRL キーを押しながら F キーを押すことで，いつでも変更できます。詳細については，158 ページ「［オブジェクト フィルタの定義］ダイアログ・ボックスについて」を参照してください。

---

3 オブジェクト・リポジトリに追加する親オブジェクト（Browser，Dialog，Window など）をクリックしてフォーカスを当てます。ツールバーの［**学習**］ボタンが有効になります。

4 ［**学習**］ボタンをクリックするか，［**移動と学習**］ツールバーにフォーカスを当てて ENTER キーを押します。フォーカスのあるウィンドウが点滅し，定義されているフィルタに基づいてオブジェクトとその子孫がオブジェクト・リポジトリに追加されます。

5 アプリケーションで，追加する次のウィンドウにナビゲートして手順 4 を繰り返します。

6 必要なオブジェクトをオブジェクト・リポジトリに追加したら，［**ナビゲートして学習**］ツールバーで［**閉じる**］ボタンをクリックするか，ESC キーを押します。［**ナビゲートして学習**］ツールバーが閉じ，オブジェクト・リポジトリ・マネージャが再表示され，先ほど共有オブジェクト・リポジトリに追加したオブジェクトが表示されます。

リポジトリ・パラメータを使用した作業

リポジトリ・パラメータを使用すると，特定のプロパティ値をパラメータ化するように指定できますが，実際のパラメータ化は，パラメータ化された認識プロパティ値が含まれているオブジェクト・リポジトリと関連付けられた各コンポーネントで定義されます。

リポジトリ・パラメータは，動的に変化するオブジェクトを対象にコンポーネントを作成し，実行する場合に便利です。オブジェクトがアプリケーション内で頻繁に更新される場合，またはデータベースなどの動的コンテンツによってプロパティ値が設定される場合，オブジェクトは動的に変化することがあります。

たとえば，ローカライズされたアプリケーションにおいて，テキストのプロパティ値がユーザ・インタフェースの言語に応じて変化するボタンがある場合があります。リポジトリ・パラメータを使用して名前のプロパティ値をパラメータ化した後，オブジェクト・リポジトリを使用する各コンポーネントにおいて，プロパティ値をどこから取得するかを指定できます。たとえば，このオブジェクト・リポジトリを使用するあるコンポーネントではプロパティ値をコンポーネントのパラメータから取得し，別のコンポーネントではローカル・パラメータから取得し，3 番目のコンポーネントでは定数値として指定できます。

特定のオブジェクト・リポジトリに対するリポジトリ・パラメータはすべて，［リポジトリ パラメータの管理］ダイアログ・ボックスを使用して定義します。各リポジトリ・パラメータは，任意で標準設定値を指定し，わかりやすい説明とともに定義します。詳細については，246 ページ「リポジトリ・パラメータの管理」を参照してください。

標準設定値が定義されていないリポジトリ・パラメータを持つオブジェクト・リポジトリを使用するコンポーネントを開くと，［欠落リソース］表示枠に，割当が必要なリポジトリ・パラメータが存在することを示す指示が表示されます。その場合は，そのコンポーネント内のリポジトリ・パラメータを必要に応じて割り当てることができます。また，標準設定値があるリポジトリ・パラメータの割り当てを行ったり，すでに割り当てのあるリポジトリ・パラメータの割り当てを変更したりもできます。リポジトリ・パラメータの割り当ての詳細については，854 ページ「未割り当ての共有オブジェクト・リポジトリ・パラメータ値の処理」を参照してください。

リポジトリ・パラメータの管理

本項で説明する機能は，［**オブジェクト リポジトリ マネージャ**］ウィンドウで作業している場合にのみ利用できます。

［リポジトリ パラメータの管理］ダイアログ・ボックスでは，1 つの共有オブジェクト・リポジトリを対象にリポジトリ・パラメータを追加，編集，および削除できます。

リポジトリ・パラメータを管理するには，次の手順を実行します。

1 パラメータを管理する対象となるオブジェクト・リポジトリがアクティブ・ウィンドウであることを確認します。

2 オブジェクト・リポジトリが読み取り専用形式の場合は，［**ファイル**］＞［**編集を有効化**］を選択するか，［**編集を有効化**］ボタンをクリックします。オブジェクト・リポジトリが編集可能になります。

3 ［**ツール**］＞［**リポジトリ パラメータの管理**］を選択するか，［**リポジトリ パラメータの管理**］ボタンをクリックします。［リポジトリ パラメータの管理］ダイアログ・ボックスが開きます。

［リポジトリ パラメータの管理］ダイアログ・ボックスには，次の情報およびオプションが表示されます。

オプション	**詳細**
［リポジトリ名］	リポジトリ・パラメータを管理する対象となるオブジェクト・リポジトリの名前およびパスが表示されます。
	新しいリポジトリ・パラメータを追加します。詳細については，247 ページ「リポジトリ・パラメータの追加」を参照してください。
	現在選択されているリポジトリ・パラメータを削除します。詳細については，249 ページ「リポジトリ・パラメータの削除」を参照してください。
パラメータ・リスト（［**名前**］，［**標準設定の値**］，および［**詳細**］）	このオブジェクト・リポジトリに現在定義されているリポジトリ・パラメータのリストが表示されます。パラメータ・リストでは，パラメータの標準設定の値および説明を直接変更できます。詳細については，249 ページ「リポジトリ・パラメータの変更」を参照してください。
［**リポジトリの中で検索**］ボタン	オブジェクト・リポジトリ・ツリー内で，選択されているリポジトリ・パラメータを使用している最初のテスト・オブジェクトを検索して強調表示します。このボタンを再度クリックすると，選択したパラメータの次の出現箇所を検索できます。

リポジトリ・パラメータの追加

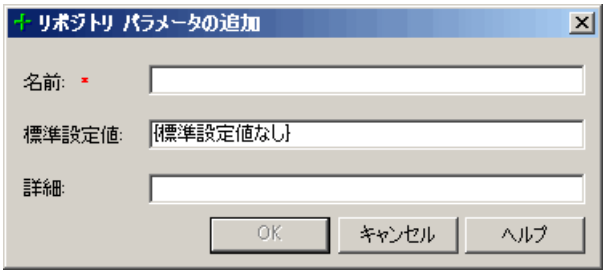
本項で説明する機能は，［**オブジェクト リポジトリ マネージャ**］ウィンドウで作業している場合にのみ利用できます。

［リポジトリ パラメータの追加］ダイアログ・ボックスでは，新しいリポジトリ・パラメータを定義できます。パラメータの標準設定値を指定したり，パラメータがコンポーネントのステップで使用されているときにその識別に役立つわかりやすい説明を指定することもできます。

リポジトリ・パラメータの詳細については，245 ページ「リポジトリ・パラメータを使用した作業」を参照してください。

リポジトリ・パラメータを追加するには，次の手順を実行します。

1 ［**リポジトリ パラメータの管理**］ダイアログ・ボックスで，［**リポジトリ パラメータの追加**］ボタンをクリックします。［リポジトリ パラメータの追加］ダイアログ・ボックスが開きます。



2 ［**名前**］ボックスに，パラメータに対してわかりやすい名前を指定します。パラメータ名は英字（ローマ字）で始める必要があり，英字（ローマ字），数字，およびアンダースコアのみを使用できます。

3 ［**標準設定の値**］ボックスに，リポジトリ・パラメータに使用する標準設定値を指定できます。この値は，このオブジェクト・リポジトリを使用するコンポーネントの値またはパラメータ・タイプに，リポジトリ・パラメータを割り当てなかった場合に使用されます。標準設定の値を指定しなかった場合，リポジトリ・パラメータは，この共有オブジェクト・リポジトリを使用するコンポーネントにおいて未割り当てとして表示されます。

ヒント：標準設定の値を指定した場合，それを後で削除するには，［リポジトリ パラメータの管理］ダイアログ・ボックスの中で該当するパラメータの［**標準設定の値**］セルをクリックした後，［**標準設定値をクリア**］ボタンをクリックします。セル内に **{ 標準設定値なし }** というテキストが表示されます。

---

4 ［**詳細**］ボックスに，リポジトリ・パラメータの説明を入力できます。この説明は，コンポーネントの中でリポジトリ・パラメータを割り当てるときに，パラメータの識別に役立ちます。

5 ［**OK**］をクリックして，［リポジトリ パラメータの管理］ダイアログ・ボックスのパラメータのリストにパラメータを追加します。

リポジトリ・パラメータの変更

本項で説明する機能は，［**オブジェクト リポジトリ マネージャ**］ウィンドウで作業している場合にのみ利用できます。

［リポジトリ パラメータの管理］ダイアログ・ボックスでは，リポジトリ・パラメータの標準設定値やリポジトリ・パラメータの説明を直接変更できます。ただし，リポジトリ・パラメータの名前は変更できません。

リポジトリ・パラメータを変更するには，次の手順を実行します。

1 ［リポジトリ パラメータの管理］ダイアログ・ボックスの中で，必要なパラメータを選択します。

2 標準設定値を変更するには，必須パラメータの［**標準設定値**］セルをクリックします。標準設定の値は，新しい値を入力して変更することも，［**標準設定値をクリア**］ボタンをクリックして削除することもできます。標準設定の値を削除すると，セル内に **{ 標準設定値なし }** というテキストが表示されます。標準設定の値を指定しなかった場合，リポジトリ・パラメータは，この共有オブジェクト・リポジトリを使用するコンポーネントにおいて未割り当てとして表示されます。

注 : テキストを手作業で削除しても，標準設定の値は削除されません。空文字列が標準設定値となります。標準設定の値を削除するには，［**標準設定値をクリア**］ボタンをクリックする必要があります。

---

3 パラメータの説明を変更するには，必須パラメータの［**詳細**］セルをクリックして，必要な説明を入力します。

リポジトリ・パラメータの削除

本項で説明する機能は，［**オブジェクト リポジトリ マネージャ**］ウィンドウで作業している場合にのみ利用できます。

リポジトリ・パラメータの定義は，不要になった場合は削除できます。テスト・オブジェクトの定義で使用されているリポジトリ・パラメータを削除すると，パラメータがなくなったにもかかわらず，認識プロパティ値はパラメータに割り当てられたままです。したがって，リポジトリ・パラメータを削除する前に，パラメータがどのテスト・オブジェクト記述でも使用されていないこと

を確認する必要があります。使用されていると，これらのテスト・オブジェクトを使用するステップがあるコンポーネントを実行すると失敗します。

ヒント：［リポジトリ パラメータの管理］ダイアログ・ボックスの［**リポジトリの中で検索**］ボタンを使用すれば，リポジトリ・パラメータが使用されている場所を確認できます。

リポジトリ・パラメータを削除するには，次の手順を実行します。

1 ［リポジトリ パラメータの管理］ダイアログ・ボックスの中で，パラメータ名の左側にある選択領域をクリックすることで，削除対象のリポジトリ・パラメータを選択します。

2 ［**リポジトリ パラメータを削除**］ボタンをクリックします。選択したリポジトリ・パラメータが削除されます。

オブジェクトの詳細の変更

ドキュメント・ウィンドウ右下にある共有オブジェクト・リポジトリのオブジェクト詳細領域では，実行セッション中にオブジェクトの識別に使用されるプロパティとプロパティ値，またはチェックポイントや出力オブジェクトのプロパティの表示や変更ができます。

共有オブジェクト・リポジトリが編集可能であり，アクティブ・ウィンドウであることを確認したら，ローカル・オブジェクトを変更するときと同じ方法で，共有オブジェクト・リポジトリ内のオブジェクトの詳細を変更します。詳細については，次を参照してください。

- 184 ページ「テスト・オブジェクト記述へのプロパティの追加」
- 188 ページ「新規認識プロパティの定義」
- 178 ページ「アプリケーション内のオブジェクトから認識プロパティを更新」
- 181 ページ「テスト・オブジェクトの標準必須プロパティの復元」
- 190 ページ「テスト・オブジェクト記述からのプロパティの削除」
- 191 ページ「序数識別子の指定」

➤ 182 ページ「テスト・オブジェクトの名前の変更」

注：必要に応じて，［**編集**］＞［**元に戻す**］や［**編集**］＞［**やり直し**］オプションまたは［**元に戻す**］ボタンや［**やり直し**］ボタンを使用して，変更の取り消しや繰り返しを行うことができます。［**元に戻す**］および［**やり直し**］オプションは，アクティブなドキュメントを対象とします。リポジトリを保存すると，保存前にファイルに対して行った操作の取り消しや，やり直しはできません。

オブジェクト・リポジトリ・マネージャを使用して，共有オブジェクト・リポジトリ内のテスト・オブジェクト記述のプロパティ値を指定します。共有オブジェクト・リポジトリにあるオブジェクトのプロパティ値の指定時に使用可能なオプションと，ローカル・リポジトリにあるオブジェクトのプロパティの指定時に使用可能なオプションは異なります。共有オブジェクト・リポジトリにあるオブジェクトのプロパティ値の指定の詳細については，251 ページ「プロパティ値の指定」を参照してください。

プロパティ値の指定

本項で説明する機能は，［**オブジェクト リポジトリ マネージャ**］ウィンドウで作業している場合にのみ利用できます。

テスト・オブジェクト記述に含まれるプロパティの値を指定または変更できます。定数値を使用して値を指定することも（単純値または正規表現を含んだ定数値のいずれか），リポジトリ・パラメータを使用して値をパラメータ化することもできます。リポジトリ・パラメータの詳細については，245 ページ「リポジトリ・パラメータを使用した作業」を参照してください。

また，チェックポイントまたは出力オブジェクトのプロパティの値を指定または変更することもできます。

テスト・オブジェクトのプロパティ値の指定と変更

テスト・オブジェクトのプロパティの値は，［**テスト オブジェクトの詳細**］領域で指定または変更します。

テスト・オブジェクトのプロパティ値を指定するには，次の手順を実行します。

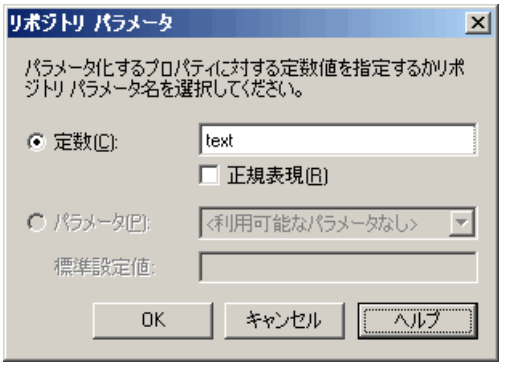
1 プロパティ値を指定するテスト・オブジェクトを選択します。

2 ［**テスト オブジェクトの詳細**］領域で，必要なプロパティの［**値**］セルをクリックします。

3 次のいずれかの方法でプロパティ値を指定します。

- 単純な定数値を指定する場合は，［**値**］セルに値を入力します。［**値**］セルに定数値を指定した場合は，以降の手順は必要ありません。次の説明のように，［リポジトリ パラメータ］ダイアログ・ボックスの中で正規表現を使用して定数値を指定することもできます。

- リポジトリ・パラメータを使用して値をパラメータ化するには，［**値**］セルの中でパラメータ化ボタンをクリックします。［リポジトリ パラメータ］ダイアログ・ボックスが開きます。



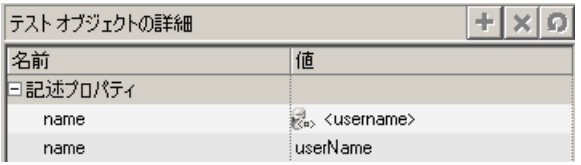
4 次のいずれかのオプションを選択して，プロパティの値を指定します。

- ［**定数**］ラジオ・ボタンを選択して，定数値を指定します。［**テスト オブジェクトの詳細**］領域の［**値**］セルに，定数値を直接入力することもできます。定数値の中で正規表現を使用した場合は，［**正規表現**］チェック・ボックスを選択します。
- ［**パラメータ**］ラジオ・ボタンを選択して，定義済みのパラメータのリストからリポジトリ・パラメータを選択します。パラメータに対して標準設定の値が定義されていれば，それも表示されます。

注：リポジトリ・パラメータを定義するには［リポジトリ パラメータの管理］ダイアログ・ボックスを使用します。詳細については，246 ページ「リポジトリ・パラメータの管理」を参照してください。

---

5 ［**OK**］をクリックして［リポジトリ パラメータ］ダイアログ・ボックスを閉じます。値をパラメータ化した場合は，次に示すように，［**テスト オブジェクトの詳細**］領域の［**値**］カラムにパラメータ名がアイコンとともに表示されます。それ以外の場合には，指定した定数値が［**値**］カラムに表示されます。



チェックポイント・オブジェクトのプロパティ値の指定と変更

チェックポイント・オブジェクトのプロパティの値は，［**オブジェクトのプロパティ**］表示枠で指定または変更します。

チェックポイント・オブジェクトのプロパティの値を指定または変更するには，次の手順を実行します。

1 プロパティ値を指定または変更するチェックポイント・オブジェクトを［**チェックポイントと出力オブジェクト**］ツリーから選択します。

2 関連するチェックポイント・プロパティのダイアログ・ボックスの場合と同じ方法で，チェックポイント・オブジェクトのプロパティの値を指定または変更します。

チェックポイント・オブジェクトのプロパティの値の指定と変更に関する詳細については，以下を参照してください。

➤ 570 ページ「[チェックポイントのプロパティ］ダイアログ・ボックスについて」

➤ 585 ページ「ビットマップ・チェックポイントのプロパティ・ダイアログ・ボックス」

出力オブジェクトのプロパティ値の指定と変更

出力オブジェクトのプロパティの値は，［**オブジェクトのプロパティ**］表示枠で指定または変更します。

出力オブジェクトのプロパティの値を指定または変更するには，次の手順を実行します。

1 プロパティ値を指定または変更する出力オブジェクトを［**チェックポイントおよび出力オブジェクト**］ツリーから選択します。

2 関連する出力値のプロパティのダイアログ・ボックスの場合と同じ方法で，出力オブジェクトのプロパティの値を指定または変更します。

出力オブジェクトのプロパティの値の指定と変更に関する詳細については，597 ページ「標準出力値の定義」を参照してください。

テスト・オブジェクトの検索

本項で説明する機能は，ローカル・オブジェクト・リポジトリのオブジェクトに関しては［**オブジェクト リポジトリ**］ウィンドウで，共有オブジェクト・リポジトリのオブジェクトに関しては［**オブジェクト リポジトリ マネージャ**］で利用できます。

オブジェクト・リポジトリ内にある特定のテスト・オブジェクトを検索するには，複数の方法があります。テスト・オブジェクトは種類に基づいて検索できます。たとえば，特定のエディット・ボックスを検索したり，アプリケーション内のオブジェクトにポイントすることで，リポジトリ内の同じオブジェクトを自動的に強調表示できます。特定のプロパティ値はほかのプロパティ値に置換できます。たとえば，プロパティ値 userName を user name という値に置換できます。また，オブジェクト・リポジトリ内のオブジェクトを選択して，アプリケーション内でそのオブジェクトを強調表示させることで，それがどのオブジェクトであるかを確認できます。

共有オブジェクト・リポジトリがアクティブ・ウィンドウであることを確認したら，ローカル・オブジェクト・リポジトリで行うのと同じ方法で，共有オブジェクト・リポジトリ内のオブジェクトを検索します。プロパティ値を置き換える場合は，オブジェクト・リポジトリが編集可能であることも確認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。

- 168 ページ「オブジェクト・リポジトリ内のオブジェクトの検索」
- 171 ページ「アプリケーション内のオブジェクトの強調表示」
- 172 ページ「オブジェクト・リポジトリ内のテスト・オブジェクトの検索」

結合操作の実行

本項で説明する機能は，［**オブジェクト リポジトリ マネージャ**］ウィンドウで作業している場合にのみ利用できます。

オブジェクト・リポジトリ結合ツールでは，オブジェクト・リポジトリ・マネージャの［**ローカル リポジトリから更新**］オプションを使用して（［**ツール**］＞［**ローカル リポジトリから更新**］），テスト・オブジェクトを 1 つ以上のコンポーネントのローカル・オブジェクト・リポジトリから共有オブジェクト・リポジトリに結合できます。たとえば，コンポーネントでテスト・オブジェクトをローカルに学習し，共有オブジェクト・リポジトリを使用するコンポーネントのすべてで利用できるように，それらのテスト・オブジェクトを共有オブジェクト・リポジトリに追加することが考えられます。また，オブジェクト・リポジトリ結合ツールを使用して，2 つの共有オブジェクト・リポジトリを 1 つの共有オブジェクト・リポジトリに結合することもできます。

オブジェクト・リポジトリ結合ツールを開くには，オブジェクト・リポジトリ・マネージャで，［**ツール**］＞［**オブジェクト リポジトリ結合ツール**］を選択します。結合操作の実行と，ローカル・オブジェクトを使用したオブジェクト・リポジトリの更新の詳細については，第 8 章「共有オブジェクト・リポジトリの結合」を参照してください。

注：

- オブジェクト・リポジトリ結合ツールを開いている間は，オブジェクト・リポジトリ・マネージャを使用した作業は行えません。
- オブジェクト・リポジトリの結合ツールでは，チェックポイントおよび出力オブジェクトは結合されません。

インポート操作およびエクスポート操作の実行

オブジェクト・リポジトリは，XML ファイルに対してインポートおよびエクスポートができます。XML は構造化されたアクセス可能な形式であり，任意の XML エディタを使用してオブジェクト・リポジトリに変更を加え，QuickTest にインポートして戻すことがきます。オブジェクト・リポジトリに必要な形式は，**HP QuickTest Professional オブジェクト・リポジトリ・スキーマのヘルプ**（［**ヘルプ**］＞［**QuickTest Professional ヘルプ**］＞［**HP QuickTest Professional Adovanced Reference**］＞［**HP QuickTest Professional Object Repository Schema**］）で，または保存したオブジェクト・リポジトリをエクスポートして確認できます。

ファイルのインポートおよびエクスポートは，ファイル・システムまたは Quality Center プロジェクト（QuickTest が Quality Center に接続されている場合）のどちらに対しても行うことができます。

QuickTest または［オブジェクト リポジトリ マネージャ］から Quality Center プロジェクトに接続するには，［**ファイル**］＞［**Quality Center への接続**］を選択するか，［**Quality Center への接続**］ボタンをクリックします。Quality Center への接続の詳細については，46 ページ「Quality Center プロジェクトへの接続」を参照してください。

詳細については，次を参照してください。

- 257 ページ「XML からのインポート」
- 258 ページ「XML へのエクスポート」
- 259 ページ「XML ファイルの構造について」

XML からのインポート

本項で説明する機能は，［**オブジェクト リポジトリ マネージャ**］ウィンドウで作業している場合にのみ利用できます。

必要な形式を使用して作成された XML ファイルをオブジェクト・リポジトリとしてインポートできます。XML 形式の詳細については，259 ページ「XML ファイルの構造について」を参照してください。XML ファイルは，オブジェクト・リポジトリ・マネージャを使用して XML 形式にエクスポートしたオブジェクト・リポジトリ，または QuickTest Siebel Test Express や独自に作成したユーティリティなどのツールを使用して作成した XML ファイルです。XML の構造および形式に準拠する必要があります。

ヒント：必要な XML の構造と形式を確認するには，**HP QuickTest Professional オブジェクト・リポジトリ・スキーマのヘルプ**（［**ヘルプ**］＞［**QuickTest Professional ヘルプ**］＞［**HP QuickTest Professional 詳細リファレンス**］＞［**HP QuickTest Professional オブジェクト リポジトリ スキーマ**］）を参照してください。また，既存の共有オブジェクト・リポジトリを XML にエクスポートして，その XML ファイルを参考にすることもできます。詳細については，258 ページ「XML へのエクスポート」を参照してください。

XML からインポートするには，次の手順を実行します。

1 ［**ファイル**］＞［**XML からインポート**］を選択します。［XML ファイルを開く］ダイアログ・ボックスが開きます。

注：XML ファイルからオブジェクト・リポジトリの内容をインポートするときは，チェックポイントと出力オブジェクトは含まれません。

---

2 サイドバーで，ファイルの場所を選択します（File System や Quality Center Test Resources など）。インポートする XML ファイルを参照して選択し，［**開く**］をクリックします。

XML ファイルがインポートされると，サマリ・メッセージ・ボックスが開き，指定のファイルから正常にインポートされたオブジェクト数，パラメータ数，およびメタデータ数に関する情報が表示されます。

3［**OK**］をクリックし，メッセージ・ボックスを閉じます。インポートされた XML ファイルが新しいオブジェクト・リポジトリとして開きます。これで，必要に応じて変更を加えたり，オブジェクト・リポジトリとして保存したりできます。

XML へのエクスポート

本項で説明する機能は，［**オブジェクト リポジトリ マネージャ**］ウィンドウで作業している場合にのみ利用できます。

オブジェクト・リポジトリのテスト・オブジェクトは XML ファイルにエクスポートできます。これにより，XML エディタを使用して編集したり，アクセス可能な柔軟な形式で保存したりできます。

XML にエクスポートするには，次の手順を実行します。

1 テスト・オブジェクトをエクスポートするオブジェクト・リポジトリがアクティブ・ウィンドウであることを確認します。

2 オブジェクト・リポジトリが保存されていることを確認します。

3［**ファイル**］＞［**XML へのテスト オブジェクトのエクスポート**］を選択します。［XML ファイルを保存］ダイアログ・ボックスが開きます。

注：XML ファイルにオブジェクト・リポジトリの内容をエクスポートするときは，チェックポイントと出力オブジェクトは含まれません。

---

4 サイドバーで，ファイルを保存する場所を選択します（File System や Quality Center Test Resources など）。

5 ファイルを保存するフォルダを参照し，そのフォルダを選択します。

6［**ファイル名**］ボックスにファイルの名前を入力し，［**保存**］をクリックします。

オブジェクト・リポジトリのテスト・オブジェクトが指定の XML ファイルにエクスポートされ，サマリ・メッセージ・ボックスが開き，指定のファイルへ正常にエクスポートされオブジェクトの数，パラメータ，およびメタデータの数に関する情報が表示されます

7［**OK**］をクリックし，メッセージ・ボックスを閉じます。これで XML ファイルを開き，任意の XML エディタを使用して表示または変更できます。

XML ファイルの構造について

QuickTest では，オブジェクト・リポジトリに対して定義された XML スキーマを使用します。XML 形式でオブジェクト・リポジトリ・ファイルを作成または変更するときは，このスキーマに従う必要があります。このファイルのスキーマは，**HP QuickTest Professional オブジェクト・リポジトリ・スキーマのヘルプ**（[**ヘルプ**] > [**QuickTest Professional ヘルプ**] > [**HP QuickTest Professional 詳細リファレンス**] > [**HP QuickTest Professional オブジェクト リポジトリ スキーマ**]）でドキュメント化されています。

オートメーションの使用によるオブジェクト・リポジトリの管理

QuickTest ではオブジェクト・リポジトリのオートメーション・オブジェクト・モデルが提供されているため，QuickTest 共有オブジェクト・リポジトリおよびそのコンテンツを QuickTest の外部から管理できます。オートメーション・オブジェクト・モデルを使用すると，スクリプティング・ツールを使用してオートメーションを介して QuickTest 共有オブジェクト・リポジトリにアクセスできます。

QuickTest Professional オートメーション・オブジェクト・モデルを使用して QuickTest の操作を自動化するのと同じように，オブジェクト・リポジトリのオートメーション・オブジェクト・モデルのオブジェクトとメソッドを使用して共有オブジェクト・リポジトリを管理するスクリプトを記述できます。オブジェクト・リポジトリ・マネージャを使用して手動でこれらの操作を実行する必要はありません。たとえば，テスト・オブジェクトの追加，削除，名前の変更，XML へのインポートとエクスポート，テスト・オブジェクトの取得とコピーなどを行えます。

テスト・オブジェクトを取得した後は，そのテスト・オブジェクト・クラスで利用できるメソッドとプロパティを使用して操作できます。たとえば，GetTOProperty メソッドと SetTOProperty メソッドを使用して，プロパティを取得および変更できます。利用可能なテスト・オブジェクト・メソッドおよびプロパティの詳細については，『**HP QuickTest Professional Object Model Reference**』（英語版）を参照してください。

オートメーション・プログラムは，同じタスクを複数回実行するのに，または複数のオブジェクト・リポジトリ上で実行するのに便利です。オートメーション・スクリプトはオートメーションをサポートする任意の言語および開発環境を使用して作成できます。たとえば，VBScript，JavaScript，Visual Basic，

Visual C++，または Visual Studio .NET を使用できます。オートメーションを使用して，QuickTest を制御する方法の一般的な情報については，911 ページ「QuickTest 操作のオートメーション」を参照してください。

QuickTest Professional オブジェクト・リポジトリ・オートメーション・リファレンスの使用について

『**QuickTest Professional Object Repository Automation Reference**』（英語版）は，QuickTest オブジェクト・リポジトリ・オートメーション・オブジェクト・モデルのオブジェクトとメソッドに関する詳細な説明，構文情報，使用例を提供するヘルプ・ファイルです。

各オートメーション・オブジェクトのヘルプ・トピックには，そのオブジェクトに関連するメソッドのリストおよび詳細が含まれています。メソッドのヘルプ・トピックには，詳細説明，構文，戻り値のタイプ，および引数値に関する情報が含まれています。

『**QuickTest Professional Object Repository Automation Reference**』（英語版）は，メインの QuickTest ヘルプから開くことができます（［**ヘルプ**］＞［**QuickTest Professional ヘルプ**］＞［**HP QuickTest Professional 詳細リファレンス**］＞［**HP QuickTest Professional オブジェクト・リポジトリ・オートメーション**］）。

注：ヘルプ・ファイルに記載されている構文と例は，VBScript 形式で書かれています。別の言語でオートメーション・プログラムを記述する場合，一部のメソッドの構文は，対応するヘルプ・トピックで紹介されている構文と異なる場合があります。使用する言語の構文については，お使いの開発環境に含まれているドキュメント，またはプログラミング言語の一般的なドキュメントを参照してください。

第 8 章

共有オブジェクト・リポジトリの結合

QuickTest Professional では，オブジェクト・リポジトリ結合ツールを使用して，2 つの共有オブジェクト・リポジトリを単一の共有オブジェクト・リポジトリに結合できます。

本章の内容

- 共有オブジェクト・リポジトリの結合について（262 ページ）
- オブジェクト・リポジトリ結合ツールについて（264 ページ）
- オブジェクト・リポジトリ結合ツールのコマンドの使用方法（270 ページ）
- 標準設定の定義（277 ページ）
- 2 つのオブジェクト・リポジトリの結合（281 ページ）
- ローカル・オブジェクト・リポジトリから共有オブジェクト・リポジトリを更新（283 ページ）
- 結合の統計情報の表示（290 ページ）
- オブジェクトの矛盾について（291 ページ）
- オブジェクトの矛盾の解決（294 ページ）
- ［ターゲット リポジトリ］表示枠に対するフィルタの設定（296 ページ）
- 特定のオブジェクトの検索（297 ページ）
- ターゲット・オブジェクト・リポジトリの保存（299 ページ）

共有オブジェクト・リポジトリの結合について

アプリケーションの同じ領域のテスト・オブジェクトが格納されている共有オブジェクト・リポジトリが複数ある場合，それらのテスト・オブジェクトを単独のオブジェクト・リポジトリに結合すると，保守が容易になり便利なことがあります。オブジェクト・リポジトリ・マネージャでオブジェクトを移動またはコピーすることで，こうした結合を行うことができます。しかし，アプリケーションの同じオブジェクトを表すテスト・オブジェクトが異なるオブジェクト・リポジトリにあり，各オブジェクト・リポジトリにあるそれらのオブジェクトの記述が同じではない場合，それらの矛盾に気づいて対応することが難しい可能性があります。

このような問題の解決に役立つのがオブジェクト・リポジトリ結合ツールです。このツールは，2 つの選択したオブジェクト・リポジトリをユーザに代わって結合し，記述が矛盾しているテスト・オブジェクトに対応するためのオプションを提供します。このツールを使用して，2 つの共有オブジェクト・リポジトリ（それぞれ「**一次**」オブジェクト・リポジトリおよび「**二次**」オブジェクト・リポジトリと呼びます）を結合して 3 つ目の新しいリポジトリ（「**ターゲット**」リポジトリと呼びます）を作成します一次オブジェクト・リポジトリと二次オブジェクト・リポジトリにあるオブジェクトは自動的に比較され，オブジェクト間の矛盾の標準の解決方法を定義した事前設定可能なルールに従って，ターゲット・リポジトリに追加されます。

結合処理の後，一次オブジェクト・リポジトリと二次オブジェクト・リポジトリにある元のオブジェクト（これらは変更されずに残っています）のほか，結合後のターゲット・オブジェクト・リポジトリにあるオブジェクトが，オブジェクト・リポジトリ結合ツールに視覚的に表示されます。矛盾のあったオブジェクトは強調表示されます。ターゲット・オブジェクト・リポジトリでオブジェクトを選択すると，その矛盾の詳しい説明が表示されます。オブジェクト・リポジトリ結合ツールには，矛盾ごとに，標準の解決方法を維持するか，矛盾の解決方法を個別に変更するかを，必要に応じて決めることができる専用のオプションが用意されています。

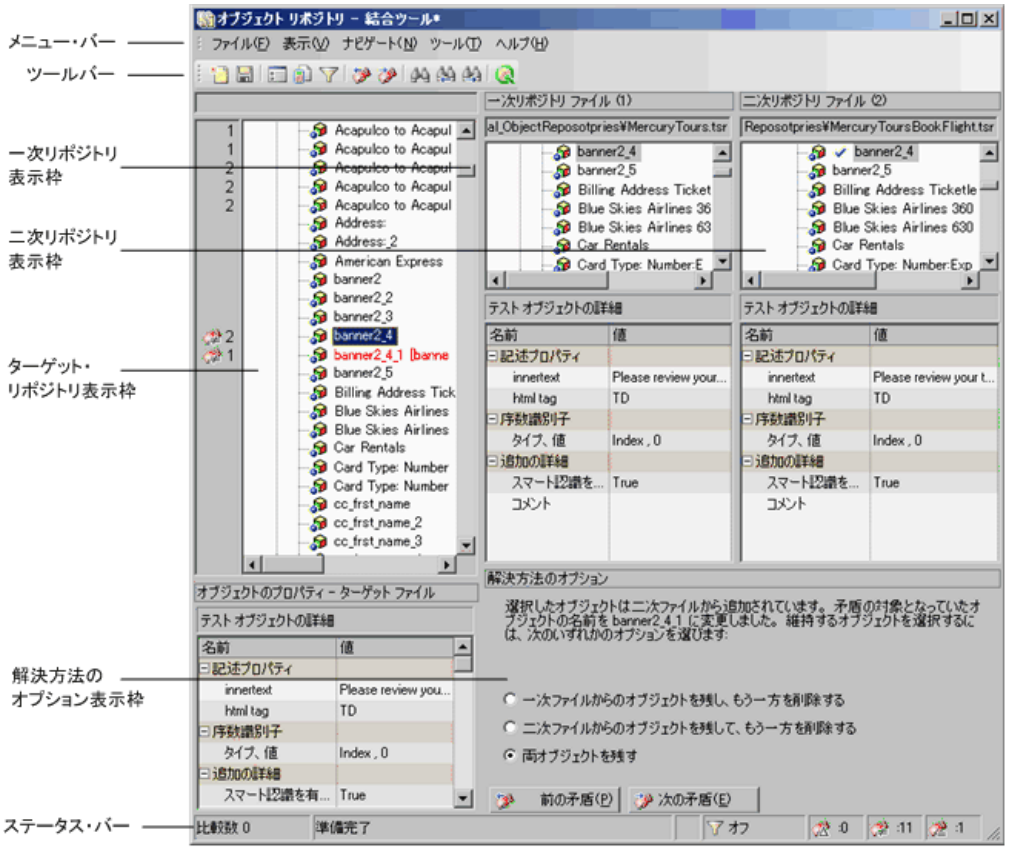
注：

- オブジェクト・リポジトリ結合ツールでは，一次オブジェクト・リポジトリと二次オブジェクト・リポジトリのチェックポイントおよび出力オブジェクトがターゲットの共有オブジェクト・リポジトリに結合されることはありません。これらのオブジェクトは，結合プロセスの完了後に，オブジェクト・リポジトリ・マネージャを使用してターゲット・オブジェクト・リポジトリにコピーまたは手作業で移動できます。
- オブジェクト・リポジトリ結合ツールが開いている間は，オブジェクト・リポジトリ・マネージャおよびオブジェクト・リポジトリ比較ツールを操作することはできません。オブジェクト・リポジトリ・マネージャの詳細については，第 7 章「オブジェクト・リポジトリの管理」を参照してください。

オブジェクト・リポジトリ結合ツールについて

オブジェクト・リポジトリ結合ツールを開くには，オブジェクト・リポジトリ・マネージャで，［**ツール**］＞［**オブジェクト リポジトリ結合ツール**］を選択します。

［オブジェクト リポジトリ ― 結合ツール］ウィンドウの例を次に示します。



注：オブジェクト・リポジトリ結合ツールによって与えられる表示の変更については，266 ページ「ビューの変更」を参照してください。

［オブジェクト リポジトリー結合ツール］ウィンドウには，次の主要な要素があります。

- **メニュー・バー**：オブジェクト・リポジトリ結合ツールのコマンドのメニューが表示されます。これらのコマンドについては，本章各所で説明します。メニュー・コマンドのショートカット・キーの詳細については，271 ページ「［ファイル］メニューのコマンド」を参照してください。
- **ツールバー**：よく使用するメニュー・コマンドのボタンがあります。オブジェクト・リポジトリの結合，管理，および保存を行うことができます。ツールバー・ボタンについては，270 ページ「ツールバー・コマンドの使用方法」を参照してください。
- **［ターゲット リポジトリ］表示枠**：一次オブジェクト・リポジトリおよび二次オブジェクト・リポジトリから結合されたオブジェクトが表示されます。［ターゲット リポジトリ］表示枠で選択したオブジェクトのプロパティが表示されるターゲット・リポジトリ・オブジェクトのプロパティ表示枠は，表示と非表示を切り替えることもできます。詳細については，266 ページ「［ターゲット リポジトリ］表示枠」を参照してください。
- **［一次リポジトリ］表示枠**：一次オブジェクト・リポジトリにあるオブジェクトが表示されます。詳細については，268 ページ「一次および二次リポジトリ表示枠」を参照してください。
- **［二次リポジトリ］表示枠**：二次オブジェクト・リポジトリにあるオブジェクトが表示されます。詳細については，268 ページ「一次および二次リポジトリ表示枠」を参照してください。
- **解決方法のオプション表示枠**：［ターゲット オブジェクト リポジトリ］表示枠にあるオブジェクトのソース，矛盾，および解決方法の詳細が表示され，選択した矛盾の解決方法を変更できます。詳細については，268 ページ「解決方法のオプション表示枠」を参照してください。
- **ステータス・バー**：［ターゲット オブジェクト リポジトリ］表示枠の中で選択したオブジェクトのソース，矛盾，および解決方法の詳細のほか，フィルタのステータスと，アイコンの凡例が表示されます。詳細については，269 ページ「ステータス・バー」を参照してください。

ビューの変更

オブジェクト・リポジトリ結合ツールに表示されるビューを自分が作業しやすいように変更することができます。

- ➤ オブジェクト・リポジトリ結合ツール・ウィンドウ内で表示枠のサイズを変更するには，表示枠の縁をドラッグします。
- ➤ オブジェクト・リポジトリ結合ツールでこれらの表示枠の表示 / 非表示を切り替えるには，［**表示**］メニューから［**一次リポジトリ**］，［**二次リポジトリ**］，［**ターゲット リポジトリ オブジェクトのプロパティ**］，または［**解決方法のオプション**］を選択します。
- ➤ オブジェクト・リポジトリ結合ツールを開くたびに表示される標準設定のビューとして現在のビューを設定するには，［**表示**］＞［**標準のレイアウトに設定**］を選択します。変更を加えた後に表示を標準設定に戻すには，［**表示**］＞［**標準のレイアウトを復元**］を選択します。

［ターゲット リポジトリ］表示枠

［ターゲット オブジェクト リポジトリ］表示枠には，一次オブジェクト・リポジトリおよび二次オブジェクト・リポジトリから結合されたオブジェクトの階層のほか，それらのオブジェクトのプロパティと値が表示されます。オブジェクト階層の左側のカラムには，各オブジェクトのソース・ファイルが表示され (一次ファイルの場合は **1** と表示され，二次ファイルの場合は **2** と表示されます)，矛盾がある場合は矛盾のタイプを表すアイコンが表示されます。

ターゲット・オブジェクト・リポジトリを保存すると，そのファイル・パスがオブジェクト階層の上に表示されます。

注 : オブジェクトのステータスをひと目で確認できるように，ターゲット・オブジェクト・リポジトリ内のオブジェクト名のテキストの色を，各オブジェクトのソースと，矛盾の原因となったかどうかに基づいて，設定することができます。詳細については，279 ページ「色の設定」を参照してください。

［ターゲット オブジェクト リポジトリ］表示枠には次の機能があります。

- ターゲット・オブジェクト・リポジトリ内のオブジェクトを選択すると，一次ソース・ファイル階層または二次ソース・ファイル階層，あるいはその両方にある対応するオブジェクトが探し出され，チェック・マークによって示されます。
- ターゲット・オブジェクト・リポジトリ内のオブジェクトを選択すると，そのプロパティと値が，［ターゲット オブジェクト リポジトリ］表示枠（［**表示**］＞［**ターゲット リポジトリ オブジェクトのプロパティ**］）の一番下にある［**オブジェクトのプロパティ - ターゲット ファイル**］領域に表示されます。
- 結合の結果として矛盾が生じた場合は，ターゲット・オブジェクト・リポジトリ内で，矛盾のあるオブジェクトの左側にアイコンが表示されます。アイコンの上にポインタを置くと，矛盾のタイプを説明するツールチップが表示されます。
- オブジェクトを右クリックすると，ショートカット・メニューが開きます。オプションを展開したり，ターゲット・オブジェクト・リポジトリの階層全体を折りたたんだり，適用できる場合は矛盾の解決方法とその結果を変更できます。
- ノードをダブルクリックすると，ノードの階層を展開したり折りたたむことができます。また，［**表示**］メニューの［**すべて折りたたみ**］または［**すべて展開**］を選択しても，ターゲット・オブジェクト・リポジトリの階層全体を展開または折りたたむことができます。

- ［**ナビゲート**］メニューの［**次の矛盾**］または［**前の矛盾**］を選択するか，ツールバーあるいは解決方法のオプション表示枠の［**次の矛盾**］または［**前の矛盾**］ボタンをクリックすると，ターゲット・オブジェクト・リポジトリ階層内の次の矛盾または前の矛盾に直接移動できます。
- ［検索］ダイアログ・ボックスを使用して，ターゲット・オブジェクト・リポジトリ内で 1 つ以上のオブジェクトを検索できます。詳細については，297 ページ「特定のオブジェクトの検索」を参照してください。
- ［**表示**］＞［**ターゲット リポジトリ オブジェクトのプロパティ**］を選択すると，ターゲット・オブジェクト・リポジトリにあるオブジェクトのプロパティの表示または非表示を切り替えることができます。

一次および二次リポジトリ表示枠

一次オブジェクト・リポジトリ表示枠および二次オブジェクト・リポジトリ表示枠には，結合を行う元のソース・オブジェクト・リポジトリにあるオブジェクトと，それらのプロパティおよび値が，階層表示されます。各オブジェクト階層の上にはファイル・パスが表示されます。

この表示枠には次の機能があります。

- 選択した項目をダブルクリックすると，その項目の階層を展開または折りたたむことができます。
- テスト・オブジェクトを該当する表示枠の中で選択すると，そのオブジェクトのプロパティと値が［**テスト オブジェクトの詳細**］領域に表示されます。
- ［**表示**］メニューの［**一次リポジトリ**］または［**二次リポジトリ**］を選択またはクリアすると，表示枠の表示または非表示を切り替えることができます。

解決方法のオプション表示枠

解決方法のオプション表示枠には，ターゲット・オブジェクト・リポジトリで選択されているオブジェクトについて，結合中に発生した矛盾に関する情報が表示されます。また，標準設定の解決方法のオプションを使用して適用された矛盾の解決方法について，それを維持するか変更するかを決めるオプションもあります。

解決方法のオプション表示枠には次の機能があります。

- ターゲット・オブジェクト・リポジトリの中で矛盾のあるオブジェクトを選択すると，矛盾を説明するテキストと，オブジェクト・リポジトリ結合ツールによって採用される解決方法が，表示枠に表示されます。採用されている解決方法の代わりとなるほかの解決方法の選択肢が用意されています。
- ラジオ・ボタンを選択することで，代わりとなる矛盾の解決方法を選択できます。変更を加えるたびに，ターゲット・オブジェクト・リポジトリが自動的に更新され，再表示されます。
- ［**前の矛盾**］ボタンまたは［**次の矛盾**］ボタンをクリックすることで，ターゲット・オブジェクト・リポジトリ階層内の次の矛盾または前の矛盾に直接移動できます。
- ローカル・オブジェクト・リポジトリの結合の場合，［**オブジェクトを無視**］ボタンをクリックすることで，特定のローカル・オブジェクト・リポジトリの

オブジェクトを結合プロセスから除外できます。結合が完了しても，除外したオブジェクトはローカル・オブジェクト・リポジトリに残っています。

➤ ［**表示**］メニューの［**解決方法のオプション**］を選択またはクリアすることで，表示枠の表示と非表示を切り替えることができます。

ステータス・バー

ステータス・バーには，結合プロセスと，表示されている結果に関する，次の情報が表示されます。

➤ ［ターゲット オブジェクト リポジトリ］表示枠の中で選択されているオブジェクトの矛盾の数（矛盾があれば）。

➤ プログレス・バーは結合プロセス中に表示されます。プロセスが完了すると，「**準備完了**」と表示されます。

➤ Quality Center アイコン。QuickTest が Quality Center プロジェクトに接続しているときに表示されます。

➤ フィルタのステータス。［フィルタ］アイコンの横に表示されます。［**オフ**］は，オブジェクト・リポジトリがフィルタ処理されておらず，すべてのオブジェクトが表示されていることを示します。［**オン**］は，フィルタが有効であり，オブジェクトの一部がフィルタによって表示から除外された可能性があることを示します。

➤ ［ターゲット オブジェクト リポジトリ］表示枠で使用されているアイコンの凡例。場合に応じて次のアイコンが表示されます。

➤ 類似記述の矛盾

➤ 同じ名前で記述が異なる矛盾

➤ 同じ記述で名前が異なる矛盾

矛盾のタイプの詳細については，291 ページ「オブジェクトの矛盾について」を参照してください。

ヒント：

- ステータス・バーの矛盾アイコンの上にポインタを置くと，矛盾のタイプを説明するツールチップが表示されます。
- 矛盾アイコンをのいずれかをクリックすると，［統計情報］ダイアログ・ボックスが表示されます。詳細については，290 ページ「結合の統計情報の表示」を参照してください。
-

ステータス・バーの［**フィルタ**］アイコンをクリックすると，［フィルタ］ダイアログ・ボックスが表示されます。フィルタが現在使用中のときは，ステータス・バーでフィルタが［**オン**］と表示されます。詳細については，296 ページ「［ターゲット リポジトリ］表示枠に対するフィルタの設定」を参照してください。

オブジェクト・リポジトリ結合ツールのコマンドの使用方法

オブジェクト・リポジトリ結合ツールのコマンドは，メニュー・バーまたはツールバーから選択できます。コマンドによってはショートカット・キーを押して実行できるものもあります。［ターゲット オブジェクト リポジトリ］表示枠でオブジェクトを選択し，ショート・カット（右クリック）メニューからコマンドを選択することもできます。

ツールバー・コマンドの使用方法

よく使用するコマンドは，オブジェクト・リポジトリ結合ツールのツールバーにあるボタンをクリックして実行できます。

	説明
	新規結合（271 ページ「[ファイル] メニューのコマンド」を参照）
	保存（271 ページ「[ファイル] メニューのコマンド」を参照）
	設定（276 ページ「[ツール] メニューのコマンド」を参照）
	統計情報（273 ページ「[表示] メニューのコマンド」を参照）
	フィルタ（276 ページ「[ツール] メニューのコマンド」を参照）
	前の矛盾（275 ページ「[ナビゲート] メニューのコマンド」を参照）
	次の矛盾（275 ページ「[ナビゲート] メニューのコマンド」を参照）
	検索（275 ページ「[ナビゲート] メニューのコマンド」を参照）
	前を検索（275 ページ「[ナビゲート] メニューのコマンド」を参照）
	次を検索（275 ページ「[ナビゲート] メニューのコマンド」を参照）
	Quality Center への接続（271 ページ「[ファイル] メニューのコマンド」を参照）

オブジェクト・リポジトリ結合ツールのコマンドの実行

よく使用するコマンドは，ツールバー・ボタンをクリックするか，該当するメニュー・オプションを選択することで実行できます。コマンドの一部は，該当するショートカット・キーを押して実行することもできます。

[ファイル] メニューのコマンド

[**ファイル**] メニューでは次のコマンドを使用して，結合されたオブジェクト・リポジトリを管理できます。

	コマンド	**ショートカット・キー**	**機能**
	[新規結合]	CTRL+N	新しい結合操作を実行する対象となる 2 つのオブジェクト・リポジトリを指定できます。
	[上書き保存]	CTRL+S	結合された共有オブジェクト・リポジトリを保存します。

	コマンド	ショートカット・キー	機能
	［名前を付けて保存］		［共有オブジェクト リポジトリの保存］ダイアログ・ボックスを開きます。結合した共有オブジェクト・リポジトリの名前，ファイルの種類，および格納場所を指定できます。
	［Quality Center への接続］		QuickTest を Quality Center プロジェクトに接続できます。詳細については，46 ページ「Quality Center プロジェクトへの接続」を参照してください。
	［終了］		［オブジェクト リポジトリ - 結合ツール］ウィンドウを閉じます（結合したオブジェクト・リポジトリをまだ保存していなければ，保存するよう求められます）。

［表示］メニューのコマンド

［**表示**］メニューでは次のコマンドを使用して，オブジェクト・リポジトリ結合ツールの画面の表示形態を管理できます。

	コマンド	機能
	［一次リポジトリ］	［一次リポジトリ・ファイル］表示枠を表示します。この表示枠には，結合した 1 番目のソース・オブジェクト・リポジトリのオブジェクトが階層表示されます。また，この表示枠の中で選択した各オブジェクトの詳細も表示されます。詳細については，268 ページ「一次および二次リポジトリ表示枠」および 281 ページ「2 つのオブジェクト・リポジトリの結合」を参照してください。
	［二次リポジトリ］	［二次リポジトリ・ファイル］表示枠を表示します。この表示枠には，結合した 2 番目のソース・オブジェクト・リポジトリのオブジェクトが階層表示されます。また，この表示枠の中で選択した各オブジェクトの詳細も表示されます。詳細については，268 ページ「一次および二次リポジトリ表示枠」および 281 ページ「2 つのオブジェクト・リポジトリの結合」を参照してください。
	［ターゲット リポジトリ オブジェクトの プロパティ］	［オブジェクトのプロパティ - ターゲット ファイル］表示枠を表示します。この表示枠には，［ターゲット リポジトリ］表示枠で選択した各テスト・オブジェクトの詳細が表示されます。詳細については，266 ページ「［ターゲット リポジトリ］表示枠」を参照してください。
	［解決方法の オプション］	［解決オプション］表示枠を表示します。この表示枠には，結合中に発生した矛盾に関する情報が示されます。詳細については，268 ページ「解決方法のオプション表示枠」および 294 ページ「オブジェクトの矛盾の解決」を参照してください。
	［標準のレイアウトを 復元］	［**標準のレイアウトに設定**］オプション（後述を参照）を使用して保存したビューを復元します。表示枠のサイズを変更した後や特定の表示枠の表示または非表示を切り替えた後に，保存したビューを復元する場合に便利です。詳細については，266 ページ「ビューの変更」を参照してください。

	コマンド	機能
	［標準のレイアウトに設定］	現在のビューを保存できます。以降，［オブジェクト リポジトリ - 結合ツール］を開くたびにこのビューが表示されます。後で，表示枠のサイズを変更したり，表示枠の表示または非表示を切り替えたりして，このビューを変更した場合，［**標準のレイアウトを復元**］オプション（前述を参照）を使用して標準のビューを復元できます。詳細については，266 ページ「ビューの変更」を参照してください。
	［統計情報］	［統計情報］ダイアログ・ボックスを開きます。このダイアログ・ボックスには，ファイルがどのように結合されたのかが示され，結合中に解決された矛盾の数と種類が表示されます。詳細については，290 ページ「結合の統計情報の表示」を参照してください。
	［すべて折りたたみ］	［ターゲット オブジェクト リポジトリ］表示枠内の階層全体を折りたたみます。 **ヒント**：単一のノードを折りたたむにはノードをダブルクリックします。
	［すべて展開］	［ターゲット オブジェクト リポジトリ］表示枠内の階層全体を展開します。 **ヒント**：単一のノードを展開するにはノードをダブルクリックします。

［ナビゲート］メニューのコマンド

［**ナビゲート**］メニューでは次のコマンドを実行できます。

	コマンド	ショートカット・キー	機能
	［次の矛盾］	F4	結合後のオブジェクト・リポジトリの中で次の矛盾オブジェクトを検索します。
	［前の矛盾］	SHIFT+F4	結合後のオブジェクト・リポジトリの中で前の矛盾オブジェクトを検索します。
	［検索］	CTRL+F	［検索］ダイアログ・ボックスを開きます。
	［次を検索］	F3	［検索］ダイアログ・ボックスの検索条件に従って，結合後のオブジェクト・リポジトリの中で次のオブジェクトを検索します。
	［前を検索］	SHIFT+F3	［検索］ダイアログ・ボックスの検索条件に従って，結合後のオブジェクト・リポジトリの中で前のオブジェクトを検索します。

［ツール］メニューのコマンド

［**ツール**］メニューでは次のコマンドを実行できます。

	コマンド	機能
	［設定］	［設定］ダイアログ・ボックスを開きます。次を実行できます。 ➤ 結合中のオブジェクトの矛盾をオブジェクト・リポジトリ結合ツールでどのように処理するかを設定できます。 ➤ ターゲット・オブジェクト・リポジトリに表示されるオブジェクト名のテキストの色を指定します。 詳細については，277 ページ「標準設定の定義」を参照してください。
	［フィルタ］	［フィルタ］ダイアログ・ボックスを開きます。このダイアログ・ボックスを使用して，［ターゲット リポジトリ］表示枠にすべてのテスト・オブジェクトを表示するか，結合中に矛盾が解決されたオブジェクトのみを表示できます。詳細については，296 ページ「［ターゲット リポジトリ］表示枠に対するフィルタの設定」を参照してください。

［ヘルプ］メニューのコマンド

［**ヘルプ**］メニューでは次のコマンドを実行できます。

コマンド	ショートカット・キー	機能
［オブジェクト リポジトリ結合ツールのヘルプ］	F1	オブジェクト・リポジトリ結合ツールのヘルプを開きます。

標準設定の定義

オブジェクト・リポジトリ結合ツールでは，あらかじめ定義された設定が用意されています。これらの設定は，オブジェクト・リポジトリを結合するとき，またはローカル・オブジェクト・リポジトリから共有オブジェクト・リポジトリを更新するときに使用されます。これらの設定は次のとおりです。

- オブジェクト・リポジトリ結合ツールが一次および二次オブジェクト・リポジトリ内のオブジェクトの矛盾をどのように処理するのかを設定します。あるいは，ローカル・オブジェクト・リポジトリからの共有オブジェクト・リポジトリを更新するときに，ローカルおよび共有オブジェクト・リポジトリ内のオブジェクトの矛盾をどのように処理するのかを設定します。
- ターゲット・オブジェクト・リポジトリに表示されるオブジェクト名のテキストの色を指定します。

これらの設定をいつでも変更して，新しい標準設定を作成できます。設定を変更すると，以降のすべての新しい結合が新しい標準設定に従って実行されます。

ヒント：2 つのオブジェクト・リポジトリを結合する前に設定を変更するには，［**キャンセル**］をクリックして［新規結合］ダイアログ・ボックスを閉じ，以降の各項の説明に従って設定を変更した後，結合を実行する必要があります。

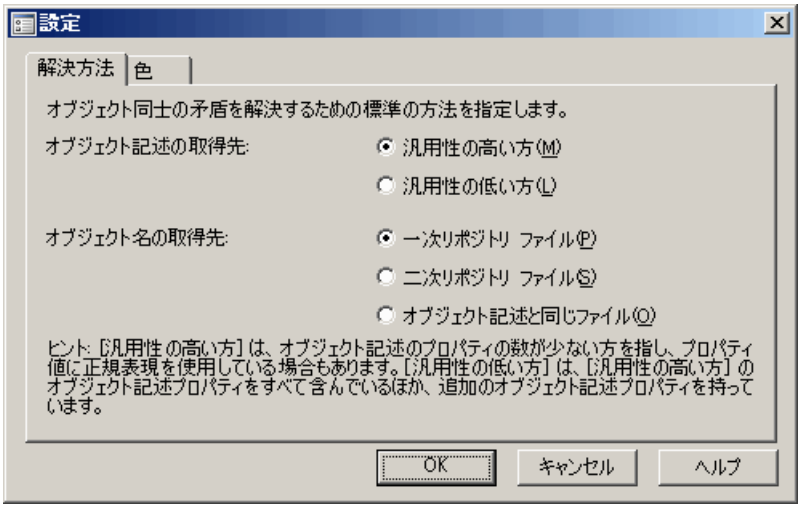
標準の解決方法の設定

結合プロセス中，または［**ローカル リポジトリから更新**］操作を実行するときに，オブジェクト・リポジトリ結合ツールでオブジェクトの矛盾をどのように自動的に処理するかを設定できます。

標準の解決方法を設定するには，次の手順を実行します。

1 ［**ツール**］＞［**設定**］を選択するか，［**設定**］ボタンをクリックします。［設定］ダイアログ・ボックスが開きます。

2［**解決方法**］タブをクリックします。



3 矛盾のあるオブジェクトを処理するときにオブジェクト・リポジトリ結合ツールに適用させる標準の解決方法を指定する適切なラジオ・ボタンを選択します。

- ［**オブジェクト記述の取得先 :**］: 名前が同じで記述が異なる 2 つのオブジェクトの矛盾を解決する方法を指定します。ターゲット・オブジェクト・リポジトリにおいて，汎用性の高い方のオブジェクト記述を採用するか，汎用性の低い方のオブジェクト記述を採用するかを指定できます。
  - ［**汎用性の高い方**］: 矛盾相手のオブジェクトよりも識別プロパティが少ないか，プロパティ値の中で正規表現を使用しているオブジェクトを採用するよう，オブジェクト・リポジトリ結合ツールを設定します。これが標準設定です
  - ［**汎用性の低い方**］: 矛盾相手のオブジェクトのすべての識別プロパティに加えてほかの識別プロパティも持っているオブジェクトを採用するよう，オブジェクト・リポジトリ結合ツールを設定します。
- ［**オブジェクト名の取得先 :**］: 2 つのオブジェクトの記述が同一または類似しながらも名前が異なる場合の矛盾を解決する方法を指定します。ターゲット・オブジェクト・リポジトリにおいて採用するオブジェクト名の取得先となるソースを選択できます。

- ［**一次リポジトリ ファイル**］: 一次オブジェクト・リポジトリ内のオブジェクトのオブジェクト名を，ターゲット・オブジェクト・リポジトリで採用します。これが標準設定です （ローカル・オブジェクト・リポジトリから共有オブジェクト・リポジトリを更新する場合は，このオプションは「**ローカル・オブジェクト・リポジトリ**」に適用されます)。
- ［**二次リポジトリ ファイル**］: 二次オブジェクト・リポジトリ内のオブジェクトのオブジェクト名を，ターゲット・オブジェクト・リポジトリで採用します（ローカル・オブジェクト・リポジトリから共有オブジェクト・リポジトリを更新する場合は，このオプションは「**共有オブジェクト・リポジトリ**」に適用されます)。
- ［**オブジェクト記述と同じファイル**］: オブジェクト記述を採用した同じオブジェクト・リポジトリ内のオブジェクトからのオブジェクト名を，ターゲット・オブジェクト・リポジトリで採用します。

注 : ローカル・オブジェクト・リポジトリから共有オブジェクト・リポジトリを更新する場合は，これらのオブジェクト・リポジトリは ローカル・オブジェクト・リポジトリおよび 共有オブジェクト・リポジトリとして参照されます。

---

4 ［**OK**］をクリックします。以降，オブジェクト・リポジトリ結合ツールで実行するリポジトリの結合においてオブジェクト間の矛盾を解決する際に，ここで選択した方法が適用されます。

注 : 結合後のオブジェクト・リポジトリを開いたままの状態で解決方法の設定に変更を加えた場合は，開いているファイルについて新しい設定でもう一度結合するかどうか尋ねられます。新しい設定でもう一度ファイルを結合する場合は，［**はい**］をクリックします。以前の設定で作成した既存の結合を維持する場合は，［**いいえ**］をクリックします。［**いいえ**］をクリックした場合，新しい設定は以降の結合にのみ適用されます。

色の設定

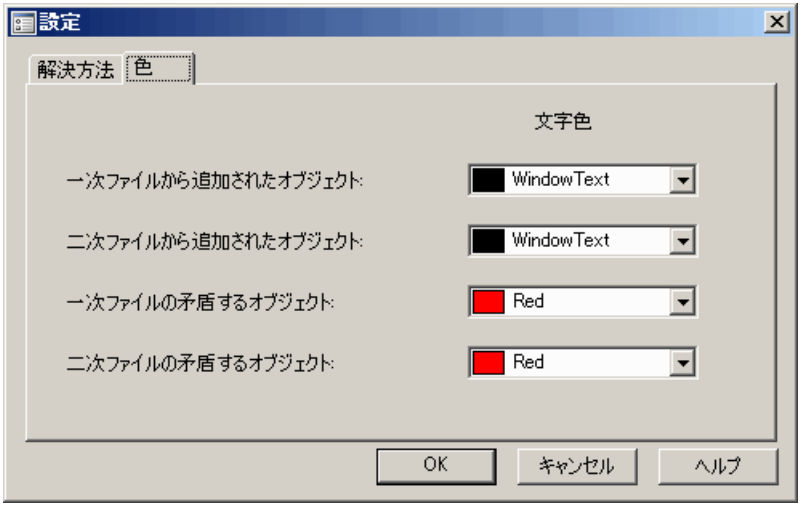
ターゲット・オブジェクト・リポジトリに表示されるオブジェクト名の色を，オブジェクトのソースと，矛盾の原因となったかどうかに基づいて指定できま

す。これを利用することで，各オブジェクトのステータスを容易に判断できるようになります。

注：［設定］ダイアログ・ボックスの［色］タブにあるオプションは，［**ローカル リポジトリから更新**］操作を実行するときに，ローカル（一次）オブジェクト・リポジトリおよび共有（二次）オブジェクト・リポジトリから追加されたオブジェクトにも適用されます。

色の設定を指定するには，次の手順を実行します。

1 ［**ツール**］＞［**設定**］を選択するか，［**設定**］ボタンをクリックします。［設定］ダイアログ・ボックスが開きます。



2 ［色］タブの項目ごとに，テキスト・ボックスの横にある下矢印をクリックし，識別用の色を［Custom］，［Web］，または［System］タブから選択します。

3 ［**OK**］をクリックします。ターゲット・オブジェクト・リポジトリ内のオブジェクト名が選択した色で表示されます。

2 つのオブジェクト・リポジトリの結合

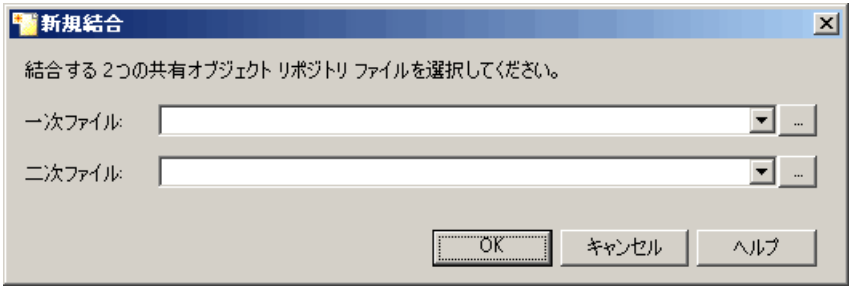
オブジェクト・リポジトリ結合ツールを使用すると，2 つのソース・オブジェクト・リポジトリを結合して新しい共有オブジェクト・リポジトリを作成できます。オブジェクト・リポジトリにあるオブジェクトは自動的に比較され，オブジェクト間の矛盾の解決方法を定義する設定可能なルールに従って，新しいオブジェクト・リポジトリに追加されます。元のソース・ファイルは変更されません。

注：別のユーザが開いているオブジェクト・リポジトリはロックされます。ロックされたファイルを統合しようとした場合，警告メッセージが表示されますが，結合プロセスはソース・ファイルを変更しないため，結合を実行することは可能です。ただし，ロックされたファイルに対してほかのユーザによって加えられた変更が，結合後のオブジェクト・リポジトリに含まれない可能性があります。

---

2 つのオブジェクト・リポジトリを結合するには，次の手順を実行します。

1 オブジェクト・リポジトリ・マネージャで，［**ツール**］＞［**オブジェクト リポジトリ結合ツール**］を選択します。［オブジェクト リポジトリ - 結合ツール］ウィンドウの手前に，［新規結合］ダイアログ・ボックスが開きます。



ヒント：

- ［オブジェクト リポジトリー結合ツール］ウィンドウがすでに開いている場合は，［**ファイル**］＞［**新規結合**］を選択するか，［**新規結合**］ボタンをクリックして［新規結合］ダイアログ・ボックスを開きます。
- オブジェクト・リポジトリを結合する前に設定済みの設定を変更するには，［**キャンセル**］をクリックして［新規結合］ダイアログ・ボックスを閉じ，277 ページ「標準設定の定義」の説明に従って設定を変更した後，結合を実行する必要があります。

2 ［**一次ファイル**］および［**二次ファイル**］ボックスで，結合後の単独のオブジェクト・リポジトリとなる **.tsr** オブジェクト・リポジトリの名前を入力するか，または参照します。各ボックスの横にある下矢印 をクリックすると，最近使用したファイルを表示して選択できます。

注：

- 一次オブジェクト・リポジトリには，最も作業内容の多いオブジェクト・リポジトリ，つまり，より多くのオブジェクト，オブジェクト・プロパティ，およびオブジェクト値を持つオブジェクト・リポジトリを選択することをお勧めします。
- 拡張子が **.tsr** でないファイル，パスが正しくない **.tsr** ファイル，または存在しないファイルを入力した場合は，対応するテキスト・ボックスの横に警告アイコンが表示されます。アイコンの上にポインタを置くと，エラーを説明するツールチップが表示されます。正しいパスを持つ既存の **.tsr** ファイルを入力または選択してください。

- バージョン 9.0 よりも前の QuickTest を使用して作成されたオブジェクト・リポジトリを結合する場合は，はじめにオブジェクト・リポジトリ・マネージャでそのオブジェクト・リポジトリを開いてから保存して新しい形式に更新する必要があります。
- Quality Center に接続している場合は，ファイル・システムと同様に Quality Center からオブジェクト・リポジトリに入ることができます（参照できます）。

3 ［**OK**］をクリックします。オブジェクト・リポジトリ結合ツールによって，設定されている解決方法の設定に従い，選択したオブジェクト・リポジトリが新しいターゲット・オブジェクト・リポジトリに自動的に結合されます。結合の結果は，［オブジェクト リポジトリ - 結合ツール］ウィンドウの手前に表示される［統計情報］ダイアログ・ボックスに表示されます。

4 290 ページ「結合の統計情報の表示」の説明を参考にして，結合に関する統計情報を確認し，［**閉じる**］をクリックします。

［オブジェクト リポジトリ - 結合ツール］ウィンドウでは，次を実行できます。

- ソース・オブジェクト・リポジトリのオブジェクト間の矛盾を解決する方法を必要に応じて変更できます。詳細については，294 ページ「オブジェクトの矛盾の解決」を参照してください。
- ターゲット・オブジェクト・リポジトリ内のオブジェクトにフィルタを適用できます。詳細については，296 ページ「［ターゲット リポジトリ］表示枠に対するフィルタの設定」を参照してください。
- ターゲット・オブジェクト・リポジトリをファイル・システムまたは Quality Center プロジェクトに保存できます。詳細については，299 ページ「ターゲット・オブジェクト・リポジトリの保存」を参照してください。

ローカル・オブジェクト・リポジトリから共有オブジェクト・リポジトリを更新

1 つ以上のコンポーネント（アプリケーション領域）に関連付けられているローカル・オブジェクト・リポジトリを，共有オブジェクト・リポジトリに結合することによって，共有オブジェクト・リポジトリを更新することができます。更新後，ローカル・オブジェクト・リポジトリから結合されたオブジェクトは，当該共有オブジェクト・リポジトリを使用する任意のコンポーネントから利用できるようになります。

結合処理では，選択したコンポーネントに対応するローカル・オブジェクト・リポジトリ内のオブジェクトが，ターゲットの共有オブジェクト・リポジトリに移動します。そして，当該コンポーネントで，更新後の共有オブジェクト・リポジトリのオブジェクトが使用されるようになります。

矛盾しているオブジェクトを更新プロセス中に処理する方法は，［設定］ダイアログ・ボックスで表示または変更できます。詳細については，277 ページ「標準設定の定義」を参照してください。

複数のコンポーネント用のローカル・オブジェクト・リポジトリを追加することを選択した場合は，QuickTest によって複数の結合が実行され，各コンポーネントのローカル・オブジェクト・リポジトリがリスト内のすべての コンポーネント に対して一度に 1 つずつターゲット・オブジェクト・リポジトリに結合されます。必要ならば，各結合の結果を表示および変更できます。

注：

- ➤ オブジェクト・リポジトリ結合ツールでは，ローカル・オブジェクト・リポジトリのチェックポイントおよび出力オブジェクトがターゲットの共有オブジェクト・リポジトリに結合されることはありません。チェックポイントおよび出力オブジェクトは，ローカル・オブジェクト・リポジトリから共有オブジェクト・リポジトリにエクスポートできます。その後，結合プロセスが完了した後に，オブジェクト・リポジトリ・マネージャを使用して，エクスポートしたオブジェクト・リポジトリからターゲット・オブジェクト・リポジトリにチェックポイントや出力オブジェクトを手動で移動できます。
- ➤ 結合できるのは，アプリケーション領域が更新対象の共有オブジェクト・リポジトリに関連付けられているコンポーネントのローカル・オブジェクト・リポジトリのみです。

ローカル・オブジェクト・リポジトリから共有オブジェクト・リポジトリを更新するには，次の手順を実行します。

1 ［**リソース**］＞［**オブジェクト リポジトリ マネージャ**］を選択します。オブジェクト・リポジトリ・マネージャが開きます。

注： オブジェクト・リポジトリ・マネージャの詳細については，第 7 章「オブジェクト・リポジトリの管理」を参照してください。

---

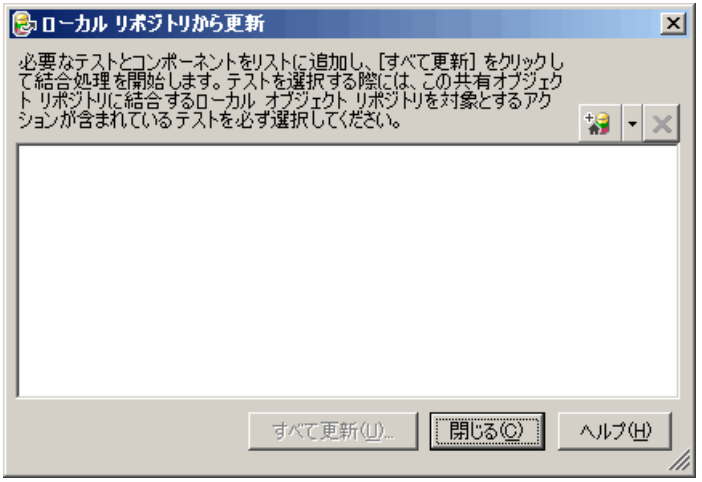
2 ［オブジェクト リポジトリ マネージャ］で，［**ファイル**］＞［**開く**］を選択するか，［**開く**］ボタンをクリックします。［共有オブジェクト リポジトリを開く］ダイアログ・ボックスが開きます。

3 サイドバーで，オブジェクト・リポジトリ・ファイルの場所を選択します（File System や Quality Center Resources など）。更新したい共有オブジェクト・リポジトリが含まれている **.tsr** ファイルを参照して選択し，［**読み取り専用モードで開く**］チェック・ボックスをクリアし，［**開く**］をクリックします。ファイルが開き，オブジェクトとプロパティが編集可能な形式で表示されます。

ヒント：オブジェクト・リポジトリを読み取り専用モードで開いた場合は，［**ファイル**］＞［**編集を有効化**］を選択するか，オブジェクト・リポジトリ・マネージャのツールバーの［**編集を有効化**］ボタンをクリックします。オブジェクト・リポジトリ・ファイルが編集可能になります。

---

4 ［**ツール**］＞［**ローカル リポジトリから更新**］を選択します。［ローカル リポジトリから更新］ダイアログ・ボックスが開きます。



5 Quality Center プロジェクトに接続していることを確認します。［**テストを追加**］ボタンの横にある下矢印 をクリックして，［**テストを参照**］を選択します。

［Quality Center プロジェクトから QuickTest コンポーネントを開く］ダイアログ・ボックスが開きます。

コンポーネントを参照します。このコンポーネントのローカル・オブジェクト・リポジトリが共有オブジェクト・リポジトリに結合されます。

注：追加可能なコンポーネントは，そのアプリケーション領域が更新対象の共有オブジェクト・リポジトリに関連付けられていて，そのローカル・オブジェクト・リポジトリにオブジェクトが含まれているコンポーネントのみです。

---

6 必要であれば，手順 5 を繰り返してコンポーネントをさらに追加します。

注：すべてのコンポーネントに関連付けられているローカル・オブジェクト・リポジトリが結合に含まれます。特定のコンポーネントを結合から除外する場合は，一覧でそれを選択して［**削除**］をクリックします。

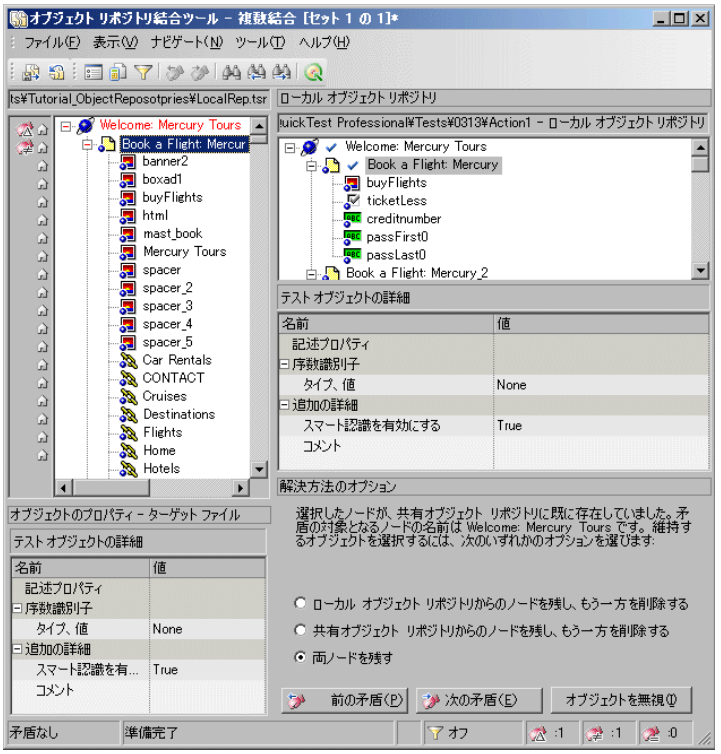
7［**すべて更新**］をクリックします。QuickTest によって，設定済みの設定に従い，最初のコンポーネントのローカル・オブジェクト・リポジトリが共有オブジェクト・リポジトリに自動的に結合されます。結合の結果は，［オブジェクト リポジトリ - 結合ツール］ウィンドウの手前に表示される［統計情報］ダイアログ・ボックスに表示されます。

注：各結合の前に，別のユーザがローカル・オブジェクト・リポジトリを使用していないかどうかが QuickTest によって確認されます。別のユーザが使用している場合，そのローカル・オブジェクト・リポジトリはロックされているので，選択したコンポーネントに対応したオブジェクトをターゲットの共有オブジェクト・リポジトリに移動することはできません。警告メッセージが表示されます。ほかのユーザがローカル・オブジェクト・リポジトリの使用を止めれば，結合を実行できます。

---

8 290 ページ「結合の統計情報の表示」の説明を参考にして，結合に関する統計情報を確認し，［**閉じる**］をクリックします。

ローカル・オブジェクト・リポジトリの結合の場合，［オブジェクト リポジトリ - 結合ツール］ウィンドウには，ローカル・オブジェクト・リポジトリが一次オブジェクト・リポジトリとして表示され，共有オブジェクト・リポジトリがターゲット・オブジェクト・リポジトリとして表示されます。



ターゲット・オブジェクト階層内の各オブジェクトの左側には，オブジェクトのソースを示すアイコンが表示されます。

は，オブジェクトがローカル・オブジェクト・リポジトリから追加されたことを示します。

は，オブジェクトがすでに共有オブジェクト・リポジトリに存在することを示します。

注 : ［ローカル リポジトリから更新］ダイアログ・ボックスで複数のコンポーネントを指定した場合は，QuickTest によって複数の結合が実行され，各コンポーネントのローカル・オブジェクト・リポジトリが一度に 1 つずつターゲット・オブジェクト・リポジトリに結合されます。この手順の後に表示される［統計情報］ダイアログ・ボックスおよび［オブジェクト リポジトリ結合ツール - 複数結合］ウィンドウには，最初の結合の結果が示されます（最初のコンポーネントのローカル・オブジェクト・リポジトリが共有オブジェクト・リポジトリに結合されます）。QuickTest では，必要に応じて各結合の結果を順に表示および変更できます。複数結合では，各結合セットの番号がタイトル・バーに表示されます。たとえば，「**[セット 3 の 2]**」などと表示されます。

---

9 共有オブジェクト・リポジトリに結合された各オブジェクトに対しては，自動結合を選択するか，解決方法のオプション表示枠を使用して次のことを行うことができます。

- ➤ 共有オブジェクト・リポジトリの特定のオブジェクトを維持し，ローカル・オブジェクト・リポジトリから矛盾しているオブジェクトを削除します。
- ➤ ローカル・オブジェクト・リポジトリの特定のオブジェクトを維持し，共有オブジェクト・リポジトリから矛盾しているオブジェクトを削除します。
- ➤ 共有オブジェクト・リポジトリとローカル・オブジェクト・リポジトリの両方の矛盾しているオブジェクトを維持します。
- ➤ 共有オブジェクト・リポジトリに含まれないように，ローカル・オブジェクト・リポジトリの特定のオブジェクトを結合プロセスから除外します。共有オブジェクト・リポジトリ表示枠でオブジェクトを選択し，解決方法のオプション表示枠の下部にある［**オブジェクトを無視**］をクリックします。オブジェクトが共有オブジェクト・リポジトリから除外され，ローカル・オブジェクト・リポジトリ・ツリーで淡色表示されます。このオブジェクトは，結合が完了すると，コンポーネントのローカル・オブジェクト・リポジトリに残ります。

注：

- ［**オブジェクトを無視**］ボタンは，ローカル・オブジェクト・リポジトリの結合用の［Merge Tool］ウィンドウにのみ表示され，ローカル・オブジェクト・リポジトリ内のオブジェクトを選択したときのみ有効になります。
- ［**オブジェクトを無視**］操作を元に戻すことはできません。オブジェクトを再び結合プロセスに含めるには，ツールバーで［**元の結合ファイルに戻す**］をクリックして結合を繰り返す必要があります。

詳細については，294 ページ「オブジェクトの矛盾の解決」を参照してください。

10 複数の結合を実行している場合は，［オブジェクト リポジトリ - 結合ツール］のツールバーの［**保存して次を結合**］ボタンをクリックして，次の結合を実行します（次のコンポーネントのローカル・オブジェクト・リポジトリが共有オブジェクト・リポジトリに結合されます)。

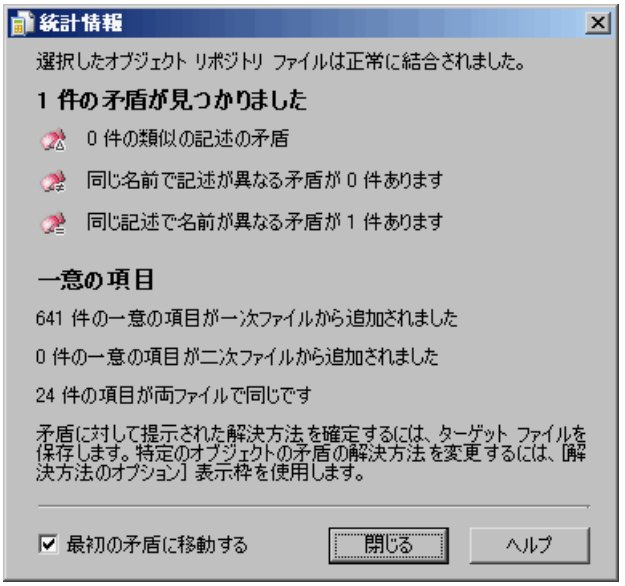
11 結合ごとに［**はい**］をクリックして変更を保存します。［**いいえ**］をクリックした場合は，現在の結合（最後のコンポーネントから結合されたオブジェクト）は保存されません。

12 手順 8 から 11 を繰り返して，複数の結合を完了させます。

13 ［**ファイル**］＞［**終了**］をクリックし，［**はい**］をクリックすると，更新されたオブジェクト・リポジトリが保存されます。

結合の統計情報の表示

2 つのオブジェクト・リポジトリを結合した後，オブジェクト・リポジトリ結合ツールには［統計情報］ダイアログ・ボックスが表示されます。このダイアログ・ボックスには，ファイルが結合された方法と，結合中に解決されたすべての矛盾の数とタイプが表示されます。



注：［**ローカル リポジトリから更新**］による結合の実行後に表示される［統計情報］ダイアログ・ボックスは，前に示したダイアログ・ボックスとは多少異なります。

ヒント：［統計情報］ダイアログ・ボックスの結合に関する統計情報は，［オブジェクト リポジトリ - 結合ツール］ウィンドウで［**表示**］＞［**統計情報**］ボタンを選択するか，ツールバーの［**統計情報**］ボタンをクリックするか，ステータス・バーの矛盾アイコンをクリックすることでいつでも表示できます。

［統計情報］ダイアログ・ボックスには次の情報が表示されます。

- ターゲット・オブジェクト・リポジトリに追加されたオブジェクト間で発生した，すべての矛盾の数とタイプ。矛盾の種類については，294 ページ「オブジェクトの矛盾の解決」を参照してください。
- ターゲット・オブジェクト・リポジトリに追加された，一次ファイルまたは二次ファイル（またはローカル・ファイル）の中の一意の項目または両方のファイルでまったく同じ項目の数。

ヒント：［**最初の矛盾に移動する**］チェック・ボックスを選択すると，［統計情報］ダイアログ・ボックスを閉じた直後に，ターゲット・オブジェクト・リポジトリ内の最初の矛盾に移動します。

オブジェクトの矛盾について

2 つのオブジェクト・リポジトリを結合する際，それらに含まれているオブジェクトどうしが類似しているために，矛盾が発生することがあります。オブジェクト・リポジトリ統合ツールでは，発生する可能性のある 3 つの矛盾タイプが識別されます。

- **類似記述の矛盾**：名前とオブジェクト階層が同じであるが，記述が若干異なる 2 つのオブジェクト。この矛盾タイプでは，一方のオブジェクトが常に他方のプロパティ・セットのサブセットを持っています。これらの矛盾の詳細については，292 ページを参照してください。

  標準設定では，このタイプの矛盾に対する矛盾の解決方法の設定は，矛盾相手のオブジェクトよりも識別プロパティの少ない方のオブジェクトがターゲット・オブジェクト・リポジトリで採用されるように設定されます。標準設定の変更方法の詳細については，277 ページ「標準設定の定義」を参照してください。

- **同じ名前で記述が異なる矛盾**：名前とオブジェクト階層が同じであるが，それらの記述が若干異なる（たとえば，プロパティが異なる，プロパティが同じでも値が異なるなど）2 つのオブジェクト。これらの矛盾の詳細については，293 ページを参照してください。

標準設定では，このタイプの矛盾に対する矛盾の解決方法の設定は，両方のファイルのオブジェクトがターゲット・オブジェクト・リポジトリで採用されるように設定されます。二次ファイルから追加されるオブジェクトの名前は変更され，1 つずつ値が大きくなる数字の接尾辞が名前に付けられます。たとえば，Edit_1 などという名前になります。標準設定の変更方法の詳細については，277 ページ「標準設定の定義」を参照してください。

➤ **同じ記述で名前が異なる矛盾**：記述がまったく同じで，オブジェクト階層も同じであるが，オブジェクト名が異なる 2 つのオブジェクト。これらの矛盾については，293 ページを参照してください。

標準設定では，このタイプの矛盾に対する矛盾の解決方法の設定は，一次ソース・ファイルのオブジェクト名がターゲット・オブジェクト・リポジトリで採用されるように設定されます。標準設定の変更方法の詳細については，277 ページ「標準設定の定義」を参照してください。

注：記述を持たない Page オブジェクトや Browser オブジェクトなどのオブジェクトは名前のみ比較されます。両方のソース・オブジェクト・リポジトリに同じオブジェクトが含まれていて，それらの名前が異なる場合は，2 つの別々のオブジェクトとしてターゲット・オブジェクト・リポジトリに結合されます。

類似記述の矛盾

一次オブジェクト・リポジトリ内のオブジェクトと二次オブジェクト・リポジトリ内のオブジェクトが，同じ名前を持ち，まったく同じではないけれども類似する記述プロパティおよび値を持っている場合です。一方のオブジェクトが常に他方のプロパティ・セットのサブセットを持っています。たとえば，二次オブジェクト・リポジトリにある Button_1 という名前のオブジェクトが，一次オブジェクト・リポジトリにある Button_1 という名前のオブジェクトと同じ記述プロパティおよび値を持っているものの，さらに追加のプロパティと値を持っているとします。

この矛盾タイプは次のようにして解決できます。

➤ 一次オブジェクト・リポジトリから追加されるオブジェクトのみを維持する。

➤ 二次オブジェクト・リポジトリから追加されるオブジェクトのみを維持する。

- 両方のオブジェクト・リポジトリからのオブジェクトを維持する。この場合，ターゲット・オブジェクト結合ツールでは，二次ファイルから追加されるオブジェクトの名前が自動的に変更され，1 つずつ値が大きくなる数字の接尾辞が名前に付けられます。たとえば，Edit_1 などという名前になります。
- ローカル・オブジェクト・リポジトリからのオブジェクトを無視し，共有オブジェクト・リポジトリからのオブジェクトを維持します（ローカル・オブジェクト・リポジトリから共有オブジェクト・リポジトリを更新する場合）。

同じ名前で記述が異なる矛盾

一次オブジェクト・リポジトリ内のオブジェクトと二次オブジェクト・リポジトリ内のオブジェクトが，同じ名前を持つものの，完全に異なる記述プロパティおよび値を持っている場合です。

この矛盾タイプは次のようにして解決できます。

- 一次オブジェクト・リポジトリから追加されるオブジェクトのみを維持する。
- 二次オブジェクト・リポジトリから追加されるオブジェクトのみを維持する。
- 両方のオブジェクト・リポジトリからのオブジェクトを維持する。この場合，ターゲット・オブジェクト結合ツールでは，二次ファイルから追加されるオブジェクトの名前が自動的に変更され，1 つずつ値が大きくなる数字の接尾辞が名前に付けられます。たとえば，Edit_1 などという名前になります。
- ローカル・オブジェクト・リポジトリからのオブジェクトを無視し，共有オブジェクト・リポジトリからのオブジェクトを維持します（ローカル・オブジェクト・リポジトリから共有オブジェクト・リポジトリを更新する場合）。

同じ記述で名前が異なる矛盾

一次オブジェクト・リポジトリ内のオブジェクトと二次オブジェクト・リポジトリ内のオブジェクトが，異なる名前を持つものの，同じ記述プロパティおよび値を持っている場合です。

この矛盾タイプは次のようにして解決できます。

- 一次オブジェクト・リポジトリにある該当オブジェクトのオブジェクト名を採用する。
- 二次オブジェクト・リポジトリにある該当オブジェクトのオブジェクト名を採用する。

➤ ローカル・オブジェクト・リポジトリからのオブジェクトを無視し，共有オブジェクト・リポジトリからのオブジェクトを維持します（ローカル・オブジェクト・リポジトリから共有オブジェクト・リポジトリを更新する場合）。

オブジェクトの矛盾の解決

一次オブジェクト・リポジトリと二次オブジェクト・リポジトリにあるオブジェクトどうしの矛盾は，オブジェクト・リポジトリ結合ツールによって，標準として設定されている解決方法に従って自動的に解決されます。標準の解決方法は，結合を実行する前に設定できます。詳細については，277 ページ「標準設定の定義」を参照してください。

ただし，オブジェクト・リポジトリ結合ツールでは，矛盾の原因となった個々のオブジェクトごとに，結合の実行方法を変更することも可能です。

たとえば，一次オブジェクト・リポジトリ内のオブジェクトが，二次オブジェクト・リポジトリ内のオブジェクトと同じ名前だったものの，記述が異なったとします。このとき，標準の設定として，汎用性の高い方のオブジェクト記述を持つオブジェクト，つまり，プロパティの数の少ない方のオブジェクトをターゲット・オブジェクト・リポジトリに追加すると定義していたとしましょう。しかし，自動結合の後に矛盾を確認した結果，その特定の矛盾を別の方法（たとえば，両方のオブジェクトを維持するなど）で扱うように判断することが考えられます。

注：標準の矛盾の解決方法に変更を加えること自体が新しい矛盾の原因となり，ターゲット・オブジェクト・リポジトリに影響を与えることがあります。前述の例では，両方のオブジェクトを維持することが名前の矛盾の原因となります。したがって，矛盾の解決方法を変更するたびにターゲット・オブジェクト・リポジトリが更新され，再表示されます。

オブジェクト・リポジトリ結合ツールの［ターゲット オブジェクト リポジトリ］表示枠で，オブジェクト名の左側に表示されるアイコンと，テキストの色によって，矛盾の原因となったオブジェクトと，矛盾のタイプを，識別することができます。矛盾のあるオブジェクトを選択すると，矛盾の詳細な説明が，オブジェクト・リポジトリ結合ツールによる矛盾の自動解決方法とともに，解決方法のオプション表示枠に表示されます。

解決方法のオプション表示枠には，代わりの解決方法のオプションが提示されます。標準の解決方法がニーズに合っていればそれを維持することも，代わりのオプションを使用して矛盾を別の方法で解決することも選択できます。また，ローカル・オブジェクト・リポジトリの結合の場合，［**オブジェクトを無視**］ボタンをクリックすることで，特定のローカル・オブジェクト・リポジトリのオブジェクトを共有オブジェクト・リポジトリから除外できます。

ヒント：また，標準の解決方法の設定を変更して，ファイルを再び結合することもできます。詳細については，277 ページ「標準設定の定義」を参照してください。

オブジェクトの矛盾の解決方法を変更するには，次の手順を実行します。

1 ターゲット・オブジェクト・リポジトリで，矛盾のあるオブジェクトを選択します。矛盾のあるオブジェクトはオブジェクト名の左側のアイコンで示されています。矛盾のあるオブジェクトがソース・オブジェクト・リポジトリで強調表示されます。

矛盾の説明と，オブジェクト・リポジトリ結合ツールが使用する解決方法が，解決方法のオプション表示枠に表示されます。使用できる代替の解決方法ごとに，対応するラジオ・ボタンが表示されます。各矛盾タイプの詳細については，291 ページ「オブジェクトの矛盾について」を参照してください。

2 解決方法のオプション表示枠で，ラジオ・ボタンを選択し，代わりとなる矛盾の解決方法を選択します。選択した解決方法に従ってターゲット・オブジェクト・リポジトリが更新され，再表示されます。

3 解決方法のオプション表示枠で，［**次の矛盾**］または［**前の矛盾**］ボタンをクリックすると，ターゲット・リポジトリ階層内の次または前の矛盾に直接移動できます。

4 必要に応じて，手順 1 から 3 を繰り返して，矛盾の解決方法を変更します。

5 ターゲット・オブジェクト・リポジトリを保存します。詳細については，299 ページ「ターゲット・オブジェクト・リポジトリの保存」を参照してください。

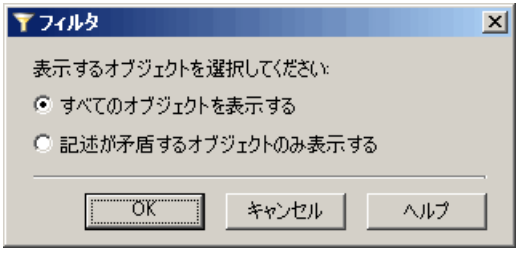
［ターゲット リポジトリ］表示枠に対するフィルタの設定

2 つのオブジェクト・リポジトリを結合した結果，ターゲット・オブジェクト・リポジトリに含まれるオブジェクトの数が非常に多くなることがあります。［ターゲット オブジェクト リポジトリ］表示枠でのナビゲーションや特定のオブジェクトの検索が容易になるように，オブジェクト・リポジトリ結合ツールでは，表示枠内のオブジェクトを絞り込み（フィルタ処理），結合中に解決された矛盾のあるオブジェクトだけを表示することができます。

注：フィルタは，［ターゲット オブジェクト リポジトリ］表示枠にどのオブジェクトを表示するかのみを決めるものです。どのオブジェクトをターゲット・オブジェクト・リポジトリに含めるのかを決めるものではありません。

［ターゲット オブジェクト リポジトリ］表示枠のオブジェクトを絞り込むには，次の手順を実行します。

1 ［**ツール**］ > ［**フィルタ**］を選択するか，［**フィルタ**］ボタンをクリックします。［フィルタ］ダイアログ・ボックスが表示されます。



ヒント：ステータス・バーの［**フィルタ**］アイコンをクリックしても，［フィルタ］ダイアログ・ボックスを表示できます。フィルタが現在使用中のときは，ステータス・バーにフィルタが［**オン**］と表示されます。

---

2 ターゲット・オブジェクト・リポジトリに表示するオブジェクトに対応したラジオ・ボタンを選択します。

- ［**すべてのオブジェクトを表示する**］: ターゲット・オブジェクト・リポジトリにあるオブジェクトがすべて表示されます。
- ［**記述が矛盾するオブジェクトのみ表示する**］: ターゲット・オブジェクト・リポジトリで，記述が矛盾しているオブジェクトだけが表示されます。

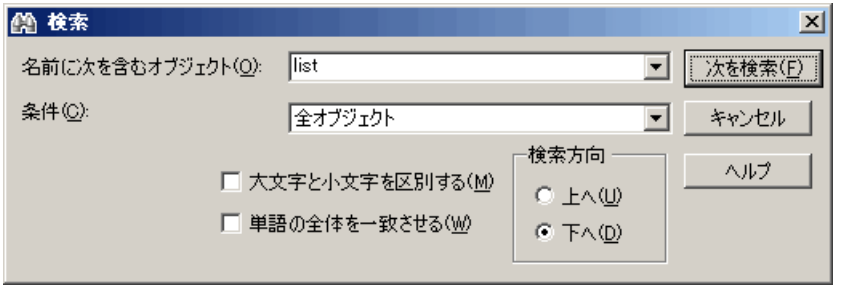
3 ［**OK**］をクリックします。表示枠内のオブジェクトが絞り込まれ，設定したオブジェクト・タイプのみがターゲット・オブジェクト・リポジトリに表示されます。フィルタ処理中は進行状況バーがステータス・バーに表示されます。

特定のオブジェクトの検索

オブジェクト・リポジトリ結合ツールの検索機能を使用して，名前に指定の文字列が含まれている 1 つ以上のオブジェクトをターゲット・オブジェクト・リポジトリの中で検索できます。見つかったオブジェクトは，対応する一次オブジェクト・リポジトリまたは二次オブジェクト・リポジトリの一方または両方でも強調表示されます。

オブジェクトを検索するには，次の手順を実行します。

1 ［**ナビゲート**］＞［**検索**］を選択するか，［**検索**］ボタンをクリックします。［検索］ダイアログ・ボックスが開きます。



2 ［**名前に次を含むオブジェクト**］ボックスに，検索するオブジェクトの名前の全体または一部を入力します。

3 ［**条件**］ボックスで，検索するオブジェクトを選択して検索対象を絞り込みます。次の条件を使用できます。

- ［**全オブジェクト**］
- ［**一方のソースにのみ存在するオブジェクト**］
- ［**矛盾するオブジェクト**］
- ［**矛盾するオブジェクトまたは一方にのみ存在するオブジェクト**］

4 次のオプションの一方または両方を選択して検索の絞り込みに役立てることができます。

- ［**大文字と小文字を区別する**］: 検索の際に大文字と小文字を区別します。［**大文字と小文字を区別する**］を選択した場合，大文字 / 小文字が［**名前に次を含むオブジェクト**］ボックスに入力した文字列と正確に一致する対象のみが QuickTest によって検索されます。
- ［**単語の全体を一致させる**］: 単語の一部ではなく単語全体が一致するオブジェクトを検索します。

5 現在のカーソルの位置からの検索方向として，［**上へ**］または［**下へ**］を指定します。検索操作がファイルの先頭または終端に達した後も，続けてオブジェクト・リポジトリ全体が検索されます。

6 ［**次を検索**］をクリックすると，ターゲット・オブジェクト・リポジトリ内で指定の条件に一致する次のオブジェクトが強調表示されます。

［検索］ダイアログ・ボックスを閉じて次のコマンドを使用することもできます。

- ［**次を検索**］ボタンをクリックするか，［**ナビゲート**］＞［**次を検索**］を選択すると，指定の条件に一致する次のオブジェクトが強調表示されます。

- ［**前を検索**］ボタンをクリックするか，［**ナビゲート**］＞［**前を検索**］を選択すると，指定の条件に一致する前のオブジェクトが強調表示されます。

ターゲット・オブジェクト・リポジトリの保存

オブジェクトの矛盾が意図どおりに解決されたことを確認したら，ターゲット・リポジトリをファイル・システムまたは Quality Center プロジェクト（QuickTest が現在 Quality Center プロジェクトに接続している場合）に保存できます。

オブジェクト・リポジトリの保存

新たに結合した共有オブジェクト・リポジトリをファイル・システムに保存できます。Quality Center に接続している場合は，結合した共有オブジェクト・リポジトリをプロジェクトのテスト・リソース・モジュールに保存することもできます。後で，オブジェクト・リポジトリを 1 つ以上の必要なアプリケーション領域と関連付けることで，コンポーネントからオブジェクト・リポジトリ内のオブジェクトにアクセスできるようになります。

オブジェクト・リポジトリをファイル・システムに保存するには，次の手順を実行します。

1 ［**ファイル**］＞［**保存**］を選択するか，［**保存**］ボタンをクリックします。ファイルを以前に保存したことがあれば，現在の変更内容が保存されます。ファイルがまだ保存されていない場合は，［共有オブジェクト リポジトリの保存］ダイアログ・ボックスが開きます。

2 サイドバーで，ファイルを保存する場所を選択します（File System や Quality Center Test Resources など）。

3 ファイルを保存するフォルダを参照し，そのフォルダを選択します。

4 ［**ファイル名**］ボックスに，ファイルの名前を入力します。ファイルを識別しやすいように，わかりやすい名前を使用します。オブジェクト・リポジトリの名前には，次の文字は使用できません。
¥ / : * " ? < > | ’

オブジェクト・リポジトリを Quality Center に保存する場合は，ファイル・パスに連続した 2 つのセミコロン（;;）を含める必要があります。

5 ［**保存**］をクリックします。

QuickTest によって，ファイル名に **.tsr** 拡張子が付けられ，指定された場所にオブジェクト・リポジトリが保存されます。そして［オブジェクト リポジトリ - 結合ツール］ウィンドウ内のターゲット・オブジェクト・リポジトリの上に，ファイル名とパスが表示されます。

第 9 章

共有オブジェクト・リポジトリの比較

QuickTest Professional では，オブジェクト・リポジトリ比較ツールを使用して，2 つの共有オブジェクト・リポジトリを比較し，オブジェクト名の違いやオブジェクト記述の違いといった，オブジェクトの相違点を表示できます。

本章の内容

- 共有オブジェクト・リポジトリの比較について（302 ページ）
- オブジェクト・リポジトリ比較ツールについて（304 ページ）
- オブジェクト・リポジトリ比較ツールのコマンドの使用（308 ページ）
- オブジェクトの相違点について（311 ページ）
- 色の設定の変更（312 ページ）
- オブジェクト・リポジトリの比較（314 ページ）
- 比較統計情報の表示（316 ページ）
- ［リポジトリ］表示枠に対するフィルタ処理（318 ページ）
- オブジェクト・リポジトリ・ビューの同期（319 ページ）
- 特定のオブジェクトの検索（320 ページ）

ヒント：バージョン・コントロールが有効な Quality Center 10.00 に接続している場合は，同じオブジェクト・リポジトリの 2 つのバージョンを比較できます。詳細については，949 ページ「QuickTest アセットのバージョンの表示と比較」を参照してください。

共有オブジェクト・リポジトリの比較について

QuickTest Professional では，オブジェクト・リポジトリ比較ツールを使用して 2 つのオブジェクト・リポジトリの既存のアセットを比較できます。このツールにはオブジェクト・リポジトリ・マネージャからアクセスでき，異なるオブジェクト・リポジトリ・リソースどうし，または同じオブジェクト・リポジトリ・リソースの異なるバージョンどうしを比較して，類似点，バリエーション，または変更点を識別できます。

2 つのオブジェクト・リポジトリ・ファイルはそれぞれ「**一次**」および「**二次**」のファイルと呼ばれますが，これらのファイル内のオブジェクト間の相違点は標準のルールに従って識別されます。比較プロセスの間，オブジェクト・リポジトリ・ファイルには変更は加えられません。オブジェクト・リポジトリ比較ツールで識別される相違点のタイプの詳細については，311 ページ「オブジェクトの相違点について」を参照してください。

比較処理の後，オブジェクト・リポジトリのオブジェクト（階層内のノードとして表示されます）が，比較ツールに視覚的に表示されます。相違点のあるオブジェクトや，一方のオブジェクト・リポジトリにのみ含まれている一意のオブジェクトを，ユーザが選択する色の設定に従って識別できます。一方のオブジェクト・リポジトリにのみ含まれているオブジェクトは，他方のオブジェクト・リポジトリでは「**存在しない**」というテキストによって識別されます。また，どちらのオブジェクト・リポジトリで個々のオブジェクトを選択しても，そのオブジェクトのプロパティと値を表示できます。

オブジェクト・リポジトリ比較ツールで表示される情報は，オブジェクト・リポジトリの管理や結合の際に利用できます。詳細については，第 9 章「共有オブジェクト・リポジトリの比較」または第 8 章「共有オブジェクト・リポジトリの結合」を参照してください。

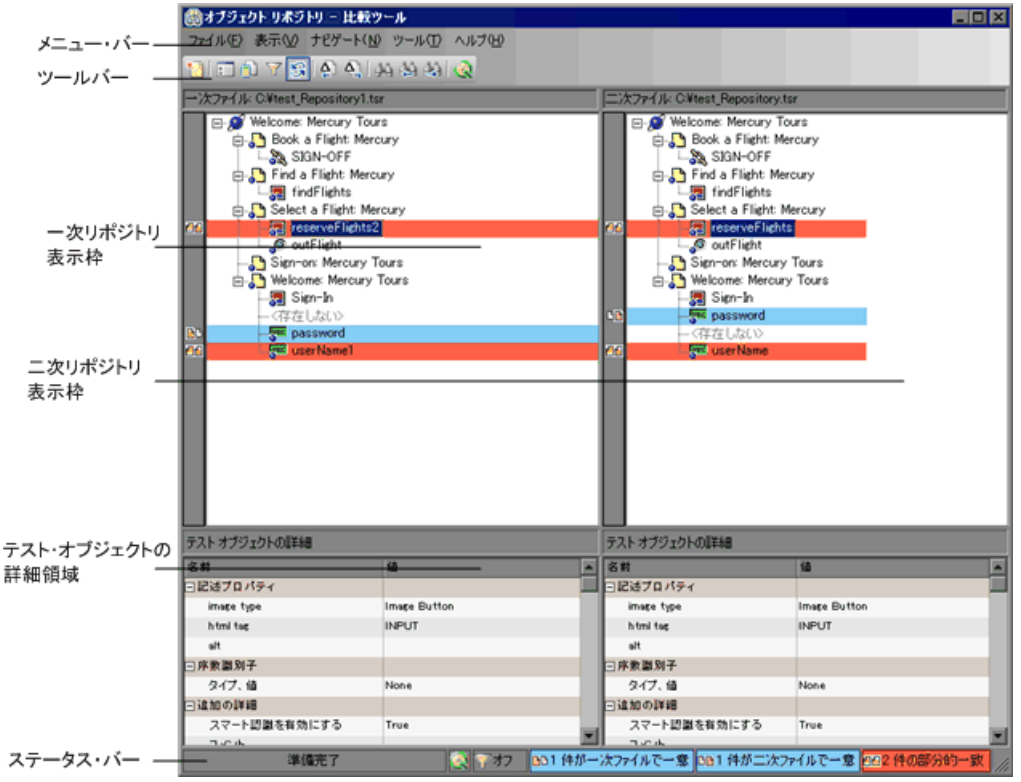
注：

- オブジェクト・リポジトリ比較ツールでは，チェックポイントまたは出力オブジェクトは比較されません。
- オブジェクト・リポジトリ比較ツールを開いているときは，オブジェクト・リポジトリ・マネージャやオブジェクト・リポジトリの結合ツールを使用することはできません。

オブジェクト・リポジトリ比較ツールについて

オブジェクト・リポジトリ比較ツールを開くには，オブジェクト・リポジトリ・マネージャで［**ツール**］＞［**オブジェクト リポジトリ比較ツール**］を選択します。

［オブジェクト リポジトリー比較ツール］ウィンドウの例を次に示します。



［オブジェクト リポジトリー比較ツール］ウィンドウには，次の主要な要素があります。

- **メニュー・バー**：オブジェクト・リポジトリ比較ツールのコマンドのメニューが表示されます。これらのコマンドについては，本章各所で説明します。メニュー・コマンドのショートカット・キーの詳細については，308 ページ「オブジェクト・リポジトリ比較ツールのメニュー・コマンドとショートカット・キー」を参照してください。
- **ツールバー**：よく使用するメニュー・コマンドのボタンがあります。オブジェクト・リポジトリの比較，オブジェクトの類似点および相違点の表示を行うことができます。ツールバー・ボタンについては，308 ページ「オブジェクト・リポジトリ比較ツールのツールバーのコマンド」を参照してください。
- **［リポジトリ］表示枠**：比較対象のオブジェクト・リポジトリ内のオブジェクトが階層表示されます。オブジェクト階層の左側のカラムには，それぞれのオブジェクトの比較を表すアイコンが各表示枠に表示されます。詳細については，305 ページ「［リポジトリ］表示枠について」を参照してください。
- **テスト オブジェクトの詳細領域**：［オブジェクト リポジトリ］表示枠で選択したオブジェクトのプロパティと値が表示されます。詳細については，305 ページ「［リポジトリ］表示枠について」を参照してください。
- **ステータス・バー**：比較プロセスのステータスや，オブジェクト・リポジトリの比較中に検出された相違点の詳細が表示されます。詳細については，307 ページ「ステータス・バーについて」を参照してください。

［リポジトリ］表示枠について

［オブジェクト・リポジトリ］表示枠には，比較対象のオブジェクト・リポジトリ・ファイルにあるオブジェクトについて，それらの階層，およびそれぞれのプロパティと値が表示されます。各オブジェクト階層の上にはファイル・パスが表示されます。

オブジェクトのステータスをひと目で確認できるように，オブジェクト・リポジトリ内のオブジェクト名のテキスト色と背景色が，検出された相違点のタイプに応じて異なる色で表示されます。

オブジェクト・リポジトリで相違点のタイプを示すのに使用される標準の色を変更することができます。詳細については，312 ページ「色の設定の変更」を参照してください。

また，［オブジェクト・リポジトリ］表示枠でオブジェクトの左側に示される次のアイコンによって，相違点を識別することもできます。

➤ 一次ファイルで一意のオブジェクト

➤ 二次ファイルで一意のオブジェクト

➤ 一次ファイルと二次ファイルの両方のファイルにあり，まったく同一ではないものの部分的に一致しているオブジェクト

相違点のすべてのタイプの詳細については，311 ページ「オブジェクトの相違点について」を参照してください。

［オブジェクト・リポジトリ］表示枠には次の機能があります。

- 一方の［オブジェクト・リポジトリ］表示枠でオブジェクトを選択すると，他方のファイル階層で対応するオブジェクトが検索され，強調表示されます。オブジェクトの選択時に CTRL キーを押すことで，他方のファイルの対応するオブジェクトを強調表示せずに，選択したオブジェクトのみを強調表示することができます。
- ［オブジェクト・リポジトリ］表示枠でオブジェクトを選択すると，オブジェクトのプロパティと値が，表示枠の下部にあるそれぞれの［**テスト オブジェクトの詳細**］領域に表示されます。
- ［オブジェクト・リポジトリ］表示枠でオブジェクトの左側にあるアイコンの上にカーソルを置くと，「**部分的一致**」や「**二次ファイルで一意**」などの比較の詳細がツールチップとして表示されます。
- ノードをダブルクリックするか，ノード名の左側にある展開記号（**+**）または折りたたみ記号（**-**）をクリックすると，親ノードの階層を展開または折りたたむことができます。また，［**表示**］メニューの［**すべて折りたたみ**］または［**すべて開く**］を選択しても，［オブジェクト・リポジトリ］表示枠の階層全体を展開または折りたたむことができます。

- ［**ナビゲート**］メニューの［**次の差異**］または［**前の差異**］を選択するか，ツールバーの［**次の差異**］または［**前の差異**］ボタンをクリックするか，キーバード・ショートカットを使用して，オブジェクト・リポジトリ階層内の次の差異または前の差異に直接移動できます。ショートカットの詳細については，308 ページ「オブジェクト・リポジトリ比較ツールのメニュー・コマンドとショートカット・キー」を参照してください。

- ［検索］ダイアログ・ボックスを使用して，［オブジェクト・リポジトリ］表示枠で 1 つ以上のオブジェクトを検索できます。詳細については，320 ページ「特定のオブジェクトの検索」を参照してください。
- 表示枠の縁をドラッグして，オブジェクト・リポジトリ比較ツール・ウィンドウ内で表示枠のサイズを変更できます。

ステータス・バーについて

ステータス・バーには，比較プロセスと，表示されている結果に関する，次の情報が表示されます。

- 進行状況バー。比較プロセス中にステータス・バーの左側に表示されます。プロセスが完了すると，［**準備完了**］が表示されます。

- Quality Center アイコン。QuickTest が Quality Center プロジェクトに接続しているときに表示されます。

- フィルタのステータス。［フィルタ］アイコンの横に表示されます。［**オフ**］は，オブジェクト・リポジトリがフィルタ処理されておらず，すべてのオブジェクトが表示されていることを示します。［**オン**］は，フィルタが有効であり，オブジェクトの一部がフィルタによって表示から除外された可能性があることを示します。［**フィルタ**］アイコンをクリックすると［フィルタ］ダイアログ・ボックスを表示できます。詳細については，318 ページ「[リポジトリ] 表示枠に対するフィルタ処理」を参照してください。
- 比較中に検出された相違点の数が，次のように表示されます。

  - 一次ファイルで一意のオブジェクトの数

  - 二次ファイルで一意のオブジェクトの数

  - 一次ファイルと二次ファイルの両方のファイルにあり，まったく同一ではないものの部分的に一致しているオブジェクトの数

相違点のすべてのタイプの詳細については，311 ページ「オブジェクトの相違点について」を参照してください。

オブジェクト・リポジトリ比較ツールのコマンドの使用

オブジェクト・リポジトリ比較ツールのコマンドは，メニュー・バーまたはツールバーから選択できます。ショートカット・キーを押して，特定のコマンドを実行することもできます。

オブジェクト・リポジトリ比較ツールのツールバーのコマンド

よく使用するコマンドは，ツールバーのボタンをクリックして実行できます。

	説明
	新規比較（309 ページ「[ファイル］メニューのコマンド」を参照）
	色設定（310 ページ「[ツール］メニューのコマンド」を参照）
	統計情報（309 ページ「[表示］メニューのコマンド」を参照）
	フィルタ（310 ページ「[ツール］メニューのコマンド」を参照）
	同期化されたノード（310 ページ「[ナビゲート］メニューのコマンド」を参照）
	前の差異（310 ページ「[ナビゲート］メニューのコマンド」を参照）
	次の差異（310 ページ「[ナビゲート］メニューのコマンド」を参照）
	検索（310 ページ「[ナビゲート］メニューのコマンド」を参照）
	前を検索（310 ページ「[ナビゲート］メニューのコマンド」を参照）
	次を検索（310 ページ「[ナビゲート］メニューのコマンド」を参照）
	Quality Center への接続（309 ページ「[ファイル］メニューのコマンド」を参照）

オブジェクト・リポジトリ比較ツールのメニュー・コマンドとショートカット・キー

よく使用するコマンドは，ツールバー・ボタンをクリックするか，該当するメニュー・オプションを選択することで実行できます。コマンドの一部は，該当するショートカット・キーを押して実行することもできます。

［ファイル］メニューのコマンド

［**ファイル**］メニューの次のコマンドを使用して，オブジェクト・リポジトリの比較を管理できます。

	コマンド	ショートカット・キー	機能
	［**新規比較**］	CTRL+N	新しい比較操作を実行する対象となる 2 つのオブジェクト・リポジトリを指定できます。
	［**Quality Center への接続時間**］		QuickTest を Quality Center プロジェクトに接続できます。詳細については，46 ページ「Quality Center プロジェクトへの接続」を参照してください。
	［**終了**］		［オブジェクト リポジトリ - 比較ツール］ウィンドウを閉じます。

［表示］メニューのコマンド

［**表示**］メニューでは次のコマンドを実行できます。

	コマンド	機能
	［**統計情報**］	［統計情報］ダイアログ・ボックスを開きます。このダイアログ・ボックスには，2 つのリポジトリどうしの比較が示され，検出された相違点の数とタイプが表示されます。詳細については，316 ページ「比較統計情報の表示」を参照してください。
	［**すべて折りたたみ**］	両方の比較表示枠内の階層全体を折りたたみます。 **ヒント**：展開されているノードをダブルクリックすると，そのノードが両方の表示枠で同時に折りたたまれます。
	［**すべて展開**］	両方の比較表示枠内の階層全体を展開します。 **ヒント**：折りたたまれているノードをダブルクリックすると，そのノードが両方の表示枠で同時に展開されます。

［ナビゲート］メニューのコマンド

［**ナビゲート**］メニューでは次のコマンドを実行できます。

	コマンド	ショートカット・キー	機能
	［次の差異］	F4	オブジェクト・リポジトリ内でオブジェクト間の次の相違点を検索します。
	［前の差異］	SHIFT+F4	オブジェクト・リポジトリ内でオブジェクト間の前の相違点を検索します。
	［検索］	CTRL+F	［検索］ダイアログ・ボックスを開きます。
	［次を検索］	F3	［検索］ダイアログ・ボックスでの検索条件に従って，オブジェクト・リポジトリ内で次のオブジェクトを検索します。
	［前を検索］	SHIFT+F3	［検索］ダイアログ・ボックスでの検索条件に従って，オブジェクト・リポジトリ内で前のオブジェクトを検索します。

［ツール］メニューのコマンド

［**ツール**］メニューでは次のコマンドを実行できます。

	コマンド	機能
	［同期化されたノード］	2 つの［オブジェクト・リポジトリ］表示枠を，同時に操作するか，または互いに独立して操作できます。詳細については，319 ページ「オブジェクト・リポジトリ・ビューの同期」を参照してください。

	コマンド	機能
	［フィルタ］	［フィルタ］ダイアログ・ボックスを開きます。このダイアログ・ボックスを使用して，表示するテスト・オブジェクトの一致のタイプを指定できます。詳細については，318 ページ「［リポジトリ］表示枠に対するフィルタ処理」を参照してください。
	［色の設定］	［設定］ダイアログ・ボックスを開きます。このダイアログ・ボックスを使用して，比較表示枠に表示されるオブジェクト名と空のノードについて，それらのテキストの色と背景を指定できます。 詳細については，312 ページ「色の設定の変更」を参照してください。

［ヘルプ］メニューのコマンド

［**ヘルプ**］メニューでは次のコマンドを実行できます。

コマンド	ショートカット・キー	機能
［オブジェクト リポジトリ比較ツールのヘルプ］	F1	オブジェクト・リポジトリ比較ツールのヘルプを開きます。

オブジェクトの相違点について

比較ツールでは，比較プロセス中にオブジェクトが自動的に識別され，次のいずれかのタイプに分類されます。

- **同じ**：両方のオブジェクト・リポジトリ・ファイルに出現するオブジェクト。これらのオブジェクトでは名前とプロパティの相違点はありません。
- **同じ記述で異なる名前**：両方のオブジェクト・リポジトリ・ファイルに出現し，名前が異なるものの，同じ記述プロパティと値を持つオブジェクト。
- **類似の記述**：両方のオブジェクト・リポジトリ・ファイルに出現し，類似しているもののまったく同じではない記述プロパティと値を持つオブジェクト。一方のオブジェクトが常に他方のプロパティ・セットのサブセットを持っています。これは，同じオブジェクトで記述の詳細が少ない可能性があることを示唆しています。たとえば，2 番目のオブジェクト・リポジトリ内にある Button_1

という名前のオブジェクトが，1 番目のオブジェクト・リポジトリにある Button_1 という名前のオブジェクトと同じ記述プロパティおよび値を持っているものの，さらに追加のプロパティと値を持っているとします。

記述を持たない Page オブジェクトや Browser オブジェクトなどのオブジェクトは名前のみ比較されます。両方のオブジェクト・リポジトリに同じオブジェクトが含まれていて，それらの名前が異なる場合は，2 つの別々のオブジェクトとしてオブジェクト・リポジトリに表示されます。

注：オブジェクト・リポジトリ比較ツールでは，オブジェクト名の一致よりもオブジェクト記述の一致の方が優先されます。このため，比較プロセス中にオブジェクト・ノードがリンクされる場合もあればリンクされない場合もあります。

- **一次ファイルで一意**または**二次ファイルで一意**：一方のオブジェクト・リポジトリ・ファイルにのみ出現するオブジェクト。
- **存在しない**：どのリポジトリ・ファイルにもないが，ほかのファイルに存在するオブジェクト。

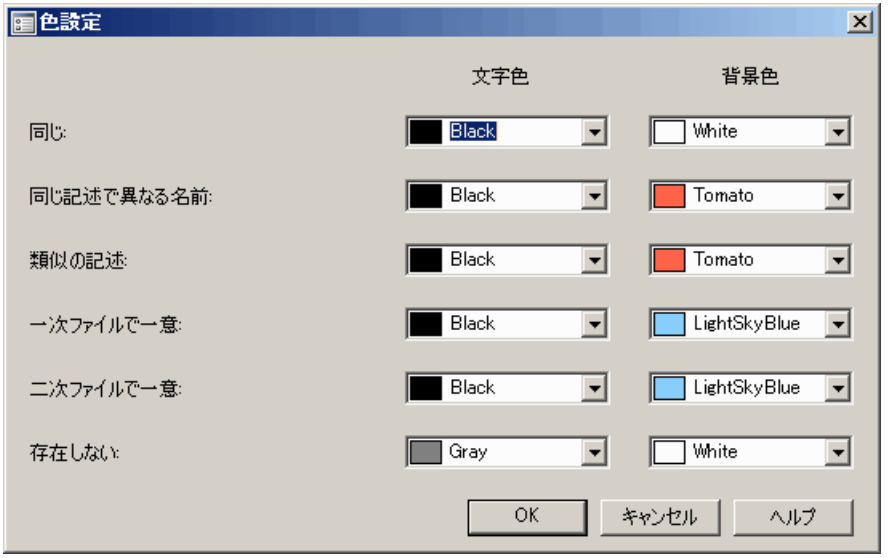
色の設定の変更

オブジェクト名と，ほかのオブジェクト・リポジトリにのみ存在するオブジェクトを示す空のノードについて，それらのテキストと背景は，比較ツールのウィンドウに相違点のタイプに応じた標準の色で表示されます。これを利用して，［オブジェクト・リポジトリ］表示枠で各オブジェクトのステータスを確認できます。これらのテキストの色は［統計情報］ダイアログ・ボックスでも使用されます。

必要に応じて，標準の色の設定を変更できます。

色の設定を変更するには，次の手順を実行します。

1 ［**ツール**］ > ［**色設定**］を選択するか，［**色設定**］ボタンをクリックします。［色設定］ダイアログ・ボックスが開きます。



2 それぞれの相違点のタイプについて，テキスト・ボックスの横にある下矢印 をクリックし，識別するテキストの色と背景色を［Custom］，［Web］，または［System］タブから選択します。

3 ［**OK**］をクリックします。オブジェクト・リポジトリの比較の実行後，それぞれの［オブジェクト・リポジトリ］表示枠内のオブジェクト名と空のノードが，ここでの選択項目に従って表示されます。

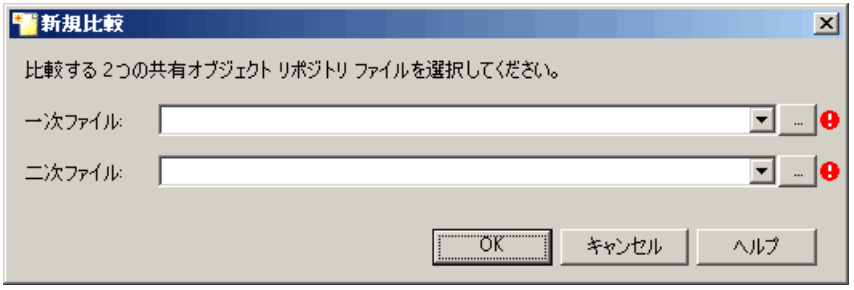
オブジェクト・リポジトリの比較

オブジェクト・リポジトリ比較ツールを使用すると，オブジェクト間の相違点の識別方法を定義した定義済みの設定に従って，2 つのオブジェクト・リポジトリを比較できます。

2 つのオブジェクト・リポジトリを比較するには，次の手順を実行します。

1 QuickTest Professional で，［**リソース**］＞［**オブジェクト リポジトリ マネージャ**］を選択します。

2 ［オブジェクト リポジトリ マネージャ］で，［**ツール**］＞［**オブジェクト リポジトリ比較ツール**］を選択します。［オブジェクト リポジトリ - 比較ツール］ウィンドウの手前に，［新規比較］ダイアログ・ボックスが開きます。



ヒント：

- ［オブジェクト リポジトリ - 比較ツール］ウィンドウがすでに開いている場合は，［**ファイル**］＞［**新規比較**］を選択するか，ツールバーの［**新規比較**］ボタンをクリックすると，［新規比較］ダイアログ・ボックスが開きます。
- オブジェクト・リポジトリを比較する前に設定済みの設定を変更するには，［**キャンセル**］をクリックして［新規比較］ダイアログ・ボックスを閉じ，312 ページ「色の設定の変更」の説明に従って設定を変更した後，比較を実行する必要があります。

---

3 ［**一次ファイル**］および［**二次ファイル**］ボックスで，比較を行う **.tsr** オブジェクト・リポジトリ・ファイルを入力または参照し，選択します。オブジェクト・リポジトリ・ファイルはファイル・システムまたは Quality Center に配置できます。標準設定では，オブジェクト・リポジトリ比較ツールを使用して比較のために選択した最後のファイルが，これらのボックスに表示されます。各ボックスの横にある下矢印 をクリックすると，最近使用したファイルを表示および選択できます。

注：

- 拡張子が **.tsr** でないファイル，パスが正しくない **.tsr** ファイル，または存在しないファイルを入力した場合は，対応するテキスト・ボックスの横に警告アイコンが表示されます。アイコンの上にポインタを置くと，エラーを説明するツールチップが表示されます。正しいパスを持つ既存の **.tsr** ファイルを入力または選択してください。
- バージョン 9.0 よりも前の QuickTest を使用して作成されたオブジェクト・リポジトリを比較する場合は，はじめにオブジェクト・リポジトリ・マネージャでそのオブジェクト・リポジトリを開いてから保存して新しい形式に更新する必要があります。
- Quality Center に接続している場合は，ファイル・システムと同様に Quality Center からもオブジェクト・リポジトリに入ることができます（参照できます）。

---

4 ［**OK**］をクリックします。オブジェクト・リポジトリ比較ツールによって，選択したオブジェクト・リポジトリ内のオブジェクトが比較され，その結果が［統計］ダイアログ・ボックスとして［オブジェクト リポジトリ - 比較ツール］ウィンドウの上に表示されます。

5 316 ページ「比較統計情報の表示」の説明を参考にして，統計情報を確認し，［**閉じる**］をクリックします。

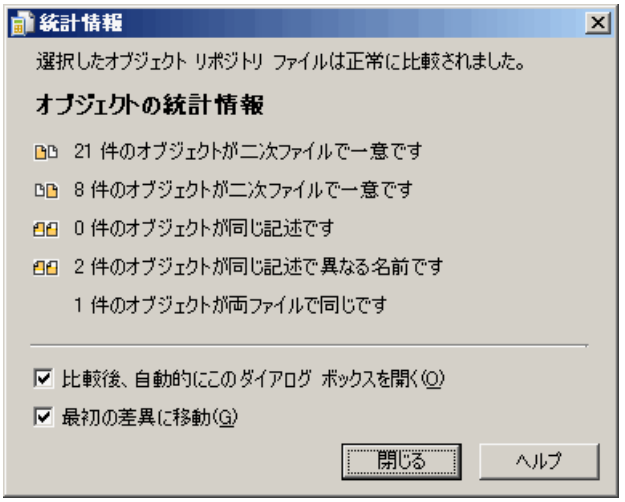
6 ［オブジェクト リポジトリ - 比較ツール］ウィンドウでは，次を実行できます。

- オブジェクト・リポジトリ内のオブジェクトにフィルタを適用できます。詳細については，318 ページ「［リポジトリ］表示枠に対するフィルタ処理」を参照してください。

➤ オブジェクト・リポジトリ内のオブジェクトを検索できます。詳細については，320 ページ「特定のオブジェクトの検索」を参照してください。

比較統計情報の表示

2 つのオブジェクト・リポジトリを比較した後，オブジェクト・リポジトリ比較ツールから［統計］ダイアログ・ボックスが表示されます。このダイアログ・ボックスには，ファイルがどのように比較されたかが示され，検出された相違点の数とタイプが表示されます。



ヒント：［**比較後、自動的にこのダイアログ ボックスを開く**］チェック・ボックスをクリアすると，オブジェクト・リポジトリを比較するたびに［統計情報］ダイアログ・ボックスが表示されないようにできます。比較ツールのウィンドウで［**表示**］＞［**統計情報**］を選択するか，ツールバーの［**統計情報**］ボタンをクリックすることで，いつでも［統計情報］ダイアログ・ボックスに比較の統計情報を表示できます。

［統計情報］ダイアログ・ボックスには次の情報が表示されます。

- オブジェクト・リポジトリ内のオブジェクト間の相違点の数とタイプ。相違点のタイプについては，311 ページ「オブジェクトの相違点について」を参照してください。
- 一次ファイルまたは二次ファイルで一意となっているか，または両方のファイルで同一となっている項目の数。

オブジェクト統計内の各相違点のタイプに対して表示されるアイコンは，［オブジェクト・リポジトリ］表示枠で使用されているものと同じです。詳細については，305 ページ「［リポジトリ］表示枠について」を参照してください。

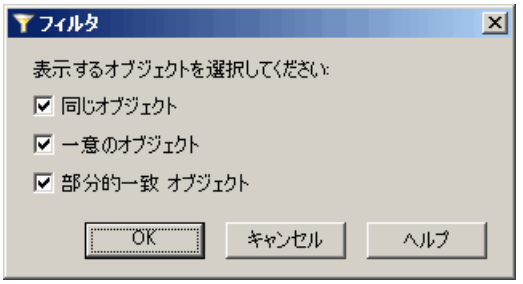
ヒント：［**最初の差異に移動**］チェック・ボックスを選択すると，［統計情報］ダイアログ・ボックスを閉じた直後に，オブジェクト・リポジトリ内の最初の相違点に移動します。

［リポジトリ］表示枠に対するフィルタ処理

オブジェクト・リポジトリには多数のオブジェクトが含まれている可能性があります。［オブジェクト・リポジトリ］表示枠でのナビゲーションや特定のオブジェクトの検索が容易になるように，オブジェクト・リポジトリ比較ツールでは，オブジェクトを絞り込み（フィルタ処理），表示を必要とするオブジェクトだけを表示することができます

［オブジェクト・リポジトリ］表示枠のオブジェクトをフィルタ処理するには，次の手順を実行します。

1 ［**ツール**］＞［**フィルタ**］を選択するか，ツールバーの［**フィルタ**］ボタンをクリックします。［フィルタ］ダイアログ・ボックスが表示されます。



ヒント：フィルタが現在使用中の場合は，ツールバーの［**フィルタ**］ボタンが境界線で囲まれます。また，ステータス・バーでフィルタが「**ON**」と表示されます。ステータス・バーの［**フィルタ**］アイコンをクリックして，［フィルタ］ダイアログ・ボックスを開くことができます。

2 オブジェクト・リポジトリに表示するオブジェクトに応じて，チェック・ボックスを 1 つ以上選択します。

- ［**同じオブジェクト**］：両方のオブジェクト・リポジトリ・ファイルに出現し，名前やプロパティに相違点がないオブジェクト。
- ［**一意のオブジェクト**］：一次オブジェクト・リポジトリ・ファイルにのみ出現するオブジェクト，または二次オブジェクト・リポジトリ・ファイルにのみ出現するオブジェクト。

➤［**部分的一致オブジェクト**］: 一致するものの，名前や記述に相違点がある，オブジェクト・リポジトリ・ファイル内のオブジェクト。

ヒント：すべてのチェック・ボックスを選択すると，両方のオブジェクト・リポジトリにあるオブジェクトがすべて表示されます。

3［**OK**］をクリックします。表示枠内のオブジェクトが絞り込まれ，設定したオブジェクト・タイプのみがオブジェクト・リポジトリに表示されます。

オブジェクト・リポジトリ・ビューの同期

オブジェクト・リポジトリ比較ツールでは，2 つのオブジェクト・リポジトリを独立して操作できます。各種の表示枠のサイズを変更して，オブジェクト・リポジトリに格納されているオブジェクトの一部だけを表示することもできます。このため，大きなオブジェクト・リポジトリを扱っている場合に，各種の表示枠にオブジェクト・リポジトリ階層の異なる領域が表示され，比較プロセスの影響を受ける特定のオブジェクトを探して追跡することが難しくなることがあります。

両方の表示枠に同じオブジェクトが表示されるようにオブジェクト・リポジトリを同期化するには，一次オブジェクト・リポジトリ，および二次オブジェクト・リポジトリに現在表示されているオブジェクトを選択し，ツールバーの［**同期化されたノード**］ボタンをクリックします。一致するノードが他方のオブジェクト・リポジトリで強調表示され，両方のオブジェクト・リポジトリが同時にスクロールします。

ヒント：オブジェクト・リポジトリが現在同期化されている場合は，ツールバーの［**同期化されたノード**］ボタンが境界線で囲まれます。2 つのオブジェクト・リポジトリを独立して操作するには，もう一度［**同期化されたノード**］ボタンをクリックします。オブジェクト・リポジトリが同期化されている場合，オブジェクトの選択時に CTRL キーを押すことで，選択したオブジェクトのみを強調表示することもできます。

特定のオブジェクトの検索

オブジェクト・リポジトリ比較ツールの検索機能を使用して，名前に指定の文字列が含まれている 1 つ以上のオブジェクトを選択したオブジェクト・リポジトリの中で検索できます。検索されたオブジェクトが他方のオブジェクト・リポジトリに存在する場合は，他方のオブジェクト・リポジトリでも強調表示されます。

オブジェクトを検索するには，次の手順を実行します。

1 検索するオブジェクトが含まれている［オブジェクト・リポジトリ］表示枠をクリックします。

2 ［**ナビゲート**］＞［**検索**］を選択するか，ツールバーの［**検索**］ボタンをクリックします。［検索］ダイアログ・ボックスが開きます。

3 ［**名前に次を含むオブジェクト**］ボックスに，検索するオブジェクトの名前の全体または一部を入力します。ボックスの横にある下矢印 をクリックすると，最近使用した文字列を表示および選択できます。

4 ［**条件**］ボックスで，検索するオブジェクトを選択して検索対象を絞込みます。次の条件を使用できます。

- **全オブジェクト**
- **一意のオブジェクト**
- **部分的一致オブジェクト**
- **一意または部分的一致オブジェクト**

5 次のオプションの一方または両方を選択して検索の絞り込みに役立てることができます。

- ［**大文字と小文字を区別する**］: 検索の際に大文字と小文字を区別します。［**大文字と小文字を区別する**］を選択した場合，大文字 / 小文字が［**名前に次を含むオブジェクト**］ボックスに入力した文字列と正確に一致する対象のみが QuickTest によって検索されます。
- ［**単語の全体を一致させる**］: 単語の一部ではなく単語全体が一致するオブジェクトを検索します。

6 現在のカーソル位置からの検索方向として，［**上へ**］または［**下へ**］を指定します。検索操作がオブジェクト・リポジトリの先頭または終端に達した後も，続けてファイル全体が検索されます。

7 ［**次を検索**］ボタンをクリックすると，オブジェクト・リポジトリ内で指定の条件に一致する次のオブジェクトが強調表示されます。

［検索］ダイアログ・ボックスを閉じて次のコマンドを使用することもできます。

- ツールバーの［**次を検索**］ボタンをクリックするか，［**ナビゲート**］＞［**次を検索**］を選択するか，あるいは F3 キーを押すと，指定の条件に一致する次のオブジェクトが強調表示されます。

- ツールバーの［**前を検索**］ボタンをクリックするか，［**ナビゲート**］＞［**前を検索**］を選択するか，あるいは SHIFT+F3 キーを押すと，指定の条件に一致する前のオブジェクトが強調表示されます。

第 III 部

関数とその他のプログラミング・タスクの定義

第 10 章

関数ライブラリ・ウィンドウを使用した作業

QuickTest 関数ライブラリ・ウィンドウを使用して，VBScript を利用した関数ライブラリを作成できます。本章では，VBScript について簡単に紹介し，いくつかの簡単なプログラミング・テクニックを使って関数ライブラリを拡張する方法を示します。

本章の内容

- 関数ライブラリ・ウィンドウを使用した作業について（326 ページ）
- エキスパート・ビューの理解と使用（857 ページ）
- 関数ライブラリ内での操作（334 ページ）
- VBScript の基本的な構文の理解（343 ページ）
- プログラム的記述の使用（350 ページ）
- プログラムによるアプリケーションの実行と終了（362 ページ）
- コメント，フロー制御，そのほかの VBScript ステートメントの使用（363 ページ）
- 認識プロパティ値の取得と設定（370 ページ）
- ネイティブ・プロパティおよび操作へのアクセス（372 ページ）
- DOS コマンドの実行（374 ページ）
- Windows API を使用したテストおよび関数ライブラリの拡張（374 ページ）
- 実行セッション中に報告するステップの選択（377 ページ）

関数ライブラリ・ウィンドウを使用した作業について

［関数ライブラリ］ウィンドウを使用して，関数ライブラリの作成や関数ライブラリを使った作業ができます。VBScript での作業の詳細については，QuickTest の［**ヘルプ**］メニューから VBScript に関するドキュメントを参照してください（［**ヘルプ**］＞［**QuickTest Professional ヘルプ**］＞［**VBScript Reference**］）。

オブジェクトの操作や，アプリケーションから情報を取得するステートメントを追加できます。たとえば，オブジェクトが存在するかどうかを検査するステップの追加や，操作の戻り値の取得ができます。

手作業でまたはステップ・ジェネレータを使用して，関数ライブラリにステップを追加することができます。ステップ・ジェネレータの使用法の詳細については，『**HP QuickTest Professional ユーザーズ・ガイド**』を参照してください。

関数ライブラリはいつでも印刷できます。印刷出力には追加の情報を含めることもできます。関数ライブラリの印刷の詳細については，399 ページ「関数ライブラリの印刷」を参照してください。

関数ライブラリでのステートメントの生成

ステートメントを生成するには，次のいずれかの方法を使用します。

- ステップ・ジェネレータを使用して，メソッドおよび関数を使用するステップを追加できます。詳細については，『**HP QuickTest Professional ユーザーズ・ガイド**』を参照してください。
- 操作を実行する VBScript ステートメントを手作業で挿入できます。QuickTest には，正しい構文を厳守して，ステートメントの関連項目を選択するのを支援する機能が用意されています。
  - ［**ステートメントの自動補完を行う**］（IntelliSense）：このオプションを有効にすると，関数ライブラリで入力するときに，変数，操作，またはプロパティをステートメントに対して選択し，関連する構文を表示するのに役立ちます。詳細については，327 ページ「ステートメントの自動補完の使用（IntelliSense）」を参照してください。
  - ［**VBScript 構文を自動的に拡張する**］：このオプションを有効にすると，関数ライブラリで VBScript のキーワードを入力し始めたときに，QuickTest によって，関連する構文またはブロックがスクリプトに自動的に追加されま

す。詳細については，333 ページ「VBScript 構文の自動補完」を参照してください。

ステートメントの自動補完の使用（IntelliSense）

関数ライブラリ，IntelliSense（QuickTest に含まれているステートメントの自動補完機能）で入力すると，ステートメントに対する変数，操作，またはプロパティをドロップダウン・リストから選択し，関連する構文を表示できます。

［**ステートメントの自動補完を行う**］オプションは標準で有効になっています。このオプションは，［エディタ オプション］ダイアログ・ボックスで設定と解除ができます。詳細については，第 11 章「関数ライブラリ・ウィンドウのカスタマイズ」を参照してください。

ヒント：

- 場合によっては，QuickTest は，実際のオブジェクトから IntelliSense 情報を取得する必要があります。このような場合には，関数ライブラリで入力するときに遅延が発生する場合があります。この遅延を回避するには，ステートメントの自動補完オプションを無力にします。
- テスト・オブジェクトのステートメントを生成しやすくするため，関数ライブラリ・ドキュメントでも以下のように IntelliSense がサポートされますが，関数に完全なオブジェクト階層ステートメントを含めることは一般にお勧めしません。関数は，ほかのオブジェクトでも使用できるように汎用的に作成することをお勧めします。
- ステートメントが正しく入力されていなかったり，構文エラーやその他の VBScript エラーが含まれたりしている場合は，QuickTest に IntelliSense 情報が表示されない場合があります。
- IntelliSense ドロップダウン・リストが表示されるフレームのサイズを変更すると，その後は，IntelliSense のドロップダウン・リストが表示されるときに，QuickTest によって新しいサイズが使用されます。
- 何も選択せずに IntelliSense ドロップダウン・リストを閉じるには，Esc キーを押します。

［**ステートメントの自動補完を行う**］オプションが有効の場合，次の種類の情報が提供されます。

➤ **利用可能な操作およびプロパティ**：ステートメントでオブジェクトまたはテスト・オブジェクトの後にピリオドを入力すると，入力したオブジェクトの後に追加できる操作およびプロパティのリストが表示されます。

操作またはプロパティの名前を入力すると，入力したテキストに一致する最初の操作またはプロパティが（アルファベット順に）強調表示されます。ENTER キーまたは SPACE キーを押すと，強調表示されている単語がステップに入力されます。

ヒント：利用可能な操作およびプロパティのリストが表示されていないときに操作またはプロパティの名前を入力した場合，オプションが 1 つしかない場合には CTRL キーを押しながら SPACE キーを押すと単語が自動的に補完され，複数ある場合にはリストが表示されて，入力したテキストに一致する最初の操作またはプロパティが（アルファベット順に）強調表示されます。ENTER キーまたは SPACE キーを押すと，強調表示されている単語がステップに入力されます。

QuickTest によって，テスト・オブジェクト，予約済みオブジェクト，関数で作成したオブジェクト，オブジェクトまたはテスト・オブジェクトが割り当てられる変数，およびオブジェクトを返すプロパティまたは操作に対して，このタイプの IntelliSense 情報が提供されます（利用できる場合）。

次に例を示します。

➤ ステートメントの中でテスト・オブジェクトの後にピリオドを入力すると，入力したオブジェクトの後に追加できる操作，プロパティおよび登録済みの関数のリストが表示されます。

➤ スクリプトで作成したオブジェクトの後にピリオドを入力すると（たとえば **CreateObject** メソッドを使用），そのオブジェクトに使用できる操作およびプロパティが表示されます。

➤ **Object** プロパティをステートメントで使用して，オブジェクト・データを現在［］起動しているアプリケーションで使用できる場合，オブジェクトに対するネイティブの操作とプロパティが表示されます。**Object** プロパティの詳細については，372 ページ「ネイティブ・プロパティおよび操作へのアクセス」を参照してください。

➤ **With** ステートメント内でピリオドを入力すると，QuickTest によって，関連するオブジェクトで利用できる操作およびプロパティのリストが表示されます。

注：With ステートメントを入力する場合（メニュー・コマンドを使用して作成しない場合），［**編集**］＞［**詳細設定**］＞［**Apply "With" to Script**］コマンドを使用して（または CTRL キーを押しながら W キーを押して），IntelliSense を **With** ステートメント内で有効にする必要があります。

➤ オブジェクトを変数に割り当てて，変数の名前に続けてピリオドを入力すると，オブジェクトで利用できる操作およびプロパティのリストが表示されます。

場合によっては，テストの編集中に変数の値を決定できない場合があります（たとえば，値が条件付きの割り当てによって設定されたり，別の関数によって返されたりする場合）。この場合には，変数の値が評価された最新のコード行に従って IntelliSense 情報が提供されます（存在する場合）。

以下に例を示します。

例 1：

```
行 1：Set x = CreateObject("Excel.Application")
行 2：z = GetValueFromUser()
行 3：If z = 2 Then
行 4：   Set x = CreateObject("Word.Application")
行 5：End If
行 6：x.
```

このテストの編集中は，QuickTest は，どのオブジェクトが実際に 6 行目の x に割り当てられるかを判断できません。しかし，x の値は 4 行目で独立して評価できるため，6 行目の変数 x に対するオブジェクト "Word.Application" に関連する IntelliSense 情報が表示されます。

例：

```
行 1：Set x = CreateObject("Excel.Application")
行 2：Set x = MyGetObject()
行 3：x.
```

このテストの編集中は，QuickTest は，**MyGetObject** 関数によって返されるオブジェクトのタイプを判断できません（2 行目）。このため，上記の例の 3 行目では，オブジェクト "Excel.Application" に関連する IntelliSense 情報が表示されます。この理由は，1 行目が x の値を評価できる最新のコード行であるためです。しかし，2 行目の前に値を評価できる行がなければ，3 行目の x に対する IntelliSense 情報は表示されません。

- **操作またはプロパティ構文**：操作またはプロパティの名前の後にスペースを入力すると，該当する構文が，その必須引数および任意引数とともに表示されます。操作またはプロパティを使用するステップを追加するときは，操作またはプロパティの必須引数に対して値を定義する必要があります。

  引数値の後にカンマを入力すると（ステップの最後の引数を除く），再び操作の構文が表示され，値を入力する必要がある次の引数が太字になります。

  また，引数を含む任意の操作または関数にカーソルを置き，CTRL + SHIFT + SPACE キーを押すか，［**編集**］＞［**詳細**］＞［**引数詳細**］を選択すると，その項目に対応するステートメント補完（引数の構文の）ツールチップが表示されます。

- **取り得る引数値** ：特定の操作では，値のリストが定義されている引数の前にスペースまたはカンマを入力すると，利用できる値のリストが表示されます。

- **利用可能な定数およびローカル変数**：定数またはローカル変数の名前を入力し始めると，入力した文字から始まる（現在のプログラムの範囲に関連する）定数およびローカル変数のリストが表示されます。一致する定数または変数が 1 つだけ定義されている場合は，その名前がステップに自動的に入力されます。

ヒント：CTRL キーを押しながら SPACE キーを押すと，追加が可能な対応する操作，プロパティ，VBScript 関数，ユーザ定義関数，VBScript 定数，およびユーティリティ・オブジェクトのリストが表示されます。

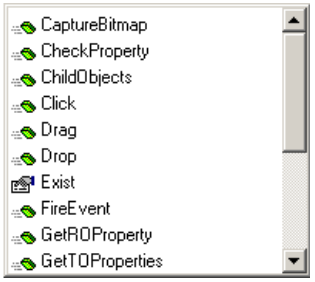
関数ライブラリでステートメントの自動補完機能を使用してステートメントを生成するには，次の手順を実行します。

1 ［**ステートメントの自動補完を行う**］オプションが選択されていることを確認します（［**ツール**］＞［**表示オプション**］＞［**一般**］タブ）。

2 関数ライブラリで作業をしている場合は，次の例のようにオブジェクトの階層全体を入力します。

```
Browser("Welcome: Mercury Tours").Page("Book a Flight:
Mercury).WebEdit("username")
```

3 オブジェクト記述の後ろには，("username"). のようにピリオド（.）を入力します。オブジェクトに対して利用できる操作およびプロパティのリストが表示されます。



ヒント：

- 操作またはプロパティの名前を入力すると，入力したテキストに一致する最初の操作またはプロパティが（アルファベット順に）強調表示されます。ENTER キーまたは SPACE キーを押すと，強調表示されている単語がステップに挿入されます。
- 利用可能な操作およびプロパティのリストが表示されていないときに，操作またはプロパティの名前を入力する場合は，CTRL キーを押しながら SPACE キーを押すか，または［**編集**］＞［**詳細設定**］＞［**単語入力候補**］を選択します。入力したテキストに一致する操作またはプロパティが 1 つしかない場合は，QuickTest によって操作またはプロパティの名前が自動的に補完されます。複数ある場合は，QuickTest によってリストが表示され，入力したテキストに一致する最初の操作またはプロパティが（アルファベット順に）強調表示します。ENTER キーまたは SPACE キーを押すと，強調表示されている単語がステップに挿入されます。

---

4 リスト内の操作またはプロパティをダブルクリックするか，矢印キーを使って操作またはプロパティを選択し，ENTER キーを押します。QuickTest によって，操作またはプロパティがステートメントに挿入されます。操作またはプロパティに引数が含まれている場合は，ツールチップにその操作またはプロパティの構文が表示されます。

上の例では，**ReportEvent** メソッドには 4 つの引数があります。

ヒント：また，引数を含む任意の操作または関数にカーソルを置き，CTRL + SHIFT + SPACE キーを押すか，［**編集**］＞［**詳細**］＞［**引数詳細**］を選択すると，その項目に対応するステートメント補完（引数の構文の）ツールチップが表示されます。

---

5 操作の後ろに操作の引数を入力します。

QuickTest 操作の詳細と例については,『**HP QuickTest Professional Object Model Reference**』（英語版）を参照してください。

VBScript の構文の詳細については，343 ページ「VBScript の基本的な構文の理解」を参照してください。

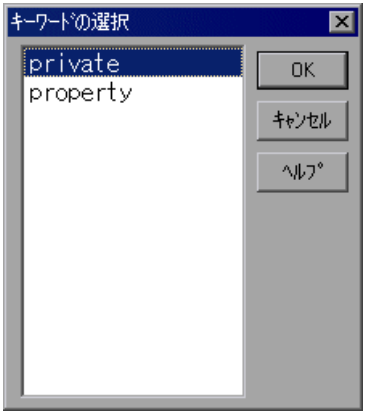
VBScript 構文の自動補完

関数ライブラリで VBScript のキーワードを入力し始めたとき，［**VBScript 構文を自動的に拡張する**］オプションが有効になっていれば，キーワードの最初の 2 文字が QuickTest によって自動的に認識され，対応する VBScript 構文またはブロックがスクリプトに追加されます。たとえば，空行の先頭で「if」という文字に続いてスペースを入力すると，自動的に次の構文が入力されます。

```
If Then
End If
```

［**VBScript 構文を自動的に拡張する**］オプションは標準で有効になっています。このオプションは，［エディタ オプション］ダイアログ・ボックスで設定と解除ができます。詳細については，381 ページ「エディタの動作のカスタマイズ」を参照してください。

入力した 2 文字が複数のキーワードの最初の 2 文字に一致する場合，［キーワードの選択］ダイアログ・ボックスが表示されるので，使用したいキーワードを選択できます。たとえば，「pr」という文字に続いてスペースを入力すると，private および property というキーワードを含んだ［キーワードの選択］ダイアログ・ボックスが表示されます。



そこでリストからキーワードを選択して［**OK**］をクリックします。対応する VBScript 構文またはブロックがスクリプトに自動的に挿入されます。

VBScript の構文の詳細については，343 ページ「VBScript の基本的な構文の理解」を参照してください。

関数ライブラリ内での操作

［移動］ダイアログ・ボックスまたはブックマークを使って，関数ライブラリ内の特定の行に移動できます。また，関数ライブラリ内の特定の文字列を検索し，必要があれば，その文字列を別の文字列で置換することが可能です。これらのオプションを使えば，長い関数のいくつものセクションの間を移動するのが容易になります。

［移動］ダイアログ・ボックスの使用

［移動］ダイアログ・ボックスを使用して，関数ライブラリ内の特定の行に移動できます。

ヒント：標準設定では，関数ライブラリに行番号が表示されます。行番号が表示されない場合は，［**ツール**］＞［**表示オプション**］＞［**一般**］タブの［**行番号を表示する**］オプションを選択します。エディタ・オプションの詳細については，第 11 章「関数ライブラリ・ウィンドウのカスタマイズ」を参照してください。

［移動］ダイアログ・ボックスを使って関数ライブラリ内の特定の行に移動するには，次の手順を実行します。

1 必要であれば，関数ライブラリをアクティブにします。

2 ［**編集**］＞［**移動**］を選択します。［移動］ダイアログ・ボックスが表示されます。

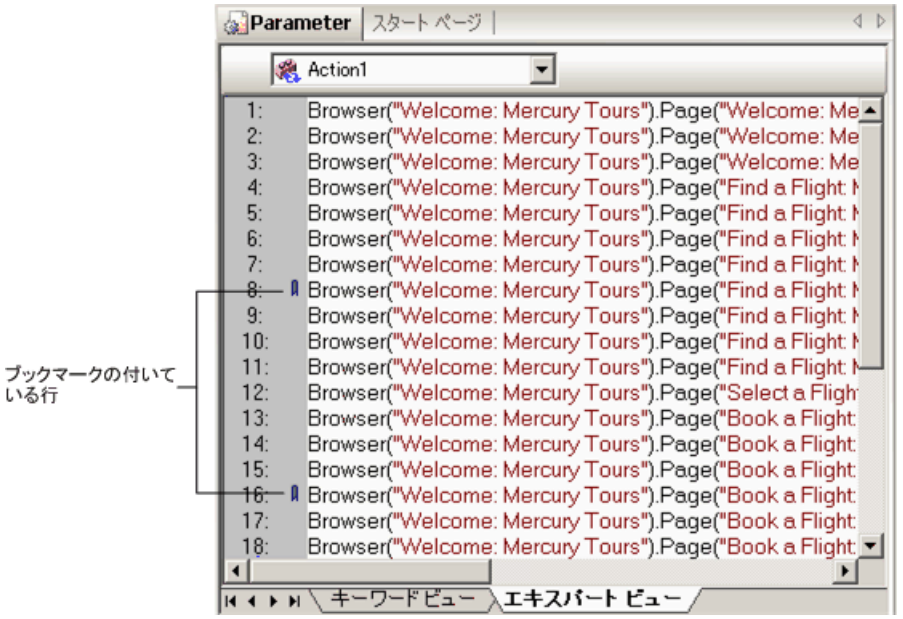
3 ［**行番号**］ボックスに移動先の行番号を入力し，［**OK**］をクリックします。指定した行の先頭にカーソルが移動します。

ブックマークの使用

ブックマークを使って，関数ライブラリの中の重要なセクションにマークを付けることで，さまざまな部分の間をより簡単に移動できます。ブックマークは，文書間を移動するときには維持されず，関数ライブラリとともに保存されません。

ブックマークを割り当てると，関数ライブラリ内で選択した行の左に，アイコンが付加されます。［ブックマーク］ダイアログ・ボックスの［**移動**］ボタンを使用して，ブックマークの付いている行に移動できます。

ブックマークの見え方はテストでも関数ライブラリでも同じです。次の例では，2 つのブックマークがテスト内のアクションに追加されています。

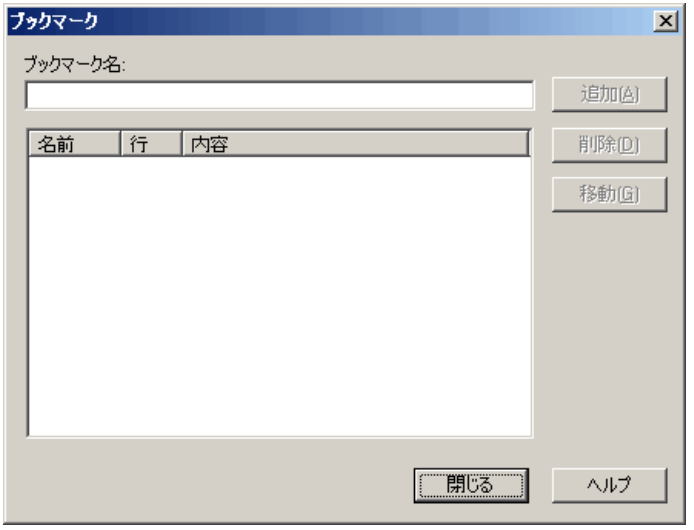


ブックマークを設定するには，次の手順を実行します。

1 必要であれば，関数ライブラリをアクティブにします。

2 ブックマークを設定する対象となる行をクリックします。

3 ［**編集**］＞［**ブックマーク**］を選択します。［ブックマーク］ダイアログ・ボックスが表示されます。



4 ［**ブックマーク名**］フィールドに，一意の名前を入力し，［**追加**］をクリックします。ブックマークが，対象となる行の行番号とそのテキスト内容とともに［ブックマーク］ダイアログ・ボックスに追加されます。また，関数ライブラリ内で選択した行の左に，ブックマーク・アイコン が付加されます。

5 ブックマークを削除するには，対象ブックマークを選択し，［**削除**］をクリックします。

特定のブックマークに移動するには，次の手順を実行します。

1 必要であれば，関数ライブラリをアクティブにします。

2 ［**編集**］＞［**ブックマーク**］を選択します。［ブックマーク］ダイアログ・ボックスが表示されます。

3 リストからブックマークを選択して［**移動**］ボタンをクリックします。QuickTest が関数ライブラリの該当する行に移動します。

テキスト文字列の検索

関数ライブラリ内で，検索する文字列を指定できます。リテラル・テキストを検索することも，正規表現を使用した高度な検索を行うこともできます。また，ほかのオプションを使用して検索結果の絞込みを行うことも可能です。

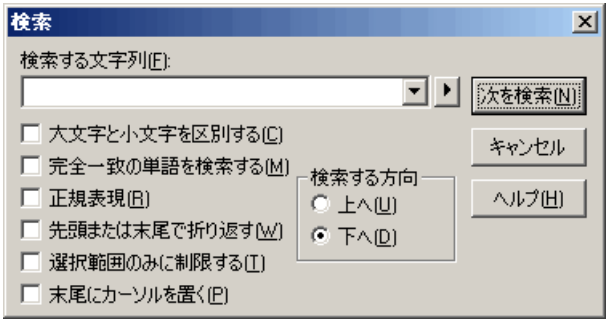
文字列を検索するには，次の手順を実行します。

1 関数ライブラリで，次のいずれかを実行します。

➤［**検索**］ボタンをクリックします。

➤［**編集**］＞［**検索**］を選択します。

［検索］ダイアログ・ボックスが開きます。



2［**検索する文字列**］ボックスに，検索する文字列を入力します。

3 指定する文字列の中で正規表現を使用したい場合には，矢印ボタン をクリックして正規表現を選択します。リストから正規表現を選択すると，その表現が［**検索する文字列**］ボックス内のカーソルの位置に自動的に挿入されます。詳細については，342 ページ「［検索］および［置換］ダイアログ・ボックスにおける正規表現の使用」を参照してください。

4 次の任意のオプションを選択して検索結果の絞込みを行うことも可能です。

➤［**大文字と小文字を区別する**］：検索の際に大文字と小文字を区別します。［**大文字と小文字を区別する**］を選択した場合，大文字小文字が，［**検索する文字列**］ボックスに入力した文字列と正確に一致する対象のみが QuickTest によって検索されます。

➤［**単語の全体を一致させる**］：単語の一部ではなく単語全体が一致する文字列を検索します。

- ［**正規表現**］：指定した文字列が正規表現として処理されます。リストから正規表現を選択した場合には，このオプションが自動的に選択されます。
- ［**先頭または末尾で折り返す**］：検索の方向に応じて，検索が関数ライブラリの先頭または末尾に達したときに，それらの先頭または末尾から検索を続けます。
- ［**選択範囲のみに制限する**］：関数ライブラリの中で選択されているテキストの範囲内に限定して検索を行います。
- ［**末尾にカーソルを置く**］：検索対象文字列が見つかったときに，その文字列を強調表示して，文字列の末尾にカーソルを移動します。

5 関数ライブラリ内の現在のカーソルの位置からどちらの方向に向かって検索を行うか指定します。［**上へ**］または［**下へ**］のいずれかを選択できます。

6 アクティブな関数ライブラリの中で，検索文字列の次の出現を強調表示するには，［**次を検索**］をクリックします。

テキスト文字列の置換

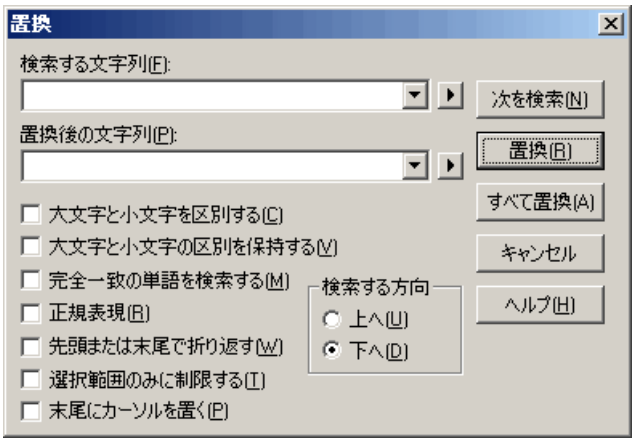
現在の関数ライブラリの中で検索する文字列，およびそれらの文字列を置き換えるための文字列を指定できます。リテラル・テキストを検索して置換することも，正規表現を使用した高度な処理を行うこともできます。また，ほかのオプションを使用して検索と置換の処理を詳細に設定することも可能です。

文字列を置換するには，次の手順を実行します。

1 関数ライブラリで，次のいずれかを実行します。

- ［**置換**］ボタンをクリックします。
- ［**編集**］＞［**置換**］を選択します。

［[置換］ダイアログ・ボックスが表示されます。



2 ［**検索する文字列**］ボックスに，検索する文字列を入力します。

3 ［**置換後の文字列**］ボックスに，見つかったテキストを置き換える文字列を入力します。

4 ［**検索する文字列**］または［**置換後の文字列**］に指定する文字列の中で正規表現を使用したい場合には，矢印ボタン▶をクリックして正規表現を選択します。リストから正規表現を選択すると，その表現が［**検索する文字列**］または［**置換後の文字列**］ボックス内のカーソルの位置に自動的に挿入されます。詳細については，342 ページ「[検索］および［置換］ダイアログ・ボックスにおける正規表現の使用」を参照してください。

5 次の任意のオプションを選択して検索結果の絞込みを行うことも可能です。

- ［**大文字と小文字を区別する**］：検索の際に大文字と小文字を区別します。［**大文字と小文字を区別する**］を選択した場合，大文字小文字が，［**検索する文字列**］ボックスに入力した文字列と正確に一致する対象のみが QuickTest によって検索されます。
- ［**大文字と小文字の区別を保持する**］：［**検索する文字列**］に指定した文字列について，全部小文字，全部大文字，先頭のみ大文字，大文字小文字入り混じりのそれぞれを調べます。［**置換後の文字列**］に指定した文字列は，大文字小文字が入り混じっている場合を除き，見つかった文字列と同じ大文字小文字に変換されます。大文字小文字が入り混じっている場合，［**置換後の文字列**］の文字列が変換されずにそのまま使用されます。

- ［**単語の全体を一致させる**］：単語の一部ではなく単語全体が一致する文字列を検索します。
- ［**正規表現**］：指定した文字列が正規表現として処理されます。リストから正規表現を選択した場合には，このオプションが自動的に選択されます。
- ［**先頭または末尾で折り返す**］：検索の方向に応じて，検索が関数ライブラリの先頭または末尾に達したときに，それらの先頭または末尾から検索を続けます。
- ［**選択範囲のみに制限する**］：関数ライブラリの中で選択されているテキストの範囲内に限定して検索を行います。
- ［**末尾にカーソルを置く**］：検索対象文字列が見つかったときに，その文字列を強調表示して，文字列の末尾にカーソルを移動します。
- ［**検索する方向**］。検索する方向を指定します。
  - ［**上へ**］。現在のテキストから関数ライブラリの先頭までに限定して検索を行います。
  - ［**下へ**］。現在のテキストから関数ライブラリの末尾までに限定して検索を行います。

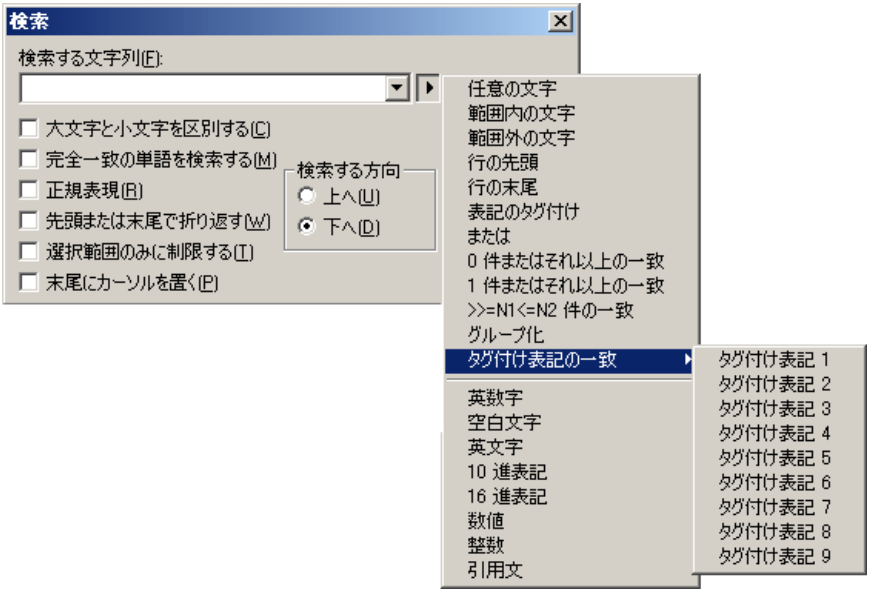
6 アクティブな関数ライブラリの中で，検索テキスト文字列の次の出現を強調表示するには，［**次を検索**］をクリックします。

7 強調表示されている文字列を［**置換後の文字列**］ボックスの文字列で置き換えるには［**置換**］をクリックします。アクティブな関数ライブラリにおいて，［**検索する文字列**］ボックスに指定した文字列のすべての出現を［**置換後の文字列**］ボックスの文字列で置き換えるには，［**すべて置換**］をクリックします。

［検索］および［置換］ダイアログ・ボックスにおける正規表現の使用

［**検索する文字列**］および［**置換後の文字列**］の文字列の中で正規表現を使用して検索を拡張できます。正規表現の概要については，617 ページ「正規表現の概要と使い方」を参照してください。［検索］および［置換］ダイアログ・ボックスで使用できる表現と QuickTest のほかの場所で使用できる表現に違いがあることに注意してください。

選択できる正規表現を表示するには，［検索］または［置換］ダイアログ・ボックスの中で矢印ボタン▶をクリックします。



定義済みの正規表現のリストから選択できます。また，タグ付きの表現も使用できます。文字列の検索に正規表現を使用するとき，見つかった文字列に応じて表現を変えたい場合があります。

たとえば，**(save¥:n)¥1** という正規表現を指定して検索をすると，**save** という文字列の後に任意の数字があり，その直後に **save** と，先に見つかった任意の数字があるという文字列が見つかります（つまり，たとえば **save6save6** という文字列が一致し，**save6save7** という文字列は一致しません）。

また，タグ付き表現を使用して，見つかった文字列の一部を置換文字列に取り込むことができます。たとえば，**save(¥:n)** を検索して，**open¥1** で置換するこ

とができます。この場合，**save** の後に任の数字がある文字列が検索され，それが **open** と，見つかった数字で置き換えられます。

検索文字列の中でタグ付き表現を示すための括弧「**()**」を挿入するには，正規表現のリストから［**表記のタグ付け**］を選択します。

使用するタグ表現を，「**¥**」に 1 から 9 のいずれかのタグ・グループ番号が続いた形式で指定するには，［**タグ付け表記の一致**］を選択してから対応するタグ・グループ番号を選択します（タグ付き表現の番号を調べるには，検索文字列の開き括弧「**(**」を数えます。最初の（最も左側にある）タグ付き表現は「**¥1**」で最後のは「**¥9**」です）。

VBScript の基本的な構文の理解

強力なスクリプト言語である VBScript を使用して，関数ライブラリを記述します。

この項では，VBScript ステートメントを使用して QuickTest 関数ライブラリを拡張するための基本的なガイドラインをいくつか示します。VBScript の使用方法の詳細については，QuickTest の［**ヘルプ**］メニュー（［**ヘルプ**］＞［**QuickTest Professional ヘルプ**］＞［**VBScript Reference**］）から VBScript のドキュメントを参照してください。

VBScript の各ステートメントには個別の構文規則があります。これらの規則に従わないと，問題のあるステップを実行したときにエラーが生成されます。関数ライブラリ・スクリプトの構文は，［**構文チェック**］ボタンをクリックするか，［**ツール**］＞［**構文チェック**］を選択することにより，いつでも確認できます。

関数ライブラリで作業をするときは，次に示す VBScript の一般構文規則とガイドラインに留意してください。

- **大文字と小文字の区別**：標準では VBScript は変数，オブジェクト，操作の名前，および定数などの大文字と小文字を区別しません。

  たとえば，次の 2 つのステートメントは VBScript においては同じです。

  ```
  Browser("Mercury").Page("Find a Flight:").WebList("toDay").Select "31"
  browser("mercury").page("find a flight:").weblist("today").select "31"
  ```

- **テキスト文字列**：値をテキスト文字列として入力するとき，文字列の前後に引用符を追加しなければなりません。たとえば，先に示したスクリプト行では，Web サイト，Web ページ，およびエディット・ボックスの名前はすべて引用符で囲まれたテキスト文字列です。

  31 の値が引用符で囲まれているのは，それが数値ではなく，数字を表すテキスト文字列だからです。

  次の例では，プロパティ名（最初の引数）のみがテキスト文字列で，引用符で囲まれています。2 番目の引数（プロパティの値）は変数なので引用符で囲まれていません。3 番目の引数（タイムアウト値を指定）は数値なので，これも引用符が不要です。

  ```
  Browser("Mercury").Page("Find a Flight:").WaitProperty("items count",
      Total_Items, 2000)
  ```

- **変数**：文字列，整数，配列，オブジェクトを格納するための変数を指定できます。変数を使用することで，スクリプトが読みやすくなり柔軟性が高くなります。詳細については，345 ページ「変数の使用」を参照してください。

- **括弧**：望む結果を得てエラーを避けるには，ステートメントの中で括弧を正しく使用することが重要です。詳細については，346 ページ「括弧の使用」を参照してください。

- **インデント**：スクリプトをステートメントの論理構造およびネストに合わせてインデントをしたりインデントを解除したりできます。詳細については，347 ページ「VBScript テキストの書式設定」を参照してください。

- **コメント**：ステートメントにコメントを追加するには，単一引用符（'）を独立の行の先頭で使用するか，ステートメントの末尾で使用します。スクリプトをわかりやすくして保守しやすいように，可能な限りのコメントを追加することをお勧めします。詳細については，347 ページ「VBScript テキストの書式設定」および 363 ページ「コメントの挿入」を参照してください。

- **スペース**：スペースを追加することでスクリプトをわかりやすくすることができます。これらのスペースは VBScript によって無視されます。

  特定の VBScript ステートメントを使用して関数ライブラリを拡張する方法の詳細については，363 ページ「コメント，フロー制御，そのほかの VBScript ステートメントの使用」を参照してください。

変数の使用

関数ライブラリの中でテスト・オブジェクトや単純な値を格納するための変数を指定することができます。テスト・オブジェクトを変数に格納する場合，ほかのステートメントの中でオブジェクト階層全体を指定することの代わりに変数を使用できます。変数をこのように使用すれば，ステートメントが読みやすくなり，保守もしやすくなります。

オブジェクトを格納する変数を指定するには，**Set** ステートメントを次の構文で使用します。

Set ObjectVar = ObjectHierarchy

次の例では，**Set** ステートメントを使用して **UserEditBox** 変数に，**username** エディット・ボックスの **Browser** ＞ **Page** ＞ **WebEdit** オブジェクト階層全体を格納するよう指定しています。その後，**UserEditBox** 変数を対象に **Set** メソッドを使用して，**username** エディット・ボックスに John という値を入力しています。

```
Set UserEditBox = Browser("Mercury Tours").Page("Mercury Tours").
    WebEdit("username")
UserEditBox.Set "John"
```

注：単純な値（文字列や数字など）を格納する変数の指定には **Set** ステートメントを使用しないでください。

Dim ステートメントを使用して，文字列，整数，配列など，ほかの型の変数を宣言できます。このステートメントは必須ではありませんが，関数ライブラリの構造を強化するために使用できます。次の例では，**Dim** ステートメントを使用して actual_value 変数を宣言し，現在の関数ライブラリのほかのステートメントの中で使用できるようにしています。

```
Dim actual_value
    ' 実際のプロパティ値を取得する
    actual_value = obj.GetROProperty(PropertyName)
```

括弧の使用

VBScript でプログラミングをするとき，ステートメント内での括弧「()」の使用・不使用に関する規則に従うことが重要です。

値を返すメソッドを呼び出し，返された値を使用する場合には，メソッドの引数を括弧で囲む必要があります。

たとえば，値を変数に返す場合，メソッドを **If** ステートメントの中で使用する場合，あるいは，**Call** キーワードを使用して関数を呼び出す場合などにメソッド引数を括弧で囲みます。

ヒント：関数ライブラリの中でステップを実行しているときに **Expected end of statement** エラー・メッセージを受け取った場合には，ステップのメソッドの引数を括弧で囲む必要があるかもしれません。

次に，括弧を使用する場合の例および使用しない場合の例をいくつか示します。

次の例では，**ChildItem** メソッドが値を変数に返すため，メソッドを括弧で囲む必要があります。

```
Set WebEditObj = Browser("Mercury Tours").Page("Method of Payment").
    WebTable("FirstName").ChildItem (8, 2, "WebEdit", 0)
WebEditObj.Set "Example"
```

次の例では，**Call** を使用しているため，メソッドの引数を括弧で囲む必要があります。

```
Call MyFunction("Hello World")
...
...
```

次の例では，**If** ステートメントの中でメソッドを使用しているため，**WaitProperty** メソッドの引数を括弧で囲む必要があります。

```
If Browser("index").Page("index").Link("All kinds of").
    WaitProperty("attribute/readyState", "complete", 4) Then
    Browser("index").Page("index").Link("All kinds of").Click
End If
```

次の例では，**Click** メソッドが値を返さないため，メソッドの引数を括弧で囲む必要はありません。

```
Browser("Mercury Tours").Page("Method of Payment").WebTable("FirstName").
    Click 3,4
```

VBScript テキストの書式設定

関数ライブラリで作業をするときは，コメントやインデントについて VBScript の慣例に従うことが重要です。

コメントを使用してスクリプトのセクションを説明するようにします。これにより読みやすさが向上し，関数ライブラリの保守や更新が容易になります。詳細については，363 ページ「コメントの挿入」を参照してください。

インデントはステートメントの論理構造およびネストを反映するために使用します。

➤ **コメントの追加**：ステートメントにコメントを追加するには，単一引用符（'）を独立の行の先頭に追加するか，ステートメントの末尾に追加します。

ヒント：

➤ ステートメントをコメントにするには，ステートメントの任意の場所をクリックして，［**コメント ブロック**］ボタンをクリックします。

➤ 選択したテキストのブロックをコメントにするには，［**コメント ブロック**］ボタンをクリックするか，［**編集**］＞［**詳細設定**］＞［**コメント ブロック**］を選択します。ブロック内の各行の先頭には単一引用符が付きます。

➤ **コメントの削除**：ステートメントからコメントを削除するには，独立の行の先頭またはステートメントの末尾の単一引用符（'）を削除します。

ヒント：選択したテキストのブロックまたは行のコメントを解除するには，［**コメント ブロック解除**］ボタンをクリックするか，［**編集**］＞［**詳細設定**］＞［**コメント ブロック解除**］を選択します。

➤ **ステートメントのインデント**：ステートメントをインデントするには，テキストを選択して，［**編集**］＞［**詳細設定**］＞［**インデント**］を選択するか，TAB キーを押します。381 ページ「エディタの動作のカスタマイズ」で説明されているように，［エディタ オプション］ダイアログ・ボックスで選択されているタブ間隔に従ってテキストがインデントされます。

注：［エディタ オプション］ダイアログ・ボックスの［**タブ キーを押して選択されたテキストでインデントを行う**］チェック・ボックスが選択されている必要があります。選択されていない場合，TAB キーを押すと選択したテキストが削除されます。

➤ **ステートメントのインデント解除**：ステートメントのインデントを解除するには，［**編集**］＞［**詳細設定**］＞［**インデント解除**］を選択するか，ステートメントの先頭のスペースを削除します。

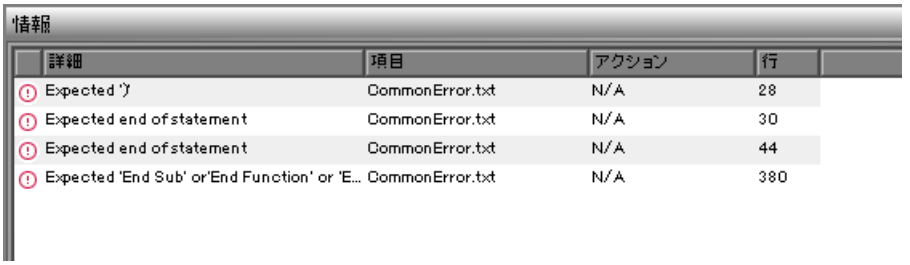
VBScript の書式設定の詳細については，QuickTest の［**ヘルプ**］メニュー（［**ヘルプ**］＞［**QuickTest Professional ヘルプ**］＞［**VBScript Reference**］）から VBScript のドキュメントを参照してください。

VBScript 構文エラーの処理方法

現在の関数ライブラリの構文は，［**構文チェック**］ボタンをクリックするか，［**ツール**］＞［**構文チェック**］を選択することで，いつでもチェックできます。エラーが検出されると，情報表示枠に表示されます。

［VBScript Reference］には個々の VBScript エラーの説明が表示されます。詳細については，［**ヘルプ**］＞［**QuickTest Professional ヘルプ**］＞［**VBScript Reference**］＞［**VBScript**］＞［**リファレンス**］＞［**エラー**］＞［**VBScript 構文エラー**］を選択してください。

「情報」表示枠に，ドキュメントの中で見つかった構文エラーの一覧が表示されるので，各構文エラーの場所を調べて修正することができます。



［情報］表示枠には，各構文エラーについて次の情報が表示されます。

表示枠内の要素	説明
［詳細］	構文エラーの詳細。たとえば，条件ブロックを **If** ステートメントで開始したけれども **End If** ステートメントで終了しなかった場合，［詳細］には Expected 'End If' と表示されます。 **注**：状況によっては，QuickTest がエラーを正確に特定できずに，次のようにいくつかの候補を表示することがあります。 Expected 'End Sub', or 'End Function', or 'End Property' この場合，示された行のステートメントを調べてどちらが該当するかを明らかにします。
［項目］	問題のステートメントが含まれている関数ライブラリの名前。
［アクション］	このカラムは，アプリケーション領域を通じてビジネス・コンポーネントに関連付けられている関数ライブラリには適用されません。
［行］	構文エラーが含まれている行。行の番号は各関数ライブラリの先頭から数えられます。

［情報］表示枠の使用

- 構文エラーの詳細の上にポインタを移動すると，現在不正である構文が表示されます。
- 特定の構文エラーを含む行に移動するには，［情報］表示枠内で構文エラーをダブルクリックします。

- ［情報］表示枠内のカラムのカラム・ヘッダをドラッグしてサイズを変更し，情報を読みやすくすることができます。
- ［情報］表示枠内の詳細情報を昇順または降順でソートするには，カラム・ヘッダをクリックします。
- ［情報］表示枠内のエラーで F1 キーを押すと，VBScript 構文エラーに関する情報が表示されます。

プログラム的記述の使用

QuickTest は，アプリケーション内のオブジェクトを学習すると，適切なテスト・オブジェクトをオブジェクト・リポジトリに追加します。オブジェクトがオブジェクト・リポジトリに追加されたら，エキスパート・ビューでステートメントを追加することで，そのオブジェクトに対して追加の操作を実行できます。ステートメントを追加するには，通常，各オブジェクトの名前（大文字小文字は区別されない）をそのオブジェクトの階層にオブジェクト記述として入力した後で，適切な操作を追加します。

たとえば，次に示すステートメントでは，「username」はエディット・ボックスの名前です。このエディット・ボックスは「Mercury Tours」という名前を持つページ上にあり，このページは「Mercury Tours」という名前を持つブラウザにあるものです。

```
Browser("Mercury Tours").Page("Mercury Tours").WebEdit("username")
```

オブジェクト・リポジトリ内の各オブジェクトは一意の名前を持っているので，指定する必要があるのはこの名前だけです。実行セッションの実行中，QuickTest はオブジェクト・リポジトリの中で名前と親オブジェクトに基づいてオブジェクトを検索し，格納されているテスト・オブジェクトの記述を使って，アプリケーション内のオブジェクトを識別します。

QuickTest に対して，オブジェクト・リポジトリまたはオブジェクト名を参照せずに，オブジェクトに対する操作を実行するように指示できます。これを行うためには，QuickTest に，操作の実行対象としたいオブジェクトを識別するために使えるプロパティと値のリストを提供します。

そのような**プログラム的記述**は，オブジェクト・リポジトリに格納されていないオブジェクトに対する操作を行う場合に，非常に便利なことがあります。プログラム的記述は，何らかの共通するプロパティを持つ複数のオブジェクトを

対象に同じ操作を行う場合や，実行セッション中に動的に決まる記述に適合するプロパティを持つ 1 つのオブジェクトに対する操作を行う場合にも使えます。

［テスト結果］では，実行セッション中にプログラム的記述または **ChildObjects** メソッドを使ってテスト・オブジェクトが動的に作成されたことを示す大括弧がテスト・オブジェクト名に付けられます。

たとえば，入力した人名情報に基づいて，雇用主のリストを表示し，リストから選択した雇用主に履歴書を送れるようにする Web サイトのテストをするものとしましょう。テストでは，リストに表示されたすべての雇用主を選択したいのに，テストを設計するときには，ページにいくつのチェック・ボックスが表示されるかわからず，もちろん各チェック・ボックスの正確なオブジェクト記述も知ることができません。こうした状況で，プログラム的記述を使うことで Set "ON" メソッドを，「HTML TAG = input，TYPE = check box」という記述に適合するすべてのオブジェクトを対象に実行することができます。

プログラム的記述には 2 つのタイプがあります。

- **静的**：VBScript ステートメントの中に，オブジェクトを記述するプロパティと値のセットを直接指定します。
- **動的**：プロパティと値のコレクションを記述オブジェクトに追加してから，ステートメントにその記述オブジェクトの名前を入力します。

  オブジェクト記述に対する要求が基本的なものであれば，**静的**記述を使ってステートメントにプログラム的記述を直接入力するほうが簡単でしょう。しかし，ほとんどの場合は**動的**記述を使ったほうが機能，効率，および柔軟性が向上します。

ステートメントへのプログラム的記述の直接入力

テスト・ステートメントにオブジェクトを直接記述するには，オブジェクトの名前を指定する代わりに，オブジェクトを記述する **property:=value** のペアを指定します。

一般的な構文は次のとおりです。

```
TestObject("PropertyName1:=PropertyValue1", "..." ,
    "PropertyNameX:=PropertyValueX")
```

TestObjec：テスト・オブジェクト・クラスです。

PropertyName:=PropertyValue：認識プロパティとその値です。
property:=value の各ペアは，カンマと二重引用符で区切る必要があります。

実行セッション中に取得するプロパティ値に基づいてオブジェクトを検索する場合には，プロパティ値として変数名を入力できます。次に例を示します。

```
MyVar="some text string"
Browser("Hello").Page("Hello").Webtable("table").Webedit("name:=" & MyVar)
```

注：QuickTest はプログラム的記述のプロパティ値をすべて正規表現として評価します。したがって，正規表現において特別な意味を持つ文字（*，?，+ など）を含んだ値を入力するには，¥（円記号）を使用して，その特殊文字をリテラルな文字として扱うように指示します。

次に示すステートメントでは，Mercury Tours ページに，author という名前と，3 というインデックスを持つ WebEdit テスト・オブジェクトを指定します。実行セッション中，QuickTest は一致するプロパティ値を持つ WebEdit オブジェクトを検索し，「Mark Twain」というテキストを入力します。

```
Browser("Mercury Tours").Page("Mercury Tours").WebEdit("Name:=Author",
    "Index:=3").Set "Mark Twain"
```

注：テスト・オブジェクト階層の特定のポイントからプログラム的記述を使用する場合には，同じステートメント内では，そのポイント以降は必ずプログラム的記述を使用する必要があります。プログラム的記述を使用して階層内のオブジェクトを指定した後に，オブジェクト・リポジトリでの名前を使用してテスト・オブジェクトを指定すると，当該オブジェクトは QuickTest によって識別されません。

たとえば，次の例ではテスト・オブジェクト階層全体を通してプログラム的記述を使用しているので，このステートメントは使用できます。

```
Browser("Title:=Mercury Tours").Page("Title:=Mercury Tours").
    WebEdit("Name:=Author", "Index:=3").Set "Mark Twain"
```

次の例も，特定のポイントから（Page オブジェクト記述以降）プログラム的記述を使用しているので，ステートメントを使用できます。

```
Browser("Mercury Tours").Page("Title:=Mercury Tours").
    WebEdit("Name:=Author", "Index:=3").Set "Mark Twain"
```

しかし，次の例では，Browser および Page オブジェクトについてはプログラム的記述を使用しているものの，WebEdit テスト・オブジェクトについてはオブジェクト・リポジトリでの名前を使おうとしているので，このステートメントは使用できません。

```
Browser("Title:=Mercury Tours").Page("Title:=Mercury Tours").
    WebEdit("Author").Set "Mark Twain"
```

QuickTest は，WebEdit オブジェクトをその名前を使用して特定しようとしますが，親オブジェクトがプログラム的記述を使用して指定されているため，リポジトリの中で当該オブジェクトを見つけることができません。

テスト・オブジェクトの使用方法の詳細については，第 5 章「オブジェクト・リポジトリ内のテスト・オブジェクトの管理」を参照してください。

1 つの関数ライブラリで同じプログラム的記述を複数回使用するために，作成したオブジェクトを変数に割り当てることができます。

たとえは，次のように入力する代わりに，

```
Window("Text:=Myfile.txt - Notepad").Move 50, 50
Window("Text:=Myfile.txt - Notepad").WinEdit("AttachedText:=Find what:").
    Set "hello"
Window("Text:=Myfile.txt - Notepad").WinButton("Caption:=Find next").Click
```

次のように入力できます。

```
Set MyWin = Window("Text:=Myfile.txt - Notepad")
MyWin.Move 50, 50
MyWin.WinEdit("AttachedText:=Find what:").Set "hello"
MyWin.WinButton("Caption:=Find next").Click
```

プログラム的記述のための Description オブジェクトの使用

Description オブジェクトを使用して，**Property** オブジェクトのセットを格納した **Properties** コレクションを返すことができます。**Property** オブジェクトは，プロパティ名とプロパティ値で構成されます。返された **Properties** コレクションを，ステートメント内でオブジェクト名の代わりに使用できます（各プロパティ・オブジェクトには，プロパティ名とプロパティ値のペアが含まれています)。

注：標準設定では，**Properties** コレクションに追加された **Property** オブジェクトの値はすべて正規表現として処理されます。したがって，正規表現において特別な意味を持つ文字（*，?，+ など）を含んだ値を入力するには，¥（円記号）を使用して，その特殊文字をリテラルな文字として扱うように指示します。

コレクション内の特定の **Property** オブジェクトの値をリテラル値として指定するには，**RegularExpression** プロパティを False に設定します。詳細については，『**HP QuickTest Professional Object Model Reference**』（英語版）の **Utility** の項を参照してください。

Properties コレクションを作成するには，次の構文を使って **Description.Create** ステートメントを入力します。

```
Set MyDescription = Description.Create()
```

Properties オブジェクト（たとえば上の例の MyDescription）を作成すれば，実行セッション中に **Properties** オブジェクトのプロパティと値の追加，編集，削除，取得を行うステートメントを入力できます。これにより，オブジェクト記述にどのプロパティをいくつ含めるかを，実行セッション中に動的に決めることができます。

Properties コレクションに Property オブジェクト（プロパティと値）のセットを設定したら，テスト・ステートメントの中で，この **Properties** オブジェクトをオブジェクト名の代わりに指定できます。

たとえは，次のように入力する代わりに，

```
Window("Error").WinButton("text:=OK", "width:=50").Click
```

次のように入力できます。

```
Set MyDescription = Description.Create()MyDescription("text").Value =
"OK"MyDescription("width").Value =
50Window("Error").WinButton(MyDescription).Click
```

注：テスト・オブジェクト階層の特定のポイントからプログラム的記述を使用する場合には，同じステートメント内では，そのポイント以降は必ずプログラム的記述を使用する必要があります。プログラム的記述を使用して階層内のオブジェクトを記述した後に，オブジェクト・リポジトリでの名前を使用してテスト・オブジェクトを指定すると，当該オブジェクトは QuickTest によって識別されません。

たとえば，Browser(Desc1).Page(Desc1).Link(desc3) ではテスト・オブジェクト階層全体を通してプログラム的記述を使用しているので，このステートメントは使用できます。

Browser("Index").Page(Desc1).Link(desc3) も，特定のポイントから（Page オブジェクト記述以降）プログラム的記述を使用しているので，このステートメントは使用できます。

しかし，Browser(Desc1).Page(Desc1).Link("Example1") の場合，Browser および Page オブジェクトにプログラム的記述を使用する一方で，Link テスト・オブジェクトにオブジェクト・リポジトリでの名前を使用しているので，このステートメントは使用できません（QuickTest によって，オブジェクト名に基づいて Link オブジェクトが検索されますが，プログラム的記述を使用して親オブジェクトを指定しているため，リポジトリの中でオブジェクトが見つかりません）。

Properties オブジェクトを使っているときには，プロパティや値の代わりに変数名を使うことで，実行セッション中に取得したプロパティや値に基づくオブジェクト記述を生成できます。

複数のオブジェクトでプログラム的記述を使いたい場合には，テスト内に複数の **Properties** オブジェクトを作成することもできます。

Description および **Properties** オブジェクト，および関連するメソッドの詳細については，『**HP QuickTest Professional Object Model Reference**』（英語版）を参照してください。

子オブジェクトの検索

ChildObjects メソッドを使って，指定された親オブジェクトの中にあるすべてのオブジェクト，あるいは特定のプログラム的記述に適合する子オブジェクトのみを取得できます。この子オブジェクトのサブセットを取得するには，**Description** オブジェクトを使って，まず記述オブジェクトを作成してから，子オブジェクト・コレクションに適合するプロパティと値の集合を追加します。

注：ChildObjects 記述引数のためのプログラム的記述を作成するには，**Description** オブジェクトを使用します。**property:=value** 構文を使ってプログラム的記述を引数に直接入力することはできません。

記述オブジェクトの中に記述を「構築」したら，次の構文を使ってその記述に適合する子オブジェクトを取得します。

Set MySubSet=TestObject.ChildObjects(MyDescription)

たとえば，次のステートメントは QuickTest に対して，Itinerary Web ページ上のすべてのチェック・ボックスを選択するよう指示します。

```
Set MyDescription = Description.Create()
MyDescription("html tag").Value = "INPUT"
MyDescription("type").Value = "checkbox"

Set Checkboxes =
Browser("Itinerary").Page("Itinerary").ChildObjects(MyDescription)
NoOfChildObjs = Checkboxes.Count
For Counter=0 to NoOfChildObjs-1
    Checkboxes(Counter).Set "ON"
Next
```

［テスト結果］では，実行セッション中に **ChildObjects** メソッドまたはプログラム的記述を使ってテスト・オブジェクトが動的に作成されたことを示す大括弧がテスト・オブジェクト名に付けられます。

ChildObjects メソッドの詳細については，『**HP QuickTest Professional Object Model Reference**』（英語版）を参照してください。

プログラム的記述での Index プロパティの使用

Index プロパティは，オブジェクトを一意に識別するための認識プロパティとして役立つ場合があります。**Index** 認識プロパティを使うと，オブジェクトはソース・コード内に出現する順序（最初の出現は 0）に基づいて識別されます。

Index プロパティ値は，各オブジェクトに固有の値です。つまり，インデックス値 3 を使って WebEdit テスト・オブジェクトを記述すると，QuickTest によってページ内の 4 番目の WebEdit オブジェクトが検索されます。

これに対し，インデックス値 3 を使って WebElement オブジェクトを記述すると，QuickTest は，タイプには関係なく，そのページの 4 番目の Web オブジェクトが検索されます。これは，WebElement オブジェクトがすべての Web オブジェクトに適用されるためです。

たとえば，次のオブジェクトが含まれるページがあるとします。

- Apple という名前の画像
- UserName という名前の画像
- UserName という名前の WebEdit オブジェクト
- Password という名前の画像
- Password という名前の WebEdit オブジェクト

次の記述は，前述のリストの 3 番目の項目を表します。ページ内で UserName という名前を持つ最初の WebEdit オブジェクトだからです。

```
WebEdit("Name:=UserName", "Index:=0")
```

一方，次の記述は，前述のリストの 2 番目の項目を表します。ページ内で UserName という名前を持つ最初の任意のタイプ（WebElement）のオブジェクトだからです。

```
WebElement("Name:=UserName", "Index:=0")
```

注：オブジェクトが 1 つのみの場合，index=0 を指定しても取得されません。この場合，**Index** プロパティをオブジェクト記述に含めるべきではありません。

プログラム的記述によるチェックの実行

プログラム的記述またはユーザ定義関数を使って，指定したオブジェクト・プロパティの実行時の値とそのプロパティの期待値を比較できます。

プログラム的記述によるチェックは，通常のチェックポイントを適用できない場合（プロパティをチェックするオブジェクトがオブジェクト・リポジトリに格納されていない場合など）に便利です。チェックの結果は，テスト結果レポートに書き込むことができます。

たとえば，Web ボタンの実行時の値をチェックするとします。**GetROProperty** または **Exist** 操作を使って，オブジェクトの実行時の値を取得したり，実行セッションのその時点でオブジェクトが存在するかどうかを確認したりできます。

以下の例は，プログラム的記述を使って **Continue** Web ボタンが実行セッション中に無効になっているかどうかをチェックする方法を示します。

GetROProperty 操作を使用する方法

```
ActualDisabledVal =
Browser(micClass:="Browser").Page(micClass:="Page").WebButton
    (alt:="Continue").GetROProperty("disabled")
```

Exist 操作を使用する方法

```
While Not Browser(micClass:="Browser").Page(micClass:="Page").WebButton
    (alt:="Continue").Exist(30)
Wend
```

Report.ReportEvent ステートメントを追加することにより，チェック結果をテスト結果に送信するように QuickTest に指示できます。

```
If ActualDisabledVal = True Then
Reporter.ReportEvent micPass, "CheckContinueButton = PASS", "The
Continue
    button is disabled, as expected."
Else
Reporter.ReportEvent micFail, "CheckContinueButton = FAIL", "The Continue
    button is enabled, even though it should be disabled."
```

また，アプリケーションが期待どおりに機能しているかどうかをチェックするユーザ定義関数を作成して使用することもできます。次の例は，オブジェクト

が無効になっているかどうかをチェックし，オブジェクトが無効になっている場合は **True** を返す関数を示します。

```
'@Description Checks whether the specified test object is disabled
'@Documentation Check whether the <Test object name> <test object
type> is enabled.
Public Function VerifyDisabled (obj)
    Dim enable_property
    ' テスト・オブジェクトから disabled プロパティを取得する
    enable_property = obj.GetROProperty("disabled")
    If enable_property = 1 Then ' 値が True（1）ならオブジェクトは無効
        Reporter.ReportEvent micPass, "VerifyDisabled Succeeded", "The test
object is disabled, as expected."
        VerifyDisabled = True
    Else
        Reporter.ReportEvent micFail, "VerifyDisabled Failed", "The test object is
enabled, although it should be disabled."
        VerifyDisabled = False
    End If
End Function
```

注：GetROProperty 操作の使用法の詳細については，372 ページ「ネイティブ・プロパティの取得」を参照してください。**While...Wend** ステートメントの使用法の詳細については，368 ページ「While...Wend ステートメント」を参照してください。特定のテスト・オブジェクト，操作，およびプロパティの詳細については，『**HP QuickTest Professional Object Model Reference**』（英語版）を参照してください。

プログラムによるアプリケーションの実行と終了

指定した場所から任意のアプリケーションを実行するには，関数ライブラリ内で **SystemUtil.Run** ステートメントを使います。これはコンポーネント内からアプリケーションを開く操作（関数）を提供したい場合，特に便利です。アプリケーションを指定して，サポートされている任意のパラメータを渡したり，ファイル名を指定して，関連付けられているアプリケーションが起動しそのファイルを開くようにできます。

ほとんどのアプリケーションは，**Close** メソッドを使って閉じることができます。また，**SystemUtil** ステートメントを使ってアプリケーションを閉じることもできます。詳細については，『**HP QuickTest Professional Object Model Reference**』（英語版）を参照してください。

たとえば，次に示すステートメントでは，**type.txt** というファイルを標準のテキスト編集アプリケーション（「メモ帳」など）で開き，happy days と入力してから，ショートカット・キーを使ってファイルを保存し，アプリケーションを終了しています。

```
SystemUtil.Run "C:¥type.txt", "","",""
Window("Text:=type.txt - Notepad").Type "happy days"
Window("Text:=type.txt - Notepad").Type micAltDwn & "F" & micAltUp
Window("Text:=type.txt - Notepad").Type micLShiftDwn & "S" & micLShiftUp
Window("Text:=type.txt - Notepad").Close
```

詳細については，『**HP QuickTest Professional Object Model Reference**』（英語版）を参照してください。

コメント，フロー制御，そのほかの VBScript ステートメントの使用

QuickTest では，関数ライブラリの論理フローを制御する条件文を追加することで，関数ライブラリに意思決定機能を組み込むことができます。さらに，QuickTest からテスト結果に送信するメッセージをテスト内に定義できます。関数ライブラリの読みやすさを向上させるために，コメントを追加することもできます。

注：「**VBScript Reference**」（［**ヘルプ**］＞［**QuickTest Professional ヘルプ**］メニューから利用できます）には，VBScript，Script Runtime，および Windows Script Host などを含む Microsoft VBScript に関するドキュメントが含まれます。

コメントの挿入

コメントとは，アポストロフィ（'）の後に書かれる，テスト・スクリプトの行または行の一部です。関数ライブラリで関数を実行する場合，コメントは QuickTest によって処理されません。読みやすさを向上させ，関数ライブラリの更新を容易にするため，テスト・スクリプトのセクションを説明するコメントを使用することをお勧めします。

次の例では，コメントを使ってその下のステートメントの目的を説明しています。

```
" 文字列 "mercury" を "username" エディット・ボックスに設定する
Browser("Mercury Tours").Page("Mercury Tours").WebEdit("username").
    Set "mercury"
```

標準設定では，コメントは関数ライブラリに緑色で表示されます。コメントの表示は［エディタ オプション］ダイアログ・ボックスでカスタマイズできます。詳細については，384 ページ「エレメントの見映えのカスタマイズ」を参照してください。

ヒント：

➤ テキストのブロックをコメントにするには，［**編集**］＞［**詳細設定**］＞［**コメント ブロック**］を選択するか，［**コメント ブロック**］ボタンをクリックします。

➤ コメントを解除するには，［**編集**］＞［**詳細設定**］＞［**コメント ブロック解除**］を選択するか，［**コメント ブロック解除**］ボタンをクリックします。

注：コメント行を追加するには，VBScript の **Rem** ステートメントを使用する方法もあります。詳細については，『**Microsoft VBScript Language Reference**』（英語版）（［**ヘルプ**］＞［**QuickTest Professional ヘルプ**］＞［**VBScript Reference**］＞［**VBScript**］を選択）を参照してください。

計算の実行

数値演算子を使って，簡単な計算を実行するステートメントを作成できます。たとえば，Web サイトで 2 つのテキスト・ボックスに表示された値を掛け算するには，乗法演算子を使用します。VBScript では，次の数値演算子が使用できます。

演算子	説明
+	加法
−	減法
−	否定（負の数値）
*	乗法

演算子	説明
/	除法
^	指数

次の例では，乗客 1 人あたりの荷物の重さの上限が 100 ポンドの場合の荷物の総重量を計算するために，乗法演算子を使用しています。

```
'GetROProperty メソッドを使って，エディット・ボックスから乗客数を取得する
passenger = Browser ("Mercury_Tours").Page ("Find_Flights").
    WebEdit("numPassengers").GetROProperty("value")
' 乗客数に 100 を乗じる
weight = passenger * 100
' メッセージ・ボックスに重量の上限を挿入する
msgbox(" この団体の荷物重量の上限は "& weight &" ポンドです。)
```

For...Next ステートメント

For...Next ループは，1 つ以上のステートメントを指定した回数だけ実行するように QuickTest に指示します。この関数の構文は次のとおりです。

For ***counter*** **=** ***start*** **to** ***end*** **[Step** ***step*****]**
    statement
Next

項目	説明
counter	反復の回数を表すカウンタとして使用する変数
start	カウンタの開始値
end	カウンタの終了値
step	各ループの終わりに増分する値。**標準設定値 = 1**。**省略可能**
statement	ループ中に実行する 1 つ以上のステートメント

次の例では，QuickTest で **For** ステートメントを使って乗客数の階乗値を計算しています。

```
passengers = Browser("Mercury Tours").Page("Find Flights").
    WebEdit("numPassengers").GetROProperty("value")
total = 1
For i=1 To passengers
    total = total * i
Next
MsgBox "!"& passengers & "=" & total
```

For...Each ステートメント

For...Each ループは，配列またはオブジェクト・コレクションの各要素に対して 1 つ以上のステートメントを実行するように QuickTest に指示します。この関数の構文は次のとおりです。

For Each item In *array*
    statement
Next

項目	説明
item	配列の要素を表す変数
array	配列の名前
statement	ループ中に実行する 1 つ以上のステートメント

次の例では，**For...Each** ループを使用して配列の各要素の値を表示しています。

```
MyArray = Array("one","two","three","four","five")
For Each element In MyArray
    msgbox element
Next
```

Do...Loop ステートメント

Do...Loop ステートメントは，条件が真である間，または条件が真になるまで，1 つ以上のステートメントを実行するように QuickTest に指示します。この関数の構文は次のとおりです。

Do [{while} {until} ***condition*****]**
    statement
Loop

項目	説明
condition	満たされるべき条件
statement	ループ中に実行する 1 つ以上のステートメント

次の例では，QuickTest で **Do...Loop** ステートメントを使って乗客数の階乗値を計算しています。

```
passengers = Browser("Mercury Tours").Page("Find Flights").
    WebEdit("numPassengers").GetROProperty("value")
total = 1
i = 1
Do while i <= passengers
     total = total * i
     i = i + 1
Loop
MsgBox "!"& passengers & "=" & total
```

While...Wend ステートメント

While...Wend ステートメントは，条件が真である間，1 つまたは複数のステートメントを実行するように QuickTest に指示します。この関数の構文は次のとおりです。

While *condition*
　statement
Wend

項目	説明
condition	満たされるべき条件
statement	ループ中に実行する 1 つ以上のステートメント

次の例では，QuickTest で **While** ステートメントを使って，乗客数が 10 人未満である間，ループを実行します。QuickTest によって，ループが 1 回実行されるたびに，乗客数が 1 ずつ増えます。

```
passengers = Browser("Mercury Tours").Page("Find Flights").
   WebEdit("numpassengers").GetROProperty("value")
While passengers < 10
   passengers = passengers + 1
Wend

msgbox(" この団体の人数は "& passengers & " 人です。
```

If...Then...Else ステートメント

If...Then...Else ステートメントは，特定の条件に基づいて 1 つまたは複数のステートメントを実行するように QuickTest に指示します。条件が満たされない場合は，次の **ElseIf** 条件または **Else** ステートメントが試されることになります。この関数の構文は次のとおりです。

```
If condition Then
    statement
ElseIf condition2 Then
    statement
Else
    statement
End If
```

項目	説明
condition	満たされるべき条件
statement	実行されるステートメント

次の例では，乗客数が 4 名未満の場合に，QuickTest によってブラウザが閉じられます。

```
passengers = Browser("Mercury Tours").Page("Find Flights").
    WebEdit("numpassengers").GetROProperty("value")
If (passengers < 4) Then
    Browser("Mercury Tours").Close
Else
    Browser("Mercury Tours").Page("Find Flights").Image("continue").Click 69,5
End If
```

次の例では，**If**，**ElseIf**，および **Else** ステートメントを使用して値が 1，2，またはそれ以外の値に等しいかどうかを調べています。

```
value = 2
If value = 1 Then
  msgbox "one"
ElseIf value = 2 Then
  msgbox "two"
Else
  msgbox "not one or two"
End If
```

認識プロパティ値の取得と設定

認識プロパティとは，各オブジェクトについて QuickTest によって定義されている一連のプロパティのことです。テスト・オブジェクトの認識プロパティ値の設定と取得ができます。また，認識プロパティの値を実行環境オブジェクトから取得することもできます。

関数を実行すると，QuickTest は，テスト・オブジェクト・リポジトリに格納されているテスト・オブジェクトの一時的なインスタンスを生成します。関数ライブラリ内の **GetTOProperty**，**GetTOProperties**，および **SetTOProperty** メソッドを使って，テスト・オブジェクトの認識プロパティ値の設定と取得ができます。

GetTOProperty および **GetTOProperties** メソッドを使って，QuickTest がオブジェクトの識別に使う，特定のプロパティ値またはすべてのプロパティと値を取得できます。

SetTOProperty メソッドを使って，QuickTest がオブジェクトを識別するために使うプロパティ値を変更できます。

注： QuickTest は実行セッション中にテスト・オブジェクトの一時的なインスタンスを参照するため，**SetTOProperty** メソッドを使用して行ったすべての変更は実行セッション中にのみ有効で，テスト・オブジェクト・リポジトリに格納されている値には影響を与えません。

たとえば，次に示すステートメントは，［**Submit**］ボタンの名前値を「my button」に設定し，次に値「my button」を取得して **ButtonName** 変数に代入しています。

```
Browser("QA Home Page").Page("QA Home Page").
    WebButton("Submit").SetTOProperty "Name", "my button"

ButtonName=Browser("QA Home Page").Page("QA Home Page").
    WebButton("Submit").GetTOProperty("Name")
```

認識プロパティの現在の値をアプリケーション内の実行環境オブジェクトから取得するには，**GetROProperty** メソッドを使います。

たとえば，次のようにして，実行セッション時にリンクのターゲット値を取得できます。

```
link_href = Browser("HP Technologies").Page("HP Technologies").
    Link("Jobs").GetROProperty("href")
```

ヒント：アプリケーション内にあるオブジェクトの認識プロパティが不明の場合は，オブジェクト・スパイを使うことでそれらを表示できます。オブジェクト・スパイの詳細については，第 3 章「テスト・オブジェクト・モデルについて」を参照してください。

各オブジェクトでサポートされている認識プロパティのリストと説明，および **GetROProperty**，**GetTOProperty**，**GetTOProperties**，**SetTOProperty** メソッドの詳細については，『**HP QuickTest Professional Object Model Reference**』（英語版）を参照してください。

ネイティブ・プロパティおよび操作へのアクセス

特定のテスト・オブジェクトで利用可能なテスト・オブジェクト操作または認識プロパティによって必要な機能が提供されていない場合，**Object** プロパティを使用することで，アプリケーションの任意の実行環境オブジェクトのネイティブの操作およびプロパティにアクセスできます。

ステートメント完了機能をオブジェクトのプロパティに対して使用すれば，オブジェクトに対して利用可能なネイティブの操作とプロパティのリストを表示できます。ステートメントの自動補完機能の詳細については，326 ページ「関数ライブラリでのステートメントの生成」を参照してください。

ヒント：オブジェクトが Web オブジェクトである場合は，プログラム的記述の中で「attribute/ ＜プロパティ＞」の形式を使用してネイティブのプロパティにもアクセスできます。詳細については，373 ページ「Web オブジェクトのユーザ定義プロパティへのアクセス」を参照してください。

ネイティブ・プロパティの取得

Object プロパティを使用することで，任意の実行環境オブジェクトのネイティブ・プロパティにアクセスできます。たとえば，ActiveX カレンダーの内部 **Day** プロパティについて現在の値を取得するには，次のようにします。

```
Dim MyDay
Set MyDay=
Browser("index").Page("Untitled").ActiveX("MSCAL.Calendar.7").Object.Day
```

Object プロパティの詳細については，『**HP QuickTest Professional Object Model Reference**』（英語版）を参照してください。

ネイティブの操作のアクティブ化

Object プロパティを使用することで，任意の実行環境オブジェクトの内部操作をアクティブ化できます。たとえば次のようにすることで，エディット・ボックスのネイティブ **focus** メソッドを呼び出せます。

```
Dim MyWebEdit
Set MyWebEdit=Browser("Mercury Tours").Page("Mercury Tours").
    WebEdit("username").Object
MyWebEdit.focus
```

Object プロパティの詳細については，『**HP QuickTest Professional Object Model Reference**』（英語版）を参照してください。

Web オブジェクトのユーザ定義プロパティへのアクセス

attribute/ ＜プロパティ名＞表記を使って，Web オブジェクトのネイティブ・プロパティにアクセスし，プログラム的記述を使ってこれらのプロパティに基づいてオブジェクトを識別することができます。

たとえば，ページ内の 2 か所に同じ企業ロゴ画像を持つ Web ページがあるとします。

```
<IMG src="logo.gif" LogoID="122">
<IMG src="logo.gif" LogoID="123">
```

ユーザ定義プロパティの LogoID を次のように記述に含めることで，クリックする画像をプログラム的記述を使って識別できます。

```
Browser("Mercury Tours").Page("Find Flights").Image("src:=logo.gif",
    "attribute/LogoID:=123").Click 68, 12
```

プログラム的記述の詳細については，350 ページ「プログラム的記述の使用」を参照してください。

DOS コマンドの実行

QuickTest 関数ライブラリの中で，VBScript Windows Scripting Host Shell オブジェクト（WScript.shell）を使って，標準の DOS コマンドを実行できます。たとえば，次のステートメントを使うことにより，DOS コマンド・ウィンドウを開いてパスを C:¥ に変更し，**DIR** コマンドを実行できます。

```
Dim oShell
Set oShell = CreateObject ("WScript.shell")
oShell.run "cmd /K CD C:¥ & Dir"
Set oShell = Nothing
```

詳細については，『**Microsoft VBScript Language Reference**』（英語版）（［**ヘルプ**］＞［**QuickTest Professional ヘルプ**］＞［**VBScript Reference**］＞［**VBScript**］を選択）を参照してください。

Windows API を使用したテストおよび関数ライブラリの拡張

Windows API を使用してテスト機能を拡張し，関数ライブラリの使い勝手と柔軟性を高めることができます。Windows オペレーティング・システムは，Windows での操作を制御，管理するために使用できる多数の関数を備えています。これらの関数を使用することで追加機能を利用できます。

Windows API は，次の URL から参照できる Microsoft MSDN Web サイトにドキュメント資料があります。http://msdn2.microsoft.com/en-us/library/Aa383750

個々の API 関数のリファレンスについては，次を参照してください。
http://msdn2.microsoft.com/en-us/library/Aa383749

Windows API 関数を使用するには，次の手順を実行します。

1 MSDN において，関数ライブラリの中で使用する関数を探します。

2 資料を読んで必要なパラメータと戻り値を把握します。

3 API 関数の場所を書き留めます。API 関数は Windows の DLL に含まれています。要求する関数が含まれている DLL の名前は通常，関数の説明の「Import Library」のセクションに記載してある名前と同じです。たとえば，説明ドキュメントの中で **User32.lib** と記載してあれば，関数は **User32.dll** という名前の DLL に含まれています。この DLL は通常，System32 ライブラリに含まれています。

4 QuickTest **Extern** オブジェクトを使用して外部関数を宣言します。詳細については，『**HP QuickTest Professional Object Model Reference**』（英語版）を参照してください。

次の例では，**user32.dll** にある **GetForegroundWindow** という関数への呼び出しを宣言しています。

```
extern.declare micHwnd, "GetForegroundWindow", "user32.dll",
"GetForegroundWindow"
```

5 hwnd = extern.GetForegroundWindows() のように，宣言した関数を呼び出し，必要な引数を渡します。

この例では，前面にあるウィンドウのハンドルが取得されます。このようにすることで，前面のウィンドウがオブジェクト・リポジトリにない場合や，あらかじめ知ることができない場合（たとえば，動的なタイトルを持つウィンドウなど）に関数ライブラリを拡張できます。このハンドルを，次のように，ウィンドウのプログラム的記述の一部として使用することも可能です。

```
Window("HWND:="&hWnd).Close
```

状況によっては，あらかじめ定義されている定数値を関数の引数として使用しなければならない場合があります。そうした定数は，関数内で定義されていないため，呼び出し先の関数に渡すためにはそれらの値を調べる必要があります。こうした定数の値は通常，使用する関数に対応するヘッダー・ファイルに宣言されています。ヘッダー・ファイルに関する記述も，各関数の説明ドキュメントの「Header」セクションに記載されています。使用しているコンピュータに Microsoft Visual Studio がインストールされていれば，ヘッダー・ファイルは通常，**X:¥Program Files¥Microsoft Visual Studio¥VC98¥Include** の下に格納されています。

たとえば，**GetWindow API** 関数では，指定されたウィンドウとハンドルを取得するウィンドウとの関係を表す数値を受け取ることが想定されています。MSDN の説明ドキュメントには次の定数が記載されています。GW_CHILD, GW_ENABLEDPOPUP, GW_HWNDFIRST, GW_HWNDLAST, GW_HWNDNEXT, GW_HWNDPREV and GW_HWNDPREV. **GetWindow** の説明ドキュメントに記載されている **WINUSER.H** ファイルを開くと，次のフラグ値が設定されているのがわかります。

```
/*
 * GetWindow() Constants
 */
#define GW_HWNDFIRST0
#define GW_HWNDLAST 1
#define GW_HWNDNEXT2
#define GW_HWNDPREV 3
#define GW_OWNER 4
#define GW_CHILD 5
#define GW_ENABLEDPOPUP 6
#define GW_MAX 6
```

例

次の例では,「メモ帳」アプリケーションの特定のメニュー項目を取得しています。

```
' 定数値
const MF_BYPOSITION = 1024
' API 関数の宣言
Extern.Declare micHwnd,"GetMenu","user32.dll","GetMenu",micHwnd
Extern.Declare
micInteger,"GetMenuItemCount","user32.dll","GetMenuItemCount",micHwnd
Extern.Declare
micHwnd,"GetSubMenu","user32.dll","GetSubMenu",micHwnd,micInteger
Extern.Declare
micInteger,"GetMenuString","user32.dll","GetMenuString",micHwnd,micInteger,
    micString+micByRef,micInteger,micInteger
' Notepad.exe
hwin = Window("Notepad").GetROProperty ("hwnd")' ウィンドウのハンドルを
取得
MsgBox hwin
' API 関数を使用する
men_hwnd = Extern.GetMenu(hwin)' ウィンドウのメイン・メニューのハンドル
を取得
MsgBox men_hwnd
item_cnt = Extern.GetMenuItemCount(men_hwnd)
MsgBox item_cnt
hSubm = Extern.GetSubMenu(men_hwnd,0)
MsgBox hSubm
rc = Extern.GetMenuString(hSubm,0,value,64 ,MF_BYPOSITION)
```

MsgBox value

実行セッション中に報告するステップの選択

Report.Filter メソッドを使って，テスト結果にどのステップあるいはどのステップのタイプを含めるかを決めることができます。ステートメントの後のステップの報告の有効化と完全な無効化を行うことができます。あるいは，以降の失敗したステップ，もしくは失敗および警告のステップだけをレポートに含めるように指定することができます。また，**Report.Filter** メソッドを使って，現在のレポート・モードを取得することもできます。

次のレポート・モードが使用できます。

モード	説明
0 または **rfEnableAll**	すべてのイベントがテスト結果に表示されます。**標準設定値**。
1 または **rfEnableErrorsAndWarnings**	ステータスが警告または失敗のイベントだけがテスト結果に表示されます。
2 または **rfEnableErrorsOnly**	ステータスが失敗のイベントだけがテスト結果に表示されます。
3 または **rfDisableAll**	テスト結果にはイベントは表示されません。

- 以降のステップの報告を行わないようにするには，次のステートメントを入力します。

  Reporter.Filter = rfDisableAll

- 以降のステップの報告を再び行うようにするには，次のように入力します。

  Reporter.Filter = rfEnableAll

- 以降の失敗したステップだけをテスト結果に含めるには，次のように入力します。

  Reporter.Filter = rfEnableErrorsOnly

➤ 以降の失敗または警告のステップだけをテスト結果に含めるには，次のように入力します。

```
Reporter.Filter = rfEnableErrorsAndWarnings
```

➤ 現在のレポート・モードを取得するには，次のように入力します。

```
MyVar=Reporter.Filter
```

詳細については，『**HP QuickTest Professional Object Model Reference**』（英語版）を参照してください。

第 11 章

関数ライブラリ・ウィンドウのカスタマイズ

関数ライブラリ・ウィンドウでの関数の表示方法をカスタマイズできます。変更したすべての内容は，すべての関数ライブラリ・ウィンドウにグローバルに適用されます。

本章の内容

- 関数ライブラリ・ウィンドウのカスタマイズについて（380 ページ）
- エディタの動作のカスタマイズ（381 ページ）
- エレメントの見映えのカスタマイズ（384 ページ）
- 編集コマンドのカスタマイズ（385 ページ）

関数ライブラリ・ウィンドウのカスタマイズについて

QuickTest には，強力でカスタマイズ可能なエディタが用意されています。これを使用して，関数ライブラリ・ウィンドウの各部を変更できます。

［エディタ オプション］ダイアログ・ボックスでは，関数ライブラリ・ウィンドウでの，関数ライブラリの表示方法を変更できます。また，関数ライブラリに表示されるテキストのフォントや文字の大きさを変更したり，コメント，文字列，QuickTest の予約語，演算子，数字などスクリプトのエレメントごとに色を変えたりできます。たとえば，すべての文字列を赤で表示することもできます。

QuickTest には，カーソルの移動，文字の削除，クリップボードを使った情報の切り取り，コピー，貼り付けを行うことができる，標準設定のキーボード・ショートカットのリストがあります。これらのショートカットは，任意のショートカットに置き換えることができます。たとえば，**Line start** コマンドを標準設定の HOME から ALT + HOME へ変更できます。

［印刷］ダイアログ・ボックスのオプションを使用して，関数ライブラリの印刷方法を変更することもできます。詳細については，399 ページ「関数ライブラリの印刷」を参照してください。関数ライブラリを使った作業の詳細については，第 12 章「ユーザ定義関数および関数ライブラリを使用した作業」を参照してください。

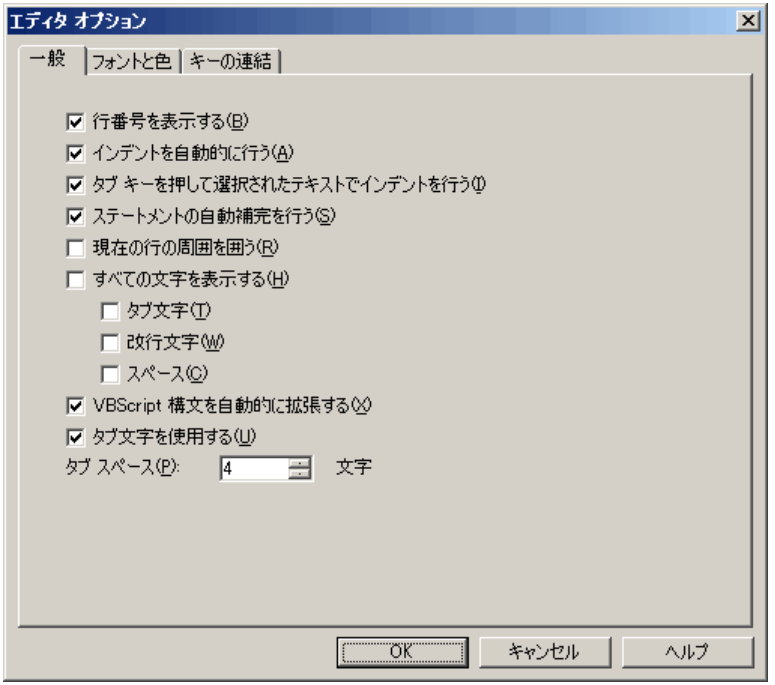
エディタの動作のカスタマイズ

関数ライブラリ・ウィンドウにおける 関数ライブラリの表示方法をカスタマイズできます。たとえば，文字記号を表示または非表示にしたり，行番号を表示するよう選択したりできます。たとえば，文字記号を表示または非表示にしたり，行番号を表示するよう選択したりできます。関数ライブラリを使った作業の詳細については，第 12 章「ユーザ定義関数および関数ライブラリを使用した作業」を参照してください。

エディタの動作をカスタマイズするには，次の手順を実行します。

1 関数ライブラリ・ウィンドウがアクティブなときに，［**ツール**］＞［**表示オプション**］を選択します。［エディタ オプション］ダイアログ・ボックスが開きます。

2 ［**一般**］タブをクリックします。



3 次のオプションから選択します。

オプション	説明
［行番号を表示する］	関数の各行の左に行番号が表示されます。
［インデントを自動的に行う］	インデントを設定した行の後に続く行が，自動的に前の行と同じ位置から開始されます。キーボードの HOME キーを押すと，カーソルが左マージンに戻ります。
［タブ キーを押して選択されたテキストでインデントを行う］	TAB キーを押すと，選択されたテキストがインデントされます。このオプションが有効でない場合は，Tab キーを押すと，選択されたテキストがタブ文字 1 つで置換されます。
［ステートメントの自動完了を行う］	このオプションを選択した場合，関数ライブラリ，IntelliSense（QuickTest に含まれているステートメントの自動補完機能）で入力すると，ステートメントに対する変数，メソッド，またはプロパティ をドロップダウン・リストから選択し，関連する構文を表示できます。 ステートメント補完（IntelliSense）機能の使用の詳細については，327 ページ「ステートメントの自動補完の使用（IntelliSense）」を参照してください。
［現在の行の周囲を囲う］	テスト内で現在カーソルがある行の周りに枠が表示されます。
［動的に囲む］	既存のコード行をブロック構造で囲み，ブロック・ステートメントを動的に展開（または折りたたみ）できるようにします。たとえば，既存のコードの前に取り囲むステートメント（if/while など）を追加する場合，矢印キーを使用してブロックを展開し，後続の行を含めることができます。その後，これらの行は正しいレベルに自動的にインデントされます。
［すべての文字を表示する］	すべてのタブ，改行，空白文字などの記号が表示されます。また，対応するチェック・ボックスを選択 / クリアすれば，これらの文字の一部だけを表示させることもできます。

オプション	説明

［VBScript 構文を自動的に拡張する］	キーワードの最初の 2 文字を自動的に認識し，該当するキーワードを入力するとそれに対応する VBScript 構文またはブロックをスクリプトに追加します。
	たとえば，エキスパート・ビューの行頭で「if」という文字に続いてスペースを入力すると，次の構文が自動的に入力されます。
	If Then
	End If

［タブ文字を使用する］	キーボードの TAB キーを使用したときに，タブ文字が挿入されます。このオプションを選択していなければ，TAB キーを使用したときに代わりに指定された数のスペース文字が挿入されます。

4 ［OK］をクリックし，変更を適用してダイアログ・ボックスを閉じます。

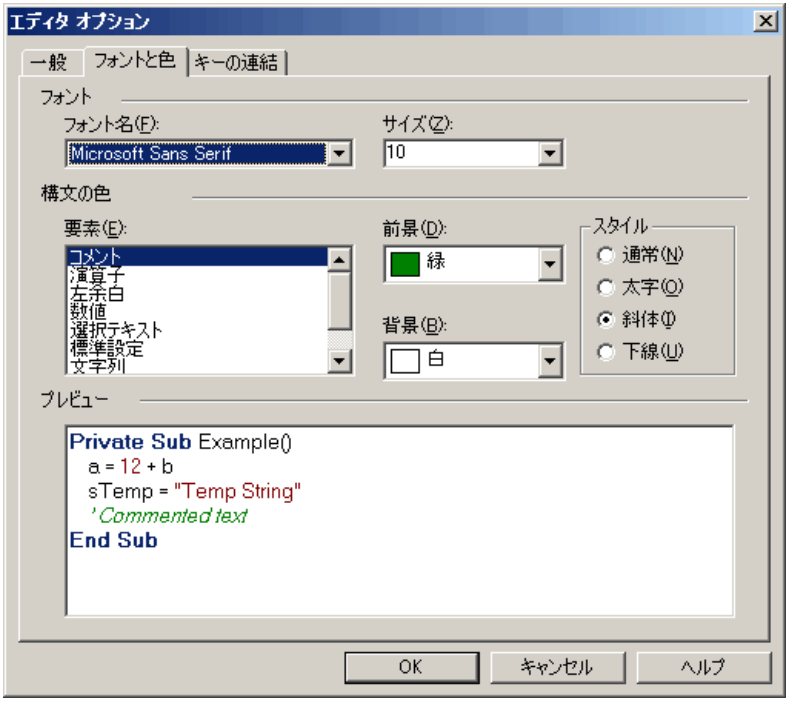
エレメントの見映えのカスタマイズ

QuickTest の関数ライブラリには，コメント，文字列，QuickTest や VBScript の予約語，演算子，数字など，さまざまなエレメントが含まれています。QuickTest の関数ライブラリのそれぞれのエレメントは，異なる色で表示できます。すべてのエレメントに使用されるフォントのスタイルや文字の大きさを指定することもできます。各エレメントに対して，独自にカスタマイズした配色を作成できます。たとえば，すべてのコメントを青の文字と黄色の背景で表示できます。

エレメントに対してフォントと色を設定するには，次の手順を実行します。

1 関数ライブラリ・ウィンドウがアクティブなときに，［**ツール**］＞［**表示オプション**］を選択します。［エディタ オプション］ダイアログ・ボックスが開きます。

2 ［**フォントと色**］タブをクリックします。



3［**フォント**］領域で，すべてのエレメントの表示に使用する［**フォント名**］と［**サイズ**］を選択します。このエディタでは Unicode フォントである Microsoft Sans Serif フォントが標準で使用されます。

注：Unicode 環境でテストを行う場合は，Unicode 対応のフォントを選択しなければなりません。Unicode 対応のフォントを選択しないと，関数ライブラリ内のエレメントが，関数ライブラリ・ウィンドウに正しく表示されない場合があります。ただし，関数ライブラリは選択したフォントに関係なく，指定前と変わりなく実行されます。Unicode 対応でない環境で作業をしている場合は，Courier などの固定幅フォントを使用して，文字の揃えがよくなるようにします。

4［**要素**］リストから，エレメントを選択します。

5 文字の色と背景の色を選択します。

6 エレメントのフォント・スタイルを選択します（［**通常**］，［**太字**］，［**斜体**］，［**下線**］）変更を適用した場合の例が，ダイアログ・ボックス下部の**プレビュー**表示枠に表示されます。

7 変更するエレメントごとに，手順 4 から 6 を繰り返します。

8［**OK**］をクリックし，変更を適用してダイアログ・ボックスを閉じます。

編集コマンドのカスタマイズ

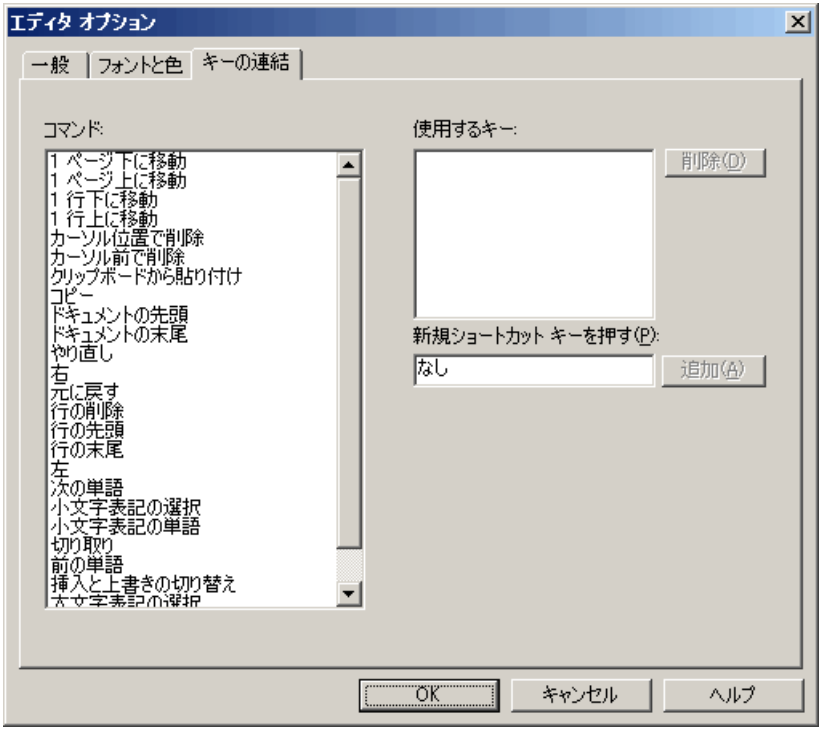
編集に使用する標準のキーボード・ショートカットをカスタマイズできます。QuickTest には，カーソルの移動，文字の削除，クリップボードを使った情報の切り取り，コピー，貼り付けを行うためのキーボード・ショートカットがあります。これらのショートカットは，任意のショートカットに置き換えることができます。たとえば，**Line end** コマンドを標準設定の END から ALT + END へ変更できます。

注：標準の QuickTest メニュー・ショートカット・キーは，ユーザが定義するキーの割り当てに優先します。たとえば，貼り付けコマンドのキーの組み合わせを CTRL+P と設定した場合でも，［印刷］ダイアログ・ボックスを開くための標準のショートカット・キー（［**ファイル**］＞［**印刷**］オプションに対応）が優先されます。QuickTest のメニュー・ショートカット・キーの一覧は，68 ページ「QuickTest コマンドの実行」を参照してください。

編集コマンドをカスタマイズするには，次の手順を実行します。

1 関数ライブラリ・ウィンドウがアクティブなときに，［**ツール**］＞［**表示オプション**］を選択します。［エディタ オプション］ダイアログ・ボックスが開きます。

2 ［**キーの連結**］タブをクリックします。



3［**コマンド**］リストからコマンドを選択します。

4［**新規ショートカット キーを押す**］ボックスをクリックして，選択されているコマンドに使用するキーを押します。たとえば，CTRL+4 を入力するには，CTRL キーを押しながら数字の 4 を押します。

5［**追加**］をクリックします。

注：指定したキーの組み合わせがサポートされていない場合，あるいは別のコマンドに対してすでに定義されている場合は，ショートカット・キー・ボックスの下にメッセージが表示されます。

---

6 ほかに追加するコマンドがあれば手順 3 から 5 を繰り返します。

7 リストからキー指定を削除するには，［**コマンド**］リスト内のコマンドを選択して［**使用するキー**］リスト内のキー（またはキーの組み合わせ）を強調表示し，［**削除**］をクリックします。

8［**OK**］をクリックし，変更を適用してダイアログ・ボックスを閉じます。

第 12 章

ユーザ定義関数および関数ライブラリを使用した作業

QuickTest テスト・オブジェクト・モデルでサポートされているテスト・オブジェクト，メソッド，および組み込み関数に加え，VBScript 関数，サブルーチン，モジュールなどが含まれる独自の関数ライブラリを定義して，その関数をコンポーネントで操作として使用できます。

注：本章では，「関数」，「メソッド」，および「操作」を同義語として使用しています。これは，ビジネス・コンポーネント・キーワード・ビューで「関数」と「メソッド」が「操作」として表示されるのに対し，QuickTest では「関数」と「メソッド」が使用されるためです。

本章の内容

- ユーザ定義関数および関数ライブラリの使い方について（390 ページ）
- 関数ライブラリの管理（391 ページ）
- 関連付けられている関数ライブラリを使用した作業（402 ページ）
- 関数定義ジェネレータの使用方法（405 ページ）
- ユーザ定義関数のテスト・オブジェクト・メソッドとしての登録（419 ページ）
- ユーザ定義関数の使い方のヒント（424 ページ）

ユーザ定義関数および関数ライブラリの使い方について

ユーザ定義関数を作成することにより，コンポーネントに機能を追加できます。ユーザ定義関数とは，何らかの処理（またはプログラミングが必要な一連のステップ）を 1 つのキーワード（または操作）にカプセル化したものです。ユーザ定義関数を使用することによって，コンポーネントの設計，理解，保守が容易になります。QuickTest エンジニアまたは各分野のエキスパートが関連するキーワード（または操作）をコンポーネントに挿入することで，そのコンポーネントからユーザ定義関数を呼び出すことができます。

ユーザ定義関数を，QuickTest テスト・オブジェクトのメソッドとして登録できます。登録したメソッドは，実行セッションの間だけ既存のテスト・オブジェクト・メソッドの機能をオーバーライドしたり，テスト・オブジェクト・クラスの新しいメソッドとして登録したりできます。ユーザ定義関数の登録の詳細については，405 ページ「関数定義ジェネレータの使用方法」および 419 ページ「ユーザ定義関数のテスト・オブジェクト・メソッドとしての登録」を参照してください。

注：ユーザ定義関数を作成するときは，組み込みの関数と同じ名前（たとえば，GetLastError，MsgBox，Print など）を指定しないようにします。組み込み関数は，ユーザ定義関数に優先します。したがって，組み込み関数と同じ名前のユーザ定義関数を呼び出しても，組み込み関数が代わりに呼び出されます。組み込み関数のリストについては，ステップ・ジェネレータ（[**挿入**] > [**ステップ ジェネレータ**]）の「**組み込み関数**」リストを参照してください。

QuickTest では，ユーザ定義関数を定義し，関数ライブラリ（標準設定では **.qfl** ファイルとして保存）に格納できます。関数ライブラリとは，VBscript 関数，サブルーチン，モジュールなどが含まれる Visual Basic スクリプトのことです。また，QuickTest では，既存の関数ライブラリ（**.vbs** ファイルまたは **.txt** ファイルなど）を変更，デバッグすることもできます。VBScript の使用の詳細については，348 ページ「VBScript 構文エラーの処理方法」を参照してください。

関数を関数ライブラリに格納し，その関数ライブラリをアプリケーション領域に関連付ければ，その関数ライブラリのパブリック関数をアプリケーション領域に関連付けられたすべてのコンポーネントで呼び出すことができます。詳細については，402 ページ「関連付けられている関数ライブラリを使用した作業」を参照してください。関連付けられている関数ライブラリに格納されている関

数は，ステップ・ジェネレータ（関数ライブラリの場合），およびキーワード・ビューの［**操作**］カラム，［使用可能なキーワード］表示枠からアクセスできます。

また，プライベート関数を定義して関数ライブラリに格納することができます。プライベート関数は，同じ関数ライブラリ内のほかの関数からのみ呼び出せる関数です。これは，パブリック関数の中でコード・セグメントを再利用する必要がある場合に便利です。

関数は，手作業で定義することも，関数定義ジェネレータを使って定義することもできます。関数定義ジェネレータは，関数の基本的な定義を自動的に作成します。関数を手作業で定義する場合でも，関数定義ジェネレータを使用すれば，ヘッダ情報の追加，テスト・オブジェクトへの関数の登録，テスト・オブジェクトの標準メソッドとしての関数の設定を行うために必要な構文を表示できます。詳細については，405 ページ「関数定義ジェネレータの使用方法」を参照してください。

関数ライブラリの管理

QuickTest で関数ライブラリを作成し，コンポーネントのアプリケーション領域に関数ライブラリを関連付けた後，その関数をコンポーネントから呼び出すことができます。関数ライブラリとは，VBscript 関数，サブルーチン，モジュールなどが含まれる独立した QuickTest ドキュメントのことです。各関数ライブラリは別々のウィンドウで開くため，同時に 1 つまたは複数の関数ライブラリを開いて作業できます。関数ライブラリの編集が終了したら，関数ライブラリは閉じて，QuickTest セッションは開いたままにしておくことができます。また，開いているすべての関数ライブラリを同時に閉じることもできます

ユーザ定義関数を関数ライブラリで実装し，その関数ライブラリをアプリケーション領域を通じてコンポーネントに関連付けることによって，各分野のエキスパートなどほかのユーザでも，プログラミングの知識がなくても，複雑な操作（コンポーネント・ステップへの if/then ステートメントとループの追加など）を実行する関数を選択できるようになります。さらに，再利用可能な関数の実装および使用により，時間とリソースを節約できます。

QuickTest には，あらゆる関数ライブラリを（外部エディタで作成された関数ライブラリでも）編集およびデバッグできるツールがあります。たとえば，QuickTest では関数の構文をチェックできます。関数ライブラリ・ウィンドウには，［エキスパート ビュー］が備えているものと同様の編集機能があります。

［エキスパート ビュー］で使用できるオプションの詳細については，『**HP QuickTest Professional ユーザーズ・ガイド**』を参照してください。

関数ライブラリの作成

新しい関数ライブラリは，いつでも作成できます。

QuickTest で新しい関数ライブラリを作成するには，次の手順を実行します。

次のいずれかを実行します。

➤［**ファイル**］＞［**新規作成**］＞［**関数ライブラリ**］を選択します。

➤［**新規作成**］ボタンの下向き矢印をクリックし，［**関数ライブラリ**］を選択します。

新しい関数ライブラリが開きます。

これで，関数ライブラリに内容を追加し，保存できます。関数ライブラリに内容を追加すると，［エキスパート ビュー］の内容に適用されるのと同じ書式設定が適用されます。書式設定は，必要に応じて変更できます。詳細については，379 ページ「関数ライブラリ・ウィンドウのカスタマイズ」を参照してください。

関数ライブラリを開く

QuickTest では，すでに別のドキュメントが開いていても，ファイル・システムまたは Quality Center プロジェクトに保存されている関数ライブラリを開くことができます。関数ライブラリは，当該ファイルに対する読み取り許可または読み書きの許可がある場合にのみ開けます。

注：関数ライブラリ内に定義されている関数をコンポーネントまたはアプリケーション領域の中で使えるようにするには，関数ライブラリを Quality Center プロジェクトに保存し，アプリケーション領域に関連付ける必要があります。詳細については，460 ページ「関数ライブラリの関連付け」を参照してください。

関数ライブラリを編集モードで開くか読み取り専用モードで開くかを選択できます。

- **編集モード**：関数ライブラリを表示，変更できます。あるコンピュータで関数ライブラリが開いている間は，ほかのユーザは，そのファイルを読み取り専用モードで表示できますが，変更はできません。
- **読み取り専用モード**：関数ライブラリを表示することはできますが，変更はできません。標準設定では，現在ほかのコンピュータで開いている関数ライブラリを開くと，読み取り専用モードになります。関数ライブラリを表示する一方で，ほかのユーザが変更できるようにもしたい場合は，関数ライブラリを読み取り専用モードで開けます。

ヒント：自分のドキュメントの関数から別の関数ライブラリの関数定義へ直接移動することもできます。詳細については，396 ページ「関数ライブラリの特定関数への移動」を参照してください。

既存の関数ライブラリを開くには，次の手順を実行します。

1 次のいずれかを実行します。

- ［**ファイル**］＞［**開く**］＞［**関数ライブラリ**］を選択します。
- ［**開く**］ボタンの下向き矢印をクリックし，［**関数ライブラリ**］を選択します。

［関数ライブラリを開く］ダイアログ・ボックスが開きます。

ヒント：関数ライブラリを読み取り専用モードで開くには，［関数ライブラリを開く］ダイアログ・ボックスで［**読み取り専用モードで開く**］チェック・ボックスを選択します。

---

2 サイドバーで，**ファイル・システム**または Quality Center **テスト・リソース**など，ファイルの場所を選択します。関数ライブラリを参照して選択し［**開く**］をクリックします。

新しいウィンドウに指定した関数ライブラリが開きます。これで，関数ライブラリの内容を表示，変更できるようになります。詳細については，397 ページ「関数ライブラリの編集」および 398 ページ「関数ライブラリのデバッグ」を参照してください。

ヒント：

最近作成した，または開いた関数ライブラリは，［**ファイル**］メニューに表示される最近使用したファイルから選択できます。

関数ライブラリが開いているコンポーネントまたはアプリケーション領域に関連付けられている場合，次の方法でも開くことができます。

- ［リソース］表示枠で，関数ライブラリをダブルクリックするか，関数ライブラリを右クリックして［**関数ライブラリを開く**］を選択します。
- ［使用可能なキーワード］表示枠で，関数ライブラリをダブルクリックするか，関数ライブラリを右クリックして［**リソースを開く**］を選択します。
- ［**リソース**］＞［**関連付けられている関数ライブラリ**］を選択します。

関数ライブラリの保存

QuickTest で関数ライブラリを作成または編集したら，Quality Center プロジェクトに関数ライブラリを保存できます。

標準設定では，QuickTest によって，**.vbs** または **.txt** などの別の拡張子を指定したり，拡張子を削除してしまわないかぎり，**.qfl** 拡張子を持つ関数ライブラリが保存されます。

ヒント：

- 関数ライブラリに変更を加えると，関数ライブラリが保存されるまで，タイトル・バーにアスタリスク（*****）が表示されます。
- 開いているドキュメントをすべて保存するには，［**ファイル**］＞［**すべて保存**］を選択します。まだ保存されていない新規ファイルについては，保存先を指定するように求められます。
- 複数のドキュメントを保存するには，［**ウィンドウ**］＞［**ウィンドウ**］を選択します。［ウィンドウ］ダイアログ・ボックスで，保存するドキュメントを選択し，［**保存**］ボタンをクリックします。まだ保存されていない新規ファイルについては，保存先を指定するように求められます。
- アクティブな関数ライブラリを別の名前または別のパスで保存するには，［**ファイル**］＞［**名前を付けて保存**］を選択します。

ファイルシ・ステムまたは Quality Center プロジェクトに関数ライブラリを保存するには，次の手順を実行します。

1 保存する対象となる関数ライブラリがアクティブなドキュメントであることを確認します（フォーカスを対象に移動するには，対象関数ライブラリのタブをクリックします)。

2 次の手順のいずれかを実行します。

- ［**保存**］ボタンをクリックします。
- ［**ファイル**］＞［**保存**］を選択します。
- 関数ライブラリ・ドキュメントのタブを右クリックし，［**保存**］を選択します。

この関数ライブラリを以前に保存したことがあれば，変更された関数ライブラリが保存されます。この関数ライブラリを初めて保存する場合は，［関数ライブラリを保存］ダイアログ・ボックスが開きます。

3 Quality Center プロジェクト関数ライブラリを保存します。関数ライブラリをビジネス・プロセス・テストで使用する場合は，関数ライブラリを Quality Center プロジェクトに保存する必要があります。

開いている QuickTest ドキュメント間でのフォーカスの移動

コンポーネントまたはアプリケーション領域が開いている間，複数の関数ライブラリを開き，開いているすべてのドキュメント間でフォーカスを移動することができます。

開いている QuickTest ドキュメント間でフォーカスを移動するには，次の手順を実行します。

次のいずれかを実行します。

➤ ドキュメント表示枠で，必要なドキュメントのタブをクリックします。

ヒント：スペースが足りず，すべてのタブが表示されない場合は，ドキュメント表示枠の左右のスクロール矢印を使用して，必要なドキュメントのタブを表示します。

➤ キーボードの CTRL キーを押しながら TAB キーを押し，開いているドキュメント間を切り替えます。

➤［**ウィンドウ**］メニューから必要なドキュメントを選択します。

➤［**ウィンドウ**］＞［**ウィンドウ**］を選択し，［ウィンドウ］ダイアログ・ボックスで必要なドキュメントを選択して，［**切り替え**］ボタンをクリックします。

関数ライブラリの特定関数への移動

関数の呼び出しを挿入した後，ソース・ドキュメント内の当該関数の定義へ直接移動できます。関数の定義は，同じ関数ライブラリ，またはコンポーネント（アプリケーション領域を通じて）に関連付けられている別の関数ライブラリに置くことができます。関数の定義が含まれるドキュメントがすでに開いている場合は，そのウィンドウがアクティブになります（そこにフォーカスが移動します）。ドキュメントが閉じている場合は，そのドキュメントが読み取り専用モードで開きます。

関数の定義に移動するには，次の手順を実行します。

1 関数ライブラリで，該当する関数が含まれているステップをクリックします。

2 次のいずれかを実行します。

➤［**編集**］＞［**詳細設定**］＞［**関数定義に移動**］を選択します。

➤ ステップを右クリックして，ショートカット・メニューの［**関数定義に移動**］を選択します。

該当するドキュメントがアクティブになり（関数の定義が別の関数ライブラリにある場合），関数の定義の先頭にカーソルが置かれます。

関数ライブラリの編集

［エキスパート ビュー］の QuickTest 編集機能を使用して，関数ライブラリをいつでも編集できます。

ドキュメント間で関数（またはその一部分）のドラッグ・アンド・ドロップが可能です（それには，［**最小化**］ボタン（QuickTest ウィンドウの［**最小化 / 最大化**］ボタンの下にあります）をクリックすることで，タブ付きのドキュメントを別々のドキュメント表示枠に分ける必要があります）。

手動で，またはステップ・ジェネレータを使用して，関数ライブラリにステップを追加できます。ステップ・ジェネレータを使用すれば，**予約オブジェクト**（ユーティリティ・オブジェクトなど，機能拡張のために提供されるオブジェクト），VBScript 関数（MsgBox など），ユーティリティ・ステートメント（Wait など），同じ関数ライブラリに定義されているユーザ定義関数を含むステップを追加できます。IntelliSense は，コンポーネントに定義されているすべての関数，または関連付けられている関数ライブラリに定義されているパブリック関数で使用できます。

注：関数ライブラリでは，IntelliSense を使用してテスト・オブジェクトの名前またはコレクションを表示できません。これは，関数ライブラリがオブジェクト・リポジトリに接続されていないためです。

構文をチェックするように QuickTest に指定するには，［**構文チェック**］ボタンをクリックするか，［**ツール**］＞［**構文チェック**］を選択します。

ヒント：VBScript の使用の詳細については，343 ページ「VBScript の基本的な構文の理解」を参照してください。

読み取り専用の関数ライブラリの編集

関数ライブラリを読み取り専用モードで開き，その後でその関数ライブラリに変更を加えることにした場合，その関数ライブラリを編集可能ファイルに変換できます。ただし，ほかのユーザがその関数ライブラリをロックしていない場合に限ります。関数ライブラリを開くときに使用できるオプションの詳細については，392 ページ「関数ライブラリを開く」を参照してください。

注：デバッグ・セッション中は，すべてのドキュメント（コンポーネントおよび関数ライブラリなど）が読み取り専用になります。デバッグ・セッション中にドキュメントを編集するには，まず，デバッグ・セッションを停止する必要があります。

読み取り専用の関数ライブラリを編集するには，次の手順を実行します。

［**ファイル**］＞［**編集可能にする**］を選択するか，［**編集可能にする**］ボタンをクリックします。これで，関数ライブラリを編集できます。

関数ライブラリのデバッグ

関数ライブラリをデバッグする前に，関数ライブラリをコンポーネント（アプリケーション領域を通じて）に関連付けて，ライブラリ内の関数への呼び出しを少なくとも 1 つ挿入しておく必要があります。次に，コンポーネントを実行し，関数ライブラリのコンテキストで実行セッションを一時停止し，関数ライブラリをデバッグできます。たとえば，［デバッグ ビューア］を使用して，関数ライブラリのオブジェクトまたは変数の現在の値を表示，設定，変更した

り，手作業で追加の VBScript コマンドを実行したりできます。関数（ユーザ定義関数を含む）のステップ・イントゥ，ブレークポイントの設定，ブレークポイントでの停止，式の表示などが可能です。デバッグは特定のステップから開始したり，特定のステップで一時停止するように指定したりできます。詳細については，739 ページ「コンポーネントと関数ライブラリのデバッグ」を参照してください。

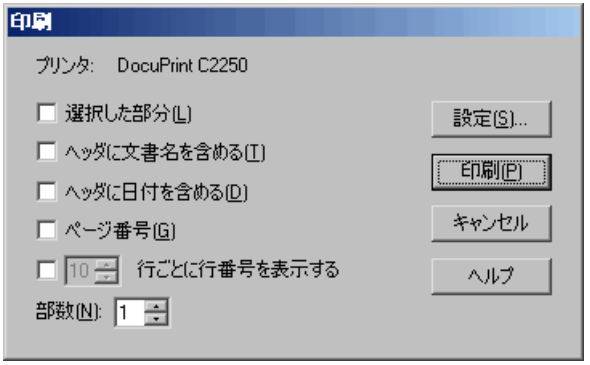
注：デバッグ・セッション中は，すべてのドキュメントが読み取り専用になり，編集できません。デバッグ・セッション中にドキュメントを編集するには，まず，デバッグ・セッションを停止する必要があります。

関数ライブラリの印刷

関数ライブラリは，いつでも印刷できます。印刷出力には追加の情報を含めることもできます。

関数ライブラリを印刷するには，次の手順を実行します。

1 ［**印刷**］ボタンをクリックするか，［**ファイル**］＞［**印刷**］を選択します。［印刷］ダイアログ・ボックスが開きます。



2 印刷オプションを指定します。

- ［**プリンタ**］: 印刷ジョブの送信先となるプリンタが表示されます。プリンタを変更するには，［**設定**］ボタンをクリックします。
- ［**選択した部分**］: 関数ライブラリ内の現在選択されている（強調表示されている）テキストのみ印刷されます。
- ［**ヘッダに文書名を含める**］: 印刷出力の上部に関数ライブラリの名前が挿入されます。
- ［**ヘッダに日付を含める**］: 印刷出力の上部にその日の日付が挿入されます。日付書式は Windows の地域のオプションに基づきます。
- ［**ページ番号**］: 印刷出力の下部にページ番号が挿入されます（例：page 1 of 3）。
- ［**X 行ごとに行番号を表示する**］: 指定どおりに，スクリプト行の左側に行番号が表示されます。
- ［**部数**］: ドキュメントを印刷する回数を指定します。

3 別のプリンタに印刷する場合やプリンタの設定を変更する場合は，［**設定**］をクリックして［プリンタの設定］ダイアログ・ボックスを表示します。

4 ［**印刷**］をクリックすると，選択内容に従って印刷されます。

関数ライブラリを閉じる

個々の関数ライブラリを閉じたり，複数の関数ライブラリが開いている場合は，その一部または全部を同時に閉じたりすることができます。いずれかの関数ライブラリが保存されていないと，保存するよう QuickTest に求められます。

個々の関数ライブラリを閉じるには，次の手順を実行します。

次のいずれかを実行します。

- 保存する関数ライブラリがアクティブなドキュメントであることを確認し（当該関数ライブラリにフォーカスを移動するには，関数ライブラリのタブをクリックします），［**ファイル**］＞［**閉じる**］を選択します。
- 関数ライブラリ・ドキュメントのタブを右クリックし，［**閉じる**］を選択します。

- 関数ライブラリ・ウィンドウの右上角にある［**閉じる**］ボタンをクリックします。
- ［**ウィンドウ**］＞［**ウィンドウ**］を選択します。［ウィンドウ］ダイアログ・ボックスで，閉じる関数ライブラリが選択されていない場合は選択し，［**ウィンドウを閉じる**］ボタンをクリックします。

複数の関数ライブラリを閉じるには，次の手順を実行します。

［**ウィンドウ**］＞［**ウィンドウ**］を選択します。［ウィンドウ］ダイアログ・ボックスで，閉じる関数ライブラリを選択し，［**ウィンドウを閉じる**］ボタンをクリックします。

開いているすべての関数ライブラリを閉じるには，次の手順を実行します。

［**ファイル**］＞［**全関数ライブラリを閉じる**］を選択するか，［**ウィンドウ**］＞［**全関数ライブラリを閉じる**］を選択します。

関連付けられている関数ライブラリを使用した作業

QuickTest では，関数，サブルーチン，モジュールなどが含まれる関数ライブラリを作成し，そのファイルをアプリケーション領域に関連付けることができます。これにより，ユーザまたは各分野のエキスパートは，関連付けられている関数ライブラリのパブリック関数やサブルーチンへの呼び出しを，該当するアプリケーション領域に関連付けられている任意のコンポーネントから挿入できるようになります（関数ライブラリに保存されているパブリック関数は，関連付けられている任意のコンポーネントから（アプリケーション領域経由で）呼び出すことができます。これに対して，プライベート関数は，同じ関数ライブラリ内からのみ呼び出すことができます）。

注：標準の VBScript 構文で書かれたあらゆるテキスト・ファイルを関数ライブラリとして使えます。

既存のアプリケーション領域に関連付けられている関数ライブラリのリストはアプリケーション領域の関数ライブラリ表示枠（[**アプリケーション領域**] > [**関数ライブラリ**] のサイドバー・ボタン）で編集できます。詳細については，460 ページ「関数ライブラリの関連付け」を参照してください。

Quality Center での関連付けられている関数ライブラリを使用した作業

Quality Center および関連関数ライブラリを使用するときには，アプリケーション領域の関数ライブラリ表示枠で関連ファイルを指定する前に，関連関数ライブラリを Quality Center プロジェクトのテスト・リソース・モジュールに保存する必要があります。Quality Center プロジェクトには新規または既存の関数ライブラリを追加できます。

コンポーネントは，そのアプリケーション領域に関連付けられている関数にアクセスします。したがって，Quality Center プロジェクトに格納されていて，アプリケーション領域に関連付けられている関数ライブラリに加えた変更は，関連コンポーネントに影響を与える場合があります。Quality Center プロジェクトに格納されていて，アプリケーション領域に関連付けられている関数ライブラリに変更を加える場合は，当該アプリケーション領域を使用するコンポーネントに及ぼす影響を考慮してください。Quality Center では，ビジネス・コンポーネント・モジュールの［依存関係］タブに，アプリケーション領域を使用するコンポーネントのリストを表示できます。詳細については，944 ページ「[依存関係］タブ」を参照してください。

アプリケーション領域との関数ライブラリの関連付け

［リソース］表示枠または現在アクティブな関数ライブラリから，開いているアプリケーション領域に関数ライブラリを関連付けることができます。

また，関連付けられている関数ライブラリ・リストを使用して，現在開いているアプリケーション領域に関数ライブラリを関連付けることもできます。詳細については，404 ページ「関数ライブラリの関連付けの変更」を参照してください。

開いている関数ライブラリをアプリケーション領域に関連付けるには，次の手順を実行します。

1 関数ライブラリを関連付ける対象となるアプリケーション領域が QuickTest で開かれていることを確認します。

2 QuickTest で関数ライブラリを作成するか開きます（次の手順に進む前に，アプリケーション領域に関連付ける関数ライブラリがアクティブなドキュメントであることを確認します。対象関数ライブラリにフォーカスを移動するには，関数ライブラリのタブをクリックします)。詳細については，391 ページ「関数ライブラリの管理」を参照してください。

3 関数ライブラリを，Quality Center プロジェクトに保存します。詳細については，394 ページ「関数ライブラリの保存」を参照してください。

4 QuickTest の中で，［**ファイル**］>［**ライブラリ ' < Function Library > ' を ' アプリケーション領域 ' に関連付ける**］を選択するか，関数ライブラリの中で右クリックして［**ライブラリ ' < Function Library > ' を ' アプリケーション領域 ' に関連付ける**］を選択します。QuickTest によって，開いているアプリケーション領域に関数ライブラリが関連付けられます。

関数ライブラリの関連付けの変更

アプリケーション領域［関数ライブラリ］表示枠で変更できます。リストに関数ライブラリを追加したり，リストから削除したりできます。また，関数ライブラリの優先順位を変更することもできます。

アプリケーション領域に関数ライブラリを関連付けるには，次の手順を実行します。

1 QuickTest で，アプリケーション領域を開いてサイドバーの［**関数ライブラリ**］ボタンをクリックします。

2 ［**関連付けられている関数ライブラリ**］リストで，［**追加**］ボタンをクリックします。参照ボタンが表示され，Quality Center プロジェクト内の関数ライブラリを参照できます。

3 アプリケーション領域に関連付ける関数ライブラリを選択し，［**開く**］をクリックします。

関連付けられている関数ライブラリの優先順位を変更するには，次の手順を実行します。

［関数ライブラリ］表示枠の関連付けられた関数ライブラリのリストで，優先順位を付ける関数ライブラリを選択し，**上向き**矢印および**下向き**矢印を使用します。

詳細については，460 ページ「関数ライブラリの関連付け」を参照してください。

関連付けられている関数ライブラリを削除するには，次の手順を実行します。

次のいずれかを実行します。

➤ ［リソース］表示枠で，関数ライブラリを右クリックして［**関数ライブラリを削除**］を選択するか，関数ライブラリを選択して DELETE キーを押します。

➤ ［関数ライブラリ］表示枠の関連付けられた関数ライブラリのリストで，削除する関数ライブラリを選択し，［**削除**］ボタンをクリックします。

詳細については，460 ページ「関数ライブラリの関連付け」を参照してください。

関数定義ジェネレータの使用方法

QuickTest の関数定義ジェネレータを使用すれば，新規ユーザ定義関数のための定義を生成し，その定義にヘッダ情報を追加できます。その後，必要に応じて，その関数をテスト・オブジェクトに登録できます。必要な情報を入力すると，関数定義ジェネレータが基本的な関数定義を自動的に作成します。関数定義を定義した後，それを関数ライブラリに挿入してアプリケーション領域に関連付けます。最後に，内容（コード）を追加して関数を完成させます。

テスト・オブジェクトに関数を登録すると，関数は，そのテスト・オブジェクトによって呼び出すことができ，そのテスト・オブジェクトの使用可能な操作のリストに表示されます。

テスト・オブジェクトに関数を登録しないと，この関数はグローバル操作となり，ステップ・ジェネレータの［**操作**］ボックスの操作一覧に（関数ライブラリの場合），キーワード・ビューで［**操作**］項目が選択されている場合は［**操作**］リストに，また IntelliSense の使用時に表示されます。関数を登録すると，その関数が登録されているテスト・オブジェクトが選択された場合にキーワード・ビューに表示される標準の操作として定義できます。

最後に，ステップ・ジェネレータ（関数ライブラリの場合）またはキーワード・ビューで操作にカーソルを位置付けたときと，IntelliSense の使用時に表示されるツールチップを定義することによって，ユーザ定義関数に説明を付けることができます。また，ユーザ定義関数を含むステップが実際に何をするかについて説明した文章を追加することもできます。この文章は，［**注釈**］カラムに表示されます。

関数定義ジェネレータに情報を入力するにつれて，でき上がりつつある関数定義が［**プレビュー**］領域に表示されます。関数の定義が完了したら，アクティブな QuickTest ドキュメントに定義を挿入します。以降，関連付けられているコンポーネント（アプリケーション領域経由）は関数にアクセスできるようになります。最後に，関数の内容（コード）を追加します。

次の項では，関数定義ジェネレータで関数を作成するときに実行する手順の概要を説明します。

関数定義ジェネレータを使用するには，次の手順を実行します。

1 406 ページ「関数定義ジェネレータの起動」の説明に従って，関数定義ジェネレータを開きます。

2 408 ページ「関数の定義」の説明に従って，関数を定義します。

3 409 ページ「関数ジェネレータを使用した関数の登録」の説明に従って，必要に応じて関数をテスト・オブジェクトに登録します。

標準設定では，テスト・オブジェクトに登録されない関数は，ステップ・ジェネレータの［**関数**］カテゴリ（関数ライブラリの場合）またはキーワード・ビューの［**操作**］項目を選択することによって呼び出せる，あるいは IntelliSense の使用時に呼び出せるグローバル関数として自動的に定義されます。関数をテスト・オブジェクトに登録した場合は，その関数（操作）をそのテスト・オブジェクトの標準設定の操作として定義することもできます。

4 413 ページ「関数の引数の指定」の説明に従って，関数に引数を追加します。

5 ヘッダ情報を追加して関数をドキュメント化します。詳細については，414 ページ「関数への説明の追加」を参照してください。

6 関数の仕上げの前に，416 ページ「関数のプレビュー」の説明に従って，その関数をプレビューします。

7 416 ページ「別のユーザ定義関数の生成」の説明に従って，必要に応じて別の関数定義を生成します。

8 417 ページ「ユーザ定義関数の仕上げ」の説明に従って，アクティブなドキュメントに各関数を挿入し，それらの関数に内容を追加して，各関数を仕上げます。

注： この項で説明する手順では，前述の手順を実行しているものと想定しています。

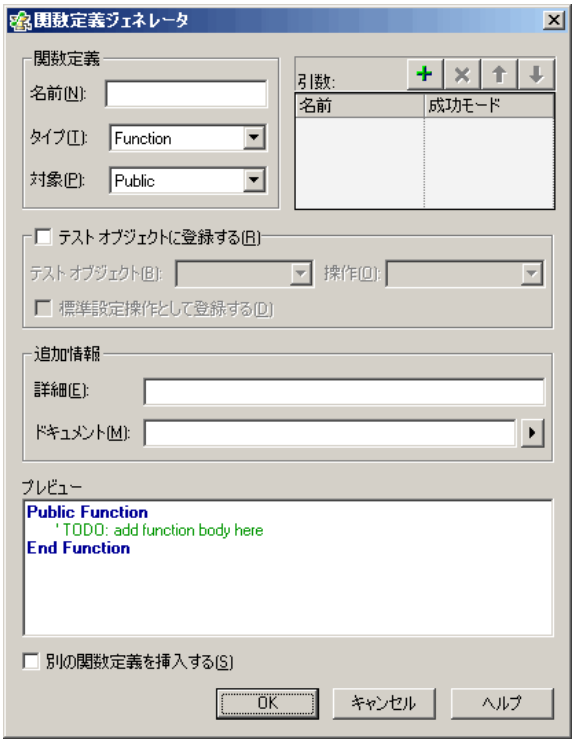
関数定義ジェネレータの起動

QuickTest から関数定義ジェネレータを開きます。

関数定義ジェネレータを開くには，次の手順を実行します。

1 関数定義を挿入する対象となる関数ライブラリがアクティブなドキュメントであることを確認します（対象ドキュメントにフォーカスを移動するには，関数ライブラリのタブをクリックします）。これは，関数定義の完了後，関数定義ジェネレータが現在アクティブなドキュメントに関数を挿入するためです。

2［**挿入**］＞［**関数定義ジェネレータ**］を選択するか，［**関数定義ジェネレータ**］ボタンをクリックします。関数定義ジェネレータが開きます。



関数定義ジェネレータを開いた後，408 ページ「関数の定義」で説明するように，新しい関数の定義を開始できます。

関数の定義

関数定義ジェネレータを開いたら，関数の定義を開始できます。

たとえば，指定されたプロパティの値を確認する関数を定義する場合，関連付けられている任意のコンポーネント（関数ライブラリがそのアプリケーション領域に関連付けられている場合のみ）から呼び出せるように，その関数に VerifyProperty という名前を付けてパブリック関数として定義できます（プライベートとして定義した関数は，同じ関数ライブラリ内にある別の場所からのみ呼び出せます。プライベート関数は，テスト・オブジェクトに登録できません）。

関数を定義するには，次の手順を実行します。

1 ［**名前**］ボックスに，新しい関数の名前を入力します。ステップ・ジェネレータ（関数ライブラリの場合）またはキーワード・ビューから簡単に選択できるように，何をする操作なのかがはっきりとわかる名前を付けてください。関数名には，英字以外の文字を含めることはできません。また，関数名は英字で始まらなければならず，スペースや次の文字を含めてはなりません。
! @ # $ % ^ & * ( ) + = [ ] ¥ { } | ; ' : "" , / < > ?

注：ユーザ定義関数に，組み込みの関数と同じ名前（たとえば，GetLastError, MsgBox，Print など）を付けないようにします。組み込み関数は，ユーザ定義関数に優先します。したがって，組み込み関数と同じ名前のユーザ定義関数を呼び出しても，組み込み関数が代わりに呼び出されます。組み込み関数のリストについては，ステップ・ジェネレータ（［**挿入**］＞［**ステップ ジェネレータ**］）の「**組み込み関数**」リストを参照してください。

---

2 関数またはサブルーチンのどちらを定義するのかに応じて，［**タイプ**］リストから［**Function**］または［**Sub**］を選択します。

3 [**対象**] リストから，関数の適用範囲として，[**Public**]（アプリケーション領域がこの関数ライブラリに関連付けられている任意のコンポーネントからこの関数を呼び出せるようにする場合）または [**Private**]（同じ関数ライブラリ内の別の場所からのみこの関数を呼び出せるようにする場合）を選択します。標準設定では，適用範囲は [**Public**] に設定されています（パブリック関数のみ，テスト・オブジェクトに登録することができます）。

注： ユーザ定義関数を手作業で作成し，範囲を [**Public**] とも [**Private**] とも定義しなかった場合，その関数は標準設定でパブリック関数として扱われます。

パブリック関数を定義した後，関数を登録できます。また，プライベート関数を定義した場合，あるいは関数を登録しない場合は，引き続き関数に引数を指定できます。詳細については，413 ページ「関数の引数の指定」を参照してください。

関数ジェネレータを使用した関数の登録

パブリック関数をテスト・オブジェクトに登録すると，その関数（操作）をテスト・オブジェクトに対して実行できるようになります。関数をテスト・オブジェクトに登録するときには，既存の操作の機能をオーバーライドすることも，そのテスト・オブジェクトに対する新しい操作として登録することもできます。

関数をテスト・オブジェクトに登録すると，キーワード・ビューの中で [**項目**] リストからそのテスト・オブジェクトを選択したときに [**操作**] リストに，また IntelliSense に，およびステップ・ジェネレータの一般的な [**操作**] リスト（関数ライブラリの場合）に，関数が操作として表示されます。関数をテスト・オブジェクトに登録すると，その関数はそのテスト・オブジェクトからしか呼び出せなくなります。

テスト・オブジェクトに関数を登録することを選択した場合，関数定義ジェネレータの右上角にある [引数] 領域に，1 番目の引数として自動的に **test_object** 引数が追加されます。また，関数定義ジェネレータは，関数定義のすぐ後ろに，適切な引数値を持つ RegisterUserFunc ステートメントを自動的に追加します。

関数をテスト・オブジェクトに登録するときに，任意でその関数をテスト・オブジェクトの標準設定の操作として定義することもできます。定義した場合は，QuickTest エンジニアまたは各分野のエキスパートが［**項目**］リスト内の関連付けられているテスト・オブジェクトを選択したときに，標準設定で［**操作**］カラムにその関数が表示されます。関数をテスト・オブジェクトの標準設定の関数として定義すると，RegisterUserFunc ステートメントの 4 番目の引数の値として **True** が指定されます。

関数を特定のテスト・オブジェクトに登録しなかった場合，その関数は自動的にグローバル関数として定義されます。グローバル関数は，ステップ・ジェネレータの［**関数**］カテゴリ（関数ライブラリの場合），またはキーワード・ビューの［**操作**］項目を選択して呼び出します。グローバル関数のリストは，ステップ・ジェネレータで［**関数**］カテゴリを選択した場合は［**操作**］ボックスに（関数ライブラリの場合），キーワード・ビューの［**項目**］リストから［**操作**］項目を選択した場合は［**操作**］リストにまた IntelliSense の使用時に，アルファベット順に表示されます。

QuickTest によって，［リソース］表示枠に表示されている順番で関数ライブラリの中で関数が検索されます。指定した関数ライブラリ内に関数名の一致する関数が複数見つかった場合は，関数ライブラリ内で最後に検出された関数が使用されます。QuickTest によって 2 つの異なる関数ライブラリで同じ名前の 2 つの関数が見つかった場合，優先順位の高い方の関数ライブラリの関数が使用されます。混乱を避けるために，1 つのアプリケーション領域に関連付けられているリソースの中では，それぞれの関数に一意の名前を付けることを推奨します。

ヒント：この時点で関数を登録しなかった場合は，後ほど手作業で，その関数の後に次の例のように RegisterUserFunc ステートメントを付け加えて，登録することができます。RegisterUserFunc "WebEdit", "MySet", "MySetFunc"

この例では，ユーザ定義関数 MySetFunc を使用して，MySet メソッド（操作）を WebEdit テスト・オブジェクトに追加しています。追加後，QuickTest エンジニアまたは各分野のエキスパートがキーワード・ビューで［**項目**］リストから WebEdit テスト・オブジェクトを選択すると，［**操作**］リストに，WebEdit テスト・オブジェクトの登録されているほかの操作，およびあらかじめ用意されている操作とともに MySet 操作が表示されます。

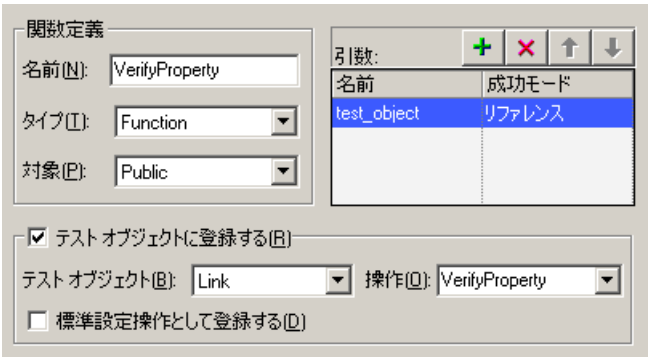
また，関数をほかのテスト・オブジェクトに登録することもできます。それには，関数コードを関数ライブラリに保存するときに，RegisterUserFunc ステートメントを複製し（コピーして貼り付けて），必要に応じて引数値を変更します。

この関数を標準設定の関数として定義するには，RegisterUserFunc ステートメントの 4 番目の引数の値を **True** に指定します。次に例を示します。
RegisterUserFunc "WebEdit", "MySet", "MySetFunc", True

注：登録した関数もグローバル関数も，コンポーネントのアプリケーション領域に関連付けられている関数ライブラリに追加された後にのみ，コンポーネントから呼び出せます。

関数をテスト・オブジェクトに登録するには，次の手順を実行します。

1 ［**テスト オブジェクトに登録する**］チェック・ボックスを選択します。この領域内のオプションが使用可能になり，関数定義ジェネレータの右上角にある［**引数**］領域内の引数のリストに test_object という新しい引数が自動的に追加されます（test_object 引数は，関数を登録する対象となるテスト・オブジェクトを受け取ります）。



注 : ［**テスト オブジェクトに登録する**］チェック・ボックスをクリアすると，標準の test_object 引数が［**引数**］領域から自動的に削除されます（名前を変更しなかった場合）。

---

2 使用可能なオブジェクトのリストから **Test object** を選択します。たとえば，サンプルの VerifyProperty 関数は，**Link** テスト・オブジェクトに登録することになるでしょう。

3 テスト・オブジェクトに追加またはオーバーライドする**操作**を指定します。

- ➤ 新しい操作を定義するには，［**操作**］ボックスに新しい操作の名前を入力します。たとえば，サンプルの VerifyProperty 関数の場合は，新しい VerifyProperty 操作を定義します。
- ➤ 既存の操作の標準機能をオーバーライドするには，［**操作**］ボックスで使用可能な操作のリストから操作を選択します。

4 QuickTest エンジニアまたは各分野のエキスパートが，関連付けられている項目を選択したときに関数を標準設定の操作として［**操作**］カラムに表示されるようにするには，［**標準設定操作として登録する**］チェック・ボックスを選択します。

たとえば，VerifyProperty 操作を Link テスト・オブジェクトの標準設定の操作として定義した場合は，RegisterUserFunc ステートメントの 4 番目の引数に値として **True** が定義されます。その構文は次のようになります。

```
RegisterUserFunc "Link", "VerifyProperty", "VerifyProperty", True
```

テスト・オブジェクト登録情報を指定した後，関数のその他の引数を指定します。を参照してください。

関数の引数の指定

基本的な関数定義を行い，テスト・オブジェクト登録情報を指定したら，必要に応じて関数の引数を指定できます。

たとえば，409 ページ「関数ジェネレータを使用した関数の登録」で説明した例のように，テスト・オブジェクトに関数を登録することにした場合は，1 番目の引数の test_object に加えて，prop_name（チェック対象プロパティの名前）および expected_value（プロパティの期待値）という引数を割り当てることができます。関数が正しく動作するためには，要求された引数を定義する必要があります。

引数は任意の順序でリスト表示できます。ただし，関数をテスト・オブジェクトに登録する場合は，常に先頭の引数がテスト・オブジェクトを受け取らなければなりません。

関数の引数を定義するには，次の手順を実行します。

［**引数**］領域で，関数の引数を指定します。必要に応じて，引数はいくつでも追加できます。わかりやすくするために，それぞれの引数には，どのような値を入力する必要があるかを示す名前を付けてください。

- 引数を追加するには， をクリックして引数の名前を入力します。引数には，その引数にどのような値を入力する必要があるかをはっきりと示す名前を付けてください。引数名には，英字以外の文字を含めることはできません。また，引数名は英字で始まらなければならず，スペースや次の文字を含めてはなりません。 ! @ # $ % ^ & * ( ) + = [ ] ¥ { } | ; ' : "" , / < > ?引数ごとに，［**成功モード**］ボックスで適切なモードを選択し，引数を関数［**値**］または［**リファレンス**］に渡すように QuickTest に指示します。
- 引数を削除するには，その引数を選択して をクリックします。これで，その引数が関数定義ジェネレータから削除されます。
- 引数の順序を設定するには，矢印 と を使用します。引数の順序は関数コードのわかりやすさに影響するだけですが，パブリック関数を登録する場合は例外です。その場合は，先頭の引数がテスト・オブジェクトを受け取る必要があります。

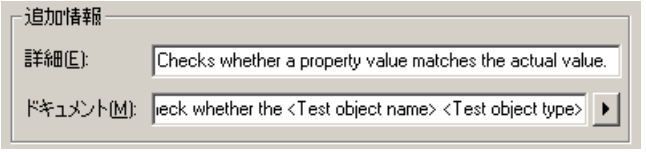
関数への説明の追加

関数定義ジェネレータでは，ユーザ定義関数にヘッダ情報を追加できます。カーソルが操作の上に置かれたときにツールチップとして表示される説明を追加できます。QuickTest エンジニアと各分野のエキスパートは，このツールチップの情報に基づいて，使用可能な操作のリストからどの操作を選択するべきかを判断できます（説明文は可能な限り短く簡潔にしておくことをお勧めします）。

さらに，当該の関数を使用するステップが正確に何をするかを指定する注釈を追加できます。テスト・オブジェクト名，テスト・オブジェクト・タイプ，およびテキスト内の任意の引数値を含めることができます。また，必要に応じてテキストを手作業で入力することもできます。ここで追加したテキストは，キーワード・ビューの［**注釈**］カラムに表示されます。したがって，明確でわかりやすい文章にしなければなりません。

たとえば，検索エンジンから「HP」へのリンクをチェックする場合，関数定義ジェネレータを使用して次の注釈を定義します。

'@Documentation Check if the <Test object name> <Test object type> <prop_name> value matches the expected value: <expected_value>.



キーワード・ビューで引数の値を選択した後は，この注釈はたとえば次のように表示されます。Check if the "Management Software" link "text" value matches the expected value: "Business Technology Optimization (BTO) Software".

ヒント：キーワード・ビューで任意のカラム・ヘッダを右クリックして，［**ドキュメントのみ**］オプションを選択すると，ステップのリストを表示または印刷することができます。このオプションを選択すると，QuickTest または Quality Center に［**注釈**］カラムだけが（ビジネス・コンポーネントの場合にはコメントも）表示されるようになります。［**編集**］＞［**ドキュメントをクリップボードにコピー**］を選択して，任意のアプリケーションに注釈を貼り付けることもできます。したがって，ステップに関してこのカラムに表示される文章は，手動テスト用の指示としても使用できる明確な文章でなければなりません。

関数に説明を付けるには，次の手順を実行します。

1 ［**詳細**］ボックスに，ツールチップとして表示されるテキストを入力します。ツールチップは，ステップ・ジェネレータの［**操作**］リスト（関数ライブラリの場合），キーワード・ビューの［**操作**］カラムおよび IntelliSense でカーソルが関数名の上に置かれたときに表示されます。

たとえば，サンプルの VerifyProperty 関数の場合は，次のように入力できます。Checks whether a property value matches the actual value.

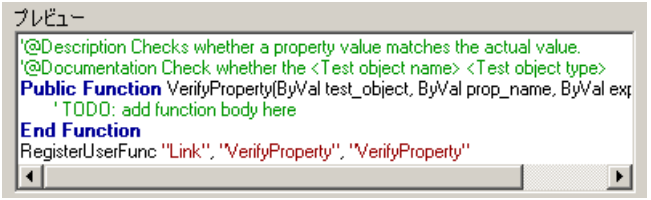
2 ［**ドキュメント**］ボックスに，キーワード・ビューの［**注釈**］カラムに表示させるテキストを入力します。［**注釈**］のテキストには引数を含めることができます。それには，▶をクリックして使用する引数を選択します。

［**テスト オブジェクトに登録する**］チェック・ボックスがオンになっている場合は，▸をクリックすることで，表示されたリストから，［**Test object name**］項目または［**Test object type**］項目，あるいはその両方を［**注釈**］カラムに追加することもできます。これらのテスト・オブジェクト項目や引数項目を［**注釈**］テキストに含めると，それらの項目は，対応するテスト・オブジェクト名，テスト・オブジェクト・タイプ，または引数値と動的に置き換えられます。

関数のプレビュー

関数を定義するにつれて，その関数のコードが［**プレビュー**］領域に読み取り専用形式で表示されます。ここで関数の内容を確認して，必要なら関数定義ジェネレータのさまざまな領域で変更を加えることができます。

たとえば，サンプルの **VerifyProperty** 関数の場合は，［**プレビュー**］領域に次のようなコードが表示されます。



コードを確認したら（アクティブなドキュメントに関数を挿入する前に），別の関数定義を生成するか，または定義した関数のコードを仕上げるかを選択できます。

別のユーザ定義関数の生成

コードをプレビューしたら（アクティブなドキュメントに関数を挿入する前に），さらにほかの関数定義を生成するかどうかを選択できます。

注：ほかの関数を定義しない場合は，次の項に進みます。

ほかのユーザ定義関数を生成するには，次の手順を実行します。

1 ［**別の関数定義を挿入する**］チェック・ボックスを選択して，［**挿入**］をクリックします。アクティブなドキュメントに関数定義が挿入され，関数定義ジェネ

レータからのデータが削除されます。関数定義ジェネレータは開いたままとなります。

2 408 ページ「関数の定義」で説明するように，新しい関数を定義します。

ユーザ定義関数の仕上げ

コードをプレビューしたら，関数をアクティブなドキュメントに挿入します。関数を関数ライブラリに挿入した場合，アプリケーション領域を通じて関数ライブラリに関連付けられているコンポーネントは，関数にアクセスできます。

コードを必要な場所に挿入した後，関数を仕上げることができます。たとえば，VerifyProperty 関数の場合，関数ライブラリに次のコードが挿入されます。

```
'@Description Checks whether a property matches its expected value
'@Documentation Check whether the <Test object name> <Test object type>
<prop_name> value matches the expected value: <expected_value>.
Public Function VerifyProperty (test_object, prop_name, expected_value)
    'TODO: add function body here
End Function
RegisterUserFunc "Link", "VerifyProperty", "VerifyProperty"
```

ヒント：RegisterUserFunc ステートメント（最後の行）が VerifyProperty 関数を Link テスト・オブジェクトに登録します。関数を複数のテスト・オブジェクトに登録するには，この行をコピーして各テスト・オブジェクト用に複製し，必要に応じて引数値を変更します。

関数を仕上げるには，その内容を追加します（TODO コメントを置き換えます）。たとえば，関数の中でプロパティの期待値が特定のテスト・オブジェクトの実際のプロパティ値と一致しているかどうかを検証するには，関数の本体に次のコードを付け加えます。

```
Dim actual_value
    ' 実際のプロパティ値を取得する
    actual_value = obj.GetROProperty(prop_name)
    ' 実際の値を期待値と比較する
    If actual_value = expected_value Then
        Reporter.ReportEvent micPass, "VerifyProperty Succeeded", "The " &
prop_name & " expected value: " & expected_value & " matches the actual
value"
        VerifyProperty = True
    Else
        Reporter.ReportEvent micFail, "VerifyProperty Failed", "The " &
prop_name & " expected value:" & expected_value & " does not match the actual
value: " & actual_value
        VerifyProperty = False
    End If
```

ユーザ定義関数を仕上げるには，次の手順を実行します。

1 ［**OK**］をクリックします。アクティブなドキュメントに関数定義が挿入され，関数定義ジェネレータが閉じます。

2 関数ライブラリの中で，必要に応じて TODO 行を置き換えて，関数コードに内容を追加します。

ヒント：実行セッション後に関数をテスト結果ツリー（［テスト結果］ウィンドウ）に表示するには，Reporter.ReportEvent ステートメントを関数コードに追加します（前出の例のように）。ユーザ定義関数の中で標準設定のテスト・オブジェクト・メソッドを使用している場合，このステップは実行セッション後に［テスト結果］ウィンドウに表示されます。ただし，Reporter.ReportEvent ステートメントを関数コードに追加して追加情報を提供したり，必要に応じてコンポーネントまたはビジネス・プロセス・テストのステータスを変更したりできます。

---

3 関数ライブラリをアプリケーション領域に関連付けて，そのユーザ定義関数にアクセスできるようにする必要があります。また，コードの構文を確認して，そのアプリケーション領域に関連付けられているコンポーネントがその関数にアクセスできることと，QuickTest エンジニアおよび各分野のエキスパートがその関数を表示し，使用できることを確認する必要もあります。詳細について

は，402 ページ「関連付けられている関数ライブラリを使用した作業」を参照してください。

ユーザ定義関数のテスト・オブジェクト・メソッドとしての登録

409 ページ「関数ジェネレータを使用した関数の登録」で説明した，QuickTest 関数定義ジェネレータを使用した関数の登録に加え，RegisterUserFunc ステートメントを使って，テスト・オブジェクトに新規メソッドを追加したり，実行セッション中に既存のテスト・オブジェクト・メソッドの振る舞いを変更したりできます。

関数をテスト・オブジェクトに登録するときに，必要に応じてその関数をテスト・オブジェクトの標準設定の操作として定義することができます。標準設定の操作は，その関数が登録されているテスト・オブジェクトが選択されたときに標準でキーワード・ビューの［**操作**］カラムに表示されます。

UnregisterUserFunc ステートメントを使用すれば，新規メソッドを無効にしたり，既存のメソッドを QuickTest の元々の振る舞いに戻したりできます。

関数をテスト・オブジェクトに登録しないと，この関数はグローバル関数になります。グローバル関数は，ステップ・ジェネレータの［**関数**］カテゴリ（関数ライブラリの場合），またはキーワード・ビューの［**操作**］項目を選択して，または IntelliSense の使用時に呼び出します。

メソッドを登録するには，まず関連関数ライブラリ内に関数を定義します。次に，関数の末尾に RegisterUserFunc ステートメントを挿入して，テスト・オブジェクト・クラス，使用する関数，および関数を呼び出すメソッド名を指定します。テスト・オブジェクト・クラスに新しいメソッドを追加することも，既存のメソッド名を使って，指定したメソッドの機能を（一時的に）オーバーライドすることもできます。

登録したメソッドは，メソッドを登録した関数ライブラリにのみ適用されます。また，QuickTest は，各実行セッションの開始時にすべての登録関数を消去します。

ユーザ定義関数の準備

登録されているメソッドを含んだステートメントを実行すると，ステートメントによって対象テスト・オブジェクトが最初の引数としてメソッドに送られます。したがって，ユーザ定義関数には少なくとも 1 つの引数がなければなりません。ユーザ定義関数は任意の数の引数を取ることができます。また，テスト・オブジェクト引数のみを取ることもできます。関数が既存のメソッドをオーバーライドする場合，その構文はオーバーライド対象の関数と正確に同じでなければなりません。つまり，最初の引数はテスト・オブジェクトで，残りの引数はすべて元のメソッド引数と一致することになります。

ヒント：parent 認識プロパティを使用して，関数の最初の引数で表されるオブジェクトの親を取得できます。例：ParentObj = obj.GetROProperty("parent")

自分で関数を書く場合には，標準の VBScript ステートメントに加え，QuickTest の任意の予約済みオブジェクト，メソッド，関数，および関数の最初の引数として渡されるテスト・オブジェクトに関連付けられている任意のメソッドが使えます。

関数がオーバーライド対象のテスト・オブジェクト・メソッドを呼び出す場合，そのメソッドの標準関数が使用されます。

たとえば，エディット・ボックスに新しい値を設定する前に現在の値を［テスト結果］に報告したいとします。その場合は，標準の QuickTest Set メソッドを，エディット・ボックスの現在の値を取得し，その値を［テスト結果］に報告してからエディット・ボックスに新しい値を設定する関数でオーバーライドします。

その関数は次のようになります。

```
Function MyFuncWithParam (obj, x)
    dim y
    y = obj.GetROProperty("value")
    Reporter.ReportEvent micDone, "previous value", y
    MyFuncWithParam=obj.Set (x)
End Function
```

注：この関数では戻り値を定義して，コンポーネントで使用されるたびに，**Set** メソッドの引数の値を返すようにしています。

ユーザ定義テスト・オブジェクト・メソッドの登録

RegisterUserFunc ステートメントを使って QuickTest に対して，コンポーネントの実行中，あるいはメソッドの登録を解除するまでの間，ユーザ定義関数を指定されたテスト・オブジェクト・クラスのメソッドとして使うように指示できます。

ユーザ定義関数をテスト・オブジェクト・メソッドとして登録するには，次の構文を使います。

RegisterUserFunc TOClass, MethodName, FunctionName, SetAsDefault

項目	説明
TOClass	任意のテスト・オブジェクト・クラス。
MethodName	登録するメソッドの名前（QuickTest 上で，たとえばキーワード・ビューや IntelliSense で表示されます）。特定のテスト・オブジェクト・クラスにすでに関連付けられているメソッドの名前を指定すると，ここで指定するユーザ定義関数が既存のメソッドをオーバーライドします。新しい名前を指定すると，オブジェクトがサポートするメソッドのリストにそのメソッドが追加されます。
FunctionName	コンポーネントから呼び出すユーザ定義関数の名前。この関数はコンポーネントのアプリケーション領域に関付けられている任意の関数ライブラリに置けます。
SetAsDefault	登録する関数を，テスト・オブジェクトの標準設定のメソッドとして使用するかどうかを示します。 キーワード・ビューでテスト・オブジェクトを選択すると，［**操作**］カラムに標準設定のメソッドが自動的に表示されます。

ヒント：関数ライブラリに RegisterUserFunc ステートメントを含めることをお勧めします。これにより，当該関数ライブラリを使用する任意のコンポーネントでメソッドをすぐに利用できるようになります。

たとえば，Find Flights Web ページに［**Country**］エディット・ボックスが含まれており，標準設定でこのボックスに「USA」という値が含まれているとします。次の例では，エディット・ボックスの標準設定の値を新しい値が入力される前に取得するために，MySet 関数を使用するように Set メソッドを登録しています。

```
Function MySet (obj, x)
    dim y
    y = obj.GetROProperty("value")
    Reporter.ReportEvent micDone, "previous value", y
    MySet=obj.Set(x)
End Function

RegisterUserFunc "WebEdit", "Set", "MySet"
Browser("MercuryTours").Page("FindFlights").WebEdit("Country").Set "Canada"
```

詳細と例については，『**HP QuickTest Professional Object Model Reference**』（英語版）を参照してください。

ユーザ定義テスト・オブジェクト・メソッドの登録解除

RegisterUserFunc ステートメントを使ってメソッドを登録すると，そのメソッドはコンポーネントによって使用されている間，または登録解除されるまで指定のテスト・オブジェクトのメソッドとして認識されます。このメソッドが QuickTest のメソッドをオーバーライドしている場合，このメソッドの登録を解除すると，メソッドは通常の動作に戻ります。ほかのメソッドの登録を解除すると，テスト・オブジェクトによってサポートされているメソッドのリストから，それらを削除することになります。

ユーザ定義メソッドの登録を解除するには，次の構文を使用します。

UnRegisterUserFunc TOClass, MethodName

項目	説明
TOClass	メソッドが登録されているテスト・オブジェクト・クラス。
MethodName	登録を解除するメソッド。

たとえば，Find Flights Web ページに［**Country**］エディット・ボックスが含まれており，標準設定でこのボックスに「USA」という値が含まれているとします。次の例では，エディット・ボックスの標準設定の値を新しい値が入力される前に取得するために，MySet 関数を使用するように Set メソッドを登録しています。［**Country**］エディット・ボックスのための WebEdit.Set ステートメントの中で，登録されたメソッドを使用した後，Set メソッドを標準の機能に戻すために UnRegisterUserFunc ステートメントを使用しています。

```
Function MySet (obj, x)
    dim y
    y = obj.GetROProperty("value")
    Reporter.ReportEvent micDone, "previous value", y
    MySet=obj.Set(x)
End Function

RegisterUserFunc "WebEdit", "Set", "MySet"
Browser("MercuryTours").Page("FindFlights").WebEdit("Country").Set "Canada"
UnRegisterUserFunc "WebEdit", "Set"
```

ユーザ定義関数の使い方のヒント

ユーザ定義関数を使用するときには，次のヒントとガイドラインを考慮してください。

- 関数定義ジェネレータを使って関数を定義し，さまざまなオプションを試してみれば，必要な構文について詳しい知識を得られます。
- 関数は，登録されると，テスト・オブジェクト・クラス全体に適用されます。特定のテスト・オブジェクトに限定してメソッドを登録することはできません。
- その他のテスト・オブジェクトから関数を呼び出す場合は，RegisterUserFunc 行をコピーして，別の関数の直後に貼り付け，適切な引数値を指定します。
- 関数ライブラリに RegisterUserFunc ステートメントを含めることをお勧めします。これにより，当該関数ライブラリを使用する任意のコンポーネントでメソッドをすぐに利用できるようになります。
- コンポーネントに関連付けられている関数ライブラリ内で Option Explicit ステートメントを使用するには，ステートメントをコンポーネントに関連付けられているすべての関数ライブラリに含める必要があります。関連付けられている関数ライブラリの一部にのみ Option Explicit ステートメントを含めた場合，すべての関数ライブラリ内のすべての Option Explicit ステートメントが無視されます。
- 各関数ライブラリのグローバル・スコープにある変数は，一意でなければなりません。2 つの関連付けられている関数ライブラリにおいて，Dim ステートメントを使用してグローバル・スコープ内で同じ変数を定義している場合，または同じ名前を持つ 2 つの定数を定義している場合，2 番目の定義によって構文エラーが発生します。グローバル・スコープにおいて同じ名前を持つ 2 つ以上の変数を使用する必要がある場合，（関数ライブラリは逆順で読み込まれるため）最後の関数ライブラリにのみ Dim ステートメントを挿入します。
- 標準設定では，ユーザ定義関数を使用するステップは，実行セッション後に［テスト結果］ウィンドウのテスト結果ツリーに表示されません。関数がテスト結果ツリーに表示されるようにするには，Reporter.ReportEvent ステートメントを関数コードに付け加える必要があります。たとえば，必要に応じて追加情報を提供したり，コンポーネントのステータスを変更したりすることが考えられます。
- 使用されている関数を関連付けられている関数ライブラリから削除すると，その関数を使用しているコンポーネント・ステップは アイコンで示されます。その後，そのコンポーネントまたはビジネス・プロセス・テストの実行セッ

ションで，存在しない関数を使用しているステップに達すると，エラーが発生します。

- コンポーネントが参照する関数ライブラリをほかのユーザが変更した場合，または，QuickTest エンジニアが外部エディタ（QuickTest 以外）を使用して関数ライブラリを変更した場合，変更は，コンポーネントを再度開くまで反映されません。
- 同じ名前を持つ 2 つ以上の関数が関数ライブラリに存在する場合，必ず最後の関数が呼び出されます。混乱を避けるために，1 つのアプリケーション領域またはコンポーネントに関連付けられているリソースの中では，それぞれの関数に一意の名前を付けてください。
- 先にメソッドを登録解除しなくても，一度登録したメソッドが異なるユーザ定義関数を使うように登録しなおすことが可能です。ただし，このメソッドの登録を解除すると，QuickTest の本来の動作に戻り（あるいは，新規のメソッドだった場合には完全に消去され），それ以前に登録されていたものには戻りません。

  たとえば，次のステートメントを入力するとします。

  ```
  RegisterUserFunc "Link", "Click", "MyClick"
  RegisterUserFunc "Link", "Click", "MyClick2"
  UnRegisterUserFunc "Link", "Click"
  ```

  UnRegisterUserFunc ステートメントを実行した後，Click メソッドは MyClick2 関数で定義されている機能を使うのを止め，QuickTest の元の Click 機能に戻り，MyClick 関数で定義されている機能には戻りません。
- VBScript を使った関数やサブルーチンの作成の詳細については，QuickTest の［**ヘルプ**］メニューから VBScript に関するドキュメントを参照してください（［**ヘルプ**］＞［**QuickTest Professional ヘルプ**］＞［**VBScript Reference**］）。

第 IV 部

アプリケーション領域を使った作業

第 13 章

アプリケーション領域の管理

アプリケーション領域は，ビジネス・コンポーネントの作成に必要なリソースと設定をすべて提供します。アプリケーション領域の設定と設定変更は，そのアプリケーション領域に関連付けられたすべてのビジネス・コンポーネントに自動的に適用されます。

注：旧バージョンの QuickTest Professional では，アプリケーション領域はビジネス・コンポーネント・テンプレートと呼ばれていました。旧バージョンでは，すべてのビジネス・コンポーネントで同じテンプレートを使用していました。現在の QuickTest では，複数のアプリケーション領域を作成して，それをアプリケーションの各領域の要件に合わせてカスタマイズできます。

本章の内容

- アプリケーション領域の使用について（430 ページ）
- アプリケーション領域の作成（433 ページ）
- アプリケーション領域を開く（436 ページ）
- アプリケーション領域の保存（439 ページ）
- アプリケーション領域の削除（443 ページ）

アプリケーション領域の使用について

アプリケーションの特定領域をテストするために一連のコンポーネントを作成するときは，同じテスト・オブジェクト，キーワード，テスト設定，およびその他のテスト・リソース（関数ライブラリや回復シナリオなど）を多数使用する必要があります。アプリケーション領域にこれらのファイルや設定を定義することにより，アプリケーションの特定部分のテストに関連するすべての要素を 1 か所で保守できるようになります。

「**アプリケーション領域**」は，ビジネス・コンポーネントの内容を作成するのに必要な設定とリソースの集まりです。リソースには共有オブジェクト・リポジトリや関数ライブラリを含めることができます。共有オブジェクト・リポジトリには，コンポーネントのテスト対象アプリケーション内のテスト・オブジェクトが収められます。関数ライブラリには，そのアプリケーションを対象としてコンポーネントによって実行されるユーザ定義の操作が収められます。コンポーネントは，関連付けられたアプリケーション領域で定義されているすべてのリソースと設定に自動的にリンクされます。

アプリケーション領域は，必要に応じていくつでも作成できます。たとえば，アプリケーション内の Web ページ，モジュール，ウィンドウ，またはダイアログ・ボックスそれぞれについて，アプリケーション領域を 1 つずつ作成することもできます。また，小さなアプリケーションでは，1 つのアプリケーション領域で十分な場合もあります。各コンポーネントに関連付けることができるアプリケーション領域は 1 つだけです。

注 : アプリケーション領域を使用するには，コンポーネントの変更とステップの追加，変更，削除に必要な権限を持っている必要があります。4 つの権限すべてが必要です。これらの権限のいずれかが割り当てられていない場合は，アプリケーション領域を読み取り専用形式でしか開くことができません。ビジネス・コンポーネント・モジュールにおける権限の設定の詳細については，『**HP Business Process Testing ユーザーズ・ガイド**』を参照してください。

アプリケーション領域の計画

アプリケーション領域を作成する前に，そのアプリケーション領域を使ってビジネス・コンポーネントを作成する各分野のエキスパートの要件を検討します。たとえば，次のような場合です。

- 各分野のエキスパートはどのようなテスト・オブジェクトを必要としていますか。
- 誰にでも意味がよくわかるように，テスト・オブジェクトやほかの項目の名前をどのように変更しますか。
- 必要な操作をすべて提供するために，どのようなユーザ定義関数を追加できますか。

必要な操作を確実に提供するために，アプリケーション領域を作成する*前に*，関数ライブラリを Quality Center プロジェクトに保存することをお勧めします（アプリケーション領域はいつでも更新できます）。QuickTest は，アプリケーション領域に関連付けることができる一連の定義済みリソース・ファイル（関数ライブラリや回復シナリオ・ファイルなど）も提供します。サンプル関数ライブラリの一部は，標準ですべての新しいアプリケーション領域に関連付けられます。これらのサンプル・ファイルは，Quality Center プロジェクトのテスト計画モジュールの **Resources/BPT Resources** の下にあります。

アプリケーション領域の作成

コンポーネントに使用するアプリケーション領域を作成するときは，次の作業を行う必要があります。

- 433 ページ「アプリケーション領域の作成」に記述されているとおり，基本のアプリケーション領域を作成する。
- 446 ページ「一般設定の定義」に記述されているとおり，アプリケーション領域の詳細な説明を入力する。
- 446 ページ「一般設定の定義」に記述されているとおり，QuickTest Professional アドインへの関連付けを指定する
- 460 ページ「関数ライブラリの関連付け」に記述されているとおり，任意の必要な関数ライブラリを関連付ける。
- 465 ページ「共有オブジェクト・ライブラリの関連付け」に記述されているとおり，任意の必要な共有オブジェクト・リポジトリを関連付ける。

- 472 ページ「使用可能なキーワードの指定」に記述されているとおり，コンポーネント・ステップの作成時に，各分野のエキスパート に対してどのキーワードを表示し，利用可能にするかを指定する。
- 452 ページ「追加設定の定義」に記述されているとおり，アプリケーション領域に関連付けられたコンポーネントが記録と実行の対象にできる Windows ベース・アプリケーションを指定する。
- 452 ページ「追加設定の定義」に記述されているとおり，必要な回復シナリオを関連付け，それらの設定を定義する。
- 439 ページ「アプリケーション領域の保存」に記述されているとおり，アプリケーション領域を保存する。

  アプリケーション領域を保存するときには，わかりやすい名前と明確な説明を付けて保存してください。各分野のエキスパートが新規のビジネス・コンポーネントを作成するときに，アプリケーション領域の用途を判断するには，この名前と説明だけが手がかりとなります。たとえば，アプリケーション領域がログイン・ダイアログ・ボックスをテストするコンポーネント用である場合は，「LoginDialog」のような名前を付けます。

アプリケーション領域の使用

アプリケーション領域を作成したら，そのアプリケーション領域を使ったビジネス・コンポーネントの作成を開始できるように，各分野のエキスパートに通知します（各分野のエキスパートは，必要に応じてアプリケーション領域の準備ができる前にコンポーネントの作成を開始し，後でアプリケーション領域をコンポーネントに関連付けることもできます)。

アプリケーション領域内のリソースや設定を変更すると，変更されたアプリケーション領域に関連付けられたすべてのビジネス・コンポーネントに自動的に変更が適用されます。

コンポーネント・ステップでリソースを使用し，後でそれらのリソースを変更すると，コンポーネントが正しく実行されない場合があります。たとえば ，コンポーネントが **MyRepository.tsr** 共有オブジェクト・リポジトリのテスト・オブジェクトを使用している場合に，このオブジェクト・リポジトリをアプリケーション領域から削除すると，このオブジェクト・リポジトリがアプリケーション領域に含まれなくなるため，コンポーネントが必要なテスト・オブジェクトにアクセスできなくなります。

このため，アプリケーション領域に対して行った変更が，そのアプリケーション領域が関連付けられたビジネス・コンポーネントに悪影響を与えないかどうかを確認することをお勧めします。

ヒント：コンポーネントは，いつでも別のアプリケーション領域に関連付けることができます。詳細については，503 ページ「コンポーネントに関連付けらているアプリケーション領域の変更」を参照してください。

アプリケーション領域の作成

新しいアプリケーション領域を作成するときは，新しいビジネス・コンポーネントを作成するのに必要なアプリケーション領域の設定とリソースをすべて定義します。

新規のアプリケーション領域を作成するには次の手順を実行します。

1 **ビジネス・コンポーネントの保存先となる Quality Center プロジェクトに接続します。**

これは，各分野のエキスパートがビジネス・コンポーネントの定義やビジネス・プロセス・テストの作成に使用する Quality Center プロジェクトです。詳細については，46 ページ「Quality Center プロジェクトへの接続」を参照してください。

2 **［ファイル］＞［新規作成］＞［アプリケーション領域］を選択し，アプリケーション領域ウィンドウを開きます。**

アプリケーション領域ウィンドウには，そのアプリケーション領域に関連付けられたビジネス・コンポーネントで使用する設定とリソース・ファイルを指定できる複数の表示枠があります。コンポーネントをアプリケーション領域に関連付けると，そのコンポーネントはそれらの設定とリソース・ファイルに自動的に結び付けられます。

アプリケーション領域ウィンドウの詳細については，434 ページ「アプリケーション領域ウィンドウ」を参照してください。

3 **アプリケーション領域を指定し，リソースを定義します。**

各表示枠で利用可能なオプションの詳細については，435 ページ「アプリケーション領域表示枠」を参照してください。

4 アプリケーション領域を保存します。

アプリケーション領域を保存する際，固有の名前を割り当て，詳細を追加します。これにより各分野のエキスパートはコンポーネントに対する正しいアプリケーション領域を選択することができます。詳細については，439 ページ「アプリケーション領域の保存」を参照してください。

アプリケーション領域ウィンドウ

詳細	新しいビジネス・コンポーネントを作成するのに必要な，アプリケーション領域のすべての設定とリソースを定義し，表示することができます。
アクセス方法	Quality Center プロジェクトに接続します。次のいずれかを実行します。 ➤［**ファイル**］＞［**新規作成**］＞［**アプリケーション領域**］メニュー・コマンドを選択します。 ➤［**新規作成**］ボタンの下向き矢印をクリックし，［**アプリケーション領域**］を選択します。 ➤ CTRL + ALT + N キーを押します。 **ヒント**：アプリケーション領域がすでに開いている場合は，［**新規作成**］ボタンをクリックすると新しいアプリケーション領域が開きます。
さらに調べる	**概念の概要**：430 ページ「アプリケーション領域の使用について」 **主要なタスク**：433 ページ「アプリケーション領域の作成」 **追加の関連トピック**： ➤ 436 ページ「アプリケーション領域を開く」 ➤ 439 ページ「アプリケーション領域の保存」 ➤ 443 ページ「アプリケーション領域の削除」

アプリケーション領域表示枠

アプリケーション領域には，次の表示枠が含まれます。各表示枠にアクセスするには，サイドバーの該当するボタンをクリックします。

表示枠	内容
[一般]	アプリケーション領域の説明を定義し，アプリケーション領域の関連アドインを指定できます。 また，次の操作もできます。 ➤ Windows ベースのオブジェクトを含むアプリケーションを指定する。アプリケーション領域に関連付けられたコンポーネントが使用できるアプリケーションです。 ➤ ブラウザのタイムアウト期間を設定する。 ➤ アプリケーション領域に関連付けられたコンポーネントが実行セッション中の予期しないイベントやエラーからどのように回復するかを指定した回復シナリオを定義する。詳細については，446 ページ「一般設定の定義」を参照してください。
[関数ライブラリ]	関数ライブラリをアプリケーション領域に関連付け，各ライブラリの優先順位を設定できます。また，関連付けられた関数ライブラリを作成および変更できます。詳細については，460 ページ「関数ライブラリの関連付け」を参照してください。
[オブジェクト リポジトリ]	共有オブジェクト・リポジトリをアプリケーション領域に関連付け，各リポジトリの優先順位を設定できます。また，関連付けられたオブジェクト・リポジトリを作成および変更できます。詳細については，465 ページ「共有オブジェクト・ライブラリの関連付け」を参照してください。
[キーワード]	各分野のエキスパートがコンポーネントの作成時に，どの組み込みおよびユーザ定義のキーワード（操作）を使用できるようにするかを決定できます。詳細については，472 ページ「使用可能なキーワードの指定」を参照してください。

アプリケーション領域を開く

アプリケーション領域を保存した後，表示や変更のために開くことができます。たとえば，回復シナリオを更新したり，ユーザ定義関数を含む関数ライブラリをアプリケーション領域に追加したりできます。

注：QuickTest の古いバージョンを実行するコンピュータ上で，QuickTest の新しいバージョンで作成されたアプリケーション領域を開くことはできません。たとえば，QuickTest 8.2 を実行するコンピュータ上で，QuickTest 10.00 で作成されたアプリケーション領域を開くことはできません。

アプリケーション領域を開くには，次の手順を実行します。

1 ビジネス・コンポーネントの保存先となる Quality Center プロジェクトに接続します。

詳細については，46 ページ「Quality Center プロジェクトへの接続」を参照してください。

2［ファイル］＞［開く］＞［アプリケーション領域］を選択し，アプリケーション領域ウィンドウを開きます。

詳細については，437 ページ「[アプリケーション領域を開く］ダイアログ・ボックス」を参照してください。

3［アプリケーション領域を開く］ダイアログ・ボックスで，アプリケーション領域を選択し，［OK］をクリックします。

選択したアプリケーション領域が開きます。これで，アプリケーション領域の設定を表示して変更できるようになります。詳細については，次を参照してください。

- 446 ページ「一般設定の定義」
- 460 ページ「関数ライブラリの関連付け」
- 465 ページ「共有オブジェクト・ライブラリの関連付け」
- 472 ページ「使用可能なキーワードの指定」
- 452 ページ「追加設定の定義」

注：このダイアログ・ボックスでアプリケーション領域を削除することもできます（どのコンポーネントにも関連付けられていない場合のみ）。詳細については，443 ページ「アプリケーション領域の削除」を参照してください。

［アプリケーション領域を開く］ダイアログ・ボックス

記述	定義済みアプリケーション領域のリストを表示します。 ［アプリケーション領域を開く］ダイアログ・ボックスで次のことができます。 ➤ アプリケーション領域を選択すると，その記述が表示されます。 ➤ アプリケーション領域を選択し，［**OK**］をクリックすることでアプリケーション領域が開きます。 ➤ アプリケーション領域を選択し，［**OK**］をクリックすることでアプリケーション領域を削除します。
アクセス方法	Quality Center プロジェクトに接続します。次のいずれかを実行します。 ➤［**ファイル**］＞［**開く**］＞［**アプリケーション領域**］メニュー・コマンドを選択します。 ➤［**新規作成**］ボタンの下向き矢印をクリックし，［**アプリケーション領域**］を選択します。 ➤ CTRL + ALT + N キーを押します。 **ヒント**：ほかのアプリケーション領域がすでに開いている場合は，［**開く**］ボタンをクリックして必要なアプリケーション領域を選択します。

重要な情報	［アプリケーション領域を開く］ダイアログ・ボックスを使用して，アプリケーション領域を開く，または削除します。
さらに調べる	**概念の概要**：430 ページ「アプリケーション領域の使用について」 **主要なタスク**：436 ページ「アプリケーション領域を開く」 **追加の関連トピック**：443 ページ「アプリケーション領域の削除」

下記が［アプリケーション領域を開く］ダイアログ・ボックスの画像です

［アプリケーション領域を開く］ダイアログ・ボックスのオプション

オプション	説明
［アプリケーション領域名］	Quality Center プロジェクトの定義されたすべてのアプリケーション領域が一覧表示されます。

オプション	説明
［**削除**］ボタン	選択したアプリケーション領域を削除できます。詳細については，443 ページ「アプリケーション領域の削除」を参照してください。
［**記述**］	選択したアプリケーション領域の説明を表示します。説明は以下の時点で入力されます。 ➤ アプリケーション領域が［アプリケーション領域］ウィンドウの［一般］表示枠で作成された時。詳細については，433 ページ「アプリケーション領域の作成」を参照してください。 ➤ アプリケーション領域が保存された時。詳細については，439 ページ「アプリケーション領域の保存」を参照してください。

アプリケーション領域の保存

アプリケーション領域は，作成した後，その設定とリソースを定義する前後で保存できます。

アプリケーション領域を保存するときには，一意の名前と，用途を明確に示す説明を付けてください。たとえば，ログイン・モジュールをテストするコンポーネントに使用するアプリケーション領域であれば，「Log In」という名前を付けて，「**ログイン・モジュールをテストするビジネス・コンポーネントで使用する**」のように用途を示す説明を追加します。

アプリケーション領域を保存するには，次の手順を実行します。

1 **QuickTest で，Business Process Testing のサポートが組み込まれた Quality Center サーバおよびプロジェクトに接続します。**

詳細については，46 ページ「Quality Center プロジェクトへの接続」を参照してください。

2 **アプリケーション領域を作成し，必要に応じてその設定を変更します。**

詳細については，433 ページ「アプリケーション領域の作成」を参照してください。

3 **［保存］をクリックするか，［ファイル］＞［保存］を選択します。**

［アプリケーション領域の保存］ダイアログ・ボックスが開きます。

ヒント：既存のアプリケーション領域に似た新しいアプリケーション領域を作成する場合は，［**名前を付けて保存**］オプションを利用できます。その後，必要に応じてアプリケーション領域に変更を加えることができます。

---

4 必要な情報を入力し，［OK］をクリックしてアプリケーション領域を保存します。

このステップでは，各分野のエキスパートがコンポーネントのアプリケーション領域を選択するときに使用する，一意の名前と説明を設定します。名前は特定の命名規則に従う必要があります。詳細については，441 ページ「［アプリケーション領域の保存］ダイアログ・ボックス」を参照してください。

［アプリケーション領域の保存］ダイアログ・ボックス

記述	アプリケーション領域を保存できます。
アクセス方法	➤［**ファイル**］＞［**保存**］メニュー・コマンドを選択します。 ➤［**保存**］ボタン をクリックします。 ➤ CTRL+S キーを押します。
さらに調べる	**概念の概要**：430 ページ「アプリケーション領域の使用について」 **主要なタスク**：439 ページ「アプリケーション領域の保存」 **追加の関連トピック**： ➤ 434 ページ「アプリケーション領域ウィンドウ」 ➤ 437 ページ「[アプリケーション領域を開く］ダイアログ・ボックス」

下記が［アプリケーション領域を開く］ダイアログ・ボックスの画像です

［アプリケーション領域を開く］ダイアログ・ボックスのオプション

オプション	［記述］
既存のアプリケーション領域	Quality Center プロジェクトの定義されたすべてのアプリケーション領域が一覧表示されます。この一覧で既存のアプリケーション領域の名前を確認すれば，保存するアプリケーション領域に一意の名前を付けられます。
名前	アプリケーション領域の名前を示します。各分野のエキスパートがそのアプリケーション領域が自分のコンポーネントに適しているかどうかをすばやく判断できるように，わかりやすい名前を入力します。 入力する名前は次のルールに従います。 ➤ このプロジェクト内で一意であること ➤ 220 文字を超えないこと ➤ 先頭，または終わりに空白を使用しないこと ➤ 以下の文字を含めないこと ¥ / : " ? < > | * ! { } &slq; % ➤ 連続する 2 つのセミコロンを含めないこと（;;）
記述	アプリケーション領域を作成するときに［アプリケーション領域］ダイアログ・ボックスの［一般］表示枠で入力した説明が表示されます。詳細については，433 ページ「アプリケーション領域の作成」を参照してください。 アプリケーション領域を作成するときに説明を入力しなかった場合は，ここで入力する必要があります。説明を入力せずにアプリケーション領域を保存することはできません。 ［一般］表示枠ですでに説明を定義していた場合は，既存の説明を変更することもできます。この説明により，各分野のエキスパートはさまざまなアプリケーション領域を簡単に識別して各自のコンポーネントに最適なアプリケーション領域を選択できるようになります。

アプリケーション領域の削除

不要になったアプリケーション領域は削除できます。アプリケーション領域を削除する前に，次のことを確認します。

- アプリケーション領域はどのコンポーネントとも関連付けされていない。
- アプリケーション領域は現在どのコンピュータ上でも開いていない。

アプリケーション領域を削除するには，次の手順を実行します。

1 **QuickTest で，削除するアプリケーション領域が保存されている Quality Center プロジェクトに接続します。**

詳細については，46 ページ「Quality Center プロジェクトへの接続」を参照してください。

2 **［ファイル］＞［開く］＞［アプリケーション領域］を選択し，［アプリケーション領域を開く］ダイアログ・ボックスを開きます。**

詳細については，437 ページ「［アプリケーション領域を開く］ダイアログ・ボックス」を参照してください。

3 **削除するアプリケーション領域を選択して［アプリケーション領域の削除］ボタンをクリックします。**

警告メッセージが表示されます。

注：現在開いているアプリケーション領域，別のオートメーション・エンジニアが現在使用しているアプリケーション領域，およびコンポーネントに関連付けられているアプリケーション領域は削除できません。

---

4 **［はい］ボタンをクリックして，確定します。**

選択したアプリケーション領域が削除されます。

5 **［OK］をクリックして，［アプリケーション領域を開く］ダイアログ・ボックスを閉じます。**

第 14 章

アプリケーション領域の設定

アプリケーション領域の作成後，設定を行います。これには，たとえば，一般設定の定義，リカバリ・シナリオの関連付け，関数ライブラリや共有オブジェクト・リポジトリなどのリソース・ファイルの関連付け，関連するコンポーネントで利用可能なキーワード（操作）の指定などがあります。

本章の内容

- 一般設定の定義（446 ページ）
- 追加設定の定義（452 ページ）
- 関数ライブラリの関連付け（460 ページ）
- 共有オブジェクト・ライブラリの関連付け（465 ページ）
- 使用可能なキーワードの指定（472 ページ）

一般設定の定義

［一般］表示枠では，説明や関連付けられているアドインなど，アプリケーション領域に関する一般情報の表示と定義ができます。アプリケーション領域の場合，各分野のエキスパートが特定のビジネス・コンポーネントに対してどのアプリケーション領域を選択するかを決定するときに得られる情報は名前と説明だけなので，明確な説明を含めることが重要です。

［一般］表示枠には，次の項目があります。

項目	説明
［名前］	アプリケーション領域の名前を示します。アプリケーション領域の名前は，保存するときに指定します。詳細については，439 ページ「アプリケーション領域の保存」を参照してください。
［作成者］	アプリケーション領域を作成した人物の Windows ユーザ名を示します。

項目	説明
［場所］	アプリケーション領域の Quality Center パスとファイル名を示します。アプリケーション領域がまだ保存されていない場合は，［**保存されていません**］と表示され，［アプリケーション領域］ダイアログ・ボックスのタイトル・バーにアスタリスクが表示されます。
［記述］	アプリケーション領域に対して指定されている説明を示します。 この必須フィールドにアプリケーション領域の明確な説明を含めることが重要です。これは，各分野のエキスパートが Quality Center で新しいコンポーネントを作成するときに，アプリケーション領域の［**名前**］と［**記述**］に基づいてどのアプリケーション領域を選択するかを決めるからです。詳細については，『**HP Business Process Testing ユーザーズ・ガイド**』を参照してください。 この説明は必要に応じて更新できます。たとえば，アプリケーション領域を作成したが，まだ定義を完了していないときに，［**記述**］領域に更新の必要性を記述しておきます。その後，アプリケーション領域が完成した後で，［**記述**］を更新できます。 **注**：［一般］表示枠で説明を入力しなかった場合は，アプリケーション領域を保存するときに説明の入力を求められます。詳細については，439 ページ「アプリケーション領域の保存」を参照してください。

項目	説明
［関連アドイン］	アプリケーション領域に関連付けられているアドインが表示されます。関連アドインとは，ビジネス・コンポーネントがアクセスされたときに QuickTest によってロードされるアドインです。 **注**：ビジネス・プロセス・テストを実行すると，QuickTest はテストの最初のコンポーネントに関連付けられたアドインをロードします。したがって，必要なすべての QuickTest アドインをビジネス・プロセス・テストの最初のコンポーネントのアプリケーション領域に確実に関連付けることが重要です。 アドインの関連付けの詳細については，450 ページ「コンポーネントへのアドインの関連付け」を参照してください。 QuickTest のアドイン環境設定の詳細については，**『HP QuickTest Professional アドイン・ガイド』**を参照してください。

項目	説明
［**追加設定**］ボタン	［アプリケーション領域の設定］ダイアログ・ボックス（ 452 ページを参照）を開きます。このダイアログ・ボックスは 2 つの部分に分かれています。左側のナビゲーション枠と右側の設定の表示枠です。次の表示枠は，ナビゲーション・ツリーで対応するノードをクリックした際に表示されます。 ➤［**アプリケーション**］。アプリケーション領域に関連付けられたコンポーネントが記録と実行の対象にできる Windows ベース・アプリケーションを指定できます。詳細については，453 ページ「アプリケーション領域のアプリケーション設定の定義」を参照してください。 ➤［**回復**］。アプリケーション領域に関連付けられたコンポーネントが実行セッション中のテスト環境で発生した予期しないイベントやエラーからどのように回復するかを定義できます。詳細については，456 ページ「アプリケーション領域の回復シナリオ設定の定義」を参照してください。 ナビゲーション・ツリーには，ロードされている QuickTest アドイン（Web，Java，SAP など）に対応する追加のノードが含まれる場合もあります。アドインに関連するノードの詳細については，『**HP QuickTest Professional アドイン・ガイド**』を参照してください。
［**変更**］ボタン	［関連アドインの変更］ダイアログ・ボックスを開きます。このダイアログ・ボックスを使って，アドインをコンポーネントに関連付けたり，関連付けを削除したりできます。変更を有効にするために，QuickTest の再起動が必要な場合があります。詳細については，450 ページ「コンポーネントへのアドインの関連付け」を参照してください。

注：アプリケーション領域を開いたときに［一般］表示枠の全体が表示されない場合は，表示枠のサイズを変更できます。たとえば，関連アドインより下の領域が情報表示枠に隠れる場合は，情報表示枠のサイズを変更できます。

コンポーネントへのアドインの関連付け

QuickTest の起動時に［アドイン マネージャ］ダイアログ・ボックスでロードするアドインを選択します。必要なアドインがロードされている任意の環境での記録が可能です。

アドインをアプリケーション領域に関連付けると，そのアプリケーション領域に関連付けられたコンポーネントを開くたびに関連アドインがロードされているかどうかをチェックするように QuickTest が設定されます。コンポーネントを新規作成するときは，コンポーネントのアプリケーション領域で定義されているアドインが関連アドインになります。

コンポーネントを開いたときに，関連アドインが現在ロードされていない場合や，（アプリケーション領域を介して）コンポーネントに現在関連付けられていないアドインがロードされている場合は，そのことが通知されます。この通知処理によって，現在開いているコンポーネントで使用するアドインを関連アドインのリストに必須アドインとして追加し，アドインがロードされていないために発生する実行セッションの失敗を確実に避けることができます。

各分野のエキスパートが Quality Center でビジネス・プロセス・テストを開くと，そのビジネス・プロセス・テストの最初のコンポーネントに関連付けられている QuickTest Professional アドインが自動的にロードされます。ビジネス・プロセス・テスト内のほかのコンポーネントに関連付けられたアドインはロードされません。したがって，必要なすべての QuickTest アドインをビジネス・プロセス・テストの最初のコンポーネントのアプリケーション領域に確実に関連付けることが重要です。

関連アドインの変更

アドインをアプリケーション領域（およびアプリケーション領域に関連付けられたコンポーネント）に関連付けたり，関連付けを解除したりするには，［一般］表示枠の［**変更**］ボタンをクリックします。［関連アドインの変更］ダイアログ・ボックスが開きます。

このダイアログ・ボックスには，アプリケーション領域に現在関連付けられているすべてのアドインが，現在 QuickTest にロードされているその他のアドインとともに一覧表示されます。アプリケーション領域に関連付けられているが，現在ロードされていないアドインは，グレーで表示されます。

注：このリストには，アドイン拡張を使って追加の環境やコントロールをサポートするために自分または第三者が開発したアドインを表す子ノードが含まれることもあります。詳細については，該当するアドイン拡張開発者ガイド（拡張のセットアップによって使用可能になる）を参照してください。

アプリケーション領域に関連付けるアドインのチェック・ボックスを選択し，アプリケーション領域に関連付けないアドインのチェック・ボックスをクリアします。［関連アドインの変更］ダイアログ・ボックスに子アドインが含まれていて，それを選択した場合は，親アドインが自動的に選択されます。親アドインのチェック・ボックスをクリアすると，その子アドインのチェック・ボックスもクリアされます。

上の例では次のようになっています。

- Web アドインは，ロードされており，アプリケーション領域に関連付けられている。
- ActiveX アドインは，ロードされているが，アプリケーション領域には関連付けられていない。
- Visual Basic アドインは，アプリケーション領域に関連付けられているが，ロードされていない。

注：現在ロードされていないアドインをアプリケーション領域に関連付けるには，QuickTest を再起動し，アドイン・マネージャでそのアドインをロードします。QuickTest の起動時に［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は，QuickTest の次回起動時に表示されるように設定できます。それには，［オプション］ダイアログ・ボックスの［一般］表示枠で，［**起動時にアドイン マネージャを表示する**］を選択します。詳細については，629 ページ「グローバル・テスト・オプションの設定」を参照してください。

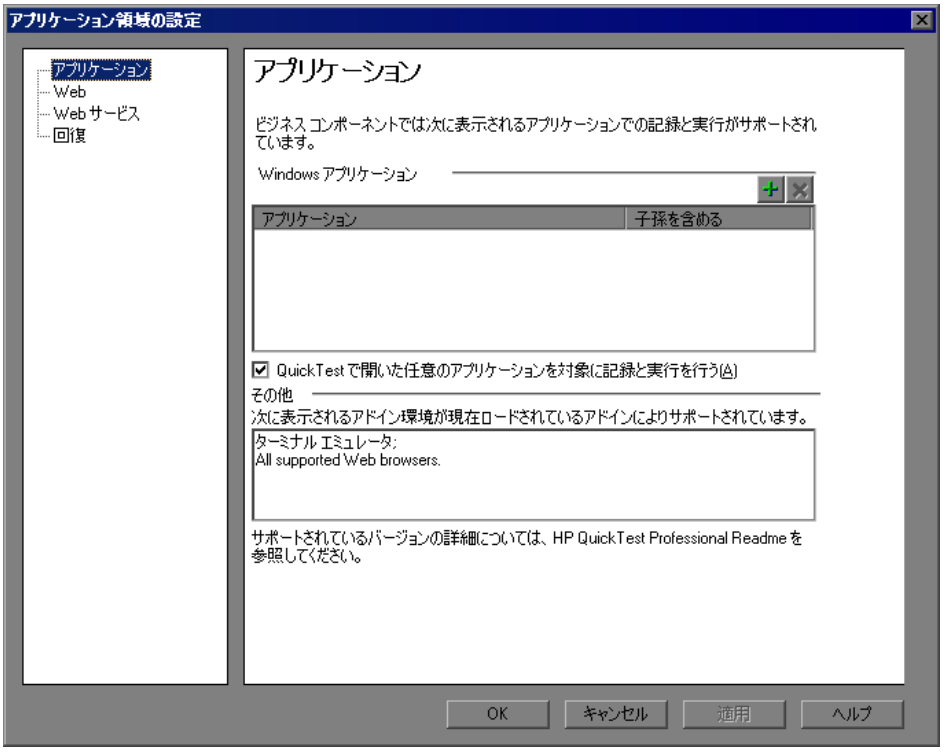
追加設定の定義

［一般］表示枠の［**追加設定**］ボタンをクリックすると，2 つの部分で構成される［アプリケーション領域の設定］ダイアログ・ボックスが開きます。左側のナビゲーション枠と右側の設定の表示枠です。ナビゲーション・ツリーから必要なノードを選択し，設定の表示枠で必要に応じてオプションを設定します。これらの表示枠を使って，アプリケーション領域に関連付けられたコンポーネントが記録と実行の対象にできるアプリケーションや，コンポーネントが実行セッション中の予期しないイベントからどのように回復するかなど，アプリケーション領域に関する特定の設定を定義できます。

注：ナビゲーション・ツリーには，ロードされているアドイン（Web，SAP，Web サービスなど）に対応する追加のノードが含まれる場合もあります。アドインに関連するノードの詳細については，該当する『**HP QuickTest Professional アドイン・ガイド**』を参照してください。

アプリケーション領域のアプリケーション設定の定義

［アプリケーション］表示枠では，現在のアプリケーション領域に関連付けられたコンポーネントが記録と実行の対象にできる Windows ベース・アプリケーションを指定できます。指定したアプリケーションが対象の場合にのみコンポーネント・ステップを記録できます。



ヒント：アプリケーションを記録するには，そのアプリケーションを手動で開くか，QuickTest の **Common.txt** 関数ライブラリで提供される OpenApp キーワード（関数）を使用します。コンポーネント用にアプリケーションを自動的に開くための設定はありません。

次のような場合に，［アプリケーション］表示枠を使ってアプリケーションにかかわる設定や変更が行えます。

- 関連コンポーネント内の 1 つ以上のステップを記録済みで，記録を続行する前に設定を変更したい。
- 以前に使用したものとは異なるアプリケーションでコンポーネントを記録および実行したい。

注 ÅF

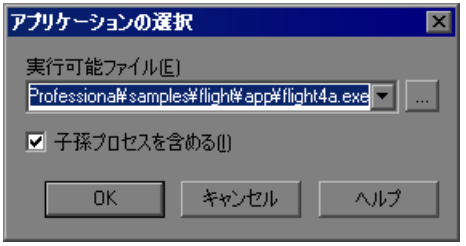
- 新しいコンポーネントを記録するときに［アプリケーション領域の設定］ダイアログ・ボックスの［アプリケーション］表示枠でアプリケーションの設定がまだ行われていなかった場合は，記録の開始時に［アプリケーション］ダイアログ・ボックスが開きます。［アプリケーション］ダイアログ・ボックスには，［アプリケーション］表示枠と同じオプションが含まれます。
- ［アプリケーション］ダイアログ・ボックスと［アプリケーション］表示枠には，コンピュータにインストールされている任意の QuickTest アドインに適用されるオプションが含まれている場合もあります。これらのオプションの詳細については，アドインに付属のドキュメントを参照してください。

［アプリケーション］表示枠には，次のオプションがあります。

オプション	説明
［Windows アプリケーション］	現在のアプリケーション領域に関連付けられたコンポーネントの記録と実行を行うアプリケーションの詳細が一覧表示されます。表示される詳細については，455 ページ「アプリケーションの選択」を参照してください。 Windows アプリケーションを対象に記録または実行を行わない場合は，アプリケーション・リストを空のままにしておきます。（これが標準設定です。）
	アプリケーション・リストにアプリケーションを追加します。10 個までのアプリケーションを追加できます。詳細については，455 ページ「アプリケーションの選択」を参照してください。
	選択したアプリケーションをアプリケーション・リストから削除します。
QuickTest で開いた任意のアプリケーションを対象に記録と実行を行う	QuickTest によって起動される任意のアプリケーションを（QuickTest の子プロセスとして）記録と実行の対象にします。たとえば，セッションの記録または実行中に OpenApp 関数を使用して開かれるアプリケーションや，アプリケーションを開く関数が含まれている別の操作によって開かれたアプリケーションなどが対象になります。
［その他］	（現在ロードされているアドインに基づいて）アプリケーション領域の関連コンポーネントが，現在記録できる環境が表示されます。

アプリケーションの選択

［アプリケーション］表示枠で［**追加**］ボタンをクリックすると，［アプリケーションの選択］ダイアログ・ボックスが開きます。



［アプリケーション］表示枠に表示されるアプリケーション・リストには最大 10 個のアプリケーションを追加できます。また，リスト内の既存のアプリケーションを編集できます。アプリケーションの子孫プロセスの記録と実行を行うかどうかを選択することもできます。

［アプリケーションの選択］ダイアログ・ボックスに入力した詳細は，［アプリケーション］表示枠の［**Windows アプリケーション**］領域にアプリケーションごとに 1 行で表示されます。

［アプリケーションの選択］ダイアログ・ボックスでは，アプリケーションに関する次の詳細を指定できます。

オプション	説明
［実行可能ファイル］	指定した実行可能ファイルを対象に記録と実行を行うように QuickTest が設定されます。
［子孫プロセスを含む］	このチェック・ボックスを選択すると，記録および実行セッション中に指定したアプリケーションによって作成されたプロセスを対象に記録と実行を行うように QuickTest が設定されます。たとえば，起動プログラムとしてのみ使用されるプロセスが，アプリケーションの機能を実際に提供する別のプロセスを作成する場合があります。したがって，このアプリケーションに対するテストを記録または実行するときにこの子孫プロセスが含まれていないと，アプリケーションの機能が記録されなかったり，実行セッションが失敗したりします。 標準設定では，このオプションは選択されています。

アプリケーション領域の回復シナリオ設定の定義

回復シナリオの設定では，ビジネス・コンポーネントが実行セッション中の予期しないイベントやエラーからどのように回復するかを指定できます。

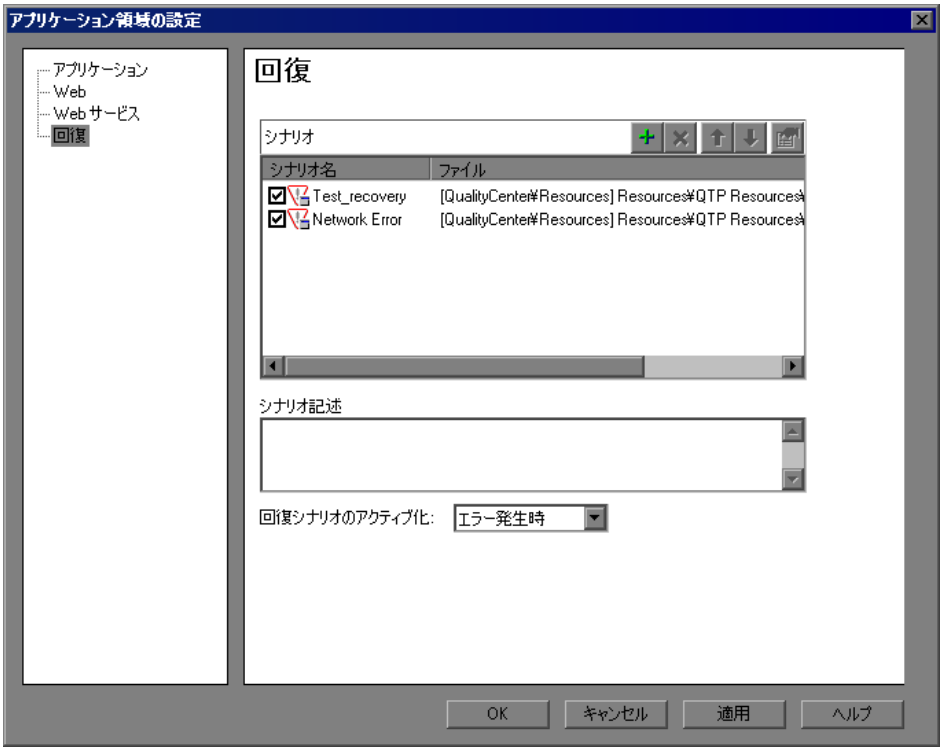
［回復］表示枠には，現在のアプリケーション領域に関連付けられているすべての回復シナリオのリストが表示されます。また，アプリケーション領域への追加の回復シナリオの関連付け，アプリケーション領域からのシナリオの削除，実行セッションに対する回復シナリオの適用順序の変更，および各シナリオのサマリの読み取り専用形式での表示が可能です。

アプリケーション領域に対する，特定のシナリオまたは回復メカニズム全体を有効または無効にできます。回復シナリオが Quality Center プロジェクトに格納

されている限り，既存の回復シナリオの追加，新しい回復シナリオの作成ができます。

注：QuickTest には，Web 関連テスト用のサンプル回復シナリオが用意されています。このファイルは，Quality Center プロジェクトの **Subject¥BPT Resources¥Recovery Scenarios¥DefaultWeb.qrs** です。

アプリケーション領域の回復シナリオは，テストの回復シナリオとまったく同じ方法で定義します。回復シナリオの詳細については，第 37 章「回復シナリオの定義と使用」を参照してください。



［回復］表示枠には，次のオプションがあります。

オプション	説明
［シナリオ］	アプリケーション領域に関連付けられている各回復シナリオの名前と回復ファイルのパスが表示されます。リスト内のシナリオの追加，削除，優先順位設定が可能なほか，選択したファイルのパスを編集できます。詳細については，458 ページ「関連回復シナリオの指定」を参照してください。
［シナリオ記述］	［**シナリオ**］ボックスで選択したシナリオの説明テキストが表示されます。
［回復シナリオのアクティブ化］	関連付けられているシナリオを次の条件に従って実行するように QuickTest を設定します。 ➤［**各ステップごと**］。ステップを実行するたびに回復メカニズムが呼び出されます（［**各ステップごと**］を選択すると，実行セッション中のパフォーマンスが低下する場合があります）。 ➤［**エラー発生時**］。エラーの戻り値を返すステップの後のみ，回復メカニズムが呼び出されます。 ➤［**なし**］。回復メカニズムが無効になります。

関連回復シナリオの指定

各シナリオの横のチェック・ボックスを選択またはクリアすることで，現在のアプリケーション領域で当該シナリオを有効または無効にできます。

また，回復シナリオ・ファイル・パスを編集するには，パスを一度クリックしてパスを強調表示し，その状態で再度クリックして編集モードに入ります。回復シナリオ・ファイルのパスを変更した場合は，現在のアプリケーション領域に関連付けられたコンポーネントを実行する前に，回復シナリオが新しいパスの場所に存在することを確認してください。

シナリオのタイプは，次のアイコンによって示されます。

アイコン	説明
	対象回復シナリオが，実行セッション中に開いているアプリケーションの特定のポップアップ・ウィンドウをきっかけとして起動されることを示します。
	対象回復シナリオが，アプリケーション内のオブジェクトのプロパティ値が特定の値に一致したときに起動されることを示します。
	対象回復シナリオが，コンポーネントのステップが正常に実行されなかったときに起動されることを示します。
	対象回復シナリオが，実行セッション中に指定のアプリケーションが失敗したときに起動されることを示します。
	対象回復シナリオが，アプリケーション領域で利用できなくなったことを示します。理由としては，回復ファイルの名前の変更や場所の移動が原因で QuickTest からアクセスできなくなったことなどが考えられます。実行セッション中に，関連付けられている回復ファイルが利用できないと，メッセージが結果に表示されます。

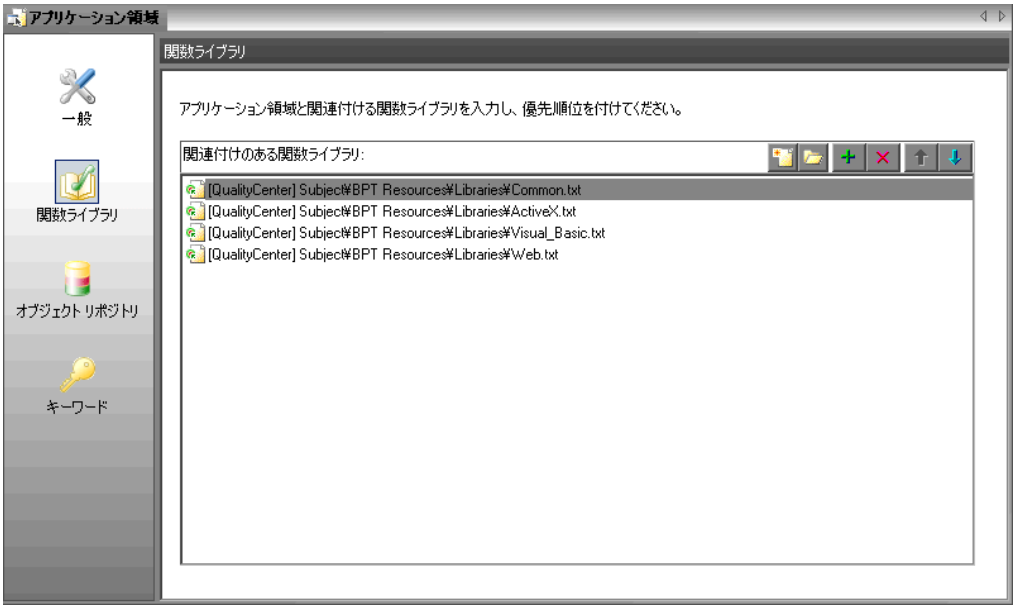
次のボタンを使用して，コンポーネントに関連付けられている回復シナリオ・ファイルの追加，削除，優先順位設定ができます。

ボタン	説明
	［回復シナリオの追加］ダイアログ・ボックスが開き，コンポーネントに 1 つ以上の回復シナリオを関連付けることができます。詳細については，第 37 章「回復シナリオの定義と使用」を参照してください。
	選択した回復シナリオをコンポーネントから削除します。
	選択したシナリオをリストの中で上に移動し，コンポーネントの実行セッション中の優先順位を上げます。
	選択したシナリオをリストの中で下に移動し，コンポーネントの実行セッション中の優先順位を下げます。
	選択した回復シナリオのプロパティのサマリを読み取り専用形式で表示します。詳細については，第 37 章「回復シナリオの定義と使用」を参照してください。

関数ライブラリの関連付け

［関数ライブラリ］表示枠では，QuickTest 関数ライブラリ，VBScript 関数ライブラリ，またはテキストファイルなどの関数ライブラリ・ファイルをアプリケーション領域に関連付けることができます。関数ライブラリをアプリケーション領域に関連付けることにより，ビジネス・コンポーネントの作成時に使用できるユーザ定義キーワードの形で追加機能を提供できます。

関連付けた関数ライブラリは，すべて Quality Center プロジェクトに保存する必要があります。



［関数ライブラリ］表示枠では，アプリケーション領域に現在関連付けられている関数ライブラリのリストが表示され，追加の関数ライブラリの関連付けや，関数ライブラリ・ファイルの変更，削除，および優先順位設定を行うことができます。関数ライブラリが Quality Center プロジェクトに格納されている限り，既存の関数ライブラリを追加したり，新しい関数ライブラリを作成することができます。

注：QuickTest には，あらかじめ定義された関数を含むサンプル関数ライブラリが用意されています。標準設定では，これらのファイルはすべての新しいアプリケーション領域に関連付けられます。標準の関数ライブラリは，Quality Center プロジェクトの **Subject¥BPT Resources¥Libraries** の下にあります。関数ライブラリにユーザ定義関数を作成する方法の詳細については，第 12 章「ユーザ定義関数および関数ライブラリを使用した作業」を参照してください。

コンポーネントに関連付けられている関数ライブラリの追加，変更，削除，および優先順位設定を行うには，次のボタンを使用します。

ボタン	説明
	新しい関数ライブラリを作成し，それを Quality Center プロジェクトに保存し，関数ライブラリのリストに追加できます。
	選択した関数ライブラリを表示または編集するために関数ライブラリ・ウィンドウで開きます。ほかの QuickTest または Quality Center ユーザが現在使用している関数ライブラリはロックされ，読み取り専用形式でのみ開くことができます。詳細については，第 12 章「ユーザ定義関数および関数ライブラリを使用した作業」を参照してください。
	Quality Center プロジェクトのテスト計画ツリーを参照し，アプリケーション領域に関連付ける既存の関数ライブラリを選択することができます。詳細については，462 ページ「既存のライブラリのアプリケーション領域への関連付け」を参照してください。
	選択した関数ライブラリをアプリケーション領域から削除します。
	選択した関数ライブラリをリスト内で上に移動することで，コンポーネント実行セッション中の優先順位を上げます。
	選択した関数ライブラリをリスト内で下に移動することで，コンポーネント実行セッション中の優先順位を下げます。

注：関連付けられた関数ライブラリを右クリックして，［**開く**］を選択すると関数ライブラリが開き，［**削除**］を選択するとアプリケーション領域との関連付けが削除されます。

関連付けられた関数ライブラリが見つからない場合（たとえば，Quality Center プロジェクトから削除された場合）は，リスト内の関数ライブラリの左側に［**見つからない関数ライブラリ**］アイコン が表示されます。見つからない関数ライブラリに対処するには，その関数ライブラリを右クリックし，［**探す**］を選択して必要な関数ライブラリを探すか，［**削除**］をクリックして見つからない関数ライブラリへの関連付けを削除します。

既存のライブラリのアプリケーション領域への関連付け

既存の関数ライブラリをアプリケーション領域に追加できます。これにより，アプリケーション領域に関連付けられたすべてのビジネス・コンポーネントから，関数ライブラリでキーワードとして定義された関数にアクセスできるようになります。

既存の関数ライブラリをアプリケーション領域に関連付けるには，次の手順を実行します。

1 QuickTest で，アプリケーション領域を開きます（まだ開いていない場合）。

➤［**ファイル**］＞［**開く**］＞［**アプリケーション領域**］を選択します。

➤［**開く**］ボタンの下向き矢印をクリックし，［**アプリケーション領域**］を選択します。

ヒント：ほかのアプリケーション領域がすでに開いている場合は，［**開く**］ボタンをクリックして必要なアプリケーション領域を選択します。

2 サイドバーの［**関数ライブラリ**］をクリックします。［関数ライブラリ］表示枠に，アプリケーション領域に現在関連付けられている関数ライブラリのリストが表示されます。

3［**関数ライブラリの追加**］ボタンをクリックします。リストに空白行と参照ボタンが追加されます。

4 参照ボタンをクリックします。［関数ライブラリを開く］ダイアログ・ボックスが開きます。

5 サイドバーでファイルの場所を選択します。たとえば［**ファイル・システム**］または ［Quality Center **テスト・リソース**］など。関数ライブラリを選択して［**開く**］をクリックします。

［関数ライブラリを開く］ダイアログ・ボックスが閉じ，アプリケーション領域の［関数ライブラリ］表示枠に選択したファイルが表示されます。関数ライブラリに構文エラーがある場合は，構文エラーのためにテストが失敗することを示すメッセージが表示されます。

新しい関数ライブラリの作成

アプリケーション領域の［関数ライブラリ］表示枠で直接新しい関数ライブラリを作成し，アプリケーション領域に自動的に関連付けることができます。

Quality Center プロジェクトに新しい関数ライブラリを作成するには，次の手順を実行します。

1 QuickTest で，アプリケーション領域を開きます（まだ開いていない場合）。

➤［**ファイル**］＞［**開く**］＞［**アプリケーション領域**］を選択します。

➤［**開く**］ボタンの下向き矢印をクリックし，［**アプリケーション領域**］を選択します。

ヒント：ほかのアプリケーション領域がすでに開いている場合は，［**開く**］ボタンをクリックして必要なアプリケーション領域を選択します。

---

2 サイドバーの［**関数ライブラリ**］をクリックします。［関数ライブラリ］表示枠に，アプリケーション領域に現在関連付けられている関数ライブラリのリストが表示されます。

3 ［関数ライブラリ］表示枠で，［**関数ライブラリの作成**］ボタンをクリックします。［関数ライブラリの保存］ダイアログ・ボックスが開きます。

4 サイドバーでファイルの場所，たとえば ［Quality Center **テスト・リソース**］などを選択します。名前を［**ファイル名**］ボックスに入力し，［**保存**］をクリックします。

新しい空の関数ライブラリが選択した場所に追加され，［関数ライブラリ］表示枠のリストに表示されます。

注：QuickTest の関数ライブラリは，サフィックスを指定しなかった場合，標準では **.qfl** ファイルとして作成されます。必要に応じて **.txt** ファイルや **.vbs** ファイルを作成することもできます。

ヒント：

- 新しい関数ライブラリに内容を追加したり，QuickTest で直接ファイルを変更したりするには，［関数ライブラリ］表示枠でファイルを選択し，［**関数ライブラリを開く**］ボタンをクリックするか，リスト内の関数ライブラリをダブルクリックします。関数ライブラリ・ウィンドウでファイルが開き，必要に応じてファイルを編集できます。変更を保存するには，ファイルを閉じて，メッセージが表示されたら［**はい**］をクリックします。
- 関数ライブラリの名前を変更するには，関数ライブラリを 2 回クリックするか，関数ライブラリを選択して F2 キーを押します。

関数ライブラリの編集の詳細については，第 12 章「ユーザ定義関数および関数ライブラリを使用した作業」を参照してください。

共有オブジェクト・ライブラリの関連付け

共有オブジェクト・リポジトリには，ビジネス・コンポーネントのステップを作成するときに使用できるテスト・オブジェクトがすべて保存されます。共有オブジェクト・リポジトリをアプリケーション領域に関連付けると，そのアプリケーション領域に関連付けられたすべてのコンポーネントが共有オブジェクト・リポジトリにアクセスできるようになります。

［オブジェクト リポジトリ］表示枠では，アプリケーション領域に現在関連付けられているオブジェクト・リポジトリのリストが表示され，追加のオブジェクト・リポジトリの関連付けや，オブジェクト・リポジトリ・ファイルの変更，削除，および優先順位設定を行うことができます。オブジェクト・リポジトリを Quality Center プロジェクトに格納する限り，既存のオブジェクト・リポジトリを追加したり，新しいオブジェクト・リポジトリを作成したりできます。

アプリケーション内のオブジェクトを学習するか，オブジェクト・リポジトリ・マネージャを使ってテスト・オブジェクトを手作業で追加することにより，テスト・オブジェクトをこの共有オブジェクト・リポジトリに追加できます。共有オブジェクト・リポジトリ内のテスト・オブジェクトの管理については，第 7 章「オブジェクト・リポジトリの管理」を参照してください。

注： QuickTest には標準の共有オブジェクト・リポジトリが用意されていますが（**Subject/BPT Resources/Object Repositories** フォルダにあります），それを使用しないよう強くお勧めします。この標準の共有オブジェクト・リポジトリをアプリケーション領域または個別のコンポーネントに関連付けた場合，この共有オブジェクト・リポジトリを使用しているコンポーネントはどれも正常に動作しなくなる可能性があります。

すでにテスト・オブジェクトが含まれている既存の共有オブジェクト・リポジトリ・ファイルを使用することも，新規の共有オブジェクト・リポジトリ・ファイルを作成することもできます。これらの共有オブジェクト・リポジトリを参照するアプリケーション領域に関連付けられたすべてのビジネス・コンポーネントは，これらの共有オブジェクト・リポジトリ・ファイルにアクセスします。詳細については，468 ページ「共有オブジェクト・リポジトリの新規作成」を参照してください。

共有オブジェクト・リポジトリにテスト・オブジェクトを追加した後で，オートメーション・エンジニアと各分野のエキスパートはそれらのテスト・オブジェクトを使ってビジネス・コンポーネントにステップを追加できます。詳細については，525 ページ「ステップの項目の選択」を参照してください。

各分野のエキスパートは，ビジネス・コンポーネントのステップを定義するときに，さまざまなテスト・オブジェクトを区別する必要があります。したがって，すべてのテスト・オブジェクト名にわかりやすい名前を付けることが重要です。QuickTest がテスト・オブジェクトを保存するときに自動的に割り当てた名前は，変更できます。たとえば，テスト・オブジェクト名が標準設定の「Edit」であり，このエディット・ボックスでユーザがユーザ名を入力する必要がある場合は，オブジェクト名を「UserName」に変更できます。

コンテナ・オブジェクトに対しては，コンテキストを指定することをお勧めします。たとえば，いくつかの確認メッセージ・ボックスがある場合，1 つを Login ＞ Confirm，もう 1 つを ChangePassword ＞ Confirm，さらにもう 1 つを BillingInfo ＞ Confirm と命名できます。

オブジェクト名を変更すると，QuickTest のキーワード・ビューと Quality Center のビジネス・コンポーネント・モジュールの［ステップ］タブに出現するそのオブジェクトの名前がすべて自動的に更新されます（古いテスト・オブジェクト名を使って作成されたステップでも同様です）。同じ共有オブジェクト・リポジトリを使用していて，変更されたオブジェクトが 1 回以上出現する別のコンポーネントを開くと，そのコンポーネント内の名前が更新されます。これにはしばらく時間がかかる場合があります。

テスト・オブジェクト名の変更の詳細については，182 ページ「テスト・オブジェクトの名前の変更」を参照してください。

アプリケーション領域（およびアプリケーション領域に関連付けられたコンポーネント）に関連付けられているオブジェクト・リポジトリの追加，変更，削除，および優先順位設定を行うには，次のボタンを使用します。

ボタン	説明
	新しいオブジェクト・リポジトリを作成し，それを Quality Center プロジェクトに保存し，オブジェクト・リポジトリのリストに追加できます。
	選択したオブジェクト・リポジトリを表示または編集するためにオブジェクト・リポジトリ・マネージャで開きます。現在ロックされているオブジェクト・リポジトリは，読み取り専用モードで開きます。オブジェクト・リポジトリ・マネージャの詳細については，第 7 章「オブジェクト・リポジトリの管理」を参照してください。
	Quality Center プロジェクトのテスト計画ツリーを参照して既存のオブジェクト・リポジトリを選択し，アプリケーション領域に関連付けることができます。詳細については，470 ページ「既存のオブジェクト・リポジトリのアプリケーション領域への関連付け」を参照してください。
	選択したオブジェクト・リポジトリをアプリケーション領域から削除します。
	選択したオブジェクト・リポジトリをリスト内で上に移動することで，コンポーネント実行セッション中の優先順位を上げます。
	選択したオブジェクト・リポジトリをリスト内で下に移動することで，コンポーネント実行セッション中の優先順位を下げます。

注：共有オブジェクト・リポジトリを右クリックして，［**開く**］を選択してオブジェクト・リポジトリ・マネージャで開くか，または［**削除**］を選択してアプリケーション領域との関連付けが削除できます。

共有オブジェクト・リポジトリが見つからない場合は，アプリケーション領域を開いたときに［欠落リソース］表示枠にその名前とパスが表示されます。見つからない共有オブジェクト・リポジトリに対処するには，関連付けられたオブジェクト・リポジトリのリストでその共有オブジェクト・リポジトリを右クリックし，［**探す**］を選択して必要な共有オブジェクト・リポジトリを探すか，［**削除**］をクリックしてその共有オブジェクト・リポジトリへの関連付けを削除します。

共有オブジェクト・リポジトリの新規作成

各分野のエキスパートがコンポーネント・ステップの実装時にアプリケーションからテスト・オブジェクトにアクセスできるようにするには，そのテスト・オブジェクトを Quality Center プロジェクト内の共有オブジェクト・リポジトリに格納する必要があります。アプリケーション領域の［オブジェクト リポジトリ］表示枠で直接新しい共有オブジェクト・リポジトリを作成し，アプリケーション領域に自動的に関連付けることができます。

Quality Center プロジェクトに新しい共有オブジェクト・リポジトリを作成するには，次の手順を実行します。

1 QuickTest で，アプリケーション領域を開きます（まだ開いていない場合）。

➤［**ファイル**］＞［**開く**］＞［**アプリケーション領域**］を選択します。

➤［**開く**］ボタンの下向き矢印をクリックし，［**アプリケーション領域**］を選択します。

ヒント：ほかのアプリケーション領域がすでに開いている場合は，［**開く**］ボタンをクリックして必要なアプリケーション領域を選択します。

2 サイドバーの［**オブジェクト リポジトリ**］をクリックします。［オブジェクト リポジトリ］表示枠に，アプリケーション領域に現在関連付けられているオブジェクト・リポジトリのリストが表示されます。

3 ［**オブジェクト リポジトリ**］表示枠で，［**オブジェクト リポジトリの作成**］ボタンをクリックします。［共有オブジェクト リポジトリの保存］ダイアログ・ボックスが表示されます。

4 サイドバーでファイルの場所，たとえば ［Quality Center テスト リソース］などを選択します。名前を［**ファイル名**］ボックスに入力し，［**保存**］をクリックします。

新しいオブジェクト・リポジトリが選択した場所に追加され，［オブジェクト リポジトリ］表示枠のリストに表示されます。

ヒント：

- 共有オブジェクト・リポジトリにテスト・オブジェクトを追加したり，QuickTest で直接ファイルを変更したりするには，［オブジェクト リポジトリ］表示枠でファイルを選択し，［**オブジェクト リポジトリを開く**］をクリックするか，リスト内のオブジェクト・リポジトリをダブルクリックします。オブジェクト・リポジトリ・マネージャでファイルが開き，必要に応じてファイルを編集できます。
- オブジェクト・リポジトリの名前を変更するには，オブジェクト・リポジトリを 2 回クリックするか，オブジェクト・リポジトリを選択して F2 キーを押します。

オブジェクト・リポジトリ・マネージャでオブジェクト・リポジトリを変更する方法の詳細については，第 7 章「オブジェクト・リポジトリの管理」を参照してください。

既存のオブジェクト・リポジトリのアプリケーション領域への関連付け

既存の共有オブジェクト・リポジトリ・ファイルをアプリケーション領域に関連付けることができます。これにより，そのアプリケーション領域が関連付けられたすべてのビジネス・コンポーネントから，これらのファイルに格納されたテスト・オブジェクトにアクセスできるようになります。

既存の共有オブジェクト・リポジトリをアプリケーション領域に関連付けるには，次の手順を実行します。

1 QuickTest で，アプリケーション領域を開きます（まだ開いていない場合）。

➤［**ファイル**］＞［**開く**］＞［**アプリケーション領域**］を選択します。

➤［**開く**］ボタンの下向き矢印をクリックし，［**アプリケーション領域**］を選択します。

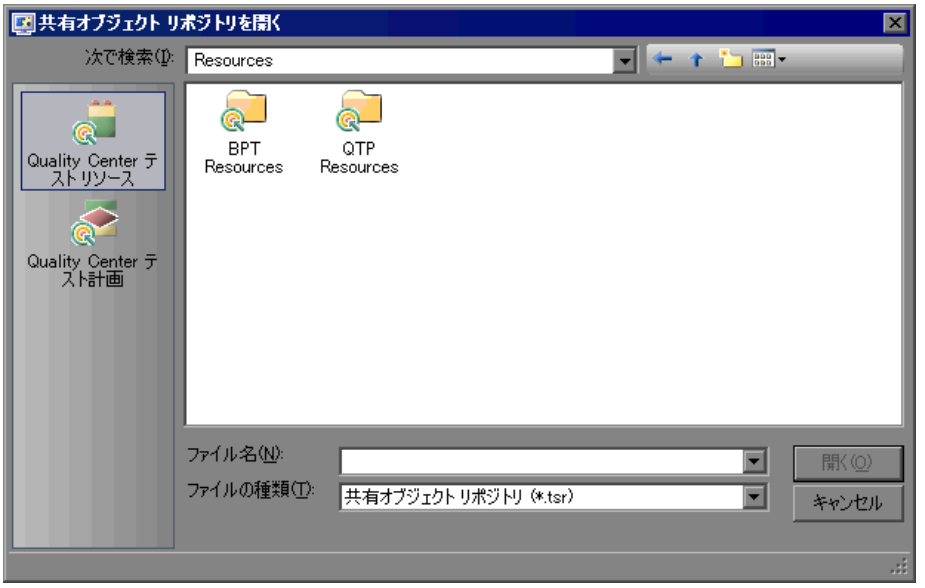
ヒント：ほかのアプリケーション領域がすでに開いている場合は，［**開く**］ボタンをクリックして必要なアプリケーション領域を選択します。

---

2 サイドバーの［**オブジェクト リポジトリ**］をクリックします。［オブジェクト リポジトリ］表示枠に，アプリケーション領域に現在関連付けられているオブジェクト・リポジトリのリストが表示されます。

3［**オブジェクト リポジトリを追加**］ボタンをクリックします。リストに空白行と参照ボタンが追加されます。

4 参照ボタンをクリックします。［共有オブジェクト リポジトリを開く］ダイアログ・ボックスが開きます。



5 必要な共有オブジェクト・リポジトリを選択し，［**開く**］をクリックします。［オブジェクト リポジトリを開く］ダイアログ・ボックスが閉じ，アプリケーション領域の［オブジェクト リポジトリ］表示枠に選択したファイルが表示されます。

6 共有オブジェクト・リポジトリにテスト・オブジェクトを追加したり，QuickTest で直接ファイルを変更したりするには，［オブジェクト リポジトリ］表示枠でファイルを選択し，［**オブジェクト リポジトリを開く**］をクリックするか，リスト内のオブジェクト・リポジトリをダブルクリックします。オブジェクト・リポジトリ・マネージャでファイルが開き，必要に応じてファイルを編集できます。

7 オブジェクト・リポジトリの名前を変更するには，オブジェクト・リポジトリを 2 回クリックするか，オブジェクト・リポジトリを選択して F2 キーを押します。

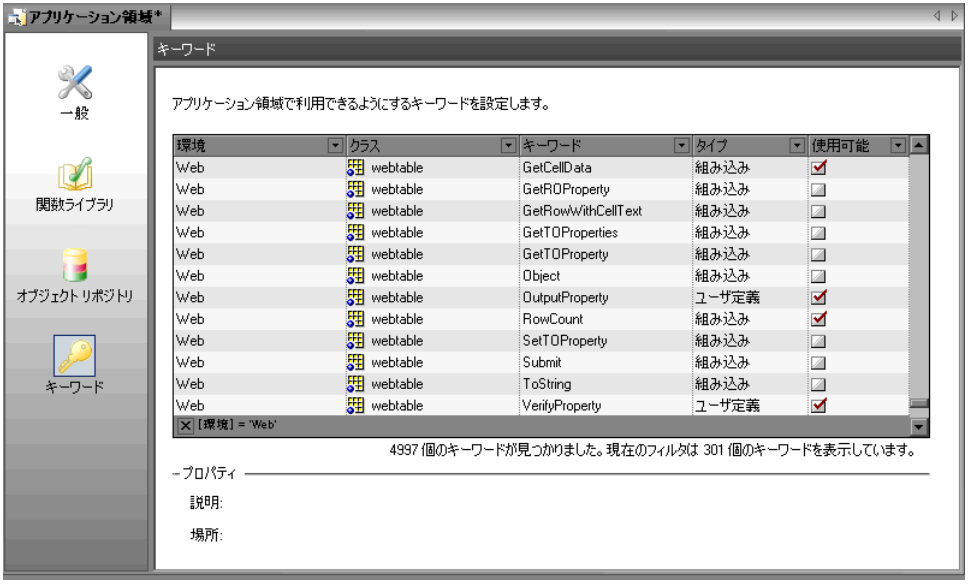
使用可能なキーワードの指定

各分野のエキスパートは，ビジネス・コンポーネントのステップを作成するときに，テスト対象のアプリケーションで実行する必要がある操作を選択します。これらの操作はキーワードとも呼ばれ，組み込みのメソッドやプロパティから派生されるほか，アプリケーション領域に関連付けられたユーザ定義関数からも派生されます。

［キーワード］表示枠には，組み込みのすべてのメソッドとプロパティ，およびユーザ定義関数ライブラリのすべての関数がキーワードとして表示されます。さらに，アドイン拡張を使用して自分または第三者が開発したテスト・オブジェクト・クラスすべてのメソッドおよびプロパティがキーワード表示枠に表示されます。

［キーワード］表示枠では，キーワードを管理したり，各分野のエキスパートがビジネス・コンポーネントの作成時にどのキーワードが使用可能かを選択することができます。標準設定では，選択された組み込みキーワードだけが使用できます。ただし，各分野のエキスパートはすべてのユーザ定義キーワードを使用できます。

注：［キーワード］表示枠はスクリプト・コンポーネントには適用されません。



各分野のエキスパートがビジネス・コンポーネントのステップの操作のリストの中から，キーワードを使用できるようにするには，［**使用可能**］カラムの該当するチェック・ボックスをクリックします。使用可能な操作のリストからキーワードを削除するには，該当するチェック・ボックスをクリアします。各分野のエキスパートは，チェック・ボックスがクリアされているキーワードを使用できなくなります。

［キーワード］表示枠では，キーワードに関する情報が次のカラムに表示されます。

カラム	説明
［環境］	キーワードが提供されるアドインの名前（Web や Visual Basic など）。表示枠には，現在ロードされているすべてのアドインで使用できるキーワードが表示されます。（アドイン拡張を使用して自分または第三者が開発したキーワードおよびアドインも含まれます。） **注**： ➤ テスト・オブジェクトに登録されているユーザ定義関数内のキーワードは，登録されている環境およびオブジェクト・クラスの下に表示されます。 ➤ テスト・オブジェクトに登録されていないユーザ定義関数内のキーワードと組み込みの VBScript 関数は，すべて［**グローバル**］環境の下に表示されます。
［クラス］	オブジェクト・クラス（Image や Winbutton など）。
［キーワード］	表示される操作名（Click や VerifyProperty など）。
［タイプ］	操作が**組み込み**（QuickTest によって提供される）と**ユーザ定義**（関数ライブラリに含まれている）のどちらであるか。
［使用可能］	各分野のエキスパートがこのキーワードをビジネス・コンポーネントのステップに使用できるかどうか。必要に応じて各チェック・ボックスを選択またはクリアできます。

リスト内のキーワードをクリックすると，そのキーワードに関する情報が表示枠の下部の［**プロパティ**］領域に表示されます。これには，キーワードの機能に関する説明テキスト，および（ユーザ定義キーワードの場合は）関数ライブラリの名前とパスが含まれます。組み込みキーワードの場所は「**内部**」として定義されます。

キーワード表示枠内のデータの表示，並べ替え，およびフィルタリングを実行することで，各分野のエキスパートに使用できるようにする（または非表示にする）キーワードを探すのが簡単になります。

ヒント：

- ［キーワード］表示枠のカラムの表示順序を変更するには，カラム・ヘッダを新しい位置にドラグします。カラムを使用可能な位置にドラッグすると，赤い矢印が表示されます。
- カラムの幅が狭いためにデータの一部が隠れる場合は，マウスを使ってカラムのサイズを変更できます。幅を調整するには，カラム・ヘッダの境界線をドラッグします。

カラムのフィルタリング

［キーワード］表示枠のデータにフィルタを適用して，使用したいキーワードだけを表示できます。1 つのカラムのデータにのみフィルタを適用することも，追加のカラムにフィルタを適用して表示される項目の数を減らすこともできます。

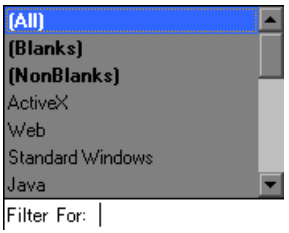
たとえば，各分野のエキスパートが現在使用できない Web アドインのキーワードのみを表示するとします。この場合，Web アドインのキーワードのみを表示するために［**環境**］カラムにフィルタを適用し，次にチェック・ボックスがクリアされているキーワードのみを表示するために［**使用可能**］カラムにフィルタを適用します（［**使用可能**］カラムのフィルタ・リストから［**(Unchecked)**］を選択します）。

カラムの下に，フィルタ条件と現在のフィルタに一致するキーワードの数が表示されます。

フィルタ条件の左側にある ✕ をクリックすると，フィルタがクリアされ，すべてのキーワードが表示されます。

カラムのデータにフィルタを適用するには，次の手順を実行します。

カラム・ヘッダの矢印 をクリックします。カラムに含まれる一意の項目のリストが表示されます。



次の操作を実行することにより，カラムのデータにフィルタを適用できます。

- リスト内の項目をクリックします。CTRL キーを使ってフィルタ・リストから複数の項目を選択できます。［キーワード］表示枠が更新され，選択した項目名を持つキーワードのデータだけが表示されます。

  次に，別のカラム・ヘッダの矢印をクリックし，そのリストの項目を選択できます。フィルタが適用されたデータに再度フィルタが適用され，選択したすべてのフィルタ条件に一致するキーワードだけが表示されます。

- フィルタ・リストの下部にある［**Filter For:**］ボックスには，**?**，*****，**#** などのワイルドカードを含むフィルタ・パターンを入力できます。ENTER キーを押すと，そのパターンに従ってデータにフィルタが適用されます。任意の 1 文字を表すには **?** を，任意の文字の 0 回以上の繰り返しを表すには ***** を，任意の数字を表すには **#** をそれぞれ使用します。また，パターン内のいずれかのオプションだけに一致する項目を指定するには，**|** を使用します。たとえば，Verify*|Check* は **Verify** または **Check** で始まるすべてのキーワードを示します。

- 複数のフィルタを同時に適用することもできます。たとえば，標準 Windows 環境と ActiveX 環境のキーワードのうち，（ユーザ定義キーワードではなく）組み込みキーワードだけを表示するには，**標準** Windows 用，ActiveX 用，および**組み込み**タイプ用の 3 つのフィルタを適用します。

カラム内容の並べ替え

カラム・ヘッダをクリックすると，カラム内のデータをアルファベットの昇順または降順で並べ替えることができます。［**使用可能**］カラムは，選択またはクリアしたチェック・ボックスに従って並べ替えられます。

並べ替えの方向は，カラム・ヘッダの矢印で示されます。データの並び順を逆にするには，カラム・ヘッダをもう一度クリックします。

第 V 部

コンポーネントを使った作業

第 15 章

ビジネス・コンポーネントの使用

ビジネス・コンポーネント・キーワード・ビューでは，QuickTest でのビジネス・コンポーネントの作成，表示，変更，デバッグを行うことができます。

本章の内容

- ビジネス・コンポーネントの使用について（482 ページ）
- 新規ビジネス・コンポーネントの作成（483 ページ）
- ビジネス・コンポーネントを開く（488 ページ）
- ビジネス・コンポーネントの保存（494 ページ）
- 手動コンポーネントを使った作業（500 ページ）
- コンポーネントに関連付けらているアプリケーション領域の変更（503 ページ）
- コンポーネントの印刷（505 ページ）

ビジネス・コンポーネントの使用について

ビジネス・コンポーネントは通常，各分野のエキスパートによって Quality Center で作成し，変更されます。詳細については，『**HP Business Process Testing ユーザーズ・ガイド**』を参照してください。ただし，ビジネス・コンポーネント・キーワード・ビューでは，必要に応じて QuickTest でビジネス・コンポーネントの作成，表示，変更，デバッグを行うことができます。

キーワード・ビューでは，モジュール化されたキーワード駆動型の表形式に，ビジネス・コンポーネントがステップごとに分けて表示されます。各ステップは，簡単に変更できる独立した部分から構成されています。ステップは，必要に応じて項目と操作を選択し，追加情報を入力することによって，作成したり変更することができます。

ビジネス・コンポーネントの各ステップには，完成したときに自動的にドキュメント化されます。これにより，ユーザはステップの説明をわかりやすい文章で読むことができます。また，ビジネス・コンポーネントに関連付けられているアプリケーション領域に関数ライブラリを追加した場合は，ユーザ定義操作（関数）を選択してステップを定義するときに，関数ライブラリに追加したドキュメントがステップに対して表示されます。詳細については，414 ページ「関数への説明の追加」を参照してください。

ビジネス・コンポーネントを作成するか開く前に，まず QuickTest をビジネス・コンポーネント，およびアプリケーション領域のリソースと設定が保存されている Quality Center プロジェクトに接続する必要があります。Quality Center プロジェクトに接続することによって，QuickTest でビジネス・コンポーネントを作成したり開いたりできるようになります。また，ビジネス・コンポーネントから，アプリケーション領域で定義されたそのコンポーネントが基づいているあらゆるリソースにアクセスできるようになります。

注：ビジネス・コンポーネントおよびアプリケーション領域を使用する前に，必要な Quality Center アクセス許可と権限を持っていることを確認する必要があります。ビジネス・コンポーネント・モジュールにおける権限の設定の詳細については，『**HP Business Process Testing ユーザーズ・ガイド**』を参照してください。

選択したアプリケーション領域に必要なリソースと設定がまだすべて含まれていない場合でも，ManualStep 関数または［**コメント**］オプションを使ってステップを追加できます。このオプションを使えば，Quality Center，あるいは Microsoft Excel や Microsoft Word などのほかのアプリケーションで入力するように，手動のステップを自由に入力できます。また，コメントを使って，ステップに関する情報を追加したり，ビジネス・コンポーネントのセクション間を区切ったりすることもできます。各手動ステップおよびコメントは，キーワード・ビュー内に独立した行として表示されます。詳細については，502 ページ「コンポーネントの手動ステップの追加および変更」および 551 ページ「コメントの使用」を参照してください。

注 ÅF

- コンポーネントを削除する場合は，QuickTest または Quality Center のどちらで作成されたものでも，Quality Center においてのみ削除できます。詳細については，『**HP Business Process Testing ユーザーズ・ガイド**』を参照してください。
- 必要であれば，ビジネス・コンポーネントをスクリプト・コンポーネントに変換できます。詳細については，第 16 章「スクリプト・コンポーネントの作成」を参照してください。

新規ビジネス・コンポーネントの作成

QuickTest を Quality Center プロジェクトに接続すると，そのプロジェクトに新規のビジネス・コンポーネントを作成できます。

各ビジネス・コンポーネントは特定のアプリケーション領域に基づいており，アプリケーション領域はそのコンポーネントの保存先となる Quality Center プロジェクトに保存されています。各アプリケーション領域は，ビジネス・コンポーネントの設定とリソースを指定します。これには，共有オブジェクト・リポジトリ，関数ライブラリ・ファイル，回復シナリオの各場所，およびその他の情報が含まれます。使用可能なアプリケーション領域が複数存在する場合もあります。作成するビジネス・コンポーネントに最も適したアプリケーション領域を選択します。詳細については，第 13 章「アプリケーション領域の管理」を参照してください。

通常，ビジネス・コンポーネントは各分野のエキスパートによって Quality Center で作成されます。詳細については，『**HP Business Process Testing ユーザーズ・ガイド**』を参照してください。ただし，必要に応じて QuickTest でビジネス・コンポーネントを作成することもできます。この項では，QuickTest で新規コンポーネントを作成する方法を説明します。

注：QuickTest でビジネス・コンポーネントを新規に作成するためには，ビジネス・プロセス・テストの作成に必要な権限が必要です。詳細については，『**HP Quality Center Administrator Guide**』（英語版）を参照してください。

新規のビジネス・コンポーネントを作成するには，次の手順を実行します。

1 ビジネス・コンポーネントの保存先となる Quality Center プロジェクトに接続します。詳細については，46 ページ「Quality Center プロジェクトへの接続」を参照してください。

2 ［**ファイル**］＞［**新規作成**］＞［**ビジネス コンポーネント**］を選択し，［ビジネス コンポーネントの新規作成］ダイアログ・ボックスを開きます。

3 ［**アプリケーション領域名**］ボックスから適切なアプリケーション領域を選択し，［**OK**］をクリックします。詳細については，485 ページ「［ビジネス コンポーネントの新規作成］ダイアログ・ボックス」を参照してください。

無題の新しいビジネス・コンポーネントがキーワード・ビュー内に開きます。新規のビジネス・コンポーネントにはまだ内容はありませんが，基盤とするアプリケーション領域に定義されている必要な設定とリソースはすべて含まれています。これらの設定は，［**ファイル**］＞［**設定**］を選択して，読み取り専用形式で表示できます。設定を変更する必要がある場合は，関連するアプリケーション領域から変更します。

注：まだアプリケーション領域を定義していない場合は，Business Process Testing に用意されている標準設定を使用して，無題の新しいビジネス・コンポーネントが開きます。アプリケーション領域の定義後，そのアプリケーション領域にビジネス・コンポーネントを関連付けることができます。詳細については，第 13 章「アプリケーション領域の管理」を参照してください。

---

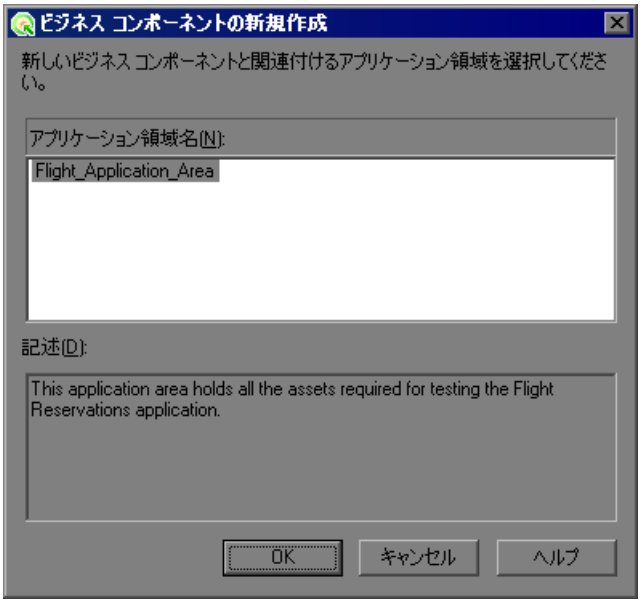
4 この時点で，次の操作を実行できます。

- ビジネス・コンポーネントへのステップとコメントの追加。詳細については，523 ページ「コンポーネントへのステップの追加」および 551 ページ「コメントの使用」を参照してください。
- コンポーネントを保存します。（ステップは後で追加できます。）詳細については，494 ページ「ビジネス・コンポーネントの保存」を参照してください。

［ビジネス コンポーネントの新規作成］ダイアログ・ボックス

説明	Quality Center プロジェクト内に新規のビジネス・コンポーネントを作成できます。
アクセス方法	Quality Center プロジェクトに接続します。次のいずれかを実行します。 ➤［**ファイル**］＞［**新規作成**］＞［**ビジネス コンポーネント**］メニュー・コマンドを選択します。 ➤［**新規作成**］ボタンの下矢印をクリックし，［**ビジネス コンポーネント**］を選択します。 ➤ CTRL+S キーを押します。 **ヒント**：アプリケーション領域がすでに開いている場合は，［**新規作成**］ボタンをクリックすると新しいアプリケーション領域が開きます。
重要な情報	［ビジネス コンポーネントの新規作成］ダイアログ・ボックスは使用可能なアプリケーション領域をすべて表示します。アプリケーション領域を選択すると，その説明が表示されます。
さらに調べる	**概念の概要**：482 ページ「ビジネス・コンポーネントの使用について」 **主要なタスク**：483 ページ「新規ビジネス・コンポーネントの作成」 **追加の関連トピック**：487 ページ「追加の参照情報」

下記が［ビジネス コンポーネントの新規作成］ダイアログ・ボックスの画像です。



［ビジネス コンポーネントの新規作成］ダイアログ・ボックス・オプション

オプション	説明
［アプリケーション領域名］	コンポーネントを関連付けることのできる使用可能なアプリケーション領域を一覧表示します。 アプリケーション領域を選択し，［**OK**］をクリックして新しいコンポーネントを作成します。
［記述］	選択したアプリケーション領域の説明を表示します。これは各分野のエキスパートが Quality Center で見るものと同じ説明です。 各分野のエキスパートは Quality Center に新規のビジネス・コンポーネントを作成するときに，これらの説明によってどのアプリケーション領域を選択するべきかを判断できます アプリケーション領域の記述はアプリケーション領域ウィンドウの［一般］タブで定義されます。詳細については，446 ページ「一般設定の定義」を参照してください。

追加の参照情報

関連するタスク	46 ページ「Quality Center プロジェクトへの接続」
関連する概念	➤ 488 ページ「ビジネス・コンポーネントを開く」 ➤ 494 ページ「ビジネス・コンポーネントの保存」 ➤ 497 ページ「[ビジネス コンポーネントの保存］ダイアログ・ボックス」 ➤ 503 ページ「コンポーネントに関連付けらているアプリケーション領域の変更」

ビジネス・コンポーネントを開く

QuickTest が Quality Center プロジェクトに接続されている場合，プロジェクトに保存されているコンポーネントを開き，表示，変更，デバッグ，実行を行うことが可能です。コンポーネントは，コンポーネント・ツリー内の場所に応じて見つけます。

現在 Quality Center，あるいはほかの QuickTest セッションで開いているコンポーネントはロックされ，読み取り専用形式でのみ開くことができます。これらのコンポーネントで作業をするには，このコンポーネントを開いている Quality Center または別の QuickTest セッションをすべて閉じなくてはなりません。

コンポーネントを開いたとき，関連付けられているアプリケーション領域が見つからない場合，別のアプリケーション領域の関連付けを求められます。

ビジネス・コンポーネントとスクリプト・コンポーネントは同じ方法で開くことができます。

ビジネス・コンポーネントまたはスクリプト・コンポーネントを開くには次のいずれかを実行します。

➤ ［**ファイル**］＞［**開く**］＞［**ビジネス / スクリプト コンポーネント**］を選択し，［ビジネス コンポーネントを開く］ダイアログ・ボックスでコンポーネントを選択します。詳細については，490 ページ「［ビジネス コンポーネントを開く］ダイアログ・ボックス」を参照してください。

➤ ［**ファイル**］メニューの最近使用したファイルのリストからコンポーネントを選択します。詳細については，494 ページ「最近使用したファイルのリストからコンポーネントを開く」を参照してください。

スクリプト・コンポーネントの詳細については，507 ページ「スクリプト・コンポーネントの作成」を参照してください。

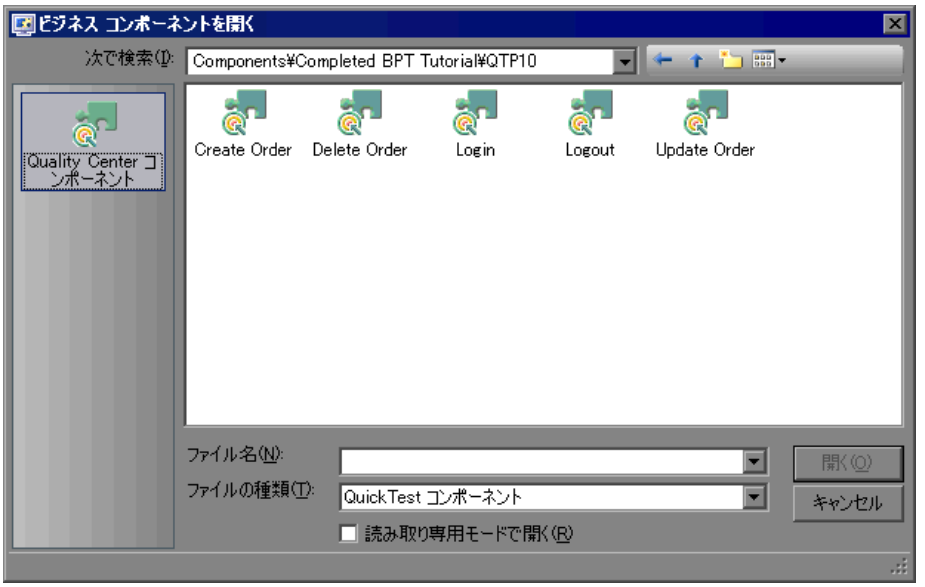
以前のバージョンの QuickTest をお使いのユーザの方へ

- QuickTest のバージョン 9.x を使用して作成したコンポーネントを変更するには，QuickTest Asset Upgrade Tool for Quality Center を使用して現在のバージョンのフォーマットにアップグレードする必要があります。ただし，アップグレード前でも読み取り専用形式での表示，および実行は可能です。（QuickTest のバージョン 8.x を使用して作成したコンポーネントを使用するには，最初に QuickTest のバージョン 9.x にアップグレードする必要があります。）
- 変換されたコンポーネントを保存した後は，そのコンポーネントは QuickTest の以前のバージョンでは使用できなくなります。
- 以前のバージョンの QuickTest を実行しているコンピュータで，それ以降のバージョンの QuickTest を使用して作成したコンポーネントを開くことはできません。たとえば，QuickTest 8.2 を実行しているコンピュータで，QuickTest MyVersion # で作成したコンポーネントを開くことはできません。

［ビジネス コンポーネントを開く］ダイアログ・ボックス

説明	既存のビジネス・コンポーネントを開くことができます。 ［ビジネス コンポーネントを開く］ダイアログ・ボックスは，Quality Center プロジェクトに保存されているコンポーネントを表示します。
アクセス方法	Quality Center プロジェクトに接続します。次のいずれかを実行します。 ➤［**ファイル**］＞［**開く**］＞［**ビジネス / スクリプト コンポーネント**］メニュー・コマンドを選択します。 ➤［**新規作成**］ボタンの下矢印をクリックし，［**ビジネス / スクリプト コンポーネント**］を選択します。 ➤ CTRL+S+O キーを押します。
重要な情報	コンポーネントが開くと，QuickTest のタイトル・バーに，**Components** という文字列と，フルパスおよびコンポーネント名が表示されます。たとえば，flight_login コンポーネントを開くと，タイトル・バーに次のように表示されます。 [Components¥Flight¥flight_login] 次の場合，コンポーネントは読み取り専用モードで開きます。 ➤［**読み取り専用モードで開く**］を選択した場合 ➤ 現在（バージョン・コントロール・サポートのプロジェクトの）バージョン・コントロール・データベースにチェック・インされているテストを開いた場合 ➤ 現在（バージョン・コントロール・サポートのプロジェクトの）ほかのユーザがチェック・アウトしているテストを開いた場合
さらに調べる	**概念の概要**：488 ページ「ビジネス・コンポーネントを開く」 **追加の関連トピック**：493 ページ「追加の参照情報」

次に［ビジネス コンポーネントを開く］ダイアログ・ボックスの画像を示します。



［ビジネス コンポーネントを開く］ダイアログ・ボックス・オプション

オプション	説明
［ファイルの場所］	Quality Center ビジネス・コンポーネント・モジュールの Quality Center パスをリストします。下向きの矢印を使用して必要なフォルダに移動するか，コンポーネント・リスト・領域のフォルダをダブルクリックして必要なフォルダに移動することができます。
サイドバー	コンポーネント，または関数ライブラリなどのほかのアセットが保存されている場所を表示します。ビジネス・コンポーネントの場合，**Quality Center コンポーネント**のみが表示されます。

オプション	説明
コンポーネント・リスト領域	現在のパスに保存されているフォルダ，またはコンポーネント，あるいはその両方を表示します。 **注**：コンポーネントがバージョン管理をサポートする Quality Center プロジェクトに保存されている場合，［**表示**］の下向き矢印をクリックし，［**詳細**］を選択することで，バージョン管理情報を表示できます。 ➤［**名前**］カラムに表示されているのは，選択したサブジェクトに属するコンポーネントの名前です。 ➤［**更新者**］カラムはコンポーネントを作成，または最後に変更した Quality Center ユーザを示します。 ➤［**チェック アウト先**］カラムはコンポーネントの現在のチェック・アウト先の Quality Center ユーザを示します。テストがチェック・インされている場合は，ここは空白です。
［**ファイル名**］	コンポーネント・リスト領域で選択されているコンポーネントの名前を表示します。
［**ファイルの種類**］	開きたいファイルの種類を選択することで，表示されるアセットのリストをフィルタすることができます。次のコンポーネントの種類のいずれかを選択することができます。 ➤［**QuickTest コンポーネント**］：QuickTest Professional または Quality Center を使用して自動化されたコンポーネントを表示します。 ➤［**手動コンポーネント**］：Quality Center で作成され，まだ自動コンポーネントに変換されていないコンポーネントを表示します。詳細については，500 ページ「手動コンポーネントを使った作業」を参照してください。 ➤［**すべてのコンポーネント**］：QuickTest の自動コンポーネントと手動コンポーネントをすべて表示します。

オプション	説明
［**読み取り専用モードで開く**］チェックボックス	コンポーネントを読み取り専用モードで開く場合に選択します。このオプションによりコンポーネントを表示できますが，変更はできません。
［**開く**］ボタン	クリックして選択したコンポーネントを開きます。 コンポーネントがバージョン管理の有効な Quality Center プロジェクトにチェック・インされている場合，［**開く**］ボタンに［**開いてチェック・アウトする**］オプションの下向き矢印が現れ，コンポーネントを開き，かつ即座にチェック・アウトすることができます。詳細については，971 ページ「バージョン・コントロール・データベースからのアセットのチェックアウト」を参照してください。 ［**開く**］をクリックした後，コンポーネントが開き，タイトル・バーがコンポーネント名を表示します。

追加の参照情報

関連するタスク	46 ページ「Quality Center プロジェクトへの接続」
関連する概念	➤ 483 ページ「新規ビジネス・コンポーネントの作成」 ➤ 494 ページ「ビジネス・コンポーネントの保存」 ➤ 497 ページ「［ビジネス コンポーネントの保存］ダイアログ・ボックス」 ➤ 503 ページ「コンポーネントに関連付けらているアプリケーション領域の変更」
その他の関連情報	［**ファイル**］メニューの最近使用したファイルのリストから選択することで，最近使用したコンポーネントを開くこともできます。Quality Center プロジェクトに接続されていないときにコンポーネントを選択した場合，あるいはほかの Quality Center プロジェクトに保存されているコンポーネントを選択した場合には，そのプロジェクトに接続するかどうかを問うメッセージが表示されます。詳細については，494 ページ「最近使用したファイルのリストからコンポーネントを開く」を参照してください。

最近使用したファイルのリストからコンポーネントを開く

［ファイル］メニューの最近使用したファイルのリストからコンポーネントを開くことができます。Quality Center プロジェクトのコンポーネントを選択したときに，現在 QuickTest が Quality Center に接続されていないか，コンポーネントの正しいプロジェクトに接続されていないと，［Quality Center プロジェクトへの接続］ダイアログ・ボックスが表示され，正しいサーバ，プロジェクト，および前回当該コンピュータでコンポーネントを開いたユーザの名前が表示されます。

プロジェクトにログインし，［**OK**］をクリックします。

また，前回異なる Quality Center ユーザ名を使用して編集したコンポーネントを開くことを選択した場合も，［Quality Center プロジェクトへの接続］ダイアログ・ボックスが開きます。表示されているユーザ名を使ってログインすることも，［**キャンセル**］をクリックして現在のユーザ名でログインしたままでいることもできます。

ビジネス・コンポーネントの保存

作成あるいは変更したビジネス・コンポーネントは，Quality Center プロジェクトに保存できます。（保存されていないコンポーネントはタイトル・バーにアスタリスクが表示され，まだ保存されていないことを示します。）

ビジネス・コンポーネントを保存する際には，わかりやすい名前を付けて，Quality Center プロジェクト（ビジネス・コンポーネント・モジュール）のコンポーネント・ツリー内の適切なフォルダに保存してください。QuickTest がコンポーネントを保存している間，実行中の操作がステータス・バーに表示されます。

また，既存のコンポーネントのコピーを，同じ Quality Center プロジェクト内の任意のフォルダに保存することもできます。すべてのユーザが各種のコンポー

ネントを区別できるように，コンポーネントのコピーの名前は，たとえ別のフォルダに保存する場合でも変更するようにします。

注：各分野のエキスパートは，Quality Center ビジネス・コンポーネント・モジュールから保存されたコンポーネントにアクセスできます。詳細については，『**HP Business Process Testing ユーザーズ・ガイド**』を参照してください。

新しいコンポーネントを保存，または既存のコンポーネントを異なる名前で保存する場合，次のようにします。

1 ［**ファイル**］＞［**保存**］を選択し，［ビジネス コンポーネントの保存］ダイアログ・ボックスを開きます。詳細については，497 ページ「［ビジネス コンポーネントの保存］ダイアログ・ボックス」を参照してください。

2 コンポーネントを保存するフォルダを選択し，コンポーネントの名前を入力（命名規則に従って）して，［**OK**］をクリックします。コンポーネントが Quality Center プロジェクトに保存されます。これで，ビジネス・コンポーネントを QuickTest で表示，変更できるようになります。

変更したコンポーネントは次のように保存します。

［**ファイル**］＞［**保存**］を選択するか，［**保存**］をクリックします。

スクリプト・コンポーネントのガイドライン

- ビジネス・コンポーネントとスクリプト・コンポーネントは同じように保存します。スクリプト・コンポーネントの詳細については，507 ページ「スクリプト・コンポーネントの作成」を参照してください。
- データ・テーブルのデータ・シート名はスクリプト・コンポーネント名と同一です。スクリプト・コンポーネントを新しい名前で保存すると（[**ファイル**] > [**名前を付けて保存**]），データ・シート名は自動的に変更されます。名前によってデータ・シートを参照するステップがある場合，実行セッションでそのステップは失敗します。これはこのステップが以前のデータ・シート名を参照するためです。スクリプト・コンポーネントを新しい名前で保存する場合，エキスパート・ビューの中で以前のデータ・シート名への参照がないか調べ，あればそれを新しいデータ・シート名で置換する必要があります。

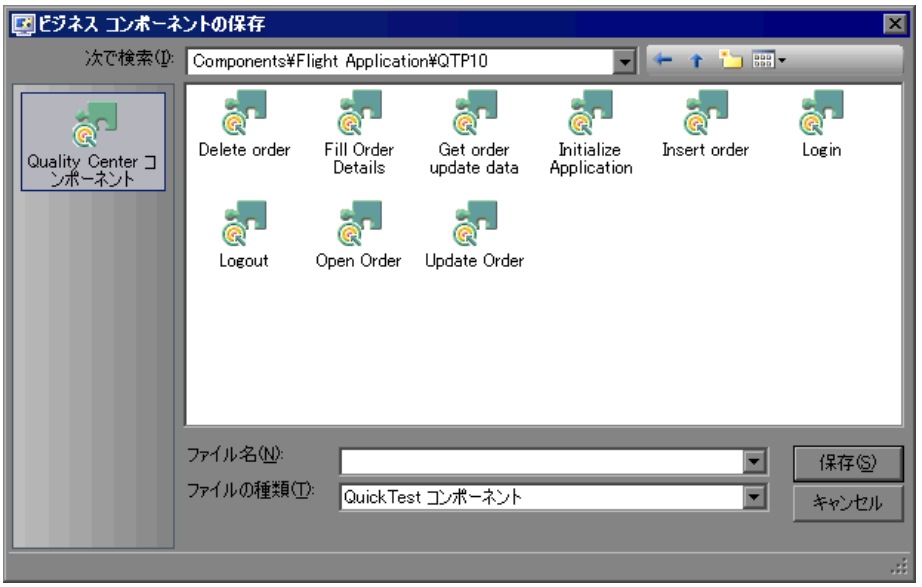
ヒント：ビジネス・コンポーネントをスクリプト・コンポーネントに変換することもできます。詳細については，514 ページ「スクリプト・コンポーネントへのビジネス・コンポーネントの変換」を参照してください。

スクリプト・コンポーネントの詳細については，507 ページ「スクリプト・コンポーネントの作成」を参照してください。

［ビジネス コンポーネントの保存］ダイアログ・ボックス

説明	新しいコンポーネントの保存，または既存コンポーネントを異なる名前で保存することができます。
アクセス方法	**新しいコンポーネントを保存するには次のいずれかを行います。** ➤［**ファイル**］＞［**保存**］メニュー・コマンドを選択します。 ➤［**保存**］ボタン をクリックします。 ➤ CTRL+S キーを押します。 **既存のコンポーネントを異なる名前で保存するには次のようにします。** ［**ファイル**］＞［**名前を付けて保存**］メニュー・コマンドを選択します。
さらに調べる	**概念の概要**：494 ページ「ビジネス・コンポーネントの保存」 **追加の関連トピック**：499 ページ「追加の参照情報」

次に［ビジネス コンポーネントの保存］ダイアログ・ボックスの画像を示します。



［ビジネス コンポーネントの保存］ダイアログ・ボックス・オプション

オプション	説明
［ファイルの場所］	Quality Center ビジネス・コンポーネント・モジュールの Quality Center パスをリストします。
［**ファイル名**］	コンポーネントに名前を付けられます。 コンポーネントの名前を入力し，［**OK**］をクリックします。誰もがコンポーネントを識別しやすいよう，わかりやすい名前を付けます。 コンポーネント名は以下の規則に従う必要があります。 ➤ 220 文字を超えてはいけません（パスも含まれます。例：Components/CompFolder1/MyComponent） ➤ 先頭，または終わりに空白を使用しないこと ➤ 次の文字を含めないこと。¥ / : " ? < > | * ! { }í % ➤ 連続する 2 つのセミコロンを含めないこと（;;）
［ファイルの種類］	保存可能なコンポーネントの種類がリストされます。例：［**ビジネス コンポーネント**］

追加の参照情報

関連する概念	➤ 483 ページ「新規ビジネス・コンポーネントの作成」 ➤ 488 ページ「ビジネス・コンポーネントを開く」 ➤ 494 ページ「ビジネス・コンポーネントの保存」 ➤ 497 ページ「［ビジネス コンポーネントの保存］ダイアログ・ボックス」 ➤ 503 ページ「コンポーネントに関連付けらているアプリケーション領域の変更」

手動コンポーネントを使った作業

QuickTest では，Quality Center で作成した手動コンポーネントを自動ビジネス・コンポーネントに変換できます。変換後は，ほかのビジネス・コンポーネントと同様に，表示，変更，デバッグ，および実行ができます。

手動コンポーネントを自動ビジネス・コンポーネントに変換した後も，その手動ステップは依然として Quality Center で表示できます。また，Quality Center マニュアル・ランナーを使用して手動コンポーネントとして実行することもできます。Quality Center の［自動化］タブで，自動コンポーネントのステップの変更や追加を行うこともできます。それらのステップは，［ステップのデザイン］タブと QuickTest で自動的に更新されます。

注：Quality Center の内部で手動コンポーネントを自動コンポーネントに変換することもできます。詳細については，『**HP Business Process Testing ユーザーズ・ガイド**』を参照してください。

手動コンポーネントを開いて変換する

QuickTest では，Quality Center プロジェクトに格納されている手動コンポーネントを開き，自動ビジネス（キーワード駆動の）コンポーネントに変換できます。この変換処理は元には戻せません。（ただし，必要な場合は Quality Center で手動ステップを表示および実行できます）。

QuickTest で手動コンポーネントを開くとき，当該コンポーネントをビジネス・コンポーネントに変換するかどうか尋ねられます。手動コンポーネントを自動コンポーネントに変換すると，手動コンポーネントの各手動ステップがキーワード・ビューの ManualStep 操作に変換されます。

項目	操作	値
操作	ManualStep	"Step1","ユーザ名を入力"
操作	ManualStep	"Step2","パスワードを入力"
操作	ManualStep	"Step3","Enter ボタンを押下"

各 ManualStep 操作について，各手動ステップの名前，説明，および期待結果が引数値として追加されます。入力および出力パラメータが定義されている場合は，それらがローカル・パラメータに変換されます。

注：現在 Quality Center，あるいはほかの QuickTest セッションで開いているコンポーネントはロックされ，読み取り専用形式でのみ開くことができます。これらのコンポーネントを対象に作業をするには，このコンポーネントを開いている Quality Center または別の QuickTest セッションは閉じなくてはなりません。

手動コンポーネントを開いて変換するには，次の手順を実行します。

1 QuickTest で，コンポーネントが保存されている Quality Center プロジェクトに接続します。Quality Center への接続の詳細については，46 ページ「Quality Center プロジェクトへの接続」を参照してください。

2 ［ビジネス コンポーネントを開く］ダイアログ・ボックスを開く。詳細については，490 ページ「［ビジネス コンポーネントを開く］ダイアログ・ボックス」を参照してください。

3 手動コンポーネントを選択します。手動コンポーネントは，アイコンの左隅に「**M**」の付いたコンポーネント・アイコンによって表されます。コンポーネントの名前が読み取り専用の［**コンポーネント名**］ボックスに表示されます。

ヒント：手動コンポーネントのみを表示するよう，コンポーネントのリストをフィルタすることもできます。詳細については，490 ページ「［ビジネス コンポーネントを開く］ダイアログ・ボックス」を参照してください。

---

4 ［**OK**］をクリックしてコンポーネントを開きます。手動コンポーネントをビジネス・コンポーネントに変換するかどうかを選択するメッセージが表示されます。

5 続けて変換を行うには，［**はい**］をクリックします。この変換処理は元には戻せません。

6 ［ビジネス コンポーネントの新規作成］ダイアログ・ボックスが開いたら，ビジネス・コンポーネントのためのアプリケーション領域を選択し，［**OK**］をクリックします。QuickTest がコンポーネントをダウンロードして開き，変換するのに応じて，実行される操作がステータス・バーに表示されます。［ビジネス コンポーネントの新規作成］ダイアログ・ボックスの詳細については，485 ページ「［ビジネス コンポーネントの新規作成］ダイアログ・ボックス」を参

照してください。アプリケーション領域の詳細については，Chapter 第 13 章 , “アプリケーション領域の管理 ” を参照してください。

これで，ほかのコンポーネントと同様にコンポーネントを使用できるようになります。Quality Center のマニュアル・ランナーを使用してビジネス・コンポーネントを手動コンポーネントとして実行できるように，手動ステップをさらに追加したり，既存の手動ステップを変更したりできます。詳細については，502 ページ「コンポーネントの手動ステップの追加および変更」を参照してください。

コンポーネントの手動ステップの追加および変更

手動コンポーネントを自動コンポーネントに変換すると，手動コンポーネントの各手動ステップがキーワード・ビューの ManualStep 操作に変換されます。

項目	操作	値
操作	ManualStep	"Step1","ユーザ名を入力"
操作	ManualStep	"Step2","パスワードを入力"
操作	ManualStep	"Step3","Enter ボタンを押下"

ステップ名，ステップの説明，および期待結果を変更するには，キーワード・ビューの該当する ManualStep 行の対応する引数値を変更します。

QuickTest では，変換後のコンポーネントに新しいステップを追加できます（通常のビジネス・コンポーネント・ステップ，および ManualStep 操作も追加できます）。Quality Center の［自動化］タブでキーワード方式のステップを追加することもできます。ステップを必要に応じて削除することもできます。

QuickTest においてコンポーネントの ManualStep 操作および通常のキーワード駆動のステップに加えた変更はすべて，変更を保存した後に，Quality Center でコンポーネントの［ステップのデザイン］タブおよび［自動化］タブに反映されます（逆もまた同様です）。このため，Quality Center および QuickTest のどちらでコンポーネントを更新しても，必要に応じて引き続き Quality Center マニュアル・ランナーを使用してコンポーネントを手動で実行できます。

キーワード・ビューにおけるステップの追加の全般的な情報については，515 ページ「キーワード・ビューを使った作業」を参照してください。ManualStep 操作の詳細については，『**HP QuickTest Professional Object Model Reference**』（英語版）の「**Utility**」を参照してください。

Quality Center でステップを追加する方法，および Quality Center マニュアル・ランナーを使って手動コンポーネントを実行する方法の詳細については，『**HP Business Process Testing ユーザーズ・ガイド**』を参照してください。

コンポーネントに関連付けらているアプリケーション領域の変更

QuickTest でビジネス・コンポーネントを作成するときは，コンポーネントを関連付ける対象となるアプリケーション領域を選択する必要があります。使用可能なアプリケーション領域が複数存在する場合もあります。コンポーネントに最も適したアプリケーション領域を選択してください。

アプリケーション，またはアプリケーション領域に関連付けられているリソース・ファイルや設定に対して，変更が加えられた場合，アプリケーション領域が不適切になることがあり，特定のコンポーネントに関連付けられているアプリケーション領域を変更しなければならないことがあります。たとえば，オブジェクト・リポジトリが変更されたり，アプリケーション領域から削除されたりする可能性があります。

あるいは，アプリケーションの開発過程で，現在関連付けられているオブジェクト・リポジトリに含まれていない追加の異なるオブジェクトが含まれることもあります。このような場合，コンポーネントやビジネス・プロセス・テストが正しく実行されなかったり，実行に失敗したりする可能性があります。

必要なリソース・ファイルや設定が別のアプリケーション領域に含まれている場合は，コンポーネントに関連付けられているアプリケーション領域を変更する必要があります。

注：QuickTest では，コンポーネントを開くたびに，コンポーネントに対して指定されているリソースが使用可能かどうかの確認が行われます。共有オブジェクト・リポジトリが見つからないなど，コンポーネントまたはアプリケーション領域が持つリソースが見つからない場合は，そのことが QuickTest によって［欠落リソース］表示枠に示されます。詳細については，第 35 章「欠落リソースの処理」を参照してください。

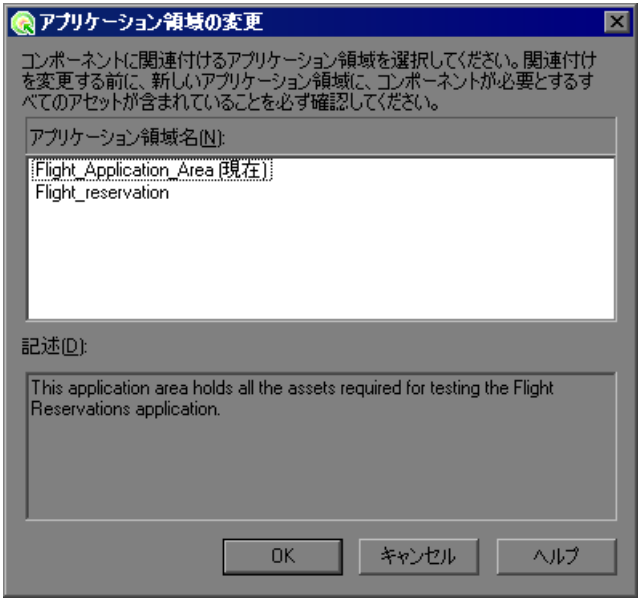
［アプリケーション領域の変更］ダイアログ・ボックスを使用して，コンポーネントに関連付けられているアプリケーション領域を変更します。詳細につい

ては，504 ページ「[アプリケーション領域の変更] ダイアログ・ボックス」を参照してください。

[アプリケーション領域の変更] ダイアログ・ボックス

説明	既存のコンポーネントをほかのアプリケーション領域と関連付けることができます。
アクセス方法	1 490 ページ「[ビジネス コンポーネントを開く] ダイアログ・ボックス」に記述された手順に従って，コンポーネントを開きます。 2 ［**ファイル**］＞［**アプリケーション領域の変更**］メニュー・コマンドを選択します。
さらに調べる	**概念の概要**：503 ページ「コンポーネントに関連付けらているアプリケーション領域の変更」

下記が［アプリケーション領域の変更］ダイアログ・ボックスの画像です



［アプリケーション領域の変更］ダイアログ・ボックスのオプション

オプション	説明
［アプリケーション領域名］	現在のコンポーネントを関連付けることのできる，使用可能なアプリケーション領域を一覧表示します。 **アプリケーション領域を変更するには，次の手順を実行します。**リストからアプリケーション領域を選択して［**OK**］をクリックします。
［記述］	アプリケーション領域ウィンドウの［一般］タブで定義した，アプリケーション領域の記述を表示します。詳細については，446 ページ「一般設定の定義」を参照してください。

コンポーネントの印刷

コンポーネントを表形式で印刷することができます。

コンポーネントを印刷するには，次の手順を実行します。

1 ［**印刷**］ボタンをクリックするか，［**ファイル**］＞［**印刷**］を選択します。標準の［印刷］ダイアログ・ボックスが開きます。

2 ［**OK**］をクリックすると，［キーワード ビュー］の内容が標準設定の Windows プリンタに出力されます。

ヒント：画面上で［キーワード ビュー］の印刷イメージを表示するには，［**ファイル**］＞［**印刷プレビュー**］を選択します。

第 16 章

スクリプト・コンポーネントの作成

スクリプト・コンポーネントは，特定のタスクを実行する，保守および再利用が可能なスクリプトです。スクリプト・コンポーネントは，テスト・アクションおよびビジネス・コンポーネントの両方に共通する機能を持っています。キーワード・ビュー，エキスパート・ビュー，その他の QuickTest ツールやオプションを使用して，QuickTest でスクリプト・コンポーネントの作成，表示，変更，デバッグが行えます。また，既存のビジネス・コンポーネントを，スクリプト・コンポーネントに変換できます。

本章の内容

- スクリプト・コンポーネントの使用について（508 ページ）
- スクリプト・コンポーネントの作成（510 ページ）
- スクリプト・コンポーネントへの変換（513 ページ）
- スクリプト・コンポーネントへのビジネス・コンポーネントの変換（514 ページ）

スクリプト・コンポーネントの使用について

スクリプト・コンポーネントを対象に作業をする際には，キーワード・ビュー，エキスパート・ビュー，および QuickTest ツールやオプションの機能を最大限に使用できます。たとえば，スクリプト・コンポーネントにメソッドや関数を追加するプロセスを案内するステップ・ジェネレータを使用できます。エキスパート・ビューを使って，標準 VBScript ステートメントや QuickTest オブジェクトやメソッドを使用するほかのプログラミング・ステートメントを手入力することによって，スクリプト・コンポーネントを拡張できます。また，スクリプト・コンポーネントのステップにユーザ定義関数を統合し，選択した項目をパラメータ化し，スクリプト・コンポーネントにチェックポイントと出力値を追加することもできます。

スクリプト・コンポーネントは，各分野のエキスパートの要求に合わせて作成できます（たとえば，ループや条件文などより複雑な機能を含むコンポーネントが必要な場合など）。Quality Center で作業する各分野のエキスパートは，こうしたスクリプト・コンポーネントをビジネス・プロセス・テストに含め，アプリケーションが希望どおりに動作することを確認できます。

スクリプト・コンポーネントを作成した後で，各分野のエキスパートは，Quality Center プロジェクトのビジネス・コンポーネント・モジュールでコンポーネントによって生成された自動ドキュメント（読み取り専用）を表示できます。各分野のエキスパートはスクリプト・コンポーネントを実行したり，ビジネス・プロセス・テストに追加したりできますが，変更が必要になった場合に QuickTest でスクリプト・コンポーネントの保守を行うのは引き続き QuickTest エンジニアです。スクリプト・コンポーネントは Quality Center では変更できません。

スクリプト・コンポーネントは，ビジネス・コンポーネントと同じように保存したり開いたりできます。詳細については，494 ページ「ビジネス・コンポーネントの保存」および 488 ページ「ビジネス・コンポーネントを開く」を参照してください。

スクリプト・コンポーネントとその他のテスト・ドキュメントとの類似点

スクリプト・コンポーネントには，QuickTest のアクションやテストと同じ機能がたくさんがあります。たとえば，次のことが行えます。

- エキスパート・ビューでのプログラム・ステートメントの使用（第 10 章「関数ライブラリ・ウィンドウを使用した作業」を参照）。
- チェックポイントおよび出力値の作成。
- 階層形式のキーワード・ビューの表示。
- 仮想オブジェクトの作成と操作。
- データ・テーブルを使用した複数の反復の実行。
- ActiveScreen を使用して，記録セッション中に特定のステップを実行したときに表示されていたアプリケーションのスナップショットを表示できます。また，アプリケーションが使用できない場合や，選択したオブジェクトに対応するステップがテストまたはコンポーネントにない場合でも，オブジェクト値をパラメータ化したり，ページ内の任意のオブジェクトを対象としたチェックポイント，メソッド，出力値を挿入したりできます。
- ランダム方式のパラメータおよび環境パラメータの使用。
- 記録または実行セッションの開始時にアプリケーションが起動するよう設定。

スクリプト・コンポーネントに利用できるすべての機能の詳細については，『**HP QuickTest Professional ユーザーズ・ガイド**』を参照してください。

スクリプト・コンポーネントは，次の点でビジネス・コンポーネントにも似ています。

- 特定のアプリケーション領域と関連付けられている。すべてのリソースをファイル・システムではなく Quality Center プロジェクトに格納する必要がある。
- ビジネス・プロセス・テストに組み込み可能なスタンドアロンのモジュラー化されたユニットである。
- ビジネス・プロセス・テストにおいて階層的ではなく直線的である。
- ネストされない。つまり別のコンポーネントを呼び出すことはできない。

スクリプト・コンポーネントの作成

QuickTest を Quality Center プロジェクトに接続すると，そのプロジェクトに新規のビジネス・コンポーネントを作成できます。

各スクリプト・コンポーネントは，特定のアプリケーション領域に基づいています。アプリケーション領域には，コンポーネントで使用されるリソースと設定が含まれています（例：共有オブジェクト・リポジトリや関数ライブラリの場所など）。作成するスクリプト・コンポーネントに最も適したアプリケーション領域を選択します。コンポーネントの保存先となる Quality Center プロジェクト内の任意のアプリケーション領域を選択できます。詳細については，429 ページ「アプリケーション領域の管理」を参照してください。

ヒント：ビジネス・コンポーネントをスクリプト・コンポーネントに変換することもできます。詳細については，514 ページ「スクリプト・コンポーネントへのビジネス・コンポーネントの変換」を参照してください。

注：まだアプリケーション領域を作成していない場合は，スクリプト・コンポーネントは Business Process Testing に用意されている標準設定のアプリケーション領域設定に基づきます。

スクリプト・コンポーネントを作成するには，次の手順を実行します。

1 ビジネス・コンポーネントの保存先となる Quality Center プロジェクトに接続します。詳細については，46 ページ「Quality Center プロジェクトへの接続」を参照してください。

2 ［**ファイル**］＞［**新規作成**］＞［**ビジネス コンポーネント**］を選択し，［ビジネス コンポーネントの新規作成］ダイアログ・ボックスを開きます。

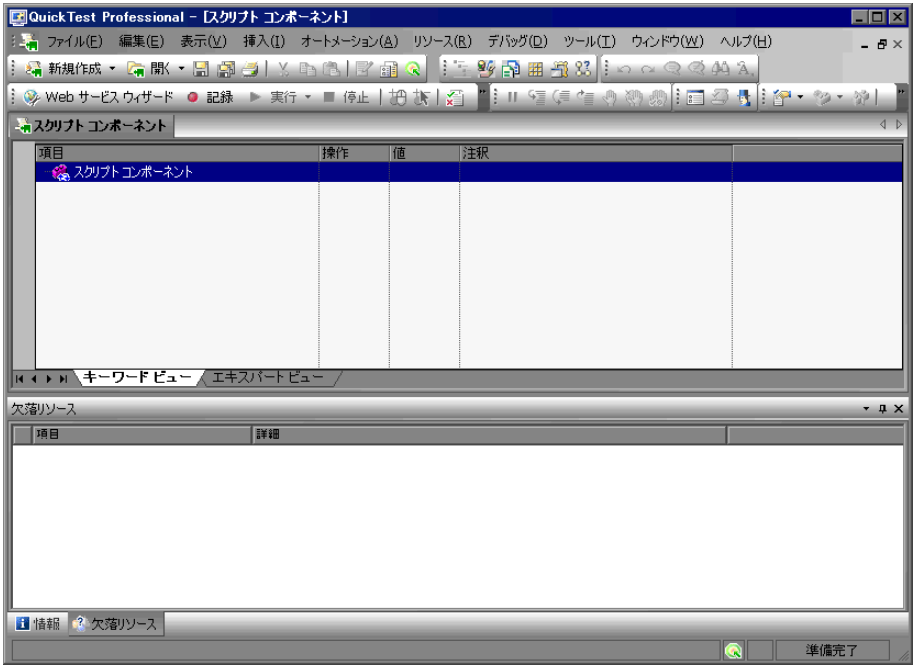
3 ［**アプリケーション領域名**］ボックスから適切なアプリケーション領域を選択します。たとえば，フライト予約モジュール用のスクリプト・コンポーネントを作成する場合は，そのために定義されたアプリケーション領域を選択します。［**OK**］をクリックします。

詳細については，485 ページ「[ビジネス コンポーネントの新規作成］ダイアログ・ボックス」を参照してください。

注：まだアプリケーション領域を作成していない場合は，Business Process Testing に用意されている標準の設定を使用して，無題の新しいスクリプト・コンポーネントが開きます。

ヒント：すでにスクリプト・コンポーネントが開いている場合は，ツールバーの［**新規作成**］ボタンをクリックして新規のスクリプト・コンポーネントを開くこともできます。

無題の新しいスクリプト・コンポーネントが開きます。新規のスクリプト・コンポーネントにはまだ内容はありませんが，基盤とするアプリケーション領域に定義されている必要な設定とリソースはすべて含まれています。



4 この時点で，次の操作を実行できます。

- QuickTest で提供されている機能とオプションを使用して，スクリプト・コンポーネントに内容を追加します。たとえば，次のような場合です。
  - たとえば，エキスパート・ビューで，標準的な VBScript ステートメントを手入力したり，QuickTest オブジェクトやメソッドを使ってステートメントを追加したりできます。
  - ステップ・ジェネレータを使って，プログラミング・ロジックを含んだステップを追加できます。
  - また，スクリプト・コンポーネントには，チェックポイントや出力値も追加できます。

スクリプト・コンポーネントの作成時に使用する機能の詳細については，**HP QuickTest Professional ユーザーズ・ガイド**を参照してください。

➤ スクリプト・コンポーネントを保存します。（内容は後で追加できます。）スクリプト・コンポーネントは，ビジネス・コンポーネントと同じように保存できます。詳細については，494 ページ「ビジネス・コンポーネントの保存」を参照してください。

注：Quality Center のビジネス・コンポーネント・モジュールの［ステップのデザイン］タブでは，各分野のエキスパートが表示したり，作業の対象にしたりできるのは，スクリプト・コンポーネント向けに定義されている手動ステップ（もしあれば）のみです。［自動化］タブの［**起動**］ボタンをクリックしてスクリプト・コンポーネントを QuickTest 内で開かないかぎり（QuickTest が Quality Center のクライアント上にインストールされている場合），各分野のエキスパートは，自動化されたステップを表示したり変更したりできません。詳細については，『**HP Business Process Testing ユーザーズ・ガイド**』を参照してください。

スクリプト・コンポーネントへの変換

必要に応じて，ビジネス・コンポーネントおよびアクションを，スクリプト・コンポーネントに変換できます。ビジネス・プロセス・テストを使用する場合，一般に，既存のビジネス・コンポーネントをスクリプト・コンポーネントに変換するよりも，Quality Center で新しいビジネス・コンポーネントを作成することをお勧めします。これにより，Quality Center で作業する各分野のエキスパートが，時とともにコンポーネントを保守できるからです。さらに，スクリプト・コンポーネントは QuickTest においてのみ変更できるため（Quality Center では変更できません），各分野のエキスパートは自動化されたステップを Quality Center で表示できませんが，手動ステップは（もしあれば）表示および変更が可能です。この変換は元に戻せません。つまり，スクリプト・コンポーネントを，元のビジネス・コンポーネントに変換することはできません。

スクリプト・コンポーネントへのビジネス・コンポーネントの変換

1 つのビジネス・コンポーネントをスクリプト・コンポーネントに変換できます。

ビジネス・コンポーネントをスクリプト・コンポーネントに変換するには，次の手順を実行します。

1 スクリプト・コンポーネントに変換するビジネス・コンポーネントを開きます。ビジネス・コンポーネントを開く方法の詳細については，488 ページ「ビジネス・コンポーネントを開く」を参照してください。

注： ビジネス・コンポーネントを読み取り専用モードで開いている場合，またはロックされている場合は，当該コンポーネントはスクリプト・コンポーネントに変換できません。

---

2 ［**ファイル**］＞［**スクリプト コンポーネントに変換**］を選択します。

3 確認を求められたら［**OK**］をクリックし，変換処理を進めます。

注： この操作で既存のビジネス・コンポーネントはスクリプト・コンポーネントに置き換えられます。取り消しはできません。

変換が完了すると，新しいスクリプト・コンポーネントが自動的に開きます。

第 17 章

キーワード・ビューを使った作業

キーワード・ビューでは，グラフィカルかつ使いやすい形式で，テストを簡単に，作成，表示，変更できます。

本章の内容

- キーワード・ビューを使った作業について（516 ページ）
- キーワード・ビュー（518 ページ）
- コンポーネントへのステップの追加（523 ページ）
- コンポーネントへのその他のタイプのステップの追加（541 ページ）
- ステップの部分の変更（541 ページ）
- コメントの使用（551 ページ）
- コンポーネントステップの管理（553 ページ）
- キーワード・ビューでのキーボード・コマンドの使用（554 ページ）
- キーワード・ビューの表示オプションの定義（555 ページ）
- キーワード・ビューのブレークポイントを使った作業（561 ページ）

キーワード・ビューを使った作業について

ビジネス・コンポーネント・キーワード・ビューでは，モジュール化されたテーブル形式で，コンポーネントのステップを作成および表示できます。各ステップはキーワード・ビューの行で，変更可能な独立した部分から構成されています。ステップを作成および変更するには，キーワード・ビューの項目と操作を選択し，必要に応じて情報を入力します。完了した各ステップは自動的にドキュメント化されるため，コンポーネントの記述はわかりやすい文で表示されます。必要に応じて，これらの記述を，手動テストの指示として使用することもできます。

キーワード・ビューを使用して，コンポーネントへの新しいステップの追加や，既存のコンポーネント・ステップの表示，変更，およびデバッグができます。ステップを追加または変更する場合，テスト・オブジェクトやほかのステップに対して必要なその他のステップのタイプを選択し，実行するメソッドまたは関数の処理を選択して，選択した操作またはステートメントに必要な値を定義します。

一般に，各分野のエキスパートは Quality Center ビジネス・コンポーネント・モジュールの［自動化］タブを使用して，コンポーネント・ステップに内容を追加したり，コンポーネント・ステップを変更したりします。ただし，これは QuickTest でもできます。QuickTest に表示されるビジネス・コンポーネント・キーワード・ビューは，各分野のエキスパートが Quality Center で使用する［自動化］タブと同じです。

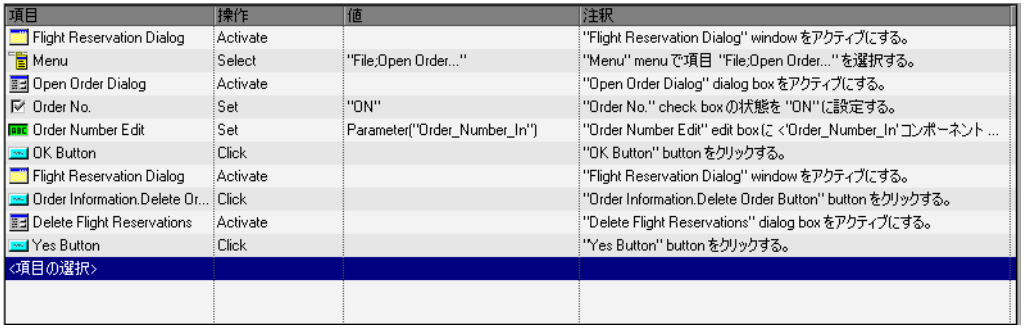
ビジネス・コンポーネント・キーワード・ビューは，ビジネス・コンポーネントを簡単に作成できるようにするために特別に設計されたコンポーネント専用のオプションを提供している点が，QuickTest テスト・キーワード・ビューと異なります。ビジネス・コンポーネント・キーワード・ビューはこの違いにより，各分野のエキスパートにとってビジネス・コンポーネントを Quality Center で作成するための簡単で直観的なツールとなっています

ビジネス・コンポーネント・キーワード・ビューには，コメントを含めることができます。これにより，ビジネス・コンポーネントに手動ステップや情報セパレータを入力できます。キーワード・ビュー内の項目（テスト・オブジェクトと操作）はすべて，たとえ先に実行されるステップまたは操作の下位のオブジェクトであっても，同じ階層レベルに表示されます。これにより，各分野のエキスパートがビジネス・コンポーネントのステップを管理しやすくなっています。

ビジネス・コンポーネント・キーワード・ビューで作業を行うには，Business Process Testing のサポートが組み込まれた Quality Center プロジェクトに QuickTest を接続する必要があります。

キーワード・ビュー

キーワード・ビューでは，コンポーネントのステップをキーワード駆動のモジュール化された表形式で作成および表示できます。ビジネス・コンポーネント・キーワード・ビューは，テーブル形式のビューで構成されています。各ステップはテーブル内の独立した行で，各カラムはステップのさまざまな部分を表します。表示されるカラムは，選択に応じて異なります。詳細については，555 ページ「キーワード・ビューの表示オプションの定義」を参照してください。



項目	操作	値	注釈
Flight Reservation Dialog	Activate		"Flight Reservation Dialog" window をアクティブにする。
Menu	Select	"File;Open Order..."	"Menu" menu で項目 "File;Open Order..." を選択する。
Open Order Dialog	Activate		"Open Order Dialog" dialog box をアクティブにする。
Order No.	Set	"ON"	"Order No." check box の状態を "ON" に設定する。
Order Number Edit	Set	Parameter("Order_Number_In")	"Order Number Edit" edit box に <'Order_Number_In' コンポーネント ...
OK Button	Click		"OK Button" button をクリックする。
Flight Reservation Dialog	Activate		"Flight Reservation Dialog" window をアクティブにする。
Order Information.Delete Or...	Click		"Order Information.Delete Order Button" button をクリックする。
Delete Flight Reservations	Activate		"Delete Flight Reservations" dialog box をアクティブにする。
Yes Button	Click		"Yes Button" button をクリックする。
<項目の選択>			

新規のビジネス・コンポーネントを作成すると，キーワード・ビューは次の図のように空です。

項目	操作	値	出力	注釈

コンポーネントにステップを追加すると，各ステップがキーワード・ビュー内の単一の行として定義されます。ステップは，現在選択されているステップの下，既存のコンポーネントの最後，または新しいコンポーネントの先頭に追加できます。また，標準チェックポイント，ビットマップ・チェックポイント，出力値，およびコメントも入力できます。

ステップ：ステップは，実行される操作を表します。ステップを作成した後，その内容を指定します。たとえば，ステップの実行対象であるテスト・オブジェクトを選択したり，ステップで実行される操作を指定したり，関連する入力値や出力値を指定することができます。Quality Center でビジネス・プロセス・テストが実行されると，関連するビジネス・コンポーネントで定義されたステップが実行されます。この項では，ビジネス・コンポーネントにステップを追加する方法を説明します。

コメント：コメントは，行全体にわたって自由に入力できるテキストです。 アイコンはコメントを示します。コメントを使って，手動ステップを定義したり，ビジネス・コンポーネント内で隣り合うステップに関する情報を入力したりできます。コメントは，ビジネス・プロセス・テストが実行されるときには処理されません。詳細については，551 ページ「コメントの使用」を参照してください。

ステップは，コンポーネントに手作業で追加することも，アプリケーションに対して実行したステップを記録することで追加することもできます。各ステップは，キーワード・ビューに行として挿入されます。たとえば，キーワード・ビューには次の行が含まれる場合があります。

項目	操作	値	出力	注釈
Login Dialog	Activate			"Login Dialog" dialog box をアクティブにする。
Agent Name:	Set	Parameter("Ag...		"Agent Name:" edit box に 'AgentName' コンポーネント パラメータ ...
Password:	SetSecure	Parameter("Ag...		暗号化された文字列 'AgentPassword' コンポーネント パラメータ の...
Login Dialog	Activate			"Login Dialog" dialog box をアクティブにする。
OK	Click			"OK" button をクリックする。

これらの行は，Mercury Tours サンプル Web サイトの「**Welcome: Mercury Tours**」ページで実行される次の 3 つのステップを示します。

- **userName** エディット・ボックスに tutorial と入力されます。
- **password** エディット・ボックスに暗号化文字列が入力されます。
- **Sign-In** 画像がクリックされます。
- ［**注釈**］カラムでは，各ステップがわかりやすい文に変換されます。

キーワード・ビューを使用すると，コンポーネントの任意の場所にステップを追加できます。ステップを追加したら，標準の編集コマンドやドラッグ・アンド・ドロップ機能を使用して，ステップの変更や削除ができます。［キーワード ビュー］の内容は，Windows の標準プリンタで印刷できます（印刷する前に内容をプレビューすることもできます）。詳細については，505 ページ「コンポーネントの印刷」を参照してください。

キーワード・ビューでは，チェックポイントおよび出力値のプロパティの表示もできます。

ビジネス・コンポーネント・キーワード・ビューには，［**項目**］，［**操作**］，［**値**］，［**出力**］，および［**注釈**］のうちのいくつかのカラムを含めることができます。以下では各カラムについて簡潔に説明します。

［項目］カラム

ステップを実行する対象の項目 &emds; テスト・オブジェクトまたはユーザ定義関数（**操作**）。ステップに内容を追加する前に，［**項目**］カラムからオプションを選択する必要があります。詳細については，525 ページ「ステップの項目の選択」を参照してください。

［操作］カラム

項目に対して実行する操作。このカラムには，**Click** や **Select** など，［**項目**］カラムで選択した項目に対して実行できる，すべての使用可能な操作（メソッド，関数，プロパティ，またはサブプロシージャ）のリストが含まれています。標準設定で，［**項目**］カラムで選択した項目に対して最もよく使用される操作が表示されます。詳細については，534 ページ「ステップの操作の選択」を参照してください。**RegisterUserFunc** メソッドを使用して，テスト・オブジェクトに対して追加の操作を定義できます。詳細については，389 ページ「ユーザ定義関数および関数ライブラリを使用した作業」を参照してください。

［値］カラム

選択された操作の引数値の内容。［**値**］セルは，［**操作**］カラムで選択したオプションの引数の数に従って分割されています。

引数に値のリストが事前に定義されている場合，QuickTest は使用可能な値のドロップダウンリストを提供します。値のリストが提供されない場合は，このボックスに値を手作業で入力できます。

選択されたオプションに応じて，定数，**ローカル・**パラメータ，または**コンポーネント・**パラメータを引数として指定できます。

ローカル・パラメータ：ローカル・パラメータは当該のビジネス・コンポーネントに固有のパラメータで，そのコンポーネントしかアクセスできません。ローカル・パラメータは，単一のステップ内で，またはコンポーネント・ステップ間で使用されます。たとえば，あるステップの出力パラメータを後続の

ステップの入力パラメータとして使用できます。詳細については，542 ページ「パラメータの使用」を参照してください。

コンポーネント・パラメータ：コンポーネント・パラメータは，Quality Center プロジェクト内のどのコンポーネントからでもアクセスできるパラメータです。詳細については，660 ページ「コンポーネントのパラメータの定義」を参照してください。

［出力］カラム

ステップの出力値が格納されるパラメータ。たとえば cCols という名前の出力パラメータを選択すると，現在のステップの出力値は **cCols** パラメータに格納されます。出力パラメータに格納された値は，同じコンポーネント内で後ほど入力パラメータとして使用できます。［**値**］カラム同様，出力パラメータを指定する場合には，2 種類のパラメータ（ローカル・パラメータまたはコンポーネント・パラメータ）を使用できます。

［注釈］カラム

自動的に生成される読み取り専用のドキュメント。「Click the "Sign-in" image.」や「Select "San Francisco" in the "toPort" list」など，わかりやすい文でステップの役割が表示されます。ステップだけを印刷または表示する場合は，このカラムだけを表示できます。たとえば，手動テストの指示だけを印刷または表示できます。

ヒント：

- カラム・ヘッダ行を右クリックして表示されるメニューから［**ドキュメントのみ**］を選択すると，コンポーネントの［**注釈**］カラムのみ表示できます。
- また，注釈をコピーすることもできます。それには，［**編集**］＞［**ドキュメントをクリップボードにコピー**］を選択するか，カラム・ヘッダ行を右クリックして表示されるメニューから［**ドキュメントをクリップボードにコピーする**］を選択し，必要に応じて，別のアプリケーションに注釈を貼り付けます。

注： これらのカラムの一部がキーワード・ビューに表示されていない場合は，［キーワード ビュー オプション］ダイアログ・ボックスを使用してそれらを表示できます。詳細については，555 ページ「キーワード・ビューの表示オプションの定義」を参照してください。

キーワード・ビューを使った作業のヒント

- 左矢印と右矢印を使用して，セルのフォーカスを左と右に移動できますが，次の例外があります。
  - ［**項目**］カラムで左矢印と右矢印を押すと，項目にサブ項目が含まれていれば折りたたんだり展開したりできます。項目にサブ項目が含まれていなければ，矢印キーはほかのカラムと同じように使用できます。
  - 値またはコメントを変更するなどでセルが編集モードの場合は，左矢印キーと右矢印キーで編集中のセルの中を移動できます。
- ［**値**］セルが選択されているときに CTRL + F11 キーを押すと，［値設定オプション］ダイアログ・ボックスが開きます。
- ステップの左をクリックしてステップ全体が選択されている場合は，［**項目**］ツリーを展開したり折りたたんだりするのに **+** キー（特定の分岐を展開），**-** キー（特定の分岐を折りたたみ），***** キー（すべての分岐を展開）を使用できます。
- 特定のカラムではなく行が選択されている場合は，最初の 1 文字を入力すればその文字で始まる次の行にジャンプできます。

注： 上記のコマンド以外に，QuickTest のショートカット・メニューも使用できます。詳細については，68 ページ「QuickTest コマンドの実行」を参照してください。

コンポーネントへのステップの追加

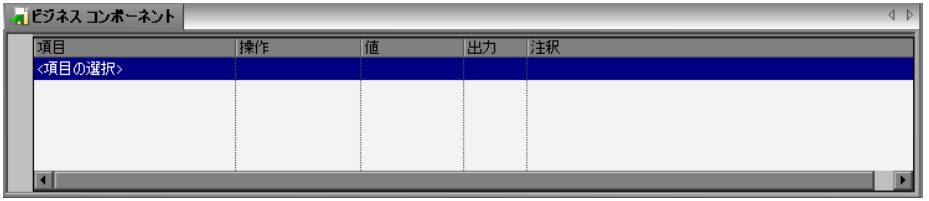
コンポーネント内の任意の場所にステップを追加できます。ステップの追加は，現在選択されているステップの下，コンポーネントの最後，または新しいコンポーネントの先頭に対して追加できます。

ステップを追加するには，次の手順を実行します。

1 次のいずれかを実行します。

- キーワード・ビュー内の任意の場所（既存のステップが存在する場合はその下）をクリックしてコンポーネントの最後にステップを追加します。まだステップを定義していなければ，ここで追加するステップがコンポーネントの最初のステップになります。
- ［**挿入**］＞［**ステップの新規作成**］を選択し，（存在する場合は）既存のステップの後ろに新しいステップを追加します。コンポーネントにステップが含まれていなければ，ここで追加するステップがコンポーネントに最初のステップになります。
- 既存のステップを選択して，［**挿入**］＞［**ステップの新規作成**］を選択し，既存のステップの間に新しいステップを追加します（最後のステップを選択した場合，コンポーネントの最後にステップが追加されます）。
- 既存のステップを右クリックして，ショートカット・メニューから［**新規ステップの挿入**］を選択します
- テスト・オブジェクトを［使用可能なキーワード］表示枠からキーワードまたはエキスパート・ビューにドラッグ・アンド・ドロップします。

キーワード・ビューの選択したステップの下に新しいステップが追加されます。



注：通常［**項目の選択**］リストは，対応するすべてのテスト・オブジェクトのほか，［**操作**］項目および［**コメント**］項目を表示するように展開されます。

---

2 ステップの一部として変更が必要なセルの中をクリックし，後述するように内容を指定して，ステップを定義します。ステップ行の各セルは，ステップの異なる部分を表します。各ステップに対して，次のものが定義できます。

- **項目**：ステップ実行の対象となるテスト・オブジェクト，またはユーザ定義関数（**操作**）。ステップに内容を追加する前に，［**項目**］カラムからオプションを選択する必要があります。詳細については，525 ページ「ステップの項目の選択」を参照してください。

  項目を指定する代わりに，「**コメント**」を追加することもできます。その場合は，マニュアル・ステップや，それ以外の任意のテキスト情報を，ステップ間に追加できます。詳細については，551 ページ「コメントの使用」を参照してください。

- **操作**：項目に対して実行される操作。詳細については，534 ページ「ステップの操作の選択」を参照してください。
- **値**（該当する場合）：選択した操作の引数値。詳細については，535 ページ「ステップの引数の値の定義」を参照してください。
- **コメント**（該当する場合）：ステップに関するテキスト形式のメモ。詳細については，551 ページ「コメントの使用」を参照してください。
- **出力**（該当する場合）：ステップの出力値が格納されるパラメータ。詳細については，538 ページ「ステップの出力値の定義」を参照してください。

注：［**注釈**］セルは読み取り専用です。このセルには，ステップの役割に関する説明が「Click the "Sign-in" image.」や「Select "San Francisco" in the "toPort" list」など，わかりやすい文で表示されます。ほとんどの場合，このセルに表示される説明は QuickTest によって自動的に生成されます。関数ライブラリを作成し，アプリケーション領域に追加した（関連付けた）場合，関数ライブラリの中で注釈のテキストが定義されていると，QuickTest には関数ライブラリの注釈しか表示されません。詳細については，414 ページ「関数への説明の追加」および 389 ページ「ユーザ定義関数および関数ライブラリを使用した作業」を参照してください。

ヒント：［**編集**］メニューまたはショートカット・メニューの標準的な編集コマンド（［**切り取り**］，［**コピー**］，［**貼り付け**］，［**削除**］）を使用して，ステップの定義や変更を簡単に行うことができます。また，ドラッグ・アンド・ドロップ操作で，ステップをコンポーネント内の別の場所に移動することもできます。詳細については，553 ページ「コンポーネントステップの管理」および 554 ページ「キーワード・ビューでのキーボード・コマンドの使用」を参照してください。

---

3 変更が終わったら，コンポーネントを Quality Center プロジェクトに保存します。詳細については，494 ページ「ビジネス・コンポーネントの保存」を参照してください。

ステップの項目の選択

項目は，共有オブジェクト・リポジトリ内のテスト・オブジェクト，またはユーザ定義関数「**操作**」です（［**操作**］項目は，コンポーネントのアプリケーション領域に関連付けられている関数ライブラリにユーザ定義関数が追加されている場合にのみ使用できます。詳細については，460 ページ「関数ライブラリの関連付け」および 389 ページ「ユーザ定義関数および関数ライブラリを使用した作業」を参照してください）。

項目を選択した後，その項目に対する操作を指定します。詳細については，534 ページ「ステップの操作の選択」を参照してください。

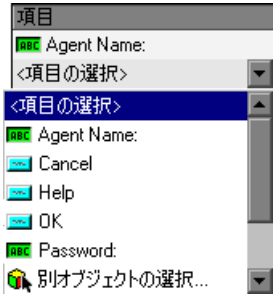
注：［**項目**］セルで項目または［**操作**］を選択するほかに，［**コメント**］を選択することもできます。［コメント］を選択すると，選択したステップが行全体にまたがる自由テキスト・セルに変換されます。ステップはコメントに変換されると，ステップに復元することはできません。手動ステップの入力，または隣接するステップに関する情報の提供を行う際に，［**コメント**］オプションを使用します。詳細については，551 ページ「コメントの使用」を参照してください。

本項では，次の項目について説明します。

- 526 ページ「［項目］リストからのテスト・オブジェクトの選択」
- 527 ページ「共有オブジェクト・リポジトリからのテスト・オブジェクトの選択」
- 529 ページ「アプリケーションからのテスト・オブジェクトの選択」
- 533 ページ「［操作］項目からの選択」

［項目］リストからのテスト・オブジェクトの選択

［**項目**］リストに表示されるテスト・オブジェクトは，前のステップのテスト・オブジェクトの兄弟または子のテスト・オブジェクトです（共有オブジェクト・リポジトリで定義されています）。次の例は，**userName** テスト・オブジェクトの後に続くステップで使用できるオブジェクトを示しています。



表示された［項目］リストからテスト・オブジェクトを選択するには，次の手順を実行します。

1 ［**項目**］セルをクリックし，矢印ボタンをクリックして，［**項目**］リストを表示します。新しいステップを作成した直後であれば，新しいステップを作成するとすぐに，リストが自動的に表示されます。

2 ［**項目**］リストで，ステップの実行対象にするテスト・オブジェクトを選択します。選択した項目が［**項目**］セルに表示されます。次に，ステップに対して操作を指定する必要があります。詳細については，534 ページ「ステップの操作の選択」を参照してください。

共有オブジェクト・リポジトリからのテスト・オブジェクトの選択

共有オブジェクト・リポジトリには，コンポーネントの基本となっているアプリケーション領域で定義されたテスト・オブジェクトがすべて含まれています(前述の［**項目**］リストに表示されるものも含みます)。

新しいステップのオブジェクト・リポジトリ・ツリーでは任意のオブジェクトを選択できます。オブジェクト・リポジトリが非常に大きい場合，オブジェクトを検索できます。たとえば，password オブジェクトを追加するとします。このオブジェクトはエディット・ボックスであることがわかっています。すべての **Edit** タイプのオブジェクトの中から password という名前のオブジェクトを検索できるほか，名前に文字 p を含むオブジェクトを検索することも可能です。

オブジェクト・リポジトリの詳細については，第 5 章「オブジェクト・リポジトリ内のテスト・オブジェクトの管理」を参照してください。Object ステートメントの詳細については，372 ページ「ネイティブ・プロパティおよび操作へのアクセス」を参照してください。

共有オブジェクト・リポジトリからテスト・オブジェクトを選択するには，次の手順を実行します。

1 ［**項目**］セルをクリックし，矢印ボタンをクリックして，［**項目**］リストを表示します。新しいステップを作成した直後であれば，新しいステップを作成するとすぐに，リストが自動的に表示されます。

2［**項目**］リストで，［**別オブジェクトの選択**］を選択します。［ステップでオブジェクトを選択］ダイアログ・ボックスが開きます。

3 オブジェクト・リポジトリ・ツリーからオブジェクトを選択します。オブジェクト・リポジトリが非常に大きい場合，次の説明に従ってオブジェクトを検索できます。検索が不要な場合は，手順 8 に進みます。

4［**名前**］ボックスに，オブジェクトの名前，または名前の一部を入力します。たとえば，p という文字が含まれるすべてのオブジェクト名を検索するには，p を入力します。

注：［**名前**］ボックスが空白のままの場合は，選択したオブジェクトのタイプのすべてのオブジェクトが条件と一致しているとみなされます。

5［**タイプ**］ボックスでは，検索するオブジェクトのタイプを選択します。すべてのオブジェクト・タイプのオブジェクトを検索するには，**＜ All ＞**を選択します。

注：このリストに示される各オブジェクト・タイプは，オブジェクトを一般的な特性に基づいてグループ分けした一般的な分類です。たとえば，［**List**］タイプには，リスト・オブジェクトやリスト・ビュー・オブジェクトに加え，コンボ・ボックスが含まれます。［**Table**］タイプには，テーブルとグリッドの両方が含まれ，［**Miscellaneous**］タイプには WebElement と WinObject などさまざまなオブジェクトが含まれます。

6［**次を検索**］ボタンをクリックします。現在選択されているノードから検索が開始され，条件に一致したオブジェクトの数が表示されます。リスト内で条件に一致した最初のオブジェクトが強調表示されます。

7 必要に応じて［**次を検索**］ボタンをクリックすることで，検索条件に一致したすべてのオブジェクトを順に確認していくことができます。検索はツリーの末尾まで続き，それから，ツリーの先頭に戻って続行されます。

ヒント：検索条件に一致する次のオブジェクトを見つけるには，F3 キーを押します。前の一致を見つけるには，SHIFT+F3 キーを押します。

8［**OK**］をクリックします。オブジェクトがキーワード・ビューの［**項目**］カラムに表示され，［**項目**］リストにも追加されます。これで，選択したオブジェクトに対する操作を指定できます。詳細については，534 ページ「ステップの操作の選択」を参照してください。

アプリケーションからのテスト・オブジェクトの選択

ステップに必要なテスト・オブジェクトが共有オブジェクト・リポジトリに含まれていない場合は，アプリケーションから直接選択して共有オブジェクト・リポジトリに追加すれば，当該のステップやほかのステップでそのテスト・オブジェクトを使用できるようになります。

アプリケーションからテスト・オブジェクトを追加するには，次の手順を実行します。

1 ［**項目**］セルをクリックし，矢印ボタンをクリックして，［**項目**］リストを表示します。新しいステップを作成した直後であれば，新しいステップを作成するとすぐに，リストが自動的に表示されます。

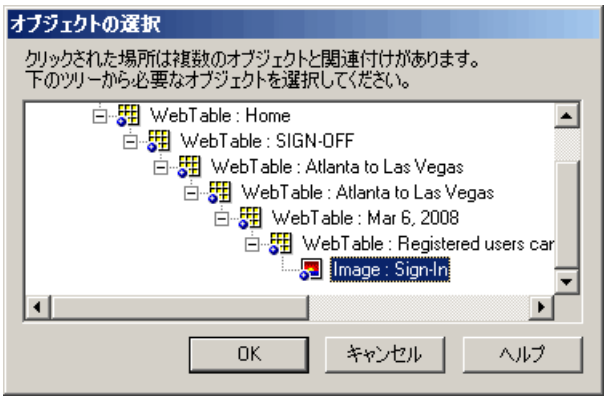
2 ［**項目**］リストで，［**リポジトリからのオブジェクト**］を選択します。［ステップでオブジェクトを選択］ダイアログ・ボックスが開きます。

3 指差しマークのボタンをクリックします。QuickTest が非表示になります。

4 指差しカーソルを使用して，アプリケーション内の必要なオブジェクトをクリックします。指差しマーク機能の使用方法の詳細については，532 ページ「指差しマークの使用法に関するヒント」を参照してください。

クリックした場所が複数のオブジェクトに関連付けられている場合は，［オブ

ジェクトの選択］ダイアログ・ボックスが開きます。



5 新しいステップのオブジェクトを選択し，［**OK**］をクリックします。オブジェクトが，［ステップでオブジェクトを選択］ダイアログ・ボックスの共有オブジェクト・リポジトリ・ツリーに表示されます。

6 ［**OK**］をクリックします。オブジェクトがキーワード・ビューの［**項目**］カラムに表示されます。これで，選択したオブジェクトに対する操作を指定できます。詳細については，534 ページ「ステップの操作の選択」を参照してください。

注：Quality Center を使用している各分野のエキスパートは，（コンポーネントのアプリケーション領域にある）共有オブジェクト・リポジトリに格納されているオブジェクトだけを選択できます。

ヒント：

- 共有オブジェクト・リポジトリ内に存在しないアプリケーションのオブジェクトを選択した場合，新しいステップを挿入すると，テスト・オブジェクトがローカル・オブジェクト・リポジトリに追加されます。新規テスト・オブジェクトをローカル・オブジェクト・リポジトリに追加した後，その名前が用途を明確に表していない場合は名前を変更することをお勧めします。たとえば，Edit（ユーザ名の入力に使用）という名前を UserName に変更できます。これにより，各分野のエキスパートは，共有オブジェクト・リポジトリに含まれるテスト・オブジェクトを使用するステップを追加する場合に適切なテスト・オブジェクトを選択できます。
- 必要なオブジェクトをローカル・オブジェクト・リポジトリに追加したら，［オブジェクト・リポジトリ ― 結合ツール］を使用して共有オブジェクト・リポジトリを更新して，ほかのコンポーネントが新しいオブジェクトを使用できるようにします。詳細については，283 ページ「ローカル・オブジェクト・リポジトリから共有オブジェクト・リポジトリを更新」を参照してください。
- コンテナ・テスト・オブジェクトを追加している場合，コンテキストを指定する方法もお勧めします。たとえば，ログイン・ページからの確認メッセージを追加する場合，「**ログイン ＞ 確認**」という名前を付けます。詳細については，182 ページ「テスト・オブジェクトの名前の変更」を参照してください。

指差しマークの使用法に関するヒント

- 左の CTRL キーを押したままにすると，指差しマークが標準ポインタに変わります。そのときに，QuickTest またはアプリケーションでウィンドウのフォーカスを変更したり，操作（右クリックしたり，スクロール・バーを使ったり，ポインタをオブジェクト上に移動してショートカット・メニューを表示するなど）を実行できます。
- 選択するオブジェクトがあるウィンドウが別のウィンドウで部分的に隠れている場合は，部分的に隠れているウィンドウが前面に表示されるまで，その上に指差しマークを数秒間置いたままにします。次に，必要なオブジェクトをポイントしてクリックします。ウィンドウを前面に出すのに必要な時間の設定は，［オプション］ダイアログ・ボックスの［一般］表示枠で行えます。

- 選択するオブジェクトがあるウィンドウが別のウィンドウで完全に隠れている場合，またはダイアログ・ボックスがウィンドウの背後に隠れている場合は，必要に応じて左の CTRL キーを押しながらウィンドウを配置します。
- 選択するオブジェクトがあるウィンドウが最小化されている場合は，左の CTRL キーを押したまま，Windows タスク・バーでアプリケーションを右クリックして，ショートカット・メニューから［**元のサイズに戻す**］を選択すると，そのウィンドウを表示できます。
- 特定のイベント（ショートカット・メニューを表示するために右クリックしたり，オブジェクト上にポインタを移動するなど）を実行するだけで，選択するオブジェクトを表示できる場合は，左の CTRL キーを押したままにします。指差しマークが一時的に標準ポインタに変わり，イベントを実行できます。選択するオブジェクトが表示されたら，左の CTRL キーを放します。ポインタが再び指差しマークに変わります。

［操作］項目からの選択

ビジネス・コンポーネントが，少なくとも 1 つの関数ライブラリを参照しているアプリケーション領域に基づいている場合，［**操作**］項目を選択して，ステップで使用する関数を選択できます。

ユーザ定義関数を使用すると，さまざまな追加操作を実行できます。たとえば，ビジネス・コンポーネントの開始時点でアプリケーションを開いたり，特定のプロパティの値をチェックしたりできます。

注：コンポーネントが基盤にしているアプリケーション領域が関数ライブラリに関連付けられていない場合には，［**項目**］リストに［**操作**］オプションは表示されません。

［操作］項目を選択するには，次の手順を実行します。

1 ［**項目**］セルをクリックし，矢印ボタンをクリックして，［**項目**］リストを表示します。新しいステップを入力した直後である場合は，新しいステップを作成するとすぐに，リストが自動的に表示されます。

2 ［**項目**］リストで，［**操作**］を選択します。［**操作**］項目が［**項目**］セルに表示されます。次に，ステップに対して操作を指定する必要があります。詳細については，534 ページ「ステップの操作の選択」を参照してください。

ステップの操作の選択

［**操作**］セルでは，［**項目**］カラムに表示された項目に対して実行する操作を指定します。使用可能な操作は，［**項目**］カラムで選択した項目に応じて異なります。項目を選択すると，その項目に（アプリケーション領域を通じて）関連する操作（キーワード）がすべて表示されます。

たとえば，WebButton オブジェクトなどのブラウザ・テスト・オブジェクトを選択した場合は，リストには，コンポーネントのアプリケーション領域の［キーワード］表示枠で使用可能なキーワードのリストから WebButton オブジェクトに対して選択された使用可能なメソッドがすべて含まれます。［**項目**］カラムで［**操作**］を選択した場合，一覧にはビジネス・コンポーネントに（アプリケーション領域を通じて）関連付けられている関連ライブラリ・ファイルの中で定義されたユーザ定義関数が表示されます。

関数ライブラリは，［アプリケーション領域］の関数ライブラリ表示枠で指定します。詳細については，460 ページ「関数ライブラリの関連付け」を参照してください。

ステップの操作を選択するには，次の手順を実行します。

［**操作**］セルをクリックします。その後下向き矢印ボタンをクリックして，項目に対して実行する操作を選択します。操作として，標準の操作またはユーザ定義関数を指定できます。ユーザ定義関数の詳細については，389 ページ「ユーザ定義関数および関数ライブラリを使用した作業」を参照してください。

注： リスト内の操作にカーソルを合わせると，その操作の実行内容を説明するツールチップが表示されます。ユーザ定義関数の場合，ツールチップは関連付けられている関数ライブラリに入力された注釈から取得されます。詳細については，414 ページ「関数への説明の追加」を参照してください。

［**項目**］カラムでテスト・オブジェクトを選択すると，［**操作**］カラムには，そのテスト・オブジェクト項目の選択されたすべての操作が（アプリケーション領域のキーワード表示枠での定義に従って）自動的に表示されます。選択したテスト・オブジェクトの［**操作**］リストには，製品にあらかじめ用意されている操作と，そのテスト・オブジェクト・タイプに登録されているユーザ定義関数が含まれます。

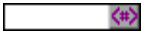
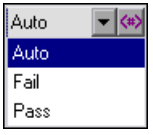
［**項目**］カラムで［**操作**］を選択すると，関数ライブラリ中に定義されている関数がアルファベット順に表示されます（コンポーネントの関数ライブラリの管理は，［アプリケーション領域］の関数ライブラリ表示枠で行います。詳細については，460 ページ「関数ライブラリの関連付け」を参照してください）。

ステップの引数の値の定義

［**値**］セルには，各操作の引数の値が一覧表示されます。各引数に定数値またはパラメータを挿入できます。パラメータを挿入する場合は，ローカル・パラメータまたはコンポーネント・パラメータを使用できます。詳細については，542 ページ「パラメータの使用」および 660 ページ「コンポーネントのパラメータの定義」を参照してください。

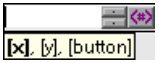
パスワードの値を暗号化することもできます。詳細については，536 ページ「で使用するパスワードを暗号化できます。メソッド引数への暗号化されたパスワードの挿入」を参照してください。

［**値**］セルは，選択した操作の引数の数に従って分割されています。次のように，パーティションに入力可能な引数のタイプに従って，各パーティションにはさまざまなオプションが含まれています。

引数のパーティション	引数のタイプ	説明
	String	引用符で囲んだ任意の英数字文字列を入力できます。引用符を入力しなかった場合は，QuickTest によって引用符が自動的に追加されます。引用符で囲まれた文字列が含まれているセルの値を変更して引用符を削除した場合，QuickTest は引用符を復元しません。また値は変数名として処理されます。
	Integer	任意の数値を入力できます。また上下の矢印を使用して数値を選択できます。
	ブール	リストから **True** または **False** の値を選択できます。
	定義済み定数	リストから，定義済みの値を選択できます。値のリストが提供されない場合は，このボックスに値を手作業で入力できます。

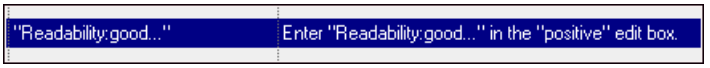
値を定義または変更するには，次の手順を実行します。

［**値**］セルの各区画をクリックして，選択した操作の引数値を入力します。
［**値**］セルをクリックすると，ツールチップに各引数の情報が表示されます。ツールチップには，現在強調表示されているパーティションの引数が太字で表示され，オプションの引数があれば大括弧で囲んで表示されます。



複数行の引数を追加するには，次の手順を実行します。

SHIFT+ENTER キーを押して，引数値に改行を追加します。複数行の引数値を入力すると，その引数値が自動的に文字列に変換され，引数の最初の行とそれに続く省略記号（...）のみ表示されます。キーワード・ビューの［注釈］カラムでも，複数行の引数値はこの形式で表示されます。



ヒント：ステップで使用される引数値をすべて表示するには，セルを選択します。引数値は，ステップ内に表示されるとおりに実行セッション中に使用されます。たとえば，引数値の一部として引用符を入力した場合，その引用符は実行セッション中に使用される引数値にも含まれます。QuickTest では，複数行の値は自動的に文字列と解釈されます。そのため，引用符を追加する必要はありません。

引数の値をパラメータ化するには，ローカルまたはコンポーネント・パラメータを使用して，次の手順を実行します。

パラメータ化が必要な［**値**］セル内で ボタンをクリックします。ローカル値のパラメータ化の詳細については，542 ページ「パラメータの使用」を参照してください。コンポーネント値のパラメータ化の詳細については，660 ページ「コンポーネントのパラメータの定義」を参照してください。

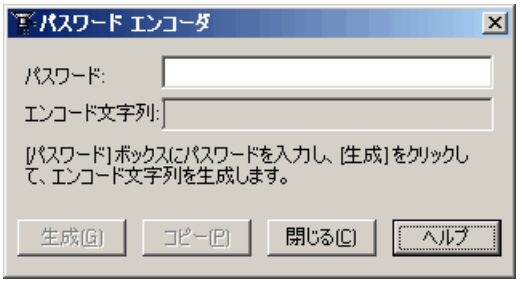
で使用するパスワードを暗号化できます。**メソッド引数への暗号化されたパス**

ワードの挿入

パスワードを暗号化し，その結果生成された文字列をメソッド引数として使用できます。たとえば，ユーザがパスワードを入力しなければならないフォームが Web サイトにあるとします。異なるパスワードにサイトがどのように応答するかをテストしたいが，同時にパスワードの安全性も確保したいとします。［**パスワード エンコーダ**］を使えばパスワードを暗号化できます。

パスワードを暗号化するには，次の手順を実行します。

1 **Windows** メニューの［**スタート**］＞［**プログラム**］＞［**QuickTest Professional**］＞［**Tools**］＞［**Password Encoder**］を選択します。［パスワード エンコーダ］ダイアログ・ボックスが開きます。



2 ［**パスワード**］ボックスにパスワードを入力します。

3 ［**生成**］をクリックします。［パスワード エンコーダ］によってパスワードが暗号化され，暗号化された値が［**エンコード文字列**］ボックスに表示されます。

4 ［**コピー**］ボタンを使用して，暗号化された値をコピーし，データ・テーブルに貼り付けます。

5 暗号化したいパスワードごとに，この手順を繰り返します。

6 ［**閉じる**］をクリックして，［パスワード エンコーダ］を閉じます。

ステップの出力値の定義

出力値の出力タイプおよび設定は，［**出力**］セルで定義します。この定義によって，出力値の保存場所と，コンポーネントの実行セッションにおいて出力値がどのように使用されるかが決まります。出力値のステップに到達すると，QuickTest によって，出力対象として選択された値が取得され，後の実行セッションで使用できるように指定された場所に保存されます。

新しい出力値ステップを作成すると，出力対象として選択された各値に標準設定の定義が割り当てられます。ビジネス・コンポーネントのステップの値を出力すると，次のようになります。

- コンポーネントで出力コンポーネント・パラメータが 1 つ以上定義されている場合，標準設定の出力タイプは「**コンポーネント パラメータ**」で，標準設定の出力名は［ビジネス コンポーネントの設定］ダイアログ・ボックスの［パラメータ］表示枠に表示されている最初の出力パラメータです。
- コンポーネントで出力コンポーネント・パラメータが定義されていない場合，標準設定の出力タイプは「**ローカル パラメータ**」で，標準設定の出力名は「**p_Local**」です。

出力パラメータは，必要に応じて変更します。ローカル・パラメータを選択した場合は，［出力オプション］ダイアログ・ボックスで直接変更できます。コンポーネント・パラメータを選択した場合，出力パラメータの詳細は読み取り専用となります。パラメータの詳細は，［ビジネス コンポーネントの設定］ダイアログ・ボックスの［パラメータ］表示枠で変更できます。詳細については，660 ページ「コンポーネントのパラメータの定義」を参照してください。

出力値を指定した後で出力値を保存しないことに決めた場合は，指定をキャンセルできます。詳細については，540 ページ「パラメータに対する出力値のキャンセル」を参照してください。

パラメータに出力値を設定するには，次の手順を実行します。

1 パラメータへの出力を作成または編集するには，［**出力**］セルをクリックします。［**出力**］ボタン をクリックするか，CTRL + F11 キーを押します。［**出力オプション**］ダイアログ・ボックスが開きます。

2 ［**出力タイプ**］ボックスで，「**コンポーネント パラメータ**」または「**ローカル パラメータ**」を選択します。［**詳細**］領域には，選択したコンポーネント・タイプで使用可能なオプションが表示されます。

注 :「**コンポーネント・パラメータ**」タイプを選択できるのは，そのコンポーネントについて出力コンポーネント・パラメータが定義されている場合だけです。コンポーネント・パラメータを選択すると，情報が読み取り専用で表示されます。詳細については，660 ページ「コンポーネントのパラメータの定義」を参照してください。

3［**名前**］ボックスから，必要なパラメータを選択します。ローカル・パラメータが 1 つも定義されていない場合は，標準設定のパラメータ名として **p_Local** が表示されます。

- ➤ 必要に応じて，新しいローカル・パラメータを作成できます。詳細については，542 ページ「パラメータの使用」を参照してください。
- ➤ ローカル・パラメータを選択した場合は，その詳細を指定します。詳細については，542 ページ「パラメータの使用」を参照してください。
- ➤ コンポーネント・パラメータを選択した場合，その詳細は読み取り専用で表示されます。

4［**OK**］をクリックします。［**出力**］セルに，出力値の保存先となるパラメータが表示されます。

ヒント：出力パラメータを指定した後でその［**出力**］セルをクリックすると，パラメータのタイプを示す次のようなアイコンがセル内に表示されます。 アイコンはコンポーネント・パラメータを示します。 アイコンは，ローカル・パラメータを示します。

パラメータに対する出力値のキャンセル

コンポーネント・ステップの出力値を保存しない場合は，出力をキャンセルできます。

パラメータに対する出力値をキャンセルするには，次の手順を実行します。

［**出力**］セルをクリックします。［**キャンセル**］ボタン をクリックするか，DELETE キーを押して，パラメータへの出力を取り消します。

コンポーネントへのその他のタイプのステップの追加

キーワード・ビューを使用してコンポーネントに標準ステップを追加できるほかに，［**挿入**］メニューから関連するオプションを使用して，次の特別なタイプのステップを挿入することもできます。各ステップはキーワード・ビューの行として入力されます。また，541 ページ「ステップの部分の変更」の説明に従って修正できます。

- チェックポイントを挿入できます。詳細については，563 ページ「チェックポイントについて」を参照してください。
- 出力値ステップを挿入できます。詳細については，593 ページ「値の出力」を参照してください。
- ステップにコメントを挿入して，コンポーネントの部分を区切り，特定の部分に関する詳細を追加することができます。

ステップの部分の変更

キーワード・ビューでは，ステップの任意の部分を変更できます。たとえば，ステップの実行対象であるテスト・オブジェクトを変更したり，ステップで実行される操作を変更したりすることができます。

キーワード・ビューで作業する際には，［**編集**］メニューまたはショートカット・メニューの標準的な編集コマンド（［**切り取り**］，［**コピー**］，［**貼り付け**］および［**削除**］）を使用して，ステップの変更を簡単に行うことができます。

ステップを変更するには，変更するステップの部分を含むセルをクリックし，セルの内容を指定します。ステップ行の各セルは，ステップの異なる部分を表します。詳細については，523 ページ「コンポーネントへのステップの追加」を参照してください。

パラメータの使用

ビジネス・コンポーネントに値を渡す入力パラメータと，コンポーネントから外部のソースに，またはステップから別のステップに値を渡す出力パラメータを定義できます。定義したパラメータを使用すれば，ステップの入力値と出力値をパラメータ化できます。

定義できるパラメータには，「**ローカル・パラメータ**」と「**コンポーネント・パラメータ**」の 2 種類があります。

ローカル・パラメータ：同じコンポーネント内での使用するために，コンポーネント内で定義される変数値。

ローカルの入力パラメータ値は，同じコンポーネント内の以降のパラメータ化されたステップで受け取って使用できます。ローカルの出力パラメータは，同じコンポーネント内での使用のために，操作またはコンポーネント・ステップによって返されます。ローカルのパラメータ出力値は，ビジネス・プロセスのテスト結果で確認できます。

ローカル・パラメータはビジネス・コンポーネント・キーワード・ビューで定義します。入力パラメータの定義には［設定値オプション］ダイアログ・ボックス，出力パラメータの定義には［出力オプション］ダイアログ・ボックスを使用します。ローカル・パラメータは削除できませんが，そのパラメータへの入力または出力をキャンセルすることはできます。

コンポーネント・パラメータ：ビジネス・プロセス・テスト内の同じコンポーネントまたは以降のコンポーネントで使用するために，コンポーネント内で定義される変数値。

コンポーネント入力パラメータ値は，コンポーネント内の特定のパラメータ化されたステップに対する値として受け取って使用できます。コンポーネント出力パラメータ値は，以降のテストで使用するコンポーネントの入力パラメータとして返されます。これらの値は，ビジネス・プロセス・テスト結果でも確認することができます。

コンポーネント・パラメータは，［ビジネス コンポーネントの設定］ダイアログ・ボックスの［パラメータ］表示枠，または Quality Center ビジネス・コンポーネント・モジュールで定義できます。

本項では，ローカル・パラメータを設定する方法と，ローカル・パラメータおよびコンポーネント・パラメータを使って入力値と出力値をパラメータ化する

方法を説明します。コンポーネント・パラメータの設定の詳細については，660 ページ「コンポーネントのパラメータの定義」を参照してください。

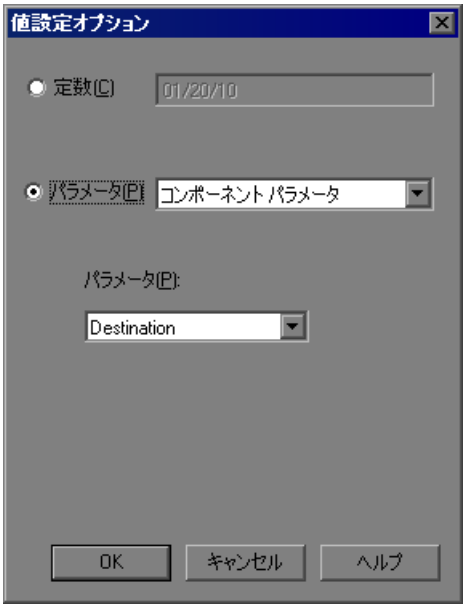
パラメータを定義したら，そのパラメータを使用して値をパラメータ化できます。また，定数値を［**値**］セルに直接入力することで，パラメータに定数値も適用できます。

入力値のパラメータ化

［**値**］セルでは，ローカル・パラメータまたはコンポーネント・パラメータを使って，ステップの入力値をパラメータ化できます。

ローカル・パラメータを使って入力値をパラメータ化するには，次の手順を実行します。

1 ［**値**］セルで，パラメータ化ボタン をクリックするか，CTRL + F11 キーを押します。［値設定オプション］ダイアログ・ボックスが開きます。



注：コンポーネントで入力コンポーネント・パラメータが 1 つ以上定義されている場合，標準設定の入力タイプは「**コンポーネント パラメータ**」で，標準設定の入力名は［ビジネス コンポーネントの設定］ダイアログ・ボックスの［パラメータ］表示枠に表示されている最初の入力パラメータです。

2［**パラメータ**］ボックスで，「**ローカル・パラメータ**」を選択します。ローカル・パラメータ・タイプの詳細が表示されます。

3 ローカル・パラメータのプロパティの詳細を指定します。

- ［**名前**］：パラメータに対するわかりやすい名前を入力するか（大文字と小文字が区別されます），リストから名前を選択します。
- ［**値**］：パラメータに対する入力値を入力します。値を指定しなかった場合は，次の表に示す標準設定の値が自動的に割り当てられます。

値のタイプ	QuickTest の標準設定値
String	空文字列
ブール	True
Date	現在の日付

値のタイプ	QuickTest の標準設定値
数値	0
パスワード	空文字列

➤［**記述**］: パラメータに関する簡単な説明を入力します。

4［**OK**］をクリックします。ステップの［**値**］セルにローカル・パラメータが表示されます。コンポーネントの実行では，ステップのパラメータに指定された値が使用されます。

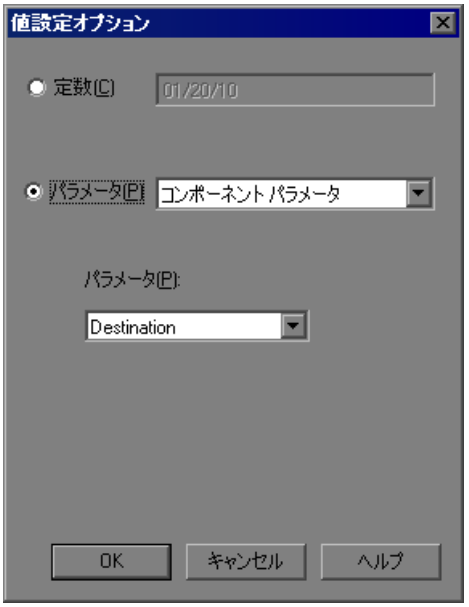
ヒント:

➤ 値のパラメータ化をキャンセルするには，［値設定オプション］ダイアログ・ボックスで［**定数**］オプションを選択して，定数値を入力します。［**定数**］ボックスには，［**値**］セルと同じ編集オプションがあり，パラメータ化ボタン をクリックすると，このダイアログ・ボックスが開きます。これらのオプションの詳細については，535 ページ「ステップの引数の値の定義」を参照してください。

➤ ローカル・パラメータの定義後に［**値**］セルをクリックすると，ローカル・パラメータが定義されたセルの各区画に アイコンが表示されます。

コンポーネント・パラメータを使って入力値をパラメータ化するには，次の手順を実行します。

1 ［**値**］セルで，パラメータ化ボタン をクリックするか，CTRL + F11 キーを押します。［値設定オプション］ダイアログ・ボックスが開きます。

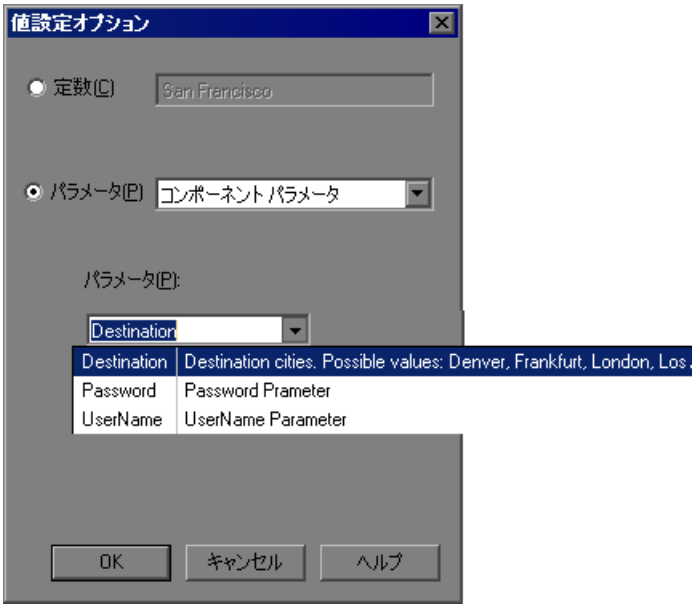


注：コンポーネントで入力コンポーネント・パラメータが 1 つ以上定義されている場合，標準設定の入力タイプは「**コンポーネント パラメータ**」で，標準設定の入力名は［ビジネス コンポーネントの設定］ダイアログ・ボックスの［パラメータ］表示枠に表示されている最初の入力パラメータです。コンポーネント・パラメータが定義されていない場合，コンポーネント・パラメータを使って入力値をパラメータ化するためには，コンポーネント・パラメータを定義する必要があります。詳細については，660 ページ「コンポーネントのパラメータの定義」を参照してください。

---

2 ［**パラメータ**］ボックスで，パラメータ化される値に使用するコンポーネント・パラメータを選択します。使用可能なコンポーネント・パラメータの名前と詳

細な説明が読み取り専用で表示されます。必要に応じて表示サイズを変更できます。また，パラメータのリストが長ければ，リストをスクロールできます。



3 [**OK**] をクリックします。ステップの [**値**] セルにコンポーネント・パラメータが表示されます。コンポーネントの実行では，ステップのパラメータに指定された値が使用されます。

ヒント：

- 値のパラメータ化をキャンセルするには，[値設定オプション] ダイアログ・ボックスで [**定数**] オプションを選択して，定数値を入力します。[**定数**] ボックスには，[**値**] セルと同じ編集オプションがあり，パラメータ化ボタン をクリックすると，このダイアログ・ボックスが開きます。これらのオプションの詳細については，535 ページ「ステップの引数の値の定義」を参照してください。
- コンポーネント・パラメータの定義後に [**値**] セルをクリックすると，コンポーネント・パラメータが定義されたセルの各区画に アイコンが表示されます。

出力値のパラメータ化

ステップの出力値は，ステップの［**出力**］セルで，ローカル・パラメータまたはコンポーネント・パラメータを使ってパラメータ化できます。出力パラメータ値は，コンポーネント内の後続のステップやビジネス・プロセス・テスト内の後続のコンポーネントで入力値として使用できます。

ローカル・パラメータを使って出力値をパラメータ化するには，次の手順を実行します。

1 ［**出力**］セルで，［出力パラメータの指定］ボタン をクリックするか，CTRL + F11 キーを押します。［出力オプション］ダイアログ・ボックスが開きます。

注：コンポーネントで出力コンポーネント・パラメータが 1 つ以上定義されている場合，標準設定の出力タイプは「**コンポーネント パラメータ**」で，標準設定の出力名は［ビジネス コンポーネントの設定］ダイアログ・ボックスの［パラメータ］表示枠に表示されている最初の出力パラメータです。

---

2 ［**出力タイプ**］ボックスで，［**Local Parameter**］を選択します。ローカル・パラメータ・タイプの詳細が表示されます。

3 ローカル・パラメータのプロパティの詳細を指定します。

- ［**名前**］: パラメータに対するわかりやすい名前を入力するか，リストから名前を選択します。
- ［**記述**］: パラメータに関する簡単な説明を入力します。

4 ［**OK**］をクリックします。ステップの［**出力**］セルにローカル・パラメータが表示されます。コンポーネントは実行時に，ステップに対して指定された出力パラメータに値を出力します。

ヒント：ローカル・パラメータの定義後に［**出力**］セルをクリックすると，ローカル・パラメータが定義されたセルの各区画に アイコンが表示されます。

コンポーネント・パラメータを使って出力値をパラメータ化するには，次の手順を実行します。

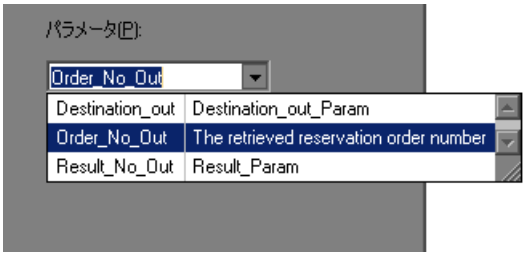
1 ［**出力**］セルで，［出力パラメータの指定］ボタン をクリックするか，CTRL + F11 キーを押します。［出力オプション］ダイアログ・ボックスが開きます。

注：コンポーネントで出力コンポーネント・パラメータが 1 つ以上定義されている場合，標準設定の出力タイプは「**コンポーネント パラメータ**」で，標準設定の出力名は［ビジネス コンポーネントの設定］ダイアログ・ボックスの［パラメータ］表示枠に表示されている最初の出力パラメータです。コンポーネント・パラメータが定義されていない場合，出力値をパラメータ化するためには，コンポーネント・パラメータを定義する必要があります。詳細については，660 ページ「コンポーネントのパラメータの定義」を参照してください。

---

2 ［**パラメータ**］ボックスで，出力値を格納するコンポーネント・パラメータを選択します。使用可能なコンポーネント・パラメータの名前と詳細な説明が読

み取り専用で表示されます。必要に応じて表示サイズを変更できます。また，パラメータのリストが長ければ，リストをスクロールできます。



3 ［**OK**］をクリックします。ステップの［**出力**］セルにコンポーネント・パラメータが表示されます。コンポーネントは実行時に，ステップに対して指定された出力パラメータに値を出力します。

ヒント：ローカル・パラメータの定義後に［**値**］セルをクリックすると，ローカル・パラメータが定義されたセルの各区画に アイコンが表示されます。

コメントの使用

コメントは，ビジネス・コンポーネントに自由に入力できるテキスト・エントリです。キーワード・ビューでは，コメントに アイコンが付いています。コメントにはいくつかの用途があります。たとえば，アプリケーションがテストできる状態になる前に，ビジネス・コンポーネントに含めるステップを計画しなければならない場合があります。その後，アプリケーションの準備が整ったときに，その計画を使用して，テストする必要のある項目がすべてコンポーネント・ステップに含まれているかどうかをチェックできます。

コンポーネントにコメントを追加して読みやすさを向上させることができ，更新が容易になります。たとえば，コンポーネントの各セクションの前に，そのセクションに何が含まれているかを説明するコメントを追加します。

追加したコメントは，1 つ以上のカラムが表示されてさえいれば，常にコンポーネント内に表示されます。表示するカラムを選択する方法については 555 ページ「キーワード・ビューの表示オプションの定義」を参照してください。また，グリッドを横方向にスクロールした場合も，コメントは常に表示されます。QuickTest がコンポーネントを実行するときに，コメントは処理されません。

注：挿入したコメントをステップに変更することはできません。

コンポーネントにコメントを追加するには，次の手順を実行します。

1 ［**挿入**］＞［**コメント**］を選択するか，［**項目**］セルをクリックして表示されたリストから［**コメント**］を選択するか，コンポーネント・ステップを右クリックして［**コメントの挿入**］を選択します。選択したステップの下にコメント行が追加されます。

2 コメント行にテキストを入力します。テキストを入力しなかった場合，そのコメントは，カーソル・フォーカスが別の場所に移動したときに自動的に削除されます。

既存のコメントを変更するには，次の手順を実行します。

［**コメント**］カラム内で，コメントをダブルクリックします。テキスト・ボックスが自由テキスト・フィールドになります。代わりに アイコンをクリックするという方法もあります。このアイコンは，コメントの状態を編集可能と読み取り専用の間で切り替えます。

コメントを削除するには，次の手順を実行します。

1 コメントを選択し，［**編集**］＞［**削除**］を選択するか，キーボードの DELETE キーを押します。または，コメントを右クリックしてショートカット・メニューから［**削除**］を選択します。

2 ［**コメントの削除**］をクリックして確定します。コメントがビジネス・コンポーネントから完全に削除されます。

コンポーネントステップの管理

コンポーネントのステップは，コンポーネント内の別のステップまたはコメントの前または後に移動できます。また，不要になったステップは削除できます。

コンポーネントステップの移動

必要に応じて，ステップをコンポーネント内の別の場所に移動できます。

コンポーネント内でステップを移動するには，次の手順を実行します。

- ［**項目**］カラムで，ステップを上または下にドラッグし，適切な位置でドロップします。選択したステップをドラッグすると，ステップの移動先を示す線が表示されます。
- ステップをクリップボードにコピーするか切り取って，目的の場所に貼り付けます。ステップをコピーするには［**編集**］＞［**コピー**］か CTRL + C キーを使用し，ステップを切り取るには［**編集**］＞［**切り取り**］か CTRL + X キーを使用します。ステップを貼り付けるには［**編集**］＞［**貼り付け**］か CTRL + V キーを使用します。

コンポーネントのステップの削除

コンポーネントのステップは，必要に応じて削除できます。ステップを削除する前に，その削除がコンポーネントの正常な動作を妨げる原因にならないことを確認してください。

注：いずれかのセルが編集モードの場合，ステップは削除できません。

ステップを削除するには，次の手順を実行します。

1. 削除するステップを選択し，［**編集**］＞［**削除**］を選択するか，DELETE キーを押します。または，ステップを右クリックしてショートカット・メニューから［**削除**］を選択します。警告メッセージが表示されます。
2. ［**ステップの削除**］をクリックして確定します。ステップがコンポーネントから削除されます。

キーワード・ビューでのキーボード・コマンドの使用

キーワード・ビューでの操作にキーボードを使用する場合は，次のキーボード・コマンドを使用できます。

- 現在選択されているステップの下に新しいステップを追加するには，F8 を押します。
- 条件ブロックまたはループ・ブロックの後に新しいステップを追加するには，SHIFT+F8 を押します。
- ステップ・ジェネレータを使用して選択されているステップの下に新しいステップを追加するには，F7 を押します。
- セルが編集モードになっていなければ，行でのフォーカスを左に移動するには SHIFT+TAB キー，右に移動するには TAB キーを使用します。この場合，ENTER を押すと編集モードが終了し，TAB キーを押すとまたフォーカスを移動できるようになります。
- リストを含んだセルが選択されている場合には，次のキーを使用できます。
  - セルのリストを開くには，SHIFT+F4 キーを押します。
  - 選択されている項目は，上矢印キーと下矢印キーで変更します。［**項目**］カラムでは，リストを開いてから矢印キーを使用します。
  - 値の最初の文字を 1 つまたはいくつか入力すると，それらの文字で始まる値に移動できます。入力した文字は白で強調表示されます。

キーワード・ビューの表示オプションの定義

［キーワード ビュー オプション］ダイアログ・ボックスを使用すると，キーワード・ビューでの情報の表示方法を選択できます。キーワード・ビューのカラム，フォント，色の表示をカスタマイズできます。設定したオプションは，それ以降，同じコンピュータ上で実行されるすべてのセッションで，すべてのテストに対して有効になります。

キーワード・ビューのカラムの表示

キーワード・ビューに表示するカラムを指定するには，［キーワード ビュー オプション］ダイアログ・ボックスの［カラム］タブを使用します。また，カラムの表示順も指定できます。

ヒント：カラム・ヘッダ行を右クリックして表示されるメニューから［**ドキュメントのみ**］を選択すると，［**注釈**］カラムのみ表示できます。その後でキーワード・ビューを印刷すれば，手動テストの指示書として使用できます。キーワード・ビューからの印刷の詳細については，505 ページ「コンポーネントの印刷」を参照してください。

表示するキーワード・ビューのカラムを指定するには，次の手順を実行します。

1 ［**ツール**］＞［**ビュー オプション**］を選択します。［キーワード ビュー オプション］ダイアログ・ボックスが表示されます。

［**利用可能なカラム**］リストには，現在キーワード・ビューに表示されていないカラムが表示されます。［**表示するカラム**］リストには，現在キーワード・ビューに表示されているカラムが表示されます。

2 ［**利用可能なカラム**］リストと［**表示するカラム**］リストの間でカラム名を移動するには，カラム名をダブルクリックするか，カラム名を選択して矢印ボタン（**>**および**<**）をクリックします。

ヒント：一方のリストから他方のリストにすべてのカラム名を移動するには，二重の矢印ボタン（**>>**および**<<**）をクリックします。選択した複数のカラム名のみを一方のリストから他方のリストに移動するには，（SHIFT キー /CTRL キーを使用して）複数のカラム名を選択して，矢印ボタン（**>** および **<**）をクリックします。

---

3 [**表示するカラム**] リストで 1 つ以上のカラムを選択し，上矢印ボタンと下矢印ボタンを使って，キーワード・ビューでのカラムの表示順を設定します。

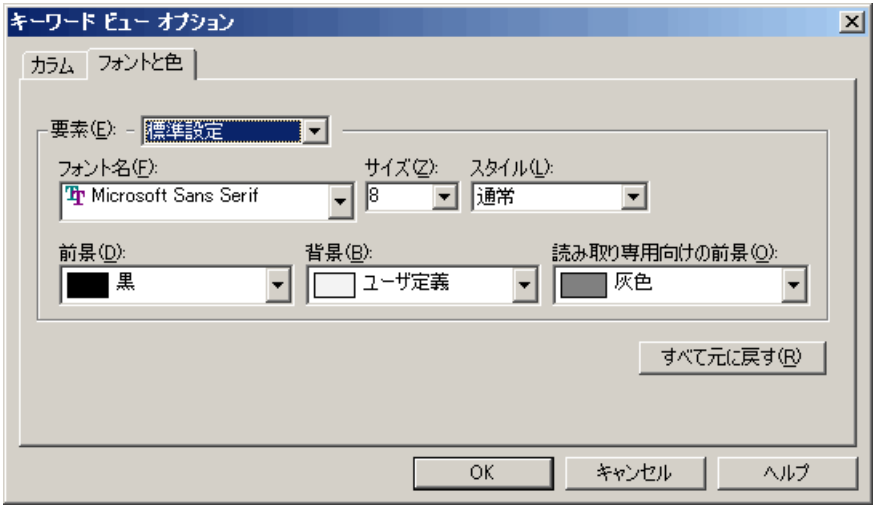
注： キーワード・ビューでのカラムの順序は，各ステップに関して完了しなければならないセルの順序には影響しません。たとえば，[**項目**] カラムの左側に [**操作**] カラムが表示されるように選択した場合も，まずその項目を選択する必要があります。その場合のみ，[**項目**] カラムで行った選択に一致するよう [**操作**] カラム・リストが更新されます。

---

4 [**OK**] をクリックしてダイアログ・ボックスを閉じ，新しいカラム表示を適用します。

キーワード・ビューのフォントと色の設定

キーワード・ビュー内の各種要素に対して異なるテキストや色の表示オプションを指定するには，[キーワード ビュー オプション] ダイアログ・ボックスの [フォントと色] タブを使用します。



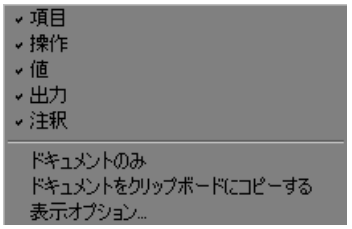
［フォントと色］タブには，次のオプションがあります。

オプション	説明
［要素］	キーワード・ビューの要素ごとに，フォントと色のオプションを個別に指定できます。現在の定義を確認および変更するには，次の要素のいずれかを選択します。 ➤［**交互の行**］: 1 行おきに使用する代替の背景色。この背景色が適用される行でも，フォントとテキストについては［**標準設定**］要素で定義したフォントとテキストの色が適用されます。 ➤［**コメント**］: コメント行の行とテキスト。ただし，使用可能なすべての書式設定オプションはコメント行全体に適用され，通常のステップ行内のコメントには適用されません。ステップ行内のコメントには，［**前景**］に指定した色だけが適用されます（その他すべての設定については，必要に応じて［**交互の行**］，［**標準設定**］，［**選択されている行**］の指定が適用されます）。 ➤［**標準設定**］: キーワード・ビューのすべての行とテキスト（次に示す要素を除く）。 ➤［**選択されている行**］: 現在選択（強調表示）されている行とテキスト。
［フォント名］	選択した要素のテキストに使用するフォントを変更できます。［**交互の行**］要素および［**選択されている行**］要素のフォントは変更できません。 **注**: Unicode 環境でテストを行う場合は，Unicode 対応のフォントを選択しなければなりません。選択しなかった場合，コンポーネント内の要素がキーワード・ビューで正しく表示されません。ただし，コンポーネントは，選択したフォントにかかわらず，同じように実行されます。
［サイズ］	選択した要素のテキストに使用するフォント・サイズを変更できます。［**交互の行**］要素および［**選択されている行**］要素のフォント・サイズは変更できません。

オプション	説明
［**スタイル**］	選択した要素のテキストに使用するフォント・スタイルを変更できます。［**通常**］，［**太字**］，［**斜体**］，または［**下線**］フォント・スタイルを選択できます。［**交互の行**］または［**選択されている行**］要素のフォント・スタイルは変更できません。
［**前景**］	選択した要素のテキストに使用する色を変更できます。［**交互の行**］の前景の色は変更できません。
［**背景**］	選択した要素の行の色を変更できます。
［**読み取り専用向けの前景**］	読み取り専用の行のテキストに使用する色を変更できます。このオプションは，［**交互の行**］が対象の場合には変更できません。
［**すべて元に戻す**］	［フォントと色］タブのすべてのオプションを標準設定にリセットします。

キーワード・ビューを使った作業のヒント

➤ 特定のカラムの表示 / 非表示を切り替えるには，キーワード・ビューで，カラム・ヘッダ行を右クリックして表示されるメニューから必要なカラムを選択または選択解除します。



たとえば，使用するステップを手動テスト用の指示書として印刷する場合などに，［**ドキュメントのみ**］を選択すると［**注釈**］カラムのみ表示できます。［**注釈**］カラムとコンポーネント内で定義されたコメントが表示されます。

➤ キーワード・ビューでカラム・ヘッダーを新しい場所にドラッグすることで，カラムを再整列できます。カラム・ヘッダを使用可能な場所にドラッグすると，赤い矢印が表示されます。

キーワード・ビューのブレークポイントを使った作業

キーワード・ビューではブレークポイントの挿入と削除を行うことができます。

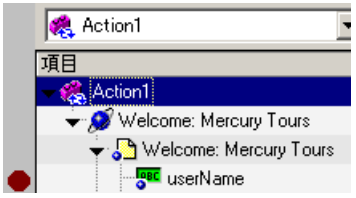
キーワード・ビューにブレークポイントを挿入するには，次の手順を実行します。

- ブレークポイントを挿入する場所で左マージンをクリックします。
- ステップを選択して F9 キーを押します。
- ［**デバッグ**］＞［**ブレークポイントの設定 / 解除**］を選択します。

赤いブレークポイントのアイコンが表示されます。

キーワード・ビューからブレークポイントを削除するには，次の手順を実行します。

- ブレークポイント・アイコンをクリックします。
- ステップを選択して F9 キーを押します。
- ［**デバッグ**］＞［**ブレークポイントの設定 / 解除**］を選択します。



ブレークポイントの詳細については，748 ページ「ブレークポイントの使用」を参照してください。

第 18 章

チェックポイントについて

アプリケーションのオブジェクトを検査して，それらが正しく機能することを確認できます。

本章の内容

- チェックポイントについて（563 ページ）
- コンポーネントへの新しいチェックポイントの追加（564 ページ）
- チェックポイントの種類について（565 ページ）

チェックポイントについて

QuickTest では，コンポーネントに検査を追加できます。「**チェックポイント**」は，指定したプロパティの現在値とそれらのプロパティの期待値を比較する確認ポイントです。チェックポイントにより，アプリケーションが正しく機能しているかどうかを確認できます。たとえば，標準チェックポイントを実行して，実際のオブジェクト・プロパティ値が期待値と一致しているか確認でき，ビットマップ・チェックポイントを実行して，アプリケーションの可視部分が正しく表示されているか確認できます。

チェックポイントを追加すると，QuickTest により，チェックポイントがキーワード・ビューの現在の行に追加され。標準設定では，チェックポイント名にはチェックポイントが作成されたテスト・オブジェクトの名前が使用されます。チェックポイントに別の名前を指定するか，標準設定の名前をそのまま使用するかを選択できます。

コンポーネントを実行すると，QuickTest によって，チェックポイントの期待結果と現在の結果が比較されます。この 2 つの結果が一致しないと，そのチェックポイントは失敗となります。チェックポイントの結果は［テスト結果］ウィンドウで見ることができます。

シンプル・モードと詳細モード

QuickTest には，関連する［チェックポイントのプロパティ］ダイアログ・ボックスでチェックポイントを挿入できる 2 つのモード，［**シンプル モード**］と［**詳細モード**］が用意されています。シンプル・モードでは，基本セットのプロパティと値をチェックできます。このタイプのチェックポイントは QuickTest でも Quality Center でも表示と編集が可能であるため，オートメーション・エンジニアが QuickTest でチェックポイントを定義または編集する作業と，Subject Matter Expert が Quality Center でチェックポイントを表示または編集する作業が可能になります。［チェックポイントのプロパティ］ダイアログ・ボックスの詳細については，570 ページ「[チェックポイントのプロパティ］ダイアログ・ボックスについて」を参照してください。

QuickTest では，Quality Center のユーザには見えない詳細なチェックポイントのプロパティも表示または編集できます。そのためには，詳細モードでチェックポイントを挿入します。チェックポイントに詳細なプロパティが含まれている場合に，Quality Center ユーザが Quality Center でそのプロパティを表示すると，検査されたが表示されていないプロパティがあることを示す警告が表示されます。

コンポーネントへの新しいチェックポイントの追加

チェックポイントは，コンポーネントの作成または編集中にも追加できます。通常は，最初のテストコンポーネントの作成後にチェックポイントを定義する方が簡単です。

コンポーネントの記録中に新しいチェックポイントを追加するには，次の手順を実行します。

［**挿入**］＞［**チェックポイント**］メニューでコマンドを使用するか，ツールバーの［**チェックポイントまたは出力値の挿入**］ボタンをクリックします。これにより，選択されているステップに関連するチェックポイントのオプションのメニューが表示されます。

チェックポイントの種類について

次に示す種類のチェックポイントを挿入して，アプリケーション内のさまざまなオブジェクトを検査できます。

- **標準チェックポイント**は，アプリケーション内のオブジェクトのプロパティの値を検査します。標準チェックポイントでボタン，ラジオ・ボタン，コンボ・ボックス，リストなど，さまざまなオブジェクトを検査できます。たとえば，ラジオ・ボタンを選択したときにアクティブになったかどうかを確認できます。また，エディット・ボックスの値を検査できます。

  標準チェックポイントはすべてのアドイン環境でサポートされています。

  標準チェックポイントの詳細については，第 19 章「標準チェックポイントを使ってオブジェクト・プロパティ値の検査」を参照してください。

- **ビットマップ・チェックポイント**では，アプリケーションの領域をビットマップとして検査します。たとえば，ユーザが指定した都市の地図を表示できる Web サイトがあるとします。この地図には，倍率を変更するためのコントロール・キーがあります。地図を拡大するコントロール・キーを 1 回クリックした後に表示される新しい地図を記録できます。ビットマップ・チェックポイントを使用すると，地図が適切に拡大されたかどうかを検査できます。

  ボタン，テキスト・ボックス，テーブルなど，アプリケーション内の任意の領域に対してビットマップ・チェックポイントを作成できます。

  ビットマップ・チェックポイントはすべてのアドイン環境でサポートされています。

  ビットマップ・チェックポイントの詳細については，第 20 章「ビットマップの検査」を参照してください。

第 19 章

標準チェックポイントを使ってオブジェクト・プロパティ値の検査

標準チェックポイントをコンポーネントに追加することにより，アプリケーションのオブジェクトのプロパティ値を期待値と比較できます。

本章の内容

- オブジェクトのプロパティ値の検査について（567 ページ）
- 標準チェックポイントの作成（568 ページ）
- ［チェックポイントのプロパティ］ダイアログ・ボックスについて（570 ページ）
- チェックポイントの変更（577 ページ）

オブジェクトのプロパティ値の検査について

標準チェックポイントを使用すると，アプリケーションのオブジェクトのプロパティ値を検査できます。標準チェックポイントは，オブジェクト・プロパティの期待値と，実行セッション中のオブジェクトの現在値を比較します。サポートされているすべてのテスト環境で標準チェックポイントを作成できます(適切なアドインがロードされている場合にかぎります)。

注：Quality Center でコンポーネントを表示しているときに，ユーザはコンポーネントのチェックポイントを作成，編集，または名前変更できません。

標準チェックポイントの作成

アプリケーション内の特定のオブジェクトが期待どおりのプロパティ値を持っているかどうかを検査するのには，コンポーネントを記録または編集しているときに，コンポーネントに標準チェックポイントを追加します。標準チェックポイントのオプションを設定するには，［チェックポイントのプロパティ］ダイアログ・ボックスを使用します。

注：キーワード・コンポーネントでは，画像，テーブル，または（Web）ページのチェックポイントは作成できません。これらの特殊な種類のチェックポイントは，テストでのみ利用できます。ただし，コンポーネントに対して標準チェックポイントを作成するときに，Web ページまたはテーブル・オブジェクトを選択した場合は，ほかのオブジェクトと同様に，それらのオブジェクトのプロパティを検査できます。

記録中に標準チェックポイントを追加するには，次の手順を実行します。

1 記録セッション中に，［**挿入**］＞［**チェックポイント**］＞［**標準チェックポイント**］を選択するか，［**チェックポイントまたは出力値の挿入**］ツールバー・ボタンをクリックします。

QuickTest ウィンドウが非表示になり，ポインタが指差しマークに変わります。

ヒント：

- 選択するウィンドウが別のウィンドウで部分的に隠れている場合は，部分的に隠れているウィンドウが前面に表示されるまで，その上に指差しマークを数秒間置いたままにします。次に，必要なオブジェクトをポイントしてクリックします。
- 左側の CTRL キーを押したままにすると，指差しマークが標準ポインタに変わり，ウィンドウのフォーカスを変更したり，オブジェクト上で右クリックやポインタの移動などの操作を実行してショートカット・メニューを表示したりできます。
- 選択するオブジェクトがあるウィンドウが最小化されている場合は，左の CTRL キーを押したまま，Windows タスク・バーでアプリケーションを右クリックして，ショートカット・メニューから［**元のサイズに戻す**］を選択すると，そのウィンドウを表示できます。

2 検査対象オブジェクトをクリックします。［オブジェクトの選択－チェックポイントのプロパティ］ダイアログ・ボックスが開きます。

3 表示されたオブジェクト・ツリーから，検査対象項目を選択します。ツリーに表示される項目の名前は，オブジェクトのクラスに対応しています。

4 ［**OK**］をクリックします。［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。

5 チェックポイントの設定を行います。詳細については，570 ページ「[チェックポイントのプロパティ］ダイアログ・ボックスについて」を参照してください。

6 ［**OK**］をクリックして，ダイアログ・ボックスを閉じます。キーワード・ビューで選択したオブジェクトにチェックポイント・ステートメントが追加されます。

編集中に標準チェックポイントを追加するには，次の手順を実行します。

1 チェックポイントを追加するステップを選択し，［**挿入**］＞［**チェックポイント**］＞［**標準チェックポイント**］を選択します。

［チェックポイントのプロパティ］ダイアログ・ボックスが開きます。

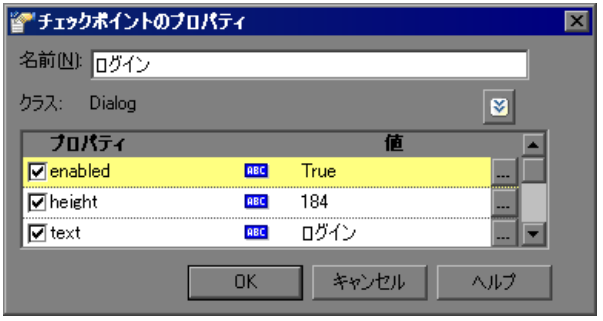
注：編集中に標準チェックポイントを追加するには，チェックポイントを作成したいオブジェクトがアプリケーションに表示されている必要があります。

2 チェックポイントの設定を指定します。詳細については，570 ページ「[チェックポイントのプロパティ］ダイアログ・ボックスについて」を参照してください。

3 ［**OK**］をクリックして，ダイアログ・ボックスを閉じます。キーワード・ビューで選択したオブジェクトにチェックポイント・ステートメントが追加されます。

［チェックポイントのプロパティ］ダイアログ・ボックスについて

［チェックポイントのプロパティ］ダイアログ・ボックスでは，検査対象のオブジェクトのプロパティを指定し，それらのプロパティの値を編集できます。［チェックポイントのプロパティ］ダイアログ・ボックスには，**シンプル・モード**と**詳細モード**という 2 つのモードがあります。どちらのモードが開くかは，詳細なプロパティがチェックポイントで選択されているかどうかによって変わります。



詳細については，次を参照してください。

➤ 571 ページ「[チェックポイントのプロパティ］ダイアログ・ボックスーシンプル・モードについて」

➤ 574 ページ「オブジェクト・プロパティの期待値の編集」

➤ 575 ページ「[チェックポイントのプロパティ] ダイアログ・ボックス－詳細モードについて」

[チェックポイントのプロパティ] ダイアログ・ボックス－シンプル・モードについて

新しいチェックポイントに対しては，[チェックポイントのプロパティ] ダイアログ・ボックスは，常にシンプル・モードで開きます。既存のチェックポイントのダイアログ・ボックスを開き，詳細なチェックポイント・プロパティが選択されていない場合も，[チェックポイントのプロパティ] ダイアログ・ボックスはシンプル・モードで開きます。シンプル・モードでは，オブジェクトの基本プロパティと期待値だけが表示されます。詳細なチェックポイントのプロパティを表示または編集するには，[**詳細モード**] ボタンをクリックします。詳細モードでは，このダイアログ・ボックスに追加のオプションも表示されます。詳細については，575 ページ「[チェックポイントのプロパティ] ダイアログ・ボックス－詳細モードについて」を参照してください。

注：チェックポイントの詳細なプロパティは，Quality Center では表示または編集することはできません。このため，1 つ以上の詳細なプロパティがチェックポイントで選択されて，Quality Center ユーザが Quality Center でそのプロパティを表示すると，ダイアログ・ボックスに，検査のために選択したプロパティの一部が表示されないことを示すテキストが表示されます。

チェックポイントの識別

ダイアログ・ボックスの最上部に，チェックポイントに関する情報が表示されます。

情報	説明
［名前］	チェックポイントの名前。標準設定では，チェックポイントの名前は，チェックポイントが作成されたテスト・オブジェクトの名前と同じです。別のチェックポイント名を指定するか，標準の名前をそのまま使用します。 チェックポイントの名前を変更する場合は，名前について以下のことを確認します。 ➤ 一意である ➤ 先頭および最後がスペースでない ➤ "（二重引用符）が含まれていない ➤ 次の文字の組み合わせが含まれていない：:=@@
［クラス］	オブジェクトのタイプ。キーワード・コンポーネントに対しては，オブジェクトのクラスは Quality Center に表示されません。

検査するオブジェクト・プロパティの選択

オブジェクトの標準プロパティが，ダイアログ・ボックスのプロパティ表示枠に表示されます。この表示枠には，プロパティとその値およびタイプが含まれています。

表示枠内の要素	説明
チェック・ボックス	それぞれのオブジェクト・クラスについて，QuickTest では標準のプロパティ検査が用意されています。標準の検査をそのまま受け入れることも，必要に応じて変更することもできます。 プロパティを検査するには，対応するチェック・ボックスを選択します。 プロパティを検査から除外するには，対応するチェック・ボックスをクリアします。
［タイプ］	アイコンは，プロパティの値が現在は定数であることを示します。 アイコンは，プロパティの値が現在はパラメータであることを示します。
プロパティ	プロパティの名前。
［値］	プロパティの期待値。プロパティ値の変更の詳細については，574 ページ「オブジェクト・プロパティの期待値の編集」を参照してください。

コンポーネントへのチェックポイントの挿入

［**ステートメントの挿入**］オプションでは，コンポーネントにおいて，いつチェックポイントを実行するかを指定します。

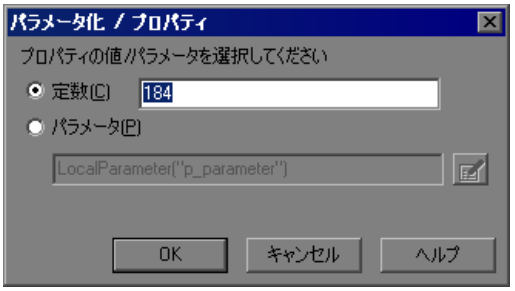
➤ 選択して強調表示したステップの実行前にオブジェクト・プロパティの値を検査するには，［**現在のステップの前**］を選択します。

➤ 選択して強調表示したステップの実行後にプロパティの値を検査するには，［**現在のステップの後**］を選択します。

注 :［**ステートメントの挿入**］オプションは，記録中にチェックポイントを追加するとき，あるいは既存のオブジェクト・チェックポイントを変更するときには使用できません。編集中に，既存のコンポーネントに新しいチェックポイントを追加するときに使用できます。

オブジェクト・プロパティの期待値の編集

...［チェックポイントのプロパティ］ダイアログ・ボックスでプロパティの［参照］ボタンをクリックすると，［パラメータ化 / プロパティ］ダイアログ・ボックスが開きます。このダイアログ・ボックスで，プロパティ値を［**定数**］または［**パラメータ**］として設定できます。標準設定は［**定数**］です。



- ［**定数**］: ステップ内で直接定義され，コンポーネントの実行中は変更されない値。

  ［**定数**］を選択すると，［**定数**］ボックスで値を直接編集できます。

- ［**パラメータ**］: ステップとは別に定義または生成され，特定のステップの実行時に取得される値。

すでにパラメータ化されている値に対して［**パラメータ**］を選択した場合，［**パラメータ**］ボックスには，その値の現在のパラメータの定義が表示されます。パラメータ化されていない値に対して［**パラメータ**］を選択した場合，［**パラメータ オプション**］ボタンをクリックして［パラメータ オプション］ダイアログ・ボックスを開くことができます。

パラメータのプロパティ詳細を指定するには，別のパラメータ・タイプを選択するか，またはパラメータ値の設定を変更します。コンポーネントでのパラメータの使用方法の詳細については，542 ページ「パラメータの使用」を参照してください。

［チェックポイントのプロパティ］ダイアログ・ボックス－詳細モードについて

詳細なチェックポイントのプロパティが選択されている場合は，［チェックポイントのプロパティ］ダイアログ・ボックスは詳細モードで開きます。このダイアログ・ボックスには，オブジェクトのサポートされているプロパティと期待値がすべて表示されます。詳細なチェックポイント・プロパティが選択されていない場合は，［**シンプル モード**］ボタンをクリックして，シンプル・モードのプロパティとオプションだけを表示できます。

詳細モードでは，［チェックポイントのプロパティ］ダイアログ・ボックスに［シンプル モード］オプション（571 ページを参照）と以下の追加オプションが表示されます

➤［**リポジトリ内を検索**］リポジトリのチェックポイントを表示するには，［**名前**］ボックスの右にある［**リポジトリ内を検索**］ボタンをクリックします。

（このオプションは，新しいチェックポイントの作成時には使用できません。既存のチェックポイントを編集するときのみ利用できます。

➤［**値の設定**］領域で，検査対象のプロパティの期待値を［**定数**］または［**パラメータ**］として定義できます。この領域を使用するのは，シンプル・モードで［参照］ボタンから開かれる［パラメータ化 / プロパティ］ダイアログ・ボックスを使うときです。詳細については，574 ページ「オブジェクト・プロパティの期待値の編集」を参照してください。

➤［**チェックポイントのタイムアウト**］: QuickTest が正常にチェックポイントを実行する時間間隔（秒単位）を指定します。チェックポイントの実行は，チェックポイントが成功するかタイムアウトが生じるまで試み続けられます。タイムアウトが生じる前にチェックポイントが成功しなければ，チェックポイントは失敗となります。

たとえば，オブジェクトが特定の状態になるまでいくらか時間を要するとします。この場合にチェックポイントのタイムアウト値を大きくしておけば，オブジェクトがその特定の状態になるまで十分な時間を確保でき，タイムアウトに達するまでに（データが一致すれば）チェックポイントが成功することが可能になります。

チェックポイントのタイムアウトを **0** 以外に指定して，チェックポイントに失敗した場合，［テスト結果］ウィンドウにチェックポイントのタイムアウトに関する情報が表示されます。

チェックポイントの変更

既存のチェックポイントの設定は変更できます。

チェックポイントを変更するには，次の手順を実行します。

1 選択チェックポイントが含まれているステップを選択して，［**編集**］＞［**ステップのプロパティ**］＞［**チェックポイントのプロパティ**］を選択します。関連するチェックポイントのダイアログ・ボックスが開きます。

2 プロパティを変更し，［**OK**］をクリックします。詳細については，570 ページ「[チェックポイントのプロパティ］ダイアログ・ボックスについて」を参照してください。

第 20 章

ビットマップの検査

QuickTest では，アプリケーションのオブジェクトのビットマップを，以前キャプチャしてコンポーネントと一緒に保管されているビットマップと比較することで，アプリケーションの可視部分が正しく表示されているか確認できます。

本章の内容

- ビットマップの検査について（579 ページ）
- ビットマップ比較の微調整（580 ページ）
- ビットマップ・チェックポイントの作成と変更（582 ページ）
- ビットマップ・チェックポイントのプロパティ・ダイアログ・ボックス（585 ページ）

ビットマップの検査について

アプリケーションの領域をビットマップとして検査できます。オブジェクト全体を検査することも，オブジェクト内の一部の領域を検査することも可能です。たとえば，ユーザが指定した都市の地図を表示できる Web サイトがあるとします。この地図には，倍率を変更するためのコントロール・キーがあります。地図を拡大するコントロール・キーを 1 回クリックした後に表示される新しい地図を記録できます。ビットマップ・チェックポイントを使用すると，地図が適切に拡大されたかどうかを検査できます。

サポートされているすべてのテスト環境でビットマップ・チェックポイントを作成できます（適切なアドインがロードされている場合にかぎります)。

ビットマップ・チェックポイントの結果が，オペレーティング・システム，画面解像度，色設定などの要素に影響を及ぼす場合があります。

ビットマップ・チェックポイントを作成すると，QuickTest によって，指定したオブジェクトの**可視**部分がビットマップとしてキャプチャされ（QuickTest では，画面に表示されていない部分や別のオブジェクトに隠れた部分などはキャプチャされません），チェックポイントがコンポーネントに挿入されます。

コンポーネントを実行すると，QuickTest によって，アプリケーションに含まれている実際のオブジェクトのビットマップがキャプチャされ，そのビットマップ（または，その中の選択領域）とチェックポイントに保管されているビットマップが比較されます。

違いがある場合は，実際のオブジェクトのビットマップが保存され，［テスト結果］ウィンドウの詳細表示枠に期待されるビットマップの隣に表示されます。［テスト結果］ウィンドウでは，不一致の内容を確認できるように，2 つのビットマップの違いを反映するビットマップも表示できます。チェックポイントが成功しても，ビットマップをテスト結果に保存しないように，または保存するように QuickTest を設定できます（［ツール］＞［オプション］＞［実行］＞［画面キャプチャ］表示枠）。チェックポイントのテスト結果の詳細については，727 ページ「チェックポイントの結果の表示」を参照してください。

ビットマップ比較の微調整

ビットマップ・チェックポイントを実行すると，アプリケーションで検査する領域とチェックポイントに保管されているビットマップがピクセルごとに比較されます。標準設定では，いずれかのピクセルが異なっていれば，チェックポイントが失敗します。［ビットマップ チェックポイントのプロパティ］ダイアログ・ボックス（585 ページを参照）には，ビットマップ比較を微調整するオプションが用意されています。

ビットマップが同一でなくても，チェックポイントが成功するように比較を調整するには，［**RGB の許容範囲**］および［**ピクセルの許容範囲**］オプションを下記の説明に従って設定します。

また，QuickTest では，ビットマップ・チェックポイントに**カスタム・コンペアラ**を使用できます。カスタム・コンペアラは，より明確なアルゴリズムに従ってチェックポイントでビットマップ比較を実行するために，ユーザまたはサードパーティが開発した COM オブジェクトです。QuickTest コンピュータに 1 つ以上のカスタム・コンペアラがインストールされ，登録されている場合，［ビットマップ 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［**コンペアラ**］オプションが表示されます。

このオプションでは，テスト要件に従ってビットマップ比較を実行する QuickTest 標準コンペアラまたはカスタム・コンペアラを選択できます。カスタム・コンペアラの作成に役立つ例については，1005 ページ「使用事例シナリオ：アプリケーション内で位置が変化する画像の処理」を参照してください。カスタム・コンペアラの開発の詳細については，付録 C「ビットマップ・チェックポイントのカスタマイズ」を参照してください。

カスタム・コンペアラを選択すると，［ビットマップ チェックポイントのプロパティ］ダイアログ・ボックスにあるオプションの一部が変わります。詳細については，585 ページ「ビットマップ・チェックポイントのプロパティ・ダイアログ・ボックス」を参照してください。

［ビットマップ チェックポイントの許容範囲］オプション

➤ ［**RGB の許容範囲**］: RGB（赤，緑，青）の許容範囲によって，実際のビットマップにあるピクセルの RGB 値が期待されるビットマップとの間で許容され，チェックポイントが成功する割合が決まります （RGB の許容範囲オプションは，24 ビットのカラー・デプスを持つビットマップに限られます）。

たとえば，同じビットマップのビットマップ・チェックポイントでも，チェックポイントを作成するときやテストを実行するときに異なるディスプレイ・ドライバを使用すると，失敗する可能性があります。あるディスプレイ・ドライバでは白色を RGB (255, 255, 255) と表示し，別のドライバでは白色を RGB (231, 231, 231) と表示するとします。これら 2 つの値の違いは約 9.4% です。**RGB の許容範囲** を 10% に設定することにより，これらのドライバのいずれかでテストを実行すると，チェックポイントが成功します。

注： QuickTest によって，実際のビットマップと期待されるビットマップの各ピクセルを比較するときに，RGB の許容範囲の設定が適用されます。各ピクセルの Red，Green，および Blue 値は別々に比較されます。いずれかの値が許容範囲より異なっていると，ピクセルの比較が失敗します。

➤［**ピクセルの許容範囲**］：ピクセルの許容範囲によって，期待のビットマップと異なることが許容され，チェックポイントが成功する実際のビットマップのピクセル数またはピクセルの割合が決まります。

たとえば，期待のビットマップに 4000 ピクセルあるとします。ピクセルの許容範囲を 50 に定義して，［**ピクセル**］ラジオ・ボタンを選択すると，実際のビットマップは最大 50 ピクセルまでなら期待のビットマップと異なることが許容され，チェックポイントが成功します。ピクセルの許容範囲を 5 に定義して，［**パーセント**］ラジオ・ボタンを選択すると，実際のビットマップは最大 200 ピクセルまでなら期待のビットマップと異なることが許容され，チェックポイントが成功します。

RGB およびピクセルの許容範囲の使い方

RGB とピクセルの許容範囲を定義すると，RGB の許容範囲が最初に計算されます。次に，ピクセルの許容範囲によって，RGB の基準で失敗しても，チェックポイントに成功するピクセルの最大数が定義されます。

たとえば，4000 ピクセルあるビットマップに，RGB の許容範囲として 10 パーセント，ピクセルの許容範囲として 5 パーセントを定義するとします。

チェックポイントが成功するには，実際のビットマップにある各ピクセルの RGB 値が，期待のビットマップの RGB 値の ± 10 パーセント以内でなければなりません。この基準に失敗すると，QuickTest によって，失敗したピクセル数が 200 未満であるかチェックされます。この基準に成功すると，チェックポイントが成功します。

ビットマップ・チェックポイントの作成と変更

コンポーネントの記録中に，ビットマップ・チェックポイントを挿入します。既存のビットマップ・チェックポイントを変更することもできます。

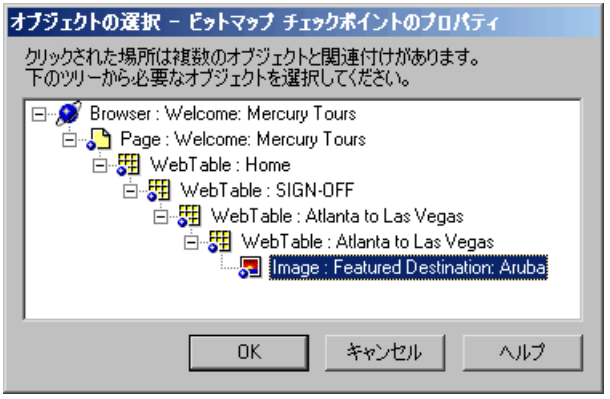
ビットマップ・チェックポイントはオブジェクトの可視部分のみをキャプチャできます。そのため，ビットマップ・チェックポイント・ステップを実行する前に，キャプチャするオブジェクトが画面上で完全に見えているか確認します。その 1 つの方法は，ビットマップ・チェックポイント・ステップの前に（関連環境用の）MakeVisible ステートメントを挿入することです。MakeVisible メソッドの詳細については，『**QuickTestObject Model Reference**』（英語版）を参照してください。

記録中にビットマップ・チェックポイントを作成するには，次の手順を実行します。

1 ［**挿入**］＞［**チェックポイント**］＞［**ビットマップ チェックポイント**］を選択するか，［**チェックポイントまたは出力値の挿入**］ボタンをクリックして，［**ビットマップ チェックポイント**］を選択します。

QuickTest ウィンドウが非表示になり，ポインタが指差しマークに変わります。指差しマーク機能の使用方法の詳細については，584 ページ「指差しマークの使用法に関するヒント」を参照してください。

2 アプリケーション内で検査対称オブジェクトをクリックします。クリックした場所がアプリケーション内の複数のオブジェクトに対応する場合は，［オブジェクトの選択－ビットマップ チェックポイントのプロパティ］ダイアログ・ボックスが開きます。



3 ツリーで，ビットマップ・チェックポイントを作成するオブジェクトを選択します。

ヒント：複数のオブジェクトを含む単独のビットマップ・チェックポイントを作成するには，ビットマップ・チェックポイントに含めるオブジェクトがすべて含まれている最上位のオブジェクトを選択する必要があります。

---

4 [**OK**] をクリックします。[ビットマップ チェックポイントのプロパティ] ダイアログ・ボックスがシンプル・モードで開きます。このダイアログ・ボックスのオプションを使って，ビットマップ・チェックポイントを作成します。詳細については，585 ページ「ビットマップ・チェックポイントのプロパティ・ダイアログ・ボックス」を参照してください。

ビットマップ・チェックポイントを変更するには，次の手順を実行します。

1 チェックポイントが含まれているステップを選択し，[**編集**] > [**ステップのプロパティ**] > [**チェックポイントのプロパティ**] を選択するか，そのステップで [**値**] セルを選択して，[**チェックポイントのプロパティ**] ボタンをクリックします。

2 [ビットマップ チェックポイントのプロパティ] ダイアログ・ボックスがシンプル・モードで開き，チェックポイントに保存したオブジェクトまたは領域が表示されます。このダイアログ・ボックスのオプションを使って，ビットマップ・チェックポイントを変更します。詳細については，585 ページ「ビットマップ・チェックポイントのプロパティ・ダイアログ・ボックス」を参照してください。

指差しマークの使用法に関するヒント

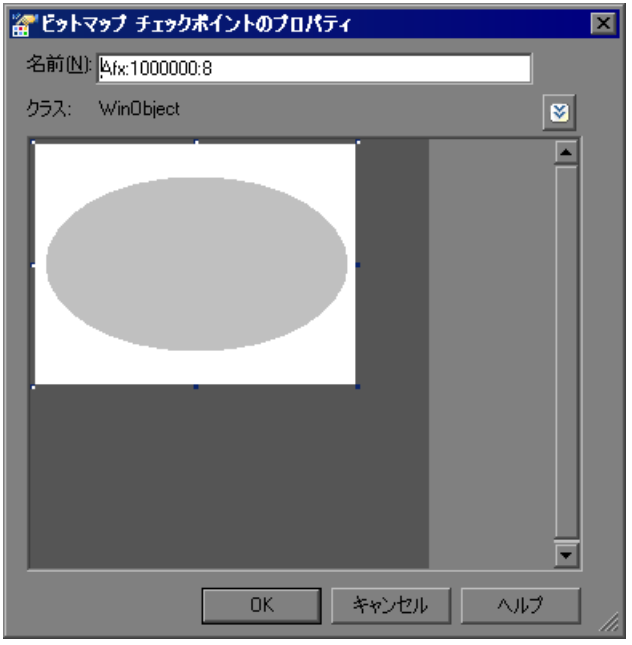
- 左の CTRL キーを押したままにすると，指差しマークが標準ポインタに変わります。そのときに，QuickTest またはアプリケーションでウィンドウのフォーカスを変更したり，操作（右クリックしたり，スクロール・バーを使ったり，ポインタをオブジェクト上に移動してショートカット・メニューを表示するなど）を実行できます。
- 選択するオブジェクトがあるウィンドウが別のウィンドウで部分的に隠れている場合は，部分的に隠れているウィンドウが前面に表示されるまで，その上に指差しマークを数秒間置いたままにします。次に，必要なオブジェクトをポイントしてクリックします。ウィンドウを前面に出すのに必要な時間の設定は，[オプション] ダイアログ・ボックスの [一般] 表示枠で行えます。
- 選択するオブジェクトがあるウィンドウが別のウィンドウで完全に隠れている場合，またはダイアログ・ボックスがウィンドウの背後に隠れている場合は，必要に応じて左の CTRL キーを押しながらウィンドウを配置します。
- 選択するオブジェクトがあるウィンドウが最小化されている場合は，左の CTRL キーを押したまま，Windows タスク・バーでアプリケーションを右クリックして，ショートカット・メニューから [**元のサイズに戻す**] を選択すると，そのウィンドウを表示できます。

➤ 特定のイベント（ショートカット・メニューを表示するために右クリックしたり，オブジェクト上にポインタを移動するなど）を実行するだけで，選択するオブジェクトを表示できる場合は，左の CTRL キーを押したままにします。指差しマークが一時的に標準ポインタに変わり，イベントを実行できます。選択するオブジェクトが表示されたら，左の CTRL キーを放します。ポインタが再び指差しマークに変わります。

ビットマップ・チェックポイントのプロパティ・ダイアログ・ボックス

詳細	ビットマップ・チェックポイントを作成したり変更できます。
アクセス方法	参照先 582 ページ「ビットマップ・チェックポイントの作成と変更」
重要な情報	➤［**詳細モード / シンプル モード**］ボタン このダイアログ・ボックスを詳細モードとシンプル・モードに切り替えます。詳細モードではダイアログ・ボックスが拡張され，次のことができます。 ➤ ビットマップの特定サブエリアを選択する ➤ RGB とピクセルの許容範囲を設定する ➤ カスタム・コンペアラを選択する（可能な場合） ➤ チェックポイントのタイムアウトを設定する ➤ リポジトリ内のチェックポイントを表示する これらのオプションを表示または編集するには，［**詳細モード**］ボタン をクリックします。シンプル・モードに戻るには，［**シンプル モード**］ボタン をクリックします。 ➤ **注**：Quality Center ユーザには，詳細なチェックポイント・プロパティを表示または編集するオプションはありません。そのため，詳細モードで使用できるオプションを 1 つ以上選択しているときに，Quality Center ユーザが Quality Center 内のチェックポイントを表示すると，検査のために選択されている一部のプロパティが表示されないことを示すテキストがダイアログ・ボックスに表示されます。
詳細情報	**概念の概要**：579 ページ「ビットマップの検査について」

標準設定では，［ビットマップ チェックポイントのプロパティ］ダイアログ・ボックスがシンプル・モードで開きます。



詳細モードではダイアログ・ボックスが拡張され，より多くのオプションが表示されます。

この画像は一例です。既存のチェックポイントを編集しながら検査する領域を選択するときに使用できるオプションを示しています。

[ビットマップ・チェックポイントのプロパティ] ダイアログ・ボックスの詳細

このダイアログ・ボックスでは，オプションがいくつかのグループに分かれています。以下の項で説明します。

- 588 ページ「説明情報」
- 589 ページ「検査する領域を選択するオプション」
- 590 ページ「[許容範囲オプション]：」
- 591 ページ「チェックポイント・タイムアウトオプション」

［説明情報］オプションの大半がシンプル・モードおよび詳細モードで使用できます。その他のオプションはすべて，詳細モードでのみ使用できます。

説明情報

説明情報は，［チェックポイントのプロパティ］ダイアログ・ボックスの上部に表示されます。

- ［**名前**］：標準設定では，チェックポイントの名前は，チェックポイントが作成されたテスト・オブジェクトの名前と同じです。QuickTest によってチェックポイントに割り当てられた名前を受け入れるか，別の名前を指定します。

  チェックポイントの名前を変更する場合は，名前について以下のことを確認します。

  - 一意である
  - 先頭および最後がスペースでない
  - "（二重引用符）が含まれていない
  - 次の文字の組み合わせが含まれていない： :=@@

- ［**クラス**］：チェックポイントを作成したテスト・オブジェクトのタイプ （読み取り専用）。

- ［**コンペアラ**］：QuickTest がチェックポイントを実行するのに使用するコンペアラを選択できます。QuickTest の標準コンペアラまたはカスタム・コンペアラを選択できます。カスタム・コンペアラを選択すると，このダイアログ・ボックスとはオプションの一部が異なります。詳細については，591 ページ「［ビットマップ チェックポイントのプロパティ］ダイアログ・ボックスのカスタム・コンペアラ・オプション」を参照してください。

  このオプションが使用できるのは，QuickTest コンピュータにカスタム・コンペアラがインストールされ，登録されている場合だけです。そうでなければ，QuickTest の標準コンペアラが使用されます。詳細については，580 ページ「ビットマップ比較の微調整」を参照してください。

  T このオプションは詳細モードでのみ使用できます。

- ［**ビットマップ表示領域**］：選択したオブジェクトのビットマップが表示されます。

➤［**リポジトリ内を検索**］: リポジトリ内のチェックポイントを表示するには，［**名前**］ボックスの右にある［**リポジトリ内を検索**］ボタン をクリックします。

(このオプションは，新しいチェックポイントの作成時には使用できません。既存のチェックポイントを編集するときのみ利用できます。

T このオプションは詳細モードでのみ使用できます。

検査する領域を選択するオプション

検査する領域を選択するオプションは，ビットマップ表示領域の下に表示されます。

➤［**ビットマップ全体の検査**］/［**選択領域のみをチェック**］: チェックポイントをビットマップ全体とと比較するか，ビットマップの特定領域のみと比較するか指定できます。［**選択領域のみをチェック**］を選択した場合，カーソルをビットマップ表示領域に移動すると，カーソルが十字形のポインタに変わります。十字形のポインタを使って，選択領域を指定する矩形を描きます。矩形を削除するには，もう一度クリックします。

十字形のポインタが表示されている間は，ポインタの現在位置の座標がビットマップ表示領域の右下隅に表示されます。十字形のポインタで矩形を描くときには，ポインタの近くで現在 選択されている領域のサイズがツールチップに表示されます。

ビットマップの特定領域のみと比較するチェックポイントを定義すると，［テスト結果］ウィンドウに表示される実際および期待のビットマップでも選択領域が強調表示されます。

➤［**選択領域のみ保存する**］: オブジェクトの選択領域だけをテストと一緒に保存できます（ディスク領域を節約するため)。［**OK**］をクリックすると，チェックポイントに保管されたビットマップが切り取られます。［テスト結果］ウィンドウに，ビットマップの選択した領域だけが表示されます。

このオプションが使用できるのは，［**選択領域のみをチェック**］を選択して，領域を指定する矩形を描いた後だけです。

注 :［**選択領域のみを保存**］チェック・ボックスを選択した場合，選択領域内で小さな領域を選択することによってチェックポイントを後で変更できますが，その前のサイズにビットマップを戻すことはできません。［**更新モード**］オプション（［**オートメーション**］＞［**更新モード**］）では，ビットマップの保存領域だけが更新されます。このオプションでは，オリジナルのフル・サイズのオブジェクトは更新されません。チェックポイントのオブジェクトを増やすには，新しいチェックポイントを作成します。

［許容範囲オプション］:

許容範囲オプションは，検査する領域を選択するオプションの下に表示されます。

➤［**RGB の許容範囲**］: 実際のビットマップにあるピクセルの RGB 値が期待のビットマップの RGB 値との間で許容され，チェックポイントが成功する割合を定義できます。

チェックボックスを選択し，手作業または上下の矢印でパーセンテージを変更します。詳細については，580 ページ「ビットマップ比較の微調整」を参照してください。

このオプションは 24 ビットのカラー・デプスを持つビットマップに限られます。

➤［**ピクセルの許容範囲**］: 期待のビットマップとの間で許容され，チェックポイントが成功する実際のビットマップのピクセル数またはピクセルの割合を定義できます。

チェックボックスを選択し，［**パーセント**］または［**ピクセル**］ラジオ・ボタンを選択して，手作業または上下の矢印で値を変更します。許容値を入力した後で，［**パーセント**］または［**ピクセル**］ラジオ・ボタンを切り替えると，その選択に基づいて値が再計算されます （100% は期待のビットマップまたは選択領域にあるピクセルの総数です)。

詳細については，580 ページ「ビットマップ比較の微調整」を参照してください。

チェックポイント・タイムアウトオプション

チェックポイント・タイムアウトオプションは，［チェックポイントのプロパティ］ダイアログ・ボックスの下部に表示されます。

- ［**チェックポイントのタイムアウト**］：QuickTest が正常にチェックポイントを実行する時間間隔（秒単位）を指定できます。チェックポイントの実行は，チェックポイントが成功するかタイムアウトが生じるまで試み続けられます。タイムアウトが生じる前にチェックポイントが成功しなければ，チェックポイントは失敗となります。

  たとえば，オブジェクトが特定の状態になるまでいくらか時間を要するとします。この場合にチェックポイントのタイムアウト値を大きくしておけば，オブジェクトがその特定の状態になるまで十分な時間を確保でき，タイムアウトに達するまでに（データが一致すれば）チェックポイントが成功することが可能になります。

  チェックポイントのタイムアウトを **0** 以外に指定して，チェックポイントに失敗した場合，［テスト結果］ウィンドウにチェックポイントのタイムアウトに関する情報が表示されます。

［ビットマップ チェックポイントのプロパティ］ダイアログ・ボックスのカスタム・コンペアラ・オプション

［ビットマップ チェックポイントのプロパティ］ダイアログ・ボックスで，カスタム・コンペアラを選択してビットマップ比較を実行する場合，ビットマップの郎域を選択するオプションと許容範囲を設定するオプションは使用できません。

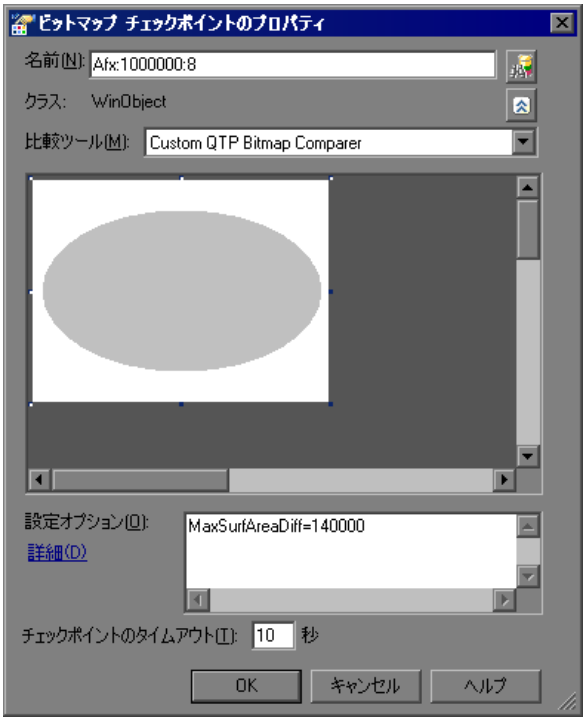
代わりに，次のオプションが使用できます（カスタム・コンペアラでサポート）。

- ［**設定オプション**］：任意の設定オプションがサポートされているため，カスタム・コンペアラに入力を（文字列形式で）供給できます。たとえば，許容範囲，許容サイズ偏差，ビットマップの位置などを指定できます。

  標準設定では，このボックスには，カスタム・コンペアラの供給する設定文字列が表示されます（可能な場合）。

- ［**詳細**］：カスタム・コンペアラによって提供されるヘルプ情報が表示されます（可能な場合）。このヘルプには，コンペアラに設定値を入力するための手順，コンペアラがビットマップを比較するのに使用するアルゴリズムに関する情報，カスタム・コンペアラの使用についての説明などが含まれます。

下図に，カスタム・コンペアラを選択した［ビットマップ チェックポイントのプロパティ］ダイアログ・ボックスを示します。



第 21 章

値の出力

QuickTest では，コンポーネントの値を取得し，その値を出力値オブジェクトに格納できます。そして，その値を実行セッションの別の段階で入力値として使用できます。

本章の内容

- 値の出力について（593 ページ）
- 出力値の作成（594 ページ）
- プロパティ値の出力（595 ページ）
- 出力タイプおよび設定の指定（602 ページ）

値の出力について

「**出力値**」ステップとは，実行セッション中にコンポーネントの特定のポイントで 1 つ以上の値がキャプチャされ，格納されるステップです。その値は，後で実行セッションの別のポイントで入力値として使用できます。

出力値ステップを作成するときに，実行セッション中の値の格納場所，および値の使用方法を指定できます。実行セッション中，QuickTest により，指定されたポイントで値が取得され，指定された場所に格納されます。実行セッション中にその値が必要になると，QuickTest によって，値がこの指定された場所から取得され，必要に応じて使用されます。

出力値の格納は，実行セッション中にのみ行われます。実行セッションが繰り返されると，出力値はリセットされます。

注：実行セッション後，セッション結果の一部として，セッション中に取得された出力値を表示できます。詳細については，733 ページ「[テスト結果］ウィンドウでのパラメータ化された値と出力値の結果の表示」を参照してください。

キーワード・コンポーネントで出力値を作成すると，［出力値のプロパティ］ダイアログ・ボックスがシンプル・モードで開き，Quality Center に表示できるプロパティと値だけが表示されます。このダイアログ・ボックスの詳細については，597 ページ「標準出力値の定義」を参照してください。

出力値の詳細なプロパティを表示または選択するには，［**詳細モード**］ボタンをクリックします。ただし，Quality Center ユーザには，詳細なプロパティを表示または編集するオプションはありません。したがって，詳細なプロパティを取得するように出力値オプションが設定されている場合，Quality Center のユーザが Quality Center でその設定を表示すると，プロパティの一部が出力に選択されているものの表示されないことを示すテキストが［出力値のプロパティ］ダイアログ・ボックスに表示されます。

出力値の作成

標準出力値を使用すれば，ほとんどのオブジェクトのプロパティ値を出力できます。たとえば，出力値としてオブジェクトの**テキスト**・プロパティを指定することによって，標準出力値でテキスト文字列を出力できます。

標準出力値の詳細については，595 ページ「プロパティ値の出力」を参照してください。

出力値の表示と編集

テストに出力値ステップを挿入すると，［キーワード ビュー］にそのステップが表示され，［**操作**］カラムには Output，［**値**］カラムには CheckPoint，その後には出力値に割り当てられた名前がそれぞれ表示されます。

プロパティ値の出力

標準出力値を使用すれば，ほとんどのオブジェクトのプロパティ値を出力できます。

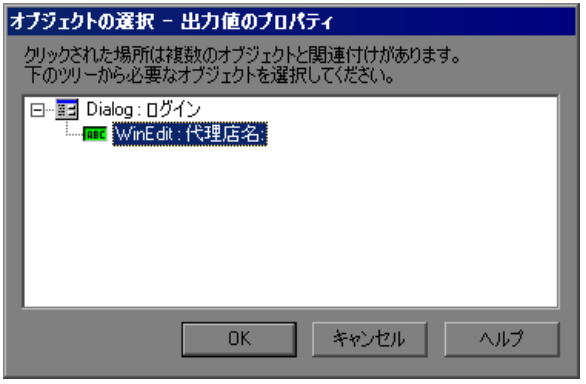
コンポーネントの記録中に，標準出力値を作成できます。

注：コンポーネントでは，画像，テーブル，または（Web）ページの出力値を作成することはできません。これらの特殊なタイプの出力値は，テストでのみ利用できます。ただし，コンポーネントの標準出力値を作成するときに Web ページまたは任意のテーブル・オブジェクトを選択した場合は，ほかのすべてのオブジェクトと同様に，そのオブジェクトのプロパティを検査できます。

記録中に標準出力値を作成するには，次の手順を実行します。

1 ［**挿入**］＞［**出力値**］＞［**標準出力値**］を選択します。または，ツールバーの［**チェックポイントまたは出力値の挿入**］ボタンの横にある矢印をクリックをクリックして［**標準出力値**］を選択します。ポインタが指差しマークに変わります。指差しマーク機能の使用方法の詳細については，596 ページ「指差しマークの使用法に関するヒント」を参照してください。

2 アプリケーションで，出力値を指定するオブジェクトをクリックします。クリックした場所が複数のオブジェクトに関連付けられている場合，［**オブジェクトの選択－出力値のプロパティ**］ダイアログ・ボックスが開きます。



3 ［オブジェクトの選択］ダイアログ・ボックスで，出力値を指定するオブジェクトを選択し，［**OK**］をクリックします。選択したオブジェクトの［**出力パラメータのプロパティ**］ダイアログ・ボックスが開きます。

4 出力するプロパティ値とその設定を指定します。詳細については，597 ページ「標準出力値の定義」を参照してください。

5 出力値の詳細の定義が終了したら，［**OK**］をクリックします。QuickTest によってコンポーネントに出力値ステップが挿入されます。

指差しマークの使用法に関するヒント

➤ 左の CTRL キーを押したままにすると，指差しマークが標準ポインタに変わります。そのときに，QuickTest またはアプリケーションでウィンドウのフォーカスを変更したり，操作（右クリックしたり，スクロール・バーを使ったり，ポ

インタをオブジェクト上に移動してショートカット・メニューを表示するなど）を実行できます。

- 選択するオブジェクトがあるウィンドウが別のウィンドウで部分的に隠れている場合は，部分的に隠れているウィンドウが前面に表示されるまで，その上に指差しマークを数秒間置いたままにします。次に，必要なオブジェクトをポイントしてクリックします。ウィンドウを前面に出すのに必要な時間の設定は，［オプション］ダイアログ・ボックスの［一般］表示枠で行えます。
- 選択するオブジェクトがあるウィンドウが別のウィンドウで完全に隠れている場合，またはダイアログ・ボックスがウィンドウの背後に隠れている場合は，必要に応じて左の CTRL キーを押しながらウィンドウを配置します。
- 選択するオブジェクトがあるウィンドウが最小化されている場合は，左の CTRL キーを押したまま，Windows タスク・バーでアプリケーションを右クリックして，ショートカット・メニューから［**元のサイズに戻す**］を選択すると，そのウィンドウを表示できます。
- 特定のイベント（ショートカット・メニューを表示するために右クリックしたり，オブジェクト上にポインタを移動するなど）を実行するだけで，選択するオブジェクトを表示できる場合は，左の CTRL キーを押したままにします。指差しマークが一時的に標準ポインタに変わり，イベントを実行できます。選択するオブジェクトが表示されたら，左の CTRL キーを放します。ポインタが再び指差しマークに変わります。

標準出力値の定義

［出力値のプロパティ］ダイアログ・ボックスでは，選択した値ごとに出力および設定対象プロパティ値を選択できます。［出力パラメータのプロパティ］ダイアログ・ボックスには，シンプル・モードと詳細モードの 2 つのモードが

あります。どちらのモードが開くかは，どの出力に対して詳細なプロパティが選択されているかによって変わります。

新しい出力値ステップを挿入するときは，ダイアログ・ボックスが常にシンプル・モードで開きます。既存の出力値オブジェクトに対してダイアログ・ボックスを開いた場合，詳細なチェックポイント・プロパティが選択されていなければ，［出力パラメータのプロパティ］ダイアログ・ボックスはシンプル・モードで開きます。シンプル・モードでは，基本的なプロパティのみが表示されます。詳細なプロパティを表示または選択するには，［**詳細モード**］ボタンをクリックします。詳細モードでは，このダイアログ・ボックスに追加のオプションも表示されます。詳細については，600 ページ「[出力値のプロパティ］ダイアログ・ボックスの詳細モード」を参照してください。

注：Quality Center ユーザには，詳細なプロパティを表示または編集するオプションはありません。したがって，出力値で詳細なプロパティが選択されている場合，ユーザが Quality Center でそのプロパティを表示すると，プロパティの一部が出力に選択されているものの表示されないことを示すテキストが［出力値のプロパティ］ダイアログ・ボックスに表示されます。

ダイアログ・ボックスを閉じる前に，同じオブジェクトに対して出力するプロパティを複数選択し，各プロパティ値に出力設定を定義することができます。実行セッション中に出力値ステップに到達すると，QuickTest によって，指定したすべてのプロパティ値が取得されます。

出力値の識別

ダイアログ・ボックスの上部に，次のような出力値についての情報が表示されます。

項目	説明
［名前］	出力値に割り当てられた名前。標準設定では，出力値名は出力値ステップが実行されるテスト・オブジェクトの名前です。出力値に別の名前を指定することも，標準設定の名前を受け入れることもできます。 出力値の名前を変更する場合は，名前について以下のことを確認します。 ➤ 一意である ➤ 先頭および最後がスペースでない ➤ "（二重引用符）が含まれていない ➤ 次の文字の組み合わせが含まれていない：:=@@
［クラス］	テスト・オブジェクトのタイプ。

出力するプロパティ値の選択

ダイアログ・ボックスの上部には，選択したオブジェクトのプロパティを値およびタイプとともに示す枠があります。この表示枠には次の項目が表示されます。

表示枠内の要素	説明
チェック・ボックス	出力するプロパティを指定するには，対応するチェック・ボックスを選択します。オブジェクトの複数のプロパティを選択し，選択した各プロパティ値に出力オプションを指定できます。
［タイプ］	アイコンは，プロパティの値が現在は定数であることを示します。 アイコンは，プロパティの値が現在はパラメータであることを示します。
［プロパティ］	プロパティの名前。
［値］	プロパティの現在の値。詳細については，600 ページ「プロパティ値の出力設定の指定」を参照してください。

プロパティ値の出力設定の指定

［出力パラメータのプロパティ］ダイアログ・ボックスで選択したプロパティの［参照］ボタン ... をクリックすると，［パラメータ化 / プロパティ］ダイアログ・ボックスが開き，選択したプロパティ値の出力定義が表示されます。

出力対象プロパティ値を選択すると，次のことが可能になります。

- ［**変更**］ボタンをクリックすれば，選択した値の出力タイプ，出力設定，あるいはその両方を変更できます。［出力オプション］ダイアログ・ボックスが開き，値の現在の出力タイプと設定が表示されます。詳細については，602 ページ「出力タイプおよび設定の指定」を参照してください。
- ［**OK**］をクリックすれば，表示されている出力定義をそのまま使用できます。

出力値ステップの場所の指定

ダイアログ・ボックスの下部に［**ステートメントの挿入**］領域が表示されている場合は，テストに新しい出力値ステップを挿入する場所を指定できます。詳細については，602 ページ「出力値ステップの場所の選択」を参照してください。

［出力値のプロパティ］ダイアログ・ボックスの詳細モード

出力の詳細なプロパティが選択されている場合は，［出力値のプロパティ］ダイアログ・ボックスが詳細モードで開きます。このダイアログ・ボックスには，オブジェクトのサポートされているプロパティと期待値がすべて表示されます。出力の詳細なプロパティが選択されていない場合は，［**シンプル モード**］

ボタンをクリックして，シンプル・モードのプロパティとオプションだけを表示できます。

詳細モードでは，［出力値のプロパティ］ダイアログ・ボックスに［シンプルモード］オプション（597 ページを参照）と以下の追加オプションが表示されます

［リポジトリ内を検索］

リポジトリ内の出力値を表示するには，［**名前**］ボックスの右にある［**リポジトリ内を検索**］ボタン をクリックします。（このオプションは，新しい出力値の作成時には使用できません。既存の出力値を編集するときのみ使用できます）。

［値の設定］領域

プロパティのチェック・ボックスを選択すると，プロパティの詳細が強調表示され，選択したプロパティ値の現在の出力定義が［**値の設定**］領域に表示されます。この領域を使用するのは，シンプル・モードで［参照］ボタンから開かれる［パラメータ化 / プロパティ］ダイアログ・ボックスを使うときです。詳細については，600 ページ「プロパティ値の出力設定の指定」を参照してください。

出力タイプおよび設定の指定

各値に定義した出力の設定によって，出力の格納場所と実行セッションにおける使用方法が決定します。出力値のステップに到達すると，QuickTest によって，出力対象として選択された値が取得され，後の実行セッションで使用できるように指定された場所に保存されます。詳細については，602 ページ「出力値ステップの場所の選択」を参照してください。

出力値ステップの場所の選択

コンポーネントの編集中に出力値を作成すると，ダイアログ・ボックスの下部に［**ステートメントの挿入**］領域が表示されます。

標準設定では，新しい出力値ステップは，現在のステップ（［**出力値**］オプションを選択したときに選んでいたステップ）の前に挿入されます。［**現在のステップの後**］オプションを選択すれば，現在のステップの後に新しい出力値ステップを挿入できます。

注：このオプションは，記録時には使用できません。QuickTest によって，その前に記録されたステップの後に自動的に新しい出力値ステップが挿入されます。また，既存の出力値ステップを変更しているときにも，このオプションは使用できません。

第 22 章

Windows ベースのオブジェクトのテキスト認識を使用した作業

QuickTest では，Windows ベースのオブジェクト内のテキスト文字列を識別するためにさまざまなメカニズムを利用しています。本章では，Windows ベースのオブジェクトに対するテキスト認識の結果を最適化するように QuickTest を設定する方法について説明します。

本章の内容

- Windows ベースのオブジェクトのテキスト認識を使用した作業について（604 ページ）
- ［オプション］ダイアログ・ボックスの［一般］＞［テキスト認識］表示枠（604 ページ）
- テキスト認識のガイドライン（608 ページ）
- テキスト認識および開発環境（609 ページ）
- 使用事例シナリオ：画像内のテキストの検査（612 ページ）

Windows ベースのオブジェクトのテキスト認識を使用した作業について

QuickTest では，Windows API ベースのメカニズムまたは OCR（光学式文字認識）メカニズムを利用して，アプリケーション内のテキストを識別します。テキストおよびテキスト領域チェックポイントまたは出力値コマンドを使用して，Windows ベースのオブジェクト内にあるテキストを確認または取得できます。あるいは，**testobject**.**GetText**（ターミナル・エミュレータ・オブジェクト用），**testobject**.**GetVisibleText**，もしくは **testobject**.**GetTextLocation** テスト・オブジェクト・メソッド，または **TextUtil.GetText** もしくは **TextUtil.GetTextLocation** 予約済みオブジェクト・メソッドを使って，必要なテキストをキャプチャできます。

標準設定では，QuickTest は Windows API ベースのメカニズムを利用して，オブジェクトからテキストを直接取得しようとします。このような方法でテキストを取得できない場合（テキストが画像の一部である場合など），QuickTest は OCR（光学文字認識）メカニズムを利用してテキストをキャプチャしようとします。望ましいテキスト認識メカニズムおよび OCR 別の設定を指定するには，［テキスト認識］表示枠を使います。

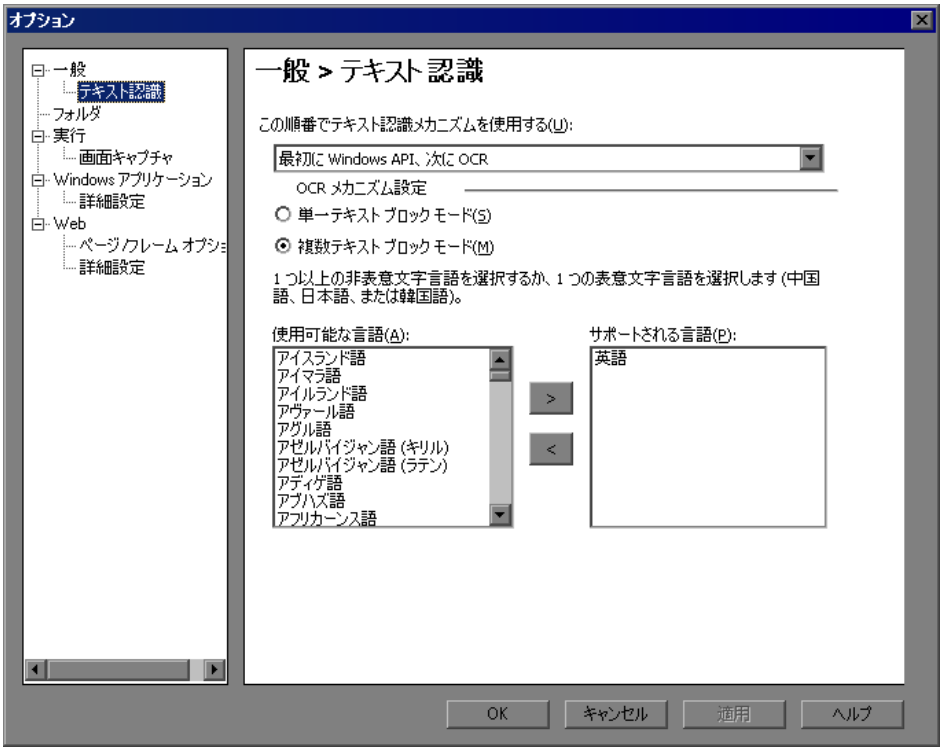
テキスト / テキスト領域チェックポイントまたは出力値を挿入する前に，608 ページ「テキスト認識のガイドライン」を確認してください。

［オプション］ダイアログ・ボックスの［一般］＞［テキスト認識］表示枠

［**詳細**］	アプリケーション内のテキストを認識する方法を設定できます。この表示枠では，標準のテキスト・キャプチャ・メカニズム，OCR（光学式文字認識）メカニズム・モード，および OCR メカニズムでテキストを認識するのに使用する言語辞書を変更できます。
［**Accessed by**］	［**ツール**］メニュー＞［**オプション**］項目＞［**一般**］ノード＞［**テキスト認識**］ノード

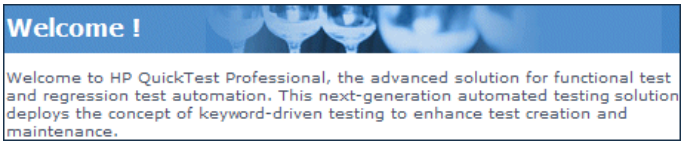
重要な情報	［一般］＞［テキスト認識］オプションは，標準 Windows，.NET WinForms，WPF，SAP Gui for Windows，Visual Basic，ActiveX など，Windows ベースのオブジェクトにのみ関係があります。
［Learn More］	**概念の概要**：604 ページ「Windows ベースのオブジェクトのテキスト認識を使用した作業について」 **その他の関連項目**：607 ページ「その他の参照」

［オプション］ダイアログ・ボックスの［一般］＞［テキスト認識］表示枠を下図に示します。



［一般］＞［テキスト認識］表示枠には次のオプションがあります。

オプション	説明
［テキスト認識メカニズムをこの順に使用する］	テキストをキャプチャするときに使用されるテキスト認識メカニズムを指定します。 **取り得る値：** **最初に Windows API，次に OCR**：（標準）最初に Windows API ベースのメカニズムを使用して，オブジェクトからテキストを直接取得するように QuickTest を設定します。テキストを取得できない場合は（テキストが画像の一部の場合など），QuickTest は，OCR（光学文字認識）メカニズムを使用してテキストの取得を試みます（CJK 言語（中国語，日本語，韓国語）を扱う場合は，この設定を強くお勧めします）。 **最初に OCR，次に Windows API**：最初に OCR メカニズムを使用してオブジェクトからテキストを取得するように QuickTest を設定します。テキストを取得できなかった場合，QuickTest は Windows API ベースのメカニズムを使用してオブジェクトからテキストを取得します。 **Windows API のみ使用する**：Windows API ベースのメカニズムのみを使用して，オブジェクトからテキストを取得するように QuickTest を設定します（OCR メカニズムは使用しません）。 **OCR のみ使用する**：OCR メカニズムのみを使用して，オブジェクトからテキストを取得するように QuickTest を設定します（Windows API ベースのメカニズムは使用しません）（Windows Vista で作業するときは必須です）。 Windows ベース環境におけるテキスト認識のサポートの詳細については，『**HP QuickTest Professional 最初にお読みください**』を参照してください。
単一テキスト・ブロック・モード	オブジェクトのテキストがフォント，サイズ，色，および背景に関して統一されている場合は，このボタンを選択します。例を次に示します。 単一テキスト・ブロック・モードでは，当該領域に焦点を合わせ，単一テキスト・ブロックとして扱うよう OCR メカニズムに指示します。このモードは，小さなオブジェクト上または小さなテキスト領域でテキストをキャプチャする場合に役立ちます。

オプション	説明
複数テキスト・ブロック・モード	このラジオ・ボタンを選択するのは，オブジェクト上のテキストがさまざまなフォント，フォント・サイズ，色，および / または背景で構成されている場合だけです。例を次に示します。  複数テキスト・ブロック・モードでは，オブジェクト内で背景，フォント，およびサイズが異なる各テキスト領域を処理するよう OCR メカニズムに指示します。OCR メカニズムにより，内部アルゴリズムに従ってテキスト・ブロックをどこで分割するかが決定されます。
［使用可能な言語］	オブジェクトからテキストを取得するときに，OCR メカニズムで使用される可能性のある言語辞書がすべて表示されます。 **OCR メカニズムで使用される言語辞書を指定するには，次の手順を実行します。**言語を選択し，右向き矢印ボタン（**>**）をクリックして，言語を［**サポート言語**］リスト・ボックスに移動します。
［サポート言語］	テキストをキャプチャするときに OCR メカニズムで使用される言語辞書が表示されます。［**サポート言語**］リスト・ボックスには，次のどちらかを表示できます。 ➤ 1 つの CJK 言語（中国語，日本語，韓国語）（**注**：標準設定では，CJK 言語のテキストをキャプチャするときには英語もサポートされます）。 ➤ 1 つ以上の非 CJK 言語。 ［**サポート言語**］**リストから言語辞書を削除するには，**言語を選択して左向き矢印（**<**）をクリックします。

その他の参照

［関連使用事例シナリオ］	612 ページ「使用事例シナリオ：画像内のテキストの検査」
［関連する概念］	608 ページ「テキスト認識のガイドライン」

テキスト認識のガイドライン

- OCR メカニズムを使用する場合，テキストが大きいほど，テキスト認識が向上します。
- 選択するテキスト領域のサイズをできるだけ小さくしてください。認識するテキストに不要な文字が付加されないようにします。

  同時に，ウィンドウ内でオブジェクトが動く可能性（座標の変化）を考慮します。たとえば，コンピュータが異なると画面解像度も異なることが多く，アプリケーション内のオブジェクトの座標に影響を与える可能性があります。また，アプリケーションの設計および開発中には，ほかのオブジェクトや美観のためにオブジェクトを動かしてスペースをあけることもあります。

  オペレーティング・システム，インストールされているサービス・パック，インストールされているツールキットなどによって，アプリケーションにおけるオブジェクトのサイズや位置が影響を受ける可能性があることを考慮します。選択するテキスト領域のサイズは，さまざまなシステム構成でも十分広いものにします。

  選択するテキスト領域のサイズは，これらの問題を考慮に入れて十分広いものにする必要があります。
- どちらのテキスト・ブロック・モードを使用すべきかわからない場合は，単一テキスト・ブロック・モードを使用します。単一テキスト・ブロック・モードで実行されるテキスト・キャプチャのほうが通常，複数テキスト・ブロックでのテキストキャプチャより正確になるからです。結果が期待したものでなければ，複数テキスト・ブロック・モードを試します。異なるテキスト・ブロック・モードを使用する場合の例については，612 ページ「使用事例シナリオ：画像内のテキストの検査」を参照してください。

ヒント：さまざまなフォントや背景が含まれている大きな領域にテキスト認識メカニズムを使用する場合は，複数のテキスト・ブロックをキャプチャする 1 つのステップを作成するのではなく，各単一テキスト・ブロックのテキストをキャプチャする複数のステップを作成することをお勧めします。

- Windows にはさまざまなテーマが用意されています。テキスト認識を使用するときは，以下の順序でテーマを適用します。
  - Windows Vista のテーマ（最良の結果のため）
  - Windows XP のテーマ
  - Windows Classic のテーマ
- 複数テキスト・ブロック・モードを使用していて，テキスト認識メカニズムによって不要なテキスト情報（隠しテキストや同じ文字列を多重表示する影付きテキストなど）が得られた場合は，単一テキスト・ブロック・モード・オプションを使用します。
- テキスト認識を使用するステップを実行するときに，テキスト認識オプションが Windows API メカニズムを使用するように設定されていると，Windows API がテキストをキャプチャするときに，アプリケーションで「点滅効果」を引き起こすことがあります。テキスト認識メカニズムを利用する連続的なステップがテストに含まれている場合，あるステップの「点滅効果」がその後のテキスト認識ステップ（またはビットマップ・チェックポイントなど，アプリケーションの外観に依存するほかのステップ）を失敗させることもあります。

  これに対処するには，そのような各ステップの前に **Wait** ステートメントを挿入します。これによって，前のステップの Windows API キャプチャが完了するまで，次のテキスト認識ステップの実行を遅らせることができます。
- オブジェクトの **text**（または同様の）プロパティを使用し，必要なテキストが含まれているオブジェクトに標準チェックポイントを挿入して，アプリケーション・ウィンドウからテキストを検査することを強くお勧めします。

テキスト認識および開発環境

次の表に，QuickTest で（アドインによって）サポートされている開発環境を示し，テキスト認識に関してサポートされている機能を明示します。

開発環境	テキスト認識	
	サポート	非サポート
ActiveX	テキスト認識の完全サポート	N/A
Delphi	テキスト認識の完全サポート	N/A

開発環境	テキスト認識	
	サポート	非サポート
Java	➤ テキスト・チェックポイント ➤ テキスト出力値 ➤ テキスト領域チェックポイント ➤ テキスト領域出力値	➤ **GetTextLocation** メソッド ➤ **GetVisibleText** メソッド
.NET WebForms	➤ ページ・オブジェクト専用のテキスト・チェックポイント ➤ ページ・オブジェクト専用のテキスト出力値	➤ その他のオブジェクト用のテキスト・チェックポイント ➤ その他のオブジェクト用のテキスト出力値 ➤ その他のオブジェクト用のテキスト領域チェックポイント ➤ その他のオブジェクト用のテキスト領域出力値 ➤ **GetTextLocation** メソッド ➤ **GetVisibleText** メソッド
.NET WinForms	テキスト認識の完全サポート	N/A
Oracle	N/A	テキスト認識は未サポート
PeopleSoft	➤ PSFrame オブジェクト専用のテキスト・チェックポイント ➤ PSFrame オブジェクト専用のテキスト出力値	➤ その他のオブジェクト用のテキスト・チェックポイント ➤ その他のオブジェクト用のテキスト出力値 ➤ その他のオブジェクト用のテキスト領域チェックポイント ➤ その他のオブジェクト用のテキスト領域出力値 ➤ **GetTextLocation** メソッド ➤ **GetVisibleText** メソッド
PowerBuilder	テキスト認識の完全サポート	N/A

開発環境	テキスト認識	
	サポート	非サポート
SAP Gui for Windows	N/A	テキスト認識は未サポート
SAP Web	➤ テキスト・チェックポイント ➤ テキスト出力値	➤ テキスト領域チェックポイント ➤ テキスト領域出力値 ➤ **GetTextLocation** メソッド ➤ **GetVisibleText** メソッド
Siebel	N/A	テキスト認識は未サポート
標準 Windows	テキスト認識の完全サポート	N/A
Stingray	テキスト認識の完全サポート	N/A
ターミナル・エミュレータ	TeScreen および TeTextScreen オブジェクト専用のテキスト出力値	➤ その他のテキスト・チェックポイント ➤ その他のテキスト出力値 ➤ テキスト領域チェックポイント ➤ テキスト領域出力値 ➤ **GetTextLocation** メソッド ➤ **GetVisibleText** メソッド
VisualAge	テキスト認識の完全サポート	N/A
Visual Basic	テキスト認識の完全サポート	N/A
Web	➤ ページ・オブジェクト専用のテキスト・チェックポイント ➤ ページ・オブジェクト専用のテキスト出力値	➤ その他のオブジェクト用のテキスト・チェックポイント ➤ その他のオブジェクト用のテキスト出力値 ➤ その他のオブジェクト用のテキスト領域チェックポイント ➤ その他のオブジェクト用のテキスト領域出力値 ➤ **GetTextLocation** メソッド ➤ **GetVisibleText** メソッド

開発環境	テキスト認識	
	サポート	非サポート
Web サービス	N/A	テキスト認識は未サポート
WPF	テキスト認識の完全サポート	N/A

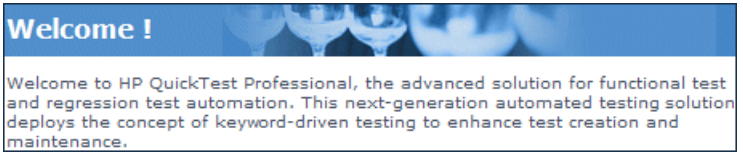
使用事例シナリオ：画像内のテキストの検査

Ben と George は品質保証エンジニアで，経験豊富な QuickTest ユーザです。George はテキスト認識についてもよく知っており，テキスト認識メカニズムについて基本的に理解しています。

Ben はテストするユーザ・インタフェースのさまざまなアイコンや画像の外観をテストするために，ビットマップ・チェックポイントをよく使っています。

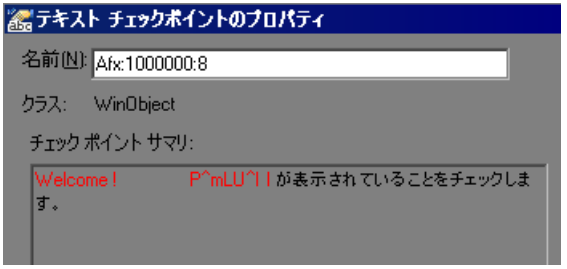
そのプロジェクトの 1 つで，Ben はグラフィック内のテキストを確認する必要があったので，テキスト・チェックポイントを使用することにしました。

Ben は Welcome ! というテキストが次のグラフィックに正しく表示されるか検査するテキスト・チェックポイントを挿入して，確認プロセスを開始することにしました。



このテキスト・チェックポイントを挿入する前に，Ben は［テキスト認識］表示枠を開き，テキスト認識設定を行いました。テキストがグラフィックの一部だったので，Ben はテキスト認識メカニズムを［**OCR のみ使用する**］に設定しました。Ben はまた，単一テキスト・ブロック・モードが通常は一番うまく機能することを知っていたので，［**単一テキスト・ブロック・モード**］オプションを選択しました。

次に，Ben は上記の領域全体にテキスト・チェックポイントを挿入しました。［テキスト・チェックポイントのプロパティ］ダイアログ・ボックスには次のような結果が表示されました。

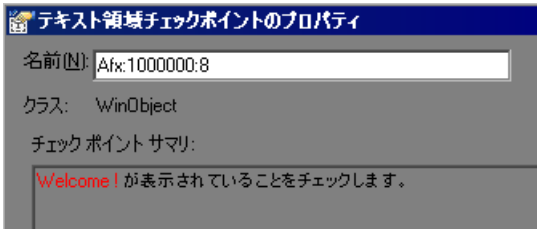


Ben はテキスト・チェックポイントの［**チェックポイント サマリ**］領域に余分な文字があることに気付きましたが，その理由はわかりませんでした。

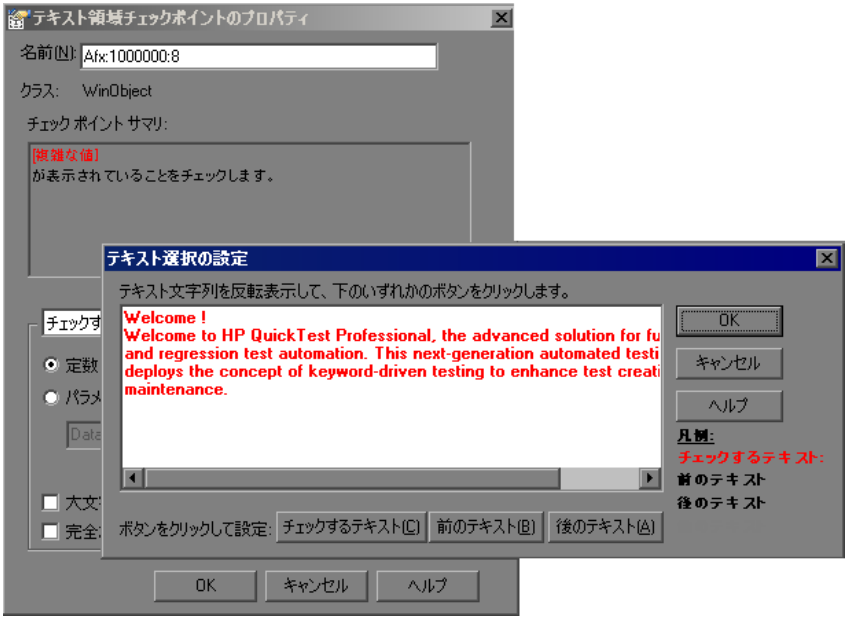
Ben は同僚の George に助けを求めました。George は Ben に，テキスト認識メカニズムがテキストを正しく認識しないと，余分な文字がテキスト・チェックポイントに追加されることがあると説明しました。

George はまた，Ben がテキスト・チェックポイントに定義した領域は，テキストのフォント・サイズ，色，または背景が均一でないので，複数のテキスト・ブロックで構成されていることも指摘しました。タイトル領域は青灰色の背景に白文字で構成されていましたが，ほかのテキストは小さく，白の背景に青色のテキストで構成されていました。

Ben は，［一般］＞［テキスト認識］表示枠で［**単一テキスト・ブロック・モード**］オプションを選択したことを思い出し，単一テキスト・ブロック・モードを使用する場合は，グラフィック全体ではなく，グラフィックの Welcome ! 領域だけにテキスト・チェックポイントを作成する必要があったことを理解しました。これを試してみると，下に示すように，OCR メカニズムによってテキストが正しく認識されました。



Ben はこの結果に満足しましたが，ほかの可能性を調べる必要があったので，今度はグラフィック全体に別のテキスト・チェックポイントを挿入しました。［テキスト認識］表示枠で［**複数テキスト・ブロック・モード**］オプションを選択しました。その結果は次のようになりました。



OCR メカニズムによって，グラフィック内のテキストがすべて正しく認識されたことに Ben は満足しました。しかし，タイトルの Welcome ! だけをテストする必要があったので，Welcome ! の後にあるすべてのテキストを**後のテキスト**としてマークして，このチェックポイントを仕上げました。

2 つのチェックポイントが成功しても，テキスト・チェックポイントを 1 つだけにする必要がありました。（**単一テキスト・ブロック・モード**を使用した）最初のチェックポイントを維持することにし，2 番目のチェックポイントを削除しました。チェックポイントが今後のテスト実行で成功するように，［テキスト認識］表示枠で［**単一テキスト・ブロック・モード**］を選択しました。

第 23 章

正規表現の使い方

また，コンポーネントの柔軟性と適応性を高めるために，値に正規表現を使用することもできます。

本章の内容

- **正規表現について（617 ページ）**
- **正規表現の概要と使い方（617 ページ）**
- **正規表現の定義（619 ページ）**

正規表現について

正規表現は，複雑な検索条件を指定する文字列です。正規表現は，可変値が含まれるオブジェクトとテキスト文字列の識別に使用します。たとえば，ウィンドウのタイトルバーの名前がファイル名に応じて変化する場合，テスト・オブジェクト記述で正規表現を使用すると，タイトルバーに特定の製品名，そしてその後にダッシュとほかのテキストが続いて表示されるウィンドウを識別できます。

正規表現の概要と使い方

正規表現を使用すれば，さまざまな値を持つオブジェクトやテキスト文字列を QuickTest で識別できます。正規表現は次のような場合に使用できます。

- ダイアログ・ボックスまたはプログラム記述内にあるオブジェクトのプロパティ値を定義する場合
- ステップをパラメータ化する場合
- 可変値が含まれるチェックポイントを作成する場合

たとえば，オブジェクトのテキスト・プロパティが日付値であるが，表示される日付が現在の日付に従って変化する場合に正規表現を使用できます。日付を正規表現として定義すると，QuickTest は正確な日付値ではなく，期待される日付形式によるテキストが含まれているオブジェクトを認識します。

正規表現は，複雑な検索条件を指定する文字列です。ピリオド（.），アスタリスク（*），キャレット（^），大括弧（[ ]）などの特殊文字を使って，検索条件を定義できます。

注：

- 正規表現は，**string** タイプの値に対してのみ使用できます。
- 正規表現の特殊文字の直前に円記号またはバックスラッシュ（¥ または \）がある場合は，その後に続く特殊文字そのものが検索されます。

正規表現の使用例の詳細については，次の項目を参照してください。

- 618 ページ「プロパティ値に対する正規表現の使用」
- 619 ページ「チェックポイントでの正規表現の使用」

正規表現の構文を含む，正規表現の定義の詳細については，619 ページ「正規表現の定義」を参照してください。

プロパティ値に対する正規表現の使用

各実行セッションでプロパティ値が予測可能な方法で変化する場合は，［オブジェクト リポジトリ］ウィンドウやプログラム的記述などでプロパティ値を定義またはパラメータ化する際に，正規表現を使用できます。プログラム的記述の詳細については，350 ページ「プログラム的記述の使用」を参照してください。

たとえば，ユーザがデータを入力し，［**送信**］ボタンをクリックして送信するフォームが Web サイトにあるとします。入力を必要とするフィールドに入力がない場合は，フォームが再表示され，フィールド入力するように求められます。ユーザは入力を済ませたらフォームを再送信するために，［**再送信**］ボタンをクリックします。この場合，ボタンをクリックするときにボタン名が変わっていることを QuickTest が無視するように，ボタンの「**名前**」プロパティの値を正規表現として定義できます。

チェックポイントでの正規表現の使用

標準チェックポイントを使用してオブジェクトのプロパティ値を検証する場合，変化する値を持つオブジェクトを検証できるように，オブジェクトのプロパティの期待値を正規表現として設定できます。

たとえば，アプリケーションのあらゆるウィンドウとダイアログ・ボックスにアプリケーションの名前が含まれ，その後にハイフン（-）と説明的なタイトルが続いていることを確認したいとします。タイトルの最初の部分にアプリケーションの名前が含まれ，その後にハイフンが続いていることを確認するために，テスト内の各ダイアログ・ボックス・オブジェクトにチェックポイントを追加できます。

正規表現の定義

定数値，コンポーネントパラメータ値，または関数ライブラリ内で使用されるプログラム記述のプロパティ値に対しては，正規表現を定義できます。

［定数値オプション］ダイアログ・ボックスまたは［パラメータ オプション］ダイアログ・ボックスの［**値**］ボックスに，文字列の正規表現構文を入力することで，正規表現を定義できます。値を正規表現として扱うよう QuickTest に指示するには，［**正規表現**］チェック・ボックスを選択します。

すべてのプログラム的記述のプロパティ値は，自動的に正規表現として扱われます。プログラム的記述の詳細については，350 ページ「プログラム的記述の使用」を参照してください。

注：正規表現は，**string** タイプの値に対してのみ使用できます。

QuickTest の標準設定では，正規表現で，ピリオド（.），ハイフン（-），アスタリスク（*），キャレット（^），大括弧（[ ]），括弧（()），ドル記号（$），パイプ記号（|），プラス記号（+），疑問符（?），円記号またはバックスラッシュ（¥ または \）を除くすべての文字が文字どおりに扱われます。これらの特殊文字の直前に円記号またはバックスラッシュ（¥ または \）がある場合，QuickTest によってその文字がリテラル文字として扱われます。

［定数値オプション］ダイアログ・ボックスまたは［パラメータ オプション］ダイアログ・ボックスの［**値**］ボックスに特殊文字を入力すると，各特殊文字の前に円記号またはバックスラッシュ（¥ または \）を追加するかどうかが尋ねるメッセージが表示されます。［**はい**］をクリックすると，特殊文字の前に円記号またはバックスラッシュ（¥ または \）が追加され，その文字は文字どおりに扱われます。［**いいえ**］をクリックすると，その特殊文字は正規表現文字として扱われます。

本項では，正規表現の作成に使用可能な一般的なオプションの一部について説明します。

- 円記号またはバックスラッシュの使用 (**¥**)
- 任意の 1 文字と一致 (**.**)
- リストの任意の 1 文字と一致 (**[xy]**)
- リストにない任意の 1 文字と一致 (**[^xy]**)
- 範囲内の任意の 1 文字と一致 (**[x, y]**)
- 特定の文字が 0 回以上一致 (*****)
- 特定の文字が 1 回以上一致 (**+**)
- 特定の文字が 0 回または 1 回一致 (**?**)
- 正規表現のグループ化 (**( )**)
- 正規表現の 1 つと一致 (**|**)
- 行の先頭と一致 (**^**)
- 行の末尾と一致 (**$**)
- アンダスコアを含む任意の英数字との一致 (**¥w**)
- 英数字以外の文字との一致 (**¥w**)
- 正規表現演算子の結合

注：サポートされている正規表現文字の完全なリストと説明については，Microsoft VBScript ドキュメントの「Regular Expressions」の項を参照してください（［**ヘルプ**］＞［**QuickTest Professional ヘルプ**］を選択して QuickTest Professional ヘルプを開きます。次に，［**VBScript Reference**］＞［**VBScript**］＞［**User's Guide**］＞［**Introduction to Regular Expressions**］を選択します）。

円記号またはバックスラッシュの使用

円記号またはバックスラッシュ（¥ または \）は，2 つの役割を果たせます。特殊文字と併せて使用すると，次の文字をリテラル文字として扱うことを示します。たとえば，¥. は，ワイルドカードではなくピリオド（.）として扱われます。あるいは，n，t，w，または d のように通常はリテラル文字として扱われる文字と併せて使用すると，その組み合わせは特殊文字を示します。たとえば，¥n は改行文字を表します。

たとえば，次のような場合です。

- 「w」は文字「w」と一致します。
- 「¥w」は，アンダスコアを含む任意の文字と一致する特殊文字です。
- 「¥¥」はリテラル文字「¥」と一致します。
- 「¥(」はリテラル文字「(」と一致します。

たとえば，次のような Web サイトを検索するとします。

```
newtours.demoaut.com
```

この場合，ピリオドは正規表現の記号と間違えられます。ピリオドが正規表現の一部ではないことを示すには，次のように入力します。

```
newtours¥.demoaut¥.com
```

注：特別な意味を持たない文字の前に円記号またはバックスラッシュを使用すると，そのバックスラッシュは無視されます。たとえば，「¥z」は「z」と一致します。

任意の 1 文字と一致

ピリオド（.）は，任意の 1 文字を表します（「¥n」を除く）。たとえば，次のような場合です。

```
welcome.
```

これは，welcomes や welcomed など，スペースを含む任意の 1 文字が続く welcome と一致します。複数のピリオドはピリオドの数と同じ数の任意の文字の並びを示します。

¥n を含む任意の 1 文字と一致するには，次のように入力します。

```
(.|¥n)
```

正規表現文字 **( )** の詳細については，624 ページ「正規表現のグループ化」を参照してください。正規表現文字 **|** の詳細については，624 ページ「正規表現の 1 つと一致」を参照してください。

リストの任意の 1 文字と一致

大括弧は，候補文字のリスト内の任意の 1 文字を表す場合に指定します。たとえば，1967 年，1968 年，1969 年のどれかを検索するには，次のように指定します。

```
196[789]
```

リストにない任意の 1 文字と一致

大括弧内の最初の文字がキャレット（^）の場合は，候補文字列に指定された文字以外の文字を表します。たとえば，次のような場合です。

```
[^ab]
```

これは，a と b 以外の文字と一致します。

注： キャレットがこのように解釈されるのは，大括弧内に指定されている文字群の先頭に指定された場合だけです。

範囲内の任意の 1 文字と一致

範囲内の 1 文字を表すには，大括弧（[ ]）とハイフン（-）を使います。たとえば，1960 年代の年を表すには，次のように入力します。

```
196[0-9]
```

ハイフンが括弧内の最初か最後，またはキャレット（^）の後にある場合は範囲の指定とは解釈されません。

たとえば，[-a-z] はハイフンまたは任意の小文字を表します。

注：括弧内では，文字「.」，「*」，「[」および「¥」はリテラルです。たとえばその括弧は，「[.*] 」は「.」または「*」を表します。括弧内の文字群の先頭に閉じ大括弧があると，その括弧はリテラル文字として認識されます。

特定の文字が 0 回以上一致

アスタリスク（*）は直前の文字の 0 回以上の繰り返しと一致します。たとえば，次のような場合です。

```
ca*r
```

これは，「car」，「caaaaaar」，「cr」に一致します。

特定の文字が 1 回以上一致

プラス記号（+）は，直前の文字の 1 回以上の繰り返しと一致します。たとえば，次のような場合です。

```
ca+r
```

これは，「car」，「caaaaaar」と一致しますが，「cr」には一致しません。

特定の文字が 0 回または 1 回一致

疑問符（?）は，直前の文字の 0 回または 1 回の繰り返しと一致します。たとえば，次のような場合です。

```
ca?r
```

これは，「car」と「cr」だけに一致します。

正規表現のグループ化

括弧は（()），数学やプログラミング言語の場合と同様に，括弧内の並びを 1 つの単位として扱うよう指示します。

グループの使用は，選択演算子（|）または反復演算子（*，+，?，{ }）の引数を区切る場合に特に便利です。

正規表現の 1 つと一致

パイプ記号（|）は，表現の選択肢の 1 つと一致します。たとえば，次のような場合です。

```
foo|bar
```

これは，「foo」または「bar」と一致します。

```
fo(o|b)ar
```

これは，「fooar」または「fobar」と一致します。

行の先頭と一致

キャレット（^）は，行の先頭の表現，または改行文字の後の表現を表す場合に指定します。

たとえば，次のような場合です。

```
book
```

これは「book」，「my book」，「book list」のいずれの行の「book」にも一致しますが，

```
^book
```

これは，「book」および「 book list」という行の「book 」とだけ一致します。

行の末尾と一致

ドル記号（$）は，行の末尾の表現，または改行文字の前の表現を表す場合に指定します。たとえば，次のような場合です。

book

これは「my book」および「book list」のどちらの行の「book」にも一致しますが，$ が後に続く文字列は，その文字列で終わる行だけ一致します。たとえば，次のような場合です。

book$

これは，「my book」という行の「book」とだけ一致します。

アンダスコアを含む任意の英数字との一致

「¥w」は，任意の英数字およびアンダスコア（A-Z，a-z，0-9，_）と一致します。

たとえば，次のような場合です。

「¥w*」は英数字（A-Z，a-z，0-9）およびアンダスコア（_）の 0 回以上の繰り返しと一致します。これは，「Ab」，「r9Cj」，「12_uYLgeu_435」などと一致します。

たとえば，次のような場合です。

「¥w{3}」は英数字（A-Z，a-z，0-9，およびアンダスコア（_））の 3 回の繰り返しと一致します。これは，「Ab4」，「r9_」，「z_M」などと一致します。

英数字以外の文字との一致

「¥W」は，英数字とアンダスコア以外の任意の文字と一致します。

たとえば，次のような場合です。

「¥W」は &，*，^，%，$，# と一致します。

正規表現演算子の結合

1 行の中で正規表現の演算子を結合すると，必要となる正確な検索基準を作成できます。

たとえば，「.」と「*」を組み合わせることで，任意の文字の 0 回以上の出現を表せます（「¥n」を除く）。

例を次に示します。

```
start.*
```

これは，「start」，「started」，「starting」，「starter」などと一致します。

また，大括弧とアスタリスクの組み合わせを使って，数値以外の英字の組み合わせを表すこともできます。たとえば，次のような場合です。

```
[a-zA-Z]*
```

0 ～ 1200 の任意の数字と一致するには，1 桁，2 桁，3 桁の数字または 1000 ～ 1200 の 4 けたの数字と一致する必要があります。

次の正規表現は，0 ～ 1200 の任意の数字と一致します。

```
([0-9]?[0-9]?[0-9]|1[01][0-9][0-9]|1200)
```

第 VI 部

設定

第 24 章

グローバル・テスト・オプションの設定

グローバル・テスト・オプションを設定することで，QuickTest とコンポーネントの連携の方法を制御できます。

本章の内容

- グローバル・テスト・オプションの設定について（629 ページ）
- ［オプション］ダイアログ・ボックスの使用（630 ページ）
- テストの一般オプションの設定（632 ページ）
- テストのフォルダ・オプションの設定（635 ページ）
- テストの実行オプションの設定（638 ページ）

グローバル・テスト・オプションの設定について

グローバル・テスト・オプションでは，コンポーネントを使った作業の方法のほか，QuickTest の全般的な表示動作を設定します。たとえば，QuickTest の起動時にスタート・ページを表示しないようにしたり，QuickTest がコンポーネントの実行時に使用するタイミングに関する設定を行ったりできます。設定した値は，すべてのコンポーネントおよび以降のテスト・セッションで有効になります。グローバル・テスト・オプションの設定は，［オプション］ダイアログ・ボックスを使用して（630 ページで説明），またはエキスパート・ビューでステートメントを挿入することによって行います。

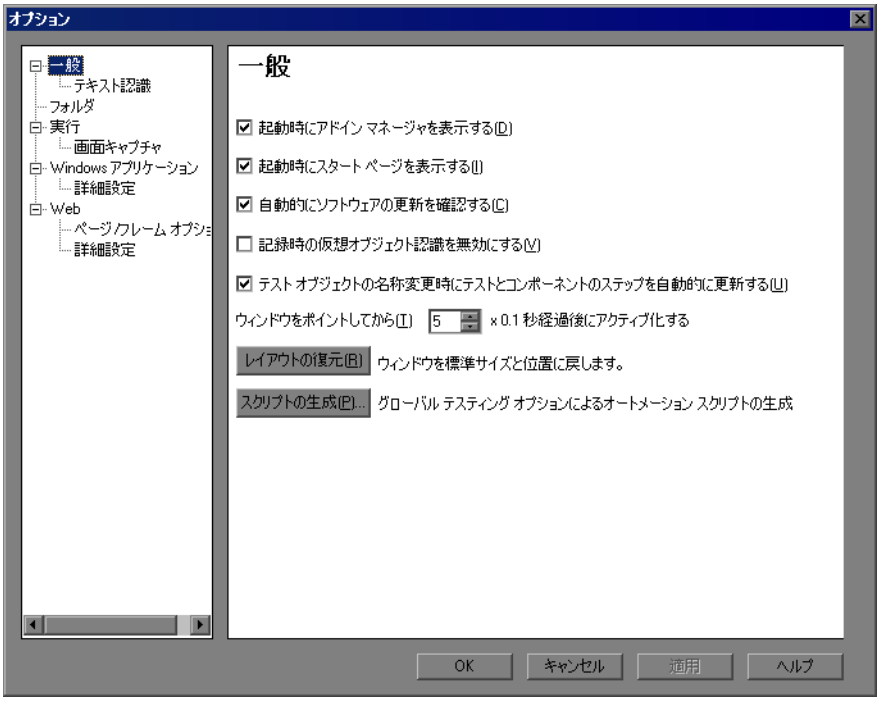
また，QuickTest で現在開いているコンポーネントだけで有効なテスト・オプションを設定することもできます。詳細については，第 25 章「ビジネス・コンポーネント設定にかかわる作業」を参照してください。

[オプション] ダイアログ・ボックスの使用

[オプション] ダイアログ・ボックスを使用してグローバル・テスト・オプションを変更できます。設定した値は，後続のすべての QuickTest セッションで有効になります。

グローバル・テスト・オプションを設定するには，次の手順を実行します。

1 [**ツール**] > [**オプション**] を選択するか，[**オプション**] ツールバー・ボタンをクリックします。[オプション] ダイアログ・ボックスが開きます。このダイアログ・ボックスは 2 つの部分に分かれており，左側はナビゲーション表示枠，右側はオプション表示枠となっています。



2 必要に応じて，ナビゲーション・ツリーからノードを選択し，オプション表示枠でオプションを設定します。各ノードで使用できるオプションの詳細については，次の表を参照してください。

3 変更を適用してダイアログ・ボックスを開いたままにしておくには［**適用**］をクリックします。変更を適用してダイアログ・ボックスを閉じるには［**OK**］をクリックします。

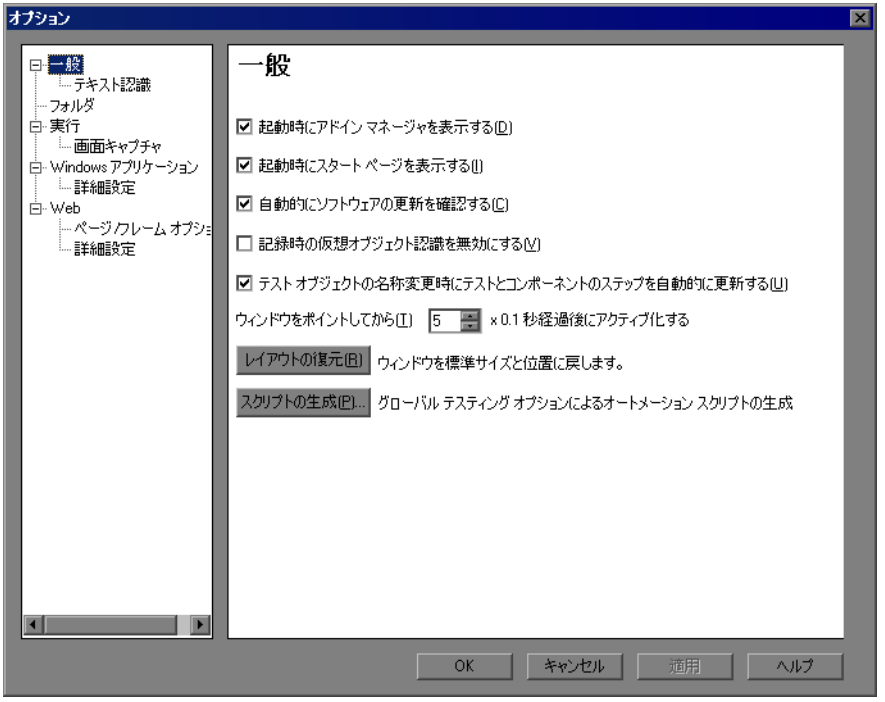
ナビゲーション・ツリーには次のノードが含まれています。

ノード	オプション
［**一般**］	コンポーネントの一般設定のオプション。詳細については，632 ページ「テストの一般オプションの設定」を参照してください。 ［一般］ノードには［テキスト認識］サブノードも含まれています。詳細については，634 ページ「テキスト認識オプションの設定」を参照してください。
［**フォルダ**］	QuickTest が，ダイアログ・ボックスおよびステートメントの中で相対パスを使って指定されているコンポーネントまたはファイルを検索するフォルダ（検索パス）を入力するオプション。コンポーネントおよびアプリケーション領域の場合，すべてのファイルを Quality Center のサブジェクト・パスに格納する必要があります。詳細については，635 ページ「テストのフォルダ・オプションの設定」を参照してください。
［**実行**］	コンポーネントを実行するためのオプション。詳細については，638 ページ「テストの実行オプションの設定」を参照してください。 ［実行］ノードには［画面キャプチャ］ノードも含まれています。詳細については，642 ページ「［オプション］ダイアログ・ボックス：［実行］ > ［画面キャプチャ］表示枠」を参照してください。
［**Windows アプリケーション**］	QuickTest コンポーネントが Windows アプリケーションとともに動作する方法を設定するためのオプション。［Windows アプリケーション］ノードには［詳細設定］ノードも含まれています。詳細については，『**HP QuickTest Professional アドイン・ガイド**』で Windows ベースのアプリケーションのテストに関する項を参照してください。

ナビゲーション・ツリーには，その時点で読み込まれているアドインに応じて対応するノードが表示される場合があります。詳細については，『**HP QuickTest Professional アドイン・ガイド**』の関連する項を参照してください。

テストの一般オプションの設定

［一般］表示枠のオプションでは，QuickTest の全般的な表示動作およびその他の一般的なテスト・オプションを指定します。



［一般］ノードには［テキスト認識］サブノードも含まれています。詳細については，604 ページ「[オプション］ダイアログ・ボックスの［一般］＞［テキスト認識］表示枠」を参照してください。

［一般］表示枠には，次のオプションがあります。

オプション	説明
［起動時にアドイン・マネージャを表示する］	QuickTest を起動したときに，アドイン・マネージャを表示するかどうかを指定します。アドイン・マネージャの使用方法の詳細については，『**HP QuickTest Professional アドイン・ガイド**』で QuickTest アドインの読み込みに関する項を参照してください。
［起動時にスタートページを表示する］	QuickTest を起動したときに，スタート・ページを表示するかどうかを指定します。
［自動的にソフトウェアの更新を確認する］	ソフトウェアの更新が自動的に確認されるように QuickTest を設定します。詳細については，40 ページ「QuickTest ソフトウェアの更新」を参照してください。
［起動時の仮想オブジェクト認識を無効にする］	記録時に，仮想オブジェクト・マネージャに格納されている定義済みの仮想オブジェクトを認識するかどうかを指定します。このオプションはテストにのみ関係しています。
［テスト オブジェクトの名称変更時にテストとコンポーネントのステップを自動的に更新する］	ローカルまたは共有オブジェクト・リポジトリ内のテスト・オブジェクトの名前を変更した場合，テストおよびコンポーネントのステップを自動的に更新するかどうかを指定します。詳細については，182 ページ「テスト・オブジェクトの名前の変更」を参照してください。
［ウィンドウをポイントしてから X x 0.1 秒経過後にアクティブ化する］	QuickTest が，（オブジェクト・スパイ，回復シナリオ・ウィザードなどのために）アプリケーション内のオブジェクトを指すのに指差しマークを使っているとき，アプリケーション・ウィンドウをアクティブにするまで待機する時間を 10 分の 1 秒単位で指定します。**標準設定 = 5**

オプション	説明
［レイアウトの復元］	表示枠およびツールバーが標準のサイズで標準の位置に表示されるように，QuickTest ウィンドウの配置を復元します。 **注**：QuickTest は，動作モード（表示 / 編集，記録，および実行）ごとに最新のウィンドウ・レイアウトを復元します。詳細については，810 ページ「QuickTest ウィンドウ・レイアウトのカスタマイズ」を参照してください。
［スクリプトの生成］	現在のグローバル・テスト・オプションを含んだ自動化スクリプトを生成します。詳細については，911 ページ「QuickTest 操作のオートメーション」，または『**QuickTest Professional Automation Object Model Reference**』（英語版）（［**ヘルプ**］＞［**QuickTest Professional ヘルプ**］＞［**HP QuickTest Professional Advanced References**］＞［**HP QuickTest Professional Automation Object Model**］）を参照してください。

テキスト認識オプションの設定

ナビゲーション・ツリーの［テキスト認識］ノードでは，［一般］＞［テキスト認識］表示枠が表示されます。ここでは，アプリケーションのテキストを識別する方法を設定できます。詳細については，604 ページ「[オプション］ダイアログ・ボックスの［一般］＞［テキスト認識］表示枠」を参照してください。

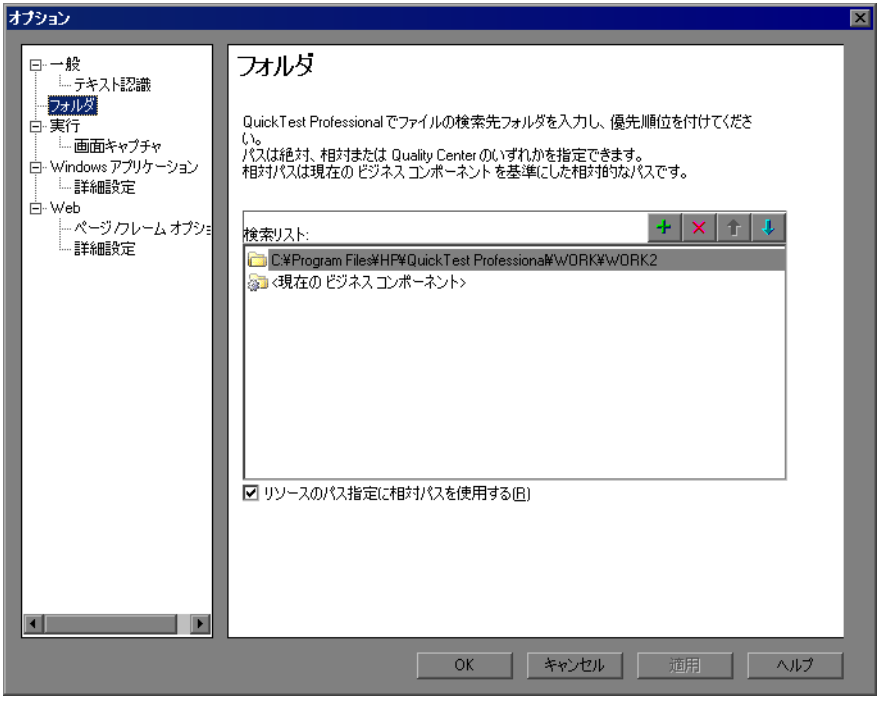
テストのフォルダ・オプションの設定

［フォルダ］表示枠には，ダイアログ・ボックスおよびステップの中で相対パスを使って指定されているコンポーネントリソース・ファイルを QuickTest が検索するフォルダ（検索パス）を入力できます。すべてのファイルを Quality Center パスに格納する必要があります。

注：

- 標準設定では現在のコンポーネントが**検索リスト**に表示されます。これは削除できません。
- 相対パスまたは絶対パスの詳細については，『**HP QuickTest Professional ユーザーズ・ガイド**』で相対パスの使用方法に関する項を参照してください。

QuickTest は，フォルダが検索リストに表示されている順に，指定されたコンポーネントまたはファイルを検索します。同じファイル名が複数のフォルダに存在する場合，QuickTest は最初に検出したファイルを使用します。

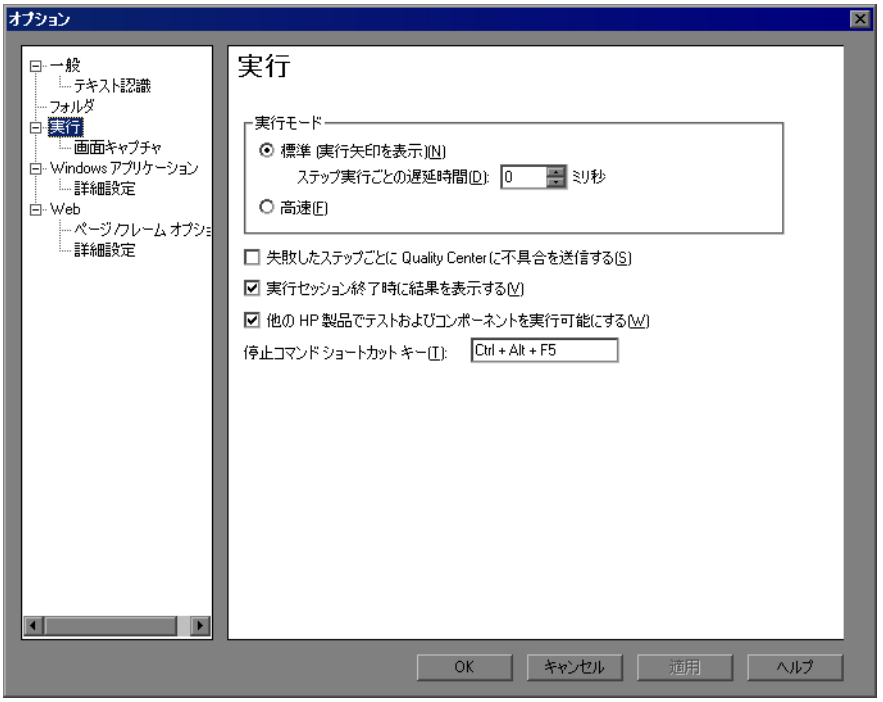


［フォルダ］表示枠には，次のオプションがあります。

オプション	説明
［検索リスト］	QuickTest がコンポーネントまたはファイルを検索するフォルダを示します。ここでフォルダを定義しておけば，ほかのダイアログ・ボックスの中でコンポーネントまたはファイルのフルパスを指定する必要がなくなります。リスト内の検索パスの順序に応じて，QuickTest が特定のファイルを検索する順序が決まります。
	新しいフォルダを検索リストに追加します。 **ヒント**： ➤ Quality Center に接続されている状態で Quality Center パスを追加するにはこのボタンをクリックします。QuickTest に [QualityCenter] が追加され，Quality Center のパスを指定するための参照ボタンが表示されます。 ➤ Quality Center に接続されていない場合は SHIFT キーを押しながらこのボタンをクリックします。QuickTest に [QualityCenter] が追加されます。ここでパスを入力します。Quality Center のパス全体を手作業で入力することもできます。入力する場合は [QualityCenter] の後にスペースを追加する必要があります。たとえば，[QualityCenter] Subject¥Tests のようにします。 ➤ ただし，QuickTest が Quality Center のプロジェクト・フォルダを検索するのは，対応する Quality Center プロジェクトに接続しているときだけです。
	選択したフォルダを検索リストから削除します。
	選択したフォルダを上方向に移動します。
	選択したフォルダを下方向に移動します。
［リソースのパス指定に相対パスを使用するように促す］	リソースを保存するときに，相対パスを使用することを促すメッセージを表示できます。詳細については，**『HP QuickTest Professional ユーザーズ・ガイド』**の相対パスの使用方法に関する項を参照してください。 **注**：QuickTest が Quality Center 10.00 プロジェクトに接続されている場合は，ファイル・システムまたは Quality Center 9.x プロジェクトのパスを選択したときにのみ，リマインダが表示されます。

テストの実行オプションの設定

［実行］表示枠のオプションでは，QuickTest がコンポーネントを実行する方法を指定します。



［実行］ノードには［画面キャプチャ］ノードも含まれています。詳細については，642 ページ「［オプション］ダイアログ・ボックス：［実行］ > ［画面キャプチャ］表示枠」を参照してください。

［実行］表示枠には，次のオプションがあります。

オプション	説明
［実行モード］	QuickTest に対してコンポーネントの実行方法を指定します。 ➤［**標準（実行矢印を表示）**］：キーワード・ビューの左に実行矢印を表示しながらコンポーネントを実行し，実行した各ステップがわかるようにします。 ［**ステップ実行ごとの遅延時間**］：連続する各ステップを実行するまでに QuickTest が待機する時間をミリ秒単位で指定できます（最大 10000 ミリ秒まで）。 次の説明のとおり，［**標準**］実行モード・オプションには，［**高速**］オプションよりも多くのシステム・リソースが必要です。 **注**：このモードを有効にするには，Microsoft Script Debugger をインストールしておく必要があります。詳細については，『**HP QuickTest Professional インストール・ガイド**』を参照してください。 ➤［**高速**］：キーワード・ビューの左に実行矢印を表示せずにコンポーネントを実行します。このオプションでは，必要とされるシステム・リソースが少なくて済みます。 **注**：Quality Center からテスト・セットを実行する場合，**標準**モードが選択されていても，コンポーネントは自動的に**高速**モードで実行されます。
［失敗したステップごとに Quality Center に不具合を送信する］	テストの場合のみ。。
［実行セッション終了時に結果を表示する］	実行セッションの後に結果を自動的に表示するように QuickTest を設定します。

オプション	説明
［他の HP 製品でテストおよびコンポーネントを実行可能にする］	このコンピュータ上で，Quality Center などのほかの HP 製品で QuickTest のコンポーネントを実行できるようになります。 **注**： ➤ このオプションは WinRunner で QuickTest のコンポーネントを実行する場合には必要ありません。 ➤ お使いのコンピュータで **Windows ファイアウォール**が**有効**になっていて，リモートの Quality Center クライアントからお使いのコンピュータでビジネス・プロセス・テストを実行できるようにする場合は，**bp_exec_agent.exe** リモート・エージェントのファイアウォール例外を手動で作成する必要もあります。詳細については，641 ページ「Business Process Testing リモート・エージェントの有効化」を参照してください。
［停止コマンドショートカットキー］	QuickTest にフォーカスがない場合や QuickTest が非表示モードの場合でも，現在の QuickTest の記録操作あるいは実行操作を停止するショートカット・キーまたはキーの組み合わせを定義できます。 フィールドをクリックし，キーボードで，必要なキーまたはキーの組み合わせを押します。 標準のキーの組み合わせは **Ctrl+Alt+F5** です。 **注** ：テスト対象アプリケーションでほかの操作のためにまだ定義されていないショートカット・キーを定義する必要があります。すでに定義されていた場合は，次のようになります。 ➤［**記録**］または［**実行**］をクリックする前に手動でアプリケーションを開いた場合は，アプリケーションに定義されているショートカット・キーがその本来の目的のために適用されます。 ➤ 記録セッションまたは実行セッションを開始し，QuickTest によってアプリケーションが開かれた場合は，［実行］表示枠で定義したショートカット・キーでセッションが停止します。

Business Process Testing リモート・エージェントの有効化

お使いのコンピュータで **Windows ファイアウォール**が**有効**になっていて，リモートの Quality Center クライアントからお使いのコンピュータでビジネス・プロセス・テストを実行できるようにする場合は，**bp_exec_agent.exe** リモート・エージェントのファイアウォール例外を手動で作成する必要があります。

Business Process Testing リモート・エージェントのファイアウォール例外を作成するには，次の手順を実行します。

1 お使いのコンピュータで少なくとも 1 回は Quality Center クライアントを開いていることを確認します。

2 コマンド・ラインから **Firewall.cpl** を実行します。［Windows Firewall］ダイアログ・ボックスが開きます。

注：以降の手順は，オペレーティング・システムによって若干異なる場合があります。

---

3 ［**Exceptions**］タブをクリックします。

4 ［Add Program］ボタンをクリックします。

5 ［Add a Program］ダイアログ・ボックスで，Quality Center クライアントがインストールされている場所を参照し，**bp_exec_agent.exe** を選択します。標準設定では，**C:¥Program Files¥Common Files¥Mercury Interactive¥Quality Center** にあります。

6 参照ダイアログ・ボックスの［**OK**］をクリックし，［Add a Program］ダイアログ・ボックスの［**OK**］もクリックします。［Windows Firewall］ダイアログ・ボックスの［Programs and Services］リストに **bp_exec_agent.exe** 項目が追加されます。

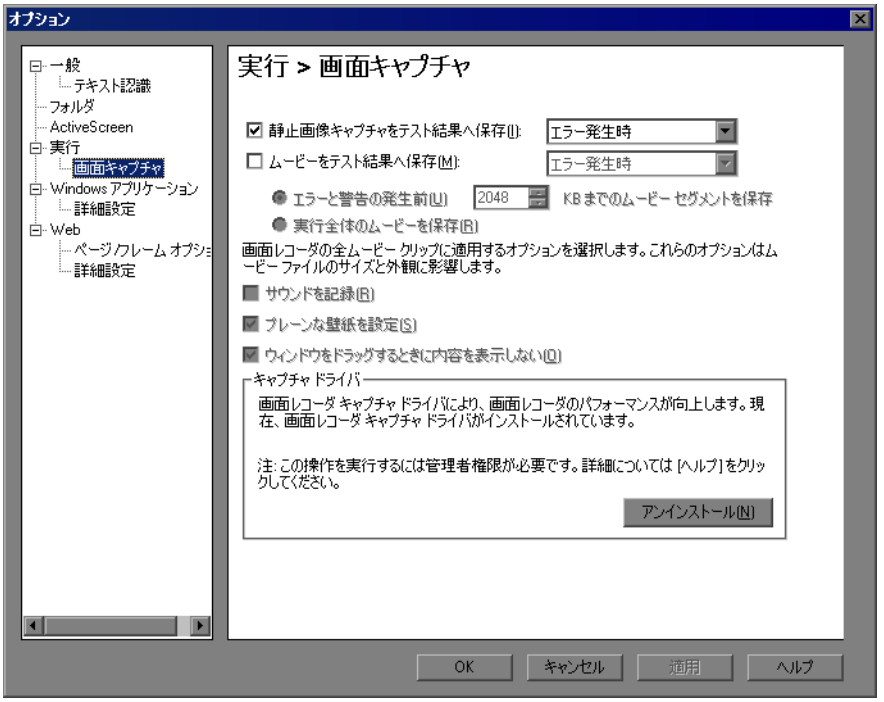
7 ［**OK**］をクリックし，［Windows Firewall］ダイアログ・ボックスを閉じます。

Windows ファイアウォール例外の詳細については，Windows のドキュメントを参照するか，組織のシステム管理者にお問い合わせください。

［オプション］ダイアログ・ボックス：［実行］＞［画面キャプチャ］表示枠

説明	テスト対象アプリケーションの画面を QuickTest がキャプチャするタイミングと方法を制御できます。
アクセス方法	➤［**ツール**］＞［**オプション**］メニュー・コマンドを選択し，［実行］＞［画面キャプチャ］ノードを選択します。 ➤［**オプション**］ツールバー・ボタン をクリックし，［実行］＞［画面キャプチャ］ノードを選択します。
重要情報	**Vista ユーザの方へ** ：下記のオプションに加えて，Windows Vista の配色テーマが **Aero** に設定されている場合は，パフォーマンスを最大限に引き出すために，実行セッションのムービーをキャプチャしている間は，QuickTest によってテーマが自動的に **Vista Basic** に設定されます。実行セッションが終了すると，配色テーマは元の設定に戻されます。
詳細情報	**その他の関連トピック**：646 ページ「補足リファレンス」

［オプション］ダイアログ・ボックスの［実行］＞［画面キャプチャ］表示枠の画像を次に示します。



［オプション］ダイアログ・ボックス：［実行］＞［画面キャプチャ］表示枠のオプション

オプション	説明
［**静止画像キャプチャをテスト結果へ保存**］	実行セッション中にアプリケーションの静止画像をキャプチャし，テスト結果に保存するタイミングを QuickTest に指定します。テスト結果の画像が使用可能なら，［テスト結果］ウィンドウの［結果の詳細］タブの下の表示枠にその画像が表示されます。 チェック・ボックスをクリアしてこのオプションを無効にするか，リストからオプションを選択します。 ➤［**毎回**］：実行時にすべてのステップの画像をキャプチャします。 ➤［**エラー発生時**］：失敗したステップの画像のみをキャプチャします。これが標準設定です。 ➤［**エラーおよび警告発生時**］：失敗または警告を返したステップのみ画像をキャプチャします。 詳細については，699 ページ「アプリケーションの静止画像とムービーの表示」を参照してください。 **注**：この設定は，テスト結果の詳細の下の表示枠に表示される，次のようなその他の情報の可用性にも影響します。 ➤ XML チェックポイントおよび XML 出力値の結果の詳細 ➤ ビットマップ・チェックポイントの画像（期待，実際，差異）

オプション	説明
［**ムービーをテスト結果へ保存**］	実行セッション中にアプリケーションのムービーをキャプチャし，実行結果に保存するタイミングを QuickTest に指定します。実行結果のムービーが使用可能なら，［テスト結果］ウィンドウの［画面レコーダ］タブにそのムービーが表示されます。標準設定ではこのオプションは無効です。 このオプションを有効にするには，チェック・ボックスを選択し，リストからオプションを選択します。 ➤［**毎回**］：実行時にすべてのステップのムービーをキャプチャします。 ➤［**エラー発生時**］：失敗したステップのムービーのみをキャプチャします。 ➤［**エラーおよび警告発生時**］：失敗または警告を返したステップのみムービーをキャプチャします。 詳細については，699 ページ「アプリケーションの静止画像とムービーの表示」を参照してください。
次のオプションは，［**ムービーをテスト結果へ保存**］チェック・ボックスが選択されている場合にかぎり有効になります。	
［**エラーと警告の発生前 __ KB までのムービーセグメントを保存**］ （［**エラー発生時**］または［**エラーおよび警告発生時**］が［**ムービーをテスト結果へ保存**］オプションで選択されている場合のみ有効）	選択すると，QuickTest によってエラー（または警告）ごとにムービー・セグメントが保存されます。各セグメントには，失敗したステップ（または警告）までの指定したキロバイト数のムービーが含まれます。400（0.4MB）から 2097152（2GB）までの任意の値を入力できます。実行セッションで複数のセグメントがキャプチャされた場合は，QuickTest によって，すべての関連するムービー・セグメントで構成される単一のムービーが保存されます。
［**実行全体のムービーを保存する**］ （［**エラー発生時**］または［**エラーおよび警告発生時**］が［**ムービーをテスト結果へ保存**］オプションで選択されている場合のみ有効）	選択すると，少なくとも 1 つのエラー（または警告）が発生したら，QuickTest によって，実行全体のムービーが保存されます。
［**サウンドを記録**］	アプリケーションのムービーとともにサウンドを保存するように QuickTest を設定します。

オプション	説明
［**プレーンな壁紙を設定**］	実行セッション中のデスクトップの壁紙を青一色に設定します。
［**ウィンドウをドラッグするときに内容を表示しない**］	実行セッション中にウィンドウがドラッグされたときに，ウィンドウの輪郭だけを表示して，内容は表示しないように Windows を設定します。
［**キャプチャ ドライバ**］領域	
［**インストール**］/ ［**アンインストール**］ボタン	画面レコーダ・キャプチャ・ドライバをインストールまたはアンインストールします。画面レコーダ・キャプチャ・ドライバは，ムービー記録中の画面レコーダのパフォーマンスを向上させます。 **注**：リモート接続を介して QuickTest を実行する場合は，画面レコーダ・キャプチャ・ドライバをインストールまたはアンインストールすることはできません。

補足リファレンス

関連ユーザ・インタフェース・トピック	699 ページ「アプリケーションの静止画像とムービーの表示」

第 25 章

ビジネス・コンポーネント設定にかかわる作業

ビジネス・コンポーネントの作成またはデバッグに先立ち，コンポーネントに関連付けられているアプリケーション領域にすでに定義されているコンポーネントの設定を［ビジネス コンポーネントの設定］ダイアログ・ボックスに表示できます。また，［ビジネス コンポーネントの設定］ダイアログ・ボックスでは，コンポーネントに対していくつかの追加設定も定義できます。

本章の内容

- ビジネス・コンポーネント設定にかかわる作業について（648 ページ）
- ［ビジネス コンポーネントの設定］ダイアログ・ボックスの使用（648 ページ）
- コンポーネント・プロパティにかかわる作業（651 ページ）
- コンポーネントのスナップショットの定義（655 ページ）
- アプリケーション設定の表示（657 ページ）
- コンポーネント・リソースの表示（659 ページ）
- コンポーネントのパラメータの定義（660 ページ）
- 回復シナリオの設定の表示（665 ページ）

注：アプリケーション領域でのコンポーネント設定の定義の詳細については，第 第 13 章 章「アプリケーション領域の管理」を参照してください。

ビジネス・コンポーネント設定にかかわる作業について

新しいアプリケーション領域を作成するときは，新しいビジネス・コンポーネントの作成に必要な設定とリソースを定義します。設定には，関連アドイン，コンポーネントが記録と実行の対象にできる Windows ベースのアプリケーション，およびコンポーネントと一緒に使用する関数ライブラリや共有オブジェクト・リポジトリの場所が含まれます。

QTP エンジニア（または各分野のエキスパート）が新しいコンポーネントを作成すると，当該コンポーネントに関連付けられているアプリケーション領域に定義されている設定が自動的に関連付けられます。［ビジネス コンポーネントの設定］ダイアログ・ボックスには，これらの設定が読み取り専用形式で表示されます。

［ビジネス コンポーネントの設定］ダイアログ・ボックスでは，入力パラメータや出力パラメータ，コンポーネントのステータスなどのいくつかの追加設定を定義できます。

注：すべてのコンポーネントに影響を与えるテスト・オプションも設定できます。詳細については，第 第 24 章 章「グローバル・テスト・オプションの設定」を参照してください。

［ビジネス コンポーネントの設定］ダイアログ・ボックスの使用

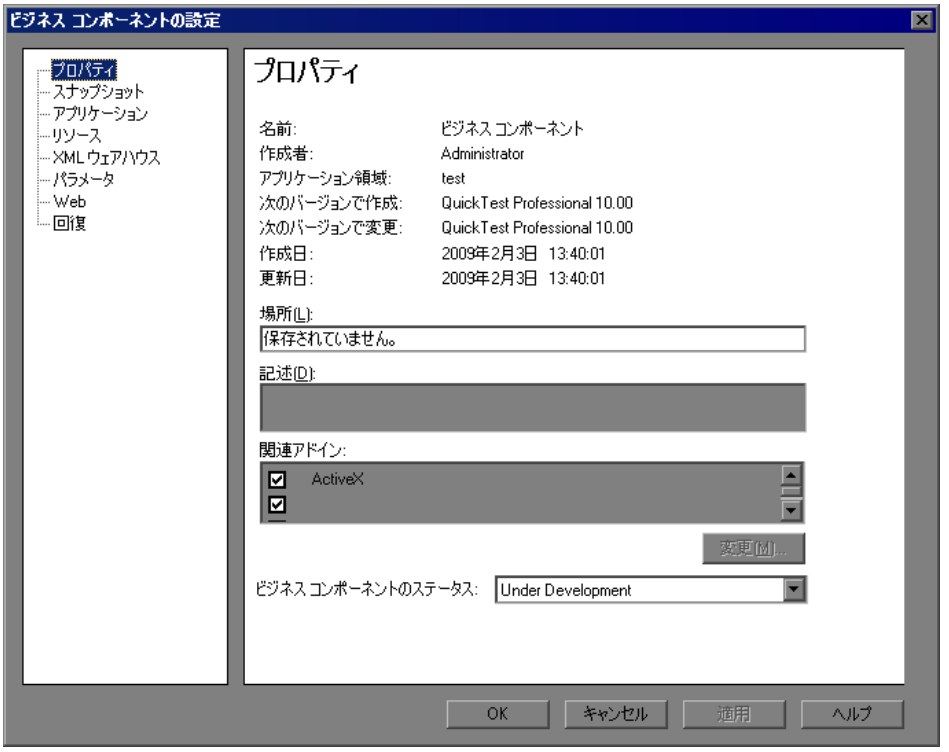
［ビジネス コンポーネントの設定］ダイアログ・ボックスでは，設定の表示やコンポーネントに対する個別のオプションの定義ができます。

［ビジネス コンポーネントの設定］ダイアログ・ボックスを開くには，次の手順を実行します。

1 設定を表示または定義する対象となるコンポーネントを開きます。

2 ［**ファイル**］＞［**設定**］を選択するか，［**テストの設定**］ツールバー・ボタンをクリックします。［ビジネス コンポーネントの設定］ダイアログ・ボックスが

開きます。ダイアログ・ボックスは 2 つの部分に分かれています。左側のナビゲーション表示枠と右側の設定の表示枠です。



3 必要に応じて，オプションの表示または設定に使用するノードを選択します。各表示枠で使用できる設定およびオプションの詳細については，次の表を参照してください。

4 変更を適用してダイアログ・ボックスを開いたままにしておくには［**適用**］をクリックします。変更を保存してダイアログ・ボックスを閉じるには［**OK**］をクリックします。

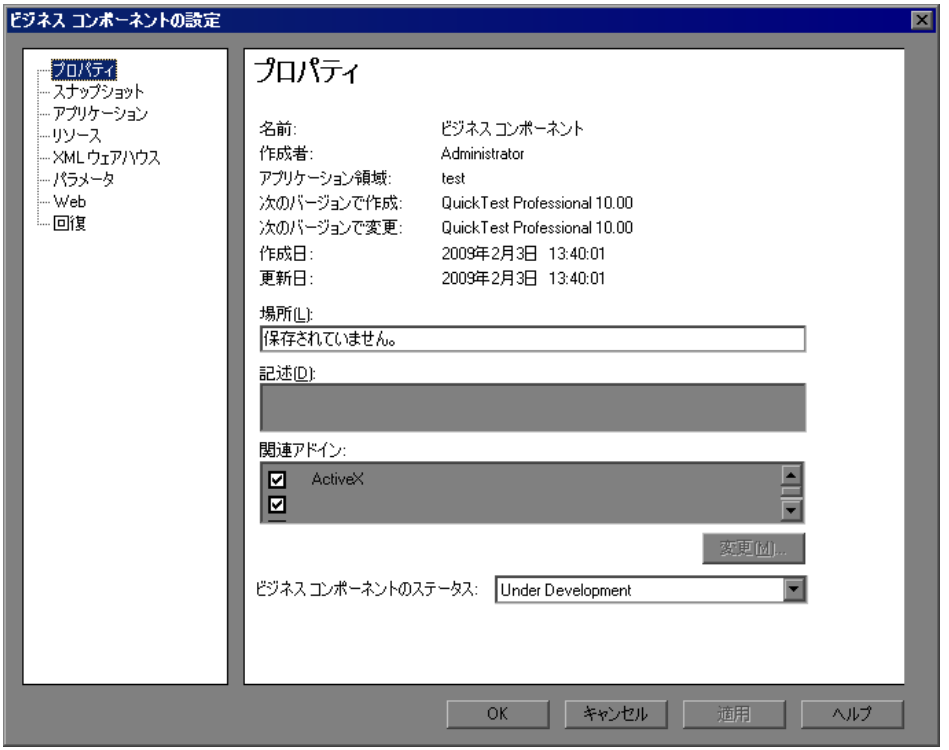
［ビジネス コンポーネントの設定］ダイアログ・ボックスには，次の表示枠があります。

表示枠の見出し	表示枠の内容
［プロパティ］	ビジネス・コンポーネントの説明や関連アドインなど，ビジネス・コンポーネントのプロパティ。コンポーネントのステータスも設定できます。詳細については，651 ページ「コンポーネント・プロパティにかかわる作業」を参照してください。
［スナップショット］	Quality Center に表示するためにコンポーネントに保存するスナップショット画像をキャプチャまたはロードするためのオプション。詳細については，655 ページ「コンポーネントのスナップショットの定義」を参照してください。
［アプリケーション］	コンポーネントが記録と実行の対象にできる Windows ベース・アプリケーション。詳細については，657 ページ「アプリケーション設定の表示」を参照してください。
［リソース］	関数ライブラリや共有オブジェクト・リポジトリの場所を含め，コンポーネントに関連付けられているリソース。詳細については，659 ページ「コンポーネント・リソースの表示」を参照してください。
［パラメータ］	コンポーネントの入力パラメータおよび出力パラメータを指定するためのオプション。詳細については，660 ページ「コンポーネントのパラメータの定義」を参照してください。
［回復］	実行セッション中に，テスト環境で生じる予期しないイベントやエラーからコンポーネントが回復する方法。詳細については，665 ページ「回復シナリオの設定の表示」を参照してください。

これらの表示枠のほかに，［ビジネス コンポーネントの設定］ダイアログ・ボックスには，スクリプト・コンポーネント用の表示枠が含まれる場合があります。これらの表示枠の詳細については，**『HP QuickTest Professional ユーザーズ・ガイド』**を参照してください。また，現在ロードされているアドイン（SAP や Web サービスなど）によっては，ほかの表示枠が含まれる場合もあります。アドインに関連するタブの詳細については，**『HP QuickTest Professional アドイン・ガイド』**の対応する項を参照してください。

コンポーネント・プロパティにかかわる作業

［ビジネス コンポーネントの設定］ダイアログ・ボックスの［プロパティ］表示枠を使用して，コンポーネントの説明やコンポーネントに関連付けられているアドインなど，コンポーネントに関する一般情報を表示できます。また，コンポーネントのステータスの設定や変更もできます。



［プロパティ］表示枠には，次の項目があります。

設定	説明
［名前］	コンポーネントの名前を示します。コンポーネントの名前は，保存するときに指定します。Quality Center のバージョン管理されたプロジェクトにコンポーネントを保存した場合，バージョン番号も表示されます。
［作成者］	コンポーネントを作成した人物の Windows ユーザ名を示します。
［アプリケーション領域］	コンポーネントに関連付けられているアプリケーション領域の名前を示します。詳細については，483 ページ「新規ビジネス・コンポーネントの作成」を参照してください。 **注**：Quality Center でコンポーネントが作成され，アプリケーション領域が選択されなかった場合には，［**未選択**］と表示されます。ビジネス・コンポーネント・ステップが実装できるようになる前に，アプリケーション領域を選択する必要があります。
［次のバージョンで作成］	コンポーネントの作成に使用した QuickTest のバージョンを示します。
［次のバージョンで変更］	最後にコンポーネントに変更を加えた QuickTest のバージョンを示します。
［作成日］	コンポーネントが作成された日時を示します。
［更新日］	コンポーネントが最後に変更された日時を示します。
［場所］	コンポーネントの Quality Center パスおよびファイル名を示します。 **注**：コンポーネントがまだ保存されていない場合は，「**保存されていません**」と表示されます。
［記述］	コンポーネントに対して指定されている詳細説明が表示されます。このフィールドの入力または変更は，Quality Center でのみ行うことができます。

設定	説明
［関連アドイン］	アプリケーション領域を通じてコンポーネントに関連付けられているアドインが表示されます。関連アドインは，ビジネス・コンポーネントへのアクセスがあったときに当該コンポーネントによってロードされます。
［ビジネス コンポーネントのステータス］	コンポーネントのステータスを示します。リストから別のオプションを選択することでコンポーネントのステータスを変更できます。ステータスのオプションの詳細については，653 ページ「コンポーネントのステータスについて」を参照してください。

コンポーネントの基礎になっているアプリケーション領域の一般情報の定義方法の詳細については，446 ページ「一般設定の定義」を参照してください。

コンポーネントのステータスについて

ビジネス・コンポーネントには，QuickTest または Quality Center でステータスを割り当てることができます。ビジネス・コンポーネントのステータスは手作業で指定できますが，場合によっては Quality Center によって自動的に割り当てられます。たとえば，ビジネス・プロセス・テスト内で実行する準備ができたビジネス・コンポーネントに ［**準備完了**］ステータスを手作業で割り当てることもできれば，ビジネス・プロセス・テスト内での正常な実行を妨げるエラーを含んだビジネス・コンポーネントに ［**エラー**］ステータスが自動的に割り当てられることもあります。

ビジネス・コンポーネントのステータスは，当該コンポーネントを含んでいるビジネス・プロセス・テストのステータスに影響するので，ビジネス・コンポーネント全体のステータスを知ることは重要です。一般に，ビジネス・プロセス・テストのステータスは，最も重大なステータスを持つコンポーネントによって決まります。たとえば，［**エラー**］ステータスを持つコンポーネントを含むビジネス・プロセス・テストのステータスは［**エラー**］になります。

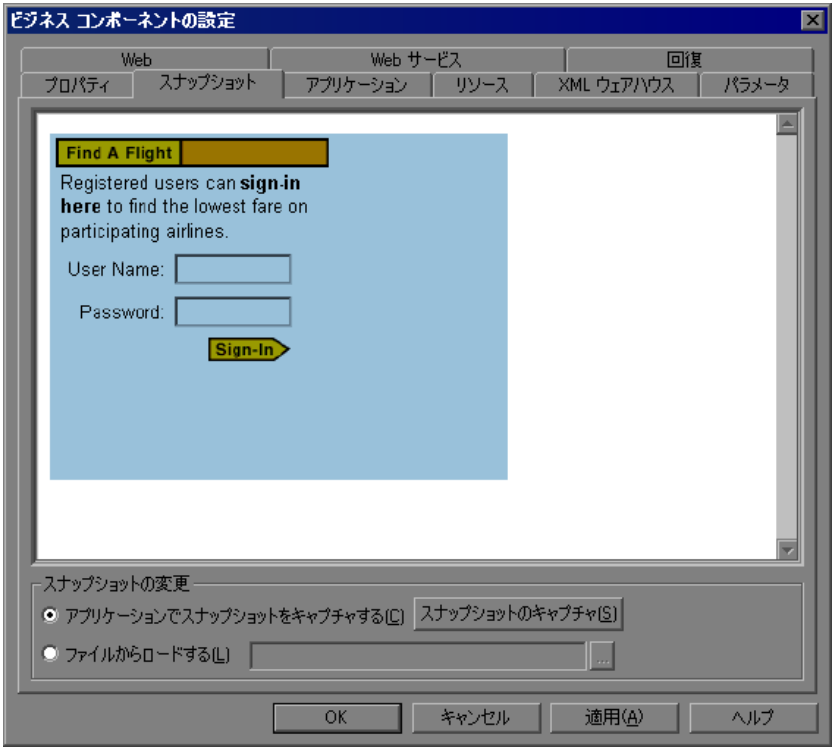
コンポーネントのステータスは次のいずれかです。

➤ ［**エラー**］修正が必要なエラーがコンポーネントに含まれています。たとえば，アプリケーションの変更が原因で起こる場合があります。このステータスのコンポーネントがビジネス・プロセス・テストに含まれていると，ビジネス・プロセス・テスト全体のステータスも［**エラー**］となります。

- ［**メンテナンス**］コンポーネントは，現在開発およびテストが行われていて実行の準備が整っていないか，以前に実装され現在はアプリケーションに加えられた変更に合わせるための変更が行われています。
- ［**準備完了**］対象コンポーネントは実装が完全に済んでおり，実行の準備が整っています。コンポーネントは指定された要件を満たしており，対象システムについて定義されている基準に基づいたテストが済んでいます。
- ［**開発中**］コンポーネントは現在開発中です。このステータスは自動的に次の項目に割り当てられます。
  - Business Process Testing サポート付きの Quality Center のビジネス・コンポーネント・モジュールを使用して作成された新規コンポーネント。
  - Business Process Testing サポート付きの Quality Center のコンポーネント・ツリーにドラッグして入れられたコンポーネント要求。
- ［**未実装**］対象コンポーネントは Quality Center のテスト計画モジュールで要求されました。ビジネス・コンポーネント・モジュールの［コンポーネント要求］フォルダからコンポーネント・ツリーにコンポーネント要求を移動すると，ステータスは自動的に［**未実装**］から［**開発中**］に変わります。

コンポーネントのスナップショットの定義

［ビジネス コンポーネントの設定］ダイアログ・ボックスの［スナップショット］表示枠は，画像をキャプチャまたはロードしてコンポーネントに保存することを可能にします。画像はコンポーネントの主要な目的を視覚的に示します。各分野のエキスパートは，Quality Center，コンポーネント，およびコンポーネントが含まれている任意のビジネス・プロセス・テストの画像を表示できます。



注：スナップショット画像は，Business Process Testing サポート機能がインストールされている場合は Quality Center の［スナップショット］タブでもキャプチャしてコンポーネントに保存できます。Quality Center のコンポーネントのためのスナップショット取得に関する詳細は，『**HP Business Process Testing ユーザーズ・ガイド**』を参照してください。

［スナップショット］表示枠には，次のオプションがあります。

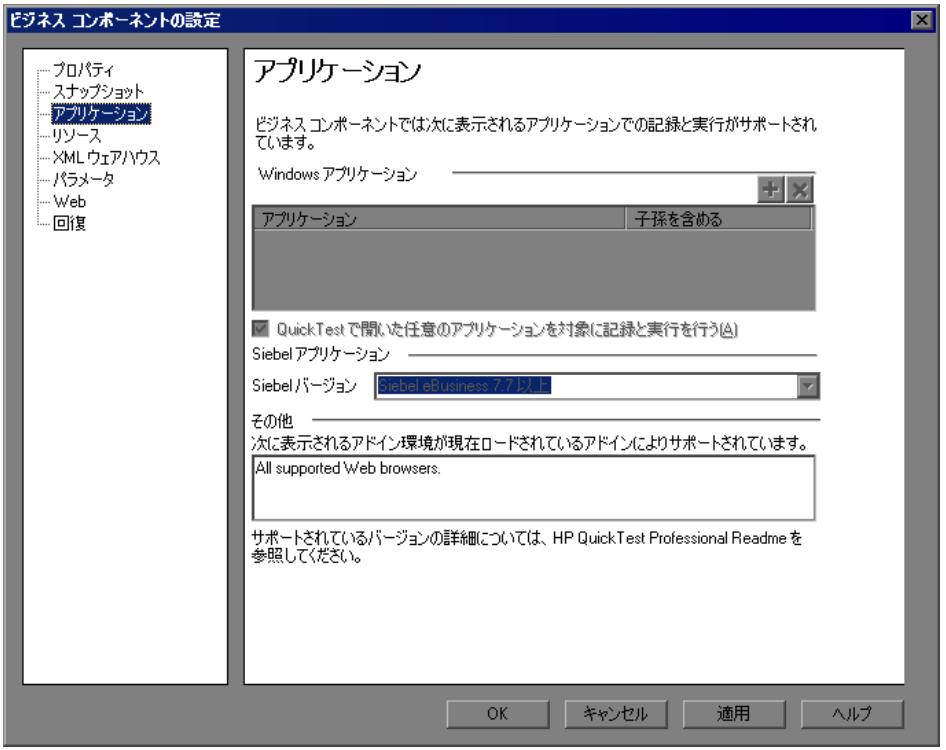
オプション	説明
［アプリケーションでスナップショットをキャプチャする］	［**スナップショットのキャプチャ**］ボタンをクリックしてキャプチャする画像を定義することを可能にします。十字ポインタをドラッグしてキャプチャ対象の領域を選択できます。マウス・ボタンを放すと，キャプチャされた領域がスナップショット表示枠に表示されます。
［ファイルからロードする］	必要な画像が格納されている **.png** ファイルまたは **.bmp** ファイルを指定します。パスおよびファイル名を入力するか，参照ボタンを使用してファイルを指定します。

［**適用**］または［**OK**］をクリックすると，画像がコンポーネントに保存され，Quality Center においてこのコンポーネントを含んでいるビジネス・プロセス・テスト内に表示されます。

アプリケーション設定の表示

［ビジネス コンポーネントの設定］ダイアログ・ボックスの［アプリケーション］表示枠には，コンポーネントが（現在のアプリケーション領域に基づいて）記録と実行の対象にできる Windows ベース・アプリケーションの読み取り専用リストが表示されます。指定したアプリケーションが対象の場合にのみステップを記録できます。

（現在ロードされているアドインに基づいて）コンポーネントが現在記録の対象にできる環境も表示されます。



関連付けられているアプリケーション領域設定で，コンポーネントが記録と実行の対象にできる Windows ベースのアプリケーションを指定します。詳細については，453 ページ「アプリケーション領域のアプリケーション設定の定義」を参照してください。

注 ÅF

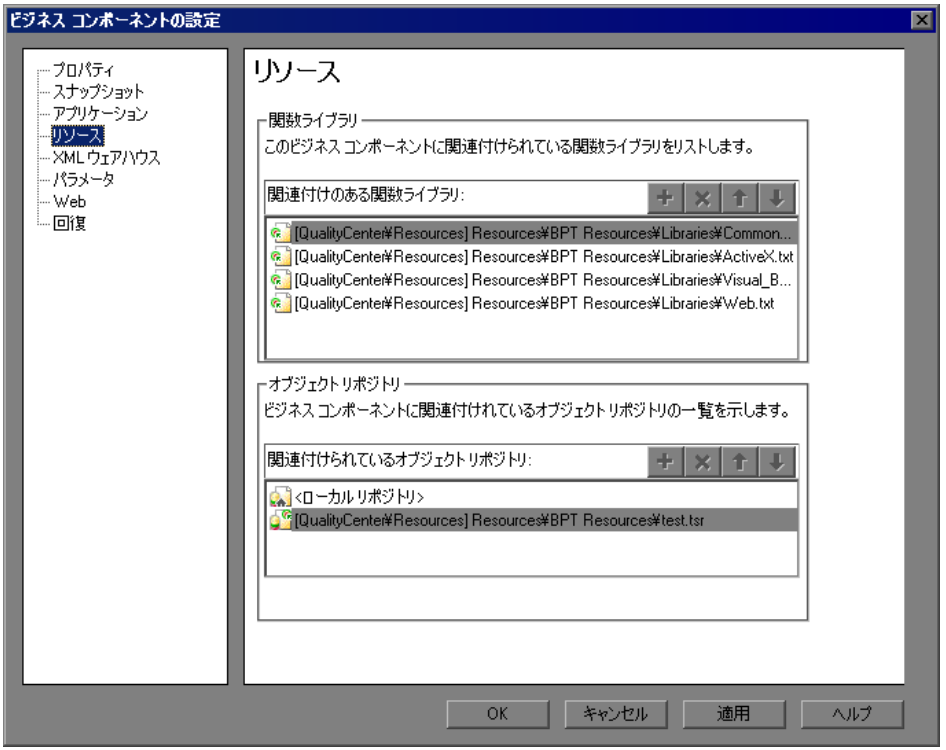
- ➤ 新しいコンポーネントを記録するときに［アプリケーション領域の設定］ダイアログ・ボックスの［アプリケーション］表示枠でアプリケーションの設定がまだ行われていなかった場合は，記録の開始時に［アプリケーション］ダイアログ・ボックスが開きます。［アプリケーション］ダイアログ・ボックスには，［アプリケーション］タブと同じオプションがあります。詳細は，453 ページ「アプリケーション領域のアプリケーション設定の定義」を参照してください。
- ➤ ［アプリケーション］ダイアログ・ボックスと［アプリケーション］表示枠には，コンピュータにインストールされている任意の QuickTest アドインに適用されるオプションが含まれている場合もあります。これらのオプションの詳細については，アドインに付属のドキュメントを参照してください。

［アプリケーション］表示枠には，次の項目があります。

設定	説明
［アプリケーション］	コンポーネントの記録と実行を行う対象となるアプリケーションの詳細が列挙されます。 Windows アプリケーションを対象に記録と実行を行わない場合には，アプリケーション・リストを空のままにしておきます（これが標準の設定です）。
QuickTest で開いた任意のアプリケーションを対象に記録と実行を行う	QuickTest によって起動される任意のアプリケーションを（QuickTest の子プロセスとして）記録と実行の対象にします。たとえば，セッションの記録または実行中に OpenApp 関数を使用して開かれるアプリケーションや，アプリケーションを開く関数が含まれている別の操作によって開かれたアプリケーションなどが対象になります。
［その他］	現在ロードされているアドインに対応するアドイン環境を一覧表示します。

コンポーネント・リソースの表示

［ビジネス コンポーネントの設定］ダイアログ・ボックスの［リソース］表示枠には，（コンポーネントに関連付けられているアプリケーション領域を通じて）コンポーネントに関連付けられている関数ライブラリおよびオブジェクト・リポジトリの読み取り専用リストが表示されます。



［リソース］表示枠には，次の項目があります。

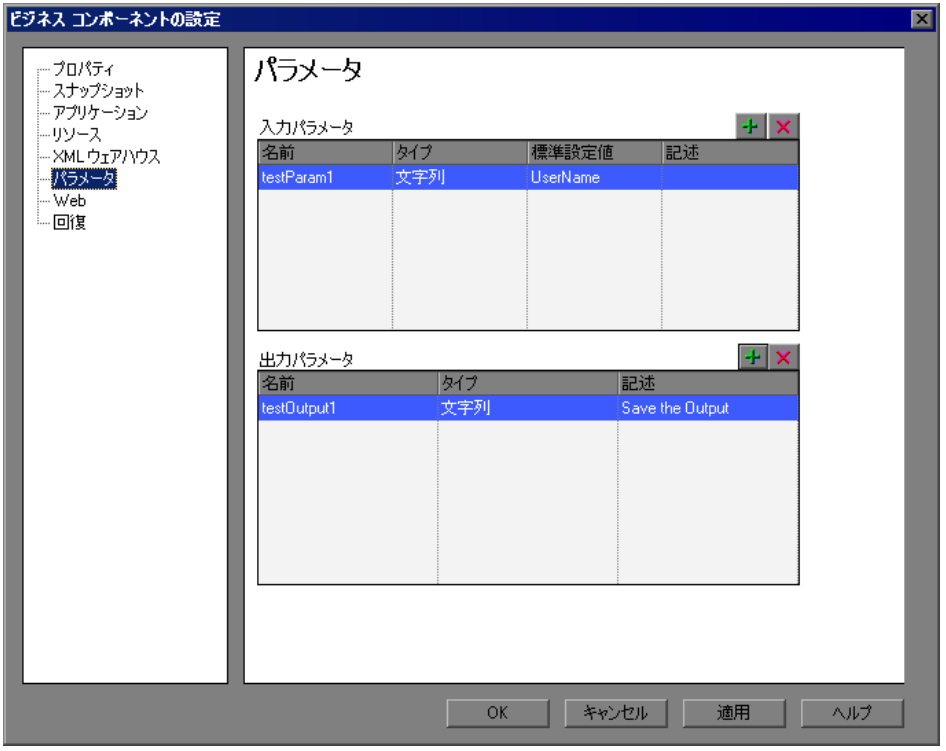
設定領域	説明
［関数ライブラリ］	コンポーネントに関連付けられているアプリケーション領域を通じてコンポーネントに現在関連付けられている関数ライブラリのリストが表示されます。関数ライブラリの関連付けの詳細については，460 ページ「関数ライブラリの関連付け」および 402 ページ「関連付けられている関数ライブラリを使用した作業」を参照してください。
［オブジェクト リポジトリ］	コンポーネントに関連付けられているアプリケーション領域を通じてコンポーネントに現在関連付けられている共有オブジェクト・リポジトリのリストが（ローカル・オブジェクト・リポジトリに加えて）表示されます。コンポーネントは，Quality Center に格納されている共有オブジェクト・リポジトリ・ファイルを使用します。オブジェクト・リポジトリのアプリケーション領域への関連付けの詳細については，465 ページ「共有オブジェクト・ライブラリの関連付け」を参照してください。

コンポーネントのパラメータの定義

［ビジネス コンポーネントの設定］ダイアログ・ボックスの［パラメータ］表示枠では，コンポーネントに値を渡す入力コンポーネント・パラメータと，コンポーネントから外部のソースに値を渡す出力コンポーネント・パラメータを定義できます。また，［パラメータ］表示枠を使用して既存のコンポーネント・パラメータの変更や削除を行うこともできます。

コンポーネント・パラメータは，コンポーネント・ステップ内の入力値と出力値をパラメータ化するために使用できるパラメータです。コンポーネント・ステップ内でパラメータ値を使用する方法については，『**HP QuickTest Professional ユーザーズ・ガイド**』のパラメータを使った作業に関する項を参照してください。ステップ内でのコンポーネント・パラメータの使用の詳細については，664 ページ「ステップにおけるコンポーネント・パラメータの使用」を参照してください。

各分野のエキスパートは，Quality Center でコンポーネント・パラメータを定義することもできます。詳細については，『**HP Business Process Testing ユーザーズ・ガイド**』を参照してください。



［パラメータ］表示枠には，次の 2 つのパラメータ・リストがあります。

- ［**入力パラメータ**］：コンポーネントが，それを実行または呼び出したソースから受け取れるパラメータを指定します。
- ［**出力パラメータ**］：コンポーネントが，それを実行または呼び出したソースに渡せるパラメータを指定します。

既存のパラメータを編集するには，該当するリストで対象のパラメータを選択して，詳細を編集します。

注：入力および出力のパラメータのリストは，Quality Center のビジネス・コンポーネント・モジュールでも変更できます。詳細については，『**HP Business Process Testing ユーザーズ・ガイド**』を参照してください。

次のボタンを使用して，ビジネス・コンポーネントの入力パラメータおよび出力パラメータの追加および削除ができます。

オプション	説明
	該当するパラメータ・リストにパラメータを追加します。新しいパラメータの名前（大文字小文字を区別）を入力し，パラメータのタイプを選択します。選択できるタイプは，**文字列**，**ブール値**，**日付**，**数値**，または**パスワード**です。コンポーネントにおける当該パラメータの用途など，パラメータの詳細を入力できます。 入力パラメータを定義する場合には，指定のパラメータのタイプに対する標準設定値が自動的に入力されます。パラメータの標準設定値を［**標準設定値**］カラムで入力または変更できます。詳細については，次に示す「入力コンポーネント・パラメータの標準設定値の定義」を参照してください。
	選択したパラメータをビジネス・コンポーネントから削除します。

入力コンポーネント・パラメータの標準設定値の定義

ビジネス・コンポーネントを実行したときにパラメータの値として実際に使用される値は，次の表に示すように，コンポーネントを呼び出したアプリケーション（QuickTest または Quality Center）から送信された値です。

ビジネス・コンポーネントの呼び出し元	パラメータ値の指定場所
QuickTest	［実行］ダイアログ・ボックスの［入力パラメータ］タブ。詳細については，676 ページ「［実行］ダイアログ・ボックス：［入力パラメータ］タブ」を参照してください。
Quality Center	［コンポーネントの反復］ダイアログ・ボックス（テスト計画モジュール）。詳細については，『**HP Business Process Testing ユーザーズ・ガイド**』を参照してください。

コンポーネントの実行時に，1 つ以上の入力パラメータに対して QuickTest または Quality Center から値が渡されなかった場合，QuickTest はパラメータの標準設定値を使用します。

［ビジネス コンポーネントの設定］ダイアログ・ボックスの［パラメータ］表示枠で新しいパラメータを定義するとき，パラメータの標準設定値を指定することも，当該パラメータのタイプに対して QuickTest が割り当てる標準設定値を受け入れることもできます。

値のタイプ	QuickTest の標準設定値
［文字列］	空文字列
［ブール値］	True
［日付］	現在の日付
［数値］	0
［パスワード］	空文字列

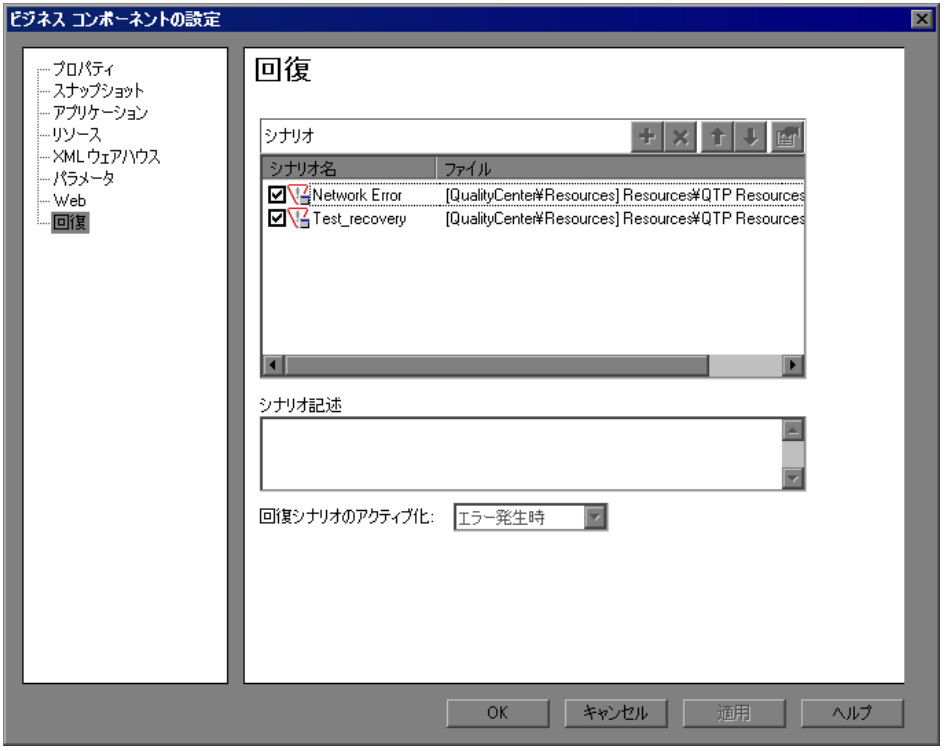
ステップにおけるコンポーネント・パラメータの使用

コンポーネント・パラメータを定義したら，［値設定オプション］ダイアログ・ボックスで入力コンポーネント・パラメータを選択するか，［出力オプション］ダイアログ・ボックスで出力コンポーネント・パラメータを選択して，コンポーネントのステップの値をパラメータ化できます。ステップ内でローカル・パラメータを使用することもできます。ステップにおけるコンポーネント・パラメータとローカル・パラメータの使用の詳細については，『**HP QuickTest Professional ユーザーズ・ガイド**』のパラメータを使った作業に関する項を参照してください。

回復シナリオの設定の表示

回復シナリオの設定では，ビジネス・コンポーネントが実行セッション中の予期しないイベントやエラーからどのように回復するかを指定できます。

［ビジネス コンポーネントの設定］ダイアログ・ボックスの［回復］表示枠には，現在のコンポーネントに関連付けられているアプリケーション領域に関連付けられているすべての回復シナリオの読み取り専用リストが表示されます。



コンポーネントの回復シナリオの設定は，当該コンポーネントのアプリケーション領域に定義します。詳細については，456 ページ「アプリケーション領域の回復シナリオ設定の定義」を参照してください。

［回復］表示枠には，次の項目があります。

設定領域	説明
［シナリオ］	コンポーネントに関連付けられているアプリケーション領域を通じてコンポーネントに関連付けられている各回復シナリオの名前と回復ファイルのパスが表示されます。シナリオ・タイプはアイコンによって示されます。詳細については，458 ページ「関連回復シナリオの指定」を参照してください。
［シナリオ記述］	［**シナリオ**］ボックスで選択したシナリオの説明テキストが表示されます。
［回復シナリオのアクティブ化］	関連付けられているシナリオを次のように実行するかどうか確認するよう QuickTest に指示する設定を表示します。 ➤ ［**各ステップごと**］。ステップを実行するたびに回復メカニズムが呼び出されます ➤ ［**エラー発生時**］。エラーの戻り値を返すステップの後のみ，回復メカニズムが呼び出されます。 ➤ ［**なし**］。回復メカニズムが無効になります。

第 VII 部

コンポーネントの実行と分析

第 26 章

コンポーネントの実行

コンポーネントの作成後は，それを実行してアプリケーションの動作を検査できます。

本章の内容

- コンポーネントの実行について（669 ページ）
- コンポーネント全体の実行（670 ページ）
- コンポーネントの一部の実行（671 ページ）
- ［実行］ダイアログ・ボックス：［結果の場所］タブ（674 ページ）
- ［実行］ダイアログ・ボックス：［入力パラメータ］タブ（676 ページ）

コンポーネントの実行について

コンポーネントを実行すると，それに含まれるステップが QuickTest によって実行されます。コンポーネントのパラメータを設定している場合は，その値を入力するよう求めるメッセージが表示されます。セッションの実行が完了すると，QuickTest によって結果の詳細を示すレポートが表示されます。結果の表示の詳細については，第 27 章「実行セッション結果の表示」を参照してください。

コンポーネント全体を先頭から実行することも，一部分だけを実行することも可能です。コンポーネントを更新して，テスト・オブジェクト記述を変更できます。動的な要素を持つオブジェクトについてコンポーネントを実行することもできます。詳細については，第 5 章「オブジェクト・リポジトリ内のテスト・オブジェクトの管理」を参照してください。

コンポーネント全体の実行

QuickTest では，コンポーネントの実行を開始するステップを特に指定しないかぎり，常に最初のステップから開始されます。ステップを指定して，そこから（またはそこまで）コンポーネントを実行するには，［**ステップから実行**］オプションまたは［**ステップまで実行**］オプションを使用します。これらの機能は，コンポーネントを最初からまたは最後まで実行せずに，コンポーネントの特定のセクションを検査する場合に役立ちます。詳細については，671 ページ「コンポーネントの一部の実行」を参照してください。

コンポーネントを開始するには，［実行］ダイアログ・ボックスを開きます。このダイアログ・ボックスでは，結果の場所を指定し，定義されたコンポーネントのパラメータの値を入力します。

コンポーネントを実行するには，次の手順を実行します。

1 コンポーネントをまだ開いていない場合は，［**ファイル**］＞［**開く**］＞［**ビジネス / スクリプト コンポーネント**］を選択するか，［**開く**］ボタンをクリックして開きます。

ヒント：最近使用したコンポーネントを開く場合は，［**ファイル**］メニューの最近使用したファイルのリストから選択することもできます。

---

2 ツールバーの［**実行**］ボタンをクリックするか，［**オートメーション**］＞［**実行**］を選択します。［実行］ダイアログ・ボックスが開きます。

3 ［実行］ダイアログ・ボックスで，その実行セッションにおける結果の場所（および，必要に応じて入力パラメータの値）を指定します。詳細については，674 ページ「［実行］ダイアログ・ボックス：［結果の場所］タブ」および 676 ページ「［実行］ダイアログ・ボックス：［入力パラメータ］タブ」を参照してください。

4 ［**OK**］をクリックします。［実行］ダイアログ・ボックスが閉じ，実行セッションが開始されます。標準では，実行セッションが終了すると，［テスト結果］ウィンドウが開きます。実行セッション結果の表示の詳細については，第 27 章「実行セッション結果の表示」を参照してください。

注：［オプション］ダイアログ・ボックスの［実行］表示枠で［**実行セッション終了時に結果を表示する**］チェック・ボックスをクリアすると，実行セッションの最後で［テスト結果］ウィンドウは開きません。［オプション］ダイアログ・ボックスの詳細については，第 24 章「グローバル・テスト・オプションの設定」を参照してください。

ヒント：実行セッションを中断する場合は，次のいずれかを行います。

- デバッグ・ツールバーの［**一時停止**］ボタン をクリックするか，［**デバッグ**］＞［**一時停止**］を選択します。実行が停止します。一時停止した実行セッションの実行を再開するには，［**実行**］ボタンをクリックするか，または［**オートメーション**］＞［**実行**］を選択します。
- ［**停止**］ボタンをクリックするか，［**オートメーション**］＞［**停止**］を選択するか，または停止コマンド・ショートカット・キーを押します（停止コマンド・ショートカット・キーの定義の詳細については，638 ページ「テストの実行オプションの設定」を参照してください）。実行セッションが停止され，［テスト結果］ウィンドウが表示されます。

ファイル操作を実行すると（たとえば，別のコンポーネントを開いたり，新規のコンポーネントを作成したりするなど），実行セッションは中断します。

コンポーネントの一部の実行

選択したステップからコンポーネントの最後までコンポーネント内の選択した部分を実行するには，［**ステップから実行**］オプションを使用します。これにより，アプリケーションの特定のセクションだけを検査したり，コンポーネントの特定の部分がスムーズに実行されるかどうかを確認したりできます。

注：また，コンポーネントをコンポーネントの最初から選択したステップまでデバッグ・モードで実行するには，［**デバッグ**］＞［**ステップまで実行**］オプションを使用します。詳細については，746 ページ「[ステップまで実行］コマンドおよび［ステップからデバッグ］コマンドの使用」を参照してください。

選択したステップからコンポーネントを実行するには，次の手順を実行します。

1 アプリケーションが，実行するステップに対応する状態にあることを確認してください。

2 次のいずれかの方法で，コンポーネントの実行を開始するステップを選択します。

選択したステップが，以前のステップで取得した値または定義したパラメータなど，以前のステップに依存していないことを確認します。

3 ［**オートメーション**］＞［**ステップから実行**］を選択します。［実行］ダイアログ・ボックスが開きます。

4 674 ページ「[実行］ダイアログ・ボックス：［結果の場所］タブ」および 676 ページ「[実行］ダイアログ・ボックス：［入力パラメータ］タブ」で説明されている手順に従い，［実行］ダイアログ・ボックスで，実行セッション結果の保存先を選択し，使用する入力パラメータを定義します。

5 ［**OK**］をクリックします。［実行］ダイアログ・ボックスが閉じ，実行セッションが開始されます。

標準設定では，実行セッションが終了すると，［テスト結果］ウィンドウが開きます。実行セッション結果の表示の詳細については，第 27 章「実行セッション結果の表示」を参照してください。

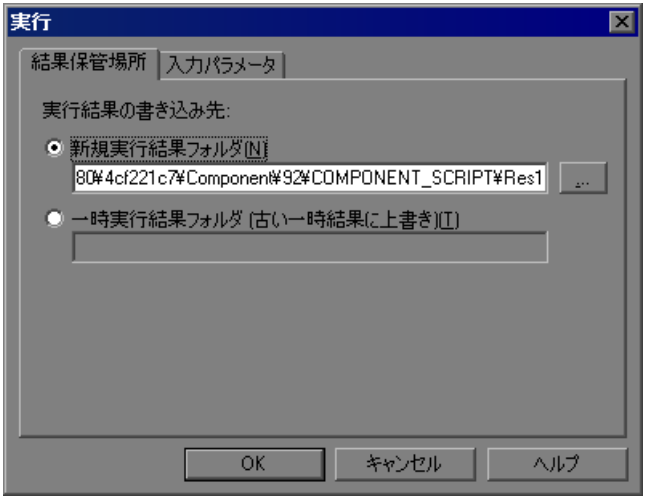
［テスト結果］サマリには，［**ステップから実行**］オプションを使用してコンポーネントが実行されたことを示す注記が表示されます。

注 : ［オプション］ダイアログ・ボックスの［実行］表示枠で［**実行セッション終了時に結果を表示する**］チェック・ボックスをクリアすると，実行セッションの最後で［テスト結果］ウィンドウは開きません。［オプション］ダイアログ・ボックスの詳細については，第 24 章「グローバル・テスト・オプションの設定」を参照してください。

［実行］ダイアログ・ボックス：［結果の場所］タブ

説明	実行セッション結果を保存する場所を指定できます。
使用方法	任意の実行モードで実行セッションを開始すると，［実行］ダイアログ・ボックスが開きます。
詳細	**概念の概要**：669 ページ「コンポーネントの実行」 **主要なタスク**： ➤ 670 ページ「コンポーネント全体の実行」 ➤ 671 ページ「コンポーネントの一部の実行」 ➤ 770 ページ「メンテナンス実行ウィザードを使ったコンポーネントの実行」 ➤ 746 ページ「［ステップまで実行］コマンドおよび［ステップからデバッグ］コマンドの使用」

以下に，［実行］ダイアログ・ボックスの［結果の場所］タブの画像を示します。



［結果の場所］タブのオプション

次のオプションのいずれかを選択します。

➤ ［**新規実行結果フォルダ**］：このオプションには，結果を保存する標準パスのフォルダの名前が表示されます。実行するたびに新しいフォルダが作成されます。標準設定では，コンポーネントの結果はお使いのコンピュータの Quality Center キャッシュ・フォルダに格納されます。

標準設定を受け入れるか，テキスト・ボックスにパスを入力するか，参照ボタンをクリックして別のフォルダを選択して，新しいパスを指定します。このフォルダは，新規であるか，空であるか，QuickTest の コンポーネントファイルのみが含まれているフォルダでなければなりません。

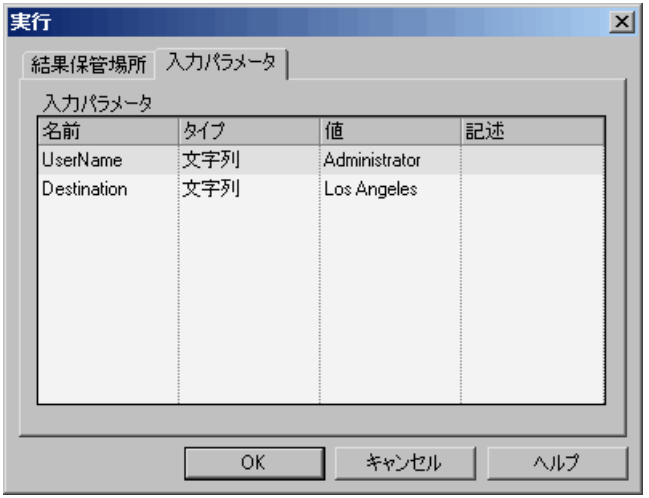
➤ ［**一時実行結果フォルダ**］：一時フォルダに実行結果を保存します。このオプションによって，このフォルダに保存されている結果はすべて上書きされます。

注：QuickTest では，すべてのコンポーネントの一時的な結果が**＜システム・ドライブ＞ ¥Documents and Settings¥ ＜ユーザ名＞ ¥Local Settings¥Temp¥ TempResults** に格納されます。［**一時実行結果フォルダ**］オプションのテキスト・ボックスに表示されるパスは，読み取り専用のため，変更できません。既存の結果フォルダに結果を保存すると，実行セッション開始時に，フォルダの内容が削除されます。

［実行］ダイアログ・ボックス：［入力パラメータ］タブ

説明	実行セッション中に使用する入力パラメータの実行時の値を指定できます。
使用方法	任意の実行モードで実行セッションを開始すると，［実行］ダイアログ・ボックスが開きます。
詳細	**概念の概要**：669 ページ「コンポーネントの実行」 **主要なタスク**： ➤ 670 ページ「コンポーネント全体の実行」 ➤ 671 ページ「コンポーネントの一部の実行」 ➤ 770 ページ「メンテナンス実行ウィザードを使ったコンポーネントの実行」 ➤ 746 ページ「［ステップまで実行］コマンドおよび［ステップからデバッグ］コマンドの使用」 **その他の関連項目**：677 ページ「その他の参照情報」

以下に，［実行］ダイアログ・ボックスの［入力パラメータ］タブの画像を示します。



［入力パラメータ］タブには，コンポーネントに定義された入力パラメータ（［**ファイル**］＞［**設定**］＞［**パラメータ**］ノードを使用）が表示されます。

実行セッション中に使用するパラメータの値を設定するには，特定のパラメータの［**値**］フィールドをクリックして，値を入力するか，ドロップダウン・リストから値を選択します。値を入力しなかった場合，実行セッション中は［ビジネス コンポーネントの設定］ダイアログ・ボックスの標準設定の値が使用されます。

その他の参照情報

関連する概念	➤ コンポーネントのパラメータの設定の詳細については，660 ページ「コンポーネントのパラメータの定義」を参照してください。 ➤ パラメータの使用方法の詳細については，『**HP QuickTest Professional ユーザーズ・ガイド**』のパラメータを使った作業に関する項を参照してください。

第 27 章

実行セッション結果の表示

コンポーネントの実行が終わったら，実行セッション中に発生した重要なイベントに関するレポートを表示できます。

注：QuickTest から［テスト結果］ウィンドウを開いた場合は，ビジネス・プロセス・テストの実行結果は表示できません。ビジネス・プロセス・テストのテスト結果を表示するには，表示する反復の結果を選択して Quality Center から開きます。

本章の内容

- 実行セッション結果の表示について（680 ページ）
- ［テスト結果］ウィンドウ（681 ページ）
- 実行セッションの結果の表示（689 ページ）
- 実行結果の削除（710 ページ）
- Quality Center プロジェクトへの手動による実行セッション中に検出された不具合の送信（718 ページ）
- テスト結果の表示のカスタマイズ（719 ページ）

実行セッション結果の表示について

実行セッションが終了すると，実行セッションの結果を［テスト結果］ウィンドウで見ることができます。標準設定では，テストの実行が終了すると，［テスト結果］ウィンドウが自動的に開きます。この動作を変更するには，［オプション］ダイアログ・ボックスの［実行］表示枠で［**実行セッションの終了時に結果を表示する**］チェック・ボックスをクリアします。

［テスト結果］ウィンドウには，実行セッション中に実行されたステップの詳細が表示されます。単一のテスト反復が表示されます。

コンポーネントの実行が終わったら，［テスト結果］ウィンドウに，実行セッションのあらゆる側面が表示されます。これには次のものがあります。

- 高レベルの結果概要レポート（テストの成功 / 失敗のステータス）
- すべてのテスト実行に使用されたデータ
- アプリケーション・エラーの発生場所を正確に示す，ステップの展開可能なツリー
- コンポーネント内で不具合が発生した正確な位置
- 特定のステップでのアプリケーションの状態の静止画像
- 特定のステップまたはコンポーネント全体のアプリケーションの状態のムービー・クリップ
- コンポーネントの各段階で，成功または失敗した各ステップの詳細な説明
- テストのための Quality Center 情報（コンポーネントが Quality Center に保存することを選択した場合）。

注 : ［テスト結果］ウィンドウには，結果をツリー階層で最大 300 レベルまで表示できます。結果が 300 レベル以上ネストされている場合は，**results.xml** ファイルを手動で開いてレポート全体を表示できます。

［テスト結果］ウィンドウ

実行セッションが終わったら，［テスト結果］ウィンドウで結果を参照します。標準設定では，実行セッションが完了すると，［テスト結果］ウィンドウが開きます。標準設定の変更方法の詳細については，638 ページ「テストの実行オプションの設定」を参照してください。

［テスト結果］ウィンドウの左側の表示枠には，実行結果ツリーが表示されます。［テスト結果］ウィンドウの右側の表示枠には，実行結果ツリーで選択したステップの詳細が表示されます。選択したステップの詳細には，コンポーネントのサマリ，ステップの詳細，アプリケーションの静止画像，またはアプリケーションのムービー

［テスト結果］ウィンドウは，［**スタート**］メニューからスタンドアロンのアプリケーションとして開くことができます。［テスト結果］ウィンドウを開くには，［**スタート**］＞［**プログラム**］＞［**QuickTest Professional**］＞［**Test Results Viewer**］を選択します。

コンポーネントの実行結果の例は次のとおりです。

注： この例では，関連付けられている関数ライブラリで実行エラーが発生したため，コンポーネントは失敗しました。実行エラーが発生していなければ，［**結果**］は［**完了**］を示します。

［テスト結果］ウィンドウには，次の主な要素があります。

- **テスト結果タイトル・バー**：コンポーネントの名前を表示します。
- **メニュー・バー**：使用可能なコマンドのメニューが表示されます。
- **実行結果ツールバー**：実行セッション結果を表示するボタンが含まれています（このツールバーを表示するには，［**表示**］＞［**テスト結果ツールバー**］を選択します）。詳細については，686 ページ「実行結果ツールバー」を参照してください。
- **実行結果ツリー**：実行結果ツリーに実行結果が視覚的に表示されます。実行結果ツリーは，［テスト結果］ウィンドウの左側の表示枠内にあります。詳細については，683 ページ「実行結果ツリー」を参照してください。
- ［**結果の詳細**］タブ：実行結果ツリーで選択したノードの詳細が表示されます。［結果の詳細］タブは，［テスト結果］ウィンドウの右側の表示枠内にあります。詳細については，684 ページ「実行結果詳細」を参照してください。
- ［**画面レコーダ**］タブ：テスト結果に関連付けられている記録されたムービーが表示されます。［画面レコーダ］タブは，［テスト結果］ウィンドウの右側の表示枠内にあります。詳細については，699 ページ「アプリケーションの静止画像とムービーの表示」を参照してください。
- ［**システム モニタ**］タブ：システム監視をコンポーネントに対して有効にすることはできません。
- **ステータス・バー**：現在選択されているコマンドのステータスが表示されます（このステータス・バーを表示するには，［**表示**］＞［**ステータス バー**］を選択します）。

［テスト結果］ウィンドウの外観は変更できます。詳細については，688 ページ「［テスト結果］ウィンドウの外観の変更」を参照してください。

実行結果ツリー

［テスト結果］ウィンドウの左側の表示枠には，**実行結果ツリー**が表示されます。実行結果ツリーには，実行セッション結果が次のように図示されます。

- ✔ 成功したステップを示します。このアイコンは，コンポーネント・ステップに次が含まれている場合にのみ表示されます。
  - VerifyProperty などの検証操作（関数）
  - **micPass** ステータスの AddToTestResults（またはそれと同等のもの）
- ✘ 失敗したステップを示します。ステップが失敗すると，そのすべての親ステップ（ルートのコンポーネントまで）が失敗となります。
- ! 警告を示します。そのステップは成功しなかったけれども，コンポーネントは失敗しなかったことを意味します。
- ! ⊗ チェックポイントのオブジェクトが見つからない場合など，予期せず失敗したステップを示します。
- スマート認識メカニズムによってオブジェクトが正しく見つかったことを示します。
- 回復シナリオがアクティブになったことを示します。
- 実行セッションが，終了する前に停止したことを示します。
- [password].SetSecure テスト・オブジェクト名が角括弧で囲まれている場合，このテスト・オブジェクトは実行セッション中に動的に作成されたことを示します。動的テスト・オブジェクトは，プログラミングによる記述，または ChildObjects メソッドによって返されるオブジェクトを使用して作成されます。また，動的テスト・オブジェクトは，オブジェクト・リポジトリに保存されません。
- **メンテナンス・モードでの更新結果**を表示します。これは，失敗したステップでメンテナンス実行ウィザードによって行われた**アクション**と，そのアクションの**詳細**を示すテーブルです。メンテナンス実行モードで実行されたコンポーネントのみが表示されます。メンテナンス実行モードの詳細については，第 30 章「コンポーネントの保守」を参照してください。

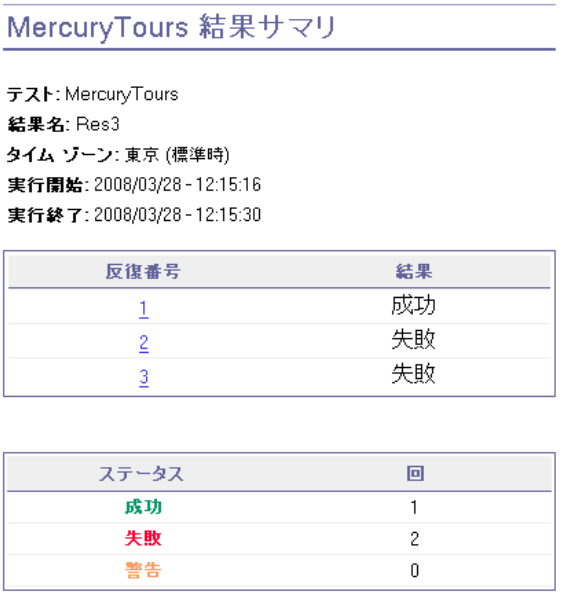
  ツリーに表示される詳細のレベルを変更するには，実行結果ツリーの分岐を折りたたむか，展開します。

実行結果詳細

標準設定では，［テスト結果］ウィンドウが開くと，コンポーネントのサマリがウィンドウの右側の表示枠の［**結果の詳細**］タブに表示されます。

［テスト結果］ウィンドウの右側の表示枠には，［**結果の詳細**］，［**画面レコーダ**］，および［**システム モニタ**］というタブがあります。実行結果ツリーの最上位ノードを選択すると，［結果の詳細］タブにコンポーネントの結果のサマリが表示されます。ツリーで分岐やステップを選択すると，［結果の詳細］タブにステップの詳細が表示されます。［結果の詳細］タブには，強調表示されているステップのアプリケーションの静止画像も表示される場合があります。

テスト結果ツリーの最上位ノードを選択すると，［結果の詳細］タブにはコンポーネントの名前，製品名（コンポーネントの場合），結果名，実行セッションの開始日時と終了日時，および反復が成功したか失敗したかが表示されます。


MercuryTours 結果サマリ

テスト: MercuryTours
結果名: Res3
タイム ゾーン: 東京 (標準時)
実行開始: 2008/03/28 - 12:15:16
実行終了: 2008/03/28 - 12:15:30

反復番号	結果
1	成功
2	失敗
3	失敗

ステータス	回
成功	1
失敗	2
警告	0

［結果の詳細］タブには，次の追加情報を表示することもできます。

➤ コンポーネントの場合，起こりうる結果は［**完了**］または［**失敗**］です。

- Web サービス・アドインがインストールされていて実行セッション中にロードされた場合，Web サービスの実行ツールキットが［結果の詳細］タブに表示されます。コンポーネントに Web サービス・ステップが含まれていなくても，実行ツールキットは表示されます。＊＊＊ここまで終了
- コンポーネントが**メンテナンス実行モード**で実行された場合，［結果の詳細］タブには［**メンテナンス サマリ**］が表示されます。［**メンテナンス サマリ**］には，コンポーネント内で更新および追加されたオブジェクトの数が表示されます。また，コンポーネント内で更新およびコメントに変換されたステップの数も表示されます。［**オブジェクト リポジトリ変更レポート**］には，メンテナンス実行ウィザードがオブジェクト・リポジトリに加えた特定の変更と次の項が表示されます。
  - ［**オブジェクトの追加**］：メンテナンス実行ウィザードによってオブジェクト・リポジトリに追加されたオブジェクトが表示されます。
  - ［**変更されたオブジェクトの説明**］：メンテナンス実行ウィザードによって実行されたオブジェクト・リポジトリへの変更が示されます。

  メンテナンス実行モードの詳細については，767 ページ「コンポーネントの保守」を参照してください。
- コンポーネントが，Quality Center から実行された場合，サーバ，プロジェクト，テスト・セット，およびテスト・インスタンスの名前も表示されます。

実行結果ツールバー

実行結果ツールバーには，実行セッションの結果を表示するためのボタンがあります。

ボタン	名前	ショートカット・キー	説明
	［開く］	CTRL+O	［テスト結果を開く］ダイアログ・ボックスが開き，ファイル・システムまたは Quality Center から保存されている実行結果を開くことができます。詳細については，690 ページ「特定の実行を表示するためのテスト結果の表示」を参照してください。
	［印刷］	CTRL+P	［印刷］ダイアログ・ボックスが開き，実行セッションの結果を印刷できます。詳細については，706 ページ「実行セッション 結果の印刷」を参照してください。
	［フィルタ］	CTRL+T	［フィルタ］ダイアログ・ボックスが開き，表示されている情報をフィルタ処理できます。詳細については，696 ページ「テスト結果のフィルタリング」を参照してください。
	［Quality Center への接続］		［Quality Center への接続 - サーバへの接続］ダイアログ・ボックスが開き，Quality Center プロジェクトに接続できます。詳細については，46 ページ「Quality Center プロジェクトへの接続」を参照してください。
	［不具合の追加］		不具合を Quality Center プロジェクトに追加できます。Quality Center に現在接続されていない場合は，［Quality Center への接続 - サーバへの接続］ダイアログ・ボックスを開きます。詳細については，718 ページ「Quality Center プロジェクトへの手動による実行セッション中に検出された不具合の送信」を参照してください。

ボタン	名前	ショートカット・キー	説明
	[検索]	CTRL+F	[検索] ダイアログ・ボックスが開き，エラーまたは警告など，特定の結果を含むステップを検索できます。詳細については，697 ページ「結果ステップの検索」を参照してください。
	[次を検索]	ALT+N	定義されている検索フィルタに一致する次のステップを検索します。検索フィルタは [検索] ダイアログ・ボックスで定義します（697 ページ「結果ステップの検索」を参照）。
	[前を検索]	ALT+P	定義されている検索フィルタに一致する前のステップを検索します。検索フィルタは [検索] ダイアログ・ボックスで定義します（697 ページ「結果ステップの検索」を参照）。
	[前のノードに移動]	BACKSPACE	実行結果ツリーで以前選択されたノードにカーソルを移動します。詳細については，693 ページ「実行結果ツリーの操作」を参照してください。
	[次のノードに移動]	ALT+RIGHT ARROW	[**前のノードに移動**] ボタンをクリックする前に実行結果ツリーで選択したノードにカーソルを移動します。詳細については，693 ページ「実行結果ツリーの操作」を参照してください。
	[QuickTest 内のステップへ移動]	CTRL+J	QuickTest ウィンドウをアクティブにして，[テスト結果] ツリーで選択したノードに対応するコンポーネントのステップを強調表示します。この機能は，実行時データ・テーブルおよびテストのサマリ・ノードでは無効になっています。詳細については，695 ページ「QuickTest 内でのステップへの移動」を参照してください。
	[ヘルプトピック]		[**HP QuickTest Professional Test Results Help**] が開きます。

［テスト結果］ウィンドウの外観の変更

標準設定では，［テスト結果］ウィンドウの外観（ルック・アンド・フィール）は QuickTest ウィンドウと同じで，Microsoft Office 2003 テーマが使用されます。［テスト結果］ウィンドウの外観は，必要に応じて変更できます。

［テスト結果］ウィンドウの外観を変更するには，次の手順を実行します。

［テスト結果］ウィンドウで［**表示**］＞［**ウィンドウのテーマ**］を選択し，使用可能なテーマのリストからウィンドウの外観を選択します。たとえば，Microsoft Office 2000 または Microsoft Windows XP のテーマを適用できます。

注：コンピュータが Windows XP テーマを使用するように設定されている場合にのみ，［テスト結果］ウィンドウに Microsoft Windows XP テーマを適用できます。

ヒント：QuickTest のメイン・ウィンドウに使用するテーマも変更できます。詳細については，50 ページ「QuickTest ウィンドウの外観の変更」を参照してください。

実行セッションの結果の表示

標準設定では，実行セッションの最後に実行結果が［テスト結果］ウィンドウに表示されます（標準設定は，［オプション］ダイアログ・ボックスで変更できます。実行の標準設定の詳細については，638 ページ「テストの実行オプションの設定」を参照）。

また，現在のコンポーネントの以前の実行結果や，ほかのコンポーネントの結果を表示することもできます。実行セッションの結果は画面でプレビュー，印刷，または HTML ファイルにエクスポートできます。

詳細については，次を参照してください。

- 690 ページ「特定の実行を表示するためのテスト結果の表示」
- 693 ページ「実行結果ツリーの操作」
- 694 ページ「結果の詳細の表示」
- 695 ページ「QuickTest 内でのステップへの移動」
- 696 ページ「テスト結果のフィルタリング」
- 697 ページ「結果ステップの検索」
- 697 ページ「Quality Center からのコンポーネント実行の結果の表示」
- 699 ページ「アプリケーションの静止画像とムービーの表示」
- 704 ページ「テスト結果のプレビュー」
- 706 ページ「実行セッション 結果の印刷」
- 707 ページ「実行結果のエクスポート」

特定の実行を表示するためのテスト結果の表示

現在のコンポーネントに関して保存されている結果を表示できます。また，ほかのコンポーネントに関して保存されている結果も表示できます。

現在のテストまたはその他のテストの保存されている結果を表示するには，次の手順を実行します。

QuickTest ウィンドウで［**結果**］ボタンをクリックするか，［**オートメーション**］＞［**結果**］を選択します。

実行に対して保存されている結果が 1 つのみの場合は，実行セッションの結果が表示されます。複数の結果がある場合，または現在のコンポーネントに対して結果がない場合は，［テスト結果を開く］ダイアログ・ボックスが開きます。

現在のテストまたはその他のテストの保存されている結果を［テスト結果］ウィンドウ内に表示するには，次の手順を実行します。

［**開く**］ボタンをクリックするか，［**ファイル**］＞［**開く**］を選択します。［テスト結果を開く］ダイアログ・ボックスが開きます。

現在のコンポーネントに関する実行セッションの結果が一覧表示されます。結果の 1 つを表示するには，結果を選択して［**開く**］をクリックします。

ヒント：

- 指定したコンポーネントのパスを変更した後に結果リストを更新するには，［**更新**］をクリックします。
- ［テスト結果］ウィンドウは，［**スタート**］メニューからスタンドアロンのアプリケーションとして開くことができます。［テスト結果］ウィンドウを開くには，［**スタート**］＞［**プログラム**］＞［**QuickTest Professional**］＞［**Test Results Viewer**］を選択します。

注：QuickTest から［テスト結果］ウィンドウを開いた場合は，ビジネス・プロセス・テストの実行結果は表示できません。ビジネス・プロセス・テストのテスト結果を表示するには，表示する反復の結果を選択して Quality Center から開きます。

ファイル・システムまたは Quality Center での結果の検索

標準設定では，コンポーネントの実行結果はコンピュータの Quality Center キャッシュ・フォルダに保存されます。コンポーネントを実行するとき，［実行］ダイアログ・ボックスの［結果保管場所］タブを使用して結果の保存場所を指定することもできます。結果ファイルの保存場所を指定すると，ファイル・システム内で結果ファイルの場所を見つけやすくなります。詳細については，674 ページ「［実行］ダイアログ・ボックス：［結果の場所］タブ」を参照してください。

コンポーネントごと，または結果ファイルごとに結果を検索できます。

コンポーネントごとに結果を検索するには，次の手順を実行します。

1 ［テスト結果］ウィンドウで［**ツール**］＞［**Quality Center への接続**］を選択するか，または［**Quality Center への接続**］ボタンをクリックして，Quality Center プロジェクトに接続します。

2 [**開く**] ボタンをクリックするか，[**ファイル**] > [**開く**] を選択します。[テスト結果を開く] ダイアログ・ボックスが開きます。

3 次の手順のいずれかを実行します。

- [テスト結果を開く] ダイアログ・ボックスで，コンポーネントの結果ファイルを含むフォルダのパスを入力します。
- 参照ボタンをクリックして [テストを開く] ダイアログ・ボックスを開きます。サイドバーで，たとえば，**ファイル・システム**または Quality Center **テスト計画**など，結果を表示するコンポーネントの場所を選択します。コンポーネントを参照して選択し，[**開く**] をクリックします。

4 [テスト結果を開く] ダイアログ・ボックスで，表示するコンポーネント結果セットを強調表示し，[**開く**] をクリックします。[テスト結果] ウィンドウに選択した結果が表示されます。

ファイル・システムで結果ファイルを検索するには，次の手順を実行します。

1 [テスト結果を開く] ダイアログ・ボックスの中で，[**ファイルを開く**] ボタンをクリックして [結果ファイルの選択] ダイアログ・ボックスを開きます。

2 コンポーネントの結果ファイルが格納されているフォルダを参照します。

3 表示する（**.xml**）結果ファイルを選択して強調表示してから，[**開く**] をクリックします。[テスト結果] ウィンドウに選択した結果が表示されます。

実行結果ツリーの操作

ツリーに表示される詳細のレベルを選択するには，実行結果ツリーの分岐を折りたたむか，展開します。

- 分岐を展開するには，その分岐を選択して分岐アイコンの左にある展開（+）記号をクリックするか，キーボードのテンキー部分にあるプラス・キー（+）を押します。ツリーの分岐の詳細が表示され，展開（+）記号が折りたたみ（-）記号に変わります。
- ［テスト結果］ウィンドウを初めて開いたときは，ツリーは一度に 1 レベル展開されます。親分岐の下の子分岐を以前に展開した場合は，この状態は，親分岐を展開または折りたたんだときに維持されます。
- 1 つの分岐と，その分岐の下にあるすべての分岐を展開するには，その分岐を選択し，キーボードのテンキーにあるアスタリスク・キー（*****）を押します。
- 実行結果ツリーの分岐をすべて展開するには，［**表示**］＞［**すべて展開**］を選択するか，分岐を右クリックして［**すべて開く**］を選択するか，またはツリーの最上位レベルを選択してキーボードのテンキーにあるアスタリスク・キー（*****）を押します。
- 分岐を閉じるには，その分岐を選択して分岐のアイコンの左にある折りたたみ（-）記号をクリックするか，キーボードのテンキー部分にあるマイナス・キー（-）を押します。結果ツリーでは，その分岐の詳細が消えて，折りたたみ記号が展開記号（+）に変わります。
- 実行結果ツリーのすべての分岐を折りたたむには，［**表示**］＞［**すべて折りたたみ**］を選択するか，または分岐を右クリックして［**すべて折りたたみ**］を選択します。

- 実行結果ツリー内で以前に選択したノード間を移動するには，［**前のノードに移動**］または［**次のノードに移動**］ボタンをクリックします。

- ［テスト結果］内で特定のステップを検索するには，［**検索**］ボタンをクリックするか，［**ツール**］＞［**検索**］を選択します。詳細については，697 ページ「結果ステップの検索」を参照してください。

結果の詳細の表示

反復またはステップの結果を表示できます。実行結果ツリーでステップを選択した場合，［テスト結果］ウィンドウの右側に選択したステップの詳細が表示されます。［オプション］ダイアログ・ボックスの［実行］表示枠での設定に応じて，［テスト結果］ウィンドウの右側は 2 つの表示枠に分割されます。下の表示枠には，選択したステップの静止画像（選択したステップによっては別のデータ）が表示されます。右側の表示枠には，実行セッションからのムービーを表示できる［画面レコーダ］タブ，［システム モニタ］タブも表示されます ビジネス・プロセス・テストでは，システム監視はサポートされていません。このため，コンポーネントで作業する場合，このタブは常に空になります。。詳細については，699 ページ「アプリケーションの静止画像とムービーの表示」および 638 ページ「テストの実行オプションの設定」を参照してください。

結果は，次のいずれかになります。

- チェックポイントは含まれない，実行が成功したステップは，［テスト結果］ウィンドウの右側に「**完了**」と表示されます
- チェックポイントを含むステップは，［テスト結果］ウィンドウの右側に「**成功**」または「**失敗**」と表示され，ツリー・ウィンドウでは ✔ や ✘ アイコンで見分けることができます。

注：成功しなかったが，コンポーネントの実行停止には至らなかったステップは，［テスト結果］ウィンドウの右側に「**警告**」と表示され，❗や❗ アイコンで示されます。コンポーネントは，まだ［**成功**］または［**完了**］と表示されている場合があります。

QuickTest 内でのステップへの移動

実行結果ツリーではノードに対応する QuickTest のステップを表示できます。

ノードに対応するテストのステップを表示するには，次の手順を実行します。

1 実行結果ツリーでノードを選択します。

2 次のいずれかを実行します。

a 実行結果ツールバーで［**QuickTest 内のステップへ移動**］ボタンをクリックします。

b 右クリックしてショートカット・メニューで［**QuickTest 内のステップへ移動**］を選択します。

c［**表示**］＞［**QuickTest 内のステップへ移動**］を選択します。

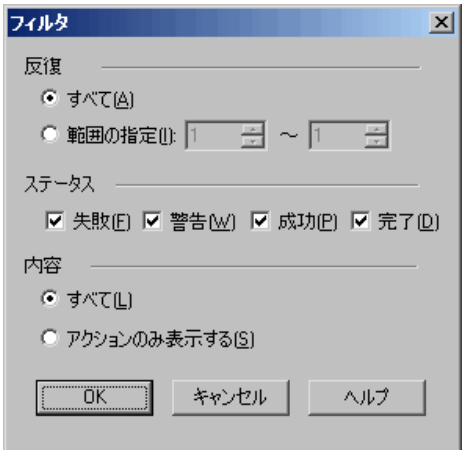
3 QuickTest ウィンドウがアクティブになり，ステップが強調表示されます。

ステップに移動するには，次の条件を満たす必要があります。

- QuickTest が実行されていて，結果が［テスト結果］ウィンドウに表示されているコンポーネントに対して開かれている。
- コンポーネントが，［QuickTest 内のステップへ移動］機能をサポートするバージョンの QuickTest で実行された。
- ノードに，QuickTest 内に対応するステップがある。この機能は，実行時データ・テーブルおよびビジネス・コンポーネントのサマリ・ノードでは無効になっている。
- 回復シナリオによってステップが実行されていない。
- ステップがデバッグ・ビューアの［ウォッチ］または［コマンド］のタブから実行されていない。
- コンポーネントが実行セッションの前に保存された。
- コンポーネントが［**標準**］モードで実行された。**標準**モードでの QuickTest の実行の詳細については，638 ページ「テストの実行オプションの設定」を参照してください。

テスト結果のフィルタリング

［フィルタ］ダイアログ・ボックスでは，［テスト結果］ウィンドウの実行結果ツリーに表示する反復を絞り込めます。

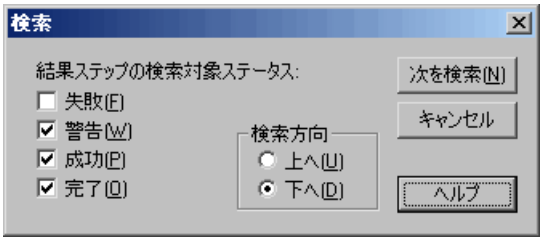


次のオプションを使用できます。

オプション	説明
反復	（このオプションはテストでのみ使用できます。） ［
ステータス	➤［**失敗**］：失敗したステップの実行結果を表示します。 ➤［**警告**］：ステータスが**警告**のステップ（成功はしなかったがテストが失敗する原因にはならなかったステップ）に関する結果を表示します。 ➤［**成功**］：失敗したステップの実行結果を表示します。 ➤［**完了**］：ステータスが**完了**のステップ（ステップの実行に成功したが，成功，失敗，警告のステータスを受け取らなかったステップ）に関する実行結果を表示します。
内容	（このオプションはテストでのみ使用できます。） ［

結果ステップの検索

［検索］ダイアログ・ボックスでは，エラーや警告など特定のステップを，テスト結果から検索することができます。たとえば，**成功**と**完了**など，ステータスの組み合わせを選択して検索できます。



次のオプションを使用できます。

オプション	説明
失敗	失敗したステップを［テスト結果］の中で検索します。
警告	警告が発行されたステップを検索します。
成功	成功したステップを［テスト結果］の中で検索します。
完了	実行を完了したステップを検索します。
検索方向	［テスト結果］のステップを先頭（上へ）に向かって検索するか，末尾（下へ）に向かって検索するかを指定します。

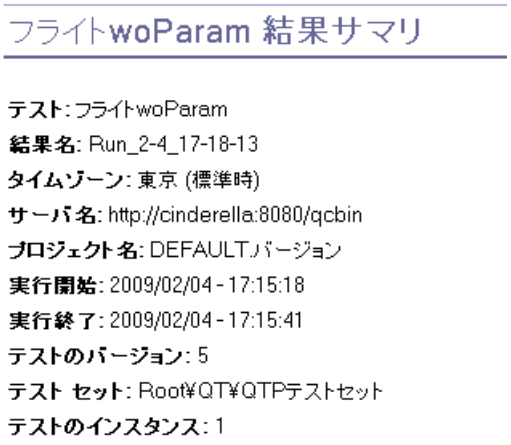
Quality Center からのコンポーネント実行の結果の表示

Quality Center から QuickTest コンポーネントが含まれるビジネス・プロセス・テストを実行すると，ホスト・コンピュータで Quality Center サーバが開き，そのコンピュータからコンポーネントを実行します。すべての実行結果は，実行されたコンポーネントの標準設定の場所に保存されます。

QuickTest のコンポーネント実行の結果は，Quality Center から表示できます。結果にアプリケーションのムービーが含まれている場合，Quality Center で表示できます。

コンポーネントが，Quality Center から実行された場合，実行結果には，681 ページ「[テスト結果］ウィンドウ」で説明したのと同じ情報，および次の追加フィールドが含まれます。

- ［**サーバ名**］：コンポーネントが実行された Quality Center サーバの名前を指定します。
- ［**プロジェクト名**］：コンポーネントが実行された Quality Center ドメインおよびプロジェクトを＜ **domain_name.project_name** ＞の形式で指定します。
- ［**テスト・セット**］：ビジネス・プロセス・テストの場所を指定します。
- ［**テストのインスタンス**］：ビジネス・プロセス・テスト内のテストのインスタンス番号を指定します。たとえば，ビジネス・プロセス・テストに同じテストが 2 回含まれている場合，テスト・インスタンス 1 とテスト・インスタンス 2 の結果を表示できます。



Quality Center に保存されているコンポーネントは，QuickTest から実行されるが，結果を一時保存場所に保存することを選択した場合，［**テスト セット**］および［**テスト インスタンス**］フィールドは結果に表示されません。

アプリケーションの静止画像とムービーの表示

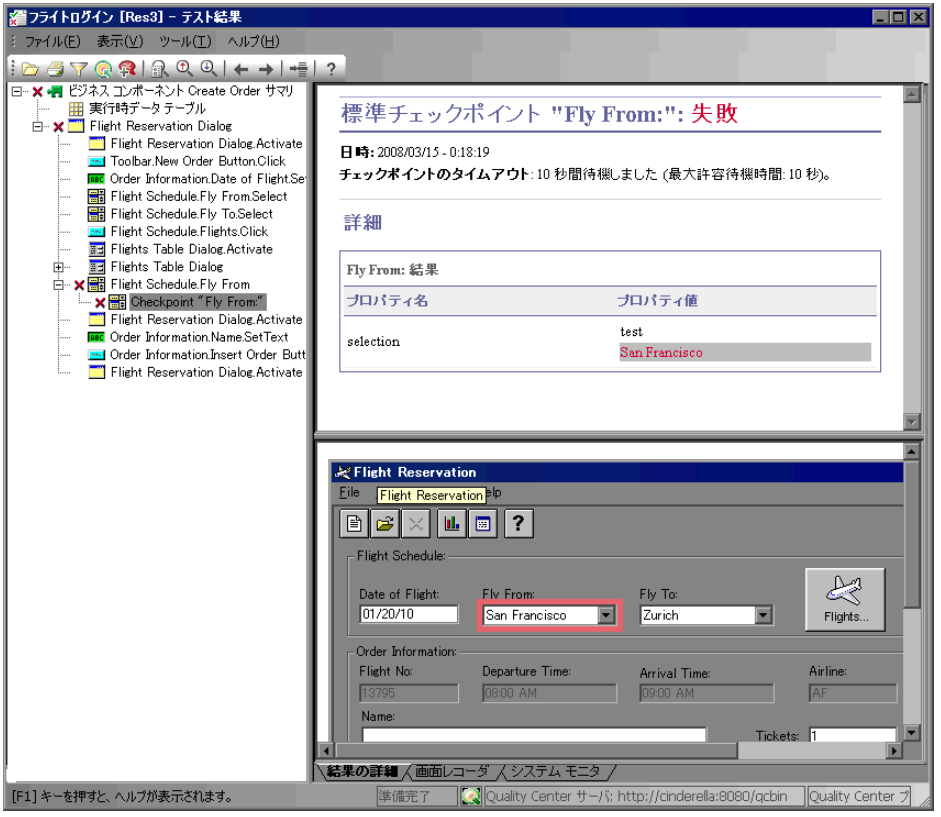
QuickTest Professional では，実行セッション中にアプリケーションの静止画像とムービーをキャプチャできます。キャプチャされたファイルは，［テスト結果］ウィンドウで表示できます。右側の表示枠の［**結果の詳細**］および［**画面レコーダ**］タブでは，静止画像とテキストの詳細，またはアプリケーションのムービーを表示できます。

ヒント：Reporter ユーティリティ・オブジェクトの［**ReportEvent**］メソッドを使用して，画像をプログラムの中から［**結果の詳細**］タブに追加することもできます。詳細については，『**QuickTest ProfessionalObject Model Reference**』（英語版）の「**ユーティリティ オブジェクト**」の項を参照してください。

［オプション］ダイアログ・ボックスの［実行］＞［画面キャプチャ］表示枠で，アプリケーションのムービーをキャプチャするように QuickTest を設定します。詳細については，638 ページ「テストの実行オプションの設定」を参照してください。

アプリケーションの静止画像の表示

標準設定では，QuickTest は失敗したステップに対してアプリケーションの静止画像を保存します。実行結果ツリーで失敗したステップを選択して［**結果の詳細**］タブを選択すると，［テスト結果］ウィンドウの右下の表示枠に，実行結果ツリーで強調表示したステップに対応するアプリケーションの画面キャプチャが表示されます。



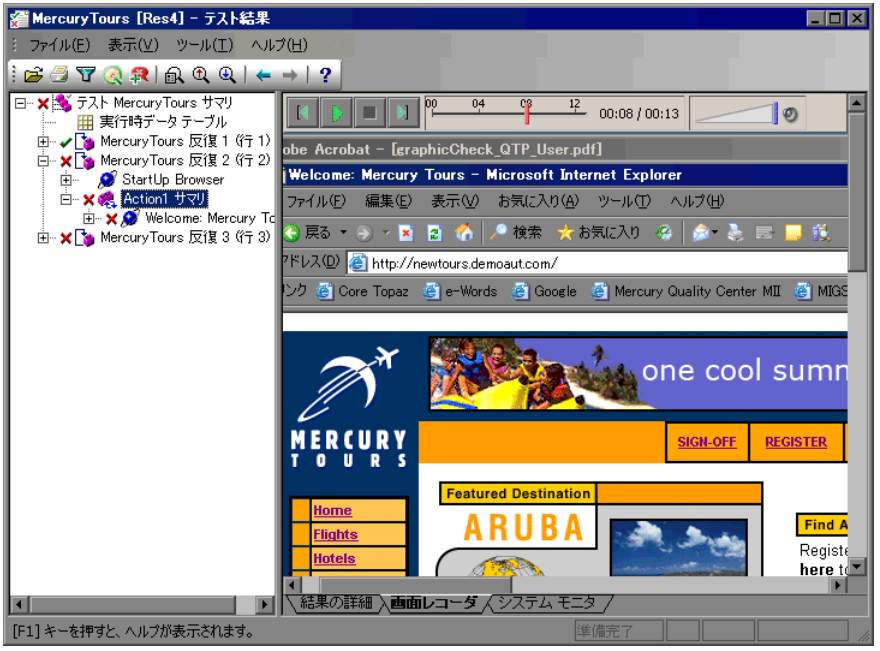
強調表示したステップにエラーがない場合，右側の表示枠には結果詳細が表示され，画面キャプチャは表示されません。

［オプション］ダイアログ・ボックスの［実行］＞［画面キャプチャ］表示枠で［**静止画像キャプチャをテスト結果へ保存**］オプションを使用することによって，静止画像をいつテスト結果に保存するかに関する条件を変更できます。詳細については，642 ページ「［オプション］ダイアログ・ボックス：［実行］＞［画面キャプチャ］表示枠」を参照してください。

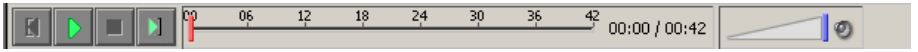
実行セッションのムービーの表示

QuickTest では，実行セッション中にアプリケーションのムービーを保存できます。これは，アプリケーションがテスト条件下でどのように動作するかの確認や，コンポーネントのデバッグに役立ちます。ムービー全体，または特定の部分を選択して，ビューに表示できます。実行結果ツリーでステップを選択して［**画面レコーダ**］タブをクリックすると，［テスト結果］ウィンドウの右下の表示枠に，実行結果ツリーで強調表示したステップに対応するムービーのフレームが表示されます。

［オプション］ダイアログ・ボックスの［実行］＞［画面キャプチャ］表示枠で［**ムービーをテスト結果へ保存**］オプションを使用すると，ムービーを保存するために QuickTest で使用する条件をカスタマイズできます。詳細については，638 ページ「テストの実行オプションの設定」を参照してください。



［画面レコーダ］タブの最上部にコントロールが表示されます。ここで再生，一時停止，停止，ムービーの最初のフレームに移動，ムービーの最後のフレームに移動，ボリューム操作ができます。また，スライダ・バーをドラッグして，ムービーをスクロールすることもできます。



ヒント：

- ［テスト結果］ウィンドウの右側の表示枠をダブルクリックすると，［画面レコーダ］が拡張して実行結果ツリーが隠れます。再度ダブルクリックすると，［画面レコーダ］は元のサイズに戻り，実行結果ツリーが表示されます。［画面レコーダ］が拡張している場合，［画面レコーダ］の最上部にある再生コントロールは最後のマウス操作，つまり［画面レコーダ］のどこかをクリックしてから約 3 秒後に自動的に隠れます。再度マウスを動かすと，再び表示されます。
- ［画面レコーダ］では，デスクトップ全体のムービーが保存されます。実行セッション中は QuickTest を最小化して，ムービーのキャプチャ中に QuickTest ウィンドウがアプリケーションの上に部分的に重ならないようにできます。実行セッション中に QuickTest を最小化する方法の詳細については，810 ページ「QuickTest ウィンドウ・レイアウトのカスタマイズ」を参照してください。

テスト結果からのムービーの削除

実行結果から保存されているムービーを削除できます。これにより，実行結果ファイルのサイズが小さくなります。実行結果からムービーを削除するには，［**ファイル**］＞［**結果からムービーを削除**］を選択します。

キャプチャしたムービー・ファイルのエクスポート

キャプチャした画面レコーダのムービーは，ファイルにエクスポートできます。このファイルは **.fbr** ファイルとして保存されます。**.fbr** ファイルは HP Micro Recorder で表示できます（703 ページ「HP Micro Player での画面レコーダ・ムービー・ファイルの表示」を参照）。Quality Center で **.fbr** ファイルを不具合に添付することもできます。Quality Center の QuickTest アドインをインストールしている Quality Center ユーザは，Quality Center からムービーを表示できます。

画面レコーダのムービーをエクスポートするには，次の手順を実行します。

1 ［**ファイル**］＞［**ムービーをファイルにエクスポート**］を選択します。［名前を付けて保存］ダイアログ・ボックスが開きます。必要に応じて，標準設定の保存先フォルダおよびファイル名を変更できます。標準設定では，ファイルは＜**コンポーネント名**＞**[** ＜**実行結果名**＞**]** と命名され，実行結果フォルダに保存されます。

2 ［**保存**］をクリックしてエクスポートされたファイル（**.fbr**）を保存し，ダイアログ・ボックスを閉じます。

HP Micro Player での画面レコーダ・ムービー・ファイルの表示

画面レコーダを使用して実行セッションのムービーをキャプチャした場合，ムービーはテスト結果フォルダに **.fbr** ファイルとして保存されます。**.fbr** ファイルはファイル・システム内の任意の場所にエクスポートできます（703 ページ「キャプチャしたムービー・ファイルのエクスポート」で説明されています）。これらの **.fbr** ファイルは，QuickTest の［テスト結果］ウィンドウを開かずに，HP Micro Player を使用して表示することもできます。

HP Micro Player で画面レコーダのムービーを再生するには，次の手順を実行します。

1 次のいずれかを実行します。

- ➤ Windows エクスプローラ内で任意の **.fbr** ファイルをダブルクリックします。
- ➤ **.fbr** ファイルを選択するには，［**スタート**］＞［**プログラム**］＞［**QuickTest Professional**］＞［**Tools**］＞［**HP Micro Player**］を選択し，Micro Player で［**ファイル**］＞［**開く**］を選択します。

HP Micro Player でムービーが開き，再生が始まります。

2 ウィンドウの最上部にあるコントロールを使用すると，ムービー内の特定の場所にアクセスしたり，音量の設定を変更したりできます。

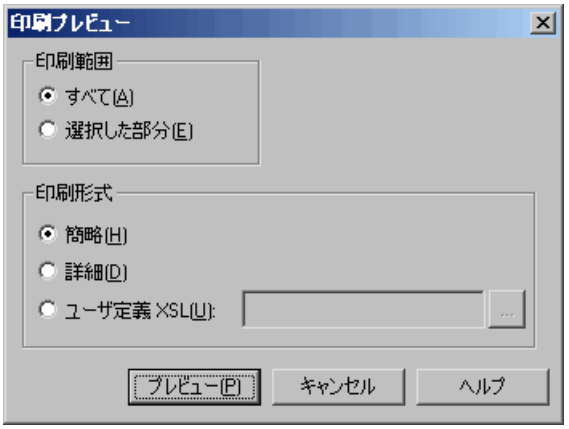
テスト結果のプレビュー

実行結果は，印刷する前に画面に表示できます。表示する情報の形式や範囲を選択できます。また，情報をユーザ定義形式で表示できます。

注：［**印刷プレビュー**］オプションは，QuickTest バージョン 8.0 以降で作成した実行結果にのみ使用できます。

実行結果をプレビューするには，次の手順を実行します。

1 ［**ファイル**］＞［**印刷プレビュー**］を選択します。［印刷プレビュー］ダイアログ・ボックスが開きます。



2 ［**印刷範囲**］オプションを選択します。

- ［**すべて**］：コンポーネント全体の実行結果をプレビューします。
- ［**選択した部分**］：テスト結果ツリーで選択した分岐の実行結果情報をプレビューします。

3 ［**印刷形式**］オプションを選択します。

- ［**簡略**］：実行結果ツリーの各項目のサマリ行（使用可能な場合）を印刷します。このオプションは，手順 2 で［**すべて**］を選択した場合にのみ使用できます。

- ［**詳細**］：手順 2 の選択に従って，実行結果ツリーの各項目，または選択した分岐の利用可能な情報をすべてプレビューします。プレビューには，実行結果のステップに関連する静止画像が含まれます。ビットマップ・チェックポイント・ステップに期待されるビットマップ，実際のビットマップ，および差異ビットマップが表示される場合は，これらも含まれます。
- ［**ユーザ定義 XSL**］：ユーザ定義の **.xsl** ファイルを参照したり選択したりできます。プレビューに含める情報やその表示形式を指定するユーザ定義の **.xsl** ファイルを作成できます。詳細については，719 ページ「テスト結果の表示のカスタマイズ」を参照してください。

4 ［**プレビュー**］をクリックし，画面に実行結果のプレビューを表示します。

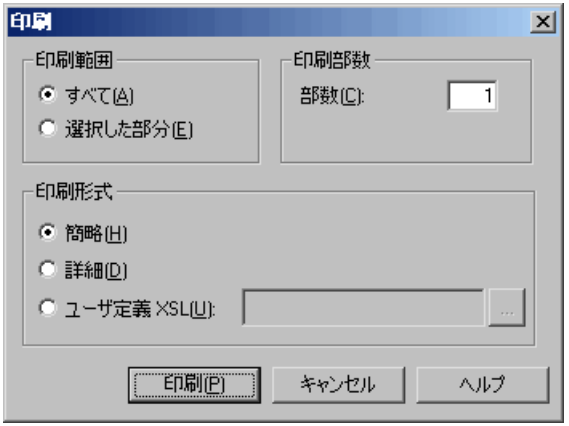
ヒント：プレビューに表示されない情報がある場合は（たとえば，チェックポイント名が長すぎてディスプレイに表示されないなど），［印刷プレビュー］ウィンドウの［**ページの設定**］ボタンをクリックして，ページの向きを［**縦**］から［**横**］に変更します。

実行セッション 結果の印刷

［テスト結果］ウィンドウから，実行結果を印刷できます。印刷するレポートのタイプを選択できます。また，ユーザ定義のレポートの作成や印刷も行えます。

実行結果を印刷するには，次の手順を実行します。

1 ［**印刷**］ボタンをクリックするか，［**ファイル**］＞［**印刷**］を選択します。［印刷］ダイアログ・ボックスが開きます。



2 ［**印刷範囲**］オプションを選択します。

- ［**すべて**］：コンポーネント全体の結果を印刷します。
- ［**選択した部分**］：実行結果ツリーで選択した分岐の実行結果情報を印刷します。

3 ［**印刷部数**］で印刷する実行結果の部数を指定します。

4 ［**印刷形式**］オプションを選択します。

- ［**簡略**］：実行結果ツリーの各項目のサマリ行（使用可能な場合）を印刷します。簡略レポートには，実行結果のステップに関連する静止画像は含まれません。このオプションは，手順 2 で［**すべて**］を選択した場合にのみ使用できます。

- ［**詳細**］：手順 2 の選択に従って，実行結果ツリーの各項目，または選択した分岐の利用可能な情報をすべて印刷します。印刷したレポートには，実行結果のステップに関連する静止画像が含まれます。ビットマップ・チェックポイント・ステップに期待されるビットマップ，実際のビットマップ，および差異ビットマップが表示される場合は，これらも印刷されたレポートに含まれます。
- ［**ユーザ定義 XSL**］：ユーザ定義の **.xsl** ファイルを参照したり選択したりできます。印刷するレポートに含める情報やその表示形式を指定するユーザ定義の **.xsl** ファイルを作成できます。詳細については，719 ページ「テスト結果の表示のカスタマイズ」を参照してください。

注：［**印刷形式**］オプションは，QuickTest バージョン 8.0 以降で作成した実行結果にのみ使用できます。

---

5 ［**印刷**］をクリックして，選択した実行結果情報を標準の Windows プリンタに出力します。

実行結果のエクスポート

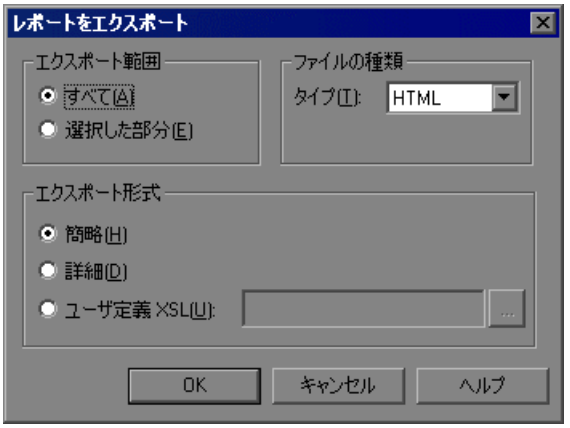
実行結果の詳細を HTML，PDF，または DOC ファイルにエクスポートできます。これにより，QuickTest 環境を使用できなくても，実行結果を表示できます。たとえば，実行結果を収めたファイルを，QuickTest をインストールしていない第三者に電子メールで送信できます。エクスポートするレポートの形式を選択できます。また，ユーザ定義のレポートの作成やエクスポートも行えます。実行結果をエクスポートする場合，［結果の詳細］タブの情報がレポートに含まれます。［画面レコーダ］の結果をエクスポートするには，これらのタブで特定のエクスポート・オプションを使用します。詳細については，701 ページ「実行セッションのムービーの表示」を参照してください。

ファイルの種類を選択するときに，特に多数の画像を含むレポートの場合には，さまざまなドキュメント・タイプを生成するために必要な時間について考慮してください。HTML ファイルが最も短時間に生成でき，PDF と DOC は HTML ファイルより時間がかかります。100 以上の画像を含むレポートを DOC ファイルにエクスポートする場合，ファイルを生成するのに長い時間がかかることを告げるダイアログ・ボックスが表示されます。このダイアログ・ボックスには，画像を含めてエクスポートするオプション，画像なしでエクスポートするオプション，および PDF にエクスポートするオプションが表示されます。

Web アプリケーションのステップを含むテスト結果をエクスポートする場合，これらのステップの画面キャプチャ画像はファイルにエクスポートされません。この理由は，Web ベースのアプリケーションでは，テスト結果ビューアには，キャプチャした画像ではなく関連する Web ページに対応する HTML（ダウンロードした画像を含む）が表示されるため，画像はレポートに保存されないためです。

実行結果をエクスポートするには，次の手順を実行します。

1 ［**ファイル**］＞［**レポートをエクスポート**］を選択します。［レポートをエクスポート］ダイアログ・ボックスが表示されます。



2 ［**エクスポート範囲**］オプションを選択します。

- ［**すべて**］：コンポーネント全体の結果をエクスポートします。
- ［**選択した部分**］：実行結果ツリーで選択した分岐の実行結果情報をエクスポートします。

3 ［**ファイルの種類**］を［**タイプ**］リストから選択します。

注：DOC 形式を使用するには，Microsoft Word 2000 以降がインストールされている必要があります。

4［**エクスポート形式**］オプションを選択します。

- ［**簡略**］：実行結果ツリーの各項目のサマリ行（使用可能な場合）をエクスポートします。簡略レポートには，実行結果のステップに関連する静止画像は含まれません。このオプションは，手順 2 で［**すべて**］を選択した場合にのみ使用できます。
- ［**詳細**］：手順 2 の選択に従って，実行結果ツリーの各項目，または選択した分岐の利用可能な情報をすべてエクスポートします。詳細レポートには，実行結果のステップに関連する静止画像が含まれます。ビットマップ・チェックポイント・ステップに期待されるビットマップ，実際のビットマップ，および差異ビットマップが表示される場合は，これらも印刷されたレポートに含まれます。
- ［**ユーザ定義 XSL**］：ユーザ定義の **.xsl** ファイルを参照したり選択したりできます。エクスポートするレポートに含める情報やその表示形式を指定するユーザ定義の **.xsl** ファイルの作成が可能です。詳細については，719 ページ「テスト結果の表示のカスタマイズ」を参照してください。

注 :［**エクスポート形式**］オプションは，QuickTest 8.0 以降で作成した実行結果にのみ使用できます。

5［**OK**］をクリックします。［名前を付けて保存］ダイアログ・ボックスが表示されます。標準設定では，ファイルは＜コンポーネント名＞［＜実行結果名＞］という名前が付けられ，実行結果フォルダに保存されます。必要に応じて，標準設定の保存先フォルダおよびファイル名を変更できます。

6［**保存**］をクリックしてファイルを保存し，ダイアログ・ボックスを閉じます。

実行結果の削除

テスト結果削除ツールを使い，定義した条件に従って，システムから不要な結果や古い実行結果を削除できます。これによって貴重なディスク・スペースを解放できます。

このツールは Windows スタイルのユーザ・インタフェースで使用することも，Windows コマンド・ラインを使用してバックグラウンドで（非対話的に）実行して，指定する条件に合う結果を直接削除できます。

テスト結果削除ツールを使った結果の削除

テスト結果削除ツールを使用して，ファイル・システムまたは Quality Center プロジェクト内の特定の場所にある，すべての実行セッション結果の一覧を表示できます。その後不要な実行結果を削除できます。

削除対象の実行結果をより簡単に特定できるように，テスト結果削除ツールでは，テスト結果を名前，日付，サイズなどで並べ替えることができます。

テスト結果削除ツールを使って実行結果を削除するには，次の手順を実行します。

1 ［**スタート**］メニューから，［**スタート**］＞［**プログラム**］＞［**QuickTest Professional**］＞［**Tools**］＞［**Test Results Deletion Tool**］を選択します。［テスト結果削除ツール］ウィンドウが開きます。

2 [**テストまたはフォルダ**] ボックスでは，実行結果を削除するパスを指定します。コンポーネントで作業する場合は，フォルダを指定することはできません。

コンポーネントを参照するには，[**参照**] ボタンの横にある下向き矢印をクリックし，[**テスト**] または [**フォルダ**] を選択します。表示されるダイアログ・ボックスのサイドバーで，削除する実行結果の場所を選択します。削除する特定の実行結果を参照して選択し，[**開く**] をクリックします。

注：実行結果を Quality Center データベースから削除するには，パスを参照または入力する前に，[**接続**] をクリックして Quality Center に接続します。標準の Quality Center 形式で Quality Center テスト・パスを指定します。次に例を示します。[Quality Center] Subject¥ ＜フォルダ名＞¥ ＜テスト名＞。この Quality Center プロジェクトに対して**実行の削除**権限があることを確認してください。

Quality Center への接続の詳細については，46 ページ「Quality Center プロジェクトへの接続」を参照してください。

Quality Center プロジェクトの権限の詳細については，Quality Center 管理者に問い合わせるか，または『**HP Quality Center Administrator Guide**』（英語版）で権限の設定に関する項を参照してください。

---

3 指定したフォルダのサブフォルダに含まれるすべての実行結果を表示する場合は，[**サブフォルダで検出されたテスト結果を含める**] を選択します。

注：[**サブフォルダで検出されたテスト結果を含める**] チェック・ボックスは，ファイル・システムのフォルダにのみ使用できます。Quality Center のテストが対象の場合にはサポートされません。

指定したテストの実行結果が，それぞれの情報とともに [テスト結果] ボックスに表示されます。[テスト結果] ボックスのカラム・タイトルをクリックすると，当該カラムのエントリに基づいてテスト結果の順番が並べ替えられます。並び順を逆にするには，同じカラム・タイトルをもう一度クリックします。

[テスト結果の削除] ウィンドウのステータス・バーには，表示されたテスト結果に関する情報が表示されます。具体的には，選択した結果の数，指定した場所にある結果の総数，ファイル・サイズなどが含まれます。

4 削除する実行結果を選択します。Windows の標準的な選択操作の方法を使用して，削除する実行結果を複数選択できます。

5 ［**削除**］をクリックします。選択した実行結果が，システムと Quality Center データベースから削除されます。

ヒント：［**更新**］をクリックすれば，いつでも［テスト結果］ボックスに表示されるテスト結果の一覧を更新できます。

Windows コマンド・ラインを使用した結果の削除

Windows コマンド・ラインを使用して，テスト結果削除ツールに対して，指定の条件に従ってテスト結果を削除するよう指示できます。たとえば，特定の日付より古いテスト結果や，最小ファイル・サイズよりも大きいテスト結果を常に削除するようにできます。

コマンド・ラインからテスト結果削除ツールを実行するには，次の手順を実行します。

Windows のコマンド・プロンプトを開き，＜QuickTest インストール先パス＞¥bin¥TestResultsDeletionTool.exe と入力します。次にスペースを入力してから，使用するコマンド・ライン・オプションを入力します。

注： -Silent コマンド・ライン・オプションを使用してテスト結果削除ツールを実行すると，指定した条件に合うテスト結果がすべて削除されます。このオプションを指定しない場合，［テスト結果削除ツール］ウィンドウが開きます。

コマンド・ライン・オプション

コマンド・ライン・オプションを使用して，削除するテスト結果の条件を指定できます。以下に各コマンド・ライン・オプションについて説明します。

注：スペースを含むコマンド・ライン・オプションを追加する場合，オプションを引用符で囲んで指定する必要があります。たとえば，次のように指定します。TestResultsDeletionTool.exe -Test "F:¥Tests¥Keep¥web objects"

-Domain ＜ Quality Center ドメイン名＞

接続先となる Quality Center ドメイン名を指定します。このオプションは，-Server, - Project，-User，-Password の各オプションと組み合わせて使う必要があります。

-FromDate ＜結果作成日＞

指定の日付より後に作成されたテスト結果を削除します。この日付またはそれよりも前に作成されたテスト結果は削除されません。日付の形式は MM/DD/YYYY です。

次の例では，2005 年 11 月 1 日より後に作成された結果がすべて削除されます。

TestResultsDeletionTool.exe -Silent -Test "C:¥tests¥test1" -FromDate "11/1/2005"

-Log ＜ログ・ファイルのパス＞

指定したフォルダまたはコンポーネント内の各テスト結果ファイルに関するエントリを含むログ・ファイルを作成します。ログ・ファイルは，削除されたテスト結果と，削除されなかった場合にはその理由を示します。たとえば，ファイル・サイズが指定の最小ファイル・サイズよりも小さいためにテスト結果が削除されないことが考えられます。

ファイル・パスと名前を指定することも，標準設定のパスと名前を使用することも可能です。ファイル名を指定しない場合は，標準設定のログ・ファイル名は，テスト結果削除ツールがあるフォルダ内の **TestResultsDeletionTool.log** となります。

次の例では，ログ・ファイルは **C:¥temp¥Log.txt** に作られます。

```
TestResultsDeletionTool.exe  -Silent -Log "C:¥temp¥Log.txt" -Test
"C:¥tests¥test1"
```

次の例では，ログ・ファイルは **TestResultsDeletionTool.log** という名前で，テスト結果削除ツールがあるフォルダ内に作られます。

```
TestResultsDeletionTool.exe  -Silent -Log  -Test "C:¥tests¥test1"
```

-MinSize ＜最小ファイル・サイズ＞

指定した最小ファイル・サイズより大きいか同じサイズのテスト結果を削除します。サイズをバイト単位で指定します。

注：-MinSize オプションは，ファイル・システムのテスト結果に対してのみ使用できます。Quality Center のコンポーネントが対象の場合はサポートされません。

次の例では，10,000 バイト以上の結果をすべて削除します。ファイル・サイズが 10,000 バイトより小さいテスト結果は削除されません。

```
TestResultsDeletionTool.exe  -Silent -Test "C:¥tests¥test1" -MinSize "10000"
```

-Name ＜結果ファイル名＞

削除する結果ファイルの名前を指定します。指定された名前のテスト結果のみが削除されます。

正規表現を使用して，削除する結果ファイルの条件を指定できます。正規表現および正規表現構文の詳細については，617 ページ「正規表現の概要と使い方」を参照してください。

次の例では，**Res1** という名前の結果が削除されます。

```
TestResultsDeletionTool.exe  -Silent -Test "C:¥tests¥test1" -Name "Res1"
```

次の例では，**Res** で始まる 4 文字の名前を持つ結果がすべて削除されます （たとえば，名前が **Res1** や **ResD** ならば削除されます。**ResDD** ならば削除されません)。

```
TestResultsDeletionTool.exe  -Silent -Test "C:¥tests¥test1" -Name "Res."
```

-Password ＜ Quality Center パスワード＞

Quality Center ユーザ名のパスワードを指定します。このオプションは，-Domain，-Server，-Project，-User オプションと組み合わせて使う必要があります。

次の例では，**http://QCServer/qcbin** にあるサーバを使用して，**Default** Quality Center ドメインに接続します。プロジェクト名は **Quality Center_Demo**，ユーザ名は **Admin**，パスワードは **PassAdmin** です。

```
TestResultsDeletionTool.exe  -Domain "Default" -Server "http://QCServer/qcbin"
-Project "Quality Center_Demo" -User "Admin" -Password "PassAdmin"
```

-Project ＜ Quality Center プロジェクト名＞

接続先となる Quality Center プロジェクト名を指定します。このオプションは，-Domain，-Server，-User，-Password の各オプションと組み合わせて使う必要があります。

-Recursive

指定したシステム・フォルダおよびサブフォルダのテスト結果をすべて削除します。-Recursive オプションを使用する場合は，-Test オプションに，（特定のテストのパスではなく）削除するテスト結果があるフォルダのパスが含まれている必要があります。

次の例では，**F:¥Tests** フォルダとそのすべてのサブフォルダの結果をすべて削除します。

```
TestResultsDeletionTool.exe  -Test "F:¥Tests" -Recursive
```

注：-Recursive オプションは，ファイル・システムのフォルダにのみ使用できます。Quality Center のコンポーネントが対象の場合はサポートされません。

-Server ＜ Quality Center サーバのパス＞

接続する Quality Center サーバの完全パスを指定します。このオプションは，-Domain, - Project，-User，-Password の各オプションと組み合わせて使う必要があります。

-Silent

テスト結果削除ツールに，ユーザ・インタフェースなしで，バックグラウンドで（非対話的に）実行するよう指示します。

次の例では，テスト結果削除ツールを非対話的に実行して，**C:¥tests¥test1** にある結果をすべて削除します。

```
TestResultsDeletionTool.exe  -Silent -Test "C:¥tests¥test1"
```

-Test ＜コンポーネントまたはフォルダのパス＞

テスト結果削除ツールがテスト結果を削除するコンポーネントまたはコンポーネント・パスを設定します。コンポーネント名，パス，ファイル・システムのパス，または Quality Center の完全パスを指定できます。

このオプションは必ず -Silent オプションと組み合わせて使用します。

注： Quality Center に接続する場合は，-Domain，-Server，-Project，-User，および -Password オプションを使用する必要があります。

次の例では，**F:¥Tests¥Keep¥webobjects** フォルダにある結果のリストを使用して，テスト結果削除ツールを開きます。

```
TestResultsDeletionTool.exe -Test "F:¥Tests¥Keep¥webobjects"
```

次の例では，Quality Center の **Tests¥webobjects** テストの結果をすべて削除します。

```
TestResultsDeletionTool.exe -Domain "Default" -Server "http://QCServer/qcbin"-Project "Quality Center_Demo592" -User "Admin" -Password "PassAdmin"-Test "Subject¥Tests¥webobjects"
```

ヒント：-Test オプションと -Recursive オプションを組み合わせて，指定したファイル・システム・フォルダとそのすべてのサブフォルダのテスト結果をすべて削除できます。

-UntilDate ＜テスト結果作成日＞

指定した日付より前に作成されたテスト結果を削除します。この日付またはそれよりも後に作成された結果は削除されません。日付の形式は MM/DD/YYYY です。

このオプションは必ず -Silent オプションと組み合わせて使用します。

次の例では，2005 年 11 月 1 日よりも前に作成された結果がすべて削除されます。

```
TestResultsDeletionTool.exe  -Silent -Test "C:¥tests¥test1" -UntilDate "11/1/2005"
```

-User ＜ Quality Center ユーザ名＞

接続先となる Quality Center プロジェクトのユーザ名を指定します。このオプションは，-Domain, - Server, - Project，-Password の各オプションと組み合わせて使う必要があります。

このオプションは必ず -Silent オプションと組み合わせて使用します。

Quality Center プロジェクトへの手動による実行セッション中に検出された不具合の送信

実行セッションの結果の表示中は，検出された不具合を，［テスト結果］ウィンドウから Quality Center プロジェクトに直接送信できます。

Quality Center および QuickTest での作業の詳細については，『**HP QuickTest Professional ユーザーズ・ガイド**』を参照してください。Quality Center の詳細については，『**Mercury Quality Center ユーザーズ・ガイド**』を参照してください。

不具合を Quality Center に直接送信するには，次の手順を実行します。

1 Quality Center クライアントがコンピュータにインストールされていることを確認します（ブラウザに Quality Center サーバの URL を入力し，ログイン画面が表示されることを確認します）。

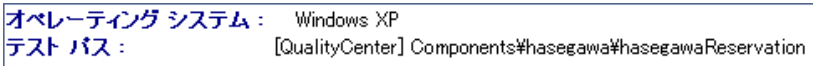
2 ［**ツール**］＞［**Quality Center への接続**］を選択するか，［**Quality Center への接続**］ボタンをクリックして，Quality Center プロジェクトに接続します。Quality Center への接続の詳細については，46 ページ「Quality Center プロジェクトへの接続」を参照してください。

注：次のステップに進む前に Quality Center プロジェクトに接続していない場合は，QuickTest によってステップを続行する前に接続するかどうかを尋ねるメッセージが表示されます。

---

3 ［**ツール**］＞［**不具合の追加**］を選択するか，［**不具合の追加**］ボタンをクリックして，指定した Quality Center プロジェクトの［新規不具合］ダイアログ・ボックスを開きます。［新規不具合］ダイアログ・ボックスが表示されます。

4 必要に応じて不具合の情報を変更できます。コンポーネントに関する基本情報は詳細領域に含まれています。



5 ［**送信**］をクリックし，Quality Center プロジェクトに不具合情報を追加します。

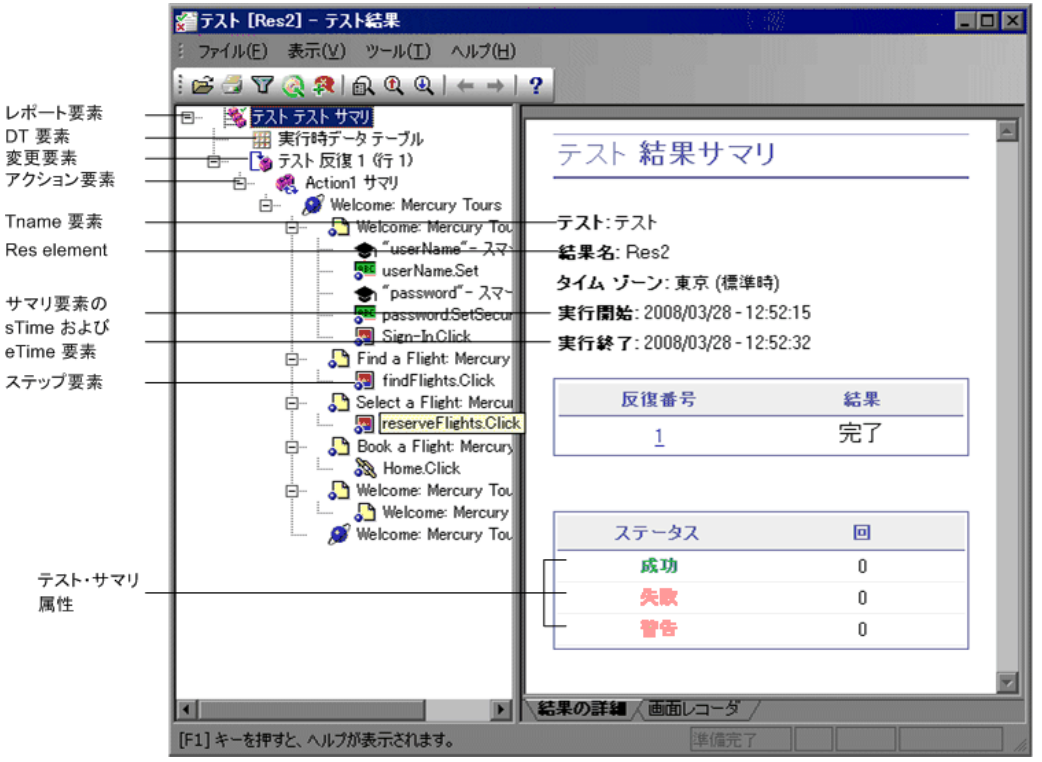
6 ［**閉じる**］をクリックし，［不具合の追加］ダイアログ・ボックスを閉じます。

テスト結果の表示のカスタマイズ

QuickTest 実行セッションの結果はそれぞれ 1 つの **.xml** ファイル（**results.xml** と呼びます）に保存されます。この **.xml** ファイルには，ディスプレイのそれぞれのテスト結果ノードに関する情報が保存されます。これらのノードの情報は，［テスト結果］ウィンドウの右上の表示枠に表示される **.htm** ファイルを動的に作成するのに使用されます。

実行結果ツリーの各ノードは **results.xml** ファイルの要素です。また，テスト結果に表示される異なるタイプの情報を表す要素もあります。**.xml** ファイルのテスト結果情報に基づき，XSL を使用して必要な情報をカスタマイズされた形式で表示できます（QuickTest の［テスト結果］ウィンドウから印刷する場合，テスト結果を独自にカスタマイズした結果ビューアで表示する場合，テスト結果を HTML ファイルにエクスポートする場合のいずれにおいても可能）。

次の図は，**.xml** ファイル内のいくつかの要素と，それらがテスト結果の中で表す項目の対応関係を示します。



ヒント：［テスト結果］ウィンドウの外観（ルック・アンド・フィール）は変更できます。詳細については，688 ページ「［テスト結果］ウィンドウの外観の変更」を参照してください。

XSL には，どのテスト結果情報を表示するか，それをどこにどのように表示，印刷，エクスポートするかを指定するための手段が用意されています。また，**.xsl** ファイルが参照する **.css** ファイルを修正して，レポートの外観（フォント，色など）を変更できます。

たとえば，**results.xml** ファイルで，ある要素タグにはコンポーネント名が，もう 1 つの要素タグには実行セッションが開始される時間についての情報が含まれているとします。XSL を使用すれば，カスタマイズしたテスト結果ビューアに，コンポーネント名をページの特定の位置に緑色の太字で表示し，時間の情報はまったく表示しないよう指示できます。

QuickTest が提供する既存の **.xsl** ファイルと **.css** ファイルに変更を加えるほうが，カスタマイズしたファイルを最初から作成するよりも簡単です。このファイルは**＜ QuickTest のインストール・フォルダ＞ ¥dat** にあり，名前は次のとおりです。

- **PShort.xsl**：［印刷］ダイアログ・ボックスまたは［HTML ファイルにエクスポート］ダイアログ・ボックスで［**簡略**］オプションを選択した場合に印刷または HTML ファイルにエクスポートされるテスト結果レポートの内容を指定します。
- **PDetails.xsl**：［印刷］ダイアログ・ボックスまたは［HTML ファイルにエクスポート］ダイアログ・ボックスで［**詳細**］オプションを選択した場合に印刷または HTML ファイルにエクスポートされるテスト結果レポートの内容を指定します。
- **PResults.css**：テスト結果の印刷プレビューの外観を指定します。このファイルは 3 つの **.xsl** ファイルすべてによって参照されています。

カスタマイズした **.xsl** ファイルを使用したテスト結果の印刷については，706 ページ「実行セッション 結果の印刷」を参照してください。

カスタマイズした **.xsl** ファイルを使用したファイルへのテスト結果のエクスポートの詳細については，707 ページ「実行結果のエクスポート」を参照してください。

XML スキーマの構造の詳細，テスト結果レポートをカスタマイズするために使用できる要素と属性の説明については，XML レポート・ヘルプ（［**ヘルプ**］＞［**QuickTest Professional ヘルプ**］＞［**QuickTest Advanced References**］＞［**QuickTest Test Results Schema**］）を参照してください。

第 28 章

実行セッション結果の分析

実行セッション中に発生した重要なイベントに関するレポートを使用して，実行セッションの結果を分析できます。

注：QuickTest から［テスト結果］ウィンドウを開いた場合は，ビジネス・プロセス・テストの実行結果は表示できません。ビジネス・プロセス・テストのテスト結果を表示するには，表示する反復の結果を選択して Quality Center から開きます。

本章の内容

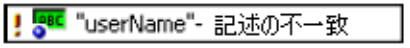
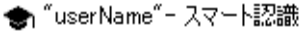
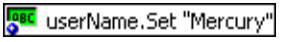
- テスト結果に含まれるスマート認識情報の分析（723 ページ）
- チェックポイントの結果の表示（727 ページ）
- ［テスト結果］ウィンドウでのパラメータ化された値と出力値の結果の表示（733 ページ）

テスト結果に含まれるスマート認識情報の分析

学習した記述では QuickTest が指定されたオブジェクトを一度で認識できず，しかもスマート認識定義がオブジェクトに対して定義されている（そして有効である）場合には，QuickTest はスマート認識メカニズムを使ってオブジェクトの認識を試みます。次の例で 2 つのシナリオを説明します。

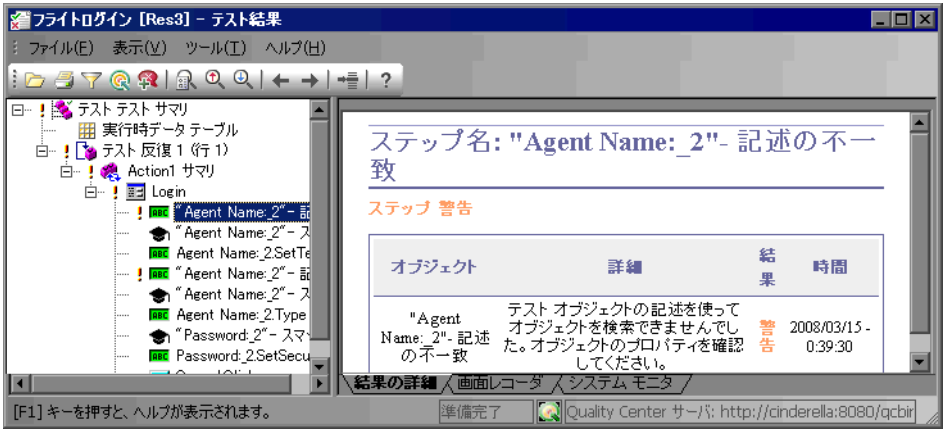
スマート認識 ― 学習した記述に一致するオブジェクトがない場合

QuickTest において，学習した記述に一致するオブジェクトが見つからず，スマート認識によってオブジェクトが見つかった場合，テスト結果に警告ステータスおよび次の情報が表示されます。

結果ツリー	結果の詳細
見つからなかったオブジェクトに対する記述不一致アイコン。例を次に示します。 	オブジェクト（たとえば userName WebEdit オブジェクト）が見つからなかったことを示します。
見つからなかったオブジェクトに対するスマート認識アイコン。例を次に示します。 	スマート認識メカニズムがオブジェクトを見つけるのに成功したことと，オブジェクトを見つけるために使われたプロパティについての情報を示します。この情報に基づいて，学習したテスト・オブジェクト記述を変更し，QuickTest が以降の実行セッションでその記述を使ってオブジェクトを見つけられるようにできます。
実際に実行されたステップ。例を次に示します。 	実行されたステップの詳細を示す通常の結果。

スマート認識メカニズムの詳細については，第 6 章「オブジェクト認識の設定」を参照してください。

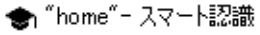
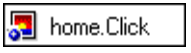
次の図に，学習した記述プロパティ値の 1 つが変更された後で，スマート認識を使って AgentName WebEdit オブジェクトを認識したコンポーネントの結果を示します。



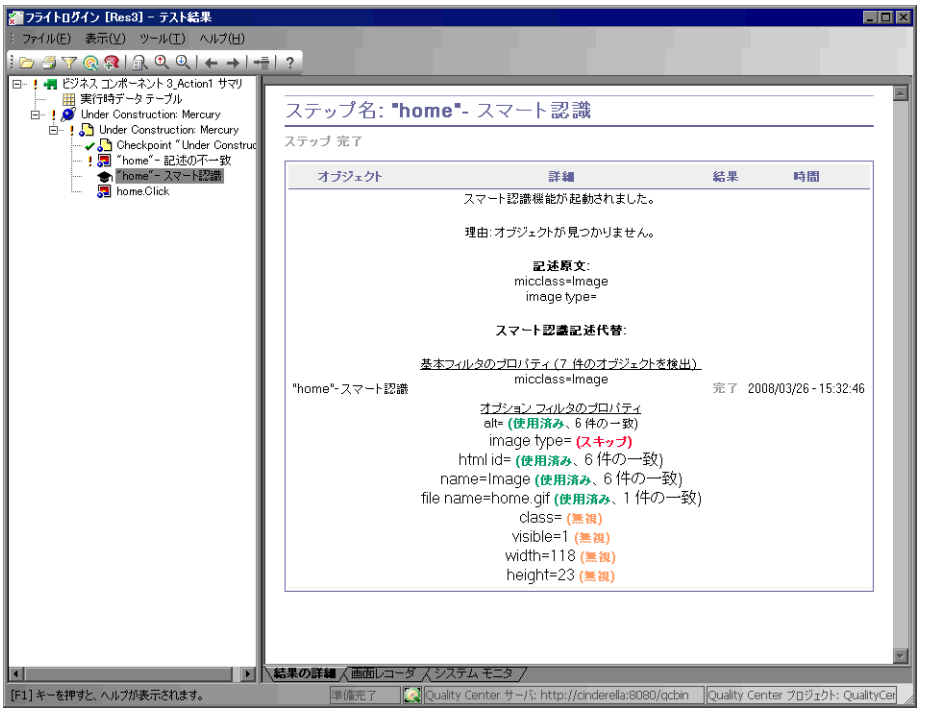
スマート認識 ― 学習した記述に複数のオブジェクトが一致する場合

学習した記述に一致する複数のオブジェクトが見つかった後で，QuickTest がスマート認識を使って 1 つのオブジェクトを見つけた場合，QuickTest は［テスト結果］ウィンドウにスマート認識情報を示します。その場合でもステップは合格となります。ほとんどの場合，スマート認識を使用しなくても，テスト・オブジェクトの記述と序数識別子の組み合わせでオブジェクトを認識できたはずだからです。

そのような場合，テスト結果には次の情報が示されます。

結果ツリー	結果の詳細
見つからなかったオブジェクトに対するスマート認識アイコン。例を次に示します。 	スマート認識メカニズムがオブジェクトを見つけるのに成功したことと，オブジェクトを見つけるために使われたプロパティについての情報を示します。この情報に基づいて，そのオブジェクトのための一意のオブジェクト記述を作成し，QuickTest がその記述を使って以降の実行セッションでオブジェクトを見つけられるようにできます。
実際に実行されたステップ。例を次に示します。 	実行されたステップの詳細を示す通常の結果。

次の図に，学習した記述が変更された結果，複数のオブジェクトが適合するようになった後で，スマート認識を使って Home オブジェクトを一意に認識したコンポーネントの結果を示します。



スマート認識メカニズムでうまくオブジェクトを認識できない場合，コンポーネントは失敗し，［テスト結果］に通常の失敗ステップが表示されます。

チェックポイントの結果の表示

コンポーネントにチェックポイントを追加すると，Web ページ，テキスト文字列，オブジェクトのプロパティなどの期待値を，アプリケーション内のこうした要素の値と比較できます。これによって，アプリケーションが期待どおりに機能していることを確認できます。

コンポーネントを実行すると，QuickTest によって，チェックポイントの期待結果と現在の結果が比較されます。この 2 つの結果が一致しないと，チェックポイントは失敗となり，コンポーネントは失敗します。チェックポイントの結果は［テスト結果］ウィンドウで見ることができます。

チェックポイントの結果を表示するには，次の手順を実行します。

1 ［テスト結果］ウィンドウでコンポーネントのテスト結果を表示します。詳細については，689 ページ「実行セッションの結果の表示」を参照してください。

2 ［テスト結果］ウィンドウの左側の表示枠で，実行結果ツリーの分岐を展開して，結果を表示するチェックポイント の分岐をクリックします。チェックポイントの結果が［テスト結果］ウィンドウに表示されます。

注：標準設定では，［テスト結果］ウィンドウの［結果の詳細］タブの下の表示枠には，選択したチェックポイントに**失敗**のステータスが含まれる場合のみ，このチェックポイントに関する情報が表示されます。ステップの画像を保存する条件を，［オプション］ダイアログ・ボックスの［実行］＞［画面キャプチャ］表示枠で変更できます。詳細については，642 ページ「［オプション］ダイアログ・ボックス：［実行］＞［画面キャプチャ］表示枠」を参照してください。

［テスト結果］ウィンドウと使用可能なオプションについての情報は，選択したチェックポイントのタイプによって決定されます。詳細については，次の情報を参照してください。

➤ 728 ページ「標準チェックポイントの結果の分析」

➤ 730 ページ「ビットマップ・チェックポイント結果の分析」

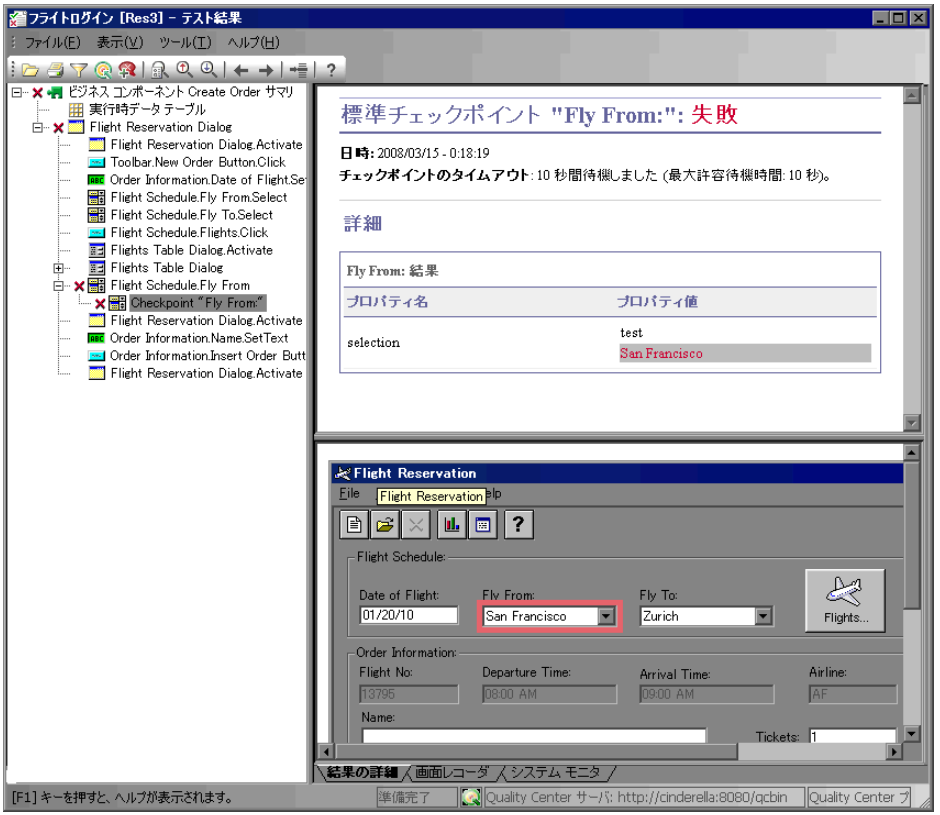
3 ［**ファイル**］＞［**終了**］を選択して，［テスト結果］ウィンドウを閉じます。

チェックポイントの詳細については，第 18 章「チェックポイントについて」を参照してください。

標準チェックポイントの結果の分析

標準チェックポイントをコンポーネントに追加すると，オブジェクト・プロパティの期待値を，実行セッション中のオブジェクトの現在値と比較できます。この 2 つの結果が一致しないと，そのチェックポイントは失敗となります。標準チェックポイントの詳細については，567 ページ「標準チェックポイントを使ってオブジェクト・プロパティ値の検査」を参照してください。

標準チェックポイントの詳細な結果は［テスト結果］ウィンドウで見ることができます。チェックポイントの結果の表示については，727 ページ「チェックポイントの結果の表示」を参照してください。



［結果の詳細］タブの上の表示枠には選択したチェックポイントの詳細結果が表示されます。ここにはチェックポイントのステータス（**成功**または**失敗**），チェックポイントが実行された日時，使用されたチェックポイント・タイムアウトの間隔（必要に応じて）などが表示されます。検査対象オブジェクトのプロパティ値も表示されます。また，期待プロパティ値と実際のプロパティ値に違いがあればそれらもすべて表示されます。

下部の表示枠には，使用可能ならばチェックポイント・ステップの画像キャプチャが表示されます。

前の例では，失敗したチェックポイントの詳細によって，期待結果と現在の結果が一致しないことが示されています。エージェント名の期待値は「**Agent2**」ですが，実際の値は「**Agent1**」です。

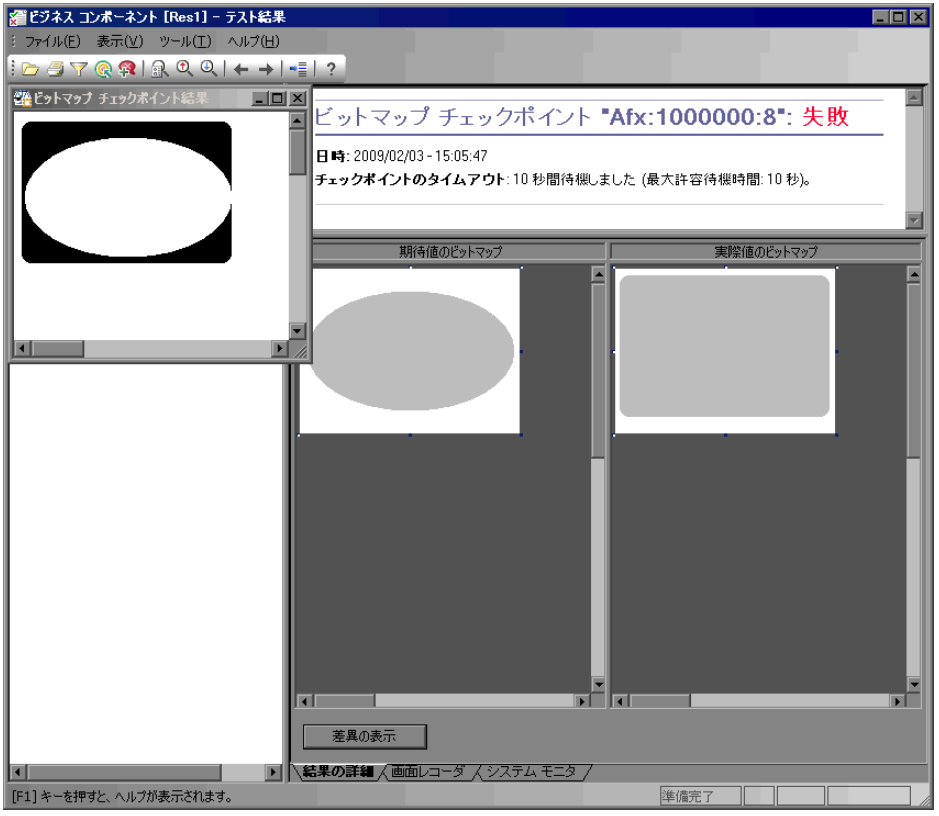
ビットマップ・チェックポイント結果の分析

コンポーネントにビットマップ・チェックポイントを追加して，キャプチャしたビットマップを照合することによって，アプリケーション内の要素の外観を検査できます。コンポーネントを実行する場合，QuickTest によって，チェックポイントに保存されている期待されるビットマップが，実行セッション中にアプリケーションからキャプチャされた実際のビットマップと比較されます。ビットマップが一致しないと，そのチェックポイントは失敗となります。ビットマップ・チェックポイントの詳細については，第 20 章「ビットマップの検査」を参照してください。

ビットマップ・チェックポイントの詳細な結果は［テスト結果］ウィンドウで見ることができます。チェックポイントの結果の表示については，727 ページ「チェックポイントの結果の表示」を参照してください。

［結果の詳細］タブの上の表示枠にはチェックポイント・ステップの結果が表示されます。ここにはチェックポイントのステータス（**成功**または**失敗**），チェックポイントが実行された日時，使用されたチェックポイント・タイムアウトの間隔（必要に応じて）などが表示されます。

［結果の詳細］タブの下の表示枠には，実行セッション中に比較された期待されるビットマップと実際のビットマップ，および［**差異の表示**］ボタンが表示されます。［**差異の表示**］ボタンをクリックすると，［ビットマップ チェックポイント結果］ウィンドウが開き，期待されるビットマップと実際のビットマップの差異を示す画像が表示されます。この画像は白黒のビットマップで，2 つの画像間で異なるピクセルは黒色で示されます。



注：標準設定では，ビットマップ・チェックポイントに失敗した場合のみ，下の表示枠の情報を利用できます。［オプション］ダイアログ・ボックスの［実行］＞［画面キャプチャ］表示枠で［**静止画像キャプチャをテスト結果へ保存**］オプションを使用することによって，ビットマップをいつテスト結果に保存するかに関する条件を変更できます。詳細については，642 ページ「［オプション］ダイアログ・ボックス：［実行］＞［画面キャプチャ］表示枠」を参照してください。

ビットマップ・チェックポイントの結果を確認する際の注意事項

- チェックポイントをビットマップの特定の領域だけを比較するように定義すると，テスト結果には，選択した領域が強調表示された状態で実際のビットマップと，期待されるビットマップが表示されます。
- 実際のビットマップと期待されるビットマップの寸法が異なる場合は，QuickTest によってビットマップは比較されることなく，チェックポイントは失敗します。この場合，［**差異の表示**］機能を結果で利用することはできません。
- 10.00 以前のバージョンの QuickTest で生成した結果を表示する場合は，［**差異の表示**］機能は利用できません。
- ビットマップ・チェックポイントを独自の比較ツールで実行する場合は，次の点に注意してください。
  - 寸法が異なる場合でも，QuickTest によってビットマップが独自の比較ツールに渡されて比較されます。
  - ［結果の詳細］タブの上の表示枠には，独自の比較ツールの名前（［ビットマップ チェックポイント プロパティ］ダイアログ・ボックスの［**Comparer**］ボックスに表示されている名前），および独自の比較ツールによって提供される追加情報も表示されます。
  - 差異ビットマップは独自の比較ツールによって提供されます。

ビットマップ・チェックポイントに独自の比較ツールを使用する方法の詳細については，580 ページ「ビットマップ比較の微調整」を参照してください。

［テスト結果］ウィンドウでのパラメータ化された値と出力値の結果の表示

パラメータ化された値に関する情報と出力値ステップの結果を［テスト結果］ウィンドウに表示できます。

［テスト結果］ウィンドウでのパラメータ化された値の表示

「**パラメータ**」とは，コンポーネント内から値が割り当てられる変数です。［テスト結果］ウィンドウにコンポーネントで定義されたパラメータの値を表示できます。

パラメータ化された値を表示するには，次の手順を実行します。

1 コンポーネントの実行結果を［テスト結果］ウィンドウに表示します。詳細については，689 ページ「実行セッションの結果の表示」を参照してください。

2 ［テスト結果］ウィンドウの左側の表示枠で，コンポーネントの名前を含んだルート・ノードを選択します。

テストの場合もコンポーネントの場合も，入力パラメータの名前と値は，右側の表示枠の下部に表示されます。

前の例は，値「**Mercury**」を使ってコンポーネント用に定義された入力パラメータ「**UserName**」を示します。

コンポーネントでのパラメータの定義と使用については，542 ページ「パラメータの使用」を参照してください。

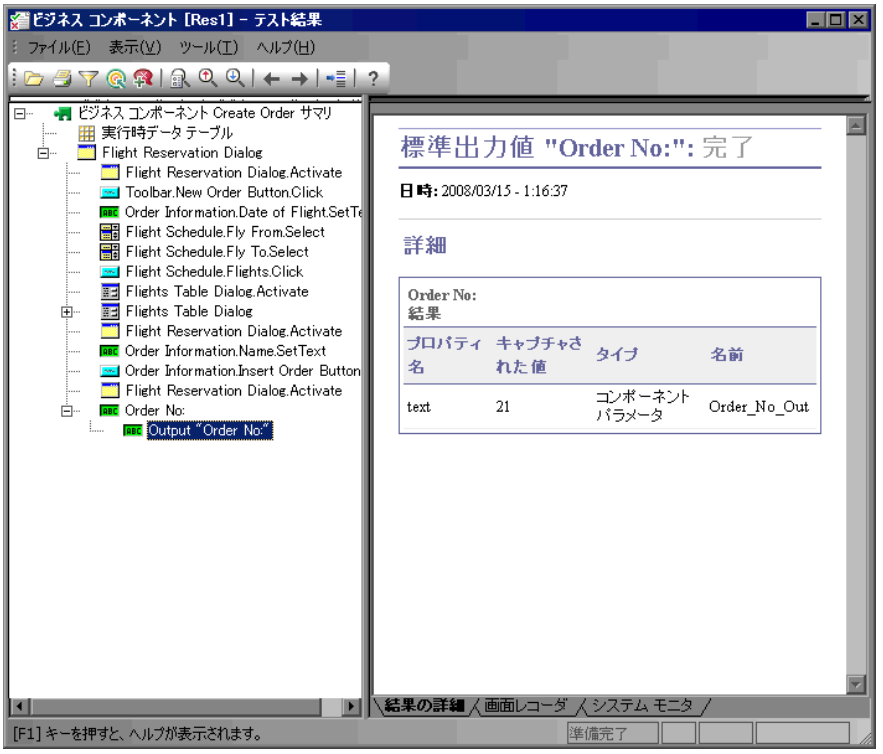
［テスト結果］ウィンドウでの出力値の結果の表示

「**出力値**」とは，実行セッション中に 1 つ以上の値がキャプチャされるステップです。キャプチャされた値は，実行内の別の時点で使用されます。実行中，値の 1 つが入力として必要とされた場合に，QuickTest によって，その値が指定された場所から取得されます。

出力値のステップの結果を表示するには，次の手順を実行します。

1 コンポーネントの実行結果を［テスト結果］ウィンドウに表示します。詳細については，689 ページ「実行セッションの結果の表示」を参照してください。

2 ［テスト結果］ウィンドウの左側の表示枠で，実行結果ツリーの分岐を展開して，結果を表示したい出力値のステップの分岐をクリックします。出力値の結果が［テスト結果］ウィンドウに表示されます。



右側の表示枠には，ステータス，出力値のステップが実行された日時を含む選択された出力値の詳細な結果が表示されます。また，実行セッション中にキャプチャされた値，そのタイプおよび名前など，出力値の詳細も表示されます。

出力値の詳細については，第 21 章「値の出力」を参照してください。

第 VIII 部

コンポーネントのデバッグと保守

第 29 章

コンポーネントと関数ライブラリのデバッグ

実行セッションを制御しデバッグすることで，コンポーネント，関数ライブラリ，および登録されたユーザ関数内の問題を特定して処理できます。

注：QuickTest でコンポーネントをデバッグするには，QuickTest と Quality Center プロジェクトの間の統合を有効にする必要があります（QuickTest で，［**ツール**］＞［**オプション**］＞［**実行**］ノードを選択し，［**他の HP 製品でテストおよびコンポーネントを実行可能にする**］チェック・ボックスを選択します。

本章の内容

- コンポーネントと関数ライブラリのデバッグについて（740 ページ）
- デバッグ・セッションの速度調節（742 ページ）
- シングル・ステップ・コマンドの使用（742 ページ）
- ［ステップまで実行］コマンドおよび［ステップからデバッグ］コマンドの使用（746 ページ）
- 実行セッションの一時停止（748 ページ）
- ブレークポイントの使用（748 ページ）
- ［デバッグ ビューア］表示枠（751 ページ）
- 実行エラーの処理（762 ページ）
- 関数のデバッグの練習（764 ページ）

コンポーネントと関数ライブラリのデバッグについて

コンポーネントまたは関数ライブラリ（登録済みユーザ関数を含む）の作成が終わったら，構文やロジックにエラーがなく，スムーズに実行されることを確認する必要があります。関数ライブラリをデバッグするには，まず関数ライブラリを（アプリケーション領域を通じて）コンポーネントに関連付け，次にそのコンポーネントから関数ライブラリをデバッグする必要があります。

QuickTest には，コンポーネントまたは関数ライブラリの不具合を検出して切り分けるために使用できるさまざまなオプションが用意されています。次に例を示します。

- ［**一時停止**］コマンド，ブレークポイント，および特定のステップに対してステップ・イントゥ，ステップ・オーバー，ステップ・アウトを実施できる，さまざまなステップ・コマンドを使用して，実行セッションを制御できます。
- また，実行セッション中に，QuickTest によって実行エラーのメッセージが表示された場合，エラー・メッセージの［**デバッグ**］ボタンをクリックし，実行を一時停止して，コンポーネントまたは関数ライブラリをデバッグすることもできます。
- 実行セッションが停止（一時停止）したら，デバッグ・ビューアを使用して，VBScript オブジェクトおよび変数の値を確認して変更したり，VBScript コマンドを手作業で実行したりできます。
- ［**ステップからデバッグ**］コマンドを使用すれば，コンポーネント内の特定のポイントでデバッグ・セッションを開始（および一時停止）できます。また，［**ステップまで実行**］コマンドを使用して，コンポーネント内の特定のポイントで実行を一時停止できます。ブレークポイントの設定後，コンポーネントまたは関数ライブラリのさまざまな部分をデバッグする際にブレークポイントを有効にしたり無効にしたりできます。
- さらに，選択したステップからコンポーネントを実行するために，［**ステップから実行**］コマンドを使用できます。これにより，アプリケーションの特定のセクションだけを検査したり，コンポーネントまたは関数ライブラリの特定の部分がスムーズに実行されるかどうかを確認したりできます。詳細については，『**HP QuickTest Professional ユーザーズ・ガイド**』の「コンポーネントの一部の実行」を参照してください。

ヒント：画面レコーダを使って，テスト中のアプリケーションのムービーをキャプチャできます。詳細については，699 ページ「アプリケーションの静止画像とムービーの表示」を参照してください。

コンポーネントと関数ライブラリのデバッグに関する考慮事項

- デバッグ・モードでコンポーネントを実行するには，Microsoft Script Debugger がインストールされている必要があります。インストールされていない場合は，QuickTest Additional Installation Requirements ユーティリティを使ってインストールできます（[**スタート**] > [**プログラム**] > [**QuickTest Professional**] > [**Tools**] > [**Additional Installation Requirements**] を選択します)。
- コンポーネントおよび関数ライブラリは，デバッグ・モードで実行されているときは読み取り専用です。内容の変更は，デバッグ・セッションの停止後（一時停止時ではなく）にできます。必要に応じて，セッションの停止後に関数ライブラリを編集可能にできます（[**ファイル**] > [**編集可能にする**])。詳細については，398 ページ「読み取り専用の関数ライブラリの編集」を参照してください。変更後，コンポーネントおよび関数ライブラリのデバッグを続行できます。
- ファイル操作を実行すると（たとえば，別のコンポーネントを開いたり，新規のコンポーネントを作成したりするなど)，デバッグ・セッションは停止します。
- QuickTest では，コンポーネントを開くと，Quality Center プロジェクトに保存した外部リソースのローカル・コピーが作成されます。したがって，関数ライブラリなど，Quality Center プロジェクトに保存されている外部リソースに適用した変更は，コンポーネントを閉じて再度開くまでコンポーネントで認識されません（外部リソースとはコンポーネントとは別に保存できるリソースのことで，関数ライブラリ，共有オブジェクト・リポジトリ，または回復シナリオなどがあります)。

デバッグ・セッションの速度調節

実行セッション中，通常 QuickTest はステップを高速で実行します。コンポーネントまたは関数ライブラリのデバッグ中は，必要に応じて実行を一時停止したり別のタスクを実行したりできるように，ステップをもっとゆっくり実行したいときがあります。各ステップ間で QuickTest が一時停止する時間（ミリ秒）を指定できます。指定するには，［オプション］ダイアログ・ボックスの［実行］表示枠（［**ツール**］＞［**オプション**］＞［**実行**］ノード）にある［**ステップ実行ごとの遅延時間**］オプションを変更します。［実行］表示枠のオプションの詳細については，638 ページ「テストの実行オプションの設定」を参照してください。

シングル・ステップ・コマンドの使用

［**ステップ イントゥ**］コマンド，［**ステップ アウト**］コマンド，および［**ステップ オーバー**］コマンドを使用して，コンポーネントまたは関数ライブラリのステップを 1 つだけ実行できます。

ヒント：デバッグ・ツールバーを表示するには，［**表示**］＞［**ツールバー**］＞［**デバッグ**］を選択します。

［ステップ イントゥ］

［**ステップ イントゥ**］では，アクティブなコンポーネントまたは関数ライブラリの現在のステップだけが実行されます。現在のステップが関数を呼び出すと，呼び出された関数が QuickTest ウィンドウに表示され，関数ライブラリは，呼び出された関数の最初の行で一時停止します。

［ステップ イントゥ］コマンドを使用するには，次の手順を実行します。

［**デバッグ**］＞［**ステップ イントゥ**］を選択し，［**ステップ イントゥ**］ボタンをクリックするか，または F11 キーを押します。

[ステップ アウト]

[**ステップ イントゥ**] を使用して関数ライブラリの関数に移動した後で，[**ステップ アウト**] コマンドを使用します。[**ステップ アウト**] では，関数の終了まで実行が継続され，呼び出し元のコンポーネントまたは関数ライブラリに戻り，次の行（存在する場合）で実行セッションを一時停止します。

[ステップ アウト] コマンドを使用するには，次の手順を実行します。

[**デバッグ**] > [**ステップ アウト**] を選択し，[**ステップ アウト**] ボタンをクリックするか，または SHIFT キーを押しながら F11 キーを押します。

[ステップ オーバー]

[**ステップ オーバー**] では，アクティブなコンポーネントまたは関数ライブラリの現在のステップだけが実行されます。現在のステップで，ユーザ定義関数が呼び出されている場合は，呼び出し先の関数の全体が実行されますが，呼び出し先の関数スクリプトは QuickTest ウィンドウに表示されません。次に，実行セッションは呼び出し元のコンポーネントまたは関数ライブラリに戻り，次の行（存在する場合）で一時停止します。

[ステップ オーバー] コマンドを使用するには，次の手順を実行します。

[**デバッグ**] > [**ステップ オーバー**] を選択し，[**ステップ オーバー**] ボタンをクリックするか，または F10 キーを押します。

シングル・ステップ・コマンドの使用 ー 例

［**ステップ イントゥ**］，［**ステップ アウト**］，および［**ステップ オーバー**］コマンドを使用して，サンプルの関数ライブラリを作成し，（コンポーネントから）実行するには，次の手順を実行します。

サンプルの関数ライブラリおよびコンポーネントを作成する	1 ［**ファイル**］＞［**新規作成**］＞［**関数ライブラリ**］を選択して新しい関数ライブラリを開きます。 2 関数ライブラリで，次の行を正確に入力します。 public Function myfunc() msgbox "one" msgbox "two" msgbox "three" End Function 3 **SampleFL.qfl** という名前で，Quality Center プロジェクトに関数ライブラリを保存します（詳細については，394 ページ「関数ライブラリの保存」を参照してください）。 4 ［**ファイル**］＞［**新規作成**］＞［**アプリケーション領域**］を選択して新しいアプリケーション領域を開きます。 5 **SampleAA** という名前を付けて Quality Center プロジェクトにアプリケーション領域を保存します（詳細については，439 ページ「アプリケーション領域の保存」を参照してください）。 6 **SampleFL.qfl** 関数ライブラリのタブをクリックして，フォーカスを対象に移動します。 7 ［**ファイル**］＞［**Associate Library 'SampleFL.qfl' with 'SampleAA'**］を選択して関数ライブラリを開いているアプリケーション領域に関連付けます。 8 ［**ファイル**］＞［**新規作成**］＞［**ビジネス コンポーネント**］を選択して新しいビジネス・コンポーネントを開きます。［**ビジネス コンポーネントの新規作成**］ダイアログ・ボックスが開いたら，コンポーネントをステップ 4 で作成したアプリケーション領域に関連付けます。 9 コンポーネントで，同じステップを 4 つ挿入します。各ステップについて，次を実行します。 ➤［**項目**］セルの中で，［**操作**］を選択します。 ➤［**操作**］セルの中で，［**myfunc**］を選択します。

関数ライブラリをコンポーネントから実行し，[ステップ イントゥ]，[ステップ アウト]，および [ステップ オーバー] コマンドを使用する	10 F9 キー（[**ブレークポイントの設定 / 解除**]）を押して，コンポーネントの最初のステップにブレークポイントを追加します。ブレークポイント記号が左マージンに表示されます。詳細については，749 ページ「ブレークポイントの設定」を参照してください。 11 コンポーネントを実行します。コンポーネントは次のブレークポイントで一時停止します。 12 F11 キー（[**ステップ イントゥ**]）を押します。実行矢印によって，関数ライブラリ内の関数の最初の行（msgbox "one"）がポイントされます。 13 F11 キー（[**ステップ イントゥ**]）をもう一度押します。メッセージ・ボックスに，「one」というテキストが表示されます。 14 ［**OK**］をクリックし，メッセージ・ボックスを閉じます。実行矢印が，関数の次の行へ移動します。 15 F11 キー（[**ステップ イントゥ**]）（および開いたメッセージ・ボックスの［**OK**］）を押し続け，実行矢印が関数から離れてコンポーネントの 2 つ目のステップ（myfunc 関数に対する 2 つ目の呼び出し）をポイントします。 16 F11 キー（[**ステップ イントゥ**]）を押して，もう一度関数を入力します。実行矢印によって，関数の最初の msgbox 行がポイントされます。 17 SHIFT + F11 キー（[**ステップ アウト**]）を押します。開かれている各メッセージ・ボックスを閉じます。3 つのうち最後のメッセージ・ボックスを閉じるまで，実行矢印は関数の最初の行をポイントし続けます。3 つ目のメッセージ・ボックスを閉じると，実行矢印はコンポーネントの次の行をポイントします（myfunc 関数に対する 3 つ目の呼び出し）。 18 F10 キー（[**ステップ オーバー**]）を押します。3 つのメッセージ・ボックスが再び開きます。このときは，［キーワード ビュー］で開きます。3 つのメッセージ・ボックスのうち最後のメッセージ・ボックスを閉じるまで，実行矢印はコンポーネントの同じステップをポイントし続けます。3 つめのメッセージ・ボックスを閉じると，実行矢印はコンポーネントの次のステップをポイントします。

［ステップまで実行］コマンドおよび［ステップからデバッグ］コマンドの使用

デバッグ中は，ステップに対するステップ・イントゥ，ステップ・アウト，ステップ・オーバーのほかに，［**ステップまで実行**］コマンドと［**ステップからデバッグ**］コマンドを使って，特定のステップに達するまでコンポーネント（関連付けられた関数ライブラリを含む）を実行するように，または特定のステップからデバッグを開始するように QuickTest に指示できます。

［ステップまで実行］

コンポーネントの最初から，あるいはコンポーネントの現在の位置から実行するように指定できます。また，特定のステップで停止するように指示することもできます。これは，ステップに一時的なブレークポイントを追加するのに似ています。たとえば特定のステップからコンポーネントのデバッグを開始する場合，そのステップまでコンポーネントを実行し，アプリケーションの関連箇所を開く必要がある場合があります。

［**ステップまで実行**］オプションを使用すると，コンポーネントの編集中に実行セッションを開始したり，または一時停止されている実行セッションを再開したりできます。

特定のステップまで実行するように QuickTest に指示するには，次のいずれかを実行します。

- テストにおいて，実行を停止するステップにカーソルを置き，［**デバッグ**］＞［**ステップまで実行**］を選択するか，CTRL キーを押しながら F10 キーを押します。
- テストにおいて，実行を停止するステップを右クリックし，ショートカット・メニューから［**ステップまで実行**］を選択します。

注 : ［**ステップまで実行**］オプションを使用して新しい実行セッションを開始すると，［実行］ダイアログ・ボックスが開き，デバッグ実行セッションの結果の格納場所と入力パラメータ値を指定できます。詳細については，次の「［ステップからデバッグ］」の項で 1 および 2 を参照してください。

［ステップからデバッグ］

コンポーネントの最初から実行を開始するのではなく，特定のステップからデバッグ・セッションを開始するように指定できます。特定のステップからデ

バッグを開始する場合は，その前に，デバッグを開始する位置に合わせてアプリケーションが開いていることを確認してください。コンポーネントの編集中にコンポーネント内の特定のステップからデバッグを開始できます。

特定のステップから実行するように指定するには，次の手順を実行します。

1 デバッグを開始するステップを選択します。

- ➤ 実行を開始するステップにカーソルを置き，［**デバッグ**］＞［**ステップからデバッグ**］を選択します。
- ➤ 実行を開始するステップを右クリックし，コンテキスト・メニューから［**ステップからデバッグ**］を選択します。

［実行］ダイアログ・ボックスが開きます。［実行］ダイアログ・ボックスのタブの詳細については，674 ページ「[実行] ダイアログ・ボックス：[結果の場所] タブ」および 676 ページ「[実行] ダイアログ・ボックス：[入力パラメータ] タブ」を参照してください。

2 必要に応じて，デバッグ実行セッションの結果の格納場所と入力パラメータの値を指定します。標準設定では，［**一時実行結果フォルダ**］オプションが選択されます。

3 ［**OK**］をクリックします。［実行］ダイアログ・ボックスが閉じ，デバッグ実行セッションが開始されます。［**ステップ イントゥ**］，［**ステップ オーバー**］，［**ステップまで実行**］などの任意の QuickTest デバッグ・オプションを使用できます。

標準設定では，実行セッションが終了すると，［テスト結果］ウィンドウが開きます。実行結果の表示の詳細については，第 27 章「実行セッション結果の表示」を参照してください。［オプション］ダイアログ・ボックスの［実行］表示枠で［**実行セッション終了時に結果を表示する**］チェック・ボックスをクリアすると，実行セッションの最後に［テスト結果］ウィンドウは開きません。［オプション］ダイアログ・ボックスの詳細については，第 24 章「グローバル・テスト・オプションの設定」を参照してください。

実行セッションの一時停止

実行セッションを一時的に停止するには，［**デバッグ**］＞［**一時停止**］を選択するか，［**一時停止**］ボタンをクリックします。一時停止されたコンポーネントまたは関数ライブラリは，それまでに解釈されたすべてのステップが実行された後に実行を停止します。

一時停止した実行を再開するには，［**実行**］ボタンをクリックするか，［**オートメーション**］＞［**実行**］を選択するか，または **F5** キーを押します。一時停止されたところから，実行が再開されます。

ヒント：実行セッションは，［**停止**］ボタンをクリックし，［**オートメーション**］＞［**停止**］を選択したり，または［停止］コマンド・ショートカット・キー（［**ツール**］＞［**オプション**］＞［**実行**］ノードで定義）を押して停止することもできます。実行セッションの最後に結果を表示しないように選択していなければ（［**ツール**］＞［**オプション**］＞［**実行**］ノード），実行セッションの停止後，［テスト結果］ウィンドウが開きます。

ブレークポイントの使用

ブレークポイントを使用すると，コンポーネントまたは関数ライブラリ内のあらかじめ定義した位置で実行セッションを一時停止するよう QuickTest に指示できます。ブレークポイントに達すると，ステップを実行する前に実行が一時停止されます。ブレークポイントまでの実行の結果を検証し，必要であれば変更を加えた上で，ブレークポイントからコンポーネントまたは関数ライブラリの実行を再開できます。ブレークポイントは，現在の QuickTest セッションにのみ適用され，コンポーネントまたは関数ライブラリには保存されません。

ブレークポイントは次のような目的に使用できます。

- 実行セッションを一時停止し，アプリケーションの状態を点検する。
- ステップ・コマンドを使ってコンポーネントまたは関数ライブラリをステップ実行する開始位置を指定する。

ブレークポイントを設定し，そのブレークポイントを一時的に有効にしたり無効にしたりできます。ブレークポイントの使用が終了したら，コンポーネントまたは関数ライブラリからブレークポイントを削除できます。

ブレークポイントの設定

ブレークポイントを設定すると，実行セッションをコンポーネントまたは関数ライブラリ内のあらかじめ定義した位置で実行セッションを一時停止できます。ブレークポイントは，選択したステップの横の左マージンに赤い丸のアイコンで示されます。

ブレークポイントを設定するには，次のいずれかを実行します。

- コンポーネントまたは関数ライブラリ内で，実行を停止するステップの左マージンをクリックします。
- ステップをクリックし，次のいずれかを実行します。

  - ［**ブレークポイントの設定 / 解除**］ボタンをクリックします。
  - ［**デバッグ**］＞［**ブレークポイントの設定 / 解除**］を選択します。
  - ［**デバッグ**］＞［**ブレークポイントの有効化 / 無効化**］を選択します。
  - F9 キーを押します。

ブレークポイント記号 が，選択したステップに隣接する左マージンに表示されます。

ブレークポイントの有効化と無効化

ブレークポイントを一時的に無効にすることによって，デバッグ・セッション中に既存のブレークポイントを無視するように指定できます。指定した場合，コンポーネントまたは関数ライブラリの実行時，QuickTest はブレークポイントが含まれるステップで停止せず，そのステップを実行します。ブレークポイントを再度有効にすると，次の実行時にそこで一時停止します。この機能は，コンポーネントまたは関数ライブラリに多くのステップがあり，その特定部分をデバッグする必要がある場合に特に役立ちます。

ブレークポイントの有効化および無効化は，個別に，またはすべて一度に行うことができます。たとえば，コンポーネントまたは関数ライブラリ全体にわたってさまざまなステップにブレークポイントを追加し，なおかつ当面はテスト・ドキュメントの特定の部分のみデバッグしたいとします。この場合，コンポーネントまたは関数ライブラリ内のすべてのブレークポイントを無効にし，それから特定ステップのブレークポイントのみ有効にします。ドキュメントの対象セクションのデバッグが済んだら，有効にしたブレークポイントを無効にし，（デバッグしたいセクションの）次のブレークポイントのセットを有効にします。ブレークポイントは無効にされていても削除はされていないため，任意のブレークポイントを必要に応じて探して有効にできます。

有効化なブレークポイント：有効になっているブレークポイントは，選択したステップの左マージンに赤い丸のアイコン で示されます。

無効なブレークポイント：無効になっているブレークポイントは，選択したステップの横の左マージンに白抜きの丸のアイコン で示されます。

特定のブレークポイントを有効または無効にするには，次の手順を実行します。

1 有効または無効にするブレークポイントが含まれているステップをクリックします。

2 ［**デバッグ**］＞［**ブレークポイントの有効化 / 無効化**］を選択するか，CTRL キーを押しながら F9 キーを押します。ブレークポイントが有効または無効になります（それまでの状態に応じて異なります）。

すべてのブレークポイントを有効または無効にするには，次の手順を実行します。

［**デバッグ**］＞［**全ブレークポイントの有効化 / 無効化**］を選択するか，［**全ブレークポイントの有効化 / 無効化**］ボタンをクリックします。少なくとも 1 つのブレークポイントが有効になっている場合は，コンポーネントまたは関数ライブラリ内のすべてのブレークポイントが無効になります。逆に，すべてのブレークポイントが無効になっている場合は，すべてのブレークポイントが有効になります。

ブレークポイントの削除

現在のコンポーネントまたは関数ライブラリに定義されている単一のブレークポイントまたはすべてのブレークポイントを削除できます。

単一のブレークポイントを削除するには，次のいずれかを実行します。

➤ ステップの左マージンでブレークポイント・アイコンをクリックします。

➤ コンポーネントまたは関数ライブラリ内のブレークポイント記号のあるステップをクリックし，次の操作を行います。

➤ ［**ブレークポイントの設定 / 解除**］ボタンをクリックします。

➤ ［**デバッグ**］＞［**ブレークポイントの設定 / 解除**］を選択します。

➤ F9 キーを押します。

テスト・ドキュメントの左マージンから，ブレークポイント記号が削除されます。

すべてのブレークポイントを削除するには，次の手順を実行します。

［**すべてのブレークポイントを削除**］ボタンをクリックするか，［**デバッグ**］＞［**すべてのブレークポイントを削除**］を選択します。テスト・ドキュメントの左マージンから，すべてのブレークポイント記号が削除されます。

［デバッグ ビューア］表示枠

説明	実行セッションが一時停止されている場合，次のいずれかのアクティビティを実行できます。 ➤ または関数ライブラリのオブジェクトまたは変数の現在値を表示，設定，または修正する。 ➤ 一時停止されている実行セッションで VBScript コマンドを実行する。 デバッグ・ビューアは，ビジネス・コンポーネント内の操作（関数）のデバッグに役立ちますが，ほかのタイプのコンポーネント・ステップで使用することを目的としたものではありません。
使用方法	［**表示**］＞［**デバッグ ビューア**］メニュー・コマンドを選択します。
重要な情報	次の状況の場合は，実行セッションを一時停止できます。 ➤ 実行セッションがブレークポイントで停止する。 ➤ ［**デバッグ**］メニュー・コマンドまたはツールバーのボタン（［**一時停止**］や［**ステップまで実行**］など）を使用して実行セッションを一時停止する。 ➤ ステップが失敗して［**デバッグ**］オプションを選択する。
詳細	**概念の概要**：740 ページ「コンポーネントと関数ライブラリのデバッグについて」 **主要なタスク**：764 ページ「関数のデバッグの練習」

［デバッグ ビューア］表示枠のタブ

［デバッグ ビューア］表示枠には，次のタブがあります。

- **［ウォッチ］タブ**：変数の現在値と種類，および［ウォッチ］タブに追加した VBScript 式が表示され，表示されている変数およびプロパティの値を変更できます。詳細については，752 ページ「[デバッグ ビューア］表示枠：［ウォッチ］タブ」を参照してください。
- **［変数］タブ**：メイン・スクリプトまたは選択したサブルーチン内にあるすべての変数の現在値および種類が表示され，これらの値を変更できます。詳細については，757 ページ「[デバッグ ビューア］表示枠：［変数］タブ」を参照してください。
- **［コマンド］タブ**：一時停止されている実行セッションで VBScript コマンドを実行できます。詳細については，760 ページ「[デバッグ ビューア］表示枠：［コマンド］タブ」を参照してください。

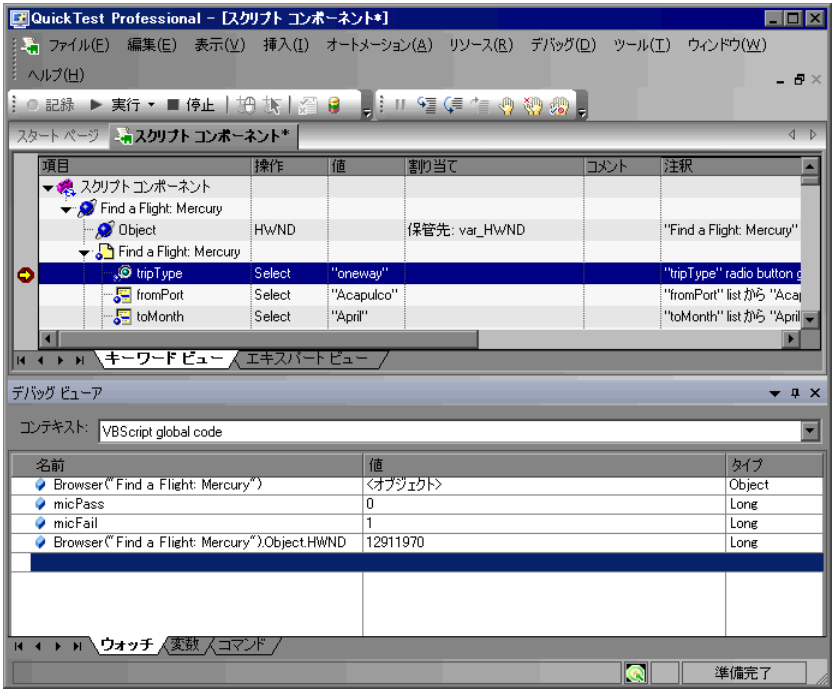
［デバッグ ビューア］表示枠：［ウォッチ］タブ

説明	実行セッションを一時停止すると，このタブに選択した変数の現在値と種類，プロパティ，および関数ライブラリの VBScript 式を表示できます。 また，このタブを使用して変数またはプロパティの値を手作業で変更することもできます。
使用方法	［**表示**］メニュー＞［**デバッグ ビューア**］項目＞［**ウォッチ**］タブ
詳細	**主要なタスク**：755 ページ「[デバッグ ビューア］表示枠の［ウォッチ］タブの使用」 **その他の関連項目**： ➤ 751 ページ「[デバッグ ビューア］表示枠」 ➤ 764 ページ「関数のデバッグの練習」

［デバッグ ビューア］表示枠の［ウォッチ］タブの画像を次に示します。

この画像には，コンポーネントのステップを実行する前に一時停止された実行セッションが表示されます。このため，［**コンテキスト**］ボックスには VBScript global code という文字列が表示され，［ウォッチ］タブに表示される値は，一時停止されたコンポーネントのコンテキスト内で評価されています。

［ウォッチ］タブに表示できる式の種類を一部見ることができます（**Find a Flight: Mercury** Browser オブジェクトの **HWND** ネイティブ・プロパティなど）。その他の種類およびコンテキストについては，757 ページ「[デバッグ ビューア］表示枠：［変数］タブ」に示す画像を参照してください。



デバッグ・ビューアの［ウォッチ］タブの詳細

項目	説明
［**コンテキスト**］ボックス	［ウォッチ］タブに表示されている式を評価するコンテキストを示します。 ➤ 実行セッションが，コンポーネントのステップを実行する前に一時停止された場合，［**コンテキスト**］ボックスには，VBScript global code という文字列が表示され，［ウォッチ］タブに表示される式は，一時停止されたコンポーネントのコンテキスト内で評価されます。 ➤ 実行セッションが関数ライブラリ内で一時停止された場合は，［**コンテキスト**］ボックスに，実行が一時停止された関数の名前が最初に表示され，同じ関数ライブラリ内のほかの関数およびサブルーチンのコンテキストに切り替えることができます。 ［ウォッチ］タブに表示される式は，選択した関数またはサブルーチンのコンテキスト内で評価されます。
［**名前**］カラム	確認対象の値を保持する VBScript 式。［ウォッチ］タブで式を追加および削除する方法の詳細については，755 ページ「［デバッグ ビューア］表示枠の［ウォッチ］タブの使用」を参照してください。 **警告**：QuickTest では，式を評価するために［ウォッチ］タブの式が実行されます。このため，評価することによってテスト・オブジェクトの状態が影響を受ける可能性があるテスト・オブジェクト・メソッドまたは式を入力しないでください。関数ライブラリで，予期しない動作が発生する可能性があります。

［**値**］カラム	式の現在の値。評価された値は，実行セッションが一時停止されたときのみ表示されます。 また，このカラムでは確認中の変数またはプロパティの値を設定または変更できます。 たとえば，［ウォッチ］タブに表示されている実行環境オブジェクト・プロパティの値を編集できます。このため，実行セッションを再開する前に，テスト中のアプリケーションのプロパティ値を変更できます。 ［ウォッチ］タブからオブジェクトの認識プロパティの実行時の値を変更することはできません。
［**タイプ**］カラム	評価後の式の値の種類（**Integer** や **String** など）。 現在のコンテキストで式を評価できない場合，表示される種類は［**エラー**］です（［**名前**］カラムのアイコンでも示されます）。

［デバッグ ビューア］表示枠の［ウォッチ］タブの使用

［ウォッチ］タブに VBScript 式を追加すると，関数ライブラリの実行セッションで異なる変数の現在値およびオブジェクトのプロパティを表示できます。実行セッションが一時停止される場合（たとえば，［**デバッグ**］＞［**一時停止**］コマンドを使用する場合，または関数ライブラリがブレークポイントで停止する場合），［ウォッチ］タブには，現在値およびタブに追加した式の種類が表示されます。

関数ライブラリ内の後続のステップを引き続き実行すると，自動的に［ウォッチ］タブが更新され，値が変更された式の現在値が表示されます。

また，このタブを使用して変数またはプロパティの値を手作業で変更することもできます。たとえば，［ウォッチ］タブに表示されている実行環境オブジェクト・プロパティの値を編集できます。このため，実行セッションを再開する前に，テスト中のアプリケーションのプロパティ値を変更できます。詳細については，752 ページ「［デバッグ ビューア］表示枠：［ウォッチ］タブ」を参照してください。

重要：QuickTest では，式を評価するために［ウォッチ］タブの式が実行されます。このため，評価することによってテスト・オブジェクトの状態が影響を受ける可能性があるテスト・オブジェクト・メソッドまたは式を追加しないでください。関数ライブラリで，予期しない動作が発生する可能性があります。

［ウォッチ］タブには，次の種類の式を追加できます。

- テスト・オブジェクトの名前
- 変数の名前
- プロパティの名前
- ほかの種類の VBScript 式

注：［ウォッチ］タブに**認識プロパティ**を追加するには，**GetROProperty** を呼び出す式を使用する必要があります。これにより，オブジェクトの認識プロパティの実行時の値を確認できます。たとえば，Calculator アプリケーションに現在表示されている値を確認するには，Window("Calculator").WinEdit("Edit").GetROPRoperty("text") という式を追加します。

［ウォッチ］タブからオブジェクトの認識プロパティの実行時の値を変更することはできません。

［ウォッチ］タブに式を追加するには，次の手順を実行します。

次のいずれかを実行します。

- 式をクリックして，［**デバッグ**］＞［**ウォッチに追加**］を選択します。
- 式をクリックして，CTRL + T キーを押します。
- 式を右クリックして，ショートカット・メニューから［**ウォッチに追加**］を選択します。
- ［ウォッチ］タブでグリッドの空行を選択し，［**名前**］カラムをクリックして式を貼り付けるか入力し，ENTER キーを押します。

注：関数ライブラリから［ウォッチ］タブに式を追加できます（ビジネス・コンポーネントからは追加できません）。

［ウォッチ］タブから式を削除するには，次の手順を実行します。

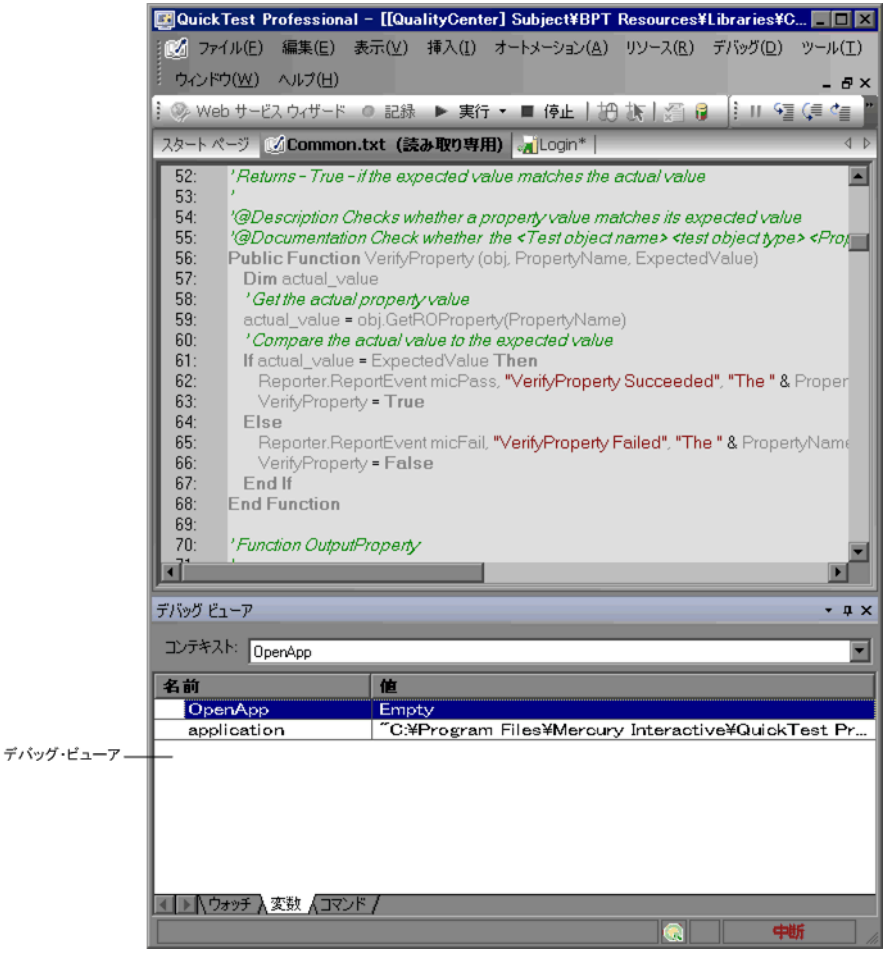
［ウォッチ］タブで削除する行を選択し，キーボードで Delete キーを押します。

［デバッグ ビューア］表示枠：［変数］タブ

説明	実行セッションが一時停止された場合，［変数］タブには関数ライブラリの選択した関数内のすべての変数の現在値および種類が表示され，それらの値を変更できます。
使用方法	［**表示**］メニュー＞［**デバッグ ビューア**］項目＞［**変数**］タブ
重要な情報	実行された最後のステップまでに認識された変数だけが，［変数］タブに表示されます。関数ライブラリ内の後続のステップを引き続き実行すると，QuickTest によって認識された変数が追加され，［変数］タブに表示されている値が更新されます。
詳細	**その他の関連項目**： ➤ 751 ページ「［デバッグ ビューア］表示枠」 ➤ 764 ページ「関数のデバッグの練習」

［デバッグ ビューア］表示枠の［変数］タブの画像を次に示します。

この画像は，関数ライブラリの関数内で一時停止された実行セッションを示しています。このため，［変数］タブには，一時停止された関数のコンテキスト内で定義された変数だけが表示されます。



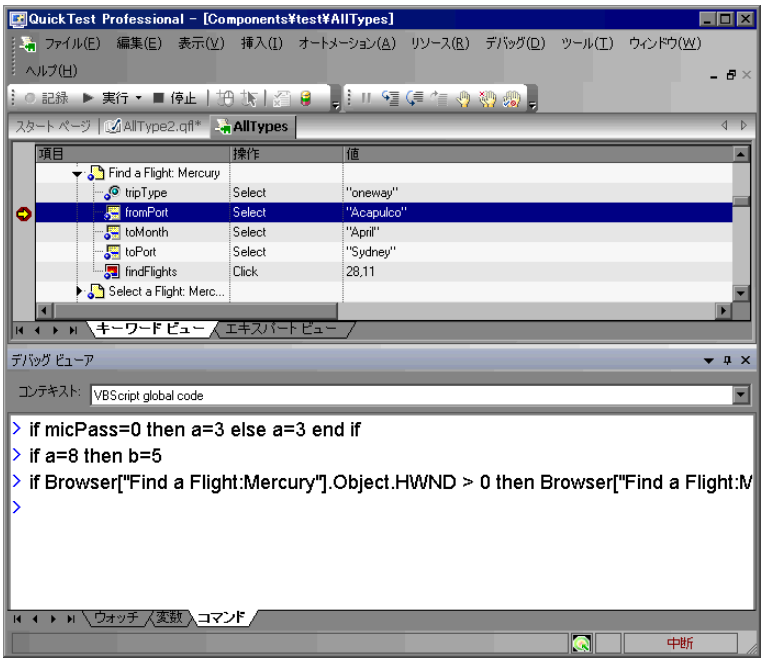
デバッグ・ビューアの［変数］タブの詳細

項目	説明
［**コンテキスト**］ボックス	このタブに表示される変数のコンテキストを示します。 ➤ 実行セッションが，コンポーネントのステップを実行する前に一時停止された場合，［**コンテキスト**］ボックスには，VBScript global code という文字列が表示され，このタブには，一時停止されたコンポーネントのコンテキスト内で定義された変数のみが表示されます。 ➤ 実行セッションが関数ライブラリ内で一時停止された場合は，［**コンテキスト**］ボックスに，実行が一時停止された関数の名前が最初に表示され，同じ関数ライブラリ内のほかの関数およびサブルーチンのコンテキストに切り替えることができます。 選択した関数またはサブルーチンのコンテキスト内で定義された変数だけが，［変数］タブに表示されます。
［**名前**］カラム	変数の名前。
［**値**］カラム	変数の現在の値。この値を編集して，実行セッションを継続する前に変数の値を設定または変更できます。
［**タイプ**］カラム	変数値の種類（**Integer** や **String** など）。

［デバッグ ビューア］表示枠： ［コマンド］タブ

説明	実行セッションが一時停止された場合，このタブでは，関数ライブラリの VBScript コードの行を実行できます。 たとえば，実行セッションを再開する前に，次のアクティビティのいずれかを実行する VBScript コードを実行できます。 ➤ テスト中のアプリケーションから情報を取得する。 ➤ テスト・オブジェクト・メソッドを実行して戻り値を表示し，メソッドの動作について詳しく確認する。 ➤ アプリケーションのネイティブ（実行環境オブジェクト）プロパティの値を変更する。 ➤ アプリケーションのネイティブ（実行環境オブジェクト）メソッドを呼び出す。
使用方法	［**表示**］メニュー＞［**デバッグ ビューア**］項目＞［**コマンド**］タブ
詳細	**その他の関連項目**： ➤ 751 ページ「[デバッグ ビューア］表示枠」 ➤ 764 ページ「関数のデバッグの練習」

［デバッグ ビューア］表示枠の［コマンド］タブの画像を次に示します。



デバッグ・ビューアの［コマンド］タブの詳細

- **［コンテキスト］ボックス**：［ウォッチ］タブおよび［変数］タブに表示される式および変数のコンテキストを示します。
- **コマンド・ライン・プロンプト**：一時停止されている実行セッションのコンテキストで VBScript コードの行を実行できます。プロンプトでコード行を入力または貼り付け，ENTER キーを押してコードを実行します。
- **コマンド・ライン履歴**：実行した VBScript コードの行が表示されます。
  - これらの行に変更を加えることはできませんが，これらの行からテキストを選択してコピーできます。
  - 上下の矢印キーを使用して，コマンド履歴を参照できます。QuickTest によってアクティブなコマンド・ラインにコマンドがコピーされるため，以前に入力したコマンドを繰り返したり再利用したりできます。

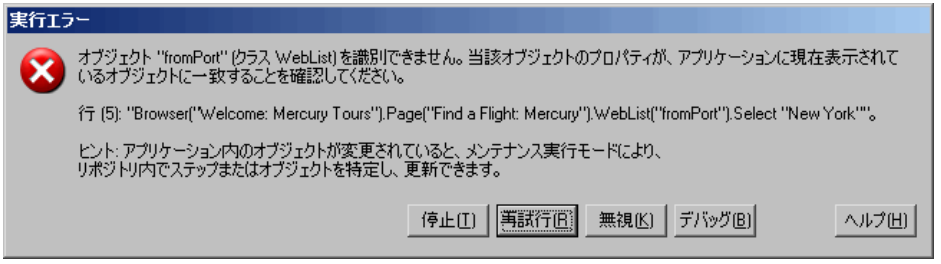
- **右クリックのショートカット・メニュー**：［コマンド］タブの内容の編集に使用できるコマンドが表示されます。
  - ［**切り取り**］，［**コピー**］，および［**貼り付け**］コマンドを使用すると，クリップボードを使用してコマンド履歴からテキストをコピーしたり，アクティブなコマンド・ラインを編集したりできます。
  - ［**すべてクリア**］コマンドを使用すると，コマンド履歴をすべて消去できます。

注：［コマンド］タブにコード行を入力できるのは，実行セッションが一時停止されている場合のみです。実行セッションが一時停止されていない場合は，コマンド履歴を表示してテキストを選択およびコピーし，ショートカット・メニューの［**すべてクリア**］コマンドを使用できます。

実行エラーの処理

実行セッション中に表示できる［実行エラー］メッセージ・ボックスには，2 つのタイプがあります。1 つは，問題が VBScript の純粋な構文エラーである場合に表示されます。構文実行エラー・メッセージが表示された場合は，メッセージ・ボックスの［**OK**］をクリックし，ステップ内のエラーを解決します。

もう 1 つのメッセージ・ボックスは，さまざまな状況で表示されます。このメッセージ・ボックスには，発生したエラーを処理するためにエラーに関する情報と複数のボタンが表示されます。



- ［**停止**］。テスト実行を停止します。実行後に実行結果を表示するように QuickTest が設定されていれば，実行結果が表示されます。
- ［**再試行**］。QuickTest によってステップの実行が再試行されます。ステップが成功すると，実行が再開されます。
- ［**無視**］：QuickTest によって，エラーが発生したステップが無視され，その次のステップから実行が再開されます。
- ［**デバッグ**］。QuickTest によって実行が中断され，コンポーネントおよびコンポーネントから呼び出された関数が含まれる関連する関数ライブラリをデバッグできます。

  本章で説明している任意のデバッグ操作を実行できます。デバッグが完了したら，コンポーネントまたは関数ライブラリが停止したステップから実行セッションを再開したり，ステップ・コマンドを使って残りの実行セッションを制御したりできます。
- ［**ヘルプ**］。表示されたエラー・メッセージに関する QuickTest のトラブルシューティング用ヘルプが表示されます。ヘルプ・トピックを読んだら，メッセージ・ボックスで別のボタンを選択できます。

  メッセージ・ボックスには，アプリケーションの意図的な変更がエラーの原因と考えられ，コンポーネント内の複数のステップやリポジトリ内のオブジェクトを更新する必要がある場合は，メンテナンス・モードの使用を検討するように推奨するメッセージも表示されます。詳細については，770 ページ「メンテナンス実行ウィザードを使ったコンポーネントの実行」を参照してください。

関数のデバッグの練習

関数ライブラリのほかの部分で使用される変数を定義する関数を作成するとします。関数にブレークポイントを追加することで，関数ライブラリの実行中に変数の値がどのように変化するかを確認できます。ブレークポイントで一時停止している間に，いずれかの変数の値を変更することによって，関数ライブラリでその新しい値がどのように処理されるかを調べることもできます。

手順 1：新しい関数を作成する

新しい関数ライブラリを開いて，**SetVariables** という新しい関数を作成します。関数での作業の詳細については，第 12 章「ユーザ定義関数および関数ライブラリを使用した作業」を参照してください。

次のように VBScript コードを入力します。

Expert View

```
Dim a
a="hello"
b="me"
MsgBox a
```

Function Library

```
Function SetVariables

Dim a
a="hello"
b="me"
MsgBox a

EndFunction
```

```
Function SetVariables

Dim a
a="hello"
b="me"
MsgBox a

EndFunction
```

手順 2：関数ライブラリをアプリケーション領域に関連付ける

1 コンポーネントに関連付けられているアプリケーション領域が開いていて，関数ライブラリにフォーカスが当てられていることを確認します。

2 ［**ファイル**］＞［**ライブラリ ' ＜関数ライブラリ名＞ ' を ' ＜アプリケーション領域名＞ ' に関連付ける**］を選択します。QuickTest によって，関数ライブラリがアプリケーション領域に関連付けられます。

手順 3：コンポーネントに関数の呼び出しを追加する

関数の呼び出しを追加するには，新しい操作を挿入し，［**操作**］リストから［**SetVariables**］を選択します。

手順 4：ブレークポイントを追加する

b="me" および MsgBox a というテキストが含まれている行にブレークポイントを追加します。ブレークポイントの追加の詳細については，749 ページ「ブレークポイントの設定」を参照してください。

手順 5：コンポーネントの実行を開始する

コンポーネントを実行します。コンポーネントまたは関数ライブラリは，対象のステップ（スクリプトの行）を実行する前の最初のブレークポイントで停止します。

手順 6：デバッグ・ビューア・ウィンドウで変数の値を検査する

1 ［**表示**］＞［**デバッグ ビューア**］を選択して，［デバッグ ビューア］表示枠を開きます（まだ開いていない場合）。［デバッグ ビューア］表示枠で［**ウォッチ**］タブをクリックします。

2 ドキュメント表示枠で変数 **a** を選択し，［**デバッグ**］＞［**ウォッチに追加**］を選択します。［ウォッチ］タブに変数 a が追加されます。ブレークポイントにより，変数 **a** の値が初期化された後に停止したため，［**値**］カラムには，**a** の値が現在「**hello**」であることが示されます。［**タイプ**］カラムには，**a** が **String** 変数であることが示されます。

3 ドキュメント表示枠で変数 **b** を選択し，［**デバッグ**］＞［**ウォッチに追加**］を選択します。［ウォッチ］タブに変数 b が追加されます。［**値**］カラムには，＜**Variable is undefined: 'b'** ＞と示されます（また，［**タイプ**］カラムには，［**エラー**］と表示されます）。この理由は，コンポーネントが変数 **b** が宣言される前に停止したためです。

4 ［デバッグ ビューア］表示枠で，［**変数**］タブをクリックします。**SetVariables**（値は **Empty**）と **a**（値は）の両方が表示されます。変数 **b** が宣言される前にコンポーネントが停止したため，変数 **b** は表示されません。

手順 7：次のブレークポイントで変数の値を検査する

［**実行**］ボタンをクリックして，コンポーネントの実行を続行します。コンポーネントは次のブレークポイントで停止します。［ウォッチ］タブと［変数］タブの変数 **a** と **b** の値が更新されています。

手順 8：［変数］タブを使って変数の値を変更する

1 ［デバッグ ビューア］表示枠で，［**変数**］タブをクリックします。

2 ［**値**］カラムで，文字列 me を選択し，文字列 you に置き換えてキーボードで ENTER キーを押します。

3 ［**ウォッチ**］タブをクリックします。［ウォッチ］タブで，変数 **b** の値も更新されたことを確認できます。

手順 9：［コマンド］タブを使って変数の値を変更する

4 ［デバッグ ビューア］ウィンドウで，［**コマンド**］タブをクリックします。

5 コマンド・プロンプトで，if b="me" then a="b is me" else a="b is you" end if と入力し，キーボードの ENTER キーを押します。

6 ［**変数**］タブをクリックして，入力したコマンドに従って変数 **a** の値が更新されたことを確認します。b is you という値が表示されます。

7 ［**実行**］ボタンをクリックして，コンポーネントの実行を続行します。メッセージ・ボックスが開いて b is you と表示されます（変更された **a** の値）。これは，［デバッグ ビューア］表示枠を使用して，**a** と **b** の両方の値が正しく変更されたことを示しています。

8 ［**OK**］をクリックし，メッセージ・ボックスを閉じます。

手順 10：コマンド履歴からコマンドを繰り返す

1 最初のブレークポイントを削除して，コンポーネントを再び実行します。コンポーネントがブレークポイントで停止する場合（メッセージ・ボックスが表示される前），［変数］タブで変数 **b** の値を not me に変更します。

2 ［**コマンド**］タブを選択し，キーボードの上向き矢印キーを押します。QuickTest によって，前のコンポーネント実行で入力したコマンド（if b="me" then a="b is me" else a="b is you" end if）が，アクティブなコマンド・ラインにコピーされます。ENTER キーを押してコマンドを実行し，［**実行**］ボタンをクリックしてコンポーネント実行を完了させます。

第 30 章

コンポーネントの保守

QuickTest には，テスト対象のアプリケーションが変更されたときにコンポーネントを保守できるツールが用意されています。たとえば，アプリケーションのオブジェクトのプロパティや記述が変更されたり，オブジェクトが存在しなくなったりする場合があります。アプリケーションの変更に基づいて，コンポーネントのチェックポイントの期待値を更新する必要が生じる場合もあります。本章では，QuickTest のツールを使ってコンポーネントを更新および保守する方法を説明します。

本章の内容

- コンポーネントが失敗する理由（768 ページ）
- メンテナンス実行ウィザードを使ったコンポーネントの実行（770 ページ）
- 更新モード・オプションを使ったコンポーネントの更新（787 ページ）

コンポーネントが失敗する理由

コンポーネントは，QuickTest によって実行できないステップが発生した場合，またはステップの結果が失敗を示している場合に失敗します。多くの場合，これはテスト中のアプリケーションが正常に機能していないことが原因です。QuickTest は，アプリケーションの修正方法を理解するのに役立つテスト結果を提供します。

場合によってはコンポーネントの作成後にテスト中のアプリケーションが変更されたために，コンポーネントが失敗する場合もあります。その場合は，QuickTest のコンポーネントを更新して，それらの変更を反映する必要があります。オブジェクト・リポジトリに，テストを実行するのに必要なオブジェクトがない場合があります。QuickTest には，これらの問題のいくつかを特定して解決するのに役立つツールが用意されています。

オブジェクトの変更

QuickTest は，コンポーネントのステップを実行するときに，そのステップで参照されているオブジェクトをそのコンポーネントに関連付けられたオブジェクト・リポジトリ内で検索します。QuickTest は，リポジトリ内のオブジェクトの記述を使ってアプリケーション内のそのオブジェクトを認識しようとします。

QuickTest は，いくつかの理由でアプリケーション内のオブジェクトを認識できない場合があります。

オブジェクトがアプリケーション内に存在しない

オブジェクト・リポジトリ内のオブジェクトの記述と一致するオブジェクトがアプリケーション内に見つかりません。メンテナンス実行ウィザードを使用すると，コンポーネントで使用するオブジェクトを認識できます。

親オブジェクトが変更されている

オブジェクト・リポジトリ内のオブジェクトと一致し，同じ階層を持つオブジェクトがアプリケーション内に見つかりません。メンテナンス実行ウィザードを使用すると，コンポーネントで使用するオブジェクトを認識できます。

オブジェクト記述プロパティの値が変更されている

オブジェクト・リポジトリ内のオブジェクトとほぼ同じであり，同じ記述プロパティ値を持つオブジェクトがアプリケーション内に見つかりません。メンテ

ナンス実行ウィザードを使用すると，コンポーネントで使用するオブジェクトを認識できます。

オブジェクトがオブジェクト・リポジトリ内に存在しない

QuickTest は，アプリケーション内のオブジェクトを認識しようとする前に，関連付けられたオブジェクト・リポジトリ内でコンポーネントが参照するオブジェクトを検索します。関連付けられているオブジェクト・リポジトリでコンポーネントのオブジェクトが見つからない場合，メンテナンス実行ウィザードによってリポジトリに追加するアプリケーション内のオブジェクトを識別し，コンポーネントで使用できます。

オブジェクトの記述セットを変更する必要がある

QuickTest は，一連のプロパティを使ってアプリケーション内のオブジェクトを認識します。オブジェクト・リポジトリ内のオブジェクトの一連の認識プロパティを使ってアプリケーション内のオブジェクトと一致する一意の記述が得られない場合，QuickTest はオブジェクトを見つけることができません。更新モードを使ってコンポーネント内のオブジェクトの一連の認識プロパティを更新することにより，［オブジェクト リポジトリ］ダイアログ・ボックスで定義された一連のプロパティと照合できます。

チェックポイントの変更

テスト中のアプリケーションに予期しない状況が発生すると，チェックポイントは失敗します。多くの場合，これはアプリケーションが正常に機能していないことが原因です。QuickTest は，アプリケーションの修正方法の理解に役立つテスト結果を提供します。

コンポーネントの作成後にアプリケーションが変更されたために，チェックポイントが失敗する場合もあります。その場合は，QuickTest のチェックポイントを更新して，それらの変更を反映する必要があります。アプリケーションの変更を反映するには，更新モードを使ってコンポーネント内のチェックポイントを更新します。

たとえば，標準設定値が＜ Enter value ＞である編集ボックスがアプリケーションに存在し，新しい値が編集ボックスに入力される前にこの値を確認するチェックポイントがあると仮定します。アプリケーションの標準設定値が＜ Enter name ＞に変更されると，チェックポイントは失敗します。アプリケーションの変更を反映するには，更新モードを使ってチェックポイントの期待値を更新します。

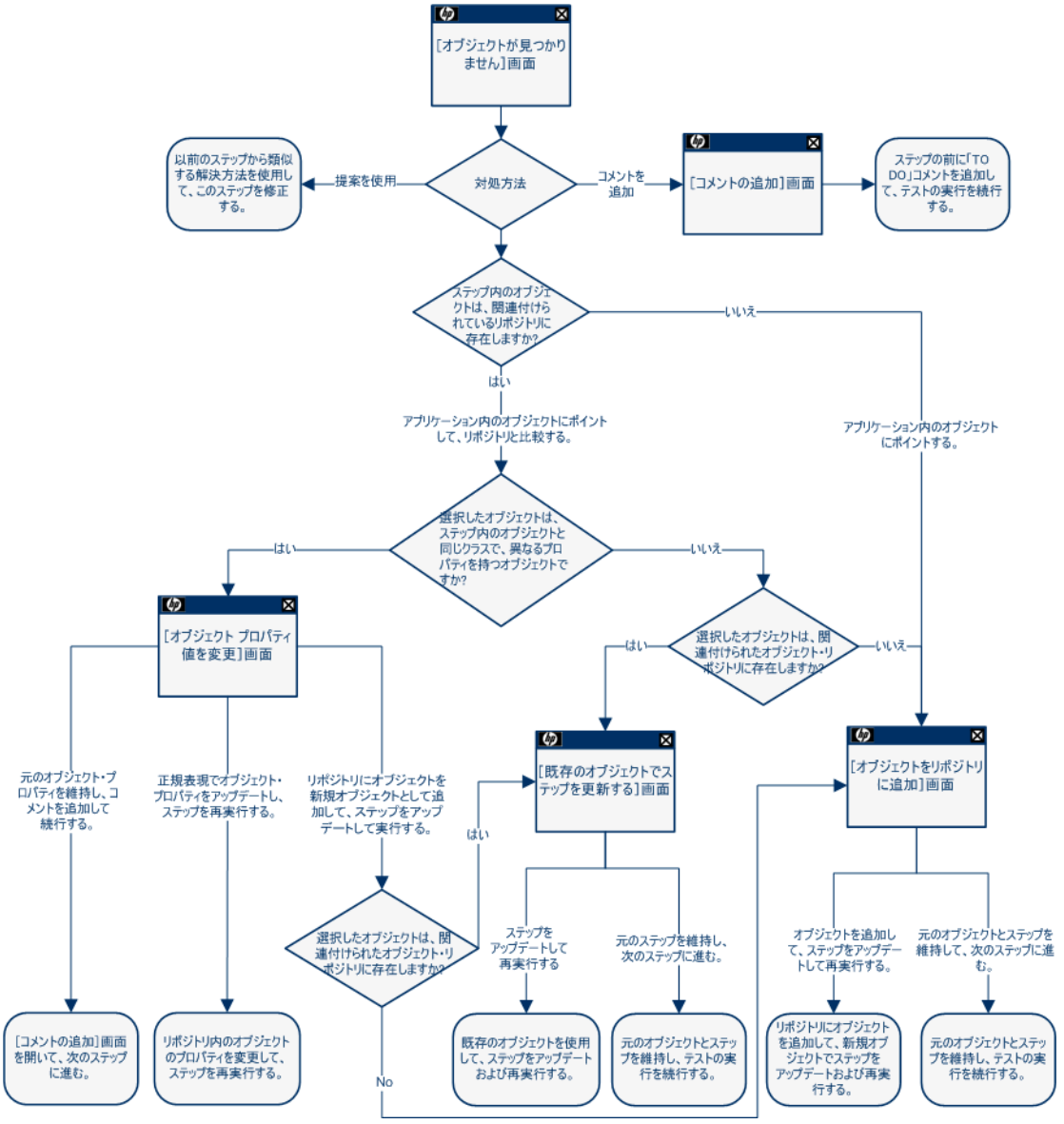
メンテナンス実行ウィザードを使ったコンポーネントの実行

メンテナンス実行ウィザードを使用すると，次の問題が発生して解決策を提供するときに，コンポーネントを保守できます。

問題	解決策
コンポーネントのオブジェクトをアプリケーション内で認識できないためにステップが失敗した。	メンテナンス実行ウィザードを使用すると，コンポーネントで使用するアプリケーション内のオブジェクトを認識できます。 テスト対象のアプリケーション内のオブジェクトを指定すると，そのオブジェクトと，関連付けられたオブジェクト・リポジトリ内のオブジェクトが比較されます。 メンテナンス実行ウィザードでは，指定したオブジェクトのプロパティ値と関連付けられたリポジトリ内のオブジェクトのプロパティ値との比較結果に応じて，コンポーネントを更新してアプリケーションの変更を反映するための方法が 1 つ示されます。 また，失敗したステップの前のコンポーネントにコメントを追加することもできます。
コンポーネントのオブジェクトが関連付けられているオブジェクト・リポジトリに存在しないために，ステップが失敗した。	メンテナンス実行ウィザードを使用すると，見つからなかったオブジェクトをリポジトリに追加できます。 また，失敗したステップの前のコンポーネントにコメントを追加することもできます。
ステップのオブジェクトはアプリケーションに存在するが，スマート認識を使わなければ認識できない。	スマート認識を使用してオブジェクトを認識すると，コンポーネントの実行速度が遅くなる場合があります (詳細については，209 ページ「スマート認識の設定」を参照してください)。メンテナンス実行ウィザードを使用すると，オブジェクトの説明を変更できます。このため，スマート認識は必要ありません。

メンテナンス実行モードでコンポーネントを実行すると，コンポーネントが実行された後，ステップとオブジェクト・リポジトリを更新する手順が順を追って示されます。メンテナンス実行ウィザードは，上記の状況に応じて起動されます。問題およびユーザの選択に基づいて，メンテナンス実行ウィザードには，複数の画面が表示されます。

次のフロー・チャートでは，状況ごとに表示する画面をウィザードおよびユーザが決定する際のロジックについて説明します。



注 : ［オブジェクトが見つかりません］画面は，テストのオブジェクトを認識するためにスマート認識を使用する際には開きません。この場合には，メンテナンス実行ウィザードによって，［オブジェクトの認識］ダイアログ・ボックスで現在定義されているプロパティに従って，オブジェクト・プロパティを更新することが提案されます。

メンテナンス実行ウィザードが終了すると，コンポーネントに対して行われた変更のサマリがメンテナンス実行ウィザードに表示されます。メインの［テスト結果］ウィンドウにも，コンポーネントに対して行われた変更の詳細を示すメンテナンスの要約が表示されます。これには，更新されたオブジェクト，追加されたオブジェクト，更新されたステップ，コメントが付けられたステップ，およびオブジェクト・リポジトリに対する変更の要約が含まれます。

メンテナンス実行ウィザードを使って作業する際の考慮事項

- メンテナンス実行モードでコンポーネントを実行するには，Microsoft Script Debugger がインストールされている必要があります。インストールされていない場合は，QuickTest Additional Installation Requirements ユーティリティを使ってインストールできます （［**スタート**］＞［**プログラム**］＞［**QuickTest Professional**］＞［**Tools**］＞［**Additional Installation Requirements**］を選択します)。
- メンテナンス実行モードで実行できるのは，QuickTest が **標準**（実行矢印を表示）実行モードを使用するように設定されている場合だけです。**高速**モードで実行することはできません。詳細については，638 ページ「テストの実行オプションの設定」を参照してください。
- ユーザ・インタフェースのないアプリケーション（Web サービスなど）は，メンテナンス実行モードではコンポーネントを実行できません。
- メンテナンス実行ウィザードを使用する代わりに，［オブジェクト リポジトリ］ウィンドウまたはオブジェクト・リポジトリ・マネージャの［**アプリケーションから更新**］オプションを使用して，アプリケーション内のオブジェクトから個々のテスト・オブジェクトの説明を更新できます。詳細については，178 ページ「アプリケーション内のオブジェクトから認識プロパティを更新」を参照してください。
- メンテナンス実行ウィザードを使ってコンポーネントを更新した後は，オブジェクト・リポジトリ・マネージャの［**ローカル リポジトリから更新**］オプ

ションを使ってローカル・オブジェクト・リポジトリのオブジェクトを共有オブジェクト・リポジトリに結合できます。詳細については，第 7 章「オブジェクト・リポジトリの管理」を参照してください。

コンポーネントをメンテナンス実行モードで実行するには，次の手順を実行します。

1 コンポーネントを開き，［**オートメーション**］＞［**メンテナンス実行モード**］を選択するか，ツールバーの［**実行**］ボタンの横にある下向き矢印をクリックして，［**メンテナンス実行モード**］を選択します。［実行］ダイアログ・ボックスが開きます。

2 メンテナンス実行モードのセッションに，結果の場所と（必要な場合は）入力パラメータの値を指定します。詳細については，674 ページ「［実行］ダイアログ・ボックス：［結果の場所］タブ」および 676 ページ「［実行］ダイアログ・ボックス：［入力パラメータ］タブ」を参照してください。

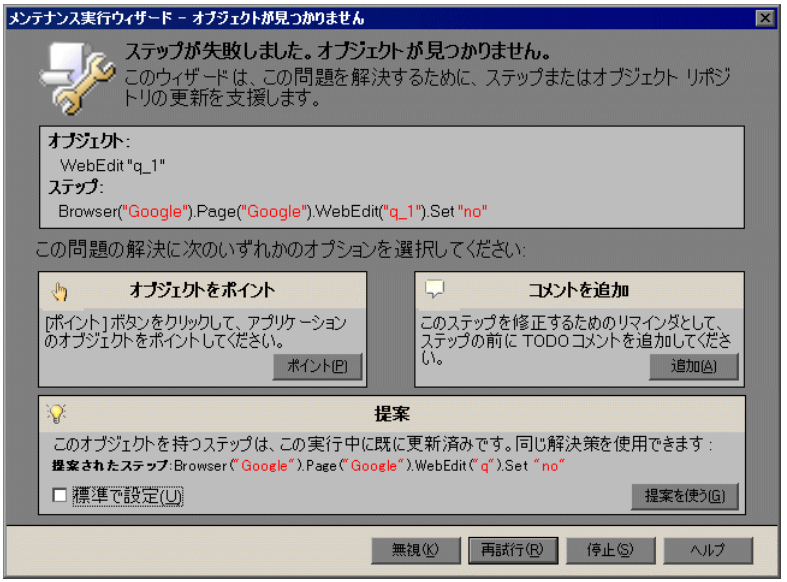
3 ［**OK**］をクリックします。［実行］ダイアログ・ボックスが閉じ，メンテナンス実行モードのセッションが開始されます。

標準設定では，実行セッションが終了すると，［テスト結果］ウィンドウが開きます。実行セッション結果の表示の詳細については，第 27 章「実行セッション結果の表示」を参照してください。［オプション］ダイアログ・ボックスの［実行］表示枠で［**実行セッション終了時に結果を表示する**］チェック・ボックスをクリアすると，実行セッションの最後で［テスト結果］ウィンドウは開きません。［オプション］ダイアログ・ボックスの詳細については，第 24 章「グローバル・テスト・オプションの設定」を参照してください。

メンテナンス実行ウィザード - ［オブジェクトが見つかりません］画面

コンポーネント内のオブジェクトがテスト中のアプリケーション内，または関連付けられているオブジェクト・リポジトリで見つからない場合は，［オブジェクトが見つかりません］画面が開きます。［オブジェクトが見つかりませ

ん］画面では，見つからなかった**オブジェクト**と QuickTest が実行しようとした**ステップ**が特定されます。



注：

［**提案**］表示枠は、1 つの実行セッションでアプリケーション内にオブジェクトを前と同じように見つけられなかった場合にのみ表示されます。

次の場合には，見つからなかったオブジェクトに対するメンテナンス実行ウィザードの［**オブジェクトをポイント**］オプションおよび［**コメントを追加**］オプションが無効になります。

- テストが読み取り専用モードで開かれている。
- オブジェクトが関数ライブラリの関数内で使用されている。
- オブジェクトのメソッドが登録されたユーザ関数として定義されている。

［オブジェクトが見つかりません］画面では，次のオプションを使って問題を解決できます。

➤ [**オブジェクトをポイント**]：[**ポイント**] ボタンをクリックして，ステップで使用する必要があるアプリケーション内のオブジェクトを指定します。アプリケーションが変更され，ステップで使用する新しいオブジェクトを認識すると問題が解決することがわかっている場合，または関連付けられているオブジェクト・リポジトリにオブジェクトが存在しない場合は，このオプションを使用します。

指定した場所が複数のオブジェクトに関連付けられている場合は，[オブジェクトの選択] ダイアログ・ボックスが開きます。ツリーから正しいオブジェクトを選択し，[**OK**] をクリックします。

指定したオブジェクトに応じて，次のいずれかの画面が開きます。

- ➤ 780 ページ「メンテナンス実行ウィザード - [既存のオブジェクトでステップを更新する] 画面」
- ➤ 782 ページ「メンテナンス実行ウィザード - [オブジェクトをリポジトリに追加] 画面」
- ➤ 777 ページ「メンテナンス実行ウィザード - [オブジェクト プロパティ値を変更] 画面」

➤ [**コメントを追加**]：失敗したステップを修正するためのメモとしてテストにコメントを追加する場合に，このオプションを使用します。[コメントの追加] 画面が開きます。

➤ [**提案**]：表示枠は、1 つの実行セッションでアプリケーション内にオブジェクトを前と同じように見つけられなかった場合にのみ表示されます。最初に見つからなかったオブジェクトを別のオブジェクトに置き換えた場合は，同じオブジェクトに置き換えることを提案するメッセージが表示されます。

- ➤ [**標準で設定**]：後続のステップで同じオブジェクトが見つからなかった場合に，見つからなかったオブジェクトを，オブジェクト・リポジトリに追加したオブジェクトで自動的に置換します。これらの後続のステップは開かれません。

これらのオプションを使用しない場合は，次のボタンを使用して継続できます。

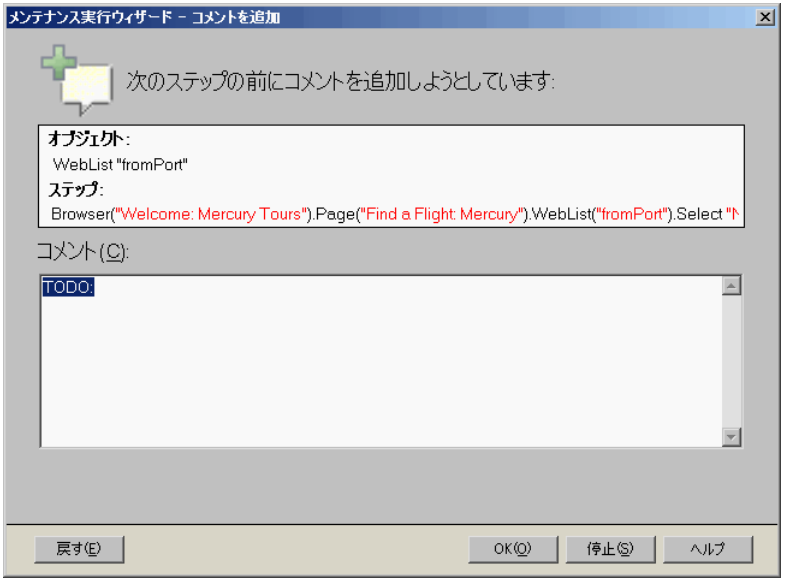
➤ [**無視**]：コンポーネントの現在のステップを無視し，コンポーネントの残りに対してメンテナンス実行ウィザードの実行を継続します。これは，問題が QuickTest のコンポーネントではなく，テスト中のアプリケーションにある場合に使用できます。

注：［**無視**］をクリックする前に，アプリケーションがコンポーネントの次のステップを実行できる状態であることを確認してください。

- ［**再試行**］：現在のステップを再試行します。
- ［**停止**］：メンテナンス実行を停止し，［メンテナンス モードのサマリ］画面を開きます。
- ［**ヘルプ**］：このヘルプ・トピックを開きます。

メンテナンス実行ウィザード -［コメントの追加］画面

［コメントの追加］画面では，コンポーネントの現在のステップの前にコメントを追加できます。これは，コンポーネントに問題があることは明確だが，アプリケーション内のオブジェクトを認識しても問題が解決しない，またはコンポーネントを手作業で修正したい場合に使用できます。

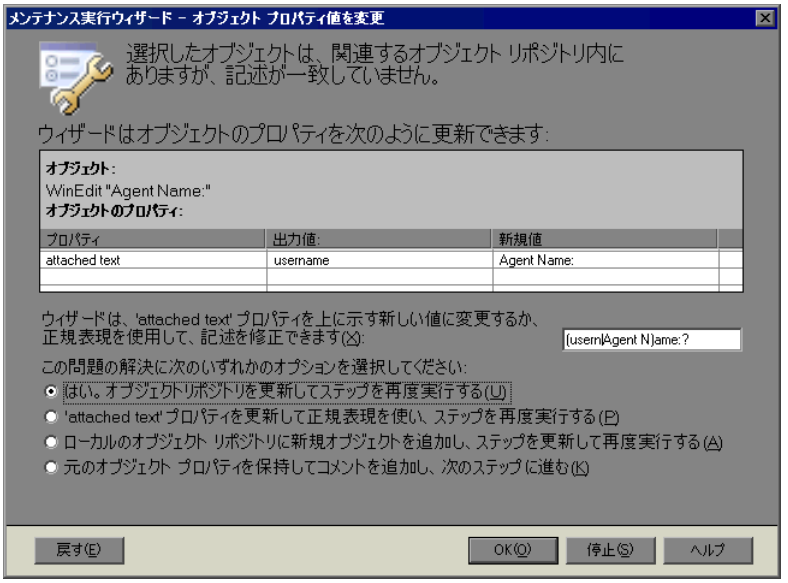


［コメントの追加］画面では，コンポーネント内に，後でステップを修正するためのメモとして，「TODO」と追加したテキストから成るコメントを作成します。

メンテナンス実行ウィザード - ［オブジェクト プロパティ値を変更］画面

［オブジェクト プロパティ値を変更］画面は，指定したオブジェクトがステップのオブジェクトと同じクラスで，記述プロパティ値が異なる場合に開きます。

［オブジェクト プロパティ値を変更］画面では，関連付けられているオブジェクト・リポジトリ内のオブジェクトのプロパティ値を更新して，アプリケーション内で指定したオブジェクトのプロパティ値と一致させることが提案されます。



注：正規表現が新しいプロパティ値に対して適切であると判断されない場合は，［オブジェクト プロパティ値を変更］画面のプロパティの表の下に，提案される正規表現は表示されません。［**＜プロパティ名＞プロパティを更新して正規表現を使い，ステップを再度実行する**］ラジオ・ボタンも表示されません。

［オブジェクト プロパティ値を変更］画面の中央には，次のセクションがあります。

セクション	**説明**
［オブジェクト］	関連付けられたオブジェクト・リポジトリ内にある，アプリケーション内で指定したオブジェクトと同じクラスのオブジェクト。
［オブジェクトのプロパティ］	オブジェクト・リポジトリ内のオブジェクトのプロパティ値に対して行われる変更を表示する表。
［プロパティ］	値が変更されるプロパティの名前。
［元の値］	オブジェクト・リポジトリ内のオブジェクトの元のプロパティ値。
［新規値］	アプリケーション内で指定したオブジェクトに基づく，オブジェクト・リポジトリ内のオブジェクトの新しいプロパティ値。
推奨される正規表現	指定したオブジェクトによっては，［オブジェクト プロパティ値を変更］画面に，関連付けられたオブジェクト・リポジトリ内のオブジェクトのプロパティ値を更新するために正規表現を使用できることを示すメッセージが表示される場合があります。エディット・ボックスで推奨される正規表現を変更することもできます。正規表現の詳細については，617 ページ「正規表現の概要と使い方」を参照してください。 **注**：複数のプロパティで正規表現を使用できる場合は，最初のプロパティ値の正規表現だけが示されます。

［オブジェクト プロパティ値を変更］画面には，次のオプションが表示されます。

➤ ［**はい。オブジェクト リポジトリを更新してステップを再度実行する**］：アプリケーション内で指定したオブジェクトのプロパティ値と一致するようにオブジェクト・リポジトリ内のオブジェクトのプロパティ値を更新して，ステップを再実行します。新しいプロパティ値は，［**新規値**］の下に表示されます。

- ［**＜プロパティ名＞プロパティを更新して正規表現を使い，ステップを再度実行する**］：正規表現を使うようにプロパティ値を更新できる場合にのみ表示されます。オブジェクト・リポジトリ内のオブジェクトのプロパティ値をエディット・ボックスに示された正規表現で更新して，ステップを再実行します。
- ［**ローカルのオブジェクト リポジトリに新規オブジェクトを追加し，ステップを更新して再度実行する**］：このオプションを指定すると，指定したオブジェクトが現在のプロパティとともに新しいオブジェクトとしてローカル・オブジェクト・リポジトリに追加されます。この新しいオブジェクトは，関連付けられているオブジェクト・リポジトリにすでに存在する場合があります。次の画面のいずれかが開きます。
  - ［既存のオブジェクトでステップを更新する］画面。この画面は，追加するオブジェクトが関連付けられたオブジェクト・リポジトリ内にすでに存在する場合に開きます。
  - ［オブジェクトをリポジトリに追加］画面。この画面は，追加するオブジェクトが関連付けられたオブジェクト・リポジトリ内に存在しない場合に開きます。
- ［**元のオブジェクト プロパティを保持してコメントを追加し，次のステップに進む**］。オブジェクト・リポジトリ内のオブジェクトの元のオブジェクト・プロパティを保持します。［コメントの追加］画面が開き，ステップの前にコメントを追加できます。その後，次のステップに進みます。

  画面の下部には，［**リセット**］ボタンが表示されます。このボタンをクリックすると，［オブジェクトが見つかりません］画面に戻り，アプリケーション内の別のオブジェクトを指定したり，このステップに対する別の措置を選択したりできます。

注：

- 指定したオブジェクトがオブジェクト・リポジトリ内のオブジェクトとは異なる親オブジェクトとプロパティ値を持っている場合は，［オブジェクト プロパティ値を変更］画面が 2 回開きます。最初の画面では，オブジェクト・リポジトリ内のオブジェクトの親オブジェクトを更新して，指定したオブジェクトの親オブジェクトと一致させることができます。2 つ目の画面では，オブジェクト・リポジトリ内のオブジェクトを更新して，指定したオブジェクトと一致させることができます。
- メンテナンス実行ウィザードでは，ローカル・オブジェクト・リポジトリのみが変更されます。新しいオブジェクトを共有オブジェクト・リポジトリに含めるには，オブジェクト・リポジトリ・マネージャを使用します。詳細については，255 ページ「結合操作の実行」を参照してください。

メンテナンス実行ウィザード - ［既存のオブジェクトでステップを更新する］画面

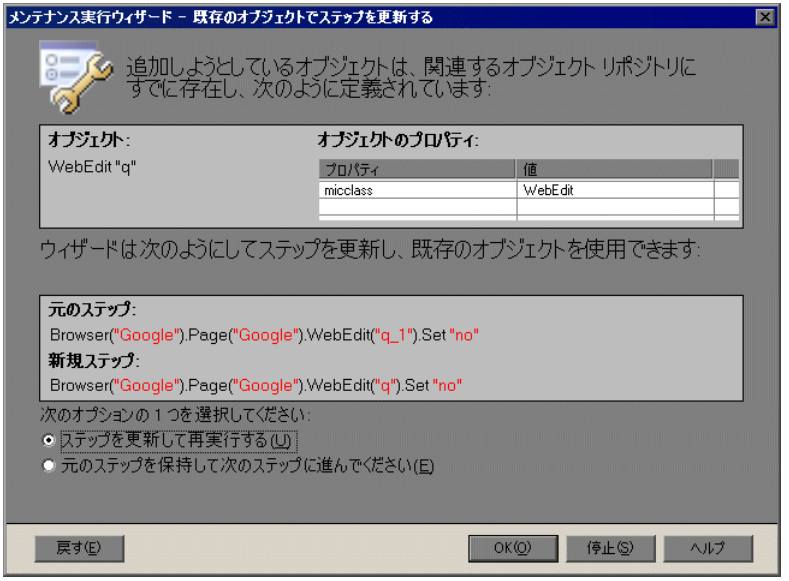
［既存のオブジェクトでステップを更新する］画面は，［オブジェクトが見つかりません］画面で指定したオブジェクトが関連付けられたオブジェクト・リポジトリ内に存在し，次の条件を満たす場合に開きます。

- 指定したオブジェクトはステップ内のオブジェクトと同じクラスではなく，異なる記述プロパティ値を保持している。

または

- ［オブジェクト プロパティ値を変更］画面で［**ローカルのオブジェクト リポジトリに新規オブジェクトを追加し，ステップを更新して再度実行する**］を選択した。

[既存のオブジェクトでステップを更新する] 画面では，テスト内のステップを更新して，関連付けられたオブジェクト・リポジトリ内にすでに存在するオブジェクトを使用することが提案されます。



[既存のオブジェクトでステップを更新する] 画面の中央には，次のセクションがあります。

セクション	説明
[オブジェクト]	関連付けられたオブジェクト・リポジトリ内にある，アプリケーション内で指定したオブジェクトと同じオブジェクト。
[オブジェクトのプロパティ]	テスト・アプリケーション内で指定したオブジェクトのプロパティおよびプロパティ値。
[元のステップ]	見つからなかったオブジェクトを含む，失敗した元のステップ。
[新規ステップ]	関連付けられたオブジェクト・リポジトリ内にすでに存在するオブジェクトを参照するようにステップを更新した場合の，新しいステップ。

［既存のオブジェクトでステップを更新する］画面には，次のオプションが表示されます。

➤［**ステップを更新して再実行する**］: 失敗したステップを［**新規ステップ**］に示された内容で更新し，ステップを再実行します。

注： メンテナンス実行ウィザードでは，コンポーネントから元のステップは削除されません。元のステップはコメントに変換され，更新されたステップがその下に追加されます。

➤［**元のステップを保持して次のステップに進んでください。**］: 元のステップを保持し，コンポーネントの残りの部分に対してメンテナンス実行ウィザードの実行を継続します。

画面の下部には，［**戻す**］ボタンが表示されます。このボタンをクリックすると，［オブジェクトが見つかりません］画面に戻り，アプリケーション内の別のオブジェクトを指定したり，このステップに対する別の措置を選択したりできます。

メンテナンス実行ウィザード - ［オブジェクトをリポジトリに追加］画面

［オブジェクトをリポジトリに追加］画面は，次の場合に開きます。

➤ **ステップ**内のオブジェクトが関連付けられているリポジトリに存在しない。

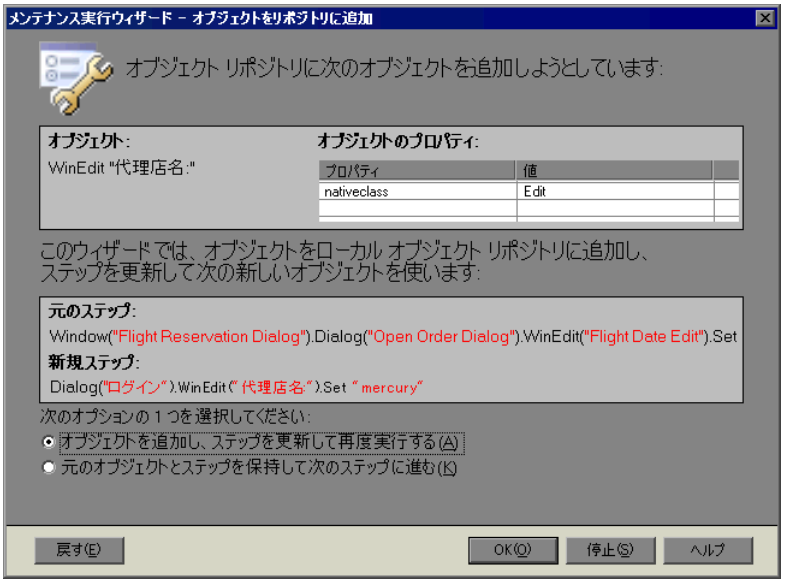
➤ **指定した**オブジェクトが関連付けられているオブジェクト・リポジトリに存在せず，次の状況に当てはまる。

  ➤ 指定したオブジェクトはステップ内のオブジェクトと同じクラスではなく，異なる記述プロパティ値を保持している。

  または

  ➤［オブジェクト プロパティ値を変更］画面で［**ローカルのオブジェクト リポジトリに新規オブジェクトを追加し，ステップを更新して再度実行する**］を選択した。

[オブジェクトをリポジトリに追加] 画面では，指定したオブジェクトをオブジェクト・リポジトリに追加することが提案されます。



[オブジェクトをリポジトリに追加] 画面の中央には，次のセクションがあります。

セクション	説明
[オブジェクト]	テスト・アプリケーション内で指定したオブジェクト。
[オブジェクトのプロパティ]	テスト・アプリケーション内で指定したオブジェクトのプロパティおよびプロパティ値。
[元のステップ]	見つからなかったオブジェクトを含む，失敗した元のステップ。
[新規ステップ]	オブジェクト・リポジトリに追加されるオブジェクトを参照するようにステップを更新した場合の，新しいステップ。

[オブジェクトをリポジトリに追加] 画面には，次のオプションが表示されます。

➤ [**オブジェクトを追加し、ステップを更新して再度実行する**]：オブジェクト・リポジトリに新しいオブジェクトを追加し，失敗したステップを［**新規ステップ**］に示された内容で更新して，ステップを再実行します。

➤ [**元のオブジェクトとステップを保持して次のステップに進む**]：元のオブジェクトを含む元のステップを保持し，コンポーネントの残りの部分に対してメンテナンス実行ウィザードの実行を継続します。

画面の下部には，［**戻す**］ボタンが表示されます。このボタンをクリックすると，［オブジェクトが見つかりません］画面に戻り，アプリケーション内の別のオブジェクトを指定したり，このステップに対する別の措置を選択したりできます。

注：

➤ メンテナンス実行ウィザードでは，ローカル・オブジェクト・リポジトリのみが変更されます。新しいオブジェクトを共有オブジェクト・リポジトリに含めるには，オブジェクト・リポジトリ・マネージャを使用します。詳細については，255 ページ「結合操作の実行」を参照してください。

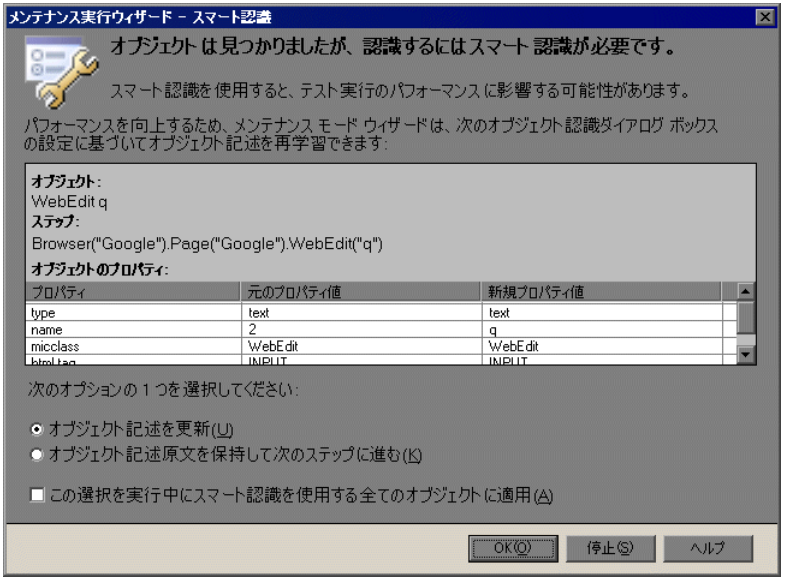
➤ メンテナンス実行ウィザードでは，コンポーネントから元のステップは削除されません。元のステップはコメントに変換され，更新されたステップがその下に追加されます。

メンテナンス実行ウィザード - ［スマート認識］画面

スマート認識画面は，コンポーネント内のオブジェクトを認識するために，QuickTest によってスマート認識メカニズムが使用されると開きます。スマート認識メカニズムの詳細については，209 ページ「スマート認識の設定」を参照してください。

スマート認識は，オブジェクト同期タイムアウトに達した後にアクティブになるため，コンポーネントの実行速度が遅くなる場合があります。

スマート認識画面に，［オブジェクトの認識］ダイアログ・ボックスで現在定義されているプロパティに従って，オブジェクト記述を更新することが提案されます。画像：確認してください。



［スマート認識］画面の中央には，次のセクションがあります。

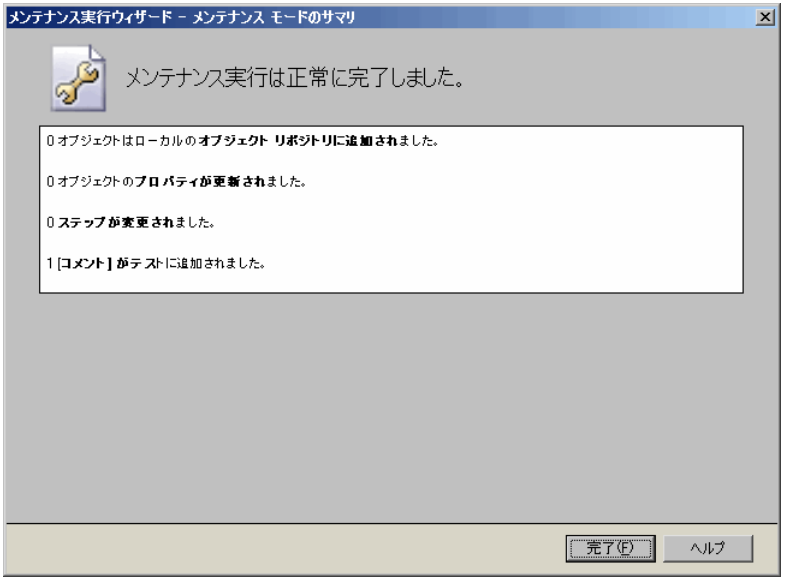
セクション	説明
［オブジェクト］	認識するのにスマート認識メカニズムを必要としたアプリケーション内のオブジェクト。
ステップ	オブジェクトが参照されるテストのステップ。
［オブジェクトのプロパティ］	［**プロパティ**］：古いオブジェクト記述および新しいオブジェクト記述のプロパティのリスト。 ［**元のプロパティ値**］：［**プロパティ**］カラムのプロパティの元の値。値を持たないプロパティは，元のオブジェクト記述の一部ではありません。 ［**新規プロパティ値**］：［**プロパティ**］カラムのプロパティの新しい値。

［スマート認識］画面では，次のオプションが提供されます。

- ［**オブジェクト記述を更新**］：テスト内のオブジェクトの［オブジェクトの認識］ダイアログ・ボックスで現在定義されている一連のプロパティを使用するために，オブジェクト記述を更新します。オブジェクトの［オブジェクトの認識］ダイアログ・ボックスに定義されている一連のプロパティで，オブジェクトを一意に認識できることを確認してください。
- ［**オブジェクト記述原文を保持して次のステップに進む**］：元のオブジェクトを含む元のステップを維持し，コンポーネントの残りに対してメンテナンス実行ウィザードの実行を継続します。実行中は，このオブジェクトに対して［スマート認識］画面が再び開かれることはありません。
- ［**この選択を実行中にスマート認識を使用する全てのオブジェクトに適用**］：認識するのにスマート認識メカニズムを必要とするコンポーネントのすべてのオブジェクトに対して，上記で選択したラジオ・ボタンの項目を使用します。

メンテナンス実行ウィザード - ［メンテナンス モードのサマリ］画面

メンテナンス実行ウィザードを終了すると，［メンテナンス モードのサマリ］画面が開きます。



［メンテナンス モードのサマリ］画面には，ローカル・**オブジェクト・リポジトリ**に追加されたオブジェクトの数，更新されたオブジェクト・**プロパティ**の数，変更された**ステップ**の数，およびテストに追加された**コメント**の数が表示されます。

［**完了**］をクリックすると，メンテナンス実行ウィザードが終了します。標準設定では，実行セッションが終了すると，［テスト結果］ウィンドウが開き，実行中に更新されたステップとオブジェクトに関する詳細が表示されます。実行セッション結果の表示の詳細については，681 ページ「［テスト結果］ウィンドウ」を参照してください。

注 : ［オプション］ダイアログ・ボックスの［実行］表示枠で［**実行セッション終了時に結果を表示する**］チェック・ボックスをクリアすると，実行セッションの最後で［テスト結果］ウィンドウは開きません。［オプション］ダイアログ・ボックスの詳細については，第 24 章「グローバル・テスト・オプションの設定」を参照してください。

更新モード・オプションを使ったコンポーネントの更新

更新モードでコンポーネントを実行すると，QuickTest によってコンポーネントが実行され，テスト・オブジェクト記述，チェックポイントの期待値，あるいはその両方が更新されます。コンポーネントを保存すると，更新されたデータが以降の実行に使用されます。

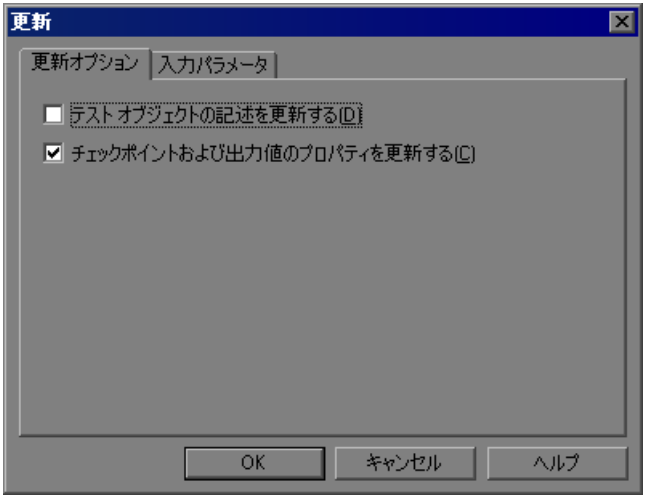
注 : QuickTest によるコンポーネントの更新時には，更新対象のオブジェクトが元々は共有オブジェクト・リポジトリに含まれていた場合でも，更新されたオブジェクトはローカル・オブジェクト・リポジトリに保存されます。ローカル・オブジェクト・リポジトリはどの共有オブジェクト・リポジトリよりも高い優先順位を持っているため，次回コンポーネントを実行したときは，ローカル・オブジェクト・リポジトリのオブジェクトが使用されます。

ヒント：更新モードを使ってコンポーネントを更新した後は，オブジェクト・リポジトリ・マネージャの［**ローカル リポジトリから更新**］オプションを使ってローカル・オブジェクト・リポジトリのオブジェクトを共有オブジェクト・リポジトリに結合できます。詳細については，第 7 章「オブジェクト・リポジトリの管理」を参照してください。

---

1 テストを開き，［**オートメーション**］＞［**更新モード**］を選択するか，ツールバー上の［**実行**］ボタンの横にある下向き矢印をクリックして［**更新モード**］を選択します。

［更新］ダイアログ・ボックスが表示されます。



2 更新プロセスの設定を行います。詳細については，790 ページ「［更新オプション］タブについて」および 676 ページ「［実行］ダイアログ・ボックス：［入力パラメータ］タブ」を参照してください。

注：更新セッションの実行結果は，常に一時保存場所に保存されます。

---

3 ［**OK**］をクリックします。［更新］ダイアログ・ボックスが閉じ，QuickTest によって更新モードでの実行が開始されます。コンポーネントの更新中は，ステータス・バーで「**更新**」というテキストが点滅します。

コンポーネントが実行され，テスト・オブジェクト記述が更新されます。実行セッションが終了すると，［テスト結果］ウィンドウが開きます。

結果を表示する方法の詳細については，第 27 章「実行セッション結果の表示」を参照してください。

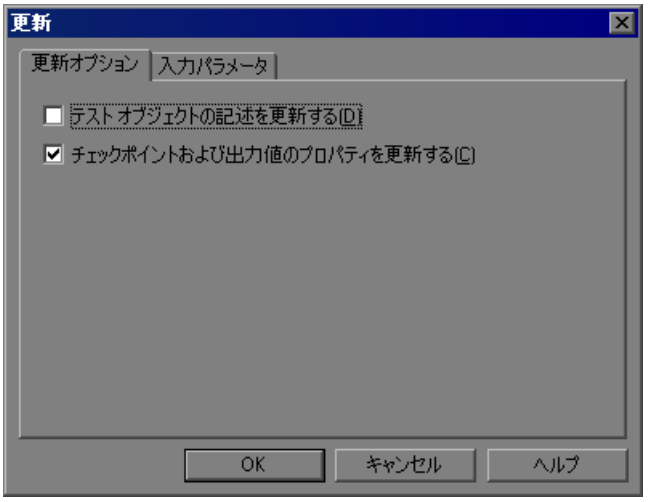
注 : ［オプション］ダイアログ・ボックスの［実行］表示枠で［**実行セッション終了時に結果を表示する**］チェック・ボックスをクリアすると，更新セッションの最後で［テスト結果］ウィンドウは開きません。［オプション］ダイアログ・ボックスの詳細については，第 24 章「グローバル・テスト・オプションの設定」を参照してください。

更新が終了すると，［テスト結果］ウィンドウには次の情報が表示されます。

- チェックポイントの更新された値。
- 更新されたテスト・オブジェクト記述。

[更新オプション] タブについて

[更新オプション] タブでは，コンポーネントのどの側面（テスト・オブジェクト記述，チェックポイントの期待値，あるいはその両方）を更新するかを指定できます。コンポーネントを保存すると，更新されたコンポーネントの結果が以降の実行に使用されます。



更新する情報のタイプを次から指定できます。

➤ [**テスト オブジェクト記述の更新**]。関連付けられたオブジェクト・リポジトリ内の各オブジェクト・クラスのプロパティ・セットが，[オブジェクトの認識] ダイアログ・ボックスで現在定義されているプロパティに従って更新されます。このオプションを使用して，特定のタイプのオブジェクトの認識に使用するプロパティのセットを変更できます。

注： あるオブジェクト・クラスに対して [オブジェクトの認識] ダイアログ・ボックスで選択したプロパティ・セットが特定のオブジェクトには適していない場合，新しいオブジェクト記述のために以降の実行が失敗する可能性があります。そのため，テストを更新する前にオブジェクト・リポジトリのコピーを保存し，必要に応じて以前のバージョンに戻せるようにしておくことをお勧めします。

このオプションは，アプリケーション内の認識しやすいオブジェクト・プロパティ値（オブジェクトのラベルなど）を使用してコンポーネントのステップの作成とデバッグを行う場合に特に便利ですが，言語やオペレーティング・システムに依存する場合があります。コンポーネントをデバッグした後で，［**更新モード**］オプションを使用し，オブジェクト記述を変更してより汎用的なプロパティ値を使用できます。

たとえば，アプリケーションの特定部分の英語版を対象にコンポーネントを設計したとします。テスト・オブジェクトは英語版の認識プロパティの値に従って認識されますが，その一部は言語に依存する場合があります。次に，同じアプリケーションの対象部分について，フランス語版でも同じコンポーネントを使用するとします。

このためには，言語に依存しないプロパティを定義して，QuickTest が，これらのプロパティをオブジェクトの認識に使用できるようにします。たとえば，リンク・オブジェクトの認識に **text** プロパティ値ではなく **target** プロパティ値を使用することもできます。プロパティを定義したら，アプリケーションの対象部分の英語版に対して，定義した新しいプロパティを使用して更新を行います。テスト・オブジェクト記述が変更され，後でフランス語版のアプリケーションに対してもコンポーネントを正しく実行できるようになります。

ヒント：正常に実行されるコンポーネントがあるが，特定のオブジェクトがスマート認識を使って認識される場合は，オブジェクトの認識に使用されるプロパティ・セットを変更した後で，［**テスト オブジェクト記述の更新**］オプションを使って，スマート認識がオブジェクトの認識に使用しているプロパティ・セットを使用するようにテスト・オブジェクト記述を更新できます。

［**テスト オブジェクト記述の更新**］を選択してコンポーネントを実行すると，各ステップに指定されているテスト・オブジェクトが，現在のテスト・オブジェクト記述に基づいて検出されます。QuickTest で記述に基づいてテスト・オブジェクトを検出できない場合は，（スマート認識が有効な場合）テスト・オブジェクトの認識にはスマート認識プロパティが使用されます。QuickTest によってテスト・オブジェクトが検出された後，［オブジェクトの認識］ダイアログ・ボックスで定義した必須プロパティと補足プロパティに基づいて，記述が更新されます。

注：更新プロセス中に認識されないテスト・オブジェクトは更新されません。その他の実行セッションの場合と同じように，更新モードでも，実行中にオブジェクトが見つからない場合は実行セッションが失敗し，失敗に関する情報がテスト結果に追加されます。このような状況では，メンテナンス実行モードを使ってこれらの問題を解決することをお勧めします。

以前のテスト・オブジェクト記述に使用され，［オブジェクトの認識］ダイアログ・ボックスで定義されるテスト・オブジェクト・クラスの記述の一部ではなくなっているすべてのプロパティは，値が正規表現としてパラメータ化または定義されている場合であっても，新しい記述から削除されます。

同じプロパティがテスト・オブジェクトの新しい記述と以前の記述の両方に出現する場合は，次のいずれかの処理が行われます。

- 以前の記述のプロパティ値が正規表現としてパラメータ化または指定され，新しいプロパティ値と一致する場合，そのプロパティの以前のパラメータ化された値または正規表現の値は保持されます。たとえば，以前のプロパティ値が正規表現 button.* として定義され，新しい値が button1 である場合，プロパティ値は button.* のままになります。
- 以前の記述のプロパティ値が新しいプロパティ値と一致しないものの，オブジェクトがスマート認識を使用して検出された場合，そのプロパティ値は新しい定数のプロパティ値に更新されます。たとえば，以前のプロパティ値が button.* であり，新しい値が My button である場合，QuickTest でオブジェクトを検出するスマート認識定義が有効になっていれば，My button が新しいプロパティ値になります。この場合，パラメータ化されているものや正規表現が使用されているものは，すべてテスト・オブジェクト記述から削除されます。

- ［**チェックポイントおよび出力値のプロパティを更新する**］：テストの作成後にアプリケーションで発生した変更を反映するためにチェックポイントの期待値が更新され，出力値ステップで取得できる項目のリストが更新されます。

  たとえば，テストの一部としてテキスト・チェックポイントを定義していて，テストの作成後にアプリケーション内のテキストが変更されたとします。テストを更新して，新しいテキストを反映するため，チェックポイントのプロパティを更新できます。

出力値オプションは，主に Web サービス・コンポーネントで使用される XML 出力値ステップに適用されます。詳細については，『**HP QuickTest Professional アドイン・ガイド**』の Web サービスについて説明している項を参照してください。

注：

- チェックポイントのプロパティの値がパラメータ化されているか正規表現を含んでいる場合は，このオプションを使用しても更新されません。
- ビットマップ・チェックポイントの作成時に［**選択領域のみを保存**］チェック・ボックスを選択した場合，［**更新モード**］オプションによって更新されるのは，ビットマップの保存された領域のみです。オリジナルのフル・サイズのオブジェクトは更新されません。チェックポイントのオブジェクトを増やすには，新しいチェックポイントを作成します。詳細については，579 ページ「ビットマップの検査」を参照してください。

第 IX 部

QuickTest IDE の使用

第 31 章

QuickTest ウィンドウのレイアウト

本章では，QuickTest ウィンドウをカスタマイズする方法，および QuickTest ドキュメントを使用する方法について説明します。

本章の内容

- QuickTest ウィンドウのレイアウト変更（797 ページ）
- ツールバーとメニューのカスタマイズ（811 ページ）
- 複数のドキュメントを使った作業（824 ページ）

QuickTest ウィンドウのレイアウト変更

QuickTest ウィンドウのレイアウトは変更できます。たとえば，表示枠の移動とサイズ変更，表示枠の表示と自動非表示の切り替え，タブ表示枠の作成，および表示するツールバーの選択が行えます。必要に応じて，標準のレイアウトを復元することもできます。

また，QuickTest の各種セッション（表示 / 編集，記録，および実行セッション）の必要に合わせて QuickTest ウィンドウのサイズを変更することも可能です。たとえば，コンポーネントまたはアプリケーション領域を作成または編集するときは，QuickTest を画面全体に表示して，実行セッション中には QuickTest ウィンドウを最小化できます。詳細については，810 ページ「QuickTest ウィンドウ・レイアウトのカスタマイズ」を参照してください。

QuickTest ウィンドウのレイアウトをカスタマイズまたは復元すると，QuickTest により，すべてのドキュメント・タイプおよびセッション・タイプに変更が適用されます。

詳細については，次を参照してください。

- 798 ページ「表示枠の移動」
- 807 ページ「表示枠の表示と非表示」
- 809 ページ「ツールバーのフローティングとドッキング」
- 809 ページ「QuickTest ウィンドウの標準のレイアウトの復元」
- 810 ページ「QuickTest ウィンドウ・レイアウトのカスタマイズ」

表示枠の移動

QuickTest ウィンドウの表示枠は，ユーザ自身の使いやすい位置に移動できます。また，表示枠を並べ替えたり，表示枠をタブ形式の表示枠に（またはその逆に）変更したりもできます。

表示枠をドラッグしている間は，QuickTest ウィンドウ上にマーカが表示されます。いずれかのマーカの上にカーソルを置くと，そのマーカによって示される領域がグレー表示され，選択した位置に表示枠を移動した場合のウィンドウのレイアウトを確認できます。

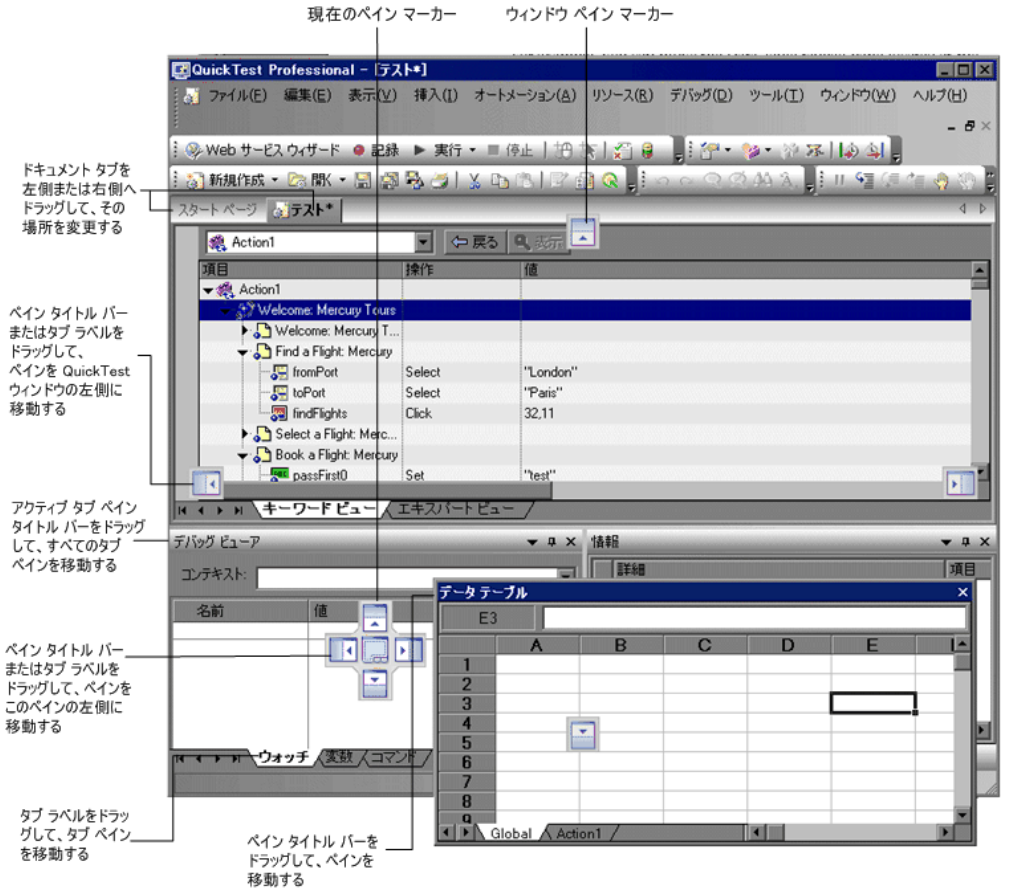
ヒント： ドッキング可能な表示枠がはめ込まれないようにして移動するには，CTRL キーを押したままで目的の場所へドラッグします。

表示枠を移動するには，次の手順を実行します。

1 QuickTest ウィンドウで，移動する表示枠のタイトル・バーまたはタブをドラッグします。目的の表示枠が QuickTest ウィンドウに表示されていない場合は，［**表示**］メニューから選択します。

たとえば，ウィンドウの中央にある［欠落リソース］表示枠を，新しいタブ形式の表示枠としてウィンドウの一番下に移動できます。表示枠をドラッグすると，アクティブな表示枠および QuickTest ウィンドウの各辺にマーカが表示さ

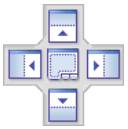
れます。


現在のペイン マーカー
ウィンドウ ペイン マーカー
ドキュメント タブを左側または右側へドラッグして、その場所を変更する
ペイン タイトル バーまたはタブ ラベルをドラッグして、ペインを QuickTest ウィンドウの左側に移動する
アクティブ タブ ペイン タイトル バーをドラッグして、すべてのタブ ペインを移動する
ペイン タイトル バーまたはタブ ラベルをドラッグして、ペインをこのペインの左側に移動する
タブ ラベルをドラッグして、タブ ペインを移動する
ペイン タイトル バーをドラッグして、ペインを移動する
QuickTest Professional - [テスト*]
ファイル(F) 編集(E) 表示(V) 挿入(I) オートメーション(A) リソース(R) デバッグ(D) ツール(T) ウィンドウ(W) ヘルプ(H)
Web サービス ウィザード
記録
実行
停止
新規作成
開く
スタート ページ
テスト*
Action1
戻る
表示
項目
操作
値
Action1
Welcome: Mercury Tours
Welcome: Mercury T...
Find a Flight: Mercury
fromPort Select "London"
toPort Select "Paris"
findFlights Click 32,11
Select a Flight: Merc...
Book a Flight: Mercury
passFirst0 Set "test"
キーワード ビュー
エキスパート ビュー
デバッグ ビュー
コンテキスト:
情報
詳細
項目
名前
値
ウォッチ
変数
コマンド
データ テーブル
E3
A B C D E
1 2 3 4 5 6 7 8 9
Global
Action1

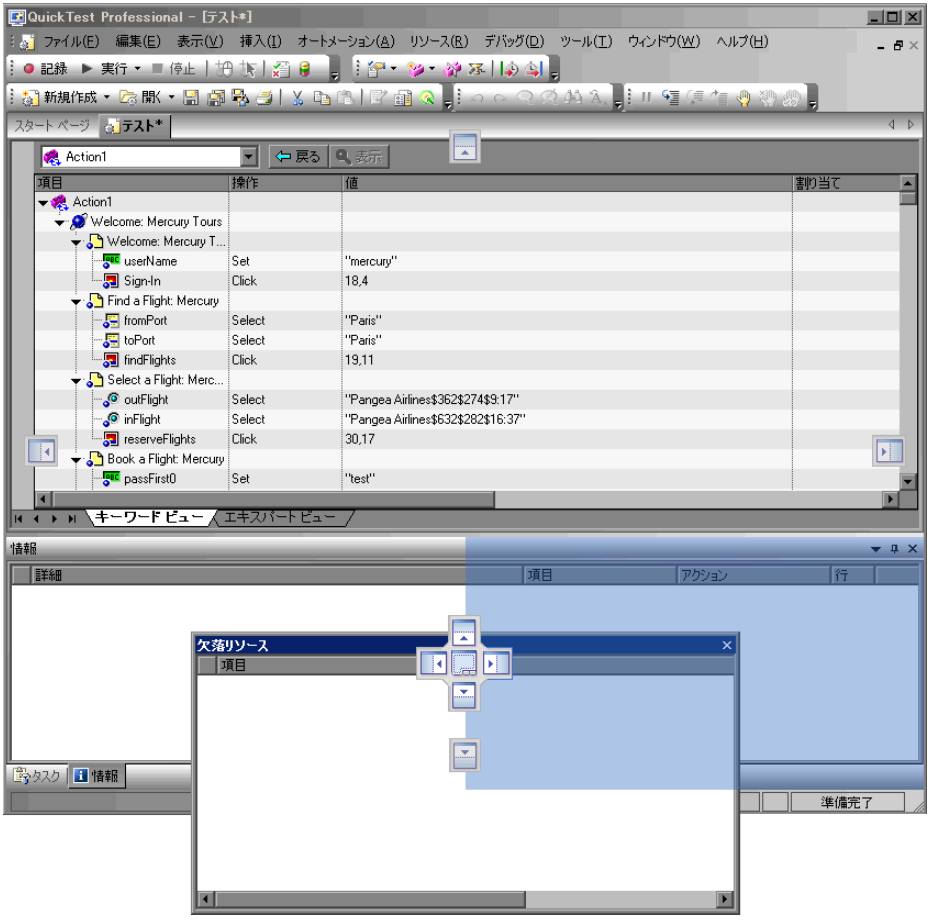
ヒント：

- タブ形式の表示枠を 1 つだけ移動するには，タブのラベルをドラッグします。タブ形式の表示枠をドラッグし始めると，そのタブが消え，タイトル・バーが表示されます。
- タブ形式の表示枠をすべて移動するには，アクティブなタブ形式の表示枠のタイトル・バーをドラッグします。

次のマーカが表示されます。

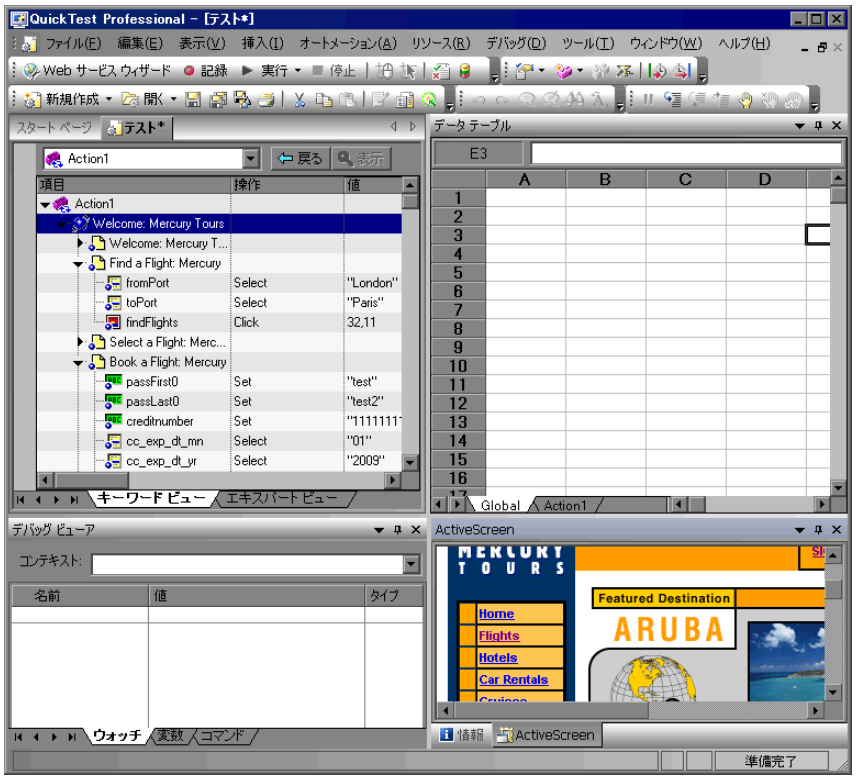
タイプ	マーカ	説明
現在表示枠マーカ		次のことが可能です。 ➤ マウス・ボタンを放したときに選択されていた矢印マーカに応じて，移動した表示枠を，アクティブな表示枠内の上下左右いずれかの側，または中央に，新しい表示枠として配置します。 ➤ 中央のマーカが選択されているときにマウス・ボタンを放して，移動した表示枠を，アクティブなウィンドウ内に新しいタブ形式の表示枠として配置します。 **注**：中央のマーカは，移動する表示枠を，（ドキュメント表示枠を除く）既存の表示枠内にドラッグしている場合にのみ表示されます。
ウィンドウ表示枠マーカ		QuickTest ウィンドウの上部に表示枠を配置します。
		QuickTest ウィンドウの右側に表示枠を配置します。
		QuickTest ウィンドウの下部に表示枠を配置します。
		QuickTest ウィンドウの左側に表示枠を配置します。

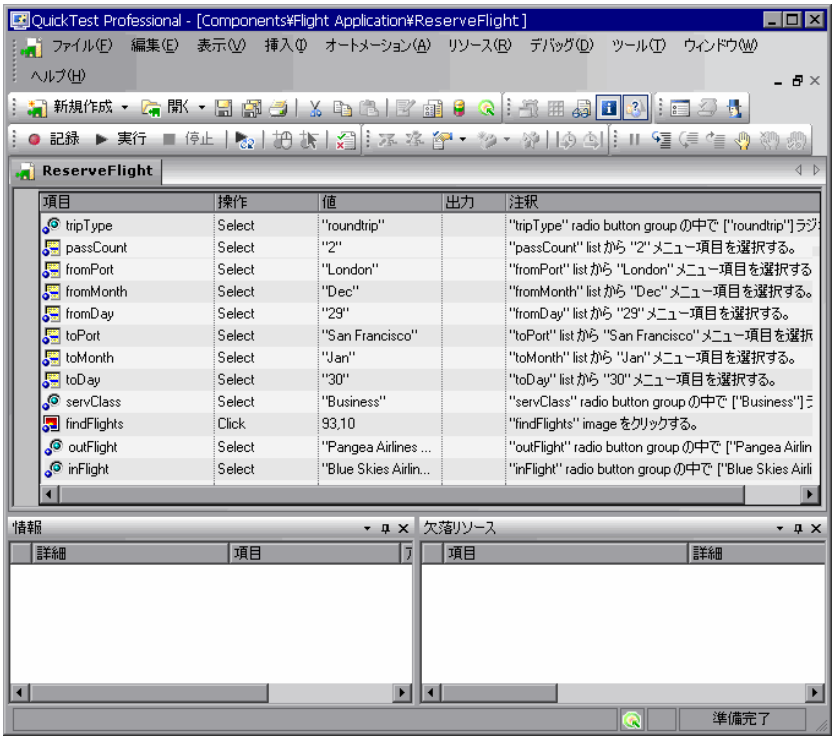
2 次の図のように，［欠落リソース］表示枠をドラッグし，アクティブな表示枠の右矢印マーカの上にカーソルを置きます。表示枠の新しい位置を示すグレーの領域が表示されます。



3 マウス・ボタンを放します。［欠落リソース］表示枠が固定され，グレー表示

されていた領域に新しい表示枠として表示されます。





ヒント：表示枠は，QuickTest ウィンドウ上または画面上の任意の場所にフローティング表示枠として表示することもできます。フローティング表示枠の詳細については，807 ページ「表示枠の表示と非表示」を参照してください。

---

4 移動する各表示枠についてこの手順を繰り返します。

表示枠の表示と非表示

表示枠を標準の位置に移動した後は，表示枠を常に表示するか，それとも自動非表示にして必要なときだけ表示するかを選択できます。

表示枠はドッキングかフローティングのいずれかの状態になります。

- **ドッキング表示枠**：ドッキング表示枠は，アプリケーション内の特定の位置に固定されます。たとえば，マーカが示す位置に表示枠を移動すると，表示枠はその位置にドッキングされます。

  ドッキング表示枠は，QuickTest ウィンドウに常に表示しておくか，それとも自動非表示にするかを選択できます。自動非表示にすると，QuickTest ウィンドウの端にサイドタブが表示され，目的のタブの上にカーソルを置いたときだけ該当する表示枠が表示されます。別の表示枠またはサイドタブを選択すると，自動非表示の表示枠は閉じ，サイドタブとして表示されます。

注：［情報］表示枠は，自動非表示に設定されている場合でも，テスト・スクリプトで構文エラーが検出されると自動的に表示されます。

自動非表示の表示枠を開くと，標準設定では，QuickTest ウィンドウ上での位置に応じた 1 つの辺全体がその表示枠の表示に使用されます。たとえば，QuickTest ウィンドウの右辺に配置したドッキング表示枠は，QuickTest ウィンドウの右端にサイドタブとして表示され，選択すると QuickTest ウィンドウの右辺いっぱいに開きます。

ヒント：タブ形式の表示枠をすべて自動非表示にするには，アクティブなタブ形式の表示枠のタイトル・バーを選択し，右クリックして［**自動非表示**］を選択します。タブ形式の表示枠はサイドタブのグループとして QuickTest ウィンドウの端に表示され，サイドタブの上にカーソルを置いたときだけ該当する表示枠が表示されます。

- **フローティング表示枠**：フローティング表示枠は，ほかのすべてのウィンドウの手前に表示されます。これらの表示枠は，画面上の任意の位置にドラッグすることができ，QuickTest ウィンドウの外にもドラッグできます。各フローティング表示枠には専用のタイトル・バーが表示されます。

注：フローティング表示枠を自動非表示にしたり，個別のタブ形式の表示枠を自動非表示したりはできません。

表示枠の表示と非表示を切り替えるには，次の手順を実行します。

QuickTest ウィンドウで，自動非表示にする表示枠を選択し，QuickTest ウィンドウのいずれかの端にサイドタブとして表示します。タイトル・バーには次のボタンが表示される場合があります。

ボタン	説明
	［**メニュー**］ボタンでは次を選択できます。 ➤［**フローティング**］：表示枠は，専用のタイトル・バーが付けられ，ほかのすべてのウィンドウおよび表示枠の手前に表示されます。 ➤［**ドッキング**］：表示枠は，QuickTest ウィンドウにドッキングされます。 ➤［**自動非表示**］：表示枠が，QuickTest ウィンドウ内での位置に応じて，QuickTest ウィンドウの上下左右いずれかの端にサイドタブとして表示されます。 ➤［**非表示**］：表示枠を閉じます。
	［**自動非表示**］ボタン：表示枠を非表示にします。 表示枠は，QuickTest ウィンドウ上での位置に応じて，QuickTest ウィンドウの上下左右いずれかの端にサイドタブとして表示されます。 表示枠を表示するには，サイドタブの上にカーソルを置きます。このボタンは，下記の［**ドッキング**］ボタンに切り替わります。
	［**自動非表示**］ボタン（横向き）は表示枠を QuickTest ウィンドウにドッキングし， 非表示になる前に配置されていた場所に戻します。また，ボタンは，前述の［**自動非表示**］ボタンに切り替わります。
	［**閉じる**］ボタンは，表示枠を閉じます。 表示枠は QuickTest ウィンドウから取り除かれます。表示枠を再度開くには，［**表示**］メニューからその表示枠を選択します。 **ヒント**：表示枠を右クリックし，ショートカット・メニューから［**非表示**］を選択して閉じることもできます。

ヒント：

- タブ形式の表示枠をすべて自動非表示にするには，アクティブなタブ形式の表示枠のタイトル・バーを選択し，右クリックして［**自動非表示**］を選択します。タブ形式の表示枠はサイドタブのグループとして QuickTest ウィンドウの端に表示され，サイドタブの上にカーソルを置いたときだけ該当する表示枠が表示されます。
- 表示枠のタイトル・バーを右クリックし，ショートカット・メニューから［**フローティング**］を選択すると，表示枠はフローティング形式になり，専用のタイトル・バーが付けられて，ほかのすべてのウィンドウおよび表示枠の手前に表示されます。表示枠のタイトル・バーをダブルクリックするか，タイトル・バーを右クリックして［**ドッキング**］を選択すると，その表示枠はドッキングし，QuickTest ウィンドウ内の元の位置に戻ります。

ツールバーのフローティングとドッキング

ツールバーの左側にあるツールバー・ハンドルの上にカーソルを置いて任意の位置にドラッグすると，そのツールバーはフローティング形式になり，タイトル・バー付きで表示されます。

メニューのタイトル・バーをダブルクリックすると，メニューがドッキングし，QuickTest ウィンドウ内の元の位置に戻ります。また，［**閉じる**］ボタンをクリックすると，メニューが閉じます。

QuickTest ウィンドウの標準のレイアウトの復元

すべてのドキュメント・タイプについて，いつでも QuickTest ウィンドウの標準のレイアウトを復元できます。

標準のレイアウトを復元するには，次の手順を実行します。

1 ［**ツール**］＞［**オプション**］＞［**一般**］ノードを選択します。［オプション］ダイアログ・ボックスが表示されます。

2 ［一般］表示枠で，［**レイアウトの復元**］ボタンをクリックします。すべてのドキュメント・タイプの表示枠とツールバーが，標準のサイズと位置に復元されます。

［オプション］ダイアログ・ボックスの詳細については，第 24 章「グローバル・テスト・オプションの設定」を参照してください。

QuickTest ウィンドウ・レイアウトのカスタマイズ

QuickTest は，複数の異なるモード（表示 / 編集，記録，および実行）で動作します。モードの機能に対応するように QuickTest のレイアウトを変更できます。たとえば，記録時には QuickTest を部分的に見えるようにすると便利な場合があります。コンポーネントの実行時には，多くの場合 QuickTest を最小化して，コンポーネント実行中にアプリケーションを見えるようにすると便利です。コンポーネントを表示または編集するときには，QuickTest ウィンドウを最大化して，すべての表示枠を表示すると便利な場合があります。

QuickTest は，モードごとにメイン・ウィンドウおよびすべての表示枠のサイズと場所を記憶します。QuickTest でいずれかのモードを開始すると，そのモードで最後に使用したレイアウトが復元されます。つまり，QuickTest のメイン・ウィンドウと各表示枠は，現在のモードの前回のレイアウトに基づいて，最小化，最大化，またはサイズ変更されます。

QuickTest の各モードのレイアウトを設定するには，次の手順を実行します。

1 記録モードを設定します。

a 新規または既存のコンポーネントを開きます。

b 記録セッションを開始します。

c 1 ステップを記録します。

d 記録モードのレイアウト設定をすべて行います。

e 記録セッションを停止します。

2 実行モードを設定します。

a テストの最初のステップの前にブレークポイントを設定します。これにより，実行セッション中にレイアウトを変更できます。ブレークポイントの設定方法の詳細については，749 ページ「ブレークポイントの設定」を参照してください。

b コンポーネントを実行します。

c QuickTest がブレークポイントに達したら，実行モードのレイアウト設定をすべて行います。

d 実行セッションを停止します。

3 表示 / 編集モードのレイアウト設定をすべて行います。

これで，これらすべてのモードのレイアウトを設定できました。QuickTest でこれらのモードのいずれかが開始されるたびに，関連するレイアウトが適用されます。

ツールバーとメニューのカスタマイズ

ユーザ定義メニューの作成，および，既存のメニューとツールバーの表示形式のカスタマイズを行うには，［カスタマイズ］ダイアログ・ボックスを使用します。

本項の内容

- 811 ページ「カスタマイズ・モード・オプション」
- 813 ページ「［ボタンの表示］ダイアログ・ボックス」
- 814 ページ「［カスタマイズ］ダイアログ・ボックス - ［コマンド］タブ」
- 817 ページ「［カスタマイズ］ダイアログ・ボックス - ［ツールバー］タブ」
- 819 ページ「［カスタマイズ］ダイアログ・ボックス - ［ツール］タブ」
- 821 ページ「［カスタマイズ］ダイアログ・ボックス - ［オプション］タブ」
- 823 ページ「ツールバーとメニューのカスタマイズの注意事項」

カスタマイズ・モード・オプション

［カスタマイズ］ダイアログ・ボックスが開いている間，QuickTest はカスタマイズ・モードになります。カスタマイズ・モードのときにメニュー・バーまたはツールバー・ボタンを右クリックすると，ショートカット・メニューで次のオプションを使用できるようになります。

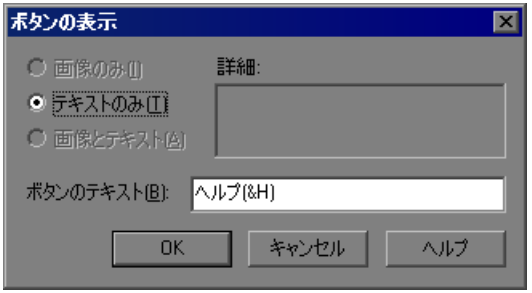
オプション	説明
［標準設定に戻す］	ボタンの標準設定に戻します。メニューに対しては，このオプションは無効です。
［ボタンの画像をコピーする］	ボタンの画像をクリップボードにコピーします。標準の画像がない項目に対しては，このオプションは無効です。
［削除］	メニューまたはボタンを削除します。 **ボタンを復元するには，次の手順を実行します。** 1 通常モードのときにカスタマイズ・ツールバー・ボタン をクリックします。 2 ［**ボタンの追加と削除**］を選択します。 3 ボタンを復元するメニューを選択し，［**ツールバーを元に戻す**］を選択します。 **メニューをメニュー・バーから復元するには，次の手順を実行します。** ［ツールバー］タブの［**メニュー バー**］を選択し，［**選択対象に戻す**］ボタンをクリックします。 **注**：メニュー・バーのカスタマイズは失われます。 詳細については，817 ページ「[カスタマイズ］ダイアログ・ボックス - ［ツールバー］タブ」を参照してください。
［ボタンの表示］	［ボタンの表示］ダイアログ・ボックスが開きます。詳細については，813 ページ「[ボタンの表示］ダイアログ・ボックス」を参照してください。
［画像］	ツールバーまたはメニューのボタンの画像が表示されます。標準の画像がない項目に対しては，このオプションは無効です。
［テキスト］	ツールバーまたはメニューのボタンのテキスト・ラベルが表示されます。メニューに対しては，このオプションは無効です。

オプション	説明
［画像とテキスト］	ツールバーまたはメニューのボタンあるいはメニューの画像とテキスト・ラベルが表示されます。標準の画像がない項目に対しては，このオプションは無効です。
［グループを開始する］	新しいボタン・グループであることを示すために，ツールバーまたはメニューの現在のボタンの前に境界線を引きます。

［ボタンの表示］ダイアログ・ボックス

説明	ボタンまたはメニューの表示形式を変更できます。
アクセス方法	次の手順のいずれかを実行します。 ➤ ［**ツール**］＞［**カスタマイズ**］メニュー・コマンドを選択し，ボタンまたはメニューを右クリックして，［**ボタンの表示**］を選択します。 ➤ カスタマイズ・ツールバー・ボタンをクリックし，［**ボタンの追加と削除**］＞［**カスタマイズ**］を選択します。次に，ボタンまたはメニューを右クリックし，［**ボタンの表示**］を選択します。 ➤ メニュー・バーまたは任意のツールバーを右クリックし，［**カスタマイズ**］を選択します。次に，ボタンまたはメニューを右クリックし，［**ボタンの表示**］を選択します。

［ボタンの表示］ダイアログ・ボックスの画像を次に示します。



［ボタンの表示］ダイアログ・ボックスのオプション

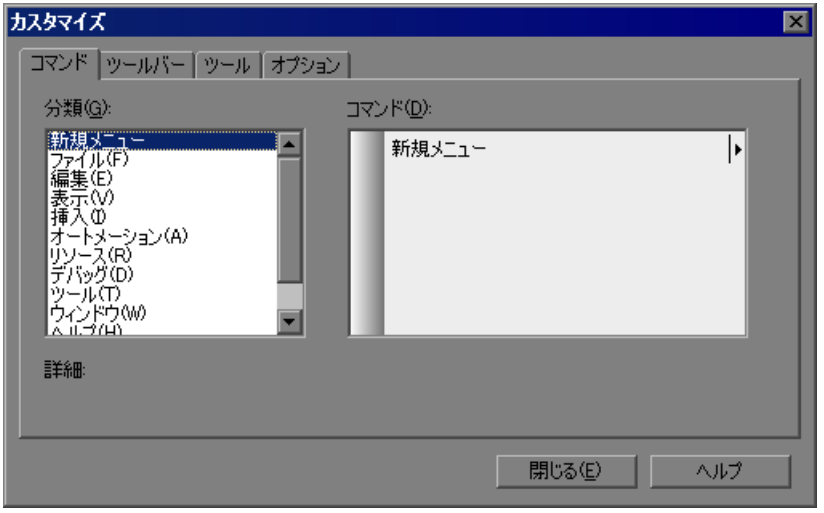
オプション	説明
画像のみ	ツールバーまたはメニューのボタンの画像が表示されます。標準の画像がない項目に対しては，このラジオ・ボタンは無効です。
テキストのみ	ツールバーまたはメニューのボタンのテキスト・ラベルが表示されます。
画像とテキスト	ツールバーまたはメニューのボタンあるいはメニューの画像とテキスト・ラベルが表示されます。標準の画像がない項目に対しては，このラジオ・ボタンは無効です。
詳細	ボタンの説明。
ボタンのテキスト	ボタンまたはメニューのテキスト・ラベル。［**テキストのみ**］ラジオ・ボタンまたは［**画像とテキスト**］ラジオ・ボタンが選択されている場合に，ボタンのテキスト・ラベルを変更できます。 ボタン・テキストには，ニーモニック（キーボードでの移動のための下線付き文字）を作成できます。ボタンのテキスト・ラベルでニーモニックとして指定する文字の前に **&** を追加します。各ボタン・テキストには，ニーモニックを 1 つだけ指定できます。

［カスタマイズ］ダイアログ・ボックス - ［コマンド］タブ

説明	ツールバーとメニューのカスタマイズ，および，新規メニューの作成が行えます。
アクセス方法	次の手順のいずれかを実行します。 ➤ ［**ツール**］＞［**カスタマイズ**］メニュー・コマンドを選択し，［**コマンド**］タブをクリックします。 ➤ カスタマイズ・ツールバー・ボタンをクリックし，［**ボタンの追加と削除**］＞［**カスタマイズ**］を選択して，［**コマンド**］タブをクリックします。 ➤ メニュー・バーまたは任意のツールバーを右クリックし，［**カスタマイズ**］を選択して，［**コマンド**］タブをクリックします。

重要情報	➤ 参照先 ➤ 811 ページ「カスタマイズ・モード・オプション」 ➤ 823 ページ「ツールバーとメニューのカスタマイズの注意事項」 **標準のボタンを既存のツールバーに追加したり，既存のツールバーから削除したりするには，次の手順も実行できます。** 1 カスタマイズ・ツールバー・ボタンを右クリックします。 2 ［**ボタンの追加と削除**］を選択します。 3 ボタンを変更するメニューを選択します。 4 特定のボタンを選択または選択解除します。 ［**ボタンの追加と削除**］の選択内容に行単位でツールバーが一覧表示されます。
詳細情報	**主要なタスク**：816 ページ「ツールバーまたはメニューにコマンドを追加するには，次の手順を実行します。」

［カスタマイズ］ダイアログ・ボックス - ［コマンド］タブの画像を次に示します。



［カスタマイズ］ダイアログ・ボックス - ［コマンド］タブのオプション

セクション	説明
分類	メニュー・バーのすべてのメニュー項目のリスト。「**新規メニュー**」と「**すべてのコマンド**」も追加されています。
コマンド	［**カテゴリ**］リストで選択されたメニュー項目で使用可能なすべてのコマンドのリスト。ドロップダウン・リストまたはサブメニューに表示されるコマンドが，個別のコマンドとして［**コマンド**］セクションに一覧表示されます。たとえば，標準ツールバーの［**新規作成**］ドロップダウン・リストの［**ビジネス コンポーネント**］は，［**ファイル**］カテゴリの個別のコマンド**新規作成：ビジネス・コンポーネント**」として表示されます。
詳細	［**コマンド**］リストで選択されたコマンドの説明。

ツールバーまたはメニューにコマンドを追加するには，次の手順を実行します。

1 ［**ツール**］＞［**カスタマイズ**］を選択し，［**コマンド**］タブをクリックします。

2 ［**カテゴリ**］リストで，ツールバーに追加するコマンドが含まれるメニューの名前を探して選択します。使用可能なすべてのコマンドをアルファベット順で表示するには，「**すべてのコマンド**」を選択します。

3 ［**コマンド**］リストで，追加するコマンドを選択し，そのコマンドをツールバーまたはメニュー・バーにドラッグします。

4 メニュー・バーまたはいずれかのツールバーの上にコマンドを載せると，コマンドが配置される場所を示すマーカが表示されます。コマンドを追加する場所までマーカをドラッグし，マウス・ボタンを放します。

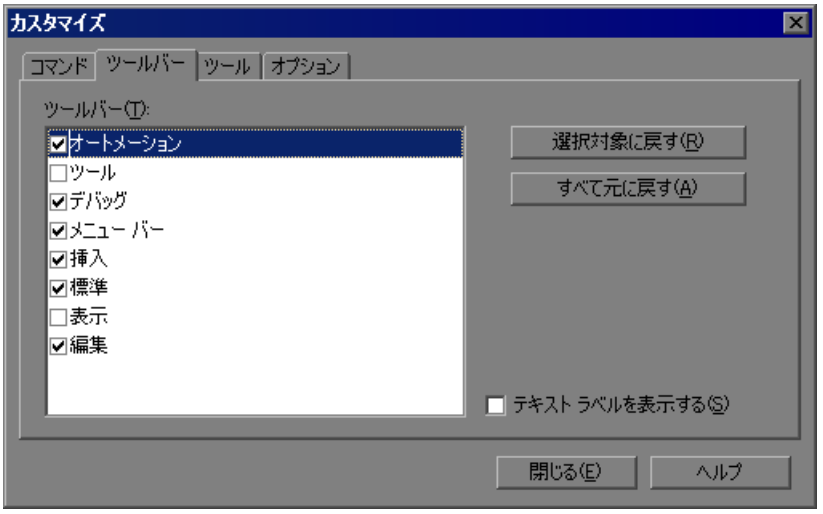
5 新しいメニューを作成する場合は，［**カテゴリ**］リストの「**新規メニュー**」を選択し，その「**新規メニュー**」項目をメニュー・バーまたはツールバーまでドラッグします。新しいメニューの名前の作成方法については，813 ページ「[ボタンの表示］ダイアログ・ボックス」を参照してください。

6 既存のメニューにコマンドを追加する場合は，そのメニュー項目の上にコマンドをドラッグします。すると，メニュー項目が広がります。メニュー内の，コマンドを追加する場所までマーカをドラッグし，マウス・ボタンを放します。

［カスタマイズ］ダイアログ・ボックス - ［ツールバー］タブ

説明	次のことが可能です。 ➤ ツールバーまたはメニュー・バーの表示と非表示を切り替えます。 ➤ 1 つまたはすべてのツールバーあるいはメニュー・バーの標準設定を復元します。 ➤ ツールバー・ボタンのテキスト・ラベルを表示します。
アクセス方法	次の手順のいずれかを実行します。 ➤［**ツール**］＞［**カスタマイズ**］メニュー・コマンドを選択し，［**ツールバー**］タブをクリックします。 ➤ カスタマイズ・ツールバー・ボタンをクリックし，［**ボタンの追加と削除**］＞［**カスタマイズ**］を選択して，［**ツールバー**］タブをクリックします。 ➤ メニュー・バーまたは任意のツールバーを右クリックし，［**カスタマイズ**］を選択して，［**ツールバー**］タブをクリックします。
重要情報	➤ ツールバーの表示と非表示の切り替えは，［**表示**］＞［**ツールバー**］メニュー・オプションを使用して行うこともできます。また，ツールバーの領域を右クリックし，ショートカット・メニューからツールバーを選択または選択解除するという方法もあります。 ➤ 参照先 ➤ 811 ページ「カスタマイズ・モード・オプション」 ➤ 823 ページ「ツールバーとメニューのカスタマイズの注意事項」

［カスタマイズ］ダイアログ・ボックス - ［ツールバー］タブの画像を次に示します。



［カスタマイズ］ダイアログ・ボックス - ［ツールバー］タブのオプション

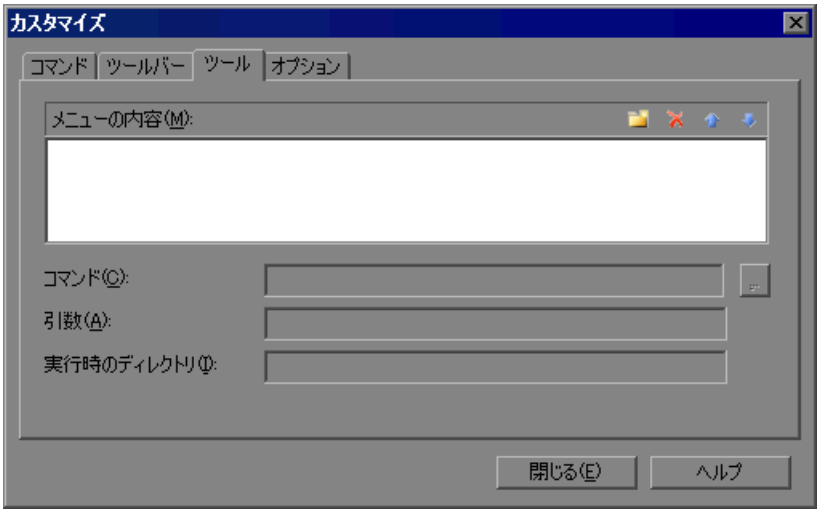
オプション	説明
ツールバー	QuickTest ウィンドウのツールバーのリスト。「**メニュー・バー**」も追加されています。ツールバーの表示と非表示を切り替えるには，チェック・ボックスを選択または選択解除します。
選択対象に戻す	選択したツールバーまたはメニュー・バーの標準のレイアウトを復元します。 **ツールバーの標準のレイアウトを復元するには，次の手順も実行できます。** 1 カスタマイズ・ツールバー・ボタンを右クリックします。メニュー・バーの場合，この方法は使用できません。 2 ［**ボタンの追加と削除**］を選択します。 3 レイアウトを復元するツールバーを選択します。 4 ［**ツールバーを元に戻す**］を選択します。 ［**ボタンの追加と削除**］の選択内容に行単位でツールバーが一覧表示されます。

オプション	説明
すべて元に戻す	すべてのツールバーの標準のレイアウトを復元します。
テキスト ラベルを表示する	現在強調表示されているツールバーのボタンのテキスト・ラベルを表示します。標準設定でテキスト・ラベルが付いているボタンの場合（たとえば，［**実行**］ボタン），このチェック・ボックスをクリアすると，標準の表示が復元され，引き続きテキスト・ラベルも表示されます。 ツールバーのテキスト・ラベルを無効にするには，［**ツールバー**］領域でツールバーを強調表示し，このチェック・ボックスの選択を解除します。 メニュー・バー・ツールバーの場合，このチェック・ボックスは無効になります。

［カスタマイズ］ダイアログ・ボックス - ［ツール］タブ

説明	QuickTest のメニューからアプリケーションを起動できるように，［ツール］メニューに項目を追加できます。
アクセス方法	次の手順のいずれかを実行します。 ➤ ［**ツール**］＞［**カスタマイズ**］メニュー・コマンドを選択し，［**ツール**］タブをクリックします。 ➤ カスタマイズ・ツールバー・ボタンをクリックし，［**ボタンの追加と削除**］＞［**カスタマイズ**］を選択して，［**ツール**］タブをクリックします。 ➤ メニュー・バーまたは任意のツールバーを右クリックし，［**カスタマイズ**］を選択して，［**ツール**］タブをクリックします。
重要情報	参照先 ➤ 811 ページ「カスタマイズ・モード・オプション」 ➤ 823 ページ「ツールバーとメニューのカスタマイズの注意事項」

［カスタマイズ］ダイアログ・ボックス - ［ツール］タブの画像を次に示します。



［カスタマイズ］ダイアログ・ボックス - ［ツール］タブのオプション

オプション	説明
メニューの内容	［ツール］メニューに追加された項目のリスト。
	［**新規作成**］：［**ツール**］メニューに新しい項目を追加できます。［**メニューの内容**］領域に空白行が追加されます。新しい項目の名前を入力します。
	［**削除**］：［**メニューの内容**］リストで選択した項目を［**ツール**］メニューから削除できます。
	［**項目を上に移動**］：選択した項目を［**ツール**］メニュー内で上に移動できます。
	［**項目を下に移動**］：選択した項目を［**ツール**］メニュー内で下に移動できます。

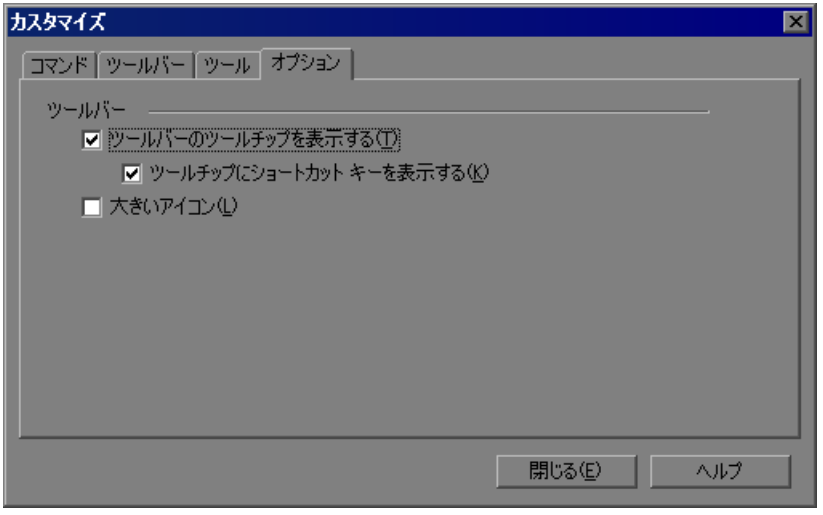
オプション	説明
コマンド	［ツール］メニューに項目を追加するアプリケーション。参照ボタン をクリックし，追加するアプリケーションに移動します。 **注**：［**コマンド**］ボックスにはファイル名およびアプリケーションのパスのみ含まれるようにします。コマンド・ライン引数を追加する場合は，［**引数**］ボックスを使用します。 **ヒント**：アプリケーションに関連付けられている，ファイル・システム内のドキュメントまたはそれ以外のファイルを指定できます（例：c:¥tmp¥a.txt）。この場合，指定されたファイルが，関連付けられているアプリケーション（この例ではメモ帳）で自動的に開きます。このオプションを使用した場合，定義されたプログラム引数は無視されます。
引数	**任意**。指定されたコマンド・ライン引数を使用してアプリケーションを開くように QuickTest を設定します。
実行時のディレクトリ	**任意**。アプリケーションの現在作業中のフォルダを指定します。実行時のディレクトリは，関連ファイルの検索のためにアプリケーションによって使用されます。実行時のディレクトリが指定されていない場合は，実行可能フォルダが実行時のディレクトリとして使用されます。

［カスタマイズ］ダイアログ・ボックス - ［オプション］タブ

説明	QuickTest 画面にツールチップ，ショートカット・キー，大きいアイコンまたは小さいアイコンを表示できます。

アクセス方法	次の手順のいずれかを実行します。 ➤［**ツール**］＞［**カスタマイズ**］メニュー・コマンドを選択し，［**オプション**］タブをクリックします。 ➤ カスタマイズ・ツールバー・ボタンをクリックし，［**ボタンの追加と削除**］＞［**カスタマイズ**］を選択して，［**オプション**］タブをクリックします。 ➤ メニュー・バーまたは任意のツールバーを右クリックし，［**カスタマイズ**］を選択して，［**オプション**］タブをクリックします。
重要情報	参照先 ➤ 811 ページ「カスタマイズ・モード・オプション」 ➤ 823 ページ「ツールバーとメニューのカスタマイズの注意事項」

［カスタマイズ］ダイアログ・ボックス - ［オプション］タブの画像を次に示します。



［カスタマイズ］ダイアログ・ボックス - ［オプション］タブのオプション

オプション	説明
ツールバーのツールチップを表示する	ツールチップを有効または無効にします。QuickTest 画面にツールチップを表示するには，このチェック・ボックスを選択します。 ➤［**ツールチップにショートカット キーを表示する**］：ツールチップにショートカット・キーを表示するには，このチェック・ボックスを選択します。
大きいアイコン	大きいアイコンを有効または無効にします。QuickTest 画面に大きいアイコンを表示するには，このチェック・ボックスを選択します。

ツールバーとメニューのカスタマイズの注意事項

- ツールバーとメニューのカスタマイズ設定は，Windows ユーザごとに作成および保存します。
- ［カスタマイズ］ダイアログ・ボックスが開いている間は，ボタンもコマンドも削除できます。削除するツールバー・ボタンをツールバーからツールバー領域外にドラッグします。すると，ツールバー・ボタンが削除されます。
- 選択したツールバーまたはすべてのツールバーの標準のボタンおよびレイアウトを復元するには，［ツールバー］タブの［**戻す**］ボタンまたは［**すべて元に戻す**］ボタンを使用します。また，カスタマイズ・ツールバー・ボタンを右クリックして［**ボタンの追加と削除**］を選択し，設定を復元するツールバーを選択して，［**ツールバーを元に戻す**］を選択するという方法でも，ツールバーの標準のボタンおよびレイアウトを復元できます。
- ［カスタマイズ］ダイアログ・ボックスが開いている間，ツールバー・ボタンをあるツールバーから別のツールバーにドラッグしたり，メニューの項目をドラッグ・アンド・ドロップして順序を変えたりできます。
- 一部の QuickTest アドインでは，QuickTest ウィンドウにコマンドまたはメニューが追加されます。アドインを使用していて，ツールバーをカスタマイズする場合は，次のことを考慮してください。
  - アドインを使った作業を続けるかぎり，QuickTest を閉じて再度開いたとしても，カスタマイズは QuickTest に記憶されます。
  - アドインなしで QuickTest が実行された場合，アドインによって追加されたコマンドとメニューはすべて，QuickTest ウィンドウから削除されます。

➤ 最初にツールバーをカスタマイズし，次にアドインが追加された QuickTest を実行した場合，追加のコマンドとメニューは，隣接する項目に基づき，意図された場所のできるだけ近くに配置されます。

複数のドキュメントを使った作業

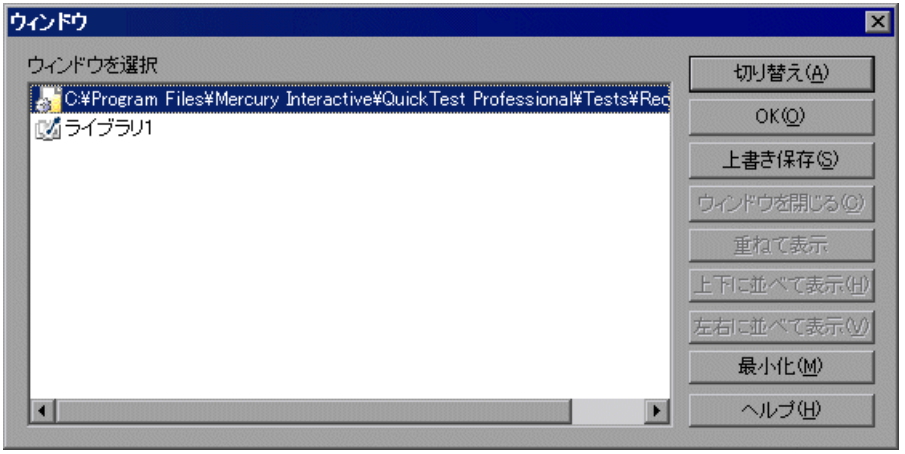
QuickTest では，1 つのコンポーネントまたはアプリケーション領域を開いて作業できるのに加え，複数の関数ライブラリを同時に開いて操作できます。現在開いているコンポーネントまたはアプリケーション領域に関連付けられているかどうかに関係なく，任意の関数ライブラリを開くことができます。

［**ウィンドウ**］メニュー・オプションでは，開いているドキュメント・ウィンドウの選択とアクティブ化（フォーカスの移動），開いているドキュメント・ウィンドウを QuickTest ウィンドウ内に配置する方法の選択，開いているすべての関数ライブラリ・ウィンドウの終了を行うことができます。

また，［ウィンドウ］ダイアログ・ボックスを使用して，開いている QuickTest ドキュメント・ウィンドウを管理することもできます。

［ウィンドウ］ダイアログ・ボックスを使用して複数のドキュメントを操作するには，次の手順を実行します。

1 ［**ウィンドウ**］＞［**ウィンドウ**］を選択します。［ウィンドウ］ダイアログ・ボックスが開きます。



［ウィンドウ］ダイアログ・ボックスには，開いているドキュメント・ウィンドウのリストが表示されます。リストには，開いているコンポーネントアプリケーション領域や，現在開いているすべての関数ライブラリ・ウィンドウなどが含まれます。

2 ［ウィンドウ］ダイアログ・ボックスには次のボタンがあり，開いているドキュメントの管理に使用できます。

ボタン	説明
［切り替え］	QuickTest ウィンドウの中で，選択したドキュメントにフォーカスを移動する。
［OK］	［ウィンドウ］ダイアログ・ボックスを閉じる。
［保存］	選択したドキュメントを保存する。
［ウィンドウを閉じる］	選択した関数ライブラリを閉じる。
［重ねて表示］	選択したドキュメントを，部分的に重ねて表示する。
［上下に並べて表示］	選択したドキュメントを，重ならないように横方向に並べて表示する。
［左右に並べて表示］	選択したドキュメントを，重ならないように縦方向に並べて表示する。
［最小化］	選択したドキュメントを最小化する。
［ヘルプ］	このダイアログ・ボックスの QuickTest Professional ヘルプ・トピックを表示する。

3 ［**OK**］をクリックし，［ウィンドウ］ダイアログ・ボックスを閉じます。

第 32 章

リソースの管理

QuickTest では，コンポーネントに関連付けられているリソースの表示およびオープンを 1 つの表示枠で行うことができます。

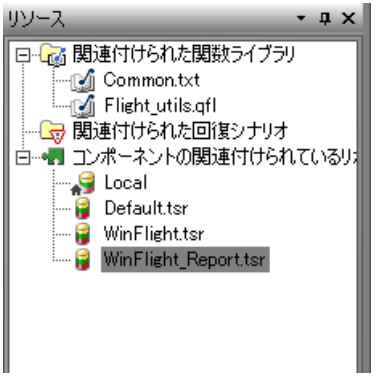
本章の内容

➤ ［リソース］表示枠（827 ページ）

［リソース］表示枠

説明	コンポーネントに関連付けられているほとんどのリソースの表示およびオープンを行うことができます。
アクセス方法	次の手順のいずれかを実行します。 ➤［**表示**］＞［**リソース**］メニュー・オプションを選択します。 ➤［**リソース**］表示枠ツールバー・ボタン をクリックします。
重要情報	コンポーネントは，関数ライブラリ，回復シナリオ，オブジェクト・リポジトリといったリソースに関連付けられています。そのほとんどは，関連付けられているアプリケーション領域を介して管理されます。 ［リソース］表示枠内のリソースは，ツリー階層で表示されます。ツリー内のノードを右クリックすると，そのリソースのショートカット・メニューが開きます。オプションによって，リソースのルート・ノードのショートカット・メニューからアクセスできるものと，特定のリソースのショートカット・メニューからアクセスできるものがあります。

［リソース］表示枠の画像を次に示します。



［リソース］表示枠のツリーのノードの種類

ノードの種類	説明
関連付けのある関数ライブラリ	コンポーネントの関連付けられたアプリケーション領域に現在関連付けられているすべての関数ライブラリが表示されます。 **注**：コンポーネントの場合は，関連付けられたアプリケーション領域を介して関数ライブラリを管理するため，ショートカット・メニュー・オプションの多くが無効になっています。 **ルート・ノードのショートカット・メニュー・オプション** **［関数ライブラリの関連付け］**：コンポーネントに対しては，このオプションは無効になっています。 **関数ライブラリ・ノードのショートカット・メニュー・オプション** ➤ **［関数ライブラリを開く］**：選択された関数ライブラリを QuickTest 関数ライブラリ・ウィンドウで開きます。関数ライブラリをダブルクリックして開くこともできます。 ➤ **［関数ライブラリをリストから削除］**：コンポーネントに対しては，このオプションは無効になっています。 ➤ ［**上に移動**］または［**下に移動**］：コンポーネントに対しては，このオプションは無効になっています。 **以下も参照してください**：389 ページ「ユーザ定義関数および関数ライブラリを使用した作業」

ノードの種類	説明
関連付けられている回復シナリオ	コンポーネントの関連付けられたアプリケーション領域に現在関連付けられているすべての回復シナリオが表示されます。 **注**：コンポーネントの場合は，関連付けられたアプリケーション領域を介して回復シナリオを管理するため，ショートカット・メニュー・オプションのすべてが無効になっています。 コンポーネントの場合は，回復シナリオをダブルクリックして［回復シナリオのプロパティ］ダイアログ・ボックスを開きます。 **ルート・ノードのショートカット・メニュー・オプション** ［**回復シナリオの関連付け**］：コンポーネントに対しては，このオプションは無効になっています。 **回復シナリオ・ノードのショートカット・メニュー・オプション** ➤［**回復シナリオのプロパティ**］：コンポーネントに対しては，このオプションは無効になっています。 ➤［**リストから回復シナリオを削除**］：コンポーネントに対しては，このオプションは無効になっています。 ➤［**上に移動**］または［**下に移動**］：コンポーネントに対しては，このオプションは無効になっています。 ➤［**回復シナリオの無効化**］または［**回復シナリオの有効化**］：コンポーネントに対しては，このオプションは無効になっています。 **以下も参照してください**：867 ページ「回復シナリオの定義と使用」

ノードの種類	説明
コンポーネントの関連付けられているリポジトリ	コンポーネントとその関連付けられたアプリケーション領域に現在関連付けられているすべてのオブジェクト・リポジトリを表示します。 **注**：コンポーネントの場合は，関連付けられたアプリケーション領域を介してオブジェクト・リポジトリを管理するため，ショートカット・メニュー・オプションの多くが無効になっています。 **リポジトリ・ノードのショートカット・メニュー・オプション** ➤［**リポジトリを開く**］：ローカルのオブジェクト・リポジトリに対して［オブジェクト リポジトリ ウィンドウ - ローカル オブジェクト リポジトリ］を開き，共有オブジェクト・リポジトリに対してオブジェクト・リポジトリ・マネージャを開きます。オブジェクト・リポジトリをダブルクリックしても，関連付けられているリポジトリ・ウィンドウが開きます。 ➤［**リストからリポジトリを削除**］：コンポーネントに対しては，このオプションは無効になっています。 ➤［**上に移動**］または［**下に移動**］：コンポーネントに対しては，このオプションは無効になっています。 **以下も参照してください**：151 ページ「オブジェクト・リポジトリ内のテスト・オブジェクトの管理」

第 33 章

コンポーネントへのキーワードの追加

QuickTest では，1 つの表示枠で利用できるキーワードをコンポーネントに表示および追加できます。

本章の内容

➤ ［使用可能なキーワード］表示枠について（831 ページ）

［使用可能なキーワード］表示枠について

［使用可能なキーワード］表示枠には，コンポーネントで利用できるキーワードが表示されます。使用可能なオブジェクトまたは関数呼び出しを表示したり，コンポーネントにドラッグ・アンド・ドロップしたりできます。オブジェクトをコンポーネントにドラッグ・アンド・ドロップすると，QuickTest によって，そのオブジェクトの標準操作を持つステップが挿入されます。関数をコンポーネントにドラッグ・アンド・ドロップすると，QuickTest によって，その関数の呼び出しが挿入されます。

たとえば，ボタン・オブジェクトをコンポーネントにドラッグ・アンド・ドロップすると，**クリック**操作（ボタン・オブジェクトの標準操作）を持つステップがボタンを使用して追加されます。

関数をコンポーネントにドラッグ・アンド・ドロップすると，コメントおよびその関数の呼び出しが追加されます。コメントには，関数に対する呼び出しがテストに追加されたこと，および必要な引数が示されます。その関数の引数をコンポーネントに渡す必要があります。キーワード・ビューでは，関数に必要な引数がツールチップに表示されます。エキスパート・ビューでは，関数に必要な引数が IntelliSense によって表示されます。

ほかの場所からテスト・オブジェクトをドラッグ・アンド・ドロップすることもできます。詳細については，次を参照してください。

- 129 ページ「[オブジェクト リポジトリ] ウィンドウ」
- 242 ページ「オブジェクト・リポジトリ・マネージャの使用によるコンポーネントへのテスト・オブジェクトの追加」

[使用可能なキーワード] 表示枠には，コンポーネントで利用できるキーワードがリソースごとにまたはキーワードごとに並べ替えられて表示されます。

[使用可能なキーワード] 表示枠を表示するには，次の手順を実行します。

[**使用可能なキーワード**] 表示枠ボタンをクリックするか，[**表示**] > [**使用可能なキーワード**] を選択します。

リソースごとに並べ替えられたキーワード

キーワードをリソースごとに並べ替えて表示するには，[**リソースごとに並べ替え**] ボタンをクリックします。キーワードは種類（ライブラリ関数，ローカル関数，オブジェクト）に応じてグループ化され，次にその種類の特定のリソースに応じてグループ化されます。

- 各関数ライブラリ内の関数はアルファベット順に並んでいます。
- 各オブジェクト・リポジトリ内のオブジェクトは，アプリケーション内で表示されるページまたはウィンドウごとにグループ化され，次にオブジェクトの種類ごとにグループ化されます。その後，アルファベット順に並べ替えられます。
- キーワードを右クリックすると，キーワードのリソースを開いたり，または選択したキーワードをクリップボードにコピーしたりできます。
- キーワードをダブルクリックするとキーワードのリソースが開き，選択したキーワードを見つけることができます。

キーワードごとに並べ替えられたキーワード

キーワードをキーワードごとに並べ替えて表示するには，［**キーワードごとに並べ替え**］ボタンをクリックします。キーワードは，リソースに関係なく種類（ライブラリ関数，ローカル関数，オブジェクト）に応じてグループ化されます。

- すべての利用できる関数は，アルファベット順に並べ替えられます。
- 利用できるオブジェクトはすべて，アプリケーション内で表示されるページまたはウィンドウに応じてグループ化され，次にオブジェクトの種類に応じてグループ化されます。その後，アルファベット順に並べ替えられます。

注：同じ名前のキーワードが 2 つある場合は，リソースの優先順位に従って表示されます。

- キーワードを右クリックすると，キーワードのリソースを開いたり，または選択したキーワードをクリップボードにコピーしたりできます。
- キーワードをダブルクリックするとキーワードのリソースが開き，選択したキーワードを見つけることができます。

第 34 章

QuickTest のタスクとコメントの管理

QuickTest では，コンポーネントおよびアプリケーション領域のタスクを作成，管理できます。また，コンポーネント，アプリケーション領域，および関数ライブラリの TODO コメントの作成と管理を行うこともできます。

本章の内容

- タスクおよび TODO コメントを使った作業（835 ページ）
- ［タスク］表示枠（836 ページ）
- ［タスク エディタ］ダイアログ・ボックス（843 ページ）

タスクおよび TODO コメントを使った作業

QuickTest では，コンポーネント，アプリケーション領域，および関数ライブラリで対処が必要な問題に関するタスクと TODO コメントを作成，管理できます。たとえば，引き渡し時に誰かほかの人に指示を与えたり，何かすべきだということを自分で思い出せるようにすることができます。

タスクは，現在開いているコンポーネントまたはアプリケーション領域にリンクしているコンポーネントまたはアプリケーション領域関連のリマインダです。タスクの作成と管理は，［タスク］表示枠および［タスク エディタ］ダイアログ・ボックスで行います。

TODO コメントは，コンポーネントまたは関数ライブラリの該当ステップの横にコメント・ステップとして挿入されるリマインダです。TODO コメントは，［タスク］表示枠からアクセスできます。また，テスト・ドキュメントから直接アクセスすることも可能です。

必要に応じて，Microsoft Excel ファイルまたは XML ファイルにタスクおよび TODO コメントをエクスポートできます。

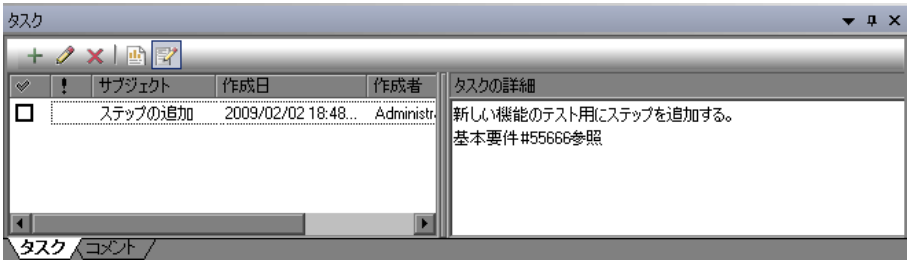
［タスク］表示枠

説明	コンポーネント関連またはアプリケーション領域関連のタスクと TODO コメントを表示および管理できます。 ［タスク］表示枠には次のタブがあります。 ➤ **［タスク］タブ**：コンポーネント関連またはアプリケーション領域関連のタスクの作成および管理を行うことができます。 ➤ **［コメント］タブ**：コンポーネントまたは現在開いている関数ライブラリの TODO コメントの表示およびアクセスを行うことができます。
アクセス方法	➤［**表示**］＞［**タスク**］メニュー項目を選択します。 ➤［**タスク**］表示枠ツールバー・ボタン をクリックします。 **注：** タスクが含まれているコンポーネントまたはアプリケーション領域を開くと，［タスク］表示枠が自動的に開きます。
詳細情報	**概念的概要**：835 ページ「タスクおよび TODO コメントを使った作業」 **その他の関連トピック**：843 ページ「［タスク エディタ］ダイアログ・ボックス」

［タスク］表示枠：［タスク］タブ

説明	［タスク］タブには，現在開いているコンポーネントまたはアプリケーション領域に定義されているすべてのタスクが表示されます。タスクの定義は［タスク エディタ］ダイアログ・ボックスで行います。タスクは，コンポーネントまたはアプリケーション領域とともに保存されます。
アクセス方法	［**表示**］メニュー＞［**タスク**］項目＞［**タスク**］タブ
詳細情報	**概念的概要**：835 ページ「タスクおよび TODO コメントを使った作業」 **その他の関連トピック**： ➤ 840 ページ「［タスク］表示枠：［コメント］タブ」 ➤ 843 ページ「［タスク エディタ］ダイアログ・ボックス」

［タスク］表示枠の［タスク］タブの画像を次に示します。



［タスク］タブの詳細

［タスク］タブには，そのコンポーネントまたはアプリケーション領域のために［タスク エディタ］ダイアログ・ボックスで作成したすべてのタスクが表示されます。

［タスク］タブのツールバー

	ツールバー・オプション	ショートカット・キー	説明
	［タスクの追加］	INSERT	［タスク エディタ］ダイアログ・ボックスが開き（843 ページを参照してください），［タスク］表示枠の［タスク］タブに新しいタスクを追加できるようになります。
	［タスクの編集］	ENTER	［タスク エディタ］ダイアログ・ボックスが開き（843 ページを参照してください），選択したタスクを変更したり，完了のマークを付けたりできるようになります。
	［タスクの削除］	DELETE	選択したタスクを［タスク］表示枠から削除します。
	［TODO リストのエクスポート］	なし	テキスト・ファイルなどの外部ファイルにタスクを保存します。 リストのタスクは次の形式で保存できます。 ➤ XML（Extensible Markup Language） ➤ XLS（Microsoft Excel ファイル） ➤ CSV（カンマ区切り値ファイル）
	［タスクの詳細の表示 / 非表示］	なし	［タスク］表示枠の右側にある［タスクの詳細］表示枠（選択したタスクに関する詳細情報が表示される）を開いたり閉じたりします。

［タスク］タブのカラム

カラム・ヘッダをクリックすると，そのカラムを基準にして並べ替えることができます。

カラム	説明
［完了］	タスクが完全に実装されたかどうかを示します。タスクに完了のマークを付けると，表示枠のそのタスクに，タスクが完了したことを示す取り消し線が適用されます。 取り得る値： ☑ **タスク完了** ☐ **タスク未完了** **ヒント**：［タスク エディタ］ダイアログ・ボックスの［**タスク完了**］チェック・ボックスを選択するという方法でも，タスクに完了のマークを付けることができます。
［優先度］	タスクの重要度を示します。 取り得る値： ➤ **高優先度** ➤ **標準優先度** ➤ **低優先度**
［サブジェクト］	タスクのトピックを示します。
［作成日］	［タスク エディタ］を開いて現在のタスクを作成した日時を示します。
［作成者］	タスクを作成したユーザの名前を示します。
［責任者］	タスクの処理を担当するユーザの名前を示します。
［タスクの詳細］	使用可能になっているときは，タスクの説明テキストが表示されます。

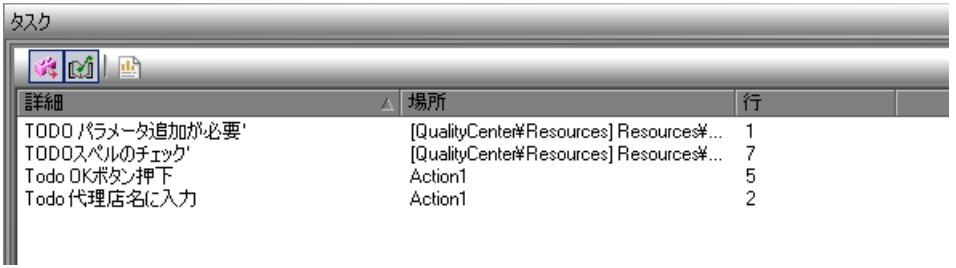
［タスク］タブのショートカット・メニュー・オプション

ショートカット・メニュー・オプション	説明
［並べ替え基準］	タブ上のタスクを並べ替える基準となるカラムを選択できます。
［複製］	選択したタスクのコピーを作成し，［タスク］タブに挿入します。これは，既存のタスクに似た新しいタスクを作成する場合に便利です。
［タスクの削除］	［タスク］表示枠の［タスク］タブからタスクを完全に削除します。

［タスク］表示枠：［コメント］タブ

説明	［コメント］タブには，To Do，todo，to-do，TODO（大文字と小文字は区別されない）といった **to do** という単語の変形で始まるコメント・ステップを表示できます。 **例** ' : to DO need to ask Sarah to add design steps **注** :［コメント］タブに表示されるテキストの文字数は，最大で 260 文字です。テキストの文字数がこの制限を超えているときにコメント全体を表示するには，［コメント］タブのコメント行をダブルクリックして，テスト・ドキュメントのコメントに移動します。 次に関する TODO コメントを表示できます。 ➤ 現在開いているコンポーネント ➤ 開かれている関数ライブラリ
アクセス方法	［**表示**］メニュー＞［**タスク**］項目＞［**コメント**］タブ
詳細情報	**概念的概要** : 835 ページ「タスクおよび TODO コメントを使った作業」 **その他の関連トピック** : ➤ 837 ページ「［タスク］表示枠：［タスク］タブ」 ➤ 843 ページ「［タスク エディタ］ダイアログ・ボックス」

［タスク］表示枠の［コメント］タブの画像を次に示します。



［コメント］タブの詳細

標準設定では，［コメント］タブには現在開いているコンポーネントのすべての TODO コメント・ステップが表示されます。また，現在開いている関数ライブラリにある TODO コメント・ステップが表示されるように選択することもできます。

［コメント］タブのツールバー

	ツールバー・オプション	説明
	［開いている関数ライブラリ内のコメント］	現在開いている関数ライブラリの TODO コメントの表示と非表示を切り替えることができるトグル・ボタン（コンポーネントの TODO コメントに加えて）
	［TODO リストのエクスポート］	テキスト・ファイルなどの外部ファイルに TODO コメントを保存します。 TODO コメントのリストは次の形式で保存できます。 ➤ XML（Extensible Markup Language） ➤ XLS（Microsoft Excel ファイル） ➤ CSV（カンマ区切り値ファイル）

［コメント］タブのカラム

カラム	説明
［説明］	TODO コメントのテキストが表示されます。
［場所］	TODO コメントが含まれるコンポーネントの名前，あるいは，関数ライブラリのパスが示されます。
［行］	コンポーネントまたは関数ライブラリの TODO コメントの行番号が示されます。

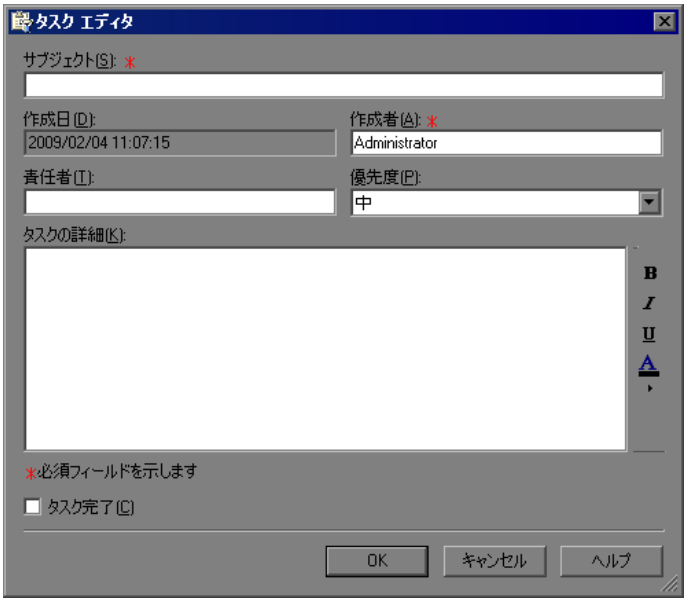
［コメント］タブのショートカット・メニュー・オプション

ショートカット・メニュー・オプション	説明
［並べ替え基準］	タブ上の TODO コメントを並べ替える基準となるカラムを選択できます。
［コメント ラインへ移動］	コンポーネントまたは関数ライブラリのコメント行にカーソルが移動します。 **ヒント**：［コメント］タブのコメントをダブルクリックする，あるいは，ENTER キーを押すという方法でも，コンポーネントまたは関数ライブラリのコメント行にカーソルを移動できます。

［タスク エディタ］ダイアログ・ボックス

説明	［タスク］表示枠へのタスクの追加，既存のタスクの編集，タスクの完了のマーク付けを行うことができます。
アクセス方法	（［タスク］表示枠の［タスク］タブからアクセスします：［**表示**］メニュー＞［**タスク**］項目＞［**タスク**］タブ） ［タスク］タブでは次のいずれかを実行します。 ➤［**タスクの追加**］ボタン をクリックするか，INSERT キーを押します。 ➤ タスクを選択し，［**タスクの編集**］ボタン をクリックするか，ENTER キーを押します。
詳細情報	**概念的概要**：835 ページ「タスクおよび TODO コメントを使った作業」 **その他の関連トピック**：836 ページ「［タスク］表示枠」

［タスク エディタ］ダイアログ・ボックスの画像を次に示します。



［タスク エディタ］ダイアログ・ボックスのオプション

タスクの作成および編集は，［タスク エディタ］ダイアログ・ボックスで行います。赤いアスタリスク（*）が付いているフィールドは必須フィールドです。

オプション	説明
［サブジェクト］	タスクのわかりやすいトピック名。最大で 260 文字の入力が行えます。（必須フィールド）
［作成日］	［タスク エディタ］を開いて現在のタスクを作成した日時。（読み取り専用）
［作成者］	自動的に生成される，タスクを作成したユーザの名前。タスクを変更するのではなく作成する場合は，［**作成者**］フィールドを変更できます。（作成する場合は必須フィールド）
［責任者］	タスクの処理を担当するユーザの名前。
［優先度］	タスクの重要度。 取り得る値： ➤ **高優先度** ➤ **標準優先度** ➤ **低優先度**
［完了］	タスクが完全に実装されたかどうかを示します。 取り得る値： ☑ **タスク完了** ☐ **タスク未完了**
［タスクの詳細］	タスクの説明テキスト。タスクの詳細のさまざまな部分が強調表示されるように，フォント・スタイル（太字，斜体，下線）と色を変更できます。

第 35 章

欠落リソースの処理

欠落リソース（欠落している共有オブジェクト・リポジトリなど）がコンポーネントまたはアプリケーション領域内にある場合や，値が定義されていないリポジトリ・パラメータがテストで使用されている場合は，［欠落リソース］表示枠にそのことが示されます。この表示枠に表示されるリソースのいずれかが実行セッション中に使用できない場合，テストは失敗する場合があります。必要に応じて，欠落リソースを割り当てるか，またはコンポーネントまたはアプリケーション領域からそれを削除できます。

本章の内容

- 欠落リソースの処理について（845 ページ）
- 欠落している関数ライブラリの処理（848 ページ）
- 欠落している共有オブジェクト・リポジトリの処理（850 ページ）
- 欠落している回復シナリオの処理（851 ページ）
- 未割り当ての共有オブジェクト・リポジトリ・パラメータ値の処理（854 ページ）

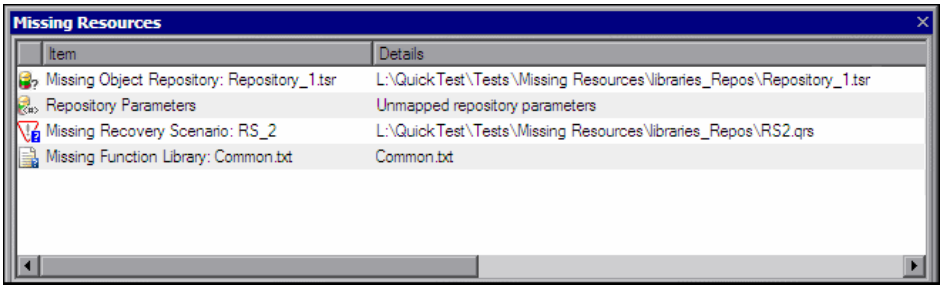
欠落リソースの処理について

QuickTest では，コンポーネントまたはアプリケーション領域を開くたびに，そのコンポーネントまたはアプリケーション領域に指定されたリソースが使用可能かどうかの確認が行われます。

欠落リソースが 1 つでもあると，まだ開いていなければ，［欠落リソース］表示枠が開きます。［欠落リソース］表示枠には，現在使用できないすべてのリソースのリスト，およびリソースを使用できるときに QuickTest によってリソースが検索される場所が表示されます。［欠落リソース］表示枠では，これらを見つけたりアプリケーション領域から削除したりできます。

注：［欠落リソース］表示枠では，コンポーネントの欠落リソースをキーワード・ビューから解決することはできません。割り当てられていないリポジトリ・パラメータを除いては，すべての欠落リソースはアプリケーション領域を介して解決する必要があります。

［欠落リソース］を正常に処理すると，そのリソースは表示枠から削除されます。



［欠落リソース］表示枠には，次のタイプの欠落リソースが表示されます。

- ［**欠落している関数ライブラリ**］：コンポーネントまたはアプリケーション領域が見つからない関数ライブラリに関連付けられている場合は，QuickTest によって，欠落している関数ライブラリを検索するために使用するパスが指定されます。詳細については，848 ページ「欠落している関数ライブラリの処理」を参照してください。
- ［**欠落しているオブジェクト リポジトリ**］：コンポーネントまたはアプリケーション領域が欠落している共有オブジェクト・リポジトリに関連付けられている場合は，欠落しているオブジェクト・リポジトリを探すためのパスが指定されます。詳細については，850 ページ「欠落している共有オブジェクト・リポジトリの処理」を参照してください。
- ［**欠落している回復シナリオ**］：コンポーネントまたはアプリケーション領域が欠落している回復シナリオに関連付けられている場合は，欠落している回復シナリオを探すためのパスが指定されます。詳細については，851 ページ「欠落している回復シナリオの処理」を参照してください。

➤［**リポジトリ パラメータ**］：コンポーネントまたはアプリケーション領域内の少なくとも 1 つのテスト・オブジェクトのプロパティ値が標準設定値のないリポジトリ・パラメータを使用してパラメータ化されている場合，この一般項目が［欠落リソース］表示枠に追加されます。詳細については，854 ページ「未割り当ての共有オブジェクト・リポジトリ・パラメータ値の処理」を参照してください。

注：欠落リソースが使用されるさまざまな画面（キーワード・ビューやテスト設定など）において，QuickTest によって，特別なアイコンやテキストを使ってリソースが欠落していることが示されます。

［欠落リソース］表示枠のフィルタ処理

［欠落リソース］表示枠には，すべての欠落リソースを表示することも，欠落リソースの 1 つのタイプだけを表示することもできます。

表示された欠落リソースのリストをフィルタ処理するには，次の手順を実行します。

［欠落リソース］表示枠を右クリックして，次のいずれかを選択します。

➤［**すべて**］：コンポーネントまたはアプリケーション領域内のすべての欠落リソースが表示されます。

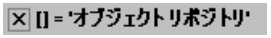
➤［**関数ライブラリ**］：欠落している関数ライブラリごとに，関数ライブラリを検索するために使用するパスを指定する 1 行が表示されます。

➤［**オブジェクト リポジトリ**］：欠落している共有オブジェクト・リポジトリごとに，その共有オブジェクト・リポジトリを探すためのパスを指定する 1 行が表示されます。

➤［**回復シナリオ**］：欠落している回復シナリオごとに，回復シナリオを検索するために使用するパスを指定する 1 行が表示されます。

➤［**リポジトリ パラメータ**］：リポジトリ内の少なくとも 1 つのテスト・オブジェクトに，標準設定値のないリポジトリ・パラメータを使用しているプロパティ値が少なくとも 1 つあることを示す汎用の行が表示されます。

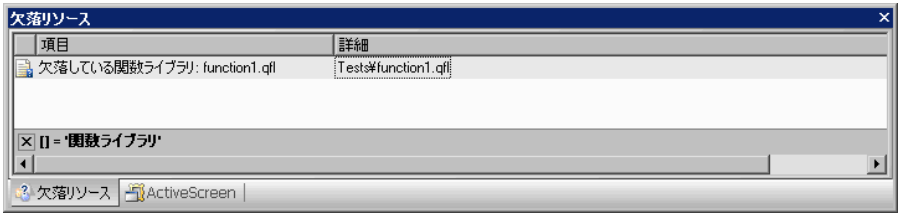
［欠落リソース］表示枠は，選択したリソース・タイプに従ってフィルタ処理され，適用されたフィルタを示す次の表示が表示枠の下部に示されます。



フィルタ表示の左側のアイコン✕をクリックすると，フィルタがキャンセルされ，すべての欠落リソースが再び表示されます。

欠落している関数ライブラリの処理

関数ライブラリが関連付けられているコンポーネントまたはアプリケーション領域を開くと，QuickTest によって，指定したライブラリがアクセス可能であることが確認されます。関数ライブラリが見つからない場合は，コンポーネントまたはアプリケーション領域を開いたときに，見つからない関数ライブラリの名前とパスが［欠落リソース］表示枠に表示されます。

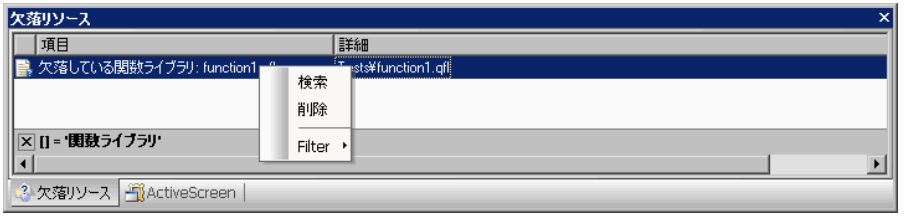


［欠落リソース］表示枠では，欠落している関数ライブラリを見つけるか削除することによって問題を解決できます。

注：［欠落リソース］表示枠では，コンポーネントの欠落リソースをキーワード・ビューから解決することはできません。割り当てられていないリポジトリ・パラメータを除いては，すべての欠落リソースはアプリケーション領域を介して解決する必要があります。

欠落している関数ライブラリを見つけるには，次の手順を実行します。

1 検出する欠落関数ライブラリを右クリックしてショートカット・メニューから［**検索**］を選択するか，検出する欠落関数ライブラリをダブルクリックします。



［関数ライブラリの検索］ダイアログ・ボックスが開きます。

2 アプリケーション領域に関連付ける関数ライブラリを参照して［**開く**］をクリックします。選択した関数ライブラリがアプリケーション領域に関連付けられ，［欠落リソース］表示枠から欠落関数ライブラリが削除されます。

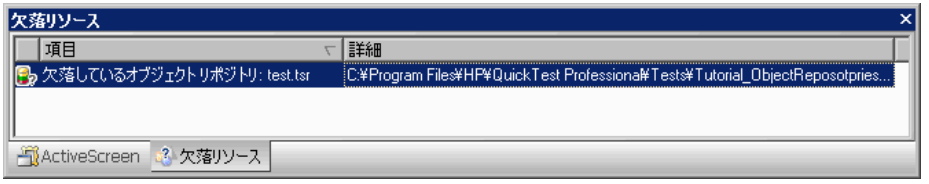
欠落している関数ライブラリを削除するには，次の手順を実行します。

アプリケーション領域で，削除する欠落関数ライブラリを右クリックし，ショートカット・メニューから［**削除**］を選択します。確認メッセージが表示されます。［**OK**］をクリックして関数ライブラリを削除します。アプリケーション領域および［欠落リソース］表示枠から欠落関数ライブラリが削除されます。

注：削除した関数ライブラリの関数の呼び出しも必ず処理してください。関数ライブラリがコンポーネントのアプリケーション領域から削除されても，これらの関数に対する呼び出しはコンポーネントから削除されません。

欠落している共有オブジェクト・リポジトリの処理

共有オブジェクト・リポジトリをアプリケーション領域に関連付けると，指定したリポジトリがアクセス可能かどうかが確認されます。また，コンポーネントまたはアプリケーション領域を開くたびに，関連付けられたすべての共有オブジェクト・リポジトリがアクセス可能かどうかも確認されます。共有オブジェクト・リポジトリが見つからない場合は，コンポーネントまたはアプリケーション領域を開いたときに，見つからないリポジトリの名前とパスが［欠落リソース］表示枠に表示されます。



たとえば，共有オブジェクト・リポジトリの名前やリポジトリが格納されているフォルダを変更した場合は，その共有オブジェクト・リポジトリを関連付けられたアプリケーション領域に割り当てる必要があります。

注：［リポジトリの関連付け］ダイアログ・ボックスを使用して，新しいオブジェクト・リポジトリをテストに関連付けることによって，欠落しているオブジェクト・リポジトリを解決します。欠落しているオブジェクト・リポジトリはテストに関連付けられたままなので，［欠落リソース］表示枠に表示されます。欠落オブジェクト・リポジトリを［欠落リソース］表示枠およびテストから削除するには，［リポジトリの関連付け］ダイアログ・ボックスの［リポジトリの削除］機能を使用する必要があります。

［欠落リソース］表示枠では，コンポーネントの欠落リソースをキーワード・ビューから解決することはできません。割り当てられていないリポジトリ・パラメータを除いては，すべての欠落リソースはアプリケーション領域を介して解決する必要があります。

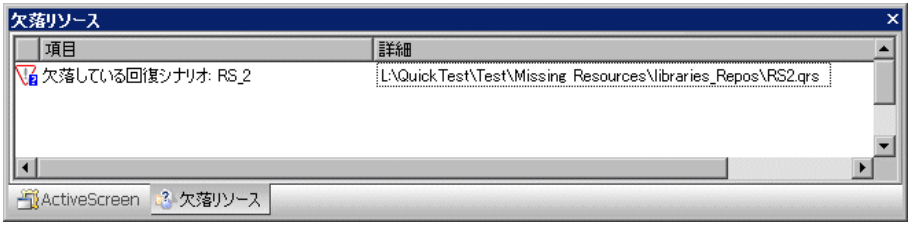
コンポーネントの場合は，欠落しているオブジェクト・リポジトリが表示された行をダブルクリックすると，そのオブジェクト・リポジトリを関連付けられたアプリケーション領域に割り当てる必要があることを示すメッセージが表示されます。この場合，ユーザまたはオートメーション・エンジニアがアプリ

ケーション領域を開き，［オブジェクト リポジトリ］表示枠で共有オブジェクト・リポジトリの関連付けを修正する必要があります。

アプリケーション領域の場合は，欠落しているオブジェクト・リポジトリが表示された行をダブルクリックすると，アプリケーション領域の［オブジェクト リポジトリ］表示枠が開き，必要に応じてオブジェクト・リポジトリの関連付けを修正したり，削除したりできます。詳細については，465 ページ「共有オブジェクト・ライブラリの関連付け」を参照してください。

欠落している回復シナリオの処理

回復シナリオが関連付けられているコンポーネントまたはアプリケーション領域を開くと，QuickTest によって，指定したシナリオがアクセス可能であることが確認されます。回復シナリオが見つからない場合は，コンポーネントまたはアプリケーション領域を開いたときに，見つからないリポジトリの名前とパスが［欠落リソース］表示枠に表示されます。



注：［欠落リソース］表示枠では，コンポーネントの欠落リソースをキーワード・ビューから解決することはできません。割り当てられていないリポジトリ・パラメータを除いては，すべての欠落リソースはアプリケーション領域を介して解決する必要があります。

［欠落リソース］表示枠では，次の操作によって欠落している回復シナリオを解決できます。

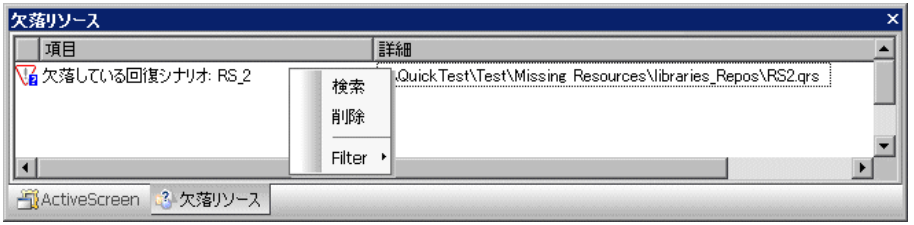
- 欠落回復シナリオの特定
- 欠落している回復シナリオの削除

欠落回復シナリオの特定

［欠落リソース］表示枠では，テスト内で欠落している回復シナリオを探せます。テストに複数の欠落回復シナリオが含まれている場合，回復ファイルで欠落シナリオを見つけると，QuickTest によって，そのファイル内でほかの欠落シナリオが見つかる場合があります。これらの欠落回復シナリオを同時に見つけるように QuickTest を指定できます。また，欠落シナリオを個別に処理することもできます。

欠落回復シナリオを見つけるには，次の手順を実行します。

1 ［欠落リソース］表示枠で，検出する回復シナリオを右クリックしてショートカット・メニューから［**検索**］を選択するか，検出する回復シナリオをダブルクリックします。



［回復シナリオの特定］ダイアログ・ボックスが開きます。

2 ［参照］ボタンをクリックして回復ファイルを選択します。［**シナリオ**］領域に，選択した回復ファイルに含まれているすべてのシナリオが表示されます。

3 欠落している回復シナリオを回復シナリオのリストから選択します。［**OK**］をクリックします。選択した回復シナリオがテストに関連付けられ，欠落している回復シナリオが［欠落リソース］表示枠から削除されます。

注：同じ回復ファイルの中で見つかる欠落回復シナリオがほかにもテストに含まれている場合，メッセージ・ボックスが開き，それらの回復シナリオも割り当てるかどうかどうか確認を求められます。欠落している回復シナリオをすべて割り当てるには，［**はい**］をクリックします。指定したシナリオのみを割り当てるには，［**いいえ**］をクリックします。

欠落している回復シナリオの削除

不要になった欠落回復シナリオは，テストから削除できます。

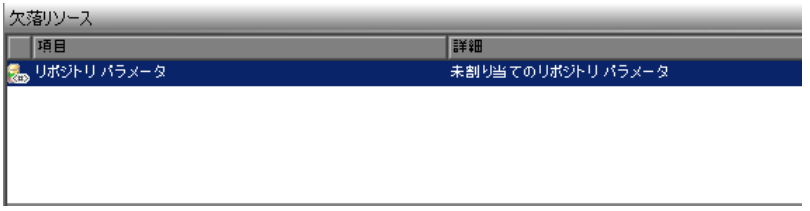
欠落回復シナリオを削除するには，削除する回復シナリオを［欠落リソース］表示枠の中で右クリックし，ショートカット・メニューから［**削除**］を選択し

ます。確認ダイアログが表示されます。[**OK**] をクリックして，回復シナリオを削除します。欠落回復シナリオが，テストと［欠落リソース］表示枠から削除されます。

未割り当ての共有オブジェクト・リポジトリ・パラメータ値の処理

コンポーネント内で使用するすべてのリポジトリ・パラメータに，値が指定されている必要があります。この値は，パラメータの作成時に指定された標準設定値か，またはコンポーネント内で指定した値です （リポジトリ・パラメータの詳細については，245 ページ「リポジトリ・パラメータを使用した作業」を参照してください)。

値がないリポジトリ・パラメータを持つオブジェクト・リポジトリを使用するコンポーネントを開くと，［欠落リソース］表示枠に［**リポジトリ パラメータ**］が表示され，この状況が示されます。



たとえば，前の画面の選択内容に応じて名前のプロパティが変化するエディット・ボックスがアプリケーション内にあるとします。リポジトリ・パラメータを使ってオブジェクト・リポジトリ内の名前プロパティの値をパラメータ化したにもかかわらず，そのリポジトリ・パラメータの標準設定値が定義されていない場合は，パラメータの値を定義する必要があります。この値をローカル・パラメータまたはコンポーネント・パラメータに割り当てることができます。また，定数値を定義することなどもできます。

［**リポジトリ パラメータ**］が表示されている行を右クリックして［**解決**］を選択するか，または［**リポジトリ パラメータ**］が表示されている行をダブルクリックすると，［オブジェクト リポジトリ パラメータの割り当て］ダイアログ・ボックスが開き，割り当てられていないオブジェクト・リポジトリ・パラメータの値を指定できます。このダイアログ・ボックスをフィルタ処理して，未割り当てのパラメータのみを表示したり，指定されたコンポーネント内の（値が割り当て済みまたは未割り当ての）パラメータをすべて表示したりできます。詳細については，144 ページ「リポジトリ・パラメータ値の割り当て」を参照してください。

第 36 章

プロセス・ガイダンスを使った作業

プロセス・ガイダンスは，特定のプロセスを最も効果的に実行する方法に関する手順と詳細を提供するツールです。プロセス・ガイダンスを使用すると，新しいプロセスについて学習したり，すでに知っているプロセスを実行するための望ましい方法について学習したりできます。このため，プロセス・ガイダンスは新しいユーザおよび経験を積んだユーザの両方にとって役に立ちます。

プロセスとは，アクティビティ，つまりサブプロセスの集合のことです。各プロセスにおいて，そのプロセスで必要なアクティビティを段階的に案内します。各プロセスのアクティビティを順にたどり，各アクティビティで説明しているタスクを実行することで，特定のプロセスの実行方法に次第に慣れてきます。

QuickTest は，複数のプロセスで構成される組み込みのパッケージを備えています。これらのプロセスでは，アプリケーション領域またはビジネス・コンポーネントの計画と作成など，最も一般的な QuickTest タスクを実行する方法に関する基本的な情報およびヒントが提示されます。

また，組織独自のカスタム・プロセスを作成し，その組織に関係のある特定の要件およびベスト・プラクティスについてユーザを案内することもできます。詳細については，997 ページ「カスタムのプロセス・ガイダンス・パッケージの作成」を参照してください。

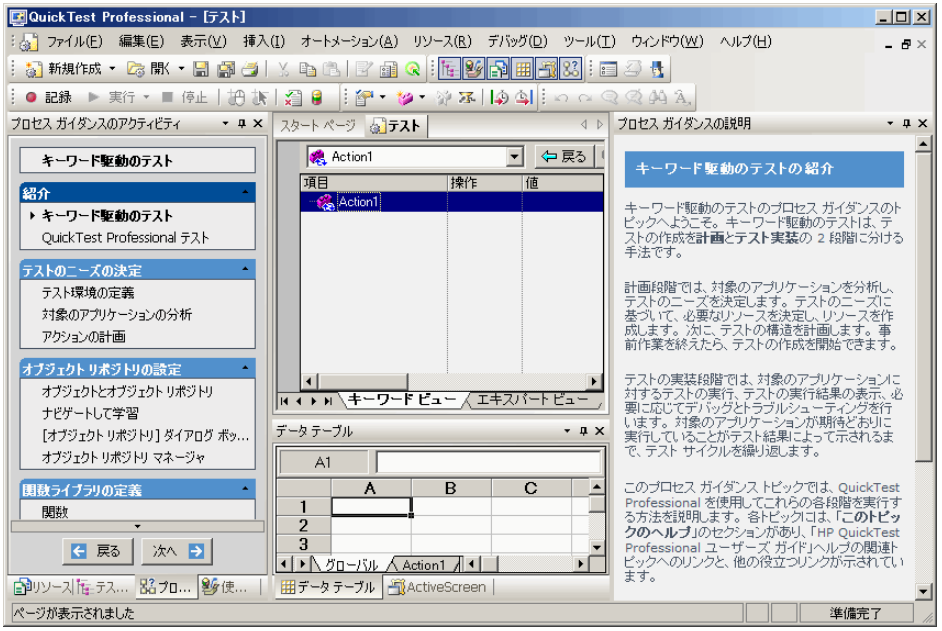
本章の内容

- ［プロセス ガイダンス］表示枠（858 ページ）
- プロセス・ガイダンスの開始（860 ページ）
- 利用できるプロセスのリストの管理（862 ページ）
- ［プロセス ガイダンス管理］ダイアログ・ボックス（862 ページ）

[プロセス ガイダンス] 表示枠

QuickTest では，プロセス・ガイダンスは，[**プロセス・ガイダンスのアクティビティ**] 表示枠と [**プロセス・ガイダンスの説明**] 表示枠という 2 つの表示枠に表示されます。

これらの表示枠の表示と非表示を切り替えるには，[**表示**] > [**プロセス ガイダンス**] を選択するか，または [**プロセス ガイダンス**] 表示枠トグル・ボタンをクリックします。



[プロセス ガイダンスのアクティビティ] 表示枠

[**プロセス ガイダンスのアクティビティ**] 表示枠（左側に表示）には，選択したプロセスの一部であるアクティビティが表示されます。アクティビティは多くの場合グループ化されており，関心のあるサブプロセスに直接移動できます。前出の例では，**キーワード駆動のテスト**・プロセスにおけるグループとアクティビティの一部を示しています。たとえば，「Determine Testing Needs」グループには，「Define Testing Environment」，「Analyze Your Application」，および「Plan Your Actions」という 3 つのアクティビティが含まれています。

［**プロセス ガイダンスのアクティビティ**］表示枠では，次の操作を実行できます。

- アクティビティをクリックして，そのアクティビティに対応するトピックを［**プロセス ガイダンスの説明**］表示枠で開く。
- どのアクティビティが［プロセス ガイダンスの詳細］表示枠に表示されているかを確認する（現在選択されているアクティビティが矢印で示される）。
- ［**戻る**］ボタンと［**次へ**］ボタンを使用してアクティビティ間を上下に移動し，［**プロセス ガイダンスの説明**］表示枠に前または次のアクティビティのトピックを表示する。
- **上矢印**または**下矢印**上にカーソルを置いてアクティビティのリストをスクロールする（上矢印は，［プロセス ガイダンス アクティビティ］タイトル・バーのすぐ下に，下矢印は，［**戻る**］ボタンおよび［**次へ**］ボタンのすぐ上にある）。

［プロセス ガイダンスの説明］表示枠

［**プロセス ガイダンスの説明**］表示枠（前出の例では右側に表示）には，選択したアクティビティに関するトピック説明が表示されます。

詳細ごとに特定のアクティビティが紹介され，そのアクティビティの実行方法に関する情報を見つけられる場所へのリンクが提示されます。さらに，多くの詳細には，ダイアログ・ボックスやほかの関連する機能を開く対話型のリンクが含まれており，説明されている機能に直接アクセスできます。

プロセス・ガイダンスの開始

プロセスは，［スタート ページ］，［**オートメーション**］メニュー，または［プロセス ガイダンス アクティビティ］表示枠から開けます。

スタート・ページ

スタート・ページの［**プロセス ガイダンス リスト**］には，利用可能なプロセスがすべて表示されます。一部のプロセスは，特定の条件のもとでのみ利用できる場合があります。たとえば，ビジネス・コンポーネント・プロセスのガイダンスを利用できるのは，ビジネス・プロセス・テストをサポートする Quality Center プロジェクトに接続されている場合にかぎられます。さらに，一部のプロセスは，特定のアドインが読み込まれている場合にのみ表示されます。たとえば，**Testing SAP GUI for Windows** 組み込みプロセスが表示されるのは，SAP Add-in が読み込まれている場合にかぎられます。

リストから QuickTest プロセスを選択すると，関連する種類のドキュメントが開きます。たとえば，テスト・ドキュメントを開いた状態で，［**アプリケーション領域**］プロセスを選択した場合，新しいアプリケーション領域が開き，選択したプロセスをたどりながらアプリケーション領域を操作できます（ビジネス・プロセス・テストがサポートされている Quality Center プロジェクトに接続されている場合）。

スタート・ページから特定のプロセスを開くには，次の手順を実行します。

1 QuickTest で，［**スタート ページ**］タブをクリックしてスタート・ページを表示します（［**スタート ページ**］タブが表示されていない場合は，［**表示**］＞［**スタート ページ**］を選択してスタート・ページを開きます）。

2 ［**プロセス ガイダンス リスト**］で，開くプロセスのリンクをクリックします。アクティビティのリストが［**プロセス ガイダンス アクティビティ**］表示枠に表示され，リストの最初のアクティビティの詳細が［**プロセス ガイダンスの詳細**］表示枠に表示されます。

ヒント :［**プロセス ガイダンス リスト**］が空の場合は，［**ファイル**］＞［**プロセス ガイダンス管理**］を選択し，［プロセス ガイダンス管理］ダイアログ・ボックスで少なくとも 1 つのプロセスを選択します。詳細については，862 ページ「[プロセス ガイダンス管理］ダイアログ・ボックス」を参照してください。

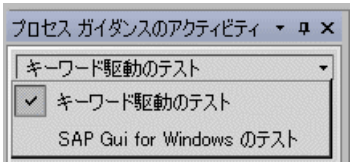
［オートメーション］メニュー・コマンド

現在開いているドキュメントの種類，または読み込まれている QuickTest アドインで利用できるプロセスを開くには，［**オートメーション**］＞［**プロセス ガイダンス リスト**］を選択し，リストからプロセスを選択します。

［**プロセス ガイダンス リスト**］が空の場合は，［**ファイル**］＞［**プロセス ガイダンス管理**］を選択し，［プロセス ガイダンス管理］ダイアログ・ボックスで少なくとも 1 つのプロセスを選択します。次に，現在のドキュメントを再び開くか，または新しいドキュメントを開いて［**プロセス ガイダンス リスト**］メニューで利用できるプロセスのリストを更新します

現在のテスト・ドキュメントまたは読み込まれている QuickTest アドインに関連のないプロセスを開くには，スタート・ページの［**プロセス ガイダンス リスト**］からプロセスを開く必要があります。

現在開いているテスト・ドキュメントに利用できるプロセスが複数ある場合は，プロセスのタイトルのドロップ・ダウン・リストで必要なプロセスを選択することによって，これらのプロセスを行き来できます。



利用できるプロセスのリストの管理

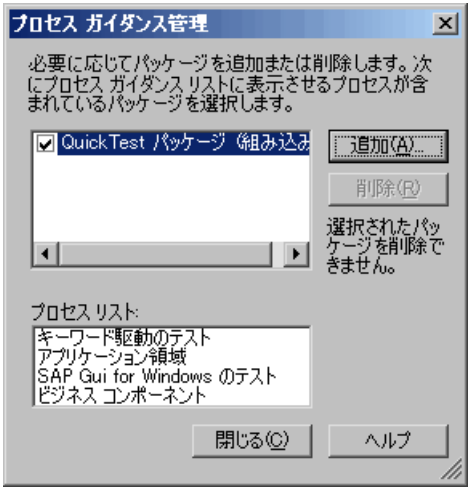
プロセスはプロセス・ガイダンス・パッケージに格納されています。QuickTest は，複数のプロセスを含んだ組み込みパッケージを備えています。このパッケージは，［プロセス ガイダンス管理］ダイアログ・ボックスに標準で表示されます。

各組織は，組織，チーム，組織での役割などに固有のプロセスが含まれている追加のパッケージを提供できます。詳細については，997 ページ「カスタムのプロセス・ガイダンス・パッケージの作成」を参照してください。

［プロセス ガイダンス管理］ダイアログ・ボックス

説明	QuickTest で使用できるプロセスのリストを管理できます。
アクセス方法	［**ファイル**］メニュー＞［**プロセス ガイダンス管理**］
重要情報	➤ QuickTest の組み込みパッケージは削除できません。 ➤ パッケージ内の個々のプロセスを追加したり除外したりすることはできません。
詳細情報	**概念的概要**：857 ページ「プロセス・ガイダンスを使った作業」 **主要なタスク**： ➤ 863 ページ「パッケージの利用と除外」 ➤ 864 ページ「プロセス・ガイダンス・パッケージの追加」 **その他の関連トピック**：863 ページ「補足リファレンス」を参照してください。

［プロセス ガイダンス管理］ダイアログ・ボックスの画像を次に示します。



［プロセス ガイダンス管理］ダイアログ・ボックスのオプション

オプション	説明
［追加］	［スタート ページ］の［プロセス ガイダンス リスト］に組織固有のプロセスを追加できます。
［削除］	［スタート ページ］の［プロセス ガイダンス リスト］からプロセスを削除できます。

補足リファレンス

関連ユーザ・インタフェース・トピック	858 ページ「［プロセス ガイダンス］表示枠」
関連作業	860 ページ「プロセス・ガイダンスの開始」

パッケージの利用と除外

QuickTest で利用できるパッケージ群に含まれているパッケージを利用するか除外するかを選択できます。

パッケージを含めることを選択すると，QuickTest によって，そのパッケージ内のすべてのプロセスが，スタート・ページの［**プロセス ガイダンス リスト**］に追加されます（現在読み込まれていない QuickTest アドインのプロセスを除きます）。現在開いているドキュメントの種類および現在読み込まれている QuickTest アドインで利用できるプロセスは，［**オートメーション**］メニューの［**プロセス ガイダンス リスト**］にも追加され，現在のドキュメントを閉じてから再び開くか同じ種類の新しいドキュメントを開くことによってリストを更新した後に開くことができます。

パッケージ内の個々のプロセスを追加したり除外したりすることはできません。

QuickTest で利用できるパッケージ群を対象にパッケージを追加したり除外したりするには，次の手順を実行します。

1 ［**ファイル**］＞［**プロセス ガイダンス管理**］を選択します。［プロセス ガイダンス管理］ダイアログ・ボックスが開きます。

2 含めたいプロセスのパッケージの横にあるチェック・ボックスを選択するか，除外したいプロセスのパッケージの横にあるチェック・ボックスをクリアします。

3 ［**閉じる**］をクリックします。QuickTest によって，該当するプロセスが［**プロセス ガイダンス リスト**］に追加または削除されます。

プロセス・ガイダンス・パッケージの追加

組織に独自のプロセスがある場合は，それらをスタート・ページの［**プロセス ガイダンス リスト**］に追加できます。これを行うには，該当するパッケージを［プロセス ガイダンス管理］ダイアログ・ボックスに追加し，選択することで表示します。

リストにパッケージを追加するには，次の手順を実行します。

1 ［**ファイル**］＞［**プロセス ガイダンス管理**］を選択します。［プロセス ガイダンス管理］ダイアログ・ボックスが開きます。

2 ［プロセス ガイダンス管理］ダイアログ・ボックスで，［**追加**］をクリックします。［ファイルを開く］ダイアログ・ボックスが開きます。

3 プロセス・ガイダンス・パッケージ・ファイルを参照して［**開く**］をクリックします。利用可能パッケージのリストにパッケージが追加されます。

第 X 部

高度な機能を使用した作業

第 37 章

回復シナリオの定義と使用

実行セッション中，テスト環境で生じる予期しないイベントやエラーから回復するよう QuickTest に指示できます。

本章の内容

- 回復シナリオの定義と使用について（867 ページ）
- 回復シナリオを使用するタイミングの決定（869 ページ）
- 回復シナリオの定義（870 ページ）
- 回復シナリオ・ウィザードについて（874 ページ）
- 回復シナリオの管理（901 ページ）
- アプリケーション領域への回復シナリオの関連付け（905 ページ）
- プログラムによる回復メカニズムの制御（909 ページ）

回復シナリオの定義と使用について

実行セッション中に，予期しないイベント，エラー，およびアプリケーション・クラッシュが発生すると，実行セッションが妨げられ，正しいテスト結果が得られない可能性があります。これは，コンポーネントを無人で実行する場合に特に問題になります。回復に必要な操作を実行するまで，コンポーネントが一時停止状態になるからです。このような状況を処理するために，QuickTest では回復シナリオを作成して特定のコンポーネントに関連付けることができます。トリガ・イベントが発生すると，［回復］シナリオによって特定の回復操作が開始されます。どのようなときに回復シナリオを使用するかについては，869 ページ「回復シナリオを使用するタイミングの決定」を参照してください。

回復シナリオ・マネージャでは，「回復シナリオ」を定義するプロセスを案内するウィザードが使用できます。回復シナリオには，予期しないイベントの定義と，実行セッションを回復するために必要な操作が含まれています。たとえば，「**Printer out of paper**」というメッセージを検出し，［**OK**］ボタンをクリックしてメッセージを閉じることによって実行セッションを回復し，コンポーネントを続行するよう QuickTest に指示できます。

回復シナリオは，次の要素で構成されています。

- **トリガ・イベント**：実行セッションを中断するイベントです。たとえば，画面上にポップアップ表示されるウィンドウや，QuickTest の実行エラーなどです。
- **回復操作**：トリガ・イベントによって実行セッションを中断された後も QuickTest でコンポーネントを実行し続けることができるようにする操作です。たとえば，ポップアップ・ウィンドウの［**OK**］ボタンをクリックすることや，Microsoft Windows の再起動などです。
- **回復後のテスト実行のオプション**：回復操作を実行してからの QuickTest の継続方法の指示，および回復操作を実行した場合はコンポーネントのどのポイントから QuickTest を継続するかの指示です。たとえば，コンポーネントを初めからやり直したり，完全に 1 つのステップをスキップしたり，コンポーネントの次のステップから続けたりすることができます。

回復シナリオは回復シナリオ・ファイルに保存されます。回復シナリオ・ファイルは回復シナリオの論理的な集合で，特定の独自の要件に従ってグループ化されています。

実行セッション中に回復シナリオを実行するよう QuickTest に指示するには，まずそのコンポーネントに（アプリケーション領域を通じて）回復シナリオを関連付ける必要があります。コンポーネントに関連付けることができる回復シナリオの数に制限はありません。コンポーネントに関連付けるシナリオに優先順位を付けて，必要な順序でトリガ・イベントを認識，処理させることができます。詳細については，456 ページ「アプリケーション領域の回復シナリオ設定の定義」を参照してください。

回復シナリオを定義した対象のコンポーネントを実行して，エラーが発生した場合は，QuickTest によって，エラーの原因である定義済みのトリガ・イベントが検索されます。トリガ・イベントが発生した場合，QuickTest によって，対応する回復操作および回復後操作が実行されます。

コンポーネントに Recovery ステートメントを挿入することで，実行セッション中に回復シナリオを制御し，呼び出すこともできます。詳細については，909 ページ「プログラムによる回復メカニズムの制御」を参照してください。

注 :［アプリケーション領域の設定］ダイアログ・ボックスの［回復］表示枠にある［**回復シナリオのアクティブ化**］ボックスで「**エラー発生時**」を選択した場合，回復メカニズムでは，コンポーネントの最後のステップで発生するトリガは処理されません。このオプションを選択し，かつコンポーネントの最後のステップで発生する可能性のある予期しないイベントやエラーから回復する必要がある場合は，コンポーネントの最後にさらにステップを追加することで，予期しないイベントやエラーから回復できます。

回復シナリオを使用するタイミングの決定

回復シナリオは，前もって予測できないイベントでの使用，またはコンポーネントの特定のステップと同期させることのできないイベント**のみ**を対象としています。たとえば，プリンタ・エラーを処理するために回復シナリオを定義できます。そうすれば，実行セッション中にプリンタ・エラーが発生したときに，回復シナリオが QuickTest に対して［プリンタ エラー］メッセージ・ボックスの標準のボタンをクリックするよう指示できます。

この例で回復シナリオを使用するのは，コンポーネントでは，このようなエラーを直接処理できないからです。なぜなら，どの時点でネットワークによってプリンタ・エラーが返されるか知ることができないためです。ユーザ定義関数においてファイルをプリンタに送信するステップの直後に If ステートメントを追加することによってこのイベントを処理しようとしても，ネットワークによって実際のプリンタ・エラーが返される前に，コンポーネントは数ステップ進む場合があります。

イベントがコンポーネントの特定のポイントで発生することが予測できる場合は，回復シナリオを使用するのではなく，If ステートメントなどのステップをユーザ定義関数に追加追加して，コンポーネントから直接イベントを処理することをお勧めします。たとえば，実行セッション中に［**保存**］ボタンがクリックされるとファイルの上書きメッセージ・ボックスが表示されることがあるという場合は，このイベントを，メッセージ・ボックスが開くと［**OK**］をクリックする If ステートメントをユーザ定義関数に追加することで処理できます。

コンポーネントから直接イベントを処理すると，回復シナリオよりもより明確にエラー処理が行えます。これは回復シナリオが一般的な予測しないイベントのセットを処理するよう設計されているためです。またテストから直接イベントを処理すれば，調整処理のタイミングを制御でき，最小の労力で最大の効果を挙げることができます。標準では，回復シナリオ操作が開始されるのは，ステップによってエラーが返された後にかぎられます。これは，最初にエラーを引き起こしたステップの数ステップ後に発生する可能性があります。別の方法としては，ステップごとにトリガ・イベントを確認することも考えられますが，パフォーマンスが低下する可能性があります。このため，予測可能なエラーはコンポーネントで直接処理することをお勧めします。

回復シナリオの定義

回復シナリオは，回復シナリオ・ウィザード（[回復シナリオ マネージャ］ダイアログ・ボックスからアクセスします）を使用して作成します。回復シナリオ・ウィザードでは，回復シナリオの各段階を定義するプロセスを案内します。作成した回復シナリオは，回復ファイルに保存します。回復ファイルは，複数の回復シナリオをまとめて整理，格納するのに便利です。

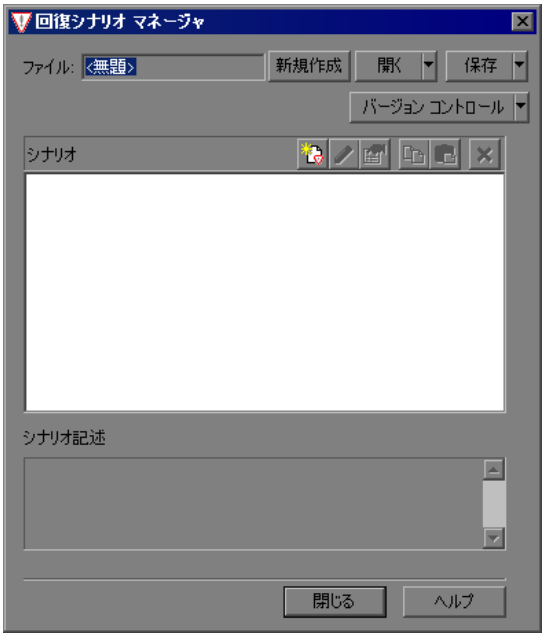
［回復シナリオ マネージャ］ダイアログ・ボックスを使用すると，任意の回復ファイルを選択してそのファイルに格納されているすべての回復シナリオを管理できます。これにより，選択した回復シナリオを編集したり，特定の回復シナリオを特定のコンポーネントに関連付けて，指定したトリガ・イベントが発生したときに回復シナリオを使用するように QuickTest を設定したりできます。

回復ファイルの作成

回復シナリオを格納するには回復ファイルを作成します。新規の回復ファイルを作成することも，既存のファイルを編集することもできます。

回復ファイルを作成するには，次の手順を実行します。

1 ［**リソース**］＞［**回復シナリオ マネージャ**］を選択します。［回復シナリオ マネージャ］ダイアログ・ボックスが開きます。



2 標準設定では，［回復シナリオ マネージャ］ダイアログ・ボックスには新規の回復ファイルが表示されます。この新規ファイルを使用できます。また，次のいずれかの方法で，既存の回復ファイルを開くこともできます。

- ［**開く**］ボタンの横にある矢印をクリックして，最近使用した回復ファイルをリストから選択します。
- 次の手順を実行します。

a ［**開く**］ボタンをクリックし，既存の回復ファイルを選択します。

b ［回復シナリオを開く］ダイアログ・ボックスのサイドバーで，ファイルが格納されている場所（たとえば，**ファイル・システム**や Quality Center テス

ト・リソース**など）を選択します。

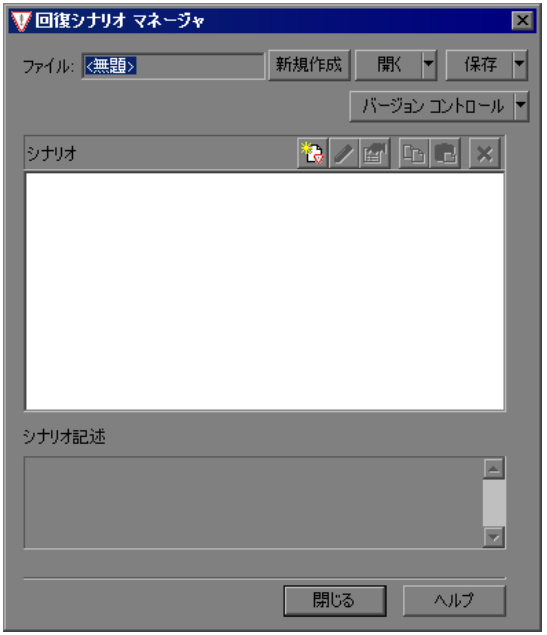
c 開く回復シナリオ・ファイルを参照して選択し，［**開く**］をクリックします。Quality Center 内のバージョン・コントロールが有効なプロジェクトに回復ファイルが格納されている場合は，［**開く**］の下矢印をクリックし，［**開いてチェックアウトする**］を選択して，ファイルをチェックアウトできます。

次の項で説明するように，回復シナリオ・ウィザードを使用して回復シナリオを作成し，回復ファイルに保存することができます。

［回復シナリオ マネージャ］ダイアログ・ボックスについて

［回復シナリオ マネージャ］ダイアログ・ボックスでは，回復ファイルの作成と編集，およびこれらのファイルに保存されている回復シナリオの作成と管理ができます。

［回復シナリオ マネージャ］ダイアログ・ボックスには，現在開いている回復ファイルの名前，回復ファイルに保存したシナリオのリスト，および各シナリオの説明が表示されます。



［回復シナリオ マネージャ］ダイアログ・ボックスには，次のツールバー・ボタンがあります。

オプション	説明
	新規の回復ファイルを作成します。詳細については，871 ページ「回復ファイルの作成」を参照してください。
	既存の回復ファイルを開きます。矢印をクリックして，最近使用した回復ファイルのリストから回復ファイルを選択することもできます。
	現在の回復ファイルを保存します。詳細については，899 ページ「回復ファイルでの回復シナリオの保存」を参照してください。
	回復シナリオのバージョン・コントロールを管理できます （バージョン・コントロールが有効になっている Quality Center プロジェクトに QuickTest が接続されている場合にのみ，オプションを使用できます)。 詳細については，968 ページ「Quality Center でのアセットのバージョン管理」および 949 ページ「QuickTest アセットのバージョンの表示と比較」を参照してください。
	回復シナリオ・ウィザードを開き，新規の回復シナリオを定義します。詳細については，874 ページ「回復シナリオ・ウィザードについて」を参照してください。
	選択した回復シナリオの回復シナリオ・ウィザードを開き，回復シナリオの設定を変更できます。詳細については，903 ページ「回復シナリオの変更」を参照してください。
	選択した回復シナリオのプロパティのサマリを表示専用の形式で表示します。詳細については，902 ページ「回復シナリオのプロパティの表示」を参照してください。
	回復シナリオを，開いている回復ファイルからクリップボードにコピーします。これによって，回復シナリオを別の回復ファイルに貼り付けることができます。詳細については，904 ページ「回復シナリオ・ファイル間での回復シナリオのコピー」を参照してください。
	回復シナリオを，クリップボードから開いている回復ファイルに貼り付けます。詳細については，904 ページ「回復シナリオ・ファイル間での回復シナリオのコピー」を参照してください。

オプション	説明
	回復シナリオを削除します。詳細については，903 ページ「回復シナリオの削除」を参照してください。

注：各回復シナリオには，その種類ごとに異なるアイコンが関連付けられています。詳細については，901 ページ「回復シナリオの管理」を参照してください。

回復シナリオ・ウィザードについて

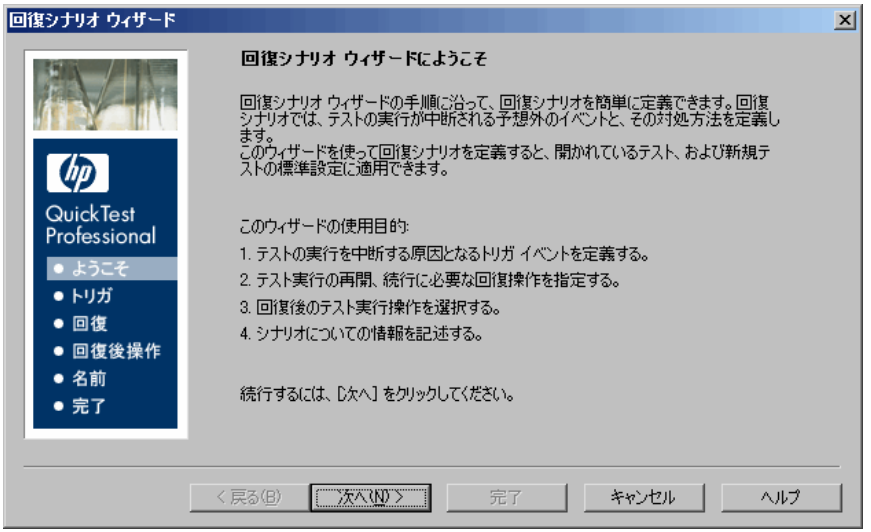
回復シナリオ・ウィザードでは，回復シナリオを作成するプロセスを段階的に案内します。回復シナリオ・ウィザードには，次の主要なステップが含まれています。

- 実行セッションを中断するトリガ・イベントの定義
- 継続に必要な回復操作の指定
- 回復後のテスト実行の操作の選択
- 回復シナリオの名前と説明の指定

回復シナリオ・ウィザードを開くには，[回復シナリオ マネージャ] ダイアログ・ボックス（[**リソース**] > [**回復シナリオ マネージャ**]）の [**新規シナリオ**] ボタンをクリックします。

[回復シナリオ ウィザードにようこそ] 画面

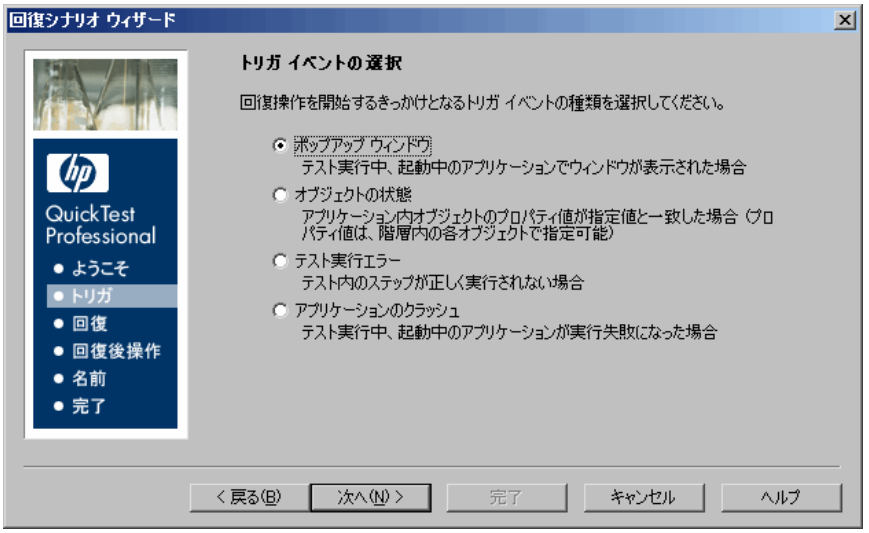
[回復シナリオ ウィザードにようこそ] 画面には，回復シナリオ・ウィザードのさまざまなオプションについての一般的な情報と，回復シナリオの定義に関連する段階の概要が表示されます。



[**次へ**] をクリックして [トリガ イベントの選択] 画面に進みます（876 ページを参照してください）。

［トリガ イベントの選択］画面

［トリガ イベントの選択］画面では，回復シナリオをトリガするイベントのタイプと，QuickTest によるイベントの認識方法を定義できます。



トリガのタイプを選択し，［**次へ**］をクリックします。ウィザードに表示される次の画面は，以下のどのトリガのタイプを選択したかに応じて異なります。

- ［**ポップアップ ウィンドウ**］：QuickTest によって，ポップアップ・ウィンドウが検出され，ウィンドウのタイトルとテキストの内容に従ってそのウィンドウが識別されます。たとえば，実行セッション中にプリンタの用紙切れを示すメッセージ・ボックスが表示される場合があります。QuickTest では，実行セッションを続行するために，このウィンドウを検出し，定義済みの回復シナリオを呼び出すことができます。

  このオプションを選択し，［**次へ**］をクリックして，［ポップアップ ウィンドウの条件を指定］画面に進みます（878 ページを参照してください）。

- ［**オブジェクトの状態**］：QuickTest によって，特定のテストのオブジェクト状態が検出され，そのプロパティ値とすべての祖先のプロパティ値に従ってオブジェクト状態が識別されます。オブジェクトはクラスではなく，プロパティ値によってのみ識別されます。

たとえば，特定のプロセスが開いている場合にダイアログ・ボックスの特定のボタンが無効になることがあります。QuickTest では，この問題のプロセスを開いているときに発生するボタンのオブジェクト・プロパティ状態を検出し，定義済みの回復シナリオを呼び出して，そのプロセスを閉じて実行セッションを続行できます。

このオプションを選択し，［**次へ**］をクリックして，［オブジェクトの選択］画面に進みます（880 ページを参照してください）。

➤ ［**テスト実行エラー**］：QuickTest では，テストの実行エラーが検出され，メソッドからの失敗した戻り値によってエラーが識別されます。たとえば，QuickTest では，メニュー項目は実行セッション中の特定の場所では利用できないため，メソッド引数で指定されたメニュー項目を識別できない場合があります。QuickTest では，実行セッションを続行するために，この実行エラーを検出し，定義済みの回復シナリオを呼び出すことができます。

このオプションを選択し，［**次へ**］をクリックして，［テスト実行エラー］画面に進みます（884 ページを参照してください）。

➤ ［**アプリケーションのクラッシュ**］：QuickTest によって，アプリケーション・クラッシュが検出され，定義済みのアプリケーションのリストに従って識別されます。たとえば，実行セッション中にステップを実行したときに，2 次的なアプリケーションがクラッシュすることが考えられます。使用中のアプリケーションの問題ではない可能性のあるこのクラッシュによって実行セッションが失敗とならないように，QuickTest では，このアプリケーションのクラッシュを検出し，定義済みの回復シナリオを呼び出して実行セッションを続行することができます。

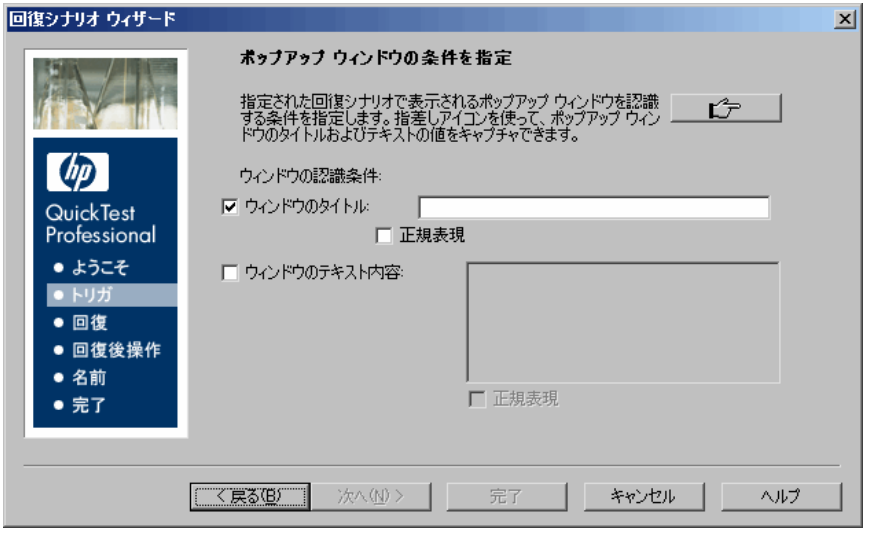
このオプションを選択し，［**次へ**］をクリックして，［回復操作］画面に進みます（887 ページを参照してください）。

注：

- 一連の回復操作は，トリガ・イベントの条件に合致する状態が発生するたびに実行されます。たとえば，特定のオブジェクト状態を定義し，2 つのオブジェクトがこの状態と一致した場合，指定した状態に各オブジェクトが一致するたびに 1 回ずつ，合計 2 回の回復操作が実行されます。
- 回復メカニズムでは，コンポーネントの最後のステップで発生するトリガは処理されません。コンポーネントの最後のステップで発生する可能性のある予期しないイベントやエラーから回復する必要がある場合は，コンポーネントの最後にさらにステップを追加することで，予期しないイベントやエラーから回復できます。

［ポップアップ ウィンドウの条件を指定］画面

［トリガ イベントの選択］画面にある［**ポップアップ ウィンドウ**］トリガを選択した場合（876 ページを参照してください），［ポップアップ ウィンドウの条件を指定］画面が開きます。



次のいずれかを実行してポップアップ・ウィンドウの識別方法を指定します。

- ポップアップ・ウィンドウの［**ウィンドウのタイトル**］または［**ウィンドウのテキスト内容**］，あるいはその両方に基づいてポップアップ・ウィンドウを識別するかどうかを選択し，ポップアップ・ウィンドウの識別に使用するテキストを入力します。ウィンドウ・タイトルまたはテキスト内容に対して正規表現を使用するには，該当する［**正規表現**］チェック・ボックスを選択し，該当する場所に正規表現を入力します。正規表現の詳細については，『**HP QuickTest Professional ユーザーズ・ガイド**』を参照してください。
- 指差しマークをクリックします。次に，ポップアップ・ウィンドウをクリックし，ウィンドウ・タイトルとウィンドウのテキスト内容をキャプチャします。指差しマークの使用方法の詳細については，879 ページ「指差しマークの使い方のヒント」を参照してください。

注：前者（［**ウィンドウのタイトル**］または［**ウィンドウのテキスト内容**］，あるいはその両方）を実行した場合，QuickTest は指定したタイトルまたはテキスト，あるいはその両方を含んだ任意のポップアップ・ウィンドウを識別します。後者（指差しマーク）を実行した場合，QuickTest は選択したウィンドウのオブジェクト・プロパティ値に一致するポップアップ・ウィンドウのみを識別します。

［**次へ**］をクリックして［回復操作］画面に進みます（887 ページを参照してください）。

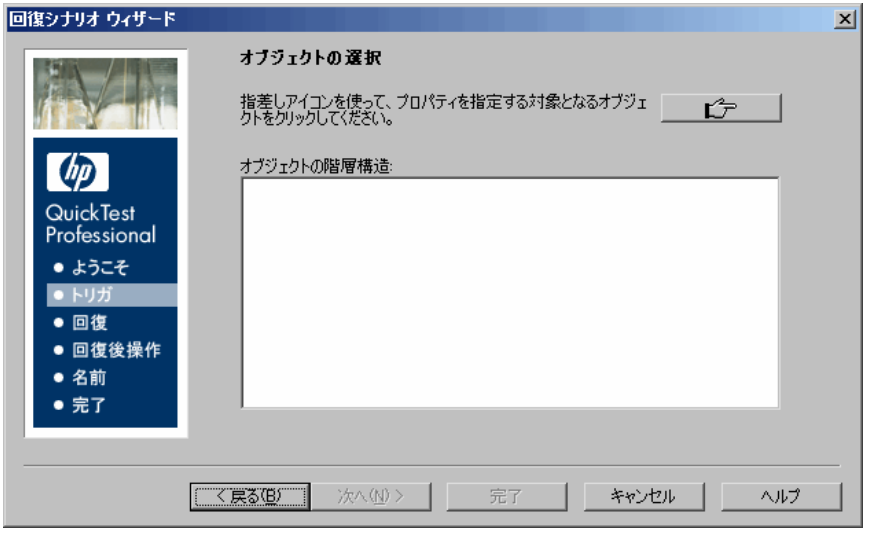
指差しマークの使い方のヒント

- 左側の CTRL キーを押し下げると，指差しマークが標準のポインタに変わります。これによって，ウィンドウのフォーカスを変更したり，ショートカット・メニューを表示するために右クリックする，スクロール・バーを使用する，オブジェクト上にポインタを移動するなどの操作を QuickTest またはアプリケーションで実行したりできるようになります。
- 選択するオブジェクトを含んでいるウィンドウが部分的に別のウィンドウに隠れている場合は，ウィンドウが前面に表示されるまで，部分的に隠れているウィンドウに指差しマークを数秒間置いたままにします。それから，必要なオブジェクトをポイントしてクリックします。ウィンドウを前面に出すのに必要な時間の設定は，［オプション］ダイアログ・ボックスの［一般］表示枠で行えます。

- 選択対象のオブジェクトを含んでいるウィンドウが別のウィンドウの背後に完全に隠れている場合，または，ダイアログ・ボックスがウィンドウの背後に隠れている場合は，必要に応じて左側の CTRL キーを押しながらウィンドウを配置します。
- 選択対象オブジェクトを含んでいるウィンドウが最小化されている場合は，左の CTRL キーを押したまま，Windows タスク・バー内のアプリケーションを右クリックして，ショートカット・メニューから［**元のサイズに戻す**］を選択することで，ウィンドウを表示できます。
- 特定のイベント（ショートカット・メニューを表示するために右クリックしたり，オブジェクトにポインタを合わせるなど）が生じないと選択対象オブジェクトが表示されない場合は，左 CTRL キーを押し下げます。すると，指差しマークが一時的に標準のポインタに変わり，イベントを実行できるようになります。選択対象のオブジェクトが表示されたら，左側の CTRL キーを放します。ポインタが再び指差しマークに変わります。

［オブジェクトの選択］画面

［トリガ イベントの選択］画面にある［**オブジェクトの状態**］トリガを選択すると（876 ページを参照してください），［オブジェクトの選択］画面が開きます。



指差しマークをクリックしてから，プロパティを指定するオブジェクトをクリックします。指差しマークの使用方法の詳細については，881 ページ「指差しマークの使い方のヒント」を参照してください。

クリックした場所が複数のオブジェクトに関連付けられている場合は，［オブジェクトの選択 - オブジェクトの状態トリガ］ダイアログ・ボックスが開きます。

指定するプロパティを含んでいるオブジェクトを選択し，［**OK**］をクリックします。選択したオブジェクトとその親が［オブジェクトの選択］画面に表示されます。

注：階層オブジェクトの選択ツリーでは，Web テーブルなど，QuickTest が通常は学習しないオブジェクト（親以外のオブジェクト）を選択することもできます。

［**次へ**］をクリックして［オブジェクトのプロパティと値の設定］画面に進みます（883 ページを参照してください）。

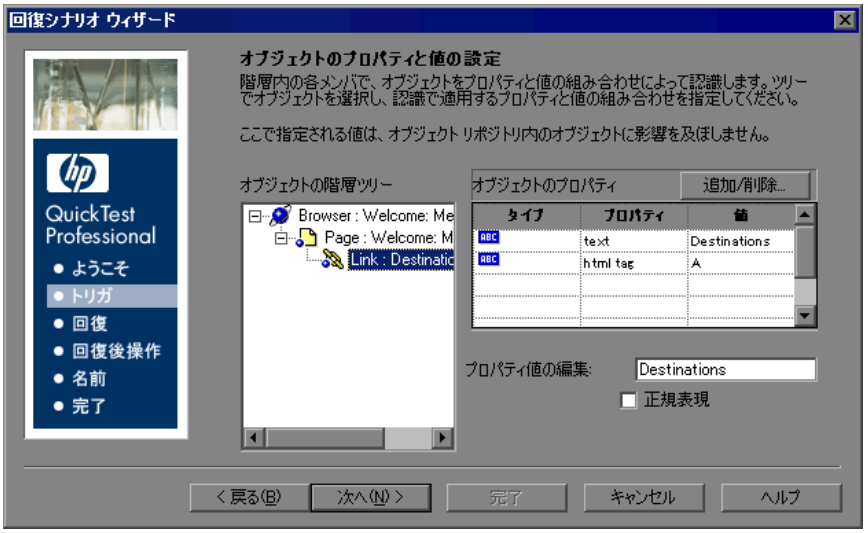
指差しマークの使い方のヒント

➤ 左側の CTRL キーを押し下げると，指差しマークが標準のポインタに変わります。これによって，ウィンドウのフォーカスを変更したり，ショートカット・メニューを表示するために右クリックする，スクロール・バーを使用する，オブジェクト上にポインタを移動するなどの操作を QuickTest またはアプリケーションで実行したりできるようになります。

- 選択するオブジェクトを含んでいるウィンドウが部分的に別のウィンドウに隠れている場合は，ウィンドウが前面に表示されるまで，部分的に隠れているウィンドウに指差しマークを数秒間置いたままにします。それから，必要なオブジェクトをポイントしてクリックします。ウィンドウを前面に出すのに必要な時間の設定は，［オプション］ダイアログ・ボックスの［一般］表示枠で行えます。
- 選択対象のオブジェクトを含んでいるウィンドウが別のウィンドウの背後に完全に隠れている場合，または，ダイアログ・ボックスがウィンドウの背後に隠れている場合は，必要に応じて左側の CTRL キーを押しながらウィンドウを配置します。
- 選択対象オブジェクトを含んでいるウィンドウが最小化されている場合は，左の CTRL キーを押したまま，Windows タスク・バー内のアプリケーションを右クリックして，ショートカット・メニューから［**元のサイズに戻す**］を選択することで，ウィンドウを表示できます。
- 特定のイベント（ショートカット・メニューを表示するために右クリックしたり，オブジェクトにポインタを合わせるなど）が生じないと選択対象オブジェクトが表示されない場合は，左 CTRL キーを押し下げます。すると，指差しマークが一時的に標準のポインタに変わり，イベントを実行できるようになります。選択対象のオブジェクトが表示されたら，左側の CTRL キーを放します。ポインタが再び指差しマークに変わります。

[オブジェクトのプロパティと値の設定] 画面

[オブジェクトの選択] 画面でプロパティを指定するオブジェクトを選択すると（880 ページを参照してください），[オブジェクトのプロパティと値の設定] 画面が開きます。



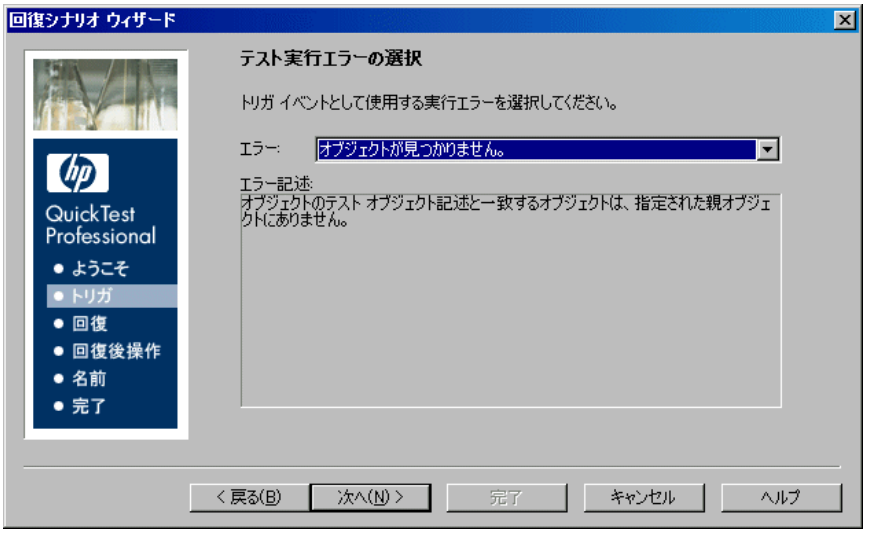
階層の各オブジェクトに対して，[**プロパティ値の編集**] ボックスで，オブジェクトの識別に使用するプロパティ値を変更できます。[**追加 / 削除**] ボタンをクリックし，検査対象のプロパティ値のリストからオブジェクト・プロパティを追加または削除することもできます。オブジェクトはクラスではなく，プロパティ値によってのみ識別されます。

プロパティ値で正規表現を使用する場合は，[**正規表現**] チェック・ボックスを選択します。正規表現の詳細については，『**HP QuickTest Professional ユーザーズ・ガイド**』を参照してください。

[**次へ**] をクリックして［回復操作］画面に進みます（887 ページを参照してください）。

［テスト実行エラー］画面

［トリガ イベントの選択］画面にある［**テスト実行エラー**］トリガを選択すると（876 ページを参照してください），［テスト実行エラーの選択］画面が開きます。



［**エラー**］リストでは，次のように，トリガ・イベントとして使用するテストの実行エラーを選択します。

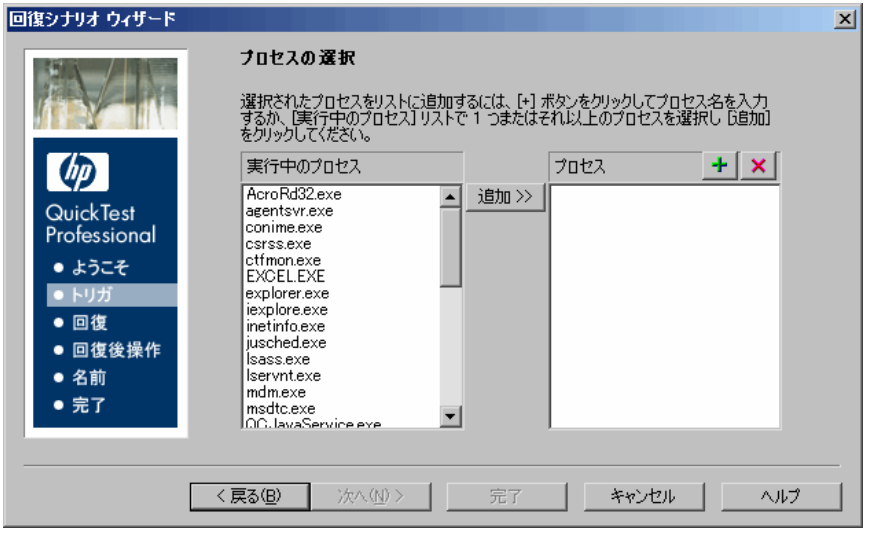
- ［**任意のエラー**］：テスト・オブジェクト・メソッドによって返されるすべてのエラー・コード。
- ［**リストまたはメニュー内の項目が一意ではありません**］：リスト，メニュー，またはツリーにある複数の項目に，メソッド引数で指定されている名前がある場合に発生します。
- ［**リストまたはメニュー内の項目が見つかりません**］：メソッド引数で指定されているリスト，メニュー，またはツリー項目が QuickTest によって識別できない場合に発生します。この原因としては，その項目を現在利用できない，または名前が変更されていることが考えられます。
- ［**物理記述に対して複数のオブジェクトが該当します**］：アプリケーションの複数のオブジェクトが，ステップで指定されているオブジェクトのテスト・オブジェクト記述で指定されている値と同一のプロパティ値を有する場合に発生します。

- ［**オブジェクトが無効になっています**］：ステップで指定されているオブジェクトが現在無効となっているため，QuickTest がステップを実行できない場合に発生します。
- ［**オブジェクトが見つかりません**］：指定された親オブジェクト内に，オブジェクトのテスト・オブジェクト記述と一致するオブジェクトがない場合に発生します。
- ［**オブジェクトが非表示になっています**］：ステップで指定されたオブジェクトが現在画面上に表示されていないため，QuickTest がステップを実行できない場合に発生します。

［**次へ**］をクリックして 887 ページ「［回復操作］画面」に進みます。

［プロセスの選択］画面

［トリガ イベントの選択］画面にある［**アプリケーションのクラッシュ**］トリガを選択すると（876 ページを参照してください），［プロセスの選択］画面が開きます。



［**実行中のプロセス**］リストには，現在実行中のすべてのアプリケーション・プロセスが表示されます。［**プロセス**］リストには，クラッシュした場合に回復シナリオをトリガするアプリケーション・プロセスが表示されます。

アプリケーション・プロセスを［**プロセス**］リストに入力するか，［**実行中のプロセス**］リストからアプリケーション・プロセスを選択することによって，［**プロセス**］リストにアプリケーション・プロセスを追加できます。

- ［**実行中のプロセス**］リストからプロセスを追加するには，［**実行中のプロセス**］リストのプロセスをダブルクリックするか，プロセスを選択して［**追加**］ボタンをクリックします。Windows で複数選択する際の標準的な方法（CTRL キーや SHIFT キー）を使用して，複数のプロセスを選択できます。

- ［**プロセス**］リストにプロセスを直接追加するには，［**新規プロセスの追加**］ボタンをクリックし，リストに追加するプロセスの名前を入力します。

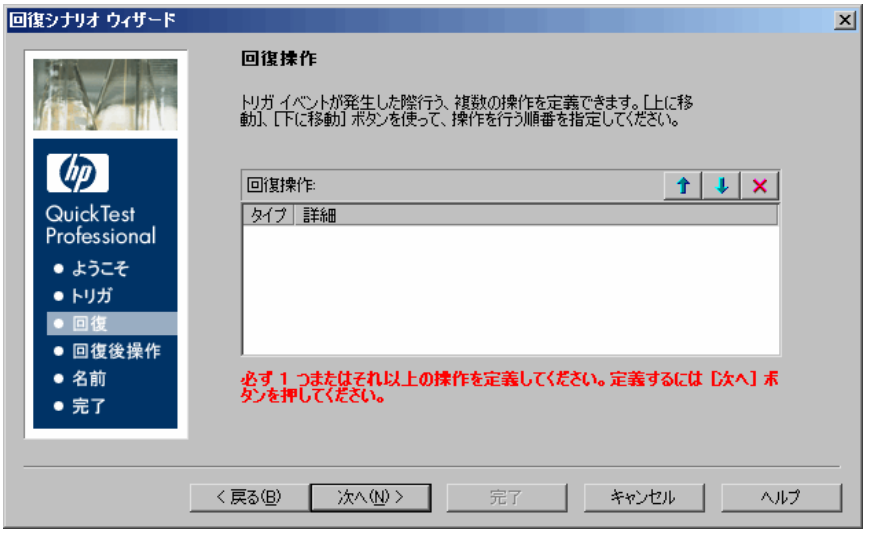
- ［**プロセス**］リストからプロセスを削除するには，プロセスを選択して［**プロセスの削除**］ボタンをクリックします。

ヒント：プロセスの名前を変更するには，［**プロセス**］リストでプロセスを選択し，プロセス名をクリックして編集します。

［**次へ**］をクリックして［回復操作］画面に進みます（887 ページを参照してください）。

［回復操作］画面

［回復操作］画面では，回復シナリオにおける一連の回復操作を管理できます。回復操作とは，QuickTest によってトリガ・イベントが認識されたときに順次実行される操作です。



少なくとも 1 つの回復操作を定義する必要があります。回復操作を定義し，それを［**回復操作**］リストに追加するには，［**次へ**］をクリックして，［回復操作］画面に進みます（888 ページを参照してください）。

2 つ以上の回復操作を定義する場合，回復操作を選択し，［**上に移動**］または［**下に移動**］ボタンを使用して，QuickTest による回復操作の実行順序を変更できます。回復操作を選択し，［**削除**］ボタンをクリックして，回復シナリオから回復操作を削除することもできます。

注：［**Microsoft Windows の再起動**］回復操作を定義する場合，常に最後の回復操作としてこの操作が挿入されるため，リストで位置は変更できません。

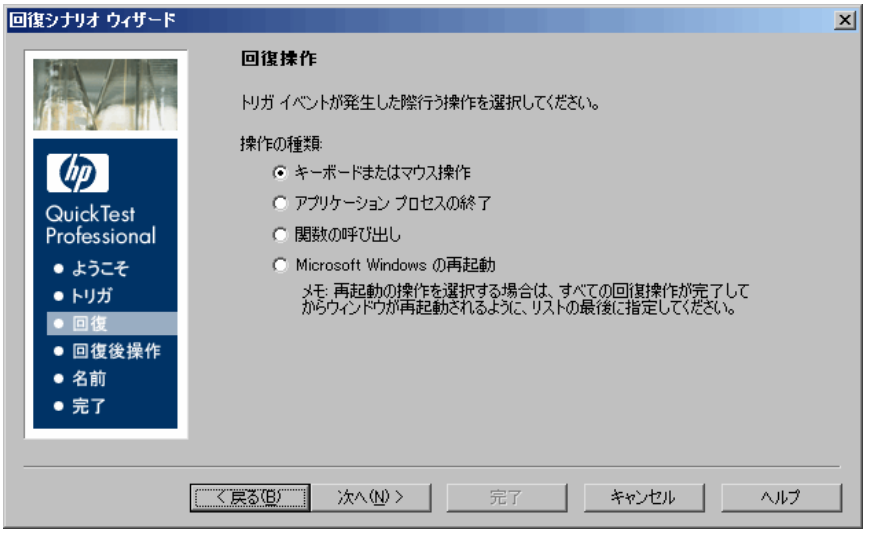
回復操作を 1 つでも定義すると，［**他の回復操作を追加する**］チェック・ボックスが表示されます。

➤ ほかの回復操作を定義するには，このチェック・ボックスを選択し，［**次へ**］をクリックします。

➤ チェック・ボックスをクリアし，［**次へ**］をクリックして［回復後のテスト実行オプション］画面に進みます（894 ページを参照してください）。

［回復操作］画面

［回復操作］画面では，QuickTest によるトリガ・イベント検出後に実行される操作を指定できます。



回復操作のタイプを選択し，［**次へ**］をクリックします。ウィザードに表示される次の画面は，選択する回復操作のタイプに応じて異なります。

次のタイプの回復操作を定義できます。

➤ ［**キーボードまたはマウス操作**］：QuickTest によって，ウィンドウのボタンをクリックする操作，またはキーボードのキーを押す操作がシミュレートされます。このオプションを選択し，［**次へ**］をクリックして，［回復操作 - ボタン、またはキーを押す］画面に進みます（889 ページを参照してください）。

➤ ［**アプリケーション プロセスの終了**］：QuickTest によって，指定のプロセスが閉じます。このオプションを選択し，［**次へ**］をクリックして，［回復操作 - プロセスの終了］画面に進みます（891 ページを参照してください）。

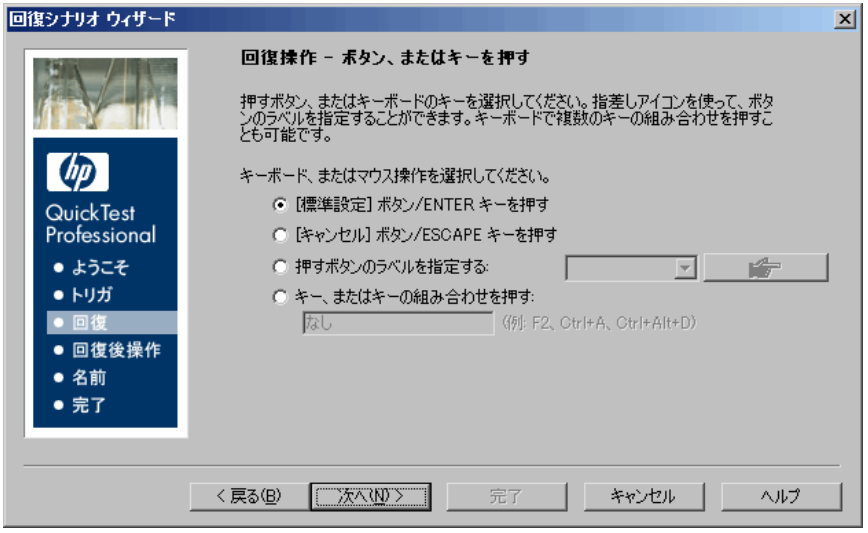
➤ ［**関数の呼び出し**］: QuickTest によって，VBScript 関数が呼び出されます。このオプションを選択し，［**次へ**］をクリックして，［回復操作 - 関数の呼び出し］画面に進みます（893 ページを参照してください）。

➤ ［**Microsoft Windows の再起動**］: QuickTest によって Microsoft Windows が再起動されます。このオプションを選択し，［**次へ**］をクリックして，［回復操作］画面に進みます（887 ページを参照してください）。

注：［**Microsoft Windows の再起動**］回復操作を使用する場合，操作を実行する前に，この回復シナリオと関連付けられているコンポーネントをすべて保存する必要があります。また，再起動時に自動ログインするように，コンポーネントを実行するコンピュータを設定する必要があります。

［回復操作 - ボタン、またはキーを押す］画面

［回復操作］画面で［**キーボードまたはマウス操作**］回復操作を選択した場合（888 ページを参照してください），［回復操作 - ボタン、またはキーを押す］画面が開きます。



QuickTest によってトリガ・イベントが検出された場合に実行させる，キーボードまたはマウスの操作を指定します。

- ［**[標準設定］ボタン /ENTER キーを押す**］：トリガが発生した場合に，表示されているウィンドウで，標準のボタンをクリックする，または ENTER キーを押すよう QuickTest に指示します。
- ［**[キャンセル］ボタン /ESCAPE キーを押す**］：トリガが発生した場合に，表示されているウィンドウで，［**キャンセル**］ボタンをクリックする，または ESCAPE キーを押すよう QuickTest に指示します。
- ［**押すボタンのラベル指定する**］：トリガが発生した場合に，表示されているウィンドウで，指定したラベルの付いたボタンをクリックするよう QuickTest に指示します。このオプションを選択した場合は，指差しマークをクリックした後，トリガ・ウィンドウの中の任意の場所をクリックします。指差しマークの使用方法の詳細については，890 ページ「指差しマークの使い方のヒント」を参照してください。

  選択されたウィンドウのボタン・ラベルはすべて，リスト・ボックスに表示されます。リストから必要なボタンを選択します。
- ［**キー、またはキーの組み合わせを押す**］：トリガが発生した場合に，表示されているウィンドウで，指定したキーボードのキーまたはキーの組み合わせを押すよう QuickTest に指示します。このオプションを選択した場合は，エディット・ボックスをクリックした後，指定する単独のキーまたはキーの組み合わせを押します。

  ［**次へ**］をクリックします。［回復操作］画面が再び開き，定義したキーボードまたはマウスの回復操作が表示されます。

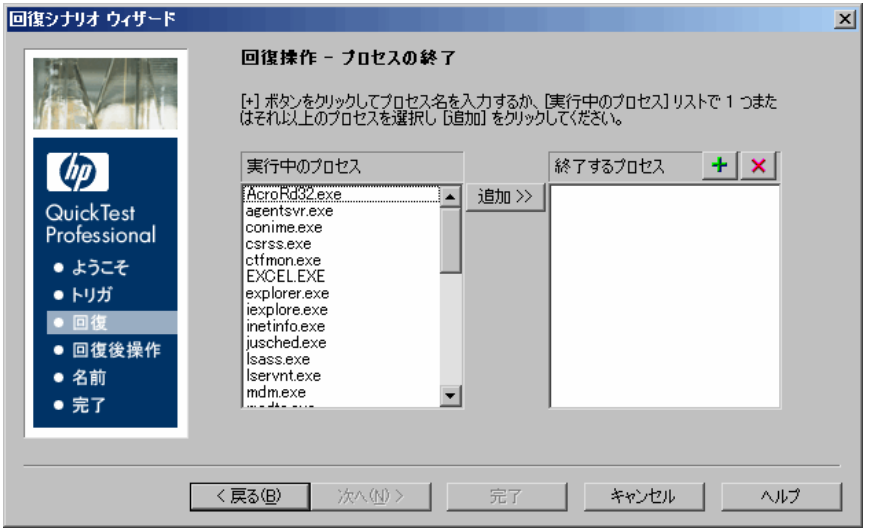
指差しマークの使い方のヒント

- 左側の CTRL キーを押し下げると，指差しマークが標準のポインタに変わります。これによって，ウィンドウのフォーカスを変更したり，ショートカット・メニューを表示するために右クリックする，スクロール・バーを使用する，オブジェクト上にポインタを移動するなどの操作を QuickTest またはアプリケーションで実行したりできるようになります。
- 選択するオブジェクトを含んでいるウィンドウが部分的に別のウィンドウに隠れている場合は，ウィンドウが前面に表示されるまで，部分的に隠れているウィンドウに指差しマークを数秒間置いたままにします。それから，必要なオブジェクトをポイントしてクリックします。ウィンドウを前面に出すのに必要な時間の設定は，［オプション］ダイアログ・ボックスの［一般］表示枠で行えます。

- 選択対象のオブジェクトを含んでいるウィンドウが別のウィンドウの背後に完全に隠れている場合，または，ダイアログ・ボックスがウィンドウの背後に隠れている場合は，必要に応じて左側の CTRL キーを押しながらウィンドウを配置します。
- 選択対象オブジェクトを含んでいるウィンドウが最小化されている場合は，左の CTRL キーを押したまま，Windows タスク・バー内のアプリケーションを右クリックして，ショートカット・メニューから［**元のサイズに戻す**］を選択することで，ウィンドウを表示できます。
- 特定のイベント（ショートカット・メニューを表示するために右クリックしたり，オブジェクトにポインタを合わせるなど）が生じないと選択対象オブジェクトが表示されない場合は，左 CTRL キーを押し下げます。すると，指差しマークが一時的に標準のポインタに変わり，イベントを実行できるようになります。選択対象のオブジェクトが表示されたら，左側の CTRL キーを放します。ポインタが再び指差しマークに変わります。

［回復操作 - プロセスの終了］画面

［回復操作］画面の［**アプリケーション プロセスの終了**］回復操作を選択すると（888 ページを参照してください），［回復操作 - プロセスの終了］画面が開きます。



［**実行中のプロセス**］リストには，現在実行中のすべてのアプリケーション・プロセスが表示されます。［**終了するプロセス**］リストには，トリガが呼び出されたときに閉じるアプリケーション・プロセスが表示されます。

➤ ［**実行中のプロセス**］リストからプロセスを追加するには，［**実行中のプロセス**］リストのプロセスをダブルクリックするか，プロセスを選択して［**追加**］ボタンをクリックします。Windows で複数選択する際の標準的な方法（CTRL キーや SHIFT キー）を使用して，複数のプロセスを選択できます。

➤ ［**終了するプロセス**］リストにプロセスを直接追加するには，［**新規プロセスの追加**］ボタンをクリックし，リストに追加するプロセスの名前を入力します。

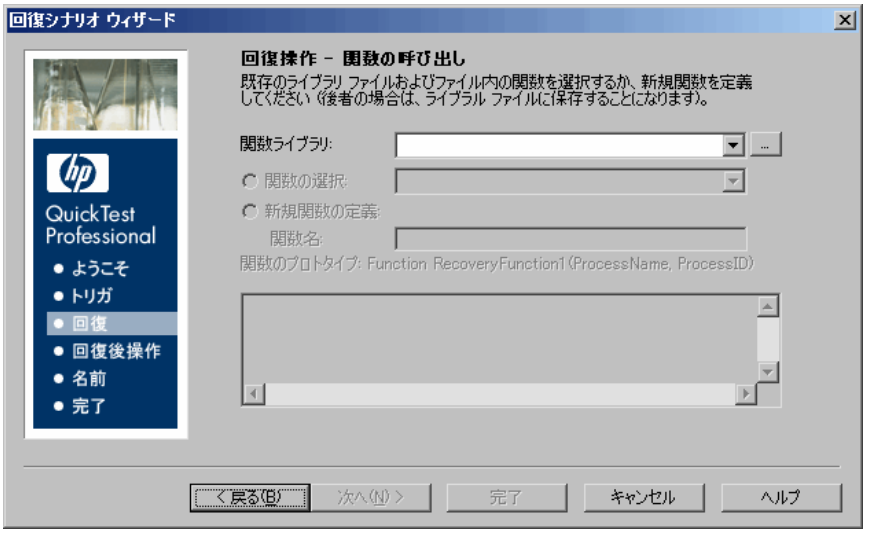
➤ ［**終了するプロセス**］リストからプロセスを削除するには，プロセスを選択して［**プロセスの削除**］ボタンをクリックします。

ヒント：プロセスの名前を変更するには，［**終了するプロセス**］リストでプロセスを選択し，プロセス名をクリックして編集します。

［**次へ**］をクリックします。［回復操作］画面が再び開き，定義した「プロセスを終了する」回復操作が表示されます。

[回復操作 - 関数の呼び出し] 画面

[回復操作] 画面で [**関数の呼び出し**] 回復操作を選択すると（888 ページを参照してください），[回復操作 - 関数の呼び出し] 画面が開きます。



[**関数ライブラリ**] ボックスで，最近指定した関数ライブラリを選択します。あるいは，参照ボタンをクリックして，既存の関数ライブラリに移動します。

注：関数ライブラリは Quality Center プロジェクトに格納されている必要があります。

関数ライブラリを選択した後，次のオプションのいずれか 1 つを選択します。

➤ [**関数の選択**]：選択した関数ライブラリから既存の関数を選択します。

876 ページ「[トリガ イベントの選択] 画面」で選択したトリガ・タイプのプロトタイプ構文と一致する関数のみが表示されます。

各トリガ・タイプのプロトタイプを次に示します。

テスト実行エラーのトリガ
OnRunStep
(
[in] Object as Object: 現在のステップのオブジェクト。
[in] Method as String: 現在のステップのメソッド。
[in] Arguments as Array: 実際のメソッドの引数。
[in] Result as Integer: 実際のメソッドの結果。
)

ポップアップ・ウィンドウとオブジェクトの状態のトリガ
OnObject
(
[in] Object as Object: 検出されたオブジェクト。
)

アプリケーションのクラッシュのトリガ
OnProcess
(
[in] ProcessName as String: 検出されたプロセスの名前。
[in] ProcessId as Integer: 検出されたプロセスの ID。
)

➤ [**新規関数の定義**]: 一意の名前を指定し，表示される関数のプロトタイプに従って［**関数名**］ボックスで関数を定義することによって，新規の関数を作成します。新しい関数が，選択した関数ライブラリに追加されます。

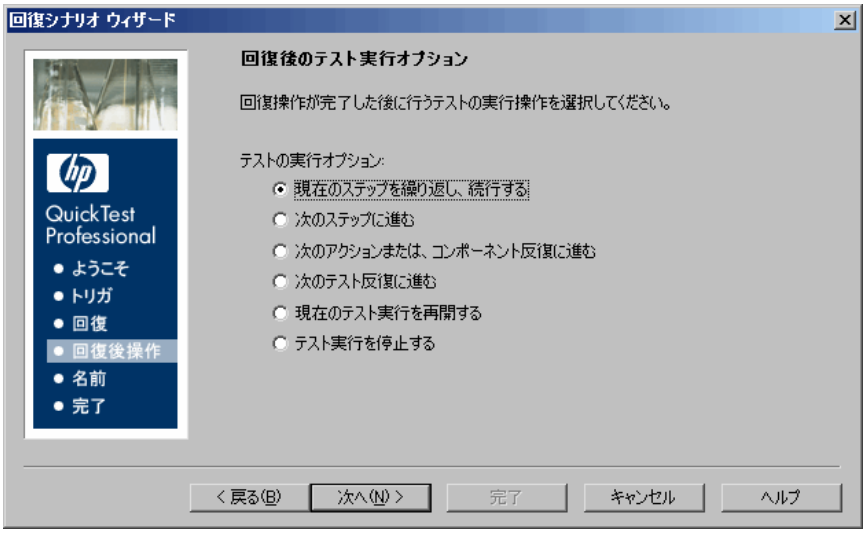
注：複数のシナリオで異なる関数ライブラリから同一の名前を持つ関数を使用すると，回復プロセスは失敗することがあります。この場合，実行セッション中に回復の失敗に関する情報が表示されます。

［**次へ**］をクリックします。［回復操作］画面が再び開き（887 ページを参照してください），定義した関数の操作が表示されます。

［回復後のテスト実行オプション］画面

［回復操作］画面の［**他の回復操作を追加する**］チェック・ボックスをクリアし（887 ページを参照してください），［**次へ**］をクリックすると，［回復後のテスト実行オプション］画面が開きます。回復後のテスト実行のオプションで

は，QuickTest によってイベントが識別され，指定の回復オプションがすべて実行された後の，実行セッションの継続方法を指定します。



定義した回復操作が実行された後，QuickTest では，次の実行セッションのオプションのいずれか 1 つを実行できます。

➤ **[現在のステップを繰り返し、続行する]**

この現在のステップとは，回復シナリオがトリガされた際に QuickTest が実行中であったステップです。回復シナリオに対して「**エラー発生時**」呼び出しオプションを使用している場合，通常，エラーを返すステップは，トリガ・イベントの発生原因となったステップより 1 つ以上後のステップです。

したがってほとんどの場合，現在のステップを繰り返しても，トリガ・イベントは再現されません。詳細については，907 ページ「回復シナリオの有効化と無効化」を参照してください。

➤ **[次のステップに進む]**

回復シナリオがトリガされた際に QuickTest が実行中であったステップをスキップします。アプリケーションに対して操作を実行するステップをスキップすると，それ以降のステップが失敗する場合があることに注意してください。

➤ **［次のアクションまたはコンポーネント反復に進む］**

現在のアクションまたはコンポーネントの反復でのステップ実行を停止し，最初（または現在のアクションまたはコンポーネントを追加的に反復する必要がない場合は次のアクションまたはコンポーネント）から次のアクションの反復を開始します。

➤ **［次のテスト反復に進む］**

現在のコンポーネントでのステップ実行を停止し，最初から次の QuickTest ビジネス・プロセス・テストの反復を開始します（またはコンポーネントを追加的に反復する必要がない場合はコンポーネントの実行を停止します）。

➤ **［現在のテスト実行を再開する］**

ステップの実行を停止し，コンポーネントを始めから再実行します。

➤ **［テストの実行を停止する］**

コンポーネントの実行を停止します。

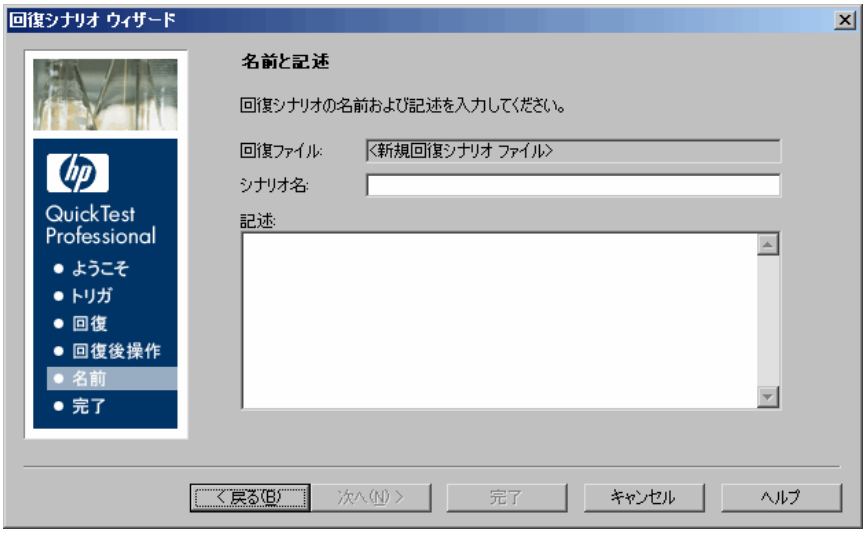
注：回復操作として［**Microsoft Windows の再起動**］を選択した場合，上記の最後 2 つのテスト実行のオプションのみから選択できます。

テスト実行オプションを選択し，［**次へ**］をクリックして，［名前と記述］画面に進みます（896 ページを参照してください）。

［名前と記述］画面

［回復後のテスト実行オプション］画面でテスト実行オプションを指定し（894 ページを参照してください），［**次へ**］をクリックすると，［名前と記述］画面が開きます。

［名前と記述］画面では，回復シナリオを識別するための名前を指定します。シナリオに関する説明情報を追加することもできます。

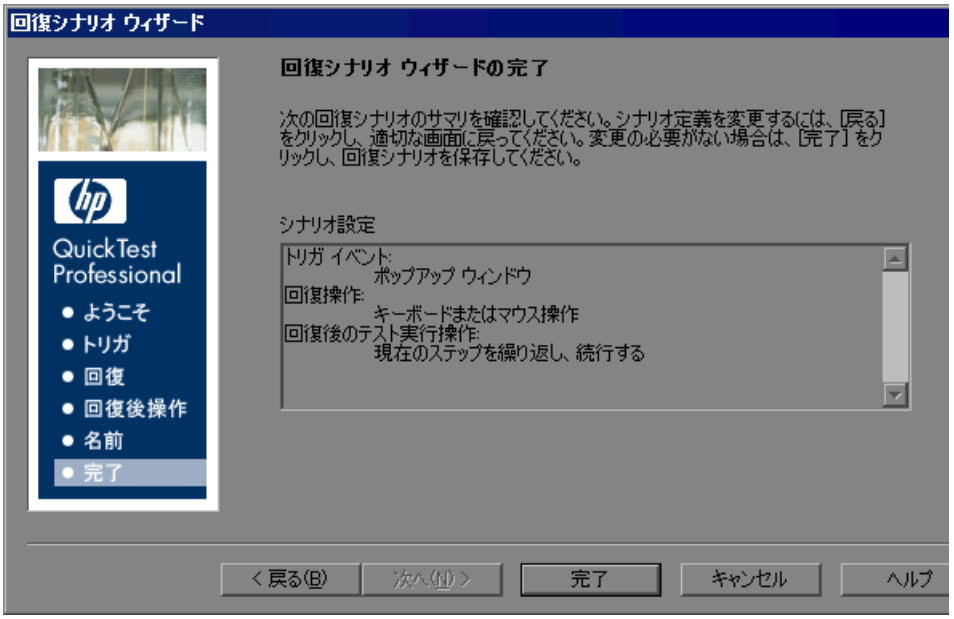


回復シナリオの名前とテキスト形式の説明を入力し，［**次へ**］をクリックして，［回復シナリオ ウィザードの完了］画面に進みます（898 ページを参照してください）。

[回復シナリオ ウィザードの完了] 画面

[名前と記述] 画面で回復シナリオの名前と説明を指定し（896 ページを参照してください），[**次へ**] をクリックすると，[回復シナリオ ウィザードの完了] 画面が開きます。

[回復シナリオ ウィザードの完了] 画面では，定義したシナリオ設定の概要を確認できます。



注： コンポーネントの回復シナリオを，コンポーネントのアプリケーション領域に関連付けます。また，特定のアプリケーション領域に関連付けられているすべての新しいコンポーネントに対して，標準設定の回復シナリオを定義することもできます。詳細については，429 ページ「アプリケーション領域の管理」を参照してください。

[**完了**] をクリックすると，回復シナリオの定義は完了します。

回復ファイルでの回復シナリオの保存

回復シナリオ・ウィザードを使用して，回復ファイル内の回復シナリオの作成または変更を行ったら，その回復ファイルを保存する必要があります。

ヒント：回復ファイルを保存せずに［回復シナリオ マネージャ］ダイアログ・ボックスの［**閉じる**］ボタンをクリックすると，回復ファイルを保存するよう求められます。［**はい**］をクリックし，下記の手順 2 に進みます。既存の回復ファイル内のシナリオの追加または変更を行い，メッセージに対して［**はい**］をクリックすると，回復ファイルとそのシナリオは保存されます。

新規の回復ファイルまたは変更した回復ファイルを保存するには，次の手順を実行します。

1 ［回復シナリオ マネージャ］ダイアログ・ボックスの［**保存**］ボタンをクリックします。既存の回復ファイル内のシナリオの追加または変更を行うと，回復ファイルとそのシナリオが保存されます。新規の回復ファイルを使用している場合は，［回復シナリオの保存］ダイアログ・ボックスが表示されます。

ヒント：［**保存**］ボタンの右側の矢印をクリックし，［**名前を付けて保存**］を選択して，別の名前で回復ファイルを保存することもできます。

---

2 サイドバーで，ファイルを保存する場所（たとえば，**ファイル・システム**や Quality Center **テスト・リソース**など）を選択します。

3 ファイルを保存するフォルダを参照し，そのフォルダを選択します。

4［**ファイル名**］ボックスにファイルの名前を入力し，［**保存**］をクリックします。

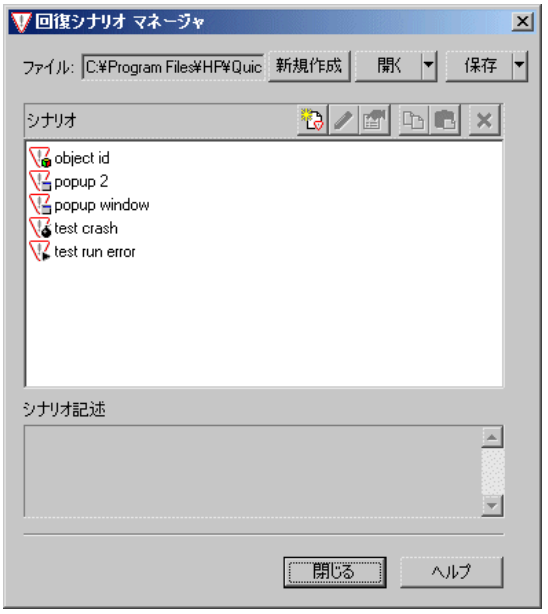
注：ファイル・システムまたは Quality Center 9.x のリソースへのパスを指定した場合は，そのパス，あるいはそのパスの一部が［オプション］ダイアログ・ボックスの［フォルダ］表示枠（［**ツール**］＞［**オプション**］＞［**フォルダ**］ノード）に存在するか検査されます。パスが存在する場合は，入力したパスの相対的な部分だけを使用してパスを定義するように求められます。パスが存在しない場合は，リソースのパスを［フォルダ］表示枠に追加して，相対的なパスを定義するように求められます。詳細については，629 ページ「グローバル・テスト・オプションの設定」を参照してください。

Quality Center 10.00 のリソースおよび依存関係モデルを使用している場合は，Quality Center の絶対パスを指定する必要があります。詳細については，940 ページ「Quality Center で相対パスを使用する場合の注意事項」を参照してください。

回復ファイルは，ファイル拡張子 **.qrs** 付きで，指定の場所に保存されます。

回復シナリオの管理

回復シナリオを作成したら，回復シナリオ・マネージャを使用してシナリオを管理できます。



回復シナリオ・マネージャには，次の回復シナリオ・アイコンがあります。

アイコン	説明
	実行セッション中，開いているアプリケーションでウィンドウがポップアップしたときに回復シナリオがトリガされることを示します。
	対象回復シナリオが，アプリケーション内のオブジェクトのプロパティ値が特定の値に一致したときに起動されることを示します。
	コンポーネントのステップが正しく実行されないときに回復シナリオがトリガされることを示します。
	実行セッション中に，開いているアプリケーションが失敗したときに回復シナリオがトリガされることを示します。

回復シナリオ・マネージャでは，次の方法で既存のシナリオを管理できます。

- 回復シナリオのプロパティの表示
- 回復シナリオの変更
- 回復シナリオの削除
- 回復シナリオ・ファイル間での回復シナリオのコピー

回復シナリオのプロパティの表示

定義済みのすべての回復シナリオのプロパティを表示できます。

回復シナリオのプロパティを表示するには，次の手順を実行します。

1 ［**シナリオ**］ボックスで，プロパティを表示する回復シナリオを選択します。

2 ［**プロパティ**］ボタンをクリックします。あるいは，［**シナリオ**］ボックスのシナリオをダブルクリックする方法もあります。［回復シナリオのプロパティ］ダイアログ・ボックスが開きます。

［回復シナリオのプロパティ］ダイアログ・ボックスには，選択したシナリオに関する，次の読み取り専用の情報が表示されます。

- ［**一般**］タブ：回復シナリオに対して定義されている名前と説明に加えて，シナリオが保存されている回復ファイルの名前とパスが表示されます。
- ［**トリガ イベント**］タブ：回復シナリオに対して定義されているトリガ・イベントの設定が表示されます。

➤ [**回復操作**] タブ：回復シナリオに対して定義されている回復操作が表示されます。

➤ [**回復後の操作**] タブ：回復シナリオに対して定義されている回復後の操作が表示されます。

回復シナリオの変更

既存の回復シナリオの設定を変更できます。

回復シナリオを変更するには，次の手順を実行します。

1 [**シナリオ**] ボックスで，変更するシナリオを選択します。

2 [**編集**] ボタンをクリックします。回復シナリオ・ウィザードが開き，選択した回復シナリオに対して定義した設定が表示されます。

3 回復シナリオ・ウィザードを操作して，必要に応じて詳細を変更します。回復シナリオ・ウィザードのオプションの詳細については，870 ページ「回復シナリオの定義」を参照してください。

注：変更は，[回復シナリオ マネージャ] ダイアログ・ボックスで [**保存**] をクリックするまで保存されません。変更を保存せずに [回復シナリオ マネージャ] ダイアログ・ボックスの [**閉じる**] ボタンをクリックすると，回復ファイルを保存するよう求められます。[**はい**] をクリックし，変更を保存します。

回復シナリオの削除

必要のない既存の回復シナリオは削除できます。回復シナリオ・マネージャから回復シナリオを削除すると，回復シナリオ・ファイルからは対応する情報も削除されます。

注：削除した回復シナリオがコンポーネントと関連付けられている場合，QuickTest では，実行セッション中はその回復シナリオが無視されます。

回復シナリオを削除するには，次の手順を実行します。

1 [**シナリオ**] ボックスで，削除するシナリオを選択します。

2 [**削除**] ボタンをクリックします。[回復シナリオ マネージャ] ダイアログ・ボックスから，回復シナリオが削除されます。

注： シナリオは，[回復シナリオ マネージャ] ダイアログ・ボックスで [**保存**] をクリックするまで実際には削除されません。削除を保存せずに [回復シナリオ マネージャ] ダイアログ・ボックスの [**閉じる**] ボタンをクリックすると，回復ファイルを保存するよう求められます。[**はい**] をクリックして回復シナリオ・ファイルを保存し，シナリオを削除します。

回復シナリオ・ファイル間での回復シナリオのコピー

ある回復シナリオ・ファイルから別の回復シナリオ・ファイルに，回復シナリオをコピーできます。

ある回復シナリオ・ファイルから別の回復シナリオ・ファイルに回復シナリオをコピーするには，次の手順を実行します。

1 [**シナリオ**] ボックスで，コピーする回復シナリオを選択します。

2 [**コピー**] ボタンをクリックします。シナリオがクリップボードにコピーされます。

3 [**開く**] ボタンをクリックして，シナリオのコピー先の回復シナリオ・ファイルを選択するか，[**新規作成**] ボタンをクリックして，シナリオをコピーする回復シナリオ・ファイルを新規作成します。

4 [**貼り付け**] ボタンをクリックします。シナリオが新規の回復シナリオ・ファイルにコピーされます。

注：

- 回復シナリオ・ファイルに同じ名前のシナリオがすでに存在する場合，それをコピーした新しいシナリオで置き換えるかどうかを選択できます。
- 変更は，［回復シナリオ マネージャ］ダイアログ・ボックスで［**保存**］をクリックするまで保存されません。変更を保存せずに［回復シナリオ マネージャ］ダイアログ・ボックスの［**閉じる**］ボタンをクリックすると，回復ファイルを保存するよう求められます。［**はい**］をクリックし，変更を保存します。

アプリケーション領域への回復シナリオの関連付け

回復シナリオを作成した後は，作成した回復シナリオと選択したコンポーネント（アプリケーション領域を介して）を関連付け，トリガ・イベントが発生した場合，実行セッション中に QuickTest によって適切なシナリオが実行されるようにします。シナリオに優先順位を付け，実行セッション中 QuickTest がシナリオを適用する順序を設定できます。アプリケーション領域に関連付けられている特定のシナリオ，またはすべてのシナリオを無効にすることもできます。アプリケーション領域内のコンポーネントの回復シナリオを定義します。詳細については，429 ページ「アプリケーション領域の管理」を参照してください。

回復シナリオのプロパティの表示

アプリケーション領域と関連付けられているすべての回復シナリオのプロパティを表示できます。

注 : [回復シナリオ マネージャ] ダイアログ・ボックスから，回復シナリオの設定を変更します。詳細については，903 ページ「回復シナリオの変更」を参照してください。

回復シナリオのプロパティを表示するには，次の手順を実行します。

1 アプリケーション領域の一般表示枠の中で，[**追加設定**] ボタンをクリックします。[アプリケーション領域の設定] ダイアログ・ボックスが開きます。

2 [**回復**] ノードをクリックします。

3 [**シナリオ**] ボックスで，プロパティを表示する回復シナリオを選択します。

4 [**プロパティ**] ボタンをクリックします。あるいは，[**シナリオ**] ボックスのシナリオをダブルクリックする方法もあります。[回復シナリオのプロパティ] ダイアログ・ボックスが開き，選択したシナリオの設定に関する読み取り専用の情報が表示されます。詳細については，902 ページ「回復シナリオのプロパティの表示」を参照してください。

回復シナリオの優先順位の設定

実行セッション中に，関連付けられたシナリオが QuickTest によって実行される順序を指定できます。トリガ・イベントが発生した場合，QuickTest によって，[アプリケーション領域の設定] ダイアログ・ボックスの [回復] 表示枠に表示されている順序に従って該当する回復シナリオが確認されます。

回復シナリオの優先順位を設定するには，次の手順を実行します。

1 アプリケーション領域の一般表示枠の中で，[**追加設定**] ボタンをクリックします。[アプリケーション領域の設定] ダイアログ・ボックスが開きます。

2 [**回復**] ノードをクリックします。

3 [**シナリオ**] ボックスで，優先順位を変更するシナリオを選択します。

4 [**上に移動**] ボタンまたは [**下に移動**] ボタンをクリックします。選択に従って，選択したシナリオの優先順位が変更されます。

アプリケーション領域からの回復シナリオの削除

［アプリケーション領域の設定］ダイアログ・ボックスを使用して，特定のシナリオとアプリケーション領域の間の関連付けを削除できます。アプリケーション領域からシナリオを削除した後も，まだシナリオ自体は存在していますが，QuickTest によって実行セッション中にシナリオが実行されることはありません。

回復シナリオをアプリケーション領域から削除するには，次の手順を実行します。

1 アプリケーション領域の一般表示枠の中で，［**追加設定**］ボタンをクリックします。［アプリケーション領域の設定］ダイアログ・ボックスが開きます。

2 ［**回復**］ノードをクリックします。

3 ［**シナリオ**］ボックスで，削除するシナリオを選択します。

4 ［**削除**］ボタンをクリックします。選択したシナリオは，アプリケーション領域との関連付けが解除されます。

回復シナリオの有効化と無効化

［アプリケーション領域の設定］ダイアログ・ボックスの［回復］表示枠では，特定のシナリオを有効または無効にしたり，QuickTest による回復シナリオのメカニズムの呼び出し条件を指定したりできます。特定のシナリオを無効にしても，アプリケーション領域との関連付けは残りますが，実行セッション中にそのシナリオが QuickTest によって実行されることはありません。そのシナリオは後で有効にできます。

また，回復シナリオを呼び出す条件を指定することもできます。

特定の回復シナリオを有効 / 無効にするには，次の手順を実行します。

1 アプリケーション領域の一般表示枠の中で，［**追加設定**］ボタンをクリックします。［アプリケーション領域の設定］ダイアログ・ボックスが開きます。

2 ［**回復**］ノードをクリックします。

3 ［**シナリオ**］ボックスで，次のいずれかを実行します。

- シナリオを有効にするには，各シナリオの左側にあるチェック・ボックスを選択します（複数選択も可）。
- シナリオを無効にするには，各シナリオの左側にあるチェック・ボックスをクリアします。

回復メカニズムを呼び出すか条件を定義するには，次の手順を実行します。

［**回復シナリオのアクティブ化**］ボックスにある次のオプションのいずれかを選択します。

- ［**各ステップごと**］: 回復メカニズムはステップごとに起動されます。［**各ステップごと**］を選択すると，テストの実行中にパフォーマンスが低下する場合があります。
- ［**エラー発生時**］: 回復メカニズムはエラーの戻り値を返したステップの後にのみ起動されます。

  エラーを返すステップは，多くの場合，例外イベントの発生原因となるステップとは同じではありません。

  たとえば，チェック・ボックスを選択するステップによって，ポップアップ・ダイアログ・ボックスが開いたとします。ポップアップ・ダイアログ・ボックスがトリガ・イベントとして定義されてはいるものの，チェック・ボックスを選択するステップの実行は成功しているため，QuickTest の処理は次のステップに進みます。続くいくつかのステップでは，アプリケーションに対する操作の実行を必要としないチェックポイント，関数，その他の条件ステートメントまたはループ・ステートメントが実行される可能性があります。そして，ポップアップ・ダイアログ・ボックスによって実行が妨げられる操作をアプリケーションに対して実行するように QuickTest に指示するステップが登場するには，10 個のステートメントを経てようやく，ということが考えられます。この場合，エラーを返し，回復メカニズムをトリガしてダイアログ・ボックスを閉じるのは，この 10 番目のステップです。回復操作が完了した後は，現在のステップはこの 10 番目のステップであり，トリガ・イベントの原因となったステップではありません。
- ［**なし**］: 回復メカニズムは無効にされます。

ヒント：実行セッション中，アプリケーション領域に関連付けられている特定のシナリオ，またはすべてのシナリオをプログラムの中から有効 / 無効にすることもできます。詳細については，909 ページ「プログラムによる回復メカニズムの制御」を参照してください。

すべての新しいコンポーネント用の標準回復シナリオの設定

すべての新規コンポーネントに対する標準の回復シナリオの定義は，コンポーネントのアプリケーション領域で行います。詳細については，429 ページ「アプリケーション領域の管理」を参照してください。

プログラムによる回復メカニズムの制御

回復オブジェクトを使用すれば，実行セッション中にプログラムの中で回復メカニズムを制御できます。たとえば，回復メカニズム全体を有効または無効にしたり，実行セッションの一部分で特定の回復シナリオを有効または無効にしたりできます。また，特定の回復シナリオに関するステータス情報を取得したり，実行セッションの特定の時点で回復メカニズムを明示的に呼び出したりできます。

標準設定では，実行セッション中にエラーが返された場合，QuickTest によって回復トリガが調べられます。回復オブジェクトの Activate メソッドを使用すると，QuickTest に，実行セッションの特定のステップの後でトリガを調べさせることができます。たとえば，オブジェクト・プロパティ・チェックポイントの実行時に，あるプロセスが開いていると，そのチェックポイントが失敗するとわかっているとします。アプリケーションにおける別の問題である可能性があるため，こうした開いているプロセスがチェックポイントの成功または失敗に影響を及ぼさないようにする必要があります。

しかし，チェックポイントの失敗は，実行エラーにはなりません。そのため，標準設定では，回復メカニズムがオブジェクトの状態によって呼び出されることはありません。オブジェクトのプロパティが特定の状態にあるときに指定の開いているプロセスを探して閉じる回復シナリオを定義できます。オブジェクトのプロパティの状態は，問題のあるプロセスが開いている場合の値を示します。QuickTest に対して，チェックポイントが失敗したときに回復メカニズムを呼び出させ，問題のプロセスが開いていないか調べさせて，あれば閉じさせ，失敗したチェックポイントを実行しなおすように指示できます。これにより，チェックポイントが 2 回目に実行されるときには，チェックポイントは開かれているプロセスによる影響を受けなくなります。

回復オブジェクトとそのメソッドの詳細については，『**HP QuickTest Professional Object Model Reference**』（英語版）を参照してください。

第 38 章

QuickTest 操作のオートメーション

QuickTest を使用してアプリケーションのテストを自動化するのと同じように，QuickTest Professional のオートメーション・オブジェクト・モデルを使用して QuickTest 操作を自動化できます。QuickTest のオートメーション・オブジェクト・モデルによって公開されているオブジェクト，メソッド，およびプロパティを使用すれば，QuickTest のオプションを設定したりコンポーネントを実行したりするスクリプトを作成して，QuickTest のインタフェースを使用して手動で行う方法の代わりに，これらを行うプログラムを作成できます。

オートメーション・スクリプトは，同じ作業を複数回実行する場合や，複数のコンポーネントを対象に実行する場合，あるいは，特定の環境またはアプリケーションでのニーズに合わせて QuickTest をすばやく設定する場合に特に有用です。

本章の内容

- QuickTest 操作の自動化について（912 ページ）
- QuickTest オートメーション・スクリプトを使用する条件（913 ページ）
- オートメーション・スクリプトの設計と実行に使用するプログラミング言語と開発環境の選択（914 ページ）
- QuickTest オートメーション・スクリプトの基本要素の学習（916 ページ）
- オートメーション・スクリプトの生成（917 ページ）
- QuickTest オートメーション・リファレンスの使用方法（918 ページ）

QuickTest 操作の自動化について

QuickTest Professional オートメーション・オブジェクト・モデルを使用して QuickTest の操作を自動化するスクリプトを作成できます。QuickTest オートメーション・オブジェクト・モデルは，別のアプリケーションから QuickTest を制御できるようにするオブジェクト，メソッド，およびプロパティを提供します。

オートメーションとは

「**オートメーション**」とは，あるアプリケーション内のソフトウェア・オブジェクトを別のアプリケーションからアクセスできるようにする Microsoft 社の技術です。これらのオブジェクトは，VBScript や VC++ などのスクリプティング言語またはプログラミング言語を使用して作成して操作できます。オートメーションを利用することで，アプリケーションの機能をプログラムの中から制御できるようになります。

「**オブジェクト・モデル**」とは，システムまたはアプリケーションの実装を構成するソフトウェア・オブジェクト（クラス）を構造化して表したものです。オブジェクト・モデルは，クラスとインタフェースのセットに加えて，プロパティ，メソッド，およびイベント，そしてそれらの関係を定義します。

QuickTest オートメーション・オブジェクト・モデルとは

QuickTest のインタフェースを通じて提供されるほとんどの設定および実行機能は，オブジェクト，メソッド，およびプロパティを通じて，QuickTest オートメーション・オブジェクト・モデルにおいて何らかの方法で表されます。必ずしも 1 対 1 の関係は示せませんが，QuickTest の大半のダイアログ・ボックスには対応するオートメーション・オブジェクトがあり，ダイアログ・ボックスの大半のオプションは対応するオブジェクト・プロパティを使用して設定と取得が可能で，ほとんどのメニュー・コマンドその他の操作は対応するオートメーション・メソッドがあります。

QuickTest オートメーション・オブジェクト・モデルによって公開されているオブジェクト，メソッド，およびプロパティを，ループや条件判断ステートメントなどの標準のプログラミング要素と組み合わせてスクリプトを設計できます。

オートメーション・スクリプトは，同じ作業を複数回実行する場合や，複数のコンポーネントを対象に実行する場合，あるいは，特定の環境またはアプリケーションでのニーズに合わせて QuickTest をすばやく設定する場合に特に有用です。

たとえば，Microsoft Visual Basic からオートメーション・スクリプトを作成および実行して，コンポーネントに必要なアドインの読み込み，可視モードでの QuickTest の起動，コンポーネントを開く操作，［オプション］，［ビジネス コンポーネントの設定］，および［記録と実行環境設定］ダイアログ・ボックスでの対応する設定，コンポーネントの実行，およびコンポーネントの保存ができます。

以後，スクリプトに簡単なループを追加し，1 つのスクリプトで上記の操作を複数のコンポーネントに対して実行するようにできます。

また，QuickTest を特定の設定で起動する初期化スクリプトを作成することも可能です。そうしておけば，テスト担当者全員にこのオートメーション・スクリプトを使用して QuickTest を起動するように指示し，テスト担当者の全員が必ず同じ設定で作業を行っていることを保証できます。

QuickTest オートメーション・スクリプトを使用する条件

役に立つ QuickTest オートメーション・スクリプトを作成するには，計画，設計時間，およびテストが必要です。常に，初期投資と，時間がかかったり煩雑だったりする作業を自動化することで実現される時間と人的資源の節約とを天秤にかけなければなりません。

何度も繰り返し実行する必要があったり，定期的に実行する必要のある QuickTest 操作は，QuickTest オートメーション・スクリプトのよい候補です。

次にいくつかの有用な QuickTest オートメーション・スクリプトを示します。

- **初期化スクリプト**：QuickTest を自動的に起動し，特定の環境でのテストに必要なオプションおよび設定を指定するスクリプトを作成できます。
- **コンポーネントの保守**コンポーネントの集合に対して繰り返し実施されるスクリプトを記述して，特定の目標を達成できます。次に例を示します。
  - **値の更新**：スクリプトを記述して適切なアドインとともに各コンポーネントを開き，更新されたアプリケーションに対して更新モードを実行し，すべてのコンポーネントの値を更新するときには保存してアプリケーションの更新された値と一致させることができます。
  - **既存のコンポーネントへの新規オプションの適用**：QuickTest の新しいバージョンにアップグレードしたときに，既存のコンポーネントに適用したいオプションが新しいバージョンに存在する場合があります。既存のコンポーネ

ントをそれぞれ開いて，新しいオプションのための値を設定し，変更を保存して閉じるというスクリプトを作成できます。

➤ **ほかのアプリケーションからの QuickTest の呼び出し**：QuickTest オートメーション・スクリプトを実行するオプションやコントロールを持った独自のアプリケーションを設計できます。たとえば，QuickTest に精通していない製品マネージャでも QuickTest の実行予定を立てられるような Web フォームや簡単な Windows インタフェースを作成できます。

オートメーション・スクリプトの設計と実行に使用するプログラミング言語と開発環境の選択

オートメーション・スクリプトの作成に使用できるオブジェクト指向プログラミング言語がいくつかあります。それぞれの言語に対して，オートメーション・スクリプトの設計と実行に使用できるいくつかの開発環境が提供されています。

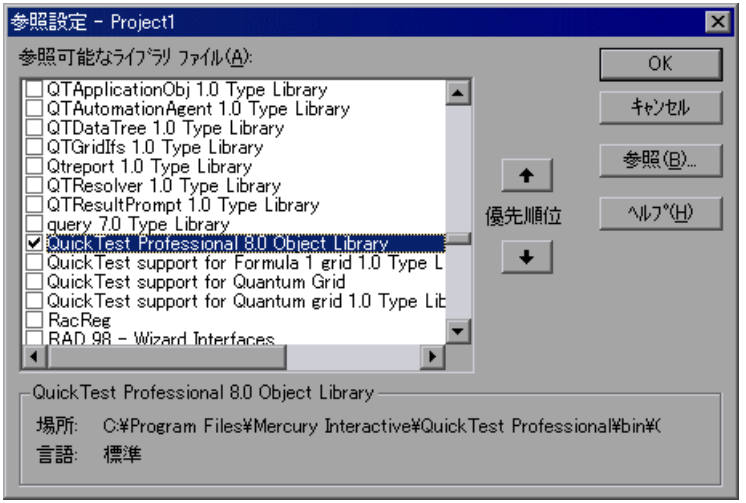
オートメーション・スクリプトの作成

QuickTest オートメーション・スクリプトはオートメーションをサポートする任意の言語および開発環境を使用して作成できます。たとえば，次が使用できます。VBScript，JavaScript，Visual Basic，Visual C++，Visual Studio .NET。

タイプ・ライブラリの参照がサポートされている開発環境もあります。**タイプ・ライブラリ**とは，オブジェクトの記述，インタフェース，その他，オブジェクト・モデルの定義を含んでいるバイナリ・ファイルです。

タイプ・ライブラリの参照をサポートする開発環境を選択した場合，Microsoft IntelliSense，自動ステートメント補完，ステータス・バーのツールチップなどの機能を，スクリプトの作成時に利用することができます。QuickTest オートメーション・オブジェクト・モデルは，**QTObjectModel.dll** という名前のタイプ・ライブラリ・ファイルを提供します。このファイルは，**＜ QuickTest のインストール・フォルダ＞ ¥bin** に格納されています。

タイプ・ライブラリをサポートする環境を選択する場合には，オートメーション・スクリプトの作成および実行を開始する前に，必ず QuickTest タイプ・ライブラリを参照するようにします。たとえば，Microsoft Visual Basic を使用して作業をする場合には，［**プロジェクト**］＞［**参照設定**］を選択してプロジェクトの［参照設定］ダイアログ・ボックスを開きます。続いて「**QuickTest Professional ＜バージョン＞ Object Library**」（ここで，＜**バージョン**＞は現在インストールされている QuickTest オートメーション・タイプ・ライブラリのバージョン）を選択します。



オートメーション・スクリプトの実行

オートメーション・スクリプトを実行できるアプリケーションがいくつあります。Microsoft Windows Script Host を使用すれば，オートメーション・スクリプトをコマンド・ラインからも実行できます。

たとえば，次のコマンド・ラインを使用してオートメーション・スクリプトを実行できます。

```
WScript.exe  /E:VBSCRIPT myScript.vbs
```

QuickTest オートメーション・スクリプトの基本要素の学習

大半のオートメーション・オブジェクト・モデルと同様に，QuickTest オートメーション・オブジェクト・モデルのルート・オブジェクトは，**Application** オブジェクトです。Application オブジェクトは，QuickTest のアプリケーション・レベルを表します。このオブジェクトを使用して，**テスト**・オブジェクト（コンポーネント・ドキュメントを表します），Options オブジェクト（［オプション］ダイアログ・ボックスを表します），または Addins コレクション（［アドイン］ダイアログ・ボックスのアドインのセットを表します）など，QuickTest の要素を返したり，アドインのロード，QuickTest の起動，コンポーネントの起動と保存，および QuickTest の終了などの操作を実行したりできます。

Application オブジェクトによって返されるオブジェクトはそれぞれほかのオブジェクトを返したり，オブジェクトに関係する操作を行ったり，オブジェクトに関連付けられているプロパティの取得と設定を行ったりできます。

オートメーション・スクリプトは必ず QuickTest Application オブジェクトの作成から始まります。オブジェクトを作成しても QuickTest は起動しません。QuickTest オートメーション・オブジェクト・モデルのほかのオブジェクト，メソッド，プロパティにアクセスするためのオブジェクトを用意するだけです。

注：必要があれば，オブジェクトを作成する対象となるリモート QuickTest コンピュータ（スクリプトを実行するコンピュータ）を指定することも可能です。詳細については，**QuickTest Professional ヘルプの『QuickTest Automation Object Model Reference』**（英語版）の「**Introduction**」の項にある「**Running Automation Programs on a Remote Computer**」を参照してください。

スクリプトの残りの部分は，スクリプトの目的に応じて異なります。QuickTest を開始する前に，コンポーネントに対応するアドインの取得，アドインのロード，QuickTest を可視モードでの開始など，いくつかの操作を行うことができます。

これらの準備を行った後，QuickTest がコンピュータでまだ起動されていなければ，Application.Launch メソッドを使用して QuickTest を起動できます。オートメーション・スクリプトの大半の操作は，Launch メソッドの後に実行します。

オートメーション・プログラムで実行できる操作の詳細については，オンラインの『**HP QuickTest Professional Object Model Reference**』（英語版）を参照してください。このヘルプ・ファイルの詳細については，918 ページ「QuickTest オートメーション・リファレンスの使用方法」を参照してください。

必要な操作を完了したら，あるいは，ロードされているアドインのセットを変更する場合など，QuickTest をいったん終了して再度起動する操作を実行したい場合には，Application.Quit メソッドを使用します。

オートメーション・スクリプトの生成

，［オプション］ダイアログ・ボックスの［一般］表示枠および，［オブジェクトの認識］ダイアログ・ボックスのそれぞれに，［**スクリプトの生成**］ボタンがあります。このボタンをクリックすると，該当するダイアログ・ボックスの現在の設定を含んだオートメーション・スクリプト・ファイル（**.vbs**）が生成されます。

生成されたスクリプトをそのまま実行すれば，スクリプトの生成に使用した QuickTest アプリケーションとまったく同じ設定で QuickTest を起動できます。また，生成されたファイルから特定の行をコピーして，自分のオートメーション・スクリプトに貼り付けることもできます。

たとえば，［オプション］ダイアログ・ボックスから生成したスクリプトは，次のようになります。

```
Dim App 'As Application
Set App = CreateObject("QuickTest.Application")
App.Launch
App.Visible = True
App.Options.DisableVORecognition = False
App.Options.AutoGenerateWith = False
App.Options.WithGenerationLevel = 2
App.Options.TimeToActivateWinAfterPoint = 500
...
...
App.Options.WindowsApps.NonUniqueListItemRecordMode = "ByName"
App.Options.WindowsApps.RecordOwnerDrawnButtonAs = "PushButtons"
App.Folders.RemoveAll
```

［**スクリプトの作成**］ボタン，［オプション］および［オブジェクトの認識］ダイアログ・ボックスで利用できるオプションの詳細については，第 6 章「オブジェクト認識の設定」および 第 24 章「グローバル・テスト・オプションの設定」を参照してください。を参照してください。

QuickTest オートメーション・リファレンスの使用方法

『**QuickTest Automation Object Model Reference**』（英語版）は，QuickTest オートメーション・オブジェクト・モデルのオブジェクト，メソッド，プロパティに関する詳細な説明，構文情報，使用例を提供するヘルプ・ファイルです。

『**HP QuickTest Professional Automation Object Model Reference**』（英語版）を開くには，次の場所にアクセスします。

- QuickTest プログラム・フォルダ（［**スタート**］＞［**プログラム**］＞［**QuickTest Professional**］＞［**文書**］＞［**QuickTest オートメーション リファレンス**］）
- QuickTest のメイン・ヘルプ（［**ヘルプ**］＞［**QuickTest Professional ヘルプ**］＞［**HP QuickTest Professional 詳細リファレンス**］＞［**HP QuickTest Professional オートメーション・オブジェクト・モデル**］）

第 XI 部

Quality Center を使用した作業

第 39 章

Quality Center との統合

アプリケーションの包括的なテストを確実に行うためには，通常，多くのコンポーネントを作成して実行する必要があります。HP Quality Center は，品質の集中管理ソリューションであり，テスト・プロセスの整理と管理に役立ちます。

注：本章に含まれる Quality Center の説明は，特に注記しないかぎり，現在サポートされている Quality Center のすべてのバージョンに適用されます。ただし，機能やオプションがお使いの Quality Center のエディションでサポートされていない可能性があります。

サポートされている Quality Center のバージョンについては，『**HP QuickTest Professional 最初にお読みください**』の一覧を参照してください。

Quality Center のエディションの詳細については，『**Mercury Quality Center ユーザーズ・ガイド**』を参照してください。

本章の内容

- Quality Center を使用した作業について（922 ページ）
- Quality Center との接続と切断（923 ページ）
- QuickTest の Quality Center との統合（929 ページ）

Quality Center を使用した作業について

QuickTest は HP の品質集中管理ソリューションである Quality Center と統合できます。Quality Center は，アプリケーションの機能のあらゆる側面を網羅するあらゆる種類のテスト（QuickTest テスト，ビジネス・プロセス・テスト，マニュアル・テスト，他の HP 製品作成されるテストなど）で構成されるプロジェクトの保守を支援します。プロジェクトでは，アプリケーションの特定のテスト要件を満たすように各テストを設計します。また，プロジェクトの目標を達成するため，プロジェクトで行うテストを個別のグループにまとめます。

Quality Center には，テストのスケジュール設定と実行，結果の収集，結果の分析，テストのバージョン管理などを行うための直感的かつ効率的なメソッドが用意されています。不具合を追跡するためのシステムも備わっているため，不具合の検出から解決に至るまで，不具合を厳密に監視することができます。

Quality Center プロジェクトとは，特定のテスト・プロセスに関係するデータを収集し，格納するためのデータベースです。QuickTest から Quality Center プロジェクトにアクセスするには，Quality Center がインストールされているローカルまたはリモートの Web サーバに接続する必要があります。QuickTest を Quality Center に接続すると，ビジネス・プロセス・テストおよびコンポーネントを作成し，Quality Center プロジェクトに保存することができます。テストおよびコンポーネントの実行後には，Quality Center で結果を確認できます。

Quality Center で QuickTest を使用するには，次のアクセス許可が必要です。

- Quality Center キャッシュ・フォルダ（Quality Center のクライアント側にあります）の読み取りおよび書き込みの許可
- Quality Center 用 QuickTest Add-in のインストール先フォルダに対する読み取りおよび書き込みの許可。

ヒント：さまざまな QuickTest アドインの詳細については，『**HP QuickTest Professional アドイン・ガイド**』を参照してください。

Quality Center を使って作業している場合，Quality Center プロジェクトのテスト・リソース・モジュールに格納されている外部ファイルをコンポーネントのアプリケーション領域に関連付けることができます。

不具合は，発生するたび自動的に，または手作業で QuickTest の［テスト結果］ウィンドウから Quality Center プロジェクトに直接報告できます。Quality Center プロジェクトへの自動または手作業による不具合の報告については，718 ページ「Quality Center プロジェクトへの手動による実行セッション中に検出された不具合の送信」を参照してください。

Quality Center での作業の詳細については，『**Mercury Quality Center ユーザーズ・ガイド**』を参照してください。QuickTest と Quality Center の統合に関する最新情報およびヒントについては，『**HP QuickTest Professional 最初にお読みください**』（［**スタート**］＞［**プログラム**］＞［**QuickTest Professional**］＞［**最初にお読みください**］から利用できます）を参照してください。

Quality Center との接続と切断

QuickTest および Quality Center の両方で作業する場合，QuickTest を Quality Center プロジェクトに接続できます。

テスト・プロセスのどの時点でも，Quality Center プロジェクトと QuickTest とを接続したり切断したりできます。ただし，Quality Center から QuickTest テストを開いているときや，QuickTest で Quality Center の共有リソース（共有オブジェクト・リポジトリやデータ・テーブル・ファイルなど）を使用しているときは，Quality Center と QuickTest との接続を切断しないでください。

注：Quality Center の現在サポートされているバージョンには接続できます。Quality Center のサポートされているバージョンについては，『**HP QuickTest Professional 最初にお読みください**』の一覧を参照してください。詳細については，929 ページ「Quality Center Connectivity Add-in」を参照してください。

注：Business Process Testing を使用するには，Quality Center プロジェクトが格納されている Quality Center サーバに QuickTest を接続する必要があります。このサーバによって，QuickTest と Quality Center プロジェクト間の接続が処理されます。Quality Center プロジェクトでは，テスト対象アプリケーションのコンポーネントおよび実行セッションに関する情報が格納されます。この情報には，ビジネス・プロセス・テストを作成して実行するために必要なすべてのリソース・ファイルおよび設定などが含まれています。初めて Quality Center サーバおよびプロジェクトに QuickTest を接続すると，QuickTest によって，標準設定の Business Process Testing フォルダとファイルがプロジェクトに設定されます。これによりビジネス・コンポーネントに必要なリソースと設定を準備できるほか，直観的なキーワード駆動型のキーワード・ビューを使ってビジネス・コンポーネントの作成，操作，デバッグを行うことができます。

QuickTest の Quality Center への接続

接続プロセスには 2 つの段階があります。まず，QuickTest をローカルまたはリモートの Quality Center サーバに接続します。このサーバによって，QuickTest と Quality Center プロジェクト間の接続が処理されます。

次に，ログインし，QuickTest のアクセス対象となるプロジェクトを選択します。このプロジェクトには，テスト対象アプリケーションのテストと実行セッションの情報が保存されます。Quality Center プロジェクトは，パスワードで保護されるため，ユーザ名とパスワードを指定する必要があります。

QuickTest を Quality Center サーバに接続するには，次の手順を実行します。

1 ［**ファイル**］＞［**Quality Center への接続**］を選択するか，［**Quality Center への接続**］ツールバー・ボタンをクリックします。［Quality Center への接続 - サーバへの接続］ダイアログ・ボックスが表示されます。

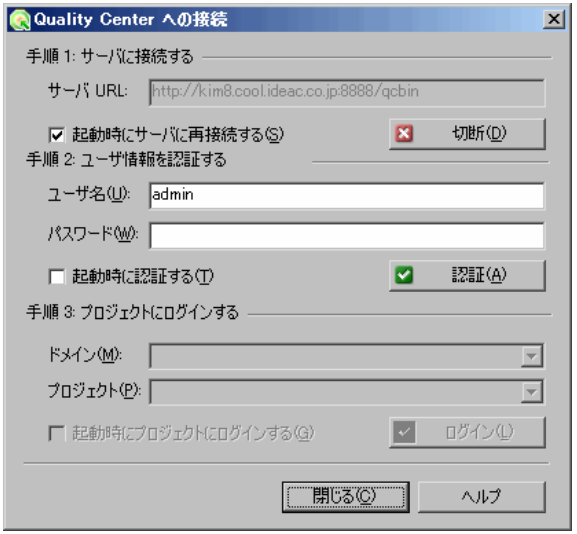
2 [**サーバ URL**] ボックスに，Quality Center がインストールされている Web サーバの URL アドレスを入力します。

注： ローカル・エリア・ネットワーク（LAN）または広域エリア・ネットワーク（WAN）を介してアクセス可能な Quality Center サーバを選択できます。

---

3 次回の QuickTest の起動時に，Quality Center サーバに自動的に再接続するには，[**起動時にサーバに再接続する**] チェック・ボックスを選択します。

4 [**接続**] をクリックします。[Quality Center への接続] ダイアログ・ボックスが表示されます。



Quality Center サーバの名前が読み取り専用形式で [サーバ URL] ボックスに表示されます。

5 [**ユーザ名**] ボックスに，Quality Center ユーザ名を入力します。

6 [**パスワード**] ボックスに，Quality Center のパスワードを入力します。

7［**認証**］をクリックし，Quality Center サーバに対してユーザ情報を認証させます。

ユーザ情報の認証が完了すると，［**ユーザ情報を認証する**］領域のエディット・ボックスが読み取り専用形式で表示されます。［**認証**］ボタンが［**ユーザを変更**］ボタンに変わります。

ヒント：同じ Quality Center サーバに別のユーザ名でログインするには，［**ユーザを変更**］をクリックし，新しいユーザ名とパスワードを入力して，再び［**認証**］をクリックします。

8［**ドメイン**］ボックスで，Quality Center プロジェクトが保存されているドメインを選択します。接続する権限があるドメインだけが表示されます。

9［**プロジェクト**］ボックスで，作業対象のプロジェクトを選択します。ユーザが定義されているプロジェクトのみが表示されます。

10［**ログイン**］をクリックします。

11 次回の QuickTest の起動時に，Quality Center サーバに自動的に再接続するには，［**起動時にサーバに再接続する**］チェック・ボックスを選択します。

12［**起動時にサーバに再接続する**］チェック・ボックスをオンにすると，［**起動時に認証する**］チェック・ボックスが有効になります。次回の QuickTest の起動時に，ユーザ情報を自動的に認証させるには，［**起動時に認証する**］チェック・ボックスを選択します。

13［**起動時に認証する**］チェック・ボックスをオンにすると，［**起動時にプロジェクトにログインする**］チェック・ボックスが有効になります。起動時に選択したプロジェクトにログインするには，［**起動時にプロジェクトにログインする**］チェック・ボックスを選択します。

14［**閉じる**］をクリックし，［Quality Center への接続］ダイアログ・ボックスを閉じます。ステータス・バーに，QuickTest が Quality Center プロジェクトに接続されていることを示す Quality Center アイコンが表示されます。

ヒント：現在の Quality Center 接続を表示するには，ステータス・バーの **Quality Center** アイコンにマウス・ポインタを合わせます。QuickTest が接続されている Quality Center サーバおよびプロジェクトの名前がツールチップに表示されます。［Quality Center への接続］ダイアログ・ボックスを開くには **Quality Center** アイコンをダブルクリックします。

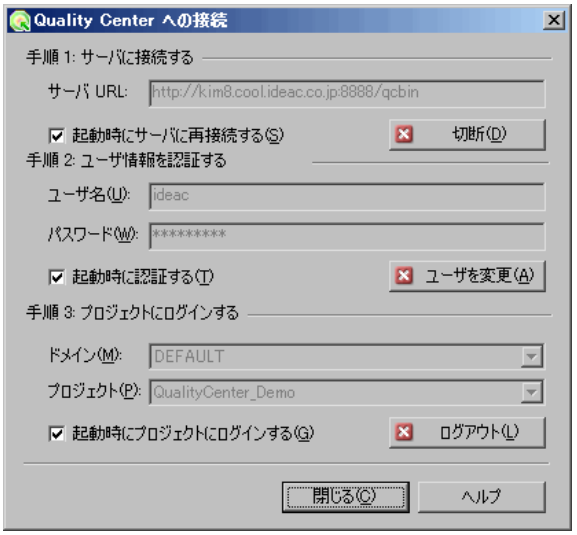
Quality Center からの QuickTest の切断

QuickTest は Quality Center プロジェクト，または Quality Center サーバから，いつでも切断できます。最初にプロジェクトへの接続を解除せずに，Quality Center サーバへの QuickTest の接続を解除すると，QuickTest から当該プロジェクト・データベースへの接続は自動的に解除されます。

注：Quality Center から切断するときに Quality Center テストまたは共有ファイル（共有オブジェクト・リポジトリやデータ・テーブル・ファイルなど）が開いていると，これらは QuickTest によって閉じられます。

Quality Center サーバから QuickTest を切断するには，次の手順を実行します。

1 ［**ファイル**］＞［**Quality Center への接続**］を選択するか，［**Quality Center への接続**］ツールバー・ボタンをクリックします。［Quality Center への接続］ダイアログ・ボックスが表示されます。



2 選択したプロジェクトから QuickTest を切断するには，［**手順 3: プロジェクトにログインする**］領域で［**ログアウト**］をクリックします。

3 選択した Quality Center サーバから QuickTest を切断するには，［**手順 1: サーバに接続する**］領域で［**切断**］をクリックします。

ヒント：同じ Quality Center サーバに別のユーザ名でログインするには，［**ユーザを変更**］をクリックし，新しいユーザ名とパスワードを入力して，再度［**認証**］をクリックします。

---

4 ［**閉じる**］をクリックし，［Quality Center への接続］ダイアログ・ボックスを閉じます。

QuickTest の Quality Center との統合

QuickTest の Quality Center との統合により，Quality Center プロジェクト内のファイルの格納やアクセスが可能になりました。また，QCUtil オブジェクトを使用して，Quality Center オープン・テスト・アーキテクチャ API に用意されている広範な種類の機能にアクセスできます。

Quality Center Connectivity Add-in

Quality Center Connectivity Add-in を使用して，QuickTest を Quality Center と統合します。このアドインは，［Quality Center への接続］ダイアログ・ボックスを使用して QuickTest を Quality Center に接続するときに，QuickTest コンピュータに自動的にインストールされます。Quality Center でこのアドインを手動でインストールするには，Quality Center アドインのページから［**ヘルプ**］＞［**Add-ins Page**］＞［**HP Quality Center Connectivity**］を選択します。

現在コンピュータにインストールされている Quality Center Connectivity Add-in のバージョンを表示するには，［**ヘルプ**］＞［**QuickTest Professional のバージョン情報**］を選択し，［**製品情報**］ボタンをクリックします。詳細については，96 ページ「製品情報の表示」を参照してください。

Quality Center との統合

最も基礎的なこととして，QuickTest の Quality Center との統合により，QuickTest が Quality Center と接続している場合，Quality Center プロジェクトの QuickTest コンポーネント，アプリケーション領域，およびリソース・ファイルの格納やアクセスが可能になります。

リソースおよび依存関係モデルで提供されるすべての機能を利用できます。詳細については，937 ページ「リソースおよび依存関係モデルの使用」を参照してください。

また，関数ライブラリで QCUtil オブジェクトを使用することにより，Quality Center OTA（オープン・テスト・アーキテクチャ）の全機能に対するアクセスや使用が可能になります。これにより，Quality Center データベースに不具合を直接報告するなど，実行セッション中の統合操作を自動化することができます。詳細については，『**HP QuickTest Professional Object Model Reference**』（英語版）の「**Utility**」の項，および『**Quality Center Open Test Architecture**』（英語版）のドキュメントを参照してください。

QuickTest 自動スクリプトの TDOTA オブジェクトを使用して，Quality Center OTA にアクセスすることもできます。詳細については，『**QuickTest Professional Automation Object Model Reference**』（英語版）（［**ヘルプ**］＞［**QuickTest Professional ヘルプ**］＞［**HP QuickTest Professional Advanced References**］＞［**HP QuickTest Professional Automation Object Model**］を選択）を参照してください。

［テストの新規作成］ダイアログ・ボックスにあります **QuickTest リモート・エージェントの設定**

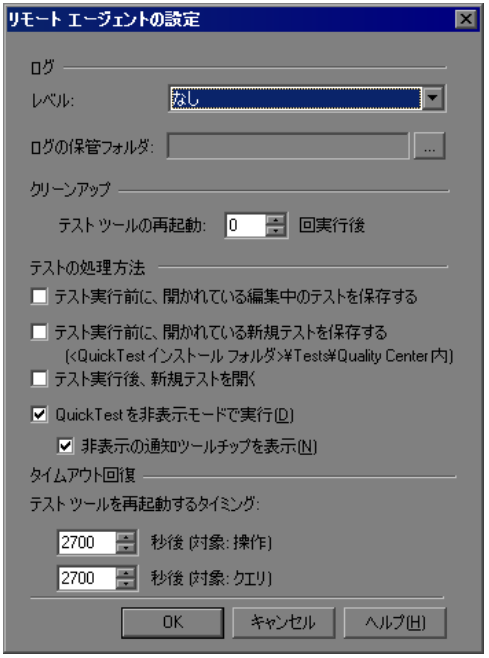
Quality Center から QuickTest ビジネス・プロセス・テストを実行すると，Quality Center コンピュータで QuickTest リモート・エージェントが開きます。テストが Quality Center などのリモートのアプリケーションによって実行された場合の QuickTest の動作は，QuickTest リモート・エージェントによって決まります。

いつでも［リモート エージェントの設定］ダイアログ・ボックスを開いて，Quality Center がコンピュータでテストを実行するときに QuickTest アプリケーションが使用している設定の表示や変更ができます。

［リモート エージェントの設定］ダイアログ・ボックスを開くには，次の手順を実行します。

1 ［**スタート**］＞［**プログラム**］＞［**QuickTest Professional**］＞［**Tools**］＞［**Remote Agent**］を選択します。リモート・エージェントが開き，**リモート・エージェント**のアイコンがタスクバー・トレイに表示されます。

2 **リモート・エージェント**・アイコンを右クリックし，［**設定**］を選択します。［リモート エージェントの設定］ダイアログ・ボックスが開きます。



3 ダイアログ・ボックスで設定の表示や修正ができます。詳細については，932 ページ「［リモート エージェントの設定］ダイアログ・ボックスについて」を参照してください。

4 ［**OK**］をクリックし，設定を保存してダイアログ・ボックスを閉じます。

5 **リモート・エージェント**・アイコンを右クリックして［**終了**］を選択し，リモート・エージェント・セッションを終了します。

［リモート エージェントの設定］ダイアログ・ボックスについて

［リモート エージェントの設定］ダイアログ・ボックスでは，Quality Center がコンピュータで QuickTest ビジネス・プロセス・テストを実行するときに QuickTest が使用する設定の表示や変更ができます。

［リモート エージェントの設定］ダイアログ・ボックスには，次のオプションが含まれています。

オプション	説明
［**レベル**］	Quality Center が QuickTest ビジネス・プロセス・テストを実行するときに作成されるログに含まれる内容のレベルです。 ［**なし**］（標準設定）：ログは作成されません。 ［**低**］：ログには Quality Center と QuickTest の間の通信エラーがすべて表示されます。 ［**中**］：ログには Quality Center と QuickTest の間の通信エラーと，Quality Center と QuickTest の間の通信に関するほかの重要な操作の情報が含まれます。 ［**高**］：ログには Quality Center と QuickTest の通信に関する，すべての利用可能な情報が含まれています。
［**ログの保管フォルダ**］	ログ・ファイルが格納されているフォルダのパスです。［**レベル**］オプションでログの種類を指定した場合に必要です。
［**テスト ツールの再起動 X 回実行後**］	ビジネス・プロセス・テストの場合，Quality Center がコンポーネントの反復を指定されている回数完了させたら，QuickTest を再起動します。ただし，ビジネス・プロセス・テスト実行の最中に反復の指定されている回数に達した場合，再起動は現在のビジネス・プロセス・テストの反復が終了してから行われます。 このオプションを使用して利用可能なメモリを最大限にできます。 テスト・セット中に QuickTest を再起動しない場合は，0（標準設定）を入力してください。

オプション	説明
［テスト実行前に、開かれている編集中のテストを保存する］	リモート・エージェントによってテストの実行が開始されるときに，QuickTest で既存の（命名済みの）ビジネス・コンポーネントが開かれている場合，このオプションにより，開いているビジネス・コンポーネントに加えられたすべての未保存の変更を保存するよう，QuickTest に指示されます。 **注**：リモート・エージェントによってテストの実行が開始されるときに，QuickTest で既存の（命名済みの）関数ライブラリが開かれている場合，関数ライブラリは保存されません。
［テスト実行前に、開かれている新規テストを保存する］	リモート・エージェントによってテストの実行が開始されるときに，QuickTest で新規の（命名されていない）ビジネス・コンポーネントまたは関数ライブラリが開かれている場合，関数ライブラリは保存されません。
［テスト実行後、新規テストを開く］	標準設定では，すべてのテストの実行が終わったとき，リモート・エージェントによって最後に実行されたテストは QuickTest で開いたままです。ただし，開いているテストに共有リソース（共有オブジェクト・リポジトリやデータ・テーブル・ファイルなど）が関連付けられている場合，これらのリソースは，テストが閉じられるまでほかのユーザにロックされています。このオプションを選択すれば，Quality Center が実行する最後のテストを閉じ，代わりに空のテストを開くことができます。

オプション	説明
［**QuickTest を非表示モードで実行**］	Quality Center でテストのラボ・モジュールのテスト・セットを実行する場合，非表示（サイレント）モードで QuickTest を実行するかどうかを指定します。標準設定では，このオプションは選択されています。 ［**非表示の通知ツールチップを表示**］：このチェック・ボックスが選択されると，QuickTest が Quality Center テストを非表示モードで実行する場合，リモート・エージェントはツールチップ・ウィンドウを表示します。ツールチップをクリックして，テスト・セットの実行中に QuickTest を表示できます。標準設定では，このオプションは選択されています。 **注：** ➤ 通知ツールチップをクリックすると，［**QuickTest を非表示モードで実行**］チェック・ボックスがクリアされ，QuickTest は通常モードで実行されます。QuickTest を再度非表示モードで実行するには，次のテスト・セットを実行する前に［**QuickTest を非表示モードで実行**］を選択し直します。 ➤ 非表示モードで実行している場合，各テストの終わり，または各テスト・セットの終わりに，必要に応じて QuickTest を再表示できます。この動作は，Quality Center サイト管理で SUPPORT_TESTSET_END パラメータを使用して設定します。詳細については，『**HP Quality Center Administrator Guide**』（英語版）の Quality Center の設定パラメータに関する項を参照してください。

オプション	説明
［テスト ツールを再起動するタイミング］	次に示す項目に指定されている秒数が経過しても応答がない場合，QuickTest を再起動します。 **操作**：**開く**，**実行**などの QuickTest 操作です。 **クエリ**：リモートのアプリケーションが実行してアプリケーションの応答を確認する，標準のステータス・クエリです（Quality Center の **get_status** クエリなど)。 両方のオプションの標準設定の値は 2700 秒（45 分）です。ただし，QuickTest 操作の応答時間が長くなることはあっても，クエリの応答時間は通常数秒しかかかりません。したがって，それぞれのオプションに異なる値を設定するとよいでしょう。 **注**：QuickTest で変更を保存していない関数ライブラリが開かれている場合，QuickTest により保存するよう促すメッセージが表示されます。関数ライブラリが 10 秒以内に保存されなければ，QuickTest は再起動し，保存していない変更は失われます。

第 40 章

リソースおよび依存関係モデルの使用

QuickTest では，リソースおよび依存関係モデルを使用して，QuickTest コンポーネントを Quality Center プロジェクトに完全に統合できます。

注：本章に含まれる Quality Center の説明は，Quality Center 10.00 に適用されます。ただし，機能やオプションがお使いの Quality Center のエディションでサポートされていない可能性があります。Quality Center のエディションの詳細については，『**Mercury Quality Center ユーザーズ・ガイド**』を参照してください。

本章の内容

- リソースおよび依存関係モデルの用語（938 ページ）
- リソースおよび依存関係モデルについて（939 ページ）
- アセットの依存関係を使用する利点（941 ページ）
- Quality Center でのリソースおよび依存関係モデルの使用（942 ページ）

リソースおよび依存関係モデルの用語

用語	説明
アセット	すべての QuickTest テスト・ドキュメントまたはリソース・ファイル。次が含まれます。 ➤ コンポーネント ➤ アプリケーション領域 ➤ 関数ライブラリ ➤ 共有オブジェクト・リポジトリ ➤ 回復シナリオ **注**：Quality Center では，QuickTest アセットは，より一般的な用語である「**エンティティ**」と呼ばれます。
リソース	コンポーネントで使用されるすべてのアセット。たとえば，コンポーネントは関連付けられた関数ライブラリの関数呼び出しを含んでおりコンポーネントのアプリケーション領域に関連付けられている共有オブジェクト・リポジトリに格納されているテスト・オブジェクトを参照する可能性があります。リソースには次が含まれます。 ➤ 関数ライブラリ ➤ 共有オブジェクト・リポジトリ ➤ 回復シナリオ
依存関係	リソースと特定のアプリケーション領域間のリンクされた関係。関連付けられているリソース・ファイルは，そのリソースを使用する各コンポーネントのアプリケーション領域にリンクされています。 絶対パスを使用して関連付けられているアセットが次のモジュールに格納されている場合，アセットは依存関係とみなされます。 ➤ **ビジネス・コンポーネント・モジュール**：コンポーネント ➤ **テスト・リソース・モジュール**：関数ライブラリ，共有オブジェクト・リポジトリ，回復シナリオ **注**：ビジネス・コンポーネント・モジュールの **Obsolete** フォルダに格納されているコンポーネントは，コンポーネントに関連付けられていないため，依存関係とみなされません。

リソースおよび依存関係モデルについて

リソースおよび依存関係モデルは，QuickTest と Quality Center との強力な統合を提供します。

- 添付ファイルの使用が，リンクされた QuickTest アセットに置き換わります。コンポーネントはビジネス・コンポーネント・モジュールに，リソース・ファイルはテスト・リソース・モジュールに格納します。リソース・ファイルをコンポーネントのアプリケーション領域に関連付けると，これらのアセットがリンクされます。アセットをリンクすることでダウンロード時間が短縮され，実行時のパフォーマンスが向上します。また，依存アセット間の関係を確実に維持します（Quality Center 10.00 より，添付ファイルを使用するとダウンロード時間が増加します）。
- コンポーネント，アプリケーション領域，およびリソース・ファイルのバージョン管理とベースラインをサポートします。QuickTest または Quality Center でこれらのアセットのバージョンを作成できます。Quality Center でアセットのバージョンとベースラインを管理します。詳細については，967 ページ「バージョン・コントロールを使ったアセット管理」を参照してください。
- Quality Center と QuickTest の両方の QuickTest アセットの表示と比較ができます。アセット比較ツールを使用して，個々の QuickTest アセットのバージョンを比較できます。また，アセット・ビューアを使用して以前のバージョンの QuickTest アセットを表示できます。これらの両方のビューアは Quality Center および QuickTest で使用できます。詳細については，949 ページ「QuickTest アセットのバージョンの表示と比較」を参照してください。
- Quality Center プロジェクト間でアセットのインポートや共有ができます。詳細については，『**Mercury Quality Center ユーザーズ・ガイド**』を参照してください。

このモデルを使用する利点の詳細については，941 ページ「アセットの依存関係を使用する利点」を参照してください。

Quality Center で相対パスを使用する場合の注意事項

アプリケーション領域に相対パスを使用して関連付けられているリソース・ファイルは，依存関係とみなされません。リソース・ファイルが確実に依存関係として認識されるようにするためには，リソース・ファイルを Quality Center のテスト・リソース・モジュールに保存し，Quality Center の完全パスを使用して関連付ける必要があります。これにより，941 ページ「アセットの依存関係を使用する利点」で説明するように，リソースおよび依存関係モデルが提供する機能を利用できます。

それでも，場合によっては相対パスを使用したいかもしれません。たとえば，アプリケーションがさまざまな言語でリリースされる場合，共有オブジェクト・リポジトリをコンポーネントのアプリケーション領域に関連付ける際に，相対パスを使用する可能性があります。これによりローカライズされた共有オブジェクト・リポジトリで同じテストを使用できます。同様に，さまざまなバージョンのアプリケーションが，それぞれバージョン固有の共有オブジェクト・リポジトリを使用している場合，そのアプリケーションをテストするために同じテストを使用する可能性もあります。

アセットが相対パス経由で関連付けられている場合，次のガイドラインを考慮します。

- 実行時のパフォーマンス時間は遅くなります。
- アセットの依存関係情報が次に表示されません。
  - Quality Center の［依存関係］タブの［使用］グリッドおよび［使用者］グリッド（944 ページ「［依存関係］タブ」を参照してください）。
  - ほかのアセットに関連付けられているアセットを削除する場合に開くメッセージ・ボックス（941 ページ「アセットの依存関係を使用する利点」を参照してください）。
- Quality Center は，ベースライン検証プロセス中にこれらのアセットが含まれることを検証しません（978 ページ「ベースラインの履歴の表示」を参照してください）。
- ベースラインからコンポーネントを開くか実行するときに，関連付けられているリソース・ファイルが相対パス経由で関連付けられているがベースラインに**含まれていない**場合，それらは欠落リソースとみなされます。これにより，コンポーネントの実行が失敗する可能性があります （アセットが相対パス経由で関連付けられていてベースラインに**含まれている**場合，関連付けられているア

セットのベースラインのバージョンが使用されます。978 ページ「ベースラインの履歴の表示」を参照してください)。

- アセット比較ツールを使用してコンポーネントを表示する場合，相対パス経由で関連付けられたアセットをドリルダウンして表示できません（950 ページ「アセット比較ツールおよびアセット・ビューアを使った作業」を参照してください)。

アセットの依存関係を使用する利点

依存リソース・ファイルをコンポーネントのアプリケーション領域に関連付けると，アセットは完全にリンクされます。これらのリンクは Quality Center（さまざまなモジュールの［依存関係］タブ）で表示できます。

- **アセットのリンクが保たれます。**コンポーネントまたはアプリケーション領域を移動するか名前を変更すると，依存アセットは自動的に更新され，変更が反映されます。これにより，実行セッション中に確実に欠落リソースがないようにできます。
- **すべてのリソース・ファイルが 1 つの Quality Center モジュールに格納されます。**リソース・ファイルがテスト・リソース・モジュールに格納されます。これにより，リソースを 1 つの場所で集中管理したり，各リソース・ファイルで使用しているアプリケーション領域をひと目で確認したりできます。テスト・リソース・モジュールの詳細については，『**HP Quality Center ユーザーズ・ガイド**』を参照してください。
- **テスト・リソース・モジュールに格納されているリソースを使用すると，実行時パフォーマンスが向上します。**（テスト計画モジュールにテストへの添付ファイルとして格納される代わりに）関連付けられたリソース・ファイルがテスト・リソース・モジュールに格納されている場合，コンポーネントの開始や実行が早くなります。
- **リソース・ファイルにもバージョン・コントロールが適用されます。**プロジェクトでバージョン・コントロールが有効な場合，バージョン・コントロール・データベースにすべてのアセットをチェックインできます。また各アセットの開発段階をキャプチャするベースラインを作成できます。詳細については，967 ページ「バージョン・コントロールを使ったアセット管理」を参照してください。

➤ **必要に応じて，アセットをほかのプロジェクトと共有したり同期化したりできます。**ほかのプロジェクトからアセットをコピーできます。これにより，新しいプロジェクトを作成するたびに新しいアセットを作成する代わりに，既存のアセットを再利用できます。たとえば，「一時的な」アセットのセットを作成して，新しいプロジェクトの基盤として使用できます。

変更が加えられた場合，両方のプロジェクトのアセットを同期化できます。あるいは各開発プロジェクトの固有のニーズに合わせてアセットをカスタマイズできます。詳細については，『**HP Quality Center ユーザーズ・ガイド**』のライブラリのインポートと同期化に関する項を参照してください。

➤ **アセットの削除が容易になります。**アセット（コンポーネントまたは関連付けられたリソース・ファイル）を削除すると，アセットがアプリケーション領域に関連付けられている場合，警告メッセージが表示されます。メッセージには，必要に応じてアプリケーション領域を変更できるように，このアセットに関連付けられているアプリケーション領域を一覧表示する［**詳細**］セクションが含まれています。これにより，ビジネス・プロセス・テストを管理し，不注意でテストが失敗しないようにできます。

Quality Center でのリソースおよび依存関係モデルの使用

Quality Center サーバで Quality Center プロジェクトを作成する場合，このプロジェクト内で作成する QuickTest コンポーネントはビジネス・コンポーネント・モジュールに保存されます。コンポーネントに関連付けられたリソース・ファイルは，リンクされた依存関係としてテスト・リソース・モジュールに保存されます。

モジュールの は，本項では，QuickTest コンポーネントおよびアプリケーション領域に関係するタブの概要を説明します。これらのタブの使用方法の詳細については，『**Mercury Quality Center ユーザーズ・ガイド**』の関連する項を参照してください。

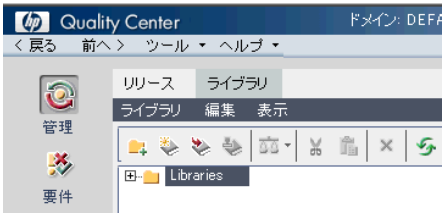
［ライブラリ］タブ

［ライブラリ］タブでは，次の作業が可能です。

➤ **ベースラインの作成，表示，比較。**詳細については，978 ページ「ベースラインの履歴の表示」および『**Mercury Quality Center ユーザーズ・ガイド**』でベースラインを説明している項を参照してください。

➤ **ほかの Quality Center プロジェクトからのアセットのインポート。**これにより，任意のアクセス可能なドメイン内の別のプロジェクトのベースラインに含まれているアセットの完全なコピーを作成できます。詳細については，『**Mercury Quality Center ユーザーズ・ガイド**』を参照してください。

［ライブラリ］タブは，管理モジュールの［リリース］タブの右側にあります。



［履歴］タブ

［履歴］タブには，選択したファイルのバージョンとベースラインの情報が一覧表示されます。ファイルの各バージョンを表示，比較したり，バージョンが格納されているベースラインを確認したりできます（存在する場合）。また，以前のバージョンをロール・バックする場合，そのバージョンをチェックアウトできます。ファイルをバージョン・コントロール・データベースに再度チェックインすると，そのバージョンが最新のバージョンになります。

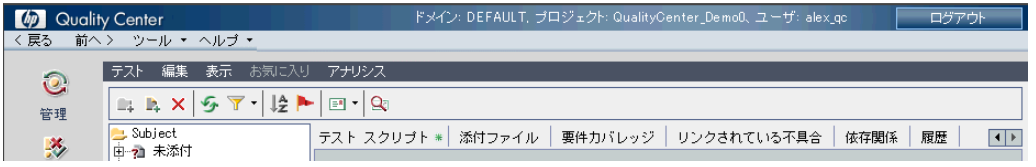
［履歴］タブは，次のモジュールで使用できます。

➤ ビジネス・コンポーネント・モジュール

➤ テスト計画モジュール

➤ テスト・リソース・モジュール

［履歴］タブは，ウィンドウの右側の表示枠内にあります。右にスクロールすると表示されます。



［表示］タブの詳細については，『**Mercury Quality Center ユーザーズ・ガイド**』を参照してください。

ヒント：QuickTest でバージョン履歴およびベースライン履歴を表示するには，［**ファイル**］＞［**Quality Center バージョン コントロール**］＞［**バージョンの履歴**］，または［**ファイル**］＞［**Quality Center バージョン コントロール**］＞［**ベースラインの履歴**］を選択します。詳細については，976 ページ「アセットのバージョンの履歴の表示」および 978 ページ「ベースラインの履歴の表示」を参照してください。

［依存関係］タブ

［依存関係］タブには，コンポーネントなどの選択したアセットと，そのアセットに関連付けられたアセット間の関係が表示されます。［依存関係］タブを使用して，特定のアセットに使用されるリソースや，特定のリソースを使用しているアセットをひと目で確認します。

たとえば，共有オブジェクト・リポジトリのオブジェクトを変更するとします。テスト・リソース・モジュール内の共有オブジェクト・リポジトリに移動して，オブジェクトが関連付けられているアプリケーション領域の一覧を表示できます。これにより，このリソース・ファイルを使用しているアセットを特定したり，予定されている変更が依存アセットに与える可能性がある影響を分析したりできます。

Quality Center では，ビジネス・コンポーネント・モジュールおよびテスト・リソースモジュールの［依存関係］タブで，［**使用**］および［**使用者**］の情報を表示できます。

［依存関係］タブは，次のモジュールで使用できます。

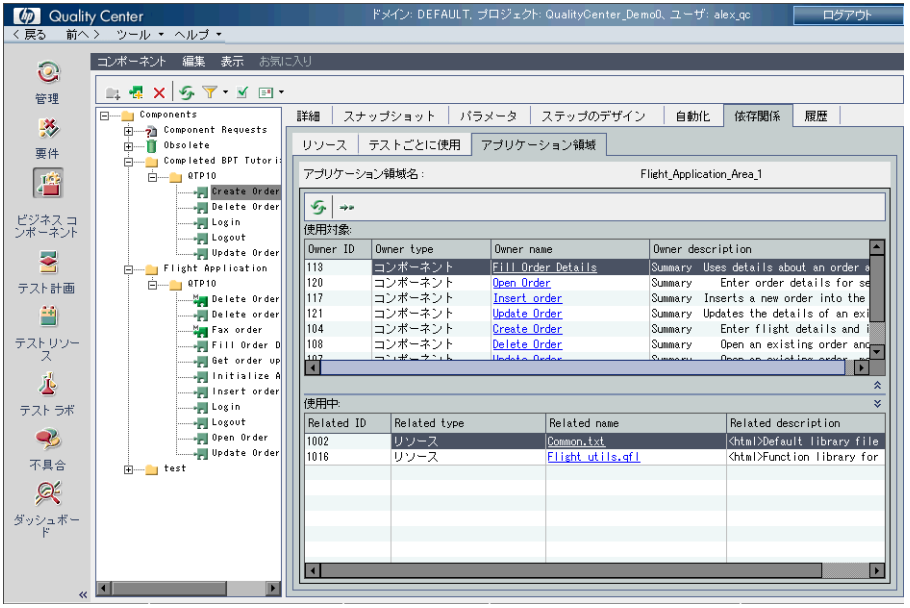
- ビジネス・コンポーネント・モジュール
- テスト計画モジュール
- テスト・リソース・モジュール

注：ビジネス・コンポーネント・モジュールでは，［依存関係］タブは次のサブタブに分かれています。

- ［**リソース**］：［使用］グリッドで，コンポーネントが関連付けられているアプリケーション領域を表示します。
- ［**使用中**］：このコンポーネントを含むビジネス・プロセス・テストを表示します。
- ［**アプリケーション領域**］：アプリケーション領域の名前を一覧表示し，［使用者］グリッドにこのアプリケーション領域に関連付けられているコンポーネントを，［使用］グリッドに，このアプリケーション領域に関連付けられているリソース・ファイルを表示します。

ビジネス・コンポーネント・モジュールの［依存関係］タブの詳細については，『**HP Business Process Testing ユーザーズ・ガイド**』を参照してください。

Quality Center の［依存関係］タブの例を次に示します。



［依存関係］タブには，［**使用対象**］グリッドと［**使用中**］グリッドがあります。［使用対象］グリッドには，選択したアセットに依存しているアセットが表示されます。［使用中］グリッドには，選択したアセットが依存するアセットが表示されます。

［使用対象］グリッド

［使用対象］グリッドには，選択したアセットに依存しているアセット，つまり選択したアセットを使用しているアセットが一覧表示されます。たとえば，共有オブジェクト・リポジトリの［使用対象］グリッドを見ているとします。［使用対象］グリッドには，この依存関係に関連付けられているすべてのアプリケーション領域が一覧表示されます。

［使用対象］グリッドには，次のカラムがあります。

カラム	説明
［**所有者 ID**］	Quality Center によって自動的に割り当てられる一意の数値 ID。［**所有者 ID**］がリンクの場合，クリックして Quality Center 内の該当のアセットに移動できます。 **例**：テスト・リソース・モジュール内の特定の関数ライブラリの［使用者］グリッドを見ているとします。［**所有者 ID**］リンクをクリックして，関連付けられているアプリケーション領域に移動できます （リンクからビジネス・コンポーネント・モジュールに移動します)。
［**所有者タイプ**］	選択したアセットを使用しているアセットのタイプ。QuickTest 関連の所有者タイプには次のものがあります。 ➤ **リソース**：テスト・リソース・モジュールに一覧表示されるリソース。 ➤ **コンポーネント（テスト・リソース・モジュール）**：ビジネス・コンポーネント・モジュールの［**依存関係**］タブ＞［**アプリケーション領域**］サブタブに一覧表示されるアプリケーション領域。 ➤ **コンポーネント（ビジネス・コンポーネント・モジュール**＞［**依存関係**］タブ＞［**アプリケーション領域**］サブタブ）：ビジネス・コンポーネント・モジュールに一覧表示されるコンポーネント。

カラム	説明
［**所有者名**］	選択したアセットを使用しているアセットの名前。［**所有者名**］がリンクの場合，クリックして Quality Center 内の該当のアセットに移動できます。
［**所有者詳細**］	選択されたアセットを使用する関連付けられているアセットの詳細。 ➤［**所有者タイプ**］が**コンポーネント**で，所有者がアプリケーション領域の場合，**アプリケーション領域の詳細**が表示されます。 ➤［**所有者タイプ**］が**コンポーネント**で，所有者がコンポーネントの場合，ビジネス・コンポーネント・モジュールの［詳細］タブの［**詳細**］領域から選択したコンポーネントの情報が表示されます。

使用グリッド

［使用］グリッドには，選択したアセットが使用しているすべての依存関係が一覧表示されます。たとえば，コンポーネントを見ているとします。コンポーネントで使用されるテスト・オブジェクトを含むすべての共有オブジェクト・リポジトリ，コンポーネントで呼び出される関数を含む関数ライブラリなどを確認できます。

使用グリッドにあるカラムは，次のとおりです。

カラム	説明
関連 ＩＤ	Quality Center によって自動的に割り当てられる一意の数値 ID。
関連タイプ	選択されたアセットが使用する関連付けられているアセットのタイプ。QuickTest 関連のタイプには次のものがあります。 ➤ **リソース**：テスト・リソース・モジュールに一覧表示されるリソース。 ➤ **コンポーネント（テスト・リソース・モジュール）**：ビジネス・コンポーネント・モジュールの［**依存関係**］タブ＞［**アプリケーション領域**］サブタブに一覧表示されるアプリケーション領域。

カラム	説明
関連名	選択されたアセットが使用する関連付けられているアセットの名前。
関連詳細	選択されたアセットが使用する関連付けられているアセットの詳細。 QuickTest 関連の詳細には次のものがあります。 ➤［**関連タイプ**］が**コンポーネント**の場合，**アプリケーション領域の詳細**が表示されます。 ➤［**関連タイプ**］が**リソース**の場合，たとえば**関数ライブラリ**などのリソース・タイプが表示されます。また，ビジネス・コンポーネント・モジュールの［詳細］タブの［**詳細**］領域から，選択したリソースの情報が表示されます。

第 41 章

QuickTest アセットのバージョンの表示と比較

本章では，アセット比較ツールを使用して Quality Center に格納されている QuickTest アセットのバージョンを比較する方法について説明します。**アセット**とは，QuickTest テスト・ドキュメント，または QuickTest テスト・ドキュメントによって使用されるリソース・ファイルです。

注：本章に含まれる Quality Center の説明は，Quality Center 10.00 に適用されます。ただし，機能やオプションがお使いの Quality Center のエディションでサポートされていない可能性があります。Quality Center のエディションの詳細については，『**Mercury Quality Center ユーザーズ・ガイド**』を参照してください。

本章の内容

➤ アセット比較ツールおよびアセット・ビューアを使った作業（950 ページ）

➤ QuickTest アセット比較ツール（952 ページ）

➤ QuickTest アセット・ビューア（962 ページ）

ヒント：2 つの異なるオブジェクト・リポジトリを比較するには，オブジェクト・リポジトリ比較ツールを使用します（301 ページを参照してください）。

アセット比較ツールおよびアセット・ビューアを使った作業

本項では，アセット比較ツールおよびアセット・ビューアを使用して最もよく実行されるタスクについて説明します。

アセットの 2 つのバージョンの比較の表示（アセット比較ツール）

アセットの 2 つのバージョンの比較を横に並べてまたは縦に重ねて表示できます。

ドリルダウンして特定の要素を比較または表示

構成要素のバージョンの比較。現在開いているバージョン比較で，特定の要素をドリルダウンして比較を表示できます。要素には，ローカル・オブジェクト・リポジトリなど，テストを構成する一部である（外部リソースとして保存されていない）リソースが含まれます。コンポーネントをチェックインすると，これらの要素もチェックインされます。これにより，これらの要素のバージョン比較をテストで直接表示できます。

関連付けられているリソース・ファイルの最新の内容の表示。関連付けられているリソース・ファイルとは，アセットによって使用されるリソース・ファイルです。たとえば，コンポーネントで使用できるリソース・ファイルの例は，関数ライブラリや共有オブジェクト・リポジトリです。コンポーネントでドリルダウンを行うと，最後に保存されたバージョンのリソース・ファイルを表示できます。これにより最新の内容を表示できます （ドリルダウンしたリソースの別のバージョンを比較する場合は，リソースを開いて新しい比較を実行します)。

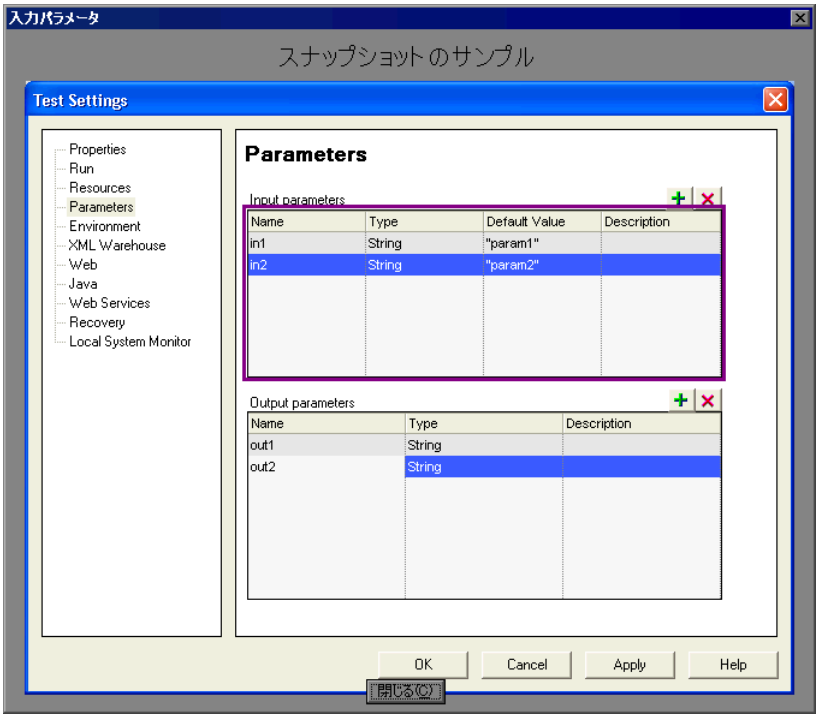
ドリルダウンを行うには，次のいずれかを行います。

- 比較できる任意のアセットの横の青いドリルダウン矢印をクリックします（ドリルダウン矢印の近くでポインタが指さし型に変わります)。
- 要素をダブルクリックします。
- 要素を右クリックし，［**ドリルダウンの表示**］を選択します。詳細については，960 ページ「QuickTest アセット比較ツール：ショートカット・メニュー・コマンド」を参照してください。

QuickTest での要素の位置の表示

該当のノードを右クリックし，[**スナップショットのサンプルの表示**] をクリックして，QuickTest での要素の位置を示す画面キャプチャを表示できます。画面キャプチャには，関連するダイアログ・ボックスの例が表示されます。右クリックしたノードのオプション（または領域）が画面キャプチャ内で強調表示されます。

たとえば，コンポーネントの比較を表示し，[**入力パラメータ**] ノードが強調表示され，変更されたことを示していることに気付いたとします。このオプションが QuickTest のどの位置にあるかわからない場合，比較ツリーでノードを右クリックし，[**スナップショットのサンプルの表示**] を選択します。QuickTest によりダイアログ・ボックスが開き，この領域が [コンポーネントの設定] ダイアログ・ボックスの [パラメータ] 表示枠にあることが示されます。



詳細については，960 ページ「QuickTest アセット比較ツール：ショートカット・メニュー・コマンド」を参照してください。

テキストと背景の色の変更

［色設定］ダイアログ・ボックスを使用して，［アセット比較ツール］ウィンドウのフィルタ・タイプ（変更済み，追加済み，削除済みなど）のテキストと背景の色を変更できます。

フィルタ・タイプの背景色を変更すると，ウィンドウの最上部にある凡例のフィルタ・タイプの色もそれに応じて変更されます。この変更は，再度変更するか標準設定に戻すまで有効です。

詳細については，961 ページ「［色設定］ダイアログ・ボックス」を参照してください。

特定の要素の相違点の数の表示

要素のサブ要素がバージョン間で異なり，その要素を表すノードを折りたたんだ場合，ノードの横に凡例が表示されます。凡例には，折りたたまれた要素に存在する相違点の数が示されます。次の例では，3 つのサブ要素が変更され，1 つのサブ要素が削除され，7 つのサブ要素が追加されています。

詳細については，952 ページ「QuickTest アセット比較ツール」と 962 ページ「QuickTest アセット・ビューア」を参照してください。

QuickTest アセット比較ツール

QuickTest アセット比較ツールにより，アプリケーション領域，コンポーネント，関数ライブラリ，共有オブジェクト・リポジトリ，または回復シナリオなど，特定の QuickTest アセットの 2 つのバージョンを比較できます。また，アセットをドリルダウンして，関連共有オブジェクト・リポジトリなど，アセットに関連付けられているエンティティの比較も表示できます。

注：QuickTest アセット比較ツールでは，相対パス経由で関連付けられたアセットをドリルダウンして表示できません。詳細については，940 ページ「Quality Center で相対パスを使用する場合の注意事項」を参照してください。

QuickTest アセット比較ツールを開く

QuickTest アセット比較ツールは，バージョン・コントロールが有効な場合は QuickTest または Quality Center から開くことができます。

アセット比較ツールは，次の場所から開くことができます。

QuickTest メイン・ウィンドウ

1 バージョンを比較するコンポーネントまたは関数ライブラリを開きます。

2 ［**ファイル**］＞［**Quality Center バージョン コントロール**］＞［**バージョンの履歴**］，または［**ファイル**］＞［**Quality Center バージョン コントロール**］＞［**ベースラインの履歴**］を選択します。［バージョンの履歴］または［ベースラインの履歴］ダイアログ・ボックスが開きます。

3 2 つのバージョンを選択して，［**比較**］をクリックします。アセット比較ツールが開きます。

オブジェクト・リポジトリ・マネージャ

1 オブジェクト・リポジトリ・マネージャを開きます（［**リソース**］＞［**オブジェクト リポジトリ マネージャ**］）。

2 比較するバージョンの共有オブジェクト・リポジトリを見つけて開きます。詳細については，234 ページ「オブジェクト・リポジトリを開く」を参照してください。

3 ［**ファイル**］＞［**Quality Center バージョン コントロール**］＞［**バージョンの履歴**］，または［**ファイル**］＞［**Quality Center バージョン コントロール**］＞［**ベースラインの履歴**］を選択します。［バージョンの履歴］または［ベースラインの履歴］ダイアログ・ボックスが開きます。

4 2 つのバージョンを選択して，［**比較**］をクリックします。アセット比較ツールが開きます。

回復シナリオ・マネージャ

1 回復シナリオ・マネージャを開きます（［**リソース**］＞［**回復シナリオ マネージャ**］）。

2 バージョンを比較する回復シナリオ・ファイルを開きます。詳細については，872 ページ「[回復シナリオ マネージャ］ダイアログ・ボックスについて」を参照してください。

3 [**バージョン コントロール**] 下向き矢印をクリックし，[**バージョンの履歴**] または [**ベースラインの履歴**] を選択します。

4 2 つのバージョンを選択して，[**比較**] をクリックします。アセット比較ツールが開きます。

Quality Center

1 Quality Center で，比較するアセットを含むプロジェクトに接続します。

2 次のいずれかを行います。

- サイドバーの [**ビジネス コンポーネント**] ボタンをクリックし，ビジネス・コンポーネント・モジュールを開きます。
- サイドバーの [**テスト リソース**] ボタンをクリックし，**テスト・リソース・**モジュールを開きます。このモジュールには，関数ライブラリ，共有オブジェクト・リポジトリ，回復シナリオなど，コンポーネントに関連付けられているリソース・ファイルが含まれています。

3 ツリーでバージョンを比較するファイルを選択します。

4 [**履歴**] タブをクリックし，[**バージョンおよびベースライン**] タブをクリックします。

5 [**右記による表示**] ボックスで，[**バージョン**] または [**ベースライン**] を選択します。

6 グリッドで，比較する 2 つのバージョンを選択し，[**比較**] ボタンをクリックします。

7 開いたウィンドウのサイドバーで，[**QTP Comparison**] または [**オートメーション**] ボタンをクリックします。アセット比較ツールが開きます。

ヒント：管理モジュールのベースラインを比較することもできます。**管理**モジュールを開くには，サイドバーの [**管理**] ボタンをクリックします。ツリーでベースラインを選択し，[**次と比較**] ボタンをクリックします。詳細については，『**Mercury Quality Center ユーザーズ・ガイド**』を参照してください。ベースラインの詳細については，968 ページ「Quality Center でのアセットのバージョン管理」を参照してください。

アセット比較ツールのコマンドとオプションについて

本項では，アセット比較ツールおよびアセット・ビューアで使用できるコマンドとオプションについて説明します。これらのツールの概要については，950 ページ「アセット比較ツールおよびアセット・ビューアを使った作業」を参照してください。

アセット比較ツールでは，アセットの 2 つのバージョンを比較できます。詳細については，952 ページ「QuickTest アセット比較ツール」を参照してください。

アセット・ビューアでは，アセットの個々の項目を表示できます。詳細については，962 ページ「QuickTest アセット・ビューア」を参照してください。

次は QuickTest アセット比較ツールの画像です。

QuickTest アセット比較ツール：メニュー，ツールバー，ボタンのオプション

	コマンド	ショートカット・キー	説明
	［ファイル］＞［終了］	ALT+F4	［アセット比較ツール］ウィンドウを閉じます。
	［表示］＞［次の差異］	CTRL+ 下向き矢印	比較対象バージョン内の要素の次の差異を検索します。

	コマンド	ショートカット・キー	説明
	［表示］＞［前の差異］	CTRL+ 上向き矢印	比較対象バージョン内の要素の前の差異を検索します。
	［表示］＞［更新］		選択したアセットのバージョンを新たに比較します。 **注**：これはアセットの現在のバージョンを比較している場合に役立ちます。アセットを変更して保存した場合，［**更新**］コマンドを使用して，更新された比較を表示できます。
	［ツール］＞［色設定］		［色設定］ダイアログ・ボックスを開き，各フィルタ・タイプのテキストと背景の色を定義できます。 961 ページ「［色設定］ダイアログ・ボックス」を参照してください。
	［ツール］＞［フィルタ］		比較ウィンドウで，次のフィルタ要素のタイプを表示または非表示にできます。 ➤ **変更済み** ➤ **削除済み** ➤ **追加済み** ➤ **同じ** フィルタ・コマンドを選択またはクリアします。比較ウィンドウには，定義されたフィルタに一致する要素のみが表示されます。 **ヒント**：ウィンドウの右上角の凡例は，各フィルタ・タイプに一致する要素の数を示します。折りたたまれたノードの横の凡例は，各フィルタ・タイプに一致するサブノードの数を示します。

	コマンド	ショートカット・キー	説明
	［ウィンドウ］＞［ウィンドウを閉じる］		現在アクティブな比較ウィンドウが比較のメイン・ウィンドウから開かれている場合，閉じます。複数の比較ウィンドウが開かれている場合にのみ有効です。 **注**：別のウィンドウを開いて，共有オブジェクト・リポジトリなど，現在比較対象のアセットに関連付けられているアセットの比較を表示できます。この処理を行うには，比較できる任意のアセットの横の青いドリルダウン矢印をクリックします **ヒント**：比較ウィンドウは，ウィンドウの最下部のタブにある **X** をクリックして閉じることもできます。
	［ウィンドウ］＞［水平に表示］または**［ウィンドウ］＞［垂直に表示］**		［**水平に表示**］：開いているドキュメントを縦に重ねて表示します。 ［**垂直に表示**］：開いているドキュメントを横に並べて表示します。
	［ウィンドウ］＞＜比較対象アセットのパス＞		開いている比較ウィンドウ間を移動できます。
	［ヘルプ］＞［アセット比較ツール ヘルプ］	F1	［アセット比較ツール ヘルプ］を開きます。

	コマンド	ショートカット・キー	説明
	［**前の 2000 行**］ボタン		テスト・ドキュメントの行数が 2000 行を超える場合，比較表示枠の最上部にこのボタンが表示されます。クリックするとテスト・ドキュメントの現在の 2000 行が非表示になり，前の 2000 行が表示されます。
	［**次の 2000 行**］ボタン		テスト・ドキュメントの行数が 2000 行を超える場合，比較表示枠の最下部にこのボタンが表示されます。クリックするとテスト・ドキュメントの現在の 2000 行が非表示になり，次の 2000 行が表示されます。

QuickTest アセット比較ツール：凡例

次の例は，［アセット比較ツール］ウィンドウの右上角に表示されるフィルタの凡例です。

記号	説明	数値
	変更済み	比較対象内で変更された要素の数が示されます。
	削除済み	比較対象のいずれかのバージョンから削除された要素の総数を示します。
	追加済み	比較対象のいずれかのバージョンに追加された要素の総数を示します。

注：

- フィルタ・タイプの背景色を（[色設定] ダイアログ・ボックスを使用して）変更すると，凡例のフィルタ・タイプの色もそれに応じて変更されます。
- 比較ウィンドウでアセットを折りたたむと，次の例のようにアセットの凡例が表示されます。

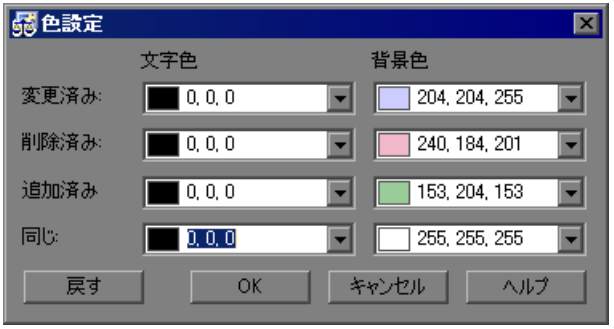
QuickTest アセット比較ツール：ショートカット・メニュー・コマンド

コマンド	ショートカット・キー	説明
[ドリルダウンの表示]	ENTER	新しいウィンドウに，選択したアセットのバージョン比較をドリルダウンして表示します（比較できるアセットにのみ該当します）。 **ヒント**：ノードの横の青いドリルダウン矢印をクリックして，新しいウィンドウにバージョン比較をドリルダウンして表示することもできます。 **注**：相対パス経由で関連付けられているアセットをドリルダウンして表示することはできません。940 ページ「Quality Center で相対パスを使用する場合の注意事項」を参照してください。
[スナップショットのサンプルの表示]	CTRL+Q	[ビジネス コンポーネントの設定] ダイアログ・ボックスの [リソース] 表示枠など，QuickTest で選択した要素のサンプル画像を含むウィンドウを開きます。スナップショット内で要素自体が強調表示されます。

［色設定］ダイアログ・ボックス

説明	［アセット比較ツール］ウィンドウ内のさまざまなフィルタ要素のテキストと背景の色を変更できます。変更は，後続のすべてのセッションで有効になります。 **注**：フィルタ・タイプの背景色を変更すると，［アセット比較ツール］ウィンドウの右上角にある凡例もそれに応じて変更されます。
アクセス方法	［アセット比較ツール］ウィンドウで，次のようにアクセスします。 ➤［**ツール**］＞［**色設定**］メニュー・コマンドを選択します。 ➤［**色設定**］ツールバー・ボタン をクリックします。

次は［色設定］ダイアログ・ボックスの画像です。



［色設定］ダイアログ・ボックスのオプション

オプション	説明
［追加済み］ **［削除済み］** **［変更済み］** **［同じ］**	該当するフィルタ要素のテキスト色と背景色を選択します。次のことができます。 ➤ 下向き矢印をクリックし，［Custom］，［Web］，または［System］タブの色の一覧から色を選択します。 ➤ RGB 値をエディット・ボックスに直接入力します。
［戻す］	クリックしてフィルタ要素ごとに標準設定の色の値に戻します。

QuickTest アセット・ビューア

QuickTest アセット・ビューアにより，アプリケーション領域，コンポーネント，関数ライブラリ，共有オブジェクト・リポジトリ，または回復シナリオなど，特定の QuickTest アセットの以前のバージョンを表示できます。また，［アセット ビューア］ウィンドウでドリルダウンして，関連共有オブジェクト・リポジトリなど，関連付けられたエンティティを表示できます。

QuickTest アセット・ビューアを開く

QuickTest アセット・ビューアは，バージョン・コントロールが有効な場合，QuickTest または Quality Center から開くことができます。

アセット・ビューアは，次の場所から開くことができます。

QuickTest メイン・ウィンドウ

1 以前のバージョンを表示するコンポーネントまたは関数ライブラリを開きます。

2 ［**ファイル**］＞［**Quality Center バージョン コントロール**］＞［**バージョンの履歴**］を選択します。［バージョンの履歴］ダイアログ・ボックスが表示されます。

3 バージョンを選択して，［**表示**］をクリックします。アセット・ビューアが開きます。

オブジェクト・リポジトリ・マネージャ

1 オブジェクト・リポジトリ・マネージャを開きます（［**リソース**］＞［**オブジェクト リポジトリ マネージャ**］）。

2 以前のバージョンを表示する共有オブジェクト・リポジトリを見つけて開きます。詳細については，234 ページ「オブジェクト・リポジトリを開く」を参照してください。

3 ［**ファイル**］＞［**Quality Center バージョン コントロール**］＞［**バージョンの履歴**］を選択します。［バージョンの履歴］ダイアログ・ボックスが表示されます。

4 バージョンを選択して，［**表示**］をクリックします。アセット・ビューアが開きます。

回復シナリオ・マネージャ

1 回復シナリオ・マネージャを開きます（[**リソース**] > [**回復シナリオ マネージャ**]）。

2 以前のバージョンを表示する回復シナリオ・ファイルを開きます。詳細については，872 ページ「[回復シナリオ マネージャ] ダイアログ・ボックスについて」を参照してください。

3 [**バージョン コントロール**] 下向き矢印をクリックし，[**バージョンの履歴**] を選択します。

4 バージョンを選択して，[**表示**] をクリックします。アセット・ビューアが開きます。

Quality Center

1 Quality Center で，表示するアセットを含むプロジェクトに接続します。

2 次のいずれかを行います。

- サイドバーの [**ビジネス コンポーネント**] ボタンをクリックし，ビジネス・コンポーネント・モジュールを開きます。
- [**テスト リソース**] ボタンをクリックし，**テスト・リソース**・モジュールを開きます。このモジュールには，関数ライブラリ，共有オブジェクト・リポジトリ，回復シナリオなど，コンポーネントに関連付けられているリソース・ファイルが含まれています。

3 ツリーで，以前のバージョンを表示するファイルを選択します。

4 [**履歴**] タブをクリックし，[**バージョンおよびベースライン**] タブをクリックします。

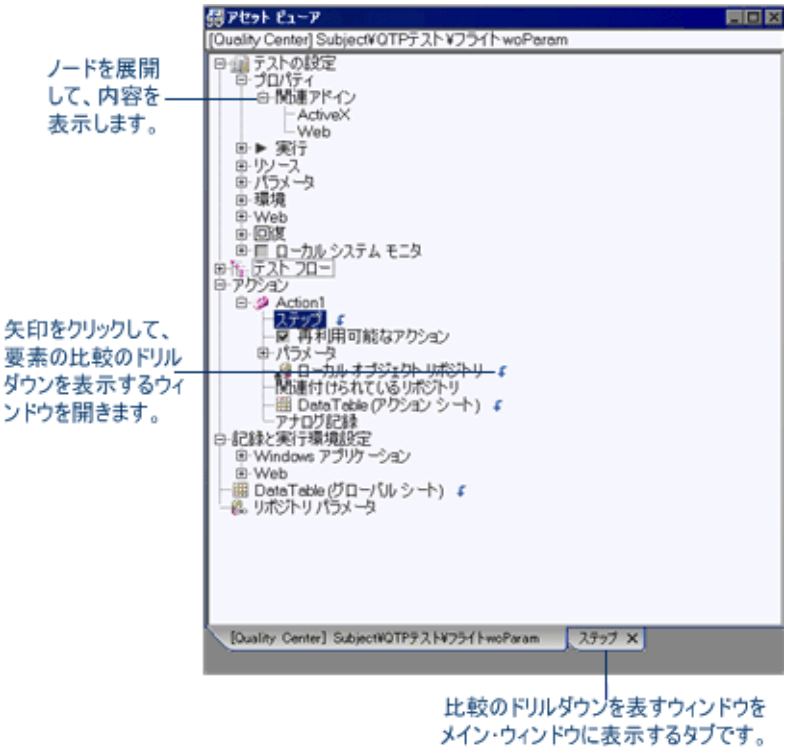
5 [**表示方法**] ボックスで，[**バージョン**] を選択します。

6 グリッドでバージョンを選択し，[**表示**] ボタンをクリックします（現在チェックアウトされているバージョンは表示できません）。ウィンドウが開きます。サイドバーあるボタンを使用して，選択したアセットのバージョン固有の情報にアクセスできます（これらのボタンは，選択したアセットの最新のバージョンのメイン・ウィンドウで右側の表示枠に表示されるタブと同じです）。詳細については，『**Mercury Quality Center ユーザーズ・ガイド**』を参照してください。

QuickTest アセット・ビューアの使用

アセット・ビューアは，アセットの機能的な概要を表示し，ビューア形式でアセットの設定を表示できます。ツリー・ビューにより，ドリルダウンして，別のダイアログ・ボックスまたは QuickTest を開く必要なしに特定の設定を表示または検証できます。

次は QuickTest アセット・ビューアの画像です。



QuickTest アセット・ビューア：ボタン・オプション

コマンド	ショートカット・キー	説明
［前の 2000 行］		テスト・ドキュメントの行数が 2000 行を超える場合，表示枠の最上部にこのボタンが表示されます。クリックするとテスト・ドキュメントの現在の 2000 行が非表示になり，前の 2000 行が表示されます。
［次の 2000 行］		テスト・ドキュメントの行数が 2000 行を超える場合，表示枠の最下部にこのボタンが表示されます。クリックするとテスト・ドキュメントの現在の 2000 行が非表示になり，次の 2000 行が表示されます。

QuickTest アセット・ビューア：ショートカット・メニュー・コマンド

コマンド	ショートカット・キー	説明
［ドリルダウンの表示］	ENTER	新しいウィンドウに，選択したアセットのバージョン比較をドリルダウンして表示します（比較できるアセットにのみ該当します）。 **ヒント**：ノードの横の青いドリルダウン矢印をクリックして，新しいウィンドウにバージョン比較をドリルダウンして表示することもできます。 **注**：相対パス経由で関連付けられているアセットをドリルダウンして表示することはできません。940 ページ「Quality Center で相対パスを使用する場合の注意事項」を参照してください。
［スナップショットのサンプルの表示］	CTRL+Q	［ビジネス コンポーネントの設定］ダイアログ・ボックスの［リソース］表示枠など，QuickTest で選択した要素のサンプル画像を含むウィンドウを開きます。スナップショット内で要素自体が強調表示されます。

第 42 章

バージョン・コントロールを使ったアセット管理

本章では，バージョン・コントロールを使用して Quality Center に格納されている QuickTest アセットを管理，使用する方法について説明します。

注：本章に含まれる Quality Center の説明は，Quality Center 10.00 に適用されます。ただし，機能やオプションがお使いの Quality Center のエディションでサポートされていない可能性があります。Quality Center のエディションの詳細については，『**Mercury Quality Center ユーザーズ・ガイド**』を参照してください。

本章の内容

- Quality Center でのアセットのバージョン管理（968 ページ）
- アセットのバージョンの履歴の表示（976 ページ）
- ベースラインの履歴の表示（978 ページ）
- バージョンの履歴とベースラインの履歴（982 ページ）

Quality Center でのアセットのバージョン管理

バージョン・コントロールがサポートされている Quality Center プロジェクトに QuickTest を接続すると，以前のバージョンを維持したまま，QuickTest アセットの更新や変更ができます。これにより，各アセットに加えた変更の追跡や，バージョン間の変更の確認が容易になります。アセットには，コンポーネント，関数ライブラリ，共有オブジェクト・リポジトリ，回復シナリオ，および外部データ・テーブルがあります。

バージョン・コントロール・データベースにアセットをチェックイン，チェックアウトすることで，アセットのバージョンが管理されます。バージョン・コントロール・サポート付きの Quality Center プロジェクトにアセットを保存すると，バージョン・コントロール・データベースにアセットが追加されます。アセットを初めて保存したときに，QuickTest によってアセットが Quality Center バージョン・コントロール・データベースに自動的にチェックインされ，バージョン番号 1 が割り当てられます。そして引き続きアセットを使用できるように，アセットがユーザに自動的にチェックアウトされます。このアセットをチェックインすると，これは内容を含めることができる最初のバージョンなので，バージョン番号 1 を保持しています。その後，アセットを再度チェックアウトしてチェックインするたびに，バージョン番号は 1 ずつ増えます。

注：Quality Center でアセットを直接作成すると，アセットにはバージョン番号 1 が割り当てられ，すぐにユーザにチェックアウトされます。Quality Center では，バージョン番号 1 は内容なしで作成されたアセットを表しています。次にこのアセットをチェックインしたときに，バージョン番号 2 が割り当てられます。

アセットはいつでもチェックインできます。たとえば，アセットを毎日チェックインしたり，タスクの完了時にチェックインしたりできます。アセットがユーザにチェックアウトされている間，ほかのユーザはアセットの最後にチェックインされたバージョンを読み取り専用モードで表示できます。しかし，アセットをチェックインするまで，ほかのユーザはアセットを変更したり加えられた変更を表示したりすることはできません。

アセットの状態	可能な作業
チェックイン	➤［**開く**］オプションを使用して，読み取り専用モードでアセットを開きます。アセットの変更はできません。 ➤ アセットを開き，［**開いてチェックアウトする**］オプションを使用してすぐにチェックアウトします。必要に応じてこのアセットを変更できます。
ユーザ自身の Quality Center ユーザ名へチェックアウト	［**開く**］オプションを使用してアセットを開き，必要に応じてアセットを変更します。
ほかの Quality Center ユーザへチェックアウト	［**開く**］オプションを使用して，読み取り専用モードでアセットを開きます。このアセットはほかの Quality Center ユーザがチェックアウトしていることを示すメッセージが表示されます。今はアセットの最後にチェックインされたバージョンを表示します。後でほかのユーザがアセットをチェックインしたら，アセットをチェックアウトできます。

QuickTest では，以前のバージョンのアセットの表示や比較はできますが，チェックアウトできるのは最新バージョンのみです。これは，Quality Center に格納されているアセットは，別のアセットにリンクされたり**依存**していることが多いためです。

たとえば，新しいバージョンの共有オブジェクト・リポジトリを使用して以前のバージョンのコンポーネントを実行しようとした場合，オブジェクト・リポジトリ内のオブジェクトがコンポーネント内のオブジェクトと必ずしも一致するとは限らないため，コンポーネントは失敗する可能性があります。

たとえば以前のバージョンをロール・バックする場合など，以前のバージョンのアセットをチェックアウトする必要があれば，Quality Center プロジェクト管理者に問い合わせてください。管理者は，関連するすべてのアセットの正しいバージョンが最新バージョンになることを確認する必要があります。

アセットのバージョンの表示や比較には，アセット比較ツールを使用します。詳細については，949 ページ「QuickTest アセットのバージョンの表示と比較」を参照してください。

製品開発中に重要なポイントに到達したときにプロジェクト管理者がプロジェクトのベースライン・バージョンを作成した場合，これらのベースラインに格納されているアセット・バージョンの表示や比較ができます。詳細については，978 ページ「ベースラインの履歴の表示」を参照してください。

注 :［**ベースラインの履歴**］オプション以外の［**ファイル**］メニューの［**Quality Center バージョン コントロール**］オプションは，バージョン・コントロールがサポートされている Quality Center プロジェクトに接続していて，Quality Center に格納されているアセットが QuickTest ウィンドウで開いている場合にのみ使用できます。

バージョン管理コマンド

QuickTest では次のバージョン・コントロール・コマンドを使用できます。

- ［**チェックアウト**］: バージョン・コントロールされているアセットをバージョン・コントロール・データベースからチェックアウトできます。詳細については，971 ページ「バージョン・コントロール・データベースからのアセットのチェックアウト」を参照してください。
- ［**チェックアウトを元に戻す**］: バージョン・コントロールされているアセットのバージョン・コントロール・データベースからのチェックアウトを取り消すことができます。詳細については，975 ページ「チェックアウト操作の取り消し」を参照してください。
- ［**チェックイン**］: アセットをバージョン・コントロール・データベースにチェックインできます。詳細については，971 ページ「バージョン・コントロール・データベースからのアセットのチェックアウト」を参照してください。
- ［**バージョンの履歴**］: 特定のアセットのバージョンを表示または比較できます。詳細については，968 ページ「Quality Center でのアセットのバージョン管理」を参照してください。
- ［**ベースラインの履歴**］: プロジェクトのベースラインに保存された特定のアセットのバージョンを表示または比較できます。詳細については，978 ページ「ベースラインの履歴の表示」を参照してください。

バージョン・コントロール・データベースへのアセットの追加

バージョン・コントロールがサポートされている Quality Center プロジェクトに［**名前を付けて保存**］を使って新規のアセットを保存すると，QuickTest によって自動的にアセットがプロジェクトに保存され，バージョン・コントロール・データベースにバージョン番号 1 としてチェックインされ，その後作業を続けられるようにチェックアウトされます。これはアセットの管理上のバージョンで，プレースフォルダとほぼ同じです。バージョン番号はアセットがデータベース内に存在することを示します。後でこのアセットをチェックインしてもバージョン番号は 1 のまま，つまりチェックインしている最初のバージョンということです。以降のチェックインでは，バージョン番号が 1 ずつ増えます。

既存のアセットに変更を保存しても，そのアセットをチェックインすることにはなりません。アセットを保存して閉じても，チェックインを選択するまではアセットはチェックアウトしたままです。詳細については，974 ページ「バージョン・コントロール・データベースへのアセットのチェックイン」を参照してください。

バージョン・コントロール・データベースからのアセットのチェックアウト

現在バージョン・コントロール・データベースにチェックインしているアセットを開くと，アセットは読み取り専用モードで開きます。チェックインしたアセットを確認できます。また，アセットを実行して結果を表示することもできます。

アセットを変更するには，アセットをチェックアウトする必要があります。アセットをチェックアウトすると，アセットは Quality Center によってユーザの一意のチェックアウト・ディレクトリ（初めてアセットをチェックアウトするときに自動的に作成される）にコピーされ，プロジェクト・データベースにロックされます。これにより，アセットに加えた変更を Quality Center プロジェクトのほかのユーザに上書きされないようにします。ただし，この場合でも，ほかのユーザはデータベースにチェックインされているテストの最新バージョンを実行できます。

アセットを保存して閉じることはできますが，Quality Center データベースにアセットを戻すまでアセットはロックされています。アセットを解放するには，アセットをチェックインするかチェックアウト操作を取り消します。アセットのチェックインの詳細については，974 ページ「バージョン・コントロール・データベースへのアセットのチェックイン」を参照してください。チェックアウトの取り消しの詳細については，975 ページ「チェックアウト操作の取り消し」を参照してください。

QuickTest では，チェックアウト・オプションはアセットの最新バージョンにアクセスします。また Quality Center では，アプリケーション領域以外の以前のバージョンのアセットをチェックアウトできます。詳細については，976 ページ「[バージョンの履歴］ダイアログ・ボックス」および『**Mercury Quality Center ユーザーズ・ガイド**』を参照してください。

アセットをチェックアウトする前に，チェックアウトするアセットが現在チェックインされていることを確認します。自分がすでにチェックアウトしているアセットを開くと，［**チェックアウト**］オプションは無効になります。ほかのユーザによってチェックアウトされているアセットを開くと，［**バージョンの履歴**］オプションを除くすべてのバージョン・コントロール・オプション群が無効になります。

バージョン番号について：Quality Center 10.00 以前のバージョン番号は，1.7.4 のようにピリオドで区切られた 3 つのセグメントで構成されていました。Quality Center 10.00 から，バージョン番号は，12 のように 1 つのセグメントで構成されます。

［開く］ダイアログ・ボックスを使用してアセットの最新のバージョンをチェックアウトするには，次の手順を実行します。

1 以下のいずれかを行います。

アセットの種類	作業
コンポーネントまたは関数ライブラリ	QuickTest のメイン・ウィンドウで，［**ファイル**］＞［**開く**］＞［**コンポーネント**］または［**ファイル**］＞［**開く**］＞［**関数ライブラリ**］を選択するか，［**開く**］の下向き矢印をクリックして一覧からアセット・タイプを選択します。
共有オブジェクト・リポジトリ	オブジェクト・リポジトリ・マネージャの中で，［**ファイル**］＞［**開く**］を選択するか，［**開く**］ボタンをクリックします。
回復シナリオ	回復シナリオ・マネージャの中で，［**開く**］ボタンをクリックします。

［Open ＜ Asset type ＞］ダイアログ・ボックスが開きます。

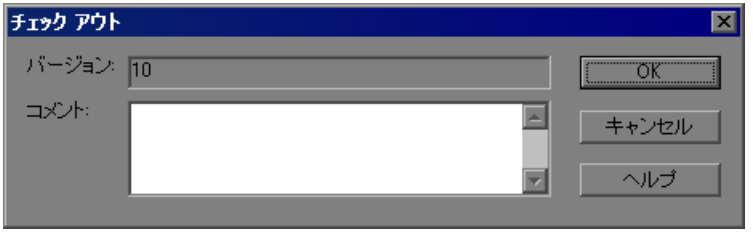
2 アセットを参照して選択します。

3 ［**開く**］の下向き矢印をクリックし，［**開いてチェックアウトする**］を選択します。アセットが開き，ユーザにチェックアウトされます。

［ファイル］メニューを使用してアセットの最新バージョンをチェックアウトするには，次の手順を実行します。

1 チェックアウトするアセットを開きます。

2 ［**ファイル**］＞［**Quality Center バージョン コントロール**］＞［**チェックアウト**］を選択します。［チェックアウト］ダイアログ・ボックスが開き，チェックアウトされるアセット・バージョンが表示されます。



3 ［**コメント**］ボックスに変更の詳細を入力します。

4 ［**OK**］をクリックします。読み取り専用のアセットが閉じて，書き込み可能アセットとして再び自動的に開きます。

5 必要に応じてアセットを表示または編集します。

注：アセットをチェックインせずに変更を保存して閉じることができますが，アセットをチェックインするまで，ほかの Quality Center ユーザは加えられた変更を利用できません。変更を加えたテストをチェックインしない場合は，チェックアウトを取り消すことができます。アセットのチェックインの詳細については，974 ページ「バージョン・コントロール・データベースへのアセットのチェックイン」を参照してください。チェックアウトの取り消しの詳細については，975 ページ「チェックアウト操作の取り消し」を参照してください。

バージョン・コントロール・データベースへのアセットのチェックイン

あるアセットがチェックアウトされていても，Quality Center ユーザは以前にチェックインされたバージョンを実行できます。たとえば，アセットのバージョン 3 をチェックアウトし，いくつか変更を加えてアセットを保存するとします。バージョン 4 としてアセットをバージョン・コントロール・データベースに再びチェックインするまで，Quality Center ユーザはバージョン 3 を引き続き実行できます。

アセットの変更が終了し，Quality Center ユーザに新しいバージョンを使用してもらう準備が整ったら，バージョン・コントロール・データベースにアセットをチェックインします。

注：Quality Center データベースにチェックインしない場合は，チェックアウト操作を取り消すことができます。詳細については，975 ページ「チェックアウト操作の取り消し」を参照してください。

アセットをバージョン・コントロール・データベースにチェックインして戻すと，Quality Center はチェックアウト・ディレクトリからアセットのコピーを削除し，Quality Center プロジェクトのほかのユーザがそのバージョンのアセットを利用できるように，データベース内のアセットのロックを解除します。

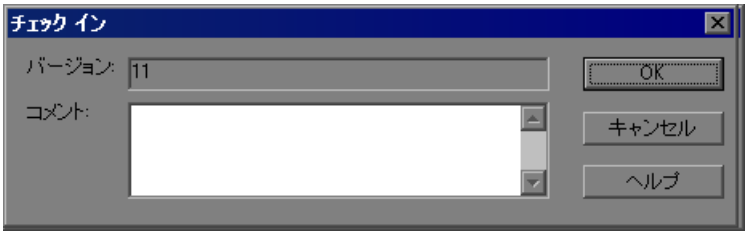
現在開いているアセットをチェックインするには，次の手順を実行します。

1 現在開いているアセットを自分がチェックアウトしていることを確認します。詳細については，976 ページ「アセットのバージョンの履歴の表示」を参照してください。

注：開いているアセットが現在チェックインされていると，［**チェック イン**］オプションは無効になります。ほかのユーザによってチェックアウトされているアセットを開くと，［**バージョンの履歴**］オプションを除くすべての［**Quality Center バージョン コントロール**］オプション群が無効になります。

---

2 [**ファイル**] > [**Quality Center バージョン コントロール**] > [**チェックイン**] を選択します。[チェックイン] ダイアログ・ボックスが表示されます。



アセットのチェックアウト時に変更の詳細を入力した場合，その詳細は [**コメント**] ボックスに表示されます。コメントの入力や修正はこのボックスで行えます。

3 [**OK**] をクリックし，アセットをチェックインします。アセットが閉じて，読み取り専用のアセットとして再び自動的に開きます。

チェックアウト操作の取り消し

アセットをチェックアウトした後に，変更を加えたアセットを Quality Center にアップロードしないと決定した場合は，ほかの Quality Center ユーザがチェックアウトできるように，チェックアウト操作を取り消す必要があります。

チェックアウト操作を取り消すには，次の手順を実行します。

1 チェックアウトしたアセットをまだ開いていない場合は開きます。

2 [**ファイル**] > [**Quality Center バージョン コントロール**] > [**チェックアウトを元に戻す**] を選択します。

3 [**はい**] をクリックし，チェックアウト操作の取り消しを確定します。チェックアウト操作が取り消されます。チェックアウトしたアセットが閉じ，前回チェックインしたバージョンが読み取り専用モードで再び開きます。

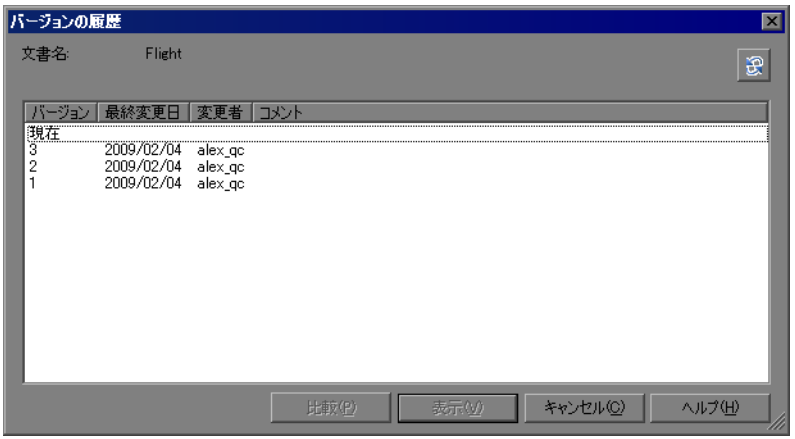
アセットのバージョンの履歴の表示

アセットのバージョンの履歴の表示には，［バージョンの履歴］ダイアログ・ボックスを使用します。このダイアログ・ボックスにより，開発のさまざまな段階でアセットの異なるバージョンの表示や比較ができます。

［バージョンの履歴］ダイアログ・ボックス

説明	アセットのバージョンの履歴の表示，アセットの以前のバージョン内容の表示，およびアセットの 2 つのバージョンの比較ができます。 **アセットのバージョンを表示する場合**：バージョンを選択して，［**表示**］をクリックします。 **アセットの 2 つのバージョンを比較する場合**：2 つのバージョンを選択して，［**比較**］をクリックします。
アクセス方法	➤ **ほとんどのアセット**：アセットを開き，［**ファイル**］＞［**Quality Center バージョン コントロール**］＞［**バージョンの履歴**］メニュー・コマンドを選択します。 ➤ **回復シナリオ**：回復シナリオ・マネージャで，回復シナリオを開いて［**バージョン コントロール**］の下向き矢印をクリックし，［**バージョンの履歴**］を選択します。
詳細	**概念の概要**：968 ページ「Quality Center でのアセットのバージョン管理」 **関連ユーザ・インタフェースのトピック**：979 ページ「[ベースラインの履歴］ダイアログ・ボックス」

次は［バージョンの履歴］ダイアログ・ボックスの画像です。



［バージョンの履歴］ダイアログ・ボックスのオプション

	オプション	説明
	［**文書名**］	現在開いているアセットの名前です。
	［**更新**］ボタン	［バージョンの履歴］ダイアログ・ボックスのバージョンを最新の変更を使用してリロードします。
	［**バージョン**］カラム	アセットのすべてのバージョンの一覧です。
	［**最終変更日**］カラム	各バージョンがチェックインされた日付です。
	［**変更者**］カラム	各バージョンをチェックインしたユーザです。
	［**コメント**］カラム	選択したアセットのバージョンがチェックインされたときに入力されたコメントです。

	オプション	説明
	［**比較**］ボタン	現在開いているアセットの 2 つのバージョンを比較できます。 **2 つのバージョンを比較するには，次の手順を実行します。**比較するバージョンを選択し，［**比較**］をクリックします。アセット比較ツールでアセットの 2 つのバージョンが開かれます。詳細については，952 ページ「QuickTest アセット比較ツール」を参照してください。
	［**表示**］ボタン	現在のアセットの選択したバージョンを表示できます。 **アセットのバージョンを表示するには，次の手順を実行します。**アセットのバージョンを選択し，［**表示**］をクリックします。アセット・ビューアで，アセットのチェックインされたバージョンが開きます。アセット・ビューアの詳細については，962 ページ「QuickTest アセット・ビューア」を参照してください。

ベースラインの履歴の表示

Quality Center では，プロジェクト管理者は，開発のさまざまな段階でプロジェクト全体（またはプロジェクトの一部）の「スナップショット」を提供するベースラインを作成できます。**ベースライン**は，プロジェクトのライフサイクル内の特定のポイントのプロジェクトのバージョンを表します。たとえば，ベースラインは，重要なポイントごとやプロジェクトの特定の段階が完了した場合によく作成されます。ベースラインは，バージョン・コントロールの有効無効にかかわらず，Quality Center プロジェクトに対して作成できます。

プロジェクト管理者は，Quality Center の管理モジュールの［ライブラリ］タブでベースラインを作成します。ベースラインの作成は 2 段階に分けられます。管理者はまずライブラリを作成し，データのインポート元のルート・フォルダを指定します。共有オブジェクト・リポジトリや関数ライブラリなど，関連するリソース・ファイルがすべて含まれていることを確認します。その後実際のベースラインを作成します。これは，ライブラリに含まれるすべてのアセットの最新バージョンで構成されます。プロジェクトでバージョン・コントロールが有効な場合，アセットはすべてチェックインされている最新バージョンのものです。

作成プロセス中，Quality Center は，これらのすべてのアセット（関連リソース・ファイルなど）がベースラインに含まれていることを確認します。いずれかのアセットが含まれていない場合，Quality Center は管理者に通知し，ライブラリとベースラインをそれに応じて変更できるようにします。詳細については，『**Mercury Quality Center ユーザーズ・ガイド**』を参照してください。

Quality Center では，ベースライン全体を表示，比較できます。

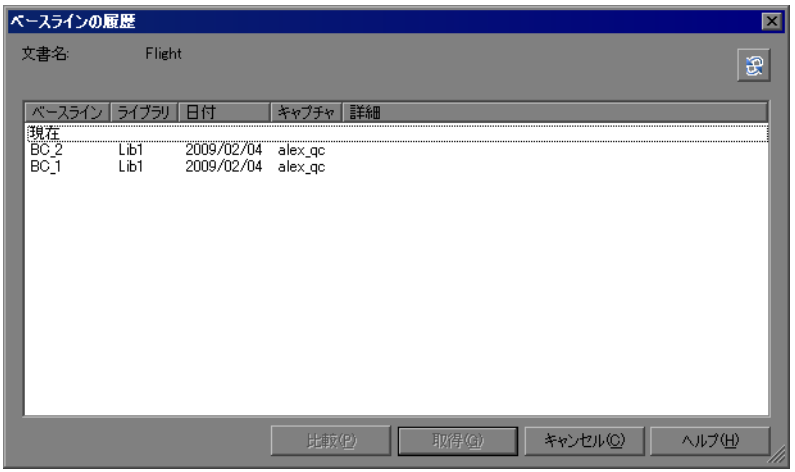
QuickTest では，ベースライン内に保存されたアセットを表示，比較できます。これにより，プロジェクトの全工程の中で，特定の段階のアセットの内容を確認できます。

ベースラインからコンポーネントを実行することもできます。

［ベースラインの履歴］ダイアログ・ボックス

説明	アセットのベースライン「スナップショット」を読み取り専用で表示，比較できます。
アクセス方法	➤ **ほとんどのアセット**：アセットを開き，［**ファイル**］＞［**Quality Center バージョン コントロール**］＞［**ベースラインの履歴**］メニュー・コマンドを選択します。 ➤ **回復シナリオ**：回復シナリオ・マネージャで，回復シナリオを開いて［**バージョン コントロール**］の下向き矢印をクリックし，［**ベースラインの履歴**］を選択します。
重要	Quality Center のテストのラボ・モジュールでは，［**Pin to Baseline**］オプションを使用して，アセットのベースライン・バージョンを実行できます。詳細については，『**Mercury Quality Center ユーザーズ・ガイド**』を参照してください。
詳細	**概念の概要**：978 ページ「ベースラインの履歴の表示」 **関連ユーザ・インタフェースのトピック**：976 ページ「［バージョンの履歴］ダイアログ・ボックス」

次は［ベースラインの履歴］ダイアログ・ボックスの画像です。



［ベースラインの履歴］ダイアログ・ボックスのオプション

	オプション	説明
	［**文書名**］	現在開いているアセットの名前を指定します。
	［**更新**］ボタン	［ベースラインの履歴］ダイアログ・ボックスのベースラインを最新の変更を使用してリロードします。たとえば，このダイアログ・ボックスを開いている間にベースラインが追加された場合，［**更新**］をクリックするとベースラインの一覧が更新されます。
	［**ベースライン**］カラム	このアセットを含むすべてのベースラインが一覧表示されます。ベースラインは Quality Center プロジェクト内で定義されます（**管理**モジュール＞［**ライブラリ**］タブ）。
	［**ライブラリ**］カラム	各ベースラインが作成されたライブラリが一覧表示されます。
	［**日付**］カラム	各ベースラインが作成された日付が一覧表示されます。
	［**キャプチャ**］カラム	表示されている各ベースラインを作成した Quality Center ユーザが一覧表示されます。

	オプション	説明
	［**詳細**］カラム	ベースラインの作成時に追加されたコメントが表示されます。
	［**比較**］ボタン	2 つのベースラインで，現在開いているアセットの比較を表示できます。 **2 つのベースラインを比較するには，次の手順を実行します。**比較するベースラインを選択し，［**比較**］をクリックします。アセット比較ツールでアセットの 2 つのベースライン・バージョンが開きます。詳細については，952 ページ「QuickTest アセット比較ツール」を参照してください。
	［**取得**］ボタン	選択したベースラインから現在のアセットを開くことができます。 **ベースラインに格納されたアセットを表示するには，次の手順を実行します。**一覧からベースラインを選択し，［**表示**］をクリックします。 ［**取得**］をクリックすると，QuickTest により次の操作が行われます。 ➤ 現在開いているアセットを閉じます。 ➤ 選択したベースラインから同じアセットを開きます。 ➤ アセットに関連付けられているリソース・ファイルのベースライン・バージョンがもし存在すれば，呼び出された場合にロードします。**注**：リソース・ファイルが相対パス経由で関連付けられている場合，ロードされるのはベースラインのバージョンではなく最新バージョンのリソース・ファイルです。

バージョンの履歴とベースラインの履歴

本項では，バージョンの履歴とベースラインの履歴の違いに焦点を当て，それぞれを利用する場合について説明します。

➤ 必要に応じて，バージョン・コントロールを使用してアセットをチェックイン，チェックアウトします。たとえば，アセットのチェックインは，毎日行うことも，重要な成果を達成した場合にだけ行うこともできます。これにより，アセットの開発を監視できます。

特定の日付や，特定のユーザがアセットをチェックインした後にアセットの内容を表示する場合，［バージョンの履歴］オプションを使用してアセットを表示，または比較します。

➤ Quality Center プロジェクト管理者はベースラインを作成します。ベースラインは，プロジェクトのライフサイクルの中のさまざまな重要ポイントで，その時点のプロジェクトのアセットの「スナップショット」を表すものです。各ベースラインは，ベースラインの作成時に管理者が指定したアセットにリンクしています。ベースラインに示されるアセットのバージョンは，常にベースラインの作成時にチェックインされていたバージョンです。

特定のポイント向けに保存されたアセットを表示する場合，［ベースラインの履歴］オプションを使用します。

第 XII 部

付録

付録 A

FAQ（よくある質問）

本章では，QuickTest の上級ユーザから寄せられることの多いいくつかの質問についてお答えします。質問と回答は次の項に分類されています。

本章の内容

- コンポーネントの作成（986 ページ）
- 関数ライブラリを使用した作業（987 ページ）
- 動的なコンテンツを使った作業（988 ページ）
- Web に関する高度な問題（990 ページ）
- 標準 Windows 環境（993 ページ）
- コンポーネントの保守（993 ページ）
- QuickTest のパフォーマンスの向上（994 ページ）

コンポーネントの作成

➤ **QuickTest でサポートされていないオブジェクトまたは環境で記録を実行するには，どのようにすればよいですか。**

さまざまな方法があります：

➤ QuickTest Professional で使用可能な任意の外部アドインをインストールしてロードします。QuickTest では，Java，Oracle，.NET，SAP Solutions，Siebel，PeopleSoft，ターミナル・エミュレータ，Web サービスなど，数多くの開発環境がサポートされています。

➤ 識別されなかったクラスやユーザ定義のクラスのオブジェクトは，標準の Windows クラスにマップできます。オブジェクトのマッピングの詳細については，219 ページ「ユーザ定義のテスト・オブジェクト・クラスの割り当て」を参照してください。

➤ QuickTest が提供しているアドイン拡張機能を使用すると，QuickTest に組み込まれている各種オブジェクトのサポートを拡張できます。これにより，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するよう QuickTest に指示でき，テスト・オブジェクトの振る舞いを指定できるようになります。また，QuickTest 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・オブジェクトの特定の振る舞いを完全にサポートするコンポーネントを作成できます。

➤ **アプリケーションをコンポーネントから開くにはどうすればよいですか。**

アプリケーションを開くステップを追加するには，［**項目**］カラムで［**操作**］を選択し，［**操作**］カラムで［**OpenApp**］を選択した後，次のように，アプリケーションの完全パスを［**値**］カラムに入力します。

C:¥Program Files¥HP¥QuickTest Professional¥samples¥flight¥app¥flight4a.exe

➤ **QuickTest は，どのようにして Web ページでのユーザのプロセスをキャプチャするのでしょうか。**

QuickTest では，Microsoft Internet Explorer ブラウザにフックをかけます。ユーザが Web ベースのアプリケーションを操作すると，QuickTest によって，ユーザの操作が記録されます （記録されたユーザ操作を変更する方法については，『**HP QuickTest Professional アドイン・ガイド**』を参照してください）。記録したコンポーネントは，ステップを元の順序で実行する QuickTest の機能を使って実行できます。

関数ライブラリを使用した作業

➤ 関数やサブルーチンを関数ライブラリに保存できますか。

関数が含まれる 1 つまたは複数の VBScript 関数ライブラリを作成できます。そして，それらをコンポーネントに関連付けられているアプリケーション領域に関連付けることによって，任意のコンポーネントで使用できます。QuickTest 関数ライブラリ・エディタを使用して，関数ライブラリの作成とデバッグを行うことができます。

関数を QuickTest テスト・オブジェクトのメソッドとして登録できます。登録したメソッドは，実行セッションの間だけ既存のテスト・オブジェクト・メソッドの機能をオーバーライドしたり，テスト・オブジェクト・クラスの新しいメソッドとして登録したりできます。

詳細については，第 12 章「ユーザ定義関数および関数ライブラリを使用した作業」および第 13 章「アプリケーション領域の管理」を参照してください。

➤ 実行セッション中に情報を入力するにはどうすればよいですか。

VBScript の InputBox 関数を，関連付する関数ライブラリに挿入してユーザ定義関数を作成できます。このユーザ定義関数を使用して，ユーザに入力を求めるダイアログ・ボックスを表示してからコンポーネントの実行を続けることができます。ユーザが入力した値は，その後の実行セッションで使用できます。InputBox 関数の詳細については，『**VBScript Reference**』（英語版）を参照してください。

➤ テスト結果をカスタマイズするにはどうすればよいですか。

次のように，**ReportEvent** メソッドを使用することで，メッセージを実行結果レポートに情報を追加できます。

```
Reporter.ReportEvent 1, "Custom Step", "The user-defined step failed"
```

詳細については，『**HP QuickTest Professional Object Model Reference**』（英語版）を参照してください。

QuickTest 実行セッションの結果はすべて 1 つの **.xml** ファイル（**results.xml** という名前です）に保存されています。必要に応じて，このファイルに変更を加えることができます。**QuickTest テスト結果のスキーマ**（QuickTest Professional ヘルプから利用できます）は，テスト結果のカスタマイズに役立ちます。

動的なコンテンツを使った作業

➤ **表示するたびに動的に変化するオブジェクトを対象とした，コンポーネントを作成し，実行するにはどうすればよいでしょうか。**

アプリケーション内のオブジェクトで動的コンテンツを持つものは内容が変化することがあります。正規表現，**Description** オブジェクト，リポジトリ・パラメータ，または **SetTOProperty** ステップを使用してコンポーネントを実行するときに，それらのオブジェクトが QuickTest によって認識されるように，オブジェクトの動的記述を作成できます。

➤ **子ウィンドウの有無を検査するには，どうすればよいですか。**

1 つのウィンドウ内のリンクが別のウィンドウを作成することがあります。

Exist プロパティを使用して，ウィンドウが存在するかどうかを検査できます。たとえば，次のような場合です。

```
If Window("Main").ActiveX("Slider").Exist Then
. . .
```

また，ChildObjects メソッドを使用して，テスクトップ上あるいはほかの親オブジェクト内のすべての子オブジェクト（または，ある記述と一致する子オブジェクトのサブセット）を取得することもできます。

例：

```
Set oDesc = Description.Create
oDesc("Class Name").Value = "Window"

Set coll = Desktop.ChildObjects(oDesc)
For i = 0 to coll.count -1
        msgbox coll(i).GetROProperty("text")
Next
```

Exist プロパティおよび ChildObjects メソッドの詳細については，『**HP QuickTest Professional Object Model Reference**』（英語版）を参照してください。

➤ QuickTest は，動的に生成される URL や Web ページをどのようにして記録するのでしょうか。

QuickTest は，リンクがページに表示されると，実際にそのリンクをクリックします。そのため，QuickTest はオブジェクト自体ではなく，ページ上のリンクなど特定のオブジェクトを検索する方法を記録します。たとえば，動的に生成された URL へのリンクが画像である場合，QuickTest は「IMG」HTML タグと，その画像の名前を記録します。これにより，それ以後 QuickTest はこの画像を検索し，その画像をクリックできるようになります。

➤ QuickTest は，タブをどのように処理するのでしょうか。

QuickTest が提供しているいくつかのメソッドを **Browser** テスト・オブジェクトで使用することにより，Web ブラウザのタブを管理できます。

OpenNewTab は，現在の Web ブラウザで新しいタブを開きます。

IsSiblingTab は，指定したタブが同じブラウザ・ウィンドウ内の現在のタブ・オブジェクトの兄弟タブかどうかを示します。

Close は，タブが複数存在する場合は現在のタブを閉じ，ブラウザにタブが 1 つしか含まれていない場合はブラウザ・ウィンドウを閉じます。

CloseAllTabs は，ブラウザ内のすべてのタブを閉じ，ブラウザ・ウィンドウを閉じます。

これらの **Browser** 関連メソッドの詳細については，『**HP QuickTest Professional Object Model Reference**』（英語版）の「**Web**」セクションを参照してください。

Web に関する高度な問題

➤ **QuickTest はクッキーをどのように処理するのでしょうか。**

CGI スクリプトなど接続のサーバ側では，クッキーを利用することで，接続のクライアント側に情報を格納したり，そこから情報を取得したりできます。

QuickTest ではユーザごとにメモリにクッキーを格納し，ブラウザは通常どおりにそれらを処理します。

➤ **Web ページのクッキーはどこにありますか。**

Internet Explorer ブラウザが使用するクッキーには，関数内の **Object** プロパティを使用して，ブラウザの DOM（ドキュメント・オブジェクト・モデル）を通じてアクセスできます。次の例では，クッキーのコレクションがブラウザから返されます。

```
Browser("Flight reservations").Page("Flight reservations").Object.Cookie
```

➤ **QuickTest は，セッション ID をどのように処理するのでしょうか。**

ブラウザでなくサーバが，通常はクッキーによって，またはすべてのリンクにセッション ID を埋め込むことによって，セッション ID を処理します。これは，QuickTest には影響を与えません。

➤ **QuickTest は，サーバのリダイレクトをどのように処理するのでしょうか。**

サーバがクライアントをリダイレクトした場合，通常はクライアントはそれに気付かないため，リダイレクトの間違いが起こることはありません。ほとんどの場合，クライアントはサーバ上の別のスクリプトにリダイレクトされます。この追加のスクリプトが，以降に表示されるページの HTML コードを生成します。これは，QuickTest にもブラウザにも影響を与えません。

➤ **QuickTest は，META タグをどのように処理するのでしょうか。**

META タグは，ページの表示に影響を与えません。META タグには通常，ページの作成者，更新頻度，ページの内容説明，およびページの内容を表すキーワードの情報だけが含まれています。したがって，QuickTest は問題なく META タグを処理できます。

➤ QuickTest は .asp および .jsp に対応していますか。

Active Server Page テクノロジを使用して動的に生成される Web ページには，.asp という拡張子が割り当てられています。Java Server Page テクノロジを使用して動的に生成される Web ページには，jsp という拡張子が割り当てられています。これらは完全にサーバ側の技術であるため，QuickTest には影響しません。

➤ QTP では AJAX はどの程度サポートされていますか。

QuickTest Professional Web Add-in Extensibility を使用すると，カスタムの Web コントロールの独自サポートを追加できます。Web Add-in Extensibility SDK によって，いくつかの ASP .NET AJAX コントロールを部分的にサポートするサンプル・ツールキット・サポート・セットがインストールされます。このサンプルを使えば，AJAX コントロールの独自サポートを作成する方法を学習できます。詳細については，『**HP QuickTest Professional Web Add-in Extensibility Developer Guide**』（英語版）を参照してください。

➤ QuickTest は，COM に対応していますか。

QuickTest は，COM 標準に準拠しています。

QuickTest は，Web ページに埋め込まれた COM オブジェクトをサポートしており（現在，COM オブジェクトは Microsoft Internet Explorer を使用している場合にだけアクセス可能です），VBScript 内で COM オブジェクトを駆動できます。

➤ QuickTest は，XML に対応していますか。

XML （eXtensible Markup Language）は，Web ドキュメント用に SGML を簡略化したものです。XML を使えば，Web デザイナーはカスタマイズした独自のタグを作成できます。QuickTest は XML に対応しており，XML タグをオブジェクトとして認識します。

詳細については，『**HP QuickTest Professional ユーザーズ・ガイド**』，および『**HP QuickTest Professional Object Model Reference**』（英語版）の「**Utility**」セクションの **XMLUtil** オブジェクトを参照してください。

➤ HTML タグに直接アクセスするにはどうすればよいですか。

QuickTest では，Internet Explorer の DOM（ドキュメント・オブジェクト・モデル）に直接アクセスでき，DOM を通じて HTML タグに直接アクセスできます。DOM へのアクセスは .Object の表記法を使用して実行します。

次の関数は，Internet Explorer ページ内のすべてのタグについて反復処理を行う方法の例です。反復処理後，この関数は Reporter オブジェクトを使用してタブの内部テキスト（タグに囲まれたテキスト）をテスト結果に出力します。

```
' すべての要素が内部テキストを持っているわけではないため，On Error オプションを使用します。
On Error Resume Next
Set Doc = Browser("CNN Interactive").Page("CNN Interactive").Object

' ページ内のすべてのオブジェクトについてループ処理します。
For Each Element In Doc.all
    TagName = Element.TagName ' タグ名を取得します。
    InnerText = Element.innerText ' 内部テキストを取得します。

    ' 情報をテスト結果に書き込みます。
    Reporter.ReportEvent 0, TagName, InnerText
Next
```

➤ Internet Explorer のドキュメント・オブジェクト・モデルに関する情報はどこで入手できますか。

Internet Explorer の DOM の詳細については，次の Web サイトを参照してください。

ドキュメント・オブジェクト：http://msdn2.microsoft.com/en-us/library/ms531073.aspx

その他の DHTML オブジェクト：http://msdn2.microsoft.com/en-us/library/ms533054.aspx

DHTML の全般的なリファレンス：http://msdn2.microsoft.com/en-us/library/ms533050.aspx

➤ キーボードのキーのコマンド（ショートカット・コマンドなど）を，Type メソッドをサポートしていないオブジェクトに送信するにはどうすればよいですか。

Type メソッドをサポートしていないオブジェクトの場合は，Windows Scripting の **SendKeys** メソッドを使用します。詳細については，『**Microsoft VBScript Language Reference**』を参照してください（［**ヘルプ**］＞［**QuickTest Professional ヘルプ**］＞［**VBScript Reference**］＞［**Windows Script Host**］を選択します）。

標準 Windows 環境

➤ **非標準のメニューを対象とした記録はどのように実行すればよいですか。**

メニューを記録するときの QuickTest の振る舞いを変更できます。この振る舞いを制御するオプションは，［Windows アプリケーション］＞［詳細設定］表示枠にあります（［**ツール**］＞［**オプション**］＞［**Windows アプリケーション**］ノード＞［**詳細設定**］ノード）。

詳細については，『**HP QuickTest Professional アドイン・ガイド**』を参照してください。

コンポーネントの保守

➤ **アプリケーションに変更が加えられた場合，コンポーネントをどのように保守すればよいですか。**

アプリケーションに変更が加えられた場合のコンポーネントの保守方法は，アプリケーションに加えられた変更の量に応じて異なります。アプリケーション全体を対象に 1 つの大きなコンポーネントを作成するのではなく，コンポーネントを小さなグループに分けて作成するべき主な理由の 1 つがこれです。

同じテスト・オブジェクトが含まれるコンポーネントが数多くある場合は，1 か所で集中的にオブジェクト情報を更新できるように，共有オブジェクト・リポジトリを使用することをお勧めします。

［**更新モード**］オプションを使用すると，チェックポイントの変更された情報を更新したり，アプリケーション内のオブジェクトの識別に使用されている 1 組の認識プロパティを変更したりできます。詳細については，787 ページ「更新モード・オプションを使ったコンポーネントの更新」を参照してください。

オブジェクト・リポジトリに保存されている認識プロパティ値とアプリケーション内のオブジェクト・プロパティ値との間に不一致がある場合は，［**メンテナンス モード**］を使用して不一致の修正に役立てることができます。メンテナンス・モードでコンポーネントを実行すると，QuickTest によってコンポーネントが実行された後，オブジェクト・リポジトリの不一致が原因で実行できないステップが出現するたびに，ステップとオブジェクト・リポジトリを更新するためのプロセスがひととおり示されます。詳細については，770 ページ「メンテナンス実行ウィザードを使ったコンポーネントの実行」を参照してください。

➤ **古いコンポーネントからのテスト結果ファイルを削除するにはどうすればよいですか。**

テスト結果削除ツールを使用して，ファイル・システムまたは Quality Center プロジェクト内の特定の場所にある，すべてのテスト結果の一覧を表示できます。その後不要なテスト結果を削除できます。

削除するテスト結果をより簡単に特定できるように，テスト結果削除ツールによって，テスト結果を名前，日付，サイズなどで並べ替えることができます。

このユーティリティを開くには，［**スタート**］＞［**プログラム**］＞［**QuickTest Professional**］＞［**Tools**］＞［**テスト結果削除ツール**］を選択します。

QuickTest のパフォーマンスの向上

QuickTest の動作速度を向上させるには，次のいずれかを行います。

- アドイン・マネージャで，QuickTest の起動時に特定の QuickTest セッションに必要なアドインのみをロードします。こうすることで，実行セッション中のオブジェクトの学習時のパフォーマンスが向上します。アドインのロードの詳細については，『**HP QuickTest Professional アドイン・ガイド**』を参照してください。
- ビジネス・プロセス・テスト内のすべてのコンポーネントで同じアプリケーション領域の使用を試みます。これにより，パフォーマンスが向上します。
- コンポーネントを「高速モード」で実行します。それには，［オプション］ダイアログ・ボックスの［実行］表示枠で，［**高速**］オプションを選択します。これにより，QuickTest は各ステップで実行矢印を表示せずにコンポーネントを実行するため，コンポーネントの実行を高速化できます。［オプション］ダイアログ・ボックスの［実行］表示枠の詳細については，638 ページ「テストの実行オプションの設定」を参照してください。

- 実行セッション結果として，アプリケーションの画像やムービーをキャプチャして保存するタイミングを指定します。エラー発生時などの特定の条件を満たした場合にだけ画面キャプチャやムービー・セグメントを保存したり，画像をまったく保存しないようにすることで，テスト実行時間を短縮し，ディスク容量を節約できます。そのためには，［オプション］ダイアログ・ボックスの［実行］＞［画面キャプチャ］表示枠で，［**静止画像キャプチャをテスト結果へ保存**］および［**ムービーをテスト結果へ保存**］オプションを使用します。詳細については，642 ページ「[オプション］ダイアログ・ボックス：[実行］＞［画面キャプチャ］表示枠」を参照してください。
- テスト結果レポートを一時フォルダに保存し，コンポーネントを実行するたびに以前の実行セッションからの結果を上書きします。詳細については，670 ページ「コンポーネント全体の実行」を参照してください。
- 結果削除ツールを使い，定義した条件に従って，システムから不要な実行結果や古い実行結果を削除できます。これによって貴重なディスク・スペースを解放できます。詳細については，710 ページ「テスト結果削除ツールを使った結果の削除」を参照してください。

付録 B

カスタムのプロセス・ガイダンス・パッケージの作成

本章では，カスタムのプロセス・ガイダンス・パッケージの作成プロセスをひととおり説明します。カスタム・パッケージは組織の QuickTest ユーザに配布できます。QuickTest ユーザは，作業時に QuickTest でカスタム・パッケージのプロセスを表示し，組織のプロセスと標準に従う上で役立てることができます。

本章の内容

- プロセス・ガイダンス・パッケージについて（997 ページ）
- パッケージ設定ファイルについて（998 ページ）
- データ・ファイルの作成（1000 ページ）
- QuickTest でのカスタムのプロセス・ガイダンス・パッケージのインストール（1001 ページ）

プロセス・ガイダンス・パッケージについて

プロセス・ガイダンス・パッケージは，パッケージ設定ファイルとデータ・ファイルの 2 つの要素で構成されています。

- **パッケージ設定ファイル**：この XML ファイルでは，パッケージに含まれる「**プロセス**」と，各プロセスの「**グループ**」および「**アクティビティ**」の構造を定義します。
- **データ・ファイル**：1 組の HTML ファイルで，各 HTML ファイルには単一のアクティビティの内容が含まれています。

プロセス・ガイダンスの概要と QuickTest での使い方については，第 36 章「プロセス・ガイダンスを使った作業」を参照してください。

パッケージ設定ファイルについて

新しいパッケージを作成するには，最初に，パッケージに含めるプロセスを記述し，各プロセスのグループとアクティビティの構造を設定する，XML ファイルを作成します。この構造は，QuickTest の［**Process Guidance Activities**］表示枠で選択されたプロセスの内容のテーブルとして表示されます。

重要：設定ファイルは **Configuration.xml** という名前で保存します。

---

2 つのプロセスが含まれているパッケージ設定ファイルの例を次に示します。

```
<?xml version="1.0" encoding="UTF-8" ?>
<ProcessGuidance Name="MyCustomPackage">
    <Process Name="My Process" ID="Process1" DocType="test" Addin="web"
SortLevel="4" >
        <Group Name="New User Overview">
            <Activity Name="Step 1" Address="Step1.html"  />
            <Activity Name="Step 2" Address="Step2.html"  />
        </Group>
    </Process>
    <Process Name="Important Processes" ID="Process2" DocType="test|AA"
SortLevel="3">
        <Group Name="Getting Started">
            <Activity Name="Open" Address="F:¥ProcessData¥open.html"  />
            <Activity Name="Create" Address="F:¥ProcessData¥create.html"  />
            <Activity Name="Test" Address="F:¥ProcessData¥test.html"  />
            <Activity Name="Debug" Address="F:¥ProcessData¥debug.html"  />
        </Group>
        <Group Name="Finish">
            <Activity Name="Save" Address="F:¥ProcessData¥save.html"  />
            <Activity Name="Close" Address="F:¥ProcessData¥close.html"  />
            <Activity Name="Exit" Address="F:¥ProcessData¥exit.html"  />
        </Group>
    </Process>
</ProcessGuidance>
```

XML の詳細

本項では，パッケージ設定ファイルで使用できる要素と属性について説明します。

- **＜Process＞要素**：新しいプロセスを定義します。この要素では次の属性がサポートされています。
  - **Name**：QuickTest の［プロセス ガイダンス］表示枠に表示するプロセスの名前。
  - **ID**：一意の識別名。この名前は，同じ名前を持つ 2 つのプロセスを区別するのに使用されます。
  - **DocType**：このプロセスが適用可能な QuickTest のドキュメント・タイプを示します。指定した場合は，関連するドキュメント・タイプが開いているときのみ，プロセスを使用できます。

    前述の例では，QuickTest ユーザがテスト・ドキュメントを開いている場合は両方のプロセスを使用できますが，アプリケーション領域ドキュメントが開いている場合は 2 番目のプロセスのみ使用できます。

    取り得る値：
    - **test**：テスト・ドキュメント。
    - **AA**：アプリケーション領域ドキュメント。
    - **BC**：ビジネス・コンポーネント・ドキュメント。
    - **SBC**：スクリプト・コンポーネント・ドキュメント。
  - **Addin**：このプロセスが適用可能な QuickTest アドインを示します。指定した場合は，関連するアドインがロードされているときのみ，プロセスを使用できます。

    前述の例では，Web アドインがロードされている場合にのみ，最初のプロセスを使用できます。2 番目のプロセスは常に表示されます。

    アドイン・マネージャに表示されるアドイン名を使用して，アドインの値を指定します。
  - **SortLevel**：プロセス・リスト内でのプロセスの位置を決めます。このリストは，［プロセス ガイダンス管理］ダイアログ・ボックス，および QuickTest の［**自動化**］＞［**プロセス ガイダンス リスト**］メニューに表示されます。

- **＜Group＞要素**：プロセス内の新しいグループを定義します。この要素では次の属性がサポートされています。
  - **Name**：前述の **＜Process＞** 要素の **Name** 属性と同じです。
  - **ID**：前述の **＜Process＞** 要素の **ID** 属性と同じです。
  - **Addin**：前述の **＜Process＞** 要素の **Addin** 属性と同じです。
- **＜Activity＞要素**：グループ内部のアクティビティを定義します。
  - **Name**：前述の **＜Process＞** 要素の **Name** 属性と同じです。
  - **ID**：前述の **＜Process＞** 要素の **ID** 属性と同じです。
  - **Addin**：前述の **＜Process＞** 要素の **Addin** 属性と同じです。
  - **Address**：該当する HTML データ・ファイルのパス。ファイル・システムまたは HTTP アドレス上の，ローカル・パスまたはネットワーク・パスになります。相対パスを指定した場合は，設定ファイルの場所を基準とする相対的な場所に解決されます。

データ・ファイルの作成

各データ・ファイルには，単一のプロセス・ガイダンス・アクティビティの HTML コンテンツが含まれています。［**プロセス ガイダンスのアクティビティ**］表示枠でアクティビティ・リンクをクリックすると，QuickTest の［**プロセス ガイダンスの説明**］表示枠のブラウザ・コントロールに HTML コンテンツが表示されます。

パッケージ・データ・ファイルには，内容を組織の標準のスタイルで表示するための **.css** ファイルへの参照を含めることができ，ブラウザで表示できる任意の内容を含めることができます。

また，HTML ページには，QuickTest のダイアログ・ボックスをアクティブにしたり，QuickTest **UI** オートメーション・オブジェクトを使ってほかのプロセス・ガイダンスのプロセスやアクティビティにジャンプする特別なコードも追加できます。詳細については，『**Automation Object Model Reference**』（［**ヘルプ**］＞［**QuickTest Professional ヘルプ**］＞［**HP QuickTest Professional 詳細リファレンス**］＞［**HP QuickTest Professional Automation Object Model Reference**］）を参照してください。

HTML ファイルと，その HTML ファイルが参照しているフォルダやファイルはすべて，ユーザのローカルのハード・ディスク・ドライブ，ファイル・システムのネットワーク上の場所，または Web サーバに格納できます。パッケージ設定ファイル（各 **Activity** 要素の **Address** 属性）によって，各アクティビティの HTML リンクが提供されます。

各アクティビティの HTML ファイルは，その内容が［Process Guidance Description］表示枠に標準のサイズで表示されるときに最小限のスクロールで済むように記述してください。

HTML ファイルが長すぎる場合は，QuickTest ユーザが作業時に参照しやすいように，ファイルを複数のプロセス・ガイダンス・アクティビティに分割することもできます。

QuickTest でのカスタムのプロセス・ガイダンス・パッケージのインストール

次の 2 つの方法で，カスタムのプロセス・ガイダンス・パッケージを配布およびインストールできます。

- zip ファイルからのプロセス・ガイダンス・パッケージのインストール
- レジストリ・キーによるプロセス・ガイダンス・パッケージのインストール

zip ファイルからのプロセス・ガイダンス・パッケージのインストール

1. **Configuration.xml** ファイルとすべての HTML データ・ファイル（および，HTML ファイル参照されるすべてのファイルまたはフォルダ）を格納するフォルダを作成します。
2. フォルダを zip ファイルに圧縮し，その **.zip** ファイルを関連するすべての QuickTest ユーザに送信するか，ユーザがアクセスできる場所に格納します。
3. QuickTest で，［**ファイル**］＞［**プロセス ガイダンス管理**］を選択します。［プロセス ガイダンス管理］ダイアログ・ボックスが開きます。
4. ［**追加**］ボタンをクリックし，**.zip** ファイルを参照します。パッケージが追加され，そのプロセスがダイアログ・ボックスに表示されます。

レジストリ・キーによるプロセス・ガイダンス・パッケージのインストール

1 **Configuration.xml** ファイルとデータ・ファイルを用意します。

2 データ・ファイルを，ローカルまたは共有ネットワーク・フォルダ，または Web サーバに置きます。**Configuration.xml** ファイルで，**Activity** 要素の **Address** 属性がこの場所を指し示していることを確認します。

3 **Configuration.xml** を QuickTest コンピュータのローカル・ドライブにコピーします。

4 レジストリ・エディタを開き，次のキーを探します。

HKEY_LOCAL_MACHINE¥SOFTWARE¥Mercury Interactive¥QuickTest Professional¥MicTest¥ProcessGuidance¥ConfFiles

5 このキーに，**Configuration.xml** ファイルへのパスを値として追加します。次回 QuickTest を開いたときに，新しいパッケージが追加されます。

付録 C

ビットマップ・チェックポイントのカスタマイズ

重要：この付録は，ビットマップ・チェックポイントでビットマップを比較するアルゴリズムをカスタマイズする COM プログラマを対象としています。

標準設定では，ビットマップ・チェックポイントでは，実際のビットマップと期待されるビットマップがピクセルごとに比較され，何らかの違いがあると失敗します。QuickTest では，ユーザがビットマップ・チェックポイントの許容範囲を定義して，ビットマップ比較を改善したり，柔軟なものにできます。詳細については，580 ページ「ビットマップ比較の微調整」を参照してください。

チェックポイントでビットマップを比較する方法をさらにカスタマイズする必要がある場合は，要件に応じてビットマップを比較するカスタム・コンペアラを開発できます。カスタム・コンペアラは COM オブジェクトとして開発し，QuickTest コンピュータにインストールして登録します。これにより，QuickTest ユーザはカスタム・コンペアラを使って，ビットマップ・チェックポイントで（チェックポイントごとに）比較を実行できます。

本章の内容

- ビットマップ・チェックポイントのカスタマイズについて（1004 ページ）
- カスタム・ビットマップ・コンペアラの開発（1007 ページ）
- チュートリアル：カスタム・コンペアラの作成（1016 ページ）
- ビットマップ・チェックポイント・カスタマイズ・サンプルの使い方（1027 ページ）

ビットマップ・チェックポイントのカスタマイズについて

ビットマップ・チェックポイントをカスタマイズするには，カスタム・コンペアラを開発します。カスタム・コンペアラは，明確なアルゴリズムに従ってチェックポイントでビットマップ比較を実行するために開発した COM オブジェクトです。開発する COM オブジェクトは，QuickTest がタイプ・ライブラリに用意しているインタフェースを実装し，QuickTest がビットマップ・コンペアラに定義しているコンポーネント・カテゴリに登録する必要があります。タイプ・ライブラリ（**BitmapComparer.tlb**）とカテゴリ ID（**ComponentCategory.h** で定義）は，**＜ QuickTest のインストールフォルダ ＞ ¥dat¥BitmapCPCustomization** にあります。

QuickTest ユーザがビットマップ・チェックポイントを作成したり編集したりすると，QuickTest によって［ビットマップ・チェックポイントのプロパティ］ダイアログ・ボックスに（QuickTest 標準コンペアラに加えて）登録されたカスタム・コンペアラが表示されます。ユーザはそのとき，テストする特定のアプリケーションまたはビットマップのテスト要件に応じてコンペアラを選択できます。QuickTest でのカスタム・コンペアラの使い方に関する詳細は，591 ページ「［ビットマップ チェックポイントのプロパティ］ダイアログ・ボックスのカスタム・コンペアラ・オプション」を参照してください。

カスタム・コンペアラの開発を始める前に，「カスタム・コンペアラの開発で考慮すべき事項」の情報を慎重に検討してください。

ビットマップ・チェックポイントのカスタマイズでビットマップ・チェックポイントの使用を強化した状況の例は，1005 ページ「使用事例シナリオ：アプリケーション内で位置が変化する画像の処理」を参照してください。

カスタム・コンペアラの開発で考慮すべき事項

- カスタム・コンペアラを開発するには，画像処理について理解し，COM オブジェクトの開発方法を知る必要があります。
- カスタム・コンペアラは，COM オブジェクトの作成をサポートする言語および開発環境を使って実装できます。
- カスタム・コンペアラは QuickTest のコンテキスト内で実行されます。そのため，カスタム・コンペアラを開発するときは，その動作と性能が QuickTest の動作と性能に影響を与えるので，注意する必要があります。

- カスタム・コンペアラを使ってコンポーネントとビットマップ・チェックポイントを実行するコンピュータには，カスタム・コンペアラをインストールして登録する必要があります。
- 新しいバージョンのカスタム・コンペアラをインストールして登録する前に，既存のコンペアラの登録を抹消します。
- 同じ QuickTest コンピュータに，複数のカスタム・コンペアラをインストールして登録できます。［ビットマップ・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，QuickTest によって使用可能なすべてのカスタム・コンペアラと，QuickTest の標準コンペアラが表示されます。QuickTest ユーザは，各ビットマップ・チェックポイントに使用する適切なコンペアラを選択できます。
- カスタム・コンペアラを実行するコンピュータには，カスタム・コンペアラの DLL を構築した設定と関連する実行環境をインストールする必要があります。
- 特定バージョンの開発環境を使ってカスタム・コンペアラの DLL を作成し，その DLL が実行されるコンピュータには，対応する実行環境をインストールする必要があります。

使用事例シナリオ：アプリケーション内で位置が変化する画像の処理

Ben は品質保証エンジニアで，QuickTest の使用に熟練しており，頻繁にビットマップ・チェックポイントを使って，テストするユーザ・インタフェースでさまざまなアイコンや画像の外観をテストしています。プログラミングの経験がありません。

Joanne はソフトウェア・エンジニアで，画像処理に熟練しており，COM プログラミングに精通しています。

Ben は家具購買アプリケーションのユーザ・インタフェースのテストを始めたときに，販売商品の画像が適切に表示されるかテストするビットマップ・チェックポイントを作成しました。チェックポイントで，彼はテストする画像が含まれているアプリケーションの表示枠の画像をキャプチャしました。実行中にアプリケーションに表示されるグラフィック画像がチェックポイント作成時にキャプチャしたものとまったく同じに見えても，ビットマップ・チェックポイントが頻繁に失敗していることに Ben は気付きました。

Ben はテスト結果に表示された実際のビットマップ，期待されるビットマップ，および別のビットマップを見直しました。アプリケーションのユーザ・インタフェースも念入りに調べました。そのアプリケーションには 3 つの表示枠がありました。左の表示枠には一般的な情報が表示され，中央の表示枠には販売商品の画像が表示され，右の表示枠には対応する商品と詳細のリストが表示されていました。左の表示枠に表示される情報によって，中央の表示枠の画像がいずれにしても表示枠内でときどき若干ずれることに Ben は気付きました。画像自体はまったく同じですが，それらの位置の変化によってビットマップ・チェックポイントが失敗しました。

この問題に対処するために，Ben はピクセルの許容範囲を使おうとしませんでした。画像内のピクセルが同じでなければ，チェックポイントを失敗させたかったからです。

Ben がこの問題を同僚に話したとき，彼女はビットマップ・チェックポイントのカスタマイズで問題を解決できるかもしれないとアドバイスして，Joanne に照会しました。Joanne は，チェックポイントを失敗させずに画像がシフトできるピクセル数を入力として受け入れるカスタム・コンペアラを開発しました。Joanne が設計したビットマップ比較では，画像がまったく同一で，すべてが同じピクセル数だけシフトしている場合にのみ，チェックポイントが成功します。このように，チェックポイントで不正な画像をキャッチしても，画像が整列されないために，アプリケーションのインタフェースが悪くなる場合があることがわかりました。

Ben はカスタム・コンペアラを QuickTest コンピュータにインストールして登録し，ビットマップ・チェックポイントに新しいカスタム・コンペアラを選択しました。ある程度の検証作業をした後で，設定文字列を入力する最適なピクセル数に気付いたので，アプリケーションのインタフェースにかなりの変更点が検出されましたが，画像の些細なシフトでチェックポイントが失敗することはありませんでした。

Ben がこのカスタム・コンペアラをしばらく使用した後で，会社はそのコンペアラをすべての QuickTest コンピュータにインストールして登録することにしました。このカスタム・コンペアラは現在，品質保証チームの誰もが利用でき，同様の状況に使用されています。

カスタム・ビットマップ・コンペアラの開発

カスタム・コンペアラを開発するには，QuickTest ビットマップ・チェックポイント・コンペアラのインタフェース（1012 ページを参照）を実装して次のタスクを実行する COM オブジェクトを作成します

- QuickTest から入力を受け取り，ビットマップ比較を実行する。
- 比較結果を QuickTest に提供する。
- （オプションで）ユーザがビットマップ・チェックポイントを作成または編集するときに，［ビットマップ・チェックポイントのプロパティ］ダイアログ・ボックスに表示される情報を提供する。

詳細については，1592 ページ「カスタム・コンペアラの開発方法」を参照してください。

QuickTest でカスタム・コンペアラを認識するには，ビットマップ・コンペアラに定義されているコンポーネント・カテゴリに登録する必要があります。カスタム・コンペアラの実装方法に応じて，コンペアラをインストールするときに登録されるように設計するか，インストール時に実行する必要がある追加プログラムを提供することができます。詳細については，1010 ページ「カスタム・コンペアラのインストールと QuickTest への登録」を参照してください。

カスタム・コンペアラの作成方法と使用方法を学習するには，1016 ページ「チュートリアル：カスタム・コンペアラの作成」のチュートリアルを実行します。このときに，ほぼ同じ方法で独自のカスタム・コンペアラを作成できます。

QuickTest には，カスタム・コンペアラを開発するときに参照またはテンプレートとして利用できるサンプルのカスタム・コンペアラが用意されています。詳細については，1027 ページ「ビットマップ・チェックポイント・カスタマイズ・サンプルの使い方」を参照してください。

カスタム・コンペアラの開発方法

開発している COM オブジェクトで，QuickTest に用意されているタイプ・ライブラリ（**＜ QuickTest インストール・フォルダ＞¥dat¥BitmapCPCustomization¥BitmapComparer.tlb** に格納）を参照し，次の項で説明する タスクを実行するインタフェースを実装します。

- 1008 ページ「入力の受け取りとビットマップの比較」
- 1008 ページ「比較結果を QuickTest に提供」
- 1009 ページ「[ビットマップ チェックポイントのプロパティ] ダイアログ・ボックスへの情報提供」

入力の受け取りとビットマップの比較

QuickTest によって，**IVerifyBitmap** インタフェースの **CompareBitmaps** メソッドが呼び出され（1012 ページを参照），期待されるビットマップと実際のビットマップが比較のためにカスタム・コンペアラに渡されます。以下を実行する **CompareBitmaps** メソッドを実装します。

- テスト要件に基づいて定義した所定のアルゴリズムに応じて，2 つのビットマップを受け取り，比較する。
- QuickTest ユーザが提供する設定情報が含まれているテキスト文字列を受け取り，それを比較で使用する。たとえば，文字列に公差仕様，画像のサイズまたは位置の許容偏差，比較に影響を与えるその他の情報が含まれていることがあります。

  文字列には任意の形式を選択できます（XML，カンマ区切り，INI ファイル形式など）。カスタム・コンペアラに提供するドキュメントには，必ず形式を記述してください。[ビットマップ・チェックポイントのプロパティ] ダイアログ・ボックスで QuickTest ユーザが入力する設定入力は，この形式に準拠する必要があります。

比較結果を QuickTest に提供

ビットマップ・チェックポイントの結果は［テスト結果］ウィンドウに表示されます。**IVerifyBitmap** インタフェース（1012 ページを参照）を実装してビットマップを比較する場合は，以下の情報を返す必要もあります。

- ビットマップが一致して，チェックポイントが成功するか。

➤ QuickTest でテスト結果に表示されるテキスト文字列。

この文字列の目的は，比較に関する情報を QuickTest ユーザに提供することですが，独自のコンペアラを開発してテストするときは，この文字列をデバッグにも利用できます。

➤ 実際のビットマップと期待されるビットマップの違いを視覚的に示すビットマップ。

このビットマップの目的は，ビットマップが失敗した理由を QuickTest ユーザが理解するのを支援するためです。カスタム・コンペアラは，選択した視覚化アプローチでこのビットマップを作成できます。たとえば，標準の QuickTest コンペアラでは，2 つの画像で異なっているすべてのピクセルに黒ピクセルが含まれた白黒ビットマップを作成します。

［ビットマップ チェックポイントのプロパティ］ダイアログ・ボックスへの情報提供

QuickTest ユーザが［ビットマップ チェックポイントのプロパティ］ダイアログ・ボックスでカスタム・コンペアラを選択すると，［**設定オプション**］テキストボックスと，オプションでカスタム・コンペアラに用意されたドキュメントへのリンクが表示されます。詳細については，591 ページ「［ビットマップ チェックポイントのプロパティ］ダイアログ・ボックスのカスタム・コンペアラ・オプション」を参照してください。これらのオプションをサポートするには，**IBitmapCompareConfiguration** インタフェース（1014 ページを参照）を実装して以下を提供します。

➤ ［ビットマップ チェックポイントのプロパティ］ダイアログ・ボックスの［**設定オプション**］ボックスに表示される標準設定文字列。

この文字列の形式は，コンペアラが入力として期待している設定文字列の形式と同じにする必要があります。

➤ QuickTest ユーザが［ビットマップ チェックポイントのプロパティ］ダイアログ・ボックスからアクセスできるコンペアラに関するドキュメント。

このドキュメントには，任意の形式を選択できます。QuickTest はユーザのコンピュータで，提供されたファイル形式と関連付けられたプログラムを使ってこのドキュメントを開きます。そのため，このドキュメントは，QuickTest ユーザが必要なプログラムを持っていると期待される形式で提供してください。

このドキュメントは QuickTest ユーザに次の情報を提供します。

- ➤ カスタム・コンペアラが実行する比較のタイプ（ユーザはそれを使ってビットマップ・チェックポイントを実行する場合を判断できます）。
- ➤ 設定文字列に要求される形式と，それに含むことができる値。
- ➤ テスト結果に表示される比較結果情報の説明（テキスト文字列と差異ビットマップ）。

カスタム・コンペアラのインストールと QuickTest への登録

開発するカスタム・コンペアラは，それを使用するビットマップ・チェックポイントが含まれているコンポーネントを実行するコンピュータにインストールする必要があります。

カスタム・コンペアラをインストールすると，QuickTest ユーザに提供するドキュメントが，**GetHelpFilename** メソッドで指定した場所に格納されます（詳細については，1015 ページ「GetHelpFilename メソッド」を参照してください）。

さらに，カスタム・コンペアラとして作成する COM オブジェクトを QuickTest で認識するには，QuickTest ビットマップ・コンペアラのコンポーネント・カテゴリに登録する必要があります。

COM オブジェクトをコンポーネント・カテゴリに登録するには，COM オブジェクトの **HKEY_CLASSES_ROOT¥CLSID¥ ＜オブジェクトの CLSID ＞ ¥Implemented Categories** キーの下にレジストリ・キーとして関連するコンポーネント・カテゴリ ID を登録します。

コンポーネント・カテゴリ ID は **HKEY_CLASSES_ROOT¥Component Categories** キーの下に登録する必要があります。QuickTest をインストールすると，この記憶場所に QuickTest ビットマップ・コンペアラのコンポーネント・カテゴリ ID がレジストリ・キーとして追加されます。

QuickTest ビットマップ・コンペアラ，**CATID_QTPBitmapComparers** のコンポーネント・カテゴリ ID は，**＜ QuickTest インストール・フォルダ＞ ¥dat¥BitmapCPCustomization¥ComponentCategory.h** に定義されています。

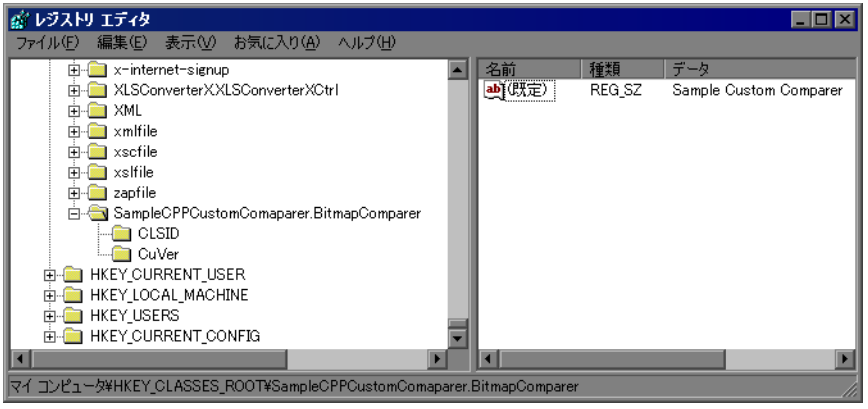
カスタム・コンペアラを設計するときには，そのコンペアラを QuickTest コンピュータにインストールすると，QuickTest ビットマップ・コンペアラのコンポーネント・カテゴリにも登録されるようにしてください。これはさまざまな方法で実現できます。

たとえば，次のような場合です。

- Microsoft Visual Studio を使って C++ でカスタム・コンペアラを開発する場合は，**DllRegisterServer** および **DllUnregisterServer** メソッドを変更して，この登録を処理できます。**regsvr32.exe** プログラムを使って DLL を実行すると，これらのメソッドが呼び出されます。このような実装の例は，「チュートリアル：カスタム・コンペアラの作成」のステップ 6（1024 ページ）を参照してください。
- 登録メソッドを変更できない環境でカスタム・コンペアラを開発する場合は，その登録を処理するプログラムを追加し，カスタム・コンペアラをインストールするユーザにそのプログラムも実行するよう指示できます。このような実装の例は，QuickTest に用意されている Visual Basic のカスタム・コンペアラサンプルで見ることができます。詳細については，1027 ページ「サンプル・カスタム・コンペアラの登録」を参照してください。

カスタム・コンペアラ名の設定

QuickTest では，［ビットマップ チェックポイントのプロパティ］ダイアログ・ボックスと［テスト結果］ウィンドウにカスタム・コンペアラの名前が表示されます。QuickTest で使用される名前は，（レジストリで）HKEY_CLASSES_ROOT キーの下にあるカスタム・コンペアラ ProgID キーの標準プロパティの値です。たとえば，下の画像で，カスタム・コンペアラの名前は **Sample Custom Comparer** です。



- Microsoft Visual Studio を使って C++ でカスタム・コンペアラを開発する場合は，［ATL 簡易オブジェクト ウィザード］の［**タイプ**］ボックスでこの名前を指定できます。

➤ Visual Basic でカスタム・コンペアラを開発する場合，この値は自動的に COM オブジェクトの ProgID に設定されます。カスタム・コンペアラを変更する場合は，コンペアラをインストールしてから，手作業でレジストリを編集するか，インストールと登録を実行して，この値を編集するプログラムを設計します。

ビットマップ・チェックポイント・コンペアラのインタフェース

カスタム・コンペアラは，この項で述べるインタフェースを実装する必要があります。カスタム・コンペアラを使用するビットマップ・チェックポイントを作成または実行するときに，QuickTest はこれらのインタフェース・メソッドを呼び出します。

IVerifyBitmap インタフェース

チェックポイントのビットマップ比較を実行する **CompareBitmaps** メソッドを実装します。

CompareBitmaps メソッド

CompareBitmaps メソッドは，ビットマップチェックポイントで比較する必要がある実際のビットマップと期待されるビットマップと，カスタム・コンペアラの設定入力を含めることができる文字列を受け取ります。

このメソッドは，カスタム・コンペアラの設計目的とされた比較アルゴリズムに応じてビットマップを比較し，その結果を QuickTest に戻す必要があります。

結果には次のものが含まれます。

➤ ビットマップが一致して，チェックポイントが成功するかどうかの表示。

➤ ビットマップ比較の結果に関する情報が含まれているテキスト文字列。

➤ 実際のビットマップと期待されるビットマップの違いを反映するビットマップ。

このメソッドが返す結果は［テスト結果］ウィンドウに表示されます。詳細については，730 ページ「ビットマップ・チェックポイント結果の分析」を参照してください。

メソッド構文：

```
HRESULT CompareBitmaps      ([in] IPictureDisp* pExpected,
                             [in] IPictureDisp* pActual,
                             [in] BSTR bstrConfiguration,
                             [out] BSTR* pbstrLog,
                             [out] IPictureDisp** ppDiff,
                             [out, retval] VARIANT_BOOL* pbMatch);
```

メソッド・パラメータ：

- **pExpected**：画像オブジェクト（入力）。

  チェックポイントに保管されている期待されるビットマップ。

- **pActual**：画像オブジェクト（入力）。

  テストするアプリケーションからキャプチャした実際のビットマップ。

- **bstrConfiguration**：テキスト文字列（入力）。

  カスタム・コンペアラへの設定入力が含まれている文字列。これは［ビットマップ チェックポイントのプロパティ］ダイアログ・ボックスの［**設定オプション**］ボックスに表示される文字列です。

  この文字列は，後述の **GetDefaultConfigurationString** メソッドでカスタム・コンペアラが QuickTest に提供する標準設定文字列，または QuickTest ユーザは入力する入力文字列になります。

  bstrConfiguration 文字列には任意の形式を選択できます（XML，カンマ区切り，ini ファイル形式など）。**GetDefaultConfigurationString** メソッドによって返される標準設定文字列は，**CompareBitmaps** メソッドに期待される形式に一致するようにしてください。さらに，カスタム・コンペアラに用意するドキュメントでは，QuickTest ユーザが［**設定オプション**］ボックスでこの文字列を編集するときに使用する必要がある形式についても説明してください。

- **pbstrLog**：テキスト文字列（出力）。

  ビットマップ比較の結果に関する情報が含まれている文字列。QuickTest では，この文字列が［テスト結果］ウィンドウに表示されます。

➤ **ppDiff**：画像オブジェクト（出力）。

実際のビットマップと期待されるビットマップの違いを反映するビットマップ（カスタム・コンペアラで作成）。QuickTest では，このビットマップが実際のビットマップおよび期待されるビットマップと一緒に［テスト結果］ウィンドウに表示されます

➤ **pbMatch**：ブール値（出力）。

ビットマップが一致して，チェックポイントが成功するかどうかを示す値。

取り得る値：

VARIANT_TRUE：実際のビットマップと期待されるビットマップが一致し，チェックポイントが成功。

VARIANT_FALSE：実際のビットマップと期待されるビットマップが一致せず，チェックポイントが失敗。

戻り値：

このメソッドが戻す HRESULT は，比較が正常に実行されたかどうかを示します（ビットマップが一致したかどうかではありません）。

IBitmapCompareConfiguration インタフェース

このインタフェースには，QuickTest で［ビットマップ チェックポイントのプロパティ］ダイアログ・ボックスに表示されるカスタム・コンペアラ・オプションをサポートするメソッドを実装します。詳細については，585 ページ「ビットマップ・チェックポイントのプロパティ・ダイアログ・ボックス」を参照してください。

GetDefaultConfigurationString メソッド

GetDefaultConfigurationString メソッドは，標準設定文字列をカスタム・コンペアラに返す必要があります。設定文字列の詳細については，1008 ページ「入力の受け取りとビットマップの比較」を参照してください。

QuickTest では，新しいビットマップ・チェックポイントを作成するユーザがカスタム・コンペアラを選択すると，［ビットマップ チェックポイントのプロパティ］ダイアログ・ボックスの［**設定オプション**］ボックスにこの文字列が表示されます。

QuickTest ユーザがこのダイアログ・ボックスで設定文字列を変更しないと，**GetDefaultConfigurationString** の提供する文字列がカスタム・コンペアラの

CompareBitmaps メソッドに渡されます。そのため，カスタム・コンペアラが **CompareBitmaps** メソッドで受け取る予定の形式に標準設定文字列が一致しているか確認する必要があります。

メソッド構文：

```
HRESULT GetDefaultConfigurationString ([out, retval] BSTR* pbstrConfiguration);
```

GetHelpFilename メソッド

GetHelpFilename メソッドは，カスタム・コンペアラに関する情報が含まれているドキュメントへのパスを QuickTest ユーザに返す必要があります。

QuickTest では，ユーザが［ビットマップ チェックポイントのプロパティ］ダイアログ・ボックスでカスタム・コンペアラを選択し，［**詳細**］をクリックすると，このドキュメントが表示されます。カスタム・コンペアラをインストールしたら，提供するドキュメントが **GetHelpFilename** メソッドで指定した記憶場所にインストールされているか確認します。

パスは次のいずれかになります。

- ファイルのフルパス。
- ファイルの相対パス（QuickTest は**＜ QuickTest インストール・フォルダ＞ ¥bin** に対してこのパスを検索します）。
- URL

カスタム・コンペアラにドキュメントを提供しない場合は，このメソッドが HRESULT E_NOTIMPL を返します。提供する情報タイプの詳細については，1009 ページ「［ビットマップ チェックポイントのプロパティ］ダイアログ・ボックスへの情報提供」を参照してください。

メソッド構文：

```
HRESULT GetHelpFilename ([out, retval] BSTR* pbstrFilename);
```

チュートリアル：カスタム・コンペアラの作成

このチュートリアルでは，Microsoft Visual Studio を使って C++ でカスタム・コンペアラを作成するプロセスを段階的に案内します。作成するカスタム・コンペアラは，QuickTest に付属しているサンプルのカスタム・コンペアラと似ています。独自のカスタム・コンペアラも同じ方法で作成できます。サンプルのカスタム・コンペアラに関する詳細は，1027 ページ「ビットマップ・チェックポイント・カスタマイズ・サンプルの使い方」を参照してください。

この項の指示に従って，次のような COM オブジェクトを作成します。

- 比較する 2 つのビットマップと設定文字列を受け取り，2 つのビットマップ（のサイズ）を比較し，必要な結果を返す **CompareBitmaps** メソッドを実装する。
- ［ビットマップ チェックポイントのプロパティ］ダイアログ・ボックスに表示される情報を返す，**GetDefaultConfigurationString** メソッドと **GetHelpFilename** メソッドを実装する。
- QuickTest ビットマップ・コンペアラのコンポーネント・カテゴリに登録する。

カスタム・コンペアラの設計が完了したら，インストールして登録し，QuickTest で使用してビットマップ・チェックポイントを実行できます。

注：チュートリアルを実行するのに使用する Microsoft Visual Studio のバージョンに応じて，コマンド名が異なることもあります。

ビットマップ・チェックポイント用のカスタム・コンペアラの作成を練習するには，次の手順を実行します。

1 **新しい ATL プロジェクトの SampleCPPCustomComparer を作成します。**

a Microsoft Visual Studio で，［**新規作成**］＞［**プロジェクト**］を選択します。［新規プロジェクト］ダイアログ・ボックスが開きます。

b［**ATL プロジェクト**］テンプレートを選択し，プロジェクトの［**名前**］ボックスに SampleCPPCustomComparer を入力して，［**OK**］をクリックします。［新規 ATL プロジェクト］ウィザードが開きます。

c［**アプリケーション設定**］で，［**属性**］オプションが選択されていないことを確認し，［**完了**］をクリックします。

2 新規クラスの CBitmapComparer を作成します。

a クラス・ビューで，［**SampleCPPCustomComparer**］プロジェクトを選択して右クリックし，［**追加**］＞［**クラス**］を選択します。［クラスの追加］ダイアログ・ボックスが表示されます。

b［**ATL Simple Object**］を選択し，［**追加**］をクリックします。［ATL 簡易オブジェクト ウィザード］が開きます。

c［**省略名**］ボックスに，BitmapComparer と入力します。ウィザードはこの名前を使って，クラス，インタフェース，および作成されるファイルの名前を作成します。

d［**タイプ**］ボックスに，Sample Custom Comparer と入力します。これは，［ビットマップ チェックポイントのプロパティ］ダイアログ・ボックスとテスト結果に表示されるカスタム・コンペアラの名前です。詳細については，1011 ページ「カスタム・コンペアラ名の設定」を参照してください。

e［**完了**］をクリックします。ウィザードは，**CBitmapComparer** クラスの実装に伴う **.cpp** および **.h** ファイルなど，追加したクラスに必要なファイルを作成します。

3［CBitmapComparer class implements the bitmap checkpoint comparer interfaces］を定義します。

a クラス・ビューで，［**CBitmapComparer**］を選択して右クリックし，［**追加**］＞［**インタフェースの実装**］を選択します。［インタフェースの実装］ウィザードが開きます。

b［**Implement interface from**］オプションで，［**ファイル**］を選択します。QuickTest ビットマップ・チェックポイントのコンペアラ・タイプ・ライブラリの場所を参照するか入力します。このタイプ・ライブラリは，**＜ QuickTest インストール・フォルダ＞ ¥dat¥BitmapCPCustomization¥ BitmapComparer.tlb** にあります。

このウィザードには，選択したタイプ・ライブラリで使用できるインタフェース，**IBitmapCompareConfiguration** と **IVerifyBitmap** が表示されます。

c 実装するインタフェースのリストに 2 つのインタフェースを追加し，［**完了**］をクリックします。

BitmapComparer.h ファイルには，ウィザードによって，インタフェースを実装するのに必要な宣言，クラス，およびメソッド・スタブが追加されます。後続のステップで，これらのメソッド・スタブに実装を追加する必要があります。

注：Microsoft Visual Studio 2005 では，ウィザードによって，**IVerifyBitmap** インタフェースで **CompareBitmaps** メソッドの署名が不正に生成されます。プロジェクトを正しくコンパイルできるようにするには，最後の引数のタイプ（**pbMatch**）を BOOL* から VARIANT_BOOL* に手作業で変更します。

---

4 **ビットマップ・チェックポイント・コンペアラ・インタフェース・メソッドの関数本体を BitmapComparer.h から BitmapComparer.cpp に移動します。**

a **BitmapComparer.h** および **BitmapComparer.cpp** ファイルを開きます。

b **BitmapComparer.h** では，（ウィザードによって作成された関数本体に基づいて）ビットマップ・チェックポイント・コンペアラ・インタフェース・メソッドの宣言，**CompareBitmaps**，**GetDefaultConfigurationString**，および **GetHelpFilename** を作成します。

c ウィザードによってビットマップ・チェックポイント・コンペアラ・インタフェース・メソッドに作成された関数本体を **BitmapComparer.h** ファイルから **BitmapComparer.cpp** ファイルに移動します。

このステップの最後には，**BitmapComparer.cpp** および **BitmapComparer.h** に次のコードが含まれます。

```
// BitmapComparer.cpp : Implementation of CBitmapComparer
#include "stdafx.h"
#include "BitmapComparer.h"

// CBitmapComparer
// IBitmapCompareConfiguration Methods
STDMETHODIMP CBitmapComparer::GetDefaultConfigurationString
                                                (BSTR * pbstrConfiguration)
{
        return E_NOTIMPL;
}
STDMETHODIMP CBitmapComparer::GetHelpFilename(BSTR * pbstrFilename)
{
        return E_NOTIMPL;
}

// IVerifyBitmap Methods
STDMETHODIMP CBitmapComparer::CompareBitmaps
                                (IPictureDisp * pExpected, IPictureDisp * pActual,
                                BSTR bstrConfiguration, BSTR * pbstrLog,
                                IPictureDisp * * ppDiff, VARIANT_BOOL * pbMatch)
{
        return E_NOTIMPL;
}
```

```
// BitmapComparer.h : Declaration of the CBitmapComparer
#pragma once
#include "resource.h"       // main symbols
#include "SampleCPPCustomComparer.h"
// CBitmapComparer
class ATL_NO_VTABLE CBitmapComparer :
        public CComObjectRootEx<CComSingleThreadModel>,
        public CComCoClass<CBitmapComparer, &CLSID_BitmapComparer>,
        public IDispatchImpl<IBitmapComparer, &IID_IBitmapComparer,
                        &LIBID_SampleCustomComparerLib, /*wMajor =*/ 1, /*wMinor =*/
0>,
        public IDispatchImpl<IBitmapCompareConfiguration,
                        &__uuidof(IBitmapCompareConfiguration),
                        &LIBID_BitmapComparerLib, /* wMajor = */ 1, /*wMinor =*/ 0>,
        public IDispatchImpl<IVerifyBitmap, &__uuidof(IVerifyBitmap),
                        &LIBID_BitmapComparerLib, /* wMajor = */ 1, /*wMinor =*/ 0>
{
    public:
        CBitmapComparer()
        {
        }
        DECLARE_REGISTRY_RESOURCEID(IDR_BITMAPCOMPARER)
        BEGIN_COM_MAP(CBitmapComparer)
                COM_INTERFACE_ENTRY(IBitmapComparer)
                COM_INTERFACE_ENTRY2(IDispatch, IBitmapCompareConfiguration)
                COM_INTERFACE_ENTRY(IBitmapCompareConfiguration)
                COM_INTERFACE_ENTRY(IVerifyBitmap)
        END_COM_MAP()
        DECLARE_PROTECT_FINAL_CONSTRUCT()
        HRESULT FinalConstruct()
        {
                return S_OK;
        }
        void FinalRelease()
        {}
        // IBitmapCompareConfiguration Methods
public:
        STDMETHOD(GetDefaultConfigurationString)(BSTR * pbstrConfiguration);
        STDMETHOD(GetHelpFilename)(BSTR * pbstrFilename);
        // IVerifyBitmap Methods
public:
        STDMETHOD(CompareBitmaps)(IPictureDisp * pExpected,
                         IPictureDisp * pActual, BSTR bstrConfiguration, BSTR *
pbstrLog,
                        IPictureDisp * * ppDiff, VARIANT_BOOL * pbMatch);
};
OBJECT_ENTRY_AUTO(__uuidof(BitmapComparer), CBitmapComparer)
```

5 ビットマップ・チェックポイント・コンペアラ・インタフェース・メソッドを実装し，必要に応じてビットマップ・チェックポイントをカスタマイズします。

このチュートリアルでは，QuickTest に付属しているサンプルのカスタム・コンペアラと似たカスタム・コンペアラを実装します。サンプルのカスタム・コンペアラに関する詳細は，1027 ページ「ビットマップ・チェックポイント・カスタマイズ・サンプルの使い方」を参照してください。

カスタム・コンペアラを作成する場合は，これがカスタム・コンペアラ・ロジックを設計するステップになります。カスタム・コンペアラで受け取れる設定値，ビットマップを比較するのに使用するアルゴリズム，および提供される出力を定義します。

BitmapComparer.cpp ファイルには，#include <atlstr.h> を追加し，ビットマップ・チェックポイント・コンペアラ・インタフェース・メソッドを次のように実装します。

➤ **GetDefaultConfigurationString** メソッド：

```
STDMETHODIMP CBitmapComparer::GetDefaultConfigurationString
                                                (BSTR * pbstrConfiguration)
{
        CComBSTR bsConfig("MaxSurfAreaDiff=140000");
        *pbstrConfiguration = bsConfig.Detach();
        return S_OK;
}
```

➤ **GetHelpFilename** メソッド：（この PDF からコードをコピーして貼り付ける場合は，ファイル名文字列から改行とタブを取り除いてください。）

```
STDMETHODIMP CBitmapComparer::GetHelpFilename(BSTR * pbstrFilename)
{
        CComBSTR bsFilename
("..¥¥samples¥¥BitmapCPSample¥¥CPPCustomComparer¥¥
                        SampleComparerDetails.txt");
        *pbstrFilename = bsFilename.Detach();
        return S_OK;
}
```

注：GetHelpFilename メソッドが相対パスを返すと，QuickTest は **＜QuickTest インストール・フォルダ＞ ¥bin** に対してこのパスを検索します。上記の実装によって，QuickTest は CPP サンプル・カスタム・コンペアラに付属してるドキュメント・ファイルを使用するよう指示されます。

➤ **CompareBitmaps** メソッド：

```
STDMETHODIMP CBitmapComparer::CompareBitmaps
                                        (IPictureDisp * pExpected, IPictureDisp *
pActual,
                                        BSTR bstrConfiguration, BSTR * pbstrLog,
                                        IPictureDisp * * ppDiff, VARIANT_BOOL *
pbMatch)
{
        HRESULT hr = S_OK;
        if (!pExpected || !pActual)
                return S_FALSE;
        CComQIPtr<IPicture> picExp(pExpected);
        CComQIPtr<IPicture> picAct(pActual);

        // Try to get HBITMAP from IPicture
        HBITMAP HbmpExp, HbmpAct;
        hr = picExp->get_Handle((OLE_HANDLE*)&HbmpExp);
        if (FAILED(hr))
                return hr;
        hr = picAct->get_Handle((OLE_HANDLE*)&HbmpAct);
        if (FAILED(hr))
                return hr;
        BITMAP ExpBmp = {0};
        if( !GetObject(HbmpExp, sizeof(ExpBmp), &ExpBmp) )
                return E_FAIL;
        BITMAP ActBmp = {0};
        if( !GetObject(HbmpAct, sizeof(ActBmp), &ActBmp) )
                return E_FAIL;

        CString s, tol;
        tol = bstrConfiguration;
        int EPos = tol.ReverseFind('=');
        tol = tol.Right(tol.GetLength() - EPos - 1);
        int maxSurfaceAreaDiff = _ttoi(tol);
        // Set output parameters
        CComPtr<IPictureDisp> Diff(pActual);
        *ppDiff = Diff;
        int DiffPixelsNumber = abs (ExpBmp.bmHeight * ExpBmp.bmWidth -
                                        ActBmp.bmHeight * ActBmp.bmWidth);
        *pbMatch = DiffPixelsNumber <= maxSurfaceAreaDiff;
        s.Format(_T("The number of different pixels is: %d."), DiffPixelsNumber);
        CComBSTR bs (s);
        *pbstrLog = bs.Detach();
        return hr;
}
```

6 QuickTest ビットマップ・コンペアラのコンポーネント・カテゴリに登録するカスタム・コンペアラを設計します。

カスタム・コンペアラとして作成する COM オブジェクトを QuickTest で認識するには，QuickTest ビットマップ・コンペアラのコンポーネント・カテゴリに登録する必要があります。コンポーネント・カテゴリ ID は，**＜QuickTest インストール・フォルダ＞¥dat¥BitmapCPCustomization¥ComponentCategory.h** に定義されています。

この登録は，ウィザードによってプロパティの一部として作成された **SampleCPPCustomComparer.cpp** ファイルの **DllRegisterServer** および **DllUnregisterServer** メソッドに実装できます。**regsvr32.exe** プログラムを使って DLL を実行すると，これらのメソッドが呼び出されます。

a **＜QuickTest インストール・フォルダ＞¥dat¥BitmapCPCustomization** フォルダをプロジェクトのインクルード・パスに追加します。

b **SampleCPPCustomComparer.cpp** ファイルを開き，#include "ComponentCategory.h" という行を追加します。

c **SampleCPPCustomComparer.cpp** ファイルで，ウィザードによって作成された **DllRegisterServer** および **DllUnregisterServer** メソッドを変更して，次のコードを含めます。

```
STDAPI DllRegisterServer(void)
{
            // registers object, typelib and all interfaces in typelib
            HRESULT hr = _AtlModule.DllRegisterServer();

            CComPtr<ICatRegister> spReg;
            hr = spReg.CoCreateInstance
                        (CLSID_StdComponentCategoriesMgr, 0, CLSCTX_INPROC);
            if (FAILED(hr))
                        return hr;

            // register comparer to the QuickTest bitmap comparers category
            CATID catid = CATID_QTPBitmapComparers;
            hr = spReg->RegisterClassImplCategories(CLSID_BitmapComparer, 1,
&catid);

            return hr;
}
```

```
STDAPI DllUnregisterServer(void)
{
	HRESULT hr = _AtlModule.DllUnregisterServer();
	CComPtr<ICatRegister> spReg;
	hr = spReg.CoCreateInstance
			(CLSID_StdComponentCategoriesMgr, 0, CLSCTX_INPROC);
	if (FAILED(hr))
		return hr;

	// unregister comparer from the QuickTest bitmap comparers category
	CATID catid = CATID_QTPBitmapComparers;
	hr = spReg->UnRegisterClassImplCategories(CLSID_BitmapComparer, 1,
&catid);

	return hr;
}
```

これらのメソッドの第 2 セクションに注意してください。QuickTest ビットマップ・コンペアラ，**CATID_QTPBitmapComparers** のコンポーネント・カテゴリへの登録を処理します。

7 独自の DLL をコンパイルし，regsvr32.exe プログラムを使って実行します。

これで，ビットマップ・チェックポイントのカスタム・コンペアラが QuickTest で使用できます。

8 ビットマップ・チェックポイントのカスタム・コンペアラを QuickTest で使います。

ビットマップ・チェックポイントの使い方に関する詳細は，第 20 章「ビットマップの検査」を参照してください。

a QuickTest を開き，Windows の電卓アプリケーション（普通の電卓）でビットマップ・チェックポイントを作成します。

［ビットマップ 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［**コンペアラ**］オプションがあり，QuickTest 標準コンペアラまたはサンプル・カスタム・コンペアラを選択できます。

b 電卓の［表示］を［**関数電卓**］に変更します。電卓オブジェクトのサイズが大きくなります。標準の QuickTest コンペアラを使って，チェックポイントを実行します。チェックポイントが失敗します。

c チェックポイントを編集し，［**コンペアラ**］ボックスで［**サンプル カスタム コンペアラ**］を選択します。

［ビットマップ チェックポイントのプロパティ］ダイアログ・ボックスの［**設定オプション**］ボックスでは，**GetHelpFilename** メソッドによって返される標準設定文字列，MaxSurfAreaDiff=140000 を表示できます。

［**詳細**］をクリックすると，サンプル・カスタム・コンペアラに関するドキュメントが含まれたテキスト・ファイルが開きます。

この練習で設計したコンペアラは，期待されるビットマップと実際のビットマップのサイズがどの程度違うか検査し，その違いが設定文字列で定義したピクセル数より大きいと，チェックポイントが失敗します。標準設定の MaxSurfAreaDiff 値を使ってチェックポイントを実行すると，チェックポイントが成功します。別の表示に設定しても，電卓オブジェクトの全体サイズの差異が 140000 ピクセル未満になるからです（差異は約 80000 ピクセルです）。MaxSurfAreaDiff を 70000 に設定すると，チェックポイントが失敗します。

テスト結果を表示して，比較の後でカスタム・コンペアラが QuickTest に提供するテキスト文字列と差異ビットマップを確認します。

ビットマップ・チェックポイント・カスタマイズ・サンプルの使い方

QuickTest には，サンプル・カスタム・コンペアラをさまざまな言語に実装するソース・ファイルが用意されています。これらの例を研究すれば，QuickTest ビットマップ・チェックポイントのカスタマイズについて学習したり，独自のカスタム・コンペアラを開発するときに参照またはテンプレートとして利用できます。ソースファイルは C++ と Visual Basic で用意されています。2 つのプロジェクトでは同様のカスタム・コンペアラが生成されます。

サンプルは**＜ QuickTest installation folder ＞ ¥samples¥BitmapCPSample** の下に格納されています。C++ プロジェクトを開くには，Microsoft Visual Studio 2003 以降を使います。Visual Basic プロジェクトを開くには，Microsoft Visual Studio 6.0 を使います。カスタム・コンペアラをコンパイルして QuickTest コンピュータにインストールすると，カスタム・コンペアラの機能を確認できます。

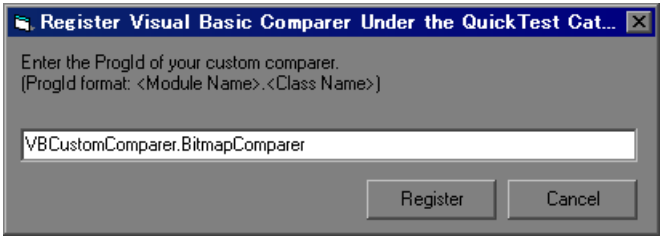
サンプル・カスタム・コンペアラの登録

カスタム・コンペアラの DLL をビルドしたら，コンピュータにインストールするために，**regsvr32.exe** プログラムで実行します。

C++ サンプル・ソースは，**DllRegisterServer** および **DllUnregisterServer** メソッドでカスタム・コンペアラを QuickTest に登録します。そのため，C++ プロジェクトを使って DLL を作成した場合は，その DLL を実行すると，カスタム・コンペアラも QuickTest ビットマップ・コンペアラのコンポーネント・カテゴリに登録されます。

カスタム・コンペアラを QuickTest ビットマップ・コンペアラのコンポーネント・カテゴリに登録しても，Visual Basic のサンプル・プロジェクトには実装されません。そのため，Visual Basic サンプルには，カスタム・コンペアラをインストールした後で，QuickTest ビットマップ・コンペアラのコンポーネント・カテゴリにカスタム・コンペアラを登録するために実行する必要がある追加ツールも含まれています。詳細については，1010 ページ「カスタム・コンペアラのインストールと QuickTest への登録」を参照してください。

＜QuickTest インストール フォルダ＞¥samples¥BitmapCPSample¥VBCustomComparer¥RegisterCategory.exe。から Visual Basic Comparer Registration Tool を実行できます（このパスを PDF からコピーして貼り付ける場合は，必ず改行を取り除いてください）



開かれるダイアログ・ボックスで，カスタム・コンペアラの ProgID を入力し，［**登録**］をクリックします。

サンプル・カスタム・コンペアラの名前

カスタム・コンペアラが QuickTest に表示される名前は，C++ プロジェクトまたは Visual Basic プロジェクトのどちらから生成するかに応じて異なります。

- C++ プロジェクトから生成される場合，QuickTest に表示されるコンペアラの名前は **Custom QTP Bitmap Comparer** になります。
- Visual Basic プロジェクトから生成される場合，コンペアラの名前は ProgId ― **VBCustomComparer.BitmapComparer** になります。

詳細については，1011 ページ「カスタム・コンペアラ名の設定」を参照してください。

サンプル・カスタム・コンペアラの機能

サンプル・カスタム・コンペアラをインストールして登録すると，QuickTest の［ビットマップ チェックポイントのプロパティ］ダイアログ・ボックスで選択し，それを使ってビットマップ・チェックポイントを実行できます。

- サンプル・コンペアラが返す（そして，［ビットマップ チェックポイントのプロパティ］ダイアログ・ボックスに表示される）標準設定文字列は MaxSurfAreaDiff=140000 です。

➤ サンプル・カスタム・コンペアラは，実際のビットマップと期待されるビットマップの内容は比較しません。それらの含まれているピクセル総数が比較されます。設定入力に関して，このコンペアラは MaxSurfAreaDiff パラメータを定義している文字列を期待しています。このコンペアラでは，ピクセル総数の差異が MaxSurfAreaDiff に定義された数より大きいとチェックポイントに失敗します。

ヒント：Windows の電卓アプリケーションでビットマップ・チェックポイントを実行すると，サンプル・コンペアラの動作をテストできます。同じオブジェクトの異なるサイズのビットマップを得るために，電卓の表示を［**普通の電卓**］または［**関数電卓**］に交互に設定します。

➤ サンプル・コンペアラに付属している（そして，［ビットマップ チェックポイントのプロパティ］ダイアログ・ボックスで開かれる）ドキュメントは，**＜QuickTest インストール・フォルダ＞¥samples¥BitmapCPSample¥CPPCustomComparer** に格納されている **SampleComparerDetails.txt** テキスト・ファイルです。

➤ テスト結果については，このサンプル・カスタム・コンペアラは，実際のビットマップを差異ビットマップとして返します。また，このコンペアラは，ピクセル総数の差異を指定するテキスト文字列を提供します。QuickTest では，この文字列がテスト結果に表示されます。

索引

A
Active Server Page テクノロジ 991
API，Windows の使用 374
API ベースのテキスト認識 604
ASP ファイル 991
attribute/ ＜プロパティ名＞表記法 373

B
Business Process Testing
　ワークフロー 35

C
CGI スクリプト 990
Close メソッド 362
COM 991
CompareBitmaps メソッド 1012
CreationTime 識別子，序数識別子 *を参照。*
CreationTime プロパティ，オブジェクトの識別に使用 205

D
Debug from Step 746
Dim ステートメント
　関数ライブラリでの 345
Do...Loop ステートメント
　関数ライブラリでの 367
Domain コマンド・ライン・オプション 713
DOS コマンド，テスト内で実行 374

E
Exit プロパティ 988
eXtensible Markup Language（XML）991

F
FAQ 985
For...Each ステートメント
　関数ライブラリでの 366
For...Next ステートメント
　関数ライブラリでの 365
FromDate コマンド・ライン・オプション 713

G
GetDefaultConfigurationString メソッド 1012
GetHelpFilename メソッド 1012
GetROProperty メソッド 371

H
HP Micro Player 703
HP Quality Center，Quality Center *を参照*
HP Software Web サイト 21
HP ソフトウェアの Web サイト 21

I
IBitmapCompareConfiguration インタフェース 1012
If...Then...Else ステートメント
　関数ライブラリでの 369
IntelliSense 327, 382
IVerifyBitmap インタフェース 1012

J
JavaScript 914

L
Location 識別子，序数識別子 *を参照。*
Log コマンド・ライン・オプション 713

M
Mercury Tours，サンプル・アプリケーション 39
META タグ 990
MinSize コマンド・ライン・オプション 714

N
Name コマンド・ライン・オプション 714

O
Object プロパティ，実行環境オブジェクト・メソッド 373
OCR 604
Option Explicit ステートメント 424

P
Password コマンド・ライン・オプション 715
Project コマンド・ライン・オプション 715

Q
QCUtil オブジェクト 929
Quality Center 921
  Connectivity Add-in 929
  QuickTest からの接続 46, 923
  QuickTest との統合 929, 949
  依存関係タブ 944
  関連付けられている関数ライブラリ 402
  コンポーネントのためのスナップショットの取得 655
  手動による不具合の報告 718
  切断 927
  相対パス 940
  バージョン・コントロール 967
  バージョン・コントロール管理 968
  ライブラリ・タブ 942
  履歴タブ 943
Quality Center OTA 929
Quality Center からの切断 927
Quality Center への QuickTest の接続 46, 923
Quality Center への接続 - サーバへの接続ダイアログ・ボックス 924
QuickTest
  アクセス権，必要な 38
  ウィンドウ，QuickTest ウィンドウ*を参照。*
  オートメーション・オブジェクト・モデル 911
  開始 43, 44
  製品情報 96
  ソフトウェアの更新 40
  レイアウト 797
    カスタマイズ 797
QuickTest Professional のバージョン情報ウィンドウ 96
QuickTest アセット・ビューア 962
QuickTest ウィンドウ
  オートメーション・ツールバー 67
  欠落リソース表示枠 61
  使用可能なキーワード表示枠 57
  情報表示枠 60
  ステータス・バー 49
  挿入ツールバー 67
  タスク表示枠 64
  タブの移動 798
  ツール・ツールバー 68
  テーマ 50
  デバッグ・ツールバー 47
  デバッグ・ビューア表示枠 59
  表示ツールバー 68
  標準ツールバー 66
  標準のレイアウトの復元 809
  表示枠の移動 798
  表示枠の自動非表示 807
  複数のドキュメント 824
  プロセス・ガイダンス表示枠 62
  編集ツールバー 67
  メニュー・バー 47
  リソース表示枠 63
  ルック・アンド・フィール 50
  レイアウトのカスタマイズ 797
QuickTest オートメーション・リファレンス 918
QuickTest 内のステップへ移動，テスト結果ウィンドウから 695
QuickTest のアセット・バージョン 968
QuickTest の開始 44

R
Recursive コマンド・ライン・オプション 715
RegisterUserFunc ステートメント 419
RGB の許容範囲，ビットマップ・チェックポイント 580

S
Server コマンド・ライン・オプション 716
SetTOProperty メソッド 149

Set ステートメント
　関数ライブラリでの 345
SGML 991
Silent コマンド・ライン・オプション 716
SystemUtil.Run メソッド 362

T
TestDirector，Quality Center *を参照*
Test コマンド・ライン・オプション 716

U
Unicode 25
UnregisterUserFunc ステートメント 409, 419, 421
UntilDate コマンド・ライン・オプション 717
Update Run ダイアログ・ボックス 790
User コマンド・ライン・オプション 717

V
VBScript 914
　関連付けられている関数ライブラリ Quality Center 402
　構文 343
　構文エラー 348
　構文を自動的に拡張 333, 383
　テキストの書式設定 347
　ドキュメント 363
VBScript 構文を自動的に拡張する 333, 383
Visual Basic 914
Visual C++ 914
Visual Studio.NET 914

W
Web
　Web オブジェクトへのキーボード・コマンドの送信 992
Web Web
　高度な問題，FAQ 990
Web コンテンツ，動的な 988
While ステートメント，関数ライブラリにおける 368
Windows API 374
Windows コマンド・ライン・オプション 712
Windows ダイアログ・ボックス 824
wscript.exe 915

X
XML
　オブジェクト・リポジトリからエクスポート 258
　オブジェクト・リポジトリとしてインポート 257

あ
アイコン
　大きいまたは小さい表示 821
アクセス権
　Quality Center に必要な 38
　QuickTest の実行に必要な 38
アセット
　使用する利点 941
　定義 938
　バージョン・コントロールからのチェックアウト 971
　バージョン・コントロールへのチェックイン 974
　バージョン・コントロールへの追加 971
アセット比較ツール 952
　色設定ダイアログ・ボックス 961
　ショートカット・メニュー・コマンド 960
　凡例 959
　開く 953
アセット・ビューア 962
　オプション 964
　長いドキュメント 965
　開く 962
値
　オブジェクトのプロパティの指定 176
　オブジェクトのプロパティ値の表示 143
　オブジェクト・プロパティの標準設定の復元 178, 181
　出力 538, 593
　出力のキャンセル 540
　出力のパラメータ化 548
　設定 617
　入力 535
　入力のパラメータ化 543
値カラム，キーワード・ビュー 520
値設定オプション・ダイアログ・ボックス 543

値セル 535
値の設定 617
値のリスト，メソッド引数 330
値のリスト，メソッド引数に使用可能な値，メソッド引数 520
アドイン
　コンポーネントへの関連付け 450
　選択の変更 451
アプリケーション
　QuickTest から起動 819
　コンポーネントに関連付けられている 657
　コンポーネントの指定 453
　実行 362
　閉じる 362
アプリケーション，サンプル 39
アプリケーションのクラッシュ・トリガ 876
アプリケーションの静止画像，キャプチャと表示 700
アプリケーションの選択ダイアログ・ボックス 455
アプリケーション・プロセスの終了 888, 891
アプリケーション領域 429
　アプリケーション領域の設定ダイアログ・ボックス 452
　一般表示枠 446
　ウィンドウ 434
　オブジェクト・リポジトリ表示枠 465
　回復シナリオ，削除 907
　回復シナリオ，設定 456
　関数ライブラリ表示枠 460
　キーワード表示枠 472
　共有オブジェクト・リポジトリの選択 470
　コンポーネントに対する変更 503
　削除 443
　作成 433
　設定 445, 446, 452
　説明 52, 430
　定義 36
　バージョンの比較 952
　開く 436, 437
　保存 439
アプリケーション領域の設定ダイアログ・ボックス
　アプリケーション表示枠 453
　リカバリ表示枠 456
アプリケーション領域の変更ダイアログ・ボックス 504
アプリケーション領域の保存ダイアログ・ボックス 441
アプリケーション領域を開くダイアログ・ボックス 437, 443
アプリケーション領域を開くボタン 436
暗号化，パスワード 536

い
依存関係
　使用グリッド 946
　使用者グリッド 946
　使用する利点 941
　定義 938
依存関係およびリソースモデル
　利点 941
依存関係タブ，Quality Center 944
　使用グリッド 947
　使用者グリッド 946
一次リポジトリ 262
一次リポジトリ表示枠 268
一般オプション 381
一般＞テキスト認識表示枠 604
一般表示枠，アプリケーション領域 446
移動ダイアログ・ボックス 334
色
　オブジェクト・リポジトリ結合ツールでの設定 279
　オブジェクト・リポジトリ比較ツールでの設定 312
　キーワード・ビューでの設定 557
色設定ダイアログ・ボックス，アセット比較ツール 961
印刷
　関数ライブラリ 399
　コンポーネント 505
印刷ダイアログ・ボックス
　テスト結果ウィンドウ 706
印刷プレビュー・ダイアログ・ボックス 704
インデックス識別子，序数識別子 *を参照。*
インデックス・プロパティ
　オブジェクトの識別に使用 203
　プログラム的記述 358
インポート

XML ファイルからオブジェクト・リポジトリ 257

え
エージェント，リモート 930
エキスパート・ビュー
一般カスタマイズ・オプション 381
テキストの検索 338
テキストの置換 339
エクスポート
オブジェクト・リポジトリから XML ファイルへ 258
画面レコーダ・ムービー 703
共有オブジェクト・リポジトリへのローカル・オブジェクト 139
エディタ・オプション・ダイアログ・ボックス 381
エラー，VBScript の構文 348

お
オートメーション
Application オブジェクト 916
オブジェクト・モデル 911
オブジェクト・リポジトリ 259
開発環境 914
言語 914
タイプ・ライブラリ 914
定義 912
オートメーション・エンジニア，ビジネス・プロセス・テストにおける役割 27, 37
オートメーション・ツールバー，QuickTest ウィンドウ 67
オブジェクト
オブジェクト・リポジトリからの削除 167
識別 103
テスト・オブジェクト*を参照。*
ドラッグ・アンド・ドロップ 57, 831
ナビゲートして学習の使用による追加 242
認識 193
表示操作 103
プロパティ，実行環境 372
メソッド，実行環境 372
オブジェクト・スパイ 119, 122
オブジェクトの状態トリガ 876
オブジェクトの選択画面 880
オブジェクトの選択ダイアログ・ボックス 529
オブジェクトの認識
自動スクリプトの生成 208
標準設定の復元 207
オブジェクトの認識ダイアログ・ボックス 195
オブジェクトのプロパティ値
指定または変更 176
表示 143
標準設定の復元 178, 181
オブジェクトのプロパティと値の設定画面 883
オブジェクトの割り当てダイアログ・ボックス 219
オブジェクト・フィルタの定義ダイアログ・ボックス 158
オブジェクト・プロパティ，ネイティブ・メソッド 373
オブジェクト・モデル
オートメーション 911
定義 912
オブジェクト・リポジトリ
XML からのインポート 257
XML へのエクスポート 258
アプリケーション領域への追加 468
以前のバージョンから変換 234
オートメーションの使用による管理 259
オブジェクトの検索 172
オブジェクトのコピー，貼り付け，および移動 164
オブジェクトの削除 167
オブジェクトの追加 152
管理 224
共有 115
欠落 845
作成 233
選択 114
閉じる 237
開く 234
変更 240
保存 235
未割り当て 850
ローカル 114
ローカル・オブジェクトのエクスポート 139

ローカル・オブジェクトのエクスポートと置換 139
オブジェクト・リポジトリ・ウィンドウ 129
オブジェクトの詳細領域 136
オプション 134
テスト・オブジェクトの詳細 175
について 128
フィルタ・ツールバー 133
編集ツールバー 131
ボタン 131
オブジェクト・リポジトリ結合ツール 261
一次リポジトリ表示枠 268
色の設定 279
ウィンドウ 264
解決方法のオプション表示枠 268
矛盾の解決方法の設定 277
ターゲット・リポジトリのフィルタ処理 296
ターゲット・リポジトリ表示枠 266
二次リポジトリ表示枠 268
ビューの変更 266
矛盾 291
矛盾の解決 294
オブジェクト・リポジトリにオブジェクトを追加ダイアログ・ボックス 155
オブジェクト・リポジトリの種類 112
オブジェクト・リポジトリ比較ツール 301
色の設定 312
ウィンドウ 304
相違点のタイプ 311
統計情報 316
リポジトリの同期化 319
リポジトリのフィルタ処理 318
リポジトリ表示枠 305
オブジェクト・リポジトリ表示枠，アプリケーション領域 465
オブジェクト・リポジトリ・マネージャ 226
オプション・ダイアログ・ボックス 630
一般＞テキスト認識表示枠 604
一般表示枠 632
実行＞画面キャプチャ表示枠 642
実行表示枠 638
スクリプトの生成オプション 632, 917
ノード 630
フォルダ表示枠 635
オンライン・ドキュメント 18
オンライン・リソース 21

か
解決方法のオプション表示枠，オブジェクト・リポジトリ結合ツール 268
ガイドライン
ユーザ定義関数 424
開発環境 914
回復後のテスト実行オプション画面 894
回復後のテスト実行のオプション 867
回復シナリオ 867
アプリケーション領域からの削除 907
アプリケーション領域での設定 456
欠落の削除 853
欠落の特定 852
コピー 904
削除 903
バージョンの比較 952
ファイル 871
プロパティの設定 906
プロパティの表示 902, 906
変更 903
保存 899
無効化 907
回復シナリオ・ウィザード 874
オブジェクトの選択画面 880
オブジェクトのプロパティと値の設定画面 883
回復後のテスト実行オプション画面 894
回復シナリオ・ウィザードの完了画面 898
回復操作画面 887, 888
回復操作 - 関数の呼び出し画面 893
回復操作 - プロセスの終了画面 891
回復操作 - ボタン，またはキーを押す画面 889
関数の呼び出し画面 893
テスト実行エラー画面 884
トリガ・イベントの選択画面 876
名前と記述画面 896
プロセスの終了画面 891
プロセスの選択画面 885
ボタン，またはキーを押す画面 889
ポップアップ・ウィンドウの条件を指定画面 878

回復シナリオ・ウィザードの完了画面 898
回復シナリオ・マネージャ・ダイアログ・ボックス 871
回復操作 867
  Microsoft Windows の再起動 888
  アプリケーション・プロセスの終了 888
  関数呼び出し 888
  キーボードまたはマウスの操作 888
学習，オブジェクト 242
拡張，ビットマップ・チェックポイント，「ビットマップ・チェックポイントのカスタマイズ」*を参照*
各分野のエキスパート，ビジネス・プロセス・テストにおける役割 27, 38
カスタマイズ
  ツールバーとメニュー 811
カスタマイズ・ダイアログ・ボックス 811
  オプション・タブ 821
  コマンド・タブ 814
  ツール・タブ 819
  ツールバー・タブ 817
カスタム・コンペアラ
  インストール 1010
  作成 1007
  説明を付ける 1009
  登録 1010
  「ビットマップ・チェックポイントのカスタマイズ」を参照
画像，コンポーネントのための取得 655
画面ショット，「スナップショット」*を参照*
画面レコーダ・タブ，テスト結果ウィンドウ 701
カラム，キーワード・ビューでの表示 555
環境
  関連付けも*参照*する コンポーネントへのアドインの
環境，コンポーネントのために表示 657
関数
  コード
    仕上げ 417
    挿入 417
  ユーザ定義 389
関数コードのプレビュー 416
関数定義ジェネレータ 408
  関数コードのプレビュー 416
  関数に説明を付ける 414
  関数の定義 408
  関数の登録 409
  説明 405
  開く 406
関数に説明を付ける 414
関数の登録 409
関数呼び出し
  ドラッグ・アンド・ドロップ 57, 831
関数ライブラリ 389
  アプリケーションの実行 362
  アプリケーションの終了 362
  アプリケーション領域 462, 463
  エレメントの強調表示 384
  管理 391
  関連付けの変更 404
  関連付けられているものを使用した作業 402
  現在のものを関連付け 403
  コンポーネントに関連付けられている 659
  実行の一時停止 748
  説明 54
  定義 37
  テキストの検索 338
  テキストの置換 339
  テスト 392
  デバッグ 398, 739
  ナビゲーション 396
  バージョンの比較 952
  開く 392, 401
  プログラミング 325
  編集 397
  保存 394
  見映えのカスタマイズ 379
  読み取り専用，編集 398
関数ライブラリ表示枠，アプリケーション領域 460
関連 ID，使用グリッド 947
関連詳細，使用グリッド 948
関連タイプ，使用グリッド 947
関連付け
  関数ライブラリ 402, 403, 404
  コンポーネントへのアドインの 450
関連名，使用グリッド 948

き
キーの割り当て
エキスパート・ビューでの 385
関数ライブラリでの 385
キーボード・コマンド，Web オブジェクトへの送信 992
キーボード・ショートカット
エキスパート・ビューでの 385
関数ライブラリでの 385
キーワード・ビューでの 554
キーワード
管理（アプリケーション領域）472
キーワード・ビュー 51, 515, 518
カラム，説明 520
カラム，表示 555
キーボードのキー 554
ステップ，削除 553
ステップ，変更 541
定義 36
表示オプション 555
フォントと色 557
キーワード表示枠 472
キーワード表示枠（アプリケーション領域の）
カラム内容の並べ替え 477
カラムのフィルタリング 475
技術情報 21
記述的プログラミング，プログラム的記述を*参照*
記述，テスト・オブジェクト 107
テスト・オブジェクトを*参照*。
矛盾の解決，オブジェクト・リポジトリ結合ツール 294
矛盾の解決方法
結合したオブジェクト・リポジトリ 294
設定，オブジェクト・リポジトリ結合ツール 277
共有オブジェクト・リポジトリ 112, 115
Quality Center への追加 468
管理 465
結合 261
選択 470
バージョンの比較 952
比較 301
未割り当て 850
ローカル・リポジトリからの更新 283
共有オブジェクト・リポジトリ・ウィンドウ 231
共有オブジェクト・リポジトリの保存ダイアログ・ボックス 299
共有オブジェクト・リポジトリ・パラメータの割り当てダイアログ・ボックス 144
共有オブジェクト・リポジトリを開くダイアログ・ボックス 470
記録
時間，向上 994

く
クッキー 990
グローバル・コンポーネントのオプション 629

け
計算
関数ライブラリでの 364
結果，「実行結果」*参照*
結果の詳細タブ，テスト結果ウィンドウ 684
結果の場所タブ，実行ダイアログ・ボックス 674
結合
共有オブジェクト・リポジトリ 261
ローカル・オブジェクト・リポジトリ 283
欠落リソース 845
欠落リソース表示枠 61
説明 845
フィルタ処理 847
未割り当ての共有オブジェクト・リポジトリ 850
未割り当てのリポジトリ・パラメータ 854
権限
Quality Center に必要な 38
QuickTest の実行に必要な 38
言語 914
言語サポート，Unicode 25
検索ダイアログ・ボックス
オブジェクト・リポジトリ結合ツール 297
オブジェクト・リポジトリ比較ツール 320
関数ライブラリ 338
テスト結果 697

検索と置換ダイアログ・ボックス 168

こ
更新を確認 632
構文エラー，VBScript 348
項目カラム，キーワード・ビュー 520
項目セル 525
項目，選択
　アプリケーションからの 529
　共有オブジェクト・リポジトリからの 527
　項目リストからの 526
項目リスト 526
コマンド
　オブジェクト・リポジトリ結合ツール 271
　オブジェクト・リポジトリ比較ツール 308
　ツールバーまたはメニューに追加 814
コマンド・ライン・オプション
　Domain 713
　FromDate 713
　Log 713
　MinSize 714
　Name 714
　Password 715
　Project 715
　Recursive 715
　Server 716
　Silent 716
　Test 716
　UntilDate 717
　User 717
　を使用したテスト結果の削除 712
コメント
　関数ライブラリ 363
　コンポーネント 519, 551
コメント・タブ，タスク表示枠 840
コレクション，プロパティの，プログラム的記述*を参照*
コンポーネント 481
　Quality Center での管理 921
　アプリケーション領域の変更 503
　印刷 505
　関連付けられている関数ライブラリ 659
　キーワード・ビュー 482
　更新 787
　作成 483
　実行結果*実行結果*参照
　実行結果。*実行結果を参照*
　実行の一時停止 748
　手動 36, 500
　スクリプト。スクリプト・コンポーネントを*参照*
　ステータス 653
　ステップ
　　移動 553
　　管理 553
　　削除 553
　　追加 523
　ステップからの実行 671
　設定の定義 647
　定義 36
　デバッグ 739
　バージョンの比較 952
　ビジネス・コンポーネントからスクリプト・コンポーネントへの変換 514
　開く 488
　保存 494
コンポーネント実行結果。*実行結果を参照*
コンポーネントの実行
　Update Run ダイアログ・ボックス 790
　期待結果の更新 787
　結果の表示 689
　高度な問題 986
　実行ダイアログ・ボックス 670
　ステップから 671
コンポーネントの設定
　「アプリケーション領域」および「アプリケーション領域の設定ダイアログ・ボックス」*を参照*
　「ビジネス・コンポーネントの設定ダイアログ・ボックス」*を参照*
コンポーネントの保守 767
コンポーネント・パラメータ 521, 535, 538, 542
　出力 37
　出力のパラメータ化 548
　ステップでの使用 664
　入力 37
　入力のパラメータ化 543
　標準設定値の定義 663

コンポーネント・リソース，欠落 845

さ
サーバ
　サーバ側の接続 990
　切断，Quality Center からの 927
　リダイレクト 990
サーバのリダイレクト 990
最初にお読みください 18
削除
　オブジェクト・リポジトリのオブジェクト 167
　リポジトリ・パラメータ 249
サマリ・カラム，キーワード・ビュー 521
サンプル・アプリケーション，Mercury Tours 39

し
実行オプション，オプション・ダイアログ・ボックスの 638
実行＞画面キャプチャ表示枠 642
実行環境
　オブジェクト 372
実行環境オブジェクト 103, 107
　プロパティと操作の表示 119
実行環境オブジェクトのプロパティ，ネイティブなプロパティ*を参照。*110
実行環境オブジェクトのメソッド，ネイティブな操作*を参照。*
実行結果 679
　印刷 706
　印刷前のプレビュー 704
　検索 693, 697
　コマンド・ライン・オプションによる削除 712
　削除，テスト結果削除ツール 710
　出力値 734
　手動による不具合の報告 718
　スキーマ 719
　チェックポイント 727
　テスト結果ウィンドウ 681
　特定の実行の表示 690
　パラメータ化された値 733
　表示のカスタマイズ 719
　ファイルにエクスポート 707
　有効化とフィルタ処理 377
実行結果を分析。*実行結果を参照*
実行セッション
　一時停止 748
　回復シナリオの無効化 907
　結果の削除 710
　テスト・オブジェクトを使った作業 149
　認識プロパティの変更 149
　プログラミングによるテスト・オブジェクトの作成 149
実行セッションの一時停止 748
実行セッションのムービー
　Quality Center での結果の表示 697
　エクスポート 703
　キャプチャ 642
　キャプチャと表示 701
　キャプチャのオプション設定 642
　テスト結果からの削除 702
実行ダイアログ・ボックス 670
シナリオ，回復シナリオ*を参照*
終了，アプリケーション・プロセス 362
出力
　値 593
　プロパティ値 595
出力オプション・ダイアログ・ボックス 538, 548
出力カラム，キーワード・ビュー 521
出力，キャンセル 540
出力セル 538
出力タイプ 602
出力値
　オブジェクトのプロパティ 597
　結果の表示 734
　定義 593
　表示 595
　標準 595
　編集 595
出力パラメータのプロパティ・ダイアログ・ボックス 597
　シンプル・モード 597
手動コンポーネント 36
手動ステップ 500, 519, 551
手動テスト 560
使用可能なキーワード表示枠 57, 831
使用グリッド 946
使用者グリッド 946

ショートカット・キー
QuickTest 68
キーワード・ビューでの 554
表示 821
情報表示枠 60
ショートカット
QuickTest 68
オブジェクト・リポジトリ結合ツール 271
オブジェクト・リポジトリ比較ツール 308
関数ライブラリでの 385
メニュー項目の 68
初期化スクリプト 913
序数識別子 201
テスト・オブジェクトの指定 191
所有者 ID，使用者グリッド 946
所有者詳細，使用者グリッド 947
所有者タイプ，使用者グリッド 946
所有者名，使用者グリッド 947
新規結合ダイアログ・ボックス 281

す
スキーマ，実行結果の 719
スクリプト・コンポーネント 36, 507, 508
さくせい 510
ビジネス・コンポーネントからの変換 513, 514
スクリプトの生成オプション 917
スタート・ページ 55
ステータス，コンポーネント 653
ステータス・バー
QuickTest ウィンドウ 49
オブジェクト・リポジトリ結合ツール 269
オブジェクト・リポジトリ比較ツール 307
ステートメント，キーワード・ビューでの使用 541
ステートメントの自動補完 327, 382
ステップ
移動 553
キーワード・ビューからの削除 553
キーワード・ビューでの表示プロパティ 561
キーワード・ビューでの変更 541
コンポーネントの管理 553
削除 553
手動 519, 551
追加 523
ステップから実行 671
ステップ・コマンド 742
ステップでオブジェクトを選択ダイアログ・ボックス 527
ステップの移動 553
ステップまで実行 746
スナップショット
コンポーネント用に取得 655
スパイオブジェクト・*スパイを参照。*
スマート認識
オブジェクトの認識ダイアログ・ボックスからの有効化 206, 208
情報の分析 723
設定 209
スマート認識プロパティ・ダイアログ・ボックス 214

せ
正規表現 617
エキスパート・ビューおよび関数ライブラリでの使用 342
関数ライブラリでの使用 342
チェックポイントでの使用 619
定義 619
バックスラッシュ（¥）621
プロパティ値 618
製品情報ウィンドウ 96
製品情報ボタン 96
セッション ID 990

そ
相違点のタイプ
オブジェクト・リポジトリ比較ツール 311
操作
項目リストからの選択 525, 526, 533
実行環境オブジェクト 110
ステップ用に選択 534
テスト・オブジェクト 110
ネイティブ 110
引数 535
操作カラム，キーワード・ビュー 520

操作セル 534
相対パス
Quality Center 940
挿入ツールバー，QuickTest ウィンドウ 67
ソフトウェア・アップデート 40

た
ターゲット・リポジトリ 262
保存 299
ターゲット・リポジトリ表示枠 266
タイプ・ライブラリ 914
タイムアウト
標準チェックポイントの指定 576
タスク・エディタ・ダイアログ・ボックス 843
タスク，管理 64
タスク・タブ，タスク表示枠 837
タスク表示枠 64, 836
コメント・タブ 840
タスク・タブ 837
単一テキスト・ブロック・モード，テキスト認識 606

ち
チェックアウト・コマンド 971
チェックイン・コマンド 971, 974
チェックポイント
オブジェクト 568
失敗 767
新規追加 564
説明 563
タイプ 565
定義 563
ビットマップ 579
変更 577
チェックポイントのプロパティ・ダイアログ・ボックス
オブジェクトの検査 570
遅延，ステップの編集 327
遅延の入力，ステップの編集時 327
置換ダイアログ・ボックス
エキスパート・ビュー 339
関数ライブラリ 339

つ
ツールチップ
表示 821
ツール・ツールバー，QuickTest ウィンドウ 68
ツールバー
QuickTest ウィンドウ
オートメーション 67
挿入 67
ツール 68
デバッグ 47, 67
表示 68
標準 66
編集 67
オブジェクト・リポジトリ結合ツール 270
オブジェクト・リポジトリ比較ツール 308
表示と非表示 817
標準設定 817
ツールバーとメニュー
カスタマイズ 811
ツールバー・ボタン
テキスト・ラベルの表示 817
ツリー・ビュー，「キーワード・ビュー」*を参照*

て
停止コマンド・ショートカット・キー 640
ディスク容量，保存 994
テキスト認識 604
ガイドライン 608
サポート環境 609
使用事例シナリオ 612
単一テキスト・ブロック・モード 606
複数テキスト・ブロック・モード 607
テキスト認識表示枠，オプションダイアログ・ボックス 604
テスト
コンポーネント 767
バージョンの表示と比較 949
テスト・オブジェクト 103, 107
アプリケーション内での強調表示 171
オブジェクト・リポジトリ内でのコピー，貼り付け，および移動 164
オブジェクト・リポジトリ内の検索 168, 172
管理 151
記述プロパティの削除 190

共有オブジェクト・リポジトリの管理 465
共有オブジェクト・リポジトリの選択 470
検索 168
識別 103
実行セッションでの 149
実行セッションでの作成 149
序数識別子の指定 191
新規の定義 161
新規プロパティの定義 188
選択
アプリケーションからの 529
共有オブジェクト・リポジトリからの 527
項目リストからの 526
追加
オブジェクト・リポジトリへの 152
記述プロパティ 184
共有オブジェクト・リポジトリのプロジェクトへの 468
ドラッグ・アンド・ドロップ 128, 242
名前の変更 182
プログラム的記述を使った作成 149
プロパティ値，置換 168
プロパティ値の取得と設定 370
プロパティと操作の表示 119
プロパティの表示 143
変更
実行セッション中のプロパティ 149
実行セッションでの 149
名前 182
プロパティ 175
ローカル・リポジトリへのコピー 141
テスト・オブジェクトの選択
アプリケーションからの 529
共有オブジェクト・リポジトリからの 527
項目リストからの 526
テスト・オブジェクトの操作 110
テスト・オブジェクトのプロパティ 110
認識プロパティ*を参照。*
テスト・オブジェクトのプロパティ，認識プロパティ*を参照。*
テスト・オブジェクト・メソッド 110
テスト・オプション
すべてのテストの設定 629
テスト結果ウィンドウ 681
QuickTest 内のステップへ移動 695
画面レコーダ・タブ 701
結果の詳細タブ 684
実行結果ツールバー 686
実行結果ツリー 683
テーマ 688
見映えのカスタマイズ 688
テスト結果削除ツール 710
コマンド・ラインからの実行 712
テスト結果，「実行結果」*参照*
テスト結果ツールバー，テスト結果ウィンドウ 686
テスト結果ツリー 683
テスト実行エラー画面 884
テスト実行エラー・トリガ 876
テスト実行時間，向上 994
テスト・データベースの維持 913
テストの実行
Update Run ダイアログ・ボックス 790
期待結果の更新 787
実行ダイアログ・ボックス 670
ステップから 671
テストの設定ダイアログ・ボックス
スクリプトの生成オプション 917
デバッグ
Debug from Step 746
関数ライブラリ 398, 739
コマンドの実行 760
コンポーネント 739
式の確認 752
実行の一時停止 748
ステップまで実行 746
テスト 739
ブレークポイント
削除 750
設定 749
無効化と有効化 749
変数へのアクセス 757
デバッグ・ツールバー，QuickTest ウィンドウ 47, 67
デバッグ・ビューア 59
デバッグ・ビューア表示枠 59, 751
ウォッチ・タブ 752
コマンド・タブ 760
変数タブ 757

データ・テーブル 59
　バージョンの比較 952

と
統計情報ダイアログ・ボックス 290
　比較ツール 316
動的な Web コンテンツ 988
動的に生成される URL や Web ページ 989
ドキュメント，オンライン 18
ドッキング表示枠 807
トラブルシューティングと技術情報 21
トランザクション終了ボタン 67
トリガ
　アプリケーションのクラッシュ 876
　イベント 867
　オブジェクト状態 876
　テスト実行エラー 876
　ポップアップ・ウィンドウ 876
トリガ・イベントの選択画面 876
ドキュメントのみのオプション 560

な
ナビゲートして学習オプション 242
名前
　テスト・オブジェクトの変更 182
名前と記述画面 896

に
二次オブジェクト・リポジトリ 262
二次リポジトリ表示枠 268
入力パラメータタブ，実行ダイアログ・ボックス 676
認識プロパティ 103, 107
　表示 119
認証
　Quality Center への接続 923

ね
ネイティブな操作 110
　表示 119
ネイティブなプロパティ 110
　表示 119
ネイティブなメソッド，ネイティブな操作を参照。

の
ノード，オプション・ダイアログ・ボックス 630

は
バージョン
　比較 952
　表示と比較 949
バージョン管理 968
バージョン・コントロール 968
　アセットのチェックアウト 971
　アセットの追加 971
　コマンド 970
　チェックアウトの取り消し 975
　テストのチェックイン 974
　バージョンの履歴 976
　ベースラインの履歴 979
バージョンの比較 952
バージョンの履歴
　ベースラインの履歴との比較 982
バージョンの履歴ダイアログ・ボックス 976
場所プロパティ，オブジェクトの識別に使用 203
パスワード，暗号化 536
パスワード・エンコーダ・ダイアログ・ボックス 537
バックスラッシュ（¥）621
パフォーマンス，向上 994
パラメータ
　コンポーネント，「コンポーネント・パラメータ」*を参照*
　コンポーネントのための指定 660
　作業 542
　出力のキャンセル 540
　未割り当てのオブジェクト・リポジトリの処理 854
　リポジトリ 245
　　管理 246
　　欠落 845
　　削除 249
　　追加 247
　　変更 249
　　割り当て 144
　ローカル，「ローカル・パラメータ」*を参照*
パラメータ・オプション・ボタン 574

パラメータ化
リポジトリ・パラメータを使用するプロパティ値 251
パラメータ化された値，テスト結果の表示 733
パラメータ・タイプ
コンポーネント・パラメータ 542
ローカル・パラメータ 542
パワー・ユーザ，高度な機能 985

ひ
比較
共有オブジェクト・リポジトリ 301
引数，定義 408
ピクセルの許容範囲，ビットマップ・チェックポイント 580
ビジネス・コンポーネント，コンポーネントを*参照*
ビジネス・コンポーネントの新規作成ダイアログ・ボックス 485
ビジネス・コンポーネントの設定ダイアログ・ボックス 647
アプリケーション表示枠 657
スナップショット表示枠 655
パラメータ表示枠 660
開く 648
プロパティ表示枠 651
リカバリ表示枠 665
リソース表示枠 659
ビジネス・コンポーネントの保存ダイアログ・ボックス 497
ビジネス・コンポーネントを開くダイアログ・ボックス 490
ビジネス・プロセス・テスト 37
ビジネス・プロセス・テストでのワークフロー 35
必須プロパティ，設定 195
ビットマップ・コンペアラ，「カスタム・コンペアラ」を参照
ビットマップ・チェックポイント 579
RGB の許容範囲 580
カスタマイズ 580, 1003
作成 582
の結果の分析 730
ピクセルの許容範囲 580
変更 582
ビットマップ・チェックポイントのカスタマイズ 1003
API 1012
サンプル 1027
チュートリアル 1016
ビットマップ・チェックポイントのカスタマイズ，「ビットマップ・チェックポイント，カスタマイズ」*を参照*
ビットマップ・チェックポイントのプロパティ・ダイアログ・ボックス 585
非表示モード 932
表現
関数ライブラリでの使用 342
表示ツールバー 68
標準出力値
作成 595
指定 597
標準設定のオブジェクトの認識設定 207
標準設定のプロパティ，変更 103, 175
標準チェックポイント
結果の分析 728
タイムアウトの指定 576
標準ツールバー，QuickTest ウィンドウ 66
表示枠
移動 798
欠落リソース 61
自動非表示 807
使用可能なキーワード 57
情報 60
タスク 64
デバッグ・ビューア 59
ドッキング 807
フローティング 807
プロセス・ガイダンス 62
リソース 63
レイアウトのカスタマイズ 798

ふ
フィルタ
オブジェクトの定義 158
フィルタ処理
オブジェクト・リポジトリ比較ツールでのリポジトリ 318
ターゲット・リポジトリ 296
フィルタ・ダイアログ・ボックス

オブジェクト・リポジトリ結合ツール 296
オブジェクト・リポジトリ比較ツール 318
フィルタ・プロパティ（スマート認識）209
フォント，キーワード・ビューでの設定 557
不具合の報告
自動 718
手動 718
不具合，報告 718
テスト結果からの 718
復元，QuickTest の標準設定のレイアウト 632
複数テキスト・ブロック・モード，テキスト認識 607
複数のドキュメントを使った作業 824
ブックマーク 335
ブレークポイント
キーワード・ビューでの使用 561
削除 750
設定 749
説明 748
フロー
定義 36
フローティング表示枠 807
プログラミング
VBScript 343
関数ライブラリ 325
プログラム的記述 149, 350
インデックス・プロパティ 358
オブジェクトに対するチェックの実行 359
記述オブジェクト 354
ステートメント 352
変数 352
プロジェクト（Quality Center）
接続 46, 923
切断 927
プロセス・ガイダンス 862
開始 860
表示枠 858
プロセス・ガイダンス表示枠 62
プロセスの選択画面 885
プロパティ
CreationTime 205
インデックス 203
回復シナリオの表示 902, 906
キーワード・ビューのステップの表示 561
実行環境オブジェクト 372
テスト・オブジェクト 110
テスト・オブジェクト記述からの削除 190
テスト・オブジェクト記述の追加 184
テスト・オブジェクトの新規プロパティの定義 188
認識プロパティ*を参照。*
ネイティブ 110, 372
場所 203
標準 103, 175
プロパティ・コレクション，プログラム的記述*を参照*
プロパティ値
テスト・オブジェクト記述での指定 251
プロパティの追加と削除ダイアログ・ボックス 184
プロパティの追加と削除ダイアログ・ボックス，オブジェクトの認識 197, 214

へ
ベースライン 978
ベースラインの履歴
バージョンの履歴との比較 982
ベースラインの履歴ダイアログ・ボックス 979
編集ツールバー，QuickTest ウィンドウ 67
変数
グローバル・スコープ内で一意 424

ほ
補足プロパティ，設定 195
ボタン
ツールバーまたはメニューに追加 814
ボタンの表示ダイアログ・ボックス 813
ポップアップ・ウィンドウ・トリガ 876
ポップアップ・ウィンドウの条件を指定画面 878

め
メソッド
実行環境オブジェクト 372
新規の追加または振る舞いの変更 419
操作*を参照。*

ネイティブ 372
ユーザ定義 419
メソッドの登録 419
メソッドの登録解除，UnregisterUserFunc ステートメントの使用 423
メニュー
新規作成 814
メニュー・バー，Quick Test ウィンドウ 47
メンテナンス実行ウィザード
オブジェクトが見つかりません画面 773
オブジェクト・プロパティ値を変更画面 777
オブジェクトをリポジトリに追加画面 782
既存のオブジェクトでステップを更新する画面 780
コメントの追加画面 776
スマート認識画面 784
メンテナンス・モードのサマリ画面 786
メンテナンス実行モード 770

も
文字セット・サポート，Unicode 25

や
役割，定義 37

ゆ
ユーザ定義
関数，ユーザ定義関数*を参照*
テスト・オブジェクト，割り当て 219
プロパティへのアクセス 373
メソッド 419
ユーザ定義オブジェクト，割り当て 219
ユーザ定義関数 389
ガイドライン 424
関数定義ジェネレータ 405
関数定義ジェネレータでのコードのプレビュー 416
仕上げ 417
説明を付ける 414
追加の生成 416
ツールチップの追加 414
登録 409
優先順位
回復シナリオの設定 906
指差しマーク
使い方のヒント 120

よ
用語，QuickTest Professional 36
用語集 36
よくある質問 985
予約済みオブジェクト 402

ら
ライセンス情報 40
修正 40
ライブラリ・タブ，Quality Center 942
ライブラリ・ファイル，関数ライブラリを*参照*

り
リソース
管理 63, 827
コンポーネント内で欠落 845
定義 938
バージョン・コントロールからのチェックアウト 971
バージョン・コントロールへのチェックイン 974
バージョン・コントロールへの追加 971
リソースおよび依存関係モデル
概要 939
用語集 938
リソース表示枠 63, 827
リポジトリ，オブジェクト・リポジトリ*を参照。*
リポジトリ内を検索ボタン 576, 589, 601
リポジトリの種類 112
リポジトリの同期化
オブジェクト・リポジトリ比較ツール 319
リポジトリ・パラメータ 245
値のパラメータ化 251
管理 246
削除 249
追加 247
変更 249
割り当て 144

リポジトリ・パラメータ・ダイアログ・ボックス 251
リポジトリ・パラメータの管理ダイアログ・ボックス 246
リポジトリ・パラメータの追加ダイアログ・ボックス 247
リモート・エージェント 930
リモート・エージェントの設定ダイアログ・ボックス 932
利用可能な値，メソッド引数 330
履歴タブ，Quality Center 943

れ
レイアウト
  QuickTest ウィンドウのカスタマイズ 797
  タブの移動 798
  標準設定の復元 809
  表示枠の移動 798
レポート，フィルタの 377
レポートをエクスポート・ダイアログ・ボックス 707

ろ
ローカル・オブジェクトのエクスポートと置換 139
ローカル・オブジェクト・リポジトリ 112, 114
  エクスポートと置換 139
  結合 283
  へのオブジェクトのコピー 141
ローカル・パラメータ 520, 535, 538, 542
  出力のパラメータ化 548
  出力パラメータの定義 37
  入力のパラメータ化 543
  入力パラメータの定義 37
ローカル・オブジェクト，共有オブジェクト・リポジトリへのエクスポート 139

わ
割り当て
  未割り当てのオブジェクト・リポジトリ 850
  未割り当てのリポジトリ・パラメータ 854
  ユーザ定義オブジェクト 219
  リポジトリ・パラメータ 144